

Melis v3.0 打包流程说明



文档版本：V1.0

发布日期：2020-04-7

前言

概述

本文档主要介绍 melis v3.0 SDK 的打包流程，旨在让客户了解打包时做了什么工作

适用范围

本文档适用于全志科技 melis v3.0 SDK

适用对象

本文档主要适用于：

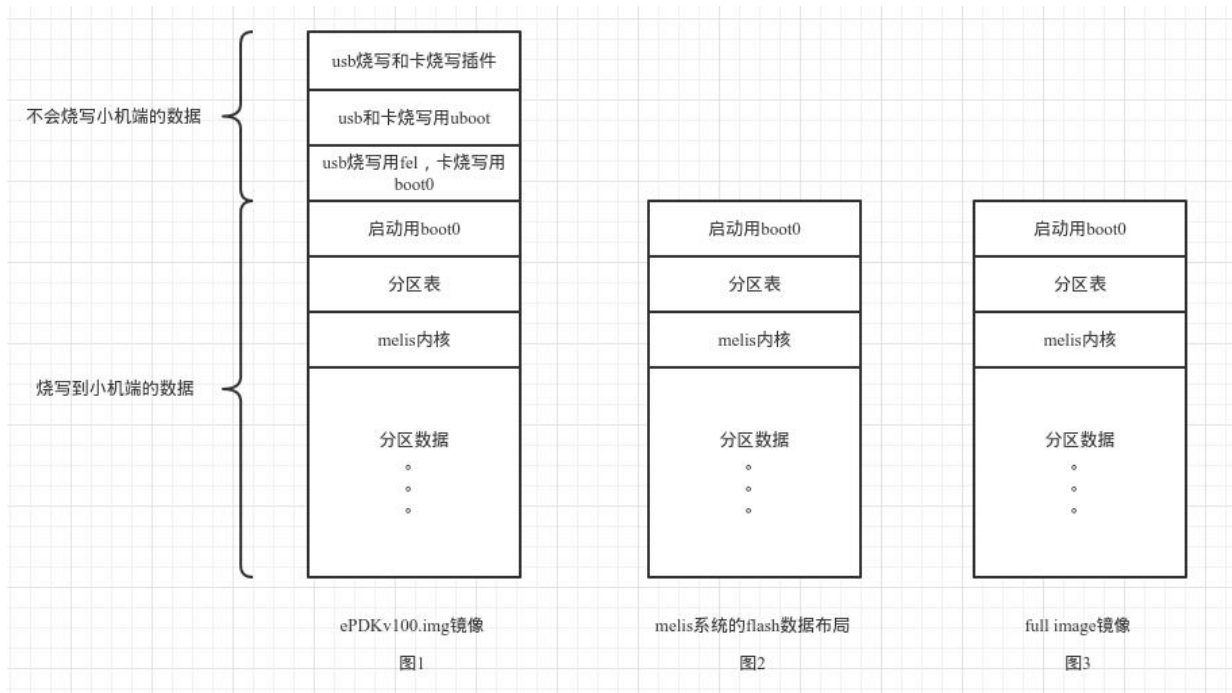
- > 需要了解 melis v3.0 SDK 打包流程的软件开发工程师
- > 技术支持工程师

修订记录

修订记录累积每次文档的更新说明，最新版本的文档包含以前所有文档版本的更新内容，请以最新版为准。

版本	修改记录	日期	作者	备注
Ver 1.0	Initial Version	2020-04-7		

一、了解打包流程前需要的基础知识



要理解打包流程，需要了解上面 3 个图

图 1 ePDKv100.img 镜像：

该镜像可以用 usb 工具进行烧写，也可以把该镜像制作成量产卡以卡量产的方式进行烧写。该镜像包含了插件部分，插件部分不会烧写到小机端的 flash，主要用于支持 usb 烧写和卡烧写。另外部分是烧写到小机端的数据，具体烧哪些数据由分区表文件 sys_partition.fex 决定。

图 2 melis 系统 flash 数据布局：

无论是 usb 还卡烧写，烧写完后 flash 的数据布局如图 2,对比图 1 和图 2 可以看出 ePDKv100.img 镜像中只有部分数据会烧到 flash

图 3 full image 镜像：

与图 2 melis flash 布局是完全一致的，full image 镜像主要是用于烧录器用。

打包流程实质上主要就是创建出 ePDKv100.img 和 full image 2 个镜像

二、打包脚本

在 melis SDK 环境下执行了 `source melis-env.sh` 后会把 `pack` 命令 `export` 到 shell 环境变量中

(其中 `pack` 命令在 `melis-v3.0/source/envsetup.sh` 文件里实现)

此时只要执行 `pack` 命令就会去执行 `pack_img.sh` 脚本 (执行 `pack` 脚本时已经是在生成固件了, 此时内核需要被编译出来, 否则会 `pack` 失败)。 `pack_img.sh` 脚本文件就是整个打包流程。查看脚本文件可以知道整个脚本文件就是按顺序执行一系列函数, 下面讲述的是这一系列函数的每个函数的作用。

三、打包脚本函数作用

1.do_select_platform 函数

(1). 这个函数在 `select.sh` 文件中实现, 主要是选择使用哪个平台, 目前 `melis v3.0 v459` 使用的是 `sun8iw19` 平台芯片

2.do_init_prepare 函数:

打包前做的一些准备

- (1). 拷贝一些 `usb` 烧写和卡烧写时使用的插件, 这些插件最终也会被打包进 `ePDKv100.img` 固件中, 在烧写的时候, 不同的 `pc` 烧写工具会从固件中获取不同的插件辅助进行烧写
- (2). 制作 `env` 分区的二进制数据

3.do_init 函数:

- (1). 删除上次生成的 `ePDKv100.img` 固件, 确保 `ePDKv100.img` 固件是重新生成的
- (2). 根据平台选择不同 `usb` 烧写工具目录

4.update 函数:

(1).拷贝启动用的 boot0

(2).拷贝制作 rootfs 所需文件到打包目录

5.script_config 函数：

处理配置文件（把脚本格式的配置文件转换成 2 进制格式的脚本文件，方便后续工具使用）

(1).用 busybox 工具把配置文件转换成 dos 格式

(2).用 script 工具把 dos 格式的配置文件（脚本格式）转换成二进制文件（方便后面其他工具使用）

（sys_config.fex 是烧写工具和启动用 boot0 会使用的文件，sys_partition.fex 文件是分区配置件）

6.update_boot 函数：

(1).用工具 update_boot0_xxx 把配置文件里的一些参数更新到启动用 boot0 和卡烧写用 boot0 中

(2).用 dragonsecboot 工具把 melis 内核打包成 boot_package 格式的包，并进行一些名字的修改

7.do_fs 函数

(1).用工具制作文件系统

8.do_mbr 函数

(1).用工具 update_mbr 把二进制的分区表配置文件转换成小机端可使用的分区表

9.do_fes1_uboot_update 函数

(1).用 update_fes1 工具把配置文件里的一些参数更新到烧写用 fes1 文件中

(2).用 update_uboot 工具把配置文件里的一些参数更新到烧写用 uboot 文件中

10.do_create_rtos_full_img 函数

(1).用 mbr_convert_to_gpt 和 merge_full_rtos_img 工具制作 full image 镜像，烧录器用

11.do_package 函数

(1).用 dragon 工具制作 usb 和 sd 卡用烧写镜像 (根据配置文件 image.cfg 把需要的文件拼起来)

usb 和卡烧写用

12.do_finish 函数

(1).删除生成的中间文件

