

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 - 702, 文章 - 0, 评论 - 311, 阅读 - 174万

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：608

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(14)

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)

14-1-网络芯片CH395Q学习开发-WEB服务器-网页到底是啥,

web服务器是啥, 网页如何显示的显示图片和视频

```
<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)(172116)
2. 1-安装MQTT服务器(Windo
ws),并连接测试(96603)
3. ESP8266刷AT固件与node
mcu固件(63793)

信,MAC地址过滤

- [12-模块使用Socket0作为IP RAW模式和调试助手
测试通信](#)
- [13-模块使用Socket0作为MAC RAW](#)
- [14-1-WEB服务器-网页到底是啥, web服务器是啥,
网页如何显示的显示图片和视频](#)
-
-
-
-
-
-

说明

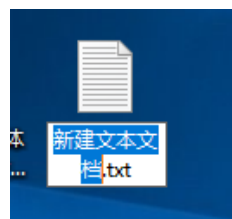
这节演示一下模块作为WEB服务器,浏览器输入模组IP进行网页访问.

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.

网页是什么?

1.写个最简短的网页程序,新建一个文本文档



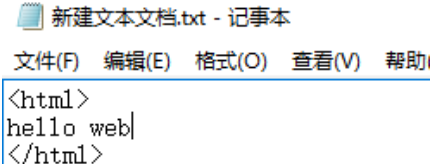
2,写上下面一段

```
<html>  
hello web  
</html>
```

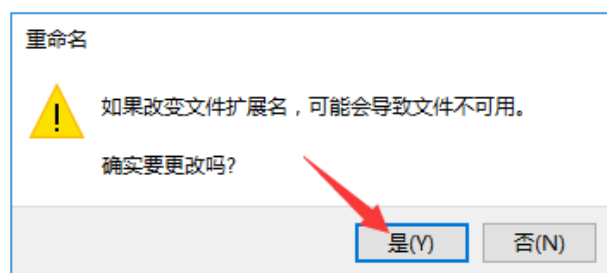
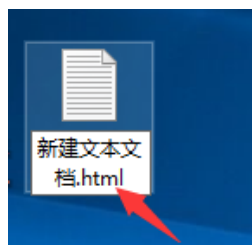
4. 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(62648)
5. 有人WIFI模块使用详解(38115)
6. (一)基于阿里云的MQTT远程控制(Android 连接MQTT服务器,ESP8266连接MQTT服务器实现远程通信控制----简单的连接通信)(35406)
7. 关于TCP和MQTT之间的转换(32263)
8. android 之TCP客户端编程(31314)
9. android客户端+eps8266+单片机+路由器之远程控制系统(31142)
10. C#中public与private与static(30980)

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)

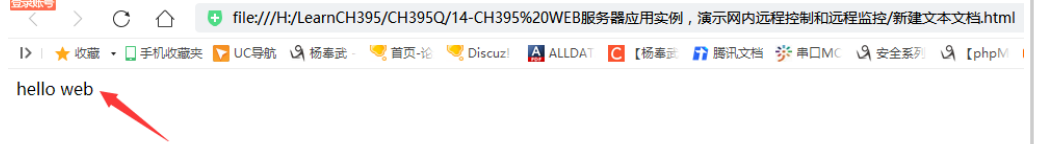


3.把文件后缀名改为 .html



4.双击打开(会使用浏览器打开)





5.上面是最简洁的网页了(具体网页的学习自行学习哈)

首先记住一句话:程序是什么? 程序是一种规定!

网页程序当然也是一种规定,咱平时学程序学的就是规定,或者说是规约.

上面的 `<html> hello web </html>` ,浏览器读取以后就开始解析.

网页程序的规定是前后加上 `<html>` `</html>`

网页更多一点的话是下面的样子.

`<head>` 这个里面写一些开头的配置,比如写整个文件的格式啦,设置网页的标题啦,执行js程序啦什么的 `</head>`

`<body>` 这个里面就是放一些展示的控制件,也就是页面展示的东西
`</body>`





h1叫做标签的控件,这里面的字体显示比较大

h2也是叫做标签的控件,这里面的字体显示比较大

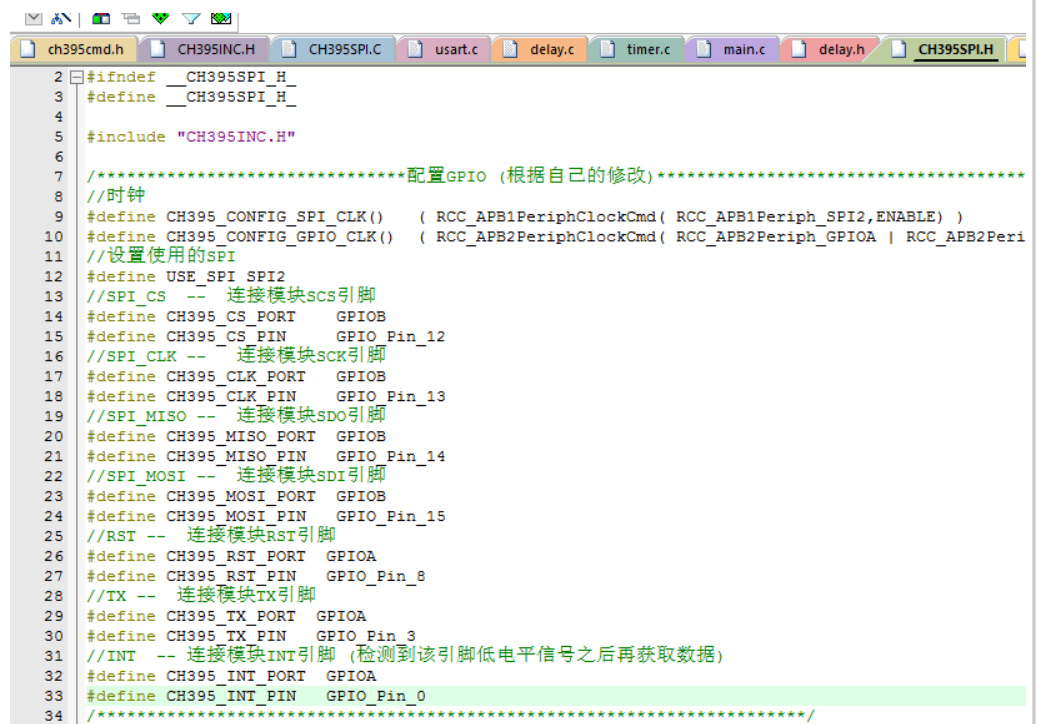
测试本节代码(STM32F103xxxx)

1.提供了3个程序,网页功能循序渐进

可以让用户彻底的理解网页是神么.

- STM32F10xSPI_1
- STM32F10xSPI_2
- STM32F10xSPI_3

2.用户可以使用杜邦线根据自己的情况设置和连接引脚



3,注意!

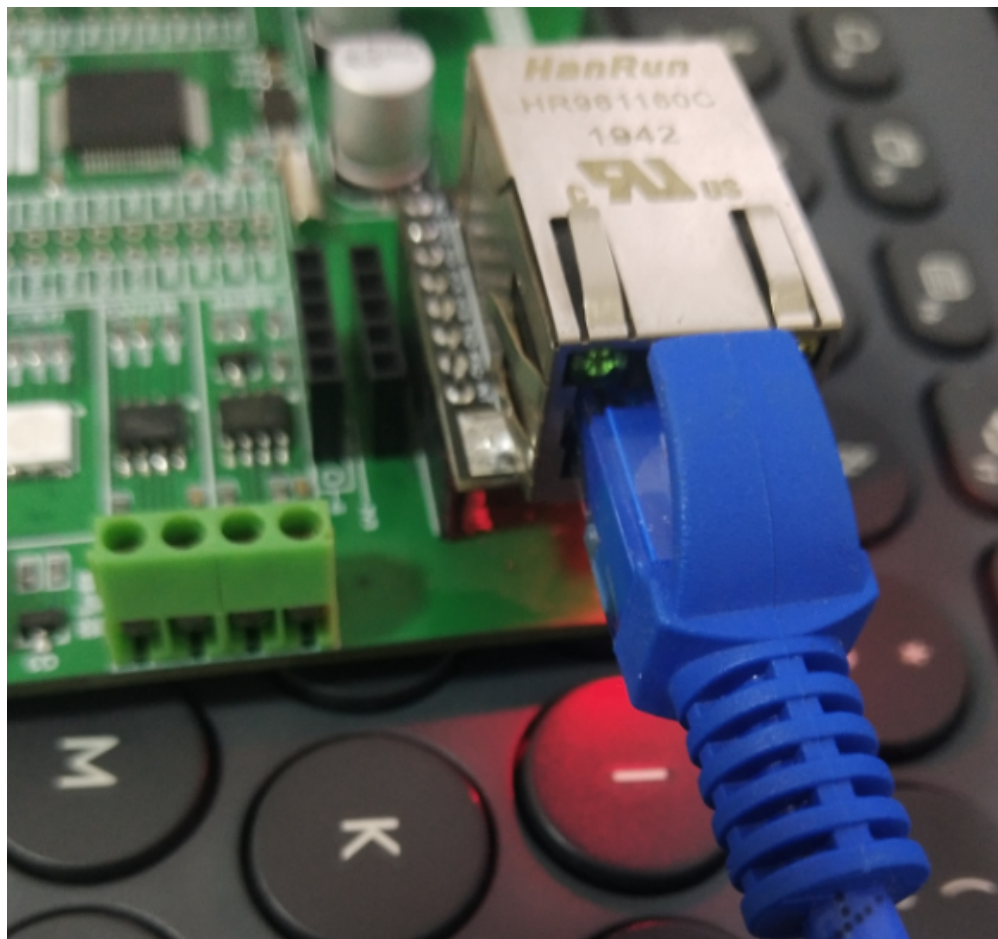
要想模块使用SPI通信,模块的TX引脚需要在模块重启之前设置为低电平.

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

4.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)



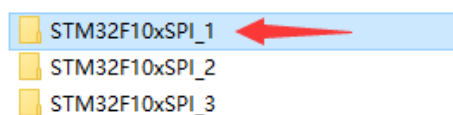
注意,连接路由器或者交换机的时候是连接其LAN口.



WAN端口：连接网线

LAN端口：连接电脑（任选一个端口就行）

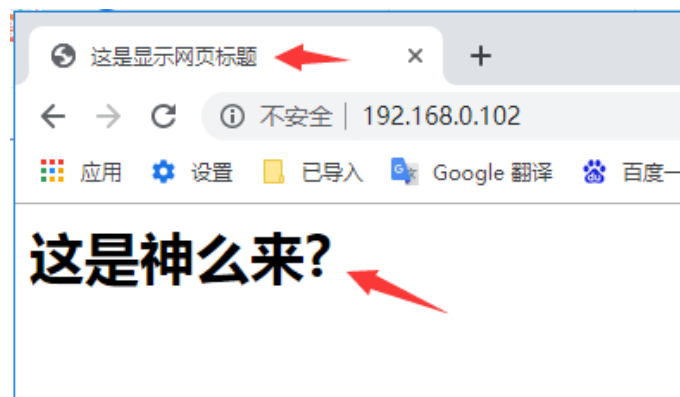
5.把第一个程序下载程序到单片机



正常情况下会打印模块的IP地址

```
ATK XCOM V2.0
CH395CMDGetVer =46
start
CH395TCPListen
PHY_CONNECTED
IP:192.168.0.102
GWIP:192.168.0.1
Mask:255.255.255.0
DNS1:192.168.1.1
DNS2:192.168.0.1
```

6.在浏览器上输出这个IP地址



程序说明

1.首先用户需要明白的事情

想让浏览器展示网页,其实就是把网页程序代码发给浏览器.

在浏览器上输入IP地址 192.168.0.102 然后回车,浏览器就会使用TCP去连接这个192.168.0.102 地址,端口号为80

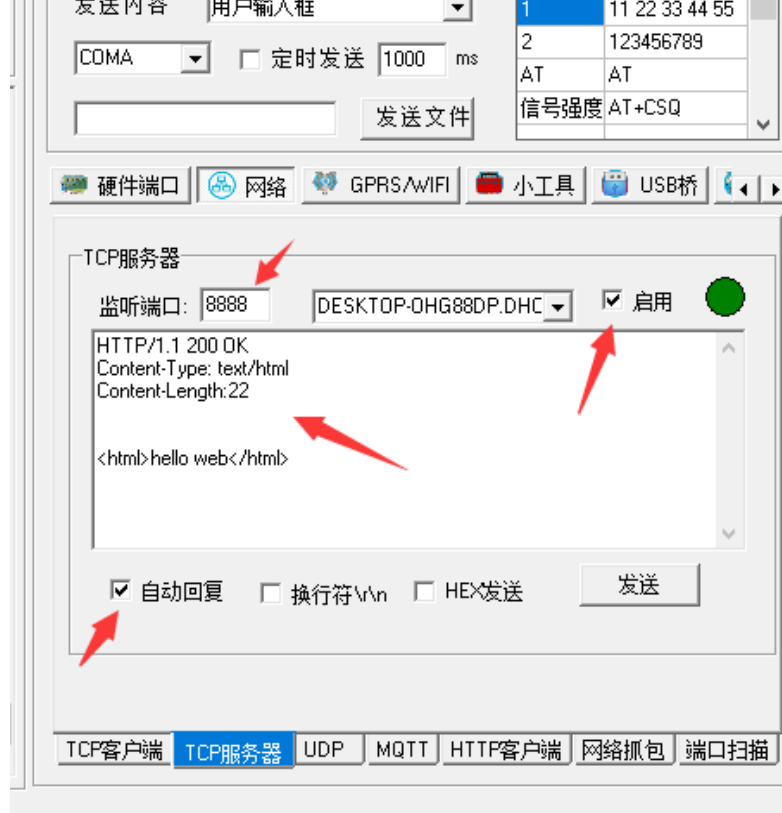
就是说不输入端口号,默认就是连接 192.168.0.102 端口号为80的TCP服务器去(一般叫WEB服务器).

也可以输入端口号,列如 192.168.0.102:8888

咱先看看下面的真实过程吧.

2,大家伙可以用TCP调试助手模拟为一个WEB服务器

开一个TCP服务器,端口号为8888,设置成自动回复网页数据



```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length:22

<html>hello web</html>
```

注意哈,22后面是两个换行,别给丢掉哈

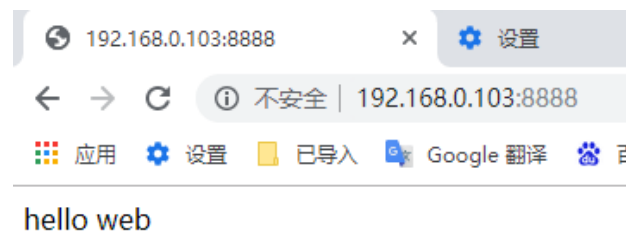
注意哈,22后面是两个换行,别给丢掉哈

注意哈,22后面是两个换行,别给丢掉哈

看下自己电脑的IP地址



然后在浏览器上输入本机的IP地址:8888 然后回车(建议使用google浏览器, 多刷新几下)



注意哈浏览器通过TCP发送两次数据给服务器(一般是两次,可以设置不让他获取图标)

第一次是询问文件数据 GET /

第二次是GET /favicon.ico 要咱网页最上面的标题栏左侧显示的图标
比如百度的



```
[2021/6/16 0:47:34 491] 192.168.0.103:1151 已连接
[2021/6/16 0:47:34 498] TCP服务器[192.168.0.103:1151]接收441字节:
[2021/6/16 0:47:34 499] GET / HTTP/1.1
Host: 192.168.0.103:8888
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/75.0.3770.100 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=
0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9

[2021/6/16 0:47:34 502] TCP Server发送文本数据87字节:
[2021/6/16 0:47:34 503] HTTP/1.1 200 OK
Content-Type: text/html
Content-Length:22


<html>hello web</html>
[2021/6/16 0:47:34 512] 192.168.0.103:1152 已连接
[2021/6/16 0:47:34 518] 192.168.0.103:1151 已断开
[2021/6/16 0:47:34 554] TCP服务器[192.168.0.103:1152]接收397字节:
[2021/6/16 0:47:34 557] GET /favicon.ico HTTP/1.1
Host: 192.168.0.103:8888
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/75.0.3770.100 Safari/537.36
Accept: image/webp,image/apng,image/*,*/*;q=0.8
Referer: http://192.168.0.103:8888/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9

[2021/6/16 0:47:34 560] TCP Server发送文本数据87字节:
[2021/6/16 0:47:34 567] HTTP/1.1 200 OK
Content-Type: text/html
Content-Length:22
```

一般规定哈,真实的web服务器接收到 GET / 路径是/
就会把index.html文件里面的数据返回给网页.
也可以在访问的时候告诉服务器我要的文件



```
[2021/6/16 0:47:34 573] 192.168.0.103:1152 已断开
[2021/6/16 0:58:41 640] 192.168.0.103:1672 已连接
[2021/6/16 0:58:41 660] 192.168.0.103:1673 已连接
[2021/6/16 0:58:41 673] TCP服务器[192.168.0.103:1672]接收420字节:
[2021/6/16 0:58:41 680] GET /1.txt HTTP/1.1
Host: 192.168.0.103:8888
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/75.0.3770.100 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=
0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
```



web服务器返回给网页的时候返回有两部分

上面的专业术语叫做 响应头部

下面的专业术语叫做 响应正文

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 22
```

```
<html>hello web</html>
```

提示:22代表下面响应正文的长度

3.程序里面初始化和启动TCP服务器监听

```

189     printf("\r\nCH395CMDCheck ERR\r\n");
190     delay_ms(100);
191 }
192
193 /*初始化模块:成功返回 0 */
194 while(CH395CMDInitCH395() != 0)
195 {
196     printf("\r\nCH395CMDInitCH395 ERR\r\n");
197     delay_ms(100);
198 }
199
200 printf("\r\nstart\r\n");
201 while(1)
202 {
203     IWDG_Feed();//喂狗
204
205
206     /*检测到没有启动服务器,则执行启动服务器*/
207     if(SocketServerStatus == 0)
208     {
209         if(ch395_socket_tcp_server_init(0,SocketServerPort) == 0)
210         {
211             if(CH395TCPListen(0) == 0) //Socket 0 启动TCP监听
212             {
213                 printf("\r\nCH395TCPListen\r\n");
214                 SocketServerStatus = 1;//服务器状态设置为启动监听
215             }
216         }
217     }
218 }

```

```

38
39 /* socket 相关定义*/
40 UINT16 SocketServerPort = 80; /*本地监听的 Socket 端口 */
41 char SocketServerStatus = 0;//SocketServer状态 0:未启动监听; 1:启动监听
42
43 //缓存服务器发送的响应头部的数据
44 char http_response_data[1024];
45
46 //缓存服务器发送的响应的正文
47 char http_html_data[1024]=
48 "<html>\r\n \
49 <head>\r\n \
50 <title>这是显示网页标题</title>\r\n \
51 </head> \
52 <body> \
53 <h1>这是神么来?</h1> \
54 </body> \
55 </html>";
56
57
58 /**
59  * @brief 初始化socket
60  * @param sockindex Socket索引(0,1,2,3,4,5,6,7)
61  * @param None
62  * @param None
63  * @param surprot 本地端口号
64  * @retval 0:初始化成功; others:初始化失败
65  * @warning None
66  * @example
67  */
68 char ch395_socket_tcp_server_init(UINT8 sockindex,UINT16 surprot)
69 {
70     CH395SetSocketProtType(sockindex,PROTO_TYPE_TCP); /* 协议类型 */
71     CH395SetSocketSourPort(sockindex,surprot); /* 本地端口号 */
72     if(CH395OpenSocket(sockindex) !=0) /* 打开Socket */
73     {
74         return 1;
75     }
76     return 0;
77 }

```

4.因为连接路由器,所以需要DHCP

```
main.c
218
219
220 //INT引脚产生低电平中断以后进去判断
221 if(Query395Interrupt())
222 {
223     /*获取中断事件*/
224     if(ch395_version>=0x44)
225     {
226         ch395_status = CH395CMDGetGlobIntStatus_ALL();
227     }
228     else
229     {
230         ch395_status = CH395CMDGetGlobIntStatus();
231     }
232
233     /* 处理PHY改变中断*/
234     if(ch395_status & GINT_STAT_PHY_CHANGE)
235     {
236         if(CH395CMDGetPHYStatus() == PHY_DISCONN)//网线断开
237         {
238             printf("\r\nPHY_DISCONN\r\n");
239         }
240         else//网线连接
241         {
242             printf("\r\nPHY_CONNECTED\r\n");
243             CH395DHCPEnable(1);//启动DHCP
244         }
245     }
246
247     /* 处理DHCP/PPPOE中断 */
248     if(ch395_status & GINT_STAT_DHCP)
249     {
250         if(CH395GetDHCPStatus() == 0)//DHCP OK
251         {
252             CH395GetIPInf(buf);//获取IP, 子网掩码和网关地址
253             printf("IP:%d.%d.%d.%d\r\n",buf[0],buf[1],buf[2],buf[3]);
254             printf("GWIP:%d.%d.%d.%d\r\n",buf[4],buf[5],buf[6],buf[7]);
255             printf("Mask:%d.%d.%d.%d\r\n",buf[8],buf[9],buf[10],buf[11]);
256             printf("DNS1:%d.%d.%d.%d\r\n",buf[12],buf[13],buf[14],buf[15]);
257             printf("DNS2:%d.%d.%d.%d\r\n",buf[16],buf[17],buf[18],buf[19]);
258         }
259     }
260 }
```

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


```

main.c
254     printf("GWIP:%d.%d.%d.%d\r\n",buf[4],buf[5],buf[6],buf[7]);
255     printf("Mask:%d.%d.%d.%d\r\n",buf[8],buf[9],buf[10],buf[11]);
256     printf("DNS1:%d.%d.%d.%d\r\n",buf[12],buf[13],buf[14],buf[15]);
257     printf("DNS2:%d.%d.%d.%d\r\n",buf[16],buf[17],buf[18],buf[19]);
258 }
259 }
260
261
262 /* 处理不可达中断, 读取不可达信息 */
263 if(ch395_status & GINT_STAT_UNREACH){
264     CH395CMDGetUnreachIPPT(buf);
265 }
266
267 /* 处理IP冲突中断, 建议重新修改CH395的 IP, 并初始化CH395*/
268 if(ch395_status & GINT_STAT_IP_CONFLI){
269
270 }
271 /* 处理 SOCK0 中断 */
272 if(ch395_status & GINT_STAT_SOCK0){
273     ch395_socket_tcp_client_interrupt(0);
274 }
275 /* 处理 SOCK1 中断 */
276 if(ch395_status & GINT_STAT_SOCK1){
277
278 }
279 /* 处理 SOCK2 中断 */
280 if(ch395_status & GINT_STAT_SOCK2){

```

接收到信息以后,没有去解析信息,直接返回的数据

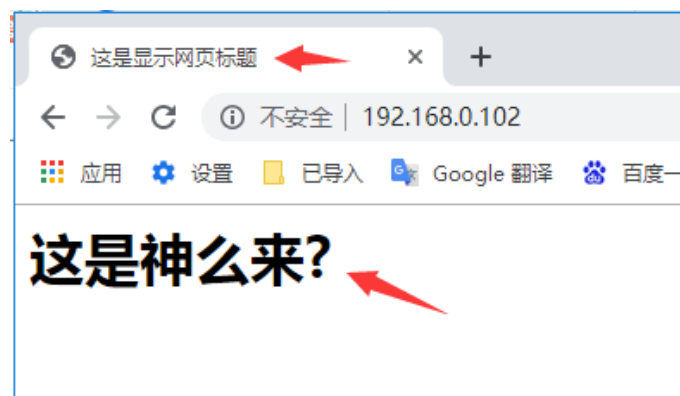
```

main.c
90 /**/
91 void ch395_socket_tcp_client_interrupt(UINT8 sockindex)
92 {
93     UINT8 sock_int_socket;
94     UINT16 len;
95
96     /* 获取socket 的中断状态 */
97     sock_int_socket = CH395GetSocketInt(sockindex);
98
99     /* 发送缓冲区空闲, 可以继续写入要发送的数据 */
100     if(sock_int_socket & SINT_STAT_SENBUF_FREE)
101     {
102
103     }
104
105     /* 发送完成中断 */
106     if(sock_int_socket & SINT_STAT_SEND_OK)
107     {
108     }
109
110     /* 接收数据中断 */
111     if(sock_int_socket & SINT_STAT_RECVD)
112     {
113         len = CH395GetRecvLength(sockindex);/* 获取当前缓冲区内数据长度 */
114         if(len == 0)return;
115         if(len > rcv_buff_len)len = rcv_buff_len;
116
117         CH395GetRecvData(sockindex,len,rcv_buff);/* 读取数据 */
118
119         /*使用串口打印接收的数据*/
120         printf("socket%d receive data:\r\n",sockindex);
121         PutData(&rb_t_uart1_send,rcv_buff,len);
122         USART_ITConfig(USART1, USART_IT_TXE, ENABLE);
123
124         /*响应头部*/
125         len = sprintf((char *)http_response_data,"HTTP/1.1 200 OK\r\n\
Content-Type: text/html\r\n\
Content-Length:%d\r\n\r\n",sizeof(http_html_data)
126 );
127 //发送响应头部
128 CH395SendData(sockindex, (UINT8*)http_response_data, len);
129 delay_ms(10);
130
131 /*发送响应正文*/
132 CH395SendData(sockindex, (unsigned char*)http_html_data, sizeof(http_html_data));
133 }
134
135
136

```

```
main.c
33
34 char ch395_version=0;//获取版本号
35
36 unsigned char buf[20];
37 int ch395_status=0;//获取中断事件
38
39 /* socket 相关定义*/
40 UINT16 SocketServerPort = 80;          /*本地监听的 Socket 端口 */
41 char SocketServerStatus = 0;//SocketServer状态 0:未启动监听; 1:启动监听
42
43 //缓存服务器发送的响应头部的数据
44 char http_response_data[1024];
45
46 //缓存服务器发送的响应的正文
47 char http_html_data[1024]=
48 "<html>\r\n \
49 <head>\r\n \
50 <title>这是显示网页标题</title>\r\n \
51 </head> \
52 <body> \
53 <h1>这是神么来?</h1> \
54 </body> \
55 </html>";
56
57
```

然后浏览器就会显示



网页访问链接显示图片或者视频又是怎么做到的呢?

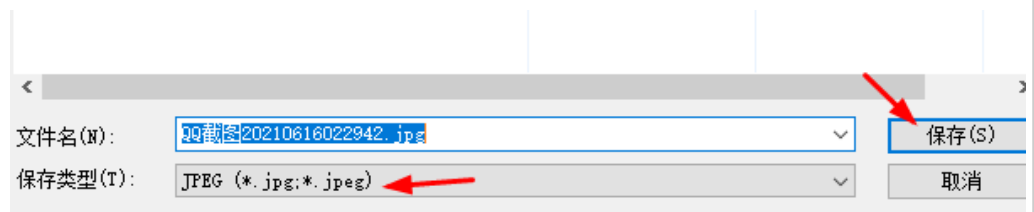
图片呢都有格式哈,列如 jpeg格式的图片

视频呢就是一张张图片切换显示的很快,就是视频啦.

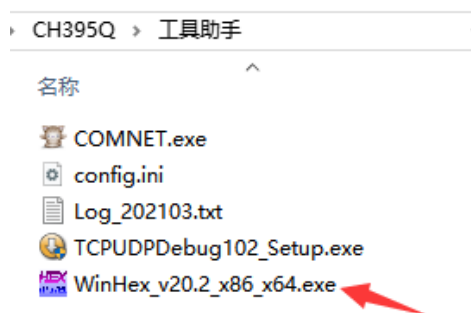
1.准备一张jpeg图片,用QQ截图,记得截的小一点哈

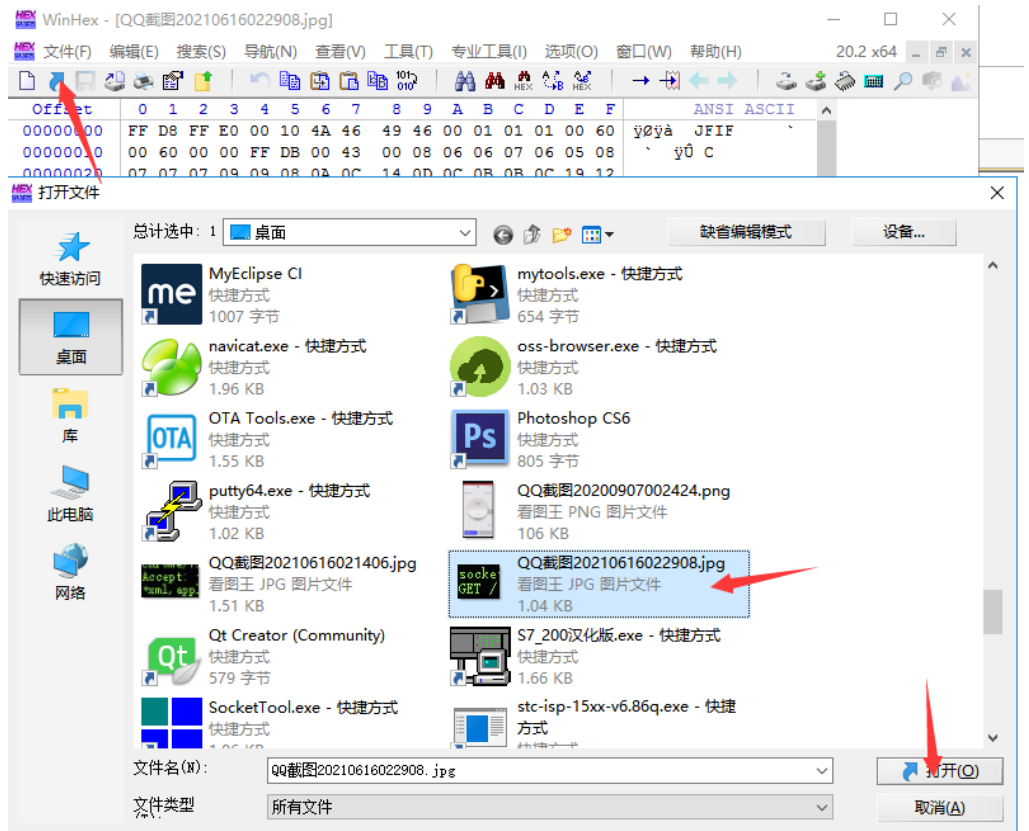


保存格式选择 jpeg

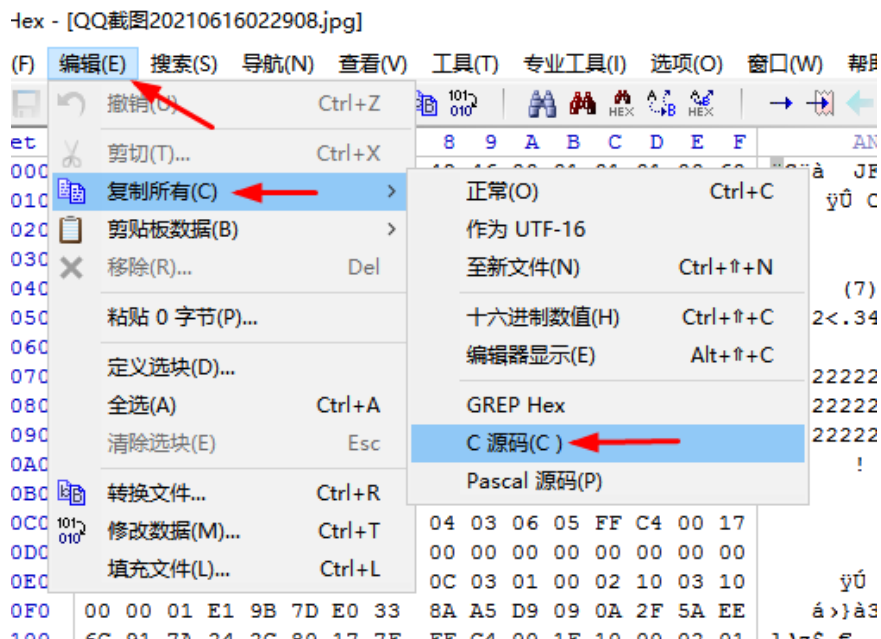


2.使用winhex打开这个图片





安装下面的步骤用鼠标点击一个C 源码,数据就被复制了



3.在程序里面 Ctrl + V 就可以粘贴出来图片jpeg的16进制格式的数据了

```
main.c
41 char SocketServerStatus = 0; //SocketServer状态 0:未启动监听; 1:启动监听
42
43 //缓存服务器发送的响应头部的数据
44 char http_response_data[1024];
45
46 //缓存服务器发送的响应的正文
47 char http_html_data[1024]=
48 "<html>\r\n \
49 <head>\r\n \
50 <title>这是显示网页标题</title>\r\n \
51 </head> \
52 <body> \
53 <h1>这是神么来?</h1> \
54 </body> \
55 </html>";
56
57
58 unsigned AnsiChar_data[1068] = {
59 0xFF, 0xD8, 0xFF, 0xE0, 0x00, 0x10, 0x4A, 0x46, 0x49, 0x46, 0x00, 0x01, 0x01, 0x01, 0x00, 0x60,
60 0x00, 0x60, 0x00, 0x00, 0xFF, 0xDB, 0x00, 0x43, 0x00, 0x08, 0x06, 0x06, 0x07, 0x06, 0x05, 0x08,
61 0x07, 0x07, 0x07, 0x09, 0x09, 0x08, 0x0A, 0x0C, 0x14, 0x0D, 0x0C, 0x0B, 0x0B, 0x0C, 0x19, 0x12,
62 0x13, 0x0F, 0x14, 0x1D, 0x1A, 0x1F, 0x1E, 0x1D, 0x1A, 0x1C, 0x1C, 0x20, 0x24, 0x2E, 0x27, 0x20,
63 0x22, 0x2C, 0x23, 0x1C, 0x1C, 0x28, 0x37, 0x29, 0x2C, 0x30, 0x31, 0x34, 0x34, 0x34, 0x1F, 0x27,
64 0x39, 0x3D, 0x38, 0x32, 0x3C, 0x2E, 0x33, 0x34, 0x32, 0xFF, 0xDB, 0x00, 0x43, 0x01, 0x09, 0x09,
65 0x09, 0x0C, 0x0B, 0x0C, 0x18, 0x0D, 0x0D, 0x18, 0x32, 0x21, 0x1C, 0x21, 0x32, 0x32, 0x32, 0x32,
66 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32,
67 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32,
68 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32,
69 0x00, 0x11, 0x08, 0x00, 0x1B, 0x00, 0x21, 0x03, 0x01, 0x22, 0x00, 0x02, 0x11, 0x01, 0x03, 0x11,
70 0x01, 0xFF, 0xC4, 0x00, 0x19, 0x00, 0x00, 0x03, 0x01, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00,
71 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x02, 0x04, 0x03, 0x06, 0x05, 0xFF, 0xC4, 0x00, 0x17,
72 0x01, 0x01, 0x01, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
73 0x00, 0x02, 0x03, 0x00, 0x01, 0xFF, 0xDA, 0x00, 0x0C, 0x03, 0x01, 0x00, 0x02, 0x10, 0x03, 0x10,
74 0x00, 0x00, 0x01, 0xE1, 0x9B, 0x7D, 0xE0, 0x33, 0x8A, 0xA5, 0xD9, 0x09, 0xA0, 0x2F, 0x5A, 0xEE,
75 0x6C, 0x91, 0x7A, 0x24, 0x2C, 0x80, 0x17, 0x7F, 0xFF, 0xC4, 0x00, 0x1F, 0x10, 0x00, 0x02, 0x01,
76 0x04, 0x02, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x02, 0x03, 0x01,
77 0x00, 0x04, 0x13, 0x14, 0x11, 0x23, 0x12, 0x20, 0x24, 0xFF, 0xDA, 0x00, 0x08, 0x01, 0x01, 0x00,
78 0x01, 0x05, 0x02, 0xC2, 0x70, 0xA9, 0x43, 0x20, 0xA2, 0xD8, 0xA5, 0x58, 0x8F, 0x1E, 0xB1, 0xD0,
79 0x4B, 0x4A, 0xD6, 0x76, 0xE5, 0x92, 0xE6, 0x2A, 0x85, 0xCD, 0x2B, 0x7A, 0xEB, 0x30, 0x61, 0xFD,
80 0xE5, 0xCF, 0x91, 0x75, 0xA3, 0xDB, 0xFF, 0xC4, 0x00, 0x19, 0x11, 0x00, 0x02, 0x03, 0x01, 0x00,
81 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x21, 0x01, 0x01, 0x10, 0x11,
82 0x12, 0xFF, 0xDA, 0x00, 0x08, 0x01, 0x03, 0x01, 0x01, 0x3F, 0x01, 0xD9, 0x18, 0xCE, 0x2F, 0xFF,
83 0xC4, 0x00, 0x1E, 0x11, 0x00, 0x01, 0x03, 0x04, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
```

改一下哈

```
main.c
38
39 /* socket 相关定义*/
40 UINT16 SocketServerPort = 80; //本地监听的 Socket 端口 *
41 char SocketServerStatus = 0; //SocketServer状态 0:未启动监听; 1:启动
42
43 //缓存服务器发送的响应头部的数据
44 char http_response_data[1024];
45
46 //缓存服务器发送的响应的正文
47 char http_html_data[1024]=
48 "<html>\r\n \
49 <head>\r\n \
50 <title>这是显示网页标题</title>\r\n \
51 </head> \
52 <body> \
53 <h1>这是神么来?</h1> \
54 </body> \
55 </html>";
56
57
58 unsigned char data_ipeg[1068] = {
59 0xFF, 0xD8, 0xFF, 0xE0, 0x00, 0x10, 0x4A, 0x46, 0x49, 0x46, 0x00, 0x01, 0x01, 0x01, 0x00, 0x60,
60 0x00, 0x60, 0x00, 0x00, 0xFF, 0xDB, 0x00, 0x43, 0x00, 0x08, 0x06, 0x06, 0x07, 0x06, 0x05, 0x08,
61 0x07, 0x07, 0x07, 0x09, 0x09, 0x08, 0x0A, 0x0C, 0x14, 0x0D, 0x0C, 0x0B, 0x0B, 0x0C, 0x19, 0x12,
62 0x13, 0x0F, 0x14, 0x1D, 0x1A, 0x1F, 0x1E, 0x1D, 0x1A, 0x1C, 0x1C, 0x20, 0x24, 0x2E, 0x27, 0x20,
63 0x22, 0x2C, 0x23, 0x1C, 0x1C, 0x28, 0x37, 0x29, 0x2C, 0x30, 0x31, 0x34, 0x34, 0x34, 0x1F, 0x27,
64 0x39, 0x3D, 0x38, 0x32, 0x3C, 0x2F, 0x33, 0x34, 0x32, 0xFF, 0xDB, 0x00, 0x43, 0x01, 0x09, 0x09,
```

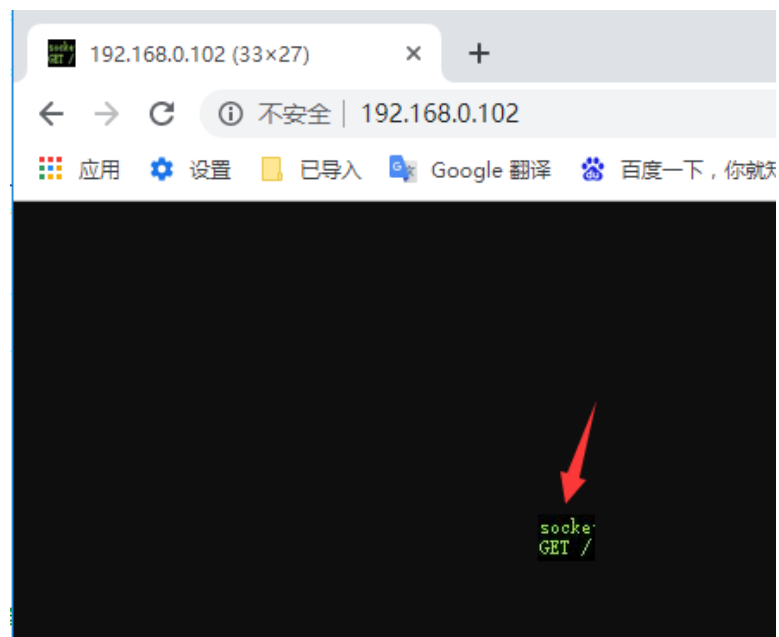
4.改一下程序,让程序字段设置为返回jpeg

```

182 if(sock_int_socket & SINT_STAT_RECV)
183 {
184     len = CH395GetRecvLength(sockindex);/* 获取当前缓冲区内数据长度 */
185     if(len == 0)return;
186     if(len > recv_buff_len)len = recv_buff_len;
187
188     CH395GetRecvData(sockindex,len,recv_buff);/* 读取数据 */
189
190     /*使用串口打印接收的数据*/
191     printf("socket%d receive data:\r\n",sockindex);
192     PutData(&rb_t_usart1_send,recv_buff,len);
193     USART_ITConfig(USART1, USART_IT_TXE, ENABLE);
194
195     /*响应头部*/
196     len = sprintf((char *)http_response_data,"HTTP/1.1 200 OK\r\n\
197 Content-Type: image/jpeg\r\n\
198 Content-Length:%d\r\n\r\n",sizeof(data_jpeg)
199 );
200     //发送响应头部
201     CH395SendData(sockindex, (UINT8*)http_response_data,len);
202     delay_ms(10);
203
204     /*发送响应正文*/
205     CH395SendData(sockindex, (unsigned char*)data_jpeg,sizeof(data_jpeg));
206 }
207
208 /* 连接中断, 仅在TCP模式下有效*/
209 if(sock_int_socket & SINT_STAT_CONNECT)
210 {
211     printf("socket%d SINT_STAT_CONNECT\r\n",sockindex);

```

5.然后在浏览器上重新访问下



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 - 608

0

0





« 上一篇: 13-网络芯片CH395Q学习开发-模块使用Socket0作为MAC RAW

posted on 2021-06-16 02:43 杨奉武 阅读(7) 评论(0) 编辑 收藏 举报


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：

- 他们翻遍用AI检测新冠的论文，一篇临床可用的也没有？！
 - 1个C文件就写出Web服务器！这个60岁程序员大有来头，网友：打算回去抄代码了
 - 天文学家搞医术，Science也挡不住
 - 拉勾招聘：618电商人才薪资15630元 同比增长4%
 - 京东云发布618十大技术应用趋势，含虚拟数字人等
- » 更多新闻...

历史上的今天：

2020-06-16 GPRS(Air202) Lua开发: 物联网MQTT ,APP扫描二维码绑定设备,实现远程通信控制



单片机,物联网,上位机,...

扫一扫二维码，入群聊。