

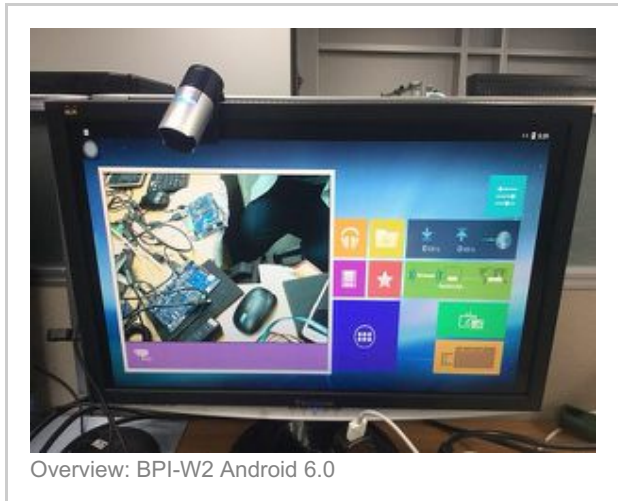
Getting Started with BPI-W2

wiki.banana-pi.org/Getting_Started_with_BPI-W2

- [1 Introduction](#)
- [2 Development](#)

Introduction

Read more: [Banana Pi BPI-W2](#)



Development

Let's get start to develop on BPI-W2, see amazing things happen.

Basic Development

Prepare to develop

Load your first image on W2

Build your own image on W2

How to make a SD card with bootloader

there are two ways to boot w2, controlled by SW4



- 0 boot from emmc.
- 1 boot from SPI + SD card.

After power on, there are 3 steps must be done before loading kernel,(Normally they all are called bootloader):

- step 1: CPU init - the code is inside CPU;
- step 2: init the flash device on which we boot from; (normally it's called preloader)
- step 3: u-boot;

RTD1296 supports booting from eMMC and SPI flash, so if we want to boot W2 from SD card, (that means we put u-boot on the SD card.) the second step code must be written to SPI flash, normally it's done in factory before the board sending out, and the code is fixed and provided by RTK without source. We need not care about it, it does a simple job.

In fact, if the board will boot from eMMC, the second code must be written to eMMC first too, remember that we introduced how to do it with u-boot together on other pages. Normally it's also been done in factory.

The u-boot for eMMC and SD card are different, normally eMMC u-boot 's name is **dvrboot.exe.bin**, and u-boot for SD card name is **u-boot.bin**.

Write u-boot to SD card

The u-boot file for booting from SD card is here:

https://drive.google.com/drive/folders/17ShSHLOvxeYA6tI9HQXOAb3O7RjqfD_6?usp=sharing

step 1: build a SD card system according:

2.1.3.2.1 How to make and run the 64-bit ubuntu 16.04 on SD card 2018-4-28

- step 2: boot up the board with eMMC and press 'Esc' key to let board stays on u-boot prompt;
- step 3: using tftp to get u-boot.bin file from network;

```
tftp 0x1500000 u-boot.bin
```

step4: using sd command to write it to SD card:

```
sd write 0x1500000 0x50 0x3f0
```

forum pthread

Ubuntu

How to make and run the 64-bit ubuntu 16.04 on SD card 2018-4-28

1.Requirement :

- 1.A PC with ubuntu as the host;
- 2.A banana Pi W2 board;
- 3.A SD card;
- 4.Downloaded latest ubuntu base tar file (ubuntu-base-16.04.4-base-arm64.tar.gz) from url:

<http://cdimage.ubuntu.com/ubuntu-base/releases/16.04/release/> 17

5.Hardware configuration file, u-boot binary file, kernel image file and audio file, which can be downloaded from:

https://drive.google.com/file/d/1LX4FwXcSzkul4BUPEVxbUyeG_UYwlkwY/view?usp=sharing 17

oRTD1296_hwsetting_BOOT_4DDR4_4Gb_s1866_padding.bin – hardware config

odvrboot.exe.bin – u-boot binary

oulmage – kernel image

obluecore.audio – audio and video firmware for RTK1296

2.Write the bootloader to the Banana Pi W2 flash:

Banana Pi W2 board can boot from EMMC flash or SPI flash on the board, the difference is the location of the bootloader, so if you will not participate in the development of u-boot, this will not matter to you. So here is how to write bootloader to EMMC.

Normally the W2 board you got had been programed in the factory, but if the version of the bootloader isn't the latest one, I recommend you to program it by yourself again. The latest version of bootloader is available on the Banana Pi website.

Requirement:

- 1.A windows PC runs serial terminal tool like hypertrm. (I recommend using “hypertrm”, because other tools sometimes fail to transmit data);
- 2.Latest version u-boot binary file from Banana Pi website (dvrboot.exe.bin);
- 3.A hardware configuration binary file for W2 (RTD1296_hwsetting_BOOT_4DDR4_4Gb_s1866_padding.bin);

Steps:

- 1.Connect the serial port between the host and the board correctly, run the terminal software and set up the relevant parameters (115200, 8N1, none);
- 2.press “ctrl+q” then to power on the board, and when the screen appears ' d/g/r ', it can be released;
- 3.Input “h” and send hardware configuration binary files

(RTD1296_hwsetting_BOOT_4DDR4_4Gb_s1866_padding.bin) in Y-modem mode on the terminal side.

- 4.Wait for the transfer to complete, then input below in console:

```
s98007058
01500000
```

- 5.Input “d” and send u-boot binary files (dvrboot.exe.bin) in Y-modem mode on the terminal side.
- 6.Wait for the transmission to complete. Enter “g” then the u-boot will be programmed to the EMMC automatically.

3.Make a ubuntu root file system for W2 on a ubuntu PC.

Ubuntu has the version for arm64 architecture, it's suitable for the Banana Pi W2, most of packages built by ubuntu for arm64 can run on the W2 board.

Requirement:

- 1.A PC runs linux (recommend ubuntu)
- 2.The latest version of ubuntu tar file for arm64 (ubuntu-base-16.04.4-base-arm64.tar.gz) is downloaded from ubuntu website:
<http://cdimage.ubuntu.com/ubuntu-base/releases/16.04/release/> 17

Steps: (on the host) :

- 1.mkdir rootfs
- 2.su root
- 3.tar zxvf ubuntu-base-16.04.4-base-arm64.tar.gz -C rootfs
- 4.write a shell script file, please copy below contents to the file ch-mount.sh (vi ch-mount.sh and copy):

```
#!/bin/bash
```

```
function mnt() {
    echo "MOUNTING"
    sudo mount -t proc /proc ${2}proc
    sudo mount -t sysfs /sys ${2}sys
    sudo mount -o bind /dev ${2}dev
    sudo mount -o bind /dev/pts ${2}dev/pts
    sudo chroot ${2}
}
```

```
function umnt() {
    echo "UNMOUNTING"
    sudo umount ${2}proc
    sudo umount ${2}sys
    sudo umount ${2}dev/pts
    sudo umount ${2}dev
}
```

```

if [ "$1" == "-m" ] && [ -n "$2" ] ;
then
    mnt $1 $2
elif [ "$1" == "-u" ] && [ -n "$2" ];
then
    umnt $1 $2
else
    echo ""
    echo "Either 1'st, 2'nd or both parameters were missing"
    echo ""
    echo "1'st parameter can be one of these: -m(mount) OR -u(umount)"
    echo "2'nd parameter is the full path of rootfs directory(with trailing '/')"
    echo ""
    echo "For example: ch-mount -m /media/sdcard/"
    echo ""
    echo 1st parameter : ${1}
    echo 2nd parameter : ${2}
fi

```

- 5.apt-get install qemu-user-static
- 6.cp /usr/bin/qemu-aarch64-static rootfs/usr/bin
- 7.cp -b /etc/resolv.conf rootfs/etc/
- 8../ch-mount.sh -m rootfs/
- 9.apt-get install iputils-ping
- 10.apt-get install udev
- 11.apt-get install net-tools
- 12.Modify the root password by running: "passwd root"
- 13.exit
- 14../ch-mount.sh -u rootfs/

Until now, the ubuntu file system have been built successfully on the host, next we will write it to the SD card.

4.Write the system to the SD card (on the host) :

Requirement:

- 1.A SD card;
- 2.Device tree binary file downloaded from Banana Pi website: bpi-w2.dtb;
- 3.Kernel image file downloaded from Banana Pi website: ulmage;
- 4.Video and Audio firmware file downloaded from Banana Pi website: bluecore.audio;

Steps:

- 1.Using fdisk tool to create two partitions in SD card: format the first partition as FAT32, and the second partition as ext4; It's recommended that the first partition size should not be less than 100M;
- 2.Copy the three files: bpi-w2.dtb, ulmage, bluecore.audio to the first partition;
- 3.Copy the all files in root filesystem which just built by us to the second partition using root privilege:

```

cp -a rootfs/* /media/xxx/sdxx/
sync

```

- 4.Eject the SD card and insert it to the W2 board, after power on the board, you can see the ubuntu is running now.
- 5.Set up network and get packages by using apt-get from ubuntu repository

Steps:

- 1.Log in with root and the password was set by yourself;
- 2.Ifconfig eth0 or eth1: ifconfig eth1 192.168.2.231
- 3.Add default gateway to let the board can reach internet: route add default gw 192.168.2.1
- 4.Add a DNS: echo 'nameserver 8.8.8.8' >> /etc/resolv.conf
- 5.apt-get update
- 6.then you can fetch and install any packages from ubuntu by using apt-get.

6.Furthermore :

Some functions are not available on the W2 now, because the kernel is not fully functional, we are working hard to make it more and more complete, please follow our official website update.

PDF format file download

<http://forum.banana-pi.org/uploads/default/original/2X/c/c6dc409abc249c57a4cb2b7aedaddec6b80a9026.pdf>

Ubuntu 16.04 Install and run Node.js and http-server

install Node.js

After running Ubuntu 16.04, it's easy to install packages on the W2. Please follow below steps to install node.js:

1. apt-get install nodejs
2. apt-get install npm
3. npm install -g http-server
4. make a link to nodejs: cd /usr/bin/; ln -sf nodejs node

then Node.js is ready on your system.

OpenWrt

Burn OpenWRT with Linux kernel 4.4

1.About this release

This release is for banana pi W2 board to run OpenWRT , you will know how to burn and run OpenWRT on W2.

2.Burn OpenWRT on W2 steps :

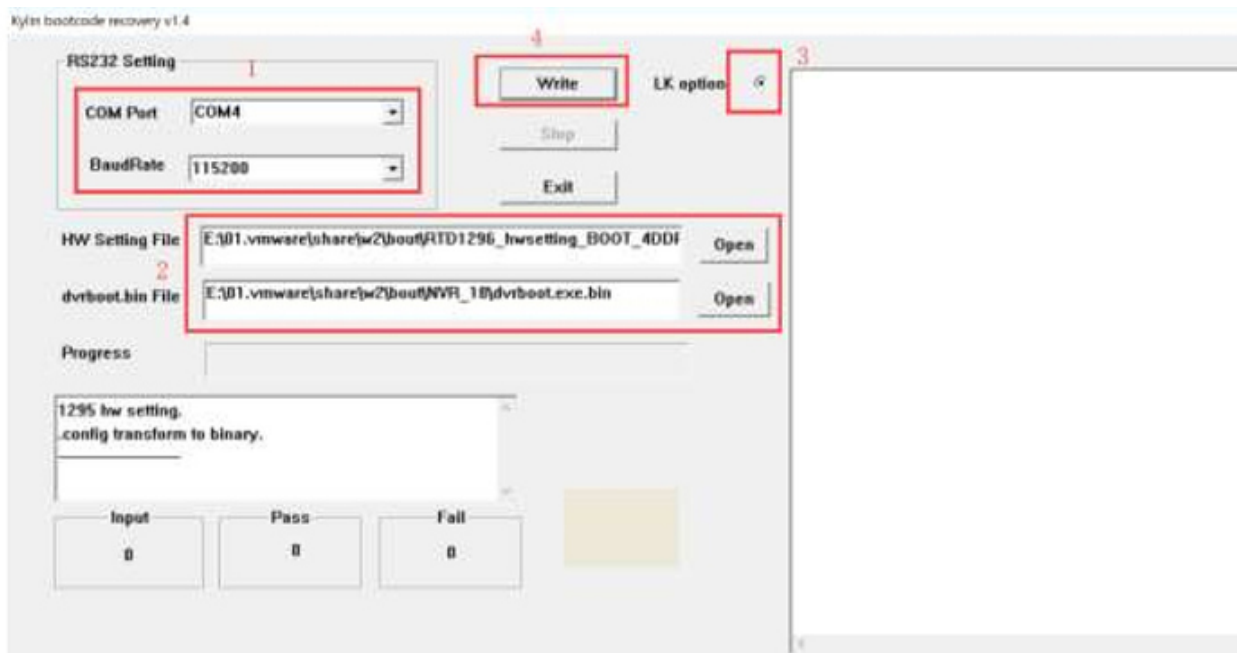
The Bootloader of OpenWRT is u-boot64, there are two ways to burn it on W2 EMMC.

A. Use RTK MP_Tool burn uboot on W2 :

1) You need these files and tool :

- MP_Tool
- RTD1296_hwsetting_BOOT_4DDR4_4Gb_s1866.config (hardware configuration file)
- dvrboot.exe.bin (u-boot binary file)

2) Connect W2 debug console to windows PC , and run MP_Tool :



- Set your console parameters
- Choose config file and uboot file
- Choose "LK option"
- Click Write button and then power on W2, and waiting for response "OK"

B. Use hyperterm burn uboot on W2 :

1) You need these files and tool :

- hyperterm
- RTD1296_hwsetting_BOOT_4DDR4_4Gb_s1866_padding.bin (hardware configuration binary file)
- dvrboot.exe.bin (u-boot binary file)

2) Connect W2 debug console to windows PC :

- Run hyperterm , config console parameters right
- Holding on press "ctrl + q" and power on board , after the screen show "d/g/r" , then loose "ctrl + q"
- Input "h" , use Y-modem way to send "RTD1296_hwsetting_BOOT_4DDR4_4Gb_s1866_padding.bin" to W2 board
- After finish transferred , input these parameters :

```
::s98007058
::01500000
```

- Input "d" , and send u-boot binary file
- after finish transferred , input "g" to burn u-boot file on W2 EMMC

After succeed to burn u-boot on W2, then install OpenWRT on W2.

1) You need these tools and files :

- U disk
- install.img

2) Burn OpenWRT on W2 steps :

- Format U disk as FAT32 filesystem
- Copy install.img to U disk root directory
- Decompress install.img to get four files :

emmc.uImage

rescue.root.emmc.cpio.gz_pad.img

bluecore.audio

rescue.emmc.dtb

- Plug U disk to W2 USB3.0 interface , holding press “Esc” button to power on W2 board
- Then W2 board will go into u-boot console
- Input these commands step by step :

```
setenv bootargs "earlycon=uart8250,mmio32,0x98007800 console=ttyS0,115200  
loglevel=7
```

```
init=/etc/init root=/dev/mmcblk0p1 rootfstype=squashfs rootwait"
```

```
save
```

```
go ru
```

Waiting for OpenWRT burning , after finished , W2 will reboot and go into OpenWRT system.

```
BusyBox v1.23.2 (2018-04-11 10:15:43 PDT) built-in shell (ash)

  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
 |__| W I R E L E S S   F R E E D O M

-----
CHAOS CALMER (Chaos Calmer, r48422)
-----
* 1 1/2 oz Gin             Shake with a glassful
* 1/4 oz Triple Sec       of broken ice and pour
* 3/4 oz Lime Juice      unstrained into a goblet.
* 1 1/2 oz Orange Juice
* 1 tsp. Grenadine Syrup
-----
root@OpenWrt:/#
```

3. Image & Tools link :

Baidu Drive : <https://pan.baidu.com/s/1Rs9F46wNJibORaeCfVh4wg> PinCode : 1w4w

Google

Drive : <https://drive.google.com/file/d/10kfP142Jw0sA92uwalYZt7CeXcDKelO1/view?usp=sharing>

4.BPI-W2 run OpenWRT video :

<https://youtu.be/eTaRy8Jnfpk>

Advanced Development

GPIO

GPIO Control

PWM Control

Develop 4G module with W2

WiFi and Ap mode on W2

WiFi Client

AP mode

Make your own image

FAQ
