UNIT I
INTRODUCTION

For Many Years Electronic Instruments Have Been EasilyIdentifiEd Products. Although They Ranged In Size And Functionality, They All Tended To Be BoxShaped ObjectsWith A Control Panel And A Display. StandAlone Electronic Instruments Are Very Powerful, Expensive And Designed ToPerform One Or More SpecifiC Tasks DefiNed By The Vendor. However, The User Generally Cannot Extend Or Customize Them. The Knobs And Buttons On The Instrument, The BuiltIn Circuitry, And The Functions Available To The User, All Of These Are SpecifiC To The Nature Of The Instrument. In Addition, Special Technology And Costly Components Must Be Developed To Build These Instruments, Making Them Very Expensive And Hard To Adapt. Widespread Adoption Of The Pc Over The Past Twenty Years Has Given Rise To A New Way For Scientists And EngiNeers To Measure And Automate The World Around Them. One Major Development Resulting From The Ubiquity Of The Pc Is The Concept Of Virtual Instrumentation. A Virtual Instrument Consists Of An IndustryStandard Computer Or Workstation Equipped With O_TheShelf Application SoftWare, Cost EffEctive Hardware Such As PlugIn Boards, And Driver Software| Which Together Perform The Functions Of Traditional Instruments. Today Virtual Instrumentation Is Coming Of Age, With Engineers And Scientists Using Virtual Instruments In Literally Hundreds Of Thousands Of AppliCations Around The Globe, Resulting In Faster Application Development, Higher Quality Products And Lower Costs.

Virtual Instruments Represent A Fundamental Shift From Traditional Hardware Centred Instrumentation SysTems Towards Software Centred Systems That Exploit The Computing Power, Productivity, Display And Connectivity Capabilities Of Popular Desktop Computers And Work staTions. Although Pc And Integrated Circuit Technologies ExpeRiencedSignifiCant Advances In The Past Two Decades, It Is

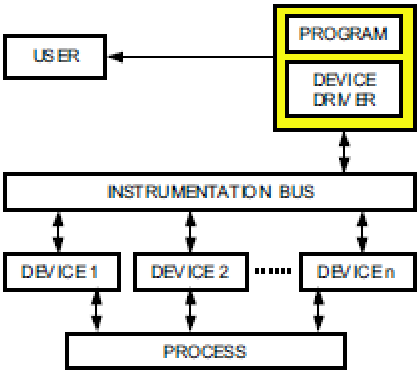


Fig. 1. Conceptual model of early computerized instrumentation

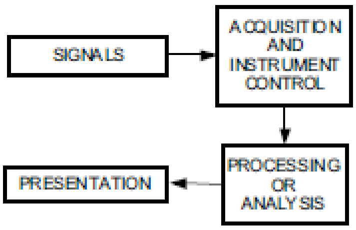


Fig. 2. The diagram of measurement process

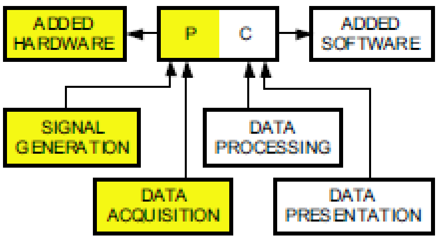


Fig. 3. The general conception of virtual instrument

measurement data. For others, a virtual instrument is a computer equipped with software for a variety of uses including drivers for various peripherals, as well as analogue to digital and digital to analogue converters, representing an alternative to expensive conventional instruments with analogue displays and electronics. Both views are more or less correct. Acquisition of data by a computer can be achieved in various ways and for this reason the understanding of the architecture of the measuring instrument becomes important. A virtual instrument can be defined as an integration of sensors by a PC equipped with specific data acquisition hardware and software to permit measurement data acquisition, processing and display. A virtual instrument can replace the traditional front panel equipped with buttons and display by a virtual front

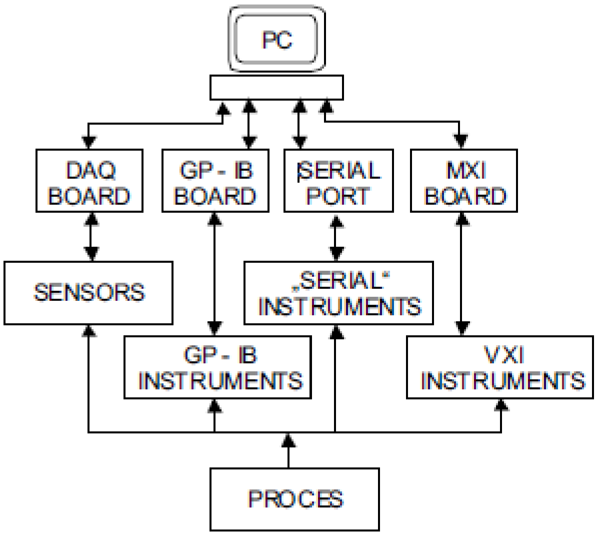


Fig. 4. Structure of the PC-based instrumentation hardware

panel on a PC monitor.

Computer and Display

The computer and the display are the heart of virtual instrument systems. These systems are typically based on a personal computer or workstation with a high resolution monitor, a keyboard, and a mouse. It is important for the chosen computer to meet the system requirements specified by the instrumentation software packages. Rapid technological advancements of PC technology have greatly enhanced virtual instrumentation. Moving rom DOS to Windows gave to PC users the graphical user interface and made 32bit software available for building virtual instruments. The advances in processor performance supplied the power needed to bring new applications within the scope of virtual instrumentation. Faster bus architectures (such as PCI) have eliminated the traditional data transfer bottleneck of older buses(ISA). The future of virtual instrumentation is tightly coupled with PC technology.

3.2 Software

If the computer is the heart of the virtual instrument systems, the software is their brain. The software uniquely denes the functionality and personality of the virtual instrument system. Most software is designed to run on industry standard operating systems on personal computers and workstations. Software implemented can be divided into several levels, which can be described in a hierarchical order.

Register level software

Register level software requires the knowledge of inner register structure of the device (DAQ board, RS 232 instrument, GPIB instrument or VXI module) for entering the bit combination taken from the instruction manual in order to program measurement functions of the device. It is the hardest way in programming. The resulting program is strongly hardware dependent and it is rarely executable on systems with different hardware. area (panel). Logic low is easily established with a point and drag action list. Test Point takes advantage of every Microsoft Windows features. Measurement Studio | is a measurement tool for data acquisition, analysis, visualization and Internet connectivity. This development tool helps you build your test system by integrating into your existing Microsoft compiler. Measurement Studio provides a collection of controls and classes designed for building virtual instrumentation systems inside Visual Basic or Visual C++. With Measurement Studio you can configure plugin data acquisition boards, GPIB instruments, and serial devices from property pages without writing any code. With user interface components you can conure real time 2Dand 3D graphs, knobs, meters, gauges, dials, tanks, thermometers, binary switches, and LEDs. With powerful Internet components, you can share live measurement data among applications via the Internet.

SCPI | Standard commands for programmable Instruments

SCPI is not a software tool as are former systems, but it is an elective aid enabling easy standardized control of programmable instruments. SCPI decreases development time and increases a readability of test programs. SCPI provides an easy understandable command set, guarantees a well defined instrument behavior under all conditions, which prevents unexpected instrument behavior. Although IEEE 488.2 is used as basis of SCPI, it defines programming commands that we can use with any type of hardware or communication link. It has an open structure. The SCPI Consortium continues in adding commands and functionality to the SCPI standard .Real time and embedded control has been long the domain of specialized programs. Advances in industry standard technologies including more reliable operating systems, more powerful processors and computer based real time engineering tools are introducing new levels of control and determinism to virtual instrumentation. This presents new opportunities for scientists to take on increasingly sophisticated real time and embedded development. Software scales across development on the PC into development in real time and embedded applications. Scientists and engineers can move into new application areas without a steep learning curve because the software itself evolves to incorporate emerging computer technologies.

Interconnect Buses

Four types of interconnect buses dominate the industry: the serial connection (serial port), the GPIB, the PC bus and VXI bus. Serial port. Serial communication based on RS232standard is the simplest way of using a computer in measurement applications and control of instruments. Serial communication is readily available via the serial port of any PC and it is limited in data transmission rate and distance (up to 19:2 Kbytes/sec, recently 115 Kbytes/sec,

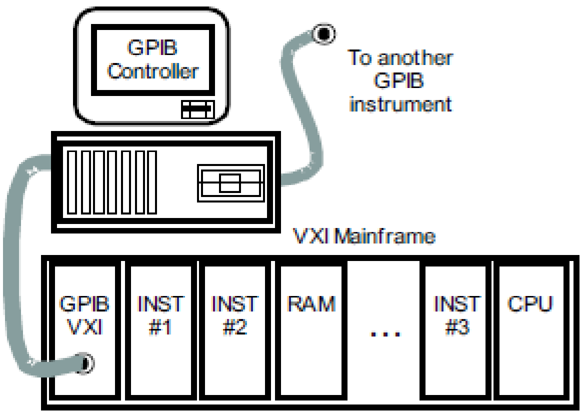


Fig. 5. A VXIbus system controlled by GPIB

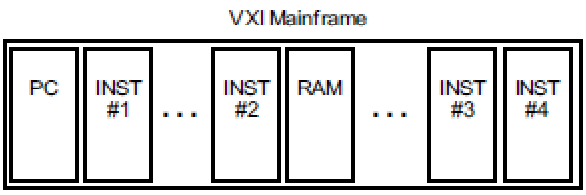


Fig. 7. A VXIbus system controlled by an embedded VXIbus computer inserted into the mainframe

known as mainframes". Mainframes include power supplies, air cooling equipment and backplane communication for the modules. The VXI bus is unique in that it combines a computer backplane based on the VME bus for high speed communication and users a quality EMC environment that allows high performance instrumentation similar to that found in GPIB. As a result, much more compact measuring systems can be built. There are three ways to communicate between the computer and the VXI bus instruments.

a) The rest method is by using GPIB. In this case, a GPIB to VX I bus converter module is plugged into the VXI us mainframe and a standard interface cable connects it and the GPIB interface card in the computer. The advantages and disadvantages of this technique are very similar to a pure GPIB design. This system tends to be easy to program, but data speeds are limited to GPIB speeds. However, because the internal data speeds within the VXI bus mainframe can exceed10 Mbytes/s, often a high speed application is solved by local high speed acquisition and processing occurring within the mainframe and high level results transfer to the computer over GPIB. Figure 5 shows an example of VXI bus system using GPIB.

b) The second technique is to use a higher speed interconnect bus between the VXI bus mainframe and the computer. The most common implementation of this is high speed exible cable interface known as MXI bus .As in GPIB, an MXI bus interface card and software are installed on the computer and a cable attaches it to an MXI bus to VXI bus converter module in the VXI bus mainframe. MXI bus is essentially an implementation of the VXI bus on aexible cable. This means that the

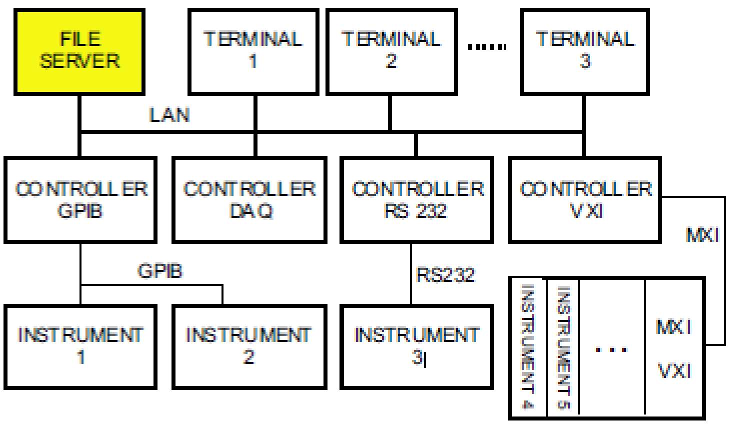


Fig. 8. Block diagram of distributed measurement system based on LAN

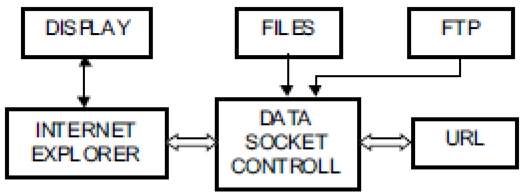


Fig. 9. The architecture of a distributed system based on Internet

to the use of such systems. As in the case of large and complex plants, a structured networked measurement system can be adopted by scaling its use to the geographical area. The geographical process to be monitored and controlled is partitioned into cells that can be dealt with by a single processing unit or a group of locally connected units.

Geographically distributed units are connected by a geographical computer network into a distributed measurement system. In this case communication delays usually cannot be neglected. This is even more relevant if the traffic in the computer network is not negligible due to the number of computers connected and the amount of communications, especially if a public computer network is used to realize the interconnections among the measuring processing units. It seems that in the near future local network (LAN)can be considered as a kind of measurement bus, from the viewpoint of measurement and control systems. Atypical example of such a system including various virtual instruments is presented in Fig. 8. It can be considered as arst step to a wider, Internet based technology .In the last few years a surprisingly rapid growth of fast and reliable communication networks has allowed an easy interchange of information and commands between computers both connected to local networks and connected to faraway site of wide area networks (WAN), such asthe Internet. Thus, network services and programmable instrumentation now permit the development of measurement laboratories distributed on a wide geographical area and simultaneously available to several users variously located in the territory.

Common Internet based software can be used to provide easy data migration between various communication to the emergence of virtual instruments, which are substantially different from their physical ancestors. Virtual instruments are manifested in different forms ranging from graphical instrument panels to complete instrument systems. Modular instrumentation building blocks are becoming more prevalent in the industry and are allowing users to develop capabilities unattainable using traditional instrument architectures. Despite these changes however, the measurement paradigm remains unaltered. This might be the proper platform for the new development. The trend in virtual instrumentation increasingly integrates the measurement systems into more complex monitoring and control systems distributed over different (possibly geographically distant) locations. The remote instrumentation control is becoming popular since the networks have become reliable and worldwide and almost every new instrument embeds programmable capabilities. The past has shown that unless proper standards are available, diversification due to adhoc solutions will slow the progress in the field. Thus, it seems a proper challenge for the future to start thinking of standardization of virtual instrumentation and distributed measurement systems.

THE CONCEPTION OFVIRTUAL INSTRUMENT

Usually instrumentation manufacturers provide specific functions to given architecture and fixed interfaces for measuring devices, and thus limit the application domain of these devices. In actual use much time is required for adjusting the measuring range and for saving and documenting the results.

The advent of microprocessors in the measurement and instrumentation fields produced rapid modifications of measuring device technology, soon followed by the appearance of computer based measurement techniques.

Conceptual model of early computerized instrumentation is given in Fig. 1.A single user controls the system, which runs exclusively on a piece of hardware. There is a single control structure, which is formed by the combination of the user and the program that controls the multiple devices attached to the instrumentation bus. The main challenges are the device coupling and the programming models.

The measurement consists of three parts, as shown in Fig. 2, acquisition of measurement data or signals, conditioning and processing of analysis of measurement signals and presentation of data. The concept of virtual instrument is frequently used in industrial measurement practice, but not always with precisely the same meaning. For some people, virtual instruments are based on standard computers and represent systems for storage, processing and presentation of means of integration of the display, control and centralization of complex measurement systems.

Industrial instrumentation applications, however, require high rates, long distances, and multi vendor instrument connectivity based on open industrial network protocols. In order to construct a virtual instrument it is necessary to combine the hardware and software element is which should perform data acquisition and control, data processing and data presentation in a different way to take maximum advantage of the PC. It seems that in the future the restrictions of instruments will move more and more from hardware. Such a general conception of virtual instrumentation is presented in Fig. 3.The vendor of virtual instrument can use the serial communication based on RS232 standard or the parallel communication based on GPIB standard (known also as HPIB, IEEE 488.12 or IEC 625.12), PC bus, or VXI bus (VME extension for Instrumentation).
The main categories of virtual instruments:
a) Graphical front panel on the computer screen to control the modules or instruments
a1) controlled module is plug in DAQ board,
a2) controlled instrument is based on GPIB board,
a3) controlled instrument is connected via serial port,
a4) controlled instrument is VXI board (or system).

b) Graphical front panel with no physical instruments at all connected to the computer. Instead, the computer acquires and analyses the data from _les or from other computers on a network, or it may even calculate its data mathematically to simulate a physical process or event rather than acquiring actual real world data.
To the PC connections according to point a) the following process measuring devices are attached:
1. Sensors
2.GPIB instruments
3. Serial instruments
4.VXI instruments
This structure is a result of international standardization allowing more freedom in using boards and instruments from various manufactures. The main representative features of virtual instruments describing their functionality are following:
1. Enhancing traditional instrument functionality with computers;
2.Opening the architecture of instruments;
3.Widespread recognition and adoption of virtual instrument software development frameworks.

BASIC COMPONENTS OF VIRTUAL INSTRUMENTS

The basic components of all virtual instruments include a computer and a display, the virtual instrument software, a bus structure (that connects the computer with the instrument hardware) and the instrument hardware.

Driver level software

One of the most important components in measurement systems today is the device driver software. Device drivers perform the actual communication and control of the instrument hardware in the system. They provided medium level easy to use programming model that enables complete access to complex measurement capabilities of the instrument In the past programmers spent a significant amount of time writing this software from scratch for each instrument of the system. Today, instrument drivers are delivered as modular, other shelf components to be used in application programs. Several leading companies formed(in 1988) the Interchangeable Virtual Instrument (IVI) Foundation. The IVI Foundation was formed to establish formal standards for instrument drivers and to address the limitations of the former approaches.

High-level tool software

Currently the most popular way of programming is based on the high level tool software. With easy to use integrated development tools, design engineers can quickly create, configure and display measurements in a user friendly form, during product design, and verification. The most known, popular tools are as follows:_
Lab VIEW (Laboratory Virtual Instrument Engineering Workbench) | is a highly productive graphical pro
gamming language for building data acquisition and instrumentation systems. To specify the system functionality one intuitively assembles block diagrams | a natural design notation for engineers. Its tight integration with measurement hardware facilities rapid development of data acquisition, analysis and presentation of solutions. Lab Windows /CVI (C for Virtual Instrumentation) |is a Windows based, interactive ANSI C programming environment designed for building virtual instrumentation applications. It delivers a draganddrop editor for building user interfaces, a complete ANSI C environment for building test program logic, and a collection of automated code generation tools, as well as utilities for building automated test systems and monitoring applications of laboratory experiments. The main power of CVI lies in
the set of libraries. HP VEE (Hewlett-Packard's Visual Engineering Environment) | allows graphical programming for instrumentation applications. It is a kind of Visual Engineering Environment, an iconic programming language for solving engineering problems. It also provides an opportunity to gather, analyze and display data without conventional(text based) programming .Test Point | is a Windows based object oriented software package that contains extensive GPIB instrument and DAQ board support. It contains a novel state of the art user interface that is easy to use. Objects, called\stocks" are selected and dragged with a mouse to a work and 15 m) and it allows only one device to be connected to a PC.GPIB. It was the rest industry standard bus for connecting computers with instrumentation. A major advantage of GPIB is that the interface can be embedded on there are of a standard instrument. This allows dual use of the instrument: as a standalone manual instrument or as a computer controlled instrument. Because of this feature, there are a wide variety of high performance GPIB instruments to choose from. The GPIB operas exiblecable that connects a GPIB interface card in the computer to up to 15 instruments over a distance of up to twenty meters. The interface card comes with software that allows transmission of commands to an instrument and reading of results. Each GPIB instrument comes with a documented list of commands for initiating each function. Typically, there is no additional software delivered with the instrument. GPIB has a maximum data rate of 1 Mbytes/s and typical data transfers are between 100and 250 Kbytes/s. It depends on the response of the measured subject. PC bus. With the rapid acceptance of the IBM personal computer in test and measurement applications, there has been a corresponding growth of plug in instrumentation cards that are inserted into spare slots. However, high accuracy instruments require significant circuit board space to achieve their intended precision. Because of the limited printed circuit board space and close proximity to sources of electromagnetic interference, PC

bus instruments tend to be of lower performance than GPIB instruments but also of lower cost. Many are simple ADCs, DACs, and digital I/O cards. PC bus instrumentation is best suited for creating small, inexpensive acquisition systems where the performance is not of paramount importance. Since these cards plug directly into the computer backplane and contain no embedded command interpreter as found in GPIB instruments, personal computer plug in cards are nearly always delivered with driver software so that they can be operated from a personal computer. This software may or may not be compatible with other virtual instrument software packages, so it is recommended to check with the vendors beforehand. Most data acquisition boards are multifunctional, ie they accept both analogue and digital signals. These plug in data acquisition boards gain wider and wider acceptance due to their low price and high edibility obtained from the associated software.

VXI bus. In the late eighties, the VME extension for Instrumentation (VXI) standard allowed communication among units with transfer over 20 Mbytes/second between VXI systems. VXI instruments are installed in arrack and are controlled by, and communicate directly with, a VXI computer. These VXI instruments do not have buttons or switches for direct local control and do not have local display typical in traditional instruments.

It is an open system instrument architecture that combines many of the advantages of GPIB and computer backplane buses. VXI bus instruments are plug in modules that are inserted into specially designed card

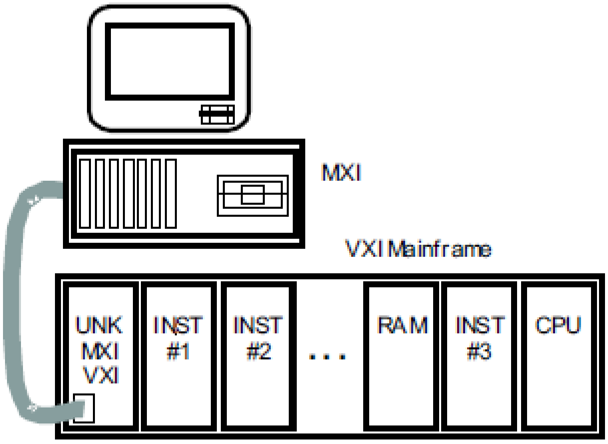


Fig. 6. A VXIbus system controlled over a high-speed MXIbus cable

cages

conversions to VXI bus are simple and fast, bringing MXI bus performance within a factor of 2 or so of native VXI bus speeds. The advantage of MXI bus is that it allows the use of the shelf computers to communicate with VXI bus instruments at a speed consider ably higher than GPIB. A disadvantage is that the MXI bus cable can be thick and unwieldy, and there is some loss of data transfer bandwidth due to the con version. Figure 6 shows an example VXI bus system using MXI bus.

c) The third way is to insert powerful VXI bus computers directly into the VXI bus mainframe. VXI bus computers tend to be repackaged versions of industry standard personal computers and workstations that run industry standard operating systems and software. The advantage of this technique is that it preserves the full communications performance of VXI bus. The disadvantage is that the choice of VXI bus computers will always be a subset of the choice of standard industry computers. VXI bus computer technology will typically lag behind the performance of the industry as a whole, offer fewer alternative configurations and be priced at a premium due to its lower volume. Figure 7 shows an example VXI bus system using an embedded computer.

3.4 Instrument Hardware The preceding subsection on interfaces also touches on the attributes found in each of the respective instrument hardware products. One note is worth to be repeated: Virtual instrumentation never eliminates the instrument hardware completely. To measure the real world there will always be some sort of measurement hardware, sensor, transducer and conditioning circuit, but the physical form factor of this instrumentation may continue to evolve.

DISTRIBUTED MEASUREMENT SYSTEMS

The present trend in interconnected measurement systems is to extend the area covered by the interconnected systems in the geographical scale. This sets a further limit pathways. Multicomputer processing systems are effective in creating complex systems by overcoming limitations of a single computer concerned with the overall computing power or the number of signals to be acquired and processed. Standard software languages such as C and Java can be used with other shelf development tools to implement the embedded network node applications and the web based applications respectively. Internet based TCP/I P protocols, Ethernet technology and/or Data Sockets can be used to design the networking infrastructure, Fig. 9.DataSocket is a software technology for Windows that makes sharing all measurements across a network (remote Web and FTP sites) as easy as writing information to able. It uses URLs to address data by the same way we use URL in a Web browser to specify Web pages. Data Socket included with any software tool is ideal when someone wishes to complete control over the distribution of the measurement but does not want to learn the intri cacies of the TCP/IP data transfer protocols.
In all types of networked and distributed measurement systems presented above, real time operation and condistributed measurement system one can take remote measurements, distribute a program's execution, or publish measurement data over the Internet. The evolved hardware and software technologies provide users with the tools they need for easy building of a powerful distributed system. By publishing your measurement or automation application over the Internet real time data can be viewed by users on remote computers. With application development environments Web servers are available so you can publish a user interface to the Internet. Without any additional programming you can publish your front panel as a Web page so users across the Internet can view these panels running within any standard Web browser. Applications have one or more measurement nodes physically separated from the computer that is controlling them and collecting data. Remote measurement applications often require high speed streaming of data and several clients connected to a single measurement. For streaming measurement data across a network Data Socket provides you with an easy to use interface. Using Data Socket you can easily stream any kind of measurement data across a local area network or the Internet to several client programs. Both Web servers and Data Socket provide a simple and convenient way to publish your measurement data.

UNIT II
DATA ACQUISTITION IN VI

DATA ACQUISITION BOARD AND CONFIGURATION SOFTWARE

The ATMIO16DE10 DAQ board utilized belongs to the family of enhanced multiple function input/output boards [8, 9]. It has digital as well as analog input/output capabilities. It has a total of 16 analog input channels, which can be used in, single ended and differential modes, all software selectable. All 16 channels can be used in single ended mode only if all the signals have a common ground. Otherwise, only eight analog input channels are available. In the differential mode, input voltage range is 0v to 1v. It has two twelve bit analog output channels, whose output range is 5v to +5v or from 0v to 10v, software selectable, with a current of 5mA to +5mA. The sampling rate is 100 k samples/sec. There are, a total of thirty two digital input/output channels. It is also equipped with 24bit, 20 MHz counter/timers. Digital channels are compatible with TTL/CMOS. The hardware configuration of the DAQ board was studied and the required channels were configured for this application. The corresponding gains and input voltages were selected.

SIGNAL CONDITIONER (SCX)

Signal conditioning is necessary to step down the voltages or to shield the signals from noise and distortion. The actual voltage and current measured have to be stepped down and isolated to protect the board. The circuit diagram of the conditioning circuitry, thus developed is shown in Fig. 2. The maximum generated voltage was 125v. The voltage divider was used to obtain 1v across the 440k resistor. As the generator voltage varies from 0 to 125v, the voltage across 440k varies from 0v to 1v**.** The two 63mA fuses protect the board against any large currents due to high voltage. Without the availability of other current sensing devices, in order to get a reference voltage for the current, a variable resistor was used in series with the load. The resistor was

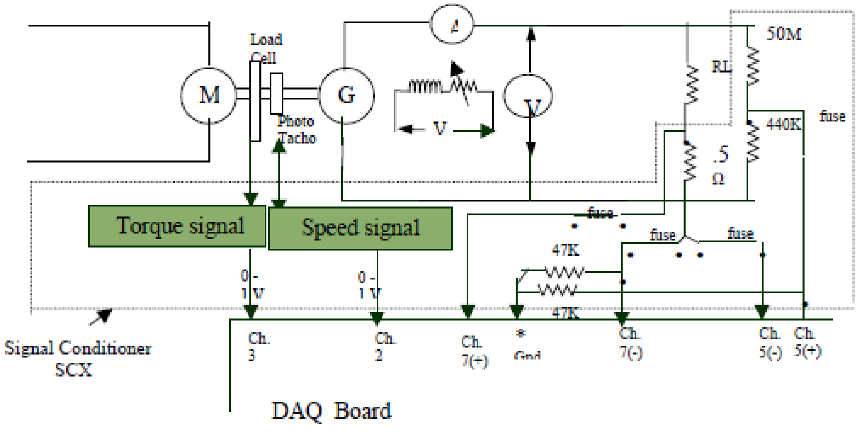


Fig. 2. Details of Signal Conditioner SCX.

adjusted to yield a 1v drop across it when maximum load current flows in the generator circuit. Two fuses protect the board against excessive currents.

Motor speed was sensed using a photo tachometer, whose output is a proportional analog signal (0 1 V) for interfacing directly to the NIDAQ board. The motor torque was measured using a Hampden load cell, with its analog output voltage used for the interface. Finally, before connecting the signal conditioning hardware to the board, the voltage at the output of the signal conditioning board was checked to make sure that they are less than one volt over the complete range of input signals.

Lab VIEW STUDENT EDITION

Lab VIEW is an abbreviation for Laboratory Virtual Instrument (VI) Engineering Workbench. It is a powerful instrumentation and analysis software system that runs on PCs.

Lab VIEW was first developed in 1983 by National Instruments, since then it has become a standard in the program development applications much like C or BASIC development systems. However, Lab VIEW departs from the sequential nature of traditional programming languages and features easy to use graphical programming environment called "G" [1014].

Lab VIEW allows design of the front panel or user interface and the block diagram or graphical code. The front panel is interactive because it simulates the panel of a physical instrument. It can contain indicators and controls such as graphs, charts, switches, numeric displays, control knobs and various other kind of controls. The indicators are program outputs whereas controls are user inputs. User input is through the keyboard and mouse while program output is displayed on the screen of the monitor.

The block diagram or graphical code constructed in the LabVIEW's graphical programming language, ‘G’ is the source code of the VI’s. It is the actual executable program. The block diagram contains lower level VI’s, built in functions, program execution control structures and constants. Wires are drawn to connect the required objects together, indicating the flow of data in a block diagram. Every object on the front panel has a corresponding terminal on the block diagram that facilitates the flow of data between the user and the program. In windows, when the LabVIEW icon is opened, the tools to build the front panel of the VI are available through the menu bar. The tool bar can be changed into run mode palette or edit mode palette by clicking on the run/edit mode icon. Some of the helpful features of the front panel and the block diagram of LabVIEW are:

- A click on the run button enables the VI to run and that on the stop button enables its stop. The run button is broken if the VI is not compiled correctly or due to an illegal connection. This prevents making serious mistakes and wasting time on debugging later on.
- The execution highlighting button causes the VI to highlight the flow of data as it passes through the diagram. When execution highlighting is on, the intermediate data values that would otherwise not appear can even be seen.
- Operating tool is used to change the values in controls, click on/off buttons and knobs.
- Positioning tool is used to select, delete, move, and resize objects etc.
- Wiring tool is used only in the block diagram. It is used to wire terminals together.
- The coloring tool enables selection of several colors to the front panel or block diagram.

FRONT PANEL

Familiarity with the different functions and menus of LabVIEW is necessary before beginning to build the front panel. As per the initial objective of the project, three parametersmotor speed, generator voltage and current had to be displayed as graphs and as numeric displays. The following procedure was used to devise the front panel and the block diagram. Using the positioning tool, with click on the right mouse button and from the pop up window selections of the choice of five different types of graphs, a Waveform Chart was chosen for the real time display of data. Once the chart was on the panel, it was duplicated to have a total of three graphs on the front panel. Numeric displays of the variables were set up from the pop up menu on the respective graphs. The data acquisition system requires two knobs to control the acquisition rate and the number of samples per second. The program execution to acquire the data from the system in operation is controlled by an on/off switch. An optional LED serves for visual indication of on/off status. The completed preliminary front panel of the project is shown in the Fig. 3.

BLOCK DIAGRAM

The block diagram is the screen where the source code for the VI is developed. In the block diagram, there is a terminal for every object created in the front panel. First the data has to be read in a continuous manner or in real time. Data input is at three different channels of the DAQ board, so a channel number is specified for each parameter. Once acquired, the data is in the range of 0 to 1v. So scaling has to be done to bring back data to their original scale before displaying the signals on the graph. Also each channel has to adopt a different scaling factor as variables’ ranges are different.

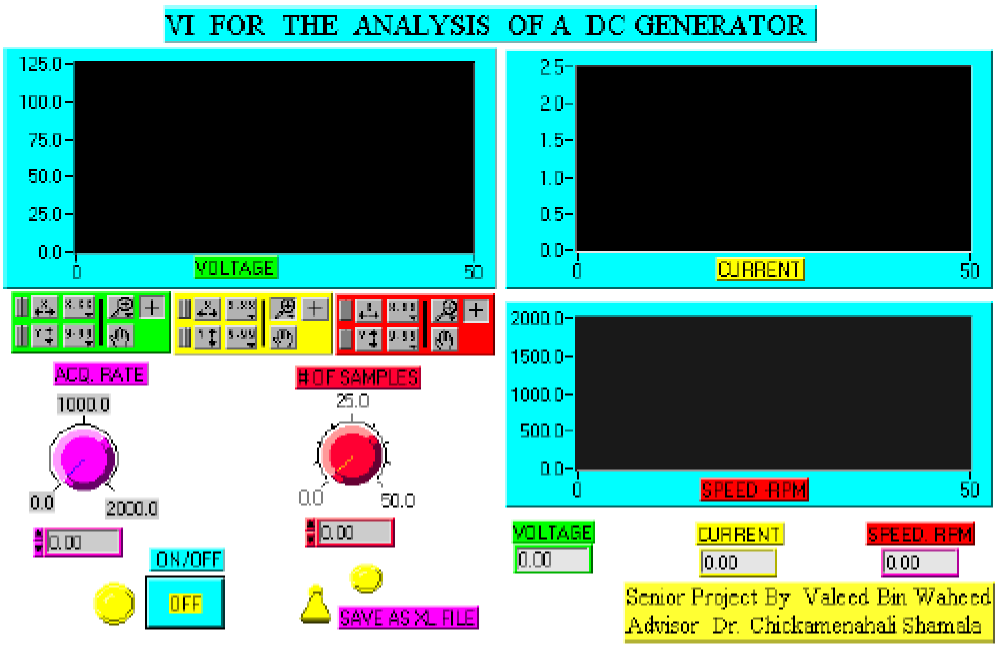


Fig. 3. Front Panel of the system.

In the block diagram environment from the pop up menu for functions, if the Data acquisition option is selected, it in turn opens another window showing several analog and digital input output VI's. There are four VI's for analog input. Two for sampling channel’s and two for acquiring waveforms. Due to entirely different variables, AI (Analog input) Acquire Waveforms(the third option) option was chosen for individual channels so as to treat them individually and condition them before displaying on the graphs. The eight terminals of the chosen VI icon have to be configured in order to attribute meaningful data acquisition and display. Three of the waveform acquire VI’s were used on the block diagram, one for each parameter to be read. In order for the VI to function, it needs the required values for each terminal. Since the differential connections at the DAQ board input were only used, it was not required to use high limit and low limit terminals on the VI. The terminals for sampling rate and number of samples are to be varied at the same time for all the three inputs. Thus these are connected to the terminals of two control knobs. The two control knobs are the ones selected on the front panel. The device and channel number entries correspond to 1 and 0 in this particular case. This readies the VI for further processing and display. The same process is repeated with the other two AI Acquire Waveform VI’s, with the only difference being the channel numbers.

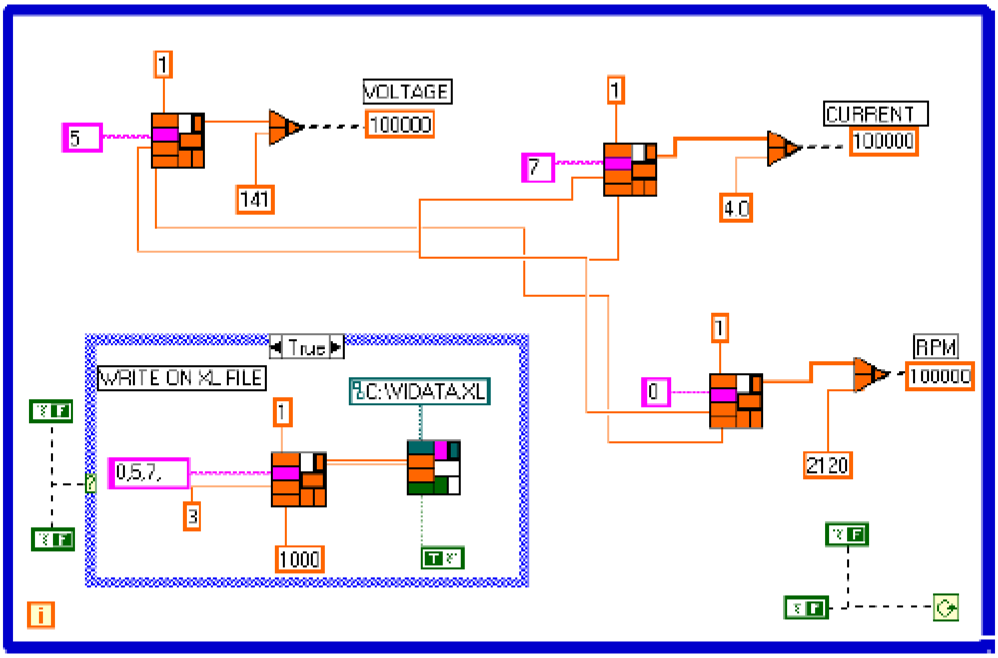


Fig. 4. System Block Diagram.

To view the displayed waveforms in actual ranges, these signals require appropriate calibration. The latter is achieved by using the arithmetic or logic function features. For the chosen channels, from the mathematics option, the multiplication function was selected to provide the required multiplication factor. This factor can be fine tuned by calibration with a physical instrument, for greater accuracy. With the system now capable of acquiring data, rescale it and display it on the graph, it however, is not set up to carry out this task in a continuous manner. In order to display the waveforms in real time, the system must function continuously. To accomplish this, once again from the pop up functions menu, selection of the Struts & Constants option allows selection of the while loop as one appropriate for this system. A while loop executes the program or diagram inside it as long as the Boolean value wired to its conditional terminal is true. It checks the conditional terminal value at the end of each iteration. If the value is true, another iteration occurs. The whole diagram is enclosed inside the loop and the conditional terminal of the loop is connected to the terminal of the Boolean switch as shown in the system block diagram in Fig. 4. This is the switch on the front panel that turns the whole system on or off and which gives control to the user. The execution highlighting function enables debugging and removing bad connections. The answers to some of the questions that arose when configuring the NIDAQ were found by browsing the National Instruments Internet site. Now the system was ready for real time data acquisition and instrumentation. The front panel and block diagram screens can be flipped back and forth by pressing < ctrl > and < f > key simultaneously or by selecting ' show panel / Show window ' option from ' windows ' in the top menu bar. This feature provides immediate information on the display configuration and the programmed block diagram.

To add to the capabilities of the system, it was desired to save the acquired data as an Excel file for further manipulation and hard copy. This is also very useful if the characteristic curves of the generator are to be drawn later. The Write to Spread Sheet File VI enabled this feature. The details of setting up the different inputs of this VI can be found in reference [5]. Ina later step, the procedure to automatically obtain the hardcopy of the data screen was achieved. As a further step in the development, the front panel and the block diagram of the Lab VIEW were both extended to incorporate the display of the motor torque variable and also the display of motor torque vs. the speed and the generator terminal voltage vs. the load current. The latter, accomplished with additional programming is illustrated in the block diagram Fig. 5. The experiment section contains the display of the front panel devised for this block diagram.

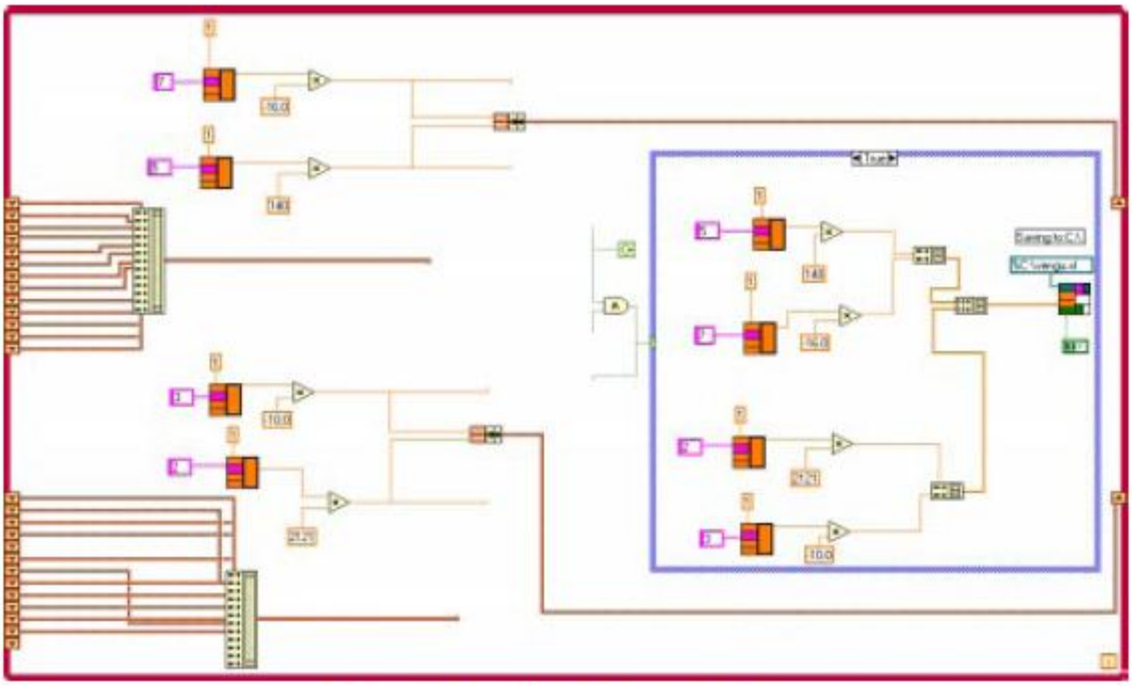


Fig. 5. System block diagram for experimental setup.

EXPERIMENTS ON MOTORGENERATOR SETUP

In order to run the system, in the Lab VIEW environment, from the file menu the VI that needs to be run is opened. With the motor generator set off, the on/off switch is enabled to turn on the VI. On the top tool bar, the run button is clicked ON. The LED next to the on/off button comes ON indicating that the system is on. The motor is turned on and brought to its normal speed of 1800 rpm. The RPM vs. time graph as well as the digital display as shown in Fig. 6. start responding to the increasing speed of the motor. Then the generator field is supplied and an observation of the generated voltage vs. time graph and the digital display is made. For the safety of the computer and board, the output of the SCX board with generator at rated voltage is checked to make sure the voltage at the board terminals is not more than 1 volt.

A resistive load is connected across the generator terminals. With increasing steps of load, the current display similarly shows a graph of the load current vs. time and also its digital display. The four graphs can be seen on one screen in real time, and change in one parameter clearly shows the effect on others.

Detailed study of the graphs can be done once the VI is turned off. This is achieved by turning off the on/off push button on the front panel. Using the tools on the tool bar, each graph can be zoomed in or zoomed out, rewound to look at previous values etc. The data saved on spreadsheet files can be viewed in Excel or any other spreadsheet environment.

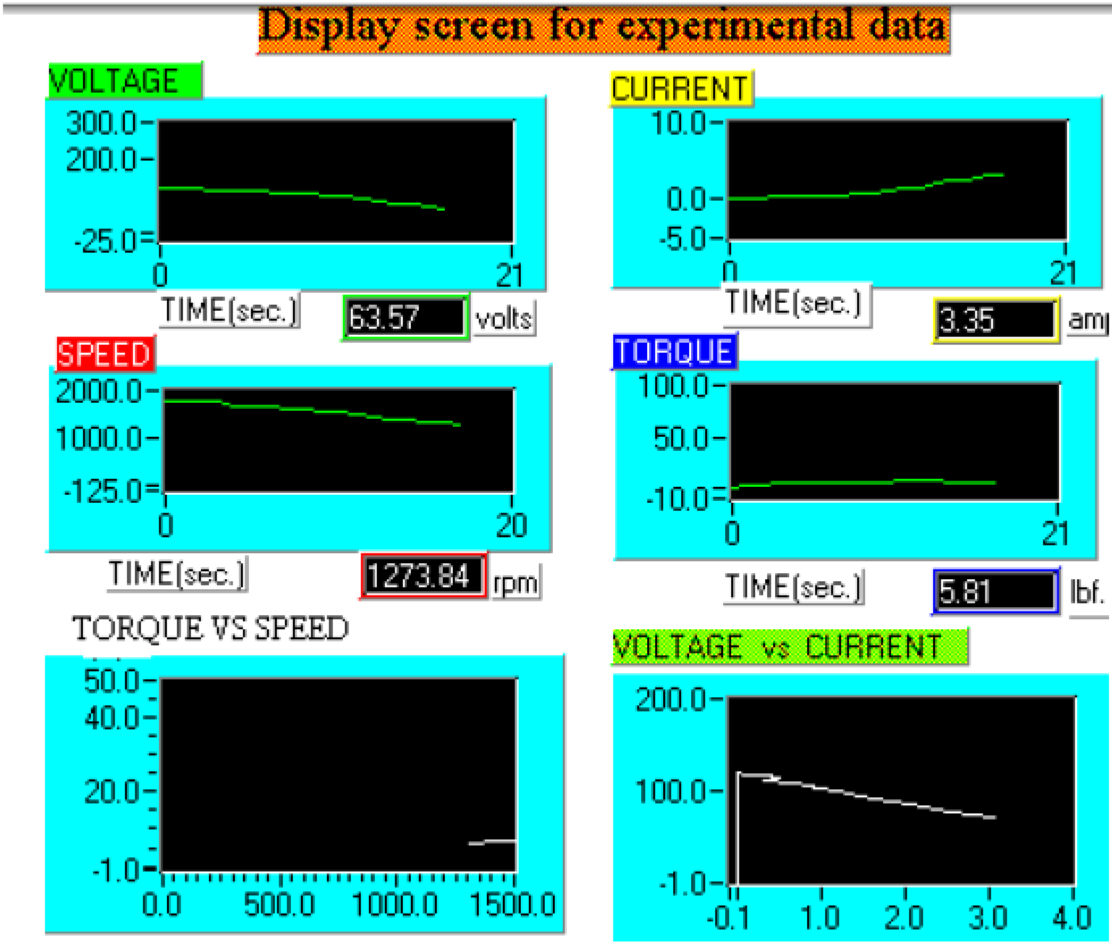


Fig. 6. Display of Experiment Waveforms from Motor-Generator Setup.

SCOPE OF FUTURE WORK

Partial results of this project served a senior undergraduate student fulfils the requirements of his EET degree in winter 1997. The project as developed was utilized as a demonstration laboratory in the Division's electric machines and drives course in the winter and fall semesters in 1997. It also has been used in three occasions as a display during Engineering Open house and Metro Detroit area public school visits, to highlight the scope of computer use in practical systems. The latter part of the development is currently being utilized to observe and analyze the performance of a machine tool system. The monitoring and display of the lathe motor torque as a function of the depth of cut of the tool is the main objective of the system. The testing of this system is awaiting the torque monitor fixture setup. The project is further to include the signals from a load monitoring system, for which separate front panel and block diagrams are being devised. This is part of a NSF funded Greenfield Coalition's Manufacturing Engineering curriculum development project for which the first author is the principal investigator. The future plans for the development include i) devising the Lab VIEW based instrumentation system as a real time visual controller rather than only as data indicator ii) to serve as an example while seeking external funding for the electric machines and drives laboratory future expansion to several stations iii) to continue to serve as a demonstration set up in other activities of the Division, iv) to serve as an a station for any measurement and control system

CONCLUSION

An instrumentation project utilizing the student edition of National Instruments LabVIEW and data acquisition tools NIDAQ is presented. Pertinent details about the methodology and the configuration of hardware and the front panel are provided. Efficient usage of graphic programming capabilities of LabVIEW are outlined. Screens showing the set up of the display of several parameters of the motor generator system are enclosed along with the actual screens of real time instrumentation. Procedures developed to save the acquired data as XL files are described. Current utilization of the project and future scope of the project are provided.

UNIT –III

COMMUNICATION NETWORKED MODULES

INTRODUCTION

These days, practically every business, no matter how small uses computers to handle various transactions and as business grows, they often need several people to input and process data simultaneously and in order to achieve this, the earlier model of a single computer serving all the organisations computational needs has been replaced by a model in which a number of separate but interconnected computers do the job and this model is known as a Computer Netw9rk. By linking individual computers over a network their productivity has been increased enormously. A most distinguishing characteristic of a general computer network is that data can enter or leave at any point and can be processed at any workstation. For example: A printer can be controlled from any word processor at any computer on the network. This is an introductory unit where, you will be learning about the basic concepts regarding Computer Networks. Here, you will learn about Networks, different types of Networks, their applications, Network topology, Network protocols, OSI Reference Mode!, TCP/IP Reference Model. We shall also examine some of the popular computer networks like Novell network, ARPANET, Internet, and ATM networks. Towards t4e end of this unit the concept of Delays in computer networks is also discussed.

OBJECTIVES

After going through this unit, you should be able to: understand the concept of computer networks differentiate between different types of computer networks understand the different application of networks compare the different network topologies signify the importance of network protocols know the importance of using networked system understand the layered organization and structuring of computer networks using OSI and TCPIIP reference model have a broad idea about some of the popular networks like Novell network, ARPANET, INTERNET, ATM etc., and understand the concept of delays.

3.0 MAIN CONTENT

What is a computer network?

Figure 1: A computer networked environment

A Computer network consists of two or more autonomous computers that are linked (connected) together in order to: Share resources (files, printers, modems, fax machines). Share Application software like MS Office. Allow Electronic communication. Increase productivity (makes it easier to share data amongstusers). *Figure 1* shows people working in a networked environment. The Computers on a network may be linked through Cables, telephones lines, radio waves, satellites etc. A Computer network includes, the network operating system in the client and server machines, the cables, which connect different computers and all supporting hardware ill between such as bridges, routers and switches. In wireless systems, antennas and towers are also part of the network. Computer networks are generally classified according to their structure and the area they are localised in as:

Local Area Network (LAN): The network that spans a relatively small area that is, in the single building or campus is known as LAN.

Metropolitan Area Network (MAN): The type of computer network that is, designed for a city or town is known as MAN.

Wide Area Network (WAN): A network that covers a large geographical area and covers different cities, states and sometimes even countries, is known as WAN. The additional characteristics that are also used to categorise different types of networks are:

Topology: Topology is the graphical arrangement of computer systems in a network. Common topologies include a bus, star, ring, and mesh.

Protocol: The protocol defines a common set of rules which areused by computers on the network that communicate between hardware and software entities. One of the most popular protocols for LANs is the Ethernet. Another popular LAN protocol for PCs is the tokenring network.

Architecture: Networks can be broadly classified as using either a peertopeer or client/server architecture.

3.2 Network Goals and Motivations

Before designing a computer network we should see that the designed network fulfils the basic goals. We have seen that a computer network should satisfy a broad range of purposes and should meet various requirements. One of the main goals of a computer network is to enable its users to share resources, to provide low cost facilities and easy addition of new processing services. The computer network thus, createsa global environment for its users and computers. Some of the basic goals that a Computer network should satisfy are: Cost reduction by sharing hardware and software resources. Provide high reliability by having multiple sources of supply. Provide an efficient means of transport for large volumes of data among various locations (High throughput).

Provide inter process communication among users and processors. Reduction driving data transport. Increase productivity by making it easier to share data amongst users. Repairs, upgrades, expansions, and changes to the network should be performed with minimal impact on the majority of network users. Standards and protocols should be supported to allow many types of equipment from different vendors to share the network (Interpretability). Provide centralized/distributed management and allocation of network resources like host processors, transmission facilities etc.

3.3 Classification of Networks

Depending on the transmission technology i.e., whether the network contains switching elements or not, we have two types of networks: Broadcast networks. Point to point or Switched networks.

3.3.1 Broadcast Networks

Broadcast networks have a single communication channel that is shared by all the machines on the network. In this type of network, short messages sent by any machine are received by all the machines on the network. The packet contains an address field, which specifies for whom the packet is intended. All the machines, upon receiving a packet check for the address field, if the packet is intended for itself, it processes it and if not the packet is just ignored. Using Broadcast networks, we can generally address a packet to all destinations (machines) by using a special code in the address field. Such packets are received and processed by all machines on the network. This mode of operation is known as “Broadcasting”. Some Broadcast networks also support transmission to a subset of machines and this is known as “Multicasting”. One possible way to achieve Multicasting is to reserve one bit to indicate multicasting and the remaining (nl) address bits contain group number. Each machine can subscribe to any or all of the groups. Broadcast networks are easily configured for geographically localised networks. Broadcast networks may be Static or dynamic, depending on how the channel is allocated. In Static allocation, time is divided into discrete intervals and using round robin method, each machine is allowed to broadcast only when its

LAN (Local Area Network)

Local Area Network is a computer network that spans over a relatively small area. Most LANs are confined to a single building or group of buildings within a campus. However, one LAN can be connected to other LANs over any distance via telephone lines and radio waves. A system of LANs connected in this way is called a wide area network (WAN). Most LANs connect workstations and personal computers. Each node(individual computer) in a LAN has its own CPU with which it executes programs, but it is also able to access data and devices anywhere on the LAN. This means that many users can share data as well as expensive devices, such as laser printers, fax machines etc. Users can also use the LAN to communicate with each other, by sending email or engaging in chat sessions. There are many different types of LANs, Ethernets being the most common for PCs. The following characteristics differentiate one LAN from another:

Topology

The geometric arrangement of devices on the network. For example, devices can be arranged in a ring or in a straight line.

Protocols

The rules and encoding specifications for sending data. The protocols also determine whether the network uses peer to peer or client/server architecture.

Media

Devices can be connected by twisted pair wire, coaxial cables, or fiber optic cables. Some networks communicate via radio waves hence, do not use any connecting media. LANs are capable of transmitting data at very fast rates, much faster than data can be transmitted over a telephone line; but the distances are limited, and there is also a limit on the number of computers that can be attached to a single LAN.

The typical characteristics of a LAN are: Confined to small areas i.e., it connects several devices over a distance of5to 10km. High speed. Most inexpensive equipment .Low error rates. Data and hardware sharing between users owned by the user. Operates at speeds ranging from 10Mbps to 100Mbps. Nowadays 1000 Mbps are available.

3.3.2 Point toPoint or Switched Networks

Point to point or switched, networks are those in which there are many connections between individual pairs of machines. In these networks, when a packet travels from source to destination it may have to first visit one or more intermediate machines. Routing algorithms play an important role in Point to point or Switched networks because often multiple routes of different lengths are available An example of switched network is the international dialup telephone system.

The different types of Point to point or Switched networks are:
Circuit Switched Networks.
Packet Switched Networks.
In Switched network, the temporary connection is established from one point to another for either the duration of the session (circuit switching) or for the transmission of one or more packets of data (packet switching).

Circuit Switched Networks

Circuit Switched networks use a networking technology that provides a temporary, but dedicated connection between two stations no matter how many switching devices are used in the data transfer route. Circuit switching was originally developed for the analog based telephone system in order to guarantee steady and consistent service for two people engaged in a phone conversation. Analog circuit switching has given way to digital circuit switching, and the digital counterpart still maintains the connection until broken (one side hangs up). This means bandwidth is continuously reserved and “silence is transmitted” just the same as digital audio in voice conversation.

Packet Switched Networks

Packet switched Networks use a networking technology that breaks up a message into smaller packets for transmission and switches them to their required destination. Unlike circuit switching, which requires a constant point to point circuit to be established, each packet in a packet switched network contains a destination address. Thus, all packets in a single message do not have to travel the same path. They can be dynamically routed over the network as lines become available or unavailable. The destination computer reassembles the packets back into their proper sequence. Packet switching efficiently handles messages of different lengths and priorities. By accounting for packets sent, a public network can charge customers for only the data they transmit. Packet switching has been widely used for data, but not for real time voice and video. However, this is beginning to change. IP and ATM technologies are expected to enable packet switching to be used for everything was X.25, which was defined when all circuits were digited and susceptible to noise. Subsequent technologies, such as frame relay and SMDS were designed for today's almost error free digital lines. ATM uses a cell switching technology that provides the bandwidth sharing efficiency of packet switching with the guaranteed bandwidth of circuit switching. Highe level protocols, such as TCPIIP, IPX/SPX and NetBIOS, are also packet based and are designed to ride over packet switched topologies. Public packet switching networks may provide value added services, such as protocol conversion and electronic mail.

Bus Topology

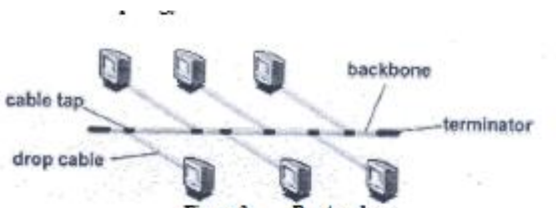


Figure 2: Bus topology

In Bus topology, all devices are connected to a central cable, called the bus or backbone. The bus topology connects workstations using a single cable. Each workstation is connected to the next workstation in a point to point fashion. All workstations connect to the same cable. *Figure 2* shows computers connected using Bus Topology. In this type of topology, if one workstation goes faulty all workstations may be affected as all workstations share the same cable for the sending and receiving of information. The cabling cost of bus systems is the least of all the different topologies. Each end of the cable is terminated using a special terminator. The common implementation of this topology is Ethernet. Here, message transmitted by one workstation is heard by all the other workstations.

Advantages of Bus Topology

Installation is easy and cheap when compared to other topologies Connections are simple and this topology is easy to use. Less cabling is required.

Disadvantages of Bus Topology

Used only in comparatively small networks. As all computers share the same bus, the performance of the network deteriorates when we increase the number of computers beyond a certain limit. Fault identification is difficult. A single fault in the cable stops all transmission.

3.4.2 Star Topology

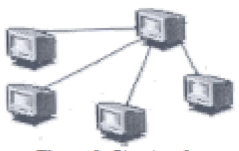


Figure 3: Star topology

Start topology uses a central hub through which, all components are connected. In a Star topology, the central hub is the host computer, and at the end of each connection is a terminal as shown in *Figure 3.*Nodes communicate across the network by passing data through the hub. A star network uses a significant amount of cable as each terminal is wired back to the central hub, even if two terminals are side by side but several hundred meters away from the host. The central hub makes all routing decisions, and all other workstations can be simple. An advantage of the star topology is, that failure, in one of the terminals does not affect any other terminal; however, failure of the central hub affects all terminals. This type of topology is frequently used to connect terminals to a large timesharing host computer.

Advantages of Star Topology
Installation and configuration of network is easy. Less expensive when compared to mesh topology. Faults in the network can be easily traced. Expansion and modification of star network is easy. Single computer failure does not affect the network. Supports multiple cable types like shielded twisted pair cable, unshielded twisted pair cable, ordinary telephone cable etc.
Disadvantages of Star Topology
Failure in the central hub brings the entire network to a halt. More cabling is required in comparison to tree or bus topology because each node is connected to the central hub.

3.4.3 Ring Topology

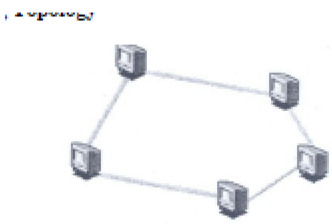


Figure 4: Ring Topology

In Ring Topology all devices are connected to one another in the shape of a closed loop, so that each device is connected directly to two other devices, one on either side of it, i.e., the ring topology connects workstations in a closed loop, which is depicted in Figure 4 Each terminal is connected to two other terminals (the next and the previous), with the last terminal being connected to the first. Data is transmitted around the ring in one direction only; each station passing on the data to the next station till it reaches its destination.

Information travels around the ring from one workstation to the next. Each packet of data sent on the ring is prefixed by the address of the station to which it is being sent. When a packet of data arrives, the workstation checks to see if the packet address is the same as its own, ifit is, it grabs the data in the packet. If the packet does not belong to it, it sends the packet to the next workstation in the ring. Faulty workstations can be isolated from the ring. When the workstation is powered on, it connects itself to the ring. When power is off, it disconnects itself from the ring and allows the information to bypass theworkstation. The common implementation of this topology is token ring. A break in the ring causes the entire network to fail. Individual workstations can be isolated from the ring.
Advantages of Ring Topology
Easy to install and modify the network. Fault isolation is simplified. Unlike Bus topology, there is no signal loss in Ring topology because the tokens are data packets that are regenerated at each node.
Disadvantages of Ring Topology
Adding or removing computers disrupts the entire network. A break in the ring can stop the transmission in the entire network. Finding fault is difficult. Expensive when compared to other topologies.

3.4.4 Tree Topology

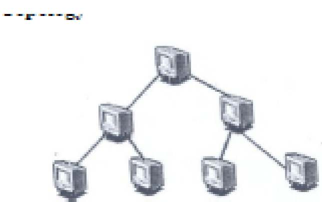


Figure 5: Tree Topology

Tree topology is a LAN topology in which only one route exists between any two nodes on the network. The pattern of connection resembles a tree in which all branches spring from one root. Figure 5 shows computers connected using Tree Topology. Tree topology is a hybrid topology, it is similar to the star topology but the nodes are connected to the secondary hub, which in turn is connected to the central hub. In this topology groups of star configured networks are connected to a linear bus backbone.

Advantages of Tree Topology

Installation and configuration of network is easy. Less expensive when compared to mesh topology. Faults in the network can be detected traced. The addition of the secondary hub allows more devices to be attached to the central hub. Supports multiple cable types like shielded twisted pair cable, unshielded twisted pair cable, ordinary telephone cable etc.

Disadvantages of Tree Topology

Failure in the central hub brings the entire network to a halt. More cabling is required when compared to bus topology because each node is connected to the central hub.

3.4.5 Mesh Topology

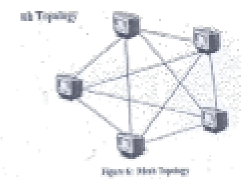


Figure 6: Mesh Topology

Devices are connected with many redundant interconnections between network nodes. In a well connected topology, every node has a connection to every other node in the network. The cable requirements are high, but there are redundant paths built in. Failure in one of the computers does not cause the network to break down, as they have alternative paths to other computers. Mesh topologies are used in critical connection of host computers (typically telephone exchanges). Alternate paths allow each computer to balance the load to other computer systems in the network by using more than one of the connection paths available. A fully connected mesh network therefore has n (n1)/2 physical channels to link n devices. To accommodate these, every device on the network must have (nil) input/output ports.

Advantages of Mesh Topology

Use of dedicated links eliminates traffic problems. network. Point to point link makes fault isolation easy. It is robust. Privacy between computers is maintained as messages travel along dedicated path.

Disadvantages of Mesh Topology

The amount of cabling required is high. A large number of I/O (input/output) ports are required.

3.4.6 Cellular Topology

Figure 7: Cellular Topology

Cellular topology, divides the area being serviced into cells. In wireless media each point transmits in a certain geographical area called a cell, each cell represents a portion of the total network area. Figure 7 shows computers using Cellular Topology. Devices that are present within the cell, communicate through a central hub. Hubs in different cells are interconnected and hubs are responsible for routing data across the network. They provide a complete network infrastructure.

Cellular topology is applicable only in case of wireless media that does not require cable connection.

Advantages of Cellular Topology

If the hubs maintain a point to point link with devices, trouble shooting is easy. Hub to hub fault tracking is more complicated, but allows simple fault isolation.

Disadvantages of Cellular Topology

When a hub fails, all devices serviced by the hub lose service (are affected).

OSI References Model

Figure 9: Layers or OSI reference model

The Open System Interconnection (OSI) model is a set of protocols that attempt to define and standardize the data communications process; we can say that it is a concept that describes how data communications should take place. The OSI model was set by the International Standards Organization(ISO) in 1984, and it is now considered the primary architectural model for inter computer communications. The OSI model has the support of most major computer and network vendors, many large customers, and most governments in different countries.

The Open Systems Interconnection (OS1) reference model describes how information from a software application in one computer moves through a network medium to a software application in another computer. The OSI reference model is a conceptual model composed of seven layers as shown in Figure 9 each specifying particular network functions and into these layers are fitted the protocol standards developed by the ISO and other standards bodies. The OSI model divides the tasks involved with moving information between networked computers into seven smaller, more manageable task groups. A task or group of tasks is then assigned to each of the seven OSI layers. Each layer is reasonably self contained so that the tasks assigned to each layer can be implemented independently. This enables the solutions offered by one layer to be updated without affecting the other layers. The OSI model is modular. Each successive layer of the OSI model works with the one above and below it. Although, each layer of the OSI model provides its own set of functions, it is possible to group the layers into two distinct categories. The first four layers i.e., physical, data link, network, and transport layer provide the end to end services necessary for the transfer of data between two systems. These layers pr9.f1de the protocols associated with the communications network used to link two computers together. Together, these are communication oriented. The top three layers i.e., the application, presentation, and session layers provide the application services required for the exchange of information. That is, they allow two applications, each running on a different node of the network to interact with each other through the services provided by their respective operating systems. Together, these are data processing oriented.

The following are the seven layers of the Open System Interconnection (OSI) reference model:

Layer 7 Application layer
Layer 6 Presentation layer
Layer 5 Session layer
Layer 4 Transport layer
Layer 3 Network layer

Layer 2 Data Link layer
Layer I Physical layer

Application Layer (Layer 7)

The Application layer is probably the most easily misunderstood layer of the model. This top layer defines the language and syntax that programs use to communicate with other programs. The application layer represents the purpose of communicating in the first place. For example, a program in a client workstation uses commands to request data from a program in the server. Common functions at this layer are opening, closing, reading and writing files, transferring files and email messages, executing remote jobs and obtaining directory information about network resources etc.

Presentation Layer (Layer 6)

The Presentation layer performs code conversion and data reformatting (syntax translation). It is the translator of the network; it makes sure the data is in the correct form for the receiving application. When data are transmitted between different types of computer systems, the presentation layer negotiates and manages the way data are represented and encoded. For example, it provides a common denominator between ASCII and EBCDIC machines as well as between different floating point and binary formats. Sun's XDR and OSI's ASN.1 are two protocols used for this purpose. This layer is also used for encryption and decryption. It also provides security features through encryption and decryption.

Session Layer (Layer 5)

The Session layer decides when to turn communication on and off between two computers. It provides the mechanism that controls the data exchange process and coordinates the interaction (communication)between them in ail orderly manner. It sets up and clears communication channels between two communicating components. It determines one way or two way communications and manages the dialogue between both parties; for example, making sure that the previous request has been fulfilled before the next one is sent. It also marks significant parts of the transmitted data with checkpoints to allow for fast recovery in the event of a connection failure.

Transport Layer (Layer 4)

The transport layer is responsible for overall end to nd validity and integrity of the transmission i.e., it ensures that data is successfully sent and received between two computers. The lower data link layer (layer 2) is only responsible for delivering packets from one node to another. Thus, if a packet gets lost in a router somewhere in the enterprise Internet, the transport layer will detect that. It ensures that if a 12MB file is sent, the full 12MB is received. If data is sent incorrectly, this layer has the responsibility of asking for retransmission of the data. Specifically, it provides a network in dependent, reliable message independent, reliable message interchange service to the top three application oriented layers. This layer acts as an interface between the bottom and top three layers. By providing the session layer (layer 5) with a reliable message transfer service, it hides the detailed operation of the underlying network from the session layer.

Network Layer (Layer 3)

The network layer establishes the route between the sending and receiving stations. The unit of data at the network layer is called a packet. It provides network routing and flow and congestion functions across computer network interface.

It makes a decision as to where to route the packet based on information and calculations from other routers, or according to static entries in the routing table. It examines network addresses in the data instead of physical addresses seen in the Data Link layer. The Network layer establishes, maintains, and terminates logical and/or physical connections. The network layer is responsible for translating logical addresses, or names, into physical addresses The main device found at the Network layer is a router

Data link Layer (Layer 2)

The data link layer groups the bits that we see on the Physical layer into Frames. It is primarily responsible for error free delivery of data on a hop. The Data link layer is split into two sub ayers i.e., the Logical Link Control (LLC) and Media Access Control (MAC). The Data Link layer handles the physical transfer, framing (the assembly of data into a single unit or block), flow control and error Control functions (and retransmission in the event of an error) over a single transmission link; it is responsible for getting the data packaged and onto the network cable. The data link layer provides the network layer (layer 3) reliable information transfer capabilities.

Physical Layer (Layer 1)

The data units on this layer are called bits. This layer defines the mechanical and electrical definition of the network medium (cable) and network hardware. This includes how data is impressed onto the cable and retrieved from it. The physical layer is responsible for passing bits onto and receiving them from the connecting medium. This layer gives the data link layer (layer 2) its ability to transport a stream of serial data bits between two communicating systems; it conveys the bits that moves along the cable. It is responsible for ensuring that the raw bits get from one place to another, no matter what shape they are in, and deals with the mechanical and electrical characteristics of the cable. This layer has no understanding 01 the meaning of the bits, but deals with the electrical and mechanical characteristics of the signals and signaling methods. The main network device found the Physical layer is a repeater. The purpose of a repeater (as the name suggests) is simply to receive the digital signal, reform it, and retransmit the signal. This has the effect of increasing the maximum length of a network, which would not be possible due to signal deterioration if, a repeater were not available. The repeater, simply regenerates cleaner digital signal so it doesn’t have to understand anything about the information it is transmitting, and processing on the repeater is nonexistent. An example of the Physical layer is RS232**.**Each layer, with the exception of the physical layer, adds information to the data as it travels from the Application layer down to the physical layer. This extra information is called a header. The physical layer does not append a header to information because it is concerned with sending and receiving information on the individual bit level. We see that the data for each layer consists of the header and data of the next higher layer. Because the data format is different at each layer, different terms are commonly used to name the data package at each level. Figure 10 summarizes these terms layer by layer.

LAYER	DATA PACKAGE NAME
Application layer	Application Protocol Data Unit
Presentation layer	Presentation Protocol Data Unit. Generic name is Protocol Data Unit (PDU). For example, at session layer it is S-PDU.
Session layer	Session Protocol Data Unit
Transport layer	Segment
Network layer	Datagram, packet
Data link layer	Frame
Physical layer	Bit

Figure 10: Data package names in the OSI reference model

OSI Protocols

The OSI model provides a conceptual framework for communication between computers, but the model itself is not a method of communication. Actual communication is made possible by using communication protocols. In the context of data networking, a protocol is a formal set of rules and conventions that governs how computers exchange information over a network medium. A protocol implements the functions of one or more of the OSI layers. A wide variety of communication protocols exist, but all tend to fall into one of the following groups: LAN protocols, WAN protocols, network protocols, and routing protocols. LAN protocols operate at the network and data link layers of the OSI model and define communication over the various LAN media. WAN protocols operate at the lowest three layers of the OSI model and define communication over the various wide area media.

Routing protocols are network layer protocols that are responsible for path determination and traffic switching. Finally, network protocols are the various upper layer protocols that exist in a given protocol suite.

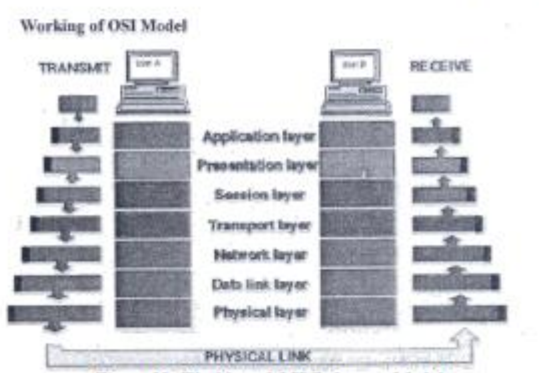


Figure 11: Working of OSI Reference Model

Information being transferred from a software application in one computer system to software application in another must pass through each of the OSI layers. Each layer communicates with three other OSI layers i.e., the layer directly above it, the layer directly below it, and its peer layer ill other networked systems. If, for example, in *Figure 10*, a software application in System A has information to transmit to a software application in System B, the application program in System Awill pass its information to the application layer (Layer 7) of System A. The application layer then passes the information to the presentation layer (Layer 6); the presentation layer reformats the data if required such that B can understand it. The formatted data is passed to the session layer (Layer 5), which in turn requests for connection establishment between session layers of A and B, it then passes the data to the transport layer. The transport layer breaks the data into smaller units called segments and sends them to the Network layer. The Network layer selects the route for transmission and if, required breaks the data packets further. These data packets are then sent to the Data link layer that is responsible for encapsulating the data packets into data frames. The Data link layer also adds source and destination addresses with error checks to each frame, for the hop. The data frames are finally transmitted to the physical layer. In the physical layer, the data is in the fond of a stream of bits and this is placed on the physical network medium and is sent across the medium to System B. B receives the bits at its physical layer and passes them on to the Data link layer, which verifies that no error has occurred. The Network layer ensures that the route selected for transmission is reliable, and passes the data to the Transport layer. The function of the Transport layer is to reassemble the data packets into the file being transferred and then, pass it on to the session layer. The session layer confirms that the transfer is complete, and if so, the session is terminated. The data is then passed to the Presentation layer, which major may not reformat it to suit the environment of B and sends it to the Application layer. Finally the Application layer of System B passes the information to the recipient Application program to complete the communication process.

Interaction between different layers of OSI model

A given layer in the OSI layers generally communicates with three other OSI layers: the layer directly above it, the layer directly below it, and its Peer layer in another networked computer system. The data link layer in System A, for example, communicates with the network layer of System A, the physical layer of System A, and the data link layer in System B.

Services provided by OSI layers

One OSI layer communicates with another layer to make use of the services provided by the second layer. The services provided by adjacent layers help a given OSI layer communicate with its peer layer in other computer systems. Three basic elements are involved inlayer services: the service user, the service provider, and the service access point (SAP).

In this context, the service user is the OSI layer .that requests services from an adjacent OSI layer. The service provider is the OSI layer that provides services to service users. OSI layers can provide services to multiple service users. The SAP is a conceptual location at which one OSI layer can request the services of another OSI layer.

OSI Model Layers and Information Exchange

The seven OSI layers use various forms of control information to communicate with their peer layers in other computer systems. This control information consists of specific requests and instructions that are exchanged between peer OSI layers. Control information typically takes one of two forms: headers and trailers. Headers are pretended to data that has been passed down from upper layers. Trailers are appended to data that has been passed down from upper layers. Headers, trailers, and data are relative concepts, depending on the layer that analyses the information unit. At the network layer, an information unit, for example, consists of a Layer 3 header and data. At the data link layer, however, all the information passed down by the network layer (the Layer 3 header and the data) is treated as data. In other words, the data portion of an information unit at a given OSI layer potentially can contain headers, trailers, and data from all the higher layers. This is known as encapsulation.

3.6.2 TCP/IP Reference Model

APPLCIATION LAYER
TRANSPORT LAYER
NETWORK LAYER
LINK/PHYSICAL LAYER

Figure 12: Layers of TCP/IP reference model

TCP/IP stands for Transmission Control Protocol I Internet Protocol. It is a protocol suite used by most communications software. TCP/IP is a robust and proven technology that was first tested in the early I 980s on ARPA Net, the U.S. military's Advanced Research Projects Agency network, and the world's first packet switched network. TCP/IP was designed as an open protocol that would enable all types of computers to transmit data to each other via a common communications language. TCP/IP is a layered protocol similar to the ones used in all the other major networking architectures, including IBM's SNA, Windows' NetBIOS, Apple's AppleTalk, Novell's NetWare and Digital's Decent. The different layers of the TCP/IP reference model are shown in *Figure13*. Layering means that after an application initiates the communications, the message (data) to be transmitted is passed through a number of stages or layers until it actually moves out onto the wire. The data are packaged with a different header at each layer.

At the receiving end, the corresponding programs at each protocol layer unpack the data, moving it "back up the stack" to the receiving application. TCP/IP is composed of two major parts: TCP (Transmission Control Protocol) at the transport layer and IP (Internet Protocol) at the network layer. TCP is a connection oriented protocol that passes its data to IP, which is a connectionless one. TCP sets up a connection at both ends and guarantees reliable delivery of the full message sent. TCP tests for errors and requests retransmission if necessary, because IP does not An alternative protocol to TCP within the TCP/IP suite is UDP (User Datagram Protocol), which does not guarantee delivery. Like IP, it is also connectionless, but very useful for real time voice and video, where it doesn't matter if a few packets get lost.

Layers of TCP/IP reference model
Application Layer (Layer 4)

The top layer of the protocol stack is the application layer. It refers to the programs that initiate communication in the first place. TCP/IP includes several application layer protocols for mail, file transfer, remote access, authentication and name resolution. These protocols are embodied in programs that operate at the top layer just as any custom made or packaged client/server application would. There are many Application Layer protocols and new protocols are always being developed. The most widely known Application Layer protocols are those used for the exchange of user information, some of them are:

The HyperText Transfer Protocol (HTTP) is used to transfer files that make up the Web pages of the World Wide Web.

The File Transfer Protocol (FTP) is used for interactive file transfer.

The Simple Mail Transfer Protocol (SMTP) is used for the transfer of mail messages and attachments.

Telnet, is a terminal emulation protocol, and, is used for remote login to network hosts.Other Application Layer protocols that help in the management of TCP/IP networks are:

The Domain Name System (DNS), which, is used to resolve a host name to an IP address.

The Simple Network Management Protocol (SNMP)

which is used between network management consoles and network devices (routers, bridges, and intelligent hubs) to collect and exchange network management information. Examples of Application Layer interfaces for TCP/IP applications are Windows Sockets and NetBIOS. Windows Sockets provides a standard application programming interface (API) under the Microsoft Windows operating system. NetBIOS is an industry standard interface for accessing protocol services such as sessions, data grams, and name resolution.

Transport Layer (Layer 3)

The Transport Layer (also known as the HosttoHost Transport Layer) is responsible for providing the Application Layer with session and datagram communication services. TCP/IP does not contain Presentation and Session layers, the services are performed if required, but they are not part of the formal TCP/IP stack. For example, Layer 6 (Presentation Layer) is where dataconversion (ASCII to EBCDIC, floating point to binary, etc,) and encryption/decryption is performed, Layer 5 is the Session Layer, which is performed in layer 4 in TCP/IP, Thus, we jump from layer 7 of OS I down to layer 4 of TCP/IP.

From Application to Transport Layer, the application delivers its data to the communications system by passing a Stream of data bytes to the transport layer along with the socket of the destination machine. The core protocols of the Transport Layer are TCP and the User Datagram Protocol (UDP).

TCP: TCP provides a one to one, connection oriented, reliable communications service. TCP is responsible for the establishment of a TCP connection, the sequencing and acknowledgment of packets sent, and the recovery of packets lost during transmission.

UDP: UDP provides a one to one or one to many, connectionless, unreliable communications service. UDP is used when the amount of data to be transferred is small (such as the data that would fit into a single packet), when the overhead of establishing a TCP connection is not desired, or when the applications or upper layer protocols provide reliable delivery. The transport Layer encompasses the responsibilities of the OSI Transport Layer and some of the responsibilities of the OSI Session Layer.

Internet Layer (Layer 2)

The internet layer handles the transfer of information across multiple networks through the use of gateways and routers. The internet layer corresponds to the part of the ORI network layer that is concerned with the transfer of packets between machines that are connected to different networks. It deals with the routing of packets across these networks as well as with the control of congestion. A key aspect of the internet layer is the definition of globally unique addresses for machines that are attached to the Internet.

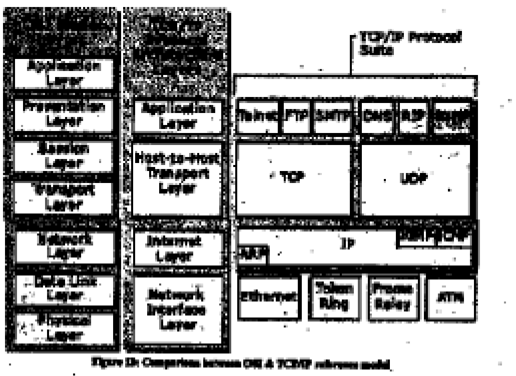
The Internet layer provides a single service namely, best effort connectionless packet transfer. IP packets are exchanged between routers without a connection setup; the packets are ‘touted independently and so they may traverse different paths. For this reason, IP packets are also called data rams. The connectionless approach makes the system robust; that is, if failures occur in the network, the packets are routed around the points of failure; hence, there is no need to set up connections, The gateways that interconnect the intermediate networks may discard packets when congestion occurs. The responsibility for recovery from these losses is I passed on to the Transport Layer, The core protocols of the Internet Layer are IP. ARP,ICMP, and IGMP.

The Internet Protocol (IP) is a routable protocol responsible for IP addressing and the fragmentation and reassembly of packets.
The Address Resolution Protocol (ARP) is responsible for the resolution of the Internet Layer address to the Network Interface Layer address, such as a hardware address.
The Internet Control Message Protocol (ICMP) is responsible for providing diagnostic functions and reporting errors or conditions regarding the delivery of IP packets.
The Internet Group Management Protocol (IGMP) is responsible for the management of IP multicast groups. The Internet Layer is analogous to the Network layer of the OSI model.

Link/Physical Layer (Layer I)

The Link/Physical Layer (also called the Network Access Layer) is responsible for placing TCP/IP packets on the network medium and receiving TCP/IP packets of the network medium. TCP/IP was designed
to be independent of the network access method, frame format, and types. This includes LAN technologies such as Ethernet or Token Ring and WAN technologies such as X.25 or Frame Relay. Independence
from any specific network technology gives TCP/IP the ability to be adapted to new technologies such as Asynchronous Transfer Mode (ATM). The Network Interface Layer encompasses the Data Link and Physical layers of the OSI Model. Note, that the Internet Layer does not take advantage of sequencing and acknowledgement services that may be present in the Data Link Layer. An unreliable Network Interface Layer is assumed, and reliable communications through session establishment and the sequencing and acknowledgement of packets is the responsibility of the Transport Layer.

Comparison between OSI and TCP/IP reference model



Both OSI and TCP/IP reference models are based on the concept of a stack of protocols. The functionality of the layers is almost similar. In both models the layers are there to provide an end o end network independent transport service to processes wishing to communicate with each other. The Two models have many differences. An obvious difference between the two models is the number of layers: the OSI model has seven layers and the TCP/IP has .four layers. Both have (inter) network, transport, and application layers, but the other layers are different. OSI uses strict layering, resulting in vertical layers whereas TCP/IP uses loose layering resulting in horizontal layers. The OSI model supports both connectionless and connection oriented communication in the network layer, but only connection oriented communication at the transport layer. The TCP/IP model has only one mode in network layer (connectionless), but supports both modes in the transport layer. With the TCP/IP model, replacing IP by a substantially different protocol would be virtually impossible, thus, defeating one of the main purposes of having layered protocols in the first place. The OSI reference model was devised before the OSI protocols were designed. The OSI model was not biased toward one particular set of protocols, which made it quite general. The drawback of this ordering is that the designers did not have much experience with the subject, and did not have a good idea of the type of functionality to put in a layer.

With TCP/IP the reverse was true: the protocols carne first and the model was really just a description of the existing protocols. There was no problem with the protocols fitting the model. The only drawback was that the model did not fit any other protocol stacks.

Figure 14 summarises the basic differences between OSI and TCP/IP reference models.

OSI MODEL	TCP/IP MODEL
Contains 7 Layers	Contains 4 Layers
Uses Strict Layering resulting in vertical layers.	Uses Loose Layering resulting in horizontal layers.
Supports both connectionless & connection-oriented communication in the Network layer, but only connection-oriented communication in Transport Layer	Supports only connectionless communication in the Network layer, but both connectionless & connection-oriented communication in Transport Layer
It distinguishes between Service, Interface and Protocol.	Does not clearly distinguish between Service, Interface and Protocol.
Protocols are better hidden and can be replaced relatively easily as technology changes (No transparency)	Protocols are not hidden and thus cannot be replaced easily. (Transparency) Replacing IP by a substantially different protocol would be virtually impossible
OSI reference model was devised before the corresponding protocols were designed.	The protocols came first and the model was a description of the existing protocols

Figure 14: Difference between OSI and TCP/IP reference model

Some of the drawbacks of OS I reference model are:

All layers are not roughly, of equal size and complexity. In practice, the session layer and presentation layer are absent from many existing architectures. Some functions like addressing, flow control, retransmission are duplicated at each layer, resulting in deteriorated performance. The initial specification of the OSI model ignored the connectionless model, thus, leaving much of the LANs behind.

Some of the drawbacks of TCP/IP model are:

TCP/IP model does not clearly distinguish between the concepts of service, interface, and protocol. TCP/IP model is not a general model and therefore it cannot be used to describe any protocol other than TCP/IP.TCP/IP model does not distinguish or even mention the Physicalor the Data link layer. A proper model should include both these layers as separate.

UNIT IV

REAL TIME CONTROL

Current mode data transmission is the preferred technique in many environments, particularly in industrial applications. Most systems employ the familiar 2wire, 4–20mA current loop, in which a single twisted pair cable supplies power to the module and carries the output signal as well.

The 3wire interface is less common but allows the delivery of more power to the module electronics. A 2wire system provides only 4mA at the line voltage (the remaining 16mA carries the signal).

Current loops offer several advantages over voltage mode output transducers:

– They do not require a precise or stable supply voltage.
– Their insensitivity to IR drops makes them suitable for long distances.
– A 2wire twistedpair cable offers very good noise immunity.
– The 4mA of current required for information transfer serves two purposes:

it can furnish power to a remote module, and it provides a distinction between zero (4 mA) and no information (no current flow).In a 2wire, 4–20mA current loop, supply current for the sensor electronics must not exceed the maximum available, which is 4mA (the remaining 16mAcarries the signal). Because a 3wire current loop is easily derived from the2wire version, the following discussion focuses on the 2wire version.

Need for a Current Loop

The 4–20mA current loop shown in Fig. 1 is a common method of transmitting sensor information in many industrial process monitoring applications. A sensor is a device used to measure physical parameters such as temperature, pressure, speed, liquid flow rates, etc. Transmitting sensory information through a current loop is particularly useful when the information has to be sent to a remote location over long distances (1,000 ft, or more). The loop's operation is straightforward: a sensor's output voltage is first converted to a proportional current, with 4mA normally representing the sensor's zero level output, and 20mA representing the sensor's full scale output. Then, a receiver at the remote end converts the 4–20mA current back into a voltage which in turn can be further processed by a computer or display module. However, transmitting a sensor's output as a voltage over long distances has several drawbacks. Unless very high input impedance devices are used, transmitting voltages over long distances produces correspondingly lower voltages at the receiving end due wiring and interconnect resistances. However, high impedance instruments can be sensitive to noise pickup since the lengthy signal carrying wires often run in close proximity to other electrically noisy system wiring. Shielded wires can be used to minimize noise pickup, but their high cost may be prohibitive when long distances are involved. Sending a current over long distances produces voltage losses proportional to the wiring's length. However, these voltage losses also known as loop drops" do not reduce the 4–20mA current as long as the transmitter an loop supply can compensate for these drops. The magnitude of the current inthe loop is not affected by voltage drops in the system wiring since all of the current (i.e., electrons) originating at the negative (−) terminal of the loop power supply has to return back to its positive (+) terminal

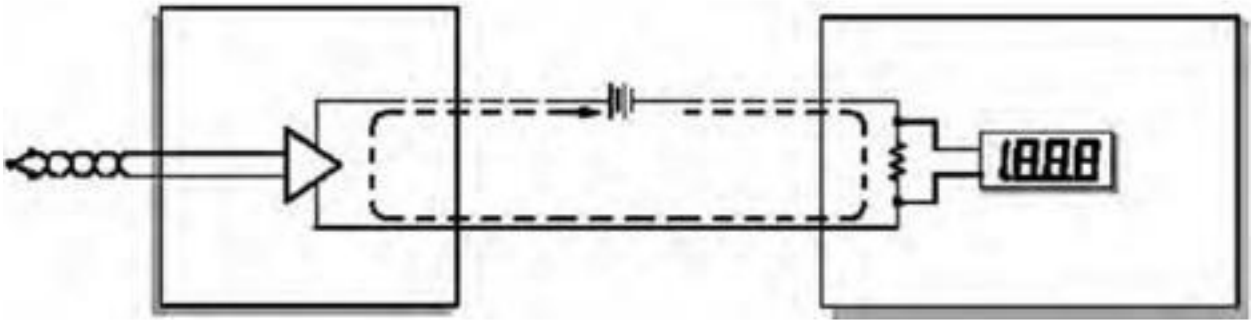


Fig1: Typical components in a loop

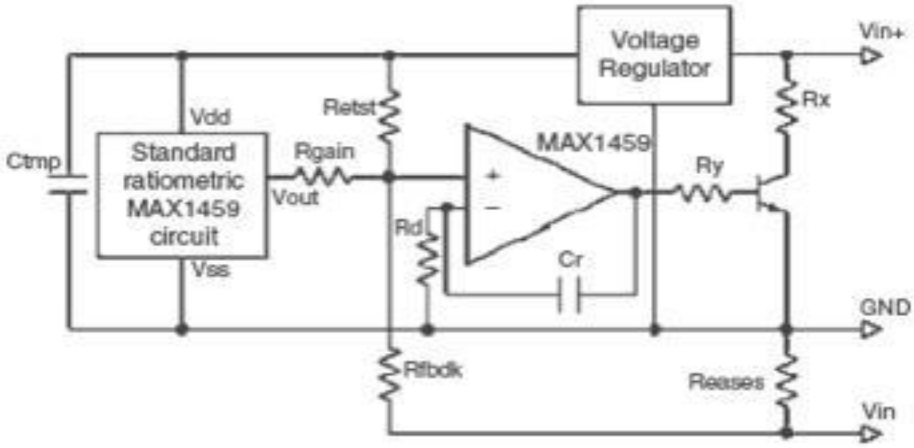


Fig2: 420 mA transmitter

Basic 2wire Circuit

A voltage output should be converted to current when configuring a ratio metric 4–20mA circuit, because current mode applications require a 4mA offset and 16mA span. This section presents the circuit details and results obtained from a current loop configuration based on the MAX1459 sensor signal conditioner. In principle, a voltage regulator is to be added, which converts the 10–32V loop voltage to a fixed 5V for operating the MAX1459. Figure shows the circuitry required for implementing a standard ratio metric version of the MAX1459 circuit. The voltage regulator can be any low cost device whose quiescent current is sufficiently low for the 4mA budget.

Advantages of 4–20mA Current Loop

Current loops offer four major advantages such as:

– Long distance transmission without amplitude loss

– Detection of offline sensors, broken transmission lines, and other failures

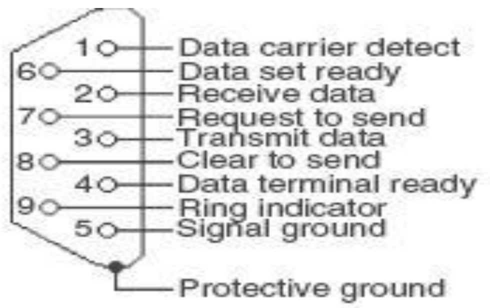
– Inexpensive 2wire cables

– Lower EMI sensitivity

RS 232C/RS 485

The RS232/485 port sequentially sends and receives bytes of information one bit at a time. Although this method is slower than parallel communication, which allows the transmission of an entire byte at once, it is simpler and can be used over longer distances because of lower power consumption. For example, the IEEE 488 standard for parallel communication states that the cable length between the equipments should not exceed 20m total, with a maximum distance of 2m between any two devices. RS232/485 cabling ,however, can extend 1,200m or greater. Typically, RS232/485 is used to transmit American Standard Code for Information Interchange (ASCII) data. Although National Instruments serial hardware can transmit 7bit as wells 8bit data, most applications use only 7bit data. Seven bit ASCII can represent the English alphabet, decimal numbers, and common punctuation marks. It is a standard protocol that virtually all hardware and software understand. Serial communication is mainly using the three transmission lines:

(1) ground, (2) transmit, and (3) receive. Because RS232/485 communications asynchronous, the serial port can transmit data on one line while receiving data on another. Other lines such as the handshaking lines are not required. The important serial characteristics are baud rate, data bits, stop bits, and parity. To communicate between a serial instrument and a serial port on computer, these parameters must match. The RS232 port, or ANSI/EIA232 port, is the serial connection found on Biocompatible PCs. It is used for many purposes, such as connecting amuse, printer, or modem, as well as industrial instrumentation. The RS232protocol can have only one device connected to each port. The RS485 (EIA485 Standard) protocol can have 32 devices connected to each port. With this enhanced multidrug capability the user can create networks of devices connected to a single RS485 serial port. Noise immunity and multi drop capability make RS485, the choice in industrial applications requiring many distributed devices networked to a PC or other controller for data collection.USB was designed primarily to connect peripheral devices to PCs, including keyboards, scanners, and disk drives.RS232 (Recommended standard232) is a standard interface approved by the Electronic Industries Association (EIA) for connecting serial devices. In other words, RS232 is a long established standard that describes the physical interface and protocol for relatively low
speed serial data



Pin Number	Signal	Description
1	DCD	Data carrier detect
2	RxD	Receive data
3	TxD	Transmit data
4	DTR	Data terminal ready
5	GND	Signal ground
6	DSR	Data set ready
7	RTS	Ready to send
8	CTS	Clear to send
9	RI	Ring indicator

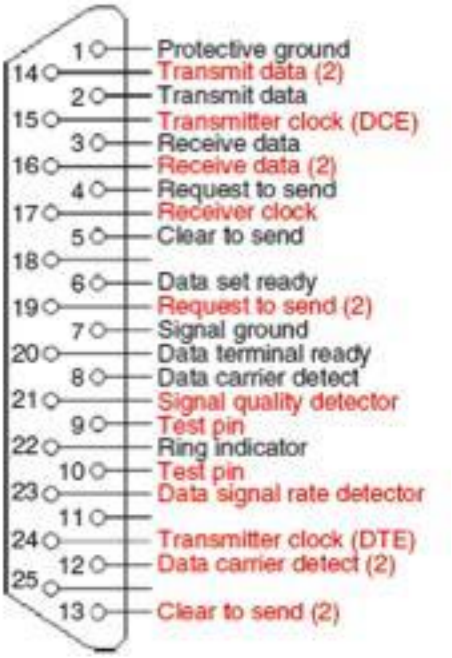


Fig 4: RS232c DB 25 pinout

Signal Descriptions

– TxD – This pin carries data from the computer to the serial device

– RXD – This pin carries data from the serial device to the computer

– DTR signals – DTR is used by the computer to signal that it is ready to communicate with the serial device like modem. In other words, DTR indicates the modem that the DTE (computer) is ON.

– DSR – Similarly to DTR, Data set ready (DSR) is an indication from the modem that it is ON.

– DCD – Data carrier detect (DCD) indicates that carrier for the transmit data is ON.

– RTS – This pin is used to request clearance to send data to a modem.

– CTS – This pin is used by the serial device to acknowledge the computers RTS signal. In most situations, RTS and CTS are constantly on throughout the communication session.

– Clock signals (TC, RC, and XTC) – The clock signals are only used for synchronous communications. The modem or DSU extracts the clock from the data stream and provides a steady clock signal to the DTE. The transmit and receive clock signals need not have to be the same, or even at the same baud rate.

– CD – CD stands for Carrier detect. Carrier detect is used by a modem to signal that it has a made a connection with another modem, or has detected a carrier tone. In other words, this is used by the modem to signal that a carrier signal has been received from a remote modem.

RI – RI stands for ring indicator. A modem toggles (keystroke) the state of this line when an incoming call rings in the phone. In other words, this is used by an auto answer modem to signal the receipt of a telephone ring signal. The carrier detect (CD) and the ring indicator (RI) lines are only available in connections to a modem. Because most modems transmit status information to a PC when either a carrier signal is detected

RS485 Serial Communication

RS485 is an EIA standard for multipoint communications. It supports several types of connectors, including DB9 and DB37. RS485 is similar to RS422 but can support more nodes per line. RS485 meets the requirements for truly multipoint communications network, and the standard specifies up to32 drivers and 32 receivers on a single (2wire) bus. With the introduction of “automatic” repeaters and high impedance drivers/receivers this “limitation "can be extended to hundreds or even thousands of nodes on a network. The RS485 and RS422 standards have much in common, and are often confused for that reason. RS485, which specifies bidirectional, half duplex data transmission, is the only EIA standard that allows multiple receivers and drivers in “bus” configurations. RS422, on the other hand, specifies a single, unidirectional driver with multiple receivers.

GPIB

The purpose of this section is to provide guidance and understanding of the General Purpose Interface Bus (GPIB) bus to new GPIB bus users or to provide more information on using the GPIB bus’s features.GPIB Data Acquisition and Control Module provides analog and digital signals for controlling virtually any kind of a device and the capability toread back analog voltages, digital signals, and temperatures. The 4867 Data Acquisition and Control module is an IEEE488.2 compatible device and has standard Commands for Programmable Instruments (SCPI) command parser that accepts SCPI and short form commands for ease of programming. Applications include device control, GPIB interfacing and data logging. The 4867is housed in a small 7 in.×7 in.

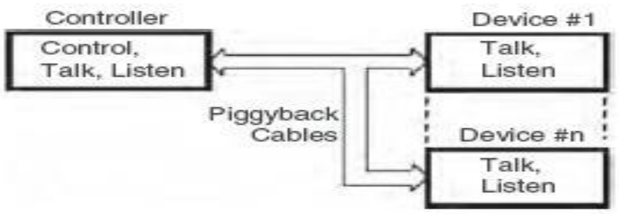


Fig 5: Block Diagram

Controllers have the ability to send commands, to talk data onto the bus and to listen to data from devices. Devices can have talk and listen capability. Control can be passed from the active controller (Controller in charge) to any device with controller capability. One controller in the system is defined as the System Controller and it is the initial controller in charge (CIC).Devices are normally addressable and have a way to set their address. Each device has a primary address between 0 and 30. Address 31 is the unlisted outtalk address.

Types of GPIB Messages

GPIB devices communicate with other GPIB devices by sending device dependent messages and interface messages through the interface system. Device dependent messages, often called data or data messages, contain device specific information, such as programming instructions, measurement results, machine status, and data files. Interface messages manage the bus. Usually called commands or command messages, interface messages perform such functions as initializing thebus.

Physical Bus Structure

GPIB is a 24 conductor as shown in Fig 6Physically, the GPIB interface system consists of 16 lowtrue signal lines and eight ground return or shielddrainlines. The 16 signal lines, discussed later, are grouped into data lines(eight), handshake lines (three), and interface management lines (five).

Data Lines

The eight data lines, DIO1 through DIO8, carry both data and command messages. The state of the Attention (ATN) line determines whether the information is data or commands. All commands and most data use the 7bitASCII or ISO code set, in which case the eighth bit, DIO8, is either un usedor used for parity.

Handshake Lines

Three lines asynchronously control the transfer of message bytes between devices. The process is called a 3wire interlocked handshake. It guarantees that message bytes on the data lines are sent and received without transmission error.

NRFD (not ready for data) – Indicates when a device is ready or not ready to receive a message byte. The line is driven by all devices when receiving commands by listeners when receiving data messages, and by the talker when enabling the HS488 protocol.

– NDAC (not data accepted) – Indicates when a device has or has not accepted a message byte. The line is driven by all devices when receiving commands, and by Listeners when receiving data messages.

– DAV (data valid) – Tells when the signals on the data lines are stable(valid) and can be accepted safely by devices. The Controller drives DAV when sending commands, and the Talker drives DAV when sending data messages. Three of the lines are handshake lines, NRFD, NDAC, and DAV, which transfer data from the talker to all devices who are addressed to listen. Thetalker drives the DAV line; the listeners drive the NDAC and NRFD lines. The remaining five lines are used to control the bus's operation.

– ATN (attention) is set true by the controller in charge while it is sending interface messages or device addresses. ATN is false when the bus is transmitting data.

– EOI (end or identify) can be asserted to mark the last character of a message or asserted with the ATN signal to conduct a parallel poll.

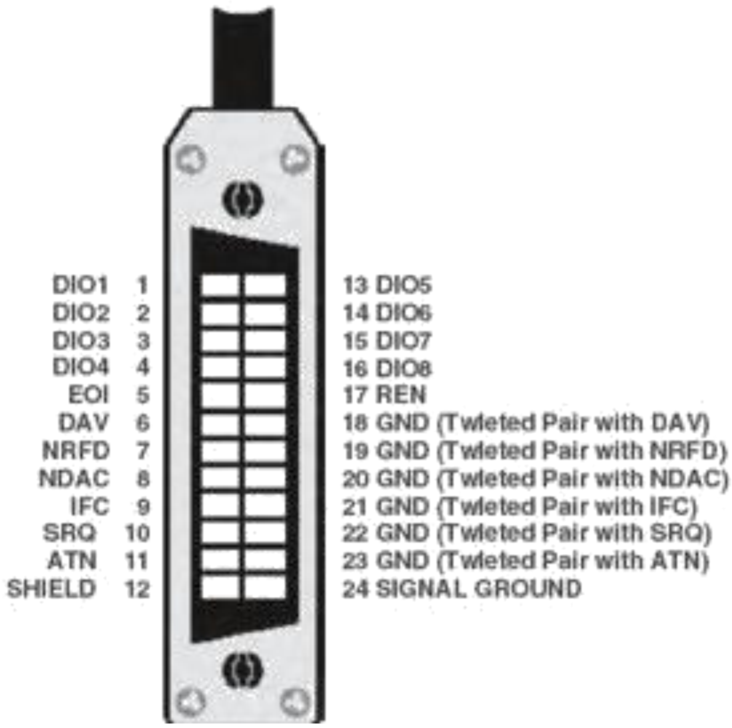


Fig :6 Bus structure of GPIB

USB

The USB is a medium speed serial data bus designed to carry relatively large amounts of data over relatively short cables: up to about 5m long. It can support data rates of up to 12Mbs−1 (megabits per second), which is fast enough for most PC peripherals such as scanners, printers, keyboards, mice, joysticks, graphics tablets, lowers digital cameras, modems, digital speakers, low speed CDROM and CD writer drives, external Zip disk drives, and soon. The USB is an addressable bus system, with a 7bit address code so it can support up to 127 different devices or nodes. However, it can have only one host. So a PC with its peripherals connected through the USB forms a star Local Area Network (LAN).On the other hand any device connected to the USB can have a number of other nodes connected to it in daisychain fashion, so it can also form the

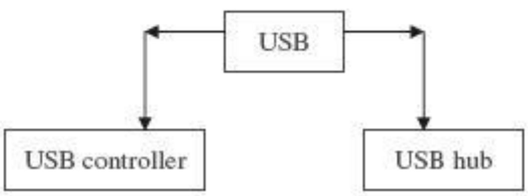


Fig 7: USB Structure

Most hubs provide either four or seven downstream ports, or less. Another important feature of the USB is that it is designed to allow hot swapping. Devices can be plugged into and unplugged from the bus without having to turn the power off and on again, reboot the PC or even, manually starta driver program. A new device can simply be connected to the USB, and the PCs operating system should recognize it and automatically set up the necessary driver to service it.

Need for USB

The USB is host controlled. There can only be one host per bus. The specification in itself does not support any form of multi master arrangement. This is aimed at and limited to single point to point connections such as a mobile phone and personal organizer and not multiple hub, multiple device desktop configurations. The USB host is responsible for undertaking all transactions and scheduling bandwidth. Data can be sent by various transaction methods using a token based protocol. One of the original intentions of USB was to reduce the amount of cabling at the back of PC. The idea came from the Apple Desktop Bus, where the keyboard, mouse and some other peripherals could be connected together (daisy chained) using the one cable .However, USB uses a tiered star topology, similar to that of 10BaseTEthernet. This imposes the use of a hub somewhere, which adds to greater expense, more boxes on desktop and more cables. However it is not as bad as it may seem. Many devices have USB hubs integrated into them. For example, keyboard may contain a hub which is connected to computer. Mouse another devices such as digital camera can be plugged easily into the back of keyboard. Monitors are just another peripheral on a long list which commonly has inbuilt hubs. This tiered star topology, rather than simply daisy chaining devices together has some benefits. First power to each device can be monitored and even switched off if an over current condition occurs without disrupting other USB devices. High, full and low speed devices can be supported, with the hub filtering out high speed and full speed transactions so lower speed devices do not receive them. Up to 127 devices can be connected to any one USB bus at any one given time. To extent devices simply add another port/host. While earlier USB hosts had two ports, most manufacturers have seen this as limiting and are starting to introduce 4 and 5 port host cards with an internal port for hard disks etc. The early hosts had one USB controller and thus both ports shared the same available USB bandwidth. As bandwidth requirements have increased, multiport cards with two or more controllers allowing individual channels are used.

Table: Pin connections

Pin connections	
Pin No.	Signal
1	+5V Power
2	− Data
3	+ Data
4	Ground

USB cables use two different types of connectors:
– *Type A*. Plugs for the upstream end.
– *Type B*. Plugs for the downstream end.

PCMCIA

Personal Computer Memory Card International Association (PCMCIA) is an international standards body and trade association founded in 1989developed a standard for small, credit card sized devices, called PC Cards. Originally designed for adding memory to portable computers, the PCMCIA standard has been expanded several times and is now suitable for many types of devices. The inclusion of PCMCIA technology in PCs delivers a variety of benefits. Besides providing an industry standard interface for third party cards (PC Cards), PCMCIA allows users to easily swap cards in and out of a PC as needed, without having to deal with the allocation of system resources for those devices. These useful features hot swapping and automatic configuration, as well as card slot power management and other PCMCIA capabilities –are supported by a variety of software components on the PCMCIA based PC. In most cases, the software aspect of PCMCIA remains relatively transparent to the user. As the demand for notebook and laptop computers began sky rocketing in the late 1980s, users realized that their expansion options were fairly limited. Mobile machines were not designed to accept the wide array of available expansion cards that their desktop counterparts could enjoy

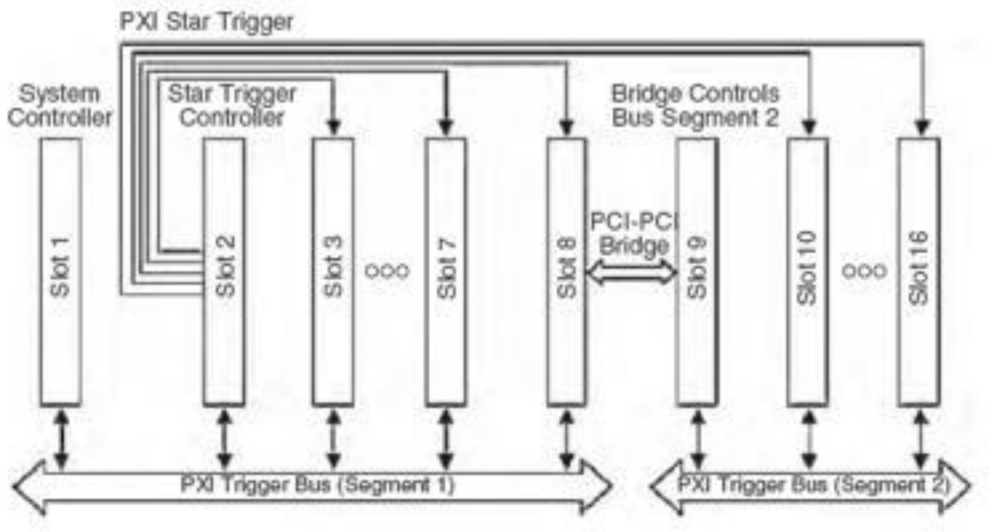


Fig 8: Triggering Architecture

Features of PCMCIA

– One rear slot, access from rear of PC
– Accept Type I/II/III Cards
– Comply with PCI Local Bus Specification Rev.2.2
– Comply with 1995 PC Card Standard
– Extra compatible registers are mapped in memory
– Use T/I 1410 PCMCIA controller
– Support PC Card with Hot Insertion and Removal
– Support 5V or 5/3.3V 16bit PC Cards
– Support Burst Transfers to Maximize Data Throughput on both PCIBuses
– Supports Distributed DMA and PC/PCI DMA

Utilities of PCMCIA Card in the Networking Category

Under the networking category, typically PCMCIA slot supports 4 types
of cards such as:
– LAN card
– Wireless LAN card
– Modem card
– ATA flash disk card

VISA

Virtual Instrumentation Software Architecture (VISA) is a standard I/O language for instrumentation programming. VISA by itself does not provide instrumentation programming capability. VISA is a high-level API that calls into lower level drivers.
VISA is capable of controlling VXI, GPIB, or Serial instruments and makes the appropriate driver calls depending on the type of instrument being used. When debugging VISA problems, it is important to keep in mind that this hierarchy exists.

Fig9: VISA Architecture

The terminology related with VISA are explained as follows:

Resources. The most important objects in the VISA language are known as resources. Operations. In object oriented terminology, the functions that can be used with an object are known as operations Attributes. In addition to the operations that can be used with an object, the object has variables associated with it that contain information related to the object. In VISA, these variables are known as attributes. There is a default resource manager at the top of the VISA hierarchy that can search for available resources or open sessions to them. Resources can be GPIB, serial message based VXI, or register based VXI. The most common operations for message based instruments are read and write.

Waveform generator

Standalone traditional instruments such as oscilloscopes and waveform generators are very powerful, expensive, and designed to perform one or more specific tasks defined by the vendor.

However, the user generally cannot extender customize them. The knobs and buttons on the instrument, the built in circuitry, and the functions available to the user, are all specific to the nature of the instrument. In addition, special technology and costly components must be developed to build these instruments, making them very expensive and slow to adapt.

UNITV

Fourier Transforms

LabVIEW and its analysis VI library provide a complete set of tools to perform Fourier and spectral analysis. The Fast Fourier Transform (FFT) and Power Spectrum VIs are optimized, and their outputs adhere to the standard DSP format.

FFT is a powerful signal analysis tool, applicable to a wide variety of fields including spectral analysis, digital filtering, applied mechanics, acoustics, medical imaging, modal analysis, numerical analysis, seismogryphy, instrumentation, and communications.

The LabVIEW analysis VIs, located on the **Signal Processing** palette, maximize analysis throughput in FFT related applications. This document discusses FFT properties, how to interpret and display FFT results, and how to manipulate FFT and power spectrum results to extract useful frequency information.

FFT Properties

The fast Fourier transform maps time domain functions into frequencydomain representations.
FFT is derived from the Fourier transform equation, which is

$$X(f) = \boldsymbol{F}\{x(t)\} = \int_{-\infty}^{\infty} x(t)\, e^{-j2\pi ft}\, dt \ ,$$

where x(t) is the time domain signal, X(f) is the FFT, and ft is the frequency to analyze. Similarly, the discrete Fourier transform (DFT) maps discrete time sequences into discrete frequency representations. DFT is given by the following equation

$$X_k = \sum_{i=0}^{n-1} x_i\, e^{-j2\pi ik/n} \qquad \text{for } k = 0, 1, 2, \dots, n-1$$

where x is the input sequence, X is the DFT, and n is the number of samples in both the discrete time and the discrete frequency domains. Direct implementation of the DFT, as shown in equation 2, requires approximately n complex operations. However, computationally efficient algorithms can require as little as n log2(n) operations.

$$F\{x\} = X = X_{Re} + j X_{Im} = Re\{X\} + j Im\{X\}$$

An inherent DFT property is the following:

$$X_{n-i} = X_{-i}$$

FFT Standard Output:

The output format of the FFT VI can now be described with X. If the total number of samples, n, of the input sequence, x, is even, the format of the complex output sequence, X, and the interpretation of the results are shown in the below table.

Table 1: Output Format for Even n, k = n ÷ 2

Array Element	**Interpretation**
X[0]	DC component
X[1]	1st harmonic or fundamental
X[2]	2nd harmonic
.	.
.	.
.	.
X[k2]	(k2)th harmonic
X[k1]	(k1)th harmonic
X[k] = X[k]	Nyquist harmonic
X[k+1]= X[n(k1)] = X[(k1)]	(k1)th harmonic
.	
.	
.	
X[n3]	3rd harmonic
X[n2]	2nd harmonic
X[n1]	1st harmonic

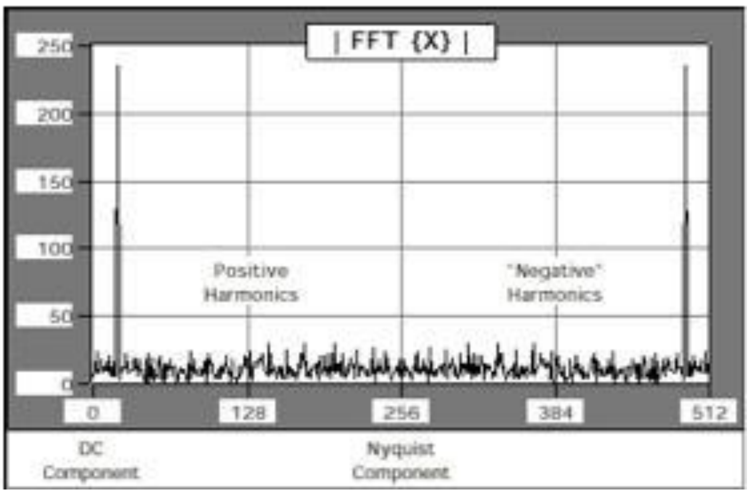


Fig 1 : Graph of table

If the total number of samples, n, of the input sequence, x, is odd, the format of the complex output sequence, X, and the interpretation of the results are shown in the below table

Table:2Output Format for Odd n, k = (n1) ÷ 2

Array Element	**Interpretation**
X[0]	DC component
X[1]	1st harmonic or fundamental
X[2]	2nd harmonic
.	.
.	.
.	.
X[k1]	kth 1harmonic
X[k]	kth harmonic
X[k+1]= X[nk] = X[k] kth	kth harmonic
. X[k+2]= X[n(k1)] = X[(k1)]	(k1)th harmonic
.	
.	
X[n3]	3rd harmonic
X[n2]	2nd harmonic
X[n1]	1st harmonic

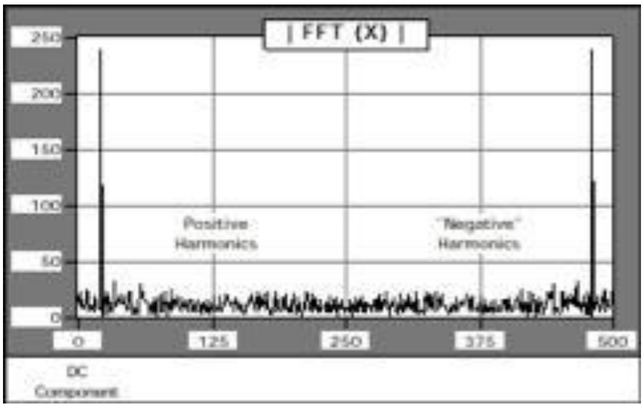


Fig 2 : Graph of table 2

Standard Output

Standard output is the format for even and odd sized discrete time sequences, described in Tables 1 and 2 of this document. This format is convenient because it does not require any further data manipulation. To graphically display the results of the FFT, wire the output arrays to the waveform graph Fig : 2 Graph of table2. The FFT output is complex and requires two graphs to display all the information.

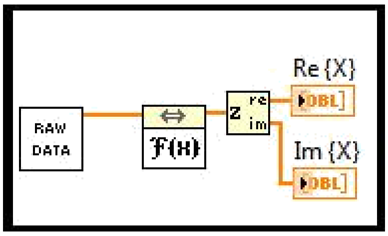


Fig 3 : Standard output by using

VI Power spectrum and Correlation:

The power spectrum reveals the existence, or the absence, of repetitive patterns and correlation structures in a signal process. These structural patterns are important in a wide range of applications such as data forecasting, signal coding, signal detection, radar, pattern recognition, and decision-making systems. The most common method of spectral estimation is based on the fast Fourier transform (FFT).

In particular, the total power is given by

$$P_x(\omega) = \sum_{k=-\infty}^{\infty} r_x[k] e^{-jk\omega}$$

One can show that Px(w) is real, even and positive. The autocorrelation can be recovered with the inverse Fourier transform

$$r_x[k] = \frac{1}{2\pi} \int_{-\pi}^{\pi} d\omega P_x(\omega) e^{jk\omega}$$

Power spectrum – properties:

In particular, the total power is given by

$$r_x[0] = \frac{1}{N} \sum_{n=1}^{N} |x[n]|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} d\omega P_x(\omega)$$

The power spectrum is sometimes called **spectral density** because it is *positive* and the signal power can always be*normalized* to *r*(0) =(2p)1 .

Example: Uncorrelated noise has a constant power spectrum

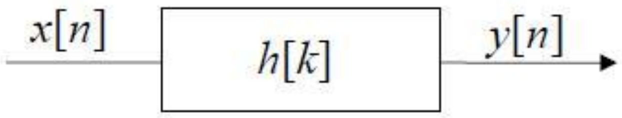
$$r[k] = \sigma^2 \delta(k)$$

$$P_x(\omega) = \sum_{k=-\infty}^{\infty} \sigma^2 \delta(k) e^{-jk\omega} = \sigma^2$$

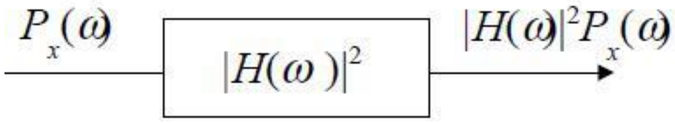
Hence it is also called **white noise**.

Effect of Filtering on Power Spectrum

A linear system with impulse response $h[k]$



Transforms the power spectrum as



Spectral Content

The power spectrum gives the spectral content of the data. To see that consider the power of a signal after filtering with a narrow band pass filter around w 0.

$$
\begin{aligned}
E[|y[n]|^2] &= \frac{1}{2\pi}\int_{-\pi}^{\pi} d\omega P_y(\omega) \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} d\omega |H(\omega)|^2 P_x(\omega) \\
&= \frac{1}{2\pi}\int_{\omega_0-\Delta\omega/2}^{\omega_0+\Delta\omega/2} d\omega P_x(\omega) \\
&\approx \frac{\Delta\omega}{2\pi} P_x(\omega_0)
\end{aligned}
$$

Power spectrum properties

The power spectrum captures the spectral content of the sequence. It can be estimate directly from the Fourier transform of the data.

$$\hat{P}_x(\omega) = \frac{1}{N}|X(\omega)|^2$$

$$X(\omega) = \sum_{k=0}^{N-1} x[k] e^{-j\omega k}$$

Correlation methods

Correlation is a statistical technique that can show whether and how strongly pairs of variables are related.

$$r_{yx}(k)=\sum_{n=-\infty}^{\infty} y^*[n]x[n+k]$$

Correlation Coefficient

The main result of a correlation is called the **correlation coefficient** (or "r"). It ranges from 1.0 to +1.0. The closer r is to +1 or 1, the more closely the two variables are related.

Cross and Autocorrelation

The cross correlation is defined as

Note that correlation is a convolution with opposite sign. It can be computed with the Fourier transform.

$$R_{xy}(\omega)=Y^*(\omega)X(\omega)$$

The autocorrelation is defined as

$$r_x(k)=\sum_{n=-\infty}^{\infty} x^*[n]x[n+k]$$

For a sample of finite length *N* this is typically normalized. We call this the sample autocorrelation

Autocorrelation properties

The autocorrelation is symmetric.

$$r_x[k]=r_x^*[-k]$$

The zero lag gives the total power of the signal

The autocorrelation has the power as an upperbound

$$r_x[0]\geq|r_x[k]| \quad r_x[0]=\sum_n |x[n]|^2$$

Windowing & flittering

Windowing is the process of taking a small subset of a larger dataset, for processing and analysis. A naive approach, the rectangular window, involves simply truncating the dataset before and after the window, while not modifying the contents of the window at all. However this is a poor method of windowing and causes power leakage.

Windowing of a simple waveform like cosωt causes its Fourier transform to develop nonzero values (commonly called spectral leakage) at frequencies other than ω. The leakage tends to be worst (highest) near ω and least at frequencies farthest from ω.

If the waveform under analysis comprises two sinusoids of different frequencies, leakage can interfere with the ability to distinguish them spectrally. If their frequencies are dissimilar and one component is weaker, then leakage from the stronger component can obscure the weaker one's presence. But if the frequencies are similar, leakage can render them irresolvable even when the sinusoids are of equal strength. The rectangular window has excellent resolution characteristics for sinusoids of comparable strength, but it is a poor choice for sinusoids of disparate amplitudes. This characteristic is sometimes described as low dynamic range.

At the other extreme of dynamic range are the windows with the poorest resolution and sensitivity, which is the ability to reveal relatively weak sinusoids in the presence of additive random noise. That is because the noise produces a stronger response with high dynamic ange windows than with high resolution windows. Therefore, high dynamic range windows are most often justified in wideband applications, where the spectrum being analyzed is expected to contain many different components of various amplitudes.

In between the extremes are moderate windows, such as Hamming and Hann. They are commonly used in narrowband applications, such as the spectrum of a telephone channel. In summary, spectral analysis involves a tradeoff between resolving comparable strength components with similar frequencies and resolving disparate strength components with dissimilar frequencies. That tradeoff occurs when the window function is chosen.

Application in Process Control projects

Software for Remote ON/OFF Control Experiment

The software for the Remote ON/OFF controller lab experiment is developed using LabVIEW.

There are two distinct parts in the software:

1. ON/OFF controller server program and logic

2. Internet communication using DataSocket protocol

These parts will be explained in the following sections

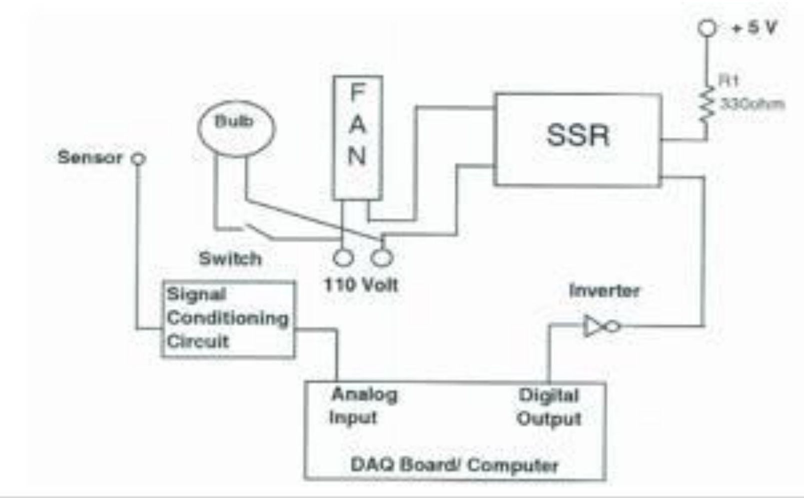


Fig 4: ONOFF Temperature Control

ON/OFF Controller Server Program and Logic

The LabVIEW program on the server first reads the voltage from the analog input channel of the DAQ board and converts it to a temperature using the equation

$$Tout = 6\ Vin + 60$$

where Tout is the process temperature, and Vin is the input voltage from the DAQ board. The desired Setpoint temperature value of the process is obtained from the client computer, and the Error is determined using the equation

$$Error = Setpoint\ .\ Tout.$$

Based on the error and neutral zone High Limit and Low Limit, an ON/OFF controller logic is implemented using LabVIEW. In this logic, Digital Output Channel is the output logic value sent to the DAQ board which controls the fan through the SSR, Cooling Fan Indicators are the LED ON/ OFF indicators on the front panel of the LabVIEW VI that show the state of the fan, and Within Limits Indicator shows if the process is operating within the neutral zone. The Cooling Fan ON Indicator is also defined as a local read only variable which is used in the logic implementation.

The LabVIEW front panel of the server program VI

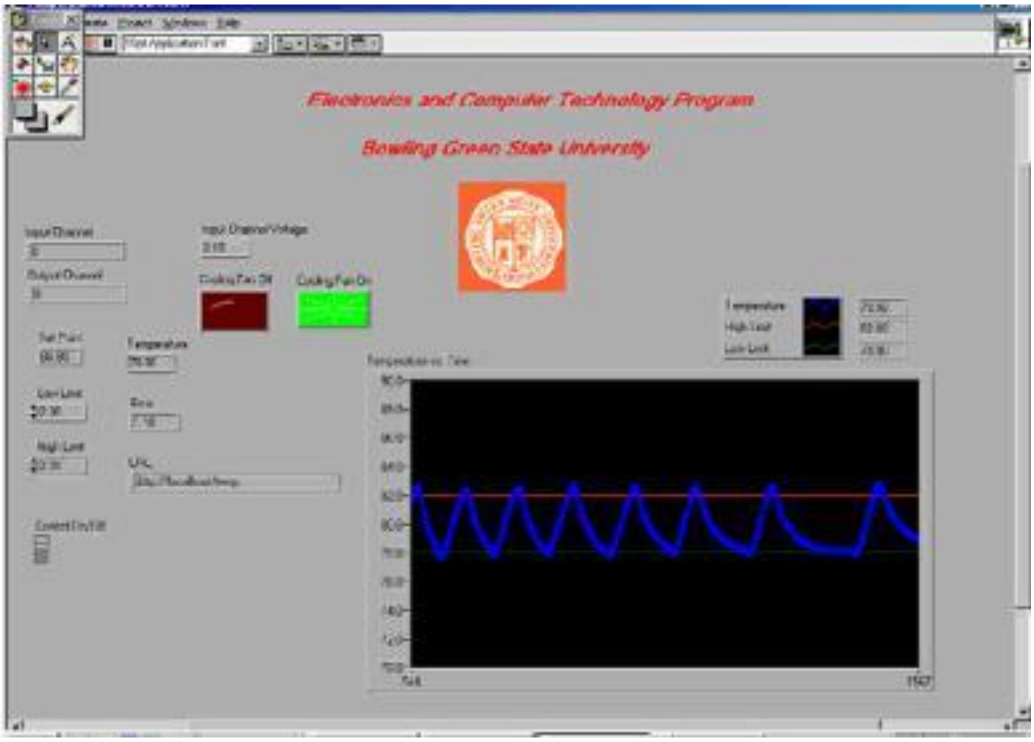


Fig 5: Server front panel

Temperature data acquisition system

The hardware for server workstation consists of a PC with Pentium III, 550 MHz processor, 128 Mb RAM, Network Interface Card (NIC), National Instruments DAQ board. The server software includes Windows 98, NIDAQ driver, LabVIEW 5.1 and NIData Socket Manager. The designed experiment is connected to the DAQ board. The server is assigned static IP address. The clients could be any PC.s with NIC that can run a LabVIEW program. The objective of the experiment is to maintain the temperature inside a wooden box at some desired set point value, within neutral zone limits, using a two state controller mode. The wooden box is heated with a light bulb. The temperature is measured using LM335 solid-state The hardware for server workstation consists of a PC with Pentium III, 550 MHz processor, 128 Mb RAM, Network Interface Card (NIC), National Instruments DAQ board. The server software includes Windows 98, NIDAQ driver, Lab VIEW 5.1 and NIData Socket Manager. The designed experiment is connected to the DAQ board. The server is assigned static IP address. The clients could be any PC.s with NIC that can run a LabVIEW program. The objective of the experiment is to maintain the temperature inside a wooden box at some desired set point value, within neutral zone limits, using a two state controller mode. The wooden box is heated with a light bulb. The temperature is measured using LM335 solid state

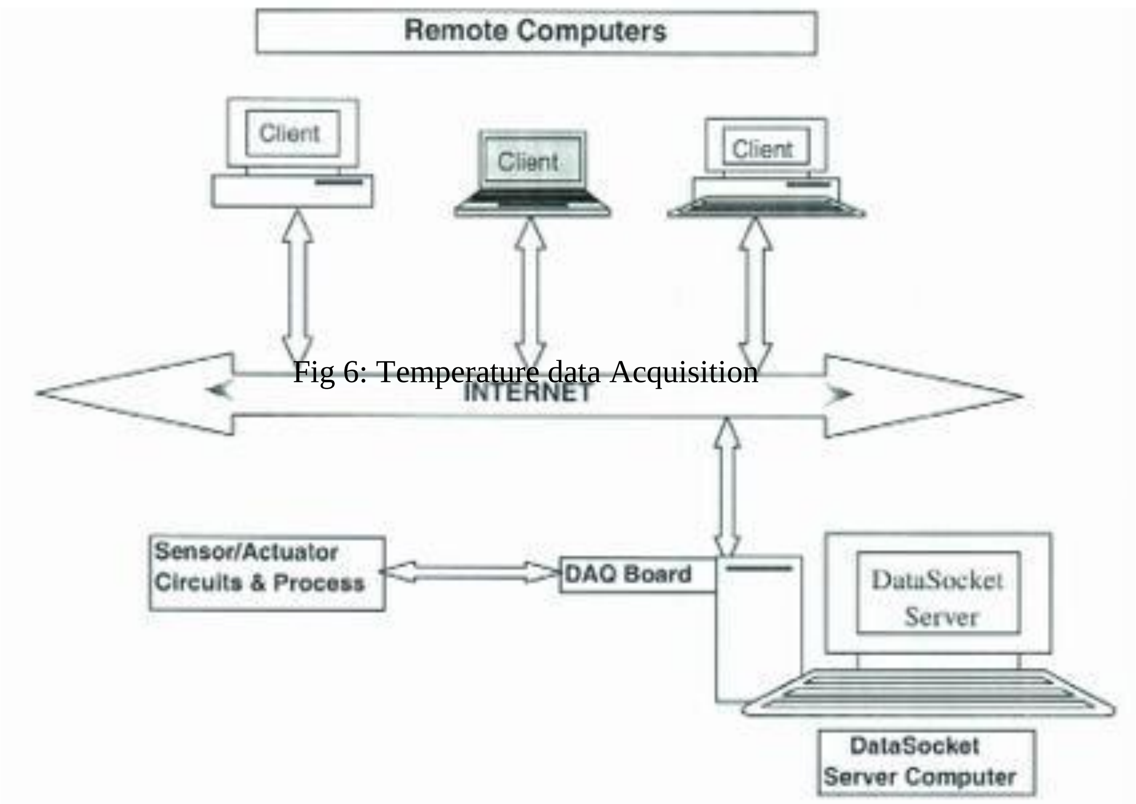


Fig 6: Temperature data Acquisition

Oscilloscope

Open Windows oscilloscopes from the 5000, 6000, 7000, and 8000 series are Windowsbased and contain scopespecific software for acquisition, connectivity, and control. By default, the scope user interface runs Windowsbased application software that performs the following scopespecific tasks:

• Presents the scope user interface

• Configures scope hardware based on commands from the user and signal content

• Displays acquired signals and derivations of acquired signals

You can use LabVIEW to extend the feature list of your scope to include the following tasks:

• Custom analysis and signal processing of acquired signals

• Automation of scope related tasks and sequences of tasks • A unique and customizable user interface

• Custom reports, including reports that you can publish live over the Internet

•

Example:

Connect the function generator to the scope and put up the sample waveform. Increase the amplitude to 2v and pull down the VISA box and click on GPIB for the oscilloscope. The display of the wave is shown in the fig 7.

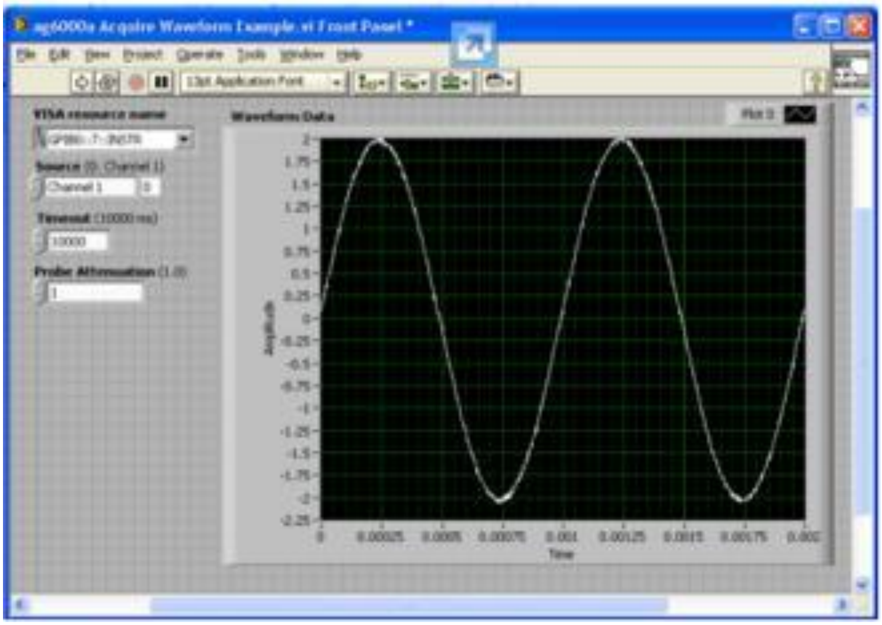


Fig 7: waveform acquired by oscilloscope

Digital Multimeter

Digital Multimeters (DMMs) are specialized in takingflexible, high resolution measurements

• Low to medium acquisition speeds

• High resolution

• Measures current or resistance in addition to voltage.

• Different form factors: PXI, PCIe, PCI, USBSoftware Support in NI LabVIEW and SignalExpressThe NI 407x series of DMMs offer unique capabilities − Isolated Digitizer Mode at up to 1.8 MS/s

− Industry Leading Accuracy

− 2year Calibration Cycle

• USB4065 & PCMCIA4050 for portable measurements

Example:

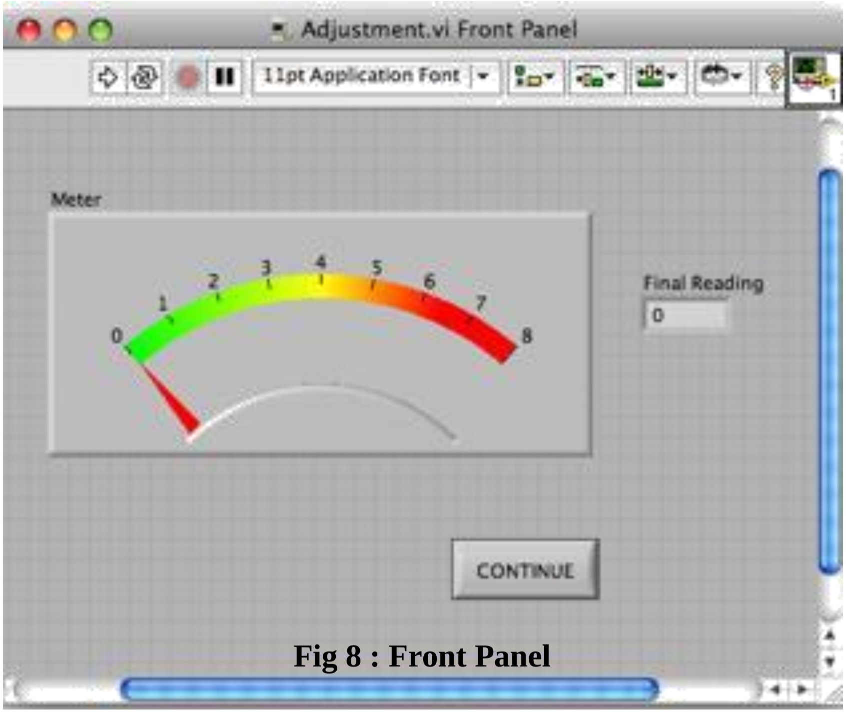


Fig 8 : Front Panel

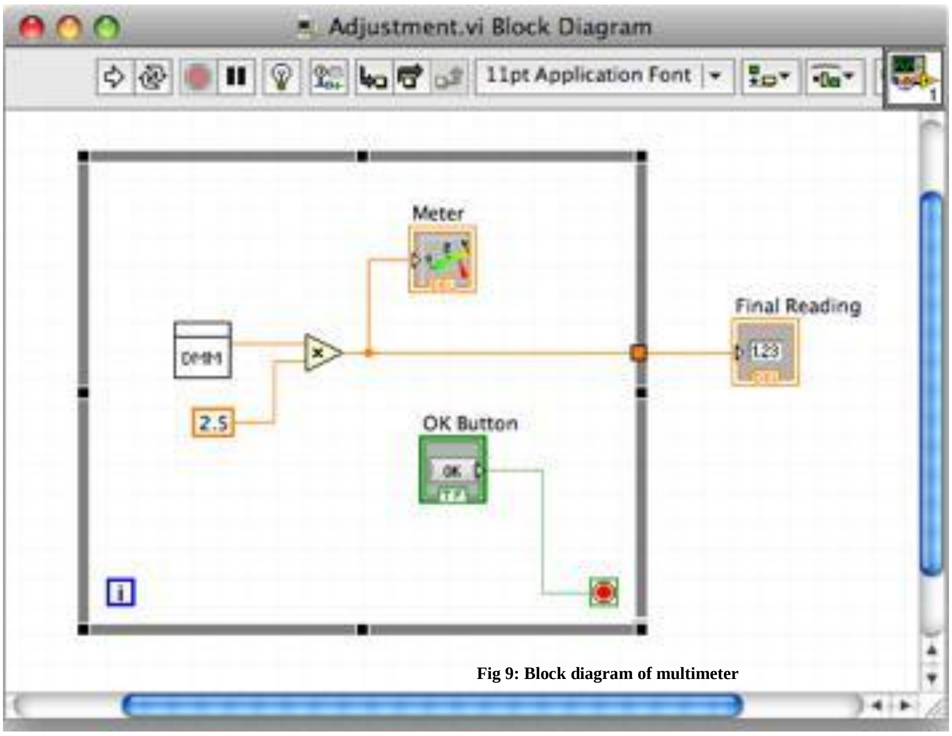


Fig 9: Block diagram of multimeter

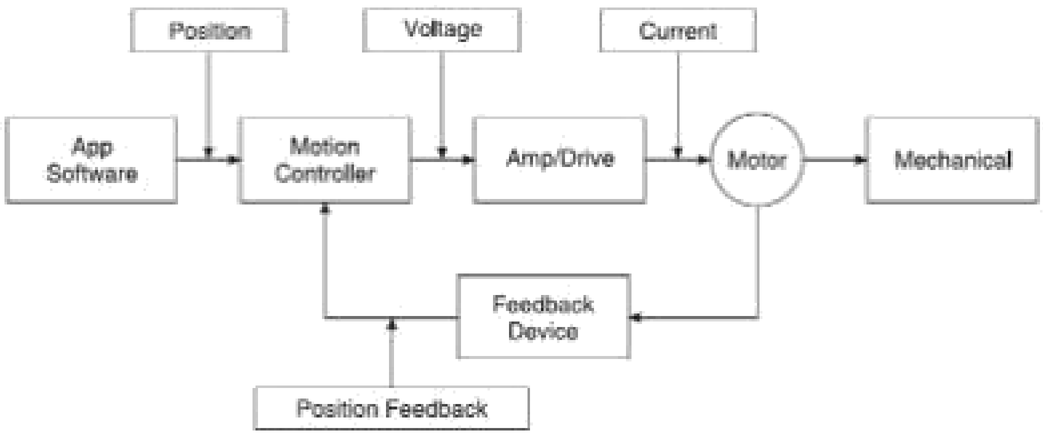


Fig 10: Components of motion control

Motion controller: The motion controller acts as brain of the system by taking the desired target positions and motion profiles and creating the trajectories for the motors to follow, but out putting a ±10 V signal for servomotors, or a step and direction pulses for stepper motors.

Amplifier or drive: Amplifiers (also called drives) take the commands from the controller and generate the current required to drive or turn the motor.

Motor: Motors turn electrical energy into mechanical energy and produce the torque required move to the desired target position.

Mechanical elements: Motors are designed to provide torque to some mechanics. These include linear slides, robotic arms and special actuators.

Feedback device or position sensor: A position feedback device is not required for some motion control applications (such as controlling stepper motors), but is vital for servomotors. The feedback device, usually a quadrature encoder, senses the motor position and reports the result to the controller.

Image Acquisition and Processing

Image analysis combines techniques that compute statistics and measurements based on the graylevel intensities of the image pixels. Image processing contains information about lookup tables, convolution kernels, spatial filters and grayscale morphology. The lookup table (LUT) transformations are basic image processing functions that highlight details in areas containing significant information at the expense of other areas. These functions include histogram equalization, gamma corrections logarithmic corrections ,and exponential corrections. NIIMAQ is a complete and robust API for image acquisition. Whether you are using LabVIEW,

Measurement Studio, Visual Basic, or Visual C++, NIIMAQ gives you high level control of National Instruments image acquisition devices. NIIMAQ performs all of the computer and board specific tasks, allowing straightforward image acquisition without register level programming. NIIMAQis compatible with NIDAQ and all other National Instruments driver software for easily integrating an imaging application into any National Instruments solution. NIIMAQ is included with your hardware at no charge that you can call from your application programming environment. These functions include routines for video configuration, image acquisition (continuous and singleshot), memory buffer allocation, triggercontrol and board configuration. NIIMAQ performs all functionality required to acquire and saveimages. For image analysis functionality, refer to the IMAQ Vision software analysis librarieswhich are discussed later in this course. NIIMAQ resolves many of the complex issues betweenthe computer and IMAQ hardware internally, such as programming interrupts and DMA controllers.

Fig 11: NI IMAQ functions

NIIMAQ and IMAQ Vision use five categories of functions to acquire and display images:

(2) **Utility functions**—Allocate and free memory used for storing images; begin and endimag acquisition sessions

(3) **Single buffer acquisition functions**—Acquire images into a single buffer using the snap and grab functions

(4) **Multiple buffer acquisition functions**—Acquire continuous images into multiple buffers using the ring and sequence functions

(5) **Display controls**—Display images for processing

(6) **Trigger functions**—Link a vision function to an event external to the computer, such as receiving a pulse to indicate the position of an item on an assembly line

Snaps and grabs are the most basic types of acquisitions. A snap is simply a snapshot, in which you acquire a single image from the camera. A grab is more like a video, in which you acquire every image that comes from the camera. The images in a grab are displayed successively ,producing a full motion video, consisting of around 25 to 30 frames per second. IMAQ Snap, IMAQ Grab Setup and IMAQ Grab Acquire are used to snap and grab images.

IMAGE PROCESSING TOOLS AND FUNCTIONS IN IMAQ VISION

Utility functions include VIs for image management and manipulation, file management, calibration and region of interest processing. Image processing functions include VIs for analysis, color processing, frequency processing, filtering, morphology, operations, and processing, includingIMAQ Histogram, IMAQ Threshold and IMAQ Morphology. Machine vision VIs are used for common inspection tasks such as checking for the presence or absence of parts in an image or measuring dimensions in comparison to specifications. Some examples of the machine vision Visage the caliper and coordinate system VIs.

Image processing is a time-consuming process, both in computer processor time and development time. National Instruments has developed an application to accelerate the design time of a machine vision application..IMAQVision Assistant allows even the first time vision developer to learn image processing techniques and test inspection strategies. In addition, more experienced developers can develop and explore vision algorithms faster with less programming.