

Rockchip Rkfacial 使用说明

文件标识：RK-SM-YF-363

发布版本：V2.0.3

日期：2022-02-16

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，由其各自拥有者所有。

版权所有© 2022 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

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

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
RK1808, RK1806	Linux 4.4
RV1126, RV1109	Linux 4.19

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
2020-10-19	V2.0.1	Zhihua Wang	增加RV1126, RV1109支持
2021-03-15	V2.0.2	Ruby Zhang	完善产品版本信息
2022-02-16	V2.0.3	Ruby Zhang	文档格式更新

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* srebuf, 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

id 用户id

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

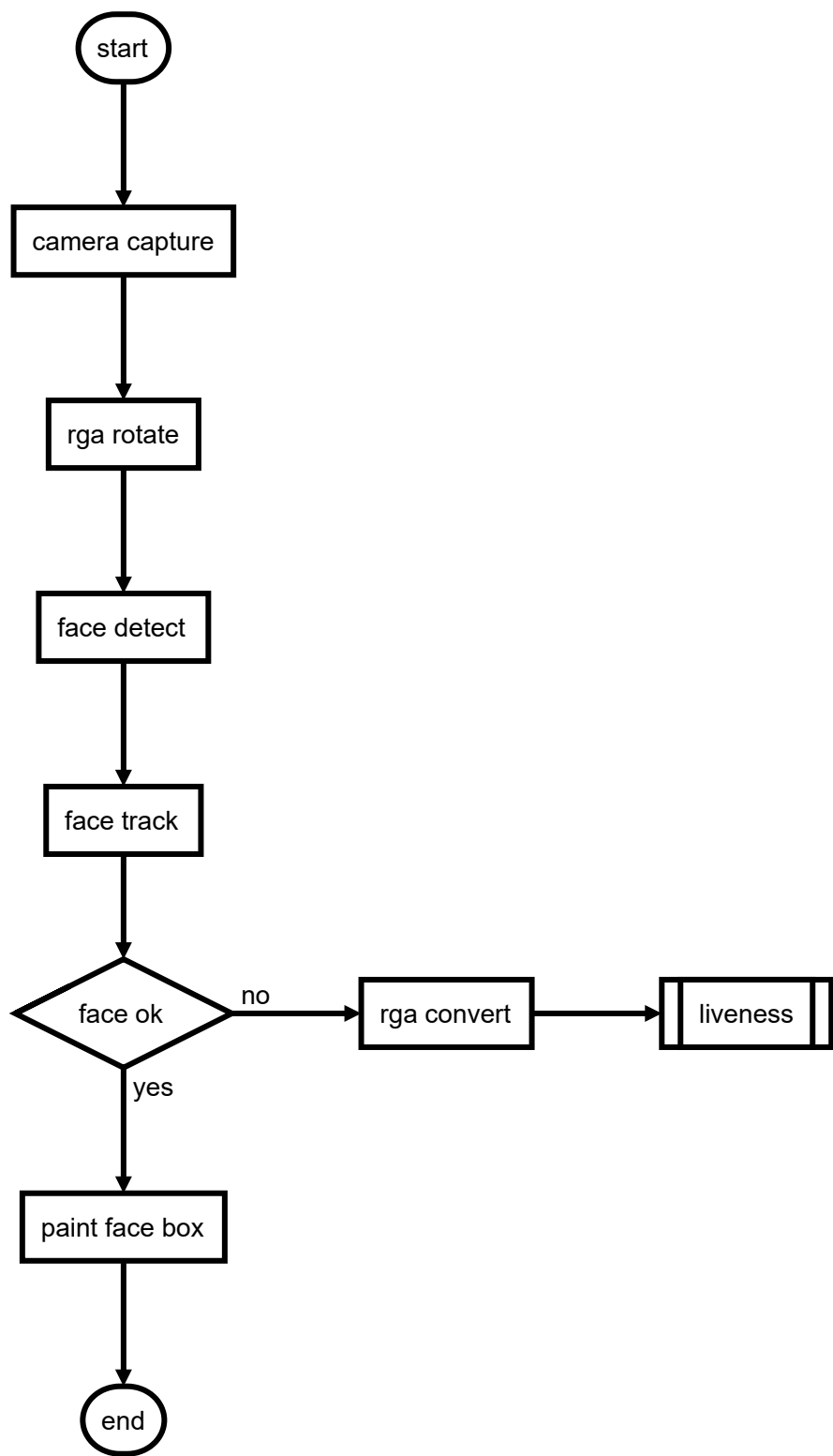
默认宏定义说明：

```
#define DEFAULT_FACE_NUMBER 1000 表示默认人脸数据库最大支持人脸数量
#define DEFAULT_FACE_PATH "/userdata" 开机默认从这个目录加载jpg文件获取特征值
#define FACE_DETECT_SCORE 0.55 人脸检测的分数，范围0-1，越大越严格
#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
#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
#define FACE_SIMILARITY_CONVERT(f) powf(2.0, -((f))) RGB人脸识别的分数转化公式
#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
#define FACE_SCORE_REGISTER 0.99 人脸注册的人脸分数，范围0-1，越大越严格
```

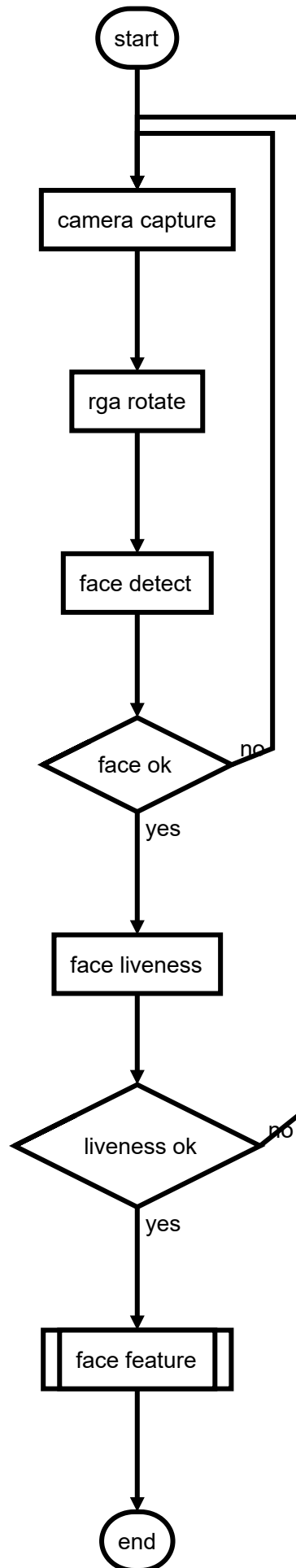
```
#define FACE_REGISTER_CNT 5 人脸注册时连续读到的多少次人脸特征值均在数据库里面，提示已经注册
#define FACE_REAL_SCORE 0.5 活体检测分数最小要求，范围0-1，越大越严格
#define LICENCE_PATH PRE_PATH "/key.lic" rockface人脸授权key存放路径
#define BAK_LICENCE_PATH BAK_PATH "/key.lic" rockface人脸授权备份key存放路径
#define FACE_DATA_PATH "/usr/lib" rockface data存放路径
#define MIN_FACE_WIDTH(w) ((w) / 5) 人脸检测、特征值提取人脸最小像素要求
#define FACE_TRACK_FRAME 0 人脸跟踪最大跟踪时间(帧)
#define FACE_RETRACK_TIME 1 人脸跟踪再次跟踪时间(秒)
#define SNAP_TIME 3 抓拍最低间隔时间(秒)
```

应用流程图

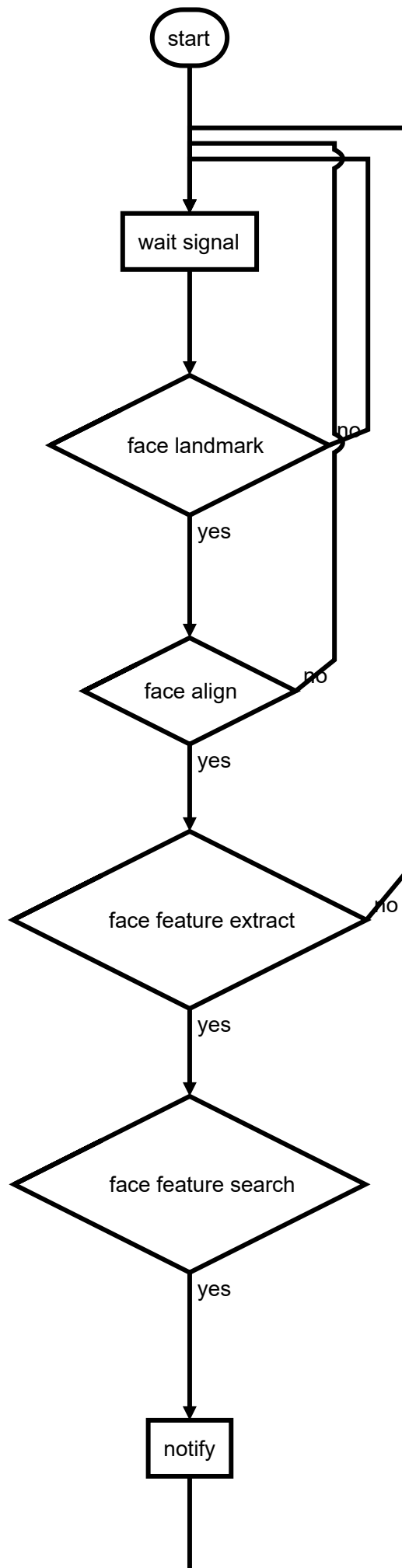
- RGB人脸检测

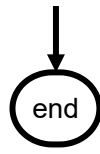


- IR人脸检测、活体检测



- 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。

```
#define NUM_CHANNELS 2
#define SAMPLE_RATE 16000
#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