

# Kernel

## Übersicht

welchen Kernel nehmen?

Überblick der Kernel-features auf [github](#)

## offiziellen Kernel kompilieren

4.4.70 auf [GitHub](#)

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

siehe auch [Patch für build.sh](#)

- Option 4 für Einstellungen am Kernel,
- 1 um alles zu kompilieren,
- 3 um nur den Kernel zu kompilieren, Option 6 danach, um die kompilierten Daten in den SD-Ordner zu kopieren

wenn alles durchlaufen ist, ist der Kernel im Ordner SD/BPI-BOOT und die Module in SD/BPI-ROOT

der Kernel kann direkt auf die Boot-Partition geschrieben werden (Pfad beachten, vorher Backup!) die Module müssen als Root rüberkopiert werden

```
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/
```

## meinen 4.4er Kernel kompilieren

4.4.x auf [GitHub](#)

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

Installation wie beim offiziellen Kernel

## kompilierter Kernel 4.4.x

v1.2.1 (23. Nov 2017), incl. mt76x2/x3 WLAN-Treiber

Bootloader (u-boot):

<https://drive.google.com/file/d/11XQ4n6WYSj7gGPtMwug4SITtYn0GZxOb/view?usp=sharing>

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

das 4.4-Archiv (Kernelversion-BPI-R2-Kernel\_boot+root.tgz) enthält 2 Ordner, deren Inhalt auf die SD/MMC mit dem selben Namen kopiert werden müssen.

- BPI-BOOT ⇒ boot-Partition, ulmage=Kernel-file, backup first your existing ulmage before overwriting it
- BPI-ROOT ⇒ partition mit dem Betriebssystem, hier liegen die Kernel-Module (/lib/modules/kernelname). Mit jedem neuen Kernel wird ein neuer Ordner angelegt, so dass ein Backup nicht unbedingt nötig ist, außer es ist die selbe Kernelversion.

## eigenen (neueren) Kernel verwenden

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

Patches, die noch nicht im Mainline-Kernel sind:

<https://patchwork.kernel.org/project/linux-mediatek/list/>

## mein github-repo verwenden

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

## standard-kernel-Repo verwenden + patches

defconfig von [GitHub](#) nach arch/arm/configs/ kopieren

defconfig von [GitHub](#) importiert und angepasst

Dateiendung .txt entfernen

```
sudo apt-get install git make gcc-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-rc5
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/ #<< mountpunkt
anpassen
sudo cp -r $INSTALL_MOD_PATH/lib /media/$USER/BPI-R00T/ #<< mountpunkt
anpassen
```

für Ramdisk-version:

config from user RyderLee  
Ramdisk

pcie.patch

damit der PCIe-Slot (wenn CONFIG\_PCIEPORTBUS,CONFIG\_PCIEPORTBUS,CONFIG\_PCIEPORTBUS  
aktiviert)

Script zum kompilieren/kopieren  
angepasste defconfig

da beim 4.4.70 mmc0=sd-karte und mmc1=emmc habe ich dieses auch beim 4.14er Kernel  
durchgeführt (sonst muss man u-boot und ggf. die Mount-Points im System permanent ändern)

einfach in der arch/arm/boot/dts/mt7623n-bananapi-bpi-r2.dts und der arch/arm/boot/dts/mt7623.dtsi  
den mmc1-Block über den mmc0-Block setzen

mmc-swap.diff

## wie build.sh verwenden

1. cd in das Kernel-Verzeichnis
  1. git clone <https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git/>
  2. oder git pull/git fetch

2. letztes „tag“ von 4.14: „git tag|grep 4.14“ #evtl. aktuelle Änderung sichern: „git diff > file.diff“ (auch aktuelle Revision merken)
3. „git reset -hard v4.14“ #« tag ändern wenn neueres existiert (löscht alle Änderungen!)
  1. alternativ git stash,git checkout tags/4.14.x,git stash apply
4. die
 

defconfig

und Patch (

mmc\_pcie.patch

) in das Kernel-Verzeichnis kopieren (über „linux“)
5. das
 

Script zum kompilieren/kopieren

in das Verzeichnis „linux“ innerhalb des Kernel-Verzeichnisses kopieren.
6. dann in „linux“ reingehen und folgendes ausführen

```
patch -p1 < ../mmc_pcie.patch
./build.sh importconfig #mt7623n_evb_fw_defconfig (.txt entfernt) muss in
übergeordnetem Verzeichnis existieren
./build.sh config #starte menuconfig
./build.sh #starte make & Kopiervorgang
```

## kompilierter 4.14.x

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

das 4.14-Archiv (bpi-r2\_kernelversion\_gitbranch.tar.gz) enthält 2 Ordner, deren Inhalt auf die SD/MMC mit dem selben Namen kopiert werden müssen.

- BPI-BOOT ⇒ boot-Partition, ulmage=Kernel-file, backup first your existing ulmage before overwriting it
- BPI-ROOT ⇒ partition mit dem Betriebssystem, hier liegen die Kernel-Module (/lib/modules/kernelname). Mit jedem neuen Kernel wird ein neuer Ordner angelegt, so dass ein Backup nicht unbedingt nötig ist, außer es ist die selbe Kernelversion. Die Dateien in den anderen Ordnern sind großteils für den internen WLAN-Chip (wmt-tools, firmware, config) und müssen nicht jedesmal kopiert werden (ändern sich kaum bis gar nicht). Diese sind nur mit vorhanden für Personen, welche sich ein Image selbst bauen.

## DTS ändern

dts(i)-Definition aus neuerem Kernel (arch/arm/boot/dts/) in älteren einbauen

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

## debugging

- <https://kernelnewbies.org/FAQ/LinuxKernelDebug101>
- [https://elinux.org/Debugging\\_by\\_printing](https://elinux.org/Debugging_by_printing)

- <https://www.kernel.org/doc/Documentation/printk-formats.txt>

## Links

- [Linux Linux-Stable](#)
- [Elixir Cross Referencer](#)
- [MTK-Patchwork MTK-Archive](#)
- [Net Net-Next Netdev-ML-Archive](#)

From:

<http://www.fw-web.de/dokuwiki/> - **FW-WEB Wiki**

Permanent link:

<http://www.fw-web.de/dokuwiki/doku.php?id=bpi-r2:kernel>

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

