Click here to comment on this document via the
National Instruments website at
http://www.natinst.com/documentation/daq

Data Acquisition Fundamentals

R. House

Introduction

Scientists and engineers use personal computers (PCs) such as the IBM PC/XT/AT, PS/2, and Macintosh for laboratory research, industrial control, and test and measurement. Each element of a PC-based data acquisition system is essential for obtaining proper results. These elements are listed as follows and illustrated in Figure 1.

- The Personal Computer
- Transducers
- Signal Conditioning
- Data Acquisition Hardware
- Analysis 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 data acquisition system.

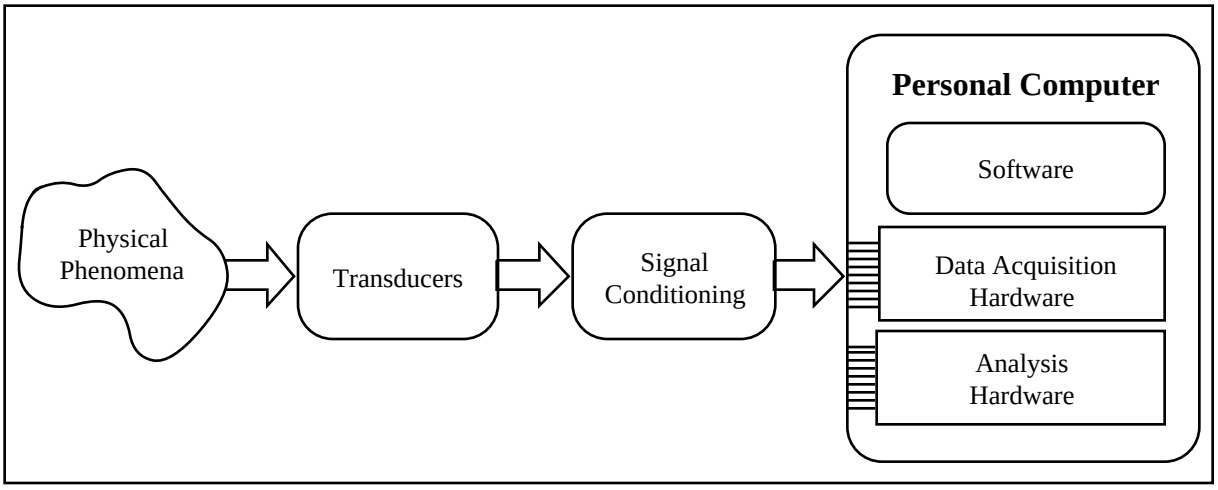


Figure 1. Elements of a PC-Based Data Acquisition System

Product names listed are trademarks of their respective manufacturers. Company names listed are trade names of their respective companies.

 © Copyright 1991 National Instruments Corporation. All rights reserved.

The Personal Computer

The computer used to acquire data can drastically affect the maximum speeds at which you are able to continuously acquire data. Newer bus architectures such as EISA, NuBus on the Macintosh II, and Micro Channel on the IBM PS/2 are able to deliver much higher throughput than the PC/XT/AT bus.

The types of data transfer available by the computer also significantly affect the performance of the data acquisition system. All PCs are capable of programmed I/O and interrupt transfers. DMA increases the system throughput by using dedicated hardware to transfer data directly into system memory. The processor is therefore free for other tasks. To use DMA or interrupt transfers, the data acquisition board must also be able to make these types of transfers.

The limiting factor for acquiring large amounts of data is often the hard disk. Therefore, for systems that need to acquire high-frequency signals, you should select a high-speed hard drive for your PC. Before performing an acquisition, make sure that there is enough contiguous (unfragmented) free disk space to hold the data. A fragmented hard drive reduces the maximum rate that acquired data can be streamed to disk.

Applications requiring real-time processing of high-frequency signals need a high-speed, 32-bit processor with its accompanying co-processor, 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 remember the end user of the system, because some systems are considered more user-friendly than others. Macintosh is known for its simple graphical user interface, although someone experienced with IBM systems is happier with a PC AT, EISA, or PS/2. Other factors that can influence your choice of PC are the other uses for the PC, cost constraints, and the availability of different computers.

Transducers

Transducers sense physical phenomena and produce electrical signals that the data acquisition hardware can accept. For example, thermocouples, resistive temperature detectors (RTDs), thermistors, and IC sensors convert temperature into an analog signal that an A/D converter (ADC) can measure. Other examples include strain gauges, flow meters, 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 converted into a form that the data acquisition board can accept. The most common task of signal conditioning is amplification. Low-level thermocouple signals, for example, should be amplified to increase the measurement resolution. For the highest possible resolution, the signal should be amplified so that the maximum voltage swing equals the maximum input range of the ADC. In addition, by placing the amplifier close to the transducer, interference from noise picked up on the lines between the transducer and the computer is minimized. This minimization occurs because the signal has already been amplified before it travels across the lines to the computer.

Another common signal conditioning task is linearization. Some signal conditioning products are designed to match the different types of thermocouples to both amplify and linearize the signal. Other signal conditioning products amplify the signal and use the software to linearize the signal.

With proper digital signal conditioning, you can use the low-current transistor-transistor logic (TTL) signals to/from the data acquisition 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 volts alternating current (VAC) at 2 A. Because the output of a digital I/O board is 0 to 5 volts direct current (VDC) at several milliamperes, a solid state relay–such as one of the National Instruments SSR Series modules–is needed to convert the signal to control the valve.

Another common application for signal conditioning is isolating the transducer signals from the computer for safety purposes. The system being monitored may contain high-voltage transients that can cause computer damage or harm the operator.

Signal conditioning is often used to filter unwanted signals. For example, the signal conditioning can contain lowpass filters that eliminate high frequencies that can produce erroneous data.

Signal conditioning products also produce excitation for some transducers. Strain gauges and RTDs, for example, require external signal sources. Signal conditioning modules for these transducers usually produce the stimulus source.

Data Acquisition Hardware

Analog Inputs

Common data acquisition board specifications are the number of channels, sampling rate, resolution, range, accuracy, noise, and nonlinearity, all of which affect the quality of the digitized signal. The number of analog channel inputs is 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 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, differential inputs should be used. 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 cancelled out.

Sampling rate determines how often conversions can take place. A faster sampling rate acquires more points in a given time and therefore can often form a better representation of the original signal. For example, audio signals converted to electrical signals by a microphone often have frequency components up to 20 kHz. To properly digitize this signal, 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. Therefore, a board with a sampling rate greater than 40 kHz (such as the National Instruments NB-A2100 high-resolution audio frequency analog I/O board) is needed to properly acquire this signal.

The signals produced by temperature transducers usually do not require a high sampling rate because temperature does not change quickly in most applications. Therefore, a lower-speed acquisition board can usually acquire temperature signals properly.

Multiplexing is a common technique for increasing the number of channels by routing multiple channels to a single ADC. 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, an AT-MIO-16H-9 sampling at 100 ksamples/sec on 10 channels effectively samples each individual channel at 10 ksamples/sec per channel. This rate is found by using the following formula:

$$\frac{\text{100 ksamples/sec}}{\text{10 channels}} = \text{10 ksamples/sec per channel}$$

If you need to measure more than 16 analog signals with an MIO-16 board, you can use the AMUX-64T analog multiplexer to measure up to 256 signals with a single board (see Figure 2). The sampling rate is reduced proportionately by this additional external multiplexing.

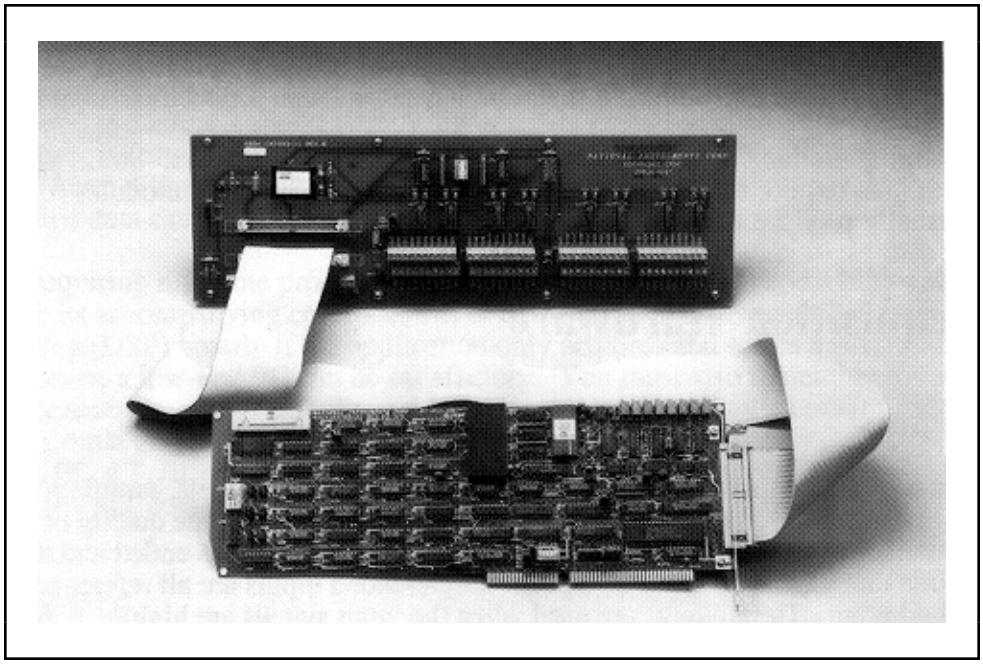


Figure 2. AMUX-64T Analog Multiplexer Connected to an AT-MIO-16

Resolution is the number of bits that the ADC uses to represent the analog signal. The higher the resolution, the higher the number of input voltage 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 a 3-bit ADC. A 3-bit converter (which is actually seldom used) divides the analog range into 2^3 or 8 divisions. Each division is represented by a binary code between 000 and 111. 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 ADC's number of codes increases from 8 to 65,536. Therefore, it can obtain an extremely accurate digital representation of the analog signal.

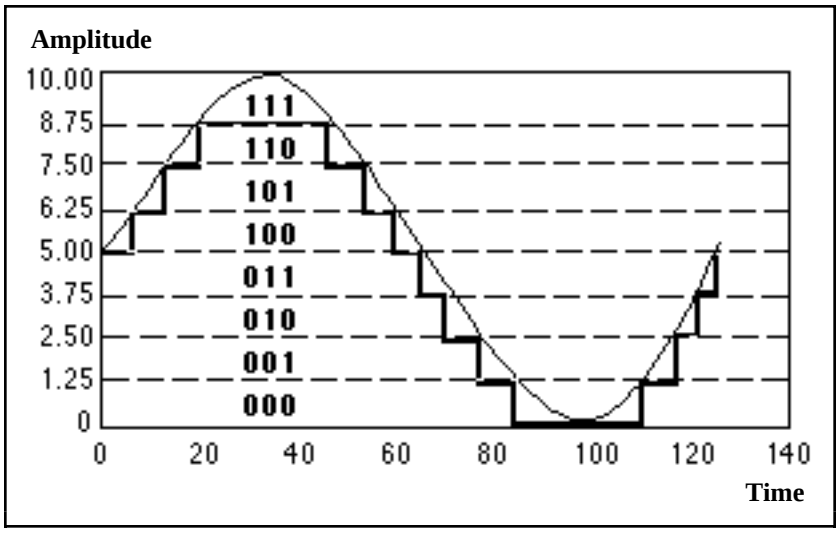


Figure 3. Digitized Sine Wave with a Resolution of Three Bits

Range refers to the minimum and maximum voltage levels that the ADC can quantize. The multifunction data acquisition boards have selectable ranges that make the board configurable to handle a variety of different voltage levels. With this flexibility, you can match the input signal's range to the board's signal range to take best advantage of the resolution available to accurately measure the signal.

The range, resolution, and gain available on a data acquisition board determine the smallest detectable change in voltage. This change in voltage represents 1 least significant bit (LSB) of the digital value, and is often called the *code width*. The ideal code width is found by using the following formula:

$$\text{ideal code width} = \frac{\text{voltage range}}{\text{gain } (2^{\text{resolution}})}$$

For example, the NB-MIO-16X (a National Instruments 16-bit multifunction data acquisition board for the Macintosh II) has voltage range selectable at 0 to 10 V, 0 to 5 V, -10 to 10 V, or -5 to 5 V and selectable gain of 1, 10, 100, 500 or 1, 2, 4, 8. With a voltage range of 0 to 5 V and a gain of 500, the ideal code width is

$$\frac{5}{500\ (2^{16})}$$

Therefore, the theoretical resolution of one bit in the digitized value is 153 nV.

This code width, along with the range, is used by the computer to scale the binary code from the ADC into an actual voltage value. Several potential errors can affect the accuracy of this translation, including gain and offset errors. Such errors affect the amplitude of the signal before it reaches the ADC, and cause the PC to use the wrong values for code width and range when scaling the binary code into a voltage value.

Gain and offset error, however, can be calibrated out using either hardware or software. Hardware calibration is accomplished by applying a known voltage to the input while adjusting potentiometers on the board until the scaled value is correct. Alternatively, software can compensate for voltage offset errors. You must determine how far the translated value is from the

actual value and add scaling routines in software to compensate for this difference. Boards with self-calibrating circuitry–such as the AT-MIO-15F-5, NB-A2000, and EISA-A2000–do not require either of these steps to calibrate out these errors.

Another error that can affect the analog input is noise. Noise reduces the resolution of the ADC because as the noise level approaches 1 LSB, the ADC can no longer distinguish between a signal increase of one code width or equivalent level noise. The errors due to noise can be reduced by sampling at a higher rate and averaging the data acquired.

Ideally, the code widths for each of the voltage divisions are the same, and a plot of voltage versus binary code would be a straight line as shown in Figure 4.

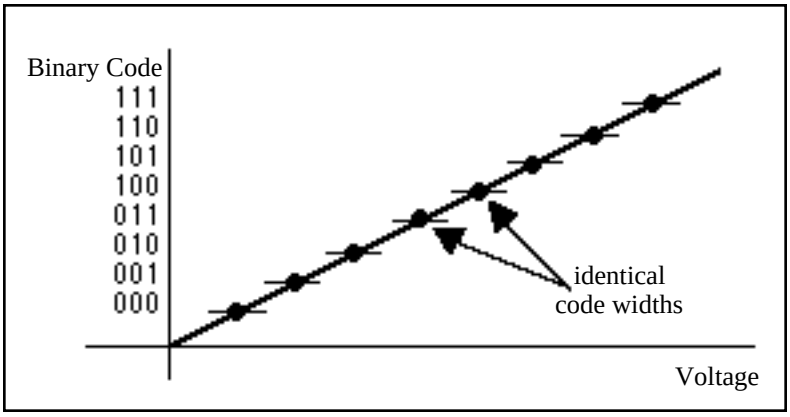


Figure 4. Ideal Binary Code Versus Voltage Plot

The integral nonlinearity specified for an ADC tells how far this plot deviates from the straight line, while the differential nonlinearity tells how equal the code widths are as Figure 5 illustrates.

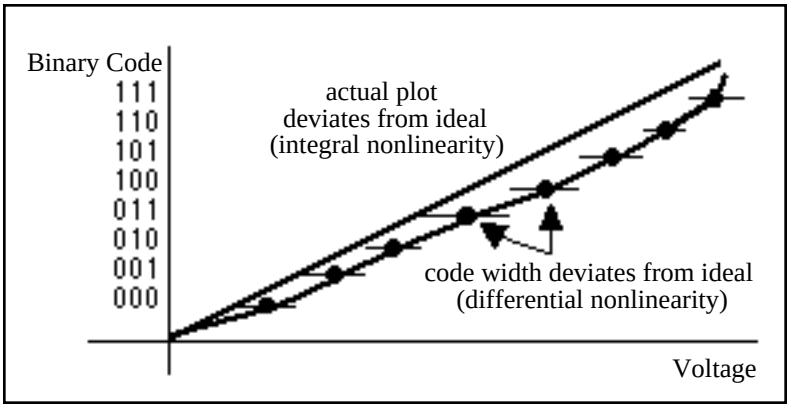


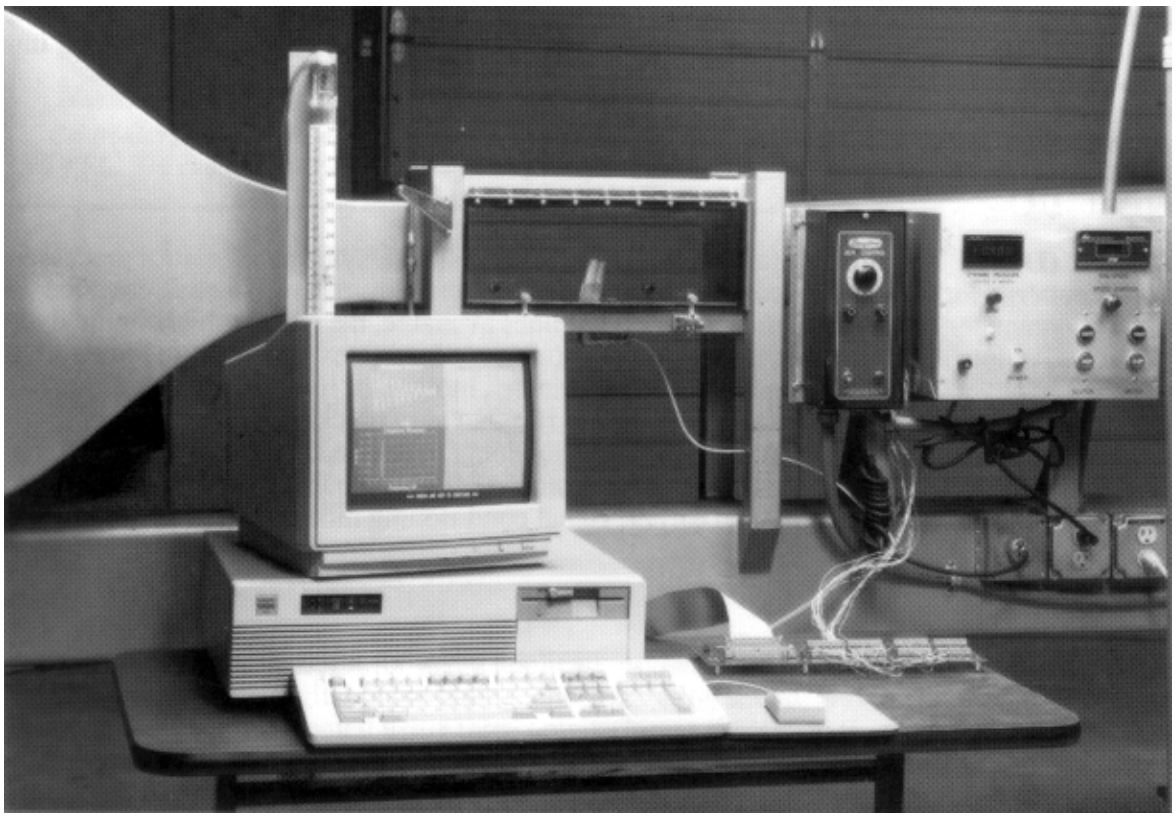
Figure 5. Binary Code Voltage Plot Showing Nonlinearity

Good integral linearity is important because accurate translation from the binary code to the voltage is a matter of simple scaling. Ideal differential nonlinearity assures that the translated voltage reading is within ±0.5 LSB of the actual voltage of the input signal.

Analog Outputs

Analog output circuitry is needed to stimulate a process or unit under test in a data acquisition system. Several specifications for digital-to-analog converters (DACs) 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 slew rate is the maximum rate of change that the DAC can produce on the output signal. Therefore, 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.

One example application area that requires high performance in these parameters is the generation of audio signals. The DAC requires 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 determines the DAC specifications.



An AT-MIO-16 controls a wind tunnel flow rate and senses resulting pressure patterns using the AMUX-64T.

Figure 6. Controlling a Wind Tunnel

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 data acquisition applications need to start or stop a data acquisition 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 data acquisition operation when an input signal reaches an analog voltage level and slope.

Real-Time System Integration (RTSI®)

The National Instruments expertise in instrumentation led to the development of the RTSI bus for our data acquisition products. The RTSI bus uses a custom gate array and a ribbon cable to route timing and trigger signals between multiple functions on one data acquisition board, or between multiple 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

Digital I/O (DIO) interfaces are often used on PC data acquisition 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 digital data can be input and output 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 needs to match the number of processes that are being 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. As mentioned earlier, signal conditioning modules such as the SSR Series can correct this problem.



Centronics printers have a parallel interface that is interfaced to the Macintosh computer through the NB-PRL. The cable connects the proper pins of the NB-PRL board to the standard Centronics connector.

Figure 7. Interfacing a Macintosh Computer to Centronics Printers

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 have handshaking circuitry for communication synchronization purposes. The number of channels, data rate, and handshaking requirements should be matched to the application.

Timing I/O

Counter/timer circuitry is useful for many applications, including counting the occurrences of a digital event, measuring digital pulse timing, and generating square waves and pulses. All of these applications can be implemented 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 provides the timebase for the operation of the counter by causing the counter to increment each time it toggles. Finally, a counter can generate digital square waves and pulses at the output line.

The most significant specifications for the operation of a counter/timer are the resolution and clock frequency. The resolution is the number of bits the counter uses to count with. A higher resolution simply means that the counter can count higher. Each counter/timer on the MIO-16 boards and the TIO-10 boards has 16-bit resolution, which corresponds to a count up to 65,536. In addition, two of these counters can be cascaded to make a 32-bit counter that can count to over

four billion. The clock frequency determines how fast the digital source can be toggled. A higher frequency lets the counter increment faster and therefore detect higher frequency signals on the input and generate higher frequency pulses and square waves on the output.

Analysis Hardware

The processing capabilities of PCs have increased to a level at which they have plenty of computational power for most data acquisition and analysis applications. In some high-performance applications, however, the PC microprocessor cannot process data fast enough to respond to a real-world signal. Other applications have numerically intensive calculations that need to be performed on large sets of data, requiring you to wait a long time before obtaining any results. Analysis hardware such as the NB-DSP2300 board, based on a high-speed digital signal processor, can solve these problems by making computations at even higher rates than the PC microprocessor. In addition, these complex computations can be downloaded to the analysis hardware so that computations are made simultaneously as the PC microprocessor executes the application program.

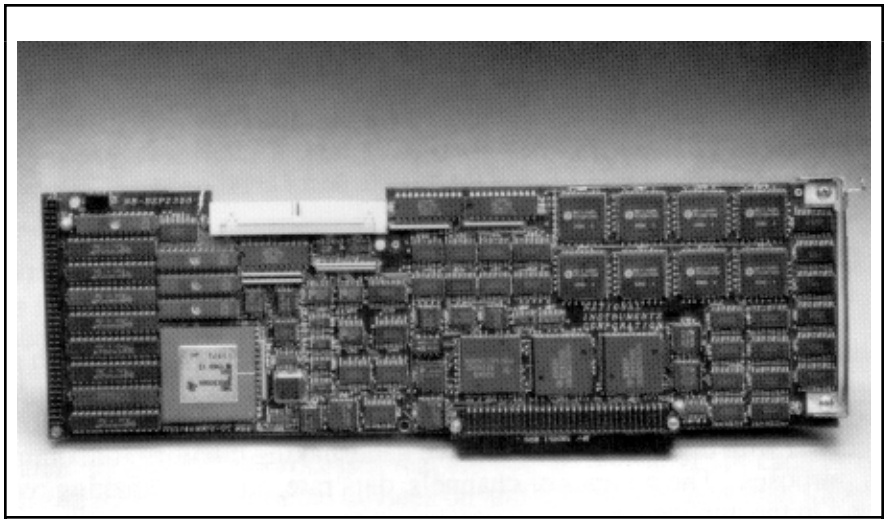


Figure 8. NB-DSP2300 Digital Signal Processing and Accelerator Board

The digital signal processor can compute much faster than a general-purpose microprocessor by multiplying and accumulating two numbers in one clock cycle. The general-purpose microprocessor in the PC requires many cycles to perform the same operation.

Digital signal processors are available in a wide range of accuracies and formats. Today's state-of-the-art digital signal processors are 32-bit accurate with floating-point format. These digital signal processors have a much higher dynamic range than fixed-point (integer) format digital signal processors. In addition, the applications developed with floating-point digital signal processors do not require the software overhead associated with fixed-point to maintain floating-point accuracy. Therefore, floating-point applications are simpler to develop and execute faster with floating-point digital signal processors.

One indication of the computational power of a floating-point digital signal processor is the number of floating-point calculations it can perform in a second. The NB-DSP2300 board equipped with the Texas Instruments TMS320C30 chip is capable of 33 million floating-point operations per second (MFLOPS). The general-purpose processors and coprocessors in the Macintosh II family of computers, the Motorola 680x0 and 6888x, are capable of only a few MFLOPS under optimal conditions. This difference in rate of computing floating-point calculations translates to as much as 100 : 1 time-improvement for computing a 1,024 word real Fast Fourier Transform (FFT) on the NB-DSP2300 instead of on the Macintosh processor.

The computational power of analysis hardware is being used in many application areas. Most applications that need to run faster can benefit from analysis hardware. Applications that have high-frequency signals and need real-time response are now possible on the PC because of digital signal processing hardware. Some of the most common application areas for digital signal processing hardware are in the fields of acoustics (sonar, speech processing, hearing aid research, and so on) as well as communications, instrumentation, and image processing.

Software

Software transforms the PC and data acquisition hardware into a complete data acquisition, analysis, and display system.

Programming the registers on the board directly is the most difficult level of data acquisition software development. You must determine the proper binary values to write to these registers on the board. The programming language that you use must be able to write to and read data from the data acquisition boards plugged into the computer. All National Instruments data acquisition boards are shipped with documentation that describes how to program the boards at this level. However, often you do not have the desire, skill, or time to program the boards at this level.

Data acquisition software, which removes the complexities associated with register-level programming, is broken into two areas: driver-level software and application-level software.

Driver software simplifies data acquisition programming by taking care of the low-level hardware programming details and giving you high-level function calls that can be used with conventional programming languages.

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

```
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);
}
```

Application software packages such as LabVIEW®, LabWindows®, and Measure add analysis and presentation capabilities to the driver software. The application software packages also integrate instrument control (GPIB, RS-232, and VXI) into data acquisition.

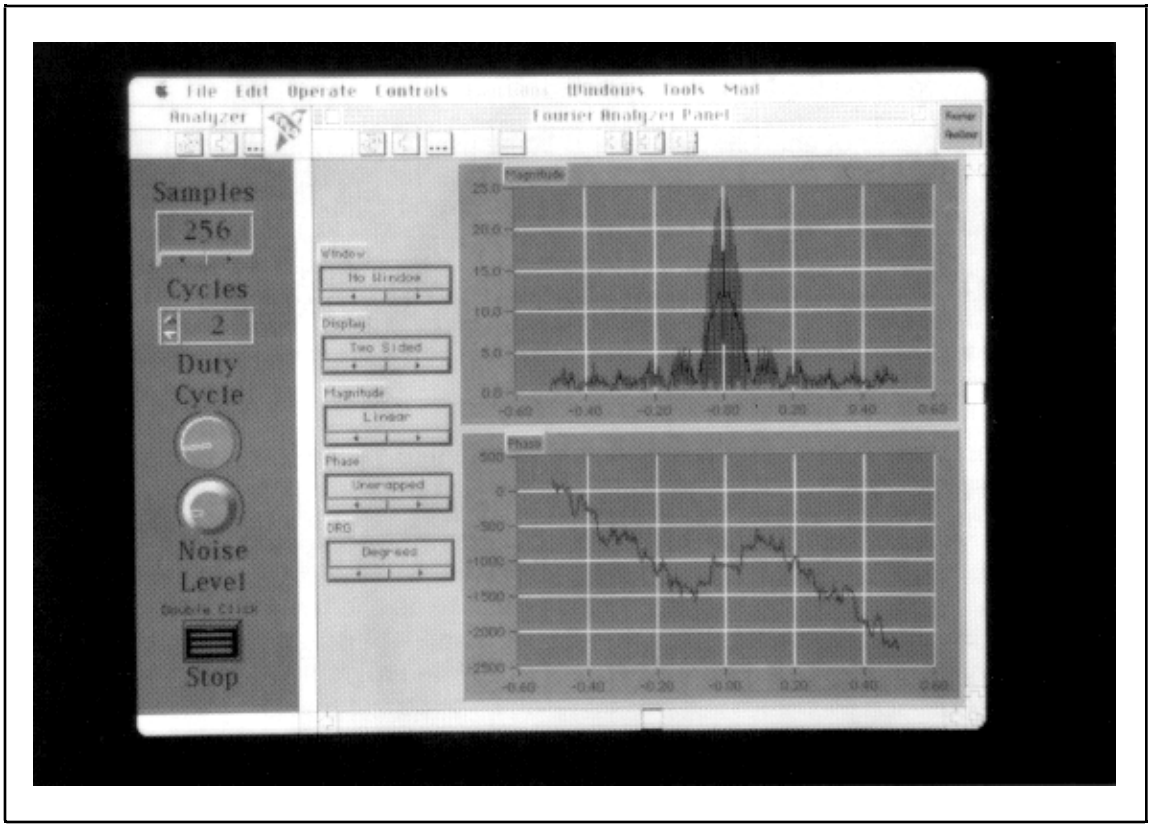


Figure 9. Fourier Analysis with LabVIEW

Developing Your System

To develop a high quality data acquisition system for measurement and control or test and measurement, you must understand each of the components involved. Of all the data acquisition system components, the element that should be examined most closely is the software. Because plug-in data acquisition 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, data acquisition hardware, and analysis hardware into a complete, functional data acquisition system.

Therefore, when developing a data acquisition 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.