VOL.1, NO.1, JUNE 2006 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

# Design of a Low Cost Multi Channel Data Logger

Sumon Saha[1], Md. Tofiqul Islam[1] and M Zakir Hossain[1]
[1]Department of Mechanical Engineering
Bangladesh University of Engineering and Technology
Dhaka-1000, Bangladesh
E-mail: sumonsaha@me.buet.ac.bd

## ABSTRACT

Development of a low-cost multi-channel (eight to twenty two channels) data logger can easily be made and easily be used to convert the analog signal of physical parameters of various tests or other purposes of engineering. By a suitable program code it can be used to read the value digitally with a PC. Our aim is to provide with a module and a software package when installed in a computer, one can remotely acquire and monitor several numbers of the same or different types of signals sequentially at a time. Signals obtained from various sensors have been effectively conditioned. Now interfacing these signals using an ADC with the parallel port of a computer satisfies the very goal of data acquisition. The user friendliness and reliability in using a PC and channel selector multiplexers further add to the versatility of the data logger. Design and implementation of such equipment cost only at US$30, makes it very inexpensive comparative to other commercially available data loggers.

**Keywords:** Data logger, parallel port, multiplexers.

## INTRODUCTION

Data logging and recording is a very common measurement application. In its most basic form, data logging is the measurement and recording of physical or electrical parameters from sensors over a period of time. The data can be temperature, strain, displacement, flow, pressure, voltage, current, resistance, power, or any of a wide range of other parameters. Real-world data logging applications are typically more involved than just acquiring and recording signals, normally involving some combination of online analysis, offline analysis, display, report generation, and data sharing.

Taking manual measurements from analog instruments such as thermometers and manometers covered the common requirements of prehistoric data logging system. These measurements were recorded into a written log, along with the time of observation. To view trends over time, people manually plotted their measurements on graph paper. In the late 19th century, it became possible to begin automating this process with machines, and strip chart recorders evolved. With the development of the personal computer in the 1970s and 80s, people began to use computers for analysis of data, data storage, and report generation. The need to bring data into the PC brought about data loggers – a new special-purpose device for data logging. In the 1990s, a further evolution in data logging took place as people began to create PC-based data logging systems. The move to PC-based data logging systems was enabled by three technological enhancements:

i. Increasing reliability of PCs.
ii. Steadily decreasing cost of hard drive space on PCs.
iii. PC-based measurement hardware that could meet or exceed measurement capabilities of stand-alone data loggers.

Data logging is used in a broad spectrum of applications. Chemists record data such as temperature, pH, and pressure when performing experiments in a lab. Design engineers log performance parameters such as vibration, temperature, and battery level to evaluate product designs. Civil engineers record strain and load on bridges over time to evaluate safety. Geologists use data logging to determine mineral formations when drilling for oil. Breweries log the conditions of their storage and brewing facilities to maintain quality. The list of applications for data logging goes on and on, but all of these applications have similar common requirements.

The purpose of this paper is to provide a general background on data logging, discuss the various functional requirements that are common to most logging applications, and design an inexpensive multi-channel data logger in order to examine most of the modern hardware and software options available to scientists and engineers for implementing PC-based data logging systems.

## TYPICAL DESIGN CONSIDERATIONS FOR A DATALOGGER

The basic equipment for computer-based measurement consists of sensors, a data-logger and a computer. Normally, the sensors are plugged into the data-logger circuit box, which in turn is connected to the computer. Optionally, a printer is also useful for printing out graphs on paper. A crucially important component is a program for the computer; this is needed for managing the collection, display, storage and analysis of data.

VOL.1, NO.1, JUNE 2006 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

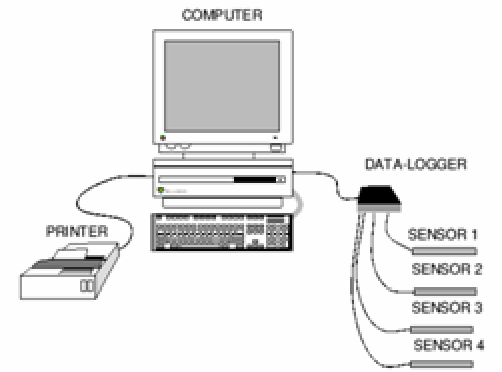



**Figure-1.** Main Component of Data Logger.

**Sensors**

The purpose of the sensor is to convert a physical quantity into an electrical signal, which can be translated by the computer into numerical values and graphical images. A wide variety of sensors are now available. For example, temperature, light intensity, sound level, angle of rotation, position, relative humidity, pH, dissolved oxygen, pulse (heart rate), breathing, wind speed, and motion.

**Data Logger**

A data-logger contains an electrical circuit, which acts as an intermediary between the sensors and the microcomputer. PC based data loggers need to be first connected to a computer so that they can receive a program of instructions for collecting and storing data from the sensors. Having received its instructions, it can be connected to the experimental data through the sensors it is going to monitor. During performing the experiment, the data-logger is concurrently connected to the computer so that the computer can retrieve the data and display it as a graph. This type of application is very useful for fieldwork and shorter-term experiments.

**Software**

In order to use sensors for taking readings in a science experiment, a data-logging program must be loaded into the computer. The common functions of a data-logging program are as follows:

i. Take readings from the sensors at regular time intervals ranging from a few microseconds to hours or even days.
ii. Present the data in a suitable form of display on the screen. This could take the form of tables of numerical values, large digits, bar charts or most commonly as conventional graphs. Also, the program provides aids for accurately analyzing the results and performing calculations on the data.
iii. Print the data table or graph on paper using a printer connected to the computer.
iv. Store the data on a computer disc for future use.

**DATA LOGGING FUNCTIONAL REQUIREMENT**

Every data logging application, from manually recording weather patterns in the 15th century, to logging the experimental parameters of a fusion reactor test in the 21st century can be broken down into a set of five common functional requirements, illustrated in Figure-2.

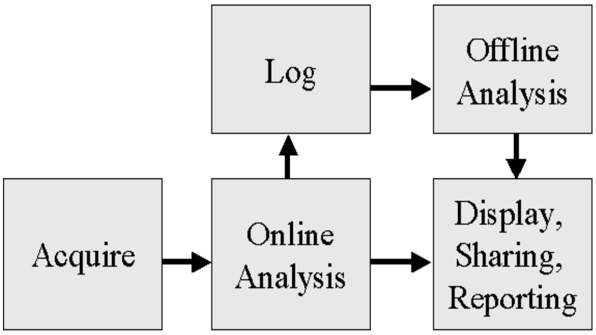



**Figure-2.** Functional Requirements of a Data Logging System.

**Acquisition**

The acquisition function is one of the most critical tasks of every data logging system. In a PC-based system, the acquisition is accomplished by the measurement hardware, which can be further broken down into sensors, signal connectivity, signal conditioning, and analog-to-digital conversion. Signal conditioning is one of the most important and overlooked components of a PC-based data logging system. Most signals require some form of preparation (conditioning) before they can be digitized. Some common types of signal conditioning are amplification, attenuation, isolation, multiplexing, filtering, excitation, linearization, cold-junction compensation, simultaneous sampling. Most sensors require a combination of these signal-conditioning technologies. After physical parameters have been converted into electrical signals and properly conditioned, it is time to convert the analog electrical signals into digital values and passes those values to the computer. The analog-to-digital conversion can be accomplished with either a plug-in data acquisition (DAQ) board, or it can be integrated into a single package with the conditioning and connectivity.

**Online Analysis**

In PC-based systems, online analysis is accomplished through software. Many different forms of online analysis can be needed in various data logging applications. Channel scaling, one of the most common forms of online analysis, is the conversion of the raw binary values returned by the acquisition system into properly scaled measurements with appropriate engineering units. Another important online analysis function is alarming and event management, which typically means monitoring a channel and providing some notification if limits are exceeded.

VOL.1, NO.1, JUNE 2006 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

**Logging and Storage**

The logging (or storage) functional block is, by definition, required in every data logging system. PC-based data logging systems typically use the hard drive of the PC, although they can also use tape drives, network drives, RAID drives, and other more exotic options. Software is of critical importance in PC-based data logging systems, because well-written logging software determines how data is stored, how quickly data can be written to disk, and how efficiently disk space is used. Logging software also gives data management capabilities, such as changing data formats, archiving data, and access to databases.

**Offline Analysis**

Offline analysis is performing mathematical functions on data after it has been acquired in order to extract important information. Types of offline analysis can include computing basic statistics of measured parameters, as well as more advanced functions such as the frequency content of signals and order analysis.

**Display and Report Generation**

Most data logging applications require some form of display to view the measurements that are being recorded. The display function can be further broken down to viewing live data and historical data. Live data display is necessary if one needs to view data as it is being acquired. Historical display lets one view data that was previously acquired. Report generation can be integrated into PC-based data logging applications for increased efficiency. With the networking capabilities found in modern data logging software, sharing data and publishing it to the network no longer requires a degree in computer science.

**EXPERIMENTAL SET UP**

All PC-based data logging systems are made up of hardware and software. The measurement hardware handles the acquisition portion of the logging application, and the hardware choice defines channel count, sensor type, acquisition speed, and measurement accuracy. The measurement software, in addition to controlling the hardware, also handles the online analysis, logging, offline analysis, display, reporting, and data sharing.

**Hardware**

The hardware portion of the data logger covered the electrical part of the whole system, which can be divided into three sub parts:

i. Circuit for channel selection,
ii. Circuit for conversion,
iii. Circuit for computer interfacing

The heart of a data logging system is an A/D converter. The speed and resolution of the A/D converter is chosen based on the signal frequency. To build up a multi-channel data logger, the primary selection of ADC0808 which has inbuilt 8 channel multiplexer, satisfy the very goal of data acquisition when more than one signal is required. But further using two channel-selector ICs such as CD4051, which is an analog multiplexer, increases the number of input channels up to 22. The pin configuration of an 8-channel analog MUX is shown in Figure-3.

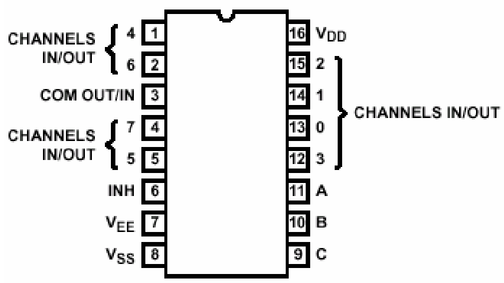
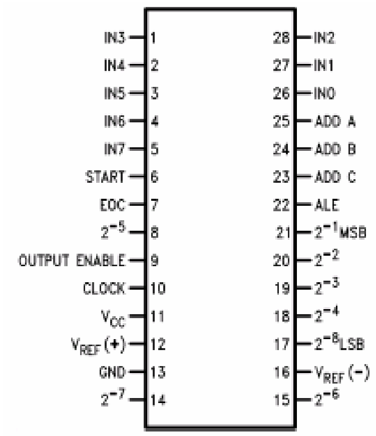



**Figure-3.** Pin Configuration of CD 4051

The ADC0808 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. It offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. The pin configuration of ADC0808 data acquisition chip is shown in Figure-4. The common outputs of two CD4051 multiplexers are connected to the first two input channels (IN0 and IN1) of ADC0808, makes a total of 16 input channels and using rest 6 analog input channels of ADC from IN3 to IN7 results 22 channels.




ADC0808

**Figure-4.** An 8-channel 8-bit data acquisition chip.

The voltage references REF+ and REF-determine the range of input signal for conversion. An external clock using 555-timer circuit was connected to the CLOCK pin. The A/D converter starts conversion of the input signal when a positive high pulse is applied to the START pin. This operation was performed by software connecting C2 control pin of LPT1 with START as well as ALE pin of ADC. It needs several clock cycles of min 10 kHz, max 1280 kHz and typically 640 kHz to complete the conversion. The 8-bit result of conversion process is placed in an output buffer. This buffer will be connected to the data bus D0 ($2^{-8}$) to D7 ($2^{-1}$). The ALE (Address latch enable) signal pin is provided for easy interface with computer.

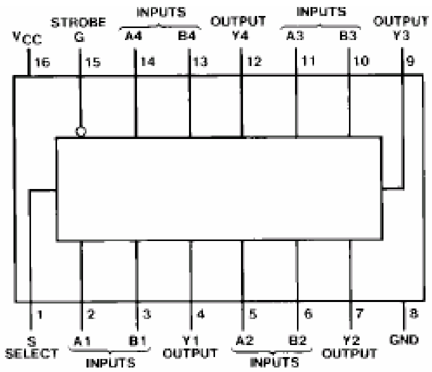



**Figure-5.** Pin Configuration of DM74LS157

ADC0808 operates on a single ended 5V power supply. The 8-bit data pins D0-D7 are connected to the input A1-A4 and B1-B4 pins of digital multiplexer 74157. These data selector/multiplexer contains inverters and drivers to supply full on-chip data selection to the four output gates. A separate strobe input is provided. A 4-bit word is selected from one of two sources and is routed to the four outputs. The pin configuration of DM74LS157 is shown in Figure-5.

Use of printer port for computer interfacing adds the enhancement of high-speed data acquisition. Each printer port consists of three port addresses: data, status and control port. These addresses are in sequential order. That is, if the data port is at address 0x0378, the corresponding status port is at 0x0379 and the control port is at 0x37a. The pin configuration of a parallel port is shown in Figure-6.

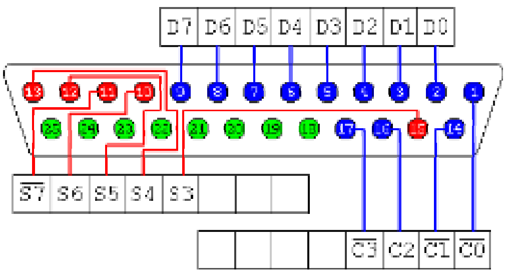



**Figure-6.** Pin Configuration of Parallel Port.

Data port is assigned for channel section. Control port also participated in conversion process, data selection as well as bitwise manipulation. Status port is only used for input operation. Nibble mode is the preferred way of reading 8 bits of data from A/D converter without placing the port in bi-directional mode and using the data lines. Nibble mode uses a Quad 2 line to 1 line multiplexer DM74157 to read a nibble of data at a time. Then it "switches" to the other nibble and reads its. Software can then be used to construct the two nibbles into a byte.

**Software**

The software has been built on Turbo C++. It has been covered channel selection, data logging, channel scaling, performing off-line analysis (calibration), alarming and event management. This data logging software has the following important features:

i. One of the primary functions of this data logging software is to handle the storage of the data. It can automatically store the data in an efficient manner, and provide a method for backing up and archiving data.
   The data of resultant of desired parameter has been written in a excel file (*.xls) and read from that file for plotting.
ii. The processed data after calibration has been graphically displayed in the output screen.

The operational steps of data logging are as given below:

i. Select the desired analog channel by sending a binary value of 8 bit to the data pins of the parallel port along with setting the control pin (C0) high or low.
ii. Have a write operation to the A/D converter.
iii. Wait for some time until the conversion process is completed.
iv. Have a read operation to the ADC and get the converted data from the data bus.
v. Acquire lower 4 bit of converted data from digital MUX by setting selector pin LOW and upper 4 bit data by setting HIGH.
vi. Combine the two 4 bit data together to make a byte using bitwise manipulation.
vii. Perform channel scaling operation to get the input voltage from corresponding binary raw data.
viii. Covert the input signal into desired physical quantity from the mathematical function as obtained by calibration previously in offline analysis.
ix. Storage of data along with current time and date for display and report generation.
x. Display acquired data either graphically or tabular format.

VOL.1, NO.1, JUNE 2006 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

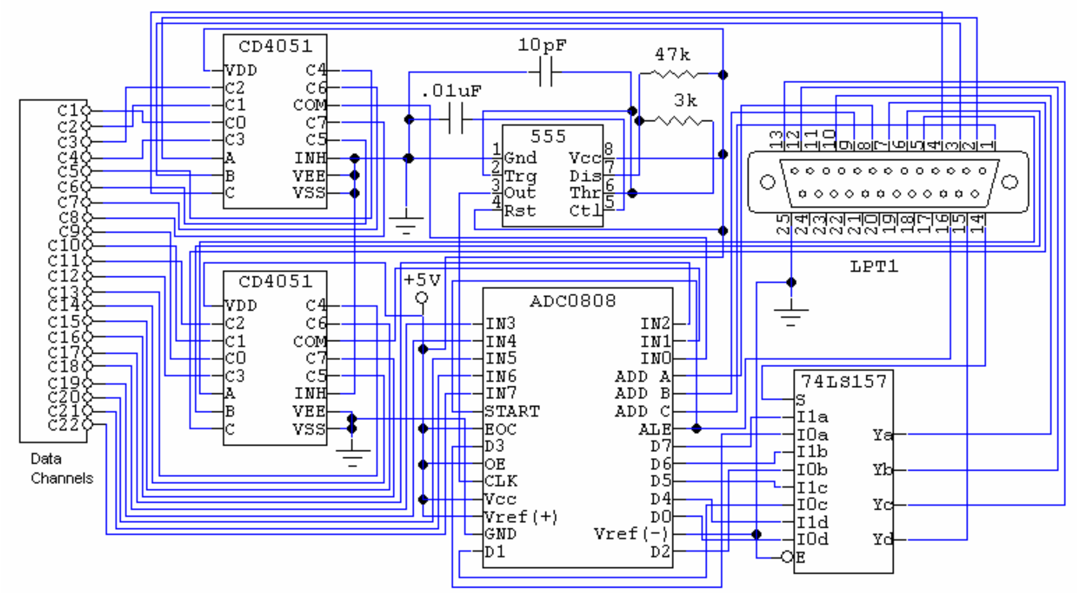



**Figure-7.** Complete Circuit Diagram of Data Logger.

## METHODOLOGY

First, the data logger is connected to a personal computer via parallel port. Then the software is used to select logging parameters (number of channels, sampling intervals, start time, etc.) and initiate the logger. The logger is then connected to selected input channels one by one and sequentially records each measurement after proper conversion of analog signals into digital format and stores it in memory along with the time and date.

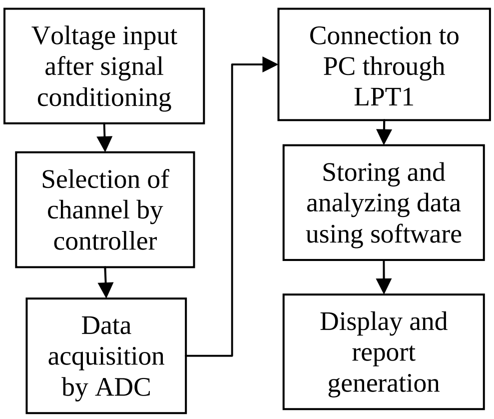



**Figure-8.** Flow Diagram of Data Logging Process.

At the same time, the software is used again to readout the data and examines the measurements as a graph, showing the profile over time. The tabular data can be viewed as well, or exported to a spreadsheet for further manipulation. The typical flow diagram for the data logging process is summarized in Figure-8.

Using eight data pins and one inverting control pin of parallel port, the channel section procedure is listed in Table-1.

**Table-1.** Channel Selection Procedure for Data Logger.

| Selected Channel | Data Port of LPT1 | | | | | | | | $\overline{C0}$ |
|---|---|---|---|---|---|---|---|---|---|
| | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | |
| 1 | 0 | 0 | 0 | X | X | X | 0 | 0 | 1 |
| 2 | 1 | 0 | 0 | X | X | X | 0 | 0 | 1 |
| 3 | 0 | 1 | 0 | X | X | X | 0 | 0 | 1 |
| 4 | 1 | 1 | 0 | X | X | X | 0 | 0 | 1 |
| 5 | 0 | 0 | 1 | X | X | X | 0 | 0 | 1 |
| 6 | 1 | 0 | 1 | X | X | X | 0 | 0 | 1 |
| 7 | 0 | 1 | 1 | X | X | X | 0 | 0 | 1 |
| 8 | 1 | 1 | 1 | X | X | X | 0 | 0 | 1 |
| 9 | X | X | X | 0 | 0 | 0 | 1 | 0 | 1 |
| 10 | X | X | X | 1 | 0 | 0 | 1 | 0 | 1 |
| 11 | X | X | X | 0 | 1 | 0 | 1 | 0 | 1 |
| 12 | X | X | X | 1 | 1 | 0 | 1 | 0 | 1 |
| 13 | X | X | X | 0 | 0 | 1 | 1 | 0 | 1 |
| 14 | X | X | X | 1 | 0 | 1 | 1 | 0 | 1 |
| 15 | X | X | X | 0 | 1 | 1 | 1 | 0 | 1 |
| 16 | X | X | X | 1 | 1 | 1 | 1 | 0 | 1 |
| 17 | X | X | X | X | X | X | 0 | 1 | 1 |
| 18 | X | X | X | X | X | X | 1 | 1 | 1 |
| 19 | X | X | X | X | X | X | 0 | 0 | 0 |
| 20 | X | X | X | X | X | X | 1 | 0 | 0 |
| 21 | X | X | X | X | X | X | 0 | 1 | 0 |
| 22 | X | X | X | X | X | X | 1 | 1 | 0 |

X = Don't Care

VOL.1, NO.1, JUNE 2006 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

## DATA COLLECTION AND ANALYSIS

Without using any offline analysis and selecting first three channels of data logger, the acquired voltage of different channels with time is shown in Table-2.

**Table-2.** Measured sample data (voltage) for first three channels in respect to time.

| Time (sec) | Output (Voltage) | | |
|---|---|---|---|
| | Channel -1 | Channel -2 | Channel -3 |
| 0 | 2.5 | 0.2 | 1.4 |
| 5 | 1.8 | 0.9 | 1.8 |
| 10 | 3.4 | 1.2 | 1.6 |
| 15 | 0.7 | 2.1 | 2.3 |
| 20 | 2.3 | 2.3 | 2.0 |
| 25 | 1.6 | 2.4 | 2.8 |
| 30 | 3.4 | 3.1 | 2.3 |
| 35 | 2.4 | 2.3 | 2.7 |
| 40 | 3.3 | 2.1 | 1.9 |

During offline analysis, measured voltage can be converted into physical quantity by using calibrated data as shown in Table-3 for channel 1.

**Table-3.** Wheatstone bridge output was calibrated to get displacement from the measured data.

| No. | Displacement (mm) | Output (Voltage) |
|---|---|---|
| | | Channel - 1 |
| 1 | 0.5 | 0.88 |
| 2 | 0.9 | 1.04 |
| 3 | 1.5 | 1.20 |
| 4 | 1.8 | 1.29 |
| 5 | 2.0 | 1.41 |
| 6 | 2.6 | 1.55 |
| 7 | 3.0 | 1.65 |
| 8 | 4.0 | 1.82 |
| 9 | 5.0 | 2.04 |
| 10 | 6.0 | 2.22 |
| 11 | 7.5 | 2.33 |

After calibration of Wheatstone bridge output by measuring displacement versus voltage which is best fitted with a $4^{th}$ order polynomial in MS Excel. The equation we found was:

$$y = 0.2312x^4 + 0.4969x^3 - 3.775x^2 + 8.3221x - 4.418$$

The R-squared value of this best fitted equation was 0.992.

## RESULTS

The sample output voltage of three channels with time is shown in Figure-9. Getting calibrated equation for each channel can make proper display of measured physical quantity with time. Hence not only for displaying plotted data with time for different channels but storing them as Excel file in tabular format for further analysis as well as for report generation.

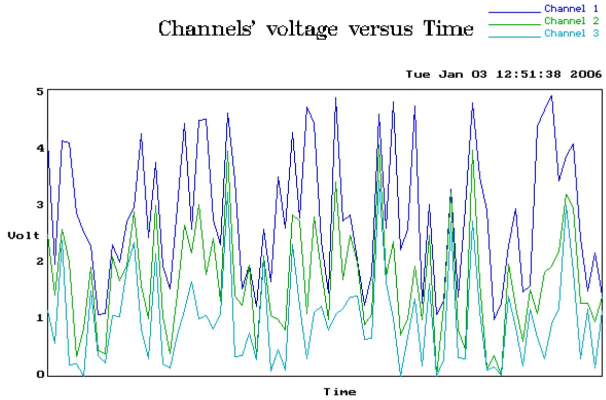



**Figure-9.** Logging and Displaying Data without calibration for first three channels.

## DISCUSSION

A main feature of our data logger is the conversion process, which emulated many of the functions found on a commercially available data logger. Here is a list of its main features:

i. Capable of sampling from 1 to 10 kHz.
ii. Capable of sampling 1 to 22 channels.
iii. Capable of logging data from 1 to 22 channels.
iv. Capable of saving logged data to disk.
v. Capable of measuring accurate voltages and times.
vi. Fully color coded graphics output.
vii. Optional grid management.
viii. Labeled time and measured data (voltage) markers.
ix. XY mode for plotting data with time.

Designing these data logger cannot meet sufficient additional operations as available in commercial data loggers but can satisfy the necessary requirements to accomplish the basic data logging operation specially multiple channel selection with very low cost.

## CONCLUSION

Using data logging, scientists and engineers can evaluate a variety of phenomena, from weather patterns to factory performance. PC-based data logging systems provide the most flexibility, customization, and integration. To define a data logging system, we must evaluate all the requirements for acquisition, analysis, logging, display, and report generation. Based on these requirements, we can customize data logging software and hardware to meet any needs.

## REFERENCES

Hall, Douglas V. 1999. Microprocessors and Interfacing: Programming and Hardware, Tata Mc-Graw-Hill Publishing Company Ltd., New Delhi.

Axelson, Jan 1996. Parallel Port Complete, Lakeview Research, Madison, WI 53704.

Braga, Newton C. 2002. Robotics, Mechatronics, and Artificial Intelligence: Experimental Circuit Blocks for Designers, Newnes, New Delhi.

Gadre, Dhananjay V. 1998. Programming the Parallel Port: Interfacing the PC for Data Acquisition and Process Control, CMP Books, Lawrence, KS 66046.

**APPENDIX**

Program Source Code (Turbo C++)

```
int a, b, c; float v;
outportb(0x0378, 0);// Initialization of data port
outportb(0x037a, 11);// Initialization of control port
outportb(0x0378, 0);
outportb(0x037a, 1);// Selection of first channel
outportb(0x037a, 5);// Start data conversion
outportb(0x037a, 1);// End data conversion
outportb(0x037a, 3);// Select lower 4 bits for acquisition
a = inportb(0x0379);// Acquisition of lower 4 bits
outportb(0x037a, 1);// Select upper 4 bits for acquisition
b = inportb(0x0379);// Acquisition of upper 4 bits
/* Bitwise manipulation */
a = a >> 3;
a = a & 0x0F;
b = b << 1;
b = b & 0xF0;
c = b | a;
/* Conversion of raw data into desired voltage signal */
v = c * (5.0 / 255);
```