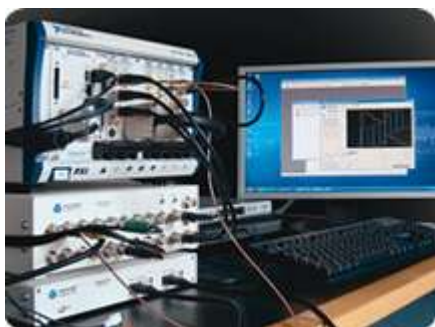


什么是模块化仪器？





设备的日趋复杂和技术的渐进融合迫使测试系统必须变得更加灵活。尽管测试系统必须适应日新月异的设备，但成本的压力要求系统具有更长的生命周期。

实现这些目标的唯一方式便是采用软件定义的模块化架构。通过共享的元器件、高速总线和用户定义的开放式软件，模块化仪器无疑是当今和未来满足自动化测试设备(ATE)各种需求的理想选择。

概览

模块化仪器：用户定义的灵活软件和可升级硬件组件

设备的日趋复杂和技术的渐进融合迫使测试系统必须变得更加灵活。尽管成本的压力要求系统具有更长的使用寿命，但测试系统也必须适应日新月异的设备。实现这些目标的唯一方式便是采用软件定义的模块化架构。本文将通过虚拟仪器系统引入软件定义的概念，为硬件平台和软件实现提供多种选择，并讨论模块化仪器系统是如何满足自动化测试设备（ATE）的要求。

基本上，目前有两种类型的仪器系统：虚拟仪器和传统仪器。图 1 描述了这些类型仪器的架构。

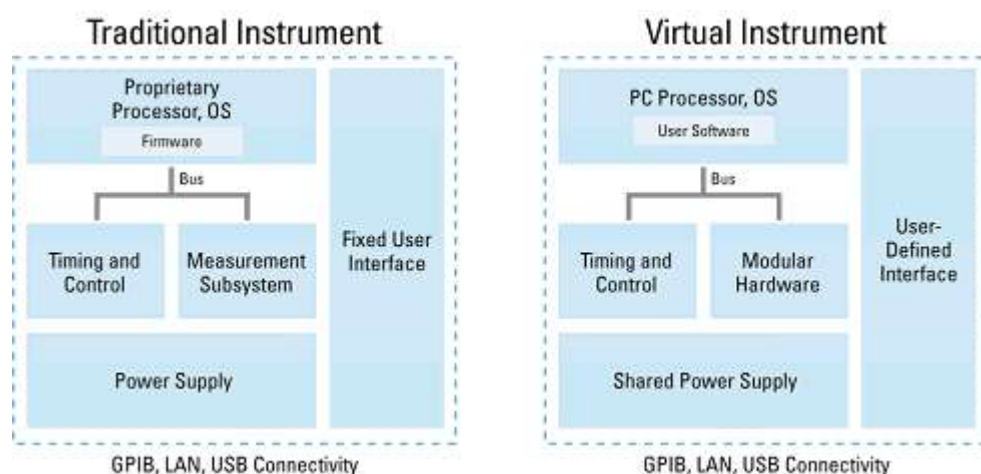


图 1. 将传统仪器和虚拟仪器架构相比较，两者的硬件组件基本相同；两种架构之间的主要区别在于软件的驻留位置和用户是否可访问。

图中显示了这两种类型仪器系统的相似之处。两者均具有测量硬件、1 个机箱、1 个电源、1 条总线、1 个处理器、1 个操作系统和 1 个用户界面。由于这两种方法使用相同的基本组件，如果单从硬

件的角度上看，两者最明显的区别在于组件的封装。传统或独立式仪器将所有组件都封装在一个适用于任何分立仪器的机箱内。独立式仪器的一个典型例子是通过 GPIB、USB 或 LAN/以太网控制的手动仪器。这些仪器是作为分立器件设计的，其主要设计目的并不是集成为系统使用。虽然传统仪器数量众多，但就仪器本身而言，其软件处理和用户界面都是固定的，仅当厂商选择更新时才可以被更新，而且如何更新也取决于厂商的选择(例如通过固件的更新)。因此，用户不可能通过传统仪器进行其功能列表未囊括的测量，而且这使得根据新的标准进行测量，或者根据需求的变化调整系统都极具挑战性。

相比之下，软件定义的虚拟仪器可帮助用户获得硬件中的原始数据，从而定义用户自己的测量和用户界面。通过这种软件定义的方法，用户可进行自定义测量，根据新标准进行测量或者根据需求变化调整系统（例如，增加仪器、通道或测量）。尽管用户定义的软件可应用于特定应用的独立式硬件，但其最理想的搭配还是通用的模块化硬件，通过这种结合，测量软件的灵活性和性能都可以得到充分的利用。模块化仪器的核心是用户定义的灵活软件与可升级硬件组件的完美结合。

硬件

支持系统可扩展性的模块化硬件

模块化仪器可采取多种形式。在设计完善的模块化仪器系统中，许多组件—如机箱和电源—都通过仪器模块共享，而不是在每个仪器功能上重复使用这些组件。这些仪器模块还包括不同类型的硬件，如示波器、函数发生器、数字化仪和 RF。在一些情况下，如图 1 所示，测量硬件仅仅是安装在其中一个主机外设端口或外设插槽的外围设备。这时，主机提供了用于运行测量软件的处理器以及电源和 I/O 的机箱。



图 1. 模块化仪器可选的测量硬件示例，如 USB 外围设备模块（左图）和 PCI Express 插件模块（右图）。

在其他情况下，如 PXI (面向仪器的 PCI 扩展)—70 多个公司支持的用于测试、测量和控制的坚固平台—测量仪器安装在工业机箱中（如图 2 所示）。

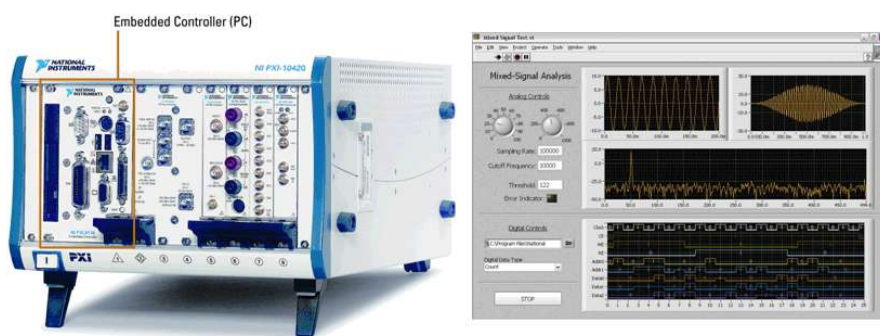


图 2. 采用 PXI 硬件和 NI LabVIEW 系统设计软件模块化仪器系统范例。

PXI 系统中的主机可以嵌入在机箱内（如图 2 所示），也可以是一个单独的笔记本电脑、台式电脑或是通过有线接口控制测量硬件的服务器。由于 PXI 系统使用的是与 PC 内部总线相同的总线—PCI 和 PCI Express—以及商业现成 PC 组件来控制系统，因此，无论是使用 PXI 系统还是 PC，均可作为模块化仪器系统的硬件平台。（但是，PXI 为模块化仪器系统提供了一些独有优点（此处未介绍），如更高的通道数、便携性和坚固性。关于 PXI 的更多信息，请查看 ni.com/pxi。）无论系统使用的是 PXI、配有内插入式模块或是带有外围设备 I/O 模块的台式计算机，这种共享机箱和控制器的方式都可大幅降低成本，同时还支持用户控制测量和分析软件。尽管模块化仪器存在多个配置可供选择，但该类型仪器与传统仪器的区别在于其软件是开放式的，用户可在测试需求更改时或进行传统仪器无法完成的测量功能时定义自己所需的测量。

值得注意的是，这并不意味着与将各种功能集合在单一外盒中的传统仪器相比，此模块化方法会存在仪器或通道间同步的问题。相反，模块化仪器的设计目的在于可被集成，以供系统使用。所有模块化仪器通过共享时钟和触发器来实现定时和同步功能。例如，就最高同步精度而言，基带、IF 和 RF 仪器可以实现仪器间偏移低于 100 ps 的相互同步—性能优于同一仪器上的多通道间的偏移。

模块化可降低成本、减小尺寸、提高吞吐量、延长生命周期

尽管“模块化”这个词有时会狭隘地只用于描述硬件的封装，但是模块化仪器所涵盖的内容远不止封装。用户可以期待模块化仪器系统带来的好处有三方面：通过共享机箱、背板和处理器来降低成本和体积；通过与主机处理器的高速连接来加快数据处理能力；通过用户定义的软件提高灵活性和延长生命周期。

如上所述，模块化仪器系统中的所有仪器共用一个电源、机箱和控制器。而分立式仪器则为每个仪器重复配置电源供应、机箱和/或控制器，从而增加了成本和体积，降低了可靠性。事实上，无论使用的是哪种总线，每个自动化测试仪器均需要一个 PC；模块化架构通过所有仪器共享处理器来分摊整个系统的成本。在模块化仪器系统中，GHz PC 处理器对数据进行分析，并使用软件完成测量。其测量吞吐量是仅由传统仪器搭建的测试系统的 10–100 倍，传统仪器使用的是内置的厂商定义固件和特定应用的处理器。例如，一个典型的向量信号分析仪(VSA)每秒可以完成 0.13 次带内功率测量，而一个 NI 模块化 VSA 每秒可以完成 4.18 次带内功率测量—速度提高了 32 倍。

模块化仪器需要一个高带宽、低时延的总线，实现从仪器模块到共享处理器的连接，以执行用户定义的测量。虽然 USB 在易用性方面提供了卓越的用户体验，但 PCI 与 PCI Express(以及基于这些总线拓展而得的 PXI 平台)在模块化仪器系统中提供了最佳的性能。PCI Express 现在可提供高达 4 GB/s 的插槽带宽，而 PXI 每个插槽的带宽高达 2 GB/s – 超过高速 USB 的 33 倍，是 100 Mb/s 以太网的 160 倍，甚至是新推出的千兆以太网的 16 倍（如图 3 所示）。外设总线(例如 LAN 与 USB)通过一个内部总线(例如 PCI Express)与 PC 处理器相连，因而性能较差。下面举一个高速总线如何影响测试与测量的范例——模块化 RF 采集系统。台式计算机或 PXI 系统中带有 4 个 2 GB/s 插槽的 PCI Express 可以将 2 个通道的 100 MS/s、16 位 IF(中频)数据以数据流的方式直接传输到处理器中进行计算。由于 LAN 与 USB 都不能满足这些需求，如果仪器要达到此性能，则需采用厂商定义的嵌入式处理器来进行测量——而这样的仪器就不再是模块化。

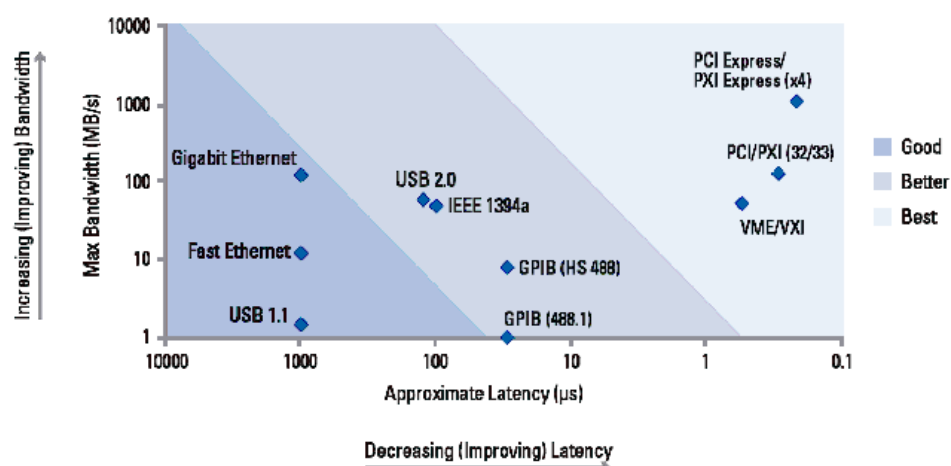


图 3. PCI 和 PCI Express 提高最高的带宽和最低的延迟，可降低测试时间，并通过用户定义的软件实现高灵活性和超长的生命周期。

在模块化仪器中，与主机的高速连接实现了该仪器的高灵活性和超长的生命周期，因为它支持软件驻留在主机上，而不是仪器上。通过主机上运行的软件，用户(而不是厂商)可以定义仪器的运行方式。这样的架构使用户能够：(1)进行那些由于通用性不强而没有包含在厂商定义的典型非模块化方法中的测量；(2)为尚未发布的标准创建测量；(3)定义用于进行特殊测量的算法。软件的用户定义特点也意味着用户可在待测设备发生变化时增加或调整仪器。用户也可以利用软件直接访问来跨网络监视或控制这些模块化仪器。

值得注意的是，这些硬件实现方式并没有牺牲测量性能。目前，利用模块化仪器系统方法开发的仪器包括业界分辨率最高的数字化仪、带宽最高的任意波形发生器和精确度最高的 7½ 位数字万用表。

软件

灵活的自定义测量软件

软件对于模块化仪器的重要性是不言而喻的。软件将来自硬件的原始比特流转化为有用的测量数据。设计完善的模块化仪器系统一般采用多层软件架构，包括系统服务、驱动、测试应用开发，和测试系统管理，如图 1 所示。



图 1. 多层软件架构通常应用于模块化仪器系统。

系统服务和驱动软件

底层架构是模块化仪器系统最关键的要素之一，但往往容易被人们忽视。该层架构包括驱动 I/O 软件和硬件配置工具。这一驱动软件的重要性在于它将测试开发软件和用于测量和控制的硬件相连接。

仪器驱动为与仪器的连接提供了一系列高级、用户可识别的功能。每个仪器驱动专为特定仪器模型量身设计，以连接仪器模型的独特功能。仪器驱动尤其重要的一点是与开发软件集成，使仪器命令无缝融合到应用开发中，成为应用开发一部分。系统开发人员所需的仪器驱动接口应专门针对开发人员可选择的开发环境（如 NI LabVIEW、ANSI C、C++或 Microsoft .NET）进行优化。

配置软件工具包括各种用于配置和测试 I/O 以及存储缩放、校准和通道混叠信息的资源。这些工具对于快速搭建和维护仪器系统、解决系统故障至关重要。

测试开发软件

测试开发软件层的软件提供了开发应用代码或流程的工具。尽管模块化仪器系统并不一定使用图形化编程，但由于图形化工具易于使用、便于快速开发，因而通常应用于这些系统中。图形化编程使用“图标”或符号函数，以图形的形式来表示所执行的操作，如图 2 所示。这些符号通过传递数据以及决定执行顺序的“连线”连接起来。LabVIEW 提供了行业内应用最为广泛且最完整的图形化开发环境。

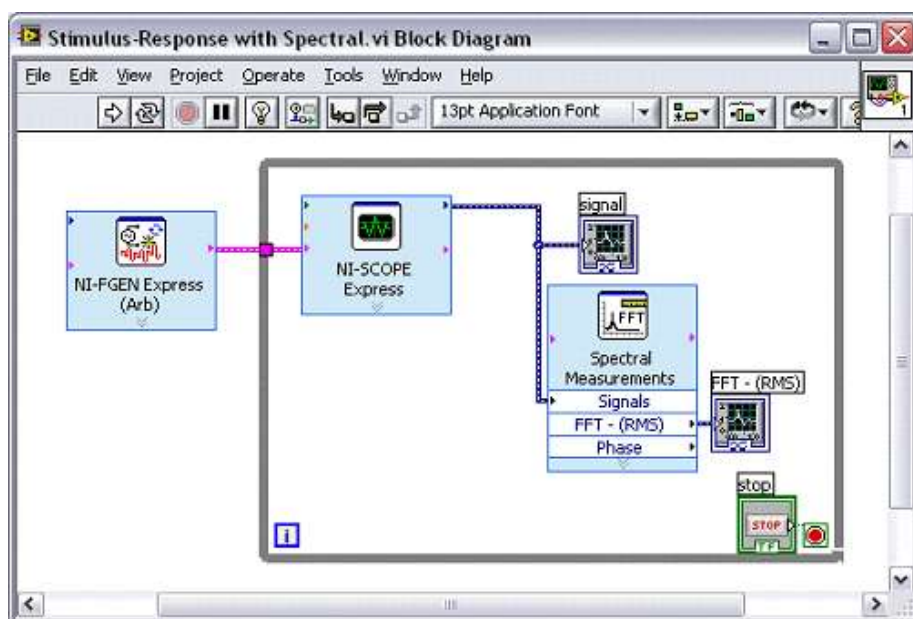


图 2. 使用模块化仪器的典型激励/响应应用的 LabVIEW 代码用于 (1) 从任意波形发生器生成信号; (2) 通过数字化仪/示波器采集信号; (3) 执行快速傅里叶变换(FFT); (4) 在用户界面 (前面板) 上显示 FFT 曲线图。

测试管理软件

部分应用还需要一个额外的软件管理层，用于执行测试或将测试数据可视化。这一软件层就称为测试管理软件层。针对高度自动化的测试系统，测试管理软件提供了一个用于生成序列、分支/循环、报告生成和数据库集成的架构。测试管理工具还必需与创建特定应用代码的开发环境紧密集成。例如，NI [测试管理软件](#)，提供了用于生成序列、分支、报告生成和数据库集成的架构，并与所有常用开发环境相连接。对于其他需要对海量测试数据进行可视化的应用，可采用其他更有效的工具。这些应用需求包括快速访问海量离散数据，连续报告生成和数据可视化等。这些软件工具可辅助管理、分析和报告数据采集时获取的数据和/或仿真时生成的数据。

