TEST AUTOMATION OF AIR DRYERS USING LabVIEW6i

PROJECT REPORT

Estd-1984

P-1393

SUBMITTED BY

BALAJI.R

PRATHAP NIRMAL.N

MANIKANDAN.J

RAMESH.B

GUIDED BY

Ms.A.Vasuki M.E.,
Asst. Professor
ECE Department

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

2002-2003

BACHELOR OF ENGINEERING IN ELECTRONICS AND COMMUNICATION ENGINEERING

OF THE BHARATHIAR UNIVERSITY, COIMBATORE.

Department of Electronics and Communication Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641 006.


Certificate

Estd-1984

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641 006

Department of Electronics and Communication Engineering

Certificate

This is to certify that this project entitled

**TEST AUTOMATION OF AIR DRYERS
USING LabVIEW6i**

has been submitted by

BALAJI.R 9927DO121
PRADHAP NIRMAL.N 9927DO146
MANIKANDAN.J 9927D0715
RAMESH.B 9927DO716

In partial fulfillment of the requirements for the award of degree of Bachelor of Engineering in the Electronics and Communication Branch of Bharathiar University, Coimbatore-641 046 during the academic year 2002-2003.

17.03.03

(Guide) **(Head of the Department)**

Certified that the Candidate was examined by us in the Project Work Viva-Voce Examination held on 18-03-2003

18/3/03

(Internal Examiner) **(External Examiner)**

RIDENT PNEUMATICS PVT. LTD.
32, K.N.G. Pudur Road,
mayampalayam P.O.,
mbatore - 641 108. India
+91 422 400492
: +91 422 401376
nail : sales@tridentpneumatics.com
bsite www.tridentpneumatics.com
w Phone No.s :
: +91 - 422 - 2400492, 2401373
w Fax No. :
: +91 - 422 - 2401376

TRIDENT
LEADING BY TECHNOLOGY

TO WHOMSOEVER IT MAY CONCERN

This is to certify that the following final year B.E Electronics and Communication Engineering students of Kumaraguru College of Technology, Coimbatore have undergone and successfully completed a project in our organization.

1. **R.BALAJI**
2. **N.PRATHAP NIRMAL**
3. **J.MANIKANDAN**
4. **B.RAMESH**

Project Title	**Test Automation of Air Dryers using LabVIEW6i**
Project period	**July 2002 to March 2003**
Department	**Research and Development (Electronics)**

During this period their attendance conduct and behaviour were found to be good.

Trident Pneumatics Pvt Ltd

TRIDENT PNEUMATICS PVT. LTD.
5/632, K.N.G. Pudur Road (Off Thadagam Road)
Somaiyampalayam P.O.
Coimbatore - 641 108. India
Phone : +91-422-400492, 401373.
Fax : +91-422-[illegible]
E-mail : [illegible]

R.Sivakumar
R&D Engineer
Coimbatore
15/3/03

DEDICATED TO OUR BELOVED PARENTS...

ACKNOWLEDGEMENT

Acknowledgement

"Success is never one man show". Our deepest gratitude to all those who have helped us in this project and made it a memorable and successful one.

First and foremost we wish to express our profound thanks to our respected Principal Dr.K.K.Padmanabhan, B.Sc., (Engg), M.Tech., Ph.D., who has given consent to this project.

We are forever indebted to our beloved Head Of the Department, Prof. Muthuraman Ramasamy, M.E., MIEEE (USA), MIE, MISTE, C.Eng.(I), MIBMESI., for his valuable suggestions and support.

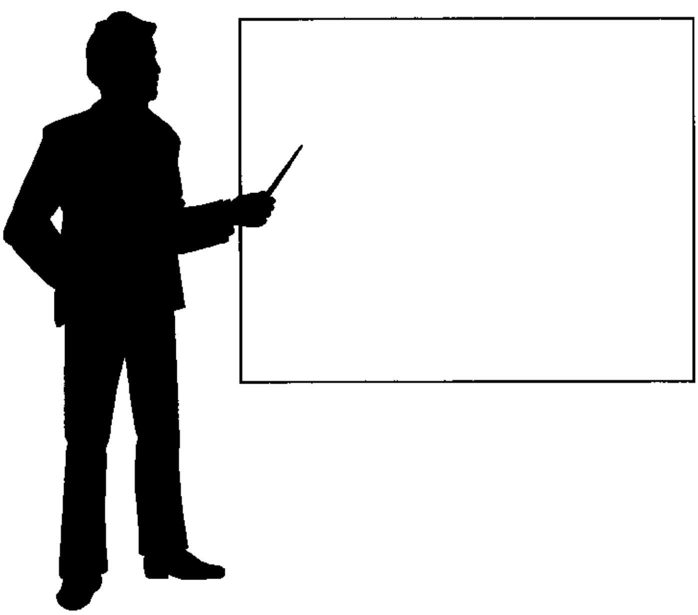
It has been a privilege to work under the guidance of our discerning guide Assistant Professor Ms.A.Vasuki M.E., who is a constant source of inspiration. We are highly indebted to her for her constant encouragement, undying patience, sincere evaluation and constructive criticisms.

We thank our Class Advisor Ms. Amrita Gowri for her good support and good will at various instances. We are very grateful to all the staff members who have extended their support during all times. Thanks to all the Lab Technicians of ECE and Mechanical Engineering Department for their full co-operation.

We are grateful to Mr.K.S.Natarajan, Managing Director, Trident Pneumatics Pvt. Ltd for providing an opportunity to pursue our project at their reputed concern.

We are grateful to our external guide Mr.R.Sivakumar, Team Leader Engineer, Trident Pneumatics Pvt. Ltd, for his valuable suggestions throughout the course of the project. A special thanks to Mr.K.Kamalraj, Design Engineer, Trident Pneumatics Pvt. Ltd for spending his invaluable time for testing our project.

Last but not least, we thank our beloved friends for their energetic support and motivation for us.



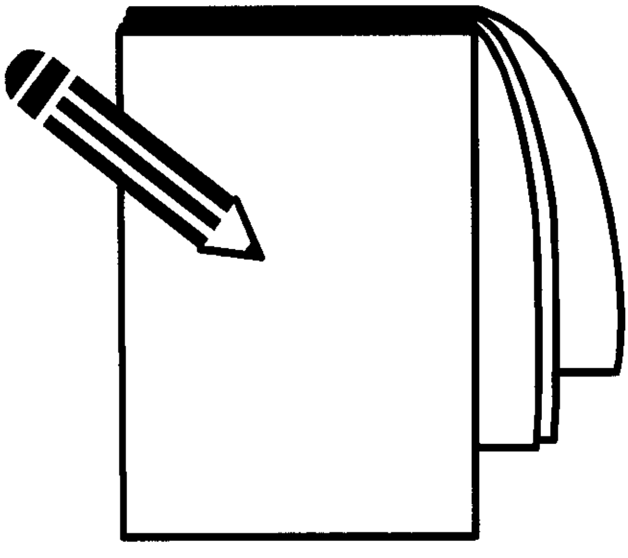
synopsis

SYNOPSIS

Automated testing system for Air Dryers is designed to test the performance of Air Dryers and plot the various characteristics of the same. Air Dryer is industrial equipment used to produce clean, dry and dust free air as output. The air used by pneumatic industrial equipments which may contain moisture, dust particles is given as input to the Air Dryer. The physical parameters required to plot the characteristics on a Personal Computer (PC) are acquired using transducers, Data Acquisition Card and its Accessories. The software package used is LabVIEW 6i (Laboratory Virtual Instrumentation Engineering Workbench). Automated Testing System reduces testing time since external data loggers are eliminated. The system also reduces the man-hours in industries used to test the performance and is cost effective. The characteristics plotted using this system is highly accurate. Enhancements such as remote testing, alarm etc., can be added in future.

CONTENTS

Contents



Introduction

CHAPTER 1
INTRODUCTION

An automated testing system is one which is designed to provide and ensure the proper working condition of a product or equipment. The testing system may be completely mechanical or electrical or electronic in nature or a combination of two or more of the above mentioned types. This depends on various factors. This project is essentially an integrated system which comprises mechanical as well as electronic types.

The system here is designed to test the performance of an air dryer which is an industrial device designed to provide clean, dry air. The parameters taken into account are change in pressure, air flow, dew point temperature and air loss. The data acquired from the dryer are input pressure, output pressure and dew point temperature. These data are converted into analogous electrical quantities by means of transducers placed at specific points in the dryer. The electrical quantities are given as inputs to the DAQ (Data Acquisition) add-on card through its accessory. The data acquisition card has an in-built analog to digital converter which converts the analog data into digital form. The digital data is acquired by the personal computer with the help of ports in the add-on slots. The other parameters i.e. airflow, air loss and change in pressure are calculated using appropriate formulas from the acquired data. Using the acquired and calculated parameters, the following characteristics of air dryers are plotted.

1. Estimation of Air loss.
2. Air flow versus Change in Pressure.

The software package used for acquiring, processing and storing the above parameters is LabVIEW6i (Laboratory Virtual Instrumentation Engineering Workbench). Using LabVIEW6i, a virtual instrument is created with three graphs to view the characteristic plots. The stored data is used to plot the various testing values to analyze the performance of the air dryer.

The Virtual Instrument is user friendly and the controls and indicators can be modified at any time. The system designed is highly adaptive for any changes in any of the parameters. The system is very fast and highly accurate providing provisions for future improvement in it.

HARDWARE DESCRIPTION

CHAPTER 2

Hardware description

2.1 Introduction

The overall drying process of the compressed moist air involves absorption of the moisture by the desiccant inside the Dryer, and depressurisation for regeneration of the Dryer.

2.2 Air Dryer description

The key element in the overall process being the Dryer with the active drying element in it ,and its efficiency to remove the moisture completely from the moist entrant.

2.2.1 Construction

The major component of Air Dryer is the pair of Towers with the desiccant Activated Alumina. The Pre-filter through which the moisture laden input air enters initially is the first block in the setup. The output of this is lead to a three-way valve, two of which leads to the Towers and the third is the exhaust valve. The valves leading to Tower leads the moist air to the drying process (i.e. Tower). Only one Tower works at a time. The output of Tower I is given to a non-return valve from where the output Dry-air is taken. When this Tower is saturated, the Tower II functions, the switchover to this being handled by the three-way valve mentioned above. The non-return valve is also connected to a Purge Economizer in which, by setting a switch in the controller at a percentage of maximum flow, the controller will close Purge valve when sufficient purge air (or) the part of the dry air output used for regeneration, has been used. This reduces the Purge Loss according to air usage. There is also an extensive mimic display on the front, displaying the current status and functioning of the overall setup.

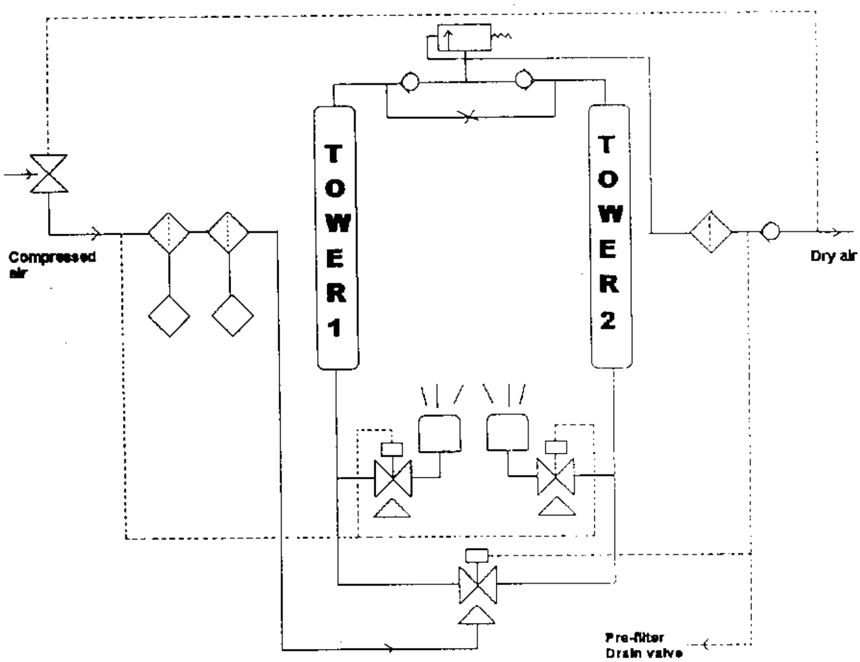


Fig 2.1 Schematic diagram of Air Dryer

2.2.2 Principle of Operation

2.2.2.1Pre-filtration The moisture laden compressed air passes through the pre filter, here moisture load is reduced through coalescence. Condensation is removed through the drain valve ADV1. At the oil filter, oil vapours are removed completely, small level of water condensation is purged through the valve ADV2. Compressed air is then passed through a three-way valve into the Tower.

2.2.2.2 Drying The Towers are filled with Activated Alumina as desiccant. When air passes through Tower I which consists of DRY desiccant, it gets completely dried, and passes through check valve and after-filter. At the after filter Desiccant fines are removed. Therefore dry compressed air passes out at the outlet.

2.2.2.3 Regeneration Regeneration takes place in 2 stages (a) De-pressurization (b) by passing Dry air. Tower II consists of fully moist desiccant. This is suddenly De-pressurized by opening the purge valve. Water molecules seep out of the desiccant and appear on the surface. "Super Dry" purge air passes through the regeneration "Nozzle" and the desiccant bed thereby completely carrying away the water molecules. Tower II gets regenerated and is ready for the next drying cycle.

2.2.3 Direct Parameters Involved

1. Inlet Pressure-Pressure at the input of Air Dryer.
2. Outlet Pressure-Pressure at the output of Air Dryer.
3. Working Temperature.
4. Valve co-efficient.

2.2.4 Applications

The Air Dryer finds a variety of applications where clean, dry air is needed. This includes Thermal Plants, Fertilizer Plants, Pneumatic Sector, and Automobile Industries.

The Dryer construction stands a complete systematic and compatible method of drying the moist air, involving a variety of applications and the whole system implementation at an economically optimal way.

2.3 TRANSDUCERS

2.3.1 Introduction

Transducers are devices which convert the physical parameters into corresponding electrical signals. The output of the transducer may be an equivalent current or voltage. Pressure transducers convert the pressure of the air applied at the input into corresponding voltage or current. The range of measurement varies according to the type of transducers used and the principle of working. The type of transducer used here is SDE-16-10V/20mA, manufactured by Festo pneumatic.

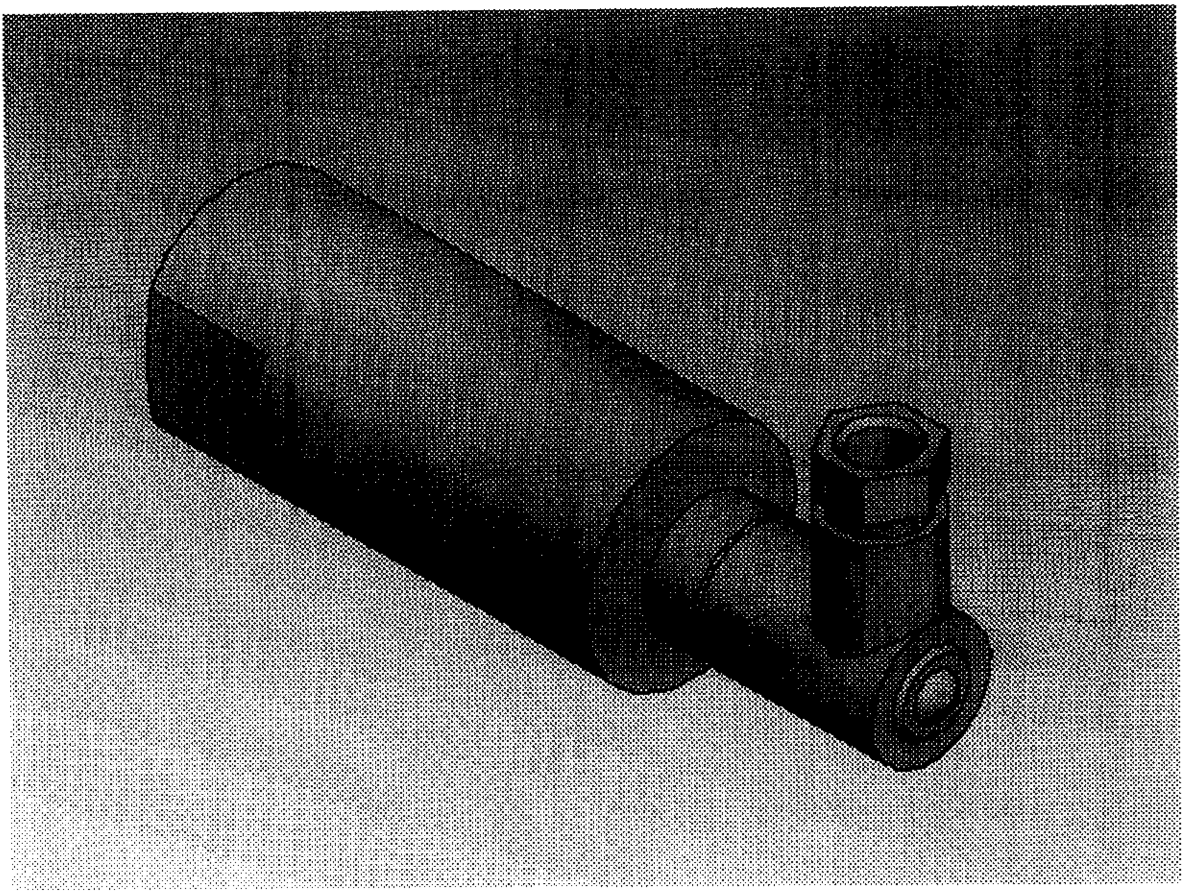
2.3.2 Specifications of SDE-16-10V/20mA

- Medium – compressed air, filtered (lubricated or unlubricated)
- Design – piezo-resistive element as relative pressure transducer
- Mounting – 32mm diameter, fitting recess or by G1/4 port thread
- Pressure measurement range – 0 to 16 bar
- Supply voltage – 12 to 30 V DC (residual ripple 10%)

- Output voltage – 0 to 10V
- Output current – 0 to 20mA
- Measurement frequency range – 100Hz
- Temperature range – 0 to +85°C
- Material – anodized aluminium housing
- Weight – 0.120 kg

2.3.3 Construction

The festo pressure sensor is a piezo-resistive pressure recorder with integral amplifier and temperature compensation, suitable for many applications. It is mounted inside compact and robust aluminium housing. It has G1/4 port for the physical input. The piezo-resistive element acts as the sensing primer. A silicon layer is provided to transmit the measured pressure to piezo-resistive element. Separate wires are provided for the supply voltage and the sensor outputs. The pressure sensor has a built in protective circuit. It gives protection against short circuit, incorrect polarity and overload up to 50 V DC.



SDE-16-10V/20mA

Fig 2.2 Festo Pressure Transducer

2.3.4 Mounting

The sensor may be connected to the pressurized system rigidly, through the G1/4" internal thread and a screwed fitting, or attached to housing elements, etc. by means of the appropriate mountings. In principle, mounting is possible in any position. However it should be noted that no condensate should be collected at the sensor's point of connection. For linking the sensor to the pressurized system, an internally threaded G1/4" port has been provided, into which an appropriate fitting (e.g. quick connector CK-1/4-PK-4) may be screwed together with a sealing ring. The required tightening torque is 50Nm. Electrical connection is accomplished by using angle plug, type SIE-WD-TR, or sensor plug SIE-GD. The connecting cable must be screened. The screening (degree of screen braiding at least 50%) must not be connected to earth. The housing must be earthed.

2.3.5 Working

The pressure of the air to be measured is supplied at the input port. It is transmitted on to the piezo-resistive element via silicon layer. When the pressure of the input air changes, the resistance of the piezo-resistive element changes correspondingly. This change in resistance of the piezo-resistive element is reflected at the output voltage and current. The output of the piezo-resistive element is supplied as input to the integration amplifier. The output is calibrated so that the mutual interchangeability of sensors is assured. The sensor has a case grounding to drain the leakage currents.

2.4 Accessory

2.4.1 BNC-2100 Series Connector Blocks

The BNC-2100 Series are shielded connector blocks with signal-labeled BNC connectors for easy connectivity of analog input, analog output, digital I/O and counter/timer signals to multifunction DAQ device, including analog input devices. The BNC-2100 series work with all E Series devices. The BNC-2100 also provides a function generator, quadrature encoder, temperature reference, thermocouple connector, and LED so that the functionality of the hardware can be tested. The BNC-2115 has 24 BNC inputs for connecting to the extended I/O channels of the 100-pin E Series DAQ devices.

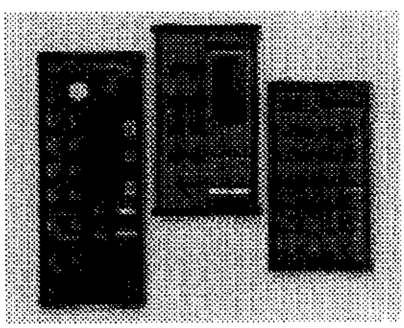


Fig 2.3 BNC connector blocks

2.4.2Ribbon I/O Cables

R6850 Ribbon Cable Kit is used to connect the BNC connector block and DAQ card. This cable kit combines a 68F-50M cable adapter and a standard 50-pin cable with female connectors on both ends. The cable kit is designed to adapt an E Series, S Series, or PCI-6013/6014 product to a third-party or custom 50-pin accessory. The length of the cable is one meter. These types of standard cables are used for lossless transmission of signals.

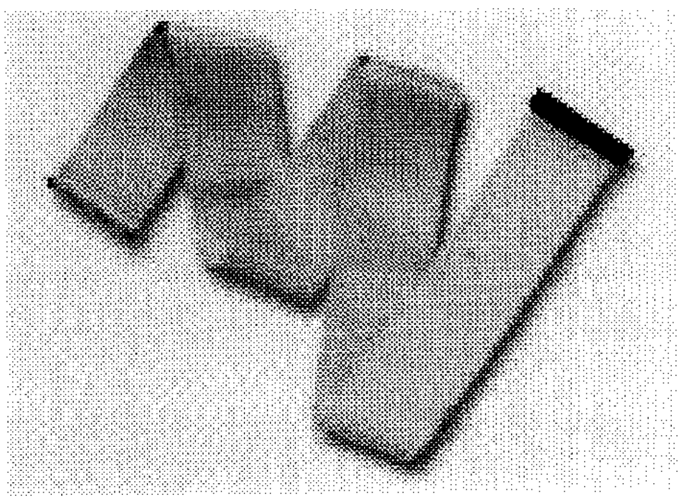


Fig 2.4 Ribbon cable

2.5 Data acquisition hardware

2.5.1 Introduction

DAQ card is connected in the add-on slot of the CPU. It acquires analog data from external world, converts it into digital data.

2.5.2 Virtual Instrumentation

Measurements are taken with the help of instruments. Instrumentation helps science and technology progress. Scientists and engineers around the world use instruments to observe, control, and understand the physical universe. The quality of life depends on the future of instrumentation—from basic research in life sciences and medicine to design, test and manufacturing of electronics, to machine and process control in countless industries.

Virtual instrumentation is defined as combining hardware and software with industry-standard computer technologies to create user-defined instrumentation solutions. The driver software is the programming interface to the hardware and is consistent across a wide range of platforms. Application software such as LabVIEW6i, LabWindows/CVI, Component Works, and Measure deliver sophisticated display and analysis capabilities required for virtual instrumentation. Virtual instrumentation can be used to create a customized system for test, measurement, and industrial automation by combining different hardware and software components. If the system changes, virtual instrument components can be reused without purchasing additional hardware or software. Different hardware and software components can make up a virtual instrumentation system. A wide variety of hardware components can be used to monitor or control a process or test a device.

2.5.2 Data acquisition devices

Measurement devices, such as general-purpose data acquisition (DAQ) devices and special-purpose instruments, are concerned with the acquisition, analysis, and presentation of measurements and other data acquired. Acquisition is the means by which physical signals, such as voltage, current, pressure, and temperature, are converted into digital format and brought into the computer. Popular methods for acquiring data include plug-in DAQ and Instrument Devices, GPIB instruments, VXI instruments, and RS-

232 instruments. Data analysis transforms raw data into meaningful information. This can involve such things as curve fitting, statistical analysis, frequency response, or other numerical operations. Data presentation is the means for communicating with the system in an intuitive, meaningful format. A wide variety of hardware components can be used to monitor or control a process or test a device.

2.5.3 Computers and DAQ Devices

Before a computer-based system can measure a physical signal, a sensor or transducer must convert the physical signal into an electrical one, such as voltage or current. The plug-in DAQ device is often considered to be the entire DAQ system, although it is actually only one system component. Unlike most stand-alone instruments, we cannot always directly connect signals to a plug-in DAQ device. In these cases, you must use accessories to condition the signals before the plug-in DAQ device converts them to digital information. The software controls the DAQ system by acquiring the raw data, analyzing the data, and presenting the results. Figure 2-1 shows two options for a DAQ system. In Option A, the plug-in DAQ

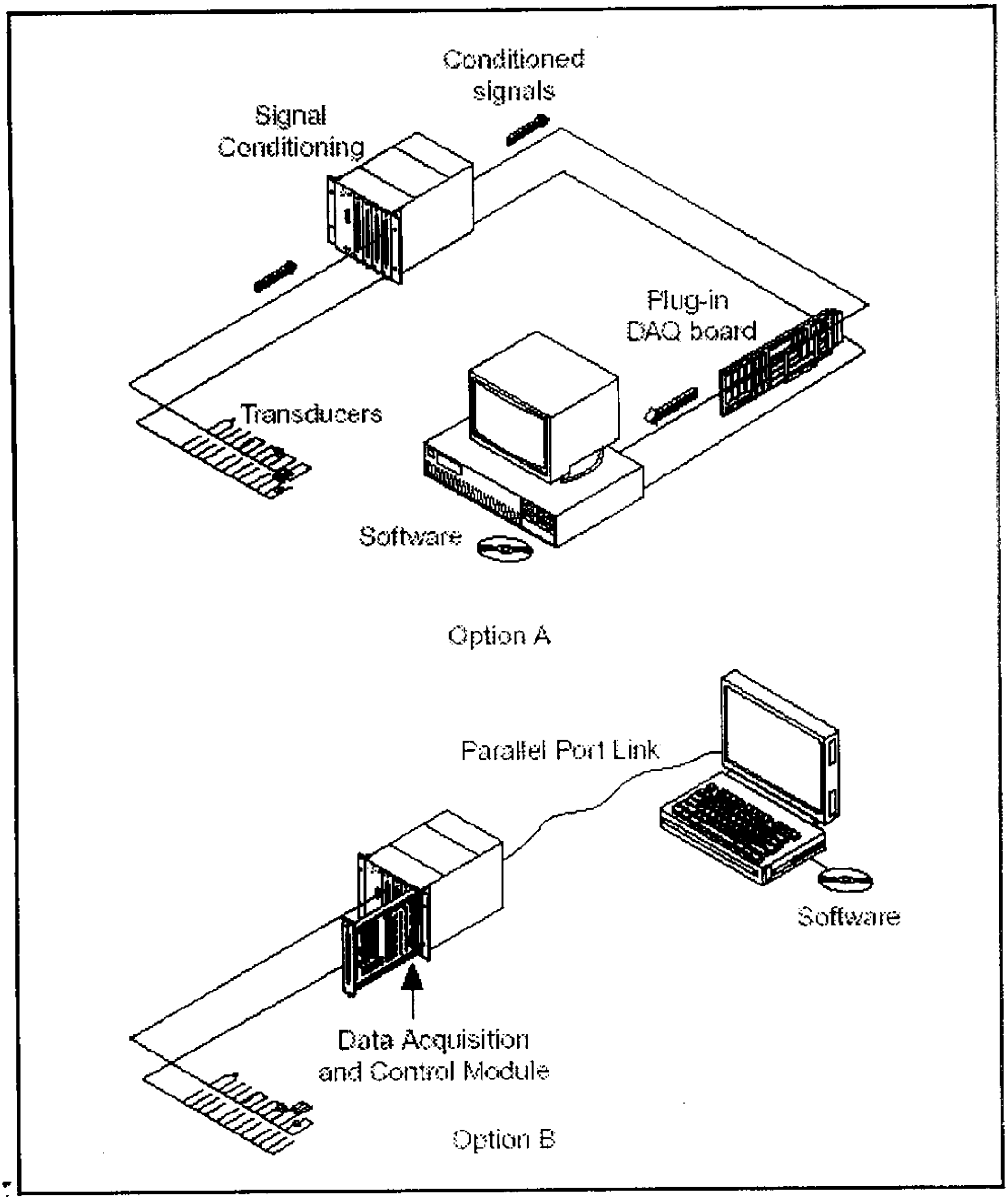


Fig 2.5 DAQ System Components

device resides in the computer. In Option B, the DAQ device is external. With an external board, you can build DAQ systems using computers without available plug-in slots, such as some laptops. The computer and DAQ module communicate through various buses such as the parallel port, serial port, and Ethernet. These systems are practical for remote data acquisition and control applications.

2.5.4 DAQ Devices Vs Special-Purpose Instruments

The fundamental task of all measurement systems is the measurement and/or generation of real-world physical signals. The primary difference between the various hardware options is the method of communication between the measuring hardware and the computer. General purpose DAQ devices are devices that connect to the computer allowing the user to retrieve digitized data values. These devices typically connect directly to the computer's internal bus through a plug-in slot. Some DAQ devices are external and connect to the computer via Serial, GPIB, or Ethernet ports. The primary distinction of a test system that utilizes general purpose DAQ devices is where measurements are performed. With DAQ devices, the hardware converts the incoming signal into a digital signal that is sent to the computer. The DAQ device does not compute or calculate the final measurement. That task is left to the software that resides in the computer. The same device can perform a multitude of measurements by simply changing the software application that is reading the data. So, in addition to controlling, measuring, and displaying the data, the user application for a computer-based DAQ system also plays the role of the firmware—the built-in software required to process the data and calculate the result—that would exist inside a special purpose instrument. While this flexibility allows the user to have one hardware device for many types of tests, the user must spend more time developing the different applications for the different types of tests.

LabVIEW6i comes with many acquisition and analysis functions to make this easy. Instruments are like the general purpose DAQ device in that they digitize data. However, they have a special purpose or a specific type of measurement capability. The software, or firmware, required to process the data and calculate the result is usually built in and cannot be modified. For example, a Multimeter can not read data the way an oscilloscope can because the program that is inside the Multimeter is permanently stored and cannot be changed dynamically. Most instruments are external to the computer and can be operated alone, or they may be controlled and monitored through a connection to the computer. The instrument has a specific protocol that the computer must use in order to communicate with the instrument. The connection to the computer could be Ethernet, Serial, GPIB, or VXI. There are some instruments that can be installed into the computer like the general purpose DAQ devices. These devices are called computer-based instruments.

2.5.5 Programs and Instruments

Instrument drivers are a key factor in test development. An instrument driver is a collection of functions that implement the commands necessary to perform the instrument's operations. LabVIEW instrument drivers simplify instrument programming to high-level commands. Instrument drivers are not necessary to use the instrument. Instrument drivers create the instrument commands and communicate with the instrument over the serial, GPIB, or VXI bus. In addition, instrument drivers receive, parse, and scale the response strings from instruments into scaled data that can be used in your test programs. Instrument drivers can help make test programs more maintainable in the long run because instrument drivers contain all of the I/O for an instrument within one library, separate from your other code. LabVIEW6i provides more than 700 instrument drivers from more than 50 vendors.

2.5.6 DAQ Card Details

The data acquisition card used here is PCI-6024E, which can sample eight analog inputs simultaneously. The details of the DAQ card are as follows.

2.5.6.1 Overview and Applications

National Instruments NI 6024E device use E Series technology to deliver high performance, reliable data acquisition capabilities. This device is used in a broad variety of applications including:

- Continuous high-speed data logging at up to 200 kiloSamples/sec.
- Externally timed and/or triggered data acquisition
- High-voltage and sensor measurements when used with NI signal conditioning
- High-channel-count system scalability with RTSI or PXI trigger bus

2.5.6.2 Features

NI 6024E device features a highly precise voltage reference used during self-calibration. A simple software call initiates self-calibration, which minimizes errors caused by temperature drift and time. This device feature the NI-PGIA, which is an instrumentation-class amplifier that guarantees settling times at all gains. Typical, commercial, off-the-shelf amplifier components might not meet the settling time requirements for high-gain measurement applications. Without the NI-PGIA, 12-bit devices with a 100X gain can have an effective resolution of only 10 bits. This device offer several methods for connecting your signals including a differential mode for eight Analog Input (AI) channels and maximum noise elimination, as well as referenced and non-referenced single ended mode for 16 AI channels.

NI 6024E device feature digital triggering, and two 24-bit, 20 MHz counter/timers. NI 6023E and NI 6024E devices feature eight digital I/O lines compatible with both 5 V TTL and CMOS while NI 6025E devices feature 32 digital I/O lines. NI 6024E and NI 6025E devices feature two 12-bit analog outputs.

SOFTWARE DESCRIPTION

CHAPTER 3

Software Description

Any system implemented using software needs the complete harnessing of the software utilities. The extent of complete usage of software makes the programming easy and precise. Then the debugging of the code also becomes a simple task.

3.1 Introduction to LabVIEW

LabVIEW6i (Laboratory Virtual Instrumentation Engineering Workbench) is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine program execution, LabVIEW6i uses dataflow programming, where the flow of data determines execution.

In LabVIEW6i, by using set of tools and objects, user interface can be built. The user interface is known as the **Front Panel**. Codes are added using graphical representations of functions to control the front panel objects. The **block diagram** contains this code. In some ways, the block diagram resembles a flowchart.

LabVIEW6i also provides numerous mechanisms for connecting to external code or software through DLLs, shared libraries ActiveX, and more. In addition, numerous add-on tools are available for a variety of application needs.

3.2 Working of LabVIEW

LabVIEW6i programs are called **Virtual Instruments**, or **VIs**, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. Every VI uses functions that manipulate input from the user interface or other sources and display that information or move it to other files or other computers.

A VI contains the following three components:

- **Front panel** – Serves as the user interface.
- **Block diagram** – Contains the graphical source code that defines the functionality of the VI.
- **Icon and connector pane** – Identifies the VI, so that a VI can be used in another VI. A VI within another VI is called a sub VI. A sub VI corresponds to a subroutine in text-based programming languages.

Several add-on software toolsets are available and are compatible for developing specialized applications. All the toolsets integrate seamlessly in LabVIEW6i.

LabVIEW6i is integrated fully for communication with hardware such as GPIB, VXI, PXI, RS 232, RS 485, and data acquisition control, vision, and motion control devices. LabVIEW also has built-in features for connecting the application to the Internet using the LabVIEW web server and software standards such as TCP/IP networking and ActiveX.

Using LabVIEW6i, 32-bit compiling applications can be created that gives the fast execution speeds needed for custom data acquisition, test, measurement, and control solutions. Stand-alone executables and shared libraries can also be created, like DLLs, because is a true 32-bit compiler.

LabVIEW6i contains comprehensive libraries for data collection, analysis, presentation, and storage. LabVIEW6i also includes traditional program development tools. Breakpoints, animate program execution, and single-step can be created through the program to make debugging and development easier.

3.3 Front Panel

The front panel is the user interface of the VI. The front panel can be built using controls and indicators, which are the interactive input and output terminals of the VI respectively. Controls are knobs, pushbuttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays. Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display data that the block diagram acquires or generates.

3.4 Block Diagram

After the front panel design, codes are added using graphical representations of functions to control the front panel objects. The block diagram contains this graphical source code. Front panel objects appear as terminals on the block diagram.

Additionally, the block diagram contains functions and structures from built-in LabVIEW VI libraries. Wires connect each of the nodes on the block diagram, including control and indicator terminals, functions and structures.

3.5 Palettes

LabVIEW palettes gives the options we need to create and edit the front panel and block diagram.

3.5.1 Tools Palette

The **Tools Palette** is available on the front panel and the block diagram. A tool is a special operating mode of the mouse cursor. By selecting a tool in the **Tools** palette, the cursor icon changes to the tool icon. Tools are used to operate and modify front panel and block diagram objects. If automatic tool selection is enabled and movement of cursor over objects on the front panel or block diagram, LabVIEW6i automatically selects the corresponding tool.

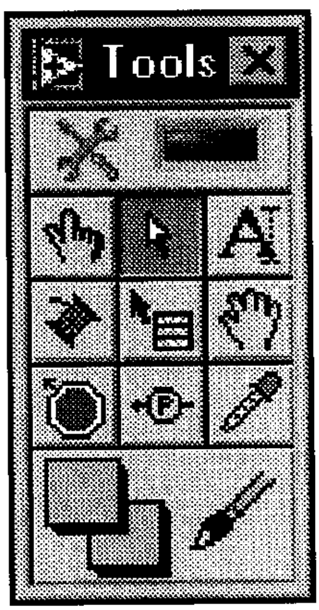


Fig. 3.1 Tools Palette

3.5.3 Functions Palette

The **Functions Palette** is available only on the block diagram. **Functions Palette** contains the VIs and functions used to build the block diagram. **Functions Palette** can also be placed anywhere on the screen.

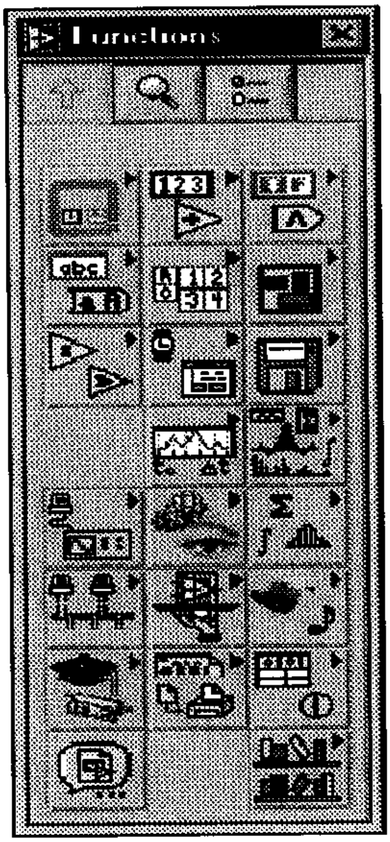


Fig. 3.2 Functions Palette

3.5.3 Controls Palette

The **Controls Palette** is available only on the front panel. The **Controls Palette** contains the controls and indicators used to create the front panel. **Controls Palette** can be placed anywhere on the screen.

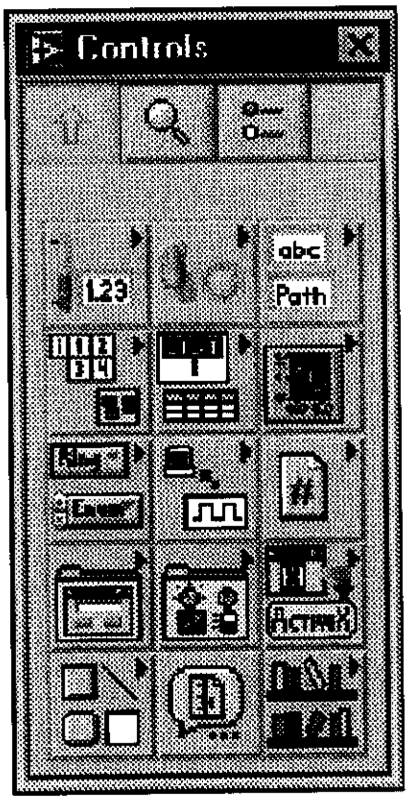


Fig 3.3 Controls Palette

3.6 Data Flow

LabVIEW follows a dataflow model for running VIs. A block diagram node executes when all its inputs are available. When a node completes execution, it supplies data to its output terminals and passes the output data to the next node in the dataflow path.

3.7 Analog Input Signals

Analog signals can be grouped into three categories: DC, time domain and frequency domain. The following fig.3.4 illustrates signals and the corresponding types of signal information.

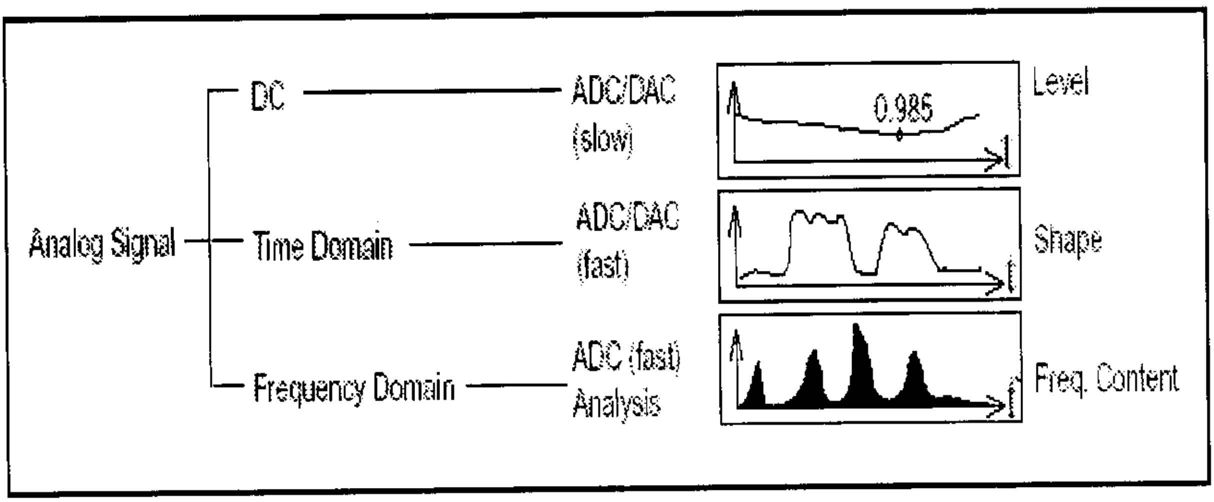


Fig. 3.4 Types of signal information

3.7.1 Grounded Signal Sources

Grounded signal sources have voltage signals that are referenced to a system ground, such as earth or a building ground. Devices that plug into a building ground through wall outlets, such as signal generators and power supplies, are the most common examples of grounded signal sources, as shown in Fig.3.5.

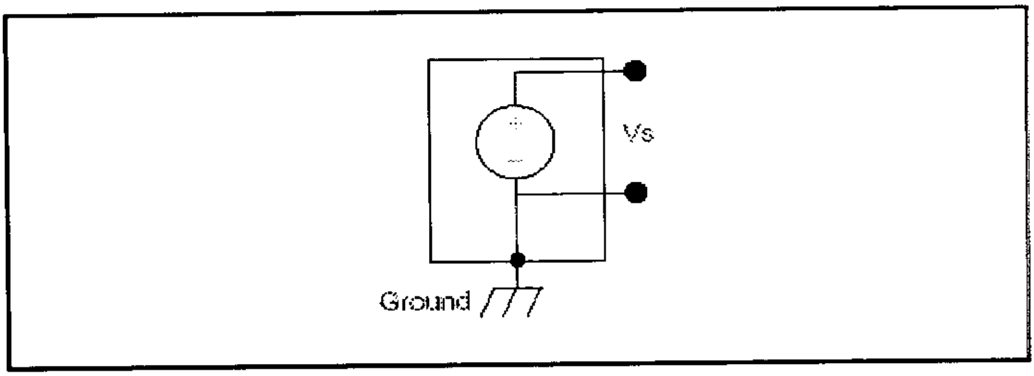


Fig 3.5 Grounded signal source

3.7.2 Floating Signal Sources

Floating signal sources contain a signal, such as a voltage, that is not connected to an absolute reference, such as earth or a building ground. Some common examples of floating signals are batteries, battery-powered sources, thermocouples, transformers, isolation amplifiers, and any instrument that explicitly floats its output signal.

Referring to Fig.3.6 neither terminal of the floating source is connected to the electrical outlet ground.

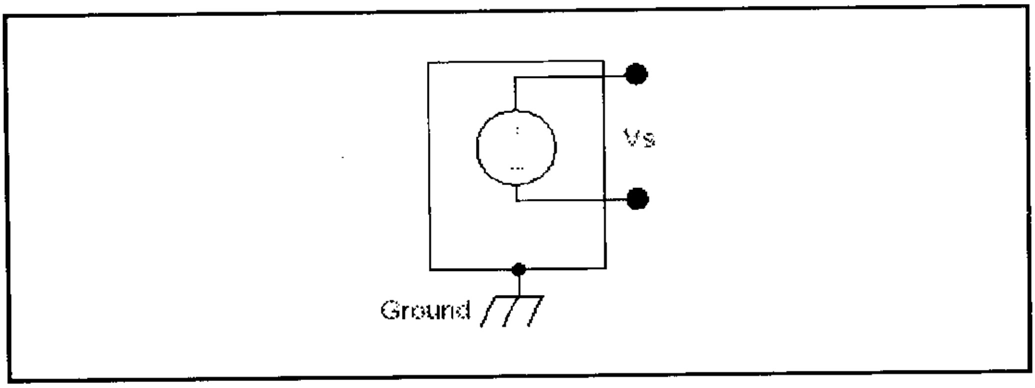


Fig 3.6 Floating signal source

3.7.3 Resolution

The number of bits used to represent an analog signal determines the resolution of the ADC. The resolution on a DAQ device can be compared to the marks on a ruler. The more marks, the more precise measurements can be obtained. Similarly, the higher the resolution, the higher the number of divisions into which the system can break down the ADC range, and therefore, the smaller the detectable change. A 3-bit ADC divides the range into 8 divisions. A binary or digital code between 000 and 111 represents each division.

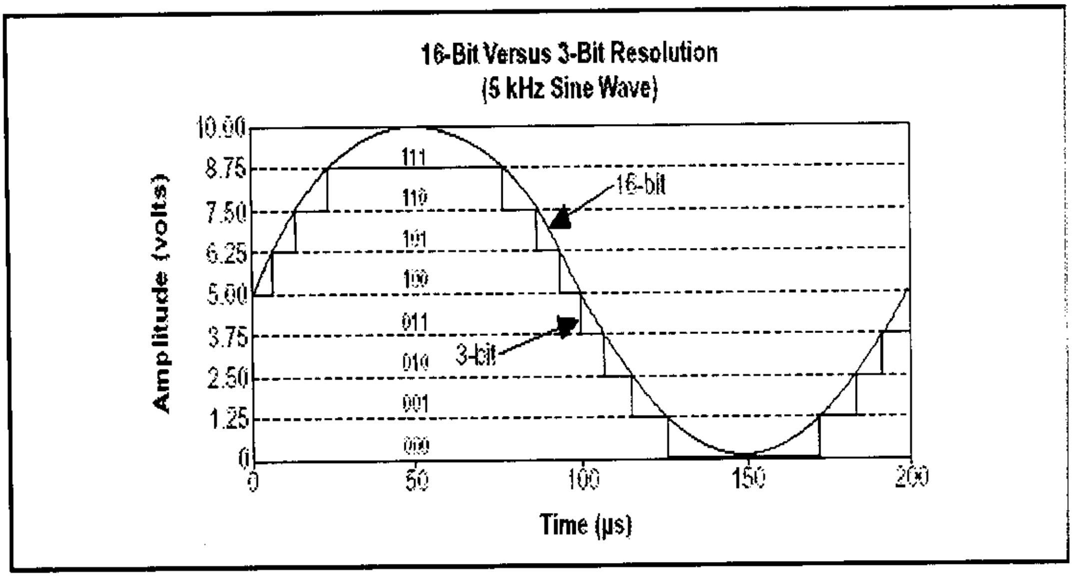
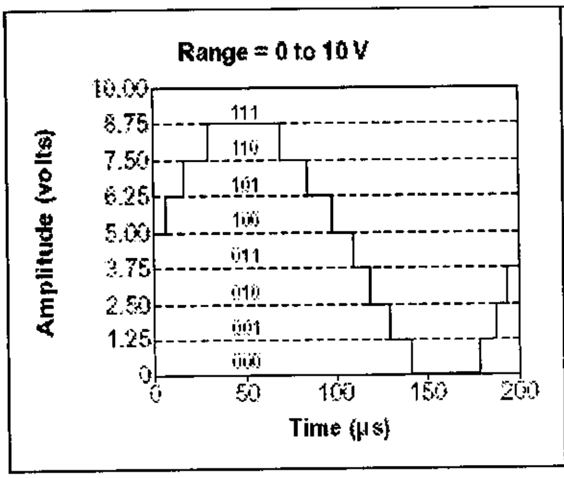


Fig 3.7 16-Bit Vs 3-bit resolution

The ADC translates each measurement of the analog signal to one of the digital divisions. The Fig.3.7 shows a sine wave digital image as obtained by a 3-bit ADC. Clearly, the digital signal does not represent the original signal adequately, because the converter has too few digital divisions to represent the varying voltages of the analog signal. By increasing the resolution to 16 bits, however, the ADC's number of divisions' increases from 8 to 65,536 (216). The ADC now can obtain an extremely accurate representation of the analog signal.

3.7.4 Device Range

Range refers to the minimum and maximum analog signal levels that the ADC can digitize. Many DAQ devices feature selectable ranges, so we can match the ADC range to that of the signal to take best advantage of the available resolution. For example, the 3-bit ADC, as shown in Fig. 3.8, has eight digital divisions in the range from 0 to 10 V. If we select a range of –10.00 to 10.00 V, as shown in Fig. 3.9, the same ADC now separates a 20 V range into eight divisions. The smallest detectable voltage increases from1.25 to 2.50 V, and we now have a much less accurate representation of the signal.



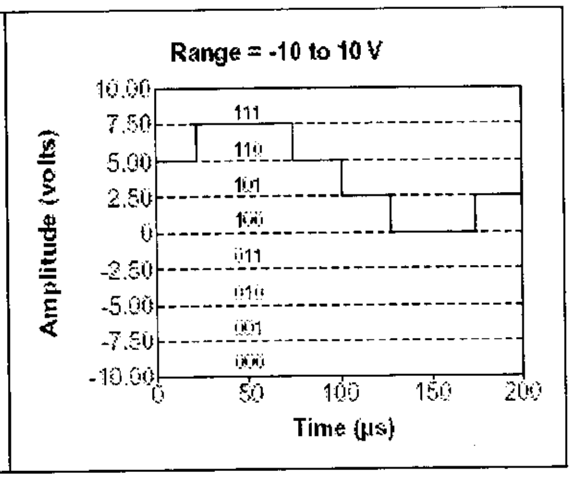


Fig 3.8

Fig 3.9

3.7.8 Signal Limit Settings

Limit settings are the maximum and minimum values of the signal we are measuring. A more precise limit setting allows the ADC to use more digital divisions to represent the signal. The following Fig.3.10 shows an example of this theory. Using a 3-bit ADC and a device range setting of 0.00 to 10.00 V, the Fig.3.10 shows the effects of a limit setting between 0 and 5 V and 0 and 10 V. With a limit setting of 0 to 10 V, the ADC uses only four of the eight divisions in the conversion. But using a limit setting of 0 to 5 V, the ADC now has access to all eight digital divisions. This makes the digital representation of the signal more accurate.

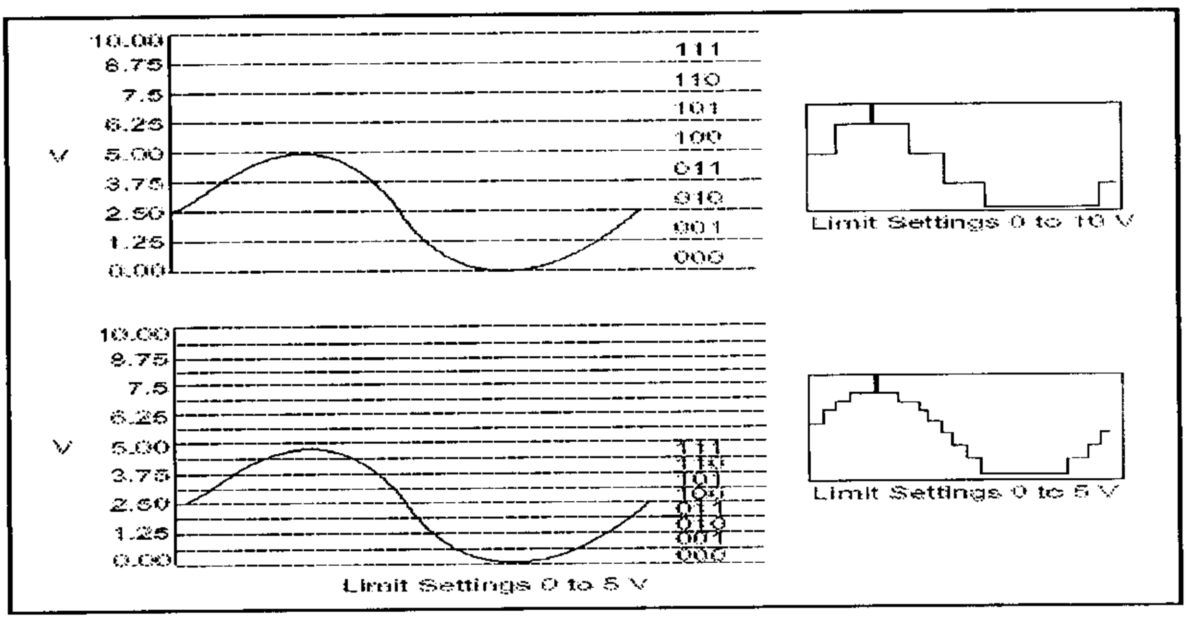


Fig 3.10 Signal limits

3.7.9 Considerations for Selecting Analog Input Settings

The resolution and device range of a DAQ device determine the smallest detectable change in the input signal. The smallest detectable change, called the *code width*, can be calculated using the following formula.

$$code\ width = \frac{device\ range}{2^{resolution}}$$

For example, a 12-bit DAQ device with a 0 to 10 V input range detects a 2.4 mV change, while the same device with a –10 to 10 V input range detects only a change of 4.8 mV.

$$\frac{device\ range}{2^{resolution}} = \frac{10}{2^{12}} = 2.4\ mV$$

$$\frac{device\ range}{2^{resolution}} = \frac{20}{2^{12}} = 4.8\ mV$$

A high-resolution A/D converter provides a smaller code width given the device voltage ranges shown above.

$$\frac{device\ range}{2^{resolution}} = \frac{10}{2^{16}} = .15\ mV$$

$$\frac{device\ range}{2^{resolution}} = \frac{20}{2^{16}} = .3\ mV$$

The smaller our code width, the more accurate our measurements will be. There are times we must know whether our signals are unipolar or bipolar. Unipolar signals are signals that range from 0 value to a positive value (for example, 0 to 5 V). Bipolar signals are signals that range from a negative to a positive value (for example, –5 to 5 V). If the signal range is smaller than the device range, set our limit settings to values that more accurately reflect our signal range.

PROJECT DESCRIPTION

CHAPTER 4

Project Description

4.1 Introduction

The interfacing of the external world signals to the PC forms the basic feature in the establishment of the whole setup. This is achieved by various tools involved in conversion, processing of the appropriate signals.

4.2 Block Diagram and explanation

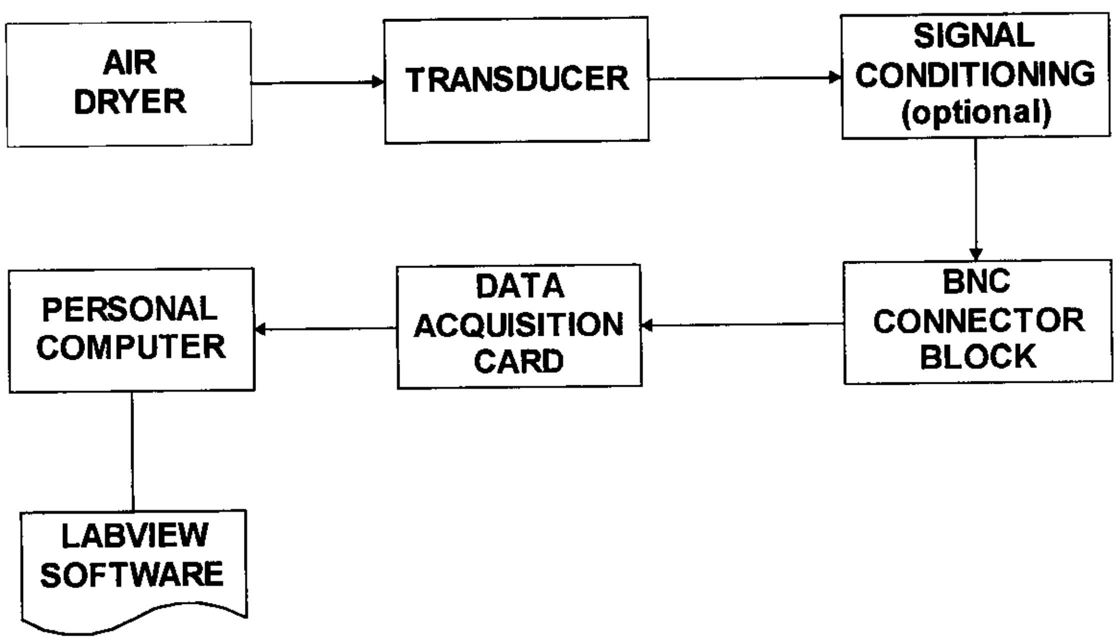


Fig. 4.1 Block Diagram

The Inlet and Outlet pressure from the air dryer is fed to the pressure transducers. The transducer converts the pressure into equivalent voltage or current. The equivalent voltage ranges from 0 to 10V where as the current ranges from 0 to 20mA. If the converted data want to transfer over a long distance, the signal may be lost or distorted. So, the data is signal conditioned if needed. The conditioned signal is connected to the DAQ card through the BNC connector block and cables. The DAQ card is fixed in the add-on slot of the mother board. The DAQ card converts the electrical signal (voltage or current) into equivalent digital data. These data are acquired using LabVIEW software and used for analyzing the performance of air dryer.

4.3 Block connections

The power supply provides 24V to the transducers. The output of the transducer is given to the analog input channels of BNC connector block using probes. Ribbon cable is used to connect BNC connector block and DAQ card. The connections between various sub-systems is shown in fig 4.2.

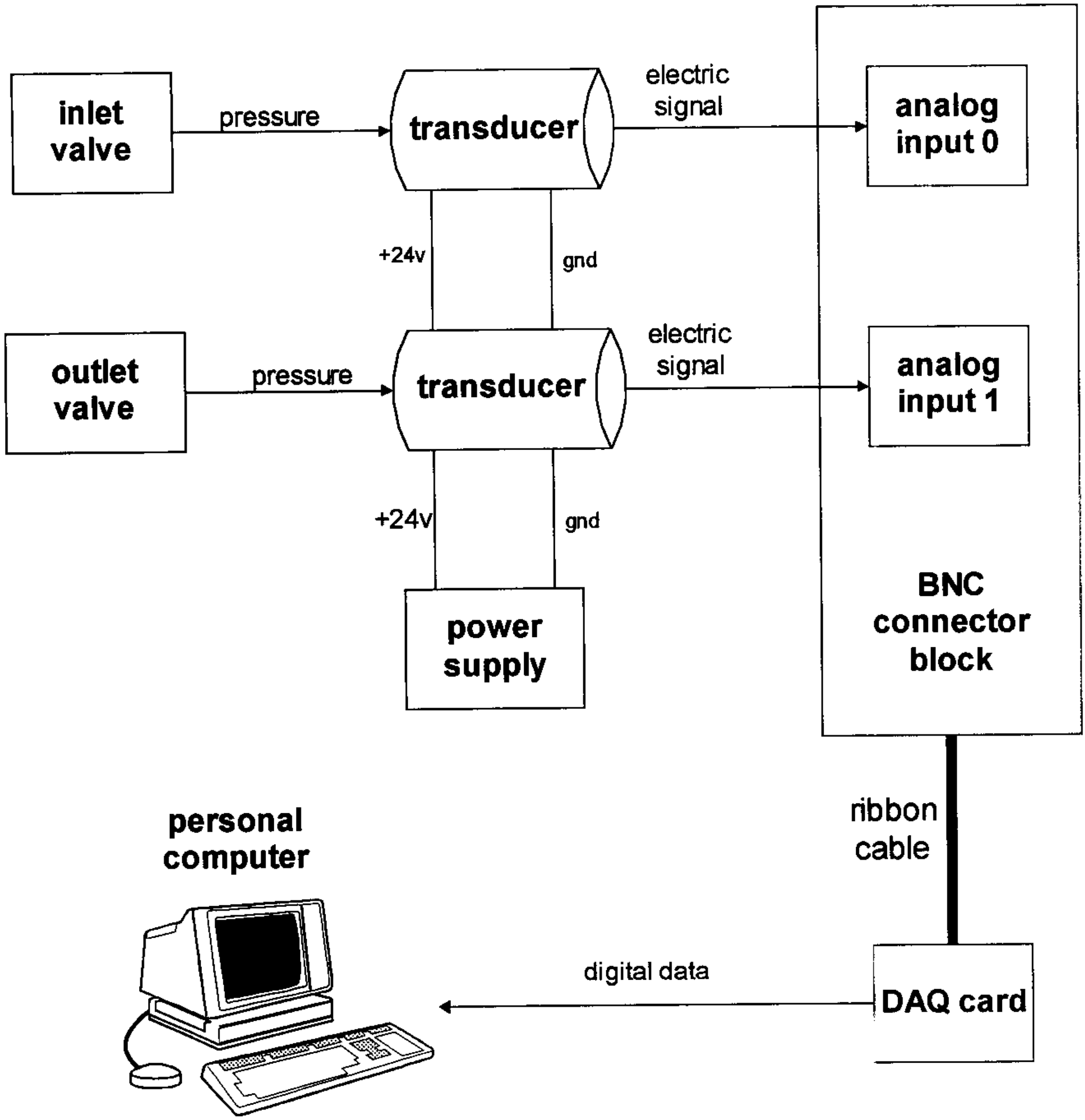


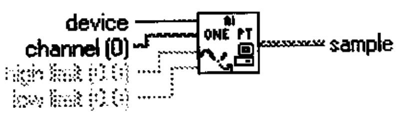
Fig 4.2 Block connections

4.4 Functions in LabVIEW

4.4.1 DAQ Analog Input

Measures the signal attached to the specified channel and returns the measured data. The AI Sample Channel VI performs a single, untimed measurement of a channel. If an error occurs, a dialog box appears, giving the option to stop the VI or continue.

Single-Point Waveform



device is the device number we assigned to the DAQ device during configuration.

channel identifies the analog input channel we want to measure. The default input is channel 0.

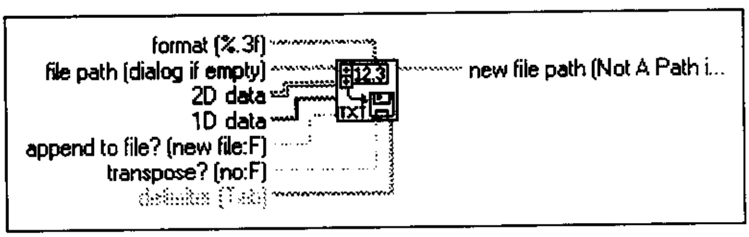
high limit is the highest expected level of the signals we want to measure. The default input is 0. If we enter 0, the system uses the limits entered in the Measurement & Automation Explorer, or if we are using channel names, selects the best limits for our channel configuration. We express the limit value in the units of our readings. Use this value to compute the gain.

low limit is the lowest expected level of the signals we want to measure. The default input is 0. If we enter 0, the system uses the limits entered in the Measurement & Automation Explorer, or if we are using channel names, selects the best limits for our channel configuration. We express the limit value in the units of our readings. Use this value to compute the gain.

sample contains the scaled analog input data for the specified channel.

4.4.2 Write to Spreadsheet file

Converts a 2D or 1D array of single-precision (SGL) numbers to a text string and writes the string to a new byte stream file or appends the string to an existing file.



format specifies how to convert the characters to numbers; the default is %.3f.

file path is the path name of the file. If file path is empty (default value) or is Not A Path, the VI displays a file dialog box from which we can select a file. Error 43 occurs if we select Cancel from the dialog box.

2D data contains the single-precision numbers the VI writes to the file if 1D data is not wired or is empty.

1D data contains the single-precision numbers the VI writes to the file if this input is not empty. The VI converts the 1D array into a 2D array before proceeding. If transpose? is FALSE, each call to this VI creates a new line or row in the file.

append to file? is set to TRUE if we want to append the data to an existing file; we can also set it to TRUE to write to a new file. Set to FALSE (default value) if we want to write the data to a new file or to replace an existing file.

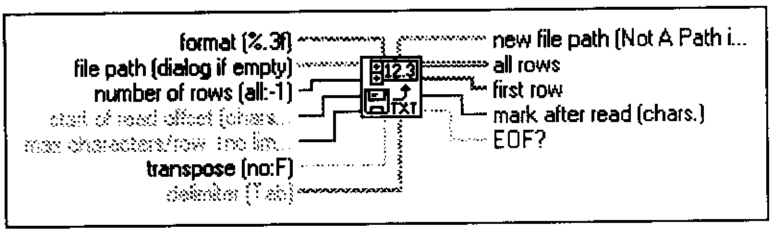
transpose? is set TRUE to transpose the data after converting it from a string. The default value is FALSE.

delimiter contains an input string, which we can use to specify a character or a string of characters as a delimiter, such as tabs, commas, and so on, to use in our spreadsheet.

new file path is the path of the file from which the VI reads data. We can use this output to determine the path of a file that we open using a file dialog box. new file path returns Not A Path if we select Cancel from the dialog box.

4.4.3 Read From Spreadsheet File

Reads a specified number of lines or rows from a numeric text file beginning at a specified character offset and converts the data to a 2D, single-precision array of numbers.



number of rows is the maximum number of rows or lines the VI reads. For this VI, a row is a character string ending with a carriage return, line feed, or a carriage return followed by a line feed; a string ending at the end of file; or a string that has the maximum line length specified by the max characters per row input. If number of rows is <0, the VI reads the entire file. The default value is -1.

start of read offset is the position in the file, measured in characters (or bytes), at which the VI begins reading.

max characters/row is the maximum number of characters the VI reads before ending the search for the end of a row or line. The default is 0, which means that there is no limit to the number of characters the VI reads.

transpose is set TRUE to transpose the data after converting it from a string. The default value is FALSE.

delimiter contains an input string, which we can use to specify a character or a string of characters as a delimiter, such as tabs, commas, and so on, to use in our spreadsheet.

new file path is the path of the file from which the VI reads data. We can use this output to determine the path of a file that we open using a file dialog box. new file path returns Not A Path if we select Cancel from the dialog box.

all rows is the data read from the file in the form of a 2D array of single-precision numbers.

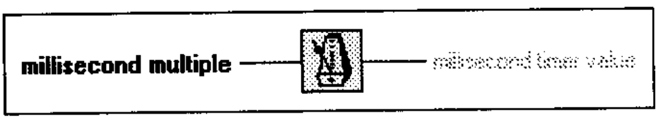
first row is the first row of the all rows array in the form of a 1D array of single-precision numbers. We can use this output when we want to read one row into a 1D array.

mark after read is the location of the file mark after the read; it points to the character (byte) in the file following the last character read.

EOF? is TRUE if we attempt to read past the end of file.

4.4.4 Wait Until Next ms Multiple

Waits until the value of the millisecond timer becomes a multiple of the specified millisecond multiple. Use this function to synchronize activities. We can call this function in a loop to control the loop execution rate. However, it is possible that the first loop period might be short. The Wait Until Next ms Multiple function makes asynchronous system calls, but the nodes themselves function synchronously. Therefore, it does not complete execution until the specified time has elapsed.

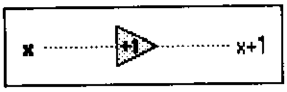


millisecond multiple is the input that specifies how many milliseconds will lapse when the VI runs.

millisecond timer value returns the value of the millisecond timer when the wait expires.

4.4.5 Increment

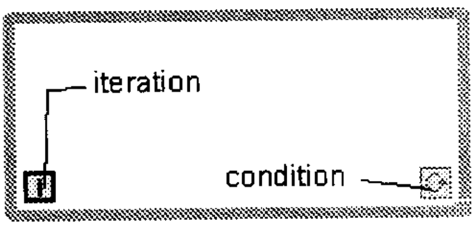
Adds 1 to the input value. If the input is an enumeration, the function wraps the output.



x can be a scalar number, array or cluster of numbers, array of clusters of numbers, and so on.

x+1 is the result of x+1.

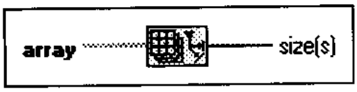
4.4.6 While Loop



The While Loop repeats the subdiagram inside it until the conditional terminal (an input terminal) receives a FALSE or TRUE Boolean value. The default behavior and appearance of the conditional terminal is Continue if True. When a conditional terminal is Continue if True, the While Loop repeats its subdiagram until a FALSE value is passed to the conditional terminal. We can change element or sub array or Element has the same type as the elements of n-dim array or Input Array.

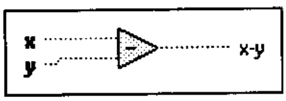
4.4.8 Array Size

Returns the number of elements in each dimension of array.



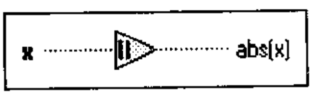
4.4.9 Subtract

Computes the difference of the inputs.



4.4.10 Absolute Value

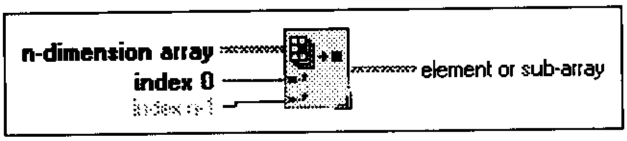
Returns the absolute value of the input.



the behavior and appearance of the conditional terminal by right-clicking the terminal or the border of the While Loop and selecting Stop if True. When a conditional terminal is Stop if True, the While Loop repeats its subdiagram until a TRUE value is passed to the conditional terminal.

4.4.7 Index Array

Returns the element or sub-array or Element of n-dim array or Input Array at index.

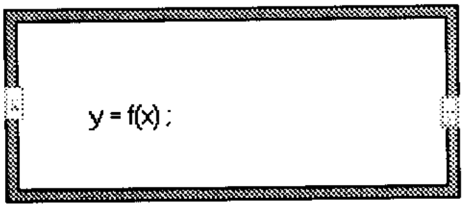


n-dim array or Input Array can be an n-dimensional array of any type, except arrays..

index can be a scalar number. The function coerces index to a 32-bit integer if we wire another representation to it. If the index is out of range (<0 or ³ N, where N is the size of n-dim array or Input Array), the value of element or sub-array or Element is the default value of the data type of array (zero for numbers, FALSE for Boolean controls, and empty for string).

4.4.11 Formula Node:

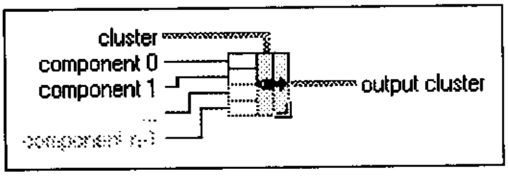
Executes mathematical formulae on the block diagram.



Where x is the input data and y is the output data which is function of x. There can be more than one inputs and outputs. But, each should be used atleast once.

4.4.12 Bundle

Assembles all the individual input components into a single cluster. We also can use this function to change the values of individual elements in an existing cluster without having to specify new values for all elements. To do so, wire the cluster we want to change to the middle terminal of the Bundle function.



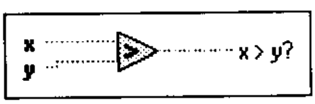
cluster is the cluster whose value we want to change. If we wire a cluster, the elements in the cluster are replaced by the elements already wired in. If we leave cluster unwired a cluster is created.

component 0, component 1, are the input parameters of this function are polymorphic and can be of any data type. If we wire cluster, only those components we want to change must be wired. The unwired components remain unchanged. If we leave cluster unwired, we must wire all the components.

output cluster is the resulting cluster.

4.4.13 Greater?

Returns TRUE if x is greater than y. Otherwise, this function returns FALSE.



4.5 Algorithm

1. Configure the analog inputs as differential inputs.

2. Read data from the channel0 (Outlet Pressure) and from the channel1 (Inlet Pressure).

3. Store the data in a file.

4. Stop the reading process when the required numbers of samples are acquired.

5. Read the stored data from the specified files.

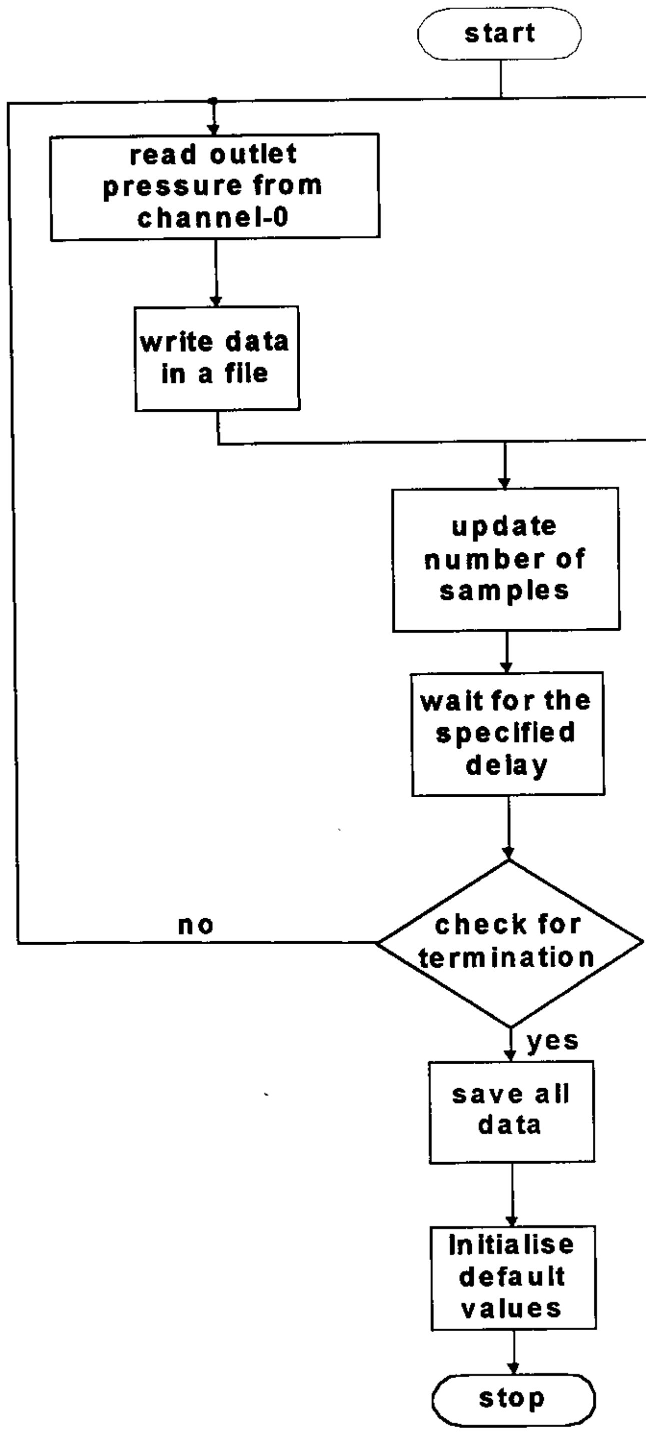
6. Process the retrieved data to obtain the required parameters such as change in pressure, Airflow and Air loss.

7. Plot the characteristics of the Air Dryer using the calculated parameters.

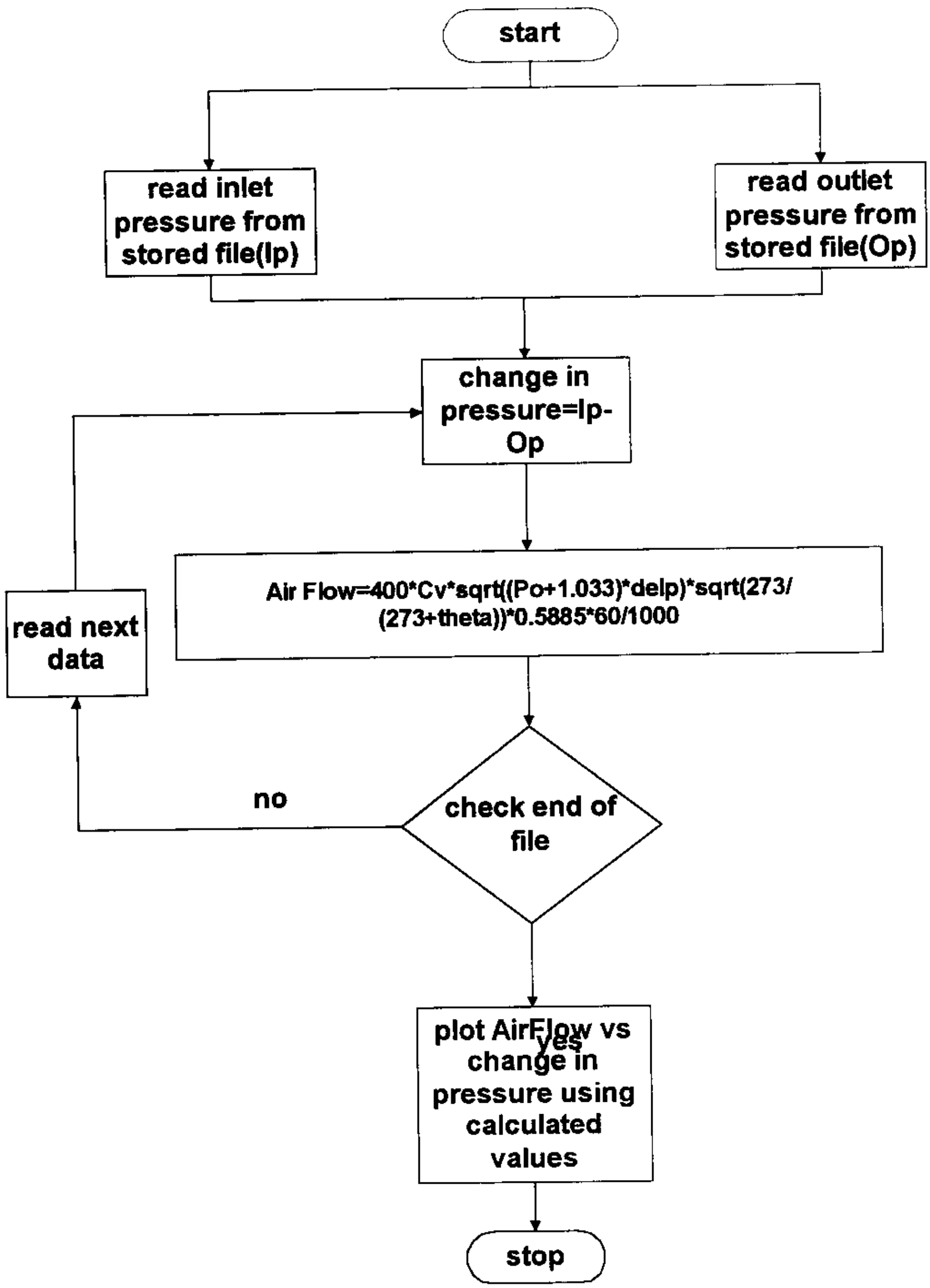
8. Display the value of Air loss in terms of %.

4.6 Flow Charts

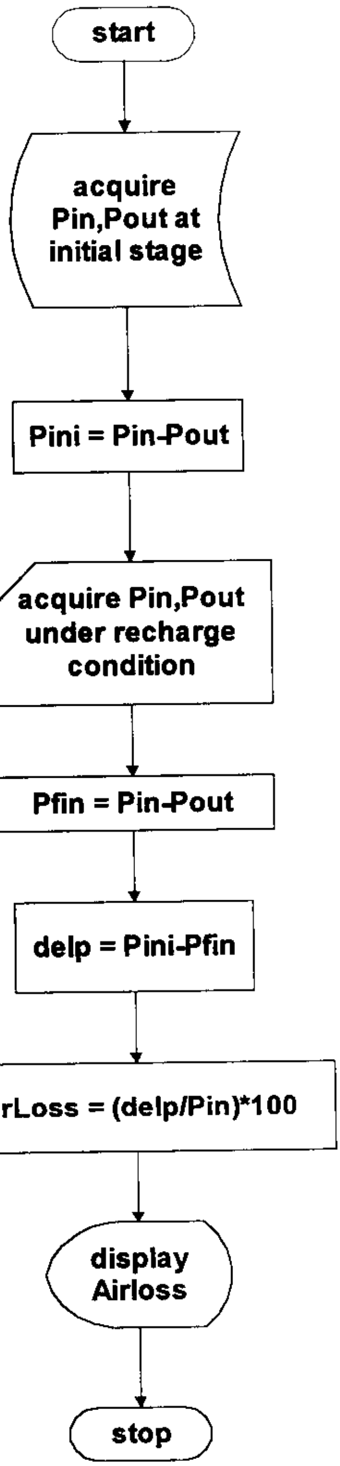
Flowchart for writing data into files



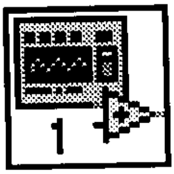
Flowchart for plotting airflow versus delP



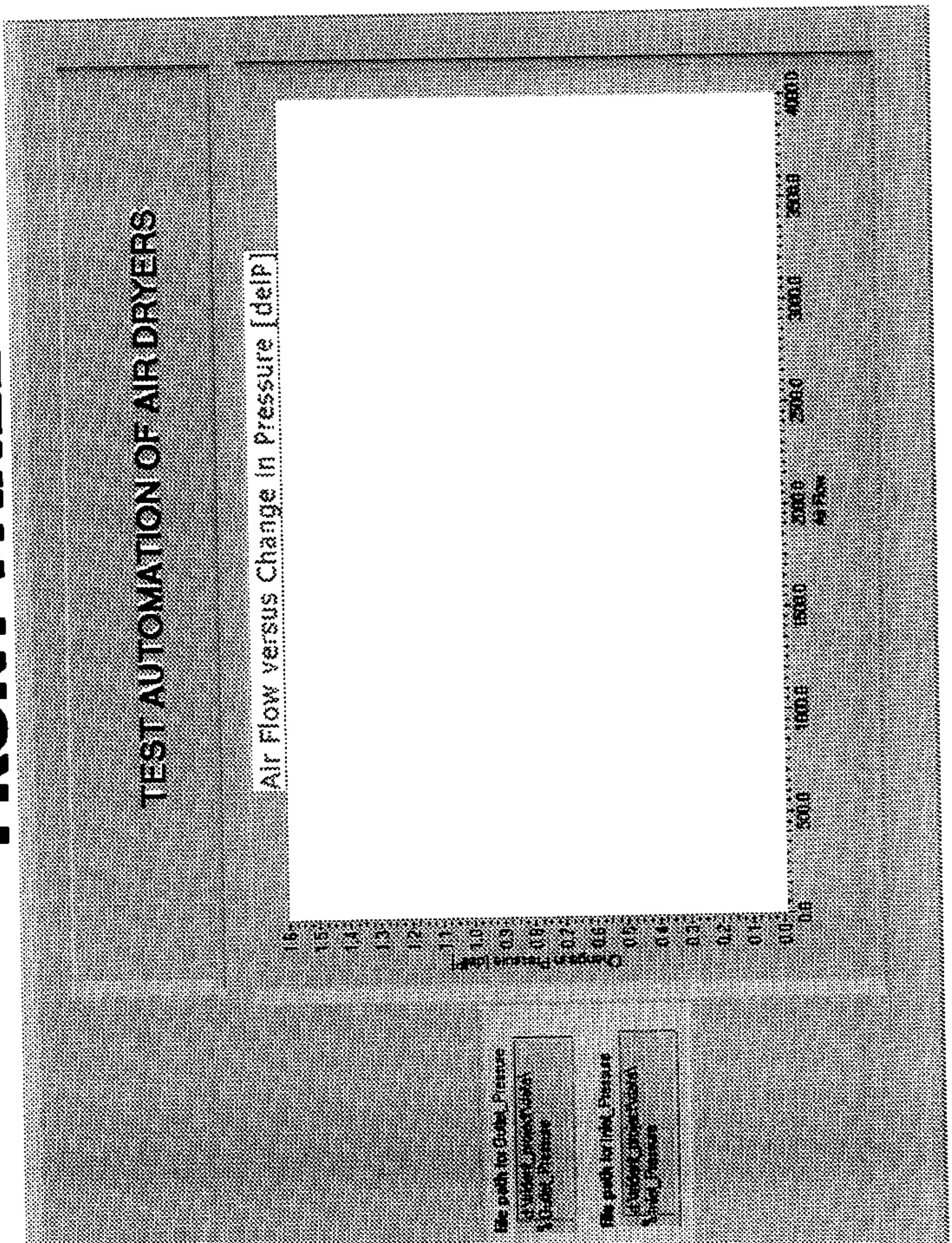
Flowchart for calculating Airloss



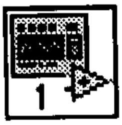
Airflowread.vi



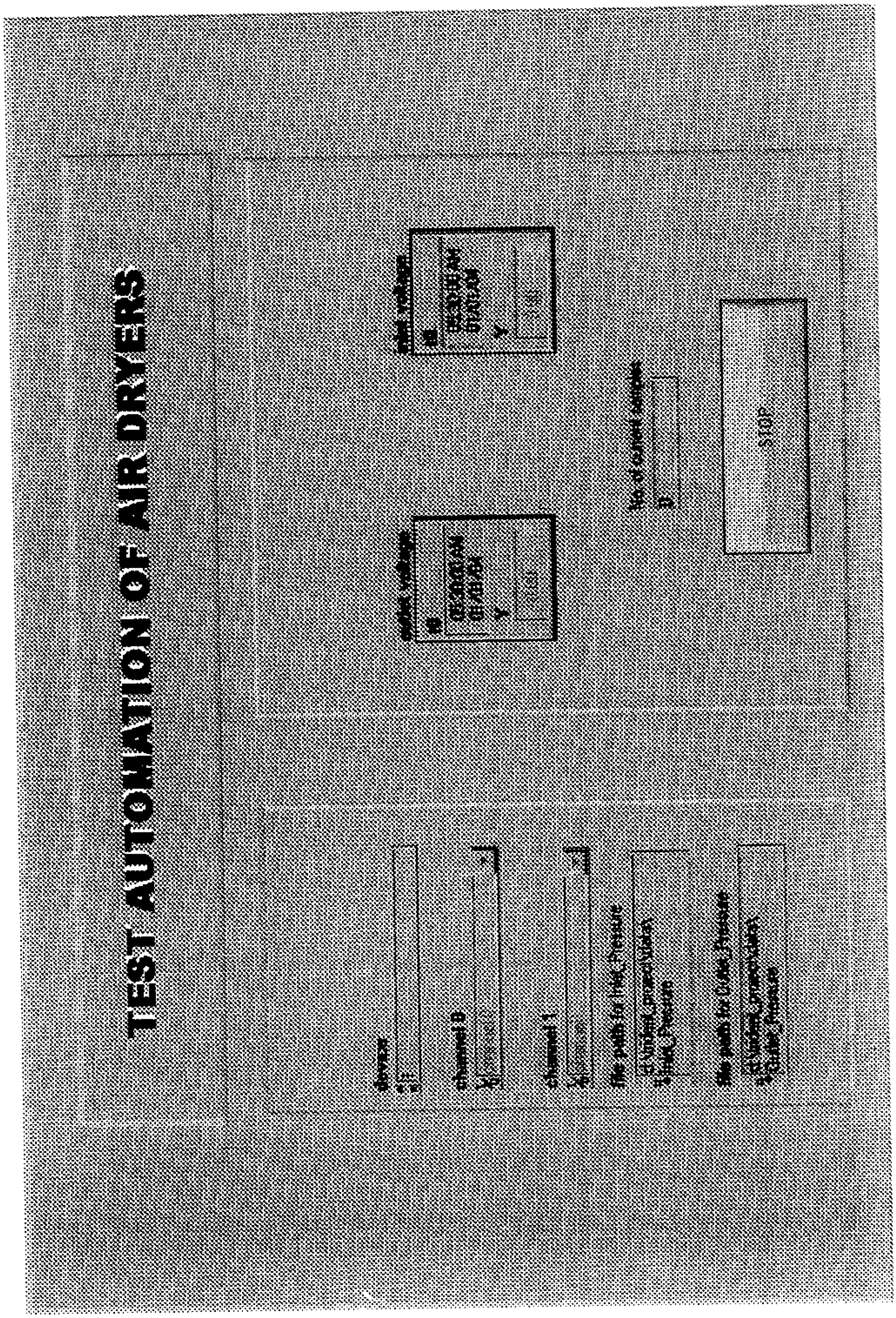
FRONT PANEL



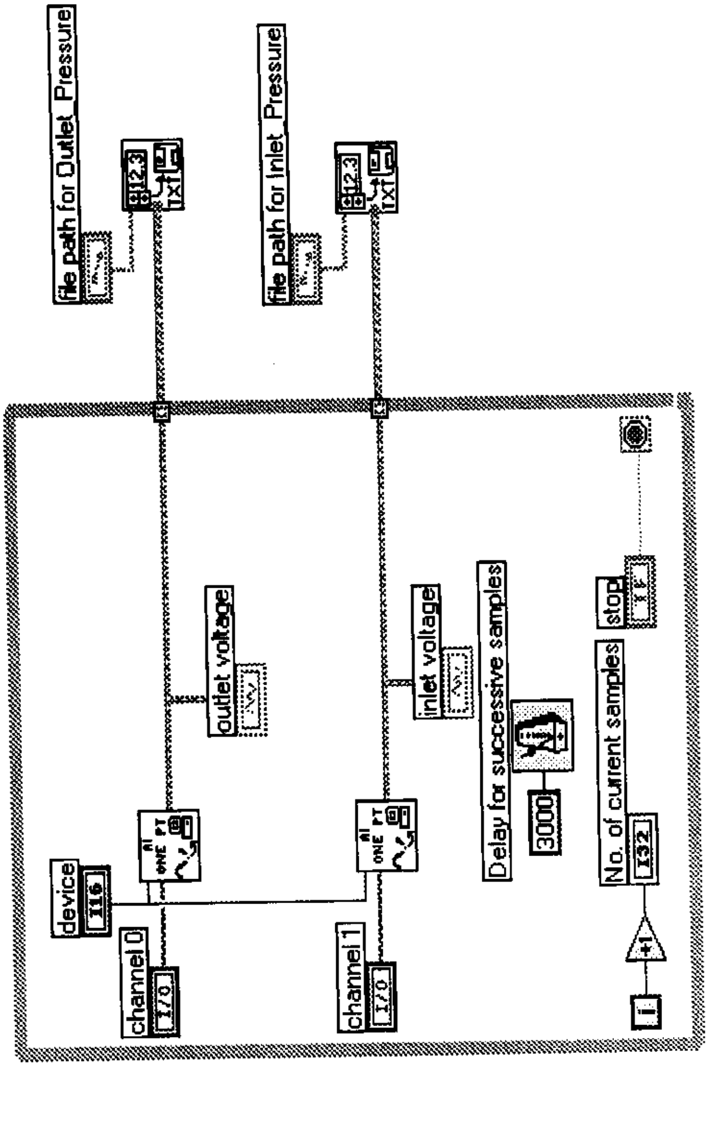
4.7 Coding
Airflowwrite.vi



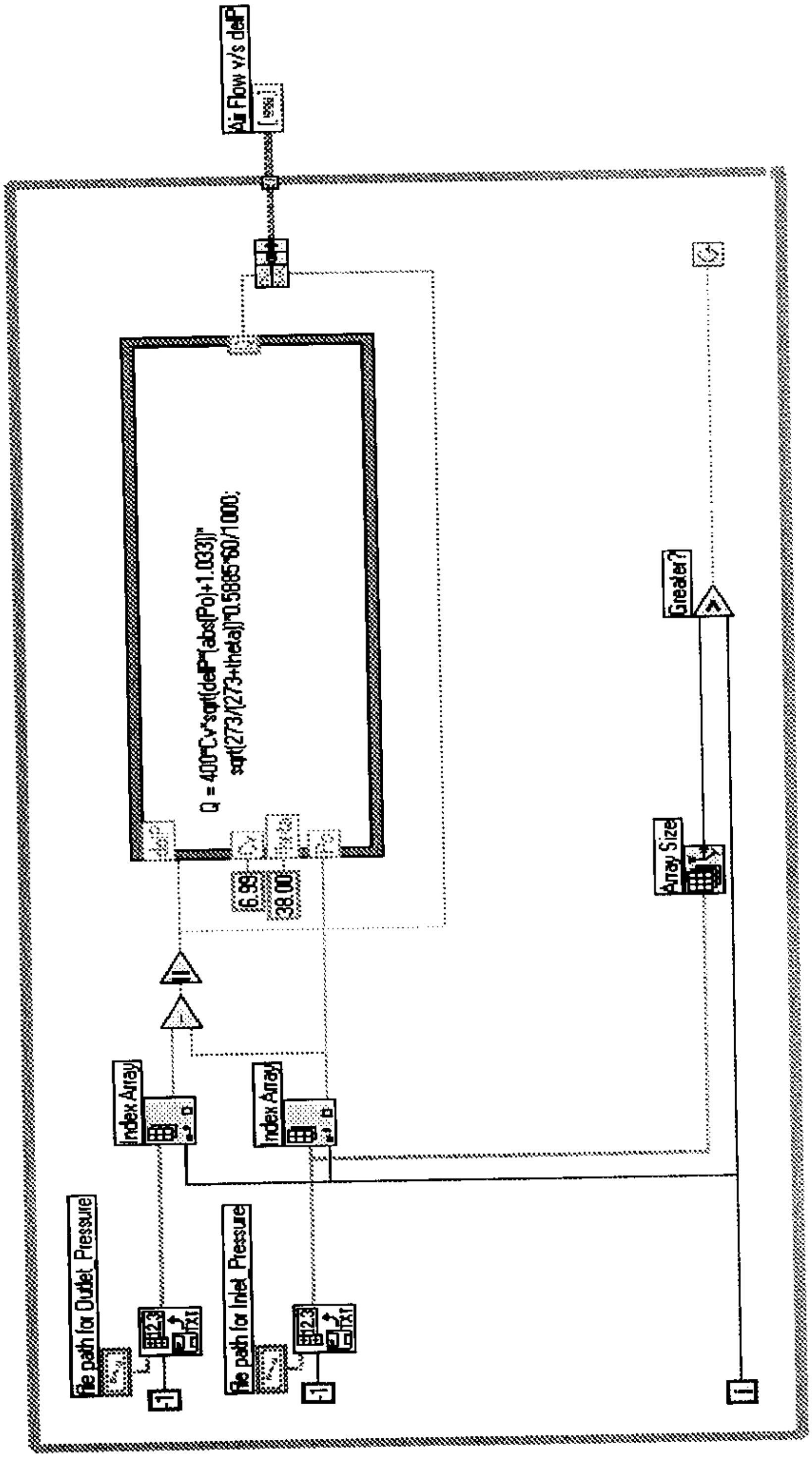
FRONT PANEL



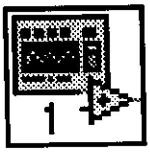
Block Diagram



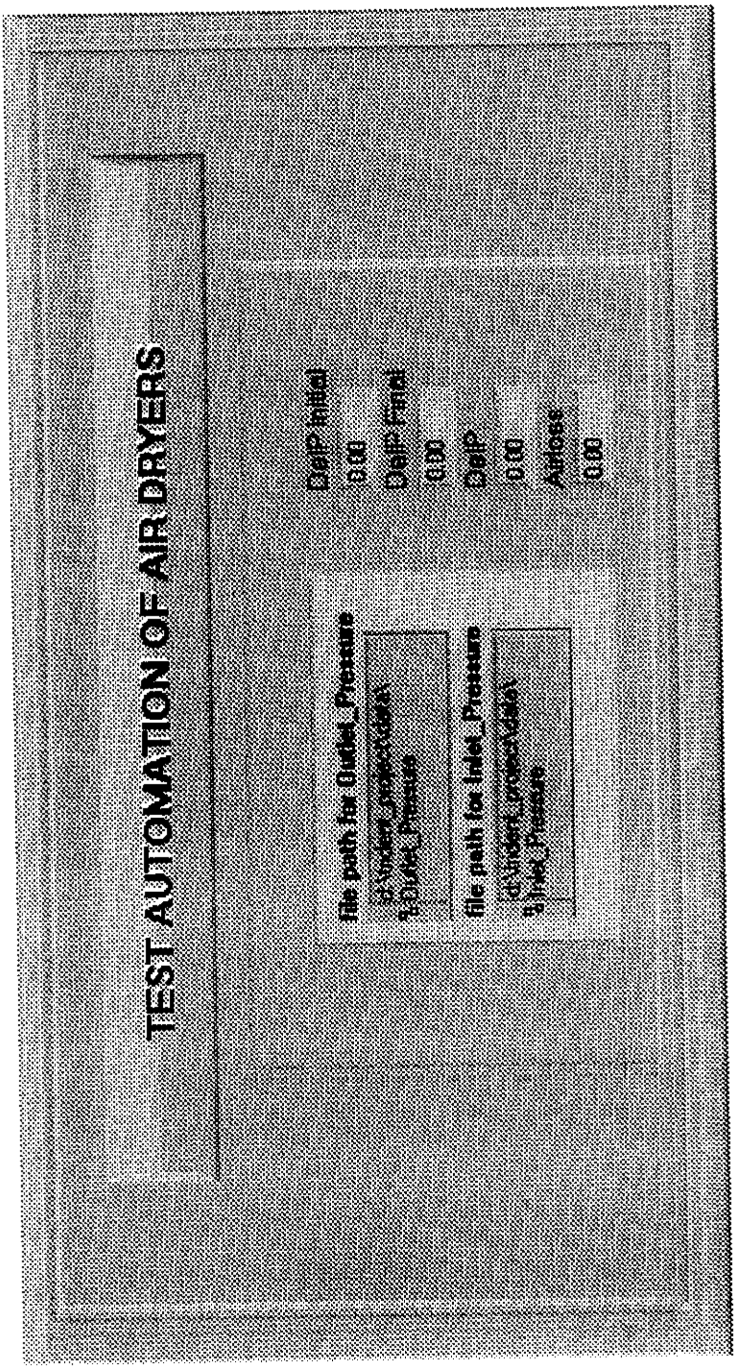
Block Diagram



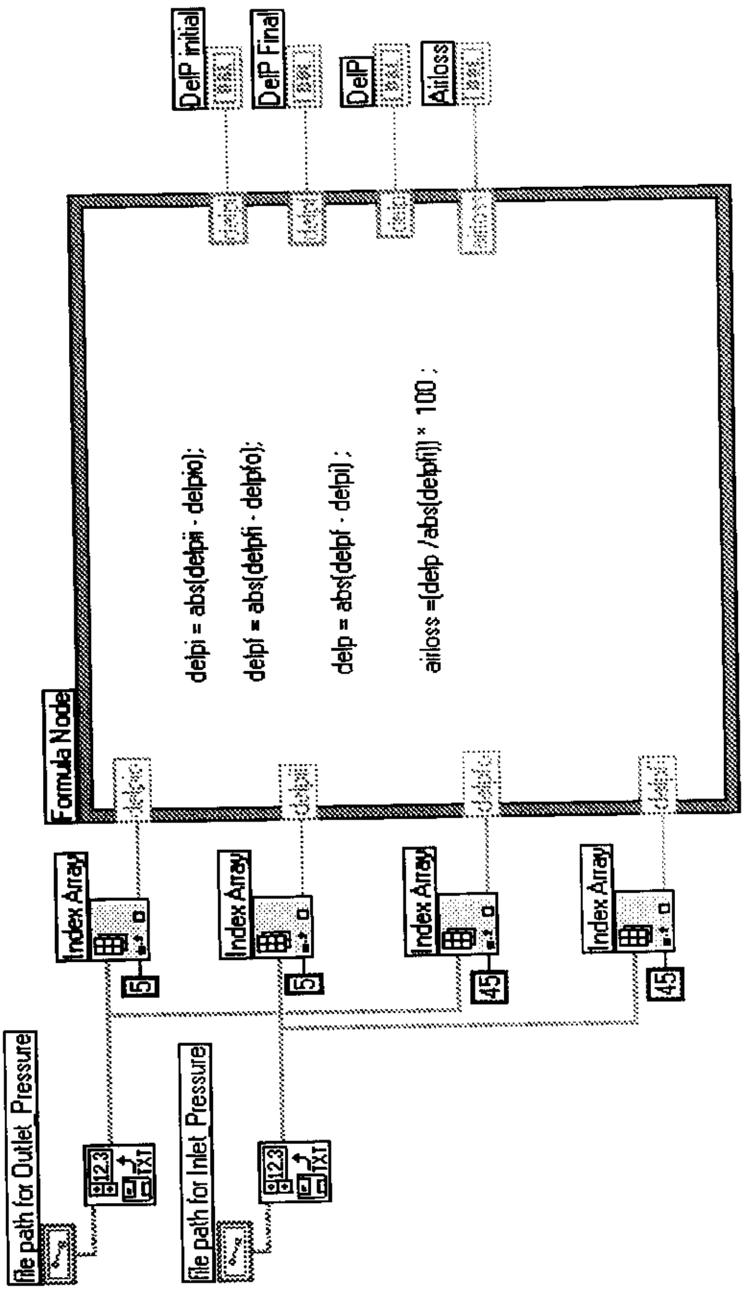
Airlossread.vi



FRONT PANEL



Block Diagram



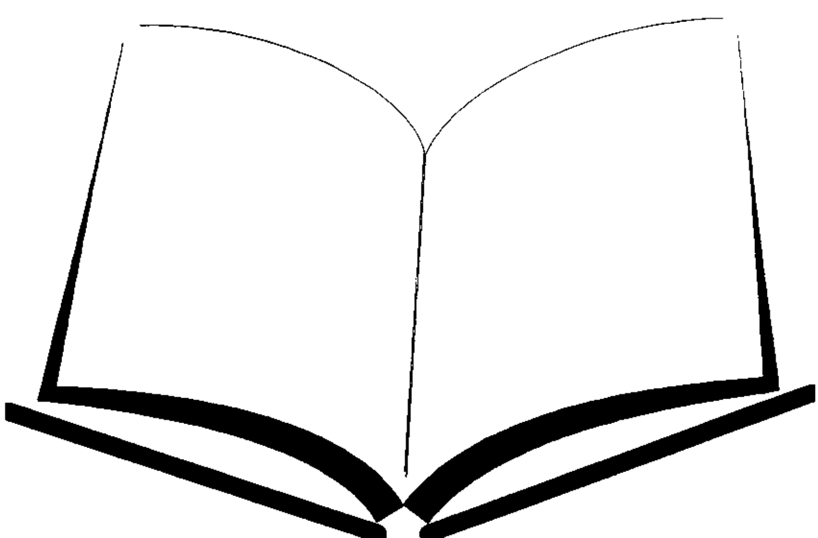
Conclusion

CHAPTER 5

CONCLUSION

The success of the project has resulted in the birth of the Test Automating System for Air Dryers using LabVIEW6i. To sum up briefly, the setup accepts the change in pressure, its corresponding electrical equivalent as the input and gives the variation of the same with Airflow. The Transducer, the Accessory, and the BNC connector serves as the interfacing unit between the device under test viz., the Dryer and the software involved. The input is finally obtained through the DAQ card which is positioned in the add-on slot of the CPU. The whole processing is done by the VIs. Since the system is highly compatible, it needs no signal conditioning provisions. The Air loss is determined with the formula given. This incorporates the basic features of data acquisition cum processing and hence the automation is successfully carried out.

BIBLIOGRAPHY



BIBLIOGRAPHY

- www.ni.com

- www.google.com

- Barry E. Paton, “Sensors, Transducers & LabVIEW”, Prentice Hall Publications.

- Robert H.Bishop, “Learning with LabVIEW”, National Instruments.

- “LabVIEW User Manual”, by National Instruments.

- “Measurement & Automation Catalog”, user guide of National Instruments.

- “Getting Started with LabVIEW”, LaVIEW user manual.

APPENDIX

APPENDIX – A
FESTO PRESSURE TRANSDUCERS

A.1 Technical data

Table A.1

Type	SDE-10 10V/20mA
Part No.	19562
Pressure measurement range	0 to 10 bar
Output voltage	0 to 10 V
Output current	0 to 20 mA
Maximum Pressure	14 bar
Operating temperature	0 to 85 °
Compensated range	10 to 70 °
Storage temperature	-25 to +125 °
Medium	Gasoline 0 Lactic acid 1 Compressed air (lubricated, unlubricated) 0 Mineral oil 0 Phosphoric acid 3 Dry natural gas 1 Nitric acid 3 Crude oil 3 Hydrochloric acid 3 Sulphuric acid 3 Water 3 Heating oil 0 Acetic acid 1
Materials	Anodized aluminium, silicon
Supply voltage	12 to 30 V DC

0 corrosion resistant
1 conditionally resistant
2 limited resistance, not recommended
3 not resistant, cannot be used

A.2 Functions and Connections

The FESTO pressure sensor is a piezo-resistive pressure recorder with integral amplifier and temperature compensation, suitable for many applications. It is mounted inside compact and robust aluminium housing. The pressure to be measured is transmitted onto a piezo-resistive element via a silicon layer. The change of signal thus generated, is output as voltage or current at the connecting plug through an integral amplifier.

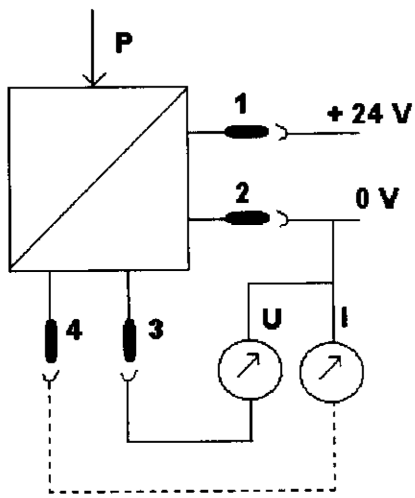


Fig A.1 Transducer Schematic

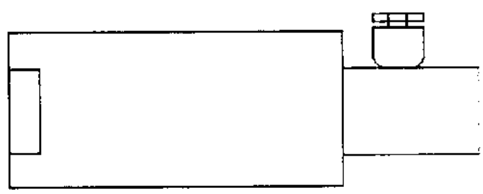


Fig A.2 Transducer General Diagram

The output signal is calibrated, so that the mutual interchangeability of sensors is assured.

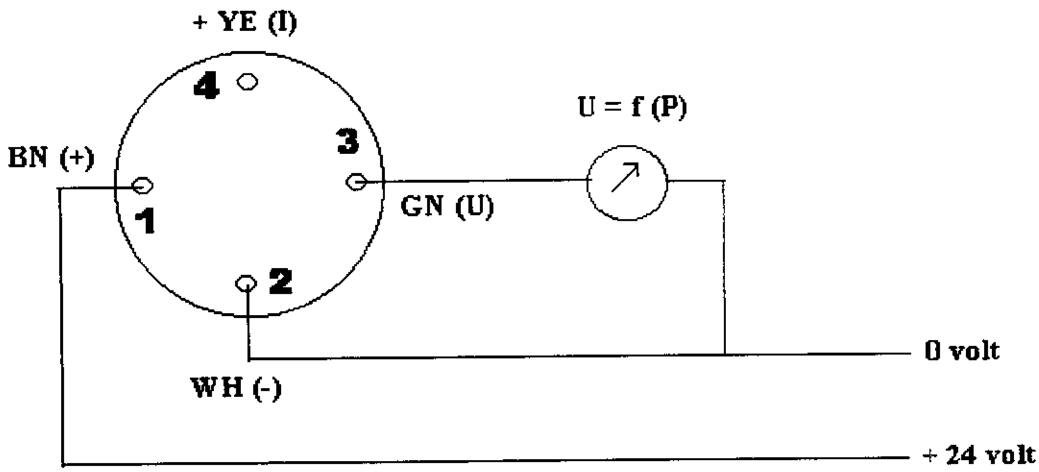


Fig A.3 Transducer connections

Pin allocation:

1. Supply voltage 12 to 30 V DC
2. Earth
3. Voltage terminals
4. Current terminalsa

+Colour code when using connecting cable type KMPYE....

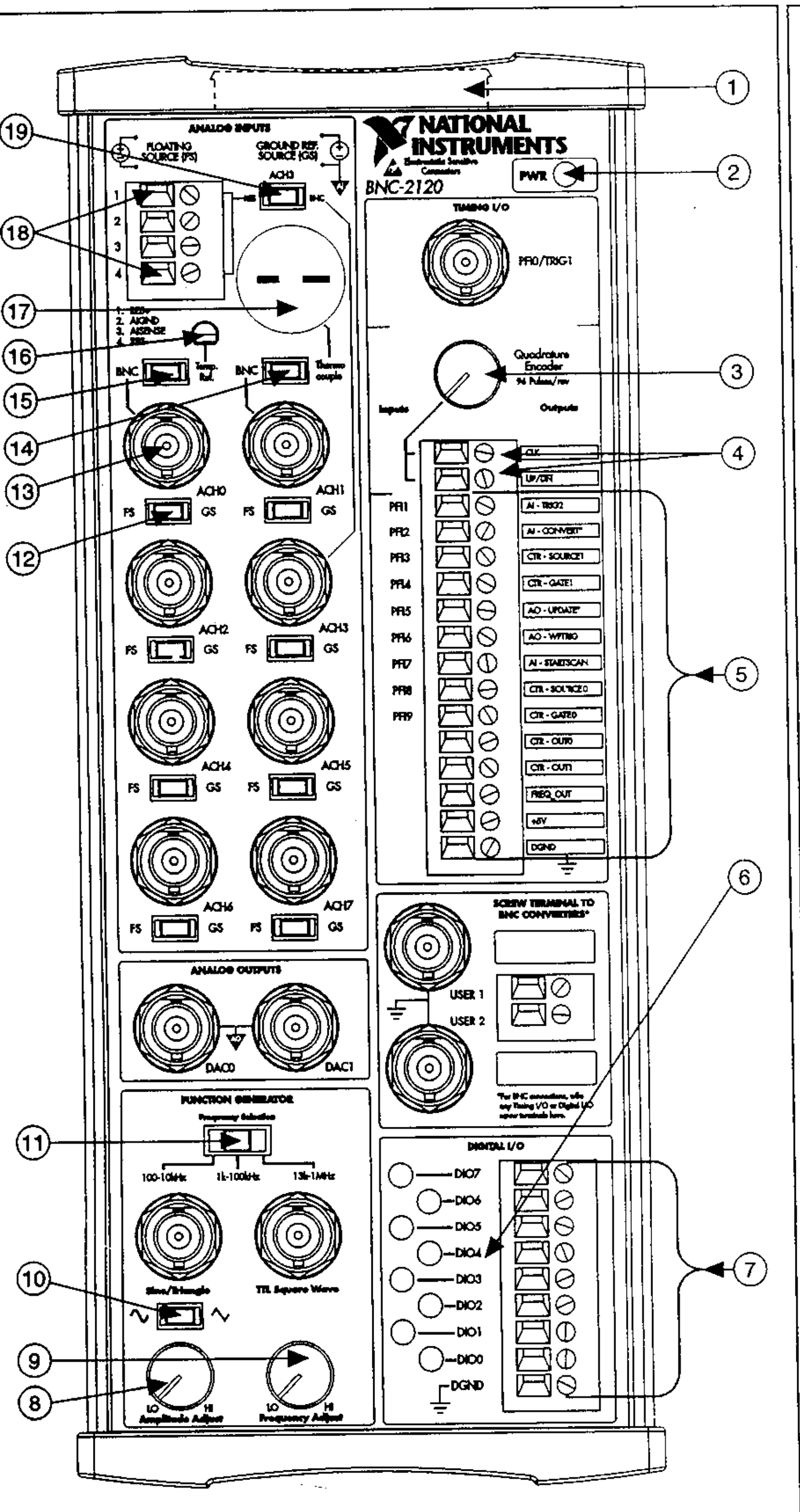
Current and voltage can be measured simultaneously.

APPENDIX-B

BNC-2120 CONNECTOR ACCESSORY FOR E SERIES DEVICES

The BNC-2120 has the following features:

- Eight BNC connectors for analog input (AI) connection with an optional thermocouple connector, an optional temperature reference and optional resistor measurement screw terminals.
- Two BNC connectors for analog output (AO) acquisition.
- Screw terminals for digital input/output (DIO) connection with state indicators.
- Screw terminals for TIO connections.
- Two user-defined BNC connectors.
- A function generator with a frequency-adjustable, TTL-compatible squarewave, and a frequency and amplitude adjustable sine wave or triangular wave.
- A quadrature encoder
- The BNC-2120 has a 68-pin input/output (I/O) connector that connects directly to your E Series drive.



1 68-Position I/O Connector
2 Power LED Indicator
3 Quadrature Encoder Knob
4 Quadrature Encoder Output Screw Terminals
5 TIO Screw Terminals
6 DIO LED State Indicators
7 DIO Screw Terminals
8 Function Generator Amplitude Adjust Knob
9 Function Generator Frequency Adjust Knob
10 Sine/Triangle Waveform Selection Switch
11 Frequency Range Selection Switch
12 Floating Source (FS)/ Ground-Referenced Source (GS) Switch
13 BNC Connector
14 BNC/Thermocouple Selection Switch
15 BNC/Temperature Reference Selection Switch
16 Temperature Reference
17 Thermocouple Input Connector
18 Resistor Measurement Screw Terminals
19 Resistor/BNC Selection Switch

Figure 1. BNC-2120 Front Panel

Specifications

This section lists the specification of the BNC-2120.The specifications are typical at 25° unless otherwise specified.

Analog Input:

Number of channels - Eight differentials.

Field connections - Eight BNC connectors.

Protection - No additional protection provided. Check your DAQ device for specifications.

Optional Inputs:

ACH0 - Temperature Sensor.

ACH1 - Thermocouple.

ACH3, ACH11 - Resistor Measurement (requires RSE configuration).

Optional Connections:

Thermocouple - Uncompensated miniature connector, mates with 2-prong miniature or sub-miniature connector.

Resistor - Two screw terminals.

Resistor measurement range - 100Ohm to 1MegaOhm.

Resistor measurement error - <5%

Screw terminals - Four positions, no larger than 24AWG wire.

Switches - Eight for selecting floating source or grounded source inputs. One for selecting BNC or temperature reference IC. One for selecting BNC or thermocouple connector. One for selecting BNC or resistor screw terminals.

Analog Output:

Field Connection - Two BNC connectors.

Digital Input/Output:

Screw terminals - Nine positions, no longer than 24AWG wire.

LED state indicators - Eight, one per DIO line.

Protection (DC max V)

Powered off - ±5.5V

Powered on	- +10/-5V

Drive

V_{ol}	- 0.6 V, 8mA 1.6V, 24mA.
V_{oh}	- 4.4V, 8mA 4V, 13mA.

Function Generator:

Square Wave	- TTL Compatible.
Frequency Range	-100Hz to 1MHz
Frequency Adjust	-Through Frequency Adjust Knob.
Rise Time	-250nS
Fall Time	-50nS

Sine/Triangle Wave

Frequency Range	-100Hz to 1MHz
Frequency Adjust	-Through Frequency Adjust Knob.
Amplitude Range	-60mV to 4.4 V
Amplitude Adjust	-Through Amplitude Adjust Knob.
Comparison	-Triangle wave is approximately two times the sine wave output.4.4V maximum.

Output Impedance	- 600Ohm

Power Requirement:

+5 V DC (±5%) - 200mA, sourced from the E Series device

Power available at +5Vscrew terminal -E Series power, less power consumed at +5 V DC (±5%).

Physical:

Dimensions - 27.2 by 11.2 by 5.21cm (10.69by 4,41by 2,05in).

I/O Connector - 68-pin male SCSI-II type.

BNC connectors - 15

Screw terminals - 31

Environment:

Operating Temperature - 0 to 50° C.

Storage Temperature - -55 to 125°C.

Relative humidity - 5 to 90% noncondensing.

Timing Input/Output:

Screw Terminals - 14 positions, no larger than 24AWG wire

BNC Connector - One, for PFIO/TRIG1

Protection (DC maxV)

Powered Off - ±1.7V

Powered On - +6.7/-1.7V

APPENDIX-C

NI DAQ-6024E

Specifications

- 16 analog inputs at 200 kS/s, 12-bit resolution
- Up to 2 analog outputs, 12-bit resolution
- 8 digital I/O lines (5 V/TTL/CMOS); two 24-bit counter/timers
- Digital triggering
- 4 analog input signal ranges
- NI-DAQ driver simplifies configuration and measurements

Models

- NI PCI-6023E
- NI PCI-6024E
- NI DAQCard-6024E for PCMCIA
- NI PCI-6025E
- NI PXI-6025E

Operating Systems

- Windows 2000/NT/XP/Me/9x
- Mac OS 9*
- Real-time performance with LabVIEW
- Others such as Linux

Recommended Software

- LabVIEW
- LabWindows/CVI
- Measurement Studio for Visual Basic
- VI Logger

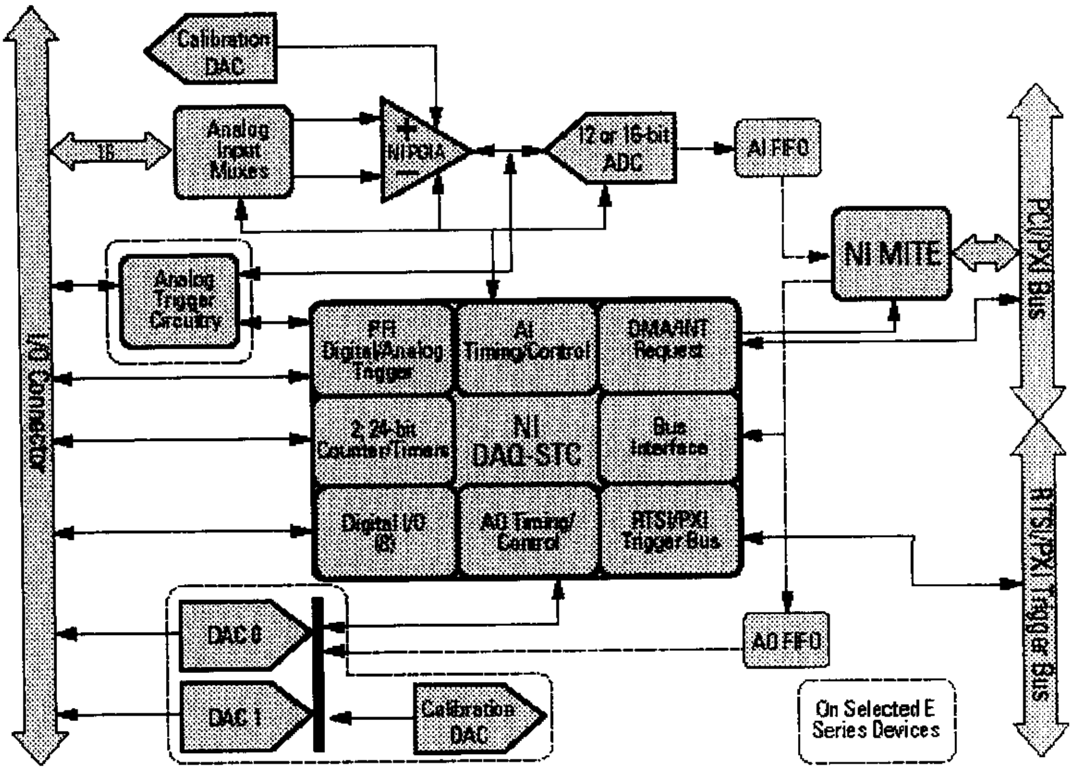
Other Compatible Software

- Visual Basic
- C/C++

Driver Software

- NI-DAQ

Block Diagram



NI-DAQ Software

Driver software for NI data acquisition and signal conditioning Hardware

Short time to first measurement with quick configuration and Application-specific example programs

Named and scaled channels remove configuration complexity

Multiple-device synchronization and integration with RTSI or PXI Trigger bus

Networking features for remote and distributed measurements

Robust double-buffered DMA data management routines.