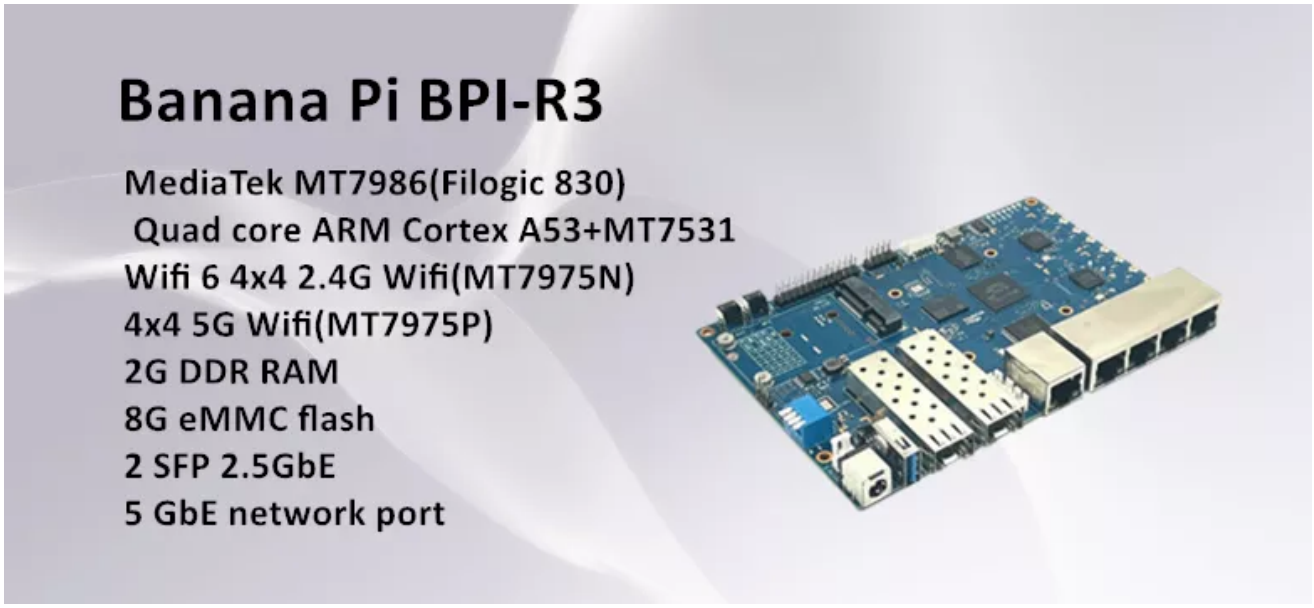
Getting Started with BPI-R3

Contents

Introduction



Banana Pi BPI-R3 Router board with MediaTek MT7986(Filogic 830) quad core ARM A53 + MT7531A chip design ,2G DDR RAM ,8G eMMC flash onboard,It is a very high performance open source router development board,support Wi-Fi 6/6E 2.4G wifi use MT7975N and 5G wifi use MT7975P, support 2 SFP 2.5GbE port, and 5 GbE network port.

MediaTek MT7986(Filogic 830)

The MT7986(Filogic 830) integrates four Arm Cortex-A53 cores up to 2GHz with up to 18,000 DMIPs of processing power and 6Gbps of dual 4x4 Wi-Fi 6/6E connectivity. It has two 2.5g Ethernet interfaces and serial peripheral interfaces (SPI). Filogic 830‘s built-in hardware acceleration engine enables fast and reliable Wi-Fi offloading and wireless network connection. In addition, the chip supports Mediatek FastPath™ technology, which is suitable for games, AR/VR and other low-latency applications.

Wi-fi 6E has many advantages over its predecessors, including lower latency, larger bandwidth capacity and faster transmission rates. Wireless network devices supporting the 6GHz band mainly use 160MHz wide channel and 6GHz uncongested bandwidth to provide multigigabit transmission and low-latency wi-fi connection, providing reliable wireless network for streaming media, games, AR/VR and other applications.

Key Features

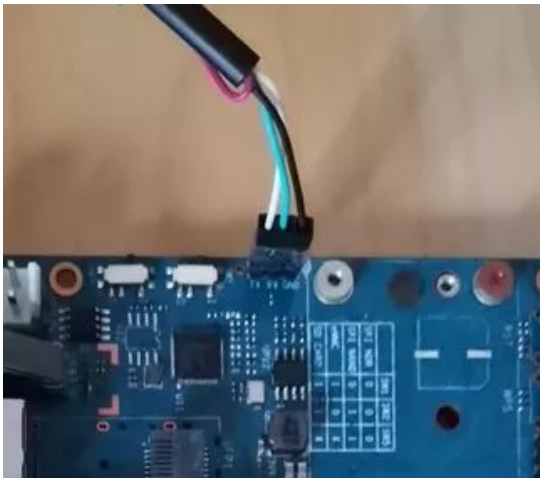
- . MediaTek MT7986(Filogic 830) Quad core ARM Cortex A53+MT7531 chip design
- . Wifi 6/6E 4x4 2.4G Wifi(MT7975N) +4x4 5G Wifi(MT7975P)
- . 2G DDR RAM
- . 8G eMMC flash
- . 2 SFP 2.5GbE
- . 5 GbE network port
- . POE support (optional)
- . Mini PCIe via USB
- . M.2 KEY-E PCIe inerface
- . Mciro SD card interface
- . 26 PIN GPIO

Development

Basic Development

Prepare to develop

```
* Prepare 8G/above TF card, USB-Serial interface, Ubuntu System
* Using your USB-Serial Connect debug console on BPI-R3
```

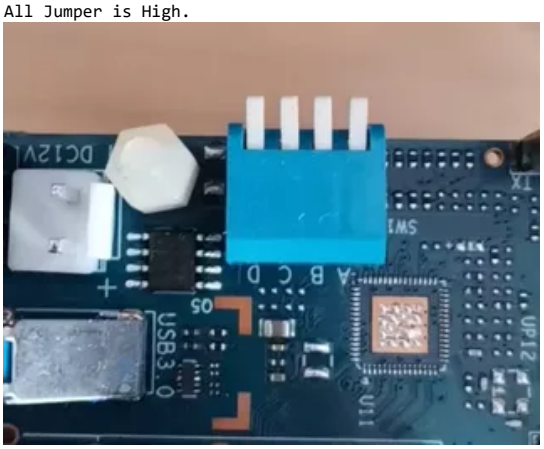


* R3 bootstrap and device select Jumper Setting
  x: Don't care

Jumper setting	SW1-A	SW1-B	SW1-C	SW1-D
SPIM-NOR	Low	Low	Low	x
SPIM-NAND	High	Low	High	x
eMMC	Low	High	x	Low
SD	High	High	x	High

Note: SW1-A and SW1-B is for boot strap selecting;
SW1-C is that SPI-Nand or SPI-Nor Device is connected to CPU's SPI bus;
SW1-D is that SD Card or EMMC device is connected tp CPU's EMMC bus.

* Examples:
  All Jumper is High.



All Jumper is Low.

How to burn image to SD card

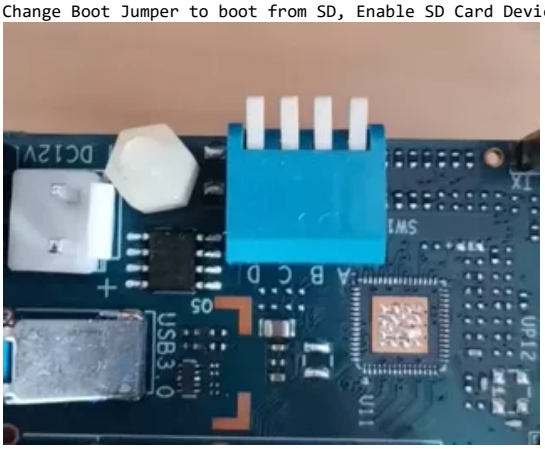
A. Note: burn image to SD card on linux computer
  1.You could download latest image from our forum
  * Here is the example image link:

  2.Install bpi-tools on your Ubuntu. If you can't access this URL or any other problems, please go to bpi-tools repo (https://github.com/BPI-SINOVOI
  * apt-get install pv
  * curl -sL https://github.com/BPI-SINOVOIP/bpi-tools/raw/master/bpi-tools | sudo -E bash

  3.After you download the image, insert your TF card into your Ubuntu
  * Execute "bpi-copy xxx.img /dev/sdx" to install image on your TF card

  4.After step 3, then you can insert your TF card into R3, and press power button to setup R3
B. Note: burn image to SD card on windows computer
  1. Download the tools from the website: https://sourceforge.net/projects/win32diskimager/
  2. Install the tools into Windows computer.
  3. flash image into SD card.

C. Change Boot Jumper to boot from SD, Enable SD Card Device.



How to burn image to onboard eMMC

Note: because SD card and EMMC device share one SOC's interface, you need flash one SD image firstly, then R3 boot from SD card, then flash nand image

Before burning image to eMMC, please prepare a SD card with flashed bootable image and a USB disk. Let's take OpenWrt image (mtk-bpi-r3-SD-WAN1-SFP1-

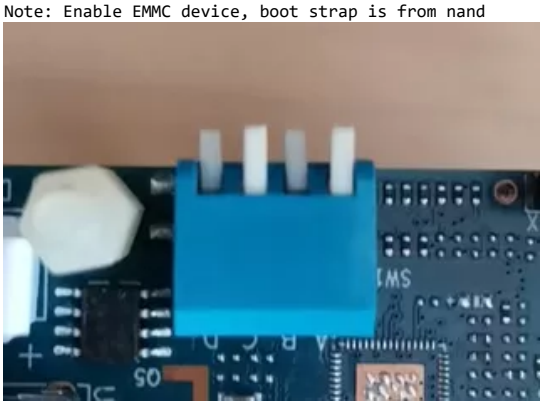
1. Insert the flashed SD card and power on to start the board.(the image "mtk-bpi-r3-SD-WAN1-SFP1-20220619-single-image.img" on the SD card can be Op

2. Copy Nand bootable and EMMC boot OpenWrt image(mtk-bpi-r3-NAND-WAN1-SFP1-20220619-single-image.bin, bl2_emmc.img, mtk-bpi-r3-EMMC-WAN1-SFP1-202206

3. Plug in USB disk to the board, and mount the USB to /mnt or other directory as follows: (you can skip mounting if it is mounted automatically)
   * mount -t vfat /dev/sda1 /mnt
   * change your directory to the mounting point, here is : cd /mnt

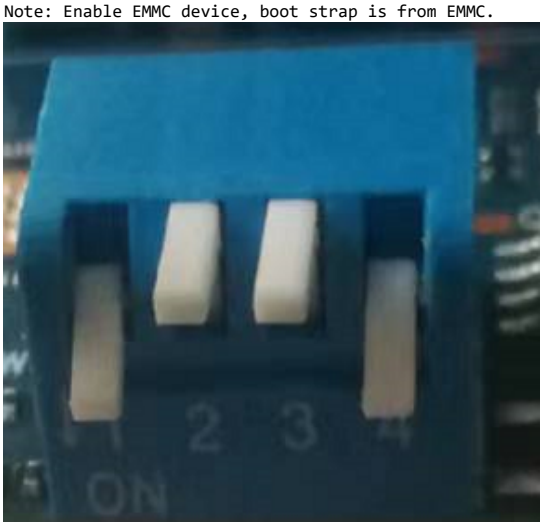
4. Execute following command to enable and copy image to nand flash:
   * mtd erase /dev/mtd0
   * dd if=mtk-bpi-r3-NAND-WAN1-SFP1-20220619-single-image.bin of=/dev/mtdblock0

5. Shutdown, remove SD card, and change bootstrap to boot from nand flash and change SD/EMMC switch jumper to EMMC, restart the board from Nand Flash
   Note: Enable EMMC device, boot strap is from nand



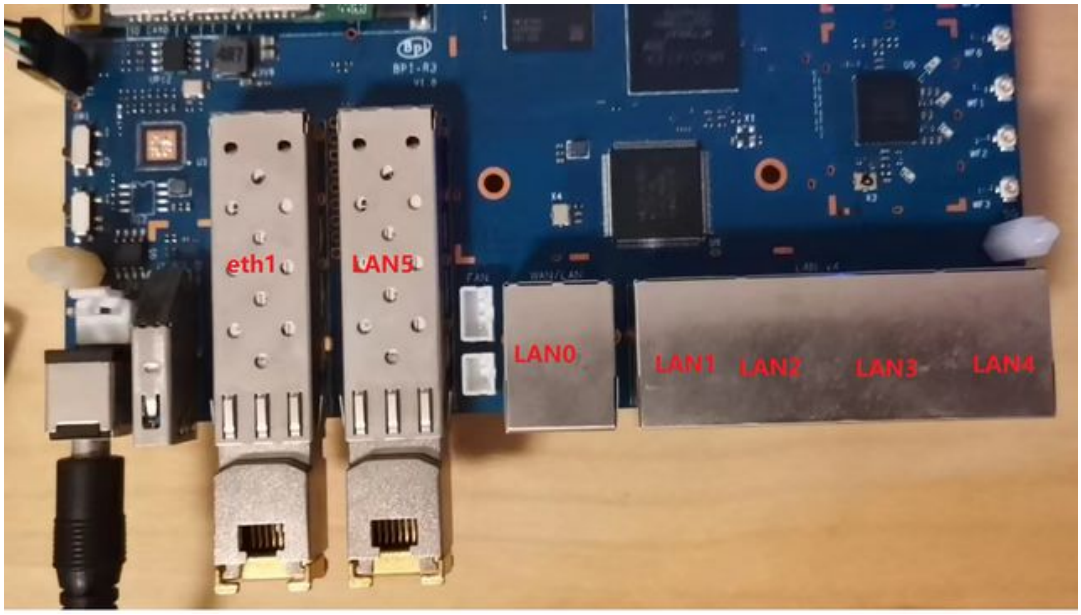
6. repeat step 3, mount u-disk to /mnt, Execute following command to enable and copy image to EMMC device:
   * mount -t vfat /dev/sda1 /mnt
   * echo 0 > /sys/block/mmcblk0boot0/force_ro
   * dd if=bl2_emmc.img of=/dev/mmcblk0boot0
   * dd if=mtk-bpi-r3-EMMC-WAN1-SFP1-20220619-single-image.img of=/dev/mmcblk0
   * mmc bootpart enable 1 1 /dev/mmcblk0

```
7. power off R3 board, remove u-disk driver, change bootstrap to boot from emmc device.
   Note: Enable EMMC device, boot strap is from EMMC.
```



Network-Configuration

- Network-Configuration refer to: http://www.fw-web.de/dokuwiki/doku.php?id=en:bpi-r2:network:start
- Network Interface: eth1, lan0 is for WAN; lan4, rax0, lan2, lan5, ra0, lan3, lan1 is for LAN, ra0 is for 2.4G wireless, rax0 is for 5G wireless.



root@OpenWrt:/# ifconfig

br-lan Link encap:Ethernet HWaddr EE:A1:57:81:CA:19

```
        inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::eca1:57ff:fe81:ca19/64 Scope:Link
        inet6 addr: fd63:8bea:d5ce::1/60 Scope:Global
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:2418 (2.3 KiB)
```

br-wan Link encap:Ethernet HWaddr EE:A1:57:81:CA:19

```
        inet6 addr: fe80::eca1:57ff:fe81:ca19/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
```

```
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:8538 (8.3 KiB)
```

eth0 Link encap:Ethernet HWaddr EE:A1:57:81:CA:19

```
        inet6 addr: fe80::eca1:57ff:fe81:ca19/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:32 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:4408 (4.3 KiB)
        Interrupt:124
```

eth1 Link encap:Ethernet HWaddr 4A:BB:84:B4:5D:3F

```
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:8674 (8.4 KiB)
        Interrupt:124
```

lan0 Link encap:Ethernet HWaddr EE:A1:57:81:CA:19

```
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

lan1 Link encap:Ethernet HWaddr EE:A1:57:81:CA:19

```
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

lan2 Link encap:Ethernet HWaddr EE:A1:57:81:CA:19

```
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

lan3 Link encap:Ethernet HWaddr EE:A1:57:81:CA:19

```
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

lan4 Link encap:Ethernet HWaddr EE:A1:57:81:CA:19

```
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

lan5 Link encap:Ethernet HWaddr EE:A1:57:81:CA:19

```
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
```

```
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:2418 (2.3 KiB)
```

lo Link encap:Local Loopback

```
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:56 errors:0 dropped:0 overruns:0 frame:0
        TX packets:56 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:4368 (4.2 KiB)  TX bytes:4368 (4.2 KiB)
```

ra0 Link encap:Ethernet HWaddr 00:0C:43:26:60:38

```
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
        Interrupt:6
```

rax0 Link encap:Ethernet HWaddr 02:0C:43:36:60:38

```
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

root@OpenWrt:/# brctl show br-wan

bridge name bridge id STP enabled interfaces br-wan 7fff.eea15781ca19 no lan0, eth1

root@OpenWrt:/# brctl show br-lan

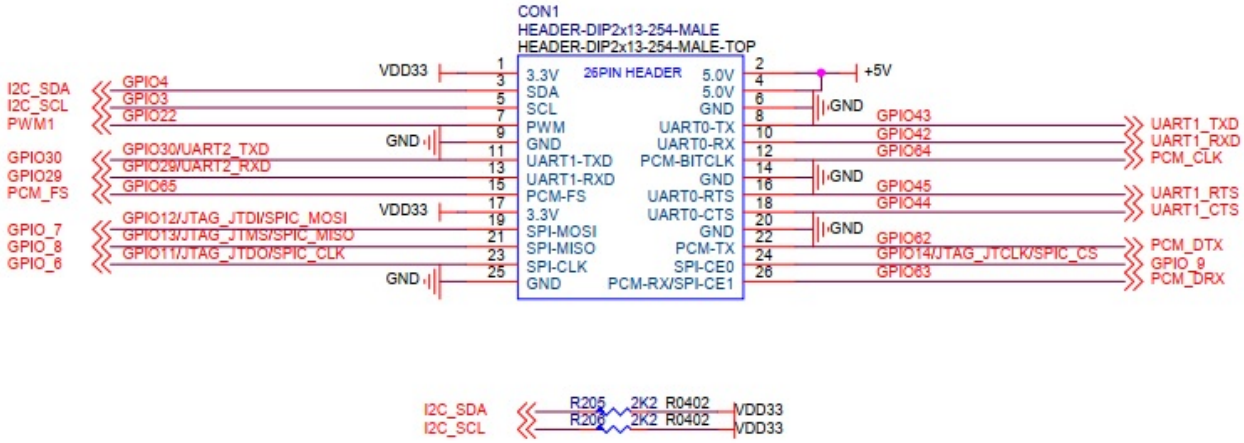
bridge name bridge id STP enabled interfaces br-lan 7fff.eea15781ca19 no lan4, rax0, lan2, lan5, ra0, lan3, lan1

root@OpenWrt:/#

Advanced Development

GPIO

26 Pins Definition



GPIO Control

- echo xxx > /sys/class/gpio/export
- echo in/out > /sys/class/gpio/gpioxxx/direction
- echo 0/1 > /sys/class/gpio/gpioxxx/value

Check the base gpio, you could see mine is 411

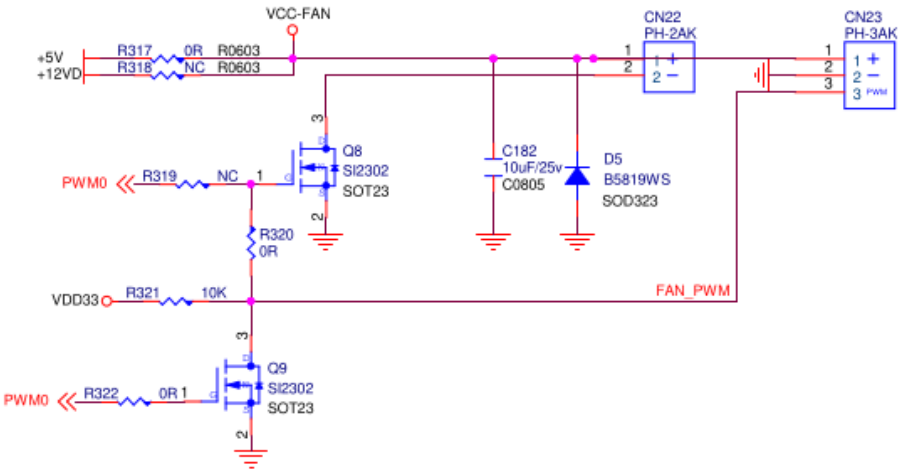
```
root@OpenWrt:/# ls /sys/class/gpio/
export         gpiochip411  unexport
root@OpenWrt:/#
```

For example: if you want to change gpio 22 as out highlevel, you need input commands like this：

- echo 433（22+411） > /sys/class/gpio/export
- echo out > /sys/class/gpio/gpio433/direction

- echo 1 > /sys/class/gpio/gpio433/value

FAN



- R317 for 5V FAN and R318 for 12V FAN.
- CN23 supports PWM control while CN22 does not support.

PWM FAN Control

- echo 0 > /sys/class/pwm/pwmchip0/export
- echo 10000 > /sys/class/pwm/pwmchip0/pwm0/period
- echo 5000 > /sys/class/pwm/pwmchip0/pwm0/duty_cycle
- echo normal > /sys/class/pwm/pwmchip0/pwm0/polarity
- echo 1 > /sys/class/pwm/pwmchip0/pwm0/enable

miniPCIe slot

currently, miniPCIe slot only support one USB 4G module, example: EC25 when you insert one EC25 module, you may check it.

```
root@OpenWrt:/# lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux 5.4.171 xhci-hcd xHCI Host Controller
Bus 001 Device 003: ID 2c7c:0125 Android Android
Bus 001 Device 002: ID 1a40:0101  USB 2.0 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux 5.4.171 xhci-hcd xHCI Host Controller
root@OpenWrt:/#
```

SFP

After high and low temperature test, the following modules are suppoted by BPI-R3:

4G&5G

- BPI-R3 supports 4G LTE EC25.
- If you want to use 5G on BPI-R3:

```
1. Insert 5G dongle into USB3.0.
2. Connect RG200U-CN to mini PCIe, connect SoC through USB2.0(speed limited).
3. Make an RG200U-CN LGA adapter board and insert it into M.2 KEY M.
```

Note: The availability of 4G/5G depends on the local carrier frequency band.

Ap mode on BPI-R3

- ra0 is MT7986a 2.4G wifi
- rax0 is MT7986a 5G wifi

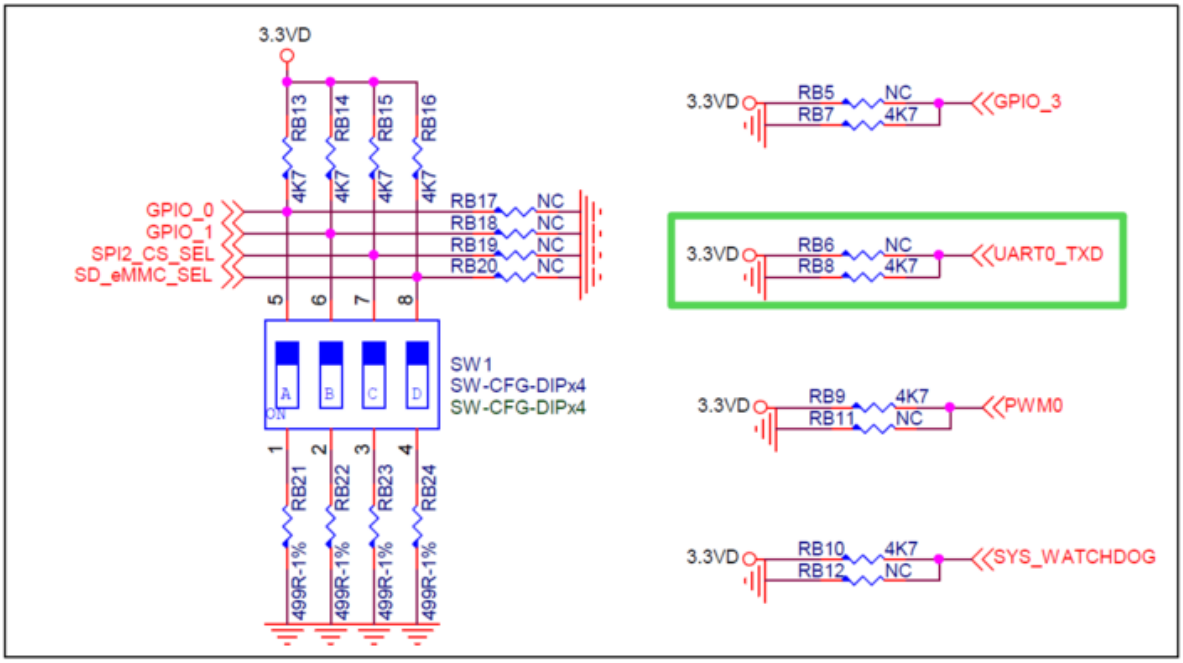
Wifi & Serial cable

- **If the chip type of serial cable is pl2303, the driver fails to load the firmware apparently and thus the wifi can't work.**
- **Other types including cp2102,ch340 and FDTI are all available, serial cable vlotage must be 3.3v LVTTL standard.**

UART_TX0 is the Boot Strapping PIN and must be kept low during power-on.

Boot Strapping

Pin Name	Description	Boot Sequence
{GPIO_1,GPIO_0}	Boot mode	00: SPIM-NOR (SPI2) 01: SPIM-NAND (SPI2) 10: eMMC 5.1(1.8V) 11: SNFI-SD Card
{GPIO_3}	A-Die Mode	0: 2 A-Die/1 Adie SB 1: 1 A-die DBDC
UART_TX0	Second A Die Xtal mode	0 : XTAL mode 1 : Buffer mode
PWM0	40MHz/80MHz Xtal select	0 : Reserved 1 : 40MHz
SYS_WATCHDOG	CPU 1.023V/ 0.85V select	0 : 0.85V 1 : 1.023V



FAQ

- MT7986a Reference Manual for Develope Board(BPi)

  Google Drive：https://drive.google.com/file/d/1biSJmxnIpNzQroYDg9mtPtSTAv4i0DFf/view?usp=sharing

TTL Voltage

The debug-uart TTL is tolerant to 3.3V.

Reference Link

http://forum.banana-pi.org/

Retrieved from "https://wiki.banana-pi.org/index.php?title=Getting_Started_with_BPI-R3&oldid=13878"

- This page was last edited on 26 February 2023, at 18:50.