AIC8800D80X2 RF TEST COMMANDS

RF_TEST

v4.0

2025-7-28

AIC SEMICONDUCTOR (SHANGHAI) CO., LTD.

Company	AIC Semiconductor (Shanghai) CO., Ltd.	
Version information	Date	Release note
V1.0	2024 年 8 月 19 日	Update pwrofst,add set_ant
V2.0	2024 年 12 月 23 日	Update config, add The TX command can set the TX power.
V3.0	2025 年 3 月 20 日	Flash write protection
V4.0	2025 年 7 月 28 日	Add Flash write protection parameter definitions and update recommended length values.

Catalogue

1. Introduction of tools

Available for linux(ubuntu /android) .Fmacfw.bin is used in normal mode, and lmacfw_rf.bin is used in test mode. Take ubuntu as an example, and enter the test command in the user interface: (The following commands all take wlan0 as an example, and the actual ifconfig display shall prevail).
Format: wifi_test if_name command parameters
COMMANDS:
enum {

1. SET_TX,
2. SET_TXSTOP,
3. SET_RX,
4. SET_RXSTOP,
5. GET_RX_RESULT,
6. SET_XTAL_CAP,
7. SET_XTAL_CAP_FINE,
8. SET_FREQ_CAL,
9. SET_FREQ_CAL_FINE,
10. GET_FREQ_CAL,
11. SET_TXTONE
12. SET_SRRC
13. SET_MAC_ADDR,
14. GET_MAC_ADDR,
15. SET_BT_MAC_ADDR,
16. GET_BT_MAC_ADDR,
17. RDWR_PWRLVL,
18. RDWR_PWROFST2X,
19. RDWR_EFUSE_PWROFST2X,
20. AIC_USERCONFIG.TXT,

};

2. WIFI_TEST COMMAND

2.1 WIFI part

2.1.1 WiFi_test

1. wifi_test wlan0 set_tx chan bw mode rate length interval(interval can be omitted) power(interval can be omitted) \\ WiFi tx test start.

1-1-1：channel

	Chan_num
2.4G	ch1-ch13
5G	Ch36-ch165

1-1-2：bandwidth

	bw
0	20M
1	40M
2	80M

1-1-3：The relationship between mode and rate

idx	mode		rate											
0	NON HT	idx	0	1	2	3	4	5	6	7	8	9	10	11
		rate	1M	2M	5.5M	11M	6M	9M	12M	18M	24M	36M	48M	54M
2	HT MF	idx	0-7											
		rate	MCS0-MCS7											
4	VHT	idx	0-9											
		rate	MCS0-MCS9											
5	HE SU	idx	0-11											
		rate	MCS0-MCS11											

idx	mode		rate
2	HT MF	idx	8-15
		rate	MIMO (MCS8-MCS15)
4	VHT	idx	16-25
		rate	MIMO (MCS0-MCS9)
5	HE SU	idx	16-27
		rate	MIMO (MCS0-MCS11)

Length：

	20M	40M	80M
B/NON-HT	1000		
HT/VHT/HE	4000（mcs7-11） 1000(mcs0)	8000（mcs7-11） 1000(mcs0)	16000(mcs7-11) 1000(mcs0)

Interval：This parameter is configured based on actual usage. If there is no requirement for the packet delivery interval, you can not write this parameter and use the default value.
(Packet delivery interval: minimum 50,unit: μs).

Powr: Power value, this parameter is based on actual usage configuration. If there is no requirement for power, this parameter can be left blank or set to 255 to use the default value. (If this parameter is input, the preceding interval parameter also needs to be provided.)

SISO:
eg: wifi_test wlan0 set_tx 1 0 2 7 4000 \\ 2412MHz ,HT 20 MCS7 ,length4000(The default value of the delivery interval).

eg: wifi_test wlan0 set_tx 1 0 2 7 4000 1000 \\ 2412MHz ,HT 20 MCS7 ,length4000, interval: 1 ms
eg: wifi_test wlan0 set_tx 1 0 2 7 4000 1000 15 \\ 2412MHz ,HT 20 MCS7 ,length4000, interval: 1 ms,power:15dBm.

MIMO:
eg: wifi_test wlan0 set_tx 1 0 2 8 4000 \\ MIMO 2412MHz ,HT20 MCS8 ,length4000
eg: wifi_test wlan0 set_tx 1 0 4 16 4000 \\ MIMO 2412MHz ,VHT20 MCS0 ,length4000
eg: wifi_test wlan0 set_tx 1 0 5 27 4000 \\ MIMO 2412MHz ,HE20 MCS11 ,length4000

2. wifi_test wlan0 set_txstop \\ WiFi TX test stop
   no parameter

3. wifi_test wlan0 set_rx chan_num bw \\ WiFi RX test start

   eg: wifi_test wlan0 set_rx 1 0 \\ 2412MHz，bandwidth 20M

4. wifi_test wlan0 set_rxstop \\ WiFi RX test stop
   no parameter

5. wifi_test wlan0 get_rx_result \\ WiFi Test the number of packets received.
   no parameter
6. Return parameter：Total number of packets received from set_rx to set_rxstop

7. wifi_test wlan0 set_ant val \\ Antenna Port Selection Settings.
   val:0 ANT0 on, ANT1 on
   val:1 ANT0 on, ANT1 off
   val:2 ANT0 off, ANT1 on

2.1.2 single tone

wifi_test wlan0 set_txtone val \\ tx single tone
val: 0 tone off
val：**1 val** tone on（val: -20–19）

0	no parameter		
1	-20- -1	0	1-19
	negative bias	Central frequency point	Positive bias

eg: wifi_test wlan0 set_txtone 1 0 \\tone on
eg: wifi_test wlan0 set_txtone 0 \\ tone off

2.1.3 SRRC commands

When the SRRC spurious test is passed, turn on the SRRC command
wifi_test wlan0 set_srrc val
val:0 srrc off
val:1 srrc on

2.1.4 Flash protection

wifi_test wlan0 exec_flash_oper val

val: 0 \\ Check the current Flash status, return value -1: SR status abnormal; 0: write protection off; 1: write protection on.

1 \\ Recover and restore Flash SR status, turn off write protection.
2 \\ Protect open Flash write protection.
3 \\ read_wcr0 reads the power-on calibration result flag; returning a non-zero value indicates that there are calibration results, while a negative value indicates that there are no calibration results.
4 \\ Erase the power-on calibration results.

Note： Applicable to AIC8800M80X2P, completes Flash switch write protection, read status, and reads/erases power-on calibration results.

2.1.5 Crystal frequency offset calibration command

AIC8800D80X2 Variable load capacitance is provided inside the crystal circuit, and crystal units with load capacitance of 9-11pF are supported.

1. wifi_test wlan0 set_xtal_cap val \\ Coarse adjustment of crystal frequency offset, default value： 16(0x10)，range ： 0-31(0x00~0x1F)

   val: Decimal signed numbers

   eg: wifi_test wlan0 set_xtal_cap -2 \\ Negative frequency offset, reducing internal load capacitance

2. wifi_test wlan0 set_xtal_cap_fine val \\ Fine adjustment of crystal frequency offset, default value: 31(0x1F) ,range :0-63 (0x00~0x3F)

   val: Decimal signed numbers

   eg： wifi_test wlan0 set_xtal_cap_fine 10 \\ Positive frequency offset to improve internal load capacitance

3. wifi_test wlan0 set_freq_cal val \\ Write the coarse adjustment value of crystal frequency offset calibration to efuse(2 times)\flash (repeat).

   val:Hexadecimal absolute value
   eg: wifi_test wlan0 set_freq_cal 1a \\ Write the crystal frequency offset calibration coarse adjustment value 0x1a to efuse(2 times)\flash (repeat).

4. wifi_test wlan0 set_freq_cal_fine val \\Write fine adjustment value of crystal frequency offset calibration to efuse(2 times)\flash (repeat).

   val:Hexadecimal absolute value
   eg: wifi_test wlan0 set_freq_cal_fine 16 \\ Write crystal frequency offset calibration fine adjustment value 0x16 to efuse(2 times)\flash (repeat).

5. wifi_test wlan0 get_freq_cal \\ Reading frequency offset value
   no parameter

Coarse adjustment calibration flow：
②Judgment frequency offset（Δf）polar,Δf>0,setxtalcap 2, on the contrary,setxtalcap -2;
③Judgment frequency offset（Δf）polar,Δf>0,setxtalcap 1,on the contrary,setxtalcap -1;

Fine-tune the calibration process：
①Judgment frequency offset（Δf）polar,Δf>0,setxtalcapfine 16,on the contrary,setxtalcapfine -16;
②Judgment frequency offset（Δf）polar,Δf>0,setxtalcapfine 8,on the contrary,setxtalcapfine -8;
③Judgment frequency offset（Δf）polar,Δf>0,setxtalcapfine 4,on the contrary,setxtalcapfine -4;
④Judgment frequency offset（Δf）polar,Δf>0,setxtalcapfine 2,on the contrary,setxtalcapfine -2;
⑤Judgment frequency offset（Δf）polar,Δf>0,setxtalcapfine 1,on the contrary,setxtalcapfine -1;

Note： The parameters corresponding to the calibration frequency offset command are all decimal relative values, that is, the offset value from the default value. After inputting the command, the actual parameters of the configured frequency offset will be returned and displayed in hexadecimal. The frequency offset calibration value written into efuse or flash is hexadecimal absolute value.

2.1.6 Reading and Writing MAC address

1. wifi_test wlan0 set_mac_addr \\ Write WiFi MAC address to efuse(2 times)\flash (repeat)

   eg： wifi_test wlan0 set_mac_addr 88 00 11 22 33 44 \\ Write WiFi MAC address

2. wifi_test wlan0 get_mac_addr \\ Reading WiFi MAC address
   no parameter

3. wifi_test wlan0 set_bt_mac_addr \\ Write BT MAC address to efuse(2 times)\flash (repeat)

   eg： wifi_test wlan0 set_bt_mac_addr 0A 1C 6B C6 96 7E \\ Write BT MAC address

4. wifi_test wlan0 get_bt_mac_addr \\ Reading BT MAC address
   no parameter

Note: If the WiFi also needs to support P2P and SoftAP at the same time, the MAC addresses of the two chips need to be at least 4 apart.

2.1.7 TXpower setting

1. wifi_test wlan0 rdwr_pwrlvl band mod idx val \\setting tx power gain level
val: decimal

4-1-1: band

	band		mod
2.4G	1	**11b+11a/g**	0
		11n/11ac	1
		11ax	2
5G	2	**11a/g**	0
		11n/11ac	1
		11ax	2

2.4G Rate Group

Fmt\Idx	0	1	2	3	4	5	6	7	8	9	10	11
11b+11a/g	1M	2M	5.5M	11M	6M	9M	12M	18M	24M	36M	48M	54M
11n/ac	MCS0	MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7	MCS8	MCS9		
11ax	MCS0	MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7	MCS8	MCS9	MCS10	MCS11

5G Rate Group

Fmt\Idx	0	1	2	3	4	5	6	7	8	9	10	11
11a/g	NA	NA	NA	NA	6M	9M	12M	18M	24M	36M	48M	54M
11n/ac	MCS0	MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7	MCS8	MCS9		
11ax	MCS0	MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7	MCS8	MCS9	MCS10	MCS11

Note：5G 11a/g is special, if multiple values are written at the same time, the first four are written to -128, which means invalid

pwrlvl setting method：

- To set one of the Rates:

eg: wifi_test wlan0 rdwr_pwrlvl 1 0 3 18 \\ 2.4G 11b 11M TX power: 18dBm

- A method to set multiple Rates in a group：

eg. wifi_test wlan0 rdwr_pwrlvl 1 1 15 15 15 15 15 14 14 14 13 13 \\ 2.4G 11n/ac MCS0-MCS9 TX power: 15dBm 15 dBm 15 dBm 15 dBm 15 dBm 14 dBm 14 dBm 14 dBm 13 dBm 13 dBm

Note：If you want to set multiple rates, you need to set all the rates in the change mode.

2. wifi_test wlan0 rdwr_pwrlvl 0
Read the power gain level, write 0 or no write to achieve the read function

2.1.8 Channel power offset

1. awifi_test wlan0 rdwr_pwrofst band rate ch ofst \\ Setting channel power offset

5-1-1: band\rate\ch\ofst

	band		Rate		ch	ofst
2.4G_ANT0	**1**	**11b**	**0**	**CH1~CH4**	**0**	**-15~15**
				CH5~CH9	**1**	**-15~15**
				CH10~CH13	**2**	**-15~15**
		ofdm_highrate	**1**	**CH1~CH4**	**0**	**-15~15**
				CH5~CH9	**1**	**-15~15**
				CH10~CH13	**2**	**-15~15**
2.4G_ANT1	**2**	**11b**	**0**	**CH1~CH4**	**0**	**-15~15**
				CH5~CH9	**1**	**-15~15**
				CH10~CH13	**2**	**-15~15**
		ofdm_highrate	**1**	**CH1~CH4**	**0**	**-15~15**
				CH5~CH9	**1**	**-15~15**
				CH10~CH13	**2**	**-15~15**
5.8G_ANT0	**3**	**ofdm_highrate**	**0**	**CH36~CH50**	**0**	**-15~15**
				CH51~CH64	**1**	**-15~15**
				CH98~CH114	**2**	**-15~15**
				CH115~CH130	**3**	**-15~15**
				CH131~CH146	**4**	**-15~15**
				CH147~CH166	**5**	**-15~15**
5.8G_ANT1	**4**	**ofdm_highrate**	**0**	**CH36~CH50**	**0**	**-15~15**
				CH51~CH64	**1**	**-15~15**
				CH98~CH114	**2**	**-15~15**
				CH115~CH130	**3**	**-15~15**
				CH131~CH146	**4**	**-15~15**
				CH147~CH166	**5**	**-15~15**

eg. wifi_test wlan0 rdwr_pwrofst 1 1 1 2 \\ 2.4G ANT0, OFDM_highrate,CH5~CH9offest 2

ofst is a signed offset value, the step is 1, corresponding to a power change of 0.5dBm, a maximum of 15, a minimum of -15, and the channel power difference can be optimized by adjusting the response channel compensation value.

Note: pwrofst2x can directly display the current transmit power gain level configuration information without parameters.

Note:2.4G is calibrated in 11b_1M,11g_6M,11g_54M, is calibrated in ch1,ch7,h13.
5G is calibrated in 11a_6M,11a_54M， is calibrated in ch42,ch58,ch106,ch122,ch138,ch155.

2. wifi_test wlan0 rdwr_efuse_pwrofst band rate ch ofst \\ Write Channel power offset value to efuse.

eg. wifi_test wlan0 rdwr_efuse_pwrofst 1 1 1 2 \\ Write 2.4G CH5~CH9 calibration value to efuse.

Note： efpwrofst 0 or later without parameters can read the EFUSE channel power compensation value.

OFDM Rate 类
2.4G

	OFDM												
	BPSK 1/2	BPSK 3/4	QPSK 1/2	QPSK 3/4	16QAM 1/2	16QAM 3/4	64QAM 2/3	64QAM 3/4	64QAM 5/6	256QAM 3/4	256QAM 5/6	1024QAM 3/4	1024QAM 5/6
NON-HT	6M	9M	12M	18M	24M	36M	48M	54M					
HT	MCS0		MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7				
VHT	MCS0		MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7	MCS8	MCS9		
HE	MCS0		MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7	MCS8	MCS9	MCS10	MCS11

5G

	OFDM												
	BPSK 1/2	BPSK 3/4	QPSK 1/2	QPSK 3/4	16QAM 1/2	16QAM 3/4	64QAM 2/3	64QAM 3/4	64QAM 5/6	256QAM 3/4	256QAM 5/6	1024QAM 3/4	1024QAM 5/6
NON-HT	6M	9M	12M	18M	24M	36M	48M	54M					
HT	MCS0		MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7				
VHT	MCS0		MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7	MCS8	MCS9		
HE	MCS0		MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7	MCS8	MCS9	MCS10	MCS11

2.1.9 config document uasge

1.aic_userconfig.txt

Copy the firmware to the /lib/firmware/ directory, change the parameters in the document, and then power off and power on again.

Enable = 0 document does not take effect, enable = 1 document takes effect, and the default value is 1.

(Please refer to 2.1.7 and 2.1.8 above for the meaning of parameters)

```
# txpwr_lvl
enable=1
lvl_11b_11ag_1m_2g4=18
lvl_11b_11ag_2m_2g4=18
lvl_11b_11ag_5m5_2g4=18
lvl_11b_11ag_11m_2g4=18
lvl_11b_11ag_6m_2g4=18
lvl_11b_11ag_9m_2g4=18
lvl_11b_11ag_12m_2g4=18
lvl_11b_11ag_18m_2g4=18
lvl_11b_11ag_24m_2g4=16
lvl_11b_11ag_36m_2g4=16
lvl_11b_11ag_48m_2g4=15
lvl_11b_11ag_54m_2g4=15
lvl_11n_11ac_mcs0_2g4=18
lvl_11n_11ac_mcs1_2g4=18
lvl_11n_11ac_mcs2_2g4=18
lvl_11n_11ac_mcs3_2g4=18
lvl_11n_11ac_mcs4_2g4=16
lvl_11n_11ac_mcs5_2g4=16
lvl_11n_11ac_mcs6_2g4=15
lvl_11n_11ac_mcs7_2g4=15
lvl_11n_11ac_mcs8_2g4=14
lvl_11n_11ac_mcs9_2g4=14
lvl_11ax_mcs0_2g4=18
lvl_11ax_mcs1_2g4=18
lvl_11ax_mcs2_2g4=18
lvl_11ax_mcs3_2g4=18
lvl_11ax_mcs4_2g4=16
lvl_11ax_mcs5_2g4=16
lvl_11ax_mcs6_2g4=15
lvl_11ax_mcs7_2g4=15
lvl_11ax_mcs8_2g4=14
lvl_11ax_mcs9_2g4=14
lvl_11ax_mcs10_2g4=13
lvl_11ax_mcs11_2g4=13
lvl_11a_6m_5g=18
lvl_11a_9m_5g=18
lvl_11a_12m_5g=18
lvl_11a_18m_5g=18
lvl_11a_24m_5g=16
lvl_11a_36m_5g=16
lvl_11a_48m_5g=15
lvl_11a_54m_5g=15
lvl_11n_11ac_mcs0_5g=18
lvl_11n_11ac_mcs1_5g=18
lvl_11n_11ac_mcs2_5g=18
lvl_11n_11ac_mcs3_5g=18
lvl_11n_11ac_mcs4_5g=16
```

```
lvl_11n_11ac_mcs5_5g=16
lvl_11n_11ac_mcs6_5g=15
lvl_11n_11ac_mcs7_5g=15
lvl_11n_11ac_mcs8_5g=14
lvl_11n_11ac_mcs9_5g=14
lvl_11ax_mcs0_5g=18
lvl_11ax_mcs1_5g=18
lvl_11ax_mcs2_5g=18
lvl_11ax_mcs3_5g=18
lvl_11ax_mcs4_5g=16
lvl_11ax_mcs5_5g=16
lvl_11ax_mcs6_5g=14
lvl_11ax_mcs7_5g=14
lvl_11ax_mcs8_5g=13
lvl_11ax_mcs9_5g=13
lvl_11ax_mcs10_5g=12
lvl_11ax_mcs11_5g=12

# txpwr_loss
loss_enable_2g4=0
loss_value_2g4=2        // If this value needs to configure the antenna gain, please set it to a positive value;
                           for power loss, please set it to a negative value.
loss_enable_5g=0
loss_value_5g=5         // If this value needs to configure the antenna gain, please set it to a positive value;
                           for power loss, please set it to a negative value.

# txpwr_ofst
ofst_enable=0
ofst_2g4_ant0_11b_chan_1_4=0
ofst_2g4_ant0_11b_chan_5_9=0
ofst_2g4_ant0_11b_chan_10_13=0
ofst_2g4_ant0_ofdm_highrate_chan_1_4=0
ofst_2g4_ant0_ofdm_highrate_chan_5_9=0
ofst_2g4_ant0_ofdm_highrate_chan_10_13=0
ofst_2g4_ant1_11b_chan_1_4=0
ofst_2g4_ant1_11b_chan_5_9=0
ofst_2g4_ant1_11b_chan_10_13=0
ofst_2g4_ant1_ofdm_highrate_chan_1_4=0
ofst_2g4_ant1_ofdm_highrate_chan_5_9=0
ofst_2g4_ant1_ofdm_highrate_chan_10_13=0
ofst_5g_ant0_ofdm_highrate_chan_42=0
ofst_5g_ant0_ofdm_highrate_chan_58=0
ofst_5g_ant0_ofdm_highrate_chan_106=0
ofst_5g_ant0_ofdm_highrate_chan_122=0
ofst_5g_ant0_ofdm_highrate_chan_138=0
ofst_5g_ant0_ofdm_highrate_chan_155=0
ofst_5g_ant1_ofdm_highrate_chan_42=0
ofst_5g_ant1_ofdm_highrate_chan_58=0
ofst_5g_ant1_ofdm_highrate_chan_106=0
ofst_5g_ant1_ofdm_highrate_chan_122=0
ofst_5g_ant1_ofdm_highrate_chan_138=0
ofst_5g_ant1_ofdm_highrate_chan_155=0

# xtal cap
xtal_enable=0
xtal_cap=24
xtal_cap_fine=31
```

2.aic_powerlimit.txt

Used to independently limit channel power, placed in the same path as the above aic_userconfig.txt, enabling CONFIG_POWER_LIMIT in the driver. The default regions are 'SRRC FCC ETSI JP UNSET', where 'UNSET' indicates 'undefined country code region'.
For example: the five '15's after 'CH01' in Table 1 represent that channel '1' is limited to '15dBm' in five regions.

After executing the command "wifi_test wlan0 country_set **" to switch the country code, it is mapped to "FCC" according to the country code region mapping table in the driver, and then compared with the power values in the aforementioned userconfig file, taking the smaller value as the transmission power value (after enabling txpwr_loss, the power values in userconfig.txt and power_limit.txt will be reduced by the loss_value from the set values).

Note: Currently, only non-signaling mode supports different bandwidth power limits.

Usage example:
Signal example 1:
"wifi_test wlan0 country_set US" switches the country. If the value of channel 1 in table 1 under FCC is 8, the target power on the instrument side will be around 8.

Non-signaling Example 1:
"wifi_test wlan0 country_set CN" switches the country, then execute the corresponding test command mentioned earlier. If the value of channel 38 (36, based on the center frequency) in Table 4 (40M) is 8, the target power at the instrument end will be around 8.

powerlimit.txt：

```
# Table 1:
##    2.4G, 20M,#5#
##    START
##       SRRC  FCC  ETSI  JP  UNSET
CH01     15   15   15    15  15
CH02     16   16   16    16  16
CH03     16   16   16    16  16
CH04     16   16   16    16  16
CH05     16   16   16    16  16
CH06     16   16   16    16  16
CH07     16   16   16    16  16
CH08     16   16   16    16  16
CH09     16   16   16    16  16
CH10     16   16   16    16  16
CH11     16   16   16    16  16
CH12     12   12   12    12  12
CH13     12   12   12    12  12
CH14     NA   NA   NA    12  NA
##    END

# Table 2:
##    2.4G, 40M,#5#
##    START
##       SRRC  FCC  ETSI  JP  UNSET
CH01     NA   NA   NA    NA  NA
CH02     NA   NA   NA    NA  NA
CH03     16   16   16    16  16
CH04     16   16   16    16  16
CH05     16   16   16    16  16
CH06     16   16   16    16  16
CH07     16   16   16    16  16
CH08     16   16   16    16  16
```

CH09 16 16 16 16 16
CH10 16 16 16 16 16
CH11 16 16 16 16 16
CH12 NA NA NA NA NA
CH13 NA NA NA NA NA
CH14 NA NA NA NA NA
END

Table 3:
5G, 20M,#5#
START
SRRC FCC ETSI JP UNSET
5G Band 1
CH36 15 16 16 16 15
CH40 15 16 16 16 15
CH44 15 16 16 16 15
CH48 15 16 16 16 15
5G Band 2
CH52 15 16 16 16 15
CH56 15 16 16 16 15
CH60 15 16 16 16 15
CH64 15 16 16 16 15
5G Band 3
CH100 NA 16 16 16 15
CH104 NA 16 16 16 15
CH108 NA 16 16 16 15
CH112 NA 16 16 16 15
CH116 NA 16 16 16 15
CH120 NA 16 16 16 15
CH124 NA 16 16 16 15
CH128 NA 16 16 16 15
CH132 NA 16 16 16 15
CH136 NA 16 16 16 15
CH140 NA 16 16 16 15
CH144 NA NA 16 16 15
5G Band 4
CH149 16 16 11 NA 11
CH153 16 16 11 NA 11
CH157 16 16 11 NA 11
CH161 16 16 11 NA 11
CH165 16 16 11 NA 11
END

Table 4:
5G, 40M,#5#
START
SRRC FCC ETSI JP UNSET
5G Band 1
CH38 15 16 16 16 15
CH46 15 16 16 16 15
5G Band 2
CH54 15 16 16 16 15
CH62 15 16 16 16 15
5G Band 3
CH102 NA 16 16 16 15
CH110 NA 16 16 16 15
CH118 NA 16 16 16 15

CH126 NA 16 16 16 15
CH134 NA 16 16 16 15
CH142 NA 16 NA 16 15
5G Band 4
CH151 16 16 11 NA 11
CH159 16 16 11 NA 11
END

Table 5:
5G, 80M,#5#
START
SRRC FCC ETSI JP UNSET
5G Band 1
CH42 15 16 16 16 15
5G Band 2
CH58 15 16 16 16 15
5G Band 3
CH106 NA 16 16 16 15
CH122 NA 16 16 16 15
CH138 NA 16 NA NA 15
5G Band 4
CH155 16 16 11 NA 11
END

2.1.10 Antenna Gain Settings

Output Power =(Target Power)-(Antenna Gain Value)

The antenna gain value set will reduce the target power, which itself depends on the country settings, see Country Code Settings. Antenna gain values are not available in all countries.
To set the antenna gain value, you need to set the txpwr_loss in the /firmware/aic8800/aic8800d80/ aic_userconfig_8800d80.txt file.
"# txpwr_loss
loss_enable_2g4=0
loss_value=2
loss_enable_5g=0
loss_value=2 "

loss_enable=1 \\ Antenna gain setting enabled
loss_enable=0 \\ Antenna gain setting disabled
loss_value \\Antenna gain setting value

Eg：If you need 2.4G, 11b 1M has an output power of 10dBm
aic_userconfig_8800d80.txt file txpwr_lvl 11b 1M configuration is as follows:
"# txpwr_lvl
enable=1
lvl_11b_11ag_1m_2g4=18"
txpwr_loss：
"# txpwr_loss
loss_enable_2g4=1
loss_value=8
loss_enable_5g=0
loss_value=2 "

Output Power = (txpwr_lvl) - (txpwr_loss)

Note: In the 802.11ax specification, a device classified as Class A must achieve a transmit power accuracy of +/- 3 dB. Therefore, a transmitter with a 1 dB step size meets this requirement.
The modem features two gain adjustment blocks: coarse and fine. The fine gain adjustment step is 1 dB. The hardware design incorporates a 1 dB DSP module, which the digital front end reuses for the transmitter's fine gain adjustment. Consequently, the fine gain step remains at 1 dB.

2.1.11 Physical information reading

For the information retrieval related to the physical mode, bandwidth, frequency, RSSI, noise, tx power, channel utilization (channel time & busy time), country code, and tx/rx physical layer rate of sta and ap, the sta mode does not require a MAC address, while in ap mode, it is necessary to include the adjacent sta MAC address. If not included, some items may be missing, such as tx/rx rate = 0.

```
./wifi_test wlp3s0 GET_CS_INFO 82:7B:19:02:DC:50
GET_CS_INFO:
phymode=4(0:B 1:G 2:A 3:N 4:AC 5:AX)
bandwidth=0(0:20 1:40 2:80)
freq=5180
rssi=-50
snr=49
noise=-99
txpwr=14
chan busy times=82/102(ms)
coutry code =00
rx nss=2, mcs=8
tx nss=2, mcs=9
```

2.1.12 Reading of production test bottom noise

You need to initiate a scan first, switching to each channel during the scan. When there is no RX on the channel, read the baseline noise value. There needs to be a judgment on the receiving mode, which may have a slight delay. The normal mode should not be opened, but when needed, enable CONFIG_READ_NOISE=y in the Makefile to read it. The two columns of values read are the reference baseline noise values for antenna 0 and 1:

```
./wifi_test wlp3s0 READ_NOISE
```

3. Compilation commands for WIFI_TEST

1. sudo cp aic8800D80X2 /lib/firmware/ -r
2. make //Compile the driver
3. Insert the usb board
4. Enter lsusb， Enter lsusb, and you can see the device with ID368b:8d90 on ubuntu.
5. sudo insmod aic_load_fw.ko testmode=1 , sudo insmod aic8800_fdrv.ko(If you want to switch from the test mode to the normal mode, please drive rmmod wifi and power on again to execute sudo insmod aic_load_fw.ko testmode=0, If using the sdio interface, replace aic_load_fw.ko with aic8800_bsp.ko.)
6. Run wifi_test

   Usage:
   wifi_test <interface> <command> [args]
   wifi_test version - Show version
   export WIFI_TEST_CHIP=<chip_type> - Change chip type
   Available chip types:
   0 - AIC8800D
   1 - AIC8800DCDW
   2 - AIC8800D80
   3 - AIC8800D80X2
   export WIFI_TEST_CHIP=3 switch to the wifi_test compatible with D80X2

Example 1：You can connect to the cable line for testing.
set_tx 1 1 2 7 4000 // chan:1 bw:20m mode:2 rate:mcs7 length:4000byte

```
liruizhe@aic:~/android_driver/USB/driver_fw/drivers/aic8800$ sudo wifi_test wlan0 set_tx 1 0 2 7 4096
set_tx:
done
liruizhe@aic:~/android_driver/USB/driver_fw/drivers/aic8800$
```

Example 2:
set_rx 14 1 // chan:14 bw:40m WiFi RX test start
set_rxstop // WiFi RX test stop
get_rx_result : // A total of 314 packages were received, of which 183 were correct.

```
liruizhe@aic:~/android_driver/USB/driver_fw/drivers/aic8800$ sudo wifi_test wlan0 set_rx 14 1
set_rx:
done
liruizhe@aic:~/android_driver/USB/driver_fw/drivers/aic8800$ sudo wifi_test wlan0 set_rxstop
set_rxstop:
done
liruizhe@aic:~/android_driver/USB/driver_fw/drivers/aic8800$ sudo wifi_test wlan0 get_rx_result
get_rx_result:
done: getrx fcsok=183, total=314
```

Example 3:
Setting frequency offset calibration：
Set_xtal_cap 6 crystal has a register value of 0x16, and a value of 0x18 after setting 1, after calibration, the last value displayed is the value that needs to be configured after calibration.

```
liruizhe@aic:~/android_driver/USB/driver_fw/drivers/aic8800$ sudo wifi_test wlan0 set_xtal_cap 0
set_xtal_cap:
done:xtal cap: 0x10
liruizhe@aic:~/android_driver/USB/driver_fw/drivers/aic8800$ sudo wifi_test wlan0 set_xtal_cap 6
set_xtal_cap:
done:xtal cap: 0x16
liruizhe@aic:~/android_driver/USB/driver_fw/drivers/aic8800$ sudo wifi_test wlan0 set_xtal_cap 1
set_xtal_cap:
done:xtal cap: 0x17
liruizhe@aic:~/android_driver/USB/driver_fw/drivers/aic8800$
```

Set the calibrated value into the hardware efuse：

```
liruizhe@aic:~/android_driver/USB/driver_fw/drivers/aic8800$ sudo wifi_test wlan0 set_freq_cal 17
set_freq_cal:
done: freq_cal: 0x17 (remain:0)
liruizhe@aic:~/android_driver/USB/driver_fw/drivers/aic8800$
```

Example 4：Write the efuse of mac address, and read it after writing.：

```
liruizhe@aic:~/android_driver/USB/driver_fw/drivers/aic8800$ sudo wifi_test wlan0 get_mac_addr
get_mac_addr:
done: get macaddr = 00 : 00 : 00 : 00 : 00 : 00
  (remain:0)
liruizhe@aic:~/android_driver/USB/driver_fw/drivers/aic8800$
```

Note 1:
The usb platform is taken as an example, and sdio platform is similar. CONFIG_USB_SUPPORT=n, CONFIG_SDIO_SUPPORT=y of driver/rwnx _ drv/fullmac/makefile are required. The user's aicrf_test can be run on the client platform.
Note 2:
Ubuntu platform suggests to make network renaming rules, so that the chip of aic8800 will be displayed as wlan0 after lsusb, otherwise it will be renamed with mac address.

```
1 cp  /lib/udev/rules.d/80-net-setup-link.rules   /etc/udev/rules.d/
```

然后执行如下命令，修改刚才复制过来的80-net-setup-link.rules文件：

```
1 sudo vim /etc/udev/rules.d/80-net-setup-link.rules
```

如下图所示，将箭头所指的ID_NET_NAME改成ID_NET_SLOT即可。

```
# do not edit this file, it will be overwritten on update

SUBSYSTEM!="net", GOTO="net_setup_link_end"

IMPORT{builtin}="path_id"

ACTION=="remove", GOTO="net_setup_link_end"

IMPORT{builtin}="net_setup_link"

NAME=="", ENV{ID_NET_NAME}!="", NAME="$env{ID_NET_NAME}"

LABEL="net_setup_link_end"
```

4. Test example:

WIFI TX SISO
11b ANT0
wifi_test wlan0 set_tx 1 0 0 0 1000
wifi_test wlan0 set_ant 1

11b ANT1
wifi_test wlan0 set_tx 1 0 0 0 1000
wifi_test wlan0 set_ant 2

11n-HT20 mcs7 ANT0
wifi_test wlan0 set_tx 1 0 2 7 4000
wifi_test wlan0 set_ant 1

11n-HT40 mcs7 ANT0
wifi_test wlan0 set_tx 1 1 2 7 4000
wifi_test wlan0 set_ant 1

11ac-VHT40 mcs9 ANT0
wifi_test wlan0 set_tx 1 1 4 9 8000
wifi_test wlan0 set_ant 1

11ax-HE80 mcs11 ANT0
wifi_test wlan0 set_tx 1 1 5 11 16000
wifi_test wlan0 set_ant 1

WIFI TX MIMO

11n-HT20 mcs15 ANT0
wifi_test wlan0 set_tx 1 0 2 15 4000

11ac-VHT40 mcs9 ANT0
wifi_test wlan0 set_tx 1 1 4 25 8000

11ax-HE80 mcs11 ANT0
wifi_test wlan0 set_tx 1 1 5 27 16000

WIFI RX

20M

wifi_test wlan0 set_rx 1 0

wifi_test wlan0 set_rxstop

wifi_test wlan0 get_rx_result

40M

wifi_test wlan0 set_rx 1 1

wifi_test wlan0 set_rxstop

wifi_test wlan0 get_rx_result

80M

wifi_test wlan0 set_rx 1 2

wifi_test wlan0 set_rxstop

wifi_test wlan0 get_rx_result