

乐鑫Esp32学习之旅 安信可 ESP32–Cam 摄像头开发板二次开发 C SDK编程，拍照图片通过有线串口传到上位机PC端。（附带设备端+PC端源码）

原创

半颗心脏

于 2021-06-06 12:41:08 发布

13572

收藏

72

版权

分类专栏：

PyQT5


乐鑫Esp32学习之旅

文章标签：

ESP32-cam拍照

ESP32摄像头上位机

ESP32-camera 串口



PyQT5 同时被 2 个专栏收录 ▾

6 订阅

3 篇文章

订阅专栏

订阅专栏

- 本系列博客学习由非官方人员 [半颗心脏](#) 潜心所力所写，仅仅做个人技术交流分享，不做任何商业用途。如有不对之处，请留言，本人及时更改。

系列一：ESP32系列模组基础学习系列笔记

- 1、爬坑学习新旅程，虚拟机搭建esp32开发环境，打印“Hellow World”。
- 2、巧用eclipses编辑器，官方教程在在Windows下搭建esp32开发环境，打印“Hellow World”。
- 3、认识基本esp32的GPIO接口，开始点亮您的第一盏 LED和中断回调实现按键功能。
- 4、体会esp32的强大的定时器功能，实现定时2s闪烁一盏LED灯。
- 5、接触实践esp32的pwm宽度脉冲功能，实现呼吸效果闪烁一盏LED灯。
- 6、smartConfig和微信airKiss在esp32的实现，一键配网轻松快捷连接路由器。
- 7、利用GPIO中断做一个按键的短按和长按的回调事件，再也无须担心触发源。
- 8、esp32上实现本地 UDP 客户端和服务端角色，在局域网内实现通讯。
- 9、esp32上实现本地 TCP 客户端和服务端角色，可断线重连原路返回数据。
- 10、乐鑫esp32 SDK编程利用rmt驱动ws2812七彩灯，实现彩虹渐变效果。
- 11、入门 乐鑫esp-adf 音频框架开发，esp32造一个蓝牙耳机，实现切换歌曲，获取歌曲信息等功能。
- 12、开源一个微信公众号airkiss配网esp32以及局域网发现功能的工程，分享一个airkiss配网小工具。

- 13、esp32 内置 dns 服务器，无需外网访问域名返回指定网页。
- 14、esp32 sdk编程实现门户强制认证，连接esp32热点之后，自动强制弹出指定的登录界面。
- 15、认识本地离线语音唤醒识别框架 esp-skainet ，实现较低成本的硬件语音本地识别控制。
- 16、学习本地语音唤醒离线识别框架 esp-skainet ，如何修改唤醒词？如何自定义命令词？如何做意图动作？
- 17、全网首发，乐鑫esp32 sdk直连京东微联·小京鱼 · IoT开放平台，实现叮咚音响语音智能控制。
- 18、入门京东微联·小京鱼的控制面板H5开发，读懂vue语法，做自己的控制页面。
- 19、重磅开源，如何在微信小程序上ble蓝牙配网esp32，blufi的那些事！
- 20、一篇好文，开发过程中编译esp32固件太大，无法正常启动？教你如何自定义分区表partitions.csv。
- 21、 esp32蓝牙配网blufi的高度封装，集成简单、使用简单、容易上手，提高开发效率！
- 22、讨论下程序员“青春饭” 那些事，分享在esp32实现多种加密算法md5 |AES CBC-ECB| Sha1 | Sha256 等！
- 23、安信可 esp32-a1s 音频开发板移植最新 esp-adf 音频框架，小试牛刀如何实现在线文字转语音播放。
- 25、分享在 esp32 SDK实现冷暖光色温平滑调节的封装，轻松集成到您的项目去。
- 26、分享下如何在window下使用CMake编译，编译速度提高传统 make 编译一个档次，支持 ESP32 和 ESP32-C3。
- 27、windows10平台下自带的Linux安装 ESP8266/ESP32 环境，再无需额外安装虚拟机了。

系列二：ESP32-Camera 摄像头开发板系列笔记

- 1、安信可 ESP32-Cam 摄像头开发板二次开发 C SDK编程，实现MQTT远程拍照传输到私有服务器。
- 2、安信可 ESP32-Cam 摄像头开发板二次开发 C SDK编程，实现本地视频流监控。
- 3、安信可 ESP32-Cam 摄像头开发板二次开发 C SDK编程，拍照图片通过有线串口传到上位机PC端。

系列三：ESP32-C3 模组系列笔记

- 1、【蓝牙Mesh笔记 ①】ESP32-C3 模组上实现天猫精灵蓝牙 BLE Mesh AliGenie 接入，无需WiFi 连接也可以实现天猫精灵语音控制。

文章目录

前言

一、原理：

二、``JPEG``图片格式：

三、设计通信协议

3.1 获取图片

3.2 设备重启

四、设备代码

4.1 配置串口

4.2 处理数据并返回数据

五、上位机代码

六、效果

其他

另外，不要把我的博客作为学习标准，我的只是笔记，难有疏忽之处，如果有，请指出来，也欢迎留言哈！

前言

前面我在学 **PyQT5**，忽然有想法配合 **ESP32** -CAM做一个上位机的显示的小玩意，几经折腾，搞了出来，全部开源，现在分享给大家， 欢迎继续关注我的公众号。

人生阅历告诉我，做人不能好为人师，看到网上许多的有技术的文章，几乎是做营销一样，提着“授人以鱼，不如授之以渔”名声写技术文章，写着标题却没有文字的文章，不禁在想，这究竟怎么了。

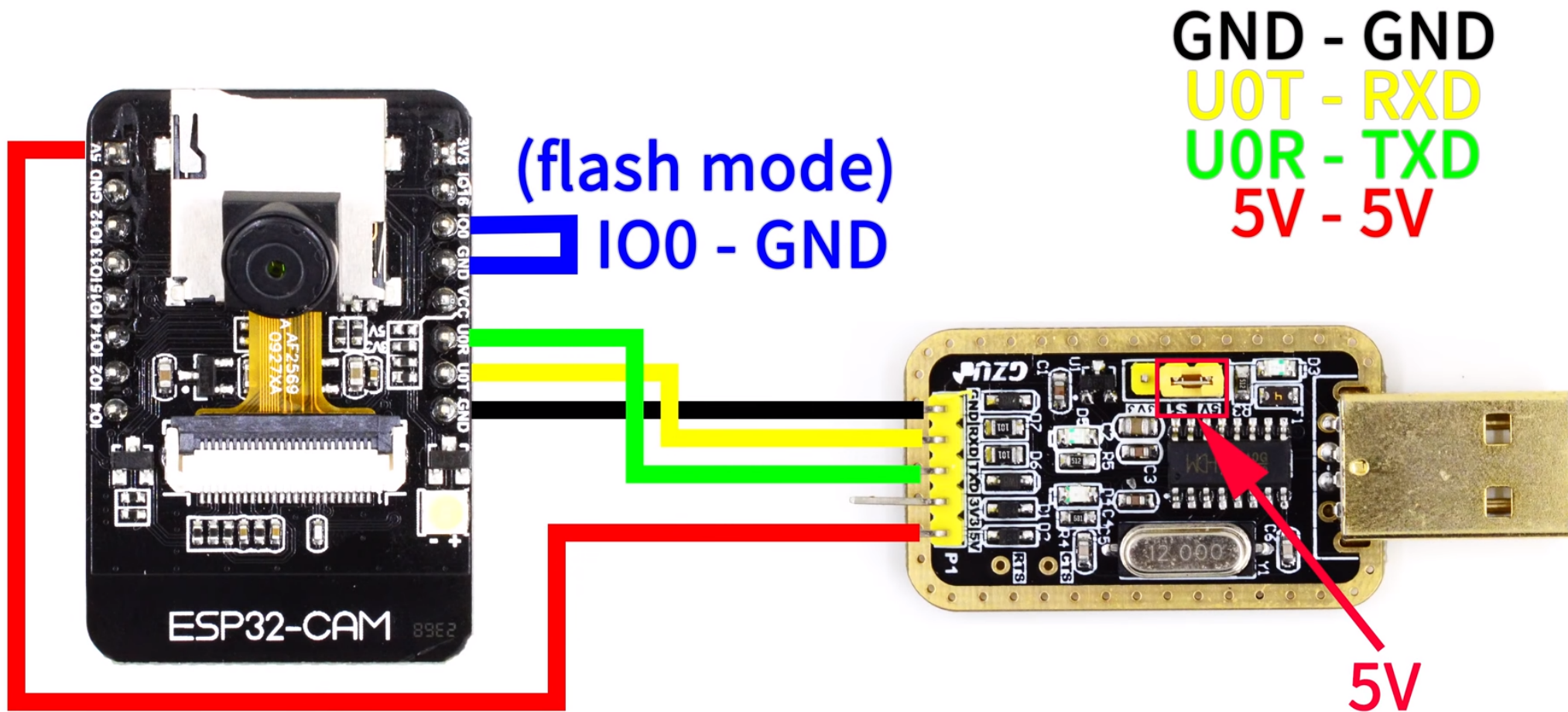
但我写文章不会这样的，应该以脚踏实地地写好每一篇技术文章，有思想地封装每个代码模块，解耦各个功能模块，变量方法类命名规则符合阿里巴巴 **JAVA手册**，这样是我所追求和推崇的；

越走越远，勿忘初心，从2016年开始写文章到如今，绝对不是为了做教案，更不是只为了零零几块钱而去逼迫自己学习，而是内心感受到敲代码所带来上的精神满足，那种修复 bug 废寝忘食的日日夜夜，正是我所追求的，当然了，这必然会带来财富的增益。

做人做事，要有长远眼光和大格局。

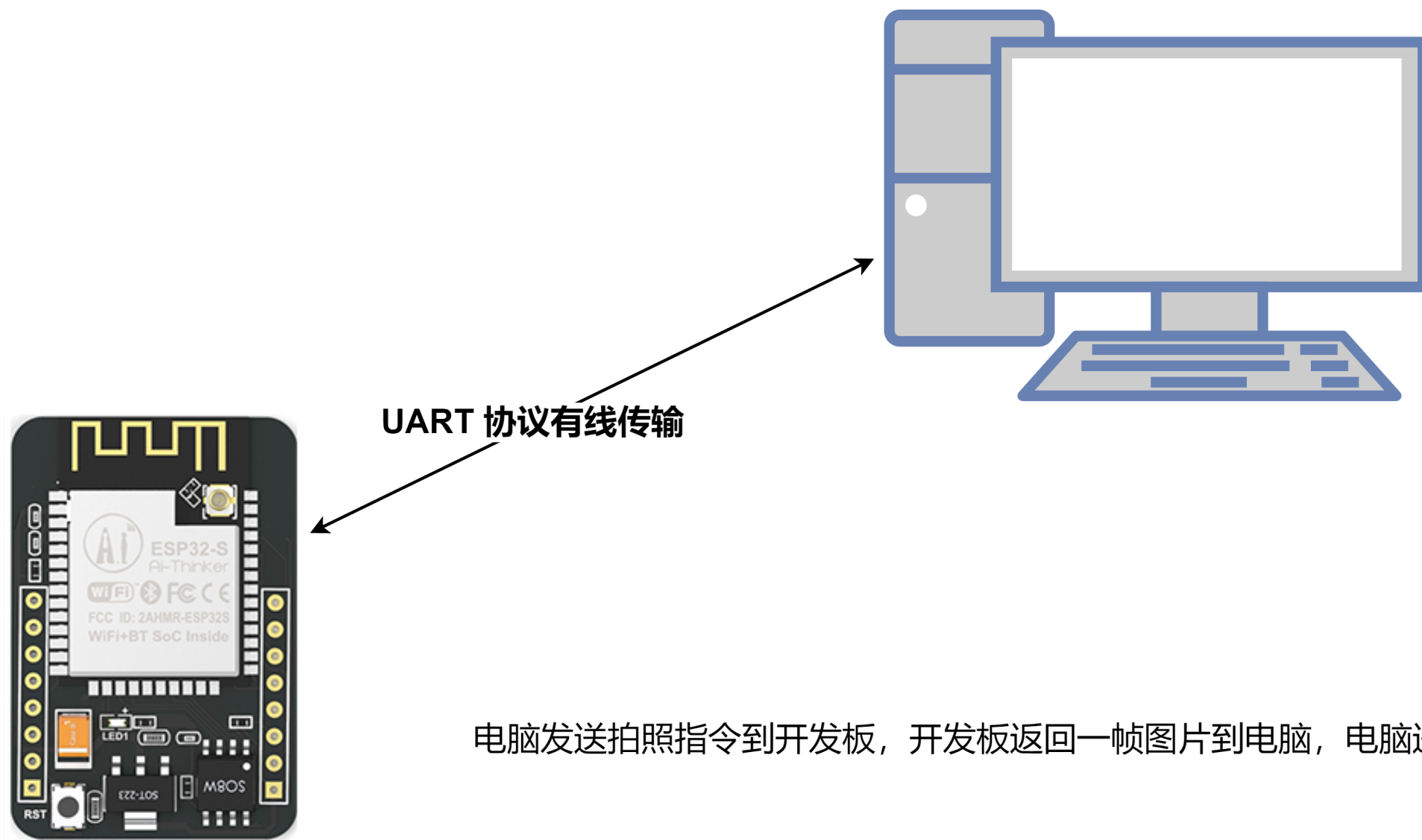
一、原理：

ESP32-CAM - CH340G



下载固件 的时候 ESP32-cam 的 io0 要接 GND，下载结束后，必须断开 io0 和 GND 的连接，再按下复位按键重启设备。ESP-CAM 接线下载固件图如上。

拍照显示的原理如下：



电脑发送拍照指令到开发板，开发板返回一帧图片到电脑，电脑进行解析。

二、JPEG 图片格式：

整个项目的难点是电脑端如何根据一端 **buff** 去解析出来一张图片，于是乎，不禁去翻阅资料：《[jpeg图片格式详解](#)》。

JPEG图片的二进制格式：

SOI（文件头）+APP0（图像识别信息）+ DQT（定义量化表）+ SOF0（图像基本信息）+ DHT（定义Huffman表） + DRI（定义重新开始间隔）+ SOS（扫描行开始）+ EOI（文件尾）

总的来说，我们拿到一段二进制图片，判断开头和结尾就好了。

描述	数值
开头两个字节：	0xFFD8
结尾两个字节：	0xFFD9

三、设计通信协议

为了更好调试，我使用了2个 **串口** ，一个串口用于通信，另外一个是为了看运行日志，接线：

开发板	USB-TTL	用途
U0TXD (GPIO1)	RXD	用于看运行日志
U0RXD (GPIO3)	TXD	用于看运行日志
GPIO2	TXD	串口通信
GPIO4	RXD	串口通信

- 波特率：115200

- 数据位： 8
- 停止位： 1
- 校验位： none
- 流控： 无

3.1 获取图片

电脑发送： `0xF0 0xA0`，模块返回一张图片的buff给到 **上位机**。

3.2 设备重启

电脑发送： `0xF0 0xA1`，模块2秒后重启。

四、设备代码

4.1 配置串口

```
1      uart_config_t uart_config = {
2          .baud_rate = 115200,
3          .data_bits = UART_DATA_8_BITS,
4          .parity = UART_PARITY_DISABLE,
5          .stop_bits = UART_STOP_BITS_1,
6          .flow_ctrl = UART_HW_FLOWCTRL_DISABLE
7      };
8      uart_param_config(UART_NUM_1, &uart_config);
9      // 配置串口1作为日志运行
10     uart_set_pin(UART_NUM_1, ECHO_LOG_TXD, ECHO_LOG_RXD, ECHO_TEST_RTS, ECHO_TEST_CTS);
11     uart_driver_install(UART_NUM_1, BUF_SIZE * 2, 0, 0, NULL, 0);
12     // 配置串口0作为通信
13     uart_param_config(UART_NUM_0, &uart_config);
14
```

```
15 |     uart_set_pin(UART_NUM_0, ECHO_TEST_TXD, ECHO_TEST_RXD, ECHO_TEST_RTS, ECHO_TEST_CTS);  
    uart_driver_install(UART_NUM_0, BUF_SIZE, 0, 0, NULL, 0);
```

4.2 处理数据并返回数据

我没有把处理收到的数据放在串口回调任务里面，而是另外开启了一个消息队列做处理。

```
1  /*  
2   * @Description: 解析下发数据的队列逻辑处理  
3   * @param: null  
4   * @return:  
5   */  
6  void Task_ParseJSON(void *pvParameters)  
7  {  
8      char data[] = {0xB0, 0, 0};  
9      size_t _jpg_buf_len;  
10     uint8_t *_jpg_buf;  
11     camera_fb_t *pic = NULL;  
12     esp_err_t res = ESP_OK;  
13     size_t fb_len = 0;  
14     while (1)  
15     {  
16         struct __User_data *pMqttMsg;  
17         xQueueReceive(ParseJSONQueueHandler, &pMqttMsg, portMAX_DELAY);  
18         printf("Task_ParseJSON_Message xQueueReceive get [%s] ... \n", pMqttMsg->allData);  
19         首先整体判断是否为一个json格式的数据  
20         cJSON *pJsonRoot = cJSON_Parse(pMqttMsg->allData);  
21         // 如果是否json格式数据  
22         if (pJsonRoot == NULL)  
23         {  
24             printf("[SY] Task_ParseJSON_Message xQueueReceive not json ... \n");  
25             data[1] = 0xA0;  
26             goto __cJSON_Delete;
```



```
27     }
28
29     cJSON *pJSON_Item_cmd = cJSON_GetObjectItem(pJsonRoot, "cmd");
30
31     if (strcmp(pJSON_Item_cmd->valuelstring, "Reboot") == 0)
32     {
33         data[2] = data[1] + 0xB0;
34         uart_write_bytes(UART_NUM_1, (const char *)data, 3);
35         ESP_LOGE(TAG, "reboot after 2s ...");
36         // 延迟2秒后重启设备
37         vTaskDelay(2000 / portTICK_PERIOD_MS);
38         esp_restart();
39     }
40     else if (strcmp(pJSON_Item_cmd->valuelstring, "Get Picture") == 0)
41     {
42
43         // 开始拍照
44         pic = esp_camera_fb_get();
45
46         if (!pic)
47         {
48             ESP_LOGE(TAG, "Camera capture failed");
49         }
50         else
51         {
52             // 拍照成功, 获取其大小、尺寸等信息
53             ESP_LOGI(TAG, "Camera capture OK , Its size was: %zu bytes", pic->len);
54             ESP_LOGI(TAG, "Camera capture OK , Its width was: %d", pic->width);
55             ESP_LOGI(TAG, "Camera capture OK , Its height was: %d ", pic->height);
56
57             // 写入串口
58             uart_write_bytes(UART_NUM_1, (const char *)pic->buf, pic->len);
59             //vTaskDelay(200 / portTICK_RATE_MS);
60         }
```

```

61         esp_camera_fb_return(pic);
62
63     }
64
65     __cJSON_Delete:
66         cJSON_Delete(pJsonRoot);
67     }
68 }

```

五、上位机代码

因为设备发送一段图片的数据，上位机会分几次接受到，所以我们代码上要处理拼接这些数据，然后再显示出来：

```

1     # 收到的数据
2     buff = (obj['data'])
3     # 收到的数据长度
4     length = obj['length']
5     # 拼接
6     picBuff.append(buff)
7     # 假如拼接之后的末端是 FF D9 ，则开始解析显示。
8     if buff[length - 1] == 217 and buff[length - 2] == 255:
9         print('图片 buff', (picBuff))
10        # 将bytes结果转化为字节流
11        bytes_stream = BytesIO(picBuff)
12        # 读取到图片
13        roiimg = Image.open(bytes_stream)
14        imgByteArr = BytesIO() # 初始化一个空字节流
15        roiimg.save(imgByteArr, format('PNG')) # 把我们得图片以'PNG'保存到空字节流
16        imgByteArr = imgByteArr.getvalue() # 无视指针，获取全部内容，类型由io流变成bytes。
17        img_name = '1.png'
18        with open(os.path.join('', img_name), 'wb') as f:
19

```

```
20         f.write(imgByteArr)
21     image = QPixmap(img_name)
22     ui.label.setPixmap(image)
    picBuff.clear()
```

六、效果

```
port = ui.cb_ports.currentText()
```

MainWindow

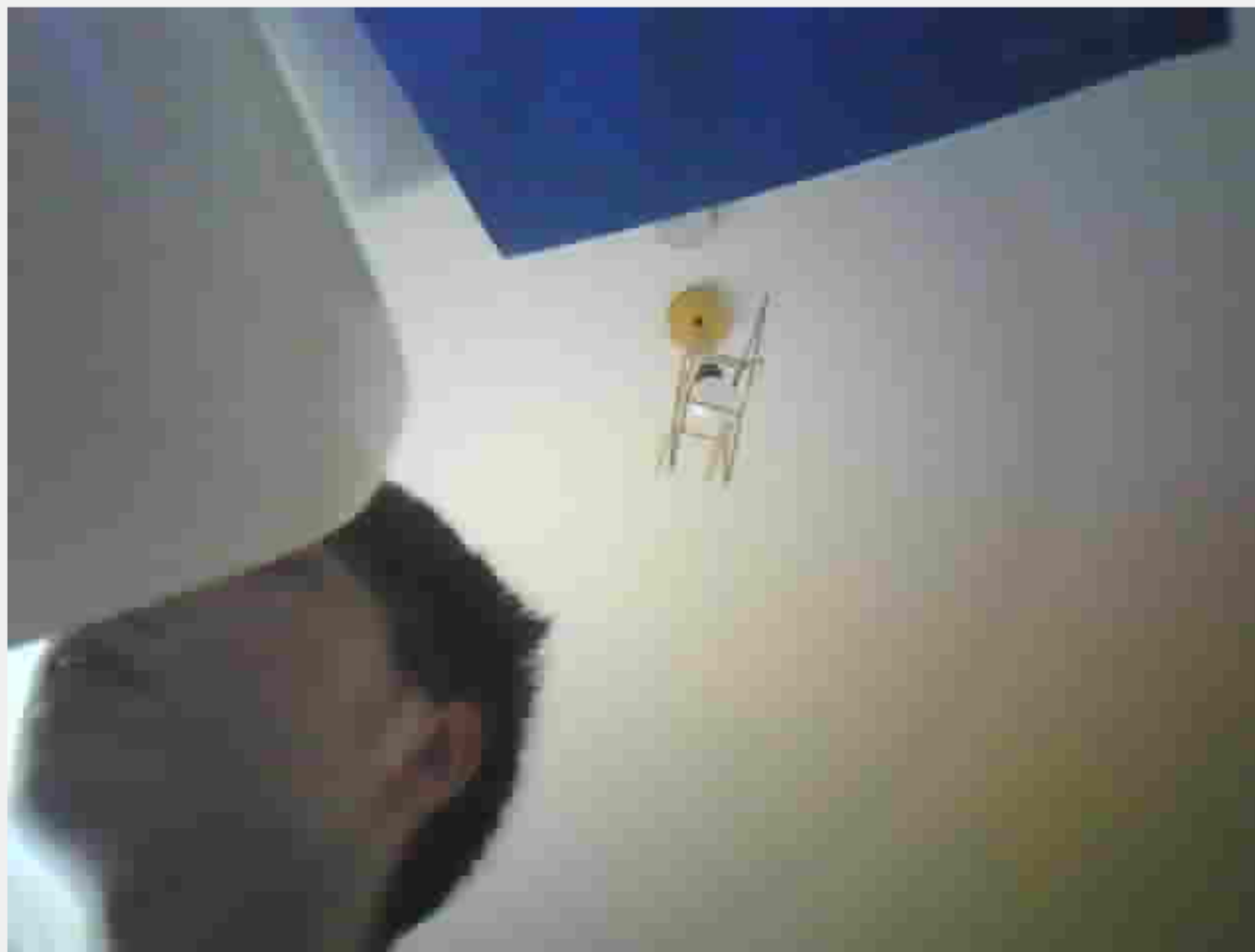
面板

串口: COM13

波特率: 115200

关闭串口

获取图片



<https://blog.csdn.net/zh370130248>

日志串口信息:

```
1 [0;32mI (493) sccb: pin_sda 26 pin_scl 27
2 [0m
3 [0;32mI (503) gpio: GPIO[32]| InputEn: 0| OutputEn: 1| OpenDrain: 0| Pullup: 0| Pulldown: 0| Intr:0 [0m
4 [0;32mI (653) camera: Detected 0V2640 camera[0m
5 [0;32mI (663) camera: Allocating 1 frame buffers (37 KB total)[0m
6 [0;32mI (663) camera: Allocating 37 KB frame buffer in OnBoard RAM[0m
7 [0;32mI (823) gpio: GPIO[4]| InputEn: 0| OutputEn: 1| OpenDrain: 0| Pullup: 0| Pulldown: 0| Intr:0 [0m
8 [SY] Task_ParseJSON_Message creat ...
9 Task_ParseJSON_Message xQueueReceive wait [217108] ...
10 Task_ParseJSON_Message xQueueReceive get [{"cmd":"Get Picture"}] ...
11 [0;32mI (11913) TAG:: Camera capture OK , Its size was: 7643 bytes[0m
12 [0;32mI (11913) TAG:: Camera capture OK , Its width was: 640[0m
13 [0;32mI (11913) TAG:: Camera capture OK , Its height was: 480 [0m
14 Task_ParseJSON_Message xQueueReceive wait [217060] ...
```

其他

- 关注本人微信公众号“徐宏blog”或扫描下方二维码，发送 **210507** 获取源码！

另外，不要把我的博客作为学习标准，我的只是笔记，难有疏忽之处，如果有，请指出来，也欢迎留言哈！

- 玩转 **esp8266** 带你飞、加群 **QQ** 群，不喜的朋友勿喷勿加：434878850
- 个人邮箱：xuhongv@yeah.net 24小时在线，有发必回复！
- esp8266源代码学习汇总（持续更新，欢迎star）：<https://github.com/xuhongv/StudyInEsp8266>
- esp32源代码学习汇总（持续更新，欢迎star）：<https://github.com/xuhongv/StudyInEsp32>
- 关注下面微信公众号二维码，干货多多，第一时间推送！

专注无线，热爱开源。



加入QQ群，干货不断推送。

QQ交流群：434878850

E-mail: yuhongy@yeah.net

