

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

## RKXX\_Android4.4-SDK\_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：	V1.3
	作     者：	胡卫国
	完成日期：	2013-12-14
	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	2013-12-14	初始版本	
V1.2	胡卫国	2014-01-16	增加模块电源控制说明	
V1.3	胡卫国	2014-02-19	修正一些错误	

## 目 录

<b>1 RK WIFI BT 支持说明.....</b>	<b>5</b>
<b>2 ANDROID 部分配置.....</b>	<b>5</b>
<b>3 KERNEL 部分配置.....</b>	<b>8</b>
<b>3.1 KERNEL WIFI 框架相关部分（所有 WIFI 都需要配置） .....</b>	<b>8</b>
3.1.1 WIFI 接口驱动配置.....	8
3.1.2 WIFI 网络协议栈配置.....	9
3.2 KERNEL 蓝牙框架相关部分.....	9
3.2.1 蓝牙接口驱动配置.....	9
3.2.2 蓝牙框架部分配置.....	9
3.3 WIFI 电源控制.....	10
3.4 RK903 & RK901 & AP6XXX 系列配置.....	12
3.4.1 选择“RK901/RK903/BCM4330/AP6XXX WIRELESS CARDS SUPPORT”.....	12
3.4.2 关闭“NL80211 TESTMODE COMMAND”.....	13
3.4.3 选择串口硬件流控“SERIAL PORT 0 CTS/RTS SUPPORT”.....	13
3.4.4 选择“ENABLE WIFI CONTROL FUNCTION ABSTRACTION”.....	13
3.4.5 WIFI 电源脚.....	13
3.4.6 WIFI 中断脚配置.....	14
3.4.7 BT 电源脚.....	14
3.5 RTL8188 配置.....	14
3.5.1 KERNEL MEMUCONFIG 配置.....	14
3.6 MT5931&MT6622 配置.....	15
3.6.1 WIFI 部分配置.....	15
3.6.2 选择串口硬件流控“SERIAL PORT 0 CTS/RTS SUPPORT”.....	16
3.6.3 WIFI 电源脚.....	18
3.6.4 BT 电源脚.....	18
3.6.5 注意板级文件修改.....	18

3.7 MT6620 && MT6628 && COMBO_MT66XX 配置.....	19
3.7.1 COMBO_MT66XX 所需要的KERNEL 完整配置.....	20
3.7.2 选择串口硬件流控“SERIAL PORT 0 CTS/RTS SUPPORT” .....	22
3.7.3 中断脚配置确认.....	23
3.8 RTL8723AU(USB 接口)配置.....	23
3.8.1 KERNEL MEMUCONFIG 配置.....	23
3.9 RTL8723BS(SDIO 接口)配置.....	25
3.9.1 KERNEL MEMUCONFIG 配置.....	25
3.9.2 WIFI 电源脚.....	26
3.9.3 BT 电源脚.....	27
3.9.4 32.768KHZ RTC CLOCK 配置.....	27
3.10 RDA587X(UART 接口)配置.....	27
3.10.1 KERNEL MEMUCONFIG 配置.....	27
3.10.2 BT 电源脚.....	28
<b>4 常见问题.....</b>	<b>28</b>
4.1.1 蓝牙默认设备名字修改.....	29
4.1.2 如何硬件不支持BT，如何去掉设置中的BT 设置部分.....	29
4.1.3 BT 4.0 支持.....	30

## 1 RK WiFi BT 支持说明

目前 Android 4.4 SDK 支持 Braodcom, Realtek, MTK 等 WiFi BT 模块，具体如下：

	模块	状态
Broadcom	RK901&RK903	支持
	AP6xxx	支持
Realtek	RTL8188/92 系列	支持
	RTL8723AU	支持
	RTL8723BS	支持
	RTL8723BU	支持
MTK	MT5931/MT6622	支持
	MT6620	支持
RDA	RDA5875	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 **rk3188**, **rk3026** 等。

具体配置如下：

**Broadcom** 系列（此为系统默认配置）：

**BOARD\_CONNECTIVITY\_VENDOR := Broadcom**

**BOARD\_CONNECTIVITY\_MODULE := ap6xxx**

(RK903 也可使用这个配置，不需要修改)

注意，使用 AP6476（带 GPS 时），需要配置成：

**BOARD\_CONNECTIVITY\_MODULE := ap6xxx\_gps**

MTK 系列：

**BOARD\_CONNECTIVITY\_VENDOR := MediaTek**

**BOARD\_CONNECTIVITY\_MODULE := mt5931\_6622**

(mt6620 选择“**combo\_mt66xx**”， mt5931 选择“**mt5931\_6622**”)

Realtek 系列：

**BOARD\_CONNECTIVITY\_VENDOR := RealTek**

**BOARD\_CONNECTIVITY\_MODULE := rtl872xU**

(rtl872x 分两种接口：**rtl872x**(SDIO 接口), **rtl872xU**(USB 接口) )

RDA587x 系列：

**BOARD\_CONNECTIVITY\_VENDOR := RealTek**

**BOARD\_CONNECTIVITY\_MODULE := rda587x**

(一般我们使用是 rda587x 蓝牙配 realtek 8188 和 rtl8192 usb wifi，所以将蓝牙配置放在 RealTek 下面，只是 BOARD\_CONNECTIVITY\_MODULE 需要修改)

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>RTL8188/RDA587x</b>	<b>BOARD_CONNECTIVITY_VENDOR := RealTek</b> <b>BOARD_CONNECTIVITY_MODULE := rda587x</b>
<b>RTL8188/MT6622</b>	<b>BOARD_CONNECTIVITY_VENDOR := RealTek</b> <b>BOARD_CONNECTIVITY_MODULE := mt6622</b>
<b>RTL8723AU</b>	<b>BOARD_CONNECTIVITY_VENDOR := RealTek</b> <b>BOARD_CONNECTIVITY_MODULE := rtl8723au</b>
<b>MT5931/MT6622</b>	<b>BOARD_CONNECTIVITY_VENDOR := MediaTek</b> <b>BOARD_CONNECTIVITY_MODULE := mt5931_6622</b>
<b>MT6620</b>	<b>BOARD_CONNECTIVITY_VENDOR := MediaTek</b> <b>BOARD_CONNECTIVITY_MODULE := combo_mt66xx</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>MT7601</b>	<b>BOARD_CONNECTIVITY_VENDOR:=MediaTek_mt7601</b> <b>BOARD_CONNECTIVITY_MODULE := mt7601</b>
<b>MT7601/RDA587x</b>	<b>BOARD_CONNECTIVITY_VENDOR:=MediaTek_mt7601</b> <b>BOARD_CONNECTIVITY_MODULE := rda587x</b>
<b>MT7601/MT6622</b>	<b>BOARD_CONNECTIVITY_VENDOR:=MediaTek_mt7601</b> <b>BOARD_CONNECTIVITY_MODULE := mt6622</b>

**注意：**配置后需要通过以下脚本来重新编译才能生效。

**cd device/rockchip/rksdk/**



`./wifi_bt_build.sh`

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 框架相关部分（所有 WiFi 都需要配置）

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 WiFi 接口驱动配置

##### 1、SDIO 接口

选择“RK29 SDMMC1 controller support(sdio)”

Location:

-> Device Drivers

-> MMC/SD/SDIO card support

-> RK29 SDMMC controller support

```
<*> RK29 SDMMC1 controller support(sdio)
[ ] write-protect for SDMMC1
[ ] sdio-irq from gpio
```

##### 2、USB 接口

选择“Rockchip USB 2.0 host controller”

Location:

-> Device Drivers

-> USB support

```
< > Rockchip USB 1.1 host controller
<*> Rockchip USB 2.0 host controller
[*] ---usb2.0 host controller enable
```

### 3.1.2 WiFi 网络协议栈配置

选择“Wireless”

Location:

-> Networking support

```
--- wireless
<*>  cfg80211 - wireless configuration API
[ ]   nl80211 testmode command
[ ]   enable developer warnings
[ ]   cfg80211 regulatory debugging
[*]   enable powersave by default
[ ]   cfg80211 DebugFS entries
[ ]   use statically compiled regulatory rules database
[*]   cfg80211 wireless extensions compatibility
[*]   wireless extensions sysfs files
< >  Common routines for IEEE802.11 drivers
[ ]   Allow reconnect while already connected
< >  Generic IEEE 802.11 Networking Stack (mac80211)
```

## 3.2

### Kernel 蓝牙框架相关部分

这部分是 RK SDK 默认打开的，一般情况下不需要再去配置。

#### 3.2.1 蓝牙接口驱动配置

1. UART 接口:

选择“Serial port 0 support”

Location:

-> Device Drivers

-> Character devices

-> Serial drivers

-> RockChip RK29/RK30 serial port support

#### 3.2.2 蓝牙框架部分配置

选择以下协议:

[\*] Networking support --->

<\*> Bluetooth subsystem support --->

[\*] L2CAP protocol support

[\*] SCO links support

<\*> RFCOMM protocol support

[\*] RFCOMM TTY support

<\*> BNEP protocol support

<\*> HIDP protocol support

Bluetooth device drivers --->

<\*> HCI UART driver

[\*] UART (H4) protocol support

[\*] HCILL protocol support

<\*> Bluetooth auto sleep

<\*> RF switch subsystem support --->

[\*] Power off on suspend

[\*] Rockchips RFKILL driver

### 3.3 WiFi 电源控制

系统中注册 bcmhdhd\_wlan 设备，WiFi 驱动加载时会调用此设备的以下接口：

```
static struct wifi_platform_data rk29sdk_wifi_control = {
    .set_power = rk29sdk_wifi_power,
    .set_reset = rk29sdk_wifi_reset,
    .set_carddetect = rk29sdk_wifi_set_carddetect,
    .mem_prealloc = rk29sdk_mem_prealloc,
};
```

其中, `rk29sdk_wifi_power` 为 WiFi 电源控制函数，包括上电与下电。

WiFi 电源一般需要控制 power 与 reset 两个脚(例如 braodcom sdio WiFi)，也可能只需要控制 power（例如 realteck usb wifi），相应的 GPIO 配置位置如下：

## 1. RK3168 & RK3188 & RK302x

位于 **board-rk31xx-xx-sdmmc-conf.c**, 需要根据不同的 WiFi 来配置:

```
#define RK30SDK_WIFI_GPIO_POWER_N          RK30_PIN3_PD0    // 电源脚

#define RK30SDK_WIFI_GPIO_POWER_ENABLE_VALUE  GPIO_HIGH      // 有效电平

#define RK30SDK_WIFI_GPIO_RESET_N          RK30_PIN2_PA7     // 复位脚

#define RK30SDK_WIFI_GPIO_RESET_ENABLE_VALUE  GPIO_HIGH      // 有效电平
```

**注意:** 在 RK3188 平台上增加了 **sdio 电平驱动强度的设置**, 有 **1.8V** 与 **3.3V** 两种电压可以配置, 这里设置的电平需要与实际的硬件电平(主控 AP0\_VCC, 模块 VDDIO)相匹配。

```
int rk31sdk_get_sdio_wifi_voltage(void)

{

}
```

## 2. RK3066

位于 **board-rk30-sdk-sdmmc.c**,

先找到以下代码

```
#else //default for RK30,RK3066 SDK

    // refer to file /arch/arm/mach-rk30

    #define WIFI_HOST_WAKE RK30_PIN3_PD2
```

再根据不同的 WiFi 来配置

```
//power

#define RK30SDK_WIFI_GPIO_POWER_N          RK30_PIN3_PC6    // 电源脚

#define RK30SDK_WIFI_GPIO_POWER_ENABLE_VALUE  GPIO_HIGH      // 有效电平

//reset

#define RK30SDK_WIFI_GPIO_RESET_N          RK30_PIN3_PD1     // 复位脚

#define RK30SDK_WIFI_GPIO_RESET_ENABLE_VALUE  GPIO_HIGH      // 有效电平
```

## 3. RK2926 & RK2928

位于 **board-rk2928-sdk-sdmmc.c**,

先找到以下代码

```
#elif defined(CONFIG_ARCH_RK2928) //refer to file ./arch/arm/mach-rk2928/include/mach/iomux.h
```

```
#define WIFI_HOST_WAKE RK2928_PIN3_PC0
```

再根据不同的 WiFi 来配置

```
#define RK30SDK_WIFI_GPIO_POWER_N          RK2928_PIN3_PD2    // 电源脚
```

```
#define RK30SDK_WIFI_GPIO_POWER_ENABLE_VALUE    GPIO_HIGH    // 有效电平
```

```
#define RK30SDK_WIFI_GPIO_RESET_N            RK2928_PIN3_PD5    // 复位脚
```

```
#define RK30SDK_WIFI_GPIO_RESET_ENABLE_VALUE    GPIO_HIGH    // 有郊电平
```

### 3.4 RK903 & RK901 & AP6xxx 系列配置

包括以下模块：RK901、RK903、AP6181、AP6210、AP6330、AP6476、BCM4330 等

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

Location:

-> Device Drivers

-> Network device support

-> Wireless LAN

-> WiFi device driver support

```
--- Wireless LAN
[*] Wireless LAN (IEEE 802.11)
[*] Enable wifi host wake up function
[*] WiFi device driver support (RK901/RK903/BCM4330/AP6XXX wireless cards support) --
    RK901/RK903/BCM4330/AP6XXX wireless cards support
[*] Enable NL80211 support
    Select the wifi module (RK903) --->
    Select the wifi module crystal freq (26M) --->
[ ] MTK wireless chip configuration --->
```

注意：[\*] Enable wifi host wake up function 这个选项从 4.4 开始一律要求必须配置。

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，绝大部分情况下不需要修改。

**3.4.2 关闭“nl80211 testmode command”**

Location:

- > Networking support
- > Wireless
- > cfg80211 - wireless configuration API

**3.4.3 选择串口硬件流控“Serial port 0 CTS/RTS support”**

Location:

- > Device Drivers
- > Character devices
- > Serial drivers
- > RockChip RK29/RK30 serial port support
- > Serial port 0 support

RK3066、RK292x、RK31xx 默认蓝牙使用的是 UART0，如果硬件上有变化，配置时需要选择对应的 UART。

**Serial port 0 DMA support** 里的值需要修改成 0，其它值只在软件流控时使用。

**3.4.4 选择“Enable WiFi control function abstraction”**

Location:

- > System Type
- > wifi control func Type

### 3.4.5 WiFi 电源脚

WiFi 电源脚：模块第 12 脚

### 3.4.6 WiFi 中断脚配置

4.4 驱动升到 OOB 方式，需要配置 WiFi 中断脚，不然 WiFi 无法正常工作。

在 board 相应文件中（例如 arch/arm/mach-rk30/board-rk3168-tb-sdmmc-conifg.c）配置

```
#define RK30SDK_WIFI_GPIO_WIFI_INT_B          RK30_PIN3_PD2

#define RK30SDK_WIFI_GPIO_WIFI_INT_B_ENABLE_VALUE  GPIO_HIGH
```

对应的 wifi 模块上的管脚为模块第 13 脚：WL\_HOST\_WAKE。

### 3.4.7 BT 电源脚

在板级文件中的 rkill\_rk\_platdata 中配置：

BT_REG_ON	INVALID_GPIO
BT_RST	模块 34 脚
BT_WAKE	模块 6 脚
BT_HOST_WAKE	模块 7 脚

## 3.5 RTL8188 配置

包括以下模块：RTL8192CU、RTL8188CTV、RTL8188EU、RTL8188ETV、RTL8192DU。

其中 RTL8192CU、RTL8188CTV 使用“**Realtek 8192CU USB WiFi Support**”。

RTL8188EU、RTL8188ETV 使用“**Realtek 8188EU USB WiFi Support**”

RTL8192DU 使用驱动“**Realtek 8192DU USB WiFi Support**”

### 3.5.1 Kernel Memuconfig 配置

1、 选择“**Realtek 8192CU USB WiFi Support**” 或

“**Realtek 8188EU USB WiFi Support**” 或

### “Realtek 8192DU USB WiFi Support”

Location:

- > Device Drivers
- > Network device support
- > Wireless LAN
- > WiFi device driver support

### 2、 关闭“nl80211 testmode command”

Location:

- > Networking support
- > Wireless
- > cfg80211 - wireless configuration API

### 3、 选择“Enable WiFi control function abstraction”

Location:

- > System Type
- > wifi control func Type

## 3.6 MT5931&MT6622 配置

### 3.6.1 WiFi 部分配置

#### 1) 选择“No WiFi”

Location:

- > Device Drivers
- > Network device support (NETDEVICES [=y])
- > Wireless LAN (WLAN [=y])
- > WiFi device driver support (<choice> [=y])



2) 选择“MediaTek MT5931 Config“

Location:

-> Device Drivers

-> Network device support (NETDEVICES [=y])

-> Wireless LAN (WLAN [=y])

-> MTK wireless chip configuration (MTK\_WIRELESS\_SOLUTION [=y])

3) 选择“MediaTek Discrete BT Chip MT6622“

Location:

-> Device Drivers

-> Misc devices (MISC\_DEVICES [=y])

-> MTK wireless chip configuration (MTK\_WIRELESS\_SOLUTION [=y])

4) 去掉选择 “RF switch subsystem support”

CONFIG\_RFKILL=n

Location:

-> Networking support (NET [=y])

-> RF switch subsystem support

5) 选择“**Enable WiFi control function abstraction**”

Location:

-> System Type

-> wifi control func Type

### 3.6.2 选择串口硬件流控“**Serial port 0 CTS/RTS support**”

Location:

-> Device Drivers

-> Character devices

-> Serial drivers

-> RockChip RK29/RK30 serial port support

-> Serial port 0 support

RK3066、RK292x、RK31xx 默认蓝牙使用的是 UART0，如果硬件上有变化，配置时需要选择对应的 UART。

### 1) UART 硬件流控

**CONFIG\_UART0\_CTS\_RTS\_RK29=y**

**CONFIG\_UART0\_DMA\_RK29=0**

**注意：**MT6622 的硬件需要特定接法：

a) 将主控的 UART0\_RTS 接到 6622 模块的 CTS 脚上。

b) 由于 MT6622 的 RTS 脚可能输出一直为高电平，导致主控串口无法通信。需要修改：将 MT6622 模块端的 RTS 悬空，主控端的 UART0\_CTS 接地。

### 2) UART 软件流控

**注意：**如果硬件上没有按以上接法连接，又不能修改硬件的情况下，可配置成 **UART 软件流控**，但是这可能引起 BT 传送大文件时不稳定，造成传送中断。具体配置为

**CONFIG\_UART0\_CTS\_RTS\_RK29=n**

**CONFIG\_UART0\_DMA\_RK29=2**

**注意：**根据配置成 UART 硬件流控与软件流控，Android 上需要作相应的修改(修改升级后需要恢复一下出厂设置才生效，因为有些保存在 data 分区里的信息需要清除掉)：

修改后可通过 `mmm hardware/mediatek/bt/mt5931_6622/` 来编译生效

```
diff --git a/bt/mt5931_6622/custom/cust_bt.h b/bt/mt5931_6622/custom/cust_bt.h
```

```
index f29c2c8..29257ee 100755
```

```

--- a/bt/mt5931_6622/custom/cust_bt.h

+++ b/bt/mt5931_6622/custom/cust_bt.h

@@@ -49,7 +49,7 @@@

/* BAUDRATE */

#define CUST_BT_BAUD_RATE    1500000

/* UART Flow Control */

#define CUST_BT_FLOW_CTL    0           // 0: software flow; 1: hardware flow

/* Low Power Mode */

#define ENABLE_LPM            1           // 0: disable LPM; 1: enable LPM

/* BT Addr Generate */

```

### 3.6.3 WiFi 电源脚

M500 模块:

WiFi 电源脚: 模块 30 脚

### 3.6.4 BT 电源脚

在板级 mt6622\_platdata 中配置:

```

power_gpio    模块第 1 脚

reset_gpio    模块第 22 脚

irq_gpio      模块第 41 脚

```

### 3.6.5 注意板级文件修改

默认的几个板级文件有对应修改，但是有可能会漏掉。这样子会造成 BT 控制不到电源，造成 BT 打不开。具体修改如下（以 board-rk3188-ds1006h.c 为例）

```

--- a/arch/arm/mach-rk3188/board-rk3188-ds1006h.c

```

```
+++ b/arch/arm/mach-rk3188/board-rk3188-ds1006h.c

-#if defined(CONFIG_MT5931_MT6622)

+#if defined(CONFIG_MT5931_MT6622) || defined(CONFIG_MTK_MT6622)

static struct mt6622_platform_data mt6622_platdata = {

    .power_gpio      = { // BT_REG_ON

        #if DS1006H_V1_2_SUPPORT

@@ -1267,7 +1267,7 @@ static struct platform_device *devices[] __initdata = {

    #ifndef CONFIG_GPS_RK

        &rk_device_gps,

    #endif

-#ifndef CONFIG_MT5931_MT6622

+#if defined(CONFIG_MT5931_MT6622) || defined(CONFIG_MTK_MT6622)

        &device_mt6622,

    #endif

    #if defined(CONFIG_MT6629)

@@ -1966,7 +1966,7 @@ static void __init machine_rk30_board_init(void)

    clk_set_rate(clk_get_sys("rk_serial.1", "uart"), 48*1000000);

    #endif

-#if defined(CONFIG_MT5931_MT6622)

+#if defined(CONFIG_MT5931_MT6622) || defined(CONFIG_MTK_MT6622)

    clk_set_rate(clk_get_sys("rk_serial.0", "uart"), 24*1000000);

    #endif

}
```

## 3.7 MT6620 && MT6628 && COMBO\_MT66XX 配置

### 3.7.1 COMBO\_MT66XX 所需要的 kernel 完整配置

1) board 文件中，若缺少相关内容，可以参考 arch/arm/mach-rk30/board-rk3168-tb.c

搜索下 CONFIG\_MTK\_COMBO\_MT66XX 的相关内容。

2) COMBO\_MT66XX 需要的相关 GPIO 设置，

可参考 arch/arm/mach-rk30/board-rk3168-tb-sdmmc-config.c

具体需要各个项目实际情况重新定义的 GPIO 有：

```
#elif defined(CONFIG_MT6620)

#define COMBO_MODULE_MT6620_CDT    0 // to control antsel2, antsel3 and gps_lan foot when using AcSip or Cdtech chip.
                                     //- 1--use Cdtech chip; 0--unuse CDT chip

//power, PMU_EN/hjc test 电源脚

#define RK30SDK_WIFI_GPIO_POWER_N          RK30_PIN0_PB5

#define RK30SDK_WIFI_GPIO_POWER_ENABLE_VALUE    GPIO_HIGH

//reset, DAIRST, SYSRST_B 复位脚

#define RK30SDK_WIFI_GPIO_RESET_N          RK30_PIN3_PD0

#define RK30SDK_WIFI_GPIO_RESET_ENABLE_VALUE    GPIO_HIGH

//VDDIO

//#define RK30SDK_WIFI_GPIO_VCCIO_WL          RK30_PIN0_PD2 //You do not get control of the foot,
                                                         // and you do not need to define the macro

//#define RK30SDK_WIFI_GPIO_VCCIO_WL_ENABLE_VALUE    GPIO_HIGH

//WIFI_INT_B WiFi 中断脚

#define RK30SDK_WIFI_GPIO_WIFI_INT_B          RK30_PIN3_PD2

#define RK30SDK_WIFI_GPIO_WIFI_INT_B_ENABLE_VALUE    GPIO_HIGH

//BGF_INT_B BT 中断脚

#define RK30SDK_WIFI_GPIO_BGF_INT_B          RK30_PIN0_PA5

#define RK30SDK_WIFI_GPIO_BGF_INT_B_ENABLE_VALUE    GPIO_HIGH

//GPS_SYNC

#define RK30SDK_WIFI_GPIO_GPS_SYNC          RK30_PIN3_PD1

#define RK30SDK_WIFI_GPIO_GPS_SYNC_ENABLE_VALUE    GPIO_HIGH

#if COMBO_MODULE_MT6620_CDT

//ANTSEL2

#define RK30SDK_WIFI_GPIO_ANTSEL2          RK30_PIN4_PD4

#define RK30SDK_WIFI_GPIO_ANTSEL2_ENABLE_VALUE    GPIO_LOW //use 6620 in CDT chip, LOW--work; High--no
```

```
work.

//ANTSEL3

#define RK30SDK_WIFI_GPIO_ANTSEL3          RK30_PIN4_PD3

#define RK30SDK_WIFI_GPIO_ANTSEL3_ENABLE_VALUE    GPIO_HIGH    //use 6620 in CDT chip, High--work; Low--no

work..

//GPS_LAN

#define RK30SDK_WIFI_GPIO_GPS_LAN          RK30_PIN4_PD6

#define RK30SDK_WIFI_GPIO_GPS_LAN_ENABLE_VALUE    GPIO_HIGH    //use 6620 in CDT chip, High--work; Low--no

work..

#endif // #if COMBO_MODULE_MT6620_CDT--#endif

#endif
```

### 3) config 中的详细配置:

#### a) CONFIG\_WIFI\_COMBO\_MODULE\_CONTROL\_FUNC=y

-> System type

-> Wifi control func Type

-> Enable WiFi\_combo\_module control function abstraction

#### b) CONFIG\_MT6620=y

-> Device drivers

-> Network device support

-> Wireless LAN

-> Wireless LAN (IEEE 802.11)

-> MTK MT6620 WiFi SDIO

#### c) CONFIG\_MTK\_COMBO\_MT66XX=y

-> Device drivers

-> Network device support

-> Wireless LAN

-> MTK wireless chip configuration

-> MediaTek combo\_mt66xx Config

--- MediaTek combo\_mt66xx WiFi Config

<\*> MediaTek Combo Chip Wi-Fi support

d) CONFIG\_MTK\_COMBO\_WIFI=y

-> Device drivers

-> Misc devices

-> MTK wireless chip configuration

-> MediaTek combo\_mt66xx Config

MediaTek combo\_mt66xx Config

<\*> MediaTek Combo Chip wireless managment tool

<\*> MediaTek Combo Chip BT driver

<\*> MediaTek Combo Chip FM driver

<\*> MediaTek GPS Support

e) CONFIG\_NL80211\_TESTMODE=y

-> Networking support

-> Wireless

-> nl80211 testmode command

f) CONFIG\_RFKILL=n

-> Networking support (NET [=y])

-> RF switch subsystem support

g) uart 的相关设置，请参考下面”选择串口硬件流控”节内容。

### 3.7.2 选择串口硬件流控“Serial port 0 CTS/RTS support”

Location:

-> Device Drivers

-> Character devices

-> Serial drivers

-> RockChip RK29/RK30 serial port support

-> Serial port 0 support

RK3066、RK292x、RK31xx 默认蓝牙使用的是 UART0，如果硬件上有变化，配置时需要选择对应的 UART。

**1) UART 硬件流控**（硬件上有接 UART\_CTS, UART\_RTS 到主控）

**CONFIG\_UART0\_CTS\_RTS\_RK29=y**

**CONFIG\_UART0\_DMA\_RK29=0**

**2) UART 软件流控**（硬件上没有接 UART\_CTS, UART\_RTS 到主控）

**注意：**如果硬件上没有接 UART\_CTS, UART\_RTS 到主控，又不能修改硬件的情况下，可配置成 **UART 软件流控**，但是这可能引起 BT 传送大文件时不稳定，造成传送中断。具体配置为

**CONFIG\_UART0\_CTS\_RTS\_RK29=n**

**CONFIG\_UART0\_DMA\_RK29=2**

另外，软件流控还需要修改如下：

hardware/mediatek/combo\_mt66xx/init.combo\_mt66xx.rc 中去掉红色标出部分

service wmt\_launcher /system/bin/logwrapper /system/bin/wmt\_launcher -b 3000000 -p

/system/etc/firmware/ -d /dev/ttyS0 -c 3

### 3.7.3 中断脚配置确认

WiFi BT 都需要中断脚才能正常工作，可通过以下方式确认中断是否正常（需要有中断才表示正常）：

```
cat proc/interrupts | busybox grep MTK
```

165:	1	0	0	0	GPIO	MTK6620_BT
282:	492	0	0	0	GPIO	MTK_SDIO_EINT

## 3.8 RTL8723AU(USB 接口)配置

### 3.8.1 Kernel Memuconfig 配置

1、选择“Realtek 8723AU USB WiFi Support”



Location:

- > Device Drivers
- > Network device support
- > Wireless LAN
- > WiFi device driver support

## 2、 关闭“nl80211 testmode command”

Location:

- > Networking support
- > Wireless
- > cfg80211 - wireless configuration API

## 3、 选择“Enable WiFi control function abstraction”

Location:

- > System Type
- > wifi control func Type

## 4. Bluetooth 相关配置

### GPIO 配置

目前不需要配置该模块的 GPIO，但要求该模块常供电

### Bluetooth 协议配置

- [\*] Networking support --->
- <\*> Bluetooth subsystem support --->
- [\*] L2CAP protocol support
- [\*] SCO links support
- <\*> RFCOMM protocol support

```

[*]      RFCOMM TTY support

<*>     BNEP protocol support

<*>     HIDP protocol support

Bluetooth device drivers --->

<*>     RTK HCI USB driver
  
```

不需要配置 rfkill / UART 等的配置，如果这些配置已经选上也没有关系，RTL8723AU 不会用到这些。

**选择 BT 驱动: CONFIG\_BT\_RTKBTUSB**

**“RTK HCI USB driver”**

Location:

```

|      -> Networking support (NET [=y])

|      -> Bluetooth subsystem support (BT [=y])

|      -> Bluetooth device drivers
  
```

## 3.9 RTL8723BS(SDIO 接口)配置

### 3.9.1 Kernel Memuconfig 配置

1、选择“**Realtek 8723BS SDIO WiFi Support**”

Location:

```

-> Device Drivers

    -> Network device support

        -> Wireless LAN

            -> WiFi device driver support
  
```

2、关闭“**nl80211 testmode command**”

Location:

```

-> Networking support
  
```

-> Wireless

-> cfg80211 - wireless configuration API

### 3、 选择“**Enable WiFi control function abstraction**”

Location:

-> System Type

-> wifi control func Type

### 4、 选择串口硬件流控“**Serial port 0 CTS/RTS support**”

Location:

-> Device Drivers

-> Character devices

-> Serial drivers

-> RockChip RK29/RK30 serial port support

-> Serial port 0 support

RK3066、RK292x、RK31xx 默认蓝牙使用的是 UART0，如果硬件上有变化，配置时需要选择对应的 UART。

**Serial port 0 DMA support** 里的值需要修改成 0，其它值只在软件流控时使用。

### 5. 选择 RFKILL

<\*> RF switch subsystem support --->

[\*] Power off on suspend

[\*] Rockchips RFKILL driver

## 3.9.2 WiFi 电源脚

WiFi 电源脚：模块第 12 脚

需要配置 WiFi 中断脚。

在 board 相应文件中（例如 arch/arm/mach-rk30/board-rk3168-tb-sdmmc-config.c）配置

```
#define RK30SDK_WIFI_GPIO_WIFI_INT_B          RK30_PIN3_PD2

#define RK30SDK_WIFI_GPIO_WIFI_INT_B_ENABLE_VALUE  GPIO_HIGH
```

对应的 wifi 模块上的管脚为模块第 13 脚：WL\_HOST\_WAKE。

### 3.9.3 BT 电源脚

在板级文件中的 rtkill\_rk\_platdata 中配置：

BT_REG_ON	INVALID_GPIO
BT_RST	模块 34 脚
BT_WAKE	模块 6 脚
BT_HOST_WAKE	模块 7 脚

### 3.9.4 32.768KHz RTC Clock 配置

RTL8723BS 模块内部自带了 32.768K RTC Clock，所以不需要外部供了。

需要按如下配置：

1. 硬件上不要供给模块 32.768K Clock，如果有供给，那么需要断开
2. 软件上修改配置如下：

```
device/rockchip/common/bluetooth/rk30_bt.mk
```

```
USE_EXT_32K_CLK := false
```

## 3.10 RDA587x(UART 接口)配置

### 3.10.1 Kernel Memuconfig 配置

- 1、选择“TCC Bluetooth dev Control power”

Location:

-> Device Drivers

-> Misc devices

-> TCC Bluetooth dev Control power

## 2、 关闭“Rockchips RFKILL driver”

Location:

-> Networking support

-> RF switch subsystem support

-> Rockchips RFKILL driver

## 4、 选择串口硬件流控“Serial port 0 CTS/RTS support”

Location:

-> Device Drivers

-> Character devices

-> Serial drivers

-> RockChip RK29/RK30 serial port support

-> Serial port 0 support

RK3066、RK292x、RK31xx 默认蓝牙使用的是 UART0，如果硬件上有变化，配置时需要选择对应的 UART。

**Serial port 0 DMA support** 里的值需要修改成 0，其它值只在软件流控时使用。

### 3.10.2 BT 电源脚

在板级文件中的 rkill\_rk\_platdata 中配置：

power\_gpio:                LDO 连接的 pin 脚

wake\_host\_gpio:            唤醒主控睡眠

## 4 常见问题

### 4.1.1 蓝牙默认设备名字修改

针对不同的模块具体修改如下：

**MT6622:**

hardware/mediatek/bt/mt5931\_6622/bdroid\_buildcfg.h

**MT6620:**

hardware/mediatek/bt/combo\_mt66xx/bdroid\_buildcfg.h

**RTL8723BS:**

device/common/bluetooth/libbt\_rtk8723bs/bdroid\_buildcfg.h

**RK903, AP6xxx, RTL8723:**

device/rockchip/\$(TARGET\_PRODUCT)/bluetooth/bdroid\_buildcfg.h

具体修改如下：

```
#define BTM_DEF_LOCAL_NAME "rk30sdk"
```

修改后重新编译以下目录：

```
mmm external/bluetooth/bluedroid/ -B
```

### 4.1.2 如何硬件不支持 BT，如何去掉设置中的 BT 设置部分

```
sdk_4.4/device/rockchip/common$ git diff
```

```
diff --git a/features/rk-core.xml b/features/rk-core.xml
```

```
index 982d786..f71a457 100644
```

```
--- a/features/rk-core.xml
```

```
+++ b/features/rk-core.xml
```

```
@@ -26,8 +26,6 @@
```

```
<feature name="android.hardware.screen.landscape" />
```

```
<feature name="android.hardware.wifi" />

<feature name="android.hardware.wifi.direct" />

- <feature name="android.hardware.bluetooth" />
- <feature name="android.hardware.bluetooth_le" />

<feature name="android.hardware.usb.accessory" />

<library                                name="com.android.future.usb.accessory"
file="/system/framework/com.android.future.usb.accessory.jar" />

<feature name="android.hardware.usb.host" />
```

#### 4.1.3 BT 4.0 支持

目前已经验证过 Broadcom AP6xxx 系列可以支持 BT4.0（默认支持，不需要特殊配置）。后继 RTL8723 系列、MT6620 也可以支持（待更新验证）。