

优秀不够，你是否无可替代

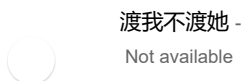
知识从未如此性感。烂程序员关心的是代码,好程序员关心的是数据结构和它们之间的关系 --QQ群: 607064330 --本人QQ:946029359 --淘宝 <https://shop411638453.taobao.com/>

随笔 - 766, 文章 - 0, 评论 - 321, 阅读 - 186万

导航

博客园
首页
新随笔
联系
订阅 
管理

公告

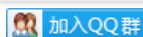


渡我不渡她 -

Not available

00:00 / 03:41

- 1 渡我不渡她
- 2 小镇姑娘
- 3 PDD洪荒之力



昵称：杨奉武
园龄：5年10个月
粉丝：649
关注：1
+加关注

搜索

我的标签

8266(88)
MQTT(50)
GPRS(33)
SDK(29)
Air202(28)
云服务器(21)
ESP8266(21)
Lua(18)
小程序(17)
STM32(16)
更多

随笔分类

Air724UG学习开发(5)
Android(22)
Android 开发(8)
C# 开发(4)
CH395Q学习开发(17)
CH579M物联网开发(4)
CH579M学习开发(7)
ESP32学习开发(18)
ESP8266 AT指令开发(基于STC89C52单片机)(3)
ESP8266 AT指令开发(基于STM32)(1)
ESP8266 AT指令开发基础入门篇备份(12)

202-ESP32_SDK开发-station模式配置模组连接路由器热点

<p> <iframe name="ifd" src="https://mnifdv.cn/resource/cnblogs/LearnESP32" frameborder="0" scrolling="auto" width="100%" height="1500"> </iframe> </p>

开源ESP32开发(源码见资料源码)

测试板链接:ESP32测试板链接

资料源码Git下载链接:<https://github.com/yangfengwu45/learn-esp32.git>

资料源码百度网

盘:<https://pan.baidu.com/s/10SBk0NsvLtJYHpDab9islg> 提取码：25oy

【点击加入乐鑫WiFi模组开发交流群】(群号822685419)https://jq.qq.com/?_wv=1027&k=fXgd3UOo

python虚拟机: [python-3.8.4-amd64.exe](#)

ESP-IDF工具安装器: [esp-idf-tools-setup-2.3.exe](#)

- [基础开源教程:ESP32开发\(arduino\)](#)
- [基础开源教程:ESP8266:LUA脚本开发](#)
- [基础开源教程:ESP8266 AT指令开发\(基于51单片机\)](#)
- [基础开源教程:Android学习开发](#)
- [基础开源教程:C#学习开发](#)
- [基础开源教程:微信小程序开发入门篇](#)
- 需要搭配的Android，C#等基础教程如上，各个教程正在整理。
- [000-ESP32开发板使用说明](#)
- [ESP32_SDK开发](#)
- [001-开发环境搭建\(Windows+VSCode\)](#)
- [002-测试网络摄像头\(OV2640\),实现远程视频监控\(花生壳http映射\)](#)
- [003-学习ESP32资料说明](#)
- [004-新建工程模板和创建新的文件](#)
- [005-新建工程补充-通过官方示例创建工程](#)
- [006-关于操作系统-任务,任务堆栈空间,任务的挂起,恢复,删除](#)
- [007-使用缓存管理传递数据](#)
- -----基本外设-----
- [101-ESP32管脚说明](#)
- [102-GPIO](#)
- [103-硬件定时器timer](#)
- [104-软件定时器esp_timer](#)
- [105-uart串口,485通信](#)
- [106-SPI](#)
- -----网络通信-----

ESP8266 LUA脚本语言开发(13)
 ESP8266 LUA开发基础入门篇备份(22)
 ESP8266 SDK开发(33)
 ESP8266 SDK开发基础入门篇备份(30)
 GPRS Air202 LUA开发(11)
 HC32F460(华大单片机)学习开发(5)
 NB-IOT Air302 AT指令和LUA脚本语言开发(27)
 PLC(三菱PLC)基础入门篇(2)
 STM32+Air724UG(4G模组)物联网开发(43)
 STM32+BC26/260Y物联网开发(37)
 STM32+CH395Q(以太网)物联网开发(24)
 STM32+ESP8266(ZLESP8266/物联网开发(1)
 STM32+ESP8266+AIR202/30:远程升级方案(16)
 STM32+ESP8266+AIR202/30:终端管理方案(6)
 STM32+ESP8266+Air302物联网开发(64)
 STM32+W5500+AIR202/302基本控制方案(25)
 STM32+W5500+AIR202/302远程升级方案(6)
 UCOSii操作系统(1)
 W5500 学习开发(8)
 编程语言C#(11)
 编程语言Lua脚本语言基础入门篇(6)
 编程语言Python(1)
 单片机(LPC1778)LPC1778(2)
 单片机(MSP430)开发基础入门篇(4)
 单片机(STC89C51)单片机开发板学习入门篇(3)
 单片机(STM32)基础入门篇(3)
 单片机(STM32)综合应用系列(16)
 电路模块使用说明(11)
 感想(6)
 更多

最新评论



























1. Re:102-CH579M学习开发-基本外设-串口IN4148是不是接反了呀?
--freemote
2. Re:006-STM32+ESP8266+AIR202/3基本控制篇(阿里云物联网平台在阿里云物联网平台上动态注册设备(基于STM32+ESP8266)你好,请问下您的这个项目使用的是阿里的LinkSdk吗
--码农29

阅读排行榜

1. ESP8266使用详解(AT,LUA,SDK)(173070)
2. 1-安装MQTT服务器(Windows),并连接测试(100144)
3. ESP8266刷AT固件与node mcu固件(65176)
4. 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(64907)

- [201-softAP模式配置模组发出的热点](#)
- [202-station模式配置模组连接路由器热点](#)
-














官方提供的例程在这里

	.git	2021
	.github	2021
	components	2021
	docs	2021
	examples	2021
	make	2021
	tools	2021
		2021
		2021
		2021
		2021
		2021
		2021
		2021
		2021
		2021
		2021
		2021
		2021
		2021
		2021
		2021
		2021
		2021
		2021
	README.md	2021/5

5. 有人WIFI模块使用详解(38710)
6. (一)基于阿里云的MQTT远程控制(Android 连接MQTT服务器,ESP8266连接MQTT服务器实现远程通信控制----简单的连接通信)(36137)
7. 关于TCP和MQTT之间的转换(33742)
8. C#中public与private与static(33119)
9. android 之TCP客户端编程(32133)
10. android服务端+eps8266+单片机+路由器之远程控制系统(31388)

推荐排行榜

1. C#委托+回调详解(9)
2. 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(8)
3. 用ESP8266+android,制作自己的WIFI小车(Android 软件)(6)
4. ESP8266使用详解(AT,LUA,SDK)(6)
5. 关于TCP和MQTT之间的转换(5)

	esp32-camera	2021/5/8 12
	espnw	2021/5/2 10
	fast_scan	2021/5/2 10
	getting_started	2021/5/2 10
	iperf	2021/5/2 10
	power_save	2021/5/2 10
	scan	2021/5/2 10
	simple_sniffer	2021/5/2 10
	smart_config	2021/5/2 10
	wpa2_enterprise	2021/5/2 10
	wps	2021/5/2 10
	liwentang12-esp32-camera-master.zip	2021/5/8 12
	README.md	2021/5/2 10

-  softAP
-  station

配置模块

连接名称为 QQQQQQ 密码为 11223344 的热点



```
#include <stdio.h>
#include <string.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/queue.h"
#include "freertos/event_groups.h"
#include "esp_system.h"
#include "esp_wifi.h"
#include "nvs_flash.h"
#include "esp_log.h"

#define EXAMPLE_ESP_WIFI_SSID      "QQQQQQ"
#define EXAMPLE_ESP_WIFI_PASS      "11223344"

static const char *TAG = "wifi station";

static void event_handler(void* arg, esp_event_base_t event_base,
                        int32_t event_id, void* event_data)
{
    if (event_base == WIFI_EVENT && event_id == WIFI_EVENT_STA_START) { //配置
        esp_wifi_connect(); //连接热点
    } else if (event_base == WIFI_EVENT && event_id == WIFI_EVENT_STA_DISCONNECTED) { //连接热点
        esp_wifi_connect(); //连接热点
        ESP_LOGI(TAG, "connect to the AP fail");
    } else if (event_base == IP_EVENT && event_id == IP_EVENT_STA_GOT_IP) { //
        ip_event_got_ip_t* event = (ip_event_got_ip_t*) event_data;
        ESP_LOGI(TAG, "got ip:" IPSTR, IP2STR(&event->ip_info.ip));
    }
}
```

```

}

void wifi_init_sta(void)
{
    ESP_ERROR_CHECK(esp_netif_init()); //初始化内部的lwip

    ESP_ERROR_CHECK(esp_event_loop_create_default()); //创建系统事件任务
    esp_netif_create_default_wifi_sta(); //创建有 TCP/IP 堆栈的默认网络接口实例绑定

    wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT();
    ESP_ERROR_CHECK(esp_wifi_init(&cfg)); //创建 Wi-Fi 驱动程序任务，并初始化 Wi-Fi

    esp_event_handler_instance_t instance_any_id;
    esp_event_handler_instance_t instance_got_ip;
    /*注册系统事件回调函数*/
    ESP_ERROR_CHECK(esp_event_handler_instance_register(WIFI_EVENT, //
                                                        ESP_EVENT_ANY_ID,
                                                        &event_handler,
                                                        NULL,
                                                        &instance_any_id));

    /*注册系统事件回调函数*/
    ESP_ERROR_CHECK(esp_event_handler_instance_register(IP_EVENT,
                                                        IP_EVENT_STA_GOT_IP,
                                                        &event_handler,
                                                        NULL,
                                                        &instance_got_ip));

    /*配置连接的热点参数*/
    wifi_config_t wifi_config = {
        .sta = {
            .ssid = EXAMPLE_ESP_WIFI_SSID,
            .password = EXAMPLE_ESP_WIFI_PASS,
            .threshold.authmode = WIFI_AUTH_WPA2_PSK, //加密方式
            /*配置pmf,当前最新加密技术*/
            .pmf_cfg = {
                .capable = true, //告诉热点这边有能力使用PMF进行加密通信(防止窃听窃密)
                .required = false //告诉热点这边不强制要求使用PMF进行加密通信(防止窃听窃密)
            },
        },
    };

    ESP_ERROR_CHECK(esp_wifi_set_mode(WIFI_MODE_STA) ); //设置STA模式
    ESP_ERROR_CHECK(esp_wifi_set_config(ESP_IF_WIFI_STA, &wifi_config) ); //配置wifi
    ESP_ERROR_CHECK(esp_wifi_start() ); //启动

    /* 取消注册事件回调 */
    //ESP_ERROR_CHECK(esp_event_handler_instance_unregister(IP_EVENT, IP_EVENT_STA_GOT_IP, &instance_got_ip));
    //ESP_ERROR_CHECK(esp_event_handler_instance_unregister(WIFI_EVENT, ESP_EVENT_ANY_ID, &instance_any_id));
}

void app_main(void)
{
    //初始化 NVS (配置WiFi的参数存储需要用到NVS)
    esp_err_t ret = nvs_flash_init();
    if (ret == ESP_ERR_NVS_NO_FREE_PAGES || ret == ESP_ERR_NVS_NEW_VERSION_FOUND) {
        ESP_ERROR_CHECK(nvs_flash_erase());
        ret = nvs_flash_init();
    }
    ESP_ERROR_CHECK(ret);

    ESP_LOGI(TAG, "ESP_WIFI_MODE_STA");
}

```

```
wifi_init_sta();  
}
```

下面这个地方是标准的流程

```
42  
43 void wifi_init_sta(void)  
44 {  
45     ESP_ERROR_CHECK(esp_netif_init()); //初始化内部的lwip  
46  
47     ESP_ERROR_CHECK(esp_event_loop_create_default()); //创建系统事件任务  
48     esp_netif_create_default_wifi_sta(); //创建有 TCP/IP 堆栈的默认网络接口实例绑定 station 或 AP。  
49  
50     wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT();  
51     ESP_ERROR_CHECK(esp_wifi_init(&cfg)); //创建 Wi-Fi 驱动程序任务，并初始化 Wi-Fi 驱动程序。  
52  
53     esp_event_handler_instance_t instance_any_id;  
54     esp_event_handler_instance_t instance_got_ip;  
55     /*注册系统事件回调函数*/  
56     ESP_ERROR_CHECK(esp_event_handler_instance_register(WIFI_EVENT, //wifi状态改变事件  
57                                                         ESP_EVENT_ANY_ID,  
58                                                         &event_handler,  
59                                                         NULL,  
60                                                         &instance_any_id));  
61  
62     /*注册系统事件回调函数*/  
63     ESP_ERROR_CHECK(esp_event_handler_instance_register(IP_EVENT, //IP地址改变事件  
64                                                         IP_EVENT_STA_GOT_IP,  
65                                                         &event_handler,  
66                                                         NULL,  
67                                                         &instance_got_ip));  
68 }
```

https://docs.espressif.com/projects/esp-idf/zh_CN/latest/esp32/api-guides/wifi.html?highlight=esp_netif_init

1. Wi-Fi/LwIP 初始化阶段

- s1.1：主任务通过调用函数 `esp_netif_init()` 创建一个 LwIP 核心任务，并初始化 LwIP 相关工作。
- s1.2：主任务通过调用函数 `esp_event_loop_create()` 创建一个系统事件任务，并初始化应用程序事件的回调函数。在此情况下，该回调函数唯一的动作就是将事件中继到应用程序任务中。
- s1.3：主任务通过调用函数 `esp_netif_create_default_wifi_ap()` 或 `esp_netif_create_default_wifi_sta()` 创建有 TCP/IP 堆栈的默认网络接口实例绑定 station 或 AP。
- s1.4：主任务通过调用函数 `esp_wifi_init()` 创建 Wi-Fi 驱动程序任务，并初始化 Wi-Fi 驱动程序。
- s1.5：主任务通过调用 OS API 创建应用程序任务。

推荐按照 s1.1 ~ s1.5 的步骤顺序针对基于 Wi-Fi/LwIP 的应用程序进行初始化。但这一顺序 **并非** 强制，您可以在第 s1.1 步创建应用程序任务，然后在应用程序任务中进行所有其它初始化操作。不过，如果您的应用程序任务依赖套接字，那么在初始化阶段创建应用程序任务可能并不适用。此时，您可以在接收到 IP 后再进行任务创建。

2. Wi-Fi 配置阶段

Wi-Fi 驱动程序初始化成功后，可以进入到配置阶段。该场景下，Wi-Fi 驱动程序处于 station 模式。因此，首先您需调用函数 `esp_wifi_set_mode()` (`WIFI_MODE_STA`) 将 Wi-Fi 模式配置为 station 模式。可通过调用其它 `esp_wifi_set_xxx` API 进行更多设置，例如：协议模式、国家代码、带宽等。请参阅 [ESP32 Wi-Fi 配置](#)。

测试

改为自己的路由器名称和密码

```
main > C:\hello_world_main.c > wifi_init_sta(void)

10 #include <stdio.h>
11 #include <string.h>
12 #include "freertos/FreeRTOS.h"
13 #include "freertos/task.h"
14 #include "freertos/queue.h"
15 #include "freertos/event_groups.h"
16 #include "esp_system.h"
17 #include "esp_wifi.h"
18 #include "nvs_flash.h"
19 #include "esp_log.h"
20
21
22 #define EXAMPLE_ESP_WIFI_SSID "QQQQQQ" //连接的热点名称
23 #define EXAMPLE_ESP_WIFI_PASS "11223344" //密码
24
25
26 static const char *TAG = "wifi station";
27
28
```

下载到开发板测试

```
PROBLEMS 输出 终端 调试控制台

I (8076) wifi station: got ip:192.168.43.156
I (11696) wifi:state: run -> init (3a0)
I (11696) wifi:pm stop, total sleep time: 4036781 us / 4578669 us

I (11696) wifi:new:<1,0>, old:<1,0>, ap:<255,255>, sta:<1,0>, prof:11
I (11706) wifi station: connect to the AP fail
I (13756) wifi station: connect to the AP fail
I (15806) wifi station: connect to the AP fail
I (17846) wifi station: connect to the AP fail
I (19896) wifi station: connect to the AP fail
I (20016) wifi:new:<1,0>, old:<1,0>, ap:<255,255>, sta:<1,0>, prof:11
I (20016) wifi:state: init -> auth (b0)
I (20026) wifi:state: auth -> assoc (0)
I (20026) wifi:state: assoc -> run (10)
I (20046) wifi:connected with QQQQQQ, aid = 1, channel 1, BW20, bssid = 94:63:72:55:c1:7d
I (20046) wifi:security: WPA2-PSK, phy: bgn, rssi: -39
I (20056) wifi:pm start, type: 1

I (20086) wifi:AP's beacon interval = 102400 us, DTIM period = 2
I (21076) esp_netif_handlers: sta ip: 192.168.43.156, mask: 255.255.255.0, gw: 192.168.43.86
I (21076) wifi station: got ip:192.168.43.156
[]
```

连接上热点,并获得IP

分类: ESP32学习开发

好文要顶

关注我

收藏该文







杨奉武
关注 - 1
粉丝 - 649
[+加关注](#)

0

0

« 上一篇: [201-ESP32_SDK开发-softAP模式配置模组发出的热点](#)

posted on 2021-09-03 01:15 杨奉武 阅读(0) 评论(0) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) 博客园首页

【推荐】阿里云云大使特惠: 新用户购ECS服务器1核2G最低价87元/年

【推荐】大型组态、工控、仿真、CAD\GIS 50万行VC++源码免费下载!

【推荐】和开发者在一起: 华为开发者社区, 入驻博客园科技品牌专区

【推广】园子与爱卡汽车爱宝险合作, 随手就可以买一份的百万医疗保险



编辑推荐:

- 浅谈 C# 更改令牌 ChangeToken
- CNN卷积神经网络详解
- 记一次 .NET 某流媒体独角兽 API 句柄泄漏分析
- 流量录制与回放技术实践
- 熟悉而陌生的新朋友——IAsyncDisposable

最新新闻:

- 苹果今日官宣! 音视频阅读类APP可使用三方支付, 游戏未松口 (2021-09-03 00:02)
 - “食盐” 电池登上Nature, 终极目标: 让电动车续航里程×6 (2021-09-02 23:46)
 - 张一鸣的头盔需要库克激活 (2021-09-02 23:30)
 - 8月蔚理鹏销量分析: 集体破万倒计时 (2021-09-02 23:15)
 - 爆火的直播电商, 怎么跨境后 “水土不服” 了? (2021-09-02 22:53)
- » 更多新闻...

历史上的今天:

- 2016-09-03 当年参加飞思卡尔自己写的双线识别算法
- 2016-09-03 TTL,COMS,USB,232,422,485电平之详细介绍及使用

Powered by:

博客园

Copyright © 2021 杨奉武

Powered by .NET 5.0 on Kubernetes



单片机,物联网,上位机,...

扫一扫二维码, 入群聊。