



The Embedded Web server based Electrical Ethernet Monitoring system using ARM

K.Bharath reddy¹, Ch. Rajendra prasad²

PG Student, Department of ECE, SR Engineering College, Warangal¹

Assistant professor, Department of ECE, SR Engineering College, Warangal²

Abstract: In the monitoring system, the existing resource of Ethernet is used to remotely monitor power network parameters. A practical design of embedded Web server for power network monitoring is introduced, which applies high speed MCU (ARM7-LPC2148) to conveniently link to Ethernet. Adopting Ethernet interface control mode, the remote network monitoring of power network is realized under Windows.

A web server in the device provides access to the user interface functions for the device through a device webpage. A web server can be embedded into any appliance and connected to the Internet so the appliance can be monitored and controlled from remote places through the browser in a desktop. The aim of the paper is to control the devices or equipment's from the remote place through a web page. The web-server circuit is connected to LAN or Internet. The client or a person on the PC is also connected to same LAN or Internet. By typing the IP-address of LAN on the web browser, the user gets a web page on screen; this page contains all the information about the status of the devices. The user can also control the devices interfaced to the web server by pressing a button provided in the web page.

Keywords: Keywords-Ethernet; embedded Web server, monitoring system; TCP/IP protocol

I. INTRODUCTION

Nowadays, controlling a system via PC is very common. A web base control and monitoring system can make us control a system without distance. So developing a cost effective, programmable and high efficiency controller webpage is necessary for the world competition. The propose of this paper is to build a remote control system through a webpage. And this system is controlled through local area network by using an embedded TCP/IP Rabbit Core Module 5170. The RCM 5170 is chosen because of have easy Program download utility, ideal for network-enabling security and lots of storage. The basic block diagram is shown in below figure.

connected by a converter that can convert the hardware like sensor and motor signal for RCM 5170. The

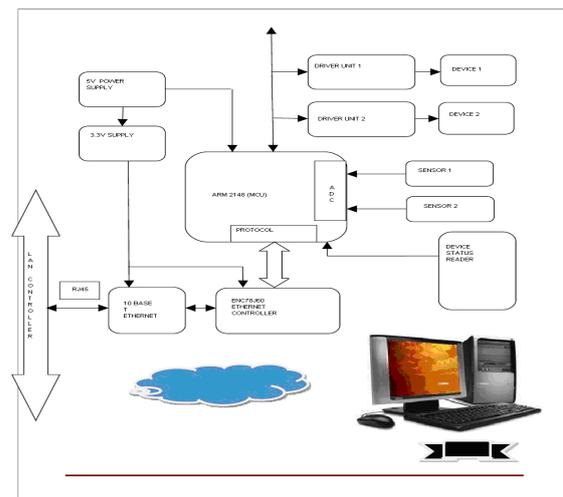


Figure 1 Basic block diagram

II. PROPOSED ELECTRICAL ETHERNET MONITERING SYSTEM

This paper is divided to hardware and software part. The hardware part is building up an I/O that can connect to the RCM 5170. The connection between the Input/output is

hardware for electrical Ethernet monitoring system is shown in figure 2

For the software part, webpage base embedded software will be build for enable personal computer to



communicate with RCM 5170 by using Dynamic C. The changing in the I/O will be shown in the personal computer by accessing the webpage.

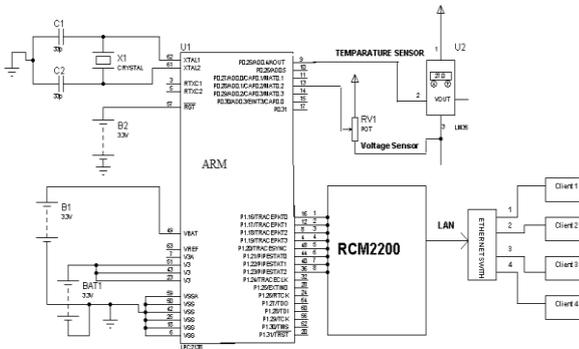


Figure 2 Electrical Ethernet monitoring system

DYNAMIC C

Dynamic C is an integrated development system for writing embedded software. It is designed for use with controllers based on the Rabbit microprocessor. The Rabbit family of processors is high-performance 8-bit microprocessors that can handle C language applications of approximately 50,000 C statements or 1 MB.

Dynamic C is based on the C Language, as the programming methodologies and the basic principles are same. Dynamic C has its own set of libraries, which include user callable functions. Dynamic C libraries are in source code, allowing the creation of customized libraries.

III. EMBEDDED WEB SERVER

A. The system diagram of Embedded Web server

The system structure of embedded Web server is shown in Fig. 3. The entire system uses B/S mode. The client PC is connected to the Internet through a browser and then gets access to the embedded Web server. Through this way, remote login and operation are realized [5]. Compared with the traditional C/S mode, this mode is simple to use, convenient to maintain, and easy to extend.

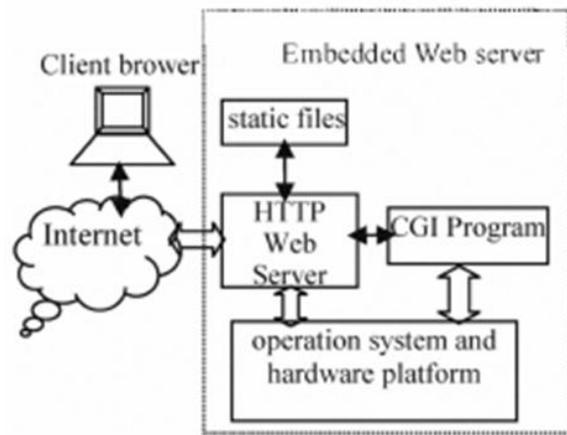


Figure 3 The system diagram of Embedded Web server

B. The choice of Embedded Web server

Generally speaking, the embedded devices have limited resources and don't need to handle the requests of many users simultaneously. Therefore they do not need to use the most commonly used Linux server Apache. Web server which is specifically designed for embedded devices are applied in such case [6]. This kind of Web server requires relatively small storage space and less memory to run, which makes it quite suitable for embedded applications. The typical embedded Web server has three kinds, namely https, Boa and tthttpd [7]. As the simplest Web server, https has the weakest functions among the three. It does not support authentication and CGI technology while Boa and tthttpd support these functions [8]. If Web server only provides some static web pages such as simple online help and system introduction, then a static server can be adopted; if you need to improve system security or interact with users such as real-time status query and landing, then you have to use dynamic Web technologies. In such situation, either Boa can achieve these goals. In the present research, we adopt Boa, the Web server suitable for embedded system, because it has less function and needs far more resources to run.

C. The principle of Embedded Web server Boa

Boa is a single task Web server. The difference between Boa and traditional Web server is that when a connection request arrives, Boa does not create a separate process for

each connection, nor handle multiple connections by copying itself. Instead, Boa handles multiple connections by establishing a list of HTTP requests, but it only forks new process for CGI program. In this way, the system resources are saved to the largest extent. Like a common Web server, an embedded web server can accomplish tasks such as receiving requests from the client, analyzing requests, responding to those requests, and finally returning results to the client. The following is its work process.

- Complete the initialization of the Web server, such as creating an environment variable, creating socket, binding a port, listening to a port, entering the loop, and waiting for connection requests from a client.
- When there is a connection request from a client, Web server is responsible for receiving the request and saving related information.
- After receiving the connection request, Boa analyzes the request, calls analysis module, and works out solutions, URL target, and information of the list. At the same time, it processes the request accordingly.
- After the corresponding treatment is finished, the Web server sends responses to the client browser and then closes the TCP connection with the client. For different request methods, the embedded Web server Boa makes different responses. If the request method is HEAD, the response header will be sent to the browser; If the request method is GET, in addition to sending the response header, it will also read out from the server the URL target file of the client request and send it to the client browser; If the request method is POST, the information of the list will be sent to corresponding CGI program, and then take the information as a CGI parameter to execute CGI program. Finally, the results will be sent to client browser.

IV. RESULTS

The setup of the embedded web server is shown in Fig. 4. The target is connected to the Ethernet controller of the network, the power to the target is given through USB and

the debugger is connected to the USB of the system and the other end to the board.

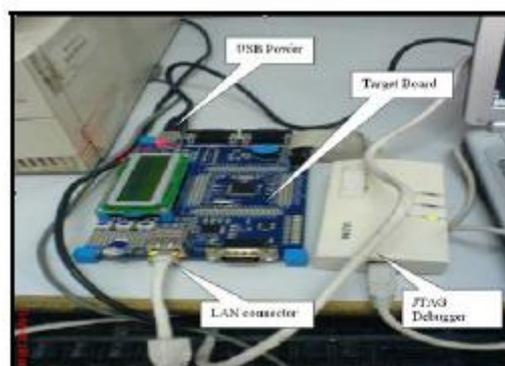


Figure 4 Setup of Ethernet monitoring system

A. Integrating RTOS and Embedded Web Server

The main part of the embedded web server is the RTOS handling the web server application. Whenever, a connection is established, a new task is created using $\mu\text{C}/\text{OS-II}$. 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 8 shows the flowchart of the embedded web server and the RTOS, managing the incoming connections.

B. Porting the Web Server

The Embedded web server application and the RTOS are ready and are to be ported on the target. The porting is done using Keil uVision3 compiler. Now, the code is loaded on the target and the target is configured over the network, to work as an embedded web server.

C. 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 $\mu\text{C}/\text{OS-II}$, it is tested over the network using ping command. The embedded web server, responding to the ping command made by the client. 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. Fig 5 shows the client entering the IP address of the server and the server send the html page to the requested clients.

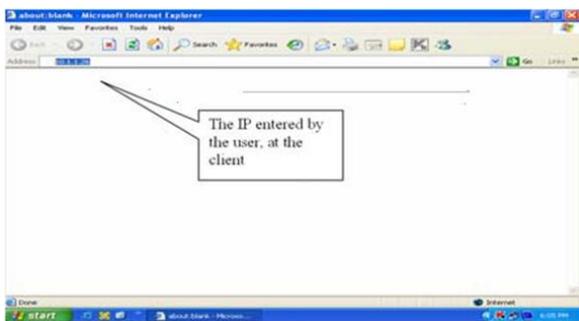


Figure 5 Users Entering the IP Address

Fig 6 shows various clients connecting to the server and getting the html page. finally client can monitor the the Channel Data on Internet Explorer

The embedded web server that has been designed can be used in educational institutions, offices and many other places. For web-based network element management provide an administrator with a simple but enhanced and more powerful user interface without additional hardware.

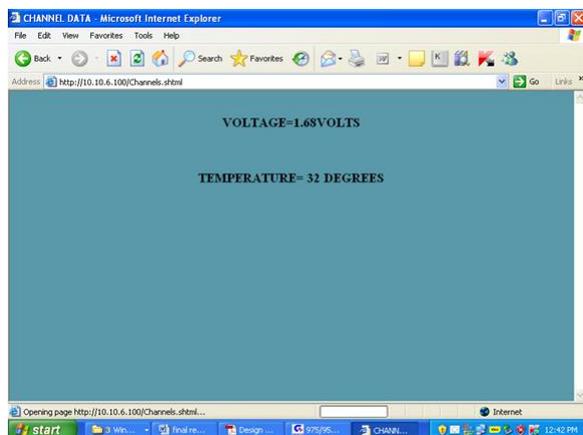


Figure 6 Monitoring the Channel Data on Internet Explorer

IV.CONCLUSION

Software contention and architectures can significantly affect web server performance. Poorly designed and configured software architectures might even generate high response times while the physical resources display low utilization. A remote user only requires a common Internet browser to carry out experiments on real hardware. The embedded web server replaces the PC, which is required for remote labs with special hard- and software.

REFERENCES

- [1] Yu-Lung Ke, and Ying-Chun Chuang. A Novel Single-Stage Power-Factor-Correction Circuit with High-Frequency Resonant Energy Tank for DC-Link Inverters. IEEE Transactions on Circuits And Systems—II: Express Briefs, Vol. 53, no. 2, FEBRUARY 2006. Pp.115-119
- [2] Qun Zhao, Fengfeng Tao, Peng Xu. Improved Performance of Continuous Current Mode Boost Converter for Power Factor Correction. Proc. of APEC97_pp.266-271
- [3] Wang Xinyu_Zhang Jing_and Ye Jiazhuo__Study on Energy Acquisition and Billing Automation System," Computer engineering and application_pp.240-243 June 200.pp.240-243.
- [4] Wu Zaijun_Wang Zheng_and Hu Mingqing__Implementation of Centralized Automatic Meter Reading Apparatus and Research on its Communication,"Power System Technology _ February 2002.pp.44-47.
- [5] Sung-JinChoi, Kyu-ChanLee and BoHyungCho.Design of Fluorescent Lamp Ballast With PFC Using a Power Piezoelectric Transformer. IEEE Transactions on Industrial Electronics, Vol. 52 ,no .6, DECEMBER 2005.pp.1573-1581.
- [6] FAN Yizhi. "Visual Basic and RS-232 serial communication control," .Beijing:chinese Young People Press.July 2002. pp.126-128.
- [7] Wang Xianchun, Guo Jierong, Hu Weiwen, and Fan Xiping, "Design and Implementation of Embedded Web Server Based on ARM and Linux," Micro Computer Information, vol. 23(5-2),2007, pp. 164-165.
- [8] Jacek W, "Embedded Internet technology in process control devices," IEEE Internet Computing, Vol. 34, 2000.
- [9] Liu Yingshui, Xiao Zhengyu, and Sun Wei, "Embedded Web Server Based on ARM and Linux," Microcontrollers & Embedded Systems, June 2007, pp. 14-21.
- [10] Mi-Joung Choi, Hong-Taek lu, Hyun-Jun Cha, Sook-Hyang Kim, and Won-Ki Hong, "An Efficient Embedded Web Server for Web-based Network Element Management," International Journal of Network Management, Vol. 10, May 2000.