Let the scope take the strain

Automating oscilloscopes: how to save time and make better measurements. By Daniel Ruebusch.

Oscilloscope measurement automation has become imperative for timely and accurate debug and validation. In many cases, the number and complexity of tests demanded by a technology standard or application prohibit any attempt at manual measurements. For example, the DisplayPort compliance tests require 312 different test conditions at eight waveform locations – 2496 unique measurements! An automated oscilloscope can complete these measurements in a few hours, where manual measurement will require multiple man days.

The benefits of automating commonplace and mission critical oscilloscope measurements are compelling.

- Save time. An automated measurement system will make the same measurements faster, so you get test results sooner, shorten test cycles and get to market faster.
- Save test and engineering resources. Let your scope do the work, freeing test and engineering resources for other challenges.
- Make better measurements. Remove user error and get better greater measurement consistency.

While the concept of saving time and resources by automation is widely understood, the value of making better measurements through automation cannot be underestimated. As data rates and signal frequencies increase, losses that could once be safely neglected now have a significant impact. Tools exist that allow oscilloscope users to de-embed lossy circuit elements and correct for these non idealities. These tools require additional steps which increase the time required to make measurements and the likelihood of measurement variations due to error. However, automating measurements allows users to accurately set up de-embedding of lossy circuit elements once, and have the scope apply these corrections in all future measurements.

The benefits of automation are compelling, but the myriad options, programming expertise and upfront investment are often difficult barriers to overcome.

Automation options

Starting with a simplified view, oscilloscope automation can be divided into two primary categories, based on the choice for the centre of automation: external controller based; or oscilloscope based automation

The difference between these two set ups is illustrated in Fig 1. External controller based automation is the traditional set up, where a central computer controls all instruments in the automated measurement environment. Here, the scope is just another instrument and measurements are made based on the instructions issued by the central controller. There are many choices available for automating oscilloscopes in this way. Common choices range from programming languages such as C/C++/C# and Python, graphical programming environments like National Instrument’s LabVIEW and Agilent’s VEE, and The MathWork’s MATLAB. While all of these are powerful tools, they all demand significant upfront investment to develop the automation environment and specific test procedures. Further, all require significant expertise to implement successfully. However, all are proven and effective automation techniques. Many companies have an existing measurement automation set up into which they want to integrate a scope and major oscilloscope manufacturers support this form of automation.

But there are many test and measurement environments in which no existing automation setups have been developed but where there remains a critical need to automate common
oscilloscope measurements. In these instances, the financial, temporal and personal investment required to set up an effective automation environment poses a critical challenge. Over the past five years, automation solutions have been developed in which the oscilloscope is the automation controller. The impetus for this shift in automation technique has largely been to mitigate these barriers.

Scope centric automation

The first and most ubiquitous form of oscilloscope centric automation is the compliance application. Compliance applications automate the measurements necessary to test compliance of a device against the requirements of the technology standard. The governing body of a technology standard, for example USB 2.0, writes a series of tests required for a device to be compliant with the standard to ensure interoperability. Major oscilloscope vendors then take these required tests and put them into an automated measurement environment within the oscilloscope software.

But what if there are other tests you want to run besides those required by the standards body? What if there is a whole suite of tests you run regularly that are completely separate from a given technology standard or its compliance tests? What if you want to control switch matrices, signal sources or temperature chambers? What if you want to de-embed the loss due to the cables and switches in your signal path or embed the response of the pcb trace proposed for your device?

You might go running for a team of software developers and countless lines of Python or C++ – in the past, this was the only option. However, this can now be accomplished directly on your oscilloscope.

Some oscilloscope manufacturers have developed software packages for engineers who want to customise their compliance testing. These tend to be based on their existing compliance application framework, so engineers are familiar with the interface and layout. Typically, application development is organized into individual tests, which are then programmed using Standard Commands for Programmable Instruments (SCPI). Learning how to put together SCPI commands is fast and intuitive; every command corresponds to a physical action, such as a button press or dial turn. Each line corresponds to one command on the scope and oscilloscope vendors usually provide an easy to use programming guide, with example commands for all major scope actions to get you started.

With this type of scope centric automation, the development environment is usually not just a SCPI command window, rather a step by step guide through the different aspects of your automated test. Examples of other capabilities include setting up communication with and control of external equipment, defining test procedures using SCPI and configuring test conditions. You can automate the de-embedding of elements in your measurement circuit to recover critical design margin. This software may also allow the user to step quickly through each SCPI command in any given test to make sure the scope is performing as expected. This real time feedback contributes to the ease of developing automated tests.

When an application has been completed, it can be installed on a scope, where it runs in exactly the same way as a normal compliance application. The user can then select all or any specific tests they want the application to run and a report is generated automatically by the application. This report indicates pass/fail and margin versus user defined standards and often includes screen shots of waveforms to help with the correction of failed devices. Many applications can also call external pre and post processing scripts, such as MATLAB, LabVIEW, Python, C++ and Visual Basic for additional analysis of your acquired waveforms.

While these are powerful tools for developing scope centric automation on oscilloscopes, the software can also reduce significantly the effort required to integrate a scope into an existing external controller based automation environment. Rather than program all of the scope commands in C++, for example, the commands can be programmed in a scope centric environment using SCPI and called in a central C++ test script. This greatly reduces scope programming time and leverages the built in reporting capability, rather than having to format the results report manually.

Author profile:
Daniel Ruebusch is strategic marketing manager for high performance oscilloscopes with Agilent Technologies.