



射频性能测试 使用手册

版本： 2.11

版权 @ 2020

www.bouffalolab.com

1 版本记录	6
2 概述	7
3 下载开发烧录软件工具包	9
4 烧写测试固件	10
5 运行测试固件	14
6 频偏补偿设置	15
7 发送设置	17
7.1 Channel 和 Power 设置	17
7.2 发送数据包模式设置	18
7.2.1 11b 数据包发送	19
7.2.2 11g 数据包发送	19
7.2.3 11n 数据包发送	19
8 接收设置	20
9 设置/使用默认发射功率及频偏补偿值	21
10 BLE 测试	23
11 PDS/DTIM 设置	24
12 HBN 设置	25
13 CW 测试模式	26
14 串口通信命令	27
14.1 Shakehand	27
14.2 TX on/off	27

14.3	TX modulation	27
14.3.1	2.4G 11n	27
14.3.2	2.4G 11g	28
14.3.3	2.4G 11b	28
14.4	2.4g channel	28
14.5	2.4g tx power	28
14.6	TX frame length	28
14.7	TX frequency	28
14.8	PDS	29
14.9	HBN	29
14.10	RX	29
14.11	Get MFG FW version	29
14.12	Get MFG FW building infomation	29
14.13	Get current power level	29
14.14	Get current channel	30
14.15	Get current tx status	30
14.16	Get tx frequency	30
14.17	Get cap code	30
14.18	Get MFG mode	30
14.19	Set cap code	30
14.20	Set MFG Test(CW) mode	30
14.21	Save parameters to efuse	30
14.21.1	Write cap code to efuse buffer	31
14.21.2	Load cap code from efuse buffer	31
14.21.3	Program cap code to efuse	31
14.21.4	Read cap code from efuse	31
14.21.5	Write power offset to efuse buffer	31
14.21.6	Load power offset from efuse buffer	31
14.21.7	Program power offset to efuse	31
14.21.8	Read power offset from efuse	32
14.21.9	Write mac address to efuse buffer	32
14.21.10	Load mac address from efuse buffer	32

14.21.11 Program mac address to efuse	32
14.21.12 Read mac address from efuse buffer	32
14.22 Save parameters to flash	32
14.22.1 save cap code to flash	32
14.22.2 read cap code from flash	33
14.22.3 save default power to flash	33
14.22.4 read default power from flash	33
14.22.5 select default power in flash	33
14.23 Reset MFG FW	33
14.24 Set tx duty	33
14.25 Get tx duty	33
14.26 BLE Test	34
14.26.1 BLE TX Power	34
14.26.2 BLE TX	34
14.26.3 BLE RX	34
14.26.4 BLE test stop	34

List of Figures

2.1 工具界面图	7
3.1 开发烧录软件工具包	9
4.1 IOT 模组评估套件	10
4.2 烧写界面	11
4.3 硬件连接原理图	12
4.4 烧录成功界面	13
5.1 程序成功运行的 log	14
6.1 更新补偿值	16
7.1 设置 Channel 和 Power 的参数	17
7.2 11b 数据包设置速率	19
7.3 11g 数据包设置速率	19
7.4 11n 数据包发送界面	19
8.1 接收数据包效果图	20
9.1 BLE Payload Type	21
11.1 PDS 模式参数设置	24
12.1 HBN 模式参数设置	25
13.1 测试模式参数设置	26

表 1.1: 修改记录

版本	更新内容
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 的设定, 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
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 Bouffalo Lab 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。

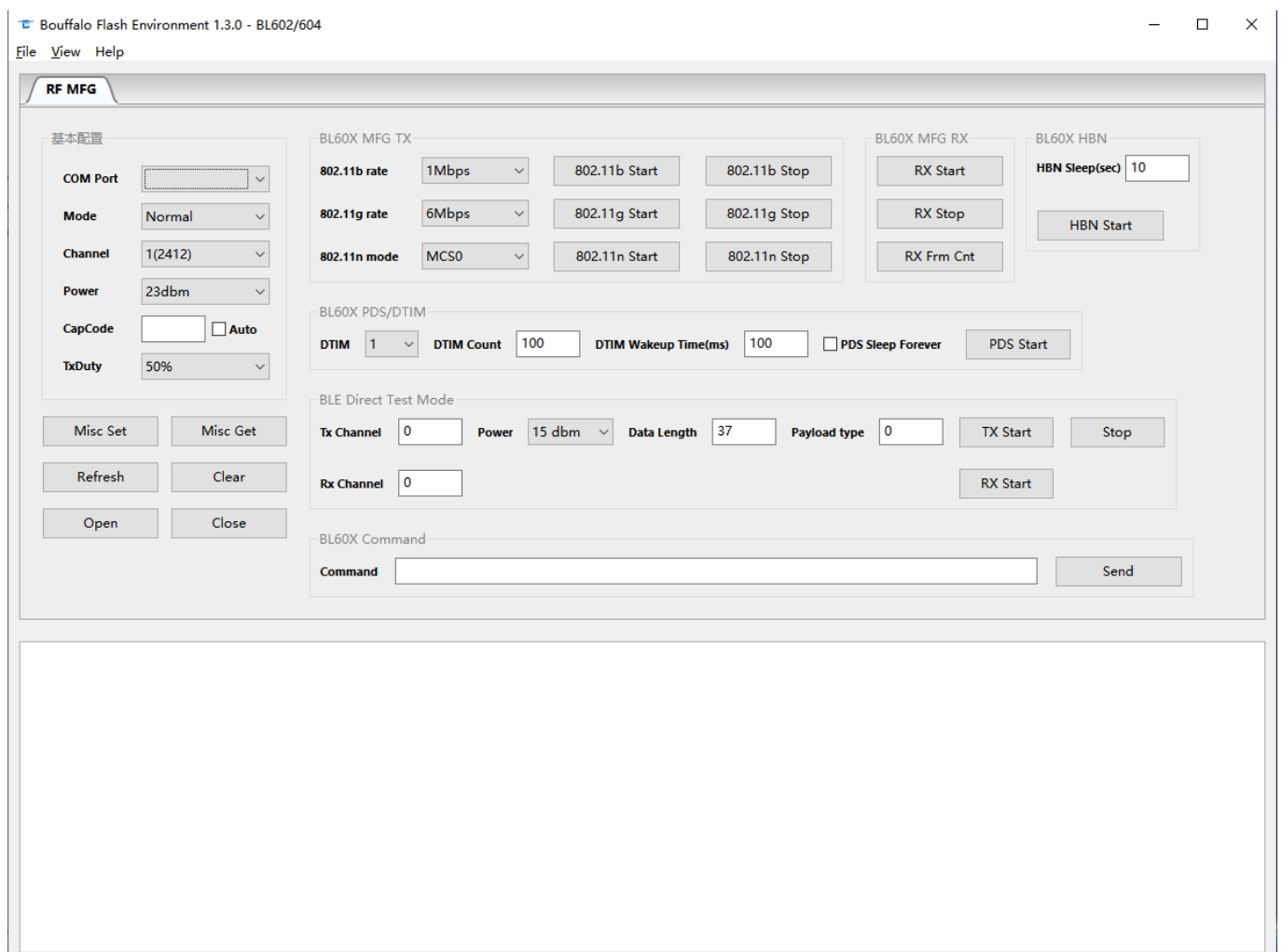


图 2.1: 工具界面图

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 数据包接收 (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

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 [Bouffalo Lab Dev Cube](#)，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。它是客户开发博流各种类型芯片使用到的工具集。

名称	修改日期	类型	大小
bl56x	2020/7/14 14:18	文件夹	
bl60x	2020/7/20 9:03	文件夹	
bl562	2020/7/14 14:19	文件夹	
bl602	2020/7/16 17:16	文件夹	
common	2020/7/14 14:18	文件夹	
docs	2020/7/14 14:18	文件夹	
openocd	2020/7/14 14:18	文件夹	
BLDevCube.exe	2020/7/20 10:01	应用程序	24,135 KB
JLink.exe	2020/6/2 14:41	应用程序	273 KB
JLinkARM_32.dll	2020/6/2 14:41	应用程序扩展	13,988 KB
openocd.exe	2020/6/2 14:41	应用程序	4,579 KB
version.txt	2020/7/20 9:58	文本文档	1 KB

图 3.1: 开发烧录软件工具包

通过 Flash 烧写工具，将 RF 的测试固件烧写到 Flash 中，镜像成功下载后，通过将 Boot 引脚跳到低电平，按下复位键就可以启动 firmware 固件程序进行 RF 测试了。

对于 RF 评估测试，如果是直接从博流拿到的 RF 评估测试板，里面会提前烧录好测试固件，用户可以跳过烧写测试固件这一步骤，直接连接评估测试板到 PC，运行固件进行 RF 测试评估，详细参考《运行测试固件》这一章节。

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 BL602 的 IOT 模组评估套件烧写为例，介绍烧写过程。BL602 的 IOT 模组评估套件如下图所示。

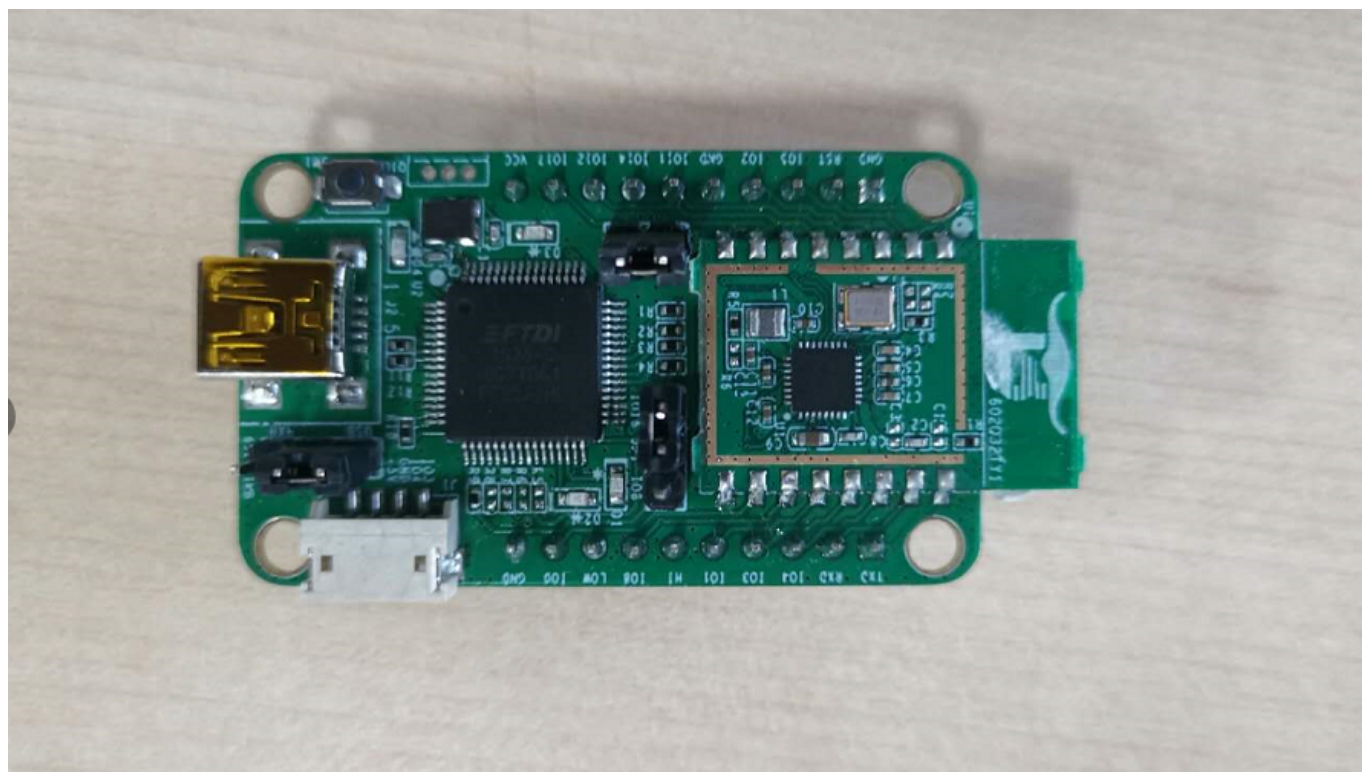


图 4.1: IOT 模组评估套件

套件由 IOT 模组和模组底板组成，模组底板使用 Mini USB 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
- DTR: 与模组的 Boot 引脚相连，用于控制芯片从 UART 启动还是 Flash 启动
- RTS: 与模组的 Reset 引脚相连，用于控制芯片的复位

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 BLDevCube.exe, 在 Chip Type 中选择 BL602/604，进入如下烧写界面。

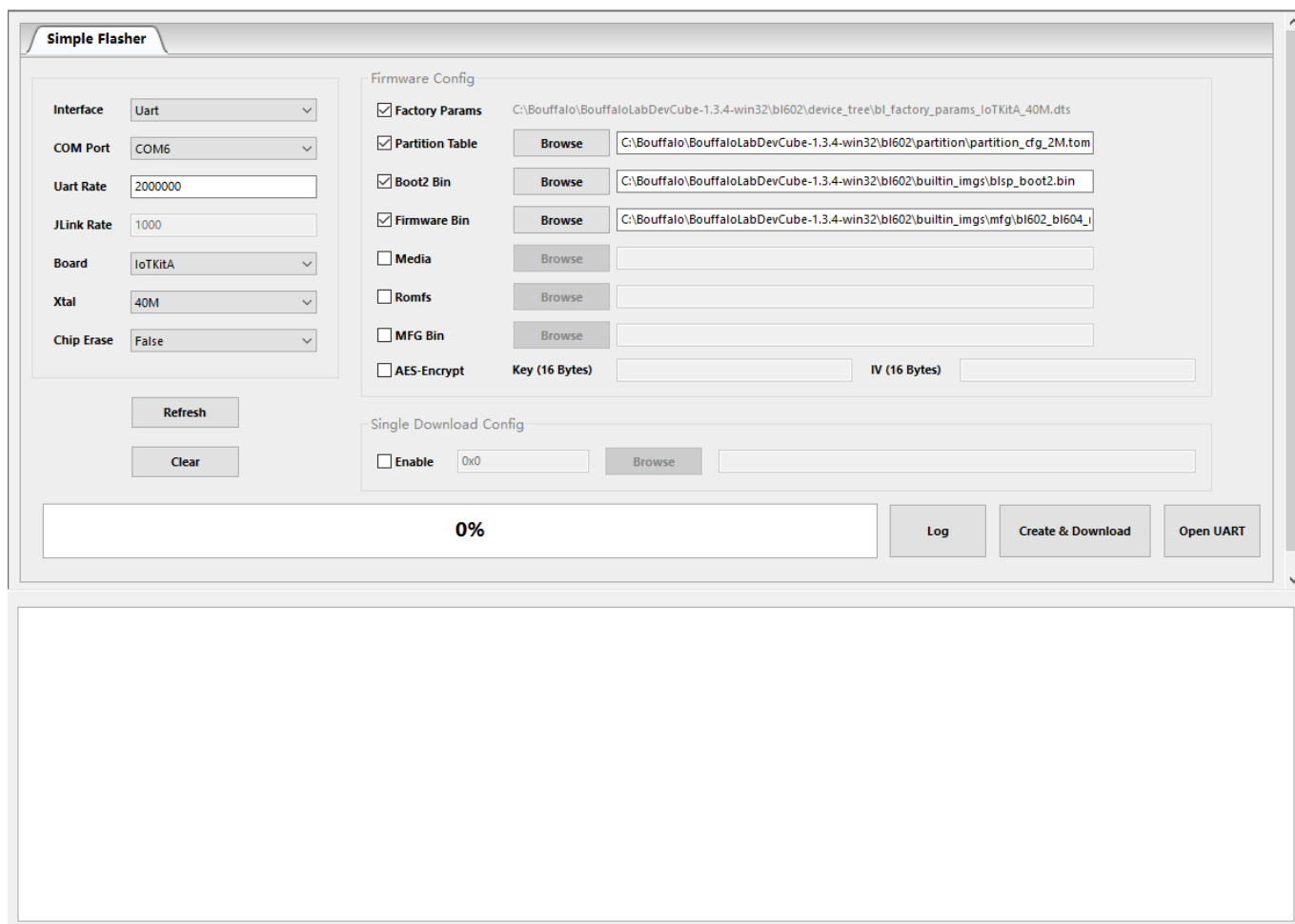


图 4.2: 烧写界面

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

- **Uart Rate:** 当选择 **UART** 进行烧写的时候，填写波特率，可以填写 **2M 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**

在右侧烧录镜像配置，分别选择：

- bl602/builtin imgs/mfg/bl602 bl604 mfg gu 40m.bin

ADBUS0 16 R30 NM R0402 VDD33

ADBUS1 17 R25 NM R0402 VDD33

ADBUS2 18 R24 NM R0402 VDD33

ADBUS3 19 R23 NM R0402 VDD33

ADBUS4 21 R10 NM R0402 VDD33

ADBUS5 22 R9 NM R0402 VDD33

ADBUS6 23 R7 NM R0402 VDD33

ADBUS7 24 R6 NM R0402 VDD33

GPIO7

UART0_TXD

GPIO2

GPIO5

CHIP_EN

GPIO8

Remove if manually

图 4.3: 硬件连接原理图

在上图中，电阻 R5,R8 是用于 USB 转串口的 DTR、RTS 与芯片的 BOOT 引脚、RST 引脚的连接。如果这两个电阻存在的话，烧写程序可自动控制芯片的启动方式。如果不存在的话，需要用户将 BOOT 引脚根据需求跳到高电平或者低电平并通过板子上的按键实现 RESET。

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

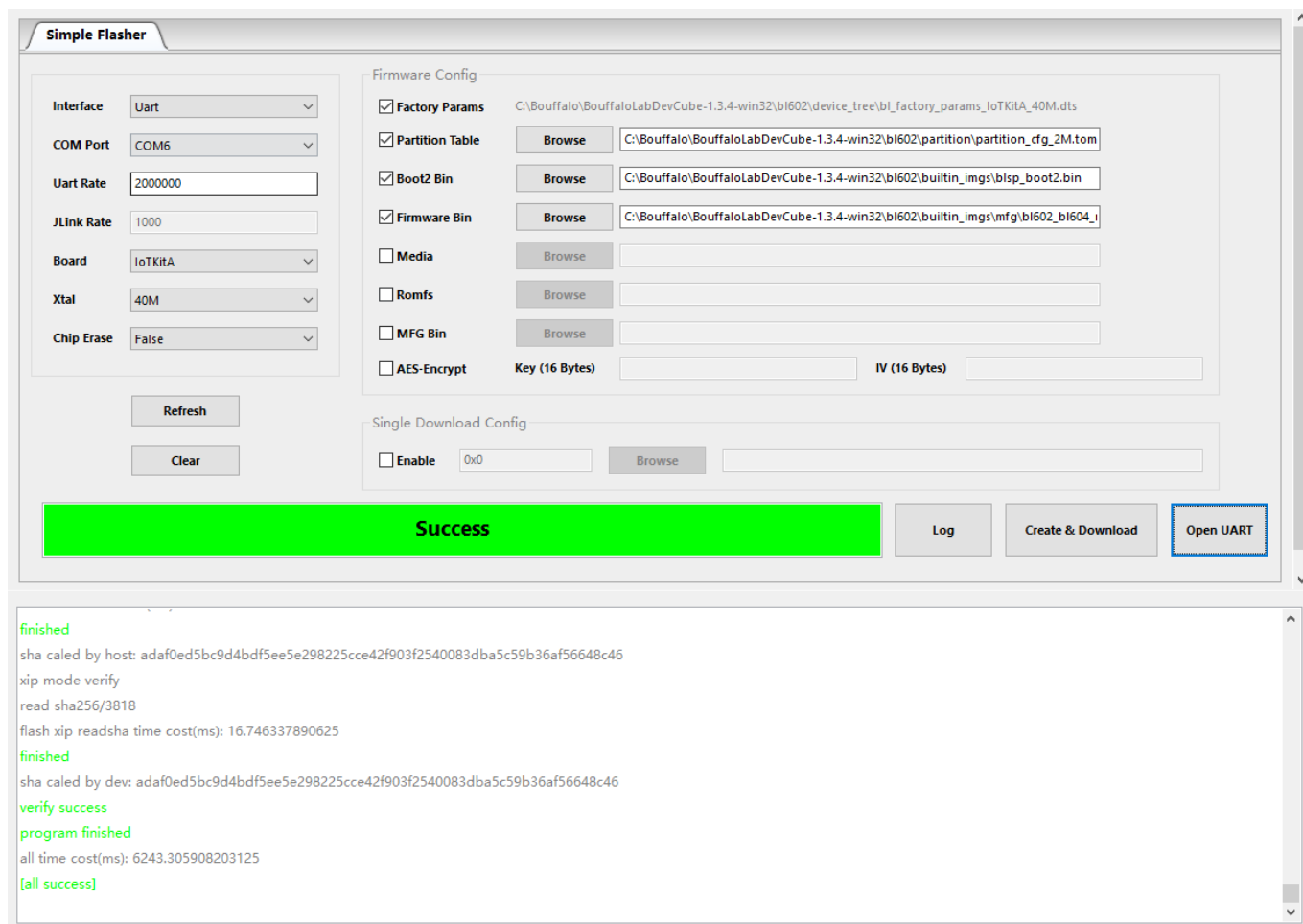


图 4.4: 烧录成功界面

运行测试固件

完成测试固件的下载以后，将芯片的 **Boot** 引脚跳到低电平，然后按下复位键，芯片就可以运行 **RF** 测试固件了，对于博流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。在运行 **BLFlashEnv.exe** 后，通过 **View->RF MFG** 进入到 **RF MFG** 测试界面。选择使用到的 **COM** 号（本例中是 **COM6**），点击 **Open** 按钮，即可看到固件程序成功运行的 **log**，示例如下。

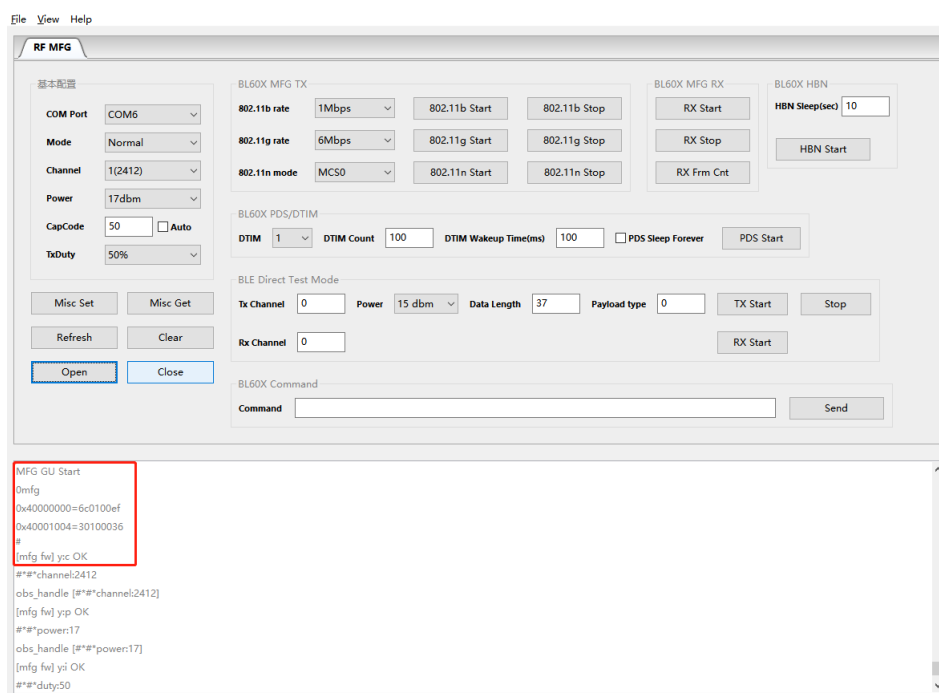


图 5.1: 程序成功运行的 log

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，BL60X 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: BL606 对应的电容补偿值

XTAL Loading Capacity (pF)	Capacity Code
8	40
10	58

表 6.2: BL602 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。

The image shows a software interface titled "基本配置" (Basic Configuration). It contains several configuration fields:

- COM Port: COM6 (dropdown)
- Mode: Normal (dropdown)
- Channel: 1(2412) (dropdown, highlighted with a red box)
- Power: 17dbm (dropdown, highlighted with a red box)
- CapCode: 33 (text input) and ☐ Auto (checkbox)
- TxDuty: 50% (dropdown)

Below the configuration fields are several buttons:

- Misc Set (highlighted with a blue dashed border)
- 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。

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	18(BL606)/20(BL602)
	5.5Mbps	18(BL606)/20(BL602)
	2Mbps	18(BL606)/20(BL602)
	1Mbps	18(BL606)/20(BL602)

BL60X 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。

BL606/BL608 系列芯片针对功率偏差补偿预留了长度为 14 的数组空间 (Power_Offset[14]), 每个元素为 4bit, MSB 为符号位, 允许的功率偏差范围为-4~3 (即-4dB~3dB), 超出该取值范围则校准失败。

BL602 相比 BL606/BL608, 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 BL602 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。

关于 BL60X 系列芯片的详细校准机制, 内容请参考《BL60X_产测校准算法》。

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。

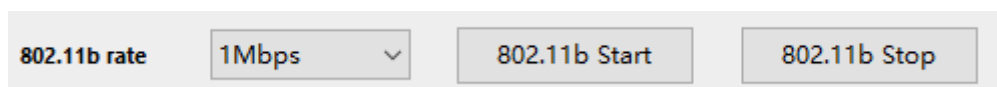


图 7.2: 11b 数据包设置速率

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。

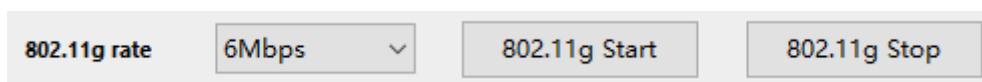


图 7.3: 11g 数据包设置速率

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。

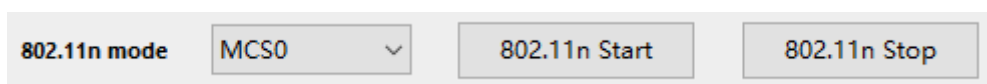


图 7.4: 11n 数据包发送界面

接收设置

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。

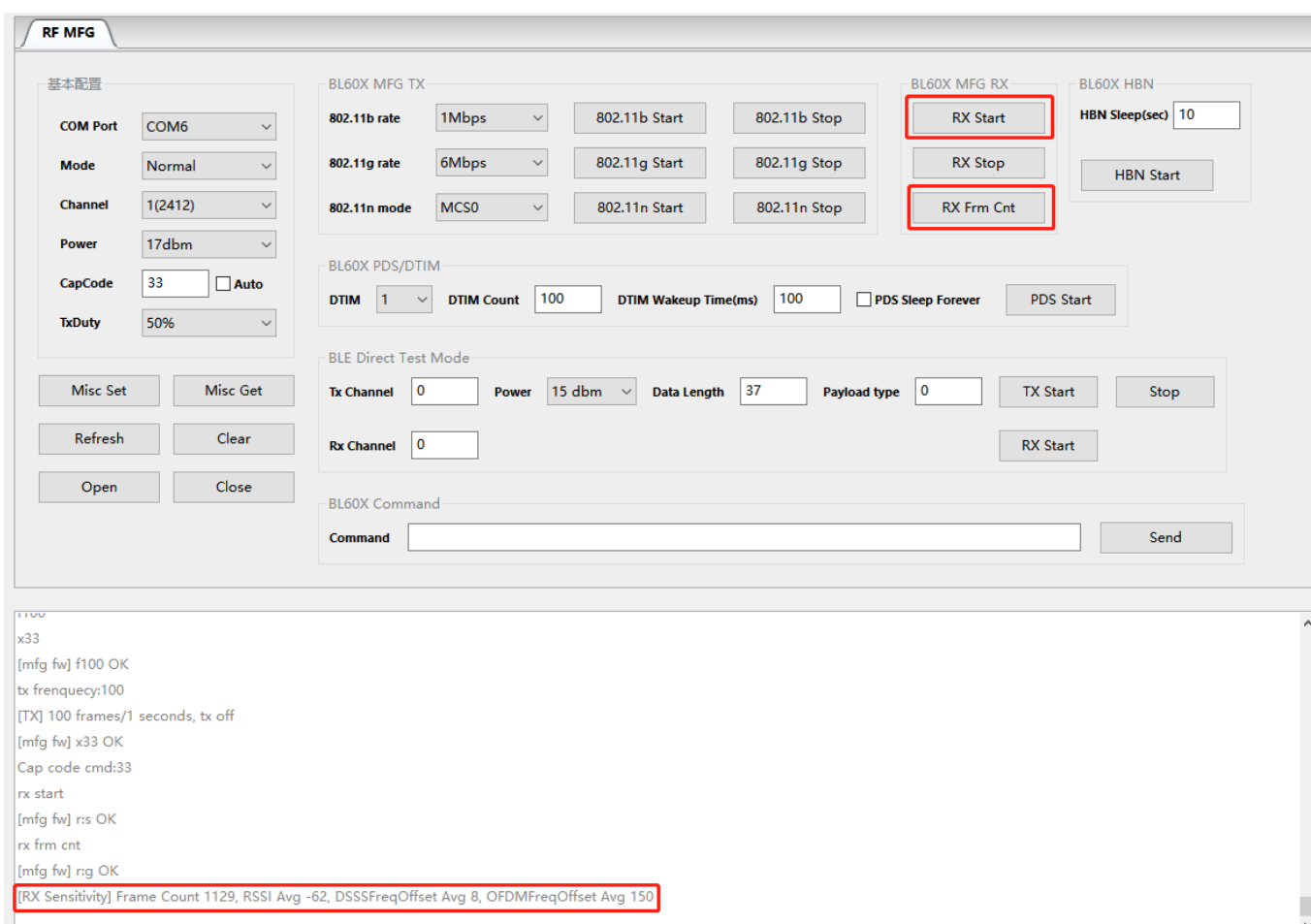


图 8.1: 接收数据包效果图

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

设置/使用默认发射功率及频偏补偿值

对于不需要对每个芯片都做 RF 功率和频偏校准的用户，RF MFG 提供了设置经验值的方法。用户可抽检一批产品，从而找出平均发射功率值和频偏值，以此 Golden 值作为默认的发射功率和频偏。

1. 设置频偏校准值

Command 输入框中输入 SFX[频偏值]，然后点击 Send 按钮。举例，如果设置频偏为 34，则输入框输入 SFX34

2. 设置 WiFi 默认功率

针对每个 Channel，每种模式，设置默认的发送功率。

1. 设置 11b 各个 channel 默认发射功率 (-1 代表是 13 个 channel) Command 输入框中输入 SFPDB-1,[Ch1 TX Power],[Ch2 TX Power]...[Ch13 TX Power]

举例如下：

SFPDB-1,18,19,18,19,18,19,18,19,18,19,18,19,20

2. 设置 11g 各个 channel 默认发射功率 (-1 代表是 13 个 channel) Command 输入框中输入 SFPDG-1,[Ch1 TX Power],[Ch2 TX Power]...[Ch13 TX Power]

举例如下：

SFPDG-1,19,19,18,19,18,19,18,19,18,19,18,19,20

3. 设置 11n 各个 channel 默认发射功率 (-1 代表是 13 个 channel) Command 输入框中输入 SFPDN-1,[Ch1 TX Power],[Ch2 TX Power]...[Ch13 TX Power]

举例如下：

SFPDN-1,20,19,18,19,18,19,18,19,18,19,18,19,20

如若设置某个 channel 的默认功率，则可以使用 SFPDB[Channel],[Channel TX Power]

或者 SFPDG[Channel],[Channel TX Power] 或者 SFPDN[Channel],[Channel TX Power]

3. 设置 BLE 默认功率

针对每个 Channel，设置默认的发送功率。

1. 设置 BLE 各个 channel 默认发射功率 (-1 代表是 40 个 channel) Command 输入框中输入 SFPE-1,[Ch0 TX Power],[Ch1 TX Power]...,[Ch39 TX Power]

举例如下：

SFPE-1,12,11,12,14,15,11,11,12,12,13,13,14,15,12,11,12,14,15,11,11,12,12,13,13,14,15,12,11,12,14,15,11,11,
12,12,13,13,14,15,12

如若设置某个 channel 的默认功率，则可以使用 SFPE[Channel],[Channel TX Power]

4. 使用默认功率和频偏

完成默认功率的设定后，对应发射功率和频偏分别选择 Auto 选项，即可使用之前设定在 Flash 区域的默认发射功率和频偏参数。

5. 查看默认功率和频偏

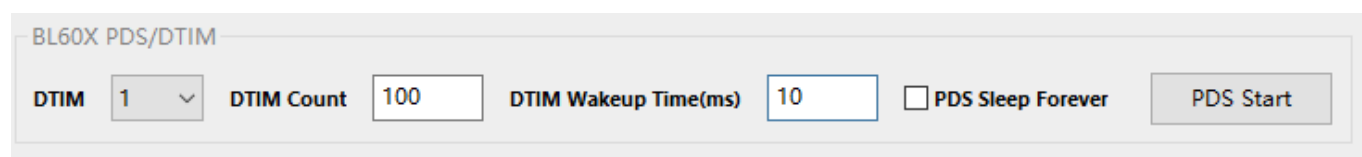
Command 输入框中输入 RFPD-1，即可通过 Log 查看存储 Flash 区域的 WiFi 默认功率和频偏。

Command 输入框中输入 RFPE-1，即可通过 Log 查看存储 Flash 区域的 BLE 默认功率。

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。



BL60X PDS/DTIM

DTIM 1 DTIM Count 100 DTIM Wakeup Time(ms) 10 ☐ PDS Sleep Forever PDS Start

图 11.1: 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。

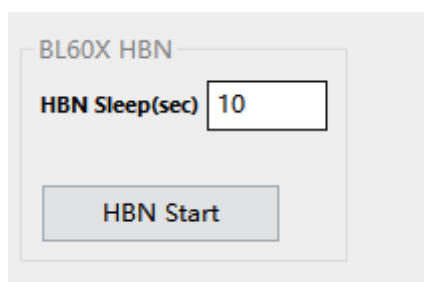


图 12.1: HBN 模式参数设置

MFG 支持 CW(Continue Wave) 测试模式，在界面 Mode 下拉菜单中选择 Test(CW) 模式即可，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。

The screenshot displays the RF MFG software interface with the following configuration details:

- 基本配置 (Basic Configuration):**
 - COM Port: COM6
 - Mode: Test(CW) (highlighted with a red box)
 - Channel: 1(2412)
 - Power: 20dbm
 - CapCode: 33 (Auto checkbox is unchecked)
 - TxDuty: 50%
- BL60X MFG TX:**
 - 802.11b rate: 1Mbps
 - 802.11g rate: 6Mbps
 - 802.11n mode: MCS0
- BL60X MFG RX:**
 - 802.11b Start, 802.11g Start, 802.11n Start: Disabled (greyed out)
 - 802.11b Stop, 802.11g Stop, 802.11n Stop: Disabled (greyed out)
 - RX Start: Enabled
 - RX Stop: Disabled (greyed out)
 - RX Frm Cnt: Disabled (greyed out)
- BL60X HBN:**
 - HBN Sleep(sec): 10
 - HBN Start: Disabled (greyed out)
- BL60X PDS/DTIM:**
 - DTIM: 1
 - DTIM Count: 100
 - DTIM Wakeup Time(ms): 100
 - PDS Sleep Forever: Unchecked
 - PDS Start: Disabled (greyed out)
- BLE Direct Test Mode:**
 - Tx Channel: 0
 - Power: 15 dbm
 - Data Length: 37
 - Payload type: 0
 - Tx Start: Disabled (greyed out)
 - Stop: Disabled (greyed out)
 - Rx Channel: 0
 - RX Start: Enabled
- BL60X Command:**
 - Command: (empty text field)
 - Send: Disabled (greyed out)

图 13.1: 测试模式参数设置

所有命令均是字符串类型。

14.1 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。

14.2 TX on/off

on:t1

off:t0

14.3 TX modulation

14.3.1 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: BL602 不支持 HT40。

14.3.2 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]

14.3.3 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]

14.4 2.4g channel

channel idx = 1 - 13

- 命令: c[channel idx]

14.5 2.4g tx power

power dbm = 12 - 23dbm

- 命令: p[power dbm]

14.6 TX frame length

- 命令: l[length]

14.7 TX frequency

max value=1000

- 命令: `f[freq]`

14.8 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]`

14.9 HBN

enter into hbn mode

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

14.10 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]`

14.11 Get MFG FW version

- 命令: `y:v`
- 返回: `***version:[version]`

14.12 Get MFG FW building infomation

- 命令: `y:d`
- 返回: `***date:[building date] time:[building time]`

14.13 Get current power level

- 命令: `y:p`
- 返回: `***power:[power level dbm]`

14.14 Get current channel

- 命令: `y:c`
- 返回: `***channel:[channel freq]`

14.15 Get current tx status

- 命令: `y:t`
- 返回: `***tx:[0 or 1]`

14.16 Get tx frequency

- 命令: `y:f`
- 返回: `***freq:[tx frequency]`

14.17 Get cap code

- 命令: `y:x`
- 返回: `***capcode:[capcode value]`

14.18 Get MFG mode

- 命令: `y:M`
- 返回: `***mfgmode:[MFG mode]`

14.19 Set cap code

- 命令: `X[cap code]`

14.20 Set MFG Test(CW) mode

0 for normal mode, 1 for CW test mode

- 命令: `M[MFG mode]`

14.21 Save 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。

14.21.1 Write cap code to efuse buffer

- 命令: WEX[cap code]

14.21.2 Load cap code from efuse buffer

- 命令: LEX
- 返回: Cap code2:[cap code]

14.21.3 Program cap code to efuse

- 命令: SEX

14.21.4 Read cap code from efuse

- 命令: REX
- 返回: Cap code2:[cap code]

14.21.5 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

14.21.6 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]

14.21.7 Program power offset to efuse

- 命令: SEP

14.21.8 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]`

14.21.9 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`

14.21.10 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`

14.21.11 Program mac address to efuse

- 命令: `SEM`

14.21.12 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`

14.22 Save parameters to flash

14.22.1 save cap code to flash

- 命令: `SFX[cap code]`

14.22.2 read cap code from flash

- 命令: `RFX`
- 返回: `Cap code2:[cap code]`

14.22.3 save WiFi default power to flash

- 命令: `SFPD[mode] [channel] , [power]`

注解: mode 可以为 B or G or N

14.22.4 read WiFi default power from flash

- 命令: `RFPD[channel]`
- 返回: `Default power:[power]`

14.22.5 select WiFi default power in flash

- 命令: `P-1`

14.22.6 save BLE default power to flash

- 命令: `SFPE[channel] , [power]`

14.22.7 read BLE default power from flash

- 命令: `RFPE[channel]`
- 返回: `Default power:[power]`

14.22.8 select BLE default power in flash

- 命令: `EP-1`

14.23 Reset MFG FW

- 命令: `Reset`

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

14.24 Set tx duty

- 命令: `d[duty]`

注解: Duty value is between 0-100

14.25 Get tx duty

- 命令: `y:i`
- 返回: `###duty:[tx duty]`

14.26 BLE Test

14.26.1 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.

14.26.2 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.

14.26.3 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.

14.26.4 BLE test stop

- 命令: `EE`