



XR829 WLAN Porting Guide(Android O)

Revision 1.0

Jun 26, 2018

Declaration

THIS DOCUMENTATION IS THE ORIGINAL WORK AND COPYRIGHTED PROPERTY OF XRADIO TECHNOLOGY ("XRADIO"). REPRODUCTION IN WHOLE OR IN PART MUST OBTAIN THE WRITTEN APPROVAL OF XRADIO AND GIVE CLEAR ACKNOWLEDGEMENT TO THE COPYRIGHT OWNER.

THE INFORMATION FURNISHED BY XRADIO IS BELIEVED TO BE ACCURATE AND RELIABLE. XRADIO RESERVES THE RIGHT TO MAKE CHANGES IN CIRCUIT DESIGN AND/OR SPECIFICATIONS AT ANY TIME WITHOUT NOTICE. XRADIO DOES NOT ASSUME ANY RESPONSIBILITY AND LIABILITY FOR ITS USE. NOR FOR ANY INFRINGEMENTS OF PATENTS OR OTHER RIGHTS OF THE THIRD PARTIES WHICH MAY RESULT FROM ITS USE. NO LICENSE IS GRANTED BY IMPLICATION OR OTHERWISE UNDER ANY PATENT OR PATENT RIGHTS OF XRADIO. THIS DATASHEET NEITHER STATES NOR IMPLIES WARRANTY OF ANY KIND, INCLUDING FITNESS FOR ANY PARTICULAR APPLICATION.

THIRD PARTY LICENCES MAY BE REQUIRED TO IMPLEMENT THE SOLUTION/PRODUCT. CUSTOMERS SHALL BE SOLELY RESPONSIBLE TO OBTAIN ALL APPROPRIATELY REQUIRED THIRD PARTY LICENCES. XRADIO SHALL NOT BE LIABLE FOR ANY LICENCE FEE OR ROYALTY DUE IN RESPECT OF ANY REQUIRED THIRD PARTY LICENCE. XRADIO SHALL HAVE NO WARRANTY, INDEMNITY OR OTHER OBLIGATIONS WITH RESPECT TO MATTERS COVERED UNDER ANY REQUIRED THIRD PARTY LICENCE.

Revision History

Version	Date	Summary of Changes
1.0	2018-06-16	Initial Version

表 0-1 Revision History

目 录

Declaration.....	2
Revision History.....	3
目 录.....	4
表.....	5
图.....	6
1 Linux kernel 移植.....	7
1.1 添加 Xradio wlan 驱动.....	7
1.2 配置 Xradio 内核编译.....	7
1.3 Xradio 平台配置.....	8
1.3.1 通用平台配置.....	8
1.3.2 Allwinner 平台配置.....	9
2 Android 移植.....	13
2.1 拷贝 xradio 移植文件包.....	13
2.2 xradio wifi-hal 厂商库配置.....	13
2.3 wlan 模组配置.....	14
2.4 初始化配置.....	15
3 移植后检查点.....	17
3.1 确认 xradio_mac, xradio_core 开机加载.....	17
3.2 确认设备存在 fw、sdd 等文件.....	18
4 FAQ.....	19
4.1 无法正常打开 wifi.....	19
4.1.1 确认 xradio_wlan.ko 模块正常加载.....	19
4.1.2 确认使用匹配的 SDD 文件.....	19
4.1.3 确认 cfg80211 已经编译进 linux 内核 (lichee)	20
4.2 如何检查软件版本号.....	21
4.3 如何支持 wowlan 功能.....	21

表

表 0-1 Revision History.....	3
表 1-1 xradio 驱动平台配置接口.....	9



图 1-1 选中“Xradio WLAN support”编译配置.....	8
图 1-2 选中“Xradio MAC featrues”和“Xradio Driver features”编译配置.....	8
图 1-3 确认“Xradio MAC featrues”的编译配置.....	8
图 1-4 确认“Xradio Driver features”的编译配置.....	8
图 1-5 确认“Drvier debug features”的编译配置.....	8
图 1-6 确认“Allwinner rfkill driver”编译配置.....	10
图 1-7 PCB 原理图的 POWER TREE.....	11
图 1-8 PCB 原理图 WLAN 电源、中断配置.....	11
图 1-9 PCB 原理图中 SDIO 配置.....	12
图 3-1 确认设备开机加载 xradio_mac.ko 和 xradio_core.ko.....	17
图 3-2 确认设备存在 init.wireless.xradio.rc 脚本文件.....	18
图 3-3 确认设备存在 fw、sdd 等文件.....	18
图 4-1 正常加载 xradio_wlan.ko 的内核打印.....	19
图 4-2 使用 sddedit 工具查看/修改 sdd 文件.....	20
图 4-3 确认 cfg80211 已编译进内核.....	20
图 4-4 查看软件版本号.....	21

1 Linux kernel 移植

下面基于 32 位 Linux-4.9 内核版本（其它平台移植可以参考此），给出具体的移植步骤。

1.1 添加 Xradio wlan 驱动

(1) 把对应内核版本的 lichee/xradio 文件夹拷贝到以下路径：

/lichee/linux-4.9/drivers/net/wireless/

(2) 在内核编译的 Makefile 文件中，加入对驱动的编译语句。

修改 /lichee/linux-4.9/drivers/net/wireless/Makefile 文件，添加以下信息：

```
obj-$(CONFIG_XR_WLAN) += xradio/
```

(3) 添加相关内核配置选项。

修改 /lichee/linux-4.9/drivers/net/wireless/Kconfig 文件，添加以下信息：

```
source "drivers/net/wireless/xradio/Kconfig"
```

1.2 配置 Xradio 内核编译

通过 Linux 配置界面进行手动逐项配置。在 Linux 的内核路径下 (lichee/linux-4.9)，使用 make ARCH=arm menuconfig 命令显示配置界面，进行以下配置：

(1)选中“Device Drivers --->Network device support --->Wireless LAN -->Xradio WLAN support”：

```
--- Wireless LAN
[ ] ADMtek devices
[ ] Atheros/Qualcomm devices
[ ] Atmel devices
[ ] Broadcom devices
[ ] Cisco devices
[ ] Intel devices
[ ] Intersil devices
[ ] Marvell devices
[ ] MediaTek devices
[ ] Ralink devices
[ ] Realtek devices
[ ] Redpine Signals Inc devices
[ ] STMicroelectronics devices
[ ] Texas Instrument devices
[ ] Zynq devices
[*] XRadio WLAN support ---->
< > Wireless RNDIS USB support
< > Broadcom FullMAC wireless cards support
< > Realtek 8723C SDIO or SPI WiFi
< > Realtek 8723BS-VQ0 WiFi
```


图 1-1 选中“Xradio WLAN support”编译配置

(2) 进入 Xradio WLAN support 子目录，分别选中 Xradio MAC features 和 Xradio Driver features:

```
--- XRadio WLAN support
<M> XRadio MAC features --->
<M> XRadio Driver features --->
```

图 1-2 选中“Xradio MAC features”和“Xradio Driver features”编译配置

(3) 进入 Xradio MAC features 子目录，确认默认情况如下选项被选中:

```
-- XRadio MAC features
(minstrel_ht) XRMAC_RC_DEFAULT
[ ] XRMAC_RC_PID
[*] XRMAC_RC_MINSTREL
[*] Minstrel 802.11n support
[*] XRMAC_DEBUGFS
```

图 1-3 确认“Xradio MAC features”的编译配置

(4) 返回上级 Xradio WLAN support 目录，再进入 Xradio Driver features 子目录，确认默认情况如下选项被选中:

```
-- XRadio Driver features
[*] SDIO bus wlan module
[*] Platform supports non-power-of-two SDIO transfer
[*] Use GPIO interrupt
[ ] Wlan power off in suspend
[ ] Support extend suspend(both Wow and power off)
[ ] Support normal suspend(Wow for Single Crystal)
Driver debug features --->
```

图 1-4 确认“Xradio Driver features”的编译配置

(5) 进入 Driver debug features 子目录，确认以下选项默认被选中:

```
[*] Enable XRADIO debug messages (DEVELOPMENT)
[*] XRADIO ETF Support for RF Test(DEVELOPMENT)
[ ] Dump kernel in case of critical error (DEVELOPMENT)
[*] Expose driver internals to DebugFS (DEVELOPMENT)
```

图 1-5 确认“Driver debug features”的编译配置

1.3 Xradio 平台配置

1.3.1 通用平台配置

平台配置一般会涉及上电、中断等配置，此外，还涉及系统唤醒、SDIO 扫描相关的接口配置。在 xradio/wlan/platform.h 声明了 xradio 驱动所需要的平台配置接口，接口实现则在 xradio/wlan/platform.c 中（代码为 demo，具体平台需重新实现）。具体说明如下表所示：

接口名称	参数描述	功能描述	备注
xradio_plat_init	无	用于平台配置的初始化，在 xradio_wlan 模块加载时被执行	
xradio_plat_deinit	无	用于平台配置的资源注销，在 xradio_wlan 模块卸载时被执行	
xradio_sdio_detect	enable: 是否插入卡	进行 SDIO 的扫描检测	
xradio_request_gpio_irq	dev: 注册中断的设备 sbus_priv: 中断处理函数以及相关参数的结构指针	注册数据中断	中断类型必须为上升沿触发，需支持 wow 功能则需支持唤醒系统
xradio_free_gpio_irq	dev: 注销中断的设备 sbus_priv: 已注册中断参数的结构指针	注销数据中断	
xradio_wlan_power	on: 1为打开，0为关闭	xradio 芯片上电实现，包括电源的开关以及复位 pin 脚的控制	上电 reset 时序，0(2ms)->1(50ms)
MCI_CHECK_READY	h:mmc_host, t:timeout	某些平台，需要用于 SDIO 数据传输后等待硬件是否 ready	

表 1-1 xradio 驱动平台配置接口

1.3.2 Allwinner 平台配置

1.3.2.1 上电控制模块修改

Allwinner 平台中，上电控制模块的代码路径为：lichee/linux-4.9/drivers/misc/sunxi-rf/

(1) 配置使用外部的 32K 时钟。修改 sunxi-wlan.c 文件中的 sunxi_wlan_probe() 函数：

将 `data->lpo = devm_clk_get(dev, NULL);`

修改为：`data->lpo = devm_clk_get(dev, "losc_out");`

(2) 确认内核编译配置：

在 lichee/linux-4.9 目录下输入 `make ARCH=arm menuconfig` 命令，依次选择“Device Drivers -> Misc devices”，确认“Allwinner rkll driver”被选中，页面配置如下：

```
< > Analog Devices Digital Potentiometers
< > Dummy IRQ handler
< > Integrated Circuits ICS932S401
< > Enclosure Services
< > Medfield Avago APDS9802 ALS Sensor module
< > Intersil ISL29003 ambient light sensor
< > Intersil ISL29020 ambient light sensor
< > Taos TSL2550 ambient light sensor
< > BH1770GLC / SFH7770 combined ALS - Proximity sensor
< > APDS990X combined als and proximity sensors
< > Honeywell HMC6352 compass
< > Dallas DS1682 Total Elapsed Time Recorder with Alarm
< > Texas Instruments DAC7512
< > FSA9480 USB Switch
< > Lattice ECP3 FPGA bitstream configuration via SPI
[ ] Generic on-chip SRAM driver
[*] Per-UID statistics
[ ] Per-TASK statistics
<*> Memory freq/bandwidth time statistics
< > Silicon Labs C2 port support ----
EEPROM support --->
Texas Instruments shared transport line discipline --->
< > STMicroelectronics LIS3LV02Dx three-axis digital accelerometer (SPI)
< > STMicroelectronics LIS3LV02Dx three-axis digital accelerometer (I2C)
*** Altera FPGA firmware download module ***
< > Altera FPGA firmware download module
*** Intel MIC Bus Driver ***
*** SCIF Bus Driver ***
*** VOP Bus Driver ***
*** Intel MIC Host Driver ***
*** Intel MIC Card Driver ***
*** SCIF Driver ***
*** Intel MIC Coprocessor State Management (COSM) Drivers ***
*** VOP Driver ***
< > Line Echo Cancellor support
[*] Allwinner rkll driver
```

图 1-6 确认“Allwinner rkll driver”编译配置

1.3.2.2 板级 sys_config.fex 配置

需要结合 PCB 原理图进行相关的硬件板级配置，这里以 AllWinner A33G 平台为例，给出相关配置。配置文件路径为：lichee/tools/pack/chips/sun8iw15p1/configs/a3/sys_config.fex

1.3.2.2.1 供电配置

(1) 查看 PCB 硬件原理图的 POWER TREE:

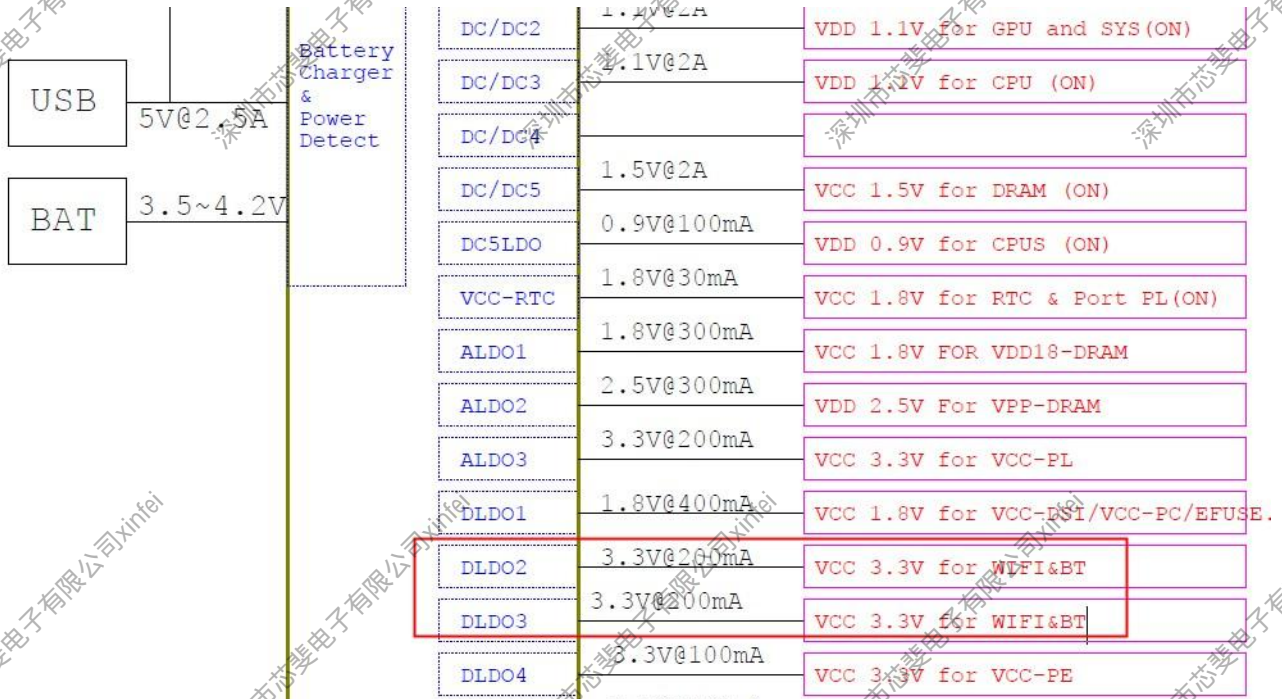


图 1-7 PCB 原理图的 POWER TREE

(2) 修改 sys_config.fex 文件，在[power_sply]元素中添加 dldo2_vol 和 dldo3_vol 的配置：

```

;-----
; power setting
;note:
;dc1sw_vol=1003300, mean set dc1sw to 3300mV and enable this dc
;dc1sw_vol=3300, mean set dc1sw to 3300mV but disable this dc
;-----

[power_sply]
dc1sw_vol      = 3300
;aldo1_vol     = 3300
;clldo3_vol    = 1800
dc1dc2_vol     = 100900
dc1dc3_vol     = 100900
dc1dc5_vol     = 1001500
dldo2_vol      = 3300
dldo3_vol      = 3300
;aldo5_vol     = 2500
    
```

1.3.2.2.2 wlan 电源、中断等配置

(1) 查看 PCB 原理图：

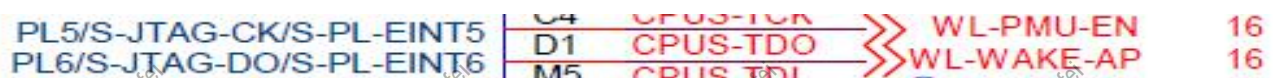


图 1-8 PCB 原理图 WLAN 电源、中断配置

(2) 在 sys_config.fex 文件中添加 wlan 电源、中断等配置:

```

;-----
;wlan configuration
;wlan_used:      0-not use, 1-use
;wlan_busnum:    0- SDC0, 1- SDC1, 2- SDC2, 3- SDC3
;wlan_power:     wlan module power supply
;wlan_io_regulator: wlan io power supply
;wlan_regon:     wlan power on
;wlan_hostwake:  wlan wakeup ap
;-----

[wlan]
compatible      = "allwinner,sunxi-wlan"
wlan_used       = 1
wlan_busnum     = 1
wlan_power      = "vcc-wifi"
wlan_io_regulator = "vcc-wifi-io"
wlan_regon      = port:PL05<1><default><default><0>
wlan_hostwake   = port:PL06<6><default><default><0>
chip_en         = port:PL07<1><default><default><0>

```

1.3.2.2.3 SDIO 配置

(1) 查看 PCB 原理图中的 SDIO 配置:

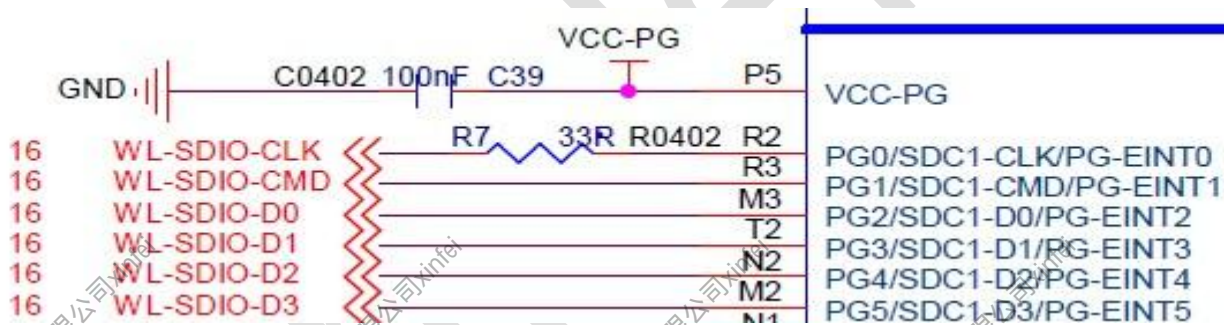


图 1-9 PCB 原理图中 SDIO 配置

(2) 在 sys_config.fex 文件中进行以下配置:

```

[sdc1]
sdc1_used      = 1
bus-width     = 4
sdc1_clk       = port:PG00<2><1><3><default>
sdc1_cmd       = port:PG01<2><1><3><default>
sdc1_d0        = port:PG02<2><1><3><default>
sdc1_d1        = port:PG03<2><1><3><default>
sdc1_d2        = port:PG04<2><1><3><default>
sdc1_d3        = port:PG05<2><1><3><default>

```

2 Android 移植

下面基于 AllWinner A33G 平台，给出 xradio wlan 在 Android 8.1 中的移植步骤。

2.1 拷贝 xradio 移植文件包

把对应 android 版本的 android/xradio 文件夹拷贝到 android/hardware/路径下。此目录包含 firmware、etf firmware、bootloader、sdd 等 bin 文件以及编译时使用的.mk 文件，驱动在初始化模组时会把这些文件下载到模组中。另外还包含了 wpa_supplicant_8_lib 文件夹和 wifi_hal 文件夹，用来生成函数库。

2.2 xradio wifi-hal 厂商库配置

(1) 文件路径: android/frameworks/opt/net/wifi/libwifi_hal/Android.mk

使用 xradio 实现的 wifi-hal 厂商库，添加以下高亮代码：

```
# Pick a vendor provided HAL implementation library.
# =====
LIB_WIFI_HAL := libwifi-hal-fallback
VENDOR_LOCAL_SHARED_LIBRARIES :=
ifeq ($(BOARD_WLAN_DEVICE), bcmhdhd)
    LIB_WIFI_HAL := libwifi-hal-bcm
else ifeq ($(BOARD_WLAN_DEVICE), qcwcn)
    LIB_WIFI_HAL := libwifi-hal-qcom
    VENDOR_LOCAL_SHARED_LIBRARIES := libcld80211
else ifeq ($(BOARD_WLAN_DEVICE), mrvl)
    # this is commented because none of the nexus devices
    # that sport Marvell's wifi have support for HAL
    # LIB_WIFI_HAL := libwifi-hal-mrvl
else ifeq ($(BOARD_WLAN_DEVICE), MediaTek)
    # support MTK WIFI HAL
    LIB_WIFI_HAL := libwifi-hal-mt66xx
else ifeq ($(BOARD_WLAN_DEVICE), realtek)
    # support Realtek WIFI HAL
    LIB_WIFI_HAL := libwifi-hal-rtk
```



```
else ifeq ($(BOARD_WLAN_DEVICE), xradio)
# support Xradio WIFI HAL
LIB_WIFI_HAL := libwifi-hal-xradio
endif
```

2.3 wlan 模组配置

1. 打开 android/device/softwinner/venus-a3/BoradConfig.mk 文件,

(1) 修改 BOARD_WIFI_VENDOR 宏的值为 **xradio**

(2) 添加 xradio wifi 配置

```
# wifi and bt configuration
# 1. Wifi Configuration

BOARD_WIFI_VENDOR := xradio

.....

# 1.3 eagle wifi configuration
ifeq ($(BOARD_WIFI_VENDOR), eagle)

    WPA_SUPPLICANT_VERSION := VER_0_8_X

    BOARD_WPA_SUPPLICANT_DRIVER := NL80211

    BOARD_WPA_SUPPLICANT_PRIVATE_LIB := lib_driver_cmd_eagle

    BOARD_HOSTAPD_DRIVER := NL80211

    BOARD_HOSTAPD_PRIVATE_LIB := lib_driver_cmd_eagle

    BOARD_USR_WIFI := esp8089

    BOARD_WLAN_DEVICE := esp8089

    include hardware/espressif/wlan/firmware/esp8089/device-esp.mk
endif

#1.4 xradio wifi config
ifeq ($(BOARD_WIFI_VENDOR), xradio)
# WiFi Configuration
WPA_SUPPLICANT_VERSION := VER_0_8_X
```

```
BOARD_WPA_SUPPLICANT_DRIVER := NL80211
BOARD_WPA_SUPPLICANT_PRIVATE_LIB := lib_driver_cmd_xr
BOARD_HOSTAPD_DRIVER := NL80211
BOARD_HOSTAPD_PRIVATE_LIB := lib_driver_cmd_xr

BOARD_WLAN_DEVICE := xradio

WIFI_DRIVER_MODULE_PATH := "/system/vendor/modules/xradio_wlan.ko"
WIFI_DRIVER_MODULE_NAME := "xradio_wlan"
WIFI_DRIVER_MODULE_ARG := ""

include hardware/xradio/wlan/kernel-firmware/xradio-wlan.mk
endif
```

2.4 初始化配置

修改 android/device/softwinner/common/init.wireless.xradio.rc 文件:

- (1) 添加 xradio_umac 和 xradio_core 开机启动加载脚本
- (2) 添加 service wpa_supplicant 启动脚本

```
on fs
    #xradio driver
    insmod /system/vendor/modules/xradio_mac.ko
    insmod /system/vendor/modules/xradio_core.ko

    insmod /system/vendor/modules/xradio_bt1pm.ko
    # bluetooth xradio firmware debugfs driver
    insmod /system/vendor/modules/xradio_btfdi.ko

    # UART device
    chmod 0660 /dev/ttyS1
    chown bluetooth net_bt_admin /dev/ttyS1
    .....
# 1. xradio wifi service
```



```

service wpa_supplicant /system/vendor/bin/hw/wpa_supplicant \
    -ip2p0 -Dnl80211 -c/data/misc/wifi/p2p_supplicant.conf \
    -I/vendor/etc/wifi/p2p_supplicant_overlay.conf \
    -e/data/misc/wifi/entropy.bin -N \
    -iwlan0 -Dnl80211 -c/data/misc/wifi/wpa_supplicant.conf \
    -I/vendor/etc/wifi/wpa_supplicant_overlay.conf \
    -O/data/misc/wifi/sockets \
    -g@android:wpa_wlan0

socket wpa_wlan0 dgram 660 wifi wifi

class main
    disabled
    oneshot
    
```

3 移植后检查点

3.1 确认 xradio_mac、xradio_core 开机加载

xradio 的 wlan 驱动 2.0 版本分为三个模块：xradio_mac.ko、xradio_core.ko 和 xradio_wlan.ko，这三个模块存在于设备上的 /system/vendor/modules/ 目录下。其中 android 系统启动时会自动加载 xradio_mac.ko 和 xradio_core.ko，而当打开 wifi 时才会触发 xradio_wlan.ko 的加载。通过以下命令确认是否开机加载了这两个模块：

```
adb shell
```

```
lsmod
```

```
venus-a3:/ # lsmod
lsmod
Module                Size  Used by
vin_v4l2              202575  1
gc0310_mipi           13720  0
gc2355_mipi           8116  0
vin_io                46000  3 vin_v4l2,gc0310_mipi,gc2355_mipi
videobuf2_v4l2        16405  1 vin_v4l2
videobuf2_dma_contig  10321  1 vin_v4l2
videobuf2_memops       2009  1 videobuf2_dma_contig
videobuf2_core        30705  2 vin_v4l2,videobuf2_v4l2
gslX680new            963052  0
mali                  309536  15
xradio_btfdi          17225  0
xradio_btlpm          12731  0
xradio_core           291170  0
xradio_mac            271492  1 xradio_core
venus-a3:/ #
```

图 3-1 确认设备开机加载 xradio_mac.ko 和 xradio_core.ko

若没有开机加载上述两个模块，可能原因是设备上不存在 `init.wireless.xradio.rc` 脚本文件。该文件应存放在设备上的以下目录：/vendor/etc/init/，请使用 “ls -l” 命令确认是否存在该文件：

```

venus-a3:/vendor/etc/init # ls -l
ls -l
total 100
-rw-r--r-- 1 root root 574 2018-06-22 18:49 android.hardware.audio@2.0-service.rc
-rw-r--r-- 1 root root 537 2018-06-22 18:49 android.hardware.bluetooth@1.0-service.rc
-rw-r--r-- 1 root root 272 2018-06-22 18:49 android.hardware.camera.provider@2.4-service.rc
-rw-r--r-- 1 root root 181 2018-06-22 18:49 android.hardware.cas@1.0-service.rc
-rw-r--r-- 1 root root 141 2018-06-22 18:49 android.hardware.configstore@1.0-service.rc
-rw-r--r-- 1 root root 181 2018-06-22 18:49 android.hardware.drm@1.0-service.rc
-rw-r--r-- 1 root root 199 2018-06-22 18:49 android.hardware.drm@1.0-service.widevine.rc
-rw-r--r-- 1 root root 125 2018-06-22 18:49 android.hardware.gatekeeper@1.0-service.rc
-rw-r--r-- 1 root root 212 2018-06-22 18:49 android.hardware.graphics.allocator@2.0-service.rc
-rw-r--r-- 1 root root 214 2018-06-22 18:49 android.hardware.graphics.composer@2.1-service.rc
-rw-r--r-- 1 root root 136 2018-06-22 18:49 android.hardware.keymaster@3.0-service.rc
-rw-r--r-- 1 root root 118 2018-06-22 18:49 android.hardware.light@2.0-service.rc
-rw-r--r-- 1 root root 199 2018-06-22 18:49 android.hardware.media.omx@1.0-service.rc
-rw-r--r-- 1 root root 125 2018-06-22 18:49 android.hardware.memtrack@1.0-service.rc
-rw-r--r-- 1 root root 119 2018-06-22 18:49 android.hardware.power@1.0-service.rc
-rw-r--r-- 1 root root 178 2018-06-22 18:49 android.hardware.sensors@1.0-service.rc
-rw-r--r-- 1 root root 120 2018-06-22 18:49 android.hardware.wifi@1.0-service.rc
-rw-r--r-- 1 root root 68 2018-06-22 18:49 android.suppllicant@2.5-optee.rc
-rw-r--r-- 1 root root 512 2018-06-22 18:49 hostapd.android.rc
drwxr-xr-x 2 root shell 4096 2018-06-22 18:49 hw
-rw-r--r-- 1 root root 0 2018-06-22 18:49 init.sensors.rc
-rw-r--r-- 1 root root 1859 2018-06-22 18:49 init.wireless.xradio.rc
-rw-r--r-- 1 root root 202 2018-06-22 18:49 preinstall.rc
-rw-r--r-- 1 root root 135 2018-06-22 18:49 radio_monitor.rc
-rw-r--r-- 1 root root 178 2018-06-22 18:49 rild.rc
-rw-r--r-- 1 root root 199 2018-06-22 18:49 vndservicemanager.rc
venus-a3:/vendor/etc/init #

```

图 3-2 确认设备存在 init.wireless.xradio.rc 脚本文件

如果不存在此文件，是 Android 层移植出现问题，请确认是 **否已正确进行“2.3 wlan 模组配置”**。另外，可通过依次手动加载 xradio_mac.ko 模块和 xradio_core.ko 模块，确认是否能成功加载。

3.2 确认设备存在 fw、sdd 等文件

上述文件存放在设备上的 /vendor/etc/firmware/ 目录，使用 ls -l 命令进行查看：

```

venus-a3:/vendor/etc/firmware # ls -l
ls -l
total 544
-rw-r--r-- 1 root root 2184 2018-06-22 18:49 boot_xr829.bin
-rw-r--r-- 1 root root 73004 2018-06-22 18:49 etf_xr829.bin
-rw-r--r-- 1 root root 172332 2018-06-22 18:49 fw_xr829.bin
-rw-r--r-- 1 root root 295500 2018-06-22 18:49 fw_xr829_bt.bin
-rw-r--r-- 1 root root 744 2018-06-22 18:49 sdd_xr829.bin
venus-a3:/vendor/etc/firmware #

```

图 3-3 确认设备存在 fw、sdd 等文件

如果不存在这些文件，是 Android 层移植出现问题。请按照第二章的移植步骤进行检查。

4 FAQ

4.1 无法正常打开 wifi

4.1.1 确认 xradio_wlan.ko 模块正常加载

在确保已经加载 xradio_mac.ko 和 xradio_core.ko 的前提下，手动加载 xradio_wlan.ko，查看内核打印信息，看是否有异常：

```
<3>[ 1215.358745] ===== XRADIO WIFI OPEN =====
<3>[ 1215.358756] [XRADIO] Driver Label:XR_V02.16.67 HT40_01.27 Jun 22 2018 18:43:28
<3>[ 1215.358795] [XRADIO] Allocated hw_priv @ ec64cda0
<38>[ 1215.380228] type=1400 audit(1514784642.186:7): avc: denied { module_load } for pid=5505 comm="insmod" path="/v
s0 tclass=system permissive=1
<6>[ 1215.414586] sunxi-wlan wlan: bus_index: 1
<6>[ 1215.420447] sunxi-wlan wlan: check wlan wlan_power voltage: 3300000
<6>[ 1215.429013] sunxi-wlan wlan: check wlan io_regulator voltage: 3300000
<6>[ 1215.546547] sunxi-mmc sdcl: sdc set ios:clk 0Hz bm PP pm UP vdd 21 width 1 timing LEGACY(SDR12) dt B
<3>[ 1215.556810] [XRADIO] Detect SDIO card 1
<6>[ 1215.570411] sunxi-mmc sdcl: no vqmmc, Check if there is regulator
<6>[ 1215.592724] sunxi-mmc sdcl: sdc set ios:clk 400000Hz bm PP pm ON vdd 21 width 1 timing LEGACY(SDR12) dt B
<6>[ 1215.620033] sunxi-mmc sdcl: sdc set ios:clk 400000Hz bm PP pm ON vdd 21 width 1 timing LEGACY(SDR12) dt B
<6>[ 1215.633293] sunxi-mmc sdcl: sdc set ios:clk 400000Hz bm PP pm ON vdd 21 width 1 timing LEGACY(SDR12) dt B
<6>[ 1215.655365] sunxi-mmc sdcl: sdc set ios:clk 400000Hz bm PP pm ON vdd 21 width 1 timing SD-HS(SDR25) dt B
<6>[ 1215.666159] sunxi-mmc sdcl: sdc set ios:clk 500000000Hz bm PP pm ON vdd 21 width 1 timing SD-HS(SDR25) dt B
<6>[ 1215.677233] sunxi-mmc sdcl: sdc set ios:clk 500000000Hz bm PP pm ON vdd 21 width 4 timing SD-HS(SDR25) dt B
<6>[ 1215.689587] mmc0: new high speed SDIO card at address 0001
<3>[ 1215.698934] [SBUS] XRadio Device:sdio clk=500000000
<3>[ 1215.706190] [XRADIO] XRADIO_HW_REV 1.0 detected.
<3>[ 1215.756596] [XRADIO] xradio_update_dpllctrl: DPLL_CTRL Sync=0x01400000.
<3>[ 1215.797020] [XRADIO] Bootloader complete
<3>[ 1215.893404] [XRADIO] Firmware completed.
<3>[ 1215.899259] [WSM] Firmware Label:XR_C09.08.52.52_DBG_02.36 HT40 May 24 2018 10:44:52
<3>[ 1215.916116] [XRADIO] Firmware Startup Done.
<3>[ 1215.921138] [XRADIO WRN] enable Multi-Rx!
<7>[ 1215.930887] ieee80211 phy0: Selected rate control algorithm 'minstrel_ht'
<5>[ 1217.647257] CPU2: shutdown
```

图 4-1 正常加载 xradio_wlan.ko 的内核打印

如有异常，请结合出错信息进行排查。

4.1.2 确认使用匹配的 SDD 文件

确认 wifi 芯片使用的晶振频率，并确认 android/hardware/xradio/wlan/kernel-firmware/中使用的 sdd 文件是否与之匹配。如平板或者盒子出现打开 wifi 后一直处于等待扫描结果，可能是 sdd 文件与晶振不匹配导致。可以使用提供的 **sddedit.exe** 工具打开 sdd 文件，查看或修改时钟配置：

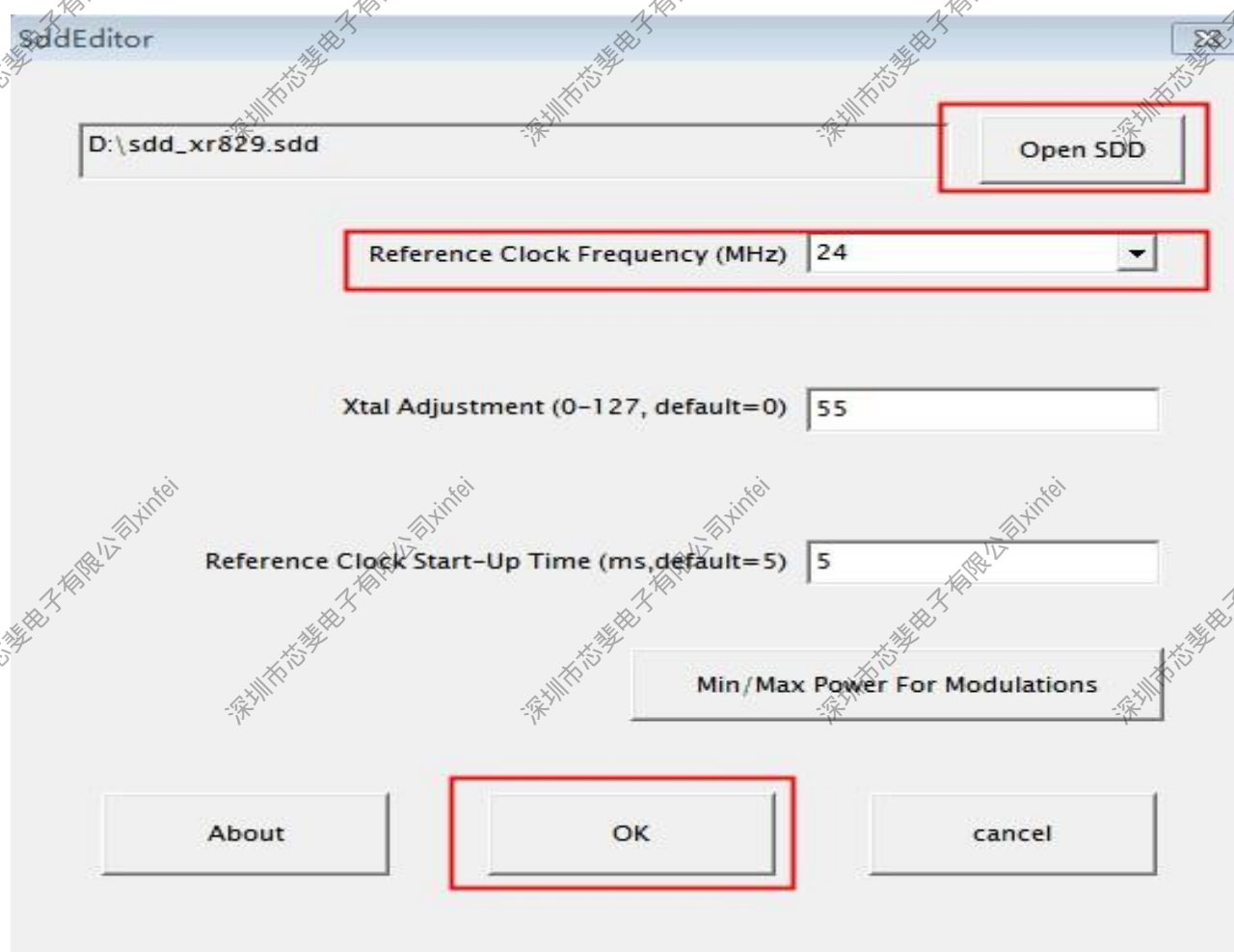


图 4-2 使用 sddedit 工具查看/修改 sdd 文件

4.1.3 确认 cfg80211 已经编译进 linux 内核 (lichee)

部分 SDK 中没有默认将 cfg80211 编进内核，导致在移植完成后不能正常使用 wifi 功能。如出现通过平板或盒子开关 wifi，driver 可以正常加载，但是 wifi 没有正常开启，请进行以下操作：

在 Linux 的内核路径下 (lichee/linux-4.9)，使用 `make ARCH=arm menuconfig` 命令显示配置界面，依次选中“Networking support ---> Wireless”，确认 cfg80211 是否被选上：

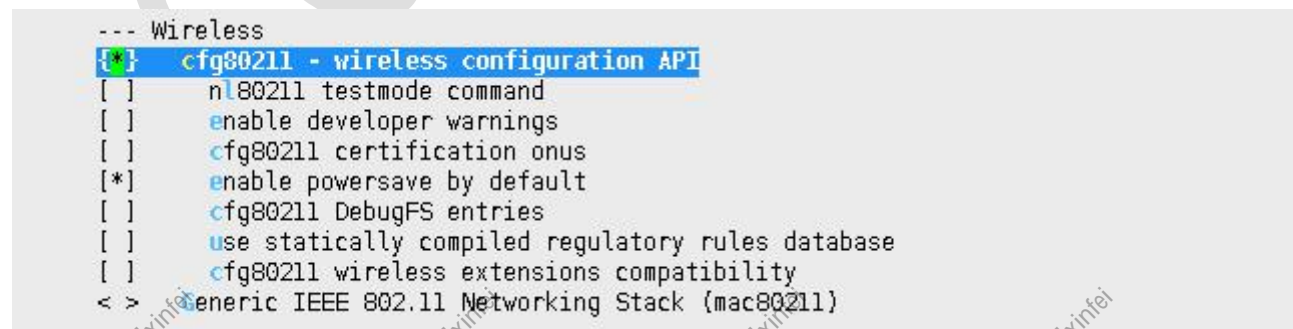


图 4-3 确认 cfg80211 已编译进内核

4.2 如何检查软件版本号

软件版本号包括驱动和固件的版本号。本移植文档只适用的 driver2.0 版本以上，请确认使用的 driver 版本。

查看软件版本的方法为：打开 wifi，输入以下命令进行查看：

```
adb shell
```

```
cd /sys/kernel/debug/ieee80211/phy*/xradio
```

```
cat version
```

```
venus-a3:/ # cd /sys/kernel/debug/ieee80211/phy*/xradio
cd /sys/kernel/debug/ieee80211/phy*/xradio
venus-a3:/sys/kernel/debug/ieee80211/phy2/xradio # cat version
cat version
Driver Label:XR_V02.16.67_HT40_01.27 Jun 22 2018 18:43:28
Firmware Label:XR_C09.08.52.52_DBG_02.36_HT40 May 24 2018 10:44:52
venus-a3:/sys/kernel/debug/ieee80211/phy2/xradio #
```

图 4-4 查看软件版本号

4.3 如何支持 wowlan 功能

在设备需要支持 wowlan 功能时，检查 android/hardware/xradio/wlan/wpa_supplicant_conf 的 wpa_supplicant_overlay.conf 文件中是否配置了 wowlan_triggers=any。在 android5.0 以上版本可以通过配置文件支持 wowlan 功能，不需要修改源代码。