API COMMAND SET

iMatrix C604

4K60 6x4 Seamless Matrix Switcher

V1.0.0 | April 16, 2025

Default Setting

Parameters	Value
Baud Rate	9600 bps
Data bits	8 bits
Parity	None
Stop bits	1 bit
Flow control	None

About Telnet Connection

Before the process of sending the telnet command, shall make telnet connection to the corresponding device.

The form of telnet command is as follow:

telnet ip (port)

ip: The unit's IP address. Suppose the IP address of the matrix is 192.168.11.143.

port: The unit's port number, this is non-required on some Telnet control tools or platforms. If required, port number is 23 by default.

Example: If the unit's IP address is 192.168.11.143, the telnet command is *telnet 192.168.11.143*

Command

Take Command `GET VER prm1<CR><LF>` as an example:

1. [`GET VER`] denotes command key words, case insensitive.
2. [`prm1`] denotes parameters, case insensitive; incorrect parameters number will not be recognized.
3. `<CR><LF>` denotes a carriage return or a line feed; all commands must be ended up with a carriage return or a line feed.

No.	Description	Command	Example
		Video Switch	
1	Switch one Input to one Output	**Command:** SET SW *in out*<CR><LF> **Return:** SW *in out*<CR><LF> **Parameter:** *in* = {in1, in2…in6, in0}; *out* = {out1, out2, out3, out4}; **Description:** in -hdmiin0 means power down output SW is short for Switch Switch one Input to one Output.	**Command:** SET SW in1 out2<CR><LF> **Return:** SW in1 out2<CR><LF> **Description:** Switch input 1 to hdmi output 2.
2	Switch one indicate input to all outputs	**Command:** SET SW *in all*<CR><LF> **Return:** SW *in all* <CR><LF> **Parameter:** *in* = { in1, in2… in6, in0}; *all* = {all}; **Description:** in - hdmiin0 means power down output SW is short for Switch Switch one indicate input to all outputs.	**Command:** SET SW in1 *all* <CR><LF> **Return:** SW in1 all<CR><LF> **Description:** Switch input1 to all output sinks.

No.	Description	Command	Example
3	Get which input is mapping to the indicate Output	**Command:** GET MP *out*<CR><LF> **Return:** Mp *in out*<CR><LF> **Parameter:** *in* = {in1,in2…in6,in0}; *out* = {out1, out2, out3, out4}; **Description:** in - hdmiin0 means power down output MP is short for mapping Get which input is mapping to the indicate Output.	**Command:** GET MP *out1*<CR><LF> **Return:** MP *hdmiin2 out1*<CR><LF> **Description:** Get which input is mapping to output 1.
4	Get correspondence between all inputs and outputs	**Command:** GET MP *all*<CR><LF> **Return:** MP *in out*<CR><LF> MP *in out*<CR><LF> ..... **Parameter:** in = {in1, in2…in6, in0}; out = {out1, out2, out3, out4}; *all* = {all}; **Description:** in - none means power down output MP is short for mapping Get correspondence between all inputs and outputs.	**Command:** GET MP *all* <CR><LF> **Return:** MP *in1 out1*<CR>... MP *in2 out2*<CR><LF> …… **Description:** Get correspondence between all inputs and outputs.

No.	Description	Command	Example
Mic Configuration			
1	Set whether the mic phantom power is set to enable	**Command:** SET MIC_PHA_PWR prm1 prm2<CR><LF> **Return:** MIC_PHA_PWR prm1 prm2<CR><LF> **Parameter:** prm1 = {1, 2}; prm2 = {on, off} **Description:** Set whether the mic phantom power is set to enable.	**Command:** SET MIC_PHA_PWR *1 ON*<CR><LF> **Return:** MIC_PHA_PWR 1 ON<CR><LF> **Description:** Set mic1's phantom power is set to enable.
2	Get whether the mic phantom power is set to enable	**Command:** GET MIC_PHA_PWR prm1<CR><LF> **Return:** MIC_PHA_PWR prm1 prm2<CR><LF> **Parameter:** prm1 = {1, 2}; prm2 = {on, off} **Description:** Get whether the mic phantom power is set to enable	**Command:** GET MIC_PHA_PWR *1*<CR><LF> **Return:** MIC_PHA_PWR 1 ON<CR><LF> **Description:** Get whether the mic1's phantom power is set to enable, the result is on.

No.	Description	Command	Example
Output Audio Mute			
1	Set output audio in video to mute / unmute	**Command:** SET MUTE out_port *prm*<CR><LF> **Return:** MUTE out_port prm<CR><LF> **Parameter:** *out_port* = {HDMIOUT1 ~ HDMIOUT4, all} *prm* = {ON, OFF} **Description:** Set output audio in video output to mute / unmute.	**Command:** SET MUTE HDMIOUT1 ON<CR><LF> **Return:** MUTE HDMIOUT1 ON<CR><LF> **Description:** Set output audio of HDMI OUT1 to mute.
2	Get mute status of output audio in video output	**Command:** GET MUTE out_port<CR><LF> **Return:** MUTE out_port prm<CR><LF> **Parameter:** *out_port* = {HDMIOUT1 ~ HDMIOUT4, all} *prm* = {ON, OFF} **Description:** Get mute status of output audio in video output.	**Command:** GET MUTE HDMIOUT1<CR><LF> **Return:** MUTE HDMIOUT1 ON<CR><LF> **Description:** Get mute status of output audio in HDMI OUT1.
GPIO & RELAY			
1	Set the mode of the GPIO IN	**Command:** SET GPIO_IN_CTL prm1 prm2 prm3<CR><LF>	**Command:** SET GPIO_IN_CTL 1 1 3<CR><LF>

No.	Description	Command	Example
		Return: GPIO_IN_CTL prm1 prm2 prm3 prm4<CR><LF> **Parameter:** prm1 = {1, 2, 3, 4}; prm2 = {}; // low-state Threshold prm3 = {}; // high-state Threshold prm4 = {high, low} **Description:** Set the mode of the GPIO IN.	**Return:** GPIO_IN_CTL 1 1 3 HIGH<CR><LF>
2	Get the mode of the GPIO IN	**Command:** GET GPIO_IN_CTL prm1<CR><LF> **Return:** GPIO_IN_CTL prm1 prm2 prm3 prm4<CR><LF> **Parameter:** prm1 = {1, 2, 3, 4}; prm2 = {1~5}; prm3 = {1~5} prm4 = {high, low} **Description:** Get the mode of the GPIO IN.	**Command:** GET GPIO_IN_CTL 1<CR><LF> **Return:** GPIO_IN_CTL 1 1 3 HIGH<CR><LF> **Description:** Get the mode of the GPIO IN.

No.	Description	Command	Example
3	Set the mode of GPIO OUT	**Command:** SET GPIO_OUT_CTL prm1 prm2 prm3<CR><LF> **Return:** GPIO_OUT_CTL prm1 prm2 prm3<CR><LF> **Parameter:** prm1 = {1, 2, 3, 4}; prm2 = {ON, OFF}; prm3 = {high, low} **Description:** Set the mode of GPIO OUT.	**Command:** SET GPIO_OUT_CTL 1 ON HIGH<CR><LF> **Return:** GPIO_OUT_CTL 1 ON HIGH<CR><LF> **Description:** Set the mode of GPIO OUT.
4	Get the mode of GPIO OUT	**Command:** GET GPIO_OUT_CTL prm1<CR><LF> **Return:** GPIO_OUT_CTL prm1 prm2 prm3<CR><LF> **Parameter:** prm1 = {1, 2, 3, 4}; prm2 = {ON, OFF}; prm3 = {high, low} **Description:** Get the mode of GPIO OUT.	**Command:** GET GPIO_OUT_CTL 1<CR><LF> **Return:** GPIO_OUT_CTL 1 ON HIGH<CR><LF> **Description:** Get the mode of GPIO OUT.

No.	Description	Command	Example
5	Set the mode of RELAY	**Command:** SET RELAY prm1 prm2<CR><LF> **Return:** RELAY prm1 prm2<CR><LF> **Parameter:** prm1 = {1, 2}; prm2 = {ON, OFF}; **Description:** Set the mode of RELAY.	**Command:** SET RELAY 1 ON<CR><LF> **Return:** RELAY 1 ON<CR><LF> **Description:** Set RELAY 1 to on.
6	Get the mode of RELAY	**Command:** GET RELAY prm1<CR><LF> **Return:** RELAY prm1 prm2<CR><LF> **Parameter:** prm1 = {1, 2}; prm2 = {ON, OFF}; **Description:** Get the mode of RELAY.	**Command:** GET RELAY 1<CR><LF> **Return:** RELAY 1 ON<CR><LF> **Description:** Get the mode of RELAY1.
CEC Control			
1	Set CEC POWER ON/OFF	**Command:** SET CEC_PWR *out* *prm*<CR><LF> **Return:** CEC_PWR *out* *prm*<CR><LF> **Parameter:** *prm* = {on, off}	**Command:** SET CEC_PWR out1 *on*<CR><LF> **Return:** CEC_PWR out1 *on*<CR><LF> **Description:** Set sink hdmi output 1 to

No.	Description	Command	Example
		out = {out1, out2…out4, all}; **Description:** Set CEC POWER ON/OFF.	power on via CEC.
2	Set CEC AUTO POWER ON/OFF	**Command:** SET AUTOCEC_FN *out prm*<CR><LF> **Return:** AUTOCEC_FN *out prm*<CR><LF> **Parameter:** *prm* = {on, off} *out* = {out1,out2…out4,all}; **Description:** Set sink auto power Function to ON or OFF.	**Command:** SET AUTOCEC_FN *out1 on*<CR><LF> **Return:** AUTOCEC_FN out1 *on*<CR><LF> **Description:** Set auto power function of out1 to on.
3	Get CEC AUTO POWER ON/OFF Status	**Command:** GET AUTOCEC_FN *out*<CR><LF> **Return:** AUTOCEC_FN *out prm*<CR><LF> **Parameter:** *prm* = {on, off} *out* = {out1,out2…out4}; **Description:** Get Sink auto power Function ON or OFF Status.	**Command:** GET AUTOCEC_FN out1<CR><LF> **Return:** AUTOCEC_FN *on* **Description:** Get Sink auto power status, and the status is ON.

No.	Description	Command	Example
4	Set CEC POWER Delay Time	**Command:** SET AUTOCEC_D *out prm*<CR><LF> **Return:** AUTOCEC_D *out prm*<CR><LF> **Parameter:** *out* = {out1,out2…out4,all}; *prm* = {1,2,3…,}// according to the actual time counter,1 means 1 minute ,2 means 2 minutes，Default wait time is 2 minutes, Max wait time is 30 minutes. 0 means when no active signal, the unit auto power off immediately. **Description:** AUTOCEC_D is short for CEC auto Power Delay Timing.	**Command:** SET AUTOCEC_D *out1 2*<CR><LF> **Return:** AUTOCEC_D out1 *2*<CR><LF> **Description:** when no active signal to hdmi1, 2 minutes later, the unit will auto power off.
5	Get CEC POWER Delay Time Status	**Command:** GET AUTOCEC_D *out* <CR><LF> **Return:** AUTOCEC_D *out prm*<CR><LF> **Parameter:** *out* = {out1,out2…out4,all}; *prm* = {1,2,3…,}// according to the actual time counter,1 means 1 minute ,2 means 2	**Command:** GET AUTOCEC_D out1 <CR><LF> **Return:** AUTOCEC_D *out1 2* <CR><LF> **Description:** Get hdmi1 auto power delay time, the result is 2 minutes.

No.	Description	Command	Example
		minutes，Default wait time is 2 minutes, Max wait time is 30 minutes. 0 means when no active signal, the unit auto power off immediately. **Description:** AUTOCEC_D is short for CEC auto Power Delay Timing.	
6	Set CEC commands of Poweron/Poweroff	**Command:** SET CECCMD_EDIT *out prm1 prm2*<CR><LF> **Return:** CECCMD_EDIT *out prm1 prm2<CR><LF>* **Parameter:** *out* = {out1,out2…out4,all}; prm1 = {pwron, pwroff}, // *prm2* = {...},// many hexadecimal numbers, the total is less than 16. Space separated **Description:** Set CEC commands of Poweron/Poweroff. If relevant commands are required, please refer to CEC specification for details.	**Command:** SET CECCMD_EDIT out1 pwron 40 04<CR><LF> **Return:** CECCMD_EDIT out1 pwron 40 04<CR><LF> **Description:** Set CEC commands of Poweron.

No.	Description	Command	Example
7	Get CEC command of Poweron/Poweroff	**Command:** GET CECCMD_EDIT *out prm1*<CR><LF> **Return:** CECCMD_EDIT *out prm1 prm2<CR><LF>* **Parameter:** *out* = {out1,out2…out4,all}; prm1 = {pwron, pwroff}, // *prm2* = {...},// many hexadecimal numbers, the total is less than 16. Space separated **Description:** Get CEC command of Poweron/Poweroff.	**Command:** GET CECCMD_EDIT out1 pwron<CR><LF> **Return:** CECCMD_EDIT out1 pwron 40 04<CR><LF>
8	Send CEC command	**Command:** SET CEC_CMD *out prm*<CR><LF> **Return:** CEC_CMD *out prm*<CR><LF> **Parameter:** *out* = {out1,out2…out4,all}; *prm* = {...},// many hexadecimal numbers, the total is less than 16. Space separated **Description:** Send CEC command. If relevant commands are required, please refer to CEC specification for details.	**Command:** SET CEC_CMD out1 40 04<CR><LF> **Return:** CEC_CMD out1 40 04<CR><LF> **Description:** Send CEC command.

No.	Description	Command	Example
		HDCP	
1	Set Input HDCP support to ON/OFF	**Command:** SET HDCP_S in prm<CR><LF> **Return:** HDCP_S *in prm*<CR><LF> **Parameter:** *prm* = {on, off} *in* = {in2…in6} **Description:** HDCP_S will control source hdcp support on or off	**Command:** SET HDCP_S in2 *on*<CR><LF> **Return:** HDCP_S in2 *on*<CR><LF> **Description:** Set hdmi input 2 hdcp support to on.
2	Get Input HDCP support ON/OFF Status	**Command:** GET HDCP_S *in* <CR><LF> **Return:** HDCP_S *in prm*<CR><LF> **Parameter:** *prm* = {on, off} *in* = {in2…in6} **Description:** HDCP_S is short for HDCP support	**Command:** GET HDCP_S in2<CR><LF> **Return:** HDCP_S *in2 on*<CR><LF> **Description:** Get hdmi2 hdcp support on or off status, and the result is on.

No.	Description	Command	Example
3	Set Output HDCP	**Command:** SET HDCP *out prm*<CR><LF> **Return:** HDCP *out prm*<CR><LF> **Parameter:** *out* = {out1,out2…out4,all};// all includes hdmiout1-hdmiout7 *prm* = { auto，hdcp1.x}// auto means source use hdcp 1.4, then output also use hdcp 1.4. the same with hdcp 2.2 and off. **Description:** Set Output HDCP.	**Command:** SET HDCP *out1 auto*<CR><LF> **Return:** HDCP *out1 auto*<CR><LF> **Description:** set output hdcp to auto mode.
4	Get Output HDCP	**Command:** GET HDCP *out*<CR><LF> **Return:** HDCP *out prm*<CR><LF> **Parameter:** *out* = {out1,out2…out4,all};// all includes hdmiout1-hdmiout7 *prm* = {auto，hdcp1.x}// auto means source use hdcp 1.4 ,then output also use hdcp 1.4. the same with hdcp 2.2 and off. **Description:** Get Output HDCP.	**Command:** GET HDCP *out1*<CR><LF> **Return:** HDCP *out1 follow*<CR><LF> **Description:** Get output hdcp, the result is follow mode

No.	Description	Command	Example
		EDID	
1	Set Input EDID	**Command:** SET EDID *in prm*<CR><LF> **Return:** EDID *in prm*<CR><LF> **Parameter:** *in* = {in1,in2...in6}; prm = {1 ~10} 1: Copy form hdmi output 1 2: Copy form hdmi output 2 3: Copy form hdmi output 3 4: Copy form hdmi output 4 5: Fixed 4K60 2.0CH PCM Audio with HDR; 6: Fixed 4K60 2.0CH PCM Audio with SDR; 7: Fixed 4K30 2.0CH PCM Audio with HDR; 8: Fixed 4K30 2.0CH PCM Audio with SDR; 9: Fixed 1080p@60Hz 2.0CH PCM Audio with HDR; 10: Fixed 1080p@60Hz 2.0CH PCM Audio with SDR; **Description:** Set Input EDID.	**Command:** SET EDID *in1 6*<CR><LF> **Return:** EDID *in1 6*<CR><LF> **Description:** Set in1 EDID to Fixed 4K60 2.0CH PCM Audio with SDR.

No.	Description	Command	Example
2	Get All Input EDID status	**Command:** GET EDID *all* <CR><LF> **Return:** EDID *in prm*<CR> EDID *in prm*<CR> EDID *in prm*<CR><LF> ..... **Parameter:** *in* = {in1, in2...in6}; prm = {1 ~10} 1: Copy form hdmi output 1 2: Copy form hdmi output 2 3: Copy form hdmi output 3 4: Copy form hdmi output 4 5: Fixed 4K60 2.0CH PCM Audio with HDR; 6: Fixed 4K60 2.0CH PCM Audio with SDR; 7: Fixed 4K30 2.0CH PCM Audio with HDR; 8: Fixed 4K30 2.0CH PCM Audio with SDR; 9: Fixed 1080p@60Hz 2.0CH PCM Audio with HDR; 10: Fixed 1080p@60Hz 2.0CH PCM Audio with SDR; **Description:** Get all input EDID Status.	**Command:** GET EDID *all* <CR><LF> **Return:** EDID in1 *01*<CR> EDID in2 *02*<CR> EDID in3 *03*<CR><LF> …… **Description:** Get all input EDID Status.

No.	Description	Command	Example
3	Get one input EDID Status	**Command:** GET EDID *in* <CR><LF> **Return:** EDID *in prm*<CR><LF> **Parameter:** *in* = {in1,in2....in6}; prm = {1 ~10} 1: Copy form hdmi output 1 2: Copy form hdmi output 2 3: Copy form hdmi output 3 4: Copy form hdmi output 4 5: Fixed 4K60 2.0CH PCM Audio with HDR; 6: Fixed 4K60 2.0CH PCM Audio with SDR; 7: Fixed 4K30 2.0CH PCM Audio with HDR; 8: Fixed 4K30 2.0CH PCM Audio with SDR; 9: Fixed 1080p@60Hz 2.0CH PCM Audio with HDR; 10: Fixed 1080p@60Hz 2.0CH PCM Audio with SDR; **Description:** Get one input EDID Status.	**Command:** GET EDID in1<CR><LF> **Return:** EDID in1 *6*<CR><LF> **Description:** Get in1 edid status, and the status is Fixed 4K60 2.0CH PCM Audio with SDR.

No.	Description	Command	Example
4	Write the EDID to one input	**Command:** SET EDID_W *in prm1 prm2*<CR><LF> **Return:** EDID_W *in prm1 prm3*<CR><LF> **Parameter:** *in* = {in1,in2...in6}; *prm1* = {block0, block1}; *prm2* = one block of 256 bytes edid ascii data with spaces (hex data need conversion into ASCII code) *prm3* = {ok, error}; error: check sum error **Description:** Write EDID content to input.	**Command:** SET EDID_W in1 *block0 XX…XX*<CR><LF> **Return:** EDID_W in1 *block0 ok*<CR><LF> **Description:** Write EDID content to input1.

No.	Description	Command	Example
5	Read Output EDID	**Command:** GET EDID_R *out*<CR><LF> **Return:** EDID_R *out prm1 prm2*<CR><LF> **Parameter:** *out* = {out1,out2...out4}; *prm1* = {block0, block1}; *prm2* ={ one block of 256 bytes edid ascii data with no spaces(hex data need conversion into ASCII code), error, unconnect}; **Description:** Read EDID content of output.	**Command:** GET EDID_R out1<CR><LF> **Return:** EDID_R out1 *block0 XX…XX*<CR><LF> EDID_R out1 *block1 XX…XX*<CR><LF> **Description:** EDID_R out1 *block0 XX…XX*<CR><LF> --- Read EDID ok or EDID_R *out1 error*<CR><LF> --- Check Sum Error or EDID_R out1 *unconnect*<CR><LF> --- Sink unconnect
System Info			
1	Factory reset	**Command:** RESET<CR><LF> **Return:** RESET<CR><LF> **Description:** Factory reset.	**Command:** RESET<CR><LF> **Return:** RESET<CR><LF> **Description:** Factory reset all boards.
2	System reboot	**Command:** REBOOT<CR><LF> **Return:** REBOOT<CR><LF> **Description:** System reboot.	**Command:** REBOOT<CR><LF> **Return:** REBOOT<CR><LF> **Description:** System reboot.

No.	Description	Command	Example
3	Get the API list	**Command:** help<CR><LF> **Return:** xxxx **Description:** Get the API list.	**Command:** help<CR><LF> **Return:** xxxx **Description:** Get the API list.
4	GET IP address	**Command:** GET IPADDR<CR><LF> **Return:** IPADDR xx.xx.xx.xx<CR><LF> **Description:** GET IP address.	**Command:** GET IPADDR<CR><LF> **Return:** IPADDR 192.168.11.243<CR><LF>
Update Info			
1	Get selected target firmware version	**Command:** GET VER<CR><LF> **Return:** VER *prm*<CR><LF> **Parameter:** *prm* = {…}// according to actual firmware version **Description:** Get selected target firmware version.	**Command:** GET VER<CR><LF> **Return:** ARM_Vx.y.z<CR><LF> **Description:** Get all module firmware version.
Preset Scene			
1	Manage Preset Scenes	**Command:** SET PRESET {ACTION} PRESETIDX [NAME] <CR><LF> **Return:** PRESET {ACTION} PRESETIDX [NAME]	**1** **Command:** SET PRESET save 1<CR><LF> **Return:** PRESET save 1<CR><LF> **Description:**

No.	Description	Command	Example
		<CR><LF> **Parameter:** PRESETIDX = {1, 2, 3, ...}; // the numbers of preset scenes ACTION = {save|restore|clear|name}; // operation types: save: Save current configurations as a preset scene with a specified number restore: restore the configurations saved in a preset scene with a specified number clear: Clear the preset scene with a specified number name: name a preset scene with a specified NAME = {name}; // When ACTION is name，NAME is the name of preset scene. for name: The label can use numbers, letters, space and "_", "-"special character. Label length should be less than 32 characters. **Description:** This command allows for saving the current configuration as a preset, restoring a preset, clearing a preset, or naming a preset, based on the specified preset index (PRESETIDX). If ACTION is	Save the current settings to preset scene 1. **2** **Command:** SET PRESET restore 2<CR><LF> **Return:** PRESET restore 2<CR><LF> **Description:** Restore the preset settings saved in preset scene 2. **3** **Command:** SET PRESET clear 3<CR><LF> **Return:** PRESET clear 3<CR><LF> **Description:** Clear the preset settings saved in preset scene 3. **4** **Command:** SET PRESET name 4 MeetingRoom<CR><LF> **Return:** PRESET name 4 MeetingRoom<CR><LF> **Description:** Set the name of preset scene 4 to "MeetingRoom".

No.	Description	Command	Example
		`name`, an additional parameter NAME is required to specify the preset scene's name.	
2	Get Preset Status	**Command:** GET PRESET {prm1} {index_or_all} <CR><LF> **Return:** PRESET {prm1} PRESETINDEX STATUS PRESETNAME <CR><LF> **Parameter:** {prm1} = the operation of request, only “Info” supported currently index_or_all = {1, 2, 3, …, all}; // the numbers of preset scenes, 'all' indicates get information of all preset scenes PRESETINDEX = the numbers of preset scenes STATUS = {0, 1, 2}; // 0 indicates empty preset scene (Empty)，1 indicates the preset is defined but inactive (Inactive)， 2 indicates the preset is defined and active (Active) PRESETNAME = the name of preset scenes **Description:** This command retrieves information about a specified preset scene or all preset scenes. The response includes the preset index, status (0 for	**1** **Command:** GET PRESET info 1<CR><LF> **Return:** PRESET info 1 2 MeetingRoom<CR><LF> **Description:** Get the information of preset scene 1. The status is 2, meaning this preset is defined and currently active, and the preset name is MeetingRoom. **2** **Command:** GET PRESET info all<CR><LF> **Return:** PRESET info 1 2 MeetingRoom<CR><LF> PRESET info 2 0 default_name<CR><LF> PRESET info 3 1 Conference<CR><LF> ** default_name：preset x **Description:** Get the information of all preset scenes. Scene 1 is named MeetingRoom and is active (status = 2), scene 2

No.	Description	Command	Example
		empty, 1 for defined but inactive, 2 for defined and active), and the preset name in the format `PRESET prm1 PRESETINDEX STATUS PRESETNAME`.	is empty (status = 0), and scene 3 is defined but inactive (status = 1).
USB Info			
1	Set which USB host the USB devices are connected to	**Command:** SET USB_HOST prm1<CR><LF> **Return:** USB_HOST prm1<CR><LF> **Parameter:** prm1 = {typec, usbhost1, usbhost2} **Description:** Set which USB host the USB devices are connected to.	**Command:** SET USB_HOST usbhost1<CR><LF> **Return:** USB_HOST usbhost1<CR><LF> **Description:** USB devices are connected to USB host1.
2	Get which USB host the USB devices are connected to	**Command:** GET USB_HOST<CR><LF> **Return:** USB_HOST prm1<CR><LF> **Parameter:** prm1 = {typec, usbhost1, usbhost2} **Description:** Get which USB host the USB devices are connected to.	**Command:** GET USB_HOST<CR><LF> **Return:** USB_HOST usbhost1<CR><LF> **Description:** Get which USB host the USB devices are connected to, the result is usbhost1.

No.	Description	Command	Example
		Output Resolution	
1	Set output resolution in manual mode	**Command:** SET VIDOUT_RES *out prm*<CR><LF> **Return:** VIDOUT_RES *out prm*<CR><LF> **Description:** **This function can only be set in SCALE manual mode** *out = {out1, out2, out3, out4};* *prm =* { AUTO, 3840x2160@60, 3840x2160@50, 3840x2160@30, 3840x2160@25, 3840x2160@24, 1920x1200@60, 1920x1080@60, 1920x1080@50, 1680x1050@60, 1600x1200@60, 1600x900@60, 1440x900@60, 1366x768@60, 1360x768@60, 1280x1024@60, 1280x960@60, 1280x800@60, 1280x768@60, 1280x720@60, 1280x720@50, 1024x768@60, 800x600@60 , }	**Command:** SET VIDOUT_RES *out1 3840x2160@60*<CR><LF> **Return:** VIDOUT_RES *out1 3840x2160@60*<CR><LF> **Description:** Set out1 resolution to 3840x2160@60.

No.	Description	Command	Example
2	Get output resolution	**Command:** GET VIDOUT_RES *out*<CR><LF> **Return:** VIDOUT_RES *out prm*<CR><LF> **Description:** **This function can only be set in SCALE manual mode** *out = {out1, out2, out3, out4};* *prm=* { AUTO, ... 1280x800@60, 1280x768@60, 1024x768@60, 800x600@60 , }	**Command:** GET VIDOUT_RES *out1*<CR><LF> **Return:** VIDOUT_RES *out1 1280x800@60*<CR><LF> **Description:** HDMI out 1's resolution is 1280x800@60.
HTTPS & TelnetS			
1	Set HTTPS service status	**Command:** SET HTTPS prm<CR><LF> **Return:** HTTPS prm <CR><LF> **Parameter:** *prm*= {on, off}; **Description:** Set HTTPS service status.	**Command:** SET HTTPSS on<CR><LF> **Return:** HTTPS on<CR><LF> **Description:** Set HTTPS service to on.

No.	Description	Command	Example
2	Get HTTPS service status	**Command:** GET HTTPS<CR><LF> **Return:** HTTPS<CR><LF> **Parameter:** *prm*= {on, off}; **Description:** Get HTTP service status.	**Command:** GET HTTPS<CR><LF> **Return:** HTTPS on<CR><LF> **Description:** Get HTTP service status, the result is on.
3	Set TelnetS service status	**Command:** SET TELNETS prm<CR><LF> **Return:** TELNETS prm <CR><LF> **Parameter:** *prm*= {on, off}; **Description:** Set TelnetS service status.	**Command:** SET TELNETS on<CR><LF> **Return:** TELNETS on<CR><LF> **Description:** Set TelnetS service to on.
4	Get TELNETS service status	**Command:** GET TELNETS<CR><LF> **Return:** TELNETS prm <CR><LF> **Parameter:** *prm*= {on, off}; **Description:** Get TELNETS service status.	**Command:** GET TELNETS<CR><LF> **Return:** TELNETS on<CR><LF> **Description:** Get TELNETS service status, the result is on.

No.	Description	Command	Example
5	Set Telnet New password	**Command:** SET TEL_PWD Oldpwd Newpwd NewpwdVerify<CR><LF> **Return:** TEL_PWD prm<CR><LF> **Parameter:** Old/Newpwd : Must be 4 to 16 characters in lentgh (alphanumeric only) prm : {0, 1}; // 0：failed, 1：success **Description:** Set the password for logging in to the telnet.	**Command:** SET TEL_PWD admin admin123 admin123<CR><LF> **Return:** TEL_PWD 1<CR><LF> **Description:** Set Telnet's password [admin] to [admin123]
DSP Control			
1	Set DSP matrix crosspoint status	**Command:** SET DSP_MIX out_port in_port value<CR><LF> **Return:** DSP_MIX out_port in_port value<CR><LF> **Parameter:** out_port = {LINEOUT1/2, DANTE1/2/3/4, USBOUT1}; in_port = {MIC1, MIC2, LINEIN1, DANTE1 ~ DANTE4, USBIN1, OUT1 ~ OUT4}; value = {ON, OFF} **Description:** Set DSP matrix crosspoint status.	**Command:** SET DSP_MIX LINEOUT1 MIC1 OFF<CR><LF> **Return:** DSP_MIX LINEOUT1 MIC1 OFF<CR><LF> **Description:** Set DSP matrix's LINEOUT1 crosspoint MIC1 status to off.

No.	Description	Command	Example
2	Get DSP matrix crosspoint status	**Command:** SET DSP_MIX out_port in_port<CR><LF> **Return:** DSP_MIX out_port in_port value<CR><LF> **Parameter:** out_port = {LINEOUT1/2, DANTE1/2/3/4, USBOUT1}; in_port = {MIC1, MIC2, LINEIN1, DANTE1 ~ DANTE4, USBIN1, OUT1 ~ OUT4}; value = {ON, OFF} **Description:** Get DSP matrix crosspoint status.	**Command:** GET DSP_MIX LINEOUT1 MIC1<CR><LF> **Return:** DSP_MIX LINEOUT1 MIC1 OFF<CR><LF> **Description:** Get DSP matrix's LINEOUT1 crosspoint MIC1 status, the result is off.
3	Set DSP expander status	**Command:** SET DSP_EXP port value<CR><LF> **Return:** DSP_EXP port value<CR><LF> **Parameter:** port = {MIC1, MIC2, LINEIN1} value = {ON, OFF}	**Command:** SET DSP_EXP MIC1 ON<CR><LF> **Return:** DSP_EXP MIC1 ON<CR><LF> **Description:** Set dsp expander status to on.
4	Get DSP expander status	**Command:** GET DSP_EXP port<CR><LF> **Return:** DSP_EXP port	**Command:** GET DSP_EXP MIC1<CR><LF> **Return:** DSP_EXP MIC1

No.	Description	Command	Example
		value<CR><LF> **Parameter:** port = {MIC1, MIC2, LINEIN1} value = {ON, OFF}	ON<CR><LF> **Description:** Get dsp expander status, the result is on.
5	Set DSP expander's properties	**Command:** SET DSP_EXP_PROP field_id port value<CR><LF> **Return:** DSP_EXP_PROP field_id port value<CR><LF> **Parameter:** field_id = {Attack, Release, Threshold, Ratio} port = {MIC1, MIC2, LINEIN1} value = {{**Attack** = {1 ~ 500 ms}, {**Release** = {1 ~ 2000 ms}, {**Threshold** = {-100 ~ 0 dB}, {**Ratio** = {1 ~ 100}}	**Command:** SET DSP_EXP_PROP attack mic1 50<CR><LF> **Return:** DSP_EXP_PROP attack mic1 50<CR><LF> **Description:** Set attack time on mic input1 to 50.

No.	Description	Command	Example
6	Get DSP expander's properties	**Command:** GET DSP_EXP_PROP field_id port<CR><LF> **Return:** DSP_EXP_PROP field_id port value<CR><LF> **Parameter:** field_id = {Attack, Release, Threshold, Ratio} port = {MIC1, MIC2, LINEIN1} value = {{**Attack** = {1 ~ 500 ms}, {**Release** = {1 ~ 2000 ms}, {**Threshold** = {-100 ~ 0 dB}, {**Ratio** = {1 ~ 100}}	**Command:** GET DSP_EXP_PROP attack mic1<CR><LF> **Return:** DSP_EXP_PROP attack mic1 50<CR><LF> **Description:** Get attack time on mic input1, the result is 50.
7	Set DSP input HPF's status	**Command:** SET DSP_IN_HPF port value<CR><LF> **Return:** DSP_IN_HPF port value<CR><LF> **Parameter:** port = {MIC1, MIC2, LINEIN1} value = {ON, OFF}	**Command:** SET DSP_IN_HPF MIC1 OFF<CR><LF> **Return:** DSP_IN_HPF MIC1 OFF<CR><LF> **Description:** Set Dsp Hpf status on mic input 1 to off.

No.	Description	Command	Example
8	Get DSP input HPF's status	**Command:** GET DSP_IN_HPF port<CR><LF> **Return:** DSP_IN_HPF port value<CR><LF> **Parameter:** port = {MIC1, MIC2, LINEIN1} value = {ON, OFF}	**Command:** GET DSP_IN_HPF MIC1<CR><LF> **Return:** DSP_IN_HPF MIC1 OFF<CR><LF> **Description:** Get Dsp Hpf status on mic input 1, the result is off.
9	Set DSP input HPF's property	**Command:** SET DSP_IN_HPF_PROP port value<CR><LF> **Return:** DSP_IN_HPF_PROP port value<CR><LF> **Parameter:** port_type = {MIC1, MIC2, LINEIN1} value = {20 ~ 20000 Hz}	**Command:** SET DSP_IN_HPF_PROP MIC1 20<CR><LF> **Return:** DSP_IN_HPF_PROP MIC1 20<CR><LF> **Description:** Set Hpf freq on mic input 1 to 20Hz.
10	Get DSP input HPF's property	**Command:** GET DSP_IN_HPF_PROP port<CR><LF> **Return:** DSP_IN_HPF_PROP port value<CR><LF> **Parameter:** port_type = {MIC1, MIC2, LINEIN1} value = {20 ~ 20000 Hz}	**Command:** GET DSP_IN_HPF_PROP MIC1<CR><LF> **Return:** DSP_IN_HPF_PROP MIC1 20<CR><LF> **Description:** Get Hpf freq on mic input 1, the result is 20Hz.

No.	Description	Command	Example
11	Set DSP output HPF's status	**Command:** SET DSP_OUT_HPF port value<CR><LF> **Return:** DSP_OUT_HPF port value<CR><LF> **Parameter:** port = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {ON, OFF}	**Command:** SET DSP_OUT_HPF LINEOUT1 OFF<CR><LF> **Return:** DSP_OUT_HPF LINEOUT1 OFF<CR><LF> **Description:** Set Dsp Hpf status on LINEOUT1 to off.
12	Get DSP output HPF's status	**Command:** GET DSP_OUT_HPF port<CR><LF> **Return:** DSP_OUT_HPF port value<CR><LF> **Parameter:** port = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {ON, OFF}	**Command:** GET DSP_OUT_HPF LINEOUT1<CR><LF> **Return:** DSP_OUT_HPF LINEOUT1 OFF<CR><LF> **Description:** Get Dsp Hpf status on LINEOUT1, the result is off.
13	Set DSP output HPF's property	**Command:** SET DSP_OUT_HPF_PROP port value<CR><LF> **Return:** DSP_OUT_HPF_PROP port value<CR><LF> **Parameter:** port_type = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {20 ~ 20000 Hz}	**Command:** SET DSP_OUT_HPF_PROP LINEOUT1 20<CR><LF> **Return:** DSP_OUT_HPF_PROP LINEOUT1 20<CR><LF> **Description:** Set Hpf freq on LINEOUT1 to 20Hz.

No.	Description	Command	Example
14	Get DSP output HPF's property	**Command:** GET DSP_OUT_HPF_PROP port<CR><LF> **Return:** DSP_OUT_HPF_PROP port value<CR><LF> **Parameter:** port_type = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {20 ~ 20000 Hz}	**Command:** GET DSP_OUT_HPF_PROP LINEOUT1<CR><LF> **Return:** DSP_OUT_HPF_PROP LINEOUT11 20<CR><LF> **Description:** Get Hpf freq on LINEOUT1, the result is 20Hz.
15	Set DSP compressor status	**Command:** SET DSP_COMP port value<CR><LF> **Return:** DSP_COMP port value<CR><LF> **Parameter:** port = {MIC1, MIC2, LINEIN1} value = {ON, OFF}	**Command:** SET DSP_COMP MIC1 OFF<CR><LF> **Return:** DSP_COMP MIC1 OFF<CR><LF> **Description:** Set compressor status on mic input 1 to off.
16	Get DSP compressor status	**Command:** GET DSP_COMP port<CR><LF> **Return:** DSP_COMP port value<CR><LF> **Parameter:** port = {MIC1, MIC2, LINEIN1} value = {ON, OFF}	**Command:** GET DSP_COMP MIC1<CR><LF> **Return:** DSP_COMP MIC1 OFF<CR><LF> **Description:** Get compressor status on mic input 1 as off.

No.	Description	Command	Example
17	Set DSP compressor's property	**Command:** SET DSP_COMP_PROP field_id port value<CR><LF> **Return:** DSP_COMP_PROP field_id port value<CR><LF> **Parameter:** field_id = {Attack, Release, Threshold, Ratio} port = {MIC1, MIC2, LINEIN1} value = {{**Attack** = {0 ~ 500 ms}, {**Release** = {0 ~ 2000 ms}, {**Threshold** = {-100 ~ 0 dB}, {**Ratio** = {1 ~ 100}}	**Command:** SET DSP_COMP attack MIC1 15<CR><LF> **Return:** DSP_COMP attack MIC1 15<CR><LF> **Description:** Set compressor attack time on mic input 1 to 15ms.
18	Get DSP compressor's property	**Command:** GET DSP_COMP_PROP field_id port<CR><LF> **Return:** DSP_COMP_PROP field_id port value<CR><LF> **Parameter:** field_id = {Attack, Release, Threshold, Ratio} port = {MIC1, MIC2, LINEIN1} value = {{**Attack** = {0 ~ 100 ms}, {**Release** = {0 ~ 10000 ms}, {**Threshold** = {-100 ~ 0 dB}, {**Ratio** = {1 ~ 100}}	**Command:** GET DSP_COMP attack MIC1<CR><LF> **Return:** DSP_COMP attack MIC1 15<CR><LF> **Description:** Get compressor attack time on mic input 1, the result is 15ms.

No.	Description	Command	Example
19	Set DSP equalizer's status	**Command:** SET DSP_EQ port value<CR><LF> **Return:** DSP_EQ port value<CR><LF> **Parameter:** port = { LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {ON, OFF}	**Command:** SET DSP_EQ LINEOUT1 OFF<CR><LF> **Return:** DSP_EQ LINEOUT1 OFF<CR><LF> **Description:** Set EQ status on LINEOUT1 to off.
20	Get DSP equalizer's status	**Command:** GET DSP_EQ port<CR><LF> **Return:** DSP_EQ port value<CR><LF> **Parameter:** port = { LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {ON, OFF}	**Command:** GET DSP_EQ LINEOUT1<CR><LF> **Return:** DSP_EQ LINEOUT1 OFF<CR><LF> **Description:** Get EQ status on LINEOUT1, the result is off.

No.	Description	Command	Example
21	Set DSP equalizer's property	**Command:** SET DSP_EQ_PROP field_id port band_index value<CR><LF> **Return:** DSP_EQ_PROP field_id port band_index value<CR><LF> **Parameter:** field_id = {level, freq, qfactor} port = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} band_index = {1 ~ 8} value = {{**level** = {-10 ~ 10 dB}, {**freq** = {20 ~ 20000 Hz}, {**qfactor** = {0 ~ 16}}	**Command:** SET DSP_EQ_PROP level LINEOUT1 1 0<CR><LF> **Return:** DSP_EQ_PROP level LINEOUT1 1 0<CR><LF> **Description:** Set EQ band1's level on LINEOUT1 to 0 dB.
22	Get DSP equalizer's property	**Command:** GET DSP_EQ_PROP field_id band_index port<CR><LF> **Return:** DSP_EQ_PROP field_id port band_index value<CR><LF> **Parameter:** field_id = {level, freq, qfactor} port = { LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} band_index = {1 ~ 8} value = {{**level** = {-10 ~ 10 dB}, {**freq** = {20 ~ 20000 Hz}, {**qfactor** = {0 ~ 16}}	**Command:** GET DSP_EQ_PROP level LINEOUT1 1<CR><LF> **Return:** DSP_EQ_PROP level LINEOUT1 1 0<CR><LF> **Description:** Get EQ band1's level on LINEOUT1, the result is 0 dB.

No.	Description	Command	Example
23	Set DSP LPF's status	**Command:** SET DSP_LPF port value<CR><LF> **Return:** DSP_LPF port value<CR><LF> **Parameter:** port = { LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {ON, OFF}	**Command:** SET DSP_LPF LINEOUT1 OFF<CR><LF> **Return:** DSP_LPF LINEOUT1 OFF<CR><LF> **Description:** Set DSP lpf status on LINEOUT1 to off.
24	Get DSP LPF's status	**Command:** GET DSP_LPF port<CR><LF> **Return:** DSP_LPF port value<CR><LF> **Parameter:** port = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {ON, OFF}	**Command:** gET DSP_LPF LINEOUT1<CR><LF> **Return:** DSP_LPF LINEOUT1 OFF<CR><LF> **Description:** Get DSP lpf status on LINEOUT1, the result is off.
25	Set DSP LPF's property	**Command:** SET DSP_LPF_PROP port value<CR><LF> **Return:** DSP_LPF_PROP port value<CR><LF> **Parameter:** port = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {20 ~ 20000 Hz}	**Command:** SET DSP_LPF_PROP LINEOUT1 20<CR><LF> **Return:** DSP_LPF_PROP LINEOUT1 20<CR><LF> **Description:** Set dsp lpf freq on LINEOUT1 to 20Hz.

No.	Description	Command	Example
26	Get DSP LPF's property	**Command:** GET DSP_LPF_PROP port<CR><LF> **Return:** DSP_LPF_PROP port value<CR><LF> **Parameter:** port = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {20 ~ 20000 Hz}	**Command:** GET DSP_LPF_PROP LINEOUT1<CR><LF> **Return:** DSP_LPF_PROP LINEOUT1 20<CR><LF> **Description:** Get dsp lpf freq on LINEOUT1, the result is 20Hz.
27	Set DSP duck's status	**Command:** SET DSP_DUCK port value<CR><LF> **Return:** DSP_DUCK port value<CR><LF> **Parameter:** port = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {ON, OFF}	**Command:** SET DSP_DUCK LINEOUT1 OFF<CR><LF> **Return:** DSP_DUCK LINEOUT1 OFF<CR><LF> **Description:** Set duck function on for LINEOUT1 to off.
28	Get DSP duck's status	**Command:** GET DSP_DUCK port<CR><LF> **Return:** DSP_DUCK port value<CR><LF> **Parameter:** port = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {ON, OFF}	**Command:** GET DSP_DUCK LINEOUT1<CR><LF> **Return:** DSP_DUCK LINEOUT1 OFF<CR><LF> **Description:** Get duck function of LINEOUT1, the result is off.

No.	Description	Command	Example
29	Set DSP duck's property	**Command:** SET DSP_DUCK_PROP field_id port value<CR><LF> **Return:** DSP_DUCK_PROP field_id port value<CR><LF> **Parameter:** field_id = {Attack, Release, Threshold, Ratio} port_index = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {{**Attack** = {1 ~ 500 ms}, {**Release** = {1 ~ 2000 ms}, {**Threshold** = {-100 ~ 0 dB}, {**Ratio** = {1 ~ 100}}}	**Command:** SET DSP_DUCK_PROP attack LINEOUT1 80<CR><LF> **Return:** DSP_DUCK_PROP attack LINEOUT1 80<CR><LF> **Description:** Set duck function's attack time on LINEOUT1 to 80ms.
30	Get DSP duck's property	**Command:** GET DSP_DUCK_PROP field_id port<CR><LF> **Return:** DSP_DUCK_PROP field_id port value<CR><LF> **Parameter:** field_id = {Attack, Release, Threshold, Ratio} port_index = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {{**Attack** = {1 ~ 500 ms}, {**Release** = {1 ~ 2000 ms}, {**Threshold** = {-100 ~ 0 dB}, {**Ratio** = {1 ~ 100}}}	**Command:** GET DSP_DUCK_PROP attack LINEOUT1<CR><LF> **Return:** DSP_DUCK_PROP attack LINEOUT1 80<CR><LF> **Description:** Get duck function's attack time on LINEOUT1, the result is 80ms.

No.	Description	Command	Example
31	Set DSP duck's master	**Command:** SET DSP_DUCK_PRI port value<CR><LF> **Return:** DSP_DUCK_PRI port value<CR><LF> **Parameter:** port_index = { LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {MIC1, MIC2, LINEIN1, DANTE1 ~ DANTE4, USBIN1, OUT1 ~ OUT4}	**Command:** SET DSP_DUCK_PRI LINEOUT1 MIC1<CR><LF> **Return:** DSP_DUCK_PRI LINEOUT1 MIC1<CR><LF> **Description:** Set duck MASTER on LINEOUT1 to MIC1.
32	Get DSP duck's master	**Command:** GET DSP_DUCK_PRI port<CR><LF> **Return:** DSP_DUCK_PRI port value<CR><LF> **Parameter:** port_index = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {MIC1, MIC2, LINEIN1, DANTE1 ~ DANTE4, USBIN1, OUT1 ~ OUT4}	**Command:** GET DSP_DUCK_PRI LINEOUT1<CR><LF> **Return:** DSP_DUCK_PRI LINEOUT1 MIC1<CR><LF> **Description:** Get duck MASTER on LINEOUT1, the result is MIC1.

No.	Description	Command	Example
33	Set input audio mute status.	**Command:** SET DSP_AUD_GAIN_MUTE port value<CR><LF> **Return:** DSP_AUD_GAIN_MUT port value<CR><LF> **Parameter:** port = {MIC1 ~ MIC2, LINEIN, DANTE1 ~ DANTE4, USBIN1, IN1 ~ IN6} value = {ON, OFF}	**Command:** SET DSP_AUD_GAIN_MUT MIC1 OFF<CR><LF> **Return:** DSP_AUD_GAIN_MUT MIC1 OFF<CR><LF> **Description:** Set MIC input1 audio gain mute status to off.
34	Get input audio mute status	**Command:** GET DSP_AUD_GAIN_MUTE port<CR><LF> **Return:** DSP_AUD_GAIN_MUT port value<CR><LF> **Parameter:** port = {MIC1 ~ MIC2, LINEIN, DANTE1 ~ DANTE4, USBIN1, IN1 ~ IN6} value = {ON, OFF}	**Command:** GET DSP_AUD_GAIN_MUT MIC1<CR><LF> **Return:** DSP_AUD_GAIN_MUT MIC1 OFF<CR><LF> **Description:** Get MIC input1 audio gain mute status, the result is off.

No.	Description	Command	Example
35	Set input audio gain	**Command:** SET DSP_AUD_GAIN port value<CR><LF> **Return:** DSP_AUD_GAIN port value<CR><LF> **Parameter:** port = {MIC1 ~ MIC2, LINEIN, DANTE1 ~ DANTE4, USBIN1, IN1 ~ IN6} value = {{MICx = {0 ~ 80 dB}, {Others = {-20 ~ 30 dB}}	**Command:** SET DSP_AUD_GAIN MIC1 60<CR><LF> **Return:** DSP_AUD_GAIN MIC1 60<CR><LF> **Description:** Set MIC input1 audio gain to 60 dB.
36	Get input audio gain	**Command:** GET DSP_AUD_GAIN port<CR><LF> **Return:** DSP_AUD_GAIN port value<CR><LF> **Parameter:** port = {MIC1 ~ MIC2, LINEIN, DANTE1 ~ DANTE4, USBIN1, IN1 ~ IN6} value = {{MICx = {0 ~ 80 dB; 0: mute}, {Others = {-20 ~ 30 dB}}	**Command:** GET DSP_AUD_GAIN MIC1<CR><LF> **Return:** DSP_AUD_GAIN MIC1 60<CR><LF> **Description:** Get MIC input1 audio gain, the result is 60 dB.

No.	Description	Command	Example
37	Set output audio mute status	**Command:** SET DSP_AUD_VOL_MUTE port value<CR><LF> **Return:** DSP_AUD_VOL_MUTE port value<CR><LF> **Parameter:** port = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {OFF, ON};	**Command:** SET DSP_AUD_VOL_MUTE LINEOUT1 OFF<CR><LF> **Return:** DSP_AUD_VOL_MUTE LINEOUT1 OFF<CR><LF> **Description:** Set LINEOUT1 audio mute status to off.
38	Get output audio mute status.	**Command:** GET DSP_AUD_VOL_MUTE port<CR><LF> **Return:** DSP_AUD_VOL_MUTE port value<CR><LF> **Parameter:** port = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {OFF, ON};	**Command:** SET DSP_AUD_VOL_MUTE LINEOUT1<CR><LF> **Return:** DSP_AUD_VOL_MUTE LINEOUT1 OFF<CR><LF> **Description:** Get LINEOUT1 audio mute status, the result is off.

No.	Description	Command	Example
39	Set output audio volume	**Command:** SET DSP_AUD_VOL port value<CR><LF> **Return:** DSP_AUD_VOL port value<CR><LF> **Parameter:** port = { LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {-100 ~ 0 dB};	**Command:** SET DSP_AUD_VOL LINEOUT1 -60<CR><LF> **Return:** DSP_AUD_VOL LINEOUT1 -60<CR><LF> **Description:** Set LINEOUT1 audio volume to -60 dB.
40	Get output audio volume	**Command:** GET DSP_AUD_VOL port<CR><LF> **Return:** DSP_AUD_VOL port value<CR><LF> **Parameter:** port = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {-100 ~ 0 dB};	**Command:** SET DSP_AUD_VOL LINEOUT1<CR><LF> **Return:** DSP_AUD_VOL LINEOUT1<CR><LF> **Description:** Get LINEOUT1 audio volume, the result is -60 dB.

No.	Description	Command	Example
41	Read DSP input's meter	**Command:** GET DSP_IN_METER port<CR><LF> **Return:** DSP_IN_METER port value<CR><LF> **Parameter:** port = {MIC1 ~ MIC2, LINEIN1, DANTE1 ~ DANTE4, USBIN1, IN1 ~ IN6} value = {{**MICx** = {0 ~ 80 dB}, {**Others** = {-20 ~ 30 dB}};	**Command:** GET DSP_IN_METER SPK1<CR><LF> **Return:** DSP_IN_METER SPK1 -60<CR><LF> **Description:** Get MIC1 meter, the result is 0 dB.
42	Read DSP output's meter	**Command:** GET DSP_OUT_METER port<CR><LF> **Return:** DSP_OUT_METER port value<CR><LF> **Parameter:** port = {LINEOUT1/2, DANTE1 ~ DANTE4, USBOUT1} value = {-100 ~ 0 dB};	**Command:** GET DSP_OUT_METER LINEOUT1<CR><LF> **Return:** DSP_OUT_METER LINEOUT1 -60<CR><LF> **Description:** Get LINEOUT1 meter, the result is -60 dB.

No.	Description	Command	Example
43	Set the connection lock status of the DSP output port	**Command:** SET DSP_LINK_LOCK port prm<CR><LF> **Return:** DSP_LINK_LOCK port prm<CR><LF> **Parameter:** port = {2 ~ 3}; 2: link DANTE1 & DANTE2; 3: link DANTE3 & DANTE4 prm = {on, off};	**Command:** SET DSP_LINK_LOCK 2 ON<CR><LF> **Return:** DSP_LINK_LOCK 2 ON<CR><LF> **Description:** Link DANTE1 and DANTE2.
44	Get the connection lock status of the DSP output port	**Command:** GET DSP_LINK_LOCK port<CR><LF> **Return:** DSP_LINK_LOCK port prm<CR><LF> **Parameter:** port = {2 ~ 3}; 2: link DANTE1 & DANTE2; 3: link DANTE3 & DANTE4 prm = {on, off};	**Command:** GET DSP_LINK_LOCK 2<CR><LF> **Return:** DSP_LINK_LOCK 2 ON<CR><LF> **Description:** Get DANTE1 and DANTE2 link status, the result is link lock.
45	Set input audio for AEC MASTER selection	**Command:** SET DSP_AEC_MASTER port<CR><LF> **Return:** DSP_AEC_MASTER port<CR><LF> **Parameter:** port = {NONE, LINEOUT1/2, DANTE_OUT1/2/3/4,	**Command:** SET DSP_AEC_MASTER MIC1<CR><LF> **Return:** DSP_AEC_MASTER MIC1<CR><LF> **Description:** Set AEC Master input audio to MIC1.

No.	Description	Command	Example
		USBOUT1, MIC1/2, LINEIN1, DANTE_IN1/4}	
46	Get input audio for AEC MASTER selection	**Command:** GET DSP_AEC_MASTER<CR><LF> **Return:** DSP_AEC_MASTER port<CR><LF> **Parameter:** port = {NONE,LINEOUT1/2, DANTE_OUT1/2/3/4, USBOUT1, MIC1/2, LINEIN1, DANTE_IN1/4}	**Command:** GET DSP_AEC_MASTER<CR><LF> **Return:** DSP_AEC_MASTER MIC1<CR><LF> **Description:** Get AEC Master input audio, the result is MIC1.
47	Set reference input audio for AEC selection	**Command:** SET DSP_AEC_REF port<CR><LF> **Return:** DSP_AEC_REF port<CR><LF> **Parameter:** port = {NONE,LINEOUT1/2, DANTE_OUT1/2/3/4, USBOUT1, MIC1, MIC2, LINEIN1, DANTE_IN1/4, USBIN1, IN1 ~ IN6}	**Command:** SET DSP_AEC_REF MIC1<CR><LF> **Return:** DSP_AEC_REF MIC1<CR><LF> **Description:** Set AEC reference input audio to MIC1.

No.	Description	Command	Example
48	Get reference input audio for AEC selection	**Command:** GET DSP_AEC_REF<CR><LF> **Return:** DSP_AEC_REF port<CR><LF> **Parameter:** port = {NONE,LINEOUT1/2, DANTE_OUT1/2/3/4, USBOUT1, MIC1, MIC2, LINEIN1, DANTE_IN1/4, USBIN1, IN1 ~ IN6}	**Command:** GET DSP_AEC_REF<CR><LF> **Return:** DSP_AEC_REF MIC1<CR><LF> **Description:** Get AEC reference input audio, the result is MIC1.