Application Research of Ethernet Communication System Based on UIP Protocol Stack

Xie Wendi¹, Yao Ling² and Wang Enliang¹*

¹Communication, AnHui XinHua University, HeFei, AnHui, 230088, China
²Communication, AnHui XinHua University, HeFei, AnHui, 230088, China

About the author: Xie WenDi (1987-), female, from YongZhou, HuNan province, graduated from NanJing university of posts and telecommunications in 2012. Now she is a lecturer of anhui xinhua university.

*443183239@qq.com

Abstract. This paper elaborates an implementation method of embedded system access Ethernet based on UIP stack, and describes the scheme design, hardware design and software design in detail. This system takes DSP (Digital Signal Processing) chip TMS320VC5510 as the control core, and chooses CS8900A as the network controller. Through the design of the protocol stack and the development of the underlying driver, it completes the ICMP test and TCP test, and it is applied to the portable communication equipment to realize data communication in Ethernet. The process of realization is simple and the cost is low.

1. Introduction

In recent years, embedded technology has developed rapidly and has been widely used in many electronic devices (such as handheld and portable devices). The embedded system must have the function of network interconnection in order to communicate with devices quickly. Ethernet is the most widely used LAN technology. Because some embedded systems require high real-time performance of the network, this paper uses the microprocessor+compact TCP/IP protocol stack--UIP protocol stack to achieve embedded system networking and voice processing functions. Its design and application are described in detail below.

2. The project design

The system is mainly composed of hardware including computer, portable measuring instrument and network equipment, and software including CCS(Code Composer Studio) development platform, serial debugging assistant and network monitoring tool. CCS platform is responsible for the program development and debugging, serial debugging assistant used to display the sent and received data, network monitoring tools to complete the data interception and analysis. The computer provides the platform of software development and debugging to complete the function of software simulation. The network device is a talker that can both speak as a speaker and as a receiver at a communication rate of 10M, and that is to say, voice data can be sent and received. Portable measuring instrument is a piece of measuring equipment that can complete the communication between PC and network equipment. It can send audio signals to network equipment and also receive signals. It mainly determines the receiving and sending functions of network equipment based on the frequency test of audio data sent by network equipment. The system design block diagram is shown in figure 1.
3. Hardware Design

The portable measuring instrument is mainly composed of four modules, including audio generation, network interface, matrix switch control and audio measurement module. The audio generation module generates audio sources, and the network interface module provides a bridge for the input and output of audio signals, and the matrix switch control module controls the channel selection of audio signals, and the audio measurement module filters and computes the received audio signals. The audio generation module generates a sinusoidal audio signal of a certain frequency from a two-channel audio signal generator through two-tone synthesis, amplification and impedance matching, and sends the signal to the network equipment through an Ethernet bus. After receiving the signal, the network equipment returns a signal of the same frequency and different amplitude to the audio measurement module through the network interface module, and finally completes the functions of A/D conversion and measurement. The block diagram of its module connection and wiring diagram of pin are shown in figure 2 and 3 respectively.

Figure 1. System design block diagram.

Figure 2. Module connection diagram.

Figure 3. Pin wiring diagram.

TMS320VC5510 chip is the first fixed-point TMS320C55x core produced by TI company, with low power consumption and high performance 16 bit digital signal processor. The CS8900A Ethernet controller is designed by Cirrus Logic, and is primarily used in industrial control applications such as embedded systems with high reliability requirements, portable products and some adapter cards,
providing a practical solution for the development of embedded Ethernet.

4. Software Design

4.1. CS8900A Programming Design
This system uses the polling mode of CS8900A I/O access to read and write datas.

1. Initialization process of CS8900A
First, it sets the direction of the GPIO pin (input or output), and then resets the CS8900A software by writing the REG15_SelfCTL register. Then it reads the REG16_SelfStatus register, and waits until INITD=1, indicating the successful reset of CS8900A. Finally, the internal register of CS8900A is configured. The first step is to load parameters and write the corresponding registers, including the configuration of full duplex mode, enable of receiver and transmitter, etc. The second step is to load the MAC address, that is, to assign the MAC address to the communication device, which is used to determine whether the frame received by CS8900A belongs to the host.

2. Reading process of CS8900A
When CS8900A receives a data frame, it calls the Ethernet_driver_read() function, which loads the frame into the UIP_buf[] receive buffer. The receiving register of frame length returns the length of the frame, and determines whether to discard or receive by whether the frame length exceeds the maximum buffer size. This process is the operation of repeatedly reading the values of the receive or send data port such as CS8900A_RTData.

3. Writing process of CS8900A
The sending process of the CS8900A controller data is initiated by writing to the send command register (TXCMD) and the send length register (TXLength). Then loop through the REG18_BusSTATUS bus status register to determine if CS8900A is ready to receive frames. After checking that CS8900A is ready to receive frames, first use CS8900A_RTDATA0 port to send 54 bytes of frame head length, then use CS8900A_RTDATA0 port to write frame data, and then start the whole transmission of frame data.

4.2. Protocol Stack Programming
DSP completes the core control functions as well as the reading and writing of packets. It controls the registers in the CS8900A Ethernet control chip through programming. The protocol stack acts as a bridge connecting the bottom and top applications of the system, so as to realize the communication function of embedded system accessing Ethernet. The interface relationship is shown in figure 4.

This paper selects the UIP protocol stack developed by Adam Dunkels from the Swedish school of computer science, which is a free and easy to implement and very small specifically for 8 bit and 16 bit processor TCP/IP stack. UIP protocol stack removed the functions not commonly used in the complete TCP/IP stack, which simplify the communication process and occupy less resources, thus the processing speed has been increased.

The UIP stack mainly supplies three functions such as uip_init(), uip_input(), and uip_inquire().

![Figure 4. Interface diagram.](image-url)
which is called by the underlying system. Its primary interface to the application is UIP_APPCALL(). uip_init() is called during system initialization, and the main initializer stack listener port and all connections are off by default. The uip_input() function is the underlying entry to the UIP stack that handles incoming IP packets. uip_inquire() function checks the state of the connections specified by the parameters. UIP stack implements the interface with the device driver through the functions uip_input() and global variables uip_buf and uip_len. uip_buf is used to store the received and to be sent packets. uip_len represents the length of data in the receiving and sending buffer, and determines whether the value of uip_len is 0 to determine whether new data has been received and whether there is data to be sent.

UIP protocol stack mainly includes IP namely Internet protocol, ARP namely address resolution protocol, and ICMP namely Internet Control Message protocol and Transport Protocol such as TCP namely Transmission Control Protocol or UDP namely User Packet Protocol.

Figure 5 shows the main program flow chart of the UIP system. Initialization work done by the underlying hardware, mainly to the physical layer interface for configuration and binding, and the packet receiving process of the UIP stack is actually the process of data packet parsing. When network drive device sends some packets, it first reads the data packets and decides whether it is an IP packet or an ARP packet. If it is an ARP package, ARP processing is performed; If it is an IP packet, then the upper layer protocol is judged to be UDP or ICMP, and then it is processed separately. When the data is sent, the network driver device will determine the packet by itself and complete the encapsulation of the packet.

5. System testing and application

5.1. ICMP test
When the computer is used for simulation, the IP address of the target machine, namely the portable measuring instrument is set as 129.27.137.103, which is different from PC’s 129.27.137.104. That is, the local area network (LAN) is formed to enable them to communicate with each other in the same segment of the network. Assuming a successful communication link has been established, this should happen when the server issues a command and the target machine responds immediately. The ICMP
test diagram under DOS operating system is shown in figure 6.

Figure 6. ICMP test of DOS system.

According to figure 6, it can be seen that all the four ICMP packets sent have been responded, among which the longest response time is 18ms, the shortest response time is less than 1ms, and the average response time is 10ms. This indicates that the development board has established a reliable connection with the computer, and the ping communication has been successful, which provides a reliable basis for the realization of various functions of the system.

5.2. TCP testing
TCP data transmission is carried out according to the Socket protocol, and the communication flow between the client and server is shown in figure 7 and 8 respectively.

Figure 7. Client Socket communication flow. Figure 8. Server Socket communication flow.

The communication mode between Socket and C5510 is to set the IP address of server C5510 to 129.37.137.103 and use 6000 port number, then click on the connect button, and the equivalent of connection requests by the client has used, and the server is opened passively, running on its uIP protocol stack in the listening state, which receives the request and responses to a synchronous confirm frame, and then the client send the end of the three-way handshake handshake at a time, after the client and the server are entering the stage of establishing a connection, and they can send and receive data.

If the connection is not successfully established, the word "It is not connected" will be displayed in the text edit box after clicking the connection button. After successful connection,"connected to 129.27.137.103" will be displayed in the text box of the interface, as shown in figure 9.
This paper realizes to send or receive data arbitrary between the Socket and C5510 through the debugging program. C5510 simply return the received data to the Socket, such as "123456789" entered in the edit box, and then click the "send" button, then the next text in the edit box shows "123456789", that means the client sends data was successfully received by the server, and return the same data, as shown in figure 10.

![TCP connection between DSP and host. Sending and receiving of TCP packets.](image)

5.3. Application analysis

After the communication between the portable measuring instrument and the network equipment is successful, the data monitored by the network analysis system is shown in figure 11.

In figure 11, UDP packets is received by measuring instrument, because the network equipment of voice data using UDP/IP protocol encapsulation, its physical layer adopts Ethernet II protocol, and the frame header takes 14 bytes, containing the destination IP address, source IP address and protocol type, and the IP header takes 20 bytes, and UDP header accounts for 8 bytes. Namely the UDP frame first accounted for a total of 42 bytes, and the rest is the UDP data.

Time/Frequency waveform is drawn by using the data graphic display function provided by the CCS development platform, as shown in figure 12. Abscissa represents the sampling time, and ordinate represents the amplitude, according to former 42 byte department headed by 21 points, namely the start for the data, and the sine wave consists of eight sampling points, and the audio sampling frequency is 8 KHZ, so the frequency of the sine wave of 1 KHZ conforms to the requirements of the system, and audio signal is sent and received correctly, which indicates that the network equipment works normally, so it completes the basic function of measuring instrument, and realizes the application of the embedded UIP protocol stack access networks.

![Time/Frequency waveform](image)
6. Conclusion

In this paper, a computer simulation test is carried out to complete ICMP and TCP communication between DSP and PC, and realizes the function of embedded system access Ethernet based on DSP and UIP protocol stack. On this basis, its application is further discussed through data communication between measuring instrument and network equipment. However, the problem that still needs to be solved is that, due to the unreliability and connectionless orientation of UDP protocol, it is inevitable for packets to arrive out of order and packets to be lost. Therefore, for these phenomena, some processing can be carried out on the data, such as sequential marking of each group of data, so that the loss and out-of-order of data can be found in the application layer and thus be corrected. Adding a check code before and after the packet can ensure correct data reception, etc., which will be discussed in the following work.

Acknowledgment
Project support: Anhui xinhua institute of fine courses - school - enterprise courses (2017jpckx16)
Textbook of anhui xinhua university (2017zbjc002)
Teaching and research project of anhui xinhua university (2018jy025)

References
[1] Zhao HongLiang, Huang HeSong. (2008) Design of TMS320C55xDSP application system. Beijing university of aeronautics and astronautics press, Beijing.
[2] Adam Dunkels. Full TCP/IP for 8-Bit Architectures. (2003) In Proceedings of the first international conference on mobile applications, systems and services, San Francisco.
[3] Adam Dunkels. (2002) UIP-A Free Small TCP/IP Stack.
[4] Lv Tao. (2016) Overview of Ethernet TCP/IP protocol stack based on embedded system. Fujian computer, 7:99-99.
[5] Mu ShuQing. (2015) Transplantation and analysis of UIP protocol stack based on STM32[D]. Tianjin university.
[6] Liu Tao Tang, Xiao Chuan. (2017) Implementation of abstract layer based on uCOS and UIP[J]. Journal of mudanjiang normal university (natural science edition), 2:60-63.
[7] Kong QingFang. (2016) Design of laser marking system based on embedded Ethernet communication. Journal of tianjin vocational college, 18:101-106.
[8] Cao Yu, Li ZiHua. (2016) Design of signal acquisition system for wheel weight sensor based on embedded Ethernet[J]. Railway rolling stock, 36:14-17.