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”):

- EVO-IP Go: EVOIPTXG & EVOIPRXG4K

Fundamental Knowledge

- **Device Name:** referred to in document as “hostname” and 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 parameters that have been defined.

Requirements of the Developers

- Familiarity with all of the features and functions of devices listed in the document
- Proficient in the knowledge of TCP/IP network communication programming including TCP, UDP, Telnet, and HTTP protocols
- Experienced in approaching Linux operating system through the console

How the Client App Interacts with Devices

The following protocols are available for the devices:

- UDP, contains UDP broadcast and UDP unicast that are 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.

Getting Started

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

- Several TX and RX units
- 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 runs 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 device 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 the above devices are powered on, each RX will pair with a certain TX by default without a client app being involved.

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:

3. 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* or “hostname” can be obtained through message parsing, and the sending device’s IP address can be obtained according to the UDP message.

Functions Based on Telnet Session

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

Functions for TX and RX

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.

/ #

Setting and Saving Device Alias

The EVO-IP Go app can set device alias and save it to the device for future reading.

Note: The device provides 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 another field name in case it is as same as the device’s existing parameter.

Supported devices: All devices

Command Example:

/ # gbparam s alias sonyDVD

/ # gbparam g alias

sonyDVD/ #

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:

- The above commands do not take effect 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*
- */ #*

Operating the Serial Port

The device serial port can be utilized in 2 ways:

- Application 1: Serial commands pass through TX and RX to control extender devices
- Application 2: Using 3rd party software, communicates with the device and send commands to peripherals connected to the device's serial port

Set Serial Parameters for pass-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: N = none; E = even; O = odd
- *s* denotes the stop digits ranging (1, 2)

Note: If commands are to pass-through between TX and RX, their serial port parameters should be set to the same.

Note: The default serial parameter of the device is 115200-8n1.

Command Example:

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

/ # soip2 -S -b 9600-8n1

- **Using a third-party program or software to send commands to a connected serial device 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 carriage return

-n: Add a line feed

-b sss-dps: Set serial parameters (sames as serial pass-through section)

Note: Each command MUST include *-b sss-dns* and *-f/dev/ttyS0* is fixed in the command

Command Example:

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

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

Supported devices: All devices

Rebooting Device

Command Structure: reboot

Supported Devices: All devices

Command Example:

/ # reboot

/ #

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

/ #

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

/ #

Functions for TX Only

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

/ #

Functions for RX Only

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

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 third-party software or app.

Command Structure:

gbparam s cec_poweron_cmd "XXXX"

gbparam s cec_standby_cmd "YYYY"

Description:

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

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"

/ #

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:

XXXX: 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"

/ #

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

/ #

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"

/ #

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

Command Example:

/ # gbconfig --source-select=341B22822FEF

/ # e e_reconnect

/ # gbconfig --source-select=NULL

/ # e e_reconnect

/ #

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

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.

Video Wall

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: EVOIPRXG 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 disable the video wall, perform the following:

/ # e e_vw_enable_0_0_0_0

/ # e e_reconnect

/ #

Instructing RX to Force a Resolution Output

Command Structure:

gbset fvo RESOLUTION

e e_reoutput

Description:

The value of *RESOLUTION*:

a. Auto

b. 3840x2160_30

c. 1080P_60

d. 720P_60

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

Supported Devices: EVOIPRXG

Command Example:

/ # gbset fvo 3840x2160_30

/ # e e_reoutput

/ #

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/ #

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: EVOIPRXG

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/ #

Obtaining and Settings 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 |

/ #

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

/ #

Switching Multiple RX's to a Source_ Simultaneously

If multiple RX's 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*

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

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:

EVOIPRXG Supports 1080P image only.

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)