

BananaPi R3 Mini

https://wiki.banana-pi.org/Banana_Pi_BPI-R3_Mini

Hardware

- MT7986(Filogic 830) quad core ARM A53 SoC
- 2G DDR RAM
- 8G eMMC flash, SPI-NAND (128 MiB)
- 2x 2.5GbE network port (airoha en8811h phy)
- dual-band wifi:
 - 2.4G
 - 5G



debug-uart

connections from the edge of board: GND,RX,TX (remember RX/TX swap on your adapter :))

Software

my R3mini came preinstalled with vendors openwrt (21.02-SNAPSHOT, Linux version 5.4.171) on emmc and nand. i suggest to leave the nand to have always a rescue-system and only flash emmc as productive system with a more recent version.

It will take some time till i can release a full emmc-image for R3 Mini, but my uboot-repo generates already a very basic bpi-r3 emmc image (Releases Page) without rootfs/kernel which can be used on r3mini as well. Userspace can generated with the buildchroot.sh of my BPI-Images Repo or extracted from a full r3 Image.

Step-by-step:

used usb-stick to get files accessible from r3mini

started with r3-img:

```
gunzip -c /mnt/sda2/bpi-r3_bookworm_nokernel.img.gz |dd bs=1024
conv=notrunc,fsync of=/dev/mmcblk0
```

then flash uboot (wonder why there is no mmcblk0boot0 block to flash the bl2 - seems to be a problem with vendor-image, in my linux i see the boot0):

```
#r3mini/bl2_r3mini_emmc_pcie.img to boot0 conv=notrunc,fsync
dd if=/mnt/sda1/r3mini/fip_r3mini_emmc_pcie.bin bs=1024 conv=notrunc,fsync
of=/dev/mmcblk0p4
```

booted original openwrt from nand....

```
root@OpenWrt:/mnt/r3mini# dd of=/dev/mmcblk0p4 if=/mnt/r3mini/bpi-
r3mini_emmc_fip.bin
631+1 records in
631+1 records out
root@OpenWrt:/mnt/r3mini# dd of=/dev/mmcblk0boot0 if=/mnt/r3mini/bpi-
r3mini_emmc_bl2.img
dd: error writing '/dev/mmcblk0boot0': Operation not permitted
1+0 records in
0+0 records out
root@OpenWrt:/mnt/r3mini# echo 0 > /sys/block/mmcblk0boot0/force_ro
root@OpenWrt:/mnt/r3mini# dd of=/dev/mmcblk0boot0 if=/mnt/r3mini/bpi-
r3mini_emmc_bl2.img
392+1 records in
392+1 records out
root@OpenWrt:/mnt/r3mini#
```

Kernel (creates the bpi-r3.itb with conf-emmc-mini bootconfig):

<https://github.com/frank-w/BPI-Router-Linux/tree/6.6-r3mini>

need to drop the „run setbootconf;“ from newboot first

```
BPI-R3M> run useusb
BPI-R3M> setenv initrd rootfs.cpio.zst
BPI-R3M> setenv fit r3mini/bpi-r3.itb
BPI-R3M> setenv bootconf "#conf-emmc-mini"
BPI-R3M> run newboot
```

how to get available bootconfigs from binary fit image:

```
dumpimage -l /media/frank/Android-x86/r3mini/bpi-r3.itb
```

thermal

```
root@bpi-r3-mini:~# cat /sys/class/thermal/thermal_zone0/temp
48408
```

manipulating trip points...this lets the fan run at 45°C silently and faster at 55°C (Case for R3mini gets really hot)

```
root@bpi-r3-mini:~# echo 45000 >
/sys/class/thermal/thermal_zone0/trip_point_3_temp
root@bpi-r3-mini:~# echo 55000 >
```

```
/sys/class/thermal/thermal_zone0/trip_point_2_temp
```

my kernel

i merged R3Mini support to my 6.6-main

<https://github.com/frank-w/BPI-Router-Linux/tree/6.6-main>

vendor-kernel

DTS for reference:

<https://github.com/BPI-SINOVOIP/BPI-R3MINI-OPENWRT-V21.02.3/blob/main/target/linux/mediatek/files-5.4/arch/arm64/boot/dts/mediatek/mt7986a-bananapi-bpi-r3mini-emmc.dts>

chainload uboot

```
BPI-R3M> fatload usb 0:1 $loadaddr r3mini/u-boot-pcie.bin
815592 bytes read in 73 ms (10.7 MiB/s)
BPI-R3M> go $loadaddr
## Starting application at 0x46000000 ...
U-Boot 2024.01-bpi-r3mini-00011-g5aadabca11e5-dirty
```

pcie/nvme

working in my r3mini uboot (2024-01-bpi-r3mini)

```
BPI-R3M> pci enum
drivers/pci/pcie_mEDIATEK_gen3.c:mtk_pcie_startup_port[261] detected a card
set trans table 0: 0x20000000 0x20000000, 0x10000000
BPI-R3M> pci
BusDevFun  VendorId  DeviceId  Device Class      Sub-Class
-----
00.00.00   0x14c3   0x1f32   Bridge device      0x04
01.00.00   0x1c5c   0x1327   Mass storage controller 0x08
BPI-R3M> nvme scan
BPI-R3M>
BPI-R3M> nvme info
Device 0: Vendor: 0x1c5c Rev: 80002C00 Prod: ND94N163610404F0R
          Type: Hard Disk
          Capacity: 244198.3 MB = 238.4 GB (500118192 x 512)
BPI-R3M>
```

network support

only working once then network is stalled

```
BPI-R3M> setenv ipaddr 192.168.90.2
BPI-R3M> ping 192.168.90.1
Using ethernet@15100000 device
host 192.168.90.1 is alive
BPI-R3M> ping 192.168.90.1
Using ethernet@15100000 device

ARP Retry count exceeded; starting again
ping failed; host 192.168.90.1 is not alive
BPI-R3M>
```

This is fixed in my 2024-04 uboot by always run autonegotiation code (genphy code seems not working here).

Writing firmware to emmc boot1:

```
usb start
mmc partconf 0 1 2 2
mmc erase 0x0 0x120
load usb 0:1 $loadaddr EthMD32.dm.bin
mmc write $loadaddr 0x0 0x20
load usb 0:1 $loadaddr EthMD32.DSP.bin
mmc write $loadaddr 0x20 0x100
mmc partconf 0 1 1 0
```

daniel wrote it here: <https://forum.banana-pi.org/t/bpi-r3-mini-boot-from-nand-fails/17340/27>

Linux way:

```
echo 0 > /sys/block/mmcblk0boot1/force_ro
dd if=/lib/firmware/airoha/EthMD32.dm.bin of=/dev/mmcblk0boot1
dd if=/lib/firmware/airoha/EthMD32.DSP.bin bs=16384 seek=1
of=/dev/mmcblk0boot1
```

checking content of boot1 is same as concatenated firmware-files:

```
# cat /dev/mmcblk0boot1 > airoha_boot1.bin
# cat /mnt/r3mini/EthMD32.dm.bin /mnt/r3mini/EthMD32.DSP.bin >
airoha_usb.bin
# ls -l airoha_usb.bin
-rw-r--r-- 1 root root 147456 Jan  1 00:03 airoha_usb.bin
# ls -l airoha_boot1.bin
-rw-r--r-- 1 root root 4194304 Jan  1 00:00 airoha_boot1.bin
# truncate --size=147456 airoha_boot1.bin
# ls -l airoha_boot1.bin
```

```
-rw-r--r-- 1 root root 147456 Jan  1 00:05 airoha_boot1.bin
# diff -u airoha_boot1.bin airoha_usb.bin
#
```

nand install

openwrt (file could be transferred also via usb):

<https://forum.banana-pi.org/t/bpi-r3-mini-boot-from-nand-fails/17340/27>

Fix bricked boot

In case nand and emmc are broken there is a way to load uboot via uart for reflash/fix bootchain of the storages.

<https://forum.banana-pi.org/t/bpi-r3-mini-boot-from-nand-fails/17340/24>

files here:

https://drive.google.com/drive/folders/1b8KWe0p7fZmsRs9eBZUJBBISv3r2oUOR?usp=drive_link

```
$ ./mtk_uartboot -s /dev/ttyUSB5 --aarch64 --payload bpi-r3mini_ram_bl2.bin
--fip bpi-r3mini_ram_fip.bin
mtk_uartboot - 0.1.1
Using serial port: /dev/ttyUSB5
Handshake...
hw code: 0x7986
hw sub code: 0x8a00
hw ver: 0xca01
sw ver: 0x1
Baud rate set to 460800
sending payload to 0x201000...
Checksum: 0x3663
Setting baudrate back to 115200
Jumping to 0x201000 in aarch64...
Waiting for BL2. Message below:
=====
NOTICE: BL2: v2.9(release):v2.9.0-357-g553a16af808e ram
NOTICE: BL2: Built : 17:17:10, Mar  3 2024
NOTICE: WDT: Cold boot
NOTICE: WDT: disabled
NOTICE: CPU: MT7986 (1998MHz)
NOTICE: EMI: Using DDR4 settings
NOTICE: EMI: Detected DRAM size: 2048MB
NOTICE: EMI: complex R/W mem test passed
NOTICE: Starting UART download handshake ...
=====
BL2 UART DL version: 0x10
Baudrate set to: 921600
FIP sent.
```

```
=====
NOTICE:  Received FIP 0x4d695 @ 0x40400000 ...
=====
```

now you can use minicom or similar tool to get uboot-console and reflash bl2/fip to nand/emmc

From:

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

Permanent link:

<https://wiki.fw-web.de/doku.php?id=en:bpi-r3mini:start>

Last update: **2024/04/21 19:19**

