SW-620-TX-W

Application Programming Interface

Document Revision	v1.0
Document Date	November 2025
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 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

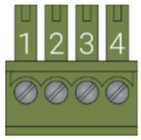
By default, the IP address is assigned by the DHCP server. If the DHCP server is not available the unit will use a link local IP address based on its MAC address, if the unit was not already configured via the Web interface.

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 a device alias
This alias is used as the receiver name for Miracast, Airplay, and other BYOD screen sharing methods. It also serves as the SSID name for the built-in soft AP.

Command Structure **SET ALIAS < DeviceName >**	Parameters: DeviceName = The alias you want to set, up to 31 characters long. Only letters, numbers, spaces, underscores ('_'), hyphens ('-'), and spaces are allowed. No spaces at the beginning or end.
Response Structure **ALIAS < DeviceName >**	
Command Example **SET ALIAS MeetingRoom**	
Response Example **ALIAS MeetingRoom**	
Additional Notes: Default value: SW-620-TX-W-XXXX, where XXXX is the last two hexadecimal digits of the MAC address. For example, SW-620-TX-W-327C	

1.2 Get Device Alias

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

1.3 Get Device Version Information

Get the version information of each software module on the device

Command Structure **GET VER < ALL | ModuleName >**	Parameters: ModuleName = The software module for which you want to get the version. Currently supports the following values: MAINSOC, USBCCHIP, USBCVIDEO, VIDEOCHIP ALL = Get the version information of all software modules, in which case the device returns the version information of all software modules line by line
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 Restart

Command the device to restart

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

1.5 Device Reset

Command to restore device to factory settings

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

1.6 Display OSD Elements

Show 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 to allow the display of a specific OSD element

Command Structure **SET SHOWOSD < DEVICENAME | IPADDRESS | WIFIPWD | ACCESSCODE > < ON | OFF >**	Parameters: For the 1st parameter, the listed four values correspond to device name, IP address, Wi-Fi password, and access code respectively.
Response Structure **SHOWOSD < DEVICENAME | IPADDRESS | WIFIPWD | ACCESSCODE > < ON | OFF >**	
Command Example **SET SHOWOSD DEVICENAME OFF**	
Response Example **SHOWOSD DEVICENAME OFF**	
Additional Notes: Default value: All four OSD elements mentioned here are allowed to display, with the corresponding configuration value being ON.	

1.8 Get OSD Element Display

Retrieve configuration information on whether OSD elements are allowed to be displayed

Command Structure **GET SHOWOSD < DEVICENAME | IPADDRESS | WIFIPWD | ACCESSCODE | ALL >**	Parameters: ALL = Get display settings for all OSD elements, returning settings line by line for each device
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: none	

1.9 Get MAC Address

Get the MAC address information of the device

Command Structure **GET MACADDR < CONTROL | SERVICE | ALL >**	Parameters: CONTROL = Get the MAC address of the control port SERVICE = Get the MAC address of the service port ALL = Get the MAC addresses for both the above 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:	

2. Input/Output Settings

2.1 Set Output Resolution

Set the resolution for the HDMI output port.
You can specify a resolution directly or let the device decide automatically.

Command Structure **SET VIDOUT_RES < OUT1 | OUT2 > < AUTO | Resolution >**	Parameters: AUTO = Automatically selects the best output resolution based on the connected display's EDID. Resolution = A specific resolution, with options as follows: 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 that 3840x2160P@60 and 3840x2160P@50 resolutions can only be used for OUT1
Response Structure **VIDOUT_RES < OUT1 | OUT2 > < AUTO | Resolution >**	
Command Example **SET VIDOUT_RES OUT1 AUTO**	
Response Example **VIDOUT_RES OUT1 AUTO**	
Additional Notes: Default: Both outputs are set to AUTO	

2.2 Get Output Resolution

Get the resolution of the HDMI output interface

Command Structure **GET VIDOUT_RES < OUT1 | OUT2 | ALL >**	Parameters: ALL = Get the resolution of both output interfaces, and the device will return the resolution of both outputs in two lines. AUTO = An output interface is set to automatic resolution DISCONNECTED = An output interface is set to automatic resolution but is currently not connected to a display
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**	
Additional Notes:	

2.3 Set Input Interface HDCP

Set whether to enable HDCP on the 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

Check if HDCP is enabled on the input interface

Command Structure **GET HDCP_S < IN1 | IN2 | ALL >**	Parameters: ALL = Get HDCP 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 Output Port HDCP

Set whether the output port has HDCP enabled

Command Structure **SET HDCP < OUT1 | OUT2 > < ON | OFF >**	Parameters:
Response Structure **HDCP < OUT1 | OUT2 > < ON | OFF >**	
Command Example **SET HDCP OUT1 OFF**	
Response Example **HDCP OUT1 OFF**	
Additional Notes: Default value: HDCP is enabled on all output ports.	

2.7 Get Output Interface HDCP

Check if the output interface HDCP is enabled

Command Structure **GET HDCP < OUT1 | OUT2 | ALL >**	Parameters: ALL = Check HDCP status on all output interfaces
Response Structure **HDCP < OUT1 | OUT2 > < ON | OFF >** **or** **HDCP OUT1 < ON | OFF >** **HDCP OUT2 < ON | OFF >**	
Command Example **GET HDCP OUT1**	
Response Example **HDCP OUT1 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

Get the connection status of input ports

Command Structure **GET VIDIN_CONNECT < IN1 | IN2 | ALL >**	Parameters: ALL = Get the connection status of all input ports
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**	
Additional Notes: none	

2.10 Get Input Signal Status

Get the status of the input signal

Command Structure **GET VIDIN_SIG < IN1 | IN2 | BYOD1 | BYOD2 | BYOD3 | BYOD4 | ALL >**	Parameters: ALL = Get the status of all input signals
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 >** **VIDIN_SIG BYOD3 < NO | VALID >** **VIDIN_SIG BYOD4 < NO | VALID >**	
Command Example **GET VIDIN_SIG IN2**	
Response Example **VIDIN_SIG IN2 VALID**	
Additional Notes:	

2.11 Get Input Video Format

Get the video format of the input signal

Command Structure
GET VIDIN_FORMAT < IN1 | IN2 | BYOD1 | BYOD2 | BYOD3 | BYOD4 | ALL >

Response Structure
VIDIN_FORMAT < IN1 | IN2 | BYOD1 | BYOD2 | BYOD3 | BYOD4 > < INVALID | VideoFormat >
or
VIDIN_FORMAT IN1 < INVALID | VideoFormat >
VIDIN_FORMAT IN2 < INVALID | VideoFormat >
VIDIN_FORMAT BYOD1 < INVALID | VideoFormat >
VIDIN_FORMAT BYOD2 < INVALID | VideoFormat >
VIDIN_FORMAT BYOD3 < INVALID | VideoFormat >
VIDIN_FORMAT BYOD4 < 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 sources
INVALID = No valid video signal on the input source

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 HDR format, the device currently doesn't support HDR, so its value is always NONE HDR.
- EncMode = Encoding mode, possible values are RGB, YUV444, YUV422, H.264
- ColorDepth = Color depth, currently only 8BIT
- An actual example of VideoFormat: 1920X1080P,60;NONE HDR;RGB;8BIT

Additional Notes:

2.12 Get Audio Input Format

Get the audio format of the input signal

Command Structure
GET AUDIN_FORMAT < IN1 | IN2 | BYOD1 | BYOD2 | BYOD3 | BYOD4 | ALL >

Response Structure
AUDIN_FORMAT < IN1 | IN2 | BYOD1 | BYOD2 | BYOD3 | BYOD4 > < INVALID | AudioFormat >
or
AUDIN_FORMAT IN1 < INVALID | AudioFormat >
AUDIN_FORMAT IN2 < INVALID | AudioFormat >
AUDIN_FORMAT BYOD1 < INVALID | AudioFormat >
AUDIN_FORMAT BYOD2 < INVALID | AudioFormat >
AUDIN_FORMAT BYOD3 < INVALID | AudioFormat >
AUDIN_FORMAT BYOD4 < 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;SampingWidth >

The value and meaning of every field are as follows:
- Encoding = The audio encoding mode, the available values are PCM, AAC, MP3 and OPUS
- SamplingRate = Sampling rate, such as 48kHz
- SampingWidth = Sampling width, the available values are 8BIT and 16BIT.
An actual example of AudioFormat: PCM;48kHz;8BIT

Additional Notes:

2.13 Set Output Volume

Set the volume for all output audio signals at once

Command Structure **SET VOLGAIN_DATA < OutputVolume >**	Parameters: The unit of the parameter OutputVolume is dB and the valid range is from -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 all outgoing audio signals to mute at the same time

Command Structure **SET AUDIO_MUTE < ON | OFF >**	Parameters: ON = Enable output mute
Response Structure **AUDIO_MUTE < ON | OFF >**	
Command Example **SET AUDIO_MUTE ON**	
Response Example **AUDIO_MUTE ON**	
Additional Notes: Default value: OFF, not muted	

2.16 Get Output Mute

Get the mute setting for output audio signal

Command Structure **GET AUDIO_MUTE**	Parameters:

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

2.17 Get Output Connection Status

Get the connection status of the output interface (whether a display is connected)

Command Structure **GET VIDOUT_CONNECT < OUT1 | OUT2 | ALL >**	Parameters: ALL = Get the connection status of both output interfaces
Response Structure **VIDOUT_CONNECT < OUT1 | OUT2 > < CONNECTED | DISCONNECTED >** **or** **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

Set the MST function for the USB-C input port. The MST function is activated when both of these conditions are met:
1. This setting is set to MST
2. Both HDMI outputs are connected
When MST is activated, the device's video switching logic will differ from the usual.

Command Structure **SET USBC_DM < SST | MST >**	Parameters: SST = USB-C input is in SST mode, does not support MST MST = USB-C input is in MST mode
Response Structure **USBC_DM < SST | MST >**	
Command Example **SET USBC_DM MST**	
Response Example **USBC_DM MST**	
Additional Notes: Default value: SST, MST function is off	

2.19 Get MST

Get MST function 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:	

3. Video Switching

3.1 Set Auto-Switch

Turn on or off the auto-switch function.
When the auto-switch function is on, the device supports these features:
1. Auto-switch video sources
2. Auto-switch multi-view layout
3. Fallback display video source

Command Structure **SET AUTOSW_FN < ON | OFF >**	Parameters: ON = Enable auto-switch function
Response Structure **AUTOSW_FN < ON | OFF >**	
Command Example **SET AUTOSW_FN OFF**	
Response Example **AUTOSW_FN OFF**	
Additional Notes: Default value: ON	

3.2 Get Auto Switch Status

Check if the auto switch feature is on or off

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 Multiview Feature

Turn on or off the multiview feature

Command Structure **SET MV_ENABLE < ON | OFF >**	Parameters: ON = Enable multiview feature
Response Structure **MV_ENABLE < ON | OFF >**	
Command Example **SET MV_ENABLE OFF**	
Response Example **MV_ENABLE OFF**	
Additional Notes: Default value: 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 an input to display full screen on all outputs

Command Structure **SET SW < IN1 | IN2 | BYOD1 | BYOD2 | BYOD3 | BYOD4 >**	Parameters:
Response Structure **SW < IN1 | IN2 | BYOD1 | BYOD2 | BYOD3 | BYOD4 >**	
Command Example **SET SW IN1**	
Response Example **SW IN1**	
Additional Notes:	

3.6 Switch Input to an Output

Switch the input to display full screen on a specific output.
When the specified output is not connected, the device will display the input source full screen on the currently connected output, and the response message will always return information about the actual output that is displaying.

Command Structure **SET SW < IN1 | IN2 | BYOD1 | BYOD2 | BYOD3 | BYOD4 > < OUT1 | OUT2 >**	Parameters:
Response Structure **SW < IN1 | IN2 | BYOD1 | BYOD2 | BYOD3 | BYOD4 > < 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 display on the single output.	

3.7 Display GUIDE Screen

Show the Guide screen on all outputs. This means no video sources are shown. If auto-standby is on, the device will go into standby after the Guide screen is displayed for some time (equal to the auto-standby timeout). Note that if the auto-standby timeout is 0, this command will put the device into 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 a specific output

Command Structure **GET SW < OUT1 | OUT2 | ALL >**	Parameters: ALL = Get the display status of all outputs GUIDE = The output is showing a guide screen instead of a video source MV = The output is in multi-view mode DISCONNECTED = The output is not connected
Response Structure **SW < IN1 | IN2 | BYOD1 | BYOD2 | BYOD3 | BYOD4 | GUIDE | MV | DISCONNECTED > < OUT1 | OUT2 >** **or** **SW < IN1 | IN2 | BYOD1 | BYOD2 | BYOD3 | BYOD4 | GUIDE | MV | DISCONNECTED > OUT1** **SW < IN1 | IN2 | BYOD1 | BYOD2 | BYOD3 | BYOD4 | GUIDE | MV | DISCONNECTED > OUT2**	
Command Example **GET SW OUT1**	
Response Example **SW IN1 OUT1**	
Additional Notes:	

3.9 Set current layout

Switch multi-view layout, parameter is 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

Change the multiview layout
The parameter is the layout ID in hexadecimal format

Command Structure **SET MV_LAYOUT < 0X100 | 0X101 >**	Parameters: The four hexadecimal IDs match the four layout names from the previous API.
Response Structure **MV_LAYOUT < 0X100 | 0X101 >**	
Command Example **SET MV_LAYOUT 0X101**	
Response Example **MV_LAYOUT 0X101**	
Additional Notes: na	

3.11 Get Current Layout ID and Name

Get the name of the current multi-view layout

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 Source

Switch multi-window layout and set video source for each sub-window

Command Structure **SET MV_WIN_SRC < FULLSCREEN | DUALVIEW | 3WAYSPLIT| CURRENT > < WinNo1 > < Src1 > < WinNo2 > < Src2 > ...**	Parameters: CURRENT = Keep the current layout unchanged, only switch video source for specified windows WinNo = Win1 Win2 Win3... Src1 Src2... Range is < IN1 | IN2 | IN3 | IN4 | BYOD1 | BYOD2 | BYOD3 | BYOD4 | NONE > - NONE = Do not display any video source in this sub-window, sub-window is empty
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**	
Additional Notes:	

3.13 Get Layout and Video Sources

Get the current layout and video sources of each sub-window

Command Structure **GET MV_WIN_SRC**	Parameters: NONE = Currently connected to dual output, not supporting multi-window.
Response Structure **MV_WIN_SRC NONE** **or** **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**	
Additional Notes: When connected to a single output with the multi-window feature turned off, this API returns normally.	

3.14 Retrieve Video Source for a Specified Window

Get the video source for a specified sub-window

Command Structure **GET MV_WIN_SRC CURRENT WinNo**	Parameters: NONE = Currently connected with dual output, multi-view is not supported.
Response Structure **MV_WIN_SRC NONE** **or** **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 single output but the multi-view function is off, this API responds normally. This API function is a subset of the previous API, designed to adapt to SYN-TOUCH10.	

4. Audio and Other Signal Switching

4.1 Set Audio Auto-Switch

Set the audio auto-switch toggle

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

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 audio auto-switching is off, use this API to make the device always output audio from a wired source

Command Structure **SET AUDIOSW < IN1 | IN2 >**	Parameters: STOP = Currently in Guide screen status, 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 off, let the device temporarily output audio from a BYOD source. The device will play audio from the BYOD source until it disconnects, then switch back to the previously set wired audio source. In short, it's like a temporary switch action.

Command Structure **SET AUDIOSW < BYOD1 | BYOD2 | BYOD3 | BYOD4 >**	Parameters:
Response Structure **AUDIOSW < BYOD1 | BYOD2 | BYOD3 | BYOD4 >**	
Command Example **SET AUDIOSW BYOD1**	
Response Example **AUDIOSW BYOD1**	
Additional Notes:	

4.5 Get Current Audio Source

Retrieves the current audio source information output by the device. When the device displays the Guide screen, it will stop audio output. At this time, this API will return the most recent wired audio source information.

Command Structure **GET AUDIOSW**	Parameters: STOP = Currently in Guide screen state, the device is not actually outputting audio
Response Structure **AUDIOSW < IN1 | IN2 | BYOD1 | BYOD2 | BYOD3 | BYOD4 > [STOP]**	
Command Example **GET AUDIOSW**	
Response Example **AUDIOSW IN1**	
Additional Notes:	

4.6 Set USB Auto Switch

Set device to automatically switch USB HOST

Command Structure **SET USB_AUTOSW_FN < ON | OFF >**	Parameters: ON = Enable the USB auto switch feature
Response Structure **USB_AUTOSW_FN < ON | OFF >**	
Command Example **SET USB_AUTOSW_FN ON**	
Response Example **USB_AUTOSW_FN ON**	
Additional Notes: Default value: ON	

4.7 Get USB Auto-Switch Status

Find out the status of the USB auto-switch

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

4.8 Set USB HOST

Manually switch USB HOST

Command Structure **SET USB_HOST <USBCIN1 | USBHOST1 | WIRELESS>**	Parameters: WIRELESS = USB HOST connected to main chip for Wireless Conference
Response Structure **USB_HOST <USBCIN1 | USBHOST1 | 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 < USBCIN1 | USBHOST1 | WIRELESS >**	
Command Example **GET USB_HOST**	
Response Example **USB_HOST USBHOST1**	
Additional Notes:	

5. Sleep/Standby and Peripheral Control

5.1 Set Auto Standby

Set the device's auto standby feature.
When the auto standby feature is on, if the device keeps showing the Guide screen for a certain time (determined by the standby timeout value), it will send a shutdown command to all connected displays and turn off the HDMI +5V and TMDS signals.

Command Structure **SET AUTO_STANDBY_FN < ON | OFF >**	Parameters: ON = Turn on the auto standby feature
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 Auto Standby

Get the auto standby switch status of the device

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 Auto Standby Timeout

Sets the auto standby timeout value

Command Structure **SET AUTO_STANDBY_D < TimeOut >**	Parameters: TimeOut value range is 0-3600, measured in seconds
Response Structure **AUTO_STANDBY_D < TimeOut >**	
Command Example **SET AUTO_STANDBY_D**	
Response Example **AUTO_STANDBY_D 100**	
Additional Notes: Default value: 120	

5.4 Get Auto Standby Timeout

Get the value for auto standby timeout

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 device power on or 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

Check if 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 off command to the specified output port

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 off command for a specific output interface

Command Structure **SET CECCMD_EDIT < OUT1 | OUT2 > < ON | OFF > < CECCode >**	Parameters: The parameter CECCode is the actual CEC message in a 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 Command

Get the CEC power on or off command for a 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 Command

Get the CEC on and off command for a specific output port

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

Get all CEC power on and off commands for 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 CEC command string to a specific output port

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 Serial Port Parameters

Set baud rate and three other serial port parameters

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

5.14 Get Serial Port Parameters

Get serial port baud rate and other settings

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

5.15 Set Serial Baud Rate

Set the serial port's baud rate settings

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 Serial Port Baud Rate

Get the baud rate settings of the serial port

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 Serial Port Command

Set serial port command

Command Structure **SET UART_CMD < CmdName > < HEX | STR > < CmdStr >**	Parameters: CmdName = Command name, besides ON and OFF, can also be other custom commands like VolumeUp HEX = Command in hexadecimal format STR = Command in string format CmdStr = Command string, no spaces are needed between adjacent bytes in hexadecimal string.
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 serial port command is defined.	

5.18 Delete Serial Port Command

Remove a serial port 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 Serial Port Command

Get one or all serial port commands

Command Structure **GET UART_CMD < CmdName | ALL >**	Parameters:
Response Structure **UART_CMD < CmdName > < HEX | STR > < CmdStr >** **or** **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 Serial Port Command

Send a predefined serial port 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 Serial Port Command String

Directly send serial port 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 Serial Port Power Control Feature

Set whether the serial port sends power on/off commands.
When the serial port is in external control mode, if this setting is enabled, the device will send the respective command through the serial port whenever it enters or exits standby mode, in addition to sending CEC commands.

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 value: OFF	

5.23 Retrieve Serial Port Power On/Off Function

Check if the serial port is involved 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 Serial Command

Send CEC or/and serial commands at the same time

Command Structure **SET SEND_CMD < ON | OFF > < RS232 | CEC | ALL >**	Parameters: RS232 = Send only serial command CEC = Send only CEC command ALL = Send both serial 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: none	

5.25 Set Serial Port Operating Mode

The serial port can be used for external control or to provide API services (i.e., accept control). This API allows you to set this.

Command Structure **SET UART_MODE < API | COM >**	Parameters: API = The serial port is used to provide API services externally, meaning that the actual data sending functions in previously mentioned APIs will be disabled. COM = The serial port 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 value: COM	

5.26 Get Serial Port Mode

Get the mode of the serial port

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

5.29 Get MUTE Input

Get the status of the MUTE input

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

6. Network and BYOD settings

6.1 Set Network Isolation

When network isolation mode is enabled, the device's network port is split into two separate VLANs, which can be used to connect to different networks. This API is used to set whether to enable network isolation.

Command Structure **SET NET_ISOLATION < ON | OFF >**	Parameters: ON = Turn on the network isolation feature
Response Structure **NET_ISOLATION < ON | OFF >**	
Command Example **SET NET_ISOLATION ON**	
Response Example **NET_ISOLATION ON**	
Additional Notes: Default value: OFF	

6.2 Get Network Isolation Settings

Check if the 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 DHCP IP address

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 business 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

Get the device's IP address

Command Structure **GET IPADDR < CONTROL | SERVICE | ALL >**	Parameters: ALL = Get IP addresses of both network ports
Response Structure **IPADDR < CONTROL | SERVICE > < DHCP | STATIC > < IPAddress > < NetMask > [Gateway [DNSServer]]** **or** **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: none	

6.6 Set USB-C Network Adapter Function

The network adapter on the USB-C port can be turned on or off. This API is used to control this feature.

Command Structure **SET USBNIC < IN1 > < ON | OFF >**	Parameters: ON = Turn on USB-C port network adapter function
Response Structure **USBNIC < IN1 > < ON | OFF >**	
Command Example **SET USBNIC IN1 OFF**	
Response Example **USBNIC IN1 OFF**	
Additional Notes: Default value: ON	

6.7 Get USB-C Network Card Switch Status

Get the switch status of the USB-C network card interface

Command Structure **GET USBNIC < IN1 | ALL >**	Parameters: ALL = Get the switch status of both USB-C input network card interfaces
Response Structure **USBNIC < IN1 > < ON | OFF >** **or** **USBNIC IN1 < ON | OFF >**	
Command Example **GET USBNIC IN1**	
Response Example **USBNIC IN1 OFF**	
Additional Notes:	

6.8 Set HTTPS Feature

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 value: 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 Set Telnet over TLS Feature

Command Structure **SET TELNETS < ON | OFF >**	Parameters: ON = Turn on Telnet over TLS
Response Structure **TELNETS < ON | OFF >**	
Command Example **SET TELNETTLS ON**	
Response Example **TELNETTLS ON**	
Additional Notes: Default value: OFF	

6.11 Get Telnet over TLS Feature Switch Status

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

6.12 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 5, corresponds to 2.4G and 5G bands WiFiChannel is the Wi-Fi channel, and its range depends 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 = 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 AUTO**	
Additional Notes: Default value: 5 AUTO Means 5G band, automatically select channel	

6.13 Get Wi-Fi Operating Mode

Get WiFi operating mode

Command Structure **GET WIFI_MODE**	Parameters: Not available
Response Structure **WIFI_MODE < WiFiBand > < WifiChannel > [AUTO]**	
Command Example **GET WIFI_MODE**	
Response Example **WIFI_MODE 5 161 AUTO**	
Additional Notes: Not available	

6.14 Set Wi-Fi Password

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

Command Structure **SET SOFTAP_PWD < WifiPassword >**	Parameters: The parameter WiFiPassword must be between 8 and 20 characters long. It can only include letters, numbers, and two 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

Get the password of 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 Set Soft AP Function

The built-in Soft AP can be turned on and off, and this API is used to control it.

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 value: ON	

6.17 Get Soft AP Function Status

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 Soft Router Function

The device supports soft routing, allowing devices connected to the soft API to access network resources connected to the device's wired ports. If this network can access the internet, then devices connected to the soft AP can also access the internet. This API controls this 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: Default value: ON	

6.19 Get Soft Router Feature 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 Set BYOD Function

Turn the Miracast and Airplay receiver feature on or off. This API controls that.

Command Structure **SET BYOD_EN < ON | OFF >**	Parameters: ON = Enable BYOD receiver feature
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 using Miracast or Airplay screen sharing, you can require users to enter an access code to avoid interference. This API controls that function.

Command Structure **SET BYOD_ACCESS_CODE < AccessCode | AUTO | NONE >**	Parameters: AccessCode = Access code, must be a 4-digit number AUTO = Automatically generates an access code that 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: Default value: NONE	

6.23 Get BYOD Access Code

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

6.24 Set SSID Broadcast

Normally, a soft AP will regularly send out broadcast packets containing its SSID information. Turning off this feature can enhance security to some extent. This API is used to control this 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 value: ON	

6.25 Get SSID Broadcast Status

Command Structure **GET SSID_BROADCAST_FN**	Parameters:
Response Structure **SSID_BROADCAST_FN < ON | OFF >**	
Command Example **GET SSID_BROADCAST_FN**	
Response Example **SSID_BROADCAST_FN OFF**	
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.