

**iTrans UC-HUB402
API Manual**

Version: V1.1

Note: This document is suitable for iTrans UC-HUB402
firmware revision V1.4.0 or later

Contents

1	Introduction	5
1.1	Preparation	5
1.1.1	Setting IP Address in Your Computer	5
1.1.2	Enabling Telnet Client	5
1.2	Logging In via Command Line Interface	6
1.3	Access API via RS232 Interface.....	7
1.4	Introduction	7
1.4.1	Terminology	7
1.4.2	Classification	7
1.5	API Syntax Overview	8
1.5.1	Case-Handling Rules.....	8
1.5.2	Proactive Notification Mechanism.....	8
1.5.3	Terms & Conventions.....	9
2	Command Details.....	10
2.1	Device management	10
2.1.1	Set device alias	10
2.1.2	Get device alias	11
2.1.3	Set IP address.....	11
2.1.4	Get IP address	12
2.1.5	Get firmware version	12
2.1.6	Get MAC address.....	13
2.1.7	System reboot.....	13
2.1.8	Factory reset	14
2.2	Input & output settings.....	14
2.2.1	Set output resolution	14
2.2.2	Get output resolution.....	15
2.2.3	Get output connection status	16
2.2.4	Set HD mode	16
2.2.5	Get HD mode.....	17
2.2.6	Set output HDCP configuration.....	17

2.2.7	Get output HDCP configuration	18
2.2.8	Get input connection status.....	18
2.2.9	Get input signal status.....	19
2.2.10	Get input video format.....	20
2.2.11	Get input audio format	21
2.2.12	Get Output Connection Status	22
2.2.13	Set USB Device Mode	22
2.2.14	Get USB Device Mode	23
2.3	Video switching	23
2.3.1	Set low latency mode.....	23
2.3.2	Get low latency mode	24
2.3.3	Set input to output.....	25
2.3.4	Get input of output	25
2.3.5	Set multiview layout.....	26
2.3.6	Get multiview layout	26
2.3.7	Set multiview layout and video source(s)	27
2.3.8	Get multiview layout and video source(s).....	28
2.4	Audio processing	28
2.4.1	Set DSP audio mixing.....	28
2.4.2	Get DSP audio mixing	29
2.4.3	Set DSP input channel	30
2.4.4	Get DSP input channel.....	30
2.4.5	Set DSP input mute	31
2.4.6	Get DSP input mute.....	31
2.4.7	Set DSP input gain	32
2.4.8	Get DSP input gain	32
2.4.9	Set DSP output mute.....	33
2.4.10	Get DSP output mute	33
2.4.11	Set DSP output volume	34
2.4.12	Get DSP output volume.....	34
2.5	Standby related & peripheral control	35
2.5.1	Set standby mode	35

2.5.2	Get standby mode.....	35
2.5.3	Set automatic standby	36
2.5.4	Get automatic standby.....	36
2.5.5	Set automatic standby timeout.....	36
2.5.6	Get automatic standby timeout	37
2.5.7	Set CEC command	37
2.5.8	Get CEC command.....	38
2.5.9	Set RS232 work mode	39
2.5.10	Get RS232 work mode.....	39
2.5.11	Set power on/off display via RS232.....	40
2.5.12	Get power on/off display via RS232	40
2.5.13	Configure RS232 settings	41
2.5.14	Get RS232 setting.....	41
2.5.15	Set RS232 baud rate	42
2.5.16	Get RS232 baud rate	42
2.5.17	Set RS232 command	42
2.5.18	Delete RS232 command	43
2.5.19	Get RS232 command.....	44
2.5.20	Send RS232 command.....	44
2.5.21	Send RS232 data.....	45
2.5.22	Send CEC or/and RS232 command.....	45
2.5.23	Send CEC data	45
2.6	NDI input & output.....	46
2.6.1	Set NDI RX group name	46
2.6.2	Get NDI RX group name	46
2.6.3	Set NDI RX device name	47
2.6.4	Get NDI RX device name	47
2.6.5	Get NDI device list	48
2.6.6	Get NDI channel list.....	48
2.6.7	Set NDI video source	49
2.6.8	Delete NDI video source.....	49
2.6.9	Get NDI video source	50

2.6.10	Set NDI TX group name	50
2.6.11	Get NDI TX group name.....	51
2.6.12	Set NDI TX device name	51
2.6.13	Get NDI TX device name.....	51
2.6.14	Set NDI TX channel name	52
2.6.15	Get NDI TX channel name	52
2.6.16	Set NDI TX encoding parameter	53
2.6.17	Get NDI TX encoding parameter	54
2.6.18	Set NDI stream output	54
2.6.19	Get NDI stream output.....	55

1 Introduction

1.1 Preparation

This section takes a third-party control device windows 10 as an example. You may also use other control devices.

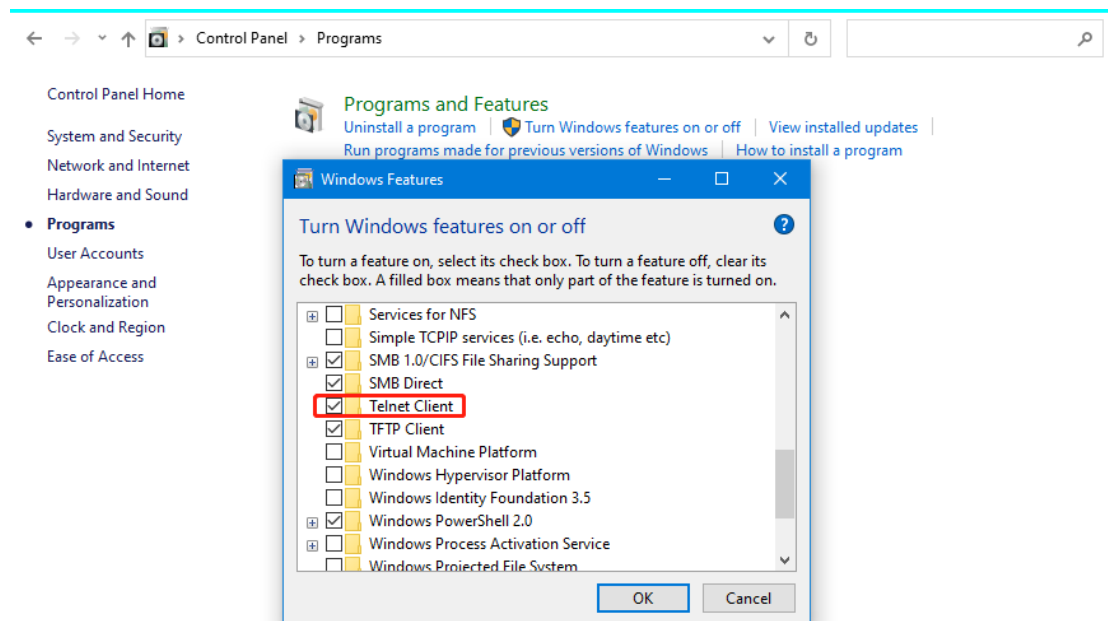
1.1.1 Setting IP Address in Your Computer

The detailed operation steps are omitted here, please consult your IT manager or search online for relevant guidance

1.1.2 Enabling Telnet Client

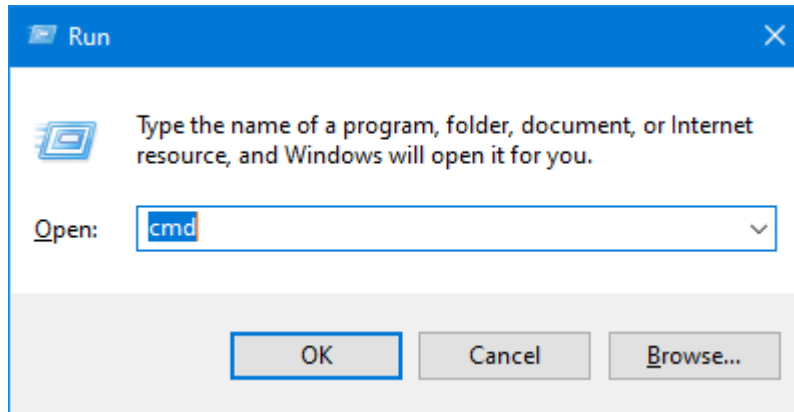
Before logging in to the device via command-line interface, make sure that **Telnet Client** is enabled. By default, **Telnet Client** is disabled in Windows OS. To turn on **Telnet Client**, do as follows.

1. Choose **Start > Windows System > Control Panel**.
2. In the popped dialog, click **Programs**.
3. In the **Programs and Features** area, click **Turn Windows features on or off**.
4. In **Windows Features** dialog box, select **Telnet Client** check box.



1.2 Logging In via Command Line Interface

1. Choose **Start > Run**.
2. In the **Run** dialog box, enter **cmd** then click **OK**.



3. Enter **telnet 169.254.1.100** if the device's IP address is 169.254.1.100, and then press **Enter**.

```
E:\>telnet 169.254.1.100
```

4. The device will show the command prompt as below:

```
Welcome!
```

Now, the device are ready to execute the CLI API command.

1.3 Access API via RS232 Interface

In addition to Telnet protocol, the API can be accessed via RS232 interface too. Here are brief instructions:

1. As the default, the RS232 interface is used to control the peripherals. To change its purpose to provide API services, you should invoke the API `SET UART_MODE` at first, the corresponding API request message is as follows

```
SET UART_MODE API
```

After completing the above operation, please reboot the device, then you can access the API via the RS232 interface.

2. The two special characters: Carriage Return and Line Breaks must be added at the end of each API request message in order. Their ASCII codes are 0x0D and 0x0A respectively. If you use a terminal emulation software (such as SecureCRT), the software will add these two characters automatically or else you should add these two characters manually.

1.4 Introduction

1.4.1 Terminology

The terminology used in API command description is listed as follows.

Terminology	Description
Device Product	The iTrans UC-HUB402 unit being controlled. Unless otherwise specified, these two words appearing in this document refer to iTrans UC-HUB402.
Hardware Source Wired Source	The video source which has dedicated hardware input interface, namely, the USB and HDMI sources, they are named hardware (video) sources or wired sources .
Network Source NDI Source	NDI video source has not dedicated hardware interface, all data is transported over the Ethernet interface, it is named network source or NDI source .

1.4.2 Classification

All APIs can be divided into the following categories according to their functions:

1. Device management

Manage the device name, reboot or reset device, etc...

2. Input & output management

Manage HDCP, output resolution, etc...

3. Video switching

Many different approaches to switch video source

4. Audio processing

Manage audio signal, such as volume, mute and routing

5. Standby related & peripheral control

Control the device to standby and wake-up, send CEC and RS232 message

6. Network

Manage IP address and network protocols

7. NDI management

Manage NDI related features

1.5 API Syntax Overview

1.5.1 Case-Handling Rules

1. For fields that are inherently case-sensitive—such as names and passwords—preserve the exact characters the user provided in the request and return them unchanged in the corresponding response. Fields currently covered by this rule are:

- ✧ *DeviceName* device name
- ✧ *CmdStr* RS232 port command string in text form
- ✧ NDI-related group, device, and channel names

2. For standardized units (e.g., dB, Hz), ignore case on input and always return the canonical form.
3. Internet domain names must be treated case-insensitively when processing requests and must always be returned in lowercase in responses.
4. For all other fields not mentioned above, treat the input case-insensitively and always return the value in UPPERCASE in the response.

1.5.2 Proactive Notification Mechanism

Any change that occurs on the device must be reported to the control clients immediately via unsolicited messages, following these rules:

1. The response message of a `GET` API is sent only to the control client that issued the `GET`

request.

2. If a `SET` API fails to execute due to an error, only the control client that issued the request shall receive the corresponding error response.
3. If a `SET` API executes successfully, the response is broadcast to every currently connected control client as a real-time notification.
4. Whenever the operating state changes on its own (i.e., not triggered by a `SET` API), the device must proactively send the corresponding `SET/GET` API response to all currently connected control clients.

1.5.3 Terms & Conventions

1. The italic string in the message format indicates that the string should be replaced by the actual parameter in the actual message, and the italic string itself should not appear in the final message body. For example

```
DeviceName
```

Indicates the device name. The actual device name should be given in the actual message body.

2. Square brackets indicate optional parameters, for example

```
[ WinNo ]
```

indicates an optional window number parameter. Obviously, optional parameters can only appear at the end of the message.

3. The combination of curly braces and vertical bars is used to specify the possible values for the parameter, for example

```
< IN1 | IN2 | IN3 | IN4 >
```

Represents a parameter whose value can be one of `IN1~IN4`.

4. The combination of square brackets and vertical bars is used to specify the possible values for optional parameters, for example

```
[ START | STOP ]
```

Indicates an optional parameter whose value can be `START` or `STOP`

5. Definition of common error response messages:

```
ERROR ErrNo ErrReqMsg
```

If any error occurs when executing the API, the device can print specific prompt information. The response message begins with the word 'ERROR', following with an error number and the copy of the request message itself. The below screenshot is an example:

```
GET NULL
ERROR 1 : GET NULL
```

6. Some commonly used parameters

✧ Wired source

WiredSrc

Valid values are `IN1`, `IN2`, `IN3` and `IN4`, represents one of four wired video sources **USB IN1**, **USB IN2**, **HDMI IN1** and **HDMI IN2**.

✧ NDI source

NdiSrc

Valid values are `NDI1`, `NDI2`, `NDI3` and `NDI4`, indicating one of four NDI video sources.

✧ All input source

Src

Valid values are `IN1`, `IN2`, `IN3`, `IN4`, `NDI1`, `NDI2`, `NDI3` and `NDI4`, represents one of four wired sources and four NDI sources.

✧ DSP audio input

DspIn

One input channel or signal of the audio DSP, valid values are `USBIN`, `HDMIIN`, `USBHOST` and `ANALOGIN`, corresponding to the audio signal from the interface **USB IN1** (or **USB IN2**), **HDMI IN1** (or **HDMI IN2**), **USB HOST** and **AUDIO IN** (analog audio input).

✧ DSP audio output

DspOut

One output channel or signal of the audio DSP, valid values are `USBIN`, `HDMIOUT`, `USBHOST` and `ANALOGOUT`, corresponding to the audio signal sent to the interface **USB IN1** (and **USB IN2**), **HDMI OUT1** (and **HDMI IN2**), **USB HOST** and **AUDIO OUT** (analog audio output).

2 Command Details

2.1 Device management

2.1.1 Set device alias

【Format】

Request	SET ALIAS < DeviceName >
Response	ALIAS < DeviceName >
Description	<p>Configure the device's alias. As a prompt, the new alias will appear on the top-right corner of the screen if the operation is successful.</p> <p>As the factory default, the alias is HUB402-xxxxxxx, xxxxxx is the hexadecimal format of the last three digits of the device MAC address.</p> <p>Note:</p> <p>The device alias must be 1~31 characters in length, furthermore, it must include only letters, numbers, space and two special character ('_' and '-'). The beginning or end character must be a letter or number.</p>

【Example】

To change the name to MeetingRoom:

Request:

```
SET ALIAS MeetingRoom
```

Response:

```
ALIAS MeetingRoom
```

2.1.2 Get device alias

【Format】

Request	GET ALIAS
Response	ALIAS < DeviceName >
Description	Obtain the device alias.

【Example】

Request:

```
GET ALIAS
```

Response:

```
ALIAS MeetingRoom
```

2.1.3 Set IP address

【Format】

Request	SET IPADDR DHCP
	Or SET IPADDR STATIC < <i>IPAddress</i> > < <i>NetMask</i> > [<i>Gateway</i> [<i>DNSServer</i>]]
Response	IPADDR DHCP
	Or IPADDR STATIC < <i>IPAddress</i> > < <i>NetMask</i> > [<i>Gateway</i> [<i>DNSServer</i>]]
Description	Configure the IP address, the two formats correspond to DHCP and static modes respectively. <i>Gateway</i> and <i>DNSServer</i> are optional parameters As the factory default, DHCP is used, if DHCP fails, the IP address 169.254.1.100 is used.

【Example】

Request:

```
SET IPADDR STATIC 192.168.1.34 255.255.255.0
```

Response:

```
IPADDR STATIC 192.168.1.34 255.255.255.0
```

2.1.4 Get IP address

【Format】

Request	GET IPADDR
Response	IPADDR < DHCP STATIC > < <i>IPAddress</i> > < <i>NetMask</i> > [<i>Gateway</i> [<i>DNSServer</i>]]
Description	Query the IP address.

【Example】

Request:

```
GET IPADDR
```

Response:

```
IPADDR STATIC 192.168.1.34 255.255.255.0
```

2.1.5 Get firmware version

【Format】

Request	GET VER [ALL <i>ModuleName</i>]
----------------	-------------------------------------

Response	VER <i>ModuleName</i> <i>Version</i>
	or VER <i>ModuleName1</i> <i>Version1</i> VER <i>ModuleName2</i> <i>Version2</i> ...
Description	Obtain the firmware version information. The device returns the version information of the module designated by the parameter <i>ModuleName</i> , the available values are MAINSOC, CAMCHIP and VIDEOCHIP.
	If the parameter value is ALL, the device responds with all modules' version information row by row.

【Example】

Request:

```
GET VER MAINSOC
```

Response:

```
VER MAINSOC 1.1.7
```

2.1.6 Get MAC address

【Format】

Request	GET MACADDR
Response	MACADDR <i>MacAddress</i>
Description	Obtain the MAC address, the response message prints the MAC address with colon separated form.

【Example】

Request:

```
GET MACADDR
```

Response:

```
MACADDR AA:BB:CC:DD:EE:07
```

2.1.7 System reboot

【Format】

Request	REBOOT
----------------	--------

Response	REBOOT
Description	Reboot the device manually

【Example】

Request:

```
REBOOT
```

Response:

```
REBOOT
```

2.1.8 Factory reset

【Format】

Request	RESET
Response	RESET
Description	This command make the device restore its factory default, to archive this, the device will reboot to enter recovery or safe mode.

【Example】

Request:

```
RESET
```

Response:

```
RESET
```

2.2 Input & output settings

2.2.1 Set output resolution

【Format】

Request	SET VIDOUT_RES < OUT1 OUT2 > < AUTO <i>Resolution</i> >
Response	VIDOUT_RES < OUT1 OUT2 > < AUTO <i>Resolution</i> >
Description	Instruct the device to change its output resolution as the command designates or automatically. The first parameter indicates which output resolution is set. If you assign <i>AUTO</i> as the <i>Resolution</i> parameter, the device will select a best resolution according to the display's EDID. The list of all available timings

	<p>is below:</p> <pre>3840x2160P@60* 3840x2160P@50* 3840x2160P@30 3840x2160P@25* 3840x2160P@24* 2560x1440P@60* 2560x1440P@30 1920x1080P@60 1920x1080P@50 1920x1080P@30 1680x1050P@60 1600x1200P@60 1440x900P@60 1366x768P@60 1280x1024P@60 1280x800P@60 1280x720P@60 1024x768P@60 720x576P@50 720x480P@60</pre> <p>*The resolution available for OUT1 only</p> <p>As the factory default, both two outputs' resolutions are set AUTO.</p>
--	---

【Example】

To use 4K@60 resolution on the output 1:

Request:

```
SET VIDOUT_RES OUT1 3840x2160P@60
```

Response:

```
VIDOUT_RES OUT1 3840x2160P@60
```

2.2.2 Get output resolution

【Format】

Request	<pre>GET VIDOUT_RES < OUT1 OUT2 USB ALL ></pre>
Response	<pre>VIDOUT_RES < OUT1 OUT2 USB > < Resolution DISCONNECTED > [AUTO]</pre> <p>or</p> <pre>VIDOUT_RES OUT1 < Resolution DISCONNECTED > [AUTO] VIDOUT_RES OUT2 < Resolution DISCONNECTED > [AUTO] VIDOUT_RES USB < Resolution DISCONNECTED ></pre>
Description	<p>Get the output resolution information of the designated output. If the parameter value is ALL, the device prints the resolution information of two outputs in two rows.</p> <p>For each row, the first field is the number of the output, the second field is the actual resolution. The third field AUTO is optional which means the corresponding output resolution is set AUTO.</p> <p>The term DISCONNECTED have two different meanings:</p> <ul style="list-style-type: none"> ◇ The resolution of the HDMI output interface is set AUTO but the output is not connected to a display. ◇ The UVC device of the USB output interface is not opened by the

host computer.

【Example】

Request:

```
GET VIDOUT_RES ALL
```

Response:

```
VIDOUT_RES OUT1 3840x2160@60 AUTO
VIDOUT_RES OUT2 DISCONNECTED AUTO
VIDOUT_RES USB DISCONNECTED
```

2.2.3 Get output connection status

【Format】

Request	GET VIDOUT_CONNECT < OUT1 OUT2 USB ALL >
Response	VIDOUT_CONNECT < OUT1 OUT2 USB > < DISCONNECTED CONNECTED > or VIDOUT_CONNECT OUT1 < DISCONNECTED CONNECTED > VIDOUT_CONNECT OUT2 < DISCONNECTED CONNECTED > VIDOUT_CONNECT USB < DISCONNECTED CONNECTED >
Description	If the parameter designates a specific video output, it means to query whether the specific output is connected to a source, the device prints CONNECTED to indicate the source is connected, and vice versa. if the parameter value is ALL, the device responds with three rows of texts to report the connection status of all video outputs.

【Example】

Request:

```
GET VIDOUT_CONNECT USB
```

Response:

```
VIDOUT_CONNECT USB CONNECTED
```

2.2.4 Set HD mode

【Format】

Request	SET UVC_HDMODE < ON OFF >
----------------	-----------------------------

Response	UVC_HDMODE < ON OFF >
Description	<p>Turn on/off the HD mode of the USB output.</p> <p>USB HOST interface provide video output function with UVC specification, the output resolution can not be designated by the device itself, on the contrary, what the device can do is to provide a list of all supported resolutions, the host computer chooses one from the list.</p> <p>This API can modify the content of the aforementioned list, when the configuration is ON, the list contains the resolutions equals to 720P@30 or higher.</p> <p>As the factory default, HD mode is enabled, the corresponding configuration is ON.</p>

【Example】

Request:

```
SET UVC_HDMODE OFF
```

Response:

```
UVC_HDMODE OFF
```

2.2.5 Get HD mode

【Format】

Request	GET UVC_HDMODE
Response	UVC_HDMODE < ON OFF >
Description	Get the HD mode configuration.

【Example】

Request:

```
GET UVC_HDMODE
```

Response:

```
UVC_HDMODE OFF
```

2.2.6 Set output HDCP configuration

【Format】

Request	SET HDCP < OUT1 OUT2 > < ON OFF >
----------------	---------------------------------------

Response	HDCP < OUT1 OUT2 > < ON OFF >
Description	Set the HDCP feature of a output. As the factory default, HDCP feature is enabled, the corresponding configuration is ON.

【Example】

Request:

```
SET HDCP OUT1 OFF
```

Response:

```
HDCP OUT1 OFF
```

2.2.7 Get output HDCP configuration

【Format】

Request	GET HDCP < OUT1 OUT2 ALL >
Response	HDCP < OUT1 OUT2 > < ON OFF > or HDCP OUT1 < ON OFF > HDCP OUT2 < ON OFF >
Description	Get the HDCP configuration of the designated output. If the parameter value is ALL, the device prints HDCP configuration information of two outputs.

【Example】

Request:

```
GET HDCP OUT1
```

Response:

```
HDCP OUT1 OFF
```

2.2.8 Get input connection status

【Format】

Request	GET VIDIN_CONNECT < Src ALL >
Response	VIDIN_CONNECT < Src > < DISCONNECTED CONNECTED >

	<p>or</p> <pre>VIDIN_CONNECT IN1 < DISCONNECTED CONNECTED > VIDIN_CONNECT IN2 < DISCONNECTED CONNECTED > VIDIN_CONNECT IN3 < DISCONNECTED CONNECTED > VIDIN_CONNECT IN4 < DISCONNECTED CONNECTED > VIDIN_CONNECT NDI1 < DISCONNECTED CONNECTED > VIDIN_CONNECT NDI2 < DISCONNECTED CONNECTED > VIDIN_CONNECT NDI3 < DISCONNECTED CONNECTED > VIDIN_CONNECT NDI4 < DISCONNECTED CONNECTED ></pre>
Description	<p>If the parameter designates a specific video source, it means to query whether the specific input is connected to a source, the device prints <code>CONNECTED</code> to indicate the source is connected, and vice versa.</p> <p>if the parameter value is <code>ALL</code>, the device responds with eight rows of texts to report the connection status of all video sources.</p>

【Example】

Request:

```
GET VIDIN_CONNECT IN2
```

Response:

```
VIDIN_CONNECT IN2 DISCONNECTED
```

2.2.9 Get input signal status

【Format】

Request	<pre>GET VIDIN_SIG < Src ALL ></pre>
Response	<pre>VIDIN_SIG < Src > < NO VALID ></pre> <p>or</p> <pre>VIDIN_SIG IN1 < NO VALID > VIDIN_SIG IN2 < NO VALID > VIDIN_SIG IN3 < NO VALID > VIDIN_SIG IN4 < NO VALID > VIDIN_SIG NDI1 < NO VALID > VIDIN_SIG NDI2 < NO VALID > VIDIN_SIG NDI3 < NO VALID > VIDIN_SIG NDI4 < NO VALID ></pre>
Description	<p>If the parameter designates a specific source, it means to query whether the specific source has valid signal, the device prints <code>VALID</code> to indicate</p>

the source has valid signal, and vice versa.

if the parameter value is `ALL`, the device responds with eight rows of texts to report the signal status of all video sources.

【Example】

Request:

```
GET VIDIN_SIG IN3
```

Response:

```
VIDIN_SIG IN3 VALID
```

2.2.10 Get input video format

【Format】

Request	<pre>GET VIDIN_FORMAT < Src ALL ></pre>
Response	<pre>VIDIN_FORMAT < Src ALL > < INVALID VideoFormat ></pre> <p>or</p> <pre>VIDIN_FORMAT IN1 < INVALID VideoFormat > VIDIN_FORMAT IN2 < INVALID VideoFormat > VIDIN_FORMAT IN3 < INVALID VideoFormat > VIDIN_FORMAT IN4 < INVALID VideoFormat > VIDIN_FORMAT NDI1 < INVALID VideoFormat > VIDIN_FORMAT NDI2 < INVALID VideoFormat > VIDIN_FORMAT NDI3 < INVALID VideoFormat > VIDIN_FORMAT NDI4 < INVALID VideoFormat ></pre>
Description	<p>Query the video format of the designated input, the device prints <code>INVALID</code> to indicate the source has not valid signal, or else the device prints the <code>VideoFormat</code> whose format is:</p> <pre>< Width x Height[p], FrameRate; HDRInfo; ColorSpace; ColorDepth ></pre> <p>The value and meaning of every field are as follows:</p> <ul style="list-style-type: none">✧ <code>Width, Height</code> The width and height of the video. In the actual message, there is no space character between the Width, height and the letter 'x', for an example, <code>1920X1080</code>✧ <code>P</code> It is optional to indicate progressive video✧ <code>FrameRate</code> Frame rate✧ <code>HDRInfo</code> Whether HDR format, now the device does not support HDR, so its value is always <code>NONE HDR</code>.

	<ul style="list-style-type: none"> ◇ <i>ColorSpace</i> Color space, available values are RGB、YCBCR444、YCBCR422 ◇ <i>ColorDepth</i> Color depth, <i>now only</i> 8BIT <p>An actual example of <i>VideoFormat</i>:</p> <pre style="background-color: #cccccc; padding: 2px;">3840X2160P,30;NONE HDR;YCBCR444;8BIT</pre> <p>if the request message parameter is ALL, the device prints the video format information of all video sources.</p>
--	--

【Example】

Request:

```
GET VIDIN_FORMAT IN3
```

Response:

```
VIDIN_FORMAT IN3 3840X2160P,30;NONE HDR;YCBCR444;8BIT
```

2.2.11 Get input audio format

【Format】

Request	<pre style="background-color: #cccccc; padding: 2px;">GET AUDIN_FORMAT < Src ALL ></pre>
Response	<pre style="background-color: #cccccc; padding: 2px;">AUDIN_FORMAT < Src ALL > < INVALID AudioFormat ></pre> <p>or</p> <pre style="background-color: #cccccc; padding: 2px;">AUDIN_FORMAT IN1 < INVALID AudioFormat > AUDIN_FORMAT IN2 < INVALID AudioFormat > AUDIN_FORMAT IN3 < INVALID AudioFormat > AUDIN_FORMAT IN4 < INVALID AudioFormat > AUDIN_FORMAT NDI1 < INVALID AudioFormat > AUDIN_FORMAT NDI2 < INVALID AudioFormat > AUDIN_FORMAT NDI3 < INVALID AudioFormat > AUDIN_FORMAT NDI4 < INVALID AudioFormat ></pre>
Description	<p>Query the audio format of the designated input, the device prints INVALID to indicate the source has not valid audio signal, or else the device prints the <i>AudioFormat</i> whose format is:</p> <pre style="background-color: #cccccc; padding: 2px;">< Encoding; SamplingRate; SamplingWidth ></pre> <p>The value and meaning of every field are as follows:</p> <ul style="list-style-type: none"> ◇ <i>Encoding</i> The audio encoding mode, the available values are PCM, AAC and MP3 ◇ <i>SamplingRate</i> Sampling rate, such as 48KHz

	<p>◇ <i>SamplingWidth</i> Sampling width, the available values are 8BIT and 16BIT.</p> <p>An actual example of <i>AudioFormat</i>:</p> <pre>PCM;48KHz;8BIT</pre> <p>if the request message parameter is ALL, the device prints the audio format information of all video sources.</p>
--	--

【Example】

Request:

```
GET AUDIN_FORMAT IN3
```

Response:

```
AUDIN_FORMAT IN3 PCM;48KHz;8BIT
```

2.2.12 Get Output Connection Status

【Format】

Request	GET VIDOUT_CONNECT < OUT1 OUT2 USB ALL >
Response	<pre>VIDOUT_CONNECT < OUT1 OUT2 USB > < CONNECTED DISCONNECTED ></pre> <p>or</p> <pre>VIDOUT_CONNECT OUT1 < CONNECTED DISCONNECTED ></pre> <pre>VIDOUT_CONNECT OUT2 < CONNECTED DISCONNECTED ></pre> <pre>VIDOUT_CONNECT USB < CONNECTED DISCONNECTED ></pre>
Description	<p>Query the status of the physical video outputs.</p> <p>if the parameter value is ALL, the device responds with status of all three physical video outputs.</p>

【Example】

Request:

```
GET VIDOUT_CONNECT OUT1
```

Response:

```
VIDOUT_CONNECT OUT1 CONNECTED
```

2.2.13 Set USB Device Mode

【Format】

Request	SET USB_DEVICE_MODE < VIDEO ALL >
Response	USB_DEVICE_MODE < VIDEO ALL >
Description	<p>Configure the media data types transmitted on the USB HOST interface. Some customers want to remove audio data from the USB HOST interface, this API instruct the device whether to transmit data on the USB HOST interface.</p> <p>The parameter value ALL means the video and audio data are transmitted, both UVC and UAC devices are declared. On another side, the value VIDEO means only UVC device is declared and audio data is removed.</p> <p>As the factory default, the configuration is ALL.</p>

【Example】

Request:

```
SET USB_DEVICE_MODE VIDEO
```

Response:

```
USB_DEVICE_MODE VIDEO
```

2.2.14 Get USB Device Mode

【Format】

Request	GET USB_DEVICE_MODE
Response	USB_DEVICE_MODE < VIDEO ALL >
Description	Query the media data types transmitted on the USB HOST interface.

【Example】

Request:

```
GET USB_DEVICE_MODE
```

Response:

```
USB_DEVICE_MODE VIDEO
```

2.3 Video switching

2.3.1 Set low latency mode

【Format】

Request	SET LOW_LATENCY_FN < ON OFF >
Response	LOW_LATENCY_FN < ON OFF >
Description	<p>Turn on/off the low latency mode.</p> <p>When this feature is enabled, the device uses 1920x1080 resolution to procession video signals whenever possible, so to achieve as low latency as possible. Here is the list of the items whose resolution affected:</p> <ul style="list-style-type: none"> ✧ Opening USB input ✧ USB output ✧ HDMI output ✧ NDI output (Primary stream) <p>Furthermore, when low latency mode is enabled, the video switching is faster.</p> <p>To employee the higher resolution, please disable this feature.</p> <p>As the factory default, low latency mode is enabled, the corresponding configuration is ON.</p>

【Example】

Request:

```
SET LOW_LATENCY_FN OFF
```

Response:

```
LOW_LATENCY_FN OFF
```

2.3.2 Get low latency mode

【Format】

Request	GET LOW_LATENCY_FN
Response	LOW_LATENCY_FN < ON OFF >
Description	Get the low latency mode configuration.

【Example】

Request:

```
GET LOW_LATENCY_FN
```

Response:

```
LOW_LATENCY_FN OFF
```

2.3.3 Set input to output

【Format】

Request	SET SW < Src NONE > [OUT1 OUT2]
Response	SW < Src NONE > [OUT1 OUT2]
Description	<p>Switch an input to an output, this command instructs the device to display a video source on the specified video output. The specific behavior is as follows:</p> <ul style="list-style-type: none"> ◇ The device displays the video source with full screen mode always, no matter whether or which multiview layout is used currently. ◇ If the request message does not carry the optional parameter, the device display the video source in full screen on all outputs. ◇ If the video source to be displayed is NONE, the device stops displaying video source the guide screen.

【Example】

Request:

```
SET SW IN3 OUT1
```

Response:

```
SW IN3 OUT1
```

2.3.4 Get input of output

【Format】

Request	GET SW < OUT1 OUT2 ALL >
Response	<pre>SW < Src NONE MV > < OUT1 OUT2 ></pre> <p>OR</p> <pre>SW < Src NONE > OUT1</pre> <pre>SW < Src NONE MV > OUT2</pre>
Description	<p>Get the information which video source is being displayed on an output. When the request message uses ALL as the parameter, the device responds with two rows to provide the status of two video output signals (USB output has the same video content with OUT2).</p> <p>In the response message, NONE means no video source is shown on the</p>

output, MV means OUT2 is in multiview state.

【Example】

Request:

```
GET SW OUT1
```

Response:

```
SW IN3 OUT1
```

2.3.5 Set multiview layout

【Format】

Request	SET MV_LAYOUT < <i>LayoutID</i> <i>LayoutName</i> >
Response	MV_LAYOUT < <i>LayoutID</i> <i>LayoutName</i> >
Description	<p>Switch the layout used by multiview mode, multiview feature works on OUT2 only, the parameter <i>LayoutID</i> or <i>LayoutName</i> is used to specify a layout. The IDs and corresponding names of all supported layouts are listed below:</p> <ul style="list-style-type: none">✧ 0x100 FullScreen✧ 0x101 PiP✧ 0x102 SbS✧ 0x103 LBRS✧ 0x104 Pyramid✧ 0x105 Quad <p>As the boot default, single full screen layout is used, the corresponding <i>LayoutID</i> is 0x100 and the <i>LayoutName</i> is FullScreen.</p>

【Example】

Request:

```
SET MV_LAYOUT QUAD
```

Response:

```
MV_LAYOUT QUAD
```

2.3.6 Get multiview layout

【Format】

Request	GET MV_LAYOUT
Response	MV_LAYOUT <i>LayoutID</i> <i>LayoutName</i>
Description	Get the multiview layout used currently, the device returns the ID and name of the layout.

【Example】

Request:

```
GET MV_LAYOUT
```

Response:

```
MV_LAYOUT 0X101 PIP
```

2.3.7 Set multiview layout and video source(s)

【Format】

Request	SET MV_WIN_SRC < <i>LayoutID</i> <i>LayoutName</i> Current > <i>WinNo1</i> < <i>Src1</i> NONE > [<i>WinNo2</i> < <i>Src2</i> NONE >] ...
Response	MV_WIN_SRC < <i>LayoutID</i> <i>LayoutName</i> > <i>WinNo1</i> < <i>Src1</i> NONE > [<i>WinNo2</i> < <i>Src2</i> NONE >] ...
Description	<p>Switch the layout and set video source(s) for the child window(s). The parameter <i>WinNo</i> can use the one of the following two forms:</p> <ul style="list-style-type: none"> ✧ Number Such as 1, 2, 3... ✧ Win1, Win2, Win3... <p>The parameter value NONE means no video source is shown in the corresponding window.</p> <p>If first parameter value of the request message is <i>Current</i>, it means to remain the current layout unchanged.</p> <p>If the designated layout is the same as the current layout, the device change the video source(s) of the window(s) appearing in the parameters only, the rest window(s) will not be impact. In short, this usage equivalent to switching the video source(s) of the specified child window(s).</p>

【Example】

Request:

```
SET MV_WIN_SRC QUAD 1 IN2
```

Response:

```
MV_WIN_SRC QUAD 1 IN2
```

2.3.8 Get multiview layout and video source(s)

【Format】

Request	GET MV_WIN_SRC
Response	MMV_WIN_SRC <i>LayoutName</i> <i>WinNo1</i> < <i>Src1</i> NONE > [<i>WinNo2</i> < <i>Src2</i> NONE >] ...
Description	Get the multiview layout and video source(s) of the child window(s).

【Example】

Request:

```
GET MV_WIN_SRC
```

Response:

```
MV_WIN_SRC SbS WIN1 IN2 WIN2 NONE
```

2.4 Audio processing

2.4.1 Set DSP audio mixing

【Format】

Request	SET_DSP_MIX < <i>DspOut</i> > < <i>DspIn</i> > < ON OFF >
Response	DSP_MIX < <i>DspOut</i> > < <i>DspIn</i> > < ON OFF >
Description	<p>The device has a built-in audio DSP to support audio matrix and mixer functions. It supports 4 input channels (<i>DspIn</i>: USBIN, HDMIIN, USBHOST, ANALOGIN) and 4 output channels (<i>DspOut</i>: USBIN, HDMIOUT, USBHOST, ANALOGOUT). The audio mixing between any input channel and any output channel is achieved.</p> <p>This API is used to set whether the specified audio input channel participates in the mixing of the specified audio output channel. To avoid audio loop or self-excitation, the following mixing relationships are prohibited:</p> <ul style="list-style-type: none">✧ USBIN --> USBIN✧ USBHOST --> USBHOST✧ ANALOGIN --> ANALOGOUT <p>As the factory default, the USBHOST channel participates the mixing of the USBIN channel, the mixing configuration of the corresponding crosspoint</p>

is ON, for the remaining crosspoints, the configurations are both OFF.

【Example】

Request:

```
SET DSP_MIX HDMIOUT HDMIIN ON
```

Response:

```
DSP_MIX HDMIOUT HDMIIN ON
```

2.4.2 Get DSP audio mixing

【Format 1】

Request	GET DSP_MIX < <i>DspOut</i> > < <i>DspIn</i> >
Response	DSP_MIX < <i>DspOut</i> > < <i>DspIn</i> > < ON OFF >

【Format 2】

Request	GET DSP_MIX < <i>DspOut</i> ALL >
Response	DSP_MIX < <i>DspOut</i> > [< <i>DspIn1</i> > [<i>DspIn2</i>] ...] or DSP_MIX USBIN [< <i>DspIn1</i> > [<i>DspIn2</i>] ...] DSP_MIX HDMIOUT [< <i>DspIn1</i> > [<i>DspIn2</i>] ...] DSP_MIX USBHOST [< <i>DspIn1</i> > [<i>DspIn2</i>] ...] DSP_MIX ANALOGOUT [< <i>DspIn1</i> > [<i>DspIn2</i>] ...]

【Comment】

Description	<p>The format 1 is used to get the mixing configuration of the specified audio input channel and the specified audio output channel.</p> <p>The format 2 can get the mixing configuration of the specified audio output channel, the response message lists all audio input channels which participate the audio mixing of the specified audio output channel.</p> <p>If the request message parameter value is ALL, it means to get the mixing configuration of all audio output channels.</p>
--------------------	---

【Example 1】

Request:

```
GET DSP_MIX USBIN USBHOST
```

Response:

```
DSP_MIX USBIN USBHOST ON
```

【Example 2】

Request:

```
GET DSP_MIX ALL
```

Response:

```
DSP_MIX USBIN USBHOST
DSP_MIX HDMIOUT HDMIIN
DSP_MIX USBHOST
DSP_MIX ANALOGOUT HDMIIN
```

2.4.3 Set DSP input channel

【Format】

Request	SET DSP_AUD_SRC << USBIN < IN1 IN2 >> < HDMIIN < IN3 IN4 >>>
Response	DSP_AUD_SRC << USBIN < IN1 IN2 >> < HDMIIN < IN3 IN4 >>>
Description	<p>Only one audio signal of the two USB inputs can be sent to the audio DSP for processing, the two HDMI inputs have the same limitation too. This API chooses the one audio signal (sent to the audio DSP) from the two USB or HDMI inputs, or in another word, the chosen audio signal will be sent to the USBIN or HDMIIN channel of the audio DSP.</p> <p>As the factory default, IN1 and IN3 are chosen for USBIN and HDMIIN respectively.</p>

【Example】

Request:

```
SET DSP_AUD_SRC USBIN IN1
```

Response:

```
DSP_AUD_SRC USBIN IN1
```

2.4.4 Get DSP input channel

【Format】

Request	GET DSP_AUD_SRC < USBIN HDMIIN ALL >
Response	DSP_AUD_SRC << USBIN < IN1 IN2 >> < HDMIIN < IN3 IN4 >>> or DSP_AUD_SRC USBIN < IN1 IN2 > DSP_AUD_SRC HDMIIN < IN3 IN4 >

Description	If the request message parameter value is ALL, it means to get the input channel configurations of USBIN and HDMIIN together.
--------------------	---

【Example】

Request:

```
GET DSP_AUD_SRC USBIN
```

Response:

```
DSP_AUD_SRC USBIN IN1
```

2.4.5 Set DSP input mute

【Format】

Request	SET DSP_AUD_GAIN_MUTE < <i>DspIn</i> > < ON OFF >
Response	DSP_AUD_GAIN_MUTE < <i>DspIn</i> > < ON OFF >
Description	Set whether the specified audio input channel is muted. As the factory default, all channels are not mute, the configuration value is OFF.

【Example】

Request:

```
SET DSP_AUD_GAIN_MUTE USBIN ON
```

Response:

```
DSP_AUD_GAIN_MUTE USBIN ON
```

2.4.6 Get DSP input mute

【Format】

Request	GET DSP_AUD_GAIN_MUTE < <i>DspIn</i> ALL >
Response	DSP_AUD_GAIN_MUTE < <i>DspIn</i> > < ON OFF > or DSP_AUD_GAIN_MUTE USBIN < ON OFF > DSP_AUD_GAIN_MUTE HDMIIN < ON OFF > DSP_AUD_GAIN_MUTE USBHOST < ON OFF > DSP_AUD_GAIN_MUTE ANALOGIN < ON OFF >
Description	If the request message parameter value is ALL, it means to get the mute

configurations of all input channels.

【Example】

Request:

```
GET DSP_AUD_GAIN_MUTE USBIN
```

Response:

```
DSP_AUD_GAIN_MUTE USBIN ON
```

2.4.7 Set DSP input gain

【Format】

Request	SET DSP_AUD_GAIN < <i>DspIn</i> > < <i>Gain</i> >
Response	DSP_AUD_GAIN < <i>DspIn</i> > < <i>Gain</i> >
Description	<p>The unit of the parameter <i>Gain</i> is dB and the available range is [-20, +30], when the device processes this API request, the mute configuration of the corresponding input channel is disabled (OFF) too.</p> <p>As the factory default, the gain is not adjusted, namely, <i>Gain</i> is 0.</p>

【Example】

Request:

```
SET DSP_AUD_GAIN USBIN 3
```

Response:

```
DSP_AUD_GAIN USBIN 3
```

2.4.8 Get DSP input gain

【Format】

Request	GET DSP_AUD_GAIN < <i>DspIn</i> ALL >
Response	<p>DSP_AUD_GAIN < <i>DspIn</i> > < <i>Gain</i> ></p> <p>or</p> <pre>DSP_AUD_GAIN USBIN < <i>Gain1</i> > DSP_AUD_GAIN HDMIIN < <i>Gain2</i> > DSP_AUD_GAIN USBHOST < <i>Gain3</i> > DSP_AUD_GAIN ANALOGIN < <i>Gain4</i> ></pre>
Description	If the request message parameter value is ALL, it means to get the gain

configurations of all input channels.

【Example】

Request:

```
GET DSP_AUD_GAIN USBIN
```

Response:

```
DSP_AUD_GAIN USBIN 3
```

2.4.9 Set DSP output mute

【Format】

Request	SET DSP_AUD_VOL_MUTE < <i>DspOut</i> > < ON OFF >
Response	DSP_AUD_VOL_MUTE < <i>DspOut</i> > < ON OFF >
Description	Set whether the specified audio output channel is muted. As the factory default, all channels are not mute, the configuration value is OFF.

【Example】

Request:

```
SET DSP_AUD_VOL_MUTE USBHOST ON
```

Response:

```
DSP_AUD_VOL_MUTE USBHOST ON
```

2.4.10 Get DSP output mute

【Format】

Request	GET DSP_AUD_VOL_MUTE < <i>DspOut</i> ALL >
Response	DSP_AUD_VOL_MUTE < <i>DspOut</i> > < ON OFF > or DSP_AUD_VOL_MUTE USBIN < ON OFF > DSP_AUD_VOL_MUTE HDMIOUT < ON OFF > DSP_AUD_VOL_MUTE USBHOST < ON OFF > DSP_AUD_VOL_MUTE ANALOGOUT < ON OFF >
Description	If the request message parameter value is ALL, it means to get the mute configurations of all output channels.

【Example】

Request:

```
GET DSP_AUD_VOL_MUTE USBHOST
```

Response:

```
DSP_AUD_VOL_MUTE USBHOST ON
```

2.4.11 Set DSP output volume

【Format】

Request	SET DSP_AUD_VOL < <i>DspOut</i> > < <i>Volume</i> >
Response	DSP_AUD_VOL < <i>DspOut</i> > < <i>Volume</i> >
Description	<p>The unit of the parameter <i>Volume</i> is dB and the available range is [-100, 0], when the device processes this API request, the mute configuration of the corresponding output channel is disabled (OFF) too.</p> <p>As the factory default, the gain is not adjusted, namely, <i>Volume</i> is 0.</p>

【Example】

Request:

```
SET DSP_AUD_VOL HDMIOUT -6
```

Response:

```
DSP_AUD_VOL HDMIOUT -6
```

2.4.12 Get DSP output volume

【Format】

Request	GET DSP_AUD_VOL < <i>DspOut</i> ALL >
Response	DSP_AUD_VOL < <i>DspOut</i> > < <i>Gain</i> > or DSP_AUD_VOL USBIN < <i>Gain1</i> > DSP_AUD_VOL HDMIOUT < <i>Gain2</i> > DSP_AUD_VOL USBHOST < <i>Gain3</i> > DSP_AUD_VOL ANALOGOUT < <i>Gain4</i> >
Description	If the request message parameter value is ALL, it means to get the volume configurations of all output channels.

【Example】

Request:

```
GET DSP_AUD_VOL HDMIOUT
```

Response:

```
DSP_AUD_VOL HDMIOUT -6
```

2.5 Standby related & peripheral control

2.5.1 Set standby mode

【Format】

Request	SET STANDBY < ON OFF >
Response	STANDBY < ON OFF >
Description	Manually instruct the device to enters or exits standby mode. ON means power on (exiting standby mode) and OFF means power off (entering standby mode).

【Example】

Request:

```
SET STANDBY OFF
```

Response:

```
STANDBY OFF
```

2.5.2 Get standby mode

【Format】

Request	GET STANDBY
Response	STANDBY < ON OFF >
Description	Query whether the device is in standby mode.

【Example】

Request:

```
GET STANDBY
```

Response:

```
STANDBY OFF
```

2.5.3 Set automatic standby

【Format】

Request	SET AUTO_STANDBY_FN < ON OFF >
Response	AUTO_STANDBY_FN < ON OFF >
Description	Set whether the automatic standby feature is enabled. As the factory default, automatic standby feature is enabled, the configuration value is ON.

【Example】

Request:

```
SET AUTO_STANDBY_FN OFF
```

Response:

```
AUTO_STANDBY_FN OFF
```

2.5.4 Get automatic standby

【Format】

Request	GET AUTO_STANDBY_FN
Response	AUTO_STANDBY_FN < ON OFF >
Description	Query the state of the automatic standby configuration

【Example】

Request:

```
GET AUTO_STANDBY_FN
```

Response:

```
AUTO_STANDBY_FN OFF
```

2.5.5 Set automatic standby timeout

【Format】

Request	SET AUTO_STANDBY_D < <i>TimeOut</i> >
----------------	---------------------------------------

Response	AUTO_STANDBY_D < <i>TimeOut</i> >
Description	<p>The parameter <i>Timeout</i> indicates how long the device will be idle (namely, displaying guide screen) before entering standby mode, its unit is second. The value 0 means entering standby mode immediately when the device starts being idle.</p> <p>Apparently this configuration takes effect only when the automatic standby feature is enabled.</p> <p>The value range is from 0 to 3600.</p> <p>As the factory default, automatic standby timeout is 2 minutes, the configuration value is set 120.</p>

【Example】

Request:

```
SET AUTO_STANDBY_D 360
```

Response:

```
AUTO_STANDBY_D 360
```

2.5.6 Get automatic standby timeout

【Format】

Request	GET AUTO_STANDBY_D
Response	AUTO_STANDBY_D < <i>TimeOut</i> >
Description	Query the value of the automatic standby timeout

【Example】

Request:

```
GET AUTO_STANDBY_D
```

Response:

```
AUTO_STANDBY_D 360
```

2.5.7 Set CEC command

【Format】

Request	SET CECCMD_EDIT < OUT1 OUT2 > < ON OFF > < <i>CECCode</i> DEFAULT >
Response	CECCMD_EDIT < OUT1 OUT2 > < ON OFF > < <i>CECCode</i> DEFAULT >

Description	<p>Configure the stored CEC command used to power on/off the display.</p> <p>The first parameter indicates which output to be set. The second parameter indicates which command will be changed. The third parameter <i>CECCode</i> is the actual CEC message string with hexadecimal format, no space between the adjacent bytes.</p> <p>DEFAULT means to restore to the default CEC code.</p> <p>As the factory default, the power on (ON) message is 4004 and power off (OFF) message is FF36.</p>
--------------------	---

【Example】

Request:

```
SET CECCMD_EDIT OUT1 ON 4004
```

Response:

```
CECCMD_EDIT OUT1 ON 4004
```

2.5.8 Get CEC command

【Format】

Request	GET CECCMD_EDIT < < OUT1 OUT2 > < ON OFF > ALL >
Response	<pre>CECCMD_EDIT < OUT1 OUT2 > < ON OFF > < CECCode ></pre> <p>Or</p> <pre>CECCMD_EDIT OUT1 ON < CECCode ></pre> <pre>CECCMD_EDIT OUT1 OFF < CECCode ></pre> <pre>CECCMD_EDIT OUT2 ON < CECCode ></pre> <pre>CECCMD_EDIT OUT2 OFF < CECCode ></pre>
Description	<p>Query a CEC command configuration.</p> <p>If the request message parameter value is ALL, the device returns all configurations in four rows.</p>

【Example】

Request:

```
GET CECCMD_EDIT OUT1 ON
```

Response:

```
CECCMD_EDIT OUT1 ON 4004
```

2.5.9 Set RS232 work mode

【Format】

Request	SET UART_MODE < API COM >
Response	UART_MODE < API COM >
Description	<p>The RS232 port supports different work modes, the value and meanings of the parameter are as follows:</p> <ul style="list-style-type: none">◇ API Receive the API commands sent by the external controller◇ COM Conventional communication, control the peripherals such as display, participate in standby related power on/off display operations. <p>As the factory default, the RS232 port is used to control the peripherals the configuration value is COM.</p>

【Example】

Request:

```
SET UART_MODE API
```

Response:

```
UART_MODE API
```

2.5.10 Get RS232 work mode

【Format】

Request	GET UART_MODE
Response	UART_MODE < API COM >
Description	Query the work mode of the RS232 port.

【Example】

Request:

```
GET UART_MODE
```

Response:

```
UART_MODE API
```

2.5.11 Set power on/off display via RS232

【Format】

Request	SET UARTPWR_FN < ON OFF >
Response	UARTPWR_FN < ON OFF >
Description	<p>When the device enters or exits standby mode, in additions to powering on/off the display through CEC messages, the device also can send the corresponding commands through RS232 port. This API is used to configure whether to send RS232 command when powering on/off the display.</p> <p>Note: The actual activation of sending power on/off command through RS232 port requires three conditions to be met:</p> <ul style="list-style-type: none">✧ The RS232 port work mode is COM, namely, control peripherals✧ The RS232 ON or OFF message is configured✧ This configuration is ON <p>As the factory default, this configuration is set OFF, when the device enters or exits standby mode, the device sends power on/off command through CEC messages only.</p>

【Example】

Request:

```
SET UARTPWR_FN ON
```

Response:

```
UARTPWR_FN ON
```

2.5.12 Get power on/off display via RS232

【Format】

Request	GET UARTPWR_FN
Response	UARTPWR_FN < ON OFF >
Description	Query the configuration.

【Example】

Request:

```
GET UARTPWR_FN
```

Response:

2.5.13 Configure RS232 settings

【Format】

Request	SET UART_CFG < <i>BaudRate</i> > < <i>Parity</i> > < <i>DataBit</i> > < <i>StopBit</i> >
Response	UART_CFG < <i>BaudRate</i> > < <i>Parity</i> > < <i>DataBit</i> > < <i>StopBit</i> >
Description	<p>Modify the RS232 port settings, the values and corresponding meanings of the parameters are below:</p> <ul style="list-style-type: none"> ◇ <i>BaudRate</i> Baud rate, available values are 9600, 19200, 38400, 57600, 115200 ◇ <i>Parity</i> NONE, ODD, EVEN ◇ <i>DataBit</i> Number of data bits, 7 or 8 ◇ <i>StopBit</i> Number of stop bits, 1 or 2 <p>As the factory default, the value of the above settings are 115200, NONE, 8, 1 respectively.</p>

【Example】

Request:

```
SET UART_CFG 9600 NONE 8 1
```

Response:

```
UART_CFG 9600 NONE 8 1
```

2.5.14 Get RS232 setting

【Format】

Request	GET UART_CFG
Response	UART_CFG < <i>BaudRate</i> > < <i>Parity</i> > < <i>DataBit</i> > < <i>StopBit</i> >
Description	Query the settings of the RS232 port.

【Example】

Request:

```
GET UART_CFG
```

Response:

```
UART_CFG 9600 NONE 8 1
```

2.5.15 Set RS232 baud rate

【Format】

Request	SET UART_B < <i>BaudRate</i> >
Response	UART_B < <i>BaudRate</i> >
Description	Configure the RS232 port baud rate separately, the available values are 9600, 19200, 38400, 57600, 115200. As the factory default, the baud rate is 115200.

【Example】

Request:

```
SET UART_B 9600
```

Response:

```
UART_B 9600
```

2.5.16 Get RS232 baud rate

【Format】

Request	GET UART_B
Response	UART_B < <i>BaudRate</i> >
Description	Query the baud rate of the RS232 port.

【Example】

Request:

```
GET UART_B
```

Response:

```
UART_B 9600
```

2.5.17 Set RS232 command

【Format】

Request	SET UART_CMD < <i>CmdName</i> > < HEX STR > < <i>CmdStr</i> >
----------------	---

Response	UART_CMD < <i>CmdName</i> > < HEX STR > < <i>CmdStr</i> >
Description	<p>Configure the message string sent to the peripherals through the RS232 port. The values and meanings of the parameters are as follows:</p> <ol style="list-style-type: none"> 1. The <i>CmdName</i> parameter indicates which command to be configured. Except ON and OFF (used to power on / off the display), the caller can define new command such as VolumeUp, VolumeDown, but the command name can't contain space. If the specified command has not been set before, it will be added, otherwise the existing command will be modified. 2. The second parameter indicates the format of the subsequent parameter <i>CmdStr</i>. HEX means hexadecimal and STR means printable string. 3. The third parameter is the actual command string. <p>◇ For hexadecimal format, there is no space between the adjacent bytes. For example:</p> <pre>112233445566</pre> <p>◇ For string format, the spaces among the words will be regarded as the part of the command string. For example</p> <pre>Power on</pre> <p>As the factory default, no RS232 message is set.</p>

【Example】

Request:

```
SET UART_CMD ON STR PowerOn
```

Response:

```
UART_CMD ON STR PowerOn
```

2.5.18 Delete RS232 command

【Format】

Request	SET UART_CMD_D < <i>CmdName</i> >
Response	UART_CMD_D < <i>CmdName</i> >
Description	Delete a defined RS232 command.

【Example】

Request:

```
SET UART_CMD_D ON
```

Response:

```
UART_CMD_D ON
```

2.5.19 Get RS232 command

【Format】

Request	GET UART_CMD < <i>CmdName</i> ALL >
Response	UART_CMD < <i>CmdName</i> > < HEX STR > < <i>CmdStr</i> > Or UART_CMD ON < HEX STR > < <i>CmdStr</i> > UART_CMD OFF < HEX STR > < <i>CmdStr</i> > ...
Description	Query a RS232 command configuration, if the request message parameter value is ALL, the device returns all defined RS232 messages row by row.

【Example】

Request:

```
GET UART_CMD ON
```

Response:

```
UART_CMD ON STR PowerOn
```

2.5.20 Send RS232 command

【Format】

Request	SET UART_CMD_S < <i>CmdName</i> >
Response	UART_CMD_S < <i>CmdName</i> >
Description	Send a command through the RS232 port.

【Example】

Request:

```
SET UART_CMD_S ON
```

Response:

```
UART_CMD_S ON
```

2.5.21 Send RS232 data

【Format】

Request	SET UART_S < HEX STR > < <i>CmdStr</i> >
Response	UART_S < HEX STR > < <i>CmdStr</i> >
Description	Send a command through the RS232 port. With this API, the controller can send any RS232 data directly, no need to use the API SET UART_CMD < <i>CmdName</i> > < HEX STR > < <i>CmdStr</i> > to define a RS232 command in advance.

Request:

```
SET UART_S STR PowerOn
```

Response:

```
UART_S STR PowerOn
```

2.5.22 Send CEC or/and RS232 command

【Format】

Request	SET SEND_CMD < ON OFF <i>CmdName</i> > [RS232 CEC ALL]
Response	SEND_CMD < ON OFF <i>CmdName</i> > [RS232 CEC ALL]
Description	Instruct the device to send CEC or/and RS232 command. The first parameter is the name of sent command, the second optional parameter indicates the sending method. If the sending method is not specified, the configuration of the API SET UARTPWR_FN < ON OFF > is used. If the RS232 work mode is not COM or the specified RS232 command is not defined, the device does not send RS232 command.

【Example】

Request:

```
SET SEND_CMD ON CEC
```

Response:

```
SEND_CMD ON CEC
```

2.5.23 Send CEC data

【Format】

Request	SET CEC_CMD < OUT1 OUT2 > < CECCode >
Response	CEC_CMD < OUT1 OUT2 > < CECCode >
Description	Send CEC data directly, this API can send any data through the CEC channel conveniently.

【Example】

Request:

```
SET CEC_CMD OUT1 4004
```

Response:

```
CEC_CMD OUT1 4004
```

2.6 NDI input & output

2.6.1 Set NDI RX group name

【Format】

Request	SET NDI_RX_GROUP < <i>GroupName</i> >
Response	NDI_RX_GROUP < <i>GroupName</i> >
Description	Configure the group name of the NDI RX module, according to NDI specification, only the NDI units in the same group can discover each other. As the factory default, the group name is <code>Public</code> .

【Example】

Request:

```
SET NDI_RX_GROUP MeetingRoom1
```

Response:

```
NDI_RX_GROUP MeetingRoom1
```

2.6.2 Get NDI RX group name

【Format】

Request	GET NDI_RX_GROUP
Response	NDI_RX_GROUP < <i>GroupName</i> >

Description	Obtain the group name of the NDI RX module.
--------------------	---

【Example】

Request:

```
GET NDI_RX_GROUP
```

Response:

```
NDI_RX_GROUP MeetingRoom1
```

2.6.3 Set NDI RX device name

【Format】

Request	SET NDI_RX_DEV_NAME < <i>DeviceName</i> >
Response	NDI_RX_DEV_NAME < <i>DeviceName</i> >
Description	Configure the device name of the NDI RX module. As the factory default, the device name is HUB402-xxxx-RX, xxxx is the hexadecimal format of the last two digits of the device MAC address.

【Example】

Request:

```
SET NDI_RX_DEV_NAME CamMixer
```

Response:

```
NDI_RX_DEV_NAME CamMixer
```

2.6.4 Get NDI RX device name

【Format】

Request	GET NDI_RX_DEV_NAME
Response	NDI_RX_DEV_NAME < <i>DeviceName</i> >
Description	Obtain the device name of the NDI RX module.

【Example】

Request:

```
GET NDI_RX_DEV_NAME
```

Response:

```
NDI_RX_DEV_NAME CamMixer
```

2.6.5 Get NDI device list

【Format】

Request	GET NDI_SCAN_DEV_LIST < TX RX ALL >
Response	NDI_SCAN_DEV_LIST < TX RX > < DeviceName1 > < IPAddress1 > NDI_SCAN_DEV_LIST < TX RX > < DeviceName2 > < IPAddress2 > ...
Description	Obtain the list of devices scanned in the network. The request message can specify the type of the obtained devices.

【Example】

Request:

```
GET NDI_SCAN_DEV_LIST TX
```

Response:

```
NDI_SCAN_DEV_LIST TX Team1 169.254.34.71
```

```
NDI_SCAN_DEV_LIST TX Team2 169.254.179.80
```

2.6.6 Get NDI channel list

【Format】

Request	GET NDI_SCAN_CHN_LIST < DeviceName TX RX ALL >
Response	NDI_SCAN_CHN_LIST < DeviceName > < TX RX > < ChannelName1 > [ChannelName2 ...] Or NDI_SCAN_CHN_LIST < DeviceName1 > < TX RX > < ChannelName11 > [ChannelName12 ...] NDI_SCAN_CHN_LIST < DeviceName2 > < TX RX > < ChannelName21 > [ChannelName22 ...]
Description	According to NDI specification, one NDI RX or TX module can have multiple channels and each channel corresponds to a video source (TX) or decoder (RX). In fact, most NDI TX or RX modules have only one channel. This API is used to obtain the channel list of a devices (or a type of devices or all devices). Each row of the response message consists of device name, type and all channels' name.

【Example】

Request:

```
GET_NDI_SCAN_CHN_LIST Team1
```

Response:

```
NDI_SCAN_CHN_LIST Team1 TX Leader Member
```

2.6.7 Set NDI video source

【Format】

Request	SET_NDI_IN_SRC < 1 2 3 4 > TXDeviceName [ChannelName]
Response	NDI_IN_SRC < 1 2 3 4 > TXDeviceName [ChannelName]
Description	<p>Set a NDI (RX) decoder receive and decode the stream of a NDI TX channel. The outputs of NDI decoder is used as the inputs of the subsequent video switcher, namely, <i>NdiSrc</i> (NDIIN1~4).</p> <p>The first parameter of the request message specifies a NDI decoder, the second and third parameter specify a NDI TX channel. If the parameter <i>ChannelName</i> is omitted, the first channel will be used.</p>

【Example】**Request:**

```
SET_NDI_IN_SRC 1 Team1 Leader
```

Response:

```
NDI_IN_SRC 1 Team1 Leader
```

2.6.8 Delete NDI video source

【Format】

Request	SET_NDI_IN_SRC < 1 2 3 4 >
Response	NDI_IN_SRC < 1 2 3 4 >
Description	Delete a NDI video source, the specified NDI (RX) decoder stops receiving or decoding the video stream and the corresponding <i>NdiSrc</i> becomes invalid too.

【Example】**Request:**

```
SET_NDI_IN_SRC 1
```

Response:

2.6.9 Get NDI video source

【Format】

Request	GET NDI_IN_SRC < 1 2 3 4 ALL >
Response	<p>NDI_IN_SRC < 1 2 3 4 > [< TXDeviceName > < ChannelName >]</p> <p>Or</p> <p>NDI_IN_SRC 1 [< TXDeviceName1 > < ChannelName1 >]</p> <p>NDI_IN_SRC 2 [< TXDeviceName2 > < ChannelName2 >]</p> <p>NDI_IN_SRC 3 [< TXDeviceName3 > < ChannelName3 >]</p> <p>NDI_IN_SRC 4 [< TXDeviceName4 > < ChannelName4 >]</p>
Description	Get the video source configuration of a certain NDI decoder or all NDI decoders. In the response message, if there is not <i>TXDeviceName</i> or <i>ChannelName</i> fields, it means the corresponding NDI decoder is not set a NDI source.

【Example】

Request:

```
GET NDI_IN_SRC 1
```

Response:

```
NDI_IN_SRC 1 Team1 Leader
```

2.6.10 Set NDI TX group name

【Format】

Request	SET NDI_TX_GROUP < GroupName >
Response	NDI_TX_GROUP < GroupName >
Description	<p>Configure the group name of the NDI TX module, according to NDI specification, only the NDI units in the same group can discover each other.</p> <p>As the factory default, the group name is <code>Public</code>.</p>

【Example】

Request:

```
SET NDI_TX_GROUP MeetingRoom1
```

Response:

```
NDI_TX_GROUP MeetingRoom1
```

2.6.11 Get NDI TX group name

【Format】

Request	GET NDI_TX_GROUP
Response	NDI_TX_GROUP < <i>GroupName</i> >
Description	Obtain the group name of the NDI TX module.

【Example】

Request:

```
GET NDI_TX_GROUP
```

Response:

```
NDI_TX_GROUP MeetingRoom1
```

2.6.12 Set NDI TX device name

【Format】

Request	SET NDI_TX_DEV_NAME < <i>DeviceName</i> >
Response	NDI_TX_DEV_NAME < <i>DeviceName</i> >
Description	Configure the device name of the NDI TX module. As the factory default, the device name is HUB402-xxxx-TX, xxxx is the hexadecimal format of the last two digits of the device MAC address.

【Example】

Request:

```
SET NDI_TX_DEV_NAME Teacher
```

Response:

```
NDI_TX_DEV_NAME Teacher
```

2.6.13 Get NDI TX device name

【Format】

Request	GET NDI_TX_DEV_NAME
Response	NDI_TX_DEV_NAME < DeviceName >
Description	Obtain the device name of the NDI TX module.

【Example】

Request:

```
GET NDI_TX_DEV_NAME
```

Response:

```
NDI_TX_DEV_NAME Teacher
```

2.6.14 Set NDI TX channel name

【Format】

Request	SET NDI_TX_CHN_NAME < ChannelName >
Response	NDI_TX_CHN_NAME < ChannelName >
Description	Configure the channel name of the NDI TX module. As the factory default, the device name is NDI.

【Example】

Request:

```
SET NDI_TX_CHN_NAME NDIOut1
```

Response:

```
NDI_TX_CHN_NAME NDIOut1
```

2.6.15 Get NDI TX channel name

【Format】

Request	GET NDI_TX_CHN_NAME
Response	NDI_TX_CHN_NAME < ChannelName >
Description	Obtain the channel name of the NDI TX module.

【Example】

Request:

```
GET NDI_TX_CHN_NAME
```

Response:

2.6.16 Set NDI TX encoding parameter

【Format】

Request	SET NDI_TX_VENC_CFG < NDI11 NDI12 > < EncodeMode > < Resolution > < FrameRate > < BitRateCtl > < BitRate > < GopSize >
Response	NDI_TX_VENC_CFG < NDI11 NDI12 > < EncodeMode > < Resolution > < FrameRate > < BitRateCtl > < BitRate > < GopSize >
Description	<p>Set the encoding parameter of NDI TX. Here are the details of the parameters:</p> <ul style="list-style-type: none"> ◇ The first parameter indicates which NDI stream to configure. It takes one of two values: NDI11 and NDI12. Their formats are NDI followed by two digits. The first digit is the NDI output number. The device currently supports only one NDI output, so this digit is always 1. The second digit is used to distinguishes the main stream from the preview stream, 1 and 2 correspond to the main stream and preview stream respectively. ◇ <i>EncodeMode</i> Encoding method, allowed values are H.264 and H.265. ◇ <i>Resolution</i> Encoding resolution, the optional values varies on the stream: Main stream(NDI11): 1280×720, 1920×1080, 3840×2160 Preview stream(NDI12): 320×240, 640×360 ◇ <i>FrameRate</i> Encoding frame rate, the range is 1 to 60 and the unit is fps. ◇ <i>BitRateCtl</i> Bitrate-control mode, CBR or VBR. ◇ <i>BitRate</i> Bitrate, the unit is kbps, the range varies on the stream: Main stream(NDI11): 64~16384 Preview stream(NDI12): 64~4096 ◇ <i>GopSize</i> Group-of-Pictures size, i.e., I-frame interval, the range is 1~150. <p>Here are the factory defaults:</p> <p>Main stream(NDI11):</p> <pre>H.264 1920x1080 30 cbr 4096 30</pre>

	Preview stream(NDI12):
	H.264 320x240 30 cbr 1024 30
	Note:
	Currently, The frame rate cannot be set arbitrarily, it must be determined automatically by the encoding module, so the parameter <i>FrameRate</i> is ignored now.

【Example】

Request:

```
SET NDI_TX_VENC_CFG NDI11 H.264 1920X1080 30 CBR 8192 60
```

Response:

```
NDI_TX_VENC_CFG NDI11 H.264 1920X1080 30 CBR 8192 60
```

2.6.17 Get NDI TX encoding parameter

【Format】

Request	SET NDI_TX_VENC_CFG < NDI11 NDI12 ALL >
Response	NDI_TX_VENC_CFG < NDI11 NDI12 > < EncodeMode > < Resolution > < FrameRate > < BitRateCtl > < BitRate > < GopSize > Or NDI_TX_VENC_CFG NDI11 < EncodeMode > < Resolution > < FrameRate > < BitRateCtl > < BitRate > < GopSize > NDI_TX_VENC_CFG NDI12 < EncodeMode > < Resolution > < FrameRate > < BitRateCtl > < BitRate > < GopSize >
Description	Get the encoding parameter of NDI TX, if the parameter value of the request message is ALL, the device returns with two rows corresponding to the two streams.

【Example】

Request:

```
GET NDI_TX_VENC_CFG NDI11
```

Response:

```
NDI_TX_VENC_CFG NDI11 H.264 1920X1080 30 CBR 8192 60
```

2.6.18 Set NDI stream output

【Format】

Request	SET NDI_TX_STREAM_FN < NDI11 NDI12 > < ON OFF >
Response	NDI_TX_STREAM_FN < NDI11 NDI12 > < ON OFF >
Description	<p>Set the NDI TX stream output feature, namely, turn on/off the NDI stream output. NDI11 and NDI12 correspond to main stream and preview stream respectively.</p> <p>As the factory default, both two streams are enabled, the configuration value is ON.</p>

【Example】

Request:

```
SET NDI_TX_STREAM_FN NDI12 OFF
```

Response:

```
NDI_TX_STREAM_FN NDI12 OFF
```

2.6.19 Get NDI stream output

【Format】

Request	GET NDI_TX_STREAM_FN < NDI11 NDI12 ALL >
Response	<pre>NDI_TX_STREAM_FN < NDI11 NDI12 > < ON OFF ></pre> <p>Or</p> <pre>NDI_TX_STREAM_FN NDI11 < ON OFF ></pre> <pre>NDI_TX_STREAM_FN NDI12 < ON OFF ></pre>
Description	Obtain the status of the NDI TX stream output feature.

【Example】

Request:

```
GET NDI_TX_STREAM_FN NDI12
```

Response:

```
NDI_TX_STREAM_FN NDI12 OFF
```