

WS73V100 特性开发

说明书

文档版本 02
发布日期 2024-10-08

前言

概述

本文档详细的描述了 WS73V100 相关特性的应用场景、实现原理及接口说明，方便读者了解并使用相关特性。

读者对象

本文档主要适用于以下工程师：

- 软件开发工程师
- 技术支持工程师
- 硬件开发工程师





产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
WS73	V100

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	表示如不可避免则将会导致死亡或严重伤害的具有高等级风险的危害。
 警告	表示如不可避免则可能导致死亡或严重伤害的具有中等级风险的危害。
 注意	表示如不可避免则可能导致轻微或中度伤害的具有低等级风险的危害。
须知	用于传递设备或环境安全警示信息。如不可避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
 说明	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

修改记录

文档版本	发布日期	修改说明
02	2024-10-08	更新“3.3 实现原理”小节内容。
01	2023-12-14	第一次正式版本发布。 更新“3.3 实现原理”小节内容。
00B01	2023-11-28	第一次临时版本发布。

目 录

前言	i
1 混杂模式特性说明	1
1.1 概述	1
1.2 应用场景	1
1.3 实现原理	2
1.4 接口说明	3
1.5 使用示例	4
2 动态国家码特性说明	5
2.1 概述	5
2.2 应用场景	5
2.3 实现原理	6
2.4 接口说明	7
2.5 使用示例	7
3 中继特性说明	9
3.1 概述	9
3.2 应用场景	9
3.3 实现原理	10
3.4 接口说明	12
3.5 使用示例	12
4 开机校准特性说明	14
4.1 概述	14
4.2 应用场景	14
4.3 实现原理	15

4.4 接口说明	19
5 wapi 特性说明	21
5.1 概述	21
5.2 应用场景	22
5.3 实现原理及使用方法	22

1

混杂模式特性说明

1.1 概述

1.2 应用场景

1.3 实现原理

1.4 接口说明

1.5 使用示例

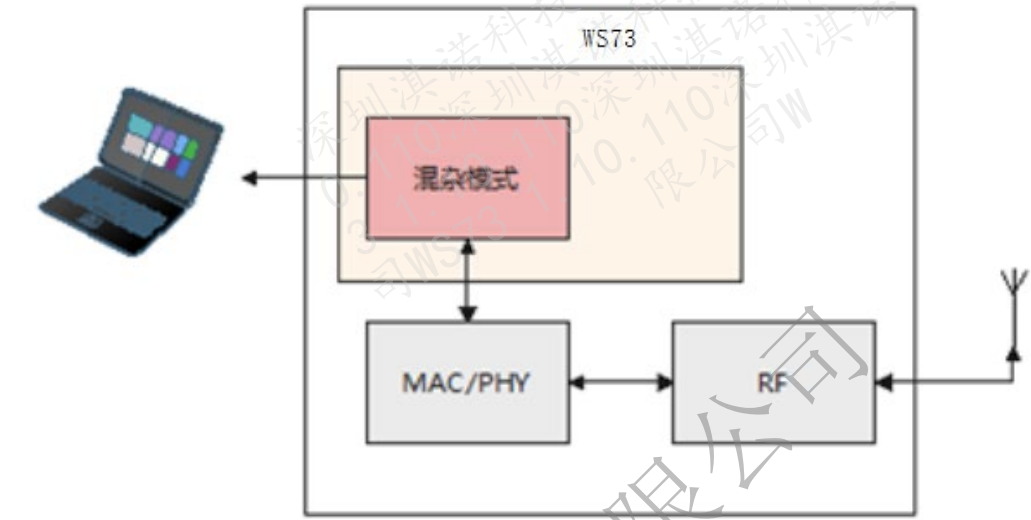
1.1 概述

混杂模式特性用于抓取所有经过本设备的管理帧/数据帧，并将抓取到的报文保存到文件中，可通过抓包软件例如：WireShark、OmniPeek 打开并查看报文信息。

1.2 应用场景

WiFi 混杂模式开启后，会抓取周边 AP 和 STA 之间的单播/组播的管理帧/数据帧，典型应用场景之一是支持作为抓包网卡。如下图，WS73 芯片作为抓包网卡，通过串口与 PC 连接，抓取 PC 周边报文。

图1-1 混杂模式进行空口抓包

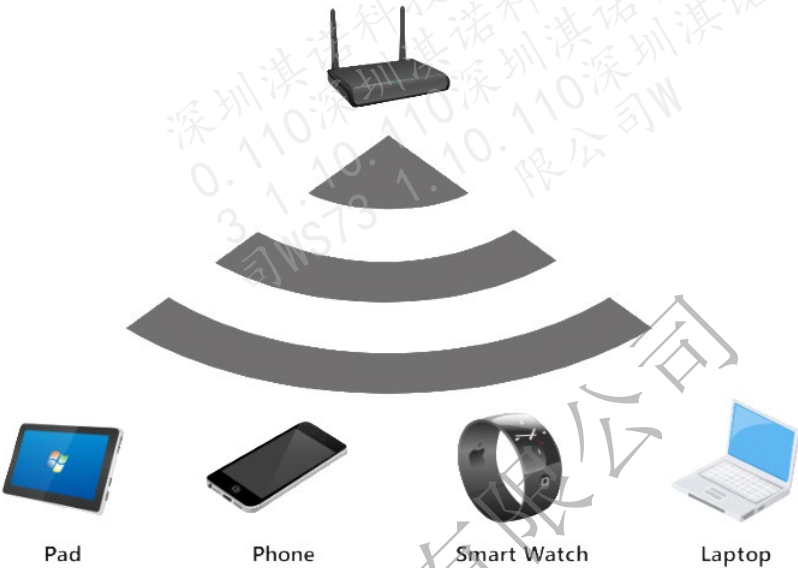


1.3 实现原理

无线网络信号在传播过程中是以发射点为中心，像波纹一样往外辐射。理论上讲，如果一个接收器处于无线信号经过的地方，它可以“听到”任何经过它的信号，只是它可能“听不懂”（无法解析报文内容）。

以下图为例，Phone 与 Smart Watch 通信，Laptop 完全有能力从空口监听他们的通信。空口抓包就是基于这个原理工作的。如果我们想要抓某个嵌入式设备的无线报文，只需在它附近运行一个具有监听功能的无线设备。

图1-2 混杂模式



1.4 接口说明

命令格式: echo “\$vap sniffer_save_file \$switch \$file_num \$file_size” > /sys/ccsys/ccpriv

参数说明:

- \$vap: 表示需要维测的 vap 名字, 通常为 wlan0。
- \$switch: 表示功能开、关、或者暂停, 对应 1、0、2。
- \$file_num: 表示使用的抓包文件个数, 取值范围为 1~15, 单位个。
- \$file_size: 表示每个抓包文件的大小, 取值范围为 1~50, 单位 MB。

注意事项: \$file_num * \$file_size 的取值范围为 1~pcap_file_len_max, pcap_file_len_max 可在 ws73_cfg.ini 配置文件中配置。

命令示例:

- 开启抓包: echo “wlan0 sniffer_save_file 1 2 5” > /sys/ccsys/ccpriv, 命令下发成功后, 会立即创建 5 个 proc 文件, 保存在内存中, 使用前 2 个 proc 文件进行抓包, 每个抓包文件大小为 5MB
- 暂停抓包: echo “wlan0 sniffer_save_file 2” > /sys/ccsys/ccpriv, 命令下发成功后, 会停止抓包但不会释放内存, 此时可将抓包文件取出。

- 关闭抓包：echo "wlan0 sniffer_save_file 0" > /sys/ccsys/ccpriv, 命令下发成功后，会关闭抓包并释放内存。

1.5 使用示例

步骤 1 执行 1.4 接口说明相关命令后，在 /proc 目录下获取抓包文件 wifisniffer_01。

步骤 2 用 tftp 命令将抓包文件传到 PC 上

```
tftp -gr /proc/wifisniffer_01 192.168.1.15
```

步骤 3 将抓包文件 wifisniffer_01 的后缀改为.pcap 格式，使用抓包工具如 WireShark 查看该抓包文件即可。

----结束

2

动态国家码特性说明

2.1 概述

2.2 应用场景

2.3 实现原理

2.4 接口说明

2.5 使用示例

2.1 概述

动态国家码特性用于支持全球发货场景，根据设备预制的国家信息，自动调整发射功率表，以符合全球各地区对发射功率的法律规范。

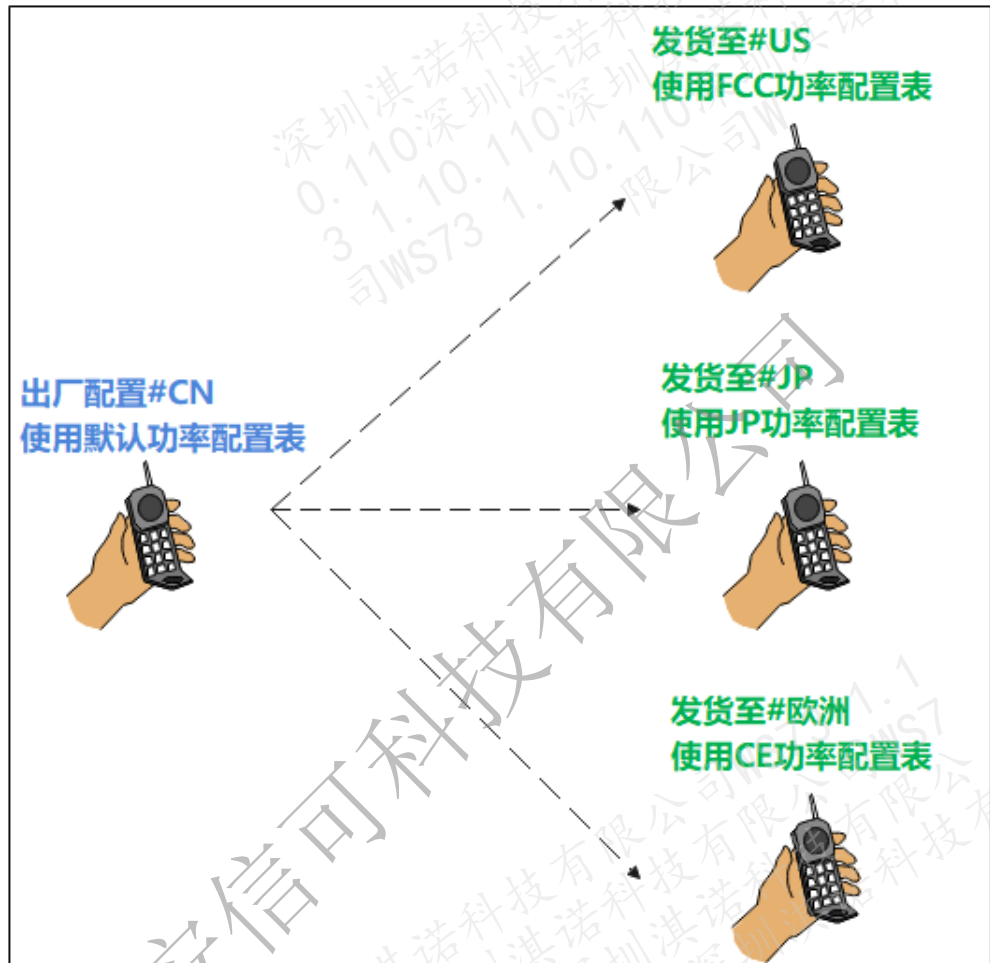
2.2 应用场景

在上网问题处理中，经常会碰到来自异国的设备出现扫描/连接/协商速率异常的情况，很多情况与 802.11d 协议（/国家码设置）相关。

国家码用来标识无线设备所在的国家，不同国家码规定了不同的无线设备射频特性，包括 AP 的发送功率、支持的信道等。配置国家码是为了使无线设备的射频特性符合不同国家或区域的法律法规要求。在第一次配置 WLAN 设备时，必须配置正确的国家码，以确保不违反当地的法律法规。

动态国家码提供一种配置方式，客户能够通过设置国家码，调整到国家对应大区（中国、亚太、北美、欧洲）的发射功率，从而符合法律规范

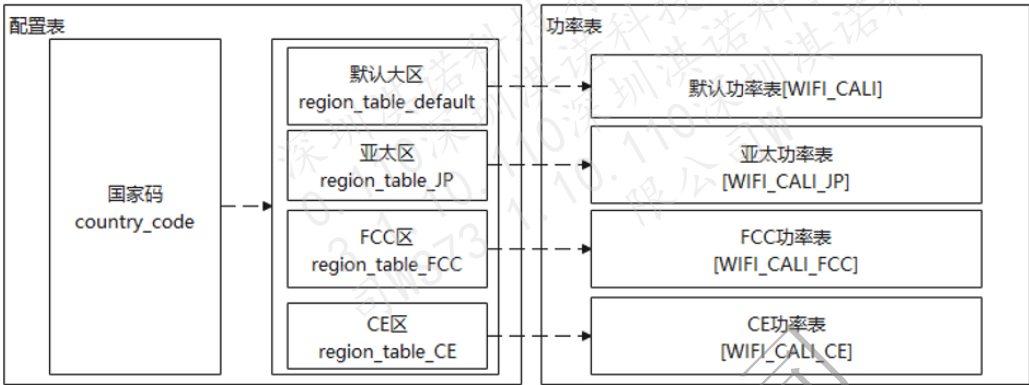
图2-1 动态国家码应用场景



2.3 实现原理

通过配置文件映射国家与大区的关系，每个国家码对应一个大区，每个大区对应一套功率表，国家码变动时，根据重新配置对应的功率表。

图2-2 动态国家码设计原理图



2.4 接口说明

设置/读取国家码

```
echo "Featureid0 setcountry $COUNTRY" > /sys/ccsys/ccpriv  
echo "Featureid0 getcountry" > /sys/ccsys/ccpriv
```

说明

- Featureid 端口是 Linux 内核中的一个虚拟端口，用于向用户空间提供有关系统功能的信息。它可以用于获取系统支持的功能列表、功能状态和相关的统计数据。
- \$COUNTRY：国家码，可配置范围：
CN,JP,US,CA,KHRU,AU,MY,ID,TR,PL,FR,PT,IT,DE,ES,AR,ZA,MA,PH,TH,GB,CO,MX,EC,
PE,CL,SA,EG,AE.

2.5 使用示例

加载驱动时，根据 ini 配置文件中 country_code 变量，决定配置的国家码。确认国家码后，选择对应的 region_table 并配置大区功率，如图 2-3 所示。

图2-3 ini 文件国家码配置示例

```
#1、var=value, 等号左右没有空格
[HOST_WIFI_NORMAL]
#私有定制化--begin
#最大带宽能力:FPGA 40M ASIC 160M (0:20M,1:40M,2:80M,3:160M)
bw_max_width=1
#txbf-cap
su_bfee=1
ldpc=1
ba_32bitmap=0
mtid_aggr_rx=0 国家
country_code=CN 大区
region_table_default=CN
region_table_jp=JP
region_table_fcc=US,CA,KH
region_table_ce=RU,AU,MY,ID,TR,PL,FR,PT,IT,DE,ES,AR,ZA,MA,PH,TH,GB,CO,MX,EC,PE,CL,SA,EG,AE
```

3

中继特性说明

3.1 概述

3.2 应用场景

3.3 实现原理

3.4 接口说明

3.5 使用示例

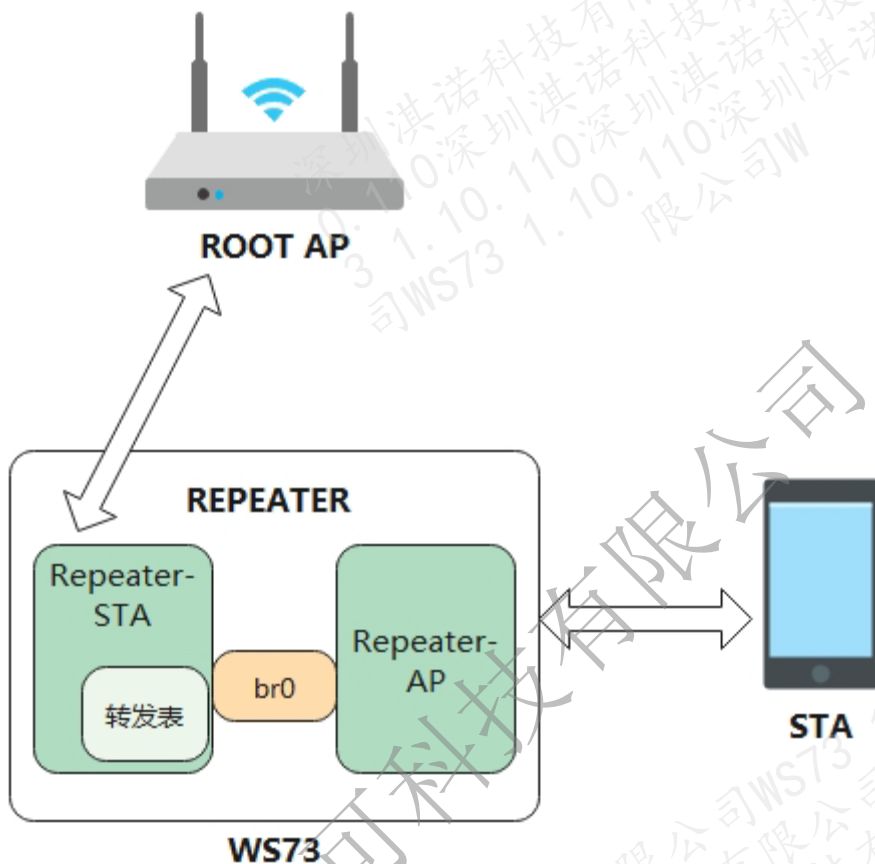
3.1 概述

中继特性主要用于无线网络连接中，通过对报文的转发，实现远距离无线通信，达到扩大网络覆盖范围、降低网络部署成本的目的。

3.2 应用场景

中继特性通过创建一个 Repeater-STA 端口与 ROOT AP 建立连接，创建一个 Repeater-AP 端口为其它 STA 提供服务，其工作原理为：将 Repeater-STA 和 Repeater-AP 对应的接口分别加入到网桥 br0 中。Repeater-STA 与 Repeater-AP 之间的报文交互通过网桥 br0 进行，从而进一步实现 ROOT AP 与 STA 的报文交互、完成网络业务的功能。Repeater-STA 支持与一个 ROOT AP 接入，Repeater-AP 支持最多 7 个设备同时接入。

图3-1 中继场景



3.3 实现原理

STA 发给 ROOT AP 的报文经过 Repeater 时，报文中的源 MAC 地址与源 IP 地址的映射关系会被记录进入转发表，并且将报文中的源 MAC 修改为 Repeater-STA 的 MAC 发送给 ROOT AP。

ROOT AP 发送给 STA 的报文经过 Repeater 时，会根据目的 IP 在转发表中查询，更改报文中的目的 MAC 地址，将报文转发到正确的 STA 中。

图3-2 Repeater-STA 上行数据场景

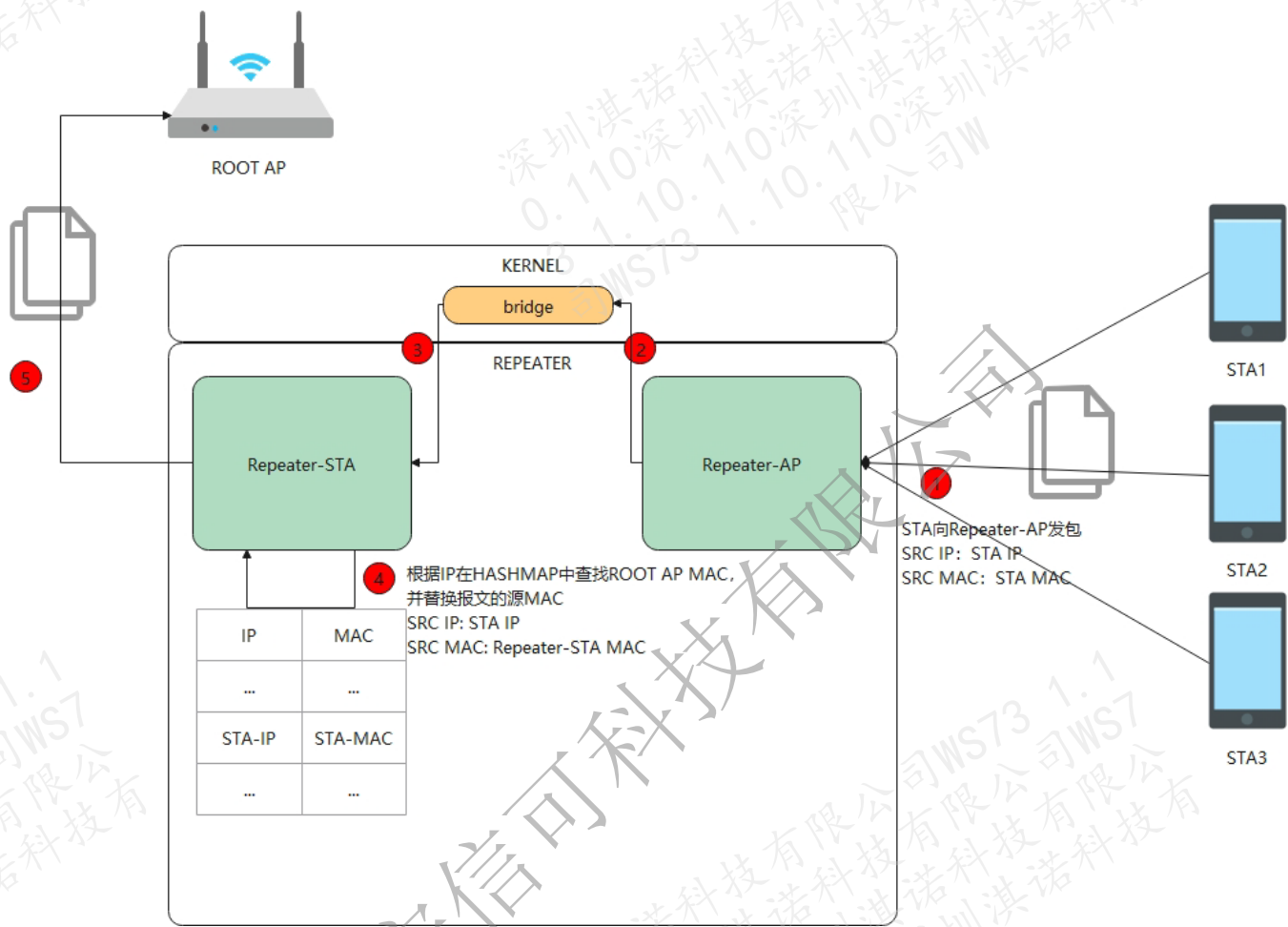
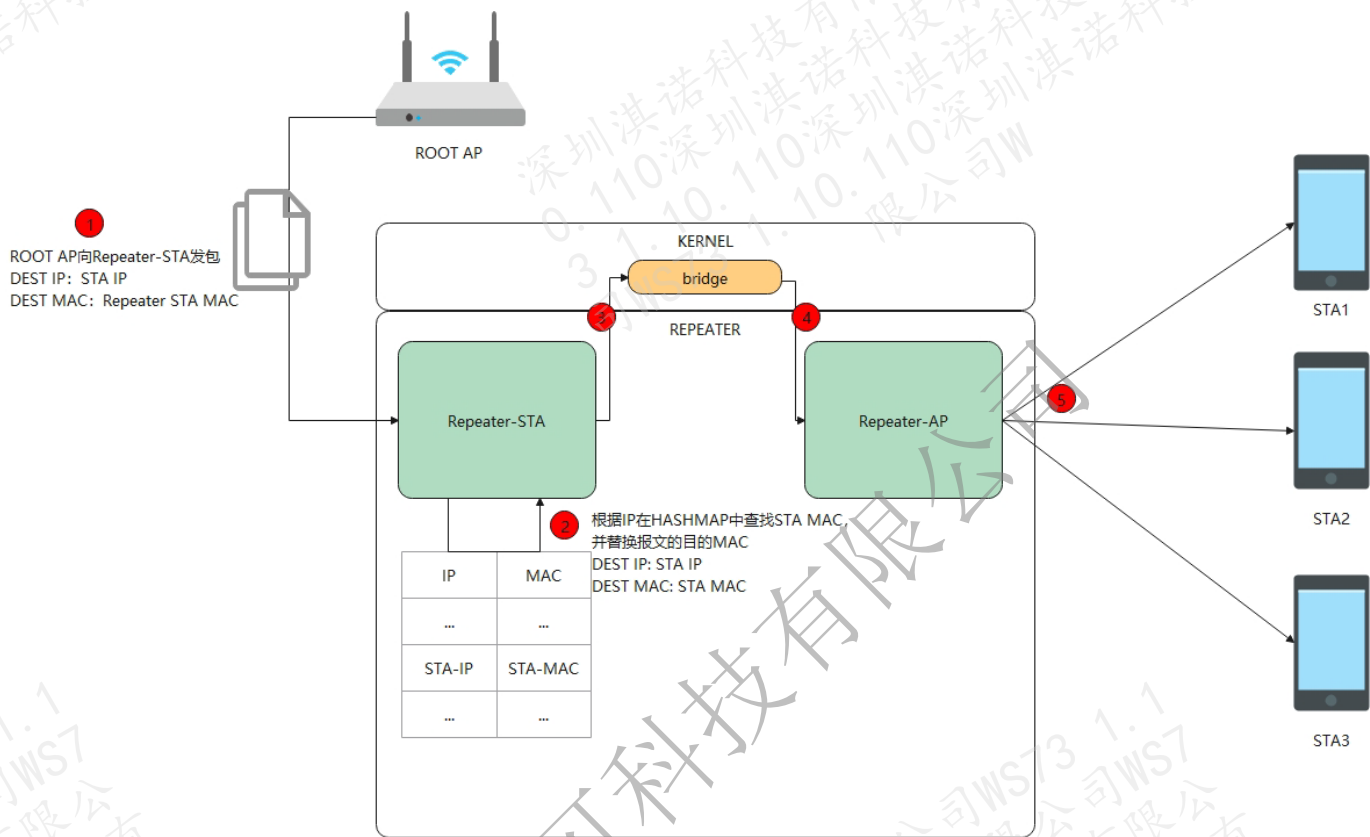


图3-3 Repeater-STA 下行数据场景



使用中继功能需要内核开启网桥功能，进入内核源码目录，执行 make menuconfig 命令，进入 Networking support->Networking options，选中 “802.1d Ethernet Bridging”。

3.4 接口说明

暂无

3.5 使用示例

步骤 1 创建 wlan1 端口（默认情况下 wlan0 端口已创建）。

```
echo "Featureid0 create wlan1 ap" > /sys/ccsys/ccpriv
```

步骤 2 创建网桥。

```
brctl addbr br0
```

步骤 3 将 wlan0 端口、wlan1 端口分别 up。

```
ifconfig wlan0 up  
ifconfig wlan1 up
```

步骤 4 将 Repeater-STA、Repeater-AP 分别加入网桥。

```
brctl addif br0 wlan0  
brctl addif br0 wlan1
```

步骤 5 网桥分配地址。

1. 对于静态 Repeater, 需要给 br0 配置一个与 ROOT AP 同网段的静态 IP 地址。

```
ifconfig br0 192.168.100.20
```

2. 对于动态 Repeater, 需要通过 DHCP 服务向 ROOT AP 申请 IP 地址。

```
udhcpc -i br0
```

步骤 6 启动 Repeater-STA 并关联 ROOT AP。

例: 关联 ROOT AP 的 SSID 为 "repeater_test", OPEN 模式。

```
wpa_supplicant -iwlan0 -Dnl80211 -c /etc/Wireless/wpa_supplicant.conf -bbr0 &  
wpa_cli -iwlan0 -p /etc/Wireless/wpa_supplicant  
add_network  
set_network 0 ssid "repeater_test"  
set_network 0 key_mgmt NONE  
select_network 0
```

步骤 7 启动 Repeater-AP。

注意: 启动前请检查 hostapd.conf 中默认端口为 wlan1。

```
hostapd /etc/Wireless/hostapd.conf &
```

步骤 8 测试完成, 将 wlan0 端口、wlan1 端口从网桥拆除, 将网桥 br0 设置为 down 并删除。

```
brctl delif br0 wlan0  
brctl delif br0 wlan1  
ifconfig br0 down  
brctl delbr br0
```

----结束

4

开机校准特性说明

4.1 概述

4.2 应用场景

4.3 实现原理

4.4 接口说明

4.1 概述

开机校准是指开机过程中对射频参数进行自调整以提升射频收发性能的功能。

4.2 应用场景

开机校准在单板上电启动过程中执行，常用于消减不同单板以及不同环境之间的性能差异。这里不同单板常指制作批次不同，使用器件有差异等；不同环境常指外部温湿度差异，存在干扰源等。

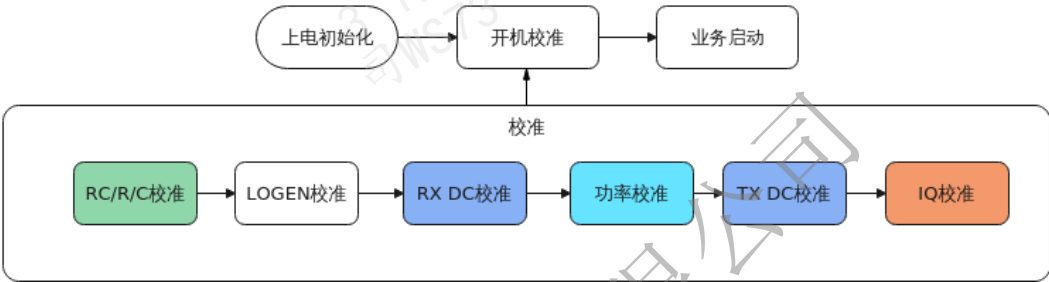
须知

上电前，务必保证射频口接有 50Ω负载，否则可能导致校准功率偏差，影响整体校准性能。

4.3 实现原理

开机校准支持多个校准项，根据不同的校准用途，以滤波器校准，功率校准，直流校准，IQ 不平衡校准四个部分来进行原理说明。

图4-1 开机校准流程

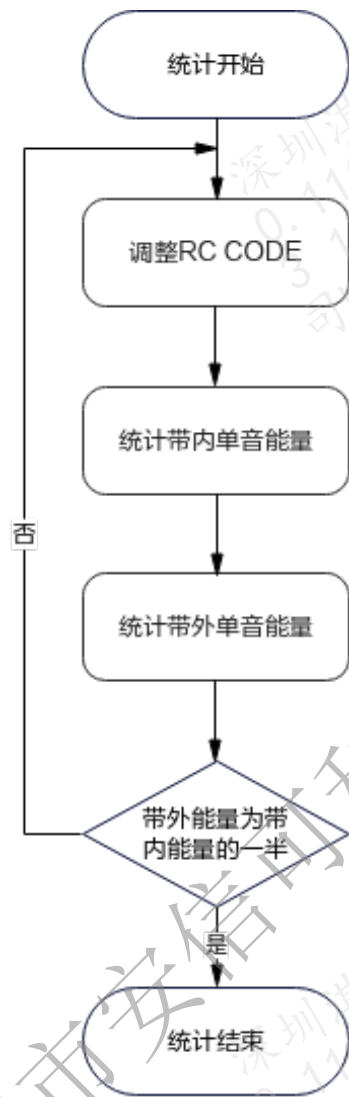


1. 滤波器校准

本校准用于对齐滤波器带宽，可用于滤波性能提升，包括 RC 校准、R 校准、C 校准。

校准时需要调节滤波器预留的电容，电阻配置。通过不断调整配置使得滤波器 3dB 带宽位置的能量比中心低 3dB，从而得到满足滤波器设计要求的配置。

图4-2 滤波器校准流程



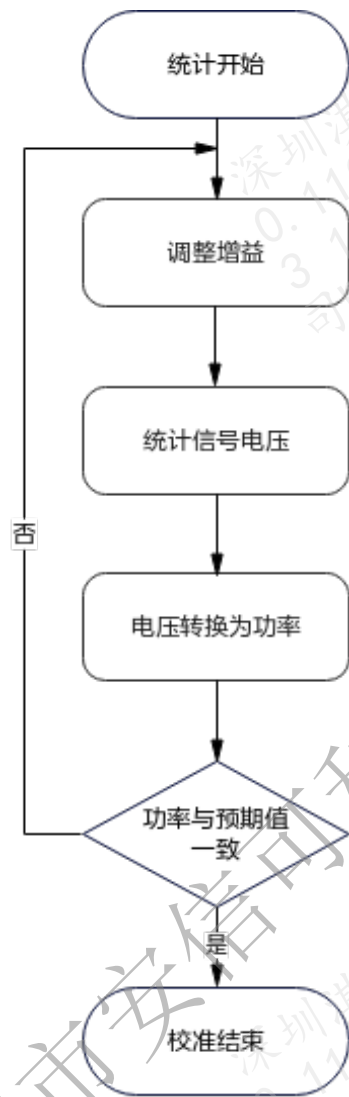
2. 功率校准

本校准用于对齐输出功率，可提升功率精度，包括功率校准。

在对齐输出功率前，首先会对发送通路的电容进行调节，使得电路满足阻抗匹配，从而保障发射通路的高效运行，即同射频增益下输出功率可以更高。

对齐输出功率是指默认的射频增益配置由于单板差异等导致实际输出功率与预期有偏差时重新调整射频增益的过程。射频预留了可调节增益配置，通过调整增益配置，可以使得发送通路中信号的能量值发生变化。在校准中选择固定的输出功率点的能量值作为参考点，调节增益配置，使得获取的能量与目标功率点的能量值最接近，则保存对应配置作为指定功率点的增益配置更新值。接着根据不同功率的增益差，将指定功率点的增益配置同步到全功率范围，从而消减不同单板差异引入的输出功率差异，保证功率精度满足要求。

图4-3 功率校准流程

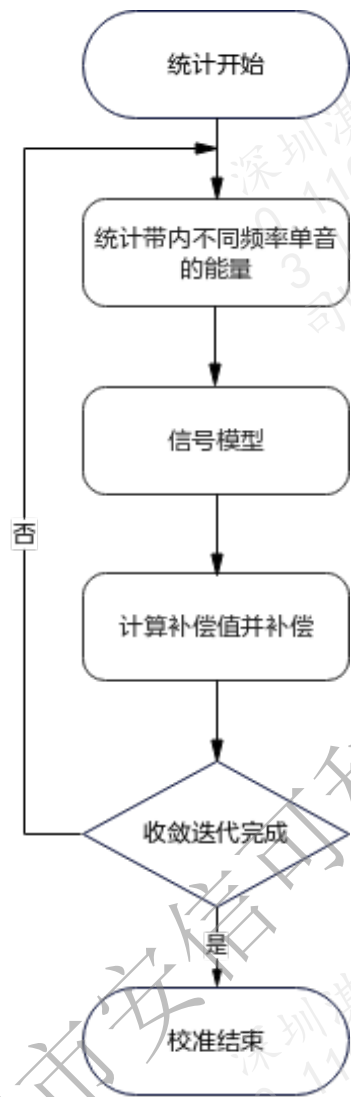


3. 直流校准

本校准用于消减射频收发通路中存在的直流偏置，可提升信噪比，保证接收能力，包括 RX DC 校准和 TX DC 校准。

TX DC 和 RX DC 校准分别用于发送和接收通路的直流偏置消减，实现原理一致。通过对收发通路中的直流偏置进行分段处理，即根据放大器件将通路分割，调整放大器件的增益可以得到不同增益下，特定通路位置的直流偏置值，从而计算获得在不同通路位置上残留的直流偏置值，进行反向补偿可以消减直流偏置。

图4-4 直流校准流程

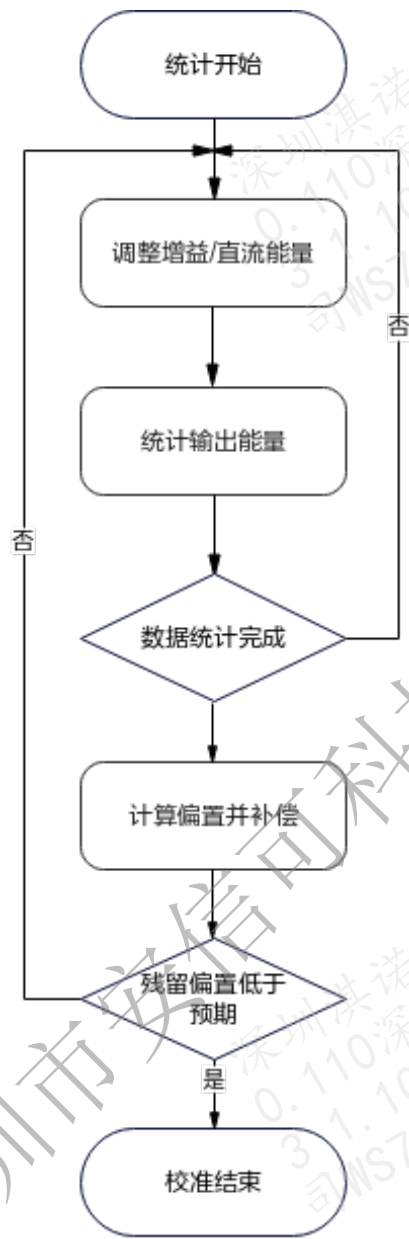


4. IQ 不平衡校准

本校准用于消减分解的正交信号的失衡（幅度失衡，相位失衡），可提升信号质量，包括 TX IQ 校准，RX IQ 校准。

TX IQ 和 RX IQ 校准分别用于发送和接收通路的失衡问题优化，实现原理一致。通过建模将相位失衡作为移相器的误差，幅度失衡则作为混频器的误差，根据移相前后的模型获得相位失衡和幅度失衡值，在收发信号时反向补偿来消减失衡，从而减小由于失衡引入的镜像信号的干扰，提升调制和解调能力。

图4-5 IQ 不平衡校准



4.4 接口说明

支持通过修改配置文件来开关每个校准项，配置参数 cali_mask 字段。该配置项为 16 位数，每一位表示一个校准功能的开关，即 0 表示关闭，1 表示开启。

图4-6 校准参数示例

```
#校准开关
#bit[15:8] RX_GAIN | DPD_CALI | DPD_COMP_40M | RXIQ | TXIQ | TXDC | TX_PWR | RXDC
#bit[7:0] PA_ICAL | LODIV | LOGEN | PPF | C | R | RC | ABB
cali_mask=0x1FAE
```

说明

默认开启的校准项目有 RC 校准 (bit1), R 校准 (bit2), C 校准 (bit3), LOGEN 校准 (bit5), iPA 电流校准 (bit7), RX DC 校准 (bit8), 功率校准 (bit9), TX DC 校准 (bit10), TX IQ 校准 (bit11), RX IQ 校准 (bit12), 即配置 cali_mask 参数为 0x1FAE。如需修改校准项, 请咨询开发人员。

5 wapi 特性说明

5.1 概述

5.2 应用场景

5.3 实现原理及使用方法

5.1 概述

WAPI (Wireless LAN Authentication and Privacy Infrastructure) 是无线局域网鉴别和保密基础结构, 是一种安全协议。当前全球无线局域网领域仅有的两个标准, 分别是美国行业标准组织提出的 IEEE 802.11 系列标准(包括 802.11a/b/g/n/ac 等), 以及中国提出的 WAPI 标准。WAPI 是我国首个在计算机宽带无线网络通信领域自主创新并拥有知识产权的安全接入技术标准。

本方案已由国际标准化组织 ISO/IEC 授权的机构 IEEE Registration Authority (IEEE 注册权威机构) 正式批准发布, 分配了用于 WAPI 协议的以太类型字段, 这也是中国在该领域唯一获得批准的协议。

WAPI 同时也是中国无线局域网强制性标准中的安全机制。

与 WIFI 的单向加密认证不同, WAPI 双向均认证, 从而保证传输的安全性。WAPI 安全系统采用公钥密码技术, 鉴权服务器 AS 负责证书的颁发、验证与吊销等, 无线客户端与无线接入点 AP 上都安装有 AS 颁发的公钥证书, 作为自己的数字身份凭证。当无线客户端登录至无线接入点 AP 时, 在访问网络之前必须通过鉴别服务器 AS 对双方进行身份验证。根据验证的结果, 持有合法证书的移动终端才能接入持有合法证书的无线接入点 AP。

无线局域网鉴别与保密基础结构 (WAPI) 系统中包含以下部分:

1、WAI 鉴别及密钥管理

2、WPI 数据传输保护

无线局域网保密基础结构 (WPI) 对 MAC 子层的 MPDU 进行加、解密处理, 分别用于 WLAN 设备的数字证书、密钥协商和传输数据的加解密, 从而实现设备的身份鉴别、链路验证、访问控制和用户信息在无线传输状态下的加密保护。

WAPI 无线局域网鉴别基础结构 (WAI) 不仅具有更加安全的鉴别机制、更加灵活的密钥管理技术, 而且实现了整个基础网络的集中用户管理。从而满足更多用户和更复杂的安全性要求。

在我们的项目中, WAPI 协议实现软件是 WS73 WiFi 软件中的一部分, WAPI 的主要流程部署在 host 侧, 其中包括最主要的 wapi 的数据加解密处理。WAI 相关的认证处理部署在 wpa_supplicant 中实现。除此之外, dev 在收到 wapi 数据之后, 将数据抛给 host 处理

5.2 应用场景

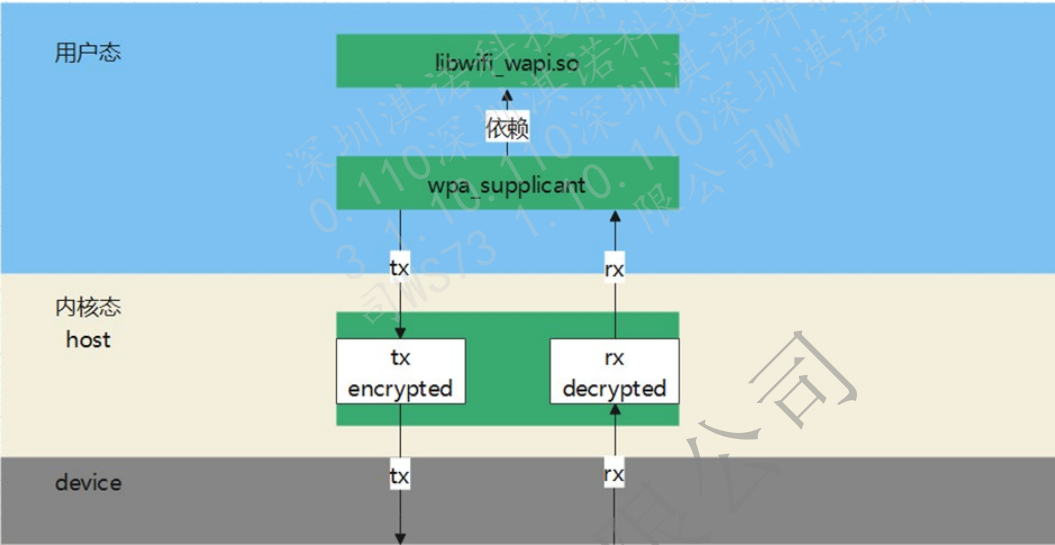
WAPI 可以理解为一种新的, 不同于 WiFi 802.11i 的加密协议, 它需要对端与端侧均支持 WAPI 时才能使用, 它是纯软件的处理, 不依赖硬件升级, 而由于美国的打压, 实际上 WAPI 一直处于无线网络加密的边缘地带, 使用的场景特别少, 目前主要是认证时会涉及。

当对端路由器配置为 wapi 加密时, WS73 作为 Sta 侧设备, 必须支持 wapi 加密才能正常接入并通信

5.3 实现原理及使用方法

wapi 的整体原理比较复杂, 因为篇幅原因这里不详细展开, 一句话总结的话就是 wapi 是不同于传统硬件加密方式的软件加密算法, 它完全不依赖硬件设备, 可以完全通过软件来实现。

图5-1 wapi 整体结构



1. `wpa_supplicant` 中的相关内容，需要合入 `wpa_supplicant_2_10_linux.patch`(位于 `sdk` 中 `open_source/wpa_supplicant/wpa_supplicant_2_10_linux.patch`)，并且在 `wpa_supplicant` 编译中打开 `WPAI` 宏 `CONFIG_WAPI` 即可
2. `libwifi_wapi.so`，因为其对应的源码不能开源，如果确实需要 `wapi` 功能，需要将主控芯片对应的编译工具链（如 `gcc`）提供给我们，我们编译出 `so` 库之后即可
3. `patch` 文件中的 3830 行指定了 `so` 库在单板上对应的位置，可以根据实际情况更改

```
3830  +#if defined( __LP64__ )
3831  +#define LIBWAPI_PATH "/vendor/lib64/libwifi_wapi.so"
3832  +#else
3833  +#define LIBWAPI_PATH "/lib/libwifi_wapi.so"
3834  +#endif
```

4. 上板使用，当启动 WAPI 宏打开的 wpa_supplicant 时，wpa_supplicant 会在启动阶段就去寻找对应的 so 库，如果找到，会出现如下打印：

```
/komod #
/komod #
/komod #
/komod # wpa_supplicant -iwlan0 -Dnl80211 -c/etc/wireless/wpa_supplicant.conf &
/komod #
/komod # Successfully initialized wpa_supplicant
rfkill: Cannot open RFKILL control device
wlanetdev_open_etc.devname is:wlan0
plat_soc:E[wlan_power_open_cmd]plat_soc:E]==wlan READY==
hwifi_custom_adapt_mac_device_priv_ini_param::ldpc[1]
hwifi_custom_adapt_mac_device_priv_ini_param::front_switch[0].
hwifi_custom_adapt_device_priv_ini_cal_mask_param::read cali_mask[8110]ret[0]
hwifi_custom_adapt_mac_device_priv_ini_param::g_uc_wlan_open_cnt[2]priv_cali_data_up_down[0x10]
hwifi_custom_adapt_device_priv_ini_param::g_uc_custom_cali_done_etc[1]auto_cali_mask[0x0]
hwifi_custom_adapt_device_priv_ini_param::data_len[46]
***hwifi_hcc_custom_ini_data_buf:46 *****
***hwifi_hcc_custom_ini_data_buf:22 *****
==hal_initialize_phy==269==
==hal_device_state_init_event==623==21=
hwifi_get_country_code_etc already set country:CN
hwifi_get_region find CN in region_table_default
hwifi_get_region find CN in region_table_default
plat_soc:E]==wlan READY==
wifi_host_init_finish![wifi_calil cost 5423 ms].
hmac_fsm_change_state_etc state 5, vap_id 1
dlopen LIBWAPIPATH is /lib/abi/wifi-wapi.so
wapi_cali_back_init enter!
```

5. 使用方法

下面以进入 wpa_supplicant 交互模式为例（默认状态下直接输入 wpa_cli 即可进入，否则需要添加-p 指定具体的文件路径后进入）详细描述 73 作为 Sta 连接 wapi 路由器的命令

a) remove_network all

清理所有的网络配置信息，如果是首次启动 wpa_supplicant，该步骤可以省略

b) add_network

添加网络，wpa_supplicant 会返回一个待配置的网络节点编号，如果执行了步骤

a)则一定会返回编号 0

c) set_network 0 ssid "wifi123456"

配置待连接的 wapi 路由器的 ssid，例子为 wifi123456

d) set_network 0 proto WAPI

配置加密方式为 wapi

e) set_network 0 key_mgmt WAPI-PSK

配置密钥管理模式为 WAPI-PSK

f) set_network 0 psk_key_type 0

配置密钥字符串形式，分为 0 ASCLL 与 1 HEX 两种

g) set_network 0 psk "12345678"

配置待连接的 wapi 路由器的密码，例子为 12345678

h) select_network 0

选择该路由器开始连接