



CV186AH Secure OTP 使用手册

Version: 0.1

Release date: 2023/12

©2022 北京晶视智能科技有限公司
本文件所含信息归北京晶视智能科技有限公司所有。
未经授权，严禁全部或部分复制或披露该等信息。

目录

1	声明	2
2	Secure OTP 使用指南	3
2.1	Secure OTP 概述	3
2.1.1	物理组织与划分	3
2.1.2	安全 block & 非安全 block	3
2.1.3	系统预留 segment & 用户自定义 segment	3
2.2	Secure OTP u-boot 命令参考	4
2.2.1	otp version	4
2.2.2	otp otp2_r	4
2.2.3	otp otp2_w	5
2.2.4	otp otp2_lock	6
2.2.5	otp otp2_isLocked	6
2.2.6	otp otp3_r	7
2.2.7	otp otp3_w	7
2.2.8	otp config_r	8
2.2.9	otp config_w	9
2.3	Secure OTP API 参考	9
2.3.1	CVI_OTP2_Read	9
2.3.2	CVI_OTP2_Write	10
2.3.3	CVI_OTP3_Read	11
2.3.4	CVI_OTP3_Write	11
2.4	数据类型	12
2.4.1	CVI_OTP_AREA_E	12
2.4.2	CVI_OTP_SECUREBOOT_E	13
2.5	安全启动 Secure OTP 设置流程	13
2.5.1	查看密钥内容	13
2.5.2	写入密钥	14
2.5.3	使能安全启动	14

修订记录

Revision	Date	Description
0.1	2023-11-01	Initial

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 Secure OTP 使用指南

2.1 Secure OTP 概述

Secure otp 是一种一次性非易失性存储器 (OTP NVM/one time program NVM), 有以下硬件特征:

1. 断电数据不丢失
2. 存储器每一位只能从 0 (默认值) 写成 1, 无法重置

2.1.1 物理组织与划分

处理器内部集成 32Kb Secure OTP 空间, 以 32bits (4 Bytes) 为操作单元, 称作 1 行, 32 行为一个 segment (1Kb), 总共 32 个 segment。

2.1.2 安全 block & 非安全 block

硬件上 Secure OTP 分成两个 block

1. OTP3 (安全 block), 只能在安全 world 访问, 预分配 4 个 segment (4Kb), segment id 为 0 ~ 3。
2. OTP2 (非安全 block), 安全 world 和非安全 world 都能访问, 预分配 28 个 segment (28Kb), segment id 为 0 ~ 27。

2.1.3 系统预留 segment & 用户自定义 segment

每个 block 分为

1. IC 预留 segment, OTP3 预留 2 个 segment (2Kb), segment id 为 2 和 3, OTP2 预留 1 个 segment (1Kb), segment id 为 0。
2. 用户自定义 segment, OTP3 有 2 个 segment (2Kb), segment id 为 0 和 1, OTP2 有 27 个 segment (27Kb), segment id 为 1 ~ 27。

注意： IC 预留 segment 是 IC 自用的，里面包含了 IC 的各种重要配置，不能随意配置，可能会导致 IC 启动失败或者功能异常。

2.2 Secure OTP u-boot 命令参考

u-boot 提供以下命令操作 Secure OTP:

```
athena2# otp
otp - OTP sub-system

Usage:
otp version - otp version
otp otp2_r <segment> [line] - read otp2 segment[1~27] line[0~31], line is option
otp otp2_w <segment> [line] <Hex value/Hex string> - program otp2 segment[1~27] line[0~31], line is option
otp otp2_lock <segment> - lock otp2 segment[1~27], WARM: it can't unlock
otp otp2_isLocked <segment> - is otp2 segment[1~27] Locked
otp otp3_r <segment> [line] - read otp3 segment[0~1] line[0~31], line is option
otp otp3_w <segment> [line] <Hex value/Hex string> - program otp3 segment[0~1] line[0~31], line is option
otp config_r <area string> ... - read config
otp config_w <area string> ... - program config
```

2.2.1 otp version

【描述】 读取 Secure OTP 版本号。

【语法】 otp version

【参数】 无

【举例】

```
athena2# otp version
version: 30d70003
athena2#
```

2.2.2 otp otp2_r

【描述】 读取 Secure OTP OTP2 block。

【语法】 otp otp2_r <segment> [line]

【参数】

参数名称	描述
segment	segment id, range 1 ~ 27
line	行号 (0 ~ 31), 可选参数, 如果未输入, 则 dump 整个 segment

【举例】 读取 OTP2 segment 1

```
athena2# otp otp2_r 1
00000000: 00 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 00 00 00 00 00 00 00 00 .....
00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
athena2#
```

读取 OTP2 segment 1 line 0

```
athena2# otp otp2_r 1 0
Read otp2 segment[0x1] addr[0x0000] : 0x00000000
athena2#
```

2.2.3 otp otp2_w

【描述】 写 Secure OTP OTP2 block。

【语法】 otp otp2_w <segment> [line] <Hex value/Hex string>

【参数】

参数名称	描述
segment	segment id, range 1 ~ 27
line	行号 (0 ~ 31), 可选参数, 如果未输入, 则使用参数 Hex string 写整个 segment
Hex value/Hex string	已输入 line id, 需输入 16 进制数据 (Hex value), 例如: 0x12345678, 写 segment 的某一行; 未输入 line id, 需输入 16 进制数据字符串 (Hex string), 例如: 1A2B3C4D5E6F7A8B, 从第一行开始写 segment

【举例】 将 0x12345678 写入 OTP2 segment 1 line 0

```
athena2# otp otp2_w 1 0 0x12345678
athena2# otp otp2_r 1 0
Read otp2 segment[0x1] addr[0x0000] : 0x12345678
athena2#
```

写 OTP2 第二个 segment

```

athena2# otp otp2_w 1 1A2B3C4D5E6F7A8B
00000000: 1a 2b 3c 4d 5e 6f 7a 8b                .+<M^oz.
athena2# otp otp2_r 1
00000000: 1a 2b 3c 4d 5e 6f 7a 8b 00 00 00 00 00 00 00 00 .+<M^oz.....
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 00 00 00 00 00 00 00 00 .....
00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
athena2#

```

2.2.4 otp otp2_lock

【描述】 锁定 otp2 的 segment，该 segment 变为只读，并且无法恢复。

【语法】 otp otp2_lock <segment>

【参数】

参数名称	描述
segment	segment id, range 1 ~ 27

【举例】 锁定 otp2 segment 1

```

athena2# otp otp2_lock 1
athena2#

```

2.2.5 otp otp2_isLocked

【描述】 判断 otp2 的 segment 是否已被锁定。

【语法】 otp otp2_isLocked <segment>

【参数】

参数名称	描述
segment	segment id, range 1 ~ 27

【举例】 判断 otp2 segment 是否已被锁定

```

athena2# otp otp2_isLocked 1
segment 1 is locked
athena2# otp otp2_isLocked 2
segment 2 unLock
athena2#

```


2.2.6 otp otp3_r

【描述】 读取 Secure OTP OTP3 block。

【语法】 `otp otp3_r <segment> [line]`

【参数】

参数名称	描述
segment	segment id, range 0 ~ 1
line	行号 (0 ~ 31), 可选参数, 如果未输入, 则 dump 整个 segment

【举例】 读取 OTP3 segment 0

```
athena2# otp otp3_r 0
00000000: 00 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 00 00 00 00 00 00 00 00 .....
00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
athena2#
```

读取 OTP3 segment 0 line 0

```
athena2# otp otp3_r 0 0
Read otp3 segment[0x0] addr[0x0000] : 0x00000000
athena2#
```

2.2.7 otp otp3_w

【描述】 写 Secure OTP OTP3 block。

【语法】 `otp otp3_w <segment> [line] <Hex value/Hex string>`

【参数】

参数名称	描述
segment	segment id, range 0 ~ 1
line	行号 (0 ~ 31), 可选参数, 如果未输入, 则使用参数 Hex string 写整个 segment
Hex value/Hex string	已输入 line, 需输入 16 进制数据 (Hex value), 例如: 0x12345678, 写 segment 的某一行; 未输入 line, 需输入 16 进制数据字符串 (Hex string), 例如: 1A2B3C4D5E6F7A8B, 从第一行开始写某个 segment

【举例】 将 0x12345678 写入 OTP3 segment 0 line 0

```

athena2# otp otp3_w 0 0 0x12345678
athena2# otp otp3_r 0 0
Read otp3 segment[0x0] addr[0x0000] : 0x12345678
athena2#

```

写 OTP2 segment 0

```

athena2# otp otp3_w 0 1A2B3C4D5E6F7A8B
00000000: 1a 2b 3c 4d 5e 6f 7a 8b          .+<M^oz.
athena2# otp otp3_r 0
00000000: 1a 2b 3c 4d 5e 6f 7a 8b 00 00 00 00 00 00 00 00 .+<M^oz.....
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 00 00 00 00 00 00 00 00 .....
00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
athena2#

```

2.2.8 otp config_r

【描述】 读取 Secure OTP IC 预留配置。

【语法】 otp config_r ...

【参数】

参数名称	描述
...	根据功能不同，参数不固定

【举例】 读取 HASH0_PUBLIC 值

```

athena2# otp config_r HASH0_PUBLIC
00000000: 00 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 .....
athena2#

```

读取 secure boot 配置

```

athena2# otp config_r SECUREBOOT
secure boot is disable
athena2#

```

2.2.9 otp config_w

【描述】 写 Secure OTP IC 预留配置。

【语法】 otp config_w ...

【参数】

参数名称	描述
...	根据功能不同，参数不固定

【举例】 写 HASH0_PUBLIC 数据

```
athena2# otp config_w HASH0_PUBLIC 1A2B3C4D5E6F7A8B
00000000: 1a 2b 3c 4d 5e 6f 7a 8b          .+<M^oz.
athena2#
```

开启 secure boot 验签

```
athena2# otp config_w SECUREBOOT 1
athena2#
```

2.3 Secure OTP API 参考

Secure OTP API 位于 MISC 模块，提供以下 API:

- CVI_OTP2_Read: 读取 OTP2 block。
- CVI_OTP2_Write: 写入 OTP2 block。
- CVI_OTP3_Read: 读取 OTP3 block。
- CVI_OTP3_Write: 写入 OTP3 block。

2.3.1 CVI_OTP2_Read

【描述】

读取 OTP2 block。

【语法】

```
CVI_S32 CVI_OTP2_Read(CVI_U32 segment, CVI_U32 addr, CVI_U32 size, CVI_U32 *value);
```

【参数】

参数名称	描述	输入/输出
segment	指定 segment id	输入
addr	指定 line id	输入
size	行个数（单位：行），1 行等于 4 Bytes	输入
value	读取到的值	输出

【返回值】

返回值	描述
<code>== 0</code>	成功
<code>< 0</code>	读取失败

【需求】

- 头文件: `cvi_misc.h`
- 库文件: `libmisc.a`

【注意】

无。

【举例】

参考 `cvi_sample_otp.c`。

2.3.2 CVI_OTP2_Write

【描述】

写入 OTP2 block。

【语法】

```
CVI_S32 CVI_OTP2_Write(CVI_U32 segment, CVI_U32 addr, CVI_U32 size, CVI_U32 *value);
```

【参数】

参数名称	描述	输入/输出
<code>segment</code>	指定 segment id	输入
<code>addr</code>	指定 line	输入
<code>size</code>	行个数（单位：行）	输入
<code>value</code>	写入的数据	输入

【返回值】

返回值	描述
<code>= 0</code>	成功
<code>< 0</code>	参考错误码

【需求】

- 头文件: `cvi_misc.h`
- 库文件: `libmisc.a`

【注意】

无。

【举例】

参考 `cvi_sample_otp.c` 。

2.3.3 CVI_OTP3_Read

【描述】

读取 OTP3 block。

【语法】

```
CVI_S32 CVI_OTP3_Read(CVI_U32 segment, CVI_U32 addr, CVI_U32 size, CVI_U32 *value);
```

【参数】

参数名称	描述	输入/输出
segment	指定 segment id	输入
addr	指定 line id	输入
size	行个数（单位：行），1 行等于 4 Bytes	输入
value	读取到的值	输出

【返回值】

返回值	描述
<code>== 0</code>	成功
<code>< 0</code>	读取失败

【需求】

- 头文件： `cvi_misc.h`
- 库文件： `libmisc.a`

【注意】

无。

【举例】

参考 `cvi_sample_otp.c` 。

2.3.4 CVI_OTP3_Write

【描述】

写入 OTP3 block。

【语法】

```
CVI_S32 CVI_OTP3_Write(CVI_U32 segment, CVI_U32 addr, CVI_U32 size, CVI_U32 *value);
```

【参数】

参数名称	描述	输入/输出
segment	指定 segment id	输入
addr	指定 line	输入
size	行个数（单位：行）	输入
value	写入的数据	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_misc.h
- 库文件：libmisc.a

【注意】

无。

【举例】

参考 cvi_sample_otp.c 。

2.4 数据类型

相关数据类型、数据结构定义如下：

- CVI_OTP_AREA_E：定义 OTP 区域
- CVI_OTP_SECUREBOOT_E：定义 Secureboot 配置

2.4.1 CVI_OTP_AREA_E

【说明】

定义 OTP 区域。

【定义】

```
typedef enum {  
    CVI_OTP_AREA_HASH0_PUBLIC = 0,  
    CVI_OTP_AREA_LOADER_EK,  
    CVI_OTP_AREA_LAST  
} CVI_OTP_AREA_E;
```

【成员】

成员名称	描述
CVI_OTP_AREA_HASH0_PUBLIC	secureboot RSA 公钥 HASH 值区域
CVI_OTP_AREA_LOADER_EK	secureboot AES 密钥区域
CVI_OTP_AREA_LAST	结束标识

【注意事项】

无。

【相关数据类型及接口】

2.4.2 CVI_OTP_SECUREBOOT_E

【说明】

定义 Secureboot 配置项

【定义】

```
typedef enum {  
    CVI_OTP_SECUREBOOT_DISABLE = 0,  
    CVI_OTP_SECUREBOOT_SIGN,  
    CVI_OTP_SECUREBOOT_SIGN_ENCRYPT,  
} CVI_OTP_SECUREBOOT_E;
```

【成员】

成员名称	描述
CVI_OTP_SECUREBOOT_DISABLE	disable secureboot
CVI_OTP_SECUREBOOT_SIGN	enable secureboot 验签
CVI_OTP_SECUREBOOT_SIGN_ENCRYPT	enable secureboot 验签和解密

【注意事项】

无。

【相关数据类型及接口】

2.5 安全启动 Secure OTP 设置流程

注意： Secure OTP 每一位写入 1 后无法重置，写入前请注意

2.5.1 查看密钥内容

在 PC 上查看密钥内容：

```
# 查看AES加解密密钥
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:978bc2031b9377dadb4c7c34467ee985806a63a3ac8ee293a3f0eddc2b789d8
Host$ .....
```

- KPUB_HASH: 后面的字符串就是所需 sha256 值

2.5.2 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# otp config_w 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# otp config_w HASH0_PUBLIC_
→978bc2031b9377dadb4c7c34467ee985806a63a3ac8ee293a3f0eddc2b789d8
```

2.5.3 使能安全启动

1. enable 验签流程

```
u-boot# otp config_w SECUREBOOT 01
```

2. enable 验签和解密流程

```
u-boot# otp config_w SECUREBOOT 02
```

注意:

1. 安全启动使能后无法再变更，密钥和开关写入前请注意数据正确
2. 使能安全启动需要和 OTP 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