Rockchip

Linux A/B 开发指南

发布版本:**1.00**

日期:**2019.02**

免责声明

本文档按“现状”提供，福州瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

版权所有 © 2019 福州瑞芯微电子股份有限公司

福州瑞芯微电子股份有限公司
Fuzhou Rockchip Electronics Co., Ltd.
地址： 福建省福州市铜盘路软件园 A 区 18 号
网址： www.rock-chips.com
客户服务电话： +86-591-83991906
客户服务传真： +86-591-83951833
客户服务邮箱： service@rock-chips.com

前言

概述

本文档主要介绍 Rockchip Linux A/B 双分区引导和升级功能，以及如何进行二次开发。

产品版本

芯片名称	内核版本
RK33XX	4.41

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

日期	版本	作者	修改说明
2019.02.27	v1.0.0	Hkh	初始文档

目录

1 Linux A/B 介绍 1-1
1.1 概述 1-1
1.2 优点 1-1
1.3 缺点 1-1
1.4 分区 1-1
2 Linux A/B 引导 2-1
2.1 数据格式及存储 2-1
2.2 引导流程 2-1
3 Linux A/B 升级 3-1
4 Bootcontrol 说明 4-1
4.1 程序功能 4-1
4.2 自动执行 4-1
5 编译说明 5-1
5.1 u-boot 5-1
5.2 Buildroot 5-1
5.3 分区表 5-1
5.4 输出固件 5-2

1 Linux A/B 介绍

1.1 概述

Linux A/B，即准备两份独立的系统固件，分别存放在 flash 上，系统可以从其中一个 slot 启动，如果当前 slot 启动失败，可以从另外一个 slot 启动，在开机状态下直接升级系统，无需进入系统升级模式，只需重启系统即可进入升级过的系统。

1.2 优点

Linux A/B 由于有两个引导 slot，所以具有以下优点：

1. 升级无需重启进入升级模式，即机器可以在当前系统上直接进行升级。
2. 防止由于升级失败导致机器变砖，如果升级失败，机器可以回到当前版本。
3. 当前系统如果由于一些误操作被破坏掉，系统会自动切换到另外一个 slot 上。

1.3 缺点

Linux A/B 有两个 slot，所以会增加 flash 上系统固件的占用率。

1.4 分区

由于 miniloader，trust，uboot，机器上原有已经进行了多备份，所以目前这几个分区暂不支持双分区方案，只对 boot 和 system 进行了双分区。分区表如下：

miniloader	uboot	trust	misc	boot_a	boot_b	system_a	system_b	userdata

2 Linux A/B 引导

2.1 数据格式及存储

存储位置为 misc 分区偏移 2K 位置，AvbABSlotData 和 AvbABData 数据结构如下：

AvbABSlotData：存储 slot_a 和 slot_b

数据名称	数据作用
unsigned char priority	分区优先级，0~15，0 为不可自动，15 为最高优先级
unsigned char tries_remaining	尝试启动次数，最高为 7 次，可修改
unsigned char successful_boot	0：不可启动，1：可启动
unsigned char is_update:1	0：升级失败，1：升级成功，后 7 位为保留数据

AvbABData：slot_a 和 slot_b 的引导信息

unsigned char magic[AVB_AB_MAGIC_LEN]	结构体头部信息：\0AB0
unsigned char version_major	版本信息
unsigned char version_minor	版本信息
unsigned char reserved1[2]	保留数据
AvbABSlotData slots[2]	分区引导信息
unsigned char last_boot	上一次成功启动的分区：0->slot_a，1->slot_b
unsigned char reserved2[11]	保留数据
unsigned char crc32	Crc 数据校验

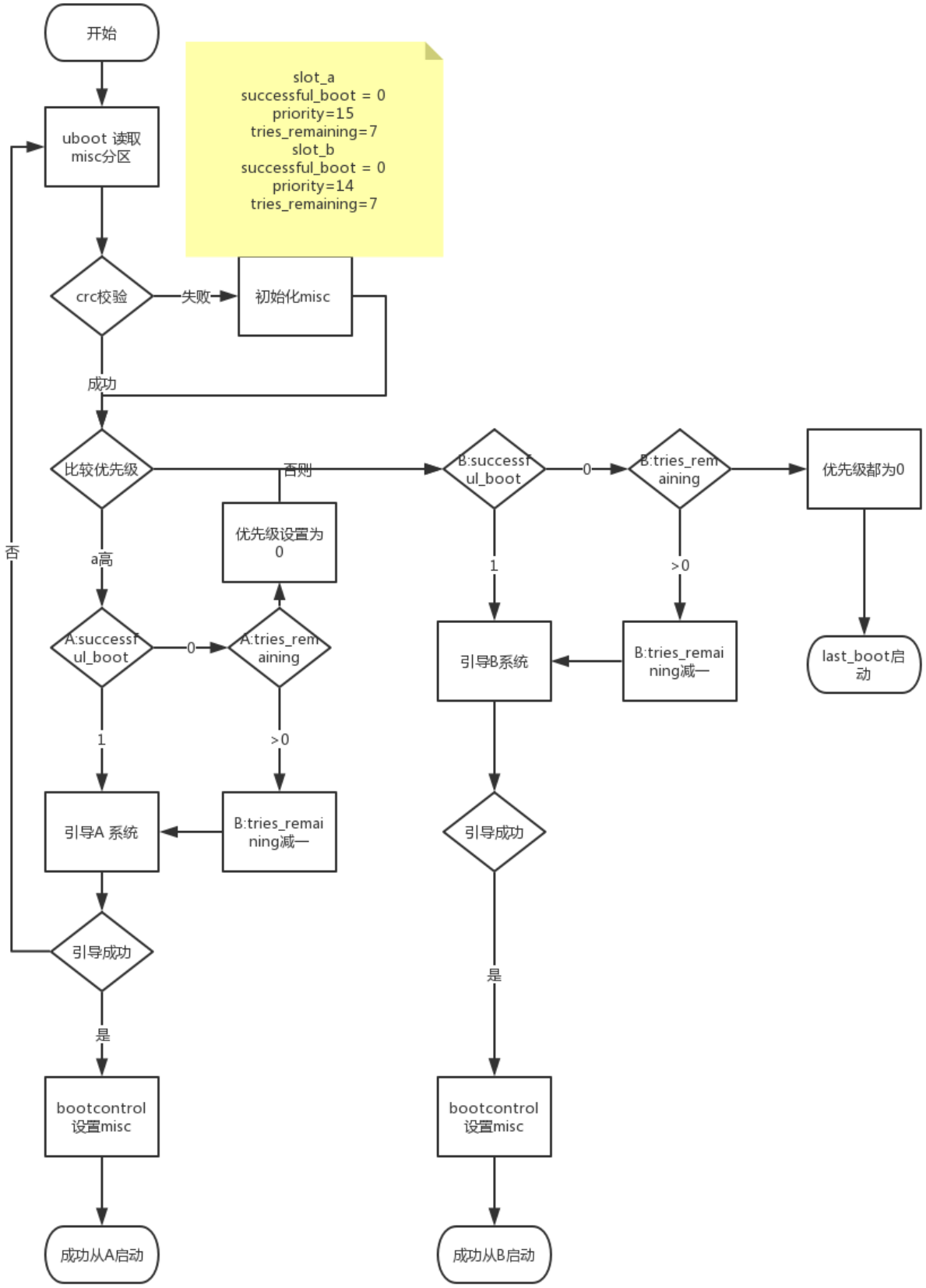
2.2 引导流程

根据上层 bootcontrol 程序的设置方式，可分为两种引导方式 successful_boot 和 reset retry。

两种模式的对比如下：

模式	优点	缺点	成功启动设置的数据（A 启动）	升级时设置的数据（A 启动，升级 B）
Successful_boot 模式	只要正常启动系统，不会回退到旧版本固件	设备长时间工作后，如果存储某些颗粒异常，会导致系统一直重启	tries_remaining=0 successful_boot=1 last_boot=0	A:priority = 14 B:priority = 15
Reset retry 模式：	始终保持 retry 机制，可以应对存储异常问题	1. 机器会回到旧的版本上，可能出现版本不可控问题 2. 如果因为客户误操作，retry 尝试次数过了，会误判为当前分区为不可启动	tries_remaining=7 last_boot=0	A:priority = 14 B:priority = 15

2.2.1 引导流程图


开始
slot_a
successful_boot = 0
priority=15
tries_remaining=7
slot_b
successful_boot = 0
priority=14
tries_remaining=7
uboot 读取misc分区
crc校验
失败
初始化misc
成功
比较优先级
否则
B:successful_boot
0
B:tries_remaining
优先级都为0
优先级设置为0
否
a高
1
>0
A:successful_boot
0
A:tries_remaining
引导B系统
B:tries_remaining减一
last_boot启动
1
>0
引导A 系统
B:tries_remaining减一
引导成功
引导成功
是
是
bootcontrol设置misc
bootcontrol设置misc
成功从A启动
成功从B启动

3 Linux A/B 升级

3.1.1 升级接口

Rockchip 提供简单的升级接口，可以参考使用，该升级程序，支持网络升级。
源码位于 external/update_engine/，提供如下接口：

函数名称	作用
void RK_ota_set_url(char *url);	设置升级包的 URL 路径
void RK_ota_start(RK_upgrade_callback cb);	开始升级
int RK_ota_get_progress();	获取升级进度
void RK_ota_get_sw_version(char *buffer, int maxLength);	获取软件版本号

头文件位置：/usr/include/libupdateengine/update.h
库文件位置：/usr/lib/libupdateengine.so

3.1.2 参考程序

接口调用参考：test_main.cpp
1. 检查版本号
检查机器版本号和本地版本号是否一致，使用如下命令：
update_engine check http://148.70.52.169:8080/version
返回值：
0：需要升级
-1：不需要升级
版本文件说明，参考 3.1.3 节
2. 升级固件
升级完自动重启：
update_engine update http://148.70.52.169:8080/update.img reboot
升级完不自动重启：
update_engine update http://148.70.52.169:8080/update.img

3.1.3 版本号文件

文件 device/rockchip/common/Version.mk 进行设置，如下图:
注意：机器版本号文件和服务器版本号文件格式需保持一致

```
#!/bin/bash

#MODEL NAME
export RK_MODEL=RKXXXX_RETROGAME

#SOFT VERSION
export RK_VERSION=V1.0.0

#OTA HOST
export RK_OTA_HOST=172.16.21.205:8080
~
~
```

4 Bootcontrol 说明

4.1 程序功能

rkboot_control 程序的主要功能是负责 misc 分区的读写，目前支持如下命令：

命令	作用
rkboot_control	打印当前 misc 分区的数据
rkboot_control now	设置当前分区为可启动分区
rkboot_control other	设置另外一个分区为升级分区
rkboot_control wipe_userdata	重启之后格式化/userdata 分区
rkboot_control wipe_userdata reboot	立即重启，且格式化/userdata 分区

4.2 自动执行

rkboot_control now 要在 system 成功引导之后执行，以标记系统成功启动，参考如下脚本

```
external/update_engine/S99_bootcontrol
case "$1" in
  start)
        /usr/bin/rkboot_control now
        ;;
  stop)
        printf "stop finished\n"
        ;;
  *)
        echo "Usage: $0 {start|stop}"
        exit 1
        ;;
esac
exit 0
```

5 编译说明

5.1 u-boot

defconfig 增加如下配置，如 rk3308 64bit：u-boot/configs/rk3308_defconfig

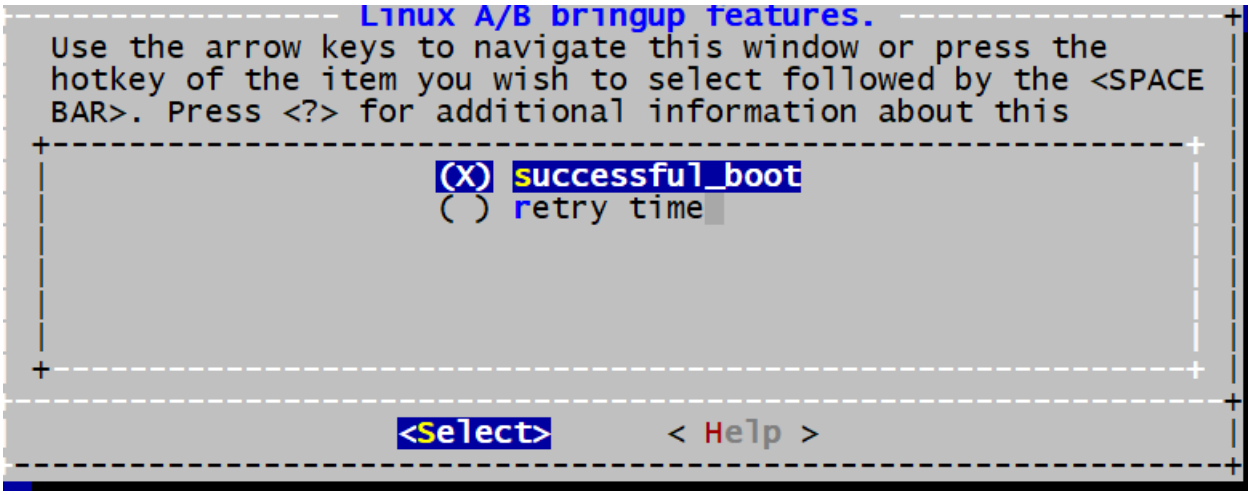
```
CONFIG_AVB_LIBAVB=y
CONFIG_AVB_LIBAVB_AB=y
CONFIG_AVB_LIBAVB_ATX=y
CONFIG_AVB_LIBAVB_USER=y
CONFIG_RK_AVB_LIBAVB_USER=y
CONFIG_ANDROID_AB=y
```

5.2 Buildroot

make menuconfig 开启如下配置

```
[*]    Rockchip LINUXAB for linux
           Linux A/B bringup features. (successful_boot)  --->
```

引导方式默认为 successful_boot 模式，可以修改为另外一个方式 retry time



注意：设置完成之后，须进行重新编译，如下：

```
make LiunxAB-dirclean
make LinuxAB
./build.sh
```

5.3 分区表

相应的 BoardConfig.mk，设置 parameter 分区表，如下：

```
#选择了 device/rockchip/rk3308/parameter-ab-64bit.txt 文件
# parameter for GPT table
export RK_PARAMETER=parameter-ab-64bit.txt
```

64bit：参考/device/rockchip/rk3308/parameter-ab-64bit.txt
32bit：参考/device/rockchip/rk3308/parameter-ab-32bit.txt

5.4 输出固件

5.4.1 生成方式

相应的 BoardConfig.mk，设置开启 Linux A/B 自动编译系统，开启方式如下：

```
#choose enable Linux A/B
export RK_LINUX_AB_ENABLE=true
```

设置完成之后，运行

```
source envsetup.sh
./build.sh
```

即可生成如下固件：

```
tree rockdev/
rockdev/
├── boot.img
├── MiniLoaderAll.bin
├── misc.img
├── oem.img
├── parameter.txt
├── recovery.img
├── rootfs.img
├── trust.img
├── uboot.img
├── update_ab.img
├── update.img
├── update_ota.img
└── userdata.img
0 directories, 13 files
```

5.4.2 升级固件

rockdev 和 IMAGE 目录下，都会有 update_ota.img，用于 OTA 升级，该 IMAGE 包，包含 boot.img 和 rootfs.img。可根据实际需求修改 tools/linux/Linux_Pack_Firmware/rockdev/rk3308-package-file-ota 文件。如下图：

```
# NAME          Relative path
#
#HWDEF         HWDEF
package-file    package-file
bootloader    Image/MiniLoaderAll.bin
parameter     Image/parameter.txt
#trust          Image/trust.img
#uboot          Image/uboot.img
boot          Image/boot.img
rootfs        Image/rootfs.img
#recovery     Image/recovery.img
#oem              Image/oem.img
#userdata:grow     Image/userdata.img
#misc         Image/misc.img
# 要写入backup分区的文件就是自身（update.img）
# SELF 是关键字，表示升级文件（update.img）自身
# 在生成升级文件时，不加入SELF文件的内容，但在头部信息中有记录
# 在解包升级文件时，不解包SELF文件的内容。
#backup       RESERVED
#update-script  update-script
#recover-script recover-script
```

5.4.3 烧写固件

rockdev 和 IMAGE 目录下，都会生成 update_ab.img，该固件用于烧写。根据需求修改该文件 tools/linux/Linux_Pack_Firmware/rockdev/rk3308-package-file-ab 文件。如下图：

```
# NAME          Relative path
#
#HWDEF         HWDEF
package-file    package-file
bootloader    Image/MiniLoaderAll.bin
parameter     Image/parameter.txt
trust         Image/trust.img
uboot         Image/uboot.img
boot_a        Image/boot.img
boot_b        Image/boot.img
system_a      Image/rootfs.img
system_b      Image/rootfs.img
oem           Image/oem.img
userdata:grow     Image/userdata.img
# 要写入backup分区的文件就是自身（update.img）
# SELF 是关键字，表示升级文件（update.img）自身
# 在生成升级文件时，不加入SELF文件的内容，但在头部信息中有记录
# 在解包升级文件时，不解包SELF文件的内容。
backup        RESERVED
#update-script  update-script
#recover-script recover-script
~
```