Design and Implementation of Embedded Web Server Based on ARM

Yilan Dong
Northwestern Polytechnical University, Xi'an 710129, China

Abstract. This article selects the STM32F103C8T6 microcontroller based on ARM Cortex-M3 core and ENC28J60 with Ethernet module as the main hardware structure, and then accesses some peripheral hardware devices on STM32 microcontroller. Then introduce embedded technology and network interconnection technology, transplant UIP protocol stack to STM32 MCU, and transmit related information and data based on HTTP protocol to realize embedded Web server.

1. Introduction
With the rapid development of science and technology, embedded systems are particularly important in the computer field. Embedded systems have the tailorability of hardware and software performance. [1] In addition, embedded systems have a streamlined system with very high timeliness and very low power consumption. Because HTML has the characteristics of language standard uniformity, as long as the network server is embedded in the embedded device, the functions of the client browser to send and receive information and control peripheral devices can be realized. [2-5] According to the network interconnection technology and network communication basic knowledge of ARM embedded system, this paper makes a reasonable analysis and simplification of the TCP/IP protocol used by embedded devices. Finally, the simplified UIP protocol stack in TCP/IP protocol is selected. Implemented an embedded web server.

2. Overall system design
In addition to the need to implement a Web server, this article also requires the client to control the switch of the appliance and the analog temperature of the button through the browser interface. After considering a series of factors such as function, cost, volume and power consumption of each module, try to satisfy its function and reduce its waste. [6] Design the following overall block diagram. Mainly with the single-chip computer and Ethernet network module as the main hardware structure, constitutes the embedded Web server, through the client interface to control the temperature update and electrical switch connected to the microcontroller. The overall block diagram of the system is shown in Figure 1.
3. Embedded UIP protocol stack data processing flow

In the process of executing the UIP protocol stack data processing, when the data packets in the network are detected by the network card of the system, it is quickly determined whether the network addresses are consistent with each other. If it is inconsistent, it will discard its data packet. Otherwise, it will continue to execute the next step to determine whether it is a request for ARP packet. If it is, it will send an ARP reply packet to the client, otherwise it will be judged as an IP packet. The IP packet will go through a series of necessary processing to determine whether it is an Internet Control Protocol request packet. [7] If it is, it will send its response packet. Otherwise, it will continue to judge whether the packet is a TCP packet and whether the IP is correct. If yes, that is, the packet is received, and if not, the packet is discarded. The data processing flow chart is shown in Figure 2.

![Data processing flow chart](image_url)
4. Implementation of the HTTP protocol

The role of the HTTP Hypertext Transfer Protocol is to build a distributed information system and then store the data on the WWW World Wide Web, which is distributed across the application layer above the TCP/IP protocol.

At present, with the development of science and technology, the version of the HTTP protocol has grown to version 2.0. In the current HTTP 2.0 version, the HTTP protocol has strong parallel processing capabilities. When transmitting data with TCP, it is utilized port 80. When implementing the HTTP protocol, it mainly consists of two aspects of HTTP request and response.

In an embedded system, a user's access to a web page file requires a series of processing by the web server. Once the web server passively receives the request from the user, it will immediately determine whether the request is reasonable. Then analyze the content contained in this message information, receive relevant information data, and finally provide the user web page. Although the message system has content in addition to the key segment, it is mainly to modify the key segment to a large extent. Because the user needs to receive the document from the server side and also store the document in the server, it is necessary to focus on the request and response processing of the GET and POST methods. Therefore, support for other aspects may not provide support for the response.

a). GET request

When a user needs a resource on the network, the user enters their URL to access, which is also a GET request. The definition of the GET request, that is, the page information required by the user request, is then immediately returned to the entity body. The local definition of the documents required by the user is in the URL. Under normal circumstances, the server generally places the contents of the document in the body of the response message when responding.

b). POST request

The response message appears in many forms, either as a status line, as a header, or as a subject. The state of the defined message and the state of the response are implemented by the status line and the status code field, respectively. Different series of codes represent different functions and functions. The meaning of the five codes in the 100 series to the 500 series means informal, successful request, customer redirection in the URL, an error in the client, and an error in the server. The implementation process is shown in Figure 3.

![Figure 3. HTTP protocol receiving processing flow chart](#)

Adding the field ASCII code in the URL is the calculation method of the URL. The URL has an identification function, which is based on the spaces in the two segments of the URL. In the POST method process, the entity part in the request information is the control data uploaded by the client browser. The identification character of the request message is two consecutive characters, which are CRLF (`\r\n`). If the control data needs to be identified, then the variable name can be used to identify the value. In the data frame transmission process under the hypertext transfer protocol, the content of the web page is segmented, because the length of the TCP message field is not unlimited.

Design and implementation of embedded WEB server

The embedded Web server implemented in this article has three basic functions.
a). Real-time display of data. Specifically, the server can provide real-time status of each device to
the webpage in a timely manner in a dynamic display manner, and the client can monitor and control
the state of its peripheral device through the Internet at any time.
b). Parameter control of equipment operating parameters and status parameters. The writing
operation of the parameter is performed when a parameter device request sent by the user is received.
c). remote control. Even if the user is not at the side of the device, the user can give instructions to
the drive system through the Internet at any time to control the operation of these remote peripherals,
thereby achieving the goal of the user remotely controlling the device.

Therefore, the embedded Web server based on system application design and implementation
solves the difference between hardware and operating system, and enables embedded Web server to be
applied on various platforms.

5. Web page storage and web page data
As long as the traditional Web server is tailored to some system structure, and the information
collection and information publishing functions of some measurement and control devices are added,
the current embedded Web server system is realized. The storage capacity of the embedded WEB
server is much smaller than that of the server on the PC, and the process of converting the web server
to the browser interface is dynamically generated, so that the web server cannot store the HTML
webpage in a large amount. In the server designed in this article, the web interface and program code
are placed in the 64K FLASH.

The format and content of the html file header are as follows:
HTTP/1.1 200 OK\nCache-control: no-cache\nConnection: Keep-Alive\nContent-Length: 775\nContent-Type: text/html\n
The web page file is a web page interface formed by editing with html editing software, and then
the page code is separately stored in the buffer array.

6. Processing and implementation of dynamic content
In the embedded Web server designed in this paper, in remote monitoring, the server is used as a
hardware device to monitor dynamically changing content. We can apply CGI (Common Network
Management Interface) technology to embedded protocols to implement dynamic web pages.

CGI technology is not only a program that runs on the server, but also an interface that can be
provided to the client's hypertext markup language page. Features that are not implemented in HTML,
such as not being able to record customer information and not being able to store special file
information. At this time, CGI programs are especially important. It not only can record this
information, but also can use the variables and parameters as well as the input and output data to
communicate with the WWW server in real time, to transfer the relevant parameters and processing
results.

CGI is a supplement to HTML, and its use is very necessary. CGI is a standardized interface that
provides timely service to the data entered by the client through the browser. If the client makes a
request to the server through the browser interface, the server will immediately implement a matching
CGI program. Transforming customer needs into a form that the server can recognize is also a feature
of CGI technology. After a series of processing, CGI presents the relevant data processed by the server
to the client, which is identifiable by the browser.

When the server obtains the information of the client browser, the data form is:
nname [1] =value [1]
nname [2] =value [2]
nname [3] =value [4]
name[nl=value[n],

The left side of the equation represents the name of the variable, which means the name of the input field in the form. The right side of the equation is the value of the variable, which means the value entered by one of the input fields entered by the user. When the client transmits data, each line of the equation string needs to be separated by ";". The type of data encoding is achieved through environment variables.

In this article, after the Web server modifies the variables in the HTML form in the browser, it can control the switch and temperature of the remote appliance.

7. Conclusion
This paper completes the implementation of the application layer HTTP protocol. On this basis, the design of the embedded Web server is completed, and the test runs successfully. Through the web browser, it is indeed possible to remotely log in to the web server and control the peripheral devices on the server side, realizing the dynamic interaction of the network. The server has the characteristics of high transmission rate, high reliability, easy access, and the like, and has broad application prospects.

References
[1] Daniel Happ, Niels Karowski, Thomas Menzel. Meeting IoT platform requirements with open pub/sub solutions.[J].Annals of Telecommunications, 2017, Vol.72 (1-2), pp.41-52
[2] Yong Jin, Ruigang Li, Huan Dai. QoS guarantee protocol based on combination of opportunistic dynamic cloud service and cooperative multimedia stream for wireless sensor networks.[J].EURASIP Journal on Wireless Communications and Networking, 2015, Vol.2015 (1), pp.1-13
[3] Lance Stout, Matthew Walker, Jérôme Laurent. Using Kestrel and XMPP to Support the STAR Experiment in the Cloud.[J].Journal of Grid Computing, 2013, Vol.11 (2), pp.249-264
[4] Ronny Klauck, Michael Kirsch. Combining Mobile XMPP Entities and Cloud Services for Collaborative Post-Disaster Management in Hybrid Network Environments.[J].Mobile Networks and Applications, 2013, Vol.18 (2), pp.253-270
[5] Komkrit Choouruang, Pongpat Mangkalakeeree. Wireless Heart Rate Monitoring System Using MQTT.[J].Procedia Computer Science, 2016, Vol.86
[6] Jorge E. Luzuriaga, Miguel Perez, Pablo Boronat. Improving MQTT Data Delivery in Mobile Scenarios: Results from a Realistic Testbed.[J].Mobile Information Systems, 2016, Vol.2016
[7] Hyun Cheon Hwang, JiSu Park, Jin Gon Shon. Design and Implementation of a Reliable Message Transmission System Based on MQTT Protocol in IoT.[J].Wireless Personal Communications, 2016, Vol.91 (4), pp.1765-1777