Document Type: Tutorial
NI Supported: Yes
Publish Date: Mar 1, 2007

A Review of PC-Based Data Logging and Recording Techniques

Table of Contents

Overview

Data logging and recording is a very common measurement application. In its most basic form, data logging is the measurement and recording of physical or electrical parameters over a period of time. The data can be temperature, strain, displacement, flow, pressure, voltage, current, resistance, power, or any of a wide range of other parameters. Real-world data logging applications are typically more involved than just acquiring and recording signals, typically involving some combination of online analysis, offline analysis, display, report generation, and data sharing. Moreover, many data logging applications are beginning to require the acquisition and storage of other types of data, such as recording sound and video in conjunction with the other parameters measured during an automobile crash test.

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

The list of applications for data logging goes on and on, but all of these applications have similar common requirements. The purpose of this paper is to provide a general background on data logging, discuss the various functional requirements that are common to most logging applications, and examine some of the modern hardware and software options available to scientists and engineers for implementing powerful PC-based data logging systems.

Figure 1. PC-Based Data Logging Application - Performance Testing of Refrigerator Designs

Historical Background

The earliest form of data logging involved taking manual measurements from analog instruments such as thermometers and manometers. These measurements were recorded into a written log, along with the time of observation. To view trends over time, people manually plotted their measurements on graph paper. In the late 19th century, it became possible to begin automating this process with machines, and strip chart recorders evolved. Strip chart recorders are analog instruments that translate electrical impulses from sensors into mechanical movement of an arm. A pen is attached to the arm, and long rolls of paper are moved at a constant rate under the pen. The result is a paper chart displaying the parameters measured over the course of time. Strip chart recorders were a great leap over manual data logging, but still had drawbacks. For example, translating the traces on the paper into meaningful engineering measurements was tedious at best, and the data recorded took up reams and reams of paper.

With the development of the personal computer in the 1970s and 80s, people began to use computers for analysis of data, data storage, and report generation. The need to bring data into the PC brought about dataloggers – a new special-purpose device for data logging. Dataloggers are stand-alone, box instruments that measure signals, convert to digital data, and store the data internally. This data must be transferred to the PC for analysis, permanent storage, and report generation. Data is typically transferred either by manually moving a storage device (such as a floppy disk) from the datalogger to the computer or by connecting the datalogger to the PC through some communications link such as serial or Ethernet.

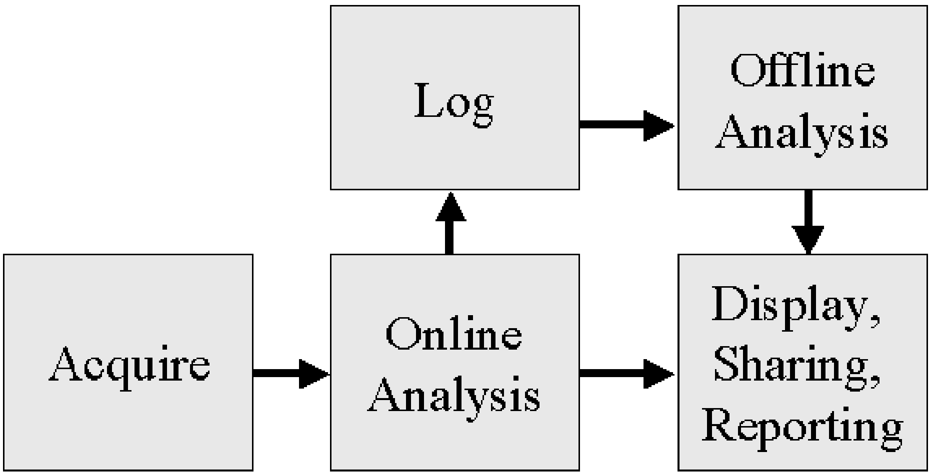
In the 1990s, a further evolution in data logging took place as people begin to create PC-based data logging systems. These systems combine the acquisition and storage capabilities of stand-alone dataloggers with the archiving, analysis, reporting, and display capabilities of modern PCs. PC-based logging systems finally brought about full automation of the data logging process. The move to PC-based data logging systems was enabled by three technological enhancements:

1. Increasing reliability of PCs.
2. Steadily decreasing cost of hard drive space on PCs.
3. PC-based measurement hardware that could meet or exceed measurement capabilities of stand-alone dataloggers.

Today, PC-based logging systems provide the widest range of measurement types, analysis capabilities, and reporting tools. The remainder of this paper will focus on the functionality necessary to implement a PC-based data logging system 1-Wire Weather.

Data Logging Functional Requirements

Every data logging application, from manually recording weather patterns in the 15th century, to logging the experimental parameters of a fusion reactor test in the 21st century, can be broken down into a set of five common functional requirements, illustrated in Figure 2. Acquiring is the process of actually measuring the physical parameters and bringing them into your logging system. Online analysis consists of any processing done to the data while you are acquiring. It includes alarms, data scaling, and sometimes control, among others. Logging, or storing, the data is an obvious requirement of every data logging system. Offline analysis is everything done with the data after it has been acquired in order to extract useful information from it. The final functional block is made up display, reporting, and data sharing. These are all the "miscellaneous" requirements that fill out the functionality of a data logging system. Let's examine how each of these functional blocks is addressed with modern, PC-based data logging systems.



[+] Enlarge Image

Figure 2. Basic Elements of a Data Logging System

Acquisition

The acquisition function is one of the most critical components of every data logging system. In a PC-based system, the acquisition is accomplished by the measurement hardware, which can be further broken down into sensors, signal connectivity, signal conditioning, and analog-to-digital conversion, as shown in Figure 3. Since these topics are well covered in other material, only a high-level overview will be given here.

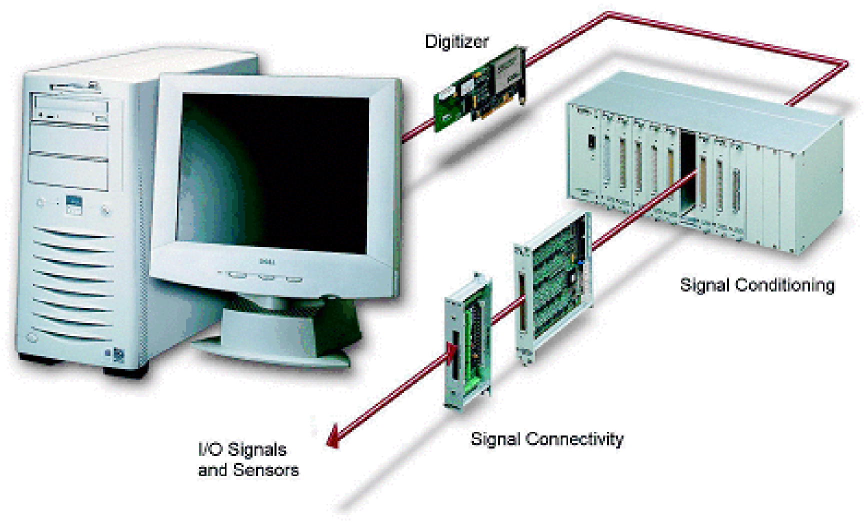


Figure 3. Measurement Hardware Components of PC-Based Data Logging System

Sensors
A wide variety of sensors is used to convert physical parameters into electrical signals. Temperature sensors such as thermocouples, RTDs, or thermistors are some of the most common sensors used in data logging applications. Other widespread sensors are flow meters, pressure transducers, strain gauges, accelerometers, and microphones, to name a few. Proper selection and installation of sensors is beyond the scope of this paper.

Signal Connectivity
After sensors are installed, they must be connected to the data logging system. Signal connectivity describes the component of your measurement hardware with which you connect your sensors to your logging system. Screw terminals, with which you can connect bare wires from sensors directly to your logging system, are the most basic form of connections. Screw terminals are a good choice for general-purpose use, particularly when you need to connect a large number of signals into a small amount of space. The disadvantage of screw terminals is that they are time-consuming to connect and difficult to reconfigure. Figure 4 shows some other standard connectivity options that are designed to make connecting and disconnecting sensors less labor-intensive. Minithermocouple connectors are a widely used option for thermocouples. BNC and SMB connectors are commonly used when electrical shielding is required for noise immunity. Banana jacks are often used when measuring current, resistance, or higher voltages. The sensor provider typically defines the connector options available for a sensor, and it is up to you to choose measurement hardware that can accept that connector.

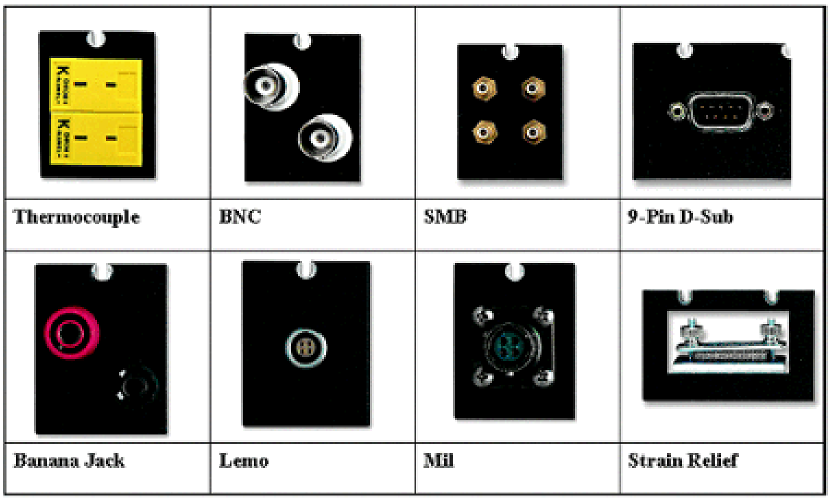


Figure 4. Examples of Signal Connectivity Options

Signal Conditioning
Signal conditioning is one of the most important, and most overlooked, components of a PC-based data logging system. Most signals require some form of preparation (conditioning) before they can be digitized. For example, thermocouples produce very low level signals that require amplification, filtering and linearization. Other sensors, such as RTDs, thermistors, strain gauges and accelerometers, require power in addition to amplification and filtering, while other signals may require isolation to protect the system from high voltages. Even pure voltage signals can require special technologies to block large common-mode signals or to attenuate high voltages for safe measurement. No single stand-alone datalogger can provide the flexibility required to make all of these measurements. However, with front-end signal conditioning, you can combine the necessary technologies to bring these various types of signals into a single PC-based data logging system.

Because of the vast array of signal conditioning technologies, the role and need for each technology can quickly become confusing. Below we list common types of signal conditioning, their functionality, and examples of when you need them.

Amplification - When the voltage levels you are measuring are very small, amplification is used maximize the effectiveness of your digitizer. By amplifying the input signal, the conditioned signal uses more of the effective range of the analog-to-digital converter (ADC) and you achieve better measurement accuracy. Typical sensors that require amplification are thermocouples and strain gauges.

Attenuation - Attenuation is the opposite of amplification. It is necessary when the voltages to be digitized are beyond the input range of the digitizer. This form of signal conditioning divides down the input signal so that the conditioned signal is within the range of the ADC. Attenuation is necessary for measuring voltages higher than 10 V.

Isolation - Voltage signals outside the range of the digitizer can damage the measurement system and harm the operator. For that reason, isolation is usually required in conjunction with attenuation to protect the system and the user from dangerous voltages or voltage spikes. Isolation may also be required when the sensor is on a different electrical ground plane from the measurement sensor (such as a thermocouple mounted on an engine).

Multiplexing - Typically, the digitizer is the most expensive part of a data acquisition system. Using multiplexing, you can automatically route several signals into a single digitizer to provide a cost-effective way to greatly expand the signal count of your system. Multiplexing is necessary for any high-channel-count application.

Filtering - Filtering is required to remove unwanted frequency components from a signal to prevent aliasing and reduce signal noise. Thermocouple measurements typically require a lowpass filter to remove power line noise from the signals. Vibration measurements normally require a higher frequency lowpass filter to remove high frequency signal components that are above the range of the acquisition system.

Excitation - Many sensor types, including RTDs, strain gauges, and accelerometers, require excitation – some form of power – to make a measurement. Excitation signals can be voltage or current, depending on the sensor type.

Linearization - Some types of sensors produce voltage signals that are not linearly related to the physical quantity they are measuring. Linearization is the process of interpreting the signal from the sensor as a physical measurement. This can be done either with signal conditioning or through software. Thermocouples are the classic example of a sensor that requires linearization.

Cold-Junction Compensation - Another technology that is required for thermocouple measurements is cold-junction compensation (CJC). Any time a thermocouple is connected to a data acquisition system, the temperature of the connection must be known in order to calculate the true temperature the thermocouple is measuring. A built-in CJC sensor must be present at the location of the connections.

Simultaneous Sampling - When it is critical to measure two or more signals at exactly the same time, simultaneous sampling is required. Front-end signal conditioning can provide a much more cost-effective simultaneous sampling solution than purchasing a digitizer with those capabilities. Typical applications that might require simultaneous sampling include vibration measurements and phase-difference measurements.

Most sensors require a combination of these signal conditioning technologies. Again, the thermocouple is the classic example because it requires amplification, linearization, cold-junction compensation, filtering, and sometimes isolation. Ideally, a good PC-based data logging platform should give you the ability to select the type of signal conditioning that is needed for your application. In some systems, front-end signal conditioning is an option, but in other systems, front-end signal conditioning is a necessity to make the required measurements. As a rule of thumb, your measurement system should include front-end signal conditioning if you are planning to use any of the following: thermocouples, RTDs, thermistors, strain gauges, LVDTs, accelerometers, switching, multiplexing, mixed low-voltage/high-voltage signals, current inputs, or resistance inputs.

Conversion
After physical parameters have been converted into electrical signals and properly conditioned, it is time to convert the analog electrical signals into digital values and passes those values back to the computer. The analog-to-digital conversion can be accomplished with either a plug-in data acquisition (DAQ) board, or it can be integrated into a single package with the conditioning and connectivity.

The combination of sensors, signal connectivity, signal conditioning, and analog-to-digital conversion makes up the measurement hardware portion of a data logging system. In a PC-based system, the measurement hardware is configured and controlled through software, and it is critical to use software that is designed to integrate smoothly with all components of your data logging system.

Online Analysis
The next functional component in a typical data logging system is online analysis. In PC-based systems, online analysis is accomplished through software. Many different forms of online analysis can be needed in various data logging applications. We will discuss some of the most common ones here.

Channel scaling is the conversion of the raw binary values returned by the acquisition system into properly scaled measurements with appropriate engineering units. One example is computing temperature (in degrees C) from a thermocouple reading. The digitizer returns binary measurements of the thermocouple voltage and the cold-junction sensor voltage. The software converts the binary measurements into voltages, and then uses a thermocouple conversion formula to compute temperature. Similar channel scaling routines are used for strain gauges, RTDs, accelerometers, and others. Fortunately, modern PC-based measurement software handles most scaling functions automatically.

Another important online analysis function is alarming and event management, which typically means monitoring a channel and providing some notification if limits are exceeded. This notification can be as basic as turning on a warning light, or as complex as paging someone with information about the problem. Alarming can also include an automated response to certain events. For example, a data logging system could shut down a machine being monitored if the oil temperature exceeded a certain limit.

A wide range of online analysis functionality can be required in different data logging applications. This functionality could include feedback control systems or advanced signal analysis. Only PC-based data logging systems have the flexibility to implement these differing requirements.

Logging and Storage

The logging (or storage) functional block is, by definition, required in every data logging system. Methods of storing data vary widely across different systems. Strip chart recorders use paper, traditional dataloggers can use internal nonvolatile memory, floppy disks, or a variety of other media. PC-based data logging systems typically use the hard drive of the PC, although they can also use tape drives, network drives, RAID drives, and other more exotic options.

Software is of critical importance in PC-based data logging systems, because well-written logging software determines how data is stored, how quickly data can be written to disk, and how efficiently disk space is used. Logging software also gives you data management capabilities, such as changing data formats, archiving data, and access to databases.

The data storage format has a strong link to the performance and ease of use of your data logging system. There are three general formats commonly used for storage in data logging systems – ASCII text files, binary files, and databases.

ASCII text files are the most common and flexible form of data storage. Text files for data logging applications are typically made up of a header section and columns of data. The header section gives information such as channel names, engineering units, test equipment, and user comments. The first data column is usually the timestamp of each sample, and it is followed by another column for each channel being logged. Text files are useful because they can be opened or imported into almost any software package, and they are easily transferred between operating systems. Some disadvantages of text files are that they use disk space inefficiently, and they require additional processing overhead to write and read from files. ASCII text files are commonly used when the speed of the acquisition is slow (<1000 samples/second), the total amount of data to log is not large, and the user needs to easily share data between different software applications.

Binary files are the most efficient method of data storage. With binary files, the raw bytes that the computer is using to store data in memory are written directly to the file. This data takes up considerably less space than the same information written in ASCII text format, and it requires much less processor overhead than formatting into text. Binary files cannot be viewed in common software applications such as MS Excel. Instead, they must be translated by a software routine into meaningful data. With PC-based data logging systems you can log scaled data that is already processed into correct engineering units, or you can log the raw binary values returned by the digitizer. The raw binary values representing the analog-to-digital conversions of each sample returned from a 16-bit DAQ device take up 16 bits, or two bytes, of memory. The channel scaling routines in your logging software automatically convert this raw data into a real number that represents the physical value you measured. Scaled data is typically handled inside your data logging software as a double precision floating-point value, which refers to a data type taking up eight bytes of memory on most computer systems.

For performance reasons, some high-speed data logging systems might log the raw binary values to disk, along with the necessary scaling constants to convert them to scaled data at a later time. Figure 5 shows the relationship between logging raw binary, scaled binary, and ASCII text. With binary files, you require less space and achieve greatly improved stream-to-disk speed. Raw binary files can be less than one tenth the size of a text file containing the same information. The disadvantage of binary files is that they typically must be translated to another format before they can be shared between different applications.

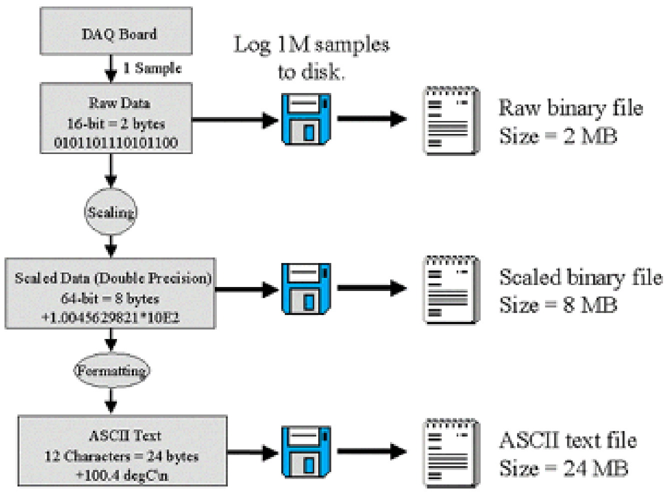


Figure 5. Data Storage File Size Example

Many data logging software packages log data into databases. Databases are typically binary files that provide a structured format for inserting and retrieving data. They are optimized for efficiently handling large amounts of data and for searching through information in the database without loading everything into memory. Databases also are often designed for easy backup and archiving of data and multiple-user access. They usually have software methods to make it easy to import data into different software packages for analysis and report generation. In many ways, databases are the ideal storage format for PC-based data logging systems. Two disadvantages of using databases for storage are that they add increased complexity, and they are difficult to implement if starting from scratch.

Many different storage media types are used for data logging. Stand-alone dataloggers can use onboard nonvolatile memory, floppy disks, PCMCIA memory cards, tapes, or a variety of other options. PC-based data logging systems usually rely on the internal PC hard drive, which is now possible because of the trend toward more reliable and higher capacity hard drives. The 20 GB (and larger) hard drives that are readily available today make hard drives one of the most economical storage devices. It is still advisable to periodically back up or archive data stored on a local hard drive.

High-speed data logging applications (more than 1Msamples/second) can start to exceed the write-to-disk speeds of normal PC hard drives. One of the advantages of PC-based logging systems is that you can move to more high-performance storage devices and higher performance computers, often with little or no modifications to your logging software or measurement hardware. One type of high-performance storage device is the RAID (redundant array of independent disks) controller. RAID controllers use multiple hard drives in concert to greatly enhance the combined stream-to-disk speed and to provide improved data integrity. Audio-visual (AV) drives are another type of storage device that is used for high-speed data logging. AV drives are optimized for streaming large amounts of audio and video information to disk, and this optimization also makes them well suited for high-performance data logging applications. Finally, some companies make custom hardware for streaming data from DAQ devices directly into a storage device via the PCI bus in the PC. The stream-to-disk rates of these devices are limited by the available bandwidth of the PCI bus, which has a theoretical maximum of 132 Mbytes/s on most computer systems.

Offline Analysis

Offline analysis is performing mathematical functions on data after it has been acquired in order to extract important information. Types of offline analysis can include computing basic statistics of measured parameters, as well as more advanced functions such as the frequency content of signals and order analysis. Offline analysis can be integrated with the rest of the data logging application, or it can occur separately through stand-alone analysis software packages. Often, offline analysis is combined with the report generation, historical display, and data sharing functions.

Display

Most data logging applications require some form of display to view the measurements that are being recorded. The display function can be further broken down to viewing live data and historical data. Live data display is necessary if you need to view data as it is being acquired. Many stand-alone dataloggers have a live data display integrated into the box with them. Historical display lets you view data that was previously acquired. Most stand-alone dataloggers require you to move the data to a PC for historical viewing. With PC-based data logging applications, you combine both live display and historical display into the same user interface. Data viewing utilities should provide an intuitive user interface, scrolling and zooming capabilities, cursors, and general customization features. Figure 6 is an example of a typical historical data display found with commercially available software.

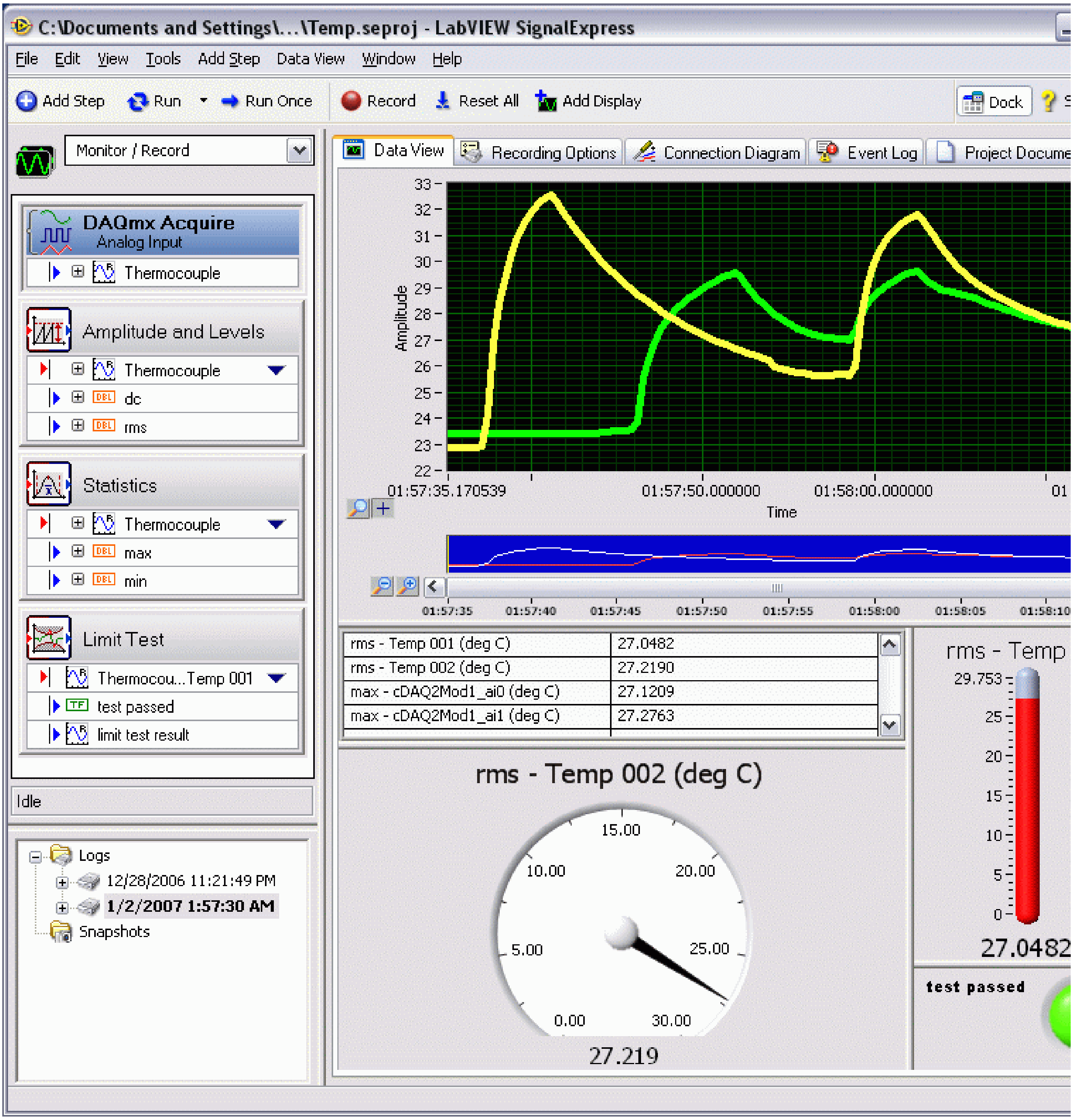


Figure 6. Example of Historical Data Display from National Instruments LabVIEW SignalExpress Software

Report Generation

Report generation is a function that is often not considered part of the data logging application. In reality, almost every data logging application requires some form of reporting capability, for the simple reason that if you're recording the data, somebody needs to see it in a presentable format. Report generation can be integrated into PC-based data logging applications for increased efficiency. The logging application can be set up to periodically generate specified reports and distribute them to the appropriate people. Powerful commercial software is available to give you advanced capabilities for analyzing data and generating reports from your measurements. Figure 7 shows an example of some of the report generation capabilities possible with commercially available packages. When choosing software for report generation, it is critical that it integrate smoothly with the rest of your data logging software. Ideally, the logging software should be able to pass data directly to the report generation application and trigger automatic reports.

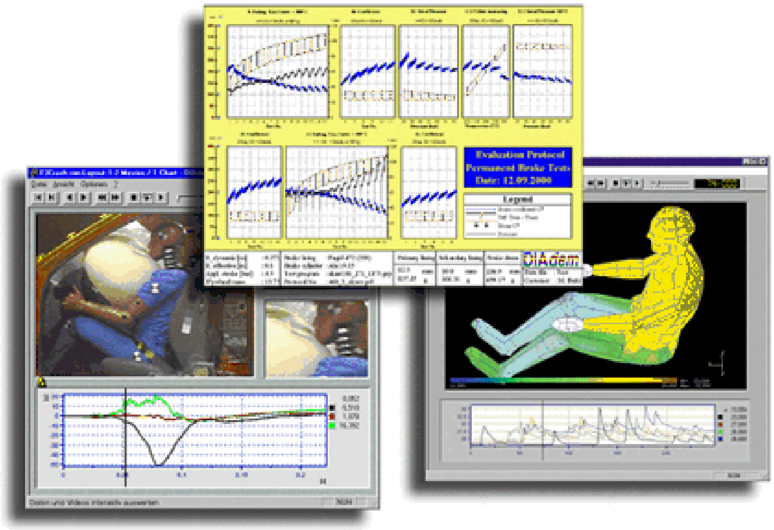


Figure 7. Advanced Report Generation Capabilities with National Instruments DIAdem Software

Data Sharing and Publishing

For data that has been logged to be useful, it must be available to the right people. With the networking capabilities found in modern data logging software, sharing data and publishing it to the network no longer requires a degree in computer science. Logging applications can be set up to publish live data to the network as it is acquired, periodically e-mail both raw data and analyzed results to key personnel, or automatically post reports to a Web page.

In widely distributed data logging applications, each logging node can publish its measurements to the network, and a main computer can serve as the central collection facility. The central computer retrieves the measurements from each node, combines them for further analysis, logs the results for permanent archiving, and periodically generates reports analyzing the data.

Data Logging Systems

Now that we've seen the functional components of a data logging system, let's examine how these components can be implemented in real systems. All PC-based data logging systems are made up of hardware and software. The measurement hardware handles the acquisition portion of the logging application, and the hardware choice defines channel count, sensor type, acquisition speed, and measurement accuracy. The measurement software, in addition to controlling the hardware, also handles the online analysis, logging, offline analysis, display, reporting, and data sharing.

Software Options

Choosing software is one of the most critical steps when defining a PC-based data logging system. Your logging system depends on software to give you a productive, flexible solution. The measurement software must be designed to integrate seamlessly with your hardware. In addition to the basic task of acquiring data and logging it to disk, your software should provide tools to handle configuration of measurement hardware, scaling of data from channels, and calibration of your system. The software should handle your entire application – including report generation, analysis, archiving, and sharing. There are two general categories of software that can be used for PC-based data logging applications – turnkey software, also known as configuration-based software, and application development environments.

Turnkey packages are ready-to-run data logging software applications that interface with your measurement hardware to acquire and log data. These applications provide a user-friendly environment for configuring your logging task and getting up and running quickly. A good configuration-based data logging software package should provide:

Intuitive user interface – The software configuration should be through a windows-based, menu driven interface with easily accessible help functions and tutorials.

Automatic data storage and archiving – One of the primary functions of any data logging software package is to handle the storage of the data. It should automatically store the data in an efficient manner, and the software should provide a method for backing up and archiving data.

Capability to export data – As a minimum, you should be able to export data to ASCII text files so you can import it into other packages. With more advanced data logging software packages, data can automatically be transferred into common databases and analysis programs.

Alarming and event management – The data logging software must provide the capability to handle alarms and events. This includes detecting whether a signal is over or under a limit, outside of a range, or inside of a range. If an alarm occurs, the software should offer a range of actions, such as e-mailing, paging, or performing some type of digital or analog output.

Display and trending tools – All turnkey logging software packages need to have a good interface for viewing both live and historical data so you can scroll through data, zoom-in on regions of interest, and see long-term trends in data.

A disadvantage of configuration-based applications is that, unless there is a method for customization, you are locked into the functionality provided by the manufacturer. If your measurement needs change and you need to add a different type of signal, you can be out of luck if your turnkey software cannot handle that measurement type. Also, if you want to integrate offline analysis, report generation, and network connectivity into your data logging application, a closed turnkey software application can make that difficult. On the other hand, there are turnkey software applications that provide methods for customization with popular application development tools. These customizable logging software packages provide the best of both worlds, so you can get up and running quickly with your logging application, but also have a method for integrating more advanced functionality at a later date.

Application development tools are the other option available for developing PC-based data logging systems. Development tools can range from text-based programming languages to graphical programming environments. Figure 8 is an example of the software code for a data logging application developed in a graphical programming environment. With development tools, you can build your own customized data logging application, which does exactly what you need. Using application development tools, you also have the ability to modify your application as your needs change, integrate customized analysis and reporting capabilities with your logging application, and fully automate your data logging system.

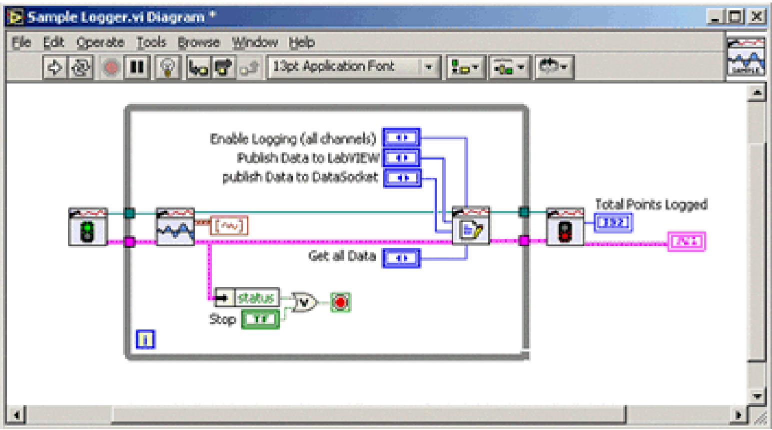


Figure 8. Graphical Programming Example of Data Logging Created with National Instruments LabVIEW

For developing data logging applications, you need to choose a development environment with all of the productivity features needed to create powerful PC-based logging systems. Some features to look for when evaluating application development tools are:

Wide range of graphical user interface components – Developing user interface components, such as graphs, displays, and controls, from scratch is extremely time-consuming. You should choose a development environment that contains high quality user interface components.

Tight integration with measurement hardware – It is critical that you use software designed to work with your measurement hardware. Not only does proper software integration result in significantly shorter development times, but also it helps ensure you get measurements you can trust.

Analysis Functions – One of the primary reasons for custom developing a data logging software application is to integrate advanced analysis functions. A good application development environment will provide a wide range of analysis functions to handle almost any requirement.

Network connectivity – In today's networked environment, the ability to connect your data logging application to the Web can be very important. Your application development software should provide tools to make publishing results to the network a trouble-free process.

Report Generation – Your application development environment should either generate reports automatically or programmatically control external report generation packages.

The choice between turnkey software and development tools depends on the complexity of your data logging application and the amount of customization required. With either choice, it is important to use a software vendor that specializes in connecting measurements to computers and that provides high quality service and support.

Hardware Options

Many different hardware platforms are available for data logging systems. The platform choice depends on your requirements for size, operating environment, and installation. Although the combinations are nearly endless, platforms for PC-based data logging can be broken down into four broad catagories – portable, desktop, rack mount/industrial, and distributed. One of the key benefits of PC-based data logging systems is that the same data logging software scales across all of these platforms.

Portable data logging solutions are needed in a variety of applications, such as in-vehicle data logging or field-testing of equipment. Portable PC-based solutions use laptop computers and measurement hardware designed to be easily portable. Figure 9 shows a portable, PC-based data logging system from National Instruments. The digitizer is a plug-in PCMCIA data acquisition card which cables to small, laptop sized boxes for signal conditioning and connectivity. Portable systems typically take advantage of either USB, Ethernet, or PCMCIA bus solutions.

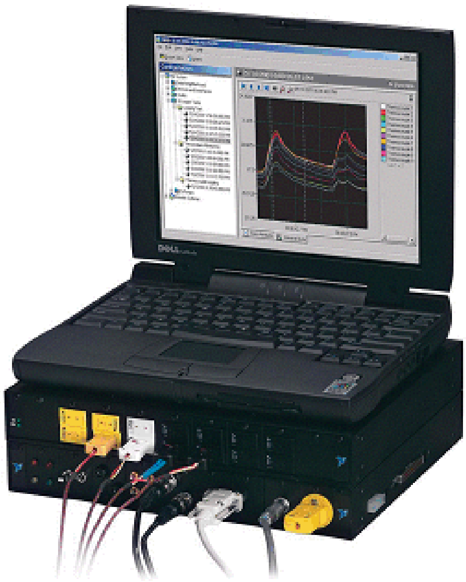


Figure 9. Portable PC-Based Data Logging System

Desktop systems, like the one shown in Figure 3, use measurement hardware designed to work with standard desktop PCs. Desktop systems are ideal for a wide range of laboratory-based data logging applications, such as validation testing of new product designs. Because fixed desktop systems are not as constrained by size, the signal connectivity and conditioning functions are typically accomplished by a modular front-end signal conditioning system with the capability to measure a wide range of sensor and signal types with easy expansion to log hundreds of channels.

Many times desktop systems take up too much space or do not fit well in environments such as large laboratories or manufacturing facilities. In these cases, the more compact and clean solution of a modular industrial PC, based on the PXI or CompactPCI standard, might be a more appropriate data logging solution. Figure 10 shows an example of a PXI-based data logging system. One modular system contains the PC, DAQ module, signal conditioning, and connectivity. These systems are designed to be rack-mountable, so they can be cleanly installed into an industrial or laboratory environment.

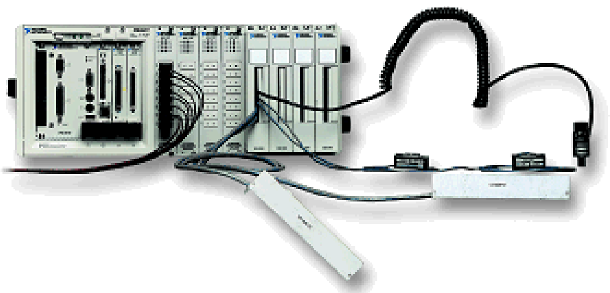


Figure 10. Rack-Mount PC-Based Industrial Data Logging System

Finally, some data logging systems need to be distributed away from the PC. This is the case when you need to log data from multiple locations around a facility, such as when logging the performance parameters of a chemical plant. Distributed logging systems should be compact so they can be mounted unobtrusively, and they typically must operate in extended temperature ranges. With distributed logging systems, you typically have multiple measurement nodes that communicate back to a central computer through a communications link such as RS-485 or Ethernet. Figure 11 is an example of a distributed logging system.

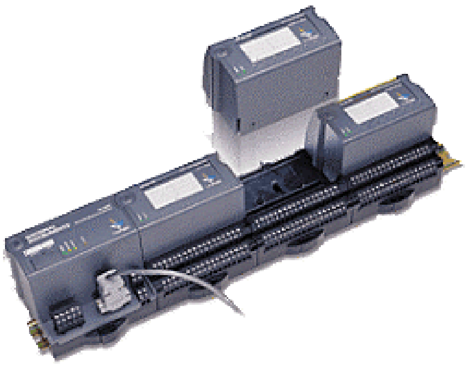


Figure 11. Distributed Data Logging System

The choice of logging platform depends on the requirements of your data logging system, and some systems might require using multiple platforms together. With properly designed hardware and software, data logging systems can scale from simple, low-channel-count laboratory systems up to very high-channel-count, distributed industrial logging systems.

Conclusions and Related Information

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

Related Information

See Also:
National Instruments Data Logging Main Page

Reader Comments | Submit a comment »

Learned that NI defines High Speed Logging as 1 MB/sec. Paper does not have a table indicating whether USB NI devices can be used for 1 MB/sec. Please provide a high throughput VI example for USB devices such as 6015 and fieldpoint
- *Victor Villavicencio, Northrop Grumman.* victorvilla93@hotmail.com - Mar 17, 2006

-
please help more in data loggings advantages and disadvantages and also description
- *mohamed, -.* itssaid@hotmail.com - Feb 19, 2006

told me that what I want to do is possible
- *Ronald Hoffman, Hoffman Tool & Die Inc..* rhoffman@hoffmantool.com - Jul 12, 2003

Legal
This tutorial (this "tutorial") was developed by National Instruments ("NI"). Although technical support of this tutorial may be made available by National Instruments, the content in this tutorial may not be completely tested and verified, and NI does not guarantee its quality in any way or that NI will continue to support this content with each new revision of related products and drivers. THIS TUTORIAL IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND AND SUBJECT TO CERTAIN RESTRICTIONS AS MORE SPECIFICALLY SET FORTH IN NI.COM'S TERMS OF USE (http://ni.com/legal/termsofuse/unitedstates/us/).