

Build your own DMX tester: With Open Lighting Architecture and Raspberry Pi

BY MICHAEL HESMOND

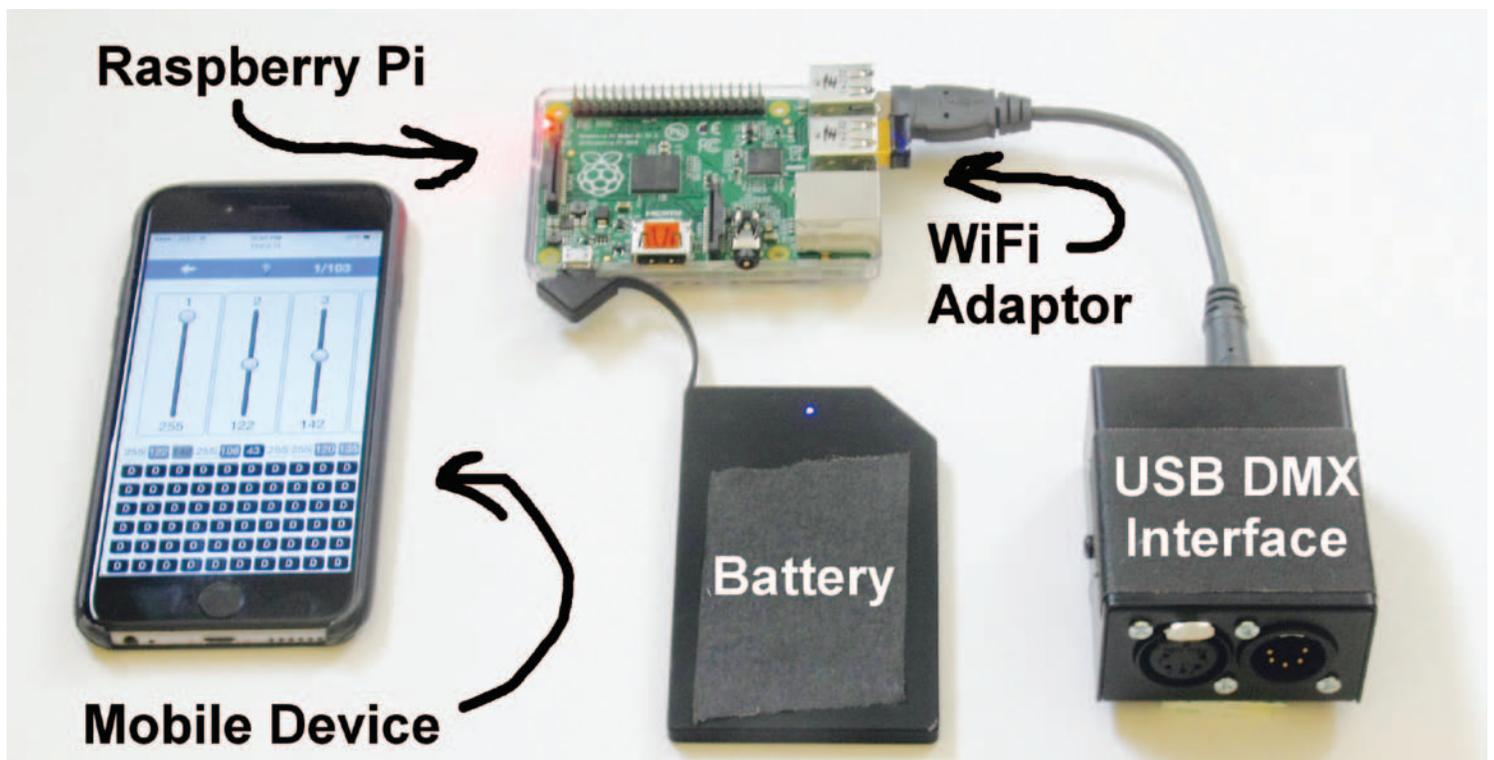


Figure 1 – What you need

Introduction

MAYBE YOU ARE FAMILIAR WITH THE RASPBERRY PI MICROCOMPUTER, or maybe the name just makes you a little hungry. For the unfamiliar, the Raspberry Pi is a credit card sized fully functioning Linux-based computer available for around \$35. In addition to having USB, networking, video, and audio ports like a regular PC, the Raspberry Pi also has a set of general-purpose input/output (GPIO) pins available for connecting external devices like LEDs, motors, sensors, cameras, and more. The Pi's low cost and I/O options make it an ideal platform for any hobbyist or professional

interested in making his or her own device that uses a DMX512 signal.

Open Lighting Architecture (OLA) is free, open source software originally created by Simon Newton and now developed by a team of contributors around the world. It runs on Linux or Mac and is capable of interfacing with USB DMX512 hardware, DMX512 over IP protocols, and the Raspberry Pi's GPIO pins. The application includes a web interface for easily creating, monitoring, and configuring DMX universes. OLA is one part of the larger Open Lighting Project, which aims to build high-quality, free software for the entertainment lighting industry.



Figure 3 – Finished device

About this project

I picked up my first Raspberry Pi about two years ago and started to look for a way to make it work with entertainment lighting equipment. I think the potential uses in the field are exciting, and definitely worth the initial investment required to begin exploring. In my quest to make the Raspberry Pi send and receive DMX512, I stumbled upon the Open Lighting Project and was blown away by the work that has already been done in this area. It is truly a global effort supported by developers and users from around the world, continually adding new features and improvements to the software. In order to get the actual DMX signal out of the Raspberry Pi, I thought I might need to break out my soldering iron, but there is no need to reinvent the wheel. OLA supports many common USB DMX512 interfaces. Simply plug one in and the Raspberry Pi is sending and receiving the DMX512 signal you know and love.

Initially, I didn't set out to make a DMX512 tester but I had my eye on some of the nicer versions that are commercially available. If you want RDM capabilities you can expect to spend over \$500, a little pricey for a tool I only need every now and then. Still, they are the way to go if you want something that works out of the box and is supported directly. As built, this project cost me around \$200. While it does not do everything some professional products can, in many ways it is capable of a lot more. Without much effort, you can use OLA to turn this tester into an Art-Net or sACN gateway, LED tape controller, or convert one DMX over IP protocol to another.

You can use this DMX tester to set simple scenes, remote focus, or troubleshoot. By attaching RDM enabled devices, you can set parameters like DMX addresses and personalities remotely. There

are no controls on the unit itself. Instead, a Wi-Fi adaptor allows you to configure and control everything from a browser on your smartphone or computer connected to a wireless network set up on the Raspberry Pi. The power requirements are similar to most tablets or mobile devices. For portability, you can choose to run the whole thing with a rechargeable cell phone battery backup. Once I got everything working correctly, I bundled it with some trusty gaff tape and threw it in with the rest of my tools. When I need it, I have a simple lighting console in my back pocket.

Words of encouragement

These instructions can get you up and running in a few hours by only entering seven commands into the Linux command line. Essentially, you only need to download the OS and setup a wireless network, everything else is plug and play. This guide assumes you have no prior experience with Linux or computer programming. Installing Open Lighting Architecture on the Raspberry Pi is a straight-forward process, however some persistence may be required. Thankfully, the Internet holds a wealth of information and support for both the Raspberry Pi and OLA. Check out raspberrypi.org/help and openlighting.org or chat with the folks in the #openlighting IRC channel on webchat.freenode.net/?channels=openlighting.

Step by Step Instructions:

Download the image file and write it onto your SD card.

The most common Operating System installed on the Raspberry Pi is called Raspbian. The Open Lighting Project distributes a stripped down version of this OS with OLA already installed. Getting the operating system onto the SD card is more than just a copy and paste process. You will need to download an application and use a Windows computer to complete this step.

1. Connect the SD card to your computer and identify the drive letter.
2. Download and unzip the latest SD image file "raspbian-ola-x.x.x.zip" from dl.openlighting.org.
3. Download and unzip the Win 32 Disk Imager application from sourceforge.net/projects/win32diskimager.
4. Right click on the Win 32 Disk Imager program and select "Run as Administrator."
5. Using Win32 Disk Imager, locate the "raspbian-ola-x.x.x.img" image file you unzipped and select the drive letter of your SD card.
6. Click "Write." This process can take a several minutes. *Now would be a good time to go grab a beverage of your choosing.*
7. When the process is complete, eject the SD card.

Required Materials		
Part	As Built	Description
Raspberry Pi	Raspberry Pi Model 1 B+	There are a few different flavors of the Raspberry Pi. The Model 1 B+ works well for this application. The Model 2 B is more powerful, but uses more power.
Micro SD card	Samsung 16GB	Any card at least 4GB should work. Be sure you get the proper adapters needed in order to connect the card to your computer.
USB DMX Interface	Enttec DMX USB Pro	OLA supports many different USB DMX interfaces. I chose this one for its cost and RDM capabilities. A complete list of supported interfaces can be found at openlighting.org/ola .
USB Wifi Adaptor	Edimax EW-7811Un	The Raspberry Pi does not have any wireless hardware built in, so for this application you will need to add one. A list of compatible adaptors can be found at elinux.org/RPi_USB_Wi-Fi_Adapters .
Micro USB Power Supply	AmazonBasics 2,000 mAh Portable External Battery Charger	For reliable and continuous operation you should use a 5V 1.2A or greater wired power supply. For portability, I chose a USB charger commonly used as a cell phone battery backup. This battery pack does not technically meet the power requirements for the Pi , but it worked in this application. You can expect one hour of battery life per 1000 mAh of battery capacity.
Raspberry Pi case or Project Box		You will want something to protect your Pi; choose something off the shelf or 3D print your own.
Windows Computer with an SD card reader.		The instructions use a Windows program to write the SD card. You can find instructions for using a Mac at elinux.org/RPi_Easy_SD_Card_Setup
Peripheral Components		HDMI monitor, USB keyboard, cables and adapters. <i>It is possible to access the Raspberry Pi's command line without a monitor and keyboard, but I recommend using them if you're just getting started.</i>
Internet Connection		A wired Internet connection is needed to download a program on the Raspberry Pi.
Access to DMX controlled devices		For testing and application

Figure 4 – Required materials list

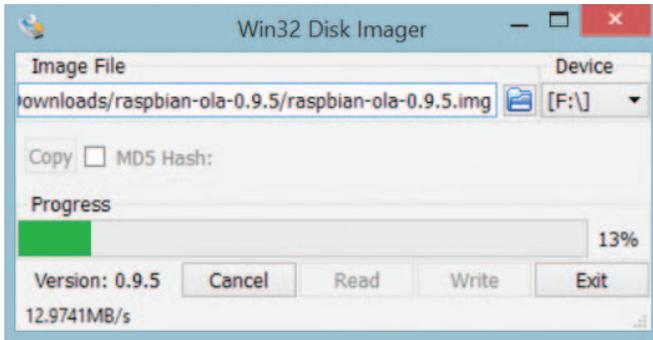


Figure 5 – Win 32 Disk Imager Application

Boot up and log into the Pi

Before you apply power and boot up, you should connect your SD card, Wi-Fi adaptor, wired network connection, monitor, keyboard, and any other peripheral devices you need. I recommend using a hard-wired power supply for this part as it can take some time and you don't want to run out of battery half way through the initial setup.

Plug the USB power supply into the input of your Raspberry Pi. If everything is working correctly you should see the LEDs on your devices start to blink and a rainbow image, followed by some text on your monitor. The system can take a few minutes to boot up fully the first time. On the first successful boot, the Pi will automatically go into the Raspberry Pi Configuration Tool (**raspi-config**). This utility allows you to configure and enable some basic hardware and user settings. You should start by expanding the filesystem so the OS has full use of the storage on your SD card.

1 Expand Filesystem

Select with **[enter]** and confirm. Use the arrow keys to navigate down to **<Finish>**, and when prompted to reboot select **<Yes>**. After the computer has finished rebooting, go ahead and login to get access to the command line.

```
raspberrypi login: pi
password: openlighting
pi@raspberrypi ~ $
```

If you made it to this point, congratulations! You are already running OLA. If you want to test to see that it's working, you can run a simple command that brings up a DMX monitor.

```
pi@raspberrypi ~ $ ola_dmxdmmon
```

There are some very useful and powerful command line tools built into OLA, but for this project the web-based UI is more user friendly. To exit from this screen simply hit the **[q]** key on the

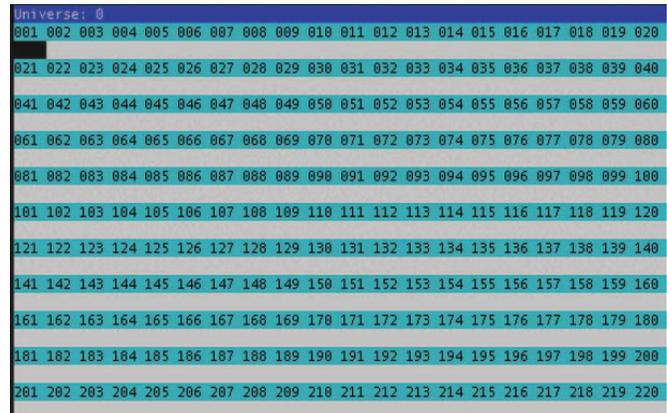


Figure 6 – OLA DMX Monitor Command

keyboard. If you want to learn more about the command line tools visit wiki.openlighting.org/index.php/OLA_Command_Line_Tools.

Setup the Wi-Fi network

OLA has a great web-based user interface that can be accessed from any browser connected to the same local network. There is no need to haul around a wireless router; you can setup a wireless network right on the Raspberry Pi. For this part, you need to edit a text file called **interfaces**. It will contain all the information needed to create your own wireless network. To edit the text you can use the built-in text editor called **nano**. Type the following into the command line:

```
pi@raspberrypi ~ $ sudo nano /etc/network/interfaces
```

sudo is a prefix added to some commands that gives you administrator privileges.

nano is a simple text editing program. No point and click here, use the arrow keys to move the cursor and the keyboard to add or delete text. Modify the file so that it contains the following:

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlan0
iface wlan0 inet static
address 10.0.3.14
netmask 255.0.0.0
wireless-channel 1
wireless-essid DMXRemote
wireless-mode ad-hoc
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

Save the file using the keys [ctrl]-[o] and confirm with [enter]. You can exit using [ctrl]-[x].

Test your Internet connection

Make sure you have a wired Internet connection hooked up to the Ethernet port of your Raspberry Pi. You can check your connection with a simple ping test.

```
pi@raspberrypi ~ $ ping google.com
```

This test will run until you quit using [ctrl]-[c]. If everything is working correctly you will see a list of packets transmitted and received that looks something like this:

```
PING google.com (216.58.216.206) 56(84) bytes of data:
64 bytes from ord31s21-in-f14.1e100.net: icmp_req=1 ttl=56 time=9.78 ms
64 bytes from ord31s21-in-f14.1e100.net: icmp_req=2 ttl=56 time=9.45 ms
64 bytes from ord31s21-in-f14.1e100.net: icmp_req=3 ttl=56 time=9.19 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss,
time 7012ms
rtt min/avg/max/mdev = 9.196/9.535/9.824/0.244 ms
```

Download and configure a DHCP server

In order to connect a Wi-Fi device to your network automatically you need to add a DHCP server that hands out IP addresses. Download and install the DHCP server called `dnsmasq` using the `apt-get` command. Before you install anything, it is good practice to make sure everything else is up to date. Use the following commands:

```
pi@raspberrypi ~ $ sudo apt-get update
pi@raspberrypi ~ $ sudo apt-get install dnsmasq
```

Next, you need to setup the server by editing another text file named `dnsmasq.conf`. Use `nano` again with the command:

```
pi@raspberrypi ~ $ sudo nano /etc/dnsmasq.conf
```

This file uses the `#` symbol to comment out lines of the code. By adding or deleting the `#` you can turn on or off certain features. For this setup, simply add the following text to the very top of the file.

```
interface=wlan0
dhcp-range=10.0.3.50,10.0.3.150,255.0.0.0,12h
```

Save, confirm and exit using [ctrl]-[o], [enter], [ctrl]-[x]. The hard part is done! Now you can shut down with

```
pi@raspberrypi ~ $ sudo halt
```

Open Lighting Architecture's user interface

One of the things I like most about OLA is the ability to log in and set up DMX universes remotely from any web browser on a computer connected to the same local network. Currently, there is a sleek new user interface under development. It is still in beta, but it is included on the most recent release for testing purposes. The new UI works a little better on mobile devices, but is still lacking some important features. There is also a "full version" that works really well on a desktop browser. This section highlights the working parts of the new UI as well as the old "full" and "mobile" versions.

Connect to the OLA server.

For this part, you will need to use another device to connect to the "DMXRemote" Wi-Fi network and access OLA through a web browser.

1. Power cycle your Raspberry Pi. After it is finished booting up, connect a computer, tablet, or smartphone to the Wi-Fi network you created. The SSID of the network is "DMXRemote." Sometimes the network can show up under a separate "devices" section in your Wi-Fi settings. No password is required. (Windows 8 will not automatically discover this ad-hoc network.)
2. Open up a web browser on your device and type in "10.0.3.14:9090". This is the IP address of your Raspberry Pi followed the port of the OLA web server.
3. If everything is working correctly you should see the OLA home screen. **Scroll down and select new UI.** Now is a good time to bookmark this page or add a link to your home screen for quick access.

Create a universe and output DMX

Any DMX data in or out of OLA must be patched to a specific Universe. Here you will create Universe Id 1 and give it a name.

1. From the OLA home screen select "Add Universe."
2. Assign the Universe ID number "1."
3. Assign the Universe Name. I used "USBDMX" but feel free to call it whatever you want.
4. Scroll down the list of available input and output ports and select your USB DMX interface under the Output Ports column. We can leave the input empty and control the output via OLA itself.
5. Scroll back up to the top and click Add Universe



Figure 7 – Connect to OLA

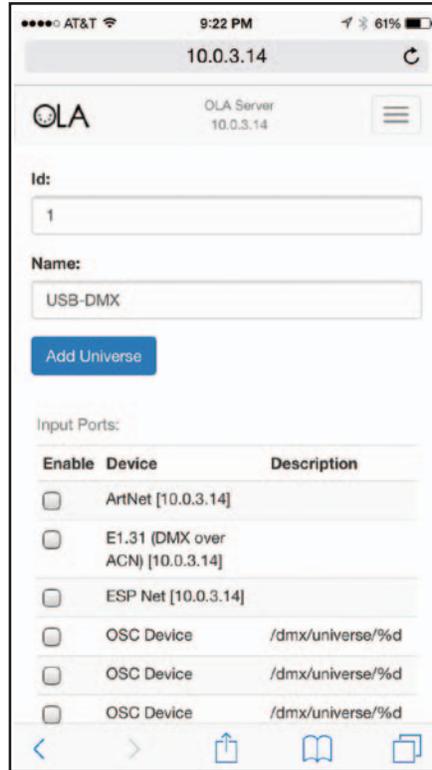


Figure 8 – Creating a Universe

contains a list of fixtures and the right column allows you to set parameters.

RDM Patcher

Click and drag fixture addressing at your fingertips. Click the magic wand to address everything automatically!

Going Deeper

If you are thinking about building your own Franken-light or controlling your next show via Twitter, you will most likely want to dive a little deeper. Here are some good first steps.

Running headless

If your Pi is wired into your home router or other network, you can easily ditch the keyboard and monitor and access the command line using the SSH protocol from a remote computer. The only tricky part can be finding the IP

address of the Raspberry Pi. Here are a few methods that might work:

- Plug in a monitor and view the IP address displayed after booting up.
- If you have a Bonjour browser like <http://hobbyistsoftware.com/bonjourbrowser> you can use it to find your Pi.
- Log into your router and view the DHCP clients
- Use the Fing app from a mobile device overlooksoft.com/fing
- Use Nmap from your computer. Available at nmap.org/download.html

For more information, check out <http://www.raspberrypi.org/documentation/troubleshooting/hardware/networking/ip-address>.

To use SSH from a Windows computer, download Puty at <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.

From Mac or Linux; open up terminal and type in the following, replacing “x.x.x.x” with the IP address of the Pi.

```
ssh pi@x.x.x.x
```

A DMX over IP node

Chances are you’ve already realized you’re only a click or two away from creating your own Art-Net or sACN gateway. In this case OLA’s Universe Id number will correspond to your protocol’s universe number. The wired networking on the Pi will accept any IP address from a DHCP server on your network. If you need to assign a static IP you can go back and edit the network config file with:

```
pi@raspberrypi ~ $ sudo nano /etc/network/interfaces
```

Save Your Configuration

In order to save your settings in OLA, you need to shutdown the application. Simply click “**Shutdown**” from the home screen in the new UI or “**Stop OLA**” in the old UI. When you cycle the power and reboot, all of your settings should be saved.

How to use

From the home screen you can select which universe you want to view. Use the tabs to access different functions.

Overview

This page gives you a view of individual DMX channel levels, similar to the `ola_dmxfmonitor` command used earlier. If you set up your USB DMX device as an input, the Overview will show you the output of a connected lighting console.

Faders

This is your lighting console; each channel has its own fader. You can page between groups of faders or use the light bulbs to set all channels to full or zero.

Keypad

Available in the old UI, this is the easiest way to bring up individual or blocks of channels.

RDM

This works best from the “full version” of the old UI. Click on the magnifying glass to run an RDM discovery. The left column

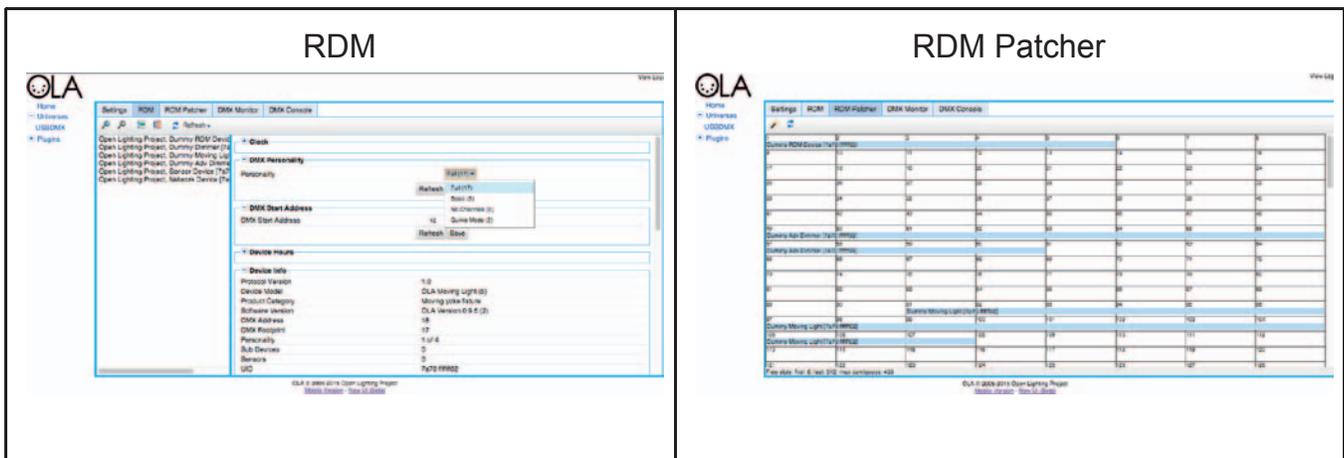
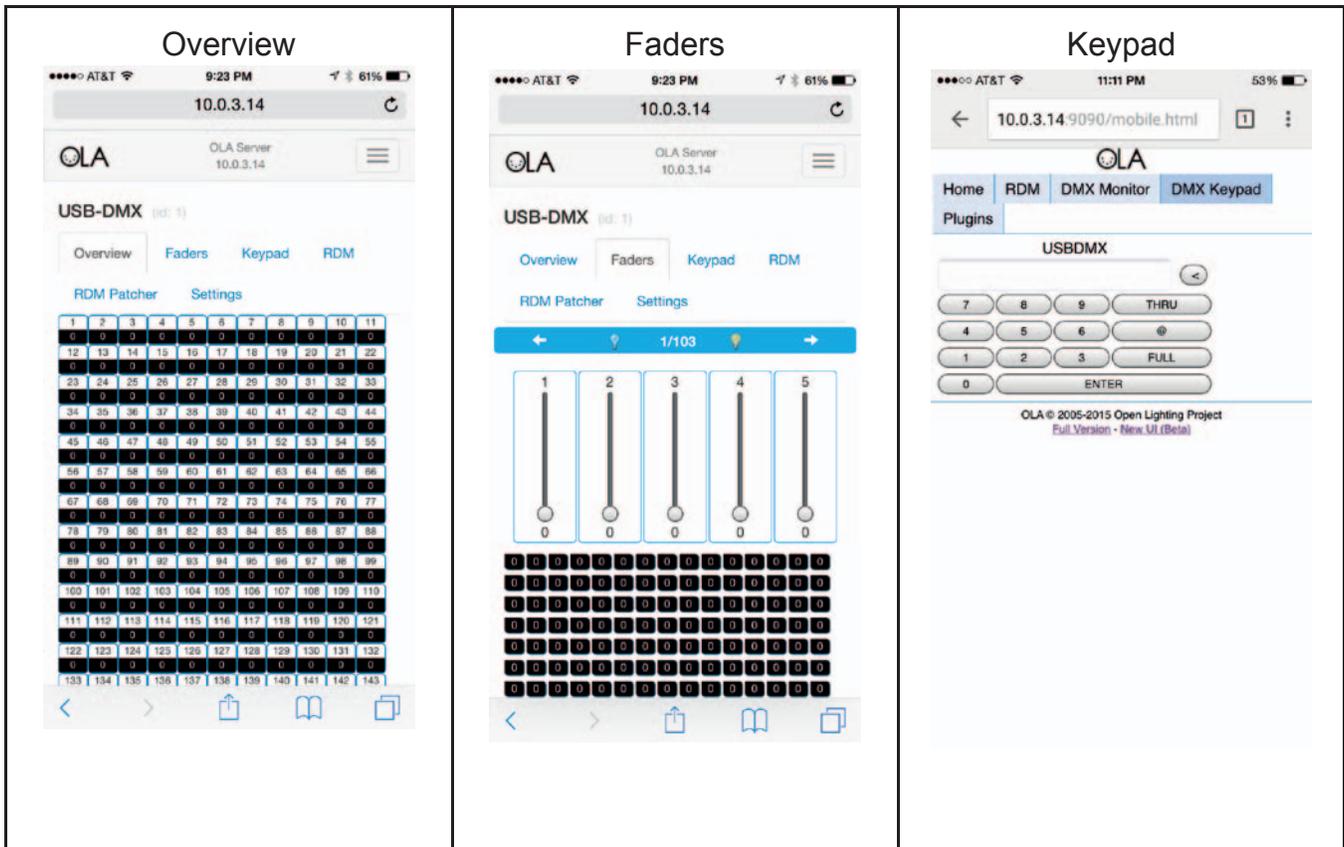


Figure 9 – Tab screenshots

The wired network port is named **eth0**.

Backup your image

You can use Win32 Disk Imager to store a backup of your SD image. Under “Image File,” name the file you’d like to create, insert your SD card and click “Read.”

Updating to a new version of OLA

New updates to OLA are released every few months. You don’t have to redownload the image each time you want to upgrade.

Simply run the following commands when connected to the Internet in order to get the latest version.

```
pi@raspberrypi ~ $ sudo apt-get update
pi@raspberrypi ~ $ sudo apt-get upgrade
```



Michael Hesmond is a freelance lighting designer, programmer, and technician with a background in electrical engineering and physics. His experience ranges from theatre and corporate events to television and concerts.