

# Rockchip Instructions Qrcode 使用说明

---

文件标识: RK-SM-YF-396

发布版本: V1.0.0

日期: 2020-10-29

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 © 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

本文阐述二维码扫码库的接口说明。

产品版本

芯片名称	内核版本
RV1109/RV1126	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

版本号	作者	修改日期	修改说明
V1.0.0	Zack.Huang	2020-10-29	初始版本

## 目录

### Rockchip Instructions Qrcode 使用说明

1. Qrcode介绍
2. Qrcode的数据类型介绍
3. Qrcode的接口说明
4. 使用范例

# 1. Qrcode介绍

Qrcode是一种扫描二维码的应用库，目前我们使用的是以zbar为基础进行优化的一套代码，在RV1126/RV1109平台上，以a库的形式提供。路径为：

```
SDK/app/mediaserver/src/utils/zbar/librkbar.a
```

头文件路径为：

```
SDK/app/mediaserver/src/utils/zbar/rkbar_scan_api.h
```

用以下方式在menuconfig中添加QR支持：

Target packages --->Rockchip BSP packages --->rockchip mediaserver ---> Enable zbar to scan QR code

# 2. Qrcode的数据类型介绍

```
typedef struct image_s {
    unsigned width, height; /* 输入图像的长和宽 */
    void *data; /* 需要喂数据的图像数据（灰度图像） */
    unsigned long datalen; /* 数据长度 */
    unsigned crop_x, crop_y; /* 扫描矩形，可以都赋值为0 */
    unsigned crop_w, crop_h; /* 输入图像的长宽 */
    void *userdata; /* 用户指定的数据与图像相关 */
    uint8_t *bin; /* 图像内存指针 */
    uint8_t *tmp; /* NULL */
} image_t;
```

# 3. Qrcode的接口说明

```
int rkbar_init(void **handle); /* 初始化rkbar句柄 */
int rkbar_scan(void *handle, image_t *src); /* 解析图像信息 */
const char *rkbar_getresult(void *handle); /* 得到解析结果 */
void rkbar_deinit(void *handle); /* 释放句柄 */
```

# 4. 使用范例

```
#include "zbar/rkbar_scan_api.h"

using namespace std;
```

```

extern "C" int zbar_test(int argc, char** argv)
{
    printf("start to qrcode_local test....\n");
    char *result_data = NULL;
    image_t *img = NULL;
    int init_width = 320;
    int init_height = 240;
    uint8_t *zoom_data = NULL;
    zoom_data = (uint8_t*)malloc(320*240*sizeof(char)+1);
    userdata image = user_read_data_fun("C:\\zbartest.bmp", IMREAD_GRAYSCALE); //
    使用自定义的方式读取数据。
    printf("start to qrcode_local test....\n");
    img = (image_t*)malloc(sizeof(image_t));
    result_data = (char*)malloc(100*sizeof(char));
    img->width = init_width;
    img->height = init_height;
    img->crop_x = 0;
    img->crop_y = 0;
    img->crop_w = init_width;
    img->crop_h = init_height;
    img->bin = (unsigned char*)malloc(img->width* img->height);
    img->tmp = NULL;
    void *rkbar_hand = NULL;
    printf("start to qrcode_local test....\n");
    int ret = rkbar_init(&rkbar_hand);
    if (ret == -1){
        printf("init is err");
        return -1;
    }

    printf("start to qrcode_local test....\n");
    img->data = image.data;

    ret = rkbar_scan(rkbar_hand, img);
    printf("\nret = %d\n", ret);
    if (ret > 0){
        const char *data = rkbar_getresult(rkbar_hand);
        memcpy(result_data, data, 100 * sizeof(char));
        printf("The decoding result is \" %s \" \n", result_data);
    }
    rkbar_deinit(rkbar_hand);
    if(zoom_data){
        free(zoom_data);
    }
    if (img){
        free(img);
    }
    if(result_data){
        free(result_data);
    }

    return 0;
}

```

