Rockchip

recovery 开发指南

发布版本:**1.0.0**

日期:**2018.09**

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 © 2018 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： www.rock-chips.com

前言

概述

本文档主要介绍 Rockchip 处理器 OTA 升级时的 recovery 开发流程以及技术细节。
本文中详细介绍了该方案的开发过程以及注意事项。

产品版本

芯片名称	内核版本
RK3308	4.4

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
2018.09.18	1.0.0	Chad.ma	初始版本

目录

插图目录

表格目录

1 OTA 升级

1.1 概述

OTA（Over-the-Air）即空间下载技术。 OTA 升级是 Android 系统提供的标准软件升级方式。它功能强大，可以无损失升级系统，主要通过网络，例如 WIFI、3G/4G 自动下载 OTA 升级包、自动升级，也支持通过下载 OTA 升级包到 SD 卡/U 盘升级，OTA 的升级包非常的小，一般几 M 到十几 M。

本文主要介绍了使用 OTA 技术升级时，本地升级程序 recovery 执行升级的流程及技术细节，方面用户在开发过程中了解升级的过程及注意事项。

1.2 编译

rootfs 主系统

rootfs 要打开 recoverySystem 的支持，configs 文件中把 BR2_PACKAGE_RECOVERYSYSTEM=y 选上。

或者配置 BR2_PACKAGE_UPDATE=y。

注意：

目前主系统中实现调用升级功能的有两套代码，recoverySystem 与 update，二者使用相同的参数，均可实现进入 recovery 模式，进行 OTA 的升级。

后续将统一使用 update。

Recovery

系统根目录下执行

```
$ ./build.sh recovery
```

会生成文件 buildroot/output/rockchip_rk3308_recovery/images/recovery.img。

```
$ ./mkfirmware.sh
```

会将生成的固件拷贝至 rockdev/目录下。

与 recovery 相关的主要源码路径：

external/recovery/ ：主要生成 recovery 二进制 bin 程序，recovery 模式下的关键程序。

external/rkupdate/：主要生成 rkupdate 二进制 bin 程序，解析 update.img 固件中各个分区数据，并执行对各分区执行升级的关键程序。

若有修改以上两个目录中的源码文件之后的编译方法：

1. Source envsetup.sh
2. 选择某一平台的 recovery 配置
3. make recovery-rebuild / make rkupdate-rebuild
4. ./build.sh recovery
5. ./mkfirmware.sh
6. 烧写 recovery.img

1.3 升级流程

升级固件准备

使用 rockchip 固件打包工具生成的 update.img。

固件的打包可参考《Rockchip Linux 升级固件打包指南》。

升级过程

- 将升级固件 update.img 放在 SD 卡或 U 盘根目录或者设备的/userdata 目录下。
- Normal 系统下执行升级程序 **recoverySystem ota /xxx/update.img**，设备会进入 recovery 模式，并进行升级。

  可使用的路径如下：

  U 盘的挂载路径：/udisk

  sdcard 的挂载路径：/mnt/sdcard/ 或/sdcard

  flash 的挂载路径：/userdata/
- 升级成功后会 reboot 到正常的 normal 系统。

升级流程图

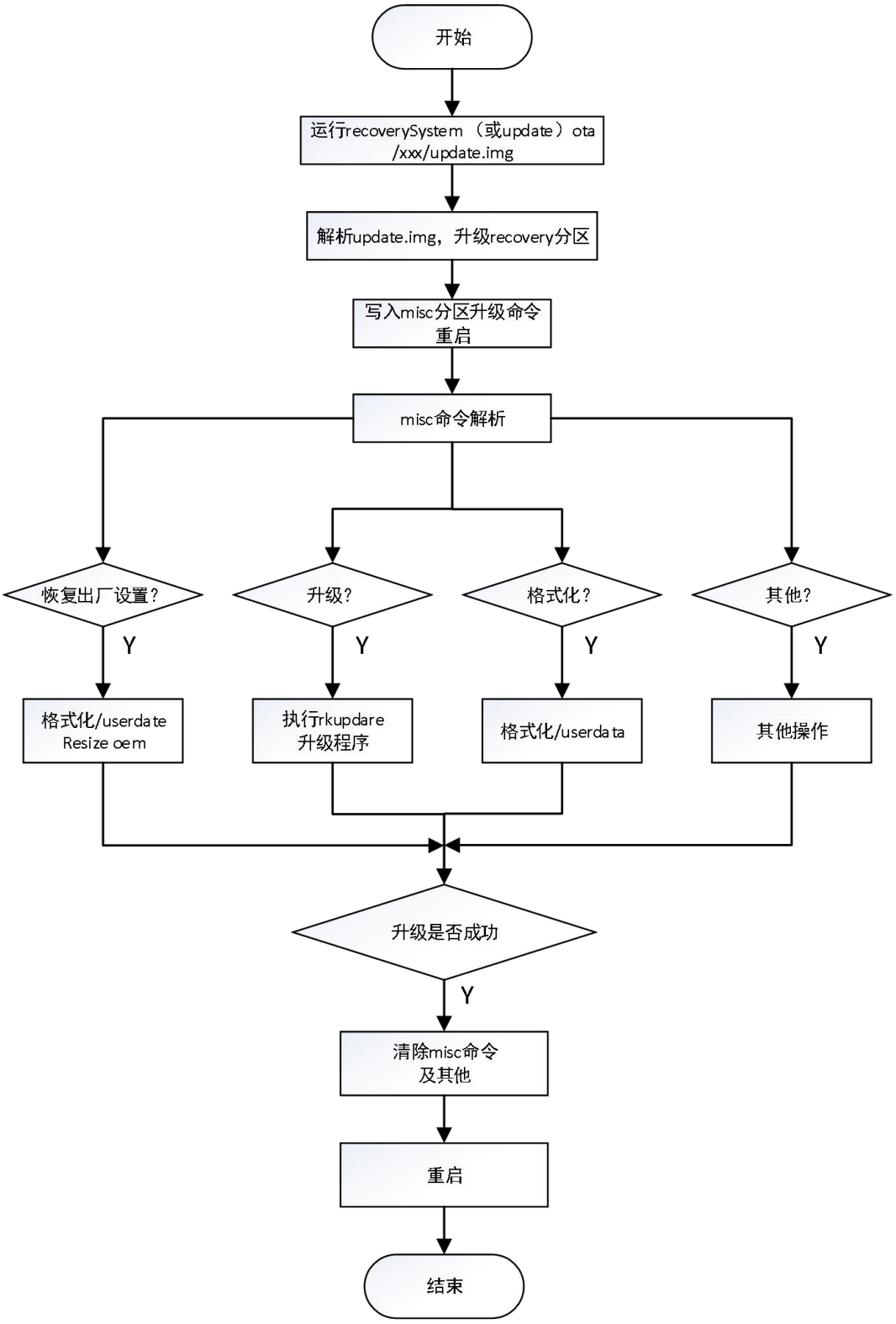

图 1 - 1 recovery 升级流程图

升级的详细流程，开发者可根据流程图，阅读升级方案的源码，这里不做进一步的展开说明。

1.4 注意事项

- 打包 update.img 固件时需要注意，升级固件不一定要全分区升级，可修改 package-file 文件，将不要升级的分区去掉，这样可以减少升级包（update.img ）的大小。

- package-file 中 **recovery.img 如果打包进去的话，不会在 recovery 模式中升级**，为了预防升级 recovery.img 过程中掉电导致后面其他分区无法正常升级的问题，该分区升级放在 normal 系统下升级，即，执行 recoverySystem 命令时会先检测 update.img 升级包中是否有打包 recovery.img，若有则升级 recovery 分区，再进入 recovery 模式升级其他分区固件。
- **misc 分区不建议打包进 update.img** 中，即使有打包进去，也会在升级程序中加载判断到而忽略该分区，原因是：即使升级了 misc 分区，升级成功后 recovery 程序仍会清空 misc 分区中所有的命令及参数，从而导致预想的结果达不到。
- 如果将 **update.img 升级包放置在 flash 中的 userdata 分区，则需要保证 package-file 中不包括 userdata.img 被打包进去**，原因是可能会导致文件系统的损坏，升级成功后可能使 oem 或 userdata 分区 mount 不成功。若从 SD 卡或 U 盘升级时，可以打包 userdata.img，从而对 userdata 分区进行升级。升级完成后会对 userdata 分区重新 resize 操作。

2 运行调试

2.1 Recovery 模式中 log 的查看

- buildroot/output/rockchip_rk3308_recovery/target 目录下

```
$ touch .rkdebug
```

创建这个隐藏文件，可将 recovery 模式中升级的 log 在串口中打印出来。

```
mlc@SYS3:~/3308_dev/buildroot/output/rockchip_rk3308_recovery/target$ ls -al
total 124
drwxr-xr-x 20 mlc mlc  4096 Sep 18 15:36 .
drwxrwxr-x  6 mlc mlc  4096 Sep 18 15:36 ..
drwxr-xr-x  2 mlc mlc  4096 Sep 18 15:36 bin
-rw-r--r--  1 mlc mlc 30195 Sep 11 10:59 busybox.config
lrwxrwxrwx  1 mlc mlc     8 Aug 27 10:56 data -> userdata
drwxr-xr-x  4 mlc mlc  4096 Aug 27 10:59 dev
drwxr-xr-x 13 mlc mlc  4096 Sep 18 15:36 etc
-rwxr-xr-x  1 mlc mlc   178 Aug 27 10:59 init
drwxr-xr-x  3 mlc mlc  4096 Sep 18 15:36 lib
lrwxrwxrwx  1 mlc mlc     3 Aug 27 10:36 lib32 -> lib
lrwxrwxrwx  1 mlc mlc    11 Aug 27 10:47 linuxrc -> bin/busybox
drwxr-xr-x 10 mlc mlc  4096 Aug 27 10:57 media
lrwxrwxrwx  1 mlc mlc    23 Sep 11 10:59 misc -> /dev/block/by-name/misc
drwxr-xr-x  3 mlc mlc  4096 Aug 27 10:56 mnt
drwxr-xr-x  2 mlc mlc  4096 Aug 27 10:56 oem
drwxr-xr-x  2 mlc mlc  4096 Aug 24 11:59 opt
drwxr-xr-x  2 mlc mlc  4096 Aug 24 11:59 proc
drwxr-xr-x  3 mlc mlc  4096 Aug 27 10:56 res
-rw-rw-r--  1 mlc mlc     0 Aug 27 17:44 .rkdebug
drwx------  2 mlc mlc  4096 Aug 24 11:59 root
drwxr-xr-x  2 mlc mlc  4096 Aug 24 11:59 run
drwxr-xr-x  2 mlc mlc  4096 Sep 18 15:36 sbin
lrwxrwxrwx  1 mlc mlc    10 Aug 27 10:56 sdcard -> mnt/sdcard
drwxr-xr-x  2 mlc mlc  4096 Aug 24 11:59 sys
-rw-r--r--  1 mlc mlc  1336 Sep 18 15:36 THIS_IS_NOT_YOUR_ROOT_FILESYSTEM
-rw-r--r--  1 mlc mlc    44 Sep 18 15:36 timestamp
drwxrwxrwt  2 mlc mlc  4096 Aug 24 11:59 tmp
lrwxrwxrwx  1 mlc mlc    10 Aug 27 10:56 udisk -> media/usb0
drwxr-xr-x  2 mlc mlc  4096 Aug 27 10:56 userdata
```

图 2 - 1 recovery 中创建隐藏文件

- 通过查看 userdata/recovery/Log 文件查看

升级之后，在设备 userdata/recovery 目录中查看 log 文件。

```
$ cat userdata/recovery/Log
```

3 附录

3.1 misc 分区说明

misc 其实是英文 miscellaneous 的前四个字母，杂项、混合体、大杂烩的意思。

misc 分区的概念来源于 Android 系统，Linux 系统中常用来作为系统升级时或者恢复出厂设置时使用。

misc 分区的读写：misc 分区在以下情况下会被读写。

1) Uboot：设备加电启动时，首先启动 uboot，在 uboot 中会读取 misc 分区的内容。根据 misc 分区中 command 命令内容决定是进入正常系统还是 recovery 模式。
   Command 为 boot-recovery，则进入 recovery 模式。
   Command 为空，则进入正常系统。

2) Recovery：在设备进入 recovery 模式中，可以读取 misc 分区中 recovery 部分的内容，从而执行不同的动作。或升级或擦除用户数据等等。

Misc 分区的结构及内容：
Misc 分区的结构组成详见下图。

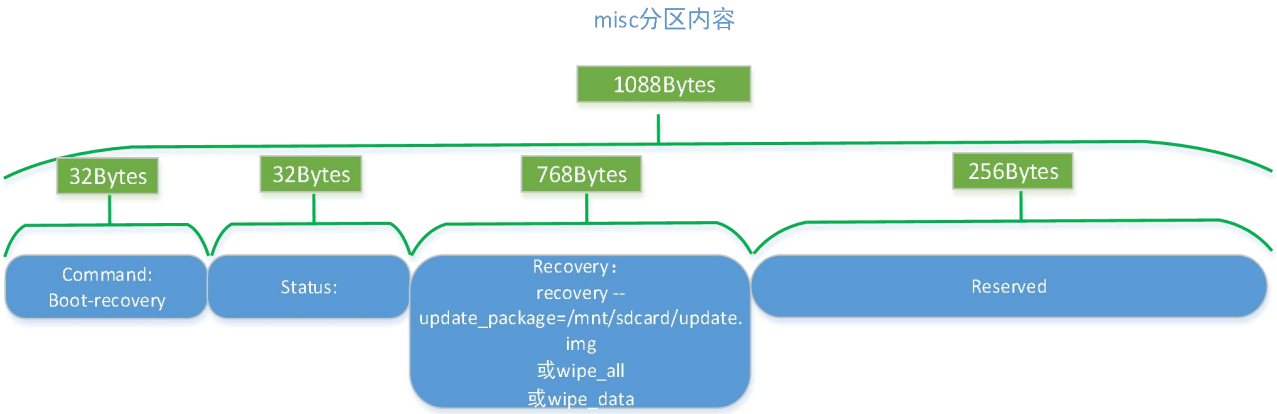

图 3 - 1 misc 分区结构内容

下面以 3308 平台使用的 misc 分区为例，使用 winhex 或 ultraEdit 等工具，以二进制形式打开 misc.img 文件，在距文件开始位置偏移 16K（16384 Byte）字节位置处开始，存放 BootLoader Msg 结构体的内容。

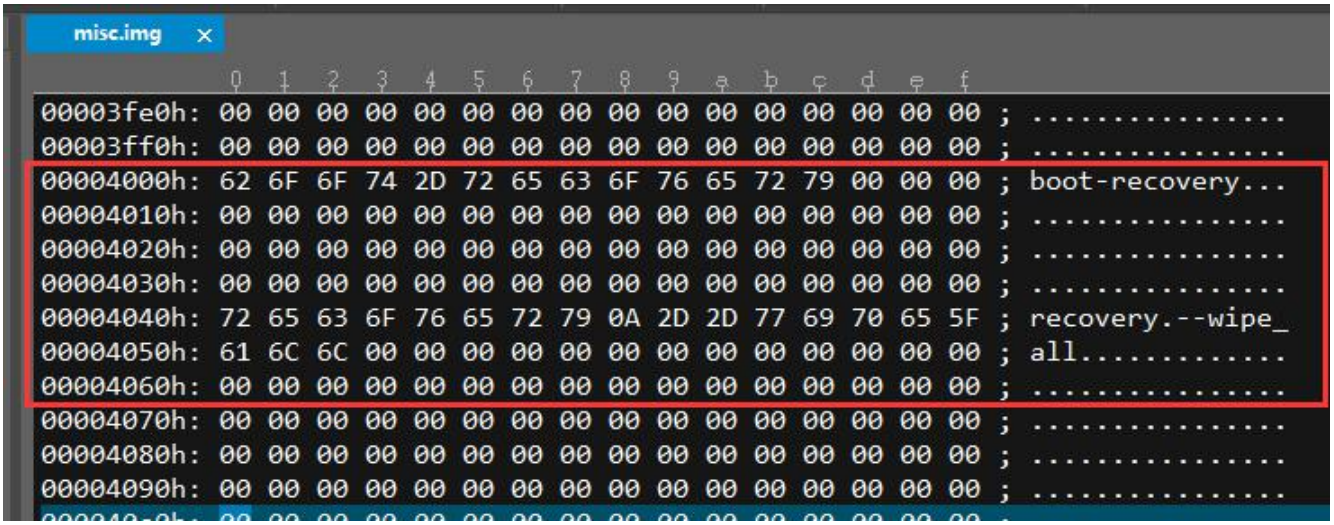

图 3 - 2 misc.img 文件内容

Recovery 中支持的命令部分，可参考 external/recovery/recovery.c 中 OPTIONS 结构中内容。

3.2 Recovery 不同场景下的使用

- 第一次开机

烧写过 misc.img、recovery.img 的机器会进入第一次开机流程。
串口有如下 log 打印:

```
I:Boot command: boot-recovery
  I:Got arguments from boot message
  Command: "recovery" "--wipe_all"
  format '/dev/block/by-name/userdata' to ext2 filesystem
  executing '/sbin/mke2fs'
  executed '/sbin/mke2fs' done
  executed '/sbin/mke2fs' return 0
  executing '/sbin/e2fsck'
  e2fsck 1.43.9 (8-Feb-2018)
  Pass 1: Checking inodes, blocks, and sizes
  Pass 2: Checking directory structure
  Pass 3: Checking directory connectivity
  Pass 4: Checking reference counts
  Pass 5: Checking group summary information
  /dev/block/by-name/userdata: 11/2304 files (0.0% non-contiguous), 82/2299 blocks
  executed '/sbin/e2fsck' done
  executed '/sbin/e2fsck' return 0
  executing '/usr/sbin/e2fsck'
  e2fsck 1.43.9 (8-Feb-2018)
  Pass 1: Checking inodes, blocks, and sizes
  Pass 2: Checking directory structure
  Pass 3: Checking directory connectivity
  Pass 4: Checking reference counts
  Pass 5: Checking group summary information
```

```
/dev/block/by-name/oem: 18/2448 files (0.0% non-contiguous), 513/16384 blocks
executed '/usr/sbin/e2fsck' done
executed '/usr/sbin/e2fsck' return 1
executing '/usr/sbin/resize2fs'
resize2fs 1.43.9 (8-Feb-2018)
The filesystem is already 16384 (1k) blocks long.  Nothing to do!
executed '/usr/sbin/resize2fs' done
executed '/usr/sbin/resize2fs' return 0
```

- 恢复出厂设置

命令行运行 recoverySystem 程序，机器会进入 recovery，并进行格式化，格式化完成之后会自动进入 normal system。

```
$ recoverySystem
```

串口会有如下 log 打印：

```
I:Boot command: boot-recovery
    I:Got arguments from boot message
    Command: "recovery" "--wipe_data"
    format '/dev/block/by-name/userdata' to ext2 filesystem
    executing '/sbin/mke2fs'
    [    4.692437] vendor storage:20160801 ret = -1
    [    6.030842] phy phy-ff008000.syscon:usb2-phy@100.0: charger =
USB_SDP_CHARGER
    [   10.891460] random: nonblocking pool is initialized
    executed '/sbin/mke2fs' done
    executed '/sbin/mke2fs' return 0
    executing '/sbin/e2fsck'
    e2fsck 1.43.9 (8-Feb-2018)
    Pass 1: Checking inodes, blocks, and sizes
    Pass 2: Checking directory structure
    Pass 3: Checking directory connectivity
    Pass 4: Checking reference counts
    Pass 5: Checking group summary information
    /dev/block/by-name/userdata: 11/2304 files (0.0% non-contiguous), 82/2299 blocks
    executed '/sbin/e2fsck' done
    executed '/sbin/e2fsck' return 0
    [   11.141033] cpu0 limit freq=816000 min=816000 max=816000
    [   11.141773] rknand_shutdown...
    [   11.142139] nand th quited
    [   11.142484] rk_ftl_de_init 0
    [   11.150824] rknand_shutdown:OK
    [   11.152054] reboot: Restarting system
```

- 升级

命令行运行 recoverySystem ota /xxx/update.img，机器会进入 recovery，并进行升级。

```
$    recoverySystem ota /udisk/update.img
```

这里以从 U 盘升级为例。

串口可能打印的 log 如下：

```
I:Boot command: boot-recovery
    I:Got arguments from boot message
    Command: "recovery" "--update_package=/udisk/update.img"
     。。。。
   librkupdate_ui_print = parameter writing....
   ########## RKA_Gpt_Download #########
   librkupdate_###### Download trust ... #######
   ui_print = trust writing....
   librkupdate_###### Download uboot ... #######
   ui_print = uboot writing....
   librkupdate_###### Download boot ... #######
   librkupdate_ui_print = boot writing....
   librkupdate_###### Download rootfs ... #######
   librkupdate_ui_print = rootfs writing....
   librkupdate_######Ignore recovery download ######
   librkupdate_###### Download oem ... #######
   ui_print = oem writing....
   librkupdate_###### Download userdata:grow ... #######
    ui_print = parameter checking....
    ui_print = trust checking....
    ui_print = uboot checking....
    ui_print = boot checking....
    ui_print = rootfs checking....
    ui_print = oem checking....
     ui_print = userdata:grow checking....
     librkupdate_###### Ignore recovery Check ######
   librkupdate_Finish to upgrade firmware.
    [   38.655387] EXT2-fs (rknand0p8): warning: mounting unchecked fs, running
e2fsck is recommended
    [   38.663735] cpu0 limit freq=816000 min=816000 max=816000
    [   38.664552] rknand_shutdown...
    [   38.664865] nand th quited
    [   38.665127] rk_ftl_de_init 0
    [   38.676436] rknand_shutdown:OK
    [   38.678078] reboot: Restarting system
```