

Blackmagic Camera Control



Contents

Developer Information	3
Camera Control REST API	3
Event Control API	4
System Control API	4
Transport Control API	9
Timeline Control API	12
Media Control API	13
Preset Control API	15
Audio Control API	17
Lens Control API	22
Video Control API	24
Color Correction Control API	28
Blackmagic SDI Camera Control Protocol	33
Example Protocol Packets	42
Blackmagic Embedded Tally Control Protocol	43
Visca Commands for PTZ control via SDI	45
Blackmagic Bluetooth Camera Control	46
Service: Device Information Service	46
Service: Blackmagic Camera Service	46
S.Bus	48
PWM	50
Wiring Diagrams	51

Developer Information

Camera Control REST API

If you are a software 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.

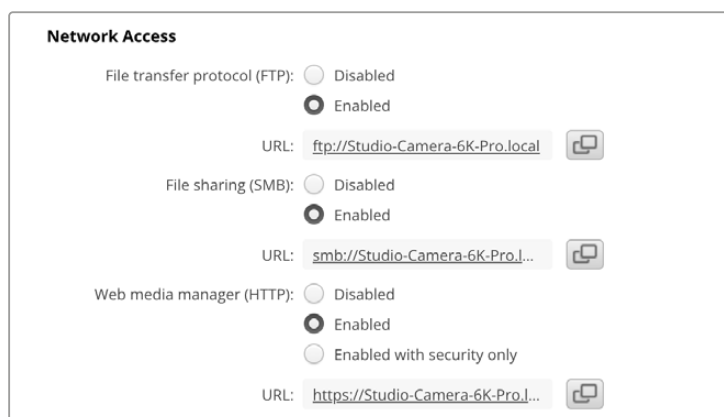
The following Blackmagic cameras are compatible with Camera Control REST API:

- Blackmagic 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://Studio-Camera-6K-Pro.local/control/api/v1/>

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



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://Studio-Camera-6K-Pro.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.

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 - OK

The response is a JSON object.

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

System Control API

API for controlling the System Modes on Blackmagic Design products.

GET /system

Get device system information.

Response

200 - OK

The response is a JSON object.

Name	Type	Description
codecFormat	object	
codecFormat.codec	string	Currently selected codec
codecFormat.container	string	Multimedia container format
videoFormat	object	
videoFormat.name	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/supportedCodecFormats

Get the list of supported codecs.

Response

200 - OK

The response is a JSON object.

Name	Type	Description
codecs	array	
codecs[i]	object	
codecs[i].codec	string	Currently selected codec
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 - OK

The response is a JSON object.

Name	Type	Description
codec	string	Currently selected codec
container	string	Multimedia container format

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

PUT /system/codecFormat

Set the codec.

Parameters

Name	Type	Description
codec	string	Currently selected codec
container	string	Multimedia container format

Response

204 - No Content

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

GET /system/videoFormat

Get the currently selected video format.

Response

200 - OK

The response is a JSON object.

Name	Type	Description
name	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 video format.

Parameters

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?

Response

204 - No Content

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 - OK

The response is a JSON object.

Name	Type	Description
formats	array	
formats[i]	object	
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 - OK

The response is a JSON object.

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

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

GET /system/format

Get current format.

Response

200 - OK

The response is a JSON object.

Name	Type	Description
codec	string	Currently selected codec
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	Currently selected codec
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 - No Content

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 - OK

The response is a JSON object.

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 - No Content

GET /transports/0/stop

Determine if transport is stopped.

Response

200 - OK

The response is a JSON object.

PUT /transports/0/stop

Stop transport.

Response

204 - No Content

GET /transports/0/play

Determine if transport is playing.

Response

200 - OK

The response is a JSON object.

PUT /transports/0/play

Start playing on transport.

Response

204 - No Content

GET /transports/0/playback

Get playback state.

Response

200 - OK

The response is a JSON object.

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

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

Response

204 - No Content

GET /transports/0/record

Get record state.

Response

200 - OK

The response is a JSON object.

Name	Type	Description
recording	boolean	Is transport in Input Record mode

PUT /transports/0/record

Set record state.

Parameters

Name	Type	Description
recording	boolean	Is transport in Input Record mode
clipName	string	Used to set the requested clipName to record to, when specifying "recording" attribute to True

Response

204 - No Content

GET /transports/0/timecode

Get device's timecode.

Response

200 - OK

The response is a JSON object.

Name	Type	Description
timecode	number	The time of day timecode in units of binary-coded decimal (BCD).
clip	number	The position of the clip timecode in units of binary-coded decimal (BCD).

GET /transports/0/timecode/source

Get timecode source selected on device

Response

200 - OK

The response is a JSON object.

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

Timeline Control API

API for controlling playback timeline.

GET /timelines/0

Get the current playback timeline.

Response

200 - OK

The response is a JSON object.

Name	Type	Description
clips	array	
clips[i]	object	
clips[i].clipUniqueId	integer	Unique ID used to identify this clip
clips[i].frameCount	integer	Number of frames in this clip on the timeline

DELETE /timelines/0

Clear the current playback timeline.

Response

204 - No Content

POST /timelines/0/add

Add a clip to the end of the timeline.

Parameters

This parameter can be one of the following types:

Name	Type	Description
clips	integer	Unique ID used to identify this clip

Name	Type	Description
clips	array	
clips[i]	integer	Unique ID used to identify this clip

Response

204 - No Content

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 - OK

The response is a JSON object.

Name	Type	Description
size	integer	The fixed size of this device's working set
workingset (required)	array	
workingset[i]	object	
workingset[i].index	integer	Index of this media in the working set
workingset[i].activeDisk	boolean	Is this current item the active disk
workingset[i].volume	string	Volume name
workingset[i].deviceName	string	Internal device name of this media device
workingset[i].remainingRecordTime	integer	Remaining record time on media device 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 - OK

The response is a JSON object.

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

PUT /media/active

Set the currently active media device.

Parameters

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

Response

204 - No Content

GET /media/devices/dofORMATSupportedFilesystems

Get the list of filesystems available to format the device.

Response

200 - OK

The response is a JSON object.

GET /media/devices/{deviceName}

Get information about the selected device.

Parameters

Name	Type	Description
{deviceName}	string	

Response

200 - OK

The response is a JSON object.

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

GET /media/devices/{deviceName}/dofORMAT

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

Parameters

Name	Type	Description
{deviceName}	string	

Response

200 - OK

The response is a JSON object.

Name	Type	Description
deviceName	string	Internal device name of this media device
key	string	The key used to format this device, it must be fetched with the GET request and then provided back with a PUT request

PUT /media/devices/{deviceName}/doformat

Perform a format of the media device.

Parameters

Name	Type	Description
{deviceName}	string	

Name	Type	Description
key	string	The key used to format this device, it must be fetched with the GET request and then provided back with a PUT request
filesystem	string	Filesystem to format to (supportedFilesystems returns list of supported fileSystems)
volume	string	Volume name to set for the disk after format

Response

204 - No Content

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 a JSON object.

Name	Type	Description
presets	array	List of the presets on the camera
presets[i]	string	

POST /presets

Send a preset file to the camera

Response

200 - OK

The response is a JSON object.

Name	Type	Description
presetAdded	string	Name of the preset uploaded

GET /presets/active

Get the list of the presets on the camera

Response

200 - OK

The response is a JSON object.

Name	Type	Description
preset	string	

PUT /presets/active

Set the active preset on the camera

Parameters

Name	Type	Description
preset	string	

Response

200 - OK

The response is a JSON object.

GET /presets/{presetName}

Download the preset file

Parameters

Name	Type	Description
{presetName}	string	

Response

200 - OK

The response is a binary file.

PUT /presets/{presetName}

Update a preset on the camera if it exists, if not create a preset and save current state with the presetName

Parameters

Name	Type	Description
{presetName}	string	

Response

200 - OK

The response is a JSON object.

DELETE /presets/{presetName}

Delete a preset from a camera if exists

Parameters

Name	Type	Description
{presetName}	string	

Response

200 - OK

The response is a JSON object.

Audio Control API

API For controlling audio on Blackmagic Design Cameras

GET /audio/channel/{channelIndex}/input

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

Parameters

Name	Type	Description
{channelIndex}	integer	

Response

200 - Currently selected input

The response is a JSON object.

Name	Type	Description
input	string	Possible values are: None, Camera - Left, Camera - Right, Camera - Mono, XLR1 - Mic, XLR1 - Line, XLR2 - Mic, XLR2 - Line, 3.5mm Left - Line, 3.5mm Left - Mic, 3.5mm Right - Line, 3.5mm Right - Mic, 3.5mm Mono - Line, 3.5mm Mono - Mic.

404 - Channel does not exist

PUT /audio/channel/{channelIndex}/input

Set the audio input for the selected channel

Parameters

Name	Type	Description
{channelIndex}	integer	

Name	Type	Description
input	string	Possible values are: None, Camera - Left, Camera - Right, Camera - Mono, XLR1 - Mic, XLR1 - Line, XLR2 - Mic, XLR2 - Line, 3.5mm Left - Line, 3.5mm Left - Mic, 3.5mm Right - Line, 3.5mm Right - Mic, 3.5mm Mono - Line, 3.5mm Mono - Mic.

Response

200 - OK

400 - Invalid 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}	integer	

Response

200 - Description of the current input of the selected channel

The response is a JSON object.

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

Response

200 - The list of supported inputs

The response is a JSON object.

Name	Type	Description
supportedInputs	array	
supportedInputs[i]	object	
supportedInputs[i].schema	object	
supportedInputs[i].schema.input	string	Possible values are: None, Camera - Left, Camera - Right, Camera - Mono, XLR1 - Mic, XLR1 - Line, XLR2 - Mic, XLR2 - Line, 3.5mm Left - Line, 3.5mm Left - Mic, 3.5mm Right - Line, 3.5mm Right - Mic, 3.5mm Mono - Line, 3.5mm Mono - Mic.
supportedInputs[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}	integer	

Response

200 - Currently set level for the selected channel

The response is a JSON object.

Name	Type	Description
gain	number	
normalised	number	

404 - Channel does not exist

PUT /audio/channel/{channelIndex}/level

Set the audio input level for the selected channel

Parameters

Name	Type	Description
{channelIndex}	integer	

Name	Type	Description
gain	number	
normalised	number	

Response

200 - OK

400 - Invalid input

404 - Channel does not exist

GET /audio/channel/{channelIndex}/phantomPower

Get the audio input phantom power for the selected channel if possible

Parameters

Name	Type	Description
{channelIndex}	integer	

Response

200 - Currently set level for the selected channel

The response is a JSON object.

Name	Type	Description
phantomPower	boolean	

404 - Channel does not exist

PUT /audio/channel/{channelIndex}/phantomPower

Set the audio phantom power for the selected channel

Parameters

Name	Type	Description
{channelIndex}	integer	

Name	Type	Description
phantomPower	boolean	

Response

200 - OK

400 - Phantom power is not supported for this input

404 - Channel does not exist

GET /audio/channel/{channelIndex}/padding

Get the audio input padding for the selected channel

Parameters

Name	Type	Description
{channelIndex}	integer	

Response

200 - Currently set padding for the selected channel

The response is a JSON object.

Name	Type	Description
padding	boolean	

404 - Channel does not exist

PUT /audio/channel/{channelIndex}/padding

Set the audio input padding for the selected channel

Parameters

Name	Type	Description
{channelIndex}	integer	

Name	Type	Description
padding	boolean	

Response

200 - OK

400 - Padding is not supported for this input

404 - Channel does not exist

GET /audio/channel/{channelIndex}/lowCutFilter

Get the audio input low cut filter for the selected channel

Parameters

Name	Type	Description
{channelIndex}	integer	

Response

200 - Currently set low cut filter for the selected channel

The response is a JSON object.

Name	Type	Description
lowCutFilter	boolean	

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}	integer	

Name	Type	Description
lowCutFilter	boolean	

Response

200 - OK

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}	integer	

Response

200 - Currently set availability for the selected channel

The response is a JSON object.

Name	Type	Description
available	boolean	

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 a JSON object.

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

PUT /lens/iris

Set lens' aperture

Parameters

Name	Type	Description
apertureStop	number	Aperture stop value
normalised	number	Normalised value
apertureNumber	number	Aperture number

Response

200 - OK

GET /lens/zoom

Get lens' zoom

Response

200 - OK

The response is a JSON object.

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

Response

200 - OK

GET /lens/focus

Get lens' focus

Response

200 - OK

The response is a JSON object.

Name	Type	Description
focus	number	Normalised value

PUT /lens/focus

Set lens' focus

Parameters

Name	Type	Description
focus	number	Normalised value

Response

200 - OK

PUT /lens/focus/doAutoFocus

Perform auto focus

Response

200 - OK

Video Control API

API For controlling the video on Blackmagic Design products

GET /video/iso

Get current ISO

Response

200 - OK

The response is a JSON object.

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

200 - OK

GET /video/gain

Get current gain value in decibels

Response

200 - OK

The response is a JSON object.

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

200 - OK

GET /video/whiteBalance

Get current white balance

Response

200 - OK

The response is a JSON object.

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

200 - OK

PUT /video/whiteBalance/doAuto

Set current white balance automatically

Response

200 - OK

GET /video/whiteBalanceTint

Get white balance tint

Response

200 - OK

The response is a JSON object.

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

200 - OK

GET /video/ndFilter

Get ND filter stop

Response

200 - OK

The response is a JSON object.

Name	Type	Description
stop	number	Current filter power (fStop)

PUT /video/ndFilter

Set ND filter stop

Parameters

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

Response

200 - OK

GET /video/ndFilter/displayMode

Get ND filter display mode on the camera

Response

200 - OK

The response is a JSON object.

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

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

200 - OK

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 a JSON object.

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	integer	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	integer	Shutter angle

Response

200 - OK

GET /video/autoExposure

Get current auto exposure mode

Response

200 - OK

The response is a JSON object.

Name	Type	Description
mode	object	Auto exposure mode
mode.mode	string	Possible values are: Off, Continuous, OneShot.
mode.type	string	Possible values are: , Iris, Shutter, Iris,Shutter, Shutter,Iris.

PUT /video/autoExposure

Set auto exposure

Parameters

Name	Type	Description
mode	object	Auto exposure mode
mode.mode	string	Possible values are: Off, Continuous, OneShot.
mode.type	string	Possible values are: , Iris, Shutter, Iris,Shutter, Shutter,Iris.

Response

200 - OK

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 a JSON object.

Name	Type	Description
red	number	
green	number	
blue	number	
luma	number	

PUT /colorCorrection/lift

Set color correction lift

Parameters

Name	Type	Description
red	number	
green	number	
blue	number	
luma	number	

Response

200 - OK

GET /colorCorrection/gamma

Get color correction gamma

Response

200 - OK

The response is a JSON object.

Name	Type	Description
red	number	
green	number	
blue	number	
luma	number	

PUT /colorCorrection/gamma

Set color correction gamma

Parameters

Name	Type	Description
red	number	
green	number	
blue	number	
luma	number	

Response

200 - OK

GET /colorCorrection/gain

Get color correction gain

Response

200 - OK

The response is a JSON object.

Name	Type	Description
red	number	
green	number	
blue	number	
luma	number	

PUT /colorCorrection/gain

Set color correction gain

Parameters

Name	Type	Description
red	number	
green	number	
blue	number	
luma	number	

Response

200 - OK

GET /colorCorrection/offset

Get color correction offset

Response

200 - OK

The response is a JSON object.

Name	Type	Description
red	number	
green	number	
blue	number	
luma	number	

PUT /colorCorrection/offset

Set color correction offset

Parameters

Name	Type	Description
red	number	
green	number	
blue	number	
luma	number	

Response

200 - OK

GET /colorCorrection/contrast

Get color correction contrast

Response

200 - OK

The response is a JSON object.

Name	Type	Description
pivot	number	Default value is: 0.5.
adjust	number	Default value is: 1.

PUT /colorCorrection/contrast

Set color correction contrast

Parameters

Name	Type	Description
pivot	number	Default value is: 0.5.
adjust	number	Default value is: 1.

Response

200 - OK

GET /colorCorrection/color

Get color correction color properties

Response

200 - OK

The response is a JSON object.

Name	Type	Description
hue	number	
saturation	number	Default value is: 1.

PUT /colorCorrection/color

Set color correction color properties

Parameters

Name	Type	Description
hue	number	
saturation	number	Default value is: 1.

Response

200 - OK

GET /colorCorrection/lumaContribution

Get color correction luma contribution

Response

200 - OK

The response is a JSON object.

Name	Type	Description
lumaContribution	number	Default value is: 1.

PUT /colorCorrection/lumaContribution

Set color correction luma contribution

Parameters

Name	Type	Description
lumaContribution	number	Default value is: 1.

Response

200 - OK

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.
				[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	–	–
	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 - sDDDdddddddddd 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
					-	-	CinemaDNG: 0 = uncompressed, 1 = lossy 3:1, 2 = lossy 4:1
				[1] = code variant	-	-	ProRes: 0 = HQ, 1 = 422, 2 = LT, 3 = Proxy, 4 = 444, 5 = 444XQ
					-	-	Blackmagic RAW: 0 = Q0, 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
				[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:

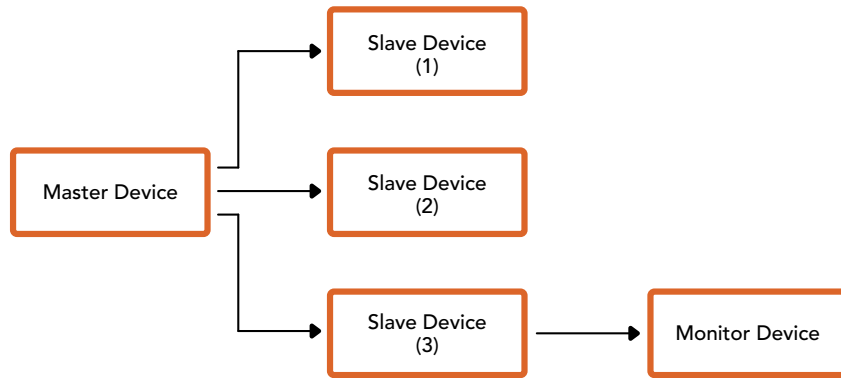
```
uint4
bit 0:  program tally status (0=off, 1=on)
bit 1:  preview tally status (0=off, 1=on)
bit 2-3: reserved (0x0)
```

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

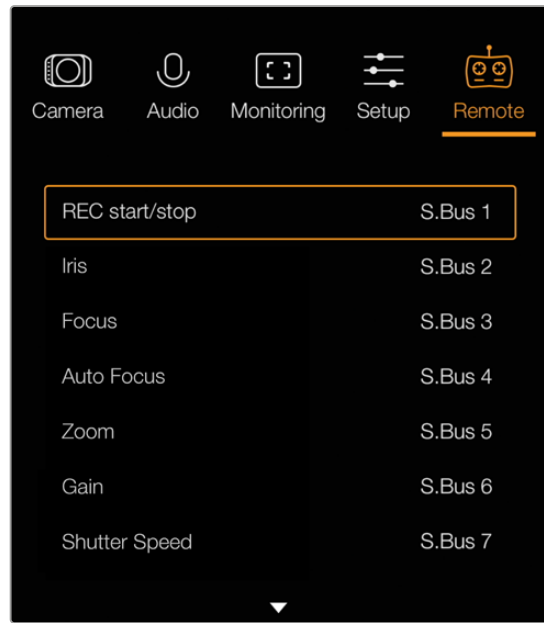
Read this value to determine the camera's supported CCU protocol version.

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.

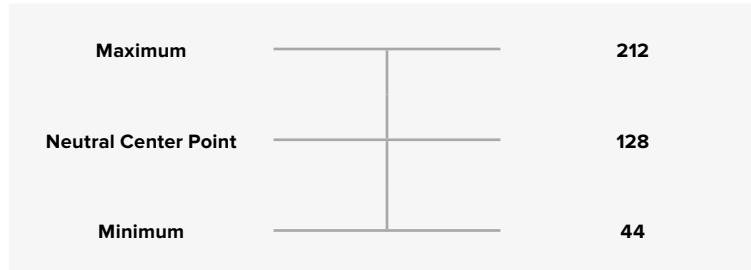
- The first type maps settings to specific ranges of the input so that sending a value within a certain range will trigger a particular setting.

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.

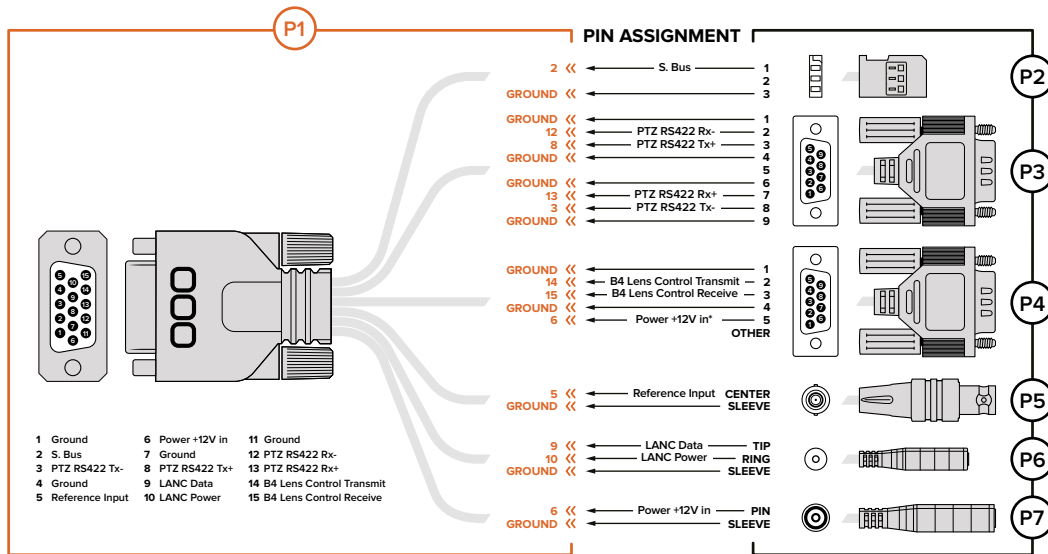
Analog radio control transmitters and receivers are typically used to drive servos on model airplanes, but can also be used for wireless remote control of the Blackmagic Micro Cinema Camera.

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

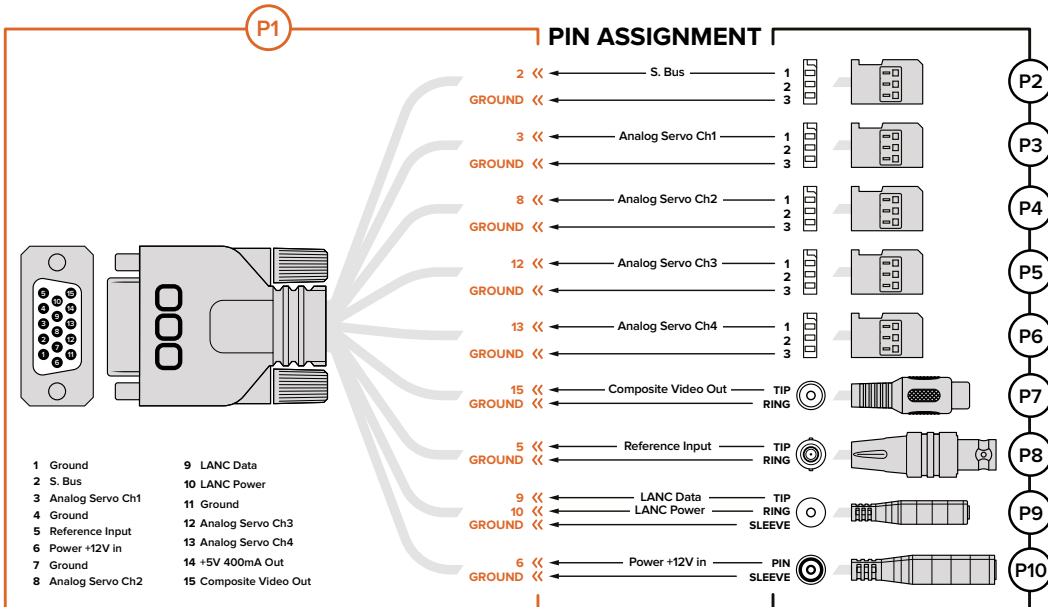
When using Blackmagic Micro Cinema Camera or Micro Studio Camera 4K's expansion port, you may only want to access one or two functions. For example, you may want to control an attached B4 Broadcast Lens while simultaneously receiving 12V power and a reference signal. It's easy to make a connector that 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.



* Power input to the camera is also used to power the lens. Beware of applying excessive voltages if you're using your own power supply to avoid damage to the lens.

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



Wiring Diagram for the Blackmagic Micro Cinema Camera Expansion Cable