



XR829 Android/Linux WLAN Software Program Guide

Revision 1.0
July 17, 2018

Declaration

THIS DOCUMENTATION IS THE ORIGINAL WORK AND COPYRIGHTED PROPERTY OF XRADIO TECHNOLOGY ("XRADIO"). REPRODUCTION IN WHOLE OR IN PART MUST OBTAIN THE WRITTEN APPROVAL OF XRADIO AND GIVE CLEAR ACKNOWLEDGEMENT TO THE COPYRIGHT OWNER.

THE INFORMATION FURNISHED BY XRADIO IS BELIEVED TO BE ACCURATE AND RELIABLE. XRADIO RESERVES THE RIGHT TO MAKE CHANGES IN CIRCUIT DESIGN AND/OR SPECIFICATIONS AT ANY TIME WITHOUT NOTICE. XRADIO 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 XRADIO. THIS DATASHEET NEITHER STATES NOR IMPLIES WARRANTY OF ANY KIND, INCLUDING FITNESS FOR ANY PARTICULAR APPLICATION.

THIRD PARTY LICENCES MAY BE REQUIRED TO IMPLEMENT THE SOLUTION/PRODUCT. CUSTOMERS SHALL BE SOLELY RESPONSIBLE TO OBTAIN ALL APPROPRIATELY REQUIRED THIRD PARTY LICENCES. XRADIO SHALL NOT BE LIABLE FOR ANY LICENCE FEE OR ROYALTY DUE IN RESPECT OF ANY REQUIRED THIRD PARTY LICENCE. XRADIO SHALL HAVE NO WARRANTY, INDEMNITY OR OTHER OBLIGATIONS WITH RESPECT TO MATTERS COVERED UNDER ANY REQUIRED THIRD PARTY LICENCE.

Revision History

Version	Data	Summary of Changes
1.0	2018-7-17	Initial Version

表 0-1 Revision History

目录

Declaration.....	2
Revision History.....	3
目 录.....	4
表.....	6
图.....	7
1 结构简介.....	8
1.1 分层结构.....	8
1.2 代码目录.....	9
2 编译配置.....	10
2.1 内核配置.....	10
2.2 Makefile 编译配置.....	11
3 电源/时钟/中断.....	13
3.1 电源控制.....	13
3.1.1 供电（PMU）配置.....	13
3.1.2 Reset 配置.....	13
3.1.3 上下电时序.....	13
3.2 时钟配置.....	14
3.2.1 高频（24M）时钟配置.....	14
3.2.2 32K 时钟配置.....	14
3.3 中断配置.....	14
3.3.1 GPIO 中断配置.....	14
3.3.2 SDIO 中断配置.....	15
4 SDIO 卡检测（扫卡）.....	16
5 MAC 地址.....	17
5.1 主控保存.....	17
5.1.1 参数传递.....	17

5.1.2 文件保存.....	17
5.1.3 分区保存.....	17
5.2 efuse 保存.....	17
5.3 随机生成.....	18
6 休眠唤醒.....	19
6.1 WoW 功能.....	19
6.2 休眠 WiFi 掉电.....	19
6.3 Extern Standby.....	19
7 其他配置.....	20
7.1 驱动移植/更新.....	20
7.2 查看/修改高频时钟.....	20
7.3 修改最大发射功率.....	20
7.4 Monitor 模式.....	22
7.4.1 iw 工具准备.....	22
7.4.2 Monitor 模式开启.....	22
7.4.3 Monitor 模式调试.....	22
7.4.4 Monitor 模式关闭.....	23
7.5 ACS 功能.....	23

表

表 0-1 Revision History.....	3
-----------------------------	---



图 1-1	Android XRadio WLAN 软件架构.....	8
图 1-2	Android 层代码目录.....	9
图 1-3	Linux 层代码目录.....	9
图 2-1	模块选择.....	10
图 2-2	MAC 层编译选项.....	10
图 2-3	Driver 层编译选项.....	11
图 2-4	Driver 层 debug 选项.....	11
图 7-1	查看/修改高频时钟.....	20
图 7-2	修改最大发射功率 1.....	21
图 7-3	修改最大发射功率 2.....	22

1 结构简介

1.1 分层结构

对于 Android 系统，XRADIO WLAN 的软件架构如下图 1-1。在 Android 层，黄色部分模块为 Android 的标准代码，其中，WPA_SUPPLICANT 主要完成网络管理的工作，包括扫描、连接、密钥交换等等，WiFi HAL 主要完成驱动加载，netd 主要完成 Softap 模式相关配置。WiFi HAL 和 netd 通过 socket 与 WPA_SUPPLICANT 进行通信。对 XRADIO WLAN 的支持，需要做少量的修改。

在 Linux kernel 层，蓝色的部分为 XRADIO WLAN Driver，主要实现 WLAN MAC 层协议，包括管理通路、数据通路，此外还包括对硬件设备的管理，包括电源管理、设备初始化。WPA_SUPPLICANT 通过 NL80211 与 Driver 进行通信，完成网络管理工作；Driver 注册内核 NetDevice，TCP/IP 协议栈通过 socket 与 Driver 进行通信，完成数据传输的任务。Driver 通过 SDIO 总线与 wlan 设备进行通信。对 Driver 的移植，主要是对电源、中断、SDIO 总线等接口的适配。

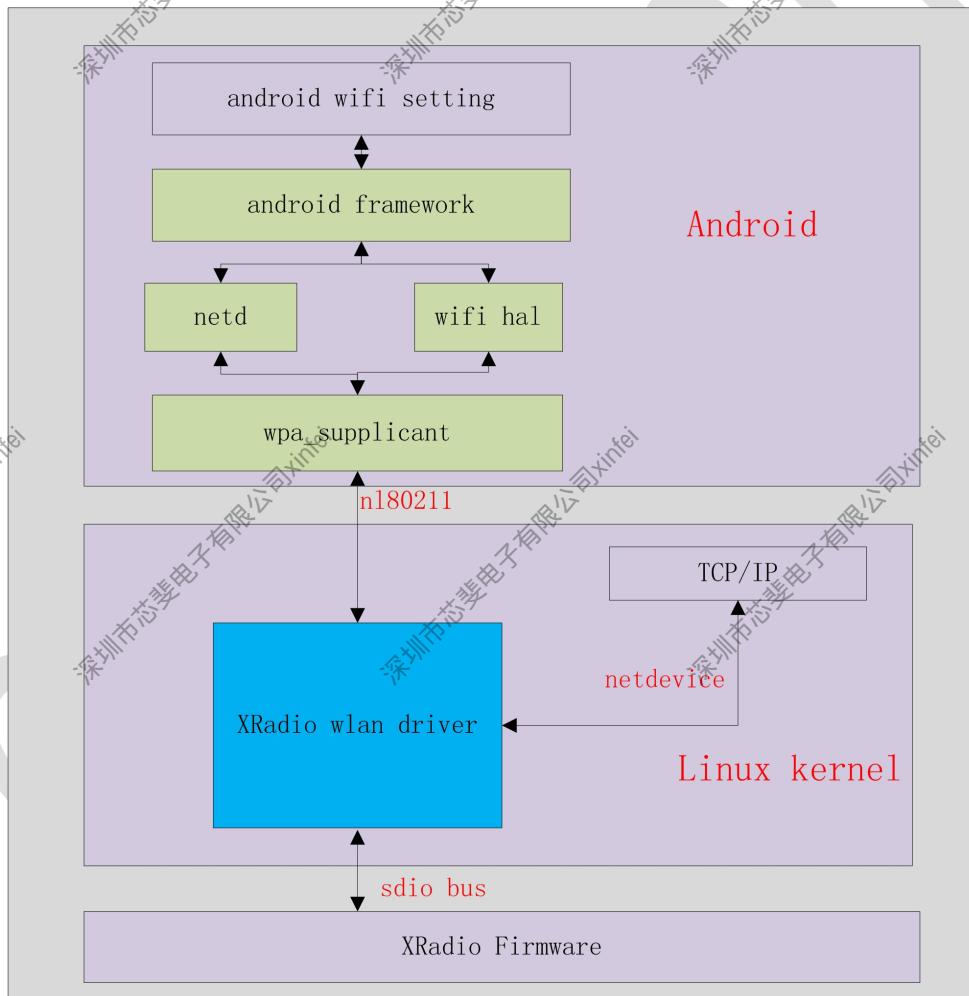


图 1-1 Android XRADIO WLAN 软件架构

1.2 代码目录

Android 层代码目录如下图 1-2，红色框内的 xradio 目录的内容为 xradio 提供，主要包含 wifi 固件、wifi hal 层适配，wpa lib 实现和 wpa 配置文件等。device 目录主要是增加 xradio 相关的初始化服务，framework 主要包含对 xradio 的选择配置。

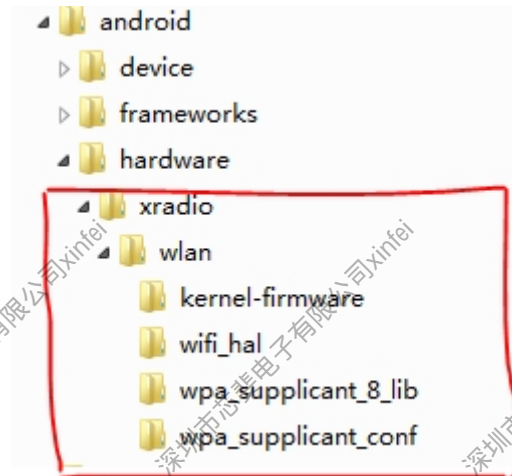


图 1-2 Android 层代码目录

Linux 层代码目录如下图 1-3，红色框内的 xradio 目录的内容为 xradio wlan driver，包含 umac 目录和 wlan 目录。umac 目录主要实现 802.11 mac 协议层，与 NL80211 进行交互；wlan 目录主要实现设备管理以及部分的 mac 协议层，与 wlan 设备进行交互。

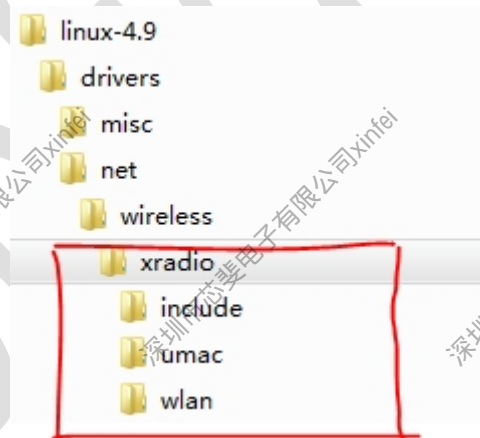


图 1-3 Linux 层代码目录

2 编译配置

2.1 内核配置

如下图 2-1 所示，为 xradio 的模块选择，该两模块分别 umac 和 wlan 两个目录，都为必选项，其中 mac 可以选择编译到内核*，或者编译成模块 M；Driver 则必须编译为模块 M。如果都编译为模块，则会生成三个目标文件，分别为 xradio_mac.ko 和 xradio_core.ko，xradio_wlan.ko。内核配置的路径为 Device Drivers --->Network device support --->Wireless LAN -->Xradio WLAN support。

其中，xradio_wlan.ko 分别依赖 xradio_mac.ko 和 xradio_core.ko，因此 xradio_wlan.ko 加载之前必须先加载 xradio_mac.ko 和 xradio_core.ko。目前，xradio_mac.ko 和 xradio_core.ko 在系统启动时加载，但 wifi 关闭时该两模块基本不会占用资源。

```
--- XRadio WLAN support
<M> XRadio MAC features --->
<M> XRadio Driver features --->
```

图 2-1 模块选择

在 MAC 层中，主要有以下内核选项，如图 2-2 所示。其中，RC_MINSTREL 和 RC_PID 为速率算法选择，必须选择一个，建议选择 RC_MINSTREL 获得更好的性能。Minstrel 802.11n support 是 minstrel 算法对 11n 速率的支持，默认为必选。DEBUGFS 选项为调试接口，方便出现问题时进行 debug，如果内存非常缺少，可以选择关闭。

```
-- XRadio MAC features
(minstrel_ht) XRMAC_RC_DEFAULT
[ ] XRMAC_RC_PID
[*] XRMAC_RC_MINSTREL
[*] Minstrel 802.11n support
[*] XRMAC_DEBUGFS
```

图 2-2 MAC 层编译选项

在 Driver 层中，主要有以下内核选项，如图 2-3、2-4 所示。其中，SDIO Bus wlan 设备通信的总线选择，目前只支持 SDIO 接口，因此该项为必选。

non-power-of-two SDIO 选项是指主控平台的 SDIO 支持以非 2 的指数倍的长度进行数据传输，比如 512 为 2 指数，而 528 则不是 2 的指数。该选项非必选，可以根据主控平台的特性进行选择，默认为选中。**注意**，如果主控平台不支持非 2 的指数长度，选中该选项会出现 SDIO 传输错误。

Use GPIO interrupt 是指通过 GPIO 作为 wlan 模块的数据中断，默认选中。如果不选中该选项，则会使用 SDIO 中断作为数据中断，但在大部分的主控平台中，SDIO 中断并不能作为系统的中断唤醒源，因此，如果要支持 wow 功能需要选中该选项。

Wlan Power off in suspend 是指在主控休眠时关闭 wifi 的电源，这种情况下，wifi 不再保持和 AP 的连接，只有在主控唤醒时会进行重新连接。对于一些在休眠期间不需要 wifi 保持连接和收发数据的方案平台，可以选中，达到降低休眠功耗的目的。默认为不选中。

Support extend suspend 该选项是针对 AllWinner 平台实现的，主控在休眠时可以根据不同用户场景选择让 wifi 进入掉电或者保持连接的状态。默认不选中，如果平台不支持 extend standby 请不要选中，否则可能导致编译失败。

Support normal suspend 该选项是针对主控向 WiFi 提供 24M 时钟，但主控休眠时无法保持 24M 时钟的情况下，又需要支持 wow 功能。默认不选中，该功能可能会增加主控的休眠功耗。**Allwinner 的平台不需要选中。**

```
-- XRadio Driver features
[*] SDIO bus wlan module
[*] Platform supports non-power-of-two SDIO transfer
[*] Use GPIO interrupt
[ ] Wlan power off in suspend
[ ] Support extend suspend(both Wow and power off)
[ ] Support normal suspend(Wow for Single Crystal)
Driver debug features --->
```

图 2-3 Driver 层编译选项

Driver debug features 主要为支持驱动层的测试和 debug 接口，具体如下。

debug messages 选项为驱动的打印输出，默认选中，能够在驱动出错时提供 log 信息。

ETF support for RF Test 选项为射频测试工具的提供接口，默认选中，关闭后会导致 RF Test APK, etf cli 等工具无法使用。

Dump kernel 是指在驱动出错时让内核报错并停止，默认不选中。

Expose driver debugfs 是指可以通过 debugfs 设备节点动态的获取更多驱动的 debug 信息，默认选中。

```
[*] Enable XRADIO debug messages (DEVELOPMENT)
[*] XRADIO ETF Support for RF Test(DEVELOPMENT)
[ ] Dump kernel in case of critical error (DEVELOPMENT)
[*] Expose driver internals to DebugFS (DEVELOPMENT)
```

图 2-4 Driver 层 debug 选项

2.2 Makefile 编译配置

在 xradio 的驱动目录中，有三个 Makefile，分别是 xradio/Makefile，xradio/umac/Makefile，xradio/wlan/Makefile。其中 xradio/Makefile 主要是为了包含子目录的 Makefile。

xradio/umac/Makefile 的配置选项基本都为内部配置使用，不完全了解的情况下不要做修改，否则可能导致 wifi 无法使用或出现异常。

对于 xradio/wlan/Makefile，包含一些可配置的选项，具体如下：

```
##===== User Options =====  
  
## Use vfs for firmware load when request_firmware  
  
# can't work on other platform.  
  
ccflags-y += -DUSE_VFS_FIRMWARE  
  
## Mac addr config, disable hex for default.  
  
#ccflags-y += -DXRADIO_MACPARAM_HEX  
  
ccflags-y += -DMONITOR_MODE
```

其中，USE_VFS_FIRMWARE 宏是指驱动通过 vfs 进行文件的读写，为了兼容 Android 和 linux，默认打开。XRADIO_MACPARAM_HEX 宏是指传入驱动的 MAC 地址参数是 16 进制的方式而非字符串，默认为字符串。MONITOR_MODE 宏是指驱动支持 monitor 模式，默认支持。

3 电源/时钟/中断

3.1 电源控制

3.1.1 供电（PMU）配置

一共需要两路电源，分别是 VDDIO 和 VBAT（包括 VBAT1 和 VBAT2）。VDDIO 供电电压为 3.3/1.8V，供电电流为 10mA 以上；VBAT 的供电电压为 2.7-5.5V，供电电流为 400mA 以上。

请根据硬件电路原理图，对相应的 LDO 进行配置，可以配置为 Always ON，或者打开 wifi 时才进行配置使能。对于 AllWinner 平台，需要在 dts 或者 sysconfig 中进行配置即可，对于其他平台需要对 xradio/wlan/platform.c 中的 xradio_wlan_power 函数进行重新实现。

如果需要使用 wow 功能，则在主控休眠时仍需保持电源供电。

3.1.2 Reset 配置

Reset 拉低时，芯片处于 reset 状态（即关闭状态）；Reset 拉高时，芯片处于上电状态（但不是 Active 状态），会有一定的功耗，具体功耗数据请查看 datasheet。

Reset 一般通过 GPIO 进行控制，根据硬件电路原图进行软件的配置。

如果需要使用 wow 功能，则在主控休眠时仍需保持 GPIO 的状态和供电。

3.1.3 上下电时序

芯片上电时，先使能电源供电，经过 50ms 稳定后，把 Reset 从低电平到高电平的变化。具体时序是：低电平（保持 2ms 以上），拉高并保持 50ms 后才能进行下一步操作。

芯片下电时，把 Reset 拉低，并关闭电源供电。伪代码如下：

```
int xradio_wlan_power(int on)
{
    if (on) { //上电
        //1. LDO ON
        //msleep(50);

        //2.Reset 拉低（如果状态默认为低电平可省略该步骤）
        //msleep(2);

        //3.Reset 拉高
        //msleep(50);
    } else { //下电
```

```
//1.Reset 拉低
//2.LDO OFF
}
}
```

3.2 时钟配置

3.2.1 高频（24M）时钟配置

通过电路图确认，WiFi 的 24M 时钟是由晶振提供，还是主控提供。如果为主控提供 24M 时钟，则需在 WiFi 打开前使能，并且如果支持 wow 功能需要在主控休眠期间保持 24M 时钟的提供。

提示：Allwinner 的平台无需软件做特别配置。

3.2.2 32K 时钟配置

通过电路图确认，主控有向 WiFi 提供 32K 时钟，并且在 WiFi 打开前使能。如果支持 wow 功能需要在主控休眠期间保持 32M 时钟的提供。

提示：Allwinner 的平台需通过 dts 或 sysconfig 进行 wlan 32K 时钟的配置。

3.3 中断配置

3.3.1 GPIO 中断配置

如果选择 GPIO 作为数据中断时，需要把内核配置的 Use GPIO interrupt 使能。对于非 Allwinner 平台，可能需要重新实现 xradio/wlan/platform.c 中的 xradio_request_gpio_irq 函数。

如果需要支持 WoW 功能的情况，则 GPIO 需配置为系统的中断唤醒源。对于 Allwinner 平台，通过 enable_irq_wake 函数使能，其他平台根据具体情况配置。代码参考示例：

```
static u32 wlan_irq_gpio = GPIOL(9); //GPIO pin 脚定义
static u32 gpio_irq_handle;
int xradio_request_gpio_irq(struct device *dev, void *sbus_priv)
{
    if (gpio_irq_handle)
        return 0;
    gpio_request(wlan_irq_gpio, "xradio_irq");
    gpio_direction_input(wlan_irq_gpio);
    gpio_irq_handle = gpio_to_irq(wlan_irq_gpio);
    ret = devm_request_irq(dev, gpio_irq_handle,
```



```
(irq_handler_t)xradio_gpio_irq_handler,
        IRQF_TRIGGER_RISING, "xradio_irq", sbus_priv);

if (IS_ERR_VALUE(ret)) {
    gpio_irq_handle = 0;
    return -1;
} else {
    enable_irq_wake(gpio_irq_handle);
}
return 0;
}
```

3.3.2 SDIO 中断配置

如果选择 GPIO 作为数据中断时，需要把内核配置的 Use GPIO interrupt 使能关闭。

4 SDIO 卡检测（扫卡）

xradio wlan 作为 SDIO slave 设备与主控进行通信。由于大部分的产品中，xradio 芯片会以 onboard 或者模组的形态固定在 PCB 上，不像 SD/TF 卡那样可以插入和拔出，系统的 mmc 驱动无法获知 SDIO slave 是否存在，进而无法进行卡的初始化和注销。

因此，需要有 xradio 驱动需要有一个接口通知 mmc 进行卡的初始化和注销。这个接口在 xradio/wlan/platform.c 中定义为 `int xradio_sdio_detect(int enable)`。当参数 enable 为真时，表示 xradio 需要进行 SDIO 卡初始化，反之表示需要进行 SDIO 卡注销。

由于不同的平台，具体的实现略有不同，因此可能需要重新实现。

5 MAC 地址

MAC 地址是 wlan 通信设备的标识，必须保持唯一性。如果在同一个局域网内存在相同 MAC 地址的设备，则该两个设备都将无法正常通信。在 xradio 驱动中，主要有三种方式配置 MAC 地址，具体如下。

获取 MAC 地址的优先级为：参数传递 > 文件保存（分区保存） > efuse 保存 > 随机生成。先通过高优先级的方式获取 MAC 地址，如果 MAC 地址不合法，则依次通过较低优先级的方式获取。

5.1 主控保存

这种方式，MAC 地址保存在主控端，在 wifi 启动时传递到 wifi 驱动。具体的传递方式也有三种。另外，MAC 地址可以以字符串的形式，也可以以 16 进制的形式，由 `XRADIO_MACPARAM_HEX` 宏确定。

5.1.1 参数传递

该方式是在 xradio 驱动加载时，通过模块参数的形式传入驱动。例如，可以通过字符串的形式（默认）

```
insmod xradio_core.ko macaddr=xx:xx:xx:xx:xx:xx
```

如果在驱动 Makefile 中定义了 `XRADIO_MACPARAM_HEX` 宏，则可以通过 16 进制的形式，例如，

```
insmod xradio_core.ko macaddr=0xDC,0x44,0x6D,0x00,0x00,0x00
```

5.1.2 文件保存

该方式是在 wifi 打开之前，把 mac 地址以字符串或者 16 进制的形式保存到指定的文件中。wifi 打开时，xradio 驱动在读取指定文件中的 mac 地址。指定文件的路径名在 `xradio/wlan/xradio.h` 文件中由 `WIFI_CONF_PATH` 宏定义，默认为 `/data/xr_wifi.conf`。

注意：在产品升级或者刷机过程中有可能被删除，导致 mac 地址丢失。

5.1.3 分区保存

由于文件保存存在丢失 mac 地址的风险，同时为了防止对 mac 地址非法的访问和改写，有些产品会把 mac 地址保存到一个特殊的分区，文件系统不可见。这种方式下，系统需向 xradio 驱动提供一个访问 mac 地址的接口，并在 `xradio/wlan/main.c` 的 `xradio_get_mac_addrs` 函数加入相关实现。

5.2 efuse 保存

XR829 芯片提供了 efuse 空间用于保存产品的 mac 地址，可以在产品的生产环节中烧录到 xradio 芯片，具体烧录的方法请查看 datasheet。此后，在 wifi 启动时会从 efuse 中读取 mac 地址，主控不需要做任何配置。

5.3 随机生成

当前面所有方式都无法获取到合法的 mac 地址后，xradio 驱动会随机生成一个合法的 mac 地址，并保存在指定的文件中，以便下次启动时使用，指定的文件路径名由 WIFI_CONF_PATH 宏定义。

注意：由于该方式获取的 mac 地址并不是从 mac 地址管理机构购买，因此一般只作为临时调试使用。

6 休眠唤醒

6.1 WoW 功能

WoW 功能的全称为 wake on wlan，是指在主控休眠时保持 wifi 连接，并且在 wifi 收到用户数据时唤醒主控，并把该数据传到主控。典型的场景，在主控休眠时，仍希望能收到 QQ，微信等发过来的消息。

对于支持 WoW 功能，除了正确的内核配置以外（请参考 2.1 内核配置），还有以下几个条件：

1. 在主控休眠期间需保持对 wifi 的供电，包括 VDDIO 和 VBAT。
2. 在主控休眠期间需保持对 wifi 的提供 24M 时钟。
3. 在主控休眠期间需保持对 wifi 的提供 32K 时钟。
4. 数据中断需支持唤醒主控。

6.2 休眠 WiFi 掉电

该模式和 WoW 功能相反，表示在主控休眠时不再保持 wifi 连接，wifi 处于掉电状态，此时 wifi 不能再接收任何的数据。对于主控休眠时不希望被唤醒或不需接收数据时，可以选择该模式以降低产品休眠功耗。通过内核配置中选中 Wlan Power off in suspend 使能该模式（具体参考 2.1 内核配置）。

6.3 Extern Standby

该模式为 AllWinner 平台特有的模式，该模式同时兼容 WoW 和休眠 WiFi 掉电模式，表示主控可以根据不同的用户场景决定进入 WoW 模式还是进入休眠 WiFi 掉电模式。

当主控休眠时有网络数据需求，可以通过锁住 wlan standby，让 wifi 进入 WoW 模式；当主控休眠时不需要收发数据，则可以释放 wlan standby，让 wifi 进入休眠 WiFi 掉电模式。（具体使用方法请参考 AllWinner 主控 SDK 相关文档）

7 其他配置

7.1 驱动移植/更新

驱动移植具体参考《XR829 WLAN Porting Guide》。

对于驱动更新，一般不需要更新 wlan/platform.h 和 wlan/platform.c，因为 platform.h 和 platform.c 主要为平台相关的代码，驱动移植时会修改以适应不同的平台，所以在驱动更新时不需要更新这两个文件。

7.2 查看/修改高频时钟

XR829 芯片的高频时钟默认使用 24M，当有需要时可以通过修改 sdd 配置文件使用其它频率的高频时钟。打开 Tools\WLAN sdd editor\sddedit.exe 工具，如下图 7-1 所示，通过 Open SDD 按钮选择打开 android/hardware/xradio/wlan/kernel-firmware/sdd_xr829.bin 文件（请注意选择文件的后缀为*.bin 或者把 sdd_xr829.bin 改为 sdd_xr829.sdd）。然后通过 Reference Clock Frequency 的下拉菜单选择需要的频率，然后点击 OK 按钮进行保存。把修改后的 sdd 文件更新到设备后重启 wifi 即可。

注意：如果高频时钟配置与实际频率不符会导致扫描不到 AP，以及收发功能。

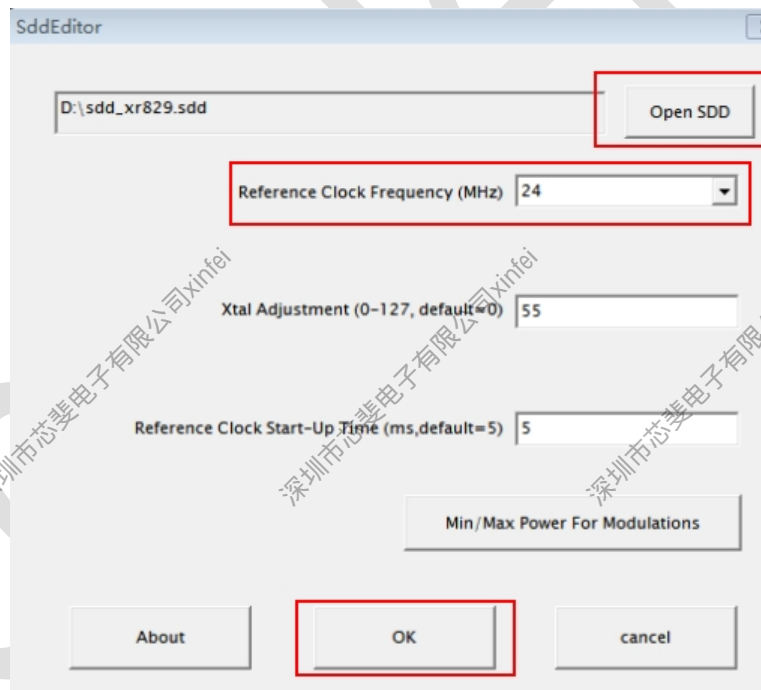


图 7-1 查看/修改高频时钟

7.3 修改最大发射功率

为了满足某些测试需求，需要对最大发射功率进行调整，但强烈建议不要輕易的修改发射功率，否则

可能导致性能较差或者无法通过认证机构的认证。具体修改方法如下图 7-2 所示，打开 `sdd_xr829.bin` 文件，点击 `Min/Max Power For Modulations`，会弹出图 7-3 的窗口，根据需要修改各速率的发射功率后点击 `OK` 按钮返回第一级界面，继续点击 `OK` 按钮保存修改到 `sdd_xr829.bin` 文件，然后把 `sdd_xr829.bin` 文件更新到设备即可。

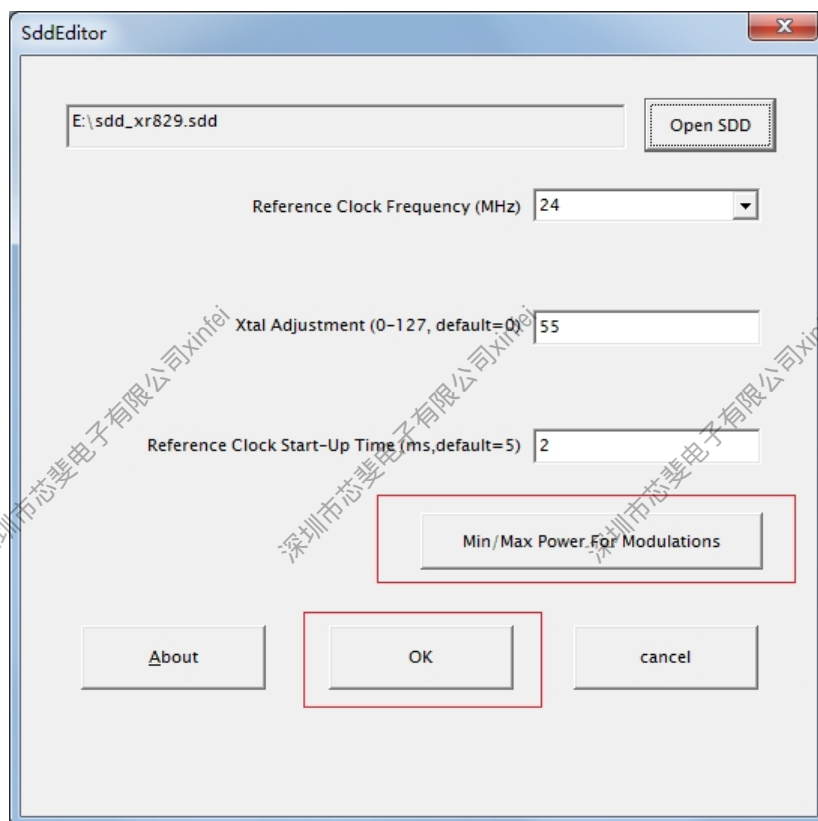


图 7-2 修改最大发射功率 1

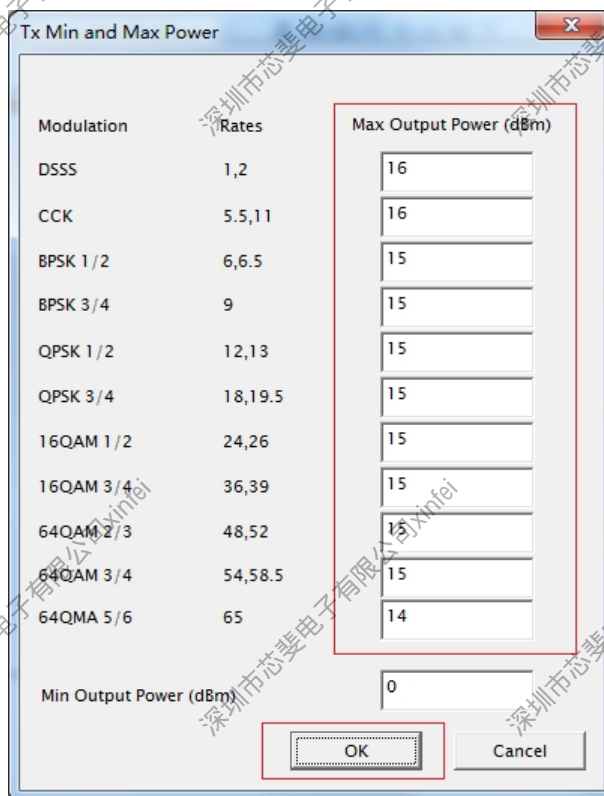


图 7-3 修改最大发射功率 2

7.4 Monitor 模式

7.4.1 iw 工具准备

确保系统中存在 iw 工具，如果没有，则网上下载源码，将整个 iw 文件夹放到 android\external 目录进行编译，编译完后把 iw 可执行文件推进小机的 system/bin 目录，或者重新刷机。建议开始调试时，先使用 adb push 的方式，这样方便调试，稳定后再进行刷机，否则调试效率较低。

7.4.2 Monitor 模式开启

```
insmod /system/vendor/modules/xradio_wlan.ko //确认 WiFi 关闭的情况下
iw wlan0 set type monitor //把 wlan0 接口配置为 monitor 模式
ifconfig wlan0 up //开启 monitor 接口
iw wlan0 set channel 6 //配置 monitor 的监听信道
```

7.4.3 Monitor 模式调试

如果需要查看 monitor 模式下接收到的包，可以通过以下命令进行查看，输出到内核打印。

```
echo 0xffff,0xffff > /sys/kernel/debug/ieee80211/phy*/ xradio/parse_flags
```

7.4.4 Monitor 模式关闭

```
ifconfig wlan0 down      //关闭 monitor 接口
iw wlan0 set type managed //把 wlan0 接口配置为 managed 模式
rmmod xradio_wlan.ko     //卸载 WiFi 驱动
```

7.5 ACS 功能

略。