

密级状态：绝密() 秘密() 内部() 公开(√)

RK_Android 平台 WiFi&BT 模块 MT5931&MT6622 常见问题说明

(技术部, MID 组)

文件状态： [] 正在修改 [√] 正式发布	当前版本：	V1.0
	作 者：	胡卫国
	完成日期：	2014-6-18
	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	2014-6-18	初始版本	

目 录

1 WIFI 调试思路.....	3
1.1 WiFi 打不开.....	3
1.2 SDIO 接口 WiFi 异常.....	5
1.3 机器一直不断重启.....	6
2 BT 调试思路.....	7
2.1 BT 打不开.....	7
2.2 BT 工作不稳定.....	14
3 MT5931 WIFI 常见问题.....	16
3.1 ANDROID 4.2 WFD 花屏问题&WFD 连接问题优化补丁.....	16
3.2 ANDROID 4.2 解决 iPhone 热点连接上后无法上网问题.....	16
3.3 二级休眠时 MT5931 功耗异常问题.....	16
3.4 ANDROID 4.4 BOX 平台出现 WiFi 打不开.....	17
3.5 ANDROID BOX 平台出现 BT 传送大文件异常.....	18
3.6 ANDROID 4.4 BOX 平台 WiFi 扫描到异常名字 AP.....	19
3.7 ANDROID 4.4 WiFi 休眠后唤醒异常.....	20
4 MT6622 BT 常见问题.....	20
4.1 ANDROID 4.4 BLUEDROID 补丁.....	20
4.2 有些 BT 键盘使用异常.....	21
4.3 ANDROID 4.2 平台随机出现 BT 打不开.....	21
4.4 使用单独 MT6622.....	21
5 补丁下载地址.....	22

1 WiFi 调试思路

1.1 WiFi 打不开

出现设置中一直在打开 WiFi 中或者打开 WiFi 后马上弹回关闭状态。

1. 先确定 kernel log 中是否有类似以下打印：

Android 4.2:

```
=====
===== Launching Wi-Fi driver! (Powered by Rockchip) =====
=====
MT5931 SDIO WiFi driver (Powered by Rockchip,Ver 3.24_mac) init.
```

Android 4.4:

```
=====
===== Launching Wi-Fi driver! (Powered by Rockchip) =====
=====
MT5931 SDIO WiFi driver (Powered by Rockchip,Ver 4.02) init.
```

如果没有以上打印，可能是 kernel 或 android 配置不对，或者编译没有生效

Android 4.2 请参考《RK Android 4.2 WiFi BT 配置及常见问题说明 v2.0.pdf》配置；

Android 4.4 请参考《RKXX_Android4.4 WiFi BT 配置说明 V1.3.pdf》配置。

配置好后，Android 最好 make clean 再 make。

Kernel 相关配置如下：

Android 4.2:

CONFIG_WIFI_CONTROL_FUNC=y

CONFIG_RFKILL=n

CONFIG_MT5931_MT6622=y

CONFIG_SDMMC1_RK29=y

CONFIG_UART0_CTS_RTS_RK29=y // UART 硬件流控

CONFIG_UART0_CTS_RTS_RK29=n // UART 软件流控

CONFIG_UART0_DMA_RK29=2

CONFIG_TUN=y

Android 4.4:

CONFIG_WIFI_CONTROL_FUNC=y

CONFIG_RFKILL=n

CONFIG_SDMMC1_RK29=y

CONFIG_UART0_CTS_RTS_RK29=y // UART 硬件流控

CONFIG_UART0_CTS_RTS_RK29=n // UART 软件流控

CONFIG_UART0_DMA_RK29=2

CONFIG_TUN=y

CONFIG_WIFI_NONE=y

CONFIG_MTK_WIRELESS_SOLUTION=y

CONFIG_MTK_MT6622=y

CONFIG_MTK_MT6622_HW_CTL=y

CONFIG_MTK_MT5931=y

CONFIG_MTK_MT5931_WIFI=y

CONFIG_MTK_MT5931_POWER=y

2 再确认 wifi 驱动是否初始化成功?

使用 netcfg 命令查看网络接口状态

netcfg

lo	UP	127.0.0.1/8	0x00000049 00:00:00:00:00:00
sit0	DOWN	0.0.0.0/0	0x00000080 00:00:00:00:00:00
ip6tnl0	DOWN	0.0.0.0/0	0x00000080 00:00:00:00:00:00
p2p0	DOWN	0.0.0.0/0	0x00001002 00:90:4c:33:22:11
wlan0	DOWN	0.0.0.0/0	0x00001002 00:90:4c:11:22:33

通过判断是否注册成功 wlan0 接口来判断 wifi 驱动是否初始化成功，如果 wifi 驱动初始化失败，那么通过《SDIO 通信异常导致 WiFi 无法打开》章节来排查。

3 如果 wifi 驱动加载成功，那么很可能是上层 WiFi 没有配置对，或者代码没有编译对或者没有编译生效造成的。

Android 4.2 请参考《RK Android 4.2 WiFi BT 配置及常见问题说明 v2.0.pdf》配置；

Android 4.4 请参考《RKXX_Android4.4 WiFi BT 配置说明 V1.3.pdf》配置。

配置好后，Android 最好 make clean 再 make。

1.2 SDIO 接口 WiFi 异常

出现如下类似 sdio 出错信息：

```
rk29_sdmmc_command_complete..2935...CMD5(arg=0x0), hoststate=1, errorTimes=1, errorStep=0x1e ! [sdio]
rk29_sdmmc_command_complete..2935...CMD55(arg=0x0), hoststate=1, errorTimes=1, errorStep=0x1e ! [sdio]
rk29_sdmmc_command_complete..2935...CMD55(arg=0x0), hoststate=1, errorTimes=1, errorStep=0x1e ! [sdio]
rk29_sdmmc_command_complete..2935...CMD55(arg=0x0), hoststate=1, errorTimes=1, errorStep=0x1e ! [sdio]
rk29_sdmmc_command_complete..2935...CMD55(arg=0x0), hoststate=1, errorTimes=1, errorStep=0x1e ! [sdio]
rk29_sdmmc_command_complete..2935...CMD1(arg=0x0), hoststate=1, errorTimes=1, errorStep=0x1e ! [sdio]
```

需要排查以下几点：

1. WiFi 模块是否工作起来

以 M500 15x15 邮票孔封装模块为例，需要确认以下部分是否正常：

- 1) 第 3 脚 VBAT，第 5 脚 VIO 供电是否正常
- 2) 第 35 脚 EXT_32K 是否有供 32.768K 的 clock，峰峰值要求为 $0.7 * vddio$ 到 $vddio$
- 3) 第 22 脚 SYSRST_B 需要高电平，在开机时软件会去拉高这个 GPIO 脚
- 4) 第 30 脚 PMU_EN 需要高电平，在打开 WiFi 时软件会去拉高这个 GPIO 脚

（测试时需要注意：在打开 WiFi 时会拉高这个脚，可能是因为打开 WiFi 异常，导致上层去关闭 WiFi，此时这个脚又会被拉低）

5) 如果发现第 30 脚没有被控制到, 确认函数 rk29sdk_wifi_power, 正常情况下有类似如下打印:

```
<4>[ 22.325040] wifi_set_power = 1
<6>[ 22.328137] rk29sdk_wifi_power: 1
```

确认 wifi 电源脚配置是否正确, 函数 rk29sdk_wifi_power 去控制配置的电源脚。

2. 可能是模块的 IO 电平与主控的 SDIO IO 电平不匹配造成的(只针对 RK31x8)

注意: 在 RK31x8 (RK3168/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)
{
}
}
```

1.3 机器一直不断重启

rk29_sdmmc_interrupt 中出现空指针异常

在串口中如下 kernel panic:

```
[ 6.097388] Unable to handle kernel NULL pointer dereference at virtual address 00000380
[ 6.112683] pgd = d77c4000
[ 6.115392] [00000380] *pgd=00000000
[ 6.118980] Internal error: Oops: 5 [#1] PREEMPT SMP
[ 6.123950] CPU: 0 Tainted: G C (3.0.36+ #165)
[ 6.129535] PC is at __raw_spin_lock_irqsave+0x2c/0xb8
[ 6.134678] LR is at try_to_wake_up+0x28/0x354
[ 6.767978] [<c0927208>] (__raw_spin_lock_irqsave+0x2c/0xb8) from [<c0473cd4>] (try_to_wake_up+0x28/0x354)
[ 6.777644] [<c0473cd4>] (try_to_wake_up+0x28/0x354) from [<c0775ec0>] (rk29_sdmmc_interrupt+0x140/0x488)
[ 6.787223] [<c0775ec0>] (rk29_sdmmc_interrupt+0x140/0x488) from [<c04b9eac>]
(handle_irq_event_percpu+0x6c/0x2ac)
[ 6.797582] [<c04b9eac>] (handle_irq_event_percpu+0x6c/0x2ac) from [<c04ba128>] (handle_irq_event+0x3c/0x5c)
[ 6.807419] [<c04ba128>] (handle_irq_event+0x3c/0x5c) from [<c04bc4dc>] (handle_fasteoi_irq+0xbc/0x164)
[ 6.816821] [<c04bc4dc>] (handle_fasteoi_irq+0xbc/0x164) from [<c04b97f8>] (generic_handle_irq+0x28/0x3c)
```



```
[ 6.826400] [<c04b97f8>] (generic_handle_irq+0x28/0x3c) from [<c043d04c>] (asm_do_IRQ+0x4c/0xac)
[ 6.835195] [<c043d04c>] (asm_do_IRQ+0x4c/0xac) from [<c0442cc4>] (__irq_usr+0x44/0xe0)
```

这可能有以下两种原因：

1) 可能是模块的 IO 电平与主控的 SDIO IO 电平不匹配造成的(只针对 RK31x8)

可参考上一节。

2) 可能模块焊接异常，导致 WiFi SDIO DATA1 数据线被异常拉低

针对这个 panic 问题，软件上可通过以下修改防止死机：

```
diff --git a/include/linux/mmc/host.h b/include/linux/mmc/host.h
@@ -355,6 +355,7 @@ static inline void mmc_signal_sdio_irq(struct mmc_host *host)
{
    host->ops->enable_sdio_irq(host, 0);

    host->sdio_irq_pending = true;

+    if(host && host->sdio_irq_thread)

    wake_up_process(host->sdio_irq_thread);
}
```

2 BT 调试思路

2.1 BT 打不开

先通过 `logcat -s [BT]` 打印 BT 初始化的 log，正常打印如下：

```
D/[BT]    ( 1417): BT_InitDevice: BT_InitDevice
D/[BT]    ( 1417): BT_Load_Firmware_Patch: BT load firmware patch
W/[BT]    ( 1417): BT_Load_Firmware_Patch: Open
/system/etc/firmware/MTK_MT6622_E2_Patch.nb0
D/[BT]    ( 1417): BT_Load_Firmware_Patch: Patch size 7976
D/[BT]    ( 1417): GORM_FW_Init_Thread: FW init thread starts
D/[BT]    ( 1417): GORMcmd_HCC_Read_Local_Version: GORMcmd_HCC_Read_Local_Version
```

如果打印到这里没有后面这条回应，那么是 BT 模块没有回应主控的第一条命令，请看下面[第](#)

2 节排查

```
D/[BT] ( 1417): GORMevt_HCE_Common_Complete: GORMevt_HCE_Common_Complete
D/[BT] ( 1417): GORM_FW_Init_Thread: The event of command 0 success
D/[BT] ( 1417): GORMcmd_HCC_Simulate_MT6612: GORMcmd_HCC_Simulate_MT6612
D/[BT] ( 1417): GORMevt_HCE_Common_Complete: GORMevt_HCE_Common_Complete
D/[BT] ( 1417): GORM_FW_Init_Thread: The event of command 1 success
D/[BT] ( 1417): GORMcmd_HCC_Fix_UART_Escape: GORMcmd_HCC_Fix_UART_Escape
D/[BT] ( 1417): GORMevt_HCE_Common_Complete: GORMevt_HCE_Common_Complete
D/[BT] ( 1417): GORM_FW_Init_Thread: The event of command 2 success
D/[BT] ( 1417): GORMcmd_HCC_GetHwVersion: GORMcmd_HCC_GetHwVersion
D/[BT] ( 1417): GORMevt_HCE_Common_Complete: GORMevt_HCE_Common_Complete
D/[BT] ( 1417): GORM_FW_Init_Thread: The event of command 3 success
D/[BT] ( 1417): GORMcmd_HCC_GetGormVersion: GORMcmd_HCC_GetGormVersion
D/[BT] ( 1417): GORMevt_HCE_Common_Complete: GORMevt_HCE_Common_Complete
D/[BT] ( 1417): GORM_FW_Init_Thread: The event of command 4 success
D/[BT] ( 1417): GORMcmd_HCC_ChangeBaudRate: Host baud 1500000 HCI_VS_UART_CONFIG
77 fc 06 00 40 60 e3 16 00
D/[BT] ( 1417): GORMevt_HCE_ChangeBaudRate: 0xfc77 Status Event
D/[BT] ( 1417): setup_uart_param: 1500000 1
D/[BT] ( 1417): set_speed: standard baudrate: 1500000 -> 0x0000100a
```

如果打印到这里没有后面这条回应，那么是 UART 速率从 115200 切换到 1500000 时异常，请

看下面第 3 节排查

```
D/[BT] ( 1417): GORMevt_HCE_ChangeBaudRate: 0xfcc0 Complete Event
D/[BT] ( 1417): GORMevt_HCE_ChangeBaudRate: 0xfc77 Complete Event
```

有以下几种异常情况：

1. 如果什么都没打印，那么是 android 上没有配置对或没有编译生效，参考以下文档配置，
make clean 再 make 生效。

Android 4.2 请参考《RK Android 4.2 WiFi BT 配置及常见问题说明 v2.0.pdf》配置；

Android 4.4 请参考《RKXX_Android4.4 WiFi BT 配置说明 V1.3.pdf》配置。

2. BT 模块没有回应主控的第一条串口命令：

A) 很大可能是 BT 模块没有工作起来，以 M500 15x15 邮票孔封装模块为例，需要确认以下部分是否正常：

- 1) 第 3 脚 VBAT，第 5 脚 VIO 供电是否正常

- 2) 第 35 脚 EXT_32K 是否有供 32.768K 的 clock，峰峰值要求为 $0.7 * vddio$ 到 $vddio$
- 3) 第 22 脚 SYSRST_B 需要高电平，在开机时软件会去拉高这个 GPIO 脚
- 4) 第 1 脚 BT_PWR_EN 需要高电平，在打开 BT 时软件会去拉高这个 GPIO 脚

（测试时需要注意：在打开 BT 时会拉高这个脚，可能因为打开 BT 异常，导致上层去关闭 BT，此时这个脚又会被拉低）

- 5) 确认 kernel 中是否有以下打印：

mt6622_probe

如果没有，那就是板级文件中 mt6622_platdata device 没有定义，这是由于升级到 4.4 平台后，控制 MT6622 的宏由原来的 CONFIG_MT5931_MT6622 变为 CONFIG_MTK_MT6622，有可能有些板级文件没有修改到，需要按如下示例板级修改：所有定义 CONFIG_MT5931_MT6622 的地方都加上 `|| defined(CONFIG_MTK_MT6622)`

```
+++ b/arch/arm/mach-rk3188/board-rk3188-sdk.c
@@ -1339,7 +1339,7 @@ struct platform_device rk_device_gps = {
};
#endif

-#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
    .io = RK30_PIN3_PD5, // set io to INVALID_GPIO for disable it
@@ -1420,7 +1420,7 @@ static struct platform_device *devices[] __initdata = {
    #if defined(CONFIG_ARCH_RK3188)
        &device_mali,
    #endif
    #ifndef CONFIG_MT5931_MT6622
    #if defined(CONFIG_MT5931_MT6622) || defined(CONFIG_MTK_MT6622)
        &device_mt6622,
    #endif
};
@@ -2276,7 +2276,7 @@ static void __init machine_rk30_board_init(void)
    clk_set_rate(clk_get_sys("rk_serial.0", "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

}
```

6) MT6622 不使用 RFKILL，使用私用的方法，其电源控制脚定义在板级文件中：

mt6622_platdata，具体引脚如下：

```
static struct mt6622_platform_data mt6622_platdata = {

// 蓝牙的电源控制

    .power_gpio    = { // BT_REG_ON

        .io        = RK30_PIN3_PC7, // set io to INVALID_GPIO for disable it

        .enable    = GPIO_HIGH,

        .iomux     = {

            .name    = "mt6622_power",

            .fgpio   = GPIO3_C7,

        },

    },

// 蓝牙的复位控制

    .reset_gpio    = { // BT_RST

        .io        = RK30_PIN3_PD1,

        .enable    = GPIO_HIGH,

        .iomux     = {

            .name    = NULL,

        },

    },

// 蓝牙通过该 IO 唤醒 HOST

    .irq_gpio      = {

        .io        = RK30_PIN0_PA5,
```

```
.enable = GPIO_LOW,

.iomux = {

.name = NULL,

},

}

};
```

B) 先确认是否已经按[第3节](#)排查，如果仍然存在问题，需要测试 UART 信号，具体：

用示波器测量主控 UART0_TX 是否有发出命令？

通过 `logcat -s [BT]` 打印，打印如下语句时，主控会发出命令

```
D/[BT] ( 1417): GORMcmd_HCC_Read_Local_Version: GORMcmd_HCC_Read_Local_Version
```

a) 如果没有发出，确认下 BT 模块是否有将 UART0_CTS 拉低，只有为低时，主控才会发出 TX 信号

b) 如果有发出，测量 UART0_RX 是否有收到 BT 模块的回应

c) 如果没有回应，确认下主控是否有将 UART0_RTS 拉低，只有为低时，BT 模块才会回应

C) 如果排查了以上原因，还会出现 BT 概率性打不开：

可以打上以下补丁，此补丁的作用是控制下 MT6622 的上电时序，具体为在打开 BT 前将 UART_RTS 切换成 GPIO 脚并拉底，在对模块上电完成后再切换回 UART_RTS 状态脚。

具体修改：在 board 板级文件中修改：

RK3066 平台修改如下：

```
+++ b/arch/arm/mach-rk30/board-rk30-sdk.c
@@ -1487,7 +1487,17 @@ static struct mt6622_platform_data mt6622_platdata = {
    .iomux = {
    .name = NULL,
    },
-   }
+   },
+   .rts_gpio = { // UART_RTS, enable or disable BT's data coming
+   .io = RK30_PIN1_PA3, // set io to INVALID_GPIO for disable it
```

```
+ .enable = GPIO_LOW,  
+ .iomux = {  
+ .name = GPIO1A3_UART0RTSN_NAME,  
+ .fgpio = GPIO1A_GPIO1A3,  
+ .fmux = GPIO1A_UART0_RTS_N,  
+ },  
+ },  
};
```

RK3188 平台修改如下:

```
+++ b/arch/arm/mach-rk3188/board-rk3188-ds1006h.c  
@@ -1236,7 +1236,17 @@ static struct mt6622_platform_data mt6622_platdata = {  
+ .iomux = {  
+ .name = NULL,  
+ },  
- }  
+ },  
+  
+ .rts_gpio = { // UART_RTS, enable or disable BT's data coming  
+ .io = RK30_PIN1_PA3, // set io to INVALID_GPIO for disable it  
+ .enable = GPIO_LOW,  
+ .iomux = {  
+ .name = "bt_rts",  
+ .fgpio = GPIO1_A3,  
+ .fmux = UART0_RTSN,  
+ },  
+ }  
};
```

修改后可通过以下打印来确认修改是否生效:

```
mt_bt_power_on rts iomux
```

3. 切换 UART 波特率后异常:

A) 按 UART 硬件流控或软件流控进行配置修改:

先确认硬件是否有连接 UART0_CTS, UART0_RTS, 也就是是否有连接 UART 流控脚。有连接流控脚, 也就是支持**硬件流控**, 没有连接流控脚, 只能支持**软件流控**。

1) UART 硬件流控配置

CONFIG_UART0_CTS_RTS_RK29=y

CONFIG_UART0_DMA_RK29=2

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) 将 MT6622 模块端的 RTS 悬空，主控端的 UART0_CTS 接地。

2) UART 软件流控配置

CONFIG_UART0_CTS_RTS_RK29=n

CONFIG_UART0_DMA_RK29=2

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

注意：根据配置成 UART 硬件流控还是软件流控，Android 上需要作相应的修改（修改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）：

Android 4.2 平台修改：

```
--- a/device/common/libbt_mtk6622/custom/cust_bt.h

/* 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 */
```

Android 4.4 平台修改：

```
--- a/hardware/mediatek/bt/mt5931_6622/custom/cust_bt.h

/* 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 */
```

Android 上的修改是否生效可通过 `logcat -s [BT]` 打印查看：

```
D/[BT] (1417): setup_uart_param: 1500000 1
```

“1500000”表示实际配置的 UART 波特率是 1.5M，蓝色“1”表示使用硬件流控，如果是“0”表示使用软件流控

B) UART 控制器 clock 设置是否合适：

如果 UART clock 配置不准确，会导致分不出精确的 1500000 的波特率，引起 UART 通信异常。

在 `arch/arm/mach-rkxx/board-xxx-sdk.c`，中的 `__init machine_rk30_board_init` 函数中添加

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

将 UART 控制器的 clock 设置成 24M，可通过 `cat /proc/clocks | busybox grep uart0` 打印 clock tree 来查看修改是否生效

2.2 BT 工作不稳定

出现概率性无法扫描或者无法传送文件，或者传送文件失败；

出现无法接收文件，或者概率性无法接收文件，接收文件中断；

1. 尝试关掉 BT 的 Sleep 功能

Android 4.2 平台修改：

```
--- a/device/common/libbt_mtk6622/custom/cust_bt.h

/* UART Flow Control */

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

/* Low Power Mode */

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

/* BT Addr Generate */
```

Android 4.4 平台修改：

```
--- a/hardware/mediatek/bt/mt5931_6622/custom/cust_bt.h

/* UART Flow Control */
```



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

```
/* Low Power Mode */
```

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

```
/* BT Addr Generate */
```

注意：修改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。

可通过以下打印来确认修改是否生效：logcat -s [BT]

```
D/[BT] (1417): mtk_fw_cfg:
```

```
[BD address de-a5-46-66-22-5c][capId 40][link key type 01]
```

```
[unit key 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00]
```

```
[encryption 00 02 10][pin code type 00]
```

```
[voice 60 00][codec 23 00 00 00]
```

```
[radio 07 80 00 06 03 07][sleep 00 40 1f 00 00 00 04]
```

```
[feature 80 00][ECLK_SEL 00]
```

2. 如果第 1 节的方法有效（不能按这个方法修改，这会造成 BT 唤醒功能异常），说明 BT sleep 功能存在异常，需要排查：

- 1) BT 模块 EXT_32K 脚是否有供 32.768K 的 clock，峰峰值要求为 $0.7 * vddio$ 到 $vddio$
- 2) BT 模块的 BGF_INT_B 中断脚是否有配置，配置正确，[参考 MT6622 板级配置](#)
- 3) 可能 BT 中断脚有异常

可通过 cat proc/interrupts 查看 BT 的中断情况

```
282:          2          0          0          0      GPIO  BT_INT_B
```

- a). 如果查看到中断数为 0，需要先查一下是否产生硬件中断：
- b). 软件里设置的默认中断有效电平是高电平，有可能跟硬件不匹配，将其设置为低电平试试
- c) BT_INT_B 所使用的 GPIO 是否可能被其它模块误使用了，导致无法产生中断

3. 如果是 RK3066 平台，出现接收文件或发送文件中断，需要修改如下：

```
--- a/kernel/arch/arm/mach-rk30/include/mach/board.h
```

```
+++ b/kernel/arch/arm/mach-rk30/include/mach/board.h
```

```
@@ -75,7 +75,7 @@ enum _codec_pll {  
  
#else  
  
#define RK30_CLOCKS_DEFAULT_FLAGS (CLK_FLG_MAX_I2S_12288KHZ|CLK_FLG_EXT_27MHZ*)  
  
#define RK30_CLOCKS_DEFAULT_FLAGS (CLK_FLG_MAX_I2S_12288KHZ|CLK_CPU_HPClk_11|CLK_FLG_EXT_27MHZ*)  
  
#if (RK30_CLOCKS_DEFAULT_FLAGS&CLK_FLG_UART_1_3M)
```

3 MT5931 WiFi 常见问题

3.1 Android 4.2 WFD 花屏问题&WFD 连接问题优化补丁

出现 WFD 很难连接上，或者连接上后显示花屏严重，请更新：

[illegible]

连接问题优化补丁.rar

3.2 Android 4.2 解决 iPhone 热点连接上后无法上网问题

出现无法连接上 iPhone 手机的热点，请更新：

[illegible]

接上后无法上网问题.rar

3.3 二级休眠时 MT5931 功耗异常问题

打开 WiFi 并连接上路由器，进入二级休眠，测试整机功耗发送波动比较大，这是 MT5931 模块引起的，目前没有根本解决方法。建议按如下处理：

将以下从原来默认的“总是”修改成“永不”，也就是机器在进入二级休眠 15 分钟左右后，wifi 连接会被断开，这样子 MT5931 功耗就正常了。

睡眠期间保持 WiFi 开启

总是

(默认, 机器休眠后 WiFi 不休眠)

仅限接入时

永不 (机器休眠后 WiFi 过 15 分钟会休眠)

如果觉得 15 分中太长，可调整：

frameworks/base/services/java/com/android/server/WifiService.java 中的 DEFAULT_IDLE_MS 时间

3.4 Android 4.4 BOX 平台出现 WiFi 打不开

问题：

BOX 平台出现 WiFi 打不开

平台：

RK3188 Android 4.4

现象：

打不开 WiFi。

分析：

从 log 来看 WiFi 驱动有加载，但是没有加载成功，具体查看：没有打印出：

```
<4>[ 95.908194] p2p Launch
```

```
<4>[ 95.908287] glRegisterP2P
```

进一步查看，发现没有打印：

```
[MTK-WIFI] WIFI_write: WIFI_write P2P
```

这是上层通过接口 fwReloadSoftap->wifi_change_fw_path，最后往"/dev/wmtWifi"里写 P2P 调用到底层的。

最后分析发现：BOX 平台的代码中以下语句被注释掉了，导致异常：

```
system/netd$ git diff
```

```
diff --git a/CommandListener.cpp b/CommandListener.cpp
```

```
index 438d6a6..80e7099 100755
```

```
--- a/CommandListener.cpp
```

```
+++ b/CommandListener.cpp

@@ -931,10 +931,11 @@ int CommandListener::SoftapCmd::runCommand(SocketClient *cli,

    #ifdef WIFI_CHIP_TYPE_ESP8089

        rc = 0;

    #else

-        //rc = sSoftapCtrl->fwReloadSoftap(argc, argv);
+        rc = sSoftapCtrl->fwReloadSoftap(argc, argv);

    #endif

    } else if (!strcmp(argv[1], "status")) {
```

3.5 Android BOX 平台出现 BT 传送大文件异常

问题:

BOX 平台出现 BT 传送大文件异常 (Ctrl H/w error event - code:0x30)

平台:

RK3188 Android 4.4

现象:

传送大文件出现中断, 然后 BT 被复位, 也就是关闭再重新打开。Log 有发现如下错误:

```
E/bt-hci ( 5064): Ctrl H/w error event - code:0x30
```

```
E/bt-btif ( 5064): Received H/W Error.
```

传送一个 100M 大小的文件, 每次都出错, 错误结果每次都一样。

分析:

分析代码发现具体错误是: HCI_HARDWARE_ERROR_EVT, 也就是 HCI 上报硬件错误。

具体原因不清楚, MTK 说是跟 32.768K clock 异常有关系, 但是没有发现异常。

解决方法:

忽略这个错误, 传送文件就可以成功继续。具体修改如下:

```
external/bluetooth/bluedroid$ git diff

diff --git a/stack/btu/btu_hcif.c b/stack/btu/btu_hcif.c
index 69b3882..f2f1db3 100644
--- a/stack/btu/btu_hcif.c
+++ b/stack/btu/btu_hcif.c

@@ -298,7 +298,7 @@ void btu_hcif_process_event (UINT8 controller_id, BT_HDR *p_msg)

    btu_hcif_command_status_evt (controller_id, p, hci_evt_len);

    break;

    case HCI_HARDWARE_ERROR_EVT:

-        btu_hcif_hardware_error_evt (p, hci_evt_len);
+        //btu_hcif_hardware_error_evt (p, hci_evt_len);

    break;

    case HCI_FLUSH_OCCURED_EVT:

        btu_hcif_flush_occured_evt (p, hci_evt_len);
```

3.6 Android 4.4 Box 平台 WiFi 扫描到异常名字 AP

问题:

BOX 平台 WiFi 扫描到异常名字 AP” NVRAM WARNING:Err=0x10”

平台:

RK3188 BOX Android 4.4

解决方法:

```
/drivers/net/wireless/mt5931_kk/drv_wlan/common/wlan_lib.c

@@ -5520,7 +5520,7 @@ wlanCheckSystemConfiguration (

-#if (CFG_NVRAM_EXISTENCE_CHECK == 1)
+#if 0/(CFG_NVRAM_EXISTENCE_CHECK == 1)

#define NVRAM_ERR_MSG "NVRAM WARNING: Err = 0x01"
```

```
@@ -5549,7 +5549,7 @@ wlanCheckSystemConfiguration (
-#if (CFG_SW_NVRAM_VERSION_CHECK == 1)
+#if 0// (CFG_SW_NVRAM_VERSION_CHECK == 1)

#define VER_ERR_MSG "NVRAM WARNING: Err = 0x%02X"
```

3.7 Android 4.4 WiFi 休眠后唤醒异常

问题:

Android 4.4 平台上，WiFi 较长时间休眠后再唤醒后异常，wifi 无法使用，必须重新开关 WiFi,通过 logcat 可发现以下错误：

```
09-11 03:34:25.080 I/wpa_supplicant( 1194): nl80211: not associated, no need to deauthenticate
09-11 03:34:25.080 E/wpa_supplicant( 1194): Could not read interface flags: Bad file number
09-11 03:34:25.080 I/wpa_supplicant( 1194): nl80211: Could not set interface Down, ret=-9
```

平台：

RK3188 Android 4.4

解决方法:

打上以下补丁：

<ftp://www.rockchip.com.cn/>

27-WiFi 相关补丁/MT5931&MT6622/mtk_wifi_nl80211_struct_dismatch.tar.gz

4 MT6622 BT 常见问题

4.1 Android 4.4 bluebird 补丁

MTK 提交了一些解决 Android 4.4 原生 android bt 问题的补丁，具体补丁如下，可作为参考。

[ftp://www.rockchip.com.cn/27-WiFi 相关补丁/MT5931&MT6622/](ftp://www.rockchip.com.cn/27-WiFi%20%E7%9C%8B%E6%8E%A5%E7%BB%B4/MT5931&MT6622/)

Android4.4 BT patch(must read).rar


```
BOARD_CONNECTIVITY_VENDOR := Espressif
```

```
BOARD_CONNECTIVITY_MODULE := mt6622
```

修改后，make clean 再编译。

5 补丁下载地址

补丁可从以下 FTP 地址下载：

FTP 地址： <ftp://www.rockchip.com.cn>

账号名： rkwifi

密码： Cng9280H8t

目录： 27-WiFi 相关补丁