



Recorder mix

混合录音接口设计说明书

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	2020-09-01	初稿



目录

1. 文档介绍	4
1.1. 文档目的	4
1.2. 参考文献	4
[1]	4
1.3. 关键词	4
2. 功能概述	4
3. 关键结构体、枚举类型及参数	5
4. 流程框架	6
4.1. 总体架构设计	6
4.2. 数据流流程图	6
5. Recorder mix 详细接口说明	7
int recorder_mix_start(void)	8
void recorder_mix_stop(void)	8
int recorder_mix_get_status(void)	8
6. Recorder_mix 录音回放接口	8
void record_file_close(void)	9
int record_file_play(void)	9
int record_file_play_by_path(char *path)	9
int record_file_get_total_time(void)	10
int record_file_dec_get_cur_time(void)	10

1. 文档介绍

1.1. 文档目的

Recorder mix 混合录音接口为各种情景录音提供 api 接口，可以支持不同音源输入，多种可选编码格式，并支持输出到不同外设，为用户在二次开发提供灵活发挥的空间。

1.2. 参考文献

[1].

1.3. 关键词

缩写、术语	解 释
Recorder mix	混合录音

2. 功能概述

Recorder mix 混合录音模块接口实现以下功能：

- (1) 录音启动/停止
- (2) 支持不同的音乐输入
 - 1) MIC
 - 2) LINEIN
 - 3) FM
 - 4) SBC
 - 5) SCO
 - 6) 其他自定义输入源（待扩展）
- (3) 支持多种编码格式可选
 - 1) ADPCM wav
 - 2) MP3
 - 3) 其他待扩展
- (4) 支持录音输出设备可选
 - 1) SD 卡

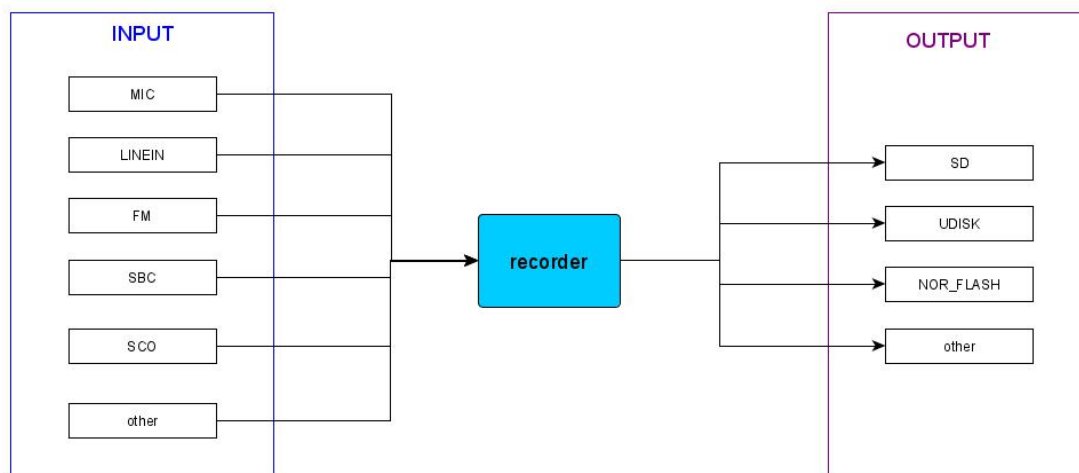
- 2) U 盘
- 3) 内置 flash
- 4) 外挂 flash
- 5) 其他待扩展设备
- (5) 获取录音时间接口

3. 关键结构体、枚举类型及参数

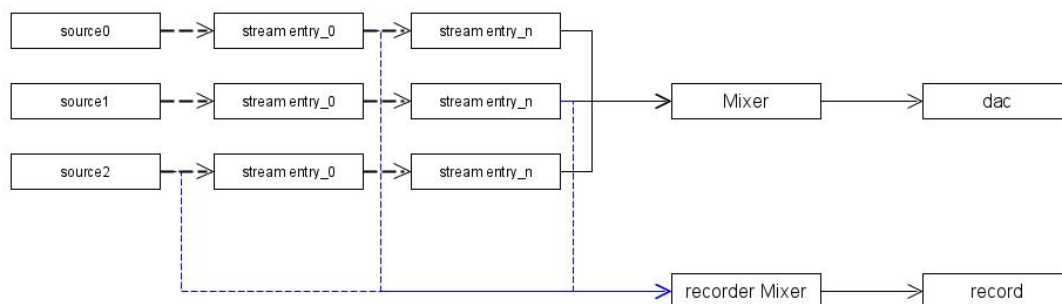
```
struct __recorder_mix {  
    //pcm 0 解码相关结构体  
    struct audio_stream          *audio_stream;// 音频流  
    struct audio_decoder         decoder;  
    struct audio_res_wait       wait;  
    struct pcm_decoder           pcm_dec;// pcm 解码句柄  
    struct audio_mixer_ch       mix_ch;  
    struct audio_mixer_ch       rec_mix_ch;  
    //其他  
    cbuffer_t                   *sco_cbuf;//通话上线数据临时缓存  
    u8                           phone_active;//录音启动时处于通话状态标志  
    u16                          timer;//定时器测试录音时间，调试用  
};
```

4. 流程框架

4.1. 总体架构设计



4.2. 数据流流程图



说明：根据方案开发需求理论上是可以将任何一个解码输出后的任何一个数据流节点混合到 recorder_mix 中，当数据混合后会在 record 数据节点进行录音数据截获，为编码器 encode 提供输入源。如：

```
137 static int pcm_dec_start(struct __recorder_mix *recorder)
138 {
139     int err = 0;
140     if (recorder == NULL) {
141         return -EINVAL;
142     }
143     err = pcm_decoder_open(&recorder->pcm_dec, &decode_task);
144     if (err) {
145         return err;
146     }
147     pcm_decoder_set_event_handler(&recorder->pcm_dec, pcm_dec_event_handler, 0);
148     pcm_decoder_set_read_data(&recorder->pcm_dec, (void *)pcm_fread, recorder);
149     pcm_decoder_set_data_handler(&recorder->pcm_dec, pcm_dec_data_handler);
150
151     audio_mixer_ch_open(&recorder->mix_ch, &mixer);
152     audio_mixer_ch_set_src(&recorder->mix_ch, 1, 0);
153
154     audio_mixer_ch_open(&recorder->rec_mix_ch, &recorder_mixer); → 创建混合通道
155     audio_mixer_ch_set_src(&recorder->rec_mix_ch, 1, 0); → 允许变采样
156
157 // 数据流串联
158 struct audio_stream_entry *entries[8] = {NULL};
159 u8 entry_cnt = 0;
160 entries[entry_cnt++] = &recorder->pcm_dec.decoder.entry;
161 entries[entry_cnt++] = &recorder->mix_ch.entry;
162 recorder->audio_stream = audio_stream_open(recorder, dec_out_stream_resume);
163 audio_stream_add_list(recorder->audio_stream, entries, entry_cnt);
164
165 audio_stream_add_entry(entries[entry_cnt - 2], &recorder->rec_mix_ch.entry);
166 err = audio_decoder_start(&recorder->pcm_dec.decoder); → 将数据流输出到recorder_mix
167 if (err == 0) {
168     printf("pcm_dec_start ok\n");
169 }
170 return err;
171 }
172 }
```

更多例子可以参考 FM(audio_dec_fm.c)、LINEIN(audio_dec_linein.c)等混合录音输出处理

5. Recorder mix 详细接口说明

```
/*-----*/
```

/**@brief 混合录音开始

@param

@return 0 成功, 非 0 失败

@note

混合录音支持录制内容:

BT sbc (蓝牙高级音频)

BT sco (蓝牙通话, 通话接通后才可以启动)

FM (内置 FM)

Linein(外部音源输入,单声道, 696 系列需要外部硬件合并左右声道)

录音参数配置:

请在 __recorder_mix_start 函数内部修改参数

1、支持设备选择, 如: sd0、udisk0 等

2、修改文件名称及文件夹名称, 默认文件夹名称为 JL_REC, 文件名 AC69****

3、编码格式(资源受限, 通话支持 adpcm wav)

4、支持砍头砍尾处理

注意:

1、通话录音录制的是 adpcm wav 格式, 其他音源录音为 mp3

2、通话录音不支持录制来电铃声

3、普通录制过程, 来电或去电会先停止原来的录音, 通话录音需要重新启动

4、通话结束，如是通话录音， 录音会主动停止

*/

/*-----*/

int recorder_mix_start(void)

/*-----*/

/**@brief 混合录音停止

@param

@return

@note

*/

/*-----*/

void recorder_mix_stop(void)

/*-----*/

/**@brief 获取混合录音状态

@param

@return

1:正在录音状态

0:录音停止状态

@note

*/

/*-----*/

int recorder_mix_get_status(void)

6. Recorder_mix 录音回放接口

注意：

AC696N SDK 以下接口不可以在蓝牙、FM 模式即时回放， 因为 FM 解码及蓝牙解码与录音回放解码所使用的空间是 overlay（复用）的，录音回放建议单独模式实现，可以在现有的 record 模式实现（根据需求修改流程），具体实现参考 record 模式录音回放 demo。对于一般设备例如 sd/udisk 设备录音，也可以在音乐模式中当成普通音乐文件播放，也可以通过使能录音区分功能，一键切换到录音设备进行播放，具体操作参考【music_player 接口设计说明文档.pdf】文档中的 music_player_play_record_folder。


```
/*-----*/
```

```
/**@brief 关闭录音文件播放
```

```
@param
```

```
@return
```

```
@note
```

```
*/
```

```
/*-----*/
```

```
void record_file_close(void)
```

```
/*-----*/
```

```
/**@brief 录音文件播放
```

```
@param
```

```
@return 0:成功
```

```
@note
```

```
*/
```

```
/*-----*/
```

```
int record_file_play(void)
```

```
/*-----*/
```

```
/**@brief 按照路径播放录音文件
```

```
@param
```

```
@return 0:成功
```

```
@note 针对特殊需求， 用户可以直接指定录音路径播放,盘符在路径中直接指定
```

```
*/
```

```
/*-----*/
```

```
int record_file_play_by_path(char *path)
```

```
/*-----*/
```

```
/**@brief 获取录音播放总事件
```

```
@param
```

```
@return 总时间
```

```
@note
```

```
*/
```

```
/*-----*/
```

int record_file_get_total_time(void)

```
/*-----*/  
/**@brief    获取录音播放当前时间  
    @param  
    @return   当前时间  
    @note  
*/  
/*-----*/
```

int record_file_dec_get_cur_time(void)