Storage

Partition		SD card	eMMC	Update(BPI-Tool)	Update(RAW command)
BootLoader	Header	0-2KB	Boot0: 0-2KB	SD (On HOST PC or target board) : *bpi-bootsel BPI-R2-720P-2k.img.gz /dev/mmcblkX* (for updating both uboot and preloader) eMMC(Only on target board): *echo 0 > /sys/block/mmcblkYboot0/force_ro* *bpi-bootsel BPI-R2-EMMC-boot0-DDR1600-0k-0905.img.gz /dev/mmcblkyboot0* (for updating bootloader to EMMC boot0) *bpi-bootsel BPI-R2-720P-2k.img.gz /dev/mmcblkY* (for updating Uboot to eMMC user block)	We just have RAW command to update EMMC preloader under uboot: *emmc write 1 84000000 0 200*
	Preloader	2-320KB	Boot0: 0-320KB		
	Uboot	320KB - 1MB	User Data Area: 320KB - 1MB		Write Uboot to SD or EMMC under uboot: mmc dev Z mmc write 84000000 280 200 Where Z is the number of disk, 0-emmc, 1-SD
	Reserved	1MB - 100MB	User Data Area: 1MB - 100MB		
File System Partition1(FAT32)		100MB - 356MB	User Data Area:100MB - 356MB		
File System Partition2(EXT4)		356MB - 7456MB	User Data Area:356MB - 7456MB	cd SD;bpi-update -c bpi-r2.conf -d /dev/<device>	

image by garywang

it looks like /dev/mmcblk1, /dev/mmcblk1boot0 and /dev/mmcblk1boot1 are independend devices (bootx not partitions inside /dev/mmcblk1)

preloader

for SD

have to be written at 2k-offset (0x800)

```
sudo dd if=BPI-R2-EMMC-boot0-DDR1600-20190722-2k.img of=/dev/sdc bs=1k seek=2
```

(SD-Card needs additional MMC_BOOT & BRLYT header, see below)

for EMMC

have to be written at 0-offset (0x0) of boot0-Partition

```
sudo dd if=BPI-R2-EMMC-boot0-DDR1600-20190722-0k.img of=/dev/mmcblk1boot0
```

new preloaderfiles from here: https://github.com/BPI-SINOVOIP/BPI-files/tree/master/SD/100MB

SD-Card

sdcard-bootsektor reverse-engineering
http://forum.banana-pi.org/t/boot-fails-with-self-build-u-boot/5460/20

http://forum.banana-pi.org/t/how-to-build-an-ubuntu-debian-sd-image-from-scratch/6805/8

SD-Headers

bpi-r2-head440-0k.img
bpi-r2-head1-512b.img

- SDMMC_BOOT-Signatur + address of 2nd header (0x00000200) - first 440 byte (before

partitiontable):

```
gunzip -c BPI-R2-HEAD440-0k.img.gz | dd of=/dev/loop8 bs=1024 seek=0
```

- BRLYT-signature + preloader-Address (0x00000800):

```
gunzip -c BPI-R2-HEAD1-512b.img.gz | dd of=/dev/loop8 bs=512 seek=1
```

short install-guide

this guide is derived from user ul90 but using a fixed partition table (existing image/sdcard) and variables for loopdevice

```
dd if=/dev/zero of=../bpi-r2-buster.img bs=1M count=7168
loopdev=$(losetup -f)
sudo losetup ${loopdev} ../bpi-r2-buster.img
echo $loopdev
sudo dd if=~/Downloads/BPI-R2-preloader-DDR1600-20190722-2k.img
of=${loopdev} bs=1k seek=2
sudo dd if=~/Downloads/BPI-R2-HEAD440-0k.img of=${loopdev} bs=1024 seek=0
sudo dd if=~/Downloads/BPI-R2-HEAD1-512b.img  bs=512 seek=1
sudo dd if=/path/to/u-boot/u-boot.bin of=${loopdev} bs=1k seek=320
sudo sfdisk ${loopdev} < ~/Downloads/parttable.dat
sudo partprobe ${loopdev}
ls ${loopdev}*
sudo mkfs -t vfat ${loopdev}p1
sudo mkfs -t ext4 ${loopdev}p2
sudo fatlabel ${loopdev}p1 BPI-BOOT
sudo e2label ${loopdev}p2 BPI-ROOT
```

install debian (from bootstrapped rootfs)

```
sudo mount ${loopdev}p2 /mnt/
sudo cp -r debian_buster_armhf/. /mnt/
#install kernel-modules to same partition
kernelpack=/path/to/bpi-r2_<version>_main.tar.gz
sudo tar -xzf ${kernelpack} -C /mnt/ --strip-components=1 BPI-ROOT
#install kernel to boot-partition
sudo umount /mnt
sudo mount ${loopdev}p1 /mnt/
sudo tar -xzf ${kernelpack} -C /mnt/ --strip-components=1 BPI-BOOT
#maybe create a uEnv.txt
sudo umount /mnt
```

```
sudo losetup -d ${loopdev}
#now write the image to card (make sure /dev/sdc is your sdcard-device and
no partition is mounted)
sudo dd if=../bpi-r2-buster.img of=/dev/sdc
sync
```

Step-by-Step-Guide from "ul90"

1. download binary files:

SD-Headers (old not supporting full uboot-size)

2. Create image file (8GB):

```
dd if=/dev/zero bs=1M count=7296 | pv | dd of=bpir2.img
```

3. Load image as virtual drive:

```
losetup /dev/loop8 bpir2.img
```

4. Make partitions and format:

```
parted -s /dev/loop8 mklabel msdos
parted -s /dev/loop8 unit MiB mkpart primary fat32 -- 100MiB 356MiB
parted -s /dev/loop8 unit MiB mkpart primary ext2 -- 356MiB 7295MiB
partprobe /dev/loop8
mkfs.vfat /dev/loop8p1 -I -n BPI-BOOT
mkfs.ext4 -O ^has_journal -E stride=2,stripe-width=1024 -b 4096 /dev/loop8p2
-L BPI-ROOT
sync
parted -s /dev/loop8 print
```

5. extended boot-headers

- SDMMC_BOOT-signature + address of 2nd header 0x00000200 (before partition-table):

```
gunzip -c BPI-R2-HEAD440-0k.img.gz | dd of=/dev/loop8 bs=1024 seek=0
```

- BRLYT-signature + preloader-address (0x00000800):

```
gunzip -c BPI-R2-HEAD1-512b.img.gz | dd of=/dev/loop8 bs=512 seek=1
```

6. Write preloader and u-boot bootloader:

```
dd if=preloader_iotg7623Np1_sd_1600M.bin of=/dev/loop8 bs=1024 seek=2
dd if=u-boot.bin of=/dev/loop8 bs=1024 seek=320
sync
```

7. Copy rootfs + kernel

create bootstraped rootfs

- Ubuntu
- Debian

Install Kernel:

Kernel

8. Remove loop device:

```
losetup -d /dev/loop8
```

MMC-Utils

with the mmc-utils you can test out of a running system, if the EMMC-partitioning is correct (should be 48 see change_partition-configuration_of_emmc).

```
./mmc extcsd read /dev/mmcblk1
....
Boot configuration bytes [PARTITION_CONFIG: 0x48]
....
```

i have added the mmc-utils also to my Kernel-Repo (with changed Makefile for Cross-Compile)

a forum user give me the tip here that partition config can be wrote with mmc-utils too

```
./mmc bootpart enable 1 1 /dev/mmcblk1
```

```
[18:02] root@bpi-r2:~# ./mmc extcsd read /dev/mmcblk1 | grep
PARTITION_CONFIG
Boot configuration bytes [PARTITION_CONFIG: 0x00]
[18:02] root@bpi-r2:~# ./mmc bootpart enable 1 1 /dev/mmcblk1
[18:03] root@bpi-r2:~# ./mmc extcsd read /dev/mmcblk1 | grep
PARTITION_CONFIG
Boot configuration bytes [PARTITION_CONFIG: 0x48]
```

install OS on EMMC

http://forum.banana-pi.org/t/bpi-r2-new-image-ubuntu-16-04-v1-2-1-bt-and-wifi-ap-mode-are-working-fine-2017-11-27/4291

1. change_partition-configuration_of_emmc
2. activate write mode /dev/mmcblk1boot0:

```
echo 0 > /sys/block/mmcblk1boot0/force_ro
```

3. write preloader from here to the emmc boot-device:
   - ```
     gunzip -c BPI-R2-EMMC-boot0-DDR1600-0k-0905.img.gz | sudo dd
     of=/dev/mmcblk1boot0 bs=1024 seek=0
     ```
   - with bpi-tools:

     ```
     bpi-bootsel BPI-R2-EMMC-boot0-DDR1600-0k-0905.img.gz
     /dev/mmcblk1boot0
     ```
4. copy of OS-image to EMMC (device=/dev/mmcblk1):
   - ```
     gunzip deb_stretch_emmc.img.gz
     dd if=deb_stretch_emmc.img of=/dev/mmcblk1 bs=1M
     ```

   - ```
     unzip -p <XXX.img.zip> | pv | dd of=<device> bs=10M status=noxfer
     ```
   - alternate (with bpi-tools):

     ```
     bpi-copy  <XXX.img.zip> <device>
     ```
5. poweroff, remove SD card and boot again

if sd-card-image does not fit emmc: shrinking image

manual copy of OS

- if you want bootable sdcard,you need to copy first 2k from existing sd-image/card.
  - first 2kB without preloader/uboot
    first MB with preloader/uboot

  ```
  gunzip bpi-r2-sd-boot*.img.gz
  dd if=bpi-r2-sd-boot1m.img of=/dev/sdx
  sync
  #remove card after "sync" and reenter
  sudo mkfs -t vfat /dev/sdx1
  sudo mkfs -t ext4 /dev/sdx2
  #set filesystem-labels, partition must not be mounted!
  sudo mlabel -i /dev/sdx1 ::BPI-BOOT
  sudo tune2fs -L BPI-ROOT /dev/sdx2
  ```
- install uboot (not needed if bpi-r2-sd-boot1m.img is flashed):

  ```
  dd if=BPI-R2-720P-2k.img of=/dev/mmcblk1 bs=1k seek=2 count=1022
  ```
- export partitiontable from SD
  parttable.dat
  and write it to emmc (not needed if bpi-r2-sd-bootx.img is flashed):
  ```

```
root@bpi-r2:~# sfdisk -d /dev/mmcblk0 > parttable.dat
root@bpi-r2:~# sfdisk /dev/mmcblk1 < parttable.dat
```

- check/resize
- create filesystems (mkfs) for p1=vfat (apt-get install dosfstools) and p2=ext4 (here emmc)

```
mkfs -t vfat /dev/mmcblk1p1
mkfs -t ext4 /dev/mmcblk1p2
```

- create+configure mount-points in existing system

```
mkdir -p /mnt/emmc/boot
mkdir -p /mnt/emmc/root
nano /etc/fstab
# <file system>         <dir>           <type>  <options>
<dump>  <pass>
/dev/mmcblk0p2          /               ext4    errors=remount-ro
0       1
/dev/mmcblk0p1          /boot           vfat    defaults
0       0
/dev/mmcblk1p2          /mnt/emmc/root  ext4    errors=remount-
ro,noauto  0       1
/dev/mmcblk1p1          /mnt/emmc/boot  vfat    defaults,noauto
0       0
```

- mount it:

```
mount /mnt/emmc/root
mount /mnt/emmc/boot
```

- unpack bootstrapped debian or copy full rootfs from sdcard

```
rsync -aAXv --
exclude={"/dev/*","/proc/*","/sys/*","/tmp/*","/run/*","/mnt/*","/media
/*","/lost+found","/boot/*"} / /mnt/emmc/root/
```

- copy kernel (p1) and modules (p2)

```
mkdir -p /mnt/emmc/boot/bananapi/bpi-r2/linux
cp /boot/bananapi/bpi-r2/linux/uImage /mnt/emmc/boot/bananapi/bpi-
r2/linux
mkdir -p /mnt/emmc/root/lib/modules/
cp -r /lib/modules/$(uname -r) /mnt/emmc/root/lib/modules/
```

- configure uboot to load kernel from right partition

```
sed 's/mmcblk0/mmcblk1/' /boot/bananapi/bpi-r2/linux/uEnv.txt >
/mnt/emmc/boot/bananapi/bpi-r2/linux/uEnv.txt
```