手把手带你在TinaSDK中适配RTL8723BS

推荐阅读

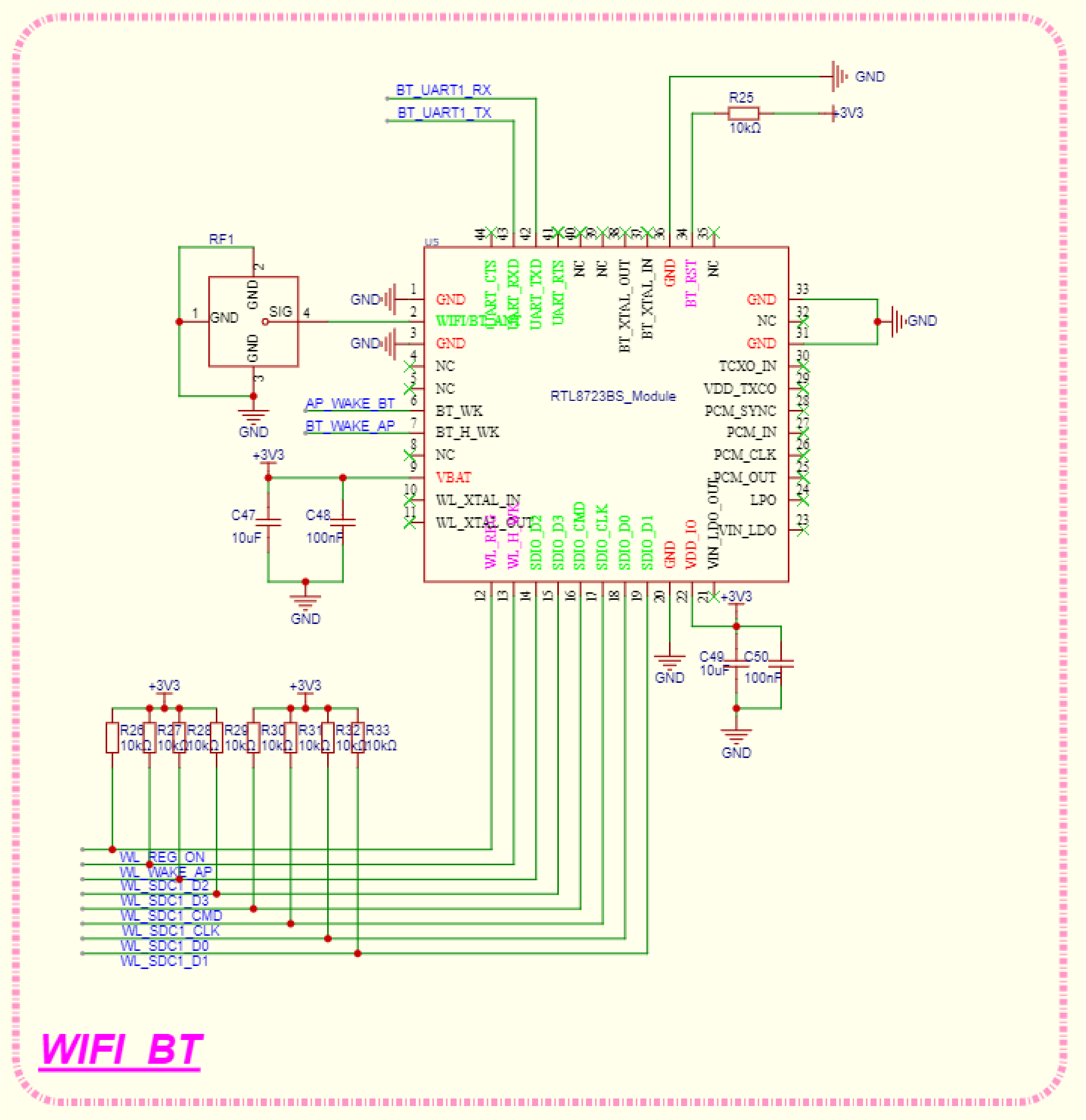
【网络专题1】Tina Wi-Fi模组移植_前导篇

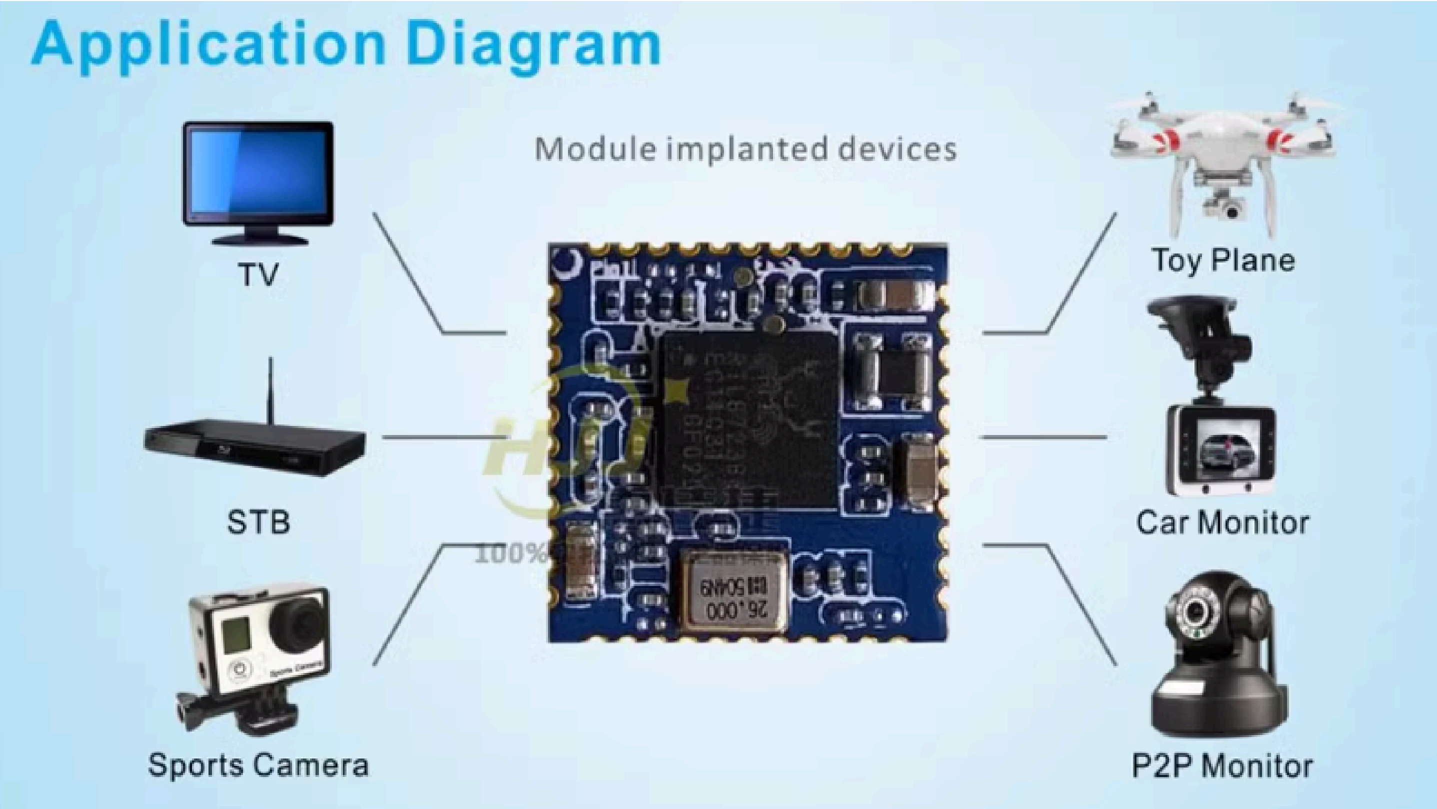
【网络专题1】Tina Wi-Fi模组移植_理论篇

【网络专题1】Tina Wi-Fi模组移植_实践篇

本教程基于我自制的全志T113-S3开发板FunnyPi，目前已在立创开源广场开源，FunnyPi-全志T113-S3卡片电脑

WIFI部分使用RTL8723BS，通过SDIO接口与T113-S3传输WIFI数据，通过UART接口传输蓝牙数据





7.1 WIFI模组的工作条件

7.1.1 供电

一版WIFI模组需要两路供电，主电源和IO电源，在上面的原理图中可以看到VBAT为主电源，VDD_IO为IO电源

7.1.2 WL_REG_ON

WL-REG-ON 信号主要用于控制 WiFi 模块的电源状态。当 WL-REG-ON 保持高电平时，WiFi 模块上电并可以工作；当 WL-REG-ON 为低电平时，WiFi 模块则处于关闭状态。在系统休眠或待机状态下，为了保持 WiFi 模块的状态，需要保持 WL-REG-ON 信号为高电平，以防止在唤醒过程中丢失 WiFi 内部状态，导致唤醒失败

7.1.3 SDIO

与SOC的通信有通过USB，SDIO 等，这里的WIFI模组使用的是SDIO

7.1.4 唤醒

WL-WAKE-AP（也称为 WL_HOST_WAKE）是一个用于唤醒主控的信号。当 WiFi 模块有数据需要处理时，它会通过这个信号唤醒主控，从而实现低功耗运行和有效数据管理。例如，在系统休眠时，如果 WiFi 模块收到数据，它会使用 WL-WAKE-AP 信号唤醒主控，处理这些数据

AP-WAKE-WL（也称为 WL_WAKE_HOST）则是另一个方向的信号，用于主控唤醒WIFI模块，当外部主机需要访问WIFI芯片时，它可以通过将WL HOST WAKE管脚拉高来唤醒芯片，使其恢复工作状态。

7.1.5 时钟

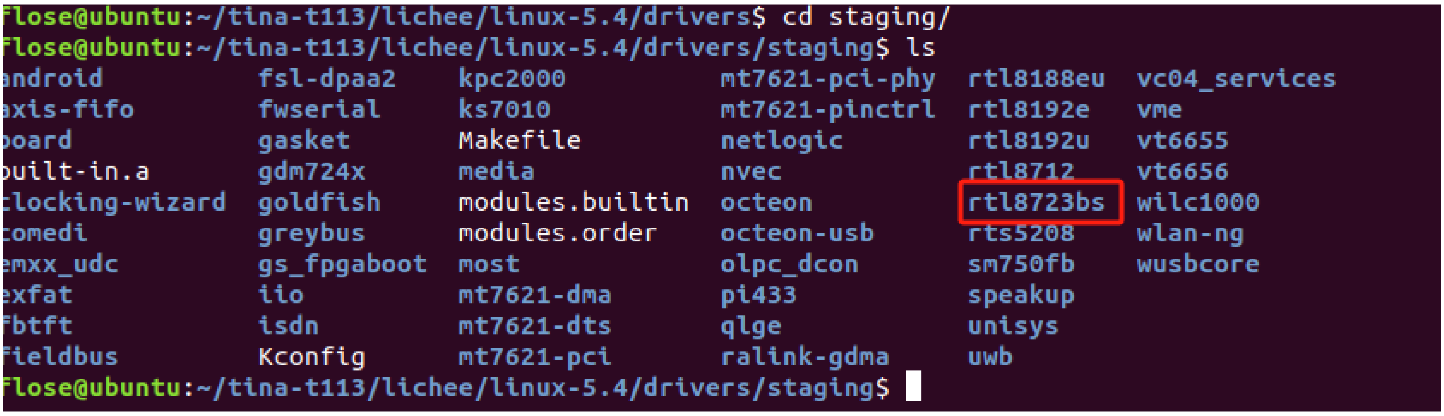
该模块通过外部晶振提供时钟源

7.2 WIFI模组的移植

因为WIFI芯片原厂驱动工程师已经编写了驱动程序，所以WIFI模组的移植可以在原厂驱动的基础上进行
allwinner device 除了可以可以通过dts 外(linux-3.4 内核无dts)，可以通过修改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

7.2.1 适配RTL8723BS

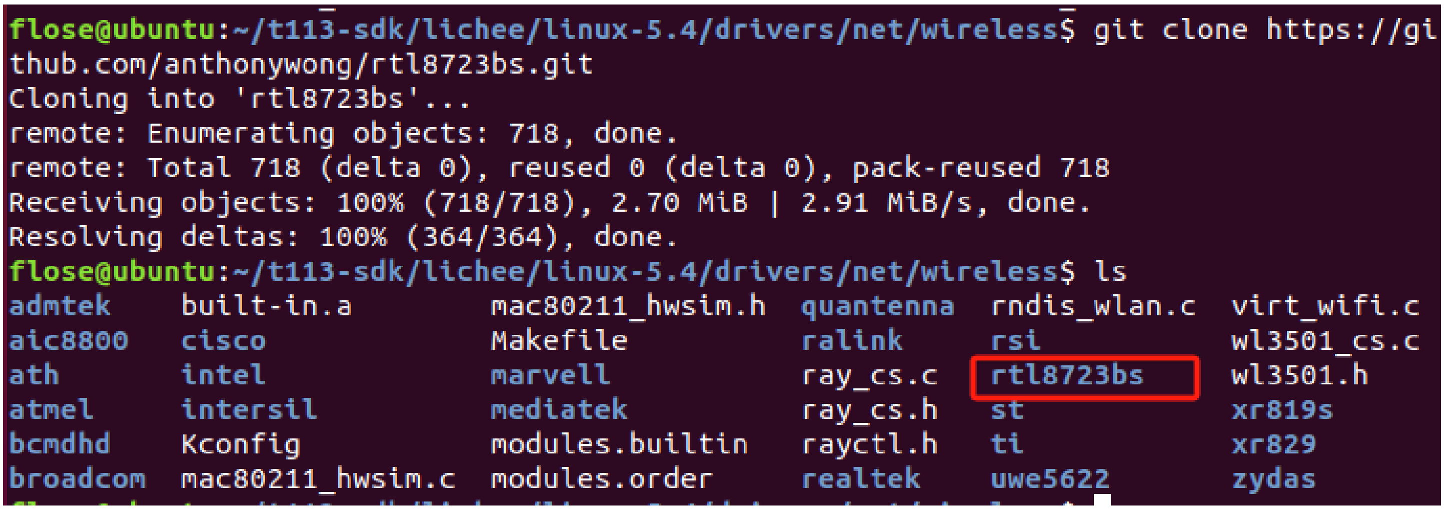
主线内核中已有rtl8723bs的驱动，但是为了将教程的通用性，我们不使用内核里面的驱动



```
flose@ubuntu:~/tina-t113/lichee/linux-5.4/drivers$ cd staging/
flose@ubuntu:~/tina-t113/lichee/linux-5.4/drivers/staging$ ls
android          fsl-dpaa2    kpc2000          mt7621-pci-phy  rtl8188eu  vc04_services
axis-fifo        fwserial     ks7010           mt7621-pinctrl  rtl8192e   vme
board            gasket       Makefile         netlogic        rtl8192u   vt6655
built-in.a       gdm724x      media            nvec            rtl8712    vt6656
clocking-wizard  goldfish     modules.builtin  octeon          rtl8723bs  wilc1000
comedi           greybus      modules.order    octeon-usb      rts5208    wlan-ng
emxx_udc         gs_fpgaboot  most             olpc_dcon       sm750fb    wusbcore
exfat            iio          mt7621-dma       pi433           speakup
fbtft            isdn         mt7621-dts       qlge            unisys
fieldbus         Kconfig      mt7621-pci       ralink-gdma     uwb
flose@ubuntu:~/tina-t113/lichee/linux-5.4/drivers/staging$
```

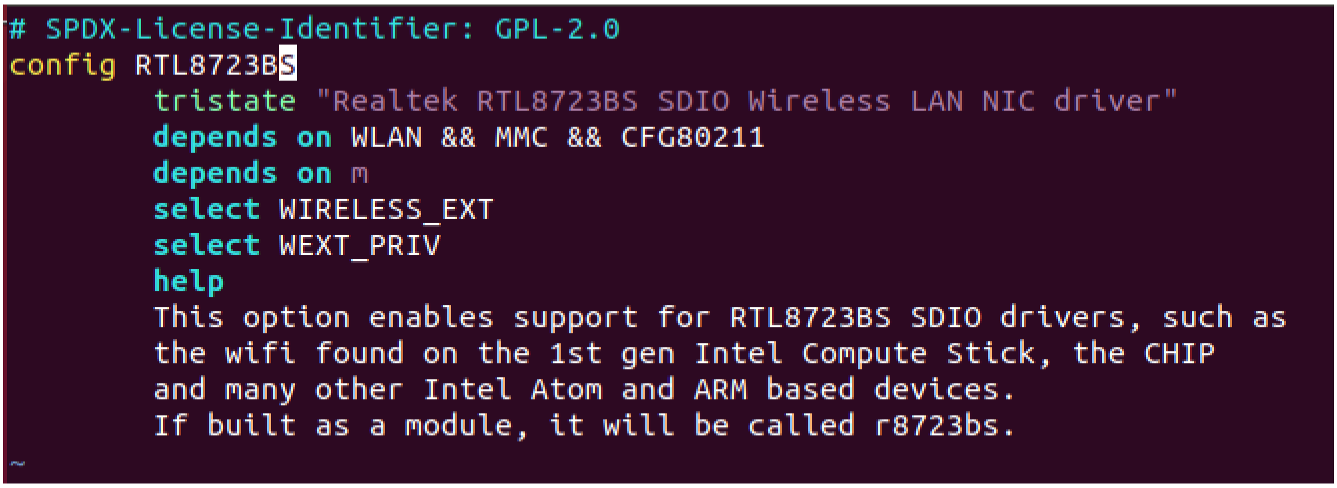
获取驱动源码，放到内核驱动路径drivers/net/wireless下。
对于现在内核源码里面已经有驱动的情况，我们直接复制内核源码里面的驱动到wireless下就行

```
cp rtl8723bs/ -rf ../net/wireless/
```



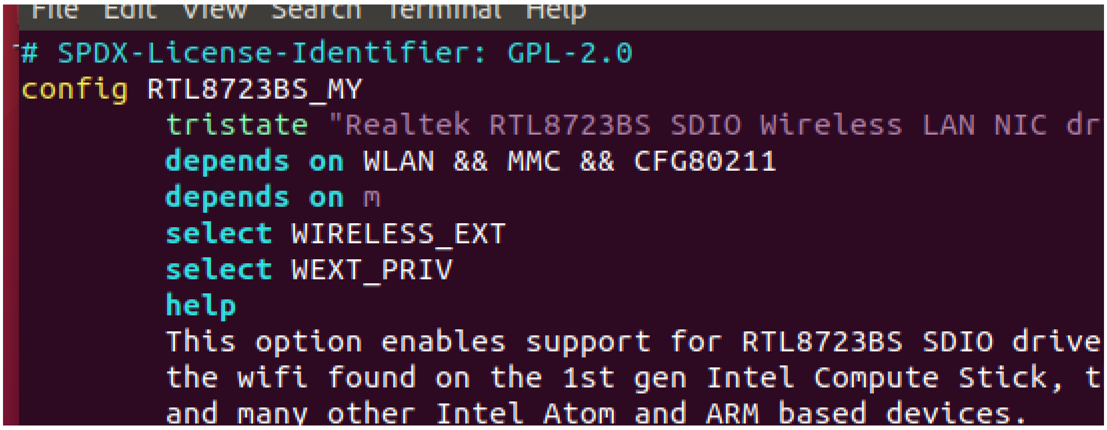
```
flose@ubuntu:~/t113-sdk/lichee/linux-5.4/drivers/net/wireless$ git clone https://gi
thub.com/anthonywong/rtl8723bs.git
Cloning into 'rtl8723bs'...
remote: Enumerating objects: 718, done.
remote: Total 718 (delta 0), reused 0 (delta 0), pack-reused 718
Receiving objects: 100% (718/718), 2.70 MiB | 2.91 MiB/s, done.
Resolving deltas: 100% (364/364), done.
flose@ubuntu:~/t113-sdk/lichee/linux-5.4/drivers/net/wireless$ ls
admtek    built-in.a        mac80211_hwsim.h  quantenna  rndis_wlan.c  virt_wifi.c
aic8800   cisco             Makefile          ralink     rsi           wl3501_cs.c
ath       intel             marvell           ray_cs.c   rtl8723bs     wl3501.h
atmel     intersil          mediatek          ray_cs.h   st            xr819s
bcmdhd    Kconfig           modules.builtin   rayctl.h   ti            xr829
broadcom  mac80211_hwsim.c  modules.order     realtek    uwe5622       zydas
```

修改rtl8723bs的Kconfig文件，防止和内核自带的rtl8723bs驱动冲突



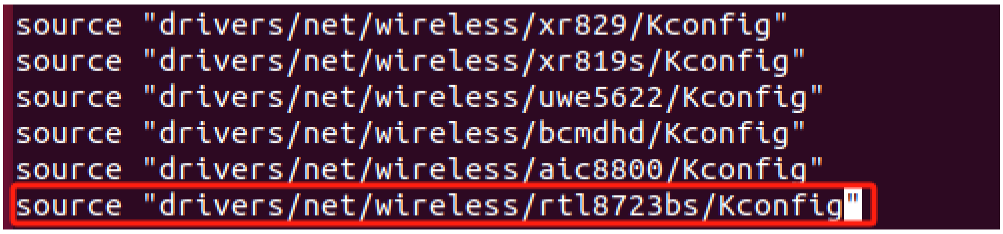
```
# SPDX-License-Identifier: GPL-2.0
config RTL8723BS
        tristate "Realtek RTL8723BS SDIO Wireless LAN NIC driver"
        depends on WLAN && MMC && CFG80211
        depends on m
        select WIRELESS_EXT
        select WEXT_PRIV
        help
        This option enables support for RTL8723BS SDIO drivers, such as
        the wifi found on the 1st gen Intel Compute Stick, the CHIP
        and many other Intel Atom and ARM based devices.
        If built as a module, it will be called r8723bs.
~
```

modify

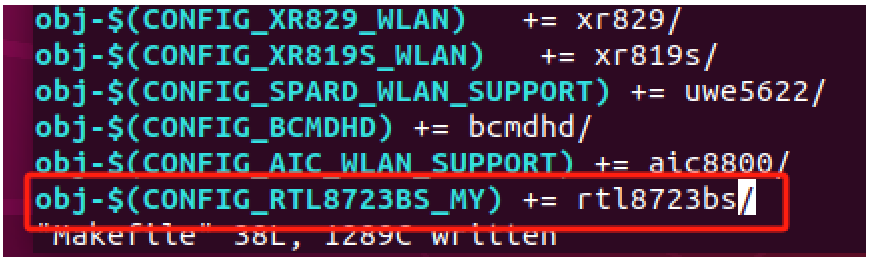


```
# SPDX-License-Identifier: GPL-2.0
config RTL8723BS_MY
        tristate "Realtek RTL8723BS SDIO Wireless LAN NIC dr
        depends on WLAN && MMC && CFG80211
        depends on m
        select WIRELESS_EXT
        select WEXT_PRIV
        help
        This option enables support for RTL8723BS SDIO drive
        the wifi found on the 1st gen Intel Compute Stick, t
        and many other Intel Atom and ARM based devices.
```

修改net/wireless目录下的Kconfig和Makefile将驱动添加到kernel_menuconfig

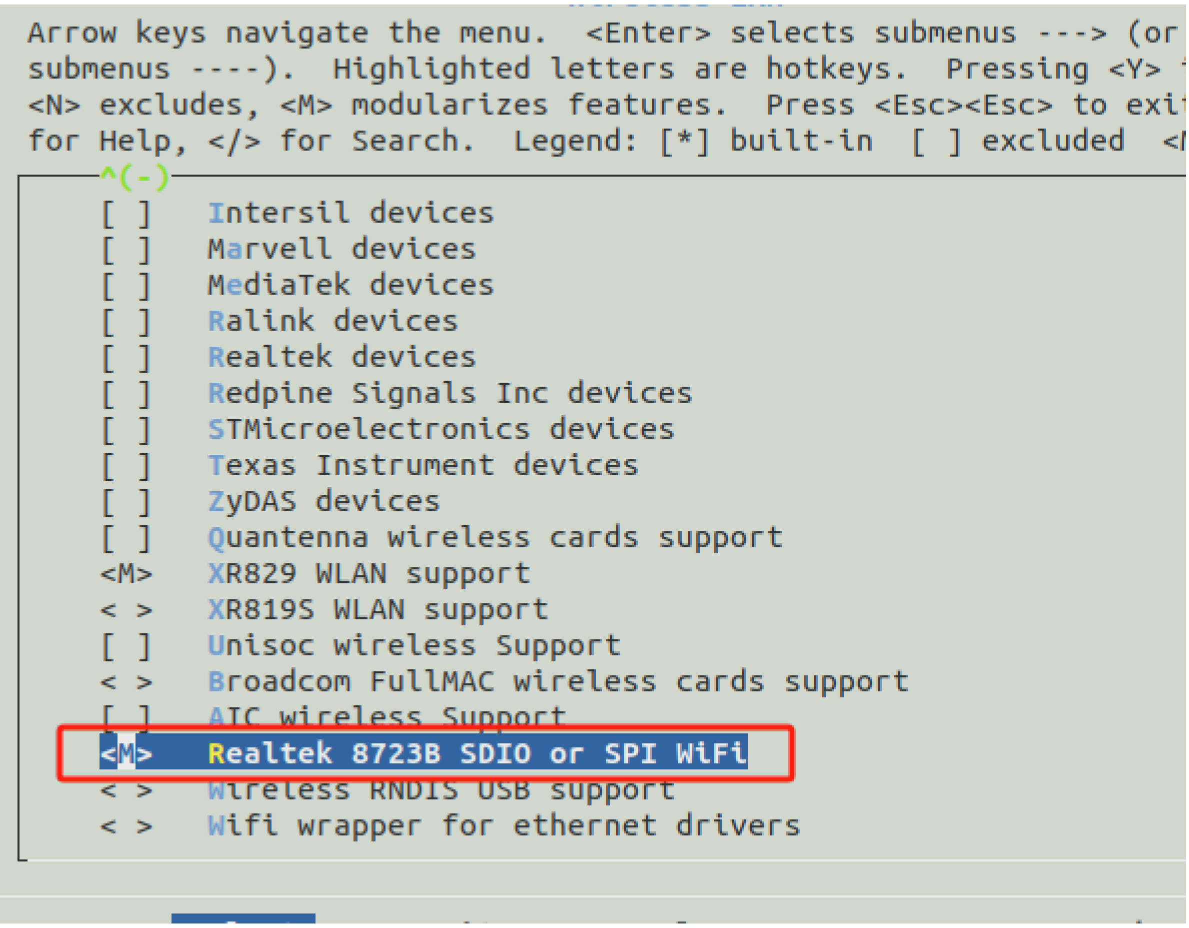


```
source "drivers/net/wireless/xr829/Kconfig"
source "drivers/net/wireless/xr819s/Kconfig"
source "drivers/net/wireless/uwe5622/Kconfig"
source "drivers/net/wireless/bcmdhd/Kconfig"
source "drivers/net/wireless/aic8800/Kconfig"
source "drivers/net/wireless/rtl8723bs/Kconfig"
```



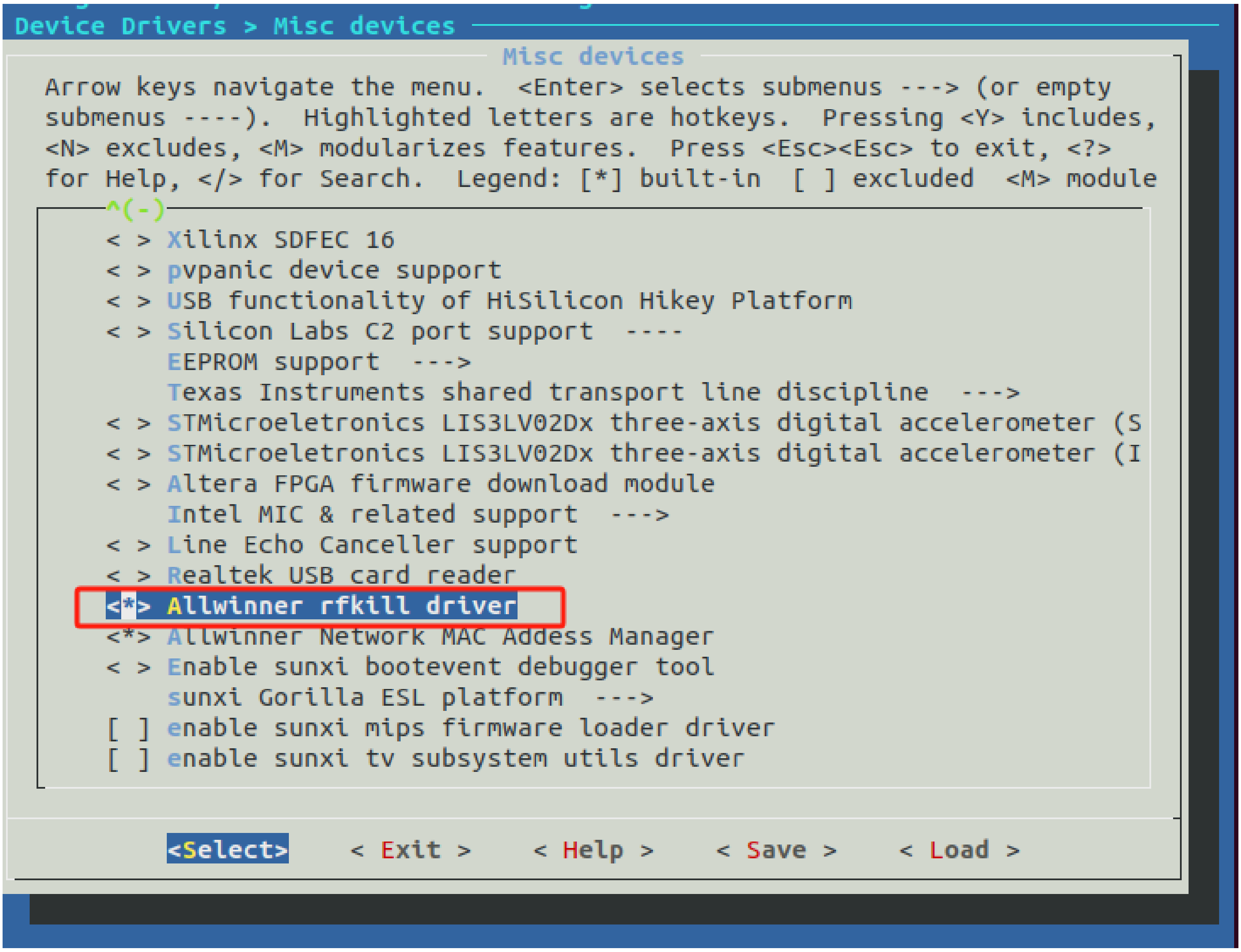
```
obj-$(CONFIG_XR829_WLAN)    += xr829/
obj-$(CONFIG_XR819S_WLAN)    += xr819s/
obj-$(CONFIG_SPARD_WLAN_SUPPORT) += uwe5622/
obj-$(CONFIG_BCMDHD) += bcmdhd/
obj-$(CONFIG_AIC_WLAN_SUPPORT) += aic8800/
obj-$(CONFIG_RTL8723BS_MY) += rtl8723bs/
"Makefile" 38L, 1289C written
```

在make kernel_menuconfig中将rtl8723bs驱动编译成module



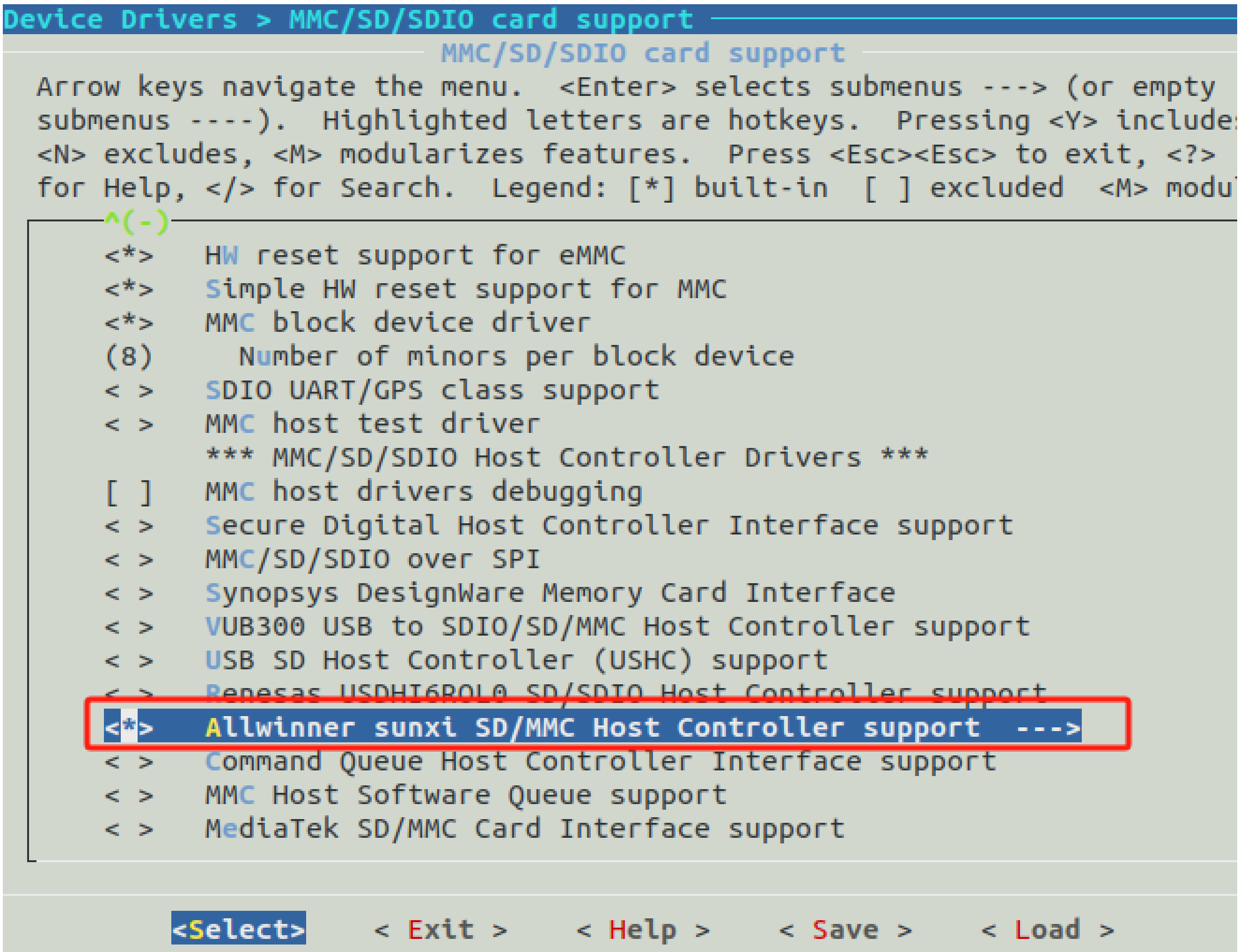
```
 Arrow keys navigate the menu.  <Enter> selects submenus ---> (or
 submenus ----).  Highlighted letters are hotkeys.  Pressing <Y> i
 <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exi
 for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <
      ^(-)
      [ ]   Intersil devices
      [ ]   Marvell devices
      [ ]   MediaTek devices
      [ ]   Ralink devices
      [ ]   Realtek devices
      [ ]   Redpine Signals Inc devices
      [ ]   STMicroelectronics devices
      [ ]   Texas Instrument devices
      [ ]   ZyDAS devices
      [ ]   Quantenna wireless cards support
      <M>   XR829 WLAN support
      < >   XR819S WLAN support
      [ ]   Unisoc wireless Support
      < >   Broadcom FullMAC wireless cards support
      [ ]   AIC wireless Support
      <M>   Realtek 8723B SDIO or SPI WiFi
      < >   Wireless RNDIS USB support
      < >   Wifi wrapper for ethernet drivers
```

打开Allwinner rfkill driver



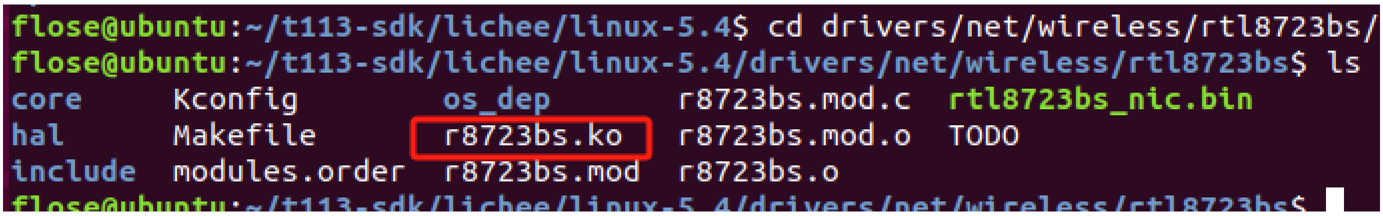
```
Device Drivers > Misc devices
                              Misc devices
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y> includes,
  <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?>
  for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module
      ^(-)
      < > Xilinx SDFEC 16
      < > pvpanic device support
      < > USB functionality of HiSilicon Hikey Platform
      < > Silicon Labs C2 port support  ----
          EEPROM support  --->
          Texas Instruments shared transport line discipline  --->
      < > STMicroeletronics LIS3LV02Dx three-axis digital accelerometer (S
      < > STMicroeletronics LIS3LV02Dx three-axis digital accelerometer (I
      < > Altera FPGA firmware download module
          Intel MIC & related support  --->
      < > Line Echo Canceller support
      < > Realtek USB card reader
      <*> Allwinner rfkill driver
      <*> Allwinner Network MAC Addess Manager
      < > Enable sunxi bootevent debugger tool
          sunxi Gorilla ESL platform  --->
      [ ] enable sunxi mips firmware loader driver
      [ ] enable sunxi tv subsystem utils driver

          <Select>    < Exit >    < Help >    < Save >    < Load >
```

打开Allwinner sunxi SD/MMC Host Controller support



```
Device Drivers > MMC/SD/SDIO card support
                          MMC/SD/SDIO card support
 Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
 submenus ----).  Highlighted letters are hotkeys.  Pressing <Y> include
 <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?>
 for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> modu
   ^(-)
     <*>   HW reset support for eMMC
     <*>   Simple HW reset support for MMC
     <*>   MMC block device driver
     (8)     Number of minors per block device
     < >   SDIO UART/GPS class support
     < >   MMC host test driver
           *** MMC/SD/SDIO Host Controller Drivers ***
     [ ]   MMC host drivers debugging
     < >   Secure Digital Host Controller Interface support
     < >   MMC/SD/SDIO over SPI
     < >   Synopsys DesignWare Memory Card Interface
     < >   VUB300 USB to SDIO/SD/MMC Host Controller support
     < >   USB SD Host Controller (USHC) support
     < >   Renesas USDHI6ROL0 SD/SDIO Host Controller support
     <*>   Allwinner sunxi SD/MMC Host Controller support  --->
     < >   Command Queue Host Controller Interface support
     < >   MMC Host Software Queue support
     < >   MediaTek SD/MMC Card Interface support

          <Select>    < Exit >    < Help >    < Save >    < Load >
```

使用mkernel命令编译内核，编译完成之后来到wireless/rtl8723bs下，可以看到已经编译出ko文件了

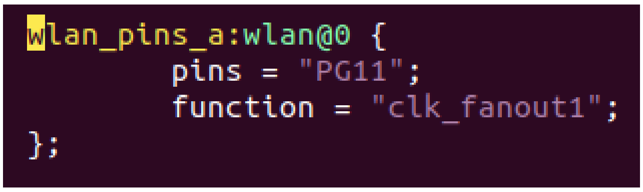


```
flose@ubuntu:~/t113-sdk/lichee/linux-5.4$ cd drivers/net/wireless/rtl8723bs/
flose@ubuntu:~/t113-sdk/lichee/linux-5.4/drivers/net/wireless/rtl8723bs$ ls
core     Kconfig        os_dep       r8723bs.mod.c  rtl8723bs_nic.bin
hal      Makefile       r8723bs.ko   r8723bs.mod.o  TODO
include  modules.order  r8723bs.mod  r8723bs.o
flose@ubuntu:~/t113-sdk/lichee/linux-5.4/drivers/net/wireless/rtl8723bs$
```

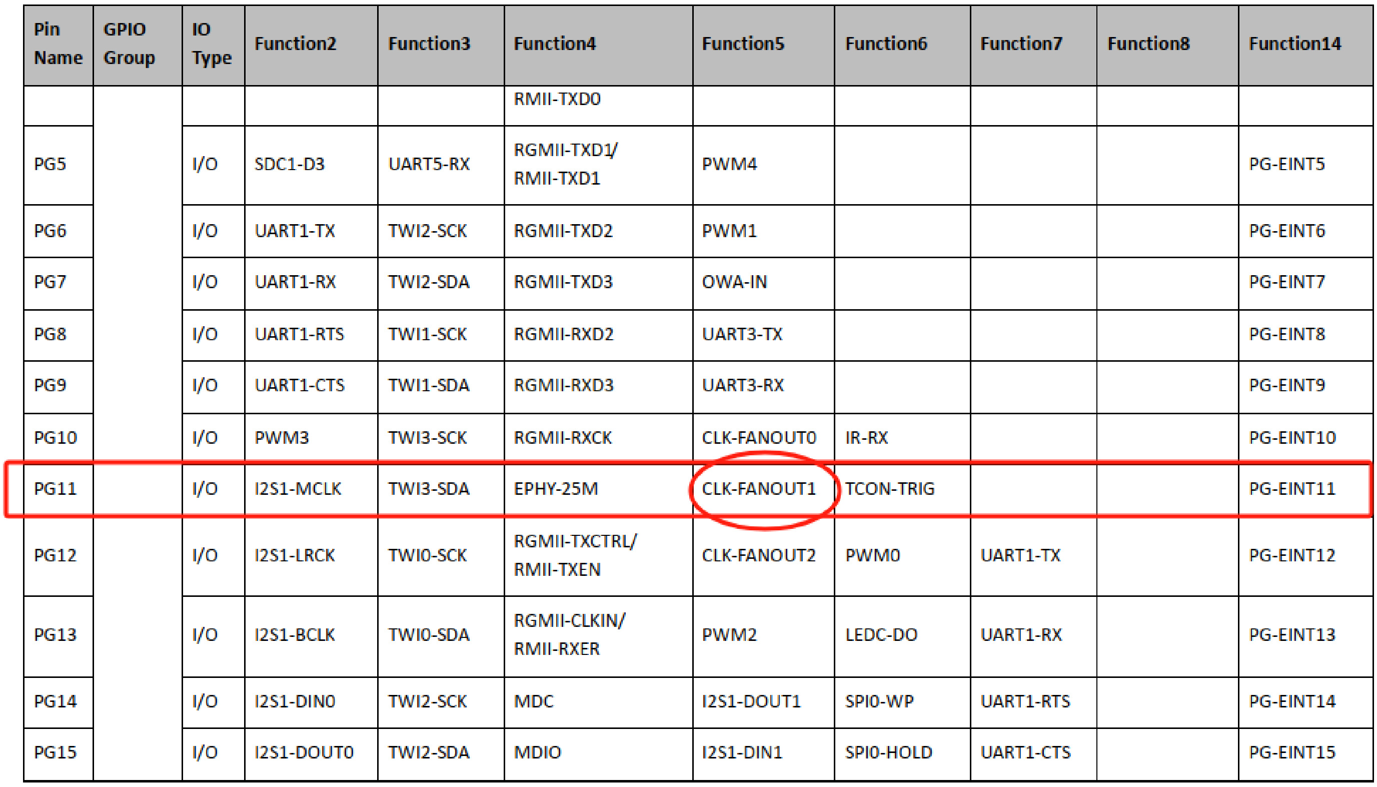
然后打开对应方案的board.dts

```
cconfigs
vim board.dts
```

我们先来分析一下设备树原本的rfkill配置

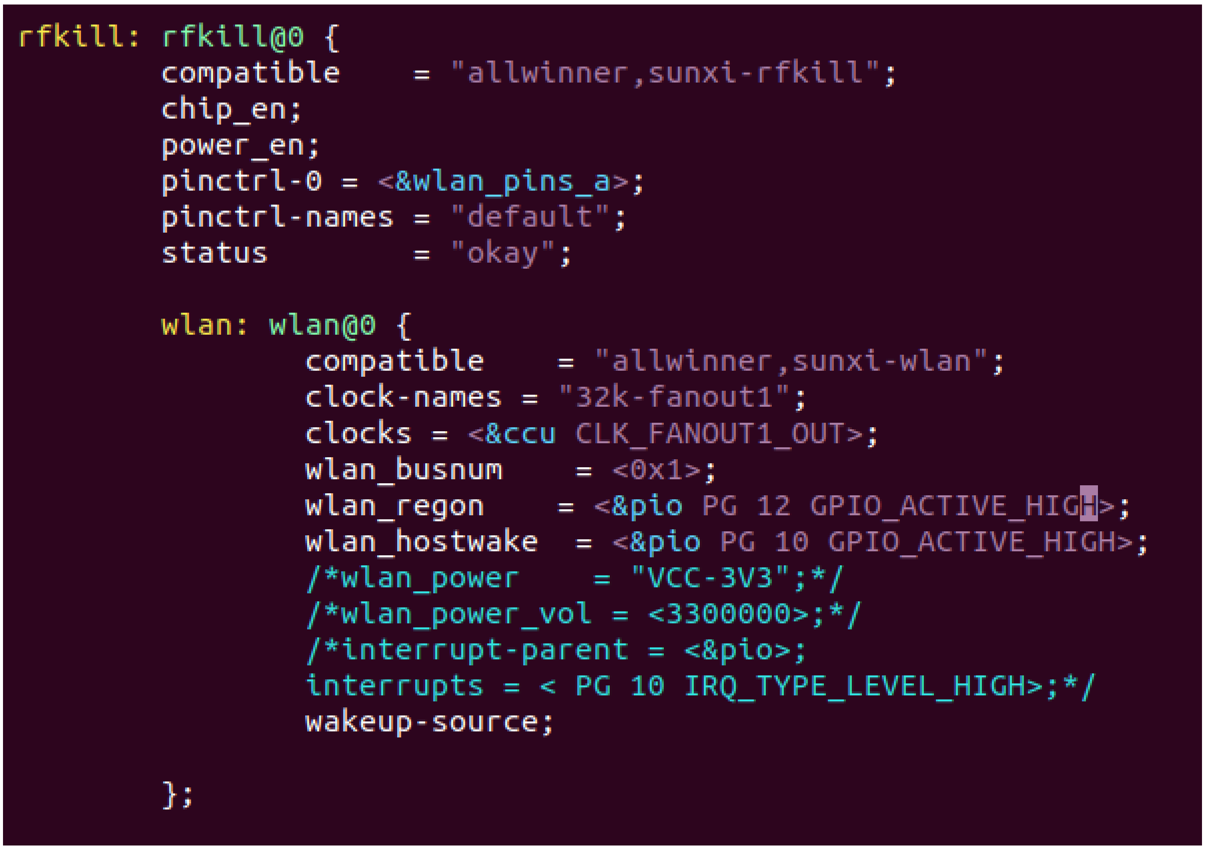


```
wlan_pins_a:wlan@0 {
        pins = "PG11";
        function = "clk_fanout1";
};
```

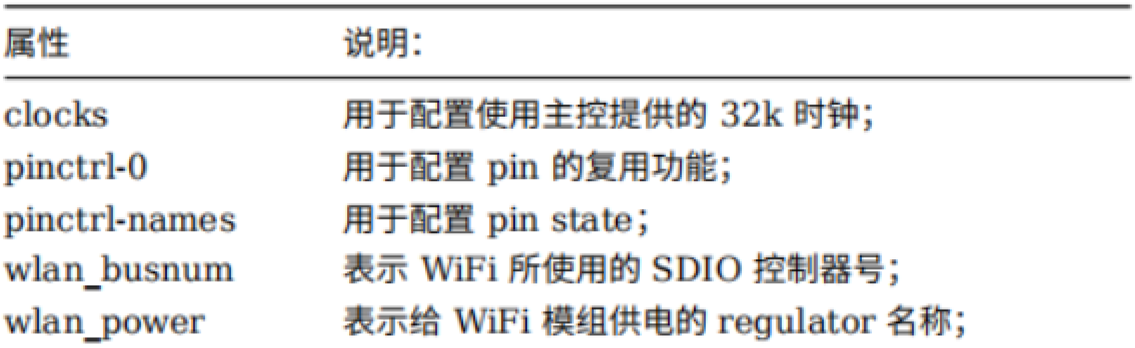


Pin Name	GPIO Group	IO Type	Function2	Function3	Function4	Function5	Function6	Function7	Function8	Function14
					RMII-TXD0					
PG5		I/O	SDC1-D3	UART5-RX	RGMII-TXD1/ RMII-TXD1	PWM4				PG-EINT5
PG6		I/O	UART1-TX	TWI2-SCK	RGMII-TXD2	PWM1				PG-EINT6
PG7		I/O	UART1-RX	TWI2-SDA	RGMII-TXD3	OWA-IN				PG-EINT7
PG8		I/O	UART1-RTS	TWI1-SCK	RGMII-RXD2	UART3-TX				PG-EINT8
PG9		I/O	UART1-CTS	TWI1-SDA	RGMII-RXD3	UART3-RX				PG-EINT9
PG10		I/O	PWM3	TWI3-SCK	RGMII-RXCK	CLK-FANOUT0	IR-RX			PG-EINT10
PG11		I/O	I2S1-MCLK	TWI3-SDA	EPHY-25M	CLK-FANOUT1	TCON-TRIG			PG-EINT11
PG12		I/O	I2S1-LRCK	TWI0-SCK	RGMII-TXCTRL/ RMII-TXEN	CLK-FANOUT2	PWM0	UART1-TX		PG-EINT12
PG13		I/O	I2S1-BCLK	TWI0-SDA	RGMII-CLKIN/ RMII-RXER	PWM2	LEDC-DO	UART1-RX		PG-EINT13
PG14		I/O	I2S1-DIN0	TWI2-SCK	MDC	I2S1-DOUT1	SPI0-WP	UART1-RTS		PG-EINT14
PG15		I/O	I2S1-DOUT0	TWI2-SDA	MDIO	I2S1-DIN1	SPI0-HOLD	UART1-CTS		PG-EINT15

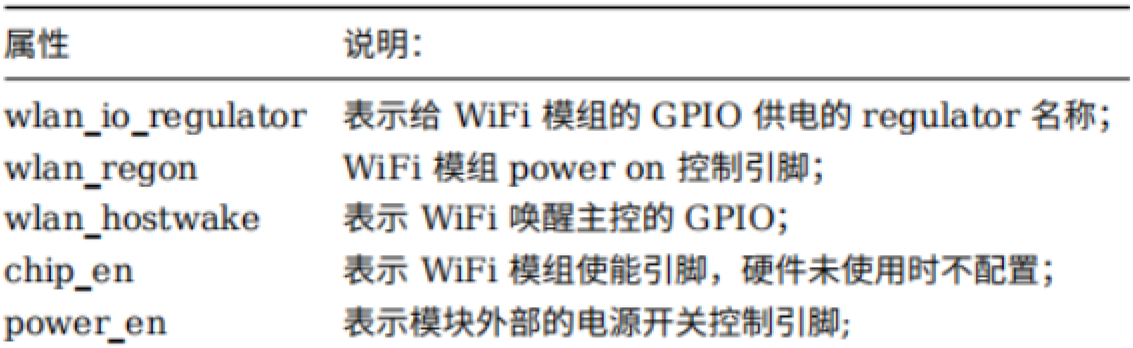
他将PG11引脚配置为CLK_FANOUT，用于给wifi芯片提供时钟信号，由于FunnyPi使用的RTL8723BS模块并没有使用主控提供的32k时钟所以这边不用管他



```
rfkill: rfkill@0 {
        compatible     = "allwinner,sunxi-rfkill";
        chip_en;
        power_en;
        pinctrl-0 = <&wlan_pins_a>;
        pinctrl-names = "default";
        status         = "okay";

        wlan: wlan@0 {
                compatible     = "allwinner,sunxi-wlan";
                clock-names = "32k-fanout1";
                clocks = <&ccu CLK_FANOUT1_OUT>;
                wlan_busnum     = <0x1>;
                wlan_regon     = <&pio PG 12 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;

        };
```



属性	说明:
clocks	用于配置使用主控提供的 32k 时钟;
pinctrl-0	用于配置 pin 的复用功能;
pinctrl-names	用于配置 pin state;
wlan_busnum	表示 WiFi 所使用的 SDIO 控制器号;
wlan_power	表示给 WiFi 模组供电的 regulator 名称;



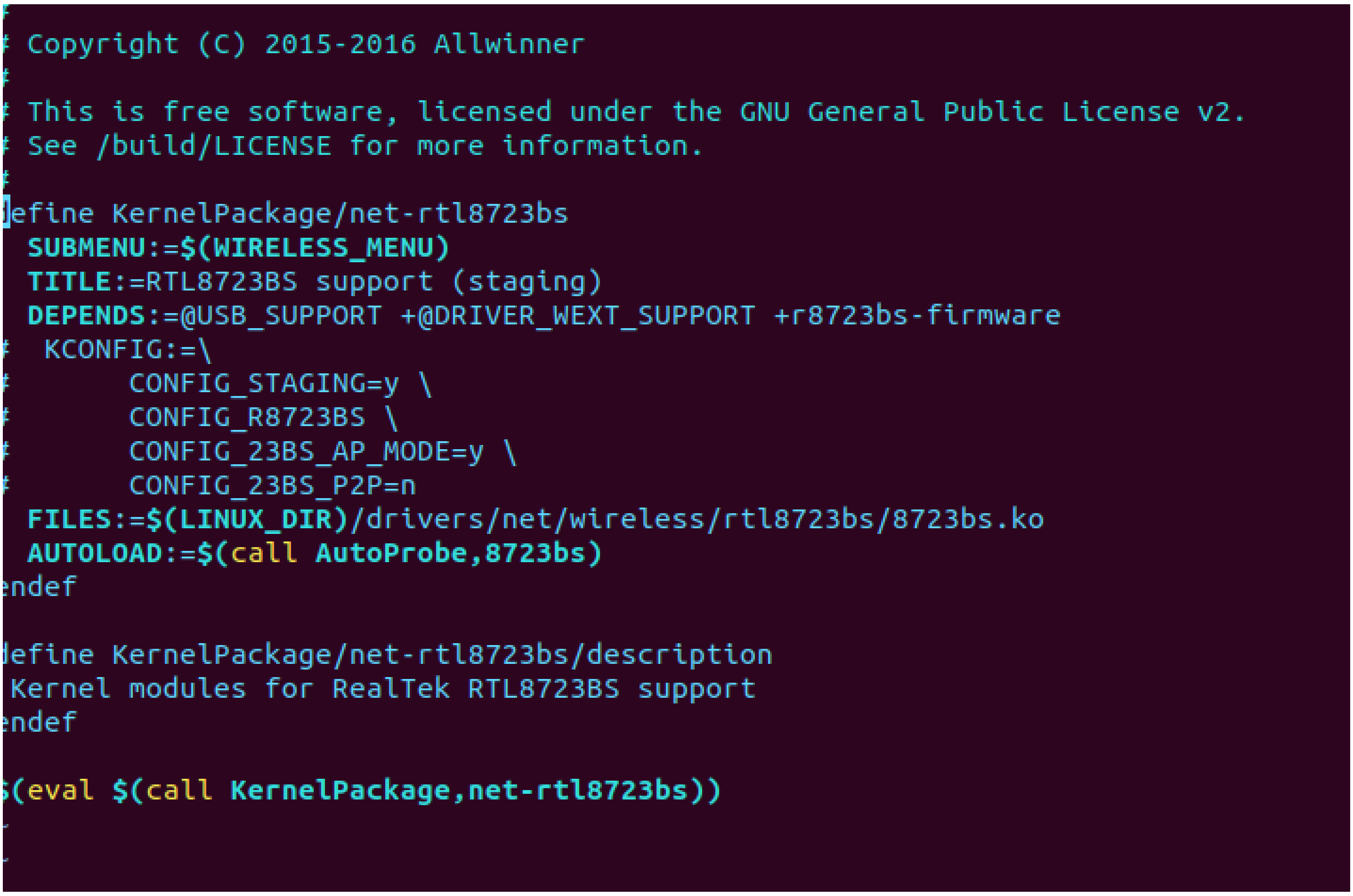
属性	说明:
wlan_io_regulator	表示给 WiFi 模组的 GPIO 供电的 regulator 名称;
wlan_regon	WiFi 模组 power on 控制引脚;
wlan_hostwake	表示 WiFi 唤醒主控的 GPIO;
chip_en	表示 WiFi 模组使能引脚，硬件未使用时不配置;
power_en	表示模块外部的电源开关控制引脚;

以上所有项必须参看原理图进行配置，配置与原理图实际使用的资源保持一致

7.2.2 方案module适配

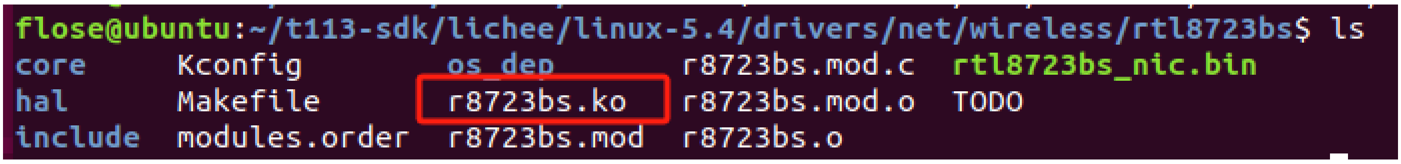
在 target/allwinner/t113-FunnyPi/modules.mk 中添加模块配置
将 t113-sdk/package/kernel/linux/modules/wireless.mk 里面关于rtl8723bs的配置复制过来

```
define KernelPackage/net-rtl8723bs-my
  SUBMENU:=$(WIRELESS_MENU)
  TITLE:=RTL8723BS support (staging)
  DEPENDS:=@USB_SUPPORT +@DRIVER_WEXT_SUPPORT +r8723bs-firmware
#  KCONFIG:=\
#       CONFIG_STAGING=y \
#       CONFIG_R8723BS \
#       CONFIG_23BS_AP_MODE=y \
#       CONFIG_23BS_P2P=n
  FILES:=$(LINUX_DIR)/drivers/net/wireless/rtl8723bs/r8723bs.ko
  AUTOLOAD:=$(call AutoProbe,r8723bs)
endef

define KernelPackage/net-rtl8723bs-my/description
 Kernel modules for RealTek RTL8723BS support
endef

$(eval $(call KernelPackage,net-rtl8723bs-my))
```



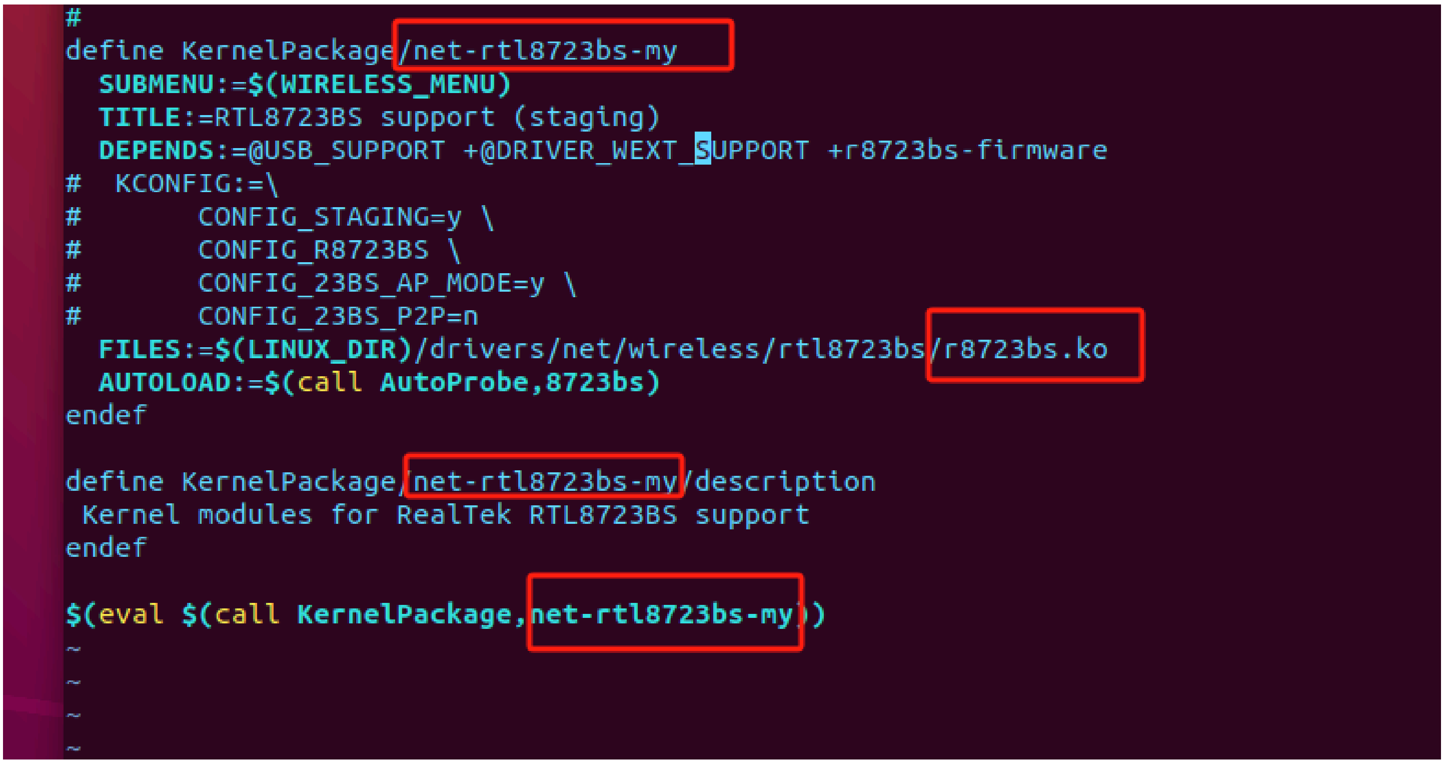
```
#
# Copyright (C) 2015-2016 Allwinner
#
# This is free software, licensed under the GNU General Public License v2.
# See /build/LICENSE for more information.
#
define KernelPackage/net-rtl8723bs
  SUBMENU:=$(WIRELESS_MENU)
  TITLE:=RTL8723BS support (staging)
  DEPENDS:=@USB_SUPPORT +@DRIVER_WEXT_SUPPORT +r8723bs-firmware
#  KCONFIG:=\
#       CONFIG_STAGING=y \
#       CONFIG_R8723BS \
#       CONFIG_23BS_AP_MODE=y \
#       CONFIG_23BS_P2P=n
  FILES:=$(LINUX_DIR)/drivers/net/wireless/rtl8723bs/8723bs.ko
  AUTOLOAD:=$(call AutoProbe,8723bs)
endef

define KernelPackage/net-rtl8723bs/description
 Kernel modules for RealTek RTL8723BS support
endef

$(eval $(call KernelPackage,net-rtl8723bs))
```

需要注意驱动文件的路径要是正确的



```
flose@ubuntu:~/t113-sdk/lichee/linux-5.4/drivers/net/wireless/rtl8723bs$ ls
core     Kconfig        os_dep       r8723bs.mod.c  rtl8723bs_nic.bin
hal      Makefile       r8723bs.ko   r8723bs.mod.o  TODO
include  modules.order  r8723bs.mod  r8723bs.o
```

我们的驱动文件前面有个r，所以要进行修改，并且为了方便区分内核自带的rtl8723bs和我们自己添加的所以我们将rtl8723bs改为rtl8723bs-my

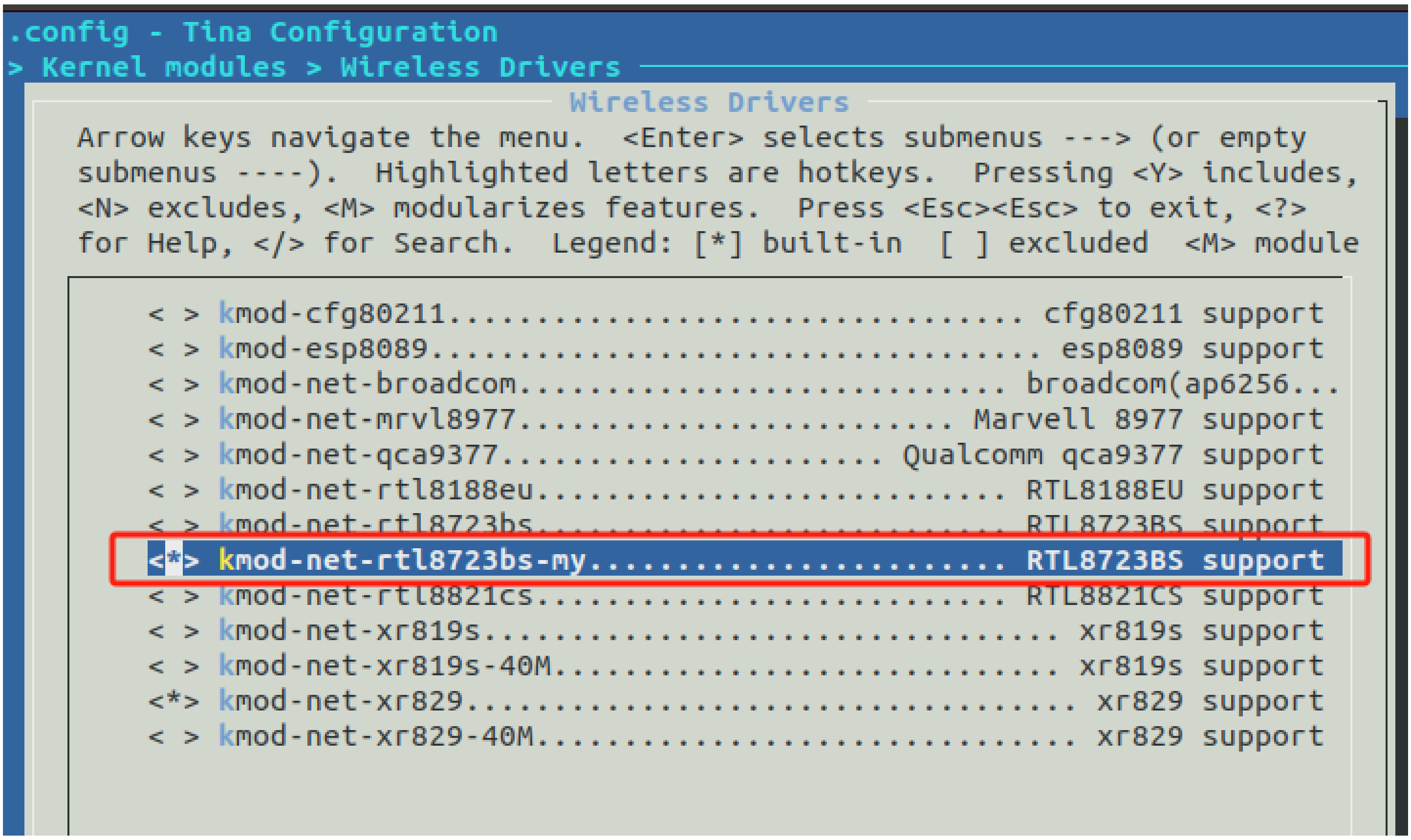


```
#
define KernelPackage/net-rtl8723bs-my
  SUBMENU:=$(WIRELESS_MENU)
  TITLE:=RTL8723BS support (staging)
  DEPENDS:=@USB_SUPPORT +@DRIVER_WEXT_SUPPORT +r8723bs-firmware
#  KCONFIG:=\
#       CONFIG_STAGING=y \
#       CONFIG_R8723BS \
#       CONFIG_23BS_AP_MODE=y \
#       CONFIG_23BS_P2P=n
  FILES:=$(LINUX_DIR)/drivers/net/wireless/rtl8723bs/r8723bs.ko
  AUTOLOAD:=$(call AutoProbe,8723bs)
endef

define KernelPackage/net-rtl8723bs-my/description
 Kernel modules for RealTek RTL8723BS support
endef

$(eval $(call KernelPackage,net-rtl8723bs-my))
~
~
~
~
```

接着在 tina 根目录执行 make menuconfig 就可以看到新添加的模组

kernel modules->wireless drivers

将他选择为*

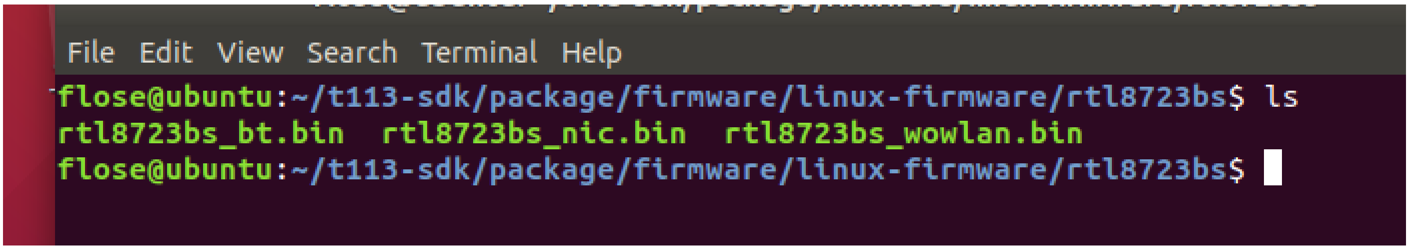


```
.config - Tina Configuration
> Kernel modules > Wireless Drivers ──────────────────────────────
                         Wireless Drivers
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y> includes,
  <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?>
  for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module

      < > kmod-cfg80211................................ cfg80211 support
      < > kmod-esp8089................................. esp8089 support
      < > kmod-net-broadcom........................... broadcom(ap6256...
      < > kmod-net-mrvl8977......................... Marvell 8977 support
      < > kmod-net-qca9377...................... Qualcomm qca9377 support
      < > kmod-net-rtl8188eu.......................... RTL8188EU support
      < > kmod-net-rtl8723bs.......................... RTL8723BS support
      <*> kmod-net-rtl8723bs-my....................... RTL8723BS support
      < > kmod-net-rtl8821cs.......................... RTL8821CS support
      < > kmod-net-xr819s................................ xr819s support
      < > kmod-net-xr819s-40M............................ xr819s support
      <*> kmod-net-xr829.................................. xr829 support
      < > kmod-net-xr829-40M.............................. xr829 support
```

7.2.3 添加Firmware

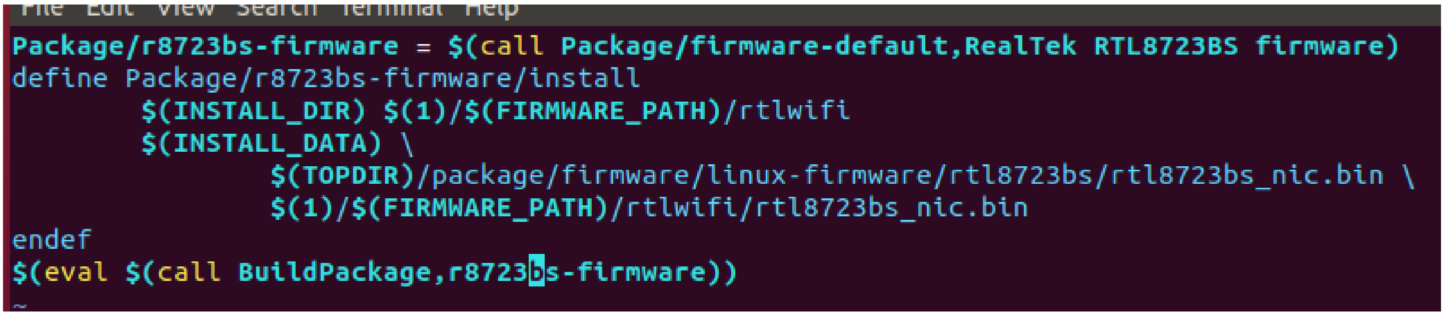
在/package/firmware/linux-firmware/rtl8723bs(需要自己创建文件夹)添加 rtl8723bs 需要的 firmware,这些文件可通过github获得

GitHub - jackeyt/RTL-8XXX-Serial-Firmware: This Repo for RTL 8XXX Serial Firmware to used by RTL 8XXX Serial's drivers~~



```
flose@ubuntu:~/t113-sdk/package/firmware/linux-firmware/rtl8723bs$ ls
rtl8723bs_bt.bin  rtl8723bs_nic.bin  rtl8723bs_wowlan.bin
flose@ubuntu:~/t113-sdk/package/firmware/linux-firmware/rtl8723bs$ 
```

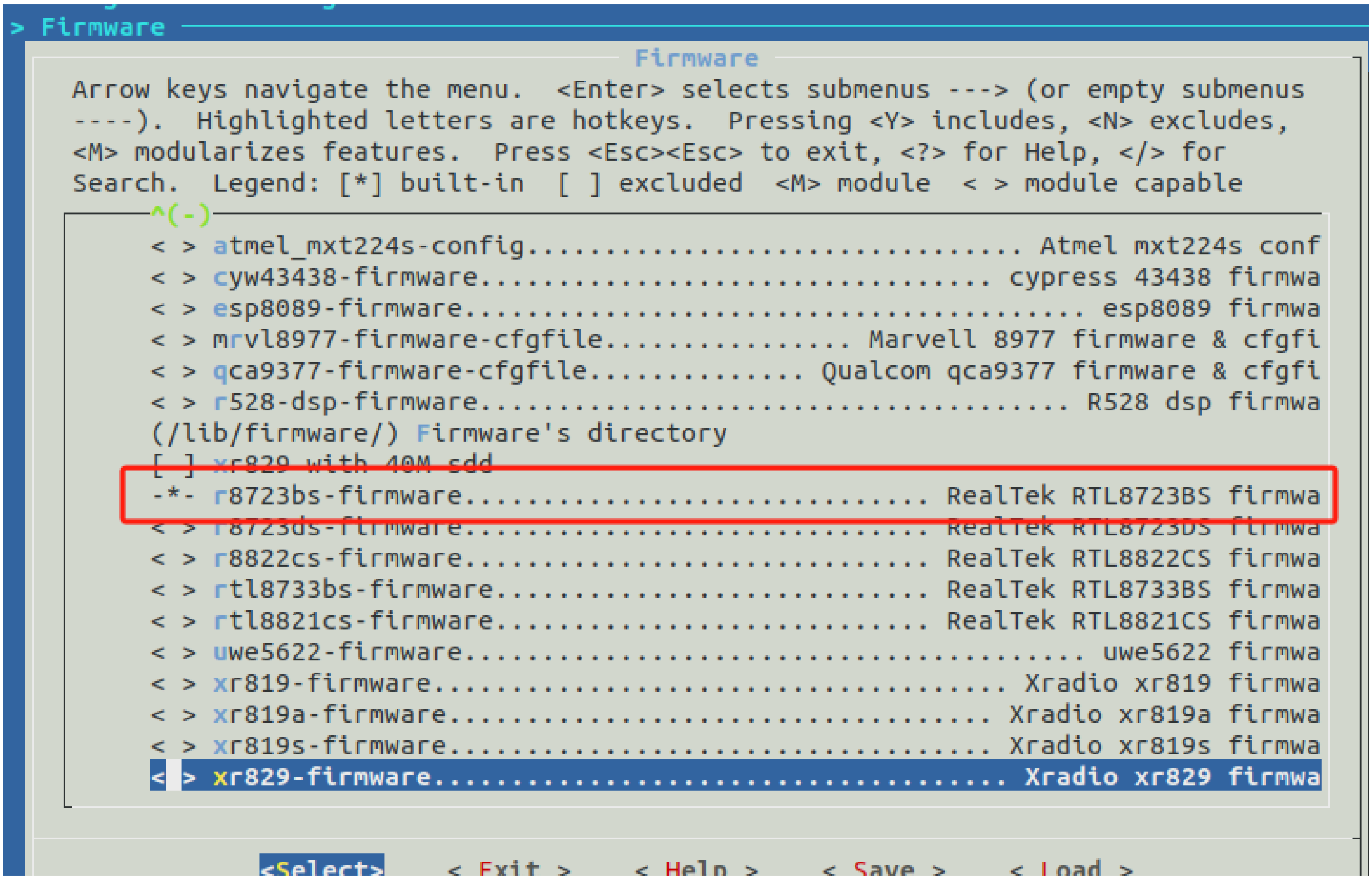
然后在rtl8723bs文件夹中创建rtl8723bs.mk文件，将rtl8723ds文件夹里面的rtl8723ds.mk复制过来，进行修改



```
Package/r8723bs-firmware = $(call Package/firmware-default,RealTek RTL8723BS firmware)
define Package/r8723bs-firmware/install
        $(INSTALL_DIR) $(1)/$(FIRMWARE_PATH)/rtlwifi
        $(INSTALL_DATA) \
                $(TOPDIR)/package/firmware/linux-firmware/rtl8723bs/rtl8723bs_nic.bin \
                $(1)/$(FIRMWARE_PATH)/rtlwifi/rtl8723bs_nic.bin
endef
$(eval $(call BuildPackage,r8723bs-firmware))
~
```

```
Package/r8723bs-firmware = $(call Package/firmware-default,RealTek RTL8723BS firmware)
define Package/r8723bs-firmware/install
        $(INSTALL_DIR) $(1)/$(FIRMWARE_PATH)/rtlwifi
        $(INSTALL_DATA) \
                $(TOPDIR)/package/firmware/linux-firmware/rtl8723bs/rtl8723bs_nic.bin \
                $(1)/$(FIRMWARE_PATH)/rtlwifi/rtl8723bs_nic.bin
endef
$(eval $(call BuildPackage,r8723bs-firmware))
```

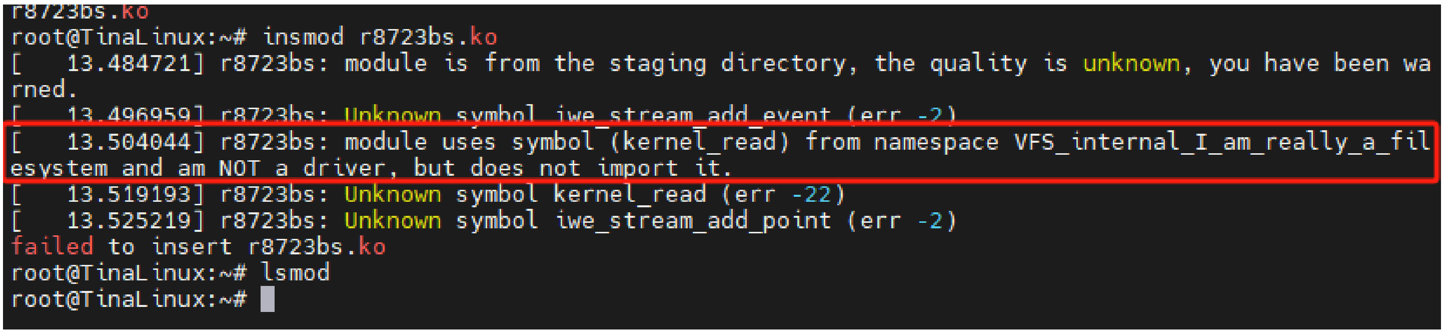
然后就可以在sdk根目录执行make menuconfig了



```
> Firmware ───────────────────────────────────────────────────────────
                              Firmware
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus
  ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,
  <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for
  Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable
      ^(-)
      < > atmel_mxt224s-config................................ Atmel mxt224s conf
      < > cyw43438-firmware................................ cypress 43438 firmwa
      < > esp8089-firmware...................................... esp8089 firmwa
      < > mrvl8977-firmware-cfgfile................ Marvell 8977 firmware & cfgfi
      < > qca9377-firmware-cfgfile.............. Qualcom qca9377 firmware & cfgfi
      < > r528-dsp-firmware...................................... R528 dsp firmwa
      (/lib/firmware/) Firmware's directory
      [ ] xr829 with 40M sdd
      -*- r8723bs-firmware............................. RealTek RTL8723BS firmwa
      < > r8723ds-firmware............................. RealTek RTL8723DS firmwa
      < > r8822cs-firmware............................. RealTek RTL8822CS firmwa
      < > rtl8733bs-firmware........................... RealTek RTL8733BS firmwa
      < > rtl8821cs-firmware........................... RealTek RTL8821CS firmwa
      < > uwe5622-firmware........................................ uwe5622 firmwa
      < > xr819-firmware..................................... Xradio xr819 firmwa
      < > xr819a-firmware.................................. Xradio xr819a firmwa
      < > xr819s-firmware.................................. Xradio xr819s firmwa
      < > xr829-firmware..................................... Xradio xr829 firmwa

              <Select>    < Exit >    < Help >    < Save >    < Load >
```

重新编译uboot和kernel并烧录系统，然后进入板卡

使用insmod命令安装驱动



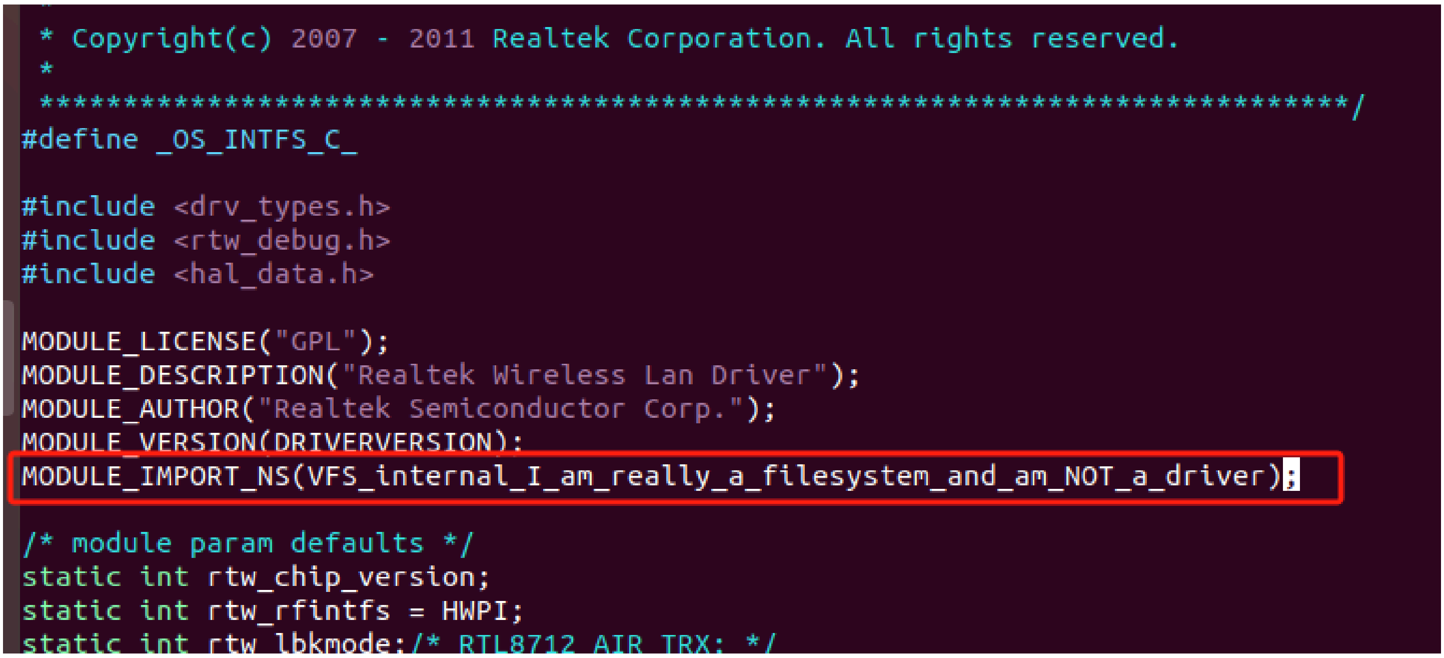
```
r8723bs.ko
root@TinaLinux:~# insmod r8723bs.ko
[   13.484721] r8723bs: module is from the staging directory, the quality is unknown, you have been wa
rned.
[   13.496959] r8723bs: Unknown symbol iwe_stream_add_event (err -2)
[   13.504044] r8723bs: module uses symbol (kernel_read) from namespace VFS_internal_I_am_really_a_fil
esystem_and_am_NOT_a_driver, but does not import it.
[   13.519193] r8723bs: Unknown symbol kernel_read (err -22)
[   13.525219] r8723bs: Unknown symbol iwe_stream_add_point (err -2)
failed to insert r8723bs.ko
root@TinaLinux:~# lsmod
root@TinaLinux:~# 
```

:::success
发现报错，出现该 WANRING 的原因是内核版本升级后文件系统存在差异，驱动调用了 kernel_read() 函数，需要进行 import VFS_internal_I_am_really_a_filesystem_and_am_NOT_a_driver 处理
:::

我们回到rtl8723bs的源码位置

在rtl8723bs/os_dep/os_intfs.c中加入

```
MODULE_IMPORT_NS(VFS_internal_I_am_really_a_filesystem_and_am_NOT_a_driver);
```

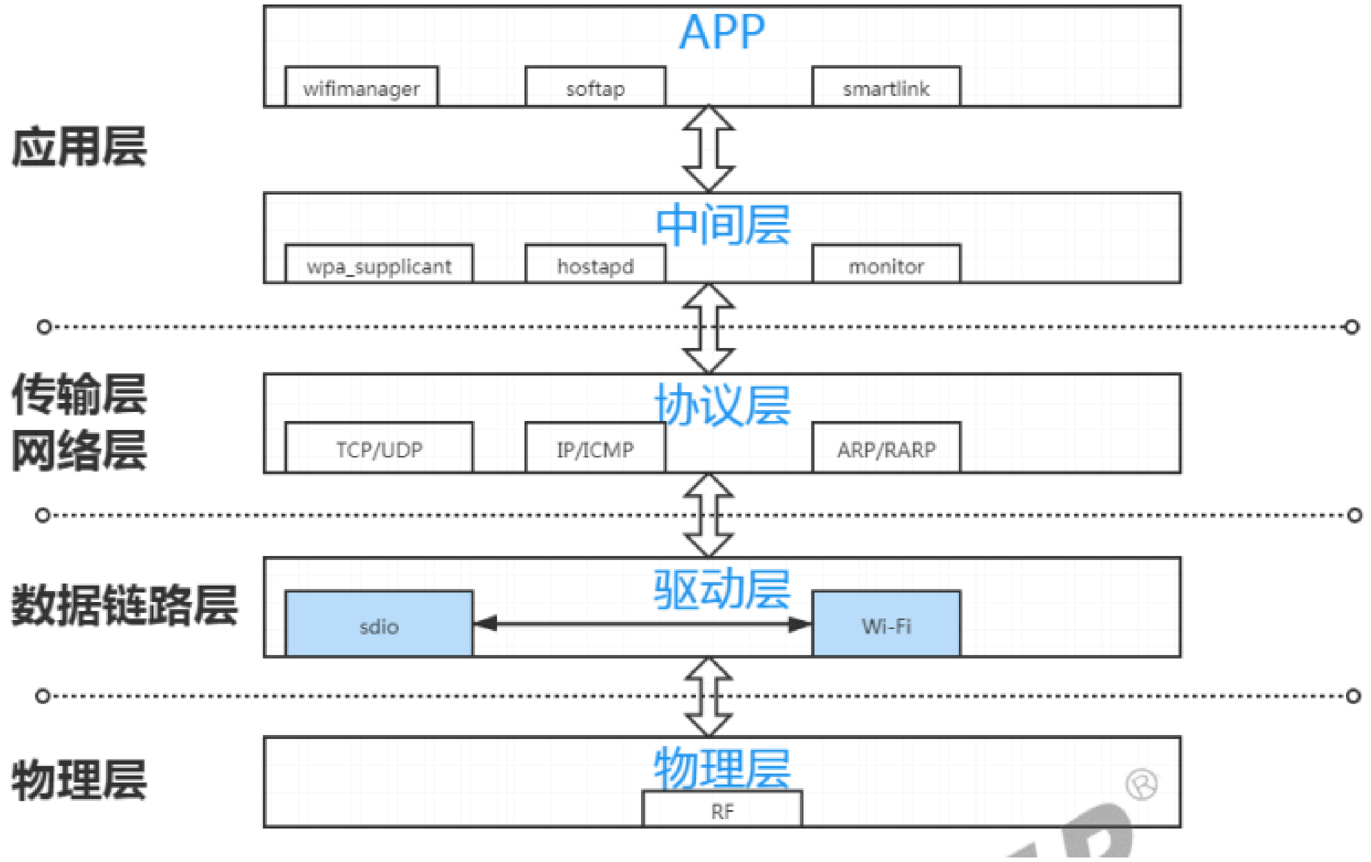


```
 * Copyright(c) 2007 - 2011 Realtek Corporation. All rights reserved.
 *
 ******************************************************************************/
#define _OS_INTFS_C_

#include <drv_types.h>
#include <rtw_debug.h>
#include <hal_data.h>

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Realtek Wireless Lan Driver");
MODULE_AUTHOR("Realtek Semiconductor Corp.");
MODULE_VERSION(DRIVERVERSION);
MODULE_IMPORT_NS(VFS_internal_I_am_really_a_filesystem_and_am_NOT_a_driver);

/* module param defaults */
static int rtw_chip_version;
static int rtw_rfintfs = HWPI;
static int rtw_lbkmode;/* RTL8712_AIR_TRX; */
```

重新编译重新加载驱动即可

7.2.4 上网配置

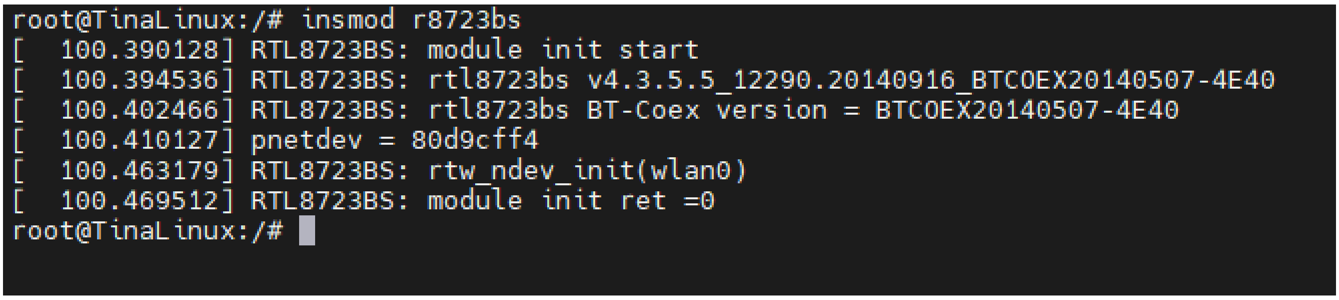
通过上面的操作我们已经成功的完成了wifi驱动的启用，接下来就是需要连接wifi并上网，这边使用的是wpa_supplicant来连接wifi



首先在menuconfig里面打开wps_supplicant，然后来到我们的开发板

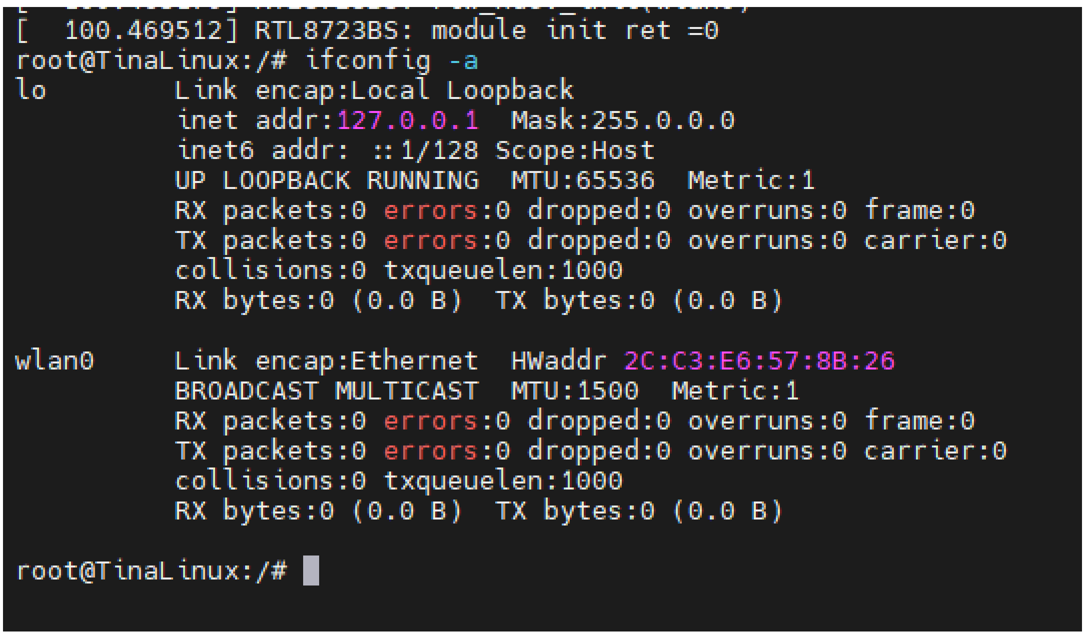
先加载驱动

```
insmod r8723bs
```



```
root@TinaLinux:/# insmod r8723bs
[  100.390128] RTL8723BS: module init start
[  100.394536] RTL8723BS: rtl8723bs v4.3.5.5_12290.20140916_BTCOEX20140507-4E40
[  100.402466] RTL8723BS: rtl8723bs BT-Coex version = BTCOEX20140507-4E40
[  100.410127] pnetdev = 80d9cff4
[  100.463179] RTL8723BS: rtw_ndev_init(wlan0)
[  100.469512] RTL8723BS: module init ret =0
root@TinaLinux:/#
```

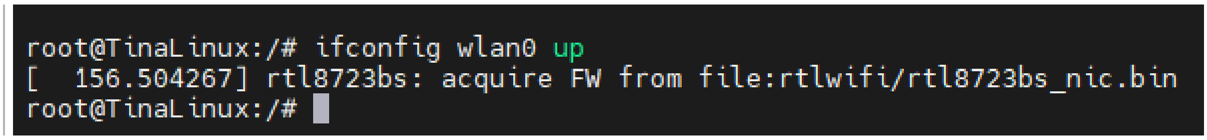
先使用ifconfig -a看一下是否有wlan0



```
[  100.469512] RTL8723BS: module init ret =0
root@TinaLinux:/# ifconfig -a
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr 2C:C3:E6:57:8B:26
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@TinaLinux:/#
```

然后开启wlan0

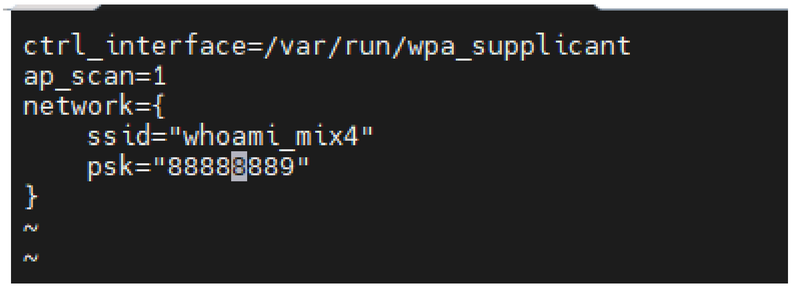
```
ifconfig wlan0 up
```



```
root@TinaLinux:/# ifconfig wlan0 up
[  156.504267] rtl8723bs: acquire FW from file:rtlwifi/rtl8723bs_nic.bin
root@TinaLinux:/#
```

修改/etc/wps_supplicant.conf文件

```
ctrl_interface=/var/run/wpa_supplicant
ap_scan=1
network={
    ssid="whoami_mix4"
    psk="88888889"
}
```



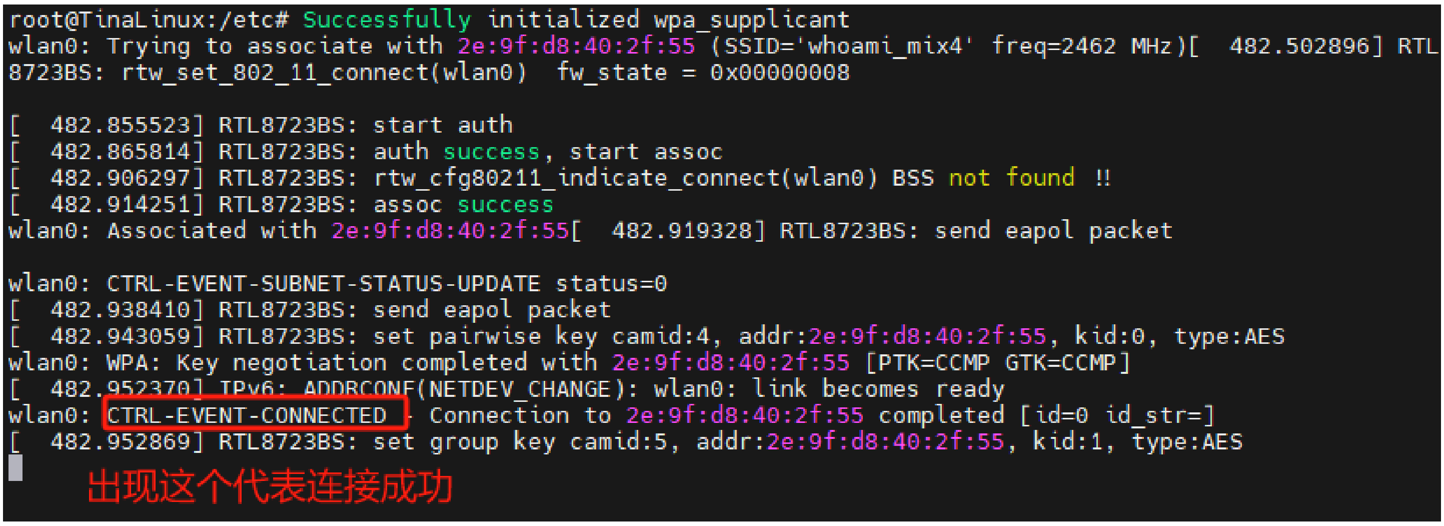
```
ctrl_interface=/var/run/wpa_supplicant
ap_scan=1
network={
        ssid="whoami_mix4"
        psk="88888889"
}
~
~
```

在/var/run 下面创建wpa_supplicant文件夹

```
mkdir -p /var/run/wpa_supplicant
```

开启wpa_supplicant连接wifi

```
wpa_supplicant -D nl80211 -c /etc/wpa_supplicant.conf -i wlan0 &
```

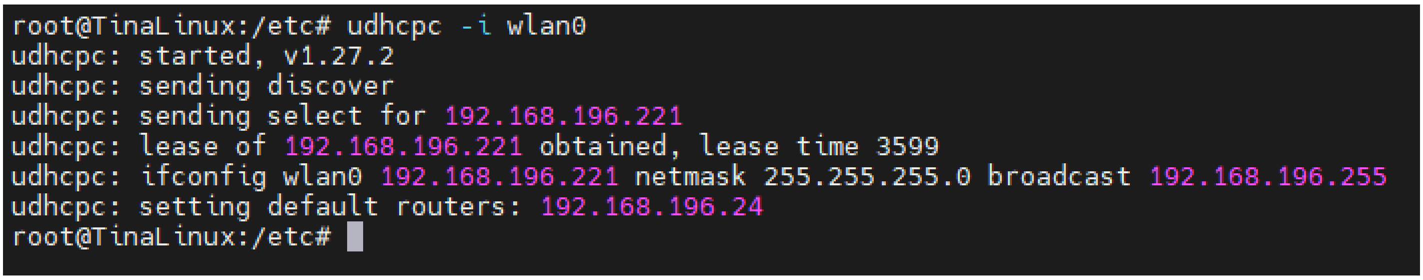


```
root@TinaLinux:/etc# Successfully initialized wpa_supplicant
wlan0: Trying to associate with 2e:9f:d8:40:2f:55 (SSID='whoami_mix4' freq=2462 MHz)[  482.502896] RTL
8723BS: rtw_set_802_11_connect(wlan0)  fw_state = 0x00000008

[  482.855523] RTL8723BS: start auth
[  482.865814] RTL8723BS: auth success, start assoc
[  482.906297] RTL8723BS: rtw_cfg80211_indicate_connect(wlan0) BSS not found !!
[  482.914251] RTL8723BS: assoc success
wlan0: Associated with 2e:9f:d8:40:2f:55[  482.919328] RTL8723BS: send eapol packet

wlan0: CTRL-EVENT-SUBNET-STATUS-UPDATE status=0
[  482.938410] RTL8723BS: send eapol packet
[  482.943059] RTL8723BS: set pairwise key camid:4, addr:2e:9f:d8:40:2f:55, kid:0, type:AES
wlan0: WPA: Key negotiation completed with 2e:9f:d8:40:2f:55 [PTK=CCMP GTK=CCMP]
[  482.952370] IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
wlan0: CTRL-EVENT-CONNECTED - Connection to 2e:9f:d8:40:2f:55 completed [id=0 id_str=]
[  482.952869] RTL8723BS: set group key camid:5, addr:2e:9f:d8:40:2f:55, kid:1, type:AES
```

出现这个代表连接成功

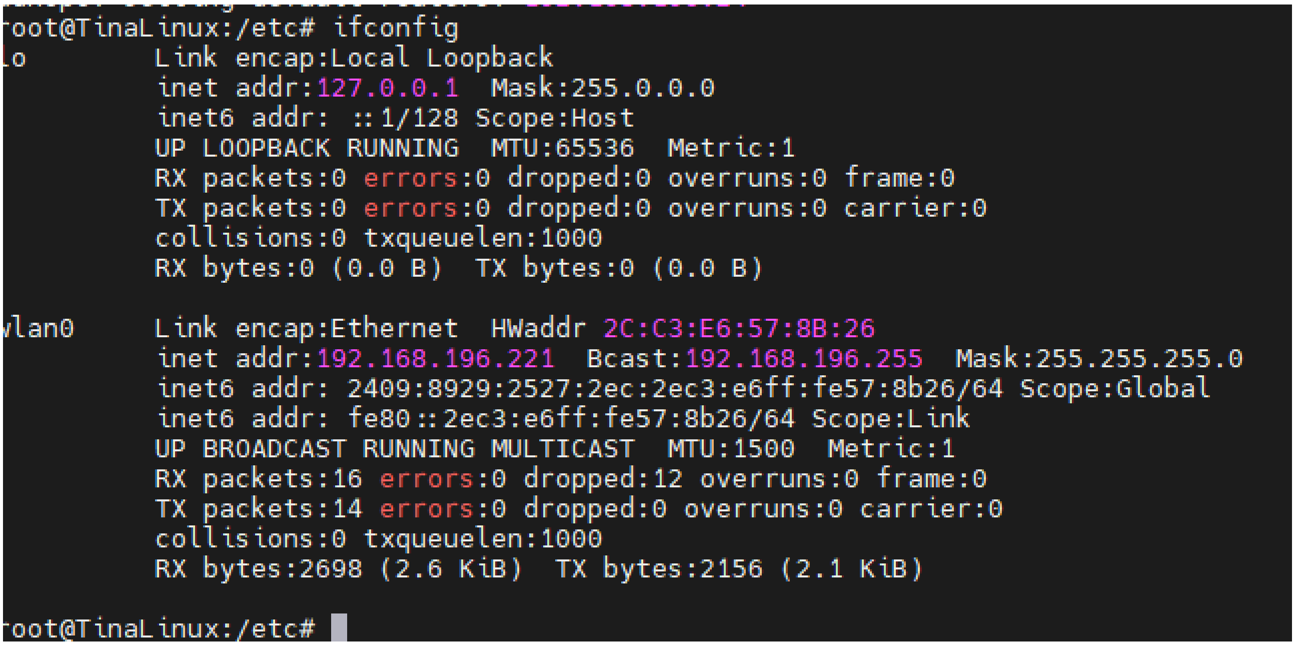
设置 wlan0 的 IP 地址，这里使用 udhcpc 命令从路由器申请 IP 地址

```
udhcpc -i wlan0
```



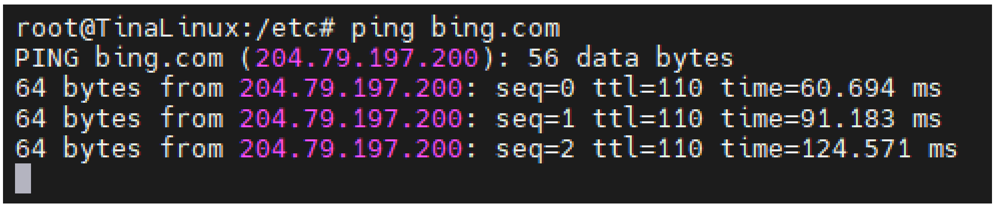
```
root@TinaLinux:/etc# udhcpc -i wlan0
udhcpc: started, v1.27.2
udhcpc: sending discover
udhcpc: sending select for 192.168.196.221
udhcpc: lease of 192.168.196.221 obtained, lease time 3599
udhcpc: ifconfig wlan0 192.168.196.221 netmask 255.255.255.0 broadcast 192.168.196.255
udhcpc: setting default routers: 192.168.196.24
root@TinaLinux:/etc#
```

使用ifconfig命令查看wlan0是否已经分配到ip了



```
root@TinaLinux:/etc# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr 2C:C3:E6:57:8B:26
          inet addr:192.168.196.221  Bcast:192.168.196.255  Mask:255.255.255.0
          inet6 addr: 2409:8929:2527:2ec:2ec3:e6ff:fe57:8b26/64 Scope:Global
          inet6 addr: fe80::2ec3:e6ff:fe57:8b26/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16 errors:0 dropped:12 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2698 (2.6 KiB)  TX bytes:2156 (2.1 KiB)

root@TinaLinux:/etc#
```

尝试ping bing.com



```
root@TinaLinux:/etc# ping bing.com
PING bing.com (204.79.197.200): 56 data bytes
64 bytes from 204.79.197.200: seq=0 ttl=110 time=60.694 ms
64 bytes from 204.79.197.200: seq=1 ttl=110 time=91.183 ms
64 bytes from 204.79.197.200: seq=2 ttl=110 time=124.571 ms
```

可以成功联网了