

# The Design and Implement of TCP/IP Protocol Cluster on AVR Singlechip

Rong Pan, Hai Zhao, Jialiang Wang, Dan Liu, Penghua Cai

School of Information Science and Engineering, Northeastern University, Shenyang, China.  
Email: panrong1012@126.com

Received June 3<sup>rd</sup> 2010; revised June 30<sup>th</sup> 2010; accepted July 6<sup>th</sup> 2010.

## ABSTRACT

*With the rapid development of the embedded technology, research and implement of the Internet of things will be a new technology revolution, yet the implement of the Internet of things is on the base of the communication between the things. For this reason, realizing the function of communication between singlechip is particularly important. Based on the characteristics of the embedded microcontroller, we analyzed the traditional PC TCP/IP protocol, and appropriately tailored TCP/IP protocol cluster on the basis of the characteristics of embedded singlechip. At last, we realized the reduced TCP/IP protocol cluster suitable for embedded singlechip, on AVR singlechip platform.*

**Keywords:** *The Internet of Things, Webit, Embedded System, TCP/IP Protocol, Ethernet*

## 1. Introduction

The Internet of things means a kind of net that via information sense equipment such as FRID, infrared sensor, GPS, laser scanner and so on, in arranged protocol, join up between anything and the Internet to communicate information and realize intelligent identification, tracing, monitoring, and management. The concept of the Internet of things is suggested in the year of 1999. It is the "Internet communicated with things". It means two aspects: the first one is, the core and foundation of the Internet of things is still in the Internet, which based on, extending and expanding the Internet; the second one is, its client side extends and expands to anything, to make the information exchanging and communication [1-3].

With the rapid development of computer and network technology, Internet has become an important means of information transmission, more and more embedded equipments are necessary to achieve the Internet's network [4-5]. Relative to the PC, computing and storage resources of embedded systems are relatively limited; therefore to achieve all the TCP/IP protocol cluster in the embedded singlechip is quite unrealistic. So that, in order to save the system resources and ensure the reliability of the system, under the condition of improving the performance of embedded system, it's necessary to targeted modular simplify TCP/IP protocol.

## 2. Adoptive Equipment and Testing Platform

Webit is an overall solution that makes the equipments

intelligent and networking. It is the new network equipment system structure with the elements of Internet and its basic idea is an independent, low-cost 3 W server embedded in equipment, to make the equipment has independent network intelligence.

Webit is an embedded Internet product decided by Liaoning Provincial Key Laboratory of Embedded Technology by themselves. Webit 1.0 is successfully pass technical appraisal and the trademark registration in the year of 2000, and Webit 2.0 (Internet non-standard electrical equipment access server) passed the appraising meeting of scientific and technological achievements held by science and technology commission of Liaoning province in May 2001. Considering that webit is AVR 8 bit singlechip, its storage unit is very limited, therefore it's very important to design a kind of TCP/IP protocol cluster suitable for the products.

The performance of Webit 2.0 as follows:

- Without depending on PC system structure;
- Using Atmel AVR RISC processors;
- User-defined Web pages;
- User-defined CGI programs used to control;
- 14 bit I/O interface (TTL level);
- TTL level UART supported 115200 bps;
- 10 M Ethernet interface (RJ-45);
- System programming (ISP);

Overview of Ethernet controller chip RTL8019AS in Webit:

RTL8019AS is a highly integrated Ethernet controller, it can simply answer plug and play NE2000 compatible

adapter, which has two-fold and power decrease characteristics. Through the three-level control characteristic, RTL8019AS is the best ideal choice for network equipment GREEN PC in all already known things. The two-fold function can simulate send and receive the spread between twisted-pair and all two-fold Ethernet switches. This not only can make bandwidth stronger from 10 Mbps to 20 Mbps, but also avoid read multiaccess agreement because of Ethernet channel fight character. Microsoft's plug and play function can alleviate user lower income and focus on the adapter resources, such as the input and output, IRQ, memory address, etc. However, in special application without plug and play function of compatibility, RTL8019AS support s JUMPER and JUMPERLESS options.

In order to provide complete plug and play solution, RTL8019AS integrated IOBASET transceiver, and auto-examination function between AUI and BNC interface. In addition, 8 IRQ BUS and 16 basic addresses BUS provide comfortable environment for large resources situation.

RTL8019AS supports 16 k, 32 k and 64 k bytes memory BROM and the flash memory interface. It provides the page model function, which can only support 4 M bytes BROM under 16 k bytes of memory system space. BROM's useless commands are used to release BROM memory space. RTL8019AS designed the singlechip with the 16 k bytes SRAM, so that not only provides more friendly function, but also saves SRAM storage resources.

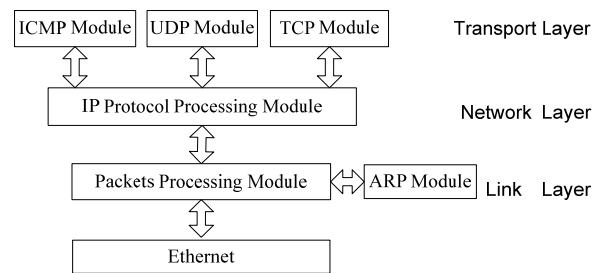
### 3. The design of Webit Reduced TCP/IP Protocol Stack

In the AVR singlechip, due to the relatively limited resources, the TCP/IP protocol cluster of complete function cannot be achieved. So according to the characteristics of AVR singlechip, we cutting the original TCP/IP protocol cluster obtains the reduced TCP/IP protocol cluster.

Meanwhile, based on the architecture of TCP/IP protocol stack, we adapted the design method of network slice model. The architecture of TCP/IP protocol cluster after simplify contains the ARP, IP, ICMP, UDP, TCP protocol processing model, etc. [6-8]. Each layer of the architecture of TCP/IP protocol stack is designed as a module of independent function, handles their data. Different modules can be invoked by function to turn over datum to upper or lower processing module [9]. **Figure 1** shows the simplified TCP/IP protocol architecture.

Known by **Figure 1**, when the AVR singlechip receives data from network, data packet processing modules will base on certain condition to choose the ARP module of link layer or the IP protocol module of network layer to process.

Likewise, when the data packets are processed by UDP and TCP protocol modules of transport layer, it will turn over processed packets to IP protocol processing mo-

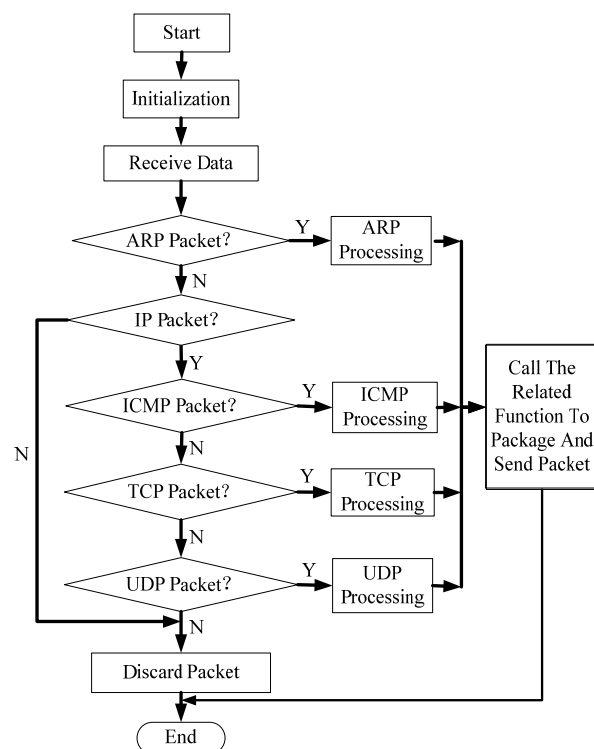


**Figure 1.** Simplified TCP/IP protocol architecture

dule, and make encapsulation for IP first address, the first (such as the fields like address, type of agreement, etc.) by corresponding function calls. Then transfer the datagram including IP first and TCP first to the lower layer by function call, until the data is sent smoothly. The TCP/IP protocol processing is shown in **Figure 2**.

### 4. The Design and Implement of Simplified Embedded TCP Protocol

First, in Webit, we format and size of the MAC and IP address, system address configuration, and size of the buffer were defined already. We make the address format of MAC, IP to become fixed value in system. The system configuration is used for setting specific value of the IP address, the port and MAC address. In this system, for the limited data needed the singlechip to process, so we don't set the buffer larger than normal.



**Figure 2.** The TCP/IP protocol processing

```
.DSEG
.ORG 0x60
LocalMAC: .BYTE 6
LocalIP: .BYTE 4
LocalPort: .BYTE 2
RemoteMAC: .BYTE 6
RemoteIP: .BYTE 4
RemotePort: .BYTE 2
Plugdelaytime: .BYTE 32
TCPCB: .byte 30*2
RevBuffer: .BYTE 260
```

### 4.1 The Implement of ARP Protocol

Because the embedded singlechip is normally in the service of the passive state. So while we design and implement the ARP protocol, we don't implement the function of address mapping table, neither realize the function of querying any client mapped IP into the MAC address, only need to achieve when other client to query the local Mac address. Packet and get feedback of the relationship between own IP and MAC address, and send.

When the embedded singlechip receives ARP packets from Ethernet, we according to the type of operation codes of the packets decide type of ARP packet, if the ARP request packet, compare destination IP address field of ARPP packet with the local settings of IP address. If it's equal, local MAC address packaging to responded ARP reply packet, if not, don't do processing, discard it. The processing flow of ARP packet is shown in **Figure 3**.

### 4.2 The Implement of IP Protocol

The IP protocol is the core of the TCP/IP protocol cluster. All the ICMP, UDP and TCP data transmit as IP datagram

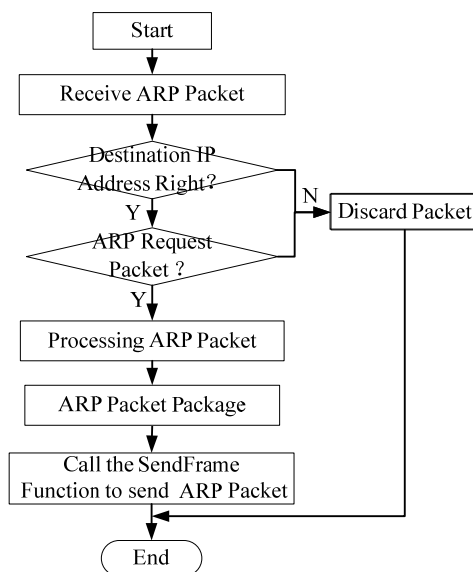


Figure 3. The ARP protocol processing

format. In the IP protocol processing modules, While implement the IP protocol module, we firstly received the IP date packet from the Ethernet and decide whether the destination IP address field values in the head of datagram equals local IP address, if not, discard; if consistent, check such field as the version number and checksum of the IP datagram, etc.

After examination, confirm the packet is right, and then decide to choose ICMP protocol, UDP protocol or TCP to submit to upper processing, according to the type of IP data. In addition, another function of IP protocol module we designed and implemented is to make the message encapsulation delivered from upper into IP data, then turn over IP data encapsulation to link layer to make data frame encapsulation and sending. The processing flow of IP protocol is shown in **Figure 4**.

### 4.3 The Implement of ICMP Protocol

ICMP protocol is a kind of information transfer control protocol. We think about the embedded singlechip as a server is responded the client commonly, as a passive device, it does not need to initiatively send back the message. So we only implement the receiving and handling the Echo Request between singlechip and other devices in the ICMP protocol module, and also send the Echo Reply. The implementation of ICMP protocol is as follows: read type code of the first byte of ICMP data packets, and check the ICMP packet types. If the type code is 8, the type of packets will be modified to 0, fill each field of packets to make the encapsulation of Echo Reply packets needed to be sent back, finally calls SendIP function, make ICMP data packets into IP datagram encapsulation to send. If the packets' type code isn't 8, discard the packet. The processing flow of ICMP protocol is shown in **Figure 5**.

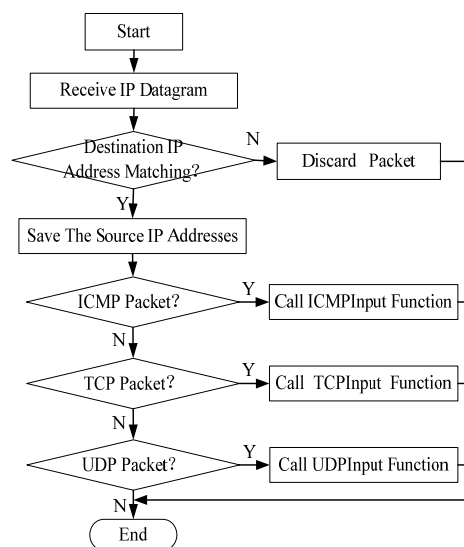


Figure 4. The IP protocol processing

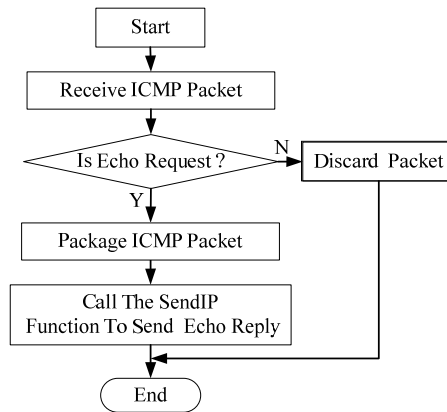


Figure 5. The ICMP protocol processing

#### 4.4 The Implement of UDP Protocol

UDP protocol provides reliable, connectionless communication between applications, it transmits datum to the IP layer and sends out, but does not guarantee they can reach the destination.

When the UDP protocol modules receive packets, first locate the port fields of UDP packets, save the remote and purpose port of the UDP packets, then the compare the objective port of the packet with the port of local regulations, if not equal, discard it; if equal, call the corresponding function. Finally, set the source port, objective port, data length, checksum field in the header of UDP packet, add datum to be sent, make encapsulation and sending by IP layer. The processing flow of UDP protocol is shown in Figure 6.

#### 4.5 The Implementation of TCP Protocol

Due to the limited resources of singlechip, and to handle TCP packet better, so while implementing the TCP protocol module, we reduced the common TCP/IP protocol, and did not implement the sliding window protocol, flow control and congestion control mechanisms. Meanwhile, we set two TCP connection control block in the TCP protocol modules, and adopt the response mode with single window. When receiving TCP packets, first locate mark field of TCP packets, if the TCP packet is required to build a new connection, check whether still exist spare TCP connection control block in the system. If present, this spare control block will be used as the control block for this connection, and establish connections. Conversely, if there is no spare TCP connection control block, and do nothing.

When the mark field of the TCP packets is another type, search whether exist TCP connection control block corresponding to the TCP packets. If present, judge according to the mark field of the value of SYN, FIN, ACK and so on, then choose corresponding function to process packets. If don't exist TCP connection control block corresponding to the TCP packets, don't do anything. Cons-

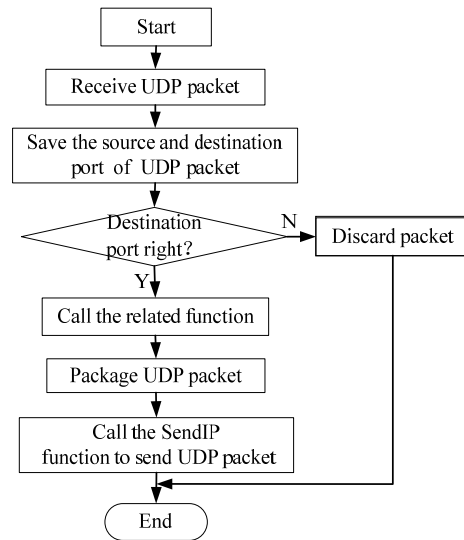


Figure 6. The UDP protocol processing

truction and closing the connection of the TCP protocol are through the “three handshakes” and “four times specific wave”. Setting the mark field in the TCP packets to different control bits is the specific approach. The processing flow of TCP protocol is shown in Figure 7.

#### 5. Testing

In order to test whether the TCP/IP protocol realized can achieve the desired objective, we carried on a series of tests.

The Ping command is the most frequently used in net-

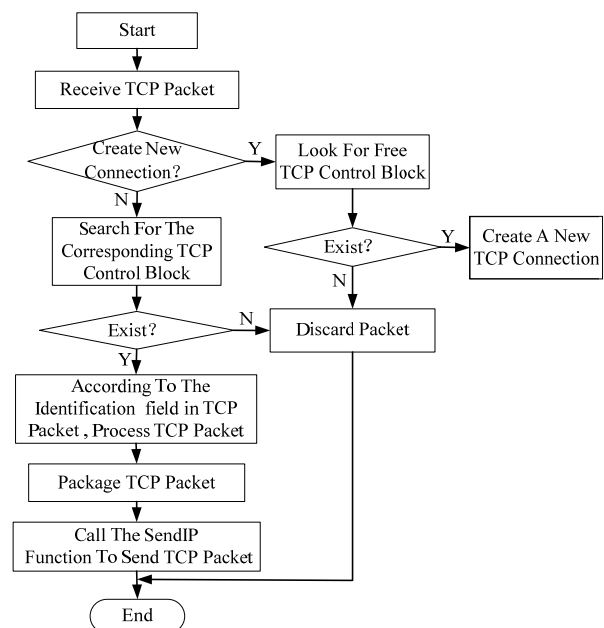


Figure 7. The TCP protocol processing

work. This command sends a network message and requests response via ICMP protocol. Therefore, through the Ping command we can determine whether the current network is connected correctly, and test whether the condition of network connections is available. So, for the testing of ARP, IP, and ICMP protocol, we can finish the testing through Ping command. The process of test is: first, connect Webit and PC, do the network configuration through WebitNetIfConfig function, make configuration of suitable IP address (in this test set IP of Webit as 192.168.180.94). Finally, input "ping 192.168.180.94" on the PC. The running test of **Figure 8** shows that the network is connected, network equipment is available. It states:

- 1) ARP module is normal, and can properly achieve the address mapping;
- 2) The IP protocol modules work normally, and can correctly analyze that this is an ICMP messages;
- 3) ICMP protocol modules work normally, and can correctly return responses message.

The response condition of ICMP packet is listed in the **Table 1**. The test is mainly by sending ping date packet to singlechip to verify the success rate of date sending. As the test date of the table shows, as the request packet sending to the singlechip is smaller, so the request can be effectively handled by server.

For the testing of TCP protocol, we can choose to write a simple Telnet-Server program based on TCP/IP protocol. Users can access on PC, according to system cue, users can input some simple commands to obtain relevant information. Testing method is: input "telnet 192.168.180.94" on PC. The result is: the system shows that the connection is established successfully, and the associated tip. Input something according to cue, and then obtain appropriate information. Finally, the test shows that TCP protocol modules in the TCP/IP protocol cluster are correct, and the network layer protocol is correct.

In addition, we also test the established time of TCP

```
C:\Documents and Settings\Administrator > ping 192.168.180.94 -t
Pinging 192.168.180.94 with 32 bytes of data:

Reply from 192.168.180.94: bytes = 32    time < 1ms  TTL=128
Reply from 192.168.180.94: bytes = 32    time < 1ms  TTL=128
Reply from 192.168.180.94: bytes = 32    time < 1ms  TTL=128
Reply from 192.168.180.94: bytes = 32    time < 1ms  TTL=128
Reply from 192.168.180.94: bytes = 32    time < 1ms  TTL=128
Reply from 192.168.180.94: bytes = 32    time < 1ms  TTL=128
```

**Figure 8. The test results of ARP, ICMP and IP protocol**

**Table 1. Response of ICMP packet**

Response of ICMP	Size	Request packets	Reply packets	Success rate
	128 Byte	50	50	100%

**Table 2. The established time of TCP**

The established time of TCP	Meanvalue	Standard deviation	Maximum
	2.314 /ms	0.53 /ms	2.437 /ms

many times.

As the test results listed in the **Table 2**, the average established time of TCP is about 2.314 ms, it means that server can quickly respond according to external request. In the process of test, while singlechip server accepting the connection request, it has well reliability. Mean while, it means after implementing the reduced TCP/IP protocol on the AVR singlechip, it can well meet the requirement of non-PC devices be connected to Internet.

For the testing of UDP protocol, we write a program according to the realized UDP protocol that creates two UDP Sockets, one is to achieve the function of sending UDP data, and the other is used to achieve the function of receiving UDP data, in order to test the correctness of the UDP protocol. The procedure of testing is:

- 1) Initialize the equipment;
- 2) Create the UDP Sockets (S1 and S2) through WebitUdpCreateSocket;
- 3) S1 sends UDP data to S2 through the function of WebitUdpSendTo, S2 receives data form S1 through the function of WebitUdpReceiveFrom, and output the received data to the serial port;
- 4) Close the two created Sockets (S1 and S2) through WebitUdpDestroySocket command.

Finally, the testing results show that S2 received data form S1 successfully; this means the UDP protocol of TCP/IP protocol cluster is correct.

## 6. Conclusions

This paper expounds the implement principle, method and technology of the TCP/IP protocol on WEBIT platform. In implement of the TCP/IP protocol we bring in the programming ideas of network layer and reduced TCP/IP system structure which adapt to the characteristics of singlechip [10]. At the same time, the paper also make certain explore and try in embedded singlechip and network applications finally, the feasibility of this protocol is confirmed in simulation experiment system, so it's significant and full of reference value for embedded Internet system design and development.

## REFERENCES

- [1] G. Y. Xu, Y. C. Shi and W. K. Xie, "Pervasive Computing," *Computer Journal*, Vol. 26, No. 9, 2003, pp. 1042-1052.
- [2] D. A. Gregory and E. D. Mynatt, "Charting Past, Present and Future Research on Ubiquitous Computing," *ACM Transaction on Computer-Human Interaction*, Vol. 7, No. 1, 2002, pp. 29-58.

- [3] H. Zhao and Y. Chen, "Pervasive Computing," Northeastern University Press, Shenyang, 2005.
- [4] H. Zhao, "Embedded Internet," Tsinghua University Press, Beijing, 2002.
- [5] T. Kindberg and A. Fox, "System Software for Ubiquitous Computing," *IEEE Pervasive Computing*, Vol. 1, No. 1, 2002, pp. 70-81.
- [6] V. Jonathan and P. Joseph, "Profiling and Reducing Processing Overheads in TCP/IP," *IEEE/ACM Transactions on Networking (TON)*, Vol. 4, No. 6, 1996, pp. 817-828.
- [7] H. Jang, S.-H. Chung and D.-H. Yoo, "Design and Implementation of a Protocol Offload Engine for TCP/IP and Remote Direct Memory Access Based on Hardware/Software Coprocessing," *Microprocessors & Microsystems*, Vol. 33, No. 5-6, 2009, pp. 333-342.
- [8] J. F. D. Rezende, M. M. D. A. E. Lima, N. L. S. D. Fonseca, "Mobility over Transport Control Protocol/Internet Protocol (TCP/IP)," In: M. Llyas Ed., *The Handbook of Ad Hoc Wireless Networks*, CRC Press, Boca Raton, 2003, pp. 329-343.
- [9] A. Dunkels, "Full TCP/IP for 8-bit Architectures," *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, San Francisco, May 2003, pp. 85-98.
- [10] S. Ichiro, "Location-Aware Communication in Smart Spaces," *Proceedings of 2007 International Conference on Multimedia and Ubiquitous Engineering (MUE 2007)*, Seoul, 26-28 April 2007, pp. 1027-1034.