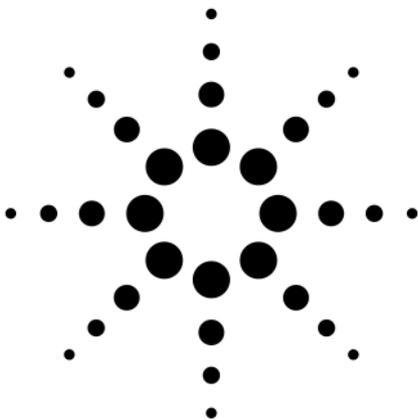


测试系统中开关 / 测量 解决方案的比较



应用指南

这篇应用指南比较了在功能测试和数据采集环境中所使用各种开关 / 测量解决方案的功能特性、执行速度和软件开发的容易程度。所考察的产品包括在 Visual Studio .NET 应用开发环境中的 Keithley 27xx、Racal 1256、Agilent 34970A 和 Agilent 34980A 开关 / 测量单元, 以及 Agilent 3499A/34401A 组合、Agilent E1411B/E1476A VXI组合和National Instruments PXI-4070/SCXI-1128 PXI 组合。

开关 / 测量系统特性

测试系统中存在各种复杂的测量功能,但系统总要包括两种部件——数字万用表(DMM)和多路继电器开关。其核心功能是开关和测量。可用三种不同方法实现这一核心功能:

- 用电缆互连的分立仪器, 如 Agilent 34401A 台式数字万用表, 它们连接到 Agilent 3499 A/B/C 开关机箱或 Racal 1256 开关机箱。
- VXI、PXI 或 PXI 混合主机, 如 Agilent E1411B VXI DMM 和 E1476A VXI 开关卡, NI PXI-4070 DMM 和 SCXI-1128 开关卡(后者为混合配置, 其控制信号由 PXI 背板和 SCXI 背板共享, 它们在一个组合机箱中, 或通过 PXI 机箱至独立 SCXI 机箱的电缆装置)。

- 采用包含 DMM 和各种开关卡的专用仪器, 如 Agilent 34980A、Agilent 34970A 和 Keithley 2701 开关 / 测量单元。

每一类解决方案都有各自的所长和所短,我们将对此作深入讨论。这些设备的使用方式可分成两大类: 数据采集和功能测试。您所选的解决方案与您的使用方式有极为密切的关系。

数据采集和功能测试的差别

在这两类测试系统中可使用同样的电压表和开关:

- **数据采集(DAQ)**, 通过获取大量数据表征设备的机械、电气或电子性能, 例如航天器重返大气层时要测量数千点的温度。
- **功能测试(EFT)**, 向被测装置(DUT)的电子设备施加激励, 监视输出响应是否与预期响应一致, 把结果与设置极限比较, 例如确定汽车的引擎控制模块是否正常工作。



Agilent Technologies

在这两种环境中,同样硬件所能实现的吞吐率有很大差异。DAQ模式为执行扫描测量对电压表和多路开关一起编程——把开关断开/闭合状态表下载到开关盒,硬件握手线把DMM测量和相应开关设置链接到一起。来自电压表的测量完成触发把状态表往前推进,或让它自由运行。DMM在发送测量完成触发后等待预先编程的延迟时间,也可等待来自开关被称为推进触发的硬件信号,该信号说明开关已进入其编程状态,并正常获取读数。硬件握手使最小化仪器间传输的I/O量成为可能。此外,每次测量不需要再花设置时间;设置信息被下载,然后通过一条初始化命令完成全部操作。因此执行速度也很高。根据要求的测量分辨率和开关速度,可达到1,000读数/秒的量级。在这样的系统中,因为要传输大量数据,所以高速背板能够提高吞吐率。VXI和PXI在这里可大显身手。

而FET系统的情况则有所不同。必须施加激励和进行测量,为覆盖DUT上的所有引脚,这一过程要重复多次,因此必须进行重复的激励和测量仪器读数以及重新配置系统(闭合继电器、取读数、断开继电器)。通常要通过轮询保证读数已就绪,这就会增加额外的I/O执行时间。因此执行速度通常为500读数/秒。由于只有较少的读数,选用高速背板并无太大意义,低价的专用开关_测量单元则是更好的选择。

卡箱结构

市场上所有开关系统都采用卡箱式结构,并提供各种类型的插卡。有些采用开放体系结构,这意味着背板接口各符合相关的工业标准,并可由多家厂商实现,如VXI和PXI。另一类采用特定厂商的内部背板,如NI的SCXI, Racal的1256, Keithley的27XX,以及Agilent的3499家族、34970A和新的34980A。许多厂家这样做的原因是供应商和用户不必提供和需要开放标准所要求的额外能力。而仅为它们的开关卡在机箱中提供刚刚

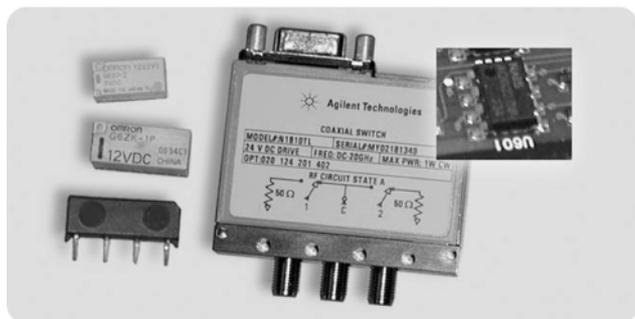
足够的功率、冷却和背板速度。这类体系结构一般也提供带有已实现地址译码器的实验电路板,从而为用户增加自己电路保留了空间。

除了多路继电器开关和DMM外,由于它们通常是在一起使用,因此开关/测量系统中一般还实现一些其它功能。即:

- 数字I/O (DIO)
- 数模转换器(DAC), 有时能快到可作低频的任意波形发生器使用
- 频率计数器/总和器
- 等温的终端接线盒
- 利用实验电路板可定制插卡, 如用于信号调理, 通常为应变片测量所要求

各种类型的开关卡可作为多路开关、矩阵和通用开关配置。它们是:

- 低带宽(DC-100 MHz)
 - FET继电器(高开关速度, 高导通电阻, 低电压和电流)
 - 干簧继电器(中开关速度, 低导通电阻, 中电压和电流; 低热电动势)
 - 电枢继电器(低开关速度, 低导通电阻, 高电压和电流; 低热电动势)
- 高带宽(高达18 GHz)
 - 射频开关(低速、低导通电阻, 低电压和电流)
 - 微波开关(低速、低导通电阻, 低电压和电流)



易于编程

应用开发环境

任何测试系统都需要编程仪器。常用的应用开发环境(ADE)有:

- 图形语言:
 - Agilent Technologies VEE Pro
 - National Instruments LabVIEW
- 文本语言:
 - Microsoft Visual Studio.NET (VB/VC++ 7.0, C# and many more)
 - National Instruments LabWindows/CVI
 - Previous versions of Microsoft languages (VB/VC++ 6.0, etc.)
 - Rocky Mountain Basic (“HP Basic”)

其中最常用的三种语言是 VEE Pro、LabVIEW 和 Visual Studio 6.0, Visual Studio.NET 的使用在迅速上升。Agilent 为 Visual Studio .NET 环境提供的软件产品是 T&M Programmers’ Toolkit, NI 是 Measurement Studio, 这两种产品都使仪器使用更容易。在本应用指南进行的基准测试中, 所有仪器都用 Agilent T&M Toolkit 和 NI Measurement Studio 在 Visual Studio .NET 中编程。

驱动程序 vs. SCPI

独立仪器通常需要在接口, 例如 GPIB、USB 或 LAN 上发送 ASCII 命令。这类命令已经实现标准化, 并被称为 SCPI(可编程仪器的标准命令)。它们可通过直接调用对相应接口讲的底层功能, 或通过高级功能调用发送到仪器。软件中包含的这些高级调用被称为驱动程序。您可自己编写驱动程序, 也可使用制造商或他人提供的驱动程序。

卡箱式系统, 例如 VXI 和 PXI 更经常地通过数据寄存器, 而非 ASCII 命令控制(VXI 有些信息插卡依赖作为 VXI 规范组成部分的内置串行协议)。虽然可按上面所述的方法通过编写程序用同样的底层接口功能调用把必要的数发送到这些寄存器, 但通常习惯是使用由制造商提供的驱动程序。也可向某些基于寄存器的仪器发送 SCPI 命令, 安捷伦为它的一些 VXI 卡提供可下载的 SCPI 驱动程序(D-SCPI)。这些代码驻存在 VXI 主机 0 槽 GPIB 控制器(E1406A 命令模块)的 Flash ROM 中。安捷伦也为它的 VXI FireWire 接口提供解释 SCPI(I-SCPI)。I-SCPI 一般比 D-SCPI 快得多, 因为它使用更快的 PC 命令分析。PXI 不提供类似设施; 所有 PXI 卡均为基于寄存器的插卡, 因此需要制造商提供的驱动程序。为 VXI 和 PXI 仪器提供的驱动程序是经编译的 DLL, 它可由任何语言使用。也可作为 Agilent VEE Pro 或 NI LabView 的专门驱动程序提供。

除了要求驱动程序的 PXI 外, 多数驱动程序不包括所有可能的仪器功能。一般是“贯通”功能, 即允许驱动程序中未实现的 SCPI 命令从驱动程序发送到仪器。

在 Visual Studio 环境中, Microsoft 的 IntelliSense 特性使驱动程序的使用非常容易, 因为程序员会从下拉菜单表中选择易于理解的名称, 观看功能及其参数说明。

在考虑选择是用 SCPI 还是用驱动程序时, 驱动程序的优点有:

1. 对其他程序员来说, 程序应该更易移植和有更高的可读性。虽然 SCPI 语法广为人知, 但要精确确定则需要经常查看仪器编程手册。还应注意驱动程序能为您最终产生 SCPI 命令, 因此在您调试应用时, 总需用各种方式掌握它。

2. 在开发期间通过IntelliSense (VB6和.NET)的内置帮助。
3. 如果有内置的状态高速缓存, 则可改进执行速度 (下面将更详细讨论)。
4. 可能已为您考虑了仪器的独特之处, 从而减少需要提供支持的概率。

除了上面已提到的PXI外, 驱动程序的主要缺点是不可能不包括全部功能。此外, 如果驱动程序提供不正确的功能, 则是极难调试的。使用 Agilent 测试和测量工具集的 I/O 监视器将简化这项任务, 它可观察实际向仪器发送的 SCPI 命令。但这也意味着您必须首先掌握 SCPI。驱动程序的发布和支持是由仪器制造商, 而非独立第三方提供。

接口驱动程序

在您的程序能够把命令传送到设备之前, 必须编写控制硬件接口, 如 LAN、USB、GPIB、MXI-3 和 FireWire 的驱动程序。管理这些硬件的底层驱动程序作为厂商 I/O 库的组件提供。NI 把它叫做 Passport 驱动程序, Agilent 叫做 Tulip 驱动程序。因为您并不直接使用它们, 因此我们这里不作详细讨论。但您应知道在同样环境中使用 NI 和 Agilent 硬件和软件时, 它们正在为您工作。NI 软件能够访问 Agilent 接口硬件, 同样 Agilent 软件也能访问 NI 接口硬件。详细说明见 Agilent I/O 库在线文档。

VISA, VXI 即插即用和 IVI

软件驱动程序是分层组织的。最简单的标准化层为 VISA (虚拟仪器软件体系结构, VISA 是否是真正驱动程序有不同见解, 我们这里不作争辩)。它允许通过 viRead 和 viWrite 这类简单 C 函数调用把 SCPI 命令发送到仪器, 以及通过 viPeek 和 viPoke 这类 C 函数调用把二进制命令发送到寄存器。Agilent 和 NI 是该软件的两家主要提供商。Agilent 还提供 SICL (标准仪器控制库), 把它作为 Agilent VISA 实现的底层代码使用。通过向“COM 友好”环境, 如 Visual Studio 提供它的 VISA C 函数调用, Agilent 也采纳工业标准 VISA 体系结构。[COM 是组件对象模型的缩写, 它是 Microsoft 的封装库标准, 使接口更易连通和维护。]

包括大多数 VXI 仪器和所有 PXI 仪器的基于寄存器的设备都需要驱动程序, 除非开发者选择使用前面所述的 viPeek 和 viPoke 命令, 不过这是非常繁琐的过程。为此建立的 VISA 顶层称为 VXI 即插即用。虽然名称中带有 VXI, 但这一概念早已延伸到非 VXI 仪器, 包括大多数台式可编程仪器。

VXI 驱动程序是专门用于仪器的, 仪器名包含在功能名中。典型 C 语言例子为:

```
hp34401_read_Q (vi, readings, numReadings);
```

(当安捷伦还是惠普公司的一部分时, 就已编写了许多 Agilent 驱动程序, 为保持后向兼容性而未改变驱动程序名。)

由于许多产品有类似的功能性,因此没有必要复制从 VXI 即插即用标准得到的功能。为此可把各种仪器归入若干门类,如数字万用表、示波器、函数发生器和开关,用于这些门类的驱动程序被命名为IVI(可互换虚拟仪器)。这些驱动程序的定义已由称为IVI基金会的组织作了标准化。有两类IVI:

- IVI-C, 由NI提出。根据所使用的语言(C、LabView等)有不同版本。典型C语言例子为:

```
lviDmm_ReadMultipoint (vi, maxTime, arraySize, readingArray, &actualPoints);
```

- IVI-COM, 由Agilent提出, 由于它有对许多开发程序环境的适应性, 因此各种环境(所有Microsoft COM环境, 包括 VEE、LabView、Visual Studio 等)不需要不同版本。Visual Basic 例子为:

```
myDMM.Measurement.ReadMultiPoint(maxTime, readings)
```

在这两个例子中, 可通过使用特定仪器的功能调用, 管理特定门类功能未映射到仪器某种可用功能性的情况。这类功能性的使用去除了IVI的“可互换”, 这就是为什么有时买来的仪器缺少可使用的功能性。您必须确定是否要求仪器的可互换性, 并自己编写代码。

VXI即插即用驱动程序和IVI驱动程序为无面板仪器提供建立软前面板的代码, 这是独立的可执行程序。您不能把它集成到自己编写的仪器控制程序中。但其中有一些能生成代码片段, 您可把它剪切和粘贴到特定开发环境中。

VXI即插即用驱动程序也带有功能面板, 从而简化了从Agilent VEE Pro 和 LabView 的C函数调用。

LabVIEW 和 VEE Pro 驱动程序

LabView使用NI特定类型的图形驱动程序, 它的正式名称是G框架VXI即插即用驱动程序。也可表示为

“G”、“G-Win”、LabVIEW 或 “LabVIEW Certified Plug and Play” 驱动程序。但由于LabView也使用Windows框架(基于C)VXI即插即用驱动程序, 因此在您能够打开和观看前, 通常难以知道正在下载的是哪种类型的驱动程序。

Agilent VEE Pro 能使用各种类型的驱动程序, 包括VXI即插即用(Windows框架)、IVI-C和IVI-COM驱动程序。过去安捷伦开发称为“Panel”的驱动程序, 这是专门针对VEE Pro的特定类型驱动程序。今天, 安捷伦已经为所有新仪器标准化了对COM和.NET友好的驱动程序, 从而为客户提供更多的开发环境选择。

速度如何?

基准测试通过使用硬件的实验表明, 在现代计算机中, VXI即插即用驱动程序和IVI驱动程序可能于SCPI(通过VISA)有相同的速度。虽然尚有争议, 但不可避免驱动程序本身对执行速度的影响。

无论您采用何种仪器编程的方法, 命令的次序都会在仪器程序中产生微妙的问题。如果您使用SCPI, 就需要知道哪些命令会造成仪器其它状态的改变。驱动程序可为您关注这一问题, 但需要向仪器发送确定所处状态的大量查询命令, 或通过发送额外命令让仪器处于给定状态。这将增加相当多的执行时间。高速缓存能解决这一问题, 它通常在IVI驱动程序中实现, 您也可自行实现高速缓存。状态缓存驱动程序保持对仪器当前状态的跟踪, 命令不会造成状态改变, 它不向仪器发送命令, 从而节省了命令传输和命令分析时间。

一般可通过向仪器发布复位命令, 使其恢复到已知状态而排除状态问题。但仪器要花相当长时间执行复位命令, 因此较好的方法是仅在程序首次开始运行时发布复位命令, 在程序运行中仔细管理仪器状态, 以保证程序结束时仪器仍处于相当复位命令的状态。通常也在产生错误时发布复位命令。

这里是冗余命令增加执行时间的一个例子: 假定您用SCPI编程, 通过发送命令“CONF: VOLT DC 10,.001”告诉DMM进入DC功能, 设置10V量程和1mV分辨率。GPIB的最高速度为1MB/s, 这21个字符的串(包括回车符)要花21 μ s, 函数调用的成本要在现代PC中约需另增10 μ s。仪器将以相对慢的处理器分析这一命令。例如34401A DMM的12MHz处理器大约还要花21ms。因此如果DMM已经处在该状态, 这21.031ms就是无需花费的时间。虽然看起来好像不多, 但典型测试程序要读成百上千个DMM读数, 因此最大化仪器I/O和相继处理的效率是十分重要的。您可在花费额外程序时间的情况下自己实现状态缓存。这例子也说明为什么对小的数据传输来说, LAN和USB并不显得比GPIB快。在所有情况下, I/O传输速度都远远快于仪器处理时间, 因此接口的选择并无太大关系。

Agilent 34980A(下图)是使用快得多内部微处理器的新型仪器。它执行同样字符串只需1.5ms, 因此不需要状态缓存。



公共开发环境

本应用指南为比较各种产品在基准测试中的表现, 有必要使用一种公共的开发环境。虽然LabView和VEE Pro都是流行的图形环境, 但许多制造商的测试是在文本环境进行。VB6、C或其衍生语言(VC++、LabWindows/CVI)在这一领域占有主导地位。也有一些用Microsoft Visual Studio .NET环境开发新应用程序的理由。Agilent和NI都已发布新的工具(Agilent的T&M Toolkit, NI的Measurement Studio), 以帮助测试系统开发者在该环境更快地创建程序。通过对老程序的支持减少, Microsoft鼓励开发者升级到.NET。VB6程序也可使用.NET内置移植工具移植到VB.NET环境。因此, 我们把VB.NET选作公共开发环境。

我们还要求NI、Keithley、Racal和Agilent硬件要在真实测试系统中一道工作。因此测试系统用所有这些产品构成, 如图1所示(第7页)。一些仪器使用LAN——Agilent 34980A和Keithley 2701, 通过LAN集线器。用MXI-3连接PC与PXI机箱。用FireWire连接PC和VXI机箱。由于VXI机箱也可通过GPIB命令模块控制, 所以还通过专用PCI GPIB接口卡连接。其它仪器使用Agilent 82357A USB/GPIB转换器经GPIB控制。

图1还示出各测试设备的相对高度, 从34401A DMM的2EIA单位(一个EIA单位或“机架单位”为1.75英寸)到“3U”PXI/SCXI机箱的4个单位(PXI卡为3U, 但机箱为4U)。这里不打算比较VXI机架尺寸。为方便起见使用的是4槽主机(3U), 虽然大型测试系统更普遍使用的是13槽主机(7U)。还应指出虽然今天新的系统趋向用非VXI硬件制造, 但对于只有VXI才具备的特殊功能卡, 4槽主机仍不失为一种好的解决方案。

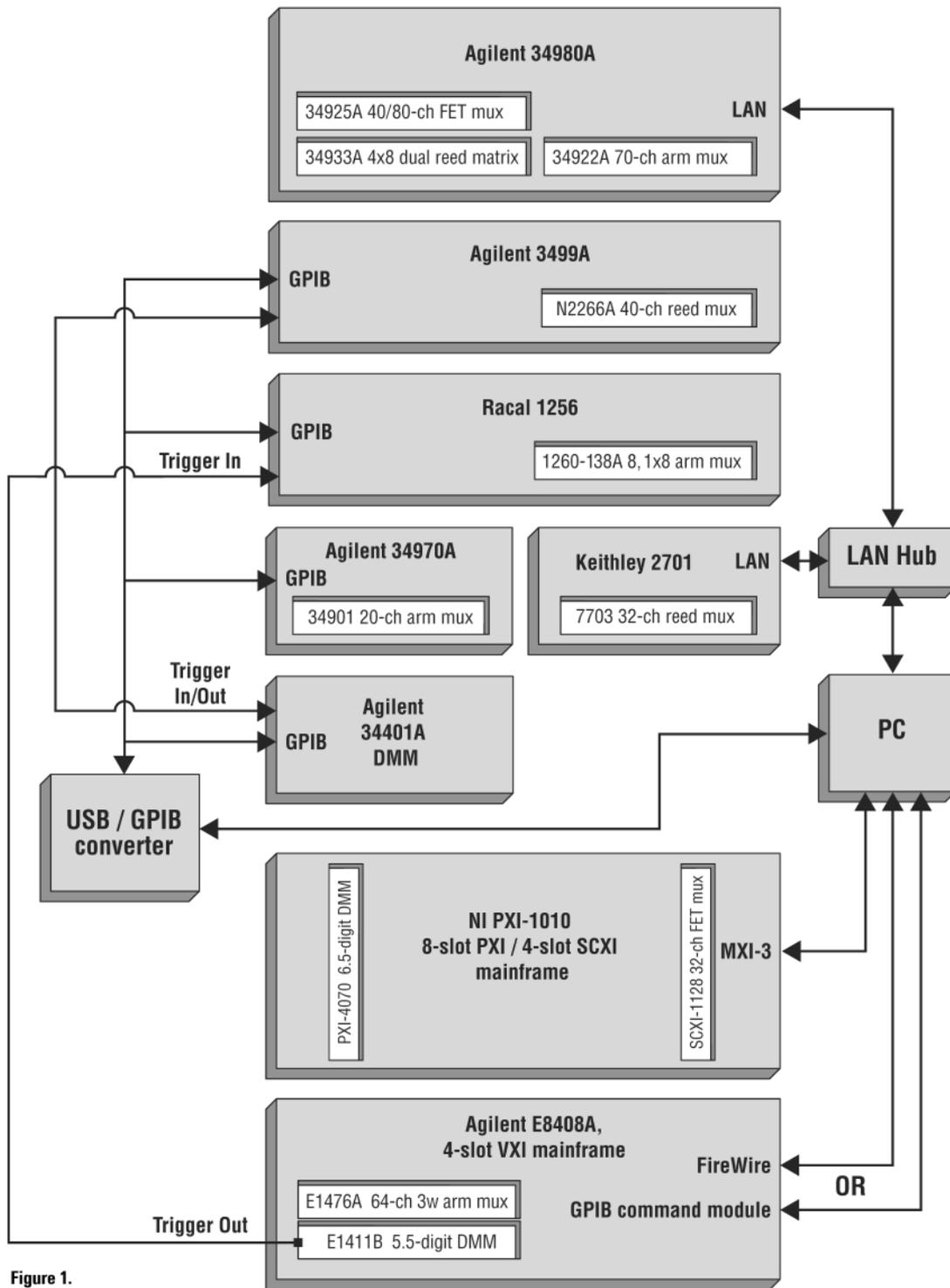


Figure 1.

基准测试结果

在 VB.NET 中编写的程序控制数据采集模式(扫描电压读数)和 EFT 模式(使用相应开关的单个读数)的所有仪器。下述软件安装在 Compaq Evo 中, 计算机使用 2.4 GHz Pentium 4 和 512 MB RAM。

- Agilent 的 T&M Toolkit 1.2
- 上述 Agilent 软件装有 I/O 库, 包括 SICL 和 Agilent VISA, 用于所有 Agilent 仪器的 .NET 包装程序, 带有 VXI 即插即用驱动程序, 包括 E1411B VXI DMM 和 E1476A VXI 开关。
- 从 Agilent 开发网(AND)网址为 E4411B 和 E1476A 下载的 VXI 即插即用驱动程序。
- 从 AND 网址为 E4411B DMM 和 E1476A 开关下载的 D-SCPI 驱动程序。
- 从 AND 网址为 34401A DMM、3499A 开关单元及 34970A 开关 / 测量单元下载的 IVI-COM 驱动程序。
- 从 ADN 网址下载的 HP VIC (不因 HP 和 Agilent 的拆分而重命名), 为使用 D-SCPI 调用初始化 GPIB 命令模块。
- NI Measurement Studio.NET 7.2
- 从 Keithley 网址为 2701 开关 / 测量单元下载 IVI-C 驱动程序。
- 从 Racal 网址为 1256 开关单元下载 IVI-C 驱动程序。
- 从 NI 驱动程序安装盘安装 NI-SWITCH、NI-DAQ、NI-DAQmx 和 NI-DMM IVI-C 驱动程序, 运行 Toolkit Driver Wrapper Wizard 建立 .NET 兼容调用。

基准定时测试结果:

EFT 时间 = 断开和闭合一个继电器, 用 1 mV(4.5 位)分辨率在 10 V 量程触发和读一个 DMM DCV 读数的时间。报告的时间是对 20 次这样测量的平均。

DAQ 时间 = 扫描 20 个通道。当各继电器闭合时进行一次测量和读取结果的时间。

	驱动程序		SCPI	
	EFT	DAQ	EFT	DAQ
Agilent 34980A				
70 通道电枢多路转换器 (34922A)	16.9 ms	10.4 ms	15.1 ms	10.1 ms
双 4 × 8 干簧矩阵 (34933A)	9.6 ms	1	8.4 ms	1
40/80 通道 FET 多路转换器 (34925A)	10.0 ms	2.7 ms	7.9 ms	2.7 ms
Keithley 2701-2703				
32 通道差分, 干簧多路转换器	440 ms	68 ms	n/a	n/a
Racal 1256-138A/E1411B				
8 1 × 8 双线电枢多路转换器	4.13 ms	113 ms ²	n/a	n/a
Agilent 34970A + 34901A				
20 通道电枢多路转换器	52.2 ms	22.8 ms	70.0 ms	25.2 ms
Agilent 3499A - N2266A/34401A				
40 通道干簧多路转换器	29.5 ms	21.4 ms	35.3 ms	27.1 ms
Agilent VXI E1476A/E1411B				
64 通道 3 线干簧多路转换器	30.1 ms	11.9 ms	46.6 ms ³	2.94 ms ³
NI SCXI-1128/PXI - 4070				
32 通道 FET 多路转换器	12.8 ms	2.91 ms ⁴	N/P	N/P

注:

- 1 用矩阵不可能进行“扫描”测量。这不是矩阵的典型使用方法, 它一般用在 EFT 测试中。
 - 2 Racal 1256 要花 2 秒钟处理“定义扫描表”命令。这就是数据采集时间如此长的原因。如果不计这一时间, 执行时间约为 13 ms。所使用的“触发延迟”参数是 10 ms, 因为推进触发输出不工作。如果去掉延迟, 执行时间约为 3 ms, 这与通常 DAQ 模式一致。
 - 3 用 E1406A GPIB 命令模块收集 VXI SCPI 数量。
 - 4 SCXI-1128 不能产生返回 DMM 的触发, 因此完整的握手是不可能的。SCXI 测量属同步型, 此时 DMM 用采样间隔取 20 个读数, 用测量完成信号把扫描推进到下一通道。因此执行时间决定于 DMM 采样间隔。报告数是采样间隔为 0.0 时的执行时间, 它表示杂项开销。随着采样间隔的加大, 报告时间将等于采样间隔。
- N/P 不可能。PXI 仪器需要使用驱动程序。

您可从这些数据发掘出一些有趣的信息。如前所述, 数据采集模式有快得多的执行时间, 有时可快一个数量级。而且执行时间有很宽的变化范围; 例如虽然 Keithley 使用快速的干簧继电器, 但并不能给执行时间带来多大帮助, 因为仪器取读数是很慢的。PXI 是最快的, 但由于它只使用 FET 开关测量, 其性能约与

34980AFET 开关相当。如果代之使用干簧或电枢继电器, 其执行时间也会加大。此外还与驱动器的实现方式有关, 使用状态缓存要快于相当的 SCPI。

产品比较

下表示出可在各种平台使用的模块类型。

	Agilent 34980A	Agilent 34970A	Agilent 3499A	Agilent VXI	NI VXI/SCXI	Keithley 2701	Racal 1256
矩阵开关							
FET	无	无	无	无	4 × 6 2 w 4 × 8 2 w	无	无
干簧	2, 2 × 8 2 w	无	无	4 × 32 2 w	4 × 16 2 w 4 × 32 2 w 4 × 64 2 w	无	无
电枢	2, 2 × 8 2 w 2, 4 × 16 2 w	4 × 8 2 w	4 × 4 2 w 4 × 8 2 w	16 × 16 4 × 16 8 × 32 8 × 8 4 × 16 2 w	4 × 6 2 w 8 × 16 2 w	6 × 8 2 w	12 × 12 2 w (7 种配置)
多路转换器							
FET	40 通道 2 w	无	8 通道 2, 4 通道 4, 2 通道 se	16 通道 3 w 32 通道 16 通道 3 w	24 通道 2 w 32 通道 2 w	20 通道	无
干簧	40 通道 2 w	20 通道 2 w	40 通道 se	16 通道 3 w 48 通道 se 64 通道 3 w 256 通道	64 通道 2 w 128 2 w 256 2 w	32 通道 2 w	42 通道(干簧或水银湿簧)
电枢	40 通道 2 w 27 通道 2 w	20 通道 2 w 40 通道 1 w	10 通道 2 w 20 通道 2 w 40 通道 2 w	64 通道 2 w		20 通道 2 w 32 通道 2 w 40 通道 2 w	23 通道 2 w 16, 1 × 4 8, 8 通道 2 w
射频	4, 1 × 4 50 或 75 Ω	2, 1 × 4 50 或 75 Ω	2, 1 × 4 2, 1 × 6 1 × 9 50 或 75 Ω	2, 1 × 4 6, 1 × 4 50 或 75 Ω	4, 1 × 4 8, 1 × 4 1 × 16 1 × 32 50 Ω	2, 1 × 4	10, 1 × 4 2, 1 × 4 1 × 6 2, 1 × 6
通用开关							
电枢	28-C/4-A 20 通道 A	20 通道 C	7/8/10/20 通道 C 40 通道 A	16 通道 C 32 通道 C 40 通道 A 64 通道 A	8/16/32/64 C 16/31/100 A	40 通道 A	12 A 12 B 12 C 20 DPDT 52 C 80 A, 低/高电压版
射频	2 通道 SPDT 3 通道 SPDT		3 通道 SPDT, 继电器驱动器	18 GHz 3 通道 SPDT 继电器驱动器	32, 64 继电器驱动器	无	20 通道 SPDT 50/75 Ω 2 SPDT 5 SPDT 2 × 2 基准开关

下接第 10 页

注释:

A = A 型
B = B 型
C = C 型
1w = 1 线
2w = 2 线

产品比较 续

	Agilent 34980A	Agilent 34970A	Agilent 3499A	Agilent VXI	NI VXI/SCXI	Keithley 2701	Racal 1256
DMM	6 1/2 位内置	6 1/2 位内置	外部	5 1/2 位或 6 1/2 位	5 1/2 位或 6 1/2 位	6 1/2 位内置	外部
数字 I/O	64 通道	在多功能卡上	16 位, 32 位 TTL	4, 8 位 72 通道输出 96 通道 dio 64 通道隔离输入	11 种插卡	在多功能卡上	48 oc 96 (ttl, cmos, oc)
DAC	4 通道隔离 任意波形	在多功能卡上	在多功能卡上	4 通道, 8/16 通道		在多功能卡上	无
计数器 / 总和器	在 DIO 卡上	在多功能卡上	在多功能卡上	E1333A: 2 通道 @100MHz +1 通道 @1GHz	4 通道或 8 通道, 至 125 MHz	在多功能卡上	无
多功能	32 dio 2 ±12 V DAC 100 kHz 总和器	16 位 dio 26 位计数器 2,16 位 dac	15 gp/16 dio 4 × 4 mat/16 dio 2 DAC/16 dio	M 模块	数化器, 计数器, dio, dac	20 通道 2 w mux, 2 DAC, 16 dio, 1 计数器 10 通道 2 w mux 32 dio	无
模拟总线	内 4 总线 2 w	无	无	无	PXI: 无 SCXI: 3 总线 2 w	DMM Hi/Lo	DMM Hi/Lo
其它	电路实验板	无	无	示波器, 任意 波形发生器等	示波器, 数化器, 任意波形发生器等	无	电路实验板

使用和维护成本

表格右面示出几种开关 / 测量解决方案的价格。这里未包括另需 DMM 和机箱的费用。对作为高速仪器 PXI 和 VXI 来说, 还需支付昂贵的机箱, 以及在计算机端和机箱端的接口卡费用。这将增加摊到每一插槽的成本。此外, 建立分立插卡解决方案要比使用开关 / 测量单机花更多时间。下面还将对此作详细讨论。使用 VB.NET, 只需几分钟时间就可在 34970A 和 34980A 上实现扫描测量。而使用 PXI 和 SCXI 作同样测量则可能需要 2 周时间, 还需拨打数个支持电话。

仪器 定价(美元, 2004 年 9 月 1 日)

Agilent 34980A	\$2350, 包括 DMM, LAN, USB, GPIB, 8 槽	(\$294/槽)
Agilent 34970A	\$1477, 包括 DMM, GPIB, RS232, 3 槽	(\$492/槽)
Keithley 2750	\$2995, 包括 DMM, GPIB, RS232, 5 槽	(\$600/槽)
NI PXI	\$5485 PXI-1042 8 槽机箱 (\$1995), 可用 6 槽 PXI-4070 DMM (\$1995), (占 1 个槽) MXI-4 接口 (PXI-PCI8331) (\$1495), (占 1 个槽)	(\$914/槽)
Agilent 34980A 插卡的售价为 \$495 至 \$2000。		
Agilent 34970A 插卡的售价为 \$340 至 \$501。		
Keithley 2701/2750 插卡的售价为 \$445 至 \$995, 加一块 \$1995 的微波模块。		
NI PXI 开关卡的售价为 \$495 至 \$1995, 加一块 \$4795 的高密度卡。		

不管从何种角度来看, 34980A 都是佼佼者。它要比许多其它解决方案容纳更多的插卡, 有覆盖更高频率范围的众多插卡选择, 有更多的计算机接口选项, 以及最低的每槽价格。它也支持更多的程序环境(通过 LabView 和 IVI-COM 驱动程序)。

易用性问题

我们在3周的评估过程中遇到了一些问题,可归纳为:

- 1. Keithley 固件需要升级。**如果 Keithley 2701 在未执行“close”的情况下程序中止或结束,仪器就不能继续使用,除非重新开机。Keithley 网址的FAQ中说 A06 版固件允许断开第二端口(2701),通过发送命令“KI2701”复位端口 1394 而使仪器继续使用。用于测试的仪器版本是 A04 版,因此从该网址下载 A06,仪器重写闪存。之后未再遇此问题,说明问题已得到解决。
- 2. 非寻常行为。**只要发送 ROUT:MULT:CLOS(闭合多个继电器)命令, Keithley 2701 就把 DMM 测试间隙时间复位到 SLOW(5 PLC)。因此需要增加一条把测试间隙时间复位到 .01 NPLC 的命令。在 DAQ 模式中还需通过配置语句设置分辨率。其它任何仪器都没有这样的行为,因此也可了解读数为何如此之慢。
- 3. NI-VISA 的设置需要改变。**如果使用 NI VISA,也就必须运行 MAX(测量自动化资源管理器)和使能 Tools->NI-VISA->VISA Options->Passports 中的“Passport to Tulip”接口驱动程序,否则就不能认可 Agilent VXI/GPIB 仪器。但安装 NI Measurement Studio 后不能使用该接口,因为 NI Passport 接口驱动程序调用 AgVisa32.dll 造成异常退出。NI 的支持者花一周时间解决这一问题。NI 的 Tulip passport 驱动程序编写者说这是一个已经知道的问题。解决办法是安装 NIVisaTulip.dll 的补丁版。它确实解决了问题,不过我们发现最新 NI 安装盘仍使用较老的版本。
- 4. NI IVI 符合性问题。**Agilent 驱动程序包装器向导只能包装按 IVI 基金会规范正确安装的 IVI-C 和 VXI 即插即用驱动程序。NI-DMM 和可能的其它 NI IVI-C 驱动程序不能正确安装,除非在安装步骤中检查“LabWindows/CVI examples”对话框。在评估时 NI 未为他们的仪器提供 .NET 兼容性,但提供了可从他们网址下载,并手动挂接到程序的 .NET 包装代码。这是为 NI-DMM 驱动程序所做的工作。但 NI 在包装器中选择了“InstrumentDriverInterop”名字空间,这与 Agilent 工具集已建的一个名字空间冲突。为解决这一冲突,把 NI 名字空间重新命名为“InstDvrInterop”。一种更为直接的方法是用快捷方式代替每次使用名字空间。例如在程序开始处规定“Imports Agilent.TMFramework”,所有框架内的子类就都可使用,而无需再用该前缀。一个这样的类是 InstrumentDriverInterop。因此您可规定:

Agilent.TMFramework.InstrumentDriverInterop.xxx

或简化的:

InstrumentDriverInterop.xxx

(这里 xxx 是下一功能层)但当把 NI 包装驱动程序与相同名字空间挂接时,必须解决冲突问题。

5. **不充分的文档。**NI 在 VB6、VC++ 和 LabView 中的例子是 PXI-4070 DMM 的使用。在 .NET 中的 DAQmx 开关例子不包括硬件触发，而只有软件触发。硬件触发要使用背板触发总线，使用参数需要字符串，但未给使用字符串的任何例子。IntelliSense 帮助弹出运行 javascript 的帮助，但它不能工作。运行 Start->Programs->NationalInstruments->NI-DAQ->DAQmx 文档手动检索需要的帮助。在使用给予代码中触发总线所需的名字时，也未发现它能工作。这里需要通过反复的试验。尝试过的名字有 PXITRig0、LBR_Trig0、TTL0 和 NI_VAL_TRIG_TTL。然后加载一个 LabVIEW 例子，以了解其作用。它展示需要新的命名约定，全部要基于路径——例如“/SC1 Mod3/TrigIn”。但在 LabVIEW 的例子中，没有一个名字能工作。因此开始拨打支持电话，得到的答复是 SCXI-1128 不支持握手，而只支持同时模式(单向触发)，无论是用较新的插卡，还是传统 DAQ。我们问 NI 如何能事先知道这一情况，回答是：“很抱歉，在购买前，如果不是问过别人或遍读帮助手册，是很难发现触发限制的。”
6. **在插拔 PXI 中的插卡时需要关掉 PC。**只有在 PC 关机时才能从 PXI 机箱拔出插卡，因为 PXI 背板是计算机内 PCI 总线的扩展。因此 PC 需要重启动，这是颇为费时的。
7. **版本冲突。**经常会遇到版本冲突。例如在查找 SCXI 开关问题时，会发现 NI-DAQ 7.1 安装在 Measurement Studio 7.0 上。所造成的各种问题只有先卸载所有 NI 软件，再重新安装才能解决。这会花多半天时间。
8. **如何从一个程序控制 Agilent 和 NI 硬件？**NI MAX 能找到所有 NI 接口上的设备，但不能直接控制 Agilent 接口，例如至 VXI, USB/GPIB 转换器或 PCI GPIB 卡的 FireWire 接口。Agilent 仪器资源管理器目前不能找到 PXI 设备。如何最好解决测试系统中这两家厂商设备间所需的通信呢？解决方案是在“side-by-side”模式中安装 Agilent I/O 库(见 I/O 库帮助文件)，然后使能如前所述 NI MAX 中的 Passport-Tulip 接口驱动程序。这能使 VISA 调用这些接口，实现 NI VISA 至 Agilent VISA 的路由，然后控制相关 Agilent 接口，直至允许 NI 接口，如 MXI-3 直接通过 NI VISA 和 Passport 驱动程序工作。

代码例子

这里是在 Visual Basic.NET 环境中对简单 EFT(闭合/测量/断开)和 DAQ(扫描)测量的编程要求, 我们使用 DC 电压 10 V 量程的 DMM 和 20 通道多路转换器。

从代码中可看到:

1. 带有内置 DMM 的开关 / 测量单元要作大量的 DAQ 测量。这是由于它为您完成了触发功能。虽然从表面看只需几分钟时间就能为这些仪器建立工作代码。但实际花了 2 周时间才使 PXI/SCXI 工作, 主要原因是难以理解触发要求。
2. 可通过级联 SCPI 串得到长串。这样做能够节省一点执行时间, 因为有了多次函数调用的额外成本。但正如前面所述, 如果无须在第一位置发送串, 那么在慢仪器中去掉命令分析时间必然会减少函数调用的执行时间。如果您的应用要求高吞吐率, 就应花时间评估状态缓存驱动程序和 SCPI 这两种方案。
3. IVI-C(带 .NET 包装器)、IVI-COM 和 VXI 即插即用驱动程序在 .NET 环境中的使用是非常接近的。它们都提供不同程度的 IntelliSense 帮助。仔细观看下面语句可看到 IVI-COM 有更好的帮助。PXI/SCXI 的帮助是不充分的, 需要经常查看在线手册和取得 NI 通过 e-mail 的支持。

所有使用 Agilent T&M Toolkit 的 VB.NET 程序都自动插入如下语句:

```
Imports Agilent.TMFramework
Imports Agilent.TMFramework.DataAnalysis
Imports Agilent.TMFramework.DataVisualization
Imports Agilent.TMFramework.InstrumentIO
Imports Agilent.TMFramework.InstrumentDriverInterop
```

在使用 NI Measurement Studio 时, 必须手工加入如下语句:

```
Imports NationalInstruments
Imports niDMM_32.NIDMMMeasurementConstants
```

Toolkit 在 “Public Class” 中自动增加声明:

VXI 即插即用驱动程序声明

```
Dim myHpe1476 As InstrumentDriverInterop.
VxipnpWrappers.Hpe1476
Dim myHpe1411 As InstrumentDriverInterop.
VxipnpWrappers.Hpe1411
Dim myKe2700 As InstrumentDriverInterop.VxipnpWrappers.
Ke2700
Dim myRi1256 As InstrumentDriverInterop.VxipnpWrappers.
Ri1256
```

Agilent 工具集驱动程序包装器向导包装的 IVI-C 驱动程序声明

```
Dim myniSwitch As InstrumentDriverInterop.IviCWrappers.
NiSwitch
```

Direct I/O(SCPI)declaratons

```
Dim myDSCPI As InstrumentIO.DirectIO
Dim my34980 As InstrumentIO.DirectIO
Dim my34970 As InstrumentIO.DirectIO
Dim my3499 As InstrumentIO.DirectIO
Dim my34401 As InstrumentIO.DirectIO
```

IVI-COM 驱动程序声明

```
Dim myAgilent34401 As Agilent.Agilent34401.Interop.
Agilent34401
Dim myAgilent34970 As Agilent.Agilent34970.Interop.
Agilent34970
Dim myAgilent3499 As Agilent.Agilent3499.Interop.Agilent3499
```

使用 NI Measurement Studio 时手工增加 IVI-C 声明

```
Dim PXIDMM As InstDvrInterop.Ivi.niDMM
Dim SCXI As InstDvrInterop.Ivi.niSwitch
```

这是仪器的初始化步骤:

对 E1411/E1476 组合的 VXI 即插即用驱动程序初始化:

单独 DMM 使用:

```
myHpe1411 = New InstrumentDriverInterop.VxipnpWrappers.Hpe1411("VXI0::24::INSTR", True, True)
```

组合 DMM/ 开关使用:

```
myHpe1411and1476 = New InstrumentDriverInterop.VxipnpWrappers.Hpe1411("VXI0::(24,25)::INSTR", True, True)
```

使用 GPIB 命令模块时 D-SCPI 与 E1411/E1476 的会话:

```
myDSCPI = New InstrumentIO.DirectIO("GPIB1::9::3::INSTR", False)
```

VXI 即插即用(经包装)驱动程序与 Ke2701 的会话:

```
myKe2700 = New VxipnpWrappers.Ke2700("TCPIP0::169.254.105.002::1394::SOCKET", False, False)  
myKe2700.Reset()
```

IVI-COM 驱动程序与 3499 的会话:

```
myAgilent3499 = New Agilent.Agilent3499.Interop.Agilent3499Class  
myAgilent3499.Initialize("GPIB0::9::INSTR", True, True, Nothing)  
myAgilent3499.Utility.Reset()
```

SCPI 与 3499 的会话:

```
my3499 = New InstrumentIO.DirectIO("GPIB0::9::INSTR", False, False)  
my3499.Timeout = 2000
```

IVI-COM 驱动程序设置 34401:

```
myAgilent34401 = New Agilent.Agilent34401.Interop.Agilent34401Class  
myAgilent34401.Initialize("GPIB0::22::INSTR", True, True, Nothing)
```

SCPI 与 34401 的会话:

```
my34401 = New InstrumentIO.DirectIO("GPIB0::22::INSTR", False, False)  
my34401.Timeout = 2000
```

VXI 驱动程序(经包装)驱动程序与 Racal 1256 的会话:

```
myRi1256 = New InstrumentDriverInterop.VxipnpWrappers.Ri1256("GPIB0::14::INSTR", True, True)
```

IVI-COM 驱动程序与 34970 的会话:

```
myAgilent34970 = New Agilent.Agilent34970.Interop.Agilent34970Class  
myAgilent34970.Initialize("GPIB0::8::INSTR", True, True, Nothing)
```

SCPI 与 34970 的会话:

```
my34970 = New InstrumentIO.DirectIO("GPIB0::8::INSTR", False, False)
```

IVI-COM 驱动程序与 34980 的会话:

```
myAgilent34980 = New Agilent.Agilent34980.Interop.Agilent34980Class  
myAgilent34980.Initialize("TCPIP0::169.254.9.80::INSTR", True, True, Nothing)  
myAgilent34980.Reset
```

SCPI 与 34980 的会话:

```
my34980 = New InstrumentIO.DirectIO("TCPIP0::169.254.9.80::INSTR")  
my34980.WriteLine("*RST")
```

IVI-C 驱动程序与 PXI/SCXI 的会话:

```
PXIDMM = New InstDvrInterop.Ivi.niDMM("DAQ::8::INSTR", True, True)  
myniSwitch = New IviCWrappers.NiSwitch("SCXI1::3", False, True)
```

Keithley 2701 例子

Dim index As Integer

Dim rdgArray(80) As Double

Dim reading As Double

Dim numPts As Integer

```
myKe2700.ConfigureAutoZeroMode(VxipnpWrappers.Ke2700.AutoZeroModeEnum.Off)
```

```
myKe2700.ConfigureMeasurement(VxipnpWrappers.Ke2700.MeasFunctionEnum.ValDcVolts, 10.0, 0.001)
```

EFT Mode(Close/Measure/Open):

For index = 1 To numChannels

```
    myKe2700.ConfigureSwitches( _  
s101, _
```

```
VxipnpWrappers.Ke2700.SwitchModeEnum.ValCloseSingleOpenOtherChannels)
```

```
    myKe2700.ConfigureApertureTimeInfo(.01, VxipnpWrappers.Ke2700.ApertureTimeUnitEnum.Nplc)
```

```
    myKe2700.Read(5000, reading)
```

```
    myKe2700.ConfigureSwitches( _  
s101, _
```

```
VxipnpWrappers.Ke2700.SwitchModeEnum.ValOpenMultiple)
```

Next

DAQ Mode(Scanned):

```
myKe2700.SetChannelList(s101120)
```

```
myKe2700.ConfigureMultiPoint( _
```

```
    1, _
```

```
    numChannels, _
```

```
    myKe2700.TriggerSourceEnum.Immediate, _
```

```
    0.0)
```

```
myKe2700.ConfigureApertureTimeInfo(.01, VxipnpWrappers.Ke2700.ApertureTimeUnitEnum.Nplc)
```

```
myKe2700.ReadMultiPoint(5000, 80, rdgArray, numPts)
```

RACAL 1256/Ext. DMM 例子

Dim index As Integer

Dim readings(20) As Double

Dim trigDelay As Double = 0.02

myHpe1411.CalZeroAuto(False)

myHpe1411.VoltDcRang(False, _
 VxipnpWrappers.Hpe1411.RangeEnum2.VoltDcRang64V)

myHpe1411.VoltDcRes(VxipnpWrappers.Hpe1411.VoltDcResEnum.VoltRes488Micro)

myHpe1411.Trigger(1, False, trigDelay, VxipnpWrappers.Hpe1411.SourceEnum1.Immediate)

myHpe1411.Sample(1, VxipnpWrappers.Hpe1411.SourceEnum.Immediate, 0.0)

EFT Mode(Close/Measure/Open):

For index = 1 To numChannels

 myRi1256.OperateSingle138(_
 x.ModuleAddressEnum1._1, _
 x.OperationEnum.Close, _
 x.RelayTypeEnum.Mux0, _
 1)

 myHpe1411.ReadQ(readings, 80)

 myRi1256.OperateSingle138(_
 x.ModuleAddressEnum1._1, _
 x.OperationEnum.Open, _
 x.RelayTypeEnum.Mux0, _
 1)

Next

DAQ Mode(Scanned):

myHpe1411.Trigger(1, False, trigDelay, VxipnpWrappers.Hpe1411.SourceEnum1.Immediate)

myHpe1411.Sample(numChannels, VxipnpWrappers.Hpe1411.SourceEnum.Timer, trigDelay)

myRi1256.ConfigOutputTrigState(VxipnpWrappers.Ri1256.OutputTrigStateEnum.Off)

myRi1256.ConfigInputTrigSource(VxipnpWrappers.Ri1256.TriggerSourceEnum1.TrigExt)

myRi1256.ArmTrigger(VxipnpWrappers.Ri1256.ArmTypeEnum.Cont)

myRi1256.DefScanList(New System.Text.StringBuilder("1(0:7,10:17,20:23)"))

myRi1256.TriggerImmediate() ' go to the first relay in the scanlist

myHpe1411.ReadQ(readings, numChannels)

Agilent 34980A 使用驱动程序的例子

Dim index As Integer

Dim rdgArray(80) As Double

EFT Mode(Close/Measure/Open):

myAgilent34980A.Scan.ScanList = 矣

myAgilent34980A.Voltage.DCVoltage.Configure(矣, 10.0,
Agilent.Agilent34980A.Interop.Agilent34980AResolutionEnum.Agilent34980AResolutionLeast)

myAgilent34980A.Voltage.DCVoltage.AutoZero(矣) =
Agilent.Agilent34980A.Interop.Agilent34980AAutoZeroEnum.Agilent34980AAutoZeroONCE

myAgilent34980A.Display.DisplayEnabled = False

myAgilent34980A.Trigger.Configure(
Agilent.Agilent34980A.Interop.Agilent34980ATriggerSourceEnum.Agilent34980ATriggerSourceImmediate, 1, 0, 1)

For index = 1 To numChannels

 myAgilent34980A.Route.Close("1001")
 myAgilent34980A.Measurement.Initiate()
 rdgArray = myAgilent34980A.Measurement.FetchNumbersOnly
 myAgilent34980A.Route.Open("1001")

Next

DAQ Mode (Scanned):

myAgilent34980A.Scan.ScanList = "1001:1020"

myAgilent34980A.Voltage.DCVoltage.Configure("1001:1020",10.0,
Agilent.Agilent34980A.Interop.Agilent34980AResolutionEnum.Agilent34980AResolutionLeast)

myAgilent34980A.Voltage.DCVoltage.AutoZero("1001:1020")
Agilent.Agilent34980A.Interop.Agilent34980AAutoZeroEnum.Agilent34980AAutoZeroONCE

myAgilent34980A.Display.DisplayEnabled = False

myAgilent34980A.Trigger.Configure(
Agilent.Agilent34980A.Interop.Agilent34980ATriggerSourceEnum.Agilent34980ATriggerSourceImmediate, 1, 0, 1)

myAgilent34980A.Route.Delay("1001:1020") = 0

myAgilent34980A.Measurement.Initiate()

rdgArray = myAgilent34980A.Measurement.FetchNumbersOnly

Agilent 34970A 使用驱动程序的例子

Dim index As Integer

Dim readings(80) As Double

Dim rdgs(80) As String

myAgilent34970.Display.DisplayEnabled = False

myAgilent34970.Voltage.AutoZero(101) =

Agilent.Agilent34970.Interop.Agilent34970AutoZeroEnum.Agilent34970AutoZeroONCE

EFT Mode(Close/Measure/Open):

myAgilent34970.Voltage.DCVoltage.Configure("101:101", 10, 0.001)

For index = 1 To numChannels

 rdgs = myAgilent34970.Scan.Read()

Next

DAQ Mode(Scanned):

myAgilent34970.Voltage.DCVoltage.Configure("101:120", 10, 0.001)

rdgs = myAgilent34970.Scan.Read()

Agilent 3499A 使用驱动程序的例子

```
Dim index As Integer
Dim readings(80) As Double
Dim numRdgs As Integer

myAgilent34401.Advanced.AutoZero = _
    Agilent.Agilent34401.Interop.Agilent34401AutoZeroEnum.Agilent34401AutoZeroOnce

myAgilent34401.Display.Enabled = False

myAgilent34401.DCVoltage.Configure(10.0, 0.001)
```

EFT Mode(Close/Measure/Open):

```
myAgilent34401.Trigger.Source =
    Agilent.Agilent34401.Interop.Agilent34401TriggerSourceEnum.Agilent34401TriggerSourceImmediate

For index = 1 To numChannels
    myAgilent3499.Route.Close("@200")
    readings(index) = myAgilent34401.Measurement.Read(5000)
    myAgilent3499.Route.Open("@200")
Next
```

DAQ Mode (Scanned):

```
myAgilent3499.Configure.ExtTriggerSource = 0
myAgilent3499.Configure.ExtTriggerOutput = True

myAgilent3499.Scan.ArmSource = _
    Agilent.Agilent3499.Interop.Agilent3499SourceEnum.Agilent3499SourceImmediate

myAgilent3499.Scan.ArmCount = 1

myAgilent3499.Scan.TriggerSource = _
    Agilent.Agilent3499.Interop.Agilent3499SourceEnum.Agilent3499SourceMIX

myAgilent3499.Configure.MuxFunction(2) = _
    Agilent.Agilent3499.Interop.Agilent3499WireEnum.Agilent3499WireWire2

myAgilent3499.Scan.ScanList = "@200: 219"

myAgilent34401.Trigger.Source = _
    Agilent.Agilent34401.Interop.Agilent34401TriggerSourceEnum.Agilent34401TriggerSourceExternal

myAgilent34401.Trigger.MultiPoint.Count = numChannels

myAgilent34401.System.WaitForOperationComplete(5000)

myAgilent34401.Measurement.Initiate()

myAgilent3499.Scan.Initiate()

myAgilent3499.System.IO.WriteString("*TRG", True)

myAgilent3499.System.WaitForOperationComplete(5000)

myAgilent34401.Measurement.FetchMultiPoint(5000, readings)
```

Agilent E1411B/E1476A 使用驱动程序的例子

```
Dim index As Integer
Dim opc As Short
Dim readings(80) As Double

myHpe1411and1476.CalZeroAuto(False)

myHpe1411and1476.VoltDcRang(False, VxipnpWrappers.Hpe1411.RangeEnum2.VoltDcRang64V)
myHpe1411and1476.VoltDcRes(VxipnpWrappers.Hpe1411.VoltDcResEnum.VoltRes488Micro)
```

EFT Mode (Close/Measure/Open):

```
For index = 1 To numChannels
    myHpe1476.ClosCardChan(1, 0)
    myHpe1411.ReadQ(readings, 80)
    myHpe1476.OpenCardChan(1, 0)
Next
```

DAQ Mode (Scanned):

```
myHpe1411and1476.ConfigureList( VxipnpWrappers.Hpe1411.FuncEnum1.ConfListVoltDc, "100:119")
myHpe1411and1476.InitImm()
myHpe1411and1476.TimedFetchQ(5000, readings, 20)
```

NI PXI-4070/SCXI-1128 例子

Dim index As Integer
Dim readings(80) As Double
Dim numRdgs As Integer
Dim Samplnt as Double

EFT Mode (Close/Measure/Open):

```
PXIDMM.ConfigureAutoZeroMode(NIDMM_VAL_AUTO_ZERO_ONCE)
PXIDMM.ConfigureMeasurement(NIDMM_VAL_DC_VOLTS, 10.0, 0.001)
PXIDMM.ConfigureTrigger(NIDMM_VAL_IMMEDIATE, 0.0)
For index = 1 To numChannels
    myniSwitch.Connect("ch0", "com0")
    PXIDMM.Initiate()
    PXIDMM.Fetch(5000, readings(index))
    myniSwitch.Disconnect("ch0", "com0")
```

Next

DAQ Mode (Scanned):

```
PXIDMM.ConfigureAutoZeroMode(NIDMM_VAL_AUTO_ZERO_ONCE)
PXIDMM.ConfigureMeasurement(NIDMM_VAL_DC_VOLTS, 10.0, 0.001)
PXIDMM.ConfigureMeasCompleteDest(NIDMM_VAL_LBR_TRIG_0)
PXIDMM.ConfigureMultiPoint(1, numChannels, NIDMM_VAL_INTERVAL, Samplnt)
myniSwitch.ConfigureScanList("sc1!md3!ch0:19->com0;" , _
    IviCWrappers.NiSwitch.ScanModeEnum.BreakBeforeMake)
myniSwitch.ConfigureScanTrigger(0.0, _
    IviCWrappers.NiSwitch.TriggerInputEnum.TtI0, _
    IviCWrappers.NiSwitch.ScanAdvancedOutputEnum.None)
myniSwitch.InitiateScan()
PXIDMM.Initiate()
PXIDMM.FetchMultiPoint(5000, numChannels, readings, numRdgs)
```

Agilent 34980A 使用 SCPI 的例子

Dim index As Integer

Dim readings As String

```
my34980.WriteLine("Rout:Scan (@)")
```

```
my34980.WriteLine("CONF:VOLT:DC 10,.001")
```

```
my34980.WriteLine("ZERO:AUTO ONCE;;DISP OFF;;TRIG:DELAY 0;;TRIG:SOUR IMM;;TRIG:COUN 1;;SAMP:COUN 1")
```

EFT Mode (Close/Measure/Open):

For index = 1 To numChannels

```
my34980.WriteLine("ROUT:CLOS (@6001)")
```

```
my34980.WriteLine("INIT;FETC?")
```

```
readings = my34980.Read()
```

```
my34980.WriteLine("ROUT:OPEN (@6001)")
```

Next

DAQ Mode (Scanned):

```
my34980.WriteLine("Rout:Scan (@6001:6020)")
```

```
my34980.WriteLine("Conf:volt:dc 10,.001,(@6001:6020)")
```

```
my34980.WriteLine("ZERO:AUTO ONCE;;DISP OFF;;TRIG:DELAY 0;;TRIG:SOUR IMM;;TRIG:COUN 1;;SAMP:COUN 1")
```

```
my34980.WriteLine("INIT;FETC?")
```

```
readings = my34980.Read()
```

Agilent 34970A 使用 SCPI 的例子

Dim index As Integer

Dim Readings As String

Dim Points As Integer

Dim replyString As String

```
my34970.WriteLine("DISP OFF;; TRIG: COUN 1; SOUR IMM")
```

EFT Mode(Close/Measure/Open):

```
my34970.WriteLine("Conf: volt: dc 10,.001,(@101:101)")
```

```
For index = 1 To numChannels
```

```
    my34970.WriteLine("INIT; FETC?")
```

```
    Readings = my34970.Read()
```

```
Next
```

DAQ Mode(Scanned):

```
my34970.WriteLine("Conf:volt:dc 10,.001,(@101:120)")
```

```
my34970.WriteLine("INIT; FETC?")
```

```
Readings = my34970.Read()
```

Agilent 3499A/34401A 使用 SCPI 的例子

Dim index As Integer

Dim Readings As String

Dim dummy As String

```
my34401.WriteLine("DISP OFF; :ZERO:AUTO ONCE;:CONF:VOLT:DC 10,.001")
```

EFT Mode(Close/Measure/Open):

```
my34401.WriteLine("TRIG:SOUR IMM;COUN 1;:SAMP:COUN 1")
```

For index = 1 To numChannels

```
    my3499.WriteLine("ROUT:CLOS (@200)")
```

```
    my34401.WriteLine("INIT;FETC?")
```

```
    Readings = my34401.Read()
```

```
    my3499.WriteLine("ROUT:OPEN (@200)")
```

Next

DAQ Mode (Scanned):

```
my3499.WriteLine("CONF:EXT:SOUR 0;OUTP 1")
```

```
my3499.WriteLine("ARM:SOUR IMM;COUN 1")
```

```
my3499.WriteLine("TRIG:SOUR MIX")
```

```
my3499.WriteLine("FUNC 2,2") '2-wire mode
```

```
my3499.WriteLine("SCAN (@200:219)")
```

```
my34401.WriteLine("TRIG:SOUR EXT;COUN" & Str(numChannels))
```

```
my34401.WriteLine("*OPC?")
```

```
dummy = my34401.Read
```

```
my34401.WriteLine("INIT")
```

```
my3499.WriteLine("INIT")
```

```
my3499.WriteLine("*TRG")
```

```
my3499.WriteLine("*OPC?")
```

```
dummy = my3499.Read
```

```
my34401.WriteLine("FETC?")
```

```
Readings = my34401.Read
```

Agilent VXI E1411B/E1476A 使用 D-SCPI 的例子

Dim index As Integer

Dim Readings As String

```
myDSCPI.WriteLine("ZERO:AUTO ONCE;; TRIG:SOUR IMM;DELAY 0")
```

EFT Mode (Close/Measure/Open):

```
myDSCPI.WriteLine("CONF:VOLT:DC 10,.001 (@100)")
```

```
myDSCPI.WriteLine("TRIG:COUN 1")
```

```
myDSCPI.WriteLine("SAMP:COUN 1")
```

For index = 1 To numChannels

```
    myDSCPI.WriteLine("INIT; FETC?")
```

```
    Readings = myDSCPI.Read()
```

Next

DAQ Mode (Scanned):

```
myDSCPI.WriteLine("CONF:VOLT:DC 10,.001 (@100:119)")
```

```
myDSCPI.WriteLine("INIT;FETC?")
```

```
Readings = myDSCPI.Read()
```

www.agilent.com

安捷伦测试和测量技术支持、服务和协助

Agilent 公司的宗旨是使您获得最大效益，而同时将您的风险和问题减少到最低限度。我们将努力确保您获得的测试和测量能力物有所值，并得到所需要的支持。我们广泛的支持和服务能帮助您选择正确的 Agilent 产品，并在应用中获得成功。我们所销售的每一类仪器和系统都提供全球保修服务。对于停产的产品，在 5 年内均可享受技术服务。“我们的承诺”和“用户至上”这两个理念高度概括了 Agilent 公司的整个技术支持策略。

我们的承诺

我们的承诺意味着 Agilent 测试和测量设备将符合其广告宣传的性能和功能。在您选择新设备时，我们将向您提供产品信息，包括切合实际的性能指标和经验丰富的测试工程师的实用建议。在您使用 Agilent 设备时，我们可以验证设备的正常工作，帮助产品投入生产，以及按要求对一些特别的功能免费提供基本的测量协助。此外，还提供一些自助软件。

用户至上

用户至上意味着 Agilent 公司将提供大量附加的专门测试和测量服务。您可以根据自己的独特技术和商务需要来获得这些服务。通过与我们联系取得有关校准、有偿升级、超过保修期的维修、现场讲解和培训、设计和系统组建、工程计划管理和其它专业服务，使用户能有效地解决问题并取得竞争优势。经验丰富的 Agilent 工程技术人员能帮助您最大限度地提高生产率，使您在 Agilent 仪器和系统上的投资有最佳回报，并在产品寿命期内得到可靠的测量精度。

Agilent Open

Agilent Open 简化连接和编程测试系统的过程，以帮助工程师设计、验证和制造电子产品。Agilent 的众多系统就绪仪器，开放工业软件，PC 标准 I/O 和全球支持，将加速测试系统的开发。要了解更详细的情况，请访问：www.agilent.com/find/openconnect。

欢迎订阅免费的



安捷伦电子期刊

www.agilent.com/find/emailupdates

得到您所选择的产品和应用的最新信息。

Agilent Direct

www.agilent.com/find/agilentdirect

高置信地快速选择和使用您的测试设备解决方案

有关安捷伦开放实验室暨测量方案中心和安捷伦测试与测量技术认证，请访问：www.agilent.com.cn/find/openlad。

请通过 Internet、电话、传真得到测试和测量帮助。

在线帮助：www.agilent.com/find/assist

热线电话：800-810-0189

热线传真：800-820-2816

安捷伦科技有限公司总部

地址：北京市朝阳区建国路乙 118 号
招商局中心 4 号楼京汇大厦 16 层
电话：800-810-0189
(010) 65647888
传真：(010) 65647666
邮编：100022

上海分公司

地址：上海市西藏中路 268 号
来福士广场办公楼 7 层
电话：(021) 23017688
传真：(021) 63403229
邮编：200001

广州分公司

地址：广州市天河北路 233 号
中信广场 66 层 07-08 室
电话：(020) 86685500
传真：(020) 86695074
邮编：510613

成都分公司

地址：成都市下南大街 2 号
天府绿洲大厦 0908-0912 室
电话：(028) 86165500
传真：(028) 86165501
邮编：610012

深圳分公司

地址：深圳市高新区南区
黎明网络大厦 3 楼东区
电话：(0755) 82465500
传真：(0755) 82460880
邮编：518057

西安办事处

地址：西安市高新区科技路 33 号
高新国际商务中心
数码大厦 23 层 01-02 号
电话：(029) 88337030
传真：(029) 88337039
邮编：710075

安捷伦科技香港有限公司

地址：香港太古城英皇道 1111 号
太古城中心 1 座 24 楼
电话：(852) 31977777
传真：(852) 25069256

香港热线：800-938-693

香港传真：(852) 25069233

Email: tm_asia@agilent.com

本文中的产品指标和说明可不经通知而更改
©Agilent Technologies, Inc. 2006
出版号：5989-1929CHCN
2006 年 7 月 印于北京



Agilent Technologies