

文件系统接口设计说明书

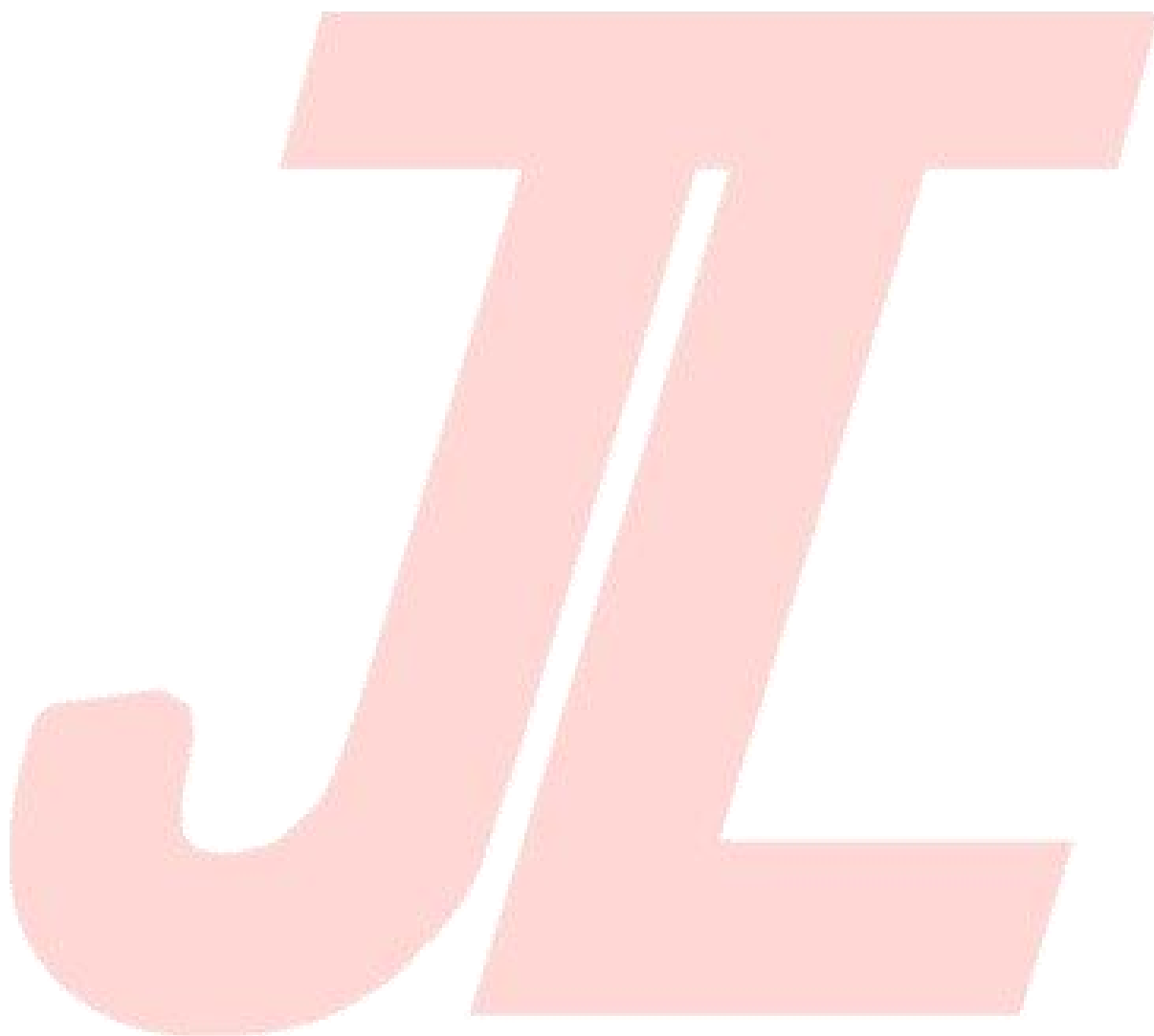
珠海市杰理科技股份有限公司

Zhuhai Jieli Technologyco.,LTD

版权所有，未经许可，禁止外传

修改记录

版本	更新日期	描述
V1.0		
V1.1		



目录

1. 文档介绍.....	4
1.1. 文档目的.....	4
1.2. 参考文献.....	4
1.3. 术语与缩写词.....	4
2. 功能概述.....	4
3. 其他系统/模块的调用关系.....	4
4. 性能要求.....	4
5. 接口模块功能介绍.....	5
5.1. 文件浏览.....	错误！未定义书签。
5.1.1. 功能介绍.....	5
5.1.2. 接口介绍.....	5
5.2. 文件删除.....	错误！未定义书签。
5.2.1. 实现介绍.....	错误！未定义书签。
5.2.2. 接口介绍.....	5
5.3. 滑动杆.....	错误！未定义书签。
5.3.1. 功能介绍.....	6
5.3.2. 接口.....	6
5.4. 消息处理.....	错误！未定义书签。
5.4.1. 功能介绍.....	6
5.4.2. 接口.....	7
5.5. 资源文件处理.....	错误！未定义书签。
5.5.1. 功能介绍.....	7
5.5.2. 接口介绍.....	7
5.6. 菜单列表.....	错误！未定义书签。
5.6.1. 功能介绍.....	8
5.6.2. 接口介绍.....	8
5.7. 进度条.....	错误！未定义书签。
5.7.1. 功能介绍.....	9
5.7.2. 接口介绍.....	9
5.8. 滚动条.....	9
5.8.1. 功能介绍.....	11
5.8.2. 接口介绍.....	11
5.9. 出错提示.....	错误！未定义书签。
5.9.1. 功能介绍.....	11
5.9.2. 接口介绍.....	11
6. COMMON 链接地址空间.....	错误！未定义书签。

1. 文档介绍

1.1. 文档目的

说明文件系统相关接口使用和作用，帮助 AP 开发人员方便使用提供的 接口。

1.2. 参考文献

[1].

1.3. 术语与缩写词

缩写、术语	解 释
AP	Application, 应用程序

2. 功能概述

文件系统接口 模块主要提供包含文件浏览，文件删除，打开，创建，读写，等接口给 AP 应用调用。

3. 其他系统/模块的调用关系

音乐，录音，解码等模式会使用到。

4. 性能要求

传入正确的参数。

5. 接口模块功能介绍

5.1. 挂载设备

5.1.1. 功能介绍

由设备名、路径、文件系统类型来挂载设备和文件系统

5.1.2. 接口介绍

函数原型	<code>struct imount *mount(const char *dev_name, const char *path, const char *fs_type, int cache_num, void *dev_arg); //挂载</code>
功能描述	挂载设备
参数说明	<code>dev_name</code> // 设备名字 <code>Path</code> //路径 <code>fs_type</code> // 文件系统类型 <code>cache_num</code> // 缓存数目 <code>dev_arg</code> //设备参数
输出	<code>* \return</code> mt 结构句柄
例子	<code>void *fmnt = mount(p->name, p->storage_path, p->fs_type, 3, NULL);</code>
关联模块	<code>dev_manager.c</code>
补充说明	注意路径和文件系统类型选择

5.2. 卸载设备

5.2.1. 功能介绍

卸载设备

5.2.2. 接口介绍

函数原型	<code>int unmount(const char *path);</code>
功能描述	卸载设备
参数说明	<code>Path</code> //路径
输出	<code>* \return</code> 返回相应的操作消息处理值 0 为成功
例子	<code>unmount(dev->parm->storage_path);</code>

关联模块	dev_manager.c
补充说明	

5.3. 格式化

5.3.1. 功能介绍

格式化设备。

5.3.2. 接口介绍

函数原型	int f_format(const char *path, const char *fs_type, u32 clust_size); //格式化接口
功能描述	格式化设备
参数说明	path// 路径 fs_type// 文件系统类型 clust_size// 簇大小
输出	*\return 0 格式化成功, 非 0 失败
关联模块	
例子	result =f_format(“storage/sd0/C/”, “fat”, 32*1024);
补充说明	暂时只支持 fat32 得格式化

5.4. 文件打开

5.4.1. 功能介绍

```
/*-----*/  
/** @brief:  
@param: fopen 扩展功能  
@note:    fopen 自动打开、创建文件夹和文件。  
说明:  
1. 设备路径+文件, 其中文件传入格式:"music/test/1/2/3/pk*.wav"   "JL_REC/AC69****.wav"  
"JL_REC/AC690000.wav"  
2. 文件名带*号, 带多少个*表示多少个可变数字, 最多为 8+3 的大小, 如表示可变数字名称变为  
XXX0001,XXXX002 这样得格式, 不带*号则只创建一个文件, 写覆盖。  
@date: 2020-07-22
```

版权所有, 侵权必究

*/

/*-----*/。

5.4.2. 接口介绍

函数原型	FILE *fopen(const char *path, const char *mode);
功能描述	打开文件
参数说明	Path// 文件名路径 Mode //打开模式（只读，写，可创建）
输出	* \return 返回获得得文件句柄
关联模块	/encode_write_file.c, app_config.c, audio_dec_record.c
例子	record_file = fopen(path, "r");
补充说明	注意路径需要带上根路径

5.5. 文件读写

5.5.1. 功能介绍

读出打开文件得内容。

5.5.2. 接口介绍

函数原型	int fread(FILE *file, void *buf, u32 len);
功能描述	读资源文件
参数说明	File// 文件句柄 Buf // 读出内容放置 buf Len // 需要读出内容得长度（一般为 buf 长度）
输出	返回读出得长度
关联模块	app_config.c, norflash_update.c, audio_dec_file.c
例子	rlen = fread(dec->file, buf, len);
补充说明	

5.6. 写文件

5.6.1. 功能介绍

文件写入内容

5.6.2. 接口介绍

函数原型	<code>int fwrite(FILE *file, void *buf, u32 len);</code>
功能描述	写资源文件
参数说明	<code>File</code> // 文件句柄 <code>Buf</code> // 写入得内容放置 <code>buf</code> <code>Len</code> // 需要写入内容得长度（一般为 <code>buf</code> 长度）
输出	返回写入得长度
关联模块	<code>encode_write_file.c</code> ， <code>norflash_update.c</code>
例子	<code>ret = fwrite(wfil->file, frame, len);</code>
补充说明	

5.7. Seek 到相应位置

5.7.1. 功能介绍

Seek 到相应得位置。

5.7.2. 接口介绍

函数原型	<code>int fseek(FILE *file, int offset, int orig);</code>
功能描述	Seek 到相应得位置
参数说明	<code>File</code> //文件句柄 <code>Offset</code> // 偏移量 <code>Orig</code> // 偏移方式
输出	<code>*\return</code> 返回偏移得值
关联模块	<code>encode_write_file.c</code> ， <code>music_id3.c</code>
例子	<code>fseek(wfil->file, 0, SEEK_SET);</code>
补充说明	偏移方式 <code>#define SEEK_SET 0</code> /* Seek from beginning of file. */

	<pre>#define SEEK_CUR 1 /* Seek from current position. */ #define SEEK_END 2 /* Seek from end of file. */</pre>
--	---

5.8. 获取文件长度

5.8.1. 功能介绍

获取文件长度

5.8.2. 接口介绍

函数原型	<code>int flen(FILE *file);</code>
功能描述	获取文件长度
参数说明	File//文件结构.
输出	* \return 当前文件长度
关联模块	audio_dec_file.c
例子	<code>len = flen(dec->file);</code>
补充说明	

5.9. 获取文件指针位置

5.9.1. 功能介绍

获取文件指针位置

5.9.2. 接口介绍

函数原型	<code>int fpos(FILE *file);</code>
功能描述	获取文件指针位置
参数说明	File//文件结构.
输出	* \return 当前文件指针位置

关联模块	/encode_write_file.c, music_id3.c
例子	u32 cur_fptr = fpos(file);
补充说明	

5.10. 关闭文件

5.10.1. 功能介绍

关闭文件

5.10.2. 接口介绍

函数原型	int fclose(FILE *file);
功能描述	关闭文件
参数说明	File //文件结构.
输出	* \return 无意义
关联模块	/encode_write_file.c, /app_config.c
例子	fclose(wfil->file);
补充说明	关闭后, 将 file=NULL 避免成为野指针

5.11. 删除文件

5.11.1. 功能介绍

删除文件

5.11.2. 接口介绍

函数原型	int fdelete(FILE *file);
功能描述	删除文件
参数说明	File //文件结构.
输出	* \return 无意义
关联模块	/encode_write_file.c, music_player.c
例子	fdelete(wfil->file);

补充说明	删除后，将 file=NULL 避免成为野指针

5.12. 获取文件名

5.12.1. 功能介绍

获取当前文件名称，支持长短文件名。

5.12.2. 接口介绍

函数原型	int fget_name(FILE *file, u8 *name, int len);
功能描述	获取当前文件名称，支持长短文件名。
参数说明	File //文件结构. Name //文件名 buf Len //长度
输出	* \return 获取到的文件名长度
关联模块	music_func.c, audio_dec_tone.c
例子	int lfn_len = fget_name(file, lfn_buf, 512);
补充说明	Buf 最好足够大，传入 len < 16 时，默认只获取短文件名

5.13. 判断获取文件名是否长文件名

5.13.1. 功能介绍

判断是不是 unicode 码。

5.13.2. 接口介绍

函数原型	int fget_name_type(char *path, int len); //判断是不是 unicode 码
功能描述	判断是不是 unicode 码
参数说明	Path //文件名 buf
	究

	Len //文件名长度
输出	* \return * \retval TRUE 是 Unicode * \retval FALSE 不是
关联模块	music_func.c
例子	int unicode = fget_name_type(lfn_buf, lfn_len);
补充说明	

5.14. 文件扫描接口

5.14.1. 功能介绍

文件扫描。

5.14.2. 接口介绍

函数原型	struct vfscan *fscan(const char *path, const char *arg, u8 max_depth);
功能描述	文件扫描
参数说明	Path //路径 Arg //s 扫描方式 max_depth // 扫描深度，最大为 9
输出	* \return vfscan 句柄
关联模块	dev_manager.c
例子	fsn = fscan(fopr->fsn_path, sel->scan_type, 9);
补充说明	/* arg: * -t 文件类型 * -r 包含子目录 * -d 扫描文件夹 * -a 文件属性 r: 读, /: 非 * -s 排序方式, t:按时间排序, n:按文件号排序 */

5.15. 可打断文件扫描接口

5.15.1. 功能介绍

文件扫描，可打断。

5.15.2. 接口介绍

函数原型	<code>struct vfscan *fscan_interrupt(const char *path, const char *arg, u8 max_depth, int (*callback)(void)); //可打断扫描</code>
功能描述	文件扫描，可打断
参数说明	<code>Path</code> //路径 <code>Arg</code> //s 扫描方式 <code>max_depth</code> // 扫描深度，最大为 9 <code>Callback</code> ///回调函数，用于打断
输出	<code>*\return</code> vfscan 句柄
关联模块	<code>dev_manager.c </code>
例子	<code>fsn = fscan_interrupt((const char *)fsn_path, parm, DEV_MANAGER_SCAN_DISK_MAX_DEPTH, callback);</code>
补充说明	回调函数返回 1 则打断

5.16. 进入指定子目录扫描

5.16.1. 功能介绍

/进入指定子目录，只扫此目录下文件信息。

5.16.2. 接口介绍

函数原型	<code>struct vfscan *fscan_enterdir(struct vfscan *fs, const char *path);</code>
功能描述	/进入指定子目录，只扫此目录下文件信息
参数说明	<code>Fs</code> //VFScan 句柄 <code>Path</code> //子目录句柄 w
输出	<code>*\return</code> 相应的句柄

关联模块	jlfat_api.c
例子	fsn = fscan_enterdir(fsn, loop_str);
补充说明	支持相对路径

5.17. 返回上一层

5.17.1. 功能介绍

返回上一层目录。

5.17.2. 接口介绍

函数原型	struct vfscan *fscan_exitdir(struct vfscan *fs);
功能描述	返回上一层目录
参数说明	Fs //VFScan 句柄
输出	* \return 相应的句柄
关联模块	jlfat_api.c
例子	fsn = fscan_exitdir(fsn);
补充说明	

5.18. 释放 VFSCAN 结构句柄

5.18.1. 功能介绍

释放句柄。

5.18.2. 接口介绍

函数原型	void fscan_release(struct vfscan *fs);
功能描述	释放句柄
参数说明	Fs //VFScan 句柄
输出	* \return 无
关联模块	dev_manager.c
例子	fscan_release(fsn);
补充说明	究

5.19. 选择指定文件

5.19.1. 功能介绍

选择指定文件打开。

5.19.2. 接口介绍

函数原型	<code>FILE *fselect(struct vfscan *fs, int selt_mode, int arg);</code> //选择指定文件
功能描述	选择指定文件打开
参数说明	<code>Fs</code> ///VFSCAN 句柄 <code>selt_mode</code> // 按什么方式选择 <code>Arg</code> /// 传入选择的参数
输出	<code>* \return</code> 文件句柄
关联模块	<code>music_player.c</code>
例子	<code>__this->file = fselect(__this->fsn, FSEL_PREV_FILE, 0);</code> ///选择上一曲
补充说明	支持，按簇号、序号、路径选择

5.20. 检查挂载目录是否存在

5.20.1. 功能介绍

检查挂载目录是否存在。

5.20.2. 接口介绍

函数原型	<code>int fdir_exist(const char *dir);</code> /
功能描述	检查挂载目录是否存在
参数说明	<code>Dir</code> ///目录路径
输出	<code>* \return</code> 0 存在 ， 1 不存在
关联模块	无

例子	Int exist = fdir_exist(dir)
补充说明	

5.21. 获取文件属性

5.21.1. 功能介绍

获取文件属性。

5.21.2. 接口介绍

函数原型	int fget_attr(FILE *file, int *attr);
功能描述	获取文件属性
参数说明	File // 文件句柄 Attr //属性
输出	* \return 获取值在 attr 中
关联模块	jlfat_api.c
例子	fget_attr(file, &attr);
补充说明	传入指针进去

5.22. 设置文件属性

5.22.1. 功能介绍

设置文件属性。

5.22.2. 接口介绍

函数原型	int fset_attr(FILE *file, int *attr);
功能描述	设置文件属性
参数说明	File // 文件句柄 Attr //属性
输出	* \return 设置值在 attr 中
关联模块	
例子	fset_attr(file, attr);
补充说明	

5.23. 获取文件相关信息

5.23.1. 功能介绍

获取文件相关信息。

5.23.2. 接口介绍

函数原型	int fget_attr(FILE *file, struct vfs_attr *attr);//获得文件相关信息如属性、簇号、大小等
功能描述	获取文件相关信息
参数说明	File // 文件句柄 Attr //记录相应信息的结构体
输出	* \return 获取值在 attr 中
关联模块	encode_write_file.c, music_player.c, /custom_cfg.c
例子	fget_attr(__this->file, &attr);
补充说明	传入结构体指针进去

5.24. 获取分区 part

5.24.1. 功能介绍

获取分区 part。

5.24.2. 接口介绍

函数原型	struct vfs_partition *fget_partition(const char *path);
功能描述	获取分区 part
参数说明	Path //路径
输出	* \return vfs_partition 句柄
关联模块	
例子	struct vfs_partition part = fget_partition(path);

5.25. 获取文件夹信息

5.25.1. 功能介绍

获取文件夹信息。

5.25.2. 接口介绍

函数原型	<code>int fget_folder(struct vfscan *fs, struct ffolder *arg);</code> //获取文件夹序号和文件夹内文件数目
功能描述	获取文件夹信息
参数说明	<code>Fs</code> ///vfscan 结构句柄 <code>Arg</code> // 文件夹信息结构句柄
输出	<code>*\return</code> 文件夹信息结构句柄
关联模块	
例子	<code>fget_folder(fopr->fsn, &folder);</code>
补充说明	

5.26. 设置长文件名 Buf

5.26.1. 功能介绍

设置长文件名 Buf。

5.26.2. 接口介绍

函数原型	<code>int fset_lfn_buf(struct vfscan *fs, void *arg);</code> //设置长文件名 buf
功能描述	设置长文件名 Buf
参数说明	<code>Fs</code> //vfscan 句柄 <code>Arg</code> // buf
输出	<code>*\return</code>
关联模块	<code>/dev_manager.c</code>
例子	<code>fset_lfn_buf(fsn, dev->lfn_buf);</code>
补充说明	

5.27. 设置长文件夹名 Buf

5.27.1. 功能介绍

设置长文件夹名 Buf。

5.27.2. 接口介绍

函数原型	<code>int fset_ldn_buf(struct vfscan *fs, void *arg);</code> //设置长文件名 buf
功能描述	设置长文件夹名 Buf
参数说明	<code>Fs</code> //vfscan 句柄 <code>Arg</code> // buf
输出	<code>* \return</code>
关联模块	
例子	<code>fset_ldn_buf(fopr->fsn, ldn_buf);</code>
补充说明	

5.28. 设置后缀类型

5.28.1. 功能介绍

设置后缀类型。

5.28.2. 接口介绍

函数原型	<code>int fset_ext_type(const char *path, void *ext_type);</code> //设置后缀类型
功能描述	设置后缀类型
参数说明	<code>Path</code> // 路径 <code>ext_type</code> // 后缀类型
输出	<code>* \return</code> 0 成功, 非 0 不成功
关联模块	<code>file_bs_deal.c</code>
例子	<code>fset_ext_type(dev_manager_get_bs_root_path(fil_bs->dev), ext_name);</code>
补充说明	

5.29. 打开目录

5.29.1. 功能介绍

文件浏览使用，打开目录。

5.29.2. 接口介绍

函数原型	<code>int fopen_dir_info(const char *path, FILE **pp_file, void *dir_dj);</code>
功能描述	打开目录
参数说明	<code>Path</code> // 路径 <code>pp_file</code> /// 文件句柄 <code>Dir_dj</code> //目录信息句柄
输出	<code>* \return</code> 目录项总数
关联模块	<code>file_bs_deal.c</code>
例子	<code>total_dir = fopen_dir_info(dev_manager_get_bs_root_path(fil_bs->dev), &fil_bs->file, 0);</code>
补充说明	

5.30. 进入目录

5.30.1. 功能介绍

文件浏览使用，进入目录。

5.30.2. 接口介绍

函数原型	<code>int fenter_dir_info(FILE *file, void *dir_dj);</code> //进入目录
功能描述	进入目录
参数说明	<code>Path</code> // 路径 <code>pp_file</code> /// 文件句柄 <code>Dir_dj</code> //目录信息句柄
输出	<code>* \return</code> 目录项总数
关联模块	<code>file_bs_deal.c</code>
例子	<code>total_dir = fenter_dir_info(fil_bs->file, dir_info);</code> //使用 open 获得的 file，无需重新申请。
补充说明	

5.31. 退出目录

5.31.1. 功能介绍

文件浏览使用，退出目录。

5.31.2. 接口介绍

函数原型	<code>int fexit_dir_info(FILE *file); //退出</code>
功能描述	退出目录
参数说明	<code>file</code> /// 文件句柄
输出	<code>* \return</code> 目录项总数
关联模块	<code>file_bs_deal.c</code>
例子	<code>total_dir = fexit_dir_info(fil_bs->file);//</code>
补充说明	

5.32. 获取目录信息

5.32.1. 功能介绍

文件浏览使用，获取目录信息。

5.32.2. 接口介绍

函数原型	<code>int fget_dir_info(FILE *file, u32 start_num, u32 total_num, void *buf_info);</code>
功能描述	获取目录信息
参数说明	<code>file</code> /// 文件句柄 <code>start_num</code> //起始位置 <code>total_num</code> // 获取目录个数 <code>Buf_info</code> //目录信息句柄
输出	<code>* \return</code> 获取的目录数
关联模块	<code>file_bs_deal.c</code>
例子	<code>real_dir = fget_dir_info(fil_bs->file, start_sn, get_cnt, buf);</code>
补充说明	

5.33. 获取歌词

5.33.1. 功能介绍

文件浏览使用，由歌曲名称获取歌词。

5.33.2. 接口介绍

函数原型	<code>int fget_file_byname_indir(FILE *file, FILE **newFile, void *ext_name);</code> //由歌曲名称获得歌词
功能描述	由歌曲名称获取歌词
参数说明	<code>file</code> /// 歌曲文件句柄 <code>newFile</code> // 歌词文件句柄 <code>ext_name</code> // 后缀名称
输出	<code>*\return</code> 0 找到了, 1 没找到
关联模块	<code>lyrics_api.c</code>
例子	<code>find_lrc_flag = fget_file_byname_indir(file, newFile, ext_name);</code>
补充说明	

5.34. 长文件名获取

5.34.1. 功能介绍

获取长文件名和长文件夹名信息。

5.34.2. 接口介绍

函数原型	<code>int fget_disp_info(FILE *file, void *arg);</code> //用于长文件名获取
功能描述	获取长文件名和长文件夹名信息
参数说明	<code>file</code> /// 歌曲文件句柄 <code>Arg</code> //长文件相关信息结构指针
输出	<code>*\return</code> 0 成功, 非 0 不成功
关联模块	<code>music_action.c</code>
例子	<code>fget_disp_info(file, &__this->fs_info);</code>
补充说明	究

5.35. 创建目录

5.35.1. 功能介绍

创建目录。

5.35.2. 接口介绍

函数原型	int fmk_dir(const char *path, char *folder, u8 mode);
功能描述	创建目录
参数说明	Path //路径 Folder // 文件夹名称 Mode //目录属性
输出	* \return 0 成功, 非 0 不成功
关联模块	
例子	fmk_dir(path, folder, 0);
补充说明	

5.36. 获取录音文件信息

5.36.1. 功能介绍

获取录音文件信息。

5.36.2. 接口介绍

函数原型	int fget_encfolder_info(const char *path, char *folder, char *ext, u32 *last_num, u32 *total_num); //获取录音文件信息
功能描述	获取录音文件信息
参数说明	Path //路径 Folder // 文件夹名称 Ext //文件名后缀 last_num // 可变数字最大数字 total_num //文件总数
输出	* \return 0 成功, 非 0 不成功
关联模块	
例子	ret = fget_encfolder_info(path, folder, ext, &last_num, &total_num);

补充说明	

