# AT-OME2-R2742

## Omega 2 HDBaseT 3 4x2 Video Switcher with Dante Audio

## API Command Reference

October 2026



# Table of Contents

**Version Information** ..... **4**
Sales & Customer Support ..... 4
**Introduction** ..... **5**
General ..... 5
Telnet ..... 5
Ports ..... 5
RS-232 ..... 5
Port and Parameter Names ..... 6
**Command Quick Reference** ..... **7**
System ..... 7
Switch ..... 7
CEC ..... 7
RS-232 ..... 8
Network ..... 8
Video Information ..... 8
HDCP Information ..... 9
Web Login ..... 9
USB Matrix ..... 9
Audio ..... 9
Priority Control ..... 9
EDID ..... 10
USB-C Settings ..... 10
GPIO & Relay ..... 10
**Commands** ..... **12**
System ..... 12
Switch ..... 14
CEC ..... 17
RS-232 ..... 21
Network ..... 26
Video Information ..... 29
HDCP Information ..... 32
Web Login ..... 33
USB Matrix ..... 34
Audio ..... 37
Priority Control ..... 40
EDID ..... 42

USB-C Settings.....45
GPIO & Relay.....47

# Version Information

This document lists the commands for controlling the AT-OME2-R2742 over RS-232 and Telnet.

| Version | Date | Notes |
|---|---|---|
| HR 1.0 | October 2026 | Initial release. Covers API command set version 2.0.3. |

## Sales & Customer Support

| Item | Details |
|---|---|
| **Main Office (US)** | Hall Research, 1234 Lakeshore Dr Suite #150, Coppell, TX 75019 |
| **Support Hours** | Monday – Friday, 8:00 a.m. – 6:00 p.m. (CST) |
| **Website** | https://www.hallresearch.com |
| **Product Page (AT-OME2-R2742)** | https://www.hallresearch.com/product/AT-OME2-R2742 |

**IMPORTANT:** Visit hallresearch.com/product/AT-OME2-R2742 for the latest firmware updates and documentation.

# Introduction

## General

This document lists the API commands for the AT-OME2-R2742, grouped by function. Commands can be sent using RS-232 (RS-232 1 port) or Telnet. Most settings use a SET command, and the matching GET command returns the current setting. The feedback repeats the command keyword followed by the current values.

**IMPORTANT:** Each command is terminated with a carriage return and line feed ([CR/LF], 0x0D 0x0A), and the feedback is also terminated with a carriage return and line feed.

## Telnet

Connect to the IP address of the unit on port 23 with a Telnet client and send the API commands. The default IP mode is DHCP. To get the IP address, send GET IPADDR through the RS-232 1 port. The Telnet user name and password can be changed with the SET TEL_USNM command.

## Ports

This product provides TCP-to-RS-232 routing, a tunnel from the network to an RS-232 port. Commands sent to the TCP port are passed through to the RS-232 port, so a control system on the network can control devices connected to the unit.

| TCP Port | Description |
|---|---|
| 5000 | RS-232 1 port |
| 5001 | RS-232 2 port |
| 5002 | RS-232 over HDBaseT |

## RS-232

A double-layer 3-pin captive screw connector is provided for RS-232 (TX, RX, and ground); row 1 is RS-232 1 and row 2 is RS-232 2. To control the unit with API commands, connect the control system or PC to the RS-232 1 port.

**NOTE:** Typical DB9 connectors use pin 2 for TX, pin 3 for RX, and pin 5 for ground. On some devices, the functions of pins 2 and 3 are reversed.

The serial port settings of the unit and the controlling device must match. The baud rate can be changed in the Web UI or with the SET UART_B command. The default settings are:

| Port | Default Settings |
|---|---|
| RS-232 | 115200 baud, 8 data bits, no parity, 1 stop bit, no flow control |

## Port and Parameter Names

The commands use the following names for the ports of the unit:

| Name in Command | Port |
|---|---|
| IN1 / IN2 | HDMI IN 1 / HDMI IN 2 |
| IN3 / IN4 | USB-C IN 3 / USB-C IN 4 |
| OUT1 / OUT2 | HDMI OUT 1 / HDMI OUT 2 |
| OUT4 | HDBaseT OUT 2 |
| UART1 / UART2 | RS-232 1 / RS-232 2 |
| UART3 | RS-232 over HDBaseT |

# Command Quick Reference

Select a command name to go to its full description.

## System

| Command | Description |
|---|---|
| help | Returns a list of all API commands |
| GET VER | Gets the firmware version of the selected target |
| RESET | Resets the unit to factory defaults |
| REBOOT | Reboots the unit |
| SET TEL_USNM | Sets a new Telnet user name and password |

## Switch

| Command | Description |
|---|---|
| SET SW | Routes an input to an output |
| GET SW | Gets the input routed to an output |
| SET AUTOSW_FN | Turns automatic switching on or off |
| GET AUTOSW_FN | Gets the automatic switching status |
| SET AUTOSW_MD | Sets the video automatic switching mode |
| GET AUTOSW_MD | Gets the video automatic switching mode |

## CEC

| Command | Description |
|---|---|
| SET CEC_PWR | Turns a display on or off using CEC |
| SET AUTOCEC_FN | Turns CEC auto power on or off |
| GET AUTOCEC_FN | Gets the CEC auto power status |
| SET AUTOCEC_D | Sets the CEC auto power delay time |
| GET AUTOCEC_D | Gets the CEC auto power delay time |
| SET CECCMD_EDIT | Sets the CEC power-on or power-off command |
| GET CECCMD_EDIT | Gets the CEC power-on or power-off command |
| SET CEC_CMD | Sends a CEC command |

## RS-232

| Command | Description |
|---|---|
| SET UART_B | Sets the RS-232 baud rate |
| GET UART_B | Gets the RS-232 baud rate |
| SET UARTPWR_FN | Turns RS-232 auto power control on or off |
| GET UARTPWR_FN | Gets the RS-232 auto power control status |
| SET UARTPWR_D | Sets the RS-232 auto power delay time |
| GET UARTPWR_D | Gets the RS-232 auto power delay time |
| SET UART_CMD | Stores an RS-232 command for display control |
| GET UART_CMD | Gets the stored RS-232 commands |
| SET UART_CMD_S | Sends a stored RS-232 command |
| SET UART_S | Sends data through an RS-232 port |

## Network

| Command | Description |
|---|---|
| SET IPADDR STATIC | Sets a static IP address |
| SET IPADDR dhcp | Sets the IP mode to DHCP |
| GET IPADDR | Gets the IP address |
| SET VLAN_ENABLE | Turns VLAN on or off |
| GET VLAN_ENABLE | Gets the VLAN status |

## Video Information

| Command | Description |
|---|---|
| GET VIDIN_SIG | Gets the video input signal status |
| SET VIDIN_HDCP_FN | Turns HDCP on or off for a video input |
| GET VIDIN_HDCP_FN | Gets the HDCP on/off status of a video input |
| GET VIDIN_HDCP | Gets the HDCP version of a video input |
| GET SCALER | Gets the scaler status of an output |
| SET SCALER | Sets the scaler status of an output |

## HDCP Information

| Command | Description |
|---|---|
| GET HDCP | Gets the output HDCP mode |

## Web Login

| Command | Description |
|---|---|
| SET WEB_USNM | Sets a new Web UI user name and password |

## USB Matrix

| Command | Description |
|---|---|
| SET USB_M | Sets the USB switching mode |
| GET USB_M | Gets the USB switching mode |
| SET USBSW | Routes a USB input to the USB outputs |
| GET USBSW | Gets the USB input routed to the USB output |
| SET USBNIC | Turns the USB-to-network function on or off |
| GET USBNIC | Gets the USB-to-network status |

## Audio

| Command | Description |
|---|---|
| SET AUD_MUTE | Mutes or unmutes an audio output |
| GET AUD_MUTE | Gets the mute status of an audio output |
| SET AUDIOSW | Routes an audio input to an audio output |
| GET AUDIOMP | Gets the audio input routed to an audio output |
| SET DANTE_IN | Routes Dante input audio to embedded video output or USB host audio |
| GET DANTE_IN | Gets the Dante input audio routing |

## Priority Control

| Command | Description |
|---|---|
| SET VIDEO_P | Sets the video input priority level |
| GET VIDEO_P | Gets the video input priority level |
| SET USB_P | Sets the USB input priority level |

| Command | Description |
|---|---|
| GET USB_P | Gets the USB input priority level |

## EDID

| Command | Description |
|---|---|
| SET EDID | Sets the EDID mode of an input |
| GET EDID | Gets the EDID mode of an input |
| SET EDID_W | Writes a custom EDID |
| GET EDID_W | Reads a custom EDID |
| GET EDID_R | Reads the EDID of an output |

## USB-C Settings

| Command | Description |
|---|---|
| SET USBC4_DM | Sets the DisplayPort mode of USB-C IN 4 |
| GET USBC4_DM | Gets the DisplayPort mode of USB-C IN 4 |
| SET USBC4_STRA | Sets the USB-C IN 3 / HDMI 2 strategy in MST mode |
| GET USBC4_STRA | Gets the USB-C IN 3 / HDMI 2 strategy in MST mode |

## GPIO & Relay

| Command | Description |
|---|---|
| GET GPIO_TYPE | Gets whether a GPIO is an input or output |
| SET GPIO_TYPE | Sets a GPIO as an input or output |
| SET GPIO_IN_CTL | Sets the GPIO input configuration |
| GET GPIO_IN_CTL | Gets the GPIO input configuration |
| SET GPIO_OUT_CTL | Sets the GPIO output configuration |
| GET GPIO_OUT_CTL | Gets the GPIO output configuration |
| SET RELAY_STATUS | Sets the relay output status |
| GET RELAY_STATUS | Gets the relay output status |
| SET AUTORELAY_FN | Turns relay auto power on or off |
| GET AUTORELAY_FN | Gets the relay auto power status |
| SET RELAY_MODE | Sets the relay mode |
| GET RELAY_MODE | Gets the relay mode |

| Command | Description |
|---|---|
| SET RELAY_PULSE_TIME | Sets the relay trigger time in momentary mode |
| GET RELAY_PULSE_TIME | Gets the relay trigger time in momentary mode |
| SET RELAY_TRIGGER_MODE | Sets the relay trigger conditions |
| GET RELAY_TRIGGER_MODE | Gets the relay trigger conditions |
| SET SENSOR 24V | Turns the sensor 24V output on or off |
| GET SENSOR 24V | Gets the sensor 24V output status |
| GET SENSOR IO | Gets the sensor I/O status |
| SET USB_VBUS_MODE | Sets the VBUS mode of a USB device port |
| GET USB_VBUS_MODE | Gets the VBUS mode of a USB device port |

# Commands

## System

### help

Returns a list of all API commands supported by the unit.

**Syntax**

```
help
```

*This command does not require any parameters.*

**Example**

```
help
```

**Feedback**

```
(list of all API commands)
```

### GET VER

Gets the firmware version of the selected target. Use ALL to return the version of every target.

**NOTE:** To get a single target, send the target name, for example GET VER Main. The feedback is Main V1.1.3.

**Syntax**

```
GET VER prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Firmware target | MAIN, ARM, USB_C_VIDEO, HDMI, USB_C_CC, CPLD, HDBT_3.0, ALL |
| prm2 | Version (feedback) | According to the actual firmware version |

**Example**

```
GET VER ALL
```

**Feedback**

```
Main V1.1.3
ARM V1.1.3
USB_C_CC V1.0.1
USB_C_VIDEO V1.0.3
HDMI V1.1.0
HDBT_3 V5.4.87
CPLD V1.0.0
```

## RESET

Resets the unit to factory defaults, including all boards.

**Syntax**

```
RESET
```

*This command does not require any parameters.*

**Example**

```
RESET
```

**Feedback**

```
RESET
```

## REBOOT

Reboots the unit.

**Syntax**

```
REB00T
```

*This command does not require any parameters.*

**Example**

```
REB00T
```

**Feedback**

```
REB00T
```

## SET TEL_USNM

Sets the user name and password used to log in to Telnet. The feedback reports whether each change succeeded.

> **NOTE:** In this example, the Telnet user name is changed from root to root123 and the Telnet password is changed from admin to admin123.

**Syntax**

```
SET TEL_USNM Oldusnm Newusnm TEL_PWD Oldpwd Newpwd
```

| Parameter | Description | Range |
|---|---|---|
| Oldusnm / Newusnm | Current / new user name | 4 to 16 characters |
| Oldpwd / Newpwd | Current / new password | 4 to 16 characters (alphanumeric only) |
| prm | Result (feedback) | 0 = failed; 1 = success |

**Example**

```
SET TEL_USNM root root123 TEL_PWD admin admin123
```

**Feedback**

```
TEL_USNM 1 TEL_PWD 1
```

## Switch

### SET SW

Routes one input source to one output. SW is short for switch.

**NOTE:** In this example, input 2 is routed to HDMI output 1.

**Syntax**

```
SET SW out in
```

| Parameter | Description | Range |
|---|---|---|
| out | Output | OUT1, OUT2 |
| in | Input | IN1, IN2, IN3, IN4 |

**Example**

```
SET SW OUT1 IN2
```

**Feedback**

```
SW OUT1 IN2
```

## GET SW

Gets which input is routed to the specified output.

**Syntax**

```
GET SW out
```

| Parameter | Description | Range |
|---|---|---|
| out | Output | OUT1, OUT2 |
| in | Input (feedback) | IN1, IN2, IN3, IN4 |

**Example**

```
GET SW OUT2
```

**Feedback**

```
SW OUT2 IN1
```

## SET AUTOSW_FN

Turns the video and USB automatic switching mode on or off.

**Syntax**

```
SET AUTOSW_FN prm
```

| Parameter | Description | Range |
|---|---|---|
| prm | Automatic switching | on, off |

**Example**

```
SET AUTOSW_FN on
```

**Feedback**

```
AUTOSW_FN on
```

## GET AUTOSW_FN

Gets the status of the video and USB automatic switching mode.

**Syntax**

```
GET AUTOSW_FN
```

| Parameter | Description | Range |
|---|---|---|
| prm | Automatic switching (feedback) | on, off |

**Example**

```
GET AUTOSW_FN
```

**Feedback**

```
AUTOSW_FN on
```

## SET AUTOSW_MD

Sets the video automatic switching mode.

**Syntax**

```
SET AUTOSW_MD prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Switching mode | LIFO, PRIORITY |

**Example**

```
SET AUTOSW_MD LIFO
```

**Feedback**

```
AUTOSW_MD LIFO
```

### GET AUTOSW_MD

Gets the video automatic switching mode.

**Syntax**

```
GET AUTOSW_MD
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Switching mode (feedback) | LIFO, PRIORITY |

**Example**

```
GET AUTOSW_MD
```

**Feedback**

```
AUTOSW_MD LIFO
```

## CEC

### SET CEC_PWR

Turns the display (sink) on the specified output on or off using CEC. The command is executed immediately.

**NOTE:** In this example, the display on HDMI output 1 is turned on.

**Syntax**

```
SET CEC_PWR out prm
```

| Parameter | Description | Range |
|---|---|---|
| out | Output | OUT1, OUT2, OUT4, ALL<br>OUT1 = HDMI OUT 1; OUT2 = HDMI OUT 2; OUT4 = HDBaseT OUT 2 |
| prm | Power | on, off |

**Example**

```
SET CEC_PWR OUT1 on
```

**Feedback**

```
CEC_PWR OUT1 on
```

## SET AUTOCEC_FN

Turns the CEC auto power function for the display (sink) on the specified output on or off. Default: on.

**Syntax**

```
SET AUTOCEC_FN out prm
```

| Parameter | Description | Range |
|---|---|---|
| out | Output | OUT1, OUT2, OUT4, ALL<br>OUT1 = HDMI OUT 1; OUT2 = HDMI OUT 2; OUT4 = HDBaseT OUT 2 |
| prm | CEC auto power | on, off |

**Example**

```
SET AUTOCEC_FN OUT1 on
```

**Feedback**

```
AUTOCEC_FN OUT1 on
```

## GET AUTOCEC_FN

Gets the status of the CEC auto power function for the specified output. Default: on.

**Syntax**

```
GET AUTOCEC_FN out
```

| Parameter | Description | Range |
|---|---|---|
| out | Output | OUT1, OUT2, OUT4, ALL<br>OUT1 = HDMI OUT 1; OUT2 = HDMI OUT 2; OUT4 = HDBaseT OUT 2 |
| prm | CEC auto power (feedback) | on, off |

**Example**

```
GET AUTOCEC_FN OUT1
```

**Feedback**

```
AUTOCEC_FN OUT1 on
```

## SET AUTOCEC_D

Sets the CEC auto power delay time for the specified output. AUTOCEC_D is short for CEC auto power delay timing.

**NOTE:** In this example, when there is no active signal on HDMI output 1, the unit turns off the display 30 seconds later.

**Syntax**

```
SET AUTOCEC_D out prm
```

| Parameter | Description | Range |
|---|---|---|
| out | Output | OUT1, OUT2, OUT4, ALL<br>OUT1 = HDMI OUT 1; OUT2 = HDMI OUT 2; OUT4 = HDBaseT OUT 2 |
| prm | Delay time, in seconds | 30, 40, 60 ... (maximum 1800) |

**Example**

```
SET AUTOCEC_D OUT1 30
```

**Feedback**

```
AUTOCEC_D OUT1 30
```

## GET AUTOCEC_D

Gets the CEC auto power delay time for the specified output.

**Syntax**

```
GET AUTOCEC_D out
```

| Parameter | Description | Range |
|---|---|---|
| out | Output | OUT1, OUT2, OUT4, ALL<br>OUT1 = HDMI OUT 1; OUT2 = HDMI OUT 2; OUT4 = HDBaseT OUT 2 |
| prm | Delay time, in seconds (feedback) | 30, 40, 60 ... (maximum 1800) |

**Example**

```
GET AUTOCEC_D OUT1
```

**Feedback**

```
AUTOCEC_D OUT1 30
```

## SET CECCMD_EDIT

Sets the CEC command used for power on or power off on the specified output.

For the command codes, refer to the CEC specification.

**Syntax**

```
SET CECCMD_EDIT out prm1 prm2
```

| Parameter | Description | Range |
|---|---|---|
| out | Output | out1, out2, out4, all<br>out1 = HDMI out 1; out2 = HDMI out 2; out4 = HDBaseT out 2 |
| prm1 | Command type | pwron, pwroff |
| prm2 | CEC command | Hexadecimal bytes separated by spaces (fewer than 16 in total) |

**Example**

```
SET CECCMD_EDIT out1 pwron 40 04
```

**Feedback**

```
CECCMD_EDIT out1 pwron 40 04
```

## GET CECCMD_EDIT

Gets the CEC command used for power on or power off on the specified output.

**Syntax**

```
GET CECCMD_EDIT out prm1
```

| Parameter | Description | Range |
|---|---|---|
| out | Output | out1, out2, out4, all<br>out1 = HDMI out 1; out2 = HDMI out 2; out4 = HDBaseT out 2 |
| prm1 | Command type | pwron, pwroff |
| prm2 | CEC command (feedback) | Hexadecimal bytes separated by spaces (fewer than 16 in total) |

**Example**

```
GET CECCMD_EDIT out1 pwron
```

**Feedback**

```
CECCMD_EDIT out1 pwron 40 04
```

### SET CEC_CMD

Sends a CEC command to the display on the specified output.

For the command codes, refer to the CEC specification.

**Syntax**

```
SET CEC_CMD out prm
```

| Parameter | Description | Range |
|---|---|---|
| out | Output | out1, out2, out4, all<br>out1 = HDMI out 1; out2 = HDMI out 2; out4 = HDBaseT out 2 |
| prm | CEC command | Hexadecimal bytes separated by spaces (fewer than 16 in total) |

**Example**

```
SET CEC_CMD out1 40 04
```

**Feedback**

```
CEC_CMD out1 40 04
```

## RS-232

### SET UART_B

Sets the baud rate of the specified RS-232 port. Default: 115200.

**NOTE:** In this example, the baud rate of RS-232 1 is set to 57600.

**Syntax**

```
SET UART_B prm1 prm2
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | RS-232 port | UART1, UART2, UART3<br>UART1 = RS-232 1; UART2 = RS-232 2; UART3 = RS-232 over HDBaseT |
| prm2 | Baud rate | 115200, 57600, 38400, 19200, 9600, 4800, 2400 |

**Example**

```
SET UART_B UART1 57600
```

**Feedback**

```
UART_B UART1 57600
```

## GET UART_B

Gets the baud rate of the specified RS-232 port. Default: 115200.

**Syntax**

```
GET UART_B prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | RS-232 port | UART1, UART2, UART3<br>UART1 = RS-232 1; UART2 = RS-232 2; UART3 = RS-232 over HDBaseT |
| prm2 | Baud rate (feedback) | 115200, 57600, 38400, 19200, 9600, 4800, 2400 |

**Example**

```
GET UART_B UART1
```

**Feedback**

```
UART_B UART1 57600
```

## SET UARTPWR_FN

Turns the RS-232 auto power control function on or off for the specified port. When on, the stored RS-232 commands are used to turn the display on or off automatically.

**Syntax**

```
SET UARTPWR_FN prm prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm | RS-232 port | UART1, UART2, UART3<br>UART1 = RS-232 1; UART2 = RS-232 2; UART3 = RS-232 over HDBaseT |
| prm1 | Auto power control | on, off |

**Example**

```
SET UARTPWR_FN UART1 on
```

**Feedback**

```
UARTPWR_FN UART1 on
```

## GET UARTPWR_FN

Gets the status of the RS-232 auto power control function for the specified port.

**Syntax**

```
GET UARTPWR_FN prm
```

| Parameter | Description | Range |
|---|---|---|
| prm | RS-232 port | UART1, UART2, UART3<br>UART1 = RS-232 1; UART2 = RS-232 2; UART3 = RS-232 over HDBaseT |
| prm1 | Auto power control (feedback) | on, off |

**Example**

```
GET UARTPWR_FN UART1
```

**Feedback**

```
UARTPWR_FN UART1 on
```

## SET UARTPWR_D

Sets the RS-232 auto power delay time. UARTPWR_D is short for UART power delay timing.

**NOTE:** In this example, when there is no active signal, the unit sends the power-off command through RS-232 1 30 seconds later.

**Syntax**

```
SET UARTPWR_D prm prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm | RS-232 port | UART1 |
| prm1 | Delay time, in seconds | 30 ... 1800 |

**Example**

```
SET UARTPWR_D UART1 30
```

**Feedback**

```
UARTPWR_D UART1 30
```

## GET UARTPWR_D

Gets the RS-232 auto power delay time.

**Syntax**

```
GET UARTPWR_D prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | RS-232 port | UART1 |
| prm2 | Delay time, in seconds (feedback) | 30 ... 1800 |

**Example**

```
GET UARTPWR_D UART1
```

**Feedback**

```
UARTPWR_D UART1 30
```

## SET UART_CMD

Stores an RS-232 command for controlling the display through the specified RS-232 port.

**Syntax**

```
SET UART_CMD prm prm1 prm2 prm3
```

| Parameter | Description | Range |
|---|---|---|
| prm | RS-232 port | UART1, UART2, UART3<br>UART1 = RS-232 1; UART2 = RS-232 2; UART3 = RS-232 over HDBaseT |
| prm1 | Command type | poweron, poweroff, volumemute, volumeunmute, volumeup, volumedown |
| prm2 | Format | hex, str |
| prm3 | Command data | Maximum 48 bytes |

**Example**

```
SET UART_CMD UART1 poweroff hex 30 31 32 33
```

**Feedback**

```
UART_CMD UART1 poweroff hex 30 31 32 33
```

## GET UART_CMD

Gets all RS-232 commands stored for the specified RS-232 port.

**Syntax**

```
GET UART_CMD prm
```

| Parameter | Description | Range |
|---|---|---|
| prm | RS-232 port | UART1, UART2, UART3<br>UART1 = RS-232 1; UART2 = RS-232 2; UART3 = RS-232 over HDBaseT |
| prm1 | Command type (feedback) | poweron, poweroff, volumemute, volumeunmute, volumeinc, volumedec |
| prm2 | Format (feedback) | hex, str |
| prm3 | Command data (feedback) | Stored command data |

**Example**

```
GET UART_CMD UART1
```

**Feedback**

```
UART_CMD UART1 poweron str 123
UART_CMD UART1 poweroff hex 31 32 33
UART_CMD UART1 volumemute hex 31 32 33
UART_CMD UART1 volumeinc hex 31 32 33
UART_CMD UART1 volumedec hex 31 32 33
```

## SET UART_CMD_S

Sends a stored RS-232 command through the specified RS-232 port.

**Syntax**

```
SET UART_CMD_S prm prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm | RS-232 port | UART1, UART2, UART3<br>UART1 = RS-232 1; UART2 = RS-232 2; UART3 = RS-232 over HDBaseT |
| prm1 | Command type | poweron, poweroff, volumemute, volumeunmute, volumeinc, volumedec |

**Example**

```
SET UART_CMD_S UART1 poweron
```

**Feedback**

```
UART_CMD_S UART1 poweron
```

### SET UART_S

Sends data through the specified RS-232 port.

**NOTE:** In this example, the string 123 is sent through RS-232 1.

**Syntax**

```
SET UART_S prm1 prm2 prm3
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | RS-232 port | uart1, uart2, uart3<br>uart1 = RS-232 1; uart2 = RS-232 2; uart3 = RS-232 over HDBaseT |
| prm2 | Format | hex, str |
| prm3 | Data | Maximum 36 bytes |

**Example**

```
SET UART_S uart1 str 123
```

**Feedback**

```
UART_S uart1 str 123
```

## Network

### SET IPADDR STATIC

Sets a static IP address, subnet mask, and gateway for the unit.

**Syntax**

```
SET IPADDR STATIC ip4addr xx.xx.xx.xx netmask xx.xx.xx.xx gateway xx.xx.xx.xx
```

| Parameter | Description | Range |
|---|---|---|
| ip4addr | IP address | xx.xx.xx.xx |
| netmask | Subnet mask | xx.xx.xx.xx |
| gateway | Gateway | xx.xx.xx.xx |

**Example**

```
SET IPADDR STATIC ip4addr 192.168.11.243 netmask 255.255.255.0 gateway 192.168.2.1
```

**Feedback**

```
IPADDR STATIC ip4addr 192.168.11.243 netmask 255.255.255.0 gateway 192.168.2.1
```

## SET IPADDR dhcp

Sets the IP mode of the unit to DHCP.

**Syntax**

```
SET IPADDR dhcp
```

*This command does not require any parameters.*

**Example**

```
SET IPADDR dhcp
```

**Feedback**

```
IPADDR dhcp
```

## GET IPADDR

Gets the IP mode and IP address settings of the unit.

**Syntax**

```
GET IPADDR
```

*This command does not require any parameters.*

**Example**

```
GET IPADDR
```

**Feedback**

```
IPADDR dhcp ip4addr 192.168.11.243 netmask 255.255.255.0 gateway 192.168.2.1
or
IPADDR STATIC ip4addr 192.168.11.243 netmask 255.255.255.0 gateway 192.168.2.1
```

## SET VLAN_ENABLE

Turns VLAN on or off.

**Syntax**

```
SET VLAN_ENABLE prm
```

| Parameter | Description | Range |
|---|---|---|
| prm | VLAN | on, off |

**Example**

```
SET VLAN_ENABLE on
```

**Feedback**

```
VLAN_ENABLE on
```

## GET VLAN_ENABLE

Gets the VLAN status.

**Syntax**

```
GET VLAN_ENABLE
```

| Parameter | Description | Range |
|---|---|---|
| prm | VLAN (feedback) | on, off |

**Example**

```
GET VLAN_ENABLE
```

**Feedback**

```
VLAN_ENABLE on
```

# Video Information

## GET VIDIN_SIG

Gets the signal status of the specified video input.

**Syntax**

```
GET VIDIN_SIG in
```

| Parameter | Description | Range |
|---|---|---|
| in | Input | IN1, IN2, IN3, IN4, all<br>IN1 = HDMI IN 1; IN2 = HDMI IN 2; IN3 = USB-C IN 3 (video); IN4 = USB-C IN 4 (video); all = all video inputs |
| prm | Signal status (feedback) | no, valid |

**Example**

```
GET VIDIN_SIG IN1
```

**Feedback**

```
VIDIN_SIG IN1 valid
```

## SET VIDIN_HDCP_FN

Turns HDCP on or off for the specified video input.

**Syntax**

```
SET VIDIN_HDCP_FN in prm
```

| Parameter | Description | Range |
|---|---|---|
| in | Input | IN1, IN2, IN3, IN4, all<br>IN1 = HDMI IN 1; IN2 = HDMI IN 2; IN3 = USB-C IN 3 (video); IN4 = USB-C IN 4 (video); all = all video inputs |
| prm | HDCP | on, off |

**Example**

```
SET VIDIN_HDCP_FN IN1 ON
```

**Feedback**

```
VIDIN_HDCP_FN IN1 ON
```

## GET VIDIN_HDCP_FN

Gets whether HDCP (HDCP 1.4 / HDCP 2.2) is on or off for the specified video input.

**Syntax**

```
GET VIDIN_HDCP_FN in
```

| Parameter | Description | Range |
|---|---|---|
| in | Input | IN1, IN2, IN3, IN4, all<br>IN1 = HDMI IN 1; IN2 = HDMI IN 2; IN3 = USB-C IN 3 (video); IN4 = USB-C IN 4 (video); all = all video inputs |
| prm | HDCP (feedback) | on, off |

**Example**

```
GET VIDIN_HDCP_FN IN1
```

**Feedback**

```
VIDIN_HDCP_FN IN1 ON
```

## GET VIDIN_HDCP

Gets the HDCP version of the signal on the specified video input.

**Syntax**

```
GET VIDIN_HDCP in
```

| Parameter | Description | Range |
|---|---|---|
| in | Input | IN1, IN2, IN3, IN4, all<br>IN1 = HDMI IN 1; IN2 = HDMI IN 2; IN3 = USB-C IN 3 (video); IN4 = USB-C IN 4 (video); all = all video inputs |
| prm | HDCP version (feedback) | no hdcp, hdcp1.4, hdcp2.2, follow |

**Example**

```
GET VIDIN_HDCP IN1
```

**Feedback**

```
VIDIN_HDCP IN1 no hdcp
```

## GET SCALER

Gets the scaler status of the specified output.

**Syntax**

```
GET SCALER out
```

| Parameter | Description | Range |
|---|---|---|
| out | Output | OUT1, OUT2, OUT4, all<br>all = all video outputs |
| prm | Scaler (feedback) | on, off, force |

**Example**

```
GET SCALER out1
```

**Feedback**

```
SCALER out1 off
```

## SET SCALER

Sets the scaler status of the specified output.

**Syntax**

```
SET SCALER out prm
```

| Parameter | Description | Range |
|---|---|---|
| out | Output | OUT1, OUT2, OUT4, all<br>all = all video outputs |
| prm | Scaler | on, off, force |

**Example**

```
SET SCALER out1 off
```

**Feedback**

```
SCALER out1 off
```

# HDCP Information

## GET HDCP

Gets the HDCP mode of the specified output. off = output without HDCP; follow = the output follows the HDCP version of the display (for example, if the display supports HDCP 1.4, the output also uses HDCP 1.4; the same applies to HDCP 2.2 and off). Default: follow.

**Syntax**

```
GET HDCP out
```

| Parameter | Description | Range |
|---|---|---|
| out | Output | OUT1, OUT2, ALL |
| prm | HDCP mode (feedback) | follow, hdcp1.4, off ... |

**Example**

```
GET HDCP OUT1
```

**Feedback**

```
HDCP OUT1 follow
```

# Web Login

## SET WEB_USNM

Sets the user name and password used to log in to the Web UI. The feedback reports the result of each change.

**Syntax**

```
SET WEB_USNM Oldusnm Newusnm WEB_PWD Oldpassword Newpassword
```

| Parameter | Description | Range |
|---|---|---|
| Oldusnm / Newusnm | Current / new user name | 4 to 16 characters |
| Oldpassword / Newpassword | Current / new password | 4 to 16 characters (alphanumeric only) |
| prm1 | User name result (feedback) | 0 = current user name or password error; 1 = changed successfully |
| prm2 | Password result (feedback) | 0 = current user name or password error; 1 = changed successfully; 2 = cannot change to the default password |

**Example**

```
SET WEB_USNM admin admin123 WEB_PWD admin admin123
```

**Feedback**

```
WEB_USNM 1 WEB_PWD 1
```

# USB Matrix

## SET USB_M

Sets the USB switching mode. FOLLOW1 = USB follows OUT1; FOLLOW2 = USB follows OUT2; INDEPENDENT = USB is switched independently.

**Syntax**

```
SET USB_M prm
```

| Parameter | Description | Range |
|---|---|---|
| prm | USB switching mode | FOLLOW1, FOLLOW2, INDEPENDENT |

**Example**

```
SET USB_M INDEPENDENT
```

**Feedback**

```
USB_M INDEPENDENT
```

## GET USB_M

Gets the USB switching mode.

**NOTE:** In this example, USB follows OUT1.

**Syntax**

```
GET USB_M
```

| Parameter | Description | Range |
|---|---|---|
| prm | USB switching mode (feedback) | FOLLOW1, FOLLOW2, INDEPENDENT |

**Example**

```
GET USB_M
```

**Feedback**

```
USB_M FOLLOW1
```

## SET USBSW

Routes one USB input source to all USB outputs. USBSW is short for USB switch.

**Syntax**

```
SET USBSW in
```

| Parameter | Description | Range |
|---|---|---|
| in | USB input | IN1, IN2, IN3, IN4 |

**Example**

```
SET USBSW IN1
```

**Feedback**

```
USBSW IN1
```

## GET USBSW

Gets which USB input is routed to the USB output.

**Syntax**

```
GET USBSW
```

| Parameter | Description | Range |
|---|---|---|
| in | USB input (feedback) | IN1, IN2, IN3, IN4 |

**Example**

```
GET USBSW
```

**Feedback**

```
USBSW i02
```

## SET USBNIC

Turns the USB-to-network function of the specified input on or off.

**Syntax**

```
SET USBNIC in prm
```

| Parameter | Description | Range |
|---|---|---|
| in | Input | IN1, IN2, IN3, IN4, ALL |
| prm | USB to network | on, off |

**Example**

```
SET USBNIC IN1 on
```

**Feedback**

```
USBNIC IN1 on
```

## GET USBNIC

Gets the USB-to-network status of the specified input.

**Syntax**

```
GET USBNIC in
```

| Parameter | Description | Range |
|---|---|---|
| in | Input | IN1, IN2, IN3, IN4, ALL |
| prm | USB to network (feedback) | on, off |

**Example**

```
GET USBNIC IN1
```

**Feedback**

```
USBNIC IN1 on
```

# Audio

## SET AUD_MUTE

Mutes or unmutes the specified audio output.

**Syntax**

```
SET AUD_MUTE prm1 prm2
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Audio output | HDMIOUT1, HDMIOUT2, HDBTOUT2, DANTEOUT, ANALOGOUT, USBHOSTAUDIO, ALL |
| prm2 | Mute | ON, OFF |

**Example**

```
SET AUD_MUTE HDMIOUT1 ON
```

**Feedback**

```
AUD_MUTE HDMIOUT1 ON
```

## GET AUD_MUTE

Gets the mute status of the specified audio output.

**Syntax**

```
GET AUD_MUTE prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Audio output | HDMIOUT1, HDMIOUT2, HDBTOUT2, DANTEOUT, ANALOGOUT, USBHOSTAUDIO, ALL |
| prm2 | Mute (feedback) | ON, OFF |

**Example**

```
GET AUD_MUTE HDMIOUT1
```

**Feedback**

```
AUD_MUTE HDMIOUT1 ON
```

## SET AUDIOSW

Routes one audio input source to one audio output. AUDIOSW is short for audio switch.

**Syntax**

```
SET AUDIOSW in out
```

| Parameter | Description | Range |
|---|---|---|
| in | Audio input | in1, in2, in3, in4<br>in1 = USB HOST; in2 = AUDIO DE-EMBED (OUT1); in3 = AUDIO DE-EMBED (OUT2); in4 = DANTE IN |
| out | Audio output | out1, all |

**Example**

```
SET AUDIOSW in1 out1
```

**Feedback**

```
AUDIOSW in1 out1
```

## GET AUDIOMP

Gets which audio input is routed to the specified audio output.

**Syntax**

```
GET AUDIOMP out
```

| Parameter | Description | Range |
|---|---|---|
| out | Audio output | out1, all |
| in | Audio input (feedback) | in1, in2, in3, in4<br>in1 = USB HOST; in2 = AUDIO DE-EMBED (OUT1); in3 = AUDIO DE-EMBED (OUT2); in4 = DANTE IN |

**Example**

```
GET AUDIOMP out1
```

**Feedback**

```
AUDIOMP in1 out1
```

## SET DANTE_IN

Sets whether Dante input audio is sent to the embedded video output audio or to the USB host audio.

**Syntax**

```
SET DANTE_IN prm1 prm2
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Destination | EMBED_VIDEO_OUT, USB_HOST_AUDIO, all |
| prm2 | State | on, off |

**Example**

```
SET DANTE_IN EMBED_VIDEO_OUT on
```

**Feedback**

```
DANTE_IN EMBED_VIDEO_OUT on
```

## GET DANTE_IN

Gets whether Dante input audio is sent to the embedded video output audio or to the USB host audio.

**Syntax**

```
GET DANTE_IN prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Destination | EMBED_VIDEO_OUT, USB_HOST_AUDIO, all |
| prm2 | State (feedback) | on, off |

**Example**

```
GET DANTE_IN EMBED_VIDEO_OUT
```

**Feedback**

```
DANTE_IN EMBED_VIDEO_OUT on
```

# Priority Control

## SET VIDEO_P

Sets the priority switching level of a video input for the specified output.

**Syntax**

```
SET VIDEO_P prm1 prm2 prm3
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Output | OUT1, OUT2 |
| prm2 | Input | IN1, IN2, IN3, IN4, ALL |
| prm3 | Priority level | 0, 1, 2, 3, 4 |

**Example**

```
SET VIDEO_P OUT1 IN1 1
```

**Feedback**

```
VIDEO_P OUT1 IN1 1
```

## GET VIDEO_P

Gets the priority switching level of a video input for the specified output.

**Syntax**

```
GET VIDEO_P prm1 prm2
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Output | OUT1, OUT2 |
| prm2 | Input | IN1, IN2, IN3, IN4, ALL |
| prm3 | Priority level (feedback) | 0, 1, 2, 3, 4 |

**Example**

```
GET VIDEO_P OUT1 IN1
```

**Feedback**

```
VIDEO_P OUT1 IN1 1
```

## SET USB_P

Sets the priority switching level of a USB input.

**Syntax**

```
SET USB_P prm1 prm2
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | USB input | IN1, IN2, IN3, IN4, ALL |
| prm2 | Priority level | 0, 1, 2, 3, 4 |

**Example**

```
SET USB_P IN1 1
```

**Feedback**

```
USB_P IN1 1
```

## GET USB_P

Gets the priority switching level of a USB input.

**Syntax**

```
GET USB_P prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | USB input | IN1, IN2, IN3, IN4, ALL |
| prm2 | Priority level (feedback) | 0, 1, 2, 3, 4 |

**Example**

```
GET USB_P IN1
```

**Feedback**

```
USB_P IN1 1
```

# EDID

## SET EDID

Sets the EDID mode of the specified input. The setting is kept through a reboot. Default for all inputs: 1 (Fixed 4K@60Hz 2.0ch).

**Syntax**

```
SET EDID prm1 prm2
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Input | IN1, IN2, IN3, IN4, IN5, ALL |
| prm2 | EDID mode | 1 = Fixed 4K@60Hz 2.0ch (default)<br>2 = Copy from HDMI OUT 1<br>3 = Copy from HDMI OUT 2<br>4 = Copy from HDBT OUT 2<br>5 = Fixed 4K@60Hz 7.1ch<br>6 = Fixed 4K@30Hz 2.0ch<br>7 = Fixed 4K@30Hz 7.1ch<br>8 = 1080P@60 2.0ch<br>9 = 1080P@60 5.1ch<br>10 = 1080P@60 7.1ch<br>11 = Custom EDID 1<br>12 = Custom EDID 2 |

**Example**

```
SET EDID IN1 1
```

**Feedback**

```
EDID IN1 1
```

## GET EDID

Gets the EDID mode of the specified input.

**Syntax**

```
GET EDID prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Input | IN1, IN2, IN3, IN4, IN5, ALL |
| prm2 | EDID mode (feedback) | 1 = Fixed 4K@60Hz 2.0ch (default)<br>2 = Copy from HDMI OUT 1<br>3 = Copy from HDMI OUT 2<br>4 = Copy from HDBT OUT 2<br>5 = Fixed 4K@60Hz 7.1ch<br>6 = Fixed 4K@30Hz 2.0ch<br>7 = Fixed 4K@30Hz 7.1ch<br>8 = 1080P@60 2.0ch<br>9 = 1080P@60 5.1ch<br>10 = 1080P@60 7.1ch<br>11 = Custom EDID 1<br>12 = Custom EDID 2 |

**Example**

```
GET EDID IN1
```

**Feedback**

```
EDID IN1 1
```

## SET EDID_W

Writes a custom EDID, one 128-byte block at a time. The setting is kept through a reboot. Factory default: all zeros.

**Syntax**

```
SET EDID_W prm1 prm2 prm3
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Custom EDID slot | cedid1, cedid2 |
| prm2 | EDID block | block0, block1 |
| prm3 | EDID data | 128 bytes |

**Example**

```
SET EDID_W cedid1 block0 {xxxxxxxx}
```

**Feedback**

```
EDID_W cedid1 block0 success
```

## GET EDID_W

Reads the specified custom EDID.

**Syntax**

```
GET EDID_W prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Custom EDID slot | cedid1, cedid2 |
| prm2 | EDID block (feedback) | block0, block1 |
| prm3 | EDID data (feedback) | 128 bytes |

**Example**

```
GET EDID_W cedid1
```

**Feedback**

```
EDID_W cedid1 block0 {128bytes}
EDID_W cedid1 block1 {128bytes}
```

### GET EDID_R

Reads the EDID content from the display connected to the specified output.

**Syntax**

```
GET EDID_R prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Output | OUT1, OUT2, OUT4<br>OUT1 = HDMI OUT 1; OUT2 = HDMI OUT 2; OUT4 = HDBaseT OUT 2 |
| prm2 | EDID block (feedback) | block0, block1 |
| prm3 | EDID data or status (feedback) | One 128-byte block of EDID ASCII data with no spaces; error (checksum error); unconnected (display not connected) |

**Example**

```
GET EDID_R out1
```

**Feedback**

```
EDID_R OUT1 block0 {xxxxxx}
EDID_R OUT1 block1 {xxxxxx}
or
EDID_R OUT1 error
or
EDID_R OUT1 unconnect
```

## USB-C Settings

### SET USBC4_DM

Sets the DisplayPort mode of USB-C IN 4. The setting is kept through a reboot. Default: SST.

**Syntax**

```
SET USBC4_DM prm
```

| Parameter | Description | Range |
|---|---|---|
| prm | DisplayPort mode | SST, MST |

**Example**

```
SET USBC4_DM SST
```

**Feedback**

```
USBC4_DM SST
```

## GET USBC4_DM

Gets the DisplayPort mode of USB-C IN 4.

**Syntax**

```
GET USBC4_DM
```

| Parameter | Description | Range |
|---|---|---|
| prm | DisplayPort mode (feedback) | SST, MST |

**Example**

```
GET USBC4_DM
```

**Feedback**

```
USBC4_M MST
```

## SET USBC4_STRA

Sets how USB-C IN 3 and HDMI 2 are configured when USB-C IN 4 is in MST mode. The setting is kept through a reboot. Default: MODE1 (one USB-C + two HDMI).

**Syntax**

```
SET USBC4_STRA prm
```

| Parameter | Description | Range |
|---|---|---|
| prm | Strategy | MODE1 = one USB-C + two HDMI; MODE2 = two USB-C + one HDMI |

**Example**

```
SET USBC4_STRA MODE1
```

**Feedback**

```
USBC4_STRA MODE1
```

### GET USBC4_STRA

Gets how USB-C IN 3 and HDMI 2 are configured when USB-C IN 4 is in MST mode.

**Syntax**

```
GET USBC4_STRA
```

| Parameter | Description | Range |
|---|---|---|
| prm | Strategy (feedback) | MODE1 = one USB-C + two HDMI; MODE2 = two USB-C + one HDMI |

**Example**

```
GET USBC4_STRA
```

**Feedback**

```
USBC4_STRA MODE1
```

## GPIO & Relay

### GET GPIO_TYPE

Gets whether the specified GPIO is set as an input or an output.

**Syntax**

```
GET GPIO_TYPE prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | GPIO | 1, 2, 3, 4, 5, 6 |
| prm2 | Type (feedback) | IN, OUT |

**Example**

```
GET GPIO_TYPE 1
```

**Feedback**

```
GPIO_TYPE 1 IN
```

## SET GPIO_TYPE

Sets the specified GPIO as an input or an output.

**Syntax**

```
SET GPIO_TYPE prm1 prm2
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | GPIO | 1, 2, 3, 4 |
| prm2 | Type | IN, OUT |

**Example**

```
SET GPIO_TYPE 1 IN
```

**Feedback**

```
GPIO_TYPE 1 IN
```

## SET GPIO_IN_CTL

Sets the configuration of the specified GPIO input.

**Syntax**

```
SET GPIO_IN_CTL prm1 prm2
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | GPIO | 1, 2, 3, 4 |
| prm2 | Level | high, low |

**Example**

```
SET GPIO_IN_CTL 1 HIGH
```

**Feedback**

```
GPIO_IN_CTL 1 HIGH
```

## GET GPIO_IN_CTL

Gets the configuration of the specified GPIO input.

**Syntax**

```
GET GPIO_IN_CTL prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | GPIO | 1, 2, 3, 4 |
| prm2 | Level (feedback) | high, low |

**Example**

```
GET GPIO_IN_CTL 1
```

**Feedback**

```
GPIO_IN_CTL 1 HIGH
```

## SET GPIO_OUT_CTL

Sets the configuration of the specified GPIO output.

**Syntax**

```
SET GPIO_OUT_CTL prm1 prm2 prm3
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | GPIO | 1, 2, 3, 4, 5, 6 |
| prm2 | Output type | push-pull, open-drain |
| prm3 | Level | high, low |

**Example**

```
SET GPIO_OUT_CTL 1 PUSH-PULL HIGH
```

**Feedback**

```
GPIO_OUT_CTL 1 PUSH-PULL HIGH
```

## GET GPIO_OUT_CTL

Gets the configuration of the specified GPIO output.

**Syntax**

```
GET GPIO_OUT_CTL prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | GPIO | 1, 2, 3, 4, 5, 6 |
| prm2 | Output type (feedback) | push-pull, open-drain |
| prm3 | Level (feedback) | high, low |

**Example**

```
GET GPIO_OUT_CTL 1
```

**Feedback**

```
GPIO_OUT_CTL 1 PUSH-PULL HIGH
```

## SET RELAY_STATUS

Sets the output status of the specified relay.

**Syntax**

```
SET RELAY_STATUS prm1 prm2
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Relay | 1, 2, all |
| prm2 | Status | NC, NO |

**Example**

```
SET RELAY_STATUS 1 NC
```

**Feedback**

```
RELAY_STATUS 1 NC
```

## GET RELAY_STATUS

Gets the output status of the specified relay.

**Syntax**

```
GET RELAY_STATUS prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Relay | 1, 2, all |
| prm2 | Status (feedback) | NC, NO |

**Example**

```
GET RELAY_STATUS 1
```

**Feedback**

```
RELAY_STATUS 1 NC
```

## SET AUTORELAY_FN

Turns the relay auto power function on or off for the specified relay.

**Syntax**

```
SET AUTORELAY_FN prm1 prm2
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Relay | 1, 2, all |
| prm2 | Auto power | on, off |

**Example**

```
SET AUTORELAY_FN 1 on
```

**Feedback**

```
AUTORELAY_FN 1 on
```

## GET AUTORELAY_FN

Gets the status of the relay auto power function for the specified relay.

**Syntax**

```
GET AUTORELAY_FN prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Relay | 1, 2, all |
| prm2 | Auto power (feedback) | on, off |

**Example**

```
GET AUTORELAY_FN 1
```

**Feedback**

```
AUTORELAY_FN 1 on
```

## SET RELAY_MODE

Sets the mode of the specified relay.

**Syntax**

```
SET RELAY_MODE prm1 prm2
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Relay | 1, 2, all |
| prm2 | Mode | latch, momentary |

**Example**

```
SET RELAY_MODE 1 Latch
```

**Feedback**

```
RELAY_MODE 1 Latch
```

## GET RELAY_MODE

Gets the mode of the specified relay.

**Syntax**

```
GET RELAY_MODE prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Relay | 1, 2, all |
| prm2 | Mode (feedback) | latch, momentary |

**Example**

```
GET RELAY_MODE 1
```

**Feedback**

```
RELAY_MODE 1 Latch
```

## SET RELAY_PULSE_TIME

Sets the trigger (pulse) time of the specified relay in momentary mode.

**NOTE:** In this example, the momentary-mode trigger time of relay 1 is set to 3 seconds.

**Syntax**

```
SET RELAY_PULSE_TIME prm1 prm2
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Relay | 1, 2, all |
| prm2 | Trigger time, in seconds | 1 ... 10 |

**Example**

```
SET RELAY_PULSE_TIME 1 3
```

**Feedback**

```
RELAY_PULSE_TIME 1 3
```

## GET RELAY_PULSE_TIME

Gets the trigger (pulse) time of the specified relay in momentary mode.

**Syntax**

```
GET RELAY_PULSE_TIME prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | Relay | 1, 2, all |
| prm2 | Trigger time, in seconds (feedback) | 1 ... 10 |

**Example**

```
GET RELAY_PULSE_TIME 1
```

**Feedback**

```
RELAY_PULSE_TIME 1 3
```

## SET RELAY_TRIGGER_MODE

Sets the relay trigger conditions.

mode1: When the automatic mode is triggered to power on, relay 1 is set to connect NC and COM; when the automatic mode is triggered to power off, relay 2 is set to connect NC and COM.

mode2: When the automatic mode is triggered to power on, relay 2 is set to connect NC and COM; when the automatic mode is triggered to power off, relay 1 is set to connect NC and COM.

mode3: When the automatic mode is triggered to power on, relay 1 and relay 2 are simultaneously set to connect NC and COM; when the automatic mode is triggered to power off, relay 1 and relay 2 are simultaneously set to connect NC and COM.

**Syntax**

```
SET RELAY_TRIGGER_MODE prm
```

| Parameter | Description | Range |
|---|---|---|
| prm | Trigger mode | mode1, mode2, mode3 |

**Example**

```
SET RELAY_TRIGGER_MODE mode1
```

**Feedback**

```
RELAY_TRIGGER_MODE mode1
```

## GET RELAY_TRIGGER_MODE

Gets the relay trigger conditions. See SET RELAY_TRIGGER_MODE for the modes.

**Syntax**

```
GET RELAY_TRIGGER_MODE
```

| Parameter | Description | Range |
|---|---|---|
| prm | Trigger mode (feedback) | mode1, mode2, mode3 |

**Example**

```
GET RELAY_TRIGGER_MODE
```

**Feedback**

```
RELAY_TRIGGER_MODE mode1
```

## SET SENSOR 24V

Turns the 24V output of the SENSOR port on or off.

**Syntax**

```
SET SENSOR 24V prm1
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | 24V output | ON, OFF |

**Example**

```
SET SENSOR 24V on
```

**Feedback**

```
SENSOR 24V on
```

## GET SENSOR 24V

Gets the status of the 24V output of the SENSOR port.

**Syntax**

```
GET SENSOR 24V
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | 24V output (feedback) | ON, OFF |

**Example**

```
GET SENSOR 24V
```

**Feedback**

```
SENSOR 24V on
```

## GET SENSOR IO

Gets the I/O status of the SENSOR port.

**Syntax**

```
GET SENSOR IO
```

| Parameter | Description | Range |
|---|---|---|
| prm1 | I/O status (feedback) | high, low |

**Example**

```
GET SENSOR IO
```

**Feedback**

```
SENSOR IO LOW
```

## SET USB_VBUS_MODE

Sets the VBUS working mode of the specified USB device channel.

**Syntax**

```
SET USB_VBUS_MODE usb_channel mode
```

| Parameter | Description | Range |
|---|---|---|
| usb_channel | USB device channel | usb1, usb2, usb3, usb4 ..., all |
| mode | VBUS working mode | follow_host, always_on, always_off |

**Example**

```
SET USB_VBUS_MODE usb1 always_on
```

**Feedback**

```
USB_VBUS_MODE usb1 always_on
```

## GET USB_VBUS_MODE

Gets the VBUS working mode of the specified USB device channel.

**Syntax**

```
GET USB_VBUS_MODE usb_channel
```

| Parameter | Description | Range |
|---|---|---|
| usb_channel | USB device channel | usb1, usb2, usb3, usb4 ..., all |
| mode | VBUS working mode (feedback) | follow_host, always_on, always_off |

**Example**

```
GET USB_VBUS_MODE usb1
```

**Feedback**

```
USB_VBUS_MODE usb1 always_on
```