EMBEDDED WEB SERVER

MUHAMMAD SHUKRI BIN AHMAD

This thesis is submitted as partial fulfillment of the requirements for the award of the Master Degree of Electrical Engineering

Faculty of Electrical & Electronic Engineering
Kolej Universiti Teknologi Tun Hussein Onn

OCTOBER, 2004

For my beloved family

ACKNOWLEDGEMENT

I am deeply grateful for the help that I received from my supervisor, Associate Professor Awtar Singh, during the development of this project. His willingness to help and ideas has kept me on my toes from the beginning stage of this project until the completion of this thesis.

I would also like to extend my gratitude to all lecturers that has given me all the basic needed for completing this project, and also to my classmates and friend for their encouragement and help.

I could not have done this project without the unconditional support, active encouragement, complete cooperation, and honest sacrifice by my family. To appreciate their immense contribution, this thesis is lovingly dedicated to them.

ABSTRAK

Dilihat dari sudut teknologi, internet berkemungkinan besar merupakan perkara yang terhebat berlaku terhadap kita selepas revolusi perindustrian. Ia membawa kita kepada era letusan maklumat. Internet telah menjadi sebahagian daripada hidup kita seharian melalui pelbagai alatan di pejabat dan rumah yang disambungkan ke internet. Faedah yang diperolehi adalah tidak terbatas. Bayangkan situasi di mana kita boleh meggunakan alatan eletrik secara kawalan jauh atau menggunakan pengantaramuka interaktif untuk mengawal perkakasan rumah. Ini membolehkan kita menjalankan pelbagai penyenggaraan perkakasan dan diagnostic daripada sebarang tempat di dunia yang mempunyai akses ke internet. Pengantaramuka web ini memberikan faedah kepada pengguna dan juga kepada pembuat perkakasan. laporan ini membincangkan tentang perkara-perkara yang perlu diambil kira ketika mereka bentuk sebuah system kawalan berasaskan pelayan web *embedded.* Sebuah sistem akan dibangunkan menggunakan PICDEM.net Development Board, yang membolehkan pengguna mengawal dan perkakasan secara jarak memantau jauh.

ABSTRACT

In term of technology, internet must be the greatest thing to happen to mankind after industrial revolution. It has led us into the age of information explosion. Internet is becoming part of our everyday life through the various devices which are connected to internet at our home and work. The benefits are endless. Consider the situation where we can 'use' equipment from a remote location or work using an interactive user interface on our household device. This enables us to perform device management and even appliance diagnostics from any where in the world where you have the internet access. Thus web interfaces bring benefits to end-users as well as to manufacturers. This report discussed the design consideration of a web server based control system. Using PICDEM.net Development Board as a project, a system will be developed, allowing a user to control and monitor device remotely.

TABLE OF CONTENTS

CHAPTER II: LITERATURE REVIEW

CHAPTER III: METHODOLOGY

CHAPTER IV: EXAMPLE APPLICATION

CHAPTER V: CONCLUSION AND RECOMMENDATION

LIST OF TABLE

LIST OF FIGURE

GLOSSARY OF ABBREVIATIONS

API	Application programming interface
ARP	Address resolution protocol
CGI	Common gateway interface
CPU	Central processing unit
CSMA/CD	Carrier sense multiple access with collision detection
DHCP	Dynamic host configuration protocol
FDDI	Fiber distributed data interface
FIN	Finish
FTP	File transfer protocol
HTML	Hypertext markup language
HTTP	Hypertext transfer protocol
ICD	In-circuit debugger
ICMP	Internet control message protocol
IP	**Internet protocol**
LAN	Local area network
MAC	Medium access control
RAM	Random access memory
ROM	Read only memory
SMTP	Simple mail transfer protocol
TCP	**Transmission control protocol**
WWW	World wide web

LIST OF APPENDIX

CHAPTER I

INTRODUCTION

1.1 Project Background

A web server is basically a program that can respond to requests from web browsers. The server *listens* to a port. A web browser then sends a request to the server (1) see Figure 1.1. The request contains the name and location of the wanted file (Uniform Resource Location, URL). The server receives the request and tries to interpret the request (2). If the web server can find the requested file it will return it to the web browser (3). Otherwise, an error message will be returned.

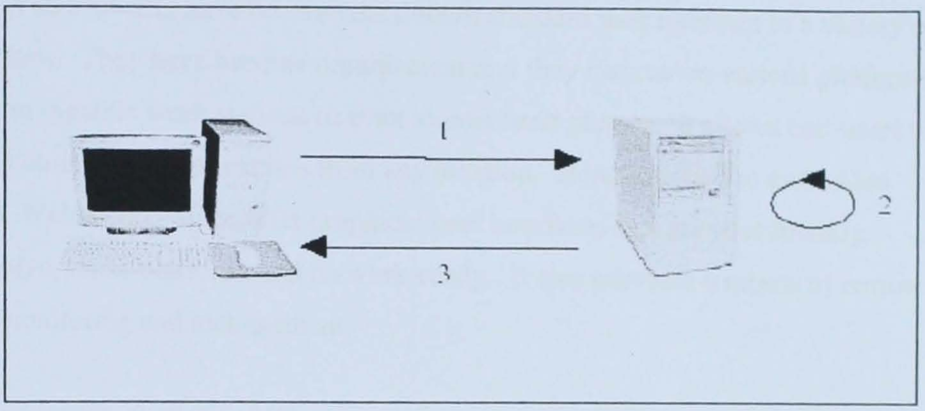

Figure 1.1: Basic Concept of a Web Server

An embedded web server is a web server that has been designed to be included in different kinds of small devices. It is an easy and well-defined way for information access. When designing an embedded web server there are some requirements to take notice of. First of all it has to have a small memory footprint (>100 kb is quite much). Often the devices do not have a file system (hard drive), so it must be possible to store the web content on ROM or to have the web pages created dynamically (created when requested).

Embedded systems are integrated hardware and software systems implementing dedicated functions, such as, electronic gadgets, communication devices, household appliances and medical instruments. We are fast approaching a point where there are more microprocessor based systems than people on the planet. They are generally not programmed by the users. In the last few years there has been a significant demand for remote monitoring and control of embedded systems. The most convenient way to do this is to connect the embedded system to the internet, replacing traditional serial user interfaces, and allowing the use of familiar Web browsers from any desktop.

Web browsers have become the default standard user interface to a variety of applications. They have become omnipresent and they can run on various platforms right from gigantic work stations to even almost hand phones. It allows end-users to access Web-enabled applications from any location. Hence applied to embedded systems, Web technologies offer graphical user interfaces that are user friendly, inexpensive, cross-platform, and network-ready. It also provides a means of remote system monitoring and management.

By using an embedded Web server, developers can format and display the same data with HTML and any standard browser. The user's response to the data can include input to modify the embedded device's configuration. Moreover, communication with the embedded application can use Ethernet or fast serial connections.

1.2 Problem Statement

The enormous growth of the internet has established itself in every corner of our life, thus making it an indispensable tool to work with. As embedded systems become more prevalent, the need for connectivity of these devices to the World Wide Web becomes inevitable. The integration of the ubiquitous internet with embedded devices brings an almost limitless possibility of application. Affordable monitoring and control of real-time environments over the Internet is of importance to the general public, as well as professionals.

1.3 Objective

The project objectives are as follows:

- To determine the design consideration of an embedded web server system
- To make it possible to control and monitoring device remotely using this system via LAN.
- To design sample application for this system

1.4 Scope of Work

The scope of the project is to design, implement and illustrate mechanisms providing connectivity between an embedded system and the .internet.

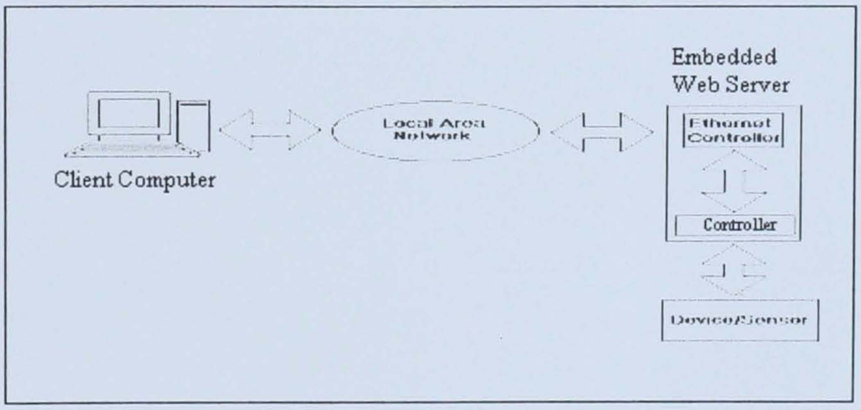

Figure 1.2: The Project Block Diagram

1.5 Project Motivation

The motivation of this project is the increasing importance of home networking. More and more intelligent devices are being used. By connecting these devices to internet there are so many advantages. Also this project is the continuation of my degree project but replacing the server computer with an embedded web server.

CHAPTER II

LITERATURE REVIEW

2.1 The Internet

Internet is an astonishing technology because the internet created a new space in our physical world. This is due to the success of the World Wide Web (WWW) and its common application. In this section of literature review, we will look at the common protocol, markup language and script used by the internet.

2.1.1 Protocols

Both the appliances and the home gateway should run standard protocols to enable interoperability of devices from different manufacturers. The standard networking language of today's Internet is TCP/IP, or Transmission Control Protocol and Internet Protocol. Each device should have an embedded TCP/IP stack with the ability to communicate over some type of standard wire, the Ethernet, wireless, etc

Web server implements four of the basic protocols in the TCP/IP protocol suite (TCP, IP, ICMP and ARP). Application layer protocols such as HTTP, FTP or SMTP can be implemented as an application. [1]

2.1.1.1 Transmission Control Protocol (TCP)

TCP is a connection-oriented, end-to-end reliable protocol designed to fit into a layered hierarchy of protocols which support multi-network applications. The TCP provides for reliable inter-process communication between pairs of processes in host computers attached to distinct but interconnected computer communication networks. Very few assumptions are made as to the reliability of the communication protocols below the TCP layer. TCP assumes it can obtain a simple, potentially unreliable datagram service from the lower level protocols. The TCP interfaces on one side to user or application processes and on the other side to a lower level protocol such as Internet Protocol. [2]

TCP uses only a single type of protocol data unit call the TCP segment. The header is shown in figure 2.1. Because one header must serve to perform all protocol mechanism, it is rather large, with minimum length of 20 octets. The fields are as follows:

- **Source port**: source TCP user
- **Destination port**: destination TCP user
- **Sequence number**: sequence number of the first data octet in this segment except when the SYN flag is set. If SYN flag is set, it is the initial sequence number (ISN) and the first data octet is ISN + 1.
- **Acknowledgement number**: a piggybacked acknowledgement. Contains the sequence number of the next data octet the entity expects to receive.
- **Data offset**: number of the 32 bit word in the header
- **Reserved**: reserved for future used.
- **Flags**: shown in table 2.1.
- **Window**: flow control credit allocation, in octet. Contains the number of the data octet beginning with the one indicated in the acknowledgement field that the sender is willing to except.
- **Checksum**: the ones complement of the sum modulo 216 – 1 of all the 16 bit words in the segment plus a pseudo header.
- **Urgent pointer**: points to the last octet in a sequence of urgent data. This allows the receiver to know how much urgent data are coming.
- **Option (variable)**: an example is the option that specifies the maximum segment size that will be accepted.

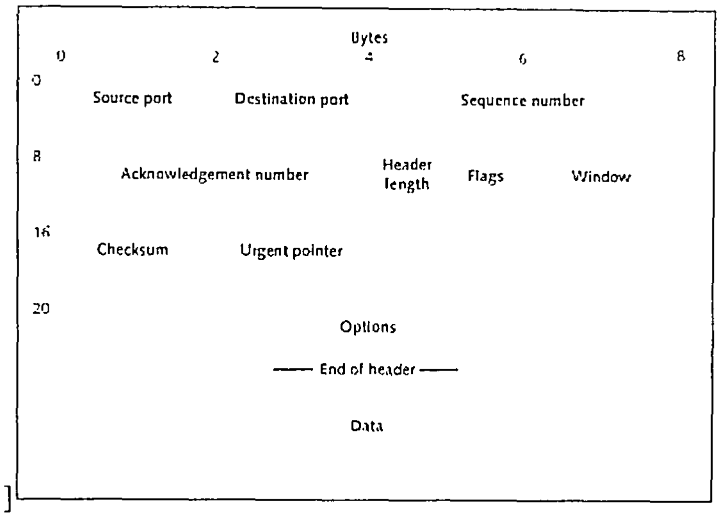

Figure 2.1: TCP segment format

Table 2.1: TCP Flags

Flags	Purpose
URG	Urgent pointer field significant
ACK	Acknowledgement field significant
PSH	Push function
RST	Reset the connection
SYN	Synchronize the sequence number
FIN	No more data from sender

Port is defined by numbers and being used by the application to determine the handlers. For example, if the port numbers are 80, it will call the http handler, usually the GET method.

Table 2.2: TCP port and services

Port	Services	
7	Echo	Echoes back all incoming character
9	Discard	Discard all incoming changes
13	Daytime	Return time and data string
21	FTP	File transfer protocol
23	Telnet	Remote login
25	SMTP	Simple mail transfer protocol
37	Time	Binary value of date and time
80	HTTP	Hypertext transfer protocol

Connection establishment in TCP always uses a three-way handshake. When the SYN flag is set, the segment is essentially a request for connection and functions. To initiate a connection, client sends a SYN, $SN = X$. where X is the initial sequence number. The server responds with SYN, $SN = Y$. $AN = X + 1$ by setting both the SYN and ACK flags. Note that the acknowledgment indicates that the receiver is now expecting to receive a segment beginning with data octet $X + 1$, acknowledging the SYN, which occupied $SN = X$. Finally, the initiator responds with $AN = Y + 1$. If both sides issue crossing SYNs, no problem results: Both sides respond with ACKs. This process is shown in figure 2.2[3].

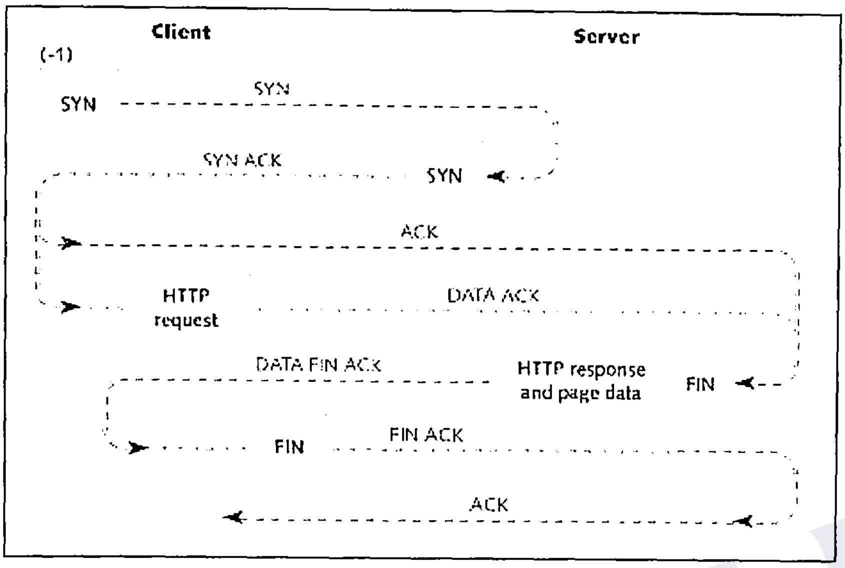

Figure 2.2: TCP connection process

2.1.1.2 Internet Protocol (IP)

The Internet Protocol is designed for use in interconnected systems of packet-switched computer communication networks. The internet protocol provides for transmitting blocks of data called datagrams from sources to destinations, where sources and destinations are hosts identified by fixed length addresses. The internet protocol implements two basic functions: addressing and fragmentation. The internet modules use the addresses carried in the internet header to transmit internet datagrams toward their destinations. The selection of a path for transmission is called routing. The internet modules use fields in the internet header to fragment and reassemble internet datagrams when necessary for transmission through "small packet" networks. [4]

REFERENCES

1 Andrew S Tanenbaum, ``Computer Networks `` Prentice Hall publications, 2002

2. Postel.J, Transmission Control Protocol, RFC 793, Internet Engineering Task Force, September 1981

3. William Stalling, "Data & Computer Communication" Prentice Hall, 2002

4. Postel.J, Internet Protocol, RFC 791, Internet Engineering Task Force, September 1981

5. Postel.J, Internet Control Message Protocol, RFC 826, Internet Engineering Task Force. September 1981

6. Peter G. Brierley,"Microsoft Windows NT Server 4.0", Course Technology 2000

7. Berners-Lee, R. Fielding and Frystyk, Hypertext Transfer Protocol-HTTP/1.1,RFC 2068, Network Working Group, 1997

8. World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C Recommendation 6 October 2000, www.w3c.org.

9. CGI Programming 101, www.cgi101.com

10 Netscape, www.netscape.com

11. Glencomm Systems, 1992, Networking Reference Charts [Online] Available from http://www.glen.com/reftech.htm

12. Love, R., et al., 1998, Migrating to High-Speed Token Ring. [Online] Available from http://www.hstra.com/trwp0798final.html

13. Nielsen, H., 1996, CERN httpd. [Online]. Available from http://www.w3.org/Daemon/

14. Lovell, B. Analysis of an Ethernet Controller, The University of Queensland, E3470 Lecture, 1998.

15. Burns, I, ehttpd – An Embedded HTTP Server, The University of Queensland, Undergraduate Thesis, 1997.

16. Comer, D., Internetworking with TCP/IP: Volume 1 – Principles, Protocols and Architecture, New Jersey: Prentice-Hall Inc, 1995.

17. Burton, H., 1997, First International Workshop on Embedded Web Technologies: Summary Report, Sixth International WWW Conference. [Online]. Available from http://diagnostics.stanford.edu/www6/EWTReport.html

18. Lawrence, S., 1997, Efficient Implementation of WWW in Embedded Systems. [Online] Available from http://diagnostics.stanford.edu/www6/papers/Agranat.html

19. Wingard, S., 1997, Embedded Web Server Technology. [Online] Available from http://diagnostics.stanford.edu/www6/papers/Spyglass.htm

20. Microtest, 1997, WebZerver from Microtest. [Online] Available from http://www.zerver.com/webzerver/html/productoverview.html

21. Dawning Technologies, 1998, Secure Network Interface. [Online] Available from http://www.dawning.com/snipage.htm

22. Lee, Burton H., "Embedded Internet Systems: Poised for Takeoff", IEEE Internet Computing, 2(3):24-29, May1998.