

A Comparison Between PXI and the Agilent 34980A for Switch/Measure Applications

Application Note



Contents

Introduction	1
Switch/Measure system characteristics	1
Cost of ownership	5
Execution speed	11
Ease of software development	16
Benchmark results	17
Conclusion	18
Appendix A	19

Introduction

This application note compares the Agilent 34980A switch/measure unit to a comparably equipped National Instruments PXI solution with respect to features, execution speed, cost of ownership, and ease of software development for functional test and data acquisition environments.

Switch/Measure System Characteristics

Test systems can be constructed with many complex stimulus and measurement instruments, but two components are almost always present – a digital multimeter (DMM) and a bank of relays (switches). A fundamental

core then, is a switch/measure function, which can be implemented in three different ways:

1. Discrete instruments with cable interconnects, such as an Agilent 34410/11A standalone high-speed DMM and a separate switching system such as the Agilent 3499A/B/C switch/control mainframe.
2. Cardcages, such as PXI or VXI, into which a DMM card and switching cards have been placed. A separate controller, such as an embedded PC or an interface card connecting the cardcage to a PC, is required.



3. Combined systems, such as the 34970A data acquisition switch unit, with built-in DMM and up to three plug-in cards, or a 34980A switch/measure unit, with a built-in controller, DMM, quiet analog backplane and the ability to insert up to eight plug-in cards.

This application note focuses on the last two concepts, specifically:

- An Agilent 34980A mainframe, with a built-in controller, front panel, built-in 6¹/₂-digit DMM and various plug-in modules
- A PXI cardcage with PCIe-MXI interface, PXI-4070 6¹/₂-digit DMM, and various plug-in modules

Feature comparison

Table 1 compares features between a PXI solution and the 34980A. In this chart, if one instrument has an advantage over the other, it is boldfaced. You can see at a glance that the 34980A features outweigh those of the PXI solution. Let's examine the differences.

Feature	PXI	34980A
Open, multi-vendor architecture	Yes	No (see open architecture)
Physical size	4U	3U
Ventilation provisions	2U additional ¹	No special requirements
Rack mounting	Non-EIA standard width ²	EIA standard width
High-speed I/O	Proprietary (MXI, PXI Express)	Industry standard LAN, USB, GPIB
Price	High (see table 2)	Low (see table 2)
Analog bus	No	8-wire
Trigger bus	Yes	Yes
Front panel	Software	Hardware and software
Web GUI	No	Yes
EMI shielding	Not required by spec ³	Designed-in and tested
Hot swap cards	No	Yes
PC reboot	Req'd when cards changed	No
Software installation	2 hours (est) ⁴	None required
Programming support		
LabVIEW	Excellent	Excellent
Visual Studio®.NET	Fair ⁵	Excellent
IVI-C	Yes	Yes
IVI-COM	No	Yes
SCPI	No	Yes

Notes:

1. The PXI-1033 requires space above and below the unit to handle airflow. This means that when it is rack mounted, you must allow 4 rack units, and you must make sure that no other instruments are directly above or below the unit so as not to block airflow. This means that a 3U PXI card requires 6U of rack space.
2. The Electronic Industries Alliance (EIA) has created standards for height, width and depth of instruments in 19" racks. Some PXI cages are not a standard width. The PXI-1033 is wider than 1/2 rack, which leaves wasted space to its sides.
3. Agilent has documented magnetic interference from the SCXI power supply to slot 8 of a combination PXI-SCXI cardcage causing loss of one whole digit of resolution in a PXI-4070 DMM. Also, a service bulletin from NI states that a high density reed matrix card should not be placed next to an embedded PC with a hard disk drive because the disk's magnetic field can change the state of the relays.
4. Typical time required to (from installation manual, in this order): turn off PC, install PCIe card and any PXI cards, connect cable to cardcage, boot PC, ignore errors finding drivers since none have been installed, find correct CD to install NI MAX (driver CD labeled differently from manual), download drivers from NI web site (600 MB for NI DMM and 400 MB for NI SWITCH) and install. (If using VS.NET, additional time is spent locating nidmm.vb and adding the driver (nidmm_32.dll) to the VS.NET program.)
5. According to NI Knowledge Base article 3939UN5B, NI does not have native .NET support for many PXI devices, including the PXI-4070 and switches. Users must locate source files and manually attach them to their programs. In addition, a memory conflict and driver loading issues have been documented (Knowledge Base article 2Q36UJ2H)

Table 1. Features of the 34980A vs. a PXI switch/measure system

Open architecture

The PXI architecture is an open, multi-vendor backplane architecture. Its physical and electrical properties are documented by an industry alliance (in this case, the PXI Systems Alliance, www.pxisa.org, which includes more than 60 companies that provide a mix of PXI instrumentation, test fixtures and systems integrators).

The 34980A also uses an open industry standard called LAN eXtensions for Instrumentation (LXI). The LXI Consortium (www.lxistandard.org) is an association of more than 45 of the top names in the electronics industry. Rather than place the “open-ness” in the backplane definition as PXI does, the 34980A and other new LXI-compliant products put the emphasis on industry-standard I/O connectivity and ease of programming in multiple development environments. This allows R&D (and you) to avoid paying for the extra capabilities demanded by open backplane standards (power, cooling, bus lines and reference clocks, for example) that may never be needed. The 34980A has just

enough power, just enough cooling, and just enough backplane speed in the box to optimize it for the switch cards for which it was designed. You can add custom electronics by using a breadboard card that has a supplied backplane interface and room for you to add your own circuitry. This keeps the costs low and makes it possible for the instrument to be tested more thoroughly so that you are not left solving problems with card-to-card interference, insufficient power, cooling and more.

Size

PXI cards are called “3U,” which would lead you to believe that they occupy an EIA standard 3 rack units (1U = 1.75 inches). It turns out that although the cards are 3U, the cardcage is 4U. In addition, ventilation requirements on some PXI cages, including the NI PXI-1033, require clearance above and below the unit to provide adequate ventilation. PXI cardcages are also sometimes not EIA standard widths –

the PXI-1033 is larger than ½-rack width – and thus there is also wasted space to the sides.

The 34980A is a standard full-rack width, and only 3U high. Ventilation air is drawn in from the sides and exhausts out the back, so no additional rack space is required.

This can be important for bench applications too, because nothing can be stacked on top of a PXI-1033. The 34980A can have other instruments stacked on top or on the bottom.

Availability of cards

In addition to relays and DMMs, there are several other classes of functionality typically implemented in switch/measure systems used for electronic functional test (EFT) because they are often used together. All of the following cards are available in PXI format and for the 34980A, but the 34980A provides industry-standard

connectors (50- and 78-pin D-sub), whereas PXI cards typically use much smaller connectors due to the space limitations on the modules.

- Digital I/O (DIO)
- Digital-to-analog converters (DACs), which are sometimes fast enough to use as waveform generators at low frequencies
- Frequency counters/totalizers
- Isothermal terminal blocks
- Customizable breadboard cards (useful for signal conditioning), such as those typically required by strain gages

Switch cards come in a variety of types, and are configured as multiplexers, matrices or general purpose switches.

- Low bandwidth (DC-100 MHz)
 - FETs (high switching speed, high on-resistance, lower voltage and current specs)
 - Reed relays (medium switching speed, low on-resistance, medium voltage and current specs; typically have high thermal EMF)

- Armature relays (lower switching speed, low on-resistance, higher voltage and current specs; typically have low thermal EMF)
- High bandwidth (up to 26 GHz or higher with the microwave switch driver)
 - RF relays (low speed, low on-resistance, low voltage and current)

- Microwave relays (low speed, low on-resistance, low voltage and current)

Other card types available in PXI and LXI formats:

- Digitizers
- RF up- or down-converters
- Power supplies

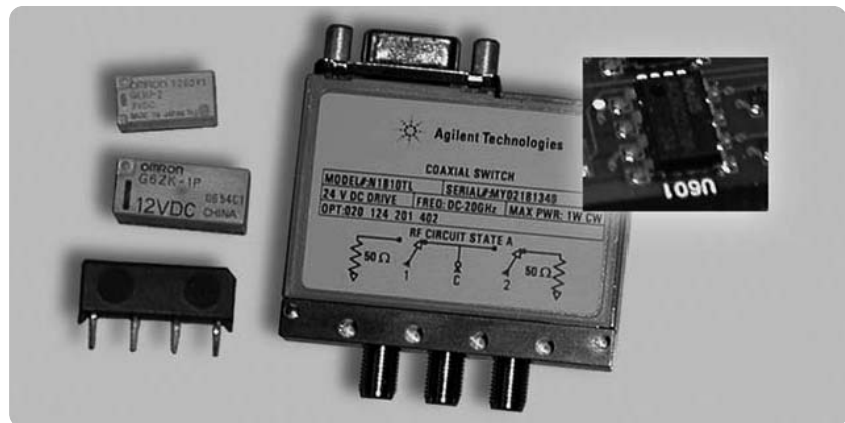


Figure 1. A few types of switches – reed, armature, RF and FET

Cost of ownership

Initial cost

The purchase price for the 34980A with 8 empty slots, built-in LAN, GPIB and USB interfaces and a 6½-digit DMM is US \$2315. The lowest-cost NI PXI frame (PXI-1033) with 4 empty slots, included PCIe-MXI interface card and NI PXI-4070 6½-digit DMM is US \$2998 (prices effective October 2006 from Agilent and NI Web sites). These prices are based on a typical data logging (DAQ) application with 300 V inputs multiplexed into a 6½-digit DMM for measurements. Other EFT configurations are shown in Tables 4-6.

100 ch, 300 V 1 A configuration:		NI	
Agilent	Price (US\$)		Price (US\$)
34980A switch/measure unit with 6½-digit DMM	\$2,315	PXI-1033 chassis	\$999
34922A 70-ch multiplexer + 34922T terminal block	\$1,450	4070 6½-digit DMM	\$1,999
34921A 40-ch multiplexer + 34921T terminal block	\$1,014	PXI-2527 32-ch multiplexer + terminal block (x4)	\$6,192
Total price	\$4,779	Total price	\$9,190
Notes	Uses 5 of 8 slots	Notes	Uses 5 of 5 slots

300 ch, 300 V 1 A configuration:		NI	
Agilent	Price (US\$)		Price (US\$)
34980A switch/measure unit with 6½-digit DMM	\$2,315	PXI-1044 chassis	\$3,495
34922A multiplexer + 34922T terminal block as above (x5)	\$7,250	4070 6½-digit DMM	\$1,999
Total price	\$9,565	MXI express kit PXIe-PCIe8361	\$995
Notes	Uses 5 of 8 slots	PXI-2527 32-chan multiplexer + terminal block (x10)	\$15,480
		Total price	\$21,969
		Notes	Uses 12 of 14 slots

500 chan, 300 V 1 A configuration:		NI	
Agilent	Price (US\$)		Price (US\$)
34980A switch/measure unit with 6½-digit DMM	\$2,315	PXI-1045 chassis	\$4,495
34922A multiplexer + 34922T terminal block (x8)	\$11,600	4070 6½-digit DMM	\$1,999
Total price	\$13,915	MXI express kit PXIe-PCIe8361	\$995
Notes	Uses 8 of 8 slots	PXI-2527 32-ch multiplexer + terminal block (x16)	\$24,768
		Total price	\$32,257
		Notes	Uses 18 of 18 slots

Table 2. Data acquisition application price comparisons

The selection of 34980A modules including prices is shown in Table 3. Up to eight of these, in any order, can be placed in the eight slots in the 34980A main-frame. The built-in DMM does not occupy a slot. This is a significant difference between PXI and the 34980A, not just because of the price difference and slot utilization, but because the 34980A's controller, analog bus and relay/DMM handshaking are all built-in; no additional wiring is involved. With PXI, the DMM occupies one slot, there is no analog bus, and there are no automatic handshake connections between the relay cards and the DMM; you must figure out how to interconnect the appropriate backplane signals in your software.

**Selection guide for Agilent 34980A
Multifunction Switch/Measure Unit and modules**

Module	Model#	Description	Price (US\$)
	34980A	Multifunction switch/measure unit	\$2,315
Multiplexer modules			
	34921A	40-channel armature multiplexer w/low thermal offset	\$790
	34922A	70-channel armature multiplexer	\$1,276
	34923A	40/80-channel reed multiplexer	\$988
	34924A	70-channel reed multiplexer	\$1,472
	34925A	40/80-channel optically isolated FET multiplexer	\$1,177
Matrix modules			
	34931A	Dual 4x8 armature matrix	\$889
	34932A	Dual 4x16 armature matrix	\$1,472
	34933A	Dual/quad 4x8 reed matrix	\$988
General-purpose modules			
	34937A	28-channel Form C and 4-channel Form A	\$889
	34938A	20-channel 5-amp Form A	\$1,078
RF and microwave modules			
	34941A	Quad 1x4 50-ohm 3-GHz RF multiplexer	\$1,571
	34942A	Quad 1x4 75-ohm 1.5-GHz RF multiplexer	\$1,571
	34945A/ 34945EXT	Microwave switch/attenuator driver	\$1,190
	34946A	Dual 1x2 SPDT terminated microwave switch	\$1,870
	34947A	Triple 1x2 SPDT unterminated microwave switch	\$1,520
System Control modules			
	34950A	64-bit digital I/O with memory and counter	\$1,177
	34951A	4-channel isolated D/A converter with waveform memory	\$988
	34952A	Multifunction module with 32-bit DIO, 2-ch D/A and totalizer	\$690
	34959A	Breadboard module	\$492

Table 3. 34980A prices (effective October 2006)

Tables 4-6 shows some additional price comparisons for similar configurations of PXI and the 34980A. Table 4 shows a simple system that uses a mixture of low and high frequency switches and some digital I/O.

Mixed-signal system example

32 low frequency GP switch channels							
32 Digital I/O channels							
8 1x4 RF muxes to 2.5 GHz							
Agilent 34980A solution:		Price (US\$)		NI PXI solution:		Price (US\$)	
34980A 8-slot mainframe		\$1,803		PXI-1044 14-slot chassis		\$3,495	
34937A 32-channel form C GP switch		\$1,778		PXI-PCI 8331 MX14 high speed serial interface kit with 3 m copper cable		\$1,495	
34937T terminal block for 34937A		\$448		PXI-2570 40-channel SPDT switch		\$1,299	
34950A 64-bit digital I/O module		\$1,177		LFH200 cable connector for PXI-2570		\$399	
34950T terminal block for 34950A		\$224		PXI-2594 one 1x4 50-ohm 2.5-GHz switch	x8	\$5,592	
34941A quad 1x4 50-ohm 3-GHz switch	x2	\$3,142		PXI-6533 32-bit digital I/O		\$1,199	
Total		\$8,572		SCB-68 terminal block for PXI-6533		\$299	
				SH68-68-D1 cable for PXI-6533		\$159	
				Total		\$13,937	

Table 4. Costs for a test setup configured for a mixed-signal example

Table 5 shows a comparison for a system that uses microwave relays, a DMM and digital I/O.

High frequency example

12 1x2 20 GHz microwave switch channels							
40 multiplexer channels							
6 1/2-digit DMM							
30 GP 1 A switch channels							
32-bit digital I/O							
Agilent 34980A solution:		Price (US\$)		NI PXI solution:		Price (US\$)	
34980A 8-slot mainframe with DMM option		\$2,315		PXI-1044 14-slot chassis		\$3,495	
34923A 40/80-channel reed relay multiplexer		\$988		PXI-PCI 8331 MXI4 high-speed serial interface kit with 3-m copper cable		\$1,495	
34923T terminal block for 34923A		\$224		PXI-2599 26 GHz dual SPDT x6		\$8,994	
34947A 20-GHz triple 1x2 unterminated switch		x4 \$6,080		PXI-4070 DMM		\$1,999	
34937A 32-channel GP switch		\$889		PXI-2530 64-channel reed relay multiplexer		\$1,799	
34937T terminal block for 34937A		\$224		TB-2630 terminal block for PXI-2530		\$249	
34952A multifunction 32-bit digital I/O totalizer plus 2 DAC channels		\$691		PXI-2570 40-ch form C GP relay (1 A)		\$1,299	
34952T terminal block for 34952A		\$174		TBX-50 DSUB screw terminal block cable connector for PXI-2570		\$249	
Total		\$11,585		PXI-6533 32-bit digital I/O		\$1,199	
				SCB-68 terminal block for PXI-6533		\$299	
				SH68-68-D1 2-m cable for PXI-6533		\$159	
				Total		\$21,236	

Table 5. Costs for a test setup configured for a high frequency example

Table 6 shows a comparison for a system that would be used in a data acquisition application.

Data acquisition example

- 30 channels of thermocouple inputs
- 10 channels of RTD inputs
- 10 voltage inputs
- 10 four-wire resistance inputs
- 4 frequency inputs (<4 MHz)
- 20 digital state inputs

Agilent 34980A system solution:		Price (US\$)	National Instruments PXI solution:		Price (US\$)
34980A mainframe with DMM option		\$2,315	PXI-1033 PCI Express chassis with 5 slots		\$999
34921A 40-channel armature multiplexer		\$790	PXI-4070 DMM		\$1,999
34921T terminal block for 34921A		\$224	PXI-2527 32-channel armature multiplexer		\$1,299
34922A 70-channel armature multiplexer		\$1,276	TB-2627 terminal block with CJC for thermocouple		\$249
34922T terminal block for 34922A		\$174	PXI-2530 64-channel multiplexer		\$1,799
34950A 64-bit digital I/O with memory and two 5 MHz counter channels		x2 \$2,354	TB-2630 terminal block for PXI-2530		\$249
34950T terminal block for 34950A		x2 \$448	PXI-6533 32-bit digital I/O		\$1,199
Total		\$7,581	SCB-68 terminal block for PXI-6533		\$299
			PXI-6602 counter/timer		\$899
			TB-2715 terminal block for PXI-6602		\$199
			Total		\$9,190

Table 6. Costs for a test setup configured for a data acquisition example

In each configuration you can see that PXI is more expensive and there is no space saving advantage when compared to the 34980A. In addition to initial purchase price, there are other factors related to cost: maintenance, support and development time.

Maintenance costs

If your system is located outside of the U.S., consider the cost and time of return-to-factory repair that NI requires, versus using Agilent's more than 50 worldwide repair centers. To minimize downtime, you may choose to use loaner or rental equipment. Agilent can arrange loaners from its hubs, and has distributors throughout the world. Rental companies stock many Agilent products due to the high demand, but you may have trouble locating PXI rental companies. The cost of spares should also be taken into account. For example an extra PXI-4070 DMM costs an additional US \$1999.

Support costs

An Agilent support engineer can remotely access your LXI instrument from anywhere in the world without installing any additional software by using the 34980A's built-in Web interface (as long as you enable the engineer to get through your firewall). This capability is not possible with PXI hardware without using custom software. (NI's free remote device access software can be used in some development environments and with some operating systems; Microsoft® Visual Studio and Mac O/S are notable exceptions.)

Development time costs

Time is money, too. PXI devices have no front panel or native command language (for example, SCPI), so they must be programmed using drivers, which take time to download and learn. The 34980A's internal SCPI command set was designed by the same engineering group that designed the hardware, so it was fully tested before the first unit left the factory and does not require the use of a driver. Drivers are available too, in multiple development environments, but they are not necessarily optimized for speed. Later we will discuss this more.

PXI cards cannot be removed or inserted with the power on, and since the PXI bus is an extension of the PC's backplane, the PC must be rebooted whenever cards are changed. This is not necessary with the 34980A, which is not only independent from the PC's backplane *du jour*, its cards can be removed and inserted while power is on. As a result, PXI users may find themselves rebooting the PC more often than 34980A users.

NI has only limited support for its PXI products in the Microsoft Visual Studio.NET environment. For example, NI does not have native .NET drivers for its products; Agilent does. NI does provide VB.NET and VC.NET source files that can be manually added to your program. Agilent and NI also offer *Driver Wrapper Wizards* that can convert NI's IVI-C drivers to .NET-compatible versions. In contrast, Agilent provides LabVIEW, IVI-C and IVI COM drivers for the 34980A, so it is ready to use in whatever development environment you use. When you use Agilent's T&M Toolkit and I/O Libraries Suite, you can be programming your 34980A within minutes in the .NET environment.

Because the 34980A has a front panel, you can be using your 34980A right after removing it from the box without ever

touching a computer. In addition, the built-in graphical Web interface makes it possible to use the 34980A from your favorite Web browser without having to first download any software. Simply click on the graphical switch representation to open, and close switches and take readings. Built-in commands and internal handshaking between the DMM and the switch cards makes it much easier to program than a comparable PXI DMM and switch. With PXI, you must handle the hand-shaking yourself, and every setup step requires a driver call. See Appendix A for a sample program. All the little steps that are required using PXI can add up to a lot of an extra development time.

Execution speed

Differences between EFT and DAQ

The same hardware can be used in both EFT and data acquisition (DAQ) applications, but the throughput you can achieve varies greatly between them. We will show some examples of this later, but now let's look at some of the differences between these two types of test environments. Understanding the differences can be important when you compare performance specs between PXI solutions and the 34980A because you may not achieve claimed speeds unless you understand the use model.

EFT is a measurement scheme in which stimuli are applied to a device under test (DUT), and the outputs are monitored and compared with expected responses. A system constructed for EFT tends to connect many instruments through fast switches in an attempt to verify functionality of an electronic module, such as an automotive engine controller. In an EFT system, switches must be closed, stimuli (including DUT power) and loads applied, a measurement taken and relays opened. This process is repeated many times to cover all pins on the DUT. Thus the overhead of individual readings and re-configuration of the stimulus and measurement instruments must be incurred repeatedly (close a relay, take a reading, open the relay). Polling is often done to make sure a reading is ready, which can add extra I/O execution time. Results are checked against limits and failures logged. The result is that execution speeds are on the order of 50-250 readings per second. The 34980A was designed to fulfill these testing requirements.

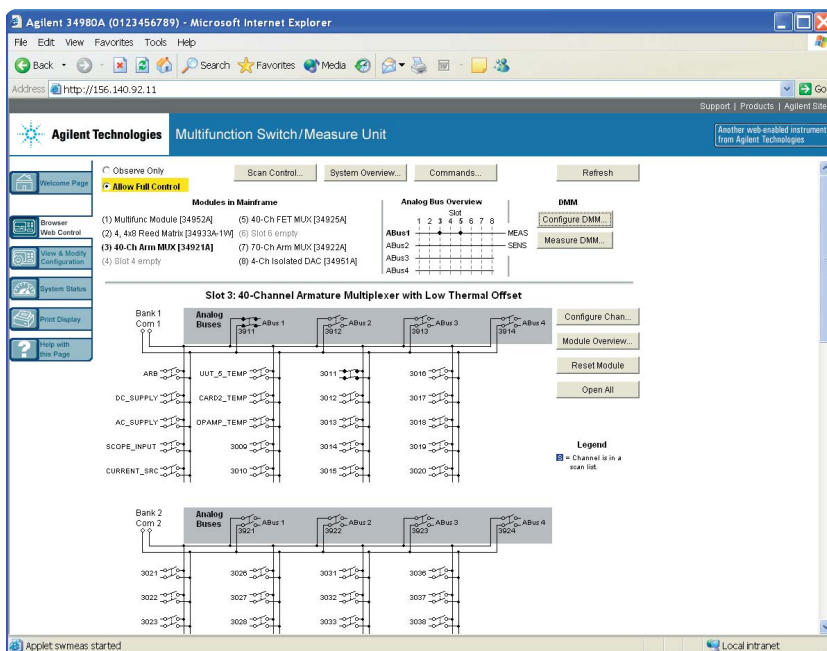


Figure 2. 34980A Graphical Web Interface

DAQ is a measurement scheme in which many sensors are multiplexed into a measurement device such as a DMM. Large amounts of data may be logged for analysis later or in real time. An example is measurement of temperatures at various points in the fuselage of the Space Shuttle. In DAQ mode, a voltmeter and a bank of relays are programmed together to do a scanned measurement – a list of switch open/close states is downloaded into the measurement/switching system, and a hardware handshake links the DMM measurement to the switch setup. A *measurement complete* trigger from the voltmeter advances relays through this list or they can simply free run. The DMM can either wait a preprogrammed delay after it sends out a measurement complete trigger or it can wait until it receives a hardware signal from the switches called an *advance trigger*, indicating that the switches have gone to their programmed state and it is OK to take a reading. Because of this hardware handshaking, it is possible to minimize the amount of I/O data being transferred to and from each instrument. In addition, the setup time is not duplicated for each measurement. Instead, setup information is downloaded and then a single initiation command starts the whole

operation. The execution speed can thus be very high depending upon the desired measurement resolution and the switching speed. FET switches can achieve reading speeds on the order of 1,000 channels/s. The 34970A data acquisition/switch unit is a proven workhorse in this arena, and the 34980A can also be used in this mode.

Fast backplane speeds such as the 2 Gbit/s bidirectional PCI Express that is used in the PXI-1033 cardcage do not help improve throughput when the bulk of the time is spent waiting for relays to close. For example,

the execution speed to close an armature relay on a PXI-2527 32-channel relay card, take one DCV reading as fast as possible on a PXI-4070, open that relay and read the DMM is about 13 ms, or 77 channels/s. When 32 such relays are scanned from a scanlist, the execution speed is slightly higher. The advertised scanning speed for that switch card is 140 channels/s. (We measured 92 channels/s without a DMM measurement using NI provided drivers in Visual Basic.NET; the Agilent 34980A can do the same job at a higher speed. See benchmark data later in this application note).

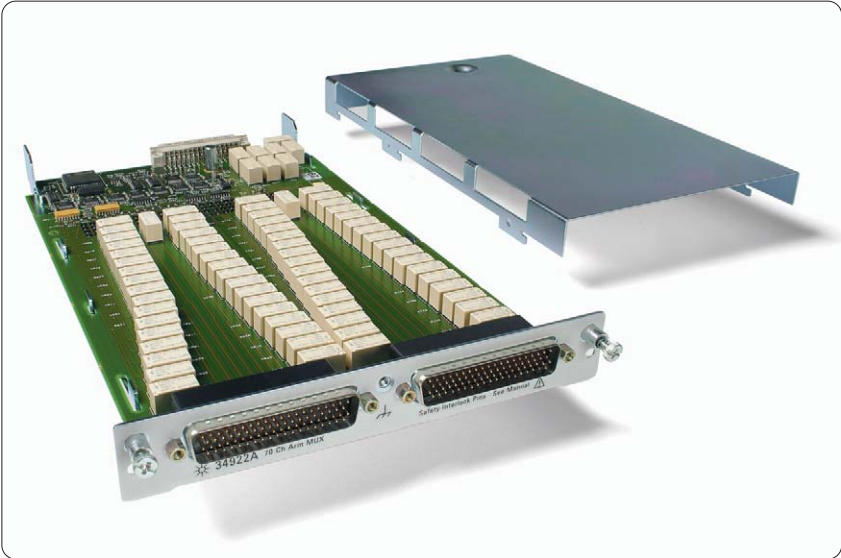


Figure 3. 34922A 70 channel armature multiplexer

Drivers vs. SCPI

Agilent's LAN, USB and GPIB-based instruments have a built-in command language called SCPI - Standard Commands for Programmable Instruments – an ASCII language that you can access via simple VISA function calls without installing drivers. Many people use SCPI because it is fast, easy-to-learn and well known. Some engineers prefer to use higher-level software, called interchangeable virtual instrument (IVI) drivers. There are two forms of IVI: IVI-C, which works with several C compilers and VB6, and IVI COM, which works with Visual Studio.NET and other COM-based development environments. NI provides only IVI-C drivers. Agilent provides both for the 34980A.

Drivers that communicate with PXI hardware use low-level register programming that would be difficult for end-users to write, so manufacturers of the hardware provide them for you. Because register reads and writes are very fast, speed optimization is not usually required, but if the driver has any bugs in it, the instrument may be unusable until it is patched, and you have no recourse but to wait for a new version. The only way to control a PXI instrument is through the driver. Therefore the driver defines 100% of the instruments functionality. In contrast the 34980A can be controlled via drivers, SCPI, Web interface or the front panel.

Drivers that communicate with SCPI instruments generate SCPI commands, and driver bugs can be overcome easily by allowing the programmer to use the native SCPI commands. In addition, SCPI instruments typically have more intelligence

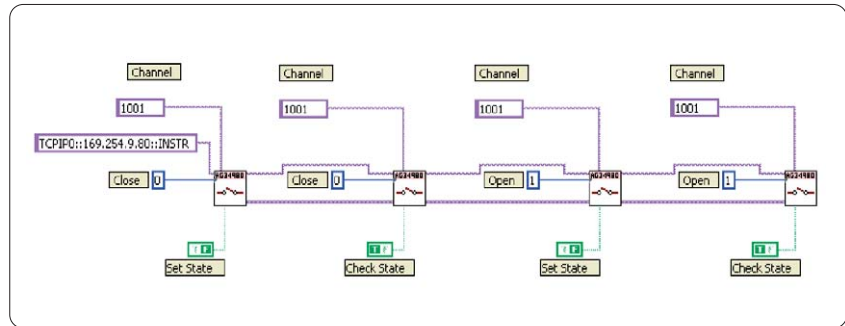
built-in than their PXI counterparts, so many complex functions can be engaged using short commands that minimize I/O traffic.

With modern computers, the time required to execute driver calls and for the driver to generate SCPI is negligible, so IVI drivers can be as fast as direct programming in SCPI provided the driver uses the SCPI efficiently. The 34980A drivers, like many drivers, are not optimized for speed but for functionality. If ultimate throughput is your goal, use the 34980A's native SCPI commands. For example, using the 34980A's LabVIEW driver, it is possible to send a "close relay; test for closed; open the relay; test for open" command sequence that can take about twice as long to execute as it would using SCPI commands in an efficient manner. When programmed efficiently, the 34980A is as fast as its PXI equivalent. Note too that a simple close/open benchmark is not indicative of real throughput, which is dependent on the types of measurements that are

made. This concept is shown at right, or for a more detailed look at this, see Agilent Application Note 5989-4886EN “Optimizing Test Systems for Highest Throughput, Lowest Cost and Easy LXI Instrument Integration.”

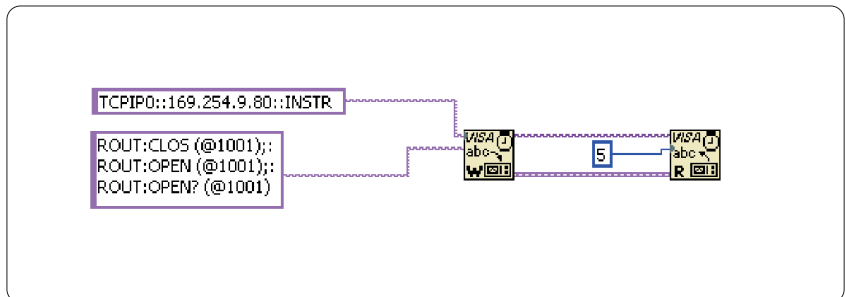
NI has touted the fast I/O speed and low latency (time to issue the first command) of its I/O interfaces. But I/O transfer time is not the only consideration in total test execution. For example, a typical test program will close a number of armature relays that can take up to 20 ms to open or close. It may also activate reed relays in a switching matrix, which open and close in about 500 micro-seconds. It could take a thousand DMM readings with an aperture time of 1 ms and perhaps 100 readings at the higher resolution aperture time of 16.7 ms. It could download a 1000-point waveform into an arbitrary waveform generator.

It could program a power supply for several voltage settings and wait hundreds of milliseconds for settling time. In all of these cases, the command parsing time of the instrumentation plus execution time inside the instrument is often much greater than the 1 ms of



This LabVIEW program emits the following SCPI code:

ROUT:CLOS (@1001)	1.4 ms	(Set state)
ROUT:CLOS? (@1001)	1.3 ms	(Check state)
Read 1	1.5 ms	
ROUT:OPEN (@1001)	1.6 ms	(Set state)
ROUT:OPEN? (@1001)	1.0 ms	(Check state)
READ 1	1.5 ms	
Total	8.3 ms	(6 LAN packets)



This LabVIEW program emits the following code:

ROUT:CLOS (@1001);:ROUT:OPEN (@1001);:ROUT:OPEN? (@1001)	1.4 ms
Read "1"	2.5 ms
Total:	3.9 ms (2 LAN packets)

LAN latency that it takes to initiate the command. In addition, today’s programmable instruments, including the 34980A, have the ability to do complete self-triggered measurements, further streamlining the I/O to and from those instruments. **The bottom line: in a large percentage of cases, I/O transfer time simply makes no difference in the total test time for these types of applications.**

We mentioned earlier that many drivers are not optimized for speed. Since the 34980A uses SCPI, you can optimize the commands yourself by writing a “mini-driver” in which you pass all of your SCPI commands through a screening routine that checks the command against the current state of the instrument and does not send that command if it would not result in a change of state. This is called *state caching*. A typical driver may do a lot of query commands to determine the state, which adds unnecessary execution time.

Here is an example of redundant commands that can increase execution time: suppose you program in SCPI and send the command `CONF:VOLT DC 10,001`. This command will tell a DMM to go to the DC function and set the range to 10 V and the resolution to 1 mV. At top GPIB speeds of 1 MB/sec, this 21-character string (including the carriage return character) would take 21 microseconds plus the overhead involved in the function calls, which would add about 10 us in a modern PC. The instrument has to parse this command using its considerably slower processor. In the case of a 34401A DMM, for example, the 12 MHz processor would take about 21 milliseconds of additional time.

Now, if the DMM is already in that state, this is 21.031 ms of time that did not need to be spent. It may not sound like much, but a typical test program can have hundreds or thousands of DMM readings, so it is important to maximize the efficiency of the I/O and subsequent processing in the instrument. A state-caching driver would know the instrument is in this state, and it would not bother to send the command.

The Agilent 34980A is a new class of instrument that uses a much faster internal micro-processor. It can execute that same string in about 1.5 ms, reducing the need for state caching.

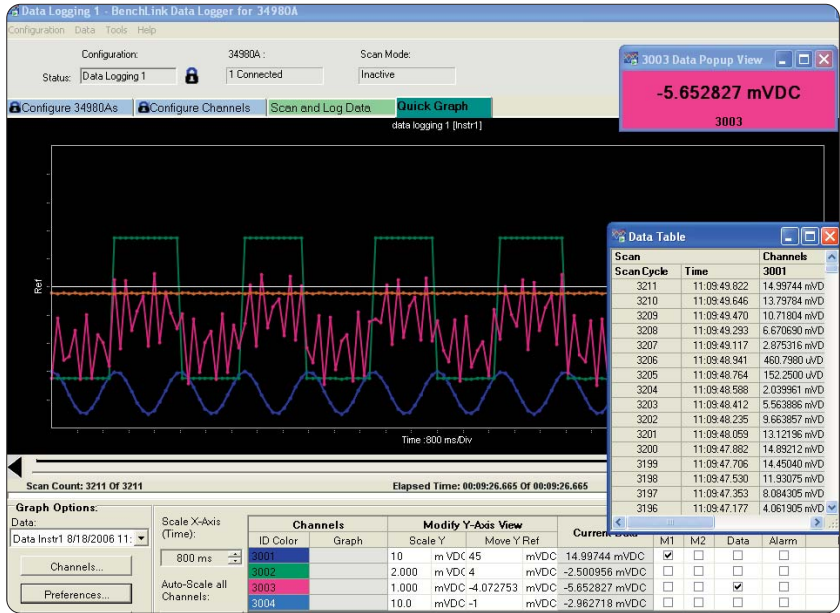


Figure 4. 34826A BenchLink Data Logger for 34980A

Ease of software development

Although NI LabVIEW and Agilent VEE Pro are popular graphical environments, much manufacturing test work is done in textual environments. In this realm, VB6 and C or its derivatives (VC++, LabWindows/CVI) dominate. For data acquisition and logging applications, Agilent also offers the inexpensive BenchLink datalogger software, (see figure 4) and for some applications, you can also work from Microsoft Word and Excel using Agilent's free IntuiLink toolbar. However, there are several reasons to develop new functional test applications using Microsoft's Visual Studio.NET environment. Both Agilent and NI have released tools to help a test system developer create programs in this environment (Agilent's T&M Toolkit and NI's Measurement Studio). You can also port VB6 programs into the VB.NET environment using a built-in porting tool.

Steps Required to Install and Use 34980A in Visual Studio 2005. These steps take approximately 15 minutes, or much less if T&M Toolkit is already installed.

1. Insert 34922A switch card in mainframe, connect terminal module and LAN cable to PC.

Optional: get LAN address from 34980A front panel and type into web browser to observe and control instrument from its web page

2. Install Agilent T&M Toolkit and Agilent VISA I/O Libraries
3. Install Agilent 34980A IVI COM driver from the provided CD or go to www.agilent.com/find/adn
4. Run Agilent I/O library configuration tool to add 34980A to config store
5. Start VS 2005, create new VB project
6. Run Agilent Instrument Explorer to find 34980A
7. Drag 34980A from Instrument Explorer to Code Window in VS 2005 and let Wizard generate initialization code, including instrument reset example

Steps Required to Install and Use PXI-1033 cardcage, PXI-2527 switch card, PXI-4070 DMM in Visual Studio 2005. These steps take much longer than the steps to install and use the Agilent 34980A, even if NI MAX, NI DMM and NI SWITCH are already installed and debugged.

1. With power off, install PXI-4070 and PXI-2527 in PXI cardcage
2. Shut down Windows, turn off PC
3. Install PCIe card in PC, connect cable to PXI cage, put covers back on PC
4. Turn on PXI cardcage, then PC (order is important), wait for Windows reboot
5. Install NI MAX (including NI VISA I/O Libraries)
6. Run NI MAX to add instruments to config store
7. Download and/or install NI DMM and NI SWITCH (1 GB total)
8. Find nidmm.vb and niswitch.vb on NI website, download
9. Start VS 2005, create new project
10. Run Driver Wrapper Wizard to create .NET-compatible ni dmm and ni switch driver
11. Add ni_dmm.vb and ni_switch.vb source code to project, write code to initialize variables
12. Write code to call DMM reset function
13. Run program. Resolve any errors, using NI Knowledge Base and online forums. (See Knowledge Base articles 3939UN5B and 2Q36UJ2H)

Benchmark results

A program was written in VB.NET that controlled a 34980A with an armature relay card and a PXI-1033 chassis with a PXI-4070 DMM and a PXI-2527 armature relay card. DAQ (scanned voltage readings) and EFT (single reading with associated switching) modes were used. The following software was installed in Visual Studio 2005.NET:

- Agilent T&M Toolkit (including Agilent I/O Libraries Suite)
- NI MAX (including NI I/O libraries)
- IVI COM driver for the 34980A switch/measure unit
- NI SWITCH and NI DMM IVI-C drivers (wrapped by T&M Toolkit for .NET)
- nidmm.vb, niswitch.vb

	EFT	DAQ
Armature switching		
Agilent 34980A w/ DMM, 34922A	60 ch/s	100 ch/s
70-ch, 300 V mux	16.7 ms	10 ms
NI PXI-1033 w/ PXI-4070, PXI-2527	77 ch/s	92 ch/s
32-ch, 300 V mux	13.0 ms	10.9 ms
High-speed FET switching*		
Agilent 34980A w/ DMM, 34925A	100 ch/s	348 ch/s
40/80-ch FET mux	10 ms	2.87 ms
NI PXI-1050 w/ PXI-4070, SCXI-1128	78 ch/s	343 ch/s
32-ch FET mux	12.8 ms	2.91 ms

Test Conditions:

EFT (Electronic Functional Test):

Set up DMM for DCV, 10 V range, lowest resolution, autozero off
Measure time to close a channel, trigger and read the DMM, open the channel

DAQ (Data Acquisition):

Download a scanlist
Set up DMM for DCV, 10 V range, lowest resolution, autozero off
Measure time to initiate the scan, return readings

*Note: the DMM is set up for 1 reading per channel. The scan speeds can increase if the scan list is repeated using trigger or sweep commands. The 34980A can scan over 1000 chan/s with the built-in trigger and sweep commands.

Conclusions

PXI solutions are expensive and hard to use in some development environments (especially Visual Studio.NET). Lack of SCPI puts you at the mercy of driver developers, and lack of LXI locks you in to constantly changing I/O bus standards. It attempts to be everything to everyone, and therefore is not as easily tuned to a particular application.

The 34980A switch/measure unit was designed to be a robust, inexpensive solution for a variety of data acquisition and EFT needs. It has just enough cooling, power, shielding and built-in functionality to

do its intended job without burdening you with the extra costs of industry-standard backplane requirements. Agilent chose to use an open industry-standard architecture – LXI – that is focused on low cost, high speed, and ease of use. In just two years since introduction, the 34980A has achieved wide industry acceptance and Agilent has sold thousands of units worldwide. It is also backed up by Agilent's worldwide support network and applications experts.

Appendix A

Code examples

Here are the programming requirements in the Visual Basic.NET environment for a simple EFT (close/measure/open) and DAQ (scanned) measurement using the DMM in DC volts on the 10 V range, scanning a 32 channel mux.

34980A and PXI Benchmark Program

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

Public Class Form1
    Dim pxidmm As InstrumentDriverInterop.Ivi.niDMM
    Dim pxiswitch As InstrumentDriverInterop.Ivi.niSwitch
    Dim myAgilent34980A As Agilent.Agilent34980A.Interop.Agilent34980A

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Dim rdg As Double
        Dim rdgs(500) As Double
        Dim t1, t2 As Long
        Dim i, n, samples, triggers, backlog As Integer
        Dim status As Short
        Dim scanlist As String
        Dim done As Boolean

        ' PXI EFT mode (using 32-channel PXI-2527, PXI-4070)

        pxidmm.reset()

        pxidmm.ConfigureAutoZeroMode(InstrumentDriverInterop.Ivi.niDMMConstants.AutoZeroOff)

        pxidmm.ConfigureTrigger( _
            InstrumentDriverInterop.Ivi.niDMMConstants.Immediate, 0.0)

        pxidmm.ConfigureMeasurement( _
            InstrumentDriverInterop.Ivi.niDMMConstants.DcVolts, 10.0, 0.001)

        n = 100
        pxiswitch.DisconnectAll()

        pxiswitch.WaitForDebounce(100)

        t1 = Agilent.TMFramework.Timing.CounterValue

        For i = 1 To n
            pxiswitch.Connect("ch0", "com0")

            pxiswitch.WaitForDebounce(100)

            If chkTrigDMM.Checked Then

                pxidmm.Initiate()
            End If
        Next i
    End Sub
End Class
```

(continued)

```
Do
    pxidmm.ReadStatus(backlog, status)
Loop Until (status <> 0)

' If desired to include execution time of fetch, read DMM results here:

If chkReadDMM.Checked Then pxidmm.Fetch(2000, rdg)

End If

pxiswitch.Disconnect("ch0", "com0")

pxiswitch.WaitForDebounce(100)
Next

t2 = Agilent.TMFramework.Timing.CounterValue

TextBox2.Text = n / Agilent.TMFramework.Timing.CalculateElapsedSeconds(t1, t2)

' PXI DAQ mode (using 32-channel PXI-2527, PXI-4070)

pxidmm.reset()

triggers = 32

samples = 1

scanlist = "ch0:31->com0;"

pxidmm.ConfigureAutoZeroMode(InstrumentDriverInterop.Ivi.niDMMConstants.AutoZeroOff)

pxiswitch.ConfigureScanList(scanlist, _
    InstrumentDriverInterop.Ivi.niSwitchConstants.BreakBeforeMake)

pxiswitch.ConfigureScanTrigger(0.0, _
    InstrumentDriverInterop.Ivi.niSwitchConstants.Ttl0, _
    InstrumentDriverInterop.Ivi.niSwitchConstants.Ttl1)

pxidmm.ConfigureMultiPoint(triggers, samples, _
    InstrumentDriverInterop.Ivi.niDMMConstants.Immediate, 0.0)

pxidmm.ConfigureTrigger( _
    InstrumentDriverInterop.Ivi.niDMMConstants.Ttl1, 0.0)

pxidmm.ConfigureMeasurement( _
    InstrumentDriverInterop.Ivi.niDMMConstants.DcVolts, 10.0, 0.001)

pxidmm.ConfigureMeasCompleteDest( _
    InstrumentDriverInterop.Ivi.niDMMConstants.Ttl0)
```

(continued)

```
t1 = Agilent.TMFramework.Timing.CounterValue

pxidmm.Initiate()

pxiswitch.InitiateScan()

pxiswitch.WaitForScanComplete(5000)

' If desired to include execution time of fetch, read DMM results here:
If chkReadDMM.Checked Then pxidmm.FetchMultiPoint(10000, samples * triggers, rdgs, i)

t2 = Agilent.TMFramework.Timing.CounterValue
TextBox3.Text = samples * triggers / Agilent.TMFramework.Timing.CalculateElapsedSeconds(t1, t2)

' 34980A EFT (using 70-channel 34922A)

myAgilent34980A.Utility.Reset()

myAgilent34980A.Scan.ConfigureScan("", 1, True)

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

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

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

myAgilent34980A.System.WaitForOperationComplete(10000)

myAgilent34980A.Display.DisplayEnabled = False

t1 = Agilent.TMFramework.Timing.CounterValue

For i = 1 To n
myAgilent34980A.Route.Close("1001")

' No need to wait for close; 34980A waits internally before executing next command
If chkTrigDMM.Checked Then

myAgilent34980A.Measurement.Initiate()

' If desired to include execution time of fetch, read DMM results here:

If chkReadDMM.Checked Then
rdgs = myAgilent34980A.Measurement.FetchNumbersOnly()
```

(continued)

```
Else ' if not returning a reading, wait for the init to finish
    myAgilent34980A.System.WaitForOperationComplete(10000)
End If
Else
    myAgilent34980A.System.WaitForOperationComplete(10000)
End If
myAgilent34980A.Route.Open("1001")
' No need to wait for open; 34980A waits internally before executing next command
Next
t2 = Agilent.TMFramework.Timing.CounterValue
TextBox5.Text = n / Agilent.TMFramework.Timing.CalculateElapsedSeconds(t1, t2)
' 34980A DAQ (using 70-channel 34922A)
myAgilent34980A.Utility.Reset()
n = 35
myAgilent34980A.Voltage.DCVoltage.Configure("1001:1035", 10.0, _
    Agilent.Agilent34980A.Interop.Agilent34980AResolutionEnum.Agilent34980AResolutionLeast)
myAgilent34980A.Voltage.DCVoltage.AutoZero("") = _
    Agilent.Agilent34980A.Interop.Agilent34980AAutoZeroEnum.Agilent34980AAutoZeroONCE
myAgilent34980A.Display.DisplayEnabled = False
triggers = 10
myAgilent34980A.Trigger.Count = triggers
myAgilent34980A.Scan.ConfigureScan("1001:1035", 1, True)
myAgilent34980A.Route.Delay("1001:1035") = 0.0
myAgilent34980A.System.WaitForOperationComplete(10000)
t1 = Agilent.TMFramework.Timing.CounterValue
myAgilent34980A.Measurement.Initiate()
```

(continued)

```
' If desired to include execution time of fetch, read DMM results here:

If chkReadDMM.Checked Then

    rdgs = myAgilent34980A.Measurement.FetchNumbersOnly

Else ' if not returning a reading, wait for the init to finish

    myAgilent34980A.System.WaitForOperationComplete(10000)

End If

t2 = Agilent.TMFramework.Timing.CounterValue

TextBox6.Text = n * triggers / Agilent.TMFramework.Timing.CalculateElapsedSeconds(t1, t2)

End Sub

Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

    pxidmm = New InstrumentDriverInterop.Ivi.niDMM("DAQ::2::INSTR", False, True)

    pxiswitch = New InstrumentDriverInterop.Ivi.niSwitch("PXI1Slot4", True, True)

    myAgilent34980A = New Agilent.Agilent34980A.Interop.Agilent34980AClass()

    myAgilent34980A.Initialize("TCPIP0::169.254.9.80::inst0::INSTR", True, True, Nothing)

End Sub

End Class
```

Related Agilent literature

Agilent 34980A Multifunction Switch/Measure Unit

Data sheet

5989-1437EN

Agilent E2094P I/O Libraries Suite 14.2

Data sheet

5989-1439EN

10 Good Reasons to Switch to LXI: Key advantages that enable better systems

Application note

5989-4372EN



Agilent Email Updates

www.agilent.com/find/emailupdates

Get the latest information on the products and applications you select.



Agilent Direct

www.agilent.com/find/quick

Quickly choose and use your test equipment solutions with confidence.



www.agilent.com/find/open

Agilent Open simplifies the process of connecting and programming test systems to help engineers design, validate and manufacture electronic products. Agilent offers open connectivity for a broad range of system-ready instruments, open industry software, PC-standard I/O and global support, which are combined to more easily integrate test system development.

LXI is the U.S. registered trademark of the LXI Consortium.

Remove all doubt

Our repair and calibration services will get your equipment back to you, performing like new, when promised. You will get full value out of your Agilent equipment throughout its lifetime. Your equipment will be serviced by Agilent-trained technicians using the latest factory calibration procedures, automated repair diagnostics and genuine parts. You will always have the utmost confidence in your measurements.

Agilent offers a wide range of additional expert test and measurement services for your equipment, including initial start-up assistance onsite education and training, as well as design, system integration, and project management.

For more information on repair and calibration services, go to

www.agilent.com/find/removealldoubt

www.agilent.com

For more information on Agilent Technologies' products, applications or services, please contact your local Agilent office. The complete list is available at:

www.agilent.com/find/contactus

Phone or Fax

United States:

(tel) 800 829 4444
(fax) 800 829 4433

Canada:

(tel) 877 894 4414
(fax) 800 746 4866

China:

(tel) 800 810 0189
(fax) 800 820 2816

Europe:

(tel) 31 20 547 2111

Japan:

(tel) (81) 426 56 7832
(fax) (81) 426 56 7840

Korea:

(tel) (080) 769 0800
(fax) (080) 769 0900

Latin America:

(tel) (305) 269 7500

Taiwan:

(tel) 0800 047 866
(fax) 0800 286 331

Other Asia Pacific Countries:

(tel) (65) 6375 8100
(fax) (65) 6755 0042

Email: tm_ap@agilent.com

Revised: 09/14/06

Product specifications and descriptions in this document subject to change without notice.

Microsoft, Visual Studio and Windows are U.S. registered trademarks of Microsoft Corporation.

© Agilent Technologies, Inc. 2006
Printed in USA, November 13, 2006
5989-5852EN



Agilent Technologies