Ceylon Journal of Science: Physical Sciences, 6(1),29-37(1999)

DEVELOPMENT OF CLIENT-SERVER BASED NETWORK CENTRIC REMOTE DATA ACQUISITION SYSTEMS

C.T. OBADAGE AND D.N. RANASINGHE
Department of Statistics and Computer Science, University of Colombo, Sri Lanka
M.K. JAYANANDA
Department of Physics, University of Sri Jayewardenepura, Sri Lanka
D.U.J. SONNADARA
Department of Physics, University of Colombo, Sri Lanka

ABSTRACT

A network centric remote data acquisition system was developed by utilizing the available network resources and technologies such as Java applets / applications and sockets. The digitizing hardware was based on an AD7569 chip that contains two ADC channels with conversion time of 2 μs and two DAC channels with settling time of 1 s both with 8-bit resolution. The system enables the analysis and the displaying of dynamic analog signals that are acquired at a remote location, on a web browser based interface of a local PC.

1. INTRODUCTION

With the rapid expansion in Information Technology, especially in the communication and networking areas, new products and instruments are introduced to the market frequently. These instruments help researchers to explore and insight into their problems in far superior ways than was available a few years ago. Today, many researchers require automated setups that can collect data without their presence. Often large amounts of data are required to overcome the noise and other interference on the signals that they study. In recent years, due to the expansion of modern society, researchers are often on the move and require means and methods to monitor and control their research activities through remote mechanisms. Moreover, they require systems that could work without any dependence on hardware platforms which are often not of their choice. With the existing technologies, today, it is possible to implement remote virtual instruments that acquire dynamic data from or control processes occurring at a remote location.

The main objective of this work is to develop the hardware and software to implement a remote data acquisition system which achieves the following goals; (i) low cost (ii) remote monitoring and control (iii) universal accessibility without special software at client end (iv) scalability and (v) customizability.

The quality of client-server based network centric systems depends heavily on the reliable and efficient transfer of data between the client and the server. Hence, TCP, which can provide reliable, point-to-point communications over client-server applications on the Internet / Intranet, was utilized. To communicate over TCP, a client program and a server program establish a connection with one another. Each program binds a socket to its end of the connection. The client and the server read from and write to the socket bound to their end of the connection[1].

The sockets used in the system discussed here are Stream sockets. Stream sockets are connection oriented (need to setup a connection between the two end points prior to transfer of data) and provide for a data flow without record boundaries (a stream of bytes). Streams are guaranteed to be delivered to the other end point of the link and to be correctly sequenced.

In this work, multitasking is used both in the socket server application and the client applet.

2. DESIGN AND IMPLEMENTATION

2.1. Overview of the System

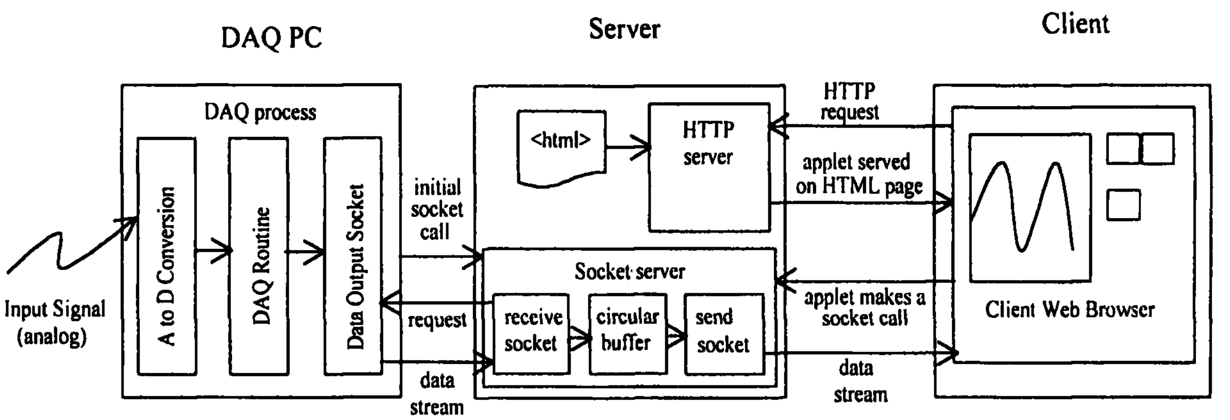

Fig. 1: The overview of the system

An analog signal is sampled at the client machine, converted to a digital signal and periodically transmitted via a Socket using TCP/IP to the receiving end of the server process. The receiving Socket of the server processes and relays the data (arrays of bytes) to another Socket which listens constantly to a transmitting port till a client makes a call to it.

A client (or multiple concurrent clients) can access the server and request an HTML document that will be served with an applet attached to it. The applet runs on the client machine. Data obtained from the server is processed and displayed on the client's web browser via the applet.

2.2. The Data Acquisition Subsystem

The data acquisition card built for the present work was based on the AD7569 chip. The AD7569 contains one 2 μs ADC and one 1 μs DAC both with 8-bit resolution. The initial design was based on the circuit developed by Joss Grot[2] for plug in I/O boards designed for the standard expansion bus of the IBM compatible computers. Most of the technical details related to circuit design are discussed therein. We used two AD7569 chips in our design to allow simultaneous digitization of two analogue signals. Only one of the input channels is utilized in this project. The following block diagram shows the essential features of this data acquisition card.

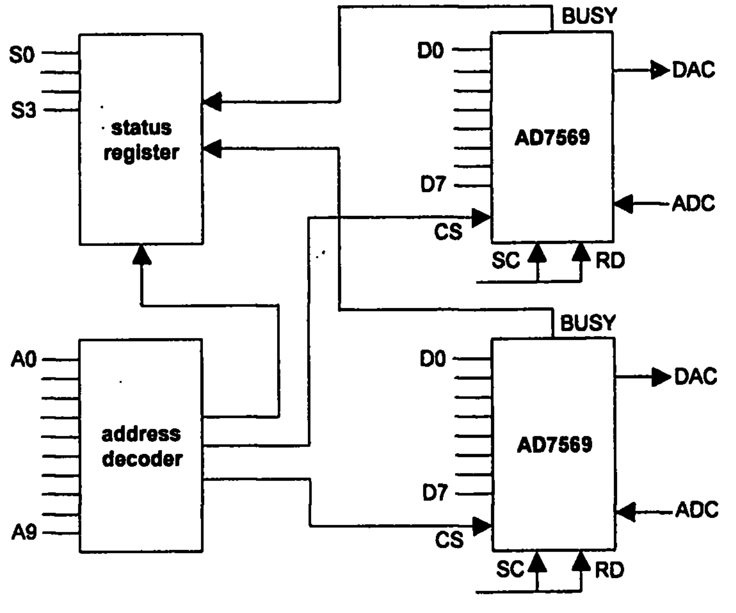

Fig.2 The block diagram of the data acquisition card based on the AD7569 chip

AD7569 contains an internal clock circuit with provisions to adjust the frequency with external components. Therefore no external clock pulse is required. Start conversion input is tied to read input so that every read operation starts a new conversion. The end of conversion can be detected by reading the status register. Both ADCs are started at the same time. The card uses three port addresses; two for the two ADC chips and one for the status register. In each ADC chip, the same address is used for reading ADC and writing to DAC.

The A to D conversion cycle of the DAQ card starts when a byte is read from port address $301 (this can be changed). The card digitizes the analog signals and stores the digitized information at port address $301. Whenever a read operation is performed on address $301 the ADC is reset and the next A to D conversion cycle is started. This mechanism guarantees that every conversion result is read and also that the sampling rate is equal to the rate at which the converted results are read from address $301 which is essential in extracting the optimum results from the system.

However, due to the fact that the sampling rate is determined by the rate at which address $301 is read, it was important to design a simple routine with minimum overheads. A C++ routine was written to read the digitized data into an array of N bytes (2000 bytes was found experimentally to be the most effective in the present network and machine setup) and to write to port 8190 on the server. This cycle is initiated with the request from the server. A flow chart for this process is shown in Fig. 3.

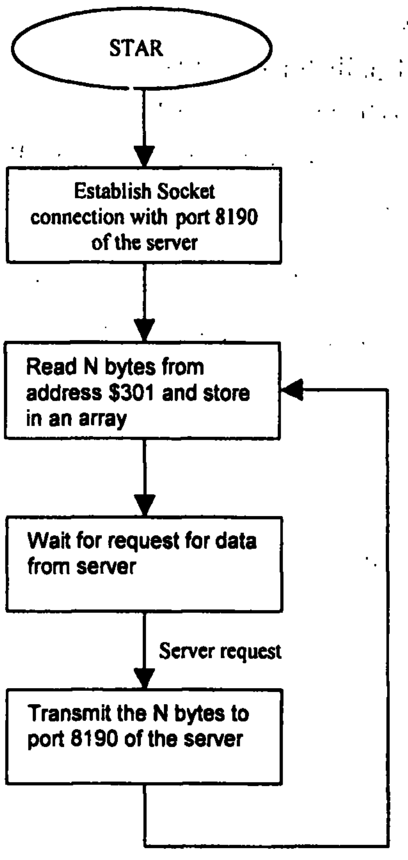

Fig. 3: The data acquisition and transmitting process

2.3 The Server

There are two distinct groups of functions of the server.

1. There has to be an HTTP server running on the server machine. Upon a request from a client(s) an HTML page with the Java applet should be passed on to the client machine[3]. Any HTTP server can be used for this purpose.
2. The applet requires the data uploaded by the data acquisition machine to be relayed to the client machine. The Java application (Socket Redirection Server) which has a 'Java runtime environment' designed for this function can be use for this purpose.

The socket redirection server is a multi-threaded application. Initially there is one control thread which listens to the input and output ports. When the data acquisition machine establishes a connection with the input port, a new thread is spawned. This sequence is shown in Fig. 4.

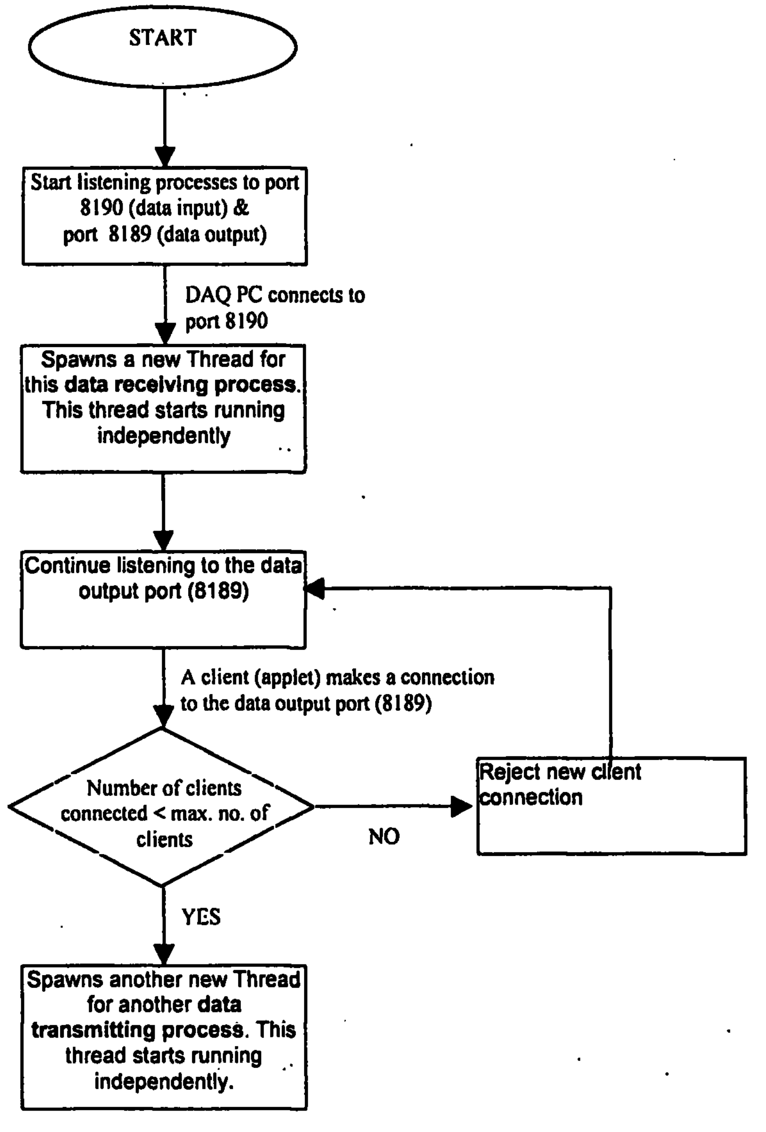

Fig. 4: The control thread of the socket redirection server

The data receiving process (thread) is entrusted with the task of receiving data from the input port and storing the data in a circular buffer.

The function of the data receiving process is shown in Fig. 5.

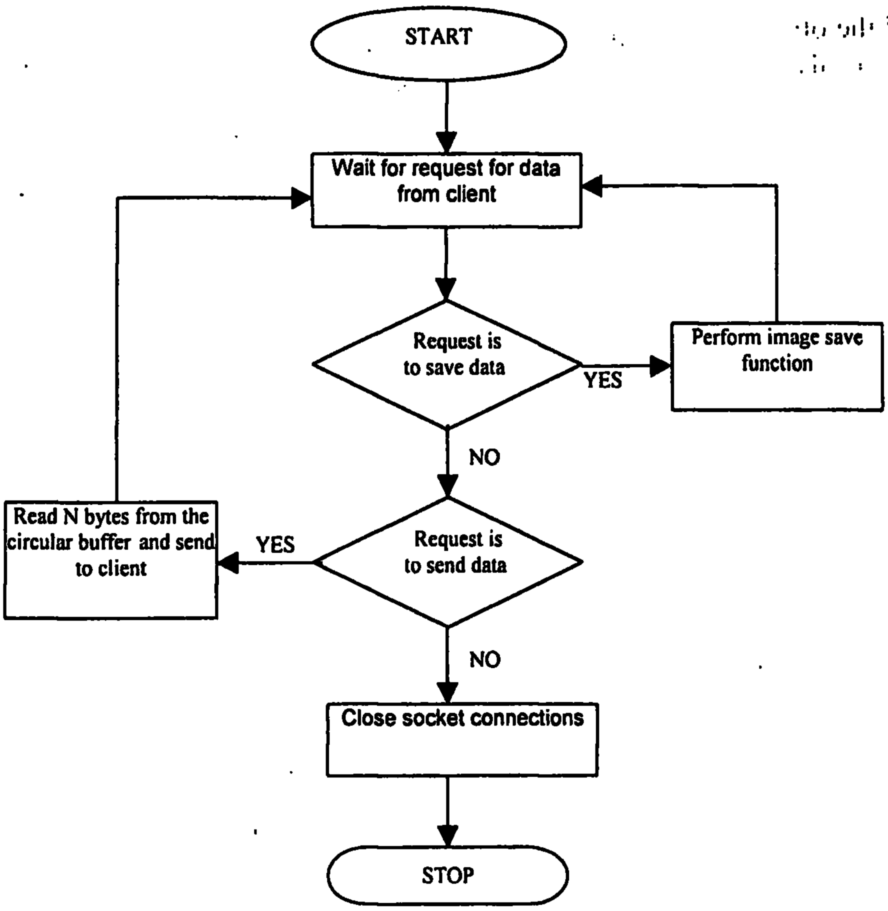

Fig. 6 The data sending thread

2.4 The Display and Analysis Subsystem

The Display and analysis subsystem is the front end of the remote data acquisition system. The user interface of this system is a Java applet that can be executed on any Java enabled web browser. When the applet is accessed via the HTTP server and loaded onto the user's machine it makes a Socket connection to port 8189 of the server. This connection is used to pipeline the data acquired at the data acquisition PC to the client PC. The data transfer from the server to the client takes place at the request of the client.

3. PERFORMANCE EVALUATION

3.1 Sampling Frequency effects and Quantization errors

Although the ADC7569 has a conversion time of 2 μs, due to the limitations in the system parameters such as multitasking and processing speed limitations, a maximum sampling speed of 25 μs was achieved with the present system. This corresponds to a sampling frequency of 40 kHz and hence frequencies up to 6 kHz could be displayed effectively on the system.

The configuration of the data acquisition card used in the present work accepts input voltage from 0-5 Volts. Since AD7569 is an 8-bit ADC, the minimum reading was limited to 0.02 V, which corresponds to the quantization error of the system□

3.2 Intermittent Interruptions to Sampling due to Network Transmission

Intermittent interruptions to sampling may occur specially if it is used over Internet, because the sampling process needs to be interrupted during the transmission of the sampled data from the data acquisition machine to the web server. Even if two threads are used, one for sampling and the other for transmitting, the instance of time that the program control will be given to the transmission thread by the OS cannot be predicted. This will result in irregular sampling intervals. Hence this routine had to be run on a single thread.

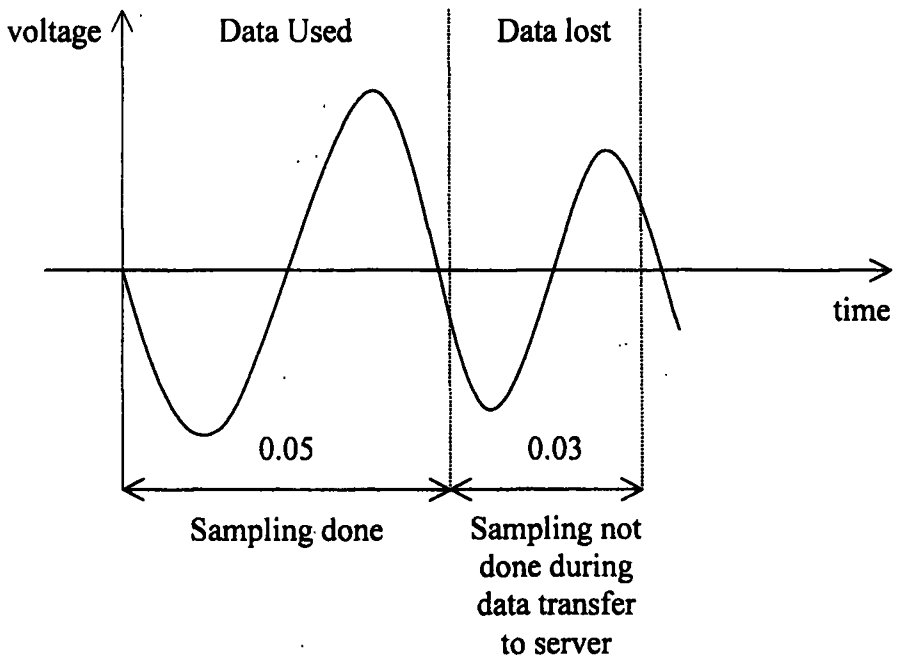

Fig. 7 Data lost during data transfer

In the present configuration, the sampling time for 2000 samples is limited to 0.05 sec. However it was found that the time to transfer a block of data over the present network configuration takes approximately 0.03 sec (see Fig. 7). Therefore it is apparent that a considerable proportion of the data (37.5%) can be lost in continuous monitoring and processing. This can be somewhat avoided by using a data acquisition card with data buffering.

3.3 Timing errors due to Windows 95 Multitasking

The sampling rate at the data acquisition machine is equal to the execution time of the inner loop of the C++ data acquisition routine. A calibration constant was used to adjust the delay in the loop to compensate for machines of different speeds. However if applications or background processes other than the data acquisition process are running at the time of data acquisition, the Windows 95 operating system may prompt and give control to another task in the middle of the data acquisition loop. This will alter the timing unpredictably.

One way of avoiding this problem is to dedicate a single machine for this work and to disable all other tasks that might run in the background.

3.4 Java applets

The Java applet code is interpreted at run time. Hence its execution is slower than a compiled code. Therefore faster and more efficient client and server programs could have been implemented in C. But in that case the ability to access the data acquisition system through the Internet and the platform independence of the system cannot be achieved.

4. CONCLUSIONS

The main objective of producing a network centric low-cost remote data acquisition system was achieved. A user only needs to have a web browser on his/her PC, which is connected to a network to activate the remote data acquisition system. For different applications of the system, only the post processing software has to be changed.

Due to the limitations of data transfer speeds over the network, the data acquisition system, in its present state, is more suitable to be used in a LAN environment. Use on the Internet will be limited to slow frequency input signals. Further work is required before using the developed system in continuous monitoring and processing environments. However it can be used, as it is in environments where continuous monitoring is not required. This project can be viewed as an initial step towards constructing a full scale Remote Virtual Instrument that can operate over the Internet.

ACKNOWLEGEMENTS

The authors wish to thank NSF (Research grant RG/96/P/06) for providing the financial support to carry out this work.

REFERENCES

1. Comer, D.E., *Internetworking with TCP/IP*, Prentice Hall (1994)
2. Joss Grot, *Analog I/O Board for PC's*, Internet White Paper (1992)
3. Vanderberg, G., *Tricks of the Java Programming Gurus*, Sams (1996)