

Melis

OTA 开发文档

1.0

2020.04.24

文档履历 (Document History)

Version No	Date	Prepared/Revised by	Content Description
1.0	2020.04.24	AW1613	创建



目录 (Table of Contents)

1. 概述	1
1.1 编写目的	1
1.2 适用范围	1
1.3 相关人员	1
2. 功能介绍	2
2.1 方案介绍	2
3. 接口介绍	3
3.1 aw_upgrade_slice	3
3.1.1 作用	3
3.1.2 函数原型	3
3.1.3 参数描述	3
3.1.4 测试命令	4
3.2 aw_upgrade_end	4
3.2.1 作用	4
3.2.2 函数原型	4
3.2.3 测试命令	5
4. OTA 应用	6
4.1 ota_task	6
4.1.1 作用	6
4.1.2 函数原型	6

4.1.3 参数描述	6
4.1.4 测试命令	7
5. 配置	9
6. Declaration	10



1. 概述

1.1 编写目的

介绍 Melis 系统 OTA 的实现及使用方法。

1.2 适用范围

Allwinner 软件平台 Melis-3.x

Allwinner 硬件平台 V459

1.3 相关人员

适用 Melis 平台的广大客户及关心 OTA 的相关人员。

2. 功能介绍

OTA 是 Over The Air 的简称, 即空中下载技术, 可通过无线网络从服务器上下载更新文件对本地系统或文件进行升级, 便于客户为其用户及时更新系统和应用以提供更好的产品服务, 这对于客户和消费者都极其重要。

2.1 方案介绍

OTA 需要支持升级 rtos, 支持升级文件系统中的指定文件。

支持 AB 系统切换。

AB 系统方案, 是将原有的系统, 增加一份。即 flash 上总共有 AB 两套系统。两套系统互相升级。

OTA 时, 若当前运行的是 A 系统, 则升级 B 系统, 升级完成后, 设置标志, 重启切换到 B 系统。

OTA 时, 若当前运行的是 B 系统, 则升级 A 系统, 升级完成后, 设置标志, 重启切换到 A 系统。

可通过 `printenv` 查看当前系统:

```
rootAB_now=A/B
```

数据写完之后可通过 `printenv` 查看 `rootAB_next` 是否更新, 会在下次系统启动的时候自动切换系统, 暂没有支持更新完之后自动重启。

升级系统时可由 `awota_task` 获取升级数据的来源, 由 `aw_upgrade` 负责决定判断需要更新的系统, 完成数据写入以及系统切换等动作。

3. 接口介绍

3.1 aw_upgrade_slice

3.1.1 作用

升级时调用，用于升级分片数据。

该函数负责将数据写入 flash 的对应位置，并保证写入后读出校验成功。

3.1.2 函数原型

```
int aw_upgrade_slice(uint8_t* target,
                    uint8_t* buffer,
                    uint32_t offset,
                    uint32_t size,
                    uint32_t flag);
```

返回值：

0：成功；非0：失败。

3.1.3 参数描述

- **target**，当前要升级的分区文件名称，是一个特定文件名。当前特定文件名支持："rtos"。
- **buffer**，指定文件为源数据。会被读到 **buffer** 中，并以 **buffer** 的形式传递给 **aw_upgrade_slice** 函数，作为更新的数据源。
- **offset**，表示该分片在目标分区的 **offset** 位置，以 **Byte** 为单位。
- **size**，表示当前的分片大小，以 **Byte** 为单位。
- **flag**，最低位表示是否是要更新到目标分区的最后一块数据，其余位保留，目前传 0 即可。

3.1.4 测试命令

控制台支持 `aw_upgrade` 命令用于测试。

`aw_upgrade` 带 5 个参数用于测试 `aw_upgrade_slice` 接口的从 `buffer` 更新功能。

实际场景中，`buffer` 的数据应该是从网络下载得到的。

假设将待更新的 `rtos` 拆分为 `/data/update_1.bin`, `update_2.bin`, `update_3.bin`, 其大小为 512000, 512000, 73728, 则调用

```
aw_upgrade rtos      0 512000 0 /data/update_1.bin
aw_upgrade rtos 512000 512000 0 /data/update_2.bin
aw_upgrade rtos 1024000 73728 1 /data/update_3.bin
```

3.2 `aw_upgrade_end`

3.2.1 作用

升级完成时调用。

该函数负责结束的动作，设置系统切换标志，下次重启时切换到新系统。

3.2.2 函数原型

```
int aw_upgrade_end(uint32_t flag);
```

参数：

flag: 特殊标志，预留，传0即可。

返回值：

0: 成功；非0: 失败。

3.2.3 测试命令

控制台支持 `aw_upgrade` 命令用于测试。

`aw_upgrade` 不带参数用于测试 `aw_upgrade_end` 接口。



4. OTA 应用

4.1 ota_task

4.1.1 作用

OTA 应用 demo, 获取网络或本地 OTA 包, 解析出文件并调用底层 OTA 接口完成更新。

升级 boot 分区时调用。

awota_task 函数会调用 aw_upgrade_slice 和 aw_upgrade_end 函数完成整个升级 boot 分区的过程。

该函数支持本地升级及网络升级, 升级时会显示进度条。

4.1.2 函数原型

```
int ota_task(char *para_in);
```

返回值:

0: 成功; 非0: 失败。

4.1.3 参数描述

para_in, 当前更新 boot 分区的数据来源, 为 buffer 或者 url。

buffer 为本地数据, url 为网络数据。

4.1.4 测试命令

控制台支持 **awota** 命令用于测试。

准备 OTA 包，编译生成的 **melis_pkg_nor.fex** 可直接当成 OTA 包来使用，升级 boot 分区。即

```
./source/workspace/suniv/beetles/melis_pkg_nor.fex
```

1. 本地升级

将 OTA 包放到 **/data/** 中，如

```
adb push ./source/workspace/suniv/beetles/melis_pkg_nor.fex /data/
```

则升级命令为

```
adb shell awota /data/melis_pkg_nor.fex
```

2. 网络升级

- 先进行联网

```
bcm_init  
bcm_sta connect ssid security passwd
```

- 准备好服务器
pc 端与开发板在同一局域网内

```
sudo python -m SimpleHTTPServer 80
```

- 升级

例如 OTA 包可下载的 url 为 `http://192.168.12.34/melis_pkg_nor.fex`, 则升级命令为

```
adb shell awota http://192.168.12.34/melis_pkg_nor.fex
```

5. 配置

配置 OTA，默认都选中。

```
make menuconfig
Kernel Setup --->
  Subsystem support --->
    Allwinner Components Support --->
      AW OTA Components --->
        [*]Melis OTA AW_UPGRADE
        [*]aw ota demo
```

OTA 功能需要系统支持 env，配置 env，默认选中。

```
make menuconfig
Kernel Setup --->
  Subsystem support --->
    [*]env support --->
      --- env support
      [4096] Env Size (bytes)
```

6. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgment to the copyright owner. The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application. tates nor implies warranty of any kind, including fitness for any particular application.