



射频性能测试 使用手册

Version: 1.6

Copyright @ 2022

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/下载测试固件 | 10 |
| 4.1 BL702/704/706 IOT 模组 (带 Flash) 烧写测试固件 | 10 |
| 4.2 BL702L/704L IOT 模组 (带 Flash) 烧写测试固件 | 13 |
| 5 运行测试固件 | 17 |
| 6 基本设置 | 18 |
| 7 Single Tone 测试 | 23 |
| 8 802.15.4 测试 | 24 |
| 8.1 802.15.4 TX 测试 | 24 |
| 8.2 802.15.4 RX 测试 | 25 |
| 9 BLE 测试 | 26 |
| 10 HBN/PDS 测试 | 29 |
| 11 RF 测试固件在量产中的使用说明 | 30 |
| 12 串口通信命令 | 33 |
| 12.1 Shakehand | 33 |
| 12.2 Get cap code | 33 |
| 12.3 Set cap code | 33 |
| 12.4 Get power | 33 |
| 12.5 Set power | 34 |
| 12.6 Set channel | 34 |
| 12.7 Set TX frequency | 34 |
| 12.8 Set sequence number | 34 |
| 12.9 TX | 34 |
| 12.10 RX | 34 |
| 12.11 HBN | 35 |
| 12.12 PDS | 35 |
| 12.13 Power Offset | 35 |
| 12.14 Single Tone | 35 |
| 12.15 CCA | 35 |
| 12.16 Active | 36 |
| 12.17 Read memory | 36 |
| 12.18 Write memory | 36 |

| | |
|---|----|
| 12.19 Write cap code to efuse register | 36 |
| 12.20 Write power offset to efuse register (BL702) | 36 |
| 12.21 Write power offset to efuse register (BL702L) | 37 |
| 12.22 Write mac address to efuse register | 37 |
| 12.23 Read cap code from efuse register | 37 |
| 12.24 Read power offset from efuse register | 37 |
| 12.25 Read mac address from efuse register | 38 |
| 12.26 Program efuse | 38 |
| 12.27 Cap code temperature calibration (BL702L) | 38 |
| 12.28 Get MFG FW version | 38 |
| 12.29 Get MFG FW build infomation | 38 |
| 12.30 Get temperature | 39 |
| 12.31 Get MAC address | 39 |
| 12.32 Exit MFG FW | 39 |
| 12.33 Reset chip | 39 |
| 12.34 BLE Test | 39 |

| | |
|------------------------------------|----|
| 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 转接板 | 11 |
| 4.3 烧写界面 | 12 |
| 4.4 烧录成功界面 | 13 |
| 4.5 IOT 模组评估套件 | 14 |
| 4.6 烧写界面 | 15 |
| 4.7 烧录成功界面 | 16 |
| 5.1 程序成功运行的 log | 17 |
| 6.1 获取参数 | 18 |
| 6.2 设置发射功率 | 19 |
| 6.3 设置功率补偿 | 20 |
| 6.4 更新电容补偿值 | 21 |
| 6.5 设置 Active 模式 | 22 |
| 7.1 Single Tone 测试 | 23 |
| 8.1 802.15.4 TX 测试 | 24 |
| 8.2 802.15.4 RX 测试 | 25 |
| 9.1 BLE Tx Payload Type | 27 |
| 9.2 BLE Tx Rate | 27 |
| 9.3 BLE Rx Rate | 28 |
| 9.4 BLE Modulation Index | 28 |

| | |
|-----------------------------|----|
| 10.1 HBN/PDS 参数设置 | 29 |
| 11.1 修改活动分区 | 31 |
| 11.2 产测烧写界面 | 32 |

表 1.1: 版本记录

| 版本 | 更新内容 |
|------|--|
| V1.0 | 初始稿 |
| V1.1 | 增加 power offset 、 efuse 相关内容 |
| V1.2 | 增加产测固件烧录方法 |
| V1.3 | 增加 efuse 中 mac 地址的读写命令 |
| V1.4 | 更新图片和 802.15.4 RX 测试章节 |
| V1.5 | 增加 BL702L 相关内容 |
| V1.6 | 更新 BL702L 相关命令 |

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。

The screenshot displays the RF MFG web interface with the following sections:

- Navigation Tabs:** IOT, MCU, MFG (selected).
- BLE Section:**
 - PHY Channel: 0, Tx Rate: 1Mbps, Tx Start button.
 - Tx Data Length: 37, Rx Rate: 1Mbps, Rx Start button.
 - Tx Payload Type: PRBS9, Modulation Index: 1, Stop button.
- 802.15.4 Section:**
 - Channel: 11, Tx Start, Tx Stop buttons.
 - Seq Num: 1, Rx Start, Rx Stop buttons.
 - Tx Interval: 10 ms.
- Single Tone Section:**
 - Channel: 2402MHz, Tx Start, Tx Stop buttons.
- HBN/PDS Section:**
 - HBN Mode: HBN0, HBN Sleep: 0 ms, HBN Start button.
 - PDS Mode: PDS31, PDS Sleep: 0 ms, PDS Start button.
- User Command Section:**
 - Command input field and Send button.
- Basic Options Section (Right Sidebar):**
 - Port: COM3, Power: 14, Power Offset: Disable, Active: Default, CapCode: 36.
 - Buttons: Refresh, Clear, Open Uart, Misc Set, Misc Get, Chip Reset.

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:

- Single Tone 测试
- 802.15.4 报文发送
- 802.15.4 报文接收
- BLE 报文发送

- BLE 报文接收
- 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。工具包的内容如图所示。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 |
|----------------------|------------------|----------------|-----------|
| chips | 2022/12/1 14:26 | 文件夹 | |
| docs | 2022/12/1 14:27 | 文件夹 | |
| utils | 2022/12/1 14:27 | 文件夹 | |
| bflb_iot_tool.exe | 2022/12/1 14:10 | 应用程序 | 13,494 KB |
| bflb_iot_tool-macos | 2022/12/1 14:20 | 文件 | 15,214 KB |
| bflb_iot_tool-ubuntu | 2022/12/1 14:18 | 文件 | 15,490 KB |
| BLDevCube.exe | 2022/12/1 14:10 | 应用程序 | 36,475 KB |
| BLDevCube-macos | 2022/12/1 14:20 | 文件 | 42,602 KB |
| BLDevCube-macos-m1 | 2022/11/24 14:35 | 文件 | 29,973 KB |
| BLDevCube-ubuntu | 2022/12/1 14:18 | 文件 | 62,751 KB |
| clear.bat | 2022/11/17 16:19 | Windows 批处理... | 4 KB |
| ReleaseNote_bl.txt | 2022/12/1 14:09 | TXT 文件 | 21 KB |

图 3.1: 开发烧录软件工具包

4.1 BL702/704/706 IOT 模组 (带 Flash) 烧写测试固件

下面以 BL704 的 IOT 模组评估套件烧写为例，介绍烧写过程。

BL704 的 IOT 模组评估套件 BL704_IoT_DevKit 如下图所示。

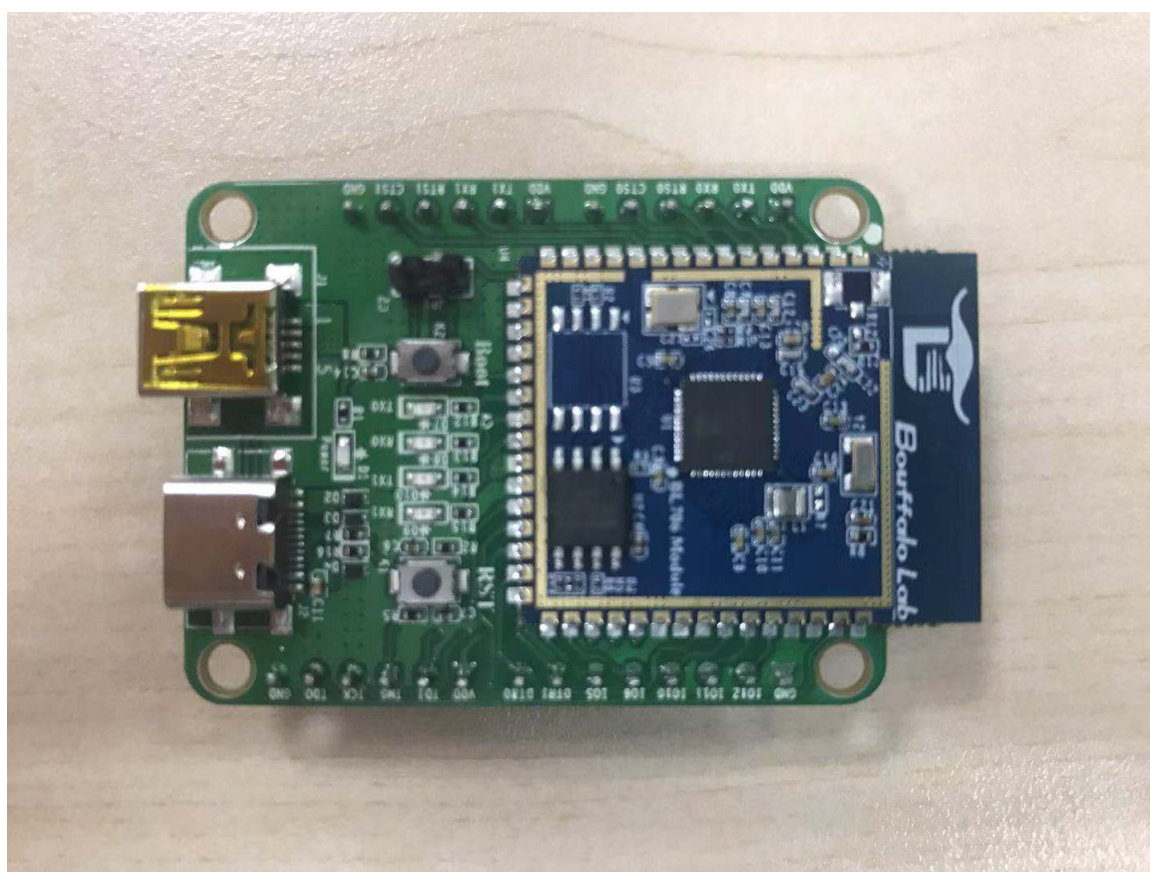


图 4.1: IOT 模组评估套件

套件由 IOT 模组和模组底板组成，模组底板使用 USB 接口供电。

另外需要一块 USB 串口转接板，下图是其中一种转接板。

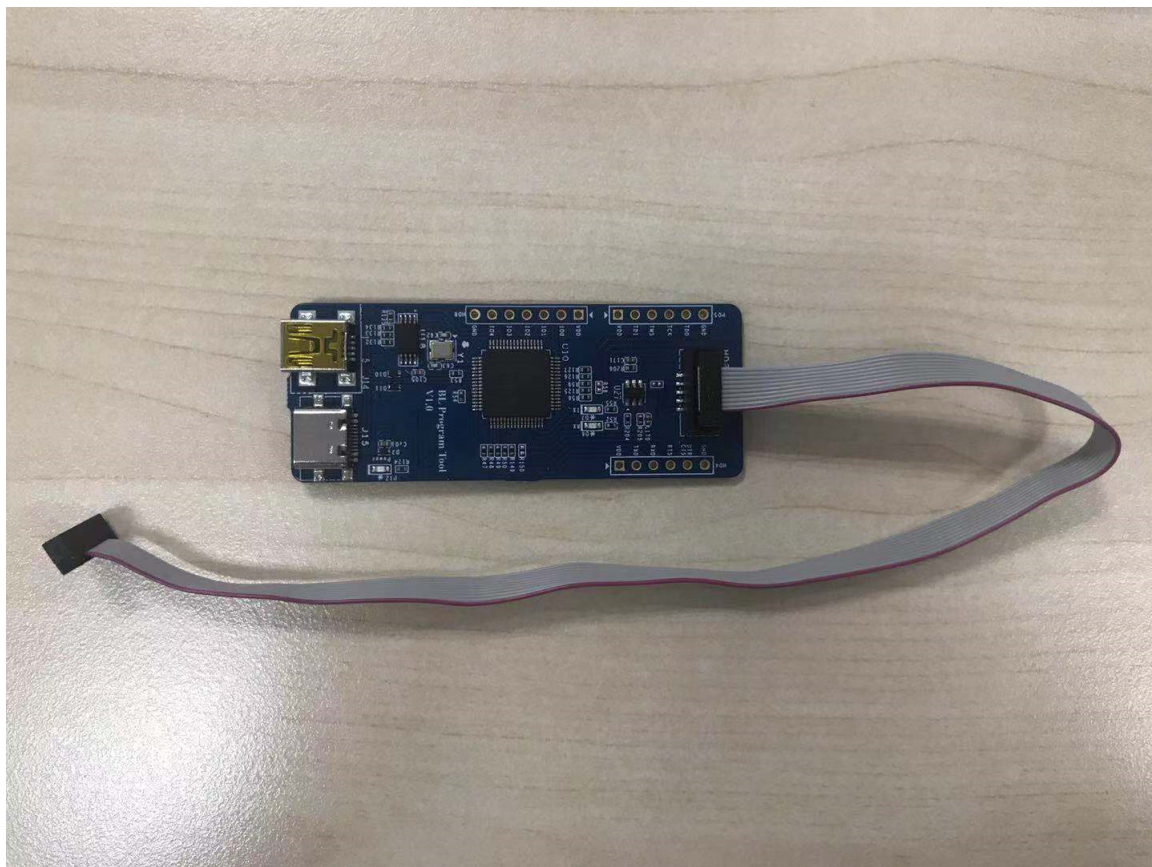


图 4.2: 转接板

转接板使用 USB 接口供电，连接模组后用于 UART 下载、UART 打印以及在线调试。

当转接板连接到 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。

在转接板连接模组后，首先运行 BLDevCube.exe，在 Chip Selection 中选择 BL702/704/706，进入如下烧写界面。

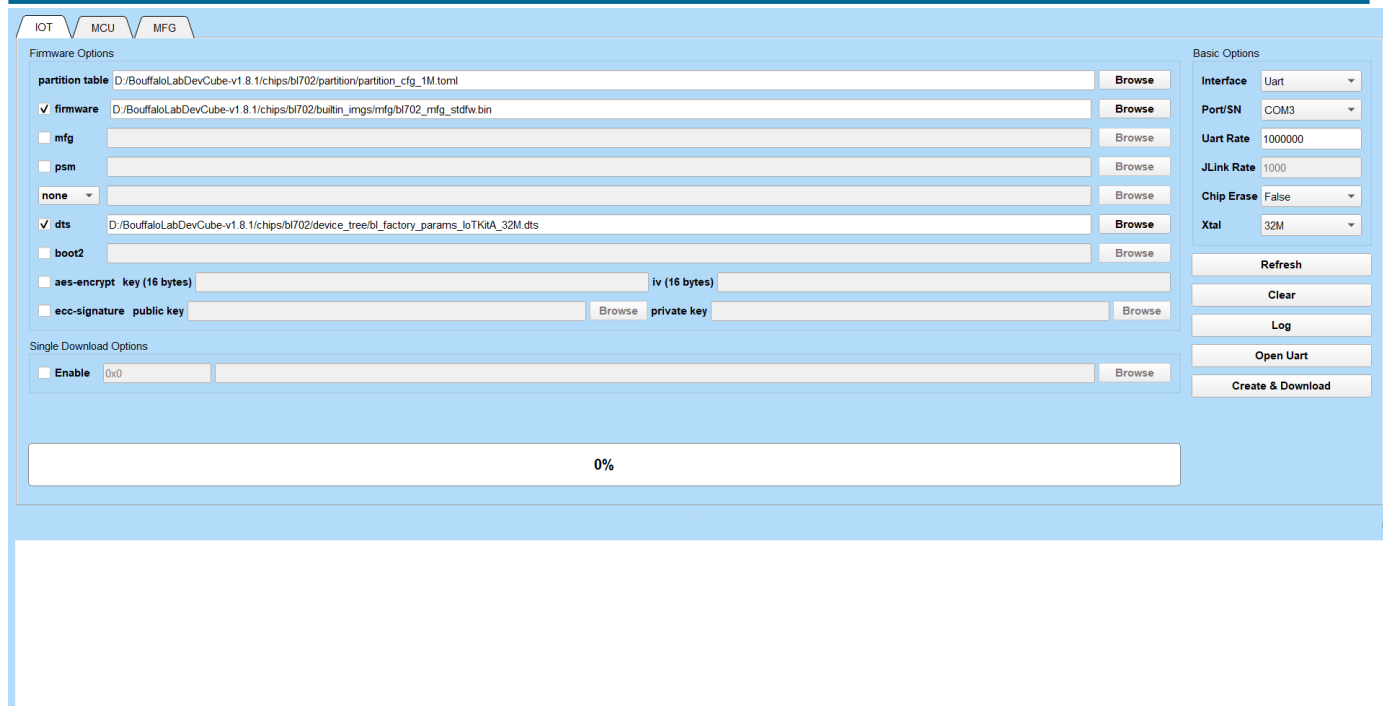


图 4.3: 烧写界面

在右侧通信接口设置中：

- **Interface:** 用于选择烧录的通信接口，这里选择 **Uart** 进行烧写
- **Port/SN:** 当选择 **Uart** 进行烧写的时候，这里选择与芯片连接的 **COM** 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，推荐使用 **1000000**
- **Chip Erase:** 选择下载前是否擦除整片 **Flash**，默认设置为不擦除
- **Xtal:** 用于选择板子所使用的晶振类型，这里选择 **32M**

其它项使用默认配置即可。

在左侧烧录镜像配置中，分别选择：

- **partition table:** 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 **bl702/partition/partition_cfg_1M.toml**
- **firmware:** 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 **Flash** 版本固件，本例中使用 **bl702/builtin_imgs/mfg/bl702_mfg_stdfw.bin**
- **dts:** 使用烧写工具目录下的对应芯片型号 **device_tree** 目录下的设备树，本例中使用 **bl702/device_tree/bl_factory_params_ioTKitA_32M.dts**

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 Boot 按键
- 按下然后松开 RST 按键
- 松开 Boot 按键

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。

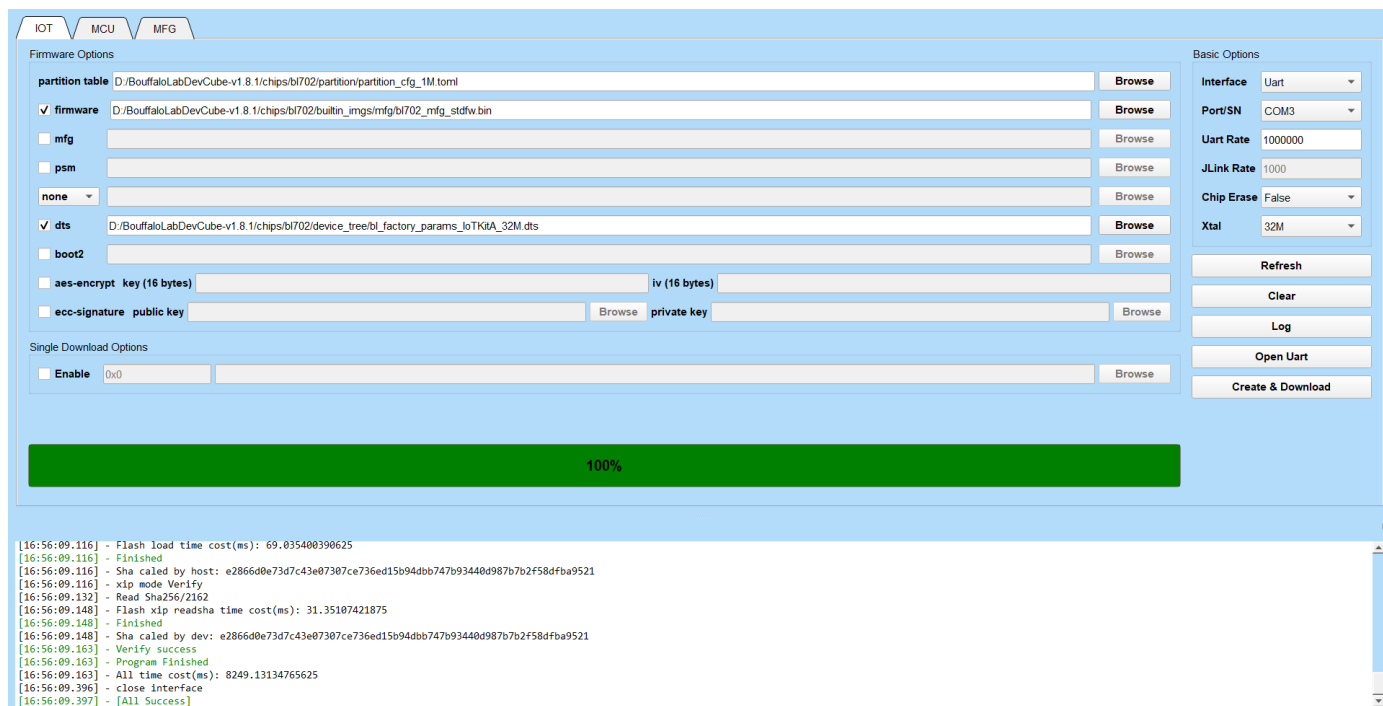


图 4.4: 烧录成功界面

4.2 BL702L/704L IOT 模组 (带 Flash) 烧写测试固件

下面以 BL702L 的 IOT 模组评估套件烧写为例，介绍烧写过程。

BL702L 的 IOT 模组评估套件 BL702L_DVK 如下图所示。

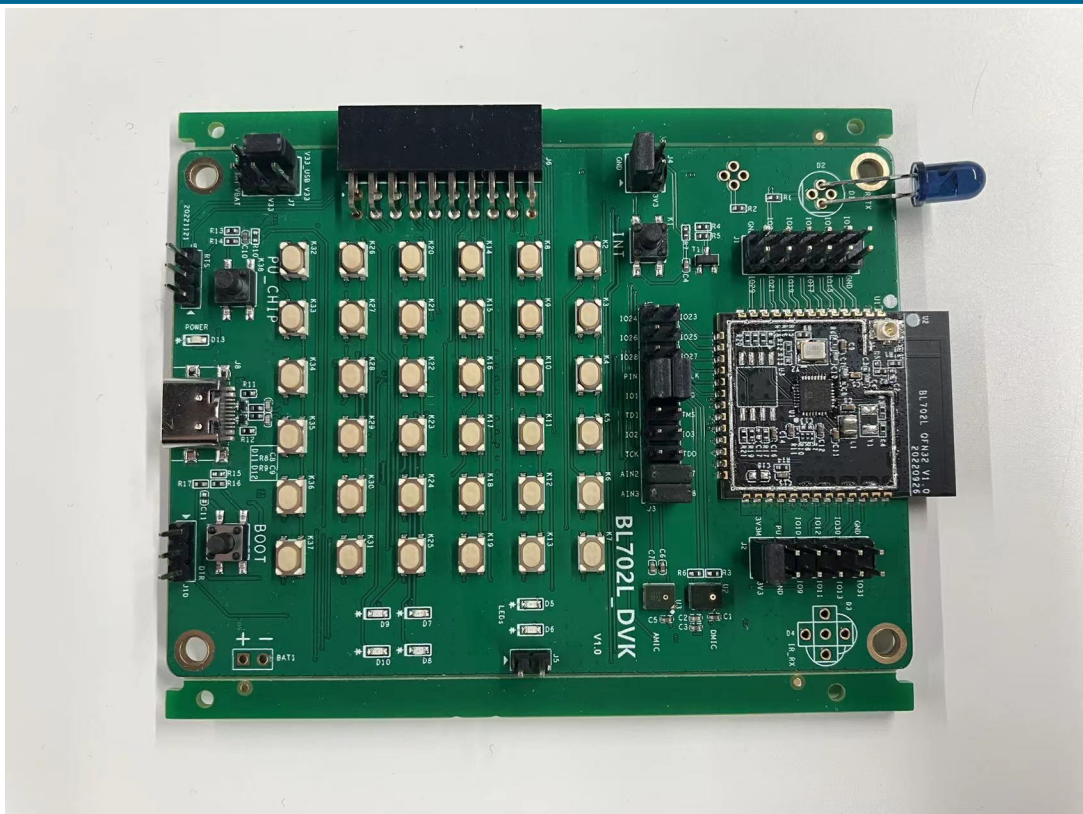


图 4.5: IOT 模组评估套件

套件由 IOT 模组和模组底板组成，模组底板使用 USB 接口供电。

当套件连接到 PC 后，会在 PC 的设备管理器出现一个 USB 转串口和一个 CKLink-Lite，用于 UART 下载、UART 打印以及在线调试。

运行 BLDevCube.exe，在 Chip Selection 中选择 BL702L/704L，进入如下烧写界面。

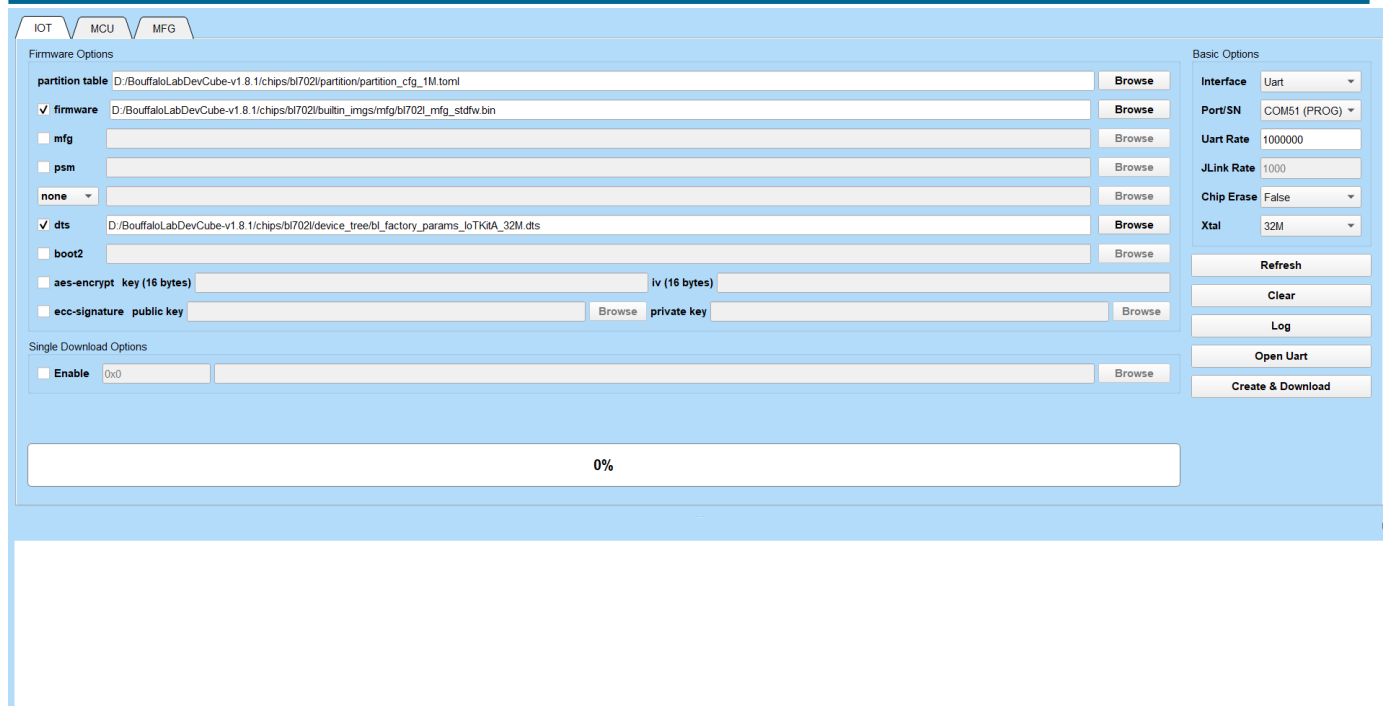


图 4.6: 烧写界面

在右侧通信接口设置中：

- **Interface:** 用于选择烧录的通信接口，这里选择 **Uart** 进行烧写
- **Port/SN:** 当选择 **Uart** 进行烧写的时候，这里选择与芯片连接的 **COM** 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，推荐使用 **1000000**
- **Chip Erase:** 选择下载前是否擦除整片 **Flash**，默认设置为不擦除
- **Xtal:** 用于选择板子所使用的晶振类型，这里选择 **32M**

其它项使用默认配置即可。

在左侧烧录镜像配置中，分别选择：

- **partition table:** 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 **bl702l/partition/partition_cfg_1M.toml**
- **firmware:** 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 **Flash** 版本固件，本例中使用 **bl702l/builtin_imgs/mfg/bl702l_mfg_std_fw.bin**
- **dts:** 使用烧写工具目录下的对应芯片型号 **device_tree** 目录下的设备树，本例中使用 **bl702l/device_tree/bl_factory_params_loTKitA_32M.dts**

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 BOOT 按键
- 按下然后松开 PU_CHIP 按键
- 松开 BOOT 按键

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。

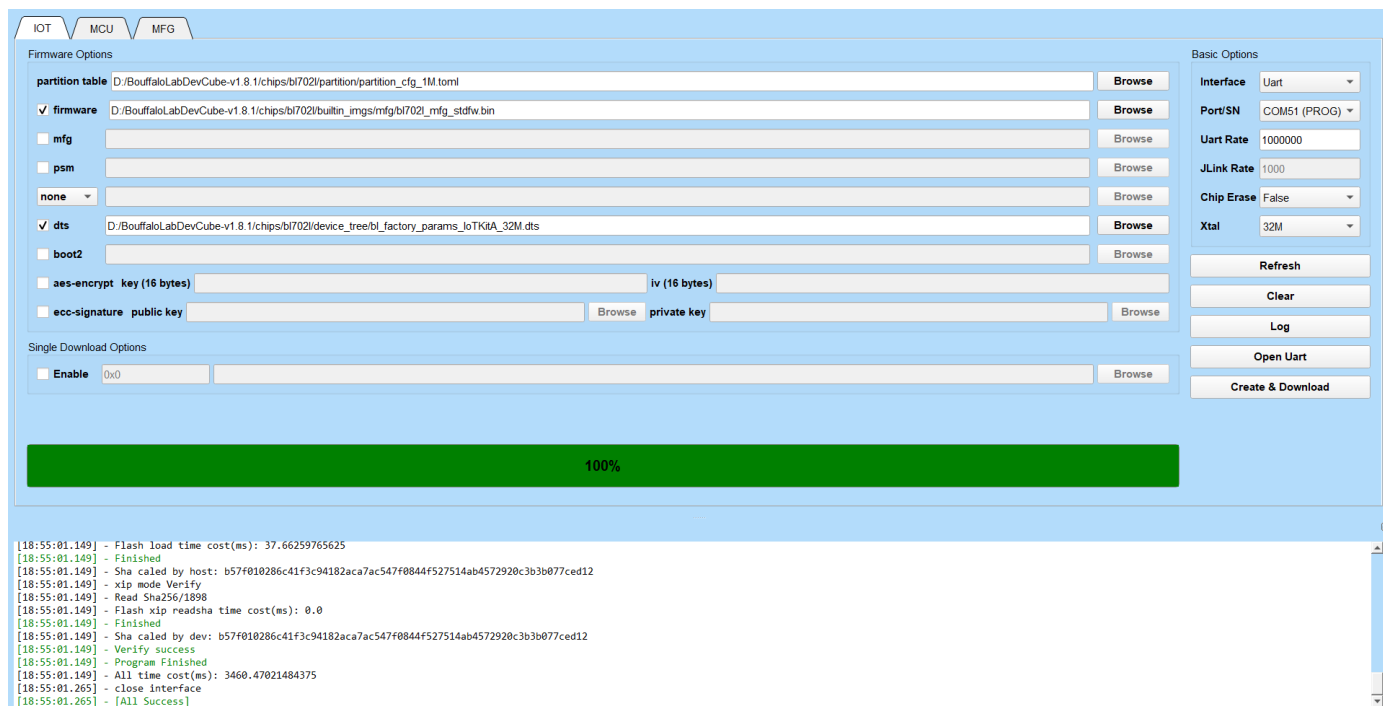
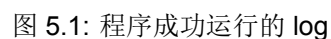


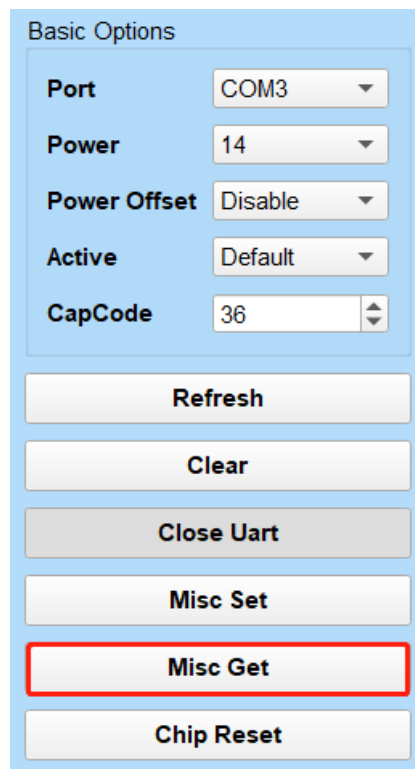
图 4.7: 烧录成功界面

在 BLDevCube.exe 界面，通过 MFG 选项卡进入 RF MFG 测试界面。选择使用到的 COM 号，点击 Open Uart 按钮，即可看到固件程序成功运行的 log，示例如下。



射频性能测试使用手册

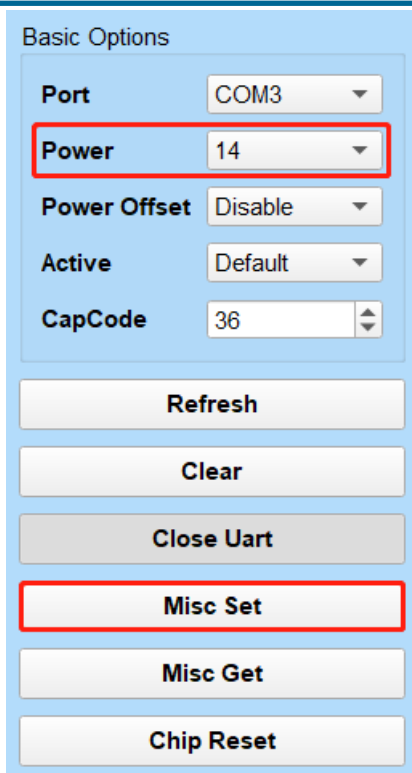
点击 **Misc Get** 按钮，获取芯片当前设置的 Power、CapCode 等参数。



The image shows a software window titled "Basic Options" with a light blue border. Inside, there are five configuration items, each with a label and a control element: "Port" with a dropdown menu showing "COM3", "Power" with a dropdown menu showing "14", "Power Offset" with a dropdown menu showing "Disable", "Active" with a dropdown menu showing "Default", and "CapCode" with a numeric spinner box showing "36". Below these items is a vertical stack of seven buttons: "Refresh", "Clear", "Close Uart", "Misc Set", "Misc Get", and "Chip Reset". The "Misc Get" button is highlighted with a red rectangular border.

图 6.1: 获取参数

通过 **Power** 下拉列表框，可以设置芯片的发射功率，点击 **Misc Set** 按钮生效。



Basic Options

| | |
|--------------|---------|
| Port | COM3 |
| Power | 14 |
| Power Offset | Disable |
| Active | Default |
| CapCode | 36 |

Refresh

Clear

Close Uart

Misc Set

Misc Get

Chip Reset

图 6.2: 设置发射功率

通过 Power Offset 下拉列表框, 可以使能或禁止功率补偿, 点击 Misc Set 按钮生效。在芯片复位后会初始化 power offset table, 每次使能功率补偿的时候会更新 power offset table。

Basic Options

Port

COM3

Power

14

Power Offset

Disable

Active

Default

CapCode

36

Refresh

Clear

Close Uart

Misc Set

Misc Get

Chip Reset

图 6.3: 设置功率补偿

针对晶体的负载电容，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。

注解：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: BL702 对应的电容补偿值

| XTAL Loading Capacity (pF) | Capacity Code |
|----------------------------|---------------|
| 12 | 32~36 |

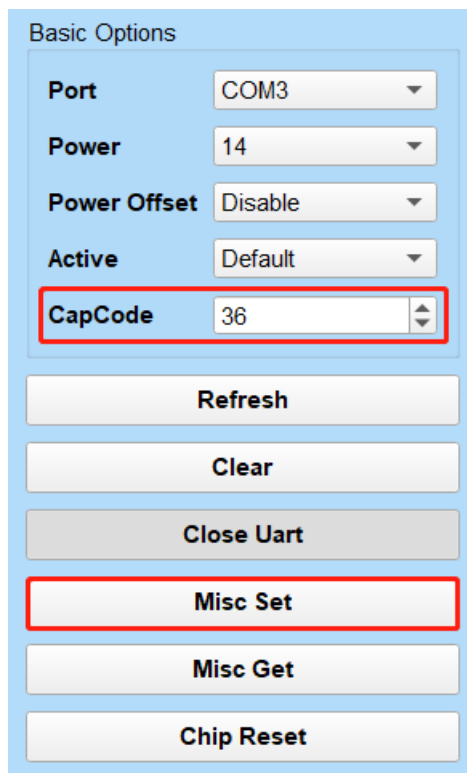
表 6.2: BL702L 对应的电容补偿值

| XTAL Loading Capacity (pF) | Capacity Code |
|----------------------------|---------------|
| 12 | 113~123 |

使用方法如下：

1. 在 CapCode 中填写需要补偿的值。

2. 点击 **Misc Set** 按钮更新补偿值。



The image shows a 'Basic Options' configuration window. It contains several dropdown menus: 'Port' set to 'COM3', 'Power' set to '14', 'Power Offset' set to 'Disable', and 'Active' set to 'Default'. The 'CapCode' field, which is a numeric spinner, is set to '36' and is highlighted with a red rectangle. Below these fields are several buttons: 'Refresh', 'Clear', 'Close Uart', 'Misc Set' (highlighted with a red rectangle), 'Misc Get', and 'Chip Reset'.

图 6.4: 更新电容补偿值

通过 **Active** 下拉列表框，可以设置芯片的 **Active** 模式，点击 **Misc Set** 按钮生效。其中 **Idle** 模式会关闭更多的时钟和外设，达到降低 **Active** 功耗的目的。

Basic Options

| | |
|--------------|---------|
| Port | COM3 |
| Power | 14 |
| Power Offset | Disable |
| Active | Default |
| CapCode | 36 |

Refresh

Clear

Close Uart

Misc Set

Misc Get

Chip Reset

图 6.5: 设置 Active 模式

Single Tone 测试

MFG 支持 Single Tone 测试模式，设置好 Channel 后，点击 Tx Start 开始测试，点击 Tx Stop 结束测试。

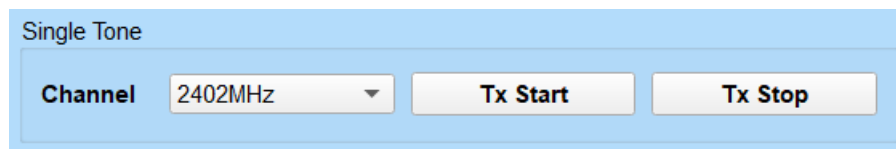


图 7.1: Single Tone 测试

802.15.4 测试

8.1 802.15.4 TX 测试

Channel 设置发送信道，Seq Num 设置报文的序列号，Tx Interval 设置报文发送间隔。

点击 Tx Start 开始测试，点击 Tx Stop 结束测试。测试结束后，会在 LOG 区显示发送了多少报文。

The screenshot shows the 802.15.4 TX Test interface. The 802.15.4 section is highlighted with a red box, showing Channel 11, Seq Num 1, and Tx Interval 10 ms. The Tx Start and Tx Stop buttons are also highlighted. The log area at the bottom shows a series of messages, with the final message '[mfg fw][zb] tx_frm_cnt:118' highlighted in red.

```

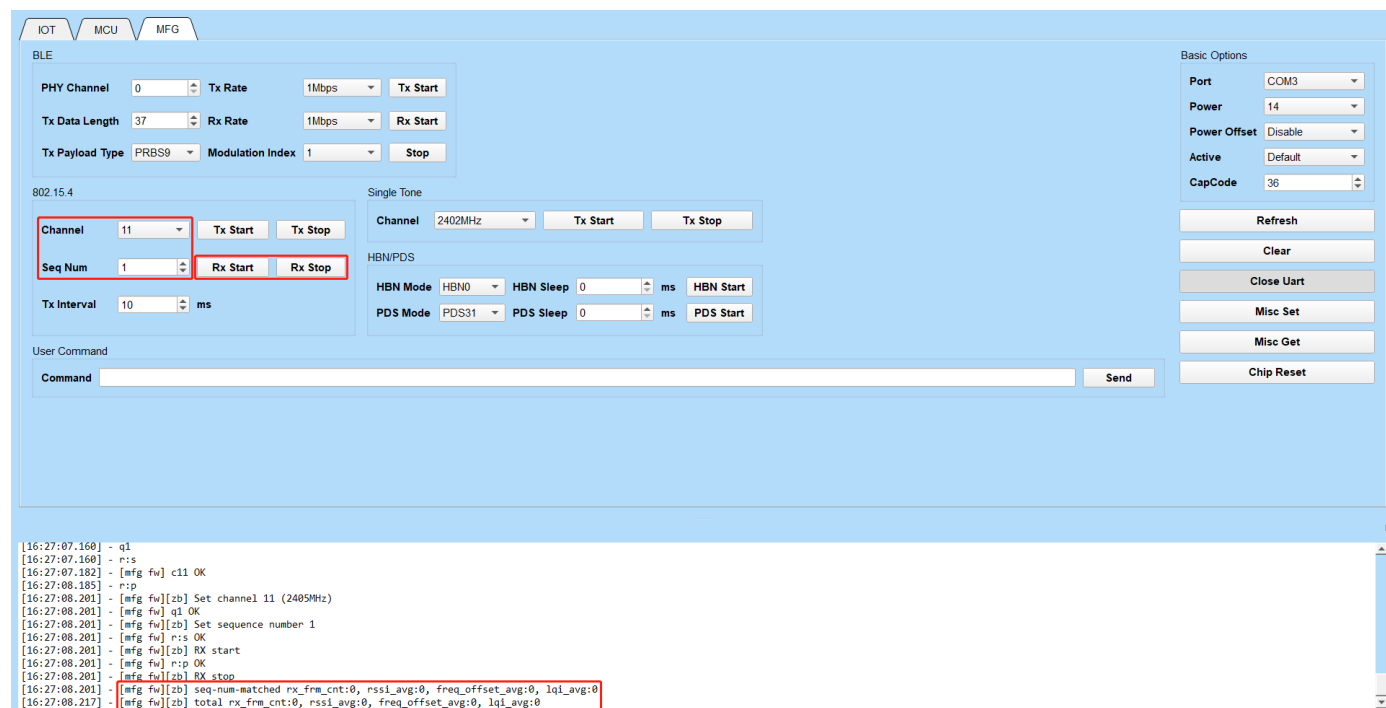
[16:30:19.413] - [mfg fw] c11 OK
[16:30:19.413] - [mfg fw][zb] Set channel 11 (2405MHz)
[16:30:19.413] - [mfg fw] q1 OK
[16:30:19.428] - [mfg fw][zb] Set sequence number 1
[16:30:19.428] - [mfg fw] f100 OK
[16:30:19.428] - [mfg fw][zb] TX 100 frames per second
[16:30:19.428] - [ 725708][INFO: hal_hwtimer.c: 182] get mux success
[16:30:19.428] - [mfg fw] t1 OK
[16:30:19.428] - [mfg fw][zb] TX toggle on
[16:30:20.515] - t0
[16:30:20.531] - [mfg fw] t0 OK
[16:30:20.531] - [mfg fw][zb] TX toggle off
[16:30:20.531] - [mfg fw][zb] tx_frm_cnt:118
  
```

图 8.1: 802.15.4 TX 测试

8.2 802.15.4 RX 测试

Channel 设置接收信道，Seq Num 设置报文的序列号。

点击 Rx Start 开始测试，点击 Rx Stop 结束测试。测试结束后，会在 LOG 区显示收到了多少报文 (序列号匹配的报文和总报文)，以及平均 rssi、平均 frequency offset 和平均 lqi。



BLE

PHY Channel 0 Tx Rate 1Mbps Tx Start

Tx Data Length 37 Rx Rate 1Mbps Rx Start

Tx Payload Type PRBS9 Modulation Index 1 Stop

802.15.4

Channel 11 Tx Start Tx Stop

Seq Num 1 Rx Start Rx Stop

Tx Interval 10 ms

Single Tone

Channel 2402MHz Tx Start Tx Stop

HBN/PDS

HBN Mode HBN0 HBN Sleep 0 ms HBN Start

PDS Mode PDS31 PDS Sleep 0 ms PDS Start

User Command

Command Send

Basic Options

Port COM3

Power 14

Power Offset Disable

Active Default

CapCode 36

Refresh

Clear

Close Uart

Misc Set

Misc Get

Chip Reset

```
[16:27:07.160] - q1
[16:27:07.160] - r:s
[16:27:07.182] - [mfg fw] c11 OK
[16:27:08.185] - r:p
[16:27:08.201] - [mfg fw][zb] Set channel 11 (2405MHz)
[16:27:08.201] - [mfg fw] q1 OK
[16:27:08.201] - [mfg fw][zb] Set sequence number 1
[16:27:08.201] - [mfg fw] r:s OK
[16:27:08.201] - [mfg fw][zb] RX start
[16:27:08.201] - [mfg fw] r:p OK
[16:27:08.201] - [mfg fw][zb] RX stop
[16:27:08.201] - [mfg fw][zb] seq-num-matched rx_frm_cnt:0, rssi_avg:0, freq_offset_avg:0, lqi_avg:0
[16:27:08.217] - [mfg fw][zb] total_rx_frm_cnt:0, rssi_avg:0, freq_offset_avg:0, lqi_avg:0
```

图 8.2: 802.15.4 RX 测试

RF MFG 提供 BLE 的 TX 和 RX 测试。

TX 测试可以设定测试的 PHY Channel, Tx Data Length, Tx Payload Type 和 Tx Rate, 然后点击 Tx Start 按钮开始测试。

RX 测试可以设定测试的 PHY Channel, Rx Rate 和 Modulation Index, 然后点击 Rx Start 按钮开始测试。

测试可以使用 Stop 按钮停止。

注解: MFG 固件支持标准 HCI 命令, 也可连接支持 HCI 命令的仪器进行测试 (如 R&S CMW500)。

Tx 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 Tx Payload Type

Tx Rate 如下图所示。

PHY:

Size: 1 octet

| Value | Parameter Description |
|------------------|--|
| 0x01 | Transmitter set to use the LE 1M PHY |
| 0x02 | Transmitter set to use the LE 2M PHY |
| 0x03 | Transmitter set to use the LE Coded PHY with S=8 data coding |
| 0x04 | Transmitter set to use the LE Coded PHY with S=2 data coding |
| All other values | Reserved for future use |

图 9.2: BLE Tx Rate

Rx Rate 如下图所示。

PHY:**Size: 1 octet**

| Value | Parameter Description |
|------------------|--------------------------------------|
| 0x01 | Receiver set to use the LE 1M PHY |
| 0x02 | Receiver set to use the LE 2M PHY |
| 0x03 | Receiver set to use the LE Coded PHY |
| All other values | Reserved for future use |

图 9.3: BLE Rx Rate

Modulation Index 如下图所示。

Modulation_Index:**Size: 1 octet**

| Value | Parameter Description |
|------------------|--|
| 0x00 | Assume transmitter will have a standard modulation index |
| 0x01 | Assume transmitter will have a stable modulation index |
| All other values | Reserved for future use |

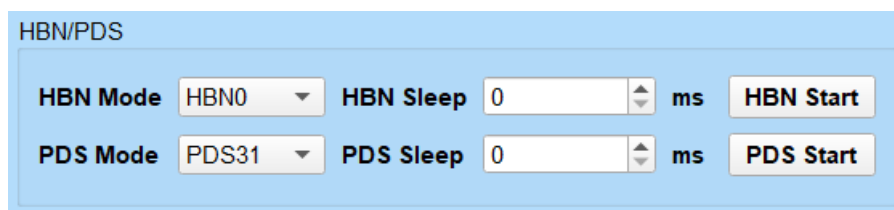
图 9.4: BLE Modulation Index

HBN/PDS 测试可以让芯片进入低功耗模式，其中 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/PDS 模式唤醒，唤醒后芯片会重新启动。目前测试工具仅仅支持 RTC 定时唤醒。

设置好 HBN/PDS 睡眠时间后，点击对应的 **Start** 按钮即可让芯片进入 HBN/PDS 模式，设定的睡眠时间到来后，芯片会重新启动。

如果睡眠时间设置为 0，代表永久睡眠。



| HBN/PDS | | | | | |
|----------|-------|-----------|---|----|-----------|
| HBN Mode | HBN0 | HBN Sleep | 0 | ms | HBN Start |
| PDS Mode | PDS31 | PDS Sleep | 0 | ms | PDS Start |

图 10.1: HBN/PDS 参数设置

RF 测试固件在量产中的使用说明

针对产测时需要同时烧写 RF 测试固件和用户应用程序的场景：在烧写完成后，芯片启动会进入 RF 测试固件，此时可以对接测试仪器（比如极致汇仪的仪器）完成产测；产测完成后，芯片再次启动会进入用户应用程序；若产测失败，芯片再次启动还是进入 RF 测试固件。

烧写步骤如下：

1. 选择一个适用于 IOT 下载界面的分区表文件，这里以 `partition_cfg_1M.toml` 为例，将 FW 分区的“activeindex”修改为 1，即从 FW1 启动，也即从 mfg 启动，因为 FW1 与 mfg 重合。

```
[pt_table]
#partition table is 4K in size
address0 = 0x1000
address1 = 0x2000

[[pt_entry]]
type = 0
name = "FW"
device = 0
address0 = 0x3000
size0 = 0x90000
address1 = 0x93000
size1 = 0x66000
# compressed image must set len,normal image can left it to 0
len = 0
activeindex = 0    0 -> 1
age = 0

[[pt_entry]]
type = 1
name = "mfg"
device = 0
address0 = 0x93000
size0 = 0x66000
address1 = 0
size1 = 0
# compressed image must set len,normal image can left it to 0
len = 0
activeindex = 0
age = 0
```

图 11.1: 修改活动分区

- 在 IOT 下载界面上, Partition Table 选择上一步修改好的分区表文件, Firmware Bin 选择用户应用程序, MFG Bin 选择 RF 测试固件, 其它保持不变。

IOTMCUMFG

Firmware Options

partition table

D:/BuffaloLabDevCube-v1.8.1/chips/bi702/partition/partition_cfg_1M.toml

Browse

firmware

D:/bi_iot_sdk/customer_app/bi702_demo_event/build_out/bi702_demo_event.bin

Browse

mfg

D:/BuffaloLabDevCube-v1.8.1/chips/bi702/builtin_imgs/mfg/bi702_mfg_stdfw.bin

Browse

psm

none

Browse

dts

D:/BuffaloLabDevCube-v1.8.1/chips/bi702/device_tree/bi_factory_params_ioTKiA_32M.dts

Browse

boot2

Browse

aes-encrypt

key (16 bytes)

iv (16 bytes)

Browse

ecc-signature

public key

Browse

private key

Browse

Basic Options

Interface

Uart

Port/SN

COM3

Uart Rate

1000000

JLink Rate

1000

Chip Erase

False

Xtal

32M

Refresh

Clear

Log

Open Uart

Create & Download

Single Download Options

Enable

0x0

Browse

0%

图 11.2: 产测烧写界面

退出产测方法:

1. 主机发送 **ATSC** 命令, 将用户应用程序分区设置为活动分区。如果用户应用程序未烧写, 则 **ATSC** 命令返回错误。
2. 主机发送 **Reset** 命令, 将芯片复位, 芯片复位后会进入用户应用程序。

所有命令均是字符串类型。

12.1 Shakehand

- 命令: H
- 返回: mfg

12.2 Get cap code

- 命令: x
- 返回: Get cap code [cap code]

12.3 Set cap code

- 命令: x[cap code]

cap code (BL702): [0, 63]; default: 36

cap code (BL702L): [0, 255]; default: 118

12.4 Get power

- 命令: p
- 返回: Get tx power [power dbm]dBm

12.5 Set power

- 命令: `p[power dbm]`

power dbm (BL702): [0, 14]; default: 14

power dbm (BL702L): [0, 10]; default: 10

12.6 Set channel

- 命令: `c[channel]`

对于 Single Tone 测试, channel 是信道频率 - channel: [2402, 2480]; default: 2402

对于 802.15.4 测试, channel 是信道索引号 - channel: [11, 26]; default: 11

12.7 Set TX frequency

- 命令: `f[freq]`

freq 是每秒发送的报文数量 - freq: [1, 1000]; default: 100

12.8 Set sequence number

- 命令: `q[seq num]`

seq num: [0, 255]; default: 1

12.9 TX

1. Tx Start: `t1`

2. Tx Stop: `t0`

- 返回: `tx_frm_cnt:[tx_frm_cnt]`

12.10 RX

1. Rx Start: `r:s`

2. Rx Stop: `r:p`

- 返回: `rx_frm_cnt:[rx_frm_cnt]`, `rss_avg:[rss_avg]`, `freq_offset_avg:[freq_offset_avg]`, `lqi_avg:[lqi_avg]`

12.11 HBN

1. set hbn level: hl[hbn level]

2. enter hbn mode: ht[sleep time ms]

hbn level: [0, 2]; default: 0

sleep time ms: [0, 131071999]; default: 0(sleep forever)

12.12 PDS

1. set pds level: sl[pds level]

2. enter pds mode: st[sleep time ms]

pds level: 31; default: 31

sleep time ms: [0, 131071999]; default: 0(sleep forever)

12.13 Power Offset

1. Enable: V1

2. Disable: V0

12.14 Single Tone

1. Tx Start: m1

2. Tx Stop: m0

12.15 CCA

1. set ed threshold: C:T[ed threshold]

2. run cca: C:M[cca mode]

• 返回: channel_busy:[channel_busy], rssi:[rssi], ed:[ed]

ed threshold: [-122, 5]; default: -71

cca mode: [0, 3]; default: 2

12.16 Active

1. Default: i0
2. Idle: i1

12.17 Read memory

- 命令: RM0x[addr]
- 返回: Read memory: 0x[addr] = 0x[val]

12.18 Write memory

- 命令: SM0x[addr]=0x[val]
- 返回: Write memory: 0x[addr] = 0x[val]

12.19 Write cap code to efuse register

- 命令: ewx[cap code]
- 返回: Write cap code [cap code] to efuse register

cap code (BL702): [0, 63]

cap code (BL702L): [0, 255]

注解: ewx 命令会先从 efuse 导出所有数据到对应的 efuse 寄存器, 再把 cap code 写入对应的 efuse 寄存器, 无写入次数限制。

12.20 Write power offset to efuse register (BL702)

- 命令: ewp[power offset low],[power offset high]
- 返回: Write power offset [power offset low],[power offset high] to efuse register

power offset low: [-8, 7]

power offset high: [-8, 7]

12.21 Write power offset to efuse register (BL702L)

- 命令: `ewp[power offset 1],[power offset 2],[power offset 3],[power offset 4]`
- 返回: Write power offset [power offset 1],[power offset 2],[power offset 3],[power offset 4] to efuse register

power offset 1: [-8, 7]

power offset 2: [-8, 7]

power offset 3: [-8, 7]

power offset 4: [-8, 7]

注解: `ewp` 命令会先从 `efuse` 导出所有数据到对应的 `efuse` 寄存器, 再把 `power offset` 写入对应的寄存器, 无写入次数限制。

12.22 Write mac address to efuse register

- 命令: `ewm[mac0]:[mac1]:[mac2]:[mac3]:[mac4]:[mac5]:[mac6]:[mac7]`
- 返回: Write mac [mac0]:[mac1]:[mac2]:[mac3]:[mac4]:[mac5]:[mac6]:[mac7] to efuse register

例子: `ewm00:00:75:09:00:42:E8:B4`

注解: `ewm` 命令会先从 `efuse` 导出所有数据到对应的 `efuse` 寄存器, 再把 `mac address` 写入对应的寄存器, 无写入次数限制。

12.23 Read cap code from efuse register

- 命令: `erx`
- 返回: Read cap code [cap code] from efuse register

12.24 Read power offset from efuse register

- 命令: `erp`
- 返回: Read power offset [power offset low],[power offset high] from efuse register

12.25 Read mac address from efuse register

- 命令: `erm`
- 返回: `Read mac [mac0]:[mac1]:[mac2]:[mac3]:[mac4]:[mac5]:[mac6]:[mac7] from efuse register`

12.26 Program efuse

- 命令: `ep`

注解: `ep` 命令会把 `efuse` 寄存器中的所有数据永久写入 `efuse`。即: `ewx+ep` 会把 `cap code` 写入 `efuse`, `ewp+ep` 会把 `power offset` 写入 `efuse`, `ewm+ep` 会把 `mac address` 写入 `efuse`。一颗芯片分别只有三次写 `cap code`、`power offset` 和 `mac address` 的机会。

12.27 Cap code temperature calibration (BL702L)

1. Enable: `X1`
2. Disable: `X0`

注解: 当使能的时候, 当前 `cap code` 寄存器的值会被保存, 并且 `efuse` 中 `cap code` 的值会被导入 `cap code` 寄存器。当禁止的时候, 之前保存的 `cap code` 寄存器的值会被恢复。

12.28 Get MFG FW version

- 命令: `y:v`
- 返回: `MFG Version: [version]`

12.29 Get MFG FW build infomation

- 命令: `y:d`
- 返回: `Build Data: [build date] Build Time: [build time]`

12.30 Get temperature

- 命令: `y:T`
- 返回: `Temperature: [temperature]`

12.31 Get MAC address

- 命令: `y:m`
- 返回: `MAC Address: [mac address]`

12.32 Exit MFG FW

- 命令: `ATSC`

注解: `ATSC` 命令将用户应用程序分区设置为活动分区, 下次芯片复位后会进入用户应用程序。如果用户应用程序未烧写, 则 `ATSC` 命令返回错误。

12.33 Reset chip

- 命令: `Reset`

12.34 BLE Test

12.34.1 BLE TX

- 命令: `ETE[channel][tx data length][tx payload type][tx rate]`

所有的参数都是 16 进制字符串。

举例: `ETE00250001`

channel: `0x00`

tx data length: `0x25`

tx payload type: `0x00`(PRBS9)

tx rate: `0x01`(1Mbps)

12.34.2 BLE RX

- 命令: ERE[channel][rx rate][modulation index]

所有的参数都是 16 进制字符串。

举例: ERE000100

channel: 0x00

rx rate: 0x01(1Mbps)

modulation index: 0x00

12.34.3 BLE test stop

- 命令: EE