# 1硬件

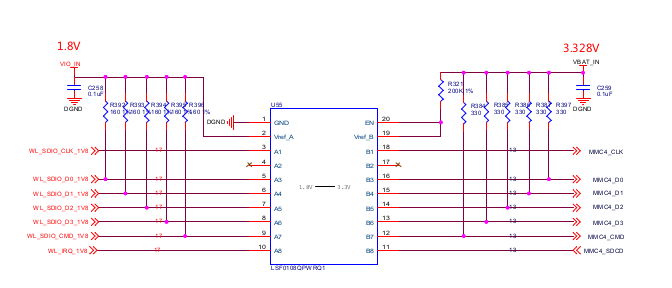


图1

在图一中的5、6、7、8、9引脚上拉160ohm电阻到VIO\_IN，12、13、14、15、16引脚上拉330ohm电阻到VBAT\_IN。

上拉是为了提供直流偏置和驱动能力，适当的上拉可以实现更高的频率转换。上拉电阻后需要将通过处于导通状态的传输晶体管的电流限制为约15mA， 以确保了260 mV至350 mV的通过电压。 如果通过传输晶体管的电流高于15mA，则导通状态下的通过电压也较高。Datasheet提供了一个计算公式：Rpu = (Vpu – 0.35 V) / 0.015 A和一个参考表格，我测试了15mA时的上拉电阻100ohm和240ohm、1.5mA的1K和2K、0.15mA的10K和20K、10mA的160ohm和330ohm，最终确定提供10mA电流的160ohm和330ohm的上拉阻值是符合要求的。

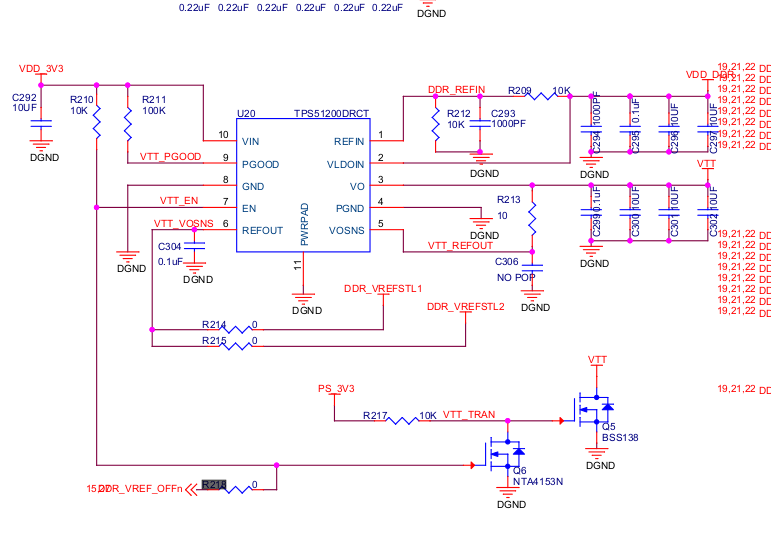


图2

删除图二中的R218(此电阻在核心板上） 。它跟MMC4\_DAT2连在一起，对于wifi运行有影响。

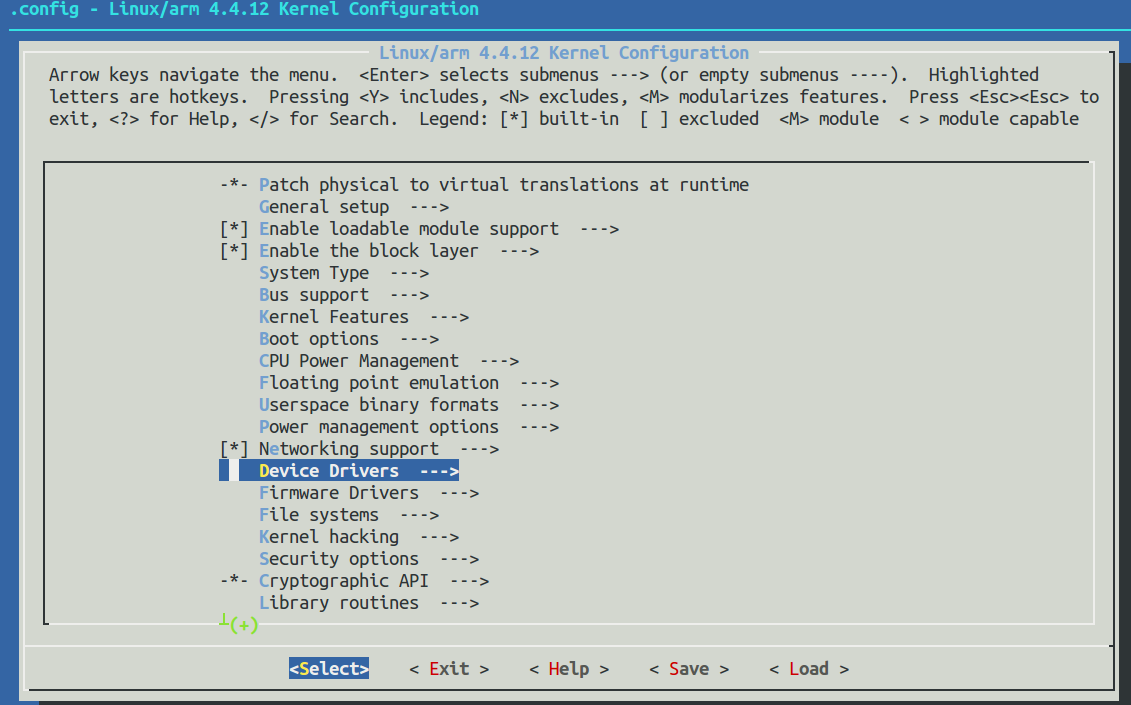
# 2软件

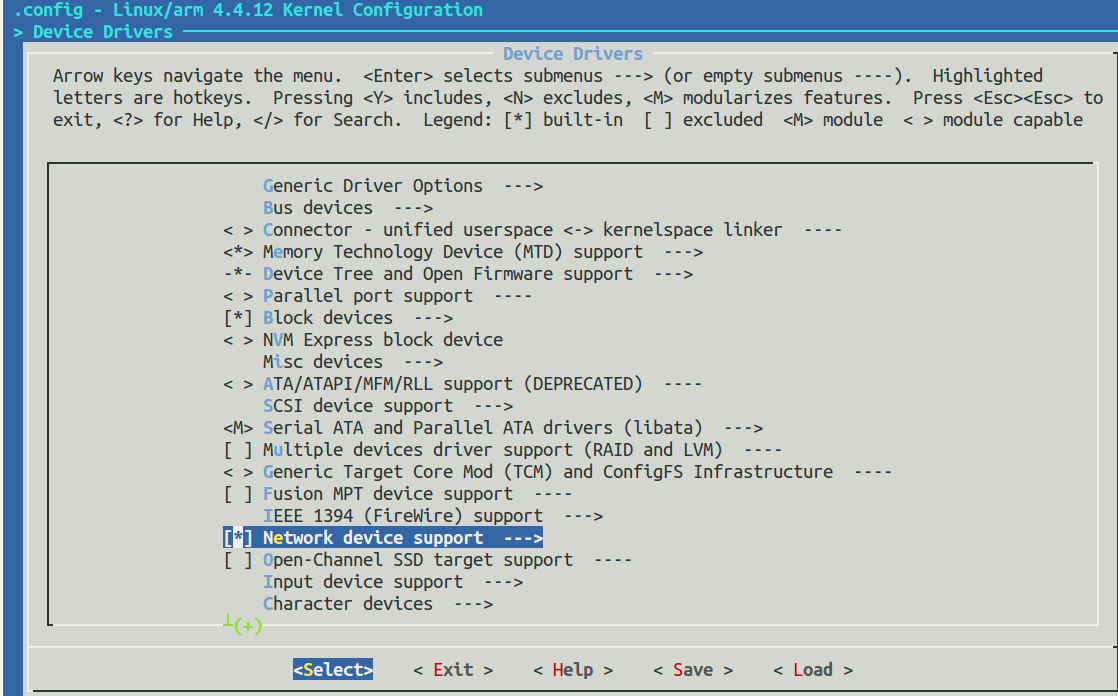
**2.1内核及驱动版本**

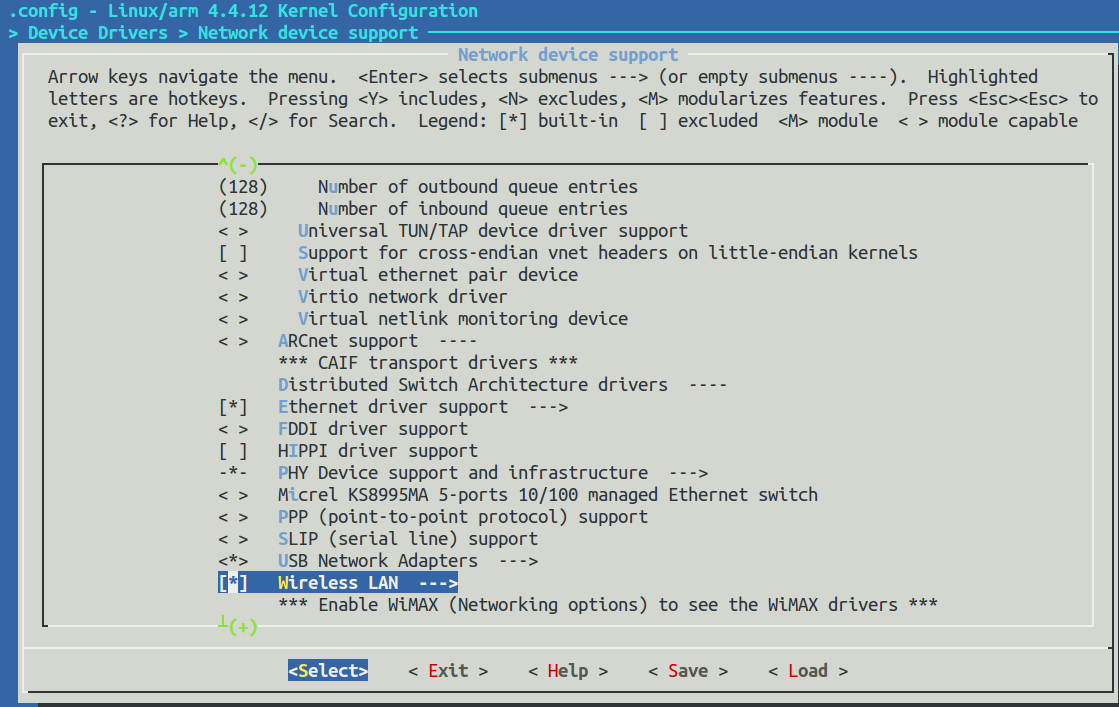
我使用的是TI官方提供的03.00.00.04 SDK，内核版本是4.4.12，wifi驱动使用编译过后的wlcore.ko、wlcore\_sdio.ko、wl18xx.ko,它们依赖于上层的cfg80211.ko和mac80211.ko。

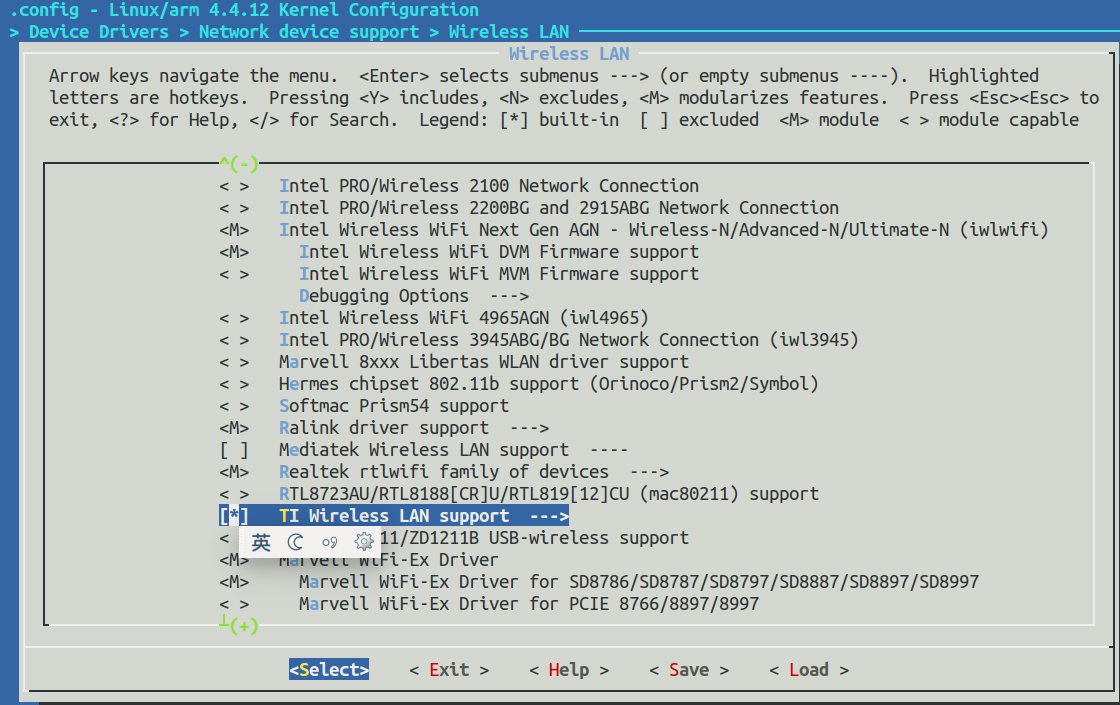
## **2.2内核配置**

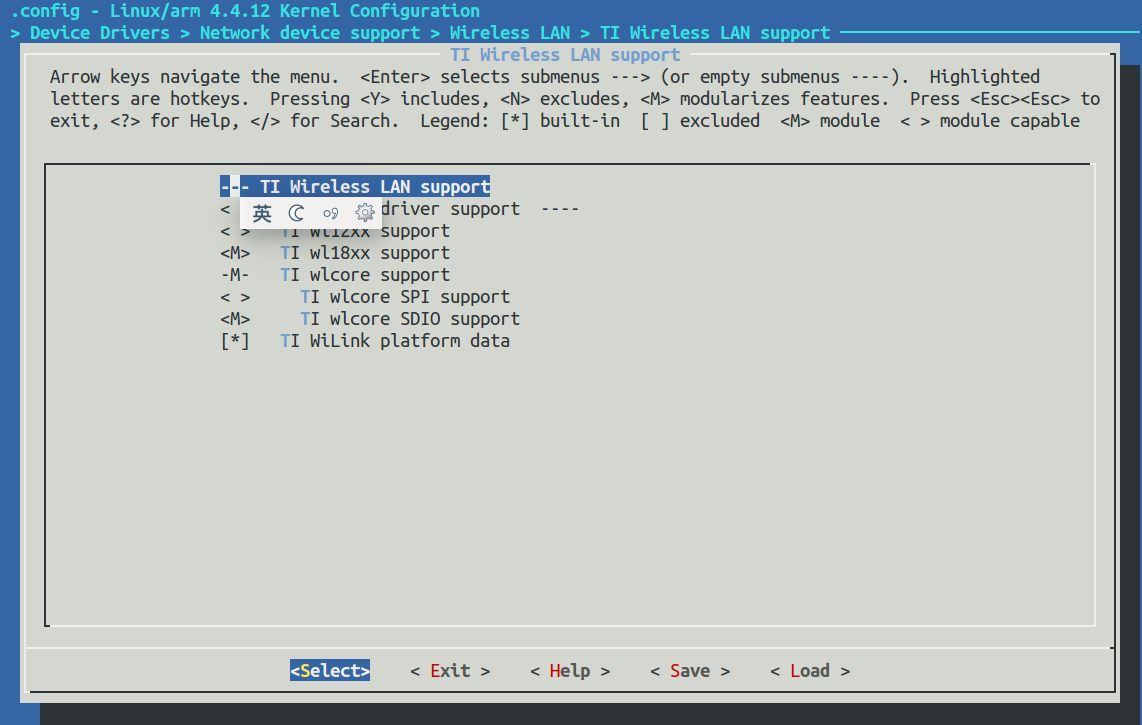
内核配置修改如下：



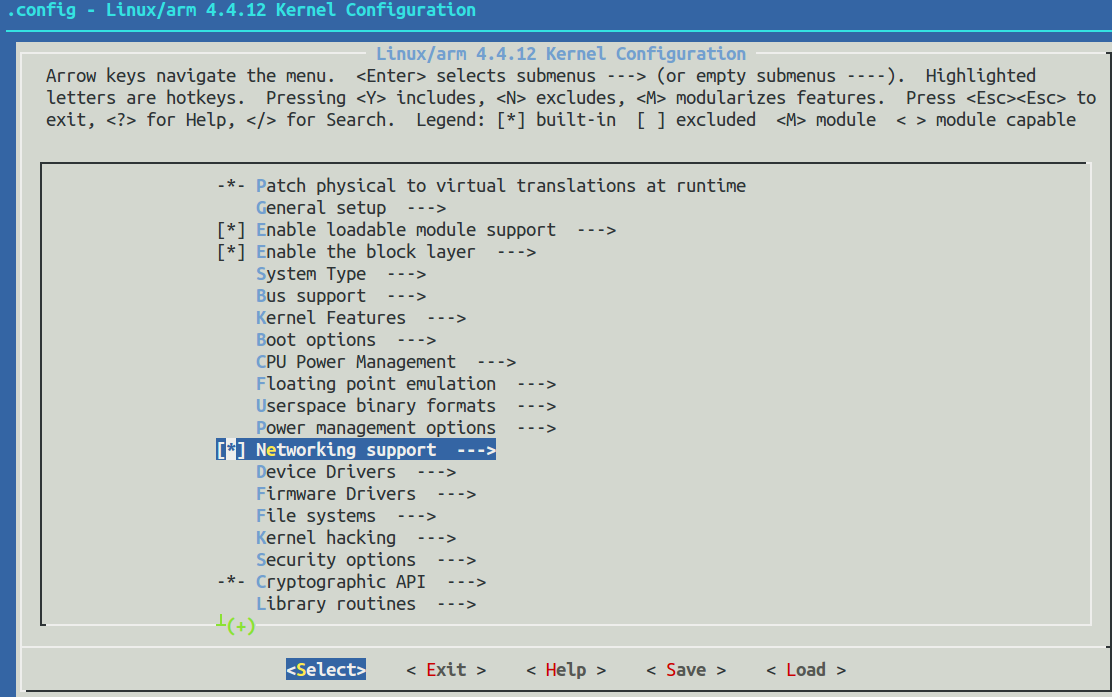


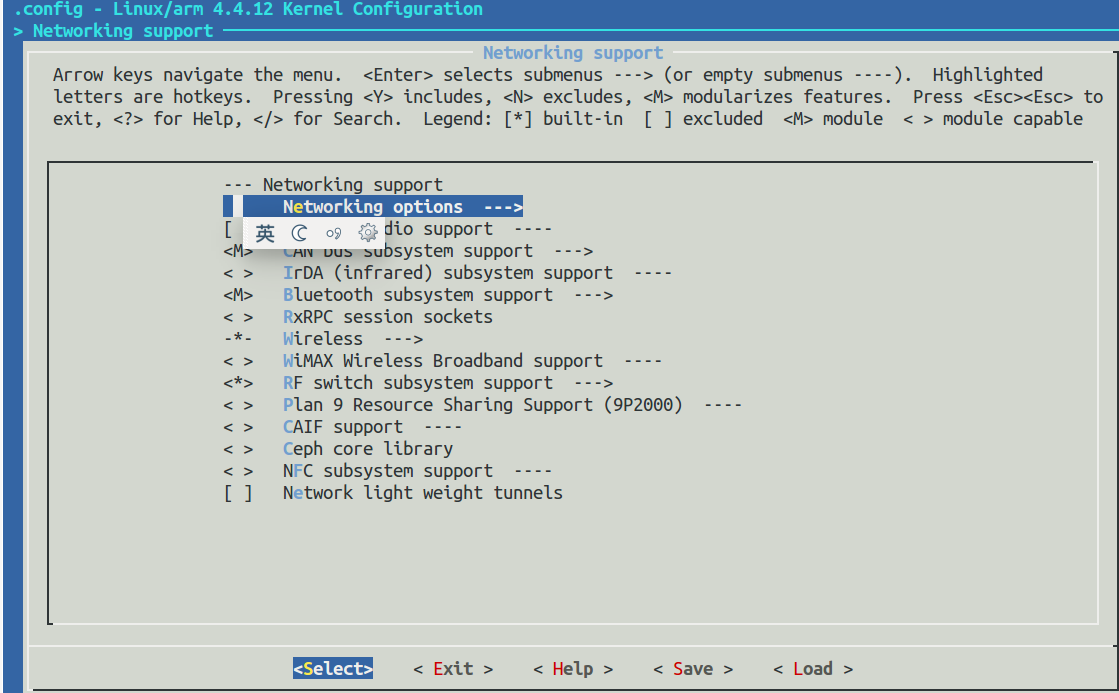


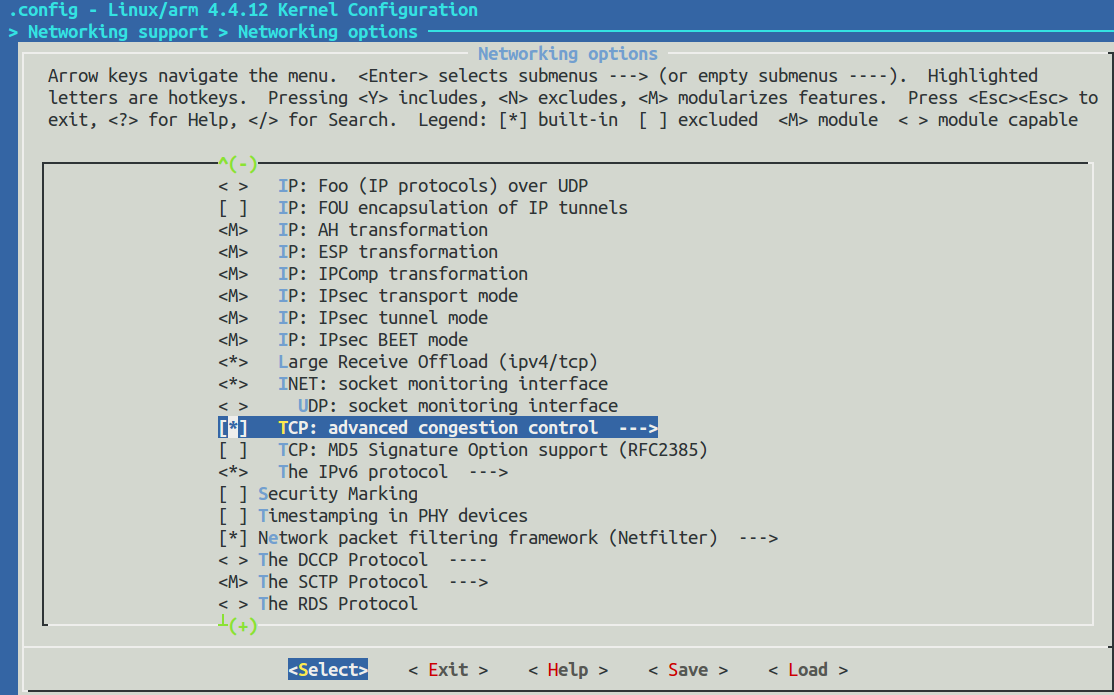


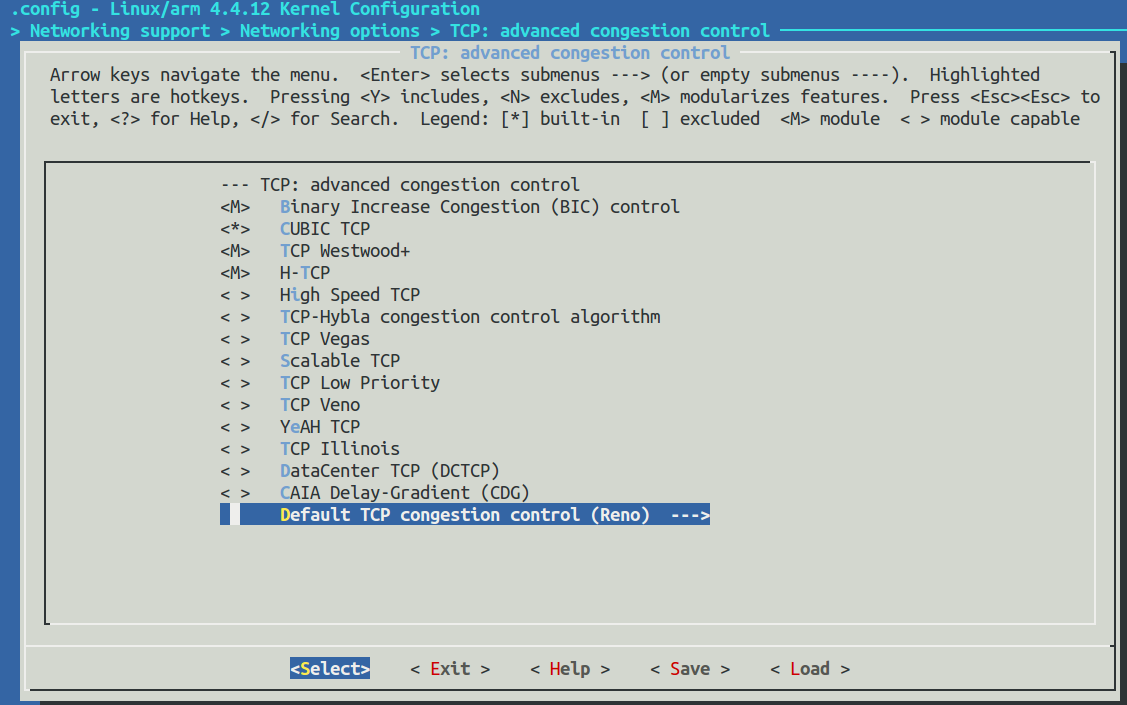


以上是驱动的选择。









以上是让WIFI能够以高速率收发。

修改后.config文件跟改了如下内容：

CONFIG\_WL\_TI=y

CONFIG\_WL18XX=m

CONFIG\_WLCORE=m

CONFIG\_WLCORE\_SDIO=m

CONFIG\_WILINK\_PLATFORM\_DATA=y

CONFIG\_TCP\_CONG\_ADVANCED=y

CONFIG\_DEFAULT\_RENO=y

CONFIG\_DEFAULT\_TCP\_CONG="reno"

设备树里添加以下内容：

/ {

vmmcwl\_fixed: fixedregulator-mmcwl {

compatible = "regulator-fixed";

regulator-name = "vmmcwl\_fixed";

regulator-min-microvolt = <1800000>;

regulator-max-microvolt = <1800000>;

gpio = <&gpio7 14 GPIO\_ACTIVE\_HIGH>;

startup-delay-us = <70000>;

enable-active-high;

};

};

&dra7\_pmx\_core {

mmc4\_pins: pinmux\_mmc4\_pins {

pinctrl-single,pins = <

0x37f0 (PIN\_INPUT\_PULLUP | MUX\_MODE3) /\* mmc4\_dat0 \*/

0x37f4 (PIN\_INPUT\_PULLUP | MUX\_MODE3) /\* mmc4\_dat1 \*/

0x37f8 (PIN\_INPUT\_PULLUP | MUX\_MODE3) /\* mmc4\_dat2 \*/

0x37fc (PIN\_INPUT\_PULLUP | MUX\_MODE3) /\* mmc4\_dat3 \*/

0x37e8 (PIN\_INPUT\_PULLUP | MUX\_MODE3) /\* mmc4\_clk \*/

0x37ec (PIN\_INPUT\_PULLUP | MUX\_MODE3) /\* mmc4\_cmd \*/

>;

};

};

&mmc4 {

status = "okay";

vmmc-supply = <&vmmcwl\_fixed>;

bus-width = <4>;

pinctrl-names = "default";

pinctrl-0 = <&mmc4\_pins>;

max-frequency = <20000000>;

ti,non-removable;

ti,needs-special-hs-handling;

cap-power-off-card;

keep-power-in-suspend;

#address-cells = <1>;

#size-cells = <0>;

wlcore: wlcore@0 {

compatible = "ti,wl1837";

reg = <2>;

interrupt-parent = <&gpio7>;

interrupts = <22 IRQ\_TYPE\_LEVEL\_HIGH>;

};

};

**2.3引脚复用**

UBOOT里引脚复用修改以下内容：

{SPI2\_SCLK,(M14 | PIN\_OUTPUT\_PULLDOWN)},/\*spi2\_sclk.gpio7\_14 wl\_en\*/

{UART1\_CTSN,(M3 | PIN\_INPUT\_PULLUP)}, /\*uart1\_ctsn.Driveroff wl\_clk\*/

{UART1\_RTSN,(M3 | PIN\_INPUT\_PULLUP)}, /\*uart1\_rtsn.gpio7\_25 wl\_cmd\*/

{UART2\_RXD,(M3 | PIN\_INPUT\_PULLUP)}, /\*N/A.Driveroff wl\_dat0\*/

{UART2\_TXD, (M3 | PIN\_INPUT\_PULLUP)}, /\*uart2\_txd.Driveroff wl\_dat1\*/

{UART2\_CTSN, (M3 | PIN\_INPUT\_PULLUP)}, /\*uart2\_ctsn.uart3\_rxd wl\_dat2\*/

{UART2\_RTSN, (M3 | PIN\_INPUT\_PULLUP)}, /\*uart2\_rtsn.uart3\_txd wl\_dat3\*/

{UART1\_RXD, (M14 | PIN\_INPUT)}, /\*uart1\_rxd.uart1\_rxd wl\_irq\*/

**3初始化设置**

在/usr/share/wl18xx一个叫configure-device.sh的初始化配置脚本，从官网可以得到，根据提示选择自己所用的配置即可。（官网下载的脚本名字可能叫wlconf\_configure-device.sh，改个名字就行）。运行configure-device.sh时需要一个初始化文件，这个文件在/usr/sbin/wlconf/official\_inis下叫WL1837MOD\_INI\_FCC\_CE.ini或是WL1837MOD\_INI\_FCC\_CE\_JP.ini，根据你运行脚本是选择不同而不一样（板子若无初始化文件夹的话，就需要自己创建目录，目录只能是/usr/sbin/wlconf/official\_inis，并在官网下载初始化文件）。

在wifi模块加载时会需要一些固件，如果没有则会在加载驱动模块时报错，他会提示你需要的是哪个固件，将需要的固件下载到/lib/firmware/ti-connectivity/就可以，我用的固件是wl18xx-fw-4.bin。