Uboot

default-uboot

```
*** U-Boot SD CARD ***

   1. System Load Linux to SDRAM via TFTP.
   2. System Load Linux Kernel then write to Flash via TFTP.
   3. Boot Linux from SD.
   4. System Load U-Boot then write to Flash via TFTP.
   5. System Load U-Boot then write to Flash via Serial.
   6. System Load ATF then write to Flash via TFTP.
   7. System Load Preloader then write to Flash via TFTP.
   8. System Load ROM header then write to Flash via TFTP.
   9. System Load CTP then write to Flash via TFTP.
   a. System Load CTP then Boot to CTP (via Flash).
   b. System Load flashimage then write to Flash via TFTP.
   c. System Load partition table then write to Flash via TFTP.
   U-Boot console


 Press UP/DOWN to move or Press 1~9,a~b to choose, ENTER to select
```

bootmenu is defined here:
https://github.com/BPI-SINOVOIP/BPI-R64-bsp/blob/master/u-boot-mt/include/configs/mt7622_evb.h

official uboot is not booting, if compiled with gcc >=5.5 (needs to be comiled with e.g. gcc4.8 in ubuntu 14.4), gcc7/8 do not compile official uboot - as upstream supports mt7622/bpi-r64 including net and my repo also supports 32bit mode you can use a recent uboot now

```
UBOOT=u-boot-mt/u-boot-mtk.bin
sudo dd if=$UBOOT   of=$O bs=1k seek=768 #768k = 0xC0000
```

my uboot

- https://github.com/frank-w/u-boot/tree/bpi-r64 (modified version of sinovoip) supporting gcc >= 5.5 (up to 8.3)
- https://drive.google.com/open?id=1Vg3eoHpx3nlZ9pTCZdDpC2l1pJ4rQh5i binaries

default environment

```
BPI-IoT> printenv
arch=arm
atf_filename=trustzone.bin
```

```
baudrate=115200
board=bpi-r64
board_name=mt7622_evb
boot0=download_setting kernel;tftpboot ${loadaddr} ${kernel_filename};
bootm
boot1=download_setting kernel;tftpboot ${loadaddr} ${kernel_filename};run
boot_wr_img;run boot_rd_img;bootm
boot10=download_setting flashimage;tftpboot ${loadaddr}
${flashimage_filename};run wr_flashimage;invaild_env
boot11=download_setting gpt;tftpboot ${loadaddr} ${gpt_filename};run wr_gpt
boot12=mmc init; run boot_normal; bootm
boot2=run boot_rd_img;bootm
boot3=download_setting uboot;tftpboot ${loadaddr} ${uboot_filename};run
wr_uboot;invaild_env
boot4=loadb;run wr_uboot;invaild_env
boot5=download_setting atf;tftpboot ${loadaddr} ${atf_filename};run wr_atf
boot6=download_setting preloader;tftpboot ${loadaddr}
${preloader_filename};run wr_pl
boot7=download_setting hdr;tftpboot ${loadaddr} ${hdr_filename};run
wr_rom_hdr
boot8=download_setting ctp;tftpboot ${loadaddr} ${ctp_filename};run wr_ctp
boot9=run boot_rd_ctp;boot_to_ctp
boot_normal=if run checksd; then echo Boot from SD ; setenv partition 1:1;
else echo Boot from eMMC ; mmc init 0 ; setenv partition 0:1 ; fi; if run
loadbootenv; then echo Loaded environment from ${bootenv}; env import -t
${scriptaddr} ${filesize}; fi; run uenvcmd; fatload mmc 0:1 ${loadaddr}
${bpi}/${board}/${service}/${kernel}; bootm
boot_rd_ctp=mmc device 1;mmc read ${loadaddr} 0x1000 0xa000
boot_rd_img=mmc device 1;mmc read ${loadaddr} 0x1000 1;image_blks 512;mmc
read ${loadaddr} 0x1000 ${img_blks}
boot_wr_img=image_blks 512  ${filesize};mmc device 1;mmc write ${loadaddr}
0x1000 ${img_blks}
bootcmd=No
bootdelay=3
bootenv=uEnv.txt
bootfile=iverson_uImage
bootmenu_0=1. System Load Linux to SDRAM via TFTP.=run boot0
bootmenu_1=2. System Load Linux Kernel then write to Flash via TFTP.=run
boot1
bootmenu_10=b. System Load flashimage then write to Flash via TFTP.=run
boot10
bootmenu_11=c. System Load partition table then write to Flash via
TFTP.=run boot11
bootmenu_2=3. Boot Linux from SD.=run boot12
bootmenu_3=4. System Load U-Boot then write to Flash via TFTP.=run boot3
bootmenu_4=5. System Load U-Boot then write to Flash via Serial.=run boot4
bootmenu_5=6. System Load ATF then write to Flash via TFTP.=run boot5
bootmenu_6=7. System Load Preloader then write to Flash via TFTP.=run boot6
bootmenu_7=8. System Load ROM header then write to Flash via TFTP.=run
boot7
bootmenu_8=9. System Load CTP then write to Flash via TFTP.=run boot8
```

```
bootmenu_9=a. System Load CTP then Boot to CTP (via Flash).=run boot9
bootmenu_delay=30
bpi=bananapi
bpiver=1
checksd=fatinfo ${device} 1:1
chip=MT7622
cpu=armv7
ctp_filename=ctp.bin
debug=7
device=mmc
ethact=mtk_eth
ethaddr=00:0C:E7:11:22:33
fdt_high=0x6c000000
flashimage_filename=flashimage.bin
gpt_filename=GPT_EMMC
hdr_filename=hdr.binary
invaild_env=no
ipaddr=10.10.10.254
kernel=uImage
kernel_filename=iverson_uImage
loadaddr=0x4007FF28
loadbootenv=fatload ${device} ${partition} ${scriptaddr}
${bpi}/${board}/${service}/${bootenv} || fatload ${device} ${partition}
${scriptaddr} ${bootenv}
partition=1:1
preloader_filename=preloader_fpga7622_64_ldvt.bin
root=/dev/mmcblk0p2
scriptaddr=0x43000000
serverip=10.10.10.3
service=linux
soc=mt7622
stderr=serial
stdin=serial
stdout=serial
uboot_filename=u-boot-mtk.bin
vendor=mediatek
wr_atf=mmc device 1;mmc write ${loadaddr} 0x400 0x200
wr_ctp=mmc device 1;mmc write ${loadaddr} 0x1000 0xa000
wr_flashimage=mmc device 1;mmc write ${loadaddr} 0x0 0xa000
wr_gpt=mmc device 1;mmc write ${loadaddr} 0x0 0x22
wr_pl=mmc device 1;mmc write ${loadaddr} 0x100 0x200
wr_rom_hdr=mmc device 1;mmc write ${loadaddr} 0x0 0x50
wr_uboot=mmc device 1;mmc write ${loadaddr} 0x600 0x400
Environment size: 3838/4092 bytes
```

modified uEnv.txt

modified uEnv.txt of debian-Images (for testing)

```
bpi=bananapi
board=bpi-r64
chip=mt7622
service=linux

root=/dev/mmcblk0p2 rootfstype=ext4 rootwait
console=ttyS0,115200n1 earlyprintk
bootopts=debug=7 initcall_debug=0 androidboot.hardware=mt7622 swiotlb=512
abootargs=setenv bootargs board=${board} console=${console} root=${root}
service=${service} ${bootopts}

kernel=uImage
fdt=mt7622-bananapi-r64.dtb

kaddr=0x44000000
rdaddr=0x46000000
dtaddr=0x47000000

aload_fdt=fatload $device $partition $dtaddr
${bpi}/${board}/${service}/dtb/${fdt}
aload_kernel=fatload $device $partition $kaddr
${bpi}/${board}/${service}/${kernel}

aboot=bootm $kaddr - $dtaddr

uenvcmd=run abootargs aload_fdt aload_kernel aboot
```

eMMC

from linux flash emmc-image to emmc user partition

```
 gunzip -c /mnt/bpi-r64_emmc.img.gz | dd bs=1M status=progress
conv=notrunc,fsync of=/dev/mmcblk0
```

from sdmmc u-boot flash bl2 and set partconf right

```
BPI-R64> mmc partconf 0 1 1 1
BPI-R64> fatload usb 0:1 $loadaddr r64/bpi-r64_emmc_bl2.img
ERROR: reserving fdt memory region failed (addr=0 size=43000000 flags=4)
65597 bytes read in 16 ms (3.9 MiB/s)
BPI-R64> printenv loadaddr
loadaddr=0x44000000
BPI-R64> mmc write ${loadaddr} 0x0 0x400

MMC write: dev # 0, block # 0, count 1024 ... 1024 blocks written: OK
BPI-R64> mmc partconf 0 1 1 0
```

https://forum.banana-pi.org/t/debian-ubuntu-image-creation/15826/75

tftp

```
BPI-R64> printenv tfu
tfu=setexpr umtkaddr ${uaddr} - 0x200;tftp ${umtkaddr} ${ufile};go ${uaddr}
BPI-R64> setenv ufile u-boot-mtk_r64_sd_rtl8367_gcc6.5.bin
BPI-R64> run tfu
Using mtk_eth device
TFTP from server 192.168.0.10; our IP address is 192.168.0.18
Filename 'u-boot-mtk_r64_sd_rtl8367_gcc6.5.bin'.
Load address: 0x41dffe00
Loading: ####################
         2.4 MiB/s
done
Bytes transferred = 297468 (489fc hex)
get filesize 0x489fc
## Starting application at 0x41E00000 ...
U-Boot 2014.04-rc1-00024-g35908bc2eb-dirty (Sep 26 2019 - 08:32:43)
DRAM:  1008 MiB
WARNING: Caches not enabled
...
BPI-R64> version

U-Boot 2014.04-rc1-00024-g35908bc2eb-dirty (Sep 26 2019 - 08:32:43)
arm-linux-gnueabihf-gcc (Ubuntu/Linaro 6.5.0-2ubuntu1~18.04) 6.5.0 20181026
GNU ld (GNU Binutils for Ubuntu) 2.30
BPI-R64>
```

Patches

alphanumeric

https://github.com/BPI-SINOVOIP/BPI-R64-bsp/blob/master/u-boot-mt/common/cmd_bootmenu.c#L119

boot_new

(r2) 🔧 Fix Me!

```
alt:
boot_normal=if run checksd; then echo Boot from SD ; setenv partition 1:1;
else echo Boot from eMMC ; mmc init 0 ; setenv partition 0:1 ; fi; if run
loadbootenv; then echo Loaded environment from ${bootenv}; env import -t
${scriptaddr} ${filesize}; fi; run uenvcmd; fatload mmc 0:1 ${loadaddr}
${bpi}/${board}/${service}/${kernel}; bootm
```

```
neu:
console=earlyprintk console=tty1 fbcon=map:0 console=ttyS0,115200
root=/dev/mmcblk0p2 rootfstype=ext4 rootwait
bootopts=vmalloc=496M debug=7 initcall_debug=0

buildargs=setenv bootargs "board=${board} console=${console} root=${root}
${bootopts}"

checkenv=test -e ${device} ${partition}
${bpi}/${board}/${service}/${bootenv}
importenv=env import -t ${scriptaddr} ${filesize}
loadbootenv=if fatload ${device} ${partition} ${scriptaddr}
${bpi}/${board}/${service}/${bootenv};then run importenv;else echo "fatload
(${bootenv}) failed";fi
newboot=run buildargs;printenv bootargs;fatload ${device} ${partition}
${loadaddr} ${bpi}/${board}/${service}/${kernel}; bootm

checksd=fatinfo ${device} 1:1
selectmmc=if run checksd; then echo Boot from SD ; setenv partition
1:1;else echo Boot from eMMC; setenv partition 0:1 ; fi;

boot1=run selectmmc; run loadbootenv; run importenv; run newboot;
```

manual boot

```
U-Boot MT7622> ls mmc 1:1
            bananapi/
         0   sd.txt
  8841288   uImage_5.4
    23721   bpi-r64-5.4.dtb

3 file(s), 1 dir(s)

U-Boot MT7622> setenv kaddr 0x44000000
U-Boot MT7622> setenv dtaddr 0x47000000
U-Boot MT7622> fatload mmc 1:1 $kaddr uImage_5.4
8841288 bytes read in 786 ms (10.7 MiB/s)
U-Boot MT7622> fatload mmc 1:1 $dtaddr bpi-r64-5.4.dtb
23721 bytes read in 8 ms (2.8 MiB/s)
U-Boot MT7622> setenv bootargs console=ttyS0,115200 root=/dev/mmcblk1p2 rw
rootwait
U-Boot MT7622> bootm $kaddr - $dtaddr
```

netboot

```
BPI-R64> setenv ipaddr 192.168.0.18
BPI-R64> setenv netmask 255.255.255.0
```

```
BPI-R64> setenv serverip 192.168.0.10
BPI-R64> setenv bootfile uImage_r64
BPI-R64> setenv bootdtbfile r64.dtb

BPI-R64> setenv bootargs console=ttyS0,115200 root=/dev/mmcblk0p2 rw
rootwait ip=dhcp

BPI-R64> setenv kaddr 0x44000000
BPI-R64> setenv dtaddr 0x47000000

BPI-R64> tftp $kaddr ${bootfile}
BPI-R64> tftp $dtaddr ${bootdtbfile}

BPI-R64> bootm $kaddr - $dtaddr
```

PCIe

```
BPI-R64> pci enum
PCI: Failed autoconfig bar 10
PCI: Failed autoconfig bar 10
BPI-R64> pci 0
Scanning PCI devices on bus 0
BusDevFun  VendorId   DeviceId   Device Class       Sub-Class
---
00.00.00   0x14c3     0x3258     Bridge device           0x04
00.01.00   0x14c3     0x3258     Bridge device           0x04
BPI-R64> pci 1
Scanning PCI devices on bus 1
BusDevFun  VendorId   DeviceId   Device Class       Sub-Class
---
01.00.00   0x168c     0x003c     Network controller      0x80
BPI-R64> pci 2
Scanning PCI devices on bus 2
BusDevFun  VendorId   DeviceId   Device Class       Sub-Class
---
02.00.00   0x14c3     0x7612     Network controller      0x80
BPI-R64>
```

SATA

```
BPI-R64> scsi scan
scanning bus for devices...
Target spinup took 0 ms.
AHCI 0001.0300 32 slots 1 ports 6 Gbps 0x1 impl SATA mode
flags: ncq stag pm clo only pmp pio slum part ccc apst
  Device 0: (0:0) Vendor: ATA Prod.: ST750LM022 HN-M7 Rev: 2AR1
            Type: Hard Disk
            Capacity: 715404.8 MB = 698.6 GB (1465149168 x 512)
```

```
BPI-R64> ls scsi 0:1
            EFI/
      512   BOOTSECT.BAK

1 file(s), 1 dir(s)

BPI-R64>
```

USB

```
BPI-R64> usb start
starting USB...
Bus usb@1a0c0000: hcd: 0x1a0c0000, ippc: 0x1a0c4700
u2p:2, u3p:1
Register 300010f NbrPorts 3
Starting the controller
USB XHCI 0.96
scanning bus usb@1a0c0000 for devices... 2 USB Device(s) found
       scanning usb for storage devices... 1 Storage Device(s) found
BPI-R64> ls usb 0:1
       91   mcurom.md5
            System Volume Information/

1 file(s), 1 dir(s)

BPI-R64>
```