

TG7100C 射频性能 评估手册

版本: 2.27

版权 @ 2023

1 版本记录	4
2 概述	6
3 下载开发烧录软件工具包	8
4 烧写/下载测试固件	9
4.1 IOT 模组 (带 Flash) 烧写测试固件	9
4.2 SDIO 模组 (不带 Flash) 下载测试固件	11
5 运行测试固件	15
6 频偏补偿设置	16
7 发送设置	18
7.1 Channel 和 Power 设置	18
7.2 发送数据包模式设置	19
8 接收设置	21
9 BLE 测试	22
10 PDS/DTIM 设置	23
11 HBN 设置	24
12 CW 测试模式	25
12.1 WiFi CW 测试	25
12.2 BLE CW 测试	26
13 DTS 文件介绍	27
14 发送功率温度补偿	29
14.1 设置功率温度补偿	29
15 验证 RF 产测参数	31
15.1 验证频偏校准数值	31
15.2 验证功率校准数值	32
16 串口通信命令	34
16.1 Shakehand	34

16.2	TX on/off	34
16.3	TX modulation	34
16.4	2.4g channel	35
16.5	2.4g tx power	35
16.6	TX frame length	36
16.7	TX frequency	36
16.8	PDS	36
16.9	HBN	36
16.10	RX	36
16.11	Get MFG FW version	36
16.12	Get MFG FW building infomation	37
16.13	Get current power level	37
16.14	Get current channel	37
16.15	Get current tx status	37
16.16	Get tx frequency	37
16.17	Get cap code	37
16.18	Get MFG mode	37
16.19	Set cap code	38
16.20	Set MFG Test(CW) mode	38
16.21	Write data to efuse	38
16.22	Save calibration parameters to efuse	39
16.23	Save calibration parameters to flash	42
16.24	Reset MFG FW	42
16.25	Set tx duty	43
16.26	Get tx duty	43
16.27	BLE Test	43

表 1.1: 修改记录

版本	更新内容
V2.27	增加 BLE CW 测试
V2.26	移除量产相关介绍，量产相关写入产测文档
V2.25	修改 mfg bin 下载方式
V2.24	修改 bin 文件的路径
V2.23	增加 SDIO 模组的下载和量产使用说明
V2.22	增加 DTS 文件的命令发送说明
V2.21	修改 MFG bin 文件的路径
V2.20	添加温补参数相关使用说明
V2.19	添加 RAM 版本 (SDIO 透传模组)MFG 固件的相关使用说明
V2.18	更新评估板硬件描述
V2.17	增加射频性能测试固件与应用固件的切换说明
V2.16	增加射频性能测试在生产中的使用说明
V2.15	增加 EFUSE 读写的相关描述
V2.14	增加温补功能的相关描述
V2.13	增加将产测校准参数写入 Flash 功能
V2.12	增加启用 efuse tx power offset 校准功能命令
V2.11	增加设置 BLE Tx Power 及 default 值功能
V2.10	增加 BLE 和 Auto(Default) TX power 功能
V2.9	将 MFG 功能集成到开发烧录工具中
V2.8	增加 TX Duty 选择，去掉固件下载功能
V2.7	增加 CW 测试模式，在该测试模式下，只能设定 Power 和 Channel
V2.6	去掉 Ch14 的设置，增加 Misc Cfg Get 功能获取 Channel, Power, Capcode
V2.5	增加 Shakehand (H) 命令，增加读取 efuse power offset 和 Cap code 并设定功能

表 1.1: 修改记录 (continued)

版本	更新内容
V2.4	增加 Reset 命令
V2.3	增加 Efuse Load 和 Save 指令，用于保存校准参数到 Efuse 时候的校验
V2.2	去除 MFG 图形界面工具中数据包长，发送频率，制式参数等冗余设定，简化用户使用步骤

RF 性能测试工具 (RF MFG) 是 TG 提供的用于 RF 评估测试的工具，包含测试工具和测试镜像 (MFG Firmware) 两部分。测试工具界面如下图所示。

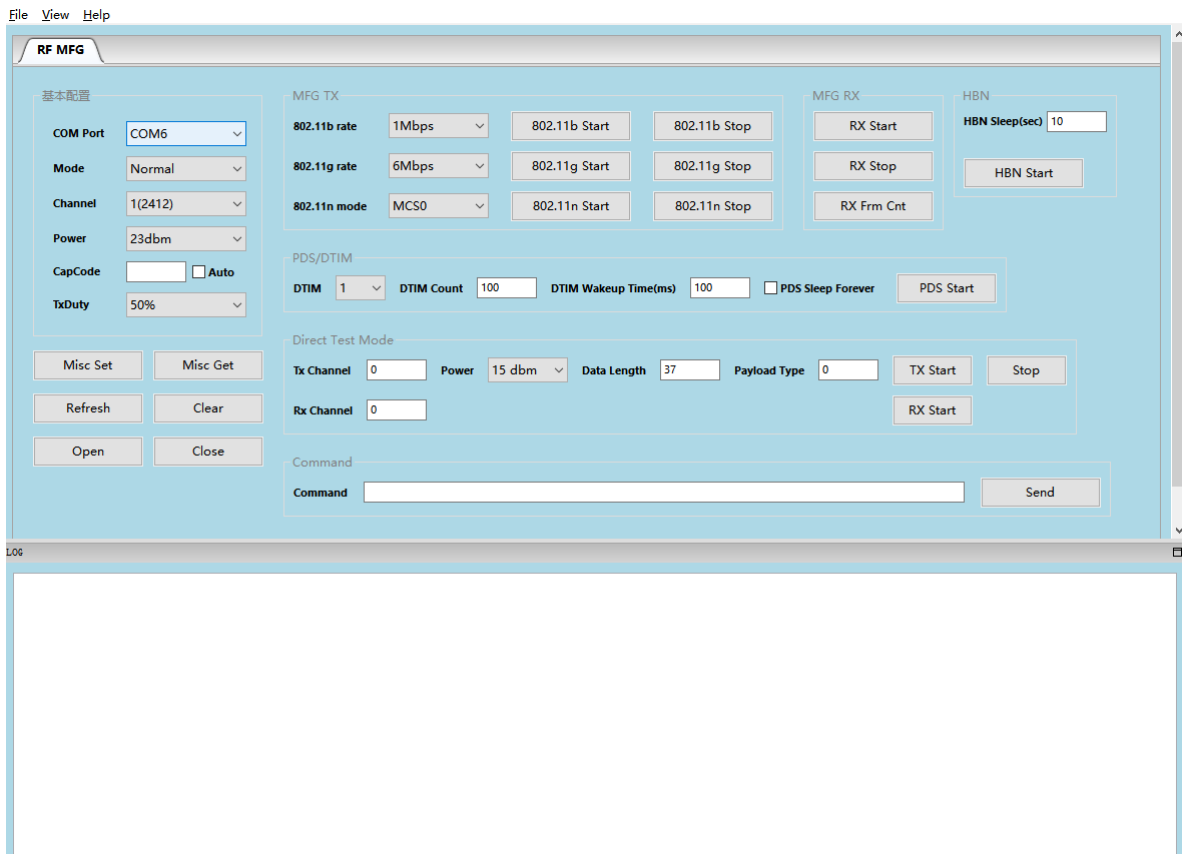


图 2.1: 工具界面图

RF 性能测试工具 (RF MFG) 可以实现的功能包括:

- WiFi 数据包发送
- WiFi 数据包接收

- BLE 数据包发送 (具有 BLE 功能的芯片)
- BLE 数据包接收 (具有 BLE 功能的芯片)
- 芯片 PDS(Power down sleep) 测试
- 芯片 HBN(Hibernate) 测试
- RF 产测
- 训练功率和频偏经验值
- RF 校准验证

下载开发烧录软件工具包

如果用户没有开发烧录工具，可以通过 TG Flash Environment，获取开发烧录软件工具包，在该工具包中，包含了 RF 测试固件，RF 测试固件烧写工具，RF 测试工具等。工具包解压后的效果如图所示。它是客户开发 TG 各种类型芯片使用到的工具集。

名称	修改日期	类型	大小
common	2020/8/10 9:51	文件夹	
openocd	2020/8/10 9:51	文件夹	
tg7100c	2020/8/10 9:51	文件夹	
JLink.exe	2020/8/10 9:39	应用程序	273 KB
JLinkARM_32.dll	2020/8/10 9:39	应用程序扩展	13,988 KB
openocd.exe	2020/8/10 9:39	应用程序	4,579 KB
TGFlashEnv.exe	2020/8/10 9:47	应用程序	24,098 KB

图 3.1: 开发烧录软件工具包

4.1 IOT 模组 (带 Flash) 烧写测试固件

将芯片 Boot 引脚跳到高电平，使其从 UART 启动，即可通过 Flash 烧写工具，将 RF 的测试固件烧写到 Flash 中。测试固件成功烧录后，将 Boot 引脚跳到低电平，按下复位键就可以启动测试固件程序进行 RF 测试了。

在开发烧录软件工具包中，包含了各个芯片的 RF 测试固件。下面以 TG7100C 的 IOT 模组评估套件烧写为例，介绍烧写过程。

套件由 IOT 模组和模组底板组成，模组底板使用 USB Type-C 接口供电同时带有一颗 FT 的 USB 转串口芯片，USB 转串口与模组的连接关系是：

- TXD: 与模组的 RXD 相连
- RXD: 与模组的 TXD 相连

当模组连接到 PC 后，会在 PC 的设备管理器出现两个 USB 转串口，并且这两个 COM 号是相邻的，与芯片的 UART 连接的是其中小号的串口。如果模组连接到 PC 后，没有自动安装驱动，请到<https://www.ftdichip.com/Drivers/VCP.htm>下载驱动自行安装。

连接模组后，首先运行 TGFlashEnv.exe，在 Chip Type 中选择 TG7100C，进入如下烧写界面。

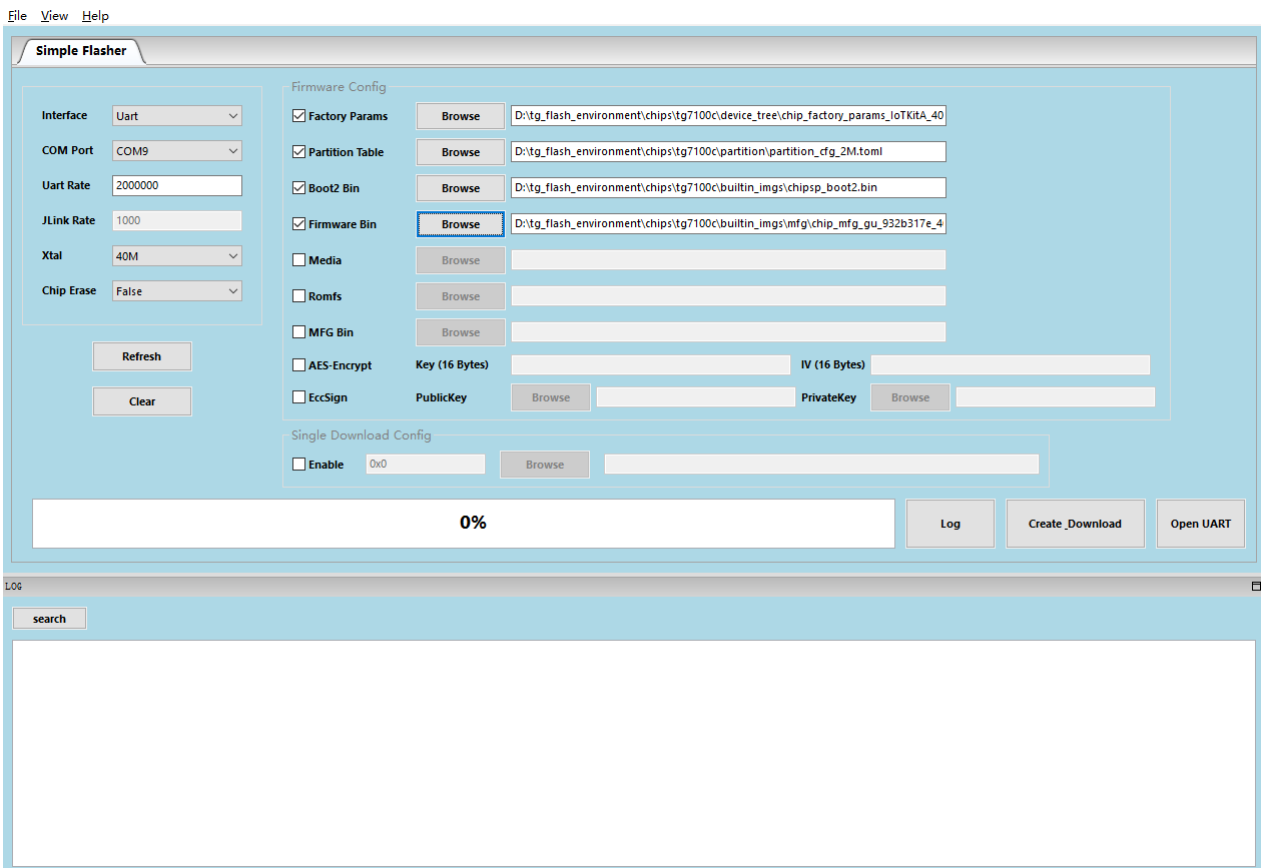


图 4.1: 烧写界面

在左侧通信接口设置中：

- **Interface:** 用于选择烧录的通信接口，这里选择 **uart** 进行烧写
- **COM Port:** 当选择 **UART** 进行烧写的时候这里选择与芯片连接的 **COM** 口号，可以点击 **Refresh** 按钮进行 **COM** 号的刷新
- **Uart Rate:** 当选择 **UART** 进行烧写的时候，填写波特率，可以填写 **2M** 即 **2000000**
- **Board:** 选择所使用的板子型号，这里选择 **IoTKitA**，当板子选定后，**Xtal** 和 **Chip/Flash** 会自动更新成与板子匹配的默认值，当然用户也是可以再次更改的
- **Xtal:** 用于选择板子所使用的晶振类型，对于评估板，这里选择 **40M**
- **Chip Erase:** 默认设置为 **False**，即下载时不擦除 **Flash**

其它项使用默认配置即可。

在右侧烧录镜像配置，分别选择：

- **分区表:** 使用烧写工具目录下的对应芯片型号 **partition** 目录下的分区表，本例中使用 **tg7100c/partition/partition_cfg_2M.toml**
- **Boot2:** 使用烧写工具目录下的对应芯片型号 **builtin_imgs** 目录下的 **Boot2**，本例中使用 **tg7100c/builtin_imgs/chipsp_**

boot2.bin

- 固件程序:使用烧写工具目录下的对应芯片型号 `builtin_imgs` 目录下的 `mfg`,本例中选择的是 `tg7100c/builtin_imgs/mfg/chip_mfg_gu_40m.bin`,

根据上述配置, 设置好 Dev Cube 以后, 将芯片配置成 UART 启动模式, 即可开始烧写。

将芯片配置成 UART 启动模式方法如下:

- 按住模组上的按键 SW2 不松, 同时按一下按键 SW1
- 松开 SW1 和 SW2

完成上述芯片启动设定后, 点击 **Create&Download** 按钮, 完成固件程序的烧录。烧录成功的示意如下。

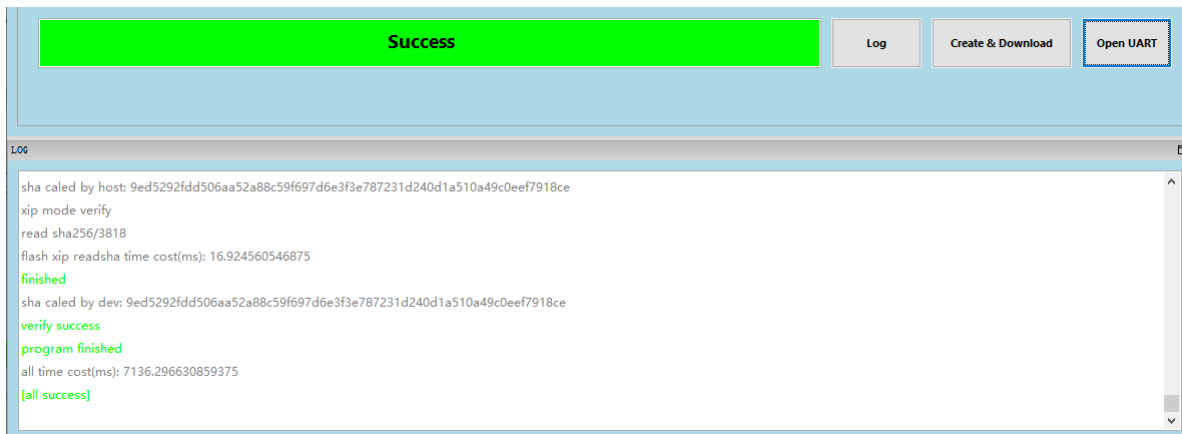


图 4.2: 烧录成功界面

4.2 SDIO 模组 (不带 Flash) 下载测试固件

4.2.1 模组带 UART 接口

如果模组带有 UART 接口, 可以使用 UART 口进行测试固件的下载。下载方式如下:

1. 设置芯片启动方式

将芯片 Boot 引脚跳到高电平, 从 UART/SDIO 启动, 按下复位按键后, 芯片即可从 UART/SDIO 启动。

2. 通过 TG Flash Environment 下载 MFG 固件

在 TG Flash Environment 界面, 通过 **View->MCU** 进入到 MCU 界面。

在左侧通信接口设置中:

- **Interface:** 用于选择下载的通信接口, 这里选择 `uart` 进行下载
- **COM Port:** 当选择 UART 进行下载的时候这里选择与芯片连接的 COM 口号, 可以点击 **Refresh** 按钮进行 COM 号的刷新

- **Uart Rate:** 当选择 UART 进行下载的时候，填写波特率，可以填写 500000
- **Xtal:** 用于选择板子所使用的晶振类型，对于评估板，这里选择 40M(SDIO/UART 下载无须关心)
- **Chip Erase:** 默认设置为 False，即下载时不擦除 Flash(SDIO/UART 下载无须关心)

其它项使用默认配置即可。

在右侧烧录镜像配置，分别选择：

- **Boot Source:** 选择 UART/SDIO
- **Image Address:** 0x22008000
- **Image File:** 在烧写工具目录下的对应芯片型号 builtin_imgs 目录下的 mfg 文件夹中，选择对应的 RAM 版本固件。本例中选择的是 chips/tg7100c/builtin_imgs/mfg/chip_mfg_ram_932b317e_40m.bin
- **Device Tree:** 非必须下载，客户可先参考《使用 DTS 文件设置 RF 相关参数》章节，了解其功能后进行选择是否下载。如需下载，在烧写工具目录下的对应芯片型号 device_tree 目录下，选择对应晶振的配置文件即可。本例中选择的是 chips/tg7100c/device_tree/chip_factory_params_ioTKitA_40M.dts

完成上述设定后，点击 Create&Download 按钮，完成固件程序的下载。下载成功的示意如下。

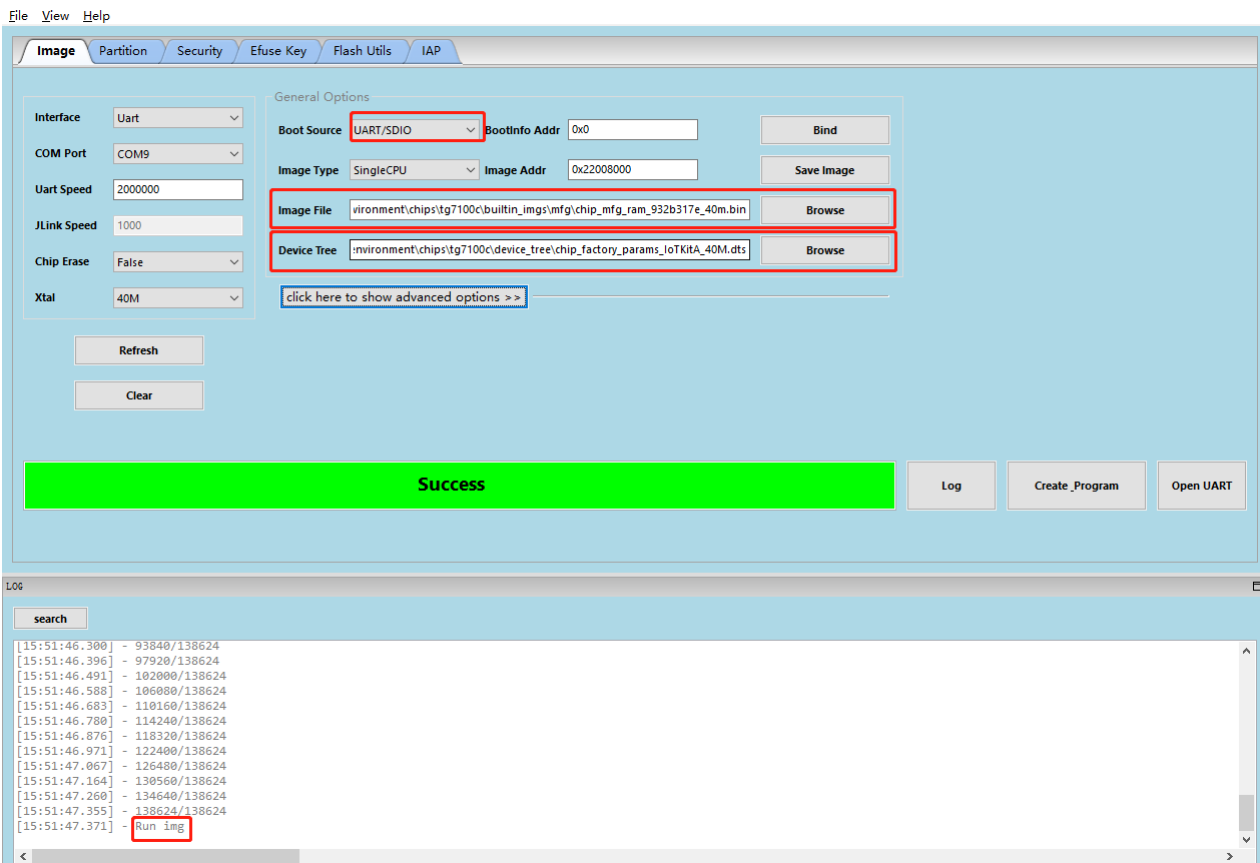


图 4.3: 烧写成功界面

4.2.2 模组不带 UART 接口

对于只有 SDIO 接口模组的 RF 性能测试评估，产测参数验证以及在不做产测情况下训练 RF 参数，我们提供了基于树莓派的下载平台。平台示意图如下：

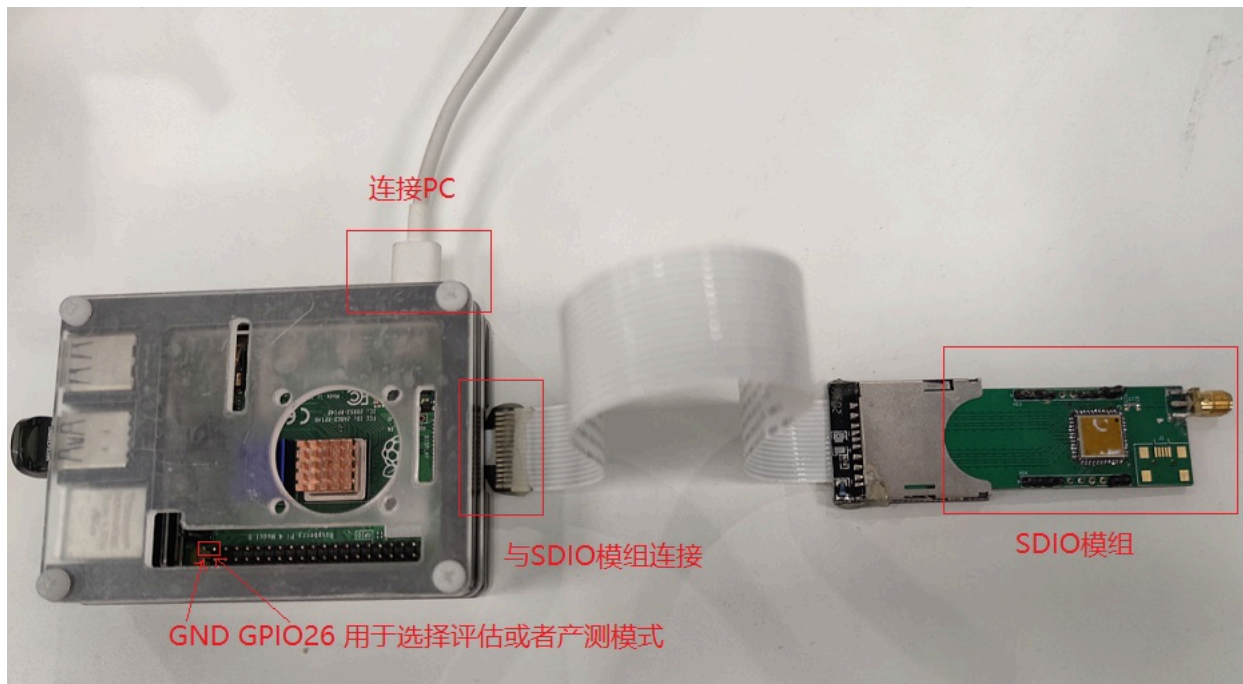


图 4.4: 下载平台示意图

下载步骤如下：

1. 设置下载模式

下载模式分为评估和产测两种模式。评估模式时，将树莓派中 GPIO26 引脚悬空。产测模式时，将 GPIO26 引脚跳到低电平。当前下载选择评估模式。

2. 将 SDIO 模组插入 SD card 接口

3. 树莓派通过 USB 口连接 PC，此时树莓派会将自己枚举成 USB 转串口（树莓派启动时间大约 30s）

4. 通过 TG Flash Environment 下载 MFG 固件

在 TG Flash Environment 界面，通过 View->MCU 进入到 MCU 界面。

在左侧通信接口设置中：

- **Interface:** 用于选择下载的通信接口，这里选择 Uart 进行下载
- **COM Port:** Dev Cube 会自动识别树莓派枚举出的串口，这里选择该串口，如果没有，可以点击 Refresh 按钮进行 COM 号的刷新
- **Uart Rate:** 当选择 UART 进行下载的时候，填写波特率，可以填写 2000000

- **Xtal:** 用于选择板子所使用的晶振类型，对于评估板，这里选择 40M(SDIO/UART 下载无须关心)
- **Chip Erase:** 默认设置为 False，即下载时不擦除 Flash(SDIO/UART 下载无须关心)

其它项使用默认配置即可。

在右侧烧录镜像配置，分别选择：

- **Boot Source:** 选择 UART/SDIO
- **Image Address:** 0x22008000
- **Image File:** 在烧写工具目录下的对应芯片型号 `builtin_imgs` 目录下的 `mfg` 文件夹中，选择对应的 RAM 版本固件。本例中选择的是 `chips/tg7100c/builtin_imgs/mfg/chip_mfg_ram_932b317e_40m.bin`
- **Device Tree:** 非必须下载，客户可先参考《使用 DTS 文件设置 RF 相关参数》章节，了解其功能后进行选择是否下载。如需下载，在烧写工具目录下的对应芯片型号 `device_tree` 目录下，选择对应晶振的配置文件即可。本例中选择的是 `chips/tg7100c/device_tree/chip_factory_params_loTKitA_40M.dts`

完成上述设定后，点击 **Create&Download** 按钮，完成固件程序的下载。

注解：如需将 SDIO 模组断电，直接插拔 SDIO 模组即可。如果插拔重启不成功，建议在 SDIO 模组插入的情况下将树莓派断电重启。

对于 IOT 模组 (带 Flash)，完成测试固件的下载以后，直接按下复位键 SW1，芯片就可以运行 RF 测试固件了，对于提供的模组评估套件，Boot 引脚会被串口程序自动控制，无需额外的设定。对于 SDIO 模组 (不带 Flash)，固件下载完毕后，无需任何设定，固件会自动运行起来。在 TGFlashEnv.exe 界面，通过 View->RF MFG 进入到 RF MFG 测试界面。选择使用到的 COM 号，点击 Open 按钮，即可看到固件程序成功运行的 log，示例如下。

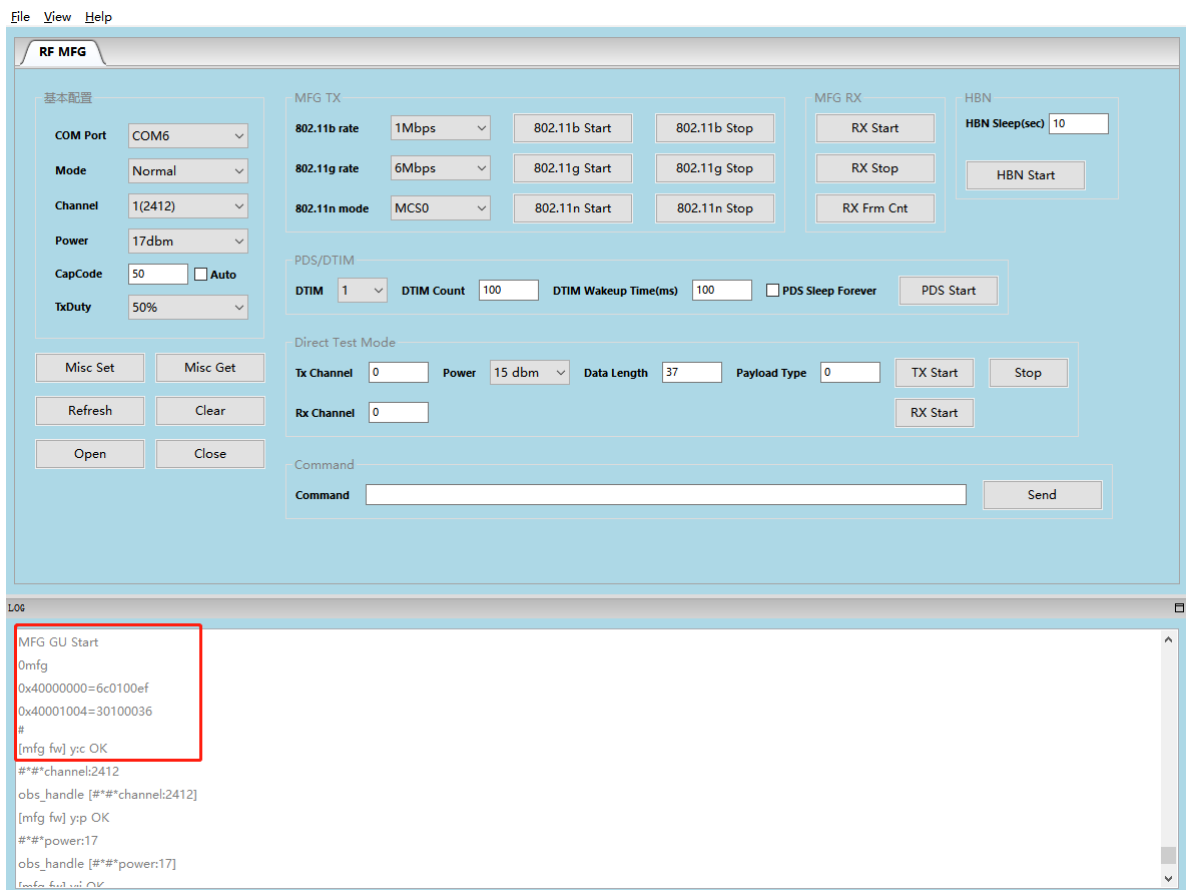


图 5.1: 程序成功运行的 log

上位机 UI 程序与测试固件通过 UART 通信，使用的波特率是 115200，数据位为 8 位，没有奇偶校验。

针对晶体的负载电容，TG7100C 系列芯片内部有电容补偿，不同的负载电容需求对应不同的电容补偿值，以下表格提供参考值。

备注：实际 PCB 走线也存在一定的寄生电容，所以最佳补偿值还是以实际测试结果为准。

表 6.1: TG7100C 对应的电容补偿值

XTAL Loading Capacity (pF)	Capacity Code
12	32~36
15	58~63

使用方法如下：

1. 在 Cap Code 中填写需要补偿的值。
2. 点击 Misc Set 按键更新补偿值。

基本配置

COM Port	COM6	▼
Mode	Normal	▼
Channel	1(2412)	▼
Power	17dbm	▼
CapCode	33	<input type="checkbox"/> Auto
TxDuty	50%	▼

Misc Set Misc Get

Refresh Clear

Open Close

图 6.1: 更新补偿值

7.1 Channel 和 Power 设置

通过 Channel 和 Power 下拉菜单框，可以设置数据包的发送通道和功率。Channel 可以选择 1-13，Power 可以选择 12-23dbm。

基本配置

COM Port	COM6
Mode	Normal
Channel	1(2412)
Power	17dbm
CapCode	33 <input type="checkbox"/> Auto
TxDuty	50%

Misc Set Misc Get

Refresh Clear

Open Close

图 7.1: 设置 Channel 和 Power 的参数

WiFi 不同的模式使用不同的调制方式，对信号质量 (EVM) 也有不同的要求，为了满足 WiFi 标准，针对不同制式推荐的最大功率如下表。

表 7.1: 最大功率表

Mode	Rate	Maximum Power(dBm)
11n	MCS7	17
	MCS6	18
	MCS5	18
	MCS4	18
	MCS3	18
	MCS2	18
	MCS1	18
	MCS0	18
11g	54Mbps	18
	48Mbps	19
	36Mbps	20
	24Mbps	20
	18Mbps	20
	12Mbps	20
	9Mbps	20
	6Mbps	20
11b	11Mbps	20
	5.5Mbps	20
	2Mbps	20
	1Mbps	20

TG7100C 系列芯片提供了功率校准机制，用户可在产品量产环节对各个 Channel 进行功率校准，将校准值写入芯片 Efuse，在应用程序启动后，根据写入的校准值纠正实际的 TX Power。

TG7100C, 在 efuse 容量上缩减了 50%，预留给功率补偿的 efuse bit 数目有大幅减少，因此 TG7100C 的功率补偿只能写入高中低 3 个信道的校准值，其余信道的补偿采用插值的方法。

7.2 发送数据包模式设置

7.2.1 11b 数据包发送

11b 数据包可以选择速率：1Mbps,2Mbps,5.5Mbps,11Mbps，前导默认选择 Long preamble。设置完毕后，就可以点击 802.11b Start 按钮进行发送，在发送期间，log 区域会打印已经发送数据包的个数。如果想要停止发送，点击 802.11b Stop 即可。

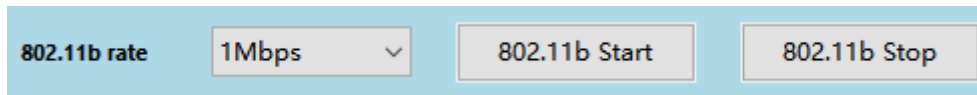


图 7.2: 11b 数据包设置速率

7.2.2 11g 数据包发送

11g 数据包可以选择速率: 6Mbps, 9Mbps, 12Mbps, 18Mbps, 24Mbps, 36Mbps, 48Mbps, 54Mbps, 设置完毕后, 就可以点击 802.11g Start 按钮进行发送, 在发送期间, log 区域会打印已经发送数据包的个数。如果想要停止发送, 点击 802.11g Stop 即可。

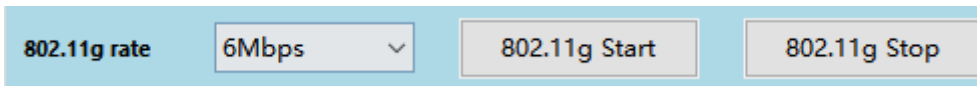


图 7.3: 11g 数据包设置速率

7.2.3 11n 数据包发送

11n 数据包可以选择速度模式 MCS0-MCS7, 默认带宽 20MHz, Long GI, 使用 HT-MF 模式。

注解: 目前 HT_GF 模式不支持。

设置完毕后, 就可以点击 802.11n Start 按钮进行发送, 在发送期间, log 区域会打印已经发送数据包的个数。如果想要停止发送, 点击 802.11n Stop 即可。

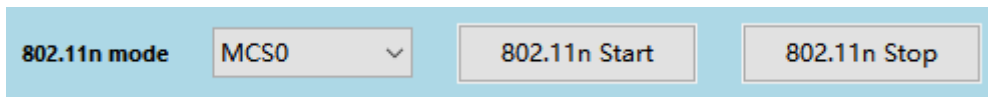


图 7.4: 11n 数据包发送界面

接收设置较为简单，点击 **RX Start** 按钮后即可进入数据包接收模式，点击 **RX Frm Cnt** 按钮可以显示数据包接收个数以及 RSSI 的平均值，效果如下图。

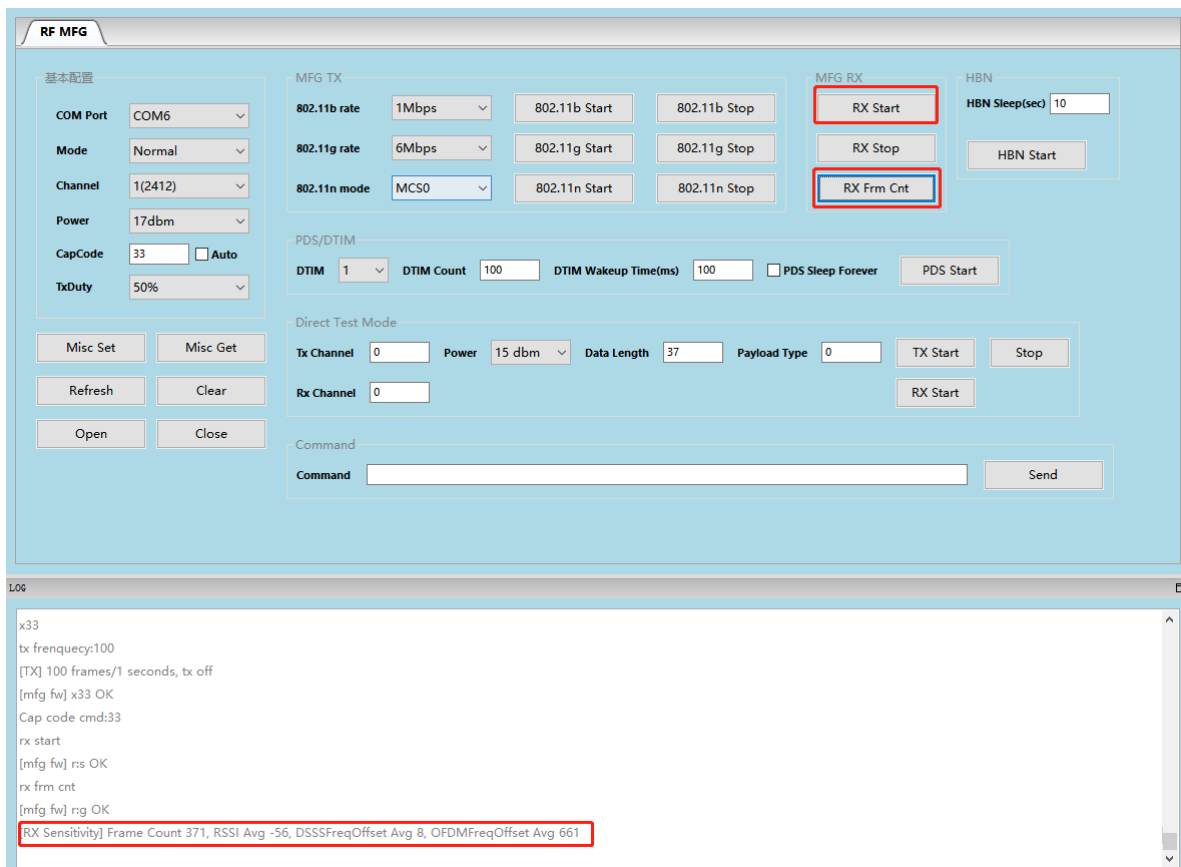


图 8.1: 接收数据包效果图

RF MFG 提供 BLE 的 TX 和 RX 测试。TX 测试可以设定测试的 Channel, Power, Data Length 和 Payload Type。设定完毕后点击 TX Start 按钮即可。RX 测试可以设定测试的 Channel, 然后点击 RX Start 按钮。测试可以使用 Stop 按钮停止。Payload Type 如下图所示。

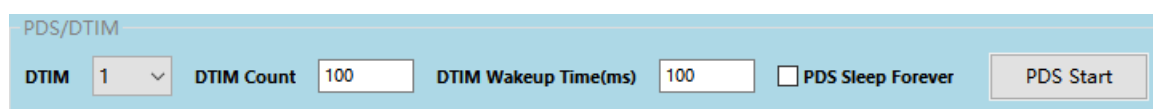
Value	Parameter Description
0x00	PRBS9 sequence '11111111100000111101...' (in transmission order) as described in [Vol 6] Part F, Section 4.1.5
0x01	Repeated '11110000' (in transmission order) sequence as described in [Vol 6] Part F, Section 4.1.5
0x02	Repeated '10101010' (in transmission order) sequence as described in [Vol 6] Part F, Section 4.1.5
0x03	PRBS15 sequence as described in [Vol 6] Part F, Section 4.1.5
0x04	Repeated '11111111' (in transmission order) sequence
0x05	Repeated '00000000' (in transmission order) sequence
0x06	Repeated '00001111' (in transmission order) sequence
0x07	Repeated '01010101' (in transmission order) sequence

图 9.1: BLE Payload Type

PDS 模式可设置芯片工作在较低功耗的同时可以唤醒 CPU 监听 WiFi AP 的 Beacon/DTIM 数据包，在检测到有数据需要接收时，启动 WiFi 的接收机进行数据的接收。PDS 模式可以设置的参数包括：

- DTIM, 设置间隔多少 Beacon 才含有 DTIM 信息，芯片会在该时刻唤醒，监听 Beacon
- DTIM Count, 设置接收多少个 DTIM（亦即睡眠次数）后，芯片进入正常的模式
- DTIM Wakeup Time, 设置芯片唤醒后，保持接收状态的时间，以便完整的接收到 Beacon

设置好上述三个参数后，点击 Start PDS 按钮即可设置芯片进入 PDS 模式。



PDS/DTIM

DTIM 1 DTIM Count 100 DTIM Wakeup Time(ms) 100 PDS Sleep Forever PDS Start

图 10.1: PDS 模式参数设置

HBN 测试可以让芯片进入 HBN 模式，在 HBN 模式下只有极少部分电路处在带电工作状态，其它电路的电源被关闭，功耗达到最低。芯片可以从 HBN 模式唤醒，唤醒后芯片会重新启动。芯片的 HBN 唤醒源支持 RTC 唤醒和 GPIO 唤醒，目前测试工具仅仅支持 RTC 定时唤醒，设置完 HBN 唤醒时间后，点击 Start HBN 按钮即可让芯片进入 HBN 模式，设定的时间到来后，芯片会重新启动。

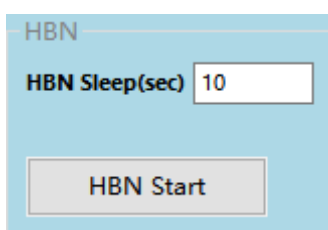


图 11.1: HBN 模式参数设置

MFG 支持 WiFi 和 BLE 的 CW(Continue Wave) 测试模式。

12.1 WiFi CW 测试

在界面 Mode 下拉菜单中选择 Test(CW) 模式，进入测试模式后，仅仅可以设定 Power 和 Channel，其它 WiFi 模式 (b/g/n) 相关的测试按钮是被禁止使用的。

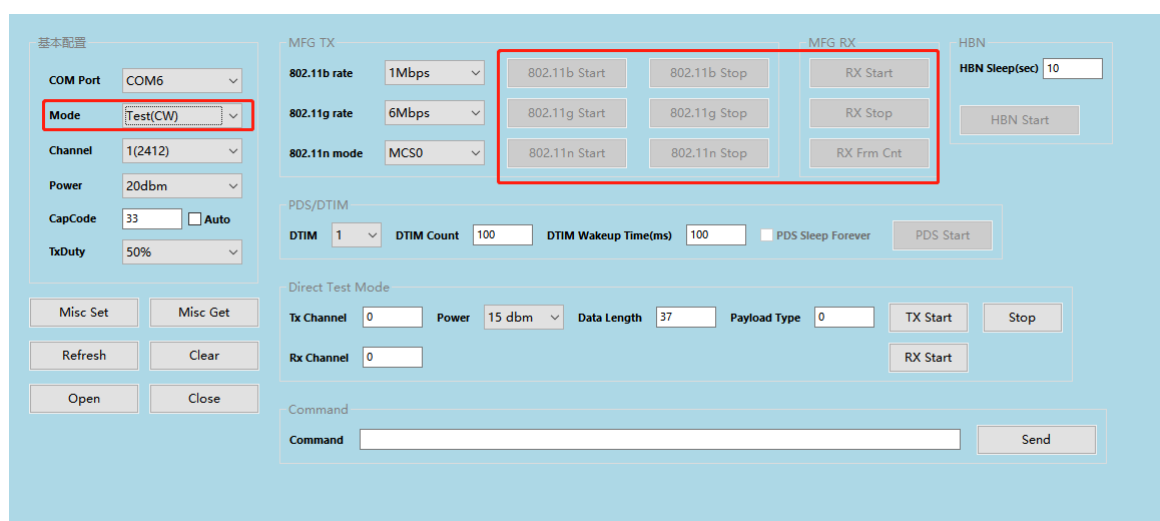


图 12.1: WiFi 测试模式参数设置

12.2 BLE CW 测试

在 RF MFG 界面的 Mode 下拉菜单中选择 Test(CW) 模式，选择需要的 Power 和 Tx Channel，点击 Tx Start 即可打开 Single Tone 波形输出。将 Mode 菜单切换到 Normal，停止 Single Tone 波形输出。在 Test(CW) 模式下如需切换 channel 和 power，不需要停止，重新设置 Tx PHY Channel 和 Power，点击 Tx Start 进行发送即可。

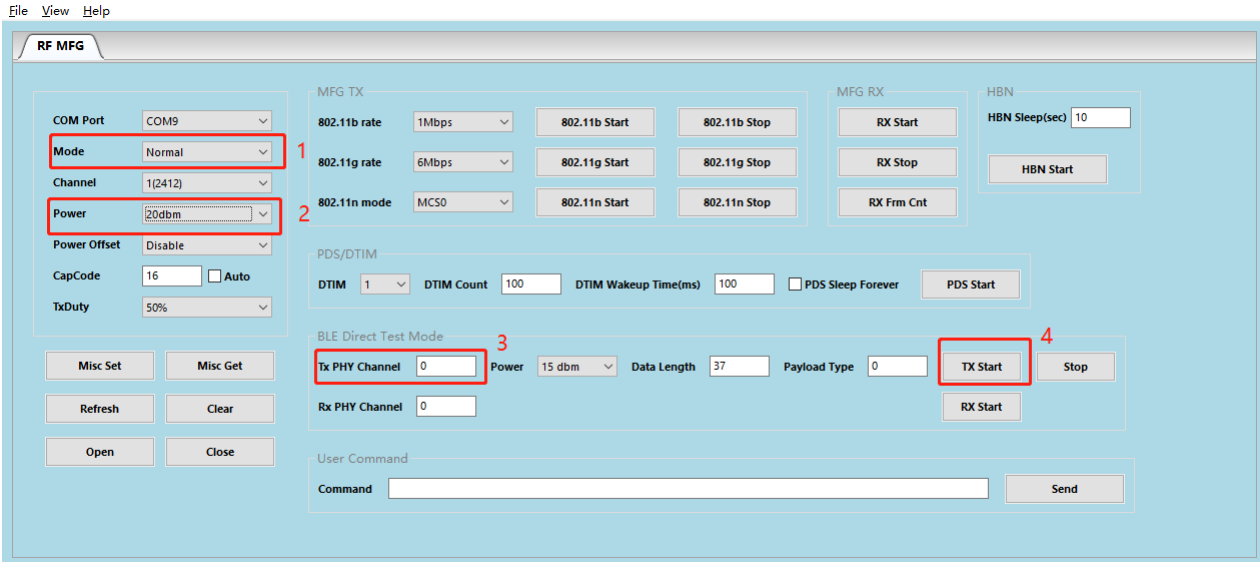


图 12.2: BLE 测试模式参数设置

DTS 文件是用于客户在不修改代码情况下，向固件传递参数的方法。

DTS 文件根据晶振大小选择,以 TG7100C 为例,DTS 文件路径为:chips/TG7100C/device_tree/chip_factory_params_loTKitA_40M.dts

DTS 文件中的配置项包括外设配置, WiFi/BLE 配置, RF 参数配置等, 本文主要介绍 RF 参数配置, Wi-Fi 固件正是通过 DTS 中这些 RF 参数, 完成射频的功率, 频偏等参数的设置。

RF 可配置的参数包括:

- 温补参数: 具体包括温补的使能和关闭, 温补参数的调整, 客户可以根据自身的补偿需求进行设定
- 功率参数: 可以设定 WiFi 在 b/g/n 三种模式以及对应速度下的发射功率, 同时也可以设定 BLE 的发送功率
- 功率校准模式及数值: 功率校准模式可以设定从 Efuse 加载功率校准数值或者从 DTS 文件中加载功率校准数值
- 频偏校准模式及数值: 频偏校准模式可以设定从 Efuse 加载频偏校准数值或者从 DTS 文件中加载频偏校准数值

设置频偏校准模式的参数为 `xtal_mode`, 其模式可以设置如下:

- `xtal_mode = "M"`: 只从 Efuse 中加载频偏校准数值 (Efuse 中数值一般是 RF 产测后写入的)
- `xtal_mode = "F"`: 只从 DTS 中加载频偏校准数值
- `xtal_mode = "MF"`: 优先从 Efuse 中加载频偏校准数值, 如果加载失败再从 DTS 中加载频偏校准数值
- `xtal_mode = "FM"`: 优先从 DTS 中加载频偏校准数值, 如果加载失败再从 Efuse 中加载频偏校准数值

设置功率校准模式的参数为 `pwr_mode`, 其模式可以设置如下:

- `pwr_mode = "B"`: 只从 Efuse 中加载功率校准数值 (Efuse 中数值一般是 RF 产测后写入的)
- `pwr_mode = "F"`: 只从 DTS 中加载功率校准数值
- `pwr_mode = "BF"`: 优先从 Efuse 中加载功率校准数值, 如果加载失败再从 DTS 中加载功率校准数值
- `pwr_mode = "FB"`: 优先从 DTS 中加载功率校准数值, 如果加载失败再从 Efuse 中加载功率校准数值

具体参数使用方法和注意事项可参考 DTS 文件中各个参数的注释。

TG7100C 系列芯片，支持发送功率温度补偿，具体介绍如下。

14.1 设置功率温度补偿

14.1.1 概述

TG7100C 采取低于室温和高于室温两段,并随信道变化的温补机制,在-40°C 到 125°C 环境温度下控制功率变化 $\leq\pm 1.5\text{dB}$ 。

14.1.2 DTS 文件中温补参数说明

```
rf_temp {
    en_tcal = <0>;
    linear_or_follow = <1>;
    Tchannels = <2412 2427 2442 2457 2472>;
    Tchannel_os = <180 168 163 160 157>;
    Tchannel_os_low = <199 186 170 165 160>;
    Troom_os = <255>;
    //negative value is NOT supported. So we use '256' for 0, '255' for -1, '257' for 1, '511' for 256
};
```

图 14.1: 温补参数说明

- en_tcal: 1= 打开温补机制, 0= 关闭温补机制 (默认)。
- Tchannels: 温补参数对应的信道。
- Tchannel_os: 环境温度 \geq 室温, 每 0.5dB 功率变化对应的环境温度变化 * 10。
- Tchannel_os_low: 环境温度 $<$ 室温, 每 0.5dB 功率变化对应的环境温度变化 * 10。
- linear_or_follow: 其它信道补偿速度, 1= 信道间插值 (默认), 0= 前一个信道 in Tchannels。
- Troom_os: 室温下, 内部温度 sensor 偏移值, 默认是 0 (256)。

14.1.3 测试获得温补参数

Step1: 将 TG7100C 置于温箱中，关闭温补机制 (en_tcal=0)，使用 MFG 进行 Wi-Fi Tx 测试，选取 50% Duty Cycle，以目标功率持续 Tx。

Step2: 环境温度 TMid=20°C，稳定 5 分钟后，记录 Tchannels 里每个信道上的 Tx 功率大小 Pwr_Mid[Channel]。

Step3: 环境温度 TLow=-40°C，稳定 5 分钟后，记录 Tchannels 里每个信道上的 Tx 功率大小 Pwr_Low[Channel]。

Step4: 环境温度 THigh=120°C，稳定 5 分钟后，记录 Tchannels 里每个信道上的 Tx 功率大小 Pwr_High[Channel]。

Step5: 计算各信道补偿速度， $Tchannel_os_low[Channel] = (T_{Mid} - T_{Low}) / |Pwr_Low[Channel] - Pwr_Mid[Channel]| / 2 * 10$

$Tchannel_os[Channel] = (T_{High} - T_{Mid}) / |Pwr_Mid[Channel] - Pwr_High[Channel]| / 2 * 10$

Step6: 增加 TG7100C 样本 (建议样本总数 ≥3)，重复 Step1~5，对 Tchannel_os_low[Channel]、Tchannel_os[Channel] 取平均值。

Step7: 更新 DTS 温补参数 Tchannel_os 和 Tchannel_os_low，打开温补机制 (en_tcal=1)，在温箱中验证功率变化。

14.1.4 示例

Sample	Channel (Mhz)	Power (dbm)	Thermostat temperature (°C)	Actual power (dbm)	20° to -40° power change (db)	20° to 120° power change (db)	20° to -40° 0.5dB power change temperature interval (°C)	20° to 120° 0.5dB power change temperature interval (°C)	Recommended compensation interval from 20° to -40°	Recommended compensation interval from 20° to 120°
1	2412	16	-40	16.5	1.5	-2.9	20	17.24137931	200	172
			20	15						
			120	12.1						
	2427	16	-40	16.9	1.5	-3.1	20	16.12903226	193	164
			20	15.4						
			120	12.3						
	2442	16	-40	17.3	1.6	-3.2	18.75	15.625	167	156
			20	15.7						
			120	12.5						
	2457	16	-40	17.8	2	-3.7	15	13.51351351	161	137
			20	15.8						
			120	12.1						
	2472	16	-40	18.2	2.2	-4.2	13.63636364	11.9047619	136	119
			20	16						
			120	11.8						

图 14.2: 示例

对于做 RF 产测的客户，如果需要在非信令模式下验证产测数值是否符合预期，可以使用 MFG 固件和对应的评估软件实现，具体介绍如下。

15.1 验证频偏校准数值

1. 准备一块做完产测的模组。
2. 修改 DTS 文件中的 `xtal_mode` 为“M”(如果确认 Efuse 中有频偏校准数值，模式设置为“MF”亦可), 即只从 Efuse 中加载频偏校准数值。
3. 按照上述烧写或者下载固件的步骤，完成 MFG 固件的烧写或者下载。
4. MFG 固件运行后，通过 View->RF MFG 进入到 RF MFG 测试界面，将基本配置项中 Cap Code 的 Auto 选项勾选，此时 MFG 固件就会按照 `xtal_mode` 设定的模式，从 Efuse 中加载频偏校准数值。
5. 设定发送数据包，使用仪器测量实际输出的频率。

勾选 Cap Code 的 Auto 选项以后，MFG 固件输出相关 log 信息如下：

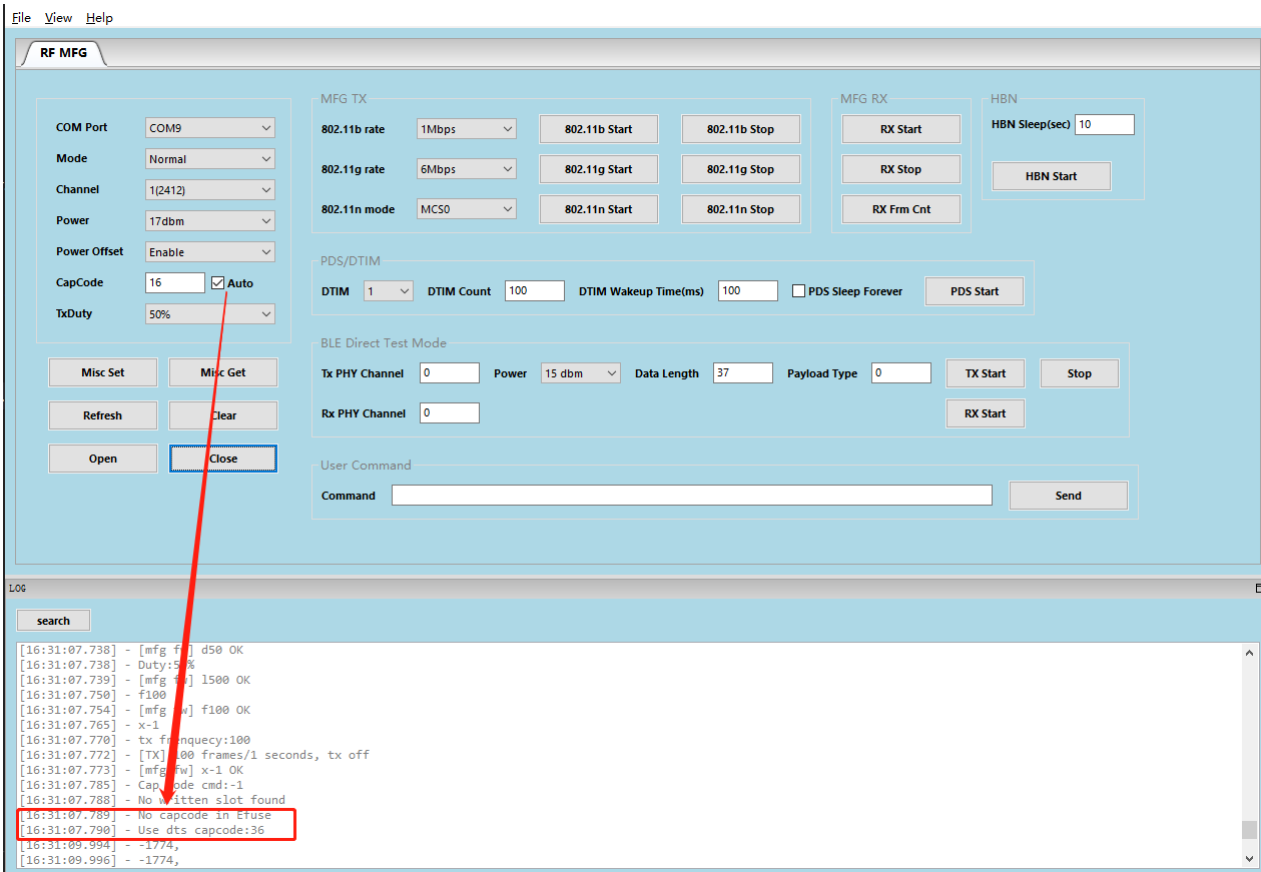


图 15.1: 勾选 Auto 后的 log 信息

注解：此时上位机软件发送给固件的命令是 X-1[r\n], 表示固件按照 DTS 文件中的 xtal_mode 加载频偏数值，如果不勾选 Auto，则上位机发送给固件的命令是 X+ 用户填写的数值。

15.2 验证功率校准数值

1. 准备一块做完产测的模组。
2. 修改 DTS 文件中的 pwr_mode 为“B”(如果确认 Efuse 中有功率校准数值，模式设置为“BF”亦可), 即只从 Efuse 中加载功率校准数值。
3. 按照上述烧写或者下载固件的步骤，完成 MFG 固件的烧写或者下载。
4. MFG 固件运行后，通过 View->RF MFG 进入到 RF MFG 测试界面，将基本配置项中的 Power Offset 设置为 Enable，此时 MFG 固件就会按照 pwr_mode 设定的模式，从 Efuse 中加载功率校准数值。
5. 设定发送功率，使用仪器测量实际输出的功率。

使能 Power Offset 以后，MFG 固件输出相关 log 信息如下：

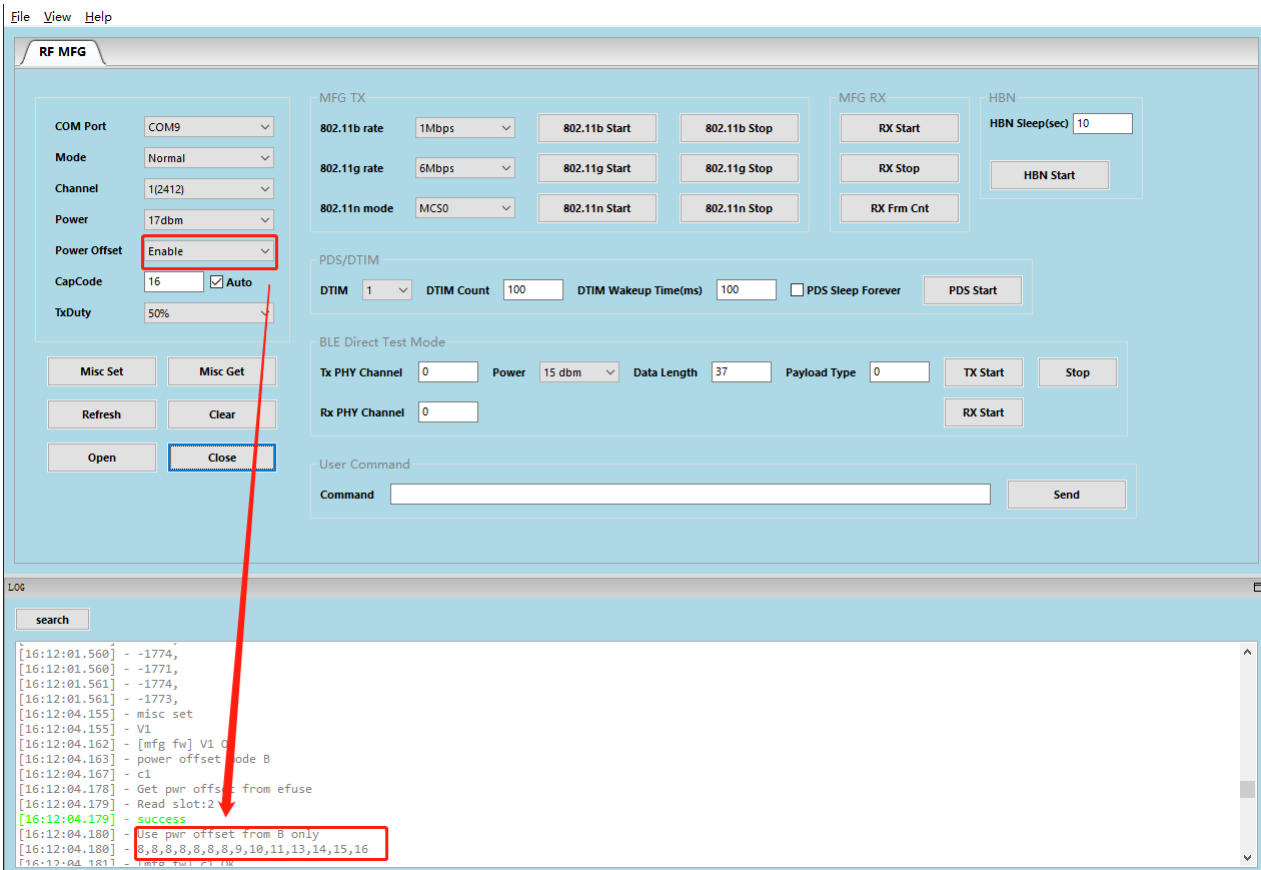


图 15.2: 使能 Power Offset 的 log 信息

注解：使能 Power Offset 功率补偿的命令是 V1[\r\n] 或者 V-1[\r\n]，MFG 固件收到这个命令后会按照 DTS 中的 pwr_mode 从 Efuse 中加载产测功率校准数值。如果 Power Offset 选择 Disable，对应的命令是 V0[\r\n]，此时固件在设置功率的时候，不会加上 Efuse 或者 DTS 中的功率补偿校准数值。

所有命令均是字符串类型。

16.1 Shakehand

- 命令: H
- 返回: mfg

主机工具应该先发送“H\r\n”去检测 MFG 固件是否已经运行起来，如果 MFG 固件已经运行起来，它收到“H\r\n”命令后会以“mfg\r\n”应答。如果 MFG 没有在运行：

1. 主机用正常固件使用的波特率（默认 9600）发送“mfg\r\n”命令，让正常固件切换到 MFG 固件。
2. 主机使用 115200 的波特率发送“H\r\n”并检查能否收到“mfg\r\n”。
3. 如果主机收不到“mfg\r\n”，重复步骤 1。
4. 主机收到“mfg\r\n”后可以正常的测试。

16.2 TX on/off

on:t1

off:t0

16.3 TX modulation

16.3.1 2.4G 11n

mcs idx = 0 - 7

1. short GI + HT-GF + HT20:msg2[mcs idx]
2. short GI + HT-MF + HT20:msm2[mcs idx]

3. long GI + HT-GF + HT20:mlg2[mcs idx]
4. long GI + HT-MF + HT20:mlm2[mcs idx]
5. short GI + HT-GF + HT40:msg4[mcs idx]
6. short GI + HT-MF + HT40:msm4[mcs idx]
7. long GI + HT-GF + HT40:mlg4[mcs idx]
8. long GI + HT-MF + HT40:mlm4[mcs idx]

注解：TG7100C 不支持 HT40。

16.3.2 2.4G 11g

rate idx = 0 - 7, 0:6Mbps 1:9Mbps 2:12Mbps 3:18Mbps 4:24Mbps 5:36Mbps 6:48Mbps 7:54Mbps

- 命令：g[rate idx]

16.3.3 2.4G 11b

rate idx = 0 - 3, 0:1Mbps 1:2Mbps 2:5.5Mbps 3:11Mbps

1. Long Preamble:B[rate idx]
2. short Preamble:b[rate idx]

16.4 2.4g channel

channel idx = 1 - 13

- 命令：c[channel idx]

16.5 2.4g tx power

power dbm = 12 - 23dbm

- 命令：p[power dbm]

注解：p-1 表示从 DTS 文件中加载功率参数，并按照 DTS 中定义规则，设置功率，详见《使用 DTS 文件设置 RF 相关参数》章节。

16.6 TX frame length

- 命令: l[length]

16.7 TX frequency

max value=1000

- 命令: f[freq]

16.8 PDS

enter into pds mode

1. sleep forever sa
2. rtc wakeup mode and dtim mode dtim:1 - 9 dtim count s: [dtim] [dtim count]
3. wakeup keep time keep ms: Unit is microsecond a:w[keep ms]

16.9 HBN

enter into hbn mode

1. rtc wake up mode hr [second]
2. gpio wake up mode TODO

16.10 RX

1. start rx r:s
2. get rx information r:g

- 返回:[RX Sensitivity] Frame Count [frame count], RSSI Avg [anverage of RSSI], DSSSFreqOffset Avg [anverage of DSSS Frequency Offset], OFDMFreqOffset Avg [anverage of OFDM Frequency Offset]

16.11 Get MFG FW version

- 命令: y:v
- 返回: ###*version:[version]

16.12 Get MFG FW building information

- 命令: y:d
- 返回: ###date:[building date] time:[building time]

16.13 Get current power level

- 命令: y:p
- 返回: ###power:[power level dbm]

16.14 Get current channel

- 命令: y:c
- 返回: ###channel:[channel freq]

16.15 Get current tx status

- 命令: y:t
- 返回: ###tx:[0 or 1]

16.16 Get tx frequency

- 命令: y:f
- 返回: ###freq:[tx frequency]

16.17 Get cap code

- 命令: y:x
- 返回: ###capcode:[capcode value]

16.18 Get MFG mode

- 命令: y:M
- 返回: ###mfgmode:[MFG mode]

16.19 Set cap code

- 命令: X[cap code]

注解: X-1 表示从 DTS 文件中加载功率参数, 并按照 DTS 中定义规则, 设置频偏, 详见《使用 DTS 文件设置 RF 相关参数》章节。

16.20 Set MFG Test(CW) mode

0 for normal mode, 1 for CW test mode

- 命令: M[MFG mode]

16.21 Write data to efuse

注意, 由于 Efuse 具有写入后不可修改的特点, 所以在对 efuse 进行读写的时候, 要确保芯片正确的收到了主机发出的数据, 为此, 主机要按照如下流程进行设定:

1. 主机使用 WEA 命令将要写入的数据发给 MFG 的 FW, 此时 FW 只是将数据暂存, 并没有写入 Efuse。
2. 主机使用 LEA 命令从 Efuse 暂存区读取设定的参数, 判断 FW 是否正确接收, 如果没有正确接收, 重复步骤 1。
3. 主机判断设定的参数正确后, 使用 SEA 命令, 将参数真正的写入 Efuse。
4. 主机使用 REA 命令, 从 Efuse 中读取设定的参数, 校验正确则可认为 Efuse 写入成功。

16.21.1 Write data to efuse buffer

- 命令: WEA[address in hex string]=[value in hex string]

示例:

写入向 0x04 地址写 0x80000008

```
WEA0x00000004=0x80000008
```

16.21.2 Load data from efuse buffer

- 命令: LEA[address in hex string]
- 返回: Read efuse [address in hex string]=[value in hex string]

示例:

读取 0x04 地址处的数据

```
LEA0x00000004
```

返回

Read efuse 0x00000004=0x80000008

16.21.3 Program data to efuse

- 命令: SEA
- 返回: Save efuse OK

16.21.4 Read data from efuse

- 命令: REA[address in hex string]
- 返回: Read efuse [address in hex string]=[value in hex string]

示例:

读取 0x04 地址处的数据

LEA0x00000004

返回

Read efuse 0x00000004=0x80000008

16.22 Save calibration parameters to efuse

注意, 由于 Efuse 具有写入后不可修改的特点, 所以在使用 Efuse 进行参数设定的时候, 要确保芯片正确的收到了主机发出的参数, 为此, 主机要按照如下流程进行设定:

1. 主机使用 WEx 命令将要写入的数据发给 MFG 的 FW, 此时 FW 只是将数据暂存, 并没有写入 Efuse。
2. 主机使用 LEx 命令从 Efuse 暂存区读取设定的参数, 判断 FW 是否正确接收, 如果没有正确接收, 重复步骤 1。
3. 主机判断设定的参数正确后, 使用 SEx 命令, 将参数真正的写入 Efuse。
4. 主机使用 REx 命令, 从 Efuse 中读取设定的参数, 校验正确则可认为 Efuse 写入成功。

16.22.1 Write cap code to efuse buffer

- 命令: WEX[cap code]

16.22.2 Load cap code from efuse buffer

- 命令: LEX
- 返回: Cap code2:[cap code]

16.22.3 Program cap code to efuse

- 命令: SEX

16.22.4 Read cap code from efuse

- 命令: REX
- 返回: Cap code2:[cap code]

16.22.5 Write power offset to efuse buffer

- 命令: WEP[Channel 1 power offset],[Channel 2 power offset]...[Channel 12 power offset],[Channel 14 power offset]

示例:

写入 Channel 1-14 的功率偏移-1,2,3,3,3,2,1,0,-1,-2,-3,-4,1,3

WEP-1,2,3,3,3,2,1,0,-1,-2,-3,-4,1,3

注解: 如果功率校准采用的是线性插值方法, 比如只做 1,7,13 通道的校准, 但是使用 WEP 命令的时候仍然需要传递 14 个通道的数值, 其它通道的偏移值可以写 0. 同样的道理, 如果只做某两个通道的校准, 也是需要传递 14 个通道的数值, 不关心的通道功率偏移值可以设置为 0.

16.22.6 Load power offset from efuse buffer

- 命令: LEP
- 返回:Power offset:[Channel 1 power offset],[Channel 2 power offset]...[Channel 12 power offset],[Channel 14 power offset]

16.22.7 Program power offset to efuse

- 命令: SEP

16.22.8 Read power offset from efuse

- 命令: REP
- 返回:Power offset:[Channel 1 power offset],[Channel 2 power offset]...[Channel 12 power offset],[Channel 14 power offset]

16.22.9 Enable power offset in efuse

- 命令: V

MFG 固件默认不会使能 TX Power Offset 的校准功能, 如果需要验证校准的准确性, 需要发送 V 命令使能 TX Power Offset 校准功能, 收到该命令后 MFG 固件启用校准功能, 并会通过 log 打印使用的 efuse 校准值。

16.22.10 Write mac address to efuse buffer

- 命令:WEM[MAC0 hex string]:[MAC1 hex string]:[MAC2 hex string]:[MAC3 hex string]:[MAC4 hex string]:[MAC5 hex string]

示例:

写入 MAC 地址: 18:B9:05:60:0E:74,

```
WEM18:B9:05:60:0E:74
```

16.22.11 Load mac address from efuse buffer

- 命令: LEM
- 返回:MAC:[MAC0 hex string]:[MAC1 hex string]:[MAC2 hex string]:[MAC3 hex string]:[MAC4 hex string]:[MAC5 hex string]

示例:

```
返回 MAC:18:B9:05:60:0E:74
```

16.22.12 Program mac address to efuse

- 命令: SEM

16.22.13 Read mac address from efuse buffer

- 命令: REM
- 返回:MAC:[MAC0 hex string]:[MAC1 hex string]:[MAC2 hex string]:[MAC3 hex string]:[MAC4 hex string]:[MAC5 hex string]

示例:

返回 MAC:18:B9:05:60:0E:74

16.23 Save calibration parameters to flash

由于 efuse 具有写入后无法改写的特点, 故产测校准参数写入到 efuse 是有次数限制的, 不同系列芯片, 最大写入次数不同。为了提供更高的灵活性, 产测固件支持将产测的频偏参数, 功率校准参数, 用户 MAC 地址等参数, 写入到 Flash。如需要将产测参数写入到 Flash, 用户需要在烧写产测固件时, 在分区表中增加“rf_para”这个分区, 产测软件启动后, 如果在分区表中找到了“rf_para”分区, 就会把 RF 的相关参数写在 flash 中, 而不会写在 efuse 中。校准参数的写入命令和流程同《Save calibration parameters to efuse》

flash 中 RF 参数结构体如下:

```
typedef struct rf_para_flash_tag{
    uint32_t magic;           //"RFPA"
    uint8_t capcode_valid;    //0x5A
    uint8_t capcode;
    uint8_t poweroffset_valid; //0x5A
    int8_t poweroffset[3];
    uint8_t mac_valid;       //0x5A
    uint8_t mac[6];
    uint8_t rsvd[3];
    uint32_t crc32;
}rf_para_flash_t;
```

16.24 Reset MFG FW

- 命令: Reset

注解: It's only design for MFG firmware running from flash.

16.25 Set tx duty

- 命令: d[duty]

注解: Duty value is between 0-100

16.26 Get tx duty

- 命令: y:i
- 返回: ###duty:[tx duty]

16.27 BLE Test

16.27.1 BLE TX Power

- 命令: EP[power]

Power 参数是 16 进制字符串。

举例: EP11

Set Tx Power 17 dbm.

16.27.2 BLE TX

- 命令: ET[channel][data length][payload type]

所有的参数都是 16 进制字符串。

举例: ET261600

To transmit le test data on RF Channel38, with data Length 22 and with PRBS9 packet payload type and data len 22.

16.27.3 BLE RX

- 命令: ER[channel]

channel 参数是 16 进制字符串。

举例: ER26

To receive le test data on RF Channel38.

16.27.4 BLE test stop

- 命令: EE