8 密级状态：绝密（ ） 秘密（ ） 内部（ ） 公开（ √ ）

ATV_Android_8.0_WiFi_BT_移植和问题汇总

（技术部，系统产品一部）

文件状态： [√] 正在修改 [ ] 正式发布	当前版本：	V1.0
	作　　者：	许学辉
	完成日期：	2017-12-20
	审　　核：	
	完成日期：	

福州瑞芯微电子有限公司

Fuzhou Rockchips Semiconductor Co . , Ltd

版 本 历 史

版本号	作者	修改日期	修改说明	备注
V1.0	许学辉	2017-12-20	初稿	

目 录

1 说明

明确 ATV android .1 平台上 wifi、bt 内核相关部分修改，自测遇到的 BUG，方便以后 android 版本移植

1.1 WIFI 移植内核部分

wifi 完全自动兼容方案 AP6xxx 系列 wifi 和 Realtek 系列 wifi 驱动都已经全部采用 module 方式，该部分兼容代码跟以前 Android 7.1 实现原理完全一样。

最新内核代码,：

CONFIG_WL_ROCKCHIP=y

CONFIG_WIFI_BUILD_MODULE=y

CONFIG_WIFI_LOAD_DRIVER_WHEN_KERNEL_BOOTUP=y

CONFIG_AP6XXX=m

CONFIG_RTL_WIRELESS_SOLUTION=y

CONFIG_RTL8188EU=m

CONFIG_RTL8188FU=m

CONFIG_RTL8189ES=m

CONFIG_RTL8189FS=m

CONFIG_RTL8723BS=m

CONFIG_RTL8723BU=m

CONFIG_RTL8723CS=m

CONFIG_RTL8723DS=m

CONFIG_RTL8822BE=m

Device Drivers --->

[*] Network device support --->

[*] Wireless LAN --->

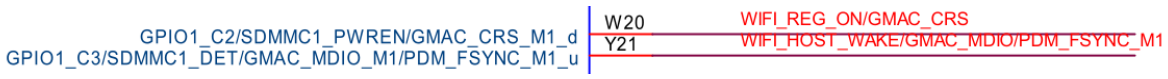
```
-*- Wireless LAN
[ ]   Wifi load driver when kernel bootup
[*]   Realtek Wireless Device Driver Support  --->
[*]   Espressif 8089 sdio Wi-Fi
<M>   RK901/RK903/BCM4330/AP6XXX wireless cards support
```

```
--- Realtek Wireless Device Driver Support
<M>   Realtek 8188E USB WiFi
<M>   Realtek 8189ES/ETV SDIO WiFi Support
< >   Realtek 8192CU USB WiFi Support
< >   Realtek 8192DU USB WiFi Support
< >   Realtek 8723AU USB WiFi Support
< >   Realtek 8723BU USB WiFi
<M>   Realtek 8723B SDIO or SPI WiFi
< >   Realtek 8723BS_VQ0 WiFi
< >   Realtek 8723C SDIO or SPI WiFi
< >   Realtek 8723D SDIO or SPI WiFi
< >   Realtek 8812AU USB WiFi Support
< >   Realtek 8189F SDIO WiFi
<M>   Realtek 8188F USB WiFi
<M>   Realtek 8822BS SDIO WiFi
<M>   Realtek 8822B USB WiFi
```

板级 dts 无需配置 WIFI 芯片类型（配置了也可以），因为加载 wifi 驱动不依赖 wifi_chip_type 节点，如果 WIFI 没有根据 RK 发布的硬件参考设计，板级 dts 先确认如下信息：

以 RK3328 （kernel 4.4）为例：

arch/arm64/boot/dts/rockchip/rk3328-evb-android.dts dts wifi 部分只需注意如下红色部分标记即可（配置好 WIFI_REG _ON 和 OOB 管脚）




```
sdio_pwrseq: sdio-pwrseq {
        compatible = "mmc-pwrseq-simple";
        pinctrl-names = "default";
        pinctrl-0 = <&wifi_enable_h>;

        /*
         * On the module itself this is one of these (depending
         * on the actual card populated):
         * - SDIO_RESET_L_WL_REG_ON
         * - PDN (power down when low)
         */
        reset-gpios = <&gpio1 18 GPIO_ACTIVE_LOW>;
};
```

```
wireless-wlan {
        compatible = "wlan-platdata";
        rockchip,grf = <&grf>;
        wifi_chip_type = "ap6354";
        sdio_vref = <1800>;
        WIFI,host_wake_irq = <&gpio1 19 GPIO_ACTIVE_HIGH>;
        status = "okay";
};
```

```
&pinctrl {
        pmic {
                pmic_int_l: pmic-int-l {
                rockchip,pins =
                        <2 RK_PA6 RK_FUNC_GPIO &pcfg_pull_up>;  /* gpio2_a6 */
                };
        };

        sdio-pwrseq {
                wifi_enable_h: wifi-enable-h {
                rockchip,pins =
                        <1 18 RK_FUNC_GPIO &pcfg_pull_none>;
                };
        };

        wireless-bluetooth {
                uart0_gpios: uart0-gpios {
                rockchip,pins =
                        <1 10 RK_FUNC_GPIO &pcfg_pull_none>;
                };
        };
};
```

1.2 蓝牙移植内核部分

目前发现 rk3328 平台，蓝牙 wake_host_irq 管教的状态不太正常，导致 UART 一次，蓝牙无法跑起来，这个脚为输出，状态为 lo,

root@rk3328:/ # cat sys/kernel/debug/gpio

GPIOs 32-63, platform/pinctrl.21, gpio1:

gpio-42 (ap6335_rts) out lo

gpio-50 (wlan_default_wlan_po) out lo

但是这个脚正常也是可以正常拉低拉高的。

拉低：io -4 0xff220004 0x00240400

拉高 io -4 0xff220004 0x04240400

问题原因：被主控拉低，因为接的一个管教是内部下拉的管脚，这个脚 ipput 时会拉 low，

考虑到其他平台也可能有相同的问题，那就不改 DTS，直接在 rkill 里面进行统一设定：

方法如下：

在拉高 BT power 之前，先吧这个脚设置为输出状态，并且为高电平，拉高 Power 后，再吧这个脚设置为输入状态，也就是还原为原本状态，内核提交号：ff766be22c901fd85992e4af80ed70e4f5ab45f2

```
commit ff766be22c901fd85992e4af80ed70e4f5ab45f2
Author: xxh <xxh@rock-chips.com>
Date:   Fri Nov 24 15:23:30 2017 +0800

    bluetooth: rfkill: rockchip: ap6356s bluetooth module for RK3328 EVB board

    Change-Id: I3f1e7e792b1c4d6952080b15e5bdb2902df4ddb1
    Signed-off-by: Xu Xuehui <xxh@rock-chips.com>
```

这样蓝牙在 kenrel 4.4 上就 UART 就正常通信了。

但是 adnroid 上层打开蓝牙，协议栈会崩溃，崩溃 LOG 如下：

01-01 00:02:02.073 F/DEBUG (4035): *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***

01-01 00:02:02.074 F/DEBUG (4035): Build fingerprint: 'Android/rk3328/rk3328:8.0.0/OPR5.170623.007/cw11222033:userdebug/test-keys'

01-01 00:02:02.074 F/DEBUG (4035): Revision: '0'

01-01 00:02:02.074 F/DEBUG (4035): ABI: 'arm64'

01-01 00:02:02.074 F/DEBUG (4035): pid: 3985, tid: 4014, name: stack_manager >>> com.android.bluetooth <<<

01-01 00:02:02.074 F/DEBUG (4035): signal 6 (SIGABRT), code -6 (SI_TKILL), fault addr --------

01-01 00:02:02.085 F/DEBUG (4035): Abort message: '[0101/000201:FATAL:alarm.cc(154)] Check failed: false.

01-01 00:02:02.085 F/DEBUG (4035): '

分析协议栈相关源码，确认协议栈需要开启 rtc，内核提交号：c19d52e5b7955883743611c828d11519643e49cc

```
Author: xxh <xxh@rock-chips.com>
Date:   Fri Nov 24 14:45:00 2017 +0800

    arm64: dts: rockchip: RK3328 EVB board for bluetooth

    Change-Id: Id54ef68dc939d3edf54d5cda1ee25e0014de2579
    Signed-off-by: Xu Xuehui <xxh@rock-chips.com>

diff --git a/arch/arm64/boot/dts/rockchip/rk3328-evb-android.dts b/arch/arm64/boot/dts/rockchip/rk3328-evb-android.dts
index 48165cc..c7b5e31 100644
--- a/arch/arm64/boot/dts/rockchip/rk3328-evb-android.dts
+++ b/arch/arm64/boot/dts/rockchip/rk3328-evb-android.dts
@@ -168,6 +168,19 @@
                vin-supply = <&vcc_io>;
        };

+       wireless-bluetooth {
+               compatible = "bluetooth-platdata";
+               clocks = <&rk805 1>;
+               clock-names = "ext_clock";
+               uart_rts_gpios = <&gpio1 10 GPIO_ACTIVE_LOW>;
+               pinctrl-names = "default", "rts_gpio";
+               pinctrl-0 = <&uart0_rts>;
+               pinctrl-1 = <&uart0_gpios>;
+               BT,power_gpio = <&gpio1 21 GPIO_ACTIVE_HIGH>;
+               BT,wake_host_irq = <&gpio1 26 GPIO_ACTIVE_HIGH>;
+               status = "okay";
+       };
+
        wireless-wlan {
                compatible = "wlan-platdata";
                rockchip,grf = <&grf>;
@@ -264,7 +277,7 @@
                clock-output-names = "xin32k", "rk805-clkout2";

                rtc {
-                       status = "disabled";
+                       status = "okay";
                };

                pwrkey {
@@ -424,6 +437,13 @@
                        <1 18 RK_FUNC_GPIO &pcfg_pull_none>;
                };
        };
+
+       wireless-bluetooth {
+               uart0_gpios: uart0-gpios {
+               rockchip,pins =
+                       <1 10 RK_FUNC_GPIO &pcfg_pull_none>;
+               };
+       };
 };

 &pwm3 {
@@ -533,6 +553,12 @@
        status = "okay";
```

2 Wifi BT 兼容 android 部分

2.1 wifi 芯片识别流程

1. 开机对 wifi 模块上电，并自动进行扫描 sdio 操作
2. 系统启动打开 wifi 操作时，分别对系统 sys/bus/sdio（sdio wifi）， sys/bus/usb(usb wifi)，sys/bus/pic (pcie wifi)文件系统下的 uevent 进行读取
3. 获取到 wifi 芯片 vid pid 加载相应的 wifi ko 驱动
4. 识别到相应的 wifi 模块后，即可知道相应的 bt 型号，走不同的 bluedroid 协议栈

Android 核心自动识别代码目录：frameworks/opt/net/wifi/libwifi_hal/ rk_wifi_ctrl.cpp,该部分 check_wifi_chip_type_string 的实现代码与 android 7.1 实现方式完全一样，原理如上描述。

8.0 上 rk_wifi_ctrl 动态库的编译有些许变化，也就是 8.0 和 7.1 的差别之处需要在 Android.bp 中添加编译规则，其他地方要用到 librkwifi-ctrl 直接引用即可

```
wifi_hal_cflags = [
    "-Wall",
    "-Werror",
    "-Wextra",
    "-Winit-self",
    "-Wno-unused-function",
    "-Wno-unused-parameter",
    "-Wshadow",
    "-Wunused-variable",
    "-Wwrite-strings",
]

cc_library_shared {
    name: "librkwifi-ctrl",
    vendor: true,
    cflags: wifi_hal_cflags,
    local_include_dirs: ["include"],
    shared_libs: ["libbase"],
    header_libs: ["libcutils_headers"],
    srcs: ["rk_wifi_ctrl.cpp"],
}
```

额外要注意的，android 8.0 上统统放到了 vendor/rockchip/common/wifi/firmware 这个目录进行拷贝到系统的 vendor 目录下，

2.1 wpa_supplicant 启动部分修改

android 8.0 上面，将原来的 wpa_suppliant 启动和 wifi 驱动加载剥离开了，以前是统一在 wifi.c 里面处理，现在分为了 libwifi_hal/wifi_hal_common.cpp 和 libwifi_system/supplicant_manager.cpp

1. wifi_hal_common.cpp 文件修改

frameworks/opt/net/wifi/ libwifi_hal/wifi_hal_common.cpp 该文件其实可以直接对应以前的 wifi.c 里面的核心部分是添加 check_wifi_chip_type_string, insmod 和 rmmod 驱动，方式跟以前的 wifi.c 一样。

2. android 8.0 启动 wpa_supplicat bin 的方式

android 8.0 上面不在 init.rc 传入 wifi wpa 启动参数，以前直接在 init.rc 传参，现在考虑到兼容和 VTS 测试()，所以考虑直接放到了原生态 external/wpa_supplicant_8/ wpa_supplicant/main.c 里面 main_loop 函数进行兼容，当然如果不做兼容，也完全可以在 init.rc 中做，就是可以要自行对 wpa 进行修改，考虑到这里，RK andriod 8.0 wp 兼容部分修改如下内容：

```
int main(int argc, char *argv[])
{
        int ret = -1;
        char module_type[20]={0};

        wpa_printf(MSG_INFO,"argc = %d\n",argc);
        if(argc == 2) {
                if (wifi_type[0] == 0) {
                        check_wifi_chip_type_string(wifi_type);
                }
                wpa_printf(MSG_INFO,"Current wifi chip is %s\n",wifi_type);
                if (0 == strncmp(wifi_type, "RTL", 3)) {
                        wpa_printf(MSG_INFO,"Start rtl_wpa_supplicant\n");
                        ret = read_wpa_param_config(REALTEK_MODULE_NAME,argv[1]);
                } else if (0 == strncmp(wifi_type, "AP", 2)) {
                        wpa_printf(MSG_INFO,"Start bcm_wpa_supplicant\n");
                        ret = read_wpa_param_config(BROADCOM_MODULE_NAME,argv[1]);
                } else if (0 == strncmp(wifi_type, "SSV", 3)) {
                        wpa_printf(MSG_INFO,"Start ssv_wpa_supplicant\n");
                        ret = read_wpa_param_config(SSV_MODULE_NAME,argv[1]);
                } else if (0 == strncmp(wifi_type, "ESP", 3)) {
                        wpa_printf(MSG_INFO,"Start esp_wpa_supplicant\n");
                        ret = read_wpa_param_config(ESP_MODULE_NAME,argv[1]);
                } else {
                        wpa_printf(MSG_INFO,"Start wpa_supplicant\n");
                        sprintf(module_type,"[%s]",wifi_type);
                        ret = read_wpa_param_config(module_type,argv[1]);
                }
        } else {
                wpa_printf(MSG_INFO,"Start wpa_supplicant\n");
                ret = main_loop(argc, argv);
        }
        return ret;
}
```

device/rockchip/common/ init.connectivity.rc 内容如下

service wpa_supplicant /vendor/bin/hw/wpa_supplicant \

/vendor/etc/wifi/wpa_config.txt

class main

socket wpa_wlan0 dgram 660 wifi wifi

disabled

oneshot

3. hardware/interfaces/wifi,该部分涉及 VTS，

WIFI-HAL: android.hardware.wifi@1.0-service 这个 servcie,也就是在以前 android 框架基础上又剥离了一层，以前 android 版本 wifi 相关命令都是 wpa_suppliant 直接通过 JNI 发送命令到驱动 frameworks/opt/net/wifi/service/jni/com_android_server_wifi_WifiNative.cpp，

现在与 wpa 通信的要经过如下的过程，如下罗列一下重要的更新内容：

① frameworks/opt/net/wifi/service/java/com/android/server/wifi/WifiNative.java （JNI operations）

② SupplicantStaIfaceHal 和 mWifiVendorHal（Vendor HAL via HIDL）

③ **external/wpa_supplicant_8**/wpa_supplicant/hidl/1.0/sta_iface.cpp

④ frameworks/opt/net/wifi/service/java/com/android/server/wifi/SupplicantStaIfaceHal.java

⑤ frameworks/opt/net/wifi/service/jni/com_android_server_wifi_WifiNative.cpp

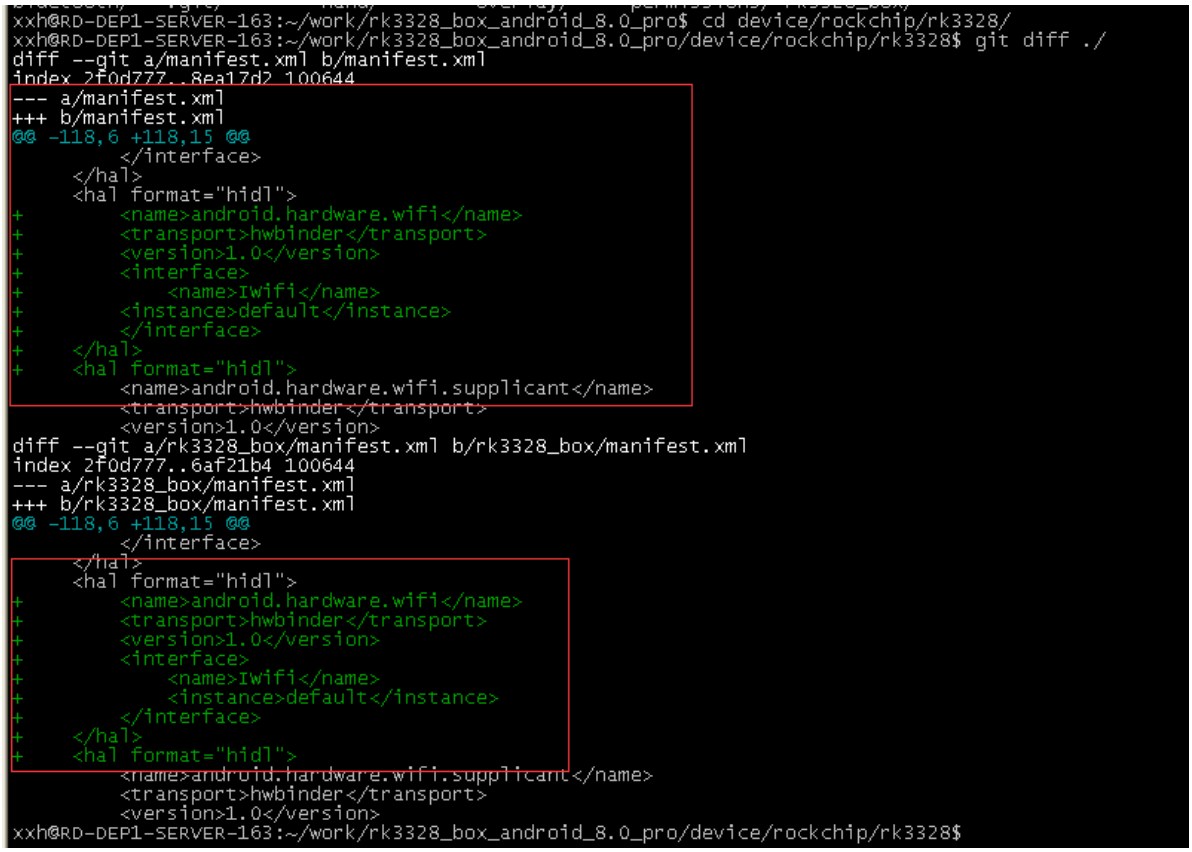
⑥ **hardware**/interfaces/wifi/supplicant/1.0/ISupplicantIface.hal （hardware 层 HAL）

⑦ **hardware**/interfaces/wifi/supplicant/1.0/ISupplicantStaIface.hal

可以看出，android 8.0 删除了原来 android 的 JNI 函数的注册,而大量使用 HIDL，HDIL 定义了很多对应 WIFI 的接口，直接调用这些接口进行 IPC 交互 ，跟 wpa_supplicant 方式也都统统改变，使得上层 Framework 往下变得解耦 ，使得上下层能独立更新，用户更新系统时不依赖于硬件的限制 ，所以也新增加了 WIFI VTS 的测试。

4. VtsHalWifiV1_0Target VTS 测试

标准 WIFI hal 需要在 device manifest 里面添加，格式如下，如下添加的标准 WIFI HAL

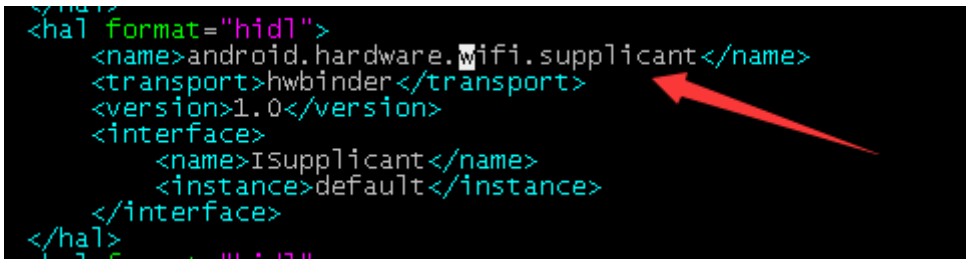


博通模块，采用 android.hardware.wifi@1.0-service+ VtsHalWifiV1_0Target 测试 VTS PASS

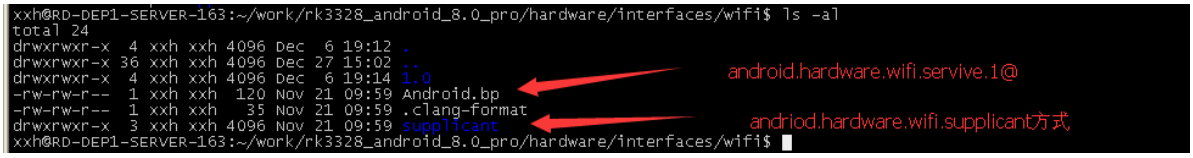
Realtek 模块，采用 android.hardware.wifi@1.0-service+ VtsHalWifiV1_0Target 测试 VTS FAIL

咨询过 realtek，目前他们的驱动对 VTS 这块还有缺陷 **(先弄个 beta 版本)**。

目前 SDK 屏蔽掉了 android.hardware.wifi@1.0-service，但是要添加驱动加载，否则 android 8.0 不会去加载 WIFI 驱动，实现方式直接加载驱动方式，由于 VTS 采用的两种方式，一种是 wpa_supplicant，一种是 wifi_hal，采用 wifi_hal VTS 都可以 PASS，**也就是说目前 SDK 屏蔽掉 hardware.wifi service** ,采用了 **android.hardware.wifi.supplicant**+VTS 测试方式，realtek 不测试 WIFI hal，具体让原厂介入修复 reltek 模块的问题，manifest.xml 内容如下

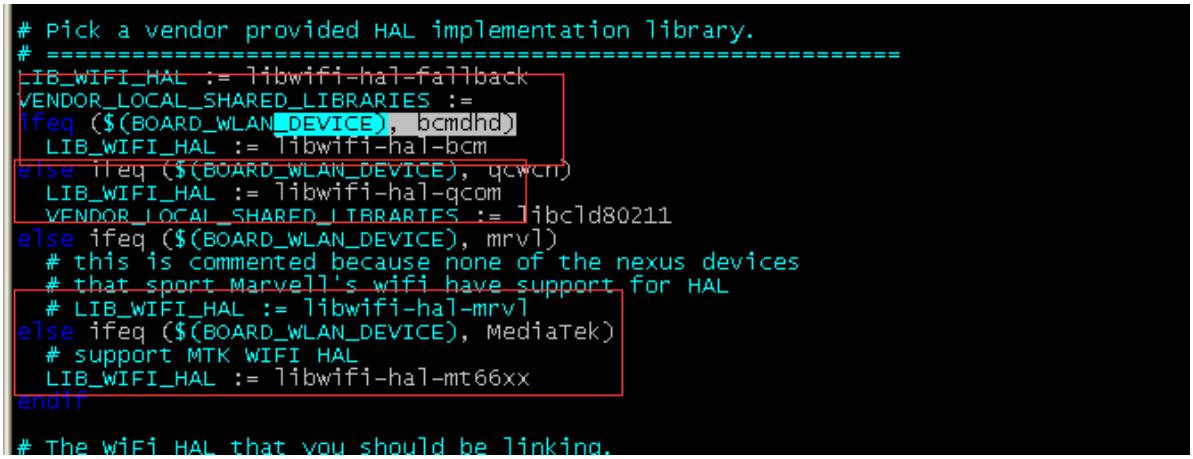


说明：反正这个方式还值得讨论，目前的做法只是考虑到 VTS 测试和功能性的问题，也就是剥离了原生态本来 libwifi_hal 和 android.hardware.wifi 的联系



遗留问题：

1. Wpa 私有库和 wpa_supplicant 的兼容



2. GMS 项目上 VTS 和 realtek 模块的兼容 （realtek 原厂处理中，进度缓慢）

3 Wifi BUG 汇总

1. Wifi 开关异常（解决 100%）

WIFI 驱动 free_irq 调用，free_irq 前，irq 已经 disable 了，这个没错，因为最后一次执行的是 disable_irq_nosync，由于 AP 系列 WIFI 都是使用的 OOB 中断，关闭 WIFI，需要 free_irq，以下警告，这个警告在 reboot 或者其他情况可能会导致其他系统问题

[11244.134024] [<ffffff80089174c4>] clk_core_disable+0x18/0x18c

[11244.134039] [<ffffff80089178f0>] clk_disable+0x2c/0x40

[11244.134064] [<ffffff80083ab5a4>] rockchip_irq_gc_mask_set_bit+0x20/0x2c

[11244.134086] [<ffffff80080f7788>] irq_shutdown+0x4c/0x68

[11244.134098] [<ffffff80080f556c>] __free_irq+0x100/0x20c

这个问题其实有两个改法：

1. 修改 WIFI 驱动，在 free_irq 之前 ，enable 一次，RK3328 上面这么修改，需要修改 OOB 管脚 pinctrl，否则会出现关闭 WIFI，直接中断打死 CPU。我在 gerrit 上也有临时版本

```
arm64: dts: rk3328: fix wifi oob interrupt infinitely increase

the reason is wifi oob pin internal pullup by rk3328

operation show as below:

1. close wifi or ifconfig wlan0 down
2. call enable_irq function
3. oob interrupt infinitely increase

Change-Id: I24b8c8afa592acb6be63599a7fa4cc74fff44e82
Signed-off-by: Xu Xuehui <xxh@rock-chips.com>
```

```
Author      Xu Xuehui <xxh@rock-chips.com>
Committer   Xu Xuehui <xxh@rock-chips.com>
Commit      413d3b7b6e3e7fb1ada1cce3807fd2bb48f7e186
Parent(s)   abb89382597546598b2d835314807b76b7e01fed
Change-Id   I24b8c8afa592acb6be63599a7fa4cc74fff44e82
```

2. 内核 irq 修复 BUG，如下是在 UPSTREAM 上查到的相关实现，一共 5 个提交，由涛哥审核后合入

e41d630 UPSTREAM: genirq/PM: Properly pretend disabled state when force resuming interrupts ff9c8bc UPSTREAM: genirq: Avoid unnecessary low level irq function calls 52e28f1 UPSTREAM: genirq: Set irq masked state when initializing irq_desc

4095964 UPSTREAM: genirq: Warn when IRQ_NOAUTOEN is used with shared interrupts 255ba95 UPSTREAM: genirq: Handle NOAUTOEN interrupt setup proper

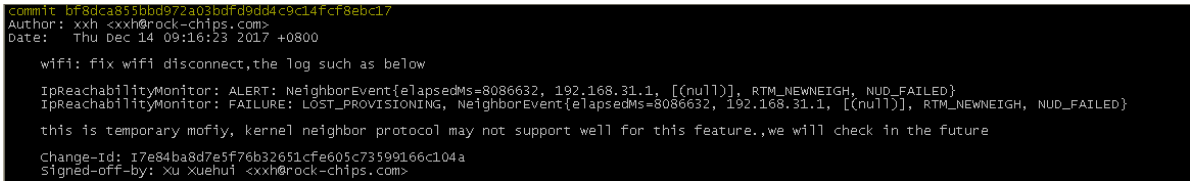
Revert "net: wireless: rockchip_wlan: ap6255: fix irq abnormal."

2. Wifi 掉线（解决 100%）

IpReachabilityMonitor: ALERT: NeighborEvent{elapsedMs=8086632, 192.168.31.1, [(null)], RTM_NEWNEIGH, NUD_FAILED}

IpReachabilityMonitor: FAILURE: LOST_PROVISIONING, NeighborEvent{elapsedMs=8086632, 192.168.31.1, [(null)], RTM_NEWNEIGH, NUD_FAILED}

frameworks/base/services 提交号：bf8dca855bbd972a03bdfd9dd4c9c14fcf8ebc17



备注：该处理方式跟以前 7.1 方式不一样，以前 7.1 我是有两个方案：

1. 直接屏蔽 IpReachabilityMonitor Service，也就是系统没有这个 serive
2. 在 wifi roming 的时候才触发 IpReachabilityMonitor,也就是保留这个 service，在 roming 触发

3. Wifi RX/TX 驱动异常（解决 100%）

dhd_dpc 跑到了 sdio 可休眠函数，直接死锁，导致 WIFI HAGUP

内核提交号：21ff1d170f97e52366d77b2bc7506483b48f85e3

[<ffffff8008b0f258>] schedule_timeout+0x3c/0x260

[<ffffff8008b0d48c>] wait_for_common+0x130/0x168

[<ffffff8008b0d4d8>] wait_for_completion+0x14/0x1c

[<ffffff800876947c>] mmc_wait_for_req+0x84/0x160

[<ffffff8008774518>] mmc_io_rw_extended+0x26c/0x2ec

[<ffffff8008775958>] sdio_io_rw_ext_helper+0x14c/0x1a0

[<ffffff8008775b50>] sdio_readsb+0x1c/0x24

[<ffffff8000b5d480>] sdioh_buffer_tofrom_bus+0xc0/0x238 [bcmdhd]

[<ffffff8000b5e910>] sdioh_request_buffer+0x1c0/0x360 [bcmdhd]

[<ffffff8000b5c1c8>] bcmsdh_recv_buf+0x88/0xc0 [bcmdhd]

[<ffffff8000b6412c>] dhd_bcmsdh_recv_buf.constprop.25+0x64/0x98 [bcmdhd]

[<ffffff8000b66fb8>] dhdsdio_readframes+0x908/0x14e0 [bcmdhd]

[<ffffff8000b6bc20>] dhd_bus_dpc+0x6c0/0xb80 [bcmdhd]

[<ffffff8000b35cd8>] dhd_dpc_thread+0x128/0x1a0 [bcmdhd

4. ATV Wifi 连接 AP 慢 (按照 box android 7.1 的方式处理)

packages/apps/TvSettings 目录如下提交

commit 3493d405dd8eaee2ac1de88fee524b8fc056f7de

Author: xxh <xxh@rock-chips.com>

Date: Fri Jan 5 08:50:38 2018 +0800

ConnectToWifiFragment: solve wifi connect ap so slow

5. Kernel 4.4 wifi ko rmmod/insmod 问题（ 处理中）

目前发现 kenrel 4.4 SDIO 框架设置为了 no-removable，导致卸载后，无法正常加载 驱动

暂时处理办法，对于 32K 的调整，再评估

```
diff --git a/arch/arm64/boot/dts/rockchip/rk3328-evb-android.dts
b/arch/arm64/boot/dts/rockchip/rk3328-evb-android.dts
index 6bda0fd..0c63896 100644
--- a/arch/arm64/boot/dts/rockchip/rk3328-evb-android.dts
+++ b/arch/arm64/boot/dts/rockchip/rk3328-evb-android.dts
@@ -112,6 +112,7 @@
                        rockchip,grf = <&grf>;
                        wifi_chip_type = "ap6354";
                        sdio_vref = <1800>;
+                       WIFI,poweren_gpio = <&gpio1 18 GPIO_ACTIVE_HIGH>;
                        WIFI,host_wake_irq = <&gpio1 19 GPIO_ACTIVE_HIGH>;
                        status = "okay";
                };
@@ -500,8 +501,6 @@
                disable-wp;
                keep-power-in-suspend;
                max-frequency = <150000000>;
-               mmc-pwrseq = <&sdio_pwrseq>;
-               non-removable;
                num-slots = <1>;
                pinctrl-names = "default";
                pinctrl-0 = <&sdmmc1_bus4 &sdmmc1_cmd &sdmmc1_clk>;
```

6. Ap6356s 驱动概率出现扫描超时，设置界面无响应 问题（ 处理中）

```
[   38.234365] Unable to handle kernel NULL pointer dereference at virtual
address 00000028
...
[   38.234420] task: dc83bf00 task.stack: dc844000
[   38.234438] PC is at wl_scan_timeout+0x11c/0x1dc
[   38.234451] LR is at _raw_spin_unlock_irqrestore+0x1c/0x24
[   38.234458] pc : [<c0675ddc>]    lr : [<c0be39cc>]    psr: 600c0113
[   38.234458] sp : dc845e10  ip : dc845d28  fp : dc845e74
[   38.234462] r10: c0e4385a  r9 : 00000020  r8 : deec14a0
[   38.234467] r7 : deec04a0  r6 : c126e864  r5 : deec16e4  r4 : deec46e8
[   38.234472] r3 : 00000000  r2 : 00000020  r1 : dc83bf00  r0 : 0000001a
```

测试命令：

while true; do echo "scan"; wpa_cli scan; sleep 2; wpa_cli scan_result; done

自测在一台样机上出现，怀疑 WIFI 驱动概率出现空指针应用，采用如下改动暂时又没发现异常，两台 SDK 不添加改动同样正常，为复现到问题，怀疑硬件可能有点问题

diff --git a/drivers/net/wireless/rockchip_wlan/rkwifi/bcmdhd/wl_cfg80211.c

b/drivers/net/wireless/rockchip_wlan/rkwifi/bcmdhd/wl_cfg80211.c

index e246088..4f0f5db 100644

--- a/drivers/net/wireless/rockchip_wlan/rkwifi/bcmdhd/wl_cfg80211.c

+++ b/drivers/net/wireless/rockchip_wlan/rkwifi/bcmdhd/wl_cfg80211.c

@@ -14082,7 +14082,7 @@

struct bcm_cfg80211 *cfg = (struct bcm_cfg80211 *)data;

struct wireless_dev *wdev = NULL;

struct net_device *ndev = NULL;

- **struct wl_scan_results *bss_list;**

+ **struct wl_scan_results *bss_list = NULL;**

struct wl_bss_info *bi = NULL;

s32 i;

u32 channel;

7. Ap6356s Softap 功能异常（ 处理中）

dhd_os_open_image: /vendor/etc/firmware/fw_bcm4356a2_ag.bin (558327 bytes) open

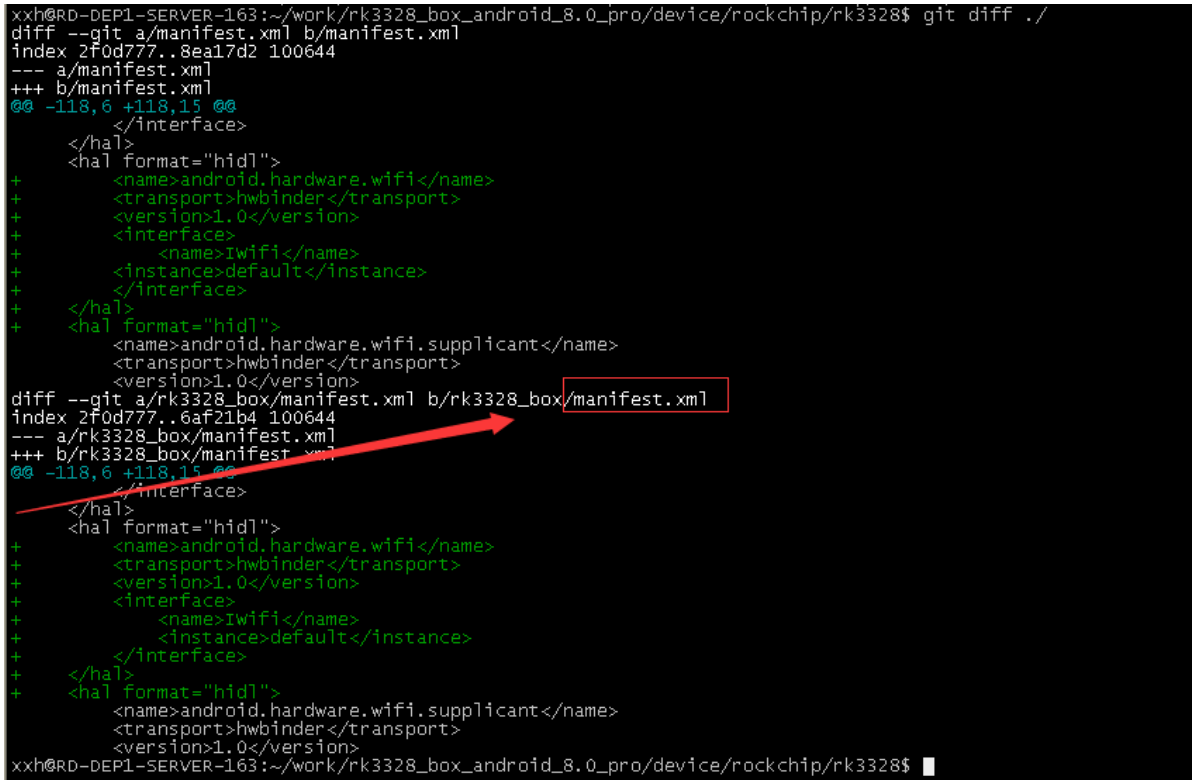
success

dhd_os_open_image: /vendor/etc/firmware/nvram_ap6356.txt (2840 bytes) open success

原因：

1. 没有跑 libwifi-hal.so 流程，也就是没有走 change_fw，这个是由于引入了 rk wifi hal 导致，目前 reltek 由于驱动原因，不支持 wifi hal，
2. **下载正常的 FW 后，同样无法连接，更换 WIFI FW 由可以正常连接 （没找到原因）**

   解决办法：



4 BT BUG 汇总

1. ATV BLE 遥控器设备连接不上（解决 100%）

需要更新蓝牙 FW，并且放开 BLE_VND_INCLUDED,否则 BLE 遥控器连接不上

```
    Change-Id: Ia0ac12c42e67740cb8d5709e6bb4a25e51600030
    Signed-off-by: Xu Xuehui <xxh@rock-chips.com>

diff --git a/bluetooth/bdroid_buildcfg.h b/bluetooth/bdroid_buildcfg.h
index 9cb8378..4c74241 100755
--- a/bluetooth/bdroid_buildcfg.h
+++ b/bluetooth/bdroid_buildcfg.h
@@ -20,5 +20,7 @@
 #define BTM_DEF_LOCAL_NAME     "rk3328"
 #define BTA_DM_COD {0x1A, 0x01, 0x10}

+#define BLE_PRIVACY_SPT FALSE
+#define BLE_VND_INCLUDED TRUE
 #endif
```

2. ATV HID 蓝牙鼠标设备配对失败 （解决 100%）

packages/apps/TvSettings 目录如下提交解决

commit c720b9b8daad4de80d50d4c250edf62bdd566d75

Author: xxh <xxh@rock-chips.com>

Date: Fri Dec 15 14:24:21 2017 +0800

Bluetooth: solve some Bluetooth HID Device Pair fail problem

Change-Id: I52a6bdb69011e6b3d15a6df9141d202c53aa9a8c

Signed-off-by: Xu Xuehui <xxh@rock-chips.com>

xxh@RD-DEP1-SERVER-163:~/work/rk3328_android_8.0_pro/packages/apps/TvSettings$

3. ATV BT remote ctrl 和 BLE 遥控器回连问题（解决 100%）

BLE 测试中有关配对的操作，需要将 RC 去掉，这是 ATV 上面 GOOGEL 自带的 RC 引起，过 CTS 需要去掉，以后有遥控器需求可以放开即可，目前 SDK 默认关闭

```
xxh@RD-DEP1-SERVER-163:~/work/rk3328_android_8.0_pro$ cd device/rockchip/rk3328/
xxh@RD-DEP1-SERVER-163:~/work/rk3328_android_8.0_pro/device/rockchip/rk3328$ git diff ./
diff --git a/bluetooth/bdroid_buildcfg.h b/bluetooth/bdroid_buildcfg.h
index 442b8da..4c74241 100755
--- a/bluetooth/bdroid_buildcfg.h
+++ b/bluetooth/bdroid_buildcfg.h
@@ -20,7 +20,7 @@
 #define BTM_DEF_LOCAL_NAME     "rk3328"
 #define BTA_DM_COD {0x1A, 0x01, 0x10}

-#define BLE_PRIVACY_SPT TRUE
+#define BLE_PRIVACY_SPT FALSE
 #define BLE_VND_INCLUDED TRUE
 #endif
```

4. ATV BLE CtsVerifier 测试（解决 100%）

关闭如下宏，测试完全 PASS

```
xxh@RD-DEP1-SERVER-163:~/work/rk3328_android_8.0_pro/device/rockchip/rk3328$ git show c7
commit c746ad7c288c4086c82872dfa1d8a809198a616d
Author: Xu Xuehui <xxh@rock-chips.com>
Date:   Tue Dec 26 10:07:36 2017 +0800

    bluetooth: remove rc for BLE cts test

    Change-Id: Id5766deb7b60478590b0dfa8cc9ca6533db70e93
    Signed-off-by: Xu Xuehui <xxh@rock-chips.com>

diff --git a/bluetooth/bdroid_buildcfg.h b/bluetooth/bdroid_buildcfg.h
index 4c74241..442b8da 100755
--- a/bluetooth/bdroid_buildcfg.h
+++ b/bluetooth/bdroid_buildcfg.h
@@ -20,7 +20,7 @@
 #define BTM_DEF_LOCAL_NAME     "rk3328"
 #define BTA_DM_COD {0x1A, 0x01, 0x10}

-#define BLE_PRIVACY_SPT FALSE
+#define BLE_PRIVACY_SPT TRUE
 #define BLE_VND_INCLUDED TRUE
 #endif
```

5. ATV 无法接受文件（解决 100%）

ATV 上默认是关闭了 OPP Profile，打开即可支持文件传输：**profile_supported_opp**

```
diff --git a/overlay/packages/apps/Bluetooth/res/values/config.xml b/overlay/packages/apps/Bluetooth/res/values/config.xml
index 2385dd2..499bfd5 100644
--- a/overlay/packages/apps/Bluetooth/res/values/config.xml
+++ b/overlay/packages/apps/Bluetooth/res/values/config.xml
@@ -15,7 +15,7 @@
 <resources>
     <bool name="profile_supported_pbap">false</bool>
     <bool name="profile_supported_map">false</bool>
-    <bool name="profile_supported_opp">false</bool>
+    <bool name="profile_supported_opp">true</bool>

     <string name="pairing_ui_package">com.android.tv.settings</string>
 </resources>
```

```
xxh@RD-DEP1-SERVER-163:~/work/rk3328_android_8.0_pro/packages/apps/Bluetooth$ git diff ./
diff --git a/src/com/android/bluetooth/opp/BluetoothOppNotification.java b/src/com/android/bluetooth/opp/BluetoothOppNotification.java
index 19d359d..4bc712b 100644
--- a/src/com/android/bluetooth/opp/BluetoothOppNotification.java
+++ b/src/com/android/bluetooth/opp/BluetoothOppNotification.java
@@ -51,6 +51,8 @@ import android.os.Message;
 import android.os.Process;

 import java.util.HashMap;
+import android.os.SystemProperties;
+import android.util.Log;

 /**
  * This class handles the updating of the Notification Manager for the cases
@@ -512,6 +514,10 @@ class BluetoothOppNotification {
             Uri contentUri = Uri.parse(BluetoothShare.CONTENT_URI + "/" + info.mID);
             Intent baseIntent = new Intent().setDataAndNormalize(contentUri)
                     .setClassName(Constants.THIS_PACKAGE_NAME, BluetoothOppReceiver.class.getName());
+            if (SystemProperties.get("ro.target.product", "atv").equals("atv")) {
+                Log.d(TAG, "this is box or atv,so dircet send file.");
+                mContext.sendBroadcast(new Intent(baseIntent).setAction(Constants.ACTION_INCOMING_FILE_CONFIRM));
+            } else {
             Notification.Action actionDecline =
                     new Notification.Action
                             .Builder(R.drawable.ic_decline,
@@ -556,6 +562,7 @@ class BluetoothOppNotification {
                             .build();
             mNotificationMgr.notify(NOTIFICATION_ID_PROGRESS, n);
         }
+        }
         cursor.close();
     }
 }
```

6. 蓝牙配对音响后，关闭蓝牙，发现蓝牙上报的断开连接时间比较长，达到 20S

如果是由 AP6356S 主动去连接音箱，这个值就会被设置，默认是 5 秒

但是当由音箱回连时候，是音箱作为 Master，但是并没有看到音箱設置 timeout 时间，因此协议栈里面默认就是 20s

define HCI_DEFAULT_INACT_TOUT 0x7D00 /* BR/EDR (20 seconds) */

这个其实跟外面蓝牙设备有关，与同样音响跟手机连接，测试发现断开连接也是 5S，音响配置无法修改，只能修改协议栈。如下修改可以解决，目前自测是 1 秒钟，（提交暂时未上 SDK，需要验证对其他蓝牙行为是否有音响，因为 BOX 和 ATV 是同一套 SDK，ATV 上不建议修改，可能会影响 CTS 或者其他测试）

修改如下：

/* link supervision timeout in 625uS (5 secs) */

#ifndef BTA_DM_LINK_TIMEOUT

Supervision_Timeout: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Connection supervision timeout. Range: 0x000A to 0x0C80 Time = N * 10 msec Time Range: 100 msec to 32 seconds

```
xxh@RD-DEP1-SERVER-163:~/work/rk3328_box_android_8.0_pro/system/bt$ git diff ./
diff --git a/bta/dm/bta_dm_cfg.cc b/bta/dm/bta_dm_cfg.cc
index 4202975..0850a78 100644
--- a/bta/dm/bta_dm_cfg.cc
+++ b/bta/dm/bta_dm_cfg.cc
@@ -44,7 +44,7 @@

 /* link supervision timeout in 625uS (5 secs) */
 #ifndef BTA_DM_LINK_TIMEOUT
-#define BTA_DM_LINK_TIMEOUT 8000
+#define BTA_DM_LINK_TIMEOUT 1600
 #endif

 /* TRUE to avoid scatternet when av is streaming (be the master) */
diff --git a/include/bt_target.h b/include/bt_target.h
index eadfc92..4c37afb 100644
--- a/include/bt_target.h
+++ b/include/bt_target.h
@@ -567,7 +567,7 @@

 /* Whether link wants to be the master or the slave. */
 #ifndef L2CAP_DESIRED_LINK_ROLE
-#define L2CAP_DESIRED_LINK_ROLE HCI_ROLE_SLAVE
+#define L2CAP_DESIRED_LINK_ROLE HCI_ROLE_MASTER
 #endif
```

5 wifi ko 编译注意事项

通过跟涛哥沟通，认为可以直接在 make kenrel.img 的时候，顺带吧系统的 moduel 编译出来，

arch/arm64/Makefile 已经集成，android 只需进行一次拷贝即可

```
        @echo '  Image:  kernel
ifdef CONFIG_MODULES
        $(Q)$(MAKE) modules
endif
```

Android 拷贝还是按照 android 7.1 的脚本做法，为了方便，**直接将该脚本集成到了./mkming.sh 文件中**

device/rockchip/common/build_wifi_ko.sh 脚本内容：

```
#!/bin/bash
TARGET_ARCH=$1
TARGET_OUT_VENDOR=$2
echo "---- make wifi ko  ----"
echo "TARGET_ARCH = $TARGET_ARCH"

#make -C kernel ARCH=$TARGET_ARCH  modules -j8

mkdir -p $TARGET_OUT_VENDOR/lib/modules/wifi

find kernel/drivers/net/wireless/rockchip_wlan/*  -name "*.ko" | \
xargs -n1 -i cp {} $TARGET_OUT_VENDOR/lib/modules/wifi/
echo "Install wifi ko to $TARGET_OUT_VENDOR/lib/modules/wifi/"
~
~
```

集成到./mkimag.sh 中的后，每次编译打包都会进行拷贝。

```
if [ `grep "CONFIG_WIFI_BUILD_MODULE=y" $KERNEL_CONFIG` ]; then
    echo "Install wifi ko to $TARGET_OUT_VENDOR/lib/modules/wifi/"
    mkdir -p $TARGET_OUT_VENDOR/lib/modules/wifi
    find kernel/drivers/net/wireless/rockchip_wlan/*  -name "*.ko" | xargs -n1 -i cp {} $TARGET_OUT_VENDOR/lib/modules/wifi/
fi
```

6 Wifi 驱动加载方式说明

如果**不需要 wifi 自动兼容方案**，可以将 wifi 驱动 build in 到内核，因此需要客户自行决定。目前最新对外服务器 andriod 7.1 代码支持 ko 或者 build in 两种方式，如果是 build in 到内核，建议使用如下表格中的方式 1，以 RTL8188EU 为例，ko 方式和 build in 方式详细配置如下表：

wifi 驱动加载方式	配置
方式 1	CONFIG_WIFI_LOAD_DRIVER_WHEN_KERNEL_BOOTUP=y WIFI_BUILD_MODULE = n CONFIG_RTL8188EU=y
方式 2	CONFIG_WIFI_LOAD_DRIVER_WHEN_KERNEL_BOOTUP=n WIFI_BUILD_MODULE = n CONFIG_RTL8188EU=y
方式 3	CONFIG_WIFI_LOAD_DRIVER_WHEN_KERNEL_BOOTUP=n WIFI_BUILD_MODULE = y CONFIG_RTL8188EU=m

说明：方式 3 即是前面章节提到完全兼容配置，方式 1 和方式 2 不支持 wifi 完全兼容,kernel 4.4 和 kernel 3.10 只需注意注意区分 64 位和 32 位，编译 wifi ko 是 make ARCH=arm64 modules 或者 make modules

方式 1：wifi 驱动 build in 到内核，内核启动阶段直接加载 wifi 驱动

方式 2：wifi 驱动 build in 到内核，打开 wifi 再加载驱动，说明方式 2 是为了兼容 android 5.1

方式 3：wifi 驱动以 ko 方式存在于系统，打开 wifi 再加载 ko

7 Wifi 、BT dts 配置

目前 RK 平台有 kenrel 4.4 和 kernel 3.10，对于不同的内核版本,DTS 有配置是有差异的，本章节对不同平台的 DTS WIFI BT 部分做相关说明

7.1 kernel 4.4 平台

以 RK3328 为例，**需要注意如下 pinctrl 的配置，其中 sdio-pwrseq 是 WIFI_REG_ON 管脚和蓝牙 rts_gpio 管脚，32K 是 RK805 提供，因此要放开此 CLK，其他 RK 平台类似这样修改。**

```
wireless-bluetooth {
        compatible = "bluetooth-platdata";
        clocks = <&rk805 1>;
        clock-names = "ext_clock";
        uart_rts_gpios = <&gpio1 10 GPIO_ACTIVE_LOW>;
        pinctrl-names = "default", "rts_gpio";
        pinctrl-0 = <&uart0_rts>;
        pinctrl-1 = <&uart0_gpios>;
        BT,power_gpio = <&gpio1 21 GPIO_ACTIVE_HIGH>;
        BT,wake_host_irq = <&gpio1 26 GPIO_ACTIVE_HIGH>;
        status = "okay";
};

wireless-wlan {
        compatible = "wlan-platdata";
        rockchip,grf = <&grf>;
        wifi_chip_type = "ap6354";
        sdio_vref = <1800>;
```

```
                WIFI,host_wake_irq = <&gpio1 19 GPIO_ACTIVE_HIGH>; /*wifi wake host 管脚*/
                status = "okay";
        };
};
&pinctrl {
        sdio-pwrseq {
                wifi_enable_h: wifi-enable-h {
                rockchip,pins =
                        <1 18 RK_FUNC_GPIO &pcfg_pull_none>;
                };
        };
        wireless-bluetooth {
                uart0_gpios: uart0-gpios {
                rockchip,pins =
                        <1 10 RK_FUNC_GPIO &pcfg_pull_none>;
                };
        };
};
```

7.2 kernel 3.10 平台

WIFI BT 管脚 DTS 配置如下，同样以 RK3328 平台为例

```
        wireless-wlan {
                compatible = "wlan-platdata";
                wifi_chip_type = "ap6335";
```

```
        sdio_vref = <1800>;
        WIFI,poweren_gpio = <&gpio1 GPIO_C2 GPIO_ACTIVE_HIGH>;
        WIFI,host_wake_irq = <&gpio1 GPIO_C3 GPIO_ACTIVE_HIGH>;
        status = "okay";
};

wireless-bluetooth {
        compatible = "bluetooth-platdata";
        uart_rts_gpios = <&gpio1 GPIO_B2 GPIO_ACTIVE_LOW>;
        pinctrl-names = "default", "rts_gpio";
        pinctrl-0 = <&uart0_rts>;
        pinctrl-1 = <&uart0_rts_gpio>;
        BT,power_gpio = <&gpio1 GPIO_C5 GPIO_ACTIVE_HIGH>;
        BT,wake_host_irq = <&gpio1 GPIO_D2 GPIO_ACTIVE_HIGH>;
        status = "okay";
};
```