

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 - 836, 文章 - 0, 评论 - 329, 阅读 - 205万

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 / 00:00

- 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3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(2)
 CH579M物联网开发(12)
 CH579M学习开发(13)
 CH581/2/3 学习开发(2)
 ESP32学习开发(30)
 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)

212-CH579M学习开发-以太网例程-TCP客户端(项目应用封装,局域网或广域网测试)

`<p><iframe name="ifd" src="https://mnifdv.cn/resource/cnblogs/LearnCH579M" frameborder="0" scrolling="auto" width="100%" height="1500"></iframe></p>`

单片机CH579M学习开发(带蓝牙和以太网口的ARM M0内核的单片机,替代STM32,替代串口转以太网DTU)

开发板链接:<https://item.taobao.com/item.htm?ft=t&id=648634562877>

芯片购买链接:<https://item.taobao.com/item.htm?ft=t&id=649533679749>

开发板原理图:<https://mnifdv.cn/resource/cnblogs/LearnCH579M/CH579M>

资料源码下载链接:<https://github.com/yangfengwu45/LearnCH579M.git>

资料源码下载链接:<https://pan.baidu.com/s/1u9TN-EOIJk5K0Ak-PqRw4A> 提取码: 8888

点击链接加入群聊【网络学习开发】： 加入QQ群

- [学习Android](#)
教程中搭配的Android，C#等教程如上，各个教程正在整理。
- [001-硬件使用说明,下载和运行第一个程序](#)
- [002-官方资料学习说明,开发板蓝牙\(蓝牙定位\),网口通信测试](#)
- -----基本外设-----
- [003-新建工程说明](#)
- [100-基本外设-GPIO输入输出](#)
- [101-基本外设-定时器](#)
- [102-基本外设-串口](#)
- [103-基本外设-引脚中断](#)
- -----以太网-----
- V1.5及其以下版本网络指示灯为PB4,PB7; V1.8版本网络指示灯为PB5,PB6;
- [200-网络指示灯GPIO选择,检测网线连接状态](#)
- [201-DHCP](#)
- [202-DNS域名解析](#)
- [210-TCP客户端\(官方代码测试,和电脑直接连接测试\)](#)
- [211-TCP客户端\(项目应用封装,和电脑直连测试通信\)](#)
- [212-TCP客户端\(项目应用封装,局域网或广域网测试\)](#)
-
-
-
-

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)物联网开发(17)
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(ZLESP8266A)物联网开发(1)
STM32+ESP8266+AIR202/302远程升级方案(16)
STM32+ESP8266+AIR202/302终端管理方案(6)
STM32+ESP8266+Air302物联网开发(65)
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)
更多

阅读排行榜

1. ESP8266使用详解(AT,LUA,SDK)(174646)
2. 1-安装MQTT服务器(Windows),并连接测试(106425)
3. 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(68562)
4. ESP8266刷AT固件与node mcu固件(67352)
5. 有人WIFI模块使用详解(39650)
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)(37341)
7. C#中public与private与static(36738)
8. 关于TCP和MQTT之间的转换(35717)
9. android 之TCP客户端编程(33267)
10. android客服端+eps8266+单片机+路由器之远程控制系统(31781)

推荐排行榜

说明

这节演示一下模组作为TCP客户端和电脑上位机TCP服务器局域网通信

准备个路由器或者交换机.

测试

1.看一下要与之通信的电脑的IP地址

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. 我的大学四年(7)
4. 用ESP8266+android,制作自己的WIFI小车(Android 软件)(6)
5. ESP8266使用详解(AT,LUA, SDK)(6)

最新评论

1. Re:用ESP8266+android,制作自己的WIFI小车(Android 软件)
可以重新发一遍源代码吗
--evakzxx
2. Re:103-CH579M学习开发-基本外设-引脚中断
esp32 gpio 中断支持 chang 类型,即只要是上升沿或者下降沿就会触发, ch579 不支持吗?
--yh251512

网络连接详细信息

网络连接详细信息(D):

属性	值
连接特定的 DNS 后缀	
描述	Realtek PCIe GbE Family Controller
物理地址	F4-B5-20-09-8A-F9
已启用 DHCP	是
IPv4 地址	192.168.1.93
IPv4 子网掩码	255.255.255.0
获得租约的时间	2022年1月10日 13:45:50
租约过期的时间	2022年1月11日 13:45:48
IPv4 默认网关	192.168.1.1
IPv4 DHCP 服务器	192.168.1.1
IPv4 DNS 服务器	192.168.1.1
IPv4 WINS 服务器	
已启用 NetBIOS over Tc...	是
IPv6 地址	2409:8a3c:f49:5030:31d1:3c94:7cbb:e
临时 IPv6 地址	2409:8a3c:f49:5030:4d45:be26:9146:e
连接-本地 IPv6 地址	fe80::31d1:3c94:7cbb:edb2%20
IPv6 默认网关	fe80::1%20
IPv6 DNS 服务器	fe80::1%20

2.修改单片机程序里面的地址

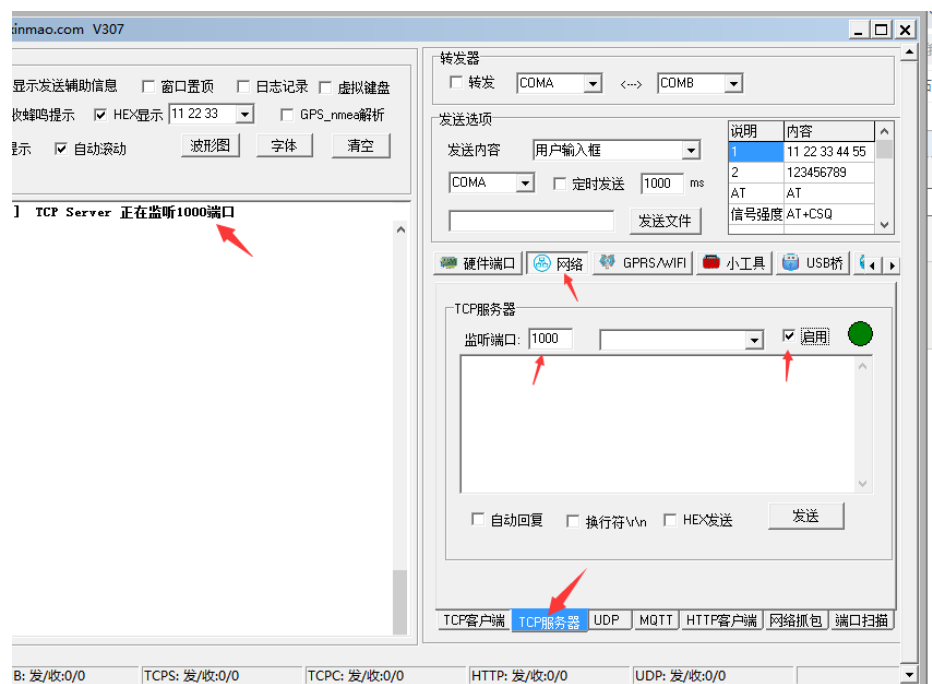
```
net_dhcp.c  CH57xNET.H  net_init.c  time.c  sys_tick.c  main.c  net_tcp_client.c
89 int main()
90 {
91     int i;
92     PWR_UnitModCfg(ENABLE, UNIT_SYS_PLL); //使能PLL(给以太网提供时钟)
93     DelayMs(3);
94     // SetSysClock(CLK_SOURCE_HSE_32MHz); /* 外部晶振 PLL 输出32MHz */
95     // SysTick_Config(FREQ_SYS/1000); //系统定时器,1ms
96     uart_init(); //初始化串口
97     net_init(); //初始化以太网
98     time0_init(); //初始化定时器
99     printf("start\r\n");
100
101
102     /*设置服务器的IP地址*/
103     net_tcp_client.ip_addr_remote[0]=192;
104     net_tcp_client.ip_addr_remote[1]=168;
105     net_tcp_client.ip_addr_remote[2]=1;
106     net_tcp_client.ip_addr_remote[3]=93;
107     /*设置端口号*/
108     net_tcp_client.port_remote =1000; //要连接的服务器端口号
109     net_tcp_client.port_local = 2000; //本地端口号
110
111     net_tcp_client.socke_index = 0; //给每个客户端设置一个标识(可以不设置)
112
113     while(1)
114     {
115         net_dhcp_while(dhcp_callback); //DHCP
116
117         CH57xNET_MainTask(); /* CH57xNET库主任务函数, 需要在主循环中不断调用 */
118
119         /* 查询事件, 如果有事件, 则调用处理函数 */
120         if(CH57xNET_QueryGlobalInt()) CH57xNET_HandleGlobalInt();
121     }
122 }
123 }
```

3.打开网络调试助手,启动TCP服务器,监听的端口号为1000

CH579开发相关

工具助手

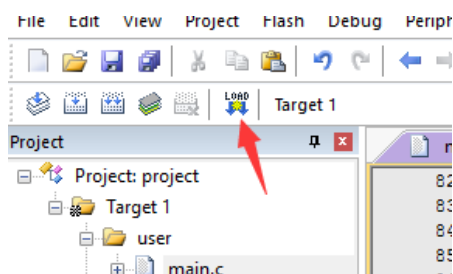
BLEAssist.apk
COMNET.exe
config.ini
Log_202103.txt
TCPUDPDebug102_Setup.exe
WinHex_v20.2_x86_x64.exe
串口驱动CH340.rar



4.使用网线连接电脑连接的路由器或者交换机

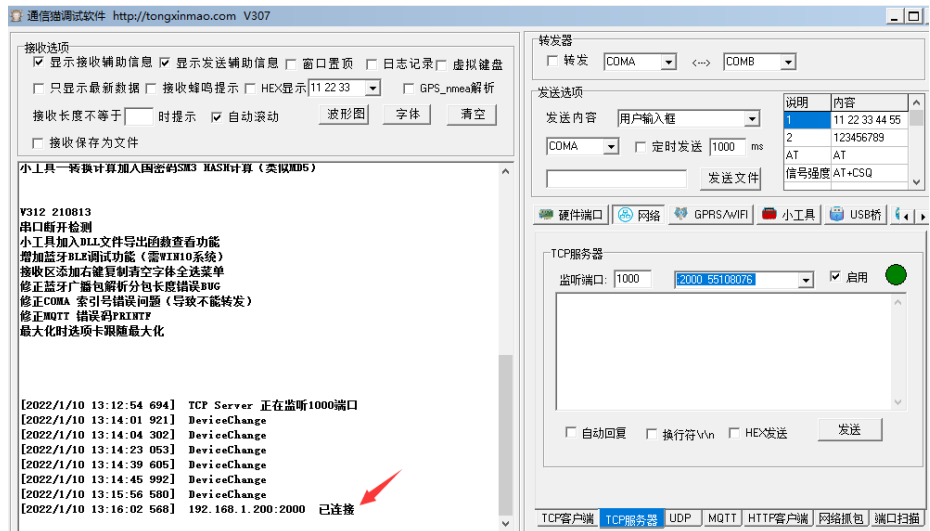


5.编译下载这节的程序到开发板



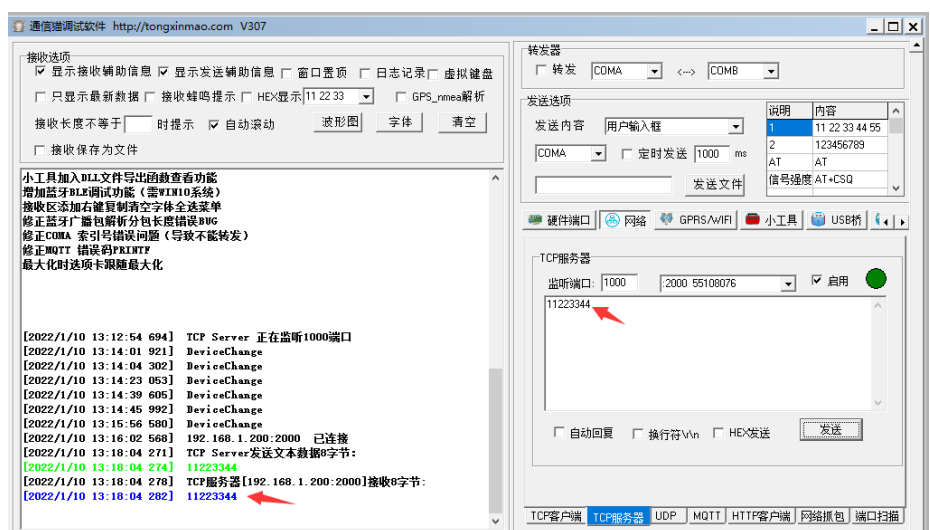
6.开发板上面的串口会打印连接成功的日志

```
ATK XCOM V2.0
net_phy_connected
GINT_STAT_PHY_CHANGE 02
DMCP SuccessIPAddr = 192.168.1.95
GWIPAddr = 192.168.1.1
IPAddr = 255.255.255.0
DNS1: 192.168.1.1
DNS2: 0.0.0.0
net_top_client_creat_socket_id: 0
net_top_client_connect.....
top connect success:0-0
```



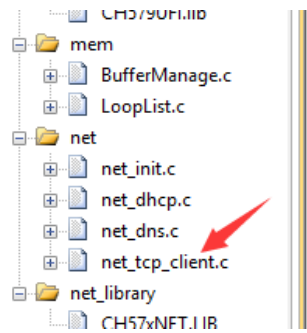
7.使用TCP调试助手发送数据给开发板

开发板会返回接收的数据



程序说明

1.为便于项目应用,我封装了下

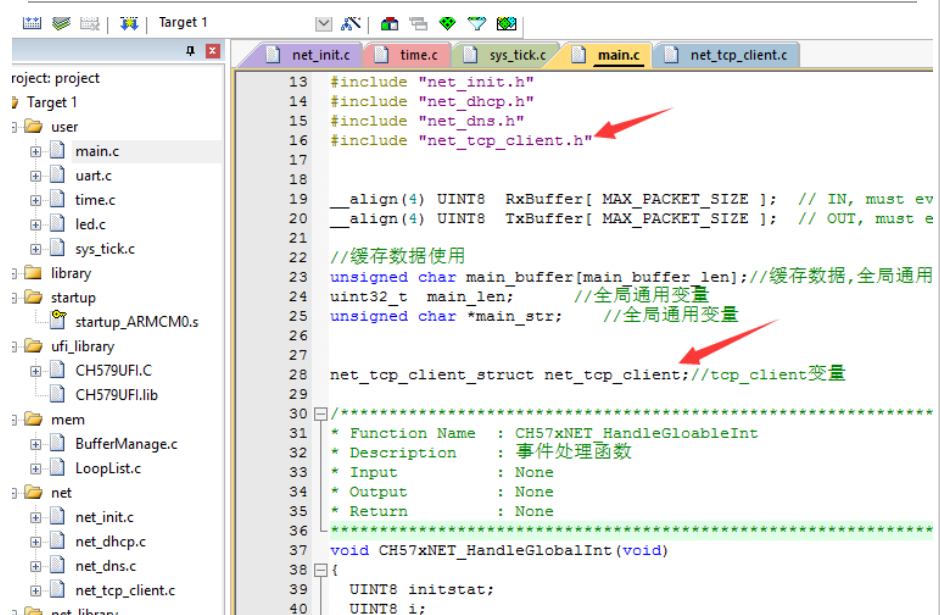


2.移植使用的时候只需要把下面的文件拷贝到自己的工程

- net_dhcp.c
- net_dhcp.h
- net_dns.c
- net_dns.h
- net_init.c
- net_init.h
- net_tcp_client.c
- net_tcp_client.h

3.创建net_tcp_client_struct 变量

如果想多路客户端可创建多个



4.把 net_tcp_client_data 函数放到网络接收数据里面

```

net_init.c  time.c  sys_tick.c  main.c  net_tcp_client.c
31  * Function Name : CH57xNET_HandleGlobalInt
32  * Description  : 事件处理函数
33  * Input       : None
34  * Output      : None
35  * Return      : None
36  *****/
37 void CH57xNET_HandleGlobalInt(void)
38 {
39     UINT8 initstat;
40     UINT8 i;
41     UINT8 socketinit;
42     initstat = CH57xNET_GetGlobalInt();
43     /* 读全局中断状态并清除 */
44     if(initstat & GINT_STAT_UNREACH){/* 不可达中断 */
45         printf("UnreachCode : %ld\n",CH57xInf.UnreachCode);
46         printf("UnreachProto : %ld\n",CH57xInf.UnreachProto);
47         printf("UnreachPort : %ld\n",CH57xInf.UnreachPort);
48         /* 查看不可达代码 */
49         /* 查看不可达协议类型 */
50         /* 查询不可达端口 */
51     }
52     if(initstat & GINT_STAT_IP_CONFLI){/* IP冲突中断 */
53     }
54     if(initstat & GINT_STAT_PHY_CHANGE){/* PHY改变中断 */
55         i = CH57xNET_GetPHYStatus();/* 获取PHY状态 */
56         printf("GINT_STAT_PHY_CHANGE %02x\n",i);
57     }
58     if(initstat & GINT_STAT_SOCKET){/* Socket中断 */
59         for(i = 0; i < CH57xNET_MAX_SOCKET_NUM; i++){
60             socketinit = CH57xNET_GetSocketInt(i);/* 读socket中断并清零 */
61             if(socketinit){/* 有事件需要处理 */
62                 net_tcp_client_data(&net_tcp_client, i, socketinit);/* 处理TCP数据 */
63             }
64         }
65     }
66 }

```

5.创建回调函数, 初始化服务器地址和端口号

```

net_dns.c  net_dhcp.c  CH57xNET.H  net_init.c  time.c  sys_tick.c  main.c  net_tcp_client.c
64  /*接收消息回调*/
65 void tcp_recv_back(unsigned char sockeid, int socke_index, unsigned char* data, unsigned long length){
66     uart_send_byte_it(1, data, length);/*串口打印*/
67     net_tcp_client_send(&net_tcp_client, data, length);/*发送数据给服务器*/
68 }
69
70 /*连接上服务器*/
71 void tcp_connected_back(unsigned char socke_id, int socke_index){
72 }
73
74 /*连接断开 或 未连接成功*/
75 void tcp_disconnected_back(unsigned char socke_id, int socke_index){
76     net_tcp_client_connect(&net_tcp_client);/*重新连接*/
77 }
78
79 //每次dhcp成功会进入此函数,如果有需要初始化的在这里面初始化
80 void dhcp_callback(void)
81 {
82     net_tcp_client_close(&net_tcp_client);/*先尝试关闭(必须加)*/
83     /*创建socket*/
84     net_tcp_client_creat_socket(&net_tcp_client, tcp_connected_back, tcp_disconnected_back, tcp_recv_back);
85     net_tcp_client_connect(&net_tcp_client);/*连接服务器*/
86 }
87
88 int main()
89 {
90     int i;
91     FWR_UnitModCfg(ENABLE, UNIT_SYS_PLL);/*使能PLL(给以太网提供时钟)*/
92     DelayMs(3);
93     // SetSysClock(CLK_SOURCE_HSE_32MHz);/* 外部晶振 PLL 输出32MHz */
94     // SysTick_Config(FREQ_SYS/1000);/*系统定时器,1ms*/
95     uart_init();/*初始化串口*/
96     net_init();/*初始化以太网*/
97     time0_init();/*初始化定时器*/
98     printf("start\r\n");
99 }
100
101 /*设置服务器的IP地址*/
102 net_tcp_client.ip_addr_remote[0]=192;
103 net_tcp_client.ip_addr_remote[1]=168;
104 net_tcp_client.ip_addr_remote[2]=1;
105 net_tcp_client.ip_addr_remote[3]=93;
106 /*设置端口号*/
107 net_tcp_client.port_remote = 1000;/*要连接的服务器端口号*/
108 net_tcp_client.port_local = 2000;/*本地端口号*/
109
110 net_tcp_client.socke_index = 0;/*给每个客户端设置一个标识(可以不设置)*/
111 net_tcp_client.socke_id = 255;/*初始化该值为255(必须设置下),设置为255就可以
112
113
114

```

6.连接服务器

因为连接了路由器,所以需要DHCP完成之后去执行连接


```

64
65 /*接收消息回调*/
66 void tcp_recv_back(unsigned char sockeid, int socke_index, unsigned char* data, unsigned long length){
67     uart_send_byte_it(1, data, length); //串口打印
68
69     net_tcp_client_send(&net_tcp_client, data, length); //发送数据给服务器
70 }
71
72 /*连接上服务器*/
73 void tcp_connected_back(unsigned char socke_id, int socke_index){
74 }
75
76 /*连接断开 或 未连接成功*/
77 void tcp_disconnected_back(unsigned char socke_id, int socke_index){
78     net_tcp_client_connect(&net_tcp_client); //重新连接
79 }
80
81 //每次dhcp成功会进入此函数,如果有需要初始化的在这里面初始化
82 void dhcp_callback(void)
83 {
84     net_tcp_client_close(&net_tcp_client); //先尝试关闭(必须加)
85     /*创建socket*/
86     net_tcp_client_creat_socket(&net_tcp_client, tcp_connected_back, tcp_disconnected_back, tcp_recv_back);
87     net_tcp_client_connect(&net_tcp_client); //连接服务器
88 }
89
90 int main()
91 {
92     int i;
93     PWR_UnitModCfg(ENABLE, UNIT_SYS_PLL); //使能PLL(给以太网提供时钟)
94     DelayMs(3);
95     // SetSysClock(CLK_SOURCE_HSE_32MHz); /* 外部晶振 PLL 输出32MHz */
96     // SysTick_Config(FREQ_SYS/1000); //系统定时器,1ms
97     // uart_init(); //初始化串口

```

6.接收发送数据

```

55 for(i = 0; i < CH57xNET_MAX_SOCKET_NUM; i++){
56     socketinit = CH57xNET_GetSocketInt(i); /* 读socket中断并清零 */
57     if(socketinit){ //有事件需要处理
58         net_tcp_client_data(&net_tcp_client, i, socketinit); //处理TCP数据
59     }
60 }
61 }
62 }
63 }
64
65 /*接收消息回调*/
66 void tcp_recv_back(unsigned char sockeid, int socke_index, unsigned char* data, unsigned long length){
67     uart_send_byte_it(1, data, length); //串口打印
68
69     net_tcp_client_send(&net_tcp_client, data, length); //发送数据给服务器
70 }
71
72 /*连接上服务器*/
73 void tcp_connected_back(unsigned char socke_id, int socke_index){
74 }
75
76 /*连接断开 或 未连接成功*/
77 void tcp_disconnected_back(unsigned char socke_id, int socke_index){
78     net_tcp_client_connect(&net_tcp_client); //重新连接
79 }
80 }
81

```

假设是连接的域名

1.把域名解析到 net_tcp_client.ip_addr_remote 数组, 然后再执行DHCP

关于域名api函数的使用参照前面的dsn域名解析部分

```

main.c
10 #include "time.h"
11 #include "sys_tick.h"
12
13 #include "net_init.h"
14 #include "net_dhcp.h"
15 #include "net_dns.h"
16 #include "net_tcp_client.h"
17
18
19 __align(4) UINT8  RxBuffer[ MAX_PACKET_SIZE ]; // IN, must eve
20 __align(4) UINT8  TxBuffer[ MAX_PACKET_SIZE ]; // OUT, must ev
21
22 //缓存数据使用
23 unsigned char main_buffer[main_buffer_len]; //缓存数据,全局通用
24 uint32_t main_len; //全局通用变量
25 unsigned char *main_str; //全局通用变量
26
27
28 char domain_name[20]="mnif.cn";
29
30 net_tcp_client_struct net_tcp_client; //tcp_client变量
31

```

```

main.c
100 net_init(); //初始化以太网
101 time0_init(); //初始化定时器
102 printf("start\r\n");
103
104
105 /*设置服务器的IP地址*/
106 net_tcp_client.ip_addr_remote[0]=0;
107 net_tcp_client.ip_addr_remote[1]=0;
108 net_tcp_client.ip_addr_remote[2]=0;
109 net_tcp_client.ip_addr_remote[3]=0;
110 /*设置端口号*/
111 net_tcp_client.port_remote =1883; //要连接的服务器端口号
112 net_tcp_client.port_local = 2000; //本地端口号
113
114 net_tcp_client.socke_index = 0; //给每个客户端设置一个标识(可以不设置)
115 net_tcp_client.socke_id = 255; //初始化该值为255(必须设置下),设置为255就可以
116
117 while(1)
118 {
119     if(net_tcp_client.ip_addr_remote[0]==0) //没有解析出来域名
120     {
121         net_dns_query(domain_name, net_tcp_client.ip_addr_remote); //解析域名
122     }
123     else
124     {
125         net_dhcp_while(dhcp_callback); //DHCP
126     }
127
128
129     CH57xNET_MainTask(); /* CH57xNET库主任务函数,需要在主循环中不断调用 */
130
131     /* 查询事件,如果有事件,则调用处理函数 */
132     if(CH57xNET_QueryGlobalInt()) CH57xNET_HandleGlobalInt();
133
134 }
135 }
136

```

其它

1.把连接函数放到断开连接回调函数里面执行可以做到断线重连

```

net_init.c | time.c | sys_tick.c | main.c | net_tcp_client.c
64
65 /*接收消息回调*/
66 void tcp_recv_back(unsigned char sockeid, int socke_index, unsigned char* data, unsig
67     uart_send_byte_it(1, data, length); //串口打印
68
69     net_tcp_client_send(&net_tcp_client, data, length); //发送数据给服务器
70 }
71
72 /*连接上服务器*/
73 void tcp_connected_back(unsigned char socke_id, int socke_index){
74 }
75
76 /*连接断开 或 未连接成功*/
77 void tcp_disconnected_back(unsigned char socke_id, int socke_index){
78     net_tcp_client_connect(&net_tcp_client); //重新连接
79 }
80
81
82 int main()
83 {
84     int i;

```

2. 连接超时时间可在 CH57xNET.h 里面设置(一般不需要设置)

当前是每隔 $10 \times 50 = 500\text{ms}$ 尝试重新连接, 总共尝试20次, 即连接超时时间为 10s

```

CH57xNET.h | net_init.c | time.c | sys_tick.c | main.c | net_tcp_client.c
31 #define NULL 0
32 #endif
33
34
35 /* Socket的个数 */
36 #ifndef CH57xNET_MAX_SOCKET_NUM
37 #define CH57xNET_MAX_SOCKET_NUM 4 /* Socket的个数, 用户可以配置, 默认为4个Socket, 最大为32 */
38 #endif
39
40
41 #ifndef RX_QUEUE_ENTRIES
42 #define RX_QUEUE_ENTRIES 4 /* CH57x MAC接收描述符队列, 此值不得小于2, 可配置默认为7, 最大为32 */
43 #endif
44
45 #ifndef RX_BUF_SIZE
46 #define RX_BUF_SIZE 600 /* CH57x MAC接收每个缓冲区长度, 为4的整数倍 */
47 #endif
48
49 #ifndef CH57xNET_PING_ENABLE
50 #define CH57xNET_PING_ENABLE TRUE /* 默认PING开启 */
51 #endif
52
53 #ifndef TCP_RETRY_COUNT
54 #define TCP_RETRY_COUNT 20 /* TCP重试次数, 位宽为5位 */
55 #endif
56
57 #ifndef TCP_RETRY_PERIOD
58 #define TCP_RETRY_PERIOD 10 /* TCP重试周期, 单位为50ms, 位宽为5位 */
59 #endif
60
61 #ifndef CH57xNETTIMEPERIOD
62 #define CH57xNETTIMEPERIOD 1 /* 定时器周期, 单位ms, 不得大于500 */
63 #endif
64
65 #define MAC_TMR_FREQ 1000 /* 1000Hz */

```

分类: CH579M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: 211-CH579M学习开发-以太网例程-TCP客户端(项目应用封装,和电脑直连测试通信)

posted on 2022-01-10 14:23 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】发布 vscode 插件 Cnblogs Client For VSCode 预览版
- 【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】参与华为 HarmonyOS 开发者创新大赛，一起创造无限可能

编辑推荐：

- asp.net core 中优雅的进行响应包装
- 全链路压测(4)：全链路压测的价值是什么？
- 神奇的滤镜！巧妙实现内凹的平滑圆角
- 实践剖析 .NET Core 如何支持 Cookie 和 JWT 混合认证、授权
- 探索 dotnet core 为何在 Windows7 系统需要补丁的原因

最新新闻：

- “难看”的互联网电视，正被年轻人抛弃？
 - 腾讯迈出元宇宙关键一步：拟收购游戏手机厂商黑鲨
 - 互联网信用付整改加速落地，“京东白条”升级为白条信用卡
 - 一切顺利！韦伯望远镜完成“完全部署”，遮阳板展开曾发生小故障
 - Tesla为FSD功能加入模式选择，让自动辅助驾驶风格可温柔或激进
- » 更多新闻...

Powered by:

博客园

Copyright © 2022 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。