A Non-Arduous Way to do Signal Routing

Larry V. Kirkland
WesTest Engineering
810 Shepard Lane
Farmington, Utah 84025
801-451-9191 ext 124

Abstract: There has been much work, re-work, duplicating work, test & find work, mind boggling work and just plain time consuming engineering work performed to accomplish optimal signal routing from the Interface Test Adapter (ITA) to the instrument. Accomplishments and technology improvements often get discarded or overlooked. All too often, the workforce starts at ground zero for every TPS routing scheme.

Optimally, it would be beneficial to establish a minimum set of requirements for any given TPS Software Development Environment (SDE) signal routing automation. When new or better techniques are invented and proven then the requirements should be upgraded. At a minimum, the user should be able to enter an Interface Definition where each UUT signal and pin is defined and enter the ITA Database Information by selecting an interface pin for each Unit Under Test (UUT) connection then the SDE should automate the routing. Also, as an option, the user should be able to manually select the routing desired from the ITA to the instrument.

This paper will show a non-arduous way or automated way to do signal routing on complex Automatic Test Equipment (ATE) platforms. The paper will present a set of requirements and more or less set an initial standard.

Another factor is the SDE output data should be used by the Test Executive to provide the user with all pertinent ATE to UUT routing information. This is extremely beneficial in determining station problems and verifying optimal routing. So, the SDE (coupled with the compiler) and the Test Executive should both handle and present all pertinent information associated with routing.

INTRODUCTION

Input and Output (I/O) multiplexers or switch matrices are common in high performance Automatic Test Systems. This allows for a limited number of dedicated test resources and reduces the overall cost of a system. Switching applications can have incredible densities and can handle low-level or high-level voltage and current requirements. High performance Automatic Test Systems have dynamic routing hardware that provide great flexibility for signal routing and signal multiplicity. Signal routing hardware is a great achievement and should be incorporated in high performance systems. With the advent of sophisticated signal routing hardware the need to aid the TPS developer with automated tools to figure signal routing through these I/O matrices should be required or requested by the customer.

Every automated test system involves some aspect of switching or signal routing. This implementation can span from simple general-purpose relays that control power to a device under test (DUT) all the way to complex matrix configurations that route thousands of test points to dozens of instruments. When the number of relays to control is small, the test code required to control them is straight-forward. When the number of relays and routes enters double-digits or spans multiple switch modules, it becomes advantageous to use some kind of routing database to manage the switch connections[1].

One would conjecture the automated routings system could consist of what is outlined in the proceeding paragraphs. Actually, automating routing at this time in ATE evolution is almost considered a requirement. It can be very difficult to include all the requirements in a system solicitation so one can say it has become expected. If minimum requirements are standardized then there can be little confusion and the requirement for a precise definition of everything the ATE is expected to do will not be needed.

AUTOMATED TECHNIQUES

The application software applies the power of a particular software package to perform a particular function. The Software Development Environment is a database program that has been developed with Microsoft's ACCESS database generator. The SDE provides the test engineer with the ability of inputting UUT testing data and generating a Test Program. The SDE can output Test Requirements Documents (TRDs), Source Code, and TPS Quality Reports. The SDE can also accept source code as an input, providing database update, translation and rehosting capabilities. It should be noted that prior to using the SDE the test engineer must develop a strategy for testing the UUT (Unit Under Test). This requires that an engineering analysis be preformed so that good data is supplied to the SDE.

At the beginning of the TPS development the UUT pin and the signal requirements needs to be identified. This Interface definition Entry form (see Figure 1) provides a means of entering the UUT pins and their test requirements into a database. The data entered in this form is used in the TPS Test Strategy Report. It is also used in automatically building the ITA database.

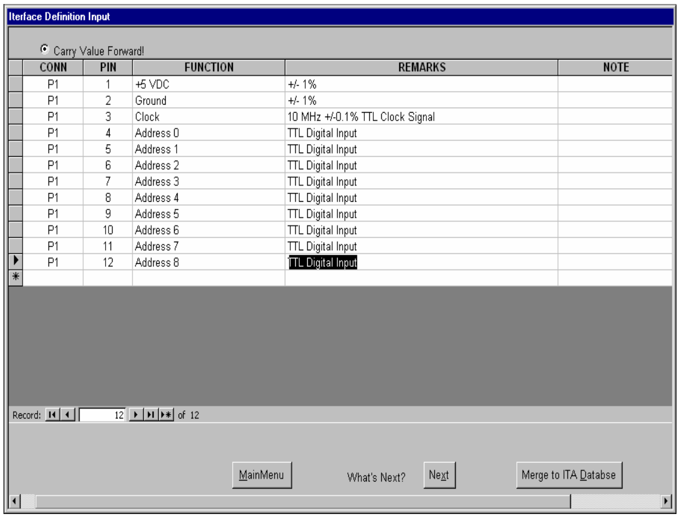

Figure 1. Interface Definition Entry

The ITA Data Base is used to create the Interface Test Adapter (ITA) Database (shown in figure 2). The ITA Database is used by the Compiler for locating UUT connections, making switch connections and assigning resources. The database is built by entering the UUT and tester connections provided by the ITA. The UUT connection information is imported from the Interface Definition Form.

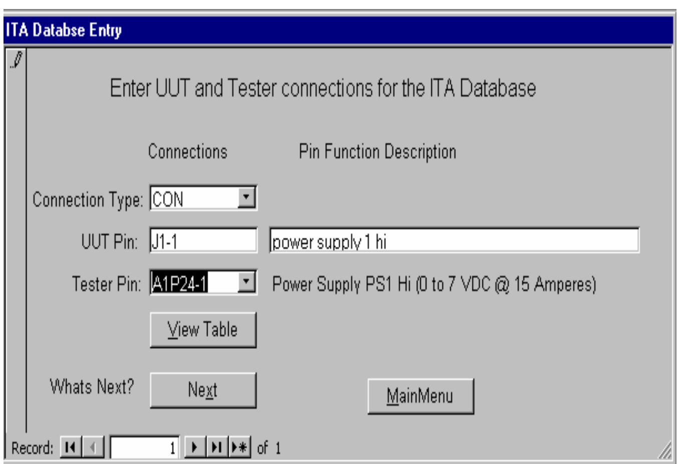

Figure 2 – ITA Database Entry Form

Connection information defines the type of connection that will be provided by the ITA. Connection Type may be CON, JMP, IN, or OUT. No other connection types are valid. CON defines UUT to Tester connections, JMP defines tester to tester connections, IN defines digital UUT inputs, OUT defines digital UUT outputs.

In the Tester Pin field enter the tester pin that the Interface Test Adapter (ITA) will connect to the UUT Pin. A description of the Tester Pin will automatically be displayed to the right of the tester Pin field. Note: This form must be completed before any tests can be generated with the Test Input Form.

The stimulus view shown in figure 3 of the test input form is used to define stimuli data as an input. Stimulus data such as UUT connections, stimuli source, and power levels are input via this form.

Source – Selects the tester resource to be used.

Connection - Enter the UUT connector pin in this field.

Special Fields – Additional fields may appear stimulus type and source selections have been made. These include such items as pulse width, frequency and reference connections.

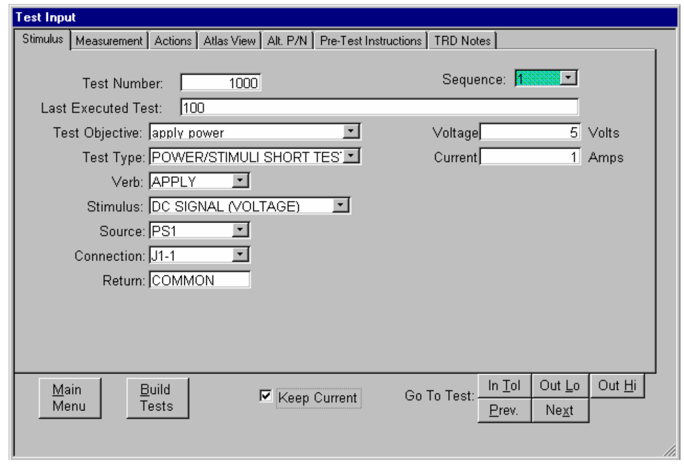

Figure 3. Stimulus Test Input

The Measurement data is entered via the Measurement View on the Test Input Form as shown in figure 4. Measurement fields such as UUT pin data, expected values, and test limits are input in this form. A test may have more than one measurement but it can only have one decision. Measurement signal routing is highly matrix oriented and should be considered a high priority when evaluating the ability of an automated routing system's performance.

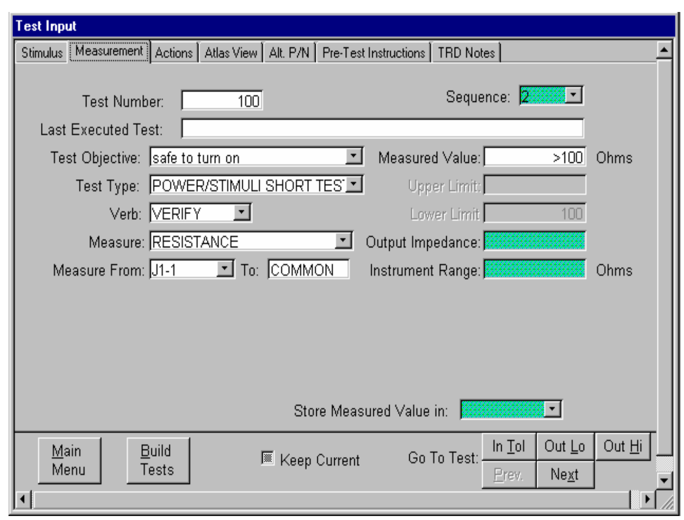

Figure 4. Test Input Form – Measurement View

Measure From:/To: – Enter the UUT connector pins in these fields. Note: COMMON is a valid entry.

SDE MANUAL ROUTING

As in figure 5, the signal routing through a relay is programmed by the APPLY SHORT entry in the stimulus window. Manual routing or the ability to create your own code rather than the automated code generation provided by quality software is important. No matter how much automation exists, there are times when the engineer must take over and produce a precise sequence of steps that make up a specific the test and diagnosis routing. Also, the ITA database sets up the routing for the CNX HI variable.

Figure 5. Relay Connect

TEST EXECUTIVE ROUTING FUNCTIONS

Another major factor to be considered is the profound power of the features of the Test Executive. In reality, the Test executive is the strength of testing and diagnosis. It is certain we must have a strong instrument sweet to test or units, however, the test executive is just as important as strong instrumentation[2].

Test Executive has a selection to show the flow diagrams as shown in figure 6. The dynamic flow generator aids the developer with a visual display of present conditions including switching data for system switch banks and relays. Updates are continuously made by the test executive. This display is a readily discernable indication which is vital not only for the technician but for the test integrator. Imagine the integrator, when typically they needed to review the code an search through the ATE schematic to determine how the signal is being routed to the instrument. The integrator can now readily determine if the setup is correct. This not only saves time but can improve the quality of the TPS.

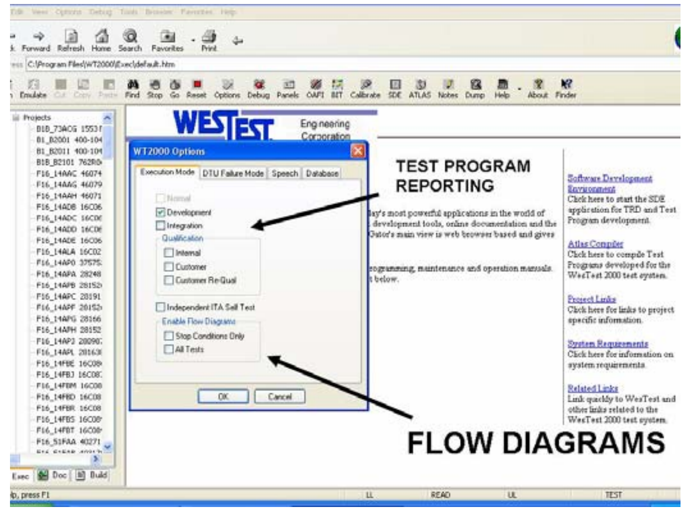

Figure 6. Flow diagrams

Figure 7 shows a selection of "Dynamic Flow Diagram Generator"; which aids with a visual display of present conditions including switching data for system switch banks and relays. Selections can be made for specific instruments or routing including Test Points.

Actual routing shown by a Test executive selection is critical for both the TPS developer and the Shop Technician. The TPS developer can use this information to debug/verify proper signal routing. The shop technician can use this information for various diagnostic purposes and to verify the ATE is working or not working. Signal tracing and tracking has always been critical to any debug procedure. It is truly beneficial to have the actual signal routing information readily available at all stages of a TPS from development to the end user.

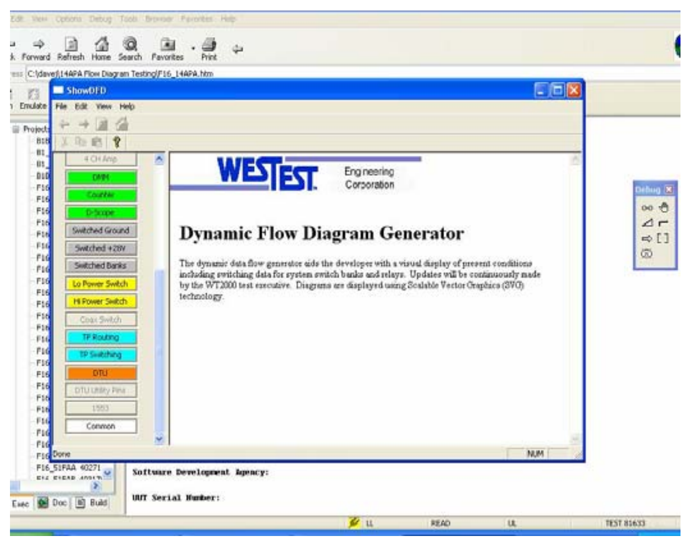

Figure 7. Dynamic Flow Selection

Figure 8 shows a selection of "Test Point Routing"; test point routing can be critical for TPS development and integration. Also, the shop technician often needs this information.

The signal flow circuit shown gives a readily discernable picture of exactly how the test signal connects to the instrument. For the technician to request this type of information be displayed is an indication of excellent job performance and an understanding of what needs to be displayed and to whom it needs to be displayed. Let us never forget that great ideas come from all users – not only the engineer but the end user also. When engineers and technicians experience truly state-of-the-art software systems it is difficult and somewhat demeaning to revert to a software system which provides very limited functions and end user information.

Figure 9 shows connections or signal routing through the power supplies. The system indicates the precise signal characteristics. All information pertaining to the connections are shown so the user can readily identify all the test equipment areas in use. There are problems which arise in testing that require checking the power supplies connections and settings.

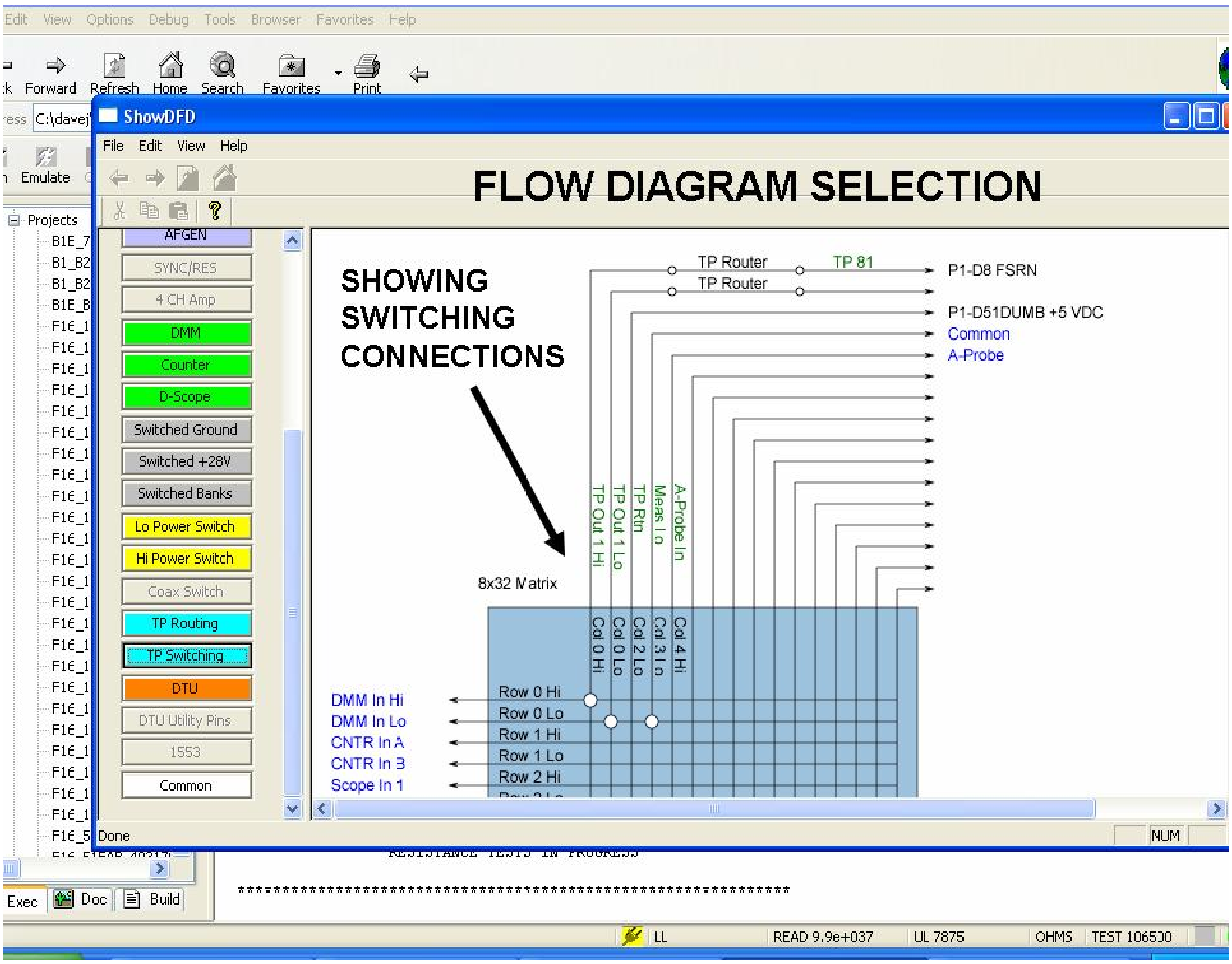

Figure 8. TP Routing

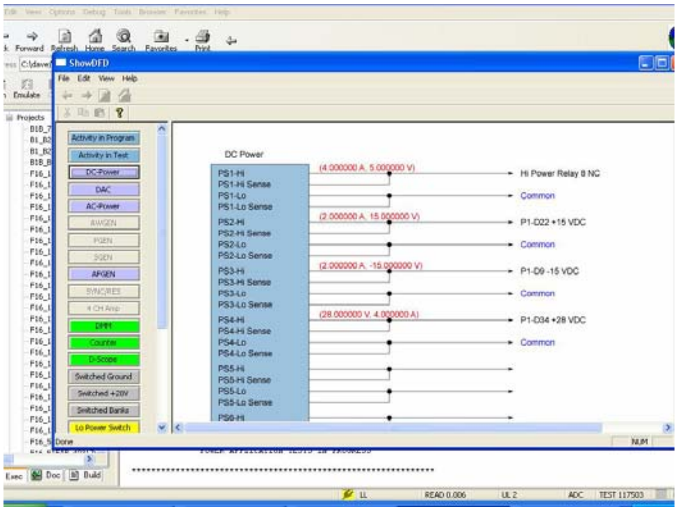

Figure 9. Dynamic Flow Selection

MINIMUM SET OF REQUIREMENTS

The minimum set of requirements for automated signal routing might consist of the following:

1. A means of entering the UUT pins and their test requirements into a database.
2. A Tester Pin field to enter the tester pin that the Interface Test Adapter (ITA) will connect to the UUT Pin.
3. Automate the signal routing for the UUT connections for Stimulus and Measurement tests.
4. A Test Executive that displays the actual signal routing from the UUT pin to the test instrument.

An established set of minimum requirements provides an avenue whereby new and improved techniques can be developed. This also prevents a re-development of an already proven science or technique. It is somewhat disappointing to see a new system with obvious weaknesses that does not provide great techniques that have proven so effective in TPS development and end user use. Software development tools are critical in TPS development or rehost strategies. Strictly viewing system hardware and not evaluating the TPS software is not cost effective. Short term additional costs can propagate into high long term costs.

To be used efficiently, all software needs certain hardware components. These pre-requisites are known as system requirements or system resources and are often used as a guideline as opposed to an absolute rule. Software can be classified as having two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resource usage in newer versions of software, system requirements/resources tend to increase over time. Software should actually play a bigger part in driving upgrades to existing systems than technological advancements.

ENGINEER'S SKILL/ROBUST SOFTWARE

Automated routing is a tremendous asset to the skilled test engineer. However, the ultimate responsibility to proper routing is still in the hands of the engineer.

No matter how many checks and double-checks, integration is always a smoke test. Lesson-learned seems to be never-ending with TPS development. It is a fact that signal routing and ITA wiring are critical to producing a quality noise-free stimulus or measurement signal.

Where things are wired to and how they are routed in the ITA to the ATE interface is always critical. Also, resource allocation is critical to proper ITA/TPS design. Resource allocation has many aspects when it comes to signal analysis, especially when precise measurements are to be taken.

Providing the skilled engineer with robust automation tools can reduce the obligation of hours of so-called busy work. The skill of the engineer can be greatly augmented by the skill of the TPS development software and the Test Executive.

Shown in Figure 10 is a LabVIEW connection scheme. The CNX VI has been programmed to route the signal. This is one method that required a full function instrument routing database.

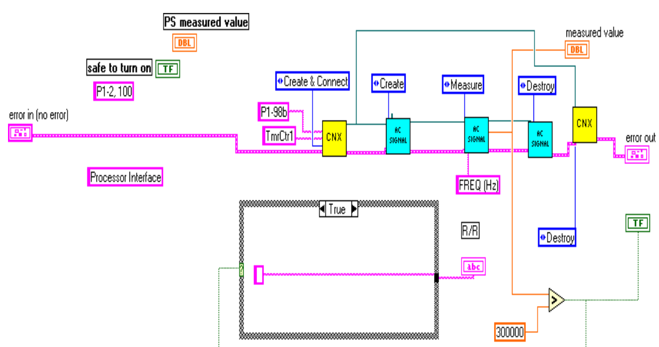

Figure 10. LabVIEW connection scheme

SUMMARY

The author has seen the evolution of great TPS development and end user tools. When I see weak systems that seem to start over with technology innovations it makes me wonder. I wonder how and why this could happen and if anyone understands what is going on and if anyone can help with the prevention of backward technology. By backward technology I mean we continue to "Re-invent the wheel"[3] instead of progressing from a certain achieved point. To reinvent the wheel is to reproduce a method that has already previously been created and *optimized* by others.

All too often, test technology takes a backward leap from the aspect of both hardware and software. I've seen new systems developed and sold which are very weak in instrument functionality and performance. These systems prove useless over a period of time but often a substantial investment has already been expended before the problem is resolved. Red flags seem to be unnoticed and ignored even when some one in the know is consulted.

Powerful software routines are an absolutely critical part of test equipment. Weak software and/or drivers produce weak TPSs and contributes to excessive software support issues. Also, weak software contributes to problems in the test and repair area. Our weapon systems need to be continually supported with advanced technology and we need to improve our processes and then set a baseline. Often the testing world has been told the Power of the Test is the software. It is true testing requires robust test hardware but the software is also a critical part of the test system.

Let's join forces and progress; we can work together for a common goal and for the common good. Re-inventing the wheel technology is not progressive technology. Shouldn't we support outstanding breakthroughs and transfer technical leaps from one system to another?

References:

1. National Instruments, Using Microsoft Excel and NI Switch Executive in Test System Development, http://zone.ni.com/devzone/cda/tut/p/id/4596
2. Kirkland, Larry V., Why did we add LabVIEW applications to our ATLAS TPSs?, IEEE AUTOTESTCON 2011, Sept. 2011.
3. Kemper, Steve, *A Story of Genius, Innovation, and Grand Ambition*, HarperCollins