ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DESIGN AND IMPLEMENTATION OF EMBEDDED THIN CLIENT FOR VNC

SAMAN FATAHPOUR
Department of Computer Engineering, Kermanshah branch, Islamic Azad University, Kermanshah, Iran
E-mail: saman.fatahpour.2016@ieee.org

ABSTRACT

In this paper, a general method to design and implement a thin client for VNC (Virtual Network Computing) is presented. In order to connect the device to network and monitor, a 16-bit network interface called WIZ830MJ is applied. Moreover, a solution to design 24-bit graphic interface, using the SSD1963 chip and a DAC (Digital-to-Analog Converter) known as ADV7125, is introduced which provide users with a resolution up to 800X480 pixels. To enhance the efficiency of the device, the translator software is designed between VNC server and client which allows applying FastLZ compression Algorithm. By modifying the RFB (Remote Frame Buffer) protocol, the designed translator avoids some problems such as "TCP Window Full" and "TCP Retransmission" and reduces the network bandwidth consumption. The device has no need to fast processor and operating system. It has a good performance using an ARM Cortex-M4 processor with working frequency of 100 MHz. This issue is important because VNC software needs an operating system like LINUX, which itself requires a more robust processor and more consumed energy than an embedded device. At the end of the present paper, the designed device is tested in a laboratory and the results are presented.

Keywords: *Thin Client, Embedded Device, VNC, RFB Protocol, Virtual Network Computing*

1. INTRODUCTION

The development of network communications and also the graphical operating systems led to the design of a system which enables users to access another device from a remote distance [1]. Utilizing the RFB protocol [2], VNC enables users to access to a user interface through a remote system based on server and client structure. To use a software client, it is necessary to run a heavy operating system which is more expensive than an embedded system. The main challenge to design a hardware client for VNC is to be equipped with a robust structure which can send and receive data through a high speed network and use compression and decompression Algorithms. Finally, the device needs a high quality screen output with at least a resolution of 640X480 Pixels. The device's support of the partial screen update which is fully described in the RFB protocol by Tristan Richardson is another feature which enhances the system's efficiency, reduces the network's overload and accelerates screen update. In general, the protocols designed for remote access are categorized into two classes as follows:

- Low level image frame Buffers - sends the image itself to the client and it is reconstructed there. RFB protocol commonly used in VNC is of this type.
- High level API display function - sends the images as something else like a set of coordinates so as to dramatically decrease the time used for reconstruction. [3] One example is RDP (remote desktop protocol). [4]

Software clients are devices designed for connecting to VNC server, but the expensive required hardware being used for software client results in high costs. In addition, this hardware has a high energy consumption level. These problems result in a need for a subsequent device that eliminates such problems. The importance of thin client lies in it's not depending on a fast processor and operating system. Our discussion on the design of a device is limited to the hardware design of a client for RFB protocol which is designed by Olivetti Research [5]. This protocol is a pull protocol, indicating that graphical updates are sent whenever the client sends a "FrameBufferUpdateRequest". The part of the frame Buffer containing pixels that are changed since the previous update, is divided into rectangles. A variety of 'flavors' of VNC exist, differing in the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

applied coding on the rectangles [6]. The processor board is STM32F4 Discovery [7] which uses the stm32f407 chip [8]. In designing the graphic interface, a driver chip for TFT LCD called SSD1963 [9] was used which the LCD output was converted into a VGA signal by a DAC named ADV7125 [10]. This structure fulfills the need for the fast processor in designing the system, because SSD1963 prepares the required space to store the pixels and also produces the signals such as HSYNC and VSYNC. In addition, such driver supports drawing a rectangle and is able to also draw RFB's rectangles. The network interface uses WIZ830MJ [11] which is equipped with W5300 chip [12]. This chip provides a memory-like space such as SRAM (Static Random Access Memory), which through reading and writing in the chip memory, it is possible to exchange data by TCP/IP. This chip provides TCP/IP in a hardware form.

2. RELATED WORKS

Daniel Zinca in a paper entitled “design of a modified RFB protocol and its implementation in an ultra-thin client” claims that: “Recently, works were made in order to develop an architecture called ultra-thin client (or zero-client) where no operating system is running on the client hardware, but a minimal TCP/IP stack. The screen contents and the keyboard/mouse activity are sent between the client and the host. The cost can be substantially lower compared to the one of thin clients because energy consumption is even lower, there is neither operating system maintenance nor software cost.” [13] One work done in this field is the design by Daniel Zinca which has been performed by AVR microcontroller and ENC28J60 chip. Another example is the library written by Adam Dunkles for connecting to VNC.

3. SYSTEM IMPLEMENTATION

In this section, implementation of the designed device has been described.

3.1 The Network Interface

W5300 is a 0.18 µm CMOS technology single chip into which 10/100 Ethernet controller, MAC, and TCP/IP are integrated. W5300 is designed for Internet embedded applications where easy implementation, stability, high performance, and effective cost are required. With single chip having TCP/IP protocol and 10/100 Ethernet MAC & PHY the Internet connectivity can be implemented easily and quickly. W5300 supports hardwired TCP/IP protocols such as TCP, UDP, IPv4, ICMP, ARP and PPPoE and so on. In order to achieve high-performance data communications, the data memory for communication of W5300 is expanded to 128K bytes, and the interface to MCU supports 16-bit data bus. Users can use the 16 independent high-speed data communication ports. [14]. The network interface uses a chip designed by WIZnet Company named W5300. This chip has been used in the module WIZ830MJ made by WIZnet Company, and WIZ830MJ is used in the present paper. Based on the datasheet provided by W5300, this chip has UDP and TCP/IP capabilities and its structure is of a hardware type, i.e. TCP stack has been designed in a hardware form. This design significantly reduces the need to fast processors and enhances system's sustainability. The network interface is of a 16-bit type and has an adjustable input and output Buffer ranging from 1 KB to 128 KB, and all the processes related to the network are done in this part without any interruption and involvement of the processor. Based on the W5300 datasheet, connecting wz830mj in the form of indirect address to the STM32F4 Discovery board is reflected in Figure 1.

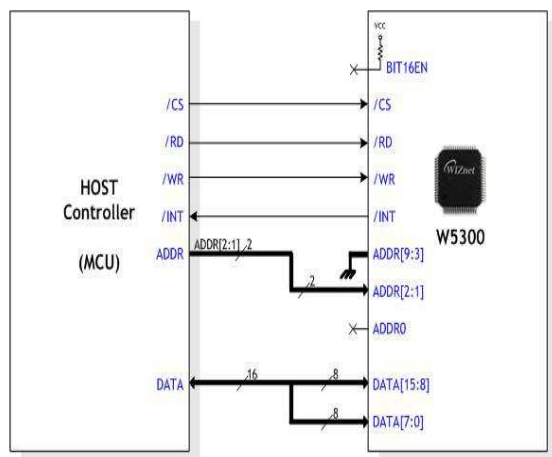

Figure 1: Pin connections for STM32F4 Discovery and WIZ830MJ

3.2 The Graphic Interface

The main task of the graphic interface in such device is to keep pixels' value in a memory space of a SRAM type with high access speed and the frequent display of the pixels, which in turn, lead to the display of image on a monitor screen. The designed graphic interface can provide up to 24 bit for each pixel and also generates maximum 800*480 pixels, and if we consider 60 Hz refresh rate totally generated bits in a second is: 800*480*60*24 = 552960000 bit. The pixels are stored through a 24-bit bus between the processor and graphic interface, and the graphic interface

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

must manage the bits like a two-way high speed Buffer, i.e. it is possible to use the structure just like a SRAM with the input data lines, CS, Reset, R/W, 24-output lines [15] which the design and manufacture of this method has a high cost. But the desired solution used in this device is using a TFT LCD driver chip named SSD1963 which its output has been converted into VGA analog signal by ADV7125 [16]. This method has a lower cost. This chip is also able to update and manage the pixels in a partial from meaning it enables update a part of frame buffer. Based on the datasheet of SSD1963 and ADV7125, the connections for graphic interface must be like what is ordered in Table 1. The outputs, R, G and B of ADV7125, are pulled down through 75 ohm Resistors, and LFRAME, LLINE of SSD1963 must be connected to HSYNC and VSYNC, respectively. Therefore, the connection of ADV7125 to VGA connecter is as suggested in Figure 2. The designed graphic interface is shown in Figure 3.

Table 1: PIN CONNECTIONS FOR SSD1963 AND ADV7125

SSD1963	ADV7125
LD[0-7]	R[0-7]
LD[8-15]	G[0-7]
LD[16-23]	B[0-7]
LSHIFT(DCLK)	CLOCK

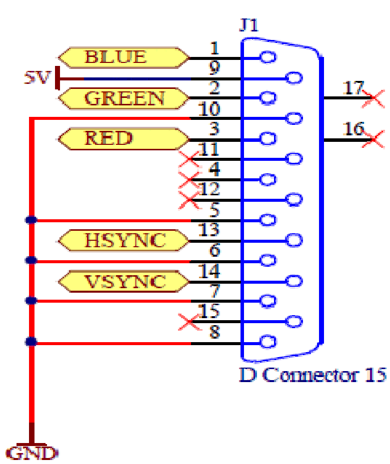

Figure 2: Pin Connections For ADV7125 And VGA Connector

Figure 3: The Designed Graphic Interface

3.3 The Processor

The device uses ARM processor, the series Cortex-M4 named STM32F407 with a working frequency of 100 MHz which is small and with low energy consumption and its working voltage is 5V. Such special device is able to run sophisticated software's using some protocols such as VNC. The processor's board is STM32F4 Discovery. Figure 4 shows the processor board. In general, all parts of the device are managed by processor and a specific bus. By utilizing data compression protocols, the processor also enables data compression and decompression in order to transfer data within the network more quickly.

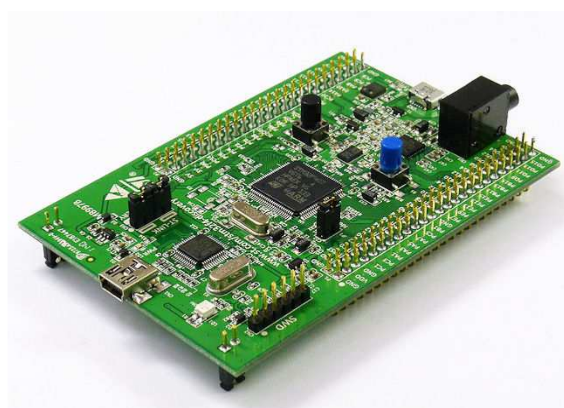

Figure 4: The Processor Board

3.4 The Compression Algorithm

To enhance the pixels' transfer speed, the device uses data compression. Such Algorithm must be quick and suitable for embedded devices. The speed of the compression and decompression of FastLZ Algorithm [17] compared to the zip Algorithm is shown in Figure 5.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

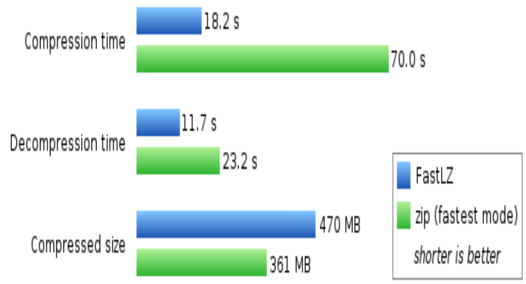

Figure 5: Compression and Decompression Timings for FastLZ and Zip Algorithms (www.fastlz.org)

3.5 The Software Translator

The software translator is a middleware between VNC server and client's device. It is installed on the server and keeps listening on a specific port of the server, and all the requests fed to VNC server by device, are passed through the translator. The most important reason to use the translator is to Prevent overflowing window space in the device's network interface. Detecting the changes in image Buffer, VNC server sends pixels toward client, which this task gradually becomes challenging due to high volume of data and limited memory space of the network interface reception Buffer. The slowness of client compared to server leads to the "TCP Window Full" and "TCP Retransmission" problems [18]. This leads to the network window overflow and a slow transfer speed. The software translator receives the pixels from VNC server and sends them in the form of 1460-byte packets. Then, after receiving the packet by the device, if the received bytes are decompressed successfully using FastLZ Algorithm, the device returns 1 to the translator as a sign of success and the device's network interface Buffer becomes free, otherwise 0 is returned by the device and the translator sends again the packets to the device. This process is continued until a rectangle is completely transferred. This method is important because users cannot utilize a variety of compression Algorithms such as Zlib [19] or Tight [20] on several processors due to the need to more ram space and high cycles for compression and decompression. Taking into account the features of other algorithms, FastLZ is in good consistency with ARM and has already been designed in a runtime compression or decompression form.

4. EXPERIMENTAL RESULTS

The device was tested in the laboratory and Figure 6 shows the final designed device. The test results are reflected in the IO graph of Wireshark (Figure 7). The graph shows the Bandwidth used by the device, the graph has a high level at the starting moment because the first frame Buffer update request returns whole the screen pixels and then it continues by partial screen updates.

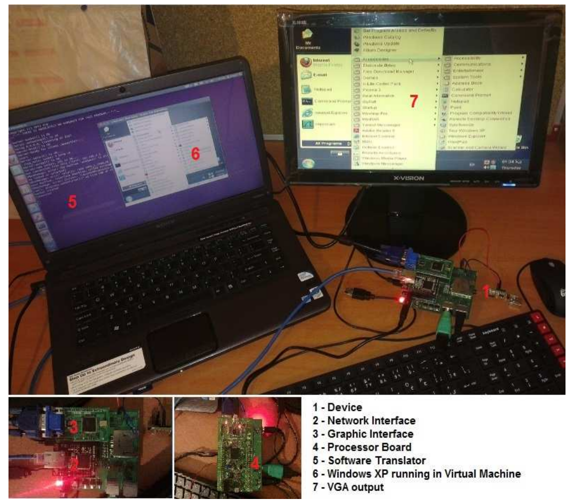

Figure 6: Testing designed Device in The Laboratory

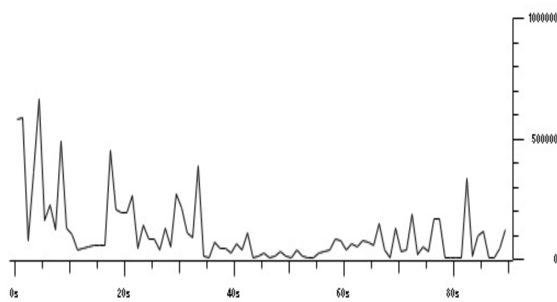

Figure 7: IO Graph of Wireshark for VNC Protocol (Bytes/Tick)

5. CONCLUSION

As it was mentioned before, the software VNC clients were expensive and high energy consumers. The use of energy and high costs of hardware, needed to be eliminated by a new device. In the present paper, a designed device was introduced and examined and some results of the final device performance presented. The device was tested properly in a laboratory, and it was also tested in terms of speed of screen update with a resolution of 640X480 Pixels and 8-bit quality where the 8 bits directly describe red, green, and blue values, typically with 3 bits for red, 3 bits for green and 2 bits for blue, and good results on the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

function of the device were achieved. The design of this device is of significance because it does not use any operating system and does not require a fast processor. In addition, the device takes privilege of all capabilities of a software VNC client.

REFRENCES:

[1] T. Richardson, Q. Stafford-Fraser, K.R. Wood and A. Hopper, "Virtual network computing," IEEE Internet Computing, vol. 2, no. 1, pp. 33-38, 1998.
[2] T. Richardson, "The RFB Protocol," http://www.realvnc.com/docs/rfbproto.pdf, 2010.
[3] N. Thuy An, T. Nguyen, S. Ko and E. Huh, "An Efficient Block Classification for Multimedia Service in Mobile Cloud Computing," in Proc. 2012 ACM Multimedia CBMAS-EH '12, pp. 9-14
[4] "Windows Remote Desktop Protocol (RDP)," http://msdn.microsoft.com/en-us/library/windows/desktop/aa383015.aspx
[5] "Olivetti Research Laboratory," http://en.wikipedia.org/wiki/Olivetti_Research_Laboratory
[6] P. Simoens, P. Praet, B. Vankeirsbilck, J. De Wachter, L. Deboosere, F. De Turck, B. Dhoedt and P. Demeester, "Design and implementation of a hybrid remote display protocol to optimize multimedia experience on thin client devices," in Proc. 2008 Australian Telecommunication Networks and Applications Conference (ANTAC), pp. 391-396.
[7] STMicroelectronics, "STM32F4 high-performance discovery board," STM32F4 Discovery datasheet, January. 2012.
[8] STMicroelectronics, "STM32F407 32-bit Microcontroller," STM32F407 datasheet, May. 2012.
[9] SOLOMON SYSTECH SEMICONDUCTOR, "LCD Display Controller," SSD1963 datasheet, Dec. 2008.
[10] ANALOG DEVICES, "Triple 8-Bit High Speed Video DAC," ADV7125 datasheet, 2011.
[11] WIZnet, "WIZ830MJ," WIZ830MJ datasheet, 2008.
[12] WIZnet, "W5300," W5300 datasheet, 2010.
[13] D. Zinca, "Design of a modified RFB protocol and its implementation in an ultra-thin client," in Proc. 2010 Electronics and Telecommunications (ISETC), pp. 157-160.
[14] Y. Shi, C. Jin and F. Gao, "The solution of ethernet based on hardware protocol stack W5300 and FPGA," in Proc. 2011 Electronic and Mechanical Engineering and Information Technology (EMEIT), pp. 1328-1331.
[15] D. Shi and X. Ye, "Design of VGA Display System Based on CPLD and SRAM," in Proc. 2013 Intelligent System Design and Engineering Applications (ISDEA), 2013 Third International Conference on, pp. 579-582.
[16] R.D. Patil and S.N. Patil, "VGA Board Using ARM Controller," International Journal of Emerging Technology and Advanced Engineering, vol. 2, no. 5, pp. 380-382, 2012.
[17] "FastLZ - lightning-fast compression library," http://FastLZ.org
[18] D. Comer, Internetworking with TCP/IP, 4th ed., : Alan Apt, 1982, p. 750.
[19] "zlib compression," http://zlib.net
[20] K.V. Kaplinsky, "VNC tight encoder-data compression for VNC," in Proc. 2001 the 7th International Scientific and Practical Conference of Students, Post-graduates and Young Scientists, pp. 155-157.