
Outernet L-band on Rasbian Documentation

Release 1.0a2

Outernet Inc <apps@outernet.is>

May 22, 2017

Contents

1 Guide contents

3

This guide shows how to deploy Outernet software on a [Raspberry Pi](#) using the stock Raspbian image, for use with Outernet's global L-band service.

Warning: This guide is a work in progress and may contain wrong and/or incomplete information. Please use the [issue tracker](#) or our [forums](#) to discuss any problems you encounter.

version	1.0a2
status	unstable/draft

Requirements

In order to follow this guide, you will need:

- Raspberry Pi
- 4GB or larger microSD card with a Raspbian image
- SDR radio dongle
- LNA
- antenna with sufficient gain
- Internet connection configured on Raspbian
- knowledge of Linux command line

The radio, LNA, and antenna, can be purchased [through Outernet](#) either individually or as a kit. You may also want to have:

- HDMI monitor
- HDMI cable
- USB or wireless keyboard
- USB or wireless mouse

This guide is written against the following version of Raspbian:

Raspbian version	May 2016
Kernel version	4.4
Download URL	http://bit.ly/raspbian-jessie-2016-05-27

Note: We will not go into the details of how the Raspbian microSD cards are created. That is left as an exercise for the reader.

Introduction

While the image is downloading, let's take a brief tour of how the L-band service works.

The files that are datacast by Outernet are encoded, modulated, and sent to several Inmarsat satellites. These satellites transmit the radio waves in the [L band frequency range](#). The waves are received by a radio on your receiver and then passed on to the software demodulator. The demodulator turns the analog signal into bits and then passes them onto the decoder, which extracts the file information from the data and reconstructs the files on local storage.

The software components involved in this process are:

- the demodulator (sdr100)
- the decoder (ondd)
- file indexer (FSAL)
- web-based UI (Librarian)

Despite all this software coming from a single vendor, they don't come as a single package for the reasons of flexibility and so that various components can be replaced by others with same or similar functionality in future. Because of this, much of this guide is going to be about ensuring that these pieces of software work together.

Note: Although these pieces of software are all part of the Outernet software eco-system, which is predominantly open-source, some of the executables are closed-source.

Virtually all of the software involved in this set-up is meant to be used as long-running background processes (a.k.a. daemons). Some of the programs already provide features that let them run as proper well-behaved daemons, while others do not. Where appropriate, we will use screen as a quick-and-dirty workaround (i.e., poor man's daemonization) solution.

Software installation

In this section, we will install all the necessary software and ensure the system is prepared to run it. The next section will discuss the programs' usage.

Preparing the device

Connect your Raspberry Pi to a router either using WiFi (Raspberry Pi 3) or LAN cable (all versions). Also plug in the SDR (software defined radio; rtl-sdr dongle), and attach the LNA and antenna. If you want to use a keyboard and a monitor, hook those up as well.

Downloading the installer kit

The installer and its files are now distributed as part of the [Outernet L-band Service on Raspberry Pi repository on GitHub](#). Download the [latest stable version](#) and extract it:

```
$ tar xvf master.tar.gz
```


Running the installer

Enter the unpacked directory and run the installer:

```
$ cd outernet-rpi-lband-master
$ sudo ./installer.sh
```

The installer will ask you a few things.

Configure udev

The radio devices are accessible only to root user by default. If you wish to run the Outernet software as a non-root user (recommended), you should answer `y` to this question. Udev will be reconfigured so that the radio device is accessible to non-root users.

Note: If you choose to have the installer reconfigure udev, you will also need to reconnect the radio for the changes to take effect.

Choose cache and storage paths

Cache path is where the decoder stores partial downloads. The storage directory is where completed downloads will be written. If you wish to customize these paths, then enter the correct paths. Otherwise, press Enter to accept the defaults.

You can also have the installer create these paths for you. When you choose this option, the directories are created, and also set to `777` permissions. You can either answer with `n` to this question, and create the paths yourself with appropriate permissions, or answer `y` to have the installer take care of it.

Install the web-based interface

The installer can install and configure the web-based interface called **Librarian**. This is not necessary to receive files, and you can always set up your own methods of accessing the files (e.g., FTP, HTTP server, etc). If you wish to try Librarian out, answer `y` to this question.

Warning: If you choose to install the web-based interface, your Raspbian install will be upgraded to Jessie testing. The process may take a while to complete, and many packages that are completely unrelated will be upgraded (e.g., Libre Office). Also note that build tools will be installed in the process. This is a requirement for some of the packages that Librarian depends on.

Uninstalling the software

To uninstall the software, run the installer and pass `uninstall` argument:

```
$ cd outernet-rpi-lband-master
$ sudo ./installer.sh uninstall
```

Note: Uninstalling does not remove downloaded files or settings.

Running the software stack

Although the individual commands can be run directly in the terminal, we will use screen so the programs can be run in the background even after we log out of the system.

Install screen

We will need to install screen before we can use it.

```
$ sudo apt-get install screen
```

Screen basics

Screen is a terminal multiplexer. Once started, you will be in a shell that does not look much different than the normal shell. The special key combination Ctrl-A can be used to give special screen-specific commands.

To create more shells (called ‘windows’), press Ctrl-A Ctrl-C. To switch between open shells, use Ctrl-A Ctrl-A (switches between the current and last used shell), or Ctrl-A Ctrl-‘ to get a list of shells to switch to.

To exit screen without closing any of the open shells, press Ctrl-A Ctrl-D. To reattach to the closed shells, run screen with `screen -rD` command.

Type `man screen` for complete usage documentation and/or use Ctrl-A Ctrl-? to get information about screen shortcuts.

Running the programs

Programs are run in this order:

- demodulator (demod)
- decoder (decoder)
- web based interface (librarian)

Let’s start screen. We will name this session ‘outernet’ so we can refer to it in future.

```
$ screen -S outernet
```

If you haven’t reconfigured udev during installation, you will need to run screen as root:

```
$ sudo screen -S outernet
```

Running the demodulator

In the first shell, let’s start the demodulator. We first need to find out the profile we should use. To do this run `demod-presets` without any arguments. Once we know the profile name (in this example, we will use ‘euraf’), we run the following command:

```
$ demod-presets euraf
```

Running the decoder

Now we need to create another shell with Ctrl-A Ctrl-C. In the new shell, we will run the decoder.

```
$ decoder
```

Starting the indexer

After creating a new shell with Ctrl-A Ctrl-C, the indexer is started using the `fsal` command:

```
$ fsal --conf /etc/fsal.ini
```

Starting the web UI

We create yet another shell with Ctrl-A Ctrl-C. In this shell, we will start Librarian:

```
$ librarian --conf /etc/librarian.ini
```

Within less than a minute, the server will start responding on the port 8080.

Detaching from the screen session

After starting all the software, we may now detach from the screen session by using the Ctrl-A Ctrl-D combination. This session will continue to run in the background as long as the device has power and is not rebooted.

At some later time, if we wish to see what the programs are doing, we can reattach to the session with `screen -rDS outernet`.