

## 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 - 818, 文章 - 0, 评论 - 327, 阅读 - 201万

### 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

- 渡我不渡她
- 小镇姑娘
- PDD洪荒之力

加入QQ群

昵称：杨奉武  
 园龄：6年2个月  
 粉丝：693  
 关注：1

### 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\(8\)](#)  
[Android\(22\)](#)  
[Android 开发\(8\)](#)  
[C# 开发\(4\)](#)  
[CH395Q学习开发\(17\)](#)  
[CH573F学习开发\(1\)](#)  
[CH579M物联网开发\(12\)](#)  
[CH579M学习开发\(8\)](#)  
[ESP32学习开发\(29\)](#)  
[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\(12\)](#)  
[ESP8266 LUA脚本语言开发\(13\)](#)

## 207-ESP32\_SDK开发-UDP传输(select方式)

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

### ESP32\_SDK开发(源码见资料源码)

#### 开发板链接:[ESP32开发板链接](#)

#### 资料源码Git下载链

接:<https://github.com/yangfengwu45/learn-esp32.git>

#### 资料源码百度网盘

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](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
- 107-flash数据存储nvs
- 网络通信-----
- 
- 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
- 203-softAP+station共存模式
- 204-TCP服务器(模组AP热点模式,支持多个客户端连

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)物联网  
开发(9)  
HC32F460(华大单片机)学习开  
发(8)  
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(ZLESP826  
6A)物联网开发(1)  
STM32+ESP8266+AIR202/3  
02远程升级方案(16)  
STM32+ESP8266+AIR202/3  
02终端管理方案(6)  
STM32+ESP8266+Air302物  
联网开发(65)  
STM32+W5500+AIR202/30  
2基本控制方案(25)  
STM32+W5500+AIR202/30  
2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)  
更多

#### 阅读排行榜

1. ESP8266使用详解(AT,LUA, SDK)(174336)
2. 1-安装MQTT服务器(Windo ws),并连接测试(105291)
3. 用ESP8266+android,制作 自己的WIFI小车(ESP8266篇) (67911)
4. ESP8266刷AT固件与node mcu固件(66961)
5. 有人WIFI模块使用详解(394 49)
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)(37105)
7. C#中public与private与stat ic(36017)
8. 关于TCP和MQTT之间的转 换(35336)
9. android 之TCP客户端编程 (33052)
10. android服务端+eps8266 +单片机+路由器之远程控制系 统(31709)

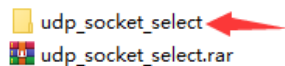
#### 推荐排行榜

接通信)(废弃,项目绝对不能)

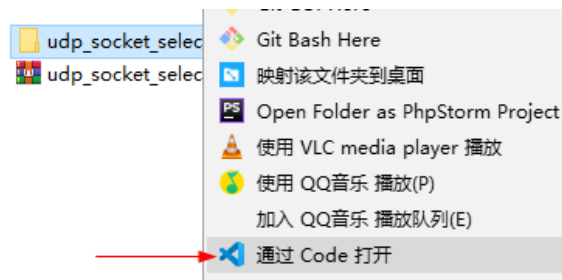
- [205-TCP服务器\(select方式,支持多连接,高速高并发 传输\)](#)
- [206-TCP客户端\(select方式,自动重连\)](#)
- [207-UDP传输\(select方式\)](#)
- -----图片传输-----  
---
- [801-ESP32\(WiFi\)把采集的摄像头照片数据通过串口 输出到串口上位机显示\(C# 串口上位机\)](#)
- [802-ESP32\(WiFi\)把采集的摄像头照片数据通过UDP 发送给UDP客户端\(C# UDP客户端\)](#)
- [803-手机连接ESP32的热点,使用微信小程序查看摄像 头图像\(WiFi视频小车,局域网视频监控\)](#)
- [804-手机连接ESP32的热点,使用android APP查看摄 像头图像\(WiFi视频小车,局域网视频监控\)](#)
- 
- [901-动手制作自己的WiFi视频遥控小车\(android\)](#)
- 

## 下载程序到开发板

### 1.把这节的代码放到英文目录



### 2.鼠标右键选择使用VScode打开

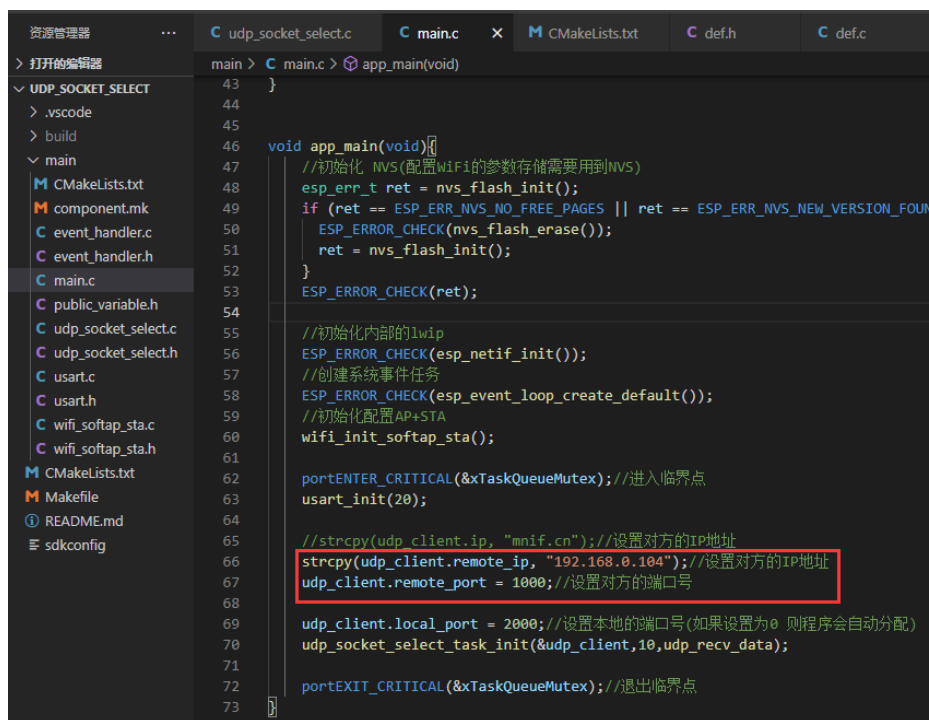


### 3.设置对方的IP地址和端口号

1. 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(9)
2. C#委托+回调详解(9)
3. 用ESP8266+android,制作自己的WIFI小车(Android 软件)(6)
4. 我的大学四年(6)
5. ESP8266使用详解(AT,LUA,SDK)(6)

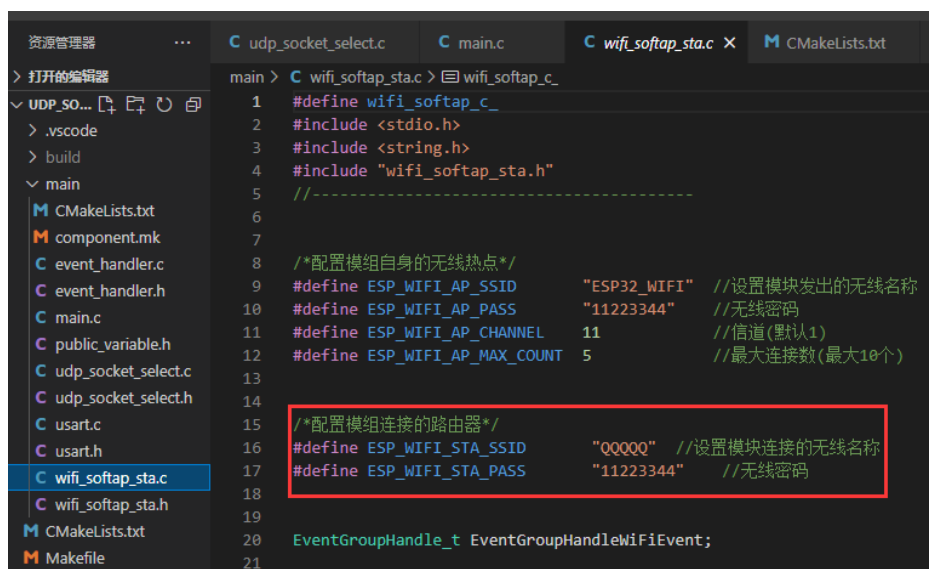
## 最新评论

1. Re:2-6-1-视频传输,监控,直播方案-手机连接ESP32的热点,使用微信小程序查看摄像头图像(WiFi视频小车,局域网视频监控)  
赞赞赞,感谢大佬无私奉献  
--SJA2C2A
2. Re:中移动M5311模块使用手册(TCP,MQTT)  
请问你用的usb转ttl是哪一种呢,我用的ch340可是开机串口助手没有SIM识别显示  
--夏洛的网娅



```
资源管理器  ...  C udp_socket_select.c  C main.c  x  CMakeLists.txt  C def.h  C def.c
> 打开的编辑器  main > C main.c > app_main(void)
43 }
44
45
46 void app_main(void){
47     //初始化 NVS(配置WiFi的参数存储需要用到NVS)
48     esp_err_t ret = nvs_flash_init();
49     if (ret == ESP_ERR_NVS_NO_FREE_PAGES || ret == ESP_ERR_NVS_NEW_VERSION_FOUND)
50         ESP_ERROR_CHECK(nvs_flash_erase());
51     ret = nvs_flash_init();
52 }
53 ESP_ERROR_CHECK(ret);
54
55 //初始化内部的lwip
56 ESP_ERROR_CHECK(esp_netif_init());
57 //创建系统事件任务
58 ESP_ERROR_CHECK(esp_event_loop_create_default());
59 //初始化配置AP+STA
60 wifi_init_softap_sta();
61
62 portENTER_CRITICAL(&TaskQueueMutex); //进入临界点
63 usart_init(20);
64
65 //strcpy(udp_client.ip, "mnif.cn"); //设置对方的IP地址
66 strcpy(udp_client.remote_ip, "192.168.0.104"); //设置对方的IP地址
67 udp_client.remote_port = 1000; //设置对方的端口号
68
69 udp_client.local_port = 2000; //设置本地的端口号(如果设置为0 则程序会自动分配)
70 udp_socket_select_task_init(&udp_client, 10, udp_recv_data);
71
72 portEXIT_CRITICAL(&TaskQueueMutex); //退出临界点
73 }
```

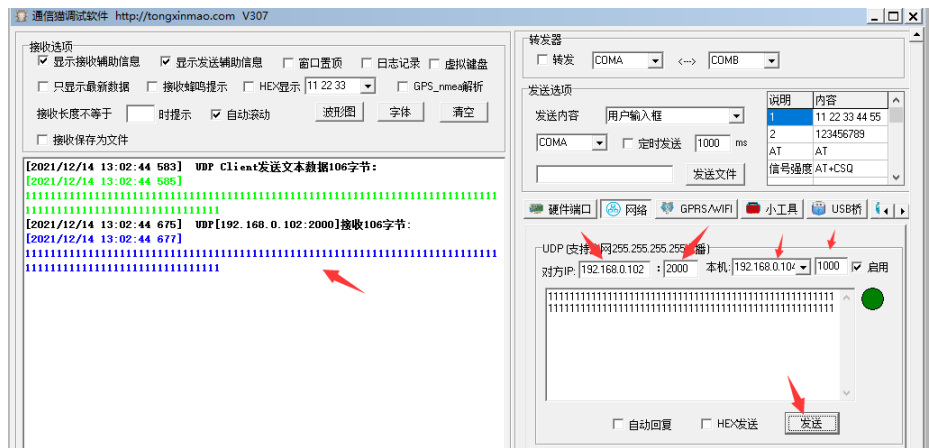
## 4.注意,如果设置的地址是域名,因为需要DNS把域名解析成IP,所以需要设置一下连接的路由器



```
资源管理器  ...  C udp_socket_select.c  C main.c  C wifi_softap_sta.c  x  CMakeLists.txt
> 打开的编辑器  main > C wifi_softap_sta.c > wifi_softap_c_
1 #define wifi_softap_c_
2 #include <stdio.h>
3 #include <string.h>
4 #include "wifi_softap_sta.h"
5 //-----
6
7 /*配置模组自身的无线热点*/
8 #define ESP_WIFI_AP_SSID "ESP32_WIFI" //设置模块发出的无线名称
9 #define ESP_WIFI_AP_PASS "11223344" //无线密码
10 #define ESP_WIFI_AP_CHANNEL 11 //信道(默认1)
11 #define ESP_WIFI_AP_MAX_COUNT 5 //最大连接数(最大10个)
12
13 /*配置模组连接的路由器*/
14 #define ESP_WIFI_STA_SSID "QQQQQ" //设置模块连接的无线名称
15 #define ESP_WIFI_STA_PASS "11223344" //无线密码
16
17 EventGroupHandle_t EventGroupHandleWifiEvent;
18
19
20
21
```

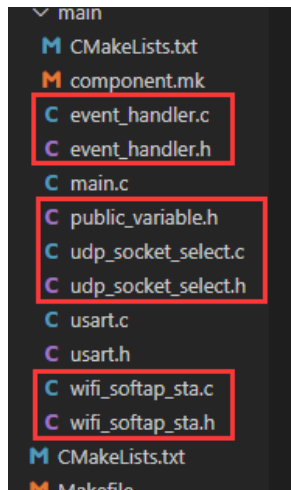
## 5.下载测试

现在是发送给UDP什么,UDP就回复什么



## 程序使用说明(先说下如何使用)

1.如果用户需要移植使用的话直接把下面的文件放到自己的工程里面就可以



2.配置一下

```
资源管理器  ...  C udp_socket_select.c  C main.c  x sdkconfig  M CMakeLists.txt  C def.h  C def.c  C sockets.h  C inet
> 打开的编辑器
  > udp_so... [?] [?] [?]
  > .vscode
  > build
  > main
  M CMakeLists.txt
  M component.mk
  C event_handler.c
  C event_handler.h
  C main.c
  C public_variable.h
  C udp_socket_select.c
  C udp_socket_select.h
  C usart.c
  C usart.h
  C wifi_softap_sta.c
  C wifi_softap_sta.h
  M CMakeLists.txt
  M Makefile
  ① README.md
  ② sdkconfig

main > C main.c > app_main(void)
11 #include "udp_socket_select.h"
12 #include "esp_log.h"
13
14 #include "nvs_flash.h"
15 #include "wifi_softap_sta.h"
16 #include "usart.h"
17 #include "udp_socket_select.h"
18
19 PRIVILEGED_DATA static portMUX_TYPE xTaskQueueMutex = portMUX_INITIALIZER_UNLOCKED;
20
21 pv_udp_socket_select_struct_t udp_client; //udp 结构体变量
22
23
24 //udp接收数据回调函数
25 void udp_recv_data(char *data, int len){
26     //获取对方地址,使用
27     char addr_str[128];
28     inet_ntoa_r(((struct sockaddr_in *)&udp_client.sockaddr)->sin_addr.s_addr, addr_str, sizeof(addr_str) - 1);
29     //获取对方端口号 (udp_client.sockaddr.sin_port 由于系统数据大小端问题,需要做下转换)
30     uint16_t port = ntohs(udp_client.sockaddr.sin_port);
31     //打印对方的IP地址和端口号
32     ESP_LOGE("main", "udp socket addr=%s %d\r\n", addr_str, port);
33
34     // **发送数据给UDP**
35     // struct sockaddr_in dest_addr;
36     // dest_addr.sin_addr.s_addr = inet_addr("192.168.0.104");//对方地址
37     // dest_addr.sin_family = AF_INET;
38     // dest_addr.sin_port = htons(1000);//对方端口号
39     // sendto(udp_client.socket_fd, data, len, 0, (struct sockaddr*)&dest_addr, sizeof(dest_addr));
40
41     /*返回数据给UDP*/
42     sendto(udp_client.socket_fd, data, len, 0, (struct sockaddr*)&udp_client.sockaddr, sizeof(udp_client.sockaddr));
43 }
44
45
46 void app_main(void){
47     //初始化 NVS(配置WiFi的参数存储需要用到NVS)
48     esp_err_t ret = nvs_flash_init();
```

```
main
M CMakeLists.txt
M component.mk
C event_handler.c
C event_handler.h
C main.c
C public_variable.h
C udp_socket_select.c
C udp_socket_select.h
C usart.c
C usart.h
C wifi_softap_sta.c
C wifi_softap_sta.h
M CMakeLists.txt
M Makefile
① README.md
② sdkconfig

42 sendto(udp_client.socket_fd, data, len, 0, (struct sockaddr*)&udp_client.sockaddr, sizeof(udp_client.sockaddr));
43 }
44
45 void app_main(void){
46     //初始化 NVS(配置WiFi的参数存储需要用到NVS)
47     esp_err_t ret = nvs_flash_init();
48     if (ret == ESP_ERR_NVS_NO_FREE_PAGES || ret == ESP_ERR_NVS_NEW_VERSION_FOUND) {
49         ESP_ERROR_CHECK(nvs_flash_erase());
50         ret = nvs_flash_init();
51     }
52     ESP_ERROR_CHECK(ret);
53
54     //初始化内部的lwip
55     ESP_ERROR_CHECK(esp_netif_init());
56     //创建系统事件任务
57     ESP_ERROR_CHECK(esp_event_loop_create_default());
58     //初始化配置AP+STA
59     wifi_init_softap_sta();
60
61     portENTER_CRITICAL(&xTaskQueueMutex); //进入临界点
62     usart_init(20);
63
64     //strcpy(udp_client.ip, "mnif.cn");//设置对方的IP地址
65     strcpy(udp_client.remote_ip, "192.168.0.104");//设置对方的IP地址
66     udp_client.remote_port = 1000;//设置对方的端口号
67
68     udp_client.local_port = 2000;//设置本地的端口号(如果设置为0 则程序会自动分配)
69     udp_socket_select_task_init(&udp_client, 10, udp_recv_data);
70
71     portEXIT_CRITICAL(&xTaskQueueMutex); //退出临界点
72
73 }
```

### 3.关于 udp\_socket\_select\_task\_init 函数

```
65 //strcpy(udp_client.ip, "mnif.cn");//设置对方的IP地址
66 strcpy(udp_client.remote_ip, "192.168.0.104");//设置对方的IP地址
67 udp_client.remote_port = 1000;//设置对方的端口号
68
69 udp_client.local_port = 2000;//设置本地的端口号(如果设置为0 则程序会自动分配)
70 udp_socket_select_task_init(&udp_client, 10, udp_recv_data);
71
72 portEXIT_CRITICAL(&xTaskQueueMutex); //退出临界点
73 }
```

```

130
131 /**
132  * @brief select UDP
133  * @param pv_udp_socket_select_value 客户端结构体
134  * @param task_priority 任务优先级
135  * @param RecvDataCallback 接收回调函数
136  * @retval none
137  */
138 void udp_socket_select_task_init(pv_udp_socket_select_struct_t *pv_udp_socket_select_value, int task_priority, void(*RecvDataCallback)(char *data, int len)){
139     pv_udp_socket_select_value->socket_fd = 1;
140     pv_udp_socket_select_value->callback = RecvDataCallback;
141     //创建TCP任务
142     xTaskCreate(udp_socket_select_task, "tcp_client_select_task", 4096, pv_udp_socket_select_value, task_priority, NULL);
143 }
144

```

## 4.接收和发送数据

如果在接收回调函数里面发送直接调用下面的函数就可以

第一个参数 tcp\_client.socket\_fd 固定,后面分别是发送的数据和数据长度,再后面的也是固定的

```

资源管理器  ...  C udp_socket_select.c  C main.c  x sdkconfig  M CMakeLists.txt  C def.h  C def.c  C sockets.h  C inet.h
打开的编辑器
main > C main.c > app_main(void)
udp_socket_select.c
> vscode
> build
main
CMakeLists.txt
component.mk
event_handler.c
event_handler.h
main.c
public_variable.h
udp_socket_select.c
udp_socket_select.h
usart.c
usart.h
wifi_softap_sta.c
wifi_softap_sta.h
CMakeLists.txt
Makefile
README.md
sdkconfig
19 #include "esp_system.h"
20 #include "esp_wifi.h"
21 #include "esp_netif.h"
22 #include "esp_log.h"
23
24 #include "nvs_flash.h"
25 #include "wifi_softap_sta.h"
26 #include "usart.h"
27 #include "udp_socket_select.h"
28
29 PRIVILEGED_DATA static portMUX_TYPE xTaskQueueMutex = portMUX_INITIALIZER_UNLOCKED;
30
31 pv_udp_socket_select_struct_t udp_client; //udp 结构体变量
32
33 //udp接收数据回调函数
34 void udp_recv_data(char *data, int len){
35     //获取对方地址,使用
36     char addr_str[128];
37     inet_ntoa_r(((struct sockaddr_in *)&udp_client.sockaddr->sin_addr->s_addr, addr_str, sizeof(addr_str) - 1);
38     //获取对方端口号 (udp_client.sockaddr.sin_port 由于系统数据存放大小端问题,需要做下转换)
39     uint16_t port = ntohs(udp_client.sockaddr.sin_port);
40     //打印对方的IP地址和端口号
41     ESP_LOGE("main", "udp socket addr-%s %d\r\n", addr_str, port);
42
43     //发送数据给UDP*
44     // struct sockaddr_in dest_addr;
45     // dest_addr.sin_addr.s_addr = inet_addr("192.168.0.104");//对方地址
46     // dest_addr.sin_family = AF_INET;
47     // dest_addr.sin_port = htons(1000);//对方端口号
48     // sendto(udp_client.socket_fd, data, len, 0, (struct sockaddr*)&dest_addr, sizeof(dest_addr));
49
50     /*返回数据给UDP*/
51     sendto(udp_client.socket_fd, data, len, 0, (struct sockaddr*)&udp_client.sockaddr, sizeof(udp_client.sockaddr));
52 }
53
54

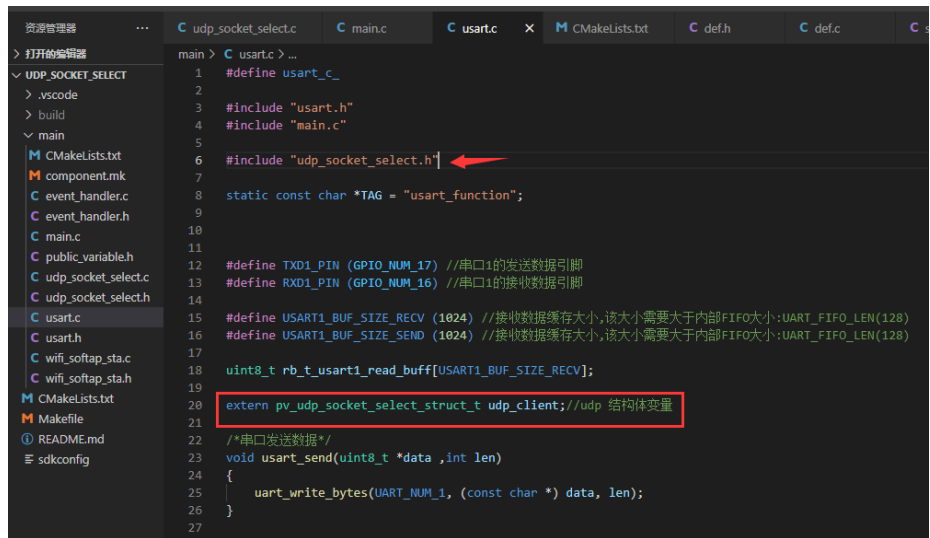
```

假设把串口接收的数据发送给UDP, 引用一下 udp\_client 变量

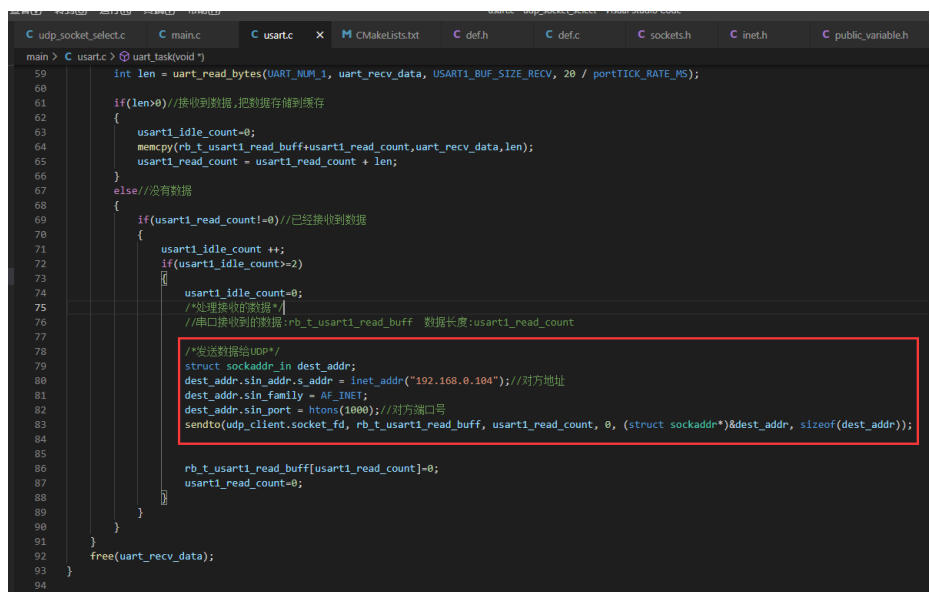
```

14 #include "nvs_flash.h"
15 #include "wifi_softap_sta.h"
16 #include "usart.h"
17 #include "udp_socket_select.h"
18
19 PRIVILEGED_DATA static portMUX_TYPE xTaskQueueMutex = portMUX_INITIALIZER_UNLOCKED;
20
21 pv_udp_socket_select_struct_t udp_client; //udp 结构体变量
22
23 //udp接收数据回调函数
24 void udp_recv_data(char *data, int len){
25     //获取对方地址,使用
26

```



```
main > C uartc > ...
1  #define usart_c_
2
3  #include "usart.h"
4  #include "main.c"
5
6  #include "udp_socket_select.h"
7
8  static const char *TAG = "usart_function";
9
10
11
12  #define TXD1_PIN (GPIO_NUM_17) //串口1的发送数据引脚
13  #define RXD1_PIN (GPIO_NUM_16) //串口1的接收数据引脚
14
15  #define USART1_BUF_SIZE_RECV (1024) //接收数据缓存大小,该大小需要大于内部FIFO大小:UART_FIFO_LEN(128)
16  #define USART1_BUF_SIZE_SEND (1024) //接收数据缓存大小,该大小需要大于内部FIFO大小:UART_FIFO_LEN(128)
17
18  uint8_t rb_t_usart1_read_buff[USART1_BUF_SIZE_RECV];
19
20  extern pv_udp_socket_select_struct_t udp_client; //udp 结构体变量
21
22  /*串口发送数据*/
23  void usart_send(uint8_t *data, int len)
24  {
25      uart_write_bytes(UART_NUM_1, (const char *) data, len);
26  }
27
```



```
main > C uartc > uart_task(void)
59  int len = uart_read_bytes(UART_NUM_1, uart_recv_data, USART1_BUF_SIZE_RECV, 20 / portTICK_RATE_MS);
60
61  if(len>0) //接收到数据,把数据存到缓存
62  {
63      usart1_idle_count=0;
64      memcpy(rb_t_usart1_read_buff+usart1_read_count, uart_recv_data, len);
65      usart1_read_count = usart1_read_count + len;
66  }
67  else //没有数据
68  {
69      if(usart1_read_count!=0) //已经接收到数据
70      {
71          usart1_idle_count ++;
72          if(usart1_idle_count>=2)
73          {
74              usart1_idle_count=0;
75              /*处理接收到的数据*/
76              //串口接收到的数据:rb_t_usart1_read_buff 数据长度:usart1_read_count
77
78              /*发送数据给UDP*/
79              struct sockaddr_in dest_addr;
80              dest_addr.sin_addr.s_addr = inet_addr("192.168.0.104"); //对方地址
81              dest_addr.sin_family = AF_INET;
82              dest_addr.sin_port = htons(1000); //对方端口号
83              sendto(udp_client.socket_fd, rb_t_usart1_read_buff, usart1_read_count, 0, (struct sockaddr*)&dest_addr, sizeof(dest_addr));
84
85              rb_t_usart1_read_buff[usart1_read_count]=0;
86              usart1_read_count=0;
87          }
88      }
89  }
90  }
91  free(uart_recv_data);
92  }
93  }
94
```

## 程序说明

### 1.程序是封装的,先按照固定步骤编写程序

```
main > C udp_socket_select.c C main.c X C usart.c M CMakeLists.txt C def.h C def.c C sockets.h C inet
> 打开的编辑器
udp_so...
.vscode
> build
main
M CMakeLists.txt
C component.mk
C event_handler.c
C event_handler.h
C main.c
C public_variable.h
C udp_socket_select.c
C udp_socket_select.h
C usart.c
C usart.h
C wifi_softap_sta.c
C wifi_softap_sta.h
M CMakeLists.txt
M Makefile
① README.md
sdkconfig

main > C main.c > app_main(void)
14 #include "nvs_flash.h"
15 #include "wifi_softap_sta.h"
16 #include "usart.h"
17 #include "udp_socket_select.h"
18
19 PRIVILEGED_DATA static portMUX_TYPE xTaskQueueMutex = portMUX_INITIALIZER_UNLOCKED;
20
21 pv_udp_socket_select_struct_t udp_client; //udp 结构体变量
22
23
24 //udp接收数据回调函数
25 void udp_recv_data(char *data, int len){
26     //获取对方地址,使用
27     char addr_str[28];
28     inet_ntoa_r(((struct sockaddr_in *)&udp_client.sockaddr)->sin_addr.s_addr, addr_str, sizeof(addr_str) - 1);
29     //获取对方端口号 (udp_client.sockaddr.sin_port 由于系统数据存放大小端问题,需要做下转换)
30     uint16_t port = ntohs(udp_client.sockaddr.sin_port);
31     //打印对方的IP地址和端口号
32     ESP_LOGE("main", "udp socket addr-%s %d\r\n", addr_str, port);
33
34     //发送数据给UDP*
35     // struct sockaddr_in dest_addr;
36     // dest_addr.sin_addr.s_addr = inet_addr("192.168.0.104");//对方地址
37     // dest_addr.sin_family = AF_INET;
38     // dest_addr.sin_port = htons(1000);//对方端口号
39     // sendto(udp_client.socket_fd, data, len, 0, (struct sockaddr*)&dest_addr, sizeof(dest_addr));
40
41     /*返回数据给UDP*/
42     sendto(udp_client.socket_fd, data, len, 0, (struct sockaddr*)&udp_client.sockaddr, sizeof(udp_client.sockaddr));
43 }
44
```

```
45
46 void app_main(void){
47     //初始化 NVS(配置WiFi的参数存储需要用到NVS)
48     esp_err_t ret = nvs_flash_init();
49     if (ret == ESP_ERR_NVS_NO_FREE_PAGES || ret == ESP_ERR_NVS_NEW_VERSION_FOUND) {
50         ESP_ERROR_CHECK(nvs_flash_erase());
51         ret = nvs_flash_init();
52     }
53     ESP_ERROR_CHECK(ret);
54
55     //初始化内部的lwip
56     ESP_ERROR_CHECK(esp_netif_init());
57     //创建系统事件任务
58     ESP_ERROR_CHECK(esp_event_loop_create_default());
59     //初始化配置AP+STA
60     wifi_init_softap_sta();
61
62     portENTER_CRITICAL(&xTaskQueueMutex); //进入临界点
63     usart_init(20);
64
65     //strcpy(udp_client.ip, "mnif.cn");//设置对方的IP地址
66     strcpy(udp_client.remote_ip, "192.168.0.104");//设置对方的IP地址
67     udp_client.remote_port = 1000;//设置对方的端口号
68
69     udp_client.local_port = 2000;//设置本地的端口号(如果设置为0 则程序会自动分配)
70     udp_socket_select_task_init(&udp_client, 10, udp_recv_data);
71
72     portEXIT_CRITICAL(&xTaskQueueMutex); //退出临界点
73 }
```

## 2.启动任务

```
130
131 /**
132  * @brief select UDP
133  * @param pv_udp_socket_select_value 客户端结构体
134  * @param task_priority 任务优先级
135  * @param RecvDataCallback 接收回调函数
136  * @return none
137  */
138 void udp_socket_select_task_init(pv_udp_socket_select_struct_t *pv_udp_socket_select_value, int task_priority, void(*RecvDataCallback)(char *data, int len)){
139     pv_udp_socket_select_value->socket_fd=-1;
140     pv_udp_socket_select_value->callback = RecvDataCallback;
141     //创建任务
142     xTaskCreate(udp_socket_select_task, "tcp_client_select_task", 4096, pv_udp_socket_select_value, task_priority, NULL);
143 }
144
```

## 3.如果需要解析域名,则等待连接上路由器



```

21  */
22  void udp_socket_select_task(void *arg){
23
24      pv_udp_socket_select_struct_t *pv_udp_socket_select_value = (pv_udp_socket_select_struct_t *)arg;
25      uint8_t need_dns=0;
26
27      struct addrinfo hints = {
28          .ai_family = AF_INET,
29          .ai_socktype = SOCK_DGRAM,
30      };
31      struct addrinfo *res;
32
33      socklen_t socklen;
34      int len;
35      int err;
36      int socket_fd;
37      char connection_state=0;
38      struct sockaddr_in server_addr, local_addr;
39      fd_set read_set;
40
41      //尝试DNS解析,如果解析失败说明确实需要DNS解析,但是需要先连接路由器
42      err = getaddrinfo(pv_udp_socket_select_value->remote_ip, NULL, &hints, &res);
43      if(err != 0 || res == NULL) {
44          my_printf(TAG, "DNS lookup failed err=%d res=%p", err, res);
45          need_dns=1;
46      }
47      if (need_dns){//等待模组连接上路由器
48          xEventGroupWaitBits(EventGroupHandleWifiEvent, EventBitsWifiStaConnected, false, true, portMAX_DELAY);
49      }
50      err = getaddrinfo(pv_udp_socket_select_value->remote_ip, NULL, &hints, &res);
51      if(err != 0 || res == NULL) {
52          my_printf(TAG, "DNS lookup failed err=%d res=%p", err, res);
53          return;
54      }
55
56      /*本地地址配置*/

```

#### 4.创建 和 绑定 socket ,然后进入select函数阻塞等待数据

提示一点: 其实也可以调用 connect, 只不过内部并不是去连接, 而是 lwip内部把UDP的信息记录了, 后面就可以使用 read 进行UDP数据接收

```

C udp_socket_select.c X C main.c C usart.c M CMakeLists.txt C def.h C def.c C so
main > C udp_socket_select.c > ...
52      my_printf(TAG, "DNS lookup failed err=%d res=%p", err, res);
53      return;
54  }
55
56      /*本地地址配置*/
57      local_addr.sin_family = res->ai_family;
58      if (pv_udp_socket_select_value->local_port!=0){
59          local_addr.sin_port = htons(pv_udp_socket_select_value->local_port);
60      }
61
62
63      /*对方地址配置*/
64      server_addr.sin_family = res->ai_family;
65      server_addr.sin_port = htons(pv_udp_socket_select_value->remote_port);
66      server_addr.sin_addr.s_addr = ((struct sockaddr_in *) (res->ai_addr))->sin_addr.s_addr;
67      freeaddrinfo(res);
68
69
70      //创建socket
71      socket_fd = socket(res->ai_family, res->ai_socktype, 0);
72      if(socket_fd<0){
73          my_printf(TAG, "Failed to allocate socket.");
74      }
75      //绑定socket
76      err=bind(socket_fd, (struct sockaddr *)&local_addr, sizeof(local_addr));
77      if (err<0){
78          my_printf(TAG, "Failed to bind socket");
79          close(socket_fd);
80      }
81
82      //清空fdset
83      FD_ZERO(&read_set);
84      //把sfd文件描述符添加到集合中
85      FD_SET(socket_fd, &read_set);
86
87      pv_udp_socket_select_value->socket_fd = socket_fd;
88
89      // connect(socket_fd, (struct sockaddr *)&server_addr, sizeof(server_addr));
90
91      while (true){
92          err = select(socket_fd+1,&read_set,NULL,NULL,NULL);
93          if(err < 0){

```

## 5.接收数据并调用回调函数

```
pv_udp_socket_select_value->socket_fd = socket_fd;

// connect(socket_fd, (struct sockaddr *)&server_addr, sizeof(server_addr));

while (true){
    err = select(socket_fd+1,&read_set,NULL,NULL,NULL);

    if(err < 0){
        my_printf(TAG, "select error \r\n");
        close(socket_fd);
        break;
    }

    if(FD_ISSET(socket_fd,&read_set)){//有数据需要读取
    {
        socklen = sizeof(pv_udp_socket_select_value->sockaddr);
        err = recvfrom(socket_fd, pv_udp_socket_select_value->read_buff, sizeof(pv_udp_socket_select_value->read_buff), 0, (struct sockaddr *)&server_addr, &socklen);
        if(err>0){
            pv_udp_socket_select_value->read_buff[err]=0;
            pv_udp_socket_select_value->CallBack(pv_udp_socket_select_value->read_buff,err);
        }
    }

    // err = recvfrom(socket_fd, pv_udp_socket_select_value->read_buff, sizeof(pv_udp_socket_select_value->read_buff), 0, (struct sockaddr *)&server_addr, &socklen);
}
```

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 - 693

0

0

« 上一篇: [901-ESP32\\_SDK开发-动手制作自己的WiFi视频遥控小车\(android\)](#)

posted on 2021-12-14 13:21 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](#)

发表评论

编辑

预览

B



</>

“ ”



支持 Markdown

//

自动补全

提交评论

退出

[Ctrl+Enter快捷键提交]

【推荐】2021中国.NET开发者峰会近50场热点技术专题揭秘

【推荐】华为开发者专区，与开发者一起构建万物互联的智能世界

【推荐】跨平台组态\工控\仿真\CAD 50万行C++源码全开放免费下载！

【推荐】华为 HMS Core 线上 Codelabs 挑战赛第4期，探索“智”感生活

**编辑推荐：**

- 一文讲透算法中的时间复杂度和空间复杂度计算方式
- .NET Core基础篇：集成Swagger文档与自定义Swagger UI
- 「译」.NET 6 中 gRPC 的新功能
- 一次缓存雪崩的灾难复盘
- 如何在 ASP.NET Core 中构建轻量级服务

**最新新闻：**

- 你是“造物主”，但显卡在燃烧！黑客帝国登陆PS5，史诗级画面刷爆全球（2021-12-14 12:59）
- 元宇宙的A股大爆炸（2021-12-14 12:50）
- 阿里巴巴新增多项员工关怀制度（2021-12-14 12:35）
- 众筹网站Kickstarter拥抱区块链（2021-12-14 12:33）
- 腾讯计划增持Supercell股份，估值110亿美元（2021-12-14 12:27）
- » 更多新闻...

**历史上的今天：**

- 2020-12-14 001-STM32+BC26基本控制篇-基础应用-购买云主机,安装MQTT服务器软件(.Wi...
- 2020-12-14 2-STM32+BC26/260Y基本控制篇-整体运行测试-微信小程序扫码绑定BC26,并通...
- 2020-12-14 7-微信小程序开发-Canvas画图入门与尺寸转换
- 2019-12-14 ESP8266 LUA脚本语言开发: 外设篇-串口
- 2019-12-14 ESP8266 LUA脚本语言开发: 外设篇-定时器,延时,看门狗
- 2019-12-14 ESP8266 LUA脚本语言开发: 外设篇-GPIO中断检测
- 2019-12-14 ESP8266 LUA脚本语言开发: 外设篇-GPIO输入检测

Powered by:

博客园

Copyright © 2021 杨奉武

Powered by .NET 6 on Kubernetes



单片机,物联网,上位机,...

扫一扫二维码，入群聊。