API Command Guide

HDIP100E
HDIP100D
4KIP200E
4KIP200D
4KIP204E
4KIP200M

AV Access
Simplify your life

Contents

1. Overview

This document briefly introduces the API commands of the following encoders and decoders for developers to develop client control applications (hereinafter referred to as "client App"):

- 4K Products:
  - ✓ 4KIP200E -- Response is displayed as "IPE935", hereinafter referred to as "TX".
  - ✓ 4KIP200D -- Response is displayed as "IPD935", hereinafter referred to as "RX".
  - ✓ 4KIP204E -- Response is displayed as "IPE9354", hereinafter referred to as "TX".
  - ✓ 4KIP200M – Response is displayed as "IPM4000", hereinafter referred to as "MRX".
- 2K Products:
  - ✓ HDIP100E -- Response is displayed as "IPE915", hereinafter referred to as "TX".
  - ✓ HDIP100D -- Response is displayed as "IPD915", hereinafter referred to as "RX".

2. Fundamental Knowledge

- **Device Name:** hereinafter referred to as "hostname", consists of the device type and the 12-digit MAC address of the network card, each of which is connected to another via the hyphen "-", e.g., "IPE935-341B22822FEF". Since the MAC address is unique globally, it is used to represent the corresponding device ID in the API interface sometimes.
- **Key-Value Pair**: The devices can use the extra storage space to store some custom parameters of the application layer, such as the device alias, which exists in the form of K-V pairs and will not be used by the devices. The client app uses *gbparam s key value* or *gbparam g key* to set and/or read these parameters.

  Note: The custom parameters shall not overlap with the device's parameters that have been defined.
- **MRX**: Denotes the device that supports Multiview, and specifically refers to 4KIP200M in this document. Compared to other RX devices, MRX features the capability to receive multiple video streams from TXs and mix them according to the predefined layout instructions, and then output the videos in Multiview form on a single screen. The features available for RXs in this document also apply to MRX unless otherwise noted.

3. Requirements of the Developers

- Familiar with all features and functions of devices listed in this document.
- Proficient in the knowledge of TCP/IP network communication programming plus TCP, UDP, TELNET and HTTP protocols.
- Experienced in approaching Linux operating system through the console.

4. How the Client App Interacts with Devices

The following protocols are available for the devices:

- UDP, contains UDP broadcast and UDP unicast that mainly used for device discovery.
- Telnet, used to log in to and interact with the device by sending shell commands after login. The interaction content includes information acquisition, setting of storage and sending of control command.
- HTTP, mainly used to obtain preview video of the TX, upload Idle Image on RX side, and upgrade firmware.

5. Before You Start

This document assumes that the following devices are deployed for the networking:

- ✓ Several units of TX
- ✓ Several units of RX
- ✓ One Gigabit Switch with support of broadcast message communication.
  - Note: Generally, most switches support broadcast message communication unless it is disabled intentionally.
  - Since TX and RX do not use multicast for communication, it is not required for the switch to support multicast.
- ✓ One device running the client app.

Important: You need to plan IP addresses for your entire system based on the actual situation, which can be any of the following types:

- Static IP address (Static)

  You can assign address and netmask for each TX and RX in the way mentioned hereinafter.

- Dynamic IP address (DHCP)

  Dynamic IP address divides into the two following situations:

  ⇨ IP address assigned by a DHCP server

  In this situation, there must be a device in the system that runs a DHCP server, which assigns automatically and dynamically IP addresses for any devices in the system.

  ⇨ No DHCP server

  There's no DHCP server in the system, the TX or RX automatically generates an IP address (169.254.X.X) and subnet mask (255.255.0.0) for itself.

  Generally, after all above devices are powered on, each RX will pair with a certain TX by default even no client app is involved.

6. Discovering Devices

After the client app is started, perform the following to search for the TX and RX devices in the network.

(1) Send the following UDP broadcast message to the broadcast address 255.255.255.255 or the subnet broadcast address (calculated based on the IP address and netmask), and the port number 3335. The message format is below:

Name	Type	Byte Size	Description
device_type	UInt32	4	Value:0
device_function	UInt32	4	Value:0

It is recommended to set an appropriate timer according to the quantity of the devices, and send this message regularly to capture changes of network devices in time.

(2) Listen on the network card for UDP messages sent to the local port number 3336. The message format is below:

Name	Type	Byte Size	Description
Reserved	UInt32	4	
Reserved	UInt32	4	
Reserved	Char	32	
Device_name	Char	64	Device name, including the ending '\0'
Reserved	UInt32	4	
Protocol version	UInt32	4	
Reserved	Char	32	
Reserved	Char	32	
Reserved	Uint32	4	

Reserved	UInt32	4	
Reserved	Char	16	
Reserved	Char	96	

The *Device_name* (hereinafter also referred to as "hostname") can be obtained through message parsing, and the sending device's IP address can be obtained according to the UDP message.

Supported devices: All devices.

7. Functions Based on Telnet Session

As methods mentioned above, after the client app obtains the IP address of a certain TX or RX, it is able to log in to the device console through Telnet protocol and establish Telnet session.

In this chapter, the Telnet client program running in Windows is used to simulate the client app for explaining various functions based on Telnet through which the client app interacts with devices.

7.1 Functions for TX and RX

7.1.1 Logging in to the Device

Command Structure: *telnet ip_address 24*

Description: *ip_address* denotes the IP address of a device found by this device. You can determine whether login is successful according to the response of "/#".

Supported devices: All devices

Command Example:

PS C:\> telnet 169.254.138.14 24
IPE935-341B22822FEF login: root
Welcome.
/ #

7.1.2 Setting and Saving Device Alias

The client app can set device alias and save it to the device for future reading.

Note: The device provides a space for storing this parameter only, and will not use this parameter.

Command Structure:

gbparam s alias xxxx
gbparam g alias

Description: *alias* denotes the custom field of the API, *xxxx* denotes device's alias. It is recommended to use "alias" directly; if not, please obtain the list of the device's all parameters through *gbparam dump* before you use other field name in case that it is as same as the device's existed parameter.

Supported devices: All devices

Command Example:

/ # gbparam s alias sonyDVD
/ # gbparam g alias
sonyDVD/ #

7.1.3 Setting/Getting IP Address and Netmask for/from Device

Command Structure:

gbparam s ip_mode IPMODE
gbparam s ipaddr IPADDR
gbparam s netmask NETMASK
gbparam g ip_mode
gbparam g ipaddr
gbparam g netmask

Description:

- *IPMODE*: IP addressing mode – autoip | static | dhcp (default)

  If IPMODE = dhcp while no DHCP server is deployed in the network, the device will select autoip until a DHCP server is running.
- *IPADDR*: IP address, e.g. 192.168.10.254
- *NETMASK*: Netmask, e.g. 255.255.255.0

Note:

- Each of the above command is not effective until the device restarts.
- The value obtained through *gbparam g* is not necessarily the one that is effective currently; for example, the device is restored to the factory defaults after the previous setting value became effective.
- If IP_MODE = DHCP, there is no need to set IP address and netmask.

Supported devices: All devices

Command Example:

/ # gbparam s ip_mode dhcp

/ # gbparam s ipaddr 192.168.10.254

/ # gbparam s netmask 255.255.255.0

/ # reboot

After logging in to the device again, query the above settings:

/ # gbparam g ip_mode

dhcp

/ # gbparam g ipaddr

192.168.10.254

/ # gbparam g netmask

255.255.255.0

/ #

7.1.4 Operating the Serial Port

(1) Usage of the serial port

The device's serial port can be used in the following two applications:

a) Application 1: Serial commands pass through between TX and RX.

b) Application 2: The third-party client software communicates with the device and sends commands to the peripheral connected to the device's serial port.

(2) Set serial parameters for passing through serial commands

Command Structure: *soip2 -S -b sss-dps*

Description:

- ⇨ *sss* denotes the baud rate ranging (50, ..., 115200)
- ⇨ *d* denotes the data bits ranging (5, 6, 7, 8)
- ⇨ *p* denotes the parity setting, and the range is N = none; E = even; O = odd
- ⇨ *s* denotes the stop digits ranging (1, 2)

Note:

- If commands are to be passing through between TX and RX, their serial port parameters should be set to the same.
- The default serial parameter of the device is 115200-8n1.

Command Example:

To set baud rate to 9600, data bits to 8, parity bit to none, stop bits to 1:

/ # soip2 -S -b 9600-8n1

(3) The third-party client sends commands to the peripheral connected to the device's Serial Port

Command Structure: *soip2 -f /dev/ttyS0 -b sss-dps [-r] [-n] -s "CONTENT" [-H]*

Description:

- ⇨ *CONTENT*: The string to be sent
- ⇨ *-H*: Optional, denotes the hex string
- ⇨ *-r:* Add a terminator of carriage-return <CR> for the command
- ⇨ *-n*: Add a terminator of line-feed <LF> for the command
- ⇨ *-b sss-dps*: Same as that in Set serial parameters for passing through serial commands section.

Note:

- "*-f /dev/ttyS0*" is a fixed item.
- Each command must include "-b sss-dps".

Command Example:

a) Send a sequence of ASCII-formatted strings terminated with a carriage-return to a peripheral:

/# soip2 -f /dev/ttyS0 -b 115200-8n1 -r -s "12345"

b) Send a sequence of hex strings terminated with a line-feed to a peripheral:

/# soip2 -f /dev/ttyS0 -b 115200-8n1 -n -s "4c 6f 72 65 6d 20 69 70 73 75 6d 20 64 6f 6c 6f 72 20 73 69 74 20 61 6d 65 74 2c 20 63 6f 6e 73 65 63 74 65 74 75 72 20 61 64 69 70 69 73 63 69 6e 67 20 65 6c 69 74 2c 20 73 65 64 20 64 6f 20 65 69 75 73 6d 6f 64 20 74 65 6d 70 6f 72 20 69 6e 63 69 64 69 64 75 6e 74 20 75 74 20 6c 61 62 6f 72 65 20 65 74 20 64 6f 6c" -H

Supported devices: All devices (Note: 4KIP200M doesn't support the application of passing through serial commands.)

7.1.5 Rebooting Device

Command Structure: reboot

Supported Devices: All devices

Command Example:

/ # reboot

```
/ #
```

7.1.6 Getting Firmware Version

Command Structure: *cat /etc/version*

Supported Devices: All devices

Command Example:

```
/ # cat /etc/version
IPE935
V1.0.23t1
Wed, 17 May 2023 07:39:21 +0000
/ #
```

7.1.7 Restoring Device to Factory Defaults

Command Structure: *reset_to_default.sh;reboot*

Supported Devices: All devices

Command Example:

```
/ # reset_to_default.sh;reboot
sh: can't kill pid 3045: No such process
/ #
```

7.2 Functions for TX Only

7.2.1 Setting HDCP

By default, HDCP is enabled on all TX devices, and can be disabled if required for specific scenarios.

Command Structure: *gbconfig --hdcp-enable VALUE*

Description: *VALUE* – y | n

Supported Devices: All TX devices

Command Example:

```
/ # gbconfig --hdcp-enable y
/ # gbconfig --show --hdcp-enable
y
/ #
```

7.3 Functions for RX Only

7.3.1 Setting Communication Modes for Controlling Display On/Off via RX

Command Structure: *gbconfig --sinkpower-mode VALUE*

Description: *VALUE* – cec | rs232 | both, default setting is cec.

Supported Devices: All RX devices

Command Example:

/ # gbconfig --sinkpower-mode rs232

7.3.2 Setting CEC Commands for Controlling Display On/Off

The device will memorize these two CEC commands, and send them to the connected display correspondingly after receiving a certain command for controlling display on/off from the client software.

Command Structure:

gbparam s cec_poweron_cmd "XXXX"

gbparam s cec_standby_cmd "YYYY"

Description:

⇨ *XXXX*: CEC command for controlling display on; default setting is *40 04*

⇨ *YYYY*: CEC command for controlling display off; default setting is *ff 36*

CEC commands for controlling display on and off may vary on different display devices, for more information, see the display's user guide.

Supported Devices: All RX devices

Command Example:

/ # gbparam s cec_poweron_cmd "40 04"
/ # gbparam s cec_standby_cmd "ff 36"
/ #

7.3.3 Setting RS232 Commands for Controlling Display On/Off

The device will memorize these two RS232 commands, and send them to the connected display correspondingly after receiving a certain command for controlling display on/off from the client app. The RS232 command to be sent shall be indicated whether it is a hex string.

Command Structure:

gbconfig --rs232-hex-cmd-enable [y|n]
gbparam s rs232_poweron_cmd "XXXX"
gbparam s rs232_standby_cmd "YYYY"

Description:

⇨ XX*XX:* RS232 command for controlling display on.

⇨ *YYYY*: RS232 command for controlling display off.

RS232 commands for controlling display on and off may vary on different display devices, for more information, see the display's user guide.

Supported Devices: All RX devices

Command Example:

/ # gbconfig --rs232-hex-cmd-enable n
/ # gbparam s rs232_poweron_cmd "XXXX"
/ # gbparam s rs232_standby_cmd "YYYY"
/ #

7.3.4 Instructing Device to Control Display On/Off

Command Structure:

Command of controlling display on: *sinkpower on*

Command of controlling display on: *sinkpower off*

Description:

Once any of these two commands is sent to the device, the device will automatically convert it into the corresponding CEC or RS232 command based on the settings in chapter 7.3.1, 7.3.2 and 7.3.3 and forward it to the connected display.

Supported Devices: All RX devices

Command Example:

/ # sinkpower off
/ # sinkpower on
/ #

7.3.5 Sending Specific CEC Commands to the Display

Command Structure:

cec -s "ADDR OPCODE; ADDR OPCODE; ..."

Description:

This command is used to request the device to send a string of CEC commands to the display. These different CEC commands can be separated with semicolons ";".

Note:

Available CEC commands for display devices may vary among vendors, for more information, see the display's user guide.

Supported Devices: All RX devices

Command Example:

/ # cec -s "40 04"
/ #

7.3.6 Assigning a Source to RX

By default, each RX device automatically searches for online TX devices, and pairs with one of them in the following methods:

(1) RX automatically pairs with one TX.

This function is mostly used in the scenarios of extenders.

(2) RX is paired with one TX manually.

RX pairs with each TX in turn automatically by long pressing the info/source button on the RX's front panel for more than 2 seconds. Source switching occurs once each time the button is being long pressed.

This function is mostly used in the small-scale networking scenarios.

(3) RX is paired with one TX through API commands.

Once API command for pairing is sent to the device, the above two pairing methods will become invalid unless the device is restored to factory defaults.

This function is used in large-scale networking scenarios.

Command Structure:

gbconfig --source-select=SOURCE
gbconfig --vsource-select=SOURCE
gbconfig --asource-select=SOURCE
gbconfig --ssource-select=SOURCE
e e_reconnect

Description:

⇨ *SOURCE*: TX's MAC address

- *--vsource-select=SOURCE*: Assign a video source to the RX
- *--asource-select=SOURCE*: Assign an audio source to the RX
- *--ssource-select=SOURCE*: Assign an RS232 source to the RX.

Note: After an RS232 source is assigned to a certain RX, the client app can still send

RS232 commands to the peripheral at RX's serial port through the soip2 program.

- *--source-select=SOURCE:* Route the video source, audio source and RS232 source of one TX as a whole to the RX. The *SOURCE* is the TX's MAC address.

If the audio source is not specified, it follows the video source by default.

If the RS232 source is not specified, it follows the video source by default.

If *SOURCE* = null, the specific source will be unbound from the RX.

Note: You need to send "*e e_reconnect*" at the end to make the command take effect.

Supported Devices: All RX devices, except MRX (MRX has specific commands).

Command Example:

/ # gbconfig --source-select=341B22822FEF
/ # e e_reconnect
/ # gbconfig --source-select=NULL
/ # e e_reconnect
/ #

7.3.7 Getting the Source Assigned to RX

Command Structure:

To get the whole source (video, audio and RS232) assigned to the RX: */ # gbconfig --show --source-select*

To get the audio source assigned to the RX: */ # gbconfig --show --asource-select*

To get the RS232 source assigned to the RX: */ # gbconfig --show --ssource-select*

To get the video source assigned to the RX: */ # gbconfig --show --vsource-select*

Description: N/A

Supported Devices: All RX devices, except MRX (MRX has specific commands)

Command Example:

/ # gbconfig --show --source-select
341B22822FEA
/ # gbconfig --show --asource-select
NULL
/ # gbconfig --show --ssource-select
NULL
/ # gbconfig --show --vsource-select
341B22822FEA

Note: The above returns the MAC address of the TX.

7.3.8 Rotating Video

Command Structure:

e e_vw_rotate_N
e e_reconnect

Description:

⇨ *N:* 0 | 90 | 180 | 270, denotes the video rotation degree in counterclockwise direction.

Supported Devices: 4KIP200D only

Command Example:

/ # e e_vw_rotate_270
270
/ # e e_reconnect
/ #

7.3.9 Video Wall

Note: Video Wall settings are not available for MRX.

7.3.9.1 Standard Video Wall

For a standard video wall, you only need to set the size of the video wall and the position of the display attached to this RX in the video wall.

Command Structure:

e e_vw_enable_M_N_X_Y
e e_reconnect

Description:

⇨ M = Number of the video wall rows -1

⇨ N = Number of the video wall columns -1.

⇨ X = Number of the rows in which RX locates -1

⇨ Y = Number of the columns that RX locates -1

Note:

- You need to send "e e_reconnect" in the end to make the above command take effect.
- If M=N=X=Y=0, the video wall configuration will be undone.

Supported Devices: 4KIP200D only

Command Example:

To set up a 2 x 2 video wall of which the display attached to this RX is located in the

upper left corner, complete the following:

/ # e e_vw_enable_1_1_0_0
/ # e e_reconnect

To undone the video wall, perform the following:

/ # e e_vw_enable_0_0_0_0
/ # e e_reconnect
/ #

7.3.9.2 Mosaic Style Video Wall

Setting up a mosaic style video wall requires devices to support video cropping and rotation.

- Video cropping command

  Command Structure:

  gbparam s xy_param X1:Y1:X2:Y2

  Description:

  X1:Y1:X2:Y2 denotes that only the video content within the rectangular area formed by the coordinate (X1,Y1) in upper left and the coordinate (X2,Y2) in bottom right is presented. Note that this command does not factor in the input video's horizontal and vertical sizes but assumes that they are both 10000, and the device will covert *X1/Y1/X2/Y2* to actual sizes.

  Supported Devices: All RX devices

  Command Example:

  The following command is used to output one-fourths of the video content in upper left corner.

  / # gbparam s xy_param 0:0:5000:5000
  / # e e_reconnect
  / #

- Video rotation command: See chapter Rotating Video.

Based on the combination of two commands above, the client app is able to set up a mosaic style video wall in which TVs in first row rotates by 180 degrees. The VDirector App we provide can realize this function as well.

Command Example:

The rotate the display by 180 degrees counterclockwise around the upper left corner and make it present one-fourths of the input video's content in the bottom right only:

/ # gbparam s xy_param 5000:5000:10000:10000
/ # e e_vw_rotate_180
180
/ # e e_reconnect
/ #

The rotate the display by 90 degrees counterclockwise around the upper left corner and make it present the content within the rectangular area [(0,0),(900,1600)] in the upper left corner of the input video:

/ # gbparam s xy_param 0:0:900:1600
/ # e e_vw_rotate_270
270
/ # e e_reconnect
/ #

7.3.10 Instructing RX to Force a Resolution Output

Command Structure:

gbset fvo RESOLUTION
e e_reoutput

Description:

The value of *RESOLUTION*:

a) auto
b) 3840x2160_60
c) 3840x2160_30
d) 1080P_60
e) 720P_60

Note: Only commonly used resolutions are listed here. By default, it is set as *auto*.

Supported Devices: HDIP100D doesn't support (b) and (c), 4KIP200D doesn't support (b), 4KIP200M supports above all.

Command Example:

/ # gbset fvo 3840x2160_30
/ # e e_reoutput
/ #

7.3.11 Instructing RX to Force Color Space (RGB or YUV) Output

Command Structure:

gbparam s fource_output_color_space COLORSPACE
e e_reoutput

Description:

⇨ COLORSPACE = auto | yuv | rgb. Default setting is auto.

Supported Devices: All RX devices

Command Example:

/ # gbparam s fource_output_color_space yuv
/ # e e_reoutput
/ # gbparam s fource_output_color_space rgb
/ # e e_reoutput
/ # gbparam s fource_output_color_space auto
/ # e e_reoutput
/ # gbparam g fource_output_color_space
auto/ #

7.3.12 Closing the Output of Idle Image (Image reads "No Source") on Detecting No Source

Some TVs or certain devices will take custom action based on whether they detect valid signal input. In this case, users can close the RX's output when RX detects no signal input.

Command Structure:

gbparam s no_source_close_screen VALUE
sleep 1;reboot

Description:

⇨ *VALUE* = y | n.

This command is not effective until the device restarts.

Supported Devices: 4KIP200D, 4KIP200M

Command Example:

/ # gbparam s no_source_close_screen y
/ # sleep 1;reboot
After re-login in to the device:
/ # gbparam g no_source_close_screen
y/ #

7.3.13 Obtaining Display EDID

Command Structure:

cat /var/tmpfs/monitor_info

Description:

To obtain the EDID data of the display attached to the RX's HDMI OUT.

Supported Devices: All RX devices

Command Example:

```
/ # cat /var/tmpfs/monitor_info
  00 ff ff ff | ff ff ff 00 | 61 a9 03 b0 | 01 00 00 00 |
  12 1f 01 03 | 80 3c 21 78 | 2a 1f 65 a4 | 55 50 9f 26 |
  0c 50 54 bd | cf 00 71 4f | 81 80 81 8c | 81 00 95 00 |
  95 0f a9 c0 | b3 00 02 3a | 80 18 71 38 | 2d 40 58 2c |
  45 00 56 50 | 21 00 00 1e | 00 00 00 fd | 00 32 4b 0f |
  64 13 00 0a | 20 20 20 20 | 20 20 00 00 | 00 ff 00 33 |
  30 38 37 38 | 30 30 30 36 | 36 38 38 33 | 00 00 00 fc |
  00 52 65 64 | 6d 69 20 32 | 37 20 4e 46 | 0a 20 01 c3 |
  02 03 1e f1 | 49 01 03 12 | 13 04 14 05 | 1f 10 67 03 |
  0c 00 10 00 | 00 26 23 09 | 07 07 83 01 | 00 00 01 1d |
  00 72 51 d0 | 1e 20 6e 28 | 55 00 56 50 | 21 00 00 1e |
  66 21 56 aa | 51 00 1e 30 | 46 8f 33 00 | 56 50 21 00 |
  00 1e e0 45 | 80 c8 70 38 | 2d 40 30 28 | 55 00 56 50 |
  21 00 00 1e | 0e 1f 00 80 | 51 00 1e 30 | 30 20 37 00 |
  56 50 21 00 | 00 1e 00 00 | 00 00 00 00 | 00 00 00 00 |
  00 00 00 00 | 00 00 00 00 | 00 00 00 00 | 00 00 00 26 |

/ #
```

7.3.14 Overlaying Translucent Texts on Video

Command Structure:

osd_show [-o INDEX -s "CONTENT" [-f FONTSIZE] -p POSITION | -c INDEX]

Description:

- ⇨ -o INDEX: To index the newly added text with a number "INDEX" for later cleanup.
- ⇨ -s CONTENT: To specify the text content to be displayed.
- ⇨ -f FONTSIZE: To specify the font size, default setting is 16, range 1-500, optional.
- ⇨ -p POSITION: To specify the display position of the text. The position contains values of an X-coordinate and a Y-coordinate which are separated by a comma ",". Maximum value for each is 10000.

  Note: The actual maximum value is affected by the length and font size of the text and

shall be determined according to the application scenarios on site.

⇨ -c INDEX: To undo the display of the text indexed with the number "INDEX".

Supported Devices: All RX devices

Command Example:

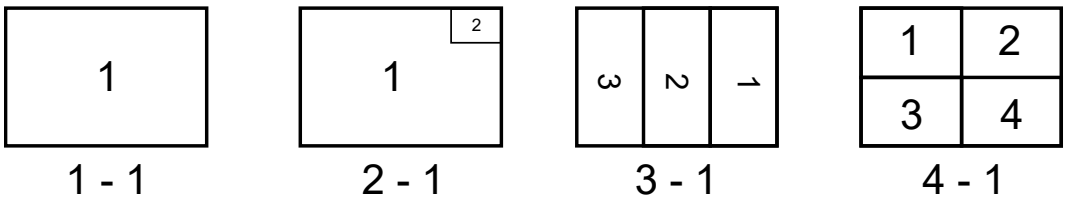
/ # osd_show -o 1 -s "1234" -p 9500,9800
/ # osd_show -o 1 -s "1234" -p 1000,500
/ # osd_show -o 2 -s "1234" -p 2000,500
/ # osd_show -o 3 -s "1234" -p 3000,500
/ # osd_show -o 4 -s "1234" -p 5000,500
/ # osd_show -c 4
/ # osd_show -c 3
/ # osd_show -c 2
/ # osd_show -c 1
/ # osd_show -o 1 -s "1234" -f 300 -p 2000,500
/ #

7.4 Functions for MRX Only

7.4.1 Design Overview

4KIP200M Supports decoding on up to four video streams and one audio stream.

The device supports the following four inbuilt screen layouts, in which each video image's location, size and quantity is fixed.



Diag 1 -- Four screen layouts

1. 1 – 1 (Single view layout)
2. 2 – 1 (Picture-in-Picture layout)
3. 3 – 1 (Portrait triple view layout)
4. 4 – 1 (Quad view layout)

The main points of screen layout setting are as follows:

- The "1-1", "3-1" and "4-1" layouts can only work in tile mode, while layout "2-1" can only work in overlay mode, for setting of the display mode see the following commands.
- The MRX with factory default settings will enter Auto layout state after startup. It automatically

selects one layout from the above based on the quantity of online encoders it detects, and outputs corresponding video sources and a certain audio stream in the selected layout. The audio of the HDMI source 1 located in the layout is automatically selected and being played.

- At factory default, if the MRX detects the front panel Layout button is being pressed, or receives commands of switching sources and screen layouts from a third-party Client App, it will turn from Auto layout to Manual layout status. The MRX cannot fall back to Auto layout status until it is reset to factory defaults.
- The MRX cycles among the four layouts on detecting the front panel Layout button is being pressed. Meanwhile, it will notify the outside of the layout changes by broadcast. It notifies not only when the layout changes but also when the corresponding video source is online or offline even if the layout remains unchanged. The third-party client app can send specific commands to MRX to obtain the latest information, change screen layout and select desired video sources.

  Based on the key points above, the interaction between the third-party client app and MRX comprises of three steps:

  a) The app obtains the MRX's four inbuilt layouts and current layout's information after it starts.

  b) Listens the MRX's notifications on current layout change.

  c) Sends commands to change video sources or layout type of the MRX.

7.4.2 Selecting Video Sources

Command Structure:

1. Set "1-1" layout and specify a video source for the MRX:
   multiview set TX1:1:0_0_1920_1080:stretch:0
   multiview set mode tile
   multiview apply
2. Set "2-1" layout and specify certain video source(s) for the MRX:
   multiview set TX1:1:0_0_1920_1080:stretch:0 TX2:1:1440_0_480_270:stretch:0
   multiview set mode overlay
   multiview apply
3. Set "3-1" layout and specify certain video source(s) for the MRX:
   Multiview set TX1:1:1280_0_640_1080:stretch:270 TX2:1:640_0_640_1080:stretch:270
   TX3:1:0_0_640_1080:stretch:270
   multiview set mode tile
   multiview apply
4. Set "4-1" layout and specify certain video source(s) for the MRX:

multiview set TX1:1:0_0_960_540:stretch:0 TX2:1:960_0_960_540:stretch:0
TX3:1:0_540_960_540:stretch:0 TX4:1:960_540_960_540:stretch:0
multiview set mode tile
multiview apply

Description:

TX1, TX2, TX3, TX4: Denotes the hostname of the video source, *null* denotes deselects the corresponding source for the MRX.

Supported Devices: 4KIP200M

Command Example:

1. Set "1-1" layout and select the source IPE935-341B22822FEF for the MRX:

/ # multiview set IPE935-341B22822FEF:1:0_0_1920_1080:stretch:0
/ # multiview set mode tile
/ # multiview apply

2. In the "1-1" layout, deselects the source:

/ # multiview set NULL:1:0_0_1920_1080:stretch:0
/ # multiview set mode tile
/ # multiview apply

3. Set "2-1" layout, and select the source IPE935-341B22822FEF as video source 1 and null as video source 2 for the MRX:

/ # multiview set IPE935-341B22822FEF:1:0_0_1920_1080:stretch:0 NULL:1:1440_0_480_270:stretch:0
/ # multiview set mode overlay
/ # multiview apply

4. Set "3-1" layout, and select the source IPE935-341B22822FEF as video source 1, 2 and 3 for the MRX:

/ # multiview set IPE935-341B22822FEF:1:1280_0_640_1080:stretch:270 IPE935-341B22822FEF:1:640_0_640_1080:stretch:270 IPE935-341B22822FEF:1:0_0_640_1080:stretch:270
/ # multiview set mode tile
/ # multiview apply

5. Set "4-1" layout, and select the source IPE935-341B22822FEF as video source 1 and null as video source 2, 3 and 4 for the MRX:

/ # multiview set IPE935-341B22822FEF:1:0_0_960_540:stretch:0 NULL:1:960_0_960_540:stretch:0 NULL:1:0_540_960_540:stretch:0 NULL:1:960_540_960_540:stretch:0
/ # multiview set mode tile
/ # multiview apply

7.4.3 Obtaining Layout Information

Command Structure:

multiview get layout

Description:

The MRX responds with a JSON message after it receives the above command, all the client app needs to do is to parse the following information:

1. The four pre-defined layouts' name.
2. Current layout name.
3. The corresponding video sources and audio sources in each layout.

For more information see the following command example (contents related to the point 1/2/3 is highlighted in red):

Command Example:


```
/ # multiview get layout
multivie layout {
  "currentLayout" : "4-1",
  "data" : {
    "aes" : false,
    "hsize" : 1920,
    "layouts" : [
      {
        "audio" : {
          "mode" : "window",
          "source" : "1"
        },
        "layoutseq" : 1,
        "mode" : "tile",
        "name" : "1-1",
        "rotate" : 0,
        "type" : "fixed",
        "windows" : [
          {
            "hsize" : 1920,
            "hstart" : 0,
            "layerseq" : 1,
            "mode" : "stretch",
            "name" : "1",
            "rotate" : 0,
            "tx" : "IPE935-361B22094013",
            "vsize" : 1080,
            "vstart" : 0
          }
        ]
      },
      {
        "audio" : {
          "mode" : "window",
          "source" : "1"
```

```
  },
  "layoutseq" : 2,
  "mode" : "overlay",
  "name" : "2-1",
  "rotate" : 0,
  "type" : "fixed",
  "windows" : [
    {
      "hsize" : 1920,
      "hstart" : 0,
      "layerseq" : 1,
      "mode" : "stretch",
      "name" : "1",
      "rotate" : 0,
      "tx" : "IPE935-341B2281632D",
      "vsize" : 1080,
      "vstart" : 0
    },
    {
      "hsize" : 480,
      "hstart" : 1440,
      "layerseq" : 2,
      "mode" : "stretch",
      "name" : "2",
      "rotate" : 0,
      "tx" : "IPE935-341B2281632D",
      "vsize" : 270,
      "vstart" : 0
    }
  ]
},
{
  "audio" : {
    "mode" : "window",
    "source" : "1"
  },
  "layoutseq" : 3,
  "mode" : "tile",
  "name" : "3-1",
  "rotate" : 90,
  "type" : "fixed",
  "windows" : [
    {
      "hsize" : 640,
      "hstart" : 1280,
      "layerseq" : 1,
      "mode" : "stretch",
```

```
      "name": "1",
      "rotate": 270,
      "tx": "IPE35-361B22094013",
      "vsize": 1080,
      "vstart": 0
    },
    {
      "hsize": 640,
      "hstart": 640,
      "layerseq": 2,
      "mode": "stretch",
      "name": "2",
      "rotate": 270,
      "tx": "IPE935-341B22822FDC",
      "vsize": 1080,
      "vstart": 0
    },
    {
      "hsize": 640,
      "hstart": 0,
      "layerseq": 3,
      "mode": "stretch",
      "name": "3",
      "rotate": 270,
      "tx": "NULL",
      "vsize": 1080,
      "vstart": 0
    }
  ]
},
{
  "audio": {
    "mode": "window",
    "source": "1"
  },
  "layoutseq": 4,
  "mode": "tile",
  "name": "4-1",
  "rotate": 0,
  "type": "fixed",
  "windows": [
    {
      "hsize": 960,
      "hstart": 0,
      "layerseq": 1,
      "mode": "stretch",
      "name": "1",
```

```
        "rotate" : 0,
        "tx" : "IPE935-341B22822FDC",
        "vsize" : 540,
        "vstart" : 0
      },
      {
        "hsize" : 960,
        "hstart" : 960,
        "layerseq" : 2,
        "mode" : "stretch",
        "name" : "2",
        "rotate" : 0,
        "tx" : "IPE935-341B22822FDC",
        "vsize" : 540,
        "vstart" : 0
      },
      {
        "hsize" : 960,
        "hstart" : 0,
        "layerseq" : 3,
        "mode" : "stretch",
        "name" : "3",
        "rotate" : 0,
        "tx" : "NULL",
        "vsize" : 540,
        "vstart" : 540
      },
      {
        "hsize" : 960,
        "hstart" : 960,
        "layerseq" : 4,
        "mode" : "stretch",
        "name" : "4",
        "rotate" : 0,
        "tx" : "IPE935-341B22822FDC",
        "vsize" : 540,
        "vstart" : 540
      }
    ]
  }
],
"trueName" : "IPM4000-5ECC51593001",
"userdefine" : [
  {
    "maxwinsnum" : 8,
    "mode" : "tile"
  },
```

```
        {
          "maxwinsnum" : 6,
          "mode" : "overlay"
        }
      ],
      "vsize" : 1080
    }
  }
```


7.4.4 Receiving Notifications of Layout Change

Command Structure:

config IPM4000:MRX-hostname_get layout

Description:

The command is sent from MRX to the third-party client software through subnet broadcast.

- ⇨ MRX-hostname: The device's hostname.
- ⇨ Destination address: The subnet broadcast address of the subnet on which MRX resides.
- ⇨ Destination port: 11002

Supported Devices: 4KIP200M

Command Example:

Assumes that the third-party client app and the MRX are in the same subnet, the client app IP address is 192.168.1.10, mark is 255.255.0.0, then the subnet broadcast address can be calculated to 192.168.255.255, therefore the client app needs to receive the following UDP message at the address 192.168.255. 255:11002:

config IPM4000:IPM4000-341B22000019_get layout

The client app is required to obtain the layout information from the corresponding MRX as soon as possible on detecting this message.

8. Switching Multiple RXs to a Source Simultaneously

If multiple RXs are required to switch to the same source simultaneously, like switching for a video wall, complete the following:

Command Structure:

To send the following UPD message to the broadcast address 255.255.255.255 or port 5010 of the

subnet broadcast address:

msg_b_reconnect tx_name:session_number:rx_number rx1 rx2 …rxN

Description:

- ⇨ tx_name: The name of the TX connected to the source; the format should be the model no. that device responses + MAC address
- ⇨ session_number: An integer that increases progressively from 1 for distinguishing different source switching sessions (Each simultaneous switching of source is described as a session).
- ⇨ rx_number: The quantity of the RXs

Supported Devices: All RX devices

Command Example:

To send the following message to 255.255.255.255:5010:

msg_b_reconnect IPE935-341B22822FEA:1:2 IPD935-341B228007BD IPD935-341B2282302C

9. Getting Preview Stream of TX

The client app can obtain the preview stream of TX in MJPEG format through a web browser or HTTP protocol.

Command Structure:

http://tx_ip_address/stream

Description:

- ⇨ tx_ip_address: The IP address of the TX whose preview stream is to be obtained.

A TX can output up to 4 preview streams.

Supported Devices: All TX devices

10. Uploading Idle Image to RX

Command Structure:

HTTP Method: POST
HTTP URL: http://rx_ip_address/upload_bg

Description:

- ⇨ rx_ip_address: The IP address of RX.

The uploaded image will replace the factory default image and become the new factory default.

The image format must be jpg.

Supported Devices:

4KIP200D/4KIP200M: Supports 1080P image only.

HDIP100D*: Supports 720P image only.

*Note: Different batches of HDIP100D may vary in terms of the Idle image's supported resolution, for more information contact AV Access.

11. Uploading an Image to Overlay on Video

RX supports overlay an image (.png) on the video to achieve special effects of displaying logos or advertisements.

11.1 Uploading Image

Command Structure:

HTTP Method POST
URI http://rx_ip_address/upload_png

Description:

⇨ rx_ip_address: The IP address of RX.

⇨ The image must be in png format.

Supported Devices:

4KIP200D/4KIP200M: Supports images to the maximum size of 1080P.

HDIP100D*: Supports images to the maximum size of 720P.

*Note: Different batches of HDIP100D may vary in terms of the overlaying image's supported resolution, for more information contact AV Access.

11.2 Displaying Images

Command Structure:

gbconfig --png-overlay-pos-h=POSH
gbconfig --png-overlay-pos-v=POSV
gbconfig --png-overlay-enable=ENABLE

Description:

⇨ POSH: The abscissa value in which the image locates horizontally, range: 0~1919

⇨ POSV: The ordinate value in which the image locates vertically, range: 0~1079

⇨ ENABLE = "y" | "n"

After the above three commands are sent, you need to continue to send the following command for the overlay setting to take effect.

e e_png_overlay

Supported Devices: All RX devices

12. Upgrading Firmware

Command Structure:

HTTP Method: POST

HTTP URL: http://rx_ip_address/upload

Description: rx_ip_address denotes the RX's IP address (applicable to TX as well)

Supported Devices:

All devices except part of HDIP100E and HDIP100D.

Note: Some batches of HDIP100E and HDIP100D may not support firmware upgrade through commands above, for more information contact AV Access.

13. Document Revision History

Version	Date	Description
V1.0.0	230522	Initial
V1.0.1	230822	Add support for 4KIP200M
V1.0.2	230911	Add *e e_png_overlay* command
V1.0.3	230912	Add "Operating the Serial Port" section

AV Access
Simplify your life