



Melis4.0 RTOS 方案快速开发指南

版本号: 0.1
发布日期: 2021.05.12

版本历史

版本号	日期	制/修订人	内容描述
0.1	2021.05.12	PDC-PD1	创建



目 录

1 前言	1
1.1 编写目的	1
1.2 使用范围	1
1.3 相关人员	1
2 目录结构	2
2.1 SDK 目录结构	2
2.2 source 目录结构	3
3 编译打包	4
4 添加方案	5
5 启动概述	6



1 前言

1.1 编写目的

介绍 melis 4.0 的开发环境和系统架构。

1.2 使用范围

Allwinner 软件平台 Melis4.0 以上版本。

1.3 相关人员

使用 Melis4.0 以上版本进行产品开发的开发者。

2 目录结构

2.1 SDK 目录结构

```
.├─ melis-4.x-release│  └─ document│  └─ source│  └─ toolchain│  └─ tools
```

文件夹	功能
document	用于存放开发文档;
source	用于存放 sdk 的源代码;
toolchain	用于存放解压后的交叉编译工具链;
tools	用于存放交叉编译工具链的压缩包;

Melis4.0 采用 gcc 交叉编译工具链，在 linux 宿主机平台下使用 make 工具对各源代码和库文件进行编译链接，生成在 SOC 平台上运行的可执行文件。

工具链以压缩包的形式存放在 tools 文件夹下，编译时解压到 toolchain 文件夹中。在编译期间，会先检测是否存在 toolchain 文件夹，如果不存在 toolchain 文件夹，则创建 toolchain 文件夹，再把 tools 中的工具链解压 toolchain 下。如下所示：

文件source/scripts/toolchain.mk

```
toolchain_check=$(strip $(shell if [ -d $(tooldir) ]; then echo yes; fi))
ifneq ("$(toolchain_check)", "yes")
    $(info Found toolchain $(notdir $(toolchain_archive)))
    $(info Extracting GCC.....)
    $(shell mkdir -p $(tooldir))
    $(shell tar --strip-components=1 -xf $(toolchain_archive) -C $(tooldir))
    $(info Done)
endif
```

2.2 source 目录结构

```
.
├── drivers -> ekernel/drivers
├── ekernel
├── elibrary
├── emodules
├── include
├── livedesk
├── out
│   └── f133-evb-board
├── projects
│   ├── f133-evb-board
│   └── v459-perfl
├── scripts
├── tools
│   ├── packtool
│   ├── phoenixcard
│   ├── phoenixplugin
│   ├── python
│   └── scripts
└── workspace
```

drivers: 软连接，指向 ekernel/drivers 文件夹，主要存放驱动层的 drv 和 hal 代码；

ekernel: 存放内核代码，包含启动、中断、OS、drivers 驱动代码、文件系统、设备管理、控制台、系统接口等组件；

emodules: 存放模块代码；

include: 存放公共性的头文件；

livedesk: 存放应用层代码；

out: 存放各方案的打包固件和打包期间生成的临时文件；

projects: 存放各方案的配置文件、生成的驱动/模块文件。目前各模块中的 Makefile 把生成的可执行文件最终拷贝到本目录；

scripts: 主要存放编译脚本，各脚本协同控制着整个 SDK 的编译流程；

tools: 主要存放打包工具和打包脚本，打包脚本的路径为 source\tools\scripts\pack_img.sh；

workspace: 未使用，被 projects 和 out 取代，配置和可执行文件放在 projects 目录，打包固件放在 out 目录；

3 编译打包

```
step01: cd melis-4.x-release/source/;
step02: source melis-env.sh;
step03: lunch;
step04: 输入数字按键后，按回车键选择方案，本例输入1后按回车键，选择f133-evb-board方案；
step05: make menuconfig后直接保存；
step06: make;
step07: pack;打包生成固件source/out/f133-evb-board/melis_f133-evb-board_uart0_8Mnor.img
step08: 使用phoenixSuit工具烧录固件到evb开发板方案；
```



4 添加方案

为方便方案扩展，source/projects 专门用于存放各方案相关的配置、驱动等文件。

```
step01: 在source/projects/目录下，复制一份f133-evb-board文件夹，并修改成想要的方案名称；
step02: make clean;清除掉以往的编译临时文件，例如.o文件；
step03: make distclean;清除掉以往方案的配置文件，例如source目录下的.config；
step04: 删除与source同级目录下的toolchain文件夹，在后续步骤中重新从tools文件夹中的压缩包解压出来；
step05: lunch;
step06: 输入方案名称前对应的数字后，按回车键选择方案；
step07: make menuconfig;后直接保存，在本步骤机会重新解压step04中被删除的toolchain工具链文件夹；
step08: make;
step09: pack;打包生成固件，固件会在source/out/文件夹下对应的方案名称文件夹中；
step10: 使用phoenixSuit工具烧录固件；
```


5 启动概述

Melis4.0 系统的启动顺序：

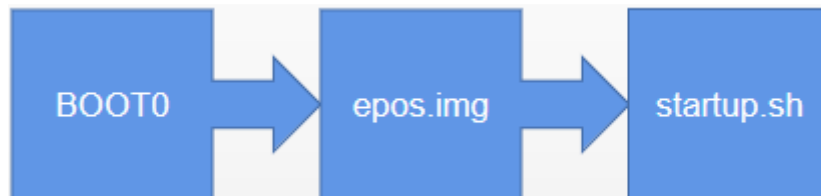


图 5-1: startup flow

模块	存储位置
boot0	source/projects/f133-evb-board/bin/
epos.img	source/projects/f133-evb-board/
startup.sh	source/projects/f133-evb-board/data/UDISK/

boot0：以.bin 文件的形式提供；

epos.img：由 ekernel 编译生成；

startup.sh：控制台会解析该脚本，并逐行执行脚本中的命令；

默认在 startup.sh 脚本中加载应用相关的模块。

如source/ekernel/subsys/finsh_cli/shell_entry.c文件568行所示：

```
568     ret = msh_exec_script(CONFIG_FINSH_EXTERN_SCRIPT_PATH"startup.sh", \  
569                          strlen(CONFIG_FINSH_EXTERN_SCRIPT_PATH"startup.sh"));
```

著作权声明

版权所有 © 2021 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

、 全志科技、（不完全列举）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

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

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。