PROTON OS

Reference Manual

Table of Contents

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).

PROTON cameras can be controlled with PROTON Control, an easy-to-use PC and Mac application (see chapter 4.1). If you want to control the device via a terminal application or a custom hardware controller, see chapter 4.2 for tool recommendations and chapters 5 and following for a detailed description of the command protocol.

1.1 Supported Devices

This manual covers the following PROTON devices:

- PROTON CAM (and its RAIN, FLEX and ZOOM variants)
- PROTON 4K (and its FLEX, ZOOM and 3D variants)
- PROTON HFR

For a full list of all supported devices and instructions on how to identify them see the `system info` and `identify` commands.

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 back side 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.

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 the power and control connector.

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

2.2 Status LED

On the back of the device is an 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	**Blinks purple**, 2.5x per second (200ms on, 200ms off)	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 `system error` 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.

Some camera models come with a cooling fan installed. By default, the fan is configured to turn on once the camera’s temperature reaches **70°C**, it turns off when the temperature has fallen by 10°. The turn-on

temperature can be changed with the `system temp fan` command. Regardless of the setting the fan will always turn on once the camera reaches **85°C** to ensure stable operation.

For more details regarding the temperature commands see section 7.3.13.

2.4 Humidity Detection

Weatherproof devices like the PROTON RAIN have an additional internal temperature and humidity sensor. You can use the `system humidity` command to keep track of the current relative humidity (%RH) inside the camera.

If the humidity exceeds **80%RH** the camera should be dried out (see below). This error case is indicated by a flashing red LED and can be checked via the `system error` command. When the threshold is exceeded a permanent “humidity event” is logged. The event counter can be read with the `system humidity count` command. Please note that the camera stays in this error state until the next power cycle or reboot, even if the humidity falls below the 80%RH threshold again. This is done so that humidity issues can easily be detected by the operator.

Drying the Camera:

When high humidity is detected, the camera must be set up in a dry and warm environment. Remove the lens completely so that the front of the camera is open and keep the camera running for at least 1 hour. The heat produced by the camera will dry out any moisture that is inside it. Afterwards install the lens again as described in the operational manual. Check the `system humidity` command to confirm that humidity is now reduced.

2.5 Integrated Microphones and Cooling Fan

Some camera models feature built-in stereo microphones which are enabled by default, the audio data is embedded into the SDI signal. To enable or disable audio or change the volume see the `system audio` commands.

On devices with a cooling fan the audio quality will be reduced when the fan is turned on. If good audio quality is required, mount the camera with a good thermal connection. Additionally, the fan start temperature can be increased with the `system temp fan` command. Using a combination of good mounting and a high fan start temperature should allow for silent operation of the camera.

Caution: Using a high fan start temperature will cause the camera to be very hot. Make sure the camera cools down before touching it!

2.6 Synchronization

Some camera models have a synchronization port. This can be used to run the camera frame synchronous to another camera or an external synchronization source.

There are 3 synchronization modes available which can be selected via the `system sync` command:

- **Off:** Camera is free running and sync output disabled.
- **Master:** The camera outputs a synchronization signal on the sync port. This mode is also used on all cameras of a multi-camera rig for internal synchronization.
- **Slave:** The camera synchronizes to an external signal received on the sync port.

When the camera is switched to slave mode it will try to synchronize to the master signal for up to 10 seconds. If synchronization fails because there is no master attached or its sync signal does not match the configured video mode, the device switches into free-running operation. In that case the error codes -113 (no master signal) or -111 (invalid master signal) are returned. In the background the camera keeps trying to synchronize. Once the synchronization was successful a short interruption of the SDI signal occurs while the image pipeline is restarted synchronously.

While in slave mode every command that restarts the pipeline (e.g. `video mode`, `video phases`, `video phases packing`, `settings load` and `settings reset`) will also trigger a re-sync to the master signal so these commands may take longer to finish. If the synchronization fails, the same error codes as for the `system sync` command are returned (-111 or -113).

2.7 3D Stereo Camera Rigs

Devices like the PROTON 3D 4K consist of two cameras integrated into one housing. Both cameras share the RS485 control interface, by default the “A” camera has the ID 1 and the “B” camera has ID 2. To control both cameras at the same time the broadcast group 0 is used. Camera “A” is configured as the broadcast master by default so only the “A” camera replies:

```
→ 0 video mode 31       # Change video mode to UHDp60 on both cameras using
                          the default broadcast ID 0
← OK                    # Reply from broadcast master camera "A"
```

In the case of a misconfiguration (e.g. both devices have the same ID) they can be reset to their default settings via the fail-safe address 100. In that case both devices will reply so the output may be scrambled.

```
→ 100 settings reset all      # Reset all settings including RS485 config
← ...                         # Ignore garbage reply
→ 0 settings save             # Save settings using default broadcast ID
← OK                          # Reply from camera "A"
```

Note that for 3D rigs the “interactive” shell mode is not supported as this would make the device inaccessible due to the shared RS485 interface.

For details on device addressing and broadcasting see chapter 5.1.1.

To synchronize the video output of both cameras in the stereo rig they have an internal sync connection. The synchronization process takes a few seconds. That means when turning the camera on or switching the video mode the slave camera will take a few seconds longer to come up than the master camera. Therefore the “B” camera of the rig acts as the sync master and the “A” camera as the slave. This ensures that the slower camera (the sync slave) is the broadcast master.

For details on camera synchronization see chapter 2.6.

2.8 Error Recovery

In case the device runs into an unexpected condition several reporting and recovery mechanisms are implemented.

Should the camera lock up due to faulty firmware, instable power-supply or other unexpected errors the internal watchdog will reboot the device trying to resume operation. When this happens the status LED blinks red afterwards and the `system error` command shows that a watchdog event was logged:

```
→ 1 system error
← Watchdog: System got reset by watchdog.
← OK
```

If this does not resolve the issue the camera will enter a safe operational mode after the watchdog got triggered three times in a row. In safe mode the camera does not output any video but can be reset to default settings or a firmware update can be performed to recover the device (see below). This mode is indicated by a red LED and a critical error message in the `system error` log:

```
→ 1 system error
← Critical: System got reset by watchdog three times, try to reset settings or
upload a different firmware to recover.
← OK
```

The error flags are cleared when the camera starts successfully after a power-cycle or reboot.

2.8.1 Recovery via Settings Reset

To recover a camera which is in safe mode first try to reset the settings to their default values (see chapters 6 and 7.3.4 for details):

```
→ 1 settings reset
← OK
→ 1 settings save
```

```
← OK
→ 1 system reboot
← OK
```

Note that the reboot is needed to clear the error flags and resume normal operation.

2.8.2 Recovery via Firmware Update

If the camera is still stuck in safe mode after a settings reset, try to perform a firmware update as described in chapter 3.

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.

This process is executed automatically by the PROTON Control or PROTON Updater software which are described in the following chapters. For details on the update process see chapter 3.3.

3.1 PROTON Control

Firmware updates via PROTON Control are described in the Control Software chapter, see section 4.1.

3.2 PROTON Updater

Firmware updates are performed via the PROTON Updater application. The app is supplied with every firmware release and runs under Windows and macOS. It can be downloaded here: https://proton-camera.com/downloads/

Note: Instead of the simplified Updater you may also use the fully featured PROTON Control application for firmware updates (see chapter 4.1).

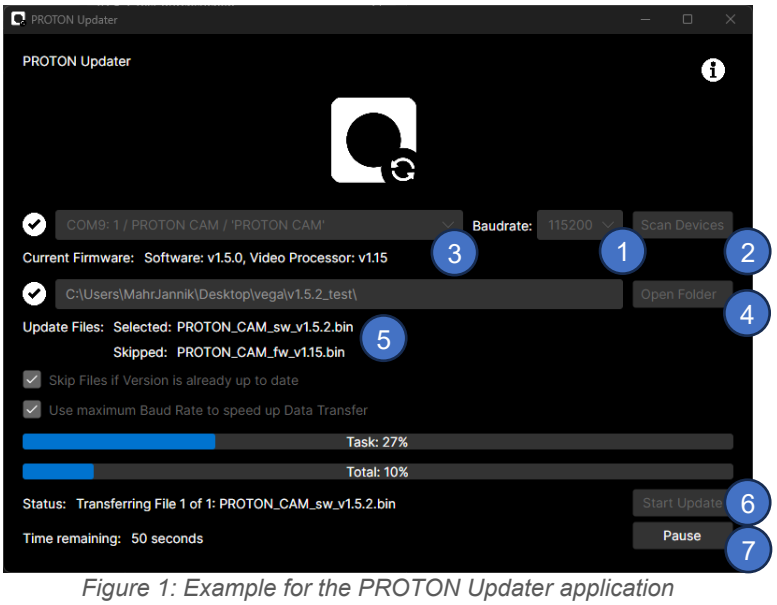

Figure 1: Example for the PROTON Updater application

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

- Connect the 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 which 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 several minutes, depending on the camera model. 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, is the update made permanent. If the camera is power cycled before all files are transferred, the progress is lost.

3.3 Update Process

This chapter is provided for reference only; users can use the PROTON Updater or PROTON Control for firmware updates which automatically perform the required steps.

3.3.1 Data Transfer

Before data transfer starts, 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.

3.3.2 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 can take **several minutes**, depending on the device. During installation, 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 the 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.

4 Control Software

4.1 PROTON Control

All features of the camera can be controlled via the PROTON Control application which runs under Windows and macOS. The app is supplied with every firmware release and can also be downloaded here: https://proton-camera.com/downloads/

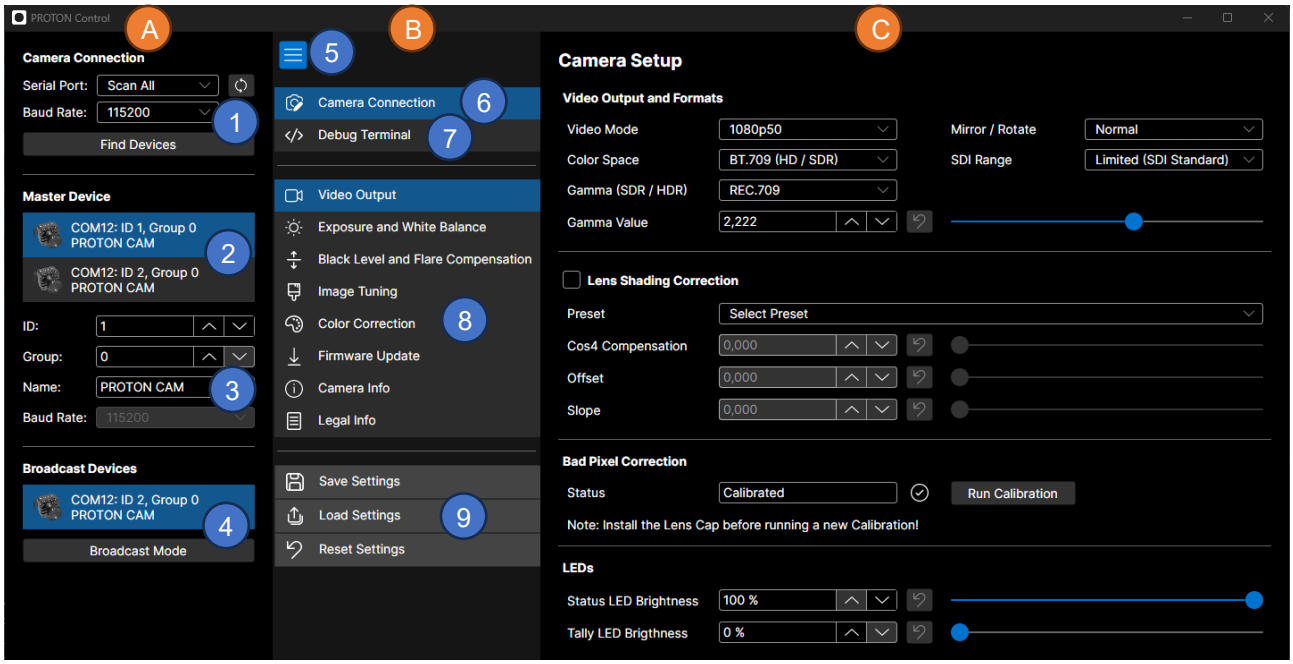

Figure 2: Example for the PROTON Control application

The application is split into three major panes which are marked with orange dots in the figure above. These are, from left to right:

(A) **Camera Connection Pane:** Find and select devices, set up serial port parameters and configure broadcasting.
(B) **Selection Pane:** Select which settings shall be displayed in the control pane and manage settings storage.
(C) **Control Pane:** Control camera settings.

Usage instructions:

- To **list available devices**, select a serial or use “Scan All” to scan on all ports. Choose a baud rate and click the “Find Devices” button **(1)**. The scan may take a few seconds depending on the amount of available serial ports.
- If only a single device is found the application automatically loads the device settings. If multiple devices are found select a master device from the list **(2)**. To **load the device settings**, move the mouse into the control pane **(C)**. For touch displays tap on the control area.
- In section **(3)** you can **change the RS485 settings** of the selected device and assign a new **device name**. Note that changing the baud rate is only supported if this is the only device on the port or all devices are in the same broadcast group and broadcasting is enabled.
- For **broadcast operation** (sending commands to multiple devices on the same port) the same group address must be assigned to all devices. This can easily be done by selecting the desired broadcast devices from the list **(4)** which will add them to the broadcast group of the current master device. To remove a device from the group simply click it again. In the example above the camera with ID 2 is already in the group of the master camera with ID 1. To enable broadcasting, click the “Broadcast Mode” toggle button.
- Click the menu button **(5)** to **expand or close the settings pane**. It is closed by default to make the app more compact.
- Click the “Camera Connection” button **(6)** to **expand or close the camera connection pane** to further decrease the size of the application when it is not needed.
- **Open a debug terminal** at the bottom of the application by clicking the “Debug Terminal” button **(7)**. It can be used to manually send commands to the device. Note that the app will not track the commands sent via the terminal, so changes to the device will not be reflected in the app.

- **Select which controls shall be displayed** in the control pane **(C)** by selecting one of the tabs **(8)**.
- **Save, load and reset settings** on the camera **(9)**. The app will automatically fetch the changed settings as needed.

4.2 Terminal Applications

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

- Putty: https://www.putty.org/
- Tera Term: https://teratermproject.github.io/index-en.html
- Serial Monitor for VS Code: https://marketplace.visualstudio.com/items?itemName=ms-vscode.vscode-serial-monitor

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.

Note: Interactive mode is not supported on 3D camera rigs as it would make the devices inaccessible due to the shared RS485 interface (see chapter 2.7). Trying to switch to interactive mode on such a device will return error code -134 (operation not supported).

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 sent to the camera must be prefixed with the camera's 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 set up 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 sent 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 sent to the broadcast group are processed by all cameras 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. To completely disable broadcasting for this device set the special address -1.

After the broadcast address has been configured select the broadcast master using the `system rs485 broadcast_master` command. The following example shows how to set up the broadcast group 10 which consists of three cameras with 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 the local echo must be turned off and the terminal program should support VT100 emulation for the best user experience (see chapter 4.2 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 sent 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 curser 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 CRLF (Carriage Return and Line Feed or "\r\n") is received. Then the command is evaluated and, if valid, executed.

During command execution no new commands should be sent until the device responds. Once execution is done the camera replies with the command's result (nothing or a string) followed by either `OK` or, in case of an error, `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:

- `0` or `1`
- `true` or `false`
- `on` or `off`
- `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 hierarchically, 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 lists the different command types.

5.2.3.1 Direct Commands

Commands that do not change a setting but execute a fixed function are called “Direct” commands. Some have no parameters like `settings save`, some do have parameters, e.g. `settings reset`.

5.2.3.2 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
```

5.2.3.3 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
```

5.2.3.4 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
```


5.2.3.5 List Commands

Some commands have a `list` subcommand (e.g. `video mode list`) which lists all valid options for this command. If the list is an enumeration (1, 2, 3, …) each value is preceded by a hash (#) so that the output cannot be confused with a command ID by other devices on the bus. Example:

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

In interactive mode a short string is added to the numbers:

```
→ video mode list
← #4 - FHD (1920x1080) p30
← #5 - FHD (1920x1080) p25
← ...
← OK
```

Some commands always add the info string even if not in interactive mode (e.g. `video lsc preset list`).

5.2.3.6 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 chapter 7.

5.2.4 Error Codes

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

Table 4: Common command error codes.

Error Code	Description
1	The help message was printed because the command was malformed.
-8	Command not found: The command is unknown and cannot be executed.
-14	An error occurred during command execution.
-19	Missing device: An internal device (e.g. temperature sensor or lens drive) is missing or does not respond.
-22 or -34	Invalid parameter value(s): The given parameters are outside the valid value range.
-28	The given parameter is too long (e.g. for the device name).
-71	Invalid number of parameters: The number of parameters is not supported by the command.
-111	Synchronization to master device failed: Invalid master signal (e.g. video mode mismatch).
-113	Synchronization to master device failed: No master signal detected (loss-of-link).
-134	Operation not supported: The requested operation is not supported by this device.
-140	Operation is currently not allowed because another setting blocks it or the 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 device 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 differently 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 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 compatiblity 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 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:

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

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

```
→ 1 system rs485 br
← FAIL -8
```

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

Note: For getter commands the device 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 the device in the future old commands may become ambiguous. Therefore, it is not recommended to use shortened commands in your controller software!

5.6 Tx Delay for slow Host Devices

The RS485 interface used to control PROTON devices only has a shared line for data Rx and Tx. By default, the camera is in Rx mode and waits for commands from a host. As soon as a command is received a reply is sent and the camera switches from Rx to Tx. Vice versa the host must switch its interface from Tx to Rx to receive the data.

In some cases, the Tx to Rx switch on the host takes some time so that replies from the camera are lost or scrambled. To work around this, use the `system rs485 tx_delay` command, e.g. to set a delay of 2000 µs (2 ms):

```
→ system rs485 tx_delay 2000
← OK
```

If a delay is set the camera ensures that all replies are delayed until at least the configured time has elapsed, e.g. if the delay is set to 2 ms the host is guaranteed to have duration of at least 2 ms to switch from Tx to Rx mode.

Note that the delay is only applied in controller mode. In interactive mode it is ignored as it would make the interactive shell unusable (see chapter 5.1 for details on the operational modes).

6 Settings Handling

The device has two types of setting 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:

- Auto-save (section 7.2.4)
- Device name (section 7.3.2)
- RS485 configuration (section 7.3.10):
    - Baud rate

  - Mode (controller / interactive)
  - Device and broadcast addresses
  - TX delay
- Synchronization to an external sync signal (section 7.3.18)

There are some settings which are considered calibration data which are never reset:

- The bad pixel table calibration data. Should you find the data to be incorrect (defect pixels are visible), recalibrate the camera as described in chapter 7.5.7.1.
- The adjustment of the SDI clock frequency can be tuned with the `system clock_tune` command described in chapter 7.3.15.4.

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-saving is enabled. For this reason, auto-saving is disabled by default and manual saving is recommended for a responsive user experience.

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

Calibration data like the defect pixel table or the SDI clock tuning are always saved automatically, regardless of auto-saving being enabled or not.

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. For details on the command types see chapter 5.2.3.
- 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	/	**Type**	Special
Description	Print a list of all available command alias.		

7.1.2 clear

Command	`clear`		
Alias	/	**Type**	Special
Description	Clear screen / terminal. Only works in interactive mode.		

7.1.3 help

Command	`help`		
Alias	/	**Type**	Special
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	/	**Type**	Special
Description	Print a list of the recently used commands.		

7.1.5 rem

Command	`rem`		
Alias	/	**Type**	Special
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	/	**Type**	Special
Description	Resize terminal output to current terminal window width.		

7.1.7 firmware

The firmware commands are used during the firmware update process and should normally not be executed manually by the user. The `firmware` command itself does not have any functionality, see subcommands below.

7.1.7.1 firmware list

Command	`firmware list`		
Alias	/	**Type**	Special
Description	List current firmware configuration.		

7.1.7.2 firmware request_upgrade

Command	`firmware request_upgrade <image_id>`		
Alias	/	**Type**	Direct
Description	Request upgrade of the image with the given ID. This command must be used *after* uploading a new image to the device, otherwise the update will not be performed.		
Parameter	`image_id`		
Type	Unsigned Integer		
Description	ID of the image that shall be upgraded.		
Valid Values	0: Upgrade software image (PROTON OS) 1: Upgrade firmware image (Video Processor)		

7.2 Settings Commands

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

7.2.1 settings save

Command	`settings save`

Alias	`save_settings`	**Type**	Direct
Description	Save current device settings to persistent storage.		

7.2.2 settings load

Command	`settings load`		
Alias	`load_settings`	**Type**	Direct
Description	Load device settings from persistent storage and apply them. Executing this command requires a restart of the image pipeline. If this camera supports synchronization and slave mode is enabled, it will return -113 (no sync signal) or -111 (invalid sync signal) when synchronization fails and the device falls back to free-running operation.		

7.2.3 settings reset

Command	`settings reset <all>`		
Alias	`reset_settings`	**Type**	Direct
Description	Resets all settings to the default value. To save settings afterwards use the `save` 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 `all` as shown below. Executing this command requires a restart of the image pipeline. If this camera supports synchronization and slave mode is enabled, it will return -113 (no sync signal) or -111 (invalid sync signal) when synchronization fails and the device falls back to free-running operation.		

Examples:

```
→ 1 settings reset            # Reset non-critical settings.
← OK
→ 1 settings reset all        # Reset all settings.
← OK
```

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

7.2.4 settings auto_save

Command	`settings auto_save <enable>`		
Alias	`auto_save`	**Type**	Getter, Setter
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	`enable`		
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	`system info`		
Alias	`version`	**Type**	Special
Description	Get system information. If called via the alias `version` 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 output:

- `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. For a full list of all devices supported by PROTON OS see the table below.
- `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 in hexadecimal format where each bit represents one mode, see table below.

Table 5: Supported Devices

Product Family	Platform String	Product
PROTON CAM	vega	PROTON CAM
	vega_rain	PROTON RAIN
	vega_flex	PROTON FLEX
	vega_zoom	PROTON ZOOM
PROTON 4K	polaris	PROTON 4K
	polaris_flex	PROTON FLEX 4K
	polaris_zoom	PROTON ZOOM 4K
	polaris_3d_a / polaris_3d_b	PROTON 3D 4K “A” and “B” camera
PROTON HFR	titan	PROTON HFR

How-To Use the Resolution Mask

The table below lists all modes which are known to PROTON OS, each camera only supports a subset of those modes. When using the `system info` command a camera will set the bits of the modes which are supported.

Example: The PROTON CAM reports the following resolution mask in hexadecimal format:

```
00000000-000007FF-00000000
```

Decoded according to Table 6:

- Left Block (HD): `0x00000000` → No bits set, none supported.
- Middle Block (2K): `0x000007FF` → Bits 1 to 11 set, all 1080p and 1080i modes supported
- Right Block (4K): `0x00000000` → No bits set, none supported

Note: A list of the supported modes per camera model can be found in the `video mode` command description. A controller may use the resolution mask or the `video mode list` command to get all supported video modes.

Table 6: Resolution Mask

Bit	HD Modes (Left Block)	FHD / 2K Modes (Middle Block)	UHD / 4K Modes (Right Block)
1	720p60[1]	1080p30	UHDp30
2	720p50[1]	1080p25	UHDp25
3	720p59[1]	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[1]
12		2Kp30[1]	4Kp25[1]
13		2Kp25[1]	4Kp24[1]
14		2Kp24[1]	4Kp23[1]
15		2Kp23[1]	4Kp29[1]
16		2Kp29[1]	4Kp50[1]
17		2Kp50[1]	4Kp60[1]
18		2Kp60[1]	4Kp59[1]
19		2Kp59[1]	4Kp48[1]
20		2Kp48[1]	4Kp47[1]
21		2Kp47[1]	

7.3.2 system name

Command	`system name <name string>`		
Alias	`name`	**Type**	Getter, Setter
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`

[1] This mode is currently not supported by any PROTON camera and reserved for future use.

Alias	`runtime`	**Type**	Pure Getter
Description	Print device runtime since boot in seconds. The counter gets reset by a power cycle or reboot.		

7.3.4 system reboot

Command	`system reboot`		
Alias	`reboot`	**Type**	Direct
Description	Reboot the device.		

7.3.5 system update

Command	`system update`		
Alias	`update`	**Type**	Direct
Description	Put the 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`	**Type**	Special
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 * 10ms ~= 1s (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
```

Note: For details on the platform string (`vega` in the above example) see the `system info` command.

7.3.7 system error

Command	`system error`		
Alias	`error`	**Type**	Special
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`	**Type**	Getter, Setter
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 device 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 ping

Command	`system ping`		
Alias	`ping`	**Type**	Special
Description	Check if device exists and replies. This command does nothing except replying with “OK”.		

7.3.10 system rs485

Command	`system rs485`		
Alias	`rs485`	**Type**	Special
Description	Setup RS485 console interface, see subcommands for details. If called without arguments prints a summary of the current settings.		

Example:

```
→ 1 system rs485
← Mode: controller - device address: 1, broadcast group: 0 (none), broadcast
master: -1 (none)
← OK
```

7.3.10.1 system rs485 mode

Command	`system rs485 mode <mode>`		
Alias	`prompt`	**Type**	Setter, Getter
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 (not allowed for cameras in a 3D rig, see chapter 2.7)		
Alias	`controller` → `system rs485 mode 0` `interactive` → `system rs485 mode 1`		
Default	0 = Controller Mode		

7.3.10.2 system rs485 device_address

Command	`system rs485 device_address <dev_addr>`		
Alias	`rs485_addr`	**Type**	Setter, Getter
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`		
Type	Unsigned Integer		
Description	Device address to set.		
Min	0		
Max	99		
Default	1 (2 for “B” cameras of a 3D rig, see chapter 2.7)		

7.3.10.3 system rs485 broadcast_address

Command	`system rs485 broadcast_address <bc_addr>`		
Alias	`rs485_bc_addr`	**Type**	Setter, Getter
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. To disable broadcasting, use the special broadcast address -1.		

Parameter	`bc_addr`
Type	Signed Integer
Description	Broadcast address to set.
Min	-1
Max	99
Default	0

7.3.10.4 system rs485 broadcast_master

Command	`system rs485 broadcast_master <dev_addr>`		
Alias	`rs485_bc_master`	**Type**	Setter, Getter
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`		
Type	Signed Integer		
Description	Device address of the device which shall become master of the broadcast group.		
Min	-1 = disable broadcast master		
Max	99		
Default	-1 = disabled (1 for the cameras of a 3D rig, see chapter 2.7)		

For an example see chapter 5.1.1.2.

7.3.10.5 system rs485 baudrate

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

7.3.10.5.1 system rs485 baudrate list

Command	`system rs485 baudrate list`		
Alias	/	**Type**	List
Description	List all supported baud rates of the RS485 interface.		

Note: Since this command does not print IDs, but explicit values, it does not use the leading hash (#) like other `list` commands.

Example:

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

7.3.10.6 system rs485 tx_delay

Command	`system rs485 tx_delay <delay_us>`		
Alias	`rs485_tx_delay`	**Type**	Setter, Getter
Description	Add a delay between receiving a command and sending the reply. This can be used to ensure that the sender of the command has enough time to perform the TX/RX switch or do other processing. Setting the value to 0 disables the delay, the maximum value is 100000 µs.		

	Note that the given delay is the minimum time before the camera sends a reply after receiving a command. The actual time can be longer depending on the device and the command. Also note that the delay is only applied in controller mode. In interactive it is ignored as it would make the interactive shell unusable.
Parameter	`delay_us`
Type	Unsigned Integer
Description	TX delay to set.
Min	0 (disabled)
Max	100000 µs
Default	0 (disabled)

7.3.11 system status_led

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

7.3.12 system tally

Command	`system tally <brightness>`		
Alias	`status_led`	**Type**	Setter, Getter
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`		
Type	Unsigned Integer		
Description	Tally light brightness in %.		
Min	0 (off)		
Max	100 (100%)		
Default	0 (off)		

7.3.13 system temp

For details on the overtemperature protection see chapter 2.3.

Command	`system temp (<sensor_idx>)`		
Alias	`temp`	**Type**	Pure Getter with optional Arguments
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 `sensor_idx` parameter. For temperature logging see subcommands. Output: `<sensor_idx> <temperature> <description>`		
Parameter	`sensor_idx`		
Type	Unsigned Integer		
Description	Optional index of the sensor		
Min	0		
Max	Depends on camera model		

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
```

7.3.13.1 system temp max

Command	system temp max		
Alias	max_temp	Type	Pure Getter
Description	Show maximum logged system temperature. It prints the following values: max_temp_user: User resettable maximum temperature in °C since boot. It can be reset with the 'temp reset' command or via a power-cycle or reboot. max_temp: Maximum temperature in °C since boot which can only be reset via power-cycle or reboot. shutdown_temp: 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 temp count command). Output: <max_temp_user> <max_temp> <shutdown_temp>		

7.3.13.2 system temp reset

Command	system temp reset		
Alias	max_temp_reset	Type	Direct
Description	Reset the user resettable maximum logged temperature (see temp max command) to the current system temperature. The temperature is also reset when the camera is power-cycled or rebooted.		

7.3.13.3 system temp count

Command	system temp count		
Alias	over_temp_count	Type	Pure Getter
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 temp max command). This counter is persistent and cannot be reset. Output: <over_temp_count>		

7.3.13.4 system temp fan

Note: This command is only available on devices with a cooling fan.

Command	system temp fan <turn_on_temp>		
Alias	fan_target	Type	Setter, Getter
Description	Set fan turn-on temperature in °C. The fan turns on when the configured temperature is reached and turns off again when the temperature is 10° below the turn-on temperature. The default turn-on temperature is 75°C. Setting the temperature to 0° permanently turns on the fan. Instead of using a temperature value one of three presets can be given: - cool: Fan always on - standard: Use default turn-on temp - silent: Only turn fan on if necessary to avoid overheating		
Parameter	turn_on_temp		
Type	Unsigned Integer		
Description	Temperature in °C at which the cooling fan turns on.		
Min	0 or cool preset (= always on)		
Max	85 or silent preset		
Default	75 or standard preset		

Note regarding the presets:

- The `standard` preset should allow for silent operation under normal operating conditions (20°C) when the camera is mounted to a metal tripod or similar mounting hardware. In this setting the camera can get hot but touching it is still possible.
- If fan noise is an issue, use the `silent` preset but make sure the camera is cooled down before touching it!
- If noise is not an issue but a cool camera is preferred, use the `cool` preset.

Despite the presents, a custom start temperature can be used to match the use case.

7.3.14 system humidity

Note: This command group is only available on weatherproof devices.

For details on the humidity detection see chapter 2.4.

Command	`system humidity`		
Alias	`humidity`	**Type**	Pure Getter
Description	Get current relative humidity inside the device in percent.		

Example:

→ `1 system humidity`
← `system humidity 53.79`

In the above example the relative humidity inside the camera (%RH) is 53.79%.

7.3.14.1 system humidity count

Command	`system humidity count`		
Alias	`humidity_count`	**Type**	Pure Getter
Description	Show number of logged humidity events. This is the number of times the device detected an internal humidity level above the safe threshold. This counter is persistent and cannot be reset. Output: `<humidity_count>`		

7.3.15 system sdi

Note: SDI commands are only available on devices with configurable SDI outputs.

Caution: These are advanced settings. We do not guarantee conformity to the SDI standard when changing any of the values in this command group.

7.3.15.1 system sdi outputs

Command	`system sdi outputs`		
Alias	`sdi_outputs`	**Type**	Pure Getter
Description	Get number of SDI output channels. The output index for the `sdi amplitude` command has a range of [0, `num_out` - 1]. Output: `<num_out>`		

7.3.15.2 system sdi amplitude

Command	`system sdi amplitude <out_idx> <amplitude>`		
Alias	`sdi_amplitude`	**Type**	Setter, Getter
Description	Tune amplitude of the SDI output(s) in percent. By default, all outputs are set to 100% for best SDI standard conformity. Depending on the cable length and SDI receiver the output signal strength can be decreased down to 0% until reception is stable (an amplitude of 0% does not mean 'no output' but 'lowest supported amplitude'). Note that the internal resolution of the amplitude can be less than 100 steps so not every change in percentage will result in a measurable difference in output amplitude. **Caution:** Reducing the amplitude decreases the achievable cable length, so only use it if needed! The first output has the index '0', the last output has index '`num_out` - 1'. The number of SDI output channels depends on the camera model and can be fetched with the `sdi`		

	`outputs` command. To set all outputs to the same value use '-1' as the `out_idx`. To list the values of all outputs at once call the command without arguments or '-1' as the `out_idx`.	
Parameter	`out_idx`	`amplitude`
Type	Unsigned Integer	Unsigned Integer
Description	Index of the SDI output channel to configure. Use -1 to set all channels, omit or use -1 to get all channels.	Output amplitude in %
Min	-1 (all channels) or 0 (first channel)	0 = lowest amplitude
Max	`num_out` – 1 (see `sdi outputs` command)	100 = maximum amplitude
Default	/	100 = maximum amplitude

Note: The `out_idx` parameter is mandatory for the setter and optional for the getter. The getter will return both the `out_idx` and the `amplitude` value.

Examples:

```
→ 1 system sdi amplitude 0 80       # Increase amplitude of output 0 to 80%
← OK
→ 1 system sdi amplitude 0          # Get amplitude of output 0
← 1 system sdi amplitude 0 80
← OK
→ 1 system sdi amplitude -1 67      # Decrease amplitude of all outputs to 67%
← OK
→ 1 system sdi amplitude -1         # Get amplitude of all outputs (in this
                                      example the device has 2 output channels)
← 1 system sdi amplitude 0 67
← 1 system sdi amplitude 1 67
← OK
→ 1 system sdi amplitude            # Get all amplitude of all outputs by not
                                      giving an index.
← 1 system sdi amplitude 0 67
← 1 system sdi amplitude 1 67
← OK
```

7.3.15.3 system sdi slew_rate

Command	`system sdi slew_rate <out_idx> <slew_rate>`		
Alias	`sdi_slew_rate`	**Type**	Setter, Getter
Description	Set SDI slew rate: **0** = Normal: Use default slew rate for all modes for maximum signal strength and cable length. **1** = Reduced UHD: Reduce the slew rate for 6G and 12G signals to improve compatibility with some SDI receivers when short SDI cables (< 1m) are used. Has no effect on 1.5G and 3G signals. **Caution:** Reducing the slew rate decreases the achievable cable length, so only use it if needed! The first output has the index '0', the last output has index '`num_out` - 1'. The number of SDI output channels depends on the camera model and can be fetched with the `sdi outputs` command. To set all outputs to the same value use '-1' as the `out_idx`. To list the values of all outputs at once call the command without arguments or '-1' as the `out_idx`. To list all slew rates that are supported by this device use the `slew_rate list` subcommand.		
Parameter	`out_idx`	`slew_rate`	
Type	Unsigned Integer	Unsigned Integer	
Description	Index of the SDI output channel to configure. Use -1 to set all channels, omit or use -1 to get all channels.	Slew Rate Mode	

Min	-1 (all channels) or 0 (first channel)	0 = Normal
Max	`num_out` – 1 (see `sdi outputs` command)	1 = Reduced UHD Slew Rate
Default	/	0 = Normal

Note: The `out_idx` parameter is mandatory for the setter and optional for the getter. The getter will return both the `out_idx` and the `amplitude` value.

For usage examples, please see the `system sdi amplitude` command which has the same syntax except for the value range of the parameter.

7.3.15.3.1 sdi slew_rate list

Command	`System sdi slew_rate list`		
Alias	/	**Type**	List
Description	List supported slew rate modes.		

7.3.15.4 system sdi clock_tune

Command	`system sdi clock_tune <ppm>`		
Alias	`sdi_clock_tune`	**Type**	Setter, Getter
Description	Fine-tune the SDI video clock frequency in ppm. The frequency of the video clock source can change slightly over time. In case you experience incompatibilities with connected devices use measuring equipment to determine the SDI clock and adjust it accordingly. The following SDI frequencies are expected: • 1.5G: 1485.00 MHz (non-fractional), 1483.52 MHz (fractional) • 3G: 2970.00 MHz (non-fractional), 2967.03 MHz (fractional) Deviations of +/- 10 ppm are within the limits of the SDI specification. The clock tuning in ppm is calculated as: $$tune = \frac{freq_{expected} - freq_{measured}}{freq_{measured}} * 10^6$$ **Example:** If the measured frequency for 3G output in a non-fractional mode (e.g. p50) is 2969.84 MHz the tuning is: $$tune = \frac{2970.00 - 2969.93}{2969.93} * 10^6 \approx 24\ ppm$$ Setting the value of 24 ppm should result in an SDI frequency of 2970 MHz. If needed the value can be further tuned up or down. Clock tuning is not possible if the system is currently configured as a sync slave (see `system sync` command). **Note:** Since this is a calibration value it is automatically saved in the persistent setting storage even if auto save is disabled. Also, this setting is not reset by the `settings reset` command.		
Parameter	`ppm`		
Type	Signed Integer		
Description	Clock frequency adjustment in ppm.		
Min	-50		
Max	50		
Default	Depends on factory calibration.		

7.3.16 system audio

Note: Audio commands are only available on devices with integrated microphones.

Note: On devices with a cooling fan, you should mount the camera with a good thermal connection and may want to increase the fan start temperature to avoid fan noise in the audio stream. For details see chapter 2.4 and the `system temp fan` command.

Command	`system audio <enable>`		
Alias	`audio_enable`	**Type**	Setter, Getter
Description	Enable stereo audio. If enabled 2 channels of audio will be embedded into the SDI signal. To change audio volume, see subcommands.		

Parameter	`enable`
Type	Boolean
Description	Enable or disable SDI audio.
Default	On

7.3.16.1 system audio gain

Command	`system audio gain <factor>`		
Alias	`audio_gain`	**Type**	Setter, Getter
Description	Set digital audio gain. Gain is given as a 4.12 fixed point number, range [0.0, 16.0) with a default of 1.0 = 4096. The digital audio gain is applied to all channels. For devices with built-in microphones this is the only gain setting available. For devices with analog audio inputs the analog gain can be set separately for each input with the `audio volume` command. Setting a gain of 0 will mute (but not disable) all audio channels in the SDI output.		
Parameter	`factor`		
Type	Unsigned 4.12 Fixed Point Number		
Description	Digital audio gain factor to set.		
Min	0.0 = 0		
Max	15.999 = 65535		
Default	1.0 = 4096		

7.3.16.2 system audio inputs

Command	`system audio inputs`		
Alias	`audio_inputs`	**Type**	Pure Getter
Description	Get number of external audio inputs. The input index for the `audio volume` and `audio bias` commands has a range of [0, `num_in` - 1]. Output: `<num_in>`		

7.3.16.3 system audio volume

Note: This command is only available on devices with at least one external audio input (`audio inputs` command returns a value >= 1).

Command	`system audio volume <in_idx> <gain>`		
Alias	`audio_volume`	**Type**	Setter, Getter
Description	Set analog audio gain for the given input. Gain is given in dB as a 6.1 fixed point number, range [0.0, 40.0] with a default of 0.0 = 0. In addition to the per input analog gain a global digital gain can be applied with the `audio gain` command. The number of analog audio inputs depends on the camera model and can be fetched with the `audio inputs` command. To list the gains for all outputs call this command without arguments.		
Parameter	`in_idx`	`gain`	
Type	Unsigned Integer	Unsigned 6.1 Fixed Point Number	
Description	Index of the audio input to configure.	Analog audio gain in dB to set.	
Min	0	0.0 dB = 0	
Max	`num_in` – 1 (see `audio inputs` command)	40.0 dB = 80	
Default	/	0.0 dB = 0	

Note: The `in_idx` parameter is mandatory for the setter and optional for the getter. The getter will return both the `in_idx` and the `gain` value.

Examples:

```
→ 1 system audio volume 0 21        # Set analog gain of input 0 to 10.5 dB
→ 1 system audio volume 0           # Read analog gain of input 0
← system audio volume 0 21
← OK
← 1 system audio volume             # List audio volume for all inputs (in this
                                      example the camera has 2 inputs)
```



```
← system audio volume 0 21
← system audio volume 1 0
← OK
```

7.3.16.4 system audio bias

Note: This command is only available on devices with at least one external audio input (`audio inputs` command returns a value >= 1).

Command	`system audio bias <in_idx> <enable>`		
Alias	`audio_bias`	**Type**	Setter, Getter
Description	If enabled a microphone bias voltage of 2.5V is enabled for the selected audio input. For line-in sources it should be disabled. The number of analog audio inputs depends on the camera model and can be fetched with the `audio inputs` command. To list the bias for all outputs call this command without arguments.		
Parameter	`in_idx`	`enable`	
Type	Unsigned Integer	Boolean	
Description	Index of the audio input to configure.	Microphone bias enable.	
Min	0	0 = Off	
Max	`num_in` - 1 (see `audio inputs` command)	1 = On	
Default	/	0 = Off	

Note: The `in_idx` parameter is mandatory for the setter and optional for the getter. The getter will return both the `in_idx` and the `enable` state.

For usage examples, please see the `system audio volume` command which has the same syntax except for the value range of the parameter.

7.3.17 system timecode

Command	`system timecode <enable>`		
Alias	`timecode_enable`	**Type**	Setter, Getter
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		

7.3.17.1 system timecode value

Command	`system timecode value <hour> <minute> <second>`		
Alias	`timecode`	**Type**	Setter, Getter
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

7.3.17.2 system timecode pause

Command	`system timecode pause <enable>`		
Alias	`timecode_hold`	**Type**	Setter, Getter
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

7.3.17.3 system timecode drop

Command	`system timecode drop <enable>`		
Alias	`timecode_drop`	**Type**	Setter, Getter
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 on non-fractional (integer) video modes.		
Parameter	`enable`		
Type	Boolean		
Description	Use ‘drop frame’ mode for fractional video modes.		
Default	Enabled		

7.3.18 system sync

Note: Synchronization commands are only available on devices with an external (or internal, in case of a 3D camera rig) synchronization input.

Command	`system sync <mode>`		
Alias	`sync`	**Type**	Setter, Getter
Description	Set synchronization mode: **0:** Off, camera is free running and sync output disabled. **1:** Master, output sync signal to slave devices, also used for internal synchronization on mutli-camera rigs. **2:** Slave, synchronize to an external sync signal. To list modes that are supported by this device use the `sync list` subcommand. Switching the device into slave mode restarts the video pipeline and tries to synchronize to the master device. If no master device is connected the operation times out and the command fails with error code -113. If an invalid master signal is detected (e.g. video mode mismatch) error code -111 is returned. In both cases the device switches to free-running mode and synchronizes to the master as soon as a valid sync signal is detected. Once the device is synchronized a short loss of the SDI output signal occurs while the video output is restarted synchronously.		
Parameter	`mode`		
Type	Unsigned Integer		
Description	Synchronization mode to set		
Valid Values	0 = Off 1 = Master 2 = Slave		
Default	Master for multi-camera devices (e.g. 3D rigs), Off for all other devices		

7.3.18.1 system sync list

Command	`system sync list`		
Alias	/	**Type**	List
Description	List supported synchronization modes.		

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`	**Type**	Setter, Getter
Description	Set camera gain as (linear gain * 1000), use `info` command to get range. To set gain to minimum or maximum value pass `min` or `max` as `value`. Setting the gain is not possible if it is currently being controlled by the auto exposure algorithm.		
Parameter	`value`		
Type	Unsigned Integer		
Description	Linear gain scaled by 1000 (see examples below).		
Min	1.0 = 1000		
Max	Depends on camera model, use `camera info` 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:

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

7.4.1.1 camera gain mode

Command	`camera gain mode <mode>`		
Alias	`gain_mode`	**Type**	Setter, Getter
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 `camera info` command. To list supported modes, use the `gain mode list` subcommand.		
Parameter	`mode`		
Type	Unsigned Integer		
Description	Gain mode to set.		
Valid Values	0 = Standard 1 = Low-Light		
Default	0 = Standard		

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

7.4.1.1.1 camera gain mode list

Command	`camera gain mode list`		
Alias	/	**Type**	List
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
```

7.4.2 camera exposure

Command	`camera exposure <value>`

Alias	`cam_exposure`	**Type**	Setter, Getter
Description	Set camera exposure time in microseconds, use `info` command to get range. To set exposure to minimum or maximum value pass `min` or `max` as `value`. Setting the exposure time is not possible if it is currently being controlled by the auto exposure algorithm.		
Parameter	`value`		
Type	Unsigned Integer		
Description	Exposure time in µs.		
Min	Depends on camera model, use `camera info` command to get range.		
Max	Depends on `video mode`, use `camera info` command to get range.		
Default	Maximum value for default `video mode`.		

7.4.3 camera info

Command	`camera info`		
Alias	`cam_info`	**Type**	Pure Getter
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>`		

7.4.4 camera auto

Command	`camera auto <enable>`		
Alias	`aec`	**Type**	Setter, Getter
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 values that have been applied by the AEC.		
Parameter	`enable`		
Type	Boolean		
Description	Enable or disable automatic exposure control.		
Default	Enabled		

Example to enable the AEC:

→ `1 camera auto 1`
← `OK`

For other AEC settings, see the subcommands in section 7.4.4.2 and following.

ProVideo Protocol Compatibility

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` setting from the ProVideo protocol is not supported and the passed value is ignored.

The following modes are supported:

- **Gain Control:** `cost_gain` > 0, `cost_tint` = `cost_apt` = 0
- **Exposure Time Control:** `cost_tint` > 0, `cost_gain` = `cost_apt` = 0
- **Combined Gain and Exposure Time Control:** `cost_gain` > 0, `cost_tint` > 0, `cost_apt` = 0
- **Iris Control:** `cost_apt` > 0, `cost_gain` = `cost_tint` = 0 (only available on devices with a motor iris)

When an unsupported combination of the cost values is used, the command returns `FAIL`.

With 1 parameter the behavior is identical to the `camera auto` command and only the enable state is changed:

```
aec <enable>
```

```
→ 1 aec 1
← OK
```

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

```
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]	`camera auto`
set_point	[256, 3000]	`camera auto target`, range gets converted to [0, 1000]
speed	[3, 30]	`camera auto speed`, 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: `camera auto mode 0`, getter returns 0 when disabled and 8000 when enabled.
cost_exp	0 + [250, 8000]	If set greater 0: `camera auto mode 1`, getter returns 0 when disabled and 8000 when enabled.
cost_apt	0 + [250, 8000]	If set greater 0: `camera auto mode 3`, getter returns 0 when disabled and 8000 when enabled.
taf	[5000, 20000]	`camera auto anti_flicker`, 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	`camera auto max_gain`
use_custom_weights	[0, 1]	`camera auto custom`

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
```

7.4.4.1 camera auto mode

Command	`camera auto mode <mode>`		
Alias	`aec_mode`	**Type**	Setter, Getter
Description	Set auto exposure mode: **0 = Gain Control:** Gain is set automatically; exposure can be set manually using the `camera exposure` command. The maximum gain used by the algorithm can be limited using the `camera auto max_gain` command. **1 = Exposure Time Control:** Exposure time is set automatically; gain is set manually using the `camera gain` command. **2 = Combined Gain and Exposure Time Control:** Both gain and exposure time are set automatically. The algorithm prefers increasing the exposure time 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 `camera auto max_gain` command. In this mode anti-flicker can be enabled with the `camera auto anti_flicker` command to limit		

	the exposure time to flicker free values when using artificial light sources. **3 = Iris Control:** The iris aperture is set automatically, gain and exposure are set manually. Only available on devices with a motor iris. To get supported modes use the `mode list` subcommand.
Parameter	`mode`
Type	Unsigned Integer
Description	Auto exposure mode to set.
Valid Values	0 = Gain Control 1 = Exposure Time Control 2 = Combined Gain and Exposure Time Control 3 = Iris Control
Default	2 = Combined Gain and Exposure Time Control

7.4.4.1.1 camera auto mode list

Command	`camera auto mode list`		
Alias	/	**Type**	List
Description	List all supported auto exposure modes.		

A controller can expect modes 0, 1 and 2 (Gain, Exposure and Combined Control) to be available on all devices. Mode 3 (Iris Control) is only available on devices with a motorized iris.

7.4.4.2 camera auto target

Command	`camera auto target <brightness>`		
Alias	`aec_target`	**Type**	Setter, Getter
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	`brightness`		
Type	Unsigned Integer		
Description	Relative brightness in percent multiplied by 10 for higher resolution.		
Min	0 = 0%		
Max	1000 = 100%		
Default	333 = 33.3%		

7.4.4.3 camera auto speed

Command	`camera auto speed <speed>`		
Alias	`aec_speed`	**Type**	Setter, Getter
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	`speed`		
Type	Unsigned Integer		
Description	AEC control speed (reaction speed to brightness changes).		
Min	0 = 0%		
Max	100 = 100%		
Default	50 = 50%		

7.4.4.4 camera auto max_gain

Command	`camera auto max_gain <value>`		
Alias	`aec_max_gain`	**Type**	Setter, Getter
Description	Set the maximum gain that the AEC algorithm is allowed to use. The gain range is identical to the `camera gain` command and can be fetched with the `camera info` command. To set the maximum gain to the minimum or maximum value pass `min` or `max` as `value`.		
Parameter	`value`		
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 `camera info` command to get range.		
Default	Maximum gain.		

7.4.4.5 camera auto anti_flicker

Command	`camera auto anti_flicker <mode>`		
Alias	`aec_anti_flicker`	**Type**	Setter, Getter
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 `anti_flicker list` subcommand.		
Parameter	`mode`		
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		

7.4.4.5.1 camera auto anti_flicker list

Command	`camera auto anti_flicker list`		
Alias	/	**Type**	List
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.

7.4.4.6 camera auto custom

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

7.4.4.7 camera auto weight

Command	`camera auto weight <index> (<weight>)`		
Alias	`aec_weight`	**Type**	Setter, Getter with optional Arguments
Description	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. 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: 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 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. These weights are only used if custom weights are enabled with the `auto custom` command, otherwise all areas are weighted equally.		
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

7.4.4.8 camera auto brightness

Command	`camera auto brightness`		
Alias	`aec_brightness`	**Type**	Pure Getter
Description	Get the mean image brightness from the auto exposure algorithm taking into consideration the configured weights (see `auto custom` and `auto weight` commands). Works regardless of the AEC being enabled or disabled so it can be used for external exposure control. The returned value is in 1.16 fixed point format with a range of [0.0 = 0, 1.0 = 4194304] where 0.0 is black and 1.0 is white.		

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`	**Type**	Setter, Getter
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 `mode list` subcommand or evaluate the resolution mask from the `system info` command. If the current exposure setting is bigger than the maximum of the new mode it will be clipped. For high-framerate cameras the resulting framerate also depends on the number of active phases, e.g. in 1080p50 mode with 3 active phases the total framerate is 3 * 50 = 150 fps. For details see the `video phases` command. Executing this command requires a restart of the image pipeline. If this camera supports synchronization and slave mode is enabled, it will return -113 (no sync signal) or -111 (invalid sync signal) when synchronization fails and the device falls back to free-running operation.		
Parameter	`mode`		
Description	Video mode ID.		
Valid Values	Depends on camera model, see Table 7 below.		
Default			

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

Table 7: Supported Video Modes.

ID	Mode	PROTON CAM (and RAIN, FLEX, ZOOM variants)	PROTON 4K (and FLEX variant)	PROTON HFR
4	1080p30	✓	✓	✓
5	1080p25	✓	✓	✓
6	1080p24	✓	✓	
7	1080p23	✓	✓	
8	1080p29	✓	✓	✓
9	1080p50	✓ Default	✓	✓ Default
10	1080p60	✓	✓	✓
11	1080i60	✓	✓	
12	1080i50	✓	✓	
13	1080i59	✓	✓	
14	1080p59	✓	✓	✓
25	UHDp30		✓	
26	UHDp25		✓	
27	UHDp24		✓	
28	UHDp23		✓	

ID	Mode	PROTON CAM (and RAIN, FLEX, ZOOM variants)	PROTON 4K (and FLEX variant)	PROTON HFR
29	UHDp29		✓	
30	UHDp50		✓ Default	
31	UHDp60		✓	
32	UHDp59		✓	
33	UHDp48		✓	
34	UHDp47		✓	

7.5.1.1 video mode list

Command	`video mode list`		
Alias	/	**Type**	List
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 phases

Note: The `phases` command and its subcommands are only available on high-framerate devices.

Command	`video phases <num>`		
Alias	`phases`	**Type**	Setter, Getter

Description	Set number of active phases for high-framerate video output.
	Since SDI has a limited maximum framerate of 60 fps the output of a high-framerate camera must be split into multiple phases. The total framerate then results from the selected video mode multiplied with the number of active phases, e.g. a video mode of 1080p50 and 3 active phases results in a framerate of 3 * 50 = 150 fps.
	Each group of images will have the same SDI timecode as they are transmitted in parallel (although they were recorded sequentially).
	Unused phases will mirror the signal of the first phase and can be used as a monitor output.
	Example output for 4 active phases:
	`SDI 1: 1, 5, 9, ...`
	`SDI 2: 2, 6, 10, ...`
	`SDI 3: 3, 7, 11, ...`
	`SDI 4: 4, 8, 12, ...`
	Example output for 2 active phases:
	`SDI 1: 1, 3, 5, ...`
	`SDI 2: 2, 4, 6, ...`
	`----------`
	`SDI 3: 1, 3, 5, ... (Mirror of SDI 1)`
	`SDI 4: 1, 3, 5, ... (Mirror of SDI 1)`
	The number of supported phases depends on the camera model, to list supported phase numbers use the `phases list` subcommand.
	Executing this command requires a restart of the image pipeline. If this camera supports synchronization and slave mode is enabled, it will return -113 (no sync signal) or -111 (invalid sync signal) when synchronization fails and the device falls back to free-running operation.		

Parameter	`num`		
Type	Unsigned Integer		

Description	Number of active phases.
Valid Values	Depends on camera model, see Table 8 below.
Default	

Table 8: Supported number of output phases.

ID	Slow-Motion Factor	PROTON HFR
1	1x (no slow-motion)	✓
2	2x	✓
3	3x	✓
4	4x	✓ Default

7.5.2.1 video phases list

Command	`video phases list`		
Alias	/	**Type**	List
Description	List supported number of phases.		

7.5.2.2 video phases packing

Command	`video phases packing <mode>`		
Alias	`phases_packing`	**Type**	Setter, Getter

Description	Control how the multiple phases of the high-framerate video output are packaged:
	0 = Single: By default, each output of the camera shows a single phase, e.g. each output shows a 1080p50 signal if this is the configured video mode.
	1 = Quad: In this mode output 1 shows a UHD image which contains all phases packed into a single frame. Output 1 is top-left, output 2 top-right, output 3 bottom-left and output 4 bottom-right:
	`---------`
	`| 1 | 2 |`
	`|---|---|`
	`| 3 | 4 |`
	`---------`
	For unused outputs a black image is inserted into the packed image, e.g. when phases are set to 3 the bottom-right part of the quad output is black.
	This mode is ideal if the signal is being compressed afterwards using a video codec as the image geometry stays intact.
	2 = 2SI: In this mode each of the four phases is directly mapped to one of the four sub images of the 12G SDI signal. For unused outputs the sub image shows a black image. This mode is ideal if the signal is transmitted uncompressed as it allows for easy decoding without a frame buffer.
	In Quad and 2SI mode all other outputs will mirror the signal of the first phase for monitoring.
	Supported modes can differ between camera models. To list supported packing modes, use the `packing list` subcommand.
	Executing this command requires a restart of the image pipeline. If this camera supports synchronization and slave mode is enabled, it will return -113 (no sync signal) or -111 (invalid sync signal) when synchronization fails and the device falls back to free-running operation.		

Parameter	`mode`		
Description	Packing mode to set.		
Valid Values	Depends on camera model, see Table 9 below.		
Default			

Table 9: Supported packing modes.

ID	Packing Mode	PROTON HFR
0	**Single:** 1 phase per output	✓ Default
1	**Quad:** 4 phases on output 1 in a 2x2 grid, other outputs mirror phase 1	✓
2	**2SI:** 4 phases on output 1, one phase per sub image, other outputs mirror phase 1	✓

7.5.2.2.1 video phases packing list

Command	`video phases packing list`		
Alias	/	**Type**	List
Description	List supported packing modes.		

7.5.2.3 video phases marker

Command	`video phases marker <enable>`		
Alias	`phases_marker`	**Type**	Setter, Getter
Description	Enable markers in each output to indicate its phase number.		
Parameter	`enable`		
Type	Boolean		
Description	Enable or disable optical markers to identify output phase.		
Default	Off		

7.5.3 video flip

Command	`video flip <mode>`		
Alias	`flip`	**Type**	Setter, Getter
Description	Flip output image. To list all modes, use the `flip list` 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		

7.5.3.1 video flip list

Command	`video flip list`		
Alias	/	**Type**	List
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.

7.5.4 video black_sensor

Command	`video black_sensor <red/all> (<green> <blue>)`		
Alias	`black_sensor`	**Type**	Setter, Getter
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		

7.5.4.1 video black_sensor red

Command	`video black_sensor red <offset>`		
Alias	`black_red`	**Type**	Setter, Getter

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

7.5.4.2 video black_sensor green

Command	`video black_sensor green <offset>`		
Alias	`black_green`	**Type**	Setter, Getter
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		

7.5.4.3 video black_sensor blue

Command	`video black_sensor blue <offset>`		
Alias	`black_blue`	**Type**	Setter, Getter
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		

7.5.5 video lsc

Command	`video lsc <enable> (<k> <radius> <slope>)`			
Alias	`lsc`	**Type**	Setter, Getter	
Description	Configure lens shading correction. The parameters `k`, `radius` 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 `lsc preset` subcommand.			
Parameter	`enable`	`k`	`radius`	`slope`
Type	Boolean	Unsigned 2.30 Fixed Point Number		
Description	Enable or disable correction.	Natural vignetting correction factor.	Artificial vignetting correction radius.	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 set up 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 `radius` and `slope` configure the artificial vignetting compensation. If the value for `radius` is increased, the radius where the compensation starts will be increased (moved to the image border). The higher the `slope`, the stronger the compensation is.

To set up 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`, `radius` 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 not be ideally illuminated. Make sure to not overcompensate, this will create a wavelike appearance of the luminescence on the monitor.
5. Set the `radius` 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 `radius` 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 off 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.

7.5.5.1 video lsc preset

Command	`video lsc preset <idx>`		
Alias	`lsc_preset`	**Type**	Setter, Getter
Description	Apply preset for selected lens. Setting a preset automatically enables the lens shade correction. Use the `preset list` 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	`idx`		
Type	Unsigned Integer		
Description	LSC preset to set.		
Valid Values	Depend on camera model, use `preset list` subcommand to get list.		
Default	-1 (no preset configured)		

7.5.5.1.1 video lsc preset list

Command	`video lsc preset list`		
Alias	/	**Type**	List
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.

Example:

```
→ 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)
← OK
```

7.5.6 video wb

Command	`video wb`		
Alias	`wb`	**Type**	Direct
Description	Run auto white balance. This command blocks until the measurement is done which takes between 500ms and 1s. Has no effect if continuous auto white balance is currently enabled. Running the white balance resets the `user gain` settings. For manual white balance and configuration of the auto white balance, see subcommands.		

7.5.6.1 video wb gain

Command	`video wb gain <red> <green> <blue>`		
Alias	`wb_gain`	**Type**	Setter, Getter
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. 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 `user gain` command to specify user gains which are applied on top of the gains from the selected color temperature or preset. This means that this command also updates the user gains and vice versa. To set a single gain value use the subcommands.		
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.		

The RGB white balance gains are linked with the user gains which can be set with the `user gain` command that means if one of them gets changed the other one is updated automatically.

The RGB gains represent the total gains which are programmed in the video processor. They are internally calculated by multiplying the calibrated gains resulting from the selected color temperature (see `wb temp` command) and the user gains:

$$Gain_{min} = \min(R_{calib}, G_{calib}, B_{calib})$$

$$R_{wb} = \frac{R_{calib}}{Gain_{min}} * R_{user}$$

$$G_{wb} = \frac{G_{calib}}{Gain_{min}} * G_{user}$$

$$B_{wb} = \frac{B_{calib}}{Gain_{min}} * B_{user}$$

Note that the calibrated gains are only used internally and are not directly visible to the user (they can be read when all user gains are at their default value of 1.0).

7.5.6.1.1 video wb gain red

Command	`video wb gain red <gain>`		
Alias	`gain_red`	**Type**	Setter, Getter

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.

7.5.6.1.2 video wb gain green

Command	`video wb gain green <gain>`		
Alias	`gain_green`	**Type**	Setter, Getter
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.		

7.5.6.1.3 video wb gain blue

Command	`video wb gain blue <gain>`		
Alias	`gain_blue`	**Type**	Setter, Getter
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.		

7.5.6.2 video wb temp

Command	`video wb temp <color_temp>`		
Alias	`wb_temp`	**Type**	Setter, Getter
Description	Set white balance temperature in Kelvin. 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 color temperature preserves the gains set with the `user gain` command. To return to a clean state they must manually be reset to 1.0.		
Parameter	`color_temp`		
Type	Unsigned Integer		
Description	Color temperature in Kelvin.		
Min	3000		
Max	10000		
Default	6500		

7.5.6.3 video wb preset

Command	`video wb preset <id>`		
Alias	`wb_preset`	**Type**	Setter, Special Getter
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 `wb temp` command instead which allows for continuous manual white balance control. Setting a preset is not possible when auto white balance is enabled. Selecting a preset automatically resets the `user gain` to 1.0.		
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.

7.5.6.4 video wb auto

Command	`video wb auto <enable>`		
Alias	`awb`	**Type**	Setter, Getter
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 `wb temp` and `wb gain` commands, it is not possible to set them manually while the automatic is enabled. Enabling the AWB resets the `user gain` settings and they cannot be changed. Use the `auto offset` 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 configure AWB behavior, see subcommands.		
Parameter	`enable`		
Type	Boolean		
Description	Enable or disable continuous auto white balance.		
Default	Enabled		

7.5.6.4.1 video wb auto speed

Command	`video wb auto speed <speed>`		
Alias	`awb_speed`	**Type**	Setter, Getter
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		

7.5.6.4.2 video wb auto offset

Command	`video wb auto offset <offset>`		
Alias	`awb_offset`	**Type**	Setter, Getter
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

7.5.6.5 video wb tint

Command	video wb tint <value>		
Alias	wb_tint	Type	Setter, Getter
Description	Set tint value for the white balance temperature calculation. This setting influences both manual white balance via the wb temp or wb preset commands and the auto white balance. The tint value is applied as a gain on the green value which is calculated from the calibration data. It is used to fine-tune the green balance if the white balance calibration does not perfectly match the used lens or lighting conditions. The value is in 1.8 fixed point format with a range of [0.8 = 205, 1.2 = 308] (1.0 = 256).		
Parameter	value		
Type	Unsigned 1.8 Fixed Point Number		
Description	Tint value (green gain)		
Min	0.8 = 205		
Max	1.2 = 308		
Default	1.0 = 256		

Setting a tint value other than 1.0 changes the formulas for the gains from chapter 7.5.6.1 as follows:

$$G_{calib,tinted} = G_{calib} * G_{tint}$$

$$Gain_{min} = \min(R_{calib}, G_{calib,tinted}, B_{calib})$$

$$G_{wb} = \frac{G_{calib,tinted}}{Gain_{min}} * G_{user}$$

Where G_{tint} is the configured tint value.

7.5.7 video bpc

Command	video bpc		
Alias	bpc	Type	Direct, Getter
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 degrade the image quality. Instead, 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.

7.5.7.1 video bpc calibrate

Command	video bpc calibrate		
Alias	dpc_auto_load	Type	Direct
Description	Run automatic bad pixel calibration which can take up to 30 seconds. For correct results this must be done with the lens cap installed! Devices with an integrated lens drive and iris control will automatically close the iris for calibration. The calibration data is considered critical, so it is not cleared by the settings reset 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! (Exception: Devices with a lens drive that supports iris control, on these the iris is closed automatically during calibration).

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.

7.5.8 video flare

Command	`video flare <red/all> (<green> <blue>)`		
Alias	`flare`	**Type**	Setter, Getter
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		

7.5.8.1 video flare red

Command	`video flare red <offset>`		
Alias	`flare_red`	**Type**	Setter, Getter
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		

7.5.8.2 video flare green

Command	`video flare green <offset>`		
Alias	`flare_green`	**Type**	Setter, Getter
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		

7.5.8.3 video flare blue

Command	`video flare blue <offset>`		
Alias	`flare_blue`	**Type**	Setter, Getter
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		

7.5.9 video color_space

Command	`video color_space <id>`

Alias	`color_space`	**Type**	Setter, Getter
Description	Set color space (also known as gamut). To list all color spaces, use the `color_space list` subcommand.		
Parameter	`id`		
Type	Unsigned Integer		
Description	Color space to set.		
Valid Values	0 = BT.709 (HD / SDR) 1 = BT.2020 (UHD / HDR) 2 = Sony S-Gamut3 3 = Sony S-Gamut3.Cine		
Default	0 = BT.709 (HD / SDR)		

7.5.9.1 video color_space list

Command	`video color_space list`		
Alias	/	**Type**	List
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.

7.5.10 video sdi_range

Command	`video sdi_range <range>`		
Alias	`sdi_range`	**Type**	Setter, Getter
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 `lut mode` 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	`range`		
Type	Unsigned Integer		
Description	SDI range to set.		
Valid Values	0 = Limited (SDI Standard) 1 = Full (Maximum Dynamic)		
Default	0 = Limited		

7.5.11 video filter

Command	`video filter <enable>`		
Alias	`filter_enable`	**Type**	Setter, Getter
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	`enable`		
Type	Boolean		
Description	Enable or bypass post processing filters.		
Default	On		

7.5.11.1 video filter detail

Command	`video filter detail <level>`		
Alias	`filter_detail`	**Type**	Setter, Getter
Description	Set detail (sharpening) level of the post processing filter from 0 to 100%.		
Parameter	`level`		
Type	Unsigned Integer		
Description	Detail level in %.		
Min	0		
Max	100		

Default	10

7.5.11.2 video filter denoise

Command	`video filter denoise <level>`		
Alias	`filter_denoise`	**Type**	Setter, Getter
Description	Set denoise level of the post processing filter from 0 to 100%.		
Parameter	`level`		
Type	Unsigned Integer		
Description	Denoise level in %.		
Min	0		
Max	100		
Default	0		

7.5.12 video mcc

Command	`video mcc <enable>`		
Alias	`mcc`	**Type**	Setter, Getter
Description	Enable multi matrix color correction (0 = bypass, 1 = enabled). Disabling the MCC disables blinking and gating (see `mcc blink` and `mcc gate` subcommands).		
Parameter	`enable`		
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.

7.5.12.1 video mcc mode

Command	`video mcc mode <mode>`		
Alias	`mcc_opmode`	**Type**	Setter, Getter
Description	Set MCC operating mode which defines the number of active phases. To get supported modes use the `mcc mode list` subcommand. Changing the operational mode disables blinking and gating (see `mcc blink` and `mcc gate` subcommands).		
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.

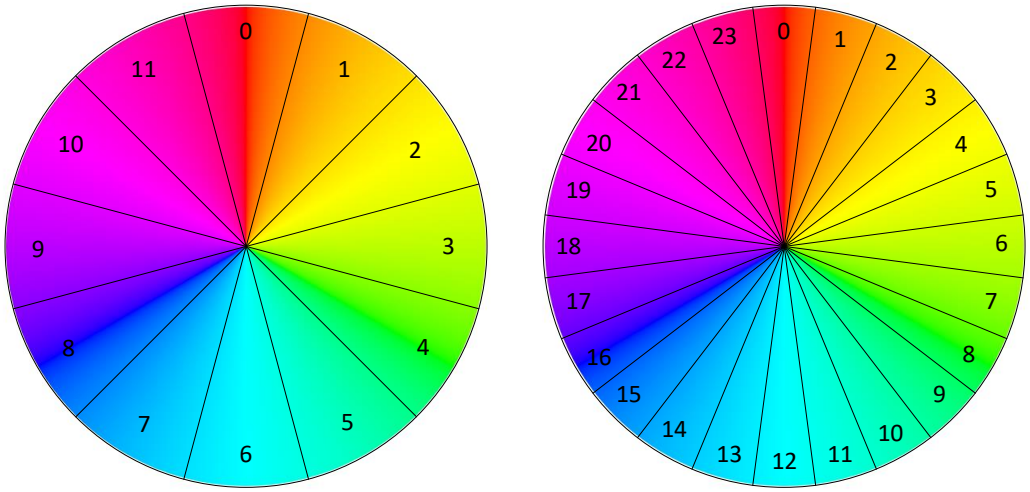

Figure 3: Example of 12 and 24 MCC Phases.

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

7.5.12.1.1 video mcc mode list

Command	`video mcc mode list`		
Alias	/	**Type**	List
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.

7.5.12.2 video mcc phase

Command	`video mcc phase <id> (<saturation> <hue>)`		
Alias	`mcc_set`	**Type**	Setter, Getter with optional Arguments
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 3. 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 results in a +/- 180° rotation. The example below shows how the hue works for phase 0:

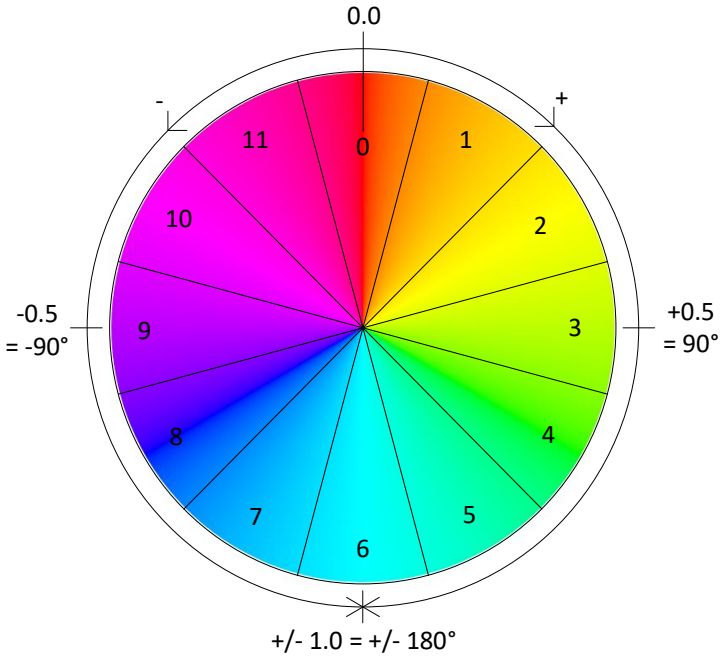

Figure 4: Hue Example for Phase 0.

7.5.12.3 video mcc blink

Command	`video mcc blink <mask> (<period>)`		
Alias	`mcc_blink`	**Type**	Setter, Getter
Description	Toggle blinking of masked multi matrix phases. Can be used to highlight areas in the image affected by certain phases. The mask parameter is a bitmask where each bit represents one phase of the multi matrix. Bits of unused phases are ignored and have no effect. 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. Enabling blinking disables gating (see `mcc gate` command) and vice versa. Changing the number of MCC phases or disabling the MCC also disables blinking. This setting is not saved, that means after a reboot or power cycle blinking will always be disabled!		
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 24 phases: 0x00FFFFFF 32 phases: 0xFFFFFFFF	10000	
Default	0	1000	

7.5.12.4 video mcc gate

Command	`video mcc gate <mask>`		
Alias	`mcc_gate`	**Type**	Setter, Getter
Description	Toggle gating of masked multi matrix phases. Can be used to highlight areas in the image affected by certain phases. The mask parameter is a bitmask where each bit represents one phase of the multi matrix. Bits of unused phases are ignored and have no effect. For each selected phase the saturation is set to 0 which will grey out all image areas that lie within that phase. To isolate a single phase set the gate bits of all other phases. Enabling gating disables blinking (see `mcc blink` command) and vice versa. Changing the number of MCC phases or disabling the MCC also disables gating.		

	This setting is not saved, that means after a reboot or power cycle gating will always be disabled!
Parameter	`mask`
Type	Unsigned Integer
Description	Each bit selects one phase for blinking.
Min	0x00000000
Max	Depends on number of active phases: 12 phases: 0x00000FFF 16 phases: 0x0000FFFF 24 phases: 0x00FFFFFF 32 phases: 0xFFFFFFFF
Default	0

Examples:

To gate a single phase set only one bit of the mask. This sets the saturation of the selected phase to 0 (turns it grey in the output image):

```
→ 1 video mcc gate 0x00000001 # Gate phase 1
← OK
```

To highlight a phase and do isolated tuning, e.g. when using a vector scope, it is useful to gate all phases except the selected phase as it will then be the only phase that shows up with color in the vector scope:

```
→ 1 video mcc gate 0xFFFFFFFE # Gate all phases except phase 1.
← OK
```

7.5.13 video black_master

Command	`video black_master <red/all> (<green> <blue>)`		
Alias	`black_master`	**Type**	Setter, Getter
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		

7.5.13.1 video black_master red

Command	`video black_master red <offset>`		
Alias	`black_master_red`	**Type**	Setter, Getter
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		

7.5.13.2 video black_master green

Command	`video black_master green <offset>`		
Alias	`black_master_green`	**Type**	Setter, Getter
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

7.5.13.3 video black_master blue

Command	video black_master blue <offset>		
Alias	black_master_blue	Type	Setter, Getter
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

7.5.14 video black_gamma

Command	video black_gamma <range> <level>		
Alias	black_gamma	Type	Setter, Getter
Description	Set the black gamma value to fine-tune the brightness in dark image areas which changes the perceived image contrast. The range parameter adjusts how far the effect stretches into the bright image areas where 0 is bypass (no effect), 1 is narrow (only affects very dark areas), 100 is wide (also affects gray areas) and 50 is the default (middle) setting. The level parameter defines the strength of the effect where the default value of 0 is bypass, negative values decrease the brightness in dark areas, positive values increase it.		

Parameter	range	level
Type	Unsigned Integer	Signed Integer
Description	Width of the black gamma curve.	Strength of the black gamma effect.
Min	0 = Bypass	-100
Max	100	100
Default	50	0 = Bypass

7.5.15 video knee

Command	video knee <enable> (<point> <slope>)		
Alias	knee	Type	Setter, Getter
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 as percentage.	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:

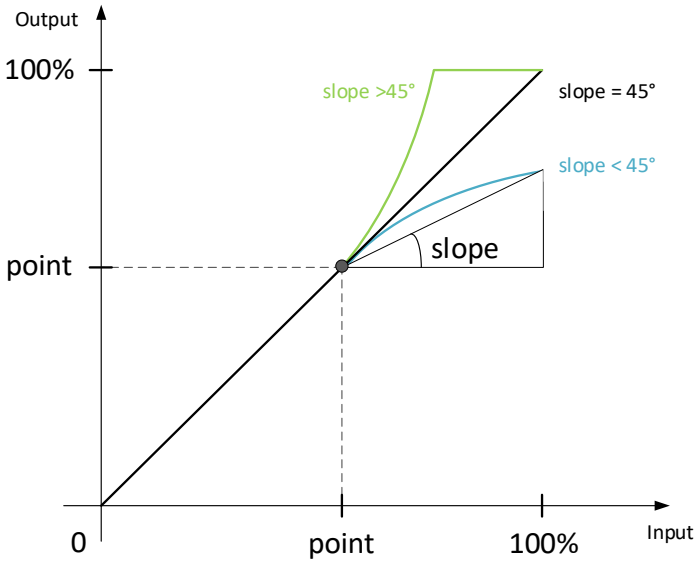

Figure 5: Knee Function Diagram.

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.

7.5.16 video lut

Command	`video lut (<index>) <enable>`		
Alias	`lut_enable`	**Type**	Setter, Getter
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 `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 with the ProVideo protocol.		
Parameter	`index`	`enable`	
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
```

7.5.16.1 video lut mode

Command	`video lut mode <mode>`		
Alias	`log_mode`	**Type**	Setter, Getter
Description	Set LUT mode. To list all modes, use the `lut mode list` subcommand. Note that in S-Log3 mode the SDI output range (see `video sdi_range` 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	`mode`		
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).

7.5.16.1.1 video lut mode list

Command	`video lut mode list`		
Alias	/	**Type**	List
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.

7.5.16.2 video lut gamma

Command	`video lut gamma <value>`		
Alias	`lut_fast_gamma`	**Type**	Setter, Getter
Description	Only applicable if `lut mode` 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 cure 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		

7.5.16.3 video lut max_brightness

Command	`video lut max_brightness <value>`		
Alias	`pq_max_brightness`	**Type**	Setter, Getter
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/m²
Min	100
Max	10000
Default	1000

7.5.16.4 video lut master_gain

Command	`video lut master_gain <value>`		
Alias	`slog3_master_gain`	**Type**	Setter, Getter
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		

7.5.16.5 video lut mode_compat

Command	`video lut mode_compat <mode>`		
Alias	`lut_mode`	**Type**	Setter, Getter
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		

7.5.17 video post

Command	`video post`		
Alias	`post`	**Type**	Special
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).

7.5.17.1 video post brightness

Command	`video post brightness <value>`		
Alias	`post_bright`	**Type**	Setter, Getter
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. This setting has no effect if the device is in HDR mode (`video lut mode` is set to > 0 = REC.709).		
Parameter	`value`		
Type	Signed 1.7 Fixed Point Number		
Description	Brightness factor.		
Min	-1.0 = -128		
Max	0.999 = 127		
Default	0.0 = 0		

7.5.17.2 video post contrast

Command	`video post contrast <value>`		
Alias	`post_cont`	**Type**	Setter, Getter
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. This setting has no effect if the device is in HDR mode (`video lut mode` is set to > 0 = REC.709).		
Parameter	`value`		
Type	Unsigned 1.7 Fixed Point Number		
Description	Contrast factor.		
Min	0.0 = 0		
Max	1.999 = 255		
Default	1.0 = 128		

7.5.17.3 video post saturation

Command	`video post saturation <value>`		
Alias	`post_sat`	**Type**	Setter, Getter
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		

7.5.17.4 video post hue

Command	`video post hue <value>`		
Alias	`post_hue`	**Type**	Setter, Getter
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°		

7.5.18 video osd

Note: The `osd` command is not available on all devices.

Command	`video osd <mode>`		
Alias	`osd`	**Type**	Setter, Getter
Description	Set on-screen display mode. To get a list of supported modes use the `osd list` subcommand.		
Parameter	`mode`		
Type	Unsigned Integer		
Description	OSD 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 OSD modes differ.

7.5.18.1 video osd list

Command	`video osd list`		
Alias	/	**Type**	List

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.

7.5.19 video user

Command	`video user`		
Alias	`user`	**Type**	Special
Description	Configure user settings which override or tune the camera calibration. See subcommands for details.		

This command has no functionality, it is only used to group the `user` subcommands (see below).

7.5.19.1 video user gain

Command	`video user gain <red> <green> <blue>`		
Alias	`user_gain`	**Type**	Setter, Getter
Description	Use the user gains to fine tune the RGB white balance gains. The user gains are multiplied with the gain values which result from the selected white balance temperature or preset. The resulting total gains can be read and set with the 'wb gain' command. Note that changing the total gains also updates the user gains. All values are in signed 5.8 fixed point format with a range of 0.0 = 0, 15.999 = 4095] (1.0 = 256). To use the original (calibrated) values, set all user gains to the default value of 1.0 = 256 (see usage example below). The user gains are preserved when changing the color temperature with the `wb temp` command but they are reset to 1.0 when a preset is selected using the `wb preset` command or if the auto white balance is run. The gains cannot be changed while auto white balance is enabled. To set a single gain value use the subcommands.		
Parameter	`red`	`green`	`blue`
Type	Unsigned 4.8 Fixed Point Number		
Description	Red gain.	Green gain.	Blue gain.
Min	0 = 0		
Max	15.999 = 4095		
Default	1.0 = 256		

The user gains are linked with the RGB white balance gains which can be set with the `wb gain` command that means if one of them gets changed the other one is updated automatically. For details see chapter 7.5.6.1.

To reset the user gains to neutral settings, they must be set to 1.0:

→ `1 video user gain 256 256 256`
← `OK`

7.5.19.1.1 video user gain red

Command	`video user gain red <gain>`		
Alias	`user_gain_red`	**Type**	Setter, Getter
Description	Set user defined red gain in unsigned 4.8 fixed point format (1.0 = 256).		
Parameter	`gain`		
Type	Unsigned 4.8 Fixed Point Number		
Description	Red user gain.		
Min	0 = 0		
Max	15.999 = 4095		
Default	0.0 = 0		

7.5.19.1.2 video user gain green

Command	`video user gain green <offset>`		
Alias	`user_gain_green`	**Type**	Setter, Getter

Description	Set user defined green gain in unsigned 4.8 fixed point format (1.0 = 256).
Parameter	`gain`
Type	Unsigned 4.8 Fixed Point Number
Description	Green user gain.
Min	0 = 0
Max	15.999 = 4095
Default	0.0 = 0

7.5.19.1.3 video user gain blue

Command	`video user gain blue <offset>`		
Alias	`user_gain_blue`	**Type**	Setter, Getter
Description	Set user defined blue gain in unsigned 4.8 fixed point format (1.0 = 256).		
Parameter	`gain`		
Type	Unsigned 4.8 Fixed Point Number		
Description	Blue user gain.		
Min	0 = 0		
Max	15.999 = 4095		
Default	0.0 = 0		

7.5.19.2 video user matrix

Command	`video user matrix <c0> ... <c8>`		
Alias	`user_matrix`	**Type**	Setter, Getter
Description	Set a user defined RGB color 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). To reset the matrix to neutral, program the identity matrix (see usage examples below).		
Parameter	`c0 … c8`		
Type	Signed 4.12 Fixed Point Number		
Description	User matrix coefficients.		
Min	-8.0 = -32768		
Max	7.999 = 32767		
Default	Identity matrix, see below.		

To reset the matrix to neutral, the identity matrix must be programmed (diagonal values set to 1.0):

→ `1 video user matrix 4096 0 0 0 4096 0 0 0 4096`
← `OK`

7.5.20 video test

Command	`video test`		
Alias	`test`	**Type**	Special
Description	Commands for testing, e.g. test pattern output.		

This command has no functionality, it is only used to group the `test` subcommands (see below).

7.5.20.1 video test pattern

Command	`video test pattern <mode>`		
Alias	`test_pattern`	**Type**	Setter, Getter
Description	Show a test pattern instead of live video. To list supported modes, use the 'pattern list' subcommand. This setting is not saved, that means after a reboot or power cycle the test pattern will always be disabled!		
Parameter	`mode`		
Type	Unsigned Integer		
Description	Test pattern mode to set.		
Valid Values	0 = Off / Live Video 1 = Black Image 2 = Color Bars		
Default	0 = Off / Live Video		

7.5.20.1.1 video test pattern list

Command	`video test pattern list`		
Alias	/	**Type**	List
Description	List all available test patterns.		

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

7.6 Lens Commands

These commands control the optional lens drive features of the camera. They are called with the `lens` command prefix. The following commands are only available on devices with an integrated motor zoom lens like PROTON ZOOM and ZOOM 4K.

Depending on the camera model the following motors can be available: **Iris, Focus, Zoom**.

Each of the available motors can be driven in two modes:

- **Absolute Mode:** The motor is driven to a new target position with a given speed.
- **Relative Mode:** The motor is driven forward or backward with a given speed until the end of the driving range is reached or speed is set to 0 again.

At any time, the current position of each motor can be queried. An additional flag indicates if the motor has reached the target position (within its mechanical accuracy).

7.6.1 lens iris

Command	`lens iris <f-number> (<speed> <at-target>)`		
Alias	`iris`	**Type**	Setter, Getter
Description	Set aperture. The value is scaled by 100 so f/2.4 is represented as '240'. To fully close the iris, e.g. for black level calibration, drive it to the maximum position. For convenience, instead of the f-number the `open` and `close` keywords can be used to fully open or close the iris. Optionally the speed as a percentage can be passed, if no speed is set it stays at the last set value. Setting a speed of 0 immediately stops the motor ignoring the given f-number value. The `<at-target>` flag is set to 1 when the motor has reached its target position and is stopped. Setting the flag has no effect.		
Parameter	`f-number`	`speed`	`at-target`
Type	Unsigned Integer	Unsigned Integer	Boolean
Description	F-number scaled by 100 (e.g. f/2.4 = 240)	Motor speed as percentage	If motor has reached its target, ignored on set
Valid Values	`open`: Fully open iris `close`: Fully close iris For value range see `range` subcommand	**0** to **100 %** where: **0:** Stop Motor **1:** Minimum Speed **100:** Maximum Speed	**0:** Motor is still moving **1:** Motor has reached its target and is stopped
Default	Fully open	100 (Maximum Speed)	/

Examples:

```
→ 1 lens iris 240 50    # Drive to position f/2.4 with a speed of 50 %
← OK
→ 1 lens iris 240 0     # Stop motor (f-number value is ignored)
← OK
→ 1 lens iris           # Get current position
← lens iris 203 5 1     # Motor has reached target at position f/2.03 (because
← OK                    # it was stopped before reaching f/2.4)
→ 1 lens iris open      # Fully open iris
← OK
→ 1 lens iris close     # Fully close iris
← OK
```


7.6.1.1 lens iris range

Command	`lens iris range`		
Alias	`iris_range`	**Type**	Pure Getter
Description	Get minimum and maximum f-number supported by the iris motor. Depending on the current zoom position, not all apertures are reachable. The motor will be driven as close to the target as possible and will adjust when the zoom position is changed. Output: `<min> <max>`		

Example:

```
→ 1 lens iris range
← lens iris range 138 12800  # Iris range is f/1.38 to f/128.00
← OK
```

7.6.1.2 lens iris relative

Command	`lens iris relative <speed>`		
Alias	`iris_relative`	**Type**	Setter, Getter
Description	Close iris (positive speed) or open iris (negative speed) until it reaches the max or min position. Use `speed` = 0 to stop the motor at the current position.		
Parameter	`speed`		
Type	Signed Integer		
Description	Motor speed		
Allowed Values	**0:** Stop motor **-100** to **100:** Drive motor backward or forward with given speed		

Example:

```
→ 1 lens iris relative 10     # Slowly close iris (10 % speed)
← OK
→ 1 lens iris relative 0      # Stop motor
← OK
→ 1 lens iris relative -90    # Open iris fast (90 % speed)
← OK
```

7.6.2 lens focus

Command	`lens focus <distance> (<speed> <at-target>)`		
Alias	`focus`	**Type**	Setter, Getter
Description	Set focus to given object distance in cm. For convenience, instead of the distance the `near` and `far` keywords can be used to set the focus to minimum object distance or infinity. Optionally the speed as a percentage can be passed, if no speed is set it stays at the last set value. Setting a speed of 0 immediately stops the motor ignoring the given object distance value. The `<at-target>` flag is set to 1 when the motor has reached its target position and is stopped. Setting the flag has no effect.		
Parameter	`distance`	`speed`	`at-target`
Type	Unsigned Integer	Unsigned Integer	Boolean
Description	Object distance in cm	Motor speed as percentage	If motor has reached its target, ignored on set
Valid Values	`near`: Min object distance `far`: Infinity For value range see `range` subcommand	**0** to **100 %** where: **0:** Stop Motor **1:** Minimum Speed **100:** Maximum Speed	**0:** Motor is still moving **1:** Motor has reached its target and is stopped
Default	150 cm (if supported by camera)	100 (Maximum Speed)	/

Examples:

```
→ 1 lens focus 2000 30  # Set focus to 2000cm = 20m with a speed of 30 %
← OK
→ 1 lens focus 2000 0   # Stop motor (distance value is ignored)
← OK
→ 1 lens focus          # Get current position
← lens focus 1270 3 1   # Motor has reached target at distance 12.7m (because it
← OK                    # was stopped before reaching 20m)
→ 1 lens focus near     # Set focus to minimum object distance
← OK
→ 1 lens focus far      # Set focus to infinity
← OK
```

7.6.2.1 lens focus range

Command	`lens focus range`		
Alias	`focus_range`	**Type**	Pure Getter
Description	Get minimum and maximum object distance in cm supported by the focus motor. The maximum distance equals “infinity”. Output: `<min> <max>`		

Example:

```
→ 1 lens focus range
← lens focus range 20 5000    # Focus range is 20cm to 50m (= infinity)
← OK
```

7.6.2.2 lens focus relative

Command	`lens focus relative <speed>`		
Alias	`focus_relative`	**Type**	Setter, Getter
Description	Drive focus far (positive speed) or near (negative speed) until it reaches the max or min position. Use `speed` = 0 to stop the motor at the current position.		
Parameter	`speed`		
Type	Signed Integer		
Description	Motor speed		
Allowed Values	**0:** Stop motor **-100** to **100:** Drive motor backward or forward with given speed		

Example:

```
→ 1 lens focus relative 10    # Slowly drive focus to “far” (10 % speed)
← OK
→ 1 lens focus relative 0     # Stop motor
← OK
→ 1 lens focus relative -90   # Quickly drive focus to “near” (90 % speed)
← OK
```

7.6.3 lens zoom

Command	`lens zoom <factor> (<speed> <at-target>)`		
Alias	`zoom`	**Type**	Setter, Getter
Description	Set the zoom factor. The value is scaled by 100 so 1.8x is represented as '180'. For convenience, instead of the factor the `wide` and `tele` keywords can be used to fully zoom out or in. Optionally the speed as a percentage can be passed, if no speed is set it stays at the last set value. Setting a speed of 0 immediately stops the motor ignoring the given zoom factor value. The `<at-target>` flag is set to 1 when the motor has reached its target position and is stopped. Setting the flag has no effect.		
Parameter	`factor`	`speed`	`at-target`
Type	Unsigned Integer	Unsigned Integer	Boolean

Description	Zoom factor scaled by 100 (e.g. 1.8x = 180)	Motor speed as percentage	If motor has reached its target, ignored on set
Valid Values	`wide`: Minimum zoom `tele`: Maximum zoom For value range see `range` subcommand	**0** to **100 %** where: **0:** Stop Motor **1:** Minimum Speed **100:** Maximum Speed	**0:** Motor is still moving **1:** Motor has reached its target and is stopped
Default	100 = 1.0x	100 (Maximum Speed)	/

Examples:

```
→ 1 lens zoom 150 80    # Set zoom to 1.5x with a speed of 80 %
← OK
→ 1 lens zoom           # Get current position
← lens zoom 150 8 1     # Motor has reached target position at 1.5x.
→ 1 lens zoom wide      # Set zoom to minimum (usually 1.0x)
← OK
→ 1 lens zoom tele      # Set zoom to maximum
← OK
```

7.6.3.1 lens zoom range

Command	`lens zoom range`		
Alias	`zoom_range`	**Type**	Pure Getter
Description	Get minimum and maximum zoom factor supported by the zoom motor. Output: `<min> <max>`		

Example:

```
→ 1 lens zoom range
← lens zoom range 100 207     # Focus range is 1.0x to 2.07x
← OK
```

7.6.3.2 lens zoom relative

Command	`lens zoom relative <speed>`		
Alias	`zoom_relative`	**Type**	Setter, Getter
Description	Zoom in (positive speed) or out (negative speed) until zoom reaches the max or min position. Use `speed` = 0 to stop the motor at the current position.		
Parameter	`speed`		
Type	Signed Integer		
Description	Motor speed		
Allowed Values	**0:** Stop motor **-100** to **100:** Drive motor backward or forward with given speed		

Example:

```
→ 1 lens zoom relative 10     # Slowly zoom in (10 % speed)
← OK
→ 1 lens zoom relative 0      # Stop motor
← OK
→ 1 lens zoom relative -90    # Quickly zoom out (90 % speed)
← OK
```

7.6.4 lens ircut

Command	`lens ircut <mode>`		
Alias	`ircut`	**Type**	Setter, Getter
Description	Set the IR cut filter mode. **Note**: The camera is calibrated with the IR cut filter in the optical path, driving the filter out breaks the color calibration.		
Parameter	`mode`		
Type	Unsigned Integer		

Description	Silent mode to set.
Valid Values	0 = Disabled 1 = Enabled
Default	1 = Enabled

7.6.5 lens silent

Command	`lens silent <enable>`		
Alias	`silent`	**Type**	Setter, Getter
Description	Enable or disable silent lens motor mode. In silent mode the lens motors get turned off when no motor is moving which eliminates noise emitted by the motor drive. When the camera is operated in a shaky environment with a lot of vibration, silent mode should be turned off otherwise the precision of the motor drive can be reduced.		
Parameter	`enable`		
Type	Boolean		
Description	Enable or disable silent mode.		
Default	On		

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 10: Command alias overview.

Alias	Full Command Name	ProVideo Command	Different Behavior when called via Alias
cam_gain	camera gain	Yes	No
cam_exposure	camera exposure	Yes	No
gain_mode	camera gain mode	No	No
cam_info	camera info	Yes	No
aec	camera auto	Yes	Yes
aec_mode	camera auto mode	No	No
aec_target	camera auto target	No	No
aec_speed	camera auto speed	No	No
aec_max_gain	camera auto max_gain	No	No
aec_anti_flicker	camera auto anti_flicker	No	No
aec_custom	camera auto custom	No	No
aec_weight	camera auto weight	Yes	No
iris	lens iris	No	No
iris_range	lens iris range	No	No
iris_relative	lens iris relative	No	No
focus	lens focus	No	No
focus_range	lens focus range	No	No
focus_relative	lens focus relative	No	No
zoom	lens zoom	No	No
zoom_range	lens zoom range	No	No
zoom_relative	lens zoom relative	No	No
ircut	lens ircut	No	No
silent	lens silent	No	No

Alias	Full Command Name	ProVideo Command	Different Behavior when called via Alias
save_settings	settings save	Yes	No
load_settings	settings load	Yes	No
reset_settings	settings reset	Yes	No
auto_save	settings auto_save	Yes	No
version	system info	Yes	Yes
name	system name	Yes	No
runtime	system runtime	Yes	No
reboot	system reboot	Yes	No
update	system update	No	No
identify	system identify	Yes	No
error	system error	No	No
volatile	system volatile	Yes	No
ping	system ping	No	No
rs485	system rs485	No	No
prompt	system rs485 mode	Yes	No
controller	system rs485 mode 0	No	No
interactive	system rs485 mode 1	No	No
rs485_addr	system rs485 device_address	Yes	No
rs485_bc_addr	system rs485 broadcast_address	Yes	No
rs485_bc_master	system rs485 broadcast_master	Yes	No
rs485_baud	system rs485 baudrate	Yes	No
rs485_tx_delay	system rs485 tx_delay	No	No
status_led	system status_led	No	No
tally	system tally	No	No
temp	system temp	Yes	No
max_temp	system temp max	Yes	No
max_temp_reset	system temp reset	Yes	No
over_temp_count	system temp count	Yes	No
humidity	system humidity	No	No
humidity_count	system humidity count	No	No
fan_target	system temp fan	Yes	No
sdi	system sdi	No	No
sdi_amplitude	system sdi amplitude	No	No
sdi_slew_rate	system sdi slew_rate	No	No
sdi_clock_tune	system sdi clock_tune	No	No
audio_enable	system audio	Yes	No
audio_gain	system audio gain	Yes	No
audio_inputs	system audio inputs	No	No
audio_volume	system audio volume	No	No
audio_bias	system audio bias	No	No
timecode_enable	system timecode	No	No
timecode	system timecode value	Yes	No
timecode_hold	system timecode pause	Yes	No

Alias	Full Command Name	ProVideo Command	Different Behavior when called via Alias
timecode_drop	system timecode drop	No	No
sync	system sync	No	No
video_mode	video mode	Yes	No
phases	video phases	No	No
phases_packing	video phases packing	No	No
phases_marker	video phases marker	No	No
flip	video flip	Yes	No
black_sensor	video black_sensor	No	No
black_red	video black_sensor red	Yes	No
black_green	video black_sensor green	Yes	No
black_blue	video black_sensor blue	Yes	No
lsc	video lsc	Yes	No
lsc_preset	video lsc preset	No	No
wb	video wb	Yes	No
wb_gain	video wb gain	No	No
gain_red	video wb gain red	Yes	No
gain_green	video wb gain green	Yes	No
gain_blue	video wb gain blue	Yes	No
wb_temp	video wb temp	No	No
wb_preset	video wb preset	Yes	No
awb	video wb auto	Yes	No
awb_speed	video wb auto speed	Yes	No
awb_offset	video wb auto offset	No	No
bpc	video bpc	No	No
dpc_auto_load	video bpc calibrate	Yes	No
flare	video flare	Yes	No
flare_red	video flare red	No	No
flare_green	video flare green	No	No
flare_blue	video flare blue	No	No
color_space	video color_space	Yes	No
sdi_range	video sdi_range	Yes	No
filter_enable	video filter	Yes	No
filter_detail	video filter detail	Yes	No
filter_denoise	video filter denoise	Yes	No
mcc	video mcc	Yes	No
mcc_opmode	video mcc mode	Yes	No
mcc_set	video mcc phase	Yes	No
mcc_blink	video mcc blink	Yes	No
mcc_gate	video mcc gate	No	No
black_master	video black_master	Yes	No
black_master_red	video black_master red	No	No
black_master_green	video black_master green	No	No
black_master_blue	video black_master blue	No	No

Alias	Full Command Name	ProVideo Command	Different Behavior when called via Alias
black_gamma	video black_gamma	No	No
knee	video knee	Yes	No
lut_enable	video lut	Yes	Yes
log_mode	video lut mode	Yes	No
lut_fast_gamma	video lut gamma	Yes	No
pq_max_brightness	video lut max_brightness	Yes	No
slog3_master_gain	video lut master_gain	Yes	No
lut_mode	video lut mode_compat	Yes	No
post	video post	No	No
post_bright	video post brightness	Yes	No
post_cont	video post contrast	Yes	No
post_sat	video post saturation	Yes	No
post_hue	video post hue	Yes	No
osd	video osd	Yes	No
user	video user	No	No
user_gain	video user gain	No	No
user_gain_red	video user gain red	No	No
user_gain_green	video user gain green	No	No
user_gain_blue	video user gain blue	No	No
user_matrix	video user matrix	No	No
test	video test	No	No
test pattern	video test pattern	No	No
iris	lens iris	No	No
iris_range	lens iris range	No	No
iris_relative	lens iris relative	No	No
focus	lens focus	No	No
focus_range	lens focus range	No	No
focus_relative	lens focus relative	No	No
zoom	lens zoom	No	No
zoom_range	lens zoom range	No	No
zoom_relative	lens zoom relative	No	No
ircut	lens ircut	No	No

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:

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

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

Revision	Date	Chapter	Changes
v1.0.0	29. May. 2024	All	Initial release.
v1.0.1	14. Jun. 2024	2.4	Reworked firmware update chapter for usage of the new PROTON Updater GUI which is now described in the new chapter 3.2.
		6.1	Updated list of critical settings.
		6.2, 7.2	Updated auto-save chapter and descriptions for the `settings reset` and `auto_save` commands: Auto-save is now done immediately after changing a setting.
		7.3	Updated command descriptions for the `system runtime`, `temp max`, `temp reset`, `temp count`, and `error` commands: Runtime and temperature logging are no longer persistent.
		7.3.5	Added `system update` command description.
		7.3.16.1	Updated default `audio gain` from 1.0 to 0.5.
		8	Updated alias reference.
v1.1.0	11. Jul. 2024	2.3	Updated maximum temperature, is now 100°C.
		3.2	Updated firmware update GUI chapter.
		7.4.4	Added chapter for new auto exposure control commands.
		7.5.6	Added auto white balance commands in section 7.5.6.4 and updated `wb`, `wb temp`, `wb gain`, `wb color_cross` and `wb preset` commands descriptions.
		7.3.10.5	Added new supported baud rates: 230400 and 250000 baud.
		7.3.16.1	Changed default audio gain from 0.5 to 1.0.
		8	Updated alias reference.
v1.1.1	22. Jul. 2024	2.2	Improve description of error state.
		2.4	Updated firmware update chapter.
		5.2.4	Added description of error code “-140”.
		7.3.5	Updated description of `system update` command.
		7.5.5	Added `lsc preset` command to LSC chapter.
		8	Updated alias reference.
v1.1.2	16. Aug. 2024	All	Fixed typos, no major changes.
v1.2.0	10. Sep. 2024	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.
		7.4.1.1, 7.4.3	The `low_light` command has been replaced with the `gain_mode` command. Changing the gain mode may also change the maximum gain value, so the `camera info` command description has been updated.
		7.4.4.1	Added missing description of combined auto exposure mode and improved description of other modes.
		7.5.4	Removed `black_sensor auto` command, it is no longer supported.
		7.5.6	Added new `wb gain offset` and `wb color_cross offset` commands and updated description of the other white balance commands accordingly. Note: The minimum color temperature of the `wb temp` command has been increased from 2400 to 3000 Kelvin and the 2800 Kelvin preset has been removed from the `wb preset` command.
		8	Updated alias list.
v1.3.0	25. Oct. 2024	2.1	Updated power/control connector table.
		4.1	Replaced ProVideo GUI with PROTON Control.

Revision	Date	Chapter	Changes
		5.1.1.2, 7.3.10.3	Broadcasting can now be disabled by using the broadcast address -1.
		5.2.3.5, 7.5.5.1.1	Added description of list commands. The elements of the list are now always prefixed with a hash (#) so that other cameras on the same bus do not interpret them as device IDs. Updated all `list` commands so that the examples now include the hash.
		7.3.16.1	Corrected default audio gain which is 1.0 and not 0.5.
		7.4.1, 7.4.1.1, 7.4.4.1	Improved `camera gain`, `camera exposure` and `camera auto mode` command descriptions.
		7.5.6	Updated `wb` command description, manual white balance now runs up to 10 frames. Removed `wb color_cross` command and moved `wb gain offset` and `wb color_cross offset` commands to new `video user` commands (with slightly different functionality). Updated related commands accordingly.
		7.5.12.3	Fix typo in command string of `mcc blink` command.
		7.5.19	Added new user command section which includes the `user gain` and `user matrix` commands.
		8	Updated alias list.
v1.3.1	04. Nov. 2024	2.3, 7.3.13.4	Added new `system temp fan` command to control fan turn-on temperature on supported devices.
		7.3.15.4	Added note that audio commands are only available on supported devices.
		7.5.6.2, 7.5.6.5	Updated formulas in `video wb gain` command description and added description of new `video wb tint` command.
		7.5.17.1, 7.5.17.2	Added note that `video post brightness` and `video post contrast` commands have no effect in HDR mode.
v1.3.2	03. Dec. 2024	1.1, 7.5.1, 7.3.1, 7.3.6	Added PROTON RAIN and FLEX as well as PROTON 4K to supported devices and added its UHD video modes to `video mode` command description. Updated `system info` and `identify` command description with notes on supported platforms.
		2.4, 7.3.14	Added chapter on humidity detection and description of the `system humidity` command group.
		2.5, 7.3.13.4, 7.3.15.4	Added chapter on audio and cooling fan operation, added cooling fan note to `system audio` command description and added presets to the `system temp fan` command.
		7.1.7	Added `firmware` command chapter. This is just for reference as the user usually does not call these commands directly.
		7.5.6	Updated `wb` command description, manual white balance now runs up to 1 second.
v1.4.0	18. Feb. 2025	1.1, 7.3.6	Added new supported device: PROTON ZOOM.
		5.2.3, 7	Introduced new “Direct” command type. Updated all command tables: Instead of having only the “Getter Function – Yes/No” information they now have a “Type” field which references the types described in chapter 5.2.3.
		5.2.4	Added new error code -14.
		7.4.1	The `gain_mode` command is now a subcommand of the `camera gain` command (new syntax: `camera gain mode` instead of `camera gain_mode`).
		7.4.4, 7.4.4.1 7.4.4.1.1	Added new Iris Control mode to auto exposure commands. Improved overall structure of the auto exposure description.

Revision	Date	Chapter	Changes
		7.4.4.8	Added new `camera auto brightness` command.
		7.5.5	Renamed “offset” to “radius” in `lsc` commands.
		7.5.7.1	Added note regarding BPC calibration on devices with a lens drive.
		7.5.9	Added Sony S-Gamut3 and S-Gamut3.Cine to supported color spaces.
		7.5.14	Added `video black_gamma` command chapter.
		7.5.12	Added new `video mcc gate` command, updated `mcc`, `mcc mode` and `mcc blink` command descriptions.
		7.5.18	Added note that `osd` command is not available on all devices and fixed copy paste errors.
		7.5.20	Added `video test` command chapter.
		7.6	Added `lens` command chapter which describes the usage of the optional lens drive (currently only used in PROTON ZOOM).
		8	Updated alias reference with new commands.
v1.4.1	31.Mar.2025	7.3.9	Added `system ping` command.
		7.6.1, 7.6.3	Fixed examples in `lens iris` and `lens zoom` command description.
		7.6.4, 7.6.5	Added `lens ircut` and `lens silent` command descriptions.
		8	Updated alias reference with new commands.
v1.4.2	22.Apr.2025	7.3.1	Clarified purpose and usage of the resolution mask in the `system info` command.
		7.5.1	Added missing ZOOM variant to supported video mode table.
		7.6.1, 7.6.2, 7.6.3	Added min and max keywords to set minimum and maximum speed for `lens iris`, `focus` and `zoom` commands.
		6.1, 6.2, 7.3.18	Added `system clock_tune` command description and added notes on calibration settings to the settings handling chapters.
V1.5.0	14.Jul.2025	1.1, 7.3.1 7.5.1, 7.5.2	Added PROTON FLEX 4K and PROTON HFR to supported devices. Updated `video mode` chapter and added `video phases` chapter with new features for PROTON HFR.
		1.1, 2.6, 2.7, 5.1 6.1, 7.2.2, 7.2.3, 7.3.1, 7.3.10 7.3.18, 7.5.1, 7.5.2, 8	Added and updated chapters for camera synchronization and notes on 3D camera rigs. Added PROTON 3D 4K to supported devices and updated all related chapters (mostly small notes regarding changes specific for 3D rigs). Added new `system sync` command to setup synchronization between the cameras of a 3D rig. Added missing and new commands to alias list.
		2.8	Added chapter on error recovery.
		5.2.4	Updated error code description.
		5.6, 7.3.10.6	Add chapters on the new configurable Tx delay.
		7.3.15.4	Added `audio inputs`, `volume` and `bias` commands for analog audio input control to `system audio` command chapter.
		7.3.15.4	Updated ranges and example of `clock_tune` command.

Revision	Date	Chapter	Changes
		7.6	Removed references to “max-speed” from the `lens` command chapter and updated examples as the speed range is now fixed to [0, 100] for all motors.
		7.6.1.1, 7.6.2.1, 7.6.3.1	Fixed examples for `lens iris range`, `focus range` and `zoom range` commands which were missing the command syntax in the reply.
1.5.1	18.Sep.2025	All	Fixed typos and errors and improved wording of some chapters.
		1.1, 7.3.1	Added new supported device: PROTON ZOOM 4K.
		7.3.15	Added new `system sdi` command group description. Moved `clock_tune` command to `sdi` command group.
		7.3.16.3, 7.3.16.4	Updated `system audio volume` and `audio bias` command description, they can now be called without arguments to list settings for all inputs.
		7.5.2.2	Added new “2SI” packing mode.
		8	Updated alias reference.