

RT-Thread一般讨论

成功在 stm32 上用 Zbar 库识别二维码！

发布于 2019-06-29 12:01:28 浏览:2768

[md]目前能在网上找到的二维码资料，都是编码居多，也就是从一串字符串生成二维码（常用算法如 QRCode），而解码识别方面的参考教程比较少。二维码解码库有 Quirc、Zbar 等，更多的开发者是用在安卓、OpenCV 和 Python 等高级环境上，对于 MCU 级别的使用，并且基本是经过封装后的实现，灵活性较差。于是便想着自己从头使用 Zbar 库的接口，实现图片中的二维码检测及识别。

测试环境

- * 开发板：STM32F429I Disco
- * CPU：Arm® 32-bit Cortex®-M4 180 MHz
- * 内存：2Mbytes Flash, 256KMbytes RAM, 64-Mbit 外部 SDRAM
- * 软件：RT-Thread 操作系统

实现过程

这里说明一下，Zbar 库要求输入的是灰度图像数据（也就是黑白照），每个像素用一个字节表示灰度，从 0x00 全白到 0xff 全黑，有了图像数据，再传入图像的宽高便能解码了。

通过 Zbar 库识别二维码，大致可以分为下面 3 个过程：

1. 创建配置 image_scanner、zbar_image 对象，并设置图像信息
2. 扫描图像并识别
3. 提取解码结果

我写了一个函数，能够直接读取 bmp 8 位灰度图片（因为 bmp 文件包含了图像数据便于处理），并打印输出其中的二维码识别结果：

```
``c
#include
#include

#include "zbar.h"

#define BITMAP_IMG_OFFSET 0x0A
#define BITMAP_IMG_WIDTH_OFFSET 0x12
#define BITMAP_IMG_HEIGHT_OFFSET 0x16
#define BITMAP_IMG_FILE_SIZE_OFFSET 0x02

zbar_image_scanner_t *rt_scanner = NULL;

void qr_bitmap_decoder(const rt_uint8_t *img_buffer)
{
    /* create a reader */
    rt_scanner = zbar_image_scanner_create();

    /* configure the reader */
    zbar_image_scanner_set_config(rt_scanner, 0, ZBAR_CFG_ENABLE, 1);
```

关注者
0

被浏览
2.8k

关于作者



jijigojiji

这家伙很懒，什么也没写！

提问
0

回答
0

被采纳
0

+ 关注TA

发私信

相关问题

- 1 有关动态模块加载的一篇论文
- 2 最近的调程序总结
- 3 晕掉了，这么久都不见layer2的踪影啊
- 4 继续K9ii的历程
- 5 [GUI相关] FreeType 2
- 6 [GUI相关]The RFB Protocol中文版
- 7 [GUI相关]嵌入式系统中文输入法的设计
- 8 20081101 RT-Thread开发者聚会总结
- 9 嵌入式系统基础
- 10 linux2.4.19在at91rm9200 上的寄存器设置

推荐文章

- 1 RT-Thread应用项目汇总
- 2 玩转RT-Thread系列教程
- 3 机器人操作系统 (ROS2) 和 RT-Thread 通信
- 4 关于STM32H7开发板上使用SDIO接口驱动SD卡挂载文件系统的问题总结
- 5 【技术三千问】之《玩转ART-Pi》，看这篇就够了！干货汇总
- 6 RT-Thread隐藏的宝藏之completion
- 7 【ART-PI】RT-Thread 开启RTC 与 Alarm组件
- 8 STM32的“GPU”——DMA2D实例详解
- 9 RTT串口V1版本的使用分析及问题排查指南（一）
- 10 国产MCU移植系列教程汇总，欢迎查看！

```
/* obtain image data */
void *raw = (void *)img_buffer;

int img_width = *(int *)&img_buffer[BITMAP_IMG_WIDTH_OFFSET];
int img_height = *(int *)&img_buffer[BITMAP_IMG_HEIGHT_OFFSET];
int img_offset = *(int *)&img_buffer[BITMAP_IMG_OFFSET];
int file_size = *(int *)&img_buffer[BITMAP_IMG_FILE_SIZE_OFFSET];

rt_kprintf("img_info: width=%d, height=%d, file_size=%d\n", img_width, img_height, file_size);

/* wrap image data */
zbar_image_t *image = zbar_image_create();
zbar_image_set_format(image, *(int*)"Y800");
zbar_image_set_size(image, img_width, img_height);
zbar_image_set_data(image, (void *)&img_buffer[img_offset], img_width * img_height, zbar_image_free_data
);

/* scan the image for barcodes */
int n = zbar_scan_image(rt_scanner, image);

/* extract results */
const zbar_symbol_t *symbol = zbar_image_first_symbol(image);
for(; symbol; symbol = zbar_symbol_next(symbol)) {
    /* do something useful with results */
    zbar_symbol_type_t typ = zbar_symbol_get_type(symbol);
    const char *data = zbar_symbol_get_data(symbol);
    rt_kprintf("decoded %s symbol \"%s\"\n",
        zbar_get_symbol_name(typ), data);
}

/* clean up */
zbar_image_destroy(image);
zbar_image_scanner_destroy(rt_scanner);
}
...
```

运行结果

来看下实测的识别效果, RT-Thread 提供了 finsh Shell, 可以很方便地使用命令行的方式运行写好的测试例程:

样本1(图片尺寸 256*256, 纯二维码):

![样本1]()

样本2(图片尺寸 480*360, 摄像头拍摄):

![样本2](https://i.loli.net/2019/06/29/5d16c5359d1c416291.png)/[md]

关注问题 我来回答 分享 收藏 感谢 举报

7 个回答 默认排序

wuhanstudio 2019-06-29

- 最新文章
- 1

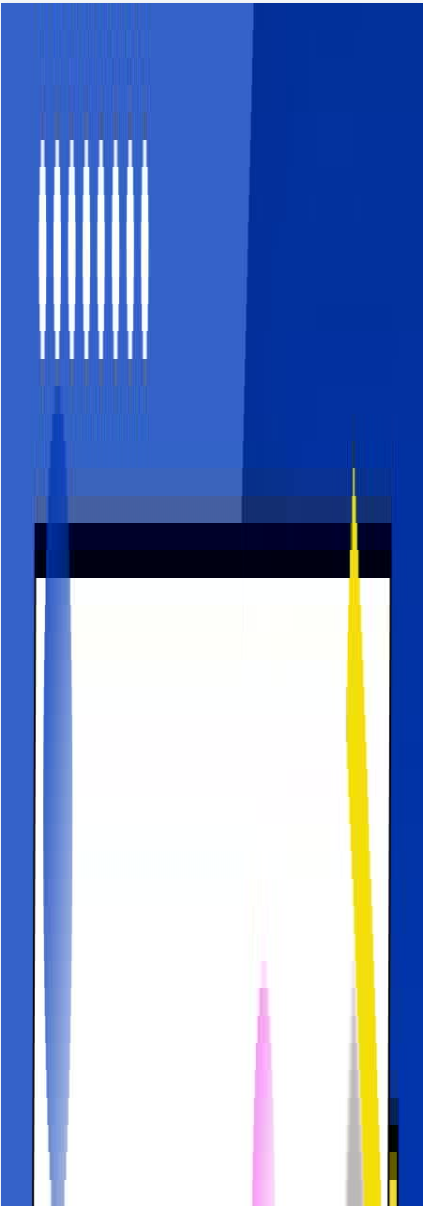
【NUC980】第2章 NUC980 IoT I2C实践
- 2

【N32G457】基于RT-Thread和N32G457的室内植物机
- 3

手把手教你基于RT-Thread Studio 使用STM32单片机的ADC外设
- 4

【N32G457】基于RT-Thread和N32G457的可控电流源
- 5

【开发板评测】Renesas RA6M4开发板之ADC、GPIO



登录

注册新账号

