API Command Set_iSwitch 402UC V1.9

		SYSTEM		Reboot Value	Reset Value
1	Obtain the list of all API commands	**Syntax** **Command:** help[CR/LF] **Return:** xxxx **Parameter:** NA **Description:** Get the API list	**Example** **Command:** help[CR/LF] **Return:** xxxx **Description:** Get the API list		
2	Get firmware version	**Syntax** **Command:** GET VER prm1[CR/LF] **Return:** VER *prm2* [CR/LF] **Parameter:** ※ *prm1 = {MAIN， ARM, USB_C_VIDEO, HDMI, USB_C_CC, CPLD,HDBT_3.0, ALL)* ※ *prm2 = {...}// according to actual firmware version* **Description:** Get firmware version LT86404 - HDMI KTM5000 - USB_C_VIDEO TPS65988 - USB_C_CC VS3000 - HDBT_3.0	**Example** **Command:** GET VER Main[CR/LF] **Return:** Main V1.1.3[CR/LF] **Description:** Get main version **Command:** GET VER ALL [CR/LF] **Return:** Main V1.1.3[CR/LF] ARM V1.1.3[CR/LF] USB_C_CC V1.0.1[CR/LF] USB_C_VIDEO V1.0.3[CR/LF] HDMI V1.1.0[CR/LF] HDBT_3 V5.4.87[CR/LF] CPLD V1.0.0[CR/LF] **Description:** Get ALL version		
3	Factory reset	**Syntax** **Command:** RESET[CR/LF] **Return:** RESET[CR/LF] **Parameter:** **Description:** Factory reset	**Example** **Command:** RESET[CR/LF] **Return:** RESET[CR/LF] **Description:** Factory reset		
4	System reboot	**Syntax** **Command:** REBOOT[CR/LF] **Return:** REBOOT[CR/LF] **Parameter:** **Description:** system reboot	**Example** **Command:** REBOOT[CR/LF] **Return:** REBOOT[CR/LF] **Description:** System reboot		
5	Set new username and password for Telnet Login	**Syntax** **Command:** SET TEL_USNM Oldusnm Newusnm TEL_PWD Oldpwd Newpwd[CR/LF] **Return:** TEL_USNM prm TEL_PWD prm[CR/LF] **Parameter:** Old/Newusnm : The length is [4, 16] Old/Newpwd : Must be 4 to 16 characters in lentgh (alphanumeric only) prm : {0, 1); // 0: failed, 1: success **Description:** Set the username and password for login to the telent	**Example** **Command:** SET TEL_USNM root root123 TEL_PWD admin admin123[CR/LF] **Return:** TEL_USNM 1 TEL_PWD 1[CR/LF] **Description:** Set the Telnet's username [root] to [root123] and Telnet's password [admin] to [admin123]		
		SWITCHING		**Reboot Value**	**Reset Value**
6	Set input to output video switching	**Syntax** **Command:** SET SW *out in* [CR/LF] **Return:** SW out in[CR/LF] **Parameter:** ※ *in* = {IN1, IN2, IN3, IN4}; ※ *out* = {OUT1, OUT2, ALL}; **Description:** ※ SW is short to Swtich ※ Switch one input source to one output sink	**Example** **Command:** SET SW OUT1 IN2[CR/LF] **Return:** SW OUT1 IN2[CR/LF] **Description:** Swtich INPUT 2 for HDMI ouput 1 **Command:** SET SW ALL IN2[CR/LF] **Return:** SW OUT1 IN2[CR/LF] SW OUT2 IN2[CR/LF] USBSW IN2[CR/LF]		
7	Get which Input switched to the according Output	**Syntax** **Command:** GET SW *out* [CR/LF] **Return:** SW *out in* [CR/LF] **Parameter:** ※ in = {IN1, IN2, IN3, IN4}; ※ out = {OUT1, OUT2}; **Description:** ※Get which Input mapping to the indicate Output	**Example** **Command:** GET SW *OUT2* [CR/LF] **Return:** SW OUT2 IN1[CR/LF] **Description:** Get which input mapping to output 2		
8	Set the status of the auto-switching function status	**Syntax** **Command:** SET AUTOSW_FN *prm1* [CR/LF] **Return:** AUTOSW_FN *prm* [CR/LF] **Parameter:** ※ *prm* = {on, off) **Description:** Set whether to open the video and USB auto-switching mode	**Example** **Command:** SET AUTOSW_FN *on* [CR/LF] **Return:** AUTOSW_FN *on* [CR/LF] **Description:** Set to open video and USB auto switch mode		
9	Get the status (ON or OFF) of the auto-switching function	**Syntax** **Command:** GET AUTOSW_FN [CR/LF] **Return:** AUTOSW_FN *prm* [CR/LF] ※ *prm* = {on, off) **Description:** Get the video and USB auto-switching mode status	**Example** **Command:** GET AUTOSW_FN [CR/LF] **Return:** AUTOSW_FN *on* [CR/LF] **Description:** Video and USB auto-switching mode is open		

10	Set the viedo auto-switching mode	**Syntax** **Command:** SET AUTOSW_MD *prm1* [CR/LF] **Return:** AUTOSW_MD *prm1* [CR/LF] **Parameter:** ※*prm1 = {LIFO, PRIORITY}* **Description:** Set the video auto-switching mode	**Example** **Command:** SET AUTOSW_MD LIFO[CR/LF] **Return:** AUTOSW_MD LIFO[CR/LF] **Description:** Set the video auto-switching mode LIFO		
11	Get the viedo auto-switching mode	**Syntax** **Command:** GET AUTOSW_MD [CR/LF] **Return:** AUTOSW_MD *prm1* [CR/LF] **Parameter:** ※*prm1 = {LIFO, PRIORITY}* **Description:** Get the video auto-switching mode	**Example** **Command:** GET AUTOSW_MD[CR/LF] **Return:** AUTOSW_MD LIFO[CR/LF] **Description:** Get the video auto-switching mode LIFO		
		CEC		**Reboot Value**	**Reset Value**
12	Set CEC: Power ON or Power OFF	**Syntax** **Command:** SET CEC_PWR *out prm* [CR/LF] **Return:** CEC_PWR *out prm* [CR/LF] **Parameter:** ※ *prm* = {on,off} ※ *out* = {OUT1, OUT2,OUT3,OUT4ALL}; **Description:** Set sink power on or off OUT1:hdmi out1 OUT2:hdmi out2 OUT3:hdbt out1 OUT4:hdbt out2	**Example** **Command:** SET CEC_PWR OUT1 *on* [CR/LF] **Return:** CEC_PWR OUT1 *on* [CR/LF] **Description:** Set sink hdmi output 1 power on		
13	Set CEC Auto Power ON or Power OFF	**Syntax** **Command:** SET AUTOCEC_FN *out prm* [CR/LF] **Return:** AUTOCEC_FN *out prm* [CR/LF] **Parameter:** ※ *prm* = {on, off} ※ *out* = {OUT1, OUT2,OUT3,OUT4,ALL}; **Description:** Set sink auto power Function ON or OFF OUT1:hdmi out1 OUT2:hdmi out2 OUT3:hdbt out1 OUT4:hdbt out2 **Default:** on	**Example** **Command:** SET AUTOCEC_FN *OUT1 on* [CR/LF] **Return:** AUTOCEC_FN OUT1 *on* [CR/LF] **Description:** Set sink hdmi output 1 auto power ON		
14	Get the status of CEC AUTO POWER ON/OFF	**Syntax** **Command:** GET AUTOCEC_FN *out* [CR/LF] **Return:** AUTOCEC_FN *out prm* [CR/LF] **Parameter:** ※ *prm* = {on, off} ※ out = {OUT1, OUT2,OUT3,OUT4,ALL}; **Description:** Get Sink auto power Function ON or OFF Status. **Default:** on	**Example** **Command:** GET AUTOCEC_FN OUT1[CR/LF] **Return:** AUTOCEC_FN OUT*1 on* [CR/LF] **Description:** Get Sink auto power status, and the status is ON.		
15	Set CEC POWER Delay Time	**Syntax** **Command:** SET AUTOCEC_D *out prm* [CR/LF] **Return:** AUTOCEC_D *out prm* [CR/LF] **Parameter:** ※ out = {OUT1, OUT2,OUT3,OUT4,ALL}; ※ *prm* = {30,40,60,...}//According to the actual time counter, in seconds. Maximum 1800s **Description:** AUTOCEC_D is short for CEC auto Power Delay Timing	**Example** **Command:** SET AUTOCEC_D *OUT1 30* [CR/LF] **Return:** AUTOCEC_D OUT1 *30* [CR/LF] **Description:** When there is no activation signal for hdmi1, the unit will automatically turn off the power after 30 seconds.		
16	Get the status of CEC POWER Delay Time	**Syntax** **Command:** GET AUTOCEC_D *out* [CR/LF] **Return:** AUTOCEC_D *out prm* [CR/LF] **Parameter:** ※ out = {OUT1, OUT2,OUT3,OUT4,ALL}; ※ prm = {30,40,60,...}//According to the actual time counter, in seconds. Maximum 1800s **Description:** AUTOCEC_D is short for CEC auto Power Delay Timing	**Example** **Command:** GET AUTOCEC_D OUT1[CR/LF] **Return:** AUTOCEC_D OUT*1 30* [CR/LF] **Description:** Obtain the automatic power delay time of hdmi1, with a result of 30 seconds.		
17	Set CEC Volume increase	**Syntax** **Command:** SET CEC_VOL_INC *out* [CR/LF] **Return:** CEC_VOL_INC *out* [CR/LF] **Parameter:** ※ out = {OUT1, OUT2,OUT3,OUT4,ALL}; **Description:** Control device volume increase via CEC	**Example** **Command:** SET CEC_VOL_INC *OUT1* [CR/LF] **Return:** CEC_VOL_INC OUT1 [CR/LF] **Description:**		
18	Set CEC Volume decrease	**Syntax** **Command:** SET CEC_VOL_DEC *out* [CR/LF] **Return:** CEC_VOL_DEC *out* [CR/LF] **Parameter:** ※ out = {OUT1, OUT2,OUT3,OUT4,ALL}; **Description:** Control device volume decrease via CEC	**Example** **Command:** SET CEC_VOL_DEC *OUT1* [CR/LF] **Return:** CEC_VOL_DEC OUT1 [CR/LF] **Description:**		

19	Set CEC Volume mute or Umute	**Syntax** **Command:** SET CEC_VOL_MUTE *out* [CR/LF] **Return:** CEC_VOL_MUTE *out prm* [CR/LF] **Parameter:** ※ out = {OUT1, OUT2,OUT3,OUT4,ALL}; ※ *prm* = {OK} **Default:** off	**Example** **Command:** SET CEC_VOL_MUTE *OUT1* [CR/LF] **Return:** CEC_VOL_MUTE OUT1 OK[CR/LF] **Description:** Set CEC control sink volume mute		
20	Set CEC commands for defined Power on/Power off	**Syntax** **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 command for defined Poweron/Poweroff. If relevant commands are required, please refer to CEC specification for details.	**Example** **Command:** SET CECCMD_EDIT out1 pwron 40 04[CR/LF] **Return:** CECCMD_EDIT out1 pwron 40 04[CR/LF] **Description:** Send CEC command		
21	Get CEC command for defined Power on/Power off	**Syntax** **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:** **G**et CEC command for defined Power ON/ Power OFF.	**Example** **Command:** GET CECCMD_EDIT out1 pwron[CR/LF] **Return:** CECCMD_EDIT out1 pwron 40 04[CR/LF] **Description:**		
22	Send CEC command	**Syntax** **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.	**Example** **Command:** SET CEC_CMD out1 40 04[CR/LF] **Return:** CEC_CMD out1 40 04[CR/LF] **Description:** Send CEC command		
		RS232		**Reboot Value**	**Reset Value**
23	Set RS232 baud rate	**Syntax** **Command:** SET UART_B *prm1 prm2* [CR/LF] **Return:** UART_B out *prm1 prm2* [CR/LF] **Parameter:** ※ *prm1 = {UART1,UART2,UART3}* ※ *prm2* = {115200, 57600, 38400, 19200, 9600~~,4800,2400~~} **Description:** 115200 UART1:RS232-1 UART2:RS232-2 UART3:RS232-HDBT	**Example** **Command:** SET UART_B *UART1 57600* [CR/LF] **Return:** UART_B UART1 57600[CR/LF] **Description:** Set the baud rate of RS232-1 to 57600		
24	Get RS232 baud rate	**Syntax** **Command:** GET UART_B *prm1* [CR/LF] **Return:** UART_B *prm1 prm2* [CR/LF] **Parameter:** ※ *prm1 = {UART1,UART2,UART3}* ※ *prm2* = {115200, 57600, 38400, 19200, 9600~~,4800,2400~~} **Description:** 115200 UART1:RS232-1 UART2:RS232-2 UART3:RS232-HDBT	**Example** **Command:** GET UART_B UART1[CR/LF] **Return:** UART_B UART1 57600[CR/LF] **Description:** The baud rate of RS232-1 to 57600		
25	Set RS232 auto control power ON/OFF the projector screen	**Syntax** **Command:** SET UARTPWR_FN prm *prm1* [CR/LF] **Return:** UARTPWR_FN *prm1 prm2* [CR/LF] **Parameter:** ※ *prm = {UART1,UART2,UART3}* ※ *prm1* = {on,off} **Description:** Set UART auto pwr function ON or OFF. UART1:RS232-1 UART2:RS232-2 UART3:RS232-HDBT	**Example** **Command:** SET UARTPWR_FN UART1 *on* [CR/LF] **Return:** UARTPWR_FN UART1 *on* [CR/LF] **Description:** Set uart1 auto power function on		
26	Get the status of the RS232's auto control power ON/OFF	**Syntax** **Command:** GET UARTPWR_FN prm[CR/LF] **Return:** UARTPWR_FN *prm prm1* [CR/LF] **Parameter:** ※ *prm = {UART1,UART2,UART3}* ※ *prm1 = {on,off}* **Description:** Get uart auto power on status UART1:RS232-1 UART2:RS232-2 UART3:RS232-HDBT	**Example** **Command:** GET UARTPWR_FN UART1[CR/LF] **Return:** UARTPWR_FN UART1 *on* [CR/LF] **Description:** Get uart1 auto pwr status, and status is ON.		

27	Set RS232's auto control power delay time	**Syntax** **Command:** SET UARTPWR_D prm *prm1* [CR/LF] **Return:** UARTPWR_D prm *prm1* [CR/LF] **Parameter:** *prm = {UART1}* *prm1* = {30,...,1800}// According to the actual time counter, the range is 30-1800s. **Description:** UARTPWR_D is short for UART Power Delay Timing	**Example** **Command:** SET UARTPWR_D UART1 *30* [CR/LF] **Return:** UARTPWR_D UART1 30[CR/LF] **Description:** When no active signal to uart1, 30 sec later, the unit will auto power OFF.		
28	Get the delay time of the RS232's auto control power	**Syntax** **Command:** GET UARTPWR_D UART1[CR/LF] **Return:** UARTPWR_D *prm1 prm2* [CR/LF] **Parameter:** *prm1 = {UART1}* *prm2* = {30,...,1800}// According to the actual time counter, the range is 30-1800s. **Description:** UARTPWR_D is short for UART Power Delay Timing	**Example** **Command:** GET UARTPWR_D UART1[CR/LF] **Return:** UARTPWR_D *UART1 30* [CR/LF] **Description:** Get uart1 auto power delay time, the result is 30 sec.		
29	Set RS232's command	**Syntax** **Command:** SET UART_CMD prm *prm1 prm2 prm3* [CR/LF] **Return:** UART_CMD prm *prm1 prm2 prm3* [CR/LF] **Parameter:** *prm = {UART1,UART2,UART3}* *prm1* = {poweron, poweroff, volumemute, volumeunmute, volumeup, volumedown...} *prm2* = {hex, str} *prm3* = {xxxxxxxxxxx} (max size is 48 byte) **Description:** control display via RS232 command UART1:RS232-1 UART2:RS232-2 UART3:RS232-HDBT	**Example** **Command:** SET UART_CMD UART1 poweroff hex 30 31 32 33[CR/LF] **Return:** UART_CMD UART1 poweroff hex 30 31 32 33[CR/LF] **Description:** SET UART1 CMD		
30	Get RS232's command	**Syntax** **Command:** GET UART_CMD UART1 [CR/LF] **Return:** UART_CMD prm *prm1 prm2 prm3 prm4* [CR] UART_CMD prm *prm1 prm2 prm3 prm4* [CR] ... UART_CMD prm *prmn* [CR/LF] **Parameter:** *prm = {UART1,UART2,UART3}* *prm1* = {poweron,poweroff,volumemute,volumeunmute,volumeinc,volumedec...} *prm2* = {hex, str} *prm3* = {xxxxxxxxxxx}//hex is ascii character **Description:** Get all RS232 command UART1:RS232-1 UART2:RS232-2 UART3:RS232-HDBT	**Example** **Command:** GET UART_CMD UART1[CR/LF] **Return:** UART_CMD UART1 *poweron str 123* [CR/LF] UART_CMD UART1 *poweroff hex 31 32 33* [CR/LF] UART_CMD UART1 volumemute hex 31 32 33[CR/LF] UART_CMD UART1 volumeinc hex 31 32 33[CR/LF] UART_CMD UART1 volumedec hex 31 32 33[CR/LF] **Description:** Get UART 1 CMD		
31	Set RS232's command to send	**Syntax** **Command:** SET UART_CMD_S *prm prm1* [CR/LF] **Return:** UART_CMD_S *prm prm1* [CR/LF] **Parameter:** *prm = {UART1,UART2,UART3}* *prm1* = {poweron, poweroff, volumemute, volumeunmute , volumeinc, volumedec...} **Description:** send via RS232 command	**Example** **Command:** SET UART_CMD_S UART1 *poweron* [CR/LF] **Return:** *UART_CMD_S UART1 poweron[CR/LF]* **Description:** SEND UART CMD power ON		
32	Set the UART DATA to send	**Syntax** **Command:** SET UART_S *prm1 prm2 prm3* [CR/LF] **Return:** UARTCMD_S *prm1 prm2 prm3* [CR/LF] **Parameter:** *prm1* = {uart1, uart2, uart3 ...} *prm2* = {hex, str} *prm3* = {xxxxxxxxxxx} (max size is 36 byte) **Description:** send via RS232 data UART1:RS232-1 UART2:RS232-2 UART3:RS232-HDBT	**Example** **Command:** SET UART_S *uart1 str 123* [CR/LF] **Return:** UART_S *uart1 str 123[CR/LF]* **Description:** SEND data string 123 for uart1.		

33	Route Uart data to designate uart	**Syntax**
		Command:
		Route *prm1 prm2 prm3 prm4 prm5 XX XX XX XX....* [CR/LF]
		
		Return:
		Route prm1 prm2 prm3 prm4 prm5 XX XX XX XX.... [CR/LF]
		
		Parameter:
		prm1 = {uar1, uart2, uart3...}
		prm2 = {9600,19200,38400,57600,115200}// baudrate
		prm3 = {none, odd, even} //Parity
		prm4 = {7,8}//Date Bit
		prm5 = {1,2} //Stop Bit
		XX XX XX .... is String format for Hex data.
		Description:
		Route Uart data to designate uart
			Example
			Command:
			Route uart1 9600 none 8 1 00 11 22 33 44 55[CR/LF]
			
			Return:
			Route uart1 9600 none 8 1 00 11 22 33 44 5[CR/LF]
			
			Description:
			Uart1 Send Hex Data 00 11 22 33 44 55 with uart config 9600 baudrate, none parity, 8 data bit, 1 stop bit.		

		NETWORK		**Reboot Value**	**Reset Value**
34	Set IP Mode	**Syntax** **Command:** SET IP MODE prm[CR/LF] **Return:** IP MODE prm [CR/LF] **Parameter:** ※pcm= {static ,dhcp} **Description:** ※Set IP mode **Default:** DHCP	**Example** **Command:** SET IP MODE dhcp[CR/LF] **Return:** IP MODE dhcp [[CR/LF] **Description:** Set IP mode is DHCP		

35	Get IP Mode	**Syntax** **Command:** GET IP MODE[CR/LF] **Return:** IP MODE prm [CR/LF] **Parameter:** ※pcm= {static ,dhcp} **Description:** ※Get IP mode **Default:** DHCP	**Example** **Command:** GET IP MODE[CR/LF] **Return:** IP MODE dhcp [[CR/LF] **Description:** IP mode is DHCP		
36	Set IP address	**Syntax** **Command:** SET IPADDR xx.xx.xx.xx xx.xx.xx.xx xx.xx.xx.xx[CR/LF] **Return:** IPADDR IP:xx.xx.xx.xx MASK: xx.xx.xx.xx GATE: xx.xx.xx.xx[CR/LF] **Parameter:** NA **Description:** SET IP address	**Example** **Command:** SET IPADDR 192.168.1.4 255.255.255.0 192.168.1.1[CR/LF] **Return:** IPADDR IP:192.168.1.4 MASK:255.255.255.0 GATE:192.168.1.1[[CR/LF] **Description:** Set IP address is 192.168.1.4, MASK is 255.255.255.0, GATE is 192.168.1.1		
37	Get IP address	**Syntax** **Command:** GET IPADDR[CR/LF] **Return:** IPADDR IP:xx.xx.xx.xx MASK: xx.xx.xx.xx GATE: xx.xx.xx.xx[CR/LF] **Parameter:** NA **Description:** GET IP address	**Example** **Command:** GET IPADDR[CR/LF] **Return:** IPADDR IP:192.168.1.4 MASK:255.255.255.0 GATE:192.168.1.1[CR/LF] **Description:** Get IP address is 192.168.1.4, MASK is 255.255.255.0, GATE is 192.168.1.1		
38	Set the VLAN status	**Syntax** **Command:** SET VLAN_ENABLE prm[CR/LF] **Return:** VLAN_ENABLE prm[CR/LF] **Parameter:** ※prm= {on, off} **Description:** Set vlan enable status	**Example** **Command:** GET VLAN_ENABLE on[CR/LF] **Return:** VLAN_ENABLE on[CR/LF] **Description:** Set vlan enable		
39	Get the status of VLAN	**Syntax** **Command:** GET VLAN_ENABLE[CR/LF] **Return:** VLAN_ENABLE prm[CR/LF] **Parameter:** ※pcm= {on, off} **Description:** Get vlan enable status	**Example** **Command:** GET VLAN_ENABLE[CR/LF] **Return:** VLAN_ENABLE on[CR/LF] **Description:** Get vlan status is on		
		VIDEO INFORMATION		**Reboot Value**	**Reset Value**
40	Get the status of the video input	**Syntax** **Command:** GET VIDIN_SIG *in* [CR/LF] **Return:** VIDIN_SIG *in prm* [CR/LF] **Parameter:** ※ *in* = {IN1, IN2, IN3, IN4, all}; ※ *prm* = {no, valid} ※ IN1 = HDMI IN1. ※ IN2 = HDMI IN2. ※ IN3 = USB-C3(video). ※ IN4 = USB-C4(video). ※ all = all video port. **Description:** Get the status of the video input	**Example** **Command:** GET VIDIN_SIG IN1[CR/LF] **Return:** VIDIN_SIG IN1 valid[CR/LF] **Description:** Get the status of the video input1		
41	Set HDCP version of input video status	**Syntax** **Command:** SET VIDIN_HDCP_FN in prm[CR/LF] **Return:** VIDIN_HDCP_FN in prm[CR/LF] **Parameter:** ※ in = {IN1, IN2, IN3, IN4, all}; ※ prm = {on, off} ※ IN1 = HDMI IN1. ※ IN2 = HDMI IN2. ※ IN3 = USB-C3(video). ※ IN4 = USB-C4(video). ※ all = all video port. **Description:** Set HDCP version of input video	**Example** **Command:** SET VIDIN_HDCP_FN IN1 ON[CR/LF] **Return:** VIDIN_HDCP_FN IN1 no hdcp[CR/LF] **Description:** Set the HDCP status of HDMI input1		
42	Get the ON/OFF status of the HDCP version of input video	**Syntax** **Command:** GET VIDIN_HDCP_FN in[CR/LF] **Return:** VIDIN_HDCP_FN *in prm* [CR/LF] **Parameter:** ※ in = {IN1, IN2, IN3, IN4, all}; ※ prm = {on, off} ※ IN1 = HDMI IN1. ※ IN2 = HDMI IN2. ※ IN3 = USB-C3(video). ※ IN4 = USB-C4(video). ※ all = all video port. **Description:** Get HDCP version of input video	**Example** **Command:** GET VIDIN_HDCP_FN *IN1* [CR/LF] **Return:** VIDIN_HDCP_FN *IN1 ON* [CR/LF] **Description:** Get the HDCP status of HDMI input1 enable HDCP 1.4/ HDCP2.2		
43	Get HDCP version of input video	**Syntax** **Command:** GET VIDIN_HDCP *in* [CR/LF] **Return:** VIDIN_HDCP *in prm* [CR/LF] **Parameter:** ※ in = {IN1, IN2, IN3, IN4, all}; ※ prm = {no hdcp, hdcp1.4, hdcp2.2,follow} ※ IN1 = HDMI IN1. ※ IN2 = HDMI IN2. ※ IN3 = USB-C3(video). ※ IN4 = USB-C4(video). ※ all = all video port. **Description:** Get HDCP version of input video	**Example** **Command:** GET VIDIN_HDCP *IN1* [CR/LF] **Return:** VIDIN_HDCP *IN1 no hdcp* [CR/LF] **Description:** Get the HDCP status of HDMI input1		

		Syntax	Example		
44	Get SCALER OUT status	**Command:** GET SCALER *out* [CR/LF] **Return:** SCALER *out prm* [CR/LF] **Parameter:** ※ out = {OUT1, OUT2,OUT4,all}; ※ prm = {on,off, force} ※ all = all video port. **Description:** Get scaler out status	**Command:** GET SCALER out1[CR/LF] **Return:** SCALER *out1 off* [CR/LF] **Description:** Get the status of scaler output 1		
45	Set SCALER OUT status	**Syntax** **Command:** SET SCALER *out parm* [CR/LF] **Return:** SCALER *out prm* [CR/LF] **Parameter:** ※ out = {OUT1, OUT2,OUT4,all}; ※ prm = {on,off, force} ※ all = all video port. **Description:** Set scaler out status	**Example** **Command:** SET SCALER out1 off[CR/LF] **Return:** SCALER *out1 off* [CR/LF] **Description:** Set the status of scaler output 1		
		HDCP INFORMATION		**Reboot Value**	**Reset Value**
46	Get Output HDCP support status	**Syntax** **Command:** GET HDCP *out* [CR/LF] **Return:** HDCP *out prm* [CR/LF] **Parameter:** ※ *out* = {OUT1, OUT2, ALL}; ※ *prm* = { follow，hdcp1.4，off...}// off means without HDCP，follow means sink is hdcp 1.4 ,then output also use hdcp 1.4. the same with hdcp 2.2 and off. **Description:** Get Output HDCP **Default:** follow	**Example** **Command:** GET HDCP *OUT1* [CR/LF] **Return:** HDCP *OUT1 follow* [CR/LF] **Description:** Get output HDCP is follow mode		
		WEB LOGIN		**Reboot Value**	**Reset Value**
47	Set Web New username and password	**Syntax** **Command:** SET WEB_USNM Oldusnm Newusnm WEB_PWD Oldpassword Newpassword[CR/LF] **Return:** WEB_USNM prm1 WEB_PWD prm2[CR/LF] **Parameter:** Username：The longest length is [4, 16] Password : Must be 4 to 16 characters in lentgh (alphanumeric only) prm1 = {0, 1} prm2 = {0, 1, 2} 0 means Oldusnm or Oldpassword error 1 means Modify username success 2 means Cannot change to default password	**Example** **Command:** SET WEB_USNM admin admin123 WEB_PWD admin admin123[CR/LF] **Return:** WEB_USNM 1 WEB_PWD 1[CR/LF] **Description:** Modify username success from root to root123		
		USB MATRIX		**Reboot Value**	**Reset Value**
48	Set USB switching mode	**Syntax** **Command:** SET USB_M *prm* [CR/LF] **Return:** USB_M *prm* [CR/LF] **Parameter:** *prm* = {*FOLLOW1,FOLLOW2,INDEPENDENT* } **Description:** Set USB Work Mode	**Example** **Command:** SET USB_M *INDEPENDENT* [CR/LF] **Return:** USB_M *INDEPENDENT* [CR/LF] **Description:** Set USB work mode is INDEPENDENT		
49	Get USB switching mode	**Syntax** **Command:** GET USB_M[CR/LF] **Return:** USB_M *prm* [CR/LF] **Parameter:** *prm* = {FOLLOW1,FOLLOW2,INDEPENDENT} **Description:** Get USB Switching Mode	**Example** **Command:** GET USB_M[CR/LF] **Return:** USB_M *FOLLOW1* [CR/LF] **Description:** Get usb work mode, and the work mode is follow OUT1		
50	USB Switch Input for Output	**Syntax** **Command:** SET USBSW *in* [CR/LF] **Return:** USBSW *in* [CR/LF] **Parameter:** *in* = {IN1,IN2,IN3,IN4}; **Description:** ※ i00 is blank ※USBSW is short for usb switch ※Set USB Switch form one input source to all output sink	**Example** **Command:** SET USBSW *IN1* [CR/LF] **Return:** USBSW *IN1* [CR/LF] **Description:** Set USB Switch from USB input1 to USB output1		
51	Get which usb input switched to the according Output	**Syntax** **Command:** GET USBSW[CR/LF] **Return:** USBSW *in* [CR/LF] **Parameter:** *in = {IN1,IN2,IN3,IN4};* **Description:** ※USBSW is short for usb switch ※Get which input mapping to the indicate Output	**Example** **Command:** GET USBSW [CR/LF] **Return:** USBSW *i02* [CR/LF] **Description:** Get which input mapping to USB out1		
52	Set USB to network (NIC) status	**Syntax** **Command:** SET USBNIC *in parm* [CR/LF] **Return:** USBNIC *in parm* [CR/LF] **Parameter:** *in* = {IN1,IN2,IN3,IN4,ALL}; parm = {on,off} **Description:** Set USB to network status	**Example** **Command:** SET USBNIC *IN1 on* [CR/LF] **Return:** USBNIC *IN1 on* [CR/LF] **Description:** Set USB in1 to network on		
53	Get USB to network (NIC) status	**Syntax** **Command:** GET USBNIC in[CR/LF] **Return:** USBNIC *in parm* [CR/LF] **Parameter:** *in = {IN1,IN2,IN3,IN4,ALL};* parm = {on,off} **Description:** Get USB to network status	**Example** **Command:** GET USBNIC IN1 [CR/LF] **Return:** USBNIC *IN1 on* [CR/LF] **Description:** Get USB in1 to network		

		MIC IN & LINE OUT		Reboot Value	Reset Value
54	Set AUDIO Mute	**Syntax** **Command:** **S**ET AUD_MUTE prm1 prm2[CR/LF] **Return:** **AUD_MUTE** prm1 PRM2[CR/LF] **Parameter:** ※pcm1= {HDMIOUT1, HDMIOUT2,HDBTOUT2,DANTEOUT,ANALOGOUT,USBHOSTAUDIO,ALL} ※pcm1= {ON, OFF,} **Description:** Set the output AUDIO MUTE	**Example** **Command:** SET AUD_MUTE HDMIOUT1 ON[CR/LF] **Return:** AUD_MUTE HDMIOUT1 ON[CR/LF] **Description:** Set the OUT1 AUDIO MUTE		
55	Get mute status of output	**Syntax** **Command:** GET AUD_MUTE prm1[CR/LF] **Return:** **AUD_MUTE** prm1 prm2[CR/LF] **Parameter:** ※pcm1= {HDMIOUT1, HDMIOUT2,HDBTOUT2,DANTEOUT,ANALOGOUT,USBHOSTAUDIO,ALL} ※pcm1= {ON, OFF,} **Description:** Get mute status of output	**Example** **Command:** GET AUD_MUTE HDMIOUT1[CR/LF] **Return:** AUD_MUTE HDMIOUT1 ON[CR/LF] **Description:** OUT1 audio is mute		
56	Set Audio Switch	**Syntax** **Command:** **S**ET AUDSW *prm* [CR/LF] **Return:** AUDSW *prm* [CR/LF] **Parameter:** ※ *prm= {OUT1, OUT2)* **Description:** Set Audio Switch	**Example** **Command**: SET AUDSW OUT1[CR/LF] **Return**: AUDSW OUT1 [CR/LF] **Description**: Set Audio follow out1	The saved configuration data before powering off.	Deafult:OUT1
57	Get Audio Switch status	**Syntax** **Command:** GET AUDSW [CR/LF] **Return:** AUDSW *prm* [CR/LF] **Parameter:** ※ *prm= {OUT1, OUT2)* **Description:** Get Audio Switch follow	**Example** **Command**: GET AUDSW [CR/LF] **Return**: AUDSW OUT2 [CR/LF] **Description**: Get Audio follow OUT2		
58	Audio Switch Input for Output	**Syntax** **Command:** SET AUDIOSW *in out* [CR/LF] **Return:** AUDIOSW *in out* [CR/LF] **Parameter:** *in* = {in1, in2, in3,}; *out* = {out1, out2, out3,all}; **Description:** ※ AUDIOSW is short for audio switch ※ Set Audio Switch form one input source to one output sink	**Example** **Command:** SET AUDIOSW *in1 out1* [CR/LF] **Return:** AUDIOSW *in1 out1* [CR/LF] **Description:** Set Audio Switch from in1 to out1		
59	Get which audio input mapping to the indicate Output	**Syntax** **Command:** GET AUDIOMP *out* [CR/LF] **Return:** AUDIOMP *in out* [CR/LF] **Parameter:** *in* = {in1, in2, in3}; *out* = {out1, out2, out3,all}; **Description:** ※AUDIOSW is short for audio switch ※Get which input mapping to the indicate Output	**Example** **Command:** GET AUDIOMP *out1* [CR/LF] **Return:** AUDIOMP *in1 out1* [CR/LF] **Description:** Get which is mapping to out1, the result is in1.		
60	Set whether to enable dante IN output to embedded video output or USB host audio.	**Syntax** **Command:** SET DANTE_IN *prm1 prm2* [CR/LF] **Return:** DANTE_IN *prm1 prm2* [CR/LF] **Parameter:** *prm1* = {EMBED_VIDEO_OUT, USB_HOST_AUDIO,all); *prm2* = {on,off}; **Description:** ※Set whether to enable dante IN output to embedded video output or USB host audio.	**Example** **Command:** SET DANTE_IN *EMBED_VIDEO_OUT on* [CR/LF] **Return:** DANTE_IN *EMBED_VIDEO_OUT on* [CR/LF] **Description:** Set to enable dante IN output to embedded video output.		
61	Get the status of whether to enable dante IN output to embedded video output or USB host audio.	**Syntax** **Command:** GET DANTE_IN *iprm1* [CR/LF] **Return:** DANTE_IN *prm1 prm2* [CR/LF] **Parameter:** *prm1* = {EMBED_VIDEO_OUT, USB_HOST_AUDIO,all); *prm2* = {on,off}; **Description:** ※Get whether to enable dante IN output to embedded video output or USB host audio.	**Example** **Command:** GET DANTE_IN *EMBED_VIDEO_OUT* [CR/LF] **Return:** DANTE_IN *EMBED_VIDEO_OUT on* [CR/LF] **Description:** Get to enable dante IN output to embedded video output.		
		EDID		**Reboot Value**	**Reset Value**
62	Set EDID IN mode	**Syntax** **Command:** **S**ET EDID *prm1 prm2* [CR/LF] **Return:** EDID *prm1 prm2* [CR/LF] **Parameter:** ※prm1= {IN1,IN2,IN3,IN4,IN5,ALL} ※prm2= {1,2,3,4,5,6,7,8,9,10,11} ※mode1：Fixed 4K@60Hz 2.0ch (Default). ※mode2：Copy from the HDMI OUT 1. ※mode3：Copy from the HDMI OUT 2. ※mode4：Copy from the HDBT OUT 2. ※mode5：Fixed 4K@60Hz 7.1ch. ※mode6：Fixed 4K@30Hz 2.0ch. ※mode7：Fixed 4K@30Hz 7.1ch. ※mode8：1080P@60 2.0ch ※mode9：1080P@60 5.1ch ※mode10：1080P@60 7.1ch ※mode11：Custom EDID 1 ※mode12：Custom EDID 2 **Description:** Set the HDMI IN1 EDID mode	**Example** **Command**: SET EDID IN1 1[CR/LF] **Return**: EDID IN1 1[CR/LF] **Description**: Set the HDMI IN1 EDID mode 1	The saved configuration data before powering off.	Deafult: ※ USB IN1 = 1. ※ USB IN2 = 1. ※ USB-C IN 3 = 1. ※ USB-C IN 4 = 1.

63	Get EDID IN mode	**Syntax** **Command:** GET EDID *prm1* [CR/LF] **Return:** EDID *prm1 prm2* [CR/LF] **Parameter:** ※ *prm1= {IN1,IN2,IN3,IN4,IN45,ALL}* ※ *prm2= {1,2,3,4,5,6,7,8,9,10,11}* ※**mode 1**：Fixed 4K@60Hz 2.0ch (Default). ※mode2：Copy from the HDMI OUT 1. ※mode3：Copy from the HDMI OUT 2. ※mode4：Copy from the HDBT OUT 2. ※mode5：Fixed 4K@60Hz 7.1ch. ※mode6：Fixed 4K@30Hz 2.0ch. ※mode7：Fixed 4K@30Hz 7.1ch. ※mode8：1080P@60 2.0ch ※mode9：1080P@60 5.1ch ※mode10：1080P@60 7.1ch ※mode11：Custom EDID 1 ※mode12：Custom EDID 2 **Description:** Get the HDMI IN1 EDID mode	**Example** **Command**: GET EDID IN1[CR/LF] **Return**: EDID IN1 1[CR/LF] **Description**: Get the HDMI IN1 EDID mode 1		
64	Set Custum EDID	**Syntax** **Command:** **S**ET EDID_W *prm1 prm2 prm3* [CR/LF] **Return:** EDID_W *prm1* [CR/LF] **Parameter:** ※ *prm1= {cedid1, cedid2}* ※ *prm2= {block0,block1}* ※ *prm3= {128 bytes}* **Description:** Set the custom EDID	**Example** **Command**: Set EDID_W cedid1 block0 {xxxxxxxx}[CR/LF] **Return**: EDID_W cedid1 block0 success [CR/LF] **Description**: Set custom EDID ID	The saved configuration data before powering off.	All 0
65	Get Custum EDID	**Syntax** **Command:** GET EDID_W *prm1* [CR/LF] **Return:** EDID_W *prm1 prm2 prm3* [CR/LF] **Parameter:** ※*prm1= {cedid1, cedid1}* ※*prm2= {block0,block1}* ※*prm3= {128 bytes}* **Description:** Get the custom EDID	**Example** **Command**: Get EDID_W cedid1 [CR/LF] **Return**: EDID_W cedid1 block0 {128bytes} [CR/LF] EDID_W cedid1 block1 {128bytes} [CR/LF] **Description**: Get custom EDID1		
66	Get/ Read Output EDID	**Syntax** **Command:** GET EDID_R *prm1* [CR/LF] **Return:** EDID_R *prm1 prm2* [CR/LF] **Parameter:** ※*prm1= {OUT1, OUT2,OUT3,OUT4}* ※*prm2= {block0, block1}* ※*prm3= {one block with 128 bytes edid asscii data with no space, error, unconnected}* **Description:** Read Edid Content from output. OUT1:hdmi out1 OUT2:hdmi out2 OUT3:hdbt out1 OUT4:hdbt out2	**Example** **Command**: Get EDID_R out1 [CR/LF] **Return**: EDID_R OUT1 block0 {xxxxxx} [CR/LF] EDID_R OUT1 block1 {xxxxxx} [CR/LF] ------read edid ok or EDID_R OUT1 error[CR/LF] --- check sum Error or EDID_R OUT1 unconnect[CR/LF]---sink unconnect **Description**: Get EDID of output 1		
		USB-C SETTINGS		**Reboot Value**	**Reset Value**
67	Set USBC4 DP mode	**Syntax** **Command:** SET USBC4_DM *prm* [CR/LF] **Return:** USBC4_DM *prm* [CR/LF] **Parameter:** ※*prm= {SST, MST}* **Description:** Set USBC4 DP mode	**Example** **Command**: SET USBC4_DM SST[CR/LF] **Return**: USBC4_DM SST [CR/LF] **Description**: Set USBC4 DP mode to SST	The saved configuration data before powering off.	Deafult:SST
68	Get USBC4 DP mode	**Syntax** **Command:** GET USBC4_DM [CR/LF] **Return:** USBC4_M prm [CR/LF] **Parameter:** ※*prm= {SST, MST}* **Description:** Get USBC3 SST MODE status	**Example** **Command**: GET USBC4_DM [CR/LF] **Return**: USBC4_M MST [CR/LF] **Description**: Get the USBC4 DP mode is MST		
69	Set under USBC4 MST mode the strategy of USBC3 and HDMI2	**Syntax** **Command:** SET USBC4_STRA *prm* [CR/LF] **Return:** USBC4_STRA *prm* [CR/LF] **Parameter:** ※ *prm= {MODE1, MODE2}* ※*MODE1:one usbc + two hdmi* ※*MODE2:two usbc + 0ne hdmi* **Description:** Set under USBC4 MST mode the strategy of USBC3 and HDMI2	**Example** **Command**: SET USBC4_STRA MODE1[CR/LF] **Return**: USBC4_STRA MODE1 [CR/LF] **Description**: Set select one USB-C and hdmi 2	The saved configuration data before powering off.	Deafult:one usbc + two hdmi
70	Get the strategy of USBC3 and HDMI2 under USBC4 MST mode	**Syntax** **Command:** GET USBC4_STRA [CR/LF] **Return:** USBC4_STRA prm [CR/LF] **Parameter:** ※*prm= {MODE1, MODE2}* ※*MODE1:one usbc + two hdmi* ※*MODE2:two usbc + 0ne hdmi* **Description:** Get the strategy of USBC3 and HDMI2 under USBC4 MST mode	**Example** **Command**: GET USBC4_STRA [CR/LF] **Return**: USBC4_STRA MODE1 [CR/LF] **Description**: Get the strategy of USBC3 and HDMI2 under USBC4 MST mode select two USB C + one HDMI.		
		LOG SETTINGS		**Reboot Value**	**Reset Value**

71	Set Log Debug mode	**Syntax** **Command:** SET DBG *prm* [CR/LF] **Return:** DBG *prm* [CR/LF] **Parameter:** ※*prm= {off, on}* **Description:** Set Log Debug mode	**Example** **Command**: SET DBG on[CR/LF] **Return**: DBG on [CR/LF] **Description**: Set Log debug mode on	The saved configuration data before powering off.	Deafult:off
72	Get Log Debug mode	**Syntax** **Command:** GET DBG [CR/LF] **Return:** ASW *DBG* [CR/LF] **Parameter:** ※*prm= {off}* **Description:** Get Log Debug mode	**Example** **Command**: GET DBG [CR/LF] **Return**: DBG off [CR/LF] **Description**: Get Log debug mode is off		
		GPIO & RELAY		**Reboot Value**	**Reset Value**
73	Get the mode for the GPIO	**Syntax** **Command:** GET GPIO_TYPE prm1[CR/LF] **Return:** GPIO_TYPE prm1 prm2[CR/LF] **Parameter:** ※ prm1 = {1, 2, 3, 4,5,6}; ※ prm2 = {IN, OUT}; **Description:** Set the mode for the GPIO	**Example** **Command:** GET GPIO_TYPE 1[CR/LF] **Return:** GPIO_TYPE 1 IN[CR/LF] **Description:** Set the mode 1 for the GPIO		
74	Set the mode for the GPIO	**Syntax** **Command:** SET GPIO_TYPE prm1 prm2[CR/LF] **Return:** GPIO_TYPE prm1 prm2[CR/LF] **Parameter:** ※ prm1 = {1, 2, 3, 4,5,6}; ※ prm2 = {IN, OUT}; **Description:** Set the mode for the GPIO	**Example** **Command:** SET GPIO_TYPE 1 IN[CR/LF] **Return:** GPIO_TYPE 1 IN[CR/LF] **Description:** Set the mode 1 for the GPIO		
75	Set the configuration of GPIO IN	**Syntax** **Command:** SET GPIO_IN_CTL prm1 prm2[CR/LF] **Return:** GPIO_IN_CTL prm1 prm2[CR/LF] **Parameter:** ※ prm1 = {1, 2, 3, 4}; ※ prm2 = {high, low} **Description:** Set the mode for the GPIO	**Example** **Command:** SET GPIO_IN_CTL 1 HIGH[CR/LF] **Return:** GPIO_IN_CTL 1 HIGH[CR/LF] **Description:** Set the mode 1 for the GPIO	**prm2 = 1, prm3 = 2**	
76	Get the configuration of GPIO IN	**Syntax** **Command:** GET GPIO_IN_CTL prm1[CR/LF] **Return:** GPIO_IN_CTL prm1 prm2[CR/LF] **Parameter:** ※ prm1 = {1, 2, 3, 4,5,6}; ※ prm2 = {high, low} **Description:** Set the mode for the GPIO	**Example** **Command:** GET GPIO_IN_CTL 1[CR/LF] **Return:** GPIO_IN_CTL 1 HIGH[CR/LF] **Description:** Set the mode 1 for the GPIO	**prm2 = 1, prm3 = 2**	
77	Set GPIO OUT configuration	**Syntax** **Command:** SET GPIO_OUT_CTL prm1 prm2 prm3[CR/LF] **Return:** GPIO_OUT_CTL prm1 prm2 prm3[CR/LF] **Parameter:** ※ prm1 = {1, 2, 3, 4,5,6}; ※ prm2 = {push-pull, open-drain}; ※ prm3 = {high, low} **Description:** Set the mode for the GPIO	**Example** **Command:** SET GPIO_OUT_CTL 1 PUSH-PULL HIGH[CR/LF] **Return:** GPIO_OUT_CTL 1 PUSH-PULL HIGH[CR/LF] **Description:** Set the mode 1 for the GPIO		
78	Get GPIO OUT configuration	**Syntax** **Command:** GET GPIO_OUT_CTL prm1[CR/LF] **Return:** GPIO_OUT_CTL prm1 prm2 prm3[CR/LF] **Parameter:** ※ prm1 = {1, 2, 3, 4,5,6}; ※ prm2 = {push-pull, open-drain}; ※ prm3 = {high, low} **Description:** Set the mode for the GPIO	**Example** **Command:** GET GPIO_OUT_CTL 1[CR/LF] **Return:** GPIO_OUT_CTL 1 PUSH-PULL HIGH[CR/LF] **Description:** Set the mode 1 for the GPIO		
79	Set the mode for the RELAY	**Syntax** **Command:** SET RELAY prm1 prm2[CR/LF] **Return:** RELAY prm1 prm2[CR/LF] **Parameter:** ※ prm1 = {1, 2,all}; ※ prm2 = {ON, OFF}; **Description:** Set the mode for the RELAY ON:Latch OFF:Momentary	**Example** **Command:** SET RELAY 1 ON[CR/LF] **Return:** RELAY 1 ON[CR/LF] **Description:** Set the mode 1 for the RELAY		

80	Get the mode for the RELAY	**Syntax** **Command:** GET RELAY prm1[CR/LF] **Return:** RELAY prm1 prm2[CR/LF] **Parameter:** ※ prm1 = {all}; ※ prm2 = {ON, OFF}; **Description:** Set the mode for the RELAY ON:Latch OFF:Momentary	**Example** **Command:** GET RELAY all[CR/LF] **Return:** RELAY 1 ON[CR/LF] **Description:** Set the mode 1 for the RELAY		
81	Set pulse time in Momentray mode	**Syntax** **Command:** SET PULSE TIME prm1 prm2[CR/LF] **Return:** PULSE TIME prm1 prm2[CR/LF] **Parameter:** ※ prm1 = {1, 2}; ※ prm2 = {3, 4, 5,...}; **Description:** Set pulse time in Momentray mode The default time is 3 seconds	**Example** **Command:** SET PULSE TIME 1 10[CR/LF] **Return:** PULSE TIME 1 10[CR/LF] **Description:** Set the pulse time to 10 seconds		
82	Get pulse time in Momentray mode	**Syntax** **Command:** GET PULSE TIME prm1[CR/LF] **Return:** PULSE TIME prm1 prm2[CR/LF] **Parameter:** ※ prm1 = {1, 2}; ※ prm2 = {3, 4, 5,...}; **Description:** Get pulse time in Momentray mode The default time is 3 seconds	**Example** **Command:** GET PULSE TIME 1[CR/LF] **Return:** PULSE TIME 1 10[CR/LF] **Description:** Get the pulse time to 10 seconds		
83	Set the sensor 24V switch	**Syntax** **Command:** SET SENSOR 24V prm1 [CR/LF] **Return:** SENSOR 24V prm1 [CR/LF] **Parameter:** ※ prm1 = {ON, OFF}; **Description:** Set the sensor 24V switch	**Example** **Command:** SET SENSOR 24V on [CR/LF] **Return:** SENSOR 24V on [CR/LF] **Description:** Set sensor 24V on		
84	Get the sensor 24V switch status	**Syntax** **Command:** GET SENSOR 24V [CR/LF] **Return:** SENSOR 24V prm1 [CR/LF] **Parameter:** ※ prm1 = {ON, OFF}; **Description:** Get the sensor 24V switch	**Example** **Command:** GET SENSOR 24V [CR/LF] **Return:** SENSOR 24V on[CR/LF] **Description:** Get sensor 24V on		
85	Get SENSOR IO status	**Syntax** **Command:** GET SENSOR IO [CR/LF] **Return:** SENSOR IO prm1 [CR/LF] **Parameter:** ※ prm1 = {high, low} **Description:** Get SENSOR IO status	**Example** **Command:** GET SENSOR IO[CR/LF] **Return:** SENSOR IO LOW [CR/LF] **Description:** Get IO status as low		
86	Set USB Device Configuration	**Syntax** **Command:** SET USB_DEVICE prm1 prm2 [CR/LF] **Return:** USB_DEVICE prm1 prm2 [CR/LF] **Parameter:** ※ prm1 = {usb-a-1,usb-a-2,usb-a-3,usb-a-4,usb-hdbt,all}; ※ prm2 = {Pass-Through, Alway-High}; **Description:** Set USB Device Configuration	**Example** **Command:** SET USB_DEVICE usb-a-1 Pass-Through[CR/LF] **Return:** USB_DEVICE usb-a-1 Pass-Through[CR/LF] **Description:** Set USB-A-1 mode to Pass Through		
87	Get USB Device Configuration	**Syntax** **Command:** GET USB_DEVICE prm1[CR/LF] **Return:** USB_DEVICE prm1 prm2[CR/LF] **Parameter:** ※ prm1 = {usb-a-1,usb-a-2,usb-a-3,usb-a-4,usb-hdbt,all}; ※ prm2 = {Pass-Through, Alway-High}; ※ prm3 = {ON, OFF}; **Description:** Get USB Device Configuration	**Example** **Command:** GET USB_DEVICE usb-a-1[CR/LF] **Return:** USB_DEVICE usb-a-1 Pass-Through[CR/LF] **Description:** Get USB-A-1 mode to Pass Through and turn		
88	Set USB Device 5V	**Syntax** **Command:** SET USB_DEVICE_5V prm1 prm2 [CR/LF] **Return:** USB_DEVICE_5V prm1 prm2 [CR/LF] **Parameter:** ※ prm1 = {usb-a-1,usb-a-2,usb-a-3,usb-a-4,usb-hdbt,all}; ※ prm2 = {ON, OFF}; **Description:** Set USB Device 5V.	**Example** **Command:** SET USB_DEVICE_5V usb-a-1 ON[CR/LF] **Return:** USB_DEVICE_5V usb-a-1 ON[CR/LF] **Description:** Set USB-A-1 5V.		
89	Get USB Device 5V	**Syntax** **Command:** GET USB_DEVICE_5V prm1[CR/LF] **Return:** USB_DEVICE_5V prm1 prm2[CR/LF] **Parameter:** ※ prm1 = {usb-a-1,usb-a-2,usb-a-3,usb-a-4,usb-hdbt,all}; ※ prm2 = {ON, OFF}; **Description:** Get USB Device 5V.	**Example** **Command:** SET USB_DEVICE_5V usb-a-1[CR/LF] **Return:** USB_DEVICE_5V usb-a-1 ON[CR/LF] **Description:** Get USB-A-1 5V		