

XR806 EFPG 中间件 开发指南

- Fight Hard State of the State

发布时间: 2020-11-18

深圳指揮推拔推開地間等四位的

·操制的操制就推翻就推翻。



·探州(大學) (1886)

文档密级: 秘密

版本历史

e ^X	版本	日期	责任人	版本描述	EXIII THE	
1	1.0	2020-11-18	AWA 1680	创建文档。	-\ <u>\</u>	-1/1



版权所有©广州芯之联科技有限公司。保留一切权利



目录

x历史			15%	
₹	·\$ ^{YY}	-\$K**		······ii
各目录				iv
扩言				1
1.1 文档简	5介			1
	转者			1
	5围			1
		*.	in	1
()	定		i siok	1
180	标志说明	180		1
1.4.2	地址与数据描述方法约定	ARX		1
1.4.3	数值单位约定			2
概述				:(*.3
2.1 背景访	台明			3
2.2 规格特	<u> </u>			3
	拉置			3
	;eFuse 烧写方案			ر
				5
- 1	ield定义	cuolini	saudini	5
应用说明	The last		, ALV	7
4.1 应用简	ī述			7
4.2 配置访	往明			7
4.3 接口说	台明			
4.3.1	eFuse 烧写工具通信接口	r	v	9
	4.3.1.1 efpg_start			9
4.3.2	eFuse 区域数据读取接口			10
	4.3.2.1 efpg_read_mac			10
	4.3.2.2 efpg_read_dcxo			
	4.3.2.3 efpg_read_pout			
	4.3.2.4 efpg_read_chipid	177	177	12
_1150	4.3.2.5 efpg_read_boot	The state of the s	Tal Siron	12
A SHOW	4.3.2.7 efpg_read_user_area	N. P. L. P.	N. T.	12
A KXY	4.3.2.8 efpg_read_all_field	X		
SUP	A. C.	1847 - 1847 - 1848 - 18	The second	1675-



文档密级: 秘密

KRAD TECH	Suolhi	Suothi	文档密级:	秘密
4.3.2.9 efpg_layout.	- 1/2 - 1/2	## P. J. Z.		14
5 示例说明				15
5.1 示例简介				15
5.1.1 获取方式:※	· · ·		- Krii ii.	.::(*15
5.1.2 准备工作				15
5.1.3 操作步骤				15
5.2 关键实现				15
5.3 效果展示				18
附录 A: 术语表				21
深圳析療精,持持關,持持,持持,持持,持,持,	AMENTAL SUBJECTION	M. A. A. B.	AND THE REPORT OF THE PARTY OF	

THE REPORT OF THE PARTY OF THE

A STATE OF THE PARTY OF THE PAR

·探測情報機構構構構開



表格目录

表	2-1	EFPG 中间件的烧写功能特性			3
表	2-2	EFPG 中间件的文件位置	-\$ ¹	-	4
表	2-3	EFPG 中间件的文件说明			4
表	4-1	XR806 EFPG 模块配置列表			7
表	4-2	EFPG 中间件模块接口简介			9
表	4-3	efpg_start 接口函数说明			9
表	4-4	efpg_read_mac 接口函数说明			10
表	4-5	efpg_mac_mode_t 枚举类型说明	(ئى.	J ^{rii})10
表	4-6	efpg_read_dcxo 接口函数说明			10
表	4-7	efpg_read_pout 接口函数说明			11
表	4-8	efpg_pout_cal_mode_t 枚举类型说明			11
表	4-9	efpg_read_chipid 接口函数说明			11
表	4-10	efpg_read_boot 接口函数说明			12
表	4-11	efpg_read_hosc 接口函数说明			12
表	4-12	efpg_read_user_area 接口函数说明			12
表	4-13	efpg_read_all_field 接口函数说明			13
表	4-14	efpg_layout 接口函数说明			14
表	A-1	术语表,:	۷	<u></u>	21 بينا

A THE TEN STORY

A HARING THE SUCKIN

A THE SUCKING THE STATE OF THE



1 前言

1.1 文档简介

本文档介绍了 XR806 平台上 EFPG 模块的使用方法。

1.2 目标读者

XR806 SDK 用户、EFPG 模块开发使用人员。

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:

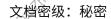
标识	说明		
企 警告	该标志后的说明应给予格外关注,如果不遵守,可能会导致人员受伤或死亡。		
注意	提醒操作中应注意的事项。不当的操作可能会损坏器件,影响可靠性、降低性能等。		
说明	为准确理解文中指令、正确实施操作而提供的补充或强调信息。		
◎、窍门	一些容易忽视的小功能、技巧。了解这些功能或技巧能帮助解决特定问题或者节省操作时间。		

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 sudhi	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

XRAD TECH

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

AND THE REPORT OF THE PARTY OF

IN THE LEGISTICAL.

HA HALL BUOL

HA TO SUO

THE LEGISTON



2 概述

2.1 背景说明

EFPG(eFuse Programming)模块主要用于 OEM(Original Equipment Manufacturer)端对 eFuse 相应区域 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 以及 USER AREA(用户区域)。EFPG 模块对上述区域 的烧写和读取功能可分为三类:

- 1. 配合 OEM 端 eFuse 烧写工具烧写 HOSC、BOOT、DCXO、POUT 和 MAC 数据。
- 2. 提供接口读取 eFuse 上 HOSC、BOOT、DCXO、POUT、MAC 以及 CHIPID 数据。
- 3. 提供接口读取 eFuse 上 USER AREA 区域,且可配合 eFuse 烧写工具对 USER AREA 区域进行烧写。

2.2 规格特性

EFPG 中间件模块提供了以下功能特性。

表 2-1 EFPG 中间件的烧写功能特性

规格类型	规格描述	备注
	支持读取 CHIPID	
	支持烧写、读取 HOSC	
\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	支持烧写、读取 BOOT	
为 eFuse 烧写工具 提供功能命令	支持烧写、读取 DCXO	
IVE IV SOUPERIS 4	支持烧写、读取 POUT	
A Lines	支持烧写、读取 MAC	
	支持烧写、读取 USER AREA	A TOPIC OF THE PERSON OF THE P
	支持读取 CHIPID	A STATE OF THE STA
	支持读取 HOSC	
₩π.₩π.₩±	支持读取 BOOT	-\/-
为用户提供功能接	支持读取 DCXO	
	支持读取 POUT	
	支持读取 MAC	
	支持读取 USER AREA	

2.3 文件位置

以 SDK 包为根目录,本中间件涉及到的主要文件位置如下。



表 2-2 EFPG 中间件的文件位置

18.		X /4, X /4,
组件名	文件分类	文件位置
	源码文件	./src/efpg
EFPG	头文件	./include/efpg.h
	示例工程	./project/example/efpg/

关键文件说明如下。

表 2-3 EFPG 中间件的文件说明

文件名	文件说明	1/2	1/2
efpg.c	EFPG 模块的主函数	[B300]	- Als July

□ 说明

XR806 SDK 可在以下 GitHub 仓库获取: https://github.com/XradioTech/xr806_sdk.git

AND THE VENEZA

AR TO THE STATE OF THE STATE OF

NA THE WALL STATE OF THE STATE

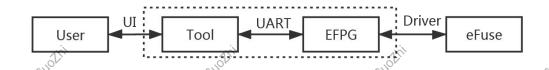


3 技术说明

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 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。

OEM 端 eFuse 烧写工具的具体使用可参考《XR806_eFuse 工具_使用指南》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,代码中 key 在

SDK/project/common/cmd/cmd_efpg.c 文件中(如下所示)。Key 的最大长度由宏 EFPG_KEY_LEN_MAX 定义,默认为 64。

```
const char *efpg_key = "efpgtest";
```

3.2 EFPG Field 定义

EFPG Field 表示通过 EFPG 模块可以直接读取的已定义区域的数据。这些已定义区域包括 OEM 端负责烧写的 HOSC、BOOT、DCXO、POUT 和 MAC,此外还包括 CHIPID。代码中对 EFPG Field 的定义为:

```
typedef enum efpg_field {
    EFPG FIELD HOSC = 0,
                              /* data buffer size: 1 byte */
    EFPG_FIELD_BOOT,
                               /* data buffer size: 32 bytes */
                                /* data buffer size: 6 bytes */
    EFPG_FIELD_MAC_WLAN,
    EFPG_FIELD_DCXO,
                               /* data buffer size: 1 byte */
    EFPG_FIELD_POUT_WLAN,
                               /* data buffer size: 3 bytes */
    EFPG_FIELD_CHIPID,
                              /* data buffer size: 16 bytes */
    EFPG_FIELD_UA,
                               /* data buffer size: V1(1447~2047) : V2(765~1023 bit) : V3(954~1022
                               /* data buffer size: 3 bytes */
    EFPG_FIELD_POUT_BT,
                                /* data buffer size: 6 bytes */
    EFPG FIELD MAC BT,
                               /* data buffer size: 16 bytes */
    EFPG_FIELD_SECRETKEY,
    EFPG_FIELD_SECURESWD,
                               /* data buffer size: 1 bytes */
                               /* data buffer size: 128 bytes */
    EFPG_FIELD_ALL,
    EFPG FIELD NUM
```



}efpg_field_t;





4 应用说明

4.1 应用简述

EFPG 中间件模块已经内嵌到 XR806 SDK,应用步骤如下:

- 1. 若需要使用 EFPG 模块对 eFuse 区域进行烧写,则只需使用 eFuse 烧写工具进行操作即可(烧写前开发板要正常启动平台初始化,且支持 cmd_efpg 命令,可参见《XR806_eFuse 工具_使用指南》文档)。
- 2. 若用户需要使用 EFPG 模块的接口,对 eFuse 区域进行读取,则需在用户应用代码中添加 EFPG 模块 头文件(参见"2.3 文件位置"章节),调用 EFPG 接口进行使用(接口使用参见"4.3 接口说明" 章节)。

4.2 配置说明

XR806 EFPG 模块可对 eFuse 中 DCXO_TRIM、POUT_CAL、BT_POUT_CAL、MAC_WLAN_ADDR、MAC_BT_ADDR 等区域的个数进行配置,配置说明如下表所示。

表 4-1 XR806 EFPG 模块配置列表

配置项	配置说明
	设置说明: 此项配置为 eFuse 中 DCXO_TRIM 区域个数(区域个数为整数,范围为[0, 2])
DCXO_TRIM 区域个数	设置位置: efpg 目录下的 efpg_i.h 文件,SDK/src/efpg/efpg_i.h 设置方式:
	配置宏 EFPG_DCXO_TRIM_NUM 的值,如: #define EFPG_DCXO_TRIM_NUM (1) //表示 DCXO_TRIM 区域个数为 1
绿树	设置说明: 此项配置为 eFuse 中 WLAN 的 POUT_CAL 区域个数(整数,范围为[0, 2])
POUT_CAL区域个数 (WLAN)	设置位置: efpg 目录下的 efpg_i.h 文件,SDK/src/efpg/efpg_i.h
	设置方式: 配置宏 EFPG_POUT_CAL_NUM 的值,如:
a a a a a a a a a a a a a a a a a a a	#define EFPG_POUT_CAL_NUM (1) //表示 WLAN 的 POUT_CAL 区域个数 为 1
BT_POUT_CAL 区域 个数(BT)	设置说明: 此项配置为 eFuse 中 BT 的 POUT CAL 区域个数(整数,范围为[0, 2])



	//\S	
1	配置项	配置说明
	<i>A</i>	设置位置:
	-[\$ [±]] [fi ²]	efpg 目录下的 efpg_i.h 文件,SDK/src/efpg/efpg_i.h
		设置方式:
		配置宏 EFPG_BT_POUT_CAL_NUM 的值,如:
		#define EFPG_BT_POUT_CAL_NUM (1) //表示 BT 的 POUT_CAL 区域个数
		为 1
İ		设置说明:
		此项配置为 eFuse 中 MAC_WLAN_ADDR 区域个数(整数,范围为[0, 4])
	cuolhi	设置位置:watri
	MAC_WLAN_ADDR 区域个数	efpg 目录下的 efpg_i.h 文件,SDK/src/efpg/efpg_i.h
7		设置方式:
	16 A	配置宏 EFPG_MAC_WLAN_ADDR_NUM,如:
	深圳市	#define EFPG_MAC_WLAN_ADDR_NUM (0) //表示 MAC_WLAN_ADDR 区域个数为 0
		设置说明:
		此项配置为 eFuse 中 MAC_BT_ADDR 区域个数(整数,范围为[0, 5])
		设置位置:
	MAC_BT_ADDR区域	efpg 目录下的 efpg_i.h 文件,SDK/src/efpg/efpg_i.h
	个数	
	oZhi	设置方式:
	The Park	配置宏 EFRG_MAC_BT_ADDR_NUM,如: //表示 MAC_BT_ADDR_区域企数
4	No.	#define EFPG_MAC_BT_ADDR_NUM (1) //表示 MAC_BT_ADDR 区域个数为 1

川 说明

- 上述区域个数配置,一般在 eFuse 烧写前统一配置好。待烧写完毕后,各区域个数配置一般不作 修改,防止出现数据读取不匹配的情况。
- 2. 区域个数配置后,其他区域大小会被自动计算和检测,若总长度超出 eFuse 区域总大小,则编译会报错,用户进行修改调整即可。



4.3 接口说明

XR806 EFPG 中间件模块主要包括两类主要接口,一类是与 eFuse 烧写工具的通信接口,用于和 eFuse 烧写工具建立连接,为便烧写工具对 eFuse 区域进行烧写和读取;另一类是为用户提供的 eFuse 区域数据读取接口。EFPG 接口简介如下表所示。

表 4-2 EFPG 中间件模块接口简介

接口名	简要介绍
eFuse 烧写工具通信接口	包括 1 个接口函数,用于 EFPG 模块和 eFuse 烧写工具进行通信。
eFuse 区域数据读取接口	包括 9 个接口函数,为用户提供,方便用户对 eFuse 区域进行数据读取。

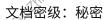
上述接口均于 SDK/src/efpg/efpg_efuse.c 文件中实现,XR806 EFPG 中间件模块接口的详细说明如下。

4.3.1 eFuse 烧写工具通信接口

4.3.1.1 efpg_start

表 4-3 efpg_start 接回函数说明

信息项	说明
	int efpg_start(uint8_t *key, uint8_t key_len, UART_ID uart_id, efpg_cb_t start_cb,
原型	efpg_cb_t stop_cb);
	使系统进入烧写模式,与 OEM 端的 eFuse 烧写工具建立通信。烧写工具发送烧
 功能	写命令后,会调用 efpg_start()函数,使系统进入 eFuse 烧写模式。此接口会在
-2280	使用 OEM 端 eFuse 烧写工具时自动被调用,烧写结束后工具会通知系统退出
	eFuse 烧写模式。
1ri	uint8_t *key
Suot	含义解释:烧写工具发送的 key 使用说明:N/A
	使用说明·N/A
	uint8_t key_len
	含义解释:key 的长度
深圳市	使用说明:N/A
	UART_ID uart_id
参数	含义解释:烧写工具与 EFPG 模块通信的 UART 的 ID
	使用说明:N/A
	efpg_cb_t start_cb
	含义解释:系统进入烧写模式前,执行的回调函数
	使用说明:N/A
.oZni	efpg_cb_t stop_cb
The Contraction of the Contracti	含义解释》系统退出烧写模式后,执行的回调函数
R. V	使用说明:N/A
返回值	0、成功
	E. The The Table 1





信息项	说明	A	A THE WAY
- K	1: 失败	Kiz-	

4.3.2 eFuse 区域数据读取接口

4.3.2.1 efpg_read_mac

表 4-4 efpg_read_mac 接口函数说明

信息项	说明
原型	uint16_t efpg_read_mac(efpg_mac_mode_t mode, uint8_t *r_data);
功能	读取 mac 地址
A STORY OF THE STO	efpg_mac_mode_t mode 含义解释:mac 地址的读取类型
参数	使用说明:mode 为枚举类型,可参考表 4-5 efpg_mac_mode_t 枚举类型说明。 uint8_t *r_data 含义解释:存放读取结果的 buffer
	使用说明:N/A
返回值	EFPG_ACK_OK: 成功 EFPG_ACK_RW_ERR: 读写错误 EFPG_ACK_NODATA_ERR: 无有效数据

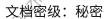
表 4-5 efpg_mac_mode_t 枚举类型说明

	成员项	说明	sustri	Sudini
16	EFPG_MAC_WLAN	含义解释:	表示读取 WLAN 的 mac 地址	A THE VE
		使用说明:	N/A	King Kanada K
	EFPG_MAC_BT	含义解释:	表示读取 BT 的 mac 地址	操欄的
		使用说明:	N/A	

4.3.2.2 efpg_read_dcxo

表 4-6 efpg_read_dcxo 接口函数说明

信息项	说明	
原型	uint16_t efpg_read_dcxo(uint8_t *r_data);	1/1
功能	读取 dcxo	- ASUO
参数	uint8_t*r_data 含义解释:存放读取结果的 buffer	A STATE OF THE STA





信息项	说明	A THE PARTY OF THE
	使用说明:N/A	A CONTRACTOR OF THE PROPERTY O
g till the	EFPG_ACK_OK:成功	EHITE TO SELECT
返回值	EFPG_ACK_RW_ERR:读写错误	- th
	EFPG_ACK_NODATA_ERR: 无有效数	据

4.3.2.3 efpg_read_pout

表 4-7 efpg_read_pout 接口函数说明

信息项	说明
原型。水	uint16_t efpg_read_pout(efpg_pout_cal_mode_t mode, uint8_t *r_data);
功能	读取 pout
	efpg_pout_cal_mode_t mode 含义解释: pout 的读取类型
₹ ₩1	使用说明:mode 为枚举类型,可参考表 4-8 efpg_pout_cal_mode_t 枚举类型说明。
参数	uint8_t *r_data
	含义解释:存放读取结果的 buffer
	使用说明:N/A
	EFPG_ACK_OK: 成功
返回值	EFPG_ACK_RW_ERR: 读写错误
	EFPG_ACK_NODATA_ERR: 无有效数据

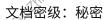
表 4-8 efpg_pout_cal_mode_t 枚举类型说明

成员项	说明			
	含义解释:	表示读取 WLAN 的 pout		
EFPG_POUT_WLAN		A TIME	::\$\frac{1}{2}	-ŵ
1,	使用说明:	N/A	1,	٧,
	含义解释:	表示读取 BT 的 pout		
EFPG_POUT_BT				
	使用说明:	N/A		

4.3.2.4 efpg_read_chipid

表 4-9 efpg_read_chipid 接口函数说明

信息项	说明	. Alsuolin	,_@\$volth
原型	uint16_t efpg_read_chipid(u	uint8_t *r_data);	
功能	读取 chipid		





	_//\sigma\		<u> </u>
信息项	说明	A THE W	A TOP OF THE PROPERTY OF THE P
	uint8_t *r_data	X	
参数	含义解释:存放读取结果的 b	uffer	NA TANK
-1/k	使用说明:N/A	-(\$\hat{x})	\$
	EFPG_ACK_OK:成功		
返回值	EFPG_ACK_RW_ERR:读写错设	룾	
	EFPG_ACK_NODATA_ERR:无存	j效数据	

4.3.2.5 efpg_read_boot

表 4-10 efpg_read_boot 接口函数说明

信息项	说明 Javathi Javathi Javathi
原型	uint16 t efpg_read_boot(uint8_t *r_data);
功能	读取 boot
参数	uint8_t *r_data 含义解释:存放读取结果的 buffer 使用说明:N/A
返回值	EFPG_ACK_OK: 成功 EFPG_ACK_RW_ERR: 读写错误 EFPG_ACK_NODATA_ERR: 无有效数据

4.3.2.6 efpg_read_hosc

表 4-11 efpg_read_hosc 接口函数说明

信息项	说明 alv	Suot Suot
原型	uint16_t efpg_read_hosc(uint8_t *r_data	a);
功能	读取 hosc	A Company of the Comp
£ till th	uint8_t *r_data	e tille.
参数	含义解释:存放读取结果的 buffer	*
	使用说明:N/A	
	EFPG_ACK_OK:成功	
返回值	EFPG_ACK_RW_ERR: 读写错误	
	EFPG_ACK_NODATA_ERR: 无有效数据	

4.3.2.7 efpg_read_user_area

表 4-12 efpg_read_user_area 接口函数说明

信息项	说明	A STATE OF THE STA	A STATE OF THE STA
原型	uint16_t efpg_re	ad_user_area(uint16_t start, uint16_	_t num, uint8_t *r_data);



信息项	说明。
功能	读取用户地址区域 user area
	uint16_t start 含义解释:读取的起始地址
	使用说明:若从第一个比特(bit)开始读,则 start 为 0
	uint16_t num
参数	含义解释:读取的比特(bit)数
	使用说明:N/A
	uint8_t *r_data
	含义解释:存放读取结果的 buffer
Sudhi	使用说明:N/A
NO TO THE PARTY OF	EFPG_ACK_OK: 成功
返回值	EFPG_ACK_RW_ERR: 读写错误
	EFPG_ACK_NODATA_ERR: 无有效数据

4.3.2.8 efpg_read_all_field

表 4-13 efpg_read_all_field 接口函数说明

信息项	说明
原型	uint16_t efpg_read_all_field(uint16_t start, uint16_t num, uint8_t *data);
功能	读取 eFuse 任意区域
参数	uint16_t start 含义解释:读取的起始地址 使用说明:N/A uint16_t num 含义解释:读取的比特(bit)数 使用说明:N/A uint8_t *r_data 含义解释:存放读取结果的 buffer 使用说明:N/A
返回值	EFPG_ACK_OK: 成功 EFPG_ACK_RW_ERR: 读写错误
	EFPG_ACK_NODATA_ERR: 无有效数据

版权所有©广州芯之联科技有限公司。保留一切权利



4.3.2.9 efpg_layout

表 4-14 efpg_layout 接口函数说明

信息项	说明
原型	void efpg_layout(void);
功能	打印 eFuse 区域布局
参数	无
返回值	无



A THE THE PROPERTY OF THE PROP

EF HARITAGE

A THE VIEW OF THE PROPERTY OF

5 示例说明

EFPG 模块接口主要用于读取指定的 eFuse 区域信息,EFPG 示例工程展示了在 XR806 SDK 中进行 eFuse 区域数据读取的代码实现方法。



eFuse 区域只能烧写一次,对 eFuse 区域的烧写需配合 OEM 端 eFuse 工具进行操作,可参见《XR806 eFuse 工具 使用指南》文档。

5.1 示例简介

本工程示例会读取 eFuse 里的各个区域的字段信息,包括 hosc 信息、boot 信息、mac 地址信息、chipid 信息以及用户区域数据。读取成功,则打印此信息;读取失败,则报错。

5.1.1 获取方式

EFPG 示例有示例工程代码,位于 XR806 SDK 的/project/example/efpg 目录,以下此示例工程简称为 EFPG 示例工程。



XR806 SDK 可在以下 GitHub 仓库获取: https://github.com/XradioTech/xr806_sdk.git

5.1.2 准备工作

EFPG 示例工程的硬件准备有如下。

- 1. 评估板:运行示例工程代码。
- 2. 串口线:连接评估板的 UARTO 插针,用于 console 控制台的输入输出。
- 3. PC 机:用于镜像烧录和 console 控制的输入输出。

EFPG 示例工程的软件准备,包括烧写工具、代码编译和烧写操作,请参见《XR806_SDK_快速入门指南》。

5.1.3 操作步骤

EFPG 示例工程无需控制命令,完成烧写后,复位即可,示例代码自动运行。

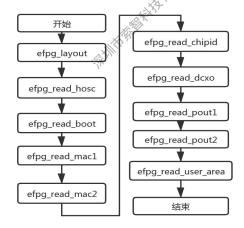
5.2 关键实现

系统平台的初始化在 XR806 SDK 的 platform_init()函数中完成,具体请阅读此函数。本文重点描述 EFPG 模块对 eFuse 各区域数据读取的操作过程,示例代码流程如下。

版权所有©广州芯之联科技有限公司。保留一切权利



图 5-1 EFPG 示例流程图



具体操作见下面的示例代码。

```
#define EFPG_USER_AREA_LEN 8
static void efpg_log(uint8_t *buf, uint8_t size)
     int i;
     for (i = 0; i < size; i++)
          printf("[%02d] 0x%02x\n", i, buf[i]);
     printf("\n");
}
static void efpg_ua_log(uint8_t *buf, int len_bits)
   int len_bytes;
extern void print_hex_dump_bytes(const void *addr, size_t len);
     len_bytes = len_bits / 8;
     print_hex_dump_bytes(buf, len_bytes);
}
static int efpg_read_example()
{
     uint16_t ret;
     int len_bits;
     uint8_t buf[32];
     uint8_t ua_buf[128];
   efpg_layout();
     ret = efpg_read_hosc(buf);
```



```
if (ret == EFPG_ACK_OK)
     printf("efpg read hosc success.\n");
     efpg_log(buf, 1);
} else {
     printf("efpg read hosc fail.ret:%u\n", ret);
}
ret = efpg_read_boot(buf);
if (ret == EFPG_ACK_OK) {
     printf("efpg read boot success.\n");
     efpg_log(buf, 32);
} else {
  printf("efpg read boot fail.ret.%u\n", ret);
ret = efpg_read_mac(EFPG_MAC_WLAN, buf);
if (ret == EFPG_ACK_OK) {
     printf("efpg read wlan mac success.\n");
     efpg_log(buf, 6);
     printf("efpg read wlan mac fail.ret:%u\n", ret);
}
ret = efpg_read_mac(EFPG_MAC_BT, buf);
if (ret == EFPG_ACK_OK) {
     printf("efpg read bt mac success.\n");
  efpg_log(buf, 6);
} else {
     printf("efpg read bt mac fail.ret:%u\n", ret);
ret = efpg_read_chipid(buf);
if (ret == EFPG_ACK_OK) {
     printf("efpg read chipid success.\n");
     efpg_log(buf, 16);
} else {
     printf("efpg read chipid fail.ret:%u\n", ret);
}
ret = efpg_read_dcxo(buf);
if (ret == EFPG_ACK_OK) {
     printf("efpg read dcxo success.\n");
     efpg_log(buf, 1);
} else {
```

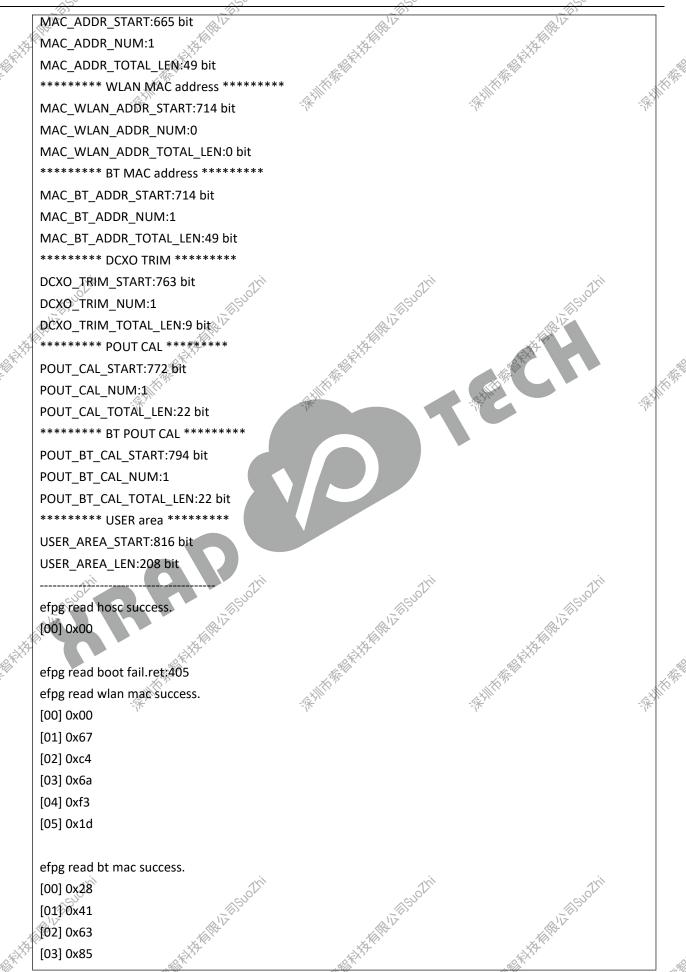


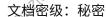
```
printf("efpg read dcxo fail.ret:%u\n", ret);
}
ret = efpg_read_pout(EFPG_POUT_WLAN, buf);
if (ret == EFPG_ACK_OK) {
    printf("efpg read wlan pout success.\n");
    efpg_log(buf, 3);
} else {
    printf("efpg read wlan pout fail.ret:%u\n", ret);
}
ret = efpg_read_pout(EFPG_POUT_BT, buf);
if (ret == EFPG_ACK_OK) {
    printf("efpg read bt pout success.\n");
    efpg_log(buf, 3);
} else {
    printf("efpg read bt pout fail.ret:%u\n", ret);
}
len_bits = EFPG_USER_AREA_LEN;
ret = efpg_read_user_area(0, len_bits, ua_buf);
if (ret == EFPG_ACK_OK) {
    printf("efpg read user area success.\n");
    efpg_ua_log(ua_buf, len_bits);
} else {
     printf("efpg read user area fail.ret:%u\n", ret);
```

5.3 效果展示

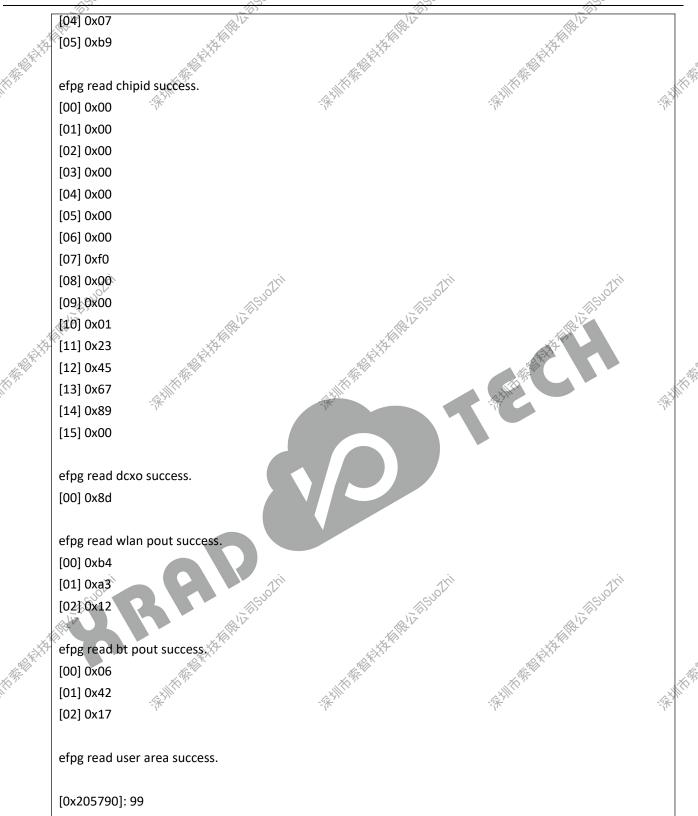
在评估板中运行 FDCM 示例程序后,在控制台中打印输出如下。











可以看到,EFPG 模块成功读取到 eFuse 的各区域信息。其中,若 eFuse 某些区域未被烧写,则会出现读取失败提示,如示例效果展示中出现"read boot fail"是因为示例板没有烧写过 eFuse 的 boot 区域。

版权所有©广州芯之联科技有限公司。保留一切权利



附录 A: 术语表

表 A-1 术语表

В		
BT	Bluetooth	蓝牙
0		
OEM	Original Equipment Manufacturer	原始设备制造商
U		
UART	Universal Asynchronous Receiver/Transmitter	通用异步收发传输器
W	1540 ¹	Suot
WŁAN	Wireless Local Area Network	无线局域网

RATE OF THE PROPERTY OF THE PR

A William Short

All the living the state of the

* High To the Strock



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



XRAD TECH、 **芯之联** (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。

》 版权所有©广州芯之联科技有限公司。保留一切权利