黑金 AN8211 千兆以太网模块用户手册

Rev. 1.00

ALINX

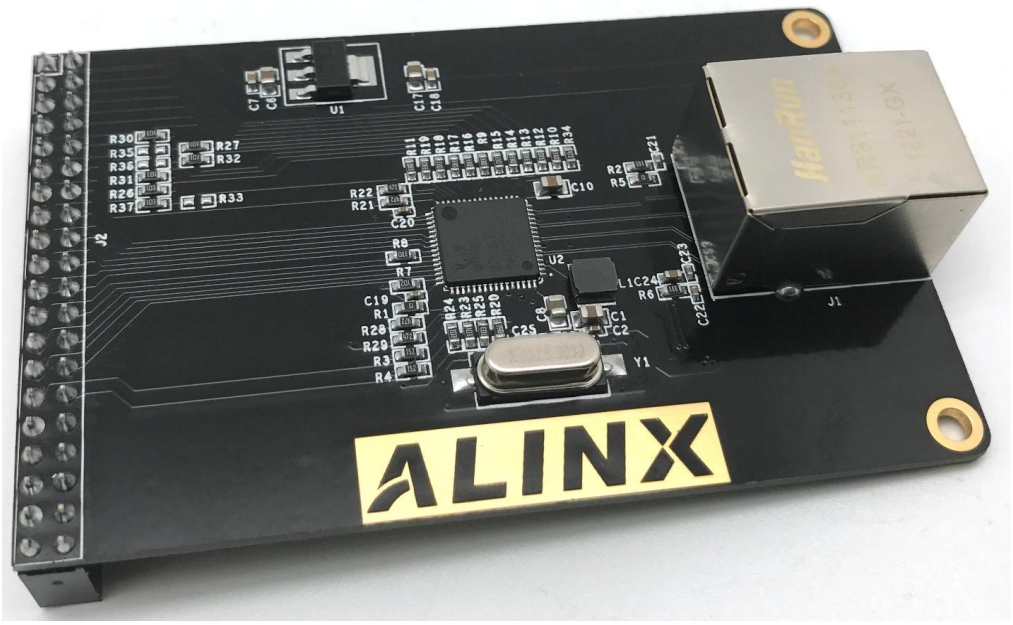
版本记录

版本	时间	作者	描述
Rev1.00	2015-9-5		First Release

第一部分 千兆以太网模块说明介绍

黑金千兆以太网模块 AN8211 采用 Realtek 公司的 RTL8211EG 以太网 GPHY 芯片，支持 10/100/1000 Mbps 网络传输速率。模块留有一个 40 针的排母用于连接 FPGA 开发板，一个千兆网口用于连接电脑的网卡或者其它网络设备（比如路由器）。

AN8211 模块实物照片如下：



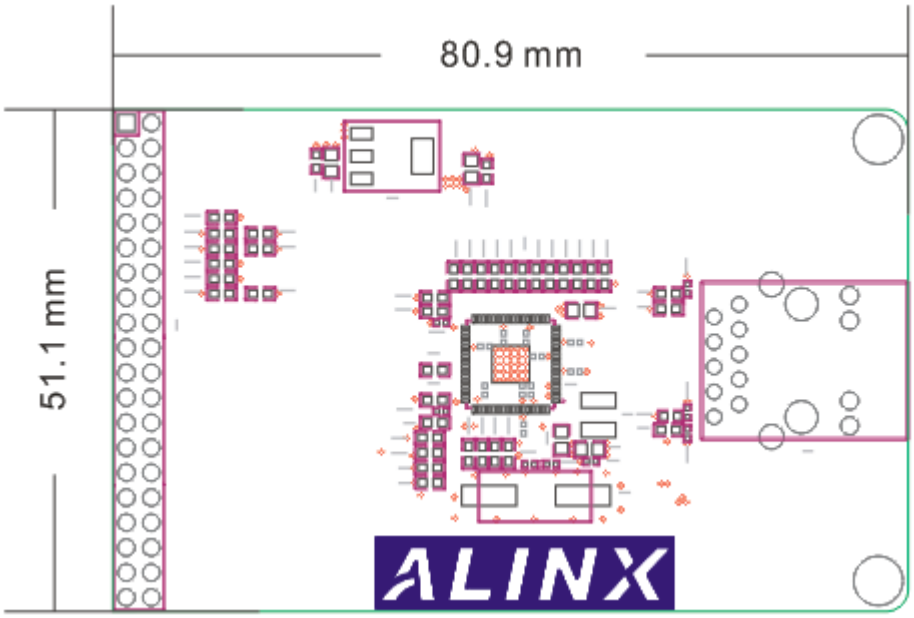
AN8211 模块正面图

AN8211 模块背面图

1.1 AN8211 模块的参数说明

以下为 AN8211 以太网模块的详细参数:

- AN8211 模块的尺寸：见下图;
- 10/100/1000 Mbps 自适应;
- 支持 GMII/RGMII/MII/RMII 通信接口;
- 网口支持MDI/MDX 自适应，Master/Slave 自适应;
- 支持 MDIO 总线进行 GPHY 的寄存器管理;
- 供电和功耗：单电源供电 5V，功耗为 0.5 瓦左右;



AN8211 以太网模块尺寸图

第二部分 模块硬件设计

RTL8211EG 上电会检测一些特定的 IO 的电平状态，从而确定自己的工作模式。在 AN8211 模块上，已经在 RTL8211EG 芯片的配置管脚上加了上拉电阻或者下拉电阻，让 GPHY 芯片能够上电就处于正常工作模式。表 1 描述了 AN8211 以太网模块 GPHY 芯片 RTL8211EG 上电之后的默认设定信息。

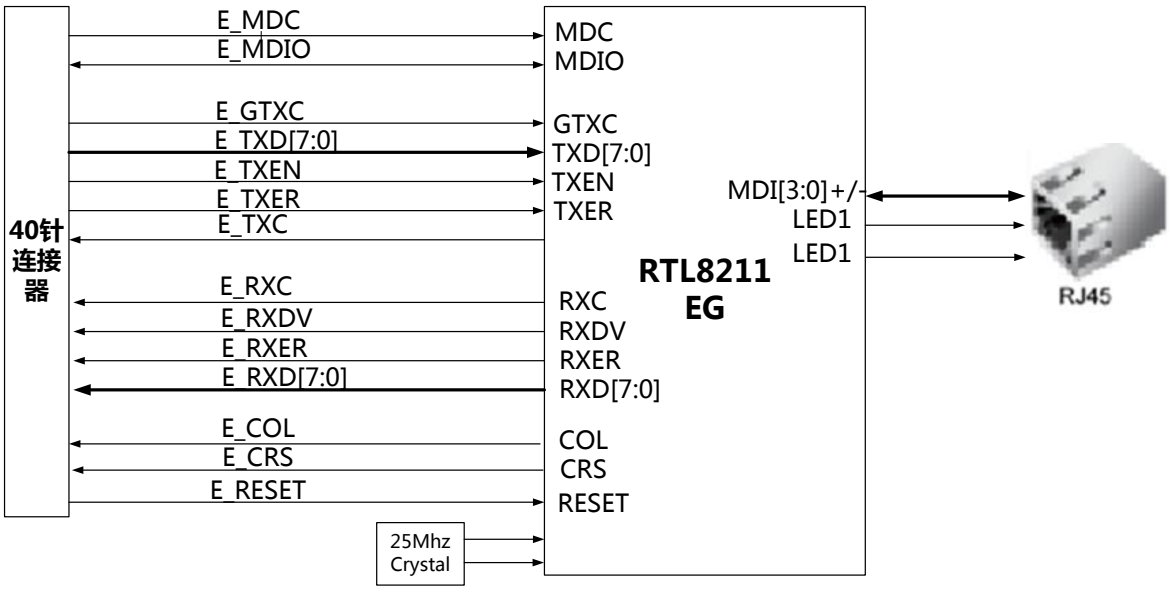
配置 Pin 脚	说明	配置值
PHYAD[2:0]	MDIO/MDC 模式的 PHY 地址	PHY Address 为 001
SELRGV	3.3V,2.5V,1.5/1.8V 电压选择	3.3V
AN[1:0]	自协商配置	(10/100/1000M)自适应
RX Delay	RX 时钟 2ns 延时	延时
TX Delay	TX 时钟 2ns 延时	延时
MODE	RGMII 或 GMII 选择	GMII

表 1 GPHY 芯片默认配置值

当网络连接到千兆以太网时，FPGA 开发板和 PHY 芯片 RTL8211EG 的数据传输时通过 GMII 总线通信，发送时钟 E_GTXC 由 FPGA 提供，频率为 125Mhz，数据为 TXD0~TXD7，数据有效信号为 TXEN，TXC 信号没有使用。接收时钟 E_RXC 由 PHY 芯片提供，数据为 RXD0~RXD7，数据有效信号为 RXDV，数据在时钟的上升沿采样。

在百兆的 MII 通信模式下，发送数据时,发送时钟为 25Mhz 的 TXC 信号，此 25Mhz 的 TXC 时钟是 PHY 输入给 FPGA 的，数据为 TXD0~TXD3，数据有效信号为 TXEN，GTXC 信号没有使用；接收数据时,接收时钟为 25Mhz 的 RXC 信号，数据为 RXD0~RXD3，数据有效信号为 RXDV。

下图为千兆以太网模块 AN8211EG 的硬件设计示意图:



以太网模块硬件设计示意图

模块 40 针排母的引脚分配：

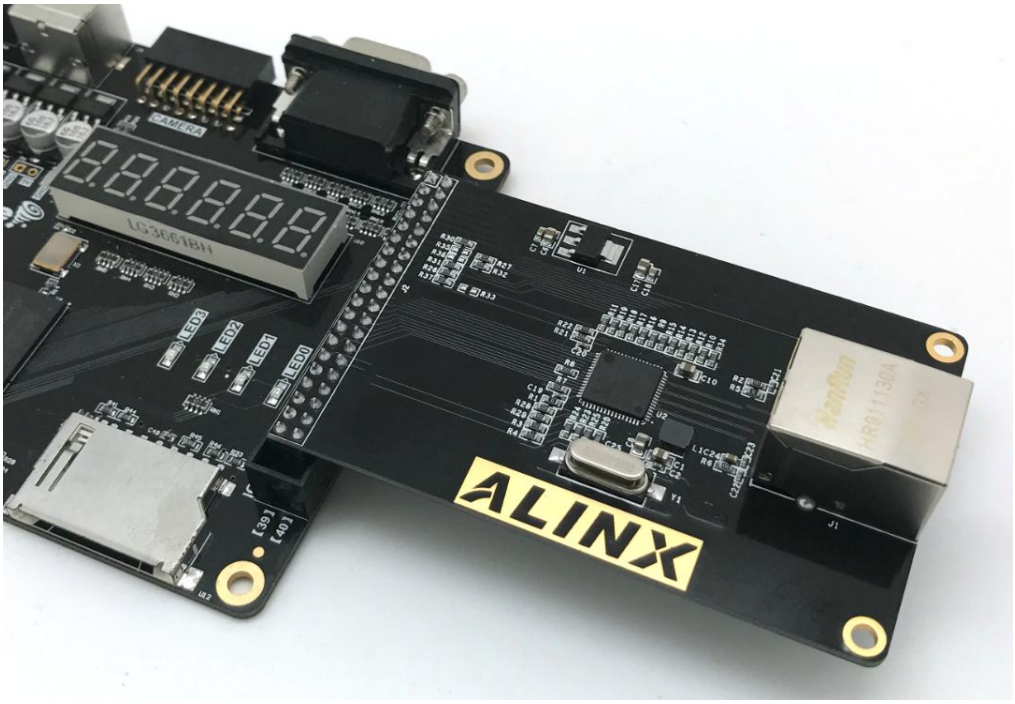
引脚号	引脚名称	备注
1	GND	地
2	+5V	5V 电源输入
3	E_RXDV	接收数据有效信号
4	E_RXD0	接收数据 Bit0
5	E_RXD1	接收数据 Bit1
6	E_RXD2	接收数据 Bit2
7	E_RXD3	接收数据 Bit3
8	E_RXC	GMII 接收时钟
9	E_RXD4	接收数据 Bit4
10	E_RXD5	接收数据 Bit5
11	E_RXD6	接收数据 Bit6
12	E_RXD7	接收数据 Bit7
13	E_RXER	接收数据错误
14	E_COL	Collision 信号
15	E_CRS	Carrier Sense 信号
16	E_GCLK	GMII 发送时钟
17	E_TXEN	发送使能信号
18	E_TXD0	发送数据 bit 0
19	E_RESET	复位信号
20	E_TXD1	发送数据 bit1
21	E_TXD2	发送数据 bit2
22	E_TXD3	发送数据 bit3
23	E_TXC	MII 发送时钟

24	**E_TXD4**	发送数据 bit4
25	**E_TXD5**	发送数据 bit5
26	**E_TXD6**	发送数据 bit6
27	**E_TXD7**	发送数据 bit7
28	**E_TXER**	发送错误信号
29	**E_MDC**	MDIO 管理时钟
30	**E_MDIO**	MDIO 管理数据
31	**-**	空脚
32	**-**	空脚
33	**-**	空脚
34	**-**	空脚
35	**-**	空脚
36	**-**	空脚
37	**-**	空脚
38	**-**	空脚
39	**-**	空脚
40	**-**	空脚

关于详细的管脚说明和GMII/MII的通信时序，请大家参考RTL8211的芯片手册

第三部分 硬件连接

千兆以太网和 FPGA 开发板的硬件连接很简单，只要把以太网模块的 40 针的母座 J2 插到 FPGA 开发板的扩展口上，连接器的管脚 1 对齐就好了 。以下为黑金 AX301 开发板和以太网模块的硬件连接图：

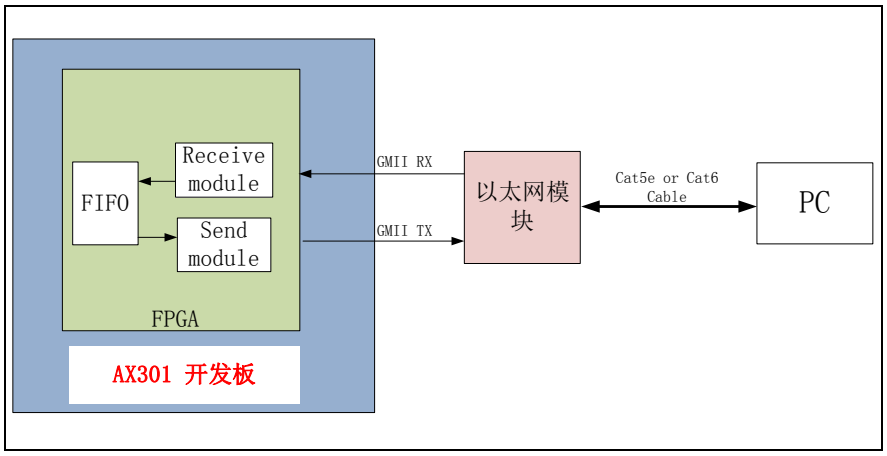


开发板上电，以太网模块的网口和电脑的网口用千兆网线(5 类加或者 6 类线）连接。接下去我们就可以开始做以太网的实验了。

第四部分 千兆以太网程序

我们提供了 2 个以太网测试程序，一个是千兆以太网测试程序，用于以太网模块和电脑之间实现千兆数据通信。另一个是百兆以太网测试程序，用于以太网模块和电脑之间实现百兆数据通信。下面为大家介绍一下千兆以太网测试程序:

测试程序的通信协议采用 Ethernet UDP 通信协议。 FPGA 通过 GMII 总线和以太网模块的 Gigabit PHY 芯片 RTL8211EG 通信, Gigabit PHY 芯片通过网线和 PC 进行数据交换。以太网数据通信的示意图如下:

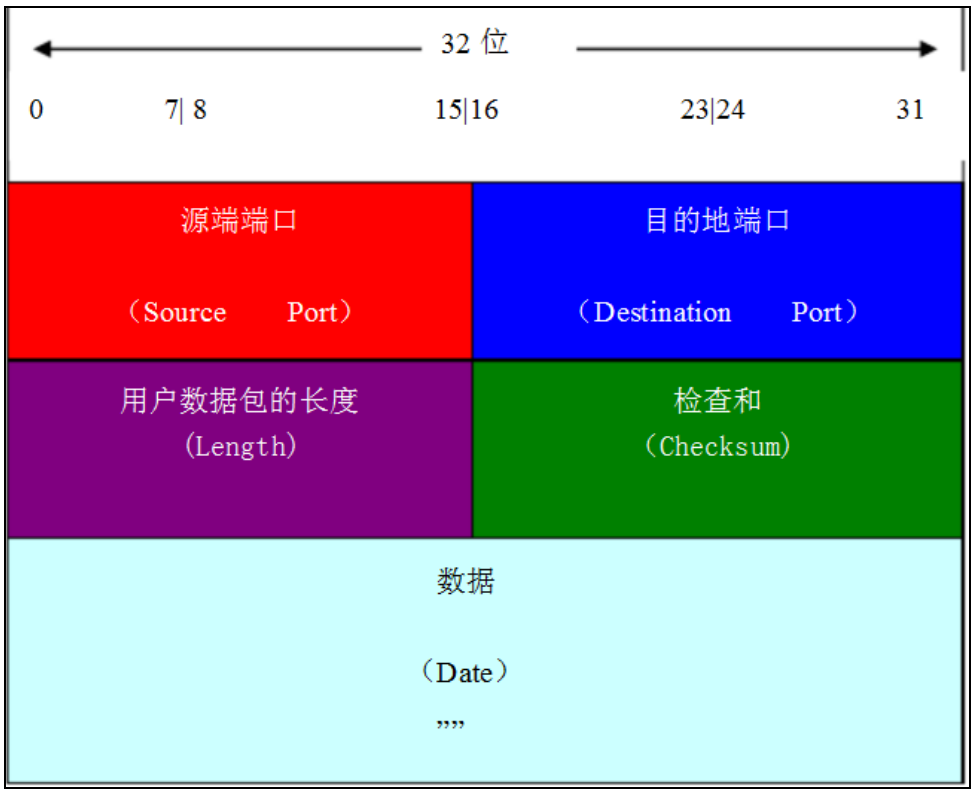


4.1 UDP协议

UDP 是 User Datagram Protocol（用户数据报协议）的英文缩写。UDP 只提供一种基本的、低延迟的被称为数据报的通讯。所谓数据报，就是一种自带寻址信息，从发送端走到接收端的数据包。UDP 协议经常用于图像传输、网络监控数据交换等数据传输速度要求比较高的场合。

1. UDP 协议的报头格式,

UDP 报头由 4 个域组成，其中每个域各占用 2 个字节，具体如下：



① UDP 源端口号

② 目标端口号

③ 数据报长度

④ 校验值

UDP 协议使用端口号为不同的应用保留其各自的数据传输通道。数据发送一方将 UDP 数据报通过源端口发送出去，而数据接收一方则通过目标端口接收数据。

数据报的长度是指包括报头和数据部分在内的总字节数。因为报头的长度是固定的，所以该域主要被用来计算可变长度的数据部分（又称为数据负载）。数据报的最大长度根据操作环境的不同而各异。从理论上说，包含报头在内的数据报的最大长度为 65535 字节。不过，一些实际应用往往会限制数据报的大小，有时会降低到 8192 字节。

UDP 协议使用报头中的校验值来保证数据的安全。校验值首先在数据发送方通过特殊的算法计算得出，在传递到接收方之后，还需要再重新计算。如果某个数据报在传输过程中被第三方篡改或者由于线路噪音等原因受到损坏，发送和接收方的校验计算值将不会相符，由此 UDP 协议可以检测是否出错。虽然 UDP 提供有错误检测，但检测到错误时，错误校正，只是简单地把损坏的消息段扔掉，或者给应用程序提供警告信息。

2. IP 数据报首部

因为 UDP 协议包只是 IP 包中的一种，所以我们也顺便来介绍一下 IP 包的数据格式。下图为 IP 分组的报文头格式，报文头的前 20 个字节是固定的，后面的可变

版本:

占 4 位,指 IP 协议的版本目前的 IP 协议版本号为 4 (即 IPv4)

首部长度:

占 4 位,可表示的最大数值是 15 个单位(一个单位为 4 字节)因此 IP 的首部长度的最大值是 60 字节

区分服务:

占 8 位,用来获得更好的服务,在旧标准中叫做服务类型,但实际上一直未被使用过.1998 年这个字段改名为区分服务.只有在使用区分服务(DiffServ)时,这个字段才起作用.一般的情况下都不使用这个字段

总长度:

占 16 位,指首部和数据之和的长度,单位为字节,因此数据报的最大长度为 65535 字节.总长度必须不超过最大传送单元 MTU

标识:

占 16 位,它是一个计数器,用来产生数据报的标识

标志(flag):

占 3 位,目前只有前两位有意义

MF

标志字段的最低位是 MF (More Fragment)

MF=1 表示后面“还有分片”。MF=0 表示最后一个分片

DF

标志字段中间的一位是 DF (Don't Fragment)

只有当 DF=0 时才允许分片

片偏移:

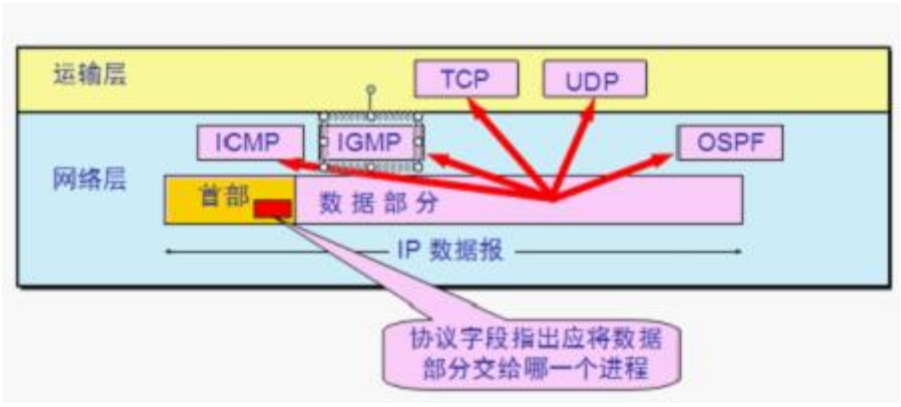
占 12 位,指较长的分组在分片后某片在原分组中的相对位置.片偏移以 8 个字节为偏移单位

生存时间:

占 8 位,记为 TTL (Time To Live) 数据报在网络中可通过的路由器数的最大值,TTL 字段是由发送端初始设置一个 8 bit 字段.推荐的初始值由分配数字 RFC 指定,当前值为 64.发送 ICMP 回显应答时经常把 TTL 设为最大值 255

协议:

占 8 位,指出此数据报携带的数据使用何种协议以便目的主机的 IP 层将数据部分上交给哪个处理过程, 1 表示为 ICMP 协议, 2 表示为 IGMP 协议, 6 表示为 TCP 协议, 17 表示为 UDP 协议



首部检验和:

占 16 位,只检验数据报的首部不检验数据部分.这里不采用 CRC 检验码而采用简单的计算方法

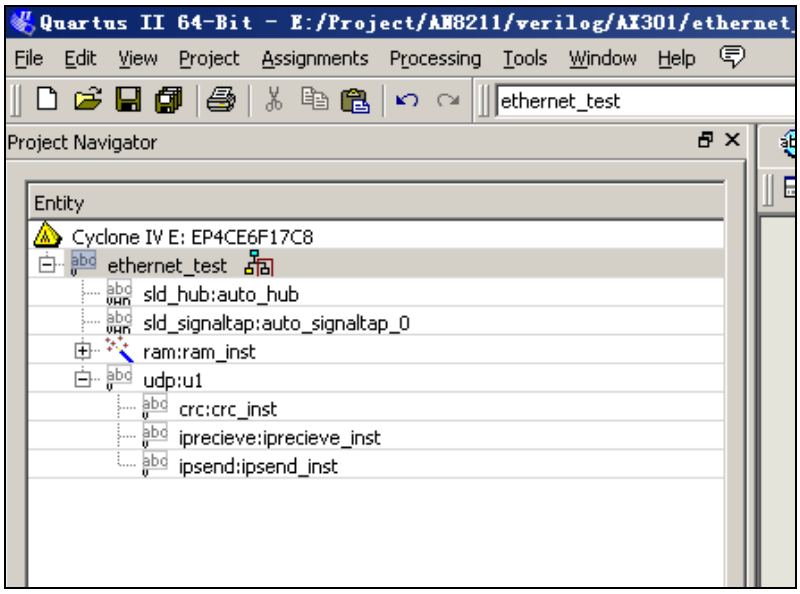
源地址和目的地址:

都各占 4 字节,分别记录源地址和目的地址

4.2 程序设计

千兆以太网 GMII 通信是基于 UDP 通信协议的 verilog 代码设计 ,程序功能是开发板上电后默认通过网口不断的发送 UDP 数据包给 PC 或网络设备 , 数据包的内容为“Hello Alinx AX301”。当 FPGA 检测网口发来的 UDP 的数据包，会把接收到的数据包存储在 RAM 中，再不断的把接收到的数据包通过网口发回到电脑或者其他网络设备。

整个 ethernet_test 项目主要由一个顶层模块 ethernet_test.v , UDP 测试程序 udp.v 和三个子模块：UDP 发送模块(ipsend.v),UDP 接收模块(iprecieve.v)和 CRC 校验模块(crc.v)组成。



1. UDP 发送模块(ipsend.v)说明

UDP 发送模块实现把 RAM 里的数据组成 UDP IP 包格式通过 GMII 总线发给 PHY 芯片,PHY 芯片再把数据发送到开发板的网口。

IP 数据包发送之前需要先发送 IP 数据包的包头，IP 包头由 8 个字节的前导码，目标 MAC Address，源 MAC 地址和两个字节的 IP 包类型组成。前导码是由 7 个 0x55, 1 个 0xD5 字节组成，表示一个 IP 数据包传输的开始; 目标 MAC Address 为数据要发送对象的 MAC 地址 ,如果开发板的网口是和您的 PC 机相连，那目标 MAC Address 的值为你 PC 机的 MAC 地址。源 MAC Address 是指开发板本地的 MAC 地址。IP 包类型值为 0x0800。

发完 IP 包头之后开始发送 IP 数据报首部 , IP 数据报首部的格式我们在前面

已经讲过，接着发送 RAM 中的数据，最后发送 4 个字节的 CRC32 的值。

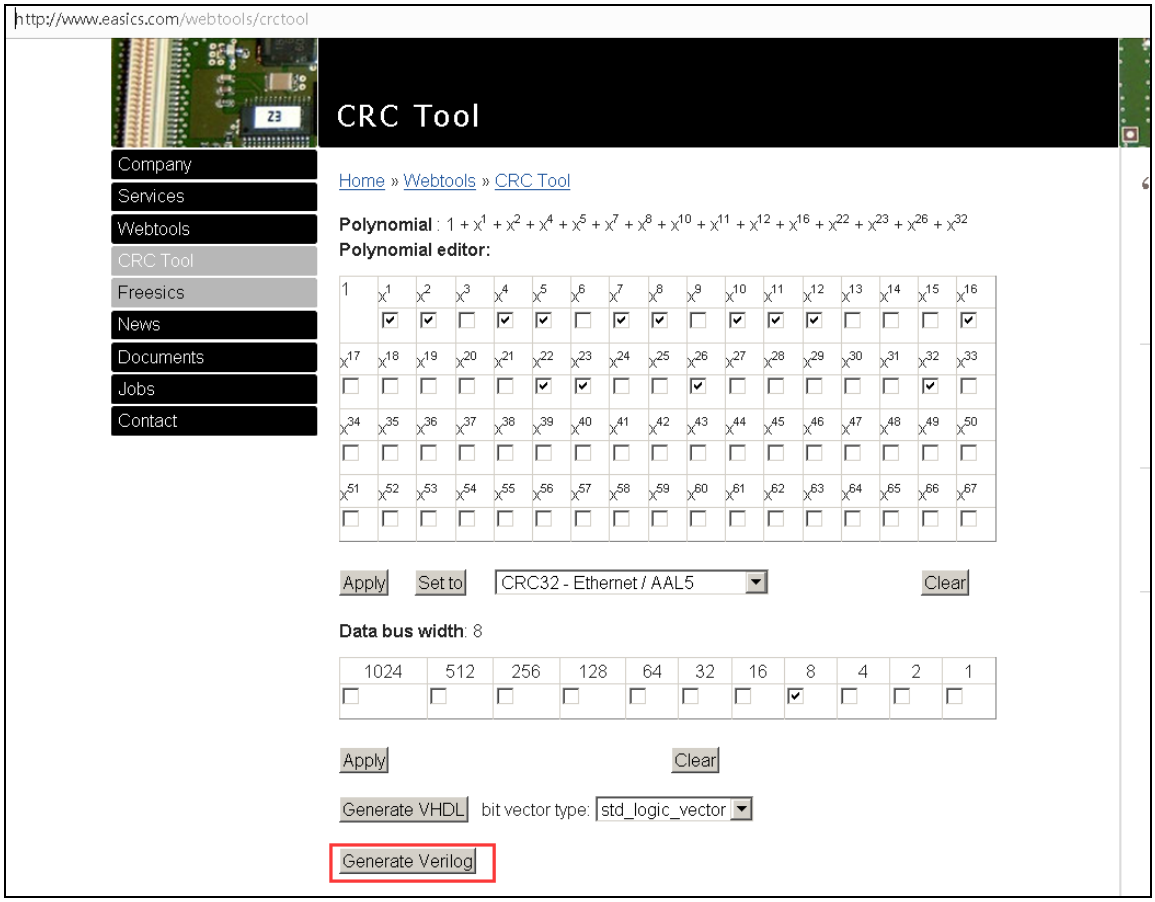
2. UDP 接收模块(iprecieve.v) 说明

UDP 数据接收程序和 UDP 数据发送是一样的，先判断是否接收到前导码，如果接收到前导码，开始接收目标 MAC Address 并判断是否和本开发板的 MAC 地址一致，一致的话再接收剩余的包头和数据。接收数据的时候把接收到的 8 bit 数据转化成 32bit 的数据宽度写入到 RAM 中。

3 . CRC 校验模块(crc.v) 说明

一个 IP 数据包的 CRC 校验是在目标 MAC Address 开始计算的，一直计算到一个包的最后一个数据为止。以太网的 CRC32 的 verilog 的算法和多项式可以在以下网站中直接生成：

http://www.easics.com/webtools/crctool



4 . udp 测试模块(udp.v) 说明

Udp 测试程序只是实例化前面编写的 3 个 UDP 发送，UDP 接收, CRC32 校

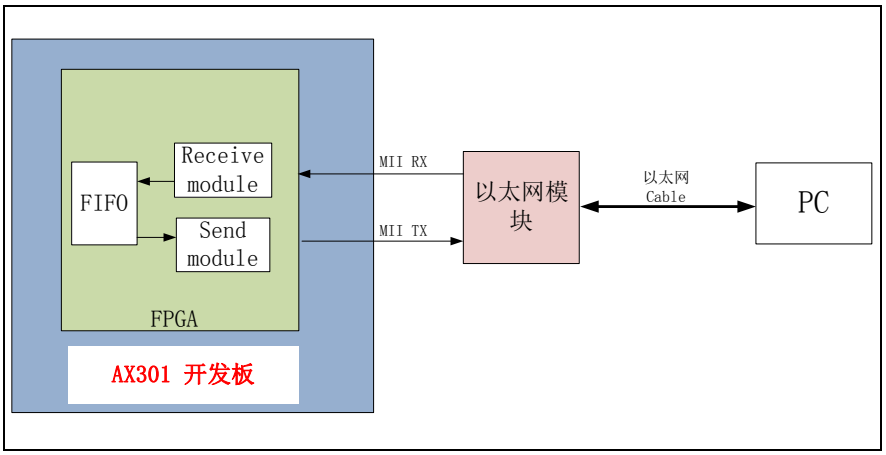
验的子模块。

5 . TOP 模块(Ethernet_test.v) 说明

TOP 程序一个功能是实例化 udp 模块；另外也调用了一个双口 RAM，读写的数据宽度为 32bit, RAM 的地址深度为 512。双口 RAM 用于存储以太网接收到的数据，程序在没有接收到以太网数据包的时候，一直往外发送存储在 RAM 中的”HELLO ALINX AX301”的字符，如果接收到以太网数据包，接收到的以太网数据包会覆盖 RAM 中的数据，程序会不断向网口发送接收到的数据包。

第五部分 百兆以太网程序

百兆以太网程序和千兆以太网程序的设计原理是一样的，主要的区别就是数据总线由 GMII 变成了 MII，时钟的频率由 125Mhz 变成了 25Mhz。发送时钟和接收时钟都有 GPHY 提供给 FPGA。以下为百兆太网数据通信的示意图:

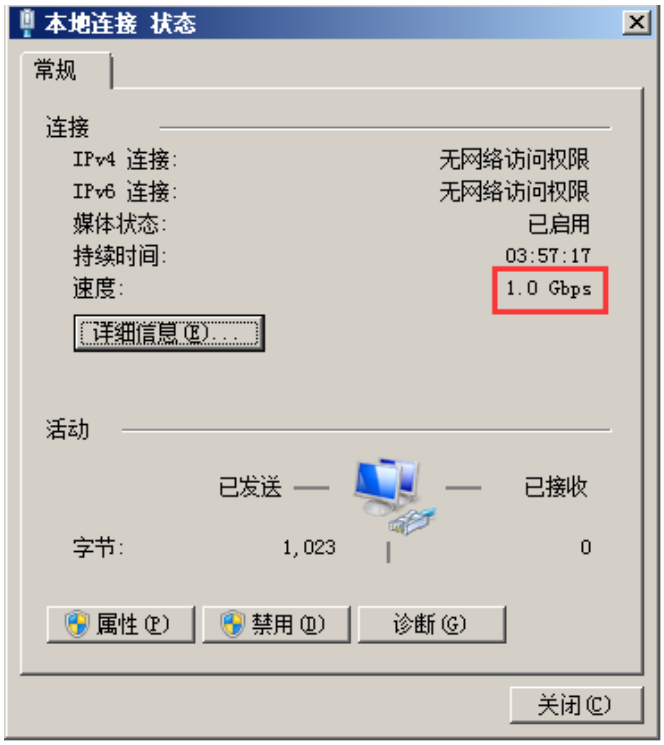


因为 MII 总线的数据宽度是 4 位，所以数据发送和数据接收的时候，一个字节需要有 2 个时钟来发送或接收，数据低四位在先，高四位在后。百兆以太网程序的每个模块的功能跟千兆以太网测试程序一样，这里就不在介绍了。

第六部分 以太网实验

1. 准备工作

第一步：首先用五类+或者六类网线连接开发板的网口和 PC 的网口，然后开发板上电，这时以太网模块的网口的灯会亮起，说明网络已经 Link 上。这时可以确认一下电脑和以太网直接的 Link 是千兆还是百兆的，点击本地连接查看状态。



注意 如果发现网口没有 Link 上，这时可能是 FPGA 的默认程序把 GPHY 的 Reset 管脚拉低了，使得 GPHY 处于 Reset 状态，只要 FPGA 下载以太网测试程序，网口就能 Link 上。

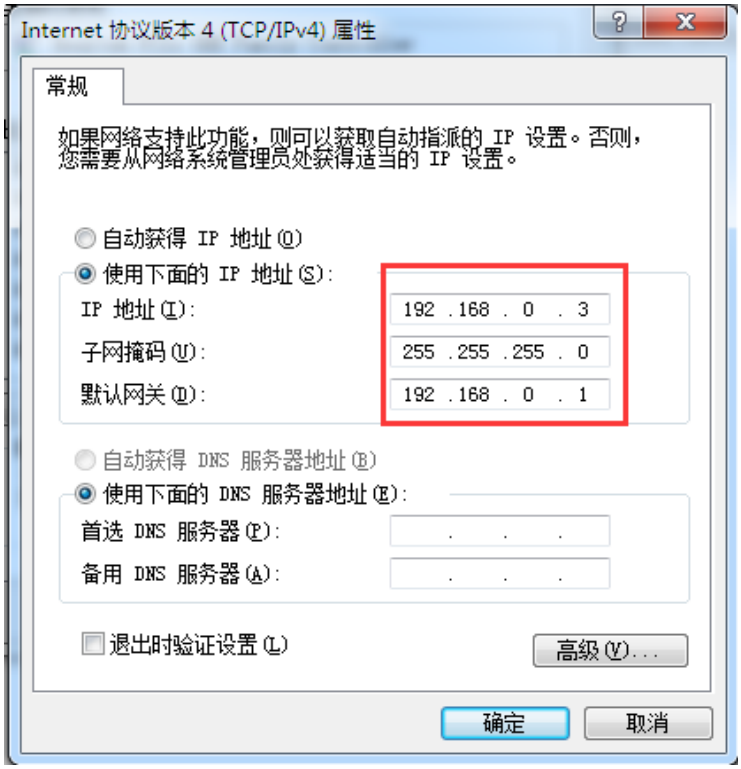
第二步 如果电脑的系统是 Window XP 的话，需要修改 UDP 发送模块(ipsend.v)中的目标 mac address 为 PC 的网卡 mac address,修改后重新编译一边。如果不知道自己 PC 网卡的 mac address，就在 DOS 命令窗口用 ipconfig –all 命令看一下。

```
46      preamble[4]<=8'h55;
47      preamble[5]<=8'h55;
48      preamble[6]<=8'h55;
49      preamble[7]<=8'hD5;
50      mac_addr[0]<=8'h28;                    //目的MAC地址 28-D2-44-DC-6E-FE
51      mac_addr[1]<=8'hD2;
52      mac_addr[2]<=8'h44;
53      mac_addr[3]<=8'hDC;
54      mac_addr[4]<=8'h6E;
55      mac_addr[5]<=8'hFE;
56      mac_addr[6]<=8'h00;                    //源MAC地址 00-0A-35-01-FE-C0
57      mac_addr[7]<=8'h0A;
```

XP操作系统修改成您自己PC的MAC Address

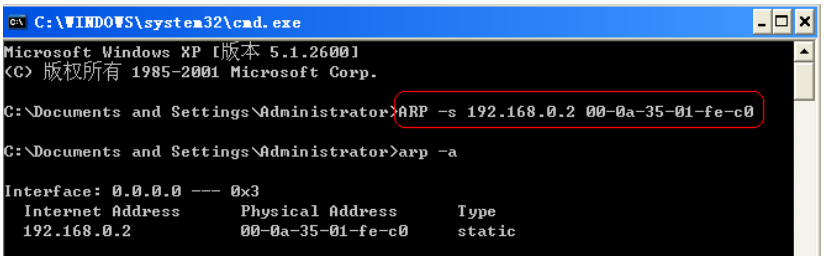
如果电脑是其它操作系统的话，这里可以不需要修改，发送目标地址全为 FF 广播包就可以。

第三步：修改 PC 的 IP 地址为 192.168.0.3。PC 的 IP Address 需要和发送模块(ipsend.v)中设置一致，不然网络调试助手会接收不到开发板发送的 UDP 数据包。



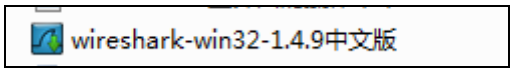
第四步：用管理员权限打开 DOS 命令窗口，绑定开发板的 IP 地址和 MAC 地址，这样网络调试助手发送 IP 为 192.168.0.2 的网络数据包的时候，目标 MAC 地址自动为开发板的 MAC 地址。

运行命令：ARP -s 192.168.0.2 00-0a-35-01-fe-c0



绑定后我们可以用 arp –a 命令来查看电脑上帮定的结果。

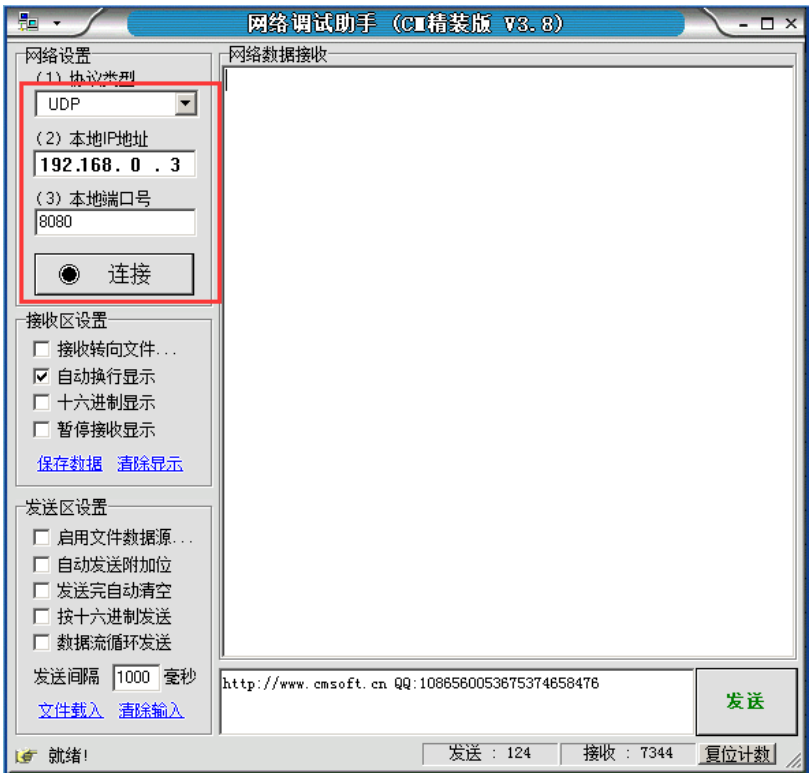
第五步：这一步为可选项，安装 Wireshark 是为了方便用户网络通信的调试，安装光盘的 TOOL 目录下的网络抓包工具 Wireshark，我们在实验的时候可以用这工具来查看 PC 网口发送的数据和接收到的数据的详细信息。



2. 以太网通信测试

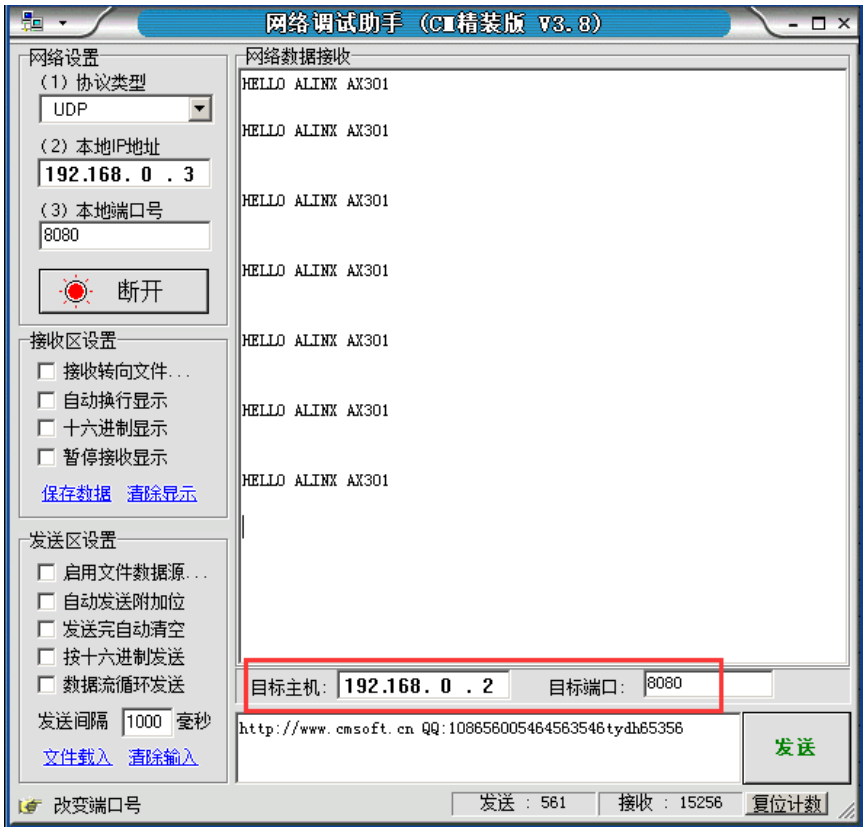
第一步：根据网口 Link 的速度来判断下载千兆测试程序还是百兆测试程序。

第二步：打开我们提供的网络调试助手并设置参数如下，再按连接按钮(这里的本地的 IP 地址为电脑的 IP Address，本地端口需要跟 FPGA 程序中的一致，为 8080)。

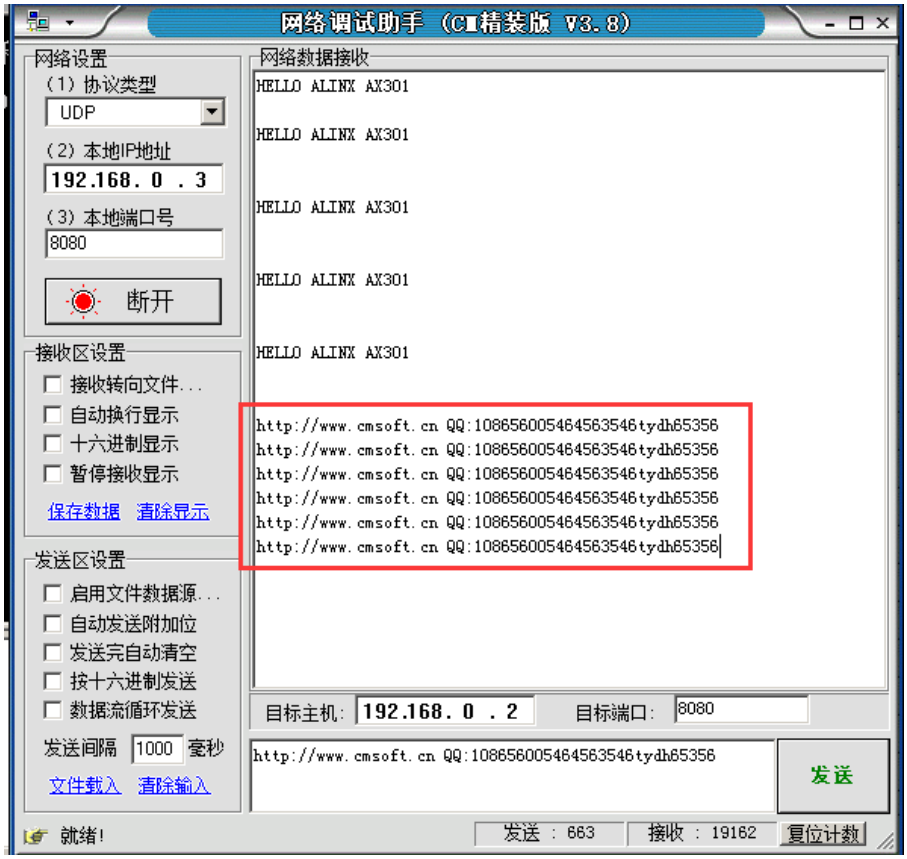


这时网络数据接收窗口会显示 FPGA 发给电脑的以太网数据包"Hello ALINX AX301"，我们再设置目标主机的 IP 地址需要和 FPGA 程序中的 IP 地址一致，目

标端口号也需要和 FPGA 程序的一致(8080)。

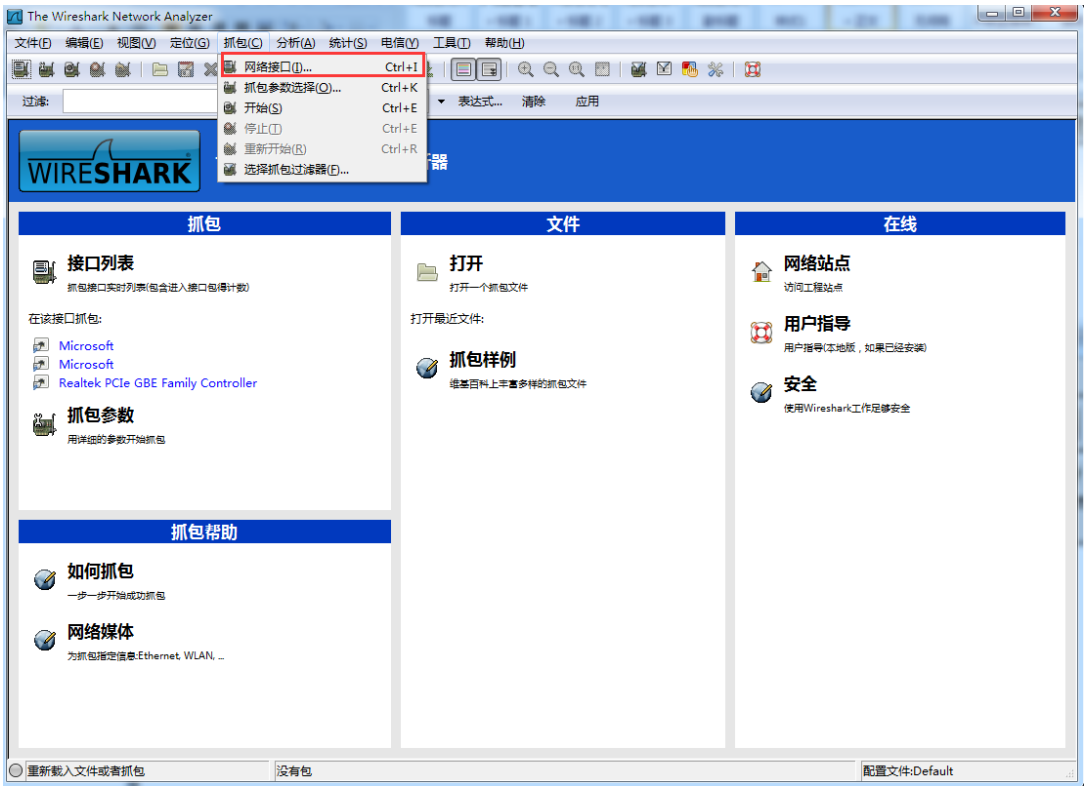


第三步：再在网络调试助手的发送窗口发送一大串字符,在网络的数据接收窗口我们可以看到从 FPGA 返回的数据也变成刚发送字符串。

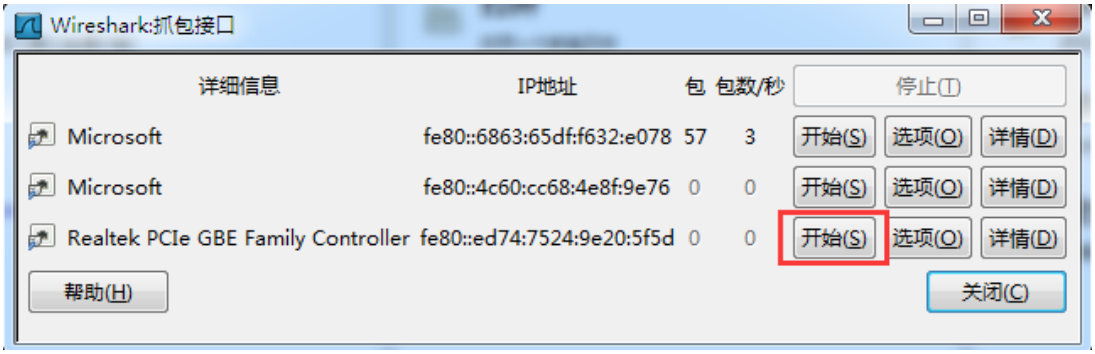


注意：以太网的数据帧的传输有包长的要求，一般在 46～1500 字节。所以在发送以太网数据包的时候,数据帧的长度不能太短，不然会导致电脑数据包发送而 FPGA 收不到数据包的情况。

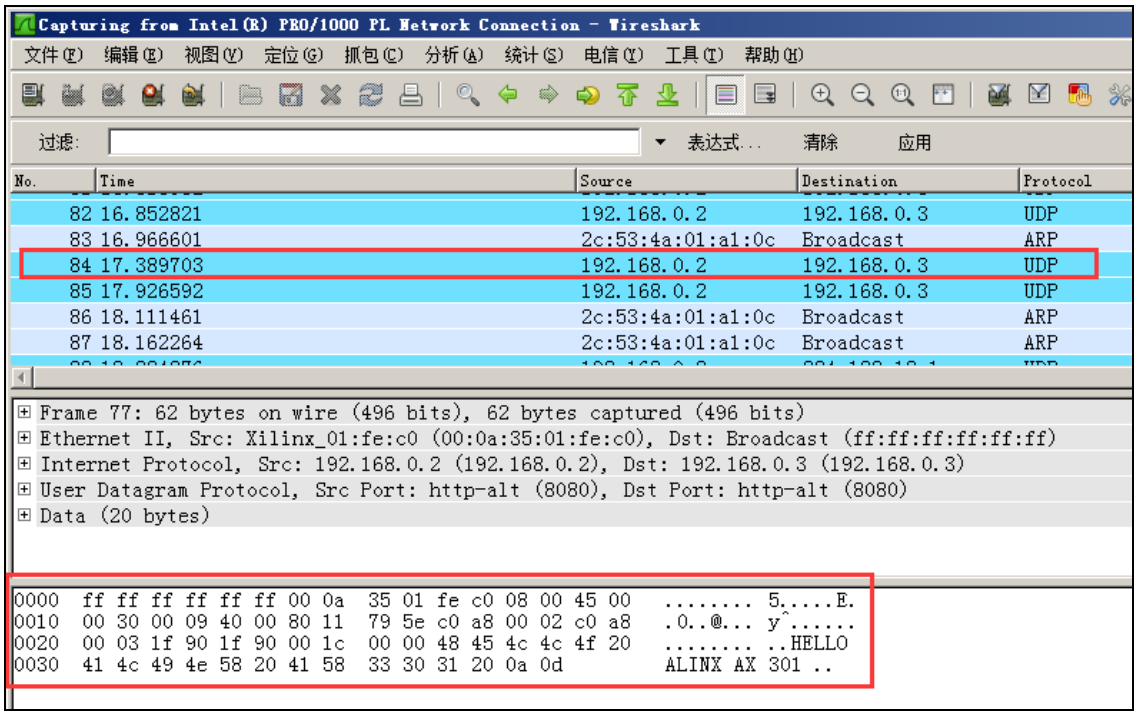
第四步：这一步对用户来讲是可选的，用户如果想查看更多数据包传输的信息，可以使用网络抓包工具 Wireshark 来查看电脑的网卡接收和发送的网络数据，打开安装好的 wireshark 抓包工具，点击菜单抓包->网络接口。



在弹出的抓包接口窗口选择您电脑的千兆网卡，按开始按钮开始抓包。



在 wireshark 抓包窗口我们可以看到开发板(192.168.0.2)向电脑网口(192.168.0.2)发来的数据包，这里会显示数据包的目标 MAC，源 MAC，IP 包头和 UDP 包等信息。



经过实际测试，UDP 通信的数据速度可以达到 900Mbps 以上，非常适合高速数据传输的场合，比如视频图像传输，高速数据采集的工作场合。