Design and Implementation of Communication Scheme between Ethernet and Multi-Serial Port

Xin He[1], Jia'nan Wu[1], Hongde Deng[1], Zean Zhen[1], Chenyang Liu[1]

[1]*School of Automation, Northwestern Polytechnical University, Xi'an, China*

Abstract. With the development of aerospace technology, the flight control system is getting more and more important for a UAV (Unmanned Aerial Vehicle) flying safely and efficiently. For collecting the experimental data without delay, this paper briefly reviews the design of the communication scheme, and provides the implemented results. Through using the controller LPC1768 to expand the serial port, and the Ethernet controller DP83848 to complete the communication by UDP protocol, it turns out that this method is able to reach the real-time requirements of the UAV semi-physical simulation.

1 Introduction

UAV (Unmanned Aerial Vehicle) its own complex function with numerous sensors, and the sensor information is transmitted by serial communication interface. For the limited serial ports of the simulation host, it is necessary for the serial ports expansion. The general method is to expand multiple serial cards on the host, however, an ordinary serial card can not guarantee access to real-time data, with complex design and long development period.

Due to the rapid development of Ethernet technology in the industrial control field, the continuous improvement of performance and the rapid reduction of cost, it is widely used in the new generation of industrial automation networks [1-2]. Therefore, this project uses Ethernet-based serial port expansion, LPC1768's own Media Access Controller (MAC) module and external physical layer (PHY) chip DP83848 to achieve the Ethernet physical layer protocol, and the ultimate realization of the communication between Ethernet and serial ports is based on Real-RAM library in Keil.

2 System Hardware Design

The system adopts the low-power, cost-effective NXP microcontrollers LPC1768 [3-5] of the LPC17xx family as the main controller based on ARM Cortex-M3 core with up to 100MHz operating frequency, up to 512KB Flash, up to 64KB SRAM, 8-channel 12-bit A/D converter, 10-bit D/A converter, Ethernet MAC, 4 UARTs, 2 CAN channels, 3 I^2C interfaces, and internal RC oscillator.

Firstly, we need to design a power supply circuit using R-783.3-0.5 to provide 3.3V and a reset circuit. The serial port supports the RS-232, RS-422, RS-485 operating modes by SP330E. For the Ethernet communication, an external physical layer (PHY) chip DP83848 is required, which supports 10/100MHz transmission rate and accords with the IEEE. The network interface HR911105A comes with network transformer and can realize signal isolation. In order to store the configuration parameters of each serial port, a two-wire serial E^2PROM AT24C256C is adopted, which is connected to the processor through the I^2C, and provides serial electrically repeatedly erasable memory and power-down protection. The structure of the whole system is shown in Fig. 1.

3. System Software Design

3.1 Ethernet Communication

Ethernet link layer is controlled by Ethernet controller DP83848 [6] and the function of MAC is implemented by LPC1768, and the Ethernet controller is controlled by the LPC1768 to complete the entire Ethernet communication transmission. Ethernet module initialization flow chart is shown in Fig. 2.

Real-Time Library (RL-ARM) in Keil contains real-time operating system RL-RTX which supports multi-tasking and RL-TCPnet which describes the TCP/IP Protocol Stack Suite component of RL-ARM explaining the various TCP/IP features, and helps in creating embedded applications that can connect to the Internet. Before using, we should install rlarm, and then it can be achieved by simply calling the relevant function.

The TCP protocol is connection-oriented and the UDP protocol is connectionless-oriented. Although the TCP protocol is reliable, the UDP protocol does not have to establish a connection between the client and the server,

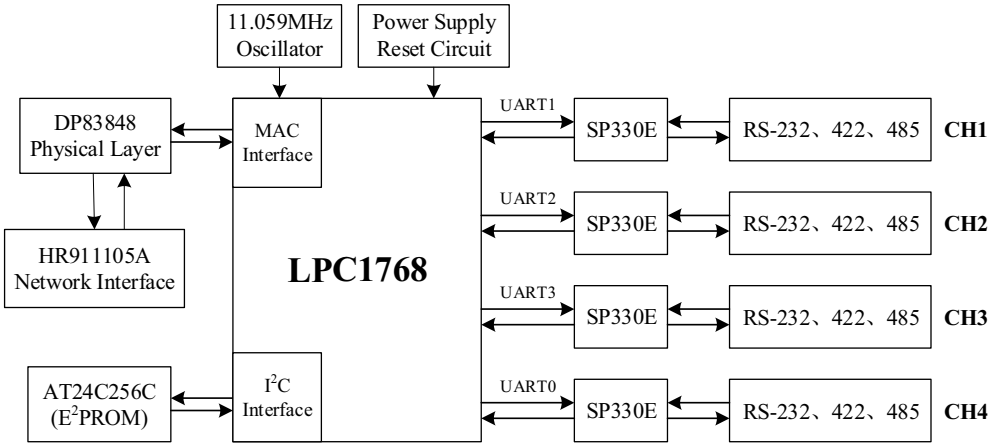

Figure 1. Structure of the Whole System

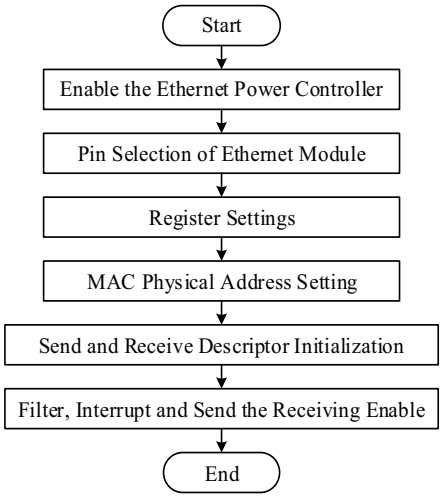

Figure 2. Ethernet Module Initialization

and the transmission speed is fast. In this programming, we adopt the UDP protocol.

During the communication, firstly, use the function udp_get_socket to assign socket. After that, call the function udp_open to open the communication socket, and bind it to a specific port. In order to address different serial ports, each serial port is assigned a dedicated port number to communicate. While sending data, call the function udp_get_buf to allocate the buffer, return the first address pointer of the buffer, and then call the function udp_send for data transmission. After the data are sent, call the function udp_close to close the communication socket, and finally call the function udp_release_socket to release the allocated memory.

3.2 Serial Port Configuration

The universal asynchronous transceiver UARTn [7] in LPC1768 has a 16 B receive and transmit FIFO and a built-in baud rate generator. The baud rate is calculated as in (1):

$$\mathrm{UARTn_{bd}} = \frac{\mathrm{PCLK}}{16\times(\mathrm{UnDLM+UnDLL})\times\left(1+\frac{\mathrm{DivAddVal}}{\mathrm{MulVal}}\right)} \quad (1)$$

where PCLK is a peripheral clock, FPCLK = FCCLK/4 = (FOSC*8)/4 = 22.1184M, UnDLM and UnDLL are UART divider registers, DivAddVal and MulVal are the values of the fractional divider, set to "0" when not in use.

It supports three kinds of operating modes RS-232, RS-422, RS-485, which is realized by sitting the pins 13 and 14 of SP330E.

Table 1. Pin Selection

Pin		Operating Modes		
		RS-232	*RS-485 (Full Duplex)*	*RS-485 (Half Duplex)*
13	RS485/$\overline{\text{RS232}}$	0	1	1
14	HALF/$\overline{\text{FULL}}$	×	0	1

3.3 Data Frame Reception Mode

3.3.1 Frame Header + Frame Length

In the process of reading the serial port receive FIFO, count the number of bytes received. According to the frame header length set, when the number of bytes received is greater than the frame header length, it needs to be compared with the frame header set; if the frame header has been found, when the number of bytes received is equal to the frame length, the whole data section can be sent by UDP; if the frame header is not found, continue to judge the newly received bytes.

3.3.2 Frame Header + Frame Tail

In the process of reading the serial port receive FIFO, count the number of bytes received. According to the frame header length set, when the number of bytes received is greater than the frame header length, it needs to be compared with the frame header set; if the frame header is not found, continue to judge the newly received bytes; if the frame header has been found, it needs to be compared with the frame tail set; if the frame tail has been found, the whole data section can be sent by UDP; if the frame tail is not found, continue to judge the newly received bytes.

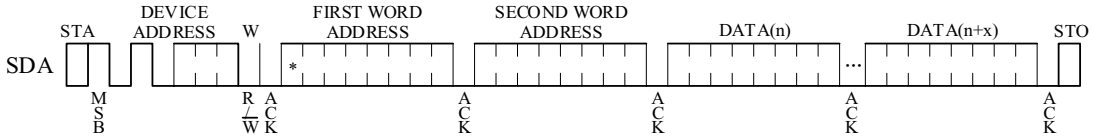

Figure 3. Write Operations.

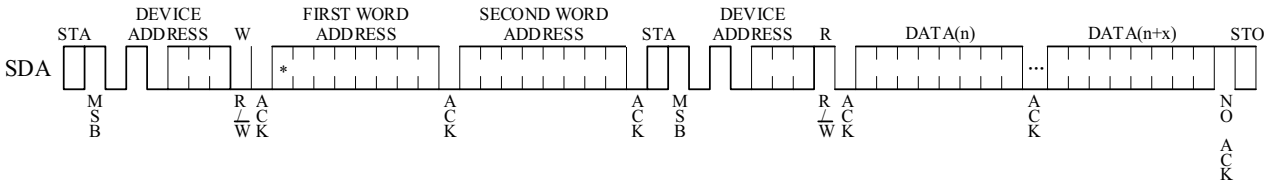

Figure 4. Read Operations.

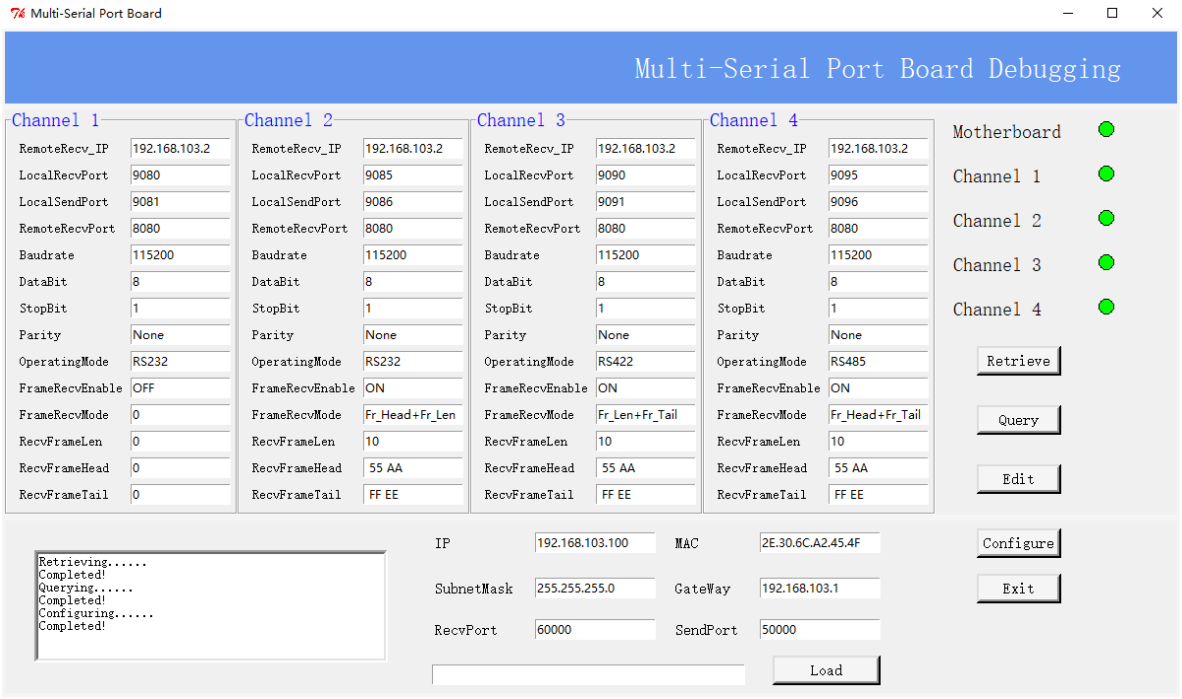

Figure 5. Upper-Computer

3.3.3 Frame Length + Frame Tail

In the process of reading the serial port receive FIFO, count the number of bytes received. According to the frame tail length set, when the number of bytes received is greater than the frame tail length, it needs to be compared with the frame tail set; if the frame tail has been found, when the number of bytes received is equal to the frame length, the whole data section can be sent by UDP; if the frame tail is not found, continue to judge the newly received bytes.

3.4 Memory Operation

The Two-wire Serial E^2PROM AT24C256C [8] supports bidirectional data transfer protocol, with 256K of storage space, which is internally organized as 512 pages of 64-bytes each.

3.4.1 Write Operations

A write operation requires two 8-bit data word addresses following the device address word, the read/write select bit (R/W) set to “0” and acknowledgment. Upon receipt of these addresses, the E^2PROM will respond with a “0” after each data word received. After the data are sent, the controller must terminate the write sequence with a stop condition, as shown in Fig. 3.

3.4.2 Read Operations

Read operations are initiated almost the same way as write operations. Once the device address word and data word address are clocked in and acknowledged by the E^2PROM, the controller must generate another start condition. Upon receipt of the device address, the read/write select bit (R/W) set to “1” and acknowledgment, the E^2PROM will respond with a “0” after each data word received except

that the last data word is sent with the no acknowledgment. Then following a stop condition, as shown in Figure 4.

3.5 Upper-Computer

The upper-computer is a computer that can issue commands directly and show various signal changes on the screen. The upper application software, which is developed based on Python, firstly retrieves the mainboard by sending broadcast. On this premise, it can query and modify the MAC address, IP address, gateway, sending and receiving port of the module, and baud rate, stop bit, check bit, data bit, frame header, frame length, frame tail and operating mode parameters of each serial port, as shown in Fig. 5.

4 Results

In the process of testing, the IP address of the PC is 192.168.103.2 and the port number is 8080, the IP address of the mainboard is 192.168.103.100, the sending port is 50000 and the receiving port is 60000. The receiving mode of channel 4 is frame length + frame tail, the frame header is 0x55, 0xAA, the frame tail is 0xFF, 0xEE, the frame length is 10. It is tested by the COM debug assist and network debug assist, as shown in Fig. 6 and Fig. 7.

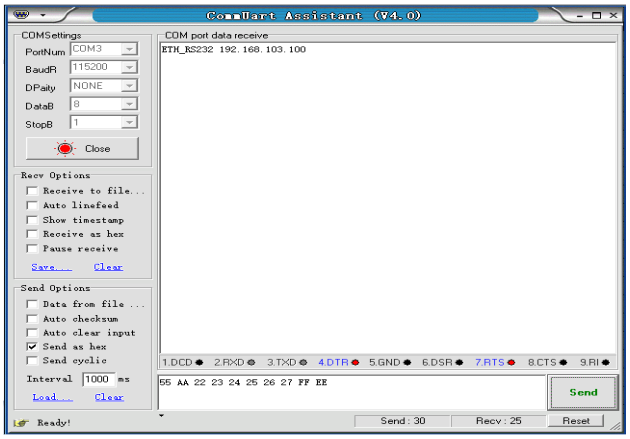

Figure 6. Uart Assist

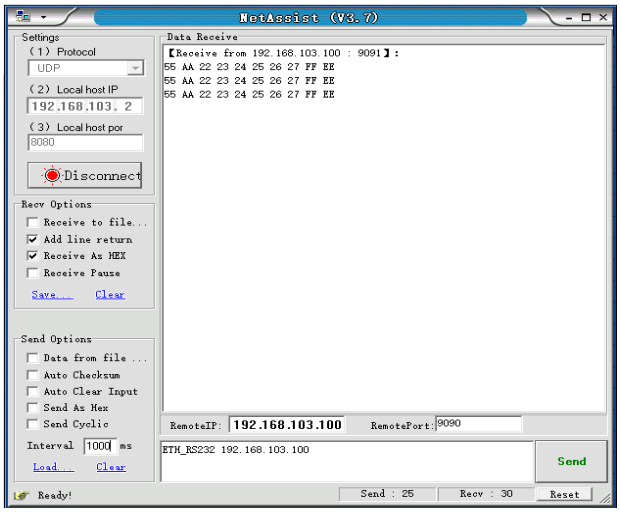

Figure 7. Net Assist

5 Conclusion

In this paper, the system uses the main controller LPC1768 for serial port expansion, and Ethernet control chip DP83848 is used to design Ethernet controller to realize the communication between Ethernet and multi-serial port. On the basis of supporting the RS-232, RS-422 and RS-485, the receiving data supports three modes of frame header + frame length, frame header + frame tail, frame length + frame tail. The results suggest that this scheme has the characteristics of reliability, real-time and low cost, and has preferable practical value.

Acknowledgment

Thanks for my supervisor Pro. Wu who gave me considerable help, encouragement and unwavering support. In addition, I deeply appreciate the contribution to this thesis made in various ways by my friends.

References

1. J. D. Decotignie, “Ethernet-Based Real-Time and Industrial Communications,” Proceedings of the IEEE, 2005, 93(6):1102-1117.
2. C Zou, “The Design of Data Acquisition and Analysis System based on Multi-Serial Port Turn to Ethernet,” Shanghai: Donghua University, 2017.
3. H Deng, Y Li, J Wu, “Design of Embedded Communication System Between Multi-Channel UART And Ethernet Based on RL-ARM Library,” Measurement and Control Technology, 2012, 31(8):75-79
4. A Sun, “ARM COTEX-M3 Embedded Devolpment Example: Based on NXP LPC1768,” Beihang University Press, 2012.
5. H Yang, Q Shao, Z Zhang, “Research and design of LPC1768 data check and encryption circuit,” Modern Electronics Technique, 2014.
6. P Zhao, P Sun, “Solution of Ethernet Communication based on DSP,” Microcomputer and application, 2014(7):57-59.
7. Z Zheng, A Hu, “Full Duplex UART Software Simulation Based on LPC1768,” Microcontrollers & Embedded Systems, 2013.
8. Q Zhou, J Liu, “Application of AT24C256 in Single Chip Microcomputer System,” Electronic Component & Device Applications, 2002.