International Journal of Scientific Research in Computer Science and Engineering
Case Study Vol-1, Issue-3 ISSN: 2320–7639

Case Study: Digital Advertisement Board

Saurabh S. Patel and Sunil N. Kore

*1 *Department of Electronics Engineering, Walchand College of Engineering, Sangli*
2 *Department of Electronics Engineering, Walchand College of Engineering, Sangli*

Available online at www.isroset.org

Received: 26 March 2013 Revised: 30 March 2013 Accepted: 29 May 2013 Published: 30 June 2013

Abstract— We are in the era of technology, everyone is getting techno-savvy. Everything surrounding us is getting compact, from our telecommunication device to our diaries. Most of them are turning into software plus hardware form like e-newspaper, electronic diaries, etc. Only one thing is still ignored from government sector to private sector, which is nothing but advertisements or notices. However it is seeing transformation in recent times, which still needs a computer with display device. We will try to design a low-cost notice board or advertisement display device using development boards like Raspberry-Pi(R-Pi) which is cheapest than its counterparts. The Debian Wheezy OS supported by R-Pi support full version Java-SE Java Virtual Machine (JVM). We will use java programming for communication between Advertisement Board and a remote server which can manipulate the notices currently being displayed. The notices must be in video form that to encoded by H.264 codec. The server will have GUI for communication with notice board.

Keywords- Digital Advertisement Board; Raspberry-Pi; Socket Communication

I. Introduction

Present world is recognized as information age. Proper information reaching to the user in time is a key to the good governance. Particularly in the institution, government office, public utility centre, etc., information or circular or notices should be available for longer period. An interactive digital notice board is requirement of the institution and public places. This board should also facilitate public to get the information on storage devices. This is actual key solution to the queries raised through Right to Information Act. Such solution will definitely save lot of paper and ink too.

II. Motivation

The advertisement boards have many constraints regarding the notices to be pinned on it. Also there is limit to number of notices we put on notice boards. Figure 1 indicates the present scenario of notice board. There are also some other shortcomings which come with the current notice boards

- Every time we need to display notice we have to take the permission from the concern authority.
- Due to ever flooding notices on notice boards many times people ignore important notices
- Reaching every notice in school or colleges is impossible and it is quite hectic too.

The need of interactive notice board is very obvious from the above discussion

Corresponding Author: *Saurabh S. Patel*

Figure: 1 A advertisement board filled with adv.

The digital advertisements are already present in institutes, which comprises of LCD Screen displaying notices which is connected to a computer. But allocating a full PC just for an advertisement board is a clever and economical way?

III. Objective

We proposed to create a cheap and efficient; web enabled advertisement board using SoC (System on Chip). Figure 2 indicate the block diagram. SoC board will have facility of SD/MMC card interface, USB ports. It is proposed to interface LCD or Touch screen to SOC board. The USB ports may be used as USB client port through which user can take the backup of notices or upload notices using USB Flash drives. SD/MMC card contain OS and notices to be displayed. We will use Ethernet port for LAN connection. If the processor can support Linux based OS like Android or Raspbian or OS like Windows-CE, the advertisement board will be more featured. One of the ways would be to develop a server-client communication program to handle advertisement to be displayed at advertisement boards.

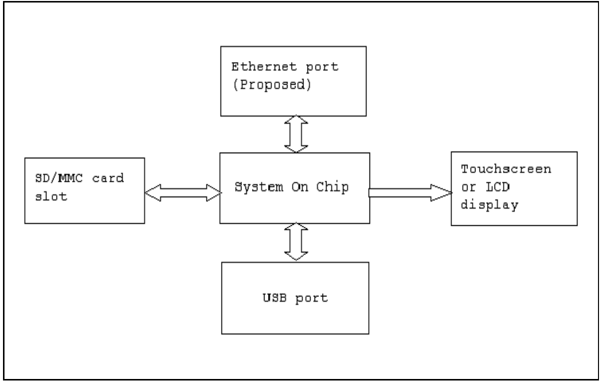

Figure 2. Block Diagram

The requirements of the System on Chip board would be as follows

- SD/MMC card slot
- Ethernet port for LAN connectivity
- HDMI/DVI/VGA port for display
- Wi-Fi module for wireless LAN connectivity(optional)
- Onboard extension to make board interactive by providing navigation buttons(optional-if LCD is used)
- USB port for mouse/touchpads (optional-if LCD is used)

IV. Programming Logic

We can use Socket communication using Java for communication between PC which will run server program and remotely located SoC which will run client program.

A. Server Program

The server program will be running on a computer. The program will allow us to send advertisement to a client, list the notices that are currently being displayed on a client and delete notices remotely. We can create a user friendly GUI for server-end. If we want to add a advertisement to a client then we need to send that file directly from server to client by selecting an advertisement file and pressing SEND button. If the concern authority want to see the list of notices which are being displayed by a client, that can be listed by a pressing a LIST button. Also, if a notice is obsolete then it can be removed from a client from server computer directly by entering the name of notice and pressing the DELETE button.

B. Client Program

The client program will be running on a development board. It will display advertisements which are mainly in the video form. The development board we propose is Raspberry-Pi(Model-B), which comes at very low price. The program would keep on looping videos infinitely and simultaneously will check for request from server like list of files, save a file or delete a file. The other way could be using Android OS on development board and creating an android application for client.

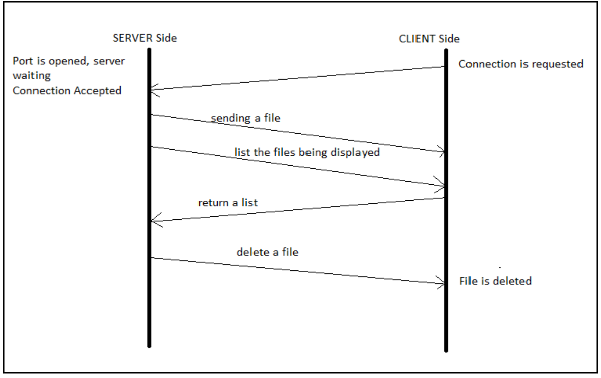

Figure 3. Flow of program

V. Hardware

Figure 4. Raspberry-Pi Model B

A. Specification

- SoC – Broadcom BCM2835
- CPU – 700MHz ARM11
- USB 2.0 – 2
- Video outputs – Composite RCA, HDMI
- Audio outputs – 3.5 mm jack, HDMI
- Onboard Storage – SD/MMC card slot
- Low-level peripherals – GPIO pins, SPI, I^2C, UART
- Power ratings – 700mA(3.5 Watt)
- Supported OS'es – Debian GNU/Linux, Arch Linux

B. Installing OS on SD card

You can either use SD card directly or you can put your MMC card in adapter and insert it in SD card slot. We need to format SD card before installing OS on it.

There are several distros available for Raspberry-Pi. The popular one is Raspbian wheezy. But it does not support Java due its Hard float architecture. Java support is extended to only soft float architecture so far. For java related support Debian Wheezy distro is used. Debian Wheezy support full Java-SE edition. The below written method is for Windows OS. For Unix/Linux you need to go through command line.

1. Download Debian Wheezy distro and Win32DiskImager from www.raspberrypi.org
2. Extract both files
3. Insert the formatted SD card in laptop
4. Run Win32DiskImager, provide path of extracted image of distro and select device as SD card. Press Write

C. Resize the partition to fill the SD card

The SD card image is sized for a 2 GB card. So, if you are using an SD card with a greater capacity, you may find that only 2 GB is available. If this is the case, then to gain more free space, the partitions must be resized. The Fedora Remix and the BerryBoot will automatically resize the partitions on the mounted card during the _rst boot. The Debian, Open SuSe and Rasbian images won't, so you will have to do it manually.

D. Installing Java on Debian Wheezy

Download the Java-SE JRE (Java Runtime Environment) tar file for Raspberry-Pi from Oracle's website. Using ssh(in case of Linux machine) or PuTTY(in case of Windows) send the downloaded file to Raspberry-Pi. Extract the file at a particular location. Now you may delete the tar file of JRE, if you want. The detail information for installing JRE is as follows

1. Set the root password for R-Pi
2. Enable the Raspberry Pi for SSH remote network connections. By default it is enabled.
3. Use scp command (Linux) or WinSCP(Windows) to transfer _le from your PC to raspberry-Pi. You can also use PuTTY on Windows.
4. Now we need to extract the tar file and include the path of java bin folder.

VI. Running the client from remote machine

First of all you need to run the server program which will open port. Now you can run the client program on R-Pi remotely in following way

1. Either you can connect using ssh(linux) or PuTTy(windows).
2. Other way would be installing tightvncserver on R-Pi and xtightvncviewer on remote host(if Linux) or Tight-vncviewer(if Windows)

References

[1] www.raspberrypi.org

[2] www.oracle.com/technetwork/articles/java/raspberrypi-1704896.html#Raspberry