

密级状态：绝密(     )     秘密(     )     内部(     )     公开( ☒ )

## RKXX\_Android 4.4&5.0 Kernel 3.10 WiFi BT 配置说明

文件状态：  [   ] 正在修改  [ <input checked="" type="checkbox"/> ] 正式发布	当前版本：	V1.0
	作     者：	胡卫国
	完成日期：	2015-01-09
	审     核：	
	完成日期：	

福州瑞芯微电子有限公司

Fuzhou Rockchips Semiconductor Co., Ltd

(版本所有, 翻版必究)

## 版 本 历 史

版本号	作者	修改日期	修改说明	备注
V1.0	胡卫国	2015-01-09	从 4.4 版本修改过来	
V1.1	胡卫国	2015-02-10	增加特殊配置部分	

## 目 录

<b>1 RK WIFI BT 支持说明.....</b>	<b>3</b>
<b>2 ANDROID 部分配置.....</b>	<b>4</b>
<b>3 KERNEL 部分配置.....</b>	<b>6</b>
3.1 KERNEL WIFI BT 框架相关部分.....	6
3.1.1 WIRELESS 无线协议部分.....	6
3.1.2 BLUETOOTH 协议部分.....	6
3.1.3 RF SWITCH SUBSYSTEM SUPPORT.....	7
3.1.4 WIFI 驱动配置.....	7
3.1.4.1 RK903 & RK901 & AP6XXX 系列配置.....	8
3.1.4.2 REALTEK 系列配置.....	9
3.1.4.3 ESP8089 系列配置.....	10
<b>4 其它特殊配置.....</b>	<b>10</b>
4.1 SDIO 工作在 3.0.....	10
4.2 提高 BT UART 波特率.....	10

## 1 RK WiFi BT 支持说明

目前 Android 4.4 & 5.0 Kernel 3.10 SDK 支持 Braodcom,Realtek、ESP 等 WiFi BT 模块，已经验证过模块如下：

	模块	状态
Broadcom	RK901&RK903	支持
	AP6xxx	支持
Realtek	RTL8188/92 系列	支持
	RTL8723BS 系列	支持
	RTL8723BU	支持
ESP	ESP8089	支持

其它模块后续支持后，会及时发布出去。

**注意：**各个 WiFi BT 模块无法做到动态兼容，需要通过特定的配置文件进行配置。

## 1 RK WiFi BT 支持说明

目前 Android 4.4 & 5.0 Kernel 3.10 SDK 支持 Braodcom,Realtek、ESP 等 WiFi BT 模块，已经验证过模块如下：

	模块	状态
Broadcom	RK901&RK903	支持
	AP6xxx	支持
Realtek	RTL8188/92 系列	支持
	RTL8723BS 系列	支持
	RTL8723BU	支持
ESP	ESP8089	支持

其它模块后续支持后，会及时发布出去。

**注意：**各个 WiFi BT 模块无法做到动态兼容，需要通过特定的配置文件进行配置。

## 2 Android 部分配置

配置文件在：device/rockchip/\$(TARGET\_PRODUCT)/wifi\_bt.mk

\$(TARGET\_PRODUCT)为 product 名字，例如 rk3288, rk312x 等。

以下为具体模块对应的配置：

<b>RK903</b>	<b>BOARD_CONNECTIVITY_VENDOR := Broadcom</b>
<b>RK901</b>	<b>BOARD_CONNECTIVITY_MODULE := ap6xxx</b>
<b>AP6210</b> <b>AP6330</b> <b>AP6181</b>	<b>BOARD_CONNECTIVITY_VENDOR := Broadcom</b> <b>BOARD_CONNECTIVITY_MODULE := ap6xxx</b>
<b>AP6476</b>	<b>BOARD_CONNECTIVITY_VENDOR := Broadcom</b> <b>BOARD_CONNECTIVITY_MODULE := ap6xxx_gps</b>
<b>RTL8188/92</b>	<b>BOARD_CONNECTIVITY_VENDOR := RealTek</b> <b>BOARD_CONNECTIVITY_MODULE := rtl81xx</b>
<b>RTL8723BS</b>	<b>BOARD_CONNECTIVITY_VENDOR := RealTek</b> <b>BOARD_CONNECTIVITY_MODULE := rtl8723bs</b>
<b>RTL8723BU</b>	<b>BOARD_CONNECTIVITY_VENDOR := RealTek</b> <b>BOARD_CONNECTIVITY_MODULE := rtl8723bu</b>
<b>RTL8723BS-VQ0</b>	<b>BOARD_CONNECTIVITY_VENDOR := RealTek</b> <b>BOARD_CONNECTIVITY_MODULE := rtl8723bs_vq0</b>
<b>ESP8089</b>	<b>BOARD_CONNECTIVITY_VENDOR := Espressif</b> <b>BOARD_CONNECTIVITY_MODULE := esp8089</b>

**注意：**配置后需要通过以下脚本来重新编译才能生效。

Android 4.4:

```
cd device/rockchip/rksdk/
```

Android 5.0

```
cd device/rockchip/common/
```

```
./wifi_bt_build.sh
```

这个 sh 脚本其实就是去重新单独 mmm 受上面两个配置控制的模块。

如果编译出错，或者升级进去后 WiFi 无法正常运行，最好重新 `make clean` 再编译。

## 3 Kernel 部分配置

### 3.1 kernel WiFi BT 框架相关部分

这部分是 RK SDK 默认打开的，一般情况下不需要再去配置。

#### 3.1.1 Wireless 无线协议部分

```
-- wireless
-*   cfg80211 - wireless configuration API
-*   nl80211 testmode command
[ ]   enable developer warnings
[ ]   cfg80211 regulatory debugging
[ ]   cfg80211 certification onus
[*]   enable powersave by default
[ ]   cfg80211 DebugFS entries
[ ]   use statically compiled regulatory rules database
[*]   cfg80211 wireless extensions compatibility
[ ]   Allow reconnect while already connected
-*   Generic IEEE 802.11 Networking Stack (mac80211)
[ ]   PID controller based rate control algorithm
[*]   Minstrel
[*]   Minstrel 802.11n support
[ ]   Default rate control algorithm (Minstrel) --->
[ ]   Enable mac80211 mesh networking (pre-802.11s) support
[ ]   Enable LED triggers
[ ]   Export mac80211 internals in DebugFS
[ ]   Trace all mac80211 debug messages
[ ]   Select mac80211 debugging features --->
```

#### 3.1.2 Bluetooth 协议部分

在 Android 4.4 版本以后，bluetooth 协议部分全部在 Android 层 bluedroid 中实现，所以其实已经不需要

以下部分配置。由于部分 BT 测试工具可能用到此部分配置，所以目前还需要保留。

```
-- Bluetooth subsystem support
<*>  RFCOMM protocol support
[*]   RFCOMM TTY support
<*>  BNEP protocol support
[*]   Multicast filter support
[*]   Protocol filter support
<*>  HIDP protocol support
[*]   Bluetooth device drivers --->
```



```
< > HCI USB driver
< > RTK HCI USB driver
< > HCI SDIO driver
<*> HCI UART driver
[*]   UART (H4) protocol support
[*]   BCSP protocol support
[*]   Atheros AR300x serial support
[*]   HCILL protocol support
[*]   Three-wire UART (H5) protocol support
< > HCI BCM203x USB driver
< > HCI BPA10x USB driver
< > HCI BlueFRITZ! USB driver
< > HCI VHCI (Virtual HCI device) driver
< > Marvell Bluetooth driver support
```

### 3.1.3 RF switch subsystem support

WiFi BT 电源控制驱动。

BT 使用标准的 rfkill 驱动框架，上层通过操作 rfkill 节点来控制 BT 芯片的电源。

WiFi 并没有使用标准的 rfkill 驱动框架，只是其电源控制部分驱动放于 rfkill 目录中，WiFi 驱动中直接调用私有电源控制接口。

```
-- RF switch subsystem support
[*]   Power off on suspend
[*]   RF switch input support
< >   Generic rfkill regulator driver
< >   GPIO RFKILL driver
[*]   Rockchips RFKILL driver
```

WiFi 电源控制驱动位于：net/rfkill/rfkill-wlan.c

BT 电源控制驱动位于：net/rfkill/rfkill-bt.c

其对应 dts 相关配置说明参考文档《RK 3.10 版本 Kernel WiFi&BT\_开发配置参考说明.pdf》

### 3.1.4 WiFi 驱动配置

WiFi 驱动代码位于：drivers/net/wireless/rockchip\_wlan/

```
-- Wireless LAN
[*]   Realtek Wireless Device Driver Support --->
[*]   Espressif 8089 sdio wi-Fi
[*]   Espressif 8089 sdio wi-Fi combo with BK3515A bluetooth
<*>   RK901/RK903/BCM4330/AP6xxx wireless cards support
      Select the wifi module (AP6335) --->
      Select the wifi module crystal freq (37_4M) --->
```

默认 SDK 编译了 Realtek、ESP8089、Broadcom AP6xxx 系列三种 WiFi 驱动。通过 dts 以下配置来决定使用那个驱动。

```
wireless-wlan {
    compatible = "wlan-platdata";

    /* wifi_chip_type - wifi chip define
     * bcmwifi ==> like ap6xxx, rk90x;
     * rtlwifi ==> like rt1818xx, rt18723xx;
     * esp8089 ==> esp8089;
     * other ==> for other wifi;
     */
    wifi_chip_type = "bcmwifi";
}
```

### 3.1.4.1 RK903 & RK901 & AP6xxx 系列配置

包括以下模块：RK901、RK903、AP6181、AP6210、AP6330、AP6476、AP6335 等

选择 “RK901/RK903/BCM4330/AP6XXX wireless cards support”

```
--- wireless LAN
[*] Realtek wireless Device Driver Support --->
[*] Espressif 8089 sdio wi-Fi
[*] Espressif 8089 sdio wi-Fi combo with BK3515A bluetooth
[*] RK901/RK903/BCM4330/AP6XXX wireless cards support
    Select the wifi module (AP6335) --->
    Select the wifi module crystal freq (37_4M) --->
```

**Select the wifi module:**

选择相应的模块。

**Select the wifi modules crystal freq:**

选择模块使用的外围晶体频率，默认是 26M。

**dts 中配置：** wifi\_chip\_type = "bcmwifi";

**WiFi 电源脚：**

WiFi 电源脚：模块第 12 脚

**WiFi 中断脚：**

需要配置 WiFi 中断脚，不然 WiFi 无法正常工作。对应的 wifi 模块上的管脚为模块第 13 脚：

WL\_HOST\_WAKE。

**BT 电源脚：**

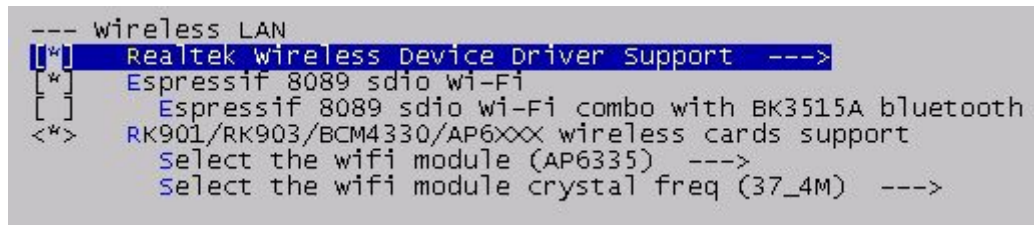
BT\_RST                      模块 34 脚

BT\_WAKE                     模块 6 脚

BT\_HOST\_WAKE               模块 7 脚

### 3.1.4.2 Realtek 系列配置

选择 “Realtek Wireless Device Driver Support”



进去选择相应的芯片



dts 中配置: wifi\_chip\_type = "rtkwifi";

如果使用 RTL8723BU，还需要配置以下 BT 驱动：

```
CONFIG_BT_RTKBTUSB=y
```

如果使用 RTL8723BU/AU，也就是 WiFi BT 二合一 USB 接口的模块。如果模块是电源是硬件常供电的，需要将 dts 中 wireless-wlan、wireless-bluetooth 里以下电源配置去掉：

```
//WIFI,poweren_gpio = <&gpio4 GPIO_D4 GPIO_ACTIVE_HIGH>;  
  
//WIFI,host_wake_irq = <&gpio4 GPIO_D6 GPIO_ACTIVE_HIGH>;  
  
//WIFI,reset_gpio = <&gpio0 GPIO_A2 GPIO_ACTIVE_LOW>;  
  
//BT,power_gpio = <&gpio4 GPIO_D3 GPIO_ACTIVE_HIGH>;  
  
//BT,reset_gpio = <&gpio4 GPIO_D5 GPIO_ACTIVE_HIGH>;
```

如果模块有 GPIO 控制电源开关：可以在 bt rfkill 驱动 probe 时将电源打开，后面不做控制，修改如下：

先按上面配置，再如下修改一点就可

BT,power\_gpio = <&gpio4 GPIO\_D3 GPIO\_ACTIVE\_HIGH>; // 配置为模块的电源 GPIO，将后面的使能标志 “GPIO\_ACTIVE\_HIGH” 相反修改，也就是说：如果是高电平使能，需要设置成低电

平。

### 3.1.4.3 ESP8089 系列配置

选择 “Espressif 8089 sdio Wi-Fi support”

```
--- Wireless LAN
[*] Realtek wireless Device Driver Support --->
[*] Espressif 8089 sdio Wi-Fi
[*] Espressif 8089 sdio Wi-Fi combo with BK3515A bluetooth
<[*> RK901/RK903/BCM4330/AP6XXX wireless cards support
      Select the wifi module (AP6335) --->
      Select the wifi module crystal freq (37_4M) --->
```

dts 中配置: wifi\_chip\_type = "esp8089";

## 4 其它特殊配置

### 4.1 SDIO 工作在 3.0

如果是 AP6335 模块, 主控是 RK3288, 可以让 SDIO 工作在 3.0, 以提高 WiFi 吞吐率, 具体如下 为:

1). dts 里面时钟要提上去

```
&sdio {
    clock-frequency = <150000000>;
    clock-freq-min-max = <200000 150000000>;
};
```

2). CONFIG\_MMC\_DW\_ROCKCHIP\_SWITCH\_VOLTAGE=y

### 4.2 提高 BT UART 波特率

BT UART 默认工作在 1500000, 如果需要调整, 修改如下:

1) dts 文件修改:

```
+++ b/arch/arm/boot/dts/rk3288-tb_8846.dts
@@ -393,6 +393,7 @@ @@
    &uart_bt {
```

```

        status = "okay";
        dma-names = "!tx", "!rx";
+       clock-frequency = <48000000>;
        pinctrl-0 = <&uart0_xfer &uart0_cts>;
    };

```

默认 UART Clock 是 24M，可以让稳定工作在 1.5M 的波特率下。如果需要修改其它波特率，请按以下公式计算出相应的 Clock：

$$\text{clock} = n * 16 * \text{rate} \quad (n = 1, 2, \dots)$$

例如调试波特率为 3M，那么 clock 需要设置为 48M。

## 2) uart 驱动修改：

```

--- a/drivers/tty/serial/rk_serial.c
+++ b/drivers/tty/serial/rk_serial.c
@@ -1251,6 +1251,13 @@ static int serial_rk_startup(struct uart_port *port)
        clk_prepare_enable(up->clk);
        clk_prepare_enable(up->pclk); // enable the config uart clock
    #endif
+
+       retval = clk_set_rate(up->clk, up->port.uartclk);
+       if(retval < 0) {
+           printk("failed to set uart clk\n");

```