CV181x/CV180x eFuse 使用指南

Version: 0.4

Release date: 2023-02-06

目录

修订记录

Revision	Date	Description
0.1	2022-06-01	Initial
0.2	2022-09-28	Rename processor
0.3	2023-02-01	更新安全启动 efuse 烧录流程
0.4	2023-02-06	CV181x/CV180x 文档融合

1 声明

法律声明

本数据手册包含北京晶视智能科技有限公司（下称“晶视智能”）的保密信息。未经授权，禁止使用或披露本数据手册中包含的信息。如您未经授权披露全部或部分保密信息，导致晶视智能遭受任何损失或损害，您应对因之产生的损失/损害承担责任。

本文件内信息如有更改，恕不另行通知。晶视智能不对使用或依赖本文件所含信息承担任何责任。

本数据手册和本文件所含的所有信息均按“原样”提供，无任何明示、暗示、法定或其他形式的保证。晶视智能特别声明未做任何适销性、非侵权性和特定用途适用性的默示保证，亦对本数据手册所使用、包含或提供的任何第三方的软件不提供任何保证；用户同意仅向该第三方寻求与此相关的任何保证索赔。此外，晶视智能亦不对任何其根据用户规格或符合特定标准或公开讨论而制作的可交付成果承担责任。

联系我们

地址 北京市海淀区丰豪东路 9 号院中关村集成电路设计园（ICPARK）1 号楼

深圳市宝安区福海街道展城社区会展湾云岸广场 T10 栋

电话 +86-10-57590723 +86-10-57590724

邮编 100094（北京）518100（深圳）

官方网站 https://www.sophgo.com/

技术论坛 https://developer.sophgo.com/forum/index.html

2 eFuse 使用指南

2.1 eFuse 概述

处理器内部集成 eFuse 空间，可供安全启动和和 448 bits 的用户自定义区域。

具体 eFuse 分区请参考eFuse 用户可写入区域 和eFuse 安全设定字段 。

表 2.1: eFuse 用户可写入区域

Name	Size	Comment
USER	40 Bytes	用户自定义区域
DEVICE_ID	8 Bytes	设备序号
HASH0_PUBLIC	32 Bytes	安全启动 RSA 公钥哈希值
LOADER_EK	16 Bytes	安全启动 AES 加密密钥
DEVICE_EK	16 Bytes	用户自定义区域，可被锁定
SECUREBOOT	4 Bytes	使能安全启动

表 2.2: eFuse 安全设定字段

Name	Comment
LOCK_HASH0_PUBLIC	锁定安全启动 RSA 公钥哈希值区域，让此区域无法读写
LOCK_LOADER_EK	锁定安全启动 AES 加密密钥区域，让此区域无法读写
LOCK_DEVICE_EK	锁定 DEVICE_EK，让此区域无法读写
SECUREBOOT	使能安全启动

2.2 安全启动 eFuse 设置流程

注意： eFuse 每一位写入 1 后无法擦除 (只允许从 0 变成 1)，写入前请注意

指定的 eFuse 锁定后无法再读取或写入，锁定前请注意

2.2.1 查看密钥内容

在 PC 上查看密钥内容:

```
# 查看AES加解密秘钥
host$ xxd -p -c 256 loader_ek.key
668f8b6655a89f7cb8ee5cbd6f2c914e

# 获取RSA验签所需 sha256 值
# 执行签署脚本fipsign.py时，脚本会打印所需sha256值，如下：
host$ ./fipsign.py ......
Host$ .......
Host$ INFO:root:KPUB_
↪HASH:978bc2031b9377dadb4c7c34467ee985806a63a3ac8ee293a3f0eddcd2b789d8
Host$ .......
```

- KPUB_HASH: 后面的字符串就是所需 sha256 值

2.2.2 写入密钥

1. 写入 loader_ek.key 进 eFuse 的“加密密钥”区域，数据为 16 个数组，以 16 进位表示成 32 个数字。如果未使用加密功能可跳过这步骤。

```
u-boot# efusew LOADER_EK 668f8b6655a89f7cb8ee5cbd6f2c914e
```

2. 写入验签所需 sha256 值进 eFuse 的“验签所需 SHA256 摘要”区域，数据为 32 个数组，以 16 进位表示成 64 个数字

```
u-boot# efusew LOADER_EK 668f8b6655a89f7cb8ee5cbd6f2c914e
```

3. 锁定密钥区域，防止误写

```
u-boot# efusew LOCK_LOADER_EK 01
u-boot# efusew LOCK_HASH0_PUBLIC 01
```

2.2.3 使能安全启动

1. enable 验签流程

```
u-boot# efusew SECUREBOOT 01
```

2. enable 验签和解密流程

```
u-boot# efusew SECUREBOOT 02
```

注意:

1. 安全启动使能后无法再变更，秘钥和开关写入前请注意数据正确
2. 使能安全启动需要和 efuse 写入数据以及 fip.bin 匹配，例如：enable 解密流程后需要使用已签名和已加密的 FIP.bin，只签名的 FIP.bin 无法烧录和启动

2.3 eFuse u-boot 命令参考

u-boot 提供以下命令存取 eFuse:

- efuser：读取 eFuse 区域。
- efusew：写入 eFuse 区域。

2.3.1 efuser

【描述】 读取 eFuse 区域数据。

【语法】 efuser EFUSE_AREA

【参数】

参数名称	描述
EFUSE_AREA	eFuse 区域名称，参考eFuse 用户可写入区域和eFuse 安全设定字段。

【举例】 读取 efuse USER 区域数据

```
cv181x/cv180x# efuser USER
00000000: 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
00000020: 00 00 00 00 00 00 00 00                            ........
cv181x/cv180x#
```

2.3.2 efusew

【描述】 将数据写入 eFuse 区域。

【语法】 efuser EFUSE_AREA DATA

【参数】

参数名称	描述
EFUSE_AREA	eFuse 区域名称，参考eFuse 用户可写入区域和eFuse 安全设定字段。
DATA	用于写入 eFuse 的数据，以 16 进位表示

【举例】 将数据 030201 写入用户自定义区域

```
cv181x/cv180x# efusew USER 030201
Write eFuse USER(0) with:
00000000: 03 02 01                                           ...
cv181x/cv180x# efuser USER
00000000: 03 02 01 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
```

(下页继续)

(续上页)

```
00000020: 00 00 00 00 00 00 00 00                        ........
cv181x/cv180x#
```

2.4 eFuse API 参考

eFuse API 位于 CIPHER 模块，提供以下 API:

- CVI_EFUSE_GetSize: 查询 eFuse 区域大小。
- CVI_EFUSE_Read: 读取 eFuse 区域。
- CVI_EFUSE_Write: 写入 eFuse 区域。
- CVI_EFUSE_EnableSecureBoot: 使能安全启动。
- CVI_EFUSE_IsSecureBootEnabled: 查询安全启动状态。
- CVI_EFUSE_EnableFastBoot: 使能快速启动。
- CVI_EFUSE_IsFastBootEnabled: 查询快速启动状态。
- CVI_EFUSE_Lock: 锁定 eFuse 区域。
- CVI_EFUSE_IsLocked: 查询 eFuse 区域是否被锁定。

2.4.1 CVI_EFUSE_GetSize

【描述】

查询 eFuse 区域大小。

【语法】

```
CVI_S32 CVI_EFUSE_GetSize(CVI_EFUSE_AREA_E area, CVI_U32 *size);
```

【参数】

参数名称	描述	输入/输出
area	指定 eFuse 区域	输入
size	eFuse 区域大小 (单位：字节)	输出

【返回值】

返回值	描述
>= 0	成功
< 0	参考错误码

【需求】

- 头文件：cvi_type.h cvi_unf_cipher.h

- 库文件：libcipher.a

【注意】

无。

【举例】

参考 sample_efuse.c 。

2.4.2 CVI_EFUSE_Read

【描述】

读取 eFuse 区域。

【语法】

```
CVI_S32 CVI_EFUSE_Read(CVI_EFUSE_AREA_E area, CVI_U8 *buf, CVI_U32 buf_size);
```

【参数】

参数名称	描述	输入/输出
area	指定 eFuse 区域	输入
buf	用于存放 eFuse 数据	输出
buf_size	数据的长度（单位：字节）	输入

【返回值】

返回值	描述
>= 0	成功
< 0	参考错误码

【需求】

- 头文件：cvi_type.h cvi_unf_cipher.h
- 库文件：libcipher.a

【注意】

无。

【举例】

参考 sample_efuse.c 。

2.4.3 CVI_EFUSE_Write

【描述】

写入 eFuse 区域。

【语法】

```
CVI_S32 CVI_EFUSE_Write(CVI_EFUSE_AREA_E area, const CVI_U8 *buf, CVI_U32 buf_
↪size);
```

【参数】

参数名称	描述	输入/输出
area	指定 eFuse 区域	输入
buf	用于写入 eFuse 数据	输入
buf_size	数据的长度（单位：字节）	输入

【返回值】

返回值	描述
>= 0	成功
< 0	参考错误码

【需求】

- 头文件：cvi_type.h cvi_unf_cipher.h
- 库文件：libcipher.a

【注意】

无。

【举例】

参考 sample_efuse.c 。

2.4.4 CVI_EFUSE_EnableSecureBoot

【描述】

使能安全启动。

【语法】

```
CVI_S32 CVI_EFUSE_EnableSecureBoot(void);
```

【参数】

无。

【返回值】

返回值	描述
>= 0	安全启动已使能
< 0	参考错误码

【需求】

- 头文件：cvi_type.h cvi_unf_cipher.h
- 库文件：libcipher.a

【注意】

无。

【举例】

参考 sample_efuse.c 。

2.4.5 CVI_EFUSE_IsSecureBootEnabled

【描述】

判断安全启动是否已使能。

【语法】

```
CVI_S32 CVI_EFUSE_IsSecureBootEnabled(void);
```

【参数】

无。

【返回值】

返回值	描述
> 0	安全启动已使能
0	安全启动尚未使能
< 0	参考错误码

【需求】

- 头文件：cvi_type.h cvi_unf_cipher.h
- 库文件：libcipher.a

【注意】

无。

【举例】

参考 sample_efuse.c 。

2.4.6 CVI_EFUSE_EnableFastBoot

【描述】

使能快速启动。

【语法】

```
CVI_S32 CVI_EFUSE_EnableFastBoot(void);
```

【参数】

无。

【返回值】

返回值	描述
0	快速启动已使能
< 0	参考错误码

【需求】

- 头文件：cvi_type.h cvi_unf_cipher.h
- 库文件：libsys.a

【注意】

无。

【举例】

参考 sample_fastboot.c 。

注意： 快速启动使能后无法再变更

2.4.7 CVI_EFUSE_IsFastBootEnabled

【描述】

判断快速启动是否已使能。

【语法】

```
CVI_S32 CVI_EFUSE_IsFastBootEnabled(void);
```

【参数】

无。

【返回值】

返回值	描述
0	快速启动已使能
< 0	快速启动尚未使能

【需求】

- 头文件：cvi_type.h cvi_unf_cipher.h
- 库文件：libsys.a

【注意】

无。

【举例】

参考 sample_efuse.c 。

2.4.8 CVI_EFUSE_Lock

【描述】

锁定 eFuse 区域。

【语法】

```
CVI_S32 CVI_EFUSE_Lock(CVI_EFUSE_LOCK_E lock);
```

【参数】

参数名称	描述	输入/输出
area	指定要锁定的 eFuse 区域	输入

【返回值】

返回值	描述
>= 0	指定的 eFuse 分区已锁定
< 0	参考错误码

【需求】

- 头文件：cvi_type.h cvi_unf_cipher.h
- 库文件：libcipher.a

【注意】

无。

【举例】

参考 sample_efuse.c 。

2.4.9 CVI_EFUSE_IsLocked

【描述】

查询 eFuse 区域是否被锁定。

【语法】

```
CVI_S32 CVI_EFUSE_IsLocked(CVI_EFUSE_LOCK_E lock);
```

【参数】

参数名称	描述	输入/输出
area	指定要锁定的 eFuse 区域	输入

【返回值】

返回值	描述
> 0	指定的 eFuse 分区已锁定
0	指定的 eFuse 分区尚未锁定
< 0	参考错误码

【需求】

- 头文件：cvi_type.h cvi_unf_cipher.h
- 库文件：libcipher.a

【注意】

无。

【举例】

参考 sample_efuse.c 。

2.5 数据类型

相关数据类型、数据结构定义如下：

- CVI_EFUSE_AREA_E：定义 eFuse 区域
- CVI_EFUSE_LOCK_E：定义 eFuse 区域锁定

2.5.1 CVI_EFUSE_AREA_E

【说明】

定义 eFuse 区域。

【定义】

```
typedef enum {
  CVI_EFUSE_AREA_USER,
  CVI_EFUSE_AREA_DEVICE_ID,
  CVI_EFUSE_AREA_HASH0_PUBLIC,
  CVI_EFUSE_AREA_LOADER_EK,
  CVI_EFUSE_AREA_DEVICE_EK,
  CVI_EFUSE_AREA_LAST
} CVI_EFUSE_AREA_E;
```

【成员】

成员名称	描述
CVI_EFUSE_AREA_USER	用户自定义区域
CVI_EFUSE_AREA_DEVICE_ID	设备序号区域
CVI_EFUSE_AREA_HASH0_PUBLIC	secureboot RSA 公钥 HASH 值区域
CVI_EFUSE_AREA_LOADER_EK	secureboot AES 密钥区域
CVI_EFUSE_AREA_DEVICE_EK	device_ek 区域
CVI_EFUSE_AREA_LAST	结束标识

【注意事项】

无。

【相关数据类型及接口】

- CVI_EFUSE_GetSize
- CVI_EFUSE_Read
- CVI_EFUSE_Write

2.5.2 CVI_EFUSE_LOCK_E

【说明】

定义 eFuse 区域锁定。

【定义】

```
typedef enum {
  CVI_EFUSE_LOCK_HASH0_PUBLIC,
  CVI_EFUSE_LOCK_LOADER_EK,
  CVI_EFUSE_LOCK_DEVICE_EK,
  CVI_EFUSE_LOCK_LAST
} CVI_EFUSE_LOCK_E;
```

【成员】

成员名称	描述
CVI_EFUSE_LOCK_HASH0_PUBLIC	锁定 secureboot RSA 公钥 hash 值区域
CVI_EFUSE_LOCK_LOADER_EK	锁定 secureboot AES 密钥区域
CVI_EFUSE_LOCK_DEVICE_EK	锁定 device_ek 区域
CVI_EFUSE_LOCK_LAST	结束标识

【注意事项】

无。

【相关数据类型及接口】

- CVI_EFUSE_Lock
- CVI_EFUSE_IsLocked