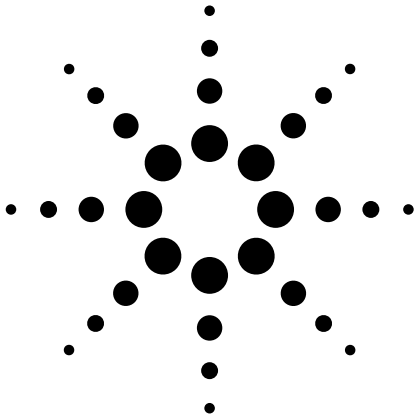
Select and configure the best data acquisition system for your application



Setting up a test system for data acquisition or electronic functional test can be a chore. But if you choose the right hardware and software for your application, you can alleviate many common headaches before you even begin assembling, configuring and programming your system. We have compiled a collection of our most popular application notes and demos to help you select, configure and make the most out of using your data acquisition system.

- **Compare data acquisition solutions from Agilent, Keithley, NI and Racal**

  This comprehensive and insightful selection guide gives on overview of switch/measure system types used in functional test and data acquisition environments and discusses factors affecting ease of software development for switch/measure solutions. In addition, it compares the most popular hardware from the industry's most respected companies. See the summary that begins on page 2.

- **Learn how to select the best temperature transducer for your application**

  Making the best temperature measurements starts with selecting the right temperature sensor for your application. This popular application note explores the advantages and disadvantages of a variety of sensors. See the summary that begins on page 6.

- **Take a tour of our newest data acquisition solution**

  The Agilent 34980A is an eight-slot mainframe that includes an optional built-in 6-? digit DMM. Choose from 19 optional plug-in modules that offer a broad range of functionality that includes DC to 20 GHz switching, counter/totalizer, digital I/O with pattern capabilities, and D/A converters – Experience the 34980A through a flash demo.

A Comparison of Leading Switch/Measure Solutions

Application Note Summary

This application note gives an overview of switch/measure system types used in functional test and data acquisition environments and discusses factors affecting ease of software development for switch/measure solutions. It also reports results of benchmark tests of switch/measure time for the Keithley 27xx DMM/data acquisition system; Racal 1256 switching system; Agilent 34970A and Agilent 34980A switch/measure units; Agilent 3499A switch/control mainframe combined with an Agilent 34401A digital multimeter; Agilent E1411B/E1476A VXI combination; and a National Instruments (NI) PXI-4070/SCXI-1128 PXI combination. The paper also discusses cost of ownership and ease of use issues for an actual system constructed from the tested hardware. It concludes with code examples in the Microsoft® Visual Studio®.NET environment for each of the instruments.

Switch/measure system types

Almost all test systems include a digital multimeter (DMM) and a bank of relays (switches), but the switch/measure function can be implemented in three different ways:

- Discrete instruments with cable interconnects
- VXI or PXI or PXI-hybrid mainframes
- Dedicated instruments with proprietary backplanes containing a DMM and a variety of switch cards

The type you choose will be heavily influenced by whether you plan to use the system for data acquisition (DAQ) or electronic functional test (ETF). For DAQ, you take numerous readings to characterize your device's performance. For EFT, you apply stimuli to a device, monitor the outputs for expected responses, and compare them to a set of limits.

The throughput you can achieve with the same hardware differs greatly in these two environments. Execution speed can be very high in DAQ mode — on the order of 1,000 readings/sec, depending on the measurement resolution and switching speed. In such systems, high-speed backplanes can improve throughput, since you need to transfer a lot of data. VXI and PXI systems shine here. An EFT system makes fewer readings and repeats them many times. Execution speeds are more like 100- 500 readings per second. A high-speed backplane does not help much for EFT applications, which makes the lower-cost dedicated switch/measure solutions a better choice.

Ease of programming

For programming instruments in test systems, engineers commonly use several application development environments (ADEs) that encompass both graphical and textual programming languages. The three most commonly used are VEE Pro, LabVIEW and Visual Studio 6.0, with Visual Studio.NET making gains. In the Visual Studio.NET environment, Agilent's T&M Programmers Toolkit and NI's Measurement Studio both make using instruments easier.

Using drivers vs. SCPI commands

If you are using standalone (rack-and-stack) test and measurements instruments in your system, you typically will send ASCII commands via a GPIB, USB or LAN interface using either SCPI (Standard Commands for Programmable Instruments) or software drivers. Cardcage-based systems such as VXI and PXI are most often controlled via data registers rather than ASCII commands, and you typically use drivers provided by the manufacturer.

The primary advantages of using drivers where there's a choice of SCPI or drivers:

1. The program is more transportable and more readable by other programmers
2. Built-in help during development via IntelliSense (VB6 and .NET)
3. Built-in state caching, if implemented, can improve execution speed
4. Instrument quirks may have been taken into account for you, reducing the chance that you will need support

The main disadvantage of using drivers is that they may not include all possible functionality. Another is that if the driver does not function correctly, it can be difficult to debug. For a detailed explanation of interface drivers, VISA, VXI*plug&play*, IVI, LabVIEW and VEE Pro drivers, see the full application note, "A Comparison of Leading Switch/Measure Solutions."

Do drivers affect execution speed?

Experiments with the hardware used in our benchmark tests show that VXI*plug&play* and IVI drivers can be as fast as SCPI (via VISA) in modern computers. Although there are some caveats to this, there is no reason to avoid drivers solely due to concerns about execution speed.

Effects of command sequencing and state caching

Whether you use SCPI or drivers, when you are programming instruments you need to pay careful attention to command sequencing. If you use SCPI, you will find that some commands cause other instrument states to change. A driver may take care of this for you, but it will have to send many query commands to the instrument to figure out what state it is in or send extra commands to the instrument to make sure it is in a given state. These steps can add execution time. You can solve the problem by using state caching, which is commonly implemented in IVI drivers. A state-caching driver keeps track of the current state of the instrument, and it saves command transfer time and command parsing time.

Benchmark tests

To maximize validity of our benchmark test results, we used VB.NET as a common development environment and ran tests on all instruments in both DAQ mode (scanned voltage readings) and EFT mode (single reading with associated switching).

We also wanted to show that NI, Keithley, Racal and Agilent hardware could work together in a real test system, so we constructed a test system using hardware from these companies. We used a variety of interfaces, including LAN for the Agilent 34980A and the Keithley 2701 (via a LAN hub); MXI-3 to connect the PC to the PXI cage; FireWire to connect the PC to the VXI cage; and GPIB, through both a dedicated PCI GPIB interface card and an Agilent 82357A USB/GPIB converter. See the system diagram (Figure 1) in the full application note.

Benchmark timing results

For details about the hardware, software and drivers we used for the test, see the full application note. We defined EFT time as the time to open and close one relay and trigger and read one DMM DCV reading on the 10-V range with a resolution of 1 mV (4.5 digits). Reported time is an average of 20 such measurements. We defined DAQ time as the time per reading to scan a 20-channel list, taking a measurement as each relay closes. See Table 1.

As you can see, DAQ mode produces much faster execution time, sometimes by as much as an order of magnitude. Also, there are wide variations in execution times; Keithley's use of fast reed relays didn't help much, for example, since the instrument was so slow at taking readings. PXI was fastest, but its performance, which was only measured using FET switches, was comparable to the 34980A FET switches.

Table 1. Benchmark test results

	Driver		SCPI	
	EFT	**DAQ**	**EFT**	**DAQ**
Agilent 34980A				
70-ch armature mux (34922A)	16.9 ms	10.4 ms	15.1 ms	10.1 ms
Dual 4x8 road matrix (34933A)	9.6 ms	[1]	8.4 ms	[1]
40/80-ch FET mux (34925A)	10.0 ms	2.7 ms	7.9 ms	2.7 ms
Keithley 2701-7703				
32-ch diff. reed mux	440 ms	68 ms	n/a	n/a
Racal 1256-138A/E1411B				
8 1x8 2-wire armature mux	4.13 ms	113 ms[2]	n/a	n/a
Agilent 34970A/34901A				
20-ch armature mux (34922A)	52.2 ms	22.8 ms	70.0 ms	252 ms
Agilent 3499A-N2266A/34401A				
20-ch armature mux (34922A)	29.5 ms	21.4 ms	35.3 ms	27.1 ms
Agilent VXI E1476A/E1411B				
64-ch 3-wire armature mux	30.1 ms	11.9 ms	46.6 ms[3]	2.94 ms[3]
NI SCXI-1128/PXI-4070				

Notes:

1 It is not possible to do a "scanned" measurement using a matrix. This is not the typical use model for a matrix, which is used most often for EFT testing.

2 The Racal 1256 took 2 seconds to process the "define scan list" command. That is why its data acquisition time was so long. Without that, the execution time would have been about 13 ms. Of that, 10 ms was the "trigger delay" parameter that was used, since the advance trigger output did not appear to work. If that delay were removed, the execution time would have been about 3 ms, which is consistent with normal DAQ modes.

3 The VXI SCPI numbers were gathered with an E1406A GPIB command module.

4 The SCXI-1128 was unable to generate a trigger back to the DMM, so a full handshake is not possible. Thus the SCXI measurement is a synchronous type, in which the DMM takes 20 readings using a sample interval, and its Measurement Complete signal is used to advance the scanning to the subsequent channel.

Cost of ownership

Overall, the Agilent 34980A and 34970A cost the least per slot. With PXI and VXI systems you need to pay for an expensive cardcage that was meant for high-speed instrumentation. You must also pay for an interface card on both the computer and the cardcage. This creates an overhead that must be added to the cost of every slot used. In addition, it can take considerably longer to implement a solution using discrete cards than by using a switch/measure box. The full application note includes a table showing prices of several switch/measure solutions

Summary of ease-of-use issues

We encountered numerous problems over the course of the 3-week evaluation period, which are summarized here:

1. **Keithley firmware update required.** If the program aborts or is ended without executing a "close" on the Keithley 2701, that instrument cannot be used again until power is cycled. We downloaded firmware revision A06 from the Web and reflashed the unit to fix the problem.

2. **Unusual behavior**. The Keithley 2701 was resetting the DMM aperture time to SLOW (5 PLC) whenever a ROUT:MULT:CLOS (Close multiple relays) command was sent. We had to add a command to reset the aperture time to .01 NPLC.

3. **NI-VISA setup changes required.** If you use NI VISA, you must also run MAX (Measurement Automation Explorer) and enable the "Passport to Tulip" interface driver or none of the Agilent VXI/GPIB instruments will be recognized. However, with NI Measurement Studio installed, this interface could not be used because the NI Passport interface driver calls AgVisa32.dll and it caused an exception upon exit.

4. **NI IVI-compliance problems.** Agilent's Driver Wrapper Wizard can only wrap IVI-C and VXI*plug&play* drivers that are properly installed according the IVI Foundation specs. NI-DMM and possibly other NI IVI-C drivers did not install correctly unless the "LabWindows/CVI examples" box was checked during the installation step. At the time of this evaluation, NI did not provide .NET compatibility for its instruments, but it did offer .NET-wrapped code that can be downloaded from its Web site and manually attached to a program. We did this for the NI-DMM driver, but we encountered namespace conflicts.

5. **Insufficient documentation.** NI's examples for use of its PXI-4070 DMM were in VB6 and VC++ and LabVIEW. NI's examples for DAQmx switches in .NET do not include hardware triggers, only software triggers. The hardware triggers require use of the backplane trigger buses, but the parameters that use these require strings, and there are no examples of usable strings. The IntelliSense help did not work, so we manually retrieved the intended help. However, when we tried using the names that it gave for trigger buses in the code they required, we could find none that worked. A LabVIEW example revealed that new path-based naming conventions were required. However, none of the names in the LabVIEW example worked. We initiated another support session and discovered that the SCXI-1128 does not support handshaking, only synchronous mode (unidirectional triggering).

6. **PC shutdown required for swapping cards in PXI.** When you move a card in a PXI cardcage, you must cycle power, but you can't do it with the PC power on since the PXI backplane is an extension of the PCI bus in the computer. A time-consuming PC reboot is required.

7. **Version conflicts.** We encountered many version conflicts. For example, while troubleshooting SCXI switching problems, NI-DAQ 7.1 was installed on a Measurement Studio 7.0 installation. This caused numerous problems that were only solved by uninstalling all NI software and reinstalling it, which took the better part of a day.

8. **How can Agilent and NI hardware be controlled from one program?**
   NI MAX can find devices on all NI interfaces, but cannot directly control Agilent interfaces, such as the FireWire interface to VXI, USB/GPIB converter or the PCI GPIB card. Agilent Instrument Explorer cannot currently find PXI devices. For a test system that needs to communicate with devices from both vendors, the solution was to install Agilent I/O libraries in "side-by-side" mode (described in the I/O Libraries Help file) and enable the Passport-Tulip interface driver in NI MAX. This causes VISA calls to those interfaces to be routed from NI VISA to Agilent VISA, which then controls the relevant Agilent interfaces while still allowing NI interfaces such as MXI-3 to work directly through NI VISA and Passport drivers.

Code examples

The full application note includes 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, with a 20-channel mux.

Looking at the code, we reached these conclusions:

1. Switch/measure units with internal DMMs take a lot of the work out of DAQ measurements, because the triggering is done for you. It took only a few minutes to create working code with these instruments. It took 2 weeks to get the PXI/SCXI measurement to work, largely because of difficulty with triggering requirements.

2. SCPI strings can be concatenated, making long strings. This saves a little execution time because there is no extra overhead in multiple function calls. However, if it is not necessary to send the string in the first place, the command parsing time in a slow instrument can easily dwarf the function call execution time. If high throughput is a requirement in your application, consider using state-caching drivers instead of SCPI.

3. IVI-C (with .NET wrappers), IVI-COM and VXI*plug&play* drivers are all very similar in usage in the .NET environment. All provide various degrees of IntelliSense help. We found that IVI-COM help is much more useful than the other two. The PXI/SCXI help was not adequate, requiring frequent reference to the on-line manuals and several e-mail support sessions with NI.

Selecting temperature transducers for data acquisition systems

Application note summary

Introduction

Understanding the advantages and disadvantages of the various approaches to measuring temperature will help you get better results when you are collecting temperature data with a data acquisition system. Choosing the right temperature transducers and using them correctly can help you avoid problems and get results you can count on. This application note provides an overview of the four most common types of temperature transducers used in data acquisition systems and discusses the advantages and disadvantages of each approach.

Transducer types

The four most common types of transducers used in data acquisition systems are resistance temperature detectors (RTDs), thermistors, IC sensors and thermocouples. Each of them works best in certain measurement situations, so it is important to know when to use which type. Factors to consider include performance, useful range, cost and convenience. Table 2 summarizes the advantages and disadvantages of each type.

For a more complete description of each of the transducer types, see the full application note. The full application note also includes a detailed look at how thermocouples work to help you overcome some of their inherent drawbacks.

Table 2.

	RTD	Thermistor	IC Sensor	Thermocouple
Measurement type	**Absolute**	**Absolute**	**Absolute**	**Relative**
Advantages	• Most stable • Most accurate • More linear than thermocouples	• High sensitivity • Fast • Two-wire measurement	• Most linear • Highest output • Inexpensive	• Self-powered • Rugged • Inexpensive • Wide variety of physical forms • Wide temperature range
Disadvantages	• Expensive • Slow • Current source required • Small resistance change • 4-wire measurement • Self-heating	• Nonlinear • LImited temperature • range • Fragile • Current source required • Self-heating	• Limited to 250°C • Power supply required • Slow • Self-heating • Limited configurations • Large mass	• Nonlinear • Low voltage • Reference required • Least stable • Least sensitive

Improving transducer accuracy

With all four types of temperature transducers, reducing noise will help you improve the accuracy of your measurements. Reducing noise is especially critical when you are using thermocouples, since electrical noise affects thermocouple measurements most dramatically.

Noise typically comes from one of three sources and each type of noise has a unique solution:

Noise type	Cause	Solution
Common mode noise	**Ground loops**. Thermocouples are very susceptible to ground loops, as their metal junction may be directly attached to a machine or component that has a higher potential than the data logger. through both the high and low leads, through the data logger and back to the source via ground.	Select a data acquisition system with high impedance to ground, often specified as common mode rejection. You also can insert electrical insulation between the
Normal mode noise	Magnetic fields creating a current in the measurement loop. Normal mode noise can occur when a thermocouple wire is run near high-current wires or machinery	Shorten leads, use twisted pair wiring, avoid running measurement wires near high-current sources
Electrostatic noise	Rotating machinery	Use a shielded measurement wire

Conclusion

It is not difficult to make accurate and reliable temperature measurements with a data acquisition system if you choose the right sensor for your application. When you select your sensor, consider transducer cost, temperature range, accuracy, ruggedness, sensor output, thermal settling times and error modes, such as self-heating. Also, pay careful attention to the instrument system you choose. The correct sensor is useless if the data acquisition system cannot measure its output accurately and repeatably.

Related Agilent Literature

- 34970A Data Sheet 5965-5290EN
- 34980A Data Sheet 5989-1437EN

www.agilent.com

www.agilent.com/find/emailupdates
Get the latest information on the products and applications you select.

www.agilent.com/find/agilentdirect
Quickly choose and use your test equipment solutions with confidence.

Agilent Open Connectivity
Agilent Open simplifies the process of connecting and programming test systems to help engineers design, validate and manufacture electronic products. Agilent combines a broad range of system-ready instruments, open industry software, PC-standard I/O and global support to accelerate test system development. For more information, see:

www.Agilent.com/find/Open.

By internet, phone, or fax, get assistance with all your test & measurement needs

Online assistance:
www.agilent.com/find/assist

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
(fax) (31 20) 547 2390

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

Korea:
(tel) (82 2) 2004 5004
(fax) (82 2) 2004 5115

Latin America:
(tel) (650) 752 5000

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

Other Asia Pacific Countries:
(tel) (65) 6375 8100
(fax) (65) 6836 0252
(e-mail) tm_asia@agilent.com

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

Printed in the USA July 29, 2005