

CM-IOT FOTA 还原库适配使用说明

Version: 1.3

Date: 2023-04-14

**iot.10086.cn**

修订记录

编号	版本号	修订日期	修订描述	修改人
1	V1.0	2023-02-01	初稿	OneOS Team
2	V1.1	2023-02-20	增加 bootloader 适配	OneOS Team
3	V1.2	2023-02-23	更新平台使用说明	OneOS Team
4	V1.3	2023-04-14	更新 fota_boot V2.1.0	OneOS Team

目 录

1 背景	3
2 适配说明	3
2.1 Bootloader 适配	3
2.1.1 资源.....	3
2.1.2 数据结构.....	3
2.1.3 接口描述.....	4
2.1.4 返回码表.....	6
2.1.5 适配流程.....	7
2.1.6 参数配置.....	9
3 平台使用说明	10
3.1 创建产品	10
3.2 制作版本包	14
3.3 制作差分包	15
3.4 删除版本	20
4 注意事项	20
5 参考资料	21
6 FAQ	21

1 背景

CMIOT FOTA 主要包含 OTA 组件，OTA_BOOTLOADER 和云平台几个部分，各个模块均附有详细的接口及使用说明文档。该文档从初次使用 CMIOT FOTA 的角度，一步一步介绍全流程操作及注意事项，以及各个模块适配注意事项，旨在帮助用户快速的体验 FOTA 升级服务。文档目前仅适用于采用自研算法和 CMS 平台的升级服务场景。

2 适配说明

2.1 Bootloader 适配

2.1.1 资源

在 bootloader 阶段，需要依赖的资源有：

- ①内存，用于存放还原过程中的数据，通常内存空间越大越好；
- ②堆栈，程序必须运行在堆栈已经完成正确初始化的 C 环境下；
- ③flash 操作接口，包括下载及升级相关的存储介质的读/写/擦除接口。

2.1.2 数据结构

```
typedef struct ota_fal_flash_ops
{
    ota_int32 (*init)(void);

    ota_int32 (*read)(ota_uint32 addr, ota_uint8 *buf, ota_size_t size);

    ota_int32 (*write)(ota_uint32 addr, const ota_uint8 *buf, ota_size_t size);

    ota_int32 (*erase)(ota_uint32 addr, ota_size_t size);

} ota_fal_flash_ops_t;
```

```
typedef struct ota_fal_flash
{

```

```
ota_int8 name[FAL_NAME_MAX];          /* flash 名称 */

ota_size_t size;                        /* flash 大小 */

ota_size_t erase_size;                  /* flash 最小擦除单位 */

ota_fal_flash_ops_t ops;                /* flash 读写擦操作函数 */

} ota_fal_flash_t;

typedef struct ota_fal_part_info
{
    ota_int8 name[FAL_NAME_MAX];         /* 分区 名称 */
    ota_int8 flash_name[FAL_NAME_MAX];   /* 分区 所在 flash 名称 */
    ota_uint32 address;                   /* 分区 地址 */
    ota_uint32 size;                      /* 分区 大小 */
    ota_uint32 part_type;                 /* 分区 类型 */
} ota_fal_part_info_t;

typedef struct ota_fal_map
{
    ota_uint32 logic_addr;                /* 分区 逻辑地址 */
    ota_uint32 phy_addr;                  /* 分区 物理地址 */
} ota_fal_map_t;
```

2.1.3 接口描述

(1) *void ota_printf(const ota_int8 *fmt, ...)*

功能描述：LOG 格式化输出接口，升级过程中的 LOG 信息，以及返回值通过该接口输出。

该接口需用户实现

(2) *void ota_update_proc_bar(ota_int32 percentage)*

功能描述：升级进度条显示

输入参数：ota_int32 percentage： 当前升级进度：0~100

该接口需用户实现，在升级过程中被调用，不需要应用代码主动调用。通常是将进度条通过屏幕显示，因此需要在该函数中实现屏幕显示进度条的功能。

(3) *ota_int32 ota_heap_init(ota_uint8 *buf, ota_size_t buf_size, ota_size_t block_size)*

功能描述：堆空间初始化接口

输入参数：ota_uint8 *buf：用于 OTA 还原升级的内存起始地址；

ota_size_t buf_size：用于 OTA 还原升级的内存大小；

ota_size_t block_size：内存块大小；

返回值： 0：成功，小于 0：失败

当 boot loader 中堆不可用时，可以使用该接口使用内置堆管理。提供给还原升级的内存应尽可能大，除了系统栈所需内存，和预留静态占用的内存外，剩余内存可以都用于还原升级。该内存可以通过定义一个静态数组的方式进行预分配。为了提高内存的利用率，block_size 不宜过大，buf_size 需与 CM_IOT_FOTA_AVAILABLE_RAM_SIZE 保持一致，内置堆管理模块会对 buf_size 和 block_size 进行合理性检查，当初始化返回失败时，根据提示 log 进行调整即可。

(4) *ota_int32 ota_fal_init(ota_fal_flash_t *flash_tab, ota_size_t flash_tab_size, ota_fal_part_info_t *part_tab, ota_size_t part_tab_size)*

功能描述：注册分区信息

输入参数：ota_fal_flash_t *flash_tab：flash 信息，包含 flash 名称，大小，擦除大小，读写擦操作函数。

ota_size_t flash_tab_size：flash 列表大小

ota_fal_part_info_t *part_tab：分区信息，包含分区名称，分区所

在 flash 名称，分区起始地址，分区大小和分区类型

ota_size_t part_tab_size: 分区信息表大小

返回值: 0: 成功, 小于 0: 失败

当需要进行地址映射时, 分区表中的地址应当使用逻辑地址。

(5) ota_int32 ota_fal_map_init(ota_fal_map_t *map_table, ota_size_t map_table_size)

功能描述: 分区地址映射初始化

输入参数: ota_fal_map_t *map_table: 逻辑地址与物理地址映射表

ota_size_t map_table_size: 逻辑地址与物理地址映射表大小

返回值: 0: 成功, 小于 0: 失败

若某些分区使用逻辑地址, 某些分区使用物理地址, 则使用物理地址的分区, 逻辑地址与物理地址保持一致即可。

(6) ota_int32 ota_start_up(ota_uint32 try_times)

功能描述: 启动升级

输入参数: ota_uint32 try_time: 升级重试次数。建议至少重试一次

返回值: 升级结果返回码

2.1.4 返回码表

表 1-1 返回码表

状态码	**说明**
1000	升级成功
1001	不需要升级
1002	内存不足
1003	无效升级包
1004	前置版本校验错误
1005	升级包类型错误
1006	升级包不存在
1007	DL 分区空间不足
1008	未找到分区
1009	分区读取错误

1010	分区写入错误
1011	分区擦除错误
1012	分区数据备份错误
1013	分区数据还原错误
1014	流信息初始化错误
1015	头信息初始化错误
1016	解压头错误
1017	解压数据体错误
1018	还原块长度错误
1019	差分包数据校验错误
1501	新 APP 块数据校验错误
1502	覆盖线差值计算错误
1503	构建新数据错误
1504	DL 备份区域擦除失败
1505	解析块还原编号错误
1506	解析块覆盖线数量错误
1507	新 APP 分区数据校验错误

2.1.5 适配流程

(1) Flash 操作接口

```
ota_int32 (*init)(void);
```

flash 的初始化一般在 bootloader 硬件初始化阶段已完成,init 接口为 NULL; 若需要在 fal 初始化中完成,则可以在 init 接口中完成 flash 的硬件初始化。成功返回 0, 失败返回-1。

```
ota_int32 (*read)(ota_uint32 addr, ota_uint8 *buf, ota_size_t size);
```

数据读取接口, 成功返回读取长度, 失败返回-1

```
ota_int32 (*write)(ota_uint32 addr, const ota_uint8 *buf, ota_size_t size);
```

数据写入接口, 成功返回写入长度, 失败返回-1

```
ota_int32 (*erase)(ota_uint32 addr, ota_size_t size);
```

数据擦除接口, 其中 addr 按 block 对齐, size 不是 block 的整数倍时, 按照最小对齐 block 擦除, 成功返回 0, 失败返回-1

将操作函数, flash 名称, 容量, 块大小(擦除大小)填写至 ota_fal_flash_t 类型的表中。如果有多个不同的 flash 设备, 根据实际情况, 实现相应的操作函

数，并将硬件信息添加到 flash 表中即可。

(2) 填写分区表

ota_fal_part_info_t 类型的分区表中包含了需要升级的“APP”分区以及“download”分区，除了 download 分区名称是固定的，其他“APP”的名称可以任意命名。每个分区，包含了分区名称，所在 flash 名称，起始地址（如果使用了逻辑地址需填写逻辑地址），分区大小，分区类型（“APP”必须为 FAL_UPGRADE_PART 类型，“download”必须为 FAL_DOWNLOAD_PART 类型），分区名称不可重复。

(3) 配置参数

根据实际情况，修改 oncos_config.h 中的配置参数（见 2.1.6 参数配置章节）

(4) 初始化地址映射（必要时）

若使用逻辑地址，需要填写 ota_fal_map_t 类型的地址映射表，表中必须包含分区表中所有的分区地址映射信息，若某些分区不使用逻辑地址，则逻辑地址与物理地址保持一致。通过 ota_fal_map_init 接口完成地址映射初始化。

(5) 初始化分区信息

将 flash 表和分区表作为参数，调用 ota_fal_init 接口完成分区信息初始化。

(6) 初始化堆空间（必要时）

若使用内置堆管理，则需要调用 ota_heap_init 接口进行堆空间初始化。

(7) 检查升级状态

在 bootloader 启动时，读取 download 分区最后一个块起始存放的前 2 个成员数值，判断是否需要进入还原升级：

```
struct cmiot_update_state
{
    unsigned char    initied;

    unsigned short update_result;

};
```


update_result 等于 100，表示需要进入还原升级

(8) 启动升级

调用 ota_start_up 接口，启动升级。对比返回码，可以查看升级结果以及错误码

2.1.6 参数配置

在 oneos_config.h 中，用户可配置的 FOTA 相关的几个重要参数如下：

(1) 可用内存大小 CMIOT_FOTA_AVAILABLE_RAM_SIZE

示例：`#define CMIOT_FOTA_AVAILABLE_RAM_SIZE 0x80000`

支持十进制/十六进制，行尾注释前必须加空格。该宏定义了 fota 还原升级时真实可用的内存大小，在条件允许的情况下，内存越大越好。

(2) Download 分区大小 CMIOT_UPDATE_SIZE

示例：`#define CMIOT_UPDATE_SIZE 0xA0000`

支持十进制/十六进制，行尾注释前必须加空格。该宏定义了差分包下载空间（即 download 分区）大小，在制作差分包时会对其进行校验，如果差分包的实际大小超过该值定义的大小，会提示制作差分包失败，用户需提前规划好 download 分区大小，不允许升级版本时再作改变。如果空间有限，可以采取多次少量进行升级。

(3) Flash 擦除大小 CMIOT_DEFAULT_SECTOR_SIZE

示例：`#define CMIOT_DEFAULT_SECTOR_SIZE 0x10000`

支持十进制/十六进制，行尾注释前必须加空格。该宏定义了 FLASH 的块擦除大小，必须和实际 flash 的块擦除大小一致。制作差分包和还原升级，会以此作为计算和存储的基本单位。

(4) 固件版本 CMIOT_FIRMWARE_VERSION

示例：`#define CMIOT_FIRMWARE_VERSION "1.0"`

字符串，行尾注释前必须加空格。该宏定义了固件版本，升级版本时，版本号需同时进行迭代。

(5) 下载块大小 CMIOT_DEFAULT_SEGMENT_SIZE_INDEX

示例: `#define CMIOT_DEFAULT_SEGMENT_SIZE_INDEX 3`

十进制/十六进制, 行尾注释前必须加空格。该宏定义了下载差分包时默认的 buff 大小, 该值越大, 单次下载的数据包长度更大, 总的下载时间会减少, 最大可设置为 5。

(6) 升级文件 CMIOT_FOTA_FILENAME_x, CMIOT_FOTA_ADDRESS_x

示例: `#define CMIOT_FOTA_FILENAME1 "mmi_res_240x240_child_circular.bin"`

`#define CMIOT_FOTA_ADDRESS1 0x90000004`

CMIOT_FOTA_FILENAME_x 文件名为字符串, 行尾注释前必须加空格。与实际文件名称对应, 制作版本包时, 会根据该名称检索文件。如果是单 bin, 则 x 为空, 如果是多 bin, x=1, 2..., 文件名必须是以常见的二进制文件结尾, 如 '*.bin', '*.img' 等, 若差分文件为其他后缀, 可以在文件末尾添加 '.bin'。CMIOT_FOTA_ADDRESS_x 文件地址, 必须是十六进制格式, 如果使用逻辑地址, 也应当符合地址的特征, 如长度为 8 位。

(7) 全量/差分升级 CMIOT_FOTA_USE_FULL_x

示例: `#define CMIOT_FOTA_USE_FULL2`

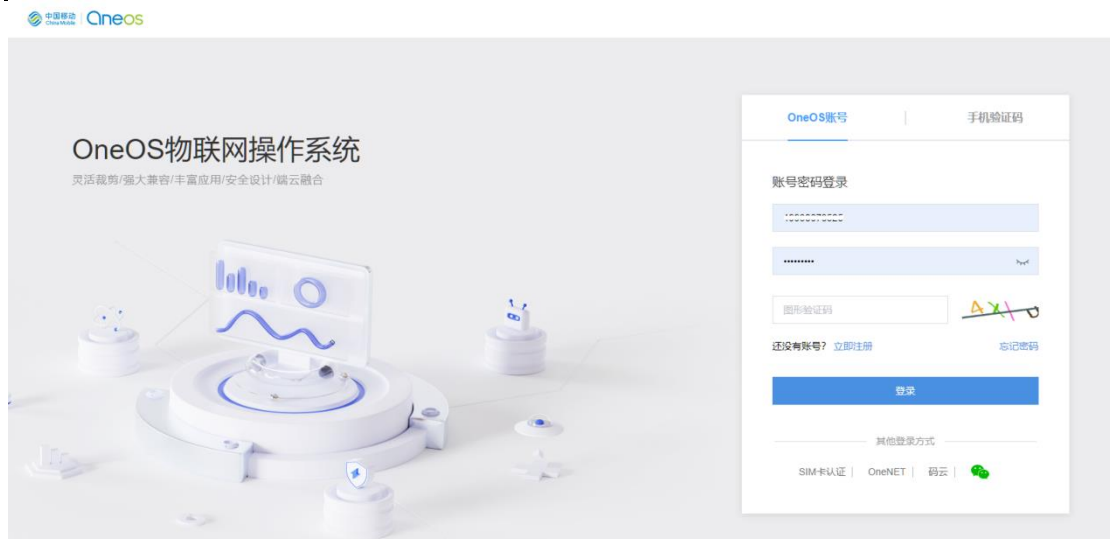
CMIOT_FOTA_USE_FULL_x, 表示对应的 CMIOT_FOTA_FILENAME_x 升级模式为全量升级。默认为差分升级。当目标版本需要做全量升级时, 可定义该宏。

3 平台使用说明

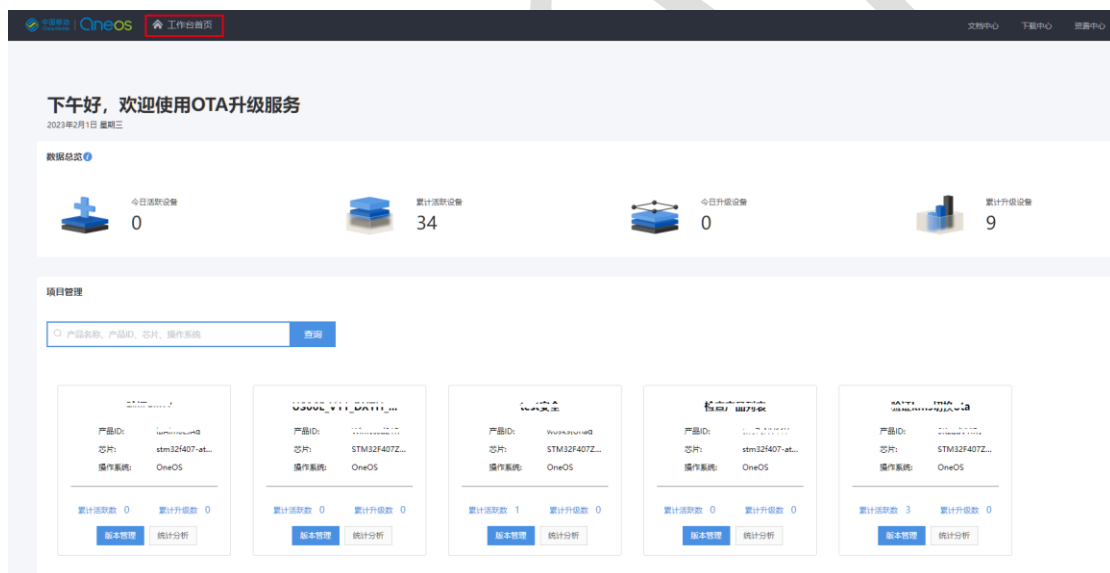
3.1 创建产品

①登录 cms 工作台 OTA 升级服务平台:

<https://os.iot.10086.cn/user-center#/user/login>



②进入工作台首页



③依次进入【产品开发】【产品管理】选项，点击【新建产品】



④完善产品信息

产品开发 / 产品管理 / 新建产品

新建产品

新建一个产品，配置需要使用的CMS服务

1 基础配置

2 完成

* 产品名称:

* 操作系统: ☐ OneOS ☐ Linux

* 芯片型号:

* 通信方式: ☐ 2G/3G/4G/5G ☐ NB-IoT ☐ WiFi ☐ Zigbee ☐ 蓝牙 ☐ 其他

* 协议选择: ☒ mqtt

产品描述:

⑤点击【完成创建】【完成】，即完成产品创建。记录产品 ID 和产品密钥备用。

1 基础配置

2 完成

* 产品名称:

* 操作系统: ☒ OneOS ☐ Linux

* 芯片型号:

* 支持服务: ☒ License授权管理服务 ☒ 远程调试服务 ☒ OnePos定位服务 ☒ OTA升级服务

【RTC实时音视频服务、CODESYS】服务需芯片适配支持，如有需要请 [提交工单咨询](#)

* 通信方式: ☒ 2G/3G/4G/5G ☐ NB-IoT ☐ WiFi ☐ Zigbee ☐ 蓝牙 ☐ 其他

* 协议选择: ☒ mqtt

产品描述:

1 基础配置

2 完成

✓ 创建成功

cmiot_fota_demo

产品ID: QGLmfK7b73

产品描述: demo product for cmiot

产品密钥: 5C807941630B487ACE3C2802E814BCE7 复制

请前往下载中心下载代码和工具，并参考开发文档进行配置

完成

下载代码

创建完成产品后，可以在【产品开发】【产品管理】选项卡中，查询到产品信息

中国移动 | Qneos 工作台首页

工作台

产品开发 / 产品管理

产品管理

用户可在此页面快速创建产品，配置产品的基本信息，获取产品密钥

新建产品

cmiot_fota_demo 查询

2G/3G/4G/5G网络

cmiot_fota_demo

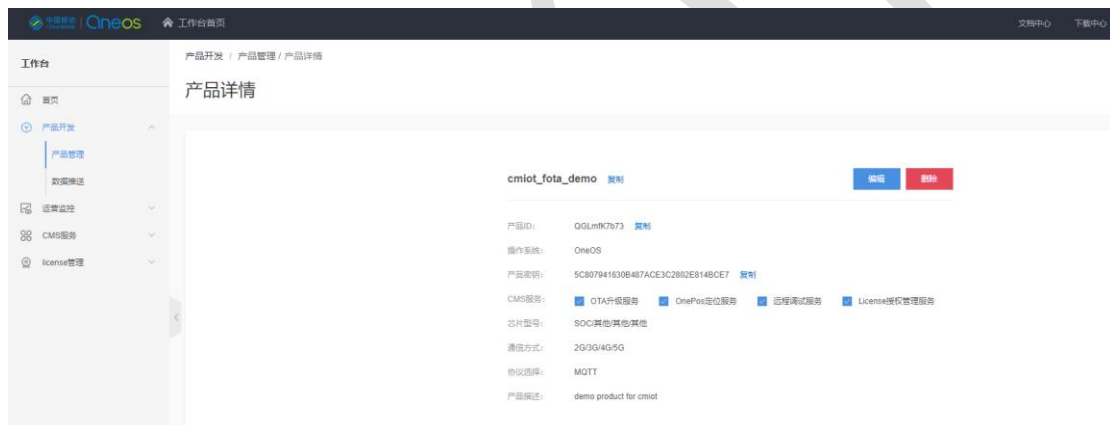
2023-02-01 14:39

芯片: 其他

产品ID: QGLmfK7b73

通信协议: MQTT

点击缩略信息可以进入产品详情页



3.2 制作版本包

①在 oneos_config.h 中修改 cms 产品信息，如果支持 OneOS Cube 配置，则通过 menuconfig 进行修改。

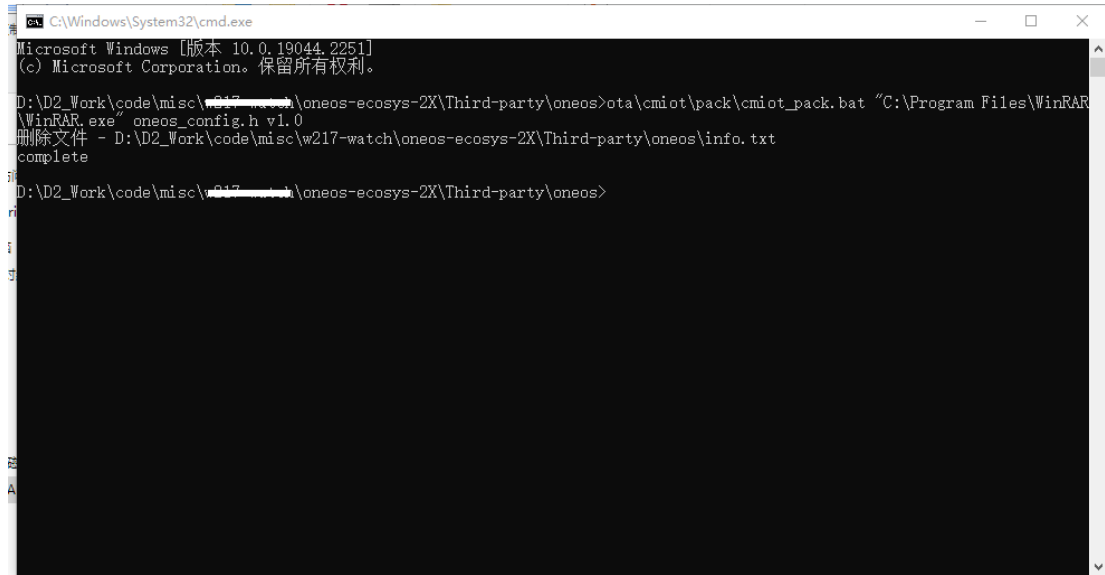
```
/* CMS ID */
#define CMS_USING_ID
#define CMS_ID_SET_PID "QGLmfK7b73" /* 产品 ID */
#define CMS_ID_KEY "5C807941630B487ACE3C2802E814BCE7" /* 产品密钥 */
#define UNILATERALISM_AUTH
/* end of CMS ID */
```

②在 oneos_config.h 中修改固件版本，如果支持 OneOS Cube 配置，则通过 menuconfig 进行修改。


```
#define CMIOT_FIRMWARE_VERSION "1.0"
```

③PC 上安装压缩软件 WinRAR，通过 `cmiot_pack.bat` 脚本完成版本包制作，在 `oneos_config.h` 所在目录，打开 `cmd`，执行如下指令

```
ota\cmiot\pack\cmiot_pack.bat "C:\Program Files\WinRAR\WinRAR.exe"
oneos_config.h v1.0
```




生成的版本包名称的前缀为版本和日期

 1.0_20230223_163504.zip

④修改源码，同时在 `oneos_config.h` 中修改版本号：例如修改为 2.0，并执行步骤③中的指令，生成 2.0 的版本包

```
#define CMIOT_FIRMWARE_VERSION "2.0"
```

 2.0_20230223_163332.zip

3.3 制作差分包

版本包制作完成后，通过平台生成差分包并发布。

①登录 `cms` 工作台 OTA 升级服务平台，并选择【CMS 服务】【OTA 升级服务】选项卡，进入 OTA 页面



②找到产品选项卡，并点击【版本管理】



③点击【添加版本】



④完善版本信息，并上传版本包

添加版本

版本号

1.0

①

版本别名

1.0

②

发布说明

2023-02-23

增加语言

中文 (中国)

中文 (中国)

1. 优化系统, 修复错误

描述

固件文件

点击上传

③

1.0_20230223_163504.zip

取消

确定

④

⑤按照步骤④的操作,完成 2.0 版本的固件上传,即可看到两个版本的信息。

FOTA升级 / 版本管理

请输入版本号

查询

+ 添加版本

版本号	版本名称	固件文件	目标版本	状态	操作
1.0	1.0	1.0_20230223_163504.zip	无	初始版本	版本详情 更多操作
2.0	2.0	2.0_20230223_163332.zip	无	无升级包	配置升级任务 版本详情 更多操作

< 1 > 10条/页 跳至 页 确定

⑥点击【配置升级任务】【配置】，进入配置页面，点击【确定】。

FOTA升级 / 版本管理

请输入版本号

查询

+ 添加版本

版本号	版本名称	固件文件	目标版本	状态	操作
1.0	1.0	1.0_20230223_163504.zip	无	初始版本	版本详情 更多操作
2.0	2.0	2.0_20230223_163332.zip	无	无升级包	配置升级任务 版本详情 更多操作

< 1 > 10条/页 跳至 页 确定

请输入版本号

查询

返回

配置升级包，编辑策略，测试，发布等操作约5分钟完成

目标版本: 2.0

源版本	发布时间	升级包名称	升级包大小	状态	升级包	升级策略	操作
1.0			0(0.00MB)	无升级包	配置		

配置升级包

操作系统

OneOS

升级类型

☒ 差分

制作差分包

☒ 线上自动差分
 ☐ 手动上传差分包

	版本号	固件文件
目标版本	2.0	2.0_20230223_163332.zip
源版本	1.0	1.0_20230223_163504.zip

!

差分包制作将按照目标版本升级可用RAM与Download分区制作，请确保与实际设备保持一致。为保证升级的可靠性，建议设备分配较大的可用RAM以及APP分区容量50%的Download分区。

升级可用RAM

512

(KB)

Download分区

640

(KB)

取消

确定

⑦稍作等待，即可完成差分包的制作接下来进入发布测试发布流程，依次点击【开始测试】【测试通过】【发布】

FOTA升级 / 版本管理 / 升级任务

请输入源版本号

查询

编辑

配置升级包，编辑策略，测试，发布等操作均5分钟后生效

目标版本：2.0

源版本	发布时间	升级包名称	升级包大小	状态	升级包	升级策略	操作
1.0	2023-02-23 16:42:43	dif_update20230223164243.zip.bin	281(0.00MB)	待测试	查看 修改 删除	编辑	开始测试

FOTA升级 / 版本管理 / 升级任务

请输入源版本号

配置升级包、编辑策略、测试、发布等操作均5分钟后生效

目标版本: 2.0

源版本	发布时间	升级包名称	升级包大小	状态	升级包	升级策略	操作
1.0	2023-02-23 16:42:43	dif_update20230223164243.zip	281(0.00MB)	测试中	查看	查看	测试通过 停止测试

FOTA升级 / 版本管理 / 升级任务

请输入源版本号

配置升级包、编辑策略、测试、发布等操作均5分钟后生效

目标版本: 2.0

源版本	发布时间	升级包名称	升级包大小	状态	升级包	升级策略	操作
1.0	2023-02-23 16:42:43	dif_update20230223164243.zip	281(0.00MB)	测试通过	查看 修改 删除	查看 编辑	发布 重新测试

注意，新版本发布需等待几分钟后，设备端方可检测到新的版本。

FOTA升级 / 版本管理 / 升级任务

请输入源版本号

配置升级包、编辑策略、测试、发布等操作均5分钟后生效

目标版本: 2.0

源版本	发布时间	升级包名称	升级包大小	状态	升级包	升级策略	操作
1.0	2023-02-23 16:42:43	dif_update20230223164243.zip	281(0.00MB)	发布中	查看	查看	停止发布

3.4 删除版本

在【版本管理】页面，点击【更多操作】可进行已发布版本的删除。



4 注意事项

①压缩软件推荐使用 WinRAR;

②download 分区除差分存放区域外，还另外需要两个区域用于备份和存放信息，所以 download 分区大小至少需要三个扇区；

③flash 操作接口适配时，当读写成功时返回读写数据的长度，读写失败时，返回-1；擦除失败时返回-1，擦除成功返回 0；

④在没有完善堆可使用 ota_heap_init 初始化升级使用内存，有完善堆空间时可不使用。

5 参考资料

[1]CMS 工作台: [文档中心 \(10086.cn\)](http://10086.cn)

[2]FOTA 服务: [文档中心 \(10086.cn\)](http://10086.cn)

[3]视频教程: [OneOS 学院 \(10086.cn\)](http://10086.cn)

6 FAQ

Q: 如何合理的配置 download 分区大小?

download 分区配置为需要升级的尺寸总和的 10~50%之间, 若空间非常紧张, 建议设置为最大应用尺寸的 20%。

Q: 可用 RAM 大小会对性能有哪些影响?

ram 空间不足时, 第一, 可能导致还原过程中内存分配失败, 从而增加升级失败的风险; 第二, 不会启用二次压缩, 相同的版本, 制作的差分包会更大, 分配给算法 ram 空间配置尽可能大, 最小不低于 3 个 block (最小擦除) 大小。

Q: 升级过程中断电会有什么后果?

升级过程中断电, 重新上电后会继续升级。但是升级过程中断电有可能带来其他风险, 因此推荐在升级在点亮充足的条件下进行, 并且过程中明确的提示用户不要断电。

Q: 哪些因素会影响升级的时间?

第一, 版本差异大小。升级前后版本差异越小, 需要更新的 flash 越少, 擦写 flash 的总次数越少, 耗时也就越少; 第二, 存储介质擦写耗时, 升级相同的版本, 存储介质擦写耗时越少, 总的耗时越少。

Q: 在一个平台上适配的还原库, 可以直接应用到其他项目上吗?

主要取决于以下几个方面: 1. 芯片架构, 2. 编译工具, 3. 算法配置; 如果这几项都一样即可可以直接应用到其他项目上。