

# U-Boot

On this page i try to summarize the information i collected across my boards... it needs a bit of time to add all into this page...

My U-Boot source (with prebuilt binaries in releases):

<https://github.com/frank-w/u-boot/>

## board specific pages

- [BPI-R2](#)
- [BPI-R64](#)
- [BPI-R2Pro](#)
- [BPI-R3](#)
- [BPI-R3mini](#)
- [BPI-R4](#)

my u-boot have builtin environment created by the uEnv\_r\*.txt which can be overridden by an uEnv.txt file when kernel-files (r2/r64: bananapi/bpi-r\*/linux, all others in root) are located on mmc BPI-BOOT partition.

## load/write data

listing files (mmc=subsystem, 0=device, 5=partition, folder optional):

```
ls mmc 0:5 folder
```

basically load works from fat partitions with the following syntax (ext4 should work too, but with [ext4load](#))...

loading files (mmc=subsystem, 0=device, 5=partition):

```
fatload mmc 0:5 $loadaddr filename
```

most complete page for writing to a device (including nand/nor) is [BPI-R3](#)

## mmc

all BPI-Router boards (R3+R4 have only 1 mmc controller so it is based on hardware switches which is available)

```
U-Boot> mmc list
mmc@11230000: 0 (eMMC)
mmc@11240000: 1 (SD)
```

```
#set mmc-device
U-Boot> mmc dev 1

#read current device
U-Boot> mmc dev
switch to partitions #0, OK
mmc1 is current device

U-Boot> mmcinfo
Device: mmc@11240000
Manufacturer ID: 1b
OEM: 534d
Name: 00000
Bus Speed: 50000000
Mode : SD High Speed (50MHz)
Rd Block Len: 512
SD version 2.0
High Capacity: Yes
Capacity: 7.6 GiB
Bus Width: 4-bit
Erase Group Size: 512 Bytes

ls mmc 0:1
```

loading files (0=device, 5=partition):

```
fatload mmc 0:5 $loadaddr filename
```

## eMMC

```
#partitionconfig
#mmc partconf dev [boot_ack boot_partition partition_access]
# - Show or change the bits of the PARTITION_CONFIG field of the specified
device
#example for mode 0x48 (needed for emmc-boot on bpi-r2)
U-Boot> mmc partconf 0
EXT_CSD[179], PARTITION_CONFIG:
BOOT_ACK: 0x1
BOOT_PARTITION_ENABLE: 0x1
PARTITION_ACCESS: 0x0

#set via
U-Boot> mmc partconf 0 1 1 0
```

## usb

all BPI-Router boards

```

BPI-R2> usb start
starting USB...
Bus usb@1a1c0000: hcd: 0x1a1c0000, ippc: 0x1a1c4700
u2p:1, u3p:1
Register 200010f NbrPorts 2
Starting the controller
USB XHCI 0.96
Bus usb@1a240000: hcd: 0x1a240000, ippc: 0x1a244700
u2p:1, u3p:1
Register 200010f NbrPorts 2
Starting the controller
USB XHCI 0.96
scanning bus usb@1a1c0000 for devices... 1 USB Device(s) found
scanning bus usb@1a240000 for devices... 2 USB Device(s) found
    scanning usb for storage devices... 1 Storage Device(s) found
BPI-R2> usb tree
USB device tree:
  1  Hub (5 Gb/s, 0mA)
      U-Boot XHCI Host Controller

  1  Hub (5 Gb/s, 0mA)
    |  U-Boot XHCI Host Controller
    |
+-2  Mass Storage (480 Mb/s, 200mA)
      USB      Flash Disk      906B030002F4

BPI-R2> ls usb 0:1
          efi/
4767728  kernel

1 file(s), 1 dir(s)

```

loading files:

```
fatload usb 0:1 $loadaddr filename
```

## sata

BPI-R2,BPI-R64 (,BPI-R2Pro)

```

BPI-R2> pci enum
BPI-R2> pci 0
Scanning PCI devices on bus 0
BusDevFun  VendorId  DeviceId  Device Class      Sub-Class
-----
00.00.00   0x14c3    0x0801   Bridge device     0x04
00.01.00   0x14c3    0x0801   Bridge device     0x04
BPI-R2> pci 1
Scanning PCI devices on bus 1

```

| BusDevFun | VendorId | DeviceId | Device Class       | Sub-Class |
|-----------|----------|----------|--------------------|-----------|
| 01.00.00  | 0x14c3   | 0x7612   | Network controller | 0x80      |

```
BPI-R2> pci 2
Scanning PCI devices on bus 2
```

| BusDevFun | VendorId | DeviceId | Device Class            | Sub-Class |
|-----------|----------|----------|-------------------------|-----------|
| 02.00.00  | 0x1b21   | 0x0611   | Mass storage controller | 0x01      |

```
BPI-R2> scsi scan
scanning bus for devices...
SATA link 0 timeout.
Target spinup took 0 ms.
AHCI 0001.0200 32 slots 2 ports 6 Gbps 0x3 impl SATA mode
flags: 64bit ncq stag led clo pmp pio slum part ccc sxs
  Device 0: (1:0) Vendor: ATA Prod.: ST750LM022 HN-M7 Rev: 2AR1
           Type: Hard Disk
           Capacity: 715404.8 MB = 698.6 GB (1465149168 x 512)
BPI-R2> ls scsi 0:1
```

## nvme

BPI-R3(mini), BPI-R4

```
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> nvme scan
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>
```

## tftp

network support is basicly on all boards possible with my uboot

- r2pro only wan-port (mtk switch driver not compatible with rockchip mac driver in uboot, but wan is directly connected to RJ45 jack)
- r3mini only lan-port (only 1 phy can be defined)
- R3/R4 only RJ45 ethernet ports (no SFP)

## netboot

```
printenv ipaddr
setenv ipaddr 192.168.0.19

tftp ${loadaddr} 192.168.0.10:bpi-router.itb
```

IP-adress of server (including the : ) can be omitted, if serverip is set correctly

## booting

if the file is at \$loadaddr you use it

### uboot.bin

```
go $loadaddr
```

## Kernel-Image

BPI-R2 (armhf) uses ulmage with appended devicetree, all other boards are using FIT image

all can be booted with bootm, FIT have bootconfig where 1 can be set as default, so this is optional

```
bootm $loadaddr#config
```

available configs in a FIT can be obtained on host-linux with uboot-tools installed (or look in its file when kernel is self-compiled)

```
dumpimage -l bpi-r3.itb
```

From:

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

Permanent link:

<https://www.fw-web.de/dokuwiki/doku.php?id=en:uboot>

Last update: **2024/04/07 13:07**

