Web based Remote Accessing of Medical Devices with ARM Cortex-M3

Gopi Krishna S, P.Anantha Christu Raj, K.Gerard Joe Nigel

***Abstract*— Remote monitoring of a medical device in critical conditions is an emerging trend where cost reduction, portability and mobility are the main focuses. Integrating web servers to these medical devices will aid in monitoring them over the Internet and also in creating effective user interfaces in the form of web pages. This paper explores the topic of an efficient and low-weight embedded Web server for Web-based medical monitoring management. In this paper, the controller that is used is an Arm cortex M3 (LM3S9B92), which support all the networking protocol. CGI script will act as a bridge between the controller and the browser, helps for accessing the devices that are connected to the controller from the browser.**

***Index Terms*— Webit's, HTML, LwIP, CGI script.**

I. INTRODUCTION

As the Web (or World-Wide Web) technology evolves, it makes its underlying technologies to be much more useful than the term browsing the Web. Web Browser is a 'De Facto' standard of user interface for various kinds of applications. It had a profound positive effect on consumer expectations in terms of speed, adaptability, accessibility to remote content, data availability from worldwide & its applications are far beyond consumer usage, such as Automation monitoring, Networking & Communication, Data transaction and Medical field.

Web-based medical device management gives an administrator the ability to monitor & manage network interfaced medical devices over the Internet using a web browser. Now, webserver technology is the combination of both embedded and networking fields, which provides flexibility to the clients for remote monitoring and managing the data within a browser and also one of the latest trends in the embedded technology.

In this paper design process has been simplified by using networking protocols (like Thin weight IP, Auto IP, DHCP) and Arm CORTEX-M3. This paper discuss about development of an efficient and low-weight EWS's (embedded web server) for web based management in medical field. It proposes the architecture for simple but powerful application interface, hardware design and protocols used. Finally it presents the performance and the result of the low-weight EWS's.

Revised Manuscript Received on 30 July 2013.

* Correspondence Author
Gopi Krishna S, PG Scholar, Electronics and Instrumentation Engineering, Karunya University
P.AnanthaChristhu Raj, , Assistant Professor, Electronics and Instrumentation, Karunya University
K.Gerard Joe Nigel, Assistant Professor, Electronics and Instrumentation, Karunya University

II. DEVICE COORDINATION IN WEB-BASED MUIS

General Webservers which are developed for general purpose computers like NT servers, work stations typically require a fast processing and giga bites of memory.

A webserver can be embedded into non-internet devices unlike the general purpose computers to provide remote access to the device from the browser if the resources required to the webserver are reduced. As the typical reduction of these resource makes portable set of codes that can be embedded into a controller which got a limited amount of resources. The embedded server can be used to serve the web documents that contain both the static and dynamic data contents that have been acquired from the device which has to be managed and monitored. Embedded servers are also used to send the user commands from a browser to the system and this sate information or data is extracted from the system and the control command will be executed in the system application. Moreover, data coming from system can be used to generate dynamic HTML pages by CGI (common gateway interface) scripts. Therefore, any device connected this way can be controlled and managed through CGI script and the outcome can be sent to the user's browser as a web page. Not only user, but also manufacturer can use web access to update the system information and regular maintenance of the appliances that are interfaced without interrupting the user.

Web-based interfaces are cross-platform of embedded and networking and easier to develop. Most of the application devices have many unused parameters that users find difficult to use, thus causing a poor interface. Thus web based management user interface (MUI) enables the manufacturer to economize on user interface devices (buttons, displays) and feedback given to the user after respective action is satisfactory. On the other hand, with the exponential growth of the web users globally this makes the web-based MUI's more flexible and reliable technology. At this point, it should be indicated that providing user interfaces as web pages facilitate easy design and maintenance of well-designed user interfaces based on web pages and associated design tools.

This paper focuses on infrastructures for using web documents as a UI's for medical field and hospital management. The benefits of embedding a web server into a controller can be summarized as follows:

- Users can interact with application using a system which contains a web browser.
- Controlling, monitoring and updating can be done from anywhere in the world.
- Manufacturer support thought the product life cycle.
- Software updating or extension of user interface.

III. OVERALL SYSTEM ARCHITECTURE

The proposed EWS's monitoring system complies the architecture depicted in fig. 1

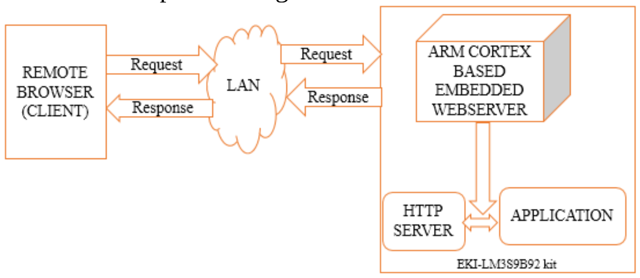


Fig. 1: Overall System Architecture

As shown in fig. 1 the system contains two main areas: remote browser, server area in which it contains the controller, file system, application interface devices. As the client request the web page through the network the request will be received and processed by the controller. In response the dynamic web-page will be generated in the client's web browser from the servers file system. The EWS is connected to the network via switch or an access point depending on the communication (either wired or wireless).
The monitoring system is a web documents (or HTML pages) generated by the server, which are dynamically updated by common gateway interface (CGI) script. Thus user can monitor and control the system terminal online.

A. *System Flow Model*

The development of the embedded web server requires the research on compiler, TCP/IP, lwip web server, HTTP server, configuring the hardware using the board support packages, configuring the compiler. The porting the web server is done by the IAR Embedded work bench. The below figure shows the system flow model for the EWS's.

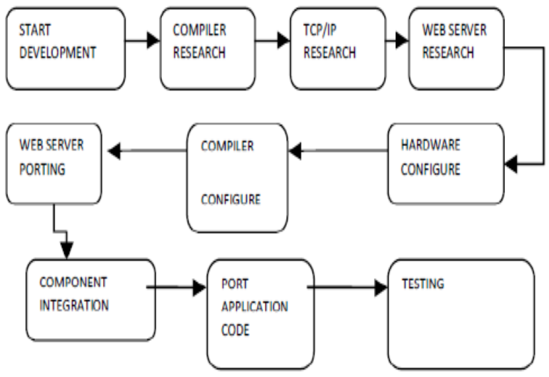


Fig 2: system flow model

The components are integrated together to form the Embedded web server and both web server and application program is ported on the target board (ARM CORTEX-M3).

IV. HARDWARE DESIGN OF THE EWS

EWS uses the ARM Cortex-M3 Processor Core with high performance of 80 MHz operation and flashes with 256 KB single cycle flash memory. The advanced communication interface that can be used is Ethernet MAC and PHY. The ARM Cortex-M3 processor provides the core for low cost platform that meets the minimal memory implementation requirement and low power consumption with outstanding computational performance

A. *Ethernet Interface*

To interface an Ethernet, boot loader should communicate using BOOTP and TFTP protocols with it. By using these standard protocols there will be a co-existence between boot loader and normal Ethernet environment. The bootstrap protocol which is a predecessor to the DHCP protocol, is used to find out the IP address of the client, server. Bootstrap protocol uses UDP/IP packets for the communication, where boot loader is treated as client. First, client sends a BOOTP request and in response server replies (broadcast message) by informing the IP address of both server and client. Thus, it will complete the BOOTP protocol. Then TFTP (trivial file transfer protocol) is used to transfer the web document from server to client. It also uses UDP/IP packets to communicate between client and the server, whereas boot loader acts as client in this protocol. As each block of data is received it is ported to flash. Once all the blocks of data are received and ported to flash, the device is reset, which causes it to start running a new web document.

V. EWS PROCESS STRUCTURE IMPLEMENTATION

The Embedded web server is a finite state machine (FSM), which processes an HTTP request as a step by step sequence. The below figure shows the finite state machine for the embedded web server.

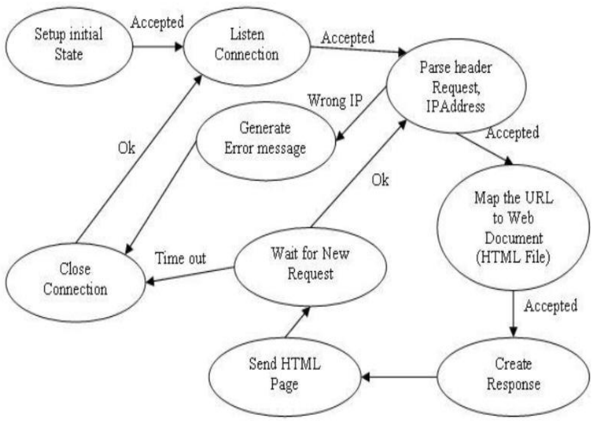


Fig 3: Embedded Webserver Process Structure

In order to support multiple connections in a single thread environment, multiple finite state machines should run in a scheduling manner within a system that uses a lightweight task structure. It consists of a pointer pointing to the function being run and a variable holding the state in the FSM. A state of the system is indicated by flag status, whether the finite state machine can be at run or blocked state. The scheduling system allocates an available FSM for an accepted connection, checks each finite state machine status, to see if it is blocked or running and moves the finite state machine one step, if it is running.

Each state in a finite state machine can check for the presence of data that is ready to be processed at the entry point. If no data is available and the system is not ready, then the finite state machine can block (or stay in wait) itself until data arrives. When data becomes available at the starting point, the finite state machine can then be in run state so its handler can perform the task of state, and the result will be used by the next state by changing the state flag and pointer to the handler.

VI. SOFTWARE IMPLIMENTATION

The embedded webserver requires a minimal set of software like application software, LwIP TCP/IP stack and driver for communication hardware. The use of Internet allows the embedded system to offer sufficient online capabilities without using system resources.

Application software is implemented by using the IAR workbench, where all the libraries, drivers that are required and webpage are called in the program. The web page is created using the general procedure (i.e., text document is saved in html format). In the webpage creation the CGI parameters are called, such that whatever data that is processed and stored in those parameters will be retrieved by the browser. Finally entire web page is converted in to hex file and it is called as header file in the main program. There are tools that can be used to convert the HTML in to .h file.

CGI (common gate way interface) script is a bridge between the controller and the client. It contains the list of parameters that are to be monitored and controlled by the client. While executing, it searches the list of parameters passed to a CGI handler and returns the index of a given parameter within that list. It determines whether or not it is a valid hexadecimal digit. Then it encodes a string for use within an HTML tag, escaping non alphanumeric characters.

While the cgi parameters are called in the web page (front end), it searches the list of parameters passed to a CGI handler for a parameter with the given name and, if found, reads the parameter value as a decimal number. If the string is found, the corresponding parameter value is read from array and checked to make sure that it is a valid decimal number. If so, the number is returned. If any error is detected, the parameter error will be written true and 0 will be returned.

VII. RESULT

A. *Unit Testing*

The various units or modules of the system is tested and integrated. Initially the ARM-Cortex kit is tested for its proper working of the code using the HyperTerminal (Here PuTTy similar to HyperTerminal is used). The kit is connected to the PC using the serial cable and the power guard of the kit is connected to the supply. The Kit will be programed, i.e., the hex file that is created, will be downloaded onto the kit. Now, the HyperTerminal will communicate with the kit using the COMx port. And when a connection is established to the kit through the HyperTerminal we can see the messages sent by the kit to the HyperTerminal. These messages are from the server and the application code displayed from the kit. Thus the embedded web server could deliver the static information contained in it.

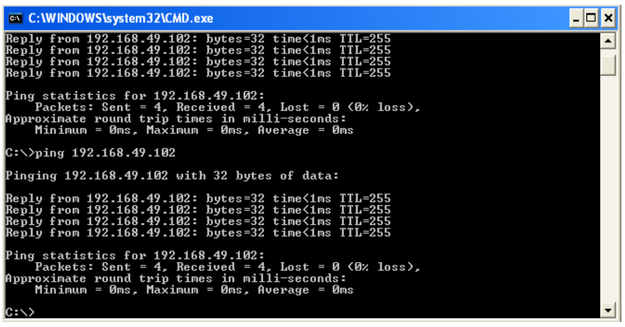
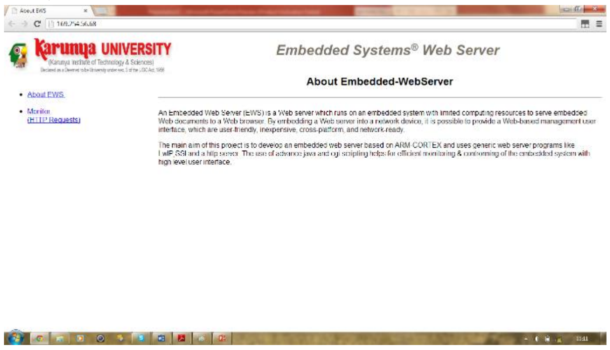

```
C:\WINDOWS\system32\CMD.exe
Reply from 192.168.49.102: bytes=32 time<1ms TTL=255
Reply from 192.168.49.102: bytes=32 time<1ms TTL=255
Reply from 192.168.49.102: bytes=32 time<1ms TTL=255
Reply from 192.168.49.102: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.49.102:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 192.168.49.102

Pinging 192.168.49.102 with 32 bytes of data:

Reply from 192.168.49.102: bytes=32 time<1ms TTL=255
Reply from 192.168.49.102: bytes=32 time<1ms TTL=255
Reply from 192.168.49.102: bytes=32 time<1ms TTL=255
Reply from 192.168.49.102: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.49.102:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

Fig: Pinging the Target

Ping is a computer network tool used to test whether a particular host is reachable across an IP network; it is also used to self-test the network interface card of the computer. It works by sending ICMP "echo request" packets to the target host and listening for ICMP "echo response" replies. Ping estimates the round-trip time, generally in milliseconds, and records any packet loss, and prints a statistical summary when finished. The web server serves the client with a webpage at the ip address 169.254.56.68 which is shown below:


About EWS
169.254.56.68
Karunya UNIVERSITY
(Karunya Institute of Technology & Sciences)
Declared as a Deemed to be University under sec. 3 of the UGC Act, 1956
Embedded Systems® Web Server
About Embedded-WebServer
- About EWS
- Monitor (HTTP Requests)

An Embedded Web Server (EWS) is a Web server which runs on an embedded system with limited computing resources to serve embedded Web documents to a Web browser. By embedding a Web server into a network device, it is possible to provide a Web-based management user interface, which are user-friendly, inexpensive, cross-platform, and network-ready.

The main aim of this project is to develop an embedded web server based on ARM-CORTEX and uses generic web server programs like LwIP,SSI and a http server. The use of advance java and cgi scripting helps for efficient monitoring & controlling of the embedded system with high level user interface.

Fig.: Webpage served by web server

B. *Observations*

The arm cortex-M3 will process the data received from the serial port and place it on the file system (i.e., webpage), such that the client can receive that in a real time bases. Here the CGI script will help to make bridge connection between the client and the system to be monitored, by sending data to and fro. Temperature sensor is interfaced and can be monitored to maintain a certain temperature of the controller for avoiding accidents.

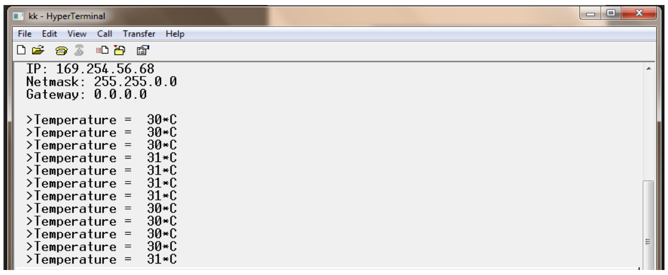
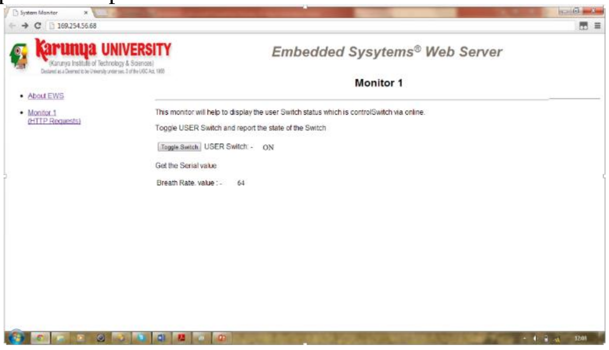

kk - HyperTerminal
File Edit View Call Transfer Help
```
IP: 169.254.56.68
Netmask: 255.255.0.0
Gateway: 0.0.0.0

>Temperature =  30•C
>Temperature =  30•C
>Temperature =  30•C
>Temperature =  31•C
>Temperature =  31•C
>Temperature =  31•C
>Temperature =  31•C
>Temperature =  30•C
>Temperature =  30•C
>Temperature =  30•C
>Temperature =  30•C
>Temperature =  31•C
```

Fig: Result of HyperTerminal showing IP address & Temp.

There is a toggle switch icon on the web page that helps the user for toggling the connection between controller and the bio medical device. This improves the control flow of data and gives flexibility for the user to receive and monitor the required samples.


System Monitor
169.254.56.68
Karunya UNIVERSITY
(Karunya Institute of Technology & Sciences)
Declared as a Deemed to be University under sec. 3 of the UGC Act, 1956
Embedded Sysytems® Web Server
Monitor 1
- About EWS
- Monitor 1 (HTTP Requests)

This monitor will help to display the user Switch status which is controlSwitch via online.

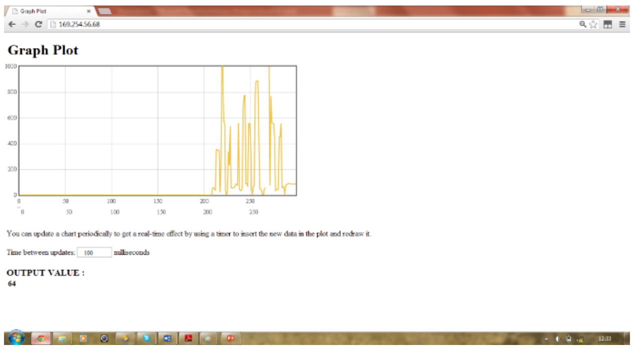
Toggle USER Switch and report the state of the Switch

Toggle Switch USER Switch: - ON

Get the Serial value

Breath Rate. value : - 64

Fig: Output page from webserver



You can update a chart periodically to get a real-time effect by using a timer to insert the new data in the plot and redraw it.

Time between updates: 100 milliseconds

OUTPUT VALUE :
64

Fig: Graph plot

The plot of real time data obtained through biomedical device is shown on the web page. The graph is plotted dynamically and the time update of the graphical plot can be reduced or increased.

VIII. CONCLUSION

In this paper the design and implementation of embedded web server which delivers the real time data of biomedical equipment and in further the machine to machine concept can be improvised to retrieve multiple data. The most important advantage is its user-friendliness and its suitable for both clinical and home environment applications.

REFERENCES

1. Joby Antony, Sachin Sharma, Basanta Mahato, Gaurav Chitranshi, “Distributed data acquisition and control system based on low cost Embedded Web servers”, International Journal of Instrumentation, Control & Automation (IJICA), Volume 1, Issue 1,Pg no. 53-56, 2011
2. Manivannan M, Kumaresan N “EMBEDDED WEB SERVER & GPRS BASED ADVANCED INDUSTRIAL AUTOMATION USING LINUX RTOS ” International Journal of Engineering Science and Technology Vol. No.2,Issue No.11,Pg no 6074-6081, 2010
3. Yilhyeong Mun, Dongsub Cho “Users Access Discrimination and Remote Control Study of Embedded System using Mini Web Server” International Conference on Advanced Language Processing and Web Information Technology, Vol. No.2,Issue No.08,Pg no 341-346, 2010
4. Rui Li, and Xiang Qiang Xiao,“Application Research of Embedded Web Technology in Traffic Monitoring System”, Proceedings of the Second Symposium International Computer Science and Computational Technology, ISCSCT, Pg no.94, 26 December 2009.
5. V.Billy Rakesh Roy, Sanket Dessai, and S. G.Shiva Prasad Yadav, “Design and Development of ARM Processor Based Web Server”, International Journal of Recent Trends in Engineering, Vol. No.1, Issue No. 4, Pg.No. 94-98., May 2009.
6. F. Fioretti, S. Pasqualini, A. Andreoli, P. Pierleoni, “Permanent Switchboard Monitoring using Embedded Web Server” International Conference on Renewable Energies and PowerQuality (ICREPQ), Vol. No.1, Issue No. 1, Pg.No. 01-06., 17th April, 2009.
7. Guangjie Han, Deokjai Choi and Tam Van Nguyen, “A Lightweight Embedded Web Server For non-Internet Devices” , International Conference of Network and Computer Applications, Issue No. 5, Pg. No. 01-05., Sept 2007 (IEEE)
8. M. Can Filibeli, Oznur Ozkasap, M. Reha Civanlar, “Embedded web server-based home appliance networks”, Journal of Network and Computer Applications, Vol. No.2, Issue No.30, Pgs. no 499-514, 2007
9. Yanzheng LI, Shuicai WU, Jia LI, Yanping BAI “ The ECG Tele-monitor Based on Embedded Web Server”, International Conference on Bioinformatics and Biomedical Engineering, Digital Object Identifier : 10.1109/ICBBE.2007.196, Pg no. 752 - 755 , 8 July 2007.

AUTHOR PROFILE

GOPI KRISHNA S, PG scholar, from karunya University, completed Bachelor of Technology in Electrical and Electronics in MREC Hyderabad, specialized (M.Tech.) in Embedded systems from Karunya University. His areas of interest are Robotics and Instrumentation.

P.ANANTHA CHRISTU RAJ, Assistant Professor from Karunya University, completed Bachelor of Engineering in Electrical and Electronics in St.Xaviers catholic collge of engineering, Nagercoil and Master of Engineering in Control and Instrumentation in Anna University. Currently pursuing Ph.D, in Energy saving optimization in Anna University. His areas of Interest are Control system and signal processing. He has published 1 international journal and 8 international conferences.

K.GERARD JOE NIGEL, Assistant Professor from Karunya University, completed Bachelor of Engineering in Electronics And Instrumentation in Karunya University,Coimbatore and Master of Engineering in Control and Instrumentation in Sathiyabama University, Chennai Currently pursuing Ph.D in Solar energy optimization in Anna University. His areas of Interest are Control system and Instrumentation. He has published 8 international conferences.