

```

1 #include "rga_func.h"
2
3 //根据librga库实际路径进行修改
4 #define LIBRGAFILE "../lib/librga.so"
5 int RGA_init(rga_context *rga_ctx) {
6     rga_ctx->rga_handle = dlopen(LIBRGAFILE, RTLD_LAZY);
7     if (!rga_ctx->rga_handle) {
8         printf("dlopen %s failed\n", LIBRGAFILE);
9         return -1;
10    }
11    rga_ctx->init_func = (FUNC_RGA_INIT)dlsym(rga_ctx->rga_handle, "c_RkRgaInit");
12    rga_ctx->deinit_func =
13        (FUNC_RGA_DEINIT)dlsym(rga_ctx->rga_handle, "c_RkRgaDeInit");
14    rga_ctx->blit_func = (FUNC_RGA_BLIT)dlsym(rga_ctx->rga_handle, "c_RkRgaBlit");
15    rga_ctx->init_func();
16    return 0;
17 }
18
19 void img_resize_fast(rga_context *rga_ctx, int src_fd, int src_w, int src_h,
20                     uint64_t dst_phys, int dst_w, int dst_h) {
21     // printf("rga use fd, src(%dx%d) -> dst(%dx%d)\n", src_w, src_h, dst_w,
22     // dst_h);
23
24     if (rga_ctx->rga_handle) {
25         int ret = 0;
26         rga_info_t src, dst;
27
28         memset(&src, 0, sizeof(rga_info_t));
29         src.fd = src_fd;
30         src.mmuFlag = 1;

```

```

31 // src.virAddr = (void *)psrc;
32
33 memset(&dst, 0, sizeof(rga_info_t));
34 dst.fd = -1;
35 dst.mmuFlag = 0;
36
37 #if defined(__arm__)
38 dst.phyAddr = (void *)((uint32_t)dst_phys);
39 #else
40 dst.phyAddr = (void *)dst_phys;
41 #endif
42
43 dst.nn.nn_flag = 0;
44
45 rga_set_rect(&src.rect, 0, 0, src_w, src_h, src_w, src_h,
46 RK_FORMAT_RGB_888);
47 rga_set_rect(&dst.rect, 0, 0, dst_w, dst_h, dst_w, dst_h,
48 RK_FORMAT_RGB_888);
49
50 ret = rga_ctx->blit_func(&src, &dst, NULL);
51 if (ret) {
52 printf("c_RkRgaBlit error : %s\n", strerror(errno));
53 }
54
55 return;
56 }
57 return;
58 }
59
60 void img_resize_slow(rga_context *rga_ctx, void *src_virt, int src_w, int src_h,
61 void *dst_virt, int dst_w, int dst_h) {
62 // printf("rga use virtual, src(%dx%d) -> dst(%dx%d)\n", src_w, src_h, dst_w,
63 // dst_h);
64
65 if (rga_ctx->rga_handle) {
66 int ret = 0;
67 rga_info_t src, dst;
68
69 memset(&src, 0, sizeof(rga_info_t));
70 src.fd = -1;
71 src.mmuFlag = 1;
72 src.virAddr = (void *)src_virt;
73
74 memset(&dst, 0, sizeof(rga_info_t));
75 dst.fd = -1;
76 dst.mmuFlag = 1;
77 dst.virAddr = dst_virt;
78
79 dst.nn.nn_flag = 0;
80
81 rga_set_rect(&src.rect, 0, 0, src_w, src_h, src_w, src_h,
82 RK_FORMAT_RGB_888);
83 rga_set_rect(&dst.rect, 0, 0, dst_w, dst_h, dst_w, dst_h,
84 RK_FORMAT_RGB_888);
85
86 ret = rga_ctx->blit_func(&src, &dst, NULL);
87 if (ret) {
88 printf("c_RkRgaBlit error : %s\n", strerror(errno));
89 }
90 return;
91 }
92 return;
93 }
94
95 int RGA_deinit(rga_context *rga_ctx) {
96 if (rga_ctx->rga_handle) {
97 dlclose(rga_ctx->rga_handle);
98 rga_ctx->rga_handle = NULL;
99 }
100 }

```

3. DRM部分代码（使用瑞芯微官方源码）

drm_func.h

```

1 #ifndef __DRM_FUNC_H__
2 #define __DRM_FUNC_H__
3 #include <errno.h>
4 #include <linux/input.h>
5 #include <stdint.h>
6 #include <stdio.h>
7 #include <stdlib.h>
8 #include <string.h>
9 #include <sys/fcntl.h> // open function
10 #include <sys/mman.h>
11 #include <unistd.h> // close function
12
13 #include "libdrm/drm_fourcc.h"
14 #include "xf86drm.h"
15
16 #ifdef __cplusplus
17 extern "C" {
18 #endif
19
20 typedef int (*FUNC_DRM_IOCTL)(int fd, unsigned long request, void *arg);
21
22 typedef struct _drm_context {
23 void *drm_handle;
24 FUNC_DRM_IOCTL io_func;
25 } drm_context;
26
27 int drm_init(drm_context *drm_ctx);
28
29 void *drm_buf_alloc(drm_context *drm_ctx, int drm_fd, int TexWidth,
30 int TexHeight, int bpp, int *fd, unsigned int *handle,
31 size_t *actual_size);
32
33 int drm_buf_destroy(drm_context *drm_ctx, int drm_fd, int buf_fd, int handle,
34 void *drm_buf, size_t size);
35
36 void drm_deinit(drm_context *drm_ctx, int drm_fd);
37
38 #ifdef __cplusplus
39 }
40 #endif
41 #endif /* __DRM_FUNC_H__ */

```

rga_func.cpp

```

1 #include "drm_func.h"
2
3 #include <dlfcn.h>

```

```

5 //根据libdrm库实际路径进行修改
6 #define LIBDRMFILE ".../lib/libdrm.so"
7
8 int drm_init(drm_context *drm_ctx) {
9     static const char *card = "/dev/dri/card0";
10    int flag = 0_RDWR;
11    int drm_fd = -1;
12
13    drm_fd = open(card, flag);
14    if (drm_fd < 0) {
15        printf("failed to open %s\n", card);
16        return -1;
17    }
18
19    drm_ctx->drm_handle = dlopen(LIBDRMFILE, RTLD_LAZY);
20    if (!drm_ctx->drm_handle) {
21        printf("failed to dlopen %s\n", LIBDRMFILE);
22        drm_deinit(drm_ctx, drm_fd);
23        return -1;
24    }
25
26    drm_ctx->io_func = (FUNC_DRM_IOCTL)dlsym(drm_ctx->drm_handle, "drm_ioctl");
27    if (drm_ctx->io_func == NULL) {
28        dlclose(drm_ctx->drm_handle);
29        drm_ctx->drm_handle = NULL;
30        drm_deinit(drm_ctx, drm_fd);
31        printf("failed to dlsym drm_ioctl\n");
32        return -1;
33    }
34    return drm_fd;
35}
36
37void drm_deinit(drm_context *drm_ctx, int drm_fd) {
38    if (drm_ctx->drm_handle) {
39        dlclose(drm_ctx->drm_handle);
40        drm_ctx->drm_handle = NULL;
41    }
42    if (drm_fd > 0) {
43        close(drm_fd);
44    }
45}
46
47void *drm_buf_alloc(drm_context *drm_ctx, int drm_fd, int TexWidth,
48                    int TexHeight, int bpp, int *fd, unsigned int *handle,
49                    size_t *actual_size) {
50    int ret;
51    if (drm_ctx == NULL) {
52        printf("drm context is invalid\n");
53        return NULL;
54    }
55    char *map = NULL;
56
57    void *vir_addr = NULL;
58    struct drm_prime_handle fd_args;
59    struct drm_mode_map_dumb mmap_arg;
60    struct drm_mode_destroy_dumb destroy_arg;
61
62    struct drm_mode_create_dumb alloc_arg;
63
64    memset(&alloc_arg, 0, sizeof(alloc_arg));
65    alloc_arg.bpp = bpp;
66    alloc_arg.width = TexWidth;
67    alloc_arg.height = TexHeight;
68    // alloc_arg.flags = ROCKCHIP_BO_CONTIG;
69
70    //获取handle和size
71    ret = drm_ctx->io_func(drm_fd, DRM_IOCTL_MODE_CREATE_DUMB, &alloc_arg);
72    if (ret) {
73        printf("failed to create dumb buffer: %s\n", strerror(errno));
74        return NULL;
75    }
76    if (handle != NULL) {
77        *handle = alloc_arg.handle;
78    }
79    if (actual_size != NULL) {
80        *actual_size = alloc_arg.size;
81    }
82    // printf("create width=%u, height=%u, bpp=%u, size=%lu dumb
83    // buffer\n", alloc_arg.width, alloc_arg.height, alloc_arg.bpp, alloc_arg.size);
84    // printf("out handle= %d\n", alloc_arg.handle);
85
86    //获取fd
87    memset(&fd_args, 0, sizeof(fd_args));
88    fd_args.fd = -1;
89    fd_args.handle = alloc_arg.handle;
90    ;
91    fd_args.flags = 0;
92    ret = drm_ctx->io_func(drm_fd, DRM_IOCTL_PRIME_HANDLE_TO_FD, &fd_args);
93    if (ret) {
94        printf("rk-debug handle_to_fd failed ret=%d,err=%s, handle=%x\n", ret,
95              strerror(errno), fd_args.handle);
96        return NULL;
97    }
98    // printf("out fd = %d, drm fd: %d\n", fd_args.fd, drm_fd);
99    if (fd != NULL) {
100        *fd = fd_args.fd;
101    }
102
103    //获取虚拟地址
104    memset(&mmap_arg, 0, sizeof(mmap_arg));
105    mmap_arg.handle = alloc_arg.handle;
106
107    ret = drm_ctx->io_func(drm_fd, DRM_IOCTL_MODE_MAP_DUMB, &mmap_arg);
108    if (ret) {
109        printf("failed to create map dumb: %s\n", strerror(errno));
110        vir_addr = NULL;
111        goto destroy_dumb;
112    }
113    vir_addr = map = mmap(0, alloc_arg.size, PROT_READ | PROT_WRITE, MAP_SHARED,
114                          drm_fd, mmap_arg.offset);
115    if (map == MAP_FAILED) {
116        printf("failed to mmap buffer: %s\n", strerror(errno));
117        vir_addr = NULL;
118        goto destroy_dumb;
119    }
120    // printf("alloc map=%x\n", map);
121    return vir_addr;
122destroy_dumb:
123    memset(&destroy_arg, 0, sizeof(destroy_arg));
124    destroy_arg.handle = alloc_arg.handle;
125    ret = drm_ctx->io_func(drm_fd, DRM_IOCTL_MODE_DESTROY_DUMB, &destroy_arg);
126    if (ret) printf("failed to destroy dumb %d\n", ret);

```

```
127     return vir_addr;
128 }
129
130 int drm_buf_destroy(drm_context *drm_ctx, int drm_fd, int buf_fd, int handle,
131                   void *drm_buf, size_t size) {
132     int ret = -1;
133     if (drm_buf == NULL) {
134         printf("drm buffer is NULL\n");
135         return -1;
136     }
137
138     munmap(drm_buf, size);
139
140     struct drm_mode_destroy_dumb destroy_arg;
141     memset(&destroy_arg, 0, sizeof(destroy_arg));
142     destroy_arg.handle = handle;
143     ret = drm_ctx->io_func(drm_fd, DRM_IOCTL_MODE_DESTROY_DUMB, &destroy_arg);
144     if (ret)
145         printf("failed to destroy dumb %d, error=%s\n", ret, strerror(errno));
146     if (buf_fd > 0) {
147         close(buf_fd);
148     }
149
150     return ret;
151 }
152
```

4. 封装接口代码

image_util.h

```
1  #ifndef _IMAGE_UTIL_H
2  #define _IMAGE_UTIL_H
3
4  #include "drm_func.h"
5  #include "opencv2/opencv.hpp"
6  #include "rga_func.h"
7
8  class ImageUtil {
9  private:
10     ImageUtil() { init(); };
11     virtual ~ImageUtil() { release(); };
12     ImageUtil(const ImageUtil &) = delete;
13     ImageUtil(ImageUtil &&) = delete;
14     ImageUtil &operator=(const ImageUtil &) = delete;
15     ImageUtil &operator=(ImageUtil &&) = delete;
16     void *drm_buf = NULL;
17     int drm_fd = -1;
18     int buf_fd = -1; // converted from buffer handle
19     unsigned int handle;
20     size_t actual_size = 0;
21     rga_context rga_ctx;
22     drm_context drm_ctx;
23     void init(void);
24     void release(void);
25
26 public:
27     static ImageUtil &getInstance();
28
29     void resize(const cv::Mat &src, const cv::Size &size, void *dstPtr);
30 };
31
32 #endif
```

image_util.cpp

```
1  #include "image_util.h"
2
3  void ImageUtil::init(void) {
4      memset(&rga_ctx, 0, sizeof(rga_ctx));
5      memset(&drm_ctx, 0, sizeof(drm_ctx));
6      drm_fd = drm_init(&drm_ctx);
7      drm_buf = drm_buf_alloc(&drm_ctx, drm_fd, 1920, 1080, 24, &buf_fd, &handle,
8                             &actual_size);
9      RGA_init(&rga_ctx);
10 }
11
12 void ImageUtil::release(void) {
13     drm_buf_destroy(&drm_ctx, drm_fd, buf_fd, handle, drm_buf, actual_size);
14     drm_deinit(&drm_ctx, drm_fd);
15     RGA_deinit(&rga_ctx);
16 }
17
18 void ImageUtil::resize(const cv::Mat &src, const cv::Size &size, void* dstPtr) {
19     if (src.empty()) {
20         printf("src is empty!\n");
21         return;
22     }
23     int img_width = src.cols;
24     int img_height = src.rows;
25     memcpy(drm_buf, src.data, img_width * img_height * 3);
26     img_resize_slow(&rga_ctx, drm_buf, img_width, img_height, dstPtr, size.width,
27                    size.height);
28 }
29
30 ImageUtil& ImageUtil::getInstance() {
31     static ImageUtil util;
32     return util;
33 }
```

至此，前置准备工作基本已准备就绪，下一步将对大家感兴趣的各类代码，进行移植与测试，欢迎各位粉丝们的积极留言与探讨。

📖 文章知识点与官方知识档案匹配，可进一步学习相关知识

OpenCV技能树 > OpenCV中的深度学习 > 图像分类 17668 人正在系统学习中

Linux rga说明文档.pdf

02-11

Rockchip 提供的librga库主要用于对常用图形处理（裁剪、格式转换、图片叠、填色等）做硬件加速。本文为librga库的说明文档，提供了api详细介绍。

drm+vv42零拷贝

06-14

嵌入式Linux平台实现v4l2和DRM实现零拷贝直接v4l2获取camera数据直接显示，v4l2获取camera数据直接显示欢迎大家下载使用！

6 条评论



m0_37147042

热评

img_resize_fast接口参数给个解释和demo，谢谢！

写评论

RK3568 应用层 libhwjpeg.so 硬件编解码的使用_Xavier 祥子的博客...

5-27

libhwjpeg是RK封装的一个对MJPEG数据进行硬件编解码的so,用起来还是比较方便.相比软编效率也有提升。应用层没有提供接口,所以需要自己封装.N接...

RK Android平台RGA功能介绍和接口封装_rga_set_rect_FreeFly的博客-C...

5-28

RGA接口函数 librga	是在用C++编写还是在文件RockchipRga.h中有以下类的方法提供接口：/*初始化rga */ int RkRgaInit(); /*申请buffer */ int RkRgaGetAlloc... Aisy_Du的专栏 @ 67
Rockchip rv1126平台在gststreamer1中添加OpenCV实现视频流切片的效果	最新发布 rv1126 gststreamer opencv negative effect Aisy_Du的专栏 @ 67
关于GPU图像处理下的DRM/RGA/MPP	tltcly的博文 @ 4009 直接渲染管理器(Direct Rendering Manager)是给予DRI客户端直接访问硬件的内核模块。该模块处理DMA、AGP内存管理，资源锁和安全硬件访问。为了...
RK3399pro-认识一下RK3399Pro的硬件_荏苒的博客	5-25 好了言归正传，此篇文章开始又要更新我的RK3399Pro的开发篇了。这次是因为工作所需，开始做产品了！介绍一下基础部分：首先要知道嵌入式开发和PC开...
RK3399PRO 硬件H264编码,RTSP推流笔记_rk3399 硬编码_sac761的博客-CSDN...	5-20 1.按照《RK3399Pro入门教程》(10)RTSP推流介绍配置好环境,安装好官方驱动,其中需要对rga降低版本到1.0.1-1.0【sudo dnf install librockchip_rga-devel...
瑞芯微rv1126 rtsp+mpv+rga取流	ZuoSeDiao的博客 @ 2398 瑞芯微rv1109 rtsp+mpv+rga取流
RV1106 stb图像库、opencv、rga对比评测	qq_32582681的博文 @ 232 rknn opencv rga
RK3399 Linux rga 调试记录_rockchip rga2_「已注销」的博客	5-15 +CONFIG_RKROCKCHIP_RGBA=y 更新内核查看/dev/rga2节点是否存在 三EMIO编译 1、下载代码到开发板的/opt目录下,并重命名为librga_linux.2、复制...
RK系列 RGA模块移植 开发	yinsui1839的博文 @ 800 RK系列 RGA模块移植 开发
rknn_yolov5执行流程	fuhanga123的博文 @ 1189 RV1109上测试rknn_yolov5_demo以及分析
RK3399-RGA使用	保持一颗敬畏之心，简单地调用C++，适当地向C++... @ 5112 目录前言正文资料的下载参考 前言 由于工作要用到这个功能，所以，弄了很久，后面发现的几个问题，确实都是比较low导致的。这里做一个记录，...
RKMEDIA--RGA的使用	未定的博文 @ 2015 rkmedia rga的使用介绍
瑞芯微RV1126/1109开发流程之驱动升级	xidaoliang的博文 @ 1834 RV1126驱动升级以及硬件参数查看
RK3588 RGA 图像操作	wexin_46935110的博文 @ 4572 RK3588 RGA 图像操作
Rockchip RV1126 模型部署（完整部署流程）	DAOCHI @ 1059 YOLO-V5在RV1126部署全流程（环境配置，模型转换，C++模型推理，芯片运行），验证有效。
瑞芯微-RV1109主板各参数介绍	DeepLearning_的博文 @ 1720 瑞芯微电子Rockchip（以下简称瑞芯微），瑞芯微电子成立于2001年，总部位于福州，在深圳、上海、北京、杭州、香港及台湾均设有分支机构，是中国...
RK3399 Qt+RTSP+FFmpeg+MPP+RGA解码	嵌入式博文 @ 4379 RK3399 Qt+RTSP+FFmpeg+MPP+RGA解码
img_resize()函数的作用和用法-单张图像变换大小	天空的味道 @ 4857 这个是一个学过半年的简单程序，可以改变图像的大小，jpg，png都可以的：#encoding=utf-8 #author: walker #date: 2014-05-15 #function: 更改图片尺寸大...
Linux aarch64交叉编译的 mesa图形库	wangshd的博文 @ 1987 对于mesa的交叉编译。该文章的目标是编译一套aarch64LinuxDebian嵌入式版本上可以运行的版本库，接下来就开始编程。老套路，先把linux桌面搞...
RockChip RGA与DRM大小端命名冲突解析	lyy901135f的博文 @ 565 RockChip RGB图像经过RGA处理后显示异常——像素格式大小端命名不统一
报错src unsupported width stride 300.rgb888 width stride should be 16 aligned（已解决）	MMH1028的博文 @ 41 在使用opencv进行图像显示时，系统报错src unsupported width stride 300.rgb888 width stride should be 16 aligned。解决方案：将图片像素改为16的倍数...
“相关推荐”对你有帮助么？	
<input checked="" type="radio"/> 非常有帮助 <input type="radio"/> 有帮助 <input type="radio"/> 一般 <input type="radio"/> 没帮助 <input type="radio"/> 非常没帮助	
关于我们 招贤纳士 商务合作 寻求报道 ☎ 400-660-0108 ✉ kefu@csdn.net ☎ 在线客服 ⌚ 工作时间 8:30-22:00	
公安备案号11010502030143 京ICP备19004658号 京公网安备 20201039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 账号管理规范 版权与免费声明 版权声明 出版物许可证 营业执照 © 1999-2023北京创新乐知网络技术有限公司	