

录音接口使用说明

iFLYTEK CO., LTD.

本手册内容若有变动，恕不另行通知。本手册例子中所用的公司、人名和数据若非特别声明，均属虚构。未得到科大讯飞股份有限公司明确的书面许可，不得以任何目的、以任何形式或手段（电子的或机械的）复制或传播手册的任何部分。

本文档可能涉及科大讯飞股份有限公司的专利（或正在申请的专利）、商标、版权或其他知识产权，除非得到科大讯飞股份有限公司的明确书面许可协议，本文档不授予使用这些专利（或正在申请的专利）、商标、版权或其他知识产权的任何许可协议。

本手册提及的其它产品和公司名称均可能是各自所有者的商标。

本软件产品受最终用户许可协议（EULA）中所述条款和条件的约束，该协议位于产品文档和 / 或软件产品的联机文档中，如果您使用本产品，表明您已阅读并接受了 EULA 的条款。

版权所有 © 科大讯飞股份有限公司
Copyright © iFLYTEK CO., LTD.

目 录

第 1 章 概述.....	3
1.1 文档说明.....	3
第 2 章 函数说明.....	5
2.1 CREATE_RECORDER.....	5
2.2 DESTROY_RECORDER.....	6
2.3 OPEN_RECORDER.....	6
2.4 CLOSE_RECORDER.....	7
2.5 START_RECORD.....	7
2.6 STOP_RECORD.....	8
2.7 IS_RECORD_STOPPED.....	8
2.8 GET_DEFAULT_INPUT_DEV.....	9
2.9 get_input_dev_num.....	9

第1章 概述

1.1文档说明

该文档描述 winrec.h 或者 linuxrec.h 中的录音接口。接口由 C 语言编写，windows 下使用 waveIn API （ [https://msdn.microsoft.com/en-us/library/windows/desktop/dd743832\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd743832(v=vs.85).aspx) ），linux 下使用 alsa-lib API

（<http://www.alsa-project.org/alsa-doc/alsa-lib/>），该代码仅作示例使用，源码形式提供，可以直接使用、改写。若因代码中存在的缺陷对使用者产生损失，我们概不承担责任。

本文简要说明录音接口的使用方法。

文档中使用的符号约定：

符号	含义
[in]	表明该参数是调用时赋值的参数——输入参数

[out]	该参数在函数返回时被赋值——输出参数
[in/out]	该参数在函数调用时作为输入、函数返回时作为输出参数

使用该接口时，一般的函数调用顺序如下

<code>create_recorder</code>	创建录音对象
<code>open_recorder</code>	打开录音机，配置录音格式
<code>start_record</code>	开始录音，录音数据在注册的回调函数中获取
<code>stop_record</code>	停止录音
<code>close_recorder</code>	关闭录音机，对应 <code>open_recorder</code>
<code>destroy_recorder</code>	销毁录音对象，对应 <code>create_recorder</code>

`open_recorder` 后只要没有 `close_recorder`，可以多次 `start_record` 和 `stop_record`。

第2章 函数说明

2.1 create_recorder

□ 函数原型

```
int create_recorder(struct recorder ** out_rec,  
                    void (*on_data_ind)(char *data,  
                    unsigned long len,  
                    void *user_para),  
                    void* user_cb_para);
```

□ 功能

创建一个录音机对象。

□ 参数

□ out_rec[out]

录音机对象指针，若函数执行成功则指向创建好的录音机对象。

□ on_data_ind[in]

回调函数。驱动提供音频数据时，回调该函数。

回调函数原型： void (*on_data_ind) (char *data, unsigned long len,
void *user_para);

返回：

void

参数：

data[in] : 音频数据地址。

len[in] : 数据长度。

user_para : 在 create_recorder 时传入的 user_cb_para 用户参数。

□ 返回值

如果函数调用成功返回 0，否则返回错误代码。



□ 说明

创建成功后的对象在不需再使用后，使用destroy_recorder销毁。

2.2 destroy_recorder

□ 函数原型

```
void destroy_recorder(struct recorder *rec);
```

□ 功能

销毁录音机对象。

□ 参数

□ **rec[in]**

录音机对象

□ 返回值

如果函数调用成功返回 0，否则返回错误代码。

□ 说明

如果已经 open_recorder 成功了，在 destroy_recorder 之前必须先 close_recorder

2.3 open_recorder

□ 函数原型

```
int open_recorder(struct recorder * rec, unsigned int dev,  
                  WAVEFORMATEX * fmt);
```

□ 功能

配置录音设备和录音格式，并打开录音机。

□ 参数

□ **rec[in]**

录音机对象。

□ **dev[in]**

录音设备，windows下用整形标示，从0开始。可以使用get_default_input_dev



获取默认的录音设备

□ **fmt[in]**

录音格式，WAVEFORMATEX 结构详见代码或者微软的 MSDN。

□ **返回值**

如果函数调用成功返回 0，否则返回错误代码。

2.4 close_recorder

□ **函数原型**

```
void close_recorder(struct recorder *rec);
```

□ **功能**

关闭录音机。

□ **参数**

□ **rec[in]**

录音机对象

□ **返回值**

如果函数调用成功返回 0，否则返回错误代码。

2.5 start_record

□ **函数原型**

```
int start_record(struct recorder * rec);
```

□ **功能**

开始录音。

□ **参数**

□ **rec[in]**

录音机对象

□ **返回值**

如果函数调用成功返回 0，否则返回错误代码。

2.6 stop_record

- 函数原型

```
int stop_record(struct recorder * rec);
```

- 功能

停止录音。

- 参数

- **rec[in]**

录音机对象

- 返回值

如果函数调用成功返回 0，否则返回错误代码。

- 说明

如果已经停止，调用直接返回 0。 Windows 下注意该函数执行返回后，回调函数可能还是会被调用到，因为此时驱动还需要弹出内核层的 buffer，具体参见 MSDN 中 WaveInStop 和 WaveInReset 函数。

2.7 is_record_stopped

- 函数原型

```
int is_record_stopped(struct recorder *rec);
```

- 功能

查询录音是否已经彻底停止掉。若返回 1 则表明回调函数不会再被调用。如果需要，返回 1 后就可以安全的去做清理和关闭录音机的动作。

- 参数

- **rec[in]**

录音机对象

- 返回值

已停止，返回 1，否则返回 0。

2.8 get_default_input_dev

- **函数原型**

`int get_default_input_dev();`

- **功能**

获取默认的设备 ID。

- **返回值**

返回默认设备 ID。

2.9 get_input_dev_num

- **函数原型**

`unsigned int get_input_dev_num();`

- **功能**

获取录音设备的总数。

- **返回值**

录音设备的总数。