

# Table of Contents

- Building OpenWRT for a BPI-R1** ..... 1
- Introduction** ..... 1
- Disclaimer** ..... 1
- Preparation** ..... 1
- Getting the sources** ..... 2
- Updating and enabling the package feeds** ..... 2
- Configuring the build** ..... 2
- Some hints ..... 2
- Injecting files into image** ..... 3
- Building OpenWRT** ..... 3
- Obtaining the final OpenWRT image** ..... 3
- Automating this ..... 4
- Sources for the document** ..... 5



# Building OpenWRT for a BPI-R1

There are [pre-built OpenWRT](#) images available for the BPI-R1, at lot more than the 2 i linked here. But, the OpenWRT BuildRoot makes building your own release very easy.

## Introduction

This document discusses possible strategies of building a custom version of OpenWRT to use on the BPI-R1 Banana Pi Router Board. Not all tasks done here are explicitly explained, for example, setting up a Linux VM for building is far beyond the scope of this document. Also, the Banana Pi Router Board seems to be referred to by a lot of names. The ones i've found so far include: BPI-R1, Banana Pi Router Board and Lamobo R1. The official Website for the Banana Pi Router Board BPI-R1 can be found [here](#).

## Disclaimer

The contents of this document and all linked documents are subject to revision without notice due to continued progress in methodology, design, and manufacturing. The author shall have no liability for any error or damages of any kind resulting from the use of this document. There is no warranty; not even for merchantability or fitness for a particular purpose.

The information contained in these documents is supplied for domestic and private use only and is not intended as a substitute for appropriate professional advice or guidance. You shall not use this information in any unlawful manner, for any unlawful purposes. You shall not use the information to transmit any material that is defamatory, offensive or otherwise objectionable.

## Preparation

You'll want to install the OpenWRT BuildRoot on a supported system to continue. I went on using Ubuntu Linux in a VM for building. They say OSX is supported, but only after some pretty heavy tweaking, which i would not recommend doing on a system you need to work. In the end, you'll need:

- a recent Linux system, preferrably Debian or Ubuntu
- a case-sensitive filesystem, otherwise OpenWRT won't build.
- all necessary packages (Debian: `sudo apt-get install build-essential subversion libncurses5-dev zlib1g-dev gawk gcc-multilib flex git-core libssl-dev unzip`)
- If you're using a VM, ssh access is preferred over 'vm console access'. Also see [here](#) for a couple of tips.

Details on what exactly to install can be found [in the OpenWRT documentation](#).

## Getting the sources

The trunk sources can be fetched via git:

```
# git clone git://git.openwrt.org/openwrt.git
```

To fetch the latest release (15.05.1 Chaos Calmer):

```
# git clone -b chaos_calmer git://github.com/openwrt/openwrt.git
```

## Updating and enabling the package feeds

To make the standard packages as well as your custom feed available in `make menuconfig`, you'll want to do this:

```
# cd ~/openwrt
# ./scripts/feeds update -a
# ./scripts/feeds install -a
```

## Configuring the build

Now its time to start the fun part! We'll now configure OpenWRT to match our needs.

```
# make prereq && make defconfig && make menuconfig
```

Make sure there are no errors from the commands above. Adding `V=s` to any make statement will turn up verbosity and tell you whats wrong. If all went well, you'll see a screen quite similar to the Linux kernel `make menuconfig`. To establish support for the BPI-R1, you have to set at least these options:

```
Target System -> 'Allwinner A1x/A20/A3x'
Target Profile -> 'Lamobo R1'
```

Anything else can be customized to fit your special needs. Take your time to study all the available options.

### Some hints

When selecting `[M]` for a package in the OpenWRT BuildRoot `make menuconfig`, this package will be built as an `opkg` package file for later installation. When selecting `[*]`, it will additionally be installed in the final image. Use the 'm', 'y' (for \*) and 'n' (to deselect) keys.

## Injecting files into image

If you want to inject some config files (or whatever) into the final image, create a `files` folder under `~/openwrt/openwrt`, consider files as being the `/` of the image. For example, to add an `/etc/hostname` file:

```
# mkdir ~/openwrt/files
# cd ~/openwrt/files
# mkdir etc
# echo "routerboard" > ./etc/hostname
```

## Building OpenWRT

Although there are other howtos hinting to use `make` with the `-j` parameter to establish parallelism (i.e. `make -j5`), i wouldn't recommend it. I ran into some things which i considered to be race conditions while experimenting with `-j` for the OpenWRT buildroot. Not using `-j` did not produce any strange errors. Besides, theres an option in `make menuconfig` to 'build certain packages in parallel', so i suspect there could be issues. The time this would have saved is in the minutes range anyways - we're hopefully building on modern hardware!

Otherwise than that, building OpenWRT is just a matter of running

```
# cd ~/openwrt
# make
```

and then walking away to have a couple of coffees. As above, fix any errors. Use `make -j1 V=s` as a parameter to increase verbosity. While this might look strange when just 2 paragraphs above i discourage the use of `-j` for `make`, but this way you can disable aforementioned `make menuconfig` option to 'build certain packages in parallel' (thus get useful console output).

## Obtaining the final OpenWRT image

The result of the build process is in `~/openwrt/openwrt/bin/sunxi`. The image file is named something along the lines of `openwrt-sunxi-Lamobo_R1-sdcard-vfat-ext4.img`. Write this image to your sdcard like so:

```
# cd ~/openwrt/openwrt/bin/sunxi
# dd if=./openwrt-sunxi-Lamobo_R1-sdcard-vfat-ext4.img
of=/dev/your_sd_reader bs=1m; sync
```

See the [OPKG technical reference](#) for details on how to make use of the packages.

Et voila, we're done! 😊

## Automating this

You could adapt the below script to automate downloading and writing the image as well as enlarging the partition and filesystem, provided that you use a mac and have e2fsck/fsck.ext4 and e2resize installed.

```
#!/bin/bash
#
# this script fetches an openwrt disk image via ssh from my build vm,
# prompts for a target disk to write it to, writes it to said disk and
# enlarges the partition and filesystem to the maximum possible size.
#

# set to 1 to automatically resize 2nd partition after writing.
DO_RESIZE=0

IMAGE_FILE="openwrt-sunxi-Lamobo_R1-sdcard-vfat-ext4.img"
IMAGE_LOCATION="build@x.y.z.a:~/openwrt/bin/sunxi/${IMAGE_FILE}.gz"
TARGET_DISK="${1}"

clear

if [ ! -b "/dev/${TARGET_DISK}" ]; then
    echo "no such device. exiting."
    exit
fi

echo -n "cleaning up... "
rm ${IMAGE_FILE}.gz 1>/dev/null 2>/dev/null
rm ${IMAGE_FILE} 1>/dev/null 2>/dev/null
echo "done."

echo -n "retrieving new image file... "
scp ${IMAGE_LOCATION} . 1>/dev/null
if [ -f ./${IMAGE_FILE}.gz ]; then
    echo "done."
else
    echo "no image file found, check build host. exiting."
fi

echo -n "unpacking... "
gunzip ${IMAGE_FILE}.gz
echo "done."

diskutil unmountdisk ${TARGET_DISK} 1>/dev/null
echo "preparation done."

echo "writing to ${TARGET_DISK}, specify password for $(whoami) when
prompted..."
```

```
sudo dd if=${IMAGE_FILE} of=/dev/r${TARGET_DISK} bs=1m 1>/dev/null
sync
diskutil unmountdisk ${TARGET_DISK} 1>/dev/null

if [ "${DO_RESIZE}" -eq "1" ]; then
    echo -n "resizing partition... "
    OFFSET=$(sudo fdisk /dev/${TARGET_DISK} | grep "2:" | awk '{print
$11}')

cat <<EOF | sudo fdisk -e /dev/${TARGET_DISK} 1>/dev/null
e 2
83
n
20480

w
q
EOF

    echo "done."

    echo -n "running e2fsck and resize2fs on ${TARGET_DISK}s2... "
    diskutil unmountdisk ${TARGET_DISK} 1>/dev/null
    sudo e2fsck -fy /dev/${TARGET_DISK}s2 1>/dev/null
    sudo resize2fs /dev/${TARGET_DISK}s2 1>/dev/null
fi

echo "all done."
echo "check for errors above, otherwise ${TARGET_DISK} is ready to use."
```

## Sources for the document

<http://wiki.openwrt.org/doc/howto/build>, <https://dev.openwrt.org/wiki/GetSource>,  
<http://wiki.openwrt.org/doc/howto/buildvm>, <http://wiki.openwrt.org/doc/howto/build>,  
[http://linux-sunxi.org/Lamobo\\_R1](http://linux-sunxi.org/Lamobo_R1), running make help in the OpenWRT source tree and a shitload of google.

[bpi-r1](#), [openwrt](#), [routerboard](#), [linux](#)

From:  
<http://wiki.geiges.net/> - **DokuWiki**

Permanent link:  
[http://wiki.geiges.net/doku.php?id=openwrt\\_bpi-r1](http://wiki.geiges.net/doku.php?id=openwrt_bpi-r1)

Last update: **2017/10/16 07:32**

