PC-Based Oscilloscope

B.Tech Summer Training Report

Submitted By

Nitin Isloorkar
Sahil Ahuja
Sumit Mathur

National Institute of Technology,
Tiruchirappalli

Under the guidance of

Prof. T. S. Rathore

Department of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai- 400 076

Acceptance Certificate

The summer training report titled "PC Based Oscilloscope" submitted by Nitin Isloorkar, Sahil Ahuja and Sumit Mathur of NIT Trichy is accepted for B.Tech Summer Training and fulfills the requirements for the same.

Date : 26th June 2006 Signature

Prof. T. S. Rathore
(Electrical Engineering Department, IITB)

Acknowledgments

With due respect and gratitude we would like to thank our supervisor Prof. T. S. Rathore for his constant support, able guidance and ever flowing stream of encouragement throughout this work.

We would also like to thank Mr. Uday Pandit Khot (PhD Student of IIT Bombay) who helped us in our endeavor and all the staff of the Electrical Engineering Department of IIT Bombay who made working on this project and completing it an enjoyable job for us.

Date : 26h June 2006

Nitin Isloorkar
Sahil Ahuja
Sumit Mathur

Table Of Contents

1 Introduction

Although most people think of an oscilloscope as a self-contained instrument in a box, a new type of "oscilloscope" is emerging that consists of an external analog-to-digital converter (sometimes with its own memory and perhaps even some data-processing ability) connected to a PC that provides the display, control interface, disc storage, networking and the electrical power. The viability of these so-called **PC-based oscilloscopes** depends on the current widespread use and low cost of standardized PCs. This makes the instruments particularly suitable for the educational market, where PCs are commonplace but equipment budgets are often low.

This PC-based oscilloscope (PCBO) is designed by interfacing a computer through the parallel port. The input signal is sampled, stored and plotted on the screen. This is done using a database for storing the sampled data and a web browser for displaying it. The PCBO is divided broadly into two parts:

1. The data acquisition system.
2. Plotting program.

Commercial PC based oscilloscopes like Virtins are available in the market with high resolution and good frequency response, but they do not have remote screening capability and are costly. Other oscilloscopes like Xoscope*[1]* are available but they use the sound card of the PC and do not use a buffer for motherboard protection and also do not have the remote screening capability.

This PCBO has remote screening capability, which means that the signal can be sampled on one computer and the plot can be displayed on another computer connected to the Internet. It is also very cheap to make, costing only Rs. 200 to make and plots successfully DC and low frequency signals (1 to 150 Hz) that a conventional cathode ray oscilloscope fails to display.

2 Block Diagram

A basic block diagram for describing the PCBO is shown in Figure 1:

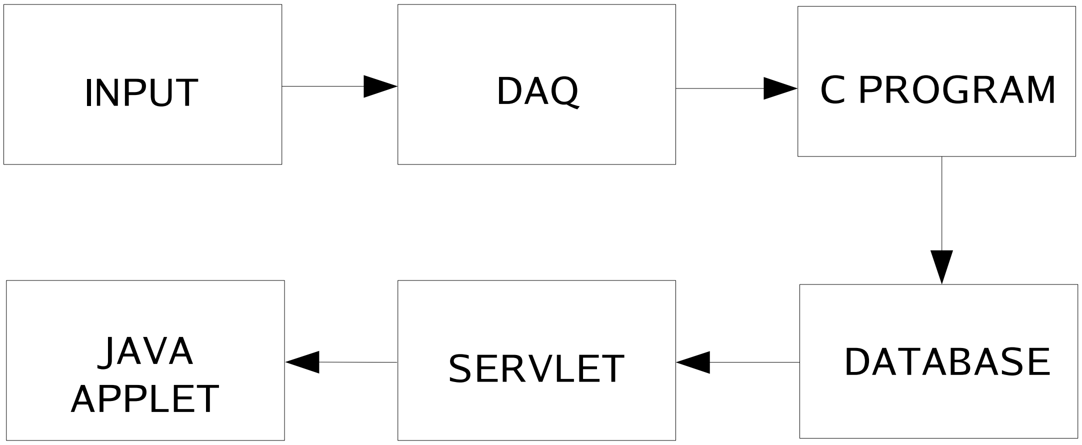


Figure 1: Block Diagram

Data flow

The input signal is first sent to the analog-to-digital converter ADC0804, where it is converted into an equivalent 8-bit digital number. It is then sent to the parallel port of the computer through a buffer 74LS245 used for electronic isolation. The data is then sent into a MySQL database. This process of controlling the ADC through the parallel port and sending the data to the MySQL database is done by C program. A Java applet then requests this data from the database through a Java servlet and displays it on the screen. Only the Java applet runs on the client.

3 Working

Data Acquisition System

The PCBO has a data acquisition system based on the parallel port. The parallel port is a 25-pin port usually used to connect a printer to the PC. It has a data rate of about 150 kilo samples per second, which is more than sufficient to handle the output of the ADC.

It consists of a hardware circuit that converts the analog/digital input to a weighted 8-bit binary number. This is achieved by a 8-bit parallel output Successive Approximation Register analog-to-digital converter ADC0804*[2]*, with a sampling rate of around 100µs. This imposes a theoretical hardware limit on our input signals to be around 1kHz. The speed of the conversion is limited by the approximation procedure of the analog-to-digital converter.

It is dangerous to connect the circuit directly to the parallel port, as a power surge in the circuit can damage the motherboard of the PC. Hence, a bidirectional octal bus transceiver (buffer), 74LS245N*[3]* between the ADC and the parallel port of the PC is used. The entire circuit is based on TTL logic.

The data is read by the computer using a C program using the PARAPIN[1]*[4]* library . The C program also inserts these readings into a MySQL database. Figure 2 demonstrates the flow of data.

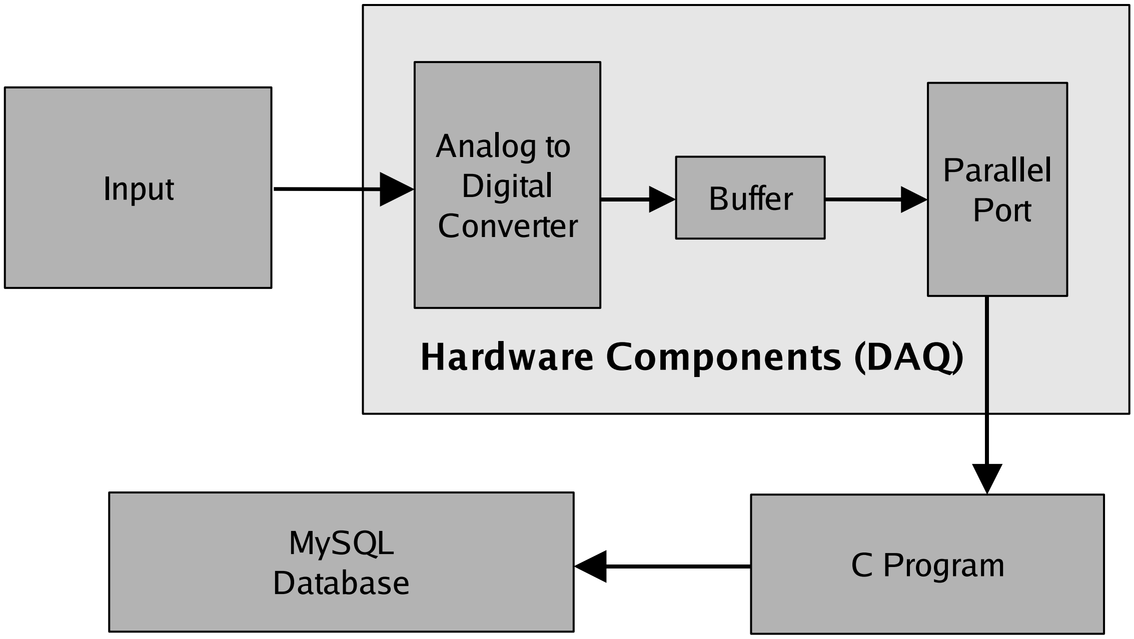


Figure 2: Data Acquisition System

[1] **PARAPIN** is a C library that makes it easy to write C code under Linux that controls individual pins on a PC parallel port.

Like most microprocessor compatible chips, the ADC0804 has control and status signals. It's control signals are CS (chip select), RD (read) and WR (write) and its status signal is INTR (interrupt). The ADC begins it's conversion when WR transits from low to high. During the conversion process, INTR remains high. When the conversion is over, the data is stored in an internal register and INTR goes low. When CS is low and RD goes from high to low, the data from the internal register is transferred into the data bus of the ADC.

Plotting Program

This PCBO uses the Fedora Core 5 Linux operating system. The actual plotting is done in a web based applet[2]. A servlet[3] server (Apache Tomcat[4]/5.5.15*[5]*) is used to transfer the data from the servlet to the applet.

Applet-Servlet Communication

Figure 3 summarizes the actual path followed by the data before it gets plotted:

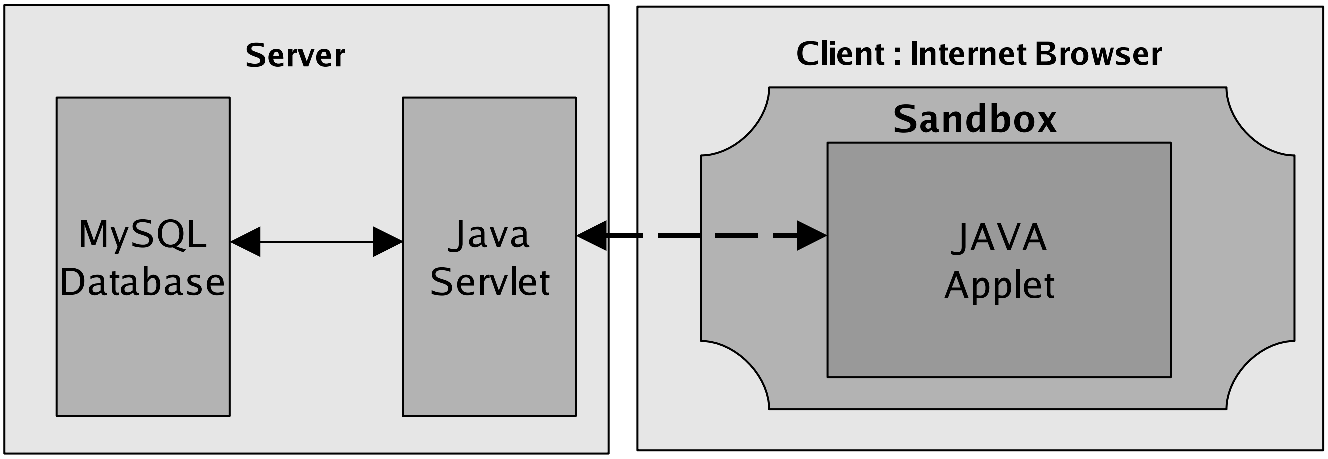


Figure 3: Applet-Servlet Communication

An applet is run in a "Sandbox" in any Internet browser. The Sandbox is a kind of virtual fencing that prevents the applet from accessing the user's computer resources. This prevents the applet from contacting any database. The only thing that the applet is allowed to connect is the server from which it was called.

So, in this PCBO, the applet is called from a server which also hosts the necessary resources to provide the applet with the required data. We used a Tomcat server. A servlet is kept on the server, which the applet queries. This servlet, in turn queries the database and sends the formatted data to the applet *[6],[7]*.

[2] **Applets** are web-based java programs that can be run in a web browser.

[3] **Servlets** are the Java counterpart to dynamic web content technologies such as CGI, PHP or ASP.

[4] **Tomcat** is a web-server that functions as a servlet container. Tomcat implements the servlet and the Java Server Pages (JSP) specifications from Sun Microsystems. It is being developed under the Jakarta Project at the Apache Software Foundation.

The applet-servlet connection*[8]* takes up significant overhead. So, in order to improve the response time, the applet queries data for many frames in one go. The servlet queries the database, converts it into a serialized[5] form, and sends it to the client, which in our case is the applet. The applet receives this chunk of data, again extracts the raw data from the serialized data, and plots the data after inflating[6] it.

Multithreading[7]

The C program sampling the data and querying the database and the applet-servlet connection and querying may also adversely affect the operation of plotting the data. In order to remove this effect, multithreading*[9]* is used in the system. One thread[8] queries the servlet while another is assigned the job of plotting the graph. Figure 4 shows the process.

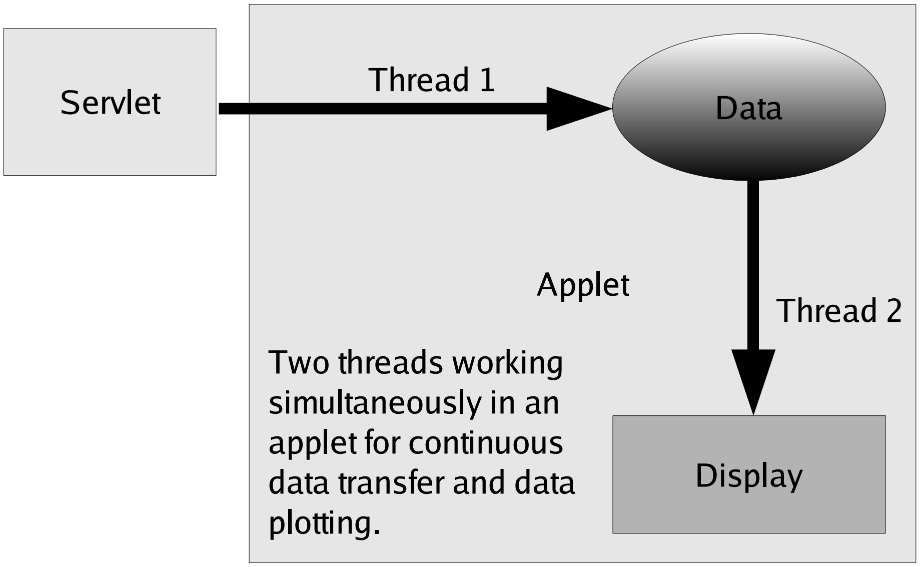

Servlet
Thread 1
Data
Applet
Thread 2
Two threads working simultaneously in an applet for continuous data transfer and data plotting.
Display

Figure 4: Multithreading

Buffering

Plotting an image is a time intensive process. If the applet is allowed to plot an image directly on the display, flickering of the display takes place. This is due to the difference in the applet's rate of plotting the graph, and the monitor's refresh rate.

This flickering is overcome by using a buffer*[10]* to plot the graph. A buffer has been used in this PCBO for the actual plotting of graph. After the plotting of the graph in the buffer is complete, the buffer is simply copied on the display, which in turn is a time efficient process. This leads to very smooth transitions of graphs.

[5] **Serialization** is the process of saving an object onto a storage medium (such as a file, or a memory buffer) or to transmit it across a network connection link such as a socket either as a series of bytes or in some human-readable format such as XML.

[6] **Inflation** is the opposite of serialization.

[7] **Multithreading** is a form of parallelism where multiple threads run concurrently and communicate via shared memory.

[8] **Threads** are a way for a program to split itself into two or more simultaneously running tasks.

4 PCBO Software and Hardware

Specifications

A PC Based Oscilloscope was designed and implemented working in the frequency range 1-150Hz. It has the following specifications:

1. Frequency range: 1Hz – 150Hz (tolerance of ±10%).
2. Voltage Range: 200 mV_{PP} to 10 V_{PP}.
3. DC plotting is possible.
4. Run/Stop, Auto Trigger, Vector mode, Ground, X-offset, Y-offset available.

Software Discussion

The C program uses the parapin.h and the mysql.h header files. These files have functions that make accessing the parallel port and the MySQL database easy. The C program has two threads that run independently. One thread gives control signals to the ADC and reads the data from the parallel port. The other thread inserts the data into the MySQL database after every 400 entries. Multithreading ensures that no data is lost as a result of time delay caused by querying the database.

After insertion of data into the MySQL database, it is transferred to the applet with the help of a servlet server. The servlet uses the standard JDBC[9] drivers for connecting to the MySQL database. After querying the data from the database, it serializes the data and transmits it to the applet. The applet inflates the data and plots it frame by frame. It buffers the images first, and then replaces the displayed image with the buffered image. Buffering is used here to to prevent flickering. The applet has two jobs that take significant overhead – inflating the data and rendering the image. To prevent interference of the jobs due to one another, these two jobs are carried out independently of each other using two different threads.

[9] **JDBC** (Java Database Connectivity) is a Sun Microsystems standard defining how Java applications access database data.

Circuit Diagram and discussion

The RD, WR and INTR pins of the ADC are connected to pins 17, 16 and 10 of the parallel port respectively. The data pins 2 – 9 and pin 10 are configured as input pins (data goes into the computer from the hardware through the parallel port) and pins 16 and 17 are configured as output pins (data is given from the computer to the hardware through the parallel port). The output data bus of the ADC, pins 11- 18, is connected to the data bus 2 -9 of the parallel port through the buffer. The external connections to this circuit are the supply voltage V_{CC}, the internal reference voltage $V_{REF/2}$, the analog input signal and the common ground line. The circuit diagram is shown in Figure 5:

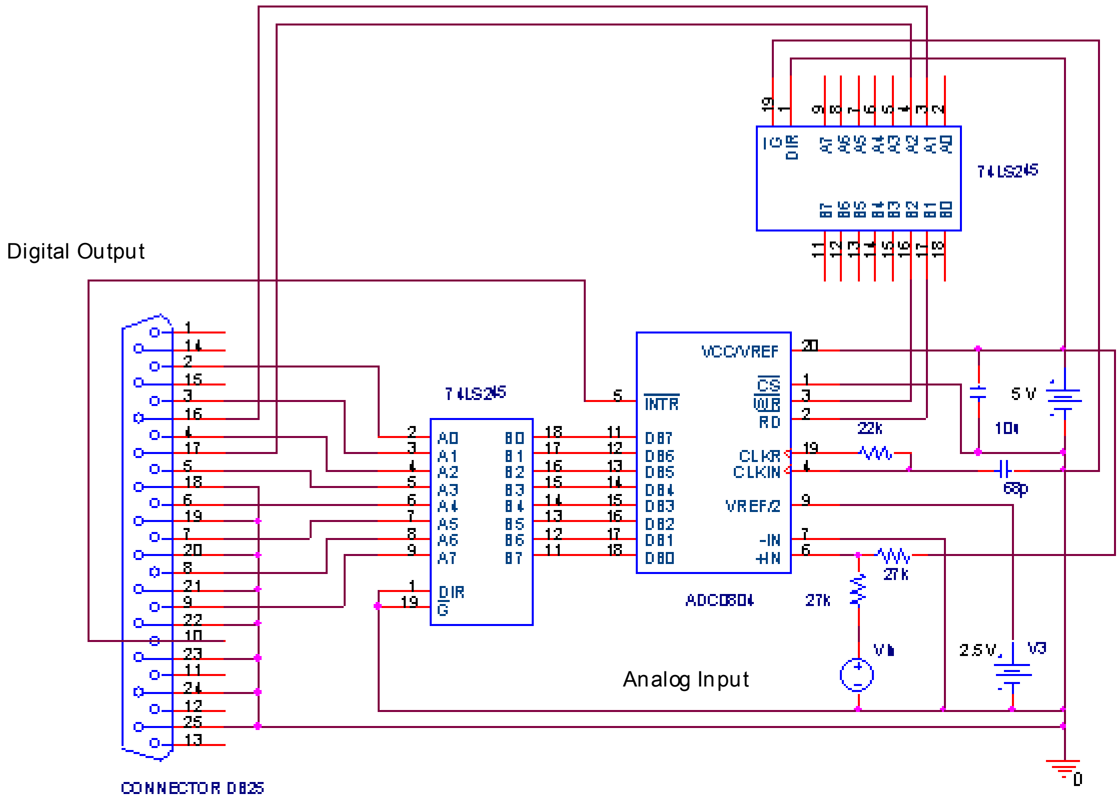


Figure 5: Circuit Diagram

5 Result

The ADC used in this PCBO was calibrated by giving analog inputs in steps of 100mV, and taking the analog equivalent voltage of the corresponding digital output. A graph was plotted between the two voltages. For a perfect ADC, we expect this line to be a perfect straight line with a slope of 45° as the converted value should correspond exactly to the analog input. The graph that was plotted shows a nearly straight line. The error percentage was calculated and was found out to be 0.1213%. The response of the ADC is illustrated in Figure 6:

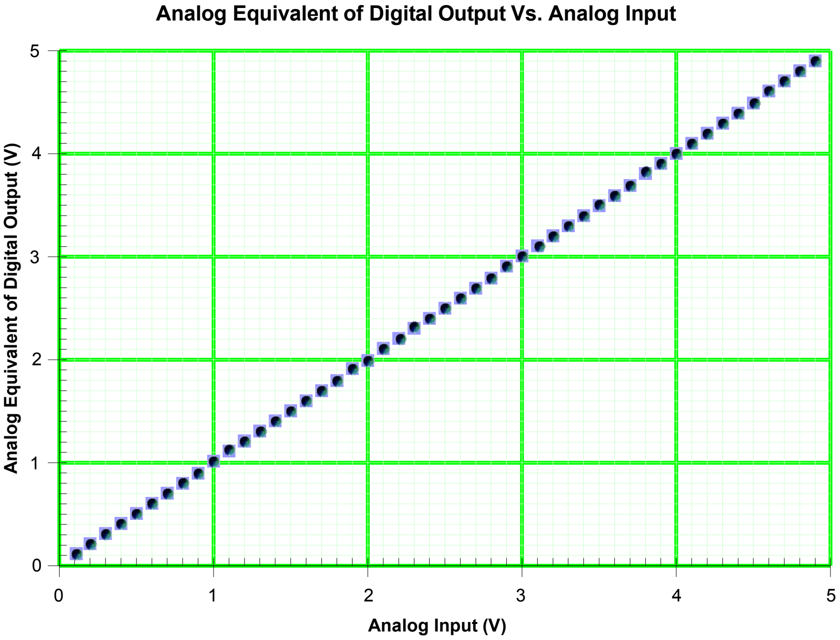
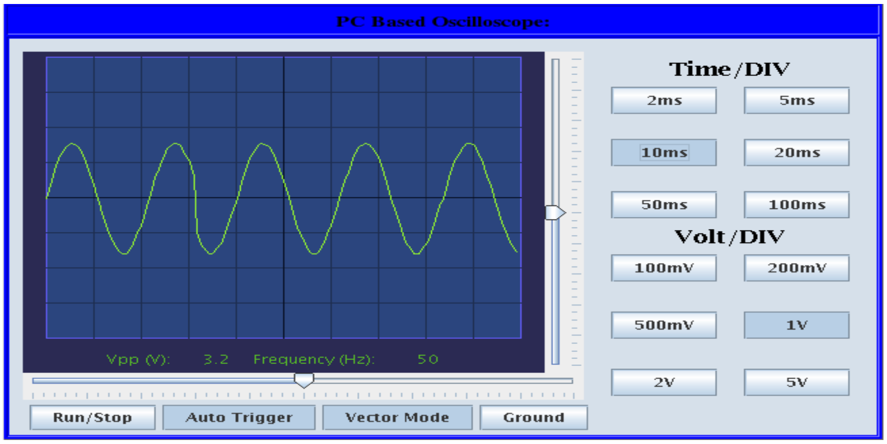


Figure 6: Response of ADC

The current display plots the waveform with a flicker. This is caused as the program takes time to store data into the MySQL database. Due to insufficient RAM on the computer, the database and the Tomcat server were setup on another computer. This introduces a 3 seconds delay between the actual signal and the plot. Currently the PCBO implements a sample-store-display algorithm. If sufficient RAM (512MB) is provided, the PCBO can be done entirely on a single computer. The storage and display part can be implemented simultaneously, thereby eliminating the flickering.

Shown in the following figures are the screenshots of the PCBO in Vector, Non-Vector and Ground Modes:

- Vector Mode :



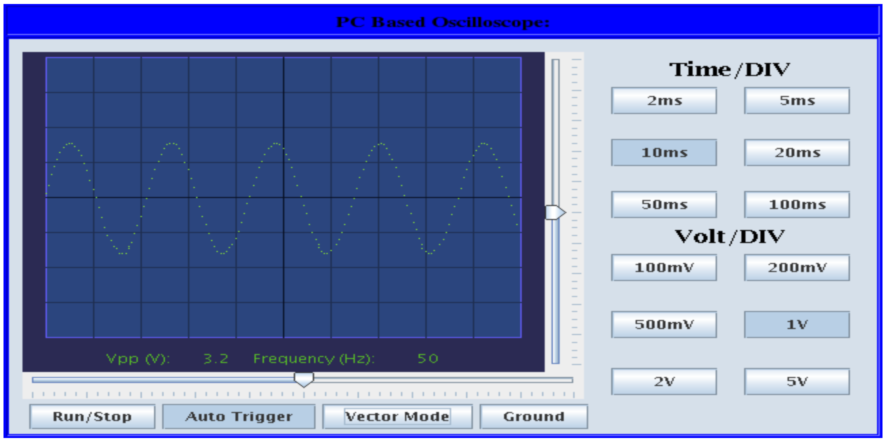
PC Based Oscilloscope:

Vpp (V): 3.2 Frequency (Hz): 50

Figure 7: Vector Mode

In the Vecotr Mode, all the sampled points are connected by line, giving the output as a continuous graph. This is a virtual output to enable the viewer to visualize the waveform.

- Non-Vector Mode:



PC Based Oscilloscope:

Vpp (V): 3.2 Frequency (Hz): 50

Figure 8: Non-Vector Mode

The Non-Vector Mode shows the actual value of the sampled data. The horizontal distance between the points in this mode represents the delay between subsequent samples.

In both these modes, the X-Axis and Y-Axis sliders are provided to allow the user to view the waveform with X or Y axis offset. The user also has the option to change Time/Div and Volt./Div settings of the PCBO with the help of toggle buttons provided in the applet interface. A ground mode has also been provided to enable the user to conveniently set the DC offset of the signal source.

Anytime during the normal working of the PCBO the user may pause the display in order to analyze the current waveform with greater detail.

6 Summary and Conclusion

Advantages

1. The advantage of this PCBO is its remote screening applications. The oscilloscope display can be transferred over the Internet and can be viewed anywhere in the world through the Internet.
2. The PCBO has a very interactive GUI making it more convenient to use than conventional oscilloscopes.
3. The circuit is extremely cheap and easy to make.
4. All softwares used are Open Source and free.

Disadvantages

1. As compared to the USB and Fire Wire ports (having a speed of 12 million samples per second), the parallel port is relatively slow (150 kilo samples per second).
2. The input voltage range is -5V to +5V and the oscilloscope cannot satisfactorily display input voltages of magnitudes less than 100mV$_{PP}$.
3. The ADC we have used has a conversion rate of only 100µs, limiting the input frequency to only 1kHz.
4. There is a 3 seconds delay between the actual waveform and the plot.

Improvements on the PCBO

1. If an ADC with a resolution of 16 bits and a conversion time of 2µs is used, signals of up to 10kHz can be plotted with precision.
2. Data Acquisition Cards (DAQ's) for USB port can also be used for getting better frequency response , but they are not economically viable.
3. The limited voltage range can be improved by providing logical amplifiers and attenuators.
4. If needed, programmed control signals can also be sent back to the source.

Applications of the PCBO

1. The PCBO can be used as a remote monitoring device due to its remote viewing capability.
2. The sampled waveform is stored in the database and can be used for future reference.
3. The data can be easily exported to standard PC softwares such as spreadsheets and word processors.

7 References

[1] http://xoscope.sourceforge.net

[2] http://www.datasheetcatalog.com/datasheets_pdf/A/D/C/0/ADC0804.shtml

[3] http://www.datasheetcatalog.com/datasheets_pdf/7/4/L/S/74LS245.shtml

[4] http://parapin.sourceforge.net/doc/parapin.html

[5] http://tomcat.apache.org/download-55.cgi

[6] http://www.codecomments.com/archive248-2004-4-182225.html

[7] http://www.stardeveloper.com/articles/display.html?article=2003090401&page=1

[8] http://www.frank-buss.de/echoservlet/

[9] http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html

[10] http://www.dgp.toronto.edu/~mjmcguff/learn/java/07-backbuffer/

[11] The Complete Reference JAVA 2 By Herbert Schildt, McGraw Hill Publishers.

[12] http://www.j-nine.com/pubs/applet2servlet/index.htm

[13] http://forum.java.sun.com/thread.jspa?forumID=48&threadID=391594

[14] http://forums.fedoraforum.org/showthread.php?t=47591&highlight=tomcat

[15] http://www.developer.com/java/data/article.php/3417381

[16] http://www.sci.usq.edu.au/staff/leighb/graph/

[17] http://javaboutique.internet.com/educational/

[18] http://tecfa.unige.ch/guides/java/staf2x/ex/jdbc/coffee-break/

[19] http://javaboutique.internet.com/SimplePlot/source.html

[20] http://www.csupomona.edu/~pbsiegel/www/stujav.html

[21] http://servlets.com/jservlet2/examples/ch10/index.html