Embedded Web Server for Monitoring and Controlling of system Using ARM Processor

SAVITA LAD, Prof. PRAMOD JADHAV, Prof. R. J. VAI DY

Abstract- An embedded web server, in general, controls the use of system resources by running the web server within tightly controlled limits, so that bugs will not compromise the system operations. Assigning multiple functionalities to a single button on an appliance help manufacturers ecomonize user interfaces, but, this can easily create confusion for the users. Since the cost of web-based interfaces is considerably low, they can be used to provide the infrastructure for the design of simple and more user-friendly interfaces for household appliances. Embedded Web servers are widely used today for IP-based element management. This paper is focused on realization of TCP/IP suite and user development platform for this embedded web server. A key goal of the present paper is to provide an effective approach of access to traditional equipments that have no Internet interface and a reduction policy of TCP/IP protocol suite. By taking advantage of modern Web technologies, the proposed architecture provides a method to develop management applications efficiently and to manage network devices effectively. It presents and discusses architectural features, limitations, performance and trends.

Keywords: ARM Processor, Embedded system, Ethernet Controller, TCP/IP protocol.

I. INTRODUCTION

Computer communication systems and especially the Internet are playing a rapidly increasingly important role in our everyday environment. Today this is not only a domain of personal computers or workstations. We are beginning to see the Internet and associated technologies manage our work and home environments through the use of intelligent embedded devices. Using these appliances, security systems, card readers and building controls that can be easily controlled using either knowledge, many applications are imaginable. Home automation, utility meters, special front-end software or a standard Internet browser client from anywhere around the world [1].

After the “everybody-in-the-Internet-wave” now obviously follows the “everything-in-the-Internet-wave”. The most coffee, vending and washing machines are still not available about the worldwide net. However the embedded Internet integration for remote maintenance and diagnostic as well as the so-called M2M communication is growing with a considerable speed rate.

The networks have to become flexible and easily integrated, with the user getting closer to the device without supplementary efforts, using large networks like the Internet. Such device, which consumes a few bytes of memory and is specifically designed for microcontroller-based embedded systems, allows designers to create modular components that can be connected to the Internet and controlled remotely using a standard Web browser. By adding Web server technology, the manufacturer gains an immediate competitive advantage through standardized access, both in terms of protocol and client application. With the explosion of the Internet and Web services, companies that have provided proprietary solutions for networking are rushing to add Internet technologies and embedded Web servers to their product lines. It provides a more open and economical alternative of the networking devices, reduces development costs and increases functionality [1].

Embedded Web Servers:

A typical case of applying Web technology to network management is to embed a Web server into a network device for element management. This type of Web Server is called an Embedded Web Server (EWS). A EWS provides users with a Web-based management interface constructed using HTML, graphics, and other features common to Web browsers. The status of a device is provided to the user by simply retrieving pages, and an operator command is sent back to the device using forms that the user completes. Accessing Web-based management user interfaces through embedded Web server offers many advantages: ubiquity, user-friendliness, low development cost, and high maintainability [3].

Embedded web server is achieved by implementing Ethernet connection to Internet, WAN and LAN. This, such in TCP/IP protocol stacks an
Embedded System. Once the system is able to communicate through TCP/IP protocol, this can be placed in any TCP/IP network Implementation of connecting Embedded Systems as Internet and LAN. And the device can be programmed to desired functionality. Typical functions are Web Server, E-mail client, FTP server, POP3 etc.

Therefore, this paper describes a new application of Embedded Web Server. Various devices can connect & control over internet by this Embedded Web Server. Our purpose is design an embedded web server and embedded it to kinds of devices so the different devices that can give user a uniform interface to access through a browser. In fact they only need a browser. It gives a uniform Internet interface to traditional equipments. It can be embedded in any equipment easily even your lamp. This paper presents a novel approach to control devices with embedded web servers over the Internet and to form device networks such that their components can make use of one another’s services and functions while improving the user interfaces.

The rest of this paper is organized as follows. In section II, gives survey of available web server. Section III, introduces the TCP/IP Protocol Suite. In section IV, hardware description on which we realize the function of the embedded web server. In section V, working of EWS. In section VI, we gives description on Web server analysis. Sections VII, gives concept of software. In section VIII, conclusion was given.

II. LITERATURE SURVEY

The basic design concept for mini embedded web servers introduced that serves the common devices interfaces and gains a good performance by an internet [10]. This paper was efficient to be used in industry, medical, and other fields at a broad scale and also set the goal of the embedded web server to link with device, many home appliances and also, the small size of this is useful to be embedded into small appliances [10].

According to one paper [15] it is said that the web access is enabled by the server for distributing the measurement or control of systems. This provides a scalable networking solution that is optimized for the instrumentation, educational laboratories, industrial and home automation. The author [15] also says that the users can monitor and control transducers on active Web pages enhanced with JavaScript.

Bill Columbia [16] presents the basic concepts on the use of embedded web servers with simple applications. The author put forth the idea of using embedded web servers for time critical and non-time critical applications.

The paper [17] presents a web-based caregiver monitoring system for assisting visually impaired people. The objective of this system is to assist blind and low vision people to walk around independently and safely in transportation centers by providing speech guidance on their current location and navigation information on how to move to a particular location. The system will also alert caregivers when the visually impaired person needs assistance using a web-based monitoring system.

The author [1] introduces a new line of high performance derivatives has many positive implications for improving the power efficiency of 8051-based designs. Its low cost is an advantage when designing embedded systems for high volume applications.

III. TCP/IP PROTOCOL SUITE

TCP/IP (Transmission Control Protocol/Internet Protocol) is the basic communication language or protocol of the Internet. It can also be used as a communications protocol in a private network (either an intranet or an extranet). When you are set up with direct access to the Internet, your computer is provided with a copy of the TCP/IP program just as every other computer that you may send messages to or get information from also has a copy of TCP/IP. TCP/IP is a two-layer program. The higher layer, Transmission Control Protocol, manages the assembling of a message or file into smaller packets that are transmitted over the Internet and received by a TCP layer that reassembles the packets into the original message. The lower layer, Internet Protocol, handles the address part of each packet so that it gets to the right destination. Each gateway computer on the network checks this address to see where to forward the message. Even though some packets from the same message are routed differently than others, they’ll be reassembled at the destination.

![Figure 1: Schematic clients-server interaction.](image-url)
TCP/IP uses the client/server model of communication in which a computer user (a client) requests and is provided a service (such as sending a Web page) by another computer (a server) in the network. TCP/IP communication is primarily point-to-point, meaning each communication is from one point (or host computer) in the network to another point or host computer. TCP/IP and the higher-level applications that use it are collectively said to be "stateless" because each client request is considered a new request unrelated to any previous one (unlike ordinary phone conversations that require a dedicated connection for the call duration). Being stateless frees network paths so that everyone can use them continuously. (Note that the TCP layer itself is not stateless as far as any one message is concerned. Its connection remains in place until all packets in a message have been received.)

Many Internet users are familiar with the even higher layer application protocols that use TCP/IP to get to the Internet. These include the World Wide Web's Hypertext Transfer Protocol (HTTP), the File Transfer Protocol (FTP), Telnet (Telnet) which lets you logon to remote computers, and the Simple Mail Transfer Protocol (SMTP). These and other protocols are often packaged together with TCP/IP as a "suite."

**TCP/IP Suite Layers**:

The TCP/IP protocol suite is a combination of different protocols at various layers. TCP/IP is normally considered to be a 4-layer system as shown in Figure 2.

<table>
<thead>
<tr>
<th>Application</th>
<th>Telnet, FTP, HTTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport</td>
<td>TCP, UDP</td>
</tr>
<tr>
<td>Network</td>
<td>IP, ICMP</td>
</tr>
<tr>
<td>Link</td>
<td>Interface card</td>
</tr>
</tbody>
</table>

Figure 2: Four Layers of TCP/IP Protocol Suite

Protocols related to TCP/IP include the User Datagram Protocol (UDP), which is used instead of TCP for special purposes. Other protocols are used by network host computers for exchanging router information. These include the Internet Control Message Protocol (ICMP), the Interior Gateway Protocol (IGP), the Exterior Gateway Protocol (EGP), and the Border Gateway Protocol (BGP). The four layers of TCP/IP suite is described as

1. **Application Layer**

   The Application layer handles the details of a particular application. Common TCP/IP applications include:
   - Telnet for remote login
   - Browser support for displaying web pages
   - File transfer applications
   - E-mail applications

   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. The three lower layers do not know anything about the specific application and only take care of communications details.

2. **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.

3. **Network Layer**

   The network layer controls the communication between hosts on the Ethernet. There is no form of transmission control to ensure that IP datagram arrive to the host or that all IP datagram 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. **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.
HTTP (Hypertext Transfer Protocol) is the set of rules for transferring files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web. As soon as a Web user opens their Web browser, the user is indirectly making use of HTTP. HTTP is an application protocol that runs on top of the TCP/IP suite of protocols (the foundation protocols for the Internet). HTTP is a simple protocol that is based on a TCP/IP protocol stack (Fig. 3). HTTP uses TCP. TCP is a relative complex and high-quality protocol to transfer data by the subordinate IP protocol. TCP itself always guarantees a Safeguarded connection between two communication partners based on an extensive three way handshake procedure. As a result the data transfer via HTTP is always protected.

IV. HARDWARE DESIGN

The Block diagram shown in the fig. 4 gives an overview of the interconnection between various devices in the design. The resources of the Micro Controller are used in such a way that simultaneous processing of information from the various sensors and also to send/receive data over the internet via the Ethernet Controller is made possible. The full 16-bit operation of both the Micro Controller and the Ethernet Controller ensures no queuing up of data received from the internet [2].

### Table 1: Specifications of EWS

<table>
<thead>
<tr>
<th>DEVICE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>LPC2148</td>
</tr>
<tr>
<td>Memory</td>
<td>SDRAM 64M</td>
</tr>
<tr>
<td>I/O Port</td>
<td>USB,MMC,CF,PCMCIA</td>
</tr>
<tr>
<td>Ethernet</td>
<td>Microchip enc28j60</td>
</tr>
<tr>
<td>Wireless LAN</td>
<td>IEEE802.11B</td>
</tr>
</tbody>
</table>

Having thought of the memory card as the bulk memory for the project and realizing just how cheap it was, the potential uses seemed to grow enormous. The web server includes enough memory to develop large applications on top of the web server protocols. 32K bytes of external SRAM are 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.

A standard RJ45 patch cable can be used to connect the module to either a 10 Mbps or 100 Mbps hub. A 100 Mbps hub automatically switches down its transfer speed to 10Mbps if it detects the ENC28J60 running at 10 Mbps.

The shortcoming can be overcome by using an Embedded Web Server in place of a PC based Server. Embedded Web Server is a single-chip implementation of the Ethernet networking standard. By embedding Ethernet onto a device, it has the capability to communicate via Ethernet without using a computer. The server enables Web access to the automation and monitoring system and provides a scalable networking solution that is optimized for instrumentation, industrial and home automation. The users can browse the home page of the system using web browser, and control the household appliances and enquire about their operational status.

V. WORKING

This project implements an EMBEDDED WEBSERVER with networking capability using ARM microcontroller. Various Analog Sensors can be connected to the ARM Board. The project includes complete implementation of an HTTP Web Server in arm ARM7 microcontroller. The websites are stored inside the program space of the ARM7 microcontroller and features a flexible pattern parsing algorithm. This supports using keywords to instruct the web server to include special data in the page delivered to the browser (i.e. current temperature as ASCII text).
Using any standard web browser on any PC you can access the web pages performing a variety of operations like viewing a temperature plot of the last 24 hours, control the servo motor, read/write any I/O pin by using a simple mouse-click, upload any file to the Data Flash storage and access files stored on the flash. The web server implements ARP, IP, TCP, UDP, HTTP (server), NTP (client), servo control, I/O Pin control, 2nd software UART etc.

VI. WEB SERVER ANALYSIS

As an example of how to use the previously described module, a demonstration HTTP server was implemented. The module must have been powered on, properly connected to LAN and the TCP/IP settings of the local host correctly configured. Then, the embedded Web server is ready. The server provides an HTML Web page that is stored in MCU flash memory. The module waits for an incoming connection, transfers the Web page, closes the connection and waits for another client to connect. The content of this Web page is adapted dynamically with analog values. Before sending a segment of TCP data, it searches the transmit buffer for special strings. If such a string is found, it is replaced by an A/D converter value [1].

The page has three HTML labels that display Analog-to-Digital (A/D) values such as CPU/air temperature and operating voltage and a radio button pair that toggles the main board Light Emitting Diode (LED) state. One purpose of a small Web server is to make a product ease of use. This page is bidirectional in that it both displays device information data and controls the board LED on or off. The new state of the LED is sent to the Web server in a post message. There is an image, which the browser loads after the HTML portion. The Fig. 5 gives the overall system design.

Most of the tested browsers establish a single connection to load both parts of the page, but others, open two separate connections to the server. Each connection comes from a different port on the browser’s machine. To handle this situation, all connection specific information, such as client IP address, port number, sequence number, ACK number, and TCP state is stored into a structure that is indexed by the connection number. When a TCP segment arrives, if it matches with an existing connection, then it uses its state information.

VII. SOFTWARE CONCEPT

The main part of the embedded web server is the RTOS handling the web server application. Whenever connection is established, a new task is created using μC/OS-II. And the web server application is executed for that user’s application, running in a task. When a new connection is established, a new task is created and the user’s application is executed in a separate task as a separate application. This process is continued for all the users connecting to the server. Fig. 6 shows the flowchart of the embedded web server and the RTOS, managing the incoming connections.

![Flowchart of Embedded Web Server Tasks](image-url)
Testing the Embedded Web Server

Testing: Initially, the target is tested for the working of operating system. This is done by booting the target using the hyper terminal. After the target is successfully booted with μC/OS-II, it is tested over the network. Now the embedded web server is responding to the clients, request is made to the server, embedded web server, by typing the IP address of the server in the client’s browser.

The user has to enter ‘10.1.1.26’ IP to access the server. This request is taken by the operating system of the client and given to the LAN controller of the client- system. The LAN controller sends the request to the router that processes and checks for the system connected to the network with the particular IP address. If the IP address entered is correct and matches to that of the server, a request is sent to the LAN controller of the server and a session is established and a TCP/IP connection is established and the server starts sending the web pages to the client[7].

VIII. CONCLUSION

Embedded processing is already powerful enough to tackle the real-world applications. We have introduced the general design concept of the embedded web server and the policy of TCP/IP reduction, special 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. This system will not only be useful in industry field, but also has great future in smart-house applications, networked lighting control system and other distributed control systems. This system is very suitable for acquiring data or signals form a large scale industry field. Hundreds of such terminals can be grouped within a network.

REFERENCES

[14]. Karia, D.C.; Adajania, V.; Agrawal, M.; Dandekar, S., “Embedded web server application based automation and monitoring system”; Sardar Patel Inst. of Technol., Univ. of Mumbai, Mumbai, India

AUTHORS PROFILE

Savita Lad is MTECH student in BVDUCOE, Pune. Her research interest area in computer networking and web services.

Prof. Pramod P. Jadhav is an Assistant Professor at PICT, Pune. His area of interest includes computer networking and web services.

Prof. R.J. Vaidya is an Associate Professor and Head of Department at BVDUCOE, Pune. His area of interest in nano technology and internet computing.