

# RDM responder testing with the Raspberry Pi

BY SIMON NEWTON



What does one man's effort to promote computing in the UK education system have to do with the entertainment lighting industry? Read on to find out.

THE AUTOMATED RDM RESPONDER TESTS, first covered in the Spring 2011 edition of *Protocol*, have been available for almost two years. The tests evaluate the quality of a product's RDM implementation and have been widely used by manufacturers both at Plugfests and during product development. Since the tests were developed, they have been continually updated based on feedback received from manufacturers. There are now 235 test cases covering a wide range of conditions, from corner cases with the RDM discovery process to out-of-range sensor values.

**My goal was to produce a system that would allow people to begin testing within minutes of downloading the software.**

However, while the tests themselves worked well, the barrier to testing was still too high. By far, the most common feedback received from manufacturers was that the tests were difficult to set up and run. The procedure required access to a Linux or Mac machine, and a complex set of instructions had to be followed to download, configure, build, and run the tests. Some people tried to run the tests using VMWare on Windows machines, but they experienced problems with VMWare's USB virtualization layer. The output from the tests also caused frustration for users. It was difficult for

people who weren't familiar with the Unix command line to save the test results or to filter the test output to isolate the problem they were interested in.

To address these concerns, I sat down in early 2012 to design a better way of running the RDM Responder Tests. My goal was to produce a system that would allow people to begin testing within minutes of downloading the software. The key to meeting the goals was the hardware platform. I required a single-board, embedded Linux system on which it was easy to perform software upgrades.

Around the time I was looking for a suitable hardware platform, across the Atlantic there was something else going on.

## Enter the Raspberry Pi

Back in 2006, while working in undergraduate admissions at the University of Cambridge, Eben Upton noticed a disturbing trend among students applying to the university's computer science program. Whereas in previous years many students entering the course had programming experience, Eben noticed that now many

students' exposure to computer science was limited to some rudimentary web design.

Eben set out to build a low-cost computer that would encourage experimentation. It had to be cheap enough not to matter if a student damaged it or lost interest after a short period of time. Together with colleagues and friends, Eben created the Raspberry Pi Foundation (<http://www.raspberrypi.org/>).

The foundation's efforts culminated in the launch of the Raspberry Pi device on February 29, 2012. The demand was so great that it slowed the websites of the distributors (Premier Farnell and RS Components) to a crawl. The first Raspberry Pi devices were sold out within minutes of launch. I missed out on one of these initial devices but was fortunate to obtain one in early July, thanks to a donation by Andrew Frazer of Stellascapes.

The model B Raspberry Pi has a 700 MHz ARM 11 CPU, 512 MB of RAM, and an Ethernet and USB controller—all for

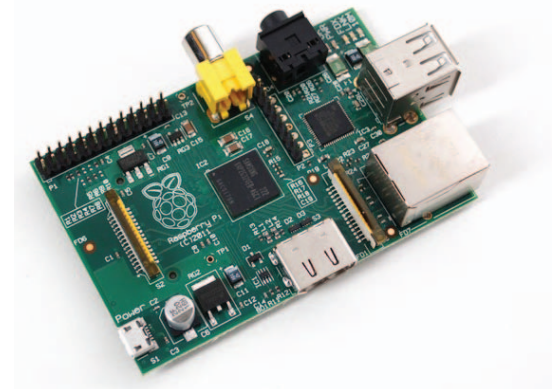


Figure 1 – A model B Raspberry Pi

just \$35 USD. To be fair, there are other single-board, embedded Linux systems available (e.g. <http://beagleboard.org/> and <http://www.carambola.cc/definition/>) that I could have used for the RDM Responder Tests. What makes the Raspberry Pi unique is the extremely low cost, which increases the popularity of the platform. A broader user base leads to more creativity and there is already an add-on DMX512 board under development for the Raspberry Pi!

## The software

Having found a suitable hardware platform, I could now focus on improving the software. Although each Raspberry Pi device has a HDMI and Composite video port, I didn't want testers to have to rely on having a monitor nearby. Recognizing that most of us have ready access to a screen in the form of our mobile phones, I decided to make use of this and build a web UI for the RDM Responder Tests.

To assist with the development, I registered the Open Lighting Project as a participant in the Google Summer of Code 2012. This global program provides university students stipends to work on an Open Source project that improves their coding skills, while also contributing to the open source community. The Open Lighting Project was one of the organizations selected for the 2012 program, and for 12 weeks during the summer our student, Ravindra Kakarla, designed and built a web interface for the RDM Responder Tests. We started beta testing the new UI with some manufacturers in late July and released it publicly in October.

The new test software can be used on Linux and Mac machines as well, but when combined with the Raspberry Pi device, it provides an easy-to-use testing solution that's hard to beat.

## Running the tests

The procedure for running the RDM Responder Tests using the Raspberry Pi is described in detail at <http://goo.gl/2g1rp>.

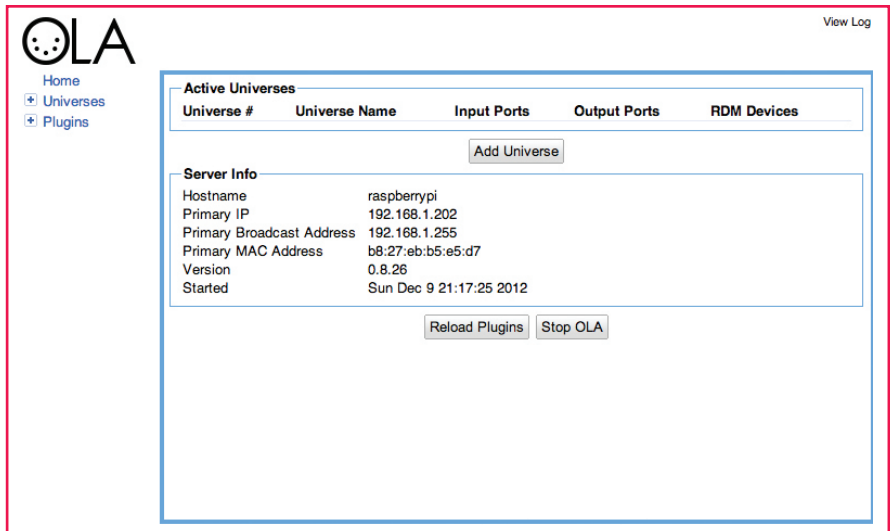


Figure 2 – The OLA interface

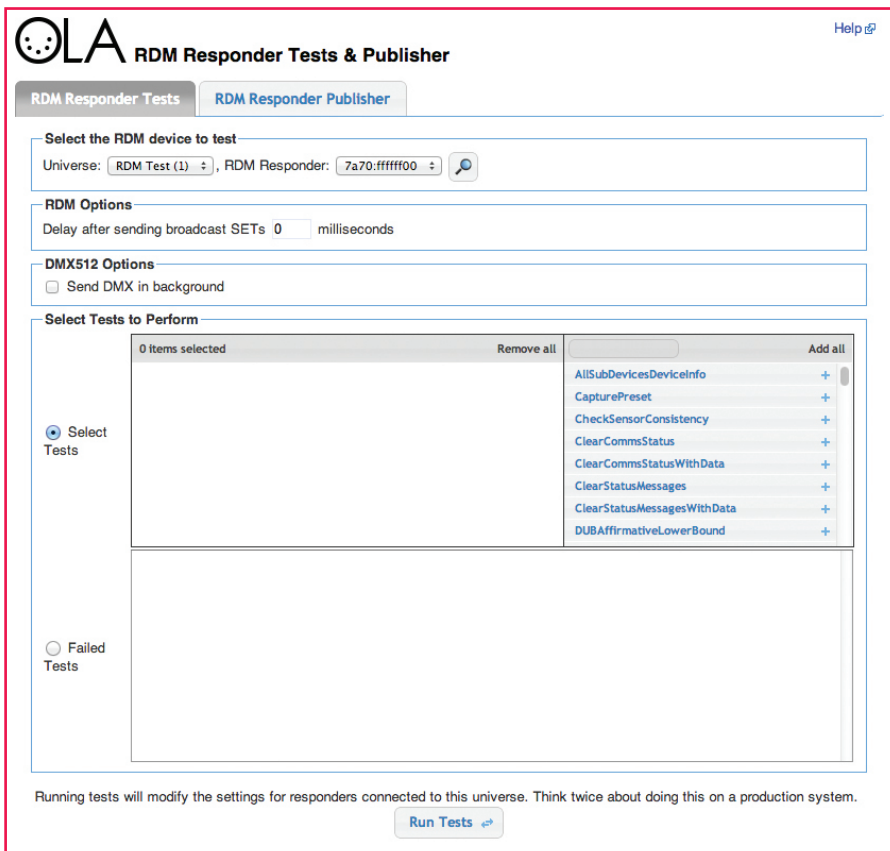


Figure 3 – The main RDM Responder Test page

I'll cover the main points here. Be sure to check with the website, as the steps may have changed slightly since this article was written.

To start the process, you'll need a Raspberry Pi device, a micro USB power

supply (most phone chargers will do), and one of the supported RDM interfaces (see <http://goo.gl/8qo3d>). Since the Raspberry Pi is limited in the amount of current it can supply through the USB ports, I strongly recommend using a powered USB hub. You

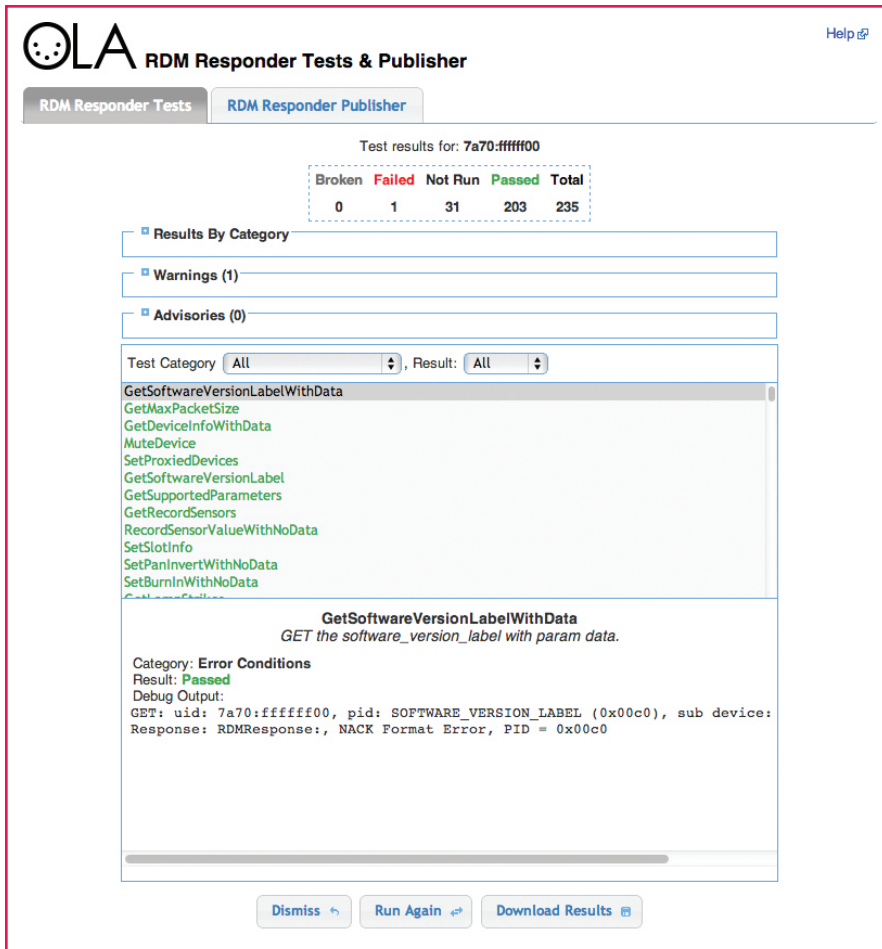


Figure 4 – The RDM Responder Test results page

## As of December 2012, the site contains more than 300 RDM products, evidence of the increasing adoption of RDM within the industry.

can buy one on Amazon for around \$20 USD. You'll also need a 4GB or greater SD card to hold the software image and a SD card reader if your computer doesn't already have one.

The software images are available at <http://goo.gl/08585>. Be sure to download the latest image. The file name should be similar to raspbian-ola-0.8.26.zip. After unzipping the archive, you will need to write the image to the SD card. The file included in the archive is similar to an ISO, so you'll need to use a special program to write it rather than the usual file copy methods. On

Windows you can use Win32DiskImager, and on Linux or Mac you can use the dd command line program. On Ubuntu, you can use the imagewriter program.

Once the copy is complete, it's a matter of plugging in the components, connecting the Pi to your local network, and applying power. The software image is configured to use DHCP, so it's best to check the local router's logs to see what IP address is assigned to the Pi. Failing this, you can always connect a monitor since the IP address of the Raspberry Pi is shown as one of the last lines during the boot process.

Now that you know the IP address, you can access the web UI and select the RDM interface you wish to use. Open a web browser, type in the IP address of the Raspberry Pi device followed by :9090. For example, if the IP address was 192.168.0.10, you need to enter <http://192.168.0.10:9090>. You should see a page similar to **Figure 2**.

From here you can click "Add Universe," select which RDM interface to use, and then assign the interface to a universe. Some RDM interfaces, such as the DMXters, need special setup, so check the website for up-to-date information.

Since people use the software for more than just RDM Responder testing, the default image doesn't automatically start the RDM Responder Test server. To complete this step, you need to ssh to the Raspberry Pi device and run `sudo dpkg-reconfigure ola-rdm-tests`. This is the only part of the new process that requires use of the command line, and we're working on eliminating it.

Now the testing can begin. Open a new browser window and connect to the RDM Responder Test UI. It's similar to the last address, only this time it uses port 9099, e.g. <http://192.168.0.10:9099>. You should see a page similar to **Figure 3**.

From this page, you can select the RDM responder to test and configure various test options. The lower part of the page allows the user to restrict the tests that are run, a useful feature if you are trying to diagnose a specific issue. If you want to run all 235 tests, click the "Add all" link on the right-hand side.

After clicking the "Run Tests" button, you'll see a results page like that in **Figure 4**.

This page displays the test results broken down by category as well as any warning or advisory messages. The tests can be filtered by category or test state (passed, failed, etc.) making it easy to see which tests didn't succeed. For each test that was run, the detailed debugging information can be viewed in the lower half of the page. It shows what RDM commands were sent and received as well as what the expected RDM responses were if the test failed. There are also options to save the results for later analysis or run the same set of tests again.

## Integration with rdm. openlighting.org and the Responder Publisher

The RDM Responder Test software is also integrated with the [rdm.openlighting.org](http://rdm.openlighting.org) site now. This site maintains information about RDM products and offers system integrators and end-users information about the features that each responder supports. As of December 2012, the site contains more than 300 RDM products, evidence of the increasing adoption of RDM within the industry.

Previously, listing RDM responders on the site was a manual process. Included in the new testing software is the RDM Responder Publisher, a tool to collect information about RDM responders and upload it to the website. This can be accessed through the “RDM Responder Publisher” tab. Simply click the “Collect Responder Information” button, and if you’re happy to make this information public after reviewing the data, click “Publish.” Once uploaded, you’ll have the option to add images and links to the manufacturer’s website for each responder. Uploaded information is held in a moderation queue before being publicly visible on the site.

If you have any questions about the new RDM Responder Tests or the Responder Publisher, please email the RDM Testing Group at <http://goo.gl/mi74x>. ■



**Simon Newton** has been interested in lighting control systems since middle school and founded the Open Lighting Project in 2004 with the aim of accelerating the adoption of new control protocols within the entertainment technology industry. He is an active member of the

Control Protocols Working Group and contributes to the RDM and RDM over IP standards. His day job finds him designing cloud computing infrastructure for an Internet company in Silicon Valley. Simon can be reached at [nomis52@gmail.com](mailto:nomis52@gmail.com) or followed at [gplus.to/nomis52](https://plus.google.com/+nomis52).