

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 - 699, 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

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

 加入QQ群

昵称：杨奉武

园龄：5年8个月

粉丝：607

关注：1

搜索

找找看

谷歌搜索

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(11)

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)

11-网络芯片CH395Q学习开发-模块使用Socket0作为UDP组播(多播)通信,MAC地址过滤

<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和电脑上位机UDP局域网通信](#)

■ [10-模块使用Socket0作为UDP广播通信](#)

■ [11-模块使用Socket0作为UDP组播\(多播\)通信MAC地址过滤](#)

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)(172084)
2. 1-安装MQTT服务器(Windo
ws),并连接测试(96505)
3. ESP8266刷AT固件与node
mcu固件(63760)

[信,MAC地址过滤](#)

-
-
-

最近感慨的事情

1.关于教育工作者,科技工作者,公务人员

不知为什么,在大众的想法里,教育工作者,科技工作者,公务人员就应该喝米粥吃咸菜!

好久之前的袁隆平老爷子摸车事件,还有现在的戴建业老爷子事件.

对于教育工作者,科技工作者,公务人员他们不是应该让他们过得更好一点吗?

只有让他们过的好,让他们更加的富足,让他们更加的长寿,他们就会对这个社会做出更多的贡献.

教育工作者,科技工作者,公务人员,只要是人家正常的挣钱养家,我才不管人家穿金戴银,只要人家过得开心,我也开心.

评价一个人不要以他个人来看,要以整个社会来看.

2.关于现在的教育

关于孩子的思想道德,人格:

现在的教育一直在改,最近有些地方开始禁止公开学生成绩了.

其实就我的观点重之之重还在于父母和老师.

4. 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(62578)
5. 有人WIFI模块使用详解(38096)
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)(35385)
7. 关于TCP和MQTT之间的转换(32235)
8. android 之TCP客户端编程(31288)
9. android客服端+eps8266+单片机+路由器之远程控制系统(31135)
10. C#中public与private与static(30946)

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)

咱国家的父母大部分认为学校教育是第一教育.其实就整个来看

家庭教育是第一教育,社会教育是第二教育,学校教育是第三教育.

孩子的思想道德,人格大部分是取决于父母!

所以如果你的孩子因为思想道德,人格而出现了各种事情,做父母的应该先好好考虑自己!

关于老师:

老祖宗几千年来就总结了: 师者,传道,授业,解惑者也!

但是现在看看现在的老师,大部分都只是授业的老师.把第一是传道都给丢了!

前些日子看了个咱国的老师去外国授课,然后有个老师说只要有成绩才能怎么怎么滴. 这老师肤浅!

咱国家的大部分老师有个特点: 现在的老师只是因为当初考试考的好,考到研究生考上博士,是考试考得好的老师但不是真正爱好的老师!

有的网友说中国是培养人才,外国是挑选人才. 其实并不对!

外国才是真正的培养人才! 他们遵循传道,授业,解惑,而且懂得因材施教.

建国初期有好多说得上名字的大家,为啥以后大家就很少了? 所以:很多事情不要怪孩子,怪教育!

说明

这节演示一下模块使用Socket0作为UDP组播(多播)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

```
ch395cmd.h CH395INC.H CH395SPI.C usart.c delay.c timer.c main.c delay.h CH395SPI.H
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!

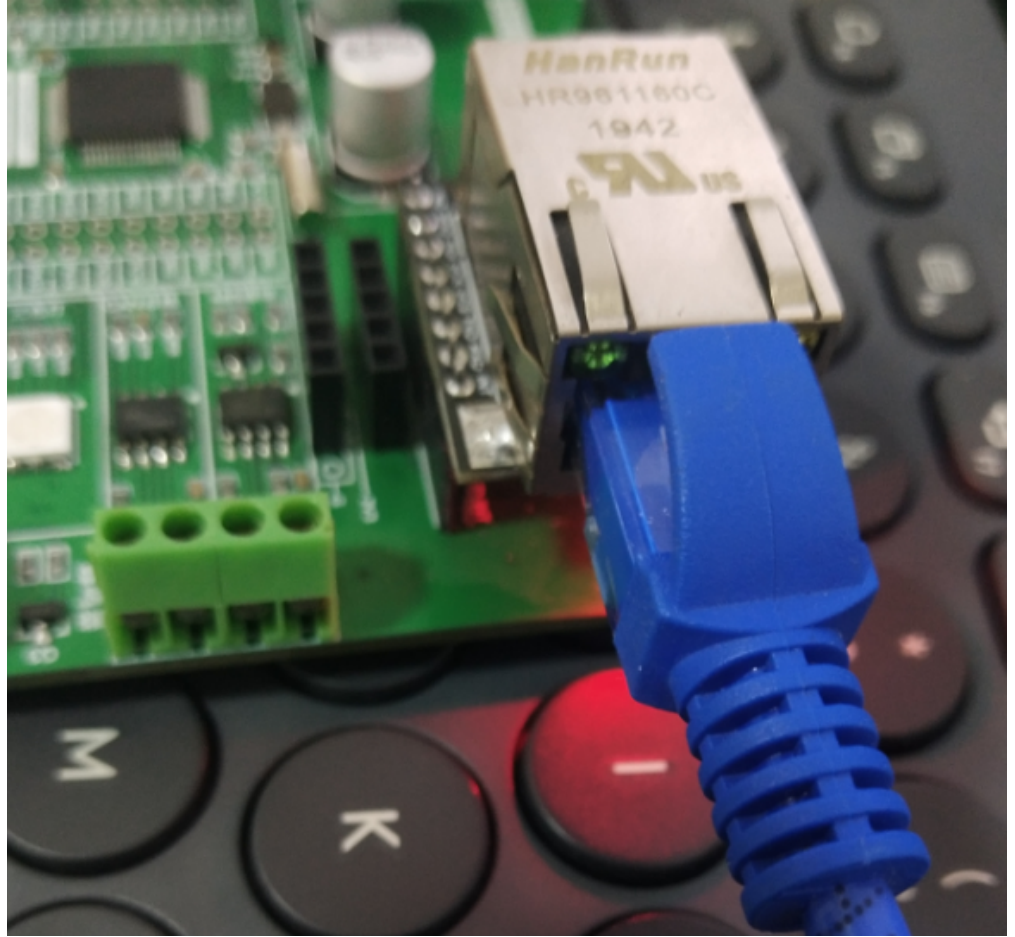
要想模块使用SPI通信,模块的TX引脚需要在模块重启之前设置为低电平.

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!

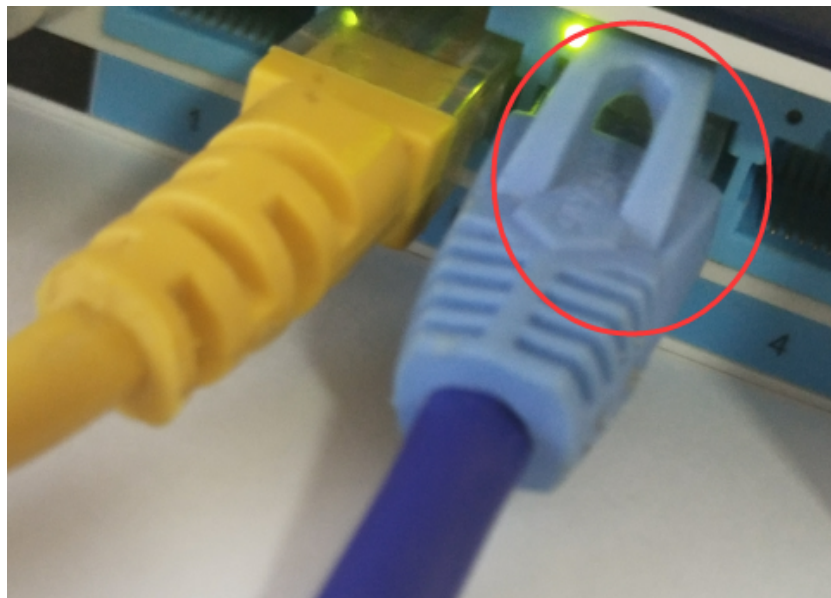
CH395 与单片机之间支持三种通讯接口: 8 位并行接口、SPI 同步串行接口、异步串口。在芯片上电复位时, CH395 将采样 SEL 和 TXD 引脚的状态, 根据这 2 个引脚状态的组合选择通讯接口, 参考下表 (表中 X 代表不关心此位, 0 代表低电平, 1 代表高电平或者悬空)。

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.设置多播地址,多播的目的端口号,本地端口号

注:下面还有个广播IP,255.255.255.255 ,是因为执行多播通信需要把UDP的目的地址配置为广播IP

多播MAC地址是为了MAC过滤,后面会讲,用户不需要设置,默认不过滤MAC

```

timer.c  usart.c  main.c  CH395CMD.H  CH395INC.H  CH395CMD.C  timer.h
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
47  UINT8 SocketDesIP[4] = {255,255,255,255}; /* 配置的UDP目的地址 */
48  UINT16 SocketDesPort = 5566; /* 配置的UDP目的端口号(多播目的端口号) */
49  UINT16 SocketSourPort = 5000; /* 配置的UDP本地端口号(多播本地端口号) */
50
51  UINT8 SocketMultiIP[4] = {224,1,1,1}; /* 多播的地址 */
52  UINT8 SocketMultiMac[6] = {1,0,0x5e,3,4,5}; /* 对应多播Mac地址 */
53
54  #endif

```

5.编译下载到单片机

正常情况会打印模块的IP地址等信息

```

ATK
COM XCOM V2.0

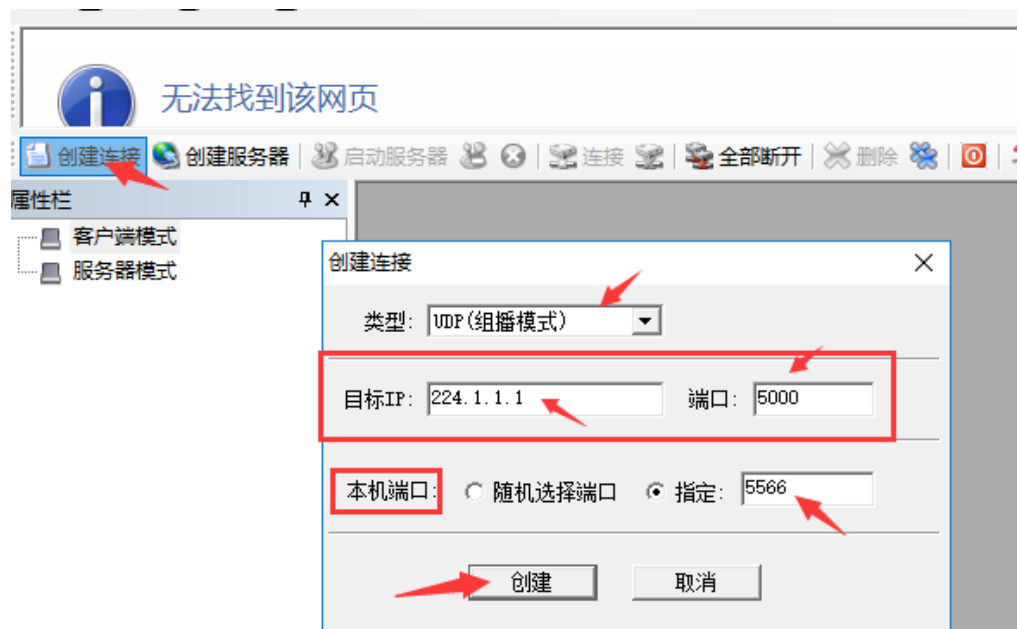
start

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

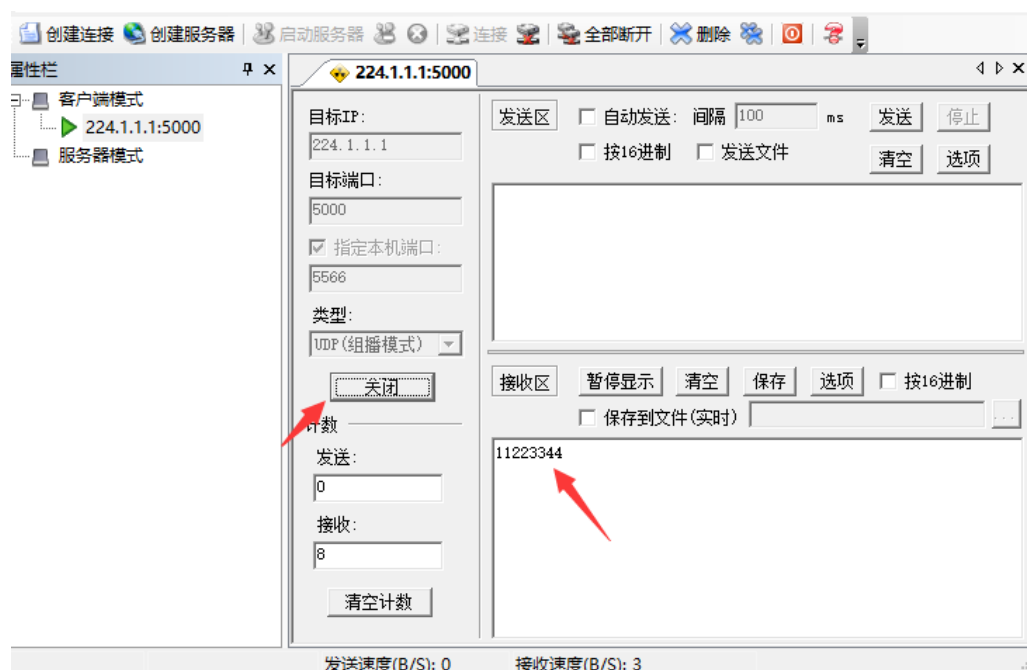
```

6.打开电脑端网络调试助手,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



创建以后会每隔一段时间收到单片机控制模块发送的组播信息

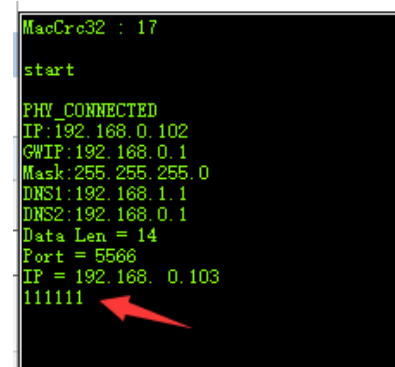


7.调试助手发送数据给单片机

单片机程序里面写的是把接收的服务器返回给服务器,并使用串口打印接收的消息



串口会打印上位机的IP地址,本身端口号,还有发送的消息



程序说明

1.初始化UDP

注意: 配置的UDP地址为广播地址: 255.255.255


```

timer.c  usart.c  main.c  CH395CMD.H  CH395INC.H  CH395CMD.C  timer.h
149  CH395_RST();
150
151  IWDG_Init(IWDG_Prescaler_256,156*10);
152
153  /*获取芯片版本*/
154  while((ch395_version = CH395CMDGetVer()) < 0x40)
155  {
156      printf("CH395CMDGetVer ERR\r\n");
157      delay_ms(100);
158  }
159
160  /*测试命令，按位取反返回说明测试通过*/
161  while(CH395CMDCheckExist(0x55) != 0xaa)
162  {
163      printf("\r\nCH395CMDCheck ERR\r\n");
164      delay_ms(100);
165  }
166
167  /*初始化模块:成功返回 0 */
168  while(CH395CMDInitCH395() != 0)
169  {
170      printf("\r\nCH395CMDInitCH395 ERR\r\n");
171      delay_ms(100);
172  }
173
174  /*初始化UDP*/
175  while( ch395_socket_udp_init(SocketIndex,SocketDesIP,SocketDesPort,SocketSourPort) != 0)
176  {
177      printf("\r\nch395_socket_udp_init ERR\r\n");
178      delay_ms(100);
179  }
180

```

```

timer.c  usart.c  main.c  CH395CMD.H  CH395INC.H  CH395CMD.C  timer.h
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
47  UINT8  SocketDesIP[4] = {255,255,255,255}; /* 配置的UDP目的地址 */
48  UINT16 SocketDesPort = 5566; /* 配置的UDP目的端口号(多播目的端口号) */
49  UINT16 SocketSourPort = 5000; /* 配置的UDP本地端口号(多播本地端口号) */
50
51  UINT8  SocketMultiIP[4] = {224,1,1,1}; /* 多播的地址 */
52  UINT8  SocketMultiMac[6] = {1,0,0x5e,3,4,5}; /* 对应多播Mac地址 */
53
54  /**
55   * @brief  初始化socket
56   * @param  sockindex  Socket索引(0,1,2,3,4,5,6,7)
57   * @param  ipaddr  目的地址
58   * @param  desprot  目的端口号
59   * @param  surprot  本地端口号
60   * @retval  0:初始化成功; others:初始化失败
61   * @warning None
62   * @example
63   */
64  char ch395_socket_udp_init(UINT8 sockindex,UINT8 *ipaddr,UINT16 desprot,UINT16 surprot)
65  {
66      CH395SetSocketDesIP(sockindex,ipaddr); /* 目的地址 */
67      CH395SetSocketProtType(sockindex,PROTO_TYPE_UDP); /* 协议类型 */
68      CH395SetSocketDesPort(sockindex,desprot); /* 目的端口号 */
69      CH395SetSocketSourPort(sockindex,surprot); /* 本地端口号 */
70      if(CH395OpenSocket(sockindex) !=0) /* 打开Socket */
71      {
72          return 1;
73      }
74      return 0;
75  }
76

```

2.设置MAC地址过滤

```
timer.c  usart.c  main.c  CH395CMD.H  CH395INC.H  CH395CMD.C  timer.h
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
47 UINT8 SocketDesIP[4] = {255,255,255,255}; /* 配置的UDP目的地址 */
48 UINT16 SocketDesPort = 5566; /* 配置的UDP目的端口号(多播) */
49 UINT16 SocketSourPort = 5000; /* 配置的UDP本地端口号(多播) */
50
51 UINT8 SocketMultiIP[4] = {224,1,1,1}; /* 多播的地址 */
52 UINT8 SocketMultiMac[6] = {1,0,0x5e,3,4,5}; /* 对应多播Mac地址 */
53
54 /**
```

```
timer.c  usart.c  main.c  CH395CMD.H  CH395INC.H  CH395CMD.C  timer.h
173
174 /*初始化UDP*/
175 while( ch395_socket_udp_init(SocketIndex,SocketDesIP,SocketDesPort,Soc
176 {
177     printf("\r\nch395_socket_udp_init ERR\r\n");
178     delay_ms(100);
179 }
180
181 /*设置MAC过滤*/
182 MacCrc32 = CH395CRCRet6Bit(SocketMultiMac);
183 printf("MacCrc32 : %2x\n", (UINT16)MacCrc32);
184 if(MacCrc32 > 31)
185 {
186     MacCrc32 -=31;
187     Hash1 |= ((UINT32)1<<MacCrc32);
188 }
189 else
190 {
191     Hash0 |= ((UINT32)1<<MacCrc32);
192 }
193 //0x1a:00011010 (发送使能,接收使能,接收所有数据)
194 CH395CMDSetMACFilt(0x1a,Hash0,Hash1);
195 delay_ms(100);
196
197
198 printf("\r\nstart\r\n");
199 while(1)
```

注意:CH395CMDSetMACFilt 函数是设置MAC地址过滤

第一个参数具体含义

我设置的为0x1a, 也就是 0001 1010 意味着不执行MAC过滤

000(保留)

bit4 = 1(SEND_ENABLE发送使能)

bit3 = 1(RECV_ENABLE接收使能)

bit1 = 1(RECV_ALL接收所有数据)

该命令用于设置 MAC 过滤模式。MAC 可以设置多种过滤模式。该命令需要输入 9 个字节数据，第 1 个字节为过滤模式，该数据的各位含义如下：

位	名称	描述
[5:7]	-	保留
4	SEND_ENABLE	发送使能
3	RCV_ENABLE	接收使能
2	RCV_MULTIPKT	接收多播包
1	RCV_ALL	接收所有数据
0	RCV_BROADPKT	接收广播包

以上各位为 1 表示开启，为 0 表示关闭。CH395 复位后默认 RCV_BROADPKT, RCV_ENABLE, SEND_ENABLE 开启。

下表为各位的含义：

RCV_ENABLE	RCV_ALL	RCV_BROADPKT	RCV_MULTIPKT	描述
1	0	0	0	接收与 MAC 地址相匹配的数据包
1	1	X	X	接收所有数据包
1	0	1	0	接收与 MAC 地址相匹配的数据包 接收广播数据包
1	0	0	1	接收与 MAC 地址相匹配的数据包 接收多播数据包
1	0	1	1	接收与 MAC 地址相匹配的数据包 接收广播数据包 接收多播数据包
0	X	X	X	禁止接收

第 2 至第 5 字节为 HASH0（哈希表 0），第 6 至第 9 字节为 HASH1（哈希表 1），HASH0 与 HASH1 仅在多播开启下有效。

HASH0 和 HASH1 共同构成一个 64 位的 HASH 表，0~31 位为 HASH0，32~63 位为 HASH1。

HASH 表计算方法：将多播地址使用标准的以太网冗余校验（CRC32）计算出一个 32 位的 CRC 值，使用此 CRC 值的高 6 位作为索引值，将 HASH 表对应的位写 1。例如，多播地址计算出 CRC 值的高 6 位为 32，则应该将 HASH1 的第 0 位写 1。

3,关于MAC过滤,我也没搞明白这个模块咋用

但是这节吧后面地址对应起来,然后设置为MAC过滤模式有效果



```

32  */
33
34
35  /*存储网络接收的数据*/
36  #define rcv_buff_len 1500
37  unsigned char rcv_buff[rcv_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
47  UINT8 SocketDesIP[4] = {255,255,255,255}; /* 配置的UDP目的地址 */
48  UINT16 SocketDesPort = 5566; /* 配置的UDP目的端口号(多播目的端) */
49  UINT16 SocketSourPort = 5000; /* 配置的UDP本地端口号(多播本地端) */
50
51  UINT8 SocketMultiIP[4] = {224,1,1,1}; /* 多播的地址 */
52  UINT8 SocketMultiMac[6] = {1,0,0x5e,1,1,1}; /* 对应多播Mac地址 */
53
54  /**
55   * @brief 初始化socket
56   * @param sockindex socket索引(0 1 2 3 4 5 6 7)

```

```
timer.c | usart.c | main.c | CH395CMD.H | CH395INC.H | CH395CMD.C | timer.h
179     }
180
181     /*设置MAC过滤*/
182     MacCrc32 = CH395CRCRet6Bit(SocketMultiMac);
183     printf("MacCrc32 : %2x\n", (UINT16)MacCrc32);
184     if (MacCrc32 > 31)
185     {
186         MacCrc32 -= 31;
187         Hash1 |= ((UINT32)1 << MacCrc32);
188     }
189     else
190     {
191         Hash0 |= ((UINT32)1 << MacCrc32);
192     }
193     //0x1a:00011010 (发送使能,接收使能,接收所有数据)
194     CH395CMDSetMACFilt(0x1d, Hash0, Hash1);
195     delay_ms(100);
196
```

RECV_ENABLE	RECV_ALL	RECV_BROADPKT	RECV_MULTIPKT	描述
1	0	0	0	接收与 MAC 地址相匹配的数据包
1	1	X	X	接收所有数据包
1	0	1	0	接收与 MAC 地址相匹配的数据包 接收广播数据包
1	0	0	1	接收与 MAC 地址相匹配的数据包 接收多播数据包
1	0	1	1	接收与 MAC 地址相匹配的数据包 接收广播数据包 接收多播数据包
0	X	X	X	禁止接收

4.因为是局域网,连接了路由器,所以需要启用DHCP

```

timer.c  usart.c  main.c  CH395CMD.H  CH395INC.H  CH395CMD.C  timer.h
209
210 //INT引脚产生低电平中断以后进去判断
211 if(Query395Interrupt())
212 {
213     /*获取中断事件*/
214     if(ch395_version>=0x44)
215     {
216         ch395_status = CH395CMDGetGlobIntStatus_ALL();
217     }
218     else
219     {
220         ch395_status = CH395CMDGetGlobIntStatus();
221     }
222
223     /* 处理PHY改变中断*/
224     if(ch395_status & GINT_STAT_PHY_CHANGE)
225     {
226         if(CH395CMDGetPHYStatus() == PHY_DISCONN)//网线断开
227         {
228             printf("\r\nPHY_DISCONN\r\n");
229         }
230         else//网线连接
231         {
232             printf("\r\nPHY_CONNECTED\r\n");
233             CH395DHCPEnable(1);//启动DHCP
234         }
235     }
236
237     /* 处理DHCP/PPPOE中断 */
238     if(ch395_status & GINT_STAT_DHCP)
239     {
240         if(CH395GetDHCPStatus() == 0)//DHCP OK
241         {
242             CH395GetIPInf(buf); //获取IP,网关和子网掩码
243             printf("IP:%d.%d.%d.%d\r\n",buf[0],buf[1],buf[2],buf[3]);
244             printf("GWIP:%d.%d.%d.%d\r\n",buf[4],buf[5],buf[6],buf[7]);
245             printf("Mask:%d.%d.%d.%d\r\n",buf[8],buf[9],buf[10],buf[11]);
246             printf("DNS1:%d.%d.%d.%d\r\n",buf[12],buf[13],buf[14],buf[15]);
247             printf("DNS2:%d.%d.%d.%d\r\n",buf[16],buf[17],buf[18],buf[19]);
248         }
249     }

```

5.每隔一段时间发送一条多播数据出去

```

timer.c  usart.c  main.c  CH395CMD.H  CH395INC.H  CH395CMD.C  timer.h
188 }
189 else
190 {
191     Hash0 |= ((UINT32)1<<MacCrc32);
192 }
193 //0x1a:00011010 (发送使能,接收使能,接收所有数据)
194 CH395CMDSetMACFilt(0x1a,Hash0,Hash1);
195 delay_ms(100);
196
197 printf("\r\nstart\r\n");
198 while(1)
199 {
200     IWDG_Feed();//喂狗
201
202     /*每隔一段时间发送多播数据*/
203     if(Timer2Cnt>3000)
204     {
205         Timer2Cnt=0;
206         // CH395SendData(SocketIndex,"11223344",8)
207         CH395UDPSendTo("11223344",8,SocketMultiIP, SocketDesPort, SocketIndex);
208     }
209 }

```

6.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
248     }
249     }
250
251     /* 处理不可达中断, 读取不可达信息 */
252     if(ch395_status & GINT_STAT_UNREACH){
253         CH395CMDGetUnreachIPPT(buf);
254     }
255
256     /* 处理IP冲突中断, 建议重新修改CH395的 IP, 并初始化CH395*/
257     if(ch395_status & GINT_STAT_IP_CONFLI){
258         printf("\r\nGINT_STAT_IP_CONFLI\r\n");
259     }
260     /* 处理 SOCK0 中断 */
261     if(ch395_status & GINT_STAT_SOCK0){
262         ch395_socket_udp_interrupt(SocketIndex);
263     }
264     /* 处理 SOCK1 中断 */
265     if(ch395_status & GINT_STAT_SOCK1){
266
267     }
268     /* 处理 SOCK2 中断 */
269     if(ch395_status & GINT_STAT_SOCK2){
270

```

```

timer.c  usart.c  main.c  CH395CMD.H  CH395INC.H  CH395CMD.C  timer.h
83  * @param None
84  * @retval None
85  * @warning None
86  * @example
87  **/
88 void ch395_socket_udp_interrupt(UINT8 sockindex)
89 {
90     UINT8 sock_int_socket;
91     UINT16 len;
92     UINT16 tmp,port;
93
94     /* 获取socket 的中断状态 */
95     sock_int_socket = CH395GetSocketInt(sockindex);
96     /* 发送缓冲区空闲, 可以继续写入要发送的数据 */
97     if(sock_int_socket & SINT_STAT_SENBUF_FREE)
98     {
99     }
100     /* 发送完成中断 */
101     if(sock_int_socket & SINT_STAT_SEND_OK)
102     {
103     }
104     /* 接收数据中断 */
105     if(sock_int_socket & SINT_STAT_RECV)
106     {
107         len = CH395GetRecvLength(sockindex);/* 获取当前缓冲区内数据长度 */
108         if(len == 0) return;
109         if(len > recv_buff_len) len = recv_buff_len;
110         CH395GetRecvData(sockindex, len, recv_buff);/* 读取数据 */
111
112         /*打印Socket信息*/
113         //0,1存储接收的数据个数
114         tmp = ((UINT16)recv_buff[1] << 8) + recv_buff[0];
115         //2,3存储Socket端口号
116         port = ((UINT16)recv_buff[3] << 8) + recv_buff[2];
117         printf("Data Len = %d\r\n",tmp);
118         printf("Port = %d\r\n",port);
119         //4,5,6,7存储Socket IP地址
120         printf("IP = %2d.%2d.%2d.%2d\r\n", (UINT16)recv_buff[4], (UINT16)recv_buff[5], (UINT16)recv_buff[6], (UINT16)recv_buff[7]);
121
122         //从8开始存储真实数据
123         CH395UDPSendTo(&recv_buff[8], (len - 8), &recv_buff[4], port, sockindex);//返回数据
124
125         /*使用串口打印接收的数据*/
126         PutData(&rb_t_usart1_send, &recv_buff[8], len);
127         USART_ITConfig(USART1, USART_IT_TXE, ENABLE);
128     }
129 }
130

```

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: 10-网络芯片CH395Q学习开发-模块使用Socket0作为UDP广播通信


[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：

- 风波不断，留给这届家长的在线教育机构不多了
- 95后职场图鉴：躺平、裸辞、焦虑、拼命，对PUA大胆说不
- 维基百科不缺钱，为什么募捐的时候总要道德绑架？
- 翻红的摩尔庄园，曾是5000万95后的第一款网游
- 牵手阿里云！奇瑞发布首款汽车生态品牌
- » 更多新闻...



单片机,物联网,上位机,...

扫一扫二维码，入群聊。