

# Kernel

## Overview

using which Kernel?

Kernel-features-overview on [github](#)

## compile official kernel

4.4.70 on [GitHub](#)

```
sudo apt-get install git make gcc-arm-linux-gnueabi u-boot-tools
git clone https://github.com/BPI-SINOVOIP/BPI-R2-bsp.git bpi-r2
cd bpi-r2
./build.sh
```

see also [Patch for build.sh](#)

- Option 4 kernel-config (menuconfig),
- 1 to compile all,
- 3 compile only Kernel, after that option 6, to copy compiled data to SD-folder

if successful finished, Kernel is in folder SD/BPI-BOOT modules in SD/BPI-ROOT

kernel can be copied to Boot-Partition (path, previous Backup!) the modules must be copied as Root

```
cp SD/BPI-BOOT/bananapi/bpi-r2/linux/uImage /media/$USER/BPI-
BOOT/bananapi/bpi-r2/linux/
sudo cp -r SD/BPI-ROOT/lib/modules/4.4.70-BPI-R2-Kernel /media/$USER/BPI-
ROOT/lib/modules/
```

## compile my 4.4 Kernel

i have forked the official kernel-repo and updated to actual kernel-version, added mt76x2 wireless driver and activated uart0+1:

4.4.x on [GitHub](#)

steps to build/install are same except the git clone :)

```
sudo apt-get install git make gcc-arm-linux-gnueabi u-boot-tools
git clone https://github.com/frank-w/BPI-R2-4.4.git bpi-r2
cd bpi-r2
```

```
./build.sh
```

## compiled kernel 4.4.x

v1.2.1 (Nov 23, 2017), including mt76x2/x3 wireless driver

Bootloader (u-boot):

<https://drive.google.com/file/d/11XQ4n6WYSj7gGpTmwug4SITtYn0GZxOb/view?usp=sharing>

<https://drive.google.com/drive/folders/1kkFyxeHoskszl7CpSsL6Wi6ROjDrBLOB?usp=sharing>

the 4.4-archive (Kernelversion-BPI-R2-Kernel\_boot+root.tgz) contains 2 folders, which contents have to be copied to the partiton of SD/MMC with the same name.

- BPI-BOOT ⇒ boot-Partition ulmage=Kernel-file, backup first your existing before overwriting it
- BPI-ROOT ⇒ partition with root-fs, here are the kernel-modules (/lib/modules/kernelname). With new kernel-version a new subfolder is created, so you need no backup else the name of kernel is the same.

## using own (newer) Kernel

<http://forum.banana-pi.org/t/what-s-the-best-practice-to-build-own-kernel/3937>

## using my github-repo

<https://github.com/frank-w/BPI-R2-4.14.git>

## manual way using kernel.org

```
sudo apt-get install git make gcc-arm-linux-gnueabi g++-arm-linux-gnueabi u-boot-tools
git clone https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git bpi-r2
cd bpi-r2

export CROSS_COMPILE=arm-linux-gnueabi-
export ARCH=arm
export INSTALL_MOD_PATH=$(pwd)/mod/
mkdir $INSTALL_MOD_PATH

#git reset --hard v4.14
cp pfad/zu/mt7623n_evb_bpi_defconfig arch/arm/configs/
make mt7623n_evb_bpi_defconfig
make

...
```

```
#cat arch/arm/boot/zImage arch/arm/boot/dts/mt7623n-bananapi-bpi-r2.dtb
>zImage_dtb
cat arch/arm/boot/zImage arch/arm/boot/dts/mt7623n-bananapi-bpi-r2.dtb >
arch/arm/boot/zImage-dtb
#mkimage -A arm -O linux -C none -T kernel -a 0x80008000 -e 0x80008000 -d
zImage_dtb uImage
mkimage -A arm -O linux -T kernel -C none -a 80008000 -e 80008000 -n
"Linux Kernel 4.14" -d arch/arm/boot/zImage-dtb ./uImage

rm -r $INSTALL_MOD_PATH/lib
make modules_install

cp uImage /media/$USER/BPI-B00T/bananapi/bpi-r2/linux/ #<< change
mountpoint
sudo cp -r $INSTALL_MOD_PATH/lib /media/$USER/BPI-R00T/ #<< change
mountpoint
```

config from user RyderLee  
ramdisk

pcie.patch

for PCIe-Slot (CONFIG\_PCIEPORTBUS,CONFIG\_PCIEPORTBUS,CONFIG\_PCIEPORTBUS have to be activated)

Script to compile/copy  
changed defconfig

above 4.14.14 only pcie-patch may be needed..

with kernel 4.4.70 mmc0=sd-card and mmc1=emmc, so i have done this also for kernel 4.14 (else you have to change u-boots uenv.txt and Mount-Points in system)

to do this set mmc1-section above mmc0-section arch/arm/boot/dts/mt7623n-bananapi-bpi-r2.dts and in arch/arm/boot/dts/mt7623.dtsi

mmc-swap.diff

## how to use the build.sh

with my [github-Repo](#):

```
sudo apt-get install git make gcc-arm-linux-gnueabi
git clone https://github.com/frank-w/BPI-R2-4.14.git bpi-r2-4.14
cd bpi-r2-4.14
./build.sh importconfig
./build.sh config #if you want to change the kernel-config
./build.sh
```

## compiled 4.14.x

<https://drive.google.com/drive/folders/1EGN1TvqCpDHdOAS-mjRg9ipi0kahnOUV?usp=sharing>

the 4.14-archive (bpi-r2\_kernelversion\_gitbranch.tar.gz) contains 2 folders, which contents have to be copied to the partition of SD/MMC with the same name.

- BPI-BOOT ⇒ boot-Partition, ulmage=Kernel-file, backup first your existing ulmage before overwriting it
- BPI-ROOT ⇒ partition with root-fs, here are the kernel-modules (/lib/modules/kernelname). With new kernel-version a new subfolder is created, so you need no backup else the name of kernel is the same. The files in the other folders are mainly to get internal wifi working (wmt-tools, firmware, config) and have not to be copied every time (currently not changing). These are only packed in for users who haven't these tools already

there are also binary-packages on github-releases (made by travis-ci) with same file-structure:

<https://github.com/frank-w/BPI-R2-4.14/releases>

[you can use this](#)

script

for installing the tar.gz-package

## install kernel

The linux-kernel has 2 parts...the „main-kernel-image“ (ulmage) and the modules.

The kernel-image is loaded by uboot from BPI-BOOT-Partition folder bananapi/bpi-r2/linux/. By default the file „ulmage“ is loaded, but you can choose another name and set variable „kernel“ in uEnv.txt.

the modules have to be put to BPI-ROOT-partition (/lib/modules/kernelname).

## change DTS (device tree)

using dts(i)-nodes (arch/arm/boot/dts/) and porting them to older kernel

- [mt7623.dtsi](#)
- [mt7623n-bananapi-bpi-r2.dts](#) ⇒ [mt7623n-bpi-r2.dts](#)

## issues

- evbug will be loaded and displays messages on terminal if local keyboard is connected
  - you can disable via „modprobe -r evbug“ (temporary, will be loaded again after next reboot) or add file /etc/modprobe.d/evbug.conf with content „blacklist evbug“ (survives reboot)

From:

<https://wiki.fw-web.de/> - **FW-WEB Wiki**

Permanent link:

<https://wiki.fw-web.de/doku.php?id=en:bpi-r2:kernel>

Last update: **2023/06/08 17:06**

