

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 - 835, 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 / 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(12)
 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)

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)

<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\(项目应用封装,和电脑直连测试通信\)](#)
-
-
-
-
-

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

- ESP8266使用详解(AT,LUA,SDK)(174645)
- 1-安装MQTT服务器(Windows),并连接测试(106417)
- 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(68562)
- ESP8266刷AT固件与node mcu固件(67351)
- 有人WIFI模块使用详解(39650)
- (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)
- C#中public与private与static(36734)
- 关于TCP和MQTT之间的转换(35714)
- android 之TCP客户端编程(33266)
- android服务端+esp8266+单片机+路由器之远程控制系统(31781)

推荐排行榜

测试

1.默认设置如下

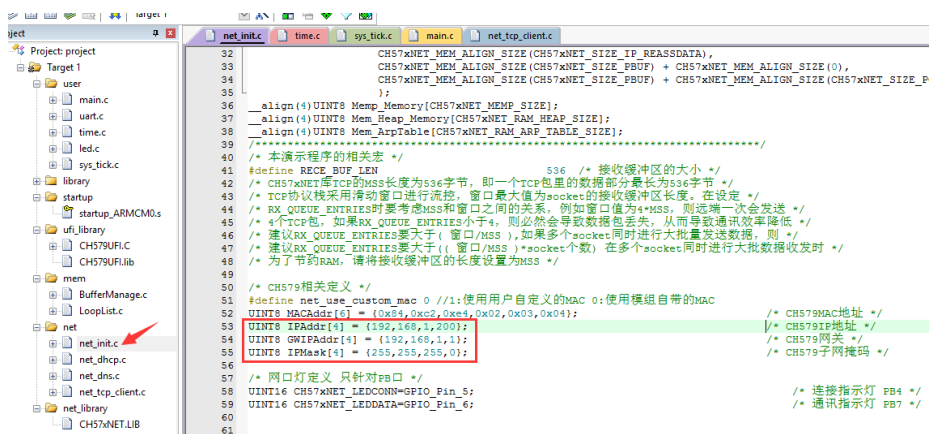
网关地址:192.168.1.1

子网掩码:255.255.255.0

电脑地址:192.168.1.100

开发板地址:192.168.1.200

2.可在单片机程序net_init文件中修改配置



2.修改电脑配置

关闭电脑防火墙

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



控制面板



系统和安全

查看你的计算机状态
通过文件历史记录保存你的文件备份副
备份和还原(Windows 7)
查找并解决问题



安全和维护

检查计算机的状态并解决问题 | 更改用户帐户控制设置 | 常见计算机问题疑难解答



Windows 防火墙

检查防火墙状态 | 允许应用通过 Windows 防火墙



系统

查看 RAM 的大小和处理器速度 | 允许远程访问 | 启动远程协助 | 查看该计算



电源选项

更改电源按钮的功能 | 更改计算机睡眠时间



文件历史记录

允许应用或功能通过 Windows 防火墙

允许应用或功能通过 Windows 防火墙



更改通知设置



启用或关闭 Windows 防火墙



还原默认值



高级设置

对网络进行疑难解答

Windows 防火墙

更新防火墙

Windows 防火墙




推荐的设置

专用网




自定义各类网络的设置

你可以修改使用的每种类型的网络的防火墙设置。

专用网络设置

-  ☐ 启用 Windows 防火墙
- ☐ 阻止所有传入连接，包括位于允许应用列表中的应用
 - ☒ Windows 防火墙阻止新应用时通知我
-  ☒ 关闭 Windows 防火墙(不推荐) 

公用网络设置

-  ☐ 启用 Windows 防火墙
- ☐ 阻止所有传入连接，包括位于允许应用列表中的应用
 - ☒ Windows 防火墙阻止新应用时通知我
-  ☒ 关闭 Windows 防火墙(不推荐) 

打开电脑的网络配置,配置下电脑

网络和共享中心

控制面板 > 网络和 Internet > 网络和共享中心

文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)

[查看基本网络信息并设置连接](#)

查看活动网络

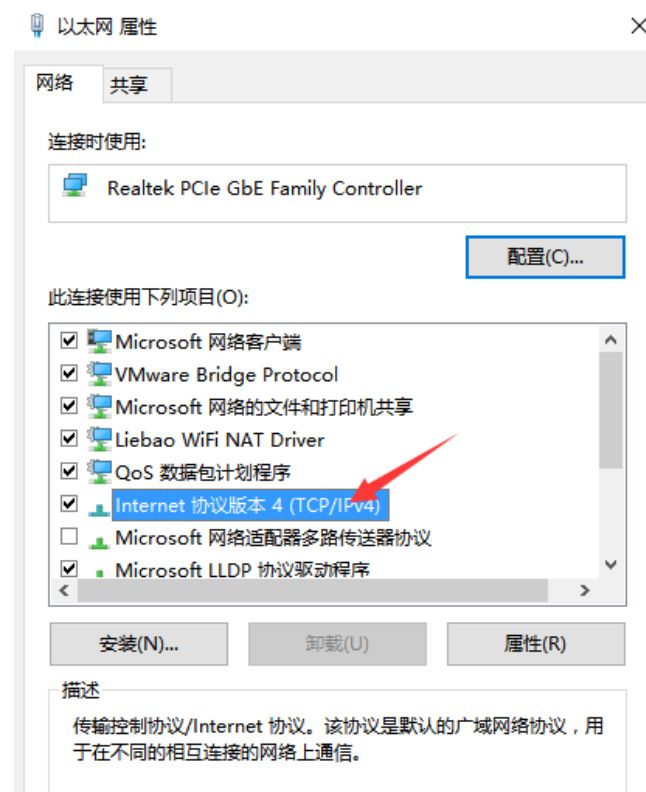
QQQQQ 4 公用网络	访问类型: Internet 连接: WLAN (QQQQQ)
未识别的网络 公用网络	访问类型: 无法连接到网络 连接: 以太网

更改网络设置

 [设置新的连接或网络](#)
设置宽带、拨号或 VPN 连接；或设置路由器或接入点。

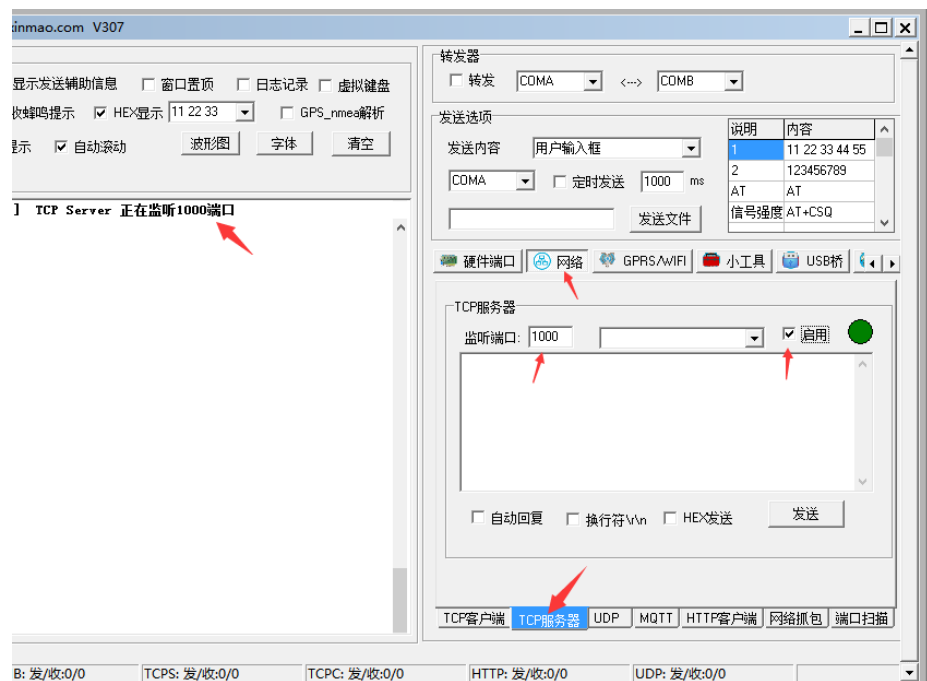
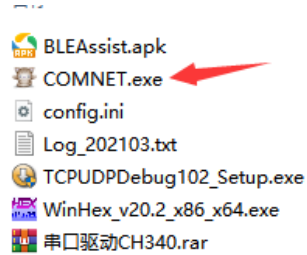
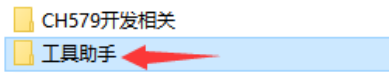
 [问题疑难解答](#)
诊断并修复网络问题，或者获得疑难解答信息。

以太网
未识别的网络, 共享的
Realtek PCIe GbE Family Contr...

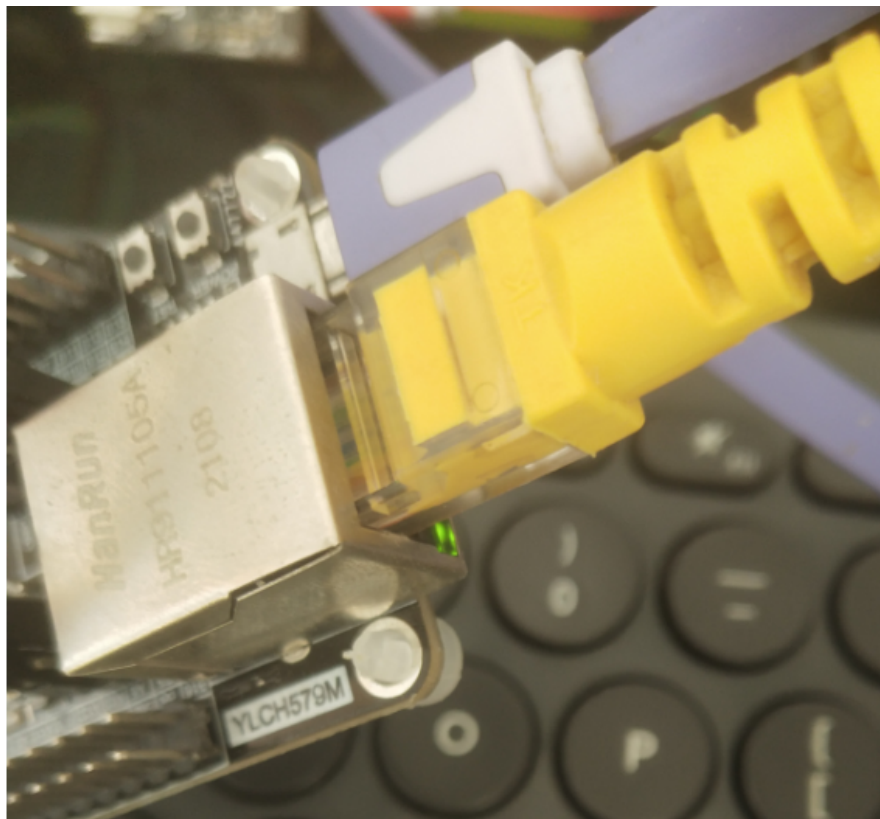




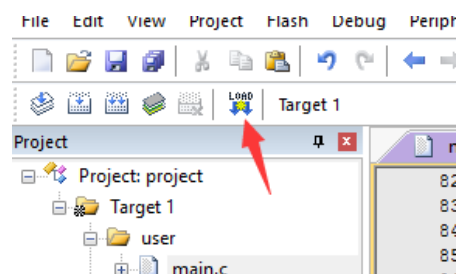
3. 打开网络调试助手, 启动TCP服务器, 监听的端口号为1000



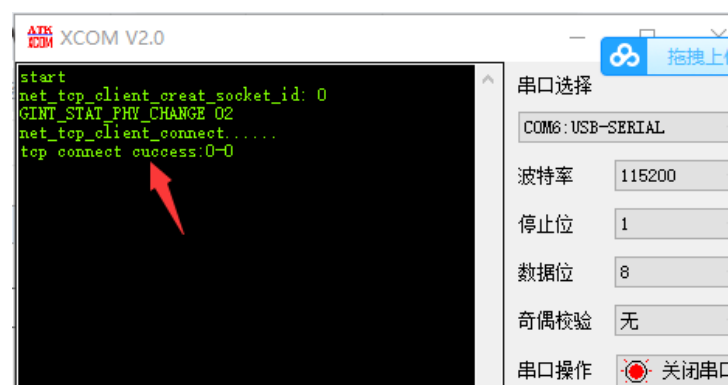
4. 使用一根网线把电脑网口和开发板网口连接

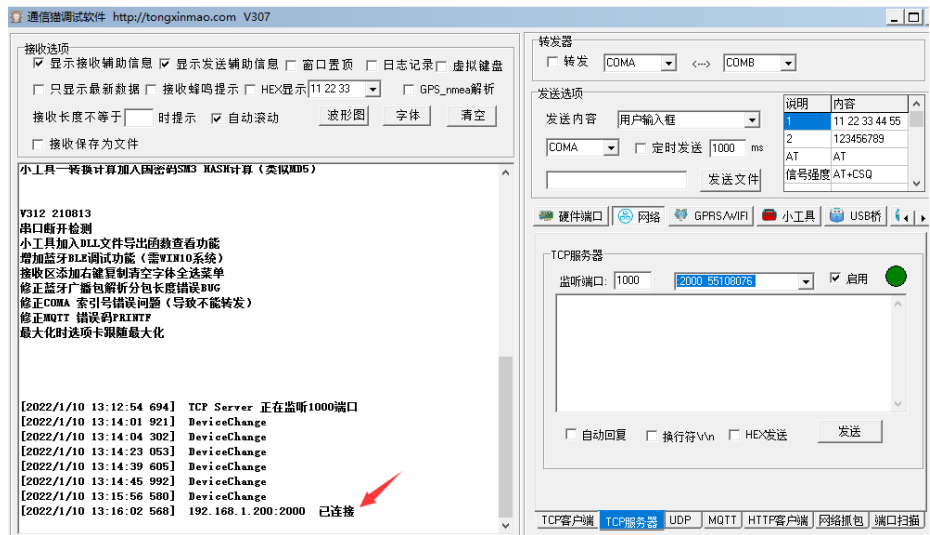


5.编译下载这节的程序到开发板



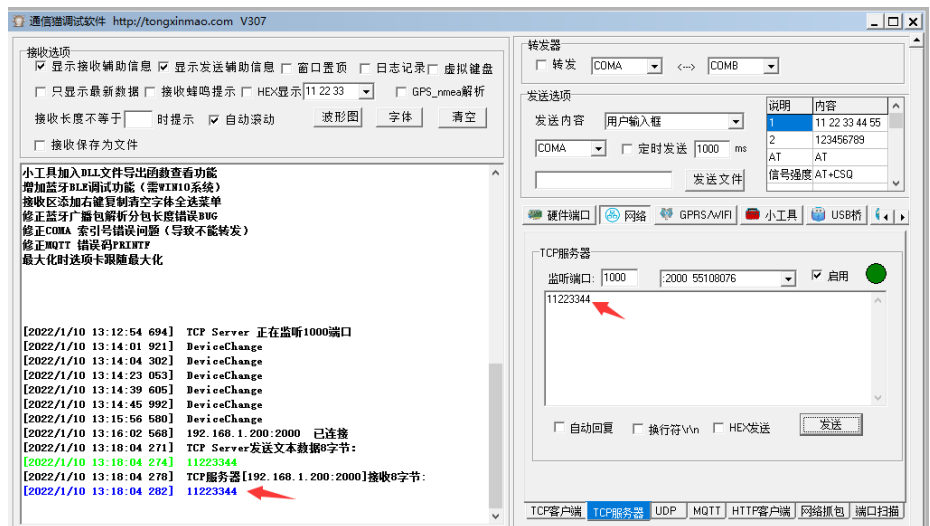
6.开发板上面的串口会打印连接成功的日志





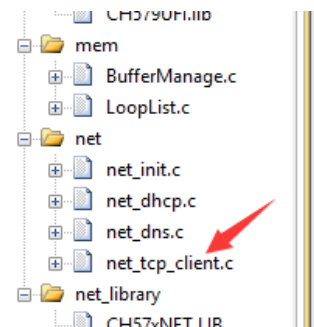
7.使用TCP调试助手发送数据给开发板

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

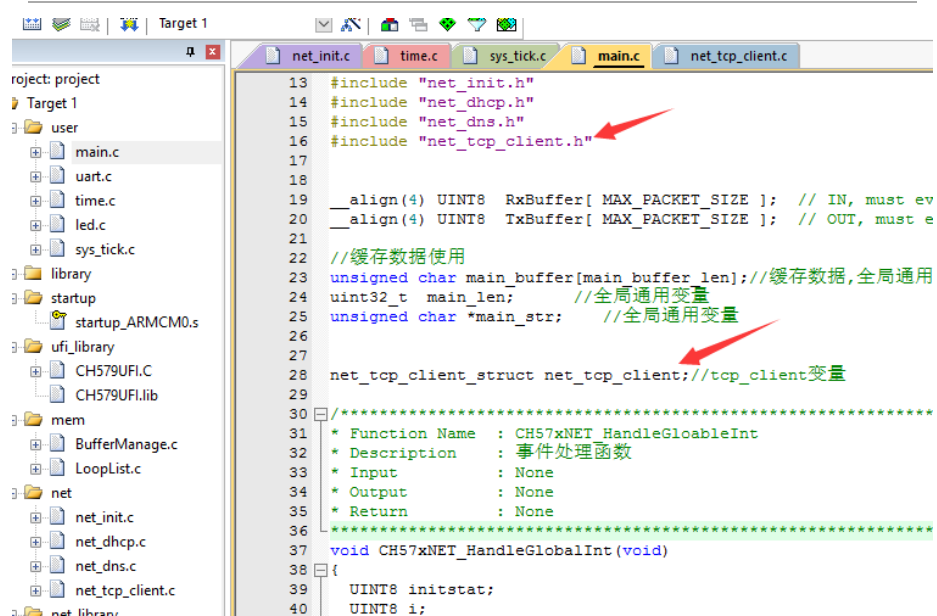


2.移植使用的时候只需要把下面的文件拷贝到自己的工程

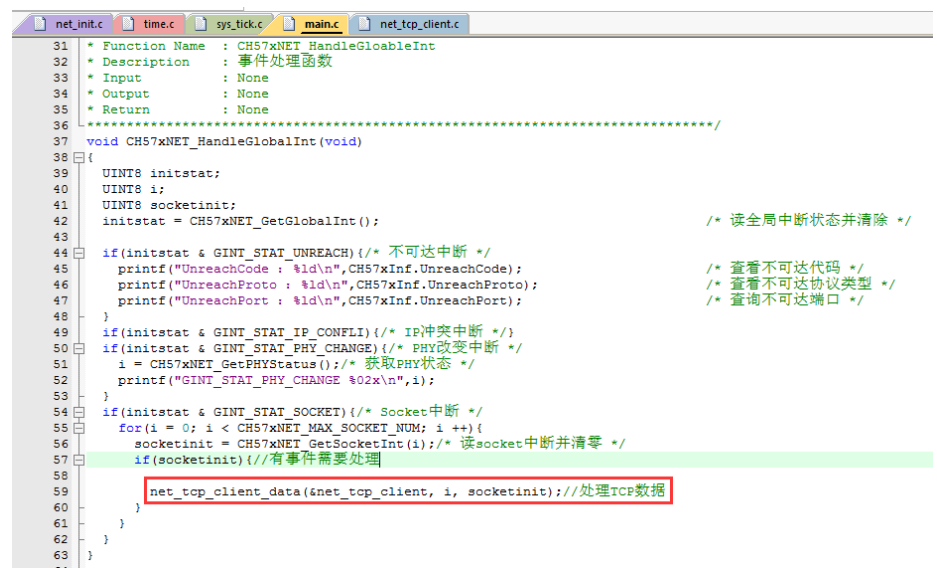
- net_dhcp.c
- net_dhcp.h
- net_dns.c
- net_dns.h
- net_init.c
- net_init.h
- net_tcp_client.c
- net_tcp_client.h

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



5. 创建回调函数, 初始化服务器地址和端口号

```
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, 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
82 int main()
83 {
84     int i;
85     PWR_UnitModCfg(ENABLE, UNIT_SYS_PLL); //使能PLL(给以太网提供时钟)
86     DelayMs(3);
87     // SetSysClock(CLK_SOURCE_HSE_32MHz); /* 外部晶振 PLL 输出32MHz */
88     // SysTick_Config(FREQ_SYS/1000); //系统定时器,1ms
89     uart_init(); //初始化串口
90     net_init(); //初始化以太网
91     time0_init(); //初始化定时器
92     printf("start\r\n");
93
94     /*设置服务器的IP地址*/
95     net_tcp_client.ip_addr_remote[0]=192;
96     net_tcp_client.ip_addr_remote[1]=168;
97     net_tcp_client.ip_addr_remote[2]=1;
98     net_tcp_client.ip_addr_remote[3]=100;
99     /*设置端口号*/
100     net_tcp_client.port_remote = 1000; //要连接的服务器端口号
101     net_tcp_client.port_local = 2000; //本地端口号
102
103     net_tcp_client.socke_index = 0; //给每个客户端设置一个标识(可以不设置)
104
105     /*创建socket*/
106     net_tcp_client_creat_socket(&net_tcp_client, tcp_connected_back, tcp_disconnected_back, tcp_recv_back);
107     while(1)
108     {
109     }
110 }
```

6. 连接服务器

```
net_init.c | time.c | sys_tick.c | main.c | net_tcp_client.c
91 time0_init(); //初始化定时器
92 printf("start\r\n");
93
94
95 /*设置服务器的IP地址*/
96 net_tcp_client.ip_addr_remote[0]=192;
97 net_tcp_client.ip_addr_remote[1]=168;
98 net_tcp_client.ip_addr_remote[2]=1;
99 net_tcp_client.ip_addr_remote[3]=100;
100 /*设置端口号*/
101 net_tcp_client.port_remote = 1000; //要连接的服务器端口号
102 net_tcp_client.port_local = 2000; //本地端口号
103
104 net_tcp_client.socke_index = 0; //给每个客户端设置一个标识(可以不设置)
105
106 /*创建socket*/
107 net_tcp_client_creat_socket(&net_tcp_client, tcp_connected_back, tcp_disconnected_back, tcp_recv_back);
108
109 while(1)
110 {
111     net_tcp_client_connect(&net_tcp_client); //连接服务器
112
113     CH57xNET_MainTask(); /* CH57xNET库主任务函数, 需要在主循环中不断调用 */
114
115     /* 查询事件, 如果有事件, 则调用处理函数 */
116     if(CH57xNET_QueryGlobalInt()) CH57xNET_HandleGlobalInt();
117 }
118
119 }
```

6. 接收发送数据

```
net_init.c | time.c | sys_tick.c | main.c | net_tcp_client.c
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     net_tcp_client_send(&net_tcp_client, data, length);/*发送数据给服务器*/
69 }
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 }
```

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, unsigned long length){
67     uart_send_byte_it(1, data, length);/*串口打印*/
68     net_tcp_client_send(&net_tcp_client, data, length);/*发送数据给服务器*/
69 }
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 #endif
```

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: [202-CH579M学习开发-以太网例程-DNS域名解析](#)

posted on 2022-01-10 13:32 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：

- [Linux 5.16 释出](#)
- [2022 年或以后注定消失的五种编程语言](#)
- [寺库还能撑多久？](#)
- [“2022全国网上年货节” 启动，美团助力2800个区县 “年货送到家”](#)

· 一加新任命：原OPPO中国区用户运营负责人李杰担任中国区总裁
» 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。