Design and Implementation of an Embedded Web Controller for Automation

D. Rohith¹, G. Mounica², K. Vinod³

¹Assistant professor, Visakha Institute of Engineering & Technology (Narava)  
²,³M.Tech Student, Department of EIE, University College of Engineering, GITAM University

Abstract—Now a days the demand for adding Ethernet connectivity to home appliances has been on rise with the development of powerful processors and network technology. A user friendly interface to access the remote network device is an important issue for the embedded systems. WebPages can be used as a rich but inexpensive user interface and can simplify connections to remote device. By taking this in to consideration, we design an Ethernet based embedded system that satisfies both the stability and reliability demanded by household devices. This embedded system need supply only Web-compatible http-server software and html (hypertext-markup-language) page information. All the hard work is done by an off-the shelf Web browser running on the user terminal, all devices requiring user interaction can be controlled and managed from one device which includes a web browser, such as a personal digital assistant, cell phone, PC, etc. Also, use of webpage based button and display designs reduce the cost of production while making the systems more user-friendly.

Keywords—Web controller, Ethernet based embedded system, BOA Server, HTTP, web pages, CGI scripting, M.PEG streamer.

I. INTRODUCTION

Smart control of household appliances and utilities begins at low end with plug and play radio frequency (RF) and power line modules and relays that can easily interface with a small home server in charge of scheduling and monitoring. At the high end, a home automation setup includes environmental controls, home security systems and many others. Security systems such as lighting that respond to motion sensors, door/window triggers and video surveillance cameras can be tied in to home automation systems, making it possible to activate lights if a window break is detected, in addition to calling the police or other standard security responses. Hence with home automation, one can set a schedule for your devices and just let them run on their own.

Therefore we designed and developed a TCP/IP based embedded web controller (using Linux programming) for home automation system.

The proposed embedded system for home automation acts as web controller that controls peripherals externally connected to the controller. The figure below shows the implemented design, which includes automatic control of lights and other electrical appliances, acquires and monitors the sensor data like temperature and live camera streaming remotely.

II. WEB CONTROLLER

The web controller is designed using a powerful 16/32 bit ARM 9 microprocessor with embedded 10/100 Mbps fast Ethernet on a single chip. For requesting the data or controlling the peripherals the user has to utilize the front end interface i.e. the web page with the help of a browser. In the web browser the user must open the html page. Initially for security purpose the user has to enter the user id and password if the entered user id and password matches then the next html page which has buttons related to various peripherals will pop up.
Now the user can control or request the data from the peripherals by clicking the buttons related to various peripherals. When the user clicks the buttons the request from the html page will be transferred to the browser and the CGI script will perform the respective action i.e. controls the peripherals or fetches the data from the board. After the data is fetched from the board, the CGI script will transfer the data to the browser and the browser will display the information on the html page.

The six buttons in the html page namely fan on, fan off, motor on, motor off, temperature, and video. When the user clicks fan on button the procedure mentioned above will be performed and the respective CGI script will be invokd and the power supply will be provided to the pin to which the fan is connected with the help of relays and when fan off button is clicked the power supply will be removed to the pin using the relay. When the video button is clicked the camera will turn on and the video will be played on the page this is done using the M-jpg streamer.

When the motor on button is pressed, the respective CGI script will be invoked and the power supply will be provided to the pin to which the motor is connected with the help of relays and when motor off button is clicked the power supply will be removed to the pin using the relay. When temperature button is pressed the respective CGI script will fetch the data from the temperature sensor and it will be displayed on the webpage. This allows the user to control and monitor all the appliances and power supplies within the local area using TCP/IP protocol and the CGI script.

Linux platform is used to implement the proposed automation system. The main reason to choose Linux platform is one can directly interact with the kernel and perform various operations, interface with various system on chips. Therefore installation of Linux (ubuntu) in friendly ARM is essential. Initially the Linux OS will reside in NOR flash of MINI2440 where as NAND flash contains the address of NOR flash. NOR contains all files that are to be loaded in NAND. Hence NAND flash just acts like flash memory.

Whenever the IP address of server (MINI2440) is given in a remote PC present in same internet connectivity, the etho file present in eth folder gets activated. In that file the Ethernet settings like IP address, DNS has to be configured.

III. BOA SERVER

Boa was created in 1991 by Paul Phillips. It is an open source web server that has to be installed in friendly ARM. Boa is a single-tasking HTTP server. That means that unlike traditional web servers, it does not fork for each incoming connection, nor does it fork many copies of itself to handle multiple connections. It internally multiplexes all of the ongoing HTTP connections, and forks only for CGI programs (which must be separate processes), automatic directory generation, and automatic file gun zipping. The primary design goals of Boa are speed and security. Security, in the sense of "can't be subverted by a malicious user," not "fine grained access control and encrypted communications".

In BOA folder web server configuration is done i.e. the Ethernet port is given as 80, the root field is given as “www”. It is nothing but a folder that consists of all the programs like html page, CGI scripting, shell scripting for turning on peripherals. The directory index must be given as index.html.The directory index is nothing but whenever the user enters the IP address of friendly ARM board, the web controller provides a corresponding webpage. The routing to html coding is based on CGI scripting.

After installing the libraries we have to set up and configure MINI2440 friendly ARM on ubuntu. The first software we need on our ubuntu system in minicom. Minicom is just like hyper terminal. Install minicom first by giving a command as **sudo apt-get install minicom**.
Now by connecting mini2440 to our system two connection one from the serial port and next from the Type B USB plug must be made to the system. On the friendly arm board the NAND/NOR switch must be placed in the NOR position. Now invoke minicom from terminal using command **Sudominicom**. After giving this command a window shown below will pop up.

![MINICOM Window](image)

**Fig (III) MINICOM Window**

Next one has to configure minicom the port etc use the command **dmesg** this command lists several other messages also we have to closely observe to which port the board is connected.

![Configuring MINICOM](image)

**Fig (IV) Configuring MINICOM**

In the below pop up window one has to change serial device to the port detected. In our case it will be /dev/ttyUSB0 (It may vary from 0 to any number) we can edit that by selecting “A” option. Also one must double check that both software and hardware flow control must be in NO.

![Serial port settings](image)

**Fig (V) Serial port settings**

After making necessary port settings, one must select save set up as dfl. After that exit from minicom and try our first command i.e. **sudo minicom**.

![saving the configuration](image)

**Fig (VI) saving the configuration**

IV. HTTP PROTOCOL

The Hypertext Transfer Protocol (HTTP) is an application-level request/response protocol for sending web content. HTTP is a simple protocol that is based on a TCP/IP protocol stack. HTTP uses TCP (Transmission Control Protocol). 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.
V. CGI SCRIPTING

One of the best things about the World Wide Web is that you can use it to interact with potentially millions of user to obtain and provide different information. Due to the dynamic nature of this information, static HTML pages are not enough. There has to be a way to display dynamic information to those surfing web site based on what they need. CGI is a mechanism that enables you to do just that. It stands for common gateway interface.

After you have your web site all set up and have created some web pages, it’s time to think about making your web site dynamic by setting up CGI scripts on your web server. CGI programs can be written in almost any programming language that will let you either create an executable program or let you interpret it in real time with other program. Some of the languages used to create CGI applications are c/c++, for tan, Pascal, Perl, visual basic, shell etc. The CGI protocol allows you to add custom programs and services to your web server. This protocol acts as an interface between software that you write in a standard programming or scripting language and web server.

VI. MOTION JPEG

MJPEG, for "motion-JPEG", is an encoding technique that simply JPEG-compresses each video frame before transmission. It is a video format in which each video frame or interlaced field of a digital video sequence is compressed separately as a JPEG image. Originally developed for multimedia PC applications, M-JPEG is now used by video-capture devices such as digital cameras, IP cameras, and webcams; and by non-linear video editing systems.

There are several advantages in utilizing this technology, as well as disadvantages. The main advantages of MJPEG are that JPEG-compression is very cheap to do in hardware and it supports almost any size video frame you want to transmit (subject only to an 8x8 tiling restriction, so images may need to be padded in software).

VII. SHELL SCRIPTING

Access to controller’s pins and peripherals can be done by giving commands using shell scripting. Being a Linux user means we play around with the command line. The command line itself is a program: the shell. Shell scripts allow us to program commands in chains and have the system execute them as scripted event, just like batch files in windows. We can define shell as command line interpreter that connects a user to operating system and allows executing commands or by creating text script. They also allowing for far more useful functions like command substitution. Scripts aren’t just invocations of commands either. They are programs in their own right. Scripting allows you to use such as for loops, if/else statements and so forth directly within operating system’s interface.
Every script file is essentially plain text. That doesn’t mean that we can write whatever we want. When a text file is attempted to be executed, shells will parse through them for clues as to whether they are scripts or not and how to handle everything properly.

- Every script should begin with `#bin/bash`
- Every new line is a new command
- Comment lines start with a `#`
- Commands are surrounded by `( )`

VIII. RESULTS

The implementation of chosen embedded automation system has been achieved by means of the embedded web server. A web server provides access to the end devices for the client by uploading web pages as per the client request. When the configured IP address is entered in the web browser, the predesigned HTML web pages gets displayed through which the client can remotely monitor the sensor status, monitor the video and control the peripherals like motor and fan though relays respectively.
IX. CONCLUSION

Therefore a user friendly interface to easy access the remote network device is an important issue for the embedded systems. WebPages can be used as a rich but inexpensive user interface and can simplify connections to remote device. By taking this in to consideration, we designed an Ethernet based embedded system that satisfies both the stability and reliability demanded by household devices. This embedded system need to supply only Web-compatible http-server software and html (hypertext-mark up-language) page information.

REFERENCES

The heading of the References section must not be numbered. All reference items must be in 8 pt font. Please use Regular and Italic styles to distinguish different fields as shown in the References section. Number the reference items consecutively in square brackets (e.g. [1]).

[5] Embedded Ethernet And Internet Complete by Jan Axel son
[6] Custom CGI Scripting with Perl [Kevin Hanegan]