

淘宝店铺

## 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 - 697, 文章 - 0, 评论 - 311, 阅读 - 173万

### 导航

博客园

首页

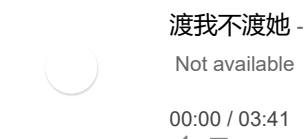
新随笔

联系

订阅 

管理

### 公告



1 渡我不渡她

2 小镇姑娘

3 PDD洪荒之力

 加入QQ群

昵称：杨奉武

园龄：5年8个月

粉丝：607

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

Android(22)  
Android 开发(8)  
C# 开发(4)  
CH395Q学习开发(9)  
ESP32学习开发(8)  
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)

## 9-网络芯片CH395Q学习开发-模块使用Socket0作为UDP和电脑上上位机UDP局域网通信

<p><iframe name="ifd" src="https://mnifdv.cn/resource/cnblogs/LearnCH395Q" frameborder="0" scrolling="auto" width="100%" height="1500"></iframe></p>

## 网络芯片CH395Q学习开发

开发板链接:[开发板链接](#)

模组原理图:[模组原理图](#)

### 资料源码下载链

接:<https://github.com/yangfengwu45/CH395Q.c>

- [学习Android](#)  
教程中搭配的Android，C#等教程如上，各个教程正在整理。
- [1-硬件测试使用说明](#)
- [2-学习资料说明,测试通信,获取硬件版本,程序移植说明](#)
- [3-芯片初始化,网线连接检测实验](#)
- [4-关于中断检测和DHCP实验](#)
- [5-模块使用Socket0作为TCP客户端和电脑上上位机TCP服务器局域网通信](#)
- [6-模块使用Socket0-3作为4路TCP客户端和电脑上上位机TCP服务器局域网通信](#)
- [7-模块使用Socket0-5作为6路TCP客户端和电脑上上位机TCP服务器局域网通信\(Socket缓存区配置\)](#)
- [8-模块使用Socket0作为TCP服务器和电脑上上位机TCP客户端局域网通信\(单连接和多连接\)](#)
- [9-模块使用Socket0作为UDP和电脑上上位机UDP局域网通信](#)

ESP8266 LUA开发基础入门篇  
备份(22)  
ESP8266 SDK开发(32)  
ESP8266 SDK开发基础入门篇  
备份(30)  
GPRS Air202 LUA开发(11)  
HC32F460(华大) +  
BC260Y(NB-IOT) 物联网开发  
(5)  
NB-IOT Air302 AT指令和LUA  
脚本语言开发(25)  
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+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(58)  
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(10)  
感想(6)  
软件安装使用: MQTT(8)  
软件安装使用: OpenResty(6)  
数据处理思想和程序架构(24)  
数据库学习开发(12)  
更多

#### 最新评论

1. Re:C#委托+回调详解  
好文，撒也不说了，直接收  
藏！  
--杨咩咩plus
2. Re:2-STM32 替换说明-  
CKS32, HK32, MM32,  
APM32, CH32, GD32,  
BLM32, AT32(推荐), N32,  
HC华大系列  
有用，谢谢！  
--你跟游戏过吧

#### 阅读排行榜

1. ESP8266使用详解(AT,LUA,  
SDK)(172077)
2. 1-安装MQTT服务器(Windo  
ws),并连接测试(96491)
3. ESP8266刷AT固件与node  
mcu固件(63751)

## 上位机UDP局域网通信

- 
- 
- 
- 
- 

## 说明

这节演示一下模块使用Socket0作为UDP和电脑上位机  
UDP局域网通信

提醒:无论是SPI,USART,并口,程序操作步骤都是一样的!  
只是不同的接口发指令发给模块,然后用不同的接收接收  
数据而已.

## 测试本节代码(STM32F103xxxx)

1.用户可以使用杜邦线根据自己的情况设置和连接引脚

- 4. 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(62544)
- 5. 有人WIFI模块使用详解(38095)
- 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)(35378)
- 7. 关于TCP和MQTT之间的转换(32225)
- 8. android 之TCP客户端编程(31284)
- 9. android客服端+eps8266+单片机,+路由器之远程控制系统(31131)
- 10. C#中public与private与static(30940)

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)

```
2  #ifndef __CH395SPI_H_
3  #define __CH395SPI_H_
4
5  #include "CH395INC.H"
6
7  /*****配置GPIO (根据自己的修改)*****/
8  //时钟
9  #define CH395_CONFIG_SPI_CLK() ( RCC_APB1PeriphClockCmd( RCC_APB1Periph_SPI2,ENABLE) )
10 #define CH395_CONFIG_GPIO_CLK() ( RCC_APB2PeriphClockCmd( RCC_APB2Periph_GPIOA | RCC_APB2Peri
11 //设置使用的SPI
12 #define USE_SPI SPI2
13 //SPI_CS -- 连接模块SCS引脚
14 #define CH395_CS_PORT GPIOB
15 #define CH395_CS_PIN GPIO_Pin_12
16 //SPI_CLK -- 连接模块SCK引脚
17 #define CH395_CLK_PORT GPIOB
18 #define CH395_CLK_PIN GPIO_Pin_13
19 //SPI_MISO -- 连接模块SDO引脚
20 #define CH395_MISO_PORT GPIOB
21 #define CH395_MISO_PIN GPIO_Pin_14
22 //SPI_MOSI -- 连接模块SDI引脚
23 #define CH395_MOSI_PORT GPIOB
24 #define CH395_MOSI_PIN GPIO_Pin_15
25 //RST -- 连接模块RST引脚
26 #define CH395_RST_PORT GPIOA
27 #define CH395_RST_PIN GPIO_Pin_8
28 //TX -- 连接模块Tx引脚
29 #define CH395_TX_PORT GPIOA
30 #define CH395_TX_PIN GPIO_Pin_3
31 //INT -- 连接模块INT引脚 (检测到该引脚低电平信号之后再获取数据)
32 #define CH395_INT_PORT GPIOA
33 #define CH395_INT_PIN GPIO_Pin_0
34 /*****/
```

2,注意!

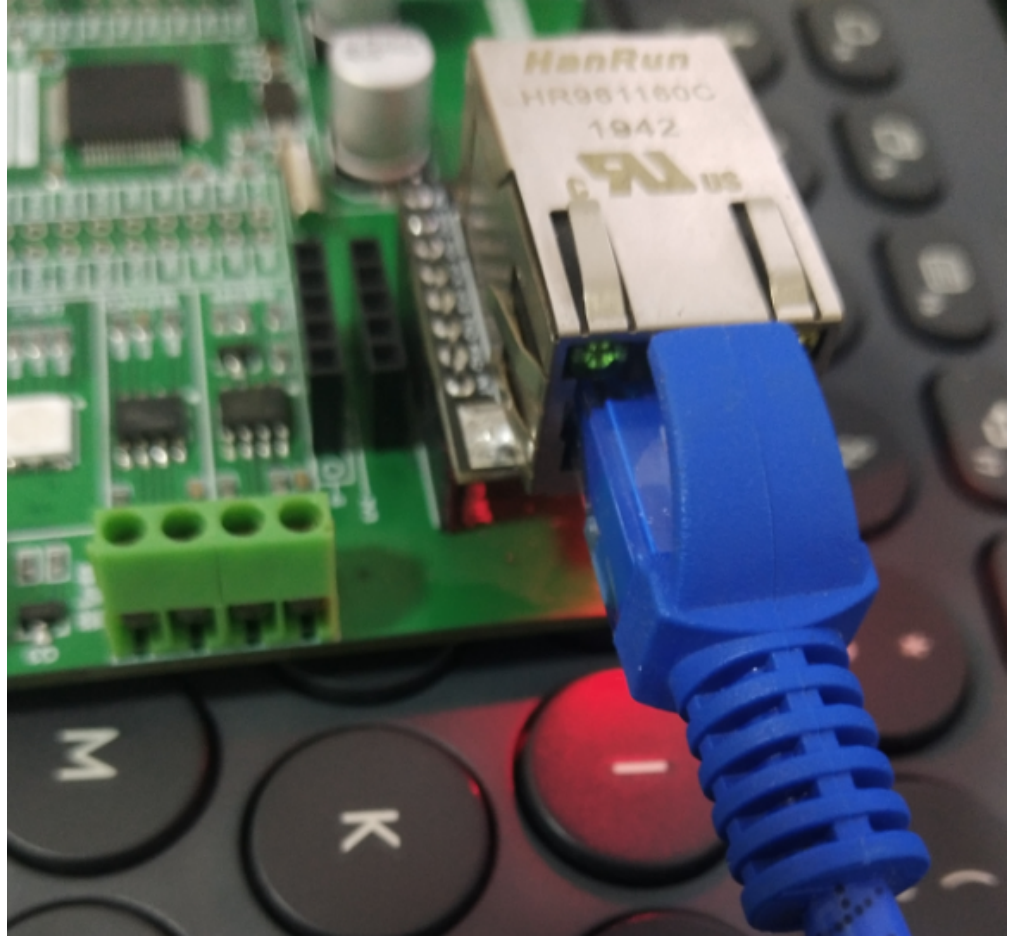
要想模块使用SPI通信,模块的TX引脚需要在模块重启之前设置为低电平.

上面的引脚分配把模块的TX引脚接到了单片机的PA3上,也就是串口2的RX上,如果用户使用了串口2,请注意!

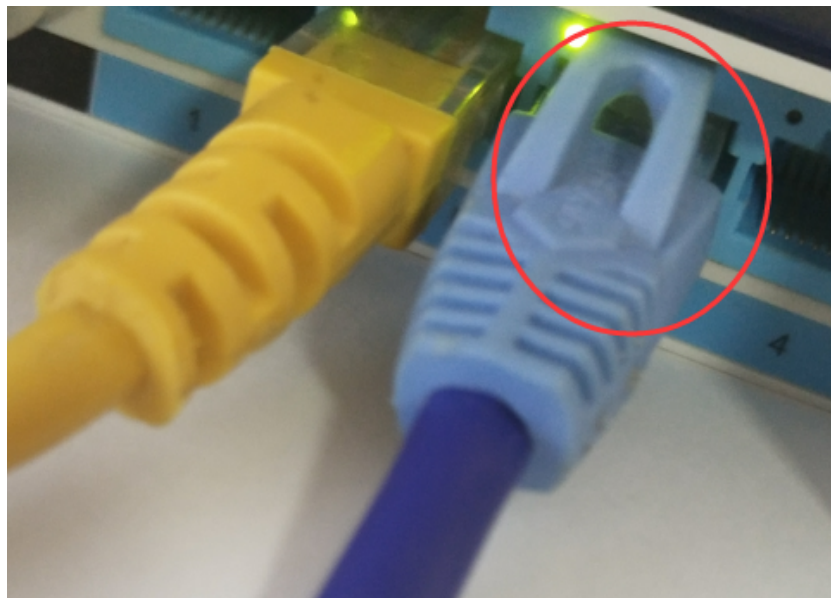
CH395 与单片机之间支持三种通讯接口: 8 位并行接口、SPI 同步串行接口、异步串口。在芯片上电复位时, CH395 将采样 SEL 和 TXD 引脚的状态, 根据这 2 个引脚状态的组合选择通讯接口, 参考下表 (表中 X 代表不关心此位, 0 代表低电平, 1 代表高电平或者悬空)。

SEL 引脚	TXD 引脚	选择通讯接口
1	1	异步串口
1	0	SPI 接口
0	1	8 位并口
0	0	错误接口

3.把模块用网线和路由器或者交换机(和上位机在同一个局域网下)



注意,连接路由器或者交换机的时候是连接其LAN口.





**WAN端口：连接网线**

**LAN端口：连接电脑（任选一个端口就行）**

#### 4.查看自己电脑的IP地址

我的为 192.168.0.103



#### 5.把程序里面的UDP目的地址改为电脑的IP地址

目的端口号6666为电脑端UDP的端口号,用户根据自己的修改

本地端口号1000为模块自身的UDP端口号



```
timer.c  usart.c  main.c  CH395CMD.H  CH395INC.H  CH395CMD.C  timer.h
26 芯片共有48块缓存区,每个缓存区512字节
27 芯片共有8个Socket,默认把48块缓存区分给了Socket0,Socket1,Socket2,Socket3
28 这四个Socket,每个 Socket 使用8块缓存区作为接收,4块缓存区作为发送,
29 即Socket0,Socket1,Socket2,Socket3的接收区各为512*8 = 4KB
30 即Socket0,Socket1,Socket2,Socket3的发送区各为512*4 = 2KB
31 如果要使用Socket4,Socket5,Socket6,Socket7需要重新分配缓存区
32 */
33
34
35 /*存储网络接收的数据*/
36 #define recv_buff_len 1500
37 unsigned char recv_buff[recv_buff_len];
38
39 char ch395_version=0;//获取版本号
40
41 unsigned char buf[20];
42 int ch395_status=0;//获取中断事件
43
44 /* socket 相关定义*/
45 UINT8 SocketIndex = 0; /* Socket 索引(0,1,2,3,4,5,6,7) */
46 UINT8 SocketDesIP[4] = {192,168,0,103}; /* Socket 目的IP地址 */
47 UINT16 SocketDesPort = 6666; /* Socket 目的端口 */
48 UINT16 SocketSourPort = 1000; /* Socket 本地端口 */
49
50 /**
```

## 6.编译下载到单片机

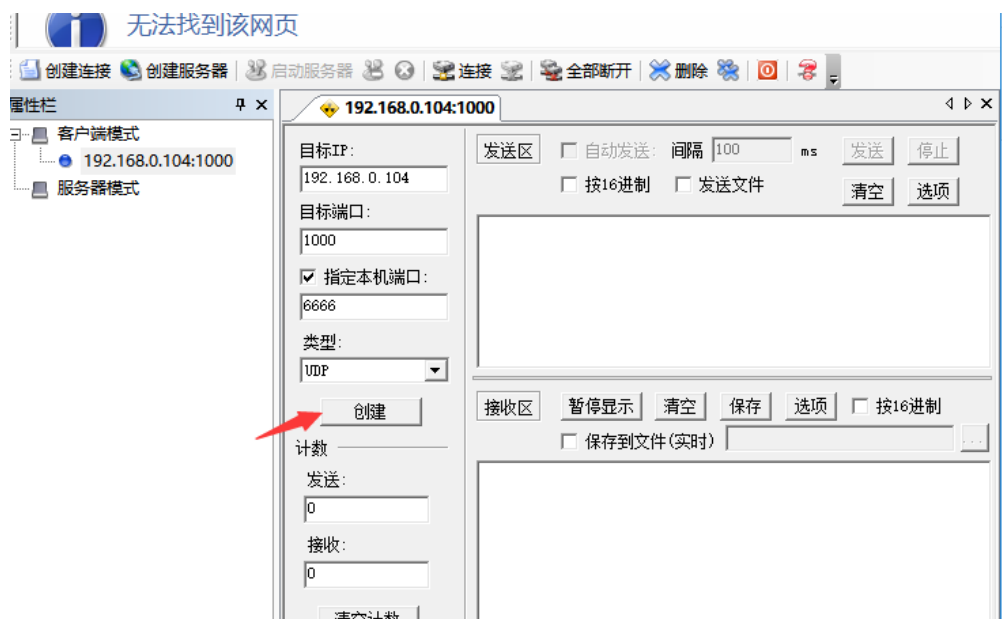
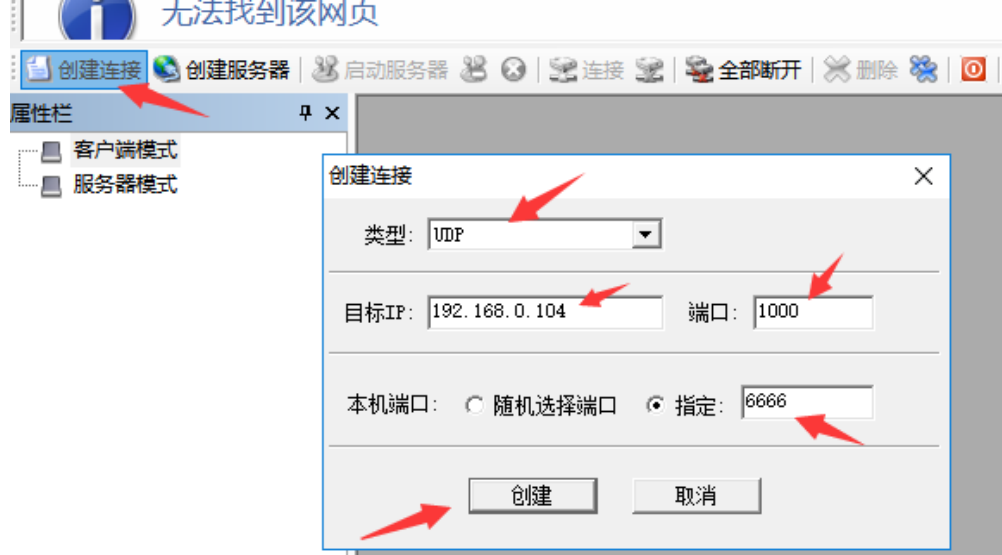
正常情况会打印模块的IP地址等信息

我的模块的IP地址是: 192.168.0.104

```
ATK XCOM V2.0
start
PHY_CONNECTED
IP:192.168.0.104
GWIP:192.168.0.1
Mask:255.255.255.0
DNS1:192.168.1.1
DNS2:192.168.0.1
```

## 6.打开电脑端TCP调试助手,并配置UDP

名称	修改日期
COMNET.exe	2021/6
config.ini	2021/6
Log_202103.txt	2021/6
TCPUDPDebug102_Setup.exe	2020/7



## 7.上位机发送数据

单片机程序里面设置的接收什么数据就返回什么数据



```
ATK
XCOM V2.0

start
PHY_CONNECTED
IP:192.168.0.104
GWIP:192.168.0.1
Mask:255.255.255.0
DNS1:192.168.1.1
DNS2:192.168.0.1

receive len = 6
111111
```

## 程序说明

### 1.初始化UDP

UDP是面向无连接的,所以只需要配置一下.



```

timer.c  usart.c  main.c  CH395CMD.H  CH395INC.H  CH395CMD.C  timer.h
128 CH395_PORT_INIT();
129 //复位 CH395
130 CH395_RST();
131
132 IWDG_Init(IWDG_Prescaler_256,156*10);
133
134 /*获取芯片版本*/
135 while((ch395_version = CH395CMDGetVer()) < 0x40)
136 {
137     printf("CH395CMDGetVer ERR\r\n");
138     delay_ms(100);
139 }
140
141 /*测试命令, 按位取反返回说明测试通过*/
142 while(CH395CMDCheckExist(0x55) != 0xaa)
143 {
144     printf("\r\nCH395CMDCheck ERR\r\n");
145     delay_ms(100);
146 }
147
148 /*初始化模块:成功返回 0 */
149 while(CH395CMDInitCH395() != 0)
150 {
151     printf("\r\nCH395CMDInitCH395 ERR\r\n");
152     delay_ms(100);
153 }
154
155 /*初始化UDP*/
156 while( ch395_socket_udp_init(SocketIndex,SocketDesIP,SocketDesPort,SocketSourPort) != 0)
157 {
158     printf("\r\nch395_socket_udp_init ERR\r\n");
159     delay_ms(100);
160 }
161
162 printf("\r\nstart\r\n");
163 while(1)

```

```

timer.c  usart.c  main.c  CH395CMD.H  CH395INC.H  CH395CMD.C  timer.h
42 int ch395_status=0;//获取中断事件
43
44 /* socket 相关定义*/
45 UINT8 SocketIndex = 0; /* Socket 索引 (0,1,2,3,4,5,6,7) */
46 UINT8 SocketDesIP[4] = {192,168,0,103}; /* Socket 目的IP地址 */
47 UINT16 SocketDesPort = 6666; /* Socket 目的端口 */
48 UINT16 SocketSourPort = 1000; /* Socket 本地端口 */
49
50 /**
51 * @brief 初始化socket
52 * @param sockindex Socket索引 (0,1,2,3,4,5,6,7)
53 * @param ipaddr 目的地址
54 * @param desprot 目的端口号
55 * @param surprot 本地端口号
56 * @retval 0:初始化成功; others:初始化失败
57 * @warning None
58 * @example
59 */
60 char ch395_socket_udp_init(UINT8 sockindex,UINT8 *ipaddr,UINT16 desprot,UINT16 surprot)
61 {
62     CH395SetSocketDesIP(sockindex,ipaddr); /* 目的地址 */
63     CH395SetSocketProtType(sockindex,PROTO_TYPE_UDP); /* 协议类型 */
64     CH395SetSocketDesPort(sockindex,desprot); /* 目的端口号 */
65     CH395SetSocketSourPort(sockindex,surprot); /* 本地端口号 */
66     if(CH395OpenSocket(sockindex) !=0) /* 打开Socket */
67     {
68         return 1;
69     }
70     return 0;
71 }
72
73

```

2.模块连接路由器通信需要启用DHCP,并打印模块分得的地址信息

```
timer.c  usart.c  main.c  CH395CMD.H  CH395INC.H  CH395CMD.C  timer.h

162     printf("\r\nstart\r\n");
163     while(1)
164     {
165         IWDG_Feed();//喂狗
166         //INT1引脚产生低电平中断以后进去判断
167         if(Query395Interrupt())
168         {
169             /*获取中断事件*/
170             if(ch395_version>=0x44)
171             {
172                 ch395_status = CH395CMDGetGlobIntStatus_ALL();
173             }
174             else
175             {
176                 ch395_status = CH395CMDGetGlobIntStatus();
177             }
178
179             /* 处理PHY改变中断*/
180             if(ch395_status & GINT_STAT_PHY_CHANGE)
181             {
182                 if(CH395CMDGetPHYStatus() == PHY_DISCONN)//网线断开
183                 {
184                     printf("\r\nPHY_DISCONN\r\n");
185                 }
186                 else//网线连接
187                 {
188                     printf("\r\nPHY_CONNECTED\r\n");
189                     CH395DHCPEnable(1);//启动DHCP
190                 }
191             }
192
193             /* 处理DHCP/PPPOE中断 */
194             if(ch395_status & GINT_STAT_DHCP)
195             {
196                 if(CH395GetDHCPStatus() == 0)//DHCP OK
197                 {
198                     CH395GetIPInf(buf);//获取IP,网关和子网掩码
199                     printf("IP:%d.%d.%d.%d\r\n",buf[0],buf[1],buf[2],buf[3]);
200                     printf("GWIP:%d.%d.%d.%d\r\n",buf[4],buf[5],buf[6],buf[7]);
201                     printf("Mask:%d.%d.%d.%d\r\n",buf[8],buf[9],buf[10],buf[11]);
202                     printf("DNS1:%d.%d.%d.%d\r\n",buf[12],buf[13],buf[14],buf[15]);
203                     printf("DNS2:%d.%d.%d.%d\r\n",buf[16],buf[17],buf[18],buf[19]);
204                 }
205             }
206         }
207     }
```

### 3.在中断检测事件里面处理Socket相关事件(本例中使用的Socket 0)

```

timer.c  usart.c  main.c  CH395CMD.H  CH395INC.H  CH395CMD.C  timer.h
201         printf("GWIP:%d.%d.%d.%d\r\n",buf[4],buf[5],buf[6],buf[7]);
202         printf("Mask:%d.%d.%d.%d\r\n",buf[8],buf[9],buf[10],buf[11]);
203         printf("DNS1:%d.%d.%d.%d\r\n",buf[12],buf[13],buf[14],buf[15]);
204         printf("DNS2:%d.%d.%d.%d\r\n",buf[16],buf[17],buf[18],buf[19]);
205     }
206 }
207
208 /* 处理不可达中断, 读取不可达信息 */
209 if(ch395_status & GINT_STAT_UNREACH){
210     CH395CMDGetUnreachIPPT(buf);
211 }
212
213 /* 处理IP冲突中断, 建议重新修改CH395的 IP, 并初始化CH395*/
214 if(ch395_status & GINT_STAT_IP_CONFLI){
215
216 }
217 /* 处理 SOCK0 中断 */
218 if(ch395_status & GINT_STAT_SOCK0){
219     ch395_socket_udp_interrupt(SocketIndex);
220 }
221 /* 处理 SOCK1 中断 */
222 if(ch395_status & GINT_STAT_SOCK1){
223
224 }
225 /* 处理 SOCK2 中断 */
226 if(ch395_status & GINT_STAT_SOCK2){
227
228 }
229 /* 处理 SOCK3 中断 */

```

```

timer.c  usart.c  main.c  CH395CMD.H  CH395INC.H  CH395CMD.C  timer.h
72
73
74 /**
75  * @brief  socket处理函数(把此函数放到全局socket中断里面)
76  * @param  sockindex  Socket索引(0,1,2,3,4,5,6,7)
77  * @param  None
78  * @param  None
79  * @param  None
80  * @retval None
81  * @warning None
82  * @example
83  */
84 void ch395_socket_udp_interrupt(UINT8 sockindex)
85 {
86     UINT8 sock_int_socket;
87     UINT16 len;
88
89     /* 获取socket 的中断状态 */
90     sock_int_socket = CH395GetSocketInt(sockindex);
91
92     /* 发送缓冲区空闲, 可以继续写入要发送的数据 */
93     if(sock_int_socket & SINT_STAT_SENBUF_FREE)
94     {
95     }
96
97     /* 发送完成中断 */
98     if(sock_int_socket & SINT_STAT_SEND_OK)
99     {
100     }
101
102     /* 接收数据中断 */
103     if(sock_int_socket & SINT_STAT_RECV)
104     {
105         len = CH395GetRecvLength(sockindex);/* 获取当前缓冲区内数据长度 */
106         printf("\r\nreceive len = %d\r\n",len);
107         if(len == 0)return;
108         if(len > recv_buff_len)len = recv_buff_len;
109         CH395GetRecvData(sockindex,len,recv_buff);/* 读取数据 */
110
111         /*把接收的数据返回给UDP*/
112         CH395SendData(sockindex,recv_buff,len);
113
114         /*使用串口打印接收的数据*/
115         PutData(&rb_t_usart1_send,recv_buff,len);
116         USART_ITConfig(USART1, USART_IT_TXE, ENABLE);
117     }
118 }

```

## 4.提醒

用户也可以开多个UDP测试,不过如果要使用Socket4-7

的时候记得重新配置下内存分配.

当然也可以接着开TCP服务器,TCP客户端. 可以同时进行.

分类: [CH395Q学习开发](#)

好文要顶

关注我

收藏该文



杨奉武

关注 - 1

粉丝 - 607

0

0

« 上一篇: [8-网络芯片CH395Q学习开发-模块使用Socket0作为TCP服务器和电脑上位机TCP客户端局域网通信\(单连接和多连接\)](#)

posted on 2021-06-12 23:59 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】百度智能云618年中大促，限时抢购，新老用户同享超值折扣

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

【推荐】阿里云爆品销量榜单出炉，精选爆款产品低至0.55折

【推荐】限时秒杀！国云大数据魔镜，企业级云分析平台

【推荐】华为应用软件专题日 | 生态市场企业特惠GO

#### 园子动态：

- 致园友们的一封检讨书：都是我们的错
- 数据库实例 CPU 100% 引发全站故障
- 发起一个开源项目：博客引擎 fluss

#### 最新新闻：

- 滴滴的中长期动力源
  - 奈雪的茶真的盈利了吗？
  - “杀疯了” 的剧本杀，如何敲开年轻人的门？
  - BOSS直聘上市，到底谁在隐身？
  - 穷人在内卷，富人在漏税
- » 更多新闻...

#### 历史上的今天：

- 2021-06-12 6-网络芯片CH395Q学习开发-模块使用Socket0-3作为4路TCP客户端和电脑上位...
- 2021-06-12 5-网络芯片CH395Q学习开发-模块使用Socket0作为TCP客户端和电脑上位机TCP...

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, 加入群聊。