



Tina Linux Wi-Fi 开发指南

版本号: 1.5
发布日期: 2022.02.15

版本历史

版本号	日期	制/修订人	内容描述
1.0	2019.03.03	AWA1381	1. 建立初始版本。
1.1	2019.11.13	AWA1381	1. 修正一些错误符号, 修改部分代码路径。
1.2	2020.05.28	AWA1381	1. 移除配网内容到配网文档。2. 添加 Tina 硬件平台 Wi-Fi 模组支持列表。3. 添加 R818/MR813 相关适配内容。
1.3	2020.06.16	AWA1381	1. 添加常用命令介绍章节。2. 添加常见问题分析章节。
1.4	2021.04.06	AWA1381	1. 添加 R528 平台相关适配内容。
1.5	2022.02.15	AWA1436	1. 添加 v853 平台相关适配内容。2. 添加 wifimanager-v2.0 相关内容

目 录

1 前言	1
1.1 文档简介	1
1.2 目标读者	1
1.3 适用范围	1
2 Wi-Fi 简介	2
2.1 Wi-Fi 工作的几种模式	2
2.2 Tina Wi-Fi 软件结构	2
2.3 Wi-Fi 常用命令介绍	3
2.3.1 station 模式	3
2.3.2 ap 模式	3
2.3.3 monitor 模式	3
2.3.4 额外功能	4
3 Wi-Fi 模组移植	5
3.1 模组移植的步骤	7
3.1.1 修改模组厂提供的 Wi-Fi driver	8
3.1.2 添加 make munconfig 的配置	9
3.1.3 配置 sys_config.fex	10
3.1.4 验证	10
3.1.5 模组移植示例	11
3.1.6 模组移植总结	12
3.2 Tina 平台已经移植的模组	13
3.2.1 BCM 系列的模组	13
3.2.2 XR 系列的模组	14
3.2.3 REALTEK 系列的模组	14
3.2.4 ESP 系列的模组	15
3.3 Tina 主要平台模组支持列表	16
4 Wi-Fi manager 介绍	17
5 Softap 介绍	18
6 常见问题	19
6.1 编译问题	19
6.1.1 找不到 wowlan 变量	19
6.1.2 找不到 xxx.ko	19
6.1.3 mmc_xxx undefined	20
6.1.4 缺少依赖库	20
6.2 驱动加载问题	20
6.2.1 R16 博通模组联网时提示: No such device	20
6.2.2 R329_XR829 模组 ifconfig 显示: No such device	21
6.2.3 MR133_XR829 can't open /etc/wifi/xr_wifi.conf, failed	21

6.2.4 驱动加载问题总结	22
6.2.4.1 配置问题	22
6.2.4.2 供电问题	22
6.2.4.3 sdio 问题	23
6.3 起 wlan0 网卡问题	23
6.3.1 R818_RTLL8723ds ifconfig wlan0 up: No such device	23
6.3.2 R328_RTLL8723ds 无法自启动 wlan0	23
6.3.3 起 wlan0 网卡问题总结	24
6.4 supplicant 服务问题	24
6.4.1 找不到 wpa_supplicant.conf 文件	24
6.5 wifimanager 使用问题	24
6.5.1 联网时出现: network not exist!	24
6.6 上层网络应用服务问题	25
6.6.1 MR133_XR829 ping 压力测试: poll time out	25

1 前言

1.1 文档简介

介绍 Allwinner 平台上 Wi-Fi 驱动移植，介绍 Tina Wi-Fi 管理框架，包括 Station，Ap 以及 Wi-Fi 常见问题。

1.2 目标读者

适用 Tina 平台的广大客户和对 Tina Wi-Fi 感兴趣的同事。

1.3 适用范围

Allwinner 软件平台 Tina v3.0 版本及以上。

Allwinner 硬件平台 R 系列 (R6, R11, R16, R18, R30, R40, R328, R331, R329, R818, R528...)。

Allwinner 硬件平台 MR 系列 (MR133, MR813...)。

Allwinner 硬件平台 H 系列 (H133...)。

Allwinner 硬件平台 V 系列 (V853...)。

2 Wi-Fi 简介

2.1 Wi-Fi 工作的几种模式

目前 Tina 平台上的 Wi-Fi 一般可处于 3 种工作模式，分别是 STATION，AP，MONITOR。

- STATION：连接无线网络的终端，大部分无线网卡默认都处于该模式，也是常用的一种模式。
- AP：无线接入点，常称热点，比如路由器功能。
- MONITOR：也称为混杂设备监听模式，所有数据包无过滤传输到主机。

2.2 Tina Wi-Fi 软件结构

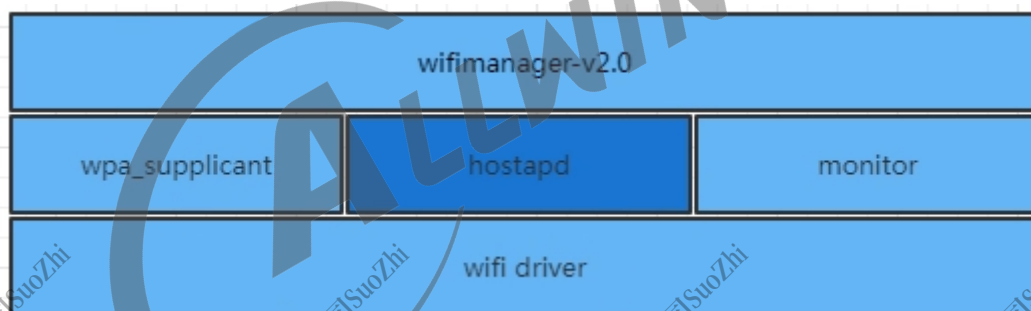


图 2-1: Tina 软件结构图

- wifimanager-v2.0：包含了 wifimanager-v1.0 的功能（用于 STATION 模式，提供 Wi-Fi 连接扫描等功能）外，还集成了 softap（启动 AP 功能）和 smartlink（多种配网模式）的功能，做到了一个应用集成了多种 wifi 功能，方便客户使用和管理。
- wpa_supplicant：开源的无线网络配置工具，主要用来支持 WEP，WPA/WPA2 和 WAPI 无线协议和加密认证的，实际上的工作是通过 socket 与驱动交互上报数据给用户。
- hostapd：是一个用户态用于 AP 和认证服务器的守护进程。
- monitor：Wi-Fi 处于混杂设备监听模式的处理应用。

2.3 Wi-Fi 常用命令介绍

2.3.1 station 模式

详情请看 Tina_linux_wifimanager2.0_ 开发指南。

执行下面的命令前请确保wifi_deamon后台进程已启动，若没有启动请先启动wifi_deamon后台进程。

wifi -o sta	以sta模式打开wifimanager
wifi -s	扫描周围网络
wifi -c ssid [passwd]	以加密或非加密的方式连接指定网络
wifi -d	断开已经连接的网络
wifi -l [all]	列出保存的网络
wifi -a [enable/disable]	重连断开的网络
wifi -r [ssid/all]	移除保存的指定网络

注：

ssid 网络名
passwd 密钥

2.3.2 ap 模式

执行下面的命令前请确保wifi_deamon后台进程已启动，若没有启动请先启动wifi_deamon后台进程。

wifi -o ap [ssid] [passwd]	以ap模式打开wifimanager
wifi -l	列出连接到ap热点的sta信息

注：

ssid 网络名
passwd 密钥
ap模式和station模式在不同模组上不一定能共存，详情看第5节介绍。

2.3.3 monitor 模式

执行下面的命令前请确保wifi_deamon后台进程已启动，若没有启动请先启动wifi_deamon后台进程。

wifi -o monitor	以monitor模式打开wifimanager
-----------------	-------------------------

注：

没有

2.3.4 额外功能

执行下面的命令前请确保wifi_deamon后台进程已启动，若没有启动请先启动wifi_deamon后台进程。

wifi -f	关闭wifimanager
wifi -p [softap/ble/xconfig/soundwave] 配网	使用softap/ble/xconfig/soundwave进行配网
wifi -D [error/warn/info/debug/dump/exce]	设置打印等级
wifi -g	获取mac地址信息
wifi -m [macaddr]	设置mac地址
wifi -h	打印wifimanager使用说明

注：
配网模式并不是所有模组都支持，要看具体的模组。
mac地址设置只能进行临时性设置。

3 Wi-Fi 模组移植

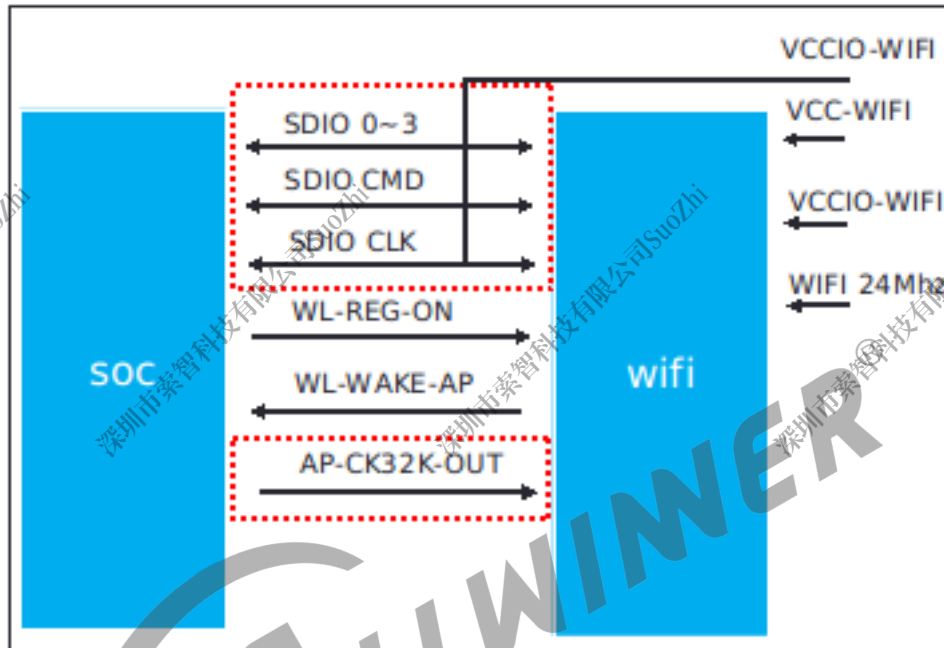


图 3-1: 主控与 Wi-Fi 硬件连接简图

Wi-Fi 模组工作的条件，如上图，需要满足以下几个条件：

- 供电：一般有两路供电，其中 VCC-Wi-Fi 为主电源，VCCIO-Wi-Fi 为 IO 上拉电源。
- 使能：要能正常工作，需要 WL-REG-ON 给高电平。
- SDIO：与 SOC 的通信有通过 USB，SDIO 等，这里以 SDIO 为例，其中 SDIO 0~3 为 SDIO 的 4 条数据线。
- 唤醒主控：当系统休眠时，Wi-Fi 模组可通过 WL-WAKE-AP 通过中断的方式唤醒主控，有些模组也通过该引脚来作为主控接收数据的中断。
- 24/26MHz 时钟信号。
- 32.768KHz 信号：根据模组而定，有些模组内部通过（5）中的输入的 clk 进行分频得到，有些需要外部单独输入该信号。

对于 Wi-Fi 模组移植，重点围绕以上的几个条件进行开展，对于以上几个工作条件 allwinner 已经提供了对应的 driver，根据总线设备驱动模型，只需要根据各个平台配置 device 即可，allwinner device 除了可以通过 dts 外 (linux-3.4 内核无 dts)，还可通过 sys_config.fex 的方式，sys_config.fex 的优先级高于 dts，一般情况下，直接配置 sys_config.fex 即可。

说明：

- Tina3.5.0及之前sys_config.fex的路径：tina/target/allwinner/xxx(cowbell_perfl)/configs/
- Tina3.5.1及之后sys_config.fex的路径：tina/device/config/chips/xxx(r328)/configs/xxx(perfl)/
- Tina3.5.0及之前dts的路径：tina/lichee/linux-xxx/arch/arm/boot/dts/
- Tina3.5.1及之后dts的路径：tina/device/config/chips/xxx(r328)/configs/xxx(perfl)/board.dts

linux 3.4 device (sys_config.fex 配置)

```
[rf_para]
module_num          = 10          /*用于区分模组型号：默认8~10*/
module_power1       = "axp22_dldo1" /*power1使用的电源树：dldo1*/
module_power1_vol    = 3300000     /*power1使用的电源树电压设置为3.3v*/
module_power2       = "axp22_dldo2" /*power2使用的电源树：dldo2*/
module_power2_vol    = 3300000     /*power2使用的电源树电压设置为3.3v*/
module_power3       = "axp22_aldo3" /*power3使用的电源树：dldo3*/
module_power3_vol    = 3300000     /*power3使用的电源树电压设置为3.3v*/

/*以上，是使用pmu供电的配置，如果直接供电，可省略*/

power_switch        =
chip_en             =             /*电源使能，备用于使用gpio高低电平打开电源*/
lpo_use_apclk       = "losc_out"   /*32khz的时钟使能*/

[wifi_para]
wifi_used           = 1           /*是否使用Wi-Fi*/
wifi_sdc_id         = 1           /*Wi-Fi所使用的sdio 卡号*/
wl_reg_on           = port:PL06<1><default><default><0> /*使能脚所使用的gpio*/
wl_host_wake        = port:PL07<4><default><default><0> /*唤醒主控所使用的gpio*/

[mmc1_para]
sdc_used            = 1
sdc_detmode         = 4
sdc_buswidth        = 4
sdc_clk             = port:PG00<2><1><1><default> /*sdio所使用的gpio号*/
sdc_cmd             = port:PG01<2><1><1><default>
sdc_d0              = port:PG02<2><1><1><default>
sdc_d1              = port:PG03<2><1><1><default>
sdc_d2              = port:PG04<2><1><1><default>
sdc_d3              = port:PG05<2><1><1><default>
sdc_det             =
sdc_use_wp          = 0
sdc_wp              =
sdc_isio            = 1
sdc_regulator       = "none"
```

linux 3.4 driver 路径，详情请参考以下代码路径

```
tina/lichee/linux-3.4/drivers/misc/rf_pm
```

linux3.4 以上，device (sys_config.fex 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>
```

```

sdci_d3                = port:PG05<2><1><3><default>
sd-uhs-sdr50           =
sd-uhs-ddr50           =
sd-uhs-sdr104          =
cap-sdio-irq           =
keep-power-in-suspend =
ignore-pm-notify       =
max-frequency          = 150000000 /*sdio所使用的最大扫卡频率*/
mix-frequency          = 150000000

[wlan]
wlan_used              = 1
compatible             = "allwinner,sunxi-wlan"
clocks                 = "losc_out"

wlan_power_num         = 2 /*使用pmu供电, 所使用的电源数量*/
wlan_power1            = "vcc-wifi1" /*使用pmu供电, 使用的power1标示*/
wlan_power2            = "vcc-wifi2" /*使用pmu供电, 使用的power2标示*/
wlan_io_regulator      = "vcc-io-wifi" /*使用pmu供电, 使用的io标示*/

/*上面4行表示使用pmu供电的配置项, 对应后面的regulator选项。如果不使用pmu供电, 可忽略。*/
wlan_busnum            = 1
wlan_regon             = port:PE06<1><1><1><0>
wlan_hostwake          = port:PE05<6><default><default><default>

[regulator0]
compatible             = "axp221s-regulator"
regulator_count        = 20
.....
regulator2             = "axp221s_dcdc2 none vdd-cpu"
regulator3             = "axp221s_dcdc3 none vdd-sys vdd-gpu"
regulator4             = "axp221s_dcdc4 none"
regulator5             = "axp221s_dcdc5 none vcc-dram"
regulator6             = "axp221s_rtc none vcc-rtc"
regulator7             = "axp221s_aldo1 none vcc-25 csi-avdd"
regulator8             = "axp221s_aldo2 none vcc-ephy0"
regulator9             = "axp221s_aldo3 none avcc vcc-pll"
regulator10            = "axp221s_dldo1 none vcc-io-wifi vcc-pg " /*io power 挂在dld01上*/
regulator11            = "axp221s_dldo2 none vcc-wifi1" /*power1 挂在dld02上*/
regulator12            = "axp221s_dldo3 none vcc-wifi2" /*power2 挂在dld03上*/
regulator13            = "axp221s_dldo4 none vdd-sata-25 vcc-pf"
regulator14            = "axp221s_eldo1 none vcc-pe csi-iovc"
.....

```

linux 3.4 以上 driver, 详情请参考以下代码路径

```
tina/lichee/linux-XXX/drivers/misc/sunxi-rf
```

3.1 模组移植的步骤

下面总结一款新模组移植到 Tina 平台的步骤。

3.1.1 修改模组厂提供的 Wi-Fi driver

模组厂提供过来的 driver，适配到 Tina 平台，主要修改的地方是调用 Tina 平台提供的有上下电，扫卡函数，修改 firmware 的 download 路径，配置 Kconfig 和 Makefile 等。

下面先说明 Tina 平台提供给 driver 的函数，其中 linux 3.4 跟其他内核稍微有些区别。

linux 3.4

```
#include <mach/sys_config.h>
#include <linux/gpio.h>

/*
*函数功能: sdio扫卡
*参数 id: 卡号, (sdio 0 or 1 ...)
*参数 insert: 0, 卡插入, 进行扫卡; 1, 卡拔出。
*返回值: 无
*/
extern void sunxi_mci_rescan_card(unsigned id, unsigned insert);

/*
*函数功能: Wi-Fi模组上电, 使能。
*参数 on: 0, 上电; 1, 掉电。
*返回值: 无
*/
extern void wifi_pm_power(int on);
```

linux 3.4 以上

```
#include <linux/mmc/host.h>
#include <linux/sunxi-gpio.h>
#include <linux/power/aw_pm.h>

/*
*函数功能: 获取所使用的sdio卡号, 对应sysconfig.fex中的wlan_busnum
*返回值: sdio 卡号
*/
extern int sunxi_wlan_get_bus_index(void);

/*
*函数功能: sdio 扫卡
*参数 id: 卡号, (sdio 0 or 1 ...)
*返回值: 无
*/
extern void sunxi_mmc_rescan_card(unsigned ids);

/*
*函数功能: Wi-Fi模组上电, 使能。
*参数 on: 0, 上电; 1, 掉电。
*返回值: 无
*/
extern void sunxi_wlan_set_power(bool on);

/*
*函数功能: 获取gpio wlan hostwake pin的申请的中断号
*参数: void
*返回值: irq number
*说明: 部分模组, 主控接收数据通过hostwake pin产生中断来触发。
```

```
/*      所以需要主控这边提供获取到中断号。
*/
extern int sunxi_wlan_get_oob_irq(void);

/*
*函数功能：获取host wake pin设置中断的标志位
*参数      : void
*返回值    : irq flag
*/
extern int sunxi_wlan_get_oob_irq_flags(void);
```

首先是将 Wi-Fi driver 放到 linux-4.9/drivers/net/wireless，填充对应的上电，扫卡等函数。

linux 3.4 的驱动请参考：

```
esp8089模组：
tina/lichee/linux-3.4/drivers/net/wireless/esp8089/sdio_stub.c

xr819模组：
tina/lichee/linux-3.4/drivers/net/wireless/xradio/wlan/platform.c
```

linux 3.4 以上的驱动请参考：

```
tina/lichee/linux-4.9/drivers/net/wireless/rtl8723ds/platform/platform_ARM_SUNnI_sdio.c
```

其次是增加内核的 menuconfig 配置以及编译，只需要修改以下地方即可。

```
tina/lichee/linux-4.9/drivers/net/wireless/Kconfig
example:
+source "drivers/net/wireless/xr829/Kconfig"
```

```
tina/lichee/linux-4.9/drivers/net/wireless/Makefile
example:
+obj-$(CONFIG_XR829_WLAN) += xr829/
```

配置完成后，可执行 make kernel_menuconfig 中选上，编译的时候，就会把指定的 driver 编译。

```
Device Drivers  --->
[*] Network device support  --->
    [*] Network device support  --->
        [*] Wireless LAN  --->
            [] xxx模块
```

3.1.2 添加 make munconfig 的配置

该步骤主要将 kernel 中编译的 ko 文件以及 firmware 拷贝到跟文件系统中。

首先是配置 firmware。firmware 文件一般以模组文件名存放在如下，并需要新增一个 mk 文件，使其在 make munconfig 中可见。

```
tina/package/firmware/linux-firmware/XXX模组
tina/package/firmware/linux-firmware/XXX模组/XXX.mk
```

```
example:
tina/package/firmware/linux-firmware/xr829

make menuconfig
  Firmware --->
    < > xr829-firmware..... Xradio xr829 firmware
```

其次是配置 ko。

```
tina/target/allwinner/xxx方案/modules.mk

example:
Tina-3.5.0及以前:
tina/target/allwinner/cowbell-perfl/modules.mk

Tina-3.5.1及以后:
tina/target/allwinner/r328s2-perfl/modules.mk

make menuconfig
  Kernel modules --->
    Wireless Drivers --->
      XXX 模块
```

3.1.3 配置 sys_config.fex

前面已经阐述，见第 3 章节开头描述。

3.1.4 验证

按照前面的配置好，make kernel menuconfig 选上对应模块，make menuconfig 选项对应 firmware 和模块，同时，make menuconfig 新增选上如下，即可进行验证了。

```
make menuconfig
Allwinner --->
  wireless --->
    <*> wifimanager-v2.0..... Tina wifimanager-v2.0
    <*> wifimanager-v2.0-demo..... Tina wifimanager-v2.0 app demo
```

验证命令

```
查看模块是否加载：lsmod
模块卸载：rmmod

ps查看wifi_deamon后台进程是否已起来，若没有起来先启动wifi_deamon后台进程
连接路由由命令：wifi -c ssid passwd
扫描周围热点：wifi -s
```


3.1.5 模组移植示例

以 RTL8723DS 为例：

1. 获取资料

1.1 建议从RTL原厂获取最新版本的完整资料，包括驱动，文档，工具。（也可以从其他内核已适配版本获取驱动）

2. 内核适配

```
2.1 将整个驱动SDK拷贝到 tina/lichee/linux-xxx/drivers/net/wireless/  
2.2 驱动重命名为 rtl8723ds  
2.3 在 tina/lichee/linux-xxx/drivers/net/wireless/ 目录修改 Kconfig 和 Makefile  
Kconfig:  
+source "drivers/net/wireless/rtl8723ds/Kconfig"  
Makefile:  
+obj-$(CONFIG_RTL8723DS) += rtl8723ds/ （注意：这里命名一定要匹配）  
2.4 修改驱动原生代码  
2.4.1 驱动的 Makefile (tina/lichee/linux-xxx/drivers/net/wireless/rtl8723ds/Makefile)  
+CONFIG_RTW_ANDROID = 0 （# CONFIG_RTW_ANDROID = 0: no Android, 4/5/6/7/8/9/10: Android  
version）  
+CONFIG_PLATFORM_I386_PC = n  
+CONFIG_PLATFORM_ARM_SUNXI = y  
2.4.2 替换适配 sunxi 的平台文件 (tina/lichee/linux-xxx/drivers/net/wireless/rtl8723ds/platform)  
可以从已经适配过的其他模组获取：platform_ARM_SUNXI_sdio.c
```

3. Tina module 适配

```
3.1. 从其他任意已经支持的IC方案中拷贝module的配置  
define KernelPackage/net-rtl8723ds  
SUBMENU:=$(WIRELESS_MENU) //make menuconfig的菜单位置，一般不更改。  
TITLE:=RTL8723DS support (staging) //make menuconfig的提示  
DEPENDS:= +r8723ds-firmware +@IPV6 +@USES_REALTEK +@PACKAGE_realtek-rftest +  
@PACKAGE_rtk_hciattach //添加tina依赖，可以理解为select  
FILES:=$(LINUX_DIR)/drivers/net/wireless/rtl8723ds/8723ds.ko  
KCONFIG:=\ //添加内核依赖可以理解为select  
...  
AUTOLoad:=$(call AutoProbe,8723ds)  
endef  
define KernelPackage/net-rtl8723ds/description //make menuconfig的描述  
Kernel modules for Realtek RTL8723DS support  
endef  
$(eval $(call KernelPackage,net-rtl8723ds))  
一个完整的module  
注：建议直接添加在平台的通用配置中：tina/target/allwinner/xxx-common/modules.mk
```

3.2. firmware 的配置
/package/firmware/linux-firmware/rtl8723ds/ //更新驱动时更新firmware文件(如果有最新的)

3.3. sys_config.fex/board.dts 的配置

```
rfkill: rfkill@0 {  
compatible = "allwinner,sunxi-rfkill";  
chip_en;  
power_en;  
status = "okay";  
  
wlan: wlan@0 {  
compatible = "allwinner,sunxi-wlan";
```

```
pinctrl-0 = <&wlan_pins_a>;
pinctrl-names = "default";
clock-names = "32k-fanout1";
clocks = <&ccu CLK_FANOUT1_OUT>;
wlan_busnum = <0x1>;
wlan_regon = <&pio PE 17 GPIO_ACTIVE_HIGH>;
wlan_hostwake = <&pio PG 10 GPIO_ACTIVE_HIGH>;
/*wlan_power = "VCC-3V3";*/
/*wlan_power_vol = <3300000>;*/
/*interrupt-parent = <&pio>;
interrupts = < PG 10 IRQ_TYPE_LEVEL_HIGH>;*/
wakeup-source;

};
...
}
```

4. 整体编译烧写

5. 验证排查

3.1.6 模组移植总结

主要就是以下几点：

- 修改模组厂提供的 driver，填充相应的上电，扫卡等函数。
- 增加 make kernel_menuconfig 和 make menuconfig 选项，涉及到 firmware, makefile, ko。
- 配置 sys_config.fex。
- 验证。

目前 Tina 平台的 linux 内核版本有 linux_3.4, linux_3.10, linux_4.4, linux_4.9, linux_5.4，由于历史原因，很有可能内核版本之间的配置有些不一样，主要体现在 device: sys_config.fex 以及 driver: sunxi-rf。用户在模组移植时，可参考对应各个内核版本进行参考。

Tina-3.5.0 及以前：

产品名称	内核版本	sys_config.fex
R16	Linux-3.4	tina/target/allwinner/astar-parrot/configs/sysconfig.fex
R18	Linux-4.4	tina/target/allwinner/tulip_d1/configs/sysconfig.fex
R328	Linux-4.9	tina/target/allwinner/cowbell_demo/configs/sysconfig.fex

Tina-3.5.1 及以后：

产品名称	内核版本	配置文件
R16	Linux-3.4	tina/device/config/chips/r16/configs/parrot/sys_config.fex
R18	Linux-4.4	tina/device/configs/chips/r18/configs/d1/sys_config.fex
R328	Linux-4.9	tina/device/config/chips/r328s2/configs/perf1/sys_config.fex(board.dts)
R528	Linux-5.4	tina/device/config/chips/r528/configs/evb1/sys_config.fex(board.dts)

3.2 Tina 平台已经移植的模组

Tina 平台上已经移植过多款 Wi-Fi 模组，支持列表如下：

```
BCM      : AP6212, AP6212A, AP6255, AP6256, AP6335等。
Realtek: RTL8723DS (linux 3.4/4.9/5.4), RTL8821cs (linux 4.9/5.4), RTL8822cs (linux 4.9),
          RTL8189FTV (linux 4.9)
Xradio  : XR819 (linux 3.4/4.4/4.9), xr829 (linux 3.4/4.4/4.9/5.4)
Esp      : esp8089 (linux 3.4)
.....
```

对于以上已经移植的模组，用户大多情况只需要在 `kernel_menuconfig` 和 `menuconfig` 选上对应的配置即可。如果按照在对应的 `menuconfig` 中选上，还是不能工作，就按照 3.1 小节的步骤依次排查原因。同时，如果有些方案可能在 `make menuconfig` 中无法显示相应的 Kernel modules，这是因为在方案下的 `modules.mk` 文件中没有添加，可按照 3.1.2 小节的方式进行添加。

以下列出各个模组 `kernel_menuconfig` 以及 `menuconfig` 的选项。

3.2.1 BCM 系列的模组

make `kernel_menuconfig` 配置

```
Device Drivers --->
Network device support --->
Wireless LAN --->
  <M> Broadcom FullMAC wireless cards support
      (/lib/firmware/fw_bcmdhd.bin) Firmware path
      (/lib/firmware/nvram.txt) NVRAM path
```

PS: BCM系列的模组，如AP6212, AP6255, AP6256..都是用的同一份driver

make `menuconfig` 配置

```
Kernel modules--->
Wireless Drivers--->
    <*> kmod-net-broadcom
Firmware--->
    <*> ap6212-firmware.          ---根据模组型号选择
```

3.2.2 XR 系列的模组

(1) XR819

make kernel_menuconfig 配置

```
Device Drivers --->
Network device support --->
Wireless LAN --->
    <M> XR819 WLAN support --->
/*or*/<M> XRadio WLAN support --->
```

make menuconfig 配置

```
Kernel modules--->
Wireless Drivers--->
    <*> kmod-net-broadcom
Firmware--->
    <*> xr819-firmware..... Xradio xr819 firmware
```

(2) XR829

make kernel_menuconfig 配置

```
Device Drivers --->
Network device support --->
Wireless LAN --->
    <M> XR829 WLAN support --->
```

make menuconfig 配置

```
Kernel modules--->
Wireless Drivers --->
    <*> kmod-net-xr829..... xr829 support (staging)
Firmware--->
    <*> xr829-firmware..... Xradio xr829 firmware
```

3.2.3 REALTEK 系列的模组

(1) RTK8723DS

make kernel_menuconfig 配置

```
Device Drivers --->
Network device support --->
Wireless LAN --->
```

<M> Realtek 8723D SDIO or SPI WiFi

make menuconfig 配置

```
Kernel modules--->
Wireless Drivers--->
  <*> kmod-net-rtl8723ds..... RTL8723DS support (staging)

Firmware--->
  <*> r8723ds-firmware..... RealTek RTL8723DS firmware
```

(2) RTK8822CS

make kernel_menuconfig 配置

```
Device Drivers --->
Network device support --->
Wireless LAN --->
  <M> Realtek 8822C SDIO WiFi
```

make menuconfig 配置

```
Kernel modules--->
Wireless Drivers--->
  <*> kmod-net-rtl8822cs..... RTL8723CS support (staging)

Firmware--->
  <*> rtl8821cs-firmware..... RealTek RTL8821CS firmware
```

(3) RTK8189FTV

make kernel_menuconfig 配置

```
Device Drivers --->
Network device support --->
Wireless LAN --->
  <M> Realtek 8189F SDIO WiFi
```

make menuconfig 配置

```
Kernel modules--->
Wireless Drivers--->
  <*> kmod-net-rtl8189fs..... RTL8189FS support (staging)
注：RTL8189FTV 在tina配置中不需要firmware配置。
```

3.2.4 ESP 系列的模组

(1) ESP8089

make kernel_menuconfig 配置

```
Device Drivers --->
Network device support --->
Wireless LAN --->
  <*> Eagle WLAN driver
```

make menuconfig 配置

```
Kernel modules--->
Wireless Drivers--->
<*> kmod-esp8089..... esp8089 support (staging)
Firmware--->
<*> esp8089-firmware..... esp8089 firmware
```

3.3 Tina 主要平台模组支持列表

硬件平台	支持模组	内核版本
R6	XR819	linux3.10
R7/R11	XR819	linux3.4
R16	AP6210/AP6212/AP6181/8723BS	linux3.4
R18	AP6212/AP6236/AP6255	linux4.4
R30	AP6181/AP6212/AP6234/AP6330/8723BS	linux4.4
R40	AP6212	linux3.10
R58	AP6212/AP6181/8723BS	linux3.4
R311	AP6212/AP6181/8723BS	linux4.9
R328	XR829/8723DS(单天线)	linux4.9
R329	XR829/8723DS(单双天线)	linux4.9
R331/R332/R333	8723DS(单天线)	linux4.9
R818	XR829/8723DS(单双天线)/8189FTV/8821CS(单双天线)	linux4.9
MR133	AP6212/AP6181/8723BS	linux4.9
MR813	XR829/8723DS(单双天线)/8189FTV	linux4.9
T7	AP6210/AP6212/AP6234	linux4.9
R528	RTL8723ds(单)/RTL8821cs/xr829	linux5.4
V853	XR829/XR819s/RTL8189FS	linux4.9

4 Wi-Fi manager 介绍

wifimanager 已升级为 wifimanager2.0，wifimanager2.0 已独立出一个单独文档，wifimanager2.0 的具体介绍请参考

Tina_linux_wifimanager2.0_ 开发指南

5 Softap 介绍

softap 功能已集成到 wifimanager2.0，wifimanager2.0 已独立出一个单独文档，wifimanager2.0 的具体介绍请参考

Tina_linux_wifimanager2.0_ 开发指南

6 常见问题

6.1 编译问题

6.1.1 找不到 wowlan 变量

1. 现象：

```
drivers/net/wireless/xr829/umac/main.c:870:17: error: 'struct wiphy' has no member named 'wowlan'
if ((hw->wiphy->wowlan->flags || hw->wiphy->wowlan->n_patterns)
```

2. 原因：

wowlan成员变量受CONFIG_PM控制，没有打开导致的。休眠唤醒的依赖。

3. 解决方案：

在内核配置

```
-Power management options --->
Device power management core functionality
```

6.1.2 找不到 xxx.ko

1. 现象：

```
sunxi_wlan_get_bus_index...xradio_core.ko undefined!
```

2. 原因：

缺少配置misc。

3. 解决方案：

在内核配置

```
m kernel_menuconfig-->
Device drivers-->
Misc devices-->
Allwinner rfkill driver
```


6.1.3 mmc_xxx undefined

1. 现象：

```
drivers/built-in.o: In function scan_device_store':
lichee/linux-4.9/drivers/misc/sunxi-rf/sunxi-wlan.c:309: undefined reference
  to sunxi_mmc_rescan_card'
lichee/linux-4.9/drivers/misc/sunxi-rf/sunxi-wlan.c:309:(.text+0x5fc40): relocation
  truncated to fit:
  R_AARCH64_CALL26 against undefined symbol `sunxi_mmc_rescan_card'
```

2. 原因：

没有配置mmc。

3. 解决方案

```
Device Drivers --->
<*> MMC/SD/SDIO card support --->
<*> Allwinner sunxi SD/MMC Host Controller support
```

6.1.4 缺少依赖库

1. 现象：

```
Package kmod-net-xr829 is missing dependencies for the following libraries:
  cfg80211.ko
  mmc_core.ko
  sunxi-wlan.ko
```

2. 原因：

依赖库需要编译进内核，不能以模块方式编译进去。

3. 解决方案。

```
在内核配置如下模块时，配置成y
CONFIG_RFKILL=y
CONFIG_CFG80211=y
CONFIG_MMC=y
CONFIG_MAC80211=y
```

6.2 驱动加载问题

6.2.1 R16 博通模组联网时提示：No such device.

1. 现象：

```
root@TinaLinux:/# wifi_add_network_test ssid passwd 1
*****
***Start wifi connect ap test!***
*****
wpa_supplicant not running!
Cannot create "/data/misc/wifi/entropy.bin": No such file or directory
Wi-Fi entropy file was not created
ifconfig: SIOCGIFFLAGS: No such device
event_label 0x0
wifi on failed!
wifi on failed event 0xf001
```

2. 原因:

```
firmware选择不匹配，导致驱动加载时下载失败。
- lsmod查看驱动已经正常加载
- dmesg 查看加载log发现:
[ 22.336336] dhdsdio_download_code_file: Open firmware file failed /lib/firmware/
fw_bcm43438a1.bin
[ 22.346331] _dhdsdio_download_firmware: dongle image file download failed
表示firmware固件缺失（这里表示缺失fw_bcm43438a1.bin）
最后发现是在配置firmware时选择了ap6212，正常应该用ap6212a
```

3. 解决方案

```
tina配置正确的firmware
firmware --->
└─> <*> ap6212a-firmware..... Broadcom AP6212A firmware
```

6.2.2 R329_XR829 模组 ifconfig 显示: No such device

1. 现象:

```
ifconfig: SIOCGIFFLAGS: No such device
```

2. 原因:

```
firmware选择不匹配。
- lsmod查看驱动已经正常加载。
- dmesg 查看加载log发现:
[ 195.966066] [XRADIO_ERR] xradio_load_firmware: Wait for wakeup:device is not responding.
XR829换了40M晶振。
```

3. 解决方案

```
tina配置选择40M晶振的firmware
firmware --->
[*] xr829 with 40M sdd
```

6.2.3 MR133_XR829 can't open /etc/wifi/xr_wifi.conf, failed

1. 现象:

lsmod驱动没有正常加载。

2. 原因：

```
- dmesg查看log:  
[ 6.802331] [XRADIO_ERR] can't open /etc/wifi/xr_wifi.conf, failed(-30)  
[ 6.802338] [XRADIO_ERR] Access_file failed, path:/etc/wifi/xr_wifi.conf!  
[ 6.914044] sunxi-mmc sdcl: no vqmmc, Check if there is regulator  
[ 7.028376] [XRADIO_ERR] xradio_load_firmware: Wait_for_wakeup: can't read control  
register.  
busnum配置错误，原理图上使用的是sdcl。
```

3. 解决方案

```
board.dts中配置wlan时  
busnum = 0;
```

6.2.4 驱动加载问题总结

6.2.4.1 配置问题

- 1.内核驱动, Tina modules, Tina firmware三者必须正确对应同一个模组。
- 2.注意common下的modules.mk的编写。
- 3.Sdio的配置一定要根据原理图选择对应busnum。

可能导致：

- 1.扫卡失败。
 - 2.下载firmware失败。
- 最终导致驱动加载失败。

6.2.4.2 供电问题

检查VCC_WIFI和VCC_IO_WIFI两路电。

不同模组对供电时序有一定要求，比如RTL8723ds需要两路电同时供电，针对有AXP的方案，一定要注意供电的配置，特别是enable的时间。

1. 硬件方面：主要排查两路电的供电方案，是否是同一路供电，若是分开供电，要考虑两路供电的时序，例如DCDC1--->VCC_WIFI, LDOA--->VCC_IO_WIFI, 那么DCDC1和LDOA的时序就得考虑。
2. 软件方面主要是sysconfig.fex或者board.dts的配置，分开供电的是否需要单独配置。

如：R818硬件设计是两路电分开供电。

可能导致：

- 1.扫卡失败。
- 2.下载firmware失败。
- 3.sdio_clk没有时钟。
- 4.32k竞争不起振。

最终导致驱动加载失败。

6.2.4.3 sdio 问题

1. sdio busnum配置错误
2. 驱动WL-REG-ON的方式不对。例如：
XR819模块出现
[SBUS_ERR] sdio probe timeout!
[XRADIO_ERR] sbus_sdio_init_failed
这个问题主要是sdio扫卡失败，跟sdio上电时序有关，可在drivers/net/wireless/xradio/wlan/platform.c中
xradio_wlan_power函数sunxi_wlan_set_power(on)后面加上一段延时。

RLT8723ds需要先高—低—高的方式。
可能导致：
1. 扫卡失败。
2. 下载firmware失败。
3. sdio_clk没有时钟。
4. 32k晶振不起振。
5. WL-REG-ON无法正常被拉高。
最终导致驱动加载失败。

6.3 起 wlan0 网卡问题

6.3.1 R818_RTL8723ds ifconfig wlan0 up: No such device

1. 现象：

ifconfig: SIOCGIFFLAGS: No such device

2. 原因：

- lsmod查看驱动已经正常加载。
- dmesg查看log未发现异常。
- 排查sdio_clk, regon_on, 32k, 都正常。
- 两路供电都正常配置。
- 对比其他平台硬件发现，供电方式不一样，两路电采用了分开供电，咨询RTL需要同时上电。

3. 解决方案

硬件更改，VCC_WIFI/VCC_IO_WIFI用同一路电供电。

6.3.2 R328_RTL8723ds 无法自启动 wlan0

1. 现象：

启动脚本/etc/init.d/wpa_supplicant中会自启动wlan0
但是每次启动启动都自启动失败，然后手动ifconfig wlan0 up正常。

2. 原因：

AP-WAKE_BT引脚被接了上拉电阻，进入测试模式了。

3. 解决方案

硬件摘除上拉电阻。

6.3.3 起 wlan0 网卡问题总结

wlan0启动失败问题目前遇到的都是与硬件相关的，如果不能自加载一般采用ifconfig wlan0 up先手动加载看看打印提示。
同时让硬件帮忙check一下供电和一些io的上下拉电阻。

6.4 supplicant 服务问题

6.4.1 找不到 wpa_supplicant.conf 文件

1. 现象：

起supplicant失败

- ps发现没有supplicant进程。
- 于是手动执行wpa_supplicant -D nl80211 -i wlan0 -c /etc/wpa_supplicant.conf -B

提示：

```
Failed to open config file '/etc/wpa_supplicant.conf', error: No such file or directory
Failed to read or parse configuration '/etc/wpa_supplicant.conf'.
```

2. 原因：

路径错误。

3. 解决方案

tina正常的路径一般在/etc/wifi/wpa_supplicant.conf
在wifimanage包下面配置正确的路径，保持和启动脚本一致。

6.5 wifimanager 使用问题

6.5.1 联网时出现：network not exist!

1. 现象：

```
wifi_connect_ap_test ssid passwd
network not exist!
```

2. 原因：

- lsmod查看驱动已经正常加载。
- ifconfig查看wlan0已经正常up。
- ps查看supplicant服务已经正常启动。

```
- 使用wifi_scan_results_test扫描网络
root@TinaLinux:/# wifi_scan_results_test
*****
***Start scan!***
*****
bssid / frequency / signal level / flags / ssid
*****
Wifi get_scan_results: Success!
*****
没有任何网络扫描到。
```

3. 解决方案

一般是信号太多，没有板载天线，尝试外加一根天线。

6.6 上层网络应用服务问题

6.6.1 MR133_XR829 ping 压力测试: poll time out

1. 现象:

ping 压力测试，一段时间后出现poll time out。

2. 原因:

ping的网络性能不好，连接的公司内网可能存在一些未知的限制。

3. 解决方案

尝试连接另外的路由器测试。

著作权声明

版权所有 © 2022 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明

、 **全志科技** （不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。