Large Screen Wireless Notice Display System

Yash Teckchandani, G. Siva Perumal, Radhika Mujumdar, Sridhar Lokanathan
Department of Instrumentation
V.E.S. Institute of Technology, Mumbai University
Mumbai, India
Email: teckchandani.yash@gmail.com

Abstract - **Wireless electronic notice boards are a faster alternative to conventional pin-up type notice boards. A major constraint of the methods used so far is the small size of the 16x2 Liquid Crystal Displays (LCD) used to display the notices. This paper proposes a method in which large screens like computer monitors or televisions can be used for displaying notices sent as text messages from a mobile phone. The proposed method uses HyperText Markup Language (HTML) to present the output since it offers many customisation options. To give high resolution output, the credit-card sized computer Raspberry pi has been used. This HTML output can be displayed by a web browser running on the Raspberry pi. The notice to be displayed is sent as a Short Message Service (SMS), which is received by a Global System for Mobile Communications (GSM) modem, making the reception of the message wireless. The GSM modem is polled at regular intervals by the Raspberry pi to display the latest messages.**

Keywords— Wireless notice board, Raspberry pi, HTML, SMS, GSM modem.

I. INTRODUCTION

Notice Boards are an important medium for displaying information and keeping people informed. The traditional notice boards involve the pinning up of printed or handwritten information on a board. But this has the disadvantages of dependency on a person for pinning up notices and wastage of paper. Some developments in notice boards, in an attempt to overcome above-mentioned drawbacks, include display of data on a screen using wireless communication. This has been implemented on Liquid Crystal Displays (LCD) and Light Emitting Diode (LED)[1] displays. Some of the available methods use ATmega32[2], LPC2148 by NXP [3]. The method used by Darshankumar C. Dalwadi.et al [2] can display only one message at a time. In the method used by Nivetha S. R.et al [3], 16x2 character LCD has been used. The disadvantage of this system is that in order to view the message, the observer should be very close to the screen. Use of Field Programmable Gate Array (FPGA) for notice boards is not economically viable [4]. Also, it requires synthesizing MicroBlaze processor on the FPGA for sending Attention (AT) commands [5]. Thus, the parallel processing capability of the FPGA is not used as it is made to work sequentially.

This paper, with an aim to increase the usability of electronic notice boards, deals with wireless reception and display of messages using Raspberry pi. Practically, all output resolutions are supported. The font size is customisable and it can display multiple notices at a time. The paper focuses on using HyperText Markup Language (HTML) for displaying messages since using HTML, display features such as font size and background colour can be easily set as per user requirement. The reason for using Raspberry pi is that it has web browsers such as Midori, which can be used for reading HTML files.

Section II talks about the proposed method. The implementation methodology is explained in Section III. Section IV gives the conclusion and Section V talks about the future direction of the proposed work.

II. PROPOSED METHOD

This section gives a basic overview of the system. Fig. 1 shows the block diagram of the system.

- The notice to be displayed is sent as an Short Message Service (SMS). The Global System for Mobile Communications (GSM) network is digital. This makes it immune to noise. Also, GSM networks are relatively free of errors.
- Since GSM network is being used, the notice can be sent from practically any location on the globe and it will be displayed on the screen.
- This SMS is received by the GSM modem and it is stored in the Subscriber Identity Module (SIM) memory. The GSM modem is polled at regular intervals by the Raspberry pi for a new message.
- This is accomplished by the means of establishing serial communication between the Raspberry Pi and the GSM modem.
- The GSM modem has an RS232 port while the Raspberry pi has a USB port. The conversion of signal from RS232 levels to USB compatible levels is done using integrated circuit Microchip MCP2200.
- To interact with computers (Raspberry pi in this case), modems need Attention (AT) commands. The Raspberry Pi sends AT commands to the GSM modem.
- In response, the SMS stored in the SIM memory is transmitted by the GSM modem. The Raspberry pi after decoding the received data sends it to the display.

978-1-4799-7849-6/15/$31.00 ©2015 IEEE

- The Raspberry pi has two video output options – composite (Radio Corporation of America - RCA) and High-Definition Multimedia Interface (HDMI).
- Display screens with VGA port can also be used by using HDMI OUT of the Raspberry pi with a HDMI to Video Graphics Array (VGA) convertor.
- Therefore, the proposed method is versatile with respect to display options.
- The operating system used in Raspberry pi is Raspbian.
- The most popular programming language for Raspberry Pi is Python. It is a high-level language and thus lesser coding effort is needed as compared to using assembly language for other microcontroller boards.
- Thus, the method proposed in this paper has several advantages over the prevalent methods used to offer the same functionality.

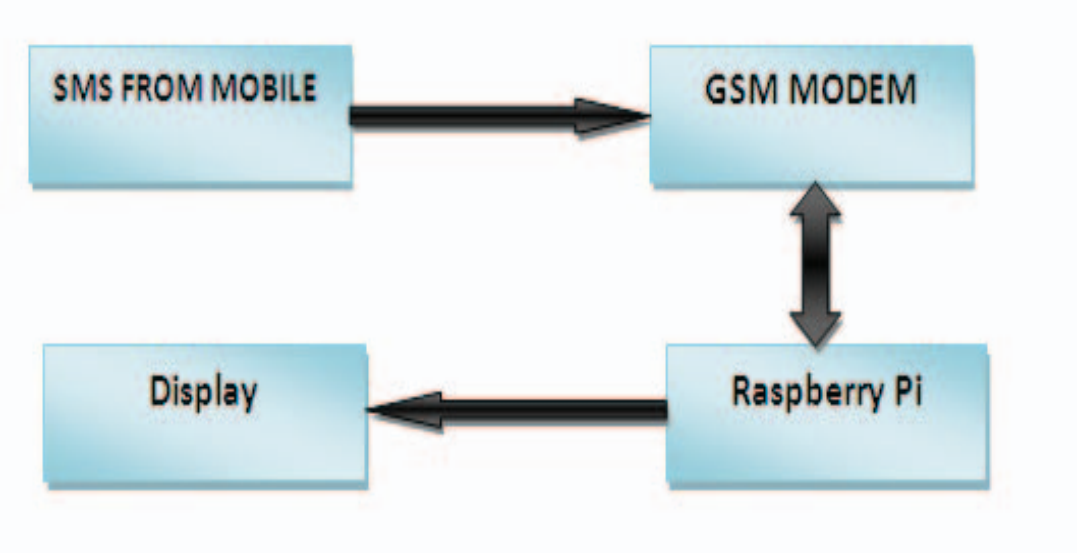

Fig.1. Block diagram of the system

Fig.2. System Assembly

III. IMPLEMENTATION

This section explains the execution flow from establishing serial communication between the GSM modem and Raspberry pi to displaying the message on the screen.

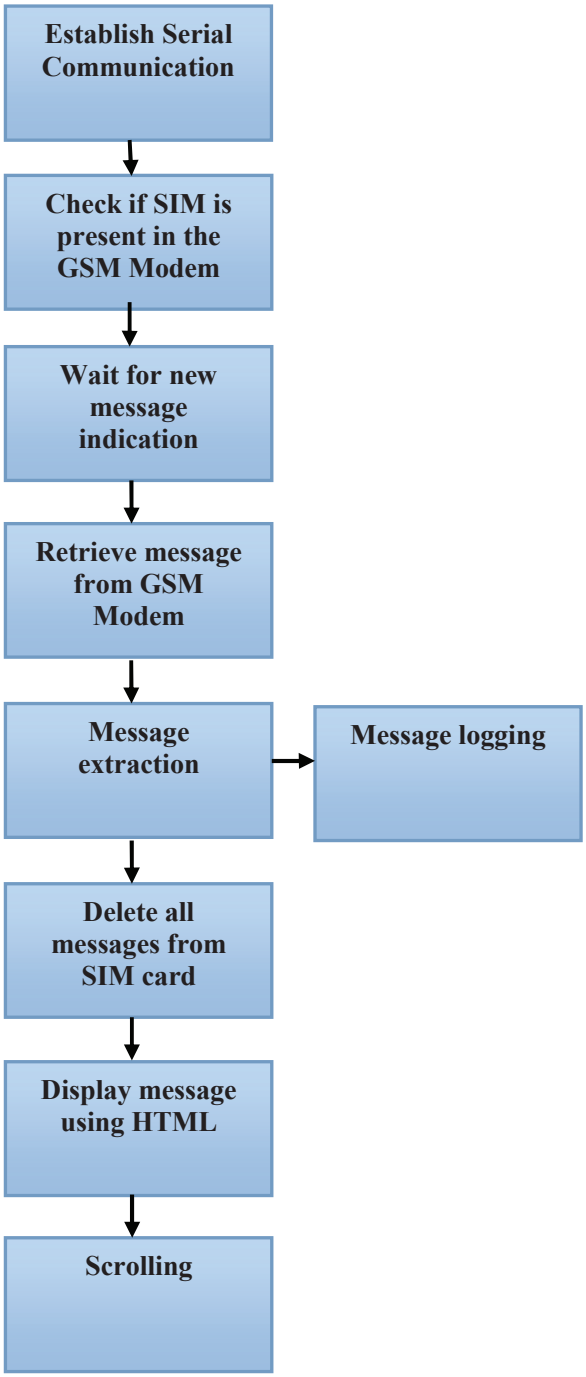

Fig.3 . Implementation flow chart

As shown in Fig.3, first the serial communication is established. Serial library is imported and baud rate is set to 9600. Instructions are given to GSM modem using AT commands. All the AT commands used are listed in Table 1.

The presence of SIM card in the GSM modem is checked using command number 1 of Table I. Fig. 4 shows SMS sent from the mobile phone.

TABLE I. LIST OF AT COMMANDS USED [6]

SR. NO.	AT COMMAND	FUNCTIONS
1	AT+CSMINS?	SIM inserted status reporting
2	AT+CMGL = "ALL"	This command returns messages with a status value
3	AT+CMGD = 0	This command deletes a message

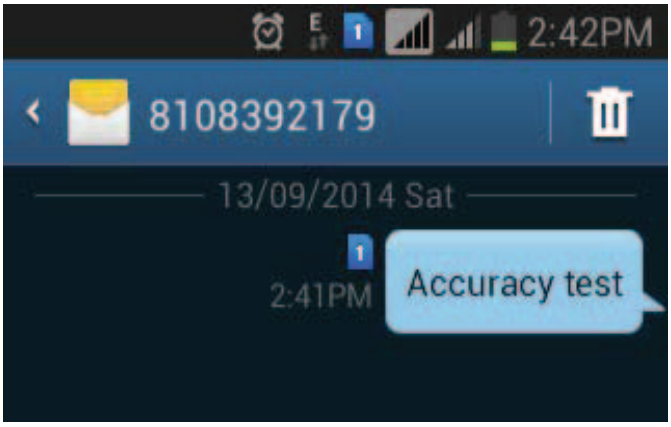

Fig.4. Screenshot showing one message only with time

When a new SMS is received, the GSM modem gives an unsolicited response. After receiving this response, an AT command 2 of Table I is sent. The reply given by the GSM modem to AT command number 2 is shown in Fig. 5. This reply is saved for archival purpose. A log file is created for that. The log file is updated as soon as a reply is received. The log file is shown in Fig. 5.

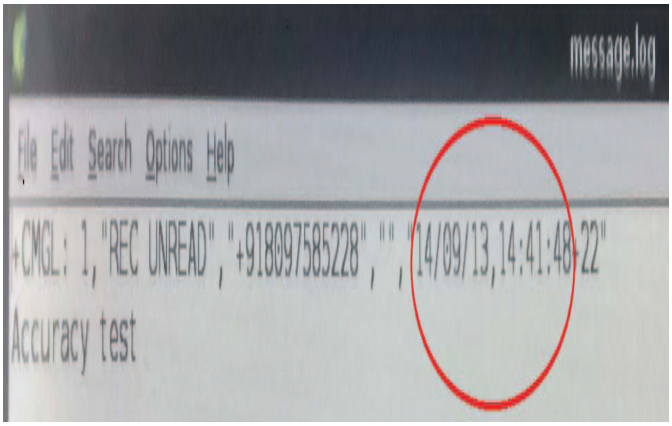

Fig. 5. Message Log highlighting time of receipt of SMS

The reply shown in Fig. 5 is truncated to extract only the SMS. SIM cards have limited memory. To avoid loss of messages received in case the SIM memory is full, the SIM memory is cleared after fetching the SMS. This is done by using another AT command (Table I serial no. 3)

In this paper, the python script has been used to communicate with the GSM modem and HTML has been used to display messages.. HTML files can be viewed using an internet browser such as Midori. In addition, browsers have wrap-around feature which helps accommodate long messages on the display. The header of the HTML file can be predefined. It is "VESIT NOTICE BOARD" in Fig. 6.

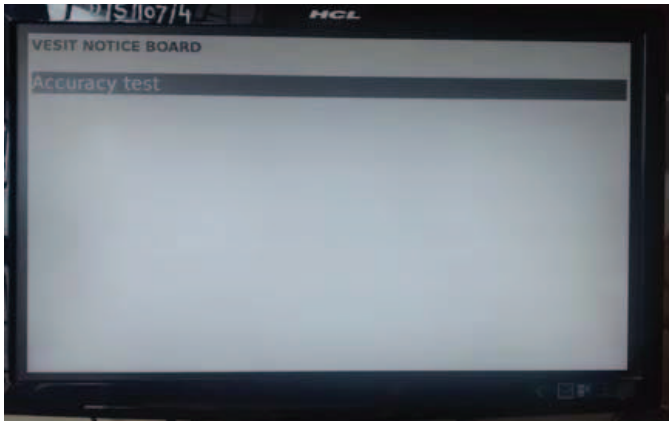

Fig.6. The message displayed

The extracted SMS forms the body of the HTML file. The output would be as seen in Fig. 6. The font used in the body of Fig. 6 is of size 40px and white colour. The background colour is grey. As seen in Fig. 6, the message has been faithfully reproduced. Fig. 6 shows only one message. However, multiple messages can be displayed. Fig. 7 for example shows three messages.

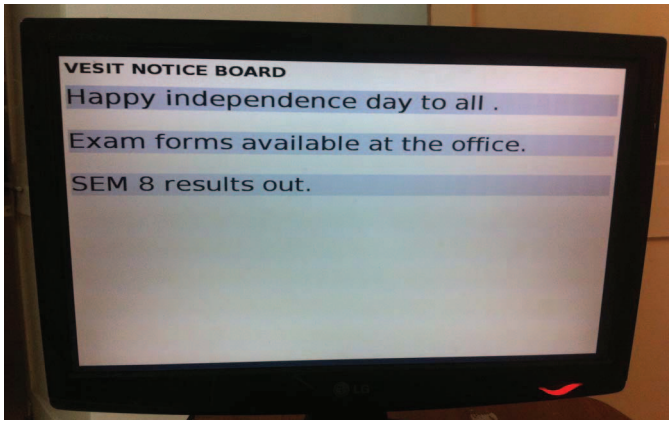

Fig 7. Multiple messages displayed

Now on arrival of a fourth message, the message list scrolls up. As seen in Fig. 8, after the fourth message is received, the message list scrolls up to display the latest message.

In case of power failure, after boot up on resumption of power supply, the browser window should open automatically so that the display screen is ready to show the notice [7]. For aesthetic reasons, the boot messages [8] and the Raspberry pi logo which also appears in the top left corner [9] of the screen can be hidden.

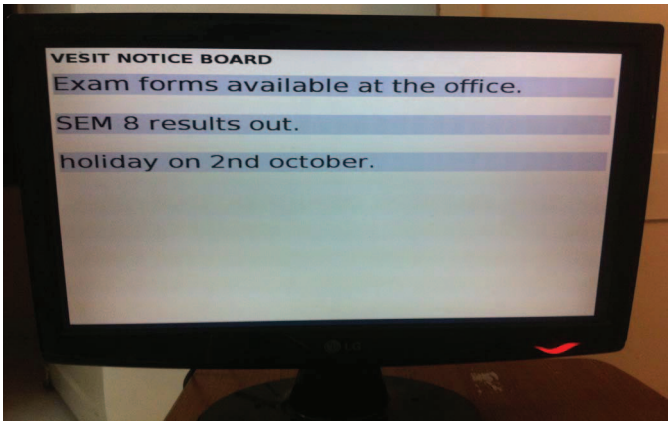

Fig 8. The first message scrolls up on receiving the fourth message

IV: CONCLUSION

This paper presents a way to incorporate messages in HTML script. It offers an edge over other notice boards because of features such as customizable font size, color and background color. The size of the screen, a major limitation of other methods, is overcome by this system.

V: FURTHER WORK

Use of GSM modem limits the application to text messages. However, Raspberry pi is also capable of displaying images as well as playing video and audio. The source of this multimedia content can be a server with whom the Raspberry pi can communicate via the internet. The Raspberry pi has an Ethernet port [10]. It can also be connected to a Wi-Fi network by using a Wi-Fi dongle [11]. This can be done by first saving the notice (text/image/video) to the server. Then, it can be fetched by the Raspberry pi over the internet. This will enhance the capability of the system.

REFERENCES

[1] Article named "Wireless data transmission over GSM Short Message Service" retrieved on 11 June 2014 from http://www.eacomm.com/downloads/products/textbox/wdtgsm.pdf

[2] Darshankumar C. Dalwadi, Ninad Trivedi , Amit Kasundra "Wireless notice board our real-time solution" National Conference on Recent Trends in Engineering & Technology.

[3] Nivetha S. R, Pujitha. R, Preethi Selvaraj & Yashvanthini S.M "SMS based Wireless Notice board with monitoring system" International Journal of Advanced Electrical and Electronics Engineering, (IJAEEE) ISSN (Print) : 2278-8948, Volume-2, Issue-3, 2013.

[4] Price of Basys™2 Spartan-3E FPGA Board retrieved on 13th August 2014 from http://www.digilentinc.com/Products/Detail.cfm?Pr od=BASYS2&NavTop=2&NavSub=649&CFID=18448141&CFTOKEN=9715b5fba0c2d007-574B45C2-5056-020102E8A5D7F3307D85

[5] P. K. Gaikwad "Development of FPGA Microblaze processor and GSM based heart rate monitoring system" International Journal of Computer Science and Mobile Applications, Vol.1 Issue. 3, September-2013, pg. 24-29 ISSN: 2321-8363

[6] SIMCom SIM900 AT Commands Manual Version 1.11 retreived 24 July 2014 from http://www.seeedstudio.com/wiki/images/7/72 /AT_Commands_v1.11.pdf.

[7] Article titled "Autorun browser on startup" retrieved 13 August 2014 from http://www.raspberry-projects.com/pi/pi-operating%20systems/raspbian/gui/auto-run-browser-on-startup

[8] Article titled "How to hide text on boot" retrieved on 20 September 2014 from http://raspberrypi-easy.blogspot.in/2013/12/how-hide-text-on-boot.html

[9] Article titled "How to hide Raspberry Pi LOGO on boot" retrieved on 20 September 2014, 11:30 A.M. from http://raspberrypi-easy.blogspot.in/2013/12/how-to-hide-raspberry-pi-logo-on-boost.html

[10] Article titled "RPi Hardware" retrieved on 27 November 2014 9:32 A.M.from http://elinux.org/RPi_Hardware

[11] Article titled "WIFI" retrieved on 27 November 2014 9:45 A.M. from https://www.raspberrypi.org/documentation/configuration/wireless/