Reconfigurable Embedded Web Server. : towards a flexible Internet appliance.

Krerk Piromsopa, Boonchai Sowanwanichakul, Prabhas Chongstitvatana.
Department of Computer Engineering
Chulalongkorn University
254 Phayathai Road Patumwan,
Bangkok Thailand 10330.
Tel. (662)-218-6956, Fax: (662)-218-6955
g41kpr@cp.eng.chula.ac.th, boonchai@cp.eng.chula.ac.th, prabhas@chula.ac.th

Abstract

This paper focuses on how to develop an embedded web server that can be reconfigure or update the control software with scripting language through the web network to easily meet any control environment. This paper covers the design and implementation of the embedded web server with a small microcontroller, the 8-bit MCS-51 with DP83902 Ethernet Network Interface Controller. The software developments comprise a subset of TCP/IP network protocol, small web server, the basic authentication system and the extra-ordinary PHP style scripting language interpreter for inquiry and control of the system.

Introduction

Embedded Web Server can hardly update or reconfigure the software function while it's running on the network. Each developer usually uses his proprietary method to update or reconfigure the system. Most of this method cannot be done through the network. As a result, the same embedded web server cannot be reprogrammed to meet various styles of functions while it's running. Moreover, since the system is open for anyone to view and monitor, the system must be protected with some security.

Traditionally, embedded devices or appliances connect to a computer system through dedicated serial port. As a result, the more devices connected, the more serial I/O interfaces required. In order to support a graphical user interface, the developers have to write a specific user interface program; otherwise the data would be dumped out as a straight text. By using the embedded web server, the graphical user interface can be easily implemented. Further more, a large number of devices can be used with the power of web network.

This paper chooses MCS-51, the popular Intel 8-bit microcontroller, for implementing an embedded device as this microcontroller is widely use in the control industry and appliances. The slow 8-bit microcontroller is powerful enough to control a small appliance. But the core architecture is not powerful enough for perform CSMA-CD encoding and decoding at 10 Mbps. A network interface controller is required. The systems needs a dedicate NIC adapter for connecting to the network

While each vendor tries to develop his web server for embedded into his own device. [1] has presented the idea of an Open Source embedded web servers to be used as a standard. Once the Open Source embedded web server is used, the developers can modify and recompile the source code to make it functioning with the preferred hardware. This is a strong drive for creating a reconfigurable embedded web server that functions as a component. Each developer that want to connect his device to the web, just uses this embedded web server as a plug-in module and write a server-side script for each function.

To gain an ability of reconfigure the function in any environment, the scripting language is involved. Scripting languages have proven particularly adept at integrating applications, where new functionality is layered on top of existing components and resources rather than built from scratch [2]. If a simple embedded web server is built, it can be configured to work with any device that connected through an I/O port of the microcontroller. Because resources were made available to the world, a security system was needed. A security authentication must implemented. A very simple but powerful method is to use the password as an authentication.

This paper considers the role of implementing a reconfigurable embedded web server. It describes the standard and protocol that must be met and how to keep the system more secure with basic web password authentication.

Hardware Design.

An embedded web server is a thin server with all the functions that should be supported in order to perform services with little resources and small memory. The first hardware requirement of the embedded web server function is a communication channel. Next requirement is the file system for storing the web page.

Networking the Embedded Systems is not as easy as networking the Personal Computer. A PC has a commercial network adapter readily attached to the mother's board. The network interfaces adapter usually comes in the standard bus such as ISA or PCI bus. As a result, the LAN adapter of this embedded system must be implemented. To create a LAN adapter, a suitable Network Interface Controller must be chosen. There are several Network Interface Controllers (NIC) from various vendors. Some of them support 100Mbs Ethernet. The NSC DP83902A is selected as it is compatible with NE2000, supports the 8-bit bus width and is easily ported the driver for the MCS-51. However, the bus speed of the MCS-51 is quite slow compared to the 10 Mbps of the Ethernet Network. To transfer the data in and out of the network, a dual DMA system is used. One port is used to transfer the data from Network to the NIC local buffer and the other port to transfer the data from the NIC local buffer to the main memory. This mechanism is shown in figure 1.

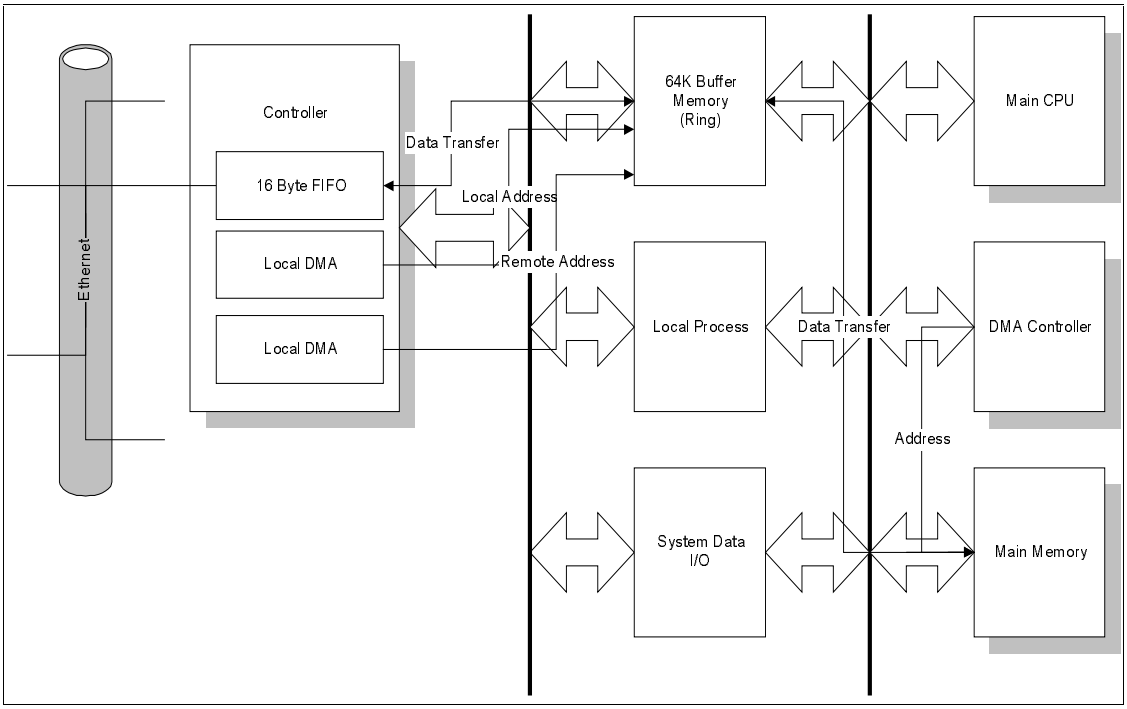

Figure 1 Dual DMA System. [3]

Since there is no secondary storage such as a hard disk file system for storing the web page, RAM must be used as a data memory, program memory, and also acted as a file system for the web server. A common web server usually stores information on the root of the directory structure. It should be easy if this thin server handles the information in the same way. A good solution is emulating the file system using memory. The battery backed RAM is used as an alternative to the more traditional non-volatile memory devices.

Since there are only 24-bit I/O port on the MCS-51 system, the memory has to be mapped as both memory and I/O interface. The overview of how components are connected is shown in figure 2.

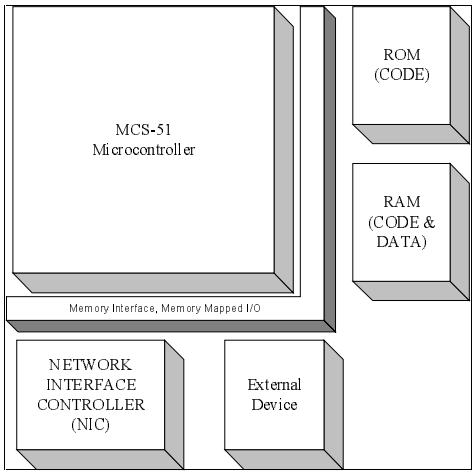

Figure 2 Hardware Block Diagram

Protocol and Standard.

There are several protocols and standards that must be met to make the system functioning as a web server. Some are the network protocol [4], [5], [6]. Others are the server standard. We will take a look at the principal protocol that is required for making an embedded web server.

TCP/IP Protocol suits.

TCP/IP Protocol suit is the common requirement of today Internet network. A full implementation of the TCP/IP protocol is so far too large and needs a lot of memory. Since there are a number of features that can be ignored in the TCP/IP system, for instance, the Silly window syndrome avoidance, the full sliding window recovery and the IP fragment. In this paper a subset of TCP/IP is chosen for easily to simplify the implementation with a small memory footprint.

ARP

The goal of ARP protocol [7] is to map the IP Address with the Ethernet MAC Address. ARP helps the client to find the true MAC address of the server. In this paper ARP is the main mechanism to assign the IP address to the embedded systems. Since the embedded web server will be placed in various networks, the easiest way to assign the IP address to the embedded web server system is let the client assign the IP Address by modified their own ARP table to make the embedded system the IP Address.

HTTP

Hypertext Transfer Protocol is the engine of the web server. The main function of HTTP protocol is to provide the data transfer between the client (Browser) and the server. The HTTP can be divided into 2 parts. The first part is the Header that will carry the request and the description of data in the following part. The second part carries the data, which can be any type of documents, for example, images, a plain text or binary stream. Most of the document type is the HTML. However, HTTP is running as the application service on the top of TCP/IP network. On the server based web server, HTTP will work as an application that transfer the data located in the file system to the network. The embedded web server's view is different. The requirement of the embedded web server is to acquire data from devices as well. We will describe later on how to acquire and setup the device using the scripting language.

Basic Web Authentication

The security systems are the combination of Authentication, Authorization and Accounting [8]. The Authorization and Accounting are different in each application and should be binding at the runtime. The only thing that web server can do for the security is the Authentication.

HTTP protocol has an extend standard that defines the method for the web Authentication. The outstanding one is called basic web authentication method. The basic authentication provides a simple authentication with base-64 encoding. This encode is usually used in the email system. This authentication method is considered suitable for the embedded system since it's secure enough for non critical security system and can be implemented with a few lines of code. The scripting language handles the Authorization and Accounting by passing the parameter as a variable of the program.

Scripting language.

There are several scripting languages existed today. Some of them are proprietary scripts. Among various scripting languages, PHP style script is the most popular one for the server-side web scripting language. Due to the ease of programming, the fast execution and the C language syntax, PHP becomes the choice of many web developers. This paper creates a derivative of the PHP style scripting language. It is called "PHP Lite Script", which has a few standard functions and new functions for controlling I/O devices.

The script language interpreter is the engine of the scripting language. A small interpreter is constructed with a small lexical analyzer and a parser. This PHP Lite Script consisted of 18 tokens and 21 derived grammars. The interpreter use less than 1 KB of memory.

Software Implementation

The system software part of the embedded web server can be divided into 2 parts. First part is the Network Connection. The second part is the Device and I/O Handling. All of the components can be referred as "Embedded Kernel". When a user requests the web page the data will flow thought each software component to the HTTP Server. After the server recognizes the request, the server will process the data, which might invoke the interpreter and the I/O handler routine. When the server finish processing the request, it will generate the output to the Network. See Figure 3 for more details.

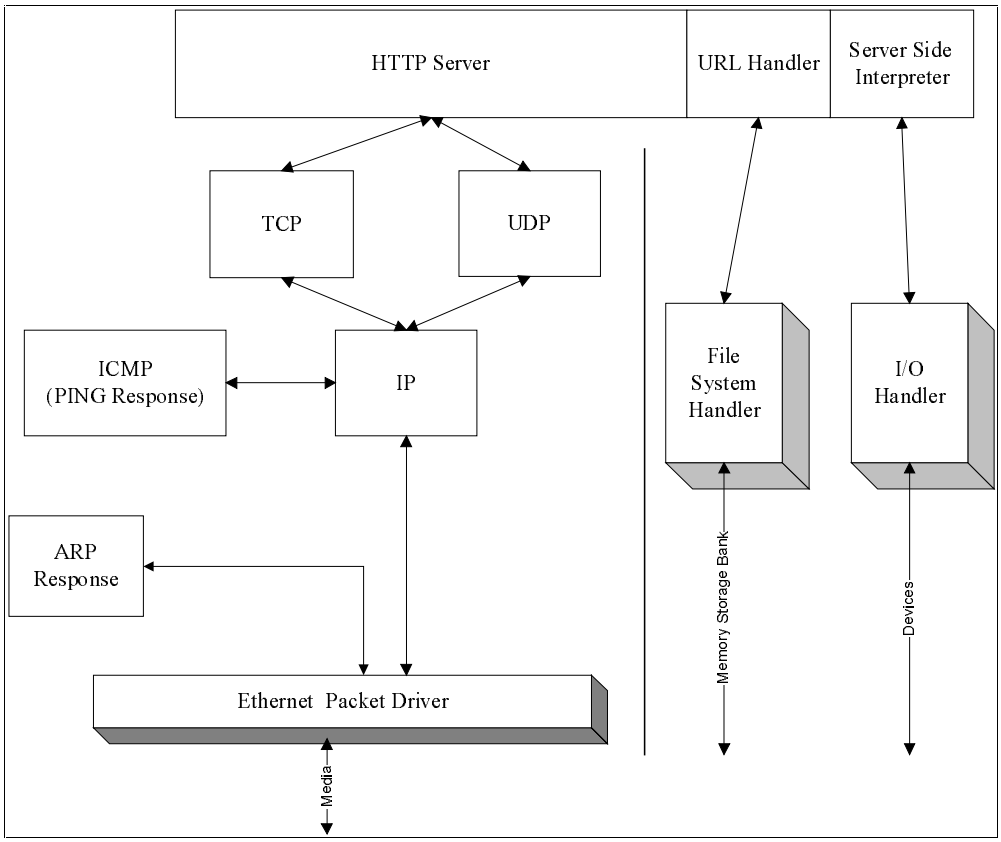

Figure 3 The software component in embedded web server.

Web Programming.

The scripting language can be written using with any text editor or the web editors, such as MS Front Page or Macromedia Dreamweaver, then uploads to the system through the web. (Figure 4). After submitting the script to the server. The server will show the complete page. To run the script, the username and password must be provided.

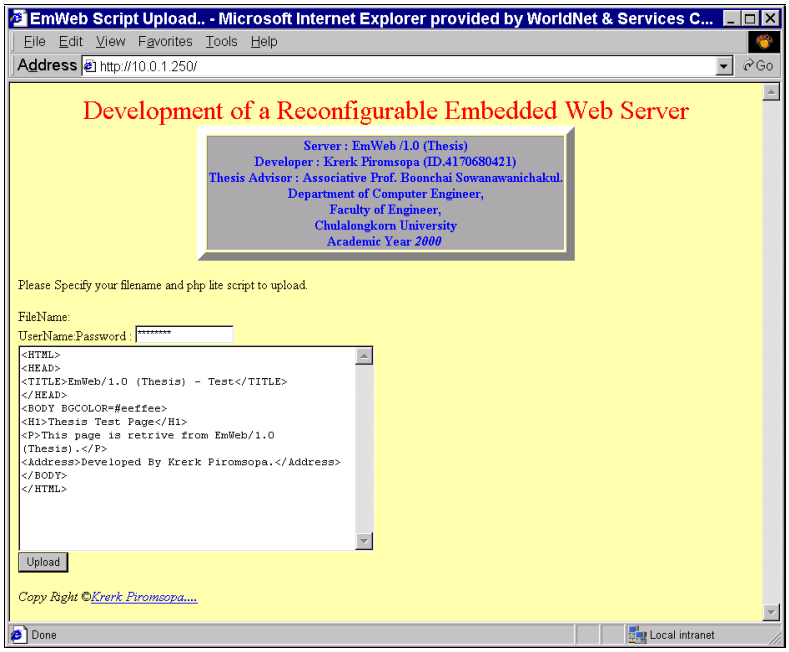

Figure 4 . Web page for uploads the script to the web.

Figure 5 shows a simple script that will receive the data from a form in the web page and outputs the data to the I/O of the system. The output of the Web page is shown in Figure 6.

```
<HTML><HEAD><TITLE>Demo EmWeb/1.0(Thesis)</TITLE></HEAD>
<BODY BGCOLOR=lightyellow>
<H1 ALIGN=CENTER>ทดสอบ/สาธิตการใช้งาน EmWeb/1.0(Thesis)</H1>
<H2><FONT COLOR=BLUE>การตั้งค่า</FONT></H2>
<FORM METHOD=GET>
กรุณาเลือก ค่าของ PORT และ LED..<BR>LED: <INPUT TYPE=TEXT NAME=val VALUE="00" SIZE=4>
<INPUT TYPE=SUBMIT VALUE="ตั้งค่า"></FORM>
<%outp(0x2000,$val);%>
<Address>โดย เกริก ภิรมย์โสภา.</Address>
</BODY></HTML>
```

Figure 5 Sample PHP Lite Script.

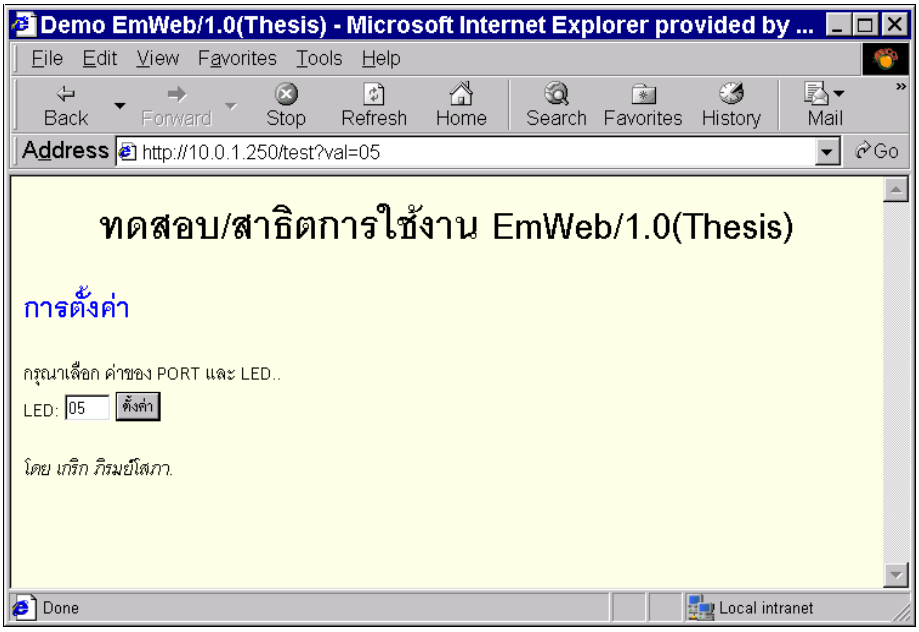

Figure 6 Sample Web page that generated from the embedded web server.

However, the server-side script can be programmed with other client-side script (such as JavaScript) or remote software Applet (Java Applet). For example, you might have JavaScript web page that is more interactive for your control application with the event-based programming. (Figure 7.)

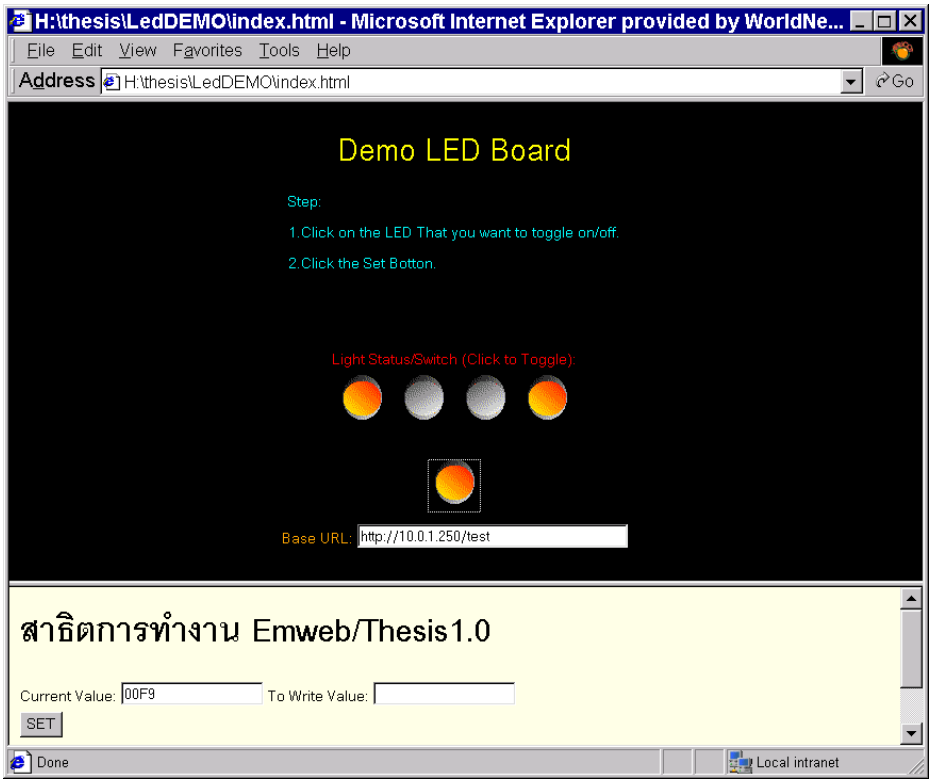

Figure 7 Web page with event-based JavaScript.

Conclusions

Embedded web server is useful in many applications. With the power of web protocol, it is easy to create a graphical user interface and to remotely retrieve the data. In the near future, every devices and appliances will be embedded with web servers to gain the power of information sharing and remote management. Suppose that you are away from your home and want to check as if the door is locked or open the air-conditioned before you are going back, then you just use your mobile web browser to connect to your home and monitor or control them as you want.

The development of the Embedded Web Server is the gate to open any device or appliance to the Internet network. The power of the reconfigurable Embedded Web Server is that it can be configured to work with any device and any environment from anywhere. Many of problems in implementation are from the limitation of the microcontroller. Presently, there are no standards for embedded web server. This work proposed a new method of creating embedded web server. This technology will create a significant impact in the future because it allows various devices to be managed and controlled from anywhere in the world.

References

[1]. O'Brien M. Open Source Embedded Web Servers. GoAhead Software. : Embedded Systems Conference, 1999.

[2]. Ousterhout J. How Scripting Adds Value to Embedded Systems. Scriptics Corporation: Embedded Systems Conference, 1999.

[3]. National Semiconductor. DP83902A ST-NIC. PRELIMINARY November 1995.

[4]. Postel, J. Internet Protocol. RFC791 September 1981a.

[5]. Postel, J. Internet Control Message Protocol. RFC792 September 1981b.

[6]. Postel, J. Transmission Control Protocol. RFC793 September 1981c.

[7]. Plummer, D. An Ethernet Address Resolution Protocol. RFC826 November, 1982.

[8]. Franks J. HTTP Authentication Basic and Digest Access Authentication. RFC2617 June, 1999.