Design and Implementation of a Laboratory Instrument for Signal Analysis Applications

Omar H. Hamad, M.Sc.

Department of Computer Engineering, College of Engineering, University of Baghdad

Abstract

The purpose of this work is to introduce an economical spectrum analyzer for laboratory and test applications. Such an analyzer is achieved by utilizing the graphic capability of a personal computer (PC) and the sufficient numerical processing of the microcontroller.

المستخلص

إن الغرض من البحث هو تقديم محلل طيف ذو جدوى اقتصادية يستخدم في التطبيقات المختبرية والتطبيقات الفحصية أو التجريبية. لقد تم إنجاز المحلل باستثمار القابلية التخطيطّية للحاسبة الشخصية والمعالجة العددية الكافية للمسيطر المتناهي الدقّة.

1. Introduction

The availability of low cost and powerful computers (such as IBM PC or compatible) has led to the development of economical laboratory instruments based around the PCs [Dale and John, 1999 and Larry, 1983]. By the analyzer it is required to display the spectrum of a signal on the monitor. The signal is applied to the input terminals of the analyzer, then sampled and processed before its spectrum is displayed.

A conceptual block diagram of the spectrum analyzer is depicted in Figure1. The analyzer performs N-point radix-2 FFTs on input data, calculates the power spectrum density and transmits this to the IBM PC through an RS232C link. A Visual-Basic language program on the PC receives and displays the spectrum. All the numerical computations associated with spectrum estimation are performed by the Intel 89C52 microcontroller.

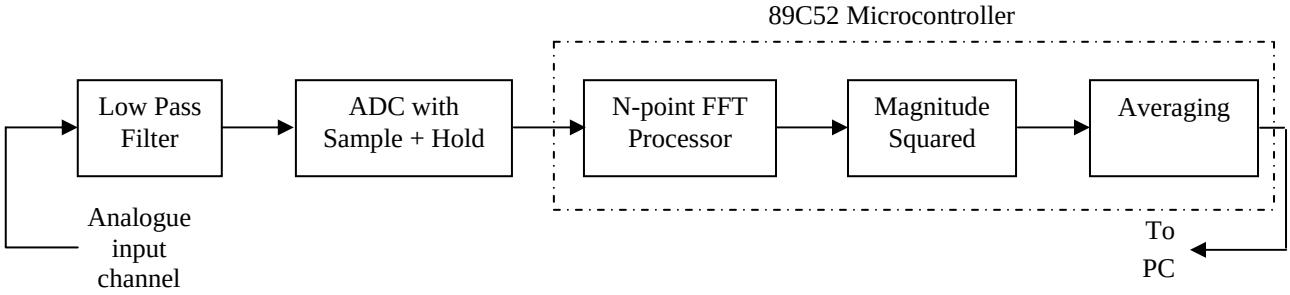

Figure1 the conceptual block diagram

In the next few sections, we will describe the features of the analyzer, the spectrum estimation techniques used, and the analyzer hardware and software.

The Fast Fourier Transform (FFT) spectrum analyzer uses the Fast Fourier Transform algorithm to evaluate the frequency spectrum of a signal from its time domain [Alan,1999]. The main features of the analyzer are as follows:

- analogue input channel, each with a 12-bit ADC;
- sampling rate < 25 kHz;

- estimates power spectrum density of signals in the frequency range 0-7kHz using the FFT algorithm;
- serial communication with a host personal computer via an RS232 port;
- display of signal spectrum on the PC using the host's graphics facilities.

Although it is implemented and tested at the present, the analyzer can readily support multivariable analysis (for example, cross-spectrum, cross-correlation), and the display of input signals in time.

2. Spectrum estimation

The spectrum analyzer uses the radix-2 FFT algorithm to convert a time domain data into a frequency spectrum displayed on the monitor. The FFT algorithm takes N data samples, *x(n) n = 0, 1,…, N-1*, and produces *N*-point complex frequency samples, *X(k) k = 0, 1, …., N-1* . The raw power spectrum density (PSD) is then obtained as the scaled magnitude squared of complex frequency samples:

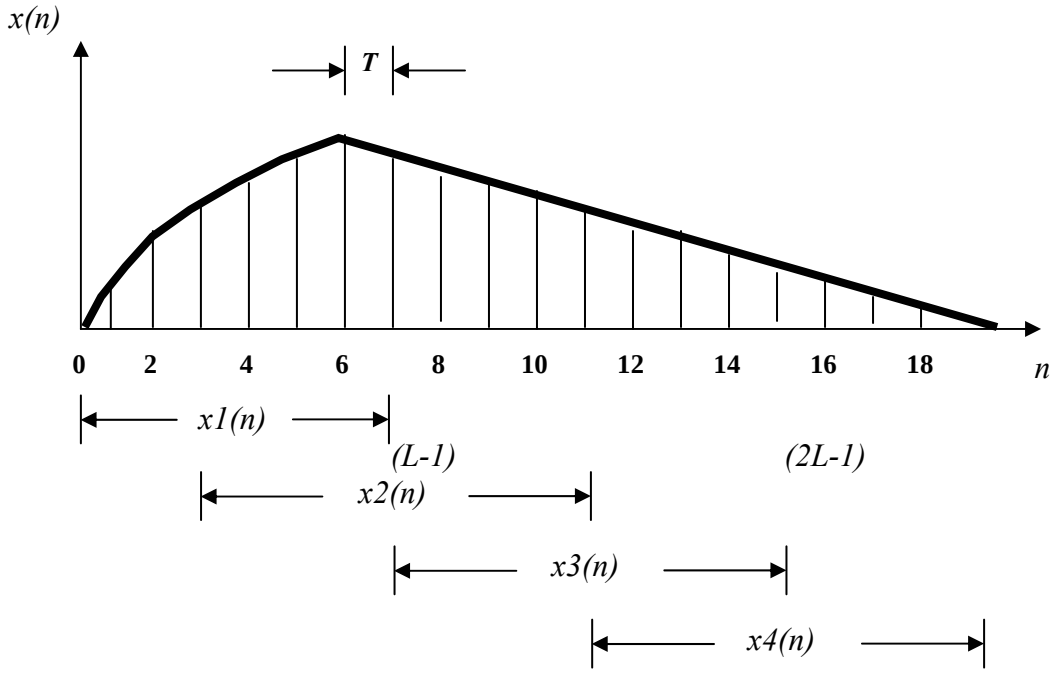

Figure 2 An illustration of subsectioning of data with 50% overlap and section length, *L = 8*.

$$\boldsymbol{P}(\boldsymbol{k}) = (1/\boldsymbol{N})|\boldsymbol{X}(\boldsymbol{k})|^2 = (1/\boldsymbol{N})\{\mathrm{Re}[\boldsymbol{X}(\boldsymbol{k})]^2 + \mathrm{Im}[\boldsymbol{X}(\boldsymbol{k})]^2\}\ \boldsymbol{k} = 0,1,...,\boldsymbol{N}/2 \quad (1)$$

where *N* is the number of data points. The PSD gives a measure of the distribution of the average power of a signal over frequency. If the waveform under investigation is long compared with the time interval over which it may be regarded as having constant statistical moments, then the spectrum estimate is likely to be inaccurate. This will also be the case when there is a large noise component in the waveform. It is then desirable to smooth the estimated spectrum to obtain an improved estimate. The purpose of spectrum smoothing is essential to remove randomness. Therefore, direct computation of the power spectrum as in Eq.1 may not yield good spectrum estimates because of random errors. To obtain consistent estimates of the PSD some form of averaging is normally used at some stage in the estimation process, for example correlation or frequency averaging. In the analyzer, we employed a method due to Welch (1967).

In the Welch method, the input data sequence (*x(n), n = 0, 1, …, N-1*) whose spectrum is to be estimated is first divided into *M* overlapping segments each of length *L*. For example, for a 50% overlap the data segments are (see Figure 2)

$$\boldsymbol{x_i(n) = x[n + iL/2], n = 0,1,2,...., L-1; i = 0,1,....., M-1} \tag{2}$$

The transform, *Xi(k),* and the raw power spectrum, $S_i(k)$, of each data segment are then obtained as

$$\boldsymbol{X_i(k) = \sum_{n=0}^{L-1} w(n) x_i(n) \exp(-j 2\pi nk/N), k = 0,1,...., L-1} \tag{3}$$

$$\boldsymbol{S_i(k) = \frac{1}{LQ} |X_i(k)|^2, k = 0,1,....., L/2} \tag{4}$$

where *w(n)* are the values of a suitable window function, for example the Hamming window, $x_i(n)$ is the *i*th data segment, and *Q* is the window energy defined as

$$\boldsymbol{Q = \frac{1}{L} \sum_{n=0}^{L-1} w^2(n)} \tag{5}$$

The average of the raw PSD estimates gives the desired 'smooth' PSD estimate of the data sequence, *x(n)*:

$$\begin{aligned} \boldsymbol{S(k)} &= \boldsymbol{\frac{1}{M} \sum_{i=0}^{M-1} S_i(k), k = 0,1,2,...., L/2} \\ \boldsymbol{S(k)} &= \boldsymbol{\frac{1}{MLQ} \sum_{i=0}^{M-1} |X_i(k)|^2, k = 0,1,2,....., L/2} \end{aligned} \tag{6}$$

The values of $X_i(k)$ in Eq.3 are of course obtained using the FFT for efficiency. To improve the spectral clarity of *S(k)*, the length, *L*, of each data segment may be extended, after windowing, to say *L+R* by adding *R* zeros, and the FFT of the zero-padded sequences then obtained.

In the analyzer three data segments were used. The segment length, *L*, was fixed at 1024 data points (that is zeros were added) and the overlap between segments was kept constant at half the segment length.

The input data sequence *x(n)* whose PSD is to be estimated is derived by sampling an analogue signal at a frequency, F_s, consistent with the sampling theorem:

$$\boldsymbol{Fs > 2f_{max};\ \ F_s = 1/T}$$

where *T* is the sampling interval and f_{max} the highest frequency in the analogue signal. When we transform *x(n)* into frequency as described above, the frequency spacing, Δf, between the frequency components is given by:

$$\boldsymbol{\Delta f = 1/LT} \tag{7}$$

This is the effective resolution of the FFT. For analyzer, L = 1024, and the maximum sampling frequency, F_s = 23 kHz, giving a resolution < 22.46 Hz.

3. Analyzer hardware

The block diagram of the system hardware is shown n Figure 3. Anti-aliasing filtering at the input is achieved using a second-order Butterworth filter for simplicity. Using this low-order anti-aliasing filter means that the sampling frequency should be sufficiently high to keep the effects of aliasing low.

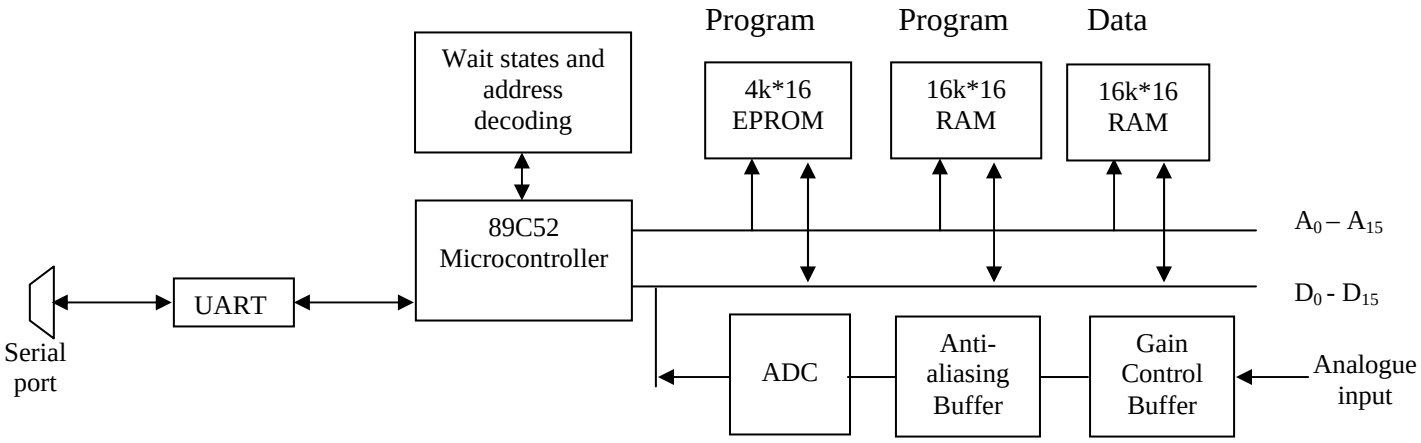

Figure 3 PC-based FFT spectrum analyzer hardware.

The program memory stores the FFT twiddle factors, precomputed sine and cosine values, as well as the window coefficients, *w(n)*, in Eq.3. The input data samples, *x(n)*, are stored first in the external data memory, and each block of data to be transformed is then transferred to the internal memory and stored in a complex format (that is, as real and imaginary); see the map for data storage in Figure 4. The FFT is performed in place. The spectrum analyzer board passes the spectrum estimates to the IBM PC for display through the serial port.

The 89C52 microcontroller target board meets all the analyzer hardware specifications and so was used to implement the analyzer.

4. Analyzer software

The analyzer software consists of a set of 89C52 microcontroller assembly language programs for computing the PSD values and transferring these to the PC, and a VB language program resident on the PC for displaying the power spectrum.

4.1. Microcontroller programs

The 89C52 programs carry out the following operations:

1. Collect N data points, $x(n)$, from the ADC (at the present, $N = 2048$) and save in the external data RAM.
2. Divide the N data points into three overlapping segments, 1024 data points each, see equation (2).
3. For each data segment perform the following operations:
   - window the data using a Hamming window
     $$\boldsymbol{x}_i(\boldsymbol{n}) = \boldsymbol{x}_i(\boldsymbol{n})\boldsymbol{w}(\boldsymbol{n}), \boldsymbol{n} = 0,1,.....,1023$$
   - transfer the windowed data into the internal RAM block B_1, and store as real and imaginary;
   - perform a 1024-point, in-place FFT on the data segment;
   - compute the raw PSD, see equation (4);

- save raw PSD in external RAM.

	B_0		B_1		B_2
0000	$x_i(0)$	1024	$x_i(0)$	2048	
	$x_i(1)$		*0*		
	⋮		$x_i(1)$		
	⋮		⋮		
	⋮		⋮		
	⋮		⋮		
	$x_i(1022)$	2046	$x_i(1023)$		
1023	$x_i(1023)$	2047	*0*	4095	
	Real Data		Complex data		Temporary storage

Figure 4 On-chip data storage for the analyzer.

4. Average the raw PSD for all segments, see equation (6).
5. Transfer averaged PSD to the PC.

The 89C52 code, written for the SXA51 assembler, consists of a main program, four subroutines, and an interrupt service routine:

- main routine;
- PSD initialization;
- interrupt service routine;
- data collection;
- PSD estimation;
- data transfer.

Some of the programs are described briefly below.

PSD initialization

The PSD initialization routine supplement the system initialization in the boot program. Its major role is to prepare the hardware for PSD computations. It performs the following operations:

- configure on-chip memory block B_0 as RAM;
- initialize internal data memories block B_0, B_1, B_2 with zero;
- program the timer for the correct sampling rate;
- unmask timer interrupt;
- enable the timer interrupt.

Programming the timer involves loading the 16-bit period register (TR), data memory location 3, with an appropriate value to set up the sampling frequency:

$$F_s = F_{clk} / N$$

where F_s is the sampling frequency, F_{clk} is the 89C52 clock out frequency, which is 12 MHZ, and N is a 16-bit unsigned integer loaded into the TR register. To generate a 23 kHz sampling frequency, the TR is loaded with 528.

Data Collection

After PSD initialization, the processor branches to the data collection subroutine where it goes into the idle state and waits until the timer interrupt occurs. On interrupt, the processor jumps to the interrupt service routine and issues a start of conversion (SC) command to the ADC by toggling the

external flag, XF. The processor then returns to the data collection subroutine and waits for the end of the conversion (EC) signal from the ADC by testing for BIO line to go low.
When the EC is received the data is read from the 12-bit ADC and stored in the external data RAM in Q15 format by shifting the data four places before it is stored in external data memory. The processor returns to the main routine when 2048 consecutive data samples have been collected.

Power spectrum density estimation

The FFT is the heart of the PSD estimation. It is a radix-2, 1024-point decimation in frequency (DIF) algorithm. For each data segment, the PSD routine initializes memory blocks B_0 and B_1 with zero and then moves the data from the external memory to the internal memory block B_0. The data is windowed by multiplying each data sample by an appropriate window coefficient and then transferred to the on-chip memory; block B_1, where it is stored in a complex form. The real data samples are stored in even addresses, starting at address 2048. The odd address locations, which contain zeros, represent the imaginary parts; see Figure 4. The complex data is then FFTed, in-place, and its raw PSD computed. The raw PSDs for the segments are then averaged to obtain a smooth PSD.

Data Transfer

The averaged PSD is transferred to the PC, via the serial interface, with each averaged PSD value represented as 2 bytes.

4.2. IBM PC display software

The VB language program on the PC reads the averaged PSD values from the analyzer hardware and displays it on the screen. When the program is first executed a menu is displayed requesting the user to select an option:

- display power spectrum density;
- save and display PSD;
- read PSD from disk and display;
- zoom;
- exit.

If option 1 and 2 is selected, control is passed to the main program in the analyzer, which computes the PSD and transfers it to the PC as described above. For option1, the PSD is then displayed. For option 2, the PSD values are first saved on the disk in a user-specified file, before the PSD is displayed. For option 3, the VB program first reads the PSD from the disk and then displays it. For option 4, the user is asked to specify the frequency range of interest, and the selected region of the current display is then expanded accordingly. Selecting option 5 returns the user to DOS.

5. Using the analyzer

Figures 5 and 6 show examples of the PSD of, respectively, a time record taken from the speech of a man says a specific sentence and the speech of a woman. Some of necessary computations are delivered with each figure. These computations are based on the obtained results (PSD) of each signal.

At the present the analyzer is used only for power spectrum estimation. It can be extended to perform other signal analysis or used in other applications, for example

- frequency response estimation;
- auto- and cross-spectral analysis of noisy or random signals;
- convolution in the frequency domain;
- correlation in the frequency domain;
- cross-spectral density;

- magnitude coherence, and

The data from the analyzer hardware may also be sent to the PC in various forms for further processing and/or display, for example as

- raw data;
- real and imaginary, and;
- magnitude and phase.

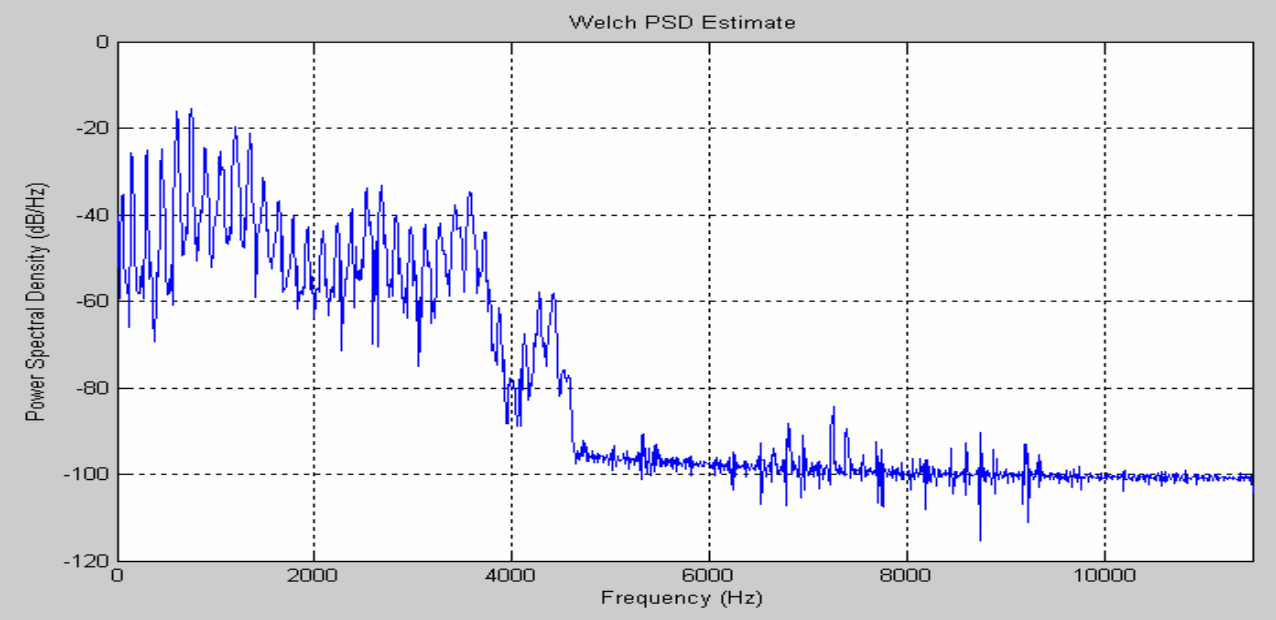

Figure 5 Frequency Spectrum of a man speech signal.
Minimum Power: -74.1245 dB, Maximum Power: -0.7077 dB, Average Power: -17.9878 dB
Signal-to-Noise Ratio: 58.9893 dB, Noise Floor: -52.2811 dB, Total Power: 8.0436 dB.

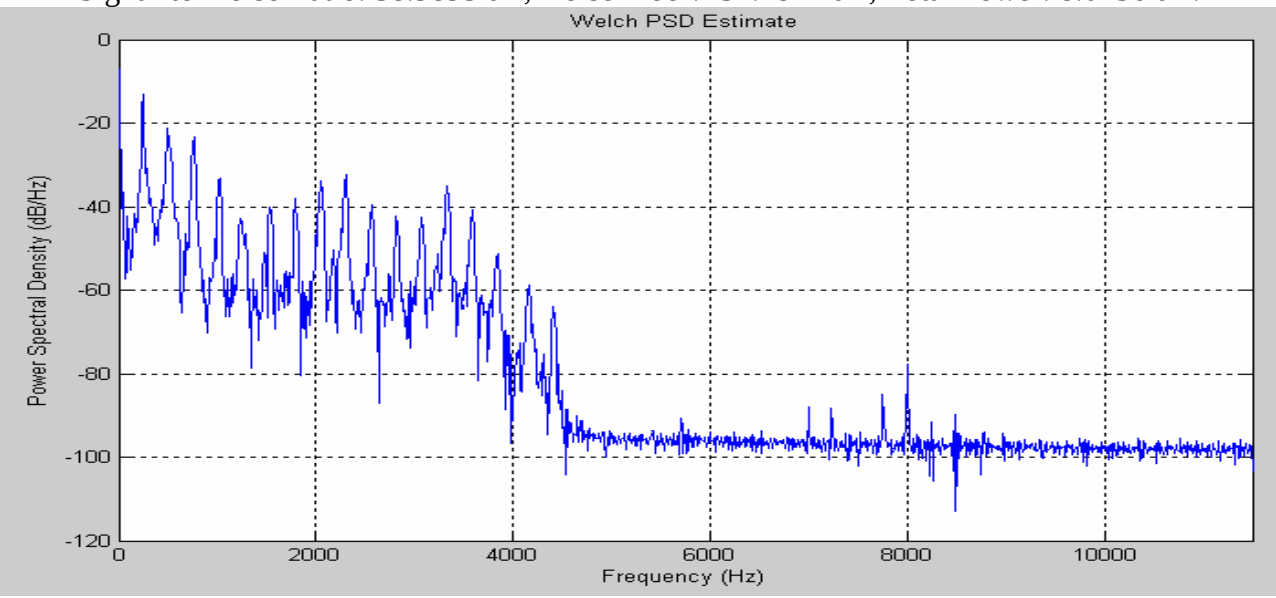

Figure 6 Frequency Spectrum of a woman speech signal.
Minimum Power: -81.8281 dB, Maximum Power: 7.8951 dB, Average Power: -15.1359 dB,
Signal-to-Noise Ratio: 63.6207 dB, Noise Floor: -54.0605 dB, Total Power: 10.8956 dB.

6. Conclusions

An FFT – based spectrum analysis procedure is proposed and implemented. The procedure is carried out using the interchangeable parts of program and data of the analyzer, and the communication ability with the PC. The PC is utilized for further processing and display. Also, it

incorporates software development tools and the software library as well as the software for communication and processing with the analyzer.

The processing operations for spectrum computation are carried out using the microcontroller. The microcontroller plays a good role in providing the processing ability for spectrum computation of stationary signals that we deal with in the power and power electronic fields and optimizing the cost and size of the system as compared with the DSP processors.

7. References

[1] Alan V. Oppenheim, and Ronald W. Schafer, Digital Signal Processing, Prentice-Hall, Inc., 1975.

[2] Alan V. Oppenheim, Ronald W. Schafter, and John R. Buck, Discrete Time Signal Processing, Prentice-Hall, Inc., 1999.

[3] Burrs, C.S., and Parks, T.W., DFT/FFT and Convolution Algorithms, New York: Wiley, 1985.

[4] Curtis D. Johnson, Microprocessor-Based Process Control, Prentice-Hall, Inc., 1984.

[5] Dale Grover, and John R. Deller, Digital Signal Processing and the Microcontroller, Prentice-Hall, Inc., 1999.

[6] Dimitris G. Manolakis, Vinay K. Ingle and Stephen M. Kogon, Statistical and Adaptive Signal Processing, Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing, McGraw-Hill Companies, Inc., 2000.

[7] Fredric J. Harris, On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform, Proceedings of the IEEE, vol., 66, No. 1, January 1978.

[8] Kenneth J. Ayala, The 8051 Microcontroller, Architecture, Programming and Application, West Publishing Co., 1997.

[9] Larry D. Jones and A. Foster Chin, Electronic Instruments and Measurements, John Wiley & Sons, 1983.

[10] Lonnine C. Ludeman, Fundamentals of Digital Signal Processing, John Wiley & Sons, 1987.

[11] Michael Cerna and Audrey F. Harvey, The Fundamentals of FFT-Based Signal Analysis and Measurement, National Instruments, July 2000.

[12] Rodriguez, Jeffrey J., An improved FFT Digit-Reversal Algorithm, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol.37, No.8, Aug 1989.

[13] Sanjit K. Mitra, Digital Signal Processing, A computer-Based Approach, McGraw-Hill Co.,