RAV-SW-4x1HDMI/USB-C

RS-232 Control Commands

RAV-SW-4x1HDMI/USB-C

4×1 auto-switcher with video conferencing support

Description	Syntax	Example
Returns a list of all API commands	**Command:** help[CR/LF] **Return:** xxxx **Parameter:** NA **Description:** Get the API list	**Command:** help[CR/LF] **Return:** xxxx **Description:** Get the API list
Get 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:** MCU VER 1.0, ARM VER 1.0[CR/LF] **Description:** Get all module firmware version
Factory reset	**Command:** RESET[CR/LF] **Return:** RESET[CR/LF] **Parameter:** **Description:** Factory reset	**Command:** RESET[CR/LF] **Return:** RESET[CR/LF] **Description:** Factory reset all board

System reboot	**Command:** REBOOT[CR/LF] **Return:** REBOOT[CR/LF] **Parameter:** **Description:** system reboot	**Command:** REBOOT[CR/LF] **Return:** REBOOT[CR/LF] **Description:** System reboot
Sets Telnet for New username and password	**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 logging in to the telnet	**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]

Sets input to output	**Command:** SET SW *out in*[CR/LF] **Return:** SW out in[CR/LF] **Parameter:** ※ *in* = {IN1, IN2, IN3, IN4}; ※ *out* = {OUT1}; **Description:** ※ SW is short for Swtich ※ Switch one input source for one output sink	**Command:** SET SW OUT1 IN2[CR/LF] **Return:** SW OUT1 IN2[CR/LF] **Description:** Swtich input 2 for hdmi ouput 1
Gets which input mapping to the indicate Output	**Command:** GET SW *out*[CR/LF] **Return:** SW *out in*[CR/LF] **Parameter:** ※ in = {IN1, IN2, IN3, IN4}; ※ out = {OUT1}; **Description:** ※Get which input mapping to the indicate Output	**Command:** GET SW *OUT2*[CR/LF] **Return:** SW OUT2 IN1[CR/LF] **Description:** Get which input mapping to output 2
Sets the status of the automatic switching function status	**Command:** SET AUTOSW_FN *prm*[CR/LF] **Return:** AUTOSW_FN *prm*[CR/LF] **Parameter:** ※ *prm* = {on, off} **Description:** Set whether to open the video and USB automatic switching mode	**Command:** SET AUTOSW_FN *on*[CR/LF] **Return:** AUTOSW_FN *on*[CR/LF] **Description:** Set to open video and USB auto switch mode

Gets the status of the automatic switching function	**Command:** GET AUTOSW_FN[CR/LF] **Return:** AUTOSW_FN *prm*[CR/LF] **Parameter:** ※ *prm* = {on, off} **Description:** Get the video and USB automatic switching mode status	**Command:** GET AUTOSW_FN[CR/LF] **Return:** AUTOSW_FN *on*[CR/LF] **Description:** video and USB auto switch mode is open
Sets CEC/PWR On/Off	**Command:** SET CEC_PWR *out prm*[CR/LF] **Return:** CEC_PWR *out prm*[CR/LF] **Parameter:** ※ *prm* = {on,off} ※ *out* = {OUT1}; **Description:** Set sink power on or off	**Command:** SET CEC_PWR OUT1 *on*[CR/LF] **Return:** CEC_PWR OUT1 *on*[CR/LF] **Description:** Set sink hdmi output 1 power on
Sets 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}; **Description:** Set sink auto power Function ON or OFF **Default:** on	**Command:** SET AUTOCEC_FN *OUT1 on*[CR/LF] **Return:** AUTOCEC_FN OUT1 *on*[CR/LF] **Description:** Set sink hdmi output 1 auto power ON

Gets 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}; **Description:** Get Sink auto power Function ON or OFF Status. **Default:** on	**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.
Sets CEC POWER Delay Time	**Command:** SET AUTOCEC_D *out prm*[CR/LF] **Return:** AUTOCEC_D *out prm*[CR/LF] **Parameter:** ※ *out* = {OUT1}; ※ *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. **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.

Gets CEC POWER Delay Time Status	**Command:** GET AUTOCEC_D *out*[CR/LF] **Return:** AUTOCEC_D *out prm*[CR/LF] **Parameter:** ※ *out* = {OUT1}; ※ *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. **Description:** AUTOCEC_D is short for CEC auto Power Delay Timing **Default:** 2	**Command:** GET AUTOCEC_D OUT1[CR/LF] **Return:** AUTOCEC_D OUT*1 2*[CR/LF] **Description:** Get hdmi1 auto power delay time, the result is 2 minutes
Sets CEC Volume increase	**Syntax** **Command:** SET CEC_VOL_INC *out*[CR/LF] **Return:** CEC_VOL_INC *out* [CR/LF] **Parameter:** ※ *out* = {OUT1}; **Description:** Control device volume increase via CEC	**Example** **Command:** SET CEC_VOL_INC *OUT1*[CR/LF] **Return:** CEC_VOL_INC OUT1 [CR/LF] **Description:**

Sets CEC Volume decrease	**Command:** SET CEC_VOL_DEC *out*[CR/LF] **Return:** CEC_VOL_DEC *out* [CR/LF] **Parameter:** ※ *out* = {OUT1}; **Description:** Control device volume decrease via CEC	**Command:** SET CEC_VOL_DEC *OUT1*[CR/LF] **Return:** CEC_VOL_DEC OUT1 [CR/LF] **Description:**
Sets CEC Volume mute	**Syntax** **Command:** SET CEC_VOL_MUTE *out*[CR/LF] **Return:** CEC_VOL_MUTE *out prm*[CR/LF] **Parameter:** ※ *out* = {OUT1}; ※ *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
Sets RS232's baud rate	**Syntax** **Command:** SET UART_B prm1[CR/LF] **Return:** UART_B out *prm1*[CR/LF] **Parameter:** ※ *prm1* = {115200, 57600, 38400, 19200, 9600,4800,2400} **Description:** 115200	**Example** **Command:** SET UART_B *57600*[CR/LF] **Return:** UART_B 57600[CR/LF] **Description:** Set the baud rate of RS232 to 57600

Gets RS232's baud rate	**Command:** GET UART_B[CR/LF] **Return:** UART_B *prm*[CR/LF] **Parameter:** ※ *prm* = {115200, 57600, 38400, 19200, 9600,4800,2400} **Description:** 115200	**Command:** GET UART_B[CR/LF] **Return:** UART_B 57600[CR/LF] **Description:** The baud rate of RS232 to 57600
Sets RS232's auto control power ON/OFF	**Command:** SET UARTPWR_FN *prm1*[CR/LF] **Return:** UARTPWR_FN *prm1* *prm2*[CR/LF] **Parameter:** *prm1* = {on,off} **Description:** Set UART auto pwr function ON or OFF.	**Command:** SET UARTPWR_FN *on*[CR/LF] **Return:** UARTPWR_FN *on*[CR/LF] **Description:** Set uart1 auto pwr function on
Gets RS232's auto control power ON/OFF status	**Command:** GET UARTPWR_FN[CR/LF] **Return:** UARTPWR_FN *prm1* [CR/LF] **Parameter:** ※ *prm1 = {on,off}* **Description:** Get uart auto pwr status	**Command:** GET UARTPWR_FN *on*[CR/LF] **Return:** UARTPWR_FN *on*[CR/LF] **Description:** Get uart1 auto pwr status ,and status is on

Sets RS232's auto control power delay time	**Command:** SET UARTPWR_D *prm1*[CR/LF] **Return:** UARTPWR_D *prm1*[CR/LF] **Parameter:** *prm1* = {0,1,2,3...,30}// 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 immediatly. **Description:** UARTPWR_D is short for UART Power Delay Timing	**Command:** SET UARTPWR_D *2* [CR/LF] **Return:** UARTPWR_D *2*[CR/LF] **Description:** when no active signal to uart1, 2 minutes later,the unit will auto power off.
Gets RS232's auto control power delay time	**Command:** GET UARTPWR_D[CR/LF] **Return:** UARTPWR_D *prm1* *prm2*[CR/LF] **Parameter:** *prm2* = {0,1,2,3...30}// 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 immediatly. **Description:** UARTPWR_D is short for UART Power Delay Timing	**Command:** GET UARTPWR_D[CR/LF] **Return:** UARTPWR_D *2*[CR/LF] **Description:** Get uart1 auto power delay time, the result is 3 minutes

Sets RS232's command	**Command:** SET UART_CMD *prm1 prm2 prm3*[CR/LF] **Return:** UART_CMD *prm1 prm2 prm3*[CR/LF] **Parameter:** *prm1* = {poweron, poweroff, volumemute, volumeunmute, volumeup, volumedown…} *prm2* = {hex, str} *prm3* = {xxxxxxxxxxx} (max size is 48 byte) **Description:** control display via RS232 command	**Command:** SET UART_CMD poweroff hex 30 31 32 33[CR/LF] **Return:** UART_CMD poweroff hex 30 31 32 33[CR/LF] **Description:** SET UART CMD
Gets RS232's command	**Command:** GET UART_CMD [CR/LF] **Return:** UART_CMD *prm1 prm2 prm3 prm4*[CR] UART_CMD *prm1 prm2 prm3 prm4*[CR] … UART_CMD *prmn*[CR/LF] **Parameter:** *prm1* = {poweron,poweroff,volumemute,volumeunmute,volumeinc,volumedec…} *prm2* = {hex, str} *prm3* = {xxxxxxxxxxx}//hex is ascii character **Description:** Get all RS232 command	**Command:** GET UART_CMD [CR/LF] **Return:** UART_CMD *poweron str 123*[CR/LF] UART_CMD *poweroff hex 31 32 33*[CR/LF] UART_CMD volumemute hex 31 32 33[CR/LF] UART_CMD volumeinc hex 31 32 33[CR/LF] UART_CMD volumedec hex 31 32 33[CR/LF] **Description:** Get UART CMD

Sets RS232's command to send	**Command:** SET UART_CMD_S *prm1*[CR/LF] **Return:** UART_CMD_S *prm2*[CR/LF] **Parameter:** *prm1* = {poweron, poweroff, volumemute, volumeunmute , volumeinc, volumedec...} **Description:** send via RS232 command	**Command:** SET UART_CMD_S *poweron*[CR/LF] **Return:** *UART_CMD_S poweron[CR/LF]* **Description:** SEND UART CMD poweron
Sets IP Mode	**Command:** SET IP MODE prm[CR/LF] **Return:** IP MODE prm [CR/LF] **Parameter:** ※pcm= {static ,dhcp} **Description:** ※Set IP mode **Default:** DHCP	**Command:** SET IP MODE dhcp[CR/LF] **Return:** IP MODE dhcp [[CR/LF] **Description:** Set IP mode is dhcp
Gets IP Mode	**Command:** GET IP MODE[CR/LF] **Return:** IP MODE prm [CR/LF] **Parameter:** ※pcm= {static ,dhcp} **Description:** ※Get IP mode **Default:** DHCP	**Command:** GET IP MODE[CR/LF] **Return:** IP MODE dhcp [[CR/LF] **Description:** IP mode is dhcp

Sets IP address	**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	**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
Gets IP address	**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	**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
Set Nic status (Ethernet adapter)	**Command:** **S**ET NIC_STATUS prm[CR/LF] **Return:** NIC_STATUS prm[CR/LF] **Parameter:** ※pcm= {on, off} **Description:** Set the switch status of the NIC function	**Command:** SET NIC_STATUS ON[CR/LF] **Return:** NIC_STATUS ON[CR/LF] **Description:** Set the switch status of the NIC function to on

Get Nic status	**Command:** GET NIC_STATUS[CR/LF] **Return:** NIC_STATUS prm[CR/LF] **Parameter:** ※pcm= {on, off} **Description:** Get the switch status of the NIC function	**Command:** GET NIC_STATUS[CR/LF] **Return:** NIC_STATUS ON[CR/LF] **Description:** Get the switch status of the NIC function as on
Gets the status of the video input	**Command:** GET VIDIN_SIG *in*[CR/LF] **Return:** VIDIN_SIG *in prm*[CR/LF] **Parameter:** ※ *in* = {IN1, IN2, IN3, IN4, all}; ※ *prm* = {no, valid} **Description:** Get the status of the video input	**Command:** GET VIDIN_SIG IN1[CR/LF] **Return:** VIDIN_SIG IN1 valid[CR/LF] **Description:** Get the status of the video input1
Gets HDCP version of input video	**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} **Description:** Get hdcp version of input video	**Command:** GET VIDIN_HDCP *IN1*[CR/LF] **Return:** VIDIN_HDCP *IN1 no hdcp*[CR/LF] **Description:** Get the HDCP status of HDMI input1

Gets Output HDCP support status	**Command:** GET HDCP *out*[CR/LF] **Return:** HDCP *out prm*[CR/LF] **Parameter:** ※ *out* = {OUT1}; ※ *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	**Command:** GET HDCP *OUT1*[CR/LF] **Return:** HDCP *OUT1 follow*[CR/LF] **Description:** Get output hdcp is follow mode
Sets Web New username and password	**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	**Command:** 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

Set USB work model	**Command:** SET USB_M *prm*[CR/LF] **Return:** USB_M *prm*[CR/LF] **Parameter:** *prm* = {follow,independent} **Description:** Set USB Work Model	**Command:** SET USB_M *follow*[CR/LF] **Return:** USB_M *follow*[CR/LF] **Description:** Set usb work model is follow
Get USB Work Model	**Command:** GET USB_M[CR/LF] **Return:** USB_M *prm*[CR/LF] **Parameter:** *prm* = {follow,independent} **Description:** Get USB Work Model	**Command:** GET USB_M[CR/LF] **Return:** USB_M *follow*[CR/LF] **Description:** Get usb work mode,and the work mode is follow
Set USB Switch Input for specific output	**Command:** SET USBSW *out in*[CR/LF] **Return:** USBSW *out in*[CR/LF] **Parameter:** *in* = {i00,i01,i02,i03,i04}; *out* = {o01,all}; **Description:** ※ i00 is blank ※USBSW is short for usb switch ※Set USB Switch form one input source to all output sink	**Command:** SET USBSW *o01 i01*[CR/LF] **Return:** USBSW *o01 i01*[CR/LF] **Description:** Set USB Switch from USB input1 to USB output1

Get which USB input mapping to the indicate output	**Command:** GET USBSW *out*[CR/LF] **Return:** USBSW *out in*[CR/LF] **Parameter:** *in = {i00,i01,i02,i03,i04};* *out = {o01,all};* **Description:** ※USBSW is short for usb switch ※Get which input mapping to the indicate Output	**Command:** GET USBSW *o01*[CR/LF] **Return:** USBSW o01 *i02*[CR/LF] **Description:** Get which input mapping to USB out1
Set Video Port Priority	**Command:** SET VIDEO_P *prm1 prm2*[CR/LF] **Return:** VIDEO_P *prm1 prm2*[CR/LF] **Parameter:** *prm1* = {i01,i02,i03,i04,all} ※ i01 = USB-C IN (video). ※ i02 = HDMI IN 2. ※ i03 = HDMI IN 3. ※ i04 = HDMI IN 4. ※ all = all video port. *prm2*= {0,1,2,3,4} ※ 0 = This port is ignored. ※ 1 = Level 1 (Highest priority). ※ 2 = Level 2 . ※ 3 = Level 3 . ※ 4 = Level 4 (Lowest priority). **Description:** Set video port priority of Auto-switching.	**Command:** SET VIDEO_P i01 1[CR/LF] **Return:** VIDEO_P i01 1[CR/LF] **Description:** USB-C port priority set to level 1

Get Video Port Priority	**Command:** GET VIDEO_P *prm1*[CR/LF] **Return:** VIDEO_P *prm1*[CR/LF] **Parameter:** *prm1* = {i01,i02,i03,i04,all} ※ i01 = USB-C IN (video). ※ i02 = HDMI IN 2. ※ i03 = HDMI IN 3. ※ i04 = HDMI IN 4. ※ all = all video port. **Description:** Get video port priority of Auto-switching.	**Command:** GET VIDEO_P all[CR/LF] **Return:** VIDEO_P i01 1[CR/LF] VIDEO_P i02 2[CR/LF] VIDEO_P i03 3[CR/LF] VIDEO_P i04 4[CR/LF] **Description:** USB-C port priority is level 1 HDMI IN 2 priority is level 2 HDMI IN 3 priority is level 3 HDMI IN 4 priority is level 4
Set USB Port Priority	**Command:** SET USB_P *prm1 prm2*[CR/LF] **Return:** USB_P *prm1 prm2*[CR/LF] **Parameter:** *prm1* = {i01,i02,i03,i04,all} ※ i01 = USB-C IN (video). ※ i02 = USB IN 2. ※ i03 = USB IN 3. ※ i04 = USB IN 4. ※ all = all video port. *prm2*= {0,1,2,3,4} ※ 0 = This port is ignored. ※ 1 = Level 1 (Highest priority). ※ 2 = Level 2 . ※ 3 = Level 3 . ※ 4 = Level 4 (Lowest priority). **Description:** Set USB port priority of Auto-switching.	**Command:** SET USB_P i02 1[CR/LF] **Return:** USB_P i02 1[CR/LF] **Description:** USB IN 2 port priority set to level 1

Get USB Port Priority	**Command:** GET USB_P *prm1*[CR/LF] **Return:** USB_P *prm1*[CR/LF] **Parameter:** *prm1* = {i01,i02,i03,i04,all} ※ i01 = USB-C IN (video). ※ i02 = USB IN 2. ※ i03 = USB IN 3. ※ i04 = USB IN 4. ※ all = all video port. **Description:** Get USB port priority of Auto-switching.	**Command:** GET USB_P all[CR/LF] **Return:** USB_P i01 1[CR/LF] USB_P i02 2[CR/LF] USB_P i03 3[CR/LF] USB_P i04 4[CR/LF] **Description:** USB-C port priority is level 1 USB IN 2 priority is level 2 USB IN 3 priority is level 3 USB IN 4 priority is level 4