Data Acquisition (DAQ) Fundamentals

Introduction

Today, most scientists and engineers are using personal computers with PCI, PXI/CompactPCI, PCMCIA, USB, IEEE1394, ISA, or parallel or serial ports for data acquisition in laboratory research, test and measurement, and industrial automation. Many applications use plug-in boards to acquire data and transfer it directly to computer memory. Others use DAQ hardware remote from the PC that is coupled via parallel or serial port. Obtaining proper results from a PC-based DAQ system depends on each of the following system elements (see Figure 1).

- The personal computer
- Transducers
- Signal conditioning
- DAQ hardware
- Software

This application note gives an overview of each of these elements and explains the most important criteria of each element. The application note also defines much of the terminology common to each of the elements of a PC-based DAQ system.

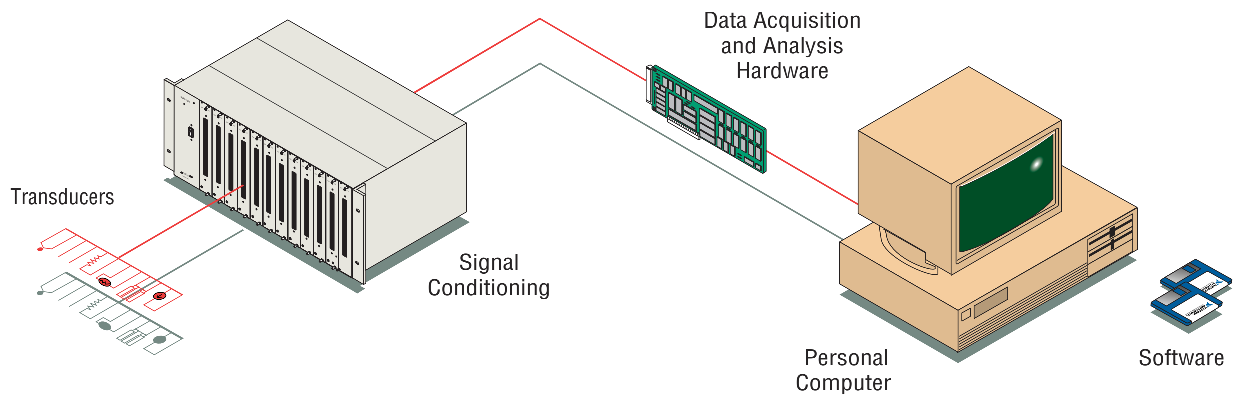


Figure 1. The Typical PC-Based DAQ System

The Personal Computer

The computer used for your data acquisition system can drastically affect the maximum speeds at which you are able to continuously acquire data. Today's technology boasts Pentium and PowerPC class processors coupled with the higher performance PCI bus architecture as well as the traditional ISA bus and USB. With the advent of PCMCIA, portable data acquisition is rapidly becoming a more flexible alternative to desktop PC based data acquisition systems. For remote data acquisition applications that use RS-232 or RS-485 serial communication, your data throughput will usually be limited by the serial communication rates.

The data transfer capabilities of the computer you use can significantly affect the performance of your DAQ system. All PCs are capable of programmed I/O and interrupt transfers. DMA transfers, not available on some computers, increases the system throughput by using dedicated hardware to transfer data directly into system memory. Using this method, the processor is not burdened with moving data and is therefore free to engage in more complex processing tasks. To reap the benefits of DMA or interrupt transfers, the DAQ board you choose must also be capable of making these types of transfers.

The limiting factor for acquiring large amounts of data is often the hard drive. Disk access time and hard drive fragmentation can significantly reduce the maximum rate at which data can be acquired and streamed to disk. For systems that need to acquire high-frequency signals, you should select a high-speed hard drive for your PC and make sure that there is enough contiguous (unfragmented) free disk space to hold the data.

Applications requiring real-time processing of high-frequency signals need a high-speed, 32-bit processor with its accompanying coprocessor, or a dedicated plug-in processor such as a digital signal processing (DSP) board. If the application only acquires and scales a reading once or twice a second, however, a low-end PC can be satisfactory.

You must also look ahead to determine which operating system and computer platform will yield the greatest long term return on investment and still able to meet your short term goals. Factors that may influence your choice may include the experience and needs of both your developers and end users, other uses for the PC both now and in the future, cost constraints, and the availability of different computers with respect to your implementation time frame. Traditional platforms include Mac OS, which is known for its simple graphical user interface, and Windows 3.x. Windows 9x, which boasts a much improved user interface over Windows 3.x, also offers the option of Plug and Play hardware configuration. In addition, Windows NT 4.0 offers a more robust 32-bit OS with the look and feel of Windows 9x.

Transducers

Transducers sense physical phenomena and provide electrical signals that the DAQ system can measure. For example, thermocouples, RTDs, thermistors, and IC sensors convert temperature into an analog signal that an ADC can measure. Other examples include strain gauges, flow transducers, and pressure transducers, which measure force, rate of flow, and pressure, respectively. In each case, the electrical signals produced are proportional to the physical parameters they are monitoring.

Signal Conditioning

The electrical signals generated by the transducers must be optimized for the input range of the DAQ board. Signal conditioning accessories can amplify low-level signals, and then isolate and filter them for more accurate measurements. In addition, some transducers require voltage or current excitation to generate a voltage output. Figure 2 depicts a typical DAQ system with SCXI signal conditioning from National Instruments.

Amplification – The most common type of conditioning is amplification. Low-level thermocouple signals, for example, should be amplified to increase the resolution and reduce noise. For the highest possible accuracy, the signal should be amplified so that the maximum voltage range of the conditioned signal equals the maximum input range of the analog-to-digital converter (ADC).

For example, SCXI has several signal conditioning modules that amplify input signals. The gain is applied to the low-level signals within the SCXI chassis located close to the transducers, sending only high-level signals to the PC, minimizing the effects of noise on the readings.

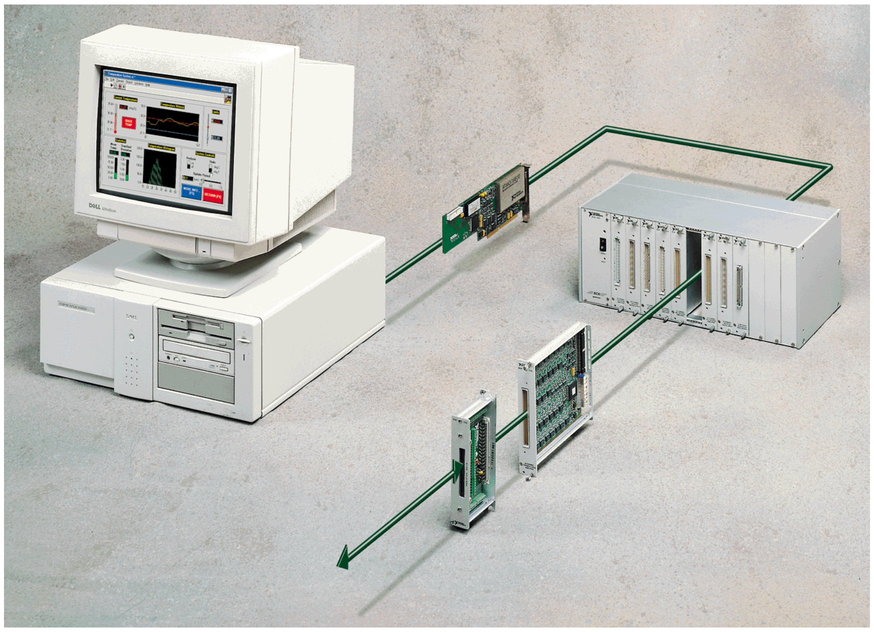


Figure 2. The SCXI Signal Conditioning Front-End System for Plug-In DAQ Boards

Isolation – Another common application for signal conditioning is to isolate the transducer signals from the computer for safety purposes. The system being monitored may contain high-voltage transients that could damage the computer.

An additional reason for needing isolation is to make sure that the readings from the plug-in DAQ board are not affected by differences in ground potentials or common-mode voltages. When the DAQ board input and the signal being acquired are each referenced to “ground,” problems occur if there is a potential difference in the two grounds. This difference can lead to what is known as a ground loop, which may cause inaccurate representation of the acquired signal, or if too large, may damage the measurement system. Using isolated signal conditioning modules will eliminate the ground loop and ensure that the signals are accurately acquired. For example, the SCXI-1120 and SCXI-1121 modules provide isolation up to 250 Vrms of common-mode voltage. The SCXI-1122 module provides up to 450 Vrms isolation.

Multiplexing – A common technique for measuring several signals with a single measuring device is multiplexing. Signal conditioning devices for analog signals often provide multiplexing for use with slowly changing signals such as temperature. This is in addition to any built-in multiplexing on the DAQ board. The ADC samples one channel, switches to the next channel, samples it, switches to the next channel, and so on. Because the same ADC is sampling many channels instead of one, the effective sampling rate of each individual channel is inversely proportional to the number of channels sampled. Our SCXI modules for analog signals employ multiplexing so that as many as 3072 signals can be measured with one DAQ board. With the AMUX-64T analog multiplexer, you can measure up to 256 signals with a single board.

Filtering – The purpose of a filter is to remove unwanted signals from the signal that you are trying to measure. A noise filter is used on DC-class signals such as temperature to attenuate higher frequency signals that can reduce the accuracy of your measurement. For example, many of the SCXI modules have 4 Hz and 10 kHz lowpass filters to eliminate noise before the signals are digitized by the DAQ board.

AC-class signals such as vibration often require a different type of filter known as an antialiasing filter. Like the noise filter, the antialiasing filter is also a lowpass filter; however, it must have a very steep cutoff rate, so that it almost completely removes all frequencies of the signal that are higher than the input bandwidth of the board. If the signals were not removed, they would erroneously appear as signals within the input bandwidth of the board. Boards designed specifically for AC-class signal measurement – the PCI-4451, PCI-4452, NI 4551, and NI 4552 dynamic signal acquisition boards – and the SCXI-1141 module have antialiasing filters built into them.

Excitation – Signal conditioning also generates excitation for some transducers. Strain gauges, thermistors, and RTDs, for example, require external voltage or current excitation signals. Signal conditioning modules for these transducers usually provide these signals. RTD measurements are usually made with a current source that converts the variation in resistance to a measurable voltage. Strain gauges, which are very low-resistance devices, typically are used in a Wheatstone bridge configuration with a voltage excitation source. The SCXI-1121 and SCXI-1122 have onboard excitation sources, configurable as current or voltage, that you can use for strain gauges, thermistors, or RTDs.

Linearization – Another common signal conditioning function is linearization. Many transducers, such as thermocouples, have a nonlinear response to changes in the phenomena being measured. Both our NI-DAQ driver software and LabVIEW, LabWindows/CVI, ComponentWorks, and VirtualBench application software include linearization routines for thermocouples, strain gauges, and RTDs.

It is important to understand the nature of your signal, the configuration that is being used to measure the signal and the affects of the surrounding environment. Based on this information you can easily determine whether signal conditioning will be a necessary part of your DAQ system.

DAQ Hardware

Analog Inputs

Basic Considerations of Analog Inputs – The analog input specifications can give you information on both the capabilities and the accuracy of the DAQ product. Basic specifications, which are available on most DAQ products, tell you the number of channels, sampling rate, resolution, and input range. The number of analog channel inputs will be specified for both single-ended and differential inputs on boards that have both types of inputs. Single-ended inputs are all referenced to a common ground point. These inputs are typically used when the input signals are high level (greater than 1 V), the leads from the signal source to the analog input hardware are short (less than 15 ft.), and all input signals share a common ground reference. If the signals do not meet these criteria, you should use differential inputs. With differential inputs, each input has its own ground reference. Noise errors are reduced because the common-mode noise picked up by the leads is canceled out.

Sampling Rate – This parameter determines how often conversions can take place. A faster sampling rate acquires more points in a given time and can therefore often form a better representation of the original signal. For example, audio signals converted to electrical signals by a microphone commonly have frequency components up to 20 kHz. To properly digitize this signal for analysis, the Nyquist sampling theorem tells us that we must sample at more than twice the rate of the maximum frequency component we want to detect. So, a board with a sampling rate greater than 40 kS/s is needed to properly acquire this signal.

Multiplexing – A common technique for measuring several signals with a single ADC is multiplexing. The ADC samples one channel, switches to the next channel, samples it, switches to the next channel, and so on. Because the same ADC is sampling many channels instead of one, the effective rate of each individual channel is inversely proportional to the number of channels sampled. For example, a PCI-MIO-16E-1 sampling at 1 MS/s on 10 channels will effectively sample each individual channel at:

$$\frac{1\text{ MS/s}}{10\text{ channels}} = 100\text{ kS/s per channel}$$

Resolution – The number of bits that the ADC uses to represent the analog signal is the resolution. The higher the resolution, the higher the number of divisions the range is broken into, and therefore, the smaller the detectable voltage change. Figure 3 shows a sine wave and its corresponding digital image as obtained by an ideal 3-bit ADC. A 3-bit converter (which is actually seldom used but a convenient example) divides the analog range into 23, or 8 divisions.

Each division is represented by a binary code between 000 and 111. Clearly, the digital representation is not a good representation of the original analog signal because information has been lost in the conversion. By increasing the resolution to 16 bits, however, the number of codes from the ADC increases from 8 to 65,536, and you can therefore obtain an extremely accurate digital representation of the analog signal if the rest of the analog input circuitry is designed properly.

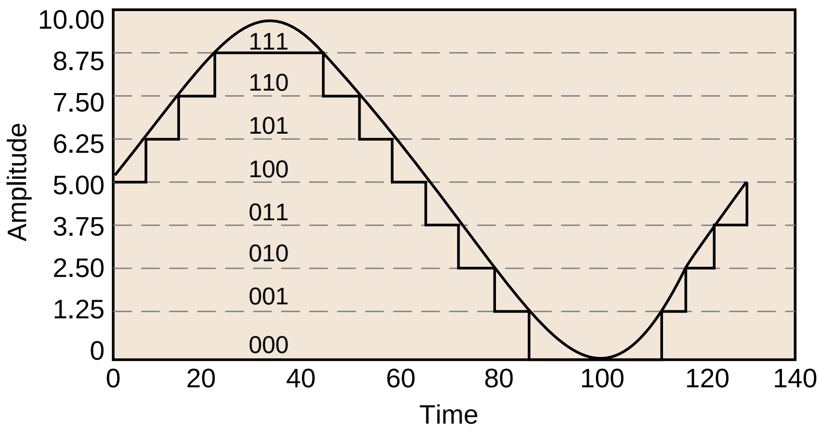


Figure 3. Digitized Sine Wave with a Resolution of Three Bits

Range – Range refers to the minimum and maximum voltage levels that the ADC can quantize. The multifunction DAQ boards offer selectable ranges so that the board is configurable to handle a variety of different voltage levels. With this flexibility, you can match the signal range to that of the ADC to take best advantage of the resolution available to accurately measure the signal.

The range, resolution, and gain available on a DAQ board determine the smallest detectable change in voltage. This change in voltage represents 1 LSB of the digital value, and is often called the code width. The ideal code width is found by dividing the voltage range by the gain times two raised to the order of bits in the resolution. For example, one of our 16-bit multifunction DAQ boards, the AT-MIO-16X, has a selectable range of 0 to 10 or –10 to 10 V and selectable gain of 1, 2, 5, 10, 20, 50, or 100. With a voltage range of 0 to 10 V, and a gain of 100, the ideal code width is:

$$\frac{10}{100 \times 2^{16}} = 1.5\mu V$$

Therefore, the theoretical resolution of one bit in the digitized value is $1.5\mu\dot{V}$.

Critical Considerations of Analog Inputs – Although the basic specifications previously described may show that a DAQ board has a 16-bit resolution ADC and a 100 kS/s sampling rate, this does not mean you can sample at full speed on all 16 channels and get full 16-bit accuracy. For example, you can purchase products on the market today with 16-bit ADCs and get less than 12 bits of useful data. How do you tell if the board that you are considering will give you the desired results? The most important thing to do is to scrutinize specifications that go beyond the resolution of the DAQ product. National Instruments offers several application notes to help you understand all the specifications on DAQ products. While evaluating DAQ products, also consider the DNL, relative accuracy, settling time of the instrumentation amplifier, and noise, because what you don't know about the DAQ board you are considering can hurt your measurements.

Ideally, as you increase the level of voltage applied to a DAQ board, the digital codes from the ADC should also increase linearly. If you were to plot the voltage versus the output code from an ideal ADC, the plot would be a straight line. Deviations from this ideal straight line are specified as the nonlinearity. DNL is a measure in LSB of the worst-case deviation of code widths from their ideal value of 1 LSB. An ideal DAQ board has a DNL of 0 LSB. Practically, a good DAQ board will have a DNL of ±0.5 LSB.

There is no upper limit on how wide a code can be. Codes do not have widths of less than 0 LSB, so the DNL is never worse than –1 LSB. A DAQ board with poor performance may have a code width equal to or very near zero, which indicates a missing code. No matter what voltage you input to the DAQ board with a missing code, the board will never quantize the voltage to the value represented by this code. Sometimes DNL is specified by stating that a DAQ board has no missing codes, which means that the DNL is bounded below by –1 LSB but does not make any specifications about the upper boundaries. All National Instruments E Series boards are guaranteed to have no missing codes, and our specifications clearly state what the DNL is in the specification so that you know the accuracy or our board linearity.

If the DAQ board in the previous example, which had a code width of 1.5 µV, had a missing code slightly above 500 µV, then increasing the voltage to 502 µV would not be detectable. Only when the voltage is increased another LSB, or in this example, beyond 503 µV, will the voltage change be detectable. As you can see, poor DNL reduces the resolution of the board.

Relative Accuracy – It is a measure in LSBs of the worst-case deviation from the ideal DAQ board transfer function, a straight line. Relative accuracy is determined on a DAQ board by connecting a voltage at negative full scale, digitizing the voltage, increasing the voltage, and repeating the steps until the input range of the board has been covered. When the digitized points are plotted, the result will be an apparent straight line (see Figure 4a). However, you can subtract actual straight-line values from the digitized values and plot these resulting points, as shown in Figure 4b. The maximum deviation from zero is the relative accuracy of the DAQ board.

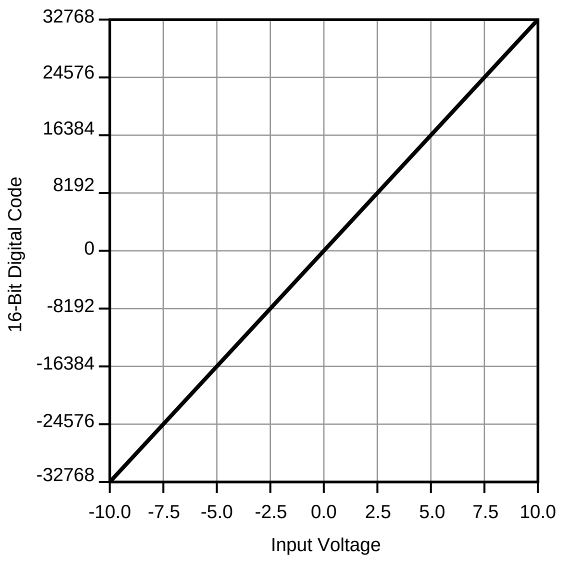


Figure 4a

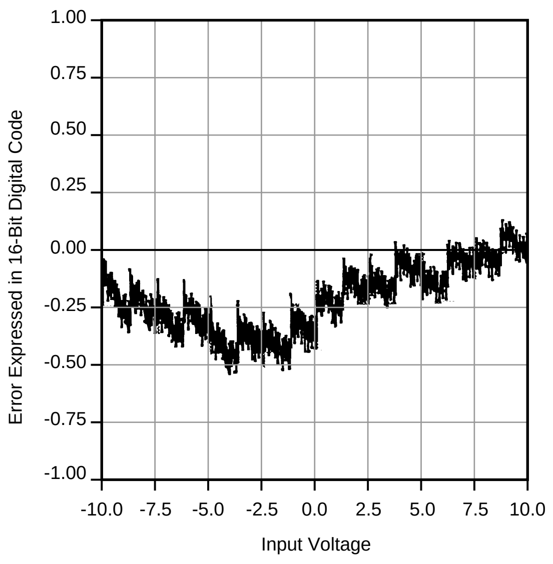


Figure 4b

Figure 4. Determining the relative accuracy of a DAQ board. Figure 4a shows the apparent straight-line plot generated by sweeping the input. Figure 4b shows, by subtracting out calculated straight-line values, that the plot is not actually straight.

The driver software for a DAQ board will translate the binary code value of the ADC to voltage by multiplying it by a constant. Good relative accuracy is important for a DAQ board because it ensures that the translation from the binary code of the ADC to the voltage value is accurate. Obtaining good relative accuracy requires that both the ADC and the surrounding analog circuitry are designed properly.

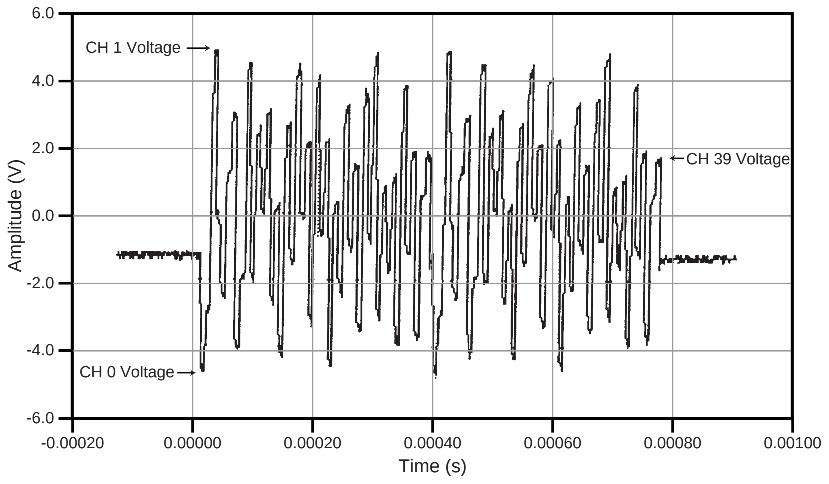


Figure 5. The input to an instrumentation amplifier that is multiplexing 40 DC signals appears to be a high-frequency AC signal.

The instrumentation amplifier is most likely not to settle when you are sampling several channels at high gains and high rates. Under such conditions, the instrumentation amplifier has difficulty tracking large voltage differences that can occur as the multiplexer switches between input signals. Typically, the higher the gain and the faster the channel switching time, the less likely it is that the instrumentation amplifier will settle. In fact, no off-the-shelf programmable-gain instrumentation amplifier can settle to 12-bit accuracy in less than 2 µs when amplifying at a gain of 100. That is why National Instruments developed the NI-PGIA specifically for DAQ board applications – so our boards that use the NI-PGIA can consistently settle at high gains and sampling rates.

Noise – Any unwanted signal that appears in the digitized signal of the DAQ board is noise. Because the PC is a noisy digital environment, acquiring data on a plug-in board takes a very careful layout on multilayer DAQ boards by skilled analog designers. Simply placing an ADC, instrumentation amplifier, and bus interface circuitry on a one or two-layer board will most likely result in a very noisy DAQ board. Designers can use metal shielding on a DAQ board to help reduce noise. Proper shielding should not only be added around sensitive analog sections on a DAQ board, but must also be built into the layers of the DAQ board with ground planes.

Figure 6 shows the DC noise plot of two DAQ products, both of which use the same ADC. Two qualities of the DAQ board can be determined from the noise plots – range of noise and the distribution. The plot in Figure 6a, which is our AT-MIO-16X, has a high distribution of samples at 0 and a very small number of points occurring at other codes. The distribution is Gaussian, which is what is expected from random noise. From the plot, the peak noise level is within ±3 LSB. The plot in Figure 6b, made with a very noisy DAQ product from a competitor, has a far different distribution. It has noise greater than 20 LSB, with many samples occurring at points other than the expected value. For the DAQ products in Figure 5, the test was run with an input range of ±10 V and a gain of 10. Therefore, 1 LSB = 31 µV, so a noise level of 20 LSB is equivalent to 620 µV of noise.

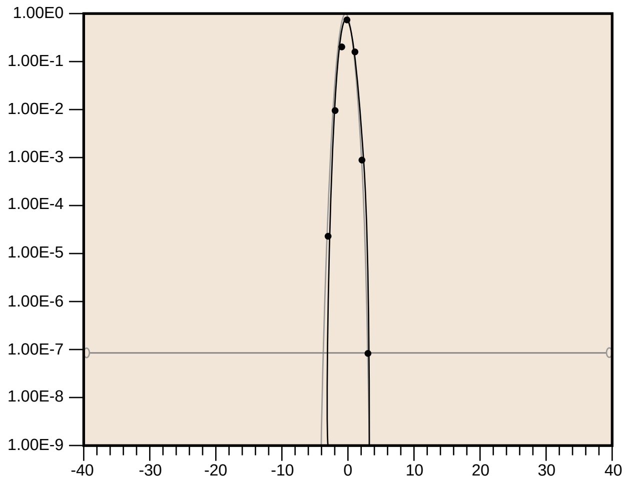


Figure 6a

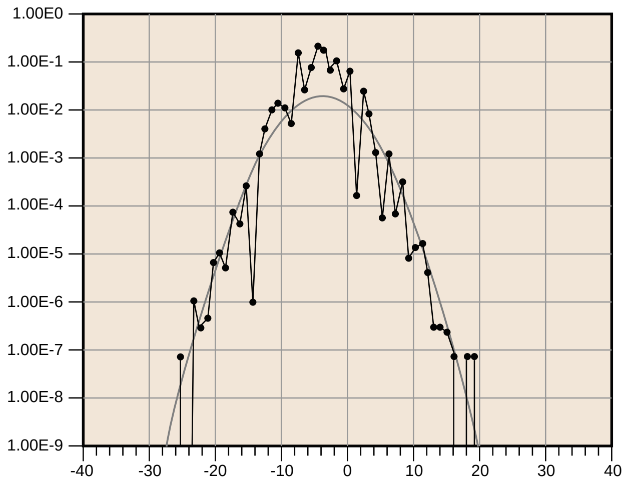


Figure 6b

Figure 6. Noise plots of two DAQ products with significantly different noise performance even though they use the same 16-bit ADC. Figure 6a. is the National Instruments AT-MIO-16X; Figure 6b is a competitor's DAQ product.

Evaluate the Specifications – With a sophisticated measuring device like a plug-in DAQ board, you can get significantly different accuracies depending on whose board you are using. National Instruments goes to great lengths to make our boards extremely accurate, in many cases more accurate than stand-alone instruments. We publish this accuracy in our specifications. Be leery of boards that are inadequately specified, because the specification omitted may be the one that causes your measurements to be inaccurate. By evaluating more analog input specifications than simply the resolution of the ADC converter, you can make sure that you are getting a DAQ product that is accurate enough for your application.

Analog Outputs

Analog output circuitry is often required to provide stimuli for a DAQ system. Several specifications for the digital-to-analog converter (DAC) determine the quality of the output signal produced – settling time, slew rate, and resolution. Settling time and slew rate work together in determining how fast the DAC can change the level of the output signal. Settling time is the time required for the output to settle to the specified accuracy. The settling time is usually specified for a full-scale change in voltage. The slew rate is the maximum rate of change that the DAC can produce on the output signal. Therefore, a DAC with a small settling time and a high slew rate can generate high-frequency signals, because little time is needed to accurately change the output to a new voltage level.

An example of an application that requires high performance in these parameters is the generation of audio signals. The DAC requires a high slew rate and small settling time to generate the high frequencies necessary to cover the audio range. In contrast, an example of an application that does not require fast D/A conversion is a voltage source that controls a heater. Because the heater cannot respond quickly to a voltage change, fast D/A conversion is not necessary. The application will determine the DAC specifications.

Output resolution is similar to input resolution. It is the number of bits in the digital code that generates the analog output. A larger number of bits reduces the magnitude of each output voltage increment, thereby making it possible to generate smoothly changing signals. Applications requiring a wide dynamic range with small incremental voltage changes in the analog output signal may need high-resolution voltage outputs.

Triggers

Many DAQ applications need to start or stop a DAQ operation based on an external event. Digital triggers synchronize the acquisition and voltage generation to an external digital pulse. Analog triggers, used primarily in analog input operations, start or stop the DAQ operation when an input signal reaches a specified analog voltage level and slope polarity.

Real-Time System Integration (RTSI™)

The National Instruments expertise in instrumentation led to the development of the RTSI bus for our DAQ products. The RTSI bus uses a custom gate array and a ribbon cable to route timing and trigger signals between multiple functions on one DAQ board, or between two or more boards. With RTSI, you can synchronize A/D conversions, D/A conversions, digital inputs, digital outputs, and counter/timer operations. For example, with RTSI, two analog input boards can capture data simultaneously while a third board generates an output pattern synchronized to the sampling rate of the inputs.

Digital I/O

DIO interfaces are often used on PC DAQ systems to control processes, generate patterns for testing, and communicate with peripheral equipment. In each case, the important parameters include the number of digital lines available, the rate at which you can accept and source digital data on these lines, and the drive capability of the lines. If the digital lines are used for controlling events such as turning on and off heaters, motors, or lights, a high data rate is usually not required because the equipment cannot respond very quickly. The number of digital lines, of course, needs to match the number of processes that are controlled. In each of these examples, the amount of current required to turn the devices on and off must be less than the available drive current from the board.

With the proper digital signal conditioning accessories, however, you can use the low-current TTL signals to/from the DAQ hardware to monitor/control high voltage and current signals from industrial hardware. For example, the voltage and current needed to open and close a large valve may be on the order of 100 VAC at 2 A. Because the output of a digital I/O board is 0 to 5 VDC at several milliamperes, an SSR Series, ER-8/16, SC-206X Series, or SCXI module is needed to switch the power signal to control the valve.

A common application is to transfer data between a computer and equipment such as data loggers, data processors, and printers. Because this equipment usually transfers data in one byte (8-bit) increments, the digital lines on a plug-in digital I/O board are arranged in groups of eight. In addition, some boards with digital capabilities will have handshaking circuitry for communication synchronization purposes. The number of channels, data rate, and handshaking capabilities are all important specifications that should be understood and matched to the application.

Timing I/O

Counter/timer circuitry is useful for many applications, including counting the occurrences of a digital event, digital pulse timing, and generating square waves and pulses. You can implement all of these applications using three counter/timer signals – gate, source, and output. The gate is a digital input that is used to enable or disable the function of the counter. The source is a digital input that causes the counter to increment each time it toggles, and therefore provides the timebase for the operation of the counter. Finally, the output generates digital square waves and pulses at the output line.

The most significant specifications for operation of a counter/timer are the resolution and clock frequency. The resolution is the number of bits the counter uses. A higher resolution simply means that the counter can count higher. The clock frequency determines how fast you can toggle the digital source input. With higher frequency, the counter increments faster and therefore can detect higher frequency signals on the input and generate higher frequency pulses and square waves on the output. The DAQ-STC counter/timer used on our E Series DAQ boards, for example, has 16 and 24-bit counters with a clock frequency of 20 MHz. The DAQ-STC is a National Instruments custom ASIC designed specifically for DAQ applications. In comparison with the off-the-shelf counter/timer chips generally used on DAQ boards, the DAQ-STC is in a league of its own. For example, the DAQ-STC is an up/down counter/timer, meaning that it can use additional external digital signals to count up or down, depending on whether the level is high or low. This type of counter/timer can measure positioning from rotary or linear encoders. Other special functions include buffered pulse-train generation, timing for equivalent time sampling, relative time stamping, and instantaneous changing of sampling rate.

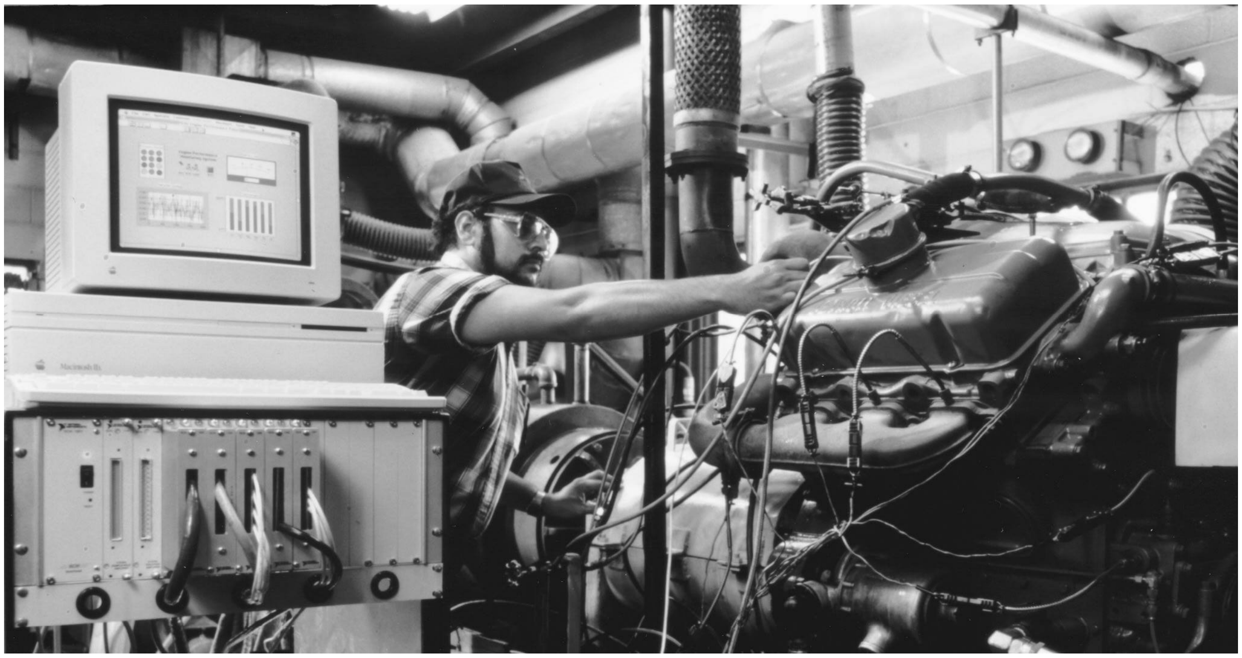


Figure 7. Automobile Lubricant Test Application showing SCXI Chassis and LabVIEW running on a Macintosh

Software

Software transforms the PC and DAQ hardware into a complete DAQ, analysis, and display system. DAQ hardware without software is useless – and DAQ hardware with poor software is almost useless. The majority of DAQ applications use driver software. Driver software is the layer of software that directly programs the registers of the DAQ hardware, managing its operation and its integration with the computer resources, such as processor interrupts, DMA, and memory. Driver software hides the low-level, complicated details of hardware programming, providing the user with an easy-to-understand interface.

For example, the code that follows shows NI-DAQ function calls used in C to read and scale a voltage from an analog input channel of an MIO-16E-10.

```
main()                     /* Program to read and scale an analog input */
{
       int    brd,         /* Which board to read analog value from */
              chan,        /* Analog input channel to read value from */
              gain,        /* Software-programmable gain to use on channel */
              reading;     /* Binary result of A/D conversion */
       Double voltage;     /* Voltage value at input channel after scaling */
       brd = 1;            /* Read from board 1, */
       chan = 3;           /* channel 3, */
```

```
        gain = 100;         /* with gain of 100 */
        AI_Read(brd, chan, gain, &reading);        /* Take a reading */
        AI_Scale(brd, gain, reading, &voltage);    /* Scale to voltage */
        printf("\nThe voltage is %lf volts", voltage);
}
```

The increasing sophistication of DAQ hardware, computers, and software continues to emphasize the importance and value of good driver software. Properly selected driver software can deliver an optimal combination of flexibility and performance, while also significantly reducing the time required to develop the DAQ application.

While selecting driver software, there are several factors to consider.

Which Functions Are Available? – Driver functions for controlling DAQ hardware can be grouped into analog I/O, digital I/O, and timing I/O. Although most drivers will have this basic functionality, you will want to make sure that the driver can do more than simply get data on and off the board. Make sure the driver has the functionality to:

- Acquire data at specified sampling rates
- Acquire data in the background while processing in the foreground
- Use programmed I/O, interrupts, and DMA to transfer data
- Stream data to and from disk
- Perform several functions simultaneously
- Integrate more than one DAQ board
- Integrate seamlessly with signal conditioning equipment

These and other functions of the DAQ driver, which are included in the National Instruments NI-DAQ driver software, can save the user a considerable amount of time.

Which Operating Systems Can You Use with the Driver? – Make sure that the driver software is compatible with the operating systems you plan to use now and in the future. The driver should also be designed to capitalize on the different features and capabilities of the OS. For example, while drivers written for Windows 3.x may run under Windows 95, only drivers written in full 32-bit code for Windows 95 can take advantage of the increased performance and robustness available with Windows 95. Drivers for Windows 95 should also be able to work together with Windows 95 Plug and Play to ensure that your system is easy to set up and configure. You may also need the flexibility to port your code easily between platforms, say from a Windows PC to a Macintosh or a Sun SPARCstation. NI-DAQ driver software is available for Windows 95/NT/3.1, DOS, Mac OS, and Sun Solaris.

NI-DAQ protects your software investment because you can switch between hardware products or operating systems with little or no modification to your application

Which Programming Languages Can You Use with the Driver? – Make sure that the driver can be called from your favorite programming language, and is designed to work well within that development environment. A programming language such as Visual Basic, for example, has an event-driven development environment that uses controls for developing the application. If you are developing in the Visual Basic environment, you need to make sure that the driver has custom controls, such as those in NI-DAQ, to match the methodology of the programming language.

Are the Hardware Functions You Need Accessible in Software? – A problem occurs when a developer purchases DAQ hardware, then combines the hardware with software, only to find that a required hardware feature is not handled by the software. The problem occurs most frequently when the hardware and software are developed by different companies. By asking this question, you can save yourself time searching through the software manuals looking for a function that does not exist. NI-DAQ, which is developed at National Instruments along with our DAQ hardware, handles every function listed on the data sheets of our DAQ hardware.

Does the Driver Limit Performance? – Because the driver is an additional layer, it may cause some performance limitations. In addition, operating systems such as Windows 3.1 can have significant interrupt latencies. If not dealt with properly, these latencies can greatly reduce the performance of the DAQ system. NI-DAQ is a high-performance driver that has code written specifically to reduce the interrupt latencies of Windows and to provide acquisition rates up to 1 MS/s.

The answers to these questions will give you an indication of the effort that has gone into developing the driver software. Ideally, you want to get your driver software from a company that has as much expertise in the development of the DAQ software as they do in the development of DAQ hardware.

Application Software – An additional way to program DAQ hardware is to use application software. But even if you use application software, it is important to know the answers to the previous questions, because the application software will use driver software to control the DAQ hardware. Application software adds analysis and presentation capabilities to the driver software. Application software also integrates instrument control (GPIB, RS-232, and VXI) with data acquisition. National Instruments offers LabWindows/CVI, application software for the traditional C programmer, and LabVIEW, application software with graphical programming methodology, for developing complete instrumentation, acquisition, and control applications. Both products can be augmented with add-on toolkits for special functionality. ComponentWorks gives complete instrumentation capabilities to Visual Basic through OLE custom controls. For the spreadsheet user, Measure offers DAQ capabilities directly within the Windows Excel environment. For the Windows DAQ user who wants ready-to-run virtual instruments, VirtualBench offers an oscilloscope, dynamic signal analyzer, function generator, DMM, and data logger. For analysis of the data you collect, we offer HiQ with extensive numerical analysis and visualization capabilities.

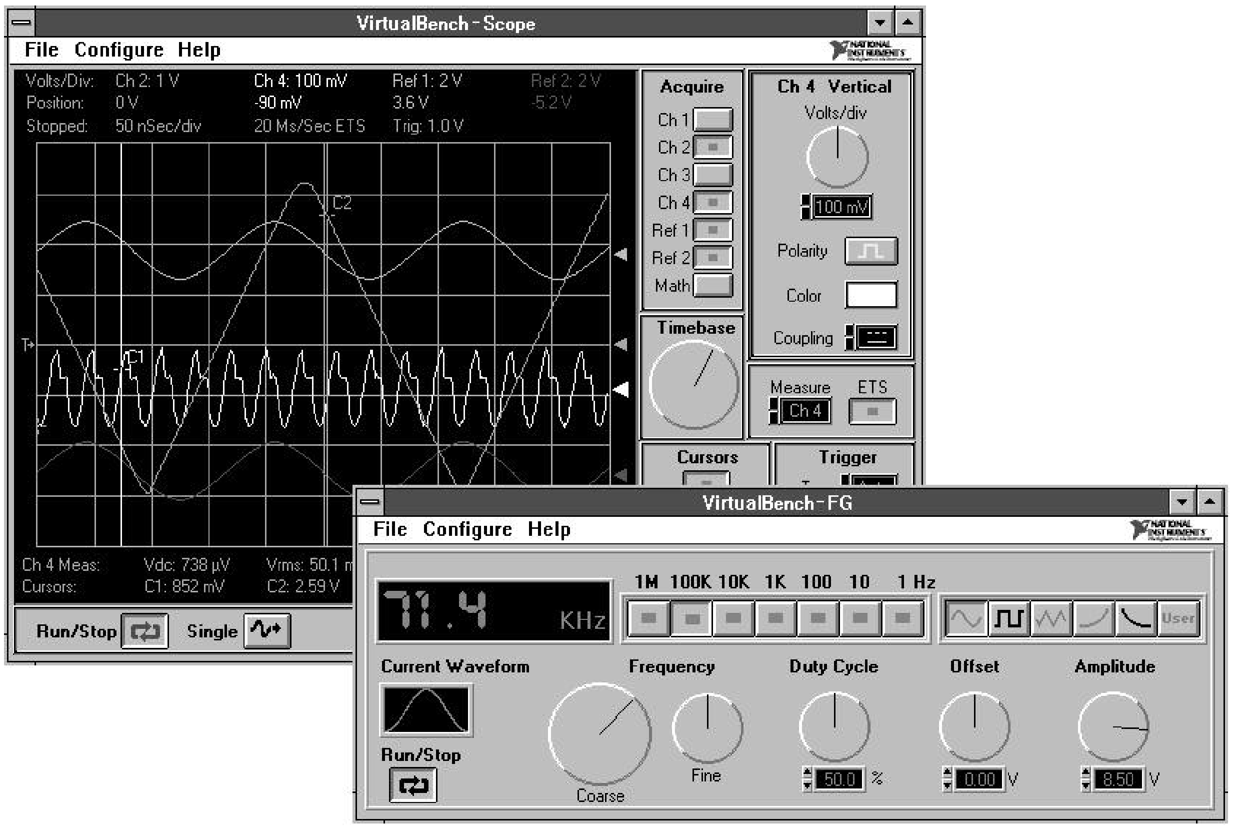
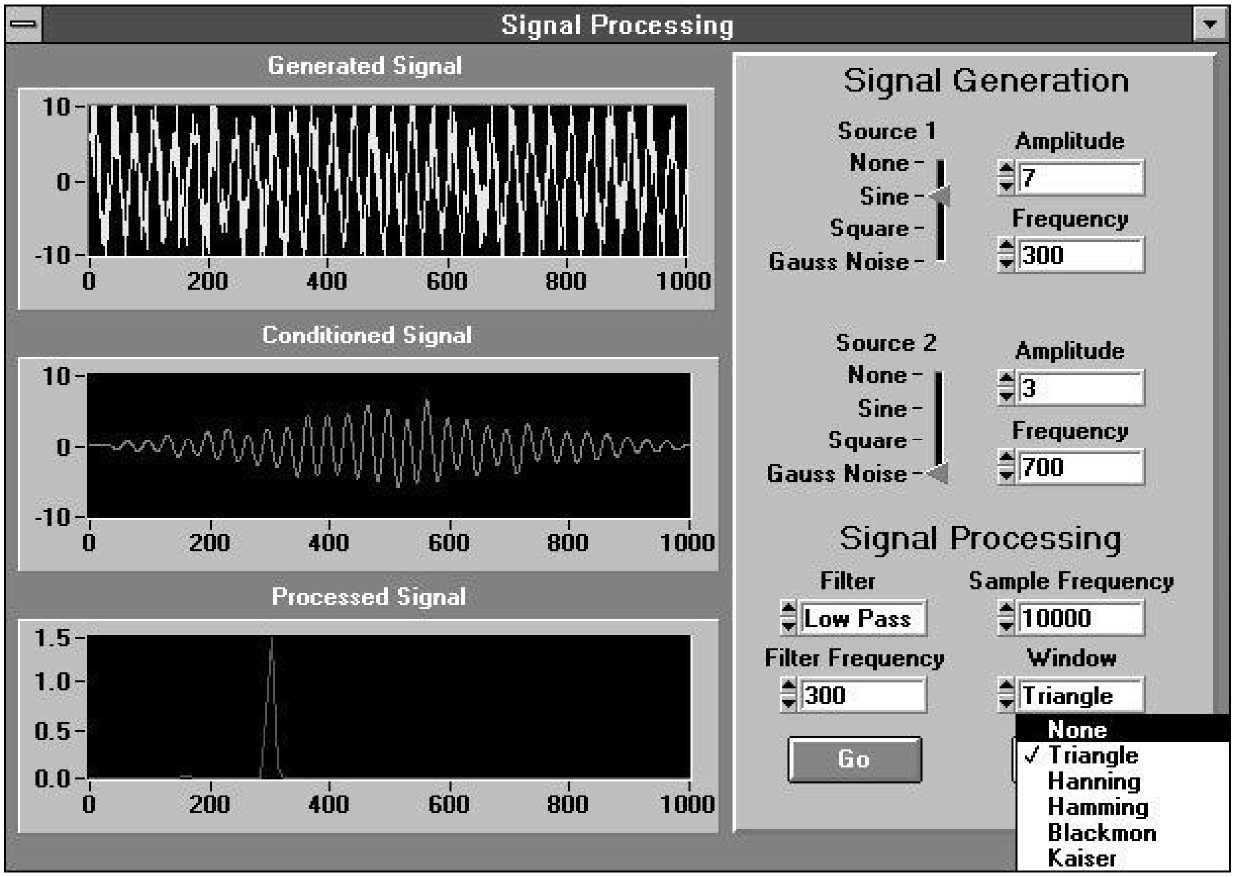


Figure 8. VirtualBench Stand-Alone Virtual Instruments – Oscilloscope and Function Generator

Developing Your System

To develop a high quality DAQ system for measurement and control or test and measurement, you must understand each of the components involved. Of all the DAQ system components, the element that should be examined most closely is the software. Because plug-in DAQ boards do not have displays, the software is the only interface you have to the system. The software is the component that relays all the information about the system, and it is the element that controls the system. The software integrates the transducers, signal conditioning, DAQ hardware, and analysis hardware into a complete, functional DAQ system.



Signal Generation

Source 1: Sine
Amplitude: 7
Frequency: 300

Source 2: Gauss Noise
Amplitude: 3
Frequency: 700

Signal Processing

Filter: Low Pass
Sample Frequency: 10000
Filter Frequency: 300
Window: Triangle

Figure 9. With the signal processing functions in the LabWindows/CVI Advanced Analysis Library, you can perform frequency analysis, filtering, and windowing operations on your data.

Therefore, when developing a DAQ system, be sure to completely evaluate the software. The hardware components can be selected by determining the requirements of your system and making sure that the hardware specifications are compatible with your system and your needs. Carefully selecting the proper software—whether it be driver level or application software—can save you a lot of development time and money.

340019C-01 Aug99