



MX-HYB/EDC API

MX-1007-HYB | MX-0804-EDC

Application Programming Interface

Document Revision	v1.01
Document Date	Feb 2024
Supported Firmware	Refer to Supported Product Firmware/Software for details.

Contents

MX-HYB/EDC API	1
1. Overview	3
1.1 Supported Product Firmware/Software.....	3
1.2 Before You Begin.....	3
2. Wiring and Communication Configuration.....	4
2.1 RS-232 Connections	4
2.2 Network Connections.....	5
2.2.1 IP Settings.....	5
3. Command Overview.....	5
3.1 Command Delimiter for Sent Commands.....	5
4. Controlling Matrix Switching.....	7
4.1 Controlling Video	7
4.2 Controlling Audio.....	8
5. USB Switching.....	26
6. GPIO and RELAY Control	27
7. Saving and Recalling an Audio/Video Scene	29
8. Controlling Display Power via CEC	30
7 Matrix EDID Settings.....	32
8 Scaling Configuration	36
8.1 Configuring the scaling outputs	36
9 Troubleshooting.....	40
10 Contacting Technical Support.....	41
11 Document Revision History.....	41
Publication Disclaimer.....	41
WyreStorm Offices.....	41

1. Overview

The following contains the connection and commands to control the matrix switchers. By following the content contained in this document, the switcher can be controlled and configured via a 3rd party RS-232 control system.

IMPORTANT NOTE!

Due to differences between matrix series and model versions within a series, some commands have different parameters based on the model and version. These differences are noted where applicable and should be followed as sending an incorrect parameter may cause the unit to lock up and become inoperative.

1.1 Supported Product Firmware/Software

The following products and firmware versions are supported by this version of the API. The firmware versions listed are the minimum supported at time of publication; firmware may be higher except where otherwise noted.

Product	Status Since Last Doc Rev	Supported Product Versions
MX-1007-HYB	added	v1 or higher
MX-0804-EDC	added	v1 or higher

1.2 Before You Begin

Verify that the following items are on hand and that all documentation is reviewed before continuing:

- Configured and Operational HYB|EDC Matrix
- Refer to **Supported Product Firmware/Software** for a complete list of supported products and versions. Control System and Control System
- Documentation.....
- PC or Mac for configuring product and telnet communications
- Network Connection with Network Passwords.....
- Current Product Firmware (if available), Software, and Documentation downloaded from **WyreStorm.com**.....

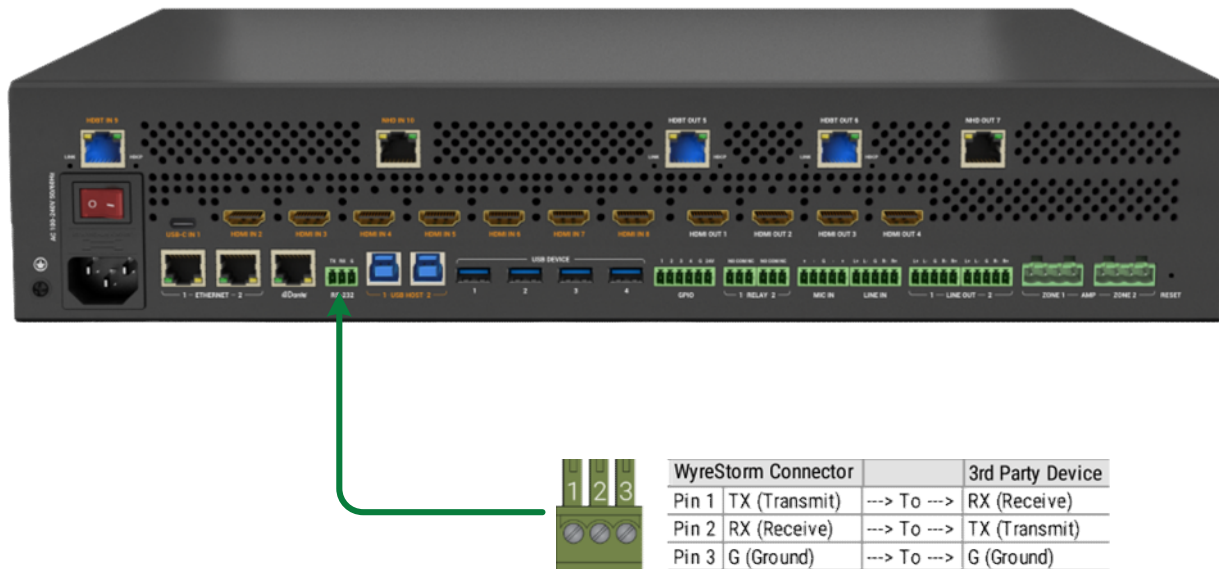
2. Wiring and Communication Configuration

WyreStorm recommends that all wiring for the installation is run and terminated prior to making connections to the switcher. Read through this section in its entirety before running or terminating the wires to ensure proper operation and to avoid damaging equipment.

2.1 RS-232 Connections

The following wiring diagrams show the pinouts for the WyreStorm device. While not shown, connect the TX (transmit) to RX (receive) pins at the control system or PC side of the cable. Most control systems and computers are configured for Digital Terminal Equipment (DTE) where pin 2 is RX and pin 3 is TX. This can vary from device to device, refer to the documentation for the connected device for pin functionality to ensure that the correct connections can be made.

Note: MX-I007-HYB shown below. Port may be in a different location for the various models.



RS-232 Port Settings

Baud rate:	9600bps
Data Bits:	8bits
Parity:	None
Stop Bits:	1 bit
Flow Control:	None

2.2 Network Connections

2.2.1 IP Settings

Default IP Address	DHCP by default Auto IP method if DHCP is not present. Refer to IP Addressing and Web UI Access
Default IP Port	23

IP Addressing and Web UI Access

These matrix switchers use an Auto IP method to generate the initial IP address based on the network connections. By default, the IP address is set to DHCP and will pull the IP address from a connected DHCP server. Should the network not contain a DHCP server, the IP address will be generated based on the unit's MAC address. The above operation will occur unless the IP Address setting in the web UI is set to static.

1. Connect the matrix to the same network as a PC.
2. Using a 3rd party network scanner, scan the network for the IP address of the matrix.
3. Open a web browser and enter the IP Address of the matrix.
4. Enter the password for the matrix. The default password is: admin.

IP Address Notes

- The installer password and general password are the same by default. WyreStorm recommends changing the password for installer login to avoid any unwanted changes being made to the matrix configuration.

3. Command Overview

3.1 Command Delimiter for Sent Commands

When sending commands using the IPv4 / Telnet API channel, or when using the RS-232 API channel, all command lines sent from the 3rd party controller to the matrix should end with a specific character. This signifies when the command is processed by the matrix. This is usually specified in 3rd party control software as the "command delimiter," "stop character," or "line terminator."

Accepted delimiter characters are:

Character	Shorthand	Hex Notation	Escape Notation	Decimal Notation
Line Feed	LF	0A	\n	10
Carriage Return + Line Feed	CR LF	0D 0A	\r\n	13 10

Please note, most 3rd party control software will either append these characters automatically or an option to specify them will be present.

Note: It is important that the last delimiter character is LF and not CR.

4. Controlling Matrix Switching

4.1 Controlling Video

Switching Video Outputs	
Command structure: SET SW <INPUT> <OUTPUT>	HYB Series <INPUT> = in1 ~ in10 , in0 <OUTPUT> = out1~out7 , all
Response Syntax: SW <INPUT> <OUTPUT>	
Example Command: SET SW in4 out1	EDC Series <INPUT> = in1 ~ in8 , in0 <OUTPUT> = out1~out4 , all Note: <INPUT> value of “in0” power downs the output.
Example Response: SW in4 out1	
Query Video Output Mapping	
Command structure: GET MP <OUTPUT>	HYB Series <INPUT> = in1 ~ in10 , in0 <OUTPUT> = out1~out7 , all
Response Syntax: MP GET <INPUT> <OUTPUT>	
Example Command: GET MP out1	EDC Series <INPUT> = in1 ~ in8 , in0 <OUTPUT> = out1~out4 , all
Example Response: MP in4 out1	

4.2 Controlling Audio

Select Audio Mixer Input	
<p>Command structure: SET MIXER_IN <PRM1> <PRM3> <PRM2></p> <p>Response Syntax: MIXER_IN <PRM1> <PRM3> <PRM2></p>	<p>HYB Series</p> <p><PRM1> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER. <PRM2> = On, Off <PRM3> = OUT1 ~ OUT7, Dantein1 ~ Dantein4, usbhost, micin1 ~ micin2, linein</p> <p>Note: Analog input: micin1, micin2, linein Digital input: Dantein1 ~ Dantein4, usbhost (usbhost does not work with Remote USB MIXER) Video De-embed: OUT1 ~ OUT7</p>
<p>Example Command: SET MIXER_IN 1 OUT1 ON</p>	<p>EDCSeries</p> <p><PRM1> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER. <PRM2> = On, Off <PRM3> = OUT1 ~ OUT4, Dantein1 ~ Dantein4, usbhost, micin1 ~ micin2, linein</p> <p>Note: Analog input: micin1, micin2, linein Digital input: Dantein1 ~ Dantein4, usbhost (usbhost does not work with Remote USB MIXER) Video De-embed: OUT1 ~ OUT4</p>

Query Audio Mixer Input

<p>Command structure: GET MIXER_IN <PRM1></p> <p>Response Syntax: MIXER_IN <PRM1> <PRM2> <PRM3></p>	<p>HYB Series</p> <p><PRM1> = 1, 2, 3</p> <p>1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p> <p><PRM2> = On, Off</p> <p><PRM3> = {OUT1 ~ OUT7, Dantein1 ~ Dantein4, usbhost, micin1 ~ micin2, linein}</p> <p>Note: Analog input: micin1, micin2, linein Digital input: Dantein1 ~ Dantein4, usbhost (usbhost does not work with Remote USB MIXER) Video De-embed: OUT1 ~ OUT7</p>
<p>Example Command: GET MIXER_IN 1</p> <p>Example Response: MIXER_IN 1 On Dantein1</p>	<p>EDCSeries</p> <p><PRM1> = 1, 2, 3</p> <p>1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p> <p><PRM2> = On, Off</p> <p><PRM3> = OUT1 ~ OUT4, Dantein1 ~ Dantein4, usbhost, micin1 ~ micin2, linein</p> <p>Note: Analog input: micin1, micin2, linein Digital input: Dantein1 ~ Dantein4, usbhost (usbhost does not work with Remote USB MIXER) Video De-embed: OUT1 ~ OUT4</p>

Set Audio Mixer Output

Command structure:

SET MIXER_OUT <PRM1> <PRM2> <PRM3>

Response Syntax:

MIXER_OUT <PRM1> <PRM2> <PRM3>

HYB Series

<PRM1> = 1, 2

1: indicates Local audio Mix 1.

2: indicates Local audio Mix 2.

<PRM2> = LINEOUT1, LINEOUT2, AMPOUT, DANTEOUT12, DANTEOUT34, NHDOUT

Available only in Local audio mixer1: LINEOUT1, AMPOUT, DANTEOUT12, NHDOUT

Available only in Local audio mixer2: LINEOUT2, DANTEOUT34

<PRM3> = On, Off

Example Command:

SET MIXER_OUT 1 LINEOUT1 ON

EDC Series

<PRM1> = 1, 2

1: indicates Local audio Mix 1.

2: indicates Local audio Mix 2.

<PRM2> = LINEOUT1, LINEOUT2, AMPOUT, DANTEOUT12, DANTEOUT34, NHDOUT

Available only in Local audio mixer1: LINEOUT1, AMPOUT, DANTEOUT12, NHDOUT

Available only in Local audio mixer2: LINEOUT2, DANTEOUT34

<PRM3> = On, Off

Example Response:

MIXER_OUT 1 LINEOUT1 ON

Get Audio Mixer Output

Command structure:

```
GET MIXER_OUT <PRM1> <PRM2> <PRM3>
```

Response Syntax:

```
MIXER_OUT <PRM1> <PRM2> <PRM3>
```

HYB Series

<PRM1> = 1, 2

1: indicates Local audio Mix 1.

2: indicates Local audio Mix 2.

<PRM2> = LINEOUT1, LINEOUT2, AMPOUT, DANTEOUT12, DANTEOUT34, NHDOUT

Available only in Local audio mixer1: LINEOUT1, AMPOUT, DANTEOUT12, NHDOUT

Available only in Local audio mixer2 :LINEOUT2, DANTEOUT34

<PRM3> = On, Off

Example Command:

```
GET MIXER_OUT 1 LINEOUT1 ON
```

EDC Series

<PRM1> = 1, 2

1: indicates Local audio Mix 1.

2: indicates Local audio Mix 2.

<PRM2> = LINEOUT1, LINEOUT2, AMPOUT, DANTEOUT12, DANTEOUT34, NHDOUT

Available only in Local audio mixer1:LINEOUT1, AMPOUT, DANTEOUT12,NHDOUT

Available only in Local audio mixer2:LINEOUT2, DANTEOUT34

<PRM3> = On, Off

Example Response:

```
MIXER_OUT 1 LINEOUT1 ON
```

Set Audio Mixer Input Volume

Command structure:

```
SET MIXER_IN_VOL <PRM1> <PRM2> <PRM3>
```

Response Syntax:

```
MIXER_IN_VOL <PRM1> <PRM2> <PRM3>
```

HYB Series

<PRM1> = 1, 2, 3

1: indicates Local audio Mix 1.

2: indicates Local audio Mix 2.

3: indicates Remote USB MIXER.

<PRM2> = OUT1 ~ OUT7, Dantein1 ~ Dantein4, usbhost, micin1 ~ micin2, linein

Anlog input: micin1, micin2, linein

Digital input: Dantein1 ~ Dantein4, usbhost

(usbhost does not work with Remote USB MIXER)

Video De-embed: OUT1 ~ OUT7

<PRM3> = {[-20, 20], [-20, 40]}

When in is LINEIN, DANTEin1~2, OUT1~OUT7, usbhost, the prm = [-20, 20]

When in is MICIN, prm = [-20, 40]

Example Command:

```
SET MIXER_IN_VOL 1 MICIN1 -20
```

Example Response:

```
MIXER_IN_VOL 1 MICIN1 -20
```

EDC Series

<PRM1> = 1, 2, 3

1: indicates Local audio Mix 1.

2: indicates Local audio Mix 2.

3: indicates Remote USB MIXER.

<PRM2> = OUT1 ~ OUT4, Dantein1 ~ Dantein4, usbhost, micin1 ~ micin2, linein}

Anlog input: micin1, micin2, linein

Digital input: Dantein1 ~ Dantein4, usbhost (usbhost does not work with Remote USB MIXER)

Video De-embed: OUT1 ~ OUT4

<PRM3> = {[-20, 20], [-20, 40]}

When in is LINEIN, DANTEin1~2, OUT1~OUT4, usbhost, the prm = [-20, 20]

When in is MICIN, prm = [-20, 40]

Get Audio Mixer Input Volume

Command structure:

```
GET MIXER_IN_VOL <PRM1> <PRM2> <PRM3>
```

Response Syntax:

```
MIXER_IN_VOL <PRM1> <PRM2> <PRM3>
```

HYB Series

<PRM1> = 1, 2, 3

1: indicates Local audio Mix 1.

2: indicates Local audio Mix 2.

3: indicates Remote USB MIXER.

<PRM2> = OUT1 ~ OUT7, Dantein1 ~ Dantein4, usbhost, micin1 ~ micin2, linein

Anlog input: micin1, micin2, linein

Digital input: Dantein1 ~ Dantein4, usbhost

(usbhost does not work with Remote USB MIXER)

Video De-embed: OUT1 ~ OUT7

<PRM3> = {[-20, 20], [-20, 40]}

When in is LINEIN, DANTEin1~2, OUT1~OUT7, usbhost, the prn = [-20, 20]

When in is MICIN, prn = [-20, 40]

Example Command:

```
GET MIXER_IN_VOL 1 MICIN1 -20
```

EDC Series

<PRM1> = 1, 2, 3

1: indicates Local audio Mix 1.

2: indicates Local audio Mix 2.

3: indicates Remote USB MIXER.

<PRM2> = OUT1 ~ OUT4, Dantein1 ~ Dantein4, usbhost, micin1 ~ micin2, linein}

Anlog input: micin1, micin2, linein

Digital input: Dantein1 ~ Dantein4, usbhost (usbhost does not work with Remote USB MIXER)

Video De-embed: OUT1 ~ OUT4

<PRM3> = {[-20, 20], [-20, 40]}

When in is LINEIN, DANTEin1~2, OUT1~OUT4, usbhost, the prn = [-20, 20]

When in is MICIN, prn = [-20, 40]

Example Response:

```
MIXER_IN_VOL 1 MICIN1 -20
```

Set Audio Mixer Output Volume

<p>Command structure: SET MIXER_OUT_VOL <PRM1> <PRM2></p> <p>Response Syntax: MIXER_OUT_VOL <PRM1> <PRM2></p>	<p>HYB Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p> <p><PRM2> = [-100, 0] -100 means mute</p>
<p>Example Command: SET MIXER_OUT_VOL 1 -100</p>	<p>EDC Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p>
<p>Example Response: MIXER_OUT_VOL 1 -100</p>	<p><PRM2> = [-100, 0] -100 means mute</p>

<p>Get Audio Mixer Output Volume</p>	
<p>Command structure: GET MIXER_OUT_VOL <PRM1></p> <p>Response Syntax: MIXER_OUT_VOL <PRM1> <PRM2></p>	<p>HYB Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p> <p><PRM2> = [-100, 0] -100 means mute</p>
<p>Example Command: GET MIXER_OUT_VOL 1</p>	<p>EDC Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p>
<p>Example Response: MIXER_OUT_VOL 1 -100</p>	<p><PRM2> = [-100, 0] -100 means mute</p>

Mute/Unmute Input Mixer Audio

Command structure:

SET MIXER_IN_MUTE <PRM1> <PRM2> <PRM3>

Response Syntax:

MIXER_IN_MUTE <PRM1> <PRM2> <PRM3>

HYB Series

<PRM1> = 1, 2, 3

1: indicates Local audio Mix 1.

2: indicates Local audio Mix 2.

3: indicates Remote USB MIXER.

<PRM2> = OUT1 ~ OUT7, Dantein1 ~ Dantein4, micin1 ~ micin2, linein

Analog input: micin1, micin2, linein

Digital input: Dantein1 ~ Dantein4, USB Host Audio(USB Host Audio does not work with Remote USB MIXER)

Video De-embed:OUT1 ~ OUT7

<PRM3> = on, off

Example Command:

SET MIXER_IN_MUTE 1 OUT1 ON

EDC Series

<PRM1> = 1, 2, 3

1: indicates Local audio Mix 1.

2: indicates Local audio Mix 2.

3: indicates Remote USB MIXER.

<PRM2> = OUT1 ~ OUT7, Dantein1 ~ Dantein4, micin1 ~ micin2, linein

Analog input: micin1, micin2, linein

Digital input: Dantein1 ~ Dantein4, USB Host Audio(USB Host Audio does not work with Remote USB MIXER)

Video De-embed:OUT1 ~ OUT4

<PRM3> = on, off

Example Response:

MIXER_IN_MUTE 1 OUT1 ON

Query Mute/Unmute Input Mixer Audio

Command structure:

GET MIXER_IN_MUTE <PRM1> <PRM2>

Response Syntax:

MIXER_IN_MUTE <PRM1> <PRM2> <PRM3>

HYB Series

< <PRM1> = 1, 2, 3

1: indicates Local audio Mix 1.

2: indicates Local audio Mix 2.

3: indicates Remote USB MIXER.

<PRM2> = OUT1 ~ OUT7, Dantein1 ~ Dantein4, micin1 ~ micin2, linein

Analog input: micin1, micin2, linein

Digital input: Dantein1 ~ Dantein4, USB Host Audio(USB Host Audio does not work with Remote USB MIXER)

Video De-embed:OUT1 ~ OUT7

<PRM3> = on, off

Example Command:

GET MIXER_IN_MUTE 1 OUT1

Example Response:

MIXER_IN_MUTE 1 OUT1 ON

EDC Series

<PRM1> = 1, 2, 3

1: indicates Local audio Mix 1.

2: indicates Local audio Mix 2.

3: indicates Remote USB MIXER.

<PRM2> = OUT1 ~ OUT7, Dantein1 ~ Dantein4, micin1 ~ micin2, linein

Analog input: micin1, micin2, linein

Digital input: Dantein1 ~ Dantein4, USB Host Audio(USB Host Audio does not work with Remote USB MIXER)

Video De-embed:OUT1 ~ OUT4

<PRM3> = on, off

Mute/Unmute Mixer Output Audio

<p>Command structure: SET MIXER_OUT_MUTE <PRM1> <PRM2></p> <p>Response Syntax: MIXER_OUT_MUTE <PRM1> <PRM2></p>	<p>HYB Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p> <p><PRM2> = ON, OFF</p>
<p>Example Command: SET MIXER_OUT_MUTE 1 ON</p>	<p>EDC Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p>
<p>Example Response: MIXER_OUT_MUTE 1 ON</p>	<p><PRM2> = ON, OFF</p>

Query Mute/Unmute Mixer Output Audio	
<p>Command structure: GET MIXER_OUT_MUTE <PRM1> <PRM2></p> <p>Response Syntax: MIXER_OUT_MUTE <PRM1> <PRM2></p>	<p>HYB Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p> <p><PRM2> = ON, OFF</p>
<p>Example Command: GET MIXER_OUT_MUTE</p>	<p>EDC Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p>
<p>Example Response: MIXER_OUT_MUTE</p>	<p><PRM2> = ON, OFF</p>

Set the Ducking Power Status for the Mixer

<p>Command structure: SET MIXER_DUCK_PWR <PRM1> <PRM2></p> <p>Response Syntax: MIXER_DUCK_PWR <PRM1> <PRM2></p>	<p>HYB Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p> <p><PRM2> = ON, OFF</p>
<p>Example Command: SET MIXER_DUCK_PWR I ON</p>	<p>EDC Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p>
<p>Example Response: MIXER_DUCK_PWR I ON</p>	<p><PRM2> = ON, OFF</p>

<p>Get the Ducking Power Status for the Mixer</p>	
<p>Command structure: GET MIXER_DUCK_PWR <PRM1></p> <p>Response Syntax: MIXER_OUT_MUTE <PRM1> <PRM2></p>	<p>HYB Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p> <p><PRM2> = ON, OFF</p>
<p>Example Command: GET MIXER_DUCK_PWR I</p>	<p>EDC Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p>
<p>Example Response: MIXER_DUCK_PWR I ON</p>	<p><PRM2> = ON, OFF</p>

Select Audio Mixer Ducking Master Input

<p>Command structure: SET DUCK_MASTER <PRM1> <PRM2></p> <p>Response Syntax: DUCK_MASTER <PRM1> <PRM2></p>	<p>HYB Series <PRM1> 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER</p> <p><PRM2> = Dantein1 ~ Dantein4, micin1 ~ micin2</p>
<p>Example Command: SET DUCK_MASTER 1 Dantein1</p>	<p>EDC Series <PRM1> 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER</p>
<p>Example Response: DUCK_MASTER 1 Dantein1</p>	<p><PRM2> = Dantein1 ~ Dantein4, micin1 ~ micin2</p>

<p>Get the selected input of the Audio Mixer Ducking Master</p>	
<p>Command structure: GET DUCK_MASTER <PRM1></p> <p>Response Syntax: DUCK_MASTER <PRM1> <PRM2></p>	<p>HYB Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1 2: indicates Local audio Mix 1 3: indicates Remote USB MIXER.</p> <p><PRM2> = Dantein1 ~ Dantein4, Micin1 ~ Micin2</p>
<p>Example Command: GET DUCK_MASTER 1</p>	<p>EDC Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1 2: indicates Local audio Mix 1 3: indicates Remote USB MIXER.</p>
<p>Example Response: DUCK_MASTER 1 Dantein1</p>	<p><PRM2> = Dantein1 ~ Dantein4, Micin1 ~ Micin2</p>

Set the Ducking Attack time for the Mixer

<p>Command structure: SET DUCK_ATTACK_TIME <PRM1> <PRM2></p> <p>Response Syntax: DUCK_ATTACK_TIME <PRM1> <PRM2></p>	<p>HYB Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1 2: indicates Local audio Mix 2 3: indicates Remote USB MIXER</p> <p><PRM2> = 1 ~ 500 (Step = 10ms)</p>
<p>Example Command: SET DUCK_ATTACK_TIME 1 20</p>	<p>EDC Series < PRM1> = 1, 2, 3 1: indicates Local audio Mix 1 2: indicates Local audio Mix 2 3: indicates Remote USB MIXER</p>
<p>Example Response: DUCK_ATTACK_TIME 1 20</p>	<p><PRM2> = 1 ~ 500 (Step = 10ms)</p>

Get the Ducking Attack time of the Mixer

<p>Command structure: GET DUCK_ATTACK_TIME <PRM1></p> <p>Response Syntax: DUCK_ATTACK_TIME <PRM1> <PRM2></p>	<p>HYB Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1 2: indicates Local audio Mix 2 3: indicates Remote USB MIXER</p> <p><PRM2> = 1 ~ 500 (Step = 10ms)</p>
<p>Example Command: GET DUCK_ATTACK_TIME 1</p>	<p>EDC Series <PRM1> = 1, 2, 3 1: indicates Local audio Mix 1 2: indicates Local audio Mix 2 3: indicates Remote USB MIXER</p>
<p>Example Response: DUCK_ATTACK_TIME 1 20</p>	<p><PRM2> = 1 ~ 500 (Step = 10ms)</p>

Set the Ducking Release time for the Mixer

<p>Command structure: SET DUCK_RELEASE_TIME <PRM1> <PRM2></p> <p>Response Syntax: DUCK_RELEASE_TIME <PRM1> <PRM2></p>	<p>HYB Series <PRM1> = 1, 2, 3</p> <p>1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p> <p><PRM2> 1 ~ 2000 (Step = 10ms)</p>
<p>Example Command: SET DUCK_RELEASE_TIME 1 20</p>	<p>EDC Series < PRM1> = 1, 2, 3</p> <p>1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p>
<p>Example Response: DUCK_RELEASE_TIME 1 20</p>	<p><PRM2> 1 ~ 2000 (Step = 10ms)</p>

Get the Ducking Release time for the Mixer

<p>Command structure: GET DUCK_RELEASE_TIME <PRM1></p> <p>Response Syntax: DUCK_RELEASE_TIME <PRM1> <PRM2></p>	<p>HYB Series <PRM1> = 1, 2, 3</p> <p>1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p> <p><PRM2> 1 ~ 2000 (Step = 10ms)</p>
<p>Example Command: GET DUCK_RELEASE_TIME 1</p>	<p>EDC Series <PRM1> = 1, 2, 3</p> <p>1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p>
<p>Example Response: DUCK_RELEASE_TIME 1 20</p>	<p><PRM2> 1 ~ 2000 (Step = 10ms)</p>

Set the Ducking Depth for the Mixer

<p>Command structure: SET DUCK_DEPTH <PRM1> <PRM2></p> <p>Response Syntax: DUCK_DEPTH <PRM1> <PRM2></p>	<p>HYB Series <PRM2> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p> <p><PRM2> = 1 ~ 100 (step = 1)</p>
<p>Example Command: SET DUCK_DEPTH 20</p>	<p>EDC Series < PRM2> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p>
<p>Example Response: DUCK_DEPTH 20</p>	<p><PRM2> = 1 ~ 100 (step = 1)</p>

<p>Get the Ducking Depth for the Mixer</p>	
<p>Command structure: GET DUCK_DEPTH <PRM1></p> <p>Response Syntax: DUCK_DEPTH <PRM1> <PRM2></p>	<p>HYB Series <PRM2> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p> <p><PRM2> = 1 ~ 100 (step = 1)</p>
<p>Example Command: GET DUCK_DEPTH </p>	<p>EDC Series < PRM2> = 1, 2, 3 1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p>
<p>Example Response: DUCK_DEPTH 20</p>	<p><PRM2> = 1 ~ 100 (step = 1)</p>

Set the Ducking trigger for the Mixer

<p>Command structure: SET DUCK_TRIGGER <PRM1> <PRM2></p> <p>Response Syntax: DUCK_TRIGGER <PRM1> <PRM2></p>	<p>HYB Series <PRM1> = 1, 2, 3</p> <p>1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p> <p><PRM2> = -90 ~ 10 (step = .5dB)</p>
<p>Example Command: SET DUCK_TRIGGER 1 10</p>	<p>EDC Series <PRM1> = 1, 2, 3</p> <p>1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p>
<p>Example Response: DUCK_TRIGGER 1 10</p>	<p><PRM2> = -90 ~ 10 (step = .5dB)</p>

Get the Ducking trigger for the Mixer

<p>Command structure: GET DUCK_TRIGGER <PRM1></p> <p>Response Syntax: DUCK_TRIGGER <PRM1> <PRM2></p>	<p>HYB Series <PRM1> = 1, 2, 3</p> <p>1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p> <p><PRM2> = -90 ~ 10 (step = .5dB)</p>
<p>Example Command: GET DUCK_TRIGGER 1 10</p>	<p>EDC Series <PRM1> = 1, 2, 3</p> <p>1: indicates Local audio Mix 1. 2: indicates Local audio Mix 2. 3: indicates Remote USB MIXER.</p>
<p>Example Response: DUCK_TRIGGER 1 10</p>	<p><PRM2> = -90 ~ 10 (step = .5dB)</p>

Set the Mode of the Amplifier

Command structure: SET AMP_CH_SEL <PRMI>	HYB Series <PRMI> = 1, 2 1: 4x 15 watts 2: 2x 30 watts
Response Syntax: AMP_CH_SEL <PRMI>	
Example Command: SET AMP_CH_SEL 1	
Example Response: AMP_CH_SEL 1	EDC Series N/A

Get the Mode of the Amplifier	
Command structure: GET AMP_CH_SEL	HYB Series <PRMI> = 1, 2 1: 4x 15 watts 2: 2x 30 watts
Response Syntax: AMP_CH_SEL <PRMI>	
Example Command: GET AMP_CH_SEL	
Example Response: AMP_CH_SEL 1	EDC Series N/A

Enable/Disable Phantom Power

<p>Command structure: SET MIC_PHA_PWR <PRM1> <PRM2></p>	<p>HYB Series <PRM1> = 1, 2</p> <p>1 = Mic1, 2 = Mic2</p> <p><PRM2> = on, off</p>
<p>Response Syntax: SET MIC_PHA_PWR <PRM1> <PRM2></p>	
<p>Example Command: SET MIC_PHA_PWR 1 on</p>	<p>EDC Series <PRM1> = 1, 2</p> <p>1 = Mic1, 2 = Mic2</p> <p><PRM2> = on, off</p>
<p>Example Response: MIC_PHA_PWR 1 on</p>	

Query Phantom Power	
<p>Command structure: GET MIC_PHA_PWR <PRM1></p>	<p>HYB Series <PRM1> = 1, 2</p> <p>1 = Mic1, 2 = Mic2</p> <p><PRM2> = on, off</p>
<p>Response Syntax: MIC_PHA_PWR <PRM1> <PRM2></p>	
<p>Example Command: GET MIC_PHA_PWR 1</p>	<p>EDC Series <PRM1> = 1, 2</p> <p>1 = Mic1, 2 = Mic2</p> <p><PRM2> = on, off</p>
<p>Example Response: MIC_PHA_PWR 1 on</p>	

5. USB Switching

Set the USB Host	
Command structure: SET USB_HOST <PRMI>	HYB Series <PRMI> = typec, usbhost1, usbhost2, in9, in10
Response Syntax: USB_HOST <PRMI>	
Example Command: SET USB_HOST typec	EDC Series <PRMI> = typec, usbhost1, usbhost2
Example Response: USB_HOST typec	
Query the USB Host	
Command structure: GET USB_HOST	HYB Series <PRMI> = typec, usbhost1, usbhost2, in9, in10
Response Syntax: USB_HOST <PRMI>	
Example Command: GET USB_HOST	EDC Series <PRMI> = typec, usbhost1, usbhost2
Example Response: USB_HOST typec	

6. GPIO and RELAY Control

Configure the GPIO IN

Command structure:

```
SET GPIO_IN_CTL <PRM1> <PRM2> <PRM3>
```

Response Syntax:

```
GPIO_IN_CTL <PRM1> <PRM2> <PRM3> <PRM4>
```

HYB Series

<PRM1> = 1, 2, 3, 4

<PRM2> = 1-5

(low-state Threshold)

<PRM3> = 1-5

(high-state Threshold)

<PRM4> = high, low

Example Command:

```
SET GPIO_IN_CTL I I 3
```

Example Response:

```
GPIO_IN_CTL I I 3 HIGH
```

EDC Series

<PRM1> = 1, 2, 3, 4

<PRM2> = 1-5

(low-state Threshold)

<PRM3> = 1-5

(high-state Threshold)

<PRM4> = high, low

Get the GPIO IN Configurations

Command structure:

```
GET GPIO_IN_CTL <PRM1>
```

Response Syntax:

```
GPIO_IN_CTL <PRM1> <PRM2> <PRM3> <PRM4>
```

HYB Series

<PRM1> = 1, 2, 3, 4

<PRM2> = 1-5

(low-state Threshold)

<PRM3> = 1-5

(high-state Threshold)

<PRM4> = high, low

Example Command:

```
GET GPIO_IN_CTL I
```

Example Response:

```
GPIO_IN_CTL I I 3 HIGH
```

EDC Series

<PRM1> = 1, 2, 3, 4

<PRM2> = 1-5

(low-state Threshold)

<PRM3> = 1-5

(high-state Threshold)

<PRM4> = high, low

Configure the GPIO OUT

Command structure:

```
SET GPIO_OUT_CTL <PRM1> <PRM2> <PRM3>
```

Response Syntax:

```
GPIO_OUT_CTL <PRM1> <PRM2> <PRM3>
```

HYB Series

<PRM1> = 1, 2, 3, 4

<PRM2> = ON, OFF

<PRM3> = high, low

Example Command:

```
SET GPIO_OUT_CTL I ON HIGH
```

Example Response:

```
GPIO_OUT_CTL I ON HIGH
```

EDC Series

<PRM1> = 1, 2, 3, 4

<PRM2> = ON, OFF

<PRM3> = high, low

GET the GPIO OUT Configurations

Command structure:

GET GPIO_OUT_CTL <PRM1>

Response Syntax:

GPIO_OUT_CTL <PRM1> <PRM2> <PRM3>

Example Command:

GET GPIO_OUT_CTL 1

Example Response:

GPIO_OUT_CTL 1 ON HIGH

HYB Series

<PRM1> = 1, 2, 3, 4

<PRM2> = ON, OFF

<PRM3> = high, low

EDC Series

< PRM1> = 1, 2, 3, 4

<PRM2> = ON, OFF

<PRM3> = high, low

Configure the Relay

Command structure:

SET RELAY <PRM1> <PRM2>

Response Syntax:

RELAY <PRM1> <PRM2>

Example Command:

SET RELAY 1 ON

Example Response:

RELAY 1 ON

HYB Series

<PRM1> = 1, 2

<PRM2> = ON, OFF

EDC Series

<PRM1> = 1, 2

<PRM2> = ON, OFF

GET the Relay Configurations

Command structure:

GET RELAY <PRM1>

Response Syntax:

RELAY <PRM1> <PRM2>

Example Command:

GET RELAY 1

Example Response:

RELAY 1 ON

HYB Series

<PRM1> = 1, 2

<PRM2> = ON, OFF

EDC Series

<PRM1> = 1, 2

<PRM2> = ON, OFF

7. Saving and Recalling an Audio/Video Scene

Save a Scene

Command structure:

SAVE PRESET <PRM>

Response Syntax:

PRESET <PRM>

Example Command:

SAVE PRESET 1

Example Response:

PRESET 1

<PRM> = 1~6

Recall a Scene

Command structure:

RESTORE PRESET <PRM>

Response Syntax:

PRESET <PRM>

Example Command:

RESTORE PRESET 1

Example Response:

PRESET 1

<PRM> = 1~6

8. Controlling Display Power via CEC

CEC Display Power	
Command structure: SET CEC_PWR <OUTPUT> <PRM>	HYB Series <PRM> = on, off <OUTPUT> = out1 ~ out6, all
Response Syntax: CEC_PWR <OUTPUT> <PRM>	EDC Series <PRM> = on, off
Example Command: SET CEC_PWR out2 on	<OUTPUT> = out1 ~ out4, all
Example Response: CEC_PWR out2 on	

Set CEC Auto Power	
Command structure: SET AUTOCEC_FN <OUTPUT> <PRM>	HYB Series <PRM> = on, off <OUTPUT> = out1 ~ out6, all
Response Syntax: AUTOCEC_PWR <OUTPUT> <PRM>	EDC Series <PRM> = on, off
Example Command: SET AUTOCEC_FN out2 on	<OUTPUT> = out1 ~ out4, all
Example Response: AUTOCEC_FN out2 on	

The matrix can automatically send a CEC Power On command to an output when an HDMI input signal is detected. CEC Power Off commands can also automatically be sent after “X” amount of time when a signal detection is lost. See “Set CEC Auto Power Off Delay” section for details.

Query CEC Auto Power	
Command structure: GET AUTOCEC_FN <OUTPUT>	HYB Series <PRM> = on, off <OUTPUT> = out1 ~ out6, all
Response Syntax: AUTOCEC_FN <OUTPUT> <PRM>	
Example Command: GET AUTOCEC_FN out2	EDC Series <PRM> = on, off
Example Response: AUTOCEC_FN out2 on	<OUTPUT> = out1 ~ out4, all

Set CEC Auto Power Off Delay	
Command structure: SET AUTOCEC_D <OUTPUT> <PRM>	HYB Series <PRM> = 1 ~ 30 <OUTPUT> = out1 ~ out6, all
Response Syntax: AUTOCEC_D <OUTPUT> <PRM>	EDC Series <PRM> = 1 ~ 30
Example Command: SET AUTOCEC_D out1 5	<OUTPUT> = out1 ~ out4, all
Example Response: AUTOCEC_FN out1 5	
Note: <PRM> is in minutes. A value of 10 is equal to a 10-minute delay.	

Query CEC Auto Power Off Delay	
Command structure: GET AUTOCEC_D <OUTPUT>	HYB Series <PRM> = 1 ~ 30
Response Syntax: AUTOCEC_D <OUTPUT> <PRM>	<OUTPUT> = out1 ~ out6, all
Example Command: GET AUTOCEC_D out1	EDC Series <PRM> = 1 ~ 30
Example Response: AUTOCEC_D out1 5	<OUTPUT> = out1 ~ out4, all
Note: <PRM> is in minutes. A value of 10 is equal to a 10-minute delay.	

7 Matrix EDID Settings

Set Input EDID

Command structure:
SET EDID <INPUT> <PRM>

Response Syntax:
EDID <INPUT> <PRM>

Example Command:
SET EDID in1 2

Example Response:
EDID in1 2

HYB Series

<INPUT> = in1~in9 , all
<PRM> = 1~26

1 : Copy form HDMI output 1
2 : Copy form HDMI output 2

...

7 : Copy form HDMI output 7
8 : Fixed 4K60 2.0CH PCM Audio with HDR;
9 : Fixed 4K60 2.0CH PCM Audio with SDR;
10 : Fixed 4K30 2.0CH PCM Audio with HDR;
11 : Fixed 4K30 2.0CH PCM Audio with SDR;
12 : Fixed 1080p@60Hz 2.0CH PCM Audio with HDR;
13 : Fixed 1080p@60Hz 2.0CH PCM Audio with SDR

EDC Series

<INPUT> in1 ~ in8
<PRM> = 1 ~ 30

1 : Copy form HDMI output 1
2 : Copy form HDMI output 2
3 : Copy form HDMI output 3
4 : Copy form HDMI output 4
5 : Fixed 4K60 2.0CH PCM Audio with HDR;
6 : Fixed 4K60 2.0CH PCM Audio with SDR;
7 : Fixed 4K30 2.0CH PCM Audio with HDR;
8 : Fixed 4K30 2.0CH PCM Audio with SDR;
9 : Fixed 1080p@60Hz 2.0CH PCM Audio with HDR;
10 : Fixed 1080p@60Hz 2.0CH PCM Audio with SDR;

GET Input EDID

Command structure:
GET EDID <INPUT>

Response Syntax:
EDID <INPUT> <PRM>

Example Command:
GET EDID in1

Example Response:
EDID in1 2

HYB Series

<INPUT> = in1~in9 , all

<PRM> = 1~26

1 : Copy form HDMI output 1
2 : Copy form HDMI output 2

...

7 : Copy form HDMI output 7
8 : Fixed 4K60 2.0CH PCM Audio with HDR;
9 : Fixed 4K60 2.0CH PCM Audio with SDR;
10 : Fixed 4K30 2.0CH PCM Audio with HDR;
11 : Fixed 4K30 2.0CH PCM Audio with SDR;
12 : Fixed 1080p@60Hz 2.0CH PCM Audio with HDR;
13 : Fixed 1080p@60Hz 2.0CH PCM Audio with SDR

EDC Series

<INPUT> in1 ~ in8

<PRM> = 1 ~ 30

1 : Copy form HDMI output 1
2 : Copy form HDMI output 2
3 : Copy form HDMI output 3
4 : Copy form HDMI output 4
5 : Fixed 4K60 2.0CH PCM Audio with HDR;
6 : Fixed 4K60 2.0CH PCM Audio with SDR;
7 : Fixed 4K30 2.0CH PCM Audio with HDR;
8 : Fixed 4K30 2.0CH PCM Audio with SDR;
9 : Fixed 1080p@60Hz 2.0CH PCM Audio with HDR;
10 : Fixed 1080p@60Hz 2.0CH PCM Audio with SDR;

Set EDID Input Write

Command structure:

```
SET EDID_W <INPUT> <PRM1> <PRM2>
```

Response Syntax:

```
EDID_W <INPUT> <PRM1> <PRM3>
```

Example Command:

```
SET EDID_W in1 block0 XX...XX<CR><LF>
```

Example Response:

```
EDID_W in1 block0 ok<CR><LF>
```

HYB Series

<INPUT> = in1~in9

<PRM1> = block0~block1

<PRM2> = one block of 256 bytes EDID ASCII data **w/ spaces** (HEX data must be converted to ASCII)

<PRM3> = ok, error (error= check sum error)

EDC Series

<INPUT> = in1~in8

<PRM1> = block0~block1

<PRM2> = one block of 256 bytes EDID ASCII data **w/ spaces** (HEX data must be converted to ASCII)

<PRM3> = ok, error

(error= check sum error)

GET EDID Input Read

Command structure:

```
GET EDID_R <OUTPUT> <PRM1> <PRM2>
```

Response Syntax:

```
EDID_R <OUTPUT> <PRM1> <PRM2>
```

Example Command:

```
GET EDID_R out1 <CR><LF>
```

Example Response:

```
EDID_R out1 block0
```

```
XX...XX<CR><LF>
```

```
EDID_R out1 block1
```

```
XX...XX<CR><LF>
```

HYB Series

<OUTPUT> = out1 ~ out7

<PRM1> = block0~block1

<PRM2> = one block of 256 bytes EDID ASCII data (HEX data must be converted to ASCII)

EDC Series

<OUTPUT> = out1 ~ out4

<PRM1> = block0~block1

<PRM2> = one block of 256 bytes EDID ASCII data **w/ spaces** (HEX data must be converted to ASCII)

8 Scaling Configuration

8.1 Configuring the scaling outputs

Should the output video need to be scaled to the native resolution of a connected display, the following commands can be implemented.

Setting output resolution	
Command structure: SET VIDOUT_RES <OUTPUT>	HYB Series <OUTPUT> = out1~out7 <PRM> = AUTO, 3840x2160@60 3840x2160@50 3840x2160@30 3840x2160@25 3840x2160@24 1920x1200@60 1920x1080@60 1920x1080@50 1680x1050@60 1600x1200@60 1600x900@60 1440x900@60 1366x768@60 1360x768@60 1280x1024@60 1280x960@60 1280x800@60 1280x768@60 1280x720@60 1280x720@50 1024x768@60 800x600@60
Response Syntax: VIDOUT_RES <OUTPUT> <PRM>	
Example Command: SET VIDOUT_RES out1 1920x1080@60	
Example Response: VIDOUT_RES out1 1920x1080@60	
	EDC Series <OUTPUT> = out1~out4 <PRM> = AUTO, 3840x2160@60 3840x2160@50 3840x2160@30 3840x2160@25 3840x2160@24 1920x1200@60 1920x1080@60 1920x1080@50 1680x1050@60 1600x1200@60 1600x900@60

1440x900@60
1366x768@60
1360x768@60
1280x1024@60
1280x960@60
1280x800@60
1280x768@60
1280x720@60
1280x720@50
1024x768@60
800x600@60

Query output resolution

Command structure:
GET VIDOUT_RES <OUTPUT>

Response Syntax:
VIDOUT_RES <OUTPUT><PRM>

HYB Series

<OUTPUT> = out1~out7
<PRM> =

AUTO,
3840x2160@60
3840x2160@50
3840x2160@30
3840x2160@25
3840x2160@24
1920x1200@60
1920x1080@60
1920x1080@50
1680x1050@60
1600x1200@60
1600x900@60
1440x900@60
1366x768@60
1360x768@60
1280x1024@60
1280x960@60
1280x800@60
1280x768@60
1280x720@60
1280x720@50
1024x768@60
800x600@60

Example Command:

```
GET VIDOUT_RES out1
```

Example Response:

```
VIDOUT_RES out1 1920x1080@60
```

EDC Series

<OUTPUT> = out1~out4
<PRM> =

AUTO,
3840x2160@60
3840x2160@50
3840x2160@30
3840x2160@25
3840x2160@24
1920x1200@60
1920x1080@60
1920x1080@50
1680x1050@60
1600x1200@60
1600x900@60
1440x900@60
1366x768@60
1360x768@60
1280x1024@60
1280x960@60
1280x800@60
1280x768@60
1280x720@60
1280x720@50

1024x768@60
800x600@60

9 Troubleshooting

Query IP Address

Command:
GET IPADDR

Response Syntax:
IPADDR <PRM>

<PRM> = IPv4 Address

Query Firmware Version

Command:
GET VER

Response Syntax:
VER <PRM>

<PRM> = current installed firmware version

Reboot Matrix

Command:
REBOOT

Response:
REBOOT

No Parameters

Restore Factory Defaults

Command:
RESET

Response:
RESET

No Parameters

10 Contacting Technical Support

Should further clarification of the content in this document or assistance on troubleshooting be required, please contact WyreStorm technical support.

Phone: UK: +44 (0) 1793 230 343 | ROW: 844.280.WYRE (9973)

Contact Request: <http://wyrestorm.com/contact-tech-support>

11 Document Revision History

V1.0 -December 2023 - Initial release

Publication Disclaimer

The material contained in this document consists of information that is the sole property of WyreStorm. This document is intended to provide information to allow interfacing to the relevant WyreStorm equipment by third party products.

WYRESTORM IS NOT RESPONSIBLE FOR MALFUNCTIONS AND/OR THE IN-OPERABILITY WHICH MAY BE CAUSED BY THE APPLICATION OF THIS INFORMATION, WHETHER EXPECTED OR NOT.

WyreStorm reserves the right to change software, control codes and specifications without notice.

WyreStorm will not be liable for any use of this information or any changes it may make to those products. The use of this information constitutes an agreement by the user to these limitations and exclusions.



WyreStorm Offices

North America: 27 Wood Road, Round Lake, NY
12151 Tel: +1 518-289-1293

EMEA: Unit 22, Ergo Business Park, Swindon, Wiltshire, SN3 3JW,
UK Tel: +44 (0) 1793 230 343

WyreStorm Technologies reserves the right to change the physical appearance or technical specification of this product at any time. Visit wyrestorm.com for the latest product information