

Openwrt sur BananaPi R64

Table of Contents

- [Openwrt sur BananaPi R64](#)
 - [Installation](#)
 - [Graver l'image sur une carte SD](#)
 - [Graver l'image sur eMMC intégré](#)
 - [Configuration réseau](#)
 - [Utilisation des interfaces PCI](#)
 - [Utilisation de Sata](#)
 - [Utilisation PCIe](#)
 - [GPIO Control](#)
 - [PWM Control](#)
 - [SPI touch panel](#)
 - [SPI Panel <-> BPIR64](#)
 - [DTS Modification:](#)
 - [Kernel config:](#)
 - [Application:](#)

Cet article (inspiré de https://wiki.banana-pi.org/Getting_Started_with_R64) présente l'installation et la configuration de OWRT sur un routeur Bananpi R64.

Installation

Graver l'image sur une carte SD

1. Télécharger la dernière image 2. Installer bpi-tools sur Ubuntu:

```
apt-get install pv  
curl -sL https://github.com/BPI-SINOVOIP/bpi-tools/raw/master/bpi-tools |  
sudo -E bash
```

3. Après avoir téléchargé l'image, insérer la carte TF, puis Exécuter `bpi-copy xxx.img /dev/sdx` pour installer l'image sur la carte TF

4. Après l'étape 3, on peut insérer la carte TF dans R64 et appuyer sur le bouton d'alimentation pour configurer R64

Graver l'image sur eMMC intégré

Avant de graver l'image sur eMMC, préparer une carte SD avec une image de démarrage flashée et un disque USB(par exemple, mtk-bpi-r64-preloader-emmc.bin,2020-04-09-OpenWRT-mtk-bpi-r64-EMMC.img) les étapes sont ci-dessous :

1. Insérer la carte SD flashée et allumer pour démarrer la carte. (l'image sur la carte SD peut être OpenWrt ou un autre système d'exploitation Linux comme Ubuntu...)
2. Copier l'image OpenWrt amorçable eMMC (mtk-bpi-r64-preloader-emmc.bin, 2020-04-09-OpenWRT-mtk-bpi-r64-EMMC.img) sur le disque USB, si l'image est compressée, il faut la décompresser avant de copier sur un disque USB.
3. Brancher le disque USB sur la carte et monter l'USB sur /mnt ou un autre répertoire comme suit : (on peut ignorer le montage s'il est monté automatiquement)

```
mount -t vfat /dev/sda1 /mnt
cd /mnt
```

1. Exécuter les commandes suivantes pour activer et copier l'image sur eMMC :

```
echo 0 > /sys/block/mmcblk0boot0/force_ro
dd if=2020-04-09-OpenWRT-mtk-bpi-r64-EMMC.img of=/dev/mmcblk0
dd if=mtk-bpi-r64-preloader-emmc.bin of=/dev/mmcblk0boot0
mmc bootpart enable 1 1 /dev/mmcblk0
```

1. Arrêter, retirer la carte SD et le disque USB et redémarrer la carte depuis eMMC.

Configuration réseau

L'interface LAN eth est eth2, utiliser la commande suivante pour l'activer:

```
ifconfig eth2 up
```

Configurer l'adresse IP:

```
ifconfig eth2 192.168.1.1".
```

Configurer le serveur DHCP, vim /etc/dhcp/dhcpd.conf, ajouter ces configurations.

```
#subnet 192.168.2.0 netmask 255.255.255.0{
# range 192.168.2.2 192.168.2.255;
# option domain-name-servers 8.8.8.8;
# option routers 192.168.2.1;
#}
subnet 192.168.1.0 netmask 255.255.255.0{
    range 192.168.1.2 192.168.1.255;
    option domain-name-servers 8.8.8.8;
    option routers 192.168.1.1;
}
```

```
# Cette déclaration permet aux clients BOOTP d'obtenir des adresses dynamiques,  
# ce qui n'est pas vraiment recommandé.
```

Démarrer le serveur DHCP:

```
dhcpd eth2
```

Puis configurer iptables et activer forward.

- ajouter `net.ipv4.ip_forward=1` à `/etc/sysctl.conf` * activer le forward:
`/sbin/sysctl -p` * définir la règle de NAT iptables `-t nat -A POSTROUTING -s 192.168.1.1/24 -o eth3 -j MASQUERADE`

Utilisation des interfaces PCI

Utilisation de Sata

Pour utiliser l'interface Sata sur R64, il faut désactiver GPIO90:

```
echo 499 > /sys/class/gpio/export  
echo out > /sys/class/gpio/gpio499/direction  
echo 0 > /sys/class/gpio/gpio499/valeur
```

Utilisation PCIe

Pour utiliser l'interface PCIe sur R64, il faut activer GPIO90:

```
echo 499 > /sys/class/gpio/export  
echo out > /sys/class/gpio/gpio499/direction  
echo 1 > /sys/class/gpio/gpio499/value
```

GPIO Control

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

Par exemple : si pour activer gpio 22, utiliser les commandes comme celle-ci

```
echo 431[22+409] > /sys/class/gpio/export  
echo out > /sys/class/gpio/gpio431/direction
```

```
echo 1 > /sys/class/gpio/gpio431/value
```

PWM Control

```
echo x >/sys/class/pwm/pwmchip0/export
echo 200000 >/sys/class/pwm/pwmchip0/pwmx/period
echo 100000 >/sys/class/pwm/pwmchip0/pwmx/duty_cycle
echo 1 >/sys/class/pwm/pwmchip0/pwmx/enable
```

SPI touch panel

SPI Panel module:

2.4" Touch Screen TFT LCD with SPI Interface, 240×320 (ILI9341 + ADS7843/XPT2046/HR2046)

SPI Panel <-> BPIR64

```
T_D0, T_DIN, T_CLK <=> SPIC_0: MOSI / MISO / CLK
T_CS <=> SPI-CE0
T_IRQ <=> IO-37
SD0, SCK, SDI <=> SPIC_1: MOSI / MISO / CLK
LED <=> PIN-31
DC <=> PIN-11
RESET <=> PIN-13
CS <=> SPI-CE1
GND <=> GND-9
VCC <=> 3.3V-1
```

DTS Modification:

```
/ {
    backlight: backlight {
        compatible = "gpio-backlight";
        gpios = <&pio 82 GPIO_ACTIVE_HIGH>;           //PIN31 IO-31 :
GPI082
        default-on;
    };
};
```

```
&pio {
    spic0_pins: spic0-pins {
        mux {
            function = "spi";
        }
    };
};
```

```

        groups = "spic0_0";
    };
};

spic1_pins: spic1-pins {
    mux {
        function = "spi";
        groups = "spic1_0";
    };
};
}

```

```

&spi0 {
    pinctrl-names = "default";
    pinctrl-0 = <&spic0_pins>;
    status = "okay";
    touch@0 {
        reg = <0>;                                //CE0
        compatible = "ti,ads7843";
        interrupt-parent = <&pio>;
        interrupts = <86 0>;                        //PIN37: IO-37 ==
GPI086
        pendown-gpio = <&pio 86 0>;
        spi-max-frequency = <1000000>;
        vcc-supply = <&reg_3p3v>;
        wakeup-source;
    };
};

```

```

&spi1 {
    pinctrl-names = "default";
    pinctrl-0 = <&spic1_pins>;
    status = "okay";
    display@0{
        compatible = "ilitek,ili9341";
        reg = <0>;                                //CE0
        spi-max-frequency = <32000000>;
        dc-gpios = <&pio 51 GPIO_ACTIVE_HIGH>;      //PIN11 UART1-TXD
: GPI051
        reset-gpios = <&pio 52 GPIO_ACTIVE_HIGH>;   //PIN13 UART1-RXD
: GPI052
        backlight = <&backlight>;
    };
};

```

Kernel config:

```
+CONFIG_FB_TFT_ILI9341
```

```
+CONFIG_FB_TFT
+CONFIG_FB
+CONFIG_BACKLIGHT_LCD_SUPPORT
+CONFIG_BACKLIGHT_CLASS_DEVICE
+CONFIG_BACKLIGHT_GPIO
+CONFIG_INPUT
+CONFIG_INPUT_TOUCHSCREEN
+CONFIG_TOUCHSCREEN_ADS7846
```

Application:

Package	Description	Source
fbv	framebuffer image viewer	https://github.com/godspeed1989/fbv
input-event-daemon	input-event-daemon with touchTEST event	https://github.com/SAM33/input-event-daemon

Exemple de configuration input-event-daemon qui affiche l'image par zone tactile :

```
[Global]
listen = /dev/input/event0
listen = /dev/input/event1
[TouchTEST]
340,400,3440,1860 = cat /dev/zero > /dev/fb0; fbv -f /root/bpi_608x429.jpg -s 1
340,2260,3440,1860 = cat /dev/zero > /dev/fb0; fbv -f /root/openwrt_449x449.png -s 1
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=prive:bpi-owrt>

Last update: **2025/02/19 10:59**

