Annual Academic Sessions 2012
ISSN 2012-9912; ©2012 Open University of Sri Lanka

PC BASED SPECTRUM ANALYZER

S. R. B. Herath*

Department of Electrical and Computer Engineering, The Open University of Sri Lanka

INTRODUCTION

The Spectrum Analyzer is a measurement device used to examine the spectral composition of some electrical, acoustic or optical waveforms. Since it is a very expensive instrument it is very hard to find in small scale electronics workshops. The size and weight of conventional Spectrum Analyzers limits their portability. These limitations can be overcome by implementing a computer based Spectrum Analyzer to fulfill the basic requirements. The PC based Spectrum Analyzer is an alternative for conventional spectrum analyzers. Although it would not fulfill all features and capabilities of a commercial Spectrum Analyzer, it will provide a good solution for the above limitations. This instrument is suitable for small workshops because it does not require additional space. Furthermore, the proposed Spectrum Analyzer can be fully configured through the computer software, is fully compatible with Microsoft Windows 7 operating system and supports USB 2.0 high speed data communication interface. Other important features to note are the compatibility of the instrument to extend its operation in the programming mode and the compatibility to take time domain measurements in the Oscilloscope mode. Finally, the Spectrum Analyzer presented would be a very useful instrument for technicians and students in electronics and communication field, due to its low cost and portability.

SYSTEM DESIGN AND IMPLIMENTATION

The PC based Spectrum analyzer consists of hardware modules and software. Appropriate firmware and software are prepared for the microcontroller and computer in order to communicate with each other via the USB. The *LabVIEW (Laboratory Virtual Instrument Engineering Workbench)* program is used to develop a software application for the computer [6].

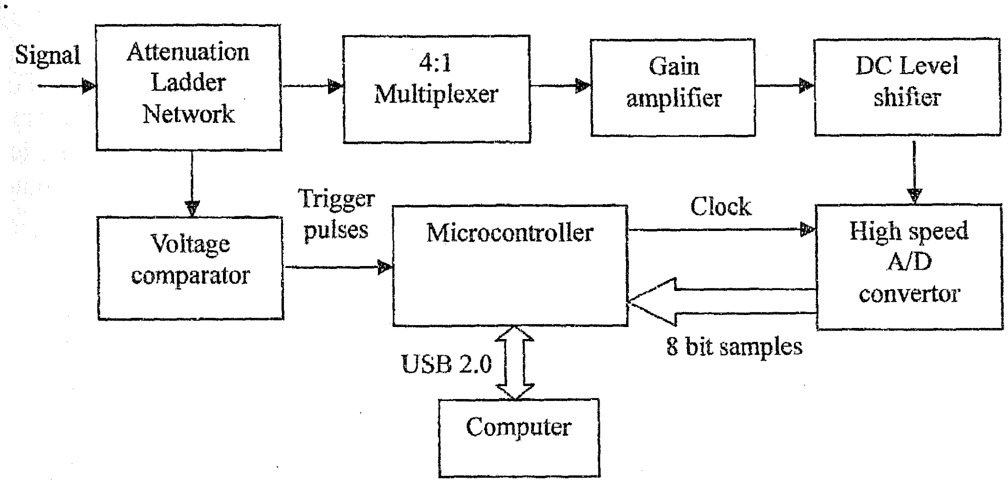

Fig. 1: Block diagram of the hardware module

The amplifier receives an analog signal which is amplified to the desired value. The microcontroller drives the multiplexer which routes the analog signal through the *R-2R* attenuation ladder network.

*Correspondences should be addressed to Eng. S. R. B. Herath, Department of Electrical and Computer Engineering, The Open University of Sri Lanka (email: ruwanthaherath@gmail.com)

The *A/D Convertor (ADC)* converts the input analog signal to digital. Each digital sample is saved in a queue in the microcontroller memory [3]. During the sampling process, collected samples are sent to the computer via the *USB* [1] [5].

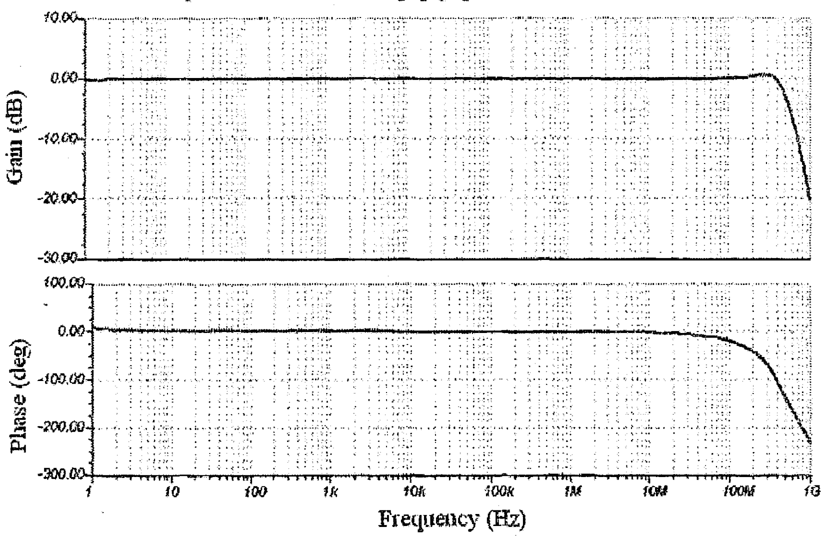

Fig. 2: Frequency responses of the Signal conditioning unit [4].

Two sampling techniques can be implemented in the instrument: Real Time Sampling *(RTS)* and Equivalent Time Sampling *(ETS)*. The RTS technique takes samples according to the Sampling theorem. When the ETS technique is used, the trigger signal is provided by the voltage comparator and it triggers an interrupt. When an interrupt triggered, initially a small delay is added to the sampling time of the ADC. At the next trigger, a small time increment is added to the delay and takes another sample. This procedure repeats to capture the whole waveform.

Signal conditioning modules have a bandwidth up to 100MHz. as an optional feature; advanced users can modify the firmware to achieve 100MHz bandwidth by using ETS technique. Normally the instrument operates with the RTS technique.

The computer software collects samples coming from the microcontroller. Next, each sample is stored in an array in the computer memory. After collecting 1000 samples, the Fast Fourier Transform *(FFT)* is calculated for the sample array and traces the result on the computer screen. In the Oscilloscope mode, it bypasses the *FFT* calculation. The *LabVIEW* Application also modifies the attenuation by sending a new attenuation factor to the microcontroller through *USB* [2].

The FFT of sampled data are computed using the *LabVIEW* built in function *FFT Spectrum (Mag-Phase).vi.* The frequency spectrum is displayed on a graphically on the screen using the *LabVIEW* built in function *Waveform Graph.vi* which automatically calibrates the *X* axis and *Y* axes of the graphic chart [6]. The *X* axis is calibrated in frequency and *Y* axis is calibrated in amplitude.

FFT based measurements are subject to errors from an effect known as leakage. This effect occurs when the *FFT* is computed from a block of data which is not periodic. To correct this problem appropriate *windowing* functions must be applied. The user must choose the appropriate *Window* function for the specific application. When *windowing* is not applied correctly, errors may be introduced in the *FFT* amplitude, frequency or overall shape of the spectrum. Difference *windowing* functions are available in *LabVIEW*. These functions are used to increase the accuracy of waveforms. Natural frequencies can be added to the spectrum. They are removed by applying a band reject filter by the software. Finally an executable file is created with *LabVIEW* [6]. This can be executed as a stand-along application independently of the *LabVIEW* platform.

TESTING AND RESULTS

To verify the correct operation of hardware and software, the instrument was tested for both sine and square waves with known amplitudes, duty cycle and frequency.

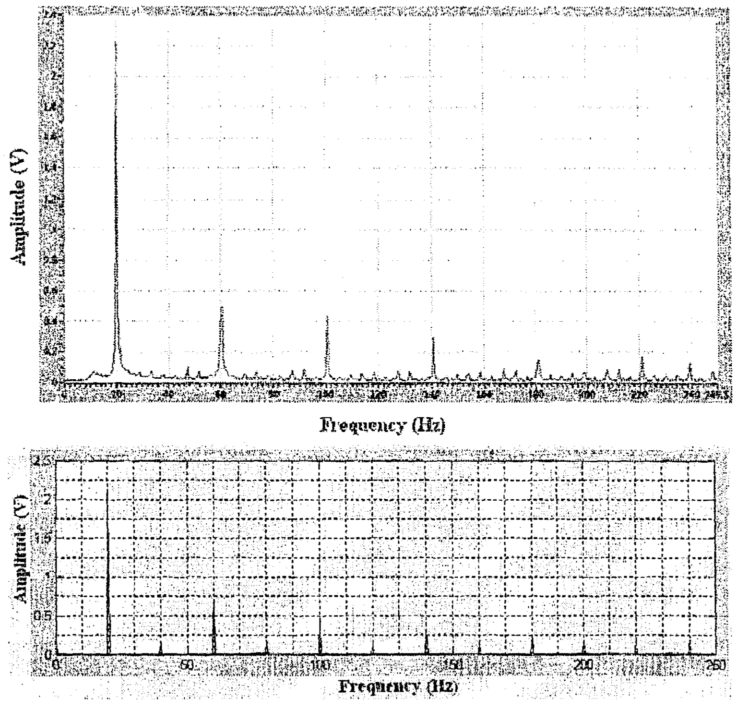

Fig. 3: Frequency spectrum of 20Hz square wave, first figure shows the real time result and second figure shows the MATLAB simulation result.

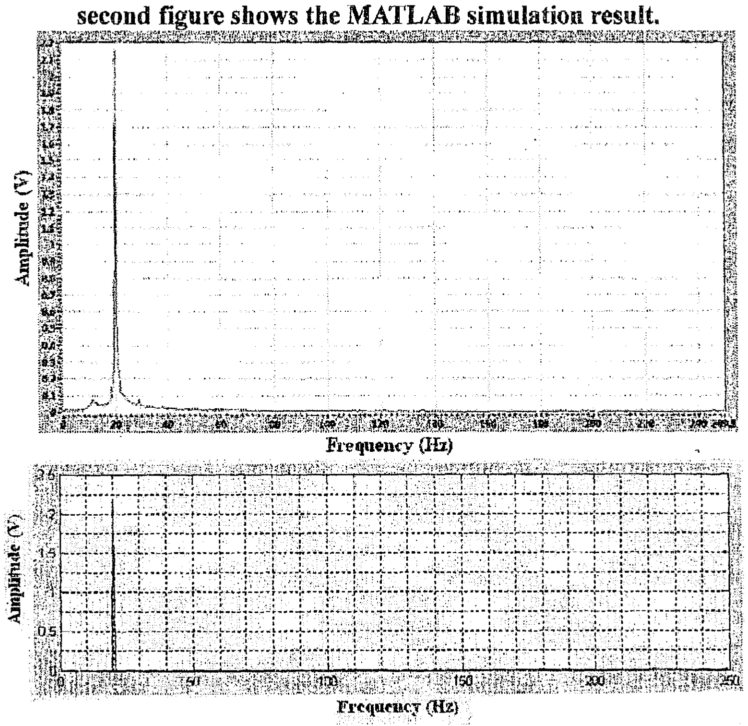

Fig. 4: Frequency spectrum of 20Hz sine wave, first figure shows the real time result and second figure shows the MATLAB simulation result.

MATLAB simulations were used to compare the waveforms given by the instrument [7]. This helped to check the accuracy of real time results.

The square wave generated by the signal generator produced a number of harmonics in the frequency spectrum. The amplitude of these harmonics depends on the duty cycle of the waveform. Figure 3 shows the frequency spectrum displayed by the Spectrum Analyzer for 20Hz, $2.2V_{p-p}$ and 50% duty cycle square wave. It was also observed that the frequency spectrum had harmonics at 60 Hz, 100Hz, and 140Hz.

Figure 4 shows the frequency spectrum produced by the spectrum analyzer for 20 Hz, $2.2V_{p-p}$ sine wave and its simulation result. The spectrum shows a single frequency component at frequency 20 Hz. According to above results it is clear that the PC based Spectrum Analyzer produce the frequency spectrum accurately.

CONCLUSION

This paper presents software and hardware implementations of a new computer based Spectrum Analyzer. The portability and cost are major factors concerned when developing the instrument. Hardware I designed to be compatible with both *ETS* and *RTS* techniques. Therefore an optional advanced programming mode had to be introduced to the system. It provides a firmware modification facility to advanced users. For higher accuracy and lower computational time, the FFT algorithm was applied on the computer platform for calculating the frequency spectrum of the sampled signal. The *USB* interface provided high speed data communication between the instrument and the computer. Finally test results verified that the functionality of software and hardware are in agreement.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to many persons who helped to make this project to success by encouraging and guiding me precede in the proper way. First all my sincere thanking goes to my parents, without them none of this would even be possible. Then my sincere thanks go to the academic coordinators and other staff members of the department of Electrical and Computer engineering, at The Open University of Sri Lanka, for their encouragements.

REFERENCES

[1] Ibrahim, D. (2008). Advanced PIC microcontroller projects in C: from USB to RTOS with the PIC18F series. USA: Elsevier Ltd.

[2] Axelson, J. (2009). USB Complete: The developer's guide (Fourth Edition). 5310 Chinook Ln., Madison WI53704: Lakeview Research LLC

[3] Mazidi, A. M., McKinlay, D. R. & Causey, D. (2008). PIC Microcontroller and Embedded Systems – Using Assembly and C for PIC18 (First Edition). Pearson Education Inc.

[4] SPICE-Based Analog Simulation Program, TINA-TI. Retrieved from http://www.ti.com/tool/tina-ti/

[5] Microchip USB framework. Retrieved from http://www.microchip.com/SoftwareLib.aspx

[6] NI LabVIEW System Design Software. Retrieved from http://www.ni.com/labview/

[7] MATLAB Signal processing tool box. Retrieved from http://www.mathworks.in/products/signal/