APO-VX20-UC v2

Application Programming Interface

Document Revision	V1.1
Document Date	January 2026
Supported Firmware	Refer to supported product firmware/software for details

Contents

0. Device Communication

0.1 Overview

The following contains the connection and commands to control the APO-VX20-UC v2. 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 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.

0. Device Management

1.1 Set Device Alias

Set the device alias.

Command Structure **SET ALIAS < DeivceName >**	Parameters: DeviceName = The alias to set, up to 31 characters. Allowed characters: letters, digits, spaces, underscore ('_'), and hyphen ('-'). Spaces cannot appear at the beginning or end.
Response Structure **ALIAS < DeivceName >**	
Command Example **SET ALIAS MeetingRoom**	
Response Example **ALIAS MeetingRoom**	
Additional Notes: This alias is used as the receiver name for BYOD screen-casting methods such as Miracast and AirPlay, and is also used as the SSID for the built-in soft AP.	

1.2 Get Device Alias

Get the device's 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 each software module on the device.

Command Structure **GET VER < ALL | ModuleName >**	Parameters: Use this command to query device information. This format queries the version information of modules on the device. 1. The parameter ModuleName specifies which module to query. Supported values are: - MAINSOC: Main program version (the commonly used firmware version number) - CPLD: CPLD firmware version - AUDIOSOC: Audio ARM firmware version - VIDEOSOC: Video ARM firmware version - VIDEOCHIP: Videochip firmware version - USBCCHIP: USB-C controller firmware version - USBCVIDEO: USB-C video firmware version 2. If you use "all" as the parameter, the command queries all module versions. The device will return multiple response lines, each giving one module's version. If no parameter is provided, the command queries the main program version.
Response Structure **VER ModuleName Version** **or** **VER ModuleName1 Version1** **VER ModuleName2 Version2** **...**	
Command Example **GET VER MAINSOC**	
Response Example **VER MAINSOC 1.2.0**	
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 the device to restore factory settings.

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

1.6 Get MAC Address

Retrieve the device's MAC address.

Command Structure **GET MACADDR**	Parameters:
Response Structure **MACADDR < MACAddress >**	
Command Example **GET MACADDR**	
Response Example **MACADDR E4:CE:02:13:5F:A1**	
Additional Notes:	

1.7 Get Device Serial Number

Get device serial number

Command Structure **GET SYS_SN**	Parameters:
Response Structure **SYS_SN < SerialNumber >**	
Command Example **GET SYS_SN**	
Response Example **SYS_SN WS29125320001**	
Additional Notes:	

1.8 Get Device Model

Retrieve the device's model information.

Command Structure **GET DEVICE_MODEL**	Parameters:
Response Structure **DEVICE_MODEL < model >**	
Command Example **GET DEVICE_MODEL**	
Response Example **DEVICE_MODEL APO-VX20-UC-V2**	
Additional Notes:	

1.9 Get Firmware Build Time

Retrieve firmware build time

Command Structure **GET BUILD_TIME**	Parameters: Prm = YYYY-MM-DD HH:MM:SS
Response Structure **BUILD_TIME < prm >**	
Command Example **GET BUILD_TIME**	
Response Example **BUILD_TIME 2025-11-11 11:29:30**	
Additional Notes:	

1.10 Set OSD Element Display

Configure whether a specific OSD element is displayed (enable or disable).

Command Structure **SET SHOWOSD < GUIDE | IP_ADDRESS > < ON | OFF >**	Parameters: Configure OSD display. Parameter GUIDE controls the Guide OSD element (ON = enable, OFF = disable). Parameter IP_ADDRESS controls the IP address OSD element (ON = enable, OFF = disable).
Response Structure **SHOWOSD < GUIDE | IP_ADDRESS > < ON | OFF >**	
Command Example **SET SHOWOSD GUIDE OFF**	
Response Example **SHOWOSD GUIDE OFF**	
Additional Notes:	

1.11 Get OSD Element Visibility

Retrieve the configuration that indicates whether an OSD element is allowed to be displayed.

Command Structure **GET SHOWOSD < GUIDE | IP_ADDRESS >**	Parameters:
Response Structure **SHOWOSD < GUIDE | IP_ADDRESS > < ON | OFF >**	
Command Example **GET SHOWOSD GUIDE**	
Response Example **SHOWOSD GUIDE OFF**	
Additional Notes:	

1.12 Set OSD Guide Name

Set the OSD Guide name

Command Structure **SET OSD_STR < GUIDE GuideNameStr >**	Parameters:
Response Structure **OSD_STR < GUIDE GuideNameStr >**	
Command Example **SET OSD_STR GUIDE Guide**	
Response Example **OSD_STR GUIDE Guide**	
Additional Notes:	

1.13 Get OSD Guide Name

Get the OSD guide name.

Command Structure **GET OSD_STR < GUIDE >**	Parameters:
Response Structure **OSD_STR < GUIDE GuideNameStr >**	
Command Example **GET OSD_STR GUIDE**	
Response Example **OSD_STR GUIDE Guide**	
Additional Notes:	

2. Input/Output Settings

2.1 Set Output Resolution

Set the resolution for the HDMI output interface. You can specify a fixed resolution directly or let the device automatically select 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 options are:
3840x2160P@60 3840x2160P@50 (only supported on OUT1) 3840x2160P@30 3840x2160P@25 3840x2160P@24 2560x1600P@60 2560x1440P@60 1920x1200P@60 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 can only be used on OUT1.

Additional Notes:

2.2 Get Output Resolution

Get the resolution of the HDMI output interface

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 ports. The device will return the two outputs on two separate lines.
AUTO = The output is set to automatic resolution
DISCONNECTED = The output is set to automatic resolution but no display is currently connected to that port

Additional Notes:

2.3 Get Output Signal HDCP

Get the HDCP encryption status of the output signal.

Command Structure
GET VIDOUT_HDCP < OUT1 | OUT2 | ALL >

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
```

Parameters:
ALL = Get the HDCP encryption status of all output signals

Additional Notes:

2.4 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 of all input interfaces

Additional Notes:

2.5 Get Input Signal Status

Get input signal status

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

Response Structure
VIDIN_SIG < IN1 | IN2 | BYOD1 | BYOD2 | BYOD3 | BYOD4 > < 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 = Retrieve the status of all input signals

Additional Notes:

2.6 Get Input Video Format

Retrieve the video format of the input signal.

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 video formats of all input interfaces
INVALID = No valid video signal detected on the input interface

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

The value and meaning of every field are as follows:
- Width = The width of the video
- Height = The height of the video
- P = Progressive video format
- FrameRate = Frame rate
- HDRInfo = Whether it's an HDR format; the device does not currently support HDR, so this value is always NONE HDR.
- EncMode = Encoding mode; available values are RGB、YUV444、YUV422、H.264、H.265
- ColorDepth = Color depth, currently only 8BIT
- An actual example of VideoFormat: 1920X1080P,60;NONE HDR;RGB;8BIT

Additional Notes:

2.7 Get Output Interface Connection Status

Retrieve the connection status of the output HDMI interface.

Command Structure **GET VIDOUT_CONNECT < OUT1 | OUT2 | ALL >**	Parameters:
Response Structure **VIDOUT_CONNECT < OUT1 | OUT2 > < CONNECTED | DISCONNECTED >** **or** **VIDOUT_CONNECT OUT1 < CONNECTED | DISCONNECTED >** **VIDOUT_CONNECT OUT2 < CONNECTED | DISCONNECTED >**	
Command Example **GET VIDOUT_CONNECT OUT1**	
Response Example **VIDOUT_CONNECT OUT1 CONNECTED**	
Additional Notes:	

3. Sleep/Standby and Peripheral Control

3.1 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 configured timeout period, it 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:	

3.2 Get Auto Standby

Retrieve the device auto-standby switch status.

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:	

3.3 Set Auto-Standby Timeout

Set the auto-standby timeout value.

Command Structure **SET AUTO_STANDBY_D < TimeOut >**	Parameters: TimeOut range: 0-3600; unit: seconds.
Response Structure **AUTO_STANDBY_D < TimeOut >**	
Command Example **SET AUTO_STANDBY_D 100**	
Response Example **AUTO_STANDBY_D 100**	
Additional Notes:	

3.4 Get Automatic Standby Timeout

Get the automatic 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:	

3.5 Set Device Power State (On/Standby)

Manually control the device to power on or enter 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:	

3.6 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:	

3.7 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:	

3.8 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: CECCode is the actual CEC message string 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:	

3.9 Get CEC Power On/Off Commands

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

Command Structure **GET CECCMD_EDIT < < out1 | out2 > [ON | OFF] | all >**	Parameters:
Response Structure **CECCMD_EDIT < OUT1 | OUT2 > < ON | OFF > < CECCode >** **或者** **CECCMD_EDIT < OUT1 | OUT2 > ON < CECCode >** **CECCMD_EDIT < OUT1 | OUT2 > OFF < CECCode >** **或者** **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 OUT1 ON**	
Response Example **CECCMD_EDIT OUT1 ON 4004**	
Additional Notes:	

3.10 Send CEC Command String

Send a CEC command string directly 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:	

3.11 Set CEC Function On/Off

Enable or disable the CEC function.

Command Structure **SET CEC_FN <prm>[CR/LF]**	Parameters: prm = {ON, OFF}
Response Structure **CEC_FN <prm>[CR/LF]**	
Command Example **SET CEC_FN on[CR/LF]**	
Response Example **CEC_FN ON[CR/LF]**	
Additional Notes: When the CEC CMD is disabled, all functions of the CEC module are inactive, including automatic CEC. When the CEC CMD is enabled, the following CEC CMDs are active.	

3.12 Get CEC Function On/Off

Get the CEC function ON/OFF status.

Command Structure **GET CEC_FN[CR/LF]**	Parameters: prm = {ON, OFF}
Response Structure **CEC_FN <prm>[CR/LF]**	
Command Example **GET CEC_FN[CR/LF]**	
Response Example **CEC_FN ON[CR/LF]**	
Additional Notes: Gets the CEC function ON/OFF status. The status is ON.	

4. Network and BYOD Related Settings

4.1 Set IP Address

Set the IP address

Command Structure **SET IPADDR < DHCP | STATIC IPAddress NetMask [Gateway [DNSServer]] >**	Parameters: For STATIC IP, provide the IP address and subnet mask. The gateway and DNS server are optional parameters.
Response Structure **IPADDR < DHCP | STATIC IPAddress NetMask [Gateway [DNSServer]] >**	
Command Example **SET IPADDR STATIC 192.168.1.34 255.255.255.0 192.168.1.1 192.168.1.2**	
Response Example **IPADDR STATIC 192.168.1.34 255.255.255.0 192.168.1.1 192.168.1.2**	
Additional Notes:	

4.2 Get IP Address

Get the device's IP address.

Command Structure **GET IPADDR**	Parameters:
Response Structure **IPADDR < DHCP | STATIC > < IPAddress > < NetMask > [Gateway [DNSServer]]**	
Command Example **GET IPADDR**	
Response Example **IPADDR STATIC 192.168.1.34 255.255.255.0 192.168.1.1 192.168.1.2**	
Additional Notes:	

4.3 Configure HTTPS

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:	

4.4 Get HTTPS Switch Status

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

4.5 Set Wi-Fi Operating Mode

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

Command Structure **SET WIFI_MODE < WiFiBand > < WifiChannel | AUTO >**	Parameters: WiFiBand = 2 or 5; they correspond to the 2.4 GHz and 5 GHz bands, respectively. WiFiChannel is the Wi-Fi channel; valid values depend on the band: 2.4G band: 1 2 3 4 5 6 7 8 9 10 11 5G band: 36 40 44 48 149 153 157 161 AUTO = the device automatically selects an appropriate channel. For VB20, implement this as specified; setting 2.4G will return an error initially.
Response Structure **WIFI_MODE < WiFiBand > < WifiChannel > [AUTO]**	
Command Example **SET WIFI_MODE 5 AUTO**	
Response Example **WIFI_MODE 5 161 AUTO**	
Additional Notes:	

4.6 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:	

4.7 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: WifiPassword must be 8-20 characters long and may contain only letters, digits, and the two special characters '_' and '-'.
Response Structure **SOFTAP_PWD < WifiPassword >**	
Command Example **SET SOFTAP_PWD 88888888**	
Response Example **SOFTAP_PWD 88888888**	
Additional Notes:	

4.8 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:	

4.9 Set Soft AP Function

The built-in soft AP can be enabled or disabled. This API is used to control that.

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

4.10 Get Soft AP Feature Switch Status

Returns the current enabled/disabled status of the device's Soft AP (software access point) function.

Command Structure **GET SOFTAP_EN**	Parameters: SOFTAP_EN: Status of the Soft AP feature. Possible values: - ON — Soft AP is enabled - OFF — Soft AP is disabled
Response Structure **SOFTAP_EN < ON | OFF >**	
Command Example **GET SOFTAP_EN**	
Response Example **SOFTAP_EN OFF**	
Additional Notes: Soft AP allows the device to act as a Wi-Fi hotspot. This command only queries the status; use the corresponding set command to change it.	

4.11 Configure Soft AP Routing

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

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

4.12 Get SoftAP 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:	

4.13 Set BYOD Function

Miracast and AirPlay receiver functions can be enabled or disabled. This API controls those settings.

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

4.14 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:	

4.15 Set BYOD Access Code

When using Miracast or AirPlay for screen casting, 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 >**	Parameters: AccessCode = Access code; must be 4 digits AUTO = Access code is auto-generated and changes dynamically NONE = No access code
Response Structure **BYOD_ACCESS_CODE < AccessCode | AUTO | NONE >**	
Command Example **SET BYOD_ACCESS_CODE 1234**	
Response Example **BYOD_ACCESS_CODE 1234**	
Additional Notes:	

4.16 Get BYOD Access Code

Command Structure **GET BYOD_ACCESS_CODE**	Parameters: The optional parameter AUTO indicates the current access code is dynamically generated.
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**	
Additional Notes:	

4.17 Set IP Conflict Detection

Configure IP conflict detection.

Command Structure **SET IP_CONFLICT < ON | OFF >**	Parameters: Sets the IP conflict detection function. ON enables detection, OFF disables detection.
Response Structure **IP_CONFLICT < ON | OFF >**	
Command Example **SET IP_CONFLICT ON**	
Response Example **IP_CONFLICT ON**	
Additional Notes:	

4.18 Get IP Conflict Detection Switch Status

Retrieve the status of the IP conflict detection feature switch.

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

5.1 Set MIC Source

Set MIC source

Command Structure **SET MIC_SRC < LOCAL | EXTEND | AUTO >**	Parameters: Set the MIC source. LOCAL indicates the local microphone; EXTEND indicates a cascaded (extended) microphone; AUTO automatically selects either the local or cascaded microphone based on an energy comparison.
Response Structure **MIC_SRC < LOCAL | EXTEND | AUTO >**	
Command Example **SET MIC_SRC AUTO**	
Response Example **MIC_SRC AUTO**	
Additional Notes:	

5. Audio Settings

5.2 Get MIC Source

Get the MIC source.

Command Structure **GET MIC_SRC**	Parameters:
Response Structure **MIC_SRC < LOCAL | EXTEND | AUTO >**	
Command Example **GET MIC_SRC**	
Response Example **MIC_SRC AUTO**	
Additional Notes:	

5.3 Set Audio Output Type

Set the audio output type.

Command Structure **SET AUDOUT_TYPE < SPEAKER | HDMI_PRIORITY | ALL >**	Parameters: Sets the audio output mode: SPEAKER = output to the local speaker. HDMI_PRIORITY = prefer HDMI OUT1 and HDMI OUT2 when they are connected to a TV; if neither HDMI OUT is connected to a TV, output falls back to the speaker. ALL = local speaker and HDMI OUTs output audio simultaneously.
Response Structure **AUDOUT_TYPE < SPEAKER | HDMI_PRIORITY | ALL >**	
Command Example **SET AUDOUT_TYPE SPEAKER**	
Response Example **AUDOUT_TYPE SPEAKER**	
Additional Notes:	

5.4 Get Audio Output Type

Get the audio output type

Command Structure **GET AUDOUT_TYPE**	Parameters:
Response Structure **AUDOUT_TYPE < SPEAKER | HDMI_PRIORITY | ALL >**	
Command Example **GET AUDOUT_TYPE**	
Response Example **AUDOUT_TYPE SPEAKER**	
Additional Notes:	

5.5 Set Audio Output Volume

Command Structure **SET AUDOUT_VOLUME < VolumeValue | INC | DEC >**	Parameters: Sets the audio output volume. VolumeValue specifies a volume percentage from 0 to 100. INC means increase volume, DEC means decrease volume. Each increase/decrease corresponds to a 2% volume change on Windows. VolumeValue is not implemented yet.
Response Structure **AUDOUT_VOLUME VolumeValue**	
Command Example **SET AUDOUT_VOLUME INC**	
Response Example **AUDOUT_VOLUME 80**	
Additional Notes:	

5.6 Get Audio Output Volume

Command Structure **GET AUDOUT_VOLUME**	Parameters:
Response Structure **AUDOUT_VOLUME VolumeValue**	
Command Example **GET AUDOUT_VOLUME**	
Response Example **AUDOUT_VOLUME 80**	
Additional Notes:	

5.7 Set Expansion MIC Mute Button Function

Command Structure **SET EXPANSION_MIC_MUTE_FN < ON | OFF >**	Parameters: Sets the expansion MIC mute button function. ON means enabled; OFF means disabled.
Response Structure **EXPANSION_MIC_MUTE_EN < ON | OFF >**	
Command Example **SET EXPANSION_MIC_MUTE_FN ON**	
Response Example **EXPANSION_MIC_MUTE_FN ON**	
Additional Notes:	

5.8 Get Expansion MIC Button Mute Function

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

6.1 Set Camera Operating Mode

Set the camera operating mode.

Command Structure **SET CAMERA_MODE < NORMAL | AUTO_FRAMING | SPEAKER_TRACKING | PRESENTER_TRACKING | INDIVIDUALS_GALLERY >**	Parameters: Sets the camera operating mode. NORMAL means AI features are not enabled; AUTO_FRAMING means automatic framing; SPEAKER_TRACKING means speaker tracking; PRESENTER_TRACKING means participant tracking; INDIVIDUALS_GALLERY means gallery mode.
Response Structure **CAMERA_MODE < NORMAL | AUTO_FRAMING | SPEAKER_TRACKING | PRESENTER_TRACKING | INDIVIDUALS_GALLERY >**	
Command Example **SET CAMERA_MODE NORMAL**	
Response Example **CAMERA_MODE NORMAL**	
Additional Notes:	

6. Camera Settings

6.2 Get Camera Operating Mode

Get the camera operating mode

Command Structure **GET CAMERA_MODE**	Parameters:
Response Structure **CAMERA_MODE < NORMAL | AUTO_FRAMING | SPEAKER_TRACKING | PRESENTER_TRACKING | INDIVIDUALS_GALLERY >**	
Command Example **GET CAMERA_MODE**	
Response Example **CAMERA_MODE NORMAL**	
Additional Notes:	

6.3 Set Camera Zoom Percentage

Command Structure **SET CAMERA_ZOOM < ZoomValue >**	Parameters: Set the Camera zoom percentage. ZoomValue range: 100-500
Response Structure **CAMERA_ZOOM < ZoomValue >**	
Command Example **SET CAMERA_ZOOM 200**	
Response Example **CAMERA_ZOOM 200**	
Additional Notes: Error codes: CAMERA_NOT_OPENED CAMERA_MODE_NOT_NORMAL	

6.4 Get Camera Zoom Percentage

Command Structure **GET CAMERA_ZOOM**	Parameters:
Response Structure **CAMERA_ZOOM < ZoomValue >**	
Command Example **GET CAMERA_ZOOM**	
Response Example **CAMERA_ZOOM 200**	
Additional Notes:	

6.5 Get Camera EPTZ Parameters

Command Structure **GET CAMERA_EPTZ**	Parameters: Query the Camera EPTZ parameters. PTZParam parameters are defined as: {AutoZoomUnit; PhyMaxZoom; xMax; xMin; xStep; yMax; yMin; yStep; MaxZoom} AutoZoomUnit: smallest zoom unit PhyMaxZoom: maximum physical zoom factor xMax: maximum coordinate value in the horizontal (left-right) direction xMin: minimum coordinate value in the horizontal (left-right) direction xStep: coordinate step for horizontal single-step movement yMax: maximum coordinate value in the vertical (up-down) direction yMin: minimum coordinate value in the vertical (up-down) direction yStep: coordinate step for vertical single-step movement MaxZoom: maximum zoom factor
Response Structure **CAMERA_EPTZ < EPTZParam >**	
Command Example **GET CAMERA_EPTZ**	
Response Example **CAMERA_EPTZ 1;500;108000;-108000;3600;108000;-108000;3600;500**	
Additional Notes:	

6.6 Set Camera Current Coordinates

Set the camera's coordinate position.

Command Structure **SET CAMERA_COORDINATE < X > < Y >**	Parameters: X is the horizontal (left-right) coordinate, Y is the vertical (up-down) coordinate. X and Y coordinates must be integer multiples of 3600.
Response Structure **CAMERA_COORDINATE < X > < Y >**	
Command Example **SET CAMERA_COORDINATE 0 0**	
Response Example **CAMERA_COORDINATE 0 0**	
Additional Notes: Error codes: CAMERA_NOT_OPENED CAMERA_MODE_NOT_NORMAL	

6.7 Set Camera Automatic Movement

Automatically and smoothly move the camera coordinates from the current position in the specified direction.

Command Structure **SET CAMERA_COORDINATE < AUTO_START > < UP | DOWN | LEFT | RIGHT > | AUTO_STOP**	Parameters: X represents the horizontal coordinate (left/right); Y represents the vertical coordinate (up/down). AUTO_START indicates starting automatic smooth movement in the specified direction. UP: move up DOWN: move down LEFT: move left RIGHT: move right AUTO_STOP indicates stopping automatic movement.
Response Structure **CAMERA_COORDINATE < AUTO_START > < UP | DOWN | LEFT | RIGHT >< X > < Y > or CAMERA_COORDINATE < AUTO_STOP > < X > < Y >**	
Command Example **SET CAMERA_COORDINATE 0 0**	
Response Example **CAMERA_COORDINATE 0 0**	
Additional Notes: Error codes: CAMERA_NOT_OPENED CAMERA_MODE_NOT_NORMAL	

6.8 Get Camera Current Coordinates

Command Structure **GET CAMERA_COORDINATE**	Parameters:
Response Structure **CAMERA_COORDINATE < X > < Y >**	
Command Example **GET CAMERA_COORDINATE**	
Response Example **CAMERA_COORDINATE 0 0**	
Additional Notes:	

6.9 Set Camera Preset Position

Command Structure **SET CAMERA_PRESET_POSITION < PresetNo >**	Parameters: PresetNo indicates the preset position number to save: valid range 1-4.
Response Structure **CAMERA_PRESET_POSITION < PresetNo >**	
Command Example **SET CAMERA_PRESET_POSITION 1**	
Response Example **CAMERA_PRESET_POSITION 1**	
Additional Notes: Error codes: ERR_CAMERA_NOT_OPENED ERR_CAMERA_MODE_NOT_NORMAL ERR_CAMERA_PRESET_NO_INVALID ERR_CAMERA_PRESET_FILE	

6.10 Get Camera Preset Position Status

Command Structure **GET CAMERA_PRESET_POSITION < PresetNo >**	Parameters: Returns TRUE if the preset position is set; returns FALSE if it is not set.
Response Structure **CAMERA_PRESET_POSITION < PresetNo > < TRUE | FALSE >**	
Command Example **GET CAMERA_PRESET_POSITION 1**	
Response Example **CAMERA_PRESET_POSITION 1 FALSE**	
Additional Notes:	

6.11 Activate Camera Preset

Command Structure **SET CAMERA_PRESET_LOAD < PresetNo >**	Parameters:
Response Structure **CAMERA_PRESET_LOAD < PresetNo >**	
Command Example **SET CAMERA_PRESET_LOAD 1**	
Response Example **CAMERA_PRESET_LOAD 1**	
Additional Notes: Error codes: ERR_CAMERA_NOT_OPENED ERR_CAMERA_MODE_NOT_NORMAL ERR_CAMERA_PRESET_NOT_EXIST ERR_CAMERA_PRESET_NO_INVALID ERR_CAMERA_PRESET_DATA	

6.12 Set Camera Horizontal Flip

Command Structure **SET CAMERA_MIRROR < ON | OFF >**	Parameters: ON indicates enabled (turned on); OFF indicates disabled (turned off).
Response Structure **CAMERA_MIRROR < ON | OFF >**	
Command Example **SET CAMERA_MIRROR ON**	
Response Example **CAMERA_MIRROR ON**	
Additional Notes:	

6.13 Get Camera Horizontal Flip

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

6.14 Set Camera HD

Set Camera HD mode

Command Structure **SET CAMERA_HD < ON | OFF >**	Parameters: ON means enabled, OFF means disabled
Response Structure **CAMERA_HD < ON | OFF >**	
Command Example **SET CAMERA_HD ON**	
Response Example **CAMERA_HD ON**	
Additional Notes:	

6.15 Get Camera HD

Retrieve the camera's HD mode.

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

6.16 Set Camera WDR

Set Camera WDR

Command Structure **SET CAMERA_WDR < WDRValue >**	Parameters: Camera WDR value range: -7 to 7
Response Structure **CAMERA_WDR < WDRValue >**	
Command Example **SET CAMERA_WDR 0**	
Response Example **CAMERA_WDR 0**	
Additional Notes:	

6.17 Get Camera WDR

Get the camera WDR (Wide Dynamic Range) setting.

Command Structure **GET CAMERA_WDR**	Parameters:
Response Structure **CAMERA_WDR < WDRValue >**	
Command Example **GET CAMERA_WDR**	
Response Example **CAMERA_WDR 0**	
Additional Notes:	

6.18 Set Camera Power Frequency

Command Structure **SET CAMERA_POWER_FREQ < 50HZ | 60HZ >**	Parameters: Valid values: 50HZ and 60HZ.
Response Structure **CAMERA_POWER_FREQ < 50HZ | 60HZ >**	
Command Example **SET CAMERA_POWER_FREQ 50HZ**	
Response Example **CAMERA_POWER_FREQ 50HZ**	
Additional Notes:	

6.19 Get Camera Power Frequency

Command Structure **GET CAMERA_POWER_FREQ**	Parameters:
Response Structure **CAMERA_POWER_FREQ < 50HZ | 60HZ >**	
Command Example **GET CAMERA_POWER_FREQ**	
Response Example **CAMERA_POWER_FREQ 50HZ**	
Additional Notes:	

6.20 Set Camera Auto-Framing Mode

Command Structure **SET CAMERA_AUTO_FRAMING_MODE < SMOOTH | IMMEDIATE >**	Parameters: Camera auto-framing mode. SMOOTH indicates a smooth transition; IMMEDIATE indicates an immediate switch.
Response Structure **CAMERA_AUTO_FRAMING_MODE < SMOOTH | IMMEDIATE >**	
Command Example **SET CAMERA_AUTO_FRAMING_MODE SMOOTH**	
Response Example **CAMERA_AUTO_FRAMING_MODE SMOOTH**	
Additional Notes:	

6.21 Get Camera Auto Framing Mode

Command Structure **GET CAMERA_AUTO_FRAMING_MODE**	Parameters:
Response Structure **CAMERA_AUTO_FRAMING_MODE < SMOOTH | IMMEDIATE >**	
Command Example **GET CAMERA_AUTO_FRAMING_MODE**	
Response Example **CAMERA_AUTO_FRAMING_MODE SMOOTH**	
Additional Notes:	

6.22 Set Camera Auto Framing Speed

Command Structure **SET CAMERA_AUTO_FRAMING_SPEED < SLOW | NORMAL | FAST >**	Parameters: Camera auto-framing speed. SLOW means slow, NORMAL means normal, FAST means fast.
Response Structure **CAMERA_AUTO_FRAMING_SPEED < SLOW | NORMAL | FAST >**	
Command Example **SET CAMERA_AUTO_FRAMING_SPEED SLOW**	
Response Example **CAMERA_AUTO_FRAMING_SPEED SLOW**	
Additional Notes:	

6.23 Get Camera Auto Framing Speed

Command Structure **GET CAMERA_AUTO_FRAMING_SPEED**	Parameters:
Response Structure **CAMERA_AUTO_FRAMING_SPEED < SLOW | NORMAL | FAST >**	
Command Example **GET CAMERA_AUTO_FRAMING_SPEED**	
Response Example **CAMERA_AUTO_FRAMING_SPEED SLOW**	
Additional Notes:	

6.24 Set Camera Speaker Tracking Mode

Command Structure **SET CAMERA_SPEAKER_TRACKING_MODE < SMOOTH | IMMEDIATE >**	Parameters: Camera speaker tracking mode: SMOOTH indicates smooth (gradual) switching; IMMEDIATE indicates immediate switching.
Response Structure **CAMERA_SPEAKER_TRACKING_MODE < SMOOTH | IMMEDIATE >**	
Command Example **SET CAMERA_SPEAKER_TRACKING_MODE SMOOTH**	
Response Example **CAMERA_SPEAKER_TRACKING_MODE SMOOTH**	
Additional Notes:	

6.25 Get Camera Speaker Tracking Mode

Command Structure **GET CAMERA_SPEAKER_TRACKING_MODE**	Parameters:
Response Structure **CAMERA_SPEAKER_TRACKING_MODE < SMOOTH | IMMEDIATE >**	
Command Example **GET CAMERA_SPEAKER_TRACKING_MODE**	
Response Example **CAMERA_SPEAKER_TRACKING_MODE SMOOTH**	
Additional Notes:	

6.26 Set Camera Maximum Resolution

Command Structure **SET CAMERA_MAX_RESOLUTION < 4K | 1080P | 720P >**	Parameters: Camera maximum resolution: 4K, 1080P, 720P
Response Structure **CAMERA_MAX_RESOLUTION < 4K | 1080P | 720P >**	
Command Example **SET CAMERA_MAX_RESOLUTION 4K**	
Response Example **CAMERA_MAX_RESOLUTION 4K**	
Additional Notes:	

6.27 Get Camera Maximum Resolution

Command Structure **GET CAMERA_MAX_RESOLUTION**	Parameters:
Response Structure **CAMERA_MAX_RESOLUTION < 4K | 1080P | 720P >**	
Command Example **GET CAMERA_MAX_RESOLUTION**	
Response Example **CAMERA_MAX_RESOLUTION 4K**	
Additional Notes:	

7. Telnet and SSH API Management

7.1 Set Telnet API Authentication Switch State

Enables or disables authentication for Telnet API login.

Command Structure **SET TELNET_AUTH_FN <prm>[CR/LF]**	Parameters: prm = {ON, OFF}: ON means enable authentication, OFF means disable authentication
Response Structure **TELNET_AUTH_FN <prm>[CR/LF]**	
Command Example **SET TELNET_AUTH_FN ON[CR/LF]**	
Response Example **TELNET_AUTH_FN ON[CR/LF]**	
Additional Notes: Applies to login security control for the Telnet API interface.	

7.2 Query Telnet API Authentication Switch Status

Checks whether Telnet API login authentication is enabled or disabled.

Command Structure **GET TELNET_AUTH_FN[CR/LF]**	Parameters: prm = {ON, OFF}: ON means authentication is enabled; OFF means authentication is disabled.
Response Structure **TELNET_AUTH_FN <prm>[CR/LF]**	
Command Example **GET TELNET_AUTH_FN[CR/LF]**	
Response Example **TELNET_AUTH_FN ON[CR/LF]**	
Additional Notes: Use together with SET TELNET_AUTH_FN to get the current Telnet API authentication status.	

7.3 Set API Access Password

Set or update the authentication password for Telnet API and SSH API access. The current password must be provided for verification.

Command Structure **SET API_PWD <oldpwd> <newpwd>[CR/LF]**	<oldpwd>: The currently active API password, used for authentication/verification. <newpwd>: The new password, 8–24 characters in length. Allowed characters: A–Z, a–z, 0–9, and the following special symbols: !@#$%^&*()_+-=[]{};:'"|,.<>/? Not allowed: spaces, non-ASCII characters, or any other symbols not listed above.
Response Structure **{Password wrong|User name or password wrong|Password changed successfully}[CR/LF]**	
Command Example **SET API_PWD 12345678 NewP@ssw0rd![CR/LF]**	
Response Example **Password changed successfully[CR/LF]**	
Additional Notes: The response is returned as result text and does not echo the plaintext password or the command. If the old password is incorrect, the device returns an error	

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.