

Blackmagic Camera Control



Blackmagic URSA Cine
Blackmagic Cinema Camera 6K
Blackmagic URSA Broadcast G2

Blackmagic Micro Studio Camera 4K G2
Blackmagic Studio Cameras
Blackmagic Pocket Cinema Cameras

Contents

Developer Information	3	Immersive Control API	66
Camera Control REST API	3	Color Correction Control API	67
Livestream Control API	5	Notification websocket - 1.0.0	72
Clips Control API	9	messages	72
Media Pool Control API	10	Blackmagic SDI Camera Control Protocol	96
Monitoring Control API	13	Example Protocol Packets	105
Event Control API	22	Blackmagic Embedded Tally Control Protocol	106
System Control API	22	Visca Commands for PTZ control via SDI	108
Transport Control API	28	Blackmagic Bluetooth Camera Control	109
Timeline Control API	32	Service: Device Information Service	109
Media Control API	35	Service: Blackmagic Camera Service	109
Slate Control API	38	S.Bus	111
Preset Control API	44	PWM	113
Audio Control API	46	Wiring Diagrams	114
Lens Control API	51	Pinout Table for URSA Cine Connector	115
Video Control API	55		
Camera Control API	63		

Developer Information

Camera Control REST API

If you are a software or hardware developer you can build custom applications or leverage ready to use tools such as REST client or Postman to seamlessly control and interact with your compatible Blackmagic camera using Camera Control REST API. This API enables you to perform a wide range of operations, such as starting or stopping recordings, accessing disk information and much more. Whether you're developing a custom application tailored to your specific needs or utilizing existing tools, this API empowers you to unlock the full potential of your Blackmagic camera with ease. We look forward to seeing what you come up with!

NOTE It's important to mention that controlling Blackmagic cameras via REST API relies on the web manager being enabled on each compatible Blackmagic camera. Enable the web media manager in the Blackmagic Camera Setup 'network access' settings for each camera you are controlling.

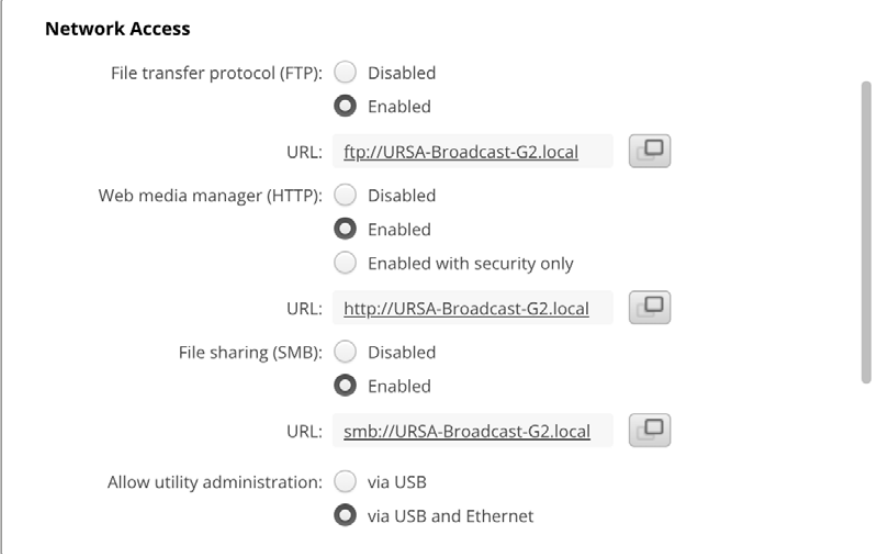
The following Blackmagic cameras are compatible with Camera Control REST API:

▪ Blackmagic URSA Cine 12K LF	▪ Blackmagic URSA Cine 17K 65
▪ Blackmagic URSA Cine Immersive	
▪ Blackmagic PYXIS 6K	
▪ Blackmagic Cinema Camera 6K	
▪ Blackmagic URSA Broadcast G2	
▪ Blackmagic Micro Studio Camera 4K G2	
▪ Blackmagic Studio Camera 4K Plus	▪ Blackmagic Studio Camera 4K Plus G2
▪ Blackmagic Studio Camera 4K Pro	▪ Blackmagic Studio Camera 4K Pro G2
▪ Blackmagic Studio Camera 6K Pro	

Sending API Commands

To send an API command to your camera from a third party application such as Postman, add /control/api/v1/ to the end of the camera's Web media manager URL or IP address. For example, <https://ursa-broadcast-g2.local/control/api/v1/>

You can find the Web media manager URL and IP address information in Blackmagic Camera Setup.



Network Access

File transfer protocol (FTP): ☐ Disabled ☒ Enabled

URL: 

Web media manager (HTTP): ☐ Disabled ☒ Enabled ☐ Enabled with security only

URL: 

File sharing (SMB): ☐ Disabled ☒ Enabled

URL: 

Allow utility administration: ☐ via USB ☒ via USB and Ethernet

The Web media manager URL in Blackmagic Camera Setup

Downloading API's from your Camera

You can download REST API YAML documentation from your camera by adding /control/documentation.html to the end of the camera's Web media manager URL or IP address. For example, <https://ursa-broadcast-g2.local/control/documentation.html>

NOTE It's worth noting that changing the camera name in Blackmagic Camera Setup will also change the camera's Web media manager URL.

Livestream Control API

API for controlling Livestreams on Blackmagic Design products.

GET /livestreams/0

Get the livestream's current status.

Response

200 - Livestream's current status.

The response is JSON.

Name	Type	Description
status (required)	string	Possible values are: Idle, Connecting, Streaming, Flushing, Interrupted.
bitrate (required)	integer	Current bitrate (bps).
effectiveVideoFormat (required)	string	Effective video format for the livestream, serialised as a string.
duration	integer	Current stream duration in seconds. Absent if livestream is idle.
cache	integer	Current stream cache usage percentage.

GET /livestreams/0/start

Determine if the livestream is active.

Response

200 - Livestream active status.

The response is JSON.

Name	Type	Description
	boolean	True when the livestream is active.

PUT /livestreams/0/start

Start the livestream.

Response

204 - Livestream started.

GET /livestreams/0/stop

Determine if the livestream is inactive.

Response

200 - Livestream inactive status.

The response is JSON.

Name	Type	Description
	boolean	True when the livestream is inactive.

PUT /livestreams/0/stop

Stop the livestream.

Response

204 - Livestream stopped.

GET /livestreams/0/activePlatform

Get the currently selected platform configuration for the livestream.

Response

200 - Livestream active platform configuration.

The response is JSON.

Livestream's current active platform configuration.

Name	Type	Description
platform (required)	string	Platform name.
server (required)	string	The platform's server name, or "Custom" when the URL is customizable.
key	string	Stream key. Assumed to be empty if missing.
passphrase	string	Passphrase. Only included for SRT streams.
quality (required)	string	Quality level name.
url	string	Livestream destination. Only included when URL is customizable.

PUT /livestreams/0/activePlatform

Set the currently selected platform configuration for the livestream.

Parameters

Livestream's current active platform configuration.

Name	Type	Description
platform (required)	string	Platform name.
server (required)	string	The platform's server name, or "Custom" when the URL is customizable.
key	string	Stream key. Assumed to be empty if missing.
passphrase	string	Passphrase. Only included for SRT streams.
quality (required)	string	Quality level name.
url	string	Livestream destination. Only included when URL is customizable.

Response

204 - Livestream active platform configuration updated.

400 - Bad Request

GET /livestreams/platforms

Get the list of available platforms.

Response

200 - List of available platforms.

The response is JSON.

Name	Type	Description
	array	List of available platforms names.
[i]	string	Platform name.

GET /livestreams/platforms/{platformName}

Get the service configuration for a platform.

Parameters

Name	Type	Description
{platformName} (required)	string	Name of the platform.

Response

200 - Service configuration for specified platform.

The response is JSON.

Livestream platform service configuration.

Name	Type	Description
platform (required)	string	Corresponding platform name.
key	string	Default stream key.
servers (required)	array	List of server configurations.
servers[i]	object	Server configuration.
servers[i].server (required)	string	Server name.
servers[i].url (required)	string	Livestream destination.
servers[i].srtExtensions	array	Miscellaneous tags used for SRT livestreams.
servers[i].srtExtensions[i]	object	Dictionary object mapping SRT tag strings to values.
servers[i].srtExtensions[i][key]	string	SRT tag value.
servers[i].group	string	Logical grouping of the server.
profiles (required)	array	List of profile configurations.
profiles[i]	object	Quality configuration.
profiles[i].profile (required)	string	Quality level name.
profiles[i].configs (required)	array	List of video format configurations.
profiles[i].configs[i]	object	Video format configuration for profiles.
profiles[i].configs[i].resolution (required)	string	Video format serialised as a string.
profiles[i].configs[i].fps (required)	string	Frames per second.
profiles[i].configs[i].bitrate (required)	integer	Pixel bitrate (bps).
profiles[i].configs[i].audioBitrate	integer	Audio bitrate (bps).

Name	Type	Description
profiles[i].configs[i].keyFrameInterval	integer	How often a key frame is sent, in seconds.
profiles[i].configs[i].videoCodecs	array	Supported video encoding algorithm/s.
profiles[i].configs[i].videoCodecs[i]	string	Video encoding algorithm. Possible values are: H264, H265.
profiles[i].lowLatency (required)	boolean	If true, fewer frames will be buffered in the livestream.
defaultProfile	string	Quality level name.
credentials	object	Credentials used for RTMP streams.
credentials.username (required)	string	The username part of the credentials. Only used for RTMP streams.
credentials.password (required)	string	Used for RTMP streams, also used as Passphrase for SRT streams.
customizableUrlEnabled	boolean	True when the server URL is customizable.

400 - Bad Request

GET /livestreams/customPlatforms

Get a list of custom platform files.

Response

200 - List of custom platform files.

The response is JSON.

Name	Type	Description
	array	List of custom platform file names.
[i]	string	Custom platform file name.

DELETE /livestreams/customPlatforms

Remove all custom configuration files.

Response

204 - All custom configuration files removed.

GET /livestreams/customPlatforms/{filename}

Get a custom platform file.

Parameters

Name	Type	Description
{filename} (required)	string	Name of the file to get.

Response

200 - Custom platform file.

The response is XML.

Blackmagic streaming XML file format.

Name	Type	Description
	object	Blackmagic streaming XML file format.

404 - Not Found

PUT /livestreams/customPlatforms/{filename}

Update a custom platform file if it exists, if not, create a new file with the given file name.

Parameters

Name	Type	Description
{filename} (required)	string	Name of the file to update/create.

Blackmagic streaming XML file format.

Name	Type	Description
	object	Blackmagic streaming XML file format.

Response

204 - Custom platform file created or updated.

400 - Bad Request

DELETE /livestreams/customPlatforms/{filename}

Remove the given custom platform file.

Parameters

Name	Type	Description
{filename} (required)	string	Name of the file to be removed.

Response

204 - Custom platform file removed.

404 - Not Found

Clips Control API

API for listing clips on disk.

GET /clips

Get the list of clips on the active disk.

Response

200 - List of clips on the active disk.

The response is JSON.

List of media clips.

Name	Type	Description
clips (required)	array	
clips[i]	object	Media clip.
clips[i].clipUniqueId (required)	integer	Unique ID used to identify this clip.
clips[i].filePath	string	Path to the file relative to the root of a mount.
clips[i].fileSize	integer	Size of file on disk in bytes.
clips[i].codecFormat	object	
clips[i].codecFormat.codec	string	Currently selected codec.
clips[i].codecFormat.container	string	Multimedia container format.

Name	Type	Description
clips[i].videoFormat	object	Video format configuration.
clips[i].videoFormat.name (required)	string	Video format serialised as a string.
clips[i].videoFormat.frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
clips[i].videoFormat.height	number	Height dimension of video format.
clips[i].videoFormat.width	number	Width dimension of video format.
clips[i].videoFormat.interlaced	boolean	Is the display format interlaced?
clips[i].startTimecode	string	Start timecode of the clip serialised as string.
clips[i].durationTimecode	string	Duration of the clip in timecode format serialised as string.
clips[i].frameCount	integer	Number of frames in clip; duration of the clip in frames.

404 - There is no active disk.

Media Pool Control API

API to manage media pool and handle uploads and project data.

GET /cloud/projects

List all projects within the media pool.

Response

200 - Successfully retrieved the list of all projects.

The response is JSON.

Name	Type	Description
	array	
[i]	object	
[i].libraryID	string	
[i].name	string	
[i].private	boolean	
[i].shared	boolean	
[i].clips	array	List of clips associated with the project.
[i].clips[i]	string	
[i].status	object	
[i].status.numClipsRequested	integer	
[i].status.numClipsComplete	integer	
[i].status.uploadPercent	integer	Percentage of upload completion.
[i].status.numClipsPaused	integer	
[i].status.outOfSpace	boolean	
[i].status.secsRemaining	integer	Estimated seconds remaining until upload is completed.
[i].status.currentByteRate	integer	Current byte rate of the upload process.

GET /cloud/projects/active

Retrieve data of the actively uploading project.

Response

200 - Successfully retrieved the active project's data.

The response is JSON.

Name	Type	Description
libraryID	string	
name	string	
private	boolean	
shared	boolean	
clips	array	List of clips associated with the project.
clips[i]	string	
status	object	
status.numClipsRequested	integer	
status.numClipsComplete	integer	
status.uploadPercent	integer	Percentage of upload completion.
status.numClipsPaused	integer	
status.outOfSpace	boolean	
status.secsRemaining	integer	Estimated seconds remaining until upload is completed.
status.currentByteRate	integer	Current byte rate of the upload process.

GET /cloud/projects/{projectID}

Retrieve specific project data by project ID.

Parameters

Name	Type	Description
{projectID} (required)	integer	Unique identifier of the project.

Response

200 - Successfully retrieved the project data.

The response is JSON.

Name	Type	Description
libraryID	string	
name	string	
private	boolean	
shared	boolean	
clips	array	List of clips associated with the project.
clips[i]	string	
status	object	
status.numClipsRequested	integer	
status.numClipsComplete	integer	
status.uploadPercent	integer	Percentage of upload completion.

Name	Type	Description
status.numClipsPaused	integer	
status.outOfSpace	boolean	
status.secsRemaining	integer	Estimated seconds remaining until upload is completed.
status.currentByteRate	integer	Current byte rate of the upload process.

404 - Project not found.

GET /cloud/clips

List all clips within the media pool.

Response

200 - Successfully retrieved the list of all clips.

The response is JSON.

Name	Type	Description
	array	
[i]	string	

GET /cloud/clips/activeUploading

Retrieve data of actively uploading clips.

Response

200 - Successfully retrieved the list of actively uploading clips.

The response is JSON.

Name	Type	Description
	array	
[i]	object	
[i].path	string	
[i].projectId	integer	
[i].status	object	
[i].status.projectID	integer	
[i].status.outOfSpace	boolean	
[i].status.proxyExtension	string	
[i].status.growingFile	boolean	
[i].status.originalUploadState	string	Possible values are: Unqueued, Paused, Queued, Uploading, Uploaded, Failed.
[i].status.proxyUploadState	string	Possible values are: Unqueued, Paused, Queued, Uploading, Uploaded, Failed.
[i].status.originalClipTotalSize	integer	
[i].status.proxyClipTotalSize	integer	
[i].status.originalClipCompletedSize	integer	
[i].status.proxyClipCompletedSize	integer	
[i].status.secsRemaining	integer	

GET /cloud/clips/{deviceName}/{path}

Retrieve specific clip data by device and path.

Parameters

Name	Type	Description
{deviceName} (required)	string	Name of the device where the clip is stored.
{path} (required)	string	Path to the clip.

Response

200 - Successfully retrieved the clip data.

The response is JSON.

Name	Type	Description
path	string	
projectId	integer	
status	object	
status.projectID	integer	
status.outOfSpace	boolean	
status.proxyExtension	string	
status.growingFile	boolean	
status.originalUploadState	string	Possible values are: Unqueued, Paused, Queued, Uploading, Uploaded, Failed.
status.proxyUploadState	string	Possible values are: Unqueued, Paused, Queued, Uploading, Uploaded, Failed.
status.originalClipTotalSize	integer	
status.proxyClipTotalSize	integer	
status.originalClipCompletedSize	integer	
status.proxyClipCompletedSize	integer	
status.secsRemaining	integer	

404 - Clip not found.

Monitoring Control API

API for monitoring and controlling display settings in video equipment.

GET /monitoring/display

Retrieve a list of all display names.

Response

200 - Returns a list of display names.

The response is JSON.

Name	Type	Description
displays	array	List of display names available.
displays[i]	string	

GET /monitoring/{displayName}/cleanFeed

Get the clean feed enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Response

200 - OK

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

404 - Display name not found.

PUT /monitoring/{displayName}/cleanFeed

Set the clean feed enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

Response

204 - Clean feed enabled/disabled successfully.

400 - Invalid input.

422 - Unable to process the contained instructions.

GET /monitoring/{displayName}/displayLUT

Get the display LUT enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Response

200 - OK

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

400 - Invalid display name.

404 - Display name not found.

PUT /monitoring/{displayName}/displayLUT

Set the display LUT enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

Response

204 - Display LUT enabled/disabled successfully.

400 - Invalid input.

422 - Unprocessable Entity - Unable to process the contained instructions.

GET /monitoring/{displayName}/zebra

Get the zebra enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Response

200 - OK

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

404 - Display name not found.

PUT /monitoring/{displayName}/zebra

Set the zebra enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

Response

204 - Zebra enabled/disabled successfully.

400 - Invalid input.

422 - Unable to process the contained instructions.

GET /monitoring/{displayName}/focusAssist

Get the focus assist enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Response

200 - OK

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

404 - Display name not found.

PUT /monitoring/{displayName}/focusAssist

Set the focus assist enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Name	Type	Description
mode	string	Mode of focus assist, e.g., 'Peak' or 'ColoredLines'. Possible values are: Peak, ColoredLines.
color	string	Color of the focus assist highlight. Possible values are: Red, Green, Blue, White, Black.
intensity	integer	Intensity of the focus assist highlight (0-100).

Response

204 - Focus assist settings updated successfully.

400 - Invalid input or configuration.

422 - Unable to process the contained instructions.

GET /monitoring/focusAssist

Get the focus assist settings.

Response

200 - OK

The response is JSON.

Name	Type	Description
mode	string	Mode of focus assist, e.g., 'Peak' or 'ColoredLines'. Possible values are: Peak, ColoredLines.
color	string	Color of the focus assist highlight. Possible values are: Red, Green, Blue, White, Black.
intensity	integer	Intensity of the focus assist highlight (0-100).

404 - Display name not found.

PUT /monitoring/focusAssist

Set the focus assist settings.

Parameters

Name	Type	Description
mode	string	Mode of focus assist, e.g., 'Peak' or 'ColoredLines'. Possible values are: Peak, ColoredLines.
color	string	Color of the focus assist highlight. Possible values are: Red, Green, Blue, White, Black.
intensity	integer	Intensity of the focus assist highlight (0-100).

Response

204 - Focus assist settings updated successfully.

400 - Invalid input or configuration.

422 - Unable to process the contained instructions.

GET /monitoring/{displayName}/frameGuide

Get the frame guide enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Response

200 - Returns the frame guide enable state.

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

404 - Display not found.

PUT /monitoring/{displayName}/frameGuide

Set the frame guide enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

Response

204 - Frame guide state updated successfully.

422 - Unable to update the frame guide state.

GET /monitoring/frameGuideRatio

Get the current frame guide ratio.

Response

200 - Returns the current frame guide ratio.

The response is JSON.

Name	Type	Description
ratio	string	The frame guide ratio.

PUT /monitoring/frameGuideRatio

Set the frame guide ratio.

Parameters

Name	Type	Description
ratio	string	The frame guide ratio.

Response

204 - Frame guide ratio updated successfully.

422 - Unable to update the frame guide ratio.

GET /monitoring/frameGuideRatio/presets

Get the presets for frame guide ratios.

Response

200 - Returns a list of preset frame guide ratios.

The response is JSON.

Name	Type	Description
presets	array	
presets[i]	string	A frame guide ratio.

GET /monitoring/{displayName}/frameGrids

Get the frame grids enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Response

200 - Returns the frame grids enable state.

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

404 - Display not found.

PUT /monitoring/{displayName}/frameGrids

Set the frame grids enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

Response

204 - Frame grids state updated successfully.

422 - Unable to update the frame grids state.

GET /monitoring/frameGrids

Get the global frame grids settings.

Response

200 - Returns the current frame grids settings.

The response is JSON.

Name	Type	Description
frameGrids	array	List of frame grids enabled.
frameGrids[i]	string	Possible values are: Thirds, Crosshair, Dot, Horizon.

PUT /monitoring/frameGrids

Set the global frame grids settings.

Parameters

Name	Type	Description
frameGrids	array	List of frame grids enabled.
frameGrids[i]	string	Possible values are: Thirds, Crosshair, Dot, Horizon.

Response

204 - Frame grids settings updated successfully.

400 - Invalid input, check the number of frame grids or values.

422 - Unable to update the frame grids settings.

GET /monitoring/{displayName}/safeArea

Get the safe area enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Response

200 - Returns the safe area enable state.

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

404 - Display not found.

PUT /monitoring/{displayName}/safeArea

Set the safe area enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

Response

204 - Safe area state updated successfully.

422 - Unable to update the safe area state.

GET /monitoring/safeAreaPercent

Get the current safe area percentage.

Response

200 - Returns the current safe area percentage.

The response is JSON.

Name	Type	Description
percent	integer	Safe area coverage percentage.

PUT /monitoring/safeAreaPercent

Set the safe area percentage.

Parameters

Name	Type	Description
percent	integer	Safe area coverage percentage to set.

Response

204 - Safe area percentage updated successfully.

400 - Invalid percentage value.

422 - Unable to update the safe area percentage.

GET /monitoring/{displayName}/falseColor

Get the false color enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNamees.

Response

200 - Returns the false color enable state.

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

404 - Display not found.

PUT /monitoring/{displayName}/falseColor

Set the false color enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNamees.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

Response

204 - False color state updated successfully.

422 - Unable to update the false color state.

Event Control API

API For working with built-in websocket.

GET /event/list

Get the list of events that can be subscribed to using the websocket API.

Response

200 - Websocket API events list.

The response is JSON.

Name	Type	Description
events	array	List of events that can be subscribed to using the websocket API.
events[i]	string	

System Control API

API for controlling the System Modes on Blackmagic Design products.

GET /system

Get device system information.

Response

200 - System summary.

The response is JSON.

The properties will be populated only with the values that are supported/implemented by the device in use.

Name	Type	Description
codecFormat	object	Codec format configuration.
codecFormat.codec	string	Codec serialised as string.
codecFormat.container	string	Multimedia container format.
videoFormat	object	Video format configuration.
videoFormat.name (required)	string	Video format serialised as a string.
videoFormat.frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
videoFormat.height	number	Height dimension of video format.
videoFormat.width	number	Width dimension of video format.
videoFormat.interlaced	boolean	Is the display format interlaced?

501 - This functionality is not implemented for the device in use.

GET /system/product

Get device product information.

Response

200 - Device product information.

The response is JSON.

Product information.

Name	Type	Description
deviceName	string	Name of device as displayed in Setup.
productName	string	Device's product name.
softwareVersion	string	Software version running on device.

501 - This functionality is not implemented for the device in use.

GET /system/supportedCodecFormats

Get the list of supported codecs.

Response

200 - List of supported codec formats.

The response is JSON.

Name	Type	Description
codecs	array	
codecs[i]	object	Codec format configuration.
codecs[i].codec	string	Codec serialised as string.
codecs[i].container	string	Multimedia container format.

501 - This functionality is not implemented for the device in use.

GET /system/codecFormat

Get the currently selected codec.

Response

200 - Current codec format.

The response is JSON.

Codec format configuration.

Name	Type	Description
codec	string	Codec serialised as string.
container	string	Multimedia container format.

501 - This functionality is not implemented for the device in use.

PUT /system/codecFormat

Update the system codec.

Parameters

Codec format configuration.

Name	Type	Description
codec	string	Codec serialised as string.
container	string	Multimedia container format.

Response

204 - The codec updated successfully.

400 - The specified codec format is unsupported.

501 - This functionality is not implemented for the device in use.

GET /system/videoFormat

Get the currently selected video format.

Response

200 - Current system video format.

The response is JSON.

Video format configuration.

Name	Type	Description
name (required)	string	Video format serialised as a string.
frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
height	number	Height dimension of video format.
width	number	Width dimension of video format.
interlaced	boolean	Is the display format interlaced?

501 - This functionality is not implemented for the device in use.

PUT /system/videoFormat

Set the system video format.

Parameters

This parameter can be one of the following types:

Name	Type	Description
name (required)	string	Video format serialised as a string.

Name	Type	Description
frameRate (required)	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
height (required)	number	Height dimension of video format.
width (required)	number	Width dimension of video format.
interlaced	boolean	Is the display format interlaced?

Response

204 - The video format updated successfully.

400 - Invalid request.

409 - Operation unsupported in the current state.

501 - This functionality is not implemented for the device in use.

GET /system/supportedVideoFormats

Get the list of supported video formats for the current system state.

Response

200 - List of supported video formats.

The response is JSON.

List of supported video formats.

Name	Type	Description
formats	array	List of video formats.
formats[i]	object	Video format configuration.
formats[i].name (required)	string	Video format serialised as a string.
formats[i].frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
formats[i].height	number	Height dimension of video format.
formats[i].width	number	Width dimension of video format.
formats[i].interlaced	boolean	Is the display format interlaced?

501 - This functionality is not implemented for the device in use.

GET /system/supportedFormats

Get supported formats.

Response

200 - List of supported formats.

The response is JSON.

Name	Type	Description
supportedFormats	array	
supportedFormats[i]	object	
supportedFormats[i].codecs	array	
supportedFormats[i].codecs[i]	string	
supportedFormats[i].frameRates	array	
supportedFormats[i].frameRates[i]	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
supportedFormats[i].maxOffSpeedFrameRate	number	
supportedFormats[i].minOffSpeedFrameRate	number	
supportedFormats[i].recordResolution	object	
supportedFormats[i].recordResolution.height	number	Height of the resolution.
supportedFormats[i].recordResolution.width	number	Width of the resolution.
supportedFormats[i].sensorResolution	object	
supportedFormats[i].sensorResolution.height	number	Height of the resolution.
supportedFormats[i].sensorResolution.width	number	Width of the resolution.

501 - This functionality is not implemented for the device in use.

GET /system/format

Get current format.

Response

200 - Current format.

The response is JSON.

Name	Type	Description
codec	string	Codec format serialised as a string.
frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
maxOffSpeedFrameRate	number	
minOffSpeedFrameRate	number	
offSpeedEnabled	boolean	
offSpeedFrameRate	number	
recordResolution	object	
recordResolution.height	number	Height of the resolution.
recordResolution.width	number	Width of the resolution.
sensorResolution	object	
sensorResolution.height	number	Height of the resolution.
sensorResolution.width	number	Width of the resolution.

501 - This functionality is not implemented for the device in use.

PUT /system/format

Set the format.

Parameters

Name	Type	Description
codec	string	Codec format serialised as a string.
frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
maxOffSpeedFrameRate	number	
minOffSpeedFrameRate	number	
offSpeedEnabled	boolean	
offSpeedFrameRate	number	
recordResolution	object	
recordResolution.height	number	Height of the resolution.
recordResolution.width	number	Width of the resolution.
sensorResolution	object	
sensorResolution.height	number	Height of the resolution.
sensorResolution.width	number	Width of the resolution.

Response

204 - System format updated.

501 - This functionality is not implemented for the device in use.

Transport Control API

API for controlling Transport on Blackmagic Design products.

GET /transports/0

Get device's basic transport status.

Response

200 - Transport status.

The response is JSON.

Name	Type	Description
mode	string	Transport mode. Possible values are: InputPreview, InputRecord, Output.

PUT /transports/0

Set device's basic transport status.

Parameters

Name	Type	Description
mode	string	Transport mode. Possible values are: InputPreview, Output.

Response

204 - Transport mode was set.

400 - Failed to set transport mode.

GET /transports/0/stop

Determine if transport is stopped.

Response

200 - Transport stop response.

The response is JSON.

Name	Type	Description
	boolean	True when transport mode is InputPreview or when in Output mode and speed is 0.

PUT /transports/0/stop

Stop transport. Deprecated, use POST /transports/0/stop instead.

Response

204 - Transport stopped.

POST /transports/0/stop

Stop transport.

Response

204 - Transport stopped.

GET /transports/0/play

Determine if transport is playing.

Response

200 - Transport play response.

The response is JSON.

Name	Type	Description
	boolean	True when transport is in Output mode and speed is non-zero.

PUT /transports/0/play

Start playing on transport. Deprecated, use POST /transports/0/play instead.

Response

204 - Transport playing.

400 - Failed to set transport to play.

POST /transports/0/play

Start playing on transport.

Response

204 - Transport playing.

400 - Failed to set transport to play.

GET /transports/0/playback

Get playback state.

Response

200 - Transport playback state.

The response is JSON.

Name	Type	Description
type	string	Possible values are: Play, Jog, Shuttle, Var.
loop	boolean	When true, playback loops from the end of the timeline to the beginning of the timeline.
singleClip	boolean	When true, playback loops from the end of the current clip to the beginning of the current clip.
speed	number	Playback speed, 1.0 for normal forward playback.
position	integer	Playback position on the timeline in units of video frames, where 0 is the first frame of the timeline.

PUT /transports/0/playback

Set playback state.

Parameters

Name	Type	Description
type	string	Possible values are: Play, Jog, Shuttle, Var.
loop	boolean	When true, playback loops from the end of the timeline to the beginning of the timeline.
singleClip	boolean	When true, playback loops from the end of the current clip to the beginning of the current clip.
speed	number	Playback speed, 1.0 for normal forward playback.
position	integer	Playback position on the timeline in units of video frames, where 0 is the first frame of the timeline.

Response

204 - Updated transport playback state.

400 - Failed to set transport playback state.

GET /transports/0/record

Get record state.

Response

200 - Recording state.

The response is JSON.

Name	Type	Description
recording	boolean	If true, transport is in InputRecord mode.

PUT /transports/0/record

Set record state. Deprecated, use POST /transports/0/record instead.

Parameters

Name	Type	Description
recording (required)	boolean	If true, starts a recording, otherwise stops.
clipName	string	Optional, sets the requested clip name to record to, when "recording" attribute is set to true.

Response

204 - Recording state updated.

400 - Failed to update recording state.

POST /transports/0/record

Start recording.

Parameters

Name	Type	Description
clipName	string	Optional, provides a specific name of clip to record to.

Response

204 - Recording started.

400 - Failed to start recording.

GET /transports/0/clipIndex

Get the clip index of the currently playing clip on the timeline.

Response

200 - Clip index response.

The response is JSON.

Name	Type	Description
clipIndex	number null	The 0-based index of the clip being played on the timeline. null when there is no timeline or an empty timeline.

GET /transports/0/timecode

Get device timecode.

Response

200 - Timecode response.

The response is JSON.

Name	Type	Description
display	string	The display timecode serialised as a string.
timeline	string	The timeline timecode serialised as a string.

GET /transports/0/timecode/source

Get timecode source selected on device.

Response

200 - Timecode source response.

The response is JSON.

Name	Type	Description
timecode	string	Possible values are: Timeline, Clip.

Timeline Control API

API for controlling playback timeline.

GET /timelines/0

Get the playback timeline.

Response

200 - Playback timeline.

The response is JSON.

Name	Type	Description
clips	array	
clips[i]	object	Timeline clip.
clips[i].clipUniqueId (required)	integer	Unique identifier used to identify this media clip. If the same media clip is added to the timeline multiple times, each timeline clip has the same clipUniqueId
clips[i].frameCount	integer	Duration of timeline clip in frames, the number of frames in this clip on the timeline.
clips[i].durationTimecode	string	Duration of the timeline clip in timecode format serialised as string. This will differ to durationTimecode reported in /clips for this clipUniqueId if clipIn or frameCount was specified when adding this clip to the timeline.
clips[i].clipIn	string	In frame offset for the clip on the timeline, where 0 is the first frame of the on-disk clip.
clips[i].inTimecode	string	Clip timecode of the first frame of this timeline clip serialised as string (clip startTimecode + clipIn frames).
clips[i].timelineIn	string	Timeline position of the first frame of this clip, where 0 is the first frame of the timeline.
clips[i].timelineInTimecode	string	Timeline timecode of the first frame of this timeline clip serialised as string.

404 - No timeline / disk available.

DELETE /timelines/0

Clear the current playback timeline. Deprecated, prefer to use POST /timelines/0/clear

Response

204 - The timeline was cleared.

501 - The operation is not supported on this device.

POST /timelines/0

Add a clip to the timeline.

Parameters

This parameter can be one of the following types:

Add multiple media clips to the timeline with optional insertion point and clip in/out points.

Name	Type	Description
insertBefore	integer	Clip(s) will be inserted before the clip at this timeline clip index, where 0 inserts to the beginning of the timeline. If omitted, inserts to the end of the timeline.
clips (required)	array	List of clips to add to the timeline.
clips[i]	object	Clip to add to the timeline, optionally cropping the clip before adding to the timeline.
clips[i].clipUniqueId (required)	integer	Unique ID (clipUniqueId) of the media clip to add to the timeline.
clips[i].clipIn	integer	Insert this clip starting from this frame within the media clip. If omitted, starts from the beginning of the clip -- frame 0.
clips[i].frameCount	integer	Number of frames of this clip to add to the timeline. If omitted, use the whole clip, or the rest of the clip if clipIn was specified.

Add multiple media clips to the timeline with optional insertion point.

Name	Type	Description
insertBefore	integer	Clip(s) will be inserted before the clip at this timeline clip index, where 0 inserts to the beginning of the timeline. If omitted, inserts to the end of the timeline.
clips (required)	array	List of clips to add to the timeline.
clips[i]	integer	Unique ID (clipUniqueId) of the media clip to add to the timeline.

Add a single clip to the timeline with optional insertion point and clip in/out points.

Name	Type	Description
insertBefore	integer	Clip(s) will be inserted before the clip at this timeline clip index, where 0 inserts to the beginning of the timeline. If omitted, inserts to the end of the timeline.
clips (required)	object	Clip to add to the timeline, optionally cropping the clip before adding to the timeline.
clips.clipUniqueId (required)	integer	Unique ID (clipUniqueId) of the media clip to add to the timeline.
clips.clipIn	integer	Insert this clip starting from this frame within the media clip. If omitted, starts from the beginning of the clip -- frame 0.
clips.frameCount	integer	Number of frames of this clip to add to the timeline. If omitted, use the whole clip, or the rest of the clip if clipIn was specified.

Add a single clip to the timeline with optional insertion point.

Name	Type	Description
insertBefore	integer	Clip(s) will be inserted before the clip at this timeline clip index, where 0 inserts to the beginning of the timeline. If omitted, inserts to the end of the timeline.
clips (required)	integer	Unique ID (clipUniqueld) of the media clip to add to the timeline.

Response

204 - The clip was added to the timeline as specified.

501 - The operation is not supported on this device.

POST /timelines/0/add

Add a clip to the end of the timeline. Deprecated, use POST /timelines/0 to add clips within the timeline.

Parameters

This parameter can be one of the following types:

Add one clip to the end of the timeline.

Name	Type	Description
clips	integer	Unique ID (clipUniqueld) of the media clip to add to the timeline.

Add many clips to the end of the timeline.

Name	Type	Description
clips	array	List of clipUniquelds of clips to add to end of timeline.
clips[i]	integer	Unique ID (clipUniqueld) of the media clip to add to the timeline.

Response

204 - The clip was added to the end of the timeline.

501 - The operation is not supported on this device.

POST /timelines/0/clear

Clear the playback timeline.

Response

204 - The timeline was cleared.

501 - The operation is not supported on this device.

DELETE /timelines/0/clips/{timelineClipIndex}

Remove the specified clip from the timeline.

Parameters

Name	Type	Description
{timelineClipIndex} (required)	integer	The (0-based) timeline clip index of the clip to remove from the timeline.

Response

204 - The specified clip was removed from the timeline.

501 - The operation is not supported on this device.

Media Control API

API for controlling media devices in Blackmagic Design products.

GET /media/workingset

Get the list of media devices currently in the working set.

Response

200 - The list of media devices in the working set.

The response is JSON.

Name	Type	Description
size	integer	The fixed size of this device's working set.
workingset	array	The device's working set.
workingset[i]	object null	Device within the working set. null if no device is present within the given working set slot.
workingset[i].volume	string	Volume name.
workingset[i].deviceName	string	Internal device name of this media device.
workingset[i].remainingRecordTime	integer	Remaining record time using current codec and video format in seconds.
workingset[i].totalSpace	integer	Total space on media device in bytes.
workingset[i].remainingSpace	integer	Remaining space on media device in bytes.
workingset[i].clipCount	integer	Number of clips currently on the device.

GET /media/active

Get the currently active media device.

Response

200 - The current active media device.

The response is JSON.

The active media device, or null if there is no active media.

Name	Type	Description
workingsetIndex	integer	Working set index of the active media device.
deviceName	string	Device name of media device.

204 - No media is currently active.

PUT /media/active

Set the currently active media device.

Parameters

Name	Type	Description
workingsetIndex	integer	Working set index of the media to make active.

Response

204 - The active media device was set successfully.

400 - Setting the currently active media device is not possible in the current state.

GET /media/devices/doformatSupportedFilesystems

Get the list of filesystems available to format a media device.

Response

200 - The list of filesystems permitted for formatting.

The response is JSON.

Name	Type	Description
	array	List of filesystems permitted for formatting media.
[i]	string	Filesystem serialised as string.

GET /media/devices/{deviceName}

Get information about a requested device.

Parameters

Name	Type	Description
{deviceName} (required)	string	Device name of the media device. Retrieved by "deviceName" member of GET /media/workingset or GET /media/active.

Response

200 - Information about the requested device.

The response is JSON.

Media device state.

Name	Type	Description
state	string	The current state of the media device. Possible values are: None, Scanning, Mounted, Uninitialised, Formatting, RaidComponent.

400 - Invalid device name.

404 - Device not found.

GET /media/devices/{deviceName}/doformat

Get a format key, used to format the device with a PUT request.

Parameters

Name	Type	Description
{deviceName} (required)	string	Device name of the media device. Retrieved by "deviceName" member of GET /media/workingset or GET /media/active.

Response

200 - Format prepared.

The response is JSON.

Name	Type	Description
deviceName	string	Device name of media device to format.
key	string	The key required to format this device, provide to PUT /media/devices/{deviceName}/doformat to perform format of media device.

400 - Cannot format the device.

404 - Device not found.

PUT /media/devices/{deviceName}/doformat

Perform a format of the specified media device.

Parameters

Name	Type	Description
{deviceName} (required)	string	Device name of the media device. Retrieved by "deviceName" member of GET /media/workingset or GET /media/active.

Name	Type	Description
key	string	The key used to format this device, retrieved from prepare format media request GET /media/devices/{deviceName}/doformat. Format key provided cannot be reused after successful format.
filesystem	string	Filesystem to format to. Supported filesystems can be retrieved with GET /media/devices/doFormatSupportedFilesystems.
volume	string	Volume name to set for the disk after format.

Response

204 - Format successful.

400 - Cannot format the device, invalid filesystem or key.

404 - Device not found.

Slate Control API

API to manage digital slate data.

GET /slates/nextClip

Retrieve the digital slate for the next clip.

Response

200 - Returns the slate data for the next clip.

The response is JSON.

Name	Type	Description
clip	object	
clip.clipName	string	
clip.reel	integer	
clip.scene	string	
clip.sceneLocation	string	Possible values are: Interior, Exterior.
clip.sceneTime	string	Possible values are: Day, Night.
clip.shotType	string	Possible values are: None, WS, MS, MCU, CU, BCU, ECU.
clip.slateFor	string	Possible values are: Clip, Next Clip.
clip.take	integer	
clip.takeType	string	Possible values are: None, PU, VFX, SER.
clip.goodTake	boolean	
lens	object	

Name	Type	Description
lens.lensType	string	
lens.iris	string	
lens.focalLength	string	
lens.distance	string	
lens.filter	string	
project	object	
project.projectName	string	
project.director	string	
project.camera	string	
project.cameraOperator	string	

409 - Slate data is not available.

PUT /slates/nextClip

Update the slate data for the next clip.

Parameters

Name	Type	Description
clip	object	
clip.reel	integer	
clip.scene	string	
clip.take	integer	
clip.shotType	string	Possible values are: None, WS, MS, MCU, CU, BCU, ECU.
clip.takeType	string	Possible values are: None, PU, VFX, SER.
clip.sceneLocation	string	Possible values are: Interior, Exterior.
clip.sceneTime	string	Possible values are: Day, Night.
clip.goodTake	boolean	
lens	object	
lens.lensType	string	
lens.iris	string	
lens.focalLength	string	
lens.distance	string	
lens.filter	string	
project	object	
project.projectName	string	
project.director	string	
project.camera	string	
project.cameraOperator	string	

Response

200 - Successfully updated the slate data.

The response is JSON.

Name	Type	Description
clip	object	
clip.clipName	string	
clip.reel	integer	
clip.scene	string	
clip.sceneLocation	string	Possible values are: Interior, Exterior.
clip.sceneTime	string	Possible values are: Day, Night.
clip.shotType	string	Possible values are: None, WS, MS, MCU, CU, BCU, ECU.
clip.slateFor	string	Possible values are: Clip, Next Clip.
clip.take	integer	
clip.takeType	string	Possible values are: None, PU, VFX, SER.
clip.goodTake	boolean	
lens	object	
lens.lensType	string	
lens.iris	string	
lens.focalLength	string	
lens.distance	string	
lens.filter	string	
project	object	
project.projectName	string	
project.director	string	
project.camera	string	
project.cameraOperator	string	

409 - Partial update with errors.

The response is JSON.

Name	Type	Description
error	string	
details	array	
details[i]	object	
details[i].field	string	
details[i].message	string	

POST /slates/nextClip/resetProjectData

Reset the project data for the next clip's slate.

Response

200 - Project data reset successfully.

POST /slates/clips/{deviceName}/{path}/resetProjectData

Reset the project data for the next clip's slate.

Parameters

Name	Type	Description
{deviceName} (required)	string	Name of the device where the clip is stored. This is the same as the web browser's device name.
{path} (required)	string	Path to the clip.

Response

200 - Project data reset successfully.

POST /slates/nextClip/resetLensData

Reset the lens data for the next clip's slate.

Response

200 - Lens data reset successfully.

POST /slates/clips/{deviceName}/{path}/resetLensData

Reset the lens data for the next clip's slate.

Parameters

Name	Type	Description
{deviceName} (required)	string	Name of the device where the clip is stored. This is the same as the web browser's device name.
{path} (required)	string	Path to the clip.

Response

200 - Lens data reset successfully.

GET /slates/clips/{deviceName}/{path}

Retrieve slate data for a specific clip.

Parameters

Name	Type	Description
{deviceName} (required)	string	Name of the device where the clip is stored. This is the same as the web browser's device name.
{path} (required)	string	Path to the clip.

Response

200 - Returns the slate data for the specified clip.

The response is JSON.

Name	Type	Description
clip	object	
clip.clipName	string	
clip.reel	integer	
clip.scene	string	
clip.sceneLocation	string	Possible values are: Interior, Exterior.
clip.sceneTime	string	Possible values are: Day, Night.
clip.shotType	string	Possible values are: None, WS, MS, MCU, CU, BCU, ECU.
clip.slateFor	string	Possible values are: Clip, Next Clip.
clip.take	integer	
clip.takeType	string	Possible values are: None, PU, VFX, SER.
clip.goodTake	boolean	
lens	object	
lens.lensType	string	
lens.iris	string	
lens.focalLength	string	
lens.distance	string	
lens.filter	string	
project	object	
project.projectName	string	
project.director	string	
project.camera	string	
project.cameraOperator	string	

404 - Clip not found.

PUT /slates/clips/{deviceName}/{path}

Update the slate data for a specific clip.

Parameters

Name	Type	Description
{deviceName} (required)	string	Name of the device where the clip is stored. This is the same as the web browser's device name.
{path} (required)	string	Path to the clip.

Name	Type	Description
clip	object	
clip.reel	integer	
clip.scene	string	
clip.take	integer	
clip.shotType	string	Possible values are: None, WS, MS, MCU, CU, BCU, ECU.
clip.takeType	string	Possible values are: None, PU, VFX, SER.
clip.sceneLocation	string	Possible values are: Interior, Exterior.
clip.sceneTime	string	Possible values are: Day, Night.
clip.goodTake	boolean	
lens	object	
lens.lensType	string	
lens.iris	string	
lens.focalLength	string	
lens.distance	string	
lens.filter	string	
project	object	
project.projectName	string	
project.director	string	
project.camera	string	
project.cameraOperator	string	

Response

200 - Successfully updated the slate data.

The response is JSON.

Name	Type	Description
clip	object	
clip.clipName	string	
clip.reel	integer	
clip.scene	string	
clip.sceneLocation	string	Possible values are: Interior, Exterior.
clip.sceneTime	string	Possible values are: Day, Night.
clip.shotType	string	Possible values are: None, WS, MS, MCU, CU, BCU, ECU.
clip.slateFor	string	Possible values are: Clip, Next Clip.

Name	Type	Description
clip.take	integer	
clip.takeType	string	Possible values are: None, PU, VFX, SER.
clip.goodTake	boolean	
lens	object	
lens.lensType	string	
lens.iris	string	
lens.focalLength	string	
lens.distance	string	
lens.filter	string	
project	object	
project.projectName	string	
project.director	string	
project.camera	string	
project.cameraOperator	string	

409 - Partial update with errors.

The response is JSON.

Name	Type	Description
error	string	
details	array	
details[i]	object	
details[i].field	string	
details[i].message	string	

Preset Control API

API For controlling the presets on Blackmagic Design products

GET /presets

Get the list of the presets on the camera

Response

200 - OK

The response is JSON.

Name	Type	Description
presets	array	List of the presets on the camera (.cset files)
presets[i]	string	

POST /presets

Send a preset file to the camera

Response

200 - OK

The response is JSON.

Name	Type	Description
presetAdded	string	Name of the preset uploaded (without .cset extension)

400 - Bad request - missing Content-Disposition header or filename

GET /presets/active

Get the currently active preset on the camera

Response

200 - OK

The response is JSON.

Name	Type	Description
preset	string	Name of the active preset (with .cset extension, or 'default')

PUT /presets/active

Set the active preset on the camera

Parameters

Name	Type	Description
preset	string	Name of the active preset (with .cset extension, or 'default')

Response

204 - No Content

404 - Preset file not found

PARAMETERS /presets/{presetName}

GET /presets/{presetName}

Download the preset file

Response

200 - OK

The response is a binary file.

404 - File does not exist

PUT /presets/{presetName}

Save current camera state as a preset

Response

204 - No Content

DELETE /presets/{presetName}

Delete a preset from the camera

Response

204 - No Content

404 - Preset file not found

Audio Control API

API For controlling audio on Blackmagic Design Cameras

GET /audio/channels

Get the total number of audio channels available

Response

200 - Returns the total number of channels

The response is JSON.

Name	Type	Description
channels	integer	Total number of audio channels available

GET /audio/supportedInputs

Get the list of supported audio inputs

Response

200 - List of all supported audio inputs

The response is JSON.

Name	Type	Description
	array	
[i]	string	A supported audio input

GET /audio/channel/{channelIndex}/input

Get the audio input (source and type) for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Response

200 - Currently selected input

The response is JSON.

Name	Type	Description
input	string	Audio input source and type

404 - Channel does not exist

PUT /audio/channel/{channelIndex}/input

Set the audio input for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Name	Type	Description
input	string	Audio input source and type

Response

204 - No Content

400 - Invalid audio input

404 - Channel does not exist

GET /audio/channel/{channelIndex}/input/description

Get the description of the current input of the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Response

200 - Description of the current input of the selected channel

The response is JSON.

Name	Type	Description
description	object	
description.gainRange	object	
description.gainRange.Min	number	The minimum gain value in dB
description.gainRange.Max	number	The maximum gain value in dB
description.capabilities	object	
description.capabilities.PhantomPower	boolean	Input supports setting of phantom power
description.capabilities.LowCutFilter	boolean	Input supports setting of low cut filter
description.capabilities.Padding	object	
description.capabilities.Padding.available	boolean	Input supports setting of padding
description.capabilities.Padding.forced	boolean	Padding is forced to be set for the input
description.capabilities.Padding.value	number	Value of the padding in dB

404 - Channel does not exist

GET /audio/channel/{channelIndex}/supportedInputs

Get the list of supported inputs and their availability to switch to for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its supported inputs are being queried. (Channels index from 0)

Response

200 - The list of supported inputs

The response is JSON.

Name	Type	Description
	array	
[i]	object	
[i].input	string	Input name
[i].available	boolean	Is the input available to be switched into from the current input for the selected channel

404 - Channel does not exist

GET /audio/channel/{channelIndex}/level

Get the audio input level for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Response

200 - Currently set level for the selected channel

The response is JSON.

Name	Type	Description
gain	number	Gain value in dB
normalised	number	Normalised level value between 0.0 and 1.0

404 - Channel does not exist

PUT /audio/channel/{channelIndex}/level

Set the audio input level for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Name	Type	Description
gain	number	Gain value in dB
normalised	number	Normalised level value between 0.0 and 1.0

Response

204 - No Content

400 - Invalid input or value out of range

404 - Channel does not exist

GET /audio/channel/{channelIndex}/phantomPower

Get the audio input phantom power status for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Response

200 - Currently set phantom power for the selected channel

The response is JSON.

Name	Type	Description
enabled	boolean	Phantom power enabled state

404 - Channel does not exist

PUT /audio/channel/{channelIndex}/phantomPower

Set the audio phantom power for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Name	Type	Description
enabled	boolean	Phantom power enabled state

Response

204 - No Content

400 - Phantom power is not supported for this input

404 - Channel does not exist

GET /audio/channel/{channelIndex}/padding

Get the audio input padding status for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Response

200 - Currently set padding for the selected channel

The response is JSON.

Name	Type	Description
enabled	boolean	Padding enabled state

404 - Channel does not exist

PUT /audio/channel/{channelIndex}/padding

Set the audio input padding for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Name	Type	Description
enabled	boolean	Padding enabled state

Response

204 - No Content

400 - Padding is not supported or is forced for this input

404 - Channel does not exist

GET /audio/channel/{channelIndex}/lowCutFilter

Get the audio input low cut filter status for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Response

200 - Currently set low cut filter for the selected channel

The response is JSON.

Name	Type	Description
enabled	boolean	Low cut filter enabled state

404 - Channel does not exist

PUT /audio/channel/{channelIndex}/lowCutFilter

Set the audio input low cut filter for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Name	Type	Description
enabled	boolean	Low cut filter enabled state

Response

204 - No Content

400 - Low cut filter is not supported for this input

404 - Channel does not exist

GET /audio/channel/{channelIndex}/available

Get the audio input's current availability for the selected channel. If unavailable, the source will be muted

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Response

200 - Currently set availability for the selected channel

The response is JSON.

Name	Type	Description
available	boolean	Whether the input is currently available

404 - Channel does not exist

Lens Control API

API For controlling the lens on Blackmagic Design products

GET /lens/iris

Get lens' aperture

Response

200 - OK

The response is JSON.

Name	Type	Description
continuousApertureAutoExposure	boolean	Is Aperture controlled by auto exposure
apertureStop	number	Aperture stop value
normalised	number	Normalised value
apertureNumber	integer	Aperture number

PUT /lens/iris

Set lens' aperture

Parameters

Name	Type	Description
apertureStop	number	Aperture stop value
normalised	number	Normalised value
apertureNumber	integer	Aperture number
adjustmentStep	integer	Signed value for relative aperture adjustment

Response

204 - No Content

400 - Bad Request if out of range value is provided

403 - Forbidden if lens iris is not controllable or is controlled by auto exposure

GET /lens/zoom

Get lens' zoom

Response

200 - OK

The response is JSON.

Name	Type	Description
focalLength	integer	Focal length in mm
normalised	number	Normalised value

PUT /lens/zoom

Set lens' zoom

Parameters

Name	Type	Description
focalLength	integer	Focal length in mm
normalised	number	Normalised value
adjustmentFocalLength	integer	Signed value for relative focal length adjustment
adjustmentNormalised	number	Signed normalized value for relative zoom adjustment

Response

204 - No Content

400 - Bad Request if out of range value is provided

403 - Forbidden if lens zoom is not controllable

GET /lens/focus

Get lens' focus

Response

200 - OK

The response is JSON.

Name	Type	Description
normalised	number	Normalised value

PUT /lens/focus

Set lens' focus

Parameters

Name	Type	Description
normalised	number	Normalised value
focusDistance	integer	Focus distance value

Response

204 - No Content

400 - Bad Request if out of range value is provided

PUT /lens/focus/doAutoFocus

Perform auto focus

Parameters

Name	Type	Description
position	object	
position.x	number	Normalized x coordinate for autofocus ROI
position.y	number	Normalized y coordinate for autofocus ROI

Response

204 - No Content

400 - Bad Request if out of range value is provided

403 - Forbidden if lens focus is not controllable

GET /lens/opticalImageStabilization

Get optical image stabilization status

Response

200 - OK

The response is JSON.

Name	Type	Description
enabled	boolean	Whether optical image stabilization is enabled

501 - Not Implemented if optical image stabilization is not supported on this product

PUT /lens/opticalImageStabilization

Enable or disable optical image stabilization

Parameters

Name	Type	Description
enabled	boolean	Enable or disable optical image stabilization

Response

204 - No Content

501 - Not Implemented if optical image stabilization is not supported on this product

GET /lens/iris/description

Get detailed description of lens' iris capabilities

Response

200 - OK

The response is JSON.

Name	Type	Description
controllable	boolean	If the iris can be controlled
apertureStop	object	
apertureStop.min	number	Minimum aperture stop
apertureStop.max	number	Maximum aperture stop

GET /lens/zoom/description

Get detailed description of lens' zoom capabilities

Response

200 - OK

The response is JSON.

Name	Type	Description
controllable	boolean	If the zoom can be controlled
focalLength	object	
focalLength.adjustable	boolean	If focal length is adjustable
focalLength.min	integer	Minimum focal length
focalLength.max	integer	Maximum focal length

GET /lens/focus/description

Get detailed description of lens' focus capabilities

Response

200 - OK

The response is JSON.

Name	Type	Description
controllable	boolean	If the focus can be controlled
focusDistance	object	
focusDistance.adjustable	boolean	If focus distance is adjustable
focusDistance.min	number	Minimum focus distance
focusDistance.max	number	Maximum focus distance

Video Control API

API For controlling the video on Blackmagic Design products

GET /video/iso

Get current ISO

Response

200 - OK

The response is JSON.

Name	Type	Description
iso	integer	Current ISO value

PUT /video/iso

Set current ISO

Parameters

Name	Type	Description
iso	integer	ISO value to set

Response

204 - No Content

403 - ISO cannot be changed in the current state

GET /video/supportedISOs

Get the list of supported ISO settings

Response

200 - List of supported ISO values

The response is JSON.

Name	Type	Description
supportedISOs	array	Array of supported ISO values
supportedISOs[i]	integer	

GET /video/gain

Get current gain value in decibels

Response

200 - OK

The response is JSON.

Name	Type	Description
gain	integer	Current gain value in decibels

PUT /video/gain

Set current gain value

Parameters

Name	Type	Description
gain	integer	Gain value in decibels to set

Response

204 - No Content

403 - Gain cannot be changed in the current state

GET /video/supportedGains

Get the list of supported gain settings in decibels

Response

200 - List of supported gain values in decibels

The response is JSON.

Name	Type	Description
supportedGains	array	Array of supported gain values in decibels
supportedGains[i]	integer	

GET /video/whiteBalance

Get current white balance

Response

200 - OK

The response is JSON.

Name	Type	Description
whiteBalance	integer	Current white balance

PUT /video/whiteBalance

Set current white balance

Parameters

Name	Type	Description
whiteBalance	integer	White balance to set

Response

204 - No Content

400 - Invalid white balance temperature

GET /video/whiteBalance/description

Get white balance range

Response

200 - OK

The response is JSON.

Name	Type	Description
whiteBalance	object	
whiteBalance.min	integer	Minimum color temperature
whiteBalance.max	integer	Maximum color temperature

PUT /video/whiteBalance/doAuto

Set current white balance automatically

Response

204 - No Content

GET /video/whiteBalanceTint

Get white balance tint

Response

200 - OK

The response is JSON.

Name	Type	Description
whiteBalanceTint	integer	Current white balance tint

PUT /video/whiteBalanceTint

Set white balance tint

Parameters

Name	Type	Description
whiteBalanceTint	integer	White balance tint to set

Response

204 - No Content

400 - Invalid white balance tint

GET /video/whiteBalanceTint/description

Get white balance tint range

Response

200 - OK

The response is JSON.

Name	Type	Description
whiteBalanceTint	object	
whiteBalanceTint.min	integer	Minimum white balance tint
whiteBalanceTint.max	integer	Maximum white balance tint

GET /video/ndFilter

Get ND filter stop

Response

200 - OK

The response is JSON.

Name	Type	Description
stop	number	Current filter power (fStop)

501 - Not implemented for this device

PUT /video/ndFilter

Set ND filter stop

Parameters

Name	Type	Description
stop	number	Filter power (fStop) to set

Response

204 - No Content

400 - Invalid ND filter stop

501 - Not implemented for this device

GET /video/supportedNDFilters

Get the list of available ND filter stops

Response

200 - List of available ND filter stops

The response is JSON.

Name	Type	Description
supportedStops	array	Array of available ND filter stops
supportedStops[i]	number	

501 - Not implemented for this device

GET /video/supportedNDFilterDisplayModes

Get the list of supported ND filter display modes

Response

200 - List of supported display modes

The response is JSON.

Name	Type	Description
supportedDisplayModes	array	Array of supported display modes
supportedDisplayModes[i]	string	Possible values are: Stop, Number, Fraction.

501 - Not implemented for this device

GET /video/ndFilter/displayMode

Get ND filter display mode on the camera

Response

200 - OK

The response is JSON.

Name	Type	Description
displayMode	string	Possible values are: Stop, Number, Fraction.

501 - Not implemented for this device

PUT /video/ndFilter/displayMode

Set ND filter display mode on the camera

Parameters

Name	Type	Description
displayMode	string	Possible values are: Stop, Number, Fraction.

Response

204 - No Content

400 - Invalid display mode for ND filter

501 - Not implemented for this device

GET /video/ndFilterSelectable

Check if ND filter adjustments are selectable via a slider

Response

200 - Indicates if ND filter is selectable

The response is JSON.

Name	Type	Description
selectable	boolean	True if ND filter adjustments are selectable via a slider

501 - Not implemented for this device

GET /video/shutter

Get current shutter. Will return either shutter speed or shutter angle depending on shutter measurement in device settings

Response

200 - OK

The response is JSON.

Name	Type	Description
continuousShutterAutoExposure	boolean	Is shutter controlled by auto exposure
shutterSpeed	integer	Shutter speed value in fractions of a second (minimum is sensor frame rate)
shutterAngle	number	Shutter angle

PUT /video/shutter

Set current shutter

Parameters

Name	Type	Description
shutterSpeed	integer	Shutter speed value in fractions of a second (minimum is sensor frame rate)
shutterAngle	number	Shutter angle

Response

204 - No Content

GET /video/shutter/measurement

Get the current shutter measurement mode

Response

200 - OK

The response is JSON.

Name	Type	Description
measurement	string	Possible values are: ShutterAngle, ShutterSpeed.

PUT /video/shutter/measurement

Set the shutter measurement mode

Parameters

Name	Type	Description
measurement	string	Possible values are: ShutterAngle, ShutterSpeed.

Response

204 - No Content

400 - Invalid shutter measurement

GET /video/supportedShutters

Get supported shutter settings based on current camera configuration

Response

200 - OK

The response is JSON.

Name	Type	Description
shutterAngles	array	Array of supported shutter angles
shutterAngles[i]	number	Shutter angle
shutterSpeeds	array	Array of supported shutter speeds
shutterSpeeds[i]	integer	Shutter speed value in fractions of a second (minimum is sensor frame rate)

GET /video/flickerFreeShutters

Get flicker-free shutter settings based on current camera configuration

Response

200 - OK

The response is JSON.

Name	Type	Description
shutterAngles	array	Array of flicker-free shutter angles
shutterAngles[i]	number	Shutter angle
shutterSpeeds	array	Array of flicker-free shutter speeds
shutterSpeeds[i]	integer	Shutter speed value in fractions of a second (minimum is sensor frame rate)

GET /video/autoExposure

Get current auto exposure mode

Response

200 - OK

The response is JSON.

Name	Type	Description
mode	string	Auto exposure mode Possible values are: Off, Continuous, OneShot.
type	string	Comma-separated list of device types in the auto exposure stack

PUT /video/autoExposure

Set auto exposure

Parameters

Name	Type	Description
mode	string	Auto exposure mode Possible values are: Off, Continuous, OneShot.
type	string	Comma-separated list of device types in the auto exposure stack

Response

204 - No Content

400 - Failed to set auto exposure mode

GET /video/detailSharpening

Get the current state of detail sharpening

Response

200 - Current detail sharpening state

The response is JSON.

Name	Type	Description
enabled	boolean	Whether detail sharpening is enabled

501 - Not implemented for this device

PUT /video/detailSharpening

Enable or disable detail sharpening

Parameters

Name	Type	Description
enabled	boolean	Enable or disable detail sharpening

Response

204 - Detail sharpening state updated

501 - Not implemented for this device

GET /video/detailSharpeningLevel

Get the current detail sharpening level

Response

200 - Current detail sharpening level

The response is JSON.

Name	Type	Description
level	string	Current detail sharpening level Possible values are: Low, Medium, High.

501 - Not implemented for this device

PUT /video/detailSharpeningLevel

Set the detail sharpening level

Parameters

Name	Type	Description
level	string	Desired level of detail sharpening Possible values are: Low, Medium, High.

Response

204 - Detail sharpening level updated

400 - Invalid detail sharpening level

501 - Not implemented for this device

Camera Control API

API For controlling the Camera specific features on Blackmagic Design products

GET /camera/colorBars

Get the status of color bars display

Response

200 - Returns the current status of color bars

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the color bars are currently enabled

PUT /camera/colorBars

Set the status of color bars display

Parameters

Name	Type	Description
enabled	boolean	Enable or disable the color bars

Response

204 - Color bars status updated successfully

GET /camera/programFeedDisplay

Get the status of program feed display

Response

200 - Returns the current status of program feed display

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the program feed display is currently enabled

PUT /camera/programFeedDisplay

Set the status of program feed display

Parameters

Name	Type	Description
enabled	boolean	Enable or disable the program feed display

Response

204 - Program feed display status updated successfully

GET /camera/tallyStatus

Get the tally status of the camera

Response

200 - Returns the current tally status of the camera

The response is JSON.

Name	Type	Description
status	string	Current tally status of the camera Possible values are: None, Preview, Program.

GET /camera/power

Get the power status of the camera

Response

200 - Returns the current power status

The response is JSON.

Name	Type	Description
source	string	Current power source of the camera Possible values are: Battery, AC, Fiber, USB, POE.
milliVolt	integer	Current voltage level in millivolts (rounded to nearest 100mV)
batteries	array	List of batteries currently connected to the camera
batteries[i]	object	
batteries[i].milliVolt	integer	Battery voltage in millivolts (rounded to nearest 100mV)
batteries[i].chargeRemainingPercent	integer	Remaining battery charge percentage
batteries[i].statusFlags	array	List of battery status flags
batteries[i].statusFlags[i]	string	Possible values are: Unknown Battery Status, Battery Is Present, Battery Is Charging, Battery Percentage Is Low, Battery Voltage Is Low, Battery Is Critically Low, Charge Remaining Percentage Is Estimated, Battery Communications Is Active, Battery Is Connected.

GET /camera/power/displayMode

Get the power display mode of the camera

Response

200 - Returns the current power display mode

The response is JSON.

Name	Type	Description
mode	string	Current power display mode Possible values are: Percentage, Voltage.

PUT /camera/power/displayMode

Set the power display mode of the camera

Parameters

Name	Type	Description
mode	string	Power display mode to set Possible values are: Percentage, Voltage.

Response

204 - Power display mode updated successfully

400 - Invalid power display mode

GET /camera/timingReferenceLock

Get the timing reference lock status

Response

200 - Returns the timing reference lock status

The response is JSON.

Name	Type	Description
locked	boolean	Indicates if timing reference is locked

Immersive Control API

API for controlling immersive camera settings on Blackmagic Design cameras

GET /immersive/display/{displayName}/eye

Get the current eye view for a specific display

Parameters

Name	Type	Description
{displayName} (required)	string	The display name to query (from /monitoring/display endpoint)

Response

200 - OK

The response is JSON.

Name	Type	Description
eye (required)	string	The eye view to display Possible values are: Left, Right.

400 - Invalid display name format

404 - Display not found

422 - Failed to get eye view

PUT /immersive/display/{displayName}/eye

Set the eye view for a specific display

Parameters

Name	Type	Description
{displayName} (required)	string	The display name to control (from /monitoring/display endpoint)

Name	Type	Description
eye (required)	string	The eye view to display Possible values are: Left, Right.

Response

204 - No Content

400 - Invalid input or display name format

404 - Display not found

422 - Failed to set eye view

Color Correction Control API

API For controlling the color correction on Blackmagic Design products based on DaVinci Resolve Color Corrector

GET /colorCorrection/lift

Get color correction lift

Response

200 - OK

The response is JSON.

Name	Type	Description
red	number	Red lift component. If omitted, value remains unchanged.
green	number	Green lift component. If omitted, value remains unchanged.
blue	number	Blue lift component. If omitted, value remains unchanged.
luma	number	Luma lift component. If omitted, value remains unchanged.

PUT /colorCorrection/lift

Set color correction lift

Parameters

Name	Type	Description
red	number	Red lift component. If omitted, value remains unchanged.
green	number	Green lift component. If omitted, value remains unchanged.
blue	number	Blue lift component. If omitted, value remains unchanged.
luma	number	Luma lift component. If omitted, value remains unchanged.

Response

204 - No Content

GET /colorCorrection/gamma

Get color correction gamma

Response

200 - OK

The response is JSON.

Name	Type	Description
red	number	Red gamma component. If omitted, value remains unchanged.
green	number	Green gamma component. If omitted, value remains unchanged.
blue	number	Blue gamma component. If omitted, value remains unchanged.
luma	number	Luma gamma component. If omitted, value remains unchanged.

PUT /colorCorrection/gamma

Set color correction gamma

Parameters

Name	Type	Description
red	number	Red gamma component. If omitted, value remains unchanged.
green	number	Green gamma component. If omitted, value remains unchanged.
blue	number	Blue gamma component. If omitted, value remains unchanged.
luma	number	Luma gamma component. If omitted, value remains unchanged.

Response

204 - No Content

GET /colorCorrection/gain

Get color correction gain

Response

200 - OK

The response is JSON.

Name	Type	Description
red	number	Red gain component. If omitted, value remains unchanged.
green	number	Green gain component. If omitted, value remains unchanged.
blue	number	Blue gain component. If omitted, value remains unchanged.
luma	number	Luma gain component. If omitted, value remains unchanged.

PUT /colorCorrection/gain

Set color correction gain

Parameters

Name	Type	Description
red	number	Red gain component. If omitted, value remains unchanged.
green	number	Green gain component. If omitted, value remains unchanged.
blue	number	Blue gain component. If omitted, value remains unchanged.
luma	number	Luma gain component. If omitted, value remains unchanged.

Response

204 - No Content

GET /colorCorrection/offset

Get color correction offset

Response

200 - OK

The response is JSON.

Name	Type	Description
red	number	Red offset component. If omitted, value remains unchanged.
green	number	Green offset component. If omitted, value remains unchanged.
blue	number	Blue offset component. If omitted, value remains unchanged.
luma	number	Luma offset component. If omitted, value remains unchanged.

PUT /colorCorrection/offset

Set color correction offset

Parameters

Name	Type	Description
red	number	Red offset component. If omitted, value remains unchanged.
green	number	Green offset component. If omitted, value remains unchanged.
blue	number	Blue offset component. If omitted, value remains unchanged.
luma	number	Luma offset component. If omitted, value remains unchanged.

Response

204 - No Content

GET /colorCorrection/contrast

Get color correction contrast

Response

200 - OK

The response is JSON.

Name	Type	Description
pivot	number	Contrast pivot point. If omitted, value remains unchanged.
adjust	number	Contrast adjustment. If omitted, value remains unchanged.

PUT /colorCorrection/contrast

Set color correction contrast

Parameters

Name	Type	Description
pivot	number	Contrast pivot point. If omitted, value remains unchanged.
adjust	number	Contrast adjustment. If omitted, value remains unchanged.

Response

204 - No Content

GET /colorCorrection/color

Get color correction color properties

Response

200 - OK

The response is JSON.

Name	Type	Description
hue	number	Color hue adjustment. If omitted, value remains unchanged.
saturation	number	Color saturation adjustment. If omitted, value remains unchanged.

PUT /colorCorrection/color

Set color correction color properties

Parameters

Name	Type	Description
hue	number	Color hue adjustment. If omitted, value remains unchanged.
saturation	number	Color saturation adjustment. If omitted, value remains unchanged.

Response

204 - No Content

GET /colorCorrection/lumaContribution

Get color correction luma contribution

Response

200 - OK

The response is JSON.

Name	Type	Description
lumaContribution	number	Luma contribution value. If omitted, value remains unchanged.

PUT /colorCorrection/lumaContribution

Set color correction luma contribution

Parameters

Name	Type	Description
lumaContribution	number	Luma contribution value. If omitted, value remains unchanged.

Response

204 - No Content

Notification websocket - 1.0.0

Service that notifies subscribers of device state changes.

messages

Subscribe (The messages from the server/device)

Websocket Opened Message (JSON)

Name	Type	Description
.data	object	
.data.action	string	Possible values are: websocketOpened.
.type	string	Possible values are: event.

Response Message (JSON)

Name	Type	Description
.data	object	
.data.action	string	Possible values are: subscribe, unsubscribe, listSubscriptions, listProperties, websocketOpened.
.data.properties	array	
.data.properties[i]	string	device property that the user can subscribe to. The user can either choose a value from the predefined enum, or provide a wildcard string. Possible values are: /media/workingset, /media/active, /system, /system/codecFormat, /system/videoFormat, /system/format, /system/supportedFormats, /timelines/0, /transports/0, /transports/0/stop, /transports/0/play, /transports/0/playback, /transports/0/record, /transports/0/timecode, /transports/0/timecode/source, /transports/0/clipIndex, /slates/nextClip, /monitoring/{displayName}/cleanFeed, /monitoring/{displayName}/displayLUT, /monitoring/{displayName}/zebra, /monitoring/{displayName}/focusAssist, /monitoring/{displayName}/frameGuide, /monitoring/{displayName}/frameGrids, /monitoring/{displayName}/safeArea, /monitoring/{displayName}/falseColor, /monitoring/focusAssist, /monitoring/frameGuideRatio, /monitoring/frameGrids, /monitoring/safeAreaPercent, /audio/channel/{channelIndex}/input, /audio/channel/{channelIndex}/supportedInputs, /audio/channel/{channelIndex}/level, /audio/channel/{channelIndex}/phantomPower, /audio/channel/{channelIndex}/padding, /audio/channel/{channelIndex}/lowCutFilter, /audio/channel/{channelIndex}/available, /audio/channel/{channelIndex}/input/description, /colorCorrection/lift, /colorCorrection/gamma, /colorCorrection/gain, /colorCorrection/offset, /colorCorrection/contrast, /colorCorrection/color, /colorCorrection/lumaContribution, /lens/iris, /lens/iris/description, /lens/focus, /lens/focus/description, /lens/zoom, /lens/zoom/description, /presets, /presets/active, /camera/colorBars, /camera/programFeedDisplay, /camera/tallyStatus, /camera/power, /camera/power/displayMode, /camera/timingReferenceLock, /video/iso, /video/supportedISOs, /video/gain, /video/supportedGains, /video/whiteBalance, /video/whiteBalance/description, /video/whiteBalanceTint, /video/whiteBalanceTint/description, /video/ndFilter, /video/supportedNDFilters, /video/ndFilter/displayMode, /video/supportedNDFilterDisplayModes, /video/ndFilterSelectable, /video/shutter, /video/shutter/measurement, /video/supportedShutters, /video/flickerFreeShutters, /video/autoExposure, /video/detailSharpening, /video/detailSharpeningLevel. Must match the pattern: *

Name	Type	Description
.data.values	object	An object with property names as the key and a property value as json. Check the next section for the device properties and their return values.
.data.success	boolean	
.data.deviceProperties	array	
.data.deviceProperties[i]	string	device property that the user can subscribe to. The user can either choose a value from the predefined enum, or provide a wildcard string. Possible values are: /media/workingset, /media/active, /system, /system/codecFormat, /system/videoFormat, /system/format, /system/supportedFormats, /timelines/0, /transports/0, /transports/0/stop, /transports/0/play, /transports/0/playback, /transports/0/record, /transports/0/timecode, /transports/0/timecode/source, /transports/0/clipIndex, /slates/nextClip, /monitoring/{displayName}/cleanFeed, /monitoring/{displayName}/displayLUT, /monitoring/{displayName}/zebra, /monitoring/{displayName}/focusAssist, /monitoring/{displayName}/frameGuide, /monitoring/{displayName}/frameGrids, /monitoring/{displayName}/safeArea, /monitoring/{displayName}/falseColor, /monitoring/focusAssist, /monitoring/frameGuideRatio, /monitoring/frameGrids, /monitoring/safeAreaPercent, /audio/channel/{channelIndex}/input, /audio/channel/{channelIndex}/supportedInputs, /audio/channel/{channelIndex}/level, /audio/channel/{channelIndex}/phantomPower, /audio/channel/{channelIndex}/padding, /audio/channel/{channelIndex}/lowCutFilter, /audio/channel/{channelIndex}/available, /audio/channel/{channelIndex}/input/description, /colorCorrection/lift, /colorCorrection/gamma, /colorCorrection/gain, /colorCorrection/offset, /colorCorrection/contrast, /colorCorrection/color, /colorCorrection/lumaContribution, /lens/iris, /lens/iris/description, /lens/focus, /lens/focus/description, /lens/zoom, /lens/zoom/description, /presets, /presets/active, /camera/colorBars, /camera/programFeedDisplay, /camera/tallyStatus, /camera/power, /camera/power/displayMode, /camera/timingReferenceLock, /video/iso, /video/supportedISOs, /video/gain, /video/supportedGains, /video/whiteBalance, /video/whiteBalance/description, /video/whiteBalanceTint, /video/whiteBalanceTint/description, /video/ndFilter, /video/supportedNDFilters, /video/ndFilter/displayMode, /video/supportedNDFilterDisplayModes, /video/ndFilterSelectable, /video/shutter, /video/shutter/measurement, /video/supportedShutters, /video/flickerFreeShutters, /video/autoExposure, /video/detailSharpening, /video/detailSharpeningLevel. Must match the pattern: .*
.type	string	Possible values are: response.
.id	number	Optional parameter that repeats the id in the output for tracking messages.

Event Message (JSON)

Name	Type	Description
.data	object	
.data.action	string	Possible values are: propertyValueChanged.

Name	Type	Description
.data.property	string	device property that the user can subscribe to. The user can either choose a value from the predefined enum, or provide a wildcard string. Possible values are: /media/workingset, /media/active, /system, /system/codecFormat, /system/videoFormat, /system/format, /system/supportedFormats, /timelines/0, /transports/0, /transports/0/stop, /transports/0/play, /transports/0/playback, /transports/0/record, /transports/0/timecode, /transports/0/timecode/source, /transports/0/clipIndex, /slates/nextClip, /monitoring/{displayName}/cleanFeed, /monitoring/{displayName}/displayLUT, /monitoring/{displayName}/zebra, /monitoring/{displayName}/focusAssist, /monitoring/{displayName}/frameGuide, /monitoring/{displayName}/frameGrids, /monitoring/{displayName}/safeArea, /monitoring/{displayName}/falseColor, /monitoring/focusAssist, /monitoring/frameGuideRatio, /monitoring/frameGrids, /monitoring/safeAreaPercent, /audio/channel/{channelIndex}/input, /audio/channel/{channelIndex}/supportedInputs, /audio/channel/{channelIndex}/level, /audio/channel/{channelIndex}/phantomPower, /audio/channel/{channelIndex}/padding, /audio/channel/{channelIndex}/lowCutFilter, /audio/channel/{channelIndex}/available, /audio/channel/{channelIndex}/input/description, /colorCorrection/lift, /colorCorrection/gamma, /colorCorrection/gain, /colorCorrection/offset, /colorCorrection/contrast, /colorCorrection/color, /colorCorrection/lumaContribution, /lens/iris, /lens/iris/description, /lens/focus, /lens/focus/description, /lens/zoom, /lens/zoom/description, /presets, /presets/active, /camera/colorBars, /camera/programFeedDisplay, /camera/tallyStatus, /camera/power, /camera/power/displayMode, /camera/timingReferenceLock, /video/iso, /video/supportedISOs, /video/gain, /video/supportedGains, /video/whiteBalance, /video/whiteBalance/description, /video/whiteBalanceTint, /video/whiteBalanceTint/description, /video/ndFilter, /video/supportedNDFilters, /video/ndFilter/displayMode, /video/supportedNDFilterDisplayModes, /video/ndFilterSelectable, /video/shutter, /video/shutter/measurement, /video/supportedShutters, /video/flickerFreeShutters, /video/autoExposure, /video/detailSharpening, /video/detailSharpeningLevel. Must match the pattern: *
.data.value	object	An object with property names as the key and a property value as json. Check the next section for the device properties and their return values.
.type	string	Possible values are: event.

Publish (The messages that user can send to the server/device)

Response Message (JSON)

Name	Type	Description
.data	object	
.data.action	string	Possible values are: subscribe, unsubscribe, listSubscriptions, listProperties, websocketOpened.
.data.properties	array	

Name	Type	Description
.data.properties[i]	string	device property that the user can subscribe to. The user can either choose a value from the predefined enum, or provide a wildcard string. Possible values are: /media/workingset, /media/active, /system, /system/codecFormat, /system/videoFormat, /system/format, /system/supportedFormats, /timelines/0, /transports/0, /transports/0/stop, /transports/0/play, /transports/0/playback, /transports/0/record, /transports/0/timecode, /transports/0/timecode/source, /transports/0/clipIndex, /slates/nextClip, /monitoring/{displayName}/cleanFeed, /monitoring/{displayName}/displayLUT, /monitoring/{displayName}/zebra, /monitoring/{displayName}/focusAssist, /monitoring/{displayName}/frameGuide, /monitoring/{displayName}/frameGrids, /monitoring/{displayName}/safeArea, /monitoring/{displayName}/falseColor, /monitoring/focusAssist, /monitoring/frameGuideRatio, /monitoring/frameGrids, /monitoring/safeAreaPercent, /audio/channel/{channelIndex}/input, /audio/channel/{channelIndex}/supportedInputs, /audio/channel/{channelIndex}/level, /audio/channel/{channelIndex}/phantomPower, /audio/channel/{channelIndex}/padding, /audio/channel/{channelIndex}/lowCutFilter, /audio/channel/{channelIndex}/available, /audio/channel/{channelIndex}/input/description, /colorCorrection/lift, /colorCorrection/gamma, /colorCorrection/gain, /colorCorrection/offset, /colorCorrection/contrast, /colorCorrection/color, /colorCorrection/lumaContribution, /lens/iris, /lens/iris/description, /lens/focus, /lens/focus/description, /lens/zoom, /lens/zoom/description, /presets, /presets/active, /camera/colorBars, /camera/programFeedDisplay, /camera/tallyStatus, /camera/power, /camera/power/displayMode, /camera/timingReferenceLock, /video/iso, /video/supportedISOs, /video/gain, /video/supportedGains, /video/whiteBalance, /video/whiteBalance/description, /video/whiteBalanceTint, /video/whiteBalanceTint/description, /video/ndFilter, /video/supportedNDFilters, /video/ndFilter/displayMode, /video/supportedNDFilterDisplayModes, /video/ndFilterSelectable, /video/shutter, /video/shutter/measurement, /video/supportedShutters, /video/flickerFreeShutters, /video/autoExposure, /video/detailSharpening, /video/detailSharpeningLevel. Must match the pattern: *
.data.values	object	An object with property names as the key and a property value as json. Check the next section for the device properties and their return values.
.data.success	boolean	
.data.deviceProperties	array	

Name	Type	Description
.data.deviceProperties[i]	string	device property that the user can subscribe to. The user can either choose a value from the predefined enum, or provide a wildcard string. Possible values are: /media/workingset, /media/active, /system, /system/codecFormat, /system/videoFormat, /system/format, /system/supportedFormats, /timelines/0, /transports/0, /transports/0/stop, /transports/0/play, /transports/0/playback, /transports/0/record, /transports/0/timecode, /transports/0/timecode/source, /transports/0/clipIndex, /slates/nextClip, /monitoring/{displayName}/cleanFeed, /monitoring/{displayName}/displayLUT, /monitoring/{displayName}/zebra, /monitoring/{displayName}/focusAssist, /monitoring/{displayName}/frameGuide, /monitoring/{displayName}/frameGrids, /monitoring/{displayName}/safeArea, /monitoring/{displayName}/falseColor, /monitoring/focusAssist, /monitoring/frameGuideRatio, /monitoring/frameGrids, /monitoring/safeAreaPercent, /audio/channel/{channelIndex}/input, /audio/channel/{channelIndex}/supportedInputs, /audio/channel/{channelIndex}/level, /audio/channel/{channelIndex}/phantomPower, /audio/channel/{channelIndex}/padding, /audio/channel/{channelIndex}/lowCutFilter, /audio/channel/{channelIndex}/available, /audio/channel/{channelIndex}/input/description, /colorCorrection/lift, /colorCorrection/gamma, /colorCorrection/gain, /colorCorrection/offset, /colorCorrection/contrast, /colorCorrection/color, /colorCorrection/lumaContribution, /lens/iris, /lens/iris/description, /lens/focus, /lens/focus/description, /lens/zoom, /lens/zoom/description, /presets, /presets/active, /camera/colorBars, /camera/programFeedDisplay, /camera/tallyStatus, /camera/power, /camera/power/displayMode, /camera/timingReferenceLock, /video/iso, /video/supportedISOs, /video/gain, /video/supportedGains, /video/whiteBalance, /video/whiteBalance/description, /video/whiteBalanceTint, /video/whiteBalanceTint/description, /video/ndFilter, /video/supportedNDFilters, /video/ndFilter/displayMode, /video/supportedNDFilterDisplayModes, /video/ndFilterSelectable, /video/shutter, /video/shutter/measurement, /video/supportedShutters, /video/flickerFreeShutters, /video/autoExposure, /video/detailSharpening, /video/detailSharpeningLevel. Must match the pattern: *
.type	string	Possible values are: response.
.id	number	Optional parameter that repeats the id in the output for tracking messages.

Device Properties

/media/workingset

The value JSON returned via the eventResponse when the `/media/workingset` property changes on the device:

Name	Type	Description
.size	integer	The fixed size of this device's working set.
.workingset	array	Array of devices within the working set. null if no device is present within the given working set slot.
.workingset[i]		

/media/active

The value JSON returned via the eventResponse when the `/media/active` property changes on the device:

Name	Type	Description
.workingsetIndex	integer	Working set index of the active media device.
.deviceName	string	Internal device name of this media device.

/system

The value JSON returned via the eventResponse when the `/system` property changes on the device:

Name	Type	Description
.codecFormat	object	Codec format configuration.
.codecFormat.codec	string	Codec format serialised as a string.
.codecFormat.container	string	Multimedia container format.
.videoFormat	object	Currently selected video format.
.videoFormat.frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
.videoFormat.height	number	Height dimension of video format.
.videoFormat.width	number	Width dimension of video format.
.videoFormat.interlaced	boolean	Is the display format interlaced?.
.videoFormat.name	string	Video format serialised as a string.

/system/codecFormat

Codec format configuration.

The value JSON returned via the eventResponse when the `/system/codecFormat` property changes on the device:

Name	Type	Description
.codec	string	Codec format serialised as a string.
.container	string	Multimedia container format.

/system/videoFormat

Currently selected video format.

The value JSON returned via the eventResponse when the `/system/videoFormat` property changes on the device:

Name	Type	Description
.frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
.height	number	Height dimension of video format.
.width	number	Width dimension of video format.
.interlaced	boolean	Is the display format interlaced?.
.name	string	Video format serialised as a string.

/system/format

The value JSON returned via the eventResponse when the `/system/format` property changes on the device:

Name	Type	Description
.codec	string	Codec format serialised as a string.
.frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
.maxOffSpeedFrameRate	number	
.minOffSpeedFrameRate	number	
.offSpeedEnabled	boolean	
.offSpeedFrameRate	number	
.recordResolution	object	
.recordResolution.height	number	Height of the resolution.
.recordResolution.width	number	Width of the resolution.
.sensorResolution	object	
.sensorResolution.height	number	Height of the resolution.
.sensorResolution.width	number	Width of the resolution.

/system/supportedFormats

The value JSON returned via the eventResponse when the `/system/supportedFormats` property changes on the device:

Name	Type	Description
.supportedFormats	array	
.supportedFormats[i]	object	
.supportedFormats[i].codecs	array	
.supportedFormats[i].codecs[i]	string	
.supportedFormats[i].frameRates	array	
.supportedFormats[i].frameRates[i]	string	Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
.supportedFormats[i].maxOffSpeedFrameRate	number	
.supportedFormats[i].minOffSpeedFrameRate	number	
.supportedFormats[i].recordResolution	object	
.supportedFormats[i].recordResolution.height	number	Height of the resolution.
.supportedFormats[i].recordResolution.width	number	Width of the resolution.
.supportedFormats[i].sensorResolution	object	
.supportedFormats[i].sensorResolution.height	number	Height of the resolution.
.supportedFormats[i].sensorResolution.width	number	Width of the resolution.

/timelines/0

The value JSON returned via the eventResponse when the `/timelines/0` property changes on the device:

Name	Type	Description
.clips	array	
.clips[i]	object	Timeline clip.
.clips[i].clipUniqueld	integer	Unique identifier used to identify this media clip. If the same media clip is added to the timeline multiple times, each timeline clip has the same clipUniqueld
.clips[i].frameCount	integer	Duration of timeline clip in frames, the number of frames in this clip on the timeline.
.clips[i].durationTimecode	string	Duration of the timeline clip in timecode format serialised as string. This will differ to durationTimecode reported in /clips for this clipUniqueld if clipIn or frameCount was specified when adding this clip to the timeline.
.clips[i].clipIn	string	In frame offset for the clip on the timeline, where 0 is the first frame of the on-disk clip.
.clips[i].inTimecode	string	Clip timecode of the first frame of this timeline clip serialised as string (clip startTimecode + clipIn frames).

Name	Type	Description
.clips[i].timelineIn	string	Timeline position of the first frame of this clip, where 0 is the first frame of the timeline.
.clips[i].timelineInTimecode	string	Timeline timecode of the first frame of this timeline clip serialised as string.

/transports/0

The value JSON returned via the eventResponse when the `/transports/0` property changes on the device:

Name	Type	Description
.mode	string	Transport mode. Possible values are: InputPreview, InputRecord, Output.

/transports/0/stop

true when transport mode is InputPreview or when in Output mode and speed is 0.

The value JSON returned via the eventResponse when the `/transports/0/stop` property changes on the device:

Name	Type	Description
	boolean	true when transport mode is InputPreview or when in Output mode and speed is 0.

/transports/0/play

True when transport is in Output mode and speed is non-zero.

The value JSON returned via the eventResponse when the `/transports/0/play` property changes on the device:

Name	Type	Description
	boolean	True when transport is in Output mode and speed is non-zero.

/transports/0/playback

The value JSON returned via the eventResponse when the `/transports/0/playback` property changes on the device:

Name	Type	Description
.type	string	Possible values are: Play, Jog, Shuttle, Var.
.loop	boolean	When true playback loops from the end of the timeline to the beginning of the timeline.
.singleClip	boolean	When true playback loops from the end of the current clip to the beginning of the current clip.
.speed	number	Playback speed, 1.0 for normal forward playback
.position	integer	Playback position on the timeline in units of video frames.

/transports/0/record

The value JSON returned via the eventResponse when the `/transports/0/record` property changes on the device:

Name	Type	Description
.recording	boolean	Is transport in Input Record mode.

/transports/0/timecode

The value JSON returned via the eventResponse when the `/transports/0/timecode` property changes on the device:

Name	Type	Description
.display	string	The display timecode serialised as a string.
.timeline	string	The timeline timecode serialised as a string.

/transports/0/timecode/source

The value JSON returned via the eventResponse when the `/transports/0/timecode/source` property changes on the device:

Name	Type	Description
.timecode	string	Possible values are: Timeline, Clip.

/transports/0/clipIndex

The value JSON returned via the eventResponse when the `/transports/0/clipIndex` property changes on the device:

Name	Type	Description
.clipIndex	number null	The 0-based index of the clip being played on the timeline. null when there is no timeline or an empty timeline.

/slates/nextClip

The value JSON returned via the eventResponse when the `/slates/nextClip` property changes on the device:

Name	Type	Description
.clip	object	
.clip.clipName	string	
.clip.reel	integer	
.clip.scene	string	
.clip.sceneLocation	string	Possible values are: Interior, Exterior.
.clip.sceneTime	string	Possible values are: Day, Night.
.clip.shotType	string	Possible values are: None, WS, MS, MCU, CU, BCU, ECU.
.clip.take	integer	
.clip.takeType	string	Possible values are: None, PU, VFX, SER.
.clip.goodTake	boolean	
.lens	object	

Name	Type	Description
.lens.lensType	string	
.lens.iris	string	
.lens.focalLength	string	
.lens.distance	string	
.lens.filter	string	
.project	object	
.project.projectName	string	
.project.director	string	
.project.camera	string	
.project.cameraOperator	string	

/monitoring/{displayName}/cleanFeed

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/cleanFeed` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/{displayName}/displayLUT

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/displayLUT` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/{displayName}/zebra

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/zebra` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/{displayName}/focusAssist

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/focusAssist` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/{displayName}/frameGuide

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/frameGuide` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/{displayName}/frameGrids

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/frameGrids` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/{displayName}/safeArea

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/safeArea` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/{displayName}/falseColor

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/falseColor` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/focusAssist

The value JSON returned via the eventResponse when the `/monitoring/focusAssist` property changes on the device:

Name	Type	Description
.mode	string	Possible values are: Peak, ColoredLines.
.color	string	Possible values are: Red, Green, Blue, White, Black.
.intensity	integer	

/monitoring/frameGuideRatio

The value JSON returned via the eventResponse when the `/monitoring/frameGuideRatio` property changes on the device:

Name	Type	Description
.ratio	string	

/monitoring/frameGrids

The value JSON returned via the eventResponse when the `/monitoring/frameGrids` property changes on the device:

Name	Type	Description
.frameGrids	array	
.frameGrids[i]	string	Possible values are: Thirds, Crosshair, Dot, Horizon.

/monitoring/safeAreaPercent

The value JSON returned via the eventResponse when the `/monitoring/safeAreaPercent` property changes on the device:

Name	Type	Description
.percent	integer	Safe area coverage percentage.

/audio/channel/{channelIndex}/input

Get the audio input (source and type) for the selected channel

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/input` property changes on the device:

Name	Type	Description
.input	string	Audio input source and type

/audio/channel/{channelIndex}/supportedInputs

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/supportedInputs` property changes on the device:

Name	Type	Description
	array	
[i]	object	
[i].input	string	Input name
[i].available	boolean	Is the input available to be switched into from the current input for the selected channel

/audio/channel/{channelIndex}/level

Get the audio input level for the selected channel

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/level` property changes on the device:

Name	Type	Description
.gain	number	Gain value in dB
.normalised	number	Normalised level value between 0.0 and 1.0

`/audio/channel/{channelIndex}/phantomPower`

Get the audio input phantom power status for the selected channel

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/phantomPower` property changes on the device:

Name	Type	Description
<code>.enabled</code>	boolean	Phantom power enabled state

`/audio/channel/{channelIndex}/padding`

Get the audio input padding status for the selected channel

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/padding` property changes on the device:

Name	Type	Description
<code>.enabled</code>	boolean	Padding enabled state

`/audio/channel/{channelIndex}/lowCutFilter`

Get the audio input low cut filter status for the selected channel

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/lowCutFilter` property changes on the device:

Name	Type	Description
<code>.enabled</code>	boolean	Low cut filter enabled state

`/audio/channel/{channelIndex}/available`

Get the audio input's current availability for the selected channel. If unavailable, the source will be muted

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/available` property changes on the device:

Name	Type	Description
<code>.available</code>	boolean	Whether the input is currently available

`/audio/channel/{channelIndex}/input/description`

Description of the current input of the selected channel

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/input/description` property changes on the device:

Name	Type	Description
<code>.description</code>	object	
<code>.description.gainRange</code>	object	
<code>.description.gainRange.Min</code>	number	The minimum gain value in dB
<code>.description.gainRange.Max</code>	number	The maximum gain value in dB
<code>.description.capabilities</code>	object	
<code>.description.capabilities.PhantomPower</code>	boolean	Input supports setting of phantom power

Name	Type	Description
.description.capabilities.LowCutFilter	boolean	Input supports setting of low cut filter
.description.capabilities.Padding	object	
.description.capabilities.Padding.available	boolean	Input supports setting of padding
.description.capabilities.Padding.forced	boolean	Padding is forced to be set for the input
.description.capabilities.Padding.value	number	An object with property names as the key and a property value as json. Check the next section for the device properties and their return values.

/colorCorrection/lift

Get color correction lift

The value JSON returned via the eventResponse when the `/colorCorrection/lift` property changes on the device:

Name	Type	Description
.red	number	Red lift component
.green	number	Green lift component
.blue	number	Blue lift component
.luma	number	Luma lift component

/colorCorrection/gamma

Get color correction gamma

The value JSON returned via the eventResponse when the `/colorCorrection/gamma` property changes on the device:

Name	Type	Description
.red	number	Red gamma component
.green	number	Green gamma component
.blue	number	Blue gamma component
.luma	number	Luma gamma component

/colorCorrection/gain

Get color correction gain

The value JSON returned via the eventResponse when the `/colorCorrection/gain` property changes on the device:

Name	Type	Description
.red	number	Red gain component
.green	number	Green gain component
.blue	number	Blue gain component
.luma	number	Luma gain component

/colorCorrection/offset

Get color correction offset

The value JSON returned via the eventResponse when the `/colorCorrection/offset` property changes on the device:

Name	Type	Description
.red	number	Red offset component
.green	number	Green offset component
.blue	number	Blue offset component
.luma	number	Luma offset component

/colorCorrection/contrast

Get color correction contrast

The value JSON returned via the eventResponse when the `/colorCorrection/contrast` property changes on the device:

Name	Type	Description
.pivot	number	Contrast pivot point
.adjust	number	Contrast adjustment

/colorCorrection/color

Get color correction color properties

The value JSON returned via the eventResponse when the `/colorCorrection/color` property changes on the device:

Name	Type	Description
.hue	number	Color hue adjustment
.saturation	number	Color saturation adjustment

/colorCorrection/lumaContribution

Get color correction luma contribution

The value JSON returned via the eventResponse when the `/colorCorrection/lumaContribution` property changes on the device:

Name	Type	Description
.lumaContribution	number	Luma contribution value

/lens/iris

Get lens' aperture

The value JSON returned via the eventResponse when the `/lens/iris` property changes on the device:

Name	Type	Description
.continuousApertureAutoExposure	boolean	Is Aperture controlled by auto exposure
.apertureStop	number	Aperture stop value
.normalised	number	Normalised value
.apertureNumber	integer	Aperture number

/lens/iris/description

Get detailed description of lens' iris capabilities

The value JSON returned via the eventResponse when the `/lens/iris/description` property changes on the device:

Name	Type	Description
.controllable	boolean	If the iris can be controlled
.apertureStop	object	
.apertureStop.min	number	Minimum aperture stop
.apertureStop.max	number	Maximum aperture stop

/lens/focus

Get lens' focus

The value JSON returned via the eventResponse when the `/lens/focus` property changes on the device:

Name	Type	Description
.normalised	number	Normalised value

/lens/focus/description

Get detailed description of lens' focus capabilities

The value JSON returned via the eventResponse when the `/lens/focus/description` property changes on the device:

Name	Type	Description
.controllable	boolean	If the focus can be controlled
.focusDistance	object	
.focusDistance.adjustable	boolean	If focus distance is adjustable
.focusDistance.min	number	Minimum focus distance
.focusDistance.max	number	Maximum focus distance

/lens/zoom

Get lens' zoom

The value JSON returned via the eventResponse when the `/lens/zoom` property changes on the device:

Name	Type	Description
.focalLength	integer	Focal length in mm
.normalised	number	Normalised value

/lens/zoom/description

Get detailed description of lens' zoom capabilities

The value JSON returned via the eventResponse when the `/lens/zoom/description` property changes on the device:

Name	Type	Description
.controllable	boolean	If the zoom can be controlled
.focalLength	object	
.focalLength.adjustable	boolean	If focal length is adjustable
.focalLength.min	integer	Minimum focal length
.focalLength.max	integer	Maximum focal length

/presets

Get the list of the presets on the camera

The value JSON returned via the eventResponse when the `/presets` property changes on the device:

Name	Type	Description
.presets	array	List of the presets on the camera (.cset files)
.presets[i]	string	

/presets/active

Get the currently active preset on the camera

The value JSON returned via the eventResponse when the `/presets/active` property changes on the device:

Name	Type	Description
.preset	string	Name of the active preset (with .cset extension, or 'default')

/camera/colorBars

Get the status of color bars display

The value JSON returned via the eventResponse when the `/camera/colorBars` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the color bars are currently enabled

/camera/programFeedDisplay

Get the status of program feed display

The value JSON returned via the eventResponse when the `/camera/programFeedDisplay` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the program feed display is currently enabled

/camera/tallyStatus

Get the tally status of the camera

The value JSON returned via the eventResponse when the `/camera/tallyStatus` property changes on the device:

Name	Type	Description
.status	string	Current tally status of the camera Possible values are: None, Preview, Program.

/camera/power

Get the power status of the camera

The value JSON returned via the eventResponse when the `/camera/power` property changes on the device:

Name	Type	Description
.source	string	Current power source of the camera Possible values are: Battery, AC, Fiber, USB, POE.
.milliVolt	integer	Current voltage level in millivolts (rounded to nearest 100mV)
.batteries	array	
.batteries[i]	object	
.batteries[i].milliVolt	integer	Battery voltage in millivolts (rounded to nearest 100mV)
.batteries[i].chargeRemainingPercent	integer	Remaining battery charge percentage
.batteries[i].statusFlags	array	List of battery status flags
.batteries[i].statusFlags[i]	string	Possible values are: Unknown Battery Status, Battery Is Present, Battery Is Charging, Battery Percentage Is Low, Battery Voltage Is Low, Battery Is Critically Low, Charge Remaining Percentage Is Estimated, Battery Communications Is Active, Battery Is Connected.

/camera/power/displayMode

Get the power display mode of the camera

The value JSON returned via the eventResponse when the `/camera/power/displayMode` property changes on the device:

Name	Type	Description
.mode	string	Current power display mode Possible values are: Percentage, Voltage.

/camera/timingReferenceLock

Get the timing reference lock status

The value JSON returned via the eventResponse when the `/camera/timingReferenceLock` property changes on the device:

Name	Type	Description
.locked	boolean	Indicates if timing reference is locked

/video/iso

Get current ISO

The value JSON returned via the eventResponse when the `/video/iso` property changes on the device:

Name	Type	Description
.iso	integer	Current ISO value

/video/supportedISOs

Get the list of supported ISO settings

The value JSON returned via the eventResponse when the `/video/supportedISOs` property changes on the device:

Name	Type	Description
.supportedISOs	array	Array of supported ISO values
.supportedISOs[i]	integer	

/video/gain

Get current gain value in decibels

The value JSON returned via the eventResponse when the `/video/gain` property changes on the device:

Name	Type	Description
.gain	integer	Current gain value in decibels

/video/supportedGains

Get the list of supported gain settings in decibels

The value JSON returned via the eventResponse when the `/video/supportedGains` property changes on the device:

Name	Type	Description
.supportedGains	array	Array of supported gain values in decibels
.supportedGains[i]	integer	

/video/whiteBalance

Get current white balance

The value JSON returned via the eventResponse when the `/video/whiteBalance` property changes on the device:

Name	Type	Description
.whiteBalance	integer	Current white balance

/video/whiteBalance/description

Get white balance range

The value JSON returned via the eventResponse when the `/video/whiteBalance/description` property changes on the device:

Name	Type	Description
.whiteBalance	object	
.whiteBalance.min	integer	Minimum color temperature
.whiteBalance.max	integer	Maximum color temperature

/video/whiteBalanceTint

Get white balance tint

The value JSON returned via the eventResponse when the `/video/whiteBalanceTint` property changes on the device:

Name	Type	Description
.whiteBalanceTint	integer	Current white balance tint

/video/whiteBalanceTint/description

Get white balance tint range

The value JSON returned via the eventResponse when the `/video/whiteBalanceTint/description` property changes on the device:

Name	Type	Description
.whiteBalanceTint	object	
.whiteBalanceTint.min	integer	Minimum white balance tint
.whiteBalanceTint.max	integer	Maximum white balance tint

/video/ndFilter

Get ND filter stop

The value JSON returned via the eventResponse when the `/video/ndFilter` property changes on the device:

Name	Type	Description
.stop	number	Current filter power (fStop)

/video/supportedNDFilters

Get the list of available ND filter stops

The value JSON returned via the eventResponse when the `/video/supportedNDFilters` property changes on the device:

Name	Type	Description
.supportedStops	array	Array of available ND filter stops
.supportedStops[i]	number	

/video/ndFilter/displayMode

Get ND filter display mode on the camera

The value JSON returned via the eventResponse when the `/video/ndFilter/displayMode` property changes on the device:

Name	Type	Description
.displayMode	string	ND filter display mode Possible values are: Stop, Number, Fraction.

/video/supportedNDFilterDisplayModes

Get the list of supported ND filter display modes

The value JSON returned via the eventResponse when the `/video/supportedNDFilterDisplayModes` property changes on the device:

Name	Type	Description
.supportedDisplayModes	array	Array of supported display modes
.supportedDisplayModes[i]	string	Possible values are: Stop, Number, Fraction.

/video/ndFilterSelectable

Check if ND filter adjustments are selectable via a slider

The value JSON returned via the eventResponse when the `/video/ndFilterSelectable` property changes on the device:

Name	Type	Description
.selectable	boolean	True if ND filter adjustments are selectable via a slider

/video/shutter

Get current shutter. Will return either shutter speed or shutter angle depending on shutter measurement in device settings

The value JSON returned via the eventResponse when the `/video/shutter` property changes on the device:

Name	Type	Description
.continuousShutterAutoExposure	boolean	Is shutter controlled by auto exposure
.shutterSpeed	integer	Shutter speed value in fractions of a second (minimum is sensor frame rate)
.shutterAngle	number	Shutter angle

/video/shutter/measurement

Get the current shutter measurement mode

The value JSON returned via the eventResponse when the `/video/shutter/measurement` property changes on the device:

Name	Type	Description
.measurement	string	Current shutter measurement mode Possible values are: ShutterAngle, ShutterSpeed.

/video/supportedShutters

Get supported shutter settings based on current camera configuration

The value JSON returned via the eventResponse when the `/video/supportedShutters` property changes on the device:

Name	Type	Description
.shutterAngles	array	Array of supported shutter angles
.shutterAngles[i]	number	
.shutterSpeeds	array	Array of flicker-free shutter speeds
.shutterSpeeds[i]	integer	

/video/flickerFreeShutters

Get flicker-free shutter settings based on current camera configuration

The value JSON returned via the eventResponse when the `/video/flickerFreeShutters` property changes on the device:

Name	Type	Description
.shutterAngles	array	Array of flicker-free shutter angles
.shutterAngles[i]	number	
.shutterSpeeds	array	Array
.shutterSpeeds[i]	integer	

/video/autoExposure

Get current auto exposure mode

The value JSON returned via the eventResponse when the `/video/autoExposure` property changes on the device:

Name	Type	Description
.mode	string	Auto exposure mode Possible values are: Off, Continuous, OneShot.
.type	string	Comma-separated list of device types in the auto exposure stack

/video/detailSharpening

Get the current state of detail sharpening

The value JSON returned via the eventResponse when the `/video/detailSharpening` property changes on the device:

Name	Type	Description
.enabled	boolean	Whether detail sharpening is enabled

/video/detailSharpeningLevel

Get the current detail sharpening level

The value JSON returned via the eventResponse when the `/video/detailSharpeningLevel` property changes on the device:

Name	Type	Description
.level	string	Current detail sharpening level of supported shutter speeds Possible values are: Low, Medium, High.

Blackmagic SDI Camera Control Protocol

Version 1.6.2

If you are a software developer you can use the Blackmagic SDI to construct devices that integrate with our products. Here at Blackmagic Design, our approach is to open up our protocols and we eagerly look forward to seeing what you come up with!

Overview

This document describes an extensible protocol for sending a unidirectional stream of small control messages embedded in the non-active picture region of a digital video stream. The video stream containing the protocol stream may be broadcast to a number of devices. Device addressing is used to allow the sender to specify which device each message is directed to.

Assumptions

Alignment and padding constraints are explicitly described in the protocol document. Bit fields are packed from LSB first. Message groups, individual messages and command headers are defined as, and can be assumed to be, 32 bit aligned.

Blanking Encoding

A message group is encoded into a SMPTE 291M packet with DID/SDID x51/x53 in the active region of VANC line 16.

Message Grouping

Up to 32 messages may be concatenated and transmitted in one blanking packet up to a maximum of 255 bytes payload. Under most circumstances, this should allow all messages to be sent with a maximum of one frame latency.

If the transmitting device queues more bytes of message packets than can be sent in a single frame, it should use heuristics to determine which packets to prioritize and send immediately. Lower priority messages can be delayed to later frames, or dropped entirely as appropriate.

Abstract Message Packet Format

Every message packet consists of a three byte header followed by an optional variable length data block. The maximum packet size is 64 bytes.

Destination device (uint8)	Device addresses are represented as an 8 bit unsigned integer. Individual devices are numbered 0 through 254 with the value 255 reserved to indicate a broadcast message to all devices.
Command length (uint8)	The command length is an 8 bit unsigned integer which specifies the length of the included command data. The length does NOT include the length of the header or any trailing padding bytes.
Command id (uint8)	The command id is an 8 bit unsigned integer which indicates the message type being sent. Receiving devices should ignore any commands that they do not understand. Commands 0 through 127 are reserved for commands that apply to multiple types of devices. Commands 128 through 255 are device specific.
Reserved (uint8)	This byte is reserved for alignment and expansion purposes. It should be set to zero.

Command data (uint8[])	The command data may contain between 0 and 60 bytes of data. The format of the data section is defined by the command itself.
Padding (uint8[])	Messages must be padded up to a 32 bit boundary with 0x0 bytes. Any padding bytes are NOT included in the command length.

Receiving devices should use the destination device address and or the command identifier to determine which messages to process. The receiver should use the command length to skip irrelevant or unknown commands and should be careful to skip the implicit padding as well.

Defined Commands

Command 0 : change configuration

Category (uint8)	The category number specifies one of up to 256 configuration categories available on the device.
Parameter (uint8)	The parameter number specifies one of 256 potential configuration parameters available on the device. Parameters 0 through 127 are device specific parameters. Parameters 128 though 255 are reserved for parameters that apply to multiple types of devices.
Data type (uint8)	The data type specifies the type of the remaining data. The packet length is used to determine the number of elements in the message. Each message must contain an integral number of data elements.

Currently defined values are:

0: void/boolean	A void value is represented as a boolean array of length zero. The data field is a 8 bit value with 0 meaning false and all other values meaning true.
1: signed byte	Data elements are signed bytes
2: signed 16 bit integer	Data elements are signed 16 bit values
3: signed 32 bit integer	Data elements are signed 32 bit values
4: signed 64 bit integer	Data elements are signed 64 bit values
5: UTF-8 string	Data elements represent a UTF-8 string with no terminating character.

Data types 6 through 127 are reserved.

128: signed 5.11 fixed point	Data elements are signed 16 bit integers representing a real number with 5 bits for the integer component and 11 bits for the fractional component. The fixed point representation is equal to the real value multiplied by 2^{11} . The representable range is from -16.0 to 15.9995 ($15 + 2047/2048$).
-------------------------------------	---

Data types 129 through 255 are available for device specific purposes.

Operation type (uint8)	The operation type specifies what action to perform on the specified parameter. Currently defined values are:
0: assign value	The supplied values are assigned to the specified parameter. Each element will be clamped according to its valid range. A void parameter may only be 'assigned' an empty list of boolean type. This operation will trigger the action associated with that parameter. A boolean value may be assigned the value zero for false, and any other value for true.
1: offset/toggle value	Each value specifies signed offsets of the same type to be added to the current parameter values. The resulting parameter value will be clamped according to their valid range. It is not valid to apply an offset to a void value. Applying any offset other than zero to a boolean value will invert that value.
Operation types 2 through 127 are reserved.	

Operation types 128 through 255 are available for device specific purposes.

Data (void)	The data field is 0 or more bytes as determined by the data type and number of elements.
--------------------	--

The category, parameter, data type and operation type partition a 24 bit operation space.

Group	ID	Parameter	Type	Index	Minimum	Maximum	Interpretation
Lens	0.0	Focus	fixed16	–	0.0	1.0	0.0 = near, 1.0 = far
	0.1	Instantaneous autofocus	void	–	–	–	trigger instantaneous autofocus
	0.2	Aperture (f-stop)	fixed16	–	-1.0	16.0	Aperture Value (where fnumber = $\sqrt{2^{AV}}$)
	0.3	Aperture (normalised)	fixed16	–	0.0	1.0	0.0 = smallest, 1.0 = largest
	0.4	Aperture (ordinal)	int16	–	0	n	Steps through available aperture values from minimum (0) to maximum (n)
	0.5	Instantaneous auto aperture	void	–	–	–	trigger instantaneous auto aperture
	0.6	Optical image stabilisation	boolean	–	–	–	true = enabled, false = disabled
	0.7	Set absolute zoom (mm)	int16	–	0	max	Move to specified focal length in mm, from minimum (0) to maximum (max)
	0.8	Set absolute zoom (normalised)	fixed16	–	0.0	1.0	Move to specified focal length: 0.0 = wide, 1.0 = tele
	0.9	Set continuous zoom (speed)	fixed16	–	-1.0	+1.0	Start/stop zooming at specified rate: -1.0 = zoom wider fast, 0.0 = stop, +1 = zoom tele fast

Group	ID	Parameter	Type	Index	Minimum	Maximum	Interpretation
Video	1.0	Video mode	int8	[0] = frame rate	–	–	fps as integer (eg 24, 25, 30, 50, 60)
				[1] = M-rate	–	–	0 = regular, 1 = M-rate
				[2] = dimensions	–	–	0 = NTSC, 1 = PAL, 2 = 720, 3 = 1080, 4 = 2kDCI, 5 = 2k16:9, 6 = UHD, 7 = 3k Anamorphic, 8 = 4k DCI, 9 = 4k 16:9, 10 = 4.6k 2.4:1, 11 = 4.6k
				[3] = interlaced	–	–	0 = progressive, 1 = interlaced
				[4] = Color space	–	–	0 = YUV
	1.1	Gain (up to Camera 4.9)	int8		1	128	1x, 2x, 4x, 8x, 16x, 32x, 64x, 128x gain
	1.2	Manual White Balance	int16	[0] = color temp	2500	10000	Color temperature in K
			int16	[1] = tint	-50	50	tint
	1.3	Set auto WB	void	–	–	–	Calculate and set auto white balance
	1.4	Restore auto WB	void	–	–	–	Use latest auto white balance setting
	1.5	Exposure (us)	int32		1	42000	time in us
	1.6	Exposure (ordinal)	int16	–	0	n	Steps through available exposure values from minimum (0) to maximum (n)
	1.7	Dynamic Range Mode	int8 enum	–	0	2	0 = film, 1 = video, 2 = extended video
	1.8	Video sharpening level	int8 enum	–	0	3	0 = off, 1 = low, 2 = medium, 3 = high
	1.9	Recording format	int16	[0] = file frame rate	–	–	fps as integer (eg 24, 25, 30, 50, 60, 120)
				[1] = sensor frame rate	–	–	fps as integer, valid when sensor-off-speed set (eg 24, 25, 30, 33, 48, 50, 60, 120), no change will be performed if this value is set to 0
				[2] = frame width	–	–	in pixels
				[3] = frame height	–	–	in pixels
				[4] = flags	–	–	[0] = file-M-rate
					–	–	[1] = sensor-M-rate, valid when sensor-off-speed-set
					–	–	[2] = sensor-off-speed
					–	–	[3] = interlaced
					–	–	[4] = windowed mode
	1.10	Set auto exposure mode	int8	–	0	4	0 = Manual Trigger, 1 = Iris, 2 = Shutter, 3 = Iris + Shutter, 4 = Shutter + Iris
	1.11	Shutter angle	int32	–	100	36000	Shutter angle in degrees, multiplied by 100
	1.12	Shutter speed	int32	–	Current sensor frame rate	5000	Shutter speed value as a fraction of 1, so 50 for 1/50th of a second
	1.13	Gain	int8	–	-128	127	Gain in decibel (dB)
	1.14	ISO	int32	–	0	2147483647	ISO value
	1.15	Display LUT	int8	[0] = selected LUT	–	–	0 = None, 1 = Custom, 2 = film to video, 3 = film to extended video
				[1] = enabled or not	–	–	0 = Not enabled, 1 = Enabled

Group	ID	Parameter	Type	Index	Minimum	Maximum	Interpretation
	1.16	ND Filter Stop	fixed16	[0] = stop	0.0	15.0	filter power, as f-stop
				[1] = display mode	–	–	0 = stop 1 = density 2 = transmittance
Audio	2.0	Mic level	fixed16	–	0.0	1.0	0.0 = minimum, 1.0 = maximum
	2.1	Headphone level	fixed16	–	0.1	1.0	0.0 = minimum, 1.0 = maximum
	2.2	Headphone program mix	fixed16	–	0.1	1.0	0.0 = minimum, 1.0 = maximum
	2.3	Speaker level	fixed16	–	0.1	1.0	0.0 = minimum, 1.0 = maximum
	2.4	Input type	int8	–	0	3	0 = internal mic, 1 = line level input, 2 = low mic level input, 3 = high mic level input
	2.5	Input levels	fixed16	[0] ch0	0.0	1.0	0.0 = minimum, 1.0 = maximum
				[1] ch1	0.0	1.0	0.0 = minimum, 1.0 = maximum
	2.6	Phantom power	boolean	–	–	–	true = powered, false = not powered
Output	3.0	Overlay enables	uint16 bit field	[0] = bit field	–	–	bit flags: [0] = display status, [1] = display frame guides [2] = clean feed Some cameras don't allow separate control of frame guides and status overlays.
			uint16 bit field	[1] = target displays bit field	–	–	bit flags: [0] = LCD [1] = HDMI [2] = EVF [3] = Main SDI [4] = Front SDI
	3.1	Frame guides style (Camera 3.x)	int8	–	0	8	0 = HDTV, 1 = 4:3, 2 = 2.4:1, 3 = 2.39:1, 4 = 2.35:1, 5 = 1.85:1, 6 = thirds
	3.2	Frame guides opacity (Camera 3.x)	fixed16	–	0.1	1.0	0.0 = transparent, 1.0 = opaque
	3.3	Overlays (replaces .1 and .2 above from Cameras 4.0)	int8	[0] = frame guides style	–	–	0 = off, 1 = 2.4:1, 2 = 2.39:1, 3 = 2.35:1, 4 = 1.85:1, 5 = 16:9, 6 = 14:9, 7 = 4:3, 8 = 2:1, 9 = 4:5, 10 = 1:1
				[1] = frame guide opacity	0	100	0 = transparent, 100 = opaque
				[2] = safe area percentage	0	100	percentage of full frame used by safe area guide (0 means off)
				[3] = grid style	–	–	bit flags: [0] = display thirds, [1] = display cross hairs, [2] = display center dot, [3] = display horizon

Group	ID	Parameter	Type	Index	Minimum	Maximum	Interpretation
Display	4.0	Brightness	fixed16	–	0.0	1.0	0.0 = minimum, 1.0 = maximum
	4.1	Exposure and focus tools	uint16 bit field	[0] = bit field	–	–	bit flags: [0] = Zebra [1] = Focus Assist [2] = False Color
			uint16 bit field	[1] = target displays bit field	–	–	bit flags: [0] = LCD [1] = HDMI [2] = EVF [3] = Main SDI [4] = Front SDI
	4.2	Zebra level	fixed16	–	0.0	1.0	0.0 = minimum, 1.0 = maximum
	4.3	Peaking level	fixed16	–	0.0	1.0	0.0 = minimum, 1.0 = maximum
	4.4	Color bar enable	int8	–	0	30	0 = disable bars, 1-30 = enable bars with timeout (seconds)
	4.5	Focus Assist	int8	[0] = focus assist method	–	–	0 = Peak, 1 = Colored lines
				[1] = focus line color	–	–	0 = Red, 1 = Green, 2 = Blue, 3 = White, 4 = Black
	4.6	Program return feed enable	int8	–	0	30	0 = disable, 1-30 = enable with timeout (seconds)
	4.7	Timecode Source	signed byte	[0] = source	–	–	0 = Clip, 1 = Timecode
Tally	5.0	Tally brightness	fixed16	–	0.0	1.0	Sets the tally front and tally rear brightness to the same level. 0.0 = minimum, 1.0 = maximum
	5.1	Front tally brightness	fixed16	–	0.0	1.0	Sets the tally front brightness. 0.0 = minimum, 1.0 = maximum
	5.2	Rear tally brightness	fixed16	–	0.0	1.0	Sets the tally rear brightness. 0.0 = minimum, 1.0 = maximum Tally rear brightness cannot be turned off
Reference	6.0	Source	int8 enum	–	0	2	0 = internal, 1 = program, 2 = external
	6.1	Offset	int32	–	–	–	+/- offset in pixels

Group	ID	Parameter	Type	Index	Minimum	Maximum	Interpretation
Confi- guration	7.0	Real Time Clock	int32	[0] time	—	—	BCD - HHMMSSFF (UCT)
				[1] date	—	—	BCD - YYYYMMDD
	7.1	System language	string	[0-1]	—	—	ISO-639-1 two character language code
	7.2	Timezone	int32	—	—	—	Minutes offset from UTC
	7.3	Location	int64	[0] latitude	—	—	BCD - s0DDdddddddddd where s is the sign: 0 = north (+), 1 = south (-); DD degrees, ddddddddddd decimal degrees
				[1] longitude	—	—	BCD - sDDDDdddddddddd where s is the sign: 0 = west (-), 1 = east (+); DDD degrees, ddddddddddd decimal degrees
Color Correction	8.0	Lift Adjust	fixed16	[0] red	-2.0	2.0	default 0.0
				[1] green	-2.0	2.0	default 0.0
				[2] blue	-2.0	2.0	default 0.0
				[3] luma	-2.0	2.0	default 0.0
	8.1	Gamma Adjust	fixed16	[0] red	-4.0	4.0	default 0.0
				[1] green	-4.0	4.0	default 0.0
				[2] blue	-4.0	4.0	default 0.0
				[3] luma	-4.0	4.0	default 0.0
	8.2	Gain Adjust	fixed16	[0] red	0.0	16.0	default 1.0
				[1] green	0.0	16.0	default 1.0
				[2] blue	0.0	16.0	default 1.0
				[3] luma	0.0	16.0	default 1.0
	8.3	Offset Adjust	fixed16	[0] red	-8.0	8.0	default 0.0
				[1] green	-8.0	8.0	default 0.0
				[2] blue	-8.0	8.0	default 0.0
				[3] luma	-8.0	8.0	default 0.0
	8.4	Contrast Adjust	fixed16	[0] pivot	0.0	1.0	default 0.5
				[1] adj	0.0	2.0	default 1.0
	8.5	Luma mix	fixed16	—	0.0	1.0	default 1.0
	8.6	Color Adjust	fixed16	[0] hue	-1.0	1.0	default 0.0
				[1] sat	0.0	2.0	default 1.0
	8.7	Correction Reset Default	void	—	—	—	reset to defaults

Group	ID	Parameter	Type	Index	Minimum	Maximum	Interpretation
Media	10.0	Codec	int8 enum	[0] = basic codec	–	–	0 = CinemaDNG, 1 = DNxHD, 2 = ProRes, 3 = Blackmagic RAW
				[1] = code variant	–	–	CinemaDNG: 0 = uncompressed, 1 = lossy 3:1, 2 = lossy 4:1
					–	–	ProRes: 0 = HQ, 1 = 422, 2 = LT, 3 = Proxy, 4 = 444, 5 = 444XQ
					–	–	Blackmagic RAW: 0 = Q0, 7 = Q1, 8 = Q3, 1 = Q5, 2 = 3:1, 3 = 5:1, 4 = 8:1, 5 = 12:1
	10.1	Transport mode	int8	[0] = mode	–	–	0 = Preview, 1 = Play, 2 = Record
				[1] = speed	–	–	-ve = multiple speeds backwards, 0 = pause, +ve = multiple speeds forwards
				[2] = flags	–	–	1<<0 = loop, 1<<1 = play all, 1<<5 = disk1 active, 1<<6 = disk2 active, 1<<7 = time-lapse recording
				[3] = slot 1 storage medium	–	–	0 = CFast card, 1 = SD, 2 = SSD Recorder
				[4] = slot 2 storage medium	–	–	0 = CFast card, 1 = SD, 2 = SSD Recorder
	10.2	Playback Control	int8 enum	[0] = clip	–	–	0 = Previous, 1 = Next
	10.5	Stream	bool	[0] = enabled	–	–	true = enabled, false = disabled
	10.6	Stream Information	void bool	[0] = enabled	–	–	true = enabled, false = disabled
	10.7	Stream Display 3D LUT	void bool	[0] = enabled	–	–	true = enabled, false = disabled

Group	ID	Parameter	Type	Index	Minimum	Maximum	Interpretation
PTZ Control	11.0	Pan/Tilt Velocity	fixed 16	[0] = pan velocity	-1.0	1.0	-1.0 = full speed left, 1.0 = full speed right
				[1] = tilt velocity	-1.0	1.0	-1.0 = full speed down, 1.0 = full speed up
	11.1	Memory Preset	int8 enum	[0] = preset command	–	–	0 = reset, 1 = store location, 2 = recall location
			int8	[1] = preset slot	0	5	–

Example Protocol Packets

Operation	Packet Length	Byte															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		header				command				data							
		destination	length	command	reserved	category	parameter	type	operation								
trigger instantaneous auto focus on camera 4	8	4	4	0	0	0	1	0	0								
turn on OIS on all cameras	12	255	5	0	0	0	6	0	0	1	0	0	0				
set exposure to 10 ms on camera 4 (10 ms = 10000 us = 0x00002710)	12	4	8	0	0	1	5	3	0	0x10	0x27	0x00	0x00				
add 15% to zebra level (15 % = 0.15 f = 0x0133 fp)	12	4	6	0	0	4	2	128	1	0x33	0x01	0	0				
select 1080p 23.98 mode on all cameras	16	255	9	0	0	1	0	1	0	24	1	3	0	0	0	0	0
subtract 0.3 from gamma adjust for green & blue (-0.3 ~ = 0xfd9a fp)	16	4	12	0	0	8	1	128	1	0	0	0x9a	0xfd	0x9a	0xfd	0	0
all operations combined	76	4	4	0	0	0	1	0	0	255	5	0	0	0	6	0	0
		1	0	0	0	4	8	0	0	1	5	3	0	0x10	0x27	0x00	0x00
		4	6	0	0	4	2	128	1	0x33	0x01	0	0	255	9	0	0
		1	0	1	0	24	1	3	0	0	0	0	0	4	12	0	0
		8	1	128	1	0	0	0x9a	0xfd	0x9a	0xfd	0	0				

Blackmagic Embedded Tally Control Protocol

Version 1.0 (30/04/14)

This section is for third party developers or anybody who may wish to add support for the Blackmagic Embedded Tally Control Protocol to their products or system. It describes the protocol for sending tally information embedded in the non-active picture region of a digital video stream.

Data Flow

A master device such as a broadcast switcher embeds tally information into its program feed which is broadcast to a number of slave devices such as cameras or camera controllers. The output from the slave devices is typically fed back to the master device, but may also be sent to a video monitor.

The primary flow of tally information is from the master device to the slaves. Each slave device may use its device id to extract and display the relevant tally information.

Slave devices pass through the tally packet on their output and update the monitor tally status, so that monitor devices connected to that individual output may display tally status without knowledge of the device id they are monitoring.

Assumptions

Any data alignment / padding is explicit in the protocol. Bit fields are packed from LSB first.

Blanking Encoding

One tally control packet may be sent per video frame. Packets are encoded as a SMPTE 291M packet with DID/SDID x51/x52 in the active region of VANC line 15. A tally control packet may contain up to 256 bytes of tally information.

Packet Format

Each tally status consists of 4 bits of information:

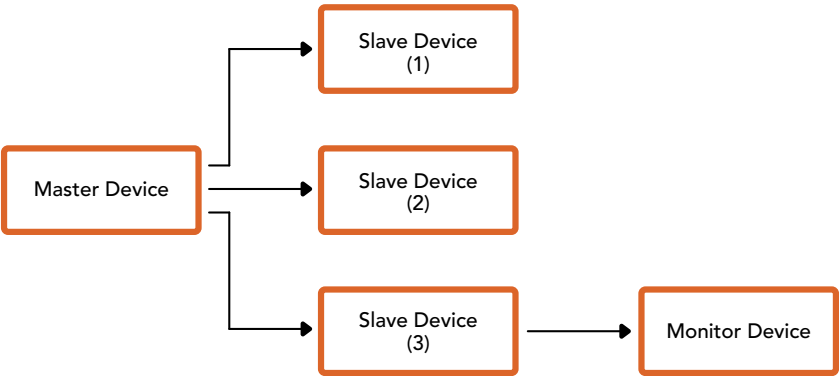
```
uint4
bit 0:    program tally status (0=off, 1=on)
bit 1:    preview tally status (0=off, 1=on)
bit 2-3:  reserved (0x0)
```

The first byte of the tally packet contains the monitor device tally status and a version number.

Subsequent bytes of the tally packet contain tally status for pairs of slave devices. The master device sends tally status for the number of devices configured/supported, up to a maximum of 510.

```
struct tally
uint8
bit 0:    monitor device program tally status (0=off, 1=on)
bit 1:    monitor device preview tally status (0=off, 1=on)
bit 2-3:  reserved (0b00)
bit 4-7:  protocol version (0b0000)
uint8[0]
bit 0:    slave device 1 program tally status (0=off, 1=on)
bit 1:    slave device 1 device preview tally status (0=off, 1=on)
bit 2-3:  reserved (0b00)
bit 4:    slave device 2 program tally status (0=off, 1=on)
bit 5:    slave device 2 preview tally status (0=off, 1=on)
bit 6-7:  reserved (0b00)
```

uint8[1]
 bit 0: slave device 3 program tally status (0=off, 1=on)
 bit 1: slave device 3 device preview tally status (0=off, 1=on)
 bit 2-3: reserved (0b00)
 bit 4: slave device 4 program tally status (0=off, 1=on)
 bit 5: slave device 4 preview tally status (0=off, 1=on)
 bit 6-7: reserved (0b00)
 ...



Byte	7 MSB	6	5	4	3	2	1	0 LSB
0	Version (0b0)	Version (0b0)	Version (0b0)	Version (0b0)	Reserved (0b0)	Reserved (0b0)	Monitor Preview	Monitor Program
1	Reserved (0b0)	Reserved (0b0)	Slave 1 Preview	Slave 1 Program	Reserved (0b0)	Reserved (0b0)	Slave 0 Preview	Slave 0 Program
2	Reserved (0b0)	Reserved (0b0)	Slave 3 Preview	Slave 3 Program	Reserved (0b0)	Reserved (0b0)	Slave 2 Preview	Slave 2 Program
3	...							

Visca Commands for PTZ control via SDI

Pan-tiltDrive	Up	8x 01 06 01 VV WW 03 01 FF	VV: Pan speed 01 to 18 WW: Tilt speed 01 to 17 YYYY: Pan position F725 to 08DB (center 0000) ZZZZ: Tilt position FE70 to 04B0 (image flip: OFF) (center 0000) Tilt position FB50 to 0190 (image flip: ON) (center 0000)
	Down	8x 01 06 01 VV WW 03 02 FF	
	Left	8x 01 06 01 VV WW 01 03 FF	
	Right	8x 01 06 01 VV WW 02 03 FF	
	UpLeft	8x 01 06 01 VV WW 01 01 FF	
	UpRight	8x 01 06 01 VV WW 02 01 FF	
	DownLeft	8x 01 06 01 VV WW 01 02 FF	
	DownRight	8x 01 06 01 VV WW 02 02 FF	
	Stop	8x 01 06 01 VV WW 03 03 FF	
	AbsolutePosition	8x 01 06 02 VV WW 0Y 0Y 0Y 0Y 0Z 0Z 0Z 0Z FF	
	RelativePosition	8x 01 06 03 VV WW 0Y 0Y 0Y 0Y 0Z 0Z 0Z 0Z FF	
	Home	0Y 0Y 0Y 0Y 0Z 0Z 0Z 0Z FF	
	Reset	8x 01 06 05 FF	
CAM_Memory	Reset	8x 01 04 3F 00 0p FF	p: Memory number (=0 to 5) Corresponds to 1 to 6 on the remote commander.
	Set	8x 01 04 3F 01 0p FF	
	Recall	8x 01 04 3F 02 0p FF	

Compatible motorized heads include the following:

- KXWell KT-PH180BMD
- PTZOptics PT-Broadcaster
 - RUSHWORKS PTX Model 1

Blackmagic Bluetooth Camera Control

Blackmagic cameras with Bluetooth LE implement a variety of features and commands that allow users to control their cameras wirelessly. Developers have full access to these features for their custom applications.

The following services and characteristics describe the full range of communication options that are available to the developer.

Service: Device Information Service

UUID: 180A

Characteristics

Camera Manufacturer

UUID: 2A29

Read the name of the manufacturer (always “Blackmagic Design”).

Camera Model

UUID: 2A24

Read the name of the camera model (eg. “URSA Mini Pro”).

Service: Blackmagic Camera Service

UUID: 291D567A-6D75-11E6-8B77-86F30CA893D3

Characteristics

Outgoing Camera Control (encrypted)

UUID: 5DD3465F-1AEE-4299-8493-D2ECA2F8E1BB

Send Camera Control messages.

These messages are identical to those described in the Blackmagic SDI Camera Control Protocol section above. Please read that section for a list of supported messages and required formatting information.

For an example of how packets are structured, please see the ‘example protocol packets’ section in this document.

Incoming Camera Control (encrypted)

UUID: B864E140-76A0-416A-BF30-5876504537D9

Request notifications for this characteristic to receive Camera Control messages from the camera.

These messages are identical to those described in the Blackmagic SDI Camera Control Protocol section above. Please read that section for a list of supported messages and required formatting information.

Timecode (encrypted)

UUID: 6D8F2110-86F1-41BF-9AFB-451D87E976C8

Request notifications for this characteristic to receive timecode updates.

Timecode (HH:MM:SS:mm) is represented by a 32-bit BCD number: (eg. 09:12:53:10 = 0x09125310)

Camera Status (encrypted)

UUID: 7FE8691D-95DC-4FC5-8ABD-CA74339B51B9

Request notifications for this characteristic to receive camera status updates.

The camera status is represented by flags contained in an 8-bit integer:

None	= 0x00
Camera Power On	= 0x01
Connected	= 0x02
Paired	= 0x04
Versions Verified	= 0x08
Initial Payload Received	= 0x10
Camera Ready	= 0x20

Send a value of 0x00 to power a connected camera off.

Send a value of 0x01 to power a connected camera on.

Device Name

UUID: FFAC0C52-C9FB-41A0-B063-CC76282EB89C

Send a device name to the camera (max. 32 characters).

The camera will display this name in the Bluetooth Setup Menu.

Protocol Version

UUID: 8F1FD018-B508-456F-8F82-3D392BEE2706

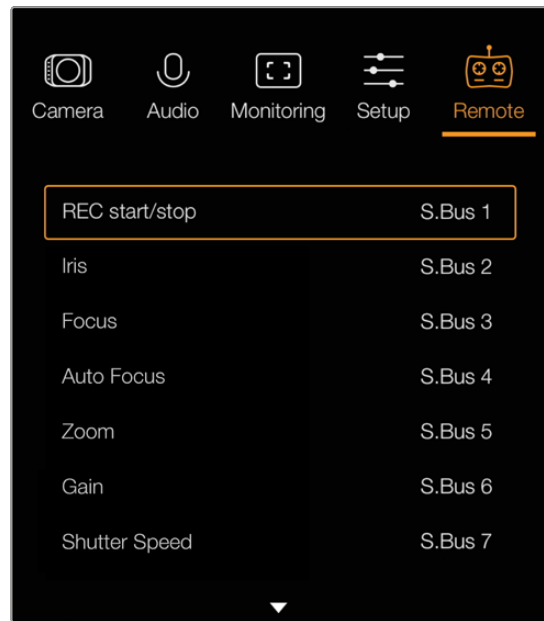
Read this value to determine the camera's supported CCU protocol version.

NOTE Encrypted characteristics can only be used once a device has successfully bonded or paired with the Blackmagic Camera. Once a connection has been established, any attempt to write to an encrypted characteristic will initiate bonding. For example, writing a 'Camera Power On' (0x01) message to the Camera Status characteristic.

Once bonding is initiated, the camera will display a 6-digit pin in the Bluetooth Setup Menu. Enter this pin on your device to establish an encrypted connection. The device will now be able to read, write and receive notifications from encrypted characteristics.

S.Bus

Blackmagic Micro Studio Camera 4K and Blackmagic Micro Cinema Camera feature a 'remote' settings menu for setting remote functions using the S.Bus protocol. This protocol uses 1 connection to control up to 17 channels, and each of these channels can be mapped to a specific camera function. S.Bus receivers and decoders can be found in most major hobby stores online as they are often used for radio remote control of airplane and helicopter models.



Remote settings – Blackmagic Micro Studio Camera 4K.

Assigning Camera Functions to S.Bus Channels

If you are using S.Bus to control your Blackmagic Micro Studio Camera 4K, you can use the 'remote' menu to assign the following functions to individual S.Bus channels:

- Trigger record
- Iris
- Focus
- Auto focus
- Zoom
- Gain
- Shutter speed
- White balance
- Audio levels
- Frame rate

If you are using S.Bus to control your Micro Cinema Camera, you can use the 'remote' menu to assign the following functions individual S.Bus channels:

- Trigger record
- Iris
- Focus
- Auto focus
- Zoom
- ISO
- Shutter angle
- White balance
- Audio levels
- Frame rate
- Codec

To assign functions to individual S.Bus channels, simply select the function you wish to control and assign an available channel using the 'up,' 'down' and 'set' buttons.

Standard radio transmitters for remote control vehicles that support the S.Bus protocol are usually setup with control ranges built into their controller output, so that all you need to do is assign camera functions to the correct individual S.Bus channels for remote control of your camera functions.

You can also use the S.Bus protocol to develop your own sophisticated custom control solutions.

Developing a Custom Controller

If you would like to develop your own custom camera control solutions, you can use the S.Bus input on the expansion cable as a way to interface camera functions on Blackmagic Micro Studio Camera 4K and Blackmagic Micro Cinema Camera.

When sending commands via the S.Bus input to your camera, the input values will need to be between 44 and 212 in order to be interpreted by the camera. A value of 128 is considered to be the midpoint or neutral position when using a radio control transmitter.

The way in which specific commands are sent to the camera will depend upon how you have mapped the camera functions to your controller.

There are two ways to map the commands to the controls.

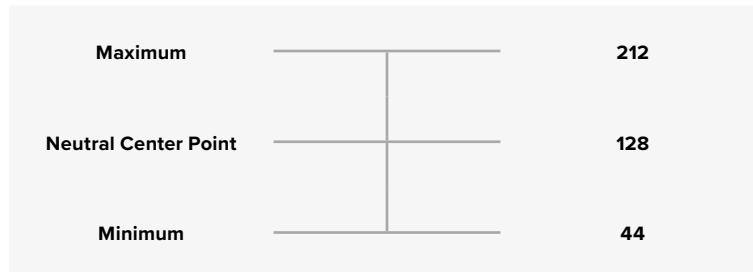
- The first type maps settings to specific ranges of the input so that sending a value within a certain range will trigger a particular setting.

For example, the f-stop settings on a lens from f1.8 to f22 will be distributed along the entire range of 44 to 212. Sending a value between 44 and 51 would set the lens to f1.8. These values will then continue along the entire range so that sending a value between 206 and 212 would select f22. Zoom and focus changes are controlled the same way.

f1.8	f2	f2.8	f4	f5.6	f8	f11	f16	f22
44–61	62–79	80–97	98–115	116–133	134–151	152–169	170–197	198–212

- The second type of control registers any change from the neutral value of 128 to a value above or below and then back to the neutral point. This will be considered by the camera as a valid toggle signal, which increases or decreases the assigned settings. Settings like the REC trigger, autofocus, gain/ISO, shutter value, white balance and frame rate work on this basis.

You could assign camera functions to a control like a spring loaded joystick which snaps back to a neutral center point after each movement up or down. In this example a value of 44 would represent the maximum downward position of the joystick and 212 would represent the maximum upward position, while the center functions as a neutral point with a value of 128.



For example, if your gain settings are mapped to a joystick in this way, then after each upward movement of the joystick it would return to the neutral point in the center which toggles the camera to increase gain by one increment, say from 0dB to 6dB.

You could also send this same information in numerical form to another type of controller that uses numerical values. In this case you would send a value of 128, followed by a value above 128 such as 212, and then back to 128 again. The camera will register this as an increment command and change the gain from 0dB to 6dB.

The way in which you assign commands will depend upon the kind of control system that you are using to control your camera and the type of control that you want to assign. Spring loaded controls that snap back to a neutral point are very common on radio control transmitters for model aircraft and drones.

If you are using a Futaba style remote control, some functions will be more suited to the rotating dials or analogue sticks, whilst other functions will be more suited to the switches.

PWM

The 'remote' settings menu also allows you to configure the PWM channels connected to the expansion port on your Blackmagic Micro Cinema Camera.

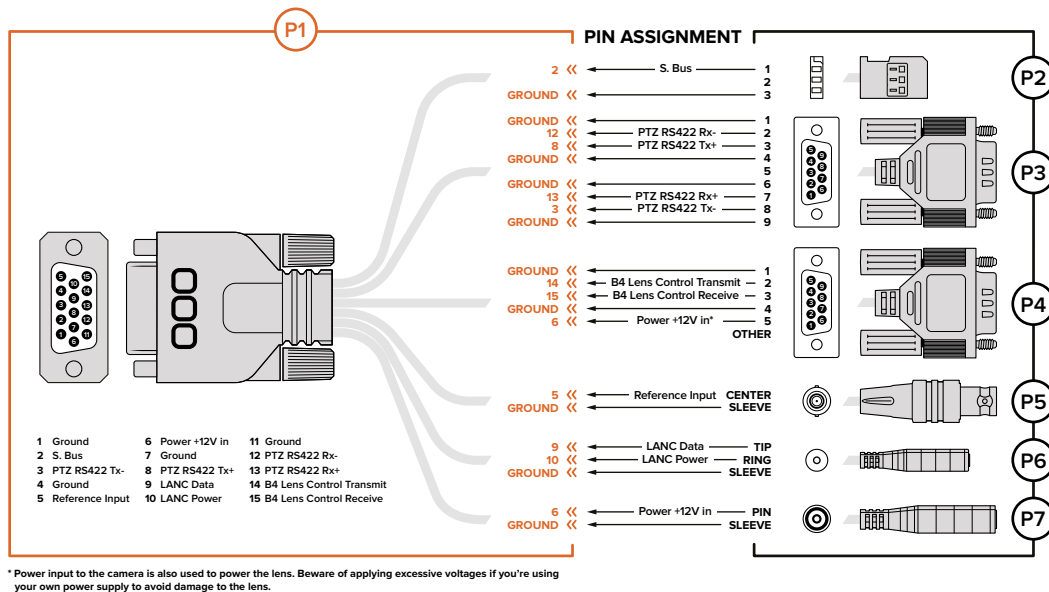
Analog radio control transmitters and receivers are typically used to drive servos on model airplanes, but can also be used for wireless remote control of the Blackmagic Micro Cinema Camera.

The 4 analog PWM radio remote control inputs on the expansion cable allow you to map camera functions to the controls on model airplane controllers. The dials, switches and joysticks on the radio controller are output on different radio channels and these channels can be assigned to four different camera functions in the remote settings menu. This provides you with a low cost, power efficient way to wirelessly control your camera. You can even generate your own PWM signals from your own Arduino or Raspberry Pi projects to control the camera.

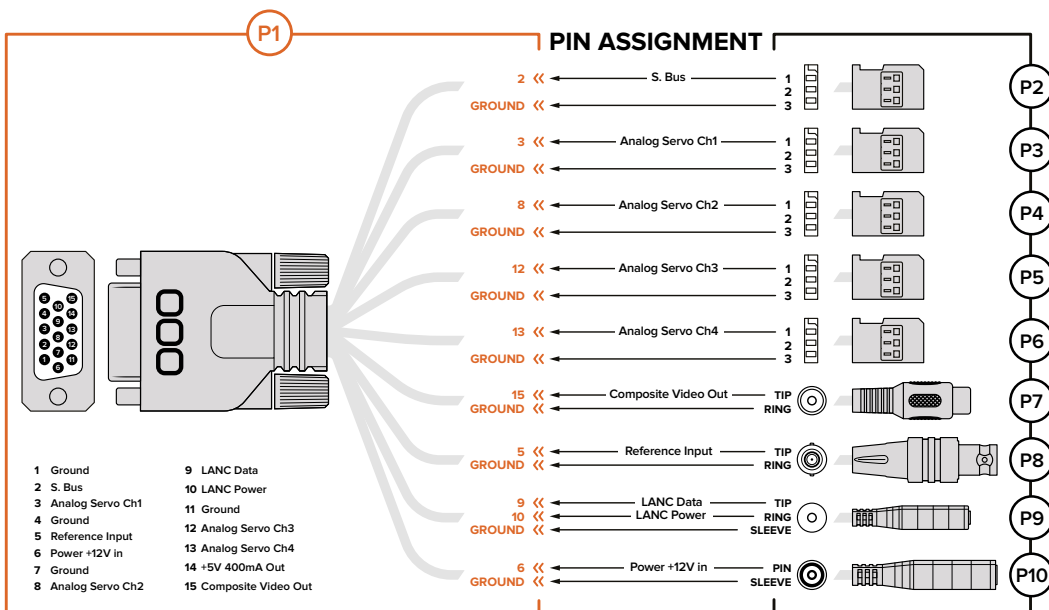
Wiring Diagrams

When using Blackmagic Micro Cinema Camera or Micro Studio Camera 4K's expansion port, you may only want to access one or two functions. For example, you may want to control an attached B4 Broadcast Lens while simultaneously receiving 12V power and a reference signal. It's easy to make a connector that will give you just these functions without the clutter of additional, unused connectors.

Use the following diagrams when wiring the expansion cables included or use it as an example of how you can wire up the connections on your own custom cable correctly. The full range of available pins are listed under group P1, while subsets used for particular functions, as well as their layout within the appropriate connectors, are shown in groups P2 and above.



Wiring diagram for the Blackmagic Micro Studio Camera 4K Expansion Cable



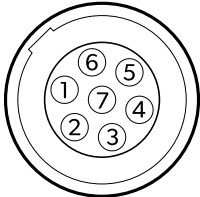
Wiring Diagram for the Blackmagic Micro Cinema Camera Expansion Cable

Pinout Table for URSA Cine Connector

The following pinout table for the 7 pin EXT connector.

The 7 pin connector on URSA Cine is compatible with Lemo style connectors and can be used for controlling URSA Cine with REST API commands via serial interface. For more information on this please contact Blackmagic developer support.

7 pin EXT connector

External View	Pins	Signal
 7 pin EXT connector EEG.1B.307	1	Serial 2 RX
	2	Serial 2 TX
	3	Serial 1 RX
	4	Serial 1 TX
	5	24V Output
	6	Ground
	7	Record Run/Stop