



XR806 eFuse 工具 使用指南

版本号：1.0

发布时间：2021-01-21

版本历史

版本	日期	责任人	版本描述
1.0	2021-01-21	AWA 1029	创建文档。

目录

版本历史.....	i
目录.....	ii
图片目录.....	iv
表格目录.....	v
1 前言.....	1
1.1 文档简介.....	1
1.2 目标读者.....	1
1.3 适用范围.....	1
1.4 文档约定.....	1
1.4.1 标志说明.....	1
1.4.2 地址与数据描述方法约定.....	1
1.4.3 数值单位约定.....	2
2 概述.....	3
2.1 背景说明.....	3
2.2 规格特性.....	3
2.3 文件位置.....	3
3 技术说明.....	5
3.1 OEM 端 eFuse 烧写方案.....	5
4 应用说明.....	6
4.1 eFuse GUI 工具.....	6
4.1.1 应用简述.....	6
4.1.2 界面说明.....	6
4.2 eFuse CLI 工具.....	7
4.2.1 应用简述.....	7
4.2.2 指令说明.....	8
4.3 eFuse API.....	9
4.3.1 应用简介.....	9
4.3.2 配置说明.....	9
4.3.2.1 efuse_api.dll.....	9
4.3.2.2 efuse_api.lib.....	10
4.3.2.3 efuse_api.h.....	10
4.3.3 接口说明.....	10

4.3.3.1 EFUSE_GetMsg.....	10
4.3.3.2 EFUSE_InitEfuse.....	11
4.3.3.3 EFUSE_ExitEfuse.....	11
4.3.3.4 EFUSE_WriteHoscType.....	11
4.3.3.5 EFUSE_ReadHoscType.....	12
4.3.3.6 EFUSE_WriteScrBoot.....	12
4.3.3.7 EFUSE_ReadScrBoot.....	12
4.3.3.8 EFUSE_WriteDcxoTrim.....	13
4.3.3.9 EFUSE_ReadDcxoTrim.....	13
4.3.3.10 EFUSE_WritePoutCal.....	13
4.3.3.11 EFUSE_ReadPoutCal.....	13
4.3.3.12 EFUSE_WriteBtPoutCal.....	14
4.3.3.13 EFUSE_ReadBtPoutCal.....	14
4.3.3.14 EFUSE_WriteMacAddr.....	14
4.3.3.15 EFUSE_ReadMacAddr.....	15
4.3.3.16 EFUSE_WriteBtMacAddr.....	15
4.3.3.17 EFUSE_ReadBtMacAddr.....	15
4.3.3.18 EFUSE_WriteUserArea.....	16
4.3.3.19 EFUSE_ReadUserArea.....	16
4.3.3.20 EFUSE_WriteSecretKey.....	17
4.3.3.21 EFUSE_ReadSecretKey.....	17
4.3.3.22 EFUSE_WriteSecureSwd.....	17
4.3.3.23 EFUSE_ReadSecureSwd.....	17
5 示例说明.....	19
5.1 eFuse GUI 工具示例.....	19
5.1.1 示例简介.....	19
5.1.2 准备工作.....	19
5.1.3 操作步骤.....	19
5.1.4 效果展示.....	19
5.2 eFuse CLI 工具示例.....	20
5.2.1 示例简介.....	20
5.2.2 准备工作.....	20
5.2.3 操作步骤.....	21
5.2.4 效果展示.....	21
5.3 eFuse API 使用示例.....	21
5.3.1 示例简介.....	21
5.3.2 准备工作.....	21
5.3.3 代码示例.....	22

图片目录

图 3-1	OEM 端 eFuse 烧写方案结构示意图.....	5
图 4-1	eFuse GUI 工具界面.....	6
图 4-2	动态链接库 dll 文件.....	9
图 4-3	将 lib 文件放在工程中.....	10
图 5-1	读取 WLAN MAC 地址界面.....	20

表格目录

表 2-1	eFuse 工具功能特性.....	3
表 2-2	eFuse 工具文件说明.....	3
表 4-1	eFuse GUI 工具界面说明.....	7
表 4-2	eFuse CLI 工具指令说明.....	8
表 4-3	opt 和 param 说明.....	8
表 4-4	响应信息说明.....	9
表 4-5	EFUSE_GetMsg 接口函数说明.....	10
表 4-6	EFUSE_InitEfuse 接口函数说明.....	11
表 4-7	EFUSE_ExitEfuse 接口函数说明.....	11
表 4-8	EFUSE_WriteHoscType 接口函数说明.....	11
表 4-9	EFUSE_ReadHoscType 接口函数说明.....	12
表 4-10	EFUSE_WriteScrBoot 接口函数说明.....	12
表 4-11	EFUSE_ReadScrBoot 接口函数说明.....	12
表 4-12	EFUSE_WriteDcxoTrim 接口函数说明.....	13
表 4-13	EFUSE_ReadDcxoTrim 接口函数说明.....	13
表 4-14	EFUSE_WritePoutCal 接口函数说明.....	13
表 4-15	EFUSE_ReadPoutCal 接口函数说明.....	13
表 4-16	EFUSE_WriteBtPoutCal 接口函数说明.....	14
表 4-17	EFUSE_ReadBtPoutCal 接口函数说明.....	14
表 4-18	EFUSE_WriteMacAddr 接口函数说明.....	14
表 4-19	EFUSE_ReadMacAddr 接口函数说明.....	15
表 4-20	EFUSE_WriteBtMacAddr 接口函数说明.....	15
表 4-21	EFUSE_ReadBtMacAddr 接口函数说明.....	15
表 4-22	EFUSE_WriteUserArea 接口函数说明.....	16
表 4-23	EFUSE_ReadUserArea 接口函数说明.....	16
表 4-24	EFUSE_WriteSecretKey 接口函数说明.....	17
表 4-25	EFUSE_ReadSecretKey 接口函数说明.....	17
表 4-26	EFUSE_WriteSecureSwd 接口函数说明.....	17
表 4-27	EFUSE_ReadSecureSwd 接口函数说明.....	17

1 前言

1.1 文档简介

此文档介绍了 XR806 eFuse 工具的功能说明和使用示例。

1.2 目标读者

XR806 用户、SDK 使用人员。

1.3 适用范围

此文档适用于 XR806 SDK，支持 XR806 系列芯片产品。

1.4 文档约定

1.4.1 标志说明

本文档采用各种醒目的标志来表示在操作过程中应该特别注意的地方，这些标志的含义如下：

标识	说明
 警告	该标志后的说明应给予格外关注，如果不遵守，可能会导致人员受伤或死亡。
 注意	提醒操作中应注意的事项。不当的操作可能会损坏器件，影响可靠性、降低性能等。
 说明	为准确理解文中指令、正确实施操作而提供的补充或强调信息。
 窍门	一些容易忽视的小功能、技巧。了解这些功能或技巧能帮助解决特定问题或者节省操作时间。

1.4.2 地址与数据描述方法约定

本文档在描述地址、数据时遵循如下约定：

符号	例子	说明
0x	0x0200, 0x79	地址或数据以 16 进制表示。
0b	0b010, 0b00 000 111	数据采用二进制表示(寄存器描述除外)。
X	00X, XX1	数据描述中，x 代表 0 或 1。 例如，00X 代表 000 或 001；XX1 代表 001, 011, 101 或 111。

1.4.3 数值单位约定

本文档在描述数据容量（如 NAND 容量）时，单位词头代表的是 1024 的倍数；描述频率、数据速率等时则代表的是 1000 的倍数。具体如下：

类型	符号	对应数值
数据容量（如 NAND 容量）	1 K	1024
	1 M	1 048 576
	1 G	1 073 741 824
频率，数据速率等	1 k	1000
	1 M	1 000 000
	1 G	1 000 000 000

2 概述

2.1 背景说明

eFuse 工具是 PC 端提供的一组工具，通过与 SDK 端的 EFPG 模块通信，对芯片上的 eFuse 的 OEM 区域进行烧写和读取。OEM 端烧写的 eFuse 区域主要有：HOSC（HOSC TYPE）、BOOT（SECURE BOOT）、DCXO（DCXO TRIM）、POUT（POUT CAL）、MAC、User Area（用户区域）、Secure SWD（安全相关功能使用区域）以及 Secret Key（Flash 加密使用的哈希密钥）。

eFuse 工具提供有 GUI 版本，CLI 版本以及供用户二次开发的库文件。GUI 版本适合研发人员进行调试使用，CLI 和库文件适合产测阶段集成产测工具或者自动化测试工具时使用。

2.2 规格特性

eFuse 工具支持如下功能特性：

表 2-1 eFuse 工具功能特性

类别	功能支持	描述	备注
读写区域	支持读取 Chip ID	每个芯片的 Chip ID	
	支持烧写、读取 HOSC	芯片选择的高频晶振频率	
	支持烧写、读取 BOOT	Secure boot 的哈希值	
	支持烧写、读取 DCXO	系统频偏值	
	支持烧写、读取 POUT	WLAN 和 BT 的功率校准值	
	支持烧写、读取 MAC	WLAN 和 BT 的 MAC 地址	
	支持烧写、读取 User Area	用户自定义区域	
	支持烧写、读取 Secure SWD	安全相关功能使用区域	
	支持烧写、读取 Secret Key	Flash 加密使用的哈希密钥	
工具平台	Windows GUI 版本		
	Windows CLI 版本		
	Windows 二次开发库		

2.3 文件位置

eFuse 工具包含以下文件。

表 2-2 eFuse 工具文件说明

文件名	文件说明
efuse_api.dll	动态链接库文件

文件名	文件说明
efuse_api.h	eFuse 动态链接库的头文件
efuse_api.lib	eFuse 动态链接库的导入库文件
efuse_cli_tool.exe	eFuse CLI 工具可执行文件
efuse_tool.exe	eFuse GUI 工具可执行文件



说明

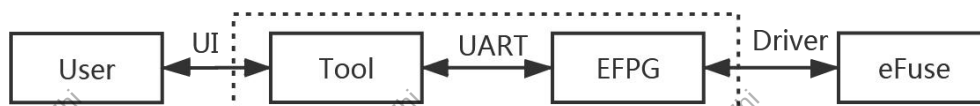
eFuse 工具包可在芯之联文档中心获取：<https://docs.xradiotech.com/>。

3 技术说明

3.1 OEM 端 eFuse 烧写方案

使用 OEM 端 eFuse 烧写工具可以实现对 HOSC、BOOT、DCXO、POUT 和 MAC 等参数的烧写，该烧写方案的整体结构如下图所示。

图 3-1 OEM 端 eFuse 烧写方案结构示意图



OEM 端 eFuse 烧写工具通过 UART 与 EFPG 模块通信，传输烧写区域和数据等信息。EFPG 模块对工具发送的信息校验通过后，将数据烧写到 eFuse 相应的区域。

EFPG 模块的相关实现和使用可参考《XR806_EFPG 中间件_开发指南》文档。为防止 eFuse 相应区域被任意烧写，烧写工具中需输入与代码中相同的 Key，其最大长度为 64byte，默认值为"efpgtest"。

4 应用说明

4.1 eFuse GUI 工具

4.1.1 应用简述

eFuse GUI 工具的使用很简单，其基本的应用步骤如下：

1. 双击 efuse_tool.exe 打开 eFuse GUI 工具；
2. 点击选择串口号和芯片型号；
3. 点击勾选需要操作的 eFuse 区域，若为写操作，则在对应区域填入需要写入的数据；
4. 点击“读取”按钮读取 eFuse 区域的数据，或者点击“烧写”按钮将数据写入 eFuse 区域；

4.1.2 界面说明

eFuse GUI 工具软件的界面如下所示。

图 4-1 eFuse GUI 工具界面

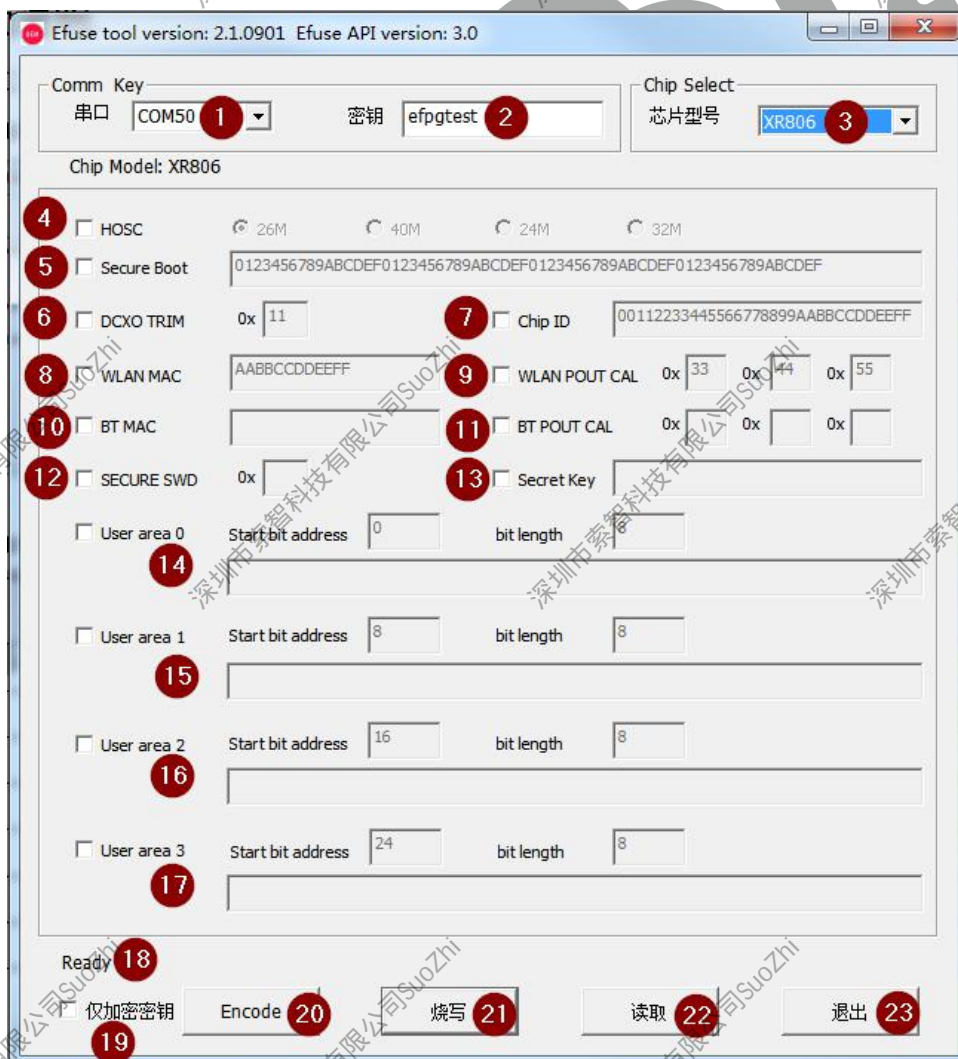


表 4-1 eFuse GUI 工具界面说明

序号	说明
1	串口选择
2	通信使用的密钥，需要与 SDK 端匹配，默认为“efpgtest”
3	芯片型号选择
4	HOSC 区域读写
5	Secure boot 区域读写
6	DCXO TRIM 区域读写，即 SDK 系统使用频偏值
7	Chip ID 区域读写
8	WLAN MAC 地址区域读写
9	WLAN POUT CAL 区域读写，即 WLAN 功率校准值
10	BT MAC 地址区域读写
11	BT POUT CAL 区域读写，即 BT 功率校准值
12	SECURE SWD 区域读写
13	Secret Key 区域读写
14	User area 0 区域读写
15	User area 1 区域读写
16	User area 2 区域读写
17	User area 3 区域读写
18	状态栏显示
19	仅加密密钥选项，主要用于密钥调试，用户无需使用
20	“Encode”按钮，主要用于密钥调试，用户无需使用
21	“烧写”按钮，将指定数据写入对应的 eFuse 区域
22	“读取”按钮，读取指定的 eFuse 区域数据
23	“退出”按钮，退出软件



说明

4 个 User area 区域并无区别，为了方便用户一次性读写多个 User area 区域，软件提供多个项目给用户填写。

4.2 eFuse CLI 工具

4.2.1 应用简述

eFuse CLI 工具提供一组参数，通过命令行运行可以读取或者写入指定 eFuse 区域的数据。

4.2.2 指令说明

表 4-2 eFuse CLI 工具指令说明

信息项	说明
原型	efuse_cli_tool.exe <opt> <param>
功能	启动 eFuse 指定区域的读取或写入数据
参数	详细参见表 4-3
响应	详细参见表 4-4
示例	以读取 WLAN MAC 地址区域为例，串口号为 COM3，通信密钥为“efpgtest” efuse_cli_tool.exe -c 3 -k efpgtest -r mac
备注	N/A

表 4-3 opt 和 param 说明

Opt 和 Param 取值	说明
-a <ua address>	User Area 读写起始地址，单位 bit
-c <com number>	串口号
-d <dcxo>	频偏值
-k <hash key>	哈希密钥，默认为“efpgtest”
-l <ua length>	User Area 长度，单位 bit
-m <mac address>	MAC 地址，WLAN 和 BT 的 MAC 地址区分如下： 1. 当-w 的参数取值为 mac 时为 WLAN 的 MAC 地址； 2. 当-w 的参数取值为 bt_mac 时为 BT 的 MAC 地址；
-o <hosc>	高频晶振频率，取值：0 - 26M 1 - 40M 2 - 24M 3 - 32M -1 - invalid
-p <pout cal>	功率校准值，[0:5]bit 数据，[6]bit 符号位，范围-64~63，三个 byte 分别为 ch_1/ch_7/ch_13，WLAN 和 BT 的功率校准区分如下： 1. 当-w 的参数取值为 pout 时为 WLAN 的功率校准值； 2. 当-w 的参数取值为 bt_pout 时为 BT 的功率校准值；
-r <eFuse area>	需要读取的 eFuse 区域，取值为{dcxo mac bt_mac hosc scr_boot pout bt_pout user_area chip_id scr_swd sct_key}
-s <scr boot>	Secure Boot 需要写入的数据
-u <ua data>	User Area 需要写入的数据
-w <eFuse area>	需要写入的 eFuse 区域，取值为{dcxo mac bt_mac hosc scr_boot pout bt_pout user_area scr_swd sct_key}
-x <scr swd>	Secure SWD 需要写入的数据
-y <sct key>	Secret Key 需要写入的数据
-h	帮助信息
-v	版本查看

表 4-4 响应信息说明

响应	说明
Send cmd error!	串口发送命令出错
Read ack1 error!	串口读取命令 ack 出错
Send data error!	串口发送数据出错
Read ack2 error!	串口读取数据 ack 出错
Operation OK!	读写操作成功
Host: Ack CheckSum error!	从机返回的 ack 的校验和出错
device 400: CheckSum error!	主机发送的命令或者数据的校验和出错
device 401: Message digest error!	主机的命令解密出错
device 402: Invalid data!	eFuse 数据不正确
device 403: Operation error, not dirty chip yet!	操作异常，还未成为脏片
device 404: Operation error, is dirty chip now!	操作异常，已成为脏片
device 405: Operation error, no valid data!	操作异常，没有可用数据
device : Unknown error!	未知错误

4.3 eFuse API

4.3.1 应用简介

efuse_api.dll 封装了一系列 eFuse 操作的接口函数，在 efuse_api.h 头文件中可以查看对应的 API 介绍。使用该动态链接库可以方便用户集成自己的生产测试工具。

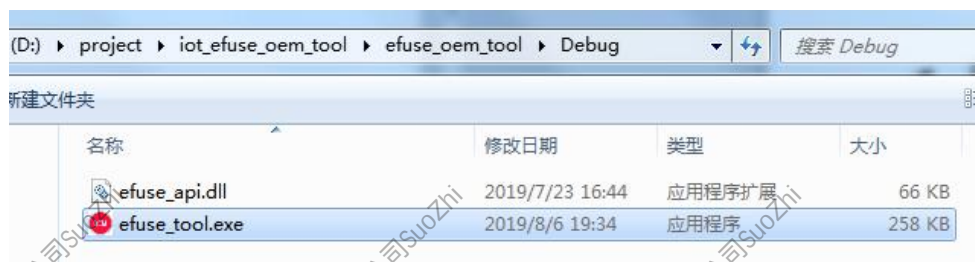
4.3.2 配置说明

下面以示例工程为例，简单介绍 efuse_api 的配置方法，总共有 efuse_api.dll、efuse_api.lib、efuse_api.h 三个文件需要配置。

4.3.2.1 efuse_api.dll

此文件是 API 的动态链接库，只需要将它与调用它的 exe 应用程序放在同一个目录下即可。

图 4-2 动态链接库 dll 文件



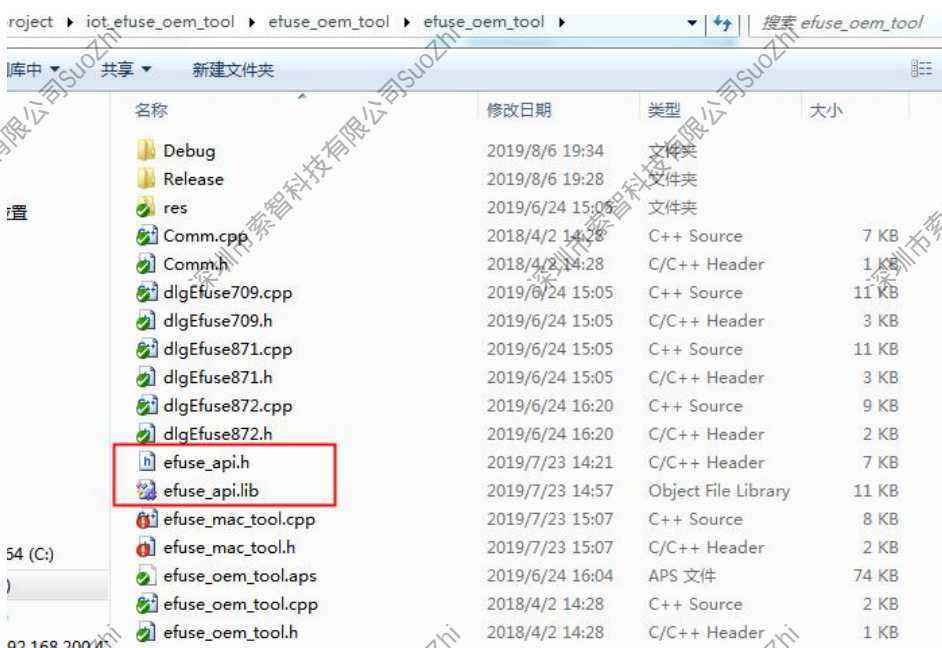
4.3.2.2 efuse_api.lib

此文件是 efuse_api.dll 动态链接库的导入库，并非静态链接库文件。需要将此文件放在工程目录下，并且在使用到 API 的文件中添加下面语句导入库文件。

```
#include "afxwin.h"
#include "efuse_api.h"
#include "Comm.h"
#include "afxcmn.h"
#pragma comment(lib,"efuse_api.lib")
```

实际文件放置如图所示：

图 4-3 将 lib 文件放在工程中



4.3.2.3 efuse_api.h

此文件是 API 的头文件，包含对 API 的一些简单介绍。需要将此文件放在工程目录下，并在使用到 API 的文件中包含此文件即可。如图 4-3 所示。

4.3.3 接口说明

4.3.3.1 EFUSE_GetMsg

表 4-5 EFUSE_GetMsg 接口函数说明

信息项	说明
原型	char* EFUSE_GetMsg()
功能	获取最后一次命令执行的状态信息。
参数	无
返回值	指向状态信息字符串的指针

4.3.3.2 EFUSE_InitEfuse

表 4-6 EFUSE_InitEfuse 接口函数说明

信息项	说明
原型	int EFUSE_InitEfuse(int comNum, char* keyBuf, int len)
功能	初始化 EFUSE，包括串口和密钥数据，进入 EFUSE 模式。
参数	int comNum 含义解释：串口号 char* keyBuf 含义解释：哈希字符串的指针 int len 含义解释：哈希字符串的长度，最大为 256
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.3 EFUSE_ExitEfuse

表 4-7 EFUSE_ExitEfuse 接口函数说明

信息项	说明
原型	int EFUSE_ExitEfuse()
功能	退出 EFUSE 模式，关闭串口。
参数	无
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.4 EFUSE_WriteHoscType

表 4-8 EFUSE_WriteHoscType 接口函数说明

信息项	说明
原型	int EFUSE_WriteHoscType(int hosc)
功能	写入 HOSC type。
参数	int hosc 含义解释：需要写入的 hosc 值 使用说明：0 - 26M 1 - 40M 2 - 24M 3 - 32M/52M -1 - invalid。
返回值	状态值，0 - 成功，非 0 - 失败



说明

XR806 系列芯片支持 24M、26M、40M 和 32M 高频晶振类型，不支持 52M 晶振类型；
XR871/XR809/XR872/XR808 系列芯片支持 24M、26M、40M 和 52M 高频晶振类型，不支持 32M 晶振类型；

efuse 工具支持以上所有系列芯片，并且 32M 和 52M 在每种芯片中对应的 hosc 值均定义为 3。

4.3.3.5 EFUSE_ReadHoscType

表 4-9 EFUSE_ReadHoscType 接口函数说明

信息项	说明
原型	int EFUSE_ReadHoscType(int* hosc)
功能	读取 HOSC type。
参数	int * hosc 含义解释：需要读取的 hosc 值 使用说明：0 - 26M 1 - 40M 2 - 24M 3 - 32M/52M -1 - invalid。
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.6 EFUSE_WriteScrBoot

表 4-10 EFUSE_WriteScrBoot 接口函数说明

信息项	说明
原型	int EFUSE_WriteScrBoot(char* hash, int len)
功能	写入 secure boot 的哈希值。
参数	char* hash 含义解释：用来存储哈希字符串的数组，只有 32byte int len 含义解释：哈希字符串的长度，只能为 32
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.7 EFUSE_ReadScrBoot

表 4-11 EFUSE_ReadScrBoot 接口函数说明

信息项	说明
原型	int EFUSE_ReadScrBoot(char* hash, int len)
功能	读取 secure boot 的哈希值。
参数	char* hash 含义解释：用来存储哈希字符串的数组，只有 32byte int len 含义解释：哈希字符串的长度，只能为 32
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.8 EFUSE_WriteDcxoTrim

表 4-12 EFUSE_WriteDcxoTrim 接口函数说明

信息项	说明
原型	int EFUSE_WriteDcxoTrim(char value)
功能	写入 DCXO TRIM 值。
参数	char value 含义解释：需要写入的 DCXO TRIM 值
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.9 EFUSE_ReadDcxoTrim

表 4-13 EFUSE_ReadDcxoTrim 接口函数说明

信息项	说明
原型	int EFUSE_ReadDcxoTrim(char* value)
功能	读取 DCXO TRIM 值。
参数	char* value 含义解释：存放 DCXO TRIM 值的指针
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.10 EFUSE_WritePoutCal

表 4-14 EFUSE_WritePoutCal 接口函数说明

信息项	说明
原型	int EFUSE_WritePoutCal(char* rfCal, int len)
功能	写入 WLAN POUT CAL 值。
参数	char* rfCal 含义解释：POUT CAL 值的数组，只有 3byte，每 byte 仅低 7bit 有效 int len 含义解释：POUT CAL 值数组的长度，只能为 3
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.11 EFUSE_ReadPoutCal

表 4-15 EFUSE_ReadPoutCal 接口函数说明

信息项	说明
原型	int EFUSE_ReadPoutCal(char* rfCal, int len)
功能	读取 WLAN POUT CAL 值。

信息项	说明
参数	<code>char* rfCal</code> 含义解释：WLAN POUT CAL 值的数组 <code>int len</code> 含义解释：WLAN POUT CAL 值数组的长度，只能为 3
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.12 EFUSE_WriteBtPoutCal

表 4-16 EFUSE_WriteBtPoutCal 接口函数说明

信息项	说明
原型	<code>int EFUSE_WriteBtPoutCal(char* rfCal, int len)</code>
功能	写入 BT POUT CAL 值。
参数	<code>char* rfCal</code> 含义解释：BT POUT CAL 值的数组，只有 3byte，每 byte 仅低 7bit 有效 <code>int len</code> 含义解释：BT POUT CAL 值数组的长度，只能为 3
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.13 EFUSE_ReadBtPoutCal

表 4-17 EFUSE_ReadBtPoutCal 接口函数说明

信息项	说明
原型	<code>int EFUSE_ReadBtPoutCal(char* rfCal, int len)</code>
功能	读取 BT POUT CAL 值。
参数	<code>char* rfCal</code> 含义解释：BT POUT CAL 值的数组 <code>int len</code> 含义解释：BT POUT CAL 值数组的长度，只能为 3
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.14 EFUSE_WriteMacAddr

表 4-18 EFUSE_WriteMacAddr 接口函数说明

信息项	说明
原型	<code>int EFUSE_WriteMacAddr(char* mac, int len)</code>
功能	写入 WLAN MAC 地址。

信息项	说明
参数	char* mac 含义解释：WLAN MAC 地址数组的指针，只有 6byte int len 含义解释：WLAN MAC 地址数组的长度，只能为 6
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.15 EFUSE_ReadMacAddr

表 4-19 EFUSE_ReadMacAddr 接口函数说明

信息项	说明
原型	int EFUSE_ReadMacAddr(char* mac, int len)
功能	读取 WLAN MAC 地址。
参数	char* mac 含义解释：WLAN MAC 地址数组的指针，只有 6byte int len 含义解释：WLAN MAC 地址数组的长度，只能为 6
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.16 EFUSE_WriteBtMacAddr

表 4-20 EFUSE_WriteBtMacAddr 接口函数说明

信息项	说明
原型	int EFUSE_WriteBtMacAddr(char* mac, int len)
功能	写入 BT MAC 地址。
参数	char* mac 含义解释：BT MAC 地址数组的指针，只有 6byte int len 含义解释：BT MAC 地址数组的长度，只能为 6
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.17 EFUSE_ReadBtMacAddr

表 4-21 EFUSE_ReadBtMacAddr 接口函数说明

信息项	说明
原型	int EFUSE_ReadBtMacAddr(char* mac, int len)
功能	读取 BT MAC 地址。

信息项	说明
参数	<code>char* mac</code> 含义解释：BT WLAN MAC 地址数组的指针，只有 6byte <code>int len</code> 含义解释：BT MAC 地址数组的长度，只能为 6
返回值	状态值，0- 成功，非 0- 失败

4.3.3.18 EFUSE_WriteUserArea

表 4-22 EFUSE_WriteUserArea 接口函数说明

信息项	说明
原型	<code>int EFUSE_WriteUserArea(char* data, int startAddr, int len)</code>
功能	从指定的位地址开始写入用户区数据。
参数	<code>char* data</code> 含义解释：存储用户区数据的数组 <code>int startAddr</code> 含义解释：起始的位地址，单位为 bit <code>int len</code> 含义解释：需要写入的用户区数据长度，单位为 bit
返回值	状态值，0- 成功，非 0- 失败

4.3.3.19 EFUSE_ReadUserArea

表 4-23 EFUSE_ReadUserArea 接口函数说明

信息项	说明
原型	<code>int EFUSE_ReadUserArea(char* data, int startAddr, int len)</code>
功能	从指定的位地址开始读取用户区数据。
参数	<code>char* data</code> 含义解释：存储用户区数据的数组 <code>int startAddr</code> 含义解释：起始的位地址，单位为 bit <code>int len</code> 含义解释：需要写入的用户区数据长度，单位为 bit
返回值	状态值，0- 成功，非 0- 失败

4.3.3.20 EFUSE_WriteSecretKey

表 4-24 EFUSE_WriteSecretKey 接口函数说明

信息项	说明
原型	int EFUSE_WriteSecretKey(char* key, int len);
功能	写入 secret key 数据
参数	char* key 含义解释：需要写入的 secret key 数组，长度为 16 int len 含义解释：secret key 数组长度，只能为 16
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.21 EFUSE_ReadSecretKey

表 4-25 EFUSE_ReadSecretKey 接口函数说明

信息项	说明
原型	int EFUSE_ReadSecretKey(char* key, int len);
功能	读取 secret key 数据
参数	char* key 含义解释：需要写入的 secret key 数组，长度为 16 int len 含义解释：secret key 数组长度，只能为 16
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.22 EFUSE_WriteSecureSwd

表 4-26 EFUSE_WriteSecureSwd 接口函数说明

信息项	说明
原型	int EFUSE_WriteSecureSwd(char ss);
功能	写入 Secure SWD 的数据
参数	char ss 含义解释：需要写入的 Secure SWD 数据，取值为 0 或 1
返回值	状态值，0 - 成功，非 0 - 失败

4.3.3.23 EFUSE_ReadSecureSwd

表 4-27 EFUSE_ReadSecureSwd 接口函数说明

信息项	说明
原型	int EFUSE_ReadSecureSwd(char* ss);

信息项	说明
功能	读取 Secure SWD 的数据
参数	char * ss 含义解释：用于读取 Secure SWD 数据的 buf
返回值	状态值，0 - 成功，非 0 - 失败

5 示例说明

5.1 eFuse GUI 工具示例

5.1.1 示例简介

本示例以读取 eFuse 的 WLAN MAC 地址区域为例，介绍 eFuse GUI 工具的使用方式。

5.1.2 准备工作

使用 eFuse GUI 工具读取 WLAN MAC 地址准备如下。

1. 评估板：运行 EFPG 示例工程代码。
2. 串口线：连接评估板的 UART0 插针，用于 console 控制台的输入输出以及与工具通信。
3. PC 机：用于镜像烧录和运行 eFuse GUI 工具。



说明

eFuse 工具包可在芯之联文档中心获取：<https://docs.xradiotech.com/>。

5.1.3 操作步骤

准备工作完成后，可以进行 eFuse 区域的读写操作。

下面以读取 eFuse 的 WLAN MAC 地址区域为例，简单介绍操作步骤。

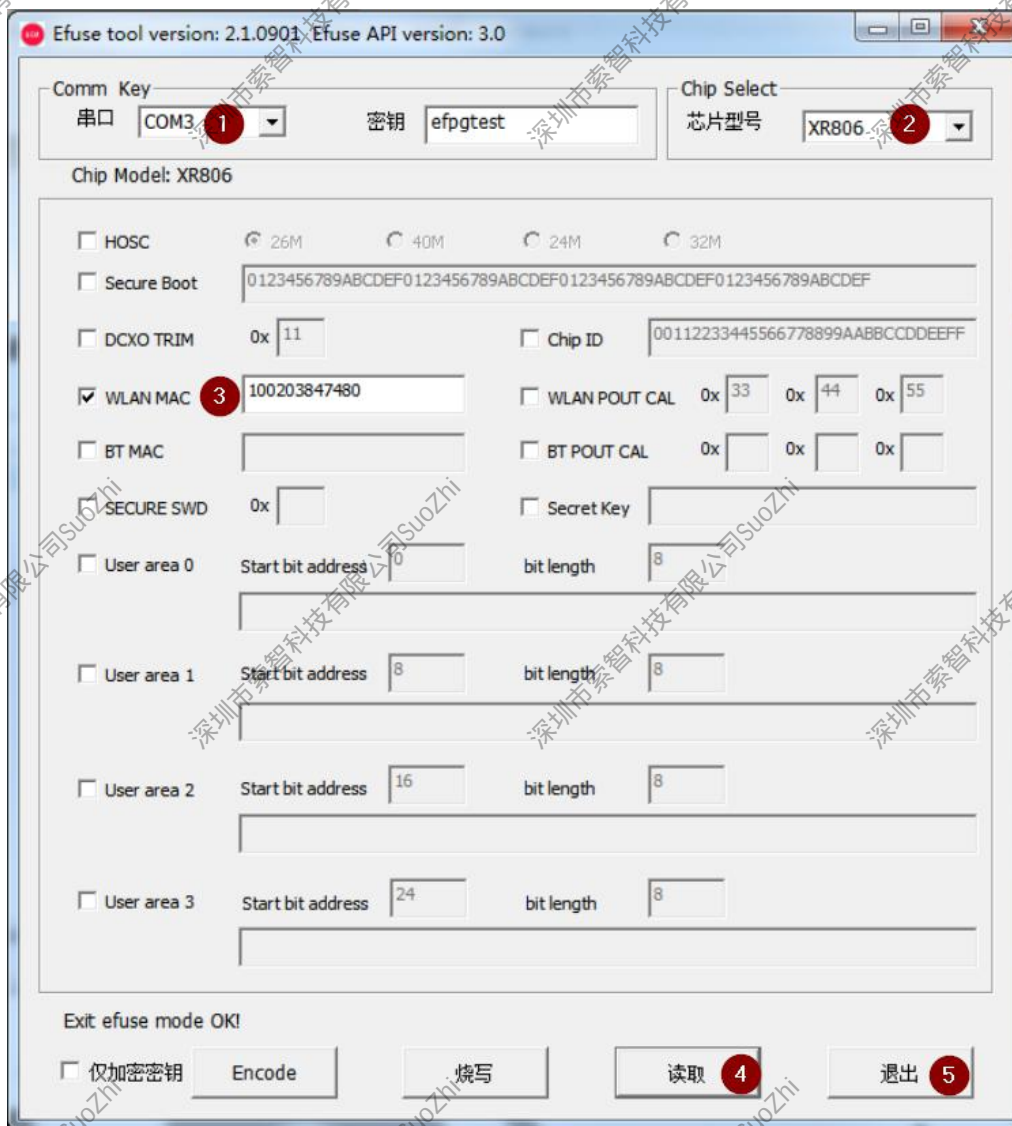
操作步骤如下：

1. 点击“串口”下拉框，选择对应串口号；
2. 点击“芯片型号”下拉框，选择 XR806；
3. 勾选 WLAN MAC 复选框；
4. 点击“读取”按钮，之后状态栏会提示操作成功，退出 eFuse 模式，并且编辑框会显示 WLAN MAC 地址数据；
5. 点击“退出”按钮退出软件；

5.1.4 效果展示

按步骤操作后，软件界面显示如下：

图 5-1 读取 WLAN MAC 地址界面



5.2 eFuse CLI 工具示例

5.2.1 示例简介

本示例以读取 eFuse 的 WLAN MAC 地址区域为例，介绍 eFuse CLI 工具的使用方式。

5.2.2 准备工作

使用 eFuse CLI 工具读取 WLAN MAC 地址准备如下。

1. 评估板：运行 EFPG 示例工程代码。
2. 串口线：连接评估板的 UART0 插针，用于 console 控制台的输入输出以及与工具通信。
3. PC 机：用于镜像烧录和运行 eFuse CLI 工具。



说明

eFuse 工具包可在芯之联文档中心获取：<https://docs.xradiotech.com/>。

5.2.3 操作步骤

准备工作完成后，可以进行 eFuse 区域的读写操作。

下面以读取 eFuse 的 WLAN MAC 地址区域为例，简单介绍操作步骤。

操作步骤如下：

1. 使用 PC 命令行进入工具所在目录；
2. 执行如下命令即可读取 WLAN MAC 地址数据：

```
efuse_cli_tool.exe -c 3 -k efgptest -r mac
```

5.2.4 效果展示

按步骤操作后，窗口输出打印信息如下：

```
D:\efuse_oem_cli_tool\Release>efuse_cli_tool.exe -c 3 -k efgptest -r mac
read mac
com:3
key:efgptest
mac: 10 02 03 84 74 80
Operation OK!
```

5.3 eFuse API 使用示例

5.3.1 示例简介

本示例以调用读取 eFuse 的 WLAN MAC 地址的 API 代码为例，介绍 eFuse API 的使用方式。

5.3.2 准备工作

使用 eFuse API 调用读取 WLAN MAC 地址的代码准备如下。

1. PC 机：用于编辑和编译生成 eFuse API 使用工具。
2. VS2008：用于开发 eFuse API 所使用工具的软件环境。



说明

eFuse 工具包可在芯之联文档中心获取：<https://docs.xradiotech.com/>。

5.3.3 代码示例

读取 eFuse 的 WLAN MAC 地址区域的值

```
static DWORD WINAPI ReadEfuseThread(LPVOID lpParameter)
{
    Cefuse_oem_toolDlg * pMain = (Cefuse_oem_toolDlg *)lpParameter;

    // 初始化 EFUSE 库
    if (EFUSE_InitEfuse(3, _T("efpgtest"), 8))
    {
        pMain->m_strMsg.Format(_T("%s"), EFUSE_GetMsg());
        SendMessage(pMain->m_hWnd, EFUSE_MSG, NULL, NULL);
        EFUSE_ExitEfuse();
        return -1;
    }

    // 读取 MAC 地址
    BYTE mac[6];
    CString strTmp;
    if (EFUSE_ReadMacAddr((char*)mac, sizeof(mac)))
    {
        pMain->m_strMsg.Format(_T("%s"), EFUSE_GetMsg());
        SendMessage(pMain->m_hWnd, EFUSE_MSG, NULL, NULL);
        EFUSE_ExitEfuse();
        return -1;
    }

    // 将 MAC 地址值显示到主界面中
    pMain->m_strMacAddr = _T("");
    for (int i = 0; i < 6; i++)
    {
        strTmp.Format(_T("%02X"), mac[i]);
        pMain->m_strMacAddr += strTmp;
    }

    // 释放 EFUSE 库资源
    EFUSE_ExitEfuse();
    pMain->m_strMsg.Format(_T("%s"), EFUSE_GetMsg());
    SendMessage(pMain->m_hWnd, EFUSE_MSG, NULL, NULL);
    return 0;
}
```

著作权声明

版权所有©2020 广州芯之联科技有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由广州芯之联科技有限公司（“芯之联”）拥有并保留一切权利。

本文档是芯之联的原创作品和版权财产，未经芯之联书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明



（不完全列举）均为广州芯之联科技有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与广州芯之联科技有限公司（“芯之联”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，芯之联概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。芯之联尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，芯之联概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予芯之联的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。芯之联不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。芯之联不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。