SW-620-TX-W

Application Programming Interface

Document Revision	V2.1
Document Date	November 2026
Supported Firmware	Refer to supported product firmware/software for details



Contents

0. Device Communication ---- 7

0.1 Overview ---- 7

0.2 Network Connection ---- 7

0.3 Serial Connection ---- 7

0.4 Command Delimiters for Sent Commands ---- 7

1. Device Management ---- 8

1.1 Set Device Alias ---- 8

1.2 Get Device Alias ---- 8

1.3 Get Device Version Information ---- 9

1.4 Device Reboot ---- 9

1.5 Device Reset ---- 10

1.6 Display OSD Elements ---- 10

1.7 Set OSD Element Display ---- 11

1.8 Get OSD Element Display ---- 11

1.9 Get MAC Address ---- 12

1.10 Get Device Serial Number ---- 12

2. Input/Output Settings ---- 13

2.1 Set Output Resolution ---- 13

2.2 Get Output Resolution ---- 13

2.3 Set Input Interface HDCP ---- 14

2.4 Get Input Interface HDCP Status ---- 14

2.5 Get Input Signal HDCP ---- 15

2.6 Set HDCP Management Feature ---- 15

2.7 Get HDCP Management Feature ---- 16

2.8 Get Output Signal HDCP ---- 16

2.9 Get Input Connection Status ---- 17

2.10 Get Input Signal Status ---- 17

2.11 Get Input Video Format ---- 18

2.12 Get Input Audio Format ---- 18

2.13 Set Output Volume ---- 19

2.14 Get Output Volume ---- 19

2.15 Set Output Mute ---------- 19
2.16.1 Get Output Mute---------- 20
2.16.2 Get All Output Mute Status ---------- 20
2.17 Get Output Connection Status---------- 21
2.18 Set MST---------- 21
2.19 Get MST ---------- 22
2.20 Volume Increment---------- 22
2.21 Volume Decrease---------- 23
3. Video Switching ---------- 23
3.1 Set Automatic Switching ---------- 23
3.2 Get Automatic Switching Status---------- 24
3.3 Set Multi-View Function ---------- 24
3.4 Get Multi-View Function Status ---------- 24
3.5 Switch Input to All Outputs ---------- 25
3.6 Switch Input to an Output ---------- 25
3.7 Display GUIDE Screen ---------- 26
3.8 Get Display Status ---------- 26
3.9 Set Current Layout ---------- 27
3.10 Set Current Layout with ID---------- 27
3.11 Get Current Layout ID and Name ---------- 27
3.12 Set Layout and Video Sources ---------- 28
3.13 Get Layout and Video Sources ---------- 28
3.14 Get Video Source of Specified Window ---------- 29
4. Audio and Other Signal Switching ---------- 29
4.1 Set Audio Auto-Switching ---------- 29
4.2 Get Audio Auto-Switch Status---------- 30
4.3 Switch to Wired Audio Source---------- 30
4.4 Switch to BYOD Audio Source---------- 31
4.5 Get Current Audio Source ---------- 31
4.6 Set USB Automatic Switching ---------- 32
4.7 Get USB Auto-Switch Status---------- 32
4.8 Set USB Host ---------- 32

4.9 Get USB Host ---- 33
4.10 Set VBUS Mode ---- 33
4.11 Get VBUS Mode ---- 34
5. Sleep/Standby and Peripheral Control ---- 34
5.1 Set Automatic Standby ---- 34
5.2 Get Automatic Standby ---- 35
5.3 Set Automatic Standby Timeout ---- 35
5.4 Get Auto Standby Timeout ---- 35
5.5 Set Device Power On or Standby ---- 36
5.6 Get Device Standby Status ---- 36
5.7 Send CEC Power On/Off Command ---- 36
5.8 Set CEC Power On/Off Command ---- 37
5.9 Get CEC Power On/Off Command ---- 37
5.10 Get CEC Commands ---- 38
5.11 Get All CEC Commands ---- 38
5.12 Send CEC Command String ---- 39
5.13 Set UART Parameters ---- 39
5.14 Get UART Parameters ---- 39
5.15 Set UART Baud Rate ---- 40
5.16 Get UART Baud Rate ---- 40
5.17 Set UART Command ---- 40
5.18 Delete UART Command ---- 41
5.19 Get UART Commands ---- 41
5.20 Send UART Command ---- 42
5.21 Send UART Command String ---- 42
5.22 Set UART Power On/Off Function ---- 43
5.23 Get UART Power Control Function ---- 43
5.24 Send CEC and/or RS232 Commands ---- 44
5.25 Set UART Operating Mode ---- 44
5.26 Get UART Operating Mode ---- 44
5.27 Get MUTE Input ---- 45
5.28 Set CEC Power On/Off Function ---- 45

5.29 Get CEC Power On/Off Function ---- 46
6. Network and BYOD Related Settings ---- 46
6.1 Set Network Isolation ---- 46
6.2 Get Network Isolation Settings ---- 47
6.3 Set IP Address - DHCP ---- 47
6.4 Set IP Address - Static ---- 48
6.5 Get IP Address ---- 48
6.6 Set USB-C Network Adapter Function ---- 49
6.7 Get USB Network Adapter On/Off Status ---- 49
6.8 Configure HTTPS Functionality ---- 50
6.9 Get HTTPS Switch Status ---- 50
6.10 Configure Telnet Function ---- 50
6.11 Get Telnet Feature Switch Status ---- 51
6.12 Set Wi-Fi Operating Mode ---- 51
6.13 Get Wi-Fi Operating Mode ---- 51
6.14 Set Wi-Fi Password ---- 52
6.15 Get Wi-Fi Password ---- 52
6.16 Configure Soft AP Function ---- 52
6.17 Get Soft AP Function Switch Status ---- 53
6.18 Set SoftAP Router Function ---- 53
6.19 Get Soft Router Feature Switch Status ---- 54
6.20 Configure BYOD Feature ---- 54
6.21 Get BYOD Feature ---- 54
6.22 Set BYOD Access Code ---- 55
6.23 Get BYOD Access Code ---- 55
6.24 Set SSID Broadcast ---- 56
6.25 Get SSID Broadcast Switch Status ---- 56
6.26 Set Forced Transcoding for Non-Flow-Control Cameras ---- 57
6.27 Get Forced Transcoding for Non-Flow-Control Cameras ---- 57
6.28 Configure SSH ---- 57
6.29 Get SSH Switch Status ---- 58
6.30 Set Telnet Login Authentication ---- 58

6.31 Get Telnet Login Authentication Feature Status ---------- 58
6.32 Set API Login Authentication Password ---------- 59
7. AP / Infrastructure Mode---------- 59
7.1 Set Wireless Network Mode ---------- 59
7.2 Get Wireless Network Mode---------- 60
7.3 Configure External Wi-Fi ---------- 60
7.4 Get External Wi-Fi ---------- 60
8. Wireless Multi Camera Settings---------- 61
8.1 Set Camera Alias---------- 61
8.2 Get Camera Aliases ---------- 61
8.3 Set Camera Priority ---------- 62
8.4 Get Camera Priority---------- 62
9. Wireless Multi Audio Device Settings---------- 63
9.1 Set Wireless Conference Audio Device Selection Mode ---------- 63
9.2 Get Wireless Conference Audio Device Selection Mode ---------- 64
9.3 Manual Selection of Wireless Conference Audio Device ---------- 64
9.4 Get Manually Selected Audio Device for Wireless Conference ---------- 64
9.5 Get USB Port Audio Device Status ---------- 65
9.6 Get USB Port Audio Device Name---------- 65
9.7 Get Active USB Audio Device---------- 66

0. Device Communication

0.1 Overview

The following contains the connection and commands to control the SW-620-TX-W. By following the content contained in this document, the device can be controlled and configured via 3rd party control system.

⚠ IMPORTANT NOTE: Due to differences between WyreStorm device model versions within a series, some commands have different parameters based on the model and version. These differences are noted where applicable and should be followed, as sending an incorrect parameter may cause the unit to lock up and become inoperable.

0.2 Network Connection

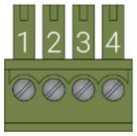
By default, the IP address is assigned by the DHCP server. If the DHCP server is not available the unit will use a link local IP address based on its MAC address, if the unit was not already configured via the Web interface.

Default communication port number is 23.

0.3 Serial Connection

The SW-620-TX-W uses a 3-pin RS-232 with no hardware flow control. Most control systems and computers are DTE where pin 2 is RX, this can vary from device to device. Refer to the documentation for the connected device for pin functionally to ensure that the correct connections can be made.

RS-232 Settings
Baud Rate: 9600
Data Bits: 8bits
Parity: None
Stop Bits: 1bit
Flow Control: None



WyreStorm Connector			3rd Party device
Pin 1	12V DC Out	No Connection	Reserved
Pin 2	TX (Transmit)	---> To --->	RX (Receive)
Pin 3	RX (Receive)	---> To --->	TX (Transmit)
Pin 4	G (Ground)	---> To --->	G (Ground)

0.4 Command Delimiters for Sent Commands

All commands should be delimited with carriage return and line feed.

Accepted delimiter characters are:

Character	Shorthand	Hex Notation	Escape Notation	Decimal Notation
Line Feed	LF	0A	\n	10
Carriage Return + Line Feed	CR LF	0D 0A	\r\n	13 10

Please note, most 3rd party control software will either append these characters automatically or an option to specify them will be present. It is important that the last delimited character is LF and not CR.

1. Device Management

1.1 Set Device Alias

Set the device alias.
This alias is used as the receiver name for BYOD casting methods such as Miracast and AirPlay, and is also used as the SSID for the built-in soft AP.

Command Structure **SET ALIAS < DeivceName >**	Parameters: DeviceName = The alias to set, up to 31 characters. Only letters, numbers, spaces, underscores ('_') and hyphens ('-') are allowed. Spaces cannot appear at the beginning or end.
Response Structure **ALIAS < DeivceName >**	
Command Example **SET ALIAS MeetingRoom**	
Response Example **ALIAS MeetingRoom**	
Additional Notes: Default value: SW-620-TX-W-XXXXXX, where XXXXXX are the hexadecimal form of the last two bytes of the MAC address. For example: SW-620-TX-W-00327C	

1.2 Get Device Alias

Command Structure **GET ALIAS**	Parameters:
Response Structure **ALIAS < DeivceName >**	
Command Example **GET ALIAS**	
Response Example **ALIAS MeetingRoom**	
Additional Notes:	

1.3 Get Device Version Information

Get version information for the device's software modules

Command Structure **GET VER < ALL | ModuleName >**	Parameters: ModuleName = The software module whose version you want to get. Supported values are: MAINSOC, USBCCHIP, USBCVIDEO, VIDEOCHIP ALL = Get version information for all software modules. When ALL is used, the device returns the version information for each module on separate lines.
Response Structure **VER ModuleName Version** **or** **VER ModuleName1 Version1** **VER ModuleName2 Version2** **...**	
Command Example **GET VER MAINSOC**	
Response Example **VER MAINSOC 1.4.0**	
Additional Notes:	

1.4 Device Reboot

Command the device to reboot.

Command Structure **REBOOT**	Parameters:
Response Structure **REBOOT**	
Command Example **REBOOT**	
Response Example **REBOOT**	
Additional Notes:	

1.5 Device Reset

Command to restore the device to factory settings.

Command Structure **RESET [GUIDE]**	Parameters: If the optional parameter GUIDE is provided, only the Guide screen image will be reset.
Response Structure **RESET [GUIDE]**	
Command Example **RESET**	
Response Example **RESET**	
Additional Notes:	

1.6 Display OSD Elements

Displays all allowed OSD elements for 10 seconds

Command Structure **SET SHOWOSD**	Parameters:
Response Structure **SHOWOSD**	
Command Example **SET SHOWOSD**	
Response Example **SHOWOSD**	
Additional Notes:	

1.7 Set OSD Element Display

Configure whether an OSD element is allowed to be displayed.

Command Structure **SET SHOWOSD < DEVICENAME | IPADDRESS | WIFIPWD | ACCESSCODE > < ON | OFF >**	Parameters: For the first parameter, the four possible values correspond respectively to device name, IP address, Wi-Fi password, and access code.
Response Structure **SHOWOSD < DEVICENAME | IPADDRESS | WIFIPWD | ACCESSCODE > < ON | OFF >**	
Command Example **SET SHOWOSD DEVICENAME OFF**	
Response Example **SHOWOSD DEVICENAME OFF**	
Additional Notes: Default: All four OSD elements mentioned are allowed to be displayed (default value ON).	

1.8 Get OSD Element Display

Retrieve the configuration that indicates whether OSD elements are allowed to be displayed.

Command Structure **GET SHOWOSD < DEVICENAME | IPADDRESS | WIFIPWD | ACCESSCODE | ALL >**	Parameters: ALL = Get the display settings for all OSD elements. The device will return each OSD element's display setting line by line.
Response Structure **SHOWOSD < DEVICENAME | IPADDRESS | WIFIPWD | ACCESSCODE > < ON | OFF >** **or** **SHOWOSD DEVICENAME < ON | OFF >** **SHOWOSD IPADDRESS < ON | OFF >** **SHOWOSD WIFIPWD < ON | OFF >** **SHOWOSD ACCESSCODE < ON | OFF >**	
Command Example **GET SHOWOSD DEVICENAME**	
Response Example **SHOWOSD DEVICENAME OFF**	
Additional Notes:	

1.9 Get MAC Address

Get the device's MAC address information

Command Structure **GET MACADDR < CONTROL | SERVICE | ALL >**	Parameters: CONTROL = Get the MAC address of the control network port SERVICE = Get the MAC address of the service network port ALL = Get the MAC addresses of both of the above network ports
Response Structure **MACADDR < CONTROL | SERVICE > < MACAddress >** **or** **MACADDR CONTROL < MACAddress0 >** **MACADDR SERVICE < MACAddress1 >**	
Command Example **GET MACADDR CONTROL**	
Response Example **MACADDR CONTROL E4:CE:02:13:5F:A1**	
Additional Notes:	

1.10 Get Device Serial Number

Read the device's serial number information.

Command Structure **GET SYS_SN**	Parameters:
Response Structure **SYS_SN < SerialNumber >**	
Command Example **GET SYS_SN**	
Response Example **SYN_SN WS29125260001**	
Additional Notes:	

2. Input/Output Settings

2.1 Set Output Resolution

Set the HDMI output interface resolution.
You can directly specify a resolution, or let the device automatically determine the best resolution.

Command Structure
SET VIDOUT_RES < OUT1 | OUT2 > < AUTO | Resolution >

Response Structure
VIDOUT_RES < OUT1 | OUT2 > < AUTO | Resolution >

Command Example
```
SET VIDOUT_RES OUT1 AUTO
```

Response Example
```
VIDOUT_RES OUT1 AUTO
```

Parameters:
AUTO = Automatically select the best output resolution based on the connected display's EDID.
Resolution = A specific resolution. Valid values are:
3840x2160P@60 3840x2160P@50 3840x2160P@30 3840x2160P@25 3840x2160P@24 1920x1080P@60 1920x1080P@50 1920x1080P@30 1920x1080P@25 1920x1080P@24 1680x1050P@60 1600x1200P@60 1440x900P@60 1366x768P@60 1280x1024P@60 1280x720P@60 1280x720P@50 1024x768P@60 800x600P@60 720x480P@60 640x480P@60
Note: 3840x2160P@60 and 3840x2160P@50 are supported only on OUT1.

Additional Notes: Default: both outputs are set to AUTO

2.2 Get Output Resolution

Retrieve the resolution of the HDMI output interface(s).

Command Structure
GET VIDOUT_RES < OUT1 | OUT2 | ALL >

Response Structure
VIDOUT_RES < OUT1 | OUT2 > < Resolution | DISCONNECTED > [AUTO]
or
VIDOUT_RES OUT1 < Resolution | DISCONNECTED > [AUTO]
VIDOUT_RES OUT2 < Resolution | DISCONNECTED > [AUTO]

Command Example
```
GET VIDOUT_RES OUT1
```

Response Example
```
VIDOUT_RES OUT1 3840x2160P@60 AUTO
```

Parameters:
ALL = Retrieve the resolutions of both output interfaces; the device will return the two outputs' resolutions on two separate lines.
AUTO = The output interface is set to automatic resolution.
DISCONNECTED = The output interface is set to automatic resolution but no display is currently connected to that interface.

Additional Notes:

2.3 Set Input Interface HDCP

Set whether HDCP is enabled on an input interface.

Command Structure **SET HDCP_S < IN1 | IN2 > < ON | OFF >**	Parameters:
Response Structure **HDCP_S < IN1 | IN2 > < ON | OFF >**	
Command Example **SET HDCP_S IN1 OFF**	
Response Example **HDCP_S IN1 OFF**	
Additional Notes: Default: HDCP is enabled on all input interfaces.	

2.4 Get Input Interface HDCP Status

Retrieve whether HDCP is enabled on input interfaces.

Command Structure **GET HDCP_S < IN1 | IN2 | ALL >**	Parameters: ALL = Get the HDCP on/off status for all input interfaces
Response Structure **HDCP_S < IN1 | IN2 > < ON | OFF >** **or** **HDCP_S IN1 < ON | OFF >** **HDCP_S IN2 < ON | OFF >**	
Command Example **GET HDCP_S IN1**	
Response Example **HDCP_S IN1 OFF**	
Additional Notes:	

2.5 Get Input Signal HDCP

Get the HDCP encryption status of the input signal

Command Structure **GET VIDIN_HDCP < IN1 | IN2 | ALL >**	Parameters: ALL = Get the HDCP encryption status for all input signals
Response Structure **VIDIN_HDCP < IN1 | IN2 > < OFF | HDCP1.4 | HDCP2.2 >** **or** **VIDIN_HDCP IN1 < OFF | HDCP1.4 | HDCP2.2 >** **VIDIN_HDCP IN2 < OFF | HDCP1.4 | HDCP2.2 >**	
Command Example **GET VIDIN_HDCP IN1**	
Response Example **VIDIN_HDCP IN1 OFF**	
Additional Notes:	

2.6 Set HDCP Management Feature

Set whether the HDCP management feature is enabled.
When HDCP management is disabled, HDCP is forced off on all input and output interfaces.

Command Structure **SET HDCP_FEATURE < ON | OFF >**	Parameters:
Response Structure **HDCP_FEATURE < ON | OFF >**	
Command Example **SET HDCP_FEATURE OFF**	
Response Example **HDCP_FEATURE OFF**	
Additional Notes: Default: OFF HDCP management feature enabled	

2.7 Get HDCP Management Feature

Retrieve the status of the HDCP management feature.

Command Structure **GET HDCP_FEATURE**	Parameters:
Response Structure **HDCP_FEATURE < ON | OFF >**	
Command Example **GET HDCP_FEATURE**	
Response Example **HDCP_FEATURE OFF**	
Additional Notes:	

2.8 Get Output Signal HDCP

Get the HDCP encryption status of the output signal.

Command Structure **GET VIDOUT_HDCP < OUT1 | OUT2 | ALL >**	Parameters: ALL = Get the HDCP encryption status of all output signals
Response Structure **VIDOUT_HDCP < OUT1 | OUT2 > < OFF | HDCP1.4 | HDCP2.2 >** **or** **VIDOUT_HDCP OUT1 < OFF | HDCP1.4 | HDCP2.2 >** **VIDOUT_HDCP OUT2 < OFF | HDCP1.4 | HDCP2.2 >**	
Command Example **GET VIDOUT_HDCP OUT1**	
Response Example **VIDOUT_HDCP OUT1 HDCP1.4**	
Additional Notes:	

2.9 Get Input Connection Status

Retrieve the connection status of the input interfaces.

Command Structure
GET VIDIN_CONNECT < IN1 | IN2 | ALL >

Response Structure
VIDIN_CONNECT < IN1 | IN2 > < DISCONNECTED | CONNECTED >
or
VIDIN_CONNECT IN1 < DISCONNECTED | CONNECTED >
VIDIN_CONNECT IN2 < DISCONNECTED | CONNECTED >

Command Example
```
GET VIDIN_CONNECT IN2
```

Response Example
```
VIDIN_CONNECT IN2 DISCONNECTED
```

Parameters:
ALL = Retrieve the connection status for all input interfaces.

Additional Notes:

2.10 Get Input Signal Status

Get the status of input signals

Command Structure
GET VIDIN_SIG < IN1 | IN2 | BYOD1 | BYOD2 | ALL >

Response Structure
VIDIN_SIG < IN1 | IN2 | BYOD1 | BYOD2 > < NO | VALID >
or
VIDIN_SIG IN1 < NO | VALID >
VIDIN_SIG IN2 < NO | VALID >
VIDIN_SIG BYOD1 < NO | VALID >
VIDIN_SIG BYOD2 < NO | VALID >

Command Example
```
GET VIDIN_SIG IN2
```

Response Example
```
VIDIN_SIG IN2 VALID
```

Parameters:
ALL = Get the status of all input signals

Additional Notes:

2.11 Get Input Video Format

Get the video format of the input signal.
In addition to physical input sources, when the wireless meeting feature is active this API can also retrieve the input video formats of each USB camera.

Command Structure
```
GET VIDIN_FORMAT < IN1 | IN2 | BYOD1 | BYOD2 | ALL >
```

Response Structure
```
VIDIN_FORMAT < IN1 | IN2 | BYOD1 | BYOD2 > < INVALID | VideoFormat >
or
VIDIN_FORMAT IN1 < INVALID | VideoFormat >
VIDIN_FORMAT IN2 < INVALID | VideoFormat >
VIDIN_FORMAT BYOD1 < INVALID | VideoFormat >
VIDIN_FORMAT BYOD2 < INVALID | VideoFormat >
```

Command Example
```
GET VIDIN_FORMAT IN2
```

Response Example
```
VIDIN_FORMAT IN2 3840X2160P,60;NONE HDR;RGB;8BIT
```

Parameters:
ALL = Get the formats of all input video signals.
INVALID = No valid video signal detected.

The format of the parameter VideoFormat:
< WidthXHeight[P],FrameRate;HDRInfo;EncMode;ColorDepth >

The value and meaning of every field are as follows:
- Width = Video width (pixels)
- Height = Video height (pixels)
- P = Progressive scan
- FrameRate = Frame rate (frames per second)
- HDRInfo = HDR information. The device does not support HDR, so this is always NONE HDR.
- EncMode = Encoding mode. Possible values: RGB, YUV444, YUV422, H.264, H.265, MJPEG
- ColorDepth = Color depth. Currently only 8BIT is supported.
- An actual example of VideoFormat: 1920X1080P,60;NONE HDR;RGB;8BIT

Additional Notes:

2.12 Get Input Audio Format

Retrieve the audio format of the input signal.

Command Structure
```
GET AUDIN_FORMAT < IN1 | IN2 | BYOD1 | BYOD2 | ALL >
```

Response Structure
```
AUDIN_FORMAT < IN1 | IN2 | BYOD1 | BYOD2 > < INVALID | AudioFormat >
or
AUDIN_FORMAT IN1 < INVALID | AudioFormat >
AUDIN_FORMAT IN2 < INVALID | AudioFormat >
AUDIN_FORMAT BYOD1 < INVALID | AudioFormat >
AUDIN_FORMAT BYOD2 < INVALID | AudioFormat >
```

Command Example
```
GET AUDIN_FORMAT IN2
```

Response Example
```
AUDIN_FORMAT IN2 PCM;48kHz;8BIT
```

Parameters:
ALL = Get the audio format of all input interfaces.

INVALID = No valid audio signal detected on the input interface.
The format of the parameter AudioFormat:
< Encoding;SamplingRate;SamplingWidth >

The value and meaning of every field are as follows:
- Encoding = The audio encoding mode. Available values: PCM, AAC, MP3, OPUS
- SamplingRate = Sampling rate, for example 48kHz
- SamplingWidth = Sampling width (bit depth). Available values: 8BIT, 16BIT.
An actual example of AudioFormat: PCM;48kHz;8BIT

Additional Notes:

2.13 Set Output Volume

Simultaneously set the volume for all output audio signals.

Command Structure **SET VOLGAIN_DATA < OutputVolume >**	Parameters: OutputVolume is in dB. Valid range: -100 to 6.
Response Structure **VOLGAIN_DATA < OutputVolume >**	
Command Example **SET VOLGAIN_DATA -3**	
Response Example **VOLGAIN_DATA -3**	
Additional Notes: Default value: 0	

2.14 Get Output Volume

Get the volume level of the output audio signal.

Command Structure **GET VOLGAIN_DATA**	Parameters:
Response Structure **VOLGAIN_DATA < OutputVolume >**	
Command Example **GET VOLGAIN_DATA**	
Response Example **VOLGAIN_DATA -3**	
Additional Notes:	

2.15 Set Output Mute

Set the mute state of audio outputs.

Command Structure **SET AUDIO_MUTE [AudioOutput] < ON | OFF >**	Parameters: AudioOutput = OUT1, OUT2 OUT2 affects HDMI OUT2 and the analog audio output at the same time AudioOutput is an optional parameter; if omitted, the mute setting for both audio outputs will be changed at the same time. ON = Enable output mute OFF = Disable output mute
Response Structure **AUDIO_MUTE [AudioOutput] < ON | OFF >**	
Command Example **SET AUDIO_MUTE OUT1 ON**	
Response Example **AUDIO_MUTE OUT1 ON**	
Additional Notes: Default: OFF; both audio outputs are unmuted.	

2.16.1 Get Output Mute

Get output mute status

Command Structure **GET AUDIO_MUTE < AudioOutput | ALL >**	Parameters: AudioOutput = OUT1, OUT2 ALL = Query mute settings for both audio outputs ON = Enable output mute OFF = Disable output mute
Response Structure **AUDIO_MUTE < AudioOutput > < ON | OFF >** **或** **AUDIO_MUTE OUT1 < ON | OFF >** **AUDIO_MUTE OUT2 < ON { OFF >**	
Command Example **GET AUDIO_MUTE OUT1**	
Response Example **AUDIO_MUTE OUT1 ON**	
Additional Notes:	

2.16.2 Get All Output Mute Status

Retrieve the current mute settings for all audio outputs.

Command Structure **GET AUDIO_MUTE**	Parameters:
Response Structure **AUDIO_MUTE < ON | OFF >** **或** **AUDIO_MUTE OUT1 < ON | OFF >** **AUDIO_MUTE OUT2 < ON { OFF >**	
Command Example **GET AUDIO_MUTE**	
Response Example **AUDIO_MUTE ON**	
Additional Notes: If the mute settings for both audio outputs are the same, only one response message is returned; otherwise, two separate response messages are returned.	

2.17 Get Output Connection Status

Retrieve the connection status of the output ports (whether a display is connected).

Command Structure **GET VIDOUT_CONNECT < OUT1 | OUT2 | ALL >**	Parameters: ALL = Get the connection status of both output ports
Response Structure **VIDOUT_CONNECT < OUT1 | OUT2 > < CONNECTED | DISCONNECTED >** **或** **VIDOUT_CONNECT OUT1 < CONNECTED | DISCONNECTED >** **VIDOUT_CONNECT OUT2 < CONNECTED | DISCONNECTED >**	
Command Example **GET VIDOUT_CONNECT OUT2**	
Response Example **VIDOUT_CONNECT OUT2 CONNECTED**	
Additional Notes:	

2.18 Set MST

Configure the MST function for the USB-C input. MST will be actually enabled only when both conditions are met:
1. This setting is set to MST
2. Both HDMI outputs are connected
When MST is enabled, the device's video switching behavior will differ from normal.

Command Structure **SET USBC_DM < SST | MST >**	Parameters: SST = USB-C input operates in SST mode; MST not supported MST = USB-C input operates in MST mode
Response Structure **USBC_DM < SST | MST >**	
Command Example **SET USBC_DM MST**	
Response Example **USBC_DM MST**	
Additional Notes: Default: SST (MST disabled)	

2.19 Get MST

Retrieve the MST (Multi-Stream Transport) capability of the USB-C input interface.

Command Structure **GET USBC_DM**	Parameters:
Response Structure **USBC_DM < SST | MST >**	
Command Example **GET USBC_DM**	
Response Example **USBC_DM MST**	
Additional Notes:	

2.20 Volume Increment

Increase the volume of both audio output channels simultaneously.

Command Structure **SET VOLGAIN_INC**	Parameters:
Response Structure **VOLGAIN_INC < OutputVolume >**	
Command Example **SET VOLGAIN_INC**	
Response Example **VOLGAIN_INC -28**	
Additional Notes: The full adjustable volume range is divided into ten segments; changing volume from the lowest to the highest level requires calling this API nine times.	

2.21 Volume Decrease

Controls the simultaneous volume decrease of both audio output channels.

Command Structure **SET VOLGAIN_DEC**	Parameters:
Response Structure **VOLGAIN_DEC < OutputVolume >**	
Command Example **SET VOLGAIN_DEC**	
Response Example **VOLGAIN_DEC -28**	
Additional Notes: The entire adjustable volume range is divided into ten steps; to go from maximum to minimum volume you need to call this API nine times.	

3. Video Switching

3.1 Set Automatic Switching

Enable or disable the automatic switching feature.
When automatic switching is enabled, the device supports the following functions:
1. Automatic switching of video sources
2. Automatic switching of multi-view layouts
3. Fallback display of video sources

Command Structure **SET AUTOSW_FN < ON | OFF >**	Parameters: ON = Enable automatic switching
Response Structure **AUTOSW_FN < ON | OFF >**	
Command Example **SET AUTOSW_FN OFF**	
Response Example **AUTOSW_FN OFF**	
Additional Notes: Default: ON	

3.2 Get Automatic Switching Status

Retrieve the on/off status of the automatic switching function.

Command Structure **GET AUTOSW_FN**	Parameters:
Response Structure **AUTOSW_FN < ON | OFF >**	
Command Example **GET AUTOSW_FN**	
Response Example **AUTOSW_FN OFF**	
Additional Notes:	

3.3 Set Multi-View Function

Enable or disable the multi-view function

Command Structure **SET MV_ENABLE < ON | OFF >**	Parameters: ON = Enable the multi-view function
Response Structure **MV_ENABLE < ON | OFF >**	
Command Example **SET MV_ENABLE OFF**	
Response Example **MV_ENABLE OFF**	
Additional Notes: Default: ON	

3.4 Get Multi-View Function Status

Get the on/off status of the multi-view function.

Command Structure **GET MV_ENABLE**	Parameters:
Response Structure **MV_ENABLE < ON | OFF >**	
Command Example **GET MV_ENABLE**	
Response Example **MV_ENABLE OFF**	
Additional Notes:	

3.5 Switch Input to All Outputs

Switch the input to full-screen on all outputs.

Command Structure **SET SW < IN1 | IN2 | BYOD1 | BYOD2 >**	Parameters:
Response Structure **SW < IN1 | IN2 | BYOD1 | BYOD2 > OUT1** **SW < IN1 | IN2 | BYOD1 | BYOD2 > OUT2**	
Command Example **SET SW IN1**	
Response Example **SW IN1 OUT1** **SW IN1 OUT2**	
Additional Notes:	

3.6 Switch Input to an Output

Switch an input to full-screen on a specified output.
If the specified output is not connected, the device will display the input full-screen on the currently connected output.
The response message always returns information about the output where the input is actually displayed.

Command Structure **SET SW < IN1 | IN2 | BYOD1 | BYOD2 > < OUT1 | OUT2 >**	Parameters:
Response Structure **SW < IN1 | IN2 | BYOD1 | BYOD2 > < OUT1 | OUT2 >**	
Command Example **SET SW IN1 OUT1**	
Response Example **SW IN1 OUT1**	
Additional Notes: If only one output is currently connected, the second parameter is ignored and the specified input will be displayed on the single connected output.	

3.7 Display GUIDE Screen

Show the GUIDE screen on all outputs.
This means no video source is being displayed. While the GUIDE screen is shown, if the auto-standby feature is enabled, the device will enter standby automatically after the GUIDE screen has been displayed for the auto-standby timeout period.
Note: If the auto-standby timeout is set to 0, this command will cause the device to enter standby immediately.

Command Structure **SET SW GUIDE**	Parameters:
Response Structure **SW GUIDE**	
Command Example **SET SW GUIDE**	
Response Example **SW GUIDE**	
Additional Notes:	

3.8 Get Display Status

Get the display status of an output

Command Structure **GET SW < OUT1 | OUT2 | ALL >**	Parameters: ALL = Get the display status of all outputs GUIDE = The output is not showing a video source; it is displaying the Guide screen MV = The output is in multi-view mode DISCONNECTED = The output is not connected
Response Structure **SW < IN1 | IN2 | BYOD1 | BYOD2 | GUIDE | MV | DISCONNECTED > < OUT1 | OUT2 >** **或** **SW < IN1 | IN2 | BYOD1 | BYOD2 | GUIDE | MV | DISCONNECTED > OUT1** **SW < IN1 | IN2 | BYOD1 | BYOD2 | GUIDE | MV | DISCONNECTED > OUT2**	
Command Example **GET SW OUT1**	
Response Example **SW IN1 OUT1**	
Additional Notes:	

3.9 Set Current Layout

Switch the multi-view layout. The parameter is the layout name.

Command Structure **SET MV_LAYOUT < FULLSCREEN | DUALVIEW >**	Parameters:
Response Structure **MV_LAYOUT < FULLSCREEN | DUALVIEW >**	
Command Example **SET MV_LAYOUT DUALVIEW**	
Response Example **MV_LAYOUT DUALVIEW**	
Additional Notes:	

3.10 Set Current Layout with ID

Switch the multiview layout. The parameter is the layout ID in hexadecimal format.

Command Structure **SET MV_LAYOUT < 0X100 | 0X101 >**	Parameters: The four hexadecimal IDs correspond to the four layout names defined in the previous API.
Response Structure **MV_LAYOUT < 0X100 | 0X101 >**	
Command Example **SET MV_LAYOUT 0X101**	
Response Example **MV_LAYOUT 0X101**	
Additional Notes:	

3.11 Get Current Layout ID and Name

Retrieve the current multi-view layout name.

Command Structure **GET MV_LAYOUT**	Parameters:
Response Structure **MV_LAYOUT < < 0X100 FULLSCREEN > | < 0X101 DUALVIEW > >**	
Command Example **GET MV_LAYOUT**	
Response Example **MV_LAYOUT 0X101 DUALVIEW**	
Additional Notes:	

3.12 Set Layout and Video Sources

Switch multi-view layouts and assign video sources to each subwindow

Command Structure **SET MV_WIN_SRC < FULLSCREEN | DUALVIEW | CURRENT > < WinNo1 > < Src1 > < WinNo2 > < Src2 > ...** Response Structure **MV_WIN_SRC < FULLSCREEN | DUALVIEW > < WinNo1 > < Src1 > < WinNo2 > < Src2 > ...** Command Example **SET MV_WIN_SRC DUALVIEW WIN1 IN1 WIN2 IN2** Response Example **MV_WIN_SRC DUALVIEW WIN1 IN1 WIN2 IN2**	Parameters: CURRENT = keep the current layout unchanged; only change the video source for the specified window(s) WinNo = Win1 Win2 Src1 Src2... valid values are < IN1 | IN2 | BYOD1 | BYOD2 | NONE > - NONE = do not display any video source in that subwindow; the subwindow will be empty
Additional Notes:	

3.13 Get Layout and Video Sources

Retrieve the current layout and the video source for each sub-window.

Command Structure **GET MV_WIN_SRC** Response Structure **MV_WIN_SRC NONE** **或** **MV_WIN_SRC < FULLSCREEN | DUALVIEW > < WinNo1 > < Src1 > < WinNo2 > < Src2 > ...** Command Example **GET MV_WIN_SRC** Response Example **MV_WIN_SRC DUALVIEW WIN1 IN1 WIN2 IN2**	Parameters: NONE = Currently connected with dual outputs; multi-view is not supported.
Additional Notes: When connected to a single output but multi-view is disabled, this API returns normally.	

3.14 Get Video Source of Specified Window

Retrieve the video source for the specified sub-window.

Command Structure **GET MV_WIN_SRC CURRENT WinNo**	Parameters: NONE = Currently connected to dual-output; multi-view is not supported.
Response Structure **MV_WIN_SRC NONE** **或** **MV_WIN_SRC CURRENT < WinNo > < Src >**	
Command Example **GET MV_WIN_SRC CURRENT WIN2**	
Response Example **MV_WIN_SRC CURRENT WIN2 IN2**	
Additional Notes: When connected to a single output with multi-view disabled, this API returns normally. This API is a subset of the previous API and is used to support SYN-TOUCH10	

4. Audio and Other Signal Switching

4.1 Set Audio Auto-Switching

Enable or disable audio auto-switching.

Command Structure **SET AUDIO_AUTOSW_FN < ON | OFF >**	Parameters:
Response Structure **AUDIO_AUTOSW_FN < ON | OFF >**	
Command Example **SET AUDIO_AUTOSW_FN OFF**	
Response Example **AUDIO_AUTOSW_FN OFF**	
Additional Notes: Default: ON	

4.2 Get Audio Auto-Switch Status

Command Structure **GET AUDIO_AUTOSW_FN**	Parameters:
Response Structure **AUDIO_AUTOSW_FN < ON | OFF >**	
Command Example **GET AUDIO_AUTOSW_FN**	
Response Example **AUDIO_AUTOSW_FN OFF**	
Additional Notes:	

4.3 Switch to Wired Audio Source

When automatic audio switching is disabled, use this API to force the device to always output audio from a wired source.

Command Structure **SET AUDIOSW < IN1 | IN2 >**	Parameters: STOP = Currently in the Guide screen state; the device is not actually outputting audio.
Response Structure **AUDIOSW < IN1 | IN2 > [STOP]**	
Command Example **SET AUDIOSW IN2**	
Response Example **AUDIOSW IN2**	
Additional Notes:	

4.4 Switch to BYOD Audio Source

When automatic audio switching is disabled, make the device temporarily output audio from a BYOD source.
The device will play the BYOD source's audio until that BYOD source disconnects, then it will switch back to the previously configured wired audio source.
In short, this is a temporary switch.

Command Structure **SET AUDIOSW < BYOD1 | BYOD2 >**	Parameters:
Response Structure **AUDIOSW < BYOD1 | BYOD2 >**	
Command Example **SET AUDIOSW BYOD1**	
Response Example **AUDIOSW BYOD1**	
Additional Notes:	

4.5 Get Current Audio Source

Retrieve the audio source currently being output by the device. When the device is displaying the Guide screen it stops outputting audio; in that case this API returns the most recent wired audio source that was output.

Command Structure **GET AUDIOSW**	Parameters: STOP = The device is currently showing the Guide screen and is not actually outputting audio
Response Structure **AUDIOSW < IN1 | IN2 | BYOD1 | BYOD2 > [STOP]**	
Command Example **GET AUDIOSW**	
Response Example **AUDIOSW IN1**	
Additional Notes:	

4.6 Set USB Automatic Switching

Set the device to automatically switch the USB host.

Command Structure **SET USB_AUTOSW_FN < ON | OFF >**	Parameters: ON = Enables USB automatic switching
Response Structure **USB_AUTOSW_FN < ON | OFF >**	
Command Example **SET USB_AUTOSW_FN ON**	
Response Example **USB_AUTOSW_FN ON**	
Additional Notes: Default: ON	

4.7 Get USB Auto-Switch Status

Retrieve the status of the USB automatic switch.

Command Structure **GET USB_AUTOSW_FN**	Parameters:
Response Structure **USB_AUTOSW_FN < ON | OFF >**	
Command Example **GET USB_AUTOSW_FN**	
Response Example **USB_AUTOSW_FN ON**	
Additional Notes:	

4.8 Set USB Host

Manually switch the USB host

Command Structure **SET USB_HOST < USBCIN | USBHOST | WIRELESS >**	Parameters: WIRELESS = USB HOST connected to the main chipset for wireless conferencing
Response Structure **USB_HOST < USBCIN | USBHOST | WIRELESS >**	
Command Example **SET USB_HOST USBHOST1**	
Response Example **USB_HOST USBHOST1**	
Additional Notes:	

4.9 Get USB Host

Retrieve the current USB host.

Command Structure **GET USB_HOST**	Parameters:
Response Structure **USB_HOST < USBCIN | USBHOST | WIRELESS >**	
Command Example **GET USB_HOST**	
Response Example **USB_HOST USBHOST1**	
Additional Notes:	

4.10 Set VBUS Mode

Controls the VBUS signal for USB Device interfaces; provides three control modes:
Follow Host: Turn on VBUS when USB host is connected, and vice versa.
Always On: Turn on VBUS always, no matter whether USB host is connected.
Always Off: Turn off VBUS always, the USB device interface is disabled.

Command Structure **SET USB_VBUS_MODE < USBPort | ALL > < VBUSMode >**	Parameters: USBPort = Identifier for the USB Device port. Valid values: USB1, USB2, USB3. ALL means set the VBUS mode for all USB Device ports simultaneously. VBUSMode = VBUS control mode. Valid values: FOLLOW_HOST, ALWAYS_ON, ALWAYS_OFF.
Response Structure **USB_VBUS_MODE < USBPort | ALL > < VBUSMode >** **或** **USB_VBUS_MODE USB1 < VBUSMode >** **USB_VBUS_MODE USB2 < VBUSMode >** **USB_VBUS_MODE USB3 < VBUSMode >**	
Command Example **SET USB_VBUS_MODE USB2 ALWAYS_ON**	
Response Example **USB_VBUS_MODE USB2 ALWAYS_ON**	
Additional Notes: Default: FOLLOW_HOST	

4.11 Get VBUS Mode

Retrieve the VBUS control mode.

Command Structure **GET USB_VBUS_MODE < USBPort | ALL >**	Parameters: ALL = Get the VBUS control mode for all USB device ports
Response Structure **USB_VBUS_MODE < USBPort | ALL > < VBUSMode >** **或** **USB_VBUS_MODE USB1 < VBUSMode >** **USB_VBUS_MODE USB2 < VBUSMode >** **USB_VBUS_MODE USB3 < VBUSMode >**	
Command Example **GET USB_VBUS_MODE USB2**	
Response Example **USB_VBUS_MODE USB2 ALWAYS_ON**	
Additional Notes:	

5. Sleep/Standby and Peripheral Control

5.1 Set Automatic Standby

Enable or disable the device's automatic standby function.
When automatic standby is enabled, if the device continuously displays the Guide screen for a period equal to the standby timeout value, the device will send a power-off command to all connected displays and disable HDMI +5V and TMDS signals.

Command Structure **SET AUTO_STANDBY_FN < ON | OFF >**	Parameters: ON = Enable automatic standby function
Response Structure **AUTO_STANDBY_FN < ON | OFF >**	
Command Example **SET AUTO_STANDBY_FN OFF**	
Response Example **AUTO_STANDBY_FN OFF**	
Additional Notes: Default value: ON	

5.2 Get Automatic Standby

Get the device's automatic standby switch status (ON/OFF).

Command Structure **GET AUTO_STANDBY_FN**	Parameters:
Response Structure **AUTO_STANDBY_FN < ON | OFF >**	
Command Example **GET AUTO_STANDBY_FN**	
Response Example **AUTO_STANDBY_FN OFF**	
Additional Notes:	

5.3 Set Automatic Standby Timeout

Set the automatic standby timeout value.

Command Structure **SET AUTO_STANDBY_D < TimeOut >**	Parameters: TimeOut range: 0-3600 seconds
Response Structure **AUTO_STANDBY_D < TimeOut >**	
Command Example **SET AUTO_STANDBY_D 100**	
Response Example **AUTO_STANDBY_D 100**	
Additional Notes: Default value: 120 (seconds)	

5.4 Get Auto Standby Timeout

Retrieve the auto standby timeout value

Command Structure **GET AUTO_STANDBY_D**	Parameters:
Response Structure **AUTO_STANDBY_D < TimeOut >**	
Command Example **GET AUTO_STANDBY_D**	
Response Example **AUTO_STANDBY_D 100**	
Additional Notes:	

5.5 Set Device Power On or Standby

Manually control the device power state: turn on or set to standby.

Command Structure **SET STANDBY < ON | OFF >**	Parameters: ON = Power on OFF = Standby
Response Structure **STANDBY < ON | OFF >**	
Command Example **SET STANDBY ON**	
Response Example **STANDBY ON**	
Additional Notes:	

5.6 Get Device Standby Status

Retrieve whether the device is powered on.

Command Structure **GET STANDBY**	Parameters:
Response Structure **STANDBY < ON | OFF >**	
Command Example **GET STANDBY**	
Response Example **STANDBY ON**	
Additional Notes:	

5.7 Send CEC Power On/Off Command

Send a CEC power-on or power-off command to the specified output interface.

Command Structure **SET CEC_PWR < OUT1 | OUT2 > < ON | OFF >**	Parameters:
Response Structure **CEC_PWR < OUT1 | OUT2 > < ON | OFF >**	
Command Example **SET CEC_PWR OUT1 ON**	
Response Example **CEC_PWR OUT1 ON**	
Additional Notes:	

5.8 Set CEC Power On/Off Command

Set the CEC power-on or power-off command for a specified output interface.

Command Structure **SET CECCMD_EDIT < OUT1 | OUT2 > < ON | OFF > < CECCode >**	Parameters: The parameter CECCode is the actual CEC message in hexadecimal format with no spaces between adjacent bytes.
Response Structure **CECCMD_EDIT < OUT1 | OUT2 > < ON | OFF > < CECCode >**	
Command Example **SET CECCMD_EDIT OUT1 ON 4004**	
Response Example **CECCMD_EDIT OUT1 ON 4004**	
Additional Notes: Default values: Power on: 4004 Power off: FF36	

5.9 Get CEC Power On/Off Command

Retrieve the CEC power-on or power-off command for the specified output interface.

Command Structure **GET CECCMD_EDIT < OUT1 | OUT2 > < ON | OFF >**	Parameters:
Response Structure **CECCMD_EDIT < OUT1 | OUT2 > < ON | OFF > < CECCode >**	
Command Example **GET CECCMD_EDIT OUT1 ON**	
Response Example **CECCMD_EDIT OUT1 ON 4004**	
Additional Notes:	

5.10 Get CEC Commands

Get the CEC power-on and power-off commands for the specified output interface.

Command Structure **GET CECCMD_EDIT < OUT1 | OUT2 >**	Parameters:
Response Structure **CECCMD_EDIT < OUT1 | OUT2 > ON < CECCode >** **CECCMD_EDIT < OUT1 | OUT2 > OFF < CECCode >**	
Command Example **GET CECCMD_EDIT OUT1**	
Response Example **CECCMD_EDIT OUT1 ON 4004** **CECCMD_EDIT OUT1 OFF FF36**	
Additional Notes:	

5.11 Get All CEC Commands

Retrieve the CEC power-on and power-off commands for all output interfaces.

Command Structure **GET CECCMD_EDIT ALL**	Parameters:
Response Structure **CECCMD_EDIT OUT1 ON < CECCode >** **CECCMD_EDIT OUT1 OFF < CECCode >** **CECCMD_EDIT OUT2 ON < CECCode >** **CECCMD_EDIT OUT2 OFF < CECCode >**	
Command Example **GET CECCMD_EDIT ALL**	
Response Example **CECCMD_EDIT OUT1 ON 4004** **CECCMD_EDIT OUT1 OFF FF36** **CECCMD_EDIT OUT2 ON 4004** **CECCMD_EDIT OUT2 OFF FF36**	
Additional Notes:	

5.12 Send CEC Command String

Directly send a CEC command string to a specified output interface.

Command Structure **SET CEC_CMD < OUT1 | OUT2 > < CECCode >**	Parameters:
Response Structure **CEC_CMD < OUT1 | OUT2 > < CECCode >**	
Command Example **SET CEC_CMD OUT1 4004**	
Response Example **CEC_CMD OUT1 4004**	
Additional Notes:	

5.13 Set UART Parameters

Set the UART's four parameters, including the baud rate.

Command Structure **SET UART_CFG < BaudRate > < Parity > < DataBit > < StopBit >**	Parameters: BaudRate is the baud rate. Valid values: 9600, 19200, 38400, 57600, 115200 Parity indicates parity. Valid values: NONE, ODD, EVEN DataBit is the number of data bits. Valid values: 7, 8 StopBit is the number of stop bits. Valid values: 1, 2
Response Structure **UART_CFG < BaudRate > < Parity > < DataBit > < StopBit >**	
Command Example **SET UART_CFG 115200 NONE 8 1**	
Response Example **UART_CFG 115200 NONE 8 1**	
Additional Notes: Default: 115200 NONE 8 1	

5.14 Get UART Parameters

Retrieve UART parameters such as baud rate, parity, data bits, and stop bits.

Command Structure **GET UART_CFG**	Parameters:
Response Structure **UART_CFG < BaudRate > < Parity > < DataBit > < StopBit >**	
Command Example **GET UART_CFG**	
Response Example **UART_CFG 115200 NONE 8 1**	
Additional Notes:	

5.15 Set UART Baud Rate

Sets the baud rate for the serial (UART) port.

Command Structure **SET UART_B < BaudRate >**	Parameters:
Response Structure **UART_B < BaudRate >**	
Command Example **SET UART_B 115200**	
Response Example **UART_B 115200**	
Additional Notes:	

5.16 Get UART Baud Rate

Get the UART baud rate parameter.

Command Structure **GET UART_B**	Parameters:
Response Structure **UART_B < BaudRate >**	
Command Example **GET UART_B**	
Response Example **UART_B 115200**	
Additional Notes:	

5.17 Set UART Command

Configure a UART (serial) command.

Command Structure **SET UART_CMD < CmdName > < HEX | STR > < CmdStr >**	Parameters: CmdName = The command name. Besides ON and OFF, it can be other custom names (e.g., VolumeUp). HEX = Indicates the command is in hexadecimal format. STR = Indicates the command is in string format. CmdStr = The command string. For hexadecimal strings, adjacent bytes do not require spaces between them.
Response Structure **UART_CMD < CmdName > < HEX | STR > < CmdStr >**	
Command Example **SET UART_CMD ON STR POWERON**	
Response Example **UART_CMD ON STR POWERON**	
Additional Notes: By default, no UART commands are defined.	

5.18 Delete UART Command

Delete a UART (serial) command.

Command Structure **SET UART_CMD_D < CmdName >**	Parameters:
Response Structure **UART_CMD_D < CmdName >**	
Command Example **SET UART_CMD_D ON**	
Response Example **UART_CMD_D ON**	
Additional Notes:	

5.19 Get UART Commands

Retrieve one or all UART (serial) commands.

Command Structure **GET UART_CMD < CmdName | ALL >**	Parameters:
Response Structure **UART_CMD < CmdName > < HEX | STR > < CmdStr >** **或** **UART_CMD < CmdName1 > < HEX | STR > < CmdStr1 >** **UART_CMD < CmdName2 > < HEX | STR > < CmdStr2 >** **...**	
Command Example **GET UART_CMD ON**	
Response Example **UART_CMD ON STR POWERON**	
Additional Notes:	

5.20 Send UART Command

Send a previously defined UART (serial) command.

Command Structure **SET UART_CMD_S < CmdName >**	Parameters:
Response Structure **UART_CMD_S < CmdName >**	
Command Example **SET UART_CMD_S ON**	
Response Example **UART_CMD_S ON**	
Additional Notes:	

5.21 Send UART Command String

Directly send UART data

Command Structure **SET UART_S < HEX | STR > < CmdStr >**	Parameters:
Response Structure **UART_S < HEX | STR > < CmdStr >**	
Command Example **SET UART_S STR POWERON**	
Response Example **UART_S STR POWERON**	
Additional Notes:	

5.22 Set UART Power On/Off Function

Configure whether to send serial (UART) commands on power on/off.
When the UART is in external-control mode, if this option is enabled, the device will send the corresponding commands over the UART whenever it automatically or manually enters or exits standby.

Command Structure **SET UARTPWR_FN < ON | OFF >**	Parameters:
Response Structure **UARTPWR_FN < ON | OFF >**	
Command Example **SET UARTPWR_FN ON**	
Response Example **UARTPWR_FN ON**	
Additional Notes: Default: OFF	

5.23 Get UART Power Control Function

Retrieves whether the UART participates in sending power on/off commands.

Command Structure **GET UARTPWR_FN**	Parameters:
Response Structure **UARTPWR_FN < ON | OFF >**	
Command Example **GET UARTPWR_FN**	
Response Example **UARTPWR_FN ON**	
Additional Notes:	

5.24 Send CEC and/or RS232 Commands

Send CEC and/or RS232 (serial) commands.

Command Structure **SET SEND_CMD < ON | OFF > < RS232 | CEC | ALL >**	Parameters: RS232 = Send only RS232 (serial) commands CEC = Send only CEC commands ALL = Send both RS232 and CEC commands
Response Structure **SEND_CMD < ON | OFF > < RS232 | CEC | ALL >**	
Command Example **SET SEND_CMD ON CEC**	
Response Example **SEND_CMD ON CEC**	
Additional Notes:	

5.25 Set UART Operating Mode

The serial port can be used for external control or to provide API services (i.e., accept commands). This API configures that mode.

Command Structure **SET UART_MODE < API | COM >**	Parameters: API = The UART is used to provide API services externally; in this mode, earlier API functions that involve sending actual serial data will be disabled. COM = The UART is used for external control
Response Structure **UART_MODE < API | COM >**	
Command Example **SET UART_MODE API**	
Response Example **UART_MODE API**	
Additional Notes: Default: COM	

5.26 Get UART Operating Mode

Retrieve the UART (serial port) operating mode.

Command Structure **GET UART_MODE**	Parameters:
Response Structure **UART_MODE < API | COM >**	
Command Example **GET UART_MODE**	
Response Example **UART_MODE API**	
Additional Notes:	

5.27 Get MUTE Input

Get the status of the MUTE input.

Command Structure **GET MUTE_CONTROL**	Parameters: HIGH: The MUTE input is shorted; mute has been forced on.
Response Structure **MUTE_CONTROL < HIGH | LOW >**	
Command Example **GET MUTE_CONTROL**	
Response Example **MUTE_CONTROL LOW**	
Additional Notes:	

5.28 Set CEC Power On/Off Function

Configure whether to send CEC commands on power on/off.
If this setting is enabled, when the device automatically or manually enters or exits standby, it will send the corresponding CEC commands to the display device.

Command Structure **SET CECPWR_FN < ON | OFF >**	Parameters:
Response Structure **CECPWR_FN < ON | OFF >**	
Command Example **SET CECPWR_FN OFF**	
Response Example **CECPWR_FN OFF**	
Additional Notes: Default: ON	

5.29 Get CEC Power On/Off Function

Retrieve whether a CEC command is sent when powering on or off.

Command Structure **GET CECPWR_FN**	Parameters:
Response Structure **CECPWR_FN < ON | OFF >**	
Command Example **GET CECPWR_FN**	
Response Example **CECPWR_FN OFF**	
Additional Notes:	

6. Network and BYOD Related Settings

6.1 Set Network Isolation

When network isolation mode is enabled, the device's network ports are split into two independent VLANs, which can be used to connect to different networks. This API sets whether network isolation is enabled.

Command Structure **SET NET_ISOLATION < ON | OFF >**	Parameters: ON = Enable network isolation
Response Structure **NET_ISOLATION < ON | OFF >**	
Command Example **SET NET_ISOLATION ON**	
Response Example **NET_ISOLATION ON**	
Additional Notes: Default: OFF	

6.2 Get Network Isolation Settings

Retrieve whether network isolation mode is enabled

Command Structure **GET NET_ISOLATION**	Parameters:
Response Structure **NET_ISOLATION < ON | OFF >**	
Command Example **GET NET_ISOLATION**	
Response Example **NET_ISOLATION ON**	
Additional Notes:	

6.3 Set IP Address - DHCP

Set the IP address to use DHCP mode

Command Structure **SET IPADDR < CONTROL | SERVICE > DHCP**	Parameters: CONTROL = Set the IP address for the main or control network SERVICE = Set the IP address for the service network in network isolation mode
Response Structure **IPADDR < CONTROL | SERVICE > DHCP**	
Command Example **SET IPADDR CONTROL DHCP**	
Response Example **IPADDR CONTROL DHCP**	
Additional Notes: By default, both networks use DHCP mode	

6.4 Set IP Address - Static

Set a static IP address.

Command Structure **SET IPADDR < CONTROL | SERVICE > STATIC < IPAddress > < NetMask > [Gateway [DNSServer]]**	Parameters: Gateway and DNS server are optional parameters.
Response Structure **IPADDR < CONTROL | SERVICE > STATIC < IPAddress > < NetMask > [Gateway [DNSServer]]**	
Command Example **SET IPADDR SERVICE STATIC 192.168.1.34 255.255.255.0 192.168.1.1 192.168.1.2**	
Response Example **IPADDR SERVICE STATIC 192.168.1.34 255.255.255.0 192.168.1.1 192.168.1.2**	
Additional Notes:	

6.5 Get IP Address

Retrieve the device's IP address.

Command Structure **GET IPADDR < CONTROL | SERVICE | ALL >**	Parameters: ALL = Get the IP addresses of both network ports
Response Structure **IPADDR < CONTROL | SERVICE > < DHCP | STATIC > < IPAddress > < NetMask > [Gateway [DNSServer]]** **或** **IPADDR CONTROL < DHCP | STATIC > < IPAddress > < NetMask > [Gateway [DNSServer]]** **IPADDR SERVICE < DHCP | STATIC > < IPAddress > < NetMask > [Gateway [DNSServer]]**	
Command Example **GET IPADDR SERVICE**	
Response Example **IPADDR SERVICE STATIC 192.168.1.34 255.255.255.0 192.168.1.1 192.168.1.2**	
Additional Notes:	

6.6 Set USB-C Network Adapter Function

The network adapter on the USB-C port can be enabled or disabled. This API is used to control that feature.

Command Structure **SET USBNIC < IN1 | IN2 > < ON | OFF >**	Parameters: ON = Enable the USB port's network adapter function
Response Structure **USBNIC < IN1 | IN2 > < ON | OFF >**	
Command Example **SET USBNIC IN1 OFF**	
Response Example **USBNIC IN1 OFF**	
Additional Notes: Default: ON	

6.7 Get USB Network Adapter On/Off Status

Get the on/off status of the USB-C network adapter.

Command Structure **GET USBNIC < IN1 | IN2 | ALL >**	Parameters: ALL = Get the on/off status of both USB network interfaces IN1 = USB-C IN interface IN2 = USB HOST interface
Response Structure **USBNIC < IN1 | IN2 > < ON | OFF >** **或** **USBNIC IN1 < ON | OFF >** **USBNIC IN2 < ON | OFF >**	
Command Example **GET USBNIC IN1**	
Response Example **USBNIC IN1 OFF**	
Additional Notes:	

6.8 Configure HTTPS Functionality

Command Structure **SET HTTPS < ON | OFF >**	Parameters: ON = Enable HTTPS
Response Structure **HTTPS < ON | OFF >**	
Command Example **SET HTTPS ON**	
Response Example **HTTPS ON**	
Additional Notes: Default: ON	

6.9 Get HTTPS Switch Status

Command Structure **GET HTTPS**	Parameters:
Response Structure **HTTPS < ON | OFF >**	
Command Example **GET HTTPS**	
Response Example **HTTPS ON**	
Additional Notes:	

6.10 Configure Telnet Function

Command Structure **SET TELNET < ON | OFF >**	Parameters: ON = Enable Telnet
Response Structure **TELNET < ON | OFF >**	
Command Example **SET TELNET ON**	
Response Example **TELNET ON**	
Additional Notes: Default: ON	

6.11 Get Telnet Feature Switch Status

Command Structure **GET TELNET**	Parameters:
Response Structure **TELNET < ON | OFF >**	
Command Example **GET TELNET**	
Response Example **TELNET ON**	
Additional Notes:	

6.12 Set Wi-Fi Operating Mode

Set the Wi-Fi module's frequency band and channel

Command Structure **SET WIFI_MODE < WiFiBand > < WiFiChannel | AUTO >**	Parameters: WiFiBand = 2 5, corresponds to the 2.4 GHz and 5 GHz bands respectively WiFiChannel is the Wi-Fi channel; valid values depend on the band: 2.4 GHz band: 1 2 3 4 5 6 7 8 9 10 11 5 GHz band: 36 40 44 48 149 153 157 161 AUTO = The device automatically selects a suitable channel
Response Structure **WIFI_MODE < WiFiBand > < WiFiChannel > [AUTO]**	
Command Example **SET WIFI_MODE 5 AUTO**	
Response Example **WIFI_MODE 5 161 AUTO**	
Additional Notes: Default: 5 AUTO That is, 5 GHz band with automatic channel selection	

6.13 Get Wi-Fi Operating Mode

Command Structure **GET WIFI_MODE**	Parameters:
Response Structure **WIFI_MODE < WiFiBand > < WiFiChannel > [AUTO]**	
Command Example **GET WIFI_MODE**	
Response Example **WIFI_MODE 5 161 AUTO**	
Additional Notes:	

6.14 Set Wi-Fi Password

Set the password for the built-in soft AP used for wireless screen casting.

Command Structure **SET SOFTAP_PWD < WiFiPassword >**	Parameters: The parameter WiFiPassword must be 8–20 characters long and may contain only letters, digits, and the special characters '_' and '-'.
Response Structure **SOFTAP_PWD < WiFiPassword >**	
Command Example **SET SOFTAP_PWD 88888888**	
Response Example **SOFTAP_PWD 88888888**	
Additional Notes: Default value: 12345678	

6.15 Get Wi-Fi Password

Retrieve the password for the built-in soft AP.

Command Structure **GET SOFTAP_PWD**	Parameters:
Response Structure **SOFTAP_PWD < WiFiPassword >**	
Command Example **GET SOFTAP_PWD**	
Response Example **SOFTAP_PWD 88888888**	
Additional Notes:	

6.16 Configure Soft AP Function

The built-in Soft AP can be enabled or disabled; this API controls that. From firmware 1.2.1, external Wi-Fi AP is supported. The functionality of this API is covered by the API in chapter 7.1. Please use the API in that chapter to configure the presentation switcher correctly.

Command Structure **SET SOFTAP_EN < ON | OFF >**	Parameters: ON = Enable Soft AP
Response Structure **SOFTAP_EN < ON | OFF >**	
Command Example **SET SOFTAP_EN OFF**	
Response Example **SOFTAP_EN OFF**	
Additional Notes: Default: ON	

6.17 Get Soft AP Function Switch Status

Starting with firmware 1.2.1, external Wi-Fi access point (AP) is supported. The function of this API is also covered by the API in chapter 7.2. Use the API in that chapter to correctly configure the presentation switcher.

Command Structure **GET SOFTAP_EN**	Parameters:
Response Structure **SOFTAP_EN < ON | OFF >**	
Command Example **GET SOFTAP_EN**	
Response Example **SOFTAP_EN OFF**	
Additional Notes:	

6.18 Set SoftAP Router Function

The device supports soft routing, which lets devices connected to the soft AP access network resources on the device's wired Ethernet port. If that network has Internet access, devices connected to the soft AP can also access the Internet.
This API is used to control that function.

Command Structure **SET SOFTAP_ROUTER < ON | OFF >**	Parameters: ON = Enable soft router
Response Structure **SOFTAP_ROUTER < ON | OFF >**	
Command Example **SET SOFTAP_ROUTER OFF**	
Response Example **SOFTAP_ROUTER OFF**	
Additional Notes: Default: ON	

6.19 Get Soft Router Feature Switch Status

Command Structure **GET SOFTAP_ROUTER**	Parameters:
Response Structure **SOFTAP_ROUTER < ON | OFF >**	
Command Example **GET SOFTAP_ROUTER**	
Response Example **SOFTAP_ROUTER OFF**	
Additional Notes:	

6.20 Configure BYOD Feature

Controls the Miracast and AirPlay receiver functions by enabling or disabling them.

Command Structure **SET BYOD_EN < ON | OFF >**	Parameters: ON = Enable BYOD receiver function
Response Structure **BYOD_EN < ON | OFF >**	
Command Example **SET BYOD_EN OFF**	
Response Example **BYOD_EN OFF**	
Additional Notes: Default value: ON	

6.21 Get BYOD Feature

Command Structure **GET BYOD_EN**	Parameters:
Response Structure **BYOD_EN < ON | OFF >**	
Command Example **GET BYOD_EN**	
Response Example **BYOD_EN OFF**	
Additional Notes:	

6.22 Set BYOD Access Code

When casting via Miracast or AirPlay, you can require users to enter an access code to prevent interference.
This API controls that feature.

Command Structure **SET BYOD_ACCESS_CODE < AccessCode | AUTO | NONE >** Response Structure **BYOD_ACCESS_CODE < AccessCode | AUTO | NONE >** Command Example **SET BYOD_ACCESS_CODE 1234** Response Example **BYOD_ACCESS_CODE 1234**	Parameters: AccessCode = Access code; must be a 4-digit number AUTO = Automatically generate an access code that changes dynamically NONE = No access code
Additional Notes: Default: NONE	

6.23 Get BYOD Access Code

Command Structure **GET BYOD_ACCESS_CODE** Response Structure **BYOD_ACCESS_CODE NONE** **或** **BYOD_ACCESS_CODE < AccessCode > [AUTO]** Command Example **GET BYOD_ACCESS_CODE** Response Example **BYOD_ACCESS_CODE 1234 AUTO**	Parameters: Optional parameter AUTO indicates that the current access code is dynamically generated.
Additional Notes:	

6.24 Set SSID Broadcast

Normally a Soft AP periodically sends broadcast frames containing its SSID. Disabling this feature can improve security to some extent.
This API is used to control that feature.

Command Structure **SET SSID_BROADCAST_FN < ON | OFF >**	Parameters: ON = Enable SSID broadcast
Response Structure **SSID_BROADCAST_FN < ON | OFF >**	
Command Example **SET SSID_BROADCAST_FN OFF**	
Response Example **SSID_BROADCAST_FN OFF**	
Additional Notes: Default: ON	

6.25 Get SSID Broadcast Switch Status

Command Structure **GET SSID_BROADCAST_FN**	Parameters: SSID_BROADCAST_FN: Indicates whether SSID broadcast (the wireless network name) is enabled. ON — SSID broadcast is enabled (network name is visible). OFF — SSID broadcast is disabled (network name is hidden).
Response Structure **SSID_BROADCAST_FN < ON | OFF >**	
Command Example **GET SSID_BROADCAST_FN**	
Response Example **SSID_BROADCAST_FN OFF**	
Additional Notes:	

6.26 Set Forced Transcoding for Non-Flow-Control Cameras

Some USB cameras support MJPEG but do not support flow control. Forwarding their raw MJPEG stream in wireless meetings can exceed the capacity of the wireless link. In such cases, the stream can be transcoded — re-encoded to a lower-bandwidth MJPEG stream. This API controls whether transcoding is always enabled for such cameras.

Command Structure **SET WCONF_MJPEG_NO_FLOWCTRL_TRANSCODE < ON | OFF >**	Parameters:
Response Structure **WCONF_MJPEG_NO_FLOWCTRL_TRANSCODE < ON | OFF >**	
Command Example **SET WCONF_MJPEG_NO_FLOWCTRL_TRANSCODE ON**	
Response Example **WCONF_MJPEG_NO_FLOWCTRL_TRANSCODE ON**	
Additional Notes: Default: OFF	

6.27 Get Forced Transcoding for Non-Flow-Control Cameras

Command Structure **GET WCONF_MJPEG_NO_FLOWCTRL_TRANSCODE**	Parameters:
Response Structure **WCONF_MJPEG_NO_FLOWCTRL_TRANSCODE < ON | OFF >**	
Command Example **GET WCONF_MJPEG_NO_FLOWCTRL_TRANSCODE**	
Response Example **WCONF_MJPEG_NO_FLOWCTRL_TRANSCODE ON**	
Additional Notes:	

6.28 Configure SSH

Command Structure **SET SSH < ON | OFF >**	Parameters: ON = enable SSH
Response Structure **SSH < ON | OFF >**	
Command Example **SET SSH ON**	
Response Example **SSH ON**	
Additional Notes: Default: OFF	

6.29 Get SSH Switch Status

Command Structure **GET SSH**	Parameters:
Response Structure **SSH < ON | OFF >**	
Command Example **GET SSH**	
Response Example **SSH ON**	
Additional Notes:	

6.30 Set Telnet Login Authentication

Command Structure **SET TELNET_AUTH_FN < ON | OFF >**	Parameters: ON = Enable Telnet login authentication
Response Structure **TELNET_AUTH_FN < ON | OFF >**	
Command Example **SET TELNET_AUTH_FN ON**	
Response Example **TELNET_AUTH_FN ON**	
Additional Notes: Default: OFF	

6.31 Get Telnet Login Authentication Feature Status

Command Structure **GET TELNET_AUTH_FN**	Parameters:
Response Structure **TELNET_AUTH_FN < ON | OFF >**	
Command Example **GET TELNET_AUTH_FN**	
Response Example **TELNET_AUTH_FN ON**	
Additional Notes:	

6.32 Set API Login Authentication Password

This password is used by both the Telnet and SSH services. SSH always requires authentication; Telnet login authentication is controlled by the configuration option above.

Command Structure **SET API_PWD < OldPwd > < NewPwd >** Response Structure **API_PWD < OldPwd > < NewPwd >** Command Example **SET API_PWD Wyrestorm 12345678** Response Example **API_PWD Wyrestorm 12345678**	Parameters: Password length 8-24 characters. Allowed characters: A-Z a-z 0-9 !@#$%^&*()_+-=[]{};:'"|,.<>/? Spaces, non-ASCII characters, and any characters not listed are not allowed.
Additional Notes: Default: Wyrestorm	

7. AP / Infrastructure Mode

7.1 Set Wireless Network Mode

Supports three wireless network modes:
1. Wireless disabled — no wireless connections supported.
2. Access Point (AP) mode — uses the device's built-in Wi-Fi module to provide wireless connections.
3. Infrastructure mode — uses the built-in Wi-Fi module for Miracast; uses an external Wi-Fi AP for AirPlay and Dongle.

Command Structure **SET WIRELESS_NETWORK_MODE < OFF | AP | INFRASTRUCTURE >** Response Structure **WIRELESS_NETWORK_MODE < OFF | AP | INFRASTRUCTURE >** Command Example **SET WIRELESS_NETWORK_MODE INFRASTRUCTURE** Response Example **WIRELESS_NETWORK_MODE INFRASTRUCTURE**	Parameters: OFF = Wireless disabled AP = Access Point mode INFRASTRUCTURE = Infrastructure mode
Additional Notes: Default value: AP	

7.2 Get Wireless Network Mode

Command Structure **GET WIRELESS_NETWORK_MODE**	Parameters:
Response Structure **WIRELESS_NETWORK_MODE < OFF | AP | INFRASTRUCTURE >**	
Command Example **GET WIRELESS_NETWORK_MODE**	
Response Example **WIRELESS_NETWORK_MODE INFRASTRUCTURE**	
Additional Notes:	

7.3 Configure External Wi-Fi

When the wireless network mode is set to Infrastructure mode, this API is used to configure the external Wi-Fi SSID and password. The information will be passed to the Dongle during the pairing process.

Command Structure **SET DONGLE_CONN_AP < SSID > < Password >**	Parameters: SSID = SSID of the external Wi-Fi, UTF-8 encoded; if it contains spaces, enclose it in double quotes. Password = Password for the external Wi-Fi
Response Structure **DONGLE_CONN_AP < SSID > < Password >**	
Command Example **SET DONGLE_CONN_AP OfficeBYOD 12345678**	
Response Example **DONGLE_CONN_AP OfficeBYOD 12345678**	
Additional Notes:	

7.4 Get External Wi-Fi

Command Structure **GET DONGLE_CONN_AP**	Parameters:
Response Structure **DONGLE_CONN_AP < SSID > < Password >**	
Command Example **GET DONGLE_CONN_AP**	
Response Example **DONGLE_CONN_AP OfficeBYOD 12345678**	
Additional Notes:	

8. Wireless Multi Camera Settings

8.1 Set Camera Alias

Set an alias for a camera, used for on-screen display (OSD) prompts in wireless conferencing.

Command Structure **SET CAMERA_ALIAS < camera_id > < camera_alias >** Response Structure **CAMERA_ALIAS < camera_id > < camera_alias >** Command Example **SET CAMERA_ALIAS 1 Teacher** Response Example **CAMERA_ALIAS 1 Teacher**	Parameters: camera_id = Camera number, matching the silkscreen on the USB DEVICE interface; valid values are 1, 2, 3. camera_alias = Camera alias, 1-31 characters; allowed characters are uppercase and lowercase letters, digits, spaces, underscores, and hyphens.
Additional Notes: Default values: Camera 1 Camera 2 Camera 3	

8.2 Get Camera Aliases

Command Structure **GET CAMERA_ALIAS < camera_id | ALL >** Response Structure **CAMERA_ALIAS < camera_id > < camera_alias >** 或 **CAMERA_ALIAS 1 < camera_alias-1 >** **CAMERA_ALIAS 2 < camera_alias-2 >** **CAMERA_ALIAS 3 < camera_alias-3 >** Command Example **GET CAMERA_ALIAS 1** Response Example **CAMERA_ALIAS 1 Teacher**	Parameters: ALL = Retrieve aliases for all cameras
Additional Notes:	

8.3 Set Camera Priority

Set priorities for the cameras. At the start of a wireless meeting, the camera with the highest priority is used; during the meeting the system switches cameras in order from highest to lowest priority. If the current camera has the lowest priority, it will switch to the highest-priority camera.

Command Structure **SET CAMERA_PRIORITY < camera_id-1 > [camera_id-2 [camera_id-3]]**	Parameters: camera_id = camera number, matching the silkscreen on the USB DEVICE interface; valid values are 1, 2, 3. List camera IDs in order from highest to lowest priority. If fewer than three parameters are provided, any cameras not mentioned are disabled (DISABLED state).
Response Structure **CAMERA_PRIORITY < camera_id-1 > [camera_id-2 [camera_id-3]]**	
Command Example **SET CAMERA_PRIORITY 2 1 3**	
Response Example **CAMERA_PRIORITY 2 1 3**	
Additional Notes: Default: 1 2 3 Camera priorities are determined by USB port numbering, ordered from high to low.	

8.4 Get Camera Priority

Command Structure **GET CAMERA_PRIORITY**	Parameters:
Response Structure **CAMERA_PRIORITY < camera_id-1 > [camera_id-2 [camera_id-3]]**	
Command Example **GET CAMERA_PRIORITY**	
Response Example **CAMERA_PRIORITY 2 1 3**	
Additional Notes:	

8.5 Get Camera Status

Command Structure **GET CAMERA_STATUS < camera_id | ALL >** Response Structure **CAMERA_STATUS < camera_id > < camera_status >** **或** **CAMERA_STATUS 1 < camera_status-1 >** **CAMERA_STATUS 2 < camera_status-2 >** **CAMERA_STATUS 3 < camera_status-3 >** Command Example **GET CAMERA_STATUS 3** Response Example **CAMERA_STATUS 3 DISABLED**	Parameters: ALL = Get the status of all cameras camera_id = Camera number, matching the silkscreen on the USB DEVICE port; valid values: 1, 2, 3. camera-status = Camera status. Possible values and meanings: UNKNOWN USB port is not connected to the main chip or the status cannot be retrieved DISABLED The USB camera is disabled DISCONNECTED No camera is connected to the port CONNECTED A camera is connected to the port ACTIVE The camera on the port is currently being used for a wireless conference
Additional Notes:	

9. Wireless Multi Audio Device Settings

9.1 Set Wireless Conference Audio Device Selection Mode

When multiple audio devices are connected, a wireless conference can use only one. Two selection modes are supported:
AUTO: The device selects the audio device using built-in rules, preferring a standalone speakerphone device.
MANUAL: The user specifies the audio device.

Command Structure **SET WCONF_AUDIO_DEV_SELECT_MODE < mode >** Response Structure **WCONF_AUDIO_DEV_SELECT_MODE < mode >** Command Example **SET WCONF_AUDIO_DEV_SELECT_MODE MANUAL** Response Example **WCONF_AUDIO_DEV_SELECT_MODE MANUAL**	Parameters: mode = selection mode. Possible values and meanings: AUTO Automatic MANUAL Manual
Additional Notes: Default: AUTO	

9.2 Get Wireless Conference Audio Device Selection Mode

Command Structure **GET WCONF_AUDIO_DEV_SELECT_MODE**	Parameters:
Response Structure **WCONF_AUDIO_DEV_SELECT_MODE < mode >**	
Command Example **GET WCONF_AUDIO_DEV_SELECT_MODE**	
Response Example **WCONF_AUDIO_DEV_SELECT_MODE MANUAL**	
Additional Notes:	

9.3 Manual Selection of Wireless Conference Audio Device

Manually select the audio device used for a wireless conference, choosing separately for input and output.

Command Structure **SET WCONF_AUDIO_DEV_PORT < direction > < usb_port >**	Parameters: direction = Direction of the audio device; possible values and meanings: INPUT Input, e.g., microphone OUTPUT Output, e.g., speaker usb_port = USB port number, valid values: 1, 2, 3. To ensure proper audio operation, select the port where the corresponding audio device is actually connected.
Response Structure **WCONF_AUDIO_DEV_PORT < direction > < usb_port >**	
Command Example **SET WCONF_AUDIO_DEV_PORT INPUT 1**	
Response Example **WCONF_AUDIO_DEV_PORT INPUT 1**	
Additional Notes: Default: The device on USB port 1 is used for both input and output.	

9.4 Get Manually Selected Audio Device for Wireless Conference

Command Structure **GET WCONF_AUDIO_DEV_PORT < direction >**	Parameters:
Response Structure **WCONF_AUDIO_DEV_PORT < direction > < usb_port >**	
Command Example **GET WCONF_AUDIO_DEV_PORT INPUT**	
Response Example **WCONF_AUDIO_DEV_PORT INPUT 1**	
Additional Notes:	

9.5 Get USB Port Audio Device Status

Command Structure
GET USB_DEV_STATUS < devtype > < usb_port > < direction >

Response Structure
USB_DEV_STATUS < devtype > < usb_port > < direction > < status >

Command Example
```
GET USB_DEV_NAME AUDIO 1 INPUT
```

Response Example
```
USB_DEV_NAME AUDIO 1 INPUT NOAUDIO
```

Parameters:
devtype = Device type. Currently only supported value: AUDIO
usb_port = USB port number. Valid values: 1, 2, 3.
direction = Direction of the audio device. Values and meanings:
INPUT Input, e.g., microphone
OUTPUT Output, e.g., speaker
status = Device status. Values and meanings:
UNKNOWN USB port is not connected to the main chip or status cannot be obtained
DISCONNECTED No device connected to the USB port
NOAUDIO A USB device is connected, but no audio device was found
CONNECTED An audio device is connected but not used by the wireless conferencing function
ACTIVE The connected audio device is being used by the wireless conferencing function

Additional Notes:

9.6 Get USB Port Audio Device Name

Command Structure
GET USB_DEV_NAME < devtype> < usb_port > < direction >

Response Structure
USB_DEV_NAME < devtype> < usb_port > < direction > < usbdevname >

Command Example
```
GET USB_DEV_NAME AUDIO 1 INPUT
```

Response Example
```
USB_DEV_NAME AUDIO 1 INPUT Microphone
```

Parameters:
devtype = Device type; currently only supported value is AUDIO
usb_port = USB port number; valid values are 1, 2, 3.
direction = Direction of the audio device; values and meanings:
INPUT Input, e.g., microphone
OUTPUT Output, e.g., speaker
usbdevname = Device name, up to 64 bytes; if no audio device is connected to the port, an empty string is returned

Additional Notes:

9.7 Get Active USB Audio Device

Command Structure **GET WCONF_AUDIO_DEV_ACTIVE < direction >** Response Structure **WCONF_AUDIO_DEV_ACTIVE < direction > < usb_port | NONE >** Command Example **GET WCONF_AUDIO_DEV_ACTIVE INPUT** Response Example **WCONF_AUDIO_DEV_ACTIVE INPUT 1**	Parameters: direction = Direction of the audio device. Possible values and meanings are: INPUT Input, e.g., microphone OUTPUT Output, e.g., speaker usb_port = USB port number, possible values: 1, 2, 3. If the response is NONE, it means no USB audio device is being used for the specified direction.
Additional Notes:	

Publication Disclaimer:

The material contained in this document consists of information that is the sole property of WyreStorm. This document is intended to provide information to allow interfacing to the relevant WyreStorm equipment by third party products

WYRESTORM IS NOT RESPONSIBLE FOR MALFUNCTIONS AND/OR THE IN-OPERABILITY WHICH MAY BE CAUSED BY THE APPLICATION OF THIS INFORMATION, WHETER EXPECTED OR NOT.

WyreStorm reserves the right to change software, control codes and specifications without notice.

WyreStorm will not be liable for any use of this information or any changes it may make to these products. The use of this information constitutes an agreement by the user to these limitations and exclusions.

WyreStorm Offices
North America: 27 Wood Road, Round Lake NY
12151 Tel: +1 518-289-1293

EMEA: Unit 22, Ergo Business Park, Swindon, Wiltshire, SN3
3JW, UK Tel: +44 (0) 1793230343

WyreStorm Technologies reserves the right to change physical appearance or technical specification of this product at any time. Visit wyrestorm.com for the latest product information.