



TIPS FOR OPTIMIZING PERFORMANCE

Larry Salant, Author
February 4, 2005

Best practices for optimizing oscilloscope performance in ATE environments.

1. Always initiate the scope from a known point. The default panel ensures that all functions are turned off. Or use a saved panel that has been configured from the default panel.

Windows scopes:

use the *RCL or RCPN command, using a user-saved default file: rcpn disk, hdd,file,'d:\setups\default1.lss', or vbs 'app.SaveRecall.Setup.DoRecallDefaultPanel'

Non-Windows:

Use the *RCL or RCPN command only.

2. Set up math definitions ahead of time; turn them off when not needed. Turn all other math and parameters off so they are not performing the extra computations.

Windows scopes:

The scope computes any parameters and/or math on traces *that are turned on*. If using custom parameters, set un-needed parameters to "none," and turn off unnecessary Math functions.

Non-Windows:

Parameter off, "crms off," and each math trace defined as a "zoom." Also limit math points.

3. When using the scope in the ATE environment "How do I know when the scope has finished its process, so I can execute the next one?"

See LeCroy Application Brief "Remote Control Synchronization." Also Chapter 2 of the Remote Control Manual, "Timing and Synchronization."

4. Turning off the scope display to run faster is highly recommended.

Windows scopes:

When display is off, variations in the update rate are minimized (scope doesn't spend any time preparing and displaying the data). Use the DISP OFF command.

Legacy solution:

This makes the legacy scope's update rate much faster, and more consistent. Use the DISP OFF command.

5. Set scope to "Optimize analysis"

Windows scopes:

Customer program dependent; does not affect update rate when little or no analysis is being done; but can affect when customer requires intensive analysis.

Non-Windows:

Not available for these models.

Small effect if Display is "off"

6. How to manage a "Service request"

See #3 above and chapter 5 of the Remote Control Manual

7. How to know when the scope calibration is finished?

See #3 above

8. Reducing calibrations when changing Volts/div settings

Windows scopes:

At the start of your program, step through the sensitivity settings that will be used. If the sample rate is not changed, the scope will maintain (assuming the instrument is at ambient operating temperature) the calibration settings, and a calibration will not be induced when the V/Div is changed. [This is true only if you have not changed the channel combination, or changed sample rate between 20, 10, or 5 GS/s.

Non-Windows:

NA

9. Reducing calibrations when changing T/div settings

Windows scopes:

Under conditions where a change in the T/div causes the sample rate to change, the scope will induce a calibration. Set the sample rate to "Fixed" to eliminate long calibrations when switching between timebases, usually between 10 and 20 GS.

Non-Windows:

N/A

10. Minimizing waveform transfer time for multiple acquisitions

Windows scopes:

If you need to acquire and read out many short waveforms, use "Sequence mode" to acquire the individual waveforms—then download to the Host computer when the acquisition is complete.

Non-Windows:

Same

11. Avoid changing between 2 and 4 channel mode.

Windows scopes:

The scope will do an extensive re-calibration of the front end when entering and exiting the 2-channel mode. Set the sample rate to "Fixed."

Non-Windows:

Non Windows scopes cannot be changed. (Most can combine channels, but the cal's are not as significant.) They don't support fixed sample rate.

12. Will inactive math and parameters slow down the scope?

Same as question #2

13. Turn "auto calibration" off. This will prevent the scope from calibrating due to a temperature change.

Windows scopes:

Turn off "Auto-calibration": AUTO_CALIBRATE OFF

Non-Windows

Turn off Auto-calibration

Turning off Auto-calibration is not recommended. See #3 to know when Auto-calibration is completed.

14. Use single-trigger mode to eliminate timing issues.

Windows scopes:

Always program using the single trigger mode and have a looping structure for multiple acquisitions. When the auto or normal trigger mode is used for multiple acquisitions, timing inconsistencies sometimes

arise. This is covered in #3.

Non-Windows:

Same

15. Avoid repeated use of Auto-setup.

In situations where the signal is unknown, Auto-setup will find it, but it is slower than recalling the appropriate panel set up.

16. Each command/query sequence incurs OS and communications overhead, which can be significant. There are ways to minimize:

a. When doing multiple single acquisitions and reading parameter(s) on each trace, overhead time can be minimized by setting up the "Trend" function of the parameter(s) and then doing the acquisitions (preferably using sequence mode) *but do not read the parameter after each acquisition.* . . . Wait until all acquisitions are complete, then read the trend function and get all the parameter values in one read session, thereby significantly reducing the cumulative overhead of doing repetitive single parameter reads.

b. For applications where the need is to acquire multiple (short) waveforms and read out the data, use sequence mode to collect the multiple acquisitions. Then read out the sequence waveform instead of reading each individual waveform, thereby significantly reducing the cumulative overhead of doing repetitive individual waveform reads.

c. Corollary: if you can't use sequence mode because you need to change the acquisition settings, or change some external device setting (and need to ignore some triggers), you can use the "Sequence-Builder" function to collect the waveforms.

d. When possible, send compound messages (i.e., multiple commands separated by semi-colons) as described in chapter one of the Remote Control Manual.

e. Do processing in the scope. Transferring waveforms is very time consuming. It is generally much faster to have the scope compute measurements and only read out the end result. If custom processing is needed, use the XDEV capability.