DEVELOPMENT OF A MULTI-CHANNEL PERSONAL COMPUTER (PC) BASED DATA LOGGER

A.L. Popoola
Department of physics, Federal University of Technology
P.M.B. 704, Akure. Ondo State, Nigeria.

ABSTRACT
PC (Personal Computer) based instrumentation is getting more attention because of three technological enhancements; increasing reliability of PCs, steadily decreasing cost of hard drive space on PCs and PC-based measurement hardware that could meet or exceed measurement capabilities of stand-alone data loggers. This paper discusses the development of a circuitry that is capable of acquiring data from sixteen different sources external to the computer. The circuitry was connected to a personal computer through the printer's port, and the acquisition time interval of the circuitry was variable with a minimum of one minute. The complete activity of the circuitry was controlled by software written in visual basic 6.0. The circuitry was used to monitor atmospheric temperature and relative humidity. The sets of data obtained were found to be accurate and reliable, with the exact time and date of acquisition automatically recorded. This design revealed almost 50% cut in the cost of using this technology when compared with the cost of importing a data logger of comparable feature.

Keywords: *Sample and Hold, signal conditioners, cascading, synchronization, FETs.*

INTRODUCTION

An important application of the personal computer is to monitor and control experiments in real time. In these applications, the PC is programmed to sense what is happening in the connected experiment and to issue control signals in response to the sensed situation. The PC can be accessed through its numerous input-output (I/O) facilities. However, the ease with which it is possible to interface a particular application to the computer depends largely on the arrangements of its input-output (I/O) features and the provision of the correct interfaces, both in hardware and software to ensure the compatibility of signal and the control of information to and from the devices attached to the computer (Crithclow, 1985 and Anderson, 2001).

Real time PC based data logging requires a special interface (Fig. 1) to connect the experimental instrument to the computer. PC-based data log-

Fig 1: Block diagram of an interface

ging permits high rate of data collection and online parameter adjustments. Also, the availability of a disk drive on PC's greatly increases the quantity of data that can be collected (National Instruments, 2001). Figure 2 shows the signal flow path in a real-time PC-based data collection system. Generally, most of the signals of interest in any data collection experiment are analogue in nature. Therefore, the basis of any measuring instrument is a transducer that converts these analogue signals to a form measurable by the instrument. For computer interface design, transducers and instruments that convert other forms of energy to electrical energy are of primary interest.

However, depending on the design of the interface, it is desirable to convert all of the analogue electrical signals into a common form and range. The various devices used to produce this result are usually classified as signal conditioners. The functions provided are voltage and current amplification, impedance transformation, calibration and referencing (National Instruments, 2001). Moreover, since the physical phenomenon of interest to be measured varies from one application/ experiment to another, the method of processing/ conditioning these parameters also differs. Therefore, the circuitry discussed in this paper centers only on the areas that is common to all forms of PC-based data logging irrespective of the variable to be logged.

CIRCUIT DESCRIPTION

The complete circuit diagram of the logger is shown in Fig. 3. The schematic contains one pro-

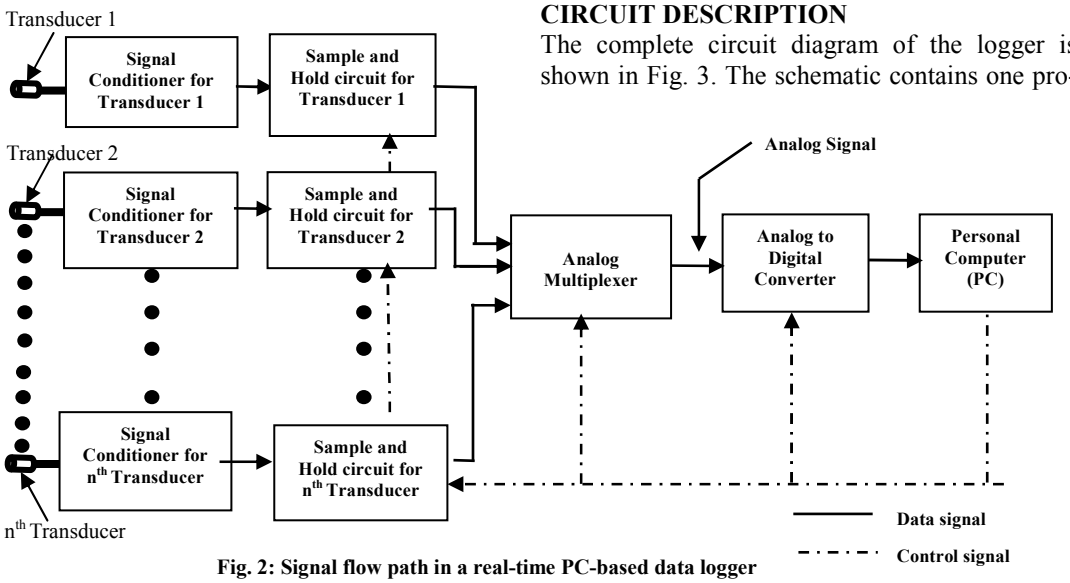

Fig. 2: Signal flow path in a real-time PC-based data logger

grammable peripheral interface adapter IC (Intel-8255), two cascaded one-of-four decoder IC's (4555), four CMOS analog switches (MC14066) and sixteen sample and hold IC's (LF398). Each of these devices is presented in the following section.

The Sample and Hold Circuit

In this design, since it was required to convert many analog signals at exactly the same instant in time, one way to do this would have been to provide a separate Analogue-to-Digital Converter (ADC) on each analog input channel and to synchronize the "*start conversion*" on each unit. However, this approach would be prohibitively expensive. A more attractive technique adopted was to provide simultaneous sample and hold capability, whereby a sample and hold circuit was connected to each analog input channel. The output of each sample and hold (S/H) was then connected through a succeeding multiplexer to the ADC.

The sample and hold was used as a "track-and-hold", freezing the analog signal only when conversion was about to begin. It acquires an analog signal at the precise time dictated by a digital control signal. The held signal was made available to the multiplexer until the S/H's were subsequently commanded to store a new data. The LF398 IC used in this design sampled/track signals when its pin 8 was switched high (5V). It holds its acquired data when pin 8 went low (0V). With a 1000pF polystyrene capacitor connected between pin 6 of the IC and ground, the acquisition time of the IC was less than 6μS following "SAMPLE" command (National Semiconductor, 2000). The held signal droops/leaks less than 25μV during the subsequent 100μS conversion time of the ADC. Several advantages were gained by using the sample and hold technique. Extremely high precision in the timing of the converter was obtained. The time of occurrence of the ADC converted signal was known precisely and the accuracy of the converted analog signal was high. It should be emphasized that to ensure that the sampled analog input was held effectively, the holding capacitor must be a high-quality low leakage type; usually a polypropylene or polystyrene is best (Loveday, 1984; Short, 1988).

The Analog Multiplexer

It is usually expedient to share a single ADC among a number of analog signal sources. An analog multiplexer is an electronic or electromechanical switch that permits a number of analog signals to be connected to the ADC. Under program control, the multiplexer connects selected analog input channels to the ADC. The major concerns in choosing a multiplexer are (Finkel, 1975; Horowitz and Winfield, 1995):

i) High input impedance – the multiplexer must present high input impedance to the signal coming into it. This is necessary to prevent the signal from being over-loaded, which can result into inaccurate readings.
ii) The speed of transfer of a signal across the analog multiplexer must be commensurate with the speed of the ADC being used.
iii) The multiplexer must possess fast settling time – this is the time required for the output of the multiplexer to have come within a specified percentage of the connected input.

Fast settling time lends itself to accurate conversions as well as high speed. However, since a single monolithic sixteen-channel analog multiplexer was not available for use during the design of this circuitry, one was constructed from available linear and digital ICs. The multiplexer comprises of a Programmable Peripheral Interface Adapter IC (8255), two cascaded address decoder ICs (4555) and four bilateral IC switch (MC14066). The discussion of these various subunits is presented as follows:

The 8255 Programmable Peripheral Interface Adapter

Intel Corporation originally manufactured this IC; the chip is now available from many manufacturers. It has 24 input-output pins, which may be

individually programmed in two groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals.

In this design, our choice of 8255 and its subsequent use in MODE 0 were due to some obvious reasons. One reason was that the 8255 could interface to many different microprocessors (Coffron and Long, 1983). The second reason is that the device has being in use for several years; hence, it is well documented, readily available and cheap. The third reason is that the outputs of the IC are latched (Gaonkar, 1986). The chip was used in MODE 0 and to do this, control word was written into the control register of the chip. The control word internally formats the chip – that is, it defines how the I/O pins should behave electrically.

The 4555 Address Decoder – Each of this IC has two separate 1-of-4-decoder units. Thus, to achieve 1-of-8-decoding operation with this IC, an inverter (4069) was connected in-between the enable inputs (pins 1 and 15) of the separate units. However, since our requirement was to have 1-of-16-decoder, two 4555 connected as 1-of-8-decoder was cascaded together and a switching arrangement was devised to switch them alternately. The first 4555 when switched addressed the first 8 channels while the second 4555 addressed the last 8 channels. The switch was constructed with a resistor R1 (220Ω), a transistor (BC 109) and a relay. The output of the address decoder was connected to the control inputs of bilateral switches housed in a 4066 IC. Depending upon the bit pattern at the input of the address decoder, only one of its outputs would be high (9V) at a time and the switch whose control bit corresponded to this high position had its channel selected.

The 4066 CMOS Analog Transmission Gate – This IC was used in the design because Field Effect Transistors (FETs), particularly Metal-Oxide Silicon Field Effect Transistors (MOSFETs) exhibits low ON resistance (all the way to zero volts), extremely high OFF resistance, low leakage currents and low capacitance for analog signals. The IC in its technology makes use of paralleled complementary MOSFETs (CMOS "Complementary Metal-Oxide Silicon") switches because of the need to switch signals that may go nearly to the supply voltages. When the IC is in this configuration, it is popularly called a "transmission gate" because either of its terminals can be the input and the signal it switches is between ground and a single positive rail (Horowitz and Winfield, 1995).

The ADC0804

Analog-to-digital converters (ADCs) are used to convert analogue signals to a form that can be accepted by digital computers. An ADC accepts analog voltages as inputs and produces the digital representation of the input signal as its output. ADC exists in different technologies, ranging from counting, integrating, parallel/flash and to successive approximation types (Franco, 1988; Crecraft, Gorham and Sparkes, 1993). The ADC0804 uses successive approximation technique for analog-to-digital conversion (National Semiconductor, 2001), which makes the ADC to operate at relatively faster speed when compared with ADC's that uses counting or integrating data conversion techniques. The ADC0804 is a 20-pin dual-in-line package with the following additional impressive features (Reis, 1991; National Semiconductor, 2001):

i) No zero adjustment is required in its operation
ii) It has separate ***read*** and ***write*** enable inputs, which makes it suitable for easy interface to all microprocessors.
iii) It has an on-chip clock, whose frequency is determined by an external resistor and capacitor. The frequency of the clock is calculated by the formula given by (1):

$$F = \left(\frac{1}{1.1}\right) \times R \times C \qquad (1)$$

iv) The data output of the ADC is 8-bits wide. This factor was an impressive one as it greatly reduced the complexity and size of the control software.

The ADC was wired for unipolar (0 – 5V) input. With ***R*** (10kΩ) and ***C*** (10pF) the ADC operated with a conversion time of 100μS.

Development of Control Software

A program whose control algorithm was prepared using Visual Basic 6.0 handled the complete operation of the data logger. Visual basic 6.0 was used as the platform for developing the control software because of its intuitive user interface. The software flowchart (Fig. 4) was optimized for *automatic data storage* and *archiving*, *data export* capability, *event management*, and *display* tools. There were two interfacing forms in the software, the main form (Fig. 5) and the calibration form (Fig. 6). There were 16 check boxes, 16 text boxes and 3 command buttons on the main form. All the sixteen check boxes allow the user to specify the channels he/she wants to acquire data from. Besides each check box were text boxes, which display in real time the processed version of the acquired data.

The "start" command button when clicked runs the software by first displaying a message box, to confirm the logging interval specified by the user. It thereafter initiates data acquisition and logging process. The "Exit" button terminates the program, while the "stop" button temporarily halts the program, should there be any need for adjustment in the parameters specified. Also displayed on the form were current time and date.

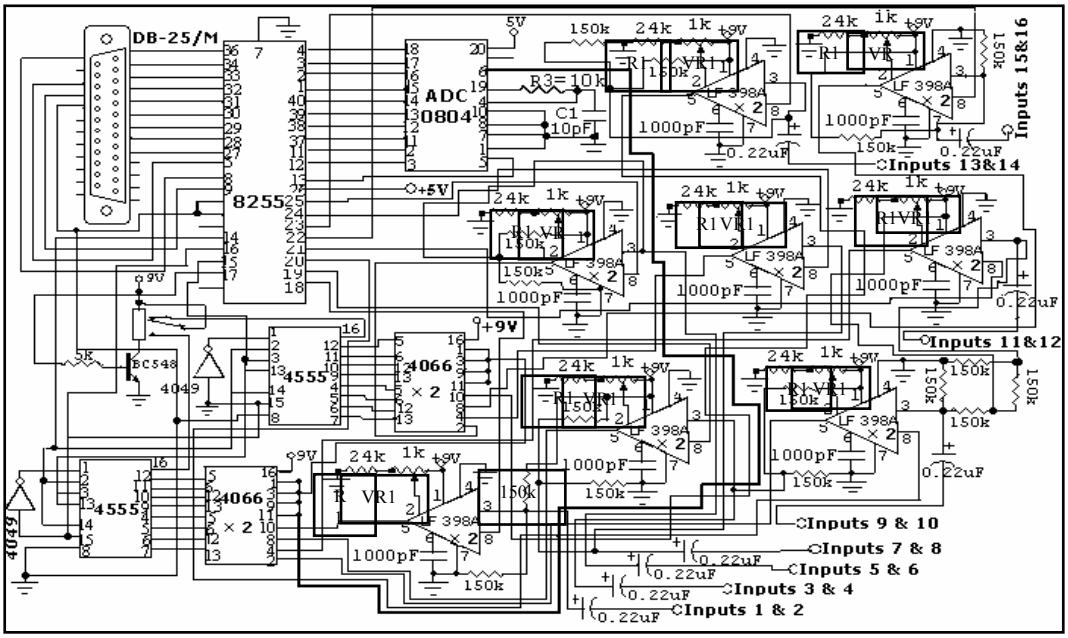

Fig. 3: Circuit diagram of the PC-based data logger

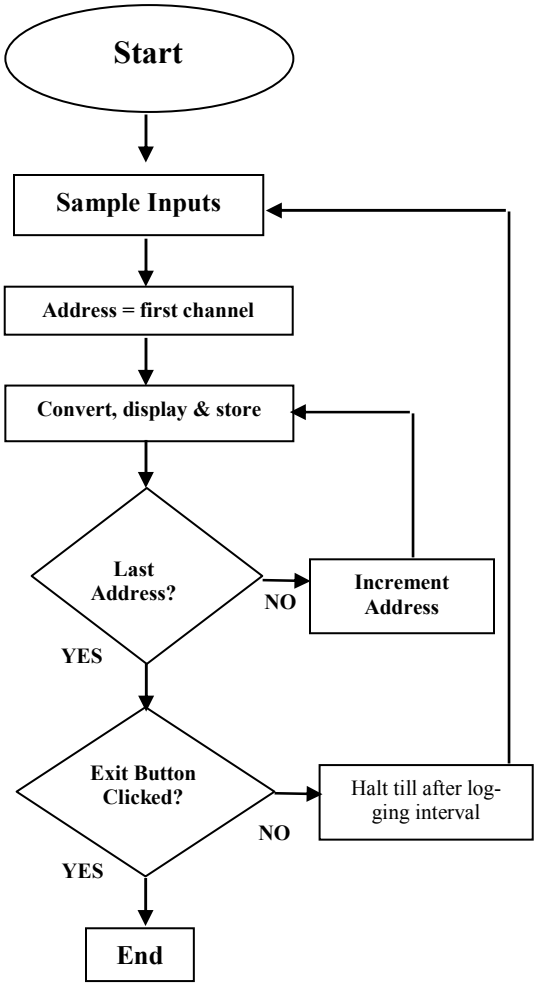

Fig. 4: The software flowchart

Operation – When the "*start"* button is clicked, a file is opened for both read and writes operations. Setting up a counter and initiating a timing sequence subsequently follows. At appropriate times, the control program calls a dynamic-link-library (port95nt), which serves as a link between the control software and the input-output ports of the PC. Next, the first channel was read and captured as a variable. The variable was processed and displayed immediately on the text box that corresponds to the channel been processed. The processed data was at the same time sent to the hard disk for permanent storage. The counter was incremented and the next channel was read, processed and stored. This process continues until all the channels has been read, processed and stored. After this, the counter resets to zero and the whole process repeats again until the program is interrupted either by clicking stop or exit button.

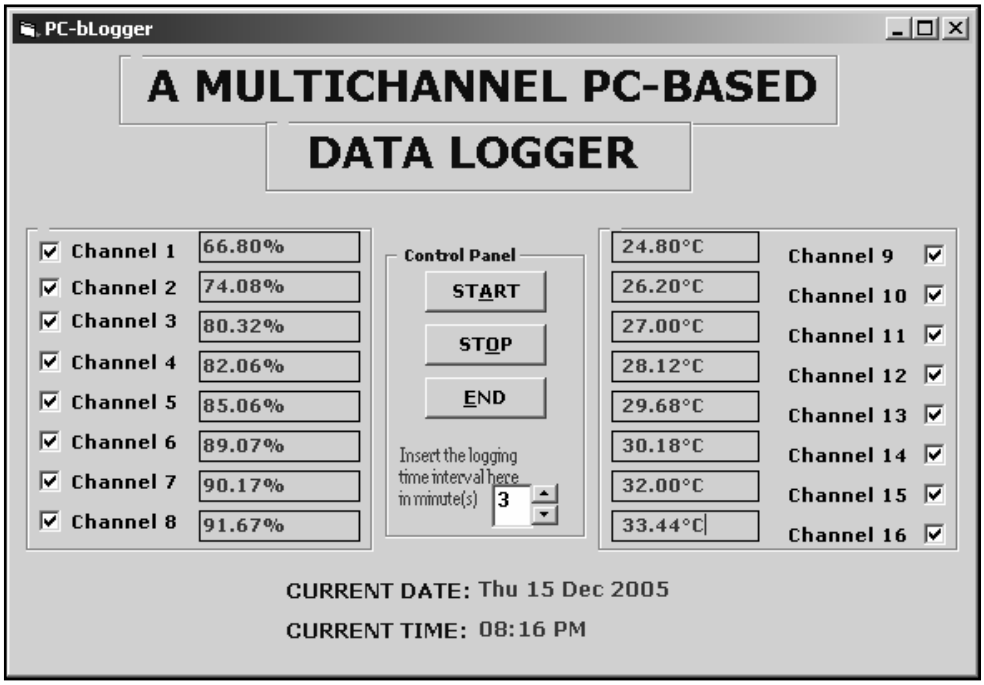

Fig. 5: The main user interface for the PC-based logger

Calibration of the data logger

The calibration of the logger was done in two ways. The first step was to offset all unwanted voltages caused by the various components used in the construction of the circuitry. To obtain full 8-bit accuracy at the computer port, an offset trim was provided to cater/compensate for three errors. The first was the error due to the sample and holds used. According to its manufacturer (National Semiconductor, 2000), the S/H

has an output offset voltage V_{OS} of 7mV (max). Also, the S/H introduces a small voltage step following HOLD command. This is due to FET gate charge injection (Horowitz and Winfield, 1995). In this case, a relatively stable 0.05mV negative step occurs. The third source of error was from the ADC. The manufacturers data sheet (National Semiconductor, 2001), has a maximum V_{OS} specified value of ± 1LSB (equivalent to 1.25mV for 0–5V input range). For optimum performance, the trimming circuit was put on the LF398, using the manufacturers recommended network. The trim network comprises of the offset trim variable resistor VR1 (1kΩ) and R1 (24KΩ) on all the sample-and-holds (Fig. 3). To achieve the offset process, a voltage of 0Volt was applied to channel 1 and the offset trim on the S/H for the channel was adjusted until the converted output of the ADC all read zero. This process was repeated for other channels.

The second calibration procedure was done through the software. The calibration interface form (Fig. 4) facilitates the use of this added feature. Located on the form are instructions for using the form, one text box, and a command button. A known voltage (say 500mV) was placed at the inputs of all the sixteen samples and hold, and the same voltage was entered into the text box

named *channel* on the calibrator form and the "OFFSET" button was clicked. The calibrator compares what was entered into the channel text box with the value acquired/decoded from each channel of the logger. It then divides the 500mV entered into the *channel* text box by whatever the software acquires from the logger. The arithmetic was used to generate a correcting factor that serves as a multiplier to other subsequent readings obtained from the logger. This approach further compensates for any error that could arise after the initial offset trimming.

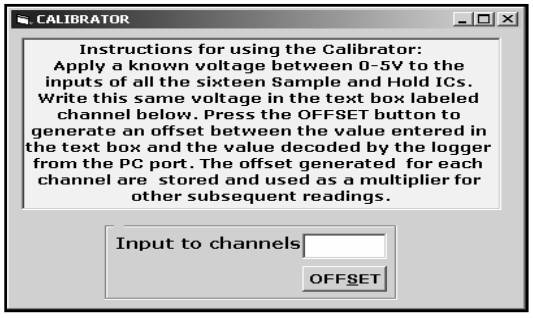

Figure 6: Interface form for software calibration of the logger

RESULT

To explore the viability of the logger, two sensors [LM35 for temperature and HIH3610-001 for relative humidity] were connected to the logger and the whole unit was used to monitor atmospheric temperature and relative humidity around physics department complex, Federal University of Technology Akure (7.15˚N, 5.12˚E). An extract from the results obtained (shown in Tables 1 and 2) was accurate when compared with that of an imported logger (Davis-vantage Pro) installed within the complex.

Table 1: Compared humidity readings obtained from the constructed logger and Davis Vantage Pro

Readings from Davis vantage Pro/%	Readings from the constructed logger/%
67.00	66.98
74.22	74.17
78.00	78.00
80.00	80.00
82.50	82.30
85.00	84.89
87.00	87.00
90.00	89.75

Table 2: Compared Temperature readings obtained from the constructed logger and Davis vantage Pro.

Readings from Davis vantage Pro/˚C	Readings from the constructed logger/˚C
25.00	24.98
26.00	25.99
27.00	26.98
28.00	28.00
29.00	29.00
30.00	30.00
31.00	31.00
32.00	32.06
33.00	33.08
34.00	34.10

CONCLUSIONS

Today, real-time data acquisition systems are becoming more complex, expensive and scarce especially in developing countries like Nigeria. PC-based logging systems still provide the widest range of measurement types, analysis capabilities and reporting tools. The development of this PC-based data logger was embarked upon to optimize the use of the existing PCs in our laboratory in experiment monitoring and real-time data acquisition. The main focus was to find an alternative approach to data collection, processing and storage at little or no extra cost in the absence of an equivalent data logger. The objective of the study was realized and the logger is presently being used in many of our data acquisition experiments. The circuitry presented in this paper is one of the ways to achieve cost effective PC-based data logging. The number of channel on the data logger can be easily increased by putting additional decoder IC, bilateral switch and sample and

hold ICs. The parallel interface used between the logger and the PC can be supplanted with a serial interface if the logger is to be located at a far distance from the PC using any popular Universal Synchronous/Asynchronous Receiver/ Transmitter (UART or USART) chips.

Finally, since the set of data obtained from the developed data logger is accurate and reliable, and the cost of achieving its construction is low ($135) compared with $450 needed to import a data logger of comparable feature, it will be helpful if scientist, engineers and experimenters becomes familiar with computer interfacing concepts and techniques discussed in this paper, so that the cost of importing data logger for every new data collection experiment will be eliminated or reduce drastically.

ACKNOWLEDGEMENTS

The author wishes to show his appreciation to Scientific Software Tools Inc. 19E. Center Avenue, Paoli. PA 19301 for supplying the DriverLinx Port I/O (port95nt) free of charge.

REFERENCES

Anderson P. H. (2001). Use of a PC Printer Port for Control and Data Acquisition. The Electronic Journal for Engineering Technology, the Technology Interface/Fall 96. www.access.digex.net/~pha

Coffron J. W. and Long W. E. (1983). Practical Interfacing Technique for Microprocessor Systems. Prentice-Hall, Inc. New Jersey, USA: pp. 261-271.

Crecraft D. I., Sparks J. J. and Thompson M. (1993). Electronics. Chapman and Hall. London: pp. 307-315.

Crithclow A. J. (1985). Introduction to Robotics. Macmillan Publishers New York, USA.

Finkel J. (1975). Computer-Aided Experimentation; *Interfacing to Microprocessor*. John Wiley and Sons, New York..

Franco S. (1988). Designing with operational amplifiers and analog ICs. McGraw-Hill publishers, New York: pp. 335-340, 556-559.

Gaonkar R. S. (1986). Microprocessor Architecture, Programming and Applications with the 8080/8085. Wiley Eastern Ltd. New Delhi: pp. 439-449.

Horowitz P. and Winfield H. (1995). The Art of Electronics. Cambridge University Press, Cambridge. UK. Second edition: pp. 551-552, 635-638.

Loveday G. C. (1984). Practical Interface Circuits for Micros. Pitman Publishing limited, London. Pg. 126.

National Instruments (2001). A Review of PC-Based Data Logging and Recording Techniques. Published by National Instruments, 11500 North Mopac Expressway Austin TX 78759-3504 USA. (www.ni.com).

National Semiconductors (2000, 2001). LF398 and ADC0804 application and data sheets. (www.national.com).

Reis R. A. (1991). Digital electronics through project analysis. Macmillan Publishers. New York, USA: pp. 321-338, 393-396.

Short K. L. (1988). Microprocessors and Programmed Logic. Prentice-Hall of India, New Delhi. Second edition. pp. 430-452.