

密级状态：绝密() 秘密() 内部() 公开(☒)

RKXX_Android 5.0 Kernel 3.10 不同 WiFi BT 芯片自动兼容版本配置说明

文件状态： <input type="checkbox"/> 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：	2015-02-27
	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	2015-02-27	初始版本	

目 录

1 RK WIFI BT 支持说明.....	4
2 ANDROID 部分配置.....	5
3 KERNEL 部分配置.....	6
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
3.1.4.2.1 USB WIFI BT 二合一模块电源控制.....	10
3.1.4.2.2 RTL8723BS 系列 UART CTS RST 脚配置.....	11
3.1.4.3 ESP8089 系列配置.....	11
4 如何更新 WIFI KO 驱动.....	11
5 其它特殊配置.....	12
5.1 SDIO 工作在 3.0.....	12
5.2 提高 BT UART 波特率.....	12

1 RK WiFi BT 支持说明

目前 Android 5.0 Kernel 3.10 SDK 支持 Braodcom,Realtek 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

其它模块后续支持后，会及时发布出去。

注意：各个 WiFi BT 模块已经做到动态兼容，Android 上层不再需要像以前一样进行特定宏的配置。

因此，后面只需要在 kernel dts 中配置 wifi_chip_type 属性就可，类似如下：

```
wifi_chip_type = "ap6335";
```

1 RK WiFi BT 支持说明

目前 Android 5.0 Kernel 3.10 SDK 支持 Braodcom,Realtek 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

其它模块后续支持后，会及时发布出去。

注意：各个 WiFi BT 模块已经做到动态兼容，Android 上层不再需要像以前一样进行特定宏的配置。

因此，后面只需要在 kernel dts 中配置 wifi_chip_type 属性就可，类似如下：

```
wifi_chip_type = "ap6335";
```

2 Android 部分配置

已经做到各个模块**动态兼容**，不再需要以下文件中的宏配置来区分不同的 wifi, bt 模块（芯片）。

device/rockchip/\${TARGET_PRODUCT}/wifi_bt.mk

\${TARGET_PRODUCT}为 product 名字，例如 **rk3288**, **rk312x** 等。

注意：第一次更新 wifi, bt 模块（芯片）动态兼容的代码后，需要 make clean 再 make 一次。

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 8188EU USB WiFi Support
<M> Realtek 8189ES/ETV SDIO WiFi Support
<M> Realtek 8192DU USB WiFi Support
<M> Realtek 8723AU USB WiFi Support
<M> Realtek 8723BU USB WiFi
<M> Realtek 8723BS SDIO or SPI WiFi
<M> Realtek 8723BS-VQ0 COB WiFi
<M> Realtek 8812AU USB WiFi Support
[*] Espressif 8089 sdio wi-fi
<*> RK901/RK903/BCM4330/AP6XXX wireless cards support
    Select the wifi module (AP6335) --->
    Select the wifi module crystal freq (37_4M) --->
```

默认 kernel buildin 了 Realtek RTL8188EU、ESP8089、Broadcom AP6xxx 系列三种 WiFi 驱动。

其它 Realtek WiFi 驱动如 RTL8723BS, RTL8723BU 等编译成 ko（一方面是因为全部 build in 的话，kernel 会越来越大，二是 realtek 驱动两个同时编译时会出现重定义冲突）。

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
     * ap6210, ap6330, ap6335
     * rtl8188eu, rtl8723bs, rtl8723bu
     * esp8089
     */
    wifi_chip_type = "ap6335";
}
```

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 8188EU USB WiFi Support
<M> Realtek 8189ES/ETV SDIO WiFi Support
<M> Realtek 8192DU USB WiFi Support
<M> Realtek 8723AU USB WiFi Support
<M> Realtek 8723BU USB WiFi
<M> Realtek 8723BS SDIO or SPI WiFi
<M> Realtek 8723BS-VQ0 COB WiFi
<M> Realtek 8812AU USB WiFi Support
[*] Espressif 8089 sdio wi-fi
<*> RK901/RK903/BCM4330/AP6XXX wireless cards support
    Select the wifi module (AP6335) --->
    Select the wifi module crystal freq (37_4M) --->
```

dts 中 wifi_chip_type 配置:

rk901	wifi_chip_type = "rk901";
rk903	wifi_chip_type = "rk903";
ap6181	wifi_chip_type = "ap6181";
ap6210	wifi_chip_type = "ap6210";
ap6234	wifi_chip_type = "ap6234";
ap6330	wifi_chip_type = "ap6330";
ap6335	wifi_chip_type = "ap6335";
ap6441	wifi_chip_type = "ap6441";
ap6476	wifi_chip_type = "ap6476";

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”

```
--- wireless LAN
[*]> Realtek 8188EU USB WiFi Support
<M> Realtek 8189ES/ETV SDIO WiFi Support
<M> Realtek 8192DU USB WiFi Support
<M> Realtek 8723AU USB WiFi Support
<M> Realtek 8723BU USB WiFi
<M> Realtek 8723BS SDIO or SPI WiFi
<M> Realtek 8723BS-VQ0 COB WiFi
<M> Realtek 8812AU USB WiFi Support
[*] Espressif 8089 sdio wi-Fi
<*> RK901/RK903/BCM4330/AP6XXX wireless cards support
      Select the wifi module (AP6335) --->
      Select the wifi module crystal freq (37_4M) --->
```

dts 中 wifi_chip_type 配置:

rtl8188eu	wifi_chip_type = "rtl8188eu";
rtl8723au	wifi_chip_type = "rtl8723au";
rtl8723bu	wifi_chip_type = "rtl8723bu";
rtl8723bs	wifi_chip_type = "rtl8723bs";
rtl8723bs_vq0	wifi_chip_type = "rtl8723bs_vq0";
rtl8189es	wifi_chip_type = "rtl8189es";
rtl8812au	wifi_chip_type = "ap6335";

如果使用 RTL8723BU，还需要配置以下 BT 驱动:

```
CONFIG_BT_RTKBTUSB=y
```

3.1.4.2.1 USB WiFi BT 二合一模块电源控制

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。

3.1.4.2.2 RTL8723BS 系列 UART CTS RST 脚配置

RTL8723BS 系列，流控引脚接法注意：

主控 UART_RTS 需要接到模块 CTS 脚

主控 UART_CTS 需要接到模块 CTS 脚或主控 UART_CTS 直接接地，这个脚不能悬空。

3.1.4.3 ESP8089 系列配置

选择“Espressif 8089 sdio Wi-Fi support”

```
--- Wireless LAN
<*> Realtek 8188EU USB wifi support
<M> Realtek 8189ES/ETV SDIO wifi support
<M> Realtek 8192DU USB wifi support
<M> Realtek 8723AU USB wifi support
<M> Realtek 8723BU USB wifi
<M> Realtek 8723BS SDIO or SPI wifi
<M> Realtek 8723BS-VQ0 COB wifi
<M> Realtek 8812AU USB wifi support
[*] Espressif 8089 sdio wi-fi
<*> 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 如何更新 wifi ko 驱动

目前以下 wifi 驱动编译成 ko

CONFIG_RTL8189ES=m

CONFIG_RTL8192DU=m

CONFIG_RTL8723AU=m

CONFIG_RTL8723BU=m

CONFIG_RTL8723BS=m

CONFIG_RTL8723BS_VQ0=m

CONFIG_RTL8812AU=m

若需要更新驱动（以 rtl8723bs 为例），需要按以下步骤进行：

- 1) 修改驱动 drivers/net/wireless/rockchip_wlan/rtl8723bs
- 2) make modules 生成 drivers/net/wireless/rockchip_wlan/rtl8723bs/8723bs.ko
- 3) 拷贝上面生成的 ko 到 vendor/rockchip/common/wifi/modules/8723bs.ko，并更新到机器中

5 其它特殊配置

5.1 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

5.2 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");
```