Rockchip Qrcode Instructions

ID: RK-SM-YF-396

Release Version: V1.0.0

Release Date: 2020-10-29

Security Level: □Top-Secret □Secret □Internal ■Public

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian,PRC

Website: www.rock-chips.com

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: fae@rock-chips.com

Preface

Overview

This document is going to introduce the interface of the QR code scanning library.

Product Version

Chipset	Kernel Version
RV1109/RV1126	Linux 4.19

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

Revision History

Version	Author	Date	Change Description
V1.0.0	Zack.Huang	2020-10-29	Initial version

Contents

1. Qrcode Overview

Qrcode is an application library for scanning QR codes. At present, the codes we used is optimized based on zbar. It is provided in the form of a library on the RV1126 or RV1109 platform. The path is:

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

The header file is located in:

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

Add QR support in menuconfig in the following way:

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

2. Qrcode Data Type Introduction

```
  typedef struct image_s {
     unsigned width, height;  /* The length and width of the input image */
     void *data;              /* Image data (grayscale image) that needs to be fed */
     unsigned long datalen;   /* Data length */
     unsigned crop_x, crop_y; /* Scanning rectangle, both can be assigned a value of 0 */
     unsigned crop_w, crop_h; /* The length and width of the input image */
     void *userdata;          /* User-specified data is related to the image */
     uint8_t *bin;            /* Image memory pointer */
     uint8_t *tmp;            /* NULL */
  } image_t;
```

3. Qrcode Interface Introduction

```
int rkbar_init(void **handle);                     /* Initialize rkbar handle */
int rkbar_scan(void *handle, image_t *src);        /* Analyze image information */
const char *rkbar_getresult(void *handle);         /* Get the analysis result */
void rkbar_deinit(void *handle);                   /* Release handle */
```

4. Examples

```
#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); //Read data in a custom way
    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;
}
```