US282A_PSP 接口设计说明书

最新版本：1.0

2015-4-30

声 明

Disclaimer

Information given in this document is provided just as a reference or example for the purpose of using Actions' products, and cannot be treated as a part of any quotation or contract for sale.

Actions products may contain design defects or errors known as anomalies or errata which may cause the products' functions to deviate from published specifications. Designers must not rely on the instructions of Actions' products marked "reserved" or "undefined". Actions reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

ACTIONS DISCLAIMS AND EXCLUDES ANY AND ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY AND ALL EXPRESS OR IMPLIED WARRANTIES OF MERCHANTABILITY, ACCURACY, SECURITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY AND THE LIKE TO THE INFORMATON OF THIS DOCUMENT AND ACTIONS PRODUCTS.

IN NO EVENT SHALL ACTIONS BE LIABLE FOR ANY DIRECT, INCIDENTAL, INDIRECT, SPECIAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES WHATSOEVER, INCLUDING, WITHOUT LIMITATION FOR LOST OF DATA, PROFITS, SAVINGS OR REVENUES OF ANY KIND ARISING FROM USING THE INFORMATON OF THIS DOCUMENT AND ACTIONS PRODUCTS. REGARDLESS OF THE FORM OF ACTION, WHETHER BASED ON CONTRACT; TORT; NEGLIGENCE OF ACTIONS OR OTHERS; STRICT LIABILITY; OR OTHERWISE; WHETHER OR NOT ANY REMEDY OF BUYER IS HELD TO HAVE FAILED OF ITS ESSENTIAL PURPOSE, AND WHETHER ACTIONS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR NOT.

Actions' products are not designed, intended, authorized or warranted for use in any life support or other application where product failure could cause or contribute to personal injury or severe property damage. Any and all such uses without prior written approval of an Officer of Actions and further testing and/or modification will be fully at the risk of the customer.

Ways of obtaining information

Copies of this document and/or other Actions product literature, as well as the Terms and Conditions of Sale Agreement, may be obtained by visiting Actions' website at:

http://www.actions-semi.com or from an authorized Actions representative.

Trademarks

The word “Actions” and the logo are the trademarks of Actions Semiconductor Co., Ltd, and Actions (Zhuhai) Technology Co., Limited is authorized to use them. Word“炬芯”is the trademark of Actions (Zhuhai) Technology Co., Limited. Names and brands of other companies and their products that may from time to time descriptively appear in this document are the trademarks of their respective holders, no affiliation, authorization, or endorsement by such persons are claimed or implied except as may be expressly stated therein.

Rights Reserved

The provision of this document shall not be deemed to grant buyers any right in and to patent, copyright, trademark, trade secret, know how, and any other intellectual property of Actions or others.

Miscellaneous

Information contained or described herein relates only to the Actions products and as of the release date of this publication, abrogates and supersedes all previously published data and specifications relating to such products provided by Actions or by any other person purporting to distribute such information.

Actions reserves the rights to make changes to information described herein at any time without notice. Please contact your Actions sales representatives to obtain the latest information before placing your product order.

Additional Support

Additional products and company information can be obtained by visiting the Actions website at: http://www.actions-semi.com

支持：

如欲获得公司及产品的其它信息，欢迎访问我公司网站：http://www.actions-semi.com

目 录

版本历史

日期	版本号	注释	作者
2015-4-30	1.0	建立初始版本	Liminxian

引 言

1.1 编写目的

对 PSP 接口进行说明，为 AP 编写和客户开发提供参考。

1.2 背景

该文档是 US280A 系统平台下底层软件开发的接口文档，适用于 US280A 对应的所有方案

1.3 参考资料

[1] US280A 系统接口设计说明书，Actions，2013.12.

1.4 术语和缩写词

缩写和术语	解 释
AP	Application, 应用程序

操作系统接口

1.5 概述

操作系统接口按照不同的功能模块分为：时间管理、设备驱动管理、虚拟文件系统管理、VM、中断管理、AP 管理、字符设备驱动、调频管理等。

1.6 函数接口定义

接口定义头文件路径：US282A\psp_rel\include\kernel_interface.h

1.6.1 时间管理接口函数

1.6.1.1 sys_set_irq_timer1

- 功能描述：用于注册毫秒级的定时器 timer1，单位为 1ms。
- 函数原型：int8 sys_set_irq_timer1(void* time_handle, uint32 ms_count)
- 输入参数描述：

  time_handle　　定时器回调函数

  ms_count　　定时周期(ms)
- 输出参数描述：

  0~5　　success 定时器索引号。最多 6 个定时器

  -1　　failed
- 说明：bank 代码，禁止在中断调用；与 sys_del_irq_timer1 配合使用

1.6.1.2 sys_del_irq_timer1

- 功能描述：删除 Timer1 定时器，单位为 10ms。
- 函数原型：int sys_del_irq_timer1(unsigned int timer)
- 输入参数描述：

  timer　　定时器索引号
- 输出参数描述：

  0　　success

  -1　　failed
- 说明：bank 代码，禁止在中断调用；与 sys_set_irq_timer1 配合使用

1.6.1.3 sys_udelay

- 功能描述：微秒级延时，CPU 空等，不释放。
- 函数原型：void sys_udelay(uint32 us)
- 输入参数描述：

  us　　延时时间
- 输出参数描述：无
- 说明：

1.6.1.4 sys_mdelay

- 功能描述：毫秒级延时，CPU 空等，不释放。
- 函数原型：void sys_mdelay(uint32 ms)

- 输入参数描述：

  ms　　延时时间
- 输出参数描述：无
- 说明：
-

1.6.1.5 sys_get_ab_timer

- 功能描述：获取绝对时间，单位 ms，毫秒精度，count 模块实现。
- 函数原型：uint32 sys_get_ab_timer(void)
- 输入参数描述：无
- 输出参数描述：返回绝对时间 ms
- 说明：

1.6.1.6 sys_set_time

- 功能描述：设置日历的时分秒。RTC 模块实现。
- 函数原型：void sys_set_time(time_t *time)
- 输入参数描述：

  time　　设置时间参数
- 输出参数描述：无
- 说明：

1.6.1.7 sys_get_time

- 功能描述：获取日历的时分秒。RTC 模块实现。
- 函数原型：void sys_get_time(time_t *time)
- 输入参数描述：无
- 输出参数描述：

  time　　获取的时间
- 说明：

1.6.1.8 sys_set_date

- 功能描述：设置日历的年月日。RTC 模块实现。
- 函数原型：void sys_set_date (date_t *date)
- 输入参数描述：

  date　　设置日期
- 输出参数描述：

  0　　success

  -1　　failed
- 说明：如果输入日期格式不对，接口会返回失败

1.6.1.9 sys_get_date

- 功能描述：获取日历的年月日。RTC 模块实现。
- 函数原型：void sys_get_date (date_t *date)
- 输入参数描述：无
- 输出参数描述：

  date　　获取的日期
- 说明：

1.6.1.10 sys_set_ alarm_time

- 功能描述：设置闹钟的时间。RTC 模块实现。
- 函数原型：void sys_set_ alarm_time (time_t *time)
- 输入参数描述：

  time　　闹钟时间
- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用

1.6.1.11 sys_get_ alarm_time

- 功能描述：获取闹钟的时间。RTC 模块实现。
- 函数原型：void sys_get_ alarm_time (time_t *time)
- 输入参数描述：无
- 输出参数描述：

  time　　获取的闹钟时间
- 说明：

1.6.1.12 sys_os_time_dly

- 功能描述：用于挂起线程睡眠，释放 CPU 控制权，延时一定时间。
- 函数原型：void sys_os_time_dly(uint16 ticks)
- 输入参数描述：

  ticks　　延时时间节拍数，单位 10ms
- 输出参数描述：无
- 说明：

1.6.1.13 sys_os_time_dly_resume

- 功能描述：恢复延时的任务。
- 函数原型：uint8 sys_os_time_dly_resume(uint8 prio)
- 输入参数描述：

  prio　　任务优先级
- 输出参数描述：

0 success
-1 failed

- 说明：

1.6.1.14 sys_sleep

- 功能描述：用于挂起线程睡眠，释放 CPU 控制权，秒级延时。
- 函数原型：void sys_sleep(uint32 s)
- 输入参数描述：
  s 延时时间，单位 s
- 输出参数描述：无
- 说明：

1.6.1.15 sys_usleep

- 功能描述：挂起线程睡眠，释放 CPU 控制权，微级延时。
- 函数原型：void sys_usleep(uint32 us)
- 输入参数描述：
  us 延时时间，单位 us
- 输出参数描述：无
- 说明：

1.6.1.16 sys_get_delay_val

- 功能描述：用于获取延时计数值。
- 函数原型：int sys_get_delay_val(uint32 delay_ms uint32 div_val)
- 输入参数描述：
  delay_ms: 延时时间，单位 ms
  div_val: 时钟分频比
- 输出参数描述：
  延时计数值
- 说明：bank 代码，禁止在中断调用

1.6.1.17 sys_us_timer_start

- 功能描述：用于初始微秒计时器。
- 函数原型：void sys_us_timer_start(void)
- 输入参数描述：无
- 输出参数描述：无
- 说明：

1.6.1.18 sys_us_timer_break

- 功能描述：用于获取微秒计时器值，微秒级误差
- 函数原型：uint32 sys_us_timer_break(void)
- 输入参数描述：无
- 输出参数描述：
  微秒时间值
- 说明：与 sys_us_timer_start 配合使用，计算过程中不能有调频动作，否则计时不准。

1.6.1.19 sys_reset_timer

- 功能描述：用于重置系统的定时计数器
- 函数原型：void sys_reset_timer(void)
- 输入参数描述：无
- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用

1.6.2 设备驱动管理接口函数

1.6.2.1 sys_drv_install

- 功能描述：装载驱动。
- 函数原型：int sys_drv_install (uint8 drv_type, void *drv_para, char* drv_name)
- 输入参数描述：

drv_type	驱动类型索引，详见 drv_type_t 说明
drv_para	传给驱动参数
drv_name	驱动文件名，如 card.drv

- 输出参数描述：

0	success
-1	failed

- 说明：bank 代码，禁止在中断调用；与 sys_drv_uninstall 一定要匹配使用。

1.6.2.2 sys_drv_uninstall

- 功能描述：卸载驱动。
- 函数原型：int sys_drv_uninstall(uint8 drv_type)
- 输入参数描述：

drv_type	驱动类型索引，详见 drv_type_t 说明

- 输出参数描述：

0	success
-1	failed

- 说明：bank 代码，禁止在中断调用；与 sys_drv_install 一定要匹配使用。

1.6.2.3 sys_get_drv_install_info

- 功能描述：获取驱动安装信息，低 8 个比特表示该种驱动被安装的次数，次低 8 个比特表示该种驱动的工作模式。务必保证多次安装时的工作模式是一样的。在双应用场景，两个应用安装的驱动都应该是 MODE_NORMAL。。
- 函数原型：int sys_get_drv_install_info(uint8 drv_type)
- 输入参数描述：

  drv_type　　驱动类型索引，详见 drv_type_t 说明
- 输出参数描述：驱动安装信息
- 说明：bank 代码，禁止在中断调用

1.6.2.4 sys_detect_disk

- 功能描述：检测存储介质是否存在。
- 函数原型：int sys_detect_disk(uint8 drv_type)
- 输入参数描述：

  drv_type　　驱动类型索引，详见 drv_type_t 说明
- 输出参数描述：

  0　　存在

  -1　　不存在
- 说明：bank 代码，禁止在中断调用

1.6.2.5 sys_ set_drv_setting

- 功能描述：设备 card 驱动配置，如 card 驱动能力等。
- 函数原型：int sys_set_drv_setting(card_pm_cfg_t *set_info)
- 输入参数描述：

  设置信息内容 card_pm_cfg_t
- 输出参数描述：

  0　　成功

  -1　　失败
- 说明：bank 代码，禁止在中断调用

1.6.2.6 sys_set_drv_ops

- 功能描述：用于重定位驱动的 ops 地址
- 函数原型：int　sys_set_drv_ops(uint8 drv_type, uint32 ops_addr)
- 输入参数描述：

  drv_type: 驱动类型索引，详见 drv_type_t 说明

  ops_addr：驱动地址表

- 输出参数描述：

  0　　成功

  -1　　失败

- 说明：bank 代码，禁止在中断调用

1.6.3　VFS 接口函数

1.6.3.1　sys_mount_fs

- 功能描述：挂载设备的文件系统
- 函数原型：int sys_mount_fs (uint8 drv_type, uint8 disk, uint8 partition_num)
- 输入参数描述：

  drv_type　　驱动类型索引，详见 drv_type_t 说明

  disk　　设备盘符，如 H,U 等

  partition_num　分区号，默认为 0

- 输出参数描述：

  0~2　　success，vfs_mount_t 数据结构的索引。以后对该用户驱动的文件系统进行操作都需要把这个索引作为第一个参数传给 vfs

  -1　　failed

- 说明：bank 代码，禁止在中断调用；装载 FS 前，一定要先装载存储驱动。

1.6.3.2　sys_unmount_fs

- 功能描述：卸载已挂载的文件系统。
- 函数原型：int sys_unmount_fs(int8 vfs_mount_index)
- 输入参数描述：

  vfs_mount_index　　vfs_mount_t 数据结构的索引。

- 输出参数描述：

  0　　success

  -1　　failed

- 说明：bank 代码，禁止在中断调用；

1.6.3.3　sys_get_fat_type_after_mount

- 功能描述：获取挂载后的文件系统的类型。
- 函数原型：uint8 sys_get_fat_type_after_mount(uint32 vfs_mount_index)
- 输入参数描述：

  vfs_mount_index　　vfs_mount_t 数据结构的索引。

- 输出参数描述：

  0　　FAT16/FAT32

  1　　exFAT

- 说明：bank 代码，禁止在中断调用；

1.6.3.4 sys_format_disk

- 功能描述：指定文件系统类型，格式化指定设备分区。
- 函数原型：int sys_format_disk(uint8 dry_type, uint8 partition_num, uint8 fat_type)
- 输入参数描述：

  dry_type　　存储驱动类型
  partition_num　　分区号，默认为 0
  fat_type　　FS 类型，FORMAT_FAT32，FORMAT_EXFAT，FORMAT_FAT16
- 输出参数描述：

  0　　success
  -1　　failed
- 说明：bank 代码，禁止在中断调用；

1.6.4 SDFS 接口函数

1.6.4.1 sys_sd_fopen

- 功能描述：打开一个 sd 区的文件。
- 函数原型：sd_file_t *sys_sd_fopen (char * filename)
- 输入参数描述：

  filename　　文件名 8+3，xxx.yyy 格式字符串，一定能够是 0 结尾
- 输出参数描述：

  非 0　　文件句柄
  0　　failed
- 说明：bank 代码，禁止在中断调用；最多支持 8 个 sdfs 句柄。

1.6.4.2 sys_sd_fclose

- 功能描述：关闭 SDFS 句柄。
- 函数原型：int sys_sd_fclose (sd_file_t *fp)
- 输入参数描述：

  fp　　文件句柄
- 输出参数描述：

  0　　success
  -1　　failed
- 说明：

1.6.4.3 sys_sd_fseek

- 功能描述：在已经打开的文件内 seek 到某个位置。
- 函数原型：int sys_sd_fseek (sd_file_t *fp, uint8 whence, int32 offset)
- 输入参数描述：

  fp 文件句柄

  whence seek 方向，

  SEEK_SET 0 只能正

  SEEK_CUR 1 正/负

  SEEK_END 2 只能正

  offset 偏移值

- 输出参数描述：

  0 success

  -1 failed

- 说明：

1.6.4.4 sys_sd_ftell

- 功能描述：获取文件当前读指针位置。
- 函数原型：int sys_sd_ftell (sd_file_t *fp)
- 输入参数描述：

  fp 文件句柄

- 输出参数描述：当前读指针的位置
- 说明：

1.6.4.5 sys_sd_fread

- 功能描述：读取已经打开的 SD 区文件的内容。
- 函数原型：int32 sys_sd_fread(sd_file_t *fp, void *buf, uint32 len)
- 输入参数描述：

  fp 文件句柄

  buf 读取文件内容到该 buffer 内

  len 要求读取的长度，单位 byte

- 输出参数描述：当前读指针的位置

  真实读取的长度，

  -1 failed

- 说明：

1.6.4.6 sys_base_set_info

- 功能描述：用于 sd 区配置信息

- 函数原型：sys_base_set_info(void *info, uint32 type)
- 输入参数描述：
  info：配置信息
  type：设置类型
- 输出参数描述：
  0 成功
  -1 失败
- 说明：bank 代码，禁止在中断调用

1.6.4.7 get_fw_info

- 功能描述：获取 SD 区指定位置数据
- 函数原型：int get_fw_info (uint8 *buf, uint32 info_addr, uint32 info_len)
- 输入参数描述：
  ptr_fw_info：存放信息内容
  info_addr：sd 位置
  info_len：数据长度(字节)
- 输出参数描述：
  0 success
  -1 failed
- 说明：

1.6.5 VM 接口函数

1.6.5.1 sys_vm_read

- 功能描述：读 VM 区数据，字节单位读
- 函数原型：int sys_vm_read(void *buf, uint32 address, uint32 len)
- 输入参数描述：
  buf 读的目标内存地址
  address 读的 VM 地址
  len 读的长度，单位 byte
- 输出参数描述：
  0 success
  -1 failed
- 说明：

1.6.5.2 sys_vm_write

- 功能描述：写入 VRAM 的数据，字节单位写入

- 函数原型：int sys_vm_write(void *buf, uint32 address, uint32 len)
- 输入参数描述：

  buf 数据内存地

  address 写入的 VM 地址

  len 写入数据长度，字节单位
- 输出参数描述：

  0 success

  -1 failed
- 说明：

1.6.6 中断管理接口函数

1.6.6.1 sys_request_irq

- 功能描述：注册中断。
- 函数原型：int sys_request_irq(uint32 irq_type, void *handle)
- 输入参数描述：

  irq_type 中断索引号，详见 irq_type_t

  handle 中断回调函数
- 输出参数描述：

  0 success

  -1 failed
- 说明：bank 代码，禁止在中断调用；与 sys_free_irq 配对使用

1.6.6.2 sys_free_irq

- 功能描述：注销中断。
- 函数原型：void sys_free_irq(uint32 irq_type)
- 输入参数描述：

  irq_type 中断索引号，详见 irq_type_t
- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用；与 sys_request_irq 配对使用

1.6.6.3 sys_local_irq_save

- 功能描述：关中断。
- 函数原型：uint32 sys_local_irq_save(void)
- 输入参数描述：无
- 输出参数描述：返回当前中断配置，必须保存。
- 说明：与 sys_local_irq_restore 配对使用

1.6.6.4 sys_local_irq_restore

- 功能描述：恢复中断。
- 函数原型：void sys_local_irq_restore(uint32 irq_save)
- 输入参数描述：

  irq_save 之前函数 sys_local_irq_save 保存的状态值
- 输出参数描述：无
- 说明：与 sys_local_irq_save 配对使用

1.6.6.5 sys_set_mpu_irq

- 功能描述：注册 mpu 模块中断方式。
- 函数原型：int32 sys_set_mpu_irq(sys_mpu_param_t *param)
- 输入参数描述：

  param：注册参数类型，见定义 sys_mpu_param_t
- 输出参数描述：中断注册 id 号，-1 表示失败
- 说明： bank 代码，禁止在中断调用；与 sys_del_mpu_irq 配对使用

1.6.6.6 sys_del_mpu_irq

- 功能描述：注销 mpu 中断。
- 函数原型：void sys_del_mpu_irq(id)
- 输入参数描述：

  id 中断注册 id 号
- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用；与 sys_set_mpu_irq 配合使用

1.6.6.7 sys_request_dsp_irq

- 功能描述：请求 DSP 协议中断
- 函数原型：int32 sys_request_dsp_irq(uint8 in_use, uint8 *cmd_context)
- 输入参数描述：

  in_use 中断类型，0~3

  cmd_context 中断协议信息，默认 0

- 输出参数描述：

  0 success

  -1 failed

- 说明：bank 代码，禁止在中断调用；

1.6.6.8 sys_respond_dsp_cmd

- 功能描述：回应 DSP 服务请求，表示服务已完成，DSP 可以开始工作

- 函数原型：int32 sys_respond_dsp_cmd(void)
- 输入参数描述：
- 输出参数描述：

0	success
-1	failed

- 说明：

1.6.7 AP 管理接口函数

1.6.7.1 sys_exece_ap

- 功能描述：装载 ap。
- 函数原型：int sys_exece_ap(char *name,uint32 type,void *argc)
- 输入参数描述：

name	应用的文件名 XXX.AP
type	应用类型，是否为引擎进程
arg	传给 ap main 的参数

- 输出参数描述：0-表示成功，-1 表示失败
- 说明：bank 代码，禁止在中断调用；sys_exece_ap 与 sys_free_ap 配对使用

1.6.7.2 sys_free_ap

- 功能描述：卸载 ap。
- 函数原型：void sys_free_ap(uint32 type)
- 输入参数描述：

type	应用类型，是否为引擎进程

- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用；sys_exece_ap 与 sys_free_ap 配对使用

1.6.7.3 sys_load_codec

- 功能描述：装载 codec。
- 函数原型：int sys_load_codec(char *name, uint32 type)
- 输入参数描述：

name	codec 名字
type	是否为引擎进程加载的 codec 驱动

- 输出参数描述：

0	success
-1	failed

- 说明：bank 代码，禁止在中断调用；sys_load_codec 与 sys_free_codec 配对使用

1.6.7.4 sys_free_codec

- 功能描述：卸载 codec。
- 函数原型：void sys_free_codec(uint32 type)
- 输入参数描述：

  type 是否为引擎进程加载的 codec 驱动
- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用；sys_load_codec 与 sys_free_codec 配对使用

1.6.7.5 sys_load_mmm

- 功能描述：装载中间件。
- 函数原型：int sys_load_mmm(char *name, uint32 type)
- 输入参数描述：

  name 中间件名字

  type 是否为引擎进程加载的 mmm 驱动
- 输出参数描述：

  0 success

  -1 failed
- 说明：bank 代码，禁止在中断调用；sys_load_mmm 与 sys_free_mmm 配对使用

1.6.7.6 sys_free_mmm

- 功能描述：装载中间件。
- 函数原型：void sys_free_mmm(uint32 type)
- 输入参数描述：

  type 是否为引擎进程加载的 mmm 驱动
- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用；sys_load_mmm 与 sys_free_mmm 配对使用

1.6.7.7 sys_load_dsp_codec

- 功能描述：加载 DSP 库。
- 函数原型：int sys_load_dsp_codec(char *name, uint32 type)
- 输入参数描述：

  name 库名字

  type 库类型
- 输出参数描述：

  0 success

  -1 failed
- 说明：bank 代码，禁止在中断调用；sys_load_dsp_codec 与 sys_free_dsp_codec 配对使

用

1.6.7.8 sys_free_dsp_codec

- 功能描述：卸载 DSP 库。
- 函数原型：void sys_free_dsp_codec(uint32 type)
- 输入参数描述：

  type 库类型
- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用；sys_load_dsp_codec 与 sys_free_dsp_codec 配对使用

1.6.8 调频管理接口函数

1.6.8.1 sys_adjust_clk

- 功能描述：用于调整系统 MISP 和 DSP 频率。
- 函数原型：uint32 sys_adjust_clk(uint32 freq, uint32 type)
- 输入参数描述：

  freq 频率值，低 8 位为 MISP 频率，高 8 位为 DSP 频率

  type 调频类型，无用，默认为 0
- 输出参数描述：返回调频前的频率值，低 8 位为 MISP 频率，高 8 位为 DSP 频率。
- 说明：bank 代码，禁止在中断调用。

1.6.8.2 sys_adjust_asrc_clk

- 功能描述：用于调整系统 ASRC 频率。
- 函数原型：uint32 sys_adjust_asrc_clk(uint32 freq)
- 输入参数描述：

  freq 频率值
- 输出参数描述：返回调频前的 ASRC 频率值。
- 说明：bank 代码，禁止在中断调用。

1.6.8.3 sys_request_clkadjust

- 功能描述：注册调频回调函数。
- 函数原型：int8 sys_request_clkadjust(void *call_back, uint32 pll_range)
- 输入参数描述：

  call_back：回调函数入口

  pll_range：限制 PLL 频率范围，低 16 位为最小值，高 16 位为最大值；若值为 0

表示不限制 PLL 频率

- 输出参数描述：

  0~5　　调频数组的管理索引号

  -1　　failed

- 说明：bank 代码，禁止在中断调用；

1.6.8.4 sys_free_clkadjust

- 功能描述：注销调频回调函数。
- 函数原型：int sys_free_clkadjust(int8 id)
- 输入参数描述：

  id　　调频数组的管理索引号。

- 输出参数描述：

  0　　success

  -1　　failed

- 说明：bank 代码，禁止在中断调用；

1.6.8.5 sys_lock_adjust_freq

- 功能描述：锁定系统当前的 MISP&DSP 频率，不接受其他频率的调节
- 函数原型：void sys_lock_adjust_freq(void)
- 输入参数描述：无
- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用；

1.6.8.6 sys_unlock_adjust_freq

- 功能描述：解锁系统的 MISP&DSP 频率调节
- 函数原型：void sys_unlock_adjust_freq(void)
- 输入参数描述：无
- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用；

1.6.9 内存管理接口函数

1.6.9.1 sys_malloc

- 功能描述：申请内存空间。

- 函数原型：void* sys_malloc(uint32 size)
- 输入参数描述：内存地址，为 0 表示失败
  size　　申请的字节大小
- 输出参数描述：
  0　　success
  -1　　failed
- 说明：sys_malloc 与 sys_free 配对使用

1.6.9.2 sys_free

- 功能描述：释放申请内存。
- 函数原型：int sys_free(void* addr)
- 输入参数描述：
  addr　　内存的指针变量的地址。
- 输出参数描述：
  0　　success
  -1　　failed
- 说明：sys_malloc 与 sys_free 配对使用。

1.6.10 消息处理接口函数

1.6.10.1 sys_mq_send

- 功能描述：发送消息。
- 函数原型：int mq_send(uint8 queue_id, void *msg)
- 输入参数描述：
  queue_id　　消息队列索引号。
  msg　　消息内容指针
- 输出参数描述：
  0　　success
  -1　　failed
- 说明：

1.6.10.2 sys_mq_receive

- 功能描述：接收消息。
- 函数原型：int sys_mq_receive (uint8 queue_id, void *msg)
- 输入参数描述：
  queue_id　　消息队列索引号。

msg 消息内容指针

- 输出参数描述：

0 success

-1 failed

- 说明：

1.6.10.3 sys_mq_flush

- 功能描述：清空消息队列。
- 函数原型：int sys_mq_flush (uint8 queue_id)
- 输入参数描述：

queue_id 消息队列索引号。

- 输出参数描述：

0 success

-1 failed

- 说明：

1.6.10.4 sys_mq_traverse

- 功能描述：查询/获取 消息，获取后的消息不会被删除
- 函数原型：int sys_mq_traverse(uint8 queue_id, void* msg, uint32 msg_index)
- 输入参数描述：

queue_id：消息类型 id

msg：存放消息内容，若为 NULL 则只查询消息总数，不获取消息内容

msg_index：获取指定的消息序号

- 输出参数描述：消息池中消息的总数，为-1 表示操作失败
- 说明：

1.6.11 共享查询机制接口函数

1.6.11.1 sys_share_query_creat

- 功能描述：创建共享内存查询管理队列，并返回可被写入的内存地址。
- 函数原型：void* sys_share_query_creat(int8 query_id, uint8 *mem_addr, uint16 size)
- 输入参数描述：

qquery_id：队列 ID

mem_addr：内存地址

size：内存大小

- 输出参数描述：可写入的内存地址，NULL -为创建失败
- 说明：bank 代码，禁止在中断调用

1.6.11.2 sys_share_query_destroy

- 功能描述：删除已创建的队列。
- 函数原型：int sys_share_query_destroy(int8 query_id)
- 输入参数描述：

  query_id：队列 ID
- 输出参数描述：

  0 success

  -1 failed
- 说明：bank 代码，禁止在中断调用；

1.6.11.3 sys_share_query_read

- 功能描述：获取共享内存消息内容
- 函数原型：int sys_share_query_read(int8 query_id, uint8 *read_addr)
- 输入参数描述：

  query_id：队列 ID

  read_addr：存入消息内容
- 输出参数描述：

  0 success

  -1 failed
- 说明：

1.6.11.4 sys_share_query_update

- 功能描述：更新已写入共享内存消息内容，并返回可被写入的内存地址
- 函数原型：void* sys_share_query_update(int8 query_id)
- 输入参数描述：

  query_id：队列 ID
- 输出参数描述：可被写入的内存地址，NULL-为更新失败
- 说明：

1.6.12 共享内存机制接口函数

1.6.12.1 sys_share_query_creat

- 功能描述：创建共享内存空间
- 函数原型：int sys_shm_creat(int8 shm_id, uint8 *shm_addr, uint16 shm_size)
- 输入参数描述：

  shm_id：共享内存标识 ID

  shm_addr：内存地址

  shm_size：内存大小

- 输出参数描述：
  0 success
  -1 failed
- 说明：bank 代码，禁止在中断调用

1.6.12.2 sys_shm_destroy

- 功能描述：删除共享内存空间
- 函数原型：int sys_shm_destroy(int8 shm_id)
- 输入参数描述：
  shm_id：共享内存标识 ID
- 输出参数描述：
  0 success
  -1 failed
- 说明：bank 代码，禁止在中断调用

1.6.12.3 sys_shm_mount

- 功能描述：请求共享内存地址
- 函数原型：uint8* sys_shm_mount(int8 shm_id)
- 输入参数描述：bank 代码，禁止在中断调用
  shm_id：共享内存标识
- 输出参数描述：共享内存地址，NULL-为请求失败
- 说明：bank 代码，禁止在中断调用

1.6.13 工作&信息配置接口

1.6.13.1 sys_enter_high_powered

- 功能描述：启动系统电气特性 性能最优模式
- 函数原型：void sys_enter_high_powered(int up_type)
- 输入参数描述：
  up_type ：提升类型，1--VDD, 2--VCC, 3--(VCC + VDD)
- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用

1.6.13.2 sys_exit_high_powered

- 功能描述：退出系统电气特性 性能最优模式
- 函数原型：void sys_exit_high_powered(void)
- 输入参数描述：无
- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用

1.6.13.3 sys_set_hosc_param

- 功能描述：设置高频电容参数值
- 函数原型：void sys_set_hosc_param(uint16 param)
- 输入参数描述：
  param 电容参数值
- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用

1.6.13.4 sys_set_sys_info

- 功能描述：设置系统信息
- 函数原型：int sys_set_sys_info(void *info, uint32 sys_info_type)
- 输入参数描述：
  sys_info_type：信息类型，见 sys_info_type_e 定义
- 输出参数描述：0 表示成功，-1 表示失败
- 说明：bank 代码，禁止在中断调用

1.6.13.5 sys_get_sys_info

- 功能描述：获取系统信息
- 函数原型：int sys_get_sys_info(void *info, uint32 sys_info_type)
- 输入参数描述：
  sys_info_type：信息类型，见 sys_info_type_e 定义
- 输出参数描述：0 表示成功，-1 表示失败
- 说明：bank 代码，禁止在中断调用

1.6.13.6 sys_random

- 功能描述：获取系统的随机数据
- 函数原型：uint32 sys_random(void)
- 输入参数描述：
- 输出参数描述：
  随机数 范围 0~0xFFFFFFFF
- 说明：

1.6.13.7 sys_read_c0count

- 功能描述：获取系统的 cpu 计数器值
- 函数原型：uint32 sys_read_c0count(void)
- 输入参数描述：
- 输出参数描述：
  计数器值 范围 0~0xFFFFFFFF
- 说明：

1.6.14 系统调试接口

1.6.14.1 sys_cpu_monitor_start

- 功能描述：启动 cpu 使用占空比监控
- 函数原型：void sys_cpu_monitor_start(void)
- 输入参数描述：无
- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用

1.6.14.2 sys_cpu_monitor_end

- 功能描述：结束 cpu 使用占空比监控，查看结果
- 函数原型：void sys_cpu_monitor_end(void)
- 输入参数描述：无
- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用

1.6.14.3 sys_dsp_print

- 功能描述：打印 dsp 缓存数据
- 函数原型：void sys_dsp_print(void)
- 输入参数描述：无
- 输出参数描述：无
- 说明：bank 代码，禁止在中断调用

1.6.14.4 sys_irq_print

- 功能描述：用于系统中断打印
- 函数原型：void sys_irq_print(char *str, uint32 data,uint8 mode)
- 输入参数描述：
  str:字符串
  data:数据

mode:打印模式，0-只打印字符串，1-只打印数据，2-打印字符串+数据

- 输出参数描述：无
- 说明：只允许在中断服务函数中使用

1.7 关键数据结构定义

1.7.1 数据结构 drv_type_t

- 功能描述：驱动类型
- 数据结构原型：

```
typedef enum {
    DRV_GROUP_STG_BASE = 0,
    DRV_GROUP_STG_CARD,
    DRV_GROUP_STG_UHOST,
    DRV_GROUP_FAT,
    DRV_GROUP_EXFAT,
    DRV_GROUP_UD,
    DRV_GROUP_LCD,
    DRV_GROUP_FM,
    DRV_GROUP_KEY,
    DRV_GROUP_CCD,
    DRV_GROUP_AUDIO_DEVICE,
    DRV_GROUP_TTS,
    DRV_GROUP_BT,
    DRV_RESERVER2,
    DRV_GROUP_SYS = 15,
} drv_type_t;
```

- 必要参数重点说明：

1.7.2 数据结构 vfs_mount_t

- 功能描述：VFS 类型
- 数据结构原型：

```
typedef struct {
    unsigned char drv_type;
    unsigned char fs_type;
```

```
    unsigned char disk;
    unsigned int start_offset;
    void *fat_private_data;
} vfs_mount_t;
```

- 必要参数重点说明：

1.7.3 数据结构 irq_type_t

- 功能描述：irq 类型
- 数据结构原型：

```
enum {
    IRQ_BT = 0,
    IRQ_NFC,
    IRQ_2HZ,/*WD & 2HZ*/
    IRQ_TIMER1,
    IRQ_TIMER0,
    IRQ_RTC,
    IRQ_UART0,
    IRQ_SIRQ0,
    IRQ_TOUCHKEY,
    IRQ_SPI,
    IRQ_USB,
    IRQ_I2C,
    IRQ_UART1,
    IRQ_SIRQ1,
    IRQ_DAC_I2S_TX,
    IRQ_ADC_I2S_RX,
    IRQ_MPU,
    IRQ_SD_MMC,
    IRQ_DMA0,
    IRQ_DMA1,
    IRQ_DMA2,
    IRQ_DMA3,
    IRQ_DMA4,
    IRQ_DMA5,
    IRQ_PCM_RX,
    IRQ_PCM_TX,
    IRQ_SPI1,
```

```
    IRQ_OUT_USER0,
    IRQ_OUT_USER1,
    IRQ_OUT_USER2,
    IRQ_OUT_USER3,
    IRQ_OUT_USER4,
};
```

- 必要参数重点说明：

1.7.4 数据结构 sys_mpu_param_t

- 功能描述：mpu 调试接口
- 数据结构原型：

```
typedef struct
{
    uint8    mpu_index; //mpu 单元
    uint8    mpu_err_ip;//配置触发错误中断位
    uint16 resever;     //保留未用
    uint32 start_addr;//起始内存地址
    uint32 end_addr;   //结束内存地址
    void *irq_handle; //中断服务程序,为 0 默认进入系统死循环
}sys_mpu_param_t;
```

- 必要参数重点说明：

1.7.5 数据结构 MQ_ID_E

- 功能描述：消息类型
- 数据结构原型：

```
typedef enum
{
    MQ_ID_MNG = 0, /*进程管理应用消息队列*/
    MQ_ID_DESK,      /*UI（前台）应用消息队列*/
    MQ_ID_EGN,       /*引擎（后台）应用消息队列*/
    MQ_ID_BT,        /*蓝牙后台消息队列*/
    MQ_ID_RES,       /*reserve*/
    MQ_ID_SYS,       /*系统消息队列*/
```

```
    MQ_ID_GUI,        /*GUI 消息队列*/
    MQ_ID_MAX
} MQ_ID_E;
```

- 必要参数重点说明：

1.7.6 数据结构

Libc 接口

1.8 概述

libc 接口按照不同的功能模块分为：线程管理、进城管理、信号量等。

1.9 函数接口定义

接口定义头文件路径：US282A\psp_rel\include\libc_interface.h

1.9.1 线程管理接口函数

1.9.1.1 libc_pthread_create

- 功能描述：创建线程。
- 函数原型：int libc_pthread_create(pthread_param_t *pthread_pram, INT8U prio, uint8 process_descr_index)
- 输入参数描述：

pthread_pram	线程创建数据结构，详见 pthread_param_t
prio	线程优先级
process_descr_index	CREATE_NOT_MAIN_THREAD，表示创建非 main 主线程。

- 输出参数描述：

0~5	success 定时器索引号。最多 6 个定时器
-1	failed

- 说明：因为 Main 主线程由运行时库创建，所以调用此接口一般为非主线程。

1.9.1.2 libc_pthread_exit

- 功能描述：线程退出。
- 函数原型：void libc_pthread_exit(void)
- 输入参数描述：无
- 输出参数描述：无
- 说明：Main 中如果没有调用 libc_pthread exit 接口退出将会退出所属进程；如果最后一个线程调用该接口后也会退出所属进程。

1.9.2 进程管理

1.9.2.1 libc_exit

- 功能描述：进程退出。
- 函数原型：void libc_exit(int8 exitval)
- 输入参数描述：

  exitval　　退出进程时传递的参数，将会由 waitpid 接口获得该参数的值
- 输出参数描述：无
- 说明：Main 中如果没有调用 pthread_exit 接口退出，将会调用该接口。

1.9.2.2 libc_waitpid

- 功能描述：等待其他进程退出。
- 函数原型：int libc_waitpid(int8 *stat_loc, int options)
- 输入参数描述：

  stat_loc　　获得 exit 中传递的参数

  options　　指示是否阻塞等待，WNOHANG 表示不阻塞返回，其他会阻塞
- 输出参数描述：

  0~3　　表示退出的进程索引号

  WAITPID_ONLY_PROCESS_MANAGER　　表示没有其他进程存在

  WAITPID_NO_PROCESS_EXIT　　表示没有其他进程退出
- 说明：

1.9.2.3 libc_get_process_struct

- 功能描述：申请进程。
- 函数原型：int libc_get_process_struct(void)
- 输入参数描述：无
- 输出参数描述：

0~3	进程索引号
-1	fail

- 说明：

1.9.2.4 libc_free_process_struct

- 功能描述：释放进程。
- 函数原型：int libc_free_process_struct(int8 index)
- 输入参数描述：

index	进程索引号

- 输出参数描述：无
- 说明：

1.9.3 信号量

1.9.3.1 libc_sem_init

- 功能描述：信号量初始化。
- 函数原型：int libc_sem_init(os_event_t * sem, unsigned int value)
- 输入参数描述：

sem	信号量，详见 os_event_t
value	资源值

- 输出参数描述：

0	success
-1	failed

- 说明：

1.9.3.2 libc_sem_wait

- 功能描述：阻塞等待资源。
- 函数原型：int libc_sem_wait(os_event_t *sem, unsigned short timeout)
- 输入参数描述：

sem	信号量
timeout	等待时间，10 毫秒为单位，0 表示无限等待

- 输出参数描述：

0	success
-1	failed

- 说明：

1.9.3.3 libc_sem_trywait

- 功能描述：尝试获得资源。
- 函数原型：int libc_sem_trywait(os_event_t *sem)
- 输入参数描述：

  sem 信号量
- 输出参数描述：

  0 success

  -1 failed
- 说明：

1.9.3.4 libc_sem_post

- 功能描述：释放资源。
- 函数原型：int libc_sem_post(os_event_t *sem)
- 输入参数描述：

  sem 信号量
- 输出参数描述：

  0 success

  -1 failed
- 说明：

1.9.3.5 libc_sem_destroy

- 功能描述：销毁资源。
- 函数原型：int libc_sem_destroy (os_event_t *sem)
- 输入参数描述：

  sem 信号量
- 输出参数描述：

  0 success

  -1 failed
- 说明：

1.9.4 互斥量

1.9.4.1 libc_pthread_mutex_init

- 功能描述：互斥量初始化。
- 函数原型：int libc_pthread_mutex_init(os_event_t * mutex)
- 输入参数描述：

  mutex 互斥量

- 输出参数描述：

  0 success

  -1 failed

- 说明：

1.9.4.2 libc_pthread_mutex_lock

- 功能描述：互斥量加锁。
- 函数原型：int libc_pthread_mutex_lock(os_event_t * mutex)
- 输入参数描述：

  mutex 互斥量

- 输出参数描述：

  0 success

  -1 failed

- 说明：

1.9.4.3 libc_pthread_mutex_unlock

- 功能描述：互斥量解锁。
- 函数原型：int libc_pthread_mutex_unlock(os_event_t * mutex)
- 输入参数描述：

  mutex 互斥量

- 输出参数描述：

  0 success

  -1 failed

- 说明：

1.9.4.4 libc_pthread_mutex_trylock

- 功能描述：不阻塞地尝试互斥量加锁。
- 函数原型：int libc_pthread_mutex_trylock(os_event_t * mutex)
- 输入参数描述：

  mutex 互斥量

- 输出参数描述：

  0 success

  -1 failed

- 说明：

1.9.4.5 libc_pthread_mutex_destroy

- 功能描述：销毁互斥量。
- 函数原型：int libc_pthread_mutex_destroy (os_event_t * mutex)
- 输入参数描述：

mutex 互斥量

- 输出参数描述：

  0 success

  -1 failed

- 说明：

1.9.5 条件变量

1.9.5.1 libc_pthread_cond_init

- 功能描述：条件变量初始化。
- 函数原型：int libc_pthread_cond_init(os_event_t * cond , uint32 init_value)
- 输入参数描述：

  cond 条件量

  init_value 变量初始化值

- 输出参数描述：

  0 success

  -1 failed

- 说明：

1.9.5.2 libc_pthread_cond_wait

- 功能描述：等待条件变量，可以定时，也可以无限时等待。
- 函数原型：int libc_pthread_cond_wait(os_event_t *cond, os_event_t *mutex, unsigned short timeout)
- 输入参数描述：

  cond 条件量

  mutex 互斥量

  timeout 等待时间，10 毫秒为单位，0 表示无限等待

- 输出参数描述：

  0 success

  -1 failed

- 说明：

1.9.5.3 libc_pthread_cond_signal

- 功能描述：等待条件变量，可以定时，也可以无限时等待。
- 函数原型：int libc_pthread_cond_signal (os_event_t *cond)
- 输入参数描述：

  cond 条件量

- 输出参数描述：

  0 success

  -1 failed

- 说明：

1.9.5.4 libc_pthread_cond_destroy

- 功能描述：注销条件变量。
- 函数原型：int libc_pthread_cond_destroy (os_event_t *cond)
- 输入参数描述：

  cond 条件量

- 输出参数描述：

  0 success

  -1 failed

- 说明：

1.9.6 字符串操作

1.9.6.1 libc_memcpy

- 功能描述：内存拷贝。
- 函数原型：void libc_memcpy(void *dst, void *src, uint32 byte_size)
- 输入参数描述：

  dst 目标地址

  src 源地址

  byte_size 字节长度

- 输出参数描述：无
- 说明：

1.9.6.2 libc_memset

- 功能描述：内存赋值。
- 函数原型：void libc_memset(void *dst, int8 value, uint32 byte_size)
- 输入参数描述：

  dst 目标地址

  value 初始值

  byte_size 字节长度

- 输出参数描述：无
- 说明：

1.9.6.3 libc_memcmp

- 功能描述：内存比较。
- 函数原型：int libc_memcmp(const void * cs,const void * ct,unsigned int count)
- 输入参数描述：

cs	目标地址
ct	源地址
byte_size	字节长度

- 输出参数描述：

0	相等
<0	cs<ct
>0	cs>ct

- 说明：

1.9.6.4 libc_strlen

- 功能描述：获取字符串长度，不包括末尾的结束符。
- 函数原型：int libc_strlen (const char * s)
- 输入参数描述：

s	字符串起始地址

- 输出参数描述：返回字符串长度
- 说明：

1.9.6.5 libc_strncat

- 功能描述：s2 指向的字符串中的字符复制到 s1 所指向的字符串后面，最多可以复制 n 个字符。返回值是和 s1 相同的值，即结果字符串。
- 函数原型：char *libc_strncat (char *s1, const char *s2, unsigned int n)
- 输入参数描述：

s1	目标字符串指针
s2	源字符串指针
n	字节长度

- 输出参数描述：返回目标字符串指针
- 说明：

1.9.6.6 libc_strncmp

- 功能描述：比较 s1 和 s2 所指向字符串的前 n 个字符，结束符后面的字符会被忽略。
- 函数原型：int libc_strncmp(const char * cs, const char * ct, unsigned int count)
- 输入参数描述：

cs	目标字符串指针
ct	源字符串指针

count 比较字节长度

- 输出参数描述：

  0 相等

  <0 cs<ct

  >0 cs>ct

- 说明：

1.9.6.7 libc_strncpy

- 功能描述：从 s2 中复制最多 n 个字符到 s1 字符串中。返回值和第一个自变量的值相同。其中 s1 和 s2 所指向的内存区域不能有重叠的地方。
- 函数原型：char *libc_strncpy (char *s1, const char *s2, unsigned int n)
- 输入参数描述：

  s1 目标字符串指针

  s2 源字符串指针

  n 字节长度

- 输出参数描述：返回目标字符串指针
- 说明：

1.9.6.8 libc_strlenuni

- 功能描述：获取 unicode 字符串的长度。
- 函数原型：int libc_strlenuni (int8 *s1)
- 输入参数描述：

  s1 字符串指针

- 输出参数描述：返回目标字符串长度
- 说明：

1.9.6.9 libc_strncpyuni

- 功能描述：从 s2 中复制最多 n 个 unicode 字符到 s1 字符串中。返回值和第一个自变量的值相同。其中 s1 和 s2 所指向的内存区域不能有重叠的地方。
- 函数原型：char *libc_strncpyuni (int8 *s1, int8 *s2, unsigned int n)
- 输入参数描述：

  s1 目标字符串指针

  s2 源字符串指针

  n 字节长度

- 输出参数描述：返回目标字符串指针
- 说明：

1.9.6.10 libc_itoa

- 功能描述：数值转字符。
- 函数原型：void libc_itoa(uint32 num, uint8 *str, uint8 counts)
- 输入参数描述：

num	数值
str	存放结果内存
counts	转换的字节长度

- 输出参数描述：返回目标字符串指针
- 说明：

1.9.7 Uart print

1.9.7.1 libc_print

- 功能描述：打印字符串，数值。
- 函数原型：void libc_print(unsigned char* s, unsigned int Data, unsigned char mode)
- 输入参数描述：

s	字符串地址
Data	数值
mode	打印模式， 0，为仅打印字符串； 1，仅打印数值； 2，字符串和数值都打印

- 输出参数描述：无
- 说明：最多打印 16 个字符。

1.10 关键数据结构定义

1.10.1 数据结构 pthread_param_t

- 功能描述：线程描述
- 数据结构原型：

typedef struct

```
{
    void *(*start_rtn)(void*);
    void *arg;
    OS_STK *ptos;
} pthread_param_t;
```

- 必要参数重点说明：

1.10.2 数据结构 os_event_t

- 功能描述：信号量描述
- 数据结构原型：

```
typedef struct {
    unsigned char os_event_type;
    unsigned char os_event_grp;
    unsigned short os_event_cnt;
    void *os_event_ptr;
    unsigned char os_event_tbl[OS_EVENT_TBL_SIZE];
}os_event_t;
```

- 必要参数重点说明：

文件系统接口

1.11 概述

文件系统为上层应用访问 FLASH，CARD 等存储设备提供接口，应用能在 CARD 或者 FLASH 上创建文件、读写文件并且可以在 CARD 和 FLASH 之间执行拷贝数据。

1.12 接口列表

表格 3-2-1： 文件系统接口一览表

函数类别	函数列表	功能
目录操作接口	vfs_cd	切换目录
	vfs_dir	查找目录或文件
	vfs_make_dir	创建目录
文件操作接口	vfs_file_open	打开文件
	vfs_file_create	创建文件
	vfs_file_close	关闭文件
	vfs_file_get_size	获取文件大小
	vfs_get_time	获取文件时间信息
	vfs_file_seek	文件定位
	vfs_file_tell	获取文件位置
	vfs_file_read	读文件
	vfs_file_write	写文件
	vfs_set_time	设置文件时间信息
	vfs_file_attralter	设置文件目录属性
	vfs_file_dir_offset	记录或设置文件位置信息
	vfs_file_dir_remove	删除文件或目录（目录必须为空）
公共操作接口	vfs_get_err_info	获取错误信息
	vfs_file_dir_exist	判断文件目录是否存在
	vfs_file_rename	重命名文件目录
	vfs_get_name	获取文件/目录名
	vfs_get_space	获取磁盘空间
	vfs_create_volume	创建磁盘卷标

1.13 函数接口定义

需要说明的是:fat,fat32 文件/目录名少于 11 个字符的时候为 8.3 格式即 11 个 byte,大写,不足补 20h，超过 11 个字符，则需要转换为 unicode 编码，以 0xFFFE 开头，即长名，exfat 均为 unicode 编码，没有长名短名之分。

1.13.1 目录操作接口

1.13.1.1 vfs_cd

- 功能描述:根据用户输入将当前目录指向当前目录的子目录,父目录或直接返回根目录。
- 函数原型：bool vfs_cd(vfs_mount_t *p_vfs_mount, uint8 mode, uint8* ptr_input_name)
- 输入参数描述：

p_vfs_mount		mount 返回的文件系统索引
mode		
	CD_UP	改变当前目录到上一级父目录，目录项指针指向目录起始位置；
	CD_BACK	改变当前目录到上一级父目录,目录项指针指向之前 CD 进去的子目录；
	CD_ROOT	改变当前目录到根目录；
	CD_SUB	改变当前目录到当前目录项对应的子目录，此时参数 c 起作用。
ptr_input_name		子目录名的 buffer 指针；为 NULL 的时候默认为进入当前目录。前两个字符为"0xfffe”则表示通过长名切换目录

- 输出参数描述：

1	success
0	failed

- 说明：

1.13.1.2 vfs_dir

- 功能描述：在当前目录按各种方式检索目录或文件
- 函数原型：uint32 vfs_dir(vfs_mount_t *p_vfs_mount, uint8 mode, uint8* ptr_input_name, uint32 ext_name_bitmap)
- 输入参数描述：

p_vfs_mount		mount 返回的文件系统索引
mode		各种 dir 的模式。
	DIR_NEXT	向后 dir；
	DIR_PREV	向前 dir；

DIR_HEAD	从目录首向后 dir；
DIR_TAIL	从目录尾向前 dir；
ptr_input_name	用来存放输入文件名的 buffer 指针。如指针为 NULL，则直接从当前目录项开始 dir 或不起作用。
ext_name_bitmap	如是有效内存地址，则表示输入扩展名字符串的地址；如不是则表示要 dir 的 bitmap。此参数如最高位为 1 表示传入的内存地址，否则则是 bitmap 值。

- 输出参数描述：

  >0 success – 实际的扩展名

  0 failed

- 说明

1.13.1.3 vfs_make_dir

- 功能描述：在当前目录下生成一个用户程序指定目录名的子目录。
- 函数原型：bool vfs_make_dir(vfs_mount_t *p_vfs_mount, uint8* ptr_input_name)
- 输入参数描述：

  p_vfs_mount mount 返回的文件系统索引

  ptr_input_name 要创建的目录名指针

- 输出参数描述：

  1 success

  0 failed

- 说明：

1.13.2 文件操作接口

1.13.2.1 vfs_file_open

- 功能描述：根据用户输入的文件名在当前目录中打开一个已存在的文件。
- 函数原型：uint32 vfs_file_open(vfs_mount_t *p_vfs_mount, uint8* ptr_input_name,uint8 mode)
- 输入参数描述：

vfs_mount	mount 返回的文件系统索引
ptr_input_name	待打开文件的文件名的输入指针；为 null 表示直接打开当前目录项指向的文件
mode	打开方式。
R_NORMAL_SEEK	普通 seek 模式
R_FAST_SEEK	快速 seek 模式(R_FAST_SEEK 每次打开后需要先 seek 到文件末尾。)
OPEN_MODIFY	修改模式（不允许跨越文件大小写）

LARGE_FILE_SEEK	大文件 seek 模式（大文件打开可能较慢）
OPEN_RECOVER	恢复模式（允许恢复写后没有正常 close 的文件，操作方式见说明）

- 输出参数描述：

  >0 success – 文件句柄

  0 failed

- 说明：

在当前系统下只支持同时打开 4 个句柄，最多支持两个快速 seek，只支持 1 个写或修改，写和修改都以 OPEN_MODIFY 方式打开。

恢复未正常关闭文件操作：

1、 写文件过程过程中记下文件 size;

2、 重现上电后以 OPEN_RECOVER 打开文件（需保证上电到打开文件期间没有进行磁盘写相关操作）

3、 通过写文件接口设置要恢复的文件大小 size ， buffer 可以直接填 NULL;

调用关闭文件接口，如果返回正确，则文件恢复成功。

1.13.2.2 vfs_file_create

- 功能描述：根据用户输入的文件名创建一个文件,用户可获得当前操作文件的句柄
- 函数原型：uint32 vfs_file_create(vfs_mount_t *p_vfs_mount, uint8* ptr_input_name,uint32 lentgh)
- 输入参数描述：

  p_vfs_mount mount 返回的文件系统索引

  ptr_input_name 为将要创建的文件的文件名字符串指针。

  lentgh 创建空文件的长度，为 0 表示创建空文件

- 输出参数描述：

  >0 success – 文件句柄

  0 failed

- 说明：

如输入长度不为 0，将先分配簇链，但文件内容不确定。

1.13.2.3 vfs_file_close

- 功能描述：关闭文件,用户输入需要操作文件的句柄
- 函数原型：bool vfs_file_close(vfs_mount_t *p_vfs_mount, uint32 fhandle)
- 输入参数描述：

  p_vfs_mount mount 返回的文件系统索引

  fhandle 文件句柄

- 输出参数描述：

1 success
0 failed

- 说明：

1.13.2.4 vfs_file_get_size

- 功能描述：取文件的长度，字节为单位
- 函数原型：bool vfs_file_get_size(vfs_mount_t *p_vfs_mount, uint32* output_length, uint8* ptr_file, uint8 mode)
- 输入参数描述：

p_vfs_mount	mount 返回的文件系统索引
output_length	返回的文件长度（以字节为单位）
ptr_file	待获取长度的文件句柄或文件名输入指针
mode	表示参数 ptr_file 的意义。为 0，表示 ptr_file 为文件句柄；为 1 表示 ptr_file 为文件名指针，ptr_file 为 null 表示当前目录项指向的文件。

- 输出参数描述：

1 success
0 failed

- 说明：

虽然 exfat 支持超过 4GB 的文件，但我们方案一般没有操作这么大的文件，所以以 4 字节表示文件长度。

1.13.2.5 vfs_get_time

- 功能描述：获取文件的创建时间或修改时间
- 函数原型：bool vfs_get_time(vfs_mount_t *p_vfs_mount, file_time_t* ptr_output_time, uint8* ptr_input_name, uint8 type)
- 输入参数描述：

p_vfs_mount	mount 返回的文件系统索引
ptr_output_time	时间输出 buf 指针
ptr_input_name	待获取时间的文件的文件名输入指针；为 null 表示当前目录项指向的文件
type	FS_TIME_CREATE 获取创建时间； FS_TIME_MODIFY 获取修改时间。

- 输出参数描述：

1 success
0 failed

- 说明：输出的时间格式见关键数据结构

1.13.2.6 vfs_file_seek

- 功能描述：

  定位文件的字节偏移，实现：

  a.根据相对文件首的偏移量，实现用户程序的顺序读和随机读。

  b.实现从当前位置往文件首或文件尾偏移，实现用户程序的顺序读和随机读。

  c.根据相对文件首的偏移量，实现用户程序的对指定位置数据的修改。

  配合 read 支持用户程序顺序读数据，快进快退以及随机定位读数据，另外可以方便实现回写修改已经生成的文件。

- 函数原型：bool vfs_file_seek(vfs_mount_t *p_vfs_mount, int32 offset, uint8 type, uint32 fhandle)
- 输入参数描述：

p_vfs_mount	mount 返回的文件系统索引
offset	对应 SEEK 偏移量，范围是 2GB
type	对应 SEEK 类型
SEEK_SET	从文件首往文件尾定位，offset 为正数
SEEK_CUR	从当前位置往文件头或尾定位；正数表示向文件尾 seek，负数表示向文件头 seek
SEEK_END	从文件尾往文件首定位，offset 为负数

- 输出参数描述：

1	success
0	failed

- 说明：

1.13.2.7 vfs_file_tell

- 功能描述：取当前读写操作的指针，指针是指相对文件头的字节偏移量。读数据时用户调用该函数记录 AB 点，配合 seek 和 read 实现数据的 AB 读取。写数据时，支持用户程序修改已生成的文件。
- 函数原型：bool vfs_file_tell(vfs_mount_t *p_vfs_mount, uint32* ptr_offset ,uint32 fhandle)
- 输入参数描述：

p_vfs_mount	mount 返回的文件系统索引
ptr_offset	文件当前读写位置相对文件头的偏移量。
fhandle	对应文件操作句柄

- 输出参数描述：

1	success
0	failed

- 说明：

1.13.2.8 vfs_file_read

- 功能描述：从文件的当前位置读数据
- 函数原型：uint32 vfs_file_read(vfs_mount_t *p_vfs_mount, uint8* ptr_data_buffer, uint32 byte_count, uint32 fhandle)
- 输入参数描述：

  p_vfs_mount　　mount 返回的文件系统索引
  ptr_data_buffer　　读操作数据输出 buffer 的指针
  byte_count　　读的字节数。
  fhandle　　操作文件的句柄
- 输出参数描述：

  >0　　实际读到的字节数
  0　　read error
- 说明：

  读完后，文件指针为当前指针加上读的字节数

1.13.2.9 vfs_file_write

- 功能描述：在当前文件位置写入数据
- 函数原型：uint32 vfs_file_write(vfs_mount_t *p_vfs_mount, uint8* ptr_data_buffer, uint32 byte_count, uint32 fhandle)
- 输入参数描述：

  p_vfs_mount　　mount 返回的文件系统索引
  ptr_data_buffer　　写数据 buffer 的指针
  byte_count　　要写的字节数。
  fhandle　　操作文件的句柄
- 输出参数描述：

  >0　　实际写入的字节数
  0　　write error
- 说明：

  写完后，文件指针为当前指针加上写的字节数，写过程中不支持 seek 操作。

  当参数 ptr_data_buffer 为 NULL 和 byte_count 为 0 时，会启用 block 功能，flash 会进行 merge 操作。此操作供录音使用，可减小录音杂音。

1.13.2.10 vfs_set_time

- 功能描述：设置文件的创建时间或修改时间
- 函数原型：bool vfs_set_time(vfs_mount_t *p_vfs_mount, file_time_t* ptr_input_time, uint8* ptr_input_name, uint8 type)

- 输入参数描述：

p_vfs_mount	mount 返回的文件系统索引
ptr_input_time	时间输入 buf 指针；
ptr_input_name	待设置时间的文件的文件名输入指针；为 null 表示前目录项指向的文件
type	FS_TIME_CREATE 设置创建时间 FS_TIME_MODIFY 设置修改时间。

- 输出参数描述：

1	success
0	failed

- 说明：时间输入格式见关键数据结构

1.13.2.11 vfs_cut_file_tail

- 功能描述：关闭文件之前去掉文件的尾部的部分数据
- 函数原型：bool vfs_cut_file_tail(vfs_mount_t *p_vfs_mount, uint32 cut_length, fhandle_t *ptr_file)
- 输入参数描述：

p_vfs_mount	mount 返回的文件系统索引
cut_length	要丢弃的长度，以字节为单位，数值最好应该是 512 的整数倍；
ptr_file	操作文件的句柄

- 输出参数描述：

1	success
0	failed

- 说明： 该接口必须在文件关闭之前调用。

1.13.2.12 vfs_file_divided

- 功能描述：将两个文件分割
- 函数原型：bool vfs_file_divided(vfs_mount_t *p_vfs_mount, uint8* source_file_name, uint8* new_file_name, uint32 divided_length)
- 输入参数描述：

p_vfs_mount	mount 返回的文件系统索引
source_file_name	源文件，也就是被分割文件的文件名；
new_file_name	分割产生的新文件的文件名
divided_length	将原文件分割的长度，如，一个文件的长度为 4096 byte，divided_length 为 1024，那么源文件会被分割为长度为 1024 的文件，而新的文件的长度就是 3072byte

- 输出参数描述：

1	success
0	failed

- 说明：由于文件系统是以簇为单位的，所以文件系统分割有误差。若分割点在一个簇的位置是在簇的 1/2 之前的话，那么该簇的所有数据将会被分割到新文件中去，若分割位置在簇的 1/2 处之后，则该簇的所有数据将会分割到源文件中。

1.13.2.13 vfs_file_insert

- 功能描述：将一个文件的数据插入到另外一个文件
- 函数原型：bool vfs_ file_insert(vfs_mount_t *p_vfs_mount, fhandle_t *fhandle, uint8* insert_file_name)
- 输入参数描述：

p_vfs_mount	mount 返回的文件系统索引
fhandle	源文件的文件句柄，插入时需要将文件的当前位置设置到要插入点；
insert_file_name	要插入文件的文件名，该文件被插入源文件后目录项会被删除

- 输出参数描述：

1	success
0	failed

- 说明：由于文件系统以簇为单位进行操作，所以插入有误差。插入文件最后一个簇未写满，剩余的部分我们会自动填零。

1.13.3 公共操作接口

1.13.3.1 vfs_file_attralter

- 功能描述：获取或修改文件的属性。
- 函数原型：uint8 vfs_file_attralter(vfs_mount_t *p_vfs_mount, uint8 attr, uint8* ptr_file, uint8 mode)
- 输入参数描述：

p_vfs_mount	mount 返回的文件系统索引
attr	ATTR_READ_ONLY (0x01) 只读文件 ATTR_HIDDEN (0x02) 隐藏文件 ATTR_SYSTEM (0x04) 系统文件 ATTR_ARCHIVE (0x20) 存档文件 为 NULL 表示获取属性
ptr_file	待获取或修改属性的文件句柄或文件名输入指针；
mode	表示参数 ptr_file 的意义。为 0，表示 ptr_file 为文件句柄；为 1，表示 ptr_file 为文件名指针，ptr_file 为 null 表示当前目录项指向的文件或目录。

- 输出参数描述：

  >0　　success – 返回要获取或设置的属性值

  0　　failed

- 说明：

1.13.3.2 vfs_file_dir_offset

- 功能描述：获取或设置当前目录项的位置信息
- 函数原型：bool vfs_file_dir_offset(vfs_mount_t *p_vfs_mount, pdir_layer_t* ptr_layer, pfile_offset_t* ptr_file_offset, uint8 mode)
- 输入参数描述：

  p_vfs_mount　　mount 返回的文件系统索引

  ptr_layer　　存储要获取或设置的目录层级信息 buffer 指针；为 NULL 表示当前

  ptr_file　　存储要获取或设置的文件位置信息 buffer 指针；

  mode　　操作方式：0 表示获取，1 表示设置

- 输出参数描述：

  1　　success

  0　　failed

- 说明：

  ptr_layer 和 pfile_offset 不能同时为 NULL。

1.13.3.3 vfs_file_dir_remove

- 功能描述：在当前目录下删除一个用户程序指定目录或文件，删除目录时要求目录为空。
- 函数原型：bool vfs_file_dir_remove(vfs_mount_t *p_vfs_mount, uint8* ptr_input_name, uint8 type)
- 输入参数描述：

  p_vfs_mount　　mount 返回的文件系统索引

  ptr_input_name　　要删除的目录或文件名；如为 null 则直接删除当前目录项指向的目录或文件；

  type　　要操作的类型：0 为目录；1 为文件

- 输出参数描述：

  1　　success

  0　　failed

- 说明：

1.13.3.4 vfs_get_err_info

- 功能描述：获取文件系统出错信息

- 函数原型：uint8 vfs_get_err_info(vfs_mount_t *p_vfs_mount)
- 输入参数描述：

  p_vfs_mount　　mount 返回的文件系统索引
- 输出参数描述：

  0　-- no error
  1　磁盘读写错误
  2　磁盘写保护
  3　磁盘未格式化
  4　文件操作超出文件边界,目录操作超出目录边界
  5　文件操作的目标文件,目录操作的目录项不存在
  6　表示文件操作时没有磁盘空间,不能写数据或者扩展目录；目录操作时没有磁盘空间,不能扩展目录,新建子目录
  7　文件操作时根目录目录项满
  8　删除目录时返回,表示删除的目录非空
- 说明：

1.13.3.5 vfs_file_dir_exist

- 功能描述：判断当前目录是否有指定的子目录或文件
- 函数原型：uint32 vfs_file_dir_exist(vfs_mount_t *p_vfs_mount, uint8* ptr_input_name, uint8 type)
- 输入参数描述：

  p_vfs_mount　　mount 返回的文件系统索引
  ptr_input_name　　要判断的文件或目录名指针
  type　　要判断的类型：0 为目录；1 为文件
- 输出参数描述：

  >0　　文件或目录存在，返回首簇号
  0　　文件或目录不存在
- 说明：

1.13.3.6 vfs_file_rename

- 功能描述：重命名文件
- 函数原型：bool vfs_file_rename(vfs_mount_t *p_vfs_mount, uint8* ptr_new_name, uint8* ptr_file, uint8 mode)
- 输入参数描述：

  p_vfs_mount　　mount 返回的文件系统索引
  ptr_new_name　　新文件名字符串指针
  ptr_file　　待重命名的文件句柄或文件名指针；

mode	表示参数 ptr_file 的意义。为 0，表示 ptr_file 为文件句柄；为 1，表示 ptr_file 为文件名指针，ptr_file 为 null 表示当前目录项指向的文件或目录

- 输出参数描述：

1	success
0	failed

- 说明：

1.13.3.7 vfs_get_name

- 功能描述：取当前的文件名（优先返回长名，没有长名则返回短名）或后缀名
- 函数原型：uint16 vfs_get_name(vfs_mount_t *p_vfs_mount, uint8* ptr_output_name, uint16 longname_length)
- 输入参数描述：

p_vfs_mount	mount 返回的文件系统索引
ptr_output_name	用来存放输出文件名的 buffer 指针。
longname_length	输入为要获取的长名字符数（包括长名标记和结束符），如为 0 表示获取后缀名。

- 输出参数描述：

>0	success – 实际返回的获取到的文件名字符数。为取长名时返回实际函数输出的长名字符的个数（包括长名标记和结束符 0x0000），为短名时返回 11；获取后缀名时返回 3。
0	failed

- 说明：

文档中所有文件名和目录名如短名则指大写的 8+3 字符数组格式,占用 11 个 byte,不足补 20h；输入的文件名长度包括长名标记和结束符共 2 个字符，所以如要获取长名的 6 个字符，则输入 longname_length 为 8，且输出 buffer 至少要 2*8=16 字节。

1.13.3.8 vfs_get_space

- 功能描述：获取磁盘分区空间，根据输入参数不同选择要求返回磁盘分区总的扇区数还是剩余扇区数
- 函数原型：bool vfs_get_space(vfs_mount_t *p_vfs_mount, uint32* ptr_sector_count, uint8 type)
- 输入参数描述：

p_vfs_mount	mount 返回的文件系统索引
ptr_sector_count	返回分区的扇区数
type	0 表示调用将返回表示当前磁盘分区总空间的扇区数； 1 表示返回当前磁盘分区剩余空间的扇区数

- 输出参数描述：

  1　　获取成功

  0　　获取失败

- 说明：

1.13.3.9 vfs_create_volume

- 功能描述：创建卷标
- 函数原型：bool vfs_create_volume(vfs_mount_t *p_vfs_mount, uint8* ptr_input_name)
- 输入参数描述：

  p_vfs_mount　　mount 返回的文件系统索引

  ptr_input_name　　卷标名字符串指针

- 输出参数描述：

  1　　success

  0　　failed

- 说明：

fat,fat32 的卷标名为 11 个字符，大写，不足十一个字符补 20h，exfat 卷标名为 11 个 unicode 编码的字符。

1.13.3.10 vfs_jump_to_direntry

- 功能描述：根据提供的目录信息，直接跳转到该目录的起始位置
- 函数原型：bool fat_jump_to_direntry(vfs_mount_t *vfs_mount, uint32 mode, uint8* direntry_info,uint8* reserve)
- 输入参数描述：

  p_vfs_mount　　mount 返回的文件系统索引

  mode　　0x80000000-从尾往前检索；其他值：从头往后检索；

  direntry_info　　需要跳转到的目录信息

  reserve　　保留，暂时无用

- 输出参数描述：

  1　　success

  0　　failed

1.13.3.11 vfs_dir_current_entry_file

- 功能描述：通过后缀检索定位到文件位置
- 函数原型：uint32 fat_dir_current_entry_file(vfs_mount_t *vfs_mount, uint32 mode, uint8* input_name, uint32 ext_name_bitmap)
- 输入参数描述：

  p_vfs_mount　　mount 返回的文件系统索引

  mode　　0x80000000-从尾往前检索；其他值：从头往后检索；如果为 0 或 0x80000000 时，表示检索定位到第一个配置后缀的文件的位置。其他值:mode=n 则表示检索定位到第 N 个配置后缀的文件的位置。

ptr_input_name	存储检索定位到的文件位置信息。
ext_name_bitmap	如是有效内存地址，则表示输入扩展名字符串的地址；如不是则表示要 dir 的 bitmap。此参数如最高位为 1 表示传入的内存地址，否则则是 bitmap 值， 具体的 bitmap 对应的意义请参照 vfs_interface.h 第 43 行下的定义

- 输出参数描述：

非 0	success
0	failed

1.13.3.12 vfs_file_move

- 功能描述：删除或增加目录项
- 函数原型： bool vfs_file_move(vfs_mount_t *vfs_mount, void *fat_direntry, uint8 *ptr_file_name, uint8 mode)
- 输入参数描述：

p_vfs_mount	mount 返回的文件系统索引
fat_direntry	目录项信息指针
ptr_file_name	待增加或删除的目录名字符串指针；当 mode==0 时，如果为 null 表示前目录项指向的文件;否则参数无效
mode	0-删除目录项；1-增加目录项

- 输出参数描述：

1	success
0	failed

1.14 关键数据结构定义

1.14.1 dir_layer_info_t

- 功能描述： 定义目录层级结构
- 数据结构原型：

```
typedef struct
{
    uint32 cluster;
    uint32 offset;
} dir_layer_info_t;
```

- 必要参数重点说明：

cluster：目录的其实簇号

offset：目录内的偏移

1.14.2 fdir_layer_t

- 功能描述： 八级目录记录结构
- 数据结构原型：

```
typedef struct
{
    dir_layer_info_t dir_layer_info[MAX_DIR_LAYER];
    uint8 cur_layer; // dir layer
} fdir_layer_t;
```

- 必要参数重点说明：

dir_layer_info：记录从根目录到当前目录的目录层次信息，及目录的首簇号及目录内的偏移

cur_layer： 当前的目录级数。0 为根目录，以此类推，不能超过八级目录。

1.14.3 fhandle_t

- 功能描述：文件句柄结构
- 数据结构原型：

```
typedef struct
{
    uint8 mode;//打开模式
    uint32 start_cluster; // starting cluster of file
    uint32 file_size; // size of file in bytes
    uint32 de_current_cluster; // the cluster of the file direntry
    uint32 de_cluster_offset; // the offset of the file direntry in cluster
    uint32 ct_current_cluster; // current cluster of file op
    uint32 ct_cluster_offset; // the offset of the file op in cluster
    uint32 postion; // file op postion
    uint16 de_creat_date; // create date
    uint16 de_creat_time; // create time
    void *ehance_addr;
} fhandle_t;
```

- 必要参数重点说明：

ehance_addr：为不同打开模式准备。以普通模式打开，此值为 NULL，若为快速 seek 模式打开的话，此值为指向 add_fhandle_t 结构的指针等。

1.14.4 fat_direntry_t

- 功能描述：目录项结构
- 数据结构原型：

```
typedef struct
{
    uint8 deName[8]; // filename, blank filled
    uint8 extension[3]; // extension, blank filled
    uint8 attributes; // file attributes
    uint8 deLowerCase; // NT VFAT lower case flags
    uint8 deCHundredth; // hundredth of seconds in CTime
    uint16 de_creat_time; // create time
    uint16 de_creat_date; // create date
    uint16 de_access_date; // access date
    uint16 high_cluster; // high bytes of cluster number
    uint16 de_modify_time; // last update time
    uint16 de_modify_date; // last update date
    uint16 start_cluster; // starting cluster of file
    uint32 file_size; // size of file in bytes
} fat_direntry_t;
```

- 必要参数重点说明：

1.14.5 file_time_t

- 功能描述：文件时间的结构
- 数据结构原型：

```
struct timebuf
{
    WORD year;
    uchar month;
    uchar day;
    uchar hour;
    uchar minute;
    uchar second;
} file_time_t;
```

- 必要参数重点说明：

1.14.6 vfs_mount_t

- 功能描述：磁盘文件系统信息

- 数据结构原型：

```
typedef struct {
    unsigned char drv_type;
    unsigned char fs_type;
    unsigned char disk;
    unsigned int start_offset;
    void *fat_private_data;
} vfs_mount_t;;
```

- 必要参数重点说明：

drv_type ： 磁盘驱动类型，标识是卡，flash，usb
fs_type： 文件系统类型，exfat，fat，fat32
fat_private_data： 磁盘的私有信息，包括 DBR，及目录结构等信息。

AUDIO_DEVICE 接口

1.15 概述

AUDIO DEVICE 驱动为上层应用访问 AD/DA/PA 等模块提供接口，通过调用这些接口能实现声音输出、监听和声音采集等功能，同时增加对硬件 asrc 的配置、开关、和采样率转换等功能。

1.16 接口列表

表格 5-1：AUDIO DEVICE 操作命令一览表

Command	函数功能说明
ENABLE_PA	打开 PA
DISABLE_PA	关闭 PA
SET_PA_VOLUME	设置 PA 的音量
GET_PA_VOLUME	获取 PA 的音量
ENABLE_DAC	打开 DAC
DISABLE_DAC	关闭 DAC
SET_DAC_RATE	设置 DAC 的采样率

Command	函数功能说明
GET_DAC_RATE	关闭 DAC 的采样率
ENABLE_AIN	打开 FM 或者 MIC 或者 LINEIN 的 AIN 输入
DISABLE_AIN	关闭 FM 或者 MIC 或者 LINEIN 的 AIN 输入
SET_AIN_GAIN	设置 AIN 输入的增益
ENABLE_AIN_OUT	打开 AA 通路开关
DISABLE_AIN_OUT	关闭 AA 通路开关
ENABLE_ADC	打开 ADC
DISABLE_ADC	关闭 ADC
SET_ADC_RATE	设置 ADC 采样率
CONFIG_ASRC	配置 asrc
CLOSE_ASRC	关闭 asrc
SET_ASRC_RATE	设置 asrc 采样率转换的抽样比
SET_EFFECT_PARAM	设置音效参数
GET_FEATURE_INFO	获取音效处理信息

1.17 函数接口定义

1.17.1 入口函数

- 功能描述：audio device 操作接口，适用于使用 PA，DAC，ADC 及 ASRC 等场合。
- 函数原型：extern void* audio_device_op_entry(void *param1, void *param2, void *param3, audio_device_cmd_e cmd_type);
- 输入参数描述：cmd_type 为命令类型，不同的命令 param1，param2，param3 含义不一样，详见下节命令一览表。
- 输出参数描述：返回结果（转为 void 型），0 值：正常，非 0 值：异常
- 说明：

1.17.2 ENABLE_PA

- 功能描述：打开 PA/I2S/SPDIF 等音频输出模块

- 函数原型：int32 enable_pa(uint32 ddv_sel, uint32 pa_swing, uint32 aout_type)
- 输入参数描述：

  ddv_sel　　类型相关的参数选择、内部 pa 用于选择直驱非直驱；对于 i2s、spdif 用于选 mfp 组

  pa_swing　　内部 pa 使用：bit6 用于选择输出峰值 bit6(1<<6)2.8pp，bit6(0<<6)1.6pp

  aout_type　　用于选择音频输出的类型，见结构体 aout_type_e
- 输出参数描述：

  0　　success

  -1　　failed
- 说明：

进入音乐应用只需打开一次，退出时关闭即可。对于内部 pa 来说第一次打开的时间花费的时间会多一些，或者有轻微杂音。

1.17.3 DISABLE_PA

- 功能描述：关闭 PA/I2S/SPDIF 等音频输出模块
- 函数原型：int32 disable_pa(uint32 aout_type, void *null2, void *null3)
- 输入参数描述：

  aout_type　　用于选择音频输出的类型，见结构体 aout_type_e
- 输出参数描述：

  0　　success

  -1　　failed
- 说明：退出音乐时关闭。

1.17.4 SET_PA_VOLUME

- 功能描述：设置内部 pa 音量
- 函数原型：int32 set_pa_volume(uint32 vol_hard, void *null2, void *null3)
- 输入参数描述：

  vol_hard　　设置到寄存器的音量值（0-40）
- 输出参数描述：

  0　　success

  -1　　failed
- 说明：

1.17.5 GET_PA_VOLUME

- 功能描述：获取内部 pa 音量
- 函数原型：int32 get_pa_volume(void *null1, void *null2, void *null3)
- 输入参数描述：
- 输出参数描述：
  音量值（0-40）
- 说明：

1. 17. 6 ENABLE_DAC

- 功能描述：打开 dac 模块
- 函数原型：int32 enable_dac(uint32 src_type, uint32 dac_chan, void *null3)
- 输入参数描述：
  src_type 选择 dac fifo 的输入源，见结构体 dac_fifo_in_e
  dac_chan 需要使能的 dac 的通道，见结构体 dac_chenel_e
- 输出参数描述：
  0 success
  -1 failed
- 说明：

1. 17. 7 DISABLE_DAC

- 功能描述：关闭 dac 模块
- 函数原型：int32 disable_dac(uint32 dac_chan, void *null2, void *null3)
- 输入参数描述：
  dac_chan 需要使能的 dac 的通道，见结构体 dac_chenel_e
- 输出参数描述：
  0 success
  -1 failed
- 说明：

1.17.8 SET_DAC_RATE

- 功能描述：设置 dac 采样率
- 函数原型：int32 set_dac_rate(uint32 dac_rate, uint32 chanel_no, void *null3)
- 输入参数描述：
  dac_rate　采样率值（单位 khz）
  chanel_no　未使用
- 输出参数描述：
  0　success
  -1　failed
- 说明：

1.17.9 GET_DAC_RATE

- 功能描述：设置 dac 采样率
- 函数原型：int32 get_dac_rate(void *null1, void *null2, void *null3)
- 输入参数描述：
- 输出参数描述：
  当前 dac 采样率值（单位 khz）
- 说明：

1.17.10 ENABLE_AIN

- 功能描述：使能模拟输入
- 函数原型：int32　enable_ain(uint32 ain_type, uint32 ain_gain, void *null3)
- 输入参数描述：
  ain_type　模拟输入选择，见结构体 mmm_ai_type_t
  ain_gain　模拟输入的增益
  0x0: -12dB
  0x1: -3dB
  0x2: 0dB
  0x3: 1.5dB
  0x4: 3.0dB
  0x5: 4.5dB
  0x6: 6.0dB

0x7: 7.5db

- 输出参数描述：
  0 success
  -1 failed
- 说明：

1.17.11 DISABLE_AIN

- 功能描述：关闭模拟输入
- 函数原型：int32 disable_ain(uint32 ain_type, void *null2, void *null3)
- 输入参数描述：
  ain_type 模拟输入选择，见结构体 mmm_ai_type_t
- 输出参数描述：
  0 success
  -1 failed
- 说明：

1.17.12 SET_AIN_GAIN

- 功能描述：设置模拟输入的增益
- 函数原型：int32 set_ain_gain(uint32 ain_type, uint32 ain_gain, void *null3)
- 输入参数描述：
  ain_type 模拟输入选择，见结构体 mmm_ai_type_t
  ain_gain 模拟输入的增益
  0x0: -12dB
  0x1: -3dB
  0x2: 0dB
  0x3: 1.5dB
  0x4: 3.0dB
  0x5: 4.5dB
  0x6: 6.0dB
  0x7: 7.5db
- 输出参数描述：
  0 success
  -1 failed
- 说明：

1.17.13 ENABLE_AIN_OUT

- 功能描述：打开 AA 通道
- 函数原型：int32 enable_ain_out(uint32 out_mode, void *null2, void *null3)
- 输入参数描述：
  out_mode　当前模拟输入，见结构体 mmm_ai_type_t

- 输出参数描述：
  0　　success
  -1　　failed
- 说明：

1.17.14 ENABLE_AIN_OUT

- 功能描述：关闭 AA 通道
- 函数原型：int32 disable_ain_out(uint32 out_mode, void *null2, void *null3)

- 输入参数描述：
  out_mode　当前模拟输入，见结构体 mmm_ai_type_t

- 输出参数描述：
  0　　success
  -1　　failed
- 说明：

1.17.15 ENABLE_ADC

- 功能描述：使能 ADC
- 函数原型：int32 enable_adc(uint32 src_type, uint32 adc_gain, void *null3)
- 输入参数描述：

src_type　adc fifo 的输出选择，见结构 adc_fifo_out_e

adc_gain　adc 的增益

0x0: 0dB
0x1: 3dB
0x2: 6dB
0x3: 9dB
0x4:12dB
0x5: 15dB
0x6: 18dB
0x7: 21db
0x8: 24dB
0x9: 27dB
0xa: 30dB
0xb: 33dB
0xc:36dB
0xd: 39dB
0xe: 42dB
0xf: 45db

- 输出参数描述：

0　success
-1　failed

- 说明：

1.17.16 DISABLE_ADC

- 功能描述：关闭 ADC
- 函数原型：int32 disable_adc(void* null, void* null, void *null3)
- 输入参数描述：
- 输出参数描述：

0　success
-1　failed

- 说明：

1.17.17 SET_ADC_RATE

- 功能描述：设置 ADC 采样率

- 函数原型：int32 set_adc_rate(uint32 adc_rate, void *null2, void *null3)
- 输入参数描述：
  adc_rate　adc 采样率（单位 khz）
- 输出参数描述：
  0　　success
  -1　　failed
- 说明：

1.17.18 CONFIG_ASRC

- 功能描述：配置 ASRC
- 函数原型：int32 config_asrc(uint32 nTestSel, uint8 asrc_mode)
- 输入参数描述：
  nTestSel　asrc 的类型选择 –头文件// ram select table
  asrc_mode asrc 参数选择
- 输出参数描述：
  0　　success
  -1　　failed
- 说明：

1.17.19 CLOSE_ASRC

- 功能描述：关闭 ASRC
- 函数原型：int32 audio_device_close_asrc(uint32 nTestSel)
- 输入参数描述：
  nTestSel　asrc 的类型选择 –头文件// ram select table
- 输出参数描述：
  0　　success
  -1　　failed
- 说明：

1.17.20 SET_ASRC_RATE

- 功能描述：配置 ASRC 抽样比
- 函数原型：int32 set_asrc_rate(uint32 asrc_rate, uint32 chanel_no, uint32 asrc_offset)
- 输入参数描述：
  asrc_rate　asrc 输入采样率
  chanel_no asrc 通道　0—通道 out0；1—通道 out1；2—通道 in
  asrc_offset　微调的偏移

- 输出参数描述：
  0　　success
  -1　　failed
- 说明：

1.17.21 SET_EFFECT_PARAM

- 功能描述：设置音效参数
- 函数原型：int32 set_effect_param(uint32 set_type, void* param_ptr, void* null3)
- 输入参数描述：
  set_type　设置类型，见结构体 set_effect_type_e
  param_ptr　参数指针，与 set_type 相关

- 输出参数描述：
  0　　success
  -1　　failed
- 说明：

1.17.22 GET_FEATURE_INFO

- 功能描述：获取音效处理信息
- 函数原型：int32 get_feature_info(uint32 set_type, void* info_ptr, void* null3)
- 输入参数描述：
  set_type　设置类型，见结构体 set_effect_type_e
  param_ptr　参数指针，与 set_type 相关

- 输出参数描述：

  0 success

  -1 failed

- 说明：

炬芯（珠海）科技有限公司

地址：珠海市唐家湾镇高新区科技四路 1 号 1#厂房一层 C 区
电话：+86-756-3392353
传真：+86-756-3392251
邮政编码：519085
网址：http://www.actions-semi.com
电子邮件 （业务）: mp-sales@actions-semi.com
（技术支持）: mp-cs@actions-semi.com