Designing Automated Test Systems

A Practical Guide to Software-Defined Test Engineering
**Contents**

*Preface* ................................................................................................................................. 5

NI Test Engineering Strategy ................................................................................................. 7

Recommended Test System Development Process ............................................................. 8

**Step 1: Identifying Measurement Needs** ........................................................................ 11

1.1 Identifying the Scope of Your Test System ................................................................. 11
   Scenario 1: Testing a Single Product ............................................................................. 11
   Scenario 2: Testing an Entire Product Family ............................................................ 11
   Scenario 3: Testing Multiple Product Families .......................................................... 12
   Future Plans and Other Considerations ........................................................................ 12
   Avoiding Scope Creep ..................................................................................................... 13
   Example ............................................................................................................................. 13

1.2 Choosing a Core Hardware Platform ............................................................................ 14
   Processing Power and Data Throughput ...................................................................... 14
   Scalability ........................................................................................................................ 14
   Measurements Diversity ............................................................................................... 14
   Communication with Other Buses and Instruments ....................................................... 14
   Timing and Synchronization ......................................................................................... 15
   Lifetime ............................................................................................................................. 15
   Example ............................................................................................................................. 15

1.3 Determining the Required Instrumentation ................................................................ 18
   Basing Instrument Choices on Measurement/Stimulus Rather Than Instrument Type .... 18
   Test Accuracy Ratio ......................................................................................................... 19
   Other Considerations ....................................................................................................... 20

**Step 2: Best Practices for Selecting Hardware** ................................................................ 21

**Step 2.1: Choosing Rack Type, Size, and Power Distribution Unit** ................................. 22

2.1.1 Choosing the Rack ....................................................................................................... 22
   Environment of Use ......................................................................................................... 22
   Portability ........................................................................................................................ 23
   Size 23 ............................................................................................................................... 23
   Cooling ............................................................................................................................... 24
   Other Rack Design Considerations ................................................................................ 25
   Real-World Example ........................................................................................................ 25

2.1.2 Designing Rack Layout ............................................................................................. 26
   Weight Distribution ......................................................................................................... 27
   Operator Needs ................................................................................................................ 27
3.2.3 Create Code Module Templates .................................................. 59
3.2.5 Document Code ......................................................................... 61
3.2.6 Use Source Code Control .......................................................... 62

Step 3.3: Choosing Your Instrument Driver Paradigm ................................ 64
3.3.1 Instrument Driver Options .......................................................... 64
3.3.2 Plug and Play Instrument Driver .................................................. 65
3.3.3 IVI Instrument Driver ................................................................. 65
3.3.4 Direct I/O .................................................................................. 66

Step 4: Assembling the Test System ...................................................... 68
4.1 Assembling Components ................................................................ 68
4.1 Building and Routing Cables ........................................................ 69
4.2 Installing and Activating Software .................................................. 73
4.3 Configuring and Validating in MAX ............................................... 74
4.4 Validating the Tester ..................................................................... 76

Step 5: Deploying the Test System ...................................................... 77
5.1 Deploying the Test System Software .............................................. 77
  System Replication ....................................................................... 77
5.2 Activating the Software ................................................................. 81
5.3 Deploying the Test System Hardware ............................................ 81
  1. Space .......................................................................................... 81
  2. Power .......................................................................................... 81
  3. Networking ................................................................................ 82
  4. Environment .............................................................................. 82
  5. Safety .......................................................................................... 82
  6. Maintenance .............................................................................. 82

©2009 National Instruments. All rights reserved. CompactRIO, CVI, LabVIEW, Measurement Studio, MXI, National Instruments, NI, ni.com, and NI TestStand are trademarks of National Instruments. The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States and other countries. Other product and company names listed are trademarks or trade names of their respective companies.
Preface

Defining a corporate test strategy is critical to reducing cost and maximizing the efficiency of your product development and manufacturing organizations. You should decide on a predominant test strategy based on where your organization is today as well as where it plans to be the next five to 10 years. At the highest level, a strategy is typically dominated by the volume and mix of your product portfolio. You can represent volume and mix in four quadrants, as shown in Figure 1.

I. It is difficult for large companies to have a common test strategy. Each division is like a separate company and has unique requirements. However, in a division, you can begin to form a common test strategy that typically falls into different quadrants. Standardization is a key strategy for balancing volume and mix. (See Hella KGaA Hueck & Cö case study on page 6.)

II. It is cost-prohibitive to build dedicated testers to support each product. Each tester should be flexible enough to support multiple products. (See Benchmark Electronics case study on page 6.)

III. Typically, a small test organization needs a handful of testers, each of which should be product-specific and optimized for cost. (See any small startup.)

IV. Corporations must optimize for continuous flow by employing test strategies such as parallel test that maximize capacity. (See Harris Communications case study on page 7.)

For more than two decades, National Instruments has collaborated with industry-leading companies to document test strategy best practices and techniques for building more effective automated test systems. NI works with a multitude of sources as well as members of the NI Automated Test Customer Advisory Board (AT-CAB) to capture these best practices. The AT-CAB community is a cross-section of industries that work to leverage constantly changing commercial technologies while maintaining long-term supportability of current test systems.
The adoption of software-defined test systems is the most significant trend among these industry-leading test engineering teams. Engineers are using software-defined test systems to achieve new levels of measurement performance and lower test costs. The quick return on investment from these benefits is contributing significantly to the mainstream adoption of the software-defined test system approach.

Thousands of companies are building software-defined test systems based on NI software tools and the open, multivendor PXI hardware standard. According to the PXI Systems Alliance, more than 100,000 PXI systems will be deployed by the end of 2009, and the number of deployed PXI systems is expected to double in the next decade. Below are three examples of large companies that have adopted the software-defined test system design approach, even though their respective company test strategies fall into different quadrants. The content for this guide has been developed using best practices shared by NI AT-CAB members and the expertise of the NI test engineering and product research and development teams.

Hella KGaA Hueck & Co.

This €3.9 billion international automotive company serves the automotive lighting and electronics needs of leading vehicle makers worldwide. Hella created a standard universal tester (SUT) to serve as a homogeneous yet modular platform capable of testing a variety of products with minimal modification and changeover time between tests. Hella was able to deploy over 200 SUTs globally.

Quadrant I - Standardization

Benchmark Electronics

Benchmark Electronics, a leading electronics contract manufacturer, created a standard test platform to serve a wide range of product functional test categories and to deliver the best instrument capability with a low cost and small production footprint. The tester is internally called the Target Tester. Benchmark has deployed over 25 test systems.

Quadrant II - Flexibility
NI Test Engineering Strategy

NI is a medium-sized ($500M to $1B USD) high-tech company whose products serve several markets including the design, industrial, and test and measurement markets. From a mix-versus-volume perspective, NI can be classified as a high-mix, low-volume organization (quadrant II) because it has more than a dozen primary product lines with more than 1,000 unique devices. Based on these characteristics, the corporate test strategy at NI emphasizes flexibility and reuse. By building flexible testers, NI test engineers can test all of the products within a product line on a single test station as well as quickly adapt the test stations to address the requirements of the more than 200 new products that NI releases each year. Each test station is used when performing regression testing and product validation/verification in design and for functional test in production.

The NI test engineering team has developed and implemented a common test software and hardware platform that can be scaled across multiple product lines. The strategy was to create and maintain a standard test development software environment with flexible capabilities that engineers can use to focus on developing tests rather than reinventing their own, unique test frameworks. This not only fostered test code reuse across new products and product lines but also provided common interfaces with enterprise systems to help improve quality tracking and consistency in test data storage.

To build the framework, the test engineering team used standard commercial off-the-shelf (COTS) technologies to maximize personnel and capital resources. They selected NI TestStand software to handle test management, development, enterprise integration, and operator interfaces and NI LabVIEW graphical programming software to develop the test modules. This software framework connects to a modular test hardware platform based on a PXI core platform and a combination of hybrid PXI and GPIB instrumentation. The hardware framework offers a common base of PXI modular instrumentation but also provides for unique configurations based on product line needs. Figure 2 shows the block diagram of the modular, software-defined test architecture. The architecture, which is maintained by the NI test engineering team, offers commonality across all test systems.
**Recommended Test System Development Process**

This five-step guide details the recommended process for building software-defined test systems from start to finish. It presents these test engineering best practices in a practical and reusable manner and features specific examples used by industry-leading test engineering teams. It also references a scalable, software-defined production test system developed by the NI test engineering team for testing I/O modules for the NI CompactRIO platform, which is shown in Figure 3. CompactRIO is an advanced embedded control and data acquisition system designed for applications that require high performance and reliability.

*Figure 3. The NI CompactRIO product family features more than 50 modules.*
Figure 4 depicts the production test system from different viewpoints.

First the guide focuses on best practices for choosing your test system hardware. Topics in this section range from making instrument decisions for your test system to choosing your rack type and power distribution unit. Next, the guide examines how to connect your instrument to your device under test (DUT) by offering best practices for designing your switch network, choosing your mass interconnect system, and building a custom fixture. Figure 5 shows a close-up of a production test system switching, mass interconnect, and custom fixture.

Figure 5. A Close-Up of a Production Test System Switching, Mass Interconnect, and Custom Fixture
After discussing various best practices for choosing hardware, the guide delves into designing a strong software framework that you can use across multiple tests and products. Topics in this section range from making appropriate driver decisions to integrating code modules into a test executive (Figure 6), which is the software layer that handles the operations common for all test scenarios with respect to a given test system. Finally, the guide features best practices for validating, deploying, and maintaining your test system over its lifetime.

Figure 6. Use the NI TestStand Sequence Editor to develop automated test systems.
Step 1: Identifying Measurement Needs

Your company has a test strategy in place. Your test architects have done an excellent job in putting together a solid software framework that takes your test engineers’ needs into consideration. Now that you know your constraints as well as priorities, you are ready to start designing your test system. The first step is to determine the measurement requirements for your device(s) under test. This section outlines the various factors to consider when evaluating the measurement needs of your test system. It also examines the process the National Instruments test engineering team underwent to choose instrumentation for the automated test system used to test more than 50 NI CompactRIO I/O modules.

1.1 Identifying the Scope of Your Test System
The first step in identifying your test measurement needs is to determine the system’s scope. Is the system testing a single product, an entire product line, or a series of product lines? Take a look at a simple example of how determining the scope can significantly change test system requirements.

Scenario 1: Testing a Single Product
Assume that you are a test engineer working for a semiconductor company. Your immediate goal is to design a system that can test the rise time, nonlinearity (integral nonlinearity or INL and differential nonlinearity or DNL), and current leakage specifications of the digital-to-analog converter (DAC) shown in Figure 1.

![Digital-to-Analog Converter](image)

- Rise Time = 5 ns
- Resolution = 8 Bits
- Current Leakage = 10 µA

*Figure 1. DAC under Test*

To ensure that you test the device rather than the test system, you need a set of instruments with better specifications than the DAC under test. Thus, your test system must have a high-speed measurement instrument with a rise time that is faster than 5 ns or a bandwidth that is greater than 700 MHz (bandwidth = 0.35/rise time). In addition, you must fit the system with an instrument that has a current sensitivity greater than 10 µA. Finally, the system must have an instrument with resolution greater than 8 bits to appropriately measure the DAC code widths and perform the nonlinearity tests.

Scenario 2: Testing an Entire Product Family
Now consider building a system that can test the rise time, INL and DNL, and current leakage specifications of the entire family of analog-to-digital converters (ADCs) shown in Figure 2.
To test the DAC product family shown in Figure 2, your system must incorporate instruments that have specifications superior to that of the entire product family. Thus, the instruments in the test system must have the following:

a. A rise time that is faster than 5 ns or a bandwidth that is greater than 700 MHz (bandwidth = 1/rise time)

b. Current sensitivity greater than 1 µA

c. Resolution higher than 16 bits

**Scenario 3: Testing Multiple Product Families**

It is tempting to widen the test system’s scope as much as possible to have a common platform for several programs; however, the following pitfalls can occur:

- To accommodate different product lines, the complexity of the core test system increases, thus increasing nonrecurring, recurring, and material costs
- Maintaining configuration control is difficult among a larger group
- Obsolescence issues increase
- Costs increase on high-production-rate product lines requiring multiple test systems even though a DUT may use only a small portion of the test system capabilities
- Designing test systems for a new product line becomes difficult because of the constraints to use only the existing capabilities of the system
- Keeping up with state-of-the-art technology grows more difficult as test capabilities start to stagnate

**Future Plans and Other Considerations**

In addition to understanding your current tester needs, you need to evaluate its future requirements. Are you going to use the tester to test additional product families going forward? Will you add new products to the current product family? If you answered “yes,” then you must also consider the measurement needs of these future additions. If you are certain that your test system needs will expand but are unsure of the measurement requirements of your future products, you must design your system...
using a modular platform that is easily scalable. For example, you should make interfaces easily available to your test system such as USB, LAN, and GPIB so that you can quickly add new measurement capability to the system that is not available in the rack such as a USB-based modular instrument.

**Other Considerations**
- Budget and timeline
- Expected life span of the test system
- Additional test requirements such as fault diagnostic capability
- Skill level of operators
- Product volume

**Avoiding Scope Creep**
Ensure you understand the project vision and spend time documenting and determining the project objectives. Produce a project plan document that describes the test system deliverables. It is a good idea to document what is in scope as well as out of scope for absolute clarity. Verify the content of this document with the key stakeholder, spending time to walk them through it, and ask them to sign off on it. You should plan for some degree of scope creep in most projects; therefore, it is important to design a process to manage these changes. You can then implement a simple process of document, consider, approve, and assign resources. Use a change control form and change log from the start of the project and communicate the process for using these forms to the customer and project team. Attach a cost and time to each change so the customer is clear about its impact. Implementing a formal process helps ensure there’s a clear business value for the requested change.

**Example**
Put the concepts discussed previously into practice by examining a real-world example. As with other scenarios in this guide, this example is based on the automated test system for testing a variety of CompactRIO I/O modules. The scope of this system is to test a product family of more than 50 I/O modules for the CompactRIO platform, which is an advanced embedded control and data acquisition system designed for applications that require high performance and reliability. In addition, the test system must be scalable to test future CompactRIO module releases.

![Figure 3. The NI CompactRIO I/O product family consists of more than 50 modules.](image)

To design a system to test the entire family of CompactRIO I/O modules, NI engineers thoroughly evaluated the measurement requirements of all 50 modules. Based on this analysis, they compiled a comprehensive list of every measurement they needed to make and identified their most stringent measurement needs. For example, NI engineers determined that the test system required the ability to
source currents up to 10A and measure voltages as low as 1 mV. Finally, they chose suitable instrumentation to address these stringent needs.

1.2 Choosing a Core Hardware Platform

After determining the measurement needs of your test system, you can begin architecting your hardware framework. Many test engineers jump straight into matching their measurement needs to instruments available on the market. A better approach is to first pinpoint a suitable test platform that can serve as the core or nucleus of your test system. You can choose from many platforms, most of which are based on one of the four most commonly used instrument backplanes/buses—PXI, GPIB, USB, and LAN. Because each of these buses has at least some advantages and limitations (as discussed in chapter 4 of the Software-Defined Test Fundamentals guide), you often have to build hybrid test systems based on multiple platforms. Even so, it is often a best practice to pick a prominent or core platform for your test architecture. This section outlines some of the factors you must consider when choosing a core platform for your test system.

Processing Power and Data Throughput

Assess the worst-case computational power and throughput rates when selecting a controller.

Scalability

Another factor is the ease with which you can scale or modify your system. This is especially important if your test system has the potential to change during the course of its lifetime. One example of this is if you are building a system to test a product family that is continually expanding. In such a case, you may need to add new functionality to the system without making significant changes that could force you to redesign your test rack.

Measurements Diversity

The platform that serves as the core of your test system must be able to address a significant portion of your test system needs. Thus, if your system requires the ability to make low-level DC measurements along with high-speed rise time measurements, you must select a platform that is capable of accommodating mixed-signal instrumentation. In general, you should choose a core platform that accommodates at least 80 percent of your test system’s measurement needs.

Communication with Other Buses and Instruments

As mentioned previously, each instrument bus and platform has distinct advantages and disadvantages. By building hybrid systems based on multiple instrument buses, you can take advantage of the strengths of several different test platforms. A hybrid architecture also increases the flexibility of your test system by allowing you to choose from a larger pool of instruments on the market. Such flexibility is especially important if you are building a complex and dynamic test system that will change over time. The first step toward building a hybrid architecture is choosing a core platform that can communicate with instruments that are based on a variety of instrument buses.
**Timing and Synchronization**
When designing a test system composed of multiple buses and platforms, you must ensure that your core platform can synchronize those instruments by sending triggers and sharing clocks.

**Lifetime**
Another factor to consider is the lifetime of your test system. If you expect to use your system for several years, you should choose a platform that can stand the test of time. Sometimes products and platforms go end of life (EOL). It is often difficult to service and maintain products like these in a test system. For this reason, you must choose a proven platform for which products and replacements will be available for several years. For long military programs, consider vendor support agreements or lifetime buys of equipment that may be required.

**Example**
These six criteria helped NI engineers select a core platform for the automated test system they used to assess the CompactRIO I/O modules. The following is an evaluation of test system needs based on the criteria:

1. **Processing Power and Data Throughput**: This system required a multicore processor and a high-speed data bus that could support the expected data analysis.
2. **Scalability**: Because the CompactRIO product line is continually evolving and growing, it was essential for the system to use a core test platform that is highly scalable to be able to add new functionality for future CompactRIO platform releases.
3. **Measurements Diversity**: The CompactRIO tester had to be capable of testing the large variety of I/O types on the more than 50 modules for the CompactRIO platform. These include ±80 mV thermocouple inputs, ±10 V simultaneous-sampling analog I/O, 24 V industrial digital I/O with up to 1 A current drive, differential/TTL digital inputs with 5 V regulated supply output for encoders, and 250 Vrms analog inputs. Because of the variety of signal types and voltage ranges these modules need to address, the NI engineers also required the tester to make a wide range of measurements.
4. **Communication with Other Buses and Instruments**: The CompactRIO platform is continually growing and future product plans are hard to predict, so NI engineers needed a tester with a high level of flexibility to accommodate a vast range of instruments that are based on different buses.
5. **Timing and Synchronization**: Because there is a possibility for multiple instruments based on different buses to coexist in the system, the core platform needed to seamlessly synchronize these instruments.
6. **Lifetime**: The test system was designed to work for the entire lifetime of the CompactRIO product line, and thus required a core platform that is continually growing and whose components were likely to be serviceable and replaceable for several years.

Based on these six requirements, NI engineers chose the NI PCI eXtensions for Instrumentation (PXI) platform as the core of the test system. PXI provides several benefits that meet test system needs.
The most prominent reason for selecting PXI is its modular and scalable architecture. In PXI, all instruments share many components, such as the chassis, power supply, and controller, so adding new instruments is as easy as plugging a module into one of the empty slots in the chassis. This framework makes it simple to add new measurement capabilities to the system in a cost-effective manner. More importantly, the modular architecture of PXI makes it possible to incorporate new capabilities without changing the physical dimensions of the test rack.

![Image](image.png)

*Figure 4. PXI provides a modular and scalable architecture.*

In addition to being scalable, PXI is capable of addressing a diverse set of measurement needs. There are nearly 1,500 PXI products that provide the following functionality:

- Analog input and output
- Boundary scan
- Bus interface and communication
- Carrier products
- Digital input and output
- Digital signal processing
- Functional test and diagnostics
- Image acquisition
- Prototyping boards
- Instruments
- Motion control
- Power supplies
- Receiver interconnect devices
- Switching
- Timing input and output
- RF and communications

This diverse range of products made PXI suitable to serve as the core of the test system, which required a broad range of functionality to test the entire family of more than 50 CompactRIO I/O modules.

Another advantage of PXI is its ability to connect to multiple instrument buses including USB, Ethernet/LAN, and GPIB. This functionality helped increase the flexibility of the test system by enabling the addition of instruments based on various instrument buses. This feature of PXI was especially useful...
when the tester’s functionality required expansion to incorporate tests for the new NI 9227 5 ADC current input module.

The test method for the NI9227 5A current input module involves connecting a precision shunt in series with the input terminals of the module, and sourcing current values from -5 to 5A (entire range of the module) through the input terminal using a power supply. To ensure that the power supply is sourcing the right value, the resultant voltage drop is measured across the precision shunt using the NI PXI-4071 7½-digit digital multimeter (DMM).

![Figure 5. Circuit for Testing the NI 9227 Current Input Module.](image)

The measured voltage is then converted to a current value using the following formula:

\[ I_{\text{shunt}} = \frac{V_{\text{shunt}}}{R_{\text{shunt}}} \]

This value is then compared with the current that was meant to be sourced by the power supply. Once it is ensured that the power supply is sourcing the right value, the current sourced by the power supply is compared to the current measured by the NI 9227. The difference is entered as a calibration value on the NI 9227.

Because there is currently no PXI module that can source a current of 5 ADC, NI engineers chose the Agilent N6702A modular power system mainframe along with the N6754 DC power module, which is capable of sourcing up to 20 ADC to conduct the test.

The GPIB port on the PXI controller enabled the easy integration of the Agilent power supply into the test system.
Additionally, PXI provided the ability to synchronize the device with other instruments in the test system. For instance, calibrating the NI 9219 analog input module requires measuring the excitation voltage being sent from the module as well as the voltage measured by the input channels of the module simultaneously. For this test, NI engineers leverage the star trigger on the PXI chassis backplane to synchronize both DMMs. In addition to the star trigger, the PXI chassis backplane has several other timing and synchronization features including the following:

- 100 MHz differential system reference clock
- 10 MHz reference clock signal
- Star trigger bus with matched-length trigger traces to minimize intermodule delay and skew
- Trigger bus to send and receive high-speed timing and triggering signals
- Differential signals for multichassis synchronization

Finally, the PXI platform’s consistently growing product portfolio along with data from the analyst firm Frost & Sullivan, which stated that PXI revenue in measurement and automation is expected to increase at a 17.6 percent compound annual growth rate (CAGR) through 2014, provided the certainty that servicing the tester throughout its lifetime would not be arduous.

These benefits made PXI the best platform to serve as the core for the CompactRIO module test system.

1.3 Determining the Required Instrumentation

Now that you have a better understanding for determining your test system measurement requirements and choosing your hardware platform, you are ready to start selecting the specific hardware instruments you need to conduct your measurements. This section features some best practices for choosing instruments for your tester.

**Basing Instrument Choices on Measurement/Stimulus Rather Than Instrument Type**

Test engineers often choose an instrument based on type rather than need. For example, many engineers select DMMs to make high-accuracy measurements even though in many applications, the accuracy of a data acquisition board may be sufficient. Such decisions often result in higher costs, so you should choose your instrument based on your measurement need rather than the instrument type.

Following this practice was highly beneficial when NI engineers selected a method to calibrate the NI 9219 thermocouple module in the test system described in this guide. Typical calibration methods involve using expensive instruments that cost upwards of $50,000 USD. In this particular test system, however, the NI 9219 is calibrated using a Keithley source measure unit (SMU) and the NI PXI-4071 7½-digit PXI DMM.
Figure 6. Circuit for Calibrating the NI 9219 Voltage Input Module

This is possible because the PXI-4071 DMM has an accuracy that is substantially higher than that of the NI 9219. Table 2 provides a comparison between the accuracy of the NI 9219 and the PXI-4071.

<table>
<thead>
<tr>
<th></th>
<th>NI 9219</th>
<th>NI PXI-4071 (2 yr cal values)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gain Error (ppm)</td>
<td>Range Offset Error (ppm)</td>
</tr>
<tr>
<td>125 mV</td>
<td>3000</td>
<td>120</td>
</tr>
<tr>
<td>60 V</td>
<td>1000</td>
<td>20</td>
</tr>
</tbody>
</table>

ppm = parts per million

Table 2. NI 9219 vs. NI PXI-4071 Accuracy Comparison.

As you can see from Table 2, the accuracy of the PXI-4071 is several times that of the NI 9219. In addition, because the PXI-4071 was already required for testing other CompactRIO modules, using it for calibrating the NI 9219 helped to reduce the overall cost of the test system substantially.

Test Accuracy Ratio

Another best practice for choosing your test system instruments is to calculate the test accuracy ratio (TAR) to ensure that the accuracy of your measurement equipment is substantially larger than the accuracy of the component you are testing. If you do not meet this criterion, then you may see significant measurement error caused by both the device under test and the test equipment, making it impossible to know the true source of error. Because of this, engineers use TAR to determine the relative accuracy of the measurement equipment and the component under test. You can calculate TAR with the following formula:

\[
TAR = \frac{\text{Desired Accuracy of the Component Under Test}}{\text{Accuracy of Measurement Equipment}}
\]

Your TAR value should equal 4 or more, depending on the test you are performing and the test certainty you require. TAR was one method NI engineers used in determining the suitability of the PXI-4071 for...
calibrating the NI 9219. Because the accuracy of the PXI-4071 is more than 10 times that of the NI 9219, it was deemed suitable for calibrating the NI 9219.

Other Considerations
In addition to determining measurement needs and pinpointing the right combination of instruments to test the device under test using the TAR, you often need to make several decisions unique to each test system. In the case of the automated test system built for testing CompactRIO modules, NI engineers had to give special consideration to accommodating the measurement needs of the NI 9219 universal input module. The NI 9219 can operate in several different modes, including a full-bridge mode. When operating in this mode, it sources a specific excitation voltage to facilitate current flow through the bridge. At the same time, the module measures the voltage drop across the load that is the DUT. The measurement recorded by the NI 9219 is the ratio of the voltage drop across resistor R and the excitation voltage provided by the NI 9219.

*Note:* No current flows through the ADC because it is a high-impedance circuit.

![Diagram of NI 9219 in Full-Bridge Mode](image)

*Figure 7. NI 9219 in Full-Bridge Mode*

In such a measurement, there are two different sources of error. The first is the voltage measurement and the second is the voltage excitation. To test the NI 9219, both sources have to be measured simultaneously. The test station therefore uses two PXI-4071 7½-digit DMMs to measure these two sources of error at the same time.