Gigabit Ethernet Module
AN8211
User Manual

ALINX

Version Record

Version	Date	Release By	Description
Rev 1.0	2015-9-5	Rachel Zhou	First Release

Part 1: Gigabit Ethernet Module General Description

ALINX Gigabit Ethernet Module AN8211, use RTL8211EG GPHY chip of Realtek, supports 10/100/1000 Mbps Network transmission rate。The module reserves a 40-pin female header that to connect FPGA development kit, a Gigabit Ethernet port is used to connect a computer's network card or other network device (such as a router).

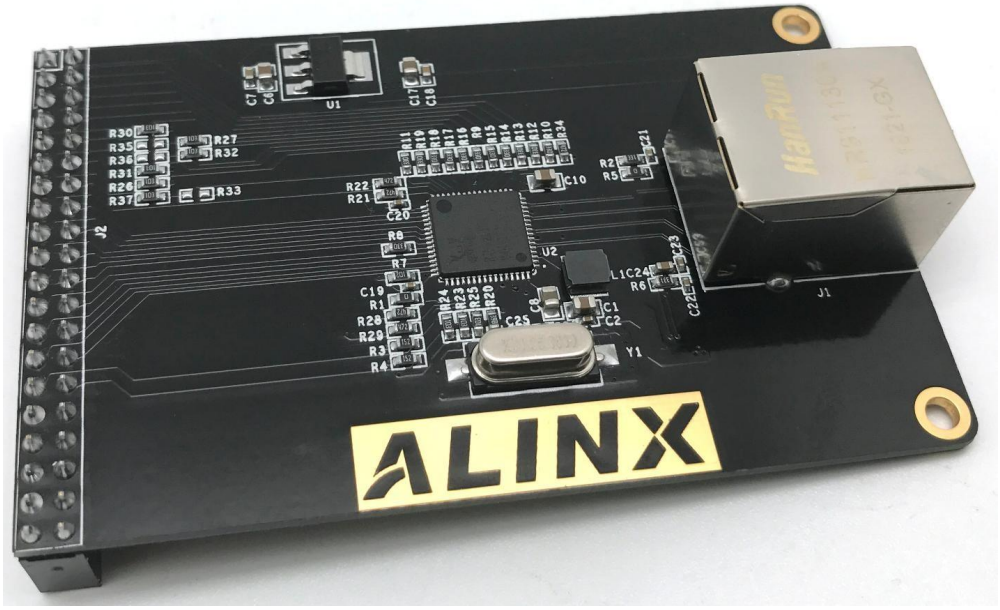


Figure 1-1: AN8211 module product photo (Front side)

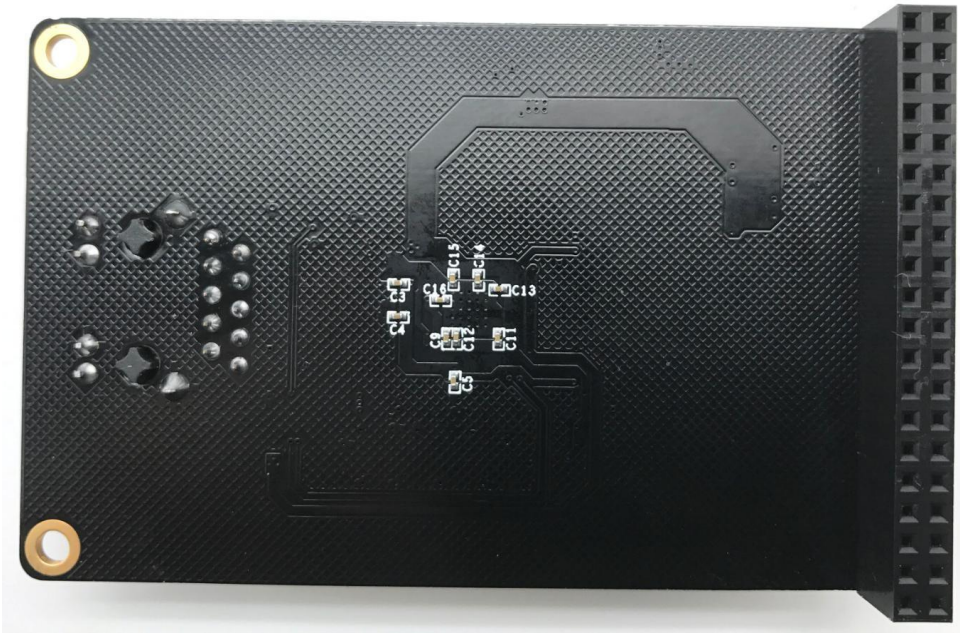


Figure 1-2: AN8211 module product photo (Back side)

1.1 AN8211 Gigabit Ethernet Module Detail Parameter

Gigabit Ethernet module detail parameter listed in below:

- AN8211 Module Dimension: Detailed as Figure 1-3
- 10/10 10/100/1000 Mbps adaptive
- Support GMII/RGMII/MII/RMII communication interface
- Network port supports MDI/MDX adaptive, Master/Slave adaptive
- Support MDIO bus for GPHY register management
- Power supply and power consumption: single power supply 5V, power consumption is about 0.5 watts;

1.2 AN8211 Gigabit Ethernet Module Parameter Description

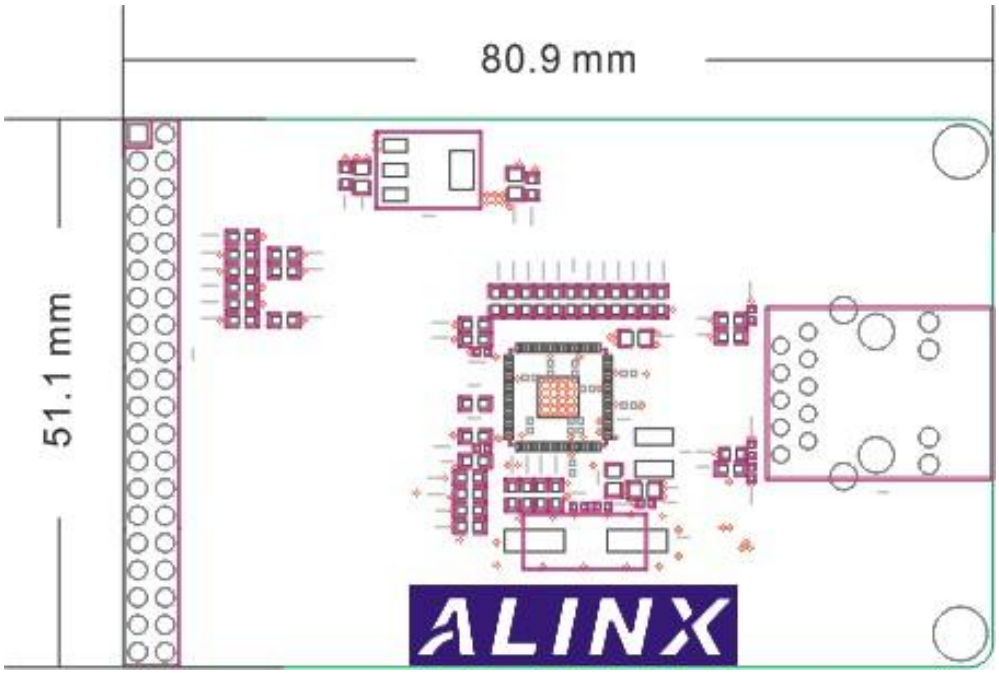


Figure 1-3: AN8211 Gigabit Ethernet Module Dimensions

Part 2: Gigabit Ethernet Module Hardware Design

After the RTL8211EG is powered up, it detects the level status of some specific IOs to determine its working mode. On the AN8211 module, a pull-up resistor or pull-down resistor has been added to the configuration pin of the RTL8211EG chip to allow the GPHY chip to be in normal

operation mode after power-on. Table 2-1 describes the default settings of the AN8211 Ethernet module GPHY chip RTL8211EG after power-on.

Configuration Pin	Descripton	Default Vaule
PHYAD[2:0]	PHY address of MDIO/MDC mode	PHY Address is 001
SELRGV	3.3V,2.5V,1.5/1.8V voltage selection	3.3V
AN[1:0]	Auto-negotiation configuration	(10/100/1000M) Adaptive
RX Delay	RX clock 2ns delay	Delay
TX Delay	TX clock 2ns delay	Delay
MODE	RGMII or GMII Selction	GMII

Table 2-1: Default setting of PHY Chip RTL8211EG after Power-on

Transmitting through the GMII bus, the transmission clock E_GTXC is provided by the FPGA, the frequency is 125Mhz, the data is TXD0~TXD7, the data valid signal is TXEN, the TXC signal is not used. The receive clock E_RXC is provided by the PHY chip, the data is RXD0~RXD7, the data valid signal is RXDV, and the data is sampled on the rising edge of the clock.

In the 100M MII communication mode, when transmitting data, then transmitting clock is a 25Mhz TXT signal, the 25Mhz TXC clock is the PHY input to the FPGA, the data is TXD0~TXD3, the data valid signal is TXEN, and the GTXC signal is not used; When receiving data, the receiving clock is a 25Mhz RXC signal, the data is RXD0~RXD3, and the data valid signal is RXDV.

The following figure 2-1 shows the hardware design of the Gigabit Ethernet module AN8211:

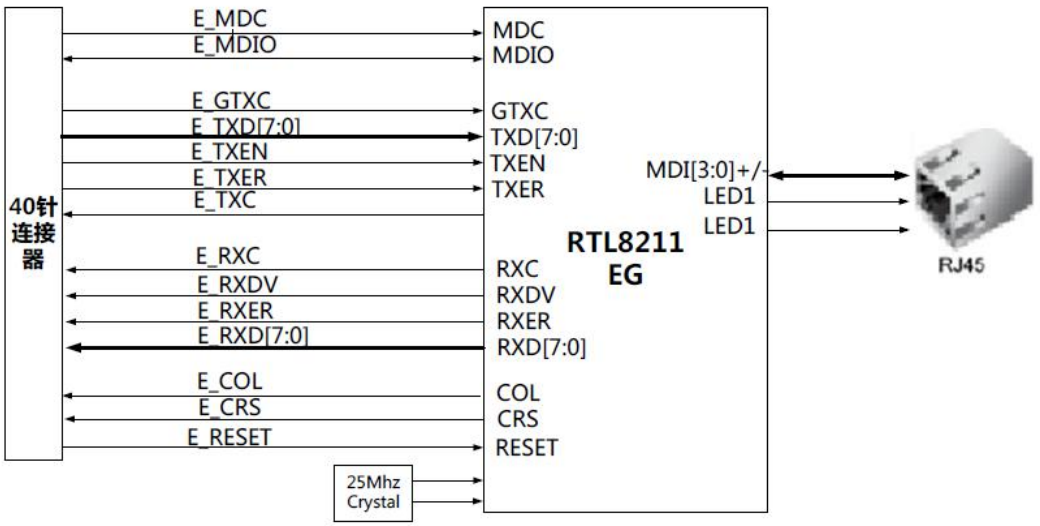


Figure 2-1: Hardware Design of the Gigabit Ethernet Module AN8211

40-pin assignment of Gigabit Ethernet Module AN8211

Pin	Pin Name	Description
1	GND	Ground
2	+5V	5V Power Supply
3	E_RXDV	Receive data valid signal
4	E_RXD0	Bit0 of receive data
5	E_RXD1	Bit1 of receive data
6	E_RXD2	Bit2 of receive data
7	E_RXD3	Bit3 of receive data
8	E_RXC	GMII Receive Clock
9	E_RXD4	Bit4 of receive data
10	E_RXD5	Bit5 of receive data
11	E_RXD6	Bit6 of receive data
12	E_RXD7	Bit7 of receive data
13	E_RXER	Receive data error
14	E_COL	Collision signal
15	E_CRS	Carrier Sense signal
16	E_GCLK	GMII transmit clock
17	E_TXEN	Transmit enalbe signal
18	E_TXD0	Bit0 of transmit data
19	E_RESET	Reset signal
20	E_TXD1	Bit1 of transmit data
21	E_TXD2	Bit2 of transmit data

22	E_TXD3	Bit3 of transmit data
23	E_TXC	MII tramsimit clock
24	E_TXD4	bit4 of transmit data
25	E_TXD5	Bit5 of transmit data
26	E_TXD6	Bit6 of transmit data
27	E_TXD7	Bit7 of transmit data
28	E_TXER	Transmitting Erro Signal
29	E_MDC	MDIO management clock
30	E_MDIO	MDIO management data
31	-	NC
32	-	NC
33	-	NC
34	-	NC
35	-	NC
36	-	NC
37	-	NC
38	-	NC
39	-	NC
40	-	NC

For detailed pin description and GMII/MII communication timing, please refer to the RTL8211 chip datasheet.

Part 3: Hardware Connection

The hardware connection between the AN8211 module and FPGA development are easy. The 40-pin female headers of module plug into the expansion board of FPGA development board. Make sure to align the Pin1 of the module and FPGA board. The figure 4-1 is hardware connection of ALINX Series FPGA development kit AX301 and the AN8211 module.

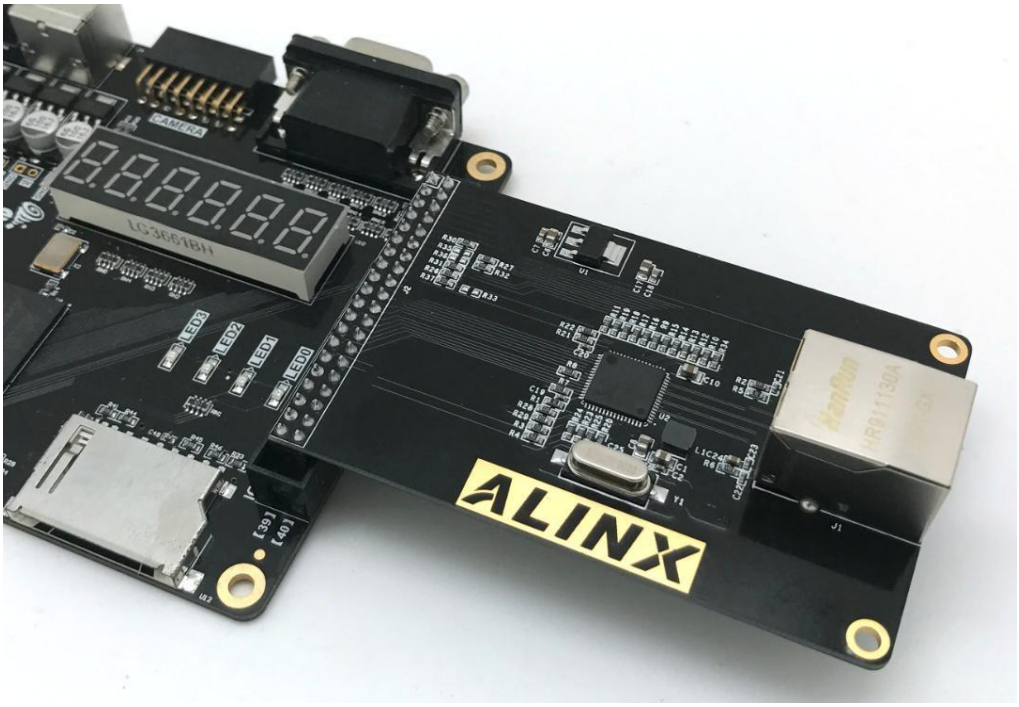


Figure 3-1: Hardware connection to AX301 FPGA Board and AN8211

After the board is powered on, use the Gigabit Ethernet cable (Category 5 or Category 6 cable) to connect the network port of the Ethernet module to the network port of the computer. After the connection is completed, you can start the experiment of Ethernet.

Part 4: Gigabit Ethernet program

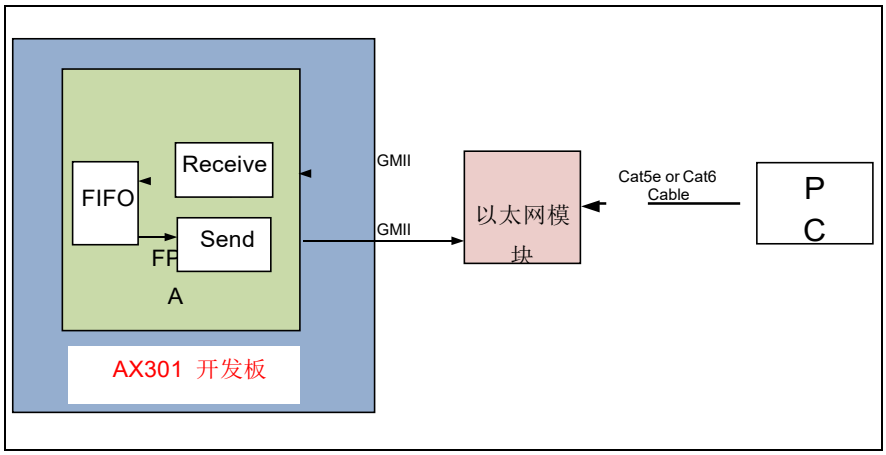
We provide 2 Ethernet test program, one is a Gigabit Ethernet testing program for Gigabit data communication between the Ethernet module and the computer. The other is the 100M Ethernet test program for 100M data communication between the Ethernet module and the computer. Let me introduce you to the Gigabit Ethernet test program

The communication protocol of the test program uses the Ethernet UDP communication protocol. The FPGA communicates with the Gigabit PHY chip RTL8211EG of the Ethernet module through the GMII bus, and the Gigabit PHY chip exchanges data through the network cable and the PC.

Figure 4-1 detailed ethernet data communication diagram.

Figure 4-1 Ethernet Data Communication Diagram

4.1 AN8211 Gigabit UDP Protocol



UDP is the abbreviation of User Datagram Protocol. UDP only provides a basic, low-latency communication called a datagram. The so-called datagram is a kind of self-addressing information, data pack from the sender to the receiver. UDP protocol is often used for image transmission, network monitoring data exchange and other data transmission speed requirements are relatively high

1. Header format of the UDP protocol

   The UDP header consists of 4 domains, each occupying 2 bytes, as follows figure 4-2:

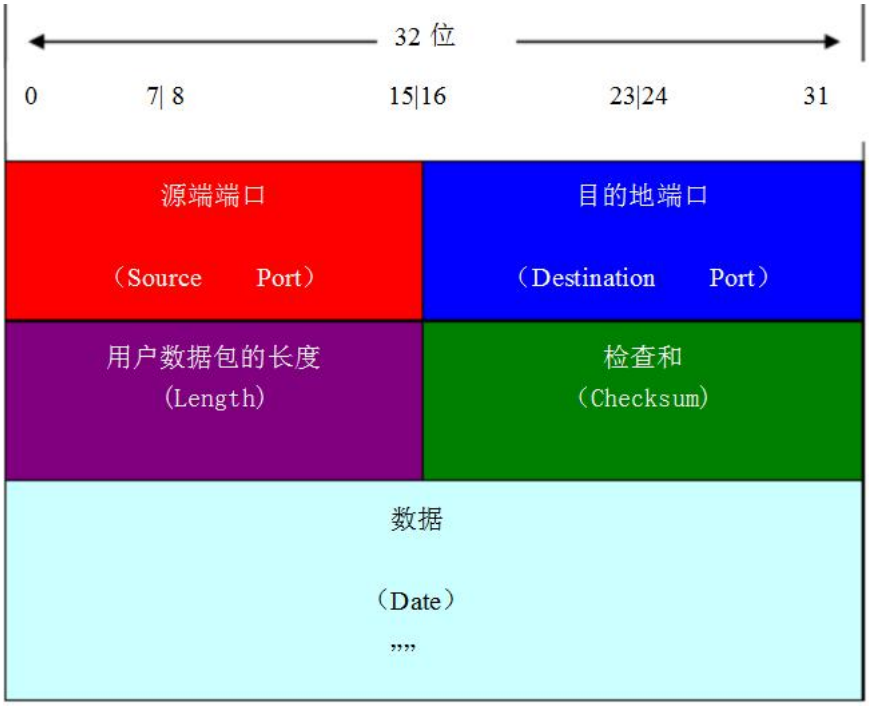


Figure 4-2: Header format of the UDP protocol

1) UPD Source Port
2) Destination Port
3) Length of Diagram
4) Checksum

The UDP protocol uses its port number to reserve its own data transmission channel for different applications. The data sender sends the UDP datagram through the source port, and the data receiver receives the data through the destination port.

The length of the datagram refers to the total number of bytes including the header and data parts. Because the length of the header is fixed, this field is primarily used to calculate the variable length portion of the data (also known as the data load). The maximum length of a datagram varies depending on the operating environment. In theory, the maximum length of a datagram containing a header is 65535 bytes. However, some practical applications tend to limit the size of the datagram, sometimes it will be reduced to 8192 bytes.

The UDP protocol uses the checksum in the header to ensure data security. The check value is first calculated by the data sender through a

special algorithm, and after being passed to the receiver, it needs to be recalculated. If a datagram is falsified by a third party during transmission or damaged due to line noise, etc., the calculated values of the sender and the receiver will not match, and the UDP protocol can detect an error. Although UDP provides error detection, when an error is detected, the error is corrected, simply throwing away the corrupted message segment or providing a warning message to the application.

2. IP Datagram Header

Since the UDP protocol packet is one of the IP packets, let us introduce the data format of the IP packet. The following figure shows the header format of the IP packet. The first 20 bytes of the packet header are fixed and the latter is variable.

Figure 4-3: Header format of the IP Packet

Revision:

4 digits, which refers to the version of the IP protocol. The current IP protocol version number is 4 (ie IPv4).

Head length:

4 bits, the maximum value that can be represented is 15 units (one unit is 4 bytes), so the maximum length of the IP header is 60 bytes.

Differentiated services:

8 bits, used to get better service, is called service type in the old standard, but it has never been used. In 1998, this field was renamed to differentiate service. This field is only works when using DiffServ. This field is not used under normal circumstances.

Total length:

16 bits, the length of the sum of the header and the data, in bytes, so the maximum length of the datagram is 65535 bytes. The total length must not exceed the maximum transmission unit MTU

Logo:

16 bits, it is a counter used to generate the identity of the datagram

Flag:

It is 3 digits, and currently only the first two meaningful MFs; the lowest digit of the flag field is MF (More Fragment)

MF=1 means "there are still fragments".

MF=0 means the last slice DF, One bit in the middle of the flag field is DF (Don't Fragment), which is allowed only when DF=0

Slice offset:

12 bits, which refers to the relative position of a slice in the original packet after fragmentation. The slice offset is offset by 8 bytes.

Time to live:

8 bits, recorded as TTL (Time To Live). The maximum number of routers that the datagram can pass through the network. The TTL field is initially set by the sender with an 8 bit field. The recommended initial value is specified by the assigned number RFC. The current value is 64. The TTL is always set to a maximum of 255 when sending an ICMP echo reply.

protocol:

8 bits, indicating which protocol is used for the data carried by this datagram, so that the IP layer of the destination host hands over the data part to which process, 1 is ICMP protocol, 2 is IGMP protocol, and 6 is TCP protocol. 17 is expressed as UDP protocol

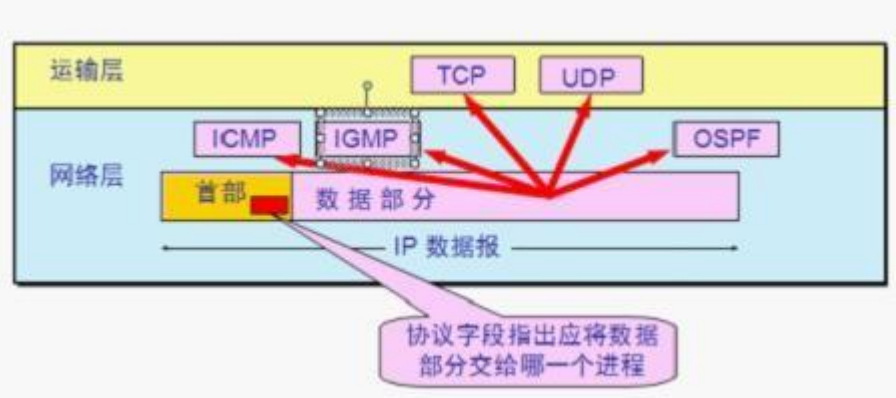


Figure 4-4: Protocol

Header Checksum

16bits, only check the header of the datagram without checking the data portion. Here, the CRC check code is not used and a simple calculation method is used.

Source address and destination address:

Each takes 4 bytes and records the source and destination addresses respectively.

4.2 Programming

Gigabit Ethernet GMII communication is a verilog code design based on the UDP communication protocol. The program function is to send UDP packets to the PC or network device through the network port by default after the board is powered on. The content of the data packet is "Hello Alinx AX301". When the FPGA detects the UDP packet sent by the network port, it stores the received data packet in the RAM, and then continuously sends the received data packet back to the computer or other network device through the network port.

The entire ethernet_test project consists of a top-level module ethernet_test.v, UDP test program udp.v and three sub-modules: UDP send module (ipsend.v), UDP receive module (iprecieve.v) and CRC check module (crc.v)

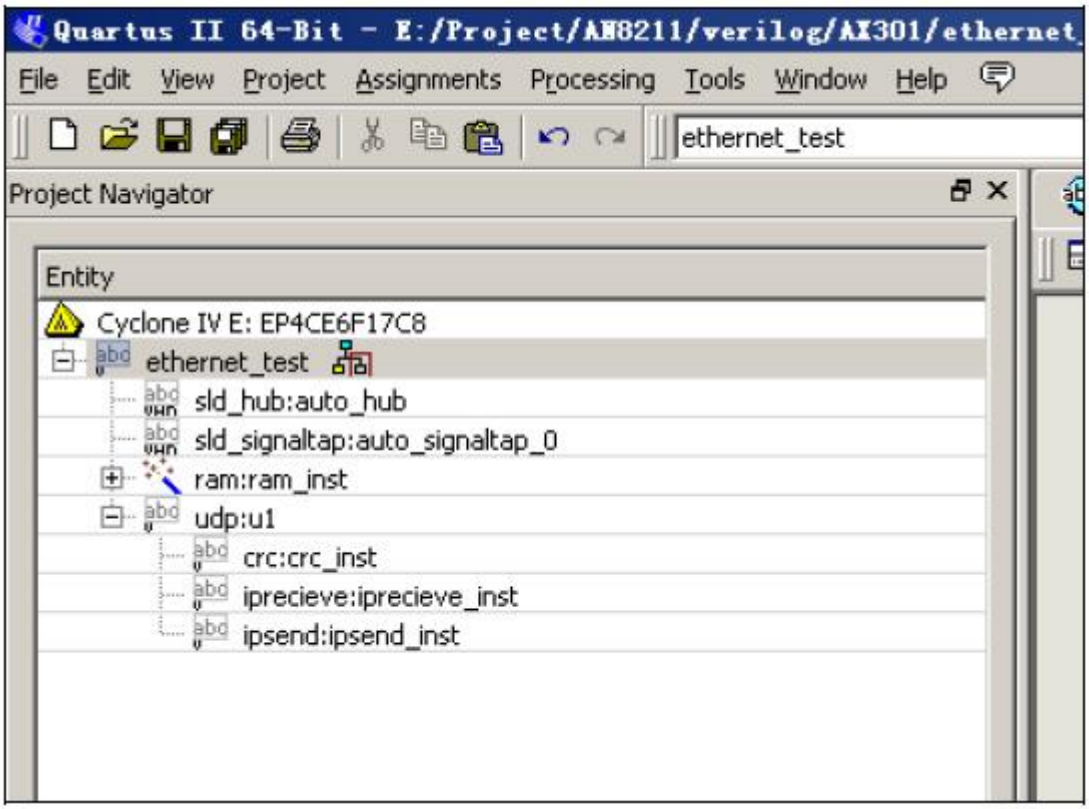


Figure 4-5: ethernet_test project

1. UDP Send Module (ipsend.v) Description

The UDP sending module combines the data in the RAM into a UDP IP packet format and sends it to the PHY chip through the GMII bus. The PHY chip then sends the data to the network port of the development board.

The IP packet header shall be sent before the IP packet is sent. The IP header consists of an 8-byte preamble, a destination MAC Address, a source MAC address, and a two-byte IP packet type. The preamble consists of 7 0x55 and 1 0xD5 bytes, indicating the beginning of an IP data packet transmission; the destination MAC Address is the MAC address of the object to which the data is to be sent. If the network port of the development board is connected to your PC, The value of the destination MAC Address is the MAC address of your PC. The source MAC Address is the MAC address local to the development board. The IP packet type value is 0x0800.

After sending the IP header, the IP datagram header is sent. The format of the IP datagram header has been discussed earlier. Then the

data in the RAM is sent, and finally the 4-byte CRC32 value is sent.

2. UDP Receive Module (iprecieve.v) Description

The UDP data receiver is the same as the UDP data transmission. First, it is judged whether the preamble is received. If the preamble is received, it starts to receive the target MAC Address and judges whether it is consistent with the MAC address of the development board. If it is consistent, the remaining header and data are received. When receiving data, the received 8-bit data is converted into a 32-bit data width and written into the RAM.

3. CRC Check Module (crc.v) Description

The CRC check of an IP packet is calculated at the destination MAC Address and is calculated until the last data of a packet. The verilog algorithm and polynomial of Ethernet's CRC32 can be generated directly at the following website in Figure4-6: http://www.easics.com/webtools/crctool

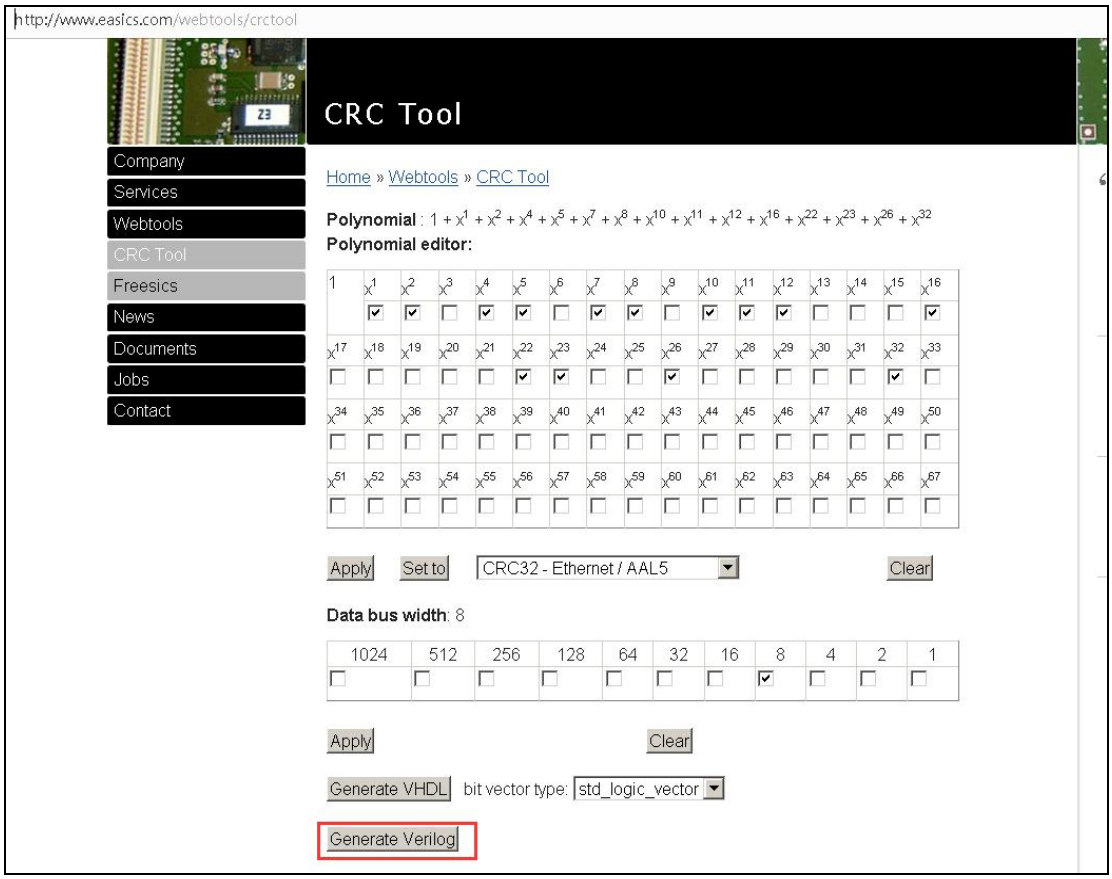


Figure 4-6: Website

4. Udp test module (udp.v) Description

The Udp test program just instantiates the three submodules written

earlier: UDP send module, UDP receive module, CRC32 check module

5. TOP module (Ethernet_test.v) Description

One function of the TOP program is to instantiate the udp module. In addition, a dual-port RAM is called. The data width of the read/write is 32 bits, and the address depth of the RAM is 512. The dual-port RAM is used to store the data received by the Ethernet. When the program does not receive the Ethernet packet, the program sends the character of "HELLO ALINX AX301" stored in the RAM all the way. If an Ethernet packet is received, the received Ethernet packet will overwrite the data in the RAM, and the program will continuously send the received packet to the network port.

Part 5: 100M Ethernet Program

The design principle of the 100M Ethernet program and the Gigabit Ethernet program is the same. The main difference is that the data bus is changed from GMII to MII, and the frequency of the clock is changed from 125Mhz to 25Mhz. Both the transmit clock and receive clock are provided by the GPHY to the FPGA. The Figure 5-1is a schematic diagram of 100 Mbps Ethernet data communication:

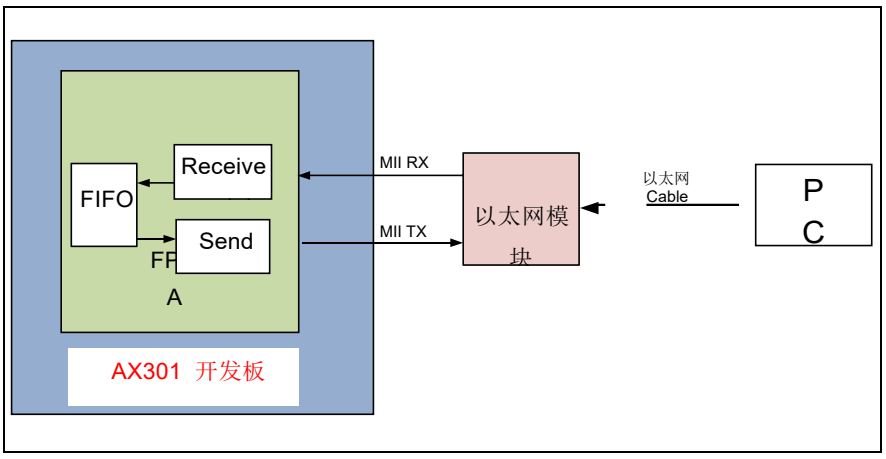


Figure 5-1: The Design Principle of the 100M Ethernet program

Since the data width of the MII bus is 4 bits, when data is transmitted and data is received, one byte needs to have 2 clocks to transmit or receive. The lower four bits of the data are first and the upper four bits are followed. The function of each module of the 100M Ethernet program is

the same as that of the Gigabit Ethernet test program, which is not introduced here.

Part 6: Ethernet Experiment

1. Prepare the Experiment

- Step 1:

  First, connect the network port of the development board and the network port of the PC with Cat5 + or Cat6 network cable, Then power on the development board. The LED of the Ethernet module network port will light up, indicating that the network is already connected to the link. You can confirm whether the direct link between the computer and Ethernet is Gigabit or 100 megabits. Click on the local connection in Figure 6-1 to view the status.

Figure 6-1: Local Connection

Note: If the network port is not connected to the link, it may be that the default program of the FPGA pull GPHY Reset pin low, so that GPHY is in the Reset state. As long as the FPGA downloads the Ethernet test program, the network port can be connected.

- Step 2:

If the computer's system is Window XP, you need to modify the target mac address in the UDP sending module (ipsend.v) to the mac address of the PC NIC, and recompile it after modification. If you don't know the mac address of your PC's NIC, look it up in the DOS command window with the ipconfig –all command.

Figure 6-2: Modify the mac address in ipsend.v

If the computer is not the Window XP operating system, you do not need to modify the target mac address in the UDP sending module (ipsend.v), and the destination address is all FF broadcast packets.

- Step 3

  Modify the IP address of the PC to 192.168.0.3P (detailed refer to Figure 6-3). The IP address of the PC needs to be the same as the setting in the sending module (ipsend.v), otherwise the network debugging assistant will not receive the UDP packets sent by the development board.

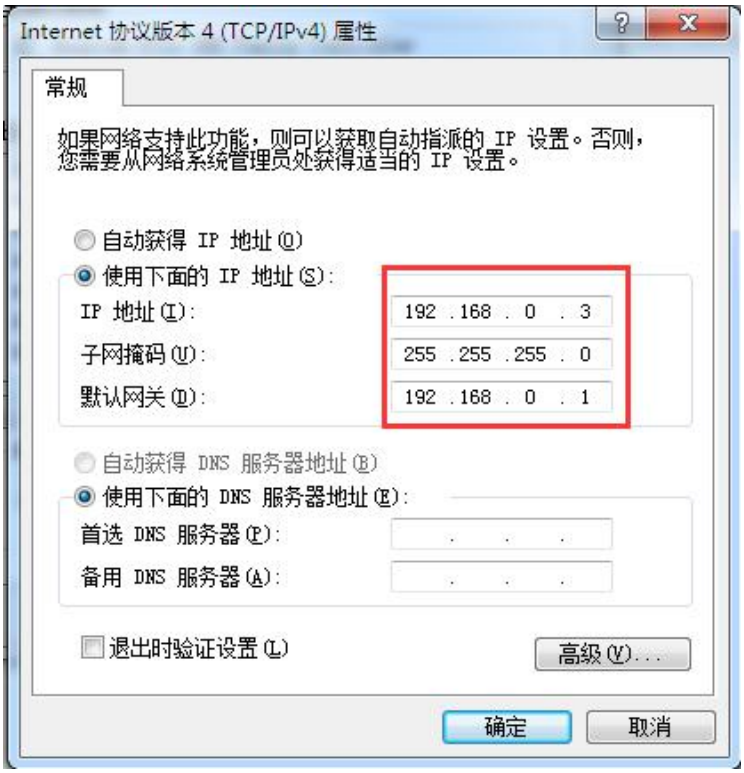


Figure 6-3: Modify the IP address of the PC

- Step 4

  Open the DOS command window with administrator privileges and bind the IP address and MAC address of the development board. When the network debugging assistant sends a network packet with IP address 192.168.0.2, the target MAC address is automatically the MAC address of the development board.

  Run the command: ARP -s 192.168.0.2 00-0a-35-01-fe-c0

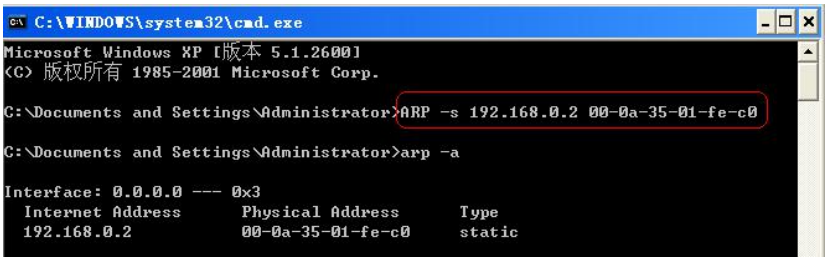


Figure 6-4: Run the Command

After binding, we can use the arp –a command to view the results of

the binding on the computer.

- Step 5

  This step is optional. Wireshark is installed to facilitate the debugging of the user's network communication. Wireshark, the network capture tool in the TOOL directory of the installation CD, can be used to view the detailed data sent and received by the PC network port during the experiment.

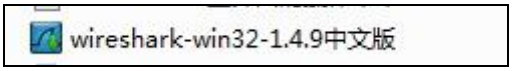


Figure 6-5: Wireshark Tool Installation Software

2. Ethernet Communication Test

- Step 1

  According to the speed of the network port link, it is judged whether to download the Gigabit test program or the 100M test program.

- Step 2

  Open the network debugging assistant we provided and set the parameters as follows, then press the connect button (the local IP address here is the computer's IP Address, the local port needs to be the same as in the FPGA program, it is 8080)

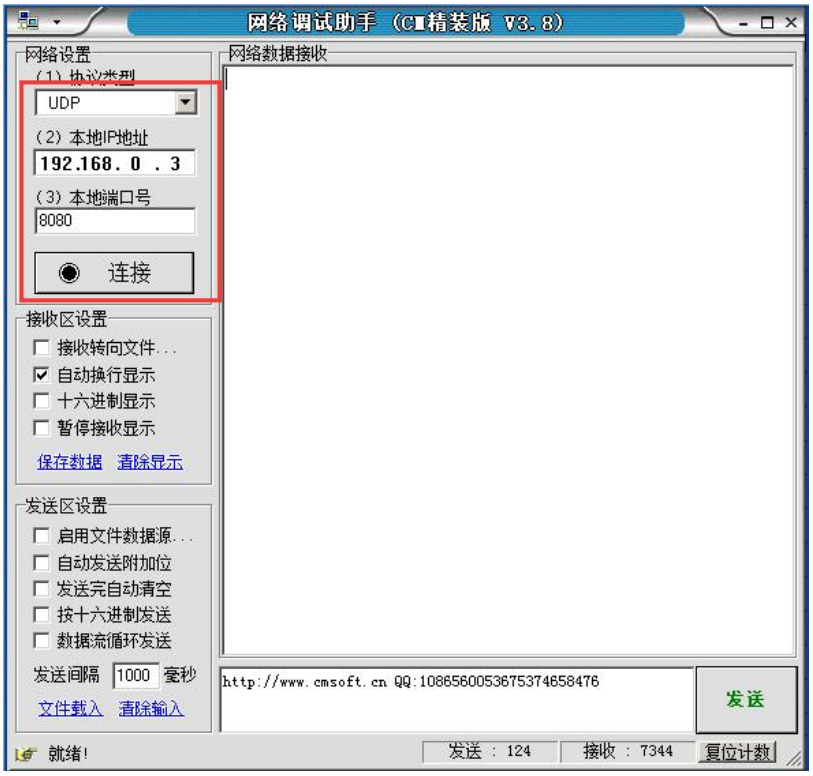


Figure 6-6: Network Debugging Assistant

At this time, the network data receiving window will display the Ethernet packet "Hello ALINX AX301" sent by the FPGA to the computer. We will set the IP address of the target host to be the same as the IP address in the FPGA program. The destination port number also needs to be the same as the FPGA program (8080).

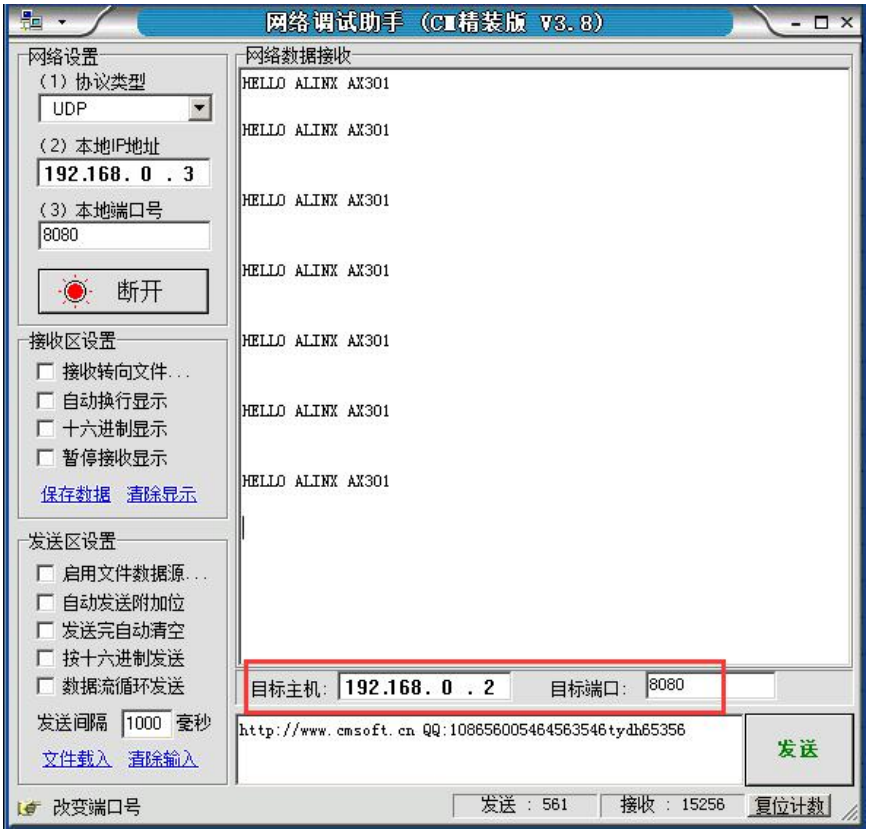


Figure 6-7: Network Debugging Assistant

Note: The transmission of Ethernet data frames has a packet length requirement of 46 to 1500 bytes. Therefore, when sending Ethernet packets, the length of the data frame should not be too short, otherwise the computer data packet will be sent and the FPGA will not receive the data packet.

- Step 4

  This step is optional for the user. If you want to view more information about the packet transmission, you can use the network capture tool Wireshark to view the network data received and sent by the computer's network card, and open the installed wireshark packet capture tool. , click on the menu capture package -> network interface.

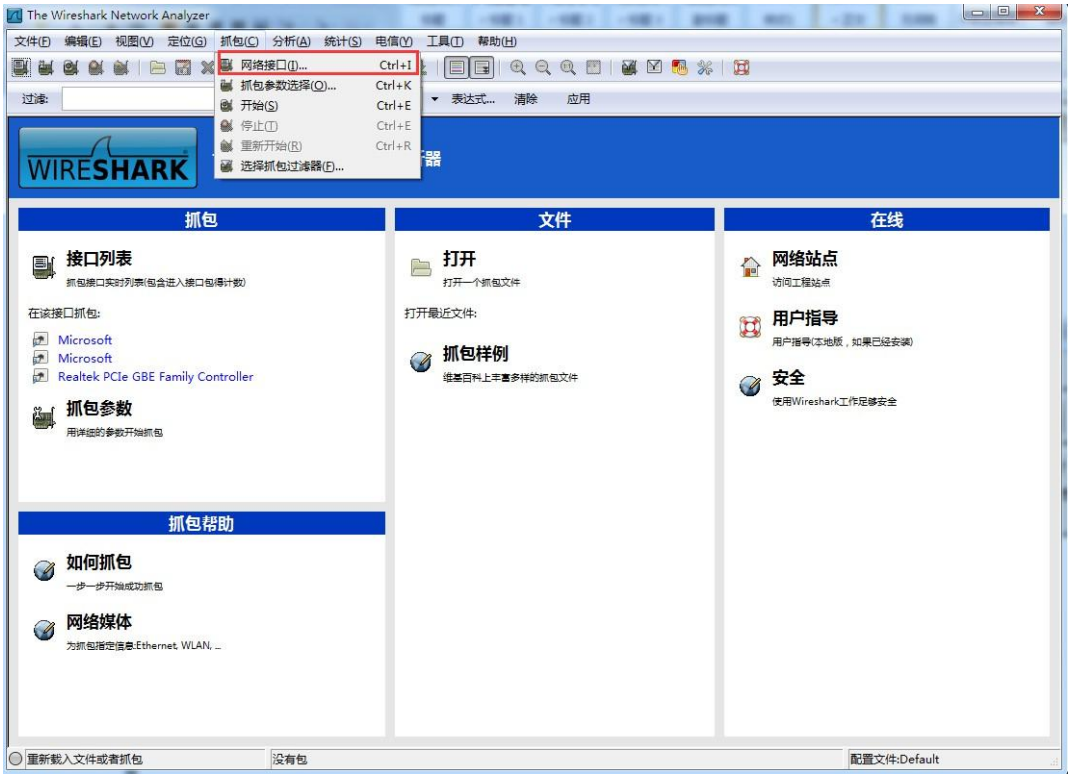


Figure 6-8: Wireshark Software

In the pop-up capture interface window, select your computer's Gigabit NIC and press the start button to start capturing packets.

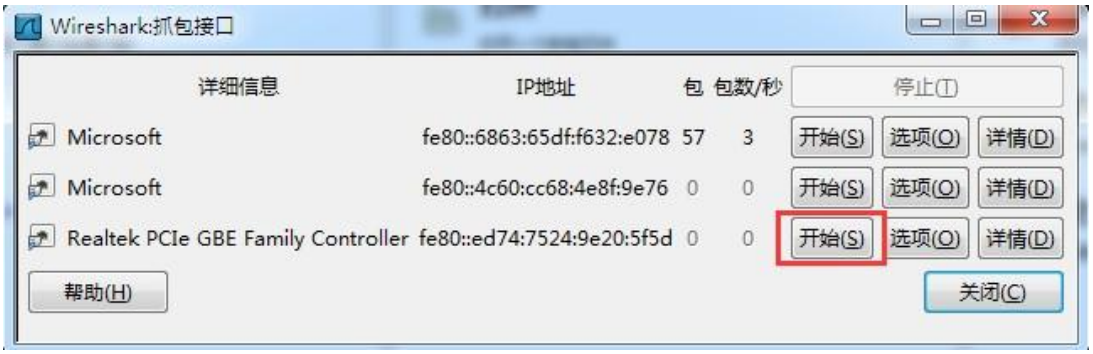


Figure 6-9: Pop-up Capture Interface Window

In the wireshark capture window, you can see the development board (192.168.0.2) to the computer network port (192.168.0.2) sent data packets.

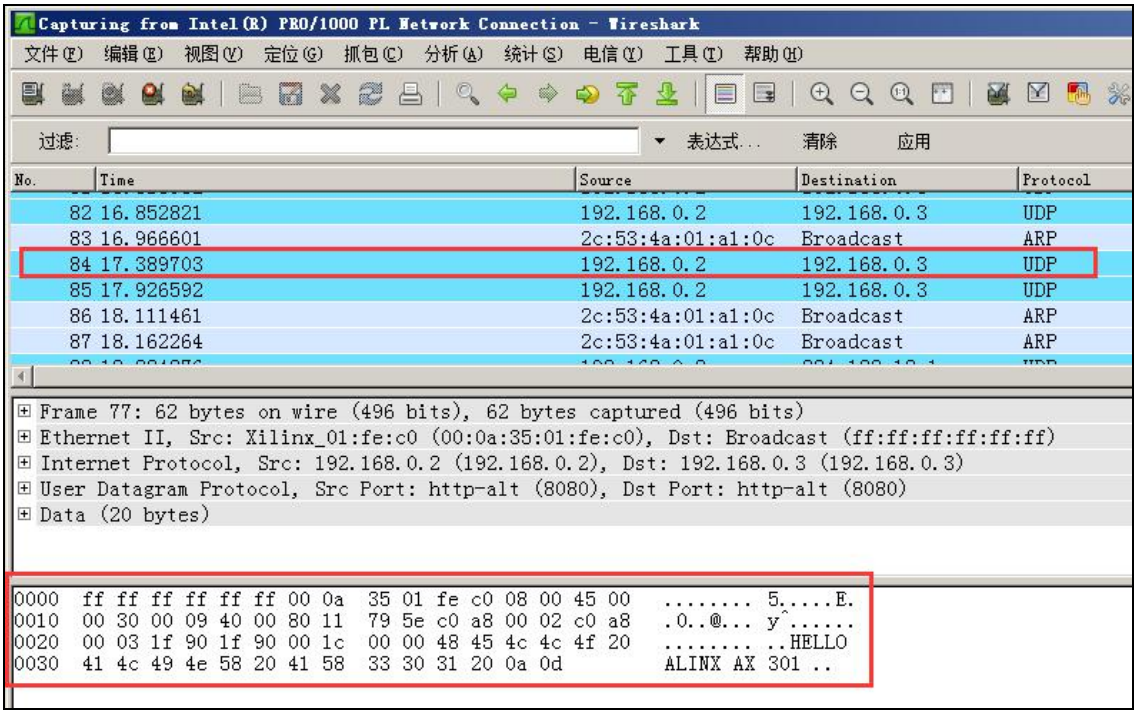


Figure 6-10: The wireshark Capture Window

After actual testing, the data speed of UDP communication can reach more than 900Mbps, which is very suitable for high-speed data transmission occasions, such as video image transmission and high-speed data acquisition.