Rockchip_Developer_Guide_Android_AB_Sys tem_Upgrading_CN

文件标识: RK-SM-YF-208

发布版本: V1.1.0

日期: 2023-02-06

文件密级:□绝密 □秘密 □内部资料 ■公开

免责声明

本文档按"现状"提供,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 © 2021 瑞芯微电子股份有限公司

超越合理使用范畴,非经本公司书面许可,任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部,并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

网址: <u>www.rock-chips.com</u>

客户服务电话: +86-4007-700-590

- 客户服务传真: +86-591-83951833
- 客户服务邮箱: fae@rock-chips.com

前言

概述

本文档描述了Rockchip A/B系统的使用说明。

产品版本

芯片名称	内核版本
RK3326/RK3399/RK356x/RK3588	4.19, 5.10

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

版本号	作者	修改日期	修改说明
V1.0.0	纪大峣	2022-02-25	初始版本
V1.1.0	纪大峣	2023-02-06	补充Android 13配置说明

Rockchip_Developer_Guide_Android_AB_System_Upgrading_CN

- 1. 概述
- 2. 系统配置说明
 - 2.1 Android系统配置说明
 - 2.1.1 Android 12系统配置
 - 2.1.2 Android 13系统配置
 - 2.2 kernel dts配置
 - 2.3 uboot配置
 - 2.4 开发烧写工具增加分区
- 3. AB系统与OTA包编译
- 4. 验证方法(客户端与服务器)
- 5. 注意事项
- 6. A/B miniloader OTA
 - 6.1 rkloader升级包构建
 - 6.2 升级命令参考

1. 概述

本文档描述了Rockchip A/B系统的使用说明,可以在Rockchip Android >= 12平台上使用。 使用该升级方法,所有待升级的分区(除super之外)都有a和b两份,相比传统的Recovery升级方式会大 幅增加存储空间需求。其优点是A/B升级是无缝升级,可以在Android系统运行过程中,根据客户定制的 客户端和服务器之间的交互策略与协议来完成升级。

2. 系统配置说明

在Rockchip Android 平台上, AB系统功能默认关闭, 要使用AB系统, 需要从Android系统、U-BOOT和 kernel dts三个方面进行配置。以下对此进行详细说明。

2.1 Android系统配置说明

2.1.1 Android 12系统配置

Android 12系统的配置包括:

1.在device\rockchip\rkxxx\BoardConfig.mk中打开**AB**升级配置项。 将BOARD_USES_AB_IMAGE配置设置为true,默认为false. 以rk3326为例:

vim device/rockchip/rk3326_s/BoardConfig.mk
#AB image definition
-BOARD_USES_AB_IMAGE := false
+BOARD_USES_AB_IMAGE := true

注意:如果要开启虚拟AB功能(不建议),则需要同时配置BOARD_USES_AB_IMAGE 和 BOARD_ROCKCHIP_VIRTUAL_AB_ENABLE为true,即: -BOARD_USES_AB_IMAGE := false -BOARD_ROCKCHIP_VIRTUAL_AB_ENABLE := false +BOARD_USES_AB_IMAGE := true +BOARD_ROCKCHIP_VIRTUAL_AB_ENABLE := true

2.在对应的device\rockchip\rkxxx目录下,确认是否已经有recovery.fstab_AB文件,如果已有,则直接跳过这一步;如果没有recovery.fstab_AB文件,则按如下步骤执行:

(1)新增针对AB的recovery fstab文件recovery.fstab_AB 与对应的fstab.rk30board的主要区别在于AB分区增加slotselect挂载参数(system/vendor/odm/product增加 slotselect参数),同时将exernal_sd, frp, parameter, baseparameter, resource的分区节点添加进去,并更改 data区的挂载方式选项。

一个参考文件如下:

device\rockchip\rk3399\rk3399_Android12\recovery.fstab_AB:



(2) device\rockchip\rkxxx下的BoardConfig中导入AB配置,包括TARGET_RECOVERY_FSTAB,使其指向刚刚创建的recovery.fstab_AB文件。

```
device\rockchip\rk3399\rk3399 Android12\BoardConfig.mk:
diff --git a/ rk3399/rk3399 Android12/BoardConfig.mk b/
rk3399/rk3399_Android12/BoardConfig.mk
index 1e78940..159a3b6 100755
--- a/ rk3399/rk3399 Android12/BoardConfig.mk
+++ b/ rk3399/rk3399 Android12/BoardConfig.mk
+# AB image definition
+BOARD_USES_AB_IMAGE := true
+BOARD ROCKCHIP VIRTUAL AB ENABLE := false
+
+ifeq ($(strip $(BOARD USES AB IMAGE)), true)
    include device/rockchip/common/BoardConfig AB.mk
     TARGET_RECOVERY_FSTAB := device/rockchip/
^{+}
rk3399/rk3399_Android11/recovery.fstab_AB
+endif
```

2.1.2 Android 13系统配置

Android 13系统的配置包括:

```
1.在device\rockchip\rkxxx\BoardConfig.mk中打开AB升级配置项。
将BOARD_USES_AB_IMAGE配置设置为true,默认为false.
以rk3326为例:
```

vim device/rockchip/rk3326_s/BoardConfig.mk
#AB image definition
-BOARD_USES_AB_IMAGE := false
+BOARD_USES_AB_IMAGE := true

注意:

(1)如果要开启虚拟AB功能(不建议,对升级性能有影响),则需要同时配置
BOARD_USES_AB_IMAGE 和BOARD_ROCKCHIP_VIRTUAL_AB_ENABLE为true,即:
-BOARD_USES_AB_IMAGE := false
-BOARD_ROCKCHIP_VIRTUAL_AB_ENABLE := false
+BOARD_USES_AB_IMAGE := true
+BOARD_ROCKCHIP_VIRTUAL_AB_ENABLE := true

 (2)从Android 13开始支持压缩的虚拟AB功能,如果要开启压缩的虚拟AB功能(不建议,对升级性能 有影响),则需要同时配置BOARD_USES_AB_IMAGE、
 BOARD_ROCKCHIP_VIRTUAL_AB_ENABLE和BOARD_ROCKCHIP_VIRTUAL_AB_COMPRESSION 为true,即: -BOARD_USES_AB_IMAGE := false -BOARD_ROCKCHIP_VIRTUAL_AB_ENABLE := false +BOARD_USES_AB_IMAGE := true +BOARD_ROCKCHIP_VIRTUAL_AB_ENABLE := true +BOARD_ROCKCHIP_VIRTUAL_AB_COMPRESSION := false

2.在对应的device\rockchip\rkxxx目录下,确认是否已经有recovery.fstab_AB文件,如果已有,则直接跳过这一步;如果没有recovery.fstab_AB文件,则按如下步骤执行:

(1)新增针对AB的recovery fstab文件recovery.fstab_AB

与对应的fstab.rk30board的主要区别在于AB分区增加slotselect挂载参数(system/vendor/odm/product增加 slotselect参数),同时将exernal_sd, frp, parameter, baseparameter, resource的分区节点添加进去,并更改 data区的挂载方式选项。

一个参考文件如下:

device\rockchip\rk3399\rk3399_Android12\recovery.fstab_AB:

# Android fstab file.					
# <src></src>	<mnt_point></mnt_point>	<type></type>	<pre><mnt_flags and="" o<="" pre=""></mnt_flags></pre>	ptions>	<fs_mgr_flags></fs_mgr_flags>
# The filesystem that contains t	the filesystem checker binary	(typically /sy	/stem) cannot		
<pre># specify MF_CHECK, and must com</pre>	ne before any filesystems that	do specify M	_CHECK		
system /system ext4 ro,barrier=1	L wait,slotselect,logical,firs	t_stage_mount			
vendor /vendor ext4 ro,barrier=1	L wait,slotselect,logical,firs	t_stage_mount			
odm /odm ext4 ro,barrier=1	L wait,slotselect,logical,firs	t_stage_mount			
product /product ext4 ro,b	oarrier=1 wait,slotselect,logi	cal,first_sta	ge_mount		
system_ext /system_ext ext	t4 ro,barrier=1 wait,slotselec	t,logical,firs	st_stage_mount		
/dev/block/by-name/metadata /met	tadata ext4 nodev,noatime,nosu	id,discard,sy	nc wait,formattable,	first_stage_mount	
/dev/block/by-name/misc	/misc emmc	defaults	defaults		
/dev/block/by-name/cache	/cache ext4	noatime, no	diratime,nosuid,node	v,noauto_da_alloc,discard	wait,check
(dev. (b)] = -1; (b)]+0;-1		1		(1-61+-
/dev/block/mmcblk0p1	/E	/mnt/ext	ternal_sd vrat	/dev/block/mmcblku	deraults
/dev/block/by-name/trp	/тгр	emmc	deraults	deraults	
/dev/block/by-name/baseparamer	/baseparamer	emmc	derautts	derautus	
/dev/block/by-name/backup	/васкир	emmc	deraults	deraults	
/dov/block/znam0	nona	CW2D	dofaults		zromsizo-50%
/dev/brock/21 allio	none	Swap	ueraures		21 allis 126-30%
/dev/block/by-name/userdata	e teb/	fofe	defaults	defaults	
~	/ occes			acriati co	
~					

(2) device\rockchip\rkxxx下的BoardConfig中导入AB配置,包括TARGET_RECOVERY_FSTAB,使其指向刚刚创建的recovery.fstab_AB文件。

```
device\rockchip\rk3399\rk3399 Android12\BoardConfig.mk:
diff --git a/ rk3399/rk3399 Android12/BoardConfig.mk b/
rk3399/rk3399_Android12/BoardConfig.mk
index 1e78940..159a3b6 100755
--- a/ rk3399/rk3399 Android12/BoardConfig.mk
+++ b/ rk3399/rk3399 Android12/BoardConfig.mk
+# AB image definition
+BOARD USES AB IMAGE := true
+BOARD ROCKCHIP VIRTUAL AB ENABLE := false
+
+ifeq ($(strip $(BOARD USES AB IMAGE)), true)
    include device/rockchip/common/BoardConfig AB.mk
    TARGET RECOVERY FSTAB := device/rockchip/
+
rk3399/rk3399 Android11/recovery.fstab AB
+endif
```

2.2 kernel dts 配置

对于Android >=12来说, kernel不需要任何配置。

2.3 uboot 配置

在uboot中,针对具体芯片的配置文件,添加CONFIG_ANDROID_AB=y配置项,参考配置如下截图所示:



2.4 开发烧写工具增加分区

下载工具增加B分区的下载项,同时所有AB分区都烧写相同的固件。

对于>=Android 12来说,根据"3.AB系统与OTA包编译"指令完成编译后,在rockdev/xxxx/下自动产生 config.cfg文件,将该配置文件替换RKDevTool工具下的同名文件,然后再重新打开RKDevTool即可,不 需要手动配置。

一个烧写工具截图如下:

#		++h++	夕宁	收久	-6
# 1		1611 1611	197 Loader	RHII RIVE some VARVeb3399 ARVARVAVE dish	
2	1	0×00000000	Parameter	F:\F_sync_\AB\rk3399_AB\AB\AVB_diab	
3	T	0x00004000	Uboot a	F:\F svnc \AB\rk3399 AB\uboot debug	
4		0x00006000	Vboot b	F:\F sync \AB\rk3399 AB\uboot debug	=
5		0x00008000	 trust_a	F:\F_sync_\AB\rk3399_AB\AB\AVB_diab	
6		0x0000A000	trust_b	F:\F_sync_\AB\rk3399_AB\AB\AVB_diab	
7		0x0000C000	Misc	F:\F_sync_\AB\rk3399_AB\AB\AVB_diab	
8		0x00028000	Dtbo_a	F:\F_sync_\AB\rk3399_AB\AB\AVB_diab	
9		0x0002A000	Dtbo_b	F:\F_sync_\AB\rk3399_AB\AB\AVB_diab	
10		0x0002C000	vbmeta_a	F:\F_sync_\AB\rk3399_AB\AB\AVB_diab	
11	$\mathbf{\nabla}$	0x0002C800	vbmeta_b	F:\F_sync_\AB\rk3399_AB\AB\AVB_diab	
12	$\mathbf{\nabla}$	0x0002D000	Boot_a	F:\F_sync_\AB\rk3399_AB\Non\Image=r	
13	$\mathbf{\nabla}$	0x0004D000	Boot_b	F:\F_sync_\AB\rk3399_AB\Non\Image=r	-
4 m					

注意,对于rk3566/rk3568/rk3588等芯片来说,没有trust分区,则不需要trust_a和trust_b分区。

3. AB系统与OTA包编译

编译AB系统固件步骤(务必按如下步骤执行): lunch xxx make installclean -j16 OR make clean -j16 make -j16 make dist -j16 mkimage_ab.sh ota 注意:

1.开启AB后,第一次需要make clean后再编译。 2.差异包的编译方法与非AB一致。

4. 验证方法(客户端与服务器)

在无缝升级过程中,升级包可以一边下载,一边升级。这时候需要有一个HTTP服务器和一个升级客户端。

请使用以下update_device.py验证。至于产品化的升级客户端和升级服务器需要客户自行搭建,升级客户端可以参考Android默认提供的update_engine_client。关于update_engine_client的使用方法请参考如下验证方法update_device.py,该脚本就是通过adb最终调用update_engine_client来实现升级的。

验证方法: update_device.py

update_device.py脚本通过adb方式,将主机变成HTTP服务器,然后调用update_engine_client来实现无缝升级。

使用方法如下:

在主机中执行如下命令(请确保该主机adb功能和python环境正常):

system/update_engine/scripts/update_device.py {升级包名字}

示例如下:

]dy@]dy-Latitude-E6440:/media/jdy/LDSecond/rK3326/rK3326_8.X
	jdy@jdy-Latitude-E6440:/media/jdy/LDSecond/rk3326/rk3326_8.X\$ sudo system/update_engine/scripts/update_device.py ~/rk3326_evb-ota-eng.jdy.zi
Ð	[sudo] idy 的密码:
	INFO: cont: Pupping: add reverse tcp:1234 tcp:37213
	TNEOLOOCT:Duning add chall su & undate enjine clientundatefollowpavload=http://127.0.0.1:1234/pavloadoffset=5501size=451914j
	had a set is used and sheet as a spear set in the sheet of the part of the part of the set of the s
	METADATA_HASH=S/dt)40HgFLYp/acitiotekoojaursouwerus4byyco=
	METADATA_SIZE=59119
	USER_AGENT=Dalvik (something, something)
	NETWORK_ID=0
1	
	<pre>[INF0:update_engine_client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_IDLE (0), 0)</pre>
-	[INF0:update_engine_client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_UPDATE_AVAILABLE (2), 0)
	127.0.0.1 [17/Jul/2018 17:24:40] "GET /payload HTTP/1.1" 206 -
1	INFO:root:Serving request for /payload from /home/jdy/rk3326_evb-ota-eng.jdy.ztp [5501, 451920072) tength: 4519145/1
	[INFO:update_engine_client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 4.842296-05)
20	[INFO:update_engine_client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.0100547)
5	[INFO:update_engine_client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.020001)
	[INFO:update_engine_client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.03000/3)
	[INFO:update_engine_client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.0400735)
1	[INFO:update engine_client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.0500798)
	[INFO:update_engine_client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.0000001)
	[INF0:update_engine_client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.0700924)
	[INFO:update_engine_client_android.cc(90)] onStatusUpdate(UPDATE_SIATUS_DUMHLOADING (3), 0.0000907)
1	[INFO:update_engine_client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.0000047)
1	[INFO:update_engine_client_android.cc(90)] onStatusUpdate(UPDAte_SIATUS_UUMALOADING (5), 0.100111)
	[INFO:update_engine_client_android.cc(90)] onstatusUpdate(update_units_status_boling(3), 0.120124)
	[INFO:update_engine_client_android.cc(90)] onstatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.13013)
Ľ	[INFO:update_engine_client_android.cc(90)] UnstatusUndate(UPDATE_STATUS_DOWNLOADING (3), 0.140136)
	[INFO:update engine citent_and otd.cc(00)] onStatusUpdate(UPDATE STATUS_DOWNLOADING (3), 0.150143)
đ	[INFO: update engine_client and old cc(90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.160149)
	INFO: update_engthe_client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.170155)
R	[INFO:update_engine_client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.180161)
	Line oupdate engine client android.cc(90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.190108)
Ĩ	[INFO:update engine client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.2001/4)
	[INFO:update_engine_client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (5), 0.22010)
	[INFO:update_engine_client_android.cc(90)] onStatusUpdate(UPDATE_SIATUS_DOWNLOADING (3), 0.230193)
1	[INFO:update_engine_client_android.cc(90)] onstatusUpdate(upnate_STATUS_DOWNLOADING (3), 0.240199)
	[INFO:update_engine_client_android.cc(90)] onstatusUpdate(Update_STATUS_DOWNLOADING (3), 0.250205)

<pre>[INF0:update_engine_client_android.cc(90)] [INF0:update_engine_client_android.cc(90)]</pre>	onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.230193) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.240199) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.260205) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.260212) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.260224) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.280224) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.280224) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.380237) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.380237) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.330243) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.330256) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.380256) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.380287) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.380281) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.380281) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.380281) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.380287) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.380283) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.390293) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.390293) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.4803) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.4803) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.4803) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.4803) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.4803) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.4803) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.48030) onStatu
<pre>[INFO:update_engine_Client_android.cc(90)] [INFO:update_engine_Client_android.cc(90)]</pre>	OnStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.420312) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.420312) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.440325) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.440337) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.440337) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.440337) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.440337) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.440337) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.440344) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.440356) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.540362) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.550362) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.520375) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.530381) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.550394) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.5604) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.5604) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.5604) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.5704060) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.5704061) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.580413) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.580413)
<pre>[INF0:update_engine_Client_android.cc(90)] [INF0:update_engine_Client_android.cc(90)] [INF0:update_engine_Client_android.cc(90)] [INF0:update_engine_Client_android.cc(90)] [INF0:update_engine_Client_android.cc(90)] [INF0:update_engine_Client_android.cc([INF0:update_engine_Client_android.cc([INF0:update_engine_Client_android.cc([INF0:update_engine_Client_android.cc([INF0:update_engine_Client_android.cc([INF0:update_engine_Client_android.cc([INF0:update_engine_Client_android.cc([INF0:update_engine_Client_android.cc([INF0:update_engine_Client_android.cc([INF0:update_engine_Client_android.cc([INF0:update_engine_Client_android.cc([INF0:update_engine_Client_android.cc(]]]</pre>	OnStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.600423) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.630443) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.630444) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.630444) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.630444) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.630444) onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.650457) 90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.790545) 90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.800551) 90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.810557) 90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.820563) 90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.83057) 90)] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.840576)
<pre>[INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc(</pre>	<pre>90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.850582) 90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.860589) 90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.870595) 90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.890607) 90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.900614) 90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.91062) 90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.920626) 90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.920633) 90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.9206363) 90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.940639) 90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.950645) 90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.950645) 90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.970658) 90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.970658) 90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.980664) 90] onStatusUpdate(UPDATE_STATUS_DOWNLOADING (3), 0.980664)</pre>
<pre>[INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc([INF0:update_engine_client_android.cc(</pre>	<pre>90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.01) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.03) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.05) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.08) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.08) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.11) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.12) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.13) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.13) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.14) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.16) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.17) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.17) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.17) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.17) 90)] onStatusUpdate(UPDATE_STATUS_FINALIZING (S), 0.21)</pre>
[INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc([INFO:update_engine_client_android.cc(INFO:update_engine_client_android.cc(90)] onStatusUpdate(UPDATE_STATUS_FIMALIZING (5), 0.23) 90)] onStatusUpdate(UPDATE_STATUS_FIMALIZING (5), 0.23) 90)] onStatusUpdate(UPDATE_STATUS_FIMALIZING (5), 0.25) 90)] onStatusUpdate(UPDATE_STATUS_FIMALIZING (5), 0.26) 90)] onStatusUpdate(UPDATE_STATUS_FIMALIZING (5), 0.29)

[INFO:update engine client android cc(00)]	
[INFO: update engine client android en(00)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.53)
[INFO:update engine client android ac(00)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.55)
[INFO:update engine client android cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.56)
[INFO:update engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.58)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.6)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.61)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.62)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.64)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.65)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.66)
[INFO: update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.68)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.7)
[INFO: update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.71)
[INFO: update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.73)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.74)
[INFO: update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.75)
[INFO: update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.77)
[INFO: update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.78)
[INFO: update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.79)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.81)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.83)
[INFO: update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.84)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.86)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.87)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.88)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.9)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.91)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.92)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.94)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.95)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.96)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.97)
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 0.99)
[INFO:update_engine_client_android.cc(90)]	<pre>onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 1)</pre>
[INFO:update_engine_client_android.cc(90)]	<pre>onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 1)</pre>
[INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_FINALIZING (5), 1)
INFO:update_engine_client_android.cc(90)]	onStatusUpdate(UPDATE_STATUS_UPDATED_NEED_REBOOT (6), 0)
[INFO:update_engine_client_android.cc(98)]	onPayloadApplic ationComplete(ErrorCode::kSuccess (θ))
INFO:root:Running: adb reverseremove tcp	:1234
INFO:root:Server Terminated	
jdy@jdy-Latitude-E6440:/media/jdy/LDSecond/	rk3326/rk3326_8.X\$

上面的截图展示了完整的升级过程,升级成功后,会有UPDATE_STATUS_UPDATED_NEED_REBOOT 地打印信息,如上截图所示。此时手动重启设备,就可以切换到新的升级后的系统。

5. 注意事项

1. 打包生成update.img时要注意	意修改package-file增加对应的分区。
比如trust和uboot配置如下,	其他分区类似。

trust_a Image/trust.img

trust_b Image/trust.img

- uboot_a Image/uboot.img
- uboot_b Image/uboot.img

注意,对于rk3566/rk3568/rk3588等芯片来说,没有trust分区,则不需要trust_a和trust_b分区。

6. A/B miniloader OTA

Miniloader是Rockchip平台的一级引导程序,正常情况下该固件不需要进行OTA升级,并且也不建议这么做,在特殊情况下,如果需要对miniloader进行OTA升级,针对Rockchip A/B系统平台,可采用本节介绍的方案来实现。

需要注意的是该miniloader OTA升级包与正常的A/B升级包没有任何关系,相互独立。也就是说当有需要升级A/B系统中的miniloader时需按照本节6.1介绍的方法生成对应的miniloader升级包,并且客户端需按照6.2的说明进行开发(在升级miniloader升级包时执行6.2的参考指令后会进入recovery模式升级A/B系统的miniloader,升级完成后自动重启)。

6.1 rkloader升级包构建

对于>=Android 12来说,以OTA方式完成固件编译后(即make installcelan && make -j16 && make dist - j16 && ./mkimage_ab.sh ota),在rockdev下自动生成的update_loader.zip文件就是rk loader OTA升级包。

6.2 升级命令参考

升级客户端apk可参考如下命令方式实现:

```
adb root
adb push update.zip /cache/
adb shell "echo \"--fw_rkloader=/cache/update.zip\" > /cache/recovery/command"
adb reboot recovery
```