Virtual Instrumentation
What Is Virtual Instrumentation?

The rapid adoption of the PC in the last 20 years catalyzed a revolution in instrumentation for test, measurement, and automation. One major development resulting from the ubiquity of the PC is the concept of virtual instrumentation, which offers several benefits to engineers and scientists who require increased productivity, accuracy, and performance.

A virtual instrument consists of an industry-standard computer or workstation equipped with powerful application software, cost-effective hardware such as plug-in boards, and driver software, which together perform the functions of traditional instruments. Virtual instruments represent a fundamental shift from traditional hardware-centered instrumentation systems to software-centered systems that exploit the computing power, productivity, display, and connectivity capabilities of popular desktop computers and workstations. Although the PC and integrated circuit technology have experienced significant advances in the last two decades, it is software that truly provides the leverage to build on this powerful hardware foundation to create virtual instruments, providing better ways to innovate and significantly reduce cost. With virtual instruments, engineers and scientists build measurement and automation systems that suit their needs exactly (user-defined) instead of being limited by traditional fixed-function instruments (vendor-defined).

This document describes powerful programming tools, flexible acquisition hardware, and the personal computer, which are the essential components for virtual instrumentation. The synergy between them offers advantages that cannot be matched by traditional instrumentation.

Virtual Instruments versus Traditional Instruments

Stand-alone traditional instruments such as oscilloscopes and waveform generators are very powerful, expensive, and designed to perform one or more specific tasks defined by the vendor. However, the user generally cannot extend or customize them. The knobs and buttons on the instrument, the built-in circuitry, and the functions available to the user, are all specific to the nature of the instrument. In addition, special technology and costly components must be developed to build these instruments, making them very expensive and slow to adapt.

Virtual instruments, by virtue of being PC-based, inherently take advantage of the benefits from the latest technology incorporated into off-the-shelf PCs. These advances in technology and performance, which are quickly closing the gap between stand-alone instruments and PCs, include powerful processors such as the Pentium 4 and operating systems and technologies such as Microsoft Windows XP, .NET, and Apple Mac OS X. In addition to incorporating powerful features, these platforms also offer easy access to powerful tools such as the Internet. Traditional instruments also frequently lack portability, whereas virtual instruments running on notebooks automatically incorporate their portable nature.

Engineers and scientists whose needs, applications, and requirements change very quickly, need flexibility to create their own solutions. You can adapt a virtual instrument to your particular needs without having to replace the entire device because of the application software installed on the PC and the wide range of available plug-in hardware.
**Flexibility**

Except for the specialized components and circuitry found in traditional instruments, the general architecture of stand-alone instruments is very similar to that of a PC-based virtual instrument. Both require one or more microprocessors, communication ports (for example, serial and GPIB), and display capabilities, as well as data acquisition modules. What makes one different from the other is their flexibility and the fact that you can modify and adapt the instrument to your particular needs. A traditional instrument might contain an integrated circuit to perform a particular set of data processing functions; in a virtual instrument, these functions would be performed by software running on the PC processor. You can extend the set of functions easily, limited only by the power of the software used.

**Lower Cost**

By employing virtual instrumentation solutions, you can lower capital costs, system development costs, and system maintenance costs, while improving time to market and the quality of your own products.

**Plug-In and Networked Hardware**

There is a wide variety of available hardware that you can either plug into the computer or access through a network. These devices offer a wide range of data acquisition capabilities at a significantly lower cost than that of dedicated devices. As integrated circuit technology advances, and off-the-shelf components become cheaper and more powerful, so do the boards that use them. With these advances in technology come an increase in data acquisition rates, measurement accuracy, precision, and better signal isolation.

Depending on the particular application, the hardware you choose might include analog input or output, digital input or output, counters, timers, filters, simultaneous sampling, and waveform generation capabilities. The wide gamut of boards and hardware could include any one of these features or a combination of them.

**Software in Virtual Instrumentation**

Software is the most important component of a virtual instrument. With the right software tool, engineers and scientists can efficiently create their own applications, by designing and integrating the routines that a particular process requires. They can also create an appropriate user interface that best suits the purpose of the application and those who will interact with it. They can define how and when the application acquires data from the device, how it processes, manipulates and stores the data, and how the results are presented to the user.

With powerful software, you can build intelligence and decision-making capabilities into the instrument so that it adapts when measured signals change inadvertently or when more or less processing power is required.

An important advantage that software provides is modularity. When dealing with a large project, engineers and scientists generally approach the task by breaking it down into functional solvable units. These subtasks are more manageable and easier to test, given the reduced dependencies that might cause unexpected behavior. You can design a virtual
instrument to solve each of these subtasks, and then join them into a complete system to solve the larger task. The ease with which you can accomplish this division of tasks depends greatly on the underlying architecture of the software.

Distributed Applications

A virtual instrument is not limited or confined to a stand-alone PC. In fact, with recent developments in networking technologies and the Internet, it is more common for instruments to use the power of connectivity for the purpose of task sharing. Typical examples include supercomputers, distributed monitoring and control devices, as well as data or result visualization from multiple locations.

Why Is LabVIEW Ideal for Creating Virtual Instruments?

LabVIEW is an integral part of virtual instrumentation because it provides an easy-to-use application development environment designed specifically with the needs of engineers and scientists in mind. LabVIEW offers powerful features that make it easy to connect to a wide variety of hardware and other software.

Graphical Programming

One of the most powerful features that LabVIEW offers engineers and scientists is its graphical programming environment. With LabVIEW, you can design custom virtual instruments by creating a graphical user interface on the computer screen through which you:

• Operate the instrumentation program
• Control selected hardware
• Analyze acquired data
• Display results

You can customize front panels with knobs, buttons, dials, and graphs to emulate control panels of traditional instruments, create custom test panels, or visually represent the control and operation of processes. The similarity between standard flow charts and graphical programs shortens the learning curve associated with traditional, text-based languages.
Figure 1. LabVIEW Virtual Instrument Front Panel

Figure 2. LabVIEW Virtual Instrument Block Diagram
You determine the behavior of the virtual instruments by connecting icons together to create block diagrams, which are natural design notations for scientists and engineers. With graphical programming, you can develop systems more rapidly than with conventional programming languages, while retaining the power and flexibility needed to create a variety of applications.

**Connectivity and Instrument Control**

Virtual instrumentation software productivity comes about because the software includes built-in knowledge of hardware integration. Designed to create test, measurement, and control systems, virtual instrumentation software includes extensive functionality for I/O of almost any kind.

LabVIEW has ready-to-use libraries for integrating stand-alone instruments, data acquisition devices, motion control and vision products, GPIB/IEEE 488 and serial/RS-232 devices, and PLCs, among others, to build a complete measurement and automation solution. LabVIEW also incorporates major instrumentation standards such as VISA, an interoperable standard for GPIB, serial, and VXI instrumentation; PXI and software and hardware based on the PXI Systems Alliance CompactPCI standard; IVI interchangeable virtual instrument drivers; and VXIplug&play, a driver standard for VXI instruments.

**Open Environment**

Although LabVIEW provides the tools required for most applications, LabVIEW also is an open development environment. Standardization of software relies greatly on the ability of the package you select to work well with other software, measurement and control hardware, and open standards, which define interoperability between multiple vendors. By selecting software that meets these criteria, you ensure that the company and applications take advantage of the products offered by several suppliers. In addition, conforming to open commercial standards reduces overall system cost.

A large number of third-party hardware and software vendors develop and maintain hundreds of LabVIEW libraries and instrument drivers to help you easily use their products with LabVIEW. However, this is not the only way to provide connectivity to LabVIEW-based applications. LabVIEW offers simple ways to incorporate ActiveX software, dynamic link libraries (DLLs), and shared libraries from other tools. In addition, you can share LabVIEW code as a DLL, built executable, or using ActiveX.

LabVIEW also offers a full range of options for communications and data standards, such as TCP/IP, OPC, SQL database connectivity, and XML data formats.

**Reduces Cost and Preserves Investment**

Because you can use a single computer equipped with LabVIEW for countless applications and purposes, it is a versatile product. It is not only versatile but also extremely cost-effective. Virtual instrumentation with LabVIEW proves to be economical, not only in the reduced development costs but also in its preservation of capital investment over a long period of time. As your needs change, you can modify systems easily without the need to buy new equipment. You can create complete instrumentation libraries for less than the cost of a single traditional, commercial instrument.
Multiple Platforms

The majority of computer systems use some variation of the Microsoft Windows operating system. Nevertheless, other options offer clear advantages for certain types of applications. Real-time and embedded development continues to grow rapidly in most industries, as computing power is packaged into smaller and more specialized packages. Minimizing losses resulting from changing to new platforms is important and choosing the right software for this purpose is a key factor.

LabVIEW minimizes this concern, because it runs on Windows 2000, NT, XP, Me, 98, 95, and NT embedded, as well as Mac OS, Sun Solaris, and Linux. LabVIEW also compiles code to run on the VenturCom ETS real-time operating system through the LabVIEW Real-Time Module. Given the importance of legacy systems, National Instruments continues to make available older versions of LabVIEW for Windows, Mac OS, and Sun operating systems. LabVIEW is platform independent; virtual instruments that you write in one platform can transparently be ported to any other LabVIEW platform by simply opening the virtual instrument.

Because LabVIEW applications are portable across platforms, you can be assured that your work today will be applicable in the future. As new computer technologies emerge, you can easily migrate your applications to new platforms and operating systems. In addition, because you can create platform-independent virtual instruments by porting applications between platforms, you save development time and other inconveniences related to platform portability.

Distributed Development

You can easily develop distributed applications with LabVIEW, even across different platforms. With easy-to-use server tools, you can offload processor-intensive routines to other machines for faster execution, or create remote monitoring and control applications. Powerful server technology can simplify the task of developing large, multicomputer applications. In addition, LabVIEW includes standard networking technologies such as TCP/IP, and incorporates robust publish and subscribe protocols.

Analysis Capabilities

Virtual instrumentation software requires comprehensive analysis and signal processing tools, because the application does not just stop when the data is collected. High-speed measurement applications in machine monitoring and control systems usually require order analysis for accurate vibration data. Closed-loop, embedded control systems might need point-by-point averaging for control algorithms to maintain stability. In addition to the advanced analysis libraries already included in LabVIEW, National Instruments provides add-on software such as the LabVIEW Signal Processing Toolset, the LabVIEW Sound and Vibration Toolkit, and the LabVIEW Order Analysis Toolkit to complement analysis offerings.
Visualization Capabilities

LabVIEW includes a wide array of built-in visualization tools to present data on the user interface of the virtual instrument – for charting and graphing as well as 2D and 3D visualization. You can instantly reconfigure attributes of the data presentation, such as colors, font size, graph types, and more, as well as dynamically rotate, zoom, and pan these graphs with the mouse. Rather than programming graphics and all custom attributes from scratch, you can simply drag-and-drop these objects onto the instrument front panels.

Flexibility and Scalability – Key Advantages

Engineers and scientists have needs and requirements that can change rapidly. They also need to have maintainable, extensible solutions that can used for a long time. By creating virtual instruments based on powerful development software such as LabVIEW, you inherently design an open framework that seamlessly integrates software and hardware. This ensures that your applications not only work well today but that you can easily integrate new technologies in the future as they become available, or extend your solutions beyond the original scope, as new requirements are identified. Moreover, every application has its own unique requirements that require a broad range of solutions.

Virtual Instrumentation in the Engineering Process

Virtual instruments provide significant advantages in every stage of the engineering process, from research and design to manufacturing test.

Research and Design

In research and design, engineers and scientists demand rapid development and prototyping capabilities. With virtual instruments, you can quickly develop a program, take measurements from an instrument to test a prototype, and analyze results, all in a fraction of the time required to build tests with traditional instruments. When you need flexibility, an scalable open platform is essential, from the desktop, to embedded systems, to distributed networks.

The demanding requirements of research and development (R&D) applications require seamless software and hardware integration. Whether you need to interface stand-alone instruments using GPIB or directly acquire signals into the computer with a data acquisition board and signal conditioning hardware, LabVIEW makes integration simple. With virtual instruments, you also can automate a testing procedure, eliminating the possibility of human error and ensuring the consistency of the results by not introducing unknown or unexpected variables.

Development Test and Validation

With the flexibility and power of virtual instruments, you can easily build complex test procedures. For automated design verification testing, you can create test routines in LabVIEW and integrate software such as National Instruments TestStand, which offers powerful test management capabilities. One of the many advantages these tools offer across the organization is code reuse. You develop code in the design process, and then plug these same programs into functional tools for validation, test, or manufacturing.
**Manufacturing Test**

Decreasing test time and simplifying development of test procedures are primary goals in manufacturing test. Virtual instruments based on LabVIEW combined with powerful test management software such as TestStand deliver high performance to meet those needs. These tools meet rigorous throughput requirements with a high-speed, multithreaded engine for running multiple test sequences in parallel. TestStand easily manages test sequencing, execution, and reporting based on routines written in LabVIEW.

TestStand integrates the creation of test code in LabVIEW. TestStand also can reuse code created in R&D or design and validation. If you have manufacturing test applications, you can take full advantage of the work already done in the product life cycle.

**Manufacturing**

Manufacturing applications require software to be reliable, high in performance, and interoperable. Virtual instruments based on LabVIEW offer all these advantages, by integrating features such as alarm management, historical data trending, security, networking, industrial I/O, and enterprise connectivity. With this functionality, you can easily connect to many types of industrial devices such as PLCs, industrial networks, distributed I/O, and plug-in data acquisition boards. By sharing code across the enterprise, manufacturing can use the same LabVIEW applications developed in R&D or validation, and integrate seamlessly with manufacturing test processes.

**Virtual Instruments beyond the Personal Computer**

Recently, commercial PC technologies have begun migrating into embedded systems. Examples include Windows CE, Intel x86-based processors, PCI and CompactPCI buses, and Ethernet for embedded development. Because virtual instrumentation relies so heavily on commercial technologies for cost and performance advantages, it also has expanded to encompass more embedded and real-time capabilities. For example, LabVIEW runs on Linux as well as the embedded ETS real-time operating system from VenturCom on specific embedded targets. The option of using virtual instrumentation as a scalable framework that extends from the desktop to embedded devices should be considered a tool in the complete toolbox of an embedded systems developer.

A dramatic technology change example that affects embedded systems development is networking and the Web. With the ubiquity of PCs, Ethernet now dominates as the standard network infrastructure for companies worldwide. In addition, the popularity of the Web interface in the PC world has overflowed into the development of cell phones, PDAs, and now industrial data acquisition and control systems.

Embedded systems at one time meant stand-alone operation, or at most interfacing at a low level with a real-time bus to peripheral components. Now, the increased demand for information at all levels of the enterprise (and in consumer products) requires you to network embedded systems while continuing to guarantee reliable and often real-time operation.
Because virtual instrumentation software can combine one development environment for both desktop and real-time systems using cross-platform compiled technology, you can capitalize on the built-in Web servers and easy-to-use networking functionality of desktop software and target it to real-time and embedded systems. For example, you could use LabVIEW to simply configure a built-in Web server to export an application interface to defined secure machines on the network on Windows, and then download that application to run on a headless embedded system that can fit in the user's hand. This procedure happens with no additional programming required on the embedded system. You then can deploy that embedded system, power it, connect to the application from a remote secure machine via Ethernet, and interface to it using a standard Web browser. For more sophisticated networking applications, you can graphically program TCP/IP or other methods with which you are already familiar in LabVIEW and then run them in the embedded system.

Embedded systems development is one of the fastest growing segments of engineering, and will continue to be for the foreseeable future as consumers demand smarter cars, appliances, homes, and so on. The evolution of these commercial technologies will propel virtual instrumentation into being more applicable to a growing number of applications. Leading companies that provide virtual instrumentation software and hardware tools need to invest in expertise and product development to serve this growing set of applications. For example, for its flagship virtual instrumentation software platform, LabVIEW, National Instruments has described a vision that includes the ability to scale from development for desktop operating systems, to embedded real-time systems, to handheld personal digital assistant targets, to FPGA-based hardware, and even to enabling smart sensors.

Next-generation virtual instrumentation tools need to include networking technology for quick and easy integration of Bluetooth, wireless Ethernet, and other standards. In addition to using these technologies, virtual instrumentation software needs a better way to describe and design timing and synchronization relationships between distributed systems in an intuitive way to help faster development and control of these often embedded systems.

The virtual instrumentation concepts of integrated software and hardware, flexible modular tools, and the use of commercial technologies combine to create a framework upon which you can rapidly complete your systems development and also maintain them for the long term. Because virtual instrumentation offers so many options and capabilities in embedded development, it makes sense for embedded developers to understand and review these tools.

**Conclusion**

Virtual instrumentation is fueled by ever-advancing computer technology and offers you the power to create and define your own system based on an open framework. This concept not only ensures that your work will be usable in the future but also provides the flexibility to adapt and extend as needs change. LabVIEW was designed with scientists and engineers in mind, providing powerful tools and a familiar development environment created specifically for the design of virtual instruments.