VIRTUAL INSTRUMENTATION

What is Virtual Instrumentation?

A virtual instrument consists of a computer 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.

Virtual Instruments versus Traditional Instruments

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 manufacturer. But, 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 and are not customizable.

In addition, special technology and costly components must be developed to build these instruments, making them very expensive.

Virtual Instruments:

Virtual instruments, by virtue of being PC-based, take advantage of the benefits from the latest technology incorporated into PCs like powerful processors and operating systems.
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.

**Advantages of VI**

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**

What makes VI different from the conventional instrument is 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 instruments.

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.