



RDA5981X AT 指令集

版本号	描述	时间
Version1.0	初始版本	2017.09.19

更多信息请访问RDA IOT论坛:

<http://bbs.rdamicro.com>



指令格式

指令类型:

设置指令: AT+<x>=<...>	eg: AT+WRF_DEF=0xDA,0x80
查询指令: AT+<x>=<...>	eg: AT+RRF_DEF=0xDA
执行指令: AT+<x>	eg: AT+RST

返回值:

+ok	执行成功
+ok=<...>	执行成功且有返回信息
+error=err code	执行失败并返回错误码

err code:

-1	无此指令
-2	不允许执行
-3	执行失败
-4	参数有错误

注: 1.指令一律使用大写。

2.参数部分以“,”分隔,字符串部分建议使用引号,防止包含特殊符号时出错。

3.未具体解释的指令均为执行指令,直接执行即可。



基础指令

AT	测试 AT 模式是否使能
AT+H	查看 AT 帮助信息
AT+RST	重启模块
AT+VER	查看软件版本
AT+ECHO	打开或者关闭串口回显，启动后默认为打开
AT+ECHO=1/0	1 表示打开，0 表示关闭
AT+WDBG	调整各模块的 debug level
AT+WDBG=DRV,2	调整 driver debug level，级别为 0~3，默认为 0
AT+WDBG=WPA,2	调整 wpa debug level，级别为 0~3，默认为 0
AT+WDBG=DRVD,1	打开/关闭 driver dump，会打印出收发的数据，1 为打开，0 为关闭，默认为关闭
AT+WDBG=WPAD,1	打开/关闭 WPA dump，会打印出某些数据，1 为打开，0 为关闭，默认为关闭



efuse 指令

指令中开始的 R 表示 Read，W 表示 Write。最后的 E 表示 efuse。所有写入的参数必须为 16 进制，1 个 byte。

AT+RE	读取所有 efuse 内容
-------	---------------

efuse 共 16 个 page，每个 page2 个 bytes。用户可以读出 page2~15 的内容，读出的内容按照 [page2 byte0] [page2 byte1] [page3 byte0] ... [page15 byte1] 排列。

page	byte1	byte0
page0	rda_reserved	
page1	rda_reserved	
page2	usb pid	
page3	usb vid	
page4	mac address[4]	mac address[5]
page5	mac address[2]	mac address[3]
page6	mac address[0]	mac address[1]
page7	mac address[4]_bak	mac address[5]_bak
page8	mac address[2]_bak	mac address[3]_bak
page9	mac address[0]_bak	mac address[1]_bak
page10	crystal_cal_bak1	crystal_cal
page11	crystal_cal_bak3	crystal_cal_bak2
page12	rda_reserved:Bit[12~15] Tx_power: Bit[6~11]:b mode Bit[0~5]:g/n mode	
page13	rda_reserved:Bit[12~15] Tx_power_bak1: Bit[6~11]:b mode Bit[0~5]:g/n mode	
page14	rda_reserved:Bit[12~15] Tx_power_bak2: Bit[6~11]:b mode Bit[0~5]:g/n mode	
page15	rda_reserved:Bit[12~15] Tx_power_bak3: Bit[6~11]:b mode Bit[0~5]:g/n mode	

Table 1 efuse page 分区信息



AT+WTPE	写 tx power 到 efuse
---------	--------------------

AT+WTPE=n1,n2

n1 为 11g/n 模式下的 TX POWER，范围为 0x25~0x64。

n2 为 11b 模式下的 TX POWER，范围为 0x15~0x54。

最多写入四次。

AT+RTPE	从 efuse 读取 tx power
---------	---------------------

返回数据格式与写入数据格式相同。

多次写入 tx power 时，返回最后写入值。

AT+WXCE	写晶体校准值到 efuse
---------	---------------

AT+WXCE=n1

n1 为 1 个 byte，取 bit7~bit1，bit0 无效。

最多可以写入四次。

AT+RXCE	从 efuse 读晶体校准值
---------	----------------

返回的数据格式与写入的数据格式相同。

多次写入晶体校准值时，返回最后写入值。

AT+WMACE	写 MAC 地址到 efuse
----------	-----------------

AT+WMACE=n1,n2,n3,n4,n5,n6

n1 的最低位不能为 0，n1~n6 不能全为 0。

最多写入两次。

AT+RMACE	从 efuse 读 MAC 地址
----------	------------------

返回的数据格式与写入的数据格式相同。

多次写入 MAC 地址时，返回最后写入值。



RF 寄存器通用指令

开始的 R 表示 Read, W 表示 Write。所有的寄存器地址和值必须为 16 进制, channel 可以为 10 进制。

在进行 Read 时, _CUR 表示从寄存器读取, _DEF 表示从 flash 读取。进行 Write 时, _CUR 表示只写入寄存器, _DEF 表示写入寄存器的同时写入 flash。

AT+WRF_CUR	写 RF 寄存器
------------	----------

AT+WRF_CUR=n1,n2

n1,n2 均为 half word。n1 为寄存器地址, n2 为要写入的值。

eg: AT+WRF_CUR=0xDA,0x80

AT+RRF_CUR	读 RF 寄存器
------------	----------

AT+RRF_CUR=n1

n1 为寄存器地址, half word。返回的数据格式与写入的数据格式相同。

eg: AT+RRF_CUR=0xDA

AT+WRFAC_CUR	写 RF 寄存器, 此寄存器与 channel 有关
--------------	----------------------------

AT+WRFAC_CUR=n1,n2,n3,...,n15

n1~n15 均为 half word。n1 为寄存器地址, n2~n15 依次为 channel1~14 的值。

eg: AT+WRFAC_CUR=0x8A,0x69A0,0x69A0,...,0x6820

AT+RRFAC_CUR	读 RF 寄存器, 此寄存器与 channel 有关
--------------	----------------------------

AT+RRFAC_CUR=n1

n1 为寄存器地址, half word。返回的数据格式与写入的数据格式相同。

eg: AT+RRFAC_CUR=0x8A



AT+WRFSC_CUR	写 RF 寄存器，只写某个 channel 的值
--------------	--------------------------

AT+WRFSC_CUR=n1,n2,n3

n1 和 n3 为 half word。n1 为寄存器地址，n2 为 channel，n3 为要写入的值。

eg: AT+WRFSC_CUR=0x8A,1,0x69A0

AT+RRFSC_CUR	读 RF 寄存器，只读某个 channel 的值
--------------	--------------------------

AT+RRFSC_CUR=n1,n2

n1 为寄存器地址，half word。n2 为 channel。返回的数据格式与写入的数据格式相同。

eg: AT+RRFSC_CUR=0x8A,1

AT+WRF_DEF	同时写 RF 寄存器和 flash
------------	-------------------

AT+WRF_DEF=n1,n2

n1,n2 均为 half word。n1 为寄存器地址，n2 为要写入的值。

eg: AT+WRF_DEF=0xDA,0x80

AT+RRF_DEF	从 flash 读取 RF 寄存器值
------------	--------------------

AT+RRF_DEF=n1

n1 为寄存器地址，half word。返回的数据格式与写入的数据格式相同。

eg: AT+RRF_DEF=0xDA



AT+WRFAC_DEF	同时写 RF 寄存器和 flash, 此寄存器与 channel 有关
--------------	-------------------------------------

AT+WRFAC_DEF=n1,n2,n3,...,n15

n1~n15 均为 half word。n1 为寄存器地址，n2~n15 依次为 channel1~14 的值。

eg: AT+WRFAC_DEF=0x8A,0x69A0,0x69A0,...,0x6820

AT+RRFAC_DEF	从 flash 读取 RF 寄存器值，此寄存器与 channel 有关
--------------	-------------------------------------

AT+RRFAC_DEF=n1

n1 为寄存器地址，half word。返回的数据格式与写入的数据格式相同。

eg: AT+RRFAC_DEF=0x8A

AT+WRFSC_DEF	同时写 RF 寄存器和 flash，只写某个 channel 的值
--------------	-----------------------------------

AT+WRFSC_DEF=n1,n2,n3

n1 和 n3 为 half word。n1 为寄存器地址，n2 为 channel，n3 为要写入的值。

eg: AT+WRFSC_DEF=0x8A,1,0x69A0

AT+RRFSC_DEF	从 flash 读取 RF 寄存器值，只读某个 channel 的值
--------------	------------------------------------

AT+RRFSC_DEF=n1,n2

n1 为寄存器地址，half word。n2 为 channel。返回的数据格式与写入的数据格式相同。

eg: AT+RRFSC_DEF=0x8A,1



AT+DRF	dump flash 中保存的 RF 寄存器及其对应的值
--------	------------------------------

AT+ERF	擦除 flash 中保存的 RF 寄存器及其对应的值
--------	----------------------------

AT+ERF=n1

n1 为寄存器地址，half word。

eg: AT+ERF=0xDA

AT+DRFAC	dump flash 中保存的与 channel 相关 RF 寄存器及其对应的值
----------	--

AT+ERFAC	擦除 flash 中保存的与 channel 相关 RF 寄存器及其对应的值
----------	--

AT+ERFAC=n1

n1 为寄存器地址，half word。

eg: AT+ERFAC=0x8A



PHY 寄存器通用指令

开始的 R 表示 Read，W 表示 Write。所有的寄存器地址和值必须为 16 进制，channel 可以为 10 进制。

在进行 Read 时，_CUR 表示从寄存器读取，_DEF 表示从 flash 读取。进行 Write 时，_CUR 表示只写入寄存器，_DEF 表示写入寄存器的同时写入 flash。

AT+WPHY_CUR	写 PHY 寄存器
-------------	-----------

AT+WPHY_CUR=n1,n2

n1,n2 均为 word。n1 为寄存器地址，n2 为要写入的值。

eg: AT+WPHY_CUR=0x11F,0x45

AT+RPHY_CUR	读 PHY 寄存器
-------------	-----------

AT+RPHY_CUR=n1

n1 为寄存器地址，word。返回的数据格式与写入的数据格式相同。

eg: AT+RPHY_CUR=0x11F

AT+WPHY_DEF	同时写 PHY 寄存器和 flash
-------------	--------------------

AT+WPHY_DEF=n1,n2

n1,n2 均为 word。n1 为寄存器地址，n2 为要写入的值。

eg: AT+WPHY_DEF=0x11F,0x45

AT+RPHY_DEF	从 flash 读取 PHY 寄存器值
-------------	---------------------

AT+RPHY_DEF=n1

n1 为寄存器地址，word。返回的数据格式与写入的数据格式相同。

eg: AT+RPHY_DEF=0x11F



AT+WTPOFS_DEF	写 11g 与 11n Tx Power 偏置到 flash
---------------	--------------------------------

AT+WTPOFS_DEF=n1

n1 为 11g 与 11n 的偏置，取值范围 0x0~0xF。

eg: AT+WTPOFS_DEF=0x5

AT+RTPOFS_DEF	从 flash 读取 11g 与 11n Tx Power 偏置
---------------	----------------------------------

AT+DPHY	dump flash 中保存的 PHY 寄存器及其对应的值
---------	-------------------------------

AT+EPHY	擦除 flash 中保存的 PHY 寄存器及其对应的值
---------	-----------------------------

AT+EPHY=n1

n1 为寄存器地址，half word。

eg: AT+EPHY=0x11F



读写 MAC 地址指令

开始的 R 表示 Read，W 表示 Write。读写 MAC 地址比较特殊，_DEF 只写入 flash 和只从 flash 读取。

MAC 地址必须为 16 进制。

AT+WMAC_DEF	写 MAC 地址，只写入 flash
-------------	--------------------

AT+WMAC_DEF=n1,n2,n3,n4,n5,n6

n1~n6 为 MAC 地址，均为 1 个 byte。

eg: AT+WMAC_DEF= 0x64,0x20,0xAB,0xC1,0x1D,0x34

AT+RMAC_DEF	读 MAC 地址，从 flash 中读取
-------------	----------------------



HUT 模式下收发测试指令

通过以下指令，可以进入测试模式。指令中所有参数都是 10 进制。

AT+TXSTART	开始 TX 测试
------------	----------

AT+TXSTART=n1,n2,n3,n4,n5,n6

n1: 主信道，取值范围 1~14。

n2: 模式，0 代表 b/g mode，1 代表 11n green field mode，2 代表 11n mixed mode。

n3: 信道带宽，0 代表 20M，1 代表 40M。

n4: 信号带宽，0 代表全 40M，1 代表主信道为 upper，2 代表 20M，3 代表主信道为 lower。

n5: 速率，11b/g 模式取值为 1 2 5 6 9 11 12 18 24 36 48 54。11n 模式取值为 0~7，代表 mcs0 到 mcs7。

n6: 发送包长，单位为 byte。

注意：在信道带宽为 40m 时，若主信道为 1~4，信号带宽只能选择 0 或者 3，若主信道为 10~13，信号带宽只能选择 1。

eg: AT+TXSTART=1,0,0,2,54,1024

AT+TXSTOP	停止 TX 测试
-----------	----------

AT+TXRESTART	按照上次的配置重新开始 TX 测试
--------------	-------------------

仅在上一次测试为 TX 测试才有效。



AT+RXSTART	开始 RX 测试
------------	----------

AT+RXSTART=n1,n2,n3,n4

n1~n4 的定义与 TX 测试相同，RX 测试不需要设置速率和包长。

eg: AT+RXSTART=1,0,0,2

AT+RXRESULT	获取 RX 测试结果
-------------	------------

当 AT+RXSTART= n1,n2,n3,n4 指令下发后，会持续统计 RX 测试结果，每下发一次 AT+RXRESULT 指令，就会上报一次 RX 测试结果。

AT+RXSTOP	停止 RX 测试
-----------	----------

AT+RXRESTART	按照上次的配置重新开始 RX 测试
--------------	-------------------

仅在上一次测试为 RX 测试才有效。



寄存器介绍

0x8A 寄存器

0x8A 寄存器是 RF 寄存器,共 16bit, 其中 bit7~bit10 用来调节模拟增益, 其他位禁止改动。
0x8A 默认为 0x6820。

15	0	tmx_div2_bias_bit[2:0]
14	1	
13	1	
12	0	tmx_bias2p4_bit[1:0]
11	1	
10	0	tmx_cap_bit[3:0]
9	0	
8	0	
7	0	
6	0	tmx_cal_clk_edge_bit
5	1	tmx_cb_bit[2:0]
4	0	
3	0	
2	0	tmx_pll_test
1	0	tmx_cal_en
0	0	tmx_cal_swap

Register DataHex

可以通过配置 channel1~14 的 0x8A 寄存器值将信道功率调平。在调节 0x8A 寄存器时, 数字增益保持不变。确定好一组 0x8A 寄存器值后, 就可以通过校准数字增益来达到目标功率。校准时需要将这组 0x8A 寄存器值写入 flash, 模块上电后驱动会调用相应信道的 0x8A 值配入 firmware。



0xDA 寄存器

0xDA 寄存器是 RF 寄存器,共 16bit, 其中 bit9~bit15 用来调节晶体频偏, 其他位禁止改动。
0xDA 默认为 0x80CF。AT 指令接口已经将低 8 位固定为 0xCF, 所以通过 AT 指令修改 0xDA 值, 只需填写高 8 位的值即可。

eg. AT+WRF_CUR=0xDA,0x80

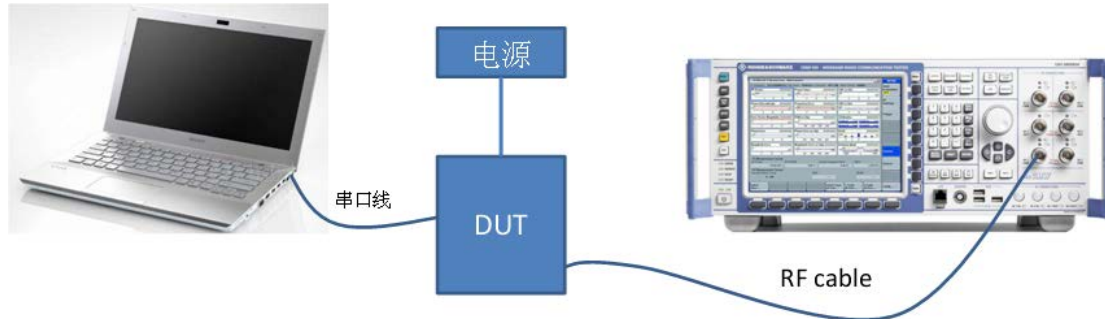
15	1	xtal_capbank_bit[6:0]
14	0	
13	0	
12	0	
11	0	
10	0	
9	0	
8	0	reserved
7	1	xtal_vamp_ibit_xosc[3:0]
6	1	
5	0	
4	0	
3	1	xtal_wf_bbpll_26m_en
2	1	xtal_intf_26m_en
1	1	xtal_usb_26m_en
0	1	xtal_pmu_26m_en

Register: DA DataHex: 80CF

Write Data Read data

注: 由于 bit8 为保留位, 且值始终为 0, 所以写入 0xDA 高 8 位值为偶数。

校准环境



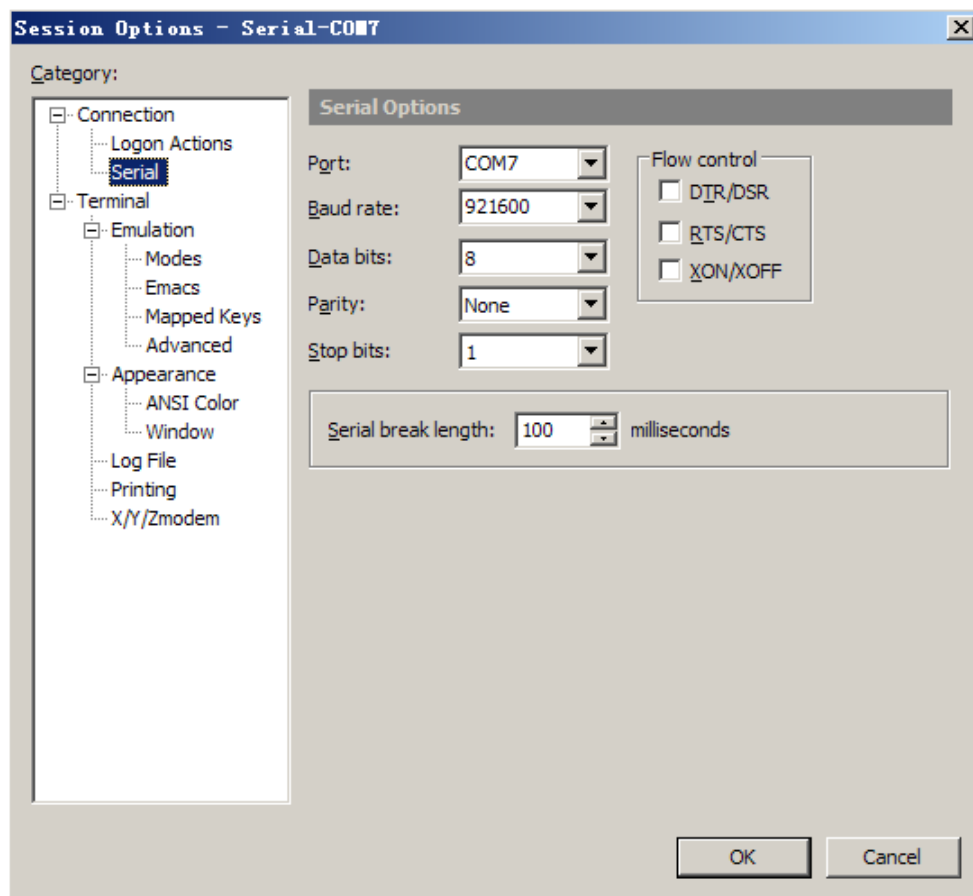
测试设备:

1. CMW500(或其他厂商 WLAN 测试仪表)
2. 直流稳压电源
3. PC
4. 串口线、RF 测试电缆

串口配置

打开串口工具 SecureCRT

串口配置如下:





Port: 选择电脑识别串口
Baud rate: 921600
Data bits: 8
Parity: None
Stop bits: 1

连接好串口后在命令编辑栏输入 AT，返回 OK 说明 AT 模式已经使能。

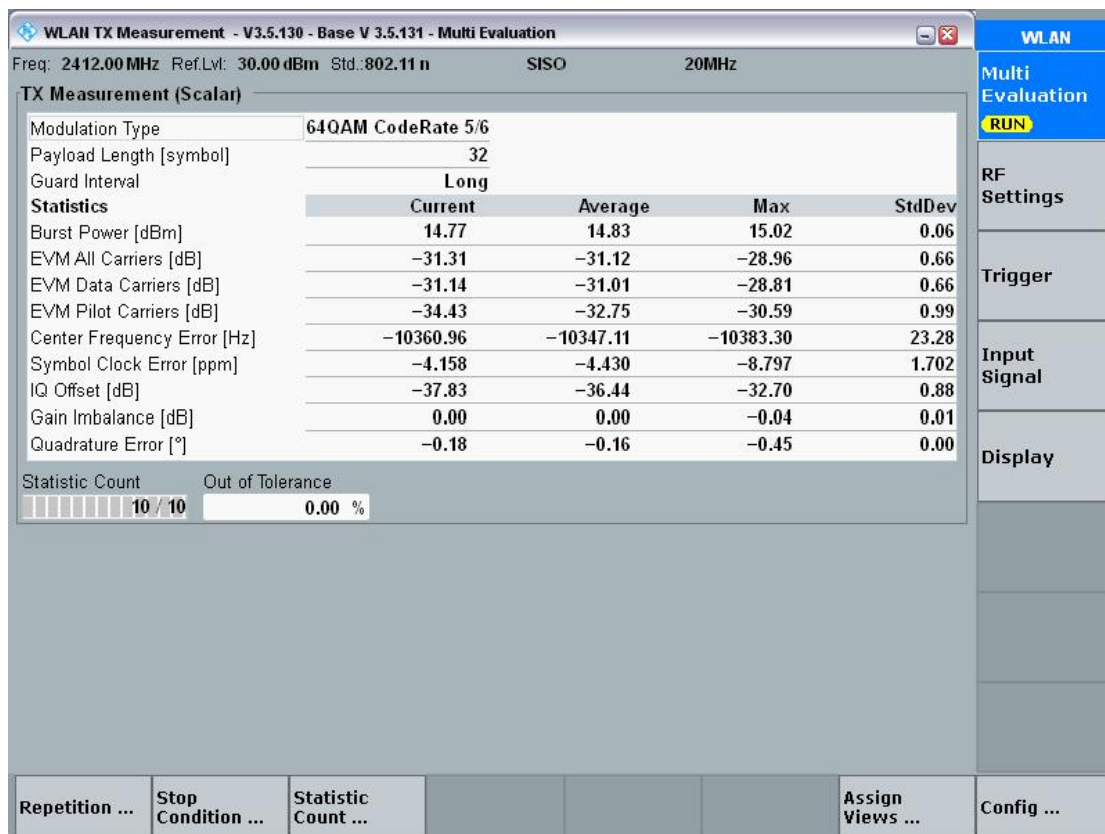
The screenshot shows a SecureCRT terminal window titled 'Serial-COM7 - SecureCRT'. The terminal displays a series of log messages from a device, including 'Start uarthur test...', 'rda5981_flash_read_mac_addr,Enter', 'rda5981_read_user_data,Enter, flag:0x1', 'rda5981_flash_read_mac_addr,Done(ret:0)64:20:ab:c1:d:34', 'wland_reg_func,wland_reg_func', 'rx_11nMixed_bw20_sig20_wmm_test', 'wland_task,wland_task', 'wland_sta_init,wland_sta_init', 'wland_preinit_cmds,wland_preinit_cmds', 'send msg : 3e 40 57 00 3c 00 0c 30 06 64 20 ab c1 d 34 0e 10 02 14 00 0c 10 02 14 00 0d 10 02 14 00 37', 'frame_len=62', 'recv frame : 0a 30 52 00 08 00 05 00 01 01', 'frame_len=10', 'mbed_recv_pkt,WLAND_WID_MSG_RESP', 'wland_preinit_cmds2,wland_preinit_cmds2', 'rda5981_read_user_data,Enter, flag:0x4000', 'rda5981_read_user_data,Enter, flag:0x8000', 'rda5981_read_user_data,Enter, flag:0x10000', 'rda5981_read_user_data,Enter, flag:0x20000', 'send msg : 0d 40 57 01 0b 00 21 20 04 cf 82 da 10', 'frame_len=13', 'recv frame : 0a 30 52 01 08 00 05 00 01 01', 'frame_len=10', 'mbed_recv_pkt,WLAND_WID_MSG_RESP'. Below these messages, the user has entered 'AT' and the terminal has responded with '+ok'. The terminal window also shows a status bar at the bottom with 'Ready', 'Serial: COM7, 921600', '28, 1', '33 Rows, 103 Cols', 'VT100', and 'CAP NUM'.

```
Start uarthur test...
rda5981_flash_read_mac_addr,Enter
rda5981_read_user_data,Enter, flag:0x1
rda5981_flash_read_mac_addr,Done(ret:0)64:20:ab:c1:d:34
wland_reg_func,wland_reg_func
rx_11nMixed_bw20_sig20_wmm_test
wland_task,wland_task
wland_sta_init,wland_sta_init
wland_preinit_cmds,wland_preinit_cmds
send msg : 3e 40 57 00 3c 00 0c 30 06 64 20 ab c1 d 34 0e 10 02 14 00 0c 10 02 14 00 0d 10 02 14 00 37
frame_len=62
recv frame : 0a 30 52 00 08 00 05 00 01 01
frame_len=10
mbed_recv_pkt,WLAND_WID_MSG_RESP
wland_preinit_cmds2,wland_preinit_cmds2
rda5981_read_user_data,Enter, flag:0x4000
rda5981_read_user_data,Enter, flag:0x8000
rda5981_read_user_data,Enter, flag:0x10000
rda5981_read_user_data,Enter, flag:0x20000
send msg : 0d 40 57 01 0b 00 21 20 04 cf 82 da 10
frame_len=13
recv frame : 0a 30 52 01 08 00 05 00 01 01
frame_len=10
mbed_recv_pkt,WLAND_WID_MSG_RESP
AT
+ok
```



TX Power 测试及校准

- 通过 AT+TXSTART 指令，使 RDA5981X 模块处于发射状态。
eg. AT+TXSTART=1,2,0,2,7,1024 模块处于 11n 20M Mixed MCS7 发射模式
- 仪表端读数如下：



其中 Burst Power 既是 TX Power 的测量值，EVM、Freq Error、Symbol Clock Error、IQ Offset、Gain Imbalance 以及 Quadrature Error 都是 TX 测试比较关注的内容

- RDA5981X TX Power 数字增益寄存器类型为 PHY 寄存器，地址如下：
0x11F 调节 11g/n TX Power 数字增益
0x120 调节 11b TX Power 数字增益

若当前 TX Power 未在所需范围内，可以通过如下方式调节：

11g/n AT+WPHY_CUR=0x11F,n1
11b AT+WPHY_CUR=0x120,n1

对于 11b，在目标功率（16~18）dbm 区间内，寄存器每改变 1bit，功率值变化 0.15db

对于 11g/n，在目标功率（12~14）dbm 区间内，寄存器每改变 1bit，功率值变化 0.12db

注：寄存器每改变 1bit 对应的功率变化并不是严格等于以上值

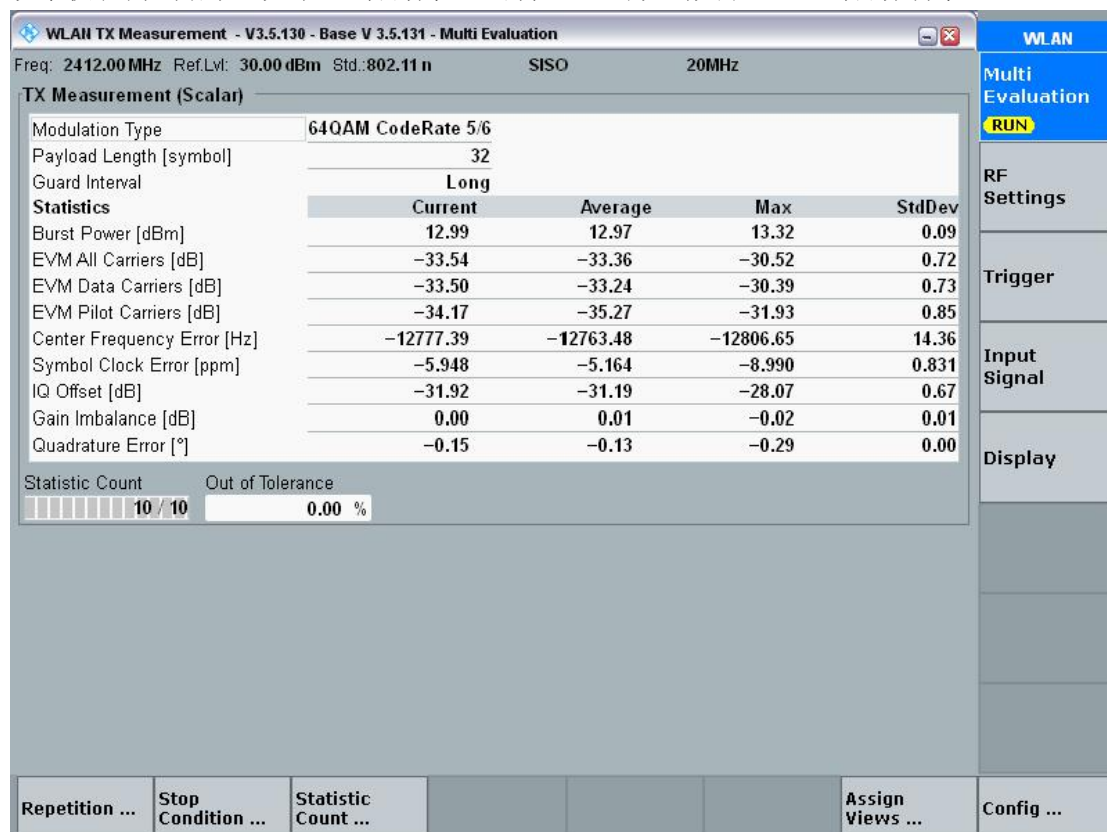


例如，如上测试模块 TX Power 为 14.83dbm，通过 AT+RPHY_CUR=0x11F 可得知当前 0x11F 寄存器值为 0x45，我们希望将 TX Power 降至 13dbm，那么寄存器相应降低 15bit，TX Power 基本就接近目标功率了。

可以通过如下指令实现：

AT+WHY_CUR=0x11F,0x36

观察仪表测试结果，如未达到目标值，可再通过此方法微调直至达到目标功率

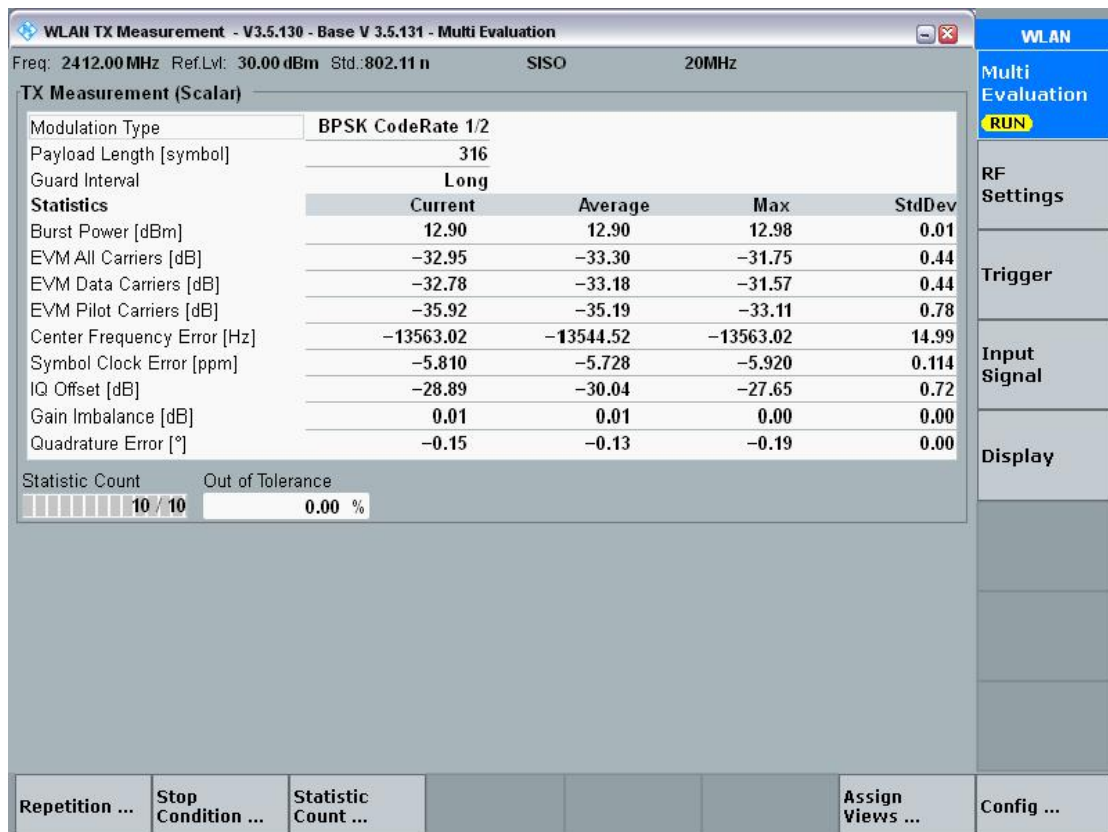


- 当 TX Power 校准至目标功率值后，将寄存器值写入 flash，指令如下：
11g/n AT+WPHY_DEF=0x11F,n1
11b AT+WPHY_DEF=0x120,n1
- 当 11g/n 目标功率不一致时，可以通过设置偏置来实现，该偏置值需要在校准时写入 flash，指令如下：
AT+WTPOFS_DEF=n1
n1 为 11g 与 11n 数字增益差值，11g 在 11n 数字增益的基础上加上此偏置值即为 11g 的数字增益。n1 为 16 进制，取值范围 0x0~0xF。



晶体频偏测试及校准

1. 通过 AT+TXSTART 指令，使 RDA5981X 模块处于发射状态。
eg. AT+TXSTART=1,2,0,2,0,1024 模块处于 11n 20M Mixed MCS0 发射模式
建议在低速率下去校准晶体频偏
2. 仪表端读数如下：

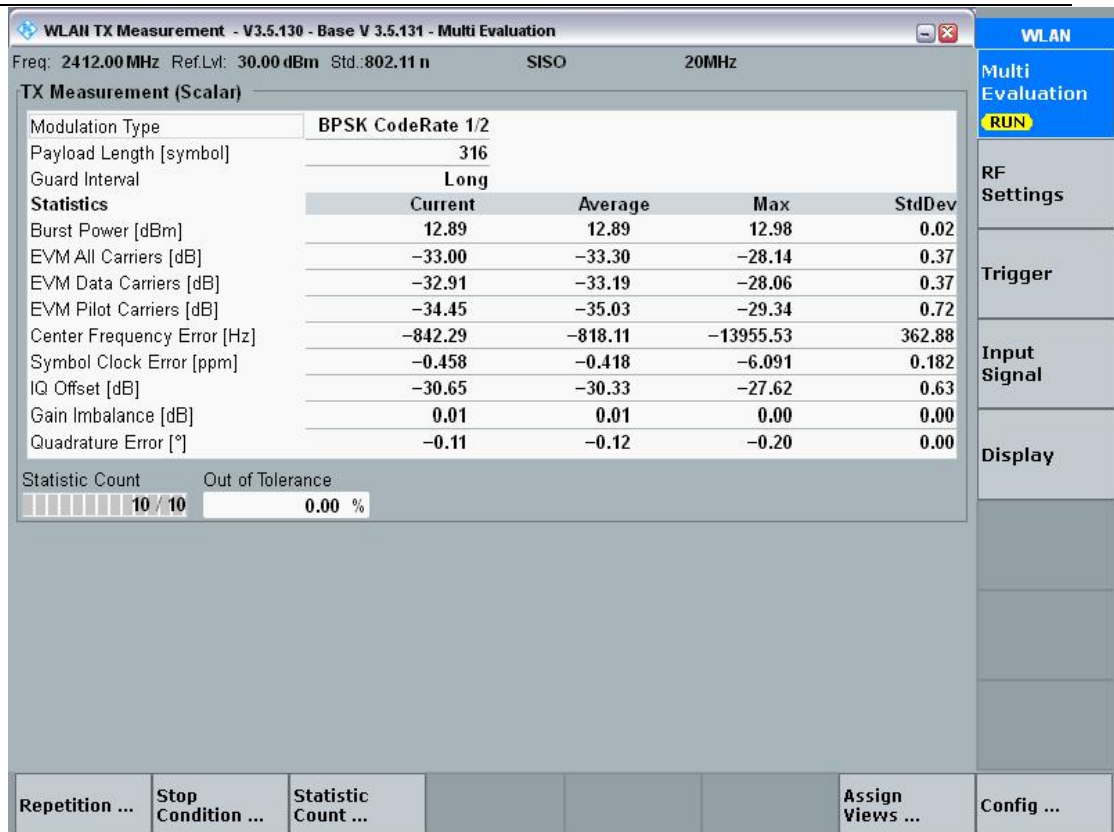


Symbol Clock Error 即为当前晶体频偏

3. 可以通过 AT+WRF_CUR 来调节晶体频偏，指令如下：
AT+WRF_CUR=0xDA,n1
其中 n1 是寄存器值，为十六进制数，范围 0x02~0xFE，通常默认 0x80

如果频偏为负偏，可以通过增加寄存器值来修正频偏，使频偏接近 0ppm。如果频偏为正偏，可以通过减小寄存器值来使频偏接近 0ppm。

例如上面测试晶体频偏为-5ppm，通过 AT+RRF_CUR=0xDA 得知当前 0xDA 寄存器值为 0x80，需要增加寄存器值来使频偏接近 0ppm，我们可以通过 AT+WRF_CUR=0xDA,0x90 修改 0xDA 寄存器值为 0x90，观察仪器读数，直至频偏接近 0ppm。



注：晶体频偏变化与寄存器值变化值对应关系与晶体型号有关，需要前期对晶体进行测量得出。

4. 当晶体频偏校准至 0ppm 后，将 0xDA 寄存器值写入 flash，指令如下：

AT+WRF_DEF=0xDA,n1