

## APPLICATION NOTE

### Linux 系统 ATBM iwpriv 命令使用说明



#### ATBM603X

1x1 802.11b/g/n  
Wi-Fi 芯片

## Table of contents

1	设置信号发射功率 1 .....	6
2	设置信号发射功率 2 .....	6
3	设置信道频点 .....	6
4	获取吞吐量 .....	7
5	获取信号强度 .....	7
6	获取当前连接状态 .....	8
7	获取 WIFI MAC 地址 .....	8
8	查询设置状态 .....	8
9	设置自适应认证功能 .....	9
10	设置 AGC .....	9
11	关闭 AGC .....	9
12	设置一段时间内不发送命令 .....	10
13	设置重传时间和重传次数 .....	10
14	发送单频 .....	10
15	停止发送单频 .....	10
16	获取当前的 EFUSE .....	11
17	设置 EFUSE 频偏 .....	11
18	设置 EFUSE 功率 .....	11
19	设置 EFUSE MAC 地址 .....	13

20	MANAGER 模式下监听 BEACON .....	13
21	清除私有配置设置 .....	13
22	查看底噪 .....	14
23	设置接收性能 .....	15
24	设置退避窗口 .....	15
25	设置固定发送速率 .....	15
26	设置最大发送速率 .....	16
27	设置最小发送速率 .....	16
28	设置固定 GI 模式 .....	17
29	设置 RTS THRESHOLD .....	17
30	获取 RTS 阈值 .....	17
31	读取温度 .....	18
32	设置不同速率的功率 .....	18
33	获取信道空闲命令 .....	18
34	获取接收灵敏度结果数据 .....	19
35	设置隐藏 SSID .....	19
36	增加 MAC 地址过滤的功能 .....	19
37	获取 LMAC MAC 过滤的结果 .....	20
38	设置接收 BEACON/PROBE REQ 帧 .....	20
39	设置接收携带自定义 IE 的帧 .....	21
40	清除所有的过滤条件 .....	21
41	查询已经设置的过滤条件 .....	21
42	获取 EFUSE 区域的第一个 MAC 地址 .....	21
43	获取 EFUSE 区域当前 MAC 地址的前一个 MAC 地址 .....	22
44	设置固定带宽命令 .....	22
45	设置发包队列命令 .....	22
46	打开 CFO 自动校准功能 .....	22
47	设置固定信道扫描 .....	23
48	设置部分 GPIO 状态 .....	23
49	设置部分 GPIO 输出状态 .....	24
50	设置 EDCA 参数 .....	25
51	获取 EDCA 参数 .....	26

52	设置发包占空比 .....	27
53	获取连接上设备支持的速率 .....	27
54	设置 POWER SAVE 模式 .....	28
55	设置不同模式 B/GN 功率配置命令 .....	28
56	设置不同模式 B/GN 不同速率的功率配置命令 .....	30
57	最佳信道选择 .....	31
58	支持 IFCONFIG 修改 MAC 地址 .....	32
59	获取当前的发包速率.....	32
60	获取第一次的 EFUSE.....	33
61	设置管理帧速率 .....	33
62	获取当前的工作信道.....	33
63	设置国家码 .....	33
64	获取国家码 .....	34
65	获取频偏和 DCX0 寄存器 .....	34
66	设置 PROBE REQUEST 插入 SSID/密码.....	34
67	监听 PROBE REQUEST 帧.....	34
68	获取 PROBE REQUEST 帧携带的数据.....	34
69	设置 CCA 门限.....	35
70	获取 CCA 门限 .....	35
71	设置 RTS DURATION 参数 .....	35
72	获取 RTS DURATION 参数 .....	36
73	获取信噪比 .....	36
74	获取底噪.....	36
75	重新加载 FIRMWARE(USB WIFI) .....	37
76	连续发送携带私有数据的 PROBE RESP.....	37
77	停止连续发送携带私有数据的 PROBE RESP .....	37
78	修改 AP 信道 .....	37
79	设置 BEACON TIM 广播/单播标志 .....	38
80	获取 BEACON TIM 当前的标志 .....	39

作者	版本	说明	时间
Yuzhihuang	V1.2	增加 37~40 命令，驱动 1584 版本以后支持	20200610

yuzhihuang	V1.3	增加 41~42 命令，驱动 1612 版本以后支持	20200618
Yuzhihuang	V1.3.1	增加 43~44 命令，其中 44 命令需要固件 10345 以后支持	20200819
yuzhihuang	V1.3.2	增加 45~50 命令，修改 29 命令，1770 以后驱动以及 1443 最新版本支持增加 51/52 命令	20201022
Yuzhihuang	V1.3.3	增加 53/54/55 命令，都要固件支持，17902 以后驱动 12040 版本以后固件支持 54/55/56 命令	20201201
Yuzhihuang	V1.3.4	增加 57/58 命令，58 命令需要 11293 以后的固件才支持	
Yuzhihuang	V1.3.5	增加获取第一次的 efuse	
Yuzhihuang	V1.3.6	加大功率可能导致的问题说明	20210716
Yuzhihuang	V1.3.7	增加国家码设置说明 修改发射功率档位说明	20220114
Yuzhihuang	V1.3.8	增加获取频偏值的功能（20818 版本驱动，14675 版本固件）  增加设置获取 proeb req 私有数据的功能	20220714
Yuzhihuang	V1.3.9	15096 版本固件支持 Cca 门限设置/获取， rts duration 设置/获取 获取信噪比 获取底噪 获取 rts 发包阈值	20221009
Yuzhihuang	V1.4.0	增加 74/75/76/77 命令	20221101
Yuzhihuang	V1.4.1	增加 79/80 命令	20221206
Yuzhihuang	V1.4.2	最大最小发送速率添加排序说明	20230113
Yuzhihuang	V1.4.3	增加 wifi6 子命令	20230209



## 1 设置信号发射功率 1

命令	<code>iwpriv [interface] fwcmd set_txpower,[idx]</code>
参数说明	<p>interface: wlan0 或者 p2p0</p> <p>idx :取值 0/3/15/63,</p> <p>0 表示正常发射功率模式</p> <p>3 表示高发射功率模式</p> <p>15 表示超高发射功率模式</p> <p>63 表示最高发射功率模式</p> <p><b>注意:</b></p> <p>如果功率增加太多, 比如超过 20dBm, 芯片功耗可能超过 500mA, 建议增加电源驱动能力。</p>
返回值	成功打印 0, 失败打印-1
例子	<pre>ifconfig wlan0 up iwpriv wlan0 fwcmd set_txpower,3</pre>

## 2 设置信号发射功率 2

命令	<code>iwpriv [interface] fwcmd set_rate_txpower_mode,[idx]</code>
参数说明	<p>interface: wlan0 或者 p2p0</p> <p>idx :取值[-16:16]</p> <p>如果设置 idx 为-16, 则发射功率为-8dB。</p> <p>如果设置 idx 为-6, 则发射功率为-3dB。</p> <p>如果设置 idx 为 0, 则发射功率为+0dB。</p> <p>如果设置 idx 为 6, 则发射功率为+3dB。</p> <p>如果设置 idx 为 16, 则发射功率为+8dB。</p> <p><b>注意:</b></p> <p>先设置了信号发射功率 1 的命令, 再设置 2 的命令, 功率会叠加</p> <p><b>注意:</b></p> <p>如果功率增加太多, 比如超过 20dBm, 芯片功耗可能超过 500mA, 建议增加电源驱动能力。</p>
返回值	无
例子	<code>Iwpriv wlan0 fwcmd set_rate_txpower_mode,-16</code>

## 3 设置信道频点

命令	<code>1、iwpriv [interface] fwcmd set_freq,[chan_number],[chan_value]</code>
----	---

	<b>2、iwpriv [interface] set_freq [chan_number],[chan_value]</b>
参数说明	Interface : wlan0 或者 p2p0 Chan_number: 信道号 , 取值 [1~13] Chan_value : 频点参数, 取值 2380 或者 2504, 或者正常的 1~13 频点  注意: 使用命令 2 设置特殊频点可以在文件系统查询到。 cat /sys/modules/atbmxxx/paramaters/wifi_freq
返回值	
例子	1、设置特殊频点 ifconfig wlan0 up iwpriv wlan0 fwcmd set_freq,1,2380 2、设置回正常频点 ifconfig wlan0 up iwpriv wlan0 fwcmd set_freq,1,2412

注:

(1) 不同的信道不能设置相同的频点, 驱动也需要做这方面的判断, 预防出错。

(2) 正常频点:

2412/2417/2422/2427/2432/2437/2442/2447/2452/2457/2462/2467/2472/2484

(3) 特殊频点: 2380/2504

## 4 获取吞吐率

命令	<b>iwpriv [interface] get_tp_rate [sta mac addr]</b>
参数说明	Interface : wlan0 或者 p2p0 sta mac addr: 设备本身做为 ap 模式需要获取连接的 sta 的当前环境的吞吐量, 该参数对应的是 sta 的 mac 地址, sta 模式该参数可以为空
返回值	单位是 <b>bits/s</b>
例子	ifconfig wlan0 up 连接上需要 ping、或者传输数据才行 iwpriv wlan0 get_tp_rate

## 5 获取信号强度

命令	<b>iwpriv [interface] common get_rssi [sta mac addr]</b>
参数说明	Interface : wlan0 或者 p2p0 sta mac addr: 设备本身做为 ap 模式需要获取连接的 sta 的信号强度, 该参数对应的是 sta 的 mac 地址, sta 模式该参数可以为空
返回值	当前的 rssi 值

例子	<pre>ifconfig wlan0 up iwpriv wlan0 common get_rssi</pre>
----	---

## 6 获取当前连接状态

命令	<code>iwpriv [interface] get_state</code>
参数说明	Interface: wlan0 或者 p2p0
返回值	打印出来 0 或者 1, 0 未连接状态 1 已连接状态
例子	<pre>ifconfig wlan0 up iwpriv wlan0 common get_state</pre>

## 7 获取 wifi mac 地址

命令	<code>Iwpriv [interface] getmac</code>
参数说明	interface: wlan0 或者 p2p0
返回值	打印出来 wlan0 或者 p2p0 的 mac 地址: dc:29:19:00:25:7f
例子	<code>Iwpriv wlan0 getmac</code>

## 8 查询设置状态

命令	<code>iwpriv [interface] fwcmd get_cmd_status</code>
参数说明	interface: wlan0 或者 p2p0
返回值	<ol style="list-style-type: none"> <li>1) 先打印: 当前信道号, 信道带宽, 主边带信道频点, 全信道中心频点。如果带宽是 20M, 则主边带信道频点和全信道中心频点一样; 否则两个值差 10M 或者 -10M。            比如: <code>current channel 6,HT20,main freq 2437MHz,middle freq 2437MHz.</code>            或者: <code>current channel 6,HT40+,main freq 2437MHz,middle freq 2447MHz.</code>            或者: <code>current channel 6,HT40-,main freq 2437MHz,middle freq 2427MHz.</code> </li> <li>2) 接着打印有设置特殊频点的信道, 比如我们用上面第 7 条命令把信道 7 设置成 2380M。则有打印: <code>channel 6,freq 2380MHz.</code>            如果没有设置特殊频点的信道, 就打印: <code>all freq is normal.</code> </li> <li>3) 接着再打印上面第 4 条命令设置的信道功率, 比如设置 <code>iwpriv wlan0 fwcmd set_txpower,63</code>, 就会打印 <code>txpower flag 63.</code> </li> <li>4) 最后再打印上面第 5/6 条命令设置的信道功率, 打印: <code>rate&lt;48M txpower change (%d/10)dB.all rate txpower change (%d/10)dB</code>, 其中第一个参数%d 是第 5 条命令设置的发射功率, 第二个参数%d 是第 6 条命令设置的发射功率。           </li> <li>5) 速率信息打印:</li> </ol>



	<p>如果没连上 ap, 且不固定速率和 gi, Lmac 打印: <b>free rate, no join in ap, free sgi.</b></p> <p>如果没连上 ap, 且固定速率和 gi, Lmac 打印: <b>fixed rate, no join in ap, fixed sgi.</b></p> <p>如果连上 ap, 且不固定速率和 gi, Lmac 打印: <b>free rate(%d/10), join in ap, free sgi.</b></p> <p>如果连上 ap, 且固定速率和 gi, Lmac 打印: <b>fixed rate(%d/10), join in ap, fixed sgi.</b></p> <p>上述 %d 打印的值是真实速率*10, 比如现在大部分数据包速率用 58.5M 速率发, 打印出来%d 就是 585, 有一点要说明: 比如打印出来是 58.5M 速率, 并不表示所有速率一定是用 58.5M 速率发, 由于速率控制一直在变化, 打印出来的值只是反应当前大部分数据包速率是用 58.5M 速率发。</p> <p>6) 共享晶体还是独立晶体打印:</p> <p>如果有打印: <b>crystal2</b> 表示共享晶体; 如果有打印: <b>crystal%d, %d</b> 打印是非 2 表示独立晶体</p>
例子	<pre>Iwpriv wlan0 fwdbg 1 iwpriv wlan0 fwcmd get_cmd_status</pre>

## 9 设置自适应认证功能

命令	<b>iwpriv wlan0 fwcmd set_adaptive,value</b>
参数说明	<p>Value = 1 开启自适应功能</p> <p>Value = 0 关闭自适应功能</p> <p>需要注意协议规定干扰信号在 -70dBm , ATBM WIFI 限制的门槛为 -75dBm , 干扰小于 -75dBm 测试就会有问题的</p>
返回值	Lmac 有打印: <b>adaprive value</b>
例子	

## 10 设置 AGC

命令	<b>iwpriv wlan0 fwcmd AgcStart</b>
参数说明	<p>Agc initial gain 越大, 越容易收到干扰, 但是灵敏度越好</p> <p>该命令表示: 表示 Agc initial gain 不是 high gain</p>
返回值	无
例子	无

## 11 关闭 AGC

命令	<b>iwpriv wlan0 fwcmd AgcStop</b>
参数说明	<p>Agc initial gain 越大, 越容易收到干扰, 但是灵敏度越好</p> <p>该命令表示: 表示 Agc initial gain 打成最大的 gain</p>
返回值	无
例子	无

## 12 设置一段时间内不发送命令

命令	<code>iwpriv wlan0 fwcmd set_tx_time,time_period,time_can_transmit</code>
参数说明	<p><b>time_period &gt; time_can_transmit</b> 并且这两个参数都不为 0 时，配置才起作用，这个时候我们芯片就会以 <b>time_period</b>（单位是 ms）时间为周期，每个周期前 <b>time_can_transmit</b>（单位是 ms）时间能发包，后 <b>(time_period - time_can_transmit)</b>（单位是 ms）时间不能发包。</p> <p>当配置不起作用，表示所有时间都能发包。</p>
返回值	Lmac 有打印: <code>set time error time_period time_can_transmit</code>
例子	无

## 13 设置重传时间和重传次数

命令	<code>iwpriv wlan0 fwcmd set_retry,retry_max_number,retry_max_time, unjoin_time</code>
参数说明	<p><b>retry_max_number</b>: 每个包最多发送次数。如果该参数超过 100，该参数配置就不起作用，每个包最多发送次数就是默认的发送次数（11 次）</p> <p><b>retry_max_time</b>: 每个包最多发送时间，单位是 ms。如果该参数超过 1000，该参数配置就不起作用，每个包最多发送时间就是默认的发送时间（256ms）</p> <p><b>unjoin_time</b>: 断网时间，单位是 s，最多是 250s，超过 250s 就用默认的是 4s</p>
返回值	Lmac 有打印: <code>rtty time %ums,num %d</code>
例子	无

## 14 发送单频

命令	<code>iwpriv wlan0 common singletone,&lt;channel&gt;</code>
参数说明	<p><b>Channel</b>: 发送 singletone 时的中心频点，取值范围 1~14</p> <p><b>Singletone 命令与 etf start_tx 以及 start_rx 命令互斥，发送 singletone 前必须先停止 start_tx 或者 start_rx</b></p>
返回值	无
例子	<p>发送没调制的 1 信道信号：</p> <pre>iwpriv wlan0 common singletone,1</pre>

## 15 停止发送单频

命令	<code>iwpriv wlan0 stop_tx</code>
----	-----------------------------------

参数说明	无
返回值	无
例子	

## 16 获取当前的 efuse

命令	<code>iwpriv wlan0 common getEfuse</code>
参数说明	无
返回值	Get efuse data is [1,80,5,4,4,6,0,0,dc:29:19:4c:81:e7] 80: 是 dcxo 频偏 5,4,4 : gain1,gain2,gain3 dc:29:19:4c:81:e7:mac 地址
例子	需要注意在所有设置 efuse 之前都必须先读取出来数据。 所以写 efuse 的流程按照如下规则： 【一定要按照下面的步骤，不然会设置失败】 1、get efuse 2、set efuse

## 17 设置 efuse 频偏

命令	<code>iwpriv wlan0 common setEfuse_dcxo,&lt;dcxo&gt;,write_rom</code>
参数说明	Dcxo : 设置频偏因子的值 , 范围 0~128 write_rom: 是否断电保存数据。1 保存, 0 不保存。 通过 get 回来的数据为基准数据: 该值改大, 频偏往小了偏 该值改小, 频偏往大了偏
返回值	无
例子	<code>iwpriv wlan0 common getEfuse</code> 返回值: Get efuse data is [1,80,5,4,4,6,0,0,dc:29:19:4c:81:e7] 当前频偏为: -3 <code>iwpriv wlan0 common setEfuse_dcxo,70,0</code> 当前频偏为: 12 (规律和晶体的曲线相关, 整体呈负相关)

## 18 设置 efuse 功率

命令	<code>iwpriv wlan0 common setEfuse_deltagain,&lt;delta_gain1&gt;,&lt;delta_gain2&gt;,&lt;delta_gain3&gt;,write_rom</code>
参数说明	delta_gain1: 低信道功率因子 : 0~31 delta_gain2: 中信道功率因子 : 0~31

	<p>delta_gain3: 高信道功率因子 : 0~31</p> <p>write_rom: 是否断电保存数据。1 保存, 0 不保存</p> <p>取值 0~15 功率增大</p> <p>取值 16~31 功率减小</p> <p><b>注意:</b></p> <p>如果功率增加太多, 比如超过 20dBm, 芯片功耗可能超过 500mA, 建议增加电源驱动能力。</p>
返回值	无
例子	<p>1、芯片内部的 efsue 小于 16</p> <pre>iwpriv wlan0 common getEfuse</pre> <p>返回值: Get efuse data is [1,80,5,4,4,6,0,0,dc:29:19:4c:81:e7]</p> <pre>iwpriv wlan0 common setEfuse_deltagain,9,8,8,1</pre> <p>功率因子修改前后的功率差别:</p> <p>低信道功率增加: <math>(9-5)/4 = 1\text{dB}</math></p> <p>中信道功率增加: <math>(8-4)/4 = 1\text{dB}</math></p> <p>高信道功率增加: <math>(8-4)/4 = 1\text{dB}</math></p> <pre>iwpriv wlan0 common setEfuse_deltagain,17,18,19,1</pre> <p>功率因子修改后的功率差别:</p> <p>低信道功率减少: <math>(17-32)/4 - 5/4 = -5\text{dB}</math></p> <p>中信道功率减少: <math>(18-32)/4 - 4/4 = -4.5\text{dB}</math></p> <p>高信道功率减少: <math>(19-32)/4 - 4/4 = -4.25\text{dB}</math></p> <p>2、芯片内部的 efsue 大于 16</p> <pre>iwpriv wlan0 common getEfuse</pre> <p>返回值: Get efuse data is [1,80,25,24,24,6,0,0,dc:29:19:4c:81:e7]</p> <pre>iwpriv wlan0 common setEfuse_deltagain,9,8,8,1</pre> <p>功率因子修改前后的功率差别:</p> <p>低信道功率增加: <math>9/4 - (25 - 32)/4</math></p> <p>中信道功率增加: <math>8/4 - (24 - 32)/4</math></p> <p>高信道功率增加: <math>8/4 - (24 - 32)/4</math></p> <pre>iwpriv wlan0 common setEfuse_deltagain,17,18,19,1</pre> <p>功率因子修改后的功率差别:</p> <p>低信道功率减少: <math>(17-32)/4 - (25 - 32)/4</math></p> <p>中信道功率减少: <math>(18-32)/4 - (24 - 32)/4</math></p> <p>高信道功率减少: <math>(19-32)/4 - (24 - 32)/4</math></p>

## 19 设置 efuse mac 地址

命令	<code>iwpriv wlan0 common setEfuse_mac,&lt;mac 地址&gt;</code>
参数说明	Mac 地址：格式必须是 <code>xx:xx:xx:xx:xx:xx</code>
返回值	无
例子	<pre>iwpriv wlan0 common getEfuse 返回值: Get efuse data is [1,80,5,4,4,6,0,0,dc:29:19:4c:81:e7] iwpriv wlan0 common setEfuse_mac,10:02:03:04:05:01</pre> <p>修改完成没有立即生效，需要立即生效需要重新卸载/加载下驱动。</p>

## 20 Manager 模式下监听 beacon

命令	<code>iwpriv wlan0 sta_channel &lt;channel&gt;</code>
参数说明	Channel : 信道,取值范围[1,14] 注意: 打开宏 <b>CONFIG_ATBM_STA_LISTEN</b>
返回值	无
例子	<pre>监听 1 信道的 beacon ifconfig wlan0 up iwpriv wlan0 sta_channel 1</pre>

## 21 清除私有配置设置

命令	<code>iwpriv wlan0 ipc_reset &lt;paramaters&gt;</code>
参数说明	Paramaters : data : 清空插入的私有数据 mac : 清空 mac 地址过滤功能 recv : 清空接收过滤回来的数据 all : 清空所有的配置 注意: 打开宏 <b>ATBM_PRIVATE_IE</b>
返回值	无
例子	<pre>1、清空插入的私有数据 ifconfig wlan0 up iwpriv wlan0 ipc_reset data</pre>

	<p>2、清空过滤的 mac 地址</p> <pre>ifconfig wlan0 up iwpriv wlan0 ipc_reset mac</pre> <p>3、清空过滤回来的所有数据</p> <pre>ifconfig wlan0 up iwpriv wlan0 ipc_reset recv</pre> <p>4、清空所有配置以及数据</p> <pre>ifconfig wlan0 up iwpriv wlan0 ipc_reset all</pre>
--	---

## 22 查看底噪

命令	<pre>iwpriv wlan0 fwcmd ccastart iwpriv wlan0 fwcmd AgcStop iwpriv wlan0 fwdbg 1</pre>
参数说明	<p>注意：</p> <p>查看底噪需要保证启动了 wpa_supplicant 或者执行了定频接收的功能</p> <p>不然打印都是 0</p>
返回值	<p>没有启动 wpa_supplicant 或者定频接收的情况：</p> <pre>[lmac]:ccast 0,bst 0,cca 0,inig 0,up 0,nodBm 0</pre> <p>启动了定频接收命令：</p> <p>接收 1 信道 HT20 的包：iwpriv wlan0 start_rx 1,0，结果如下：</p> <pre>ccast 8,bst 0,cca 14,inig 60,up 60,nodBm -86</pre> <p>执行了 wpa_supplicant 以后的结果如下：</p> <pre>[lmac]:ccast 1,bst 2,cca 15,inig 60,up 60,nodBm -90</pre> <p><b>底噪看：nodBm -90</b></p> <p>必须要保证 inig 这个值为 60 ， 底噪才是准确的。</p> <p>说明如下没有打印可能是系统打印等级不够,有下面几种方法：</p> <ol style="list-style-type: none"> <li>1、执行 cat /proc/kmsg &amp;</li> <li>2、echo 7 &gt; /proc/sys/kernel/printk</li> <li>3、</li> </ol>
例子	

## 23 设置接收性能

命令	<code>iwpriv wlan0 fwcmd set_rxlevel,&lt;idx&gt;</code>
参数说明	<p><code>idx</code> : 取值范围[0,10]</p> <p><code>idx</code> 设置为 0, 芯片接收性能正常。LMAC 会打印: <code>rx level normal</code></p> <p><code>idx</code> 设置为 1, 芯片接收性能会差 23dB 左右。</p> <p>如果设置 <code>idx</code> 为 2, 芯片接收性能会差 25dB 左右。</p> <p>如果设置 <code>idx</code> 为 3, 芯片接收性能会差 27dB 左右。</p> <p>...</p> <p>如果设置 <code>idx</code> 为 8, 芯片接收性能会差 37dB 左右。</p> <p>如果设置 <code>idx</code> 为 9, 芯片接收性能会差 39dB 左右。</p> <p>如果设置 <code>idx</code> 为 10, 芯片接收性能会差 41dB 左右。</p>
返回值	无
例子	<p>设置芯片接收性能差 25dB</p> <pre>iwpriv wlan0 fwcmd set_rxlevel,2</pre> <p>设置芯片接收性能正常</p> <pre>iwpriv wlan0 fwcmd set_rxlevel,0</pre>

## 24 设置退避窗口

命令	<code>iwpriv wlan0 fwcmd set_backoff,&lt;idx&gt;</code>
参数说明	<p><code>idx</code>:取值范围[0,1]</p> <p><code>idx</code> 取值为 1 , 减小退避窗口</p> <p><code>idx</code> 取值为其他, 正常退避窗口</p>
返回值	无
例子	<p>减小退避窗口</p> <pre>iwpriv wlan0 fwcmd set_backoff,1</pre> <p>恢复为正常退避窗口</p> <pre>iwpriv wlan0 fwcmd set_backoff,0</pre>

## 25 设置固定发送速率

命令	<code>iwpriv wlan0 fwcmd lmac_rate,&lt;rate&gt;</code>
参数说明	<p>Rate 为要固定的速率, <math>rate = \text{实际的 rate} * 10</math>, 所以有如下取值:</p> <p>11b 速率是: 10, 20, 55, 110;</p> <p>11g 速率是: 60, 90, 120, 180, 240, 360, 480, 540;</p> <p>11n 速率是: 65, 130, 195, 260, 390, 520, 585, 650;</p> <p>设置成 0 就是恢复成算法控制</p>
返回值	无
例子	设置 5.5M 的发送速率:

	Iwpriv wlan0 fwcmd lmac_rate,55
--	---------------------------------

## 26 设置最大发送速率

命令	<code>iwpriv wlan0 fwcmd lmac_max_rate,&lt;rate&gt;</code>
参数说明	<p>Rate 为要固定的速率, <math>rate = \text{实际的 rate} * 10</math>, 所以有如下取值:</p> <p>11b 速率是: 10, 20, 55, 110;</p> <p>11g 速率是: 60, 90, 120, 180, 240, 360, 480, 540;</p> <p>11n 速率是: 65, 130, 195, 260, 390, 520, 585, 650;</p> <p><b>注意:</b></p> <p>其中速率的排序为 11n &gt; 11g &gt; 11b</p> <p>先按照模式排序, 同一模式按照速率大小排序</p> <p>65M &gt; 6.5M &gt; 54M &gt; 6M &gt; 11M &gt; 1M</p>
返回值	无
例子	<p>设置最大的发送速率为 36M:</p> <p><code>Iwpriv wlan0 fwcmd lmac_max_rate,360</code></p>

## 27 设置最小发送速率

命令	<code>iwpriv wlan0 fwcmd lmac_min_rate,&lt;rate&gt;</code>
参数说明	<p>Rate 为要固定的速率, <math>rate = \text{实际的 rate} * 10</math>, 所以有如下取值:</p> <p>11b 速率是: 10, 20, 55, 110;</p> <p>11g 速率是: 60, 90, 120, 180, 240, 360, 480, 540;</p> <p>11n 速率是: 65, 130, 195, 260, 390, 520, 585, 650;</p> <p><b>注意:</b></p> <p>其中速率的排序为 11n &gt; 11g &gt; 11b</p> <p>先按照模式排序, 同一模式按照速率大小排序</p> <p>65M &gt; 6.5M &gt; 54M &gt; 6M &gt; 11M &gt; 1M</p>
返回值	无
例子	<p>举例:</p> <ol style="list-style-type: none"> <li>1) 如果固定速率设置 39M, 则固定最大速率和固定最小速率命令不起作用。</li> <li>2) 如果不设置固定速率, 固定最大速率设置 65M, 固定最小速率设置 52M, 则发送速率在 52M 和 65M 之间。</li> <li>3) 如果不设置固定速率, 固定最大速率设置 52M, 固定最小速率设置 52M, 则发送速率是 52M。</li> <li>4) 如果不设置固定速率, 固定最大速率设置 39M, 固定最小速率设置 52M, 则发送速率也</li> </ol>



	是 52M。
--	--------

## 28 设置固定 GI 模式

命令	<code>iwpriv wlan0 fwcmd lmac_sgi,&lt;gi_mode&gt;</code>
参数说明	<p><b>Gi_mode</b> 取值范围为[0,1,other]</p> <p>该参数为 1 时，发送包 GI 模式都是 short GI，</p> <p>该参数为 0 时，发送包 GI 模式都是 long GI，</p> <p>该参数为 2 时，发送包 GI 模式是由底层速率算法控制</p>
返回值	无
例子	<p>设置为 short gi:</p> <pre>Iwpriv wlan0 fwcmd lmac_sgi,1</pre>

## 29 设置 rts threshold

命令	<code>iwpriv wlan0 rts_threshold package_len</code>
参数说明	<p><b>package_len</b>:设置触发 rts 的包长度，取值范围(0,2000)</p> <p>设置超过 2000 关闭 rts</p>
返回值	无
例子	<p>关闭发 rts:</p> <pre>iwpriv wlan0 rts_threshold 2000</pre> <p>包长超过 1000 触发 rts</p> <pre>iwpriv wlan0 rts_threshold 1000</pre>

## 30 获取 rts 阈值

指令	<code>iwpriv wlan0 common get_rts_threshold</code>
参数	无
功能描述	获取设置的 rts 阈值。
返回值	<p>打印: <code>rts_threshold:%d</code></p> <p>失败打印: <code>rts_threshold get fail</code></p>
指令示例	<code>iwpriv wlan0 common get_rts_threshold</code>

## 31 读取温度

命令	<code>iwpriv wlan0 fwcmd get_temp</code>
参数说明	获取当前芯片内部温度
返回值	无
例子	

## 32 设置不同速率的功率

命令	<code>iwpriv wlan0 fwcmd set_spec_rate_txpower_mode,rate_idx,rate_txpower</code>
参数说明	<p><b>rate_idx: 0~10, 对应的速率如下,</b></p> <p>0→ 1/2M  1→ 5.5/11M  2→ 6/6.5M  3→ 9M  4→ 12/13M  5→ 18/19.5M  6→ 24/26M  7→ 36/39M  8→ 48/52M  9→ 54/58.5M  10→ 65M</p> <p><b>rate_txpower: -16~16</b></p> <p><b>注意:</b></p> <p>如果功率增加太多, 比如超过 20dBm, 芯片功耗可能超过 500mA, 建议增加电源驱动能力。</p>
返回值	无
例子	<p>设置 MCS7 速率功率增加 1dB:</p> <p><code>Iwpriv wlan0 fwcmd set_spec_rate_txpower_mode,10,2</code></p> <p>设置 54M 速率功率增加 2dB</p> <p><code>Iwpriv wlan0 fwcmd set_spec_rate_txpower_mode,9,4</code></p>

## 33 获取信道空闲命令

命令	<code>iwpriv wlan0 common get_channel_idle</code>
参数说明	
返回值	<p><b>current_idle:value</b>, 其中信道空闲 <b>value</b> 取值范围是 1~1024。</p> <p>值越大, 信道越空闲, 信道空闲测量底层是每 2~3s 才计算一次。</p> <p>如果信道空闲值小于 100, 建议换信道或者降低码率。</p> <p>当打印信道空闲值为 0 时, 表示信道测量还没测出来, 可以 3s 后再测量一次。</p>

	<p>当读取信道空闲值和上次值一样，表示当前的信道空闲还没计算出来，这个时候打印的还是上次测量的信道空闲值</p> <p>由于一次信道空闲值可能不准，建议读取多次信道空闲值做平均，注意做平均时需要去掉和上次信道空闲值一样的值。</p> <p>比如：</p> <p>多次读取信道空闲值时：</p> <pre>current_idle:100, current_idle:200, current_idle:200, current_idle:150, current_idle:200, current_idle:100,</pre> <p>平均出来值应该是 <math>(100+200+150+200+100)/5=150</math>， 而不是 <math>(100+200+200+150+200+100)/6=950/6</math></p>
例子	

## 34 获取接收灵敏度结果数据

命令	<code>iwpriv wlan0 rx_statist</code>
参数说明	<p>该命令必须要在已经执行了,接收灵敏度统计命令 <code>iwpriv wlan0 start_rx chan,is_40M</code> 前提下执行的。</p> <p>直接获取当前时刻统计的数据。</p> <p>如果没有执行接收灵敏度统计命令，直接执行 <code>iwpriv wlan0 rx_statist</code> 获取结果会执行失败。</p>
返回值	成功返回 0，失败返回 -1
例子	

## 35 设置隐藏 ssid

命令	<code>iwpriv wlan0 common ap_hide_ssid,&lt;enable&gt;</code>
参数说明	<p>该命令必须要在已经启动了 AP 模式才有效果。</p> <p>Enable:1 隐藏 ssid，0 关闭隐藏 ssid 功能</p>
返回值	成功返回 0，失败返回 -1
例子	

## 36 增加 mac 地址过滤的功能

命令	<code>iwpriv wlan0 common monitor_mac,switch,mac_addr</code>
----	--

参数说明	Switch : 1 打开 lmac mac 过滤 , 0 关闭 lmac_mac 过滤 Mac_addr: 需要过滤的 mac 地址
返回值	成功返回 0, 失败返回-1
例子	<p>1、打开 lmac mac 过滤功能</p> <p>先打开 lmac 打印: iwpriv wlan0 fwdbg 1, 在执行命令确认有设置到 lmac</p> <p>iwpriv wlan0 common monitor_mac,1,11:22:33:44:55:66</p> <p>[lmac]:monitor,staDC,29,19,4C,81,E7</p> <p>2、关闭 lmac mac 过滤功能</p> <p>iwpriv wlan0 common monitor_mac,0</p>

### 37 获取 lmac mac 过滤的结果

命令	iwpriv wlan0 common monitor_status
参数说明	无
返回值	成功返回 0, 失败返回-1
例子	<pre>[/ext/demo]## iwpriv wlan0 common monitor_status wlan0      common:found=[1],rssi=[-23],delta_time(1996)ms [/ext/demo]## iwpriv wlan0 common monitor_status wlan0      common:found=[1],rssi=[-23],delta_time(3108)ms [/ext/demo]## iwpriv wlan0 common monitor_status wlan0      common:found=[1],rssi=[-22],delta_time(32)ms ---- 32ms 前收到这个包 [/ext/demo]## iwpriv wlan0 common monitor_status wlan0      common:found=[1],rssi=[-22],delta_time(1024)ms</pre> <p>结果说明:</p> <p>found 代表收到 mac 地址对应的设备, 只要收到就一直为 1, 直到关闭 mac 过滤</p> <p>rssi 设备的信号强度</p> <p>delta_time 距离上一次收到这个设备无线包的时间。</p>

### 38 设置接收 beacon/probe req 帧

命令	iwpriv wlan0 common filter_frame,<idx>
参数说明	<p>Idx :</p> <p>80 接收 beacon 帧</p> <p>40 接收 probe req 帧</p> <p>需要启动 hostapd 或者 wpa_supplicant</p>
返回值	成功返回 0, 失败返回-1
例子	1、设置过滤 beacon 帧

	Iwpriv wlan0 common filter_frame,80 2、设置过滤 probe req 帧 Iwpriv wlan0 common filter_frame,40  数据处理函数 hal_apollo/mac80211/special_filter.c ieee80211_special_filter_rx_package_handle

## 39 设置接收携带自定义 IE 的帧

命令	iwpriv wlan0 common filter_ie,<id>,<oui1>,<oui2>,<oui3>
参数说明	Id:IE 对应的 id 号,十进制 Oui1-oui2-oui3:IE 对应的 oui, 十进制 Oui1,oui2,oui3 可为空  <b>需要启动 hostapd 或者 wpa_supplicant</b>
返回值	成功返回 0, 失败返回-1
例子	1、设置过滤 221 字段的帧 Iwpriv wlan0 common filter_ie,221  2、设置过滤 221 字段 oui 为 5-6-7-的帧 Iwpriv wlan0 common filter_ie,221,5,6,7  数据处理函数 hal_apollo/mac80211/special_filter.c ieee80211_special_filter_rx_package_handle

## 40 清除所有的过滤条件

命令	iwpriv wlan0 common filter_clear
参数说明	无
返回值	成功返回 0, 失败返回-1
例子	需要注意如果设置使用的网口是 p2p0 那么清除也需要使用 p2p0 网口进行清除数据

## 41 查询已经设置的过滤条件

命令	iwpriv wlan0 common filter_show
参数说明	无
返回值	成功返回 0, 失败返回-1
例子	

## 42 获取 efuse 区域的第一个 mac 地址

命令	iwpriv wlan0 common get_first_mac
----	-----------------------------------

参数说明	无
返回值	成功返回 0，失败返回 -1
例子	

### 43 获取 efuse 区域当前 mac 地址的前一个 mac 地址

命令	<code>iwpriv wlan0 common get_last_mac</code>
参数说明	无
返回值	成功返回 0，失败返回 -1
例子	

### 44 设置固定带宽命令

命令	<code>iwpriv wlan0 fwcmd lmac_bw,bw_mode</code>
参数说明	<b>bw_mode:</b> 该参数为 1 时，我们发送包带宽是 40M，lmac 有打印: <b>fixeBW 1.</b> 该参数为 0 时，我们发送包带宽是 20M，lmac 有打印: <b>fixeBW 0</b> <b>其他参数默认由算法控制。</b>
返回值	成功返回 0，失败返回 -1
例子	

### 45 设置发包队列命令

命令	<code>iwpriv wlan0 fwcmd set_ac,sta_queue,ap_queue</code>
参数说明	<b>sta_queue:</b> sta 模式发包的优先级队列设置，取值 0~4 <b>ap_queue :</b> ap 模式发包的优先级队列设置，取值 0~4 0~4 优先级递增，3~4 一般用于音视频 设置成 0，默认发包优先级
返回值	成功返回 0，失败返回 -1
例子	

### 46 打开 cfo 自动校准功能

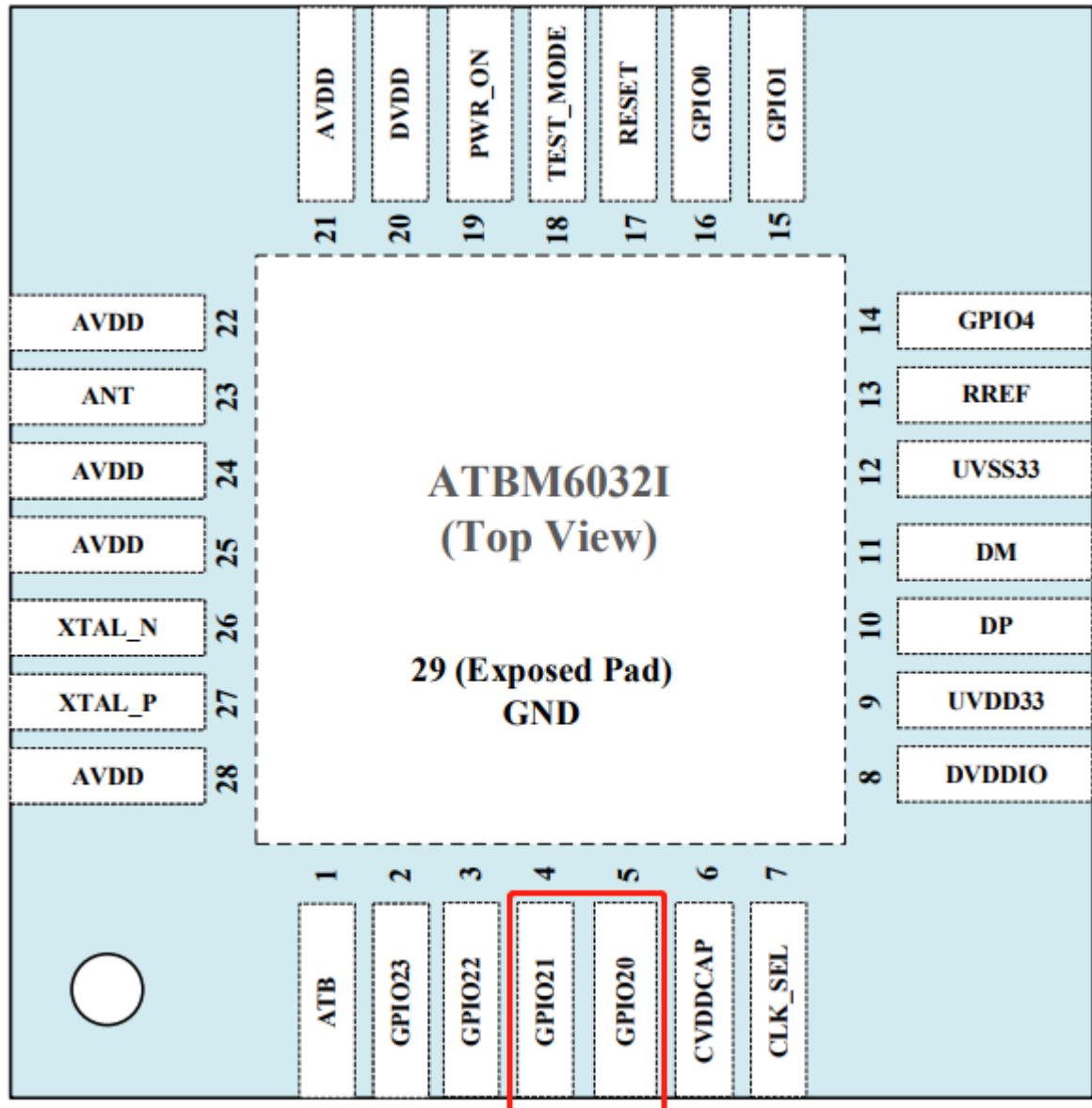
命令	<code>iwpriv wlan0 fwcmd cfo,p1,[p2]</code>
参数说明	需要注意打开和关闭命令格式不同
返回值	成功返回 0，失败返回 -1
例子	打开自动频偏校准: <code>Iwpriv wlan0 fwcmd cfo,1</code> 关闭自动频偏校准: <code>Iwpriv wlan0 fwcmd cfo,0,0</code>

## 47 设置固定信道扫描

命令	<code>iwpriv wlan0 common ap_conf,channel</code>
参数说明	驱动需要打开宏 <code>CONFIG_ATBM_SUPPORT_AP_CONFIG</code> <code>channel:1~14</code>
返回值	成功返回 0，失败返回 -1
例子	<p>设置固定 1 信道扫描</p> <pre>Iwpriv wlan0 common ap_conf,3</pre> <p>Iwlist wlan0 scan 此时只会扫描 3 信道</p> <p>关闭固定信道扫描</p> <pre>Iwpriv wlan0 common ap_conf,0</pre> <p>Iwlist wlan0 scan 此时会扫描 1~14 信道</p>

## 48 设置部分 GPIO 状态

命令	<code>iwpriv wlan0 common gpio_conf,gpio,dir,pup or pud</code>
参数说明	gpio 取值为 : [20,21] , 对应芯片 pin map 的 <b>GPIO20/GPIO21</b>



**Figure 2 ATBM6032I pin map (Top View)**

dir : 设置 gpio 输入输出状态, 0 输入, 1 输出

pup or pud : 设置内部上下拉, 0 内部下拉, 1 内部上拉

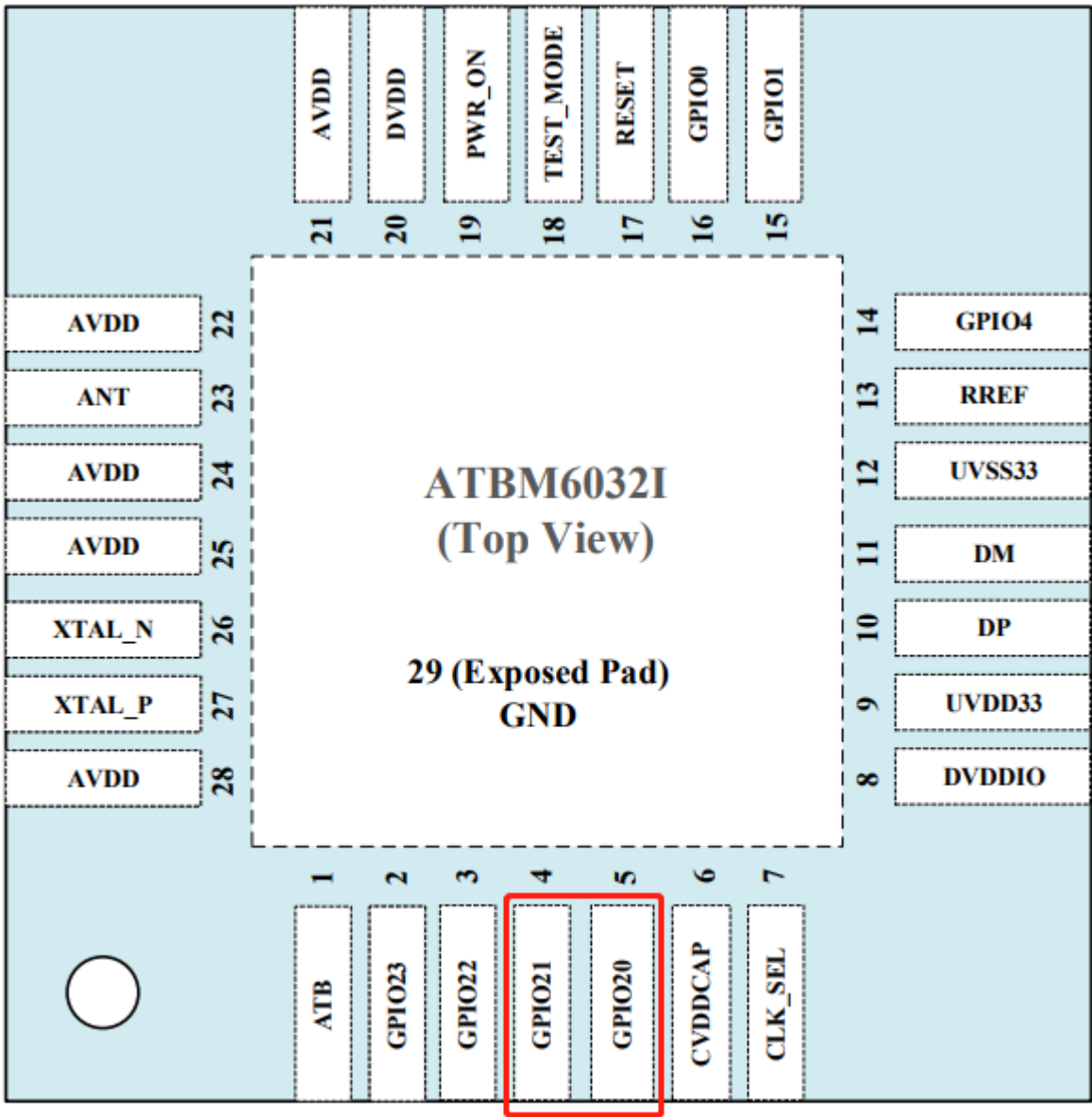
返回值 成功返回 0, 失败返回 -1

例子 设置 GPIO20 为输出上拉状态  
iwpriv wlan0 common gpio\_conf,20,1,1

## 49 设置部分 GPIO 输出状态

命令 iwpriv wlan0 common gpio\_out,gpio,val



<p>参数说明</p>	<p>gpio 取值为：[20,21]，对应芯片 pin map 的 <b>GPIO20/GPIO21</b></p>  <p style="text-align: center;"><b>ATBM6032I (Top View)</b></p> <p style="text-align: center;"><b>29 (Exposed Pad) GND</b></p> <p style="text-align: center;"><b>Figure 2 ATBM6032I pin map (Top View)</b></p> <p>value: 0 低电平 1 高电平</p>
<p>返回值</p>	<p>成功返回 0，失败返回 -1</p>
<p>例子</p>	<p>设置 GPIO20 为输出上拉状态 iwpriv wlan0 common gpio_out,20,1</p>

## 50 设置 edca 参数

命令	iwpriv wlan0 common edca_params,queue,aifs,cw_min,cw_max,txop
参数说明	queue:

	<p>0:voice traffic 1:video traffic 2:best effort 3:background traffic</p> <p>txop; 传输机会, 参数值为 TXOPlimit, 代表占用信道发包最长时间, 取值范围: 0~65535, 取值为 0 代表只发送一个 MPDU</p> <p>cw_min; 最小竞争窗口, 越小的 CWmin 其优先级越高, 取值范围: 0~15</p> <p>cw_max; 最大竞争窗口, 越小的 CWmax 其优先级越高, CWmax &gt; CWmin, 取值范围: 0~15</p> <p>aifs; EDCA 模式的 QSTA 要获得传输机会时, 必须等待的信道空闲时间, 时间越大代表优先级越低, 取值范围: 2~15</p> <p><b>注意:</b></p> <p><b>sta 这边配置了一套 edca 参数, 如果该 sta 连上 ap, 那么 sta 这边原本配置的 EDCA 参数会被刷新为 ap 的 EDCA 参数。</b></p> <p><b>连接上 ap 以后重新设置会再次生效。</b></p> <p><b>抓包看变化不明显, 主要在本地会生效, 自己的发包策略会变化。</b></p>
返回值	成功返回 0, 失败返回 -1
例子	

## 51 获取 edca 参数

命令	iwpriv wlan0 common get_edca_params
参数说明	
返回值	成功返回 0, 失败返回 -1
例子	<pre>[/tmp]## iwpriv p2p0 common get_edca_params [atbm_log]:[voice_traffic] [atbm_log]:aifns : 1 [atbm_log]:cwMax : 7 [atbm_log]:cwMin : 3 [atbm_log]:txOpLimit : 1504 [atbm_log]:[video_traffic] [atbm_log]:aifns : 1 [atbm_log]:cwMax : 15 [atbm_log]:cwMin : 7 [atbm_log]:txOpLimit : 3008 [atbm_log]:[best_effort] [atbm_log]:aifns : 3 [atbm_log]:cwMax : 63 [atbm_log]:cwMin : 15 [atbm_log]:txOpLimit : 0 [atbm_log]:[background_traffic] [atbm_log]:aifns : 7</pre>

	[atbm_log]:cwMax : 1023 [atbm_log]:cwMin : 15 [atbm_log]:txOpLimit : 0
--	--

## 52 设置发包占空比

命令	iwpriv wlan0 common duty_ratio,value
参数说明	Value:取值范围 0~1 <b>注意:</b> 只能在定频时候使用, 并且只修改 11n 的占空比
返回值	成功返回 0, 失败返回 -1
例子	设置当前的占空比为 50% iwpriv wlan0 common duty_ratio,0.5

## 53 获取连接上设备支持的速率

命令	iwpriv wlan0 common stations_show
参数说明	无
返回值	成功返回 0, 失败返回 -1
例子	<pre> ~ # iwpriv wlan0 common stations_show wlan0      common:station show --&gt; mac[fc:be:7b:0c:8f:65],rssi[-77],bg[fff],11n[ff]  ~ # iwpriv wlan0 common stations_show wlan0      common:station show --&gt; mac[fc:be:7b:0c:8f:65],rssi[-76],bg[fff],11n[ff]  ~ # iwpriv wlan0 common stations_show wlan0      common:station show --&gt; mac[fc:be:7b:0c:8f:65],rssi[-76],bg[fff],11n[ff]  ~ # iwpriv wlan0 common stations_show wlan0      common:station show --&gt; mac[fc:be:7b:0c:8f:65],rssi[-76],bg[fff],11n[ff]  ~ # iwpriv wlan0 common stations_show wlan0      common:station show --&gt; mac[fc:be:7b:0c:8f:65],rssi[-75],bg[fff],11n[ff] </pre> <p>说明:</p> <p>mac: 连接设备的 mac 地址</p> <p>rssi:信号强度</p> <p>bg/11n 协商的速率:</p> <p>其中: bg[fff] --&gt;转成二进制共 12 位对应 b/g 12 个速率:  从低到高对应速率: [1M 2M 5.5M 11M 6M 9M 12M 18M 24M 36M 48M 54M]  如上图: bg[fff] 代表支持 b/g 全速率  如果: bg[55f] 代表支持[1M 2M 5.5M 11M 6M 12M 24M 48M]  其中: 11n[ff] 从低到高对应的是 MCS0~MCS7  如上图: 11n[ff] 代表当前连接支持 MCS0~MCS7  如果: 11n[53] 代表支持[MCS0 MCS1 MCS4 MCS6]</p>

## 54 设置 power save 模式

命令	<code>iwpriv wlan0 common power_save,value</code>
参数说明	Value: 1:开启休眠功能 0:关闭休眠功能 进入休眠以后会发 null data 包, 包里面 power save 位为 1
返回值	成功返回 0, 失败返回 -1
例子	

## 55 设置不同模式 b/gn 功率配置命令

命令	<code>iwpriv wlan0 common settxpower_byfile</code>
参数说明	 <p><b>注意:</b> 如果功率增加太多, 比如超过 20dBm, 芯片功耗可能超过 500mA, 建议增加电源驱动能力。</p>
返回值	成功返回 0, 失败返回 -1
例子	<p>根据配置有两种设置 efuse 的方式。</p> <p>1、直接覆盖 efuse 区域的数据直接采用配置文件配置的数据。 在原 efuse 数据的基础上进行叠加。 配置文件格式:</p> <pre>atbm_txpwer_dcxo_cfg.txt - 记事本 文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H) delta_gain1:1 delta_gain2:2 delta_gain3:3 dcxo:59 b_delta_gain1:1 b_delta_gain2:2 b_delta_gain3:3 gn_delta_gain1:1 gn_delta_gain2:2 gn_delta_gain3:3</pre> <p>芯片 efuse 默认数据: delta_gain1:1 delta_gain2:1 delta_gain3:1 dcxo:59</p>

## (1) 覆盖式

### a. 配置文件数据:

```
delta_gain1:2 delta_gain2:2 delta_gain3:2 dcxo:75
b_delta_gain1:3 b_delta_gain2:3 b_delta_gain3:3
gn_delta_gain1:4 gn_delta_gain2:4 gn_delta_gain3:4
```

### Dcxo 75 写入寄存器

b 模 信道 1~4 功率增加  $(2/4+3/10) = 0.8\text{dB}$

b 模 信道 5~9 功率增加  $(2/4+3/10) = 0.8\text{dB}$

b 模 信道 10~14 功率增加  $(2/4+3/10) = 0.8\text{dB}$

gn 模 信道 1~4 功率增加  $(2/4+4/10) = 0.9\text{dB}$

gn 模 信道 5~9 功率增加  $(2/4+4/10) = 0.9\text{dB}$

gn 模 信道 10~14 功率增加  $(2/4+4/10) = 0.9\text{dB}$

### b. 配置文件数据:

```
delta_gain1:18 delta_gain2:19 delta_gain3:20 dcxo:66
b_delta_gain1:3 b_delta_gain2:3 b_delta_gain3:3
gn_delta_gain1:4 gn_delta_gain2:4 gn_delta_gain3:4
```

### Dcxo 66 写入寄存器

b 模 信道 1~4 功率增加  $(3/10 - 1/4 + (18-32)/4) = -3.45\text{dB}$

b 模 信道 5~9 功率增加  $(3/10 - 1/4 + (19-32)/4) = -3.2\text{dB}$

b 模 信道 10~14 功率增加  $(3/10 - 1/4 + (20-32)/4) = -2.95\text{dB}$

gn 模 信道 1~4 功率增加  $(4/10 - 1/4 + (18-32)/4) = -3.35\text{dB}$

gn 模 信道 5~9 功率增加  $(4/10 - 1/4 + (19-32)/4) = -3.1\text{dB}$

gn 模 信道 10~14 功率增加  $(4/10 - 1/4 + (20-32)/4) = -2.85\text{dB}$

## (2) 叠加式

第一行数据都设置在取值范围之外即按照 efuse 区域的数据进行处理。

### 配置文件数据:

```
delta_gain1:-1 delta_gain2:-1 delta_gain3:-1 dcxo:-1
b_delta_gain1:3 b_delta_gain2:3 b_delta_gain3:3
gn_delta_gain1:4 gn_delta_gain2:4 gn_delta_gain3:4
```

### Dcxo 59 写入寄存器

b 模 信道 1~4 功率增加  $(1/4+3/10) = 0.55\text{dB}$

b 模 信道 5~9 功率增加  $(1/4+3/10) = 0.55\text{dB}$


b 模 信道 10~14 功率增加  $(1/4+3/10) = 0.55\text{dB}$

gn 模 信道 1~4 功率增加  $(1/4+4/10) = 0.65\text{dB}$

gn 模 信道 5~9 功率增加  $(1/4+4/10) = 0.65\text{dB}$

gn 模 信道 10~14 功率增加  $(1/4+4/10) = 0.65\text{dB}$

## 56 设置不同模式 b/gn 不同速率的功率配置命令

命令	<code>iwpriv wlan0 common set_rate_power,&lt;useflag&gt;</code>
参数说明	<p><b>useflag:</b></p> <p>1:读取配置进行功率设置</p> <p>0:复位设置</p> <p>配置文件路径设置 hal_apollo/Makefile:</p> <pre># # 可以用来选择不同模式, CONFIG_USB_AGGR_URB_TX 定义上 # 1&gt; CONFIG_USB_AGGR_URB_TX + CONFIG_TX_NO_CONFIRM, 每 # 2&gt; CONFIG_USB_AGGR_URB_TX 每个urb 发送多个txframe 并! # 3&gt; 所有宏都不定义, 每个urb 发送一个txframe 并且需要cc # 4&gt; 定义CONFIG_USE_DMA_ADDR_BUFFER, 每个urb 发送一个tx # 5&gt; 定义CONFIG_USE_DMA_ADDR_BUFFER+ CONFIG_TX_NO_CONFI # 6&gt; 定义CONFIG_TX_NO_CONFIRM, 每个urb 发送一个txframe # ##对于调度慢+cpu强的板子使用 模式1 或者 模式2 ##对于调度快+cpu慢的板子使用 模式3 或者 模式6 ## 默认使用模式 3 ##### #ccflags-y += -DCONFIG_ATBM_APOLLO_5GHZ_SUPPORT #ccflags-y += -DCONFIG_ATBM_APOLLO_WAPI_SUPPORT #ccflags-y += -I\$(shell pwd)/../include/ \     -include \$(shell pwd)/../include/linux/compat-2     -DCOMPAT_STATIC ##指定配置文件路径, 当配置文件存在时使用配置文件中的dcx ATBM_CONFIG_FILE="/atbm txpwer dcxo cfg.txt" ##指定配置文件路径, 当配置文件存在时使用配置文件中的det ATBM_RATE_POWER_CONFIG_FILE="/set_rate_power.txt"</pre> <p><b>注意:</b></p> <p>如果功率增加太多, 比如超过 20dBm, 芯片功耗可能超过 500mA, 建议增加电源驱动能力。</p>
返回值	成功返回 0, 失败返回-1
例子	<p>配置文件格式:</p>  <pre>b_1M_2M=1 b_5_5M_11M=2 g_6M_n_6_5M=3 g_9M=4 g_12M_n_13M=5 g_18M_n_19_5M=-6 g_24M_n_26M=-7 g_36M_n_39M=-8 g_48M_n_52M=-9 g_54M_n_58_5M=-10 n_65M=11 </pre> <p>数据只能取【-16: 16】, 在发射功率基础上调整【-8: 0.5: 8】dB 如果超过这个范围, 该命令对应增加的发射功率直接为 0</p> <p>如果设置 b_1M_2M 为 10, 则 1/2M 发射功率增加 5dB。</p> <p>如果设置 g_6M_n_6_5M 为 -10, 则 6/6.5M 发射功率减少 5dB。。</p> <p><b>注意,</b></p> <ol style="list-style-type: none"> <li>1) 该配置命令只在有烧过 efuse 模组或 EVB 才起作用。</li> <li>2) 驱动在加载的时候不会默认去读取这个文件并根据配置文件内容生效。</li> </ol> <p>运行过程中修改了配置文件内容, 并想立即生效直接执行命令:</p> <pre>iwpriv wlan0 common set_rate_power,1</pre> <p>驱动会去刷新配置并立即生效。</p>

使用命令将设置好的配置复位:

**`iwpriv wlan0 common set_rate_power,0`**

## 57 最佳信道选择

命令	<code>iwpriv wlan0 best_ch_scan &lt; <b>BestChannelSelect_t</b> &gt;</code>
参数说明	<p>有特殊信道使用需要打开宏: CONFIG_ATBM_5G_PRETEND_2G</p> <p>参数说明:</p> <pre>#define CHANNEL_NUM 18 //信道个数 typedef struct _best_ch_scan_result_ {     //每个信道 AP 的个数     unsigned int channel_ap_num[CHANNEL_NUM];      //每个信道的信道忙比例, 范围 0~128 (0: 信道全空闲, 128: 信道全忙)     unsigned int busy_ratio[CHANNEL_NUM];     //每个信道比重     unsigned int weight[CHANNEL_NUM];     //统计后, 建议的信道     unsigned char suggest_ch; }Best_Channel_Scan_Result;  typedef struct BestChannelSelect{     unsigned char start_channel;     unsigned char end_channel;     unsigned char SpecialFlag;     Best_Channel_Scan_Result scan_result; }<b>BestChannelSelect_t</b>;</pre> <p>说明:</p> <p><code>start_channel</code>; 起始统计信道</p> <p><code>end_channel</code>; 终止统计信道</p> <p><code>SpecialFlag</code>; 是否统计特殊信道</p> <p><code>scan_result</code>: 保存返回的结果</p>
返回值	成功返回 0, 失败返回 -1
例子	<p>建议使用自带的 <code>iwpriv_demo</code> 进行测试:</p> <p>1、<code>iwpriv wlan0 best_ch_scan</code></p> <p>默认统计 1~14 信道的最佳信道, 但是应用层获取不到数据, 内核只打印最佳信道</p> <p>2、<code>iwpriv_demo wlan0 best_ch_scan start_ch end_ch specialFlag</code></p> <p>2.1) <code>iwpriv_demo wlan0 best_ch_scan 1 11 0</code></p> <p>统计 1~11 信道</p> <p>2.2) <code>iwpriv_demo wlan0 best_ch_scan 1 14 0</code></p> <p>统计 1~14 信道</p>

	<p>2.3) iwpriv_demo wlan0 best_ch_scan 1 14 1 统计 1~14 信道</p> <p>2.4) iwpriv_demo wlan0 best_ch_scan 1 11 1 统计 1~11 信道</p> <p>2.5) iwpriv_demo wlan0 best_ch_scan 1 42 0 统计 1~14 信道</p> <p>2.6) iwpriv_demo wlan0 best_ch_scan 1 42 1 统计 1~14 信道 + 统计 36/38/40/42 信道</p> <p>2.7) iwpriv_demo wlan0 best_ch_scan 1 36 1 统计 1~14 信道+统计 36 信道</p> <p>2.8) iwpriv_demo wlan0 best_ch_scan 1 36 0 统计 1~14 信道</p>
--	--

## 58 支持 ifconfig 修改 mac 地址

命令	ifconfig wlan0 hw ether [mac_addr]
参数说明	<p>1.wlan0/1 网口状态必须处于 down 状态</p> <p>2.只能修改 wlan0 网口的 mac 地址。</p> <p>3.打开宏 <code>ccflags-y += -DCONFIG_ATBM_SUPPORT_REALTIME_CHANGE_MAC</code></p>
返回值	成功返回 0，失败返回 -1
例子	ifconfig wlan0 hw ether 00:11:22:33:44:55

## 59 获取当前的发包速率

命令	iwpriv wlan0 common get_maxrate[,mac_addr]
参数说明	<p>mac_addr:</p> <p>wlan0 做为 AP 模式时候使用的参数，mac_addr 为连接上 AP 的 sta mac 地址</p>
返回值	成功返回 0，失败返回 -1
例子	<p>1、sta 模式获取当前的主要发包速率</p> <p><code>iwpriv wlan0 common get_maxrate</code></p> <p>2、ap 模式获取与连接的 sta (00:11:22:33:44:55) 当前的主要发包速率</p>



Iwpriv wlan0 common get_maxrate,00:11:22:33:44:55
---

## 60 获取第一次的 efuse

命令	iwpriv wlan0 common getefusefirst
参数说明	无
返回值	Get efuse data is [1,80,5,4,4,6,0,0,dc:29:19:4c:81:e7] 80: 是 dcxo 频偏 5,4,4 : gain1,gain2,gain3 dc:29:19:4c:81:e7:mac 地址
例子	

## 61 设置管理帧速率

命令	iwpriv wlan0 common set_frame_rate,value
参数说明	在 NVR IPC 固件，默认将管理帧的速率设置为 2Mbps Value: 1 : 设置管理帧为 1Mbps 0 : 恢复为 2Mbps 速率
返回值	
例子	

## 62 获取当前的工作信道

命令	iwpriv wlan0 common get_work_channel
参数说明	无
返回值	
例子	

## 63 设置国家码

命令	iwpriv wlan0 common country_code,[country]
参数说明	country:只支持如下三个参数 CN : 中国, 支持 1~13 信道 US : 美国, 支持 1~11 信道 JP : 日本, 支持 1~14 信道
返回值	
例子	设置国家码为美国 ifconfig wlan0 up iwpriv wlan0 common country_code,US

## 64 获取国家码

## 65 获取频偏和 dcxo 寄存器

命令	<code>iwpriv wlan0 common get_cali_data</code>
参数说明	无
返回值	
例子	<code>iwpriv wlan0 common get_cali_data</code>

## 66 设置 probe request 插入 ssid/密码

命令	<code>iwpriv wlan0 common user_vendor_ie,ssid,psk</code>
参数说明	<b>ssid:</b> 可连接的 ap 的 ssid <b>psk:</b> 可连接的 ap 的密码
返回值	如果 WIFI 有工作信道则只在本工作信道发送，如果无工作信道则在 1~14 信道发送
例子	<code>iwpriv wlan0 common user_vendor_ie,yzh_test,12345678</code> 抓空中包能看到发送的 probe req 携带私有数据，私有 IE 为 221

## 67 监听 probe request 帧

命令	<code>iwpriv wlan0 common listen_probe_req,channel</code>
参数说明	<b>channel:</b> 需要监听的信道，如果 sta 已经连接上 ap，那么监听信道只能在 ap 所在的信道，channel 参数无效 <b>注意:</b> <b>wlan0 只能是 sta 模式</b>
返回值	
例子	<code>iwpriv wlan0 common listen_probe_req,1</code> 在 1 信道监听接收 probe req 帧

## 68 获取 probe request 帧携带的数据

命令	<code>iwpriv wlan0 common get_user_vendor_ie</code>
参数说明	无

返回值	
例子	<pre>iwpriv wlan0 common get_user_vendor_ie</pre> <p>如果有监听到数据:</p> <pre>atbm_ioctl_get_vendor_ie : ssid[yzh_test],password[12345678]</pre> <p>如果没有数据:</p> <pre>atbm_ioctl_get_vendor_ie : ssid[],password[]</pre>

## 69 设置 cca 门限

指令	<code>iwpriv wlan0 fwcmd cca,0</code>
参数	<p>&lt;set_cca_thr&gt;:</p> <p>该参数默认 0，最大能设置【-25：25】dB，超过这个范围参数设置无效，也就是用上次设置值 set_cca_thr（没有设置就用默认值 0）；物理意义就是在默认 cca 门限基础上提高 set_cca_thr dB。</p> <p>注意，我们芯片检测信道空闲或者繁忙有两个方法：收到一个包（包检测方法）或者检测到能量比较大（能量检测方法）都会产生信道忙，只有没收到包且能量小才会产生信道空闲。</p> <p>这里设置 cca 门限只是设置能量检测的门限，值越大能量检测越容易出信道空闲；如果在空气中有很多包或者在跑吞吐率，则设这个 cca 门限基本没用，因为收到包就触发信道忙，根本不走能量检测逻辑。</p>
功能描述	设置 CCA 门限。
返回值	打印: set_cca_thr%d dB
指令示例	<pre>iwpriv wlan0 fwcmd cca,6</pre> <pre>iwpriv wlan0 fwcmd cca,-6</pre>

## 70 获取 cca 门限

指令	<code>iwpriv wlan0 common get_cca_threshold</code>
参数	无
功能描述	获取 CCA 门限。
返回值	<p>打印: cca_threshold:%d</p> <p>失败打印: cca_threshold get fail</p>
指令示例	<code>iwpriv wlan0 common get_cca_threshold</code>

## 71 设置 rts duration 参数

指令	<code>iwpriv wlan0 fwcmd set_dur,val</code>
参数	<p>val 目前默认为 8000，单位是 us</p> <p>val 是 u32 类型</p>

	取值范围是：[1,8000] 超出范围设置无效
功能描述	修改 rts duration 值。
返回值	打印：set_cca_thr%d dB
指令示例	iwpriv wlan0 fwcmd set_dur,4000 iwpriv wlan0 fwcmd set_dur,1

## 72 获取 rts duration 参数

指令	iwpriv wlan0 common get_rts_duration
参数	无
功能描述	获取设置的 rts duration 值。
返回值	打印：rts_duration:%d 失败打印：rts_duration get fail
指令示例	iwpriv wlan0 common get_rts_duration

## 73 获取信噪比

指令	iwpriv wlan0 common get_snr
参数	无
功能描述	获取设置的 rts duration 值。
返回值	获取成功打印：lmac get snr_val = %d ,real snr = 10*log10(4096/%d) 失败打印：snr get fail  有效数据获取出来的 snr 值需要在应用端进行计算出实际有效的 snr： snr = 10*log10(4096/ snr_val)
指令示例	iwpriv wlan0 common get_rts_duration

## 74 获取底噪

指令	iwpriv wlan0 common get_noise_level
参数	无
功能描述	获取设置的 rts 阈值。
返回值	打印： 底噪有效打印： exact noise_level_dBm = %d 底噪是无效数据的打印： initg = %d,reckon noise_level_dBm = %d  失败打印：noise get fail

指令示例	<code>iwpriv wlan0 common get_noise_level</code>
------	--

## 75 重新加载 firmware(USB wifi)

指令	<code>iwpriv wlan0 common fw_reset</code>
参数	无
功能描述	重新加载 firmware
返回值	返回 0
指令示例	<code>iwpriv wlan0 common fw_reset</code>

## 76 连续发送携带私有数据的 probe resp

指令	<code>iwpriv wlan0 common send_probe_resp,ssid,psk</code>
参数	ssid: ap 要广播的 ssid psk: ap 的密码
功能描述	Ap 模式下使用有效，在 ap 模式下以间隔为 5ms 的时间一直发送携带私有数据的 probe resp 包
返回值	成功返回 0，失败非 0
指令示例	Hostapd.conf 配置的 ssid 和 psk 为: yzh_test 12345678 <code>iwpriv wlan0 common send_probe_resp,yzh_test,12345678</code>

## 77 停止连续发送携带私有数据的 probe resp

指令	<code>iwpriv wlan0 common stop_probe_resp</code>
参数	无
功能描述	AP 模式下使用，停止连续发送携带私有数据的 probe resp
返回值	成功返回 0，失败返回非 0
指令示例	<code>iwpriv wlan0 common stop_probe_resp</code>

## 78 修改 ap 信道

指令	<code>iwpriv wlan0 common change_chan,chan,ht40</code>
参数	chan: 信道范围: [1,14] ht40: 是否是 HT40 模式:

	0, 非 HT40 1, HT40 模式
功能描述	AP 模式下使用，不用重启 hostapd 实现切换信道，必须使用高拓提供的 hostapd 才行，hostapd -v 显示为：hostapd_xx_ATBM_XX_XXXXXX
返回值	成功返回 0，失败返回非 0
指令示例	切换到 1 信道 HT20 模式 iwpriv wlan0 common change_chan,1,0 切换到 1 信道 HT40 模式 iwpriv wlan0 common change_chan,1,1

## 79 设置 beacon TIM 广播/单播标志

指令	iwpriv wlan0 common set_tim,ena,tim_val
参数	ena : 使能， 关闭：0，由 WIFI 原本逻辑进行控制 使能：1，由用户进行控制，WIFI 原本逻辑无效 tim_val : 十六进制方式设置 Bitmap control & Partial Virtual Bitmap 的值，其中低 8bit 为 Bitmap control 控制位，可以设置为 0/1。 AltoBeam wifi Partial Virtual Bitmap 有 3 字节共 24bit 控制位，当连接的 sta 个数少于 8 个时候，可以控制的只有 8bit，当连接的 sta 个数超过 8 个，可以控制的 16bit。
功能描述	<b>通过该方式控制：</b> <b>min_inactivity &amp; max_inactivity 不能设置为 0，请设置为正常值。</b>
返回值	成功返回 0，失败返回非 0
指令示例	<b>例子：</b> 使能： Bitmap control 设置为 1 ==》 01 Partial Virtual Bitmap 的第 0bit，第 5bit 设置为 1 ==》 21=== 0010 0001 iwpriv p2p0 common set_tim,1,2101 <pre> [/tmp]## [/tmp]## iwpriv p2p0 common set_tim,1,2101 [atbm_log]:atbm_ioctl_common_cmd(), length 15 [atbm_log]:get data : ena:1, val:2101 [atbm_log]:extra: set_tim success  p2p0      common: set_tim success </pre>

```

Length: 1 [64]
Channel: 1 [65]
Traffic Indication Map
Element ID: 5 Traffic Indication Map [66]
Length: 6 [67]
DTIM Count: 1 [68]
DTIM Period: 2 [69]
Bitmap Control: %0000000 [70 Mask 0xFE]
----- . Bitmap Offset: 0 [70 Mask 0xFFFFFFFF]
.... ...1 Traffic Ind: Group Frames Buffered at AP
Part Virt Bmap: 0x210000 [71-73]
ERP Information
Element ID: 42 ERP Information [74]
Length: 1 [75]

```

关闭:

```
iwpriv p2p0 common set_tim,1,0
```

```
iwpriv p2p0 common set_tim,0,0
```

关闭需要两条的原因是，Bitmap control 只由 WIFI 驱动这边控制，所以如果不先清空，那么 lmac 那边这个标志就会一直为 1，直到 WIFI 驱动这边将该位清 0。

Packet	Source	Destination	BSSID	Flags	Channel	Signal dbm	Data Rate	Size	Absolute Time	Rel...	Decoder	Part Virt Bmap	Protocol
75272	9E:1C:37:87:B1:07	9E:1C:37:87:B1:07	9E:1C:37:87:B1:07	*	1	-18	2.0	157	16:17:19.772589	1...			002.11 Probe Rsp
75273	9E:1C:37:87:B1:07	9E:1C:37:87:B1:07	9E:1C:37:87:B1:07	+	1	-19	2.0	157	16:17:19.773576	1...			002.11 Probe Rsp
75274	9E:1C:37:87:B1:07	9E:1C:37:87:B1:07	9E:1C:37:87:B1:07	+	1	-19	2.0	157	16:17:19.801532	1...			002.11 Probe Rsp
75275	9E:1C:37:87:B1:07	FF:FF:FF:FF:FF:FF	9E:1C:37:87:B1:07	*	1	-18	2.0	165	16:17:19.844807	1...	0x210000		002.11 Beacon
75276	9E:1C:37:87:B1:07	FF:FF:FF:FF:FF:FF	9E:1C:37:87:B1:07	*	1	-18	2.0	165	16:17:19.844818	1...	0x210000		002.11 Beacon
75277	9E:1C:37:87:B1:07	58:64:92:04:8A:50	9E:1C:37:87:B1:07	+	1	-19	2.0	157	16:17:20.109846	1...			002.11 Probe Rsp
75278	9E:1C:37:87:B1:07	26:63:92:AF:EB:59	9E:1C:37:87:B1:07	*	1	-18	2.0	157	16:17:20.243703	1...			002.11 Probe Rsp
75279	26:63:92:AF:EB:59	9E:1C:37:87:B1:07	9E:1C:37:87:B1:07	*	1	-64	1.0	14	16:17:20.244810	1...			002.11 Ack
75280	9E:1C:37:87:B1:07	FF:FF:FF:FF:FF:FF	9E:1C:37:87:B1:07	*	1	-19	2.0	165	16:17:20.252422	1...	0x210000		002.11 Beacon
75281	9E:1C:37:87:B1:07	FF:FF:FF:FF:FF:FF	9E:1C:37:87:B1:07	*	1	-19	2.0	165	16:17:20.663527	1...	0x210000		002.11 Beacon
75282	9E:1C:37:87:B1:07	FF:FF:FF:FF:FF:FF	9E:1C:37:87:B1:07	*	1	-19	2.0	165	16:17:20.663527	1...	0x210000		002.11 Beacon
75283	9E:1C:37:87:B1:07	06:PS:AD:90:AC:52	9E:1C:37:87:B1:07	*	1	-19	2.0	157	16:17:20.769145	1...			002.11 Probe Rsp
75284	9E:1C:37:87:B1:07	06:PS:AD:90:AC:52	9E:1C:37:87:B1:07	+	1	-19	2.0	157	16:17:20.778187	1...			002.11 Probe Rsp
75285	9E:1C:37:87:B1:07	06:PS:AD:90:AC:52	9E:1C:37:87:B1:07	+	1	-19	2.0	157	16:17:20.771857	1...			002.11 Probe Rsp
75286	9E:1C:37:87:B1:07	26:63:92:AF:EB:59	9E:1C:37:87:B1:07	*	1	-19	2.0	157	16:17:20.839266	1...			002.11 Probe Rsp
75287	26:63:92:AF:EB:59	9E:1C:37:87:B1:07	9E:1C:37:87:B1:07	*	1	-47	1.0	14	16:17:20.839578	1...			002.11 Ack
75288	9E:1C:37:87:B1:07	FF:FF:FF:FF:FF:FF	9E:1C:37:87:B1:07	+	1	-71	2.0	157	16:17:20.844843	1...			002.11 Probe Rsp
75289	9E:1C:37:87:B1:07	9E:1C:37:87:B1:07	9E:1C:37:87:B1:07	+	1	-54	1.0	14	16:17:20.848780	1...			002.11 Ack
75290	9E:1C:37:87:B1:07	9E:1C:37:87:B1:07	9E:1C:37:87:B1:07	+	1	-50	1.0	14	16:17:20.841956	1...			002.11 Ack
75291	9E:1C:37:87:B1:07	FF:FF:FF:FF:FF:FF	9E:1C:37:87:B1:07	*	1	-19	2.0	165	16:17:20.805825	1...	0x000000		002.11 Beacon
75292	9E:1C:37:87:B1:07	FF:FF:FF:FF:FF:FF	9E:1C:37:87:B1:07	*	1	-19	2.0	165	16:17:20.970813	1...	0x000000		002.11 Beacon
75293	9E:1C:37:87:B1:07	FF:FF:FF:FF:FF:FF	9E:1C:37:87:B1:07	*	1	-19	2.0	165	16:17:21.275217	1...	0x000000		002.11 Beacon
75294	9E:1C:37:87:B1:07	7E:00:00:20:53:F0	9E:1C:37:87:B1:07	+	1	-19	2.0	157	16:17:21.317312	1...			002.11 Probe Rsp
75295	9E:1C:37:87:B1:07	26:63:92:AF:EB:59	9E:1C:37:87:B1:07	*	1	-18	2.0	157	16:17:21.361395	1...			002.11 Probe Rsp
75296	26:63:92:AF:EB:59	9E:1C:37:87:B1:07	9E:1C:37:87:B1:07	*	1	-50	1.0	14	16:17:21.361706	1...			002.11 Ack
75297	9E:1C:37:87:B1:07	FF:FF:FF:FF:FF:FF	9E:1C:37:87:B1:07	*	1	-20	2.0	165	16:17:21.388018	1...	0x000000		002.11 Beacon
75298	9E:1C:37:87:B1:07	FF:FF:FF:FF:FF:FF	9E:1C:37:87:B1:07	*	1	-19	2.0	165	16:17:21.491204	1...	0x000000		002.11 Beacon
75299	9E:1C:37:87:B1:07	FF:FF:FF:FF:FF:FF	9E:1C:37:87:B1:07	*	1	-19	2.0	165	16:17:21.585068	1...	0x000000		002.11 Beacon

## 80 获取 beacon TIM 当前的标志

指令	iwpriv p2p0 common get_tim
参数	
功能描述	
返回值	成功返回 0，失败返回非 0

指令示例

```
[/tmp]##
[/tmp]## iwpriv p2p0 common set_tim,1,2101
[atbm_log]:atbm_ioctl_common_cmd(), length 15
[atbm_log]:get data : ena:1, val:2101
[atbm_log]:extra:
set_tim success

p2p0      common:
set_tim success

[/tmp]##
[/tmp]##
[/tmp]## iwpriv p2p0 common get_tim
[atbm_log]:atbm_ioctl_common_cmd(), length 8
[atbm_log]:extra:
beacon tim :customer control tim,Bitmap control:0x01 , Partial Virtual Bitmap:0x21

p2p0      common:
beacon tim :customer control tim,Bitmap control:0x01 , Partial Virtual Bitmap:0x21

[/tmp]##
[/tmp]## I

[/tmp]##
[/tmp]##
[/tmp]##
[/tmp]## iwpriv p2p0 common set_tim,1,0
[atbm_log]:atbm_ioctl_common_cmd(), length 12
[atbm_log]:get data : ena:1, val:0
[atbm_log]:extra:
set_tim success

p2p0      common:
set_tim success

[/tmp]##
[/tmp]## iwpriv p2p0 common get_tim
[atbm_log]:atbm_ioctl_common_cmd(), length 8
[atbm_log]:extra:
beacon tim :customer control tim,Bitmap control:0x00 , Partial Virtual Bitmap:0x0

p2p0      common:
beacon tim :customer control tim,Bitmap control:0x00 , Partial Virtual Bitmap:0x0

[/tmp]##
[/tmp]## iwpriv p2p0 common set_tim,0,0
[atbm_log]:atbm_ioctl_common_cmd(), length 12
[atbm_log]:get data : ena:0, val:0
[atbm_log]:extra:
set_tim success

p2p0      common:
set_tim success

[/tmp]##
[/tmp]## iwpriv p2p0 common get_tim
[atbm_log]:atbm_ioctl_common_cmd(), length 8
[atbm_log]:extra:
beacon tim :aotu control tim,Bitmap control:0x00 , Partial Virtual Bitmap:0x0

p2p0      common:
beacon tim :aotu control tim,Bitmap control:0x00 , Partial Virtual Bitmap:0x0

[/tmp]##
```

提示由用户控制

提示驱动自动控制

## 81 undefine

## 82 WIFI6 相关命令

### 82.1 设置固定 tx 速率

指令	iwpriv wlan0 common tx_rate,idx
参数	Idx:



	<p>表示为速率 id 号</p> <p>设置 255 表示使用自动速率控制，设置下面的值表示固定速率</p> <pre> #define RATE_INDEX_B_1M                0 #define RATE_INDEX_B_2M                1 #define RATE_INDEX_B_5_5M              2 #define RATE_INDEX_B_11M               3 #define RATE_INDEX_PBCC_22M             4    // not supported/unused #define RATE_INDEX_PBCC_33M            5    // not supported/unused #define RATE_INDEX_A_6M                 6 #define RATE_INDEX_A_9M                 7 #define RATE_INDEX_A_12M                8 #define RATE_INDEX_A_18M                9 #define RATE_INDEX_A_24M               10 #define RATE_INDEX_A_36M                11 #define RATE_INDEX_A_48M                12 #define RATE_INDEX_A_54M                13 #define RATE_INDEX_N_6_5M               14 #define RATE_INDEX_N_13M                15 #define RATE_INDEX_N_19_5M              16 #define RATE_INDEX_N_26M                17 #define RATE_INDEX_N_39M                18 #define RATE_INDEX_N_52M                19 #define RATE_INDEX_N_58_5M              20 #define RATE_INDEX_N_65M                21 #define RATE_INDEX_N_MCS32_6M           22 #define RATE_INDEX_HE_MCSE0             23 #define RATE_INDEX_HE_MCSE1             24 #define RATE_INDEX_HE_MCSE2             25 #define RATE_INDEX_HE_MCS0              26 #define RATE_INDEX_HE_MCS1              27 #define RATE_INDEX_HE_MCS2              28 #define RATE_INDEX_HE_MCS3              29 #define RATE_INDEX_HE_MCS4              30 #define RATE_INDEX_HE_MCS5              31 #define RATE_INDEX_HE_MCS6              32 #define RATE_INDEX_HE_MCS7              33 #define RATE_INDEX_HE_MCS8              34 #define RATE_INDEX_HE_MCS9              35 #define RATE_INDEX_HE_MCS10             36 #define RATE_INDEX_HE_MCS11            37 </pre>
功能描述	
返回值	成功返回 0，失败返回非 0
指令示例	设置固定速率为 MCS10

	<code>iwpriv wlan0 common tx_rate,36</code>
--	---



---

### **CONTACT INFORMATION**

AltoBeam (China) Inc.

Address: B808, Tsinghua Tongfang Hi-Tech Plaza, Haidian, Beijing, China 100083

Tel: (8610) 6270 1811

Fax: (8610) 6270 1830

Website: [www.altobeam.com](http://www.altobeam.com)

Email: [support@altobeam.com](mailto:support@altobeam.com)

### **DISCLAIMER**

Information in this document is provided in connection with AltoBeam products. No license, express or implied, by estoppels or otherwise, to any intellectual property rights is granted by this document. Except as provided in AltoBeam's terms and conditions of sale for such products, AltoBeam assumes no liability whatsoever, and AltoBeam disclaims any express or implied warranty, relating to sale and/or use of AltoBeam products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right.

AltoBeam may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." AltoBeam reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Unauthorized use of information contained herein, disclosure or distribution to any third party without written permission of AltoBeam is prohibited.

AltoBeam™ is the trademark of AltoBeam. All other trademarks and product names are properties of their respective owners.

Copyright © 2007~2020 AltoBeam, all rights reserved