



# Record 应用 详细设计说明书

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	2020/08/11	初稿



## 目录

1. 引 言.....	4
1.1. 编写目的.....	4
1.2. 参考资料.....	4
1.3. 术语和缩写词.....	4
2. 总体设计.....	4
2.1. 需求概述.....	4
2.2. 总体架构设计.....	5
2.3. 详细流程框图.....	6
2.4. 关键数据结构说明.....	6
3. 应用系统事件.....	7
3.1. 事件分类.....	7
3.2. 按键事件处理.....	7
3.3. 设备事件处理.....	7
4. 录音文件存放.....	7
5. 详细接口注释.....	8
static void record_mic_start(void).....	8
static void record_mic_stop(void).....	8
static void record_key_pp().....	9
static void record_tone_play_end_callback(void *priv, int flag).....	9
static int record_key_event_opr(struct sys_event *event).....	9
static void record_task_start(void).....	9
static void record_task_close().....	10
static int record_sys_event_handler(struct sys_event *event).....	10
int record_app_check(void).....	10
void app_record_task().....	10

## 1. 引言

### 1.1. 编写目的

该文档为基于 AC696N soundbox 平台开发 Record 应用的人员提供相应的设计开发文档。也可以为测试 Record 应用的测试人员提供参考。

文档中详细定义了 Record 应用的总体功能、系统的接口和数据属性；对程序的基本结构、功能模块以及各个程序的名称进行了划分，以便于 Record 应用的详细设计和二次开发。

### 1.2. 参考资料

[1]

### 1.3. 术语和缩写词

缩写和术语	解 释
REC_FOLDER_NAME	录音文件夹名称配置，在 app_config.h 中定义
AC69****	录音文件名定义，“****”每录制一个录音文件底层底层会自动排序（如：AC690000、AC690001、AC690002...），如果是完整名称，底层按照完整名称生成录音文件
录音区分	正常音乐播放过程跳过指定录音文件夹内容只播放音乐文件，并可以通过一键调制录音文件夹播放

## 2. 总体设计

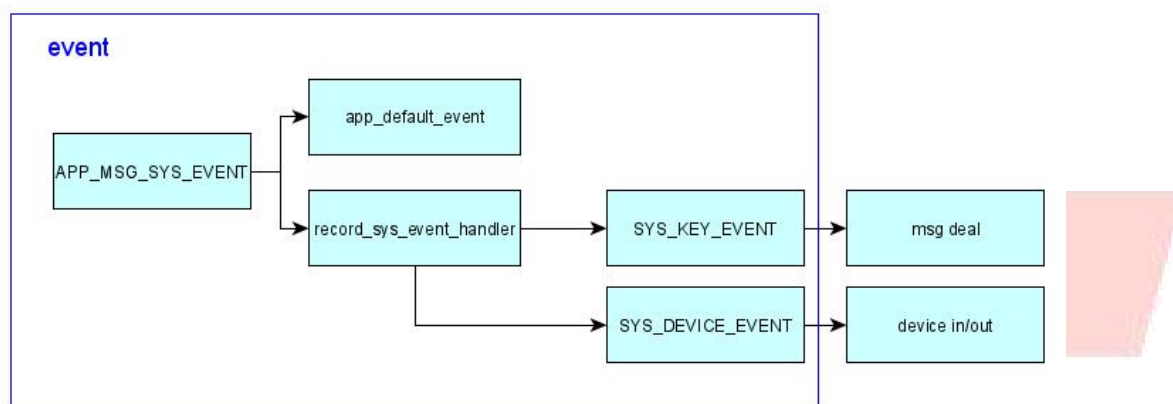
### 2.1. 需求概述

本应用主要是基于 AC696N soundbox 的 SDK 系统开发包来实现 mic 录音及录音回放功能。

Record 应用主要实现的功能包括：

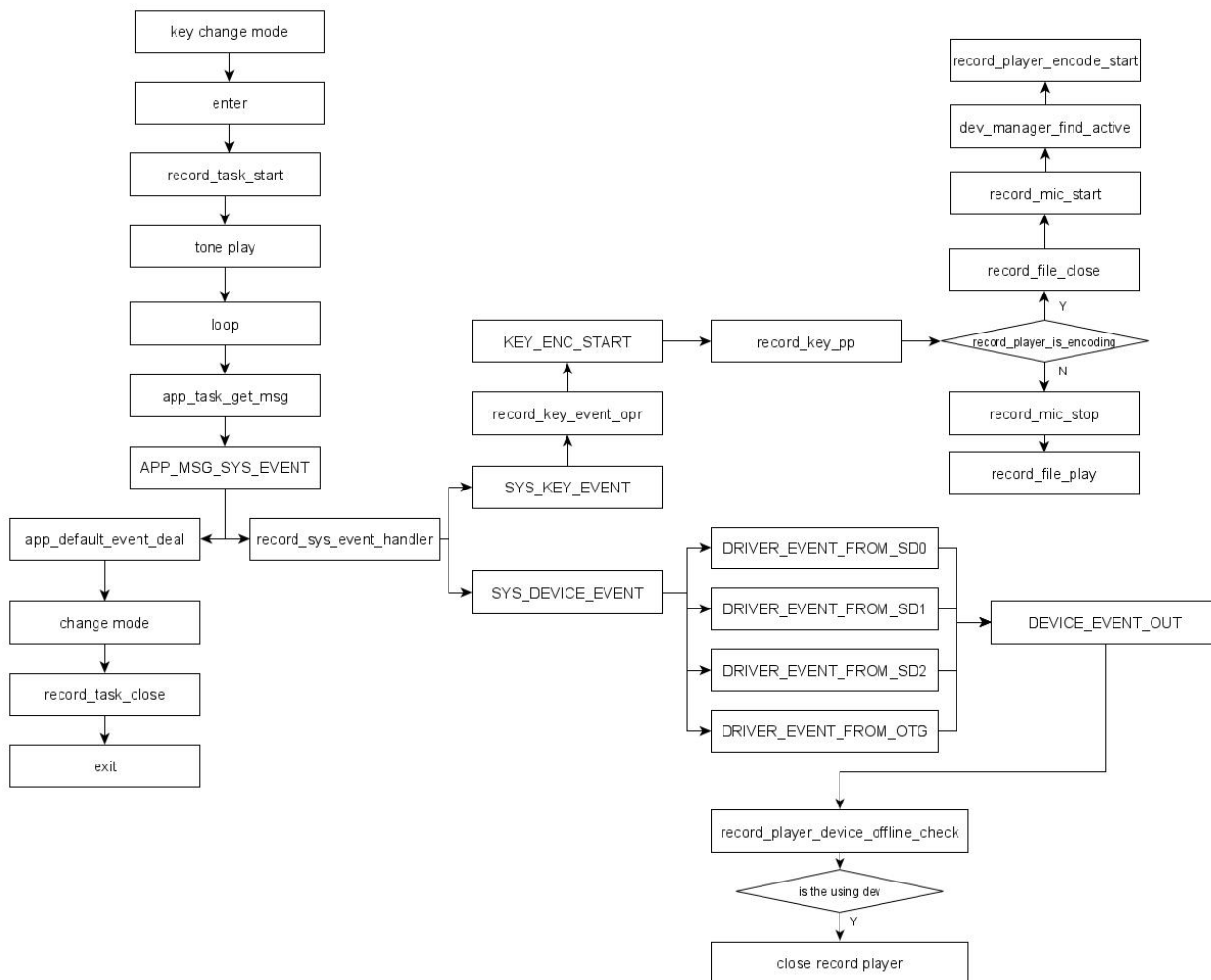
- (1) 实现 MP3、WAV 等编码格式录音
- (2) 支持 sd、udisk、nor\_fs（外挂 flash）等设备录音
- (3) 提供录音启动、回放
- (4) 录音文件录制时，支持砍掉头部杂音、尾部杂音、限制最小录音文件大小配置

## 2.2. 总体架构设计



模块名称	功能简述	对应文件
event	处理系统产生的各种事件	record.c
msg_deal	按键消息处理（record_sys_event_handler） 负责调用响应一些按键消息例如启动录音、停止录音、录音回放、录音回放停止操作	record.c
device in/out	设备上下线处理	record.c

## 2.3. 详细流程框图



## 2.4. 关键数据结构说明

```

struct record_file_fmt {
    u8 gain;//增益配置
    u8 channel;//声道数, 1:单声道, 2: 双声道
    u16 sample_rate;//采样率
    u32 bit_rate;//码率
    u32 coding_type;//编码格式
    char *dev;//录音设备盘符
    char *folder;//录音文件夹
    char *filename;//录音文件名
    u32 cut_head_time;//录音文件去头时间,单位 ms
    u32 cut_tail_time;//录音文件去尾时间,单位 ms
    u32 limit_size;//录音文件大小最小限制, 单位 byte

```

版权所有，侵权必究

```
u8 source;//录音输入源
};
```

录音参数配置请参考接口 `record_mic_start` 接口内部实现。

## 3. 应用系统事件

### 3.1. 事件分类

系统事件分类：

- (1) 按键事件 (`SYS_KEY_EVENT`)  
主要是各种按键触发的事件及软件流程 `post` 的事件
- (2) 设备事件 (`SYS_DEVICE_EVENT`)  
主要是各种设备上下线事件，本应用主要响应以下事件 (`in/out`)：  
`DRIVER_EVENT_FROM_SD0`  
`DRIVER_EVENT_FROM_SD1`  
`DRIVER_EVENT_FROM_SD2`  
`DEVICE_EVENT_FROM_OTG`

以应用情景对事件分类：

- (1) 应用模式内部事件 (`record_sys_event_handler`)
- (2) 公共事件 (`app_default_event_deal`)  
公共事件同样会响应一些例如公共按键（模式切换/音量加减等）、设备上下线（`linein/pc` 上下线等）、蓝牙事件（连接、通话、播放等）

### 3.2. 按键事件处理

参考 2.3 详细流程框图。

### 3.3. 设备事件处理

参考 2.3 详细流程框图

## 4. 录音文件存放

录音生成的文件将会放在设备根目录的 `REC_FOLDER_NAME` 下（`app_config.h` 中定义），录音回放将在这个文件夹中寻找最后录制的文件进行回放。

## 5. 录音区分

说明：

录音文件夹播放实现方式，本 SDK 是将录音文件夹当成一个等同 sd0/udisk0 的设备 sd0\_rec/udisk0\_rec，实现文件夹切换实际上是切换不同的设备（如：切换到录音文件夹播放 sd0-->sd0\_rec，切换到正常音乐播放 sd0\_rec-->sd0）

如要播放完整的录音文件夹内容可以切换到音乐模式进行播放，如果需要在支持**录音区分**播放功能可以按照以下步骤实现：

- (1) 在对应板卡使能 TCFG\_RECORD\_FOLDER\_DEV\_ENABLE
- (2) 切换到音乐模式
- (3) 调用接口 music\_player\_play\_record\_folder 便可以实现录音文件夹与非录音文件夹之间的切换

## 6. 详细接口注释

```
/*-----*/  
/**@brief    mic 录音启动  
    @param    无  
    @return  
    @note  
*/  
/*-----*/
```

static void record\_mic\_start(void)

```
/*-----*/  
/**@brief    mic 录音停止  
    @param    无  
    @return  
    @note  
*/  
/*-----*/
```

static void record\_mic\_stop(void)

```
/*-----*/  
/**@brief    mic 录音与回放切换
```

版权所有，侵权必究



```
@param    无
@return
@note
*/
/*-----*/

static void record_key_pp()

/*-----*/
/**@brief    录音模式提示音结束处理
    @param    无
    @return
    @note
*/
/*-----*/

static void  record_tone_play_end_callback(void *priv, int flag)

/*-----*/
/**@brief    record 按键消息入口
    @param    无
    @return    1、消息已经处理，不需要发送到 common  0、消息发送到 common 处理
    @note
*/
/*-----*/

static int record_key_event_opr(struct sys_event *event)

/*-----*/
/**@brief    record 模式初始化
    @param    无
    @return
    @note
*/
/*-----*/

static void record_task_start(void)

/*-----*/
/**@brief    record 退出
    @param    无
```

版权所有，侵权必究

9

```
@return
@note
*/
/*-----*/

static void record_task_close()

/*-----*/
/**@brief    record 模式活跃状态 所有消息入口
    @param    无
    @return   1、当前消息已经处理，不需要发送 comomon
             0、当前消息不是 linein 处理的，发送到 common 统一处理
    @note
*/
/*-----*/

static int record_sys_event_handler(struct sys_event *event)

/*-----*/
/**@brief    record 在线检测 切换模式判断使用
    @param    无
    @return   1 设备在线 0 设备不在线
    @note
*/
/*-----*/

int record_app_check(void)

/*-----*/
/**@brief    record 主任务
    @param    无
    @return   无
    @note
*/
/*-----*/

void app_record_task()
```