**RK3288 WIFI兼容调试方法**

|  |  |  |
| --- | --- | --- |
| **文件状态： [√] 草稿 [ ] 正在修改**  **[ ] 正式发布** | **部门** | **系统软件组** |
| **版本** | 0.1 |
| **作者** | **朱坤华** |
| **完成时间** |  |
| **审核** |  |
| **审核时间** |  |
| **密级状态：绝密( ) 秘密( ) 内部资料(√) 公开( )** | | |

**修改记录:**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修订者 | 时间 | 说明 |
| 0.1 | 朱坤华 | 2018 / 4 / 26 | RK3288 WIFI兼容调试记录文档 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**一.背景及问题:**

RK3288上现在有ATV1560K和ATV1660K的两个产品，代码是同一份，WIFI部分做过自动兼容，该文档记录新的WIFI的兼容调试方法。

**二. 调试记录：**

目录

[简介 3](#_Toc27751)

[模块上电 3](#_Toc13043)

[USB WIFI底层驱动调试方法 3](#_Toc5656)

[SDIO WIFI底层驱动调试方法 4](#_Toc20730)

[WIFI模块识别 4](#_Toc6483)

[WPA\_SUPPLICANT 调试 7](#_Toc9087)

简介：

调试WIFI兼容相关资料路径：

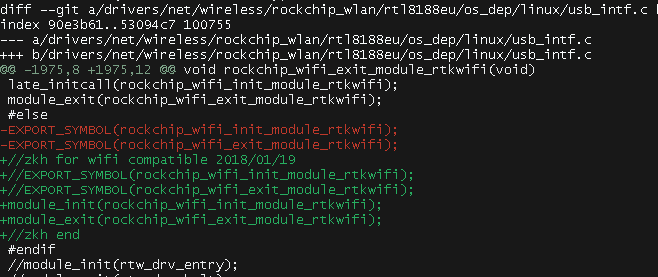
\\192.168.1.8\work\home\zkh\RK\项目资料\ATV1560K\WIFI 兼容

模块上电**：**

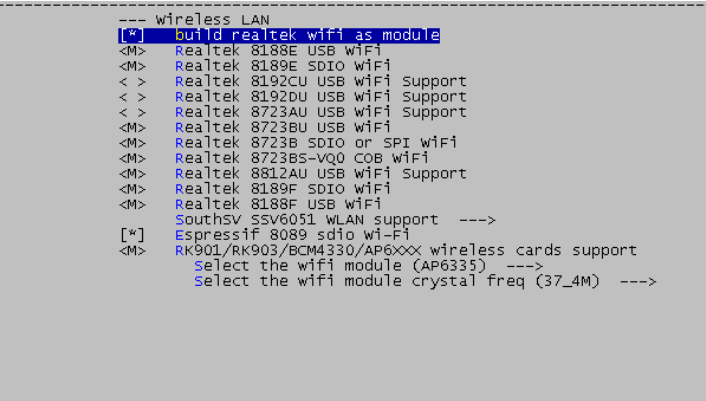
模块上电部分没有改动，按照原理图实际情况配置dts即可。

USB WIFI底层驱动调试方法**：**

兼容usb wifi的底层驱动是做成ko文件，调试新的wifi的时候也需要将驱动编译成ko文件。比如realtek系列的，可以在驱动里面，把文件usb\_intf.c 的驱动加载模式改为module的。



然后make menuconfig，将驱动编译成模块ko文件。



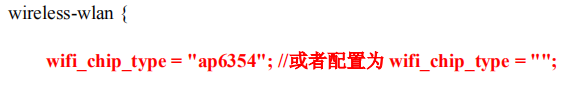
编译出的模块ko文件放到板子上，命令行下直接insmod文件，看看驱动加载情况，加载成功后，用iwlist wlan0 scan命令扫描wifi，能扫描到ssid，则可以确认底层驱动通了。底层驱动好了后，把ko文件拷贝到目录：

vendor/rockchip/common/wifi/modules

SDIO WIFI底层驱动调试方法**：**

目前SDIO的时钟是50MHZ的，也就是SDIO2.0的CLK。如果是支持SDIO3.0的WIFI模块，代码需要另外去更改，RK3288平台的CLK最高可以支持到125M。

AP6XXX系列的WIFI驱动用的是通用的博通的驱动boardcom，bcmdhd.ko，博通的这个驱动支持RK901&RK903&AP6XXX，一般情况下不需要再改动驱动了，做wifi兼容之前需要在dts里面配置支持的wifi，但最新的是不需要配置了，配置了也不会生效。



ko文件可以直接insmod驱动，然后iwlist扫描wifi ssid看看是否通了。

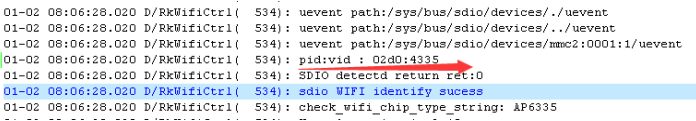
WIFI模块识别**：**

WIFI兼容的原理是通过PID/VID来区分什么模块。net/rfkill/rfkill-wlan.c 文件的 prob 函数中直接调用上电函数和扫卡函数来区分不同的 wifi chip。由于 USB 结构可能比较多，不能单独对某个路径进行读取，比如有的机器可能会有 HUB，其挂载的 USB 路径跟 SDK 默认的 USB 路径会不一致，所以采用读取 UEVENT 文件的方式（ uevent 文件是 USB 协议标准规范），同样 SDIO 识别同样类似处理。UEVENT 在文件系统的挂载路径如下：

USB :sys/bus/usb/devices

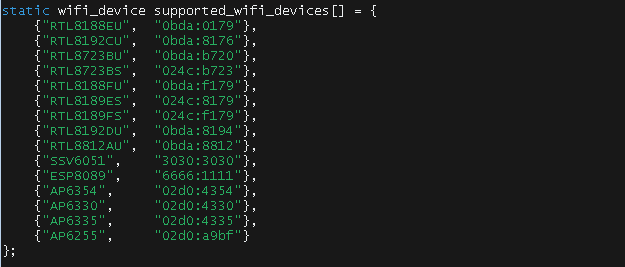
Sdio : /sys/bus/sdio/devices

logcat | grep "RkWifiCtrl"读取出的 wifi 芯片如下：



获取到这个PID和VID后，加入到代码中并填入：

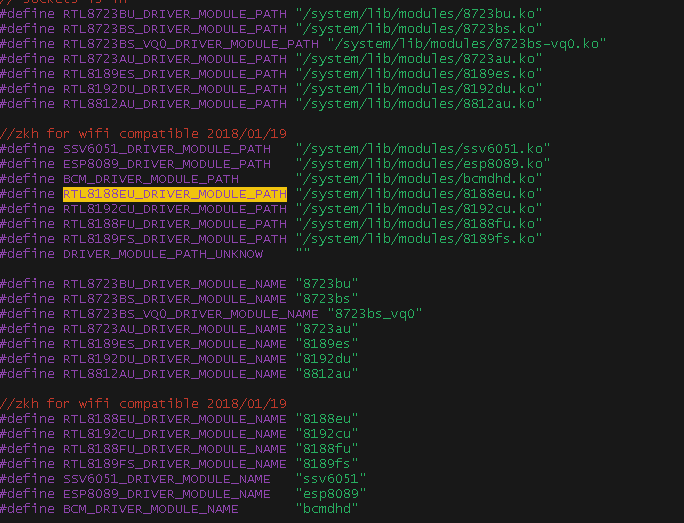
hardware/libhardware\_legacy/wifi/rk\_wifi\_ctrl.c



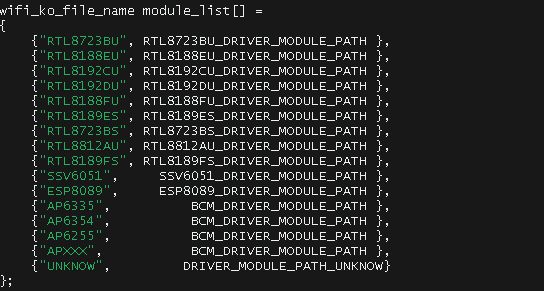
如想深入了解识别原理，可查看函数wifi\_get\_sdio\_device\_id和wifi\_get\_usb\_device\_id查看识别的过程。

加完PID和VID以后，模块的驱动ko名称和WIFI名称也要确认加到代码里面：

hardware/libhardware\_legacy/wifi/wifi.c



并加到模块列表：



wifi\_load\_driver函数会遍历module\_list来识别是否匹配识别到的模块并加载ko文件。

WPA\_SUPPLICANT 调试 **：**

正常情况下wpa\_supplicant都已经配好了，如果有问题，需要检查wpa\_supplicant，在device/rockchip/common/init.connectivity.rc。

wpa\_supplicant的启动在hardware/libhardware\_legacy/wifi/wifi.c，因为做了兼容，所以启动方式通过判断模块类型：



Broadcom, Realtek, ESP8089 三者使用的 wpa\_supplicant 可能有些区别，目前没有将 Realtek, ESP8089 分别独立出来编译作区分，如果有问题，可以单独编译出来，丢到板子上测试看看。