



OCTO MK2 – API

API requirement for the 8-universe eDMX to LED pixel controller with network chaining in a compact 4-module DIN-rail form factor.



Document Version	1.0
Last Updated	3 May 2023
Compatible firmware	OCTO MK2 Firmware Version V4.0-4.1

Purpose

This document specifies the interface requirements for text-based application programs such as Command Prompt or third-party applications to communicate with the ENTTEC OCTO MK2 Firmware Version V4.0 -4.1.

Application Messages

This section demonstrates the API message format required for different communication purposes alongside the parameters for each setting.

The API messages are presented in examples factored in the followings:

- PC Command Line Interface (CLI): CURL. **Note:** Replace ‘curl --’ if other CLI is used.
- The IP address of OCTO MK2: 10.10.3.51. **Note:** Replace this IP address with the one from your device.

1. Current Configuration

This example message requests OCTO MK2 to display the current configuration.

Example Message

```
curl --http0.9 http://10.10.3.51/config.cgi
```

2. Change Settings

This application message allows user to change settings of the OCTO MK2 within one command.

Example Message

Space here!

```
curl --header "Content-Type: application/x-www-form-urlencoded" -d
"ip=192,168,0,10&netmask=255,255,255,0&dhcp_enable=1&gateway_ip=192,168,0,254&config_name=Octo&le
d_strip_protocol=16,16&lighting_protocol=0&universe_count=4,4&start_universe=0,0&dmx_start_address=
1,1&pixel_fanout=1,1&pixel_order=0,0&used_pixels=0&apa102_global=31,31&tm1814_cc=63,63&custom_bit1
=1090,750&custom_bit0=300,300&custom_bitlength=1410,1350&custom_reset=280000,100000&custom_pro
tocol=0,0" 10.10.3.51/settings.html
```

The message is composited with 3 elements:

- Header line
- Configuration line: where each individual setting is placed together with “&” and the changeable values are highlighted in magenta.
- Device IP and command area line: replace the IP address from your device.

To change settings, enter the **Example Message** above with your IP address replaced and adjust the changeable values by following the **Application Message Format and Parameter Table**.

The changes will be conducted directly without a return message. Request current configuration information for verification after change.

Note: Due to pdf format restriction, it is recommended to edit the code in plain text editing tool such as Notepad before copying the code to the prompt command.

Application Message Format and Parameter Table:

Settings	Description	Parameter
ip	The Static IP when DHCP is disabled.	Format: ip= <i>x,x,x,x</i> Parameter <i>x</i> = 0~255 <i>E.g.</i> 192.168.0.10 = 192,168,0,10
netmask	The netmask when static IP is enabled.	Format netmask= <i>x,x,x,x</i> Parameter <i>x</i> = 0~255
dhcp_enable	Enable or disable DHCP.	Format dhcp_enable= <i>x</i> Parameter <i>x</i> = 0 or 1 0: disable 1: enable
gateway_ip	Gateway IP when static IP is enabled.	Format gateway_ip= <i>x,x,x,x</i> Parameter <i>x</i> = 0~255
config_name	The node name discoverable on the network.	Format config_name= <i>x</i> Parameter <i>x</i> = any value in alphabet and number. <i>E.g.</i> Node Name: OCTO MK2 config_name=OCTO MK2

led_strip_protocol	Select the output LED protocol.	<p>Format led_strip_protocol=<i>x,y</i></p> <p>Parameter <i>x</i>= 0~22, Port 1 LED protocol <i>y</i>= 0~22, Port 2 LED protocol</p> <p><i>LED protocol list in Appendix 1.</i></p>
lighting_protocol	Choose input eDMX protocol.	<p>Format lighting_protocol=<i>x</i></p> <p>Parameter <i>x</i>= 0/1/2/3</p> <p>0: Art-Net 1: sACN 2: ESP 3: KiNet</p>
universe_count	Set universe capacity for Port 1 and 2.	<p>Format universe_count=<i>x,y</i></p> <p>Parameter <i>x</i>= 0/1/2/3/4, Port 1 universe capacity <i>y</i>= 0/1/2/3/4, Port 2 universe capacity</p>
start_universe	Define starting universe for Port 1 and 2.	<p>Format start_universe=<i>x,y</i></p> <p>Parameter <i>x</i>= see below, Port 1 starting universe <i>y</i>= see below, Port 2 starting universe</p> <p>0-32767 for Art-Net 0-255 for ESP 0-63999 for sACN 0-65535 for KiNet</p>
dmx_start_address	Nominate the starting DMX channel address for OCTO MK2 to output from Port 1 and 2.	<p>Format dmx_start_address=<i>x,y</i></p> <p>Parameter <i>x</i>= 1-512, Port 1 DMX starting address <i>y</i>= 1-512, Port 2 DMX starting address</p>
pixel_fanout	Pixel grouping which allows multiple pixels to be controlled as one 'virtual pixel'.	<p>Format pixel_fanout=<i>x,y</i></p> <p>Parameter <i>x</i>= 0-512*, Port 1 grouped pixels <i>y</i>= 0-512*, Port 2 grouped pixels</p>
pixel_order	Configure how R, G, B, W colours are mapped to pixels	<p>Format pixel_order=<i>x,y</i></p> <p>Parameter <i>x</i>= 0-35, Port 1 pixel order <i>y</i>= 0-35, Port 2 pixel order</p> <p><i>Pixel order list in Appendix 1.</i></p>
used_pixels	Define the number of mapped pixels.	<p>Format used_pixels=<i>x,y</i></p> <p>Parameter <i>x</i>= 0~680*, Port 1 mapped pixels. <i>y</i>= 0~680*, Port 2 mapped pixels.</p>

* Max. value depending on other output settings such as universe, pixel order and DMX start address. If set value is higher than the max value allowed, OCTO MK2 will proceed with the max value allowed.

apa102_global	Maximum global brightness when APA-102 is selected.	<p>Format apa102_global=<i>x,y</i></p> <p>Parameter <i>x</i>= 0-63, Port 1 APA-102 global brightness <i>y</i>= 0-63, Port 2 APA-102 global brightness</p>
tm1814_cc	Maximum global brightness when TM-1814 is selected.	<p>Format tm1814_cc=<i>x,y</i></p> <p>Parameter <i>x</i>= 0-63, Port 1 TM-1814 global brightness <i>y</i>= 0-63, Port 2 TM-1814 global brightness</p>
custom_bit1	Bit 1 High Time (T1H) for custom LED protocol. This will only take effect when custom protocol is enabled.	<p>Format custom_bit1=<i>x,y</i></p> <p>Parameter <i>x</i>= 100-3,000#, Port 1 T1H <i>y</i>= 100-3,000#, Port 2 T1H</p>
custom_bit0	Bit 0 High Time (T0H) for custom LED protocol. This will only take effect when custom protocol is enabled.	<p>Format custom_bit0=<i>x,y</i></p> <p>Parameter <i>x</i>= 100-3,000#, Port 1 T0H <i>y</i>= 100-3,000#, Port 2 T0H</p>
custom_bitlength	Overall bit Time (TH+TL) for custom LED protocol. This will only take effect when custom protocol is enabled.	<p>Format custom_bitlength=<i>x,y</i></p> <p>Parameter <i>x</i>= 500-3,000, Port 1 overall bit time <i>y</i>= 500-3,000, Port 2 overall bit time</p>
custom_reset	Reset Time (RES) for custom LED protocol. This will only take effect when custom protocol is enabled.	<p>Format custom_reset=<i>x,y</i></p> <p>Parameter <i>x</i>= 25,000-1,000,000, Port 1 Reset time <i>y</i>= 25,000-1,000,000, Port 2 Reset time</p>
custom_protocol	Enable or disable LED custom protocol for Port 1 and 2.	<p>Format custom_protocol=<i>x,y</i></p> <p>Parameter <i>x</i>= 0 or 1, Port 1 custom protocol status <i>y</i>= 0 or 1, Port 2 custom protocol status</p> <p>0: disable 1: enable</p>

Max. value depending on overall bit length set. If set value is higher than the max value allowed, OCTO MK2 will proceed with the max value allowed.

3. Reset to Factory Default

This message requests the OCTO MK2 resume settings back to factory default.

Example Message

```
curl --http0.9 http://10.10.3.51/index.html?set\_to\_default
```

The device will be reset to factory default without a return message. Request current configuration information for verification after change.

4. Access to Boot

This message requests to access the boot of the OCTO MK2.

Example Message

```
curl --http0.9 http://10.10.3.51/index.html?firmware
```

Once successful, the return message will be *'curl: (56) Recv failure: Connection was reset.'*

5. Reboot Device

This message requests the OCTO MK2 to reboot or to exit the boot.

Example Message

```
curl --http0.9 http://10.10.3.51/index.html?reboot
```

Once successful, the return message would be *'curl: (56) Recv failure: Connection was reset.'*

6. Identify

This message requests OCTO MK2 to send output to light up all the connected pixels for correct wiring identification without the need to provide control data.

Example Message

```
curl --http0.9 http://10.10.3.51/index.html?identification
```

7. Art-Net Stats

This API message requests OCTO MK2 to display the current Art-Net Stats.

Example Message

```
curl --http0.9 http://10.10.3.51/stat.cgi?artnet
```

8. ESP Stats

This API message requests OCTO MK2 to display the current ESP Stats.

Example Message

```
curl --http0.9 http://10.10.3.51/stat.cgi?esp
```

9. sACN Stats

This API message requests OCTO MK2 to display the current sACN Stats.

Example Message

```
curl --http0.9 http://10.10.3.51/stat.cgi?acn
```

10. KiNet Stats

This message requests OCTO MK2 to display the current KiNet Stats.

Example Message

```
curl --http0.9 http://10.10.3.51/stat.cgi?kinet
```

Conclusion

This brings us to the end of the guide. By utilizing the API messages, OCTO MK2 allows third-party hardware integration and communications from user's preferable command system. This guide provides the message format required for communication interface in examples. While there will be return messages for most of the commands, the Change Settings and Reset to Factory Default command will be executed directly. It is recommended to request current configuration information for verification.

enttec.com

MELBOURNE AUS / LONDON UK / RALEIGH-DURHAM USA / DUBAI UAE

Due to constant innovation, information within this document is subject to change.

Appendix 1

Parameter Table

This section provides the parameter for the LED Strip Protocol and Pixel Order in [Change Settings](#).

LED Strip Protocol

0	APA-104
1	SK6812
2	SPXL_16BIT
3	SPXL_8BIT
4	TM1804
5	TM1812
6	TM1814
7	UCS1903
8	UCS2903
9	UCS2904
10	UCS8903_16BIT
11	UCS8903_8BIT
12	UCS8904_16BIT
13	UCS8904_8BIT
14	WS2811
15	WS2812
16	WS2812B
17	WS2813
18	WS2815
19	WS2818
20	9PDOT_16BIT
21	9PDOT_8BIT
22	APA-102

Pixel Order

0	RGB
1	RBG
2	GRB
3	GBR
4	BRG
5	BGR
6	RGBW
7	RGWB
8	RBGW
9	RBWG
10	RWGB
11	RWBG
12	GRBW
13	GRWB
14	GBRW
15	GBWR
16	GWRB
17	GWBR
18	BRGW
19	BRWG
20	BGRW
21	BGWR
22	BWRG
23	BWGR
24	WRGB
25	WRBG
26	WGRB
27	WGBR
28	WBRG
29	WBGR
30	WWA_RGB
31	WWA_RBG
32	WWA_GRB
33	WWA_GBR
34	WWA_BRG
35	WWA_BGR