

# Rockchip Rkfacial 使用说明

---

文件标识: RK-SM-YF-363

发布版本: V2.0.0

日期: 2020-07-28

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

## 免责声明

本文档按“现状”提供, 瑞芯微电子股份有限公司(“本公司”, 下同)不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因, 本文档将可能在未经任何通知的情况下, 不定期进行更新或修改。

## 商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标, 归本公司所有。

本文档可能提及的其他所有注册商标或商标, 由其各自所有者所有。

版权所有 © 2020 瑞芯微电子股份有限公司

超越合理使用范畴, 非经本公司书面许可, 任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部, 并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

网址: [www.rock-chips.com](http://www.rock-chips.com)

客户服务电话: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: [fae@rock-chips.com](mailto:fae@rock-chips.com)

## 前言

## 概述

本文描述rkfacil各个模块的接口说明。

## 产品版本

平台名称	内核版本
Linux	4.4

## 读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

## 修订记录

日期	版本	作者	修改说明
2020-05-21	V1.0.0	Zhihua Wang	初始版本
2020-06-08	V1.1.0	Zhihua Wang	修改用户信息回调
2020-07-28	V2.0.0	Zhihua Wang	更新接口和流程图

# 目录

## Rockchip Rkfacial 使用说明

### 1. 代码模块说明

#### 1.1 rockface\_control

- 1.1.1 int rockface\_control\_init(int face\_cnt)
- 1.1.2 void rockface\_control\_exit(void)
- 1.1.3 int rockface\_control\_add\_ui(int id, const char \*name, void \*feature)
- 1.1.4 int rockface\_control\_add\_web(int id, const char \*name)
- 1.1.5 int rockface\_control\_add\_local(const char \*name)
- 1.1.6 int rockface\_control\_delete(int id, const char \*pname, bool notify)

#### 1.2 database

- 1.2.1 int database\_init(void)
- 1.2.2 void database\_exit(void)
- 1.2.3 void database\_bak(void)
- 1.2.4 int database\_insert(void \*data, size\_t size, char \*name, size\_t n\_size, bool sync\_flag)
- 1.2.5 int database\_record\_count(void)
- 1.2.6 int database\_get\_data(void \*dst, const int cnt, size\_t d\_size, size\_t d\_off, size\_t n\_size, size\_t n\_off)
- 1.2.7 bool database\_is\_name\_exist(char \*name)
- 1.2.8 bool database\_is\_id\_exist(int id, char \*name, size\_t size)
- 1.2.9 int database\_get\_user\_name\_id(void)
- 1.2.10 void database\_delete(char \*name, bool sync\_flag)

#### 1.3 db\_monitor

- 1.3.1 void db\_monitor\_init()
- 1.3.2 void db\_monitor\_face\_list\_add(int id, char \*path, char \*name, char \*type)
- 1.3.3 void db\_monitor\_face\_list\_delete(int id)
- 1.3.4 void db\_monitor\_snapshot\_record\_set(char \*path)
- 1.3.5 void db\_monitor\_control\_record\_set(int face\_id, char \*path, char \*status, char \*similarity)
- 1.3.6 void db\_monitor\_get\_user\_info(struct user\_info \*info, int id)

#### 1.4 display

- 1.4.1 int display\_init(int width, int height)
- 1.4.2 void display\_exit(void)
- 1.4.3 void display\_switch(enum display\_video\_type type)

#### 1.5 camrgb\_control

- 1.5.1 int camrgb\_control\_init(void)
- 1.5.2 void camrgb\_control\_exit(void)
- 1.5.3 void camrgb\_control\_expo\_weights\_270(int left, int top, int right, int bottom)
- 1.5.4 void camrgb\_control\_expo\_weights\_90(int left, int top, int right, int bottom)
- 1.5.5 void camrgb\_control\_expo\_weights\_default(void)
- 1.5.6 void set\_rgb\_display(display\_callback cb)
- 1.5.7 void set\_rgb\_rotation(int angle)

#### 1.6 camir\_control

- 1.6.1 int camir\_control\_init(void)
- 1.6.2 void camir\_control\_exit(void)
- 1.6.3 bool camir\_control\_run(void)
- 1.6.4 void set\_ir\_display(display\_callback cb)
- 1.6.5 void set\_ir\_rotation(int angle)

#### 1.7 shadow\_display

- 1.7.1 void shadow\_display(void \*src\_ptr, int src\_fd, int src\_fmt, int src\_w, int src\_h)
- 1.7.2 void shadow\_display\_vertical(void \*src\_ptr, int src\_fd, int src\_fmt, int src\_w, int src\_h)
- 1.7.3 void shadow\_paint\_box(int left, int top, int right, int bottom)
- 1.7.4 void shadow\_paint\_info(struct user\_info \*info, bool real)
- 1.7.5 void shadow\_get\_crop\_screen(int \*width, int \*height)

#### 1.8 load\_feature

- 1.8.1 int count\_file(const char \*path, char \*fmt)
- 1.8.2 int load\_feature(const char \*path, char \*fmt, void \*data, unsigned int cnt)

#### 1.9 play\_wav

- 1.9.1 int play\_wav\_thread\_init(void)
- 1.9.2 void play\_wav\_thread\_exit(void)
- 1.9.3 void play\_wav\_signal(char \*name)
- 1.10 rga\_control
  - 1.10.1 int rga\_control\_buffer\_init(bo\_t \*bo, int \*buf\_fd, int width, int height, int bpp)
  - 1.10.2 void rga\_control\_buffer\_deinit(bo\_t \*bo, int buf\_fd)
- 1.11 rkfacial
  - 1.11.1 typedef void (\*display\_callback)(void \*ptr, int fd, int fmt, int w, int h, int rotation)
  - 1.11.2 void set\_rgb\_param(int width, int height, display\_callback cb, bool expo)
  - 1.11.3 void set\_ir\_param(int width, int height, display\_callback cb)
  - 1.11.4 void set\_usb\_param(int width, int height, display\_callback cb)
  - 1.11.5 void set\_face\_param(int width, int height, int cnt)
  - 1.11.6 int rkfacial\_init(void)
  - 1.11.7 void rkfacial\_exit(void)
  - 1.11.8 void rkfacial\_register(void)
  - 1.11.9 void rkfacial\_delete(void)
  - 1.11.10 void register\_rkfacial\_paint\_box(rkfacial\_paint\_box\_callback cb)
  - 1.11.11 void register\_rkfacial\_paint\_info(rkfacial\_paint\_info\_callback cb)
- 1.12 snapshot
  - 1.12.1 int snapshot\_init(struct snapshot \*s, int w, int h)
  - 1.12.2 void snapshot\_exit(struct snapshot \*s)
  - 1.12.3 int snapshot\_run(struct snapshot \*s, rockface\_image\_t \*image, rockface\_det\_t \*face, RgaSURF\_FORMAT fmt, long int sec, char mark)
- 1.13 turbojpeg\_decode
  - 1.13.1 void \*turbojpeg\_decode\_get(const char \*name, int \*w, int \*h, int \*b)
  - 1.13.2 void turbojpeg\_decode\_put(void \*data)
- 1.14 usb\_camera
  - 1.14.1 int usb\_camera\_init(void)
  - 1.14.2 void usb\_camera\_exit(void)
  - 1.14.3 void set\_usb\_display(display\_callback cb)
  - 1.14.4 void set\_usb\_rotation(int angle)
- 1.15 vpu decode(MJPEG decode)
  - 1.15.1 int vpu\_decode\_jpeg\_init(struct vpu\_decode\* decode, int width, int height)
  - 1.15.2 int vpu\_decode\_jpeg\_doing(struct vpu\_decode\* decode, void\* in\_data, RK\_S32 in\_size, int out\_fd, void\* out\_data)
  - 1.15.3 int vpu\_decode\_jpeg\_done(struct vpu\_decode\* decode)
- 1.16 vpu encode(MJPEG encode)
  - 1.16.1 int vpu\_encode\_jpeg\_init(struct vpu\_encode\* encode, int width, int height, int quant, MppFrameFormat format)
  - 1.16.2 int vpu\_encode\_jpeg\_doing(struct vpu\_encode\* encode, void\* srcbuf, int src\_fd, size\_t src\_size, void \*dst\_buf, int dst\_fd, size\_t dst\_size)
  - 1.16.3 void vpu\_encode\_jpeg\_done(struct vpu\_encode\* encode)

# 1. 代码模块说明

---

## 1.1 rockface\_control

### 1.1.1 int rockface\_control\_init(int face\_cnt)

#### 说明

完成rockface各个算法的初始化，完成人脸数据库的初始化，并从指定目录的jpg文件提取人脸特征值到数据库。

#### 参数

face\_cnt 人脸数据库最大支持的人脸数量

#### 返回

int 0成功，-1失败

### 1.1.2 void rockface\_control\_exit(void)

#### 说明

完成rockface各个算法的反初始化。

#### 参数

void

#### 返回

void

### 1.1.3 int rockface\_control\_add\_ui(int id, const char \*name, void \*feature)

#### 说明

通过UI注册用户

#### 参数

id 用户id

name 用户名

feature 用户人脸特征值

#### 返回

int 0成功

### 1.1.4 int rockface\_control\_add\_web(int id, const char \*name)

说明

通过web server注册用户

参数

name 用户名

返回

int 0成功

### 1.1.5 int rockface\_control\_add\_local(const char \*name)

说明

通过本机存储图片注册用户

参数

name 用户名

返回

int 0成功

### 1.1.6 int rockface\_control\_delete(int id, const char \*pname, bool notify)

说明

删除用户

参数

id 用户id

pname 用户名，web server删除使用

notify 是否通知web server

返回

int 0成功

默认宏定义说明：

```
1  #define DEFAULT_FACE_NUMBER 1000 表示默认人脸数据库最大支持人脸数量
2  #define DEFAULT_FACE_PATH "/userdata" 开机默认从这个目录加载jpg文件获取特征值
3  #define FACE_DETECT_SCORE 0.55 人脸检测的分数，范围0-1，越大越严格
4  #define FACE_SCORE_LANDMARK_RUNNING 0.9 RGB预览人脸特征值的分数，范围0-1，越大越严格
5  #define FACE_SCORE_LANDMARK_IMAGE 0.5 RGB照片人脸特征值的分数，范围0-1，越大越严格
6  #define FACE_SIMILARITY_CONVERT(f) powf(2.0, -((f))) RGB人脸识别的分数转化公式
7  #define FACE_SIMILARITY_SCORE 1.0 RGB人脸识别的分数，范围建议0.7-1.3，越小越严格
8  #define FACE_SCORE_REGISTER 0.99 人脸注册的人脸分数，范围0-1，越大越严格
9  #define FACE_REGISTER_CNT 5 人脸注册时连续读到的多少次人脸特征值均在数据库里面，提示已经注册
10 #define FACE_REAL_SCORE 0.5 活体检测分数最小要求，范围0-1，越大越严格
```

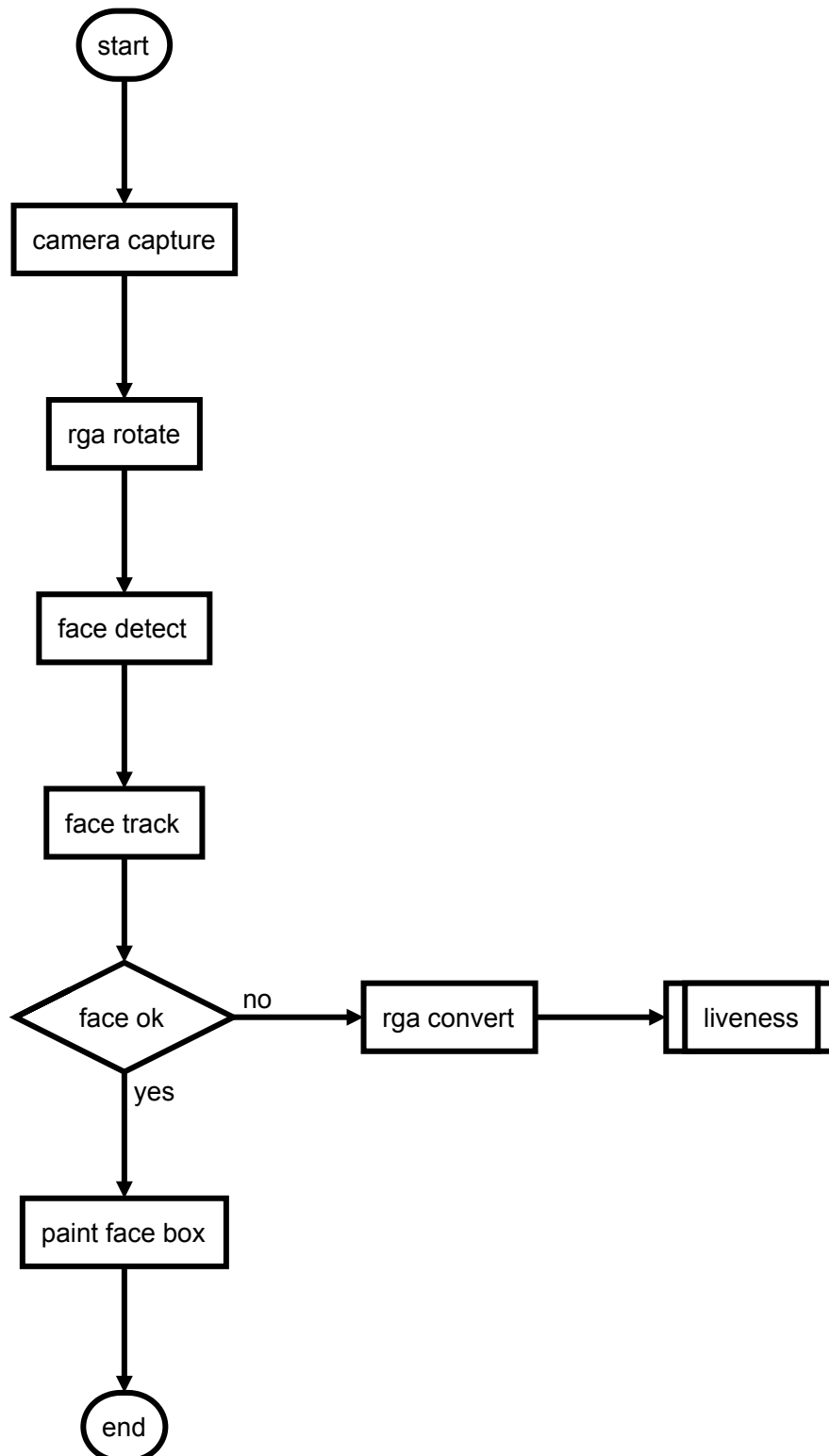
```

11 #define LICENCE_PATH PRE_PATH "/key.lic" rockface人脸授权key存放路径
12 #define BAK_LICENCE_PATH BAK_PATH "/key.lic" rockface人脸授权备份key存放路径
13 #define FACE_DATA_PATH "/usr/lib" rockface data存放路径
14 #define MIN_FACE_WIDTH(w) ((w) / 5) 人脸检测、特征值提取人脸最小像素要求
15 #define FACE_TRACK_FRAME 0 人脸跟踪最大跟踪时间(帧)
16 #define FACE_RETRACK_TIME 1 人脸跟踪再次跟踪时间(秒)
17 #define SNAP_TIME 3 抓拍最低间隔时间(秒)

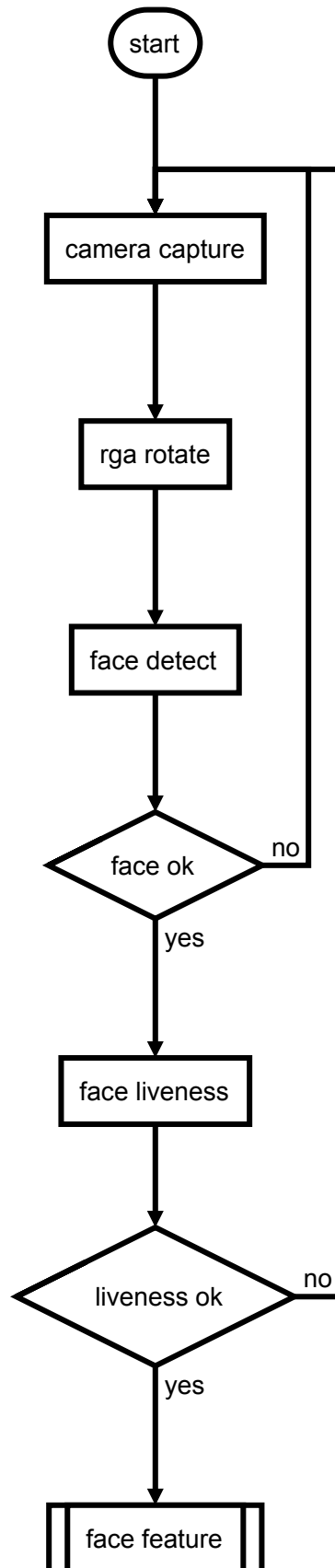
```

## 应用流程图

- RGB人脸检测

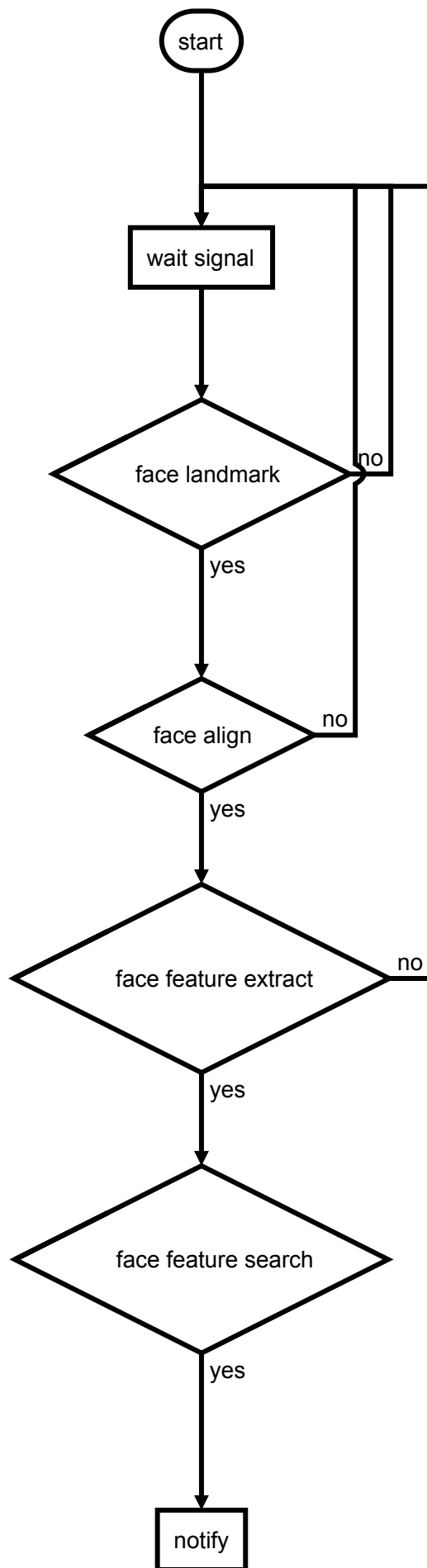


- IR人脸检测、活体检测



- RGB人脸特征值识别





## 1.2 database

### 1.2.1 int database\_init(void)

说明

完成数据库的初始化

参数

void

返回

int 0成功，-1失败

### 1.2.2 void database\_exit(void)

说明

完成数据库的反初始化

参数

void

返回

void

### 1.2.3 void database\_bak(void)

说明

完成数据库的备份

参数

void

返回

void

### 1.2.4 int database\_insert(void \*data, size\_t size, char \*name, size\_t n\_size, bool sync\_flag)

说明

完成插入一条数据到数据库

参数

data 特征值数据地址

size 特征值大小

name 用户名

n\_size 用户名大小

sync\_flag 为true时会实时sync保存数据库

返回

int 0成功，-1失败

### 1.2.5 int database\_record\_count(void)

说明

获取记录的人脸特征值数量

参数

void

返回

int 记录的人脸特征值数量

### 1.2.6 int database\_get\_data(void \*dst, const int cnt, size\_t d\_size, size\_t d\_off, size\_t n\_size, size\_t n\_off)

说明

把数据库最多cnt个特征值提取出来保存到dst，dst提供给rockface做特征值数据库搜索匹配

参数

dst 存储数据的指针

cnt 最多可以提取特征值的数量

d\_size 特征值大小

d\_off 特征值在用户数据结构体的偏移

n\_size 名字的大小

n\_off 名字在用户数据结构体的偏移

返回

int 获取到的特征值的数量

### 1.2.7 bool database\_is\_name\_exist(char \*name)

说明

判断用户名是否已经存在于数据库

参数

name 用户名

返回

bool true存在，false不存在

### **1.2.8 bool database\_is\_id\_exist(int id, char \*name, size\_t size)**

说明

判断用户id是否已经存在于数据库

参数

id 用户id

name 用户名存储地址

size 用户名存储大小

返回

bool true存在，false不存在

### **1.2.9 int database\_get\_user\_name\_id(void)**

说明

用于用户实时注册时可以使用的id号，id号从0开始，找到未使用的即可以给用户注册。

参数

void

返回

int 可以使用的id号

### **1.2.10 void database\_delete(char \*name, bool sync\_flag)**

说明

通过用户名删除数据库里面的一个记录

参数

name 用户名

sync\_flag 为true时会实时sync保存数据库

返回

void

## **1.3 db\_monitor**

### **1.3.1 void db\_monitor\_init()**

说明

db\_monitor初始化，用于接收web server相关消息

参数

void

返回

void

### **1.3.2 void db\_monitor\_face\_list\_add(int id, char \*path, char \*name, char \*type)**

说明

发送添加用户消息到web server

参数

id 用户id

path 用户路径

name 用户名

type 用户类型

返回

void

### **1.3.3 void db\_monitor\_face\_list\_delete(int id)**

说明

发送删除用户消息到web server

参数

id 用户id

返回

void

### **1.3.4 void db\_monitor\_snapshot\_record\_set(char \*path)**

说明

发送抓拍消息到web server

参数

path 抓拍路径

返回

void

### **1.3.5 void db\_monitor\_control\_record\_set(int face\_id, char \*path, char \*status, char \*similarity)**

说明

发送控制消息到web server

参数

face\_id 用户id

path 抓拍路径

status 开关状态

similarity 相似度

返回

void

### **1.3.6 void db\_monitor\_get\_user\_info(struct user\_info \*info, int id)**

说明

通过web server获取用户信息

参数

info 存储用户信息

id 用户id

返回

void

## **1.4 display**

### **1.4.1 int display\_init(int width, int height)**

说明

双层vop视频显示层初始化

参数

width 屏幕宽

height 屏幕高

返回

void

### **1.4.2 void display\_exit(void)**

说明

双层vop视频显示层反初始化

参数

void

返回

void

### **1.4.3 void display\_switch(enum display\_video\_type type)**

说明

切换IR/RGB/USB摄像头显示

参数

type 需要显示的摄像头类型

返回

void

## **1.5 camrgb\_control**

### **1.5.1 int camrgb\_control\_init(void)**

说明

实现了rgb video的初始化，在process线程进行图像数据的读取、旋转、显示、送数据给rockface做人脸识别、检测、特征值提取、轨迹跟踪处理。

参数

void

返回

int 0成功， -1失败

### **1.5.2 void camrgb\_control\_exit(void)**

说明

实现了rgb video的反初始化。

参数

void

返回

void

### **1.5.3 void camrgb\_control\_expo\_weights\_270(int left, int top, int right, int bottom)**

说明

用于实现rgb图像顺时针旋转270度局部曝光，可以实现人脸坐标在暗处局部曝光。

参数

left 人脸矩形框左边的坐标

top 人脸矩形框顶部的坐标

right 人脸矩形框右边的坐标

bottom 人脸矩形框底部的坐标

返回

void

### **1.5.4 void camrgb\_control\_expo\_weights\_90(int left, int top, int right, int bottom)**

说明

用于实现rgb图像顺时针旋转90度局部曝光，可以实现人脸坐标在暗处局部曝光。

参数

left 人脸矩形框左边的坐标

top 人脸矩形框顶部的坐标

right 人脸矩形框右边的坐标

bottom 人脸矩形框底部的坐标

返回

void

### **1.5.5 void camrgb\_control\_expo\_weights\_default(void)**

说明

用于配置还原默认的曝光设置。

参数

void

返回

void

### **1.5.6 void set\_rgb\_display(display\_callback cb)**

说明



设置RGB摄像头显示回调

参数

cb RGB摄像头显示回调

返回

void

### **1.5.7 void set\_rgb\_rotation(int angle)**

说明

设置RGB摄像头旋转角度

参数

angle 旋转角度，支持90，270

返回

void

## **1.6 camir\_control**

### **1.6.1 int camir\_control\_init(void)**

说明

实现了ir video的初始化，在process线程进行图像数据的读取、旋转、送数据给rockface做活体检测处理。

参数

void

返回

int 0成功，-1失败

### **1.6.2 void camir\_control\_exit(void)**

说明

实现了ir video的反初始化。

参数

void

返回

void

### **1.6.3 bool camir\_control\_run(void)**

说明

判断IR摄像头是否运行

参数

void

返回

bool true, 运行; false, 没运行

### 1.6.4 void set\_ir\_display(display\_callback cb)

说明

设置IR摄像头显示回调

参数

cb IR摄像头显示回调

返回

void

### 1.6.5 void set\_ir\_rotation(int angle)

说明

设置IR摄像头旋转角度

参数

angle 旋转角度, 支持90, 270

返回

void

## 1.7 shadow\_display

### 1.7.1 void shadow\_display(void \*src\_ptr, int src\_fd, int src\_fmt, int src\_w, int src\_h)

说明

实现横屏显示摄像头图像的功能, 会根据屏幕比例和摄像头图像比例裁剪出更合适的图像显示在横屏上面。

参数

src\_ptr 图像的数据地址

src\_fd 图像的数据fd

src\_fmt 图像的数据格式

src\_w 图像的宽

src\_h 图像的高

返回

void

### **1.7.2 void shadow\_display\_vertical(void \*src\_ptr, int src\_fd, int src\_fmt, int src\_w, int src\_h)**

说明

实现竖屏显示摄像头图像的功能，会根据屏幕比例和摄像头图像比例裁剪出更合适的图像显示在竖屏上面。

参数

src\_ptr 图像的数据地址

src\_fd 图像的数据fd

src\_fmt 图像的数据格式

src\_w 图像的宽

src\_h 图像的高

返回

void

### **1.7.3 void shadow\_paint\_box(int left, int top, int right, int bottom)**

说明

发送绘制人脸框消息给UI。

参数

left 人脸矩形框左边的坐标

top 人脸矩形框顶部的坐标

right 人脸矩形框右边的坐标

bottom 人脸矩形框底部的坐标

返回

void

### **1.7.4 void shadow\_paint\_info(struct user\_info \*info, bool real)**

说明

发送用户信息消息给UI。

参数

info 用户信息

返回

void

### 1.7.5 void shadow\_get\_crop\_screen(int \*width, int \*height)

说明

获取裁剪后屏幕尺寸范围。

参数

width 屏幕的宽

height 屏幕的高

返回

void

## 1.8 load\_feature

### 1.8.1 int count\_file(const char \*path, char \*fmt)

说明

计算某个目录包括子目录下面所有对应图像格式的文件数量。

参数

path 目录路径

fmt 图像格式

返回

int 文件的数量

### 1.8.2 int load\_feature(const char \*path, char \*fmt, void \*data, unsigned int cnt)

说明

从某个目录包括子目录下面所有对应格式文件的特征值和文件名读取到data对应的数据结构体指针，最多读取cnt个。

参数

path 目录路径

fmt 图像格式

data 存放读取的特征值和文件名的指针

cnt 最多读取多少个

返回

void

## 1.9 play\_wav

### 1.9.1 int play\_wav\_thread\_init(void)

#### 说明

完成play\_wav的初始化，并完成play\_wav\_thread线程的初始化，play\_wav\_thread线程等待接收signal播放指定wav文件。

#### 参数

void

#### 返回

int 0成功，-1失败

### 1.9.2 void play\_wav\_thread\_exit(void)

#### 说明

完成play\_wav的反初始化，并完成播放线程的反初始化。

#### 参数

void

#### 返回

void

### 1.9.3 void play\_wav\_signal(char \*name)

#### 说明

通过指定名字播放wav音频。

#### 参数

name wav音频文件名

#### 返回

void

音频格式要求16000采样率，双通道，16bit，可以修改以下3个宏指定其他音频格式。

```
1 | #define NUM_CHANNELS 2
2 | #define SAMPLE_RATE 16000
3 | #define BITS_PER_SAMPLE 16
```

添加wav音频：中文放到wav/cn即可，英文放到wav/en即可。

CMakeLists.txt里install(DIRECTORY wav/cn/ DESTINATION ../etc)指定使用中文或者英文音频文件安装到指定目录。

## 1.10 rga\_control

### 1.10.1 int rga\_control\_buffer\_init(bo\_t \*bo, int \*buf\_fd, int width, int height, int bpp)

说明

申请drm内存

参数

bo 申请目标内存bo参数

buf\_fd 申请目标内存buf\_fd参数

width 申请目标内存的宽

height 申请目标内存的高

bpp 申请目标内存一个像素对应的比特

返回

void

### 1.10.2 void rga\_control\_buffer\_deinit(bo\_t \*bo, int buf\_fd)

说明

释放drm内存

参数

bo 申请目标内存bo参数

buf\_fd 申请目标内存buf\_fd参数

返回

void

## 1.11 rkfacial

### 1.11.1 typedef void (\*display\_callback)(void \*ptr, int fd, int fmt, int w, int h, int rotation)

说明

显示回调

参数

ptr buffer的内存地址

fd buffer的内存地址对应fd

fmt buffer的格式

w buffer的宽

h buffer的高

rotation buffer旋转参数（参考linux-rga定义）

返回

void

### **1.11.2 void set\_rgb\_param(int width, int height, display\_callback cb, bool expo)**

说明

设置rgb摄像头的参数

参数

width rgb摄像头初始化宽

height rgb摄像头初始化高

cb rgb摄像头显示回调，可以为NULL

expo rgb摄像头局部曝光

返回

void

### **1.11.3 void set\_ir\_param(int width, int height, display\_callback cb)**

说明

设置ir摄像头的参数

参数

width ir摄像头初始化宽

height ir摄像头初始化高

cb ir摄像头显示回调，可以为NULL

返回

void

### **1.11.4 void set\_usb\_param(int width, int height, display\_callback cb)**

说明

设置USB摄像头的参数

参数

width USB摄像头初始化宽

height USB摄像头初始化高

cb USB摄像头显示回调，可以为NULL

返回

void

### **1.11.5 void set\_face\_param(int width, int height, int cnt)**

说明

设置face初始化的参数

参数

width face初始化宽

height face初始化高

cnt face最大数量

返回

void

### **1.11.6 int rkfacial\_init(void)**

说明

rkfacial初始化

参数

void

返回

int 0成功，-1失败

### **1.11.7 void rkfacial\_exit(void)**

说明

rkfacial退出

参数

void

返回

void

### **1.11.8 void rkfacial\_register(void)**

说明

rkfacial注册人脸



参数

void

返回

void

### **1.11.9 void rkfacial\_delete(void)**

说明

rkfacial删除人脸

参数

void

返回

void

### **1.11.10 void register\_rkfacial\_paint\_box(rkfacial\_paint\_box\_callback cb)**

说明

注册UI绘制人脸框回调

参数

cb UI绘制人脸框回调

返回

void

### **1.11.11 void register\_rkfacial\_paint\_info(rkfacial\_paint\_info\_callback cb)**

说明

注册UI绘制用户信息回调

参数

cb UI绘制用户信息回调

返回

void

## **1.12 snapshot**

### **1.12.1 int snapshot\_init(struct snapshot \*s, int w, int h)**

说明

snapshot初始化

参数

s snapshot信息

w snapshot 图片宽

h snapshot图片高

返回

int 0成功

### 1.12.2 void snapshot\_exit(struct snapshot \*s)

说明

snapshot反初始化

参数

s snapshot信息

返回

void

### 1.12.3 int snapshot\_run(struct snapshot \*s, rockface\_image\_t \*image, rockface\_det\_t \*face, RgaSURF\_FORMAT fmt, long int sec, char mark)

说明

snapshot拍照

参数

s snapshot信息

image 图像image

face 人脸face

fmt 图像格式

sec 抓拍最小间隔（秒）

mark 抓拍标记

返回

int 0成功

## 1.13 turbojpeg\_decode

### 1.13.1 void \*turbojpeg\_decode\_get(const char \*name, int \*w, int \*h, int \*b)

说明

turbojpeg解码MJPEG获取图像buffer

参数

name 图像路径名

w 存储图像分辨率宽

h 存储图像分辨率高

b 存储图像每个像素的byte数

返回

void \* MJPEG解码后buffer

### 1.13.2 void turbojpeg\_decode\_put(void \*data)

说明

释放turbojpeg解码MJPEG获取图像的buffer

参数

data MJPEG解码后buffer

返回

void

## 1.14 usb\_camera

### 1.14.1 int usb\_camera\_init(void)

说明

usb camera初始化

参数

void

返回

int 0成功

### 1.14.2 void usb\_camera\_exit(void)

说明

usb camera反初始化

参数

void

返回

void

### 1.14.3 void set\_usb\_display(display\_callback cb)

说明

设置usb camera显示回调

参数

cb 显示回调

返回

void

### 1.14.4 void set\_usb\_rotation(int angle)

说明

设置usb camera

参数

angle 旋转角度，支持90，270

返回

void

## 1.15 vpu decode(MJPEG decode)

### 1.15.1 int vpu\_decode\_jpeg\_init(struct vpu\_decode\* decode, int width, int height)

说明

vpu decode初始化

参数

decode vpu\_decode信息

width MJPEG图像宽

height MJPEG图像高

返回

int 0成功

### 1.15.2 int vpu\_decode\_jpeg\_doing(struct vpu\_decode\* decode, void\* in\_data, RK\_S32 in\_size, int out\_fd, void\* out\_data)

说明

vpu decode解码

参数

decode vpu\_decode信息

in\_data MJPEG图像数据

in\_size MJPEG图像大小

out\_fd 解码数据fd

out\_data 解码数据data

返回

int 0成功

### 1.15.3 int vpu\_decode\_jpeg\_done(struct vpu\_decode\* decode)

说明

vpu decode反初始化

参数

decode vpu\_decode信息

返回

int 0成功

## 1.16 vpu encode(MJPEG encode)

### 1.16.1 int vpu\_encode\_jpeg\_init(struct vpu\_encode\* encode, int width, int height, int quant, MppFrameFormat format)

说明

vpu encode初始化

参数

encode vpu\_encode信息

width MJPEG图像宽

height MJPEG图像高

quant MJPEG编码quant

format 编码输入图像格式

返回

int 0成功

### 1.16.2 int vpu\_encode\_jpeg\_doing(struct vpu\_encode\* encode, void\* srcbuf, int src\_fd, size\_t src\_size, void \*dst\_buf, int dst\_fd, size\_t dst\_size)

说明

vpu\_encode初始化

参数

encode vpu\_encode信息

srcbuf 输入图像buffer

src\_fd 输入图像fd

src\_size 输入图像大小

dst\_buf 输出图像buf

dst\_fd 输出图像fd

dst\_size 输出图像buf初始化大小

返回

int 0成功

### **1.16.3 void vpu\_encode\_jpeg\_done(struct vpu\_encode\* encode)**

说明

vpu\_encode反初始化

参数

encode vpu\_encode信息

返回

void