

密级状态: 绝密 () 秘密 () 内部资料 () 公开 (✓)

文档编号: RKXXXX (芯片型号) - XXXXXX (英文、数字)

RealTek wifi 驱动移植说明

(系统产品一部)

文件状态: [] 正在修改 [✓] 正式发布	当前版本:	V1.1
	作 者:	许学辉
	完成日期:	2016-02-15

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

[illegible]

目 录

1. 驱动修改说明:	4
1. 修改 RealTek wifi 驱动入口函数	4
2. 修改 RealTek wifi 驱动 Makefile	4
3. RealTek wifi 驱动 debug LOG 开关	5
2. REALTEK WIFI 驱动 KO 编译方法.....	7
3. REALTEK WIFI 驱动中相关重要参数说明	8
4. REALTEK 驱动添加相关功能说明	8

1. 驱动修改说明:

1. 修改 RealTek wifi 驱动入口函数

一般来说，RealTek 的驱动，都有统一的入口函数

Realtek USB WIFI 驱动入口在 wifi_dirver/os_dep/linux/usb_intf.c 的 rtw_drv_entry 函数中

Realtek SDIO WIFI 驱动入口在 wifi_dirver/os_dep/linux/sdio_intf.c 的 rtw_drv_entry 函数中

确认如下内容即可:

```
static int __init rtw_drv_entry(void)

static void __exit rtw_drv_halt(void)

module_init(rtw_drv_entry);

module_exit(rtw_drv_halt);
```

2. 修改 RealTek wifi 驱动 Makefile

```
CONFIG_PLATFORM_ANDROID_X86 = n

CONFIG_PLATFORM_ARM_RK3188 = y

ifneq ($(CONFIG_PLATFORM_ARM_RK3188), y)
EXTRA_CFLAGS += -DCONFIG_LITTLE_ENDIAN -DCONFIG_PLATFORM_ANDROID -DCONFIG_PLATFORM_ROCKCHIPS
# default setting for Android 4.1, 4.2, 4.3, 4.4
EXTRA_CFLAGS += -DCONFIG_IOCTL_CFG80211 -DRTW_USE_CFG80211_STA_EVENT
EXTRA_CFLAGS += -DCONFIG_CONCURRENT_MODE
# default setting for power control
EXTRA_CFLAGS += -DRTW_ENABLE_WIFI_CONTROL_FUNC
#EXTRA_CFLAGS += -DRTW_SUPPORT_PLATFORM_SHUTDOWN
# default setting for special function
EXTRA_CFLAGS += -DCONFIG_P2P_IPS
EXTRA_CFLAGS += -DCONFIG_RADIO_WORK
EXTRA_CFLAGS += -DCONFIG_RESUME_IN_WORKQUEUE
ARCH := arm
#CROSS_COMPILE := /home/android_sdk/rockchip/Rk3188/prebuilts/gcc/linux-x86/arm/arm-eabi-4.6/bin/arm-eabi-
#CROSS_COMPILE := /home1/xxh/work/3288_cmcc_sdk/prebuilts/gcc/linux-x86/arm/arm-eabi-4.6/bin/arm-eabi-
CROSS_COMPILE := /home1/xxh/work/rk3368_5.1_develop/prebuilts/gcc/linux-x86/aarch64/aarch64-linux-android-4.9/bin/aarch64-linux-android-
KSRC := /home/xxh/work/3.10-kernel/kernel
MODULE_NAME := 8822bs
```

需要修改 MODULE_NAME 名称，默认是 wlan, 如果是 RTL8822BS 模块，修改为 8822bs，其他模块类似修改即可。

Makfile 中几个重要的宏在 RealTek Realse 的文档中都会有说明，这些配置打开后驱动才会支持 WFD P2P 以及 CFG80211，具体可以参考 RealTek 驱动说明文档中的如下内容:

6. Driver Configurations for Android JB

Android JB support two scenarios for Wi-Fi solution:

- STA/AP – Switch between STA and AP mode
- (STA+P2P)/AP – Switch between STA+P2P concurrent and AP mode

The configuration of driver to fit the requirement of each scenario, see the following table:

MACRO	STA/AP	(STA+P2P)/AP	Kernel ver.
CONFIG_IOCTL_CFG80211	Defined	Defined	ver. >= 2.6.35
RTW_USE_CFG80211_STA_EVENT	Defined	Defined	ver. >= 3.2.0
CONFIG_CONCURRENT_MODE	Undefined	Defined	-
CONFIG_P2P_IPS	Don't Care	Defined	-

这里列举一下 RTL8723BU 的这部分内容：

```
ifeq ($(CONFIG_PLATFORM_ARM_RK3188), y)

EXTRA_CFLAGS += -DCONFIG_LITTLE_ENDIAN -DCONFIG_PLATFORM_ANDROID
-DCONFIG_PLATFORM_ROCKCHIPS

# default setting for Android 4.1, 4.2, 4.3, 4.4

EXTRA_CFLAGS += -DCONFIG_IOCTL_CFG80211 -DRTW_USE_CFG80211_STA_EVENT

EXTRA_CFLAGS += -DCONFIG_CONCURRENT_MODE

# default setting for Special function

EXTRA_CFLAGS += -DCONFIG_P2P_IPS

EXTRA_CFLAGS += -DCONFIG_RADIO_WORK //兼容 android 5.0 的 P2P

EXTRA_CFLAGS += -DCONFIG_RESUME_IN_WORKQUEUE //控制 wifi resume 的过程，
将相关动作放在 workqueue 中完成，避免休眠唤醒慢的问题

ARCH := arm

CROSS_COMPILE :=

/home/android_sdk/Rockchip/Rk3188/prebuilts/gcc/linux-x86/arm/arm-eabi-4.6/bin/arm-eabi-

KSRC := /home/android_sdk/Rockchip/Rk3188/kernel

MODULE_NAME := 8723bu
```

3. RealTek wifi 驱动 debug LOG 开关

#define CONFIG_DEBUG /* DBG_871X, etc... */ #注释掉这行可以关闭掉 wifi 驱动的 debug 信息

```
//Older Android kernel doesn't has CONFIG_ANDROID defined,
//add this to force CONFIG_ANDROID defined
//ifndef CONFIG_PLATFORM_ANDROID
//define CONFIG_ANDROID
//endif
```

4. RealTek wifi 驱动整合到内核

将驱动按照如下步骤整合：

- 1、修改 drivers/net/wireless/Makefile

```

xxh@RD-DEPL-SERVER-163:~/work/rk3328/kernel$ vim drivers/net/wireless/Makefile
#
Makefile for the Linux wireless network device drivers.
<
obj-y += rockchip_wlan/wifi_sys/rkwifi_sys_iface.o
obj-$(CONFIG_RTL8192CU) += rockchip_wlan/rtl8192cu/
obj-$(CONFIG_RTL8192DU) += rockchip_wlan/rtl8192du/
obj-$(CONFIG_RTL8188EU) += rockchip_wlan/rtl8188eu/
obj-$(CONFIG_RTL8189ES) += rockchip_wlan/rtl8189es/
obj-$(CONFIG_RTL8723AU) += rockchip_wlan/rtl8723au/
obj-$(CONFIG_RTL8723BU) += rockchip_wlan/rtl8723bu/
obj-$(CONFIG_RTL8812AU) += rockchip_wlan/rtl8812au/
obj-$(CONFIG_RKWIFI) += rockchip_wlan/rkwifi/
obj-$(CONFIG_RTL8723BS) += rockchip_wlan/rtl8723bs/
obj-$(CONFIG_RTL8723BS_VQ0) += rockchip_wlan/rtl8723bs-vq0/
obj-$(CONFIG_RTL8723CS) += rockchip_wlan/rtl8723cs/
obj-$(CONFIG_RTL8723DS) += rockchip_wlan/rtl8723ds/
obj-$(CONFIG_RTL8189FS) += rockchip_wlan/rtl8189fs/
obj-$(CONFIG_RTL8188FU) += rockchip_wlan/rtl8188fu/
obj-$(CONFIG_ESP8089) += rockchip_wlan/esp8089/
obj-$(CONFIG_RTL8822BS) += rockchip_wlan/rtl8822bs/

```

2、修改 drivers/net/wireless/Kconfig

```

config BUILD_REALTEK_WIFI_MODULES
    bool "build realtek wifi as module"
    default y

source "drivers/net/wireless/rockchip_wlan/rtl8188eu/kconfig"
source "drivers/net/wireless/rockchip_wlan/rtl8189es/kconfig"
source "drivers/net/wireless/rockchip_wlan/rtl8192cu/kconfig"
source "drivers/net/wireless/rockchip_wlan/rtl8192du/kconfig"
source "drivers/net/wireless/rockchip_wlan/rtl8723au/kconfig"
source "drivers/net/wireless/rockchip_wlan/rtl8723bu/kconfig"
source "drivers/net/wireless/rockchip_wlan/rtl8723bs/kconfig"
source "drivers/net/wireless/rockchip_wlan/rtl8723bs-vq0/kconfig"
source "drivers/net/wireless/rockchip_wlan/rtl8723cs/kconfig"
source "drivers/net/wireless/rockchip_wlan/rtl8723ds/kconfig"
source "drivers/net/wireless/rockchip_wlan/rtl8812au/kconfig"
source "drivers/net/wireless/rockchip_wlan/rtl8189fs/kconfig"
source "drivers/net/wireless/rockchip_wlan/rtl8188fu/kconfig"
source "drivers/net/wireless/rockchip_wlan/rtl8822bs/kconfig"
source "drivers/net/wireless/rockchip_wlan/esp8089/kconfig"
source "drivers/net/wireless/rockchip_wlan/rkwifi/kconfig"
source "drivers/net/wireless/rockchip_wlan/ssv6xxx/kconfig"

```

3、添加驱动到 drivers/net/wireless/rockchip_wlan 目录

```

~/work/rk3328/kernel/drivers/net/wireless/rockchip_wlan$
4096 Feb  9 15:03 esp8089
4096 Feb  9 15:03 rkwifi
4096 Feb  9 15:04 rtl8188eu
4096 Dec  9 10:50 rtl8188fu
4096 Dec  9 10:50 rtl8189es
4096 Dec 27 18:13 rtl8189fs
4096 Dec  9 10:50 rtl8192cu
4096 Dec  9 10:50 rtl8192du
4096 Dec  9 10:50 rtl8723as
4096 Dec  9 10:50 rtl8723au
4096 Feb  9 15:04 rtl8723bs
2998 Jan 12 08:55 rtl8723bs.7z
4096 Feb  5 10:18 rtl8723bs-vq0
4096 Dec  9 10:50 rtl8723bu
4096 Dec 27 18:13 rtl8723cs
4096 Jan 12 09:07 rtl8723ds
4096 Dec  9 10:50 rtl8812au
4096 Feb 15 12:01 rtl8822bs
0460 Jan 12 08:51 rtl8822bs.7z
4096 Feb  5 10:18 rtl8822bu
4697 Jan 12 08:53 rtl8822bu.7z
4096 Dec  9 10:50 ssv6xxx

```

2. RealTek wifi 驱动 KO 编译方法

1、如果是 3.0 内核需要将驱动解压到内核 kernel/drivers/net/wireless 目录下；如果是 3.10 内核，解压到 kernel/drivers/net/wireless/rockchip_wlan

2、一些使用相同驱动的需要先执行 make_drv 并选择，比如 8188etv or 8189es

3、然后修改驱动源码包中的 Makefile 文件

ARCH := arm

#CROSS_COMPILE

:= /usr/src/release_fae_version/toolchain/arm-eabi-4.4.0/bin/arm-eabi-

KSRC := /xxxx/kernel #这个改成实际内核的路径，也就是具体项目的内核路径

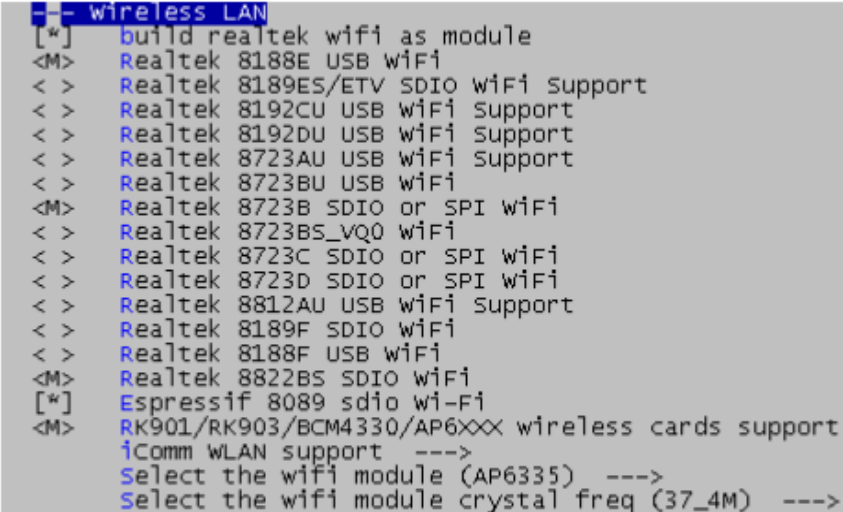
4、修改内核.config 或者 make menuconfig 根据实际情况选择需要编译的 WIFI 驱动

make ARCH=arm64 menuconfig

Device Drivers --->

[*] Network device support --->

[*] Wireless LAN --->



```
[-] wireless LAN
[*] build realtek wifi as module
<M> Realtek 8188E USB wifi
< > 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
< > Realtek 8188F USB wifi
<M> Realtek 8822BS SDIO wifi
[*] Espressif 8089 sdio wi-fi
<M> RK901/RK903/BCM4330/AP6xxx wireless cards support
iComm WLAN support --->
Select the wifi module (AP6335) --->
Select the wifi module crystal freq (37_4M) --->
```

在内核根目录直接运行 **make modules** 即可编译出相应的 **wifi ko**

5、ko 编译成功后，通过如下方式可以确定使用的 wifi 驱动版本：

方法 1：直接查看驱动源码 include/rtw_version.h

方法 2：烧写固件后，进入 adb 或者串口 cat proc/net/rtl8822bs/drv_cfg 查看

rk3328_box:/ # cat proc/net/rtl8822bs/drv_cfg

Kernel Version: 3.10.104

Driver Version: v5.1.7_20554.20161209_COEX20161026-3434_aml

```
-----
CFG80211
RTW_USE_CFG80211_STA_EVENT
DBG:0
CONFIG_CONCURRENT_MODE
LOAD_PHY_PARA_FROM_FILE - REALTEK_CONFIG_PATH=/system/etc/firmware/
RTW_DEF_MODULE_REGULATORY_CERT=0x00
CONFIG_TXPWR_BY_RATE_EN=1
CONFIG_TXPWR_LIMIT_EN=0
CONFIG_RTW_ADAPTIVITY_EN = 0
CONFIG_TX_AGGREGATION
MAX_XMITBUF_SZ = 20480
MAX_RECVBUF_SZ = 65536
```

3. RealTek wifi 驱动中相关重要参数说明

1、SURVEY_TO ==> channel remain time;

定义位置在 include/rtw_mlme_ext.h 文件中。

#define SURVEY_TO (100) 说明每个 channel 扫描的时候保持的时间为100ms。

增大这个时间可以一定程度上提高扫描 ap 数量稳定性，特别是对于信号强度比较差的 AP，但是也会拉长一次扫描的时间，太大了反而影响其他 channel 扫描。一般情况下不建议修改。

2、int rtw_channel_plan = RT_CHANNEL_DOMAIN_MAX; ==> channel set;

这个定义是 wifi 驱动 country code 定义，可以修改来使默认支持的通道数量比如1~11、1~13、1~14。默认情况下是13个通道，

RT_CHANNEL_DOMAIN_FCC ==> US

RT_CHANNEL_DOMAIN_ETSI ==> EU

RT_CHANNEL_DOMAIN_MKK ==> JP

RT_CHANNEL_DOMAIN_CHINA ==> CN

RT_CHANNEL_DOMAIN_GLOBAL_DOAMIN ==> IN

3、notify_signal ==> wifi signal level;

这个变量在文件 os_dep/linux/ioctl_cfg80211.c 中是每个扫描到的 AP 往上层上报 RSSI 大小的地方，修改这个可以改变设置中扫描到的所有 AP 的信号强度，比如在原先基础上增大 100*10，那么设置中所有的 AP 都将增大10dbm 的信号强度。

一般情况下不建议修改，这个是信号认为造假的行文，对实际通信没有任何好处，一般信号问题都是硬件方面的问题，解决信号问题需要从硬件方面去改善或者优化才是最有效的方法。

4. RealTek 驱动添加相关功能说明

1、自定义 MAC 地址功能

目前 android 4.4 已经都默认添加了自定义 MAC 地址功能，在打开 wifi 的时候会先去 flash 中的特定存储区域读取 MAC 地址，先判断 MAC 地址是否合法（是否都是 0，或者是否组播

地址), 如果是合法的 wifi 驱动将使用读取到的 MAC 地址, 否则将是是否 wifi 模组本身的 MAC 地址。

修改的地方在驱动 core/rtw_ieee80211.c 文件 void rtw_macaddr_cfg(u8 *mac_addr)这个函数中, **rk29sdk_wifi_mac_addr(macbuf)**这个函数在 board 文件中定义作用是从 flash 中读取 MAC 地址并判断 MAC 地址是否合法。注意, 在最新 SDK kernel 3.10 上, 函数名为 int **rockchip_wifi_mac_addr(unsigned char *buf)**, 在 SDK 上 RealTek 中已经添加这个函数, 可以参考各个 RealTek WIFI 驱动相应问文件修改。

```
printk("Wifi Efuse Mac => %02x:%02x:%02x:%02x:%02x:%02x\n", mac_addr[0],
mac_addr[1], mac_addr[2], mac_addr[3], mac_addr[4], mac_addr[5]);
```

```
if (!rk29sdk_wifi_mac_addr(macbuf)) {
    int jj, kk;
    printk("=====> get mac address from flash %s\n", macbuf);
    for( jj = 0, kk = 0; jj < ETH_ALEN; jj++, kk += 3 )
    {
        mac[jj] = key_2char2num(macbuf[kk], macbuf[kk+ 1]);
    }
    _rtw_memcpy(mac_addr, mac, ETH_ALEN);
} else {
    // Use the mac address stored in the Efuse
    _rtw_memcpy(mac, mac_addr, ETH_ALEN);
}
```