

## Compiling a PXI ATE system

# ATE by the numbers



*So, have you finally decided to build your next PXI system yourself? Well, if you are, read on. Building your own test system can be very rewarding, both from a professional standpoint as well as from a support standpoint, as you will know the system intimately. You can also potentially, with emphasis on potentially, save the cost of a system integrator. In this article, Bob will review the sequence of events and the questions to ask when building a test system.*

*By Bob Stasonis*

If anyone in my audience is a *seasoned veteran*, I invite you to write me and correct or enhance what I'm doing here. Perhaps we'll use the commentary for a future article.

The goal is to make your process as efficient and logical as possible. Now given the premise that test is kind of a *black art*, I can never give you all the answers, but hopefully enough to limit the potential problems that can ensue. Development of a test system can be broken down to a sequence of technical decisions. As I see it, there are ten basic elements:

- UUT (Unit Under test) specifications
- Test times
- Budgets
- System configuration
- Instrumentation
- Switching
- System controller
- Test system language
- Interface and cabling to UUT
- Reality check

Let's look a little closer at each.

### UUT specifications

Unless you are building a *universal test system* (which is a topic that is too broad for this article), your first task is to thoroughly understand exactly what you are tasked with testing and what engineers expect to be tested. There is often a fine line between what engineers may expect, the amount of time allotted for test in the manufacturing cycle, and what you can afford to build. Very often the first step in designing a test system is to negotiate the required tests, time allotted in the manufacturing process for test, and expected first pass yield.

If possible, the test spec should have as a minimum:

- UUT theory of operation
- I/O requirements (such as number of connections, clocks, voltages)
- List of tests and required responses

- UUT configuration – for example, are there multiple UUTs in one test pass (PCB Pallet)

### Test times

While most test engineers would probably like to ignore this spec, the fact of the matter is that the overall test time is often as critical as the fault detection capability. I once heard from a test engineering manager that the biggest fear that he had was *more concrete*. What he meant was if the test time was too long, they needed more test systems, which equals more floor space (hence the concrete reference), and also more system operators, capital equipment purchases, and so forth. So the fastest test times, without sacrificing fault coverage, are important. Sometimes testing multiple UUTs simultaneously makes sense. In any case, efficient test code and the optimal instrumentation are factors.

Another factor to consider is what is expected if a test fails. Do you note the test fault and fail the board, or is the tester expected to perform some initial diagnostics? Of course, diagnostics will add to test times. If first pass yield has been calculated, you can factor in roughly how a fault will affect the efficiency of the test system.

## Budgets

The test system budget is very often a rather arbitrary figure that only includes the cost of the hardware. Labor is often considered free when a system is built in house. I personally don't recommend this as it makes it difficult to do a true cost comparison between building a test system and outsourcing system integration to a 3rd party.

A system budget should include:

- All hardware and software investments (programming language environment and any utilities required)
- Labor to assemble the systems
- Estimated software development time
- Test fixturing
- Time required to debug and perform test system acceptance.

## System configuration

By configuration, I mean how will this system be used in test? Will UUT handling be automated or manned by one or more operators? Automation has its own set of issues that range from the cost of conveyors to how the tester will integrate into the production line.

Manual operation using system operators has its own set of issues (shown in Figure 1). Ergonomics comes into play early in the game, as this will tell you just how much room you will have for instrumentation, switching, loads, and so on. How will the operator work with the system? Standing or sitting? How convenient are the system controls and displays? Do they take into account any safety regulations?

In one example, a company was building testers for their Asian facilities. Turns out that the Asian managers like to look over the test systems to see that the operators are working efficiently. That meant that the test system could be no more than 1.5 meters in height!

## Instrumentation

Now that we know what we are testing, we can select the instrumentation. For purposes of compactness and easy synchronization where necessary, we should look at PXI availability first. As soon as possible, determine how many instrument modules will be necessary. Also, how many module slots wide is each instrument? This will be important when selecting the PXI chassis or chassis' needed. Typically, PXI chassis are available



Figure 1

in 4, 8, 14, 18, 21, and 31 slots. Multiple chassis may also be considered. You should also keep in mind that you may need to add an additional module or two in the future, so open chassis slots are important.

If the measurement or stimulus type you require is not available in PXI, what are the alternatives? Choices include GPIB, LAN, USB, and PCI. Each selection can potentially influence system size and controller type. PXI modules all have Windows drivers, but your alternative instrument choices may not. Verify that the vendor of choice supports the Windows environment up front.

## Switching

Signal switching is needed for many reasons – applying power, connecting loads, and sharing resources. The latter is usually considered only when multiple UUTs are being tested – test one UUT and switch the test to the next, and so on. But the sharing of resources is almost always possible in a single UUT environment as well. To reduce the number of instruments required, you should consider whether a particular stimulus or measurement channel could be shared during the test sequence. Keep in mind that this may effect test times unless you are careful.

When selecting switching, there are several questions to ask:

- What are the basic parameters? This includes voltage, current, and bandwidth.
- What is the purpose of switching? That will determine whether basic relays, multiplexers, or a full cross point matrix are required.
- Be very aware of your entire measurement channel. This means you must look at the test specifications and the planned measurement path – this includes the test fixture, Interface Test Adapter (ITA), cabling, connector types, switching, and the instrument in the path. In Figure 2, a measurement channel including instrumentation, switching, and signal conditioning are highlighted. For another example, too often a test engineer selects an RF/Microwave switch based on the specified insertions losses and does not consider the rest of the measurement path. They find out too late that they should have specified a more accurate switch as the insertion loss of the switch equals the engineering spec for the entire measurement channel.

### System controller

The choice of a controller may seem trivial, but it can affect system footprint, upgradeability, and overall budget. There are two choices when it comes to PXI:

- Embedded controllers – Basically a two-to-four-module-wide PC that plugs into slot 1 of a PXI chassis, embedded controllers are designed to help create a small, compact system footprint, as the system is potentially enclosed in a single PXI chassis. In some applications, embedded controllers are available to facilitate real-time testing.

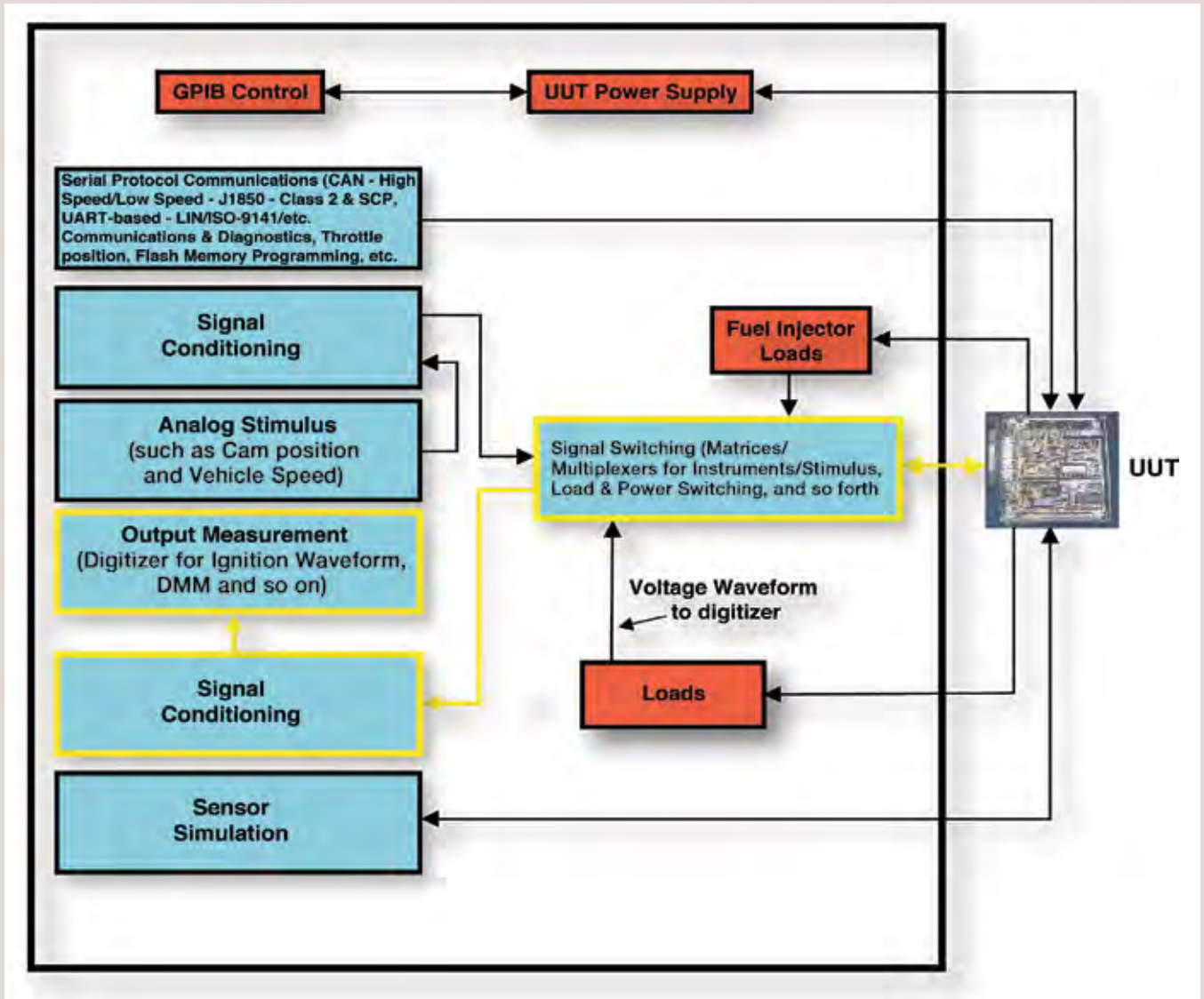


Figure 2



Figure 3

The small size creates some problems. Embedded controllers usually require external peripherals or a network connection to install or update software. There are no PCI slots available, which is an issue if an instrument you require is only available in a PCI form/function. The cost of an embedded controller can be two to three times that of a standard PC as well.

- PCI-to-PXI Bridge, Laptop-to-PXI Bridge – This device with the long name basically allows use of lower cost external PCs, either a desktop or a laptop variety. Basically, this is a pair of modules where the user plugs a module into the PCI bus of an external PC (or the PCMCIA or CardBus slot of a laptop) and the other module plugs into slot 1 of a PXI chassis (either an electrical or fiber optic cable completes the connection). Also in the case of multiple chassis configuration, it's essential and much easier to have all modules in synchronization, as you can daisy chain multiple modules together (shown in Figure 3). Hardware and software upgrades are inherently easier as the external PC is designed for such purposes.

The downside is that your tester footprint is potentially larger. There is also an added bridge segment, as this option is essentially the same as the bridging in the PXI backplane. Highly coherent testing can be slightly compromised, as this adds clock cycles to tester operation.

## Test system language

Now you're ready to write your test *symphony* – what will you play it on? There are many choices of test languages. If your company has not yet standardized on a programming language, you have the awesome responsibility to set the test department's direction. There are several directions you can go – graphical or textual. Let's highlight both.

The graphical de facto standard in PXI is National Instruments' LabVIEW. It is a graphical programming that is well recognized by many test engineers and system integrators. It is relatively easy to find someone in test engineering who understands this language. There are some users who feel that LabVIEW is difficult to document processes. If you are in an industry where software documentation is a critical issue (industries like the medical industry and Mil/Aero applications come to mind), you may want to look closely and judge for yourself before going down the graphical path. Agilent's Vee is a similar environment and basically has the same pros and cons of LabVIEW.

Then there are those people who still want text-based test program development. For you, there are many choices:

- Visual Basic and Visual C++ from Microsoft
- ATEasy from Geotest
- LabWindows from National Instruments



Figure 4

Each offers good reasons to choose their product, and has the same good and bad points of a text based language. It should be noted that ATEasy and LabWindows are focused strictly on test applications, whereas Microsoft products are more generic.

The bottom line is that you need to select a test language based on several factors: local support, similar applications that show it applies to your needs, and a comfort level that the software is evolving and will be around for a while.

## Interface and cabling to UUT

So by now, you've designed your tester, ordered the materials, and selected a test language ... now all that's left is to connect it to the UUT and start testing, right? If only it were that easy. We all know that a chain is only as strong as the weakest link. The way you connect and disconnect from the UUT will affect long term reliability and repeatability.

Before defining a test system interface, it is a good thing to know the estimated quantity of products to be tested. High volume production requirements call for very robust test fixturing. Lower volume applications may need something as simple as some cables and mating connectors. If you will be testing more than one product or several variations of a product, you may need to spend more on an ITA. ITAs (shown in Figure 4) allow for fast test fixture replacement with minimal effort. A good quality ITA should allow for many test fixture changeovers with minimal degradation of the connections for repeatability.

It is also important to keep in mind the types of cabling you will be using. Remember I mentioned earlier about considering the entire measurement path? The cable quality, length, and wire sizes all play a factor in the accuracy of a test system.

## Reality check

Now, after all that homework is complete, ask yourself this question – was it really cheaper or faster to do it yourself? Now, this question applies to the actual cost of your labor rather than just the cost of the hardware. Can you put your engineers to better use in other areas beside integration? Too often, the actual labor cost seems negligible, as you already have the staff. There is no single answer here. But, if you have long term plans for additional test systems, you need to ask that right after your first system is built.

“ The bottom line is that you need to select a test language based on several factors: local support, similar applications that show it applies to your needs, and a comfort level that the software is evolving and will be around for a while. ”

### Conclusions

I hope I have not frightened any of my readers. And I hope I have not missed any major element – hence the reason I asked our more expert readers for input to be discussed in future issues. But I firmly believe that a logical approach to a test system design can make the effort easier and potentially save money in the long run. It really is easier than ever to build your own test systems. Just don't think it's as easy as opening a catalog and calling your friendly neighborhood PXI vendor. **PXI**

**Bob Stasonis** is a sales and marketing manager for Pickering Interfaces. He has written numerous papers and articles on the subject of electronic test. Over the last 25 years, Bob has held technical, sales, and marketing positions with Teradyne, GenRad, Schlumberger, and Cimtek (formerly Wayne Kerr). Bob is a member of the board of directors and past President of the PXI Systems Alliance, and a member of the board of directors of the LXI Consortium. Bob is also currently the chairperson of the Boston chapter of the IEEE Instrumentation and Measurement society, and the Vice President of Marketing for the American Society of Test Engineers (ASTE).



### Pickering Interfaces

2900 NW Vine Street

Grants Pass, OR 97526

Tel: 541-471-0700 • Fax: 541-471-8828

Email: [bob.stasonis@pickeringtest.com](mailto:bob.stasonis@pickeringtest.com)

Website: [www.pickeringtest.com](http://www.pickeringtest.com)