EMBEDDED WEB SERVER

SARIKA CHHATWANI
Department of E & TC, Amravati University, SSGMCE,
Shegaon, Maharashtra, India

Dr. K.B. KHANCHANDANI
HOD, Department of E & TC, SSGMCE,
Shegaon, Maharashtra, India

Abstract:
The embedded web server technology is the combination of embedded device and Internet technology, which provides a flexible remote device monitoring and management function based on Internet browser and it has become an advanced development trend of embedded technology. Through this embedded web server user can access their equipments remotely. The equipment mentioned here could be home appliances and factory devices. This paper is focused on realization of TCP/IP suite and user development platform for this embedded web server. The embedded web server design includes a complete web server with TCP/IP support and Ethernet interface. It also includes support for sending mail, and software for automatic configuration of the web server in the network. The web server reference design includes complete source code written in C-language.

Keywords: TCP/IP protocol suite; LPC2124 Kit.

1. Introduction
Of all the semiconductor industries, the Embedded systems marketplace is the most conservative. Engineering decisions in this market are usually conservative, leaning towards established, low-risk solutions. Because of this, the basic infrastructure of embedded systems has only evolved slowly over the past ten years. Intelligent homes will be connected to the Internet and requires a microcontroller to communicate with the other network devices. Recently, ubiquitous computing technology and home network technology have developed rapidly and are in practical use. Therefore, more and more connectivity to the Internet will be required for home appliance such as lighting fixtures, refrigerators and audio-visual equipment. When these appliances are fitted with servers which are connectable to the WWW, interactive communication will become possible and new functions and services will become available. The embedded web server can simplify the design process for embedded web server applications. There has been exponential growth of Internet use in recent years. This has generated a strong trend toward using Internet protocols.

2. System Description
The embedded web server reference design is designed for integration in digital equipment. The embedded web server can be plugged into any Ethernet interface and communicate with a standard web browser. Figure 1 illustrates some situations where the web server can be used.

As shown in Figure 1, various consumer electronics can be controlled from a computer Connected to the Internet. The web page is the “Control Center” for the embedded web server. Suppose the embedded web server is embedded in several units in a house. Every server is connected to the network. A computer located at home as on Figure controls all devices and can receive requests from other computers on the Internet. The web server is identified by its unique IP address and can be controlled remotely from anywhere in the world as long as the authorization is in order.
3. Hardware Structure

High performance chip was produced abundantly every year. The price is acceptable. The prime consideration is the size and system cost. Many embedded computers are physically located within some large artifact. The embedded web server reference design is designed to be flexible for future development. The web server primarily communicates through an Ethernet connection. But it is also able to communicate with SLIP and modem connection using the built-in UART. A CPLD is included in the reference design to allow memory mapping of other devices to the system. With only SRAM and the Ethernet controller connected to the data bus the CPLD can be omitted. The hardware design is flexible, making it usable for many applications. It is designed to be used with Ethernet, SLIP or PPP connections, either through a LAN, another computer or through a dial-up connection. It is also possible to connect other external peripherals to the system. The on-board Flash memory can be expanded without hardware changes.

3.1 Memory

The web server includes enough memory to develop large applications on top of the web server protocols. 32K bytes of external SRAM is used for buffering data. A 2-Mbit External Data Flash is used for storing web pages to allow a large amount of pages to be stored. The SRAM is connected to the address bus and data bus. The Serial Peripheral Interface (SPI) is used for communication with the Data Flash.
3.2 Ethernet-Controller

The Ethernet controller was originally a 16-bit ISA device, but can also be controlled in 8-bit mode. The Ethernet controller is configured as an 8-bit device. The Ethernet controller features 4K bytes of internal memory which is accessed through the I/O registers or directly through memory mapping of the entire memory. Default operation on the Ethernet controller is I/O mode and address 0300h. Since only address lines A0 - A12 are connected (need only 4 K bytes of address space), the I/O registers are mapped to address 8300h - 830Fh (I/O mode when the address lines into the PLD have the following configuration: bit 15 is high and bit 14 is low). By configuring the Ethernet controller through the I/O registers, the address can be changed and memory mode can be enabled. Memory mode operations can be mapped into address locations C000h - D000h.

4. Software

The software running on the embedded web server follows the same layered structure as used in the TCP/IP protocol suite. The TCP/IP protocol suite allows computers of all sizes, running different operating systems, to communicate with each other. It forms the basis for what is called the worldwide Internet; a Wide Area Network (WAN) of several million computers. The TCP/IP protocol suite is a combination of different protocols at various layers. Every layer acts independently from each other. An Ethernet controller driver controls the Ethernet interface. The Address Resolution Protocol (ARP) translates IP addresses to Ethernet MAC addresses (and vice versa) The Internet Protocol (IP) delivers packets to Transmission Control Protocol (TCP), UDP, and Internet Control Message Protocol (ICMP), the ICMP answers to PING requests and TCP/UDP delivers data to the applications. The applications can communicate with the transport layer through buffers with data and variables with control information. This section explains how the TCP/IP protocol suite is built up in our approach.

Fig. 3 Protocol Stack

4.1 Link Layer

Data-link or Network Interface Layer is another common name of this layer. The Link Layer normally includes the device driver in the operating system and the corresponding Network interface (card) in the computer. Together they handle all the hardware details of physically interfacing with the cable. The Ethernet controller is configured to generate an interrupt every time a packet addressed directly to the Ethernet address arrives or when a broadcast arrives. When an interrupt occurs, the microcontroller reads the whole Ethernet frame into memory. A buffer of 1514 bytes, which is the maximum frame size on Ethernet, is reserved for this frame. Once the frame is transferred to the microcontroller, the Ethernet header is checked in order to ensure not receiving a misplaced frame. If the Ethernet address seen by the receiver is either a broadcast (all binary 1’s) or addressed directly to specific Ethernet device, the frame is sent to the next layer or protocol according to the field, protocol type, in the Ethernet header.

4.2 Network Layer

This layer is sometimes called the Internet Layer. It handles the movements of packets around the network. Routing of packets, for example, takes place here. IP (Internet Protocol) and ICMP (Internet Control Message Protocol) provides the Network Layer in the TCP/IP Protocol Suite. The network layer controls the communication between hosts on the Ethernet. There is no form of transmission control to ensure that IP datagram’s arrive to the host or that all IP datagram’s from another host is received. This makes the layer rather
easy to make. The ICMP sends messages between hosts and is only used to answer PING requests from a host. The IP handles communication for the overlaying Transport Layer.

4.3 Transport Layer

On the transport layer there are two major protocols which offer two different kinds of service; TCP which is a reliable delivery service and UDP which offers an unreliable service. TCP also offers flow control for retransmission of segments and acknowledgement of received segments.

4.4 Application Layer

The Application layer handles the details of a particular application. Several applications may be implemented in the embedded web server. The main limitation is memory usage and performance. Running several applications at once means lower performance.

5. Protocol Dependencies

Figure shows the protocol dependencies for the modules in the Embedded Web server. The protocols are described below:

- HTTPD needs TCP and a file-system/Data Flash to operate.
- FTPD needs TCP and a file-system/Data Flash to operate.
- SMTP needs a running TCP implementation to operate.
- DHCP needs IP and UDP to operate. In the initialization phase DHCP requires that the IP and UDP protocol forward any IP packets delivered before the IP address is configured.

![Protocol Dependencies Diagram]

- TCP needs IP to operate.
- UDP needs IP to operate.
- ICMP needs IP to operate.
- IP needs ETHERNET to operate.
- ARP needs ETHERNET to operate.
- ETHERNET depends only on the hardware Ethernet controller.
- CONFIG needs file-system/Data Flash to read the configuration file. If the configuration file is unavailable it can return standard values.
- FILE needs DATAFLASH to work.
- DATAFLASH is only dependent on the hardware Data Flash.
- MAIN must initialize Ethernet, Data Flash, and file-system, TCP, DHCP and HTTPD if these protocols are to be used.
- DHCP, FTPD and HTTPD require repeatedly polling to operate.
6. **User Developing Platform**

When an application sends data using TCP, the data is sent down the protocol stack, through each layer, until it is sent as a stream of bits across the network. Each layer adds information to the data by prepending headers and adding trailers to the data it receives.

Some abbreviations:
- **TCP segment**: The unit of data that TCP sends to IP.
- **IP datagram**: The unit of data that IP sends to the network interface.
- **Frame**: The stream of bits that flows across the Ethernet.

IP (Internet Protocol) adds an identifier to the IP header it generates to indicate which layer the data belongs to. IP handles this by storing an 8-bit value in its header called the protocol field. Similarly, many different applications can be using TCP or UDP at any time. The Transport Layer protocol stores an identifier in the header they generate to identify the application. Both TCP and UDP use 16-bit port numbers to identify applications. The TCP and UDP store the source port number and the destination port number in their respective headers. The network interface sends and receives frames on behalf of IP, ARP, and RARP. There must be some form of identification in the Ethernet header indicating which network layer protocol generates the data. To handle this, there is a 16-bit frame type field in the Ethernet header.

![Fig.4 Encapsulation of Data as It Goes Down the Protocol Stack](image)

7. **Case Study-The web server apply to a Air Condition**

The goal of this embedded web server is link to device and home appliances, the small size of allow that it can embedded into any little appliances. We have built several applications using this web server here we present an example of application that allow users to control and monitor the state of air-condition.

Embedded web server connect with air-condition with I2C bus. On another end embedded web server connect Ethernet with RJ45 interface. The homepage of this application is about 28K byte in size; it includes two HTML files, thirteen pictures at jpg format and two Java Applet files. Pictures are used to display the state of air-condition vividly, Java Applet used to communication between user and embedded web server. When user press the button on homepage of this air-condition corresponding command are send to embedded web server, between embedded web server and browser we use UDP protocol. Because UDP is an unreliable protocol so we reliability character is guaranteed at Java Applet. When embedded web server received the command that user send out from remote browser it deal with this command according to the protocol with air-condition. I2C bus used between them. So air-condition got the command that remote user send. When air-condition complete the action then send back a response to embedded web server transport this response to remote user.
8. Conclusion

We have introduced the general design concept of the embedded web server and the policy of TCP/IP reduction, especial the reduction of TCP, whose goal is to allow easy access to and exploitation of remote equipment. This web server gives the common devices an Internet interface and gains a good performance. It can be used broadly in industry, medical, and other fields, more important it can bring us a new home life. With the web server embedded, we will begin to see the application of computing technologies in settings where they are unusual today device and appliance networking in the home; faithful capture of scientific experiments in the laboratory; and automated full-time monitoring of patient.

9. References