

Melis RTOS OTA 开发指南

版本号: 1.0

发布日期: 2020-10-22





版本历史

版本号	日期	制/修订人	内容描述
1.0	2020.08.21	AWA1046	初始版本。







目 录

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	配置	2
3	功能介绍	3
	3.1 分区情况	3
	3.1.1 boot0	3
	3.1.2 gpt	3
	3.1.3 env/env-redund	4
	3.1.4 rtosA/rtosB	4
	3.2 方案介绍	4
4	接口介绍	5
	4.1 aw_upgrade_slice	5
	4.2 aw_upgrade_end	5
	4.2 aw_upgrade_end	6
5	OTA 应用	7
	5.1 aw_upgrade	7
	5.2 测试命令	7
	5.3 调试命令	5



概述

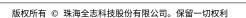
1.1 编写目的

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

・八八 使用 Melis 平台的广大客户及关心 OTA 的相关人员。





配置项如下

make menuconfig

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





3 功能介绍

3.1 分区情况

flash 上有如下数据

数据	描述
boot0	bootloader, 全志私有格式
gpt	分区表,标准 gpt 格式
env	env 配置,同开源 uboot 的 env 格式
env-redund	env 配置,同开源 uboot 的 env 格式
rtosA	rtos 系统本身
rtosB	rtos 系统本身
UDISK	文件系统,一般是 littlefs

rtos 中为这些分区生成了文件系统节点,类似 Linux 下的块设备节点,应用直接读写 /dev/分区 名即可读写到对应分区的数据。

3.1.1 boot0

BROM 决定了 nor 方案上仅支持一份 boot0, 故升级 boot0 中途掉电, 有导致机器无法启动的风险。不建议升级。

如需升级,直接将新 boot0 数据写入 /dev/boot0 节点即可。

3.1.2 gpt

对应源码中的分区表配置文件位于方案配置目录

source/projects/defconfig/...(方案)/sys_partition.fex



3.1.3 env/env-redund

存放一些键值对的数据,格式跟 uboot 的 env 格式保持一致。

boot0 和 rtos 均可读写。即 env 可作为 bootloader 和 rtos 沟通的渠道。

为了保证掉电安全, 避免写 env 中途掉电导致 env 数据不完整, 设置了两份 env 数据。

写入是对两份 env 交替更新,读出是自动读取较新的一份 env, env 数据有专门的读写接口, 上 层调用接口时只需当成一份 env 来使用即可。

对应源码中的 env 配置文件位于方案配置目录

source/projects/defconfig/...(方案)/env.cfg

在控制台可使用 fw printenv 和 fw setenv 对 env 数据进行读写。

3.1.4 rtosA/rtosB

IER 为了保证掉电安全,设置了两份 rtos,分别放在 rtosA 和 rtosB 分区。

如需升级, 直接将新 rtos 数据写入 /dev/rtosA 或 /dev/rtosB 节点即可。

3.2 方案介绍

env 分区存在 rtosAB next 和 rtosAB now 两个变量, 取值为 A 或 B。

boot0 会根据 rtosAB next 的值来决定启动 rtosA 或 rtosB, 并设置 rtosAB now 的值。

在 rtos 中, 可读取 rtosAB now 的值, 来获知本次启动的 AB 哪个系统, 从而选中另外一个作为 本次要升级的分区。

rtos 可设置 rtosAB next 的值,来指示下次要启动的系统。



4 接口介绍

为了对上屏蔽 AB 系统,屏蔽实际写 flash 和操作 env 等细节,提供两个 OTA 接口函数。上层 OTA 应用可使用本节所述接口完成 OTA,不再需要关心底层细节。

4.1 aw upgrade slice

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。

本接口自动判断要升级的是 A 还是 B,将数据写到目标分区的指定偏移处。

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

参数
  target: 指定升级目标,目前仅支持取值为 "rtos"
  buffer: 数据缓冲区,保存要写入flash的数据
  offset: 数据在目标区中的偏移量
  size: 数据长度
  flag: 特殊标志,具体取值由宏定义设置
    目前仅支持设置为 0 或 1,表示此片数据为对应目标的最后一片数据

返回值
  0: 成功
  非0: 失败
```

4.2 aw upgrade end

本接口在升级完成后,设置 env 标志,切换系统。

```
int aw_upgrade_end(uint32_t flag);

参数
flag: 特殊标志,未启用,传0即可。

返回值
0: 成功
```





非0: 失败

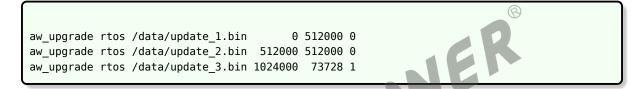
4.3 测试命令

控制台支持 aw_upgrade 命令用于测试 OTA 底层接口。

1. 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



2. aw_upgrade 不带参数用于测试 aw_upgrade_end 接口。





5 OTA 应用

OTA 应用主要负责获取并解析 OTA 包, 并将数据传给底层 OTA 接口。

由于 dram 和 flash 空间均有限,一般需要采用流式升级,即边从网络下载数据,边写入 flash 中。

5.1 aw upgrade

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。

int ota_task(char *para_in);

参数
 para_in: OTA包,可以为 url 或本地文件路径。
 返回值
 0: 成功
 非0: 失败

5.2 测试命令

控制台支持 awota 命令用于测试 ota task 接口。awota 的源码位于

ekernel/subsys/aw/ota/demo/aw-ota.c

可从本地文件系统或网络获取数据,再调用上述 OTA 接口进行更新。

- 1. 准备 OTA 包准备 OTA 包,编译生成的 melis_pkg_nor.fex 可直接当成 OTA 包来使用,升级 rtosA/rtosB 分区。
- 2. 本地升级

将 OTA 包放置到 /data/ 中, pc 端执行:





adb push melis_pkg_nor.fex /data

使用 adb 调用升级命令:

adb shell awota /data/melis_pkg_nor.fex

- 3. 网络升级
- 小机端先进行联网:

具体联网命令请参考网络相关文档



• 准备好服务器, pc 端与小机端在同一局域网下, PC 端执行:

sudo python -m SimpleHTTPServer 80

● 升级假设 OTA 包可下载的 url 为 http://192.168.xx.xx/melis_pkg_nor.fex,则 PC 端可用 adb 调用升级命令:

adb shell awota http://192.168.xx.xx/melis_pkg_nor.fex

4. 判断结果若需要在升级后判断是否确实升级成功,可准备一份有差异的系统(例如添加指定打印),升级后观察启动 log

也可在升级后,通过 adb pull /dev/rtosA 取出对应分区的数据,跟 PC 端的原始数据进行比较。

5.3 调试命令

控制台支持 printenv 和 setenv 命令用于查看和设置 env 分区的值。





printenv #打印env变量 setenv rtosAB_next B #标记下次启动B系统





著作权声明

版权所有 © 2020 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护,其著作权由珠海全志科技股份有限公司("全志")拥有并保留 一切权利。

本文档是全志的原创作品和版权财产,未经全志书面许可,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



举)均为珠海全志科技股份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



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。全志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,全志概不负责。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全志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。全志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)。全志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。