

V40 项目

wifi 模块移植说明文档 V1.0

文档履历

[illegible]

目 录

V40 项目	1
wifi 模块移植说明文档 V1.0	1
1. 前言	3
1.1. 编写目的	3
1.2. 适用范围	3
1.3. 相关人员	3
2. 模块介绍	4
3. 配置说明	5
3.1. Wifi 相关配置	5
3.2. 蓝牙相关配置	5
4. F&Q	7
4.1. 如何编译	7
4.2. 编译报错	7
4.3. 修改不起效	7
4.4. 蓝牙无法使用问题排查	7
4.4.1. 从硬件和配置定位问题	8
4.4.2. 从 log 打印定位问题	9
5. Declaration	10

1. 前言

1.1. 编写目的

本文档编写目的是让相关人员能按照文档说明把 SDK 所支持的 wifi 使用起来，要使用 SDK 所支持 wifi 的其中一款只需要修改相关配置即可。

1.2. 适用范围

本文档适用于 android4.4 上移植 wifi 使用。

1.3. 相关人员

本文档适合需要使用或移植 wifi 的人员查阅。

2. 模块介绍

目前 V40 项目已支持 wifi 模组有 4 款，本文档将以 V40 CDR 原型机为例说明如何配置每款 wifi 模组。

wifi 模组可分 USB 接口和 SDIO 接口两种类型，部分模组带蓝牙功能，wifi 的全功能包括 station、softap 和 wifi direct，其中 station 是 wifi 的最基本功能，每款 wifi 模组均支持。

本文档会不断更新，文档和代码对应可能会稍有差别。

表 1 V40 sdk wifi 模组支持列表

模组	功能（协议）	接口	wifi 功能		
			Station	Softap	wifi direct
rtl8188eu/etv	802.11 b/g/n	USB	√	√	√
rtl8189etv	802.11b/g/n	SDIO	√	√	√
rtl8723bs	802.11b/g/n BT4.0	SDIO/PCM/UART	√	√	√
rtl8192eu	802.11b/g/n	USB	√	√	√
ap6212	802.11b/g/n	SDIO	√	√	√

3. 配置说明

目前 V40 项目所支持的 wifi 仅需要修改 sys_config，修改说明如下所示：

3.1. Wifi 相关配置

```
[wlan]
wlan_used          = 1
wlan_busnum        = 1
wlan_usbnum        = 3
;clocks            = "&clk_losc_out"
wlan_power         = "vcc-wifi"
wlan_io_regulator   = "vcc-wifi-io"
wlan_en            = port:PL07<1><default><default><0>
wlan_regon         = port:PL03<1><default><default><0>
wlan_hostwake      = port:PA11<6><default><default><0>
```

说明：

- 1、“；”符号起注释作用；
- 2、“wlan_busnum”宏表示 wifi 所使用的 SDIO 控制器号；
- 3、“wlan_usbnum”宏表示 wifi 所使用的 USB 控制器号；
- 4、“clocks”宏表示是否使用主控的 32k 时钟，如果不需要使用则将其注释即可，如需使用则打开；
- 5、“wlan_power”宏表示给 wifi 模组供电的 axp 名称；
- 6、“wlan_io_regulator”宏表示给 wifi 的 gpio 供电的 axp 的名称；
- 7、“wlan_en”宏表示控制 wifi 电源的 gpio；
- 8、“wlan_regon”宏表示 wifi 的功能使能脚；
- 9、“wlan_hostwake”宏表示 wifi 唤醒主控的 gpio；

注意：

- 1、模组供电电源和使用哪些 pin 做为控制需要根据具体平台而定；

3.2. 蓝牙相关配置

```
[bt]
bt_used          = 1
;clocks          = "&clk_losc_out"
bt_power         = "vcc-wifi"
bt_io_regulator   = "vcc-wifi-io"
bt_rst_n         = port:PL01<1><default><default><0>

[btlpm]
btlpm_used       = 1
uart_index       = 1
bt_wake          = port:PA12<1><default><default><1>
bt_hostwake      = port:PL06<6><default><default><0>
```

说明：

- 1、“；”符号起注释作用；
- 2、“clocks”宏表示是否使用主控的 32k 时钟，如果不需要使用则将其注释即可，如需使用则打开；

- 3、“bt_power”宏表示给蓝牙模组供电的 axp 名称;
- 4、“bt_io_regulator”宏表示给蓝牙的 gpio 供电的 axp 的名称;
- 5、“bt_rst_n”宏表示蓝牙的功能使能脚;
- 6、“uart_index”宏表示蓝牙所使用的 uart 控制器号;
- 7、“bt_wake”宏表示主控唤醒蓝牙的 gpio;
- 8、“bt_hostwake”宏表示蓝牙唤醒主控的 gpio;

注意:

- 1、模组供电电源和使用哪些 pin 做为控制需要根据具体平台而定;

4. F&Q

4.1. 如何编译

修改配置文件后有两种编译方法可以使得修改生效，第一种是整体编译重烧固件，第二种是单独的模块编译然后把生成的文件 push 进相应的目录，详细操作如下。

整体编译

整体编译有两种方法可选

方法一：在 android 主目录下先 “make clean” 后再 “make -j8”；

方法二：分别到 android\hardware\libhardware_legacy\wifi\和 external\wpa_supplicant_8\

wpa_supplicant\目录下使用 “touch *” 命令更新目录下所有文件的时间戳，再把 android 主目录下使用 “make -j8” 命令编译；

单独模块编译

先分别到 android\hardware\libhardware_legacy\wifi\和 external\wpa_supplicant_8\wpa_suppli

cant\目录下使用 “touch *” 命令更新目录下所有文件的时间戳，然后在相应目录下使用 “mm” 命令进行编译，再把生的 libhardware_legacy.so 和 wpa_supplicant 推入目标机器目录，最后更改执行权限并重启。

若生成的 libhardware_legacy.so 和 wpa_supplicant 存放在电脑 D 盘下，则用如下命令把其推到目标机器目录下。

```
adb push D:\wap_supplicant /system/bin/wpa_supplicant
adb push D:\libhardware_legacy.so /system/lib/libhardware_legacy.so
adb shell chmod +x /system/bin/ wpa_supplicant
adb shell chmod 666 /system/lib/libhardware_legacy.so
```

注意：单独模块编译适合调试使用。

4.2. 编译报错

修改配置文件后，如果编译的时候报与 wpa_supplicant_8 相关的错误，可用以下两种方法解决。

方法一：将 out/target/product/方案名/obj/EXECUTABLES/wpa_supplicant_intermediates 下面的中间生成文件删除，再次编译。

方法二：使用方法一后编译还是报错，则需要先使用 make clean 命令清除中间文件和目标文件后，再使用 make -j8 命令进行整体编译。

4.3. 修改不起效

修改配置文件并进行整体编译重烧固件或单独模块编译把生成文件推进目标机器后发现指定的 wifi 模组不能工作，修改配置前那款 wifi 模组还是能正常工作。该问题是因为由于只修改配置文件，wifi 相关的编译文件的时间戳没改变使用不会去重新编译，详细解决方法可参考 “如何编译”。

4.4. 蓝牙无法使用问题排查

Android 上蓝牙无法使用从现象上来看只有一个，就是蓝牙打不开。看到此现象时，需要从两大方面来定位问题，首先是硬件和配置上的检查，第二是从蓝牙打不开的 log 打印去分析。

4.4.1. 从硬件和配置定位问题

可以按照以下几个步骤进行：

(1) 检查蓝牙模组电源

使用万用表量蓝牙模组的 VBAT 引脚，看电压是否为 3.3V 左右。如果不是，则需要从 sys_config 配置上来确定所使用的 axp 电源是否正常，务必确保 VBAT 电源正确再往下进行排查。如果 wifi 可以打开，那么也可以说明 VBAT 电源是正确的。

(2) 检查 UART 连接

这一步一般没什么问题，不过还是需要在原理图上检查一下，就是主控的 UART_TX、UART_RX、UART_CTS 和 UART_RTS 和模组的这几个引脚是交叉连接的。

(3) 检查 32K 时钟

为了节约成本，绝大多数情况下都是使用主控的 32k 时钟作为蓝牙模组的低功耗时钟，APXXX 的所有模组和 realtek 的 COB 模组都需要这个时钟。有两种方法检查，第一是用万用表量蓝牙模组的 LPO 引脚，看电压是否为 1.5V 左右，第二是用示波器量 LPO 引脚的信号是否为 32KHz。

主控不输出 32k 时钟有可能是硬件问题，也有可能是软件问题，软件问题可以通过查看寄存器来确认。硬件定位方法：

主控的 32k 时钟来源于外部的一个 32k 晶振，如果 32k 晶振本身有问题，那么主控肯定也是不能输出 32k 时钟的，因为需要从源头处排查。32k 晶振与主控的连接原理图如图 2 所示，用示波器量 X32KI 和 X32KO 这两处，正常情况下至少有一处为 32KHz，如果不是，则说明 32k 晶振没工作。

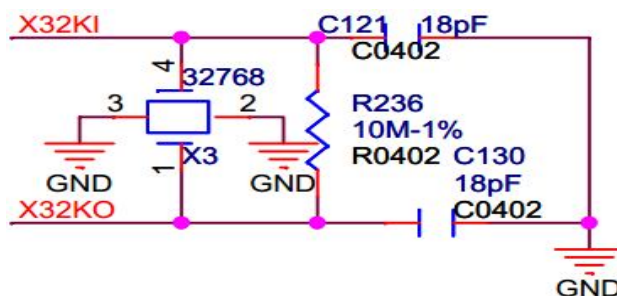


图 2 32k 晶振原理图

如果 32k 晶振没问题，接下来检查主控 32k 时钟引脚到模组之间的电路，确保 32k 时钟输出的上拉电阻已经焊接上。

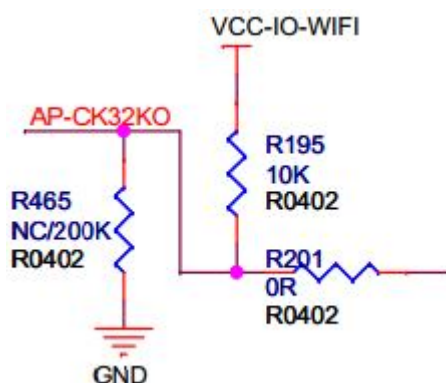


图 3 32k 时钟外部硬件原理图

经过以上这些排查，一定可以知道并确保 32k 时钟是正常工作。

(4) 检查 BT_RST_N 引脚

根据原理图，检查 sys_config 里的 BT_RST_N 引脚配置是否正确，在打开蓝牙时，用万用表量这个引脚一段时间，看电平是否会发生变化，正常情况下是会变化的。

4.4.2. 从 log 打印定位问题

蓝牙打不开，从 log 来看，绝大多数情况下都会出现以下这句打印：

```
...preload_wait_timeout (retried:%d/max-retry:%d)...
```

导致这个问题出现的原因有这些：主控与蓝牙模组的 UART 连接不正确；32k 时钟没起来；BT_RST_N 引脚配置不正确，需要重点检查后两者。对于 APXXX 的模组，还需要确保以下代码被执行了（可以通过加打印来确定）。

hardware\broadcom\libbt\src\serial_vendor.c 文件：

```
if (strcmp(module_info.mod_name, "ap6210") == 0)
{
    usleep(100000);
    close(vnd_serial.fd);
    if ((vnd_serial.fd = open(vnd_serial.port_name, O_RDWR)) == -1)
    {
        ALOGE("userial vendor open: unable to open %s", vnd_serial.port_name);
        return -1;
    }
}
else if (strcmp(module_info.mod_name, "ap6335") == 0)
{
    usleep(500000);
    close(vnd_serial.fd);
    if ((vnd_serial.fd = open(vnd_serial.port_name, O_RDWR)) == -1)
    {
        ALOGE("userial vendor open: unable to open %s", vnd_serial.port_name);
        return -1;
    }
}
else if (strcmp(module_info.mod_name, "ap6212") == 0)
{
    usleep(300000);
    close(vnd_serial.fd);
    if ((vnd_serial.fd = open(vnd_serial.port_name, O_RDWR)) == -1)
    {
        ALOGE("userial vendor open: unable to open %s", vnd_serial.port_name);
        return -1;
    }
}
```

另外，Android 的 config.disable_bluetooth 属性可以控制是否使用蓝牙，如果这个属性的值为 true，那么蓝牙肯定是打不开的。因此需要确保此属性值为 false 或者没有这个属性。

5. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner 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 Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.