

**TG7100C 量产指导**

**RF 产测篇**

版本: 1.1

版权 @ 2023

1 版本记录 . . . . .	4
2 概述 . . . . .	5
3 DTS 文件介绍 . . . . .	6
4 RF 产测方案 1--训练频偏和功率补偿经验值 . . . . .	8
4.1 训练频偏和功率补偿经验值 . . . . .	8
4.2 验证频偏和功率补偿经验值 . . . . .	11
5 RF 产测方案 2--RF 产测校准方案 . . . . .	14
5.1 频偏校准的流程 . . . . .	15
5.2 功率校准的流程 . . . . .	16
5.3 验证频偏和功率校准结果 . . . . .	17
5.4 方案 1 与方案 2 的对比 . . . . .	19
6 MAC 地址写入 . . . . .	20
7 RF 参数存储方案介绍 . . . . .	21
8 RF 测试命令 . . . . .	22
8.1 Shakehand . . . . .	23
8.2 TX on/off . . . . .	23
8.3 TX modulation . . . . .	23
8.3.1 2.4G 11n . . . . .	23
8.3.2 2.4G 11g . . . . .	24
8.3.3 2.4G 11b . . . . .	24
8.4 2.4g channel . . . . .	24
8.5 2.4g tx power . . . . .	24
8.6 TX frame length . . . . .	25
8.7 TX frequency . . . . .	25
8.8 PDS . . . . .	25
8.9 HBN . . . . .	25

8.10	RX . . . . .	25
8.11	Get MFG FW version . . . . .	25
8.12	Get MFG FW building infomation . . . . .	26
8.13	Get current power level . . . . .	26
8.14	Get current channel . . . . .	26
8.15	Get current tx status . . . . .	26
8.16	Get tx frequency . . . . .	26
8.17	Get cap code . . . . .	26
8.18	Get MFG mode . . . . .	26
8.19	Set cap code . . . . .	27
8.20	Set MFG Test(CW) mode . . . . .	27
8.21	Write data to efuse . . . . .	27
8.21.1	Write data to efuse buffer . . . . .	27
8.21.2	Load data from efuse buffer . . . . .	27
8.21.3	Program data to efuse . . . . .	28
8.21.4	Read data from efuse . . . . .	28
8.22	Save calibration parameters to efuse . . . . .	28
8.22.1	Write cap code to efuse buffer . . . . .	28
8.22.2	Load cap code from efuse buffer . . . . .	29
8.22.3	Program cap code to efuse . . . . .	29
8.22.4	Read cap code from efuse . . . . .	29
8.22.5	Write power offset to efuse buffer . . . . .	29
8.22.6	Load power offset from efuse buffer . . . . .	29
8.22.7	Program power offset to efuse . . . . .	30
8.22.8	Read power offset from efuse . . . . .	30
8.22.9	Enable power offset in efuse . . . . .	30
8.22.10	Write mac address to efuse buffer . . . . .	30
8.22.11	Load mac address from efuse buffer . . . . .	30
8.22.12	Program mac address to efuse . . . . .	30
8.22.13	Read mac address from efuse buffer . . . . .	31
8.23	Save calibration parameters to flash . . . . .	31
8.24	Reset MFG FW . . . . .	31
8.25	Set tx duty . . . . .	32
8.26	Get tx duty . . . . .	32
8.27	BLE Test . . . . .	32
8.27.1	BLE TX Power . . . . .	32
8.27.2	BLE TX . . . . .	32
8.27.3	BLE RX . . . . .	32
8.27.4	BLE test stop . . . . .	33

表 1.1: 修改记录

版本	更新内容
V1.0	initial version
V1.1	Update xtal mode and pwr mode when training experience value

本文仅适用于 BL602/BL604 系列芯片的产测指导。主要包括:

- DTS 文件介绍
- 产测方案与产测流程说明
- MAC 地址写入
- RF 参数存储方案介绍
- RF 测试命令

常见的 RF 产测方案有两种，一种是通过经验值设定频偏校准参数和功率校准参数，一种是通过产线 RF 射频测试仪器，精确校准频偏和功率。本文将对两种方案分别介绍。

通过经验值设定频偏校准参数和功率校准参数是存储在 DTS 文件中的，此外对于射频模块在各个制式下的发射功率也是通过 DTS 文件设定的，功率和频偏校准值的加载策略等也是通过 DTS 文件设定的，所以在介绍具体产测方案之前，先介绍些 DTS 文件。

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\_XXX.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。

## RF 产测方案 1--训练频偏和功率补偿经验值

### 4.1 训练频偏和功率补偿经验值

对于不需要做 RF 频偏和功率校准的用户，MFG 提供了设置经验值的方法。用户可抽检一批产品，首先根据实际需求调整 dts 中的 `pwr_table_11b` / `pwr_table_11g` / `pwr_table_11n` / `pwr_table_ble` 四个表项的功率数值，此时由于板材器件等等诸多因素，实际发射功率和设定的功率数值存在偏差，用户可以通过调整 `pwr_offset`，对实际发射的功率进行补偿调整。最终 `pwr_offset` 数值，可以通过测量一批电路板的平均值进行确定。频偏参数同样可以通过调整 `xtal` 来实现。训练功率和频偏经验值的测试环境示意图如下所示。

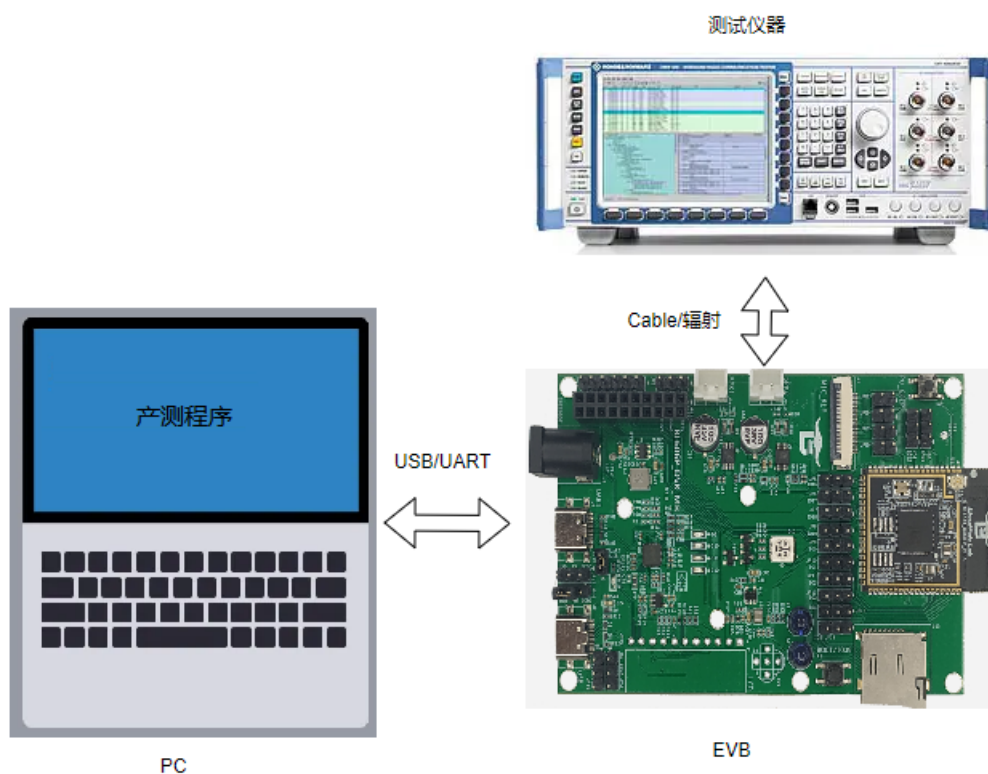


图 4.1: 训练功率和频偏经验值



训练频偏经验值方法如下：

1. 按示意图连接好开发板、测试 PC 和射频测试仪器；
2. 完成自启动 MFG 固件烧写或者下载；
3. 通过 DevCube 的 View->RF MFG 进入到 RF MFG 测试界面，选择对应的串口，点击 Open，复位模组，可以在 Log 窗口中看到 MFG 固件启动 Log；
4. 将基本配置项中 Power 选项设置为用户期望的目标功率 Pset（比如 17dBm），设置待测试信道（比如信道 1），CapCode 的 Auto 选项不能勾选，默认值配置为 32，设定发送数据包（推荐 11n MCS7）；
5. 使用射频仪器测量实际输出信号频率；
6. 若频偏值在目标范围内，则训练结束，此时的 capcode 值则为训练结果；
7. 如频偏值超出目标范围，则调整 capcode 值，并点击 misc set 按键，重复步骤 4 和 6；
8. 将步骤 6 中得到的训练结果记录下来，将所有待测板测试完毕后，求取平均值，并将均值写入 DTS 的 xtal 数组的第 1 项和第 2 项，例如训练结果为 31，则写入的值为:xtal = <31 31 0 60 60>;

使用 DevCube 烧写自启动的 MFG 固件示意如下 (因为是做 RF 参数训练，所以无需烧写应用固件)：

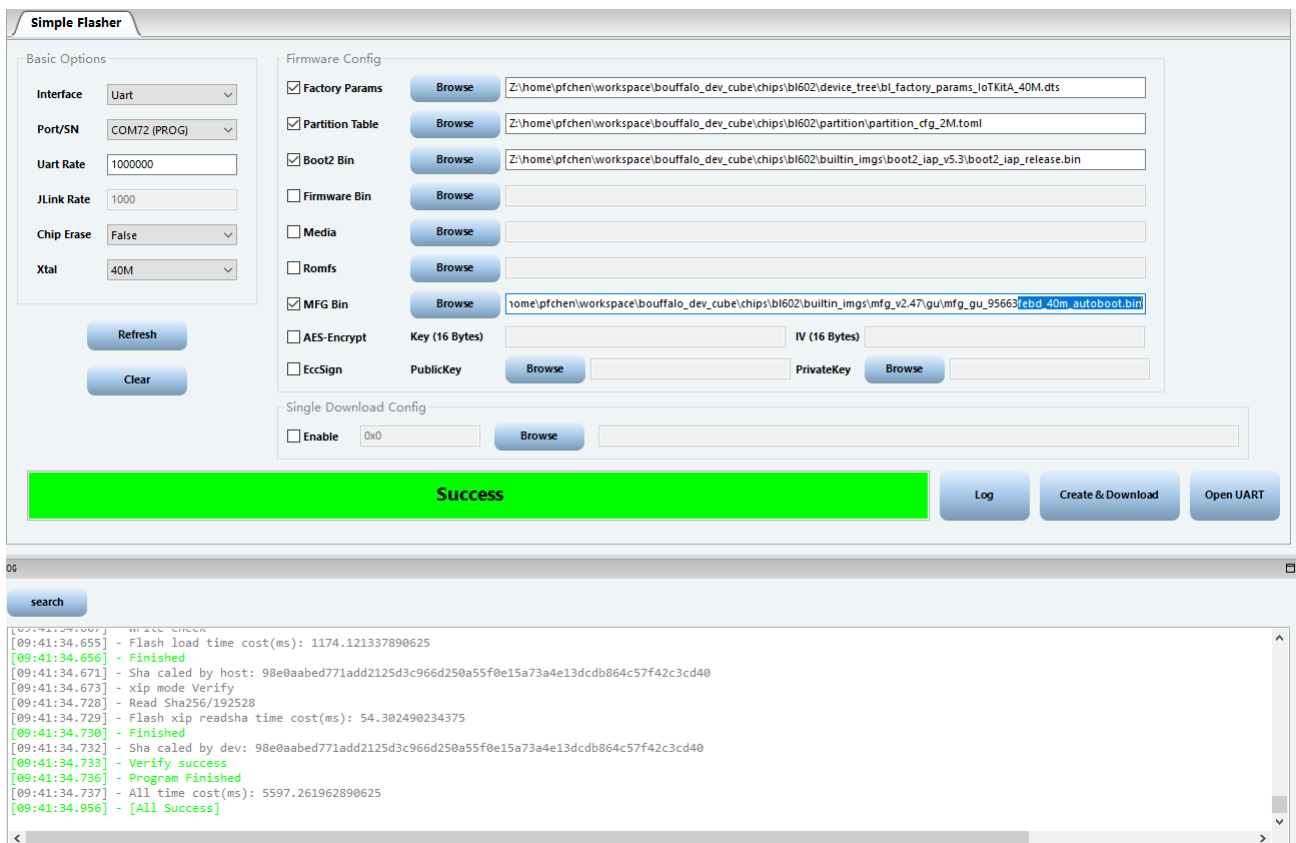


图 4.2: 烧写自启动的 MFG 固件

RF MFG 界面主要配置示意图下：

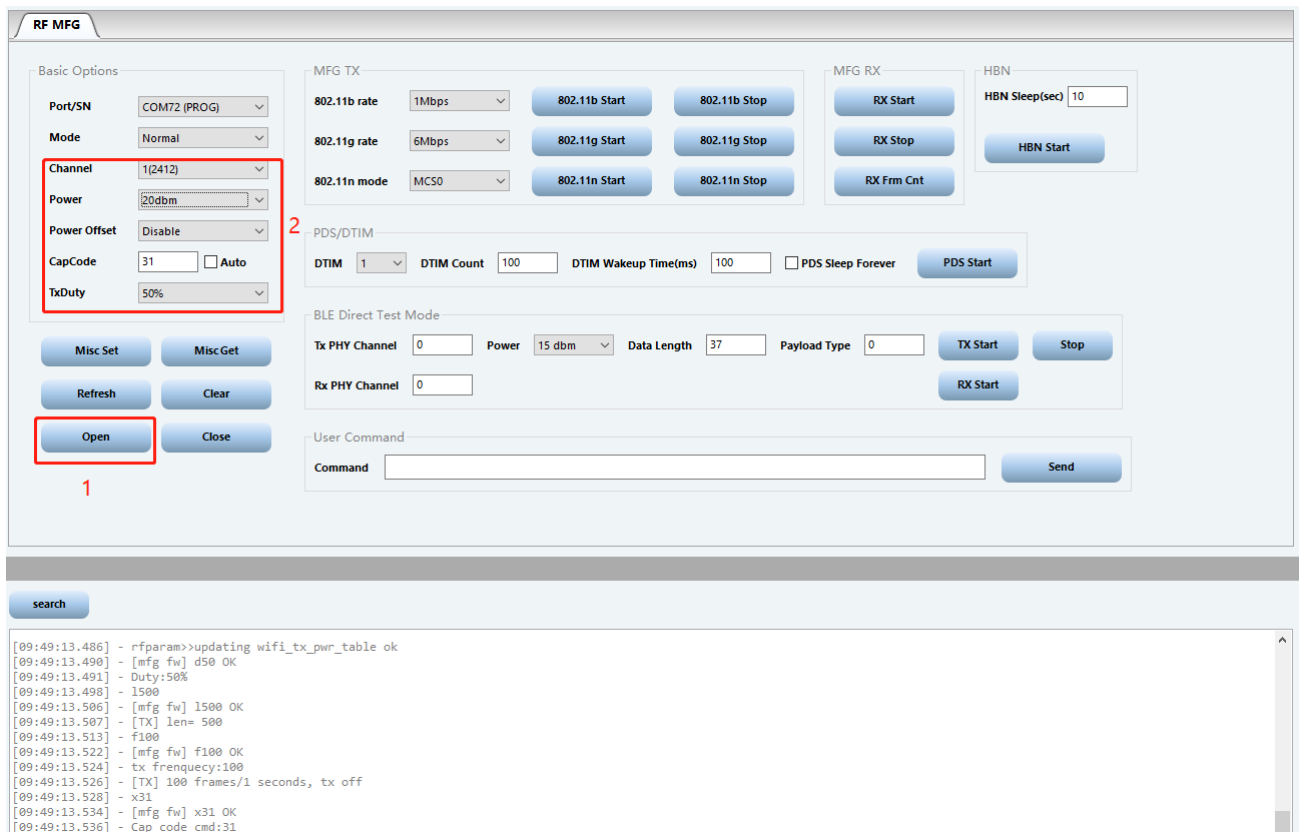


图 4.3: RF MFG 界面主要配置

注解：完成频偏校准之后，后续功率校准时 **capcode** 中一定要填写正确的频偏校准值，以便确保射频频率正确。

训练功率补偿经验值方法如下：

1. 按示意图连接好开发板、测试 PC 和射频测试仪器；
2. 完成自启动 MFG 固件烧写或者下载 (如已经下载可跳过)；
3. 通过 DevCube 的 View->RF MFG 进入到 RF MFG 测试界面，选择对应的串口，点击 Open，复位模组，可以在 Log 窗口中看到 MFG 固件启动 Log；
4. 将 Power Offset 设置为 Disable, 将基本配置项中 Power 选项设置为用户期望的目标功率 Pset（比如 17dBm）；
5. 切换待测试信道（比如信道 1）；
6. 设定发送数据包（推荐 11n MCS7），使用射频仪器测量实际输出功率 Pmea；
7. 计算功率偏差值  $Perr = Pset - Pmea$ （此处需要四舍五入取整），即为该信道的功率偏差值；
8. 计算功率训练结果  $pwr\_offset = 10 + Perr$ ，即为该信道下 DTS 文件中 pwr\_offset 补偿值；
9. 针对信道 1-14 重复步骤 4-8，将所有待测板测试完毕后，求取平均值，并将均值写入 DTS 文件中 pwr\_offset 数组；

---

注解：在训练功率补偿经验值时，一定确保 **capcode** 中填写正确的频偏校准值，并且 **Auto** 选项没有勾选。

---

---

注解：DTS 文件中的 **pwr\_table\_11b** / **pwr\_table\_11g** / **pwr\_table\_11n** / **pwr\_table\_ble** 数组分别是用户设定的不同模式下的目标功率。**pwr\_offset** 数组是用于调整不通信道下的功率补偿值，补偿的单位为 **1dbm**(与产测时使用仪器校准不同，仪器校准写入到 **efuse** 中的功率补偿单位为 **0.25dbm**)，补偿范围为 **-4** 到 **3dbm**，便于负数的表示我们使用了 **10** 作为 **base**，即 **pwr\_offset=10** 时，不做补偿，**pwr\_offset=11** 时，功率增加 **1dbm**，**pwr\_offset=9** 时，功率减少 **1dbm**。

---

## 4.2 验证频偏和功率补偿经验值

验证频偏经验值方法如下：

1. 准备一块待测的模组。
2. 确保 DTS 文件中的 **xtal\_mode** 为“FM”，**xtal** 已经填写正确的训练值。
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，此时会将 DTS 文件中的配置一起烧录进去。
4. MFG 固件运行后，通过 DevCube 的 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 DTS 文件中 **xtal\_mode** 设定的模式，从 DTS 加载频偏值。
5. 设定发送数据包，使用仪器测量实际输出的频率。

勾选 CapCode 的 **Auto** 选项以后，MFG 固件输出相关 log 信息如下，没有在 **Efuse** 中找到 **Capcode** 的频偏校准，使用 **dtb** 生成的 **rftlv** 对应的数值：

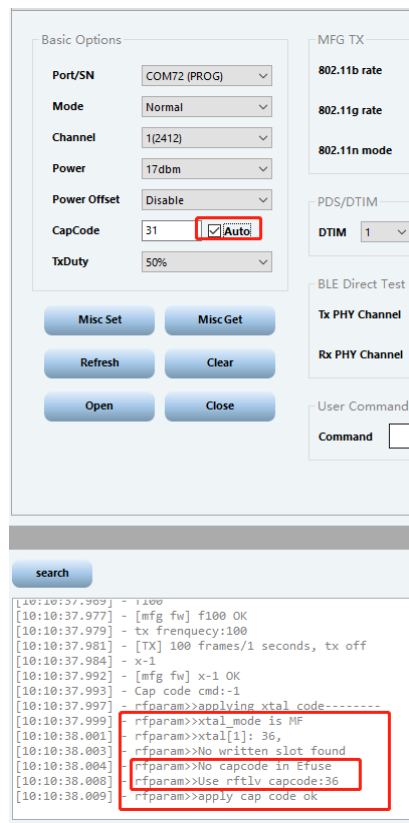


图 4.4: 勾选 Auto 后的 log 信息

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 设置的顺序加载频偏值，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。

验证功率补偿经验值方法如下：

1. 准备一块待测的模组。
2. 确保 DTS 文件中的 pwr\_mode 为"FB"，pwr\_offset 已经填写正确的训练值。
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，此时会将 DTS 文件中的配置一起烧录进去。
4. MFG 固件运行后，通过 DevCube 的 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 DTS 加载功率校准值。
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，没有在 Efuse 中找到 pwoffset 的功率校准，使用 dts 生成的 rftlv 对应的数值：

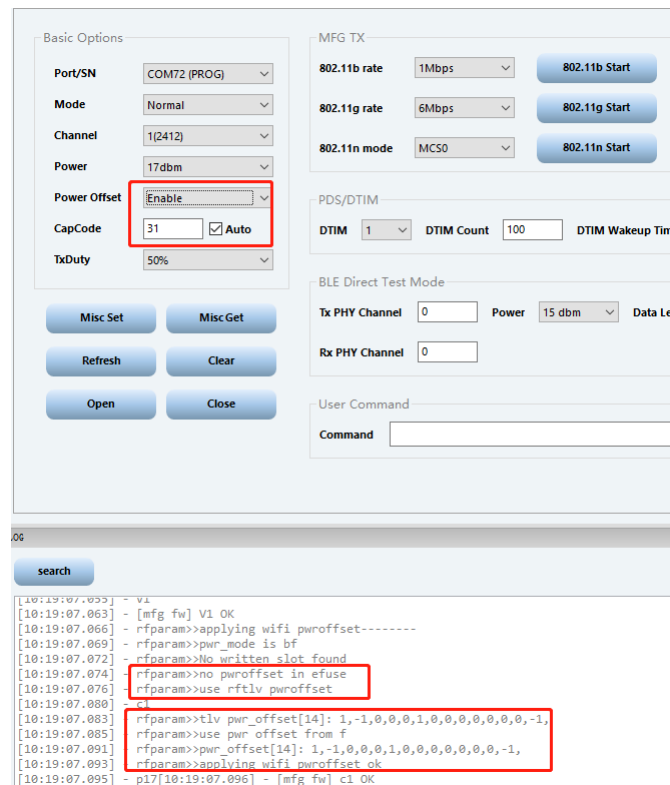


图 4.5: 使能 Power Offset 的 log 信息

注解：在验证功率补偿经验值时，如果做了频偏校准，需要将 CapCode 的 Auto 选项勾选，使用训练后的频偏值。

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 pwr\_mode 设置的加载顺序加载功率校准，先从 Efuse 加载，加载失败后从 DTS 文件中读取。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。

## RF 产测方案 2--RF 产测校准方案

对于需要做 RF 功率和频偏校准的用户,MFG 提供了校准的方法。用户可以测试每台产品获得功率偏移和频率偏移,并且将功率偏移和频率偏移写到 **Efuse** 中,再次启动时,会根据 DTS 的模式设置功率偏移和频率偏移。产测/校准的测试环境示意图如下所示。

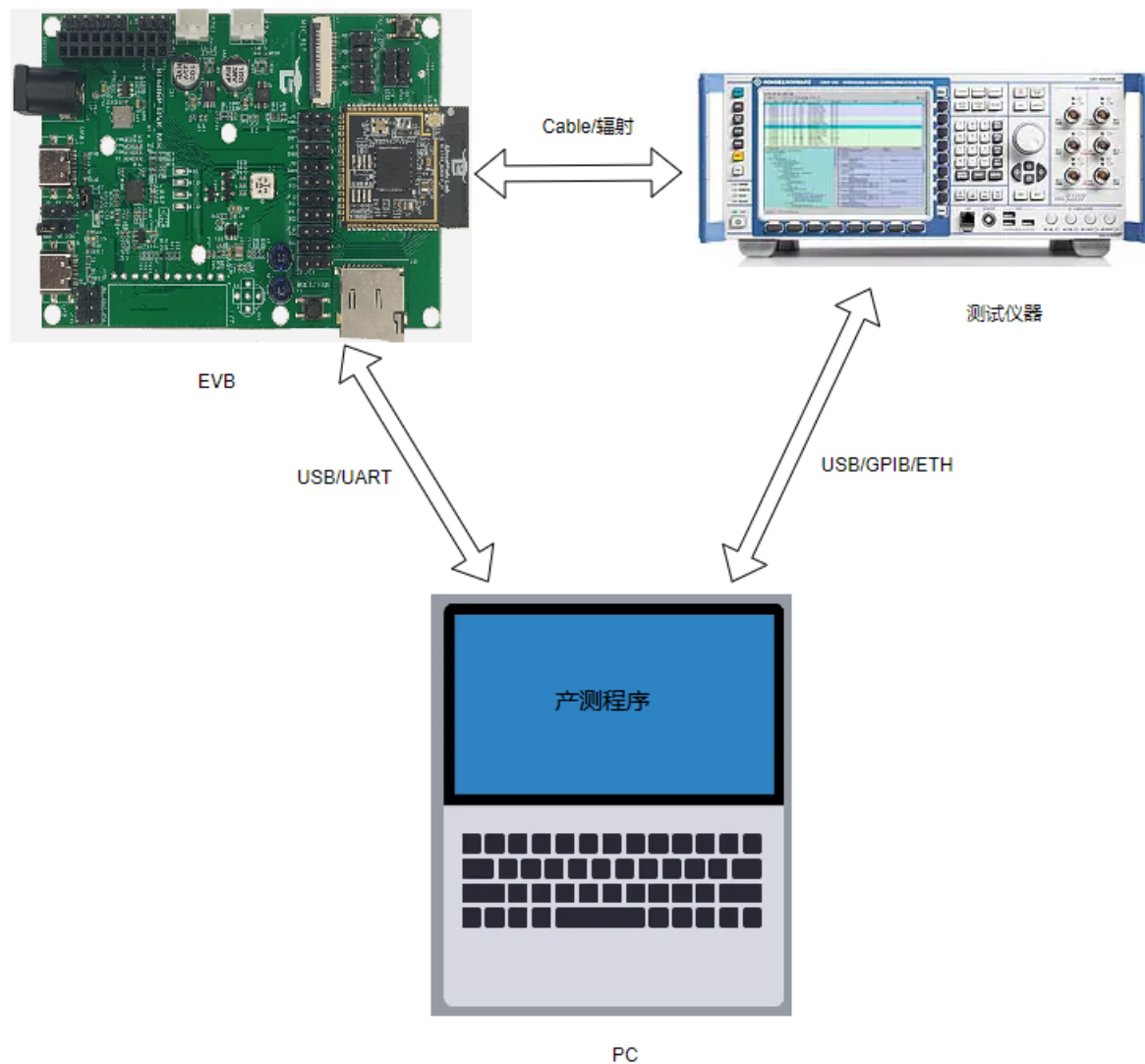


图 5.1: RF 产测/校准方案

## 5.1 频偏校准的流程

1. 按示意图连接好开发板,PC 和 RF 测试仪器。
2. 完成 MFG 固件的烧写或者下载, 具体请参考文档 < 量产指导 (烧录篇) >。
3. MFG 固件运行后, 主要的发送命令如下所示, 命令说明见本文章节 <RF 测试命令>。

```

V0      //去掉之前的校准值
c1      //设置channel为1, 用户可以根据实际情况设置
p15     //设置功率
d50     //设置发包的duty
1500    //设置数据包的长度
f100    //设置数据包的发送频率

```

(continues on next page)

(continued from previous page)

```
mlm27 //设置数据包制式
t1     //使能发送
```

4. 这时候产测芯片处于发射状态, 使用仪器测量信号频偏, 根据测量结果, 调整 **CapCode** 值, 假如这里设置的值为 30, 则发送命令:

```
X30
```

5. 重复步骤 4, 直到得到最终正确频率。
6. 假设调整频偏值后得到校准后的频偏值 31, 使用 **WEX** 等命令写入频偏校准值, 频偏校准完成。

```
WEX31 //预写入数据
LEX    //读取预写入的数据, 确保写入正确
SEX    //写入数据
REX    //读取efuse中的数据, 确保写入正确
```

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。

对于功率校准, 写 **MAC** 地址等写 **Efuse** 的操作, 也是同样的流程。

## 5.2 功率校准的流程

1. 按示意图连接好开发板, PC 和 RF 测试仪器。
2. 完成 **MFG** 固件的烧写或者下载, 具体请参考文档 < 量产指导 (烧录篇) >。
3. **MFG** 固件运行后, 主要的发送命令如下所示, 命令说明见本文章节 < RF 测试命令 >。

```
V0      //去掉之前的校准值
c1      //设置channel为1, 用户可以根据实际情况设置
p15     //设置功率
d50     //设置发包的duty
1500    //设置数据包的长度
f100    //设置数据包的发送频率
mlm27   //设置数据包制式
t1      //使能发送
```



- 使用仪器测量实际输出的功率, 并计算出设置功率与实际功率的偏差, 然后先停止发送, 再更改待测信道, 最后再次使能发送。

```
t0    //停止发送
c7    //更换channel到7, 用户可以根据实际情况设置
t1    //使能发送
```

- 测试信道 1,7,13 后得到 3 个功率偏差值, 假设偏差值为 0.25dB,-0.25dB,0.5db, 使用 WEP 命令写入功率校准值, 功率校准完成。

```
WEP1,0,0,0,0,0,-1,0,0,0,0,0,2,0 //预写入数据
LEP    //读取预写入的数据, 确保写入正确
SEP    //写入数据
REP    //读取efuse中的数据, 确保写入正确
```

注解: 由于 Efuse 的空间限制, 我们只存储了高中低三个 channel 的功率校准, 其他 channel 依靠芯片固件自动插值计算, efuse 存储的功率校准步长为 0.25dB, 因此当误差为 0.25dB 时, 写入 1, 当误差为 0.5dB 时, 写入 2。

### 5.3 验证频偏和功率校准结果

验证频偏校准结果方法如下:

- 准备一块做完产测的模组。
- 确保 DTS 文件中的 `xtal_mode` 为"MF"。
- 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。
- MFG 固件运行后, 通过 DevCube 的 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, 优先从 Efuse 中加载频偏校准数值。
- 设定发送数据包, 使用仪器测量实际输出的频率。

勾选 CapCode 的 Auto 选项以后, MFG 固件输出相关 log 信息如下, 在 Efuse 找到 capcode, 直接使用:

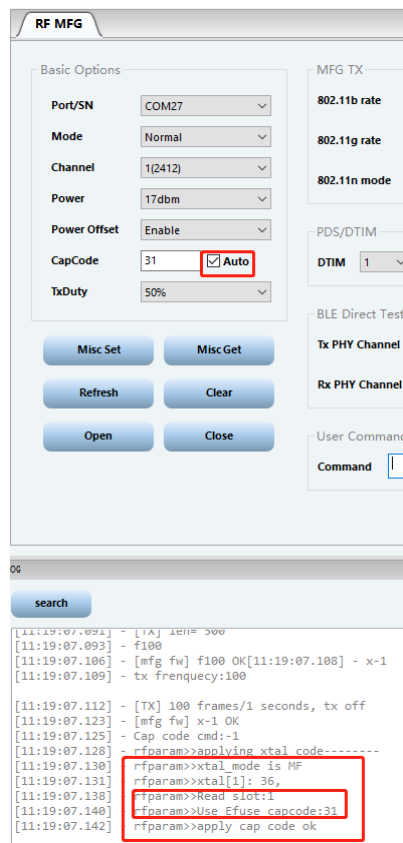


图 5.2: 勾选 Auto 后的 log 信息

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` 优先从 Efuse 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。

验证功率校准结果方法如下：

1. 准备一块做完产测的模组。
2. 确保 DTS 文件中的 `pwr_mode` 为“BF”。
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 DevCube 的 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，优先从 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，在 Efuse 找到 `pwr offset`, 直接使用：

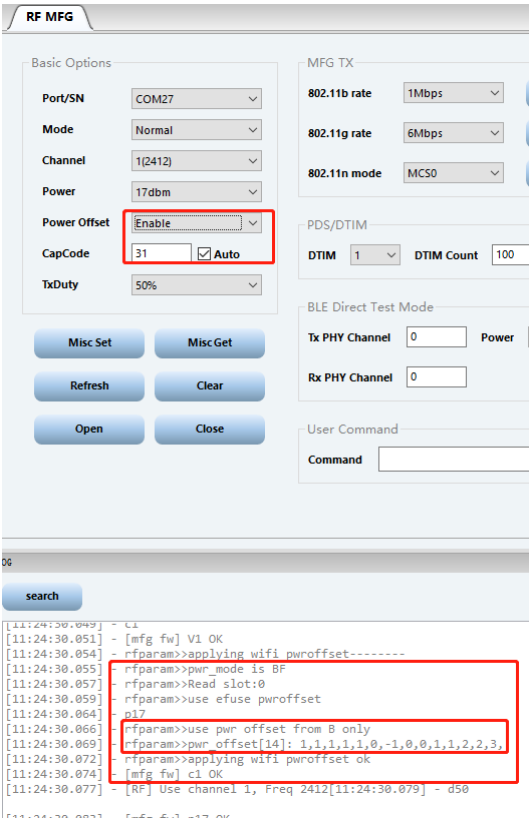


图 5.3: 使能 Power Offset 的 log 信息

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 优先从 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。

5.4 方案 1 与方案 2 的对比

表 5.1: 测试方案对比

差异点	方案 1	方案 2
测试设备	PC/RF 测试仪器	PC/RF 测试仪器
生产设备	PC	PC/RF 测试仪器
测试成本	高	高
生产成本	低	高
性能一致性	低	高

## MAC 地址写入

如果用户需要在 RF 产测完毕后，写入自己的 MAC 地址，可以通过如下接口实现：

```
WEM18:B9:05:60:0E:74 //预写入MAC地址 18:B9:05:60:0E:74
LEM //读取预写入的数据，确保写入正确
SEM //写入数据
REM //读取 efuse 中的数据，确保 MAC 地址写入正确
```

对于使用三元组或者五元组的客户，一般情况下并不是直接将 MAC 地址单独写入 Efuse 中，而是将三元组或者五元组烧写到 Flash 中，这时候，应该固件直接从 Flash 上读取保存 MAC 地址的三元组或者五元组信息，提取 MAC 地址作为设备的 MAC 地址，因此不需要在产测时写入 MAC 地址这一步。

## RF 参数存储方案介绍

对于《方案 1--训练频偏和功率补偿经验值》，其功率补偿和频偏数值是直接存储在 DTS 文件中，训练完成后，将 DTS 文件交给研发，再生成最终的量产文件即可。

对于《方案 2--RF 产测校准方案》，其功率补偿和频偏数值是产线校准程序直接写入到芯片对应的 **Efuse** 中的。由于 **Efuse** 具有一次性写入的特点，而且 **Efuse** 空间有限，所以对于频偏和功率校准的次数是有限制的，具体情况如下：

项目	可写入次数	备注
频偏校准	2 次	
功率校准	3 次	
MAC 地址	2 次	芯片出厂已经带有厂商 MAC 地址

上位机程序与 MFG 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，所有命令均是字符串类型，常用的命令表总结如下：

命令	参数	说明/示例
H	\	握手命令,应返回字符串"mfg\r\n"
T/t	0/1	t1/T1(开始发送),t0/T0(停止发送)
r	:s/:p/:g	r:s(开始接收),r:g(统计接收信息),r:p(停止接收)
f	频率值	f500,表示发送的频率为500Hz
p	功率值	p10-p23,表示设置功率为10dbm-23dbm
l	包长度	l500,表示发送的包长为500 byte
d	50/100	d50表示发送的调制波占空比为50%
B/b	0/1/2/3	B0(11b,1M),B1(11b,2M),B2(11b,5.5M),B3(11b,11M)
G	0/1/2/3/4/5/6/7	Long Preamble, 11g, G1-G7分别对应速率 6M,9M,12M,18M,24M,36M,48M,54M
g	0/1/2/3/4/5/6/7	Short Preamble, 11g, G1-G7分别对应速率 6M,9M,12M,18M,24M,36M,48M,54M
m	[s/l][g/m][2/4][0/1/2/3/4/5/6/7]	[s/l] s表示short_gi,l表示long_gi. [g/m]g表示FORMATMOD_HT_GF,m表示FORMATMOD_HT_MF [2/4]2表示带宽20M,4表示带宽40M [0/1/2/3/4/5/6/7]表示速率MCS0~MCS7的索引 例mlm20表示long_gi,调制方式为FORMATMOD_HT_MF,20M带宽,以 MCS0速率发送
c	1~14	c10,表示设置发送信道10
X/x	0-255	设置频偏值
Y/y	:i/:v/:d/:p/:c/:t/:f/:x/:M	获取当前状态,:i/:v/:d/:p/:c/:t/:f/:x/:M分别表示获取 tx_duty/版本/时间/功 率/信道/发送状态/频率/capcode/发送模式 信息
M	0/1	0(normal mode),1(单载波模式)
WEP	channel 1 power offset~channel 14 power offset	写入WiFi功率偏移值,例WEP1,-2,3,-4,0,0,0,0,0,0,0,0,0,0
REP	\	返回功率偏移值
WEX	频偏值	写入频偏值,例 WEX32

REX	\	返回频偏值
WEE	power offset 1~ power offset 20	写入BLE功率偏移值,例WEE1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0
REE	\	返回BLE功率偏移值
ETE	[00-27][00-f][00-07][00-04][00-14]	ETE命令的参数均以16进制表示 [00-27]表示信道0-信道39 [01-f]表示发送数据的长度 [00-07]payload type [00-04]表示速率,分别对应1Mbps/2Mbps/125kbps/500kbps [00-14]表示功率0dbm~20dbm
ERE	[00-27][00-03]	ERE命令的参数均以16进制表示 [00-27]表示信道0-信道39 [00-03]表示接收速率,分别对应1Mbps/2Mbps/Coded PHY
EE	\	停止BLE发送

图 8.1: 命令列表

全部支持的命令详解如下:

## 8.1 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”后可以进行正常的测试。

## 8.2 TX on/off

on:t1

off:t0

## 8.3 TX modulation

### 8.3.1 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。

---

### 8.3.2 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]

### 8.3.3 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]

## 8.4 2.4g channel

channel idx = 1 - 13

- 命令：c[channel idx]

## 8.5 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。

---



## 8.6 TX frame length

- 命令: l[length]

## 8.7 TX frequency

max value=1000

- 命令: f[freq]

## 8.8 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]

## 8.9 HBN

enter into hbn mode

1. rtc wake up mode hr[second]
2. gpio wake up mode TOD0

## 8.10 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]

## 8.11 Get MFG FW version

- 命令: y:v
- 返回: ###version:[version]

## 8.12 Get MFG FW building infomation

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

## 8.13 Get current power level

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

## 8.14 Get current channel

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

## 8.15 Get current tx status

- 命令: y:t
- 返回: ###tx:[0 or 1]

## 8.16 Get tx frequency

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

## 8.17 Get cap code

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

## 8.18 Get MFG mode

- 命令: y:M
- 返回: ###mfgmode:[MFG mode]

## 8.19 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。

## 8.20 Set MFG Test(CW) mode

0 for normal mode, 1 for CW test mode

- 命令: M[MFG mode]

## 8.21 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。

### 8.21.1 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

### 8.21.2 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

### 8.21.3 Program data to efuse

- 命令: SEA
- 返回: Save efuse OK

### 8.21.4 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

## 8.22 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。

### 8.22.1 Write cap code to efuse buffer

- 命令: WEX[cap code]

### 8.22.2 Load cap code from efuse buffer

- 命令: LEX
- 返回: Cap code2:[cap code]

### 8.22.3 Program cap code to efuse

- 命令: SEX

### 8.22.4 Read cap code from efuse

- 命令: REX
- 返回: Cap code2:[cap code]

### 8.22.5 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.

---

### 8.22.6 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]

### 8.22.7 Program power offset to efuse

- 命令: SEP

### 8.22.8 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]

### 8.22.9 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。

### 8.22.10 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

### 8.22.11 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

### 8.22.12 Program mac address to efuse

- 命令: SEM

### 8.22.13 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

## 8.23 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;
```

## 8.24 Reset MFG FW

- 命令: Reset

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

## 8.25 Set tx duty

- 命令: `d[duty]`

---

注解: Duty value is between 0-100

---

## 8.26 Get tx duty

- 命令: `y:i`
- 返回: `###duty:[tx duty]`

## 8.27 BLE Test

### 8.27.1 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.

### 8.27.2 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.

### 8.27.3 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.



#### 8.27.4 BLE test stop

- 命令：EE