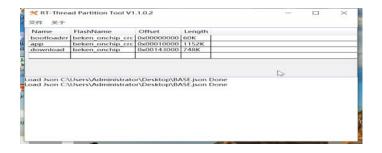
bk7251 基于 RT-THREAD 固件(OTA)升级

(使用不同的存储作为 download 分区的使用方法)

- 一、内部 flash 作为 download 分区,使用步骤如下
- 1. 配置 boot 分区



2. 配置固件打包 config.json

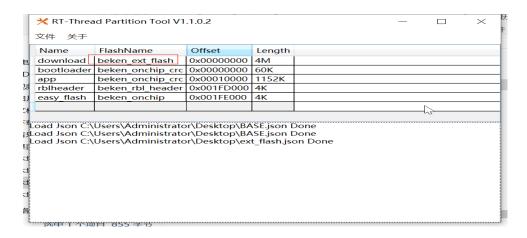
Note:

- 1) 注意 boot 分区和 config.json 中的地址的关系要对应
- 2) boot 区的地址和长度是固定的, app 区的起始地址也是固定的; 其它地址和长度需要用户根据使用的情况配置
- 3) boot 分区和 config.json 中的分区范围不能重叠覆盖!
- 3. 升级

见文档《BK7251_RTOS_SDK_API_Reference_xxx.pdf》

二、外部 flash 作为 download 分区,使用步骤如下

1. 配置 boot 分区



- 1) 将 download 分区的 FlashName 配置为 beken_ext_flash
- 2)easy_flash 和 rblheader 分区根据实际使用芯片的大小配置其起始地址,长度为 4K. 如芯片的 flash 大小为 2M bytes,则 easy_flash 和 rblheader 分别为 0x001FE000, 0x001FD000;如果芯片为 4M bytes,则 easy_flash 和 rblheader 分别为 0x003FE000,0x003FD000,以此类推。
- 2. 配置固件打包 config.json

Note:

- 1) 注意 boot 分区和 config.json 中的地址的关系要对应
- 2) download、rblheader、easy_flash 分区不需要在 config.json 中显式配置。但 boot 分区 和 config.json 中的分区范围不能重叠覆盖!
- 3. 升级

参考例程 E:\rtt_audio\bk7251_sdk\samples\ota\port\ fal_flash_extern_port.c

三、sd 卡作为 download 分区,使用步骤如下

- 1. 配置程序相关参数
 - 1) 根据原理图实际情况配置宏 CFG SD HOST INTF,如果使用 SD 卡的 D0 引脚为

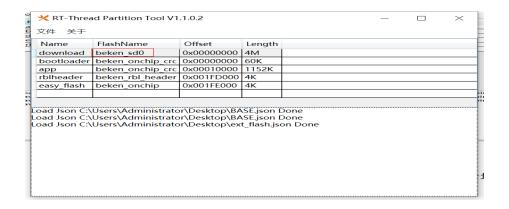
gpio17,则选择 SD_HOST_INTF, 否则选择 SD1_HOST_INTF

```
133:
134: /*section 17----support sta power sleep*/
sys_config.h
                                                        135: #define CFG_USE_STA_PS
136:
     # CFG_USE_SDCARD_HOST
     SD HOST INTF
                                                                  /*section 18----AP support stas in power save*/
                                                         138: #define CFG_USE_AP_PS
      # CFG SD HOST INTE
     # CONFIG APP MP3PLAYER
# CFG SUPPORT OTA HTTP
# CFG SUPPORT OTA_TFTP
# CFG SARADC_CALIBRATE
                                                                  /*section 19----for SDCARD HOST*/
                                                         141: #define CFG_USE_SDCARD_HOST
                                                        141: #define CFG_SE_SDCARD_M
142: //select SD or SD1
143: #define SD_HOST_INTF
144: #define SD1_HOST_INTF
145: #define CFG_SD_HOST_INTF
     # CFG SARADC CALIBRATE
# CFG SYS REDUCE NORMAL P
# CFG LESS MEMERY IN RWNX
# CFG LUSE AUDIO
# CFG USE AUD DAC
# CFG USE AUD DAC
# CFG USE AUD ADC
# CFG USE SPI MASTER
# CFG USE SPI MST FLASH
                                                                                                                                                                SD_HOST_INTF
                                                                  /*section 20 ---- support mp3 decoder*/
                                                                 #define CONFIG_APP_MP3PLAYER
                                                                                                                                                               a
```

2)如果使用芯片引脚 VDDRAM 供电,需要开启图中截图部分

```
main.c (rtt_audio\...\applications) × saradc_intf.c (rtt_audio\...\saradc_intf) sdcard.c (rtt_audio\...\sdcard) sdio_driver.c (rtt_audio\...\sdcard) II Search Results sys_c
                                                 extern const struct romfs_dirent romfs_root;
                                           46: extern const struct rom
47: #define DFS_ROMFS_ROOT
                                                                                           (&romfs_root)
 Symbol Name (Alt+L)
                                            48: extern rt_err_t rt_audio_codec_hw_init(void);
                                           49: extern int player_system_init(void);
  # include "include.h"
  # include include.n
# include "driver_pub.h"
# include "func_pub.h"
# include "app.h"
# include "ate_app.h"
# include "shell.h"
# include <fal.h>
                                           50: extern void user_main( beken_thread_arg_t args );
                                           51:
                                           52: extern int rt_hw_flash_disk_readonly_init(const char *name, uint32
                                           54: int main(int <u>argc</u>, char ** argv)
  # include "sys_ctrl_pub.h"
wlan_app_init
romfs_root
                                           55: {
                                                      saradc_config_vddram_voltage(PSRAM_VDD_3_3V);
  # DES ROMES ROOT
  □ rt_audio_codec_hw_init
□ player_system_init
                                                        * mount ROMFS as root directory */
                                            59: #if defined(RT_USING_DFS_ROMFS)
                                                      if (dfs_mount(RT_NULL, "/", "rom", 0, (const void *)DFS_ROMFS_ROOT) == 0)
                                            60:
  rt hw_flash_disk_readonly_init
                                                           rt_kprintf("ROMFS File System initialized!\n");
  # ifdef BEKEN_USING_WLAN
```

2. 配置 boot 分区



1)如果第一步中使用的是 SD_HOST_INTF,则 FlashName 配置为 beken_sd0,否则为

beken_sd1

- 2) easy_flash 和 rblheader 分区根据实际使用芯片的大小配置其起始地址,长度为 4K. 如芯片的 flash 大小为 2M bytes,则 easy_flash 和 rblheader 分别为 0x001FE000, 0x001FD000;如果芯片为 4M bytes,则 easy_flash 和 rblheader 分别为 0x003FE000,0x003FD000,以此类推。
- 3. 配置固件打包 config.json

Note:

- 1)注意 boot 分区和 config.json 中的地址的关系要对应
- 2) download、rblheader、easy_flash 分区不需要在 config.json 中显式配置。但 boot 分区和 config.json 中的分区范围不能重叠覆盖!

4. 升级

- 1) OTA 下载固件到 sd 卡,或者本地拷贝固件到 sd 卡
- 2)copy sd 卡中固件 rtthread.rbl 的 rblheader 头到内部 flash 指定位置,位置与 boot 配置的 rblheader 分区的起始地址保持一致。(copy 的代码参考 test_flash)
- 3) 软件重启或者硬件重启(进入 boot 完成一系列校验和固件搬运, rblheader 区固件 头的删除, 跳转到应用等工作)

Note:

1) rtthread.rbl 是原始的应用固件使用 beken_packager.exe 打包而成(不需要将

rtthread.rbl 打包成文件系统)。

2) rtthread.rbl 要放在 sd 卡的根目录下

说在后面:

easy_flash 分区不是必需的!

以下两种情况才需要 easy_flash 分区。针对有外部 MOS 管开关机电路以及 OTA 时 led 指示的情况:

比如使用 gpio20 来控制 MOS 管,则在 APP 中下载代码完成后,重启系统之前,先把该信息保存到 easy flash 区:

调用 ef_set_env ("power_control_gpio","20") 以及 ef_save_env()

另外在重启之前,最好提示用户按开机键继续完成升级;

使用 gpio21 来表示正在升级,则在 APP 中下载代码完成后,重启系统之前,先把该信息保存到 easy flash 区:

调用 ef_set_env("led_control_gpio",21)以及 ef_save_env()