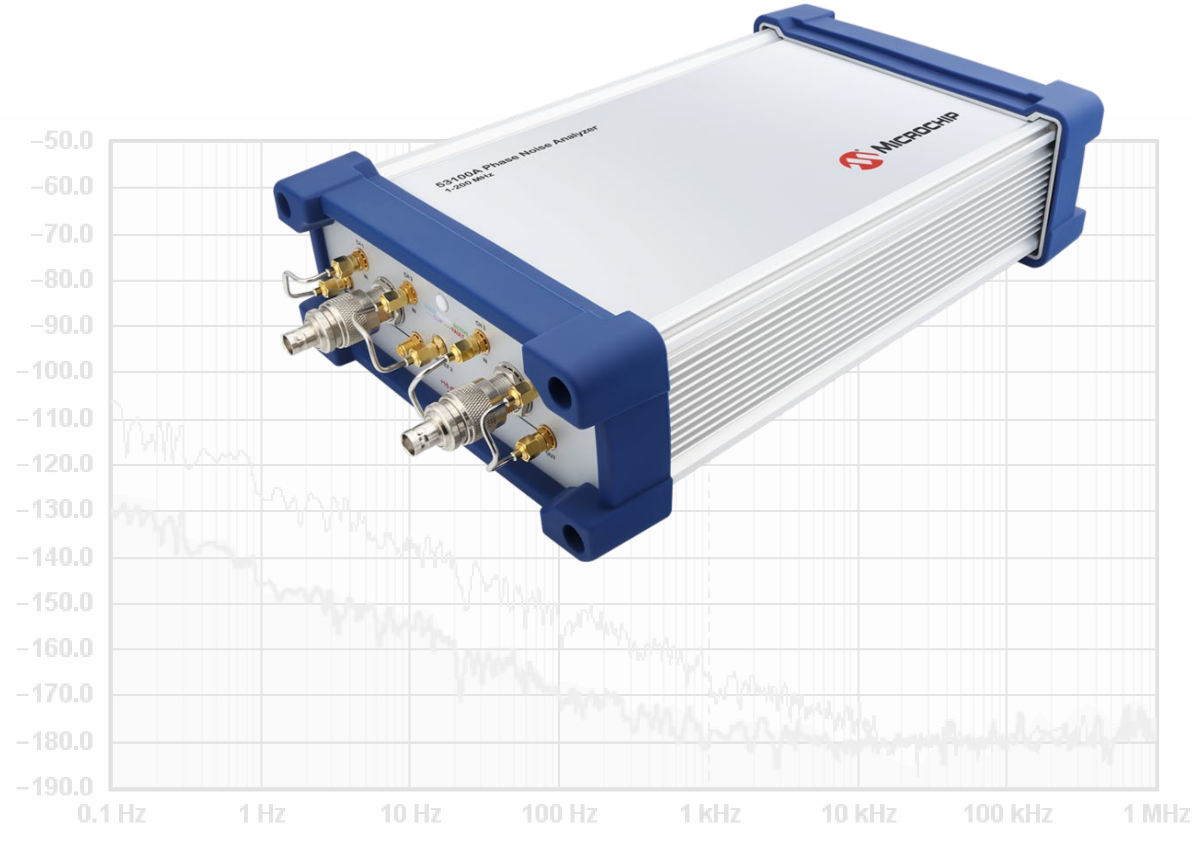
PhaseStation™

53100A Phase Noise Test Set

User's Manual

Revision 1.09

December 12, 2023



JACKSON LABS

Rev 1.00 (1-Oct-19)	Initial
Rev 1.01 (29-Apr-20)	Added reference to TSERVE remote access server
Rev 1.02 (30-Jan-21)	Updated for TimeLab beta 1.53
Rev 1.03 (3-Mar-21)	Updated for TimeLab beta 1.54
Rev 1.04 (25-Apr-22)	Updated for TimeLab beta 1.6
Rev 1.05 (3-Jul-22)	Added */timerec* parameter and *save* command descriptions to TSERVE section
Rev 1.06 (18-Jan-23)	Revised 53100A measurement role descriptions; added Option IR specifications. Other minor revisions for beta 1.7
Rev 1.07 (26-Feb-23)	Power consumption rating changed from <20W to <40W due to addition of Option IR
Rev 1.08 (12-Nov-23)	Updated absolute PN specifications for 100 MHz INT REF outputs Microsemi 3120A compatibility Other minor revisions for beta 1.72
Rev 1.09 (12-Dec-23)	Additional instructions for Option IR user calibration Added *Display->Frequency Y axis labels in Hz* and DisplayYAxisHz() descriptions

Table of Contents

Introduction 15

Specifications 17

Specifications (Option IR Dual 100 MHz Internal Reference) 19

Getting Started 21

What's in the box? 21

USB driver and software installation 21

Support 21

Installing TimeLab™ 23

Performance notes 24

Front panel features 25

Rear panel features 27

Guidelines for input signal connections 28

Choosing reference sources 28

Working with dual reference sources 30

Making your first measurements 33

Tips for new users 36

Next steps 37

Block diagram 39

A brief architectural note 41

Making Measurements 43

"What's all this ADEV stuff, anyhow?" 45

Allan Deviation (a) 46

Modified Allan Deviation (m) 47

Hadamard Deviation (h) 47

Time Deviation (t) 48

Maximum Time Interval Error (M) 48

Examining changes in stability over time 49

Common artifacts in ADEV and related measurements 50

Hints for xDEV measurements 55

Working with phase- and frequency-difference traces 57

Phase difference (Original) (w) 57

Phase difference (Unwrapped) (p) 57
Frequency difference (f) 58
How does TimeLab measure frequency? 59
Phase/frequency measurements with the 53100A 59
Examining traces in detail 60
Navigating zoomed graphs 61
Hints for phase/frequency stability measurements 63
Phase noise, AM noise, and jitter 65
Integrated noise and jitter measurement 67
The spur table 68
Show or hide known spurs 68
Spur measurement options 69
Is it a spur, or isn’t it? 70
Understanding instrument spurs 71
Manual spur removal 72
Using noise markers 73
Hints for noise measurements 74
TimeLab Command Reference 75
File menu 77
Load .TIM file (l) 77
Save image or .TIM file (s) 77
Copy image to clipboard (Ctrl-c) 78
Import .RES or .PNP phase noise data (N) 78
Import ASCII phase or frequency data (L) 78
Export ASCII phase data (x) 79
Export ASCII frequency data (X) 79
Export binary phase data 80
Export phase data to Stable32 (Ctrl-x) 80
Export ASCII xDEV trace 80
Export ASCII AM/PM noise trace 81
Export ASCII integrated PN data 81
Export AM/PM spur table 81
Exported ASCII files include headers for Microsemi TimeMonitor 82
Print image 82
Scale file dialogs by window size 82
Confirmation prompts default to Yes 82
Limit MTIE processing to 10K points 82
Warn before exiting with unsaved plots 82
Warn before retriggering measurement in progress 83

Reset all parameters, options, and settings at next startup 83
Close selected plot (Del) 83
Close all visible plots (Home) 83
Delete selected plot's .TIM file (Ctrl-Del) 83
Quit (q or Esc) 84
Edit Menu 85
Trace properties (e) 85
Flatten selected noise region (Ctrl-F8) 87
Restore flattened noise regions (Ctrl-F9) 88
Flatten selected or zoomed phase data (Ctrl-f) 88
Remove selected or zoomed phase data (F4) 89
Subtract global linear phase trend (frequency offset) (Ctrl-o) 90
Subtract global linear frequency trend (drift line) (Ctrl-l) 90
Subtract quadratic linear frequency trend (drift curve) (Ctrl-q) 90
Apply notch filter to phase data (Ctrl-n) 91
Undo last flatten or subtract operation (Ctrl-z) 92
Trace Menu 93
Phase/frequency traces begin at zero (z) 93
Show linear phase/frequency residual (r) 93
Show linear phase/frequency trend (Ctrl-t) 95
Phase/frequency Y axis unlocked in zoom mode (y) 95
Averaging window for phase/frequency traces (g) 96
Increase averaging window (Ctrl +) 96
Decrease averaging window (Ctrl -) 96
Draw xDEV traces with spline interpolation (i) 97
Show xDEV error bars (Ctrl-e) 98
Clip xDEV traces by noise bandwidth (Ctrl-b) 98
Clip xDEV traces by confidence (Ctrl-v) 98
Show separated xDEV variances (N-cornered hat) when available (Ctrl-h) 99
Show cross ADEV traces when available (Ctrl-j) 100
Show original traces in computed xDEV displays (F6) 100
Show correlation gain for selected noise trace (Ctrl-g) 100
Show FFT segment filter slopes (Ctrl-i) 101
Show imaginary part of cross spectrum (Ctrl-F3) 101
Show estimated instrument noise (F2) 101
Mark spurs in noise traces (Ctrl-m) 102
Suppress spurs in noise traces (Ctrl-s) 102
Smooth noise traces (Ctrl-w) 102
Show raw noise trace(s) (Ctrl-r) 103

Show AM noise in PN view (F8) 104
Tick marks (k) 104
Toggle trace thickness for current measurement (T) 105
Display Menu 106
Edit colors 106
Edit selected trace color (E) 106
Restore default colors 106
Restore selected trace color 106
High contrast (C) 107
Set spot cursor positions (j) 107
Show numeric tables (Ctrl-n) 107
Clear noise markers (Ctrl-F4) 108
Show integrated PN traces 108
Show cursor time (Ss) 108
Show cursor time (Hh:Mm:Ss) 108
Show cursor time/datestamp 108
Do not show cursor values 108
Phase/frequency X axis labels in Ss format 109
Phase/frequency X axis labels in Hh:Mm:Ss format 109
Phase/frequency X axis labels in Hh:Mm format 109
Phase/frequency X axis labels in Hh format 109
Frequency Y axis labels in Hz (Z) 109
Tau labels in scientific notation 109
Browse plots one at a time (b) 109
Overlay all loaded plots (o) 109
Toggle visibility of selected plot (v) 109
Select next plot in chart (+ or down arrow) 110
Select previous plot in chart (- or up arrow) 110
Move selected plot up (Ctrl-up arrow) 110
Move selected plot down (Ctrl-down arrow) 110
X zoom in ( ]) 110
X zoom out ([ ) 110
Y zoom in (}) 110
Y zoom out ({) 110
Decrease font size ((or Ctrl-mouse wheel) 111
Increase font size ( ) or Ctrl-mouse wheel) 111
Refresh (F5) 111
Legend Menu 112
Measurement Menu 117

Masks Menu 118
Clear mask selection 118
User-defined mask entries 119
Edit mask definitions 119
Scripts Menu 120
Creating a new test script 120
Examples 120
Run script … 121
Edit script … 121
Delete script … 121
Run last-executed script (Shift-Space) 122
Toggle script console for selected plot (F11) 122
Stop all running scripts (F12) 122
Acquire Menu 123
Jackson Labs PhaseStation / Microchip 53100A 123
Miles Design TimePod and Microsemi 3120A 123
Acquire from counter in Talk-Only mode 123
Acquire from live ASCII file 124
HP 53131A/53132A/53181A and 53220A/53230A 124
HP 5335A and 5370A/B 124
HP 5371A/5372A 124
Philips/Fluke PM6680 124
Picotest/Array U6200A series 124
Stanford Research SR620 124
Wavecrest DTS-2050/2070 series 124
Symmetricom 5115A / 5120A / 5125A (Frequency stability) 125
Symmetricom 5115A / 5120A / 5125A (Phase noise) 125
Stop/repeat acquisition (Space) 125
Abort and retrigger selected acquisition (Ctrl-a) 126
Keep and retrigger selected acquisition (Ctrl-k) 126
Acquisition options (Ctrl-F7) 127
Enable deferred acquisition (Ctrl-d) 128
Trigger deferred acquisition(s) (Enter) 128
Help Menu 129
User guide (F1) 129
About TimeLab 129
Debug mode 129
Check for updates (Ctrl-u) 129
PhaseStation 53100A Measurement Roles 131

Measure single DUT with single external reference ... 131
Measure single DUT with dual internal references ... 132
Measure single DUT with dual external references ... 132
Measure two DUTs with single external reference ... 133
Measure three DUTs with single external reference ... 133
Measure three independent DUTs (three-cornered hat) ... 135
Measure single DUT with cross ADEV statistics ... 136
Manual configuration ... 136
Cross Allan Deviation Measurements ... 139
Two-Port Residual Measurements ... 142
JavaScript API Function Reference ... 145
AcqChannel([Number user_value,] Number channel) ... 145
AcqCheckOptions(String option_name [{, String option_name}]) ... 145
AcqClearParams() ... 146
AcqFreqMHz([Number avg_secs]) ... 146
AcqFreqErr([Number avg_secs]) ... 146
AcqGetAMSpurTable(Array offset_array, Array amplitude_array) ... 146
AcqGetPMSpurTable(Array offset_array, Array amplitude_array) ... 146
AcqGetXDEVBins(String selection, Array tau_array, Array sigma_array) ... 146
AcqGetXDEVChart(Array tau_array, Array sigma_array) ... 146
AcqNumAcquiredPoints() ... 147
AcqNumChannels() ... 147
AcqParam(String key) ... 147
AcqStrParam(String key) ... 147
AcqNumParam(String key) ... 147
AcqReadAMNoiseTrace(Number offset_Hz) ... 148
AcqReadPMNoiseTrace(Number offset_Hz) ... 148
AcqReadAMNoiseFloor(Number offset_Hz) ... 148
AcqReadPMNoiseFloor(Number offset_Hz) ... 148
AcqSetParam(String key, String value) ... 148
AcqStartAcquisition(String menu_entry, Boolean skip_dialog [, Number user_value]) ... 149
AcqStopAcquisition() ... 150
AcqSetupEnd() ... 150
DisplayFontSize([Number points]) ... 150
DisplayNoiseMarkers([String marker_list]) ... 150
DisplayOverlayMode([Boolean overlay_mode]) ... 150
DisplayPlotVisibility(Number selection [, Boolean status]) ... 150
DisplayShowIntTraces([Boolean status]) ... 151
DisplayShowNumTab([Boolean status]) ... 151

DisplayRefresh([Number mask]) 151

DisplaySpotTau(Number secs) 151

DisplaySpotTime(Number secs) 151

DisplaySpotOffset(Number Hz) 151

DisplaySpotLowerOffset(Number Hz) 151

DisplaySpotUpperOffset(Number Hz) 151

DisplayYAxisHz([Boolean status]) 152

EventAcqDone([success][,user]) 152

EventAcqSetup([user]) 152

EventAcqTriggered([user]) 152

EventRunScript() 153

EventTimer() 153

FileCloseAllChannels() 153

FileCloseAllPlots() 154

FileClosePlot([Number selection]) 154

FileExecute[Wait](String path_to_file [, String params]) 154

FileExecute() 154

FileExitApplication([code]) 155

FileExists(String path, [String subdir,] String filename) 155

FileExportASCIIData(String descriptor, String path, [String subdir,] String filename) 155

FileImageURI() 157

FileLoad(String path, [String subdir,] String filename) 157

FileNumLoadedPlots() 157

FileNumUnsavedPlots() 157

FileSave(String path, [String subdir,] String filename) 157

FileSaveText(String path, [String subdir,] String filename, String contents) 158

LegendSelect([String field_name] [,] [Boolean status]) 158

MaskResultMargin([Number mask_num]) 158

MaskResultValid([Number mask_num]) 159

MaskSelect([String mask_name [, Number mask_num]) 159

MeasurementSelectView(String hotkey) 160

MeasurementDeferSelectView(String hotkey) 160

MeasurementView() 160

Print(…) 160

ScriptBindToPlot(Number selection) 161

ScriptBoundToPlot() 161

ScriptEnd() 161

ScriptFilename(Boolean full_path) 162

ScriptGetString(String heading, String message, Object user_text, String b1_text [, String b2_text [, String b3_txt]]) 162

ScriptHostFilename([bool full_path]) 163

ScriptLastStatusMessage() 163
ScriptMessageBox(String heading, String message, String b1_text [, String b2_text [, String b3_txt]]) 163
ScriptMinsToHMS(Number minutes) 164
ScriptNumParam(Number n) 164
ScriptNumVal(String format_string, Number value) 164
ScriptParamCount() 164
ScriptPlaySound(String filename) 165
ScriptSetWindowPos([String text]) 165
ScriptSetWindowTitle([String text]) 165
ScriptSetTTYHeader(Number linenum [, String text]) 165
ScriptShowTTY([Boolean show]) 166
ScriptStrParam(Number n) 166
ScriptUnbind() 166
SetupDetectedAmpldBfs 166
SetupDetectedValid 166
SetupDetectedFreqHz 166
SetupDetectSignals 166
SetupMeasuredAmpldBm 166
SetupMeasuredFreqHz 166
SetupMeasureSignals 166
SetupNumQuery 166
SetupSetDefaultAttenuation 166
SetupStrQuery 166
TimeHoursSinceTrigger() 167
TimeMinsSinceTrigger() 167
TimeSecsSinceTrigger() 167
TimeSetTimer(Number msec [, Boolean persist]) 167
TraceAvgWindow([Number seconds]) 167
TraceFreqZero([Boolean status]) 168
TraceMarkSpurs([Boolean status]) 168
TraceOverlayAMPM([Boolean status]) 168
TracePhaseFreqAutoY([Boolean status]) 168
TracePhaseFreqResid([Boolean status]) 168
TracePhaseFreqTrend([Boolean status]) 168
TracePhaseFreqZero([Boolean status]) 169
TracePhaseZero([Boolean status]) 169
TraceShowCorrGain([Boolean status]) 169
TraceShowCrossStatistics([Boolean status]) 169
TraceShowHatStatistics([Boolean status]) 169
TraceShowImag([Boolean status]) 169

TraceShowNoiseFloor([Boolean status]) 169
TraceShowOriginal([Boolean status]) 170
TraceShowRaw([Boolean status]) 170
TraceShowSlopes([Boolean status]) 170
TraceSmoothNoise([Boolean status]) 170
TraceSuppressSpurs([Boolean status]) 170
TraceThickness([Number status]) 170
TraceTickMarks([Boolean status]) 171
TraceXDEVBars([Boolean status]) 171
TraceXDEVClipBW([Boolean status]) 171
TraceXDEVClipConfidence([Boolean status]) 171
TraceXDEVSpline([Boolean status]) 171
Accessing the PhaseStation and TimePod Remotely 173
Introduction to STREAM 173
Launching STREAM 173
Using STREAM 175
/serial:<sernum> 175
/file:<filename> 175
/logfile:<filename> 176
/port:<portnum> 176
/msglvl:<0-5> 177
/format:<P, F, TSC> 177
/window:<samples> 178
/rate:<1, 10, 100, 1000> 180
/timestamp:<s, MJD> 180
/sep:<character> 180
/ref:<Hz> 181
/input:<Hz> 181
/autoupdate 182
/noexit 183
/notcp 183
/nowarnings 183
Introduction to TSERVE 184
Launching TSERVE 184
TSERVE command line options and defaults 185
/am:<true | false> 186
/ch:<chanexpr> 186
/chart:<secs> 187

/cmd_port:<port #> 187
/data_port:<port #> 187
/dateformat:<1,2,3> 188
/duration:<seconds> 188
/file:<filename> 188
/format:<P, F, TSC> 189
/inputfreq:<MHz> 189
/logfile:<filename> 190
/model:<name> 190
/msglvl:<0-7> 190
/noprompt 190
/notify 190
/overlap 191
/phasedec 191
/phaserate 191
/referencefreq:<MHz> 192
/roundfreq:<MHz> 192
/ sep:<character> 193
/serial:<sernum> 194
/smooth:<0,1,2> 195
/spur_dB:<dB> 195
/spur_Hz:<Hz> 195
/start 195
/timeformat:<12,24> 195
/timerec:<s> 196
/timestamp:<s, MJD, UNIX> 196
/title:<string> 196
/update 197
/wait 197
/warn 198
TSERVE remote commands 199
beep 200
close 200
exit 201
help [<command>] 201
history 201
logout 201
measurelinear 201
open <filename> 201
pause <chart> 202

prompt <on|off> ... 202
quit ... 202
removelinear <on|off> ... 202
reset ... 202
resume <chart> ... 203
save <filename.TIM> ... 203
selftest ... 204
set <. . .> ... 205
show <. . .> ... 206
shutdown ... 214
start ... 214
Compatibility notes ... 215
Measurement example: Phase data streaming ... 216
Measurement example: Phase noise plot acquisition ... 218
Appendix: 53100A Dual 100 MHz Reference (Option IR) ... 221
Front and rear panel layouts ... 221
Optional internal atomic standard (STD) ... 221
Block diagram ... 222
Why 100 MHz? ... 223
Software support ... 223
User and factory calibration controls ... 224
User calibration instructions ... 225
Firmware updates ... 226
Online help ... 226
Application examples ... 227
Phase noise of 100 MHz VCXO (1-minute and 30-minute runs) ... 227
Short-term stability of ultrastable 5 MHz DOCXO ... 229
Close-in phase noise of ultrastable 5 MHz DOCXO ... 232
Additional tips for high-performance measurements with Option IR ... 233
Allow sufficient stabilization time after applying power or changing loop parameters ... 233
Observe lock status before and during the measurement ... 233
Use high-grade cables to connect the DUT ... 234
Shield DUT from environmental effects ... 234
Appendix: Legal and regulatory notices ... 235
Federal Communications Commission Statement ... 236

EC Declaration of Conformity 236

Performance Certification and Validation 237

Warranty 237

Limitation of Warranty........ 237

Exclusive Remedies 237

Software License: TimeLab 238

Software License: FFTSS........ 238

Software License: FIDLIB........ 240

Software License: V8 JavaScript Engine 245

Introduction

The **53100A Phase Noise Test Set** measures the amplitude, phase and frequency stability of high-performance RF sources and two-port devices. Carrier frequencies from 1 MHz to 200 MHz are supported with direct front-panel access to all four measurement channels. The 53100A tells you everything you need to know about the stability characteristics of your devices, at timescales ranging from femtoseconds to days.

Measurements made by the 53100A include the following:

- Real-time ‘strip charts’ of phase and frequency differences at subpicosecond precision
- Absolute frequency counts at 13+ digits per second, 17 digits maximum
- Allan deviation (ADEV) typically less than 5E-15 at t=1s
- Modified Allan deviation (MDEV), Hadamard deviation (HDEV), and time deviation (TDEV)
- Fast realtime computation of Maximum Time Interval Error (MTIE)
- Phase noise and AM noise at offsets from 0.001 Hz to 1 MHz and levels below -175 dBc/Hz
- RMS-integrated time jitter with less than 50 fs residual jitter from 0.1 Hz to 1 MHz
- RMS-integrated phase noise, residual FM, and SSB carrier/noise ratio

Using high performance host-based DSP techniques on a Windows® PC, all of these measurements can be made simultaneously. Results appear as you watch – and you can save, view, compare, export, or print them at any time.   Accuracy and stability are inherited from a user-supplied external reference which can run at any frequency within the supported range, with no calibration required by the instrument itself.

Specifications

Input frequency and level (RF)	1 MHz – 200 MHz, 0 dBm - +15 dBm, 50 ohm N-F
Reference frequency and level	1 MHz – 200 MHz, 0 dBm - +15 dBm, 50 ohm N-F
Input/reference VSWR (<= 100 MHz)	1.5:1 or better
Input/reference port isolation (10 MHz)	130 dB or better
Maximum DC voltage at any input	+/- 24V
Allan deviation (1 MHz-200 MHz, t=1s)	7E-15 specified, 5E-15 typical (cross ADEV) 7E-14 specified, 5E-14 typical (standard)
Allan deviation (1 MHz-200 MHz, t=1000s)	2E-16 specified, 1E-16 typical (cross ADEV) 5E-16 specified, 2E-16 typical (standard)
Phase stability (5 MHz)	Less than 10 ps/hour after 2 hour warmup Typically below 2 ps/hour
Residual phase noise floor (5 MHz, 1 Hz)	-140 dBc/Hz specified, < -142 dBc/Hz typical
Residual phase noise floor (100 MHz, 1 Hz)	-120 dBc/Hz specified, < -122 dBc/Hz typical
Residual phase noise floor (5 MHz, 10 kHz)	-175 dBc/Hz specified, < -177 dBc/Hz typical
Residual phase noise floor (100 MHz, 10 kHz)	-170 dBc/Hz specified, < -172 dBc/Hz typical
Residual AM noise floor (5 MHz, 10 kHz)	-160 dBc/Hz specified, < -165 dBc/Hz typical
Spurious responses (5 MHz-100 MHz)	Less than -100 dBc (phase noise) or -90 dBc (AM noise) Typically below -120 dBc
Physical dimensions	380 mm x 230 mm x 100 mm, 3.2 kg 15” x 9” x 4”, 7 pounds
Power requirements	90-264 VAC, 47-63 Hz, < 40W 3-pole AC inlet IEC320-C14
Ambient temperature	15C to 35C operating, -20C to +50C storage 60F to 95F operating, 0F to 125F storage

Note: Due to the use of cross correlation to cancel instrument noise, performance depends strongly on measurement time and available signal levels. Specifications are based on measurements at +15 dBm after a two-hour warmup in a temperature-stable environment, with trace smoothing and overlapped acquisition enabled. Under these conditions, 5-10 MHz signals are typically measurable to better than -175 dBc/Hz at offsets > 10 kHz after less than 5 minutes. Lower-amplitude signals, signals near the frequency-coverage limits, and measurements of very low close-in noise may require more time to converge.

Specifications (Option IR Dual 100 MHz Internal Reference)

Output frequency (Both channels)	100 MHz sinewave
Output power (Both channels)	+13 ± 2 dBm into 50 ohms
Channel isolation (100 MHz)	>100 dB specified, >110 dB typical
Output match	VSWR < 1.2:1
Spurious outputs within 1 MHz of carrier	< -100 dBc specified, < -110 dBc typical
Spurious outputs (non-harmonic, DC-1 GHz)	< -80 dB typical
Harmonic outputs	< -40 dBc typical at 200 MHz < -70 dBc typical at >200 MHz
Operating modes (INT REF 1)	Free run Lock to optional internal atomic standard (STD) Lock to external 10 MHz at REFERENCE IN jack Lock to external 1PPS at rear-panel 1PPS IN jack
Operating modes (INT REF 2)	Free run Lock to INT REF 1
Lock time constant	0.1s to 10000s, both channels independently adjustable
Frequency error (Base)	Initial: < ±1.00E-8
	Aging: < 5E-10/day after 30 days, < 5E-8/year
Frequency error (Option STD)	Initial: < ±1.00E-10
	Aging: < ±1.00E-11/day after 90 days, < 5E-9/year
Allan deviation (Free-running or option STD)	< 1E-12 @ t=1s, < 8E-13 typical
	< 4E-12 @ t=10s
	< 5E-12 @ t=100s
Allan deviation (Option STD only)	< 5E-12 @ t=1000s

Phase noise (100 MHz)	< -89 dBc/Hz @ 1 Hz, < -91 typical
	< -130 dBc/Hz @ 100 Hz, < -135 typical
	< -165 dBc/Hz @ 100 kHz, < -169 typical
Typical Allan deviation measurement floor	< 2E-13 @ t=1s
	< 5E-13 @ t=10s
	< 5E-12 @ t=100s
	< 5E-12 @ t=1000s (Option STD only)
PN measurement floor (10 MHz, 1 Hz)	< -120 dBc/Hz specified, < -125 dBc/Hz typical
PN measurement floor (10 MHz, > 10 kHz)	< -170 dBc/Hz specified, < -175 dBc/Hz typical
PN measurement floor (100 MHz, 1 Hz)	< -100 dBc/Hz specified, < -110 dBc/Hz typical
PN measurement floor (100 MHz, > 10 kHz)	< -170 dBc/Hz specified, < -175 dBc/Hz typical
Ambient temperature	15C to 35C operating, -20C to +50C storage 60F to 95F operating, 0F to 125F storage

Notes:

1) Stability specifications with Option STD atomic standard assume t >= 200s loop time constant
2) All specifications apply after 24-hour stabilization in laboratory environment
3) Measurements in disciplined mode inherit accuracy and stability of external reference. Typical residual stability approximately 2E-13/tau (INT REF 1 output, t=1s, 10 MHz external reference with loop time constant=0.1s)
4) All internal sources support both factory- and user-level calibration

Getting Started

What’s in the box?

Please check the contents of your package carefully upon arrival. Each PhaseStation 53100A unit from Jackson Labs Technologies should be accompanied by the following items:

- (1) USB 2.0 cable, A Male / B Male
- (1) Power supply
- (2) N-M / BNC-F coax adapters
- (4) 1” (25.4 mm) SMA-M / SMA-M coax jumpers (preinstalled)

Additionally, each PhaseStation 53100A delivered to North American customers includes a standard IEC320 / NEMA 5-15P power cord for 120V service. For operation in other countries, the 53100A’s power supply accepts 50/60 Hz AC power at all standard line voltages from 100V to 240V, and is compatible with IEC320-C14 power cords available locally.

USB driver and software installation

The current release build of TimeLab may be downloaded at https://www.miles.io/timelab/mchp/setup.exe, while the latest beta version containing bug fixes and feature additions is posted at https://www.miles.io/timelab/beta.htm. **Please ensure you are running an up-to-date build before using your new 53100A.**

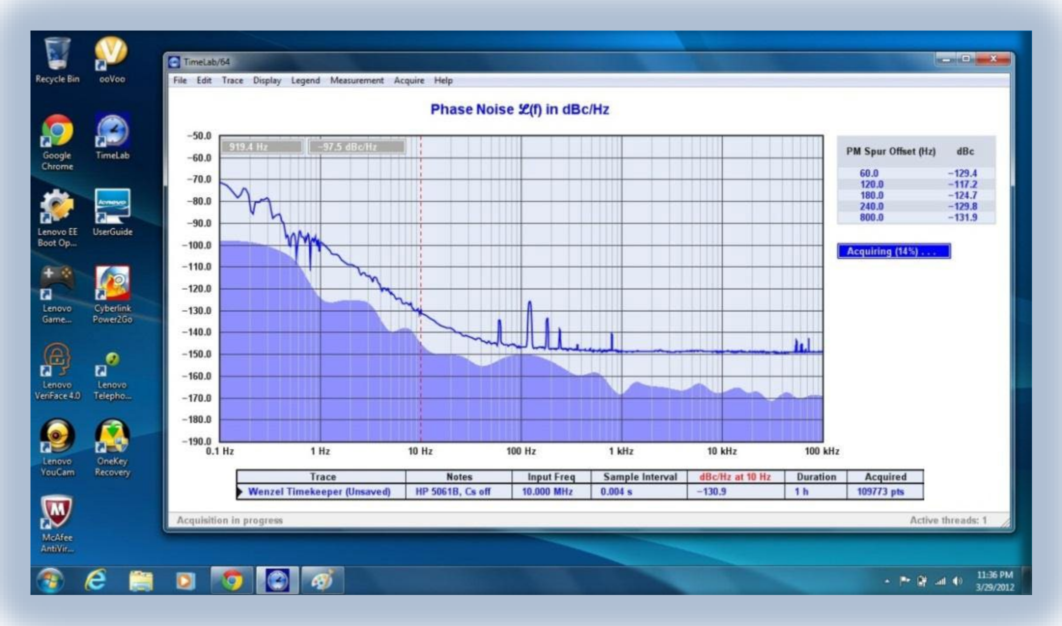
The 53100A user manual also serves as the primary software documentation for TimeLab. The latest version of the 53100A user manual is available at http://www.miles.io/PhaseStation_53100A_user_manual.pdf.

Support

For prompt assistance with software issues, contact Miles Design LLC at john@miles.io .

For hardware and warranty-support issues, please contact Jackson Labs Technologies, Inc. at http://www.jackson-labs.com/index.php/support, or the distributor from whom you purchased the instrument.

Installing TimeLab™



PM Spur Offset (Hz)	dBc
60.0	−129.4
120.0	−117.2
180.0	−124.7
240.0	−129.8
800.0	−131.9

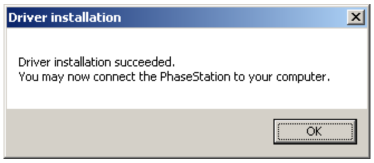
Trace	Notes	Input Freq	Sample Interval	dBc/Hz at 10 Hz	Duration	Acquired
Wenzel Timekeeper (Unsaved)	HP 5061B, Cs off	10.000 MHz	0.004 s	−130.9	1 h	109773 pts

TimeLab is a compact Windows application designed for general-purpose phase/frequency stability and noise measurement. It runs on Intel® or AMD® x86-based PCs equipped with Microsoft Windows® 7 SP1 or later versions. TimeLab supports a wide variety of time and frequency measurement instrumentation in addition to the PhaseStation, including the TimePod 5330A cross-spectrum analyzer and the 5115A/5120A/5125A/53100A phase noise test sets from Microchip.

Minimum system requirements are 100 MB of disk space, 1 GB RAM and a CPU with SSE2 support. For use with 53100A instruments, **a processor with four or more physical cores is recommended**.

By default, TimeLab will automatically check the Miles Design web site on a weekly basis and inform you if a newer version is available for download. Updates are always free of charge. To configure or disable automatic update notifications, select **Help→Check for updates**.

For best results, TimeLab should be installed prior to connecting the 53100A to your PC's USB port for the first time. Before exiting from the install program, make sure that the box labeled **Install USB Driver** is checked. This will help Windows locate the USB driver without further intervention when the 53100A is connected. After the driver has been successfully installed on your system, a confirmation message will appear.

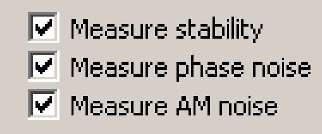

Driver installation

Driver installation succeeded.
You may now connect the PhaseStation to your computer.

Performance notes

Measurements made with the 53100A rely on multithreaded signal processing operations that can place significant performance demands on the host PC. Referring to the benchmarks at http://www.cpubenchmark.net/common_cpus.html, the minimum PassMark score for reliable acquisition falls in the 1,600 to 2,000 range.   In practice, this means that a 2.0 GHz Intel Core 2 Duo or faster processor is required. Additional CPU cores are often more helpful than higher clock speeds.

Use of a system with inadequate CPU performance may result in acquisition errors, often accompanied by a flashing red fault indication on the 53100A's status LED. **It may be necessary to disable one or more measurement types to achieve reliable operation in such cases.** For example, if you are interested only in phase/frequency stability measurement, uncheck the **Measure phase noise** and **Measure AM noise** boxes in the acquisition dialog, as shown at right. By default, the 53100A will measure phase noise, AM noise, and phase/frequency stability simultaneously.



☑ Measure stability
☑ Measure phase noise
☑ Measure AM noise

Figure 1: Available measurement types

It is strongly recommended to select the "High performance" power plan in the Windows control panel, particularly when the PC is running on batteries. The **System power** control on the **Additional options** page of the 53100A acquisition dialog provides a convenient way to switch to the high-performance power plan during acquisition. When a given power plan is selected with **System power**, the power plan specified in the Windows control panel will be restored automatically when the measurement ends. Using the **System power** control to select the power plan for use during measurements also guards against modification or reversion of the power plan selection by the operating system or any other applications.

When noise measurements on slower PCs are required, another strategy to improve reliability is to uncheck the **Overlapped acquisition** box in the **Additional options** page of the acquisition dialog. Doing so will increase the averaging time required to achieve extremely low noise floors, but will have little effect on most measurements. **Note that overlapped acquisition is turned off by default**.

For improved performance, a 64-bit version of TimeLab will automatically be installed on x64-based systems. If you need to record long phase records with the 53100A (or any other equipment supported by TimeLab), an x64 system with several GB of RAM is recommended.

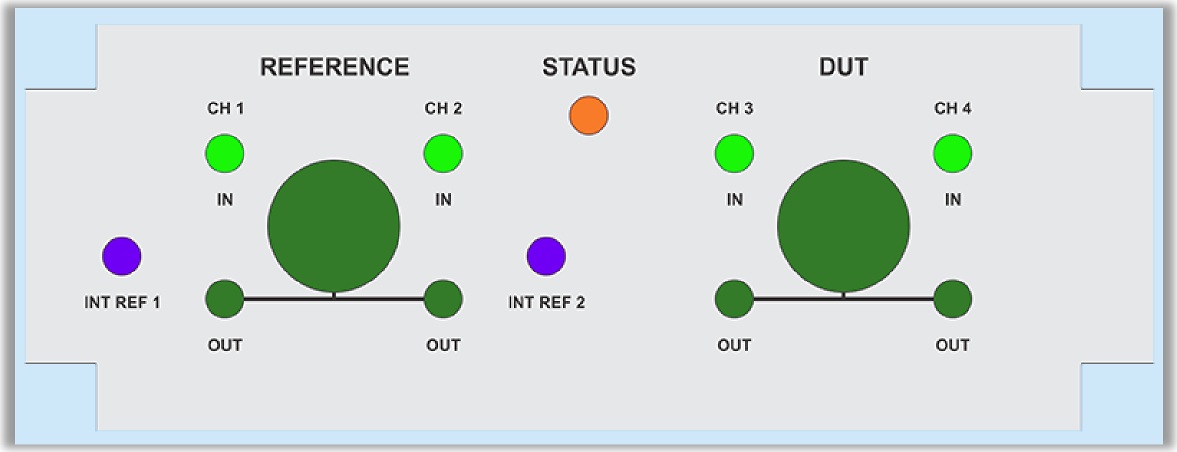


Figure 2: Front panel controls and features

Front panel features

The 53100A's front panel includes the following features, located as shown in the color-keyed image above:

- Two AC-coupled **N-F input jacks** that correspond to the primary REFERENCE and DUT signal inputs. Measurements of RF signals from a single DUT and a single reference source will normally use these two jacks as inputs.

- Each of the N-F input jacks is connected to an internal power splitter with two output ports, which in turn are routed to a pair of **SMA-F output jacks** below.

  It's important to note that the jacks associated with the power splitters are unconnected to anything else internally. They are provided for operational convenience. Each splitter presents a nominal 50-ohm load at its input port[1], where the signal is fed to two independent, identical amplifiers whose outputs are also source-terminated at 50 ohms. Net gain from each N input jack to its corresponding SMA output jacks is 0 dB nominal. These active splitters offer advantages over traditional resistive or Wilkinson implementations in cross-correlated noise measurement applications, including high isolation and well-defined port impedances with minimal differential-mode noise.

- Four AC-coupled SMA-F input jacks that drive the individual ADC channels.

[1] Specifically, both the **DUT and REFERENCE 'N'** jacks are coupled to their respective input splitters via 'L' networks consisting of 6.8 nF 100V capacitors and 22 uH inductors. The four SMA input jacks are capacitively coupled as well.

By default, each of these input jacks is strapped to the **output jack** directly beneath it. This effectively divides the four-input 53100A into two independent two-channel “instruments” for cross-correlated phase noise and AM noise measurements. By sending identical copies of the reference signal to channels 1 and 2 and the DUT signal to channels 3 and 4, uncorrelated noise in all four channels will average out over time. In principle, only the noise at the splitter input terminations is common to the two “instruments,” corresponding to -177 dBc/Hz with a 0-dBm input.

Note that the above describes only one of several possible measurement configurations.
A number of other measurement roles, including multichannel stability measurements, are also possible. These are discussed in the *PhaseStation 53100A Measurement Roles* section on page 131.

- Units equipped with the dual internal reference option are provided with two **SMA-F output jacks** for these 100 MHz sinewave reference signals. Typical output level is +12 dBm.

- The **STATUS** indicator is a high-brightness RGB LED that reveals the instrument status at a glance:

Blue Present for the first few minutes while the 53100A’s internal clock oscillator warms up.

White The 53100A is ready to acquire data. Stability measurements at 50 Hz and higher bandwidths may be performed as soon as the LED turns from blue to white, although a 30-minute warmup period is recommended. Input/reference drift warnings may occur with shorter warmup periods.

Green A measurement is either pending or in progress. No fatal errors have occurred.

Flashing purple Excessive power is present at either the **REFERENCE** or **DUT** inputs or at one or more of the channel input jacks, causing signal clipping in one or more ADC channels. An unexpected increase in signal level after measurement has begun can cause this indication. The acquired data is questionable.

While the inputs are protected against DC potentials up to 24 volts, the maximum recommended RF power at any front-panel input jack is +15 dBm. Damage may occur at RF input levels greater than +20 dBm.

Flashing red A USB data overrun has occurred. This is a fatal error. Insufficient CPU power (or disconnection of the USB cable during a measurement) is typically responsible. You may be able to run the measurement successfully if you uncheck one or more of the noise or stability measurement options in the acquisition dialog.

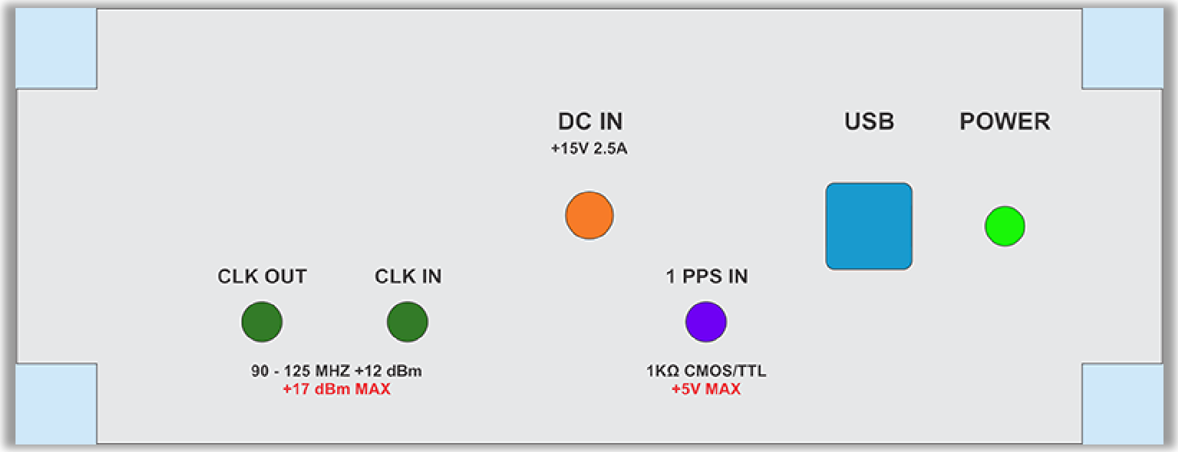


Figure 3: Rear panel controls and features

Rear panel features

The 53100A's rear panel includes the following features, located as shown in the color-keyed image above:

- The 2.1-mm **DC input jack** is intended for use with the provided 15V external power supply. Note that the 53100A's power supply is equipped with a locking power connector. To prevent inadvertent disconnection, it is strongly suggested that you secure the locking ring to the threaded portion of the input jack when connecting the power supply. Use the **power switch** to turn the 53100A on and off, rather than repeatedly connecting and disconnecting the barrel connector.

- Two **SMA-F jack locations** are reserved for future use.

- One **SMA-F input jack** for use with the optional dual-channel internal reference module (Option IR).

- A **USB Series 'B' receptacle** is provided for PC connectivity. Guidelines for PC selection appear in the *Performance Notes* section on page 24.

USB 2.0 High Speed support is required for all 53100A acquisitions. In most cases the use of passive or active USB hubs with the 53100A is acceptable, but if connection problems occur, you may find it helpful to connect the supplied USB cable directly to the host PC, or try a different USB port.

Guidelines for input signal connections

Although the N-F and SMA-F input jacks offer superior mechanical stability, BNC-N coax adapters are also supplied with the 53100A for operator convenience. Regardless of your choice of coax fittings, double-shielded cables such as RG223 or RG400 are recommended for low-level measurements that may be affected by crosstalk and environmental interference. **Use of RG58, RG174 and other common single-shielded cables can cause artifacts in stability and noise plots**, particularly with longer cables.

Input signals should be greater than 0 dBm for best performance -- +5 to +15 dBm is recommended. While the 53100A is intended for use with sinewave oscillators, it's possible to measure logic-level clocks as well if you equip them with suitable matching networks[2]. It's a good practice to measure an oscillator with a load that's similar to what it will encounter in its ultimate application, keeping in mind that most logic-level sources are not specified to drive 50-ohm loads at all. Overloading of the source, the 53100A, or both can occur if these devices are connected directly to the instrument's input jacks. Consult the manufacturer's data for the source in question to determine its output impedance and/or current drive capability.

The required matching network can be as simple as a resistive series termination or voltage divider, but the 53100A's own spur performance and broadband noise floor will likely be degraded when measuring non-sinusoidal sources unless you provide additional bandpass filtering. Careful shielding of matching networks, buffers, and other interfacing elements is also necessary, since isolation of DUT and reference signals from stray RF signals (and from each other) is a prerequisite for clean measurements. Experimentation will likely be required to achieve the desired results with non-sinusoidal signals.

Choosing reference sources

In addition to the host PC and an appropriate power source, measurements made with the 53100A require either one reference signal to be supplied at the REFERENCE jack or two reference signals supplied at the CH 1 IN and CH 2 IN jacks. **Virtually all aspects of measurement performance – accuracy, repeatability, noise floor, spurious responses – depend on your ability to provide the best reference signal(s) possible.**

But what does the "best reference possible" mean? It depends on the goals of your measurement. Very few reference sources are ideal for both short- and long-term measurements. Typical "house clocks" that distribute 5 or 10 MHz signals at levels between +5 and +15 dBm often work well for ADEV and other time/frequency measurements, but atomic and GPS standards that are often used as sources for centralized clock distribution may exhibit more short-term phase noise than an undisciplined crystal oscillator. Distribution amplifiers can also add substantial noise of their own, as well as phase drift at longer timescales due to temperature sensitivity.

[2] See http://www.wenzel.com/documents/waveform.html for some suggested approaches.

Furthermore, even when using the best standards, distribution amps and double-shielded cables, it's almost impossible to build a large clock distribution network that's free of environmental spurs. As a result, you should consider using a standalone low-noise OCXO – or even better, two of them -- for phase noise and AM noise measurements.

Unlike most other stability and noise analyzers, the 53100A can work with any reference whose frequency lies within its specified RF carrier range (1-200 MHz), regardless of the input frequency from the device under test (DUT). **Frequency readings and phase noise levels are always referred to the frequency at the INPUT jack.**

For example, if your 10 MHz DUT's phase noise when measured with a 5 MHz reference is -160 dBc/Hz at 10 kHz and its Allan deviation is 3E-12 at t=10s, you will still measure -160 dBc/Hz and ADEV=3E-12 if you switch to a 10 MHz reference, or one at any other frequency.

One complication, however, is the need to handicap the reference's phase noise and FM/PM spur amplitudes by 20*log10(F_{DUT}/F_{REF}) dB when the DUT and reference frequencies differ. This is the same effect that classically applies when a signal undergoes frequency multiplication or division. The effective phase noise and spur performance of the reference source would be 6 dB worse in the example above where a 5 MHz reference is used to characterize a 10 MHz device. Likewise, the reference's long-term stability characteristics (e.g., Allan deviation) would be degraded by a factor of 2. Ideally, the phase noise of your reference after any applicable 20*log10(F_{DUT}/F_{REF}) correction should be at least 8-10 dB lower than the expected performance of the DUT at the offsets of interest. The reference's phase noise will not affect the measurement to any great extent as long as this margin is maintained.[3]

Bear in mind that lower-frequency crystal oscillators will usually exhibit better close-in noise than those at higher frequencies, even when the 20*log(N) rule is taken into consideration. This is a consequence of the physical properties of the crystals themselves. For example, the highest-quality 5 and 10 MHz OCXOs on the market can perform at levels below -130 dBc/Hz at 1 Hz from the carrier. These oscillators make superior references for HF frequency stability and close-in noise measurement, easily outperforming the best VHF oscillators. But at the same time, they will be less effective than low-noise VHF oscillators as references for broadband noise measurement of VHF sources. Low-frequency reference oscillators have no inherent advantages at offsets beyond a few kHz where crystal Q no longer dominates, and the 20*log(N) principle will work against them when measuring higher-frequency DUTs.

An additional consideration when specifying reference oscillators is that industry-standard frequencies such as 5, 10, and 100 MHz benefit from a larger selection of crystals at the factory. Within a given product line, oscillators at unusual frequencies will rarely outperform units manufactured at standard frequencies.

[3] This being said, it's sometimes useful to measure the phase noise or stability of a DUT by using an identical device as the reference. In such a case, where the phase noise of the reference and DUT is presumed to be identical but uncorrelated, the resulting PN graph will be 3 dB too high, while Allan deviation and related measurements will be artificially elevated by a factor of sqrt(2). You can use the **Rescale Phase** field in TimeLab's **Edit→Trace properties** dialog to correct the ADEV of two identical devices by scaling the phase data by 0.707. Similarly, the phase noise may be corrected by entering -3 in the **PN Gain** field to lower the trace by 3 dB.

Working with dual reference sources

The use of dual references carries significant advantages in noise measurement due to the instrument's use of cross spectrum averaging to remove the influence of uncorrelated reference sources over time. Multiple references can be helpful in stability measurement as well, since the 53100A can easily acquire the multichannel data necessary for so-called "three-cornered hat" processing.

For a detailed "case study" featuring the use of dual reference sources in both stability and noise measurement roles, please refer to Application Note 53100A-2, *Dual Reference Noise and Stability Measurements with the PhaseStation™ 53100A*. Also, please see the *53100A Dual 100 MHz Reference (Option IR)* section on page 221 – it contains helpful information and guidelines that apply to all dual-reference configurations, not just the factory-installed internal reference option module.

The use of dual independent references relaxes some of the constraints on the stability and/or noise performance of the reference signal sources, but the advantages come with a new requirement: **the references must be truly independent and uncorrelated**. Otherwise, cross-correlated phase noise and AM noise measurements will not converge at the DUT's true noise level, and three-cornered hat measurements will not benefit from the statistical independence required to separate the frequency variances at the individual channels.

Two OCXOs running from the same power supply, for instance, may not be as independent as they look. They may even injection-lock to each other through their power supply connections if decoupling isn't adequate. When working with dual references, consider enclosing each of them in its own RF-tight box or other sealed metal container. Use high-quality DC feedthrough capacitors rather than conventional power jacks or connectors. These precautions are always a good idea, but they're absolutely mandatory with OCXOs whose output signals are obtained from discrete pins or SMT pads rather than coaxial outputs.

53100A instruments equipped with optional dual internal references are particularly easy to set up since reference channel isolation isn't a concern. There are no inherent performance differences associated with the choice of external versus internal reference pairs, but the internal reference option offers the practical advantage of allowing the 53100A to measure high-performance sources with no connections other than USB, power, and the DUT itself.

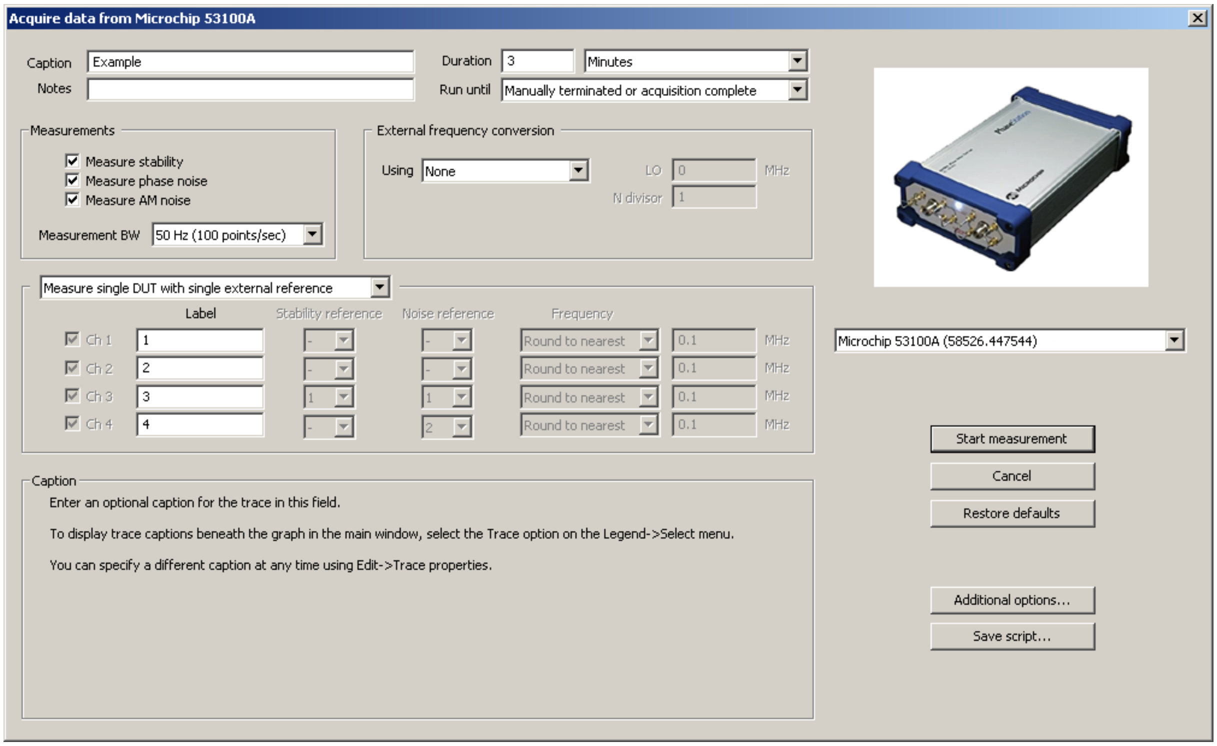
Internal and external reference jack configurations are shown in the *PhaseStation 53100A Measurement Roles* section (page 131.) When reviewing the dual-reference measurement role descriptions, note that the REFERENCE, CH 1 OUT, and CH 2 OUT jacks remain unconnected when dual references are used, whether internal or external, except when a 10 MHz external standard is connected to the front-panel REFERENCE jack to discipline the Option IR internal reference sources. As with the DUT, CH 3 OUT, and CH 4 OUT connectors on the right side of the front panel, these jacks correspond to the input and output ports of an internal RF splitter. The REFERENCE and DUT channel splitters are provided for user convenience, and have no other internal connections except as noted above.

Dual-channel references used with the 53100A do not have to operate at the same frequency. That being said, when working with external dual-channel sources it can sometimes be helpful to either phase-lock them together with a loop bandwidth well below the narrowest noise offset of interest, or separate them by several dozen hertz or more. If the oscillators are only a few hertz apart, it may be difficult to spot crosstalk problems in the external RF or power connections.

Finally, note that basic single-reference frequency stability measurements don't use the second (CH 2) reference or the second DUT input (CH 4) at all. The CH 2 reference may be powered down or left unconnected if noise measurements are disabled in the acquisition dialog and no DUT channels are using CH 2 as a stability reference. Stability measurements that demand the ultimate in Allan deviation performance can take advantage of the CH 2 and CH 4 inputs through the use of measurement roles that support cross Allan deviation, as described on page 131.

Making your first measurements

To paraphrase Kernighan and Ritchie[4], TimeLab is not a large program, and it's not well served by a large manual. Likewise, the 53100A hardware is easy to work with after some basic measurement concepts are understood. As an introductory exercise – or a quick operational check – you may wish to perform a measurement using the default acquisition parameters in TimeLab.


Acquire data from Microchip 53100A

Caption: Example
Notes:
Duration: 3 Minutes
Run until: Manually terminated or acquisition complete

Measurements
☑ Measure stability
☑ Measure phase noise
☑ Measure AM noise
Measurement BW: 50 Hz (100 points/sec)

External frequency conversion
Using: None
LO: 0 MHz
N divisor: 1

Measure single DUT with single external reference

	Label	Stability reference	Noise reference	Frequency		
☑ Ch 1	1	-	-	Round to nearest	0.1	MHz
☑ Ch 2	2	-	-	Round to nearest	0.1	MHz
☑ Ch 3	3	1	1	Round to nearest	0.1	MHz
☑ Ch 4	4	-	2	Round to nearest	0.1	MHz

Microchip 53100A (58526.447544)

Caption
Enter an optional caption for the trace in this field.
To display trace captions beneath the graph in the main window, select the Trace option on the Legend->Select menu.
You can specify a different caption at any time using Edit->Trace properties.

Figure 4: Acquisition dialog for all 53100A instruments, including PhaseStation and Microchip-labeled versions

Getting started with the PhaseStation is as easy as connecting the 53100A's USB cable to your PC and launching TimeLab. Once TimeLab is running, select the **Acquire→Jackson Labs PhaseStation** menu option. (This option may be named **Acquire→Microchip 53100A** instead, depending on the version of TimeLab installed on your PC.) The dialog box that pops up should contain a list of available device(s) in a dropdown box at upper right, just below the instrument photo.

Assuming the appropriate 53100A instrument is selected, the acquisition parameters are all valid, you've connected stable reference and DUT input sources to the 53100A, and you've allowed at least a few minutes' warmup time, all you need to do is press the **Start measurement** button. The acquisition dialog will disappear and the **STATUS** indicator on the 53100A should change from white to green. Over the next ten seconds or so, various informational messages will appear in the TimeLab status bar at the

[4] *The C Programming Language*, Second Edition (1988), Bell Telephone Laboratories

bottom of the program's main window as the software characterizes the device under test (DUT) and reference input signals.

If all is well, the status bar message should soon change to "Acquisition in progress." Measurement data will now begin to appear in graphical form. Unless you've selected a different view from the **Measurement** menu, you will see an Allan deviation (ADEV) plot for your DUT, rendered in real time at 100 points per second.

Unless you changed the **Duration** value, the measurement will run for three minutes. This will be long enough to allow the ADEV trace to reach τ=one minute, and for the phase noise trace to reach 1 Hz. You can try any of these suggested actions at any time, either during the measurement or after it finishes:

- Left-click on the ADEV graph or any other log-log plot. Observe that the red *spot cursor* can be placed at any vertical column. The graph's Y value at the spot cursor will be displayed in the *legend table* beneath the graph area, updated in real time to reflect the latest reading. At the same time, ADEV values at assorted tau periods should be displayed in a chart on the right side of the graph.

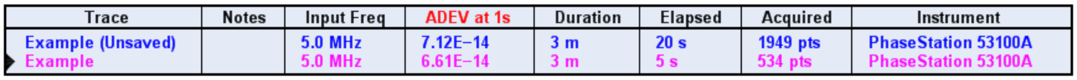
- Left-click and drag a box anywhere on the plot. When you release the left mouse button, the view should zoom in to magnify the specified area. You can drag an edge (or a corner) of the box slightly outside of the graph area to expand the plot in that direction. Right-click to return to the unzoomed view.

- Try selecting a few of the different graph types in the **Measurement** menu. Assuming you didn't uncheck any of the measurement options in the acquisition dialog, all of the different graph types should be accessible, updating continuously in a "live" fashion as long as the measurement is still in progress.

- Now is a good time to start picking up the keyboard shortcuts. Instead of selecting **Measurement→Frequency difference**, try the **f** key. To return to the Allan deviation plot, use **a**. Uppercase **A** selects AM noise, while uppercase **P** selects phase noise. (In some cases, the TimeLab features and commands described in this manual will be referenced by their hotkeys rather than their menu entry names, after the menu entries have been introduced.)

- Save the measurement data to a .TIM file with **File→Save image or .TIM file (s)**. A .TIM file is an ASCII text file that contains the data needed to recreate all of the available graphs associated with a given measurement. This data includes the entire phase record, as well as the FFT bins from the phase noise and AM noise plots. You don't have to wait for the measurement to end; at any time during a measurement, you can save the available data to a .TIM file or load another .TIM file for display alongside the existing plot(s) that have been loaded or acquired.

- Use **File→Load .TIM file** to read the .TIM file you just saved. Now you should have two copies of the same plot in memory. At the left edge of the legend table entry for the most recently-initiated or loaded plot, you'll see a black triangular cursor, indicating the plot that's currently selected for editing, browsing, or other forms of control interaction in TimeLab. In the table below, the most-recently loaded plot has been automatically selected:



Trace	Notes	Input Freq	ADEV at 1s	Duration	Elapsed	Acquired	Instrument
Example (Unsaved)		5.0 MHz	7.12E−14	3 m	20 s	1949 pts	PhaseStation 53100A
▶ Example		5.0 MHz	6.61E−14	3 m	5 s	534 pts	PhaseStation 53100A

Figure 5: Selection cursor

If the measurement is still in progress, you can left-click on the original plot's row in the legend table or press the **Up arrow** key to re-select it, then switch to the **p)hase** or **f)requency difference** views to watch its phase record continue to grow while the saved copy remains unaffected.

For "power users," three of the most common commands in TimeLab are **Display→Browse plots one at a time (b)**, **Display→Overlay all loaded plots (o)**, and **Display→Toggle visibility of selected plot (v)**. Like many other commands, the **b**, **o**, and **v** commands operate on the so-called "selected" plot as described above. If you've followed along with the suggested command demonstrations above, then you should have two (or more) plots loaded, and you can experiment with these concepts.

- First, select any desired **Measurement** view and press the **b** key to enter Browse mode. Observe that only one of your plots is now visible.

- Furthermore, notice that two things happen when you press the **Up-** or **Down-arrow** keys (**Display→Select next plot in chart** / **Display→Select previous plot in chart**). The small black triangular cursor at the leftmost edge of the legend table moves up and down, and the plot that it points to becomes visible. (You can also select a new plot by left-clicking on its row in the table.) In **Display→Browse** mode, entries in the legend table corresponding to all of the unselected plot(s) are grayed out, and those plots aren't shown in the graph.

- You can put things back to normal by pressing **o** to return to **Display→Overlay** mode. The idea of a "selected" plot still exists in **Display→Overlay** mode, but it no longer determines which plots are visible in the legend table and graph areas. In **Display→Overlay** mode, all of the loaded plots are visible, all of the time, unless individually hidden with the **Display→Toggle visibility of selected plot (v)** function.

- To avoid confusion, the **Display→Toggle visibility of selected plot (v)** function doesn't do anything in **Display→Browse** mode. But in **Display→Overlay** mode it does just what it sounds like – it toggles the visibility of the selected plot in both the graph area and the legend table. This panoply of options may seem confusing at first, but it's nothing compared to the jumble of line segments and shaded areas that can appear on the graph when all nine available slots in the legend table are populated with visible plots!

- Regardless of the choice of **Display→Browse** or **Display→Overlay** mode or the visibility status of any given plot, the "selected plot" concept is still useful for indicating which plot should contribute to the frequency-count chart in the **Measurement→Frequency Difference (f)** view, the sigma-tau charts in the various xDEV measurement views, and the noise marker and spur tables in the noise measurement views. The selected plot is also the only one that's saved, exported, modified, edited, or moved up and down in the legend table when the corresponding commands are issued.

The last point above is an important one – many users new to TimeLab are surprised to discover that a single .TIM file doesn't represent an entire screen full of plots. Each plot must be saved, loaded, and otherwise processed individually. None of the commands on the File or Edit menus in TimeLab operate on more than one plot at a time, except for **File→Close all plots (Home)** and the various image-based operations like **File→Print image (Ctrl-p)**, **File→Copy image to clipboard (Ctrl-c)**, and **File→Save image or .TIM file (s)** when the latter command is used to save a .png, .bmp, .tga, .gif, or .pcx image of the entire screen.

Tips for new users

- In TimeLab, almost everything you can do from the menus has a keyboard equivalent, in many cases a single key. **Time spent becoming familiar with the keyboard shortcuts will be rewarded!**

- Use the 53100A in an environment with good ambient temperature control for best results. Avoid placing the hardware near HVAC vents or other sources of drafts when making sensitive measurements.

- Also for best results, avoid placing the 53100A immediately next to equipment that may operate at high temperatures, as well as equipment that should not be exposed to heat. Normal case temperatures may reach 35C to 40C after several hours' operation.

- Do not disturb the measurement setup mechanically, move cables around, or otherwise interact with the environment unnecessarily during acquisition. After the measurement ends, you can use TimeLab's phase-record editing features to get rid of known glitches, but nothing can be done to restore a corrupted noise plot. Any glitches that affect the phase noise or AM noise plots will require a retake.

- At startup time, TimeLab will open any .TIM file(s) specified on the timelab[64].exe command line. Alternatively, a JavaScript program filename ending in .JS may be specified for execution.

- Every measurement in TimeLab runs in its own background thread, so you can load, save, and manipulate plots or even launch additional acquisitions with other instruments while an acquisition is running. This is subject to available memory and CPU resources, of course. A typical PhaseStation 53100A acquisition takes about 16 MB/sec of USB bandwidth and two cores' worth of CPU horsepower.

- When equipped with multiple USB host controllers, the fastest available PCs can acquire data from up to four 53100A devices concurrently. Most PCs will be limited to two concurrent acquisitions at most.

- The acquisition dialogs in TimeLab all have large "mouseover" help windows. The mouseover help text serves as the primary hardware-specific documentation for the various analyzers and counters supported by TimeLab. Read it carefully before changing any fields from their default values.

- Attempting to run stability measurements with large phase records and high data rates may result in "Couldn't allocate phase record" or similar messages indicating a lack of available RAM. Running on an x64 system can help avoid memory problems, as they can access much more RAM than legacy 32-bit Windows systems.

- When contacting Miles Design for technical support, **it's a good idea to attach .TIM file(s) associated with the issue, rather than image files.** .TIM files greater than 3 MB in size should be zipped or otherwise compressed. If you don't receive a reply within 48 hours, please email again (without the attachment) to confirm delivery.

Next steps

Two application notes are provided to illustrate specific measurement techniques and applications that almost all 53100A users will find helpful. These are AN 53100A-1 (*Oscillator Measurement and Calibration with the PhaseStation 53100A*) and AN 53100A-2 (*Dual Reference Noise and Stability Measurements with the PhaseStation 53100A*).

If your 53100A is equipped with the optional dual 100 MHz internal reference module (Option IR), please review the section beginning on page 221 for helpful usage guidelines and measurement tips.

Block diagram

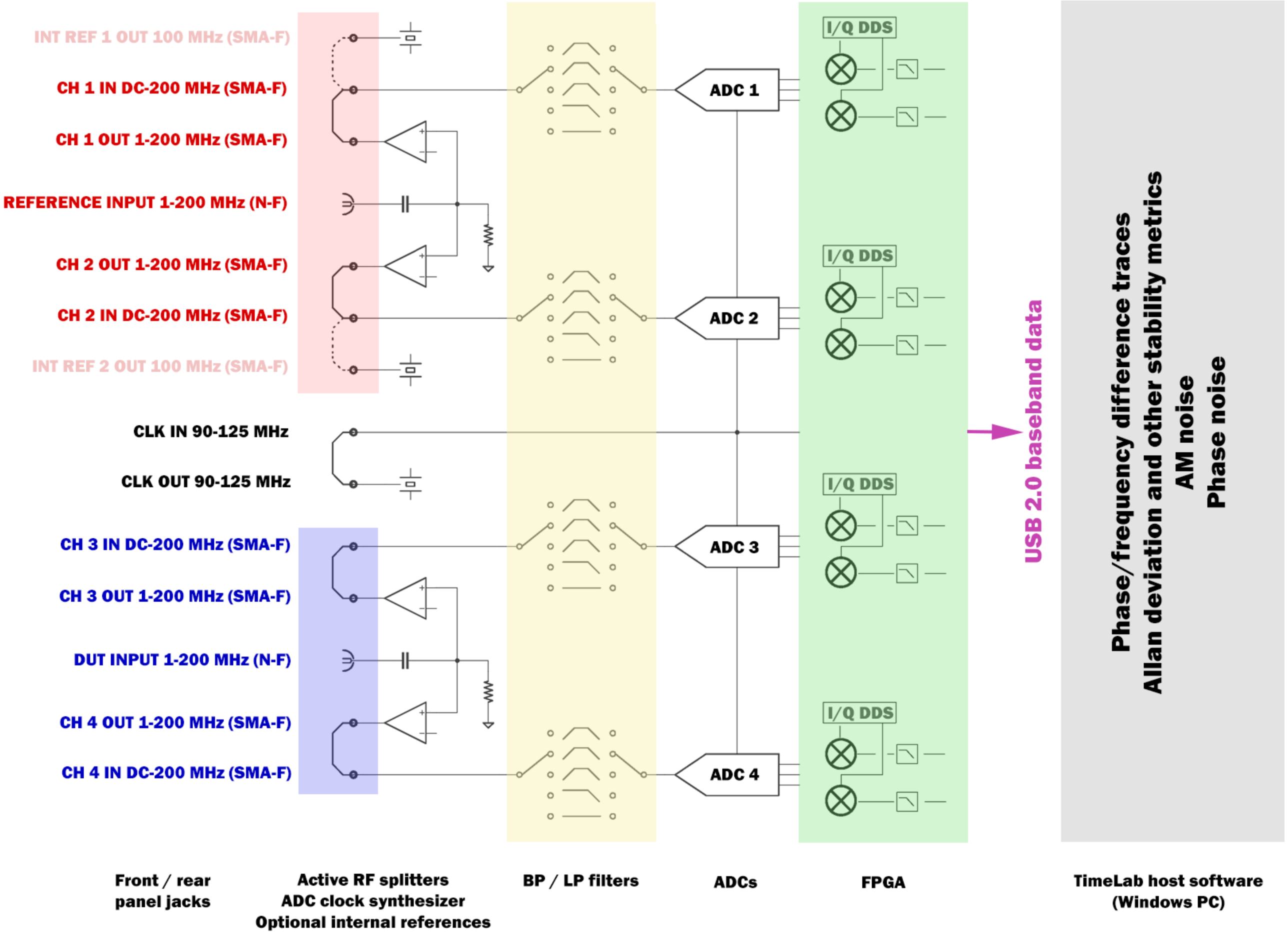

INT REF 1 OUT 100 MHz (SMA-F)
CH 1 IN DC-200 MHz (SMA-F)
CH 1 OUT 1-200 MHz (SMA-F)
REFERENCE INPUT 1-200 MHz (N-F)
CH 2 OUT 1-200 MHz (SMA-F)
CH 2 IN DC-200 MHz (SMA-F)
INT REF 2 OUT 100 MHz (SMA-F)
CLK IN 90-125 MHz
CLK OUT 90-125 MHz
CH 3 IN DC-200 MHz (SMA-F)
CH 3 OUT 1-200 MHz (SMA-F)
DUT INPUT 1-200 MHz (N-F)
CH 4 OUT 1-200 MHz (SMA-F)
CH 4 IN DC-200 MHz (SMA-F)
ADC 1
ADC 2
ADC 3
ADC 4
I/Q DDS
I/Q DDS
I/Q DDS
I/Q DDS
USB 2.0 baseband data
Phase/frequency difference traces
Allan deviation and other stability metrics
AM noise
Phase noise
Front / rear
panel jacks
Active RF splitters
ADC clock synthesizer
Optional internal references
BP / LP filters
ADCs
FPGA
TimeLab host software
(Windows PC)

Figure 6: Functional block diagram

The diagram above provides an overview of the signal and data pathways that lie between the DUT and reference signals supplied to the 53100A and the results that ultimately appear on the screen. Although the block diagram may appear complex at first, the features most likely to be of interest are the signal connections available to the user, all of which are shown at left. Front-panel jacks typically connected to the device(s) under test are highlighted in **blue**, while those associated with internal or external reference signals are shown in **red**.

Other notable aspects of the block diagram include the following points.

- The four SMA M-M jumpers shown in the DUT and reference input areas are depicted in their factory-default configuration. With these connections, signals applied to the two N-female connectors on the front panel are split into two identical paths in the classical topology for cross-correlated noise measurements. Many other connection possibilities are available, however, since all four ADC channel inputs are accessible to the user. These will be discussed later.

- Active splitters are used at the two N-F input jacks rather than typical resistive or Wilkinson types. The active splitters present a well-defined load to the input signals as well as a consistent source impedance to filters and other components in the analog front end. Reverse isolation is approximately 70 dB at 100 MHz, which helps to minimize load-impedance fluctuations caused by attenuator and filter selection when a measurement begins. These load variations can affect the stability of devices being measured, and they can also interfere with other equipment connected to the DUT or reference sources.

Additionally, the active splitters do not present a differential-mode source of Johnson noise at their outputs, which may help prevent certain artifacts in measurements at extremely low noise levels near the thermal floor of -177 dBm/Hz. The shared load resistors do act as a common-mode noise source at the thermal floor, however. It's uncommon for real-world measurements with the 53100A to reach this limit, but the SMA jumpers may be used to connect other types of splitters supplied by the user if required. The use of the internal splitters is always optional.

- On instruments equipped with the dual independent internal reference option, the reference channel jumpers may be moved to connect the two internal 100 MHz OCXOs directly to the input jacks. In this configuration, no connections are made to the CH 1 OUT and CH 2 OUT jacks. However, an external 10 MHz frequency standard may be connected to the REFERENCE input jack to serve as a disciplining source for the 100 MHz INT REF 1 and/or INT REF 2 outputs.

- The CLK IN and CLK OUT jack locations on the rear panel are intended to provide access to the internal ADC clock signal path for future applications. These jacks are unpopulated by default.

- The SMA-F jacks labeled CH 1 IN, CH 2 IN, CH 3 IN, and CH 4 IN are connected to identical banks of switched filters and attenuators that condition the input signals and pass them to the individual ADCs.

A brief architectural note

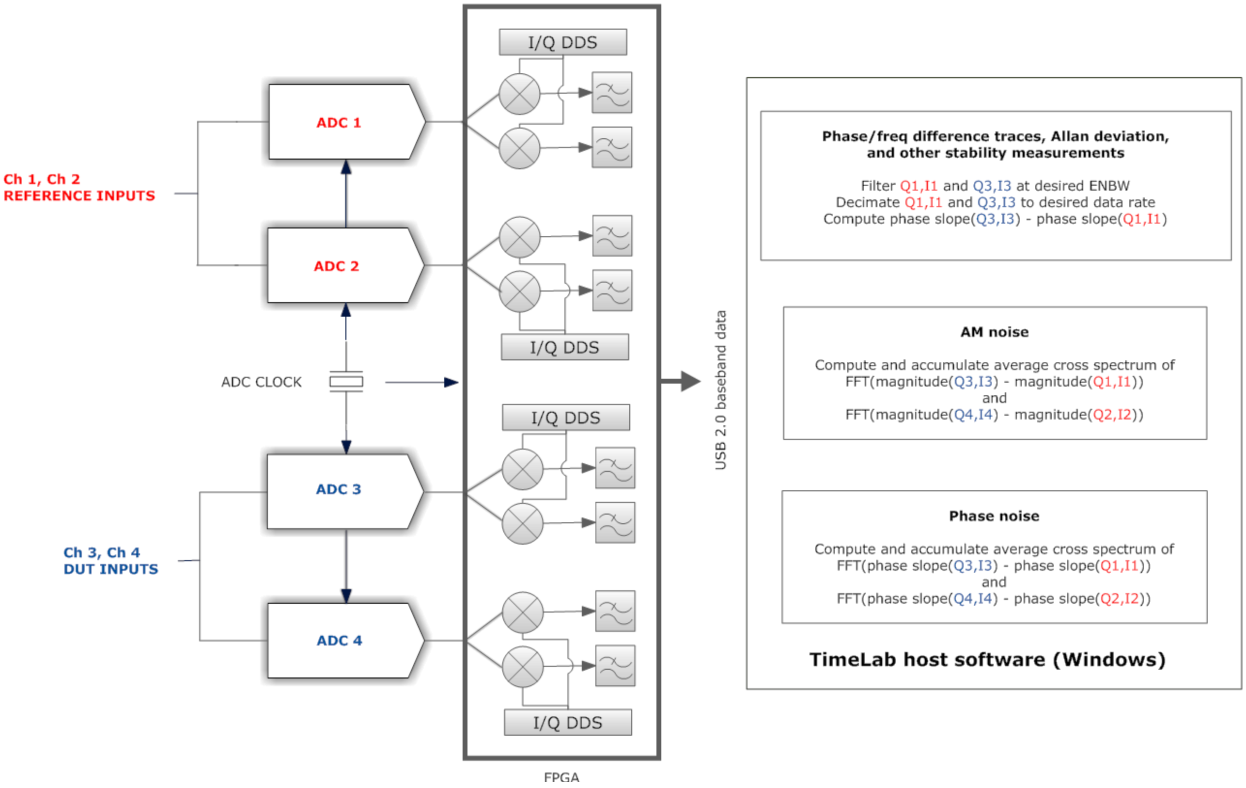

Ch 1, Ch 2
REFERENCE INPUTS
ADC 1
ADC 2
ADC CLOCK
ADC 3
ADC 4
Ch 3, Ch 4
DUT INPUTS
I/Q DDS
I/Q DDS
I/Q DDS
I/Q DDS
FPGA
USB 2.0 baseband data
Phase/freq difference traces, Allan deviation, and other stability measurements
Filter Q1,I1 and Q3,I3 at desired ENBW
Decimate Q1,I1 and Q3,I3 to desired data rate
Compute phase slope(Q3,I3) - phase slope(Q1,I1)
AM noise
Compute and accumulate average cross spectrum of
FFT(magnitude(Q3,I3) - magnitude(Q1,I1))
and
FFT(magnitude(Q4,I4) - magnitude(Q2,I2))
Phase noise
Compute and accumulate average cross spectrum of
FFT(phase slope(Q3,I3) - phase slope(Q1,I1))
and
FFT(phase slope(Q4,I4) - phase slope(Q2,I2))
TimeLab host software (Windows)

Figure 7: Digital processing pipeline

It can be helpful to understand some basic details about the 53100A's DSP topology, especially when measurements may be constrained by available memory and CPU resources. A key observation is that the two basic measurement categories corresponding to the **Measure stability** and **Measure phase noise / Measure AM noise** options in the TimeLab acquisition dialog use **completely separate** data records and internal processing pipelines.

The "phase record" mentioned in the documentation and help text has nothing to do with **Phase noise** measurements, for example. Instead, the phase record is a block of low-bandwidth phase difference samples from a single DUT/reference channel pair. It is used only for **Stability** measurements, which include ADEV, MDEV, TDEV, HDEV, and MTIE as well as the frequency- and phase-difference plots.

The diagram above represents a more detailed view of the right half of the block diagram from the preceding section. While TimeLab allows phase records from multiple channel pairs to be recorded during a single measurement, the most common measurements yield a single phase record created from two parallel streams of complex data samples arriving from one DUT input channel (Q3,I3) and one reference input channel (Q1,I1). (Channels 2 and 4 do not participate in single-DUT stability measurements by default unless a measurement role supporting cross ADEV stability measurements has been selected.) The two streams from channels 1 and 3 are decimated to accommodate the requested measurement bandwidth (ENBW) – typically 5 Hz to 500 Hz, with rates up to 50 kHz supported through the use of additional decimation to avoid excessive phase data record sizes. The phase differences between corresponding samples in the two decimated streams are then calculated and used to construct the various xDEV plots as well as the phase- and frequency-difference plots.

So, when you save a measurement to a .TIM file that was made with the **Measure stability** option checked, you're actually saving these low-bandwidth phase differences. If you use **Edit→Remove selected or zoomed phase data (F4)** or the various flattening or trend-removal functions to edit parts of the phase record after acquisition, the xDEV and frequency/phase-difference plots will be recalculated, but **nothing will happen to the phase noise or AM noise plots**. The noise traces are normally based on the phase and/or amplitude differences for all four baseband streams – Q1,I1, Q2,I2, Q3,I3, and Q4,I4 – which are processed at the full baseband data rate exceeding 200 kS/s. The resulting data is then passed to the FFT and cross-spectrum averaging routines.

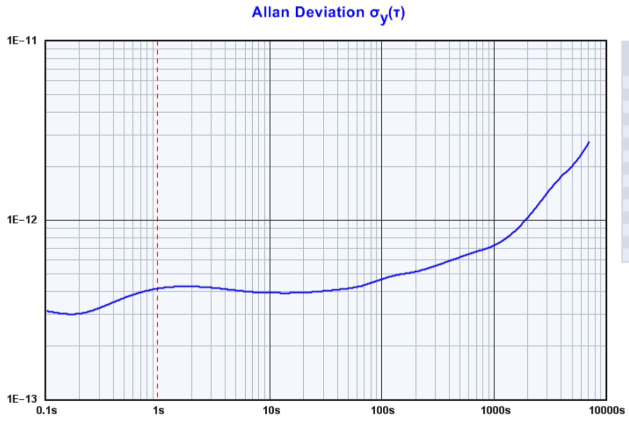
Because it would be impractical to retain the original high-bandwidth complex baseband data or any phase or amplitude information derived from it, only the averaged FFT output bins are written to the .TIM file when saving the phase noise or AM noise portion of a measurement.

Making Measurements

Three views of the same PhaseStation 53100A measurement appear at left, revealing the performance secrets of some of the industry's highest-quality crystal oscillators .

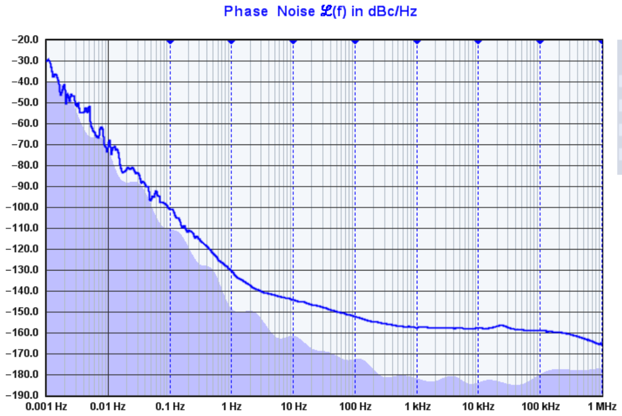
The Allan Deviation plot displays frequency stability at intervals from milliseconds to days, while the Phase Noise plot renders the signal's spectral signature at offsets ranging from 0.001 Hz to 1 MHz. In this example, dual references help the PhaseStation's noise measurement floor approach -150 dBc/Hz at 1 Hz from the carrier.

Meanwhile, a mixture of short-term and long-term stability trends can be examined in the Frequency Difference display. The 53100A's frequency-count chart offers over 13 digits per second of *usable* precision at "gate times" from 0.1 second to over 10000 seconds.



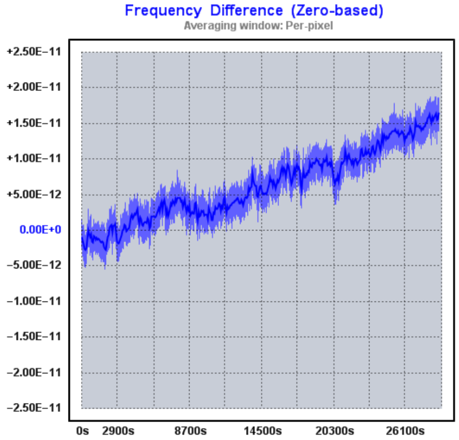
Tau	Sigma(Tau)
1s	4.20E−13
2s	4.30E−13
4s	4.16E−13
8s	3.99E−13
10s	3.97E−13
20s	3.99E−13
40s	4.13E−13
80s	4.52E−13
100s	4.75E−13
200s	5.22E−13
400s	6.00E−13
800s	6.93E−13
1000s	7.32E−13
2000s	1.06E−12
4000s	1.79E−12

Input Freq	Input Amplitude	Ref Freq	Sample Interval	ADEV at 1s	Duration	Instrument	Source A	Source B
5.0 MHz	8 dBm	5.0 MHz	0.100 s	4.20E−13	8h	PhaseStation 53100A	BVA 8600-B	BVA 8607



Noise Marker (Hz)	dBc/Hz
0.1 Hz	−100.9
1 Hz	−130.4
10 Hz	−144.3
100 Hz	−152.1
1 kHz	−157.3
10 kHz	−157.6
100 kHz	−158.8
1 MHz	−165.7

Trace	Notes	Input Freq	Input Amplitude	Ref Freq	Duration	Instrument
DUT=BVA 8607	Ref=2x BVA 8600	5.0 MHz	8 dBm	5.0 MHz	8h	PhaseStation 53100A



Origin	Drift (Hz/sec)	Drift (Hz/hr)
−2.06E−12	+2.91E−9	+1.05E−5

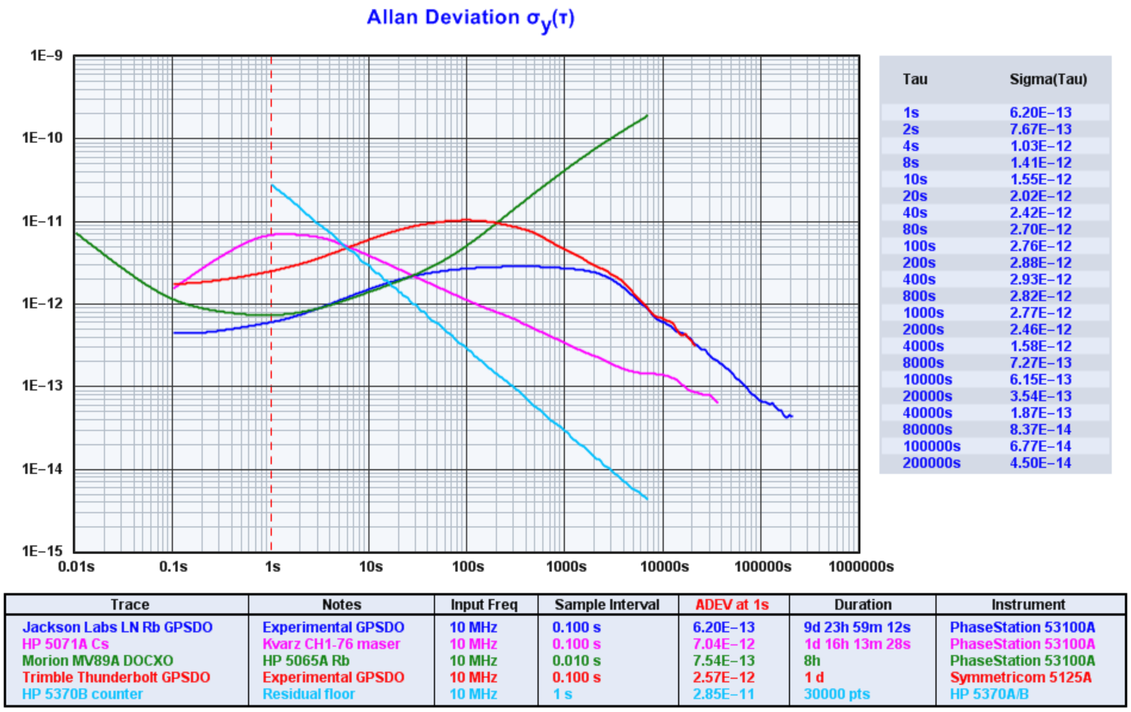
Avg Time (s)	Freq (Hz) at 28800s	Error
0 . 100	5 000 000 . 017 687 799	+3.54E−9
0 . 300	5 000 000 . 017 687 799	+3.54E−9
1	5 000 000 . 017 688 489	+3.54E−9
3	5 000 000 . 017 688 006	+3.54E−9
10	5 000 000 . 017 686 367	+3.54E−9
30	5 000 000 . 017 688 204	+3.54E−9
100	5 000 000 . 017 684 438	+3.54E−9
300	5 000 000 . 017 683 063	+3.54E−9
1 000	5 000 000 . 017 682 337	+3.54E−9
3 000	5 000 000 . 017 676 191	+3.54E−9

Input Freq	Input Amplitude	Ref Freq	ENBW	Duration	Instrument	Source A	Source B
5.0 MHz	8 dBm	5.0 MHz	5 Hz	8h	PhaseStation 53100A	BVA 8600-B	BVA 8607

“What’s all this ADEV stuff, anyhow?”

With apologies to the late Bob Pease and his editors at *Electronic Design*, here are a few tips to help you understand what Allan deviation (ADEV) graphs really tell you, and how to get the most out of the statistical deviation measurements made by TimeLab.

Allan deviation is defined as the *two-sample deviation of fractional frequency differences at a given time interval*. While technically correct, such definitions may be unenlightening to new users. Often, the first question asked is something like, “Why did my plot stop at 1000 seconds? It took an hour to run!”



Tau	Sigma(Tau)
1s	6.20E−13
2s	7.67E−13
4s	1.03E−12
8s	1.41E−12
10s	1.55E−12
20s	2.02E−12
40s	2.42E−12
80s	2.70E−12
100s	2.76E−12
200s	2.88E−12
400s	2.93E−12
800s	2.82E−12
1000s	2.77E−12
2000s	2.46E−12
4000s	1.58E−12
8000s	7.27E−13
10000s	6.15E−13
20000s	3.54E−13
40000s	1.87E−13
80000s	8.37E−14
100000s	6.77E−14
200000s	4.50E−14

Trace	Notes	Input Freq	Sample Interval	ADEV at 1s	Duration	Instrument
Jackson Labs LN Rb GPSDO	Experimental GPSDO	10 MHz	0.100 s	6.20E−13	9d 23h 59m 12s	PhaseStation 53100A
HP 5071A Cs	Kvarz CH1-76 maser	10 MHz	0.100 s	7.04E−12	1d 16h 13m 28s	PhaseStation 53100A
Morion MV89A DOCXO	HP 5065A Rb	10 MHz	0.010 s	7.54E−13	8h	PhaseStation 53100A
Trimble Thunderbolt GPSDO	Experimental GPSDO	10 MHz	0.100 s	2.57E−12	1 d	Symmetricom 5125A
HP 5370B counter	Residual floor	10 MHz	1 s	2.85E−11	30000 pts	HP 5370A/B

Figure 8: Examples of ADEV measurements

The answer is that these plots are not simply linear depictions of frequency drift over time. (In TimeLab, that’s the purpose of the **Measurement→Frequency Difference (f)** view.) Nor are they depictions of a single Allan deviation measurement that describes the clock’s stability from one interval of a given duration to the next. Instead, ADEV and similar statistical graphs usually portray an entire collection of deviation measurements, each based on a list of phase or frequency data points spaced at an interval unique to that measurement.

Imagine that you own a watch that exhibits a certain accuracy in parts per million from one minute to the next. To arrive at this figure, you’ve checked your watch against a better clock several times, waiting one minute between each trial, and then calculated the Allan deviation of the resulting list of phase errors. (You might have used conventional standard deviation, but in this case you’ve heard, correctly, that STDEV has major drawbacks for timekeeping work.)

You're fairly certain you'll get a different result if you calculate the Allan deviation for a set of readings taken one week apart, and still another result for a series of readings taken annually. So you record at least a few data points at these intervals as well.

Now you'd like to draw a graph that shows off your watch's performance, perhaps for bragging rights at the neighborhood pub. What sort of plot would be needed to portray everything you've learned about the watch? The X axis obviously needs to be logarithmic, given that your timescales of interest range from minutes to years. The deviation figures will need to be plotted on a log axis as well, because they might be thousands of times worse at one-year intervals than they are at one-minute intervals. Finally, even though you recorded several years' worth of annual readings, your graph will need to end at the one-year point. There's simply not enough data to describe the watch's behavior at longer timescales.

If you keep these guidelines in mind as you graph your data, you'll end up with a legitimate ADEV plot. It will have three data points, each containing the result (*sigma*) of an independent calculation based on at least a few samples of the clock's frequency that were taken at a specific interval (*tau*).   Lines or curves can be fitted to this collection of discrete points, and the result will be a plot like those drawn by TimeLab and other ADEV applications.

TimeLab's statistical capabilities aren't limited to Allan deviation -- the **Measurement** menu allows you to view any of five types of statistical plots, during or after acquisition. Formulas exist[5] to calculate Allan deviation and other statistics on the basis of both phase and frequency samples. TimeLab always works with phase data internally, using incremental calculations for fast processing and overlapped algorithms for high statistical quality. As with other measurements made by TimeLab, the statistical plots are updated simultaneously in real time.

Allan Deviation (a)

Allan deviation, or ADEV, is the square root of the Allan variance, $\sigma^2_y(\tau)$. Like its related statistics MDEV and HDEV, ADEV describes the fractional frequency deviation (σ, or *sigma*) of a set of samples taken at an interval τ (*tau*).

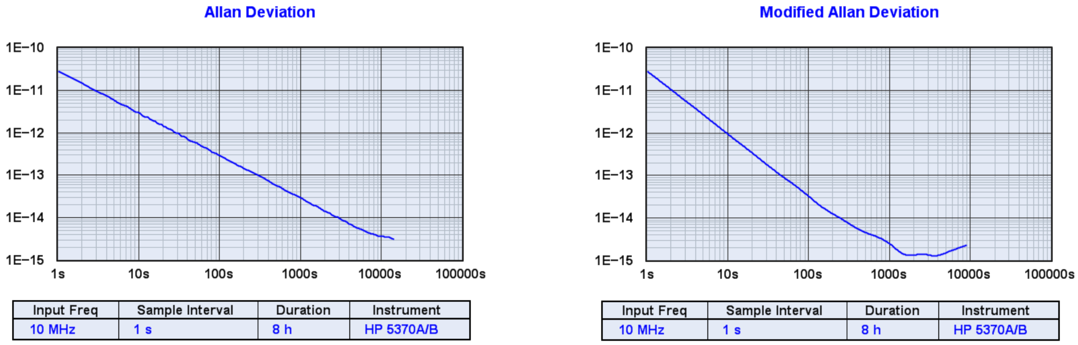
ADEV has a few disadvantages that are addressed by other deviation types, such as inability to distinguish white PM and flicker PM noise. It also exhibits relatively low confidence in the presence of smaller sample sizes, compared to some newer statistics. Still, ADEV is among the most commonly-used performance metrics for high-performance clocks and frequency standards.

[5] W. J. Riley, *Handbook of Frequency Stability Analysis* (http://tf.nist.gov/general/pdf/2220.pdf)

Modified Allan Deviation (m)

MDEV, or Mod $\sigma_y(\tau)$, is a slight variation on ADEV that can discriminate between white PM and flicker PM noise. Both ADEV and MDEV render flicker PM noise at a slope of τ^{-1}. White PM noise is also rendered by ADEV at τ^{-1}, but it appears as a steeper downward slope of $\tau^{-3/2}$ in MDEV. Consequently an ADEV plot may appear artificially elevated when either of these two noise types is present. Switching to **Measurement→Modified Allan Deviation** (m) will render the two noise types separately at the intervals where they belong.

As an example, here are ADEV and MDEV plots showing the residual noise of an HP 5370B time interval counter. For this test, the counter's START input was fed with a 1-pps divider, which was driven by a 10 MHz clock that was also connected to the STOP input using an RF splitter. The counter's noise floor consists almost entirely of white PM and flicker PM noise.



Input Freq	Sample Interval	Duration	Instrument
10 MHz	1 s	8 h	HP 5370A/B

Input Freq	Sample Interval	Duration	Instrument
10 MHz	1 s	8 h	HP 5370A/B

Figure 9: Contrasting the HP 5370B's ADEV and MDEV floors

The ADEV plot shows a consistent τ^{-1} slope to 10000 seconds, which could be indicative of either white PM or flicker PM noise. The MDEV plot renders the white PM noise at a slope of $\tau^{-3/2}$, transitioning to flicker PM near τ=400s or so. Limited sample availability beyond τ=2000s makes it difficult to draw further conclusions, but it's clear that MDEV reveals more about the counter's noise characteristics than ADEV.

Hadamard Deviation (h)

HDEV, expressed as $H\sigma_y(\tau)$, can be thought of as a 3-sample alternative to ADEV. HDEV plots of drift-free sources will generally appear similar to ADEV, but while ADEV fails to converge in the presence of linear drift, HDEV is unaffected by it. This means that switching from ADEV to HDEV will yield results similar to viewing ADEV after an **Edit→Subtract global linear frequency trend (drift line)** (Ctrl-l) operation. (The results will not be identical because HDEV also responds to some high-divergence noise types.)

One interesting use for HDEV is predicting the ADEV performance of crystal oscillators that have not yet fully warmed up. Unless the oscillator is exhibiting frequency jumps or inconsistent drift characteristics, its short-term ADEV will eventually resemble the HDEV performance that was observed only a few minutes after power-up.

Time Deviation (t)

TDEV, or $\sigma_x(\tau)$, expresses the time stability of phase at the specified tau, in units of seconds. Numerically, TDEV is equal to (MDEV * τ) / sqrt(3). It's similar to the TIE (Time Interval Error) statistic used by the telecommunications sector.

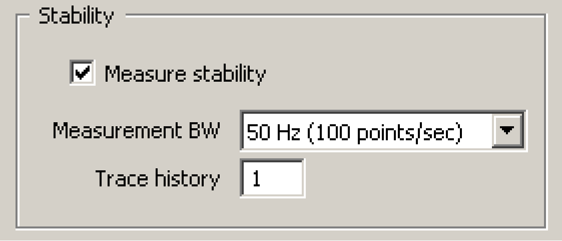
Maximum Time Interval Error (M)

Simply put, MTIE represents the worst-case timing performance of the clock under test over the tau intervals covered by the measurement, as observed by sliding windows of the corresponding tau durations through the phase record. Like TDEV, MTIE is frequently encountered in telecommunications standards specifications.

Along with the other stability metrics supported by TimeLab, the MTIE plot is computed on an incremental basis as successive data points arrive from the counter or phase-difference analyzer and updated in real time. Because MTIE calculations are more time-consuming than the other stability metrics supported by TimeLab, they are performed on only the first 10,000 points of the phase record by default. This limitation can be removed by deselecting the **File→Limit MTIE processing to 10K points** option.

Examining changes in stability over time

To make basic "dynamic ADEV" measurements[6], refer to the help text for the **Duration** and **Run Until** fields in the acquisition dialog, as well as the **Trace History** field shown at right.



Stability

☑ Measure stability

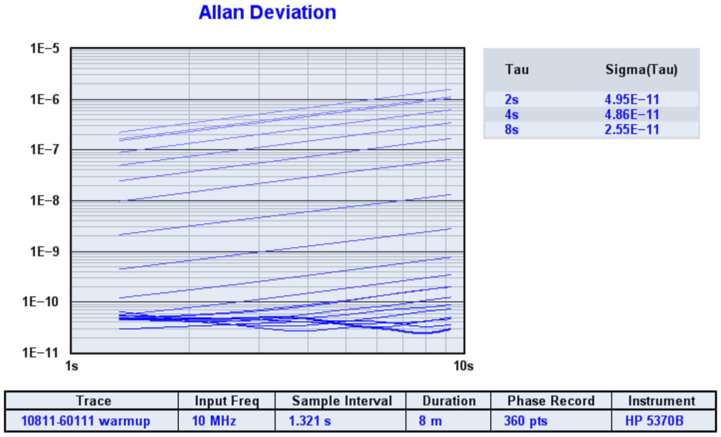
Measurement BW: 50 Hz (100 points/sec)

Trace history: 1

Figure 10: Basic stability measurement parameters

Specifying a **Trace History** value greater than 1 causes TimeLab to divide the phase record into the given number of regions. Subsequent displays of xDEV measurements will assign each region its own trace. As the measurement runs, more recently-acquired regions towards the end of the phase record are drawn in darker colors, revealing how the stability of a given device changes over time. (If **Trace→Toggle trace thickness (T)** is enabled, only the most recent region will be displayed with a heavy trace.)

You can specify a new Trace History value at any time after acquisition in the **Edit→Trace properties (e)** dialog. In the example below, eight minutes of frequency readings were taken from an OCXO shortly after power-up, and later rendered with **Trace History** set to 20. Consequently, each ADEV trace represents about 24 seconds of phase data. As the oscillator warms up, the later traces exhibit better stability. Values displayed on the *Sigma(Tau)* chart apply to the most recent trace.



Tau	Sigma(Tau)
2s	4.95E−11
4s	4.86E−11
8s	2.55E−11

Trace	Input Freq	Sample Interval	Duration	Phase Record	Instrument
10811-60111 warmup	10 MHz	1.321 s	8 m	360 pts	HP 5370B

Figure 11: Dynamic Allan deviation

For more advanced displays of dynamic AVAR/ADEV and related metrics as described in the literature, consider using the **File→Export phase data to Stable32 (Ctrl-x)** command described on page 80 for offline analysis.

[6] Galleani, L. , Tavella, P.; Interpretation of the dynamic Allan variance of nonstationary clock data, *Frequency Control Symposium, 2007 Joint with the 21st European Frequency and Time Forum. IEEE International,* pp. 992-997, 2007

Common artifacts in ADEV and related measurements

High-resolution plots captured by the PhaseStation 53100A can reveal artifacts that don't seem to be present when the same measurement is made by other instruments. For example, spurs due to AC power coupling or ground loops, discussed below, appear very different when acquired at high sample rates and rendered with 20 or more ADEV bins per decade. Line-related spurs may not have been resolvable before, but that doesn't mean they weren't present!

Measurement bandwidth and τ_0

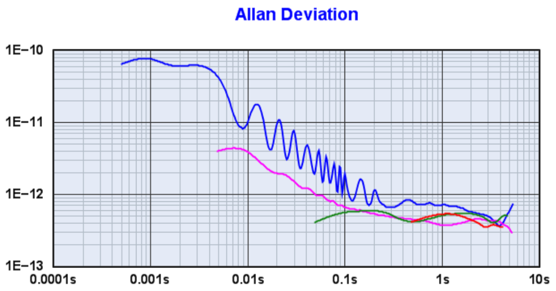


Figure 12: Artifact near beginning of unclipped traces

One artifact that *does* originate in the 53100A driver software is shown in the plot at right. Four successive ADEV acquisitions are shown, each conducted at a different noise-equivalent measurement bandwidth (ENBW) setting. Near the beginning of each trace, some "droop" is present that doesn't accurately reflect the behavior of the DUT.

What causes this artifact? Stein[7] has shown that the optimum ENBW for a given ADEV τ_0[8] is simply the 1/(2 * τ_0) Hz Nyquist rate. This bandwidth is unachievable with a non-ideal antialiasing filter, so the 53100A driver internally acquires oversampled phase data that corresponds to an artificially short τ_0 interval relative to the selected ENBW. If allowed to appear on an ADEV plot, the antialiasing filter's response is easy to mistake for a real effect.

TimeLab supports two different ways to avoid displaying invalid data near τ_0. First, **Trace→Clip xDEV traces by noise bandwidth (**Ctrl-b**)** is enabled by default. (This option was turned off to demonstrate the issue above.) With noise bandwidth clipping enabled, as seen in the figure at left, TimeLab does not render any portion of an xDEV trace at taus shorter than 1/(2 * ENBW) seconds.

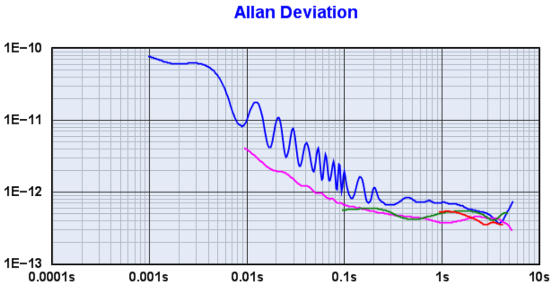


Figure 13: Traces clipped by bandwidth

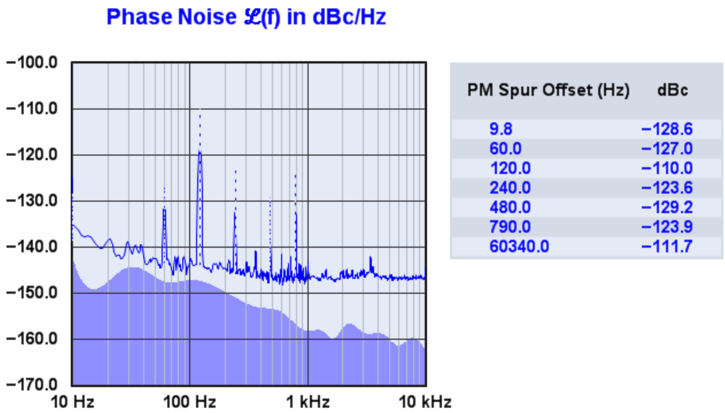
[7] *The Allan Variance – Challenges and Opportunities*, Samuel R Stein, Symmetricom, Inc., Boulder, Colorado USA

[8] τ_0 refers to the very first tau at the left end of an xDEV trace, which typically corresponds to the period of the phase or frequency samples being analyzed.

Additionally, the 53100A driver can decimate the phase-difference data it acquires. Readings are collected by sampling the internal phase-difference data stream at a rate determined by the **Output Decimation** field in the **Additional options** page of the acquisition dialog. This feature is primarily for convenience. It can be used to reduce the size of phase records generated by very long acquisitions, for example, or when high data rates must be used in order to track drifting sources.

Decimation factors of 2:1 or greater also have the effect of satisfying the previous τ_0 >= 1/(2 * ENBW) criterion. With the default **Output Decimation** value of 2:1 or any higher ratio, phase-difference data that's affected by lowpass attenuation will be kept out of the measurement, even if **Trace→Clip xDEV traces by noise bandwidth (**Ctrl-b**)** is turned off.

Note that **Trace→Clip xDEV traces by noise bandwidth (**Ctrl-b**)** will have no effect on plots rendered with data captured from counters and other instruments that don't report their measurement bandwidth. It should be left enabled in most cases.



PM Spur Offset (Hz)	dBc
9.8	−128.6
60.0	−127.0
120.0	−110.0
240.0	−123.6
480.0	−129.2
790.0	−123.9
60340.0	−111.7

Figure 14: AC line interference responsible for ripple in ADEV plots

Line-related spurs

The most conspicuous artifact in the ADEV plots above is the "ripple" in the **blue** and **magenta** traces. This is caused by AC line interference, primarily a single 120-Hz spur:

Note that 1/120 Hz = 0.008s, which corresponds to the first (genuine) trough in the ADEV traces above. Depending on how they're introduced to the measurement, AC line spurs commonly appear at either the fundamental power-line frequency or its second harmonic. Users in the EU and other locales with 50 Hz power will see similar effects near τ=0.010 and 0.020s. In all of these cases, the device's "true" ADEV is close to an imaginary line drawn through all of the lowest points in the trace.

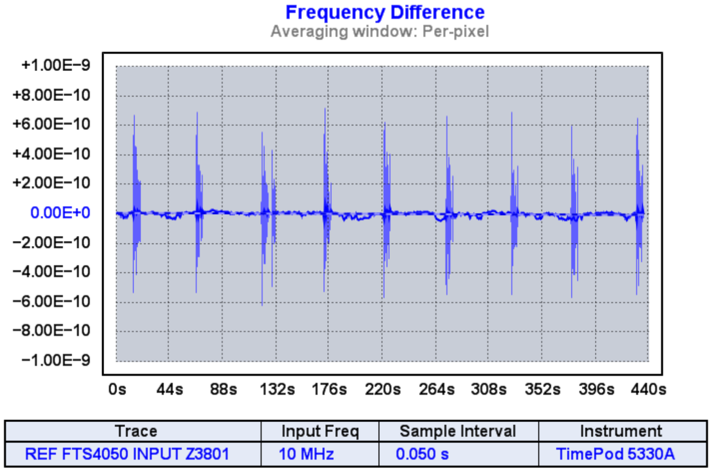
Getting rid of AC line spurs can be a challenge. Shielding at 50/60 Hz is usually impractical; if the interference is magnetically coupled into the measurement, there is probably no cure besides identifying the offending source(s) and moving sensitive equipment and cables away.

Ground loops are a more likely offender, since all ports on the 53100A have ground pins or shields that are bonded to the instrument's metal enclosure. If you can identify the offending loop, you may be able to break its RF path with a coaxial balun[9].

You can also attack the source of the loop by bonding all equipment to a common ground that's connected to the building's power distribution network in only one place. To do this, ensure that all devices participating in the measurement – DUT, reference, instrument, computer, and everything connected to them – are all part of the same AC circuit, preferably a power strip that allows all equipment to be plugged into a single outlet.

Using battery power can help, but batteries shouldn't be your first resort. Less drastic solutions can usually be found that will let you use the available AC power. (Try switching to MDEV, for instance.) If you are measuring stable sources with sufficient warmup time, you can attenuate line spurs by selecting the 5 Hz **Measurement BW** option in the acquisition dialog. As another alternative, consider the **Edit→Apply notch filter to phase data (n)** command, described on page 91.

Other environmental hazards



Trace	Input Freq	Sample Interval	Instrument
REF FTS4050 INPUT Z3801	10 MHz	0.050 s	TimePod 5330A

Figure 15: Interference from nearby cellular telephone

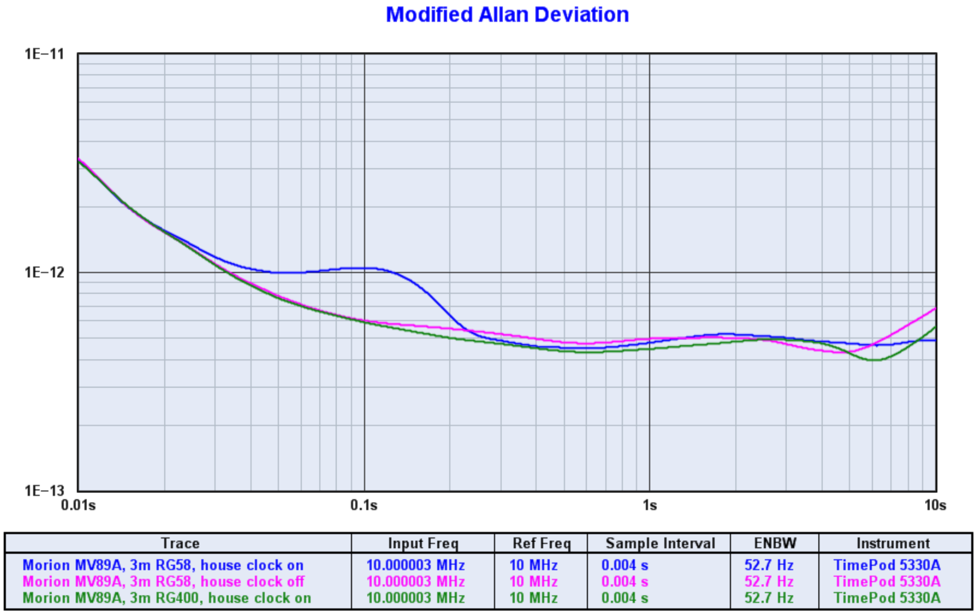
[9] E.g., http://www.minicircuits.com/pdfs/FTB-1-1+.pdf. When using coaxial baluns, make sure they're actually helping, though. In particular, watch out for elevated phase noise floors, crosstalk artifacts, and assorted spurious responses picked up by the now-ungrounded coax shield. Coaxial baluns turn loop antennas into dipole antennas.

Above, a measurement of two 10 MHz sources was interrupted by bursts of apparently-random noise at roughly 50-second intervals. The culprit turned out to be a cellular phone sitting on a nearby bench. In another instance, a breadboarded RF amplifier undergoing residual noise testing was affected by a WiFi access point, with interference appearing as an intermittent train of low-level impulses near 10 Hz. Try to keep all RF radiation sources – intentional and otherwise! – well away from your testing area.

Crosstalk

Finally, don't confuse power-line spurs or other sources of low-frequency interference with *crosstalk*. This is an insidious artifact that may be observed in high-performance measurements – or worse, may go unnoticed.

When working with direct-digital analyzers such as the PhaseStation and TimePod, crosstalk tends to appear when the signal frequencies at the input ports are nearly the same, but not precisely. As the two signals' phases slowly approach, coincide, and separate, any coupling between them creates a spurious response at the difference frequency. Appearing at a similar magnitude in both AM and phase noise plots, these spurs also cause peaks and nulls in ADEV and related measurements.



Trace	Input Freq	Ref Freq	Sample Interval	ENBW	Instrument
Morion MV89A, 3m RG58, house clock on	10.000003 MHz	10 MHz	0.004 s	52.7 Hz	TimePod 5330A
Morion MV89A, 3m RG58, house clock off	10.000003 MHz	10 MHz	0.004 s	52.7 Hz	TimePod 5330A
Morion MV89A, 3m RG400, house clock on	10.000003 MHz	10 MHz	0.004 s	52.7 Hz	TimePod 5330A

Figure 16: Interference from external 10 MHz source

Crosstalk can also occur when a *third* signal, incidental to the measurement environment, causes beatnote effects that are asynchronous to both test inputs. This case is illustrated in the plot above, where a TimePod 5330A was used to measure a high-stability OCXO against another one of similar quality.

In this case, while the OCXO driving **REF IN** was very close to 10 MHz, the OCXO driving the TimePod's **INPUT** port was set approximately 3 Hz higher for demonstration purposes. Also present in the room (but not participating in the measurement) was a GPS-disciplined frequency standard and distribution amplifier supplying 10 MHz to various instruments through long cables.

The 5330A's reference source was connected with a double-shielded RG400 cable in all three trials. The **blue** trace and **magenta** traces were taken with the input source connected with a 3m length of RG58. A significant MDEV peak was visible in the **blue** trace… but when the GPSDO was powered down for the **magenta** trace, the spurious response vanished.

Likewise, as the **green** trace shows, replacing the RG58 input cable with an equal length of RG400 also eliminated the interference from the house standard.

As can be seen, the recommendation to use double-shielded cables with these instruments is not born of excessive caution. Even plots acquired by conventional frequency and TI counters are vulnerable to crosstalk! With the widespread availability of GPSDOs and other inexpensive frequency standards, many facilities from international physics laboratories to ham shacks are equipped with 5- and 10-MHz distribution networks. **Don't use low-quality cables with high-performance instrumentation.** (Tip: make it a habit to measure the end-to-end shield resistance of the coaxial cables that you use for time and frequency measurements, whether single- or double-shielded. Reject any that exceed a few tenths of an ohm.)

Hints for xDEV measurements

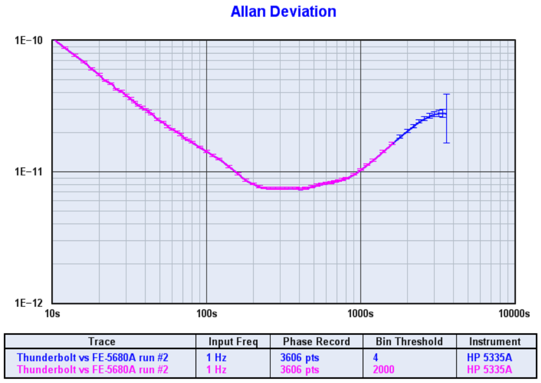
- The 53100A does not display its own measurement floor in the ADEV, MDEV, HDEV, TDEV, or MTIE views. Instead, you can (and should) make residual plots for measurements that may approach the instrument floor, using similar frequencies and signal levels. These residual plots can be saved, loaded and shown with **Display→Overlay all loaded plots (**o**)** next to your actual measurement traces.

- There's a direct relationship between measurement bandwidth and the instrument noise floor in ADEV and other phase/frequency measurements . You may find it helpful to observe drift-prone sources or oscillator startup characteristics at 5 kHz or 50 kHz ENBW, but be aware that the stability noise floor will be somewhat higher. Again, a separate residual plot at the frequencies and signal levels of concern can tell you exactly where the limits are.

- The measurement bandwidth also influences the size of the phase record that must be allocated before a measurement of a given duration can begin. As a result, you may experience out-of-memory errors if you tell TimeLab to record several hours' worth of phase data at 500 Hz ENBW/1K points per second. Increasing the **Output decimation** value in the **Additional Options** dialog is a great way to reduce acquisition memory requirements, but it will not improve the xDEV noise floor. The only way to lower the xDEV floor is to select a narrower measurement bandwidth.

- Fewer samples are available to contribute to bins near the right end of an xDEV trace. As a result, measurement confidence decreases at longer tau intervals. To increase the minimum number of samples that must contribute to a given xDEV bin in order for that bin to be displayed, you can change the **Bin Threshold** parameter in the acquisition dialog, or use **Edit→Trace properties (**e**)** to specify a new value for the same field at any time after acquisition.

However, because TimeLab employs overlapped xDEV algorithms, small changes to **Bin Threshold** may have no effect. Once the phase record grows long enough to provide data at a given tau interval, each additional sample adds another data point to the corresponding xDEV bin. If this is an issue, try setting the **Bin Threshold** to a large fraction of the total phase record size. The idea is to force TimeLab to render the xDEV trace as if much less data were available.



Trace	Input Freq	Phase Record	Bin Threshold	Instrument
Thunderbolt vs FE-5680A run #2	1 Hz	3606 pts	4	HP 5335A
Thunderbolt vs FE-5680A run #2	1 Hz	3606 pts	2000	HP 5335A

Figure 17: Bin threshold adjustment

In the example at left, two copies of the same 3600-sample .TIM file have been loaded. The **blue** trace was rendered with the default **Bin Threshold** of 4, while the shorter **magenta** trace was rendered with **Bin Threshold** set to 2000. Enabling **Trace→Show xDEV error bars** (Ctrl-e**)** reveals the **blue** trace's low confidence at its longest tau.

Generally, as long as **Trace→Clip xDEV traces by confidence** (Ctrl-v**)** is enabled, it will not be necessary to alter the **Bin Threshold** value to display xDEV trace data of good statistical quality. The default bin threshold and clipping options will suffice for almost all measurements, particularly those made at high data rates by instruments such as the PhaseStation 53100A.

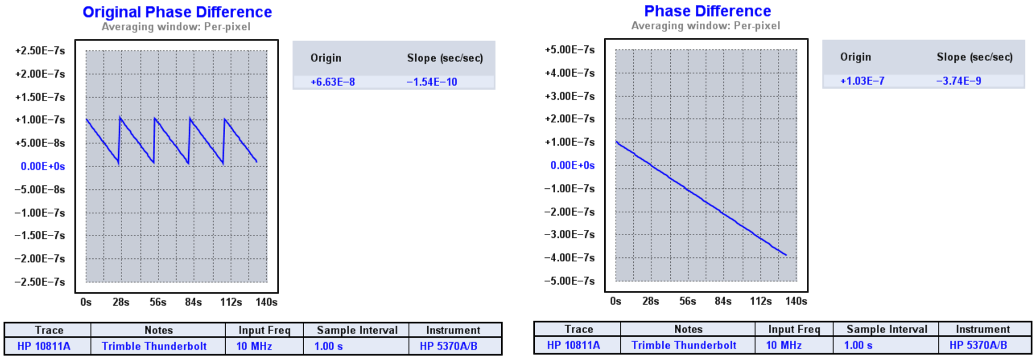
Working with phase- and frequency-difference traces

Phase and frequency stability data acquired by TimeLab is represented internally as an array of phase-difference samples. This is true regardless of whether the data came from a frequency counter, a time-interval counter, or a direct-digital timing analyzer. (Frequency readings are converted to phase-difference samples on the basis of their deviation from the first frequency reading acquired.)

In addition to the statistical measurements such as Allan deviation that TimeLab provides, "strip chart" views are available to give you a closer look at the raw phase-difference data. These views correspond to the **Phase difference (Unwrapped)** (p), **Phase difference (Original)** (w), and **Frequency-difference** (f) options on the **Measurement** menu.

Phase difference (Original) (w)

Phase difference (Unwrapped) (p)



Origin	Slope (sec/sec)
+6.63E−8	−1.54E−10

Trace	Notes	Input Freq	Sample Interval	Instrument
HP 10811A	Trimble Thunderbolt	10 MHz	1.00 s	HP 5370A/B

Origin	Slope (sec/sec)
+1.03E−7	−3.74E−9

Trace	Notes	Input Freq	Sample Interval	Instrument
HP 10811A	Trimble Thunderbolt	10 MHz	1.00 s	HP 5370A/B

Figure 18: Original phase data received from TIC (w) compared with unwrapped view (p)

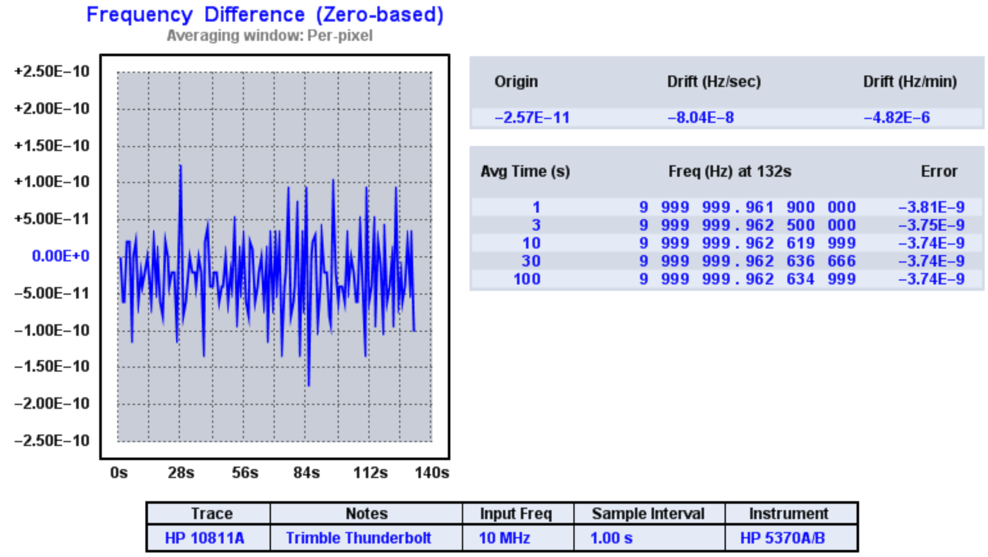
The distinction between these two measurement types is relevant when receiving time-interval data from a traditional counter. The **Phase difference (Original)** (w) measurement shows the original TI samples as they arrive from the hardware. When acquired from a TI counter, these readings are subject to "wraparound" when they exceed the period of the signal that determines the START-to-STOP channel interval. For instance, if an oscillator near 10 MHz is measured against a drift-free source at exactly 10 MHz, the TI readings from the counter will increase or decrease from one trigger interval to the next. When the START-to-STOP interval exceeds 100 nanoseconds or falls below 0 nanoseconds, the TI readings will necessarily wrap back around to the opposite "rail." This phenomenon gives rise to the sawtooth-shaped phase trace familiar to users of vector network analyzers. As seen in the measurement example above, the phase-wrapping effect is visible in the **Phase difference (Original)** (w) measurement view in TimeLab.

Conversely, TimeLab's **Measurement→Phase difference (Unwrapped)** (p) view displays the TI samples in the form of phase data that's been "unwrapped." The sawtooth discontinuities are removed by adding or subtracting the input signal's period whenever the sample-to-sample time difference exceeds half of that period.

It's seldom necessary to refer to the original phase-difference graph except when troubleshooting configuration problems with live TI counter measurements. In most cases you should select the **Measurement→Phase difference (Unwrapped)** (p) view when you wish to inspect the principal phase record for your acquisition. Additionally, all of the flattening, detrending, and data-removal commands on the **Edit** menu work by replacing the original phase data with modified data based on the unwrapped record. After executing any of these commands, the **p** and **u** views will always be identical, just as if the data had been acquired from a frequency counter or direct-digital analyzer.

Further references to the phase-difference view in this manual will appear simply as **Measurement→Phase difference** (p), omitting the distinction between wrapped and original phase data. When working with the PhaseStation and TimePod analyzers, the original and unwrapped phase difference displays are equivalent.

Frequency difference (f)



Origin	Drift (Hz/sec)	Drift (Hz/min)
−2.57E−11	−8.04E−8	−4.82E−6

Avg Time (s)	Freq (Hz) at 132s	Error
1	9 999 999 . 961 900 000	−3.81E−9
3	9 999 999 . 962 500 000	−3.75E−9
10	9 999 999 . 962 619 999	−3.74E−9
30	9 999 999 . 962 636 666	−3.74E−9
100	9 999 999 . 962 634 999	−3.74E−9

Trace	Notes	Input Freq	Sample Interval	Instrument
HP 10811A	Trimble Thunderbolt	10 MHz	1.00 s	HP 5370A/B

Figure 19: Frequency difference measurement view with count chart

Shown above is a third way to view the phase data from the earlier example. Each point on the **Measurement→Frequency difference** (f) trace is calculated by dividing the difference between successive pairs of phase-difference samples by the sample rate. Taking the time derivative of *n* adjacent phase-difference points in this manner yields a stream of *n-1* frequency-difference points. Because of this differentiation process, the

Measurement→Frequency difference (f) view of the phase record is often better at revealing glitches and other short-lived artifacts in the phase record than the actual phase-difference trace is. Long-term oscillator drift and temperature-dependent effects are also easily spotted.

When the optional chart to the right of the trace area is enabled with **Display→Numeric table (Ctrl-n)**, TimeLab can be used to emulate a frequency counter with impressive performance. The chart can be viewed at any time during or after acquisition. Its entries are computed in real time by integrating the frequency-difference values in the currently visible trace at successively longer intervals. The integration process moves from right to left, starting with the newest sample points and moving farther back in history as more data becomes available.

How does TimeLab measure frequency?

It's important to understand that the "frequency differences" shown on the graph are actually fractional deviations from a value known as the *input frequency*. (Under most conditions this is the same value as the *DUT frequency* in the **Legend→Select** menu.) If the measurement's DUT input frequency is exactly 10,000,000 Hz, then a frequency-difference reading of +1E-7 would mean that the absolute frequency is 10,000,001 Hz at that moment in time.

But what exactly is this "input frequency?" Where does it come from, and how accurate is it? When acquiring data with a frequency counter, it's easy enough to determine the nominal input frequency for the measurement: we simply use the first reading. All subsequent data points in the phase record are computed using the difference between the incoming frequency readings and the first one recorded… and all of these readings can be assumed to carry the same accuracy and precision.

When a time-interval counter (TIC) is used, however, TimeLab sees only a series of START-to-STOP interval times. It's impossible to infer the input frequency automatically. Instead, you must enter it into the appropriate field of the acquisition dialog, specifying at least as much precision as you expect to obtain when viewing the frequency count chart.

Phase/frequency measurements with the 53100A

On the 53100A, the process of determining the input frequency can be thought of as a choice between the frequency-counter and TIC methods. Options in the **Acquire→Jackson Labs PhaseStation / Microchip 53100A** acquisition dialog allow you to specify the anticipated DUT and reference frequencies directly or measure them at a specified rounding precision. For more details, please see the *PhaseStation 53100A Measurement Roles* section on page 131.

Examining traces in detail

In the example below, a crystal oscillator was measured over a 12-hour period. The **Measurement→Phase difference** (p) and **Measurement→Frequency difference (f)** views appear side-by-side for comparison.

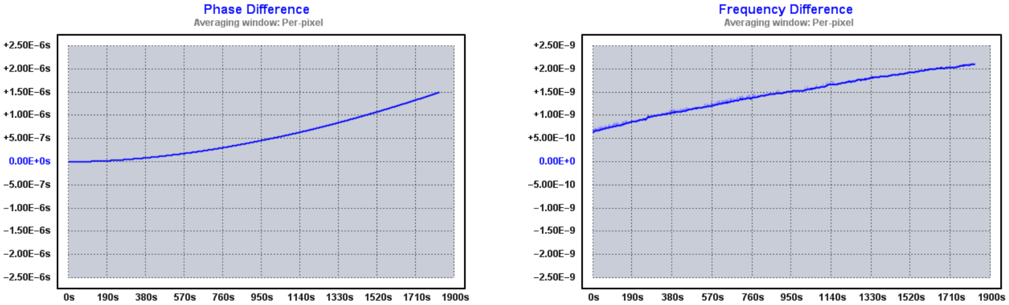


Figure 20: Two views of the same phase data record

Not much detail is visible in either trace. Some sporadic frequency jumps are present, but they're not easily examined since their height is such a small fraction of the plot's overall magnitude. **Subtract global linear phase trend (frequency offset)** (Ctrl-o**)** command expands the phase-difference trace somewhat by removing its linear trend, but it only moves the frequency-difference trace down a bit:

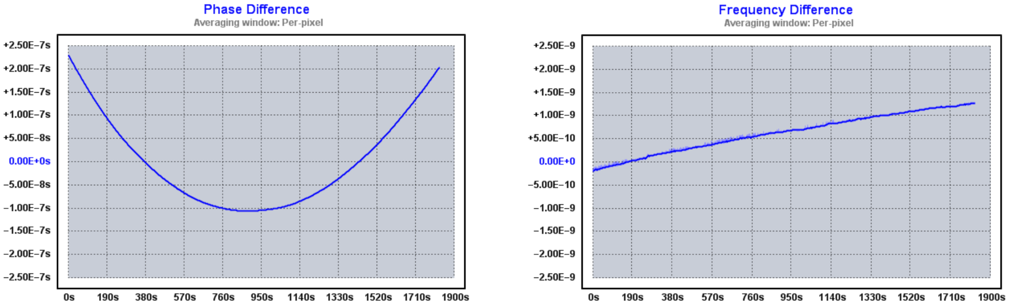


Figure 21: Subtracting the linear phase trend

In fact, we could have achieved a similar effect without altering the phase record by viewing the phase-difference trace with **Trace→Show linear phase/frequency residual**

(r**)**, or by viewing the frequency-difference trace with **Trace→Phase/frequency traces begin at zero** (z).[10]

For these reasons, **Subtract global linear phase trend (frequency offset)** (Ctrl-o) is not a commonly-needed command. At first glance, **Subtract global linear frequency trend (drift line)** (Ctrl-l) seems more useful:

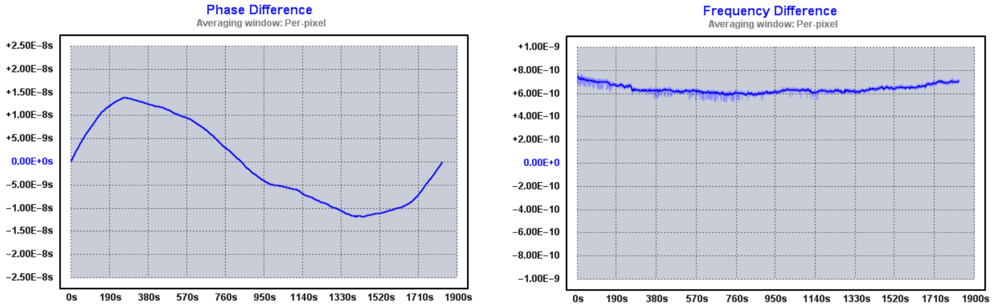


Figure 22: Removal of linear frequency drift

Essentially, **Subtract global linear frequency trend (drift line)** (Ctrl-l) removes the quadratic trend from the phase record, which has the effect of removing the *linear* trend from the frequency-difference plot. But there's still an arbitrary offset in the frequency-difference view that's forcing the Y-axis scale to +/- 1E-9, and the artifacts we'd like to examine are much smaller than this.

Navigating zoomed graphs

One strategy is to unlock the Y-axis with **Trace→Phase/frequency Y axis unlocked in zoom mode** (y)**,** and then use the left mouse button to drag a box around the trace. When the button is released in the view at left, the result is the zoomed view at right:

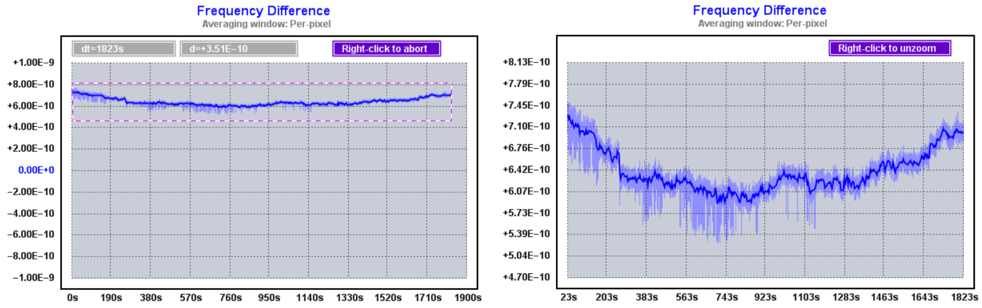


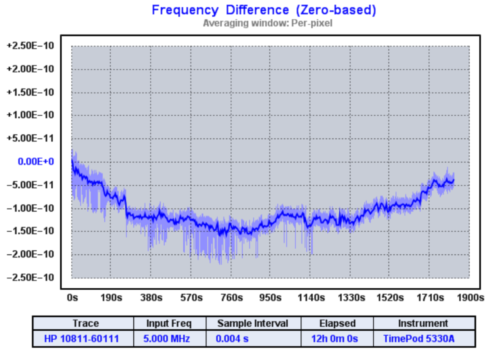
Figure 23: Zooming in

[10] The **Trace→Phase/frequency traces begin at zero** (z) option can actually be more useful – not only can it be toggled on and off without altering the phase data, but it also forces the beginning of the frequency-difference trace to midscale. See page 66.

TimeLab is designed to take full advantage of a PC mouse with three buttons and a scroll wheel. All of the graph types are "zoomable," and you can always return to the unmagnified view by right-clicking anywhere on the plot. Unlike the xDEV and noise measurement displays, though, the phase- and frequency-difference measurement views support additional navigation options. You can expand and contract the zoomed area with the mouse wheel, or surf through the data by dragging with the middle button.

Unlocking the Y axis with **Trace→Phase/frequency Y axis unlocked in zoom mode** (y) keeps TimeLab from constantly adjusting the Y axis to accommodate the graph's vertical range. With the Y axis unlocked, you can expand and scroll the graph in both directions with the mouse, rather than being limited to side-to-side navigation.

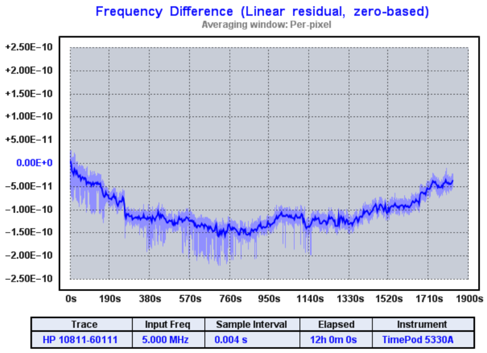
In this case, though, unlocking the Y axis and drawing a box requires a lot of user interaction. Worse, you'll end up with an "unfriendly" Y-axis scale whose random-looking values make the graph difficult to interpret. **Trace→Phase/frequency traces begin at zero** (z) offers a better alternative with a single keystroke:



Trace	Input Freq	Sample Interval	Elapsed	Instrument
HP 10811-60111	5.000 MHz	0.004 s	12h 0m 0s	TimePod 5330A

Figure 24: The 'z' zero baseline mode

To sum up, selecting **Measurement→Frequency difference (f)**, applying **Subtract global linear frequency trend (drift line)** (Ctrl-l) , and enabling **Trace→Phase/frequency traces begin at zero** (z) can reveal details which are impossible to spot in the raw phase-difference trace. All of this being said, in order to get the same detailed view, we *really* only needed to enable **Trace→Show linear phase/frequency residual (r)** while viewing the frequency difference trace!



Trace	Input Freq	Sample Interval	Elapsed	Instrument
HP 10811-60111	5.000 MHz	0.004 s	12h 0m 0s	TimePod 5330A

Figure 25: The linear residual display

The **Trace→Phase/frequency traces begin at zero** (z) and **Trace→Show linear phase/frequency residual** (r) commands are both extremely useful. Enabling one or both of these features can give you a detailed view of most phase- or frequency-difference traces without altering the phase data itself[11]. Keep the r and z keys in mind while working with these traces, and experiment with them frequently.

Hints for phase/frequency stability measurements

- Due to slight variations in FPGA/USB interface temperature that occur when acquisition begins, the 53100A's own impact on critical residual-phase measurements can be minimized if you use **Acquire→Enable deferred acquisition** (ctrl-d) to force the acquisition to throw away incoming data until you press Enter to 'trigger' it. This feature is normally used to improve synchronization in multiple acquisitions, but you can also use it to force a single instrument to reach its equilibrium temperature before the software saves any data.

  You can accomplish the same thing by selecting the **Measurement→Phase difference** (p) view, using the left mouse button to select the first few minutes of data where the phase slope may be artificially degraded. When you release the button it will zoom to the selected area of the plot. You can then hit **F4** to delete that chunk of data from the phase record. The ADEV and other phase-derived graphs will be recalculated. (This will *not* affect the phase or AM noise plots; as mentioned elsewhere, they are acquired with a completely different "signal path" than the one that decimates and records the phase-difference data.)

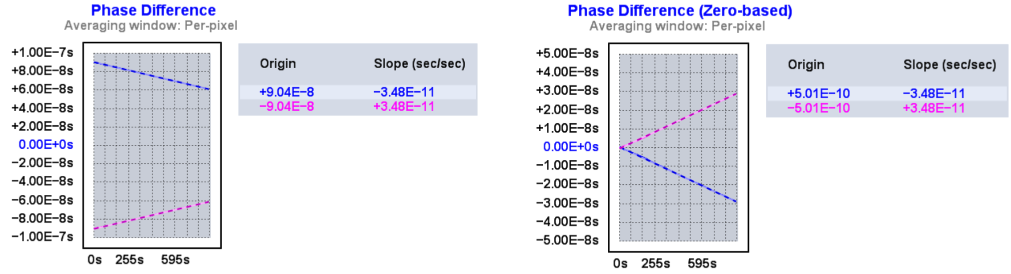
- Pressing **Scroll Lock** will latch the current Y-axis limits, preventing them from changing regardless of the zoom status or **Trace→Phase/frequency Y axis unlocked in zoom mode** (y) option. This feature can be useful when you want to keep the view from being rescaled after removing, detrending, or flattening trace data, such as when similar modifications are

[11] Note that the Subtract quadratic linear frequency trend (drift curve) (Ctrl-q) operation has no "view-only" equivalent. It operates in a manner similar to the Subtract global linear phase trend (frequency offset) (Ctrl-o) and Subtract global linear frequency trend (drift line) (Ctrl-l) commands, and can be even better at revealing details in plots like this one.

being applied to multiple traces in an unzoomed view. Turning **Scroll Lock** off will restore the previous Y-axis control state.

Most users will not need to latch the Y axis limits. **Scroll Lock** is one of the few operations in TimeLab with no menu-based equivalent.

- To invert the phase slope, you can use the **Edit→Trace properties… (e)** dialog to specify a **Rescale Phase** value of -1. This will invert both the slope and the origin of the phase data, but you can still use **Trace→Phase/frequency traces begin at zero (z)** to display the inverted trace at a fixed origin of zero.

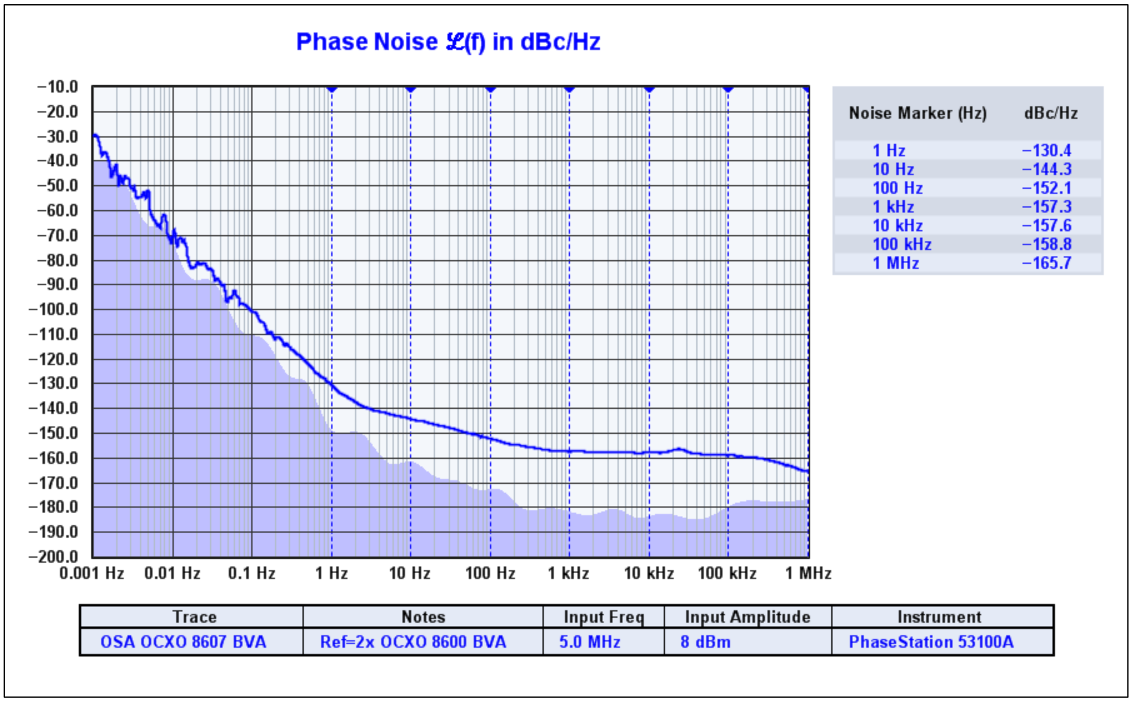


Origin	Slope (sec/sec)
+9.04E−8	−3.48E−11
−9.04E−8	+3.48E−11

Origin	Slope (sec/sec)
+5.01E−10	−3.48E−11
−5.01E−10	+3.48E−11

Figure 26: Phase inversion

Phase noise, AM noise, and jitter



Noise Marker (Hz)	dBc/Hz
1 Hz	−130.4
10 Hz	−144.3
100 Hz	−152.1
1 kHz	−157.3
10 kHz	−157.6
100 kHz	−158.8
1 MHz	−165.7

Trace	Notes	Input Freq	Input Amplitude	Instrument
OSA OCXO 8607 BVA	Ref=2x OCXO 8600 BVA	5.0 MHz	8 dBm	PhaseStation 53100A

With the 53100A, the most challenging oscillator measurements are as easy to perform as the simplest ones

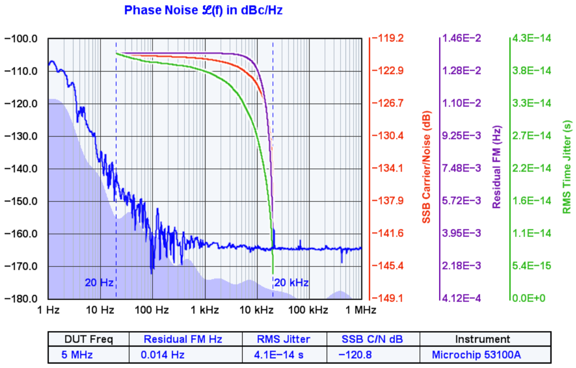
Thanks to the magic of cross-spectrum averaging, the PhaseStation 53100A can measure exceptionally low levels of phase noise and AM noise on HF and VHF signals. Below are some guidelines to help you make the most of this capability.

Integrated noise and jitter measurement

You can display various measures of integrated phase noise between two arbitrary offsets by enabling any or all of the following fields in the **Legend→Select** menu:

- **Residual FM**
- **RMS Integrated Noise (Degs)**
- **RMS Integrated Noise (Rads)**
- **RMS Time Jitter**
- **SSB Carrier/Noise**

When one or more of these parameters is enabled for display in the legend table, two “spot cursors” will appear in blue, as shown at right. You can move these lower and upper limit cursors with **Ctrl-left click** and **Ctrl-right click**, respectively.



DUT Freq	Residual FM Hz	RMS Jitter	SSB C/N dB	Instrument
5 MHz	0.014 Hz	4.1E−14 s	−120.8	Microchip 53100A

Figure 27: Measuring jitter from 20 Hz - 20 kHz with RMS noise integration and *Display→Show integrated PN traces*

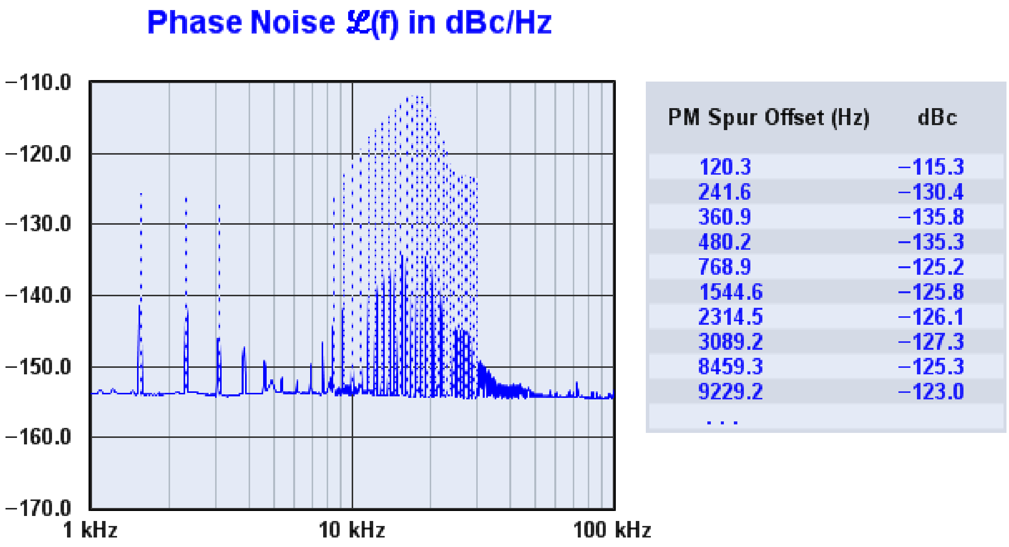
To gain insight regarding the integrated jitter and noise distribution versus offset frequency, use **Display→Show integrated PN traces** in conjunction with one or more of the above parameter selections in the **Legend→Select** menu. Color-coded traces will be overlaid on the graph that correspond to the integrated noise levels that would be reported if the lower limit of integration were moved right-to-left along the X axis, starting with the upper limit offset and progressing toward the user-specified lower limit cursor location.

TimeLab does not make a distinction between random and deterministic jitter, so any spurs in phase noise plots are “integrated” as if they contained only random noise. Consequently, when viewing any of the RMS integrated noise values, you should use either **Trace→Smooth noise traces** (Ctrl-w) or **Trace→Suppress spurs** (Ctrl-s) to remove as much coherent energy as possible from the phase noise trace. Otherwise, time jitter and integrated-noise readings may be artificially exaggerated.

Along with many other sources, MT-008: Converting Oscillator Phase Noise to Time Jitter from Analog Devices provides a concise summary of the math behind phase-noise integration.

The spur table

The *spur table* is a list of all coherent amplitude- or phase-modulated spectral components detected by the instrument in the course of measuring AM noise or phase noise, respectively. Each plot acquired from an instrument that supports spur detection has its own spur table. Spur levels are reported as CW power levels relative to the carrier, each at a given offset from the carrier frequency.



PM Spur Offset (Hz)	dBc
120.3	−115.3
241.6	−130.4
360.9	−135.8
480.2	−135.3
768.9	−125.2
1544.6	−125.8
2314.5	−126.1
3089.2	−127.3
8459.3	−125.3
9229.2	−123.0
. . .	

Figure 28: Typical phase noise measurement with accompanying spur table

The default **Spur Threshold** value of 6 dB, specified in the **Additional options** page of the 53100A acquisition dialog, ensures that artifacts of lower amplitude relative to neighboring areas of the trace are not classified as spurs. To illustrate the difference between coherent spur amplitudes and their apparent levels on the 1-Hz normalized noise trace in the figure above, **Trace→Suppress spurs in noise traces (Ctrl-s)** is disabled, and **Trace→Mark spurs in noise traces (Ctrl-m)** is turned on.

Show or hide known spurs

Several commands in TimeLab determine how spurs are handled in phase noise and AM noise graphs:

- **Trace→Mark spurs in noise traces** (Ctrl-m**)** draws shaded or dashed lines, depending on the **Trace→Toggle trace thickness for current measurement** (T**)** setting, for each detected spur on the AM or phase noise graphs. These lines reflect the amplitude of each spur as reported in the spur table, *which will not be the same* as the level of any corresponding bumps or spikes in the noise trace itself. Because phase noise and AM noise traces undergo normalization to 1 Hz bandwidth, the Y-axis labels don't reflect the true amplitude of any coherent trace features. They must be recognized as spurs, and their amplitudes indicated separately.

- **Trace→Suppress spurs in noise traces (**Ctrl-s**)** attempts to remove known spurs from phase noise and AM noise traces. As noted above, spurs that appear as artifacts in noise traces are not rendered at their true amplitude levels due to the use of FFT noise-bandwidth normalization. If you are concerned with spur amplitudes in a given measurement, you should use **Trace→Suppress spurs in noise traces (**Ctrl-s**)** to erase each detected spur from the graph and **Trace→Mark spurs in noise traces (**Ctrl-m**)** to replace them with vertical lines showing their true CW amplitude levels relative to the carrier.

- **Trace→Smooth noise traces (**w**)** always implies **Trace→Suppress spurs in noise traces (**Ctrl-s**)**. For better visual quality, TimeLab will attempt to erase any known spurs prior to smoothing the trace.

- **Display→Numeric table (**Ctrl-n**)** can be used to show or hide the spur chart for the selected plot. Like the numeric charts and tables associated with other measurement types, the spur chart appears to the right of the graph area. The Ctrl-n command toggles table visibility for all measurement types, not just the currently visible one.

Spur measurement options

The **Additional options** page in the PhaseStation 53100A's acquisition dialog contains four controls that influence how spurs are recognized and recorded.

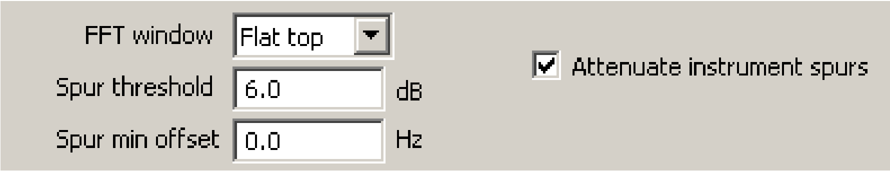

FFT window: Flat top
Spur threshold: 6.0 dB
Spur min offset: 0.0 Hz
☑ Attenuate instrument spurs

Figure 29: Window and spur controls

- The choice of **FFT Window** implies a compromise between the accuracy at which a given spur's offset frequency and amplitude are reported. The default **Flat top** window excels at measuring spur amplitude levels, but the **Hann** window offers better frequency resolution, and may improve the odds of detecting a given spur as well.

- The **Spur threshold** field defaults to 6.0 dB. It defines the amplitude level, relative to the average level of nearby FFT bins, below which a given bin's amplitude will not be considered indicative of a coherent spur. Lowering the threshold to 3 dB can sometimes be helpful when tight mask-test requirements need to be met.

- **Spur min offset** is normally 0.0 Hz. It can be set prior to acquisition by entering an offset frequency below which spurs will not be logged for processing. The minimum spur offset can also be adjusted in the **Edit→Trace properties** dialog at any time after acquisition is complete. This feature can be especially helpful when large 'bumps' in the flicker noise region are incorrectly classified by TimeLab as either real spurs or instrument spurs. 1 Hz is a reasonable value to start with in these cases.

- When the **Attenuate instrument spurs** option is checked, the 53100A driver software will use the phase component of a given spur's peak FFT bin to attempt to classify it as an instrument artifact or a genuine result of the measurement. Instrument spur discrimination is enabled by default and discussed in detail below.

Additionally, note that both spur-identification performance and acquisition time will be affected if the **Edit segment table** button is used to change the FFT kernel sizes used in the individual noise-trace segments. While lower-resolution segment definitions can result in faster measurements, they can also cause large elevated regions to appear where "forests" of discrete spurs are too closely-spaced to resolve.

Is it a spur, or isn't it?

Identifying genuine spurs while disregarding false ones is a difficult problem in the general case, and no spur-discrimination algorithm is infallible. The 53100A's DSP software may occasionally fail to detect spurs when they occur in clusters near the FFT segment's resolution limits, in steeply-sloped areas of the noise trace, or in other parts of the trace where an obvious baseline is not clearly discernible.

You can lower the **Spur Threshold** for a given measurement, but the risk of misclassifying random noise spikes as coherent spurs will become greater. When you see random spurs appear and disappear from the spur chart over the course of a measurement, you can increase the **Spur Threshold**, try changing FFT window types, or simply wait for the measurement to run long enough to reduce the trace variance ("grass").

As mentioned above, you can also tell TimeLab to ignore spurs close to the carrier with the **Min Spur Offset** field in the **Edit→Trace Properties** dialog. This option can also be specified at acquisition time in the 53100A acquisition dialog's **Additional options** page.

Understanding instrument spurs

While TimeLab can sometimes overlook true spurs and report nonexistent ones, a third type of measurement error may need to be considered as well: the presence of spurs which originate within the instrument itself.

The spur-free dynamic range (SFDR) of the ADCs used in the 53100A is approximately 100 dB, which determines the corresponding specification limit of the instrument as a whole. In principle, your noise plots may contain fictitious spurs as high as -100 dBc, but the 53100A's typical performance is much better than this. Any spurs with amplitudes above -120 dBc that remain visible as the measurement converges are likely to be "real" in the sense that they originate from the DUT, from the reference, or from environmental RFI.

Regardless of the equipment used, spur-free observations of AM noise and phase noise become challenging below -130 dBc. There are simply too many potential sources of coherent artifacts to rule out in most cases. Still, spurs that originate within the 53100A can sometimes be identified manually. An informal technique that may help determine whether a given spur or spur cluster originates within the 53100A hardware itself is illustrated in the side-by-side plots below.

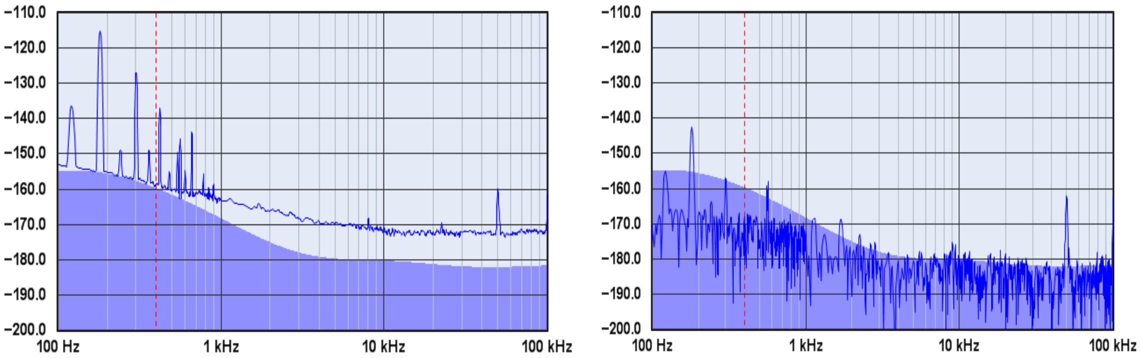


Figure 30: Imaginary and real components of the cross-correlated noise spectrum

At left, the phase noise of a crystal oscillator has been measured over the course of several hours. Most of the visible spurs are below the 53100A's specified limits, so it isn't clear which ones the instrument may be responsible for.

However, because complex signals are used throughout the 53100A's DSP pipeline, we can take advantage of a mathematical observation.[12] While the cross spectrum of a physically-realizable signal should fall on the real axis of the Argand plane, at least some instrument spurs are caused by data-conversion errors that can show up elsewhere in

[12] See Rubiola and Vernotte, The cross spectrum experimental method, and Grove *et al.*, Direct-digital phase noise measurement

phase space. Like random ADC noise, the numerical distribution of these errors is not limited to the real axis… but unlike random noise, the real amplitude of some of these spurs may not average to zero over time. These spurs appear on the graph as instrument artifacts.

Switching to the **Trace→Show imaginary part of cross spectrum display (**Ctrl-F3**)** view at right reveals the noise and spurs that correspond to the imaginary part of the cross-spectrum average, specifically the absolute value of the Q-component at each bin. Ordinarily this data is used only to create the shaded area used by the **Trace→Show estimated instrument noise (**F2**)** feature, after undergoing heavy averaging. But Ctrl-F3 lets us view the imaginary part of the cross spectrum at full resolution. It's readily apparent that the spur near 50 kHz has *almost the same amplitude in the imaginary cross spectrum as it does in the real measurement.*

This is a strong indicator that the spur is an instrument artifact. When **Attenuate instrument spurs** is selected in the 53100A acquisition dialog, spurs with high imaginary amplitude components are removed from the plot regardless of the state of all other smoothing and spur-suppression options.

There are other spurs in the imaginary cross spectrum that correspond to similar spurs in the real plot, but most appear to be AC line harmonics that both the 53100A and the sources under test are exposed to, and their amplitudes are substantially different. Although the Ctrl-F3 technique isn't foolproof, it can offer some valuable insight when you need to account for unanticipated low-level spurs.

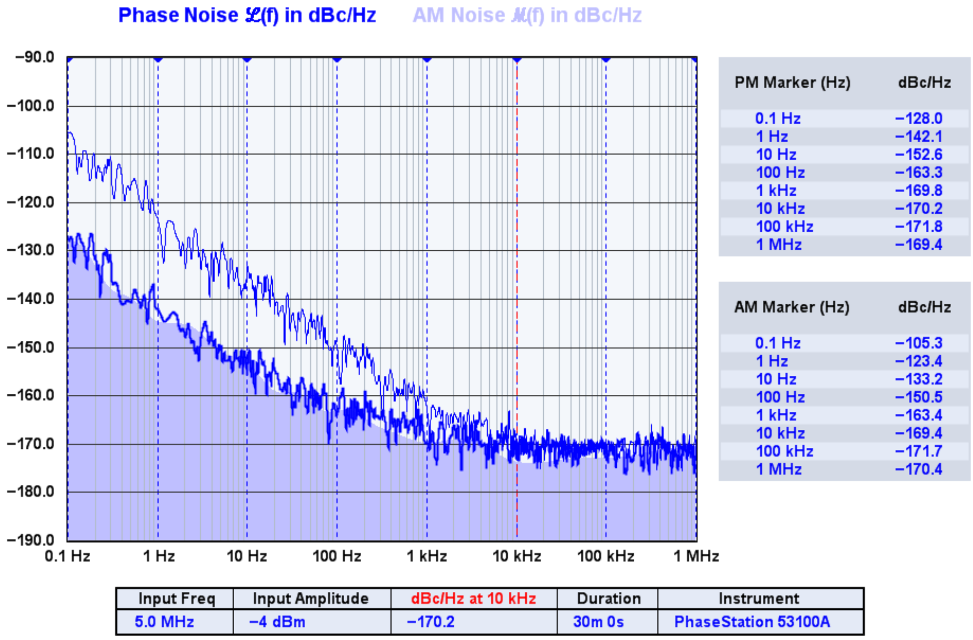
Manual spur removal

As noted above, instrument spurs can be overlooked by the software for several reasons. They may be below the spur-detection amplitude or Fourier offset thresholds specified in the instrument's acquisition options dialog (where applicable), or they may be too low in amplitude to be recognized as a spur at all. At the same time, a spur that *is* recognized may not be correctly flagged and removed as an instrument spur if its phase lies close to the real axis.

These cases often resolve themselves if the measurement is allowed to run long enough for the SNR to improve. If not, the **Edit→Flatten selected noise region (**Ctrl-F8**)** command can help in particularly stubborn cases – for instance, when an unrecognized artifact must be removed in order to pass a mask test, or when a trace with excessive noise variance needs to be cleaned up for publication. Refer to the **Edit→Flatten selected noise region** command description for more information.

Using noise markers

To add up to 10 noise markers at specified offset frequencies, hold down the **Shift** key and **left-click** on the graticule area of the noise plot near the desired column. A marker will appear at the top of the graticule area. If the selected plot includes valid trace data at the specified offset frequency, a corresponding entry will appear in the noise marker table at right, displaying the offset frequency and noise power at that point in the trace.



PM Marker (Hz)	dBc/Hz
0.1 Hz	−128.0
1 Hz	−142.1
10 Hz	−152.6
100 Hz	−163.3
1 kHz	−169.8
10 kHz	−170.2
100 kHz	−171.8
1 MHz	−169.4

AM Marker (Hz)	dBc/Hz
0.1 Hz	−105.3
1 Hz	−123.4
10 Hz	−133.2
100 Hz	−150.5
1 kHz	−163.4
10 kHz	−169.4
100 kHz	−171.7
1 MHz	−170.4

Input Freq	Input Amplitude	dBc/Hz at 10 kHz	Duration	Instrument
5.0 MHz	−4 dBm	−170.2	30m 0s	PhaseStation 53100A

Figure 31: Noise markers

To remove an existing noise marker, simply **shift-left click** near the marker's column on the screen. All of the currently-defined markers may be cleared with **Display→Clear noise markers (**Ctrl-F4**)**.

The visibility of the noise marker table can be toggled with **Display→Show numeric tables (**Ctrl-n**)**. If the noise marker table is not visible despite being enabled, ensure that at least one marker is placed at an offset that lies within the limits of the selected trace. Markers located at offsets where no trace data exists will be shown on the graticule, but will not appear in the table. When no noise markers are eligible for display, the table itself will not be rendered.

Hints for noise measurements

- The 53100A can render an estimate of its own noise floor in AM/PM noise plots if you enable **Trace→Show estimated instrument noise (**F2**)**. See page 101 for more information.

- For AM and phase noise, a good indication that the noise trace has converged on its final level is the variance of the trace, or its overall fuzziness/thickness. If you measure a conventional signal generator or noisy oscillator, you'll notice that the trace stops falling relatively quickly after the measurement begins, and then becomes smoother over time. This 'smoothing' effect is an important cue that tells you how trustworthy your noise measurement is. With quiet sources, close-in segments can take an especially long time to converge at a final, smoothed level, since they are updated much less often.

- Crosstalk and AC power-line interference can create discrete spurs on phase noise and AM noise graphs, just as they can degrade ADEV and other statistical plots. See page 51 for a discussion of causes and cures.

- TimeLab has a graph-smoothing feature (**Trace→Smooth noise traces (**ctrl-w**)**) but it's not always needed with a cross-correlating analyzer such as a PhaseStation, TimePod, or Microsemi 5120A/5125A. The best way to obtain a smoother trace with these instruments is often to let your measurement run longer. Low-noise plots can benefit from **Trace→Smooth noise traces (**ctrl-w**)**, but many measurements don't need additional smoothing at all.

- Another tip for smoother phase noise and AM noise traces, suitable for 53100A users with faster PCs: enable the **Overlapped acquisition** option in the **Additional options** page of the acquisition dialog. Overlapped acquisition provides much more efficient utilization of incoming data by the FFT routines, but requires significantly more CPU time. (Note that FFT overlapping is different from the algorithm used in overlapped Allan deviation and other stability statistics. Overlapped statistical calculations are always performed, while overlapped FFT processing in noise data acquisitions is turned off by default.)

- When overlaying several measurements on the screen, it can be helpful to run with spurs removed from the trace (**Trace→Suppress spurs (**Ctrl-s**)** enabled) and unmarked on the graph (**Trace→Mark spurs (**Ctrl-m**)** disabled). **Trace→Smooth noise traces (**ctrl-w**)** will automatically suppress spurs.

TimeLab Command Reference

Virtually all of the menu-based commands in TimeLab also have single-key shortcuts for faster, more interactive workflow. When present, the shortcut key is generally shown after the name of the menu option.

File menu

The File menu helps you work with data files and images. Supported actions include opening, saving, closing, and deleting plots stored in .TIM files, saving image files, printing or copying the currently displayed window contents, and importing and exporting raw measurement data. Various "user preference" toggles are also provided on this menu.

Load .TIM file (l)

Loads a saved .TIM file. .TIM files are ASCII text files that contain all measurement data acquired or imported by TimeLab, together with any applicable metadata. Specific fields and tables within each .TIM file may vary depending on the measurement type(s) saved in the file.

For frequency stability measurements made with counters or timing analyzers, the .TIM file contains the phase record obtained from the instrument during acquisition. The phase record may have been derived from a series of frequency readings or other raw data stream formats. It may also have been decimated to conform to a specified measurement rate or bandwidth.

For phase noise and AM noise measurements, the .TIM file contains the FFT bin contents needed to render the applicable graphs.

Save image or .TIM file (s)

This command can either save the selected plot as a .TIM file, or save the current contents of the TimeLab window as an image file in .PNG, .GIF, .TGA, .PCX, or .BMP format.

By default, .TIM files will be saved. To save an image file, simply add one of the suffixes above when you enter the filename. *Keep in mind that image files do not preserve your original measurement data, and cannot be reloaded into TimeLab.* If a trace caption was either provided at acquisition time or added with **Edit→Trace properties (e)**, it will be used as the default filename.

Generally, the .PNG format is best for saving images from TimeLab for email attachment or inclusion in other documents. .GIF and .PCX files will discard some color information, while .TGA and .BMP files are unnecessarily large.

When saving an image, all visible status and error messages, notifications, prompts, and mouse-cursor query values are removed. The black-triangle cursor associated with the currently selected plot will also be omitted from saved images.

Copy image to clipboard (Ctrl-c)

This command offers a handy shortcut for cutting/pasting TimeLab screen images into other programs via the operating system's clipboard, avoiding the need to save an image file.

When copying the screen image to the clipboard, all visible status and error messages, notifications, prompts, and mouse-cursor query values are removed. The black-triangle cursor associated with the currently selected plot will also be omitted from the copied image.

Use caution when pasting images from the clipboard into email messages. Some clients may encode them in uncompressed form, resulting in messages too large to email reliably. When in doubt, consider attaching a saved .PNG file instead.

Import .RES or .PNP phase noise data (N)

This command can be used to import a .RES file created by the HP 3048A, an industry-standard phase noise analysis system, or a .PNP file saved by PN.EXE, a popular freeware phase-noise measurement application from the KE5FX GPIB Toolkit.

The latter option supports phase noise plots from various RF spectrum analyzers not otherwise recognized by TimeLab.

Import ASCII phase or frequency data (L)

Reads an ASCII text file containing phase-difference, frequency, frequency-difference, or timestamp data, one entry per line. The imported data is converted to a standard TimeLab phase record, and may be rendered, manipulated, and saved as a .TIM file.

As with other acquisition dialogs, TimeLab's data-import dialog provides detailed mouseover help text for all fields. Refer to this help text for usage information.

To "watch" a continuously updated data file, use **Acquire→Acquire from live ASCII file** instead.

Export ASCII phase data (x)

This command is supported only when viewing the selected plot's phase record contents using **Measurement→Phase difference** (p) or **Measurement→Frequency-difference** (f). It will save an ASCII text file containing phase difference data in seconds, one entry per line, at 16 digits of precision to the right of the decimal point. When working with cross ADEV plots, both primary and secondary phase records may be exported if required.

If a selected or zoomed area has been defined by dragging with the mouse, only that portion of the phase record will be exported.

By default, the exported phase data will be saved with the suffix .CSV for ease of reading into Excel and other spreadsheets. Exported data may be re-imported into TimeLab, if necessary, by the **File→Import ASCII phase or frequency data** command. Select "All files" as the file type if the data was not exported with the suffixes .TXT or .DAT, then select **Phase difference (sec)** as the file data type.

Optionally, each sample in the ASCII phase data file can be timestamped with its absolute MJD or the number of elapsed seconds since the beginning of the phase record.

Export ASCII frequency data (X)

This command is supported only when viewing the selected plot's phase record contents using **Measurement→Phase difference** (p) or **Measurement→Frequency-difference** (f). It will save an ASCII text file containing absolute frequency readings in hertz, one entry per line, at 16 digits of precision to the right of the decimal point.

If a selected or zoomed area has been defined by dragging with the mouse, only that portion of the phase record will be exported.

By default, the exported frequency readings will be saved with the suffix .CSV for ease of reading into Excel and other spreadsheets. Exported data may be re-imported into TimeLab, if necessary, by the **File→Import ASCII phase or frequency data** command. Select "All files" as the file type if the data was not exported with the suffixes .TXT or .DAT, then select **Frequency (Hz)** as the file data type.

Optionally, each sample in the ASCII phase data file can be timestamped with its absolute MJD or the number of elapsed seconds since the beginning of the phase record.

Export binary phase data

This command is supported only when viewing the selected plot's phase record contents using **Measurement→Phase difference (**p**)** or **Measurement→Frequency-difference (**f**)**. It will save a binary file containing phase difference data in seconds, represented as a block of double-precision IEEE 754 values in little-endian (Intel) format.

If a selected or zoomed area has been defined by dragging with the mouse, only that portion of the phase record will be exported.

As with **File→Export ASCII phase data (**x**)**, the default file suffix is .CSV. This is usually inappropriate for binary data; you should specify a suffix appropriate to your application. Exported binary phase data may not be re-imported into TimeLab.

Export phase data to Stable32 (Ctrl-x)

This command is useful for transferring data from the selected plot's phase record directly into a third-party analysis application such as Stable32 from Hamilton Technical Services. Stable32 is a popular application for offline data analysis. It supports many statistical features and display options beyond the limited set implemented by TimeLab.

This command works by exporting the selected plot's phase record to a temporary file using the same functionality as the **File→Export ASCII phase data (**x**)** command. If a selected or zoomed area has been defined by dragging with the mouse, only that portion of the phase record will be exported.

When the command is issued, a dialog box allows you to specify the location of the Stable32 executable or other third-party program, as well as the command line to be passed to it. The command line may contain various predefined variables described in the dialog box's help text, including placeholders for the name of the temporary file and the data rate (tau zero) in samples per second. Upon pressing the **Launch Stable32** button, the temporary file will be generated and the specified program launched. TimeLab will then continue to run normally.

Temporary files created by TimeLab are cleaned up when TimeLab exits, so it's advisable to leave TimeLab running until after the third-party analysis application has exited.

Export ASCII xDEV trace

Available in the **Measurement→Allan Deviation (**a**), Measurement→Modified Allan Deviation (**m**), Measurement→Hadamard Deviation (**h**),** or **Measurement→Time Deviation (**t**)** views, this command creates an ASCII text file containing a *tau, sigma(tau)* value pair for each visible xDEV bin in the selected plot. One value pair is written per line; each value is rendered with 16 digits of precision to the right of the decimal point.

The number and distribution of xDEV bins may be specified prior to acquisition with the **Bin Density** field in the **Frequency Stability** tab of the acquisition dialog. Additionally, the selected plot's bin density may be adjusted after acquisition through the **Edit→Trace properties** dialog. See the help fields in these dialogs for details.

By default, the exported *tau, sigma(tau)* pairs will be saved with the suffix .CSV for ease of reading into Excel and other spreadsheets.

Export ASCII AM/PM noise trace

Export ASCII integrated PN data

Available in the **Measurement→Phase noise (**P**)** and **Measurement→AM noise (**A**)** views, the **File→Export ASCII AM/PM noise trace** command creates an ASCII text file containing *offset, dBc/Hz* values corresponding to the visible noise trace for the selected plot. One value pair is written per line; each value is rendered with 16 digits of precision to the right of the decimal point. Because the data is obtained from the displayed graph, if you zoom in on a portion of the phase noise or AM noise graph, only the trace data corresponding to the visible area of the selected plot will be exported. Conversely, if the graticule's X-axis boundaries extend beyond the minimum or maximum offset for the selected plot, no *offset, dBc/Hz* pairs outside the valid range will appear in the exported file.

By default, the exported *offset, dBc/Hz* pairs will be saved with the suffix .CSV for ease of reading into Excel and other spreadsheets. (If TimeMonitor headers have been enabled, the default suffix will be .TXT instead.)

Note that when **Show imaginary part of cross spectrum (**Ctrl-F3**)** is active, the **File→Export AM/PM noise trace** command will export the imaginary cross spectrum trace data rather than the normal AM noise or phase noise trace.

File→Export ASCII integrated PN data operates in a similar fashion, but exports one or more integrated phase noise traces rather than the standard L(f) phase noise trace. See page 67 for more details.

Export AM/PM spur table

Available in the **Measurement→Phase noise (**P**)** and **Measurement→AM Noise (**A**)** views, this command creates an ASCII text file containing *offset, dBc* values corresponding to entries in the spur table for the selected plot. One value pair is written per line; each value is rendered with 16 digits of precision to the right of the decimal point.

See "Spurs" on page 74 for more about spur detection and measurement.

By default, the exported *offset, dBc* pairs will be saved with the suffix .CSV for ease of reading into Excel and other spreadsheets.

Exported ASCII files include headers for Microsemi TimeMonitor

When enabled, this option causes text files written by **Export ASCII xDEV trace**, **Export ASCII AM/PM noise trace**, **Export ASCII frequency data**, and **Export ASCII phase data** to be prefixed with a comment header that will be recognized by the Microsemi TimeMonitor application.

The default filename suffix for exported ASCII files will be .TXT when this option is active, rather than .CSV. By default, this option is turned off.

Print image

After displaying a standard dialog box for printer selection and configuration, this command prints the current contents of the TimeLab window.

The printed image will not contain any visible status or error messages, notifications, prompts, or mouse-cursor query values. The black-triangle cursor associated with the currently selected plot will also be omitted.

Scale file dialogs by window size

When this option is enabled, common file dialogs in Windows XP will be centered in the TimeLab window and resized proportionally. In later Windows versions, the option has no effect. (Default: disabled)

Confirmation prompts default to Yes

When this option is enabled, the default button in most confirmation prompt dialog boxes will be 'Yes' rather than 'No.' (Default: disabled)

Limit MTIE processing to 10K points

When this option is enabled, MTIE calculations will stop after the first 10,000 points in the phase record have been processed. This helps avoid excessively-long trace refresh periods when loading .TIM files or refreshing plots. (Default: enabled)

Warn before exiting with unsaved plots

If enabled, this option causes TimeLab to prompt for confirmation before exiting if any plots have been acquired or edited but not saved. (Default: enabled)

Warn before retriggering measurement in progress

If enabled, this option causes TimeLab to prompt for confirmation before acting on the **Acquire→Keep and retrigger selected acquisition (Ctrl-k)** or **Acquire→Abort and retrigger selected acquisition (Ctrl-a)** commands. (Default: enabled)

Reset all parameters, options, and settings at next startup

This option causes a dialog box to appear when you exit from TimeLab, confirming that you would like to reset all options, settings, colors, and dialog fields to their default values the next time the program is launched. If confirmation is given, the .INI files associated with TimeLab and all of its instrument drivers will be deleted, forcing them to be recreated with default values when the program is relaunched. (Default: disabled)

If desired, the .INI files used by TimeLab and its instrument drivers may be copied, backed up, or edited manually. The location of the directory where TimeLab stores its .INI files varies depending on the operating system version. In Windows 7, these files are normally located at **c:\Users\<*user name*>\AppData\Local\Miles Design\TimeLab**, also accessible at **%LOCALAPPDATA%\Miles Design\TimeLab**. In Windows XP, the .INI files are normally located in the hidden directory **c:\Documents and Settings\<*user name*>\Local Settings\Application Data\Miles Design\TimeLab**.

Note that TimeLab does not generally store configuration data in the Windows registry or anywhere else besides its .INI files; only installation-related data is stored in the registry.

Close selected plot (Del)

Closing the selected plot frees the memory it occupies and removes it from the graph and legend table. If the plot is associated with a saved .TIM file, the file is not deleted.

After closing a plot, the next plot in the legend table, if any, becomes the selected plot. Up to 9 plots may be loaded into memory at once; to acquire, load, or import additional plots, you must close at least one.

Close all visible plots (Home)

Closes all loaded plots at once, removing them from the graph and legend table.

Delete selected plot’s .TIM file (Ctrl-Del)

After prompting for confirmation, this command unloads the selected plot’s data from memory and also deletes its associated .TIM file, if any.

After closing a plot, the next plot in the legend table, if any, becomes the selected plot. Up to 9 plots may be loaded into memory at once; to acquire, load, or import additional plots, you must close at least one.

Quit (q or Esc)

Exits from TimeLab. Depending on the setting of **File→Warn before exiting with unsaved plots**, a confirmation prompt may be issued.

Unless you hold the Shift key as the program exits, TimeLab will write all of its global settings and dialog fields (as well as a list of known **Legend** menu entries and currently loaded plots) to a file called **TIMELAB.INI**. These properties will be restored when the program is next launched. The “Shift-exit” feature can be handy if you have opened or closed various plots or changed any options or colors, but don’t wish to retain the changes or otherwise lose your previous startup defaults.

Note that acquisition and file-import dialog field contents are not backed up in **TIMELAB.INI**. The most recently used dialog entries associated with these operations are stored in instrument-specific .INI files. The instrument driver updates its .INI file as soon as you select **Start Measurement** or otherwise initiate an acquisition or import operation, not when the program exits.

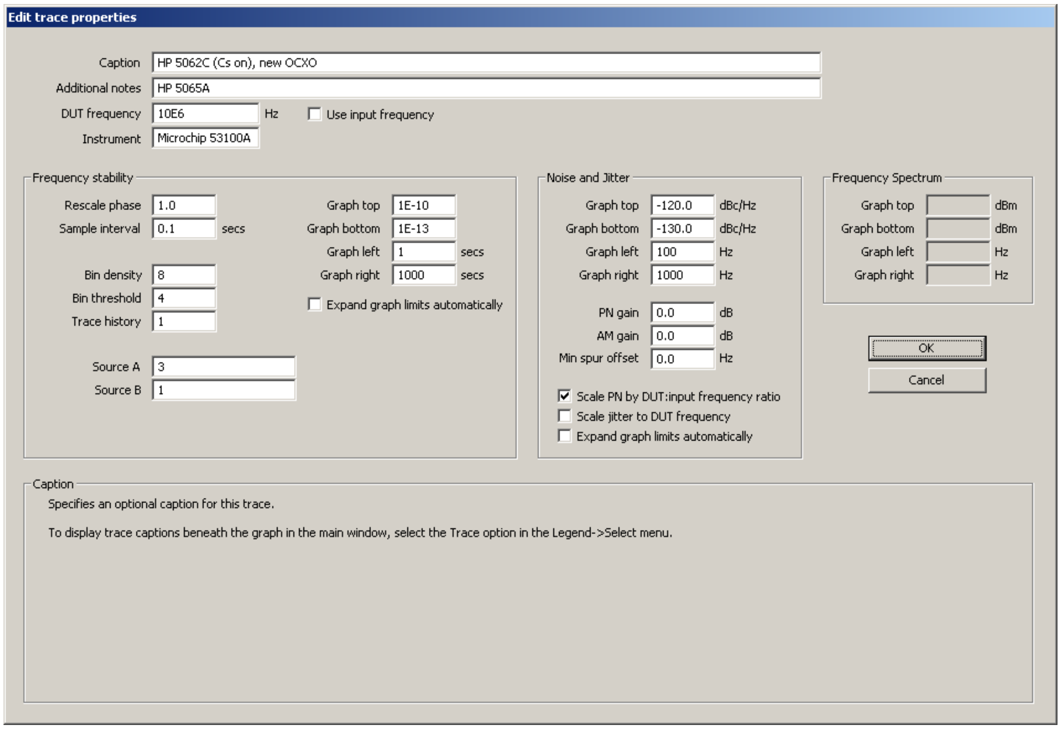
If desired, the .INI files used by TimeLab and its instrument drivers may be copied, backed up, or edited manually. The location of the directory where TimeLab stores its .INI files varies depending on the operating system version. In Windows 7, these files are normally located at **c:\Users\*<user name>*\AppData\Local\Miles Design\TimeLab**, also accessible at **%LOCALAPPDATA%\Miles Design\TimeLab**. In Windows XP, the .INI files are normally located in the hidden directory **c:\Documents and Settings\*<user name>*\Local Settings\Application Data\Miles Design\TimeLab**. TimeLab does not generally store configuration data in the Windows registry or anywhere else besides its .INI files; only installation-related data is stored in the registry.

After updating **TIMELAB.INI** at program termination, any temporary files that may have been created by instrument drivers or the **File→Export phase data to Stable32 (**Ctrl-x**)** command are deleted. For debugging purposes, temporary-file cleanup is also inhibited if you hold down the Shift key while the program terminates.

Edit Menu

Plots that have been loaded or acquired in TimeLab may be displayed with a variety of options and transforms. However, any global changes to the measurement data or display parameters associated with a specific plot must be made through the commands in the Edit menu.

Trace properties (e)



Edit trace properties

Caption: HP 5062C (Cs on), new OCXO

Additional notes: HP 5065A

DUT frequency: 10E6 Hz   □ Use input frequency

Instrument: Microchip 53100A

Frequency stability

Rescale phase: 1.0

Sample interval: 0.1 secs

Bin density: 8

Bin threshold: 4

Trace history: 1

Source A: 3

Source B: 1

Graph top: 1E-10

Graph bottom: 1E-13

Graph left: 1 secs

Graph right: 1000 secs

□ Expand graph limits automatically

Noise and Jitter

Graph top: -120.0 dBc/Hz

Graph bottom: -130.0 dBc/Hz

Graph left: 100 Hz

Graph right: 1000 Hz

PN gain: 0.0 dB

AM gain: 0.0 dB

Min spur offset: 0.0 Hz

☑ Scale PN by DUT:input frequency ratio

□ Scale jitter to DUT frequency

□ Expand graph limits automatically

Frequency Spectrum

Graph top: dBm

Graph bottom: dBm

Graph left: Hz

Graph right: Hz

OK

Cancel

Caption

Specifies an optional caption for this trace.

To display trace captions beneath the graph in the main window, select the Trace option in the Legend->Select menu.

Figure 32: Editing the properties of the selected trace

As seen above, the **Edit→Trace properties (e)** dialog lets you change various properties associated with the selected plot that determine how its measurement data is presented. These properties were assigned their initial values in the acquisition dialog before the measurement began.

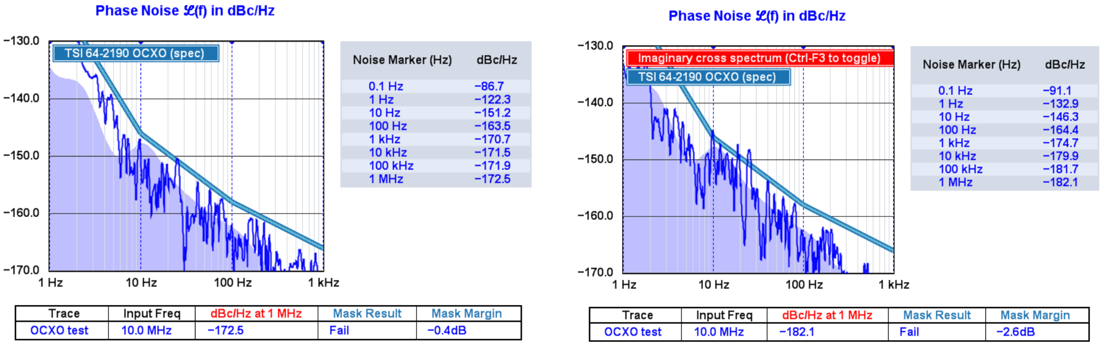
Like the various instrument-specific dialogs on the **Acquire** menu, the trace-properties dialog will typically vary from the image shown above. The dialog has extensive "mouseover help" text that is maintained alongside TimeLab itself, so the individual fields will not be described here at length. Simply position your cursor over each field to learn more about it.

Some areas of the dialog may be disabled depending on the data type(s) available in the selected plot, as well as on the status of particular dialog controls. Also, certain properties – including the controls in the **Frequency Stability** area that determine the bin configuration for Allan deviation and other statistical measurements -- cannot be edited while acquisition is in progress. If you didn’t enter the desired values in the acquisition dialog before data collection began, you’ll need to wait for the measurement to finish before you can change them.

Flatten selected noise region (Ctrl-F8)

This feature allows you to remove known spurs and other artifacts from a phase noise or AM noise plot. It can be helpful in cases where the automated spur identification process falls short, such as when a PN or AM noise mask test is failing due to a low-level instrument artifact or a known external interference source.

Figure 33 below shows a phase noise mask test that has failed with a -0.4 dB margin due to an unrecognized spur at an offset near 25 Hz. This spur is too close to the surrounding noise to be flagged and suppressed by TimeLab, either as a real spur or an instrument artifact, but it's strong enough to exceed the limit line and cause a test failure. Pressing Ctrl-F3 (**Trace→Show imaginary part of cross spectrum**) reveals that the spur is a low-level instrument artifact. It is not only visible in the imaginary part of the cross spectrum, but its amplitude is actually stronger there. This spur is a good candidate for manual removal.



Noise Marker (Hz)	dBc/Hz
0.1 Hz	−86.7
1 Hz	−122.3
10 Hz	−151.2
100 Hz	−163.5
1 kHz	−170.7
10 kHz	−171.5
100 kHz	−171.9
1 MHz	−172.5

Trace	Input Freq	dBc/Hz at 1 MHz	Mask Result	Mask Margin
OCXO test	10.0 MHz	−172.5	Fail	−0.4dB

Noise Marker (Hz)	dBc/Hz
0.1 Hz	−91.1
1 Hz	−132.9
10 Hz	−146.3
100 Hz	−164.4
1 kHz	−174.7
10 kHz	−179.9
100 kHz	−181.7
1 MHz	−182.1

Trace	Input Freq	dBc/Hz at 1 MHz	Mask Result	Mask Margin
OCXO test	10.0 MHz	−182.1	Fail	−2.6dB

Figure 33

Figure 34 at right illustrates the workflow: after switching back to the real cross spectrum with Ctrl-F3 and pressing F2 to hide the noise-floor estimate for better visibility (**Trace→Show estimated instrument noise**), simply drag a box to zoom in on a portion of the plot containing the base of the spur in question. Then, hold down the Shift key and left-drag to create a flat line segment that spans the area to be flattened. Choose endpoints that will blend effectively with the areas of the trace to be retained.

Once the segment has been drawn in the desired location, use Ctrl-F8 (**Edit→Flatten selected noise region**) to create a trace segment between the specified endpoints. Right-click anywhere in the window to zoom back out, and observe that the mask test now passes, as shown in Figure 35 below.

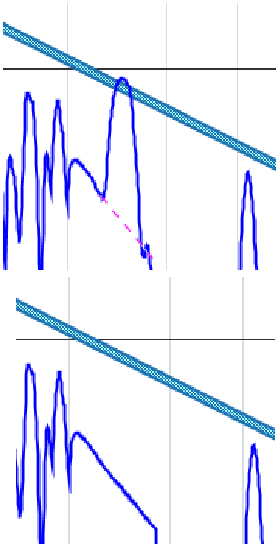
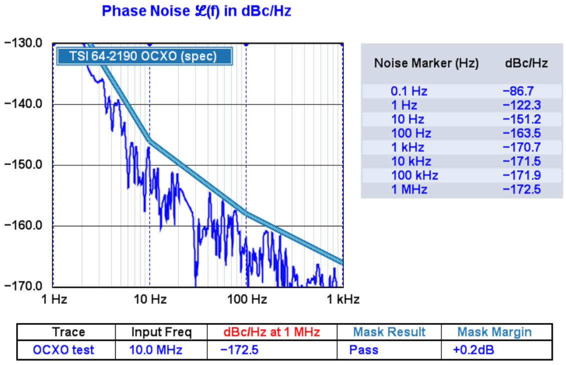


Figure 34



Noise Marker (Hz)	dBc/Hz
0.1 Hz	−86.7
1 Hz	−122.3
10 Hz	−151.2
100 Hz	−163.5
1 kHz	−170.7
10 kHz	−171.5
100 kHz	−171.9
1 MHz	−172.5

Trace	Input Freq	dBc/Hz at 1 MHz	Mask Result	Mask Margin
OCXO test	10.0 MHz	−172.5	Pass	+0.2dB

Figure 35

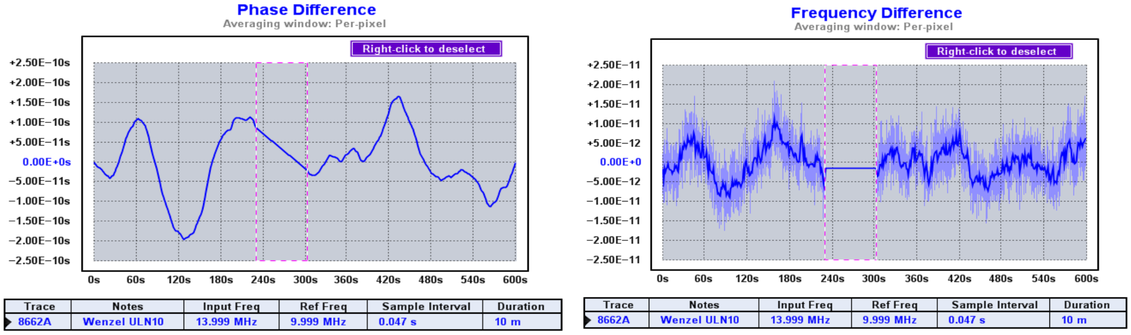
Internally, this command works by allocating an instrument spur record in the specified location. Up to 300 regions may be flattened, less the total number of real and instrument spurs in the measurement.

Ctrl-z (**Edit→Undo last flatten or subtract operation**) may be used to restore the most recently-flattened noise segment, while Ctrl-F9 (**Edit→Restore flattened noise regions**) will restore all manually-flattened segments.

Restore flattened noise regions (Ctrl-F9)

This command clears all flattened regions previously defined with **Edit→Flatten selected noise region (Ctrl-F8)**, restoring any manually-removed spurs. It cannot be undone.

Flatten selected or zoomed phase data (Ctrl-f)



Trace	Notes	Input Freq	Ref Freq	Sample Interval	Duration
8662A	Wenzel ULN10	13.999 MHz	9.999 MHz	0.047 s	10 m

Trace	Notes	Input Freq	Ref Freq	Sample Interval	Duration
8662A	Wenzel ULN10	13.999 MHz	9.999 MHz	0.047 s	10 m

Figure 36: Flattening a section of phase data

Above, the **Flatten selected or zoomed phase data (Ctrl-f)** command has bene used to replace the selected region of the phase record with a straight line. The result appears as a flat spot in the frequency difference view.

This command is supported only when viewing the selected plot's phase-record contents using **Measurement→Phase difference (p)** or **Measurement→Frequency difference (f)**. In addition, a region within the plot must first be specified. You can specify a region with a Shift-Left drag operation that leaves the region boundaries visible within the existing graph, as in the images above, or you can simply drag with the left mouse button to zoom in on the desired region. In either case, the command will replace the data within the specified region of the selected plot's phase record with a straight line segment that connects the data points at the region boundaries. As seen in the figures above, a straight line in the phase record corresponds to a flat region in the **Measurement→Frequency difference (**f**)** view.

You can use **Edit→Flatten selected or zoomed phase data (**Ctrl-f**)** to remove glitches and outliers from an acquisition. Subsequently, **Edit→Undo last flatten or subtract operation (**Ctrl-z**)** can be used to restore the flattened data. Unless you absolutely must preserve the original length of the phase record, though, it's better to use **Edit→Remove selected or zoomed phase data (**F4**)** in most cases.

No active background tasks may be under way for the selected plot at the time this command is issued. Active background tasks are those in which the plot's memory-resident phase record is being written, such as during initial acquisition and when reloading a saved .TIM file. If the phase record has finished loading but the xDEV statistical traces are still being refreshed, the refresh operation will be cancelled, then restarted after the command finishes.

Remove selected or zoomed phase data (F4)

Like **Edit→flatten selected or zoomed phase data (**Ctrl-f**)**, this command is useful for removing glitches and outliers from the selected plot's phase record. Instead of replacing the affected region with a straight line, however, **Edit→Remove selected or zoomed phase data (**F4**)** shortens the phase record by removing the data altogether.

This command has the same requirements and constraints as **Edit→Flatten selected or zoomed phase data (**Ctrl-f**)**, except that it cannot be undone with **Edit→Undo last flatten or subtract operation (**Ctrl-z**).** Consider saving your original data to a .TIM file first so that it can be recovered if necessary.

When removing data from the middle of the phase record, you'll be asked if you want to maintain phase continuity. In virtually all circumstances you should respond "Yes" to this prompt. TimeLab will then ensure a seamless transition between the two remaining regions by applying a constant offset to the phase data following the removed region. Otherwise, an abrupt phase change due to lost continuity may appear as a glitch in the data.

No active background tasks may be under way for the selected plot at the time this command is issued. Active background tasks are those in which the plot's memory-resident phase record is being written, such as during initial acquisition and when reloading a saved .TIM file. If the phase record has finished loading but the xDEV statistical traces are still being refreshed, the refresh operation will be cancelled, then restarted after the command finishes.

When the phase record associated with a cross ADEV measurement's primary or secondary plot is being altered by this command, the same region of data will automatically be removed from both of the phase records. See page 136 for more information on cross ADEV measurements.

Subtract global linear phase trend (frequency offset) (Ctrl-o)

Subtract global linear frequency trend (drift line) (Ctrl-l)

Subtract quadratic linear frequency trend (drift curve) (Ctrl-q)

These related commands are used to remove global offsets and trends from the selected plot's phase record. This is typically done to improve visibility of local trends and features whose magnitude is small compared to the Y-axis range. For detailed usage information and examples, as well as some alternative viewing methods that don't alter the phase record contents, see page 60.

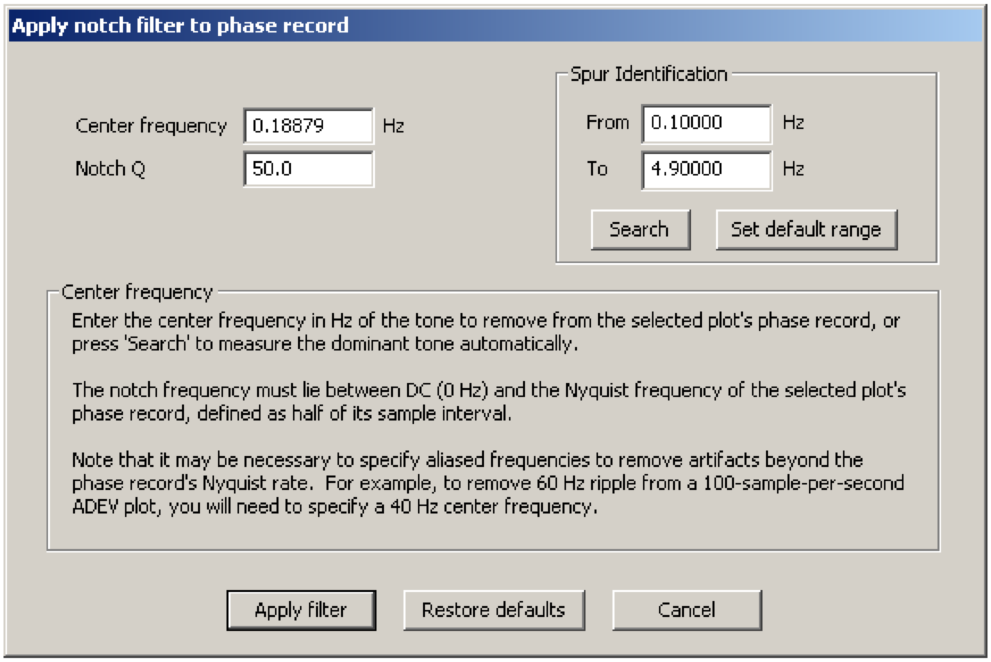
Two commands on the Trace menu, **Phase/frequency traces begin at zero (z)** and **Show linear phase/frequency residual (r)**, provide non-destructive functionality similar to the **Edit→Subtract global linear phase trend (frequency offset) (Ctrl-o)** command. See page 93 for more information.

If the selected plot has any background tasks in progress when any of these commands are issued, an error message will be displayed. Background tasks are those in which the plot's memory-resident phase record is being accessed by a background thread during initial acquisition, when reloading a saved .TIM file, and when reconstructing the xDEV statistical traces after a previous modification to the phase record.

Note that subtracting trends from data records that exhibit a large amount of variance relative to their overall slope – such as might be obtained from comparisons of GPSDO or atomic standards -- can result in numerically unstable outcomes that vary randomly based on the number of points in the data record. You may wish to toggle the subtraction operation with **Edit→Undo last flatten or subtract operation (Ctrl-z)** to verify that the command performed as expected.

Apply notch filter to phase data (Ctrl-n)

This command applies a biquad notch filter with the specified center frequency and Q parameters to the phase data record. It is typically used to attenuate unwanted tonal artifacts such as crosstalk and AC line interference. The filter has no effect on phase noise or AM noise data.



Center frequency: 0.18879 Hz

Notch Q: 50.0

From: 0.10000 Hz

To: 4.90000 Hz

Center frequency

Enter the center frequency in Hz of the tone to remove from the selected plot's phase record, or press 'Search' to measure the dominant tone automatically.

The notch frequency must lie between DC (0 Hz) and the Nyquist frequency of the selected plot's phase record, defined as half of its sample interval.

Note that it may be necessary to specify aliased frequencies to remove artifacts beyond the phase record's Nyquist rate. For example, to remove 60 Hz ripple from a 100-sample-per-second ADEV plot, you will need to specify a 40 Hz center frequency.

Figure 37: Notch filter properties

In the dialog box above, the desired center frequency and Q should be entered in the fields indicated.

The **Search** button can optionally be used to identify the dominant tone within the specified frequency range and copy it to the **Center frequency** field. When the **Set default range** button is pressed, the frequency range controls are prefilled with values based on the DC and Nyquist band limits for the selected plot's phase record, leaving 2% margins at either endpoint. Tone frequencies lying outside this range can be entered manually into the **Center frequency** field. Alternatively, the **From / To** fields can be adjusted manually to constrain the spur search within the desired limits.

When working with direct-digital test sets such as the PhaseStation 53100A and TimePod 5330A, line-frequency ripple can be particularly troublesome in ADEV plots captured at these instruments' default measurement bandwidth of 50 Hz. Since decimation to the specified ENBW occurs prior to phase detection at twice the final sample rate, or 200 Hz, any content at 60 Hz will appear as an alias at 40 Hz in the final 100 Hz phase data record. This frequency, rather than 60 Hz, should be entered in the Center Frequency field in order to suppress fundamental interference from ground loops in countries that utilize North American power distribution standards.

Note that AC line-frequency interference in 50 Hz environments may appear as long-term artifact(s) with a period close to DC. If this is suspected, consider using the 5 Hz or 500 Hz measurement bandwidths instead. The former setting will eliminate most AC line interference adequately, while the latter setting will allow it to be removed with the notch filter.

The effect of the notch filter can be reverted with the **Undo last flatten or subtract operation (**Ctrl-z**)** command.

Undo last flatten or subtract operation (Ctrl-z)

This command rolls back the most recent change to the selected plot performed by any of the following other commands:

- **Edit→Flatten selected noise region (**Ctrl-F8**)**
- **Edit→Flatten selected or zoomed phase data (**Ctrl-f**)**
- **Edit→Subtract global linear phase trend (frequency offset) (**Ctrl-o**)**
- **Edit→Subtract global linear frequency trend (drift line) (**Ctrl-l)
- **Edit→Subtract quadratic linear frequency trend (drift curve) (**Ctrl-q**)**
- **Edit→Trace properties (**e**) (**applies to **Rescale Phase** only**)**
- **Edit→Apply notch filter to phase record (**Ctrl-n**)**

A second “undo” command will redo the last-reverted action in all cases except for flattening operations in AM noise or phase noise plots. These commands must be reissued manually.

If the selected plot has any background tasks in progress when this command is issued, an error message will be displayed. Background tasks are those in which the plot’s memory-resident phase record is being accessed by a background thread during initial acquisition, when reloading a saved .TIM file, and when reconstructing the xDEV statistical traces after a previous modification to the phase record.

Trace Menu

The **Trace** menu contains options that affect the way measurement data is processed and rendered as traces on the graph.

Unlike the **Edit** menu options, the **Trace** options apply to all visible traces, rather than only the selected plot, and they never alter the underlying measurement data. Most options on the **Trace** menu apply only to a given measurement family – xDEV statistics, phase/frequency differences, or phase noise/AM noise measurements. For example, **Trace→Averaging window for phase/frequency traces** has no effect on phase noise or Allan deviation traces.

The majority of **Trace** options are simple on/off toggles. In the **Measurement→Phase difference** (p) display, for instance, applying the **Edit→Subtract global linear phase trend (frequency offset)** (Ctrl-o) command to each visible trace would have the same visible effect as **Trace→Show linear phase/frequency residual** (r), as discussed below. But **Trace→Show linear phase/frequency residual** (r) does not actually subtract the trend from the phase record. Issuing the command again will restore normal trace-rendering behavior.

Phase/frequency traces begin at zero (z)

Show linear phase/frequency residual (r)

These two commands can be used individually or together to maximize visible detail in phase/frequency plots.

When enabled, **Trace→Phase/frequency traces begin at zero** (z) causes all phase- and frequency-difference traces to be displayed with their first data point pinned to Y=0.0. In unzoomed views – or in zoomed views with **Trace→Phase/frequency Y axis unlocked in zoom mode** (y) disabled to permit automatic Y-axis scaling – this will force the left end of the trace to the midpoint of the graph's Y axis. Since autoscaled Y axes in TimeLab are symmetrical about zero, this may allow the trace to occupy more of the visible graph area:

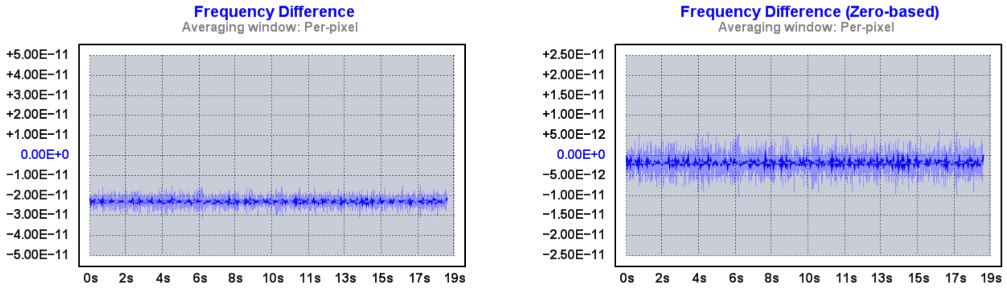


Figure 38: Toggling the zero-baseline view (z)

When used in the **Measurement→Frequency difference (**f**)** view, the effect of **Trace→Phase/frequency traces begin at zero (**z**)** is somewhat similar to applying **Edit→Subtract global linear phase trend (frequency offset) (**Ctrl-o**)** to each visible plot. It's not *quite* the same, though, because a trace's first data point is not necessarily anywhere near its trend line. Subtracting the trend will often do a better job at centering individual traces symmetrically about the Y axis, as seen below.

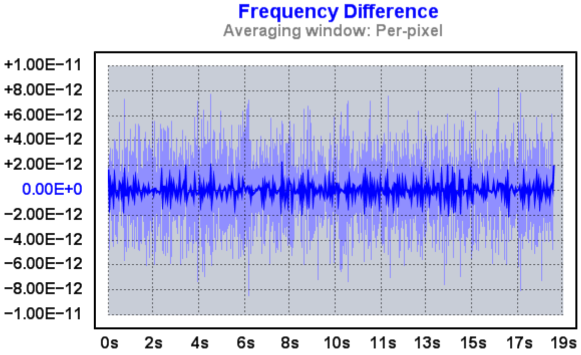


Figure 39: Linear trend subtracted from the phase data

Of course, **Edit→Subtract global linear phase trend (frequency offset) (**Ctrl-o**)** has the drawback of altering the selected plot's phase data record and removing potentially useful information (namely the phase trendline). Fortunately, there's also a non-destructive global option that has the same visual effect, **Trace→Show linear phase/frequency residual (**r**)**:

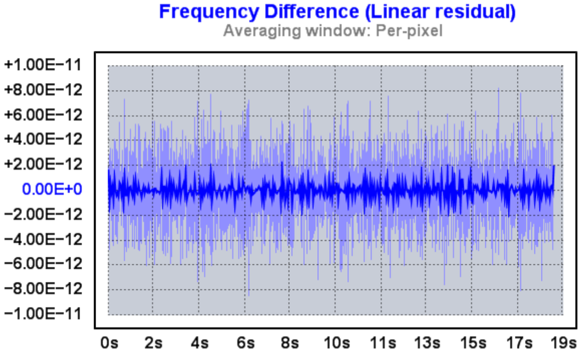


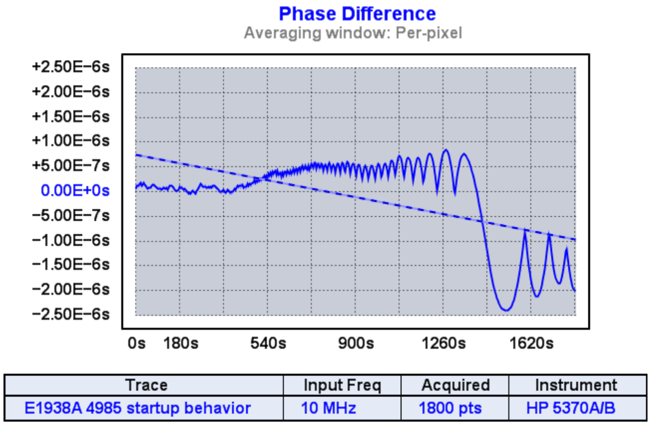
Figure 40: Viewing the removed trend without altering the data

These two commands, **Show linear phase/frequency residual (**r**)** and **Phase/frequency traces begin at zero (**z**)**, are among the most commonly-used options on the **Trace** menu. If you spend much time working with phase- and frequency-difference traces, their r and z keyboard shortcuts will quickly become familiar. z can increase visible detail while preserving the graph's original shape, while r is better at revealing short-term noise and other effects that become harder to spot as the graph expands to accommodate a larger overall trend. Some additional usage examples can be found on page 60.

Note that although only one **Phase/frequency traces begin at zero (**z**)** option appears on the **Trace** menu, Timelab maintains separate copies of the associated preference variable that retain the last change made in each of the respective views.

Show linear phase/frequency trend (Ctrl-t)

When enabled, this option causes TimeLab to display a dashed line on unzoomed phase- and frequency-difference traces to indicate the data's linear trend.



Trace	Input Freq	Acquired	Instrument
E1938A 4985 startup behavior	10 MHz	1800 pts	HP 5370A/B

Figure 41: Viewing the trend line

The trend line is not displayed in magnified views, or when **Trace→Show linear phase/frequency residual (**r**)** is enabled.

Use caution when interpreting drift figures derived from linear or quadratic trends. Measurement traces that exhibit a large amount of variance relative to their overall slope – such as might be obtained from comparisons of GPSDO or atomic standards -- can result in numerically unstable solutions that vary randomly based on the number of points in the data record.

Phase/frequency Y axis unlocked in zoom mode (y)

By default, the Y axis in phase- and frequency-difference measurement graphs is automatically scaled to accommodate the largest absolute value in all visible traces. The peak magnitude is rounded up to the next decade, half decade, or quarter decade, then mirrored to place zero at center scale. This approach allows the 10-division Y axis to be labeled in "nice" multiples of 1, 2, or 5.

When zooming in with the mouse to magnify a portion of the visible trace(s), it's sometimes helpful to toggle the Y-autoscaling algorithm off by selecting

Phase/frequency Y axis unlocked in zoom mode (y). This will have no effect on an unzoomed display, but when dragging with the left mouse button to magnify a desired area of the graph, you'll be able to move in the both X *and* Y directions, instead of only being able to drag a pair of vertical cursors. When you release the button, the resulting zoomed view will conform to the specified extents in both directions.

With the Y axis unlocked, the **Display→Y zoom in (})** and **Display→Y zoom out ({)** commands will be usable, and the scroll wheel will expand or contract the magnified region in both directions at once. Dragging with the middle mouse button will scrub the zoomed region in both the X and Y directions.

You can enable Y-axis autoscaling in zoom mode at any time. A quick way to "zoom out" in Y while maintaining the current X position and magnification factor is to tap the y key twice in succession – once to temporarily re-enable autoscaling, and again to restore 2D navigation with the middle button, scroll wheel, or ( ) { } shortcuts.

Averaging window for phase/frequency traces (g)

Increase averaging window (Ctrl +)

Decrease averaging window (Ctrl -)

These commands allow you to specify the length of an *averaging window* up to 10000 seconds in length that will be applied to phase- and frequency-difference traces. The averaging process makes it easier to spot trends and patterns that may be obscured by short-term noise. Setting the window length to its default value of zero selects a hybrid "per-pixel" peak/average detection algorithm (discussed below.)

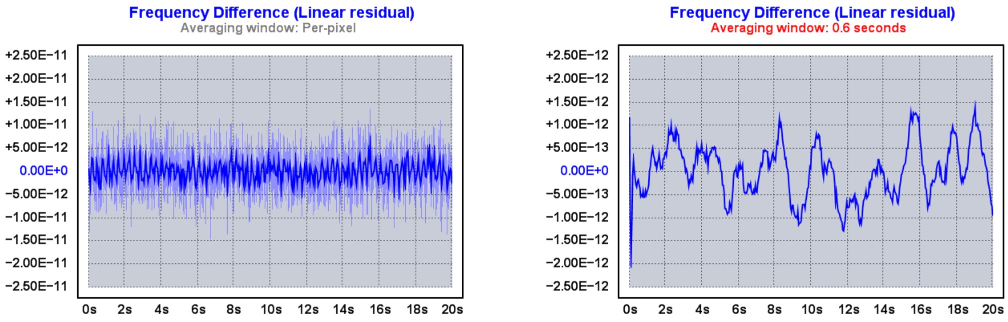


Figure 42: Per-pixel and time-based averaging

As a reminder that averaging is in effect, nonzero averaging times will be displayed in red above phase- and frequency-difference graphs. You can quickly disable averaging with the keyboard sequence g 0 <Enter>.

Averaging also determines how phase records are decimated for display in the phase- and frequency difference views. When rendering a trace whose phase record length in samples exceeds the graph width in pixels, the samples that fall within each graph column must either be peak-detected or combined by averaging. When the averaging window is set to zero seconds, TimeLab effectively does both, using a "per-pixel" algorithm to provide as much information about the skipped phase-record samples as possible. As shown in the first plot above, a shaded vertical line is drawn in each pixel column that connects the minimum and maximum data values that fall within that column. The darker portion of the trace in the middle of the shaded area represents the average of *all* data points covered by each individual pixel column. Consequently the effective averaging time is not fixed, but is equal to the duration of the visible portion of the trace divided by the width of the graph in pixels.

This algorithm is simple, fast, and effective, but since the peak-detected line segments are drawn in a lighter shade of the trace color, it's easy to overlook brief glitches that fit within a single pixel column. Consider enabling **Trace→Tick marks (k)** when looking for outliers in per-pixel mode.

Draw xDEV traces with spline interpolation (i)

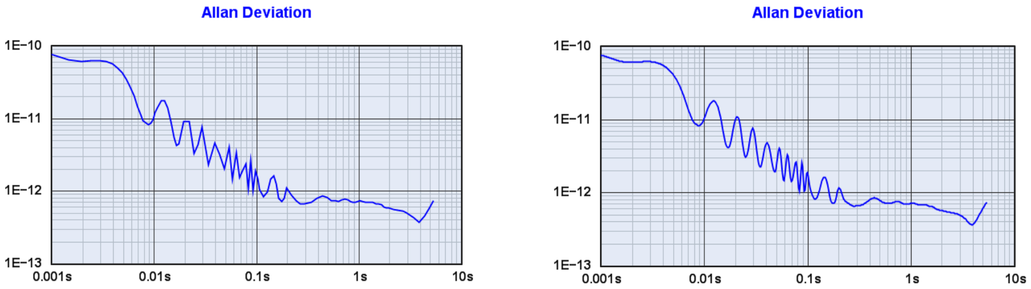


Figure 43: Spline interpolation

By default, screenspace points corresponding to the tau bins in Allan deviation and related statistical views are connected with a cubic spline. Spline interpolation results in a smooth, visually-appealing curve, but it can shift data points subtly, distort or exaggerate ringing artifacts, and cause other abrupt transitions in the graph. When **Trace→Draw xDEV traces with spline interpolation (i)** is toggled off, straight lines are used to connect the xDEV bins (above left).

Spline interpolation is not available when displaying MTIE plots, and should be used with particular care when viewing cross Allan deviation plots that have not yet fully converged to a continuous trace.

Show xDEV error bars (Ctrl-e)

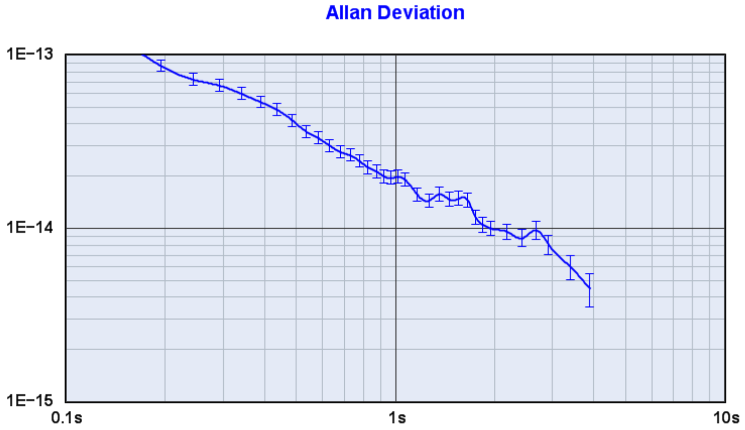


Figure 44: Allan deviation error bars

When enabled, this option draws “error bars” at each tau bin in Allan deviation and other statistical plots. These bars approximate the ± one-sigma (68%) confidence interval at ± σ/sqrt(N/M), where N is the number of phase-record samples that contribute to each bin and M is the bin’s τ_0 multiple.

This calculation does not take either the deviation type or the noise slope into account, so it should be considered a rough estimate of the actual confidence interval obtainable through more sophisticated offline analysis. Consider exporting your phase data to Stable32 (page 80) if your application requires better confidence-interval estimates.

Clip xDEV traces by noise bandwidth (Ctrl-b)

This display option is used to suppress artifacts near the tau-zero points in Allan deviation and other statistical plots that can appear when incoming data is oversampled for lowpass filtering. xDEV values at tau intervals close to the sample rate may appear to “droop” in such cases, since the data has been band-limited at a cutoff frequency below its Nyquist rate.

A discussion of this effect appears on page 50.

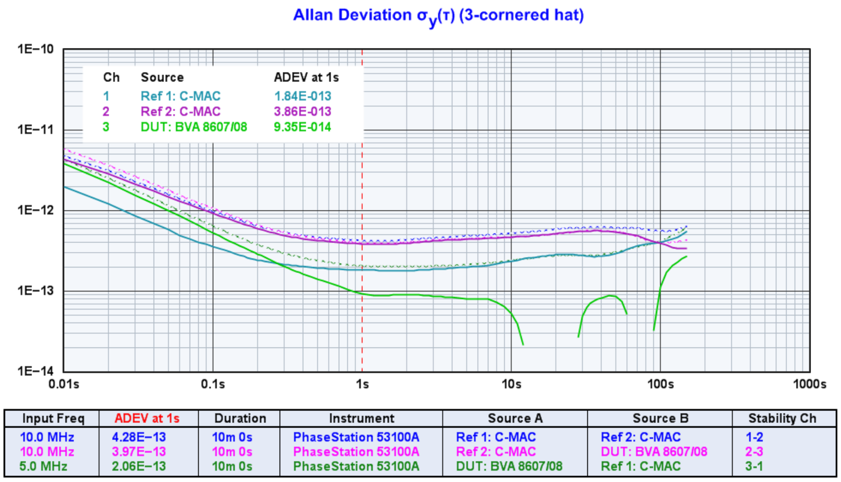
Clip xDEV traces by confidence (Ctrl-v)

This display option is used to avoid rendering Allan deviation and other xDEV traces beyond the point where insufficient data is available for good statistical confidence. It is enabled by default.

Specifically, when this option is enabled, the only xDEV bins that are drawn are those whose phase data sample count (N) is >= 2x their τ_0 multiple (M).

Show separated xDEV variances (N-cornered hat) when available (Ctrl-h)

This display option provides an alternative way to view Allan deviation and other statistical traces. Under appropriate conditions, it can reveal aspects of oscillator performance that would otherwise be inaccessible in a conventional measurement with a single reference source.



Ch	Source	ADEV at 1s
1	Ref 1: C-MAC	1.84E-013
2	Ref 2: C-MAC	3.86E-013
3	DUT: BVA 8607/08	9.35E-014

Input Freq	ADEV at 1s	Duration	Instrument	Source A	Source B	Stability Ch
10.0 MHz	4.28E−13	10m 0s	PhaseStation 53100A	Ref 1: C-MAC	Ref 2: C-MAC	1-2
10.0 MHz	3.97E−13	10m 0s	PhaseStation 53100A	Ref 2: C-MAC	DUT: BVA 8607/08	2-3
5.0 MHz	2.06E−13	10m 0s	PhaseStation 53100A	DUT: BVA 8607/08	Ref 1: C-MAC	3-1

Figure 45: Three-cornered hat display

When the N-cornered hat view is enabled in the N=3 case as shown above, TimeLab will attempt to isolate and display the individual contribution of each of three devices to a set of three stability measurements. Although an in-depth discussion of this technique is outside the scope of this document, Application Note 53100A-2 (*Dual-Reference Noise and Stability Measurements with the PhaseStation 53100A)* presents a detailed "case study" in which the three-cornered hat methodology is used to provide mutual assessments of high-performance crystal oscillators. Please refer to this application note for more information.

When viewing three-cornered hat traces, it is strongly recommended that only the three participating measurements are enabled for display. All other plots should either be cleared with **File→Close selected plot** (Del) or marked invisible with **Display→Toggle visibility of selected plot** (v).

To configure the PhaseStation 53100A for three-cornered hat measurements, see the description of the **Measure three independent DUTs (three-cornered hat)** measurement role on page 134. Note that four-cornered hat plots are supported by TimeLab, but require six concurrent measurements rather than three. Consequently, they cannot be acquired simultaneously by the 53100A due to its limitation of four concurrent plots.

Cross Allan deviation measurements are often preferable to the three-cornered hat methodology, particularly when the goal of the measurement is to gather data for only one DUT. See the **Cross Allan Deviation Measurements** section on page 139 for more information.

Show cross ADEV traces when available (Ctrl-j)

Cross Allan deviation measurements take advantage of all four instrument input channels, potentially enhancing the short-term ADEV measurement floor by an order of magnitude. On the 53100A, these measurements are performed when the DUT and reference sources are routed to both of their respective front-panel input jack pairs with internal or external splitters and the **Measure single DUT with cross ADEV statistics** measurement role is selected.   Cross ADEV measurements are also performed by default with the dual-reference measurement roles, and by manually-configured measurements that assign individual stability reference channels to DUT channels 3 and 4.

If this option is toggled off, cross ADEV plots will be rendered as conventional ADEV measurements when the Allan deviation measurement view is selected with the **a** hotkey. Only the primary phase record from channels 3-1 will contribute to the display, and measurement performance enhancements associated with cross ADEV processing will be unavailable.

For more information, see the **Cross Allan Deviation Measurements** section on page 139, as well as the list of measurement role options on page 131.

Show original traces in computed xDEV displays (F6)

This option can be used to show or hide the original xDEV traces that contribute to an N-cornered hat display, as seen in the example above.

Alternatively, when cross ADEV measurements are in use, this command determines whether the standard ADEV traces for the primary and secondary phase records are displayed alongside the primary trace that represents the cross ADEV solution. To avoid confusion, these traces are always rendered with a broken (dashed) line style. They can be hidden with **F6** or by simply toggling the entire plot's visibility with the **v** hot key.

Show correlation gain for selected noise trace (Ctrl-g)

Intended primarily as a diagnostic aid, this option is supported when viewing phase noise or AM noise plots from instruments such as the PhaseStation 53100A that support real-time cross correlation. When enabled, the FFT segments in the selected phase noise or AM noise plot will be color-coded for identification. A table will appear as an overlay in the graph area that displays the number of cross-spectrum averages performed so far in each color-coded segment, along with the theoretical improvement in ADC noise expressed in dB.

The estimated improvement in each segment's effective noise floor is calculated as 5 * $\log_{10}(N)$ dB, where N is the segment's number of averages.

Trace smoothing is not performed on the selected noise plot when this option is enabled.

Show FFT segment filter slopes (Ctrl-i)

This option is a diagnostic aid intended for use when editing FFT segment table files for phase noise and AM noise acquisition with the PhaseStation 53100A or TimePod 5330A.

When acquiring noise plots on the PhaseStation 53100A, data for each visible FFT segment undergoes highpass and/or lowpass filtering based on the declarations in the *segment table*, accessible via the **Additional options** page in the 53100A acquisition dialog. By rendering the entire extent of each segment record, **Trace→Show FFT segment filter slopes (Ctrl-i)** provides a visual display of the filter skirts for each segment, as well as the overlap between them. In this mode, you can temporarily hide the skirts by holding the **Shift** key.

For detailed information about the segment table's format and contents, refer to the comments within the default segment table file. This file can be accessed by pressing the **Additional options** button in the 53100A acquisition dialog, followed by **Edit segments**.

Most users will not need to edit the segment table.

Show imaginary part of cross spectrum (Ctrl-F3)

Show estimated instrument noise (F2)

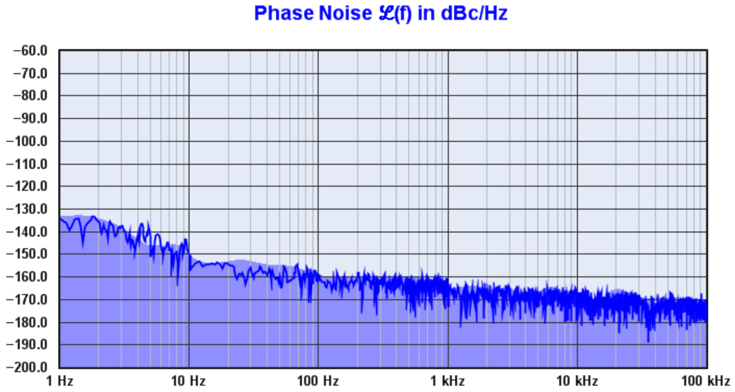


Figure 46: Shaded noise floor estimate

Phase noise and AM noise plots acquired with the PhaseStation 53100A, TimePod 5330A, and Symmetricom 512X analyzers can be displayed with a shaded area that provides an estimate of the instrument noise floor, if you enable **Trace→Show estimated instrument noise** (F2).

In the 53100A's case, the floor estimate is simply a heavily-smoothed rendition of the imaginary portion of the cross-spectrum average shown by the **Show imaginary part of cross spectrum** (Ctrl-F3) command. For more information about the imaginary cross spectrum and the Ctrl-F3 command in particular, see "Understanding instrument spurs" on page 71.

Use caution when relying on the noise-floor estimate. While reasonably accurate, it may appear artificially high in the vicinity of instrument spurs. Conversely, certain effects can raise the real measurement floor slightly above the shaded floor estimate. As with ADEV measurements, the best way to obtain "proof of performance" for analyzers in this class is to run residual tests at similar frequencies and signal levels.

Particularly when viewing multiple traces in **Display→Overlay** (o) mode, it may be preferable to leave the floor display turned off in order to reduce display clutter.

Note that when **Show imaginary part of cross spectrum** (Ctrl-F3) is active, the **File→Export ASCII AM/PM noise trace** command will export the imaginary cross spectrum trace data rather than the normal AM noise or phase noise trace.

Mark spurs in noise traces (Ctrl-m)

Suppress spurs in noise traces (Ctrl-s)

For a detailed discussion of these commands and other aspects of spur detection and rendering in phase noise and AM noise plots, see page 68.

Smooth noise traces (Ctrl-w)

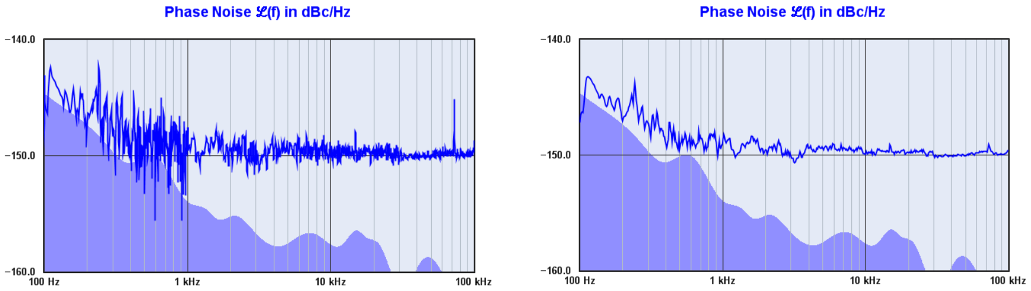


Figure 47: Trace smoothing

This command applies some light exponential smoothing to traces in the phase noise and AM noise measurement views. Detected spurs are always removed from smoothed traces. If enabled, smoothing is turned off automatically when the **Trace→Suppress spurs in noise traces** (Ctrl-s) command is used to toggle spur attenuation.

To obtain a smoother trace with cross-correlating analyzers such as the PhaseStation 53100A, TimePod 5330A and Symmetricom 512x models, it's often better to let the measurement run longer. As the trace converges on the true noise level its variance will diminish, resulting in a more accurate measurement with less visible "grass."

Show raw noise trace(s) (Ctrl-r)

This command displays the raw phase noise or AM noise spectrum from the ADCs in both measurement channels of the PhaseStation 53100A and TimePod 5330A.

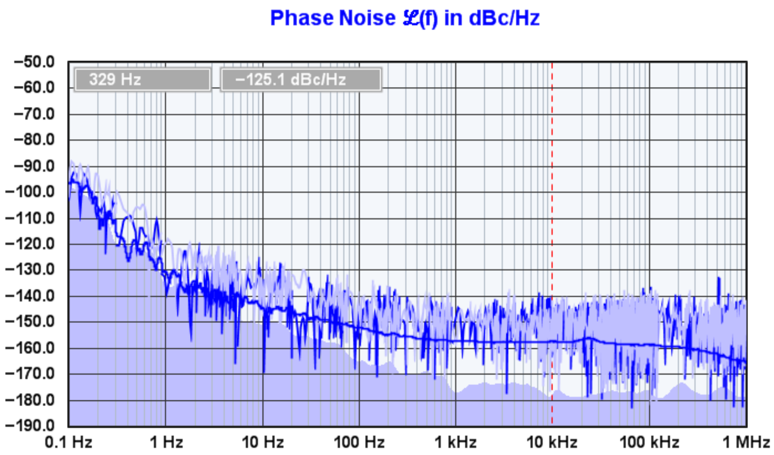
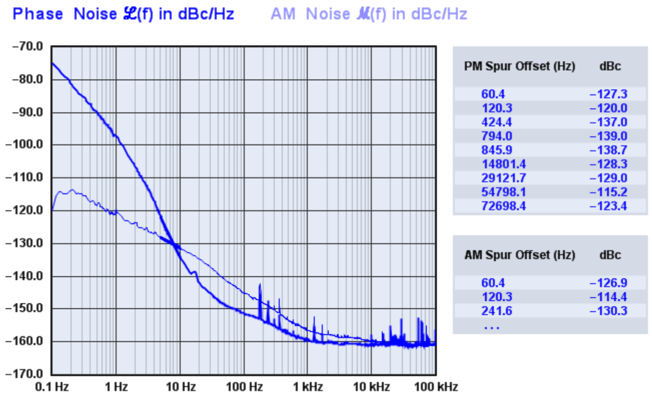


Figure 48: Viewing the raw ADC noise traces

Note that while the **Trace** menu option shows the raw ADC data traces alongside the actual measurement traces, the **Ctrl-r** keyboard shortcut cycles between measurement-only, measurement+raw, and raw-only displays.

Although intended primarily for instrument diagnostic purposes, the appearance of the raw trace can sometimes be helpful in tracking down intermittent problems in the DUT or reference source associated with one of the two channel pairs. The darker trace corresponds to the first DUT/noise reference channel pair, while the lighter trace corresponds to the second pair of inputs. Watching for "jumpy" behavior in the light trace, the dark trace, or both can help identify issues that don't appear in the relatively low-bandwidth stability measurements.

Show AM noise in PN view (F8)



PM Spur Offset (Hz)	dBc
60.4	−127.3
120.3	−120.0
424.4	−137.0
794.0	−139.0
845.9	−138.7
14801.4	−128.3
29121.7	−129.0
54798.1	−115.2
72698.4	−123.4

AM Spur Offset (Hz)	dBc
60.4	−126.9
120.3	−114.4
241.6	−130.3
. . .	

Figure 49: Overlaying the AM and PM noise traces

When enabled, this command renders a copy of the AM noise trace in the **Measurement→Phase noise** (P) view, using a lighter color or trace weight. Both PM and AM spur tables are displayed, space permitting.

Tick marks (k)

This command toggles tick marks on and off, as shown below.

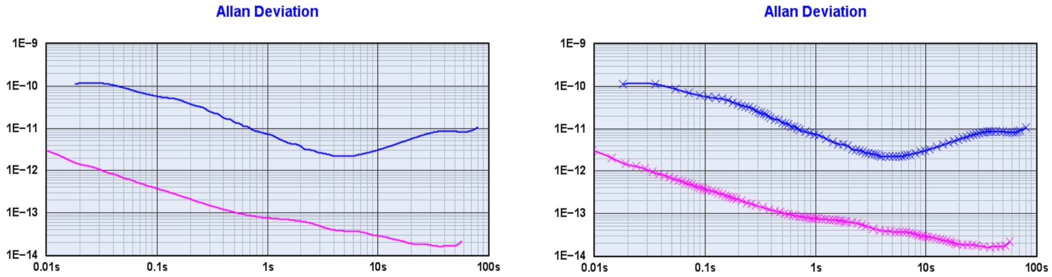


Figure 50: Tick marks

Tick marks are supported in most trace types and display modes. They can be especially handy when examining phase/frequency difference plots. In cases where many data points contribute to a single pixel column in the graph and no explicit averaging time has been specified, TimeLab's "per-pixel" renderer displays the average of all of each pixel column's contributing points, while the maximum and minimum points that fall within each pixel column are rendered in a lighter shade of the same color. The k command can help you spot brief glitches and frequency jumps in these plots that would otherwise go unnoticed.

Toggle trace thickness for current measurement (T)

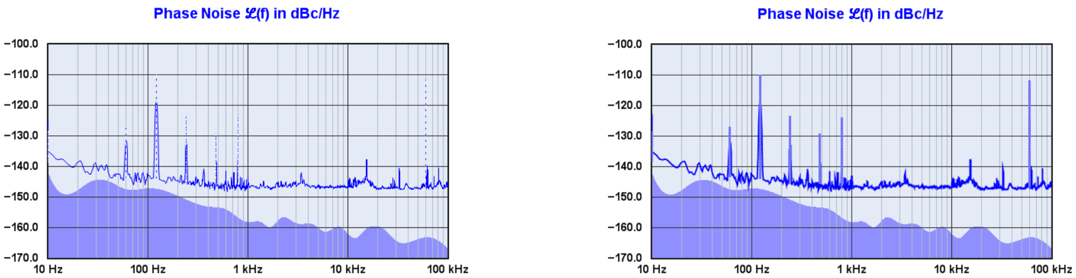


Figure 51: Trace thickness options

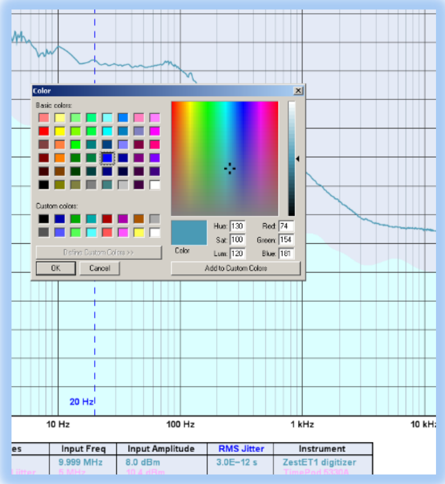
This command switches between heavy and light traces in the currently-active measurement view. TimeLab keeps track of the requested trace thickness for each **Measurement** menu entry.

In N-cornered hat displays, the trace thickness property controls the appearance of the original xDEV traces that contribute to the solution. Heavy traces are rendered with thin lines in this case, while light traces are rendered with broken lines.

Display Menu

Display-related options that aren't measurement-specific appear on the **Display** menu. Controls on this menu determine the visibility and selection status of loaded plots, the order in which plots appear in the legend table below the graph, and the choice of **Display→Overlay** (o) or **Display→Browse** (b) mode that determines whether all loaded plots are displayed or only the selected plot.

Color, contrast, font size, and graph-magnification options also appear on the **Display** menu, along with a choice of formats for the "mouseover" information at the upper left corner of the graticule.



Hue: 130
Sat: 100
Lum: 120
Red: 74
Green: 154
Blue: 181

es	Input Freq	Input Amplitude	RMS Jitter	Instrument
	9.999 MHz	8.0 dBm	3.0E−12 s	ZestET1 digitizer
Jitter	5 MHz	10.4 dBm		TimePod 5330A

Figure 52: The color palette editor

Edit colors

This command leads to a second-level menu offering a choice of various graphical elements in the TimeLab window, from font and trace colors to table borders and backgrounds. Once an element is chosen, a standard Windows color palette dialog appears. As you click on the palette areas to change the selected color, the TimeLab window is updated in real time, making the color-selection process interactive and easy to use.

Edit selected trace color (E)

This command provides convenient access to the trace color options in the **Display→Edit colors** submenu.

Restore default colors

This command restores the default color palette without resetting any other preferences or control states.

Restore selected trace color

This command restores the default color for the selected trace without resetting any other colors, preferences or control states.

High contrast (C)

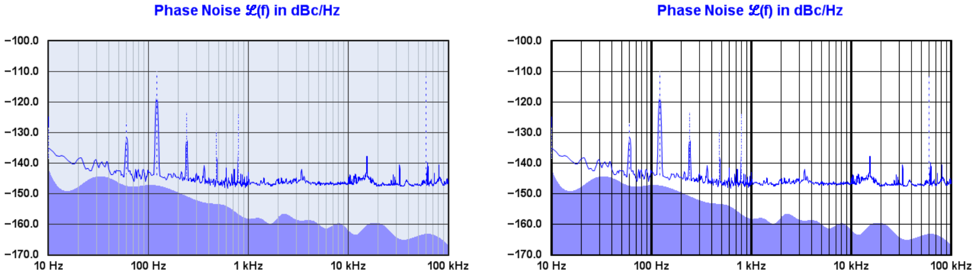
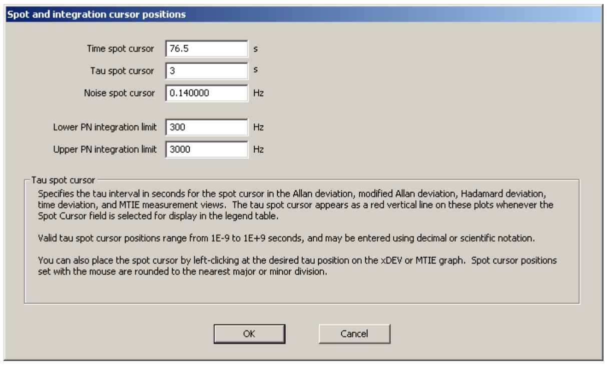


Figure 53: Contrast adjustment

This command toggles the display between normal and high-contrast rendering. The latter option may be desirable when saving screenshots for publications or Web pages whose color fidelity may not be sufficient to reveal details in the plot such as minor graticule lines.

Set spot cursor positions (j)

This command opens a dialog box similar to the one shown at right, allowing the spot cursor and phase noise integration limit cursors to be placed at specific locations.



Spot and integration cursor positions

Time spot cursor 76.5 s

Tau spot cursor 3 s

Noise spot cursor 0.140000 Hz

Lower PN integration limit 300 Hz

Upper PN integration limit 3000 Hz

Tau spot cursor

Specifies the tau interval in seconds for the spot cursor in the Allan deviation, modified Allan deviation, Hadamard deviation, time deviation, and MTIE measurement views. The tau spot cursor appears as a red vertical line on these plots whenever the Spot Cursor field is selected for display in the legend table.

Valid tau spot cursor positions range from 1E-9 to 1E+9 seconds, and may be entered using decimal or scientific notation.

You can also place the spot cursor by left-clicking at the desired tau position on the xDEV or MTIE graph. Spot cursor positions set with the mouse are rounded to the nearest major or minor division.

Figure 54

The spot cursor may be enabled for display by selecting the "Spot Cursor" field in the **Legend** menu. While it is normally placed by left-clicking within the graph area, **Display→Set spot cursor positions (j)** allows you to specify its location more precisely, as well as to place it at offsets not normally permitted (such as between minor divisions on the phase noise and AM noise graticules.) Alternatively, left-clicking on the graph while the spot cursor is turned off will issue a prompt to enable it.

Refer to the mouseover help text in the dialog box for more information.

Show numeric tables (Ctrl-n)

The *numeric table* appears to the right of the graph area in each measurement view, unless toggled off with this command. The format of the table is different for each measurement type. In the Allan deviation and other statistical views, a chart displays the sigma(tau) values at assorted tau intervals. In the **Measurement→Frequency difference (f)** view, both drift/trend statistics and a high-precision frequency-count chart are displayed, while only the former is available in the **Measurement→Phase difference**

(p) view. Finally, tables of user-defined noise markers and detected spurs are displayed in the phase noise and AM noise views.

Clear noise markers (Ctrl-F4)

This command clears the list of user-defined noise markers, removing it from phase noise and AM noise plots. To add a new noise marker, simply **Shift-left click** near the desired column on either the phase noise or AM noise plot.

Show integrated PN traces

Enables or disables the display of integrated SSB noise and RMS jitter traces when one or more of the integrated noise options is selected for viewing in the **Legend→Select** menu. See page 67 for more details.

Show cursor time (Ss)

Show cursor time (Hh:Mm:Ss)

Show cursor time/datestamp

Do not show cursor values

These options determine the content of the "mouseover" cursor readout display in the **Measurement→Phase difference** (p) and **Measurement→Frequency difference** (f) views. When the mouse cursor is within the graph area in these views, the readout at upper left displays the time and Y-axis value corresponding to its position. You can choose to display the number of seconds relative to the beginning of the phase record (**Ss**), the same elapsed time value in **Hh:Mm:Ss** format, or the absolute **time/datestamp** when the data at the mouse cursor location was logged.

Note that **Display→Do not show cursor values** turns off the cursor readout in *all* views. To re-enable the cursor readout in the xDEV and noise displays, select any of the other three options.

The mouse cursor readout is not rendered when saving image files or capturing images with **File→Copy image to clipboard** (Ctrl-c).

Phase/frequency X axis labels in Ss format

Phase/frequency X axis labels in Hh:Mm:Ss format

Phase/frequency X axis labels in Hh:Mm format

Phase/frequency X axis labels in Hh format

These options determine the resolution at which X-axis (time) units are labeled on the phase and frequency difference traces.

Supported X axis label formats include seconds only (the default), hours:minutes:seconds, hours:minutes, or hours only. The latter three options also include days, months, and years when appropriate.

Frequency Y axis labels in Hz (Z)

Determines whether the Y axis in the **Measurement→Frequency difference** (f) view is based on fractional frequency differences or absolute frequency differences in Hz. In both cases the frequency-difference values are relative to the frequency shown in the DUT Freq field in the legend table. By default, fractional frequency differences are displayed.

In addition to the Y-axis labels, the spot cursor field in the legend table is also adjusted based on this setting, as are mouse-cursor queries shown at upper left in the graph area and the Origin column in the frequency slope table at upper right. However, the Error column in the frequency-count chart always displays fractional frequency differences. The frequency counts themselves are also not affected by this setting.

Tau labels in scientific notation

Log-x plots used to display stability measurements in TimeLab are labeled with tau values in seconds corresponding to each decade, from 1E-6 (microseconds) to 1E+9 (billions of seconds). The decades may be labeled in scientific notation or as literal decimals (the default).

Browse plots one at a time (b)

Overlay all loaded plots (o)

Toggle visibility of selected plot (v)

Since TimeLab allows up to nine plots to be loaded and displayed at once, these three commands are needed to maintain a legible display. They are often used together in rapid succession, so they will be described together and referenced in abbreviated form.

Simply stated, in **b)rowse** mode only the selected plot is rendered. In **o)verlay** mode, all loaded plots that haven't had their **v)isibility** toggled off are rendered, with the graph scale adjusted as necessary to accommodate them.

It's common to switch between **b)rowse** and **o)verlay** mode when inspecting and comparing several plots, as the display can become rather crowded. At the same time, the **v)isibility** command is often needed when you want to view a subset of plots in o)verlay mode. **b)rowse** mode will exclude all but the selected plot from the display, while the **v)isibility** toggle gives you the control needed to display some, all, or none of the loaded plots in **o)verlay** mode. If you want to look at several overlaid plots while temporarily hiding one or two of them, the **v)isibility** toggle is the answer.

Select next plot in chart (+ or down arrow)

Select previous plot in chart (- or up arrow)

Move selected plot up (Ctrl-up arrow)

Move selected plot down (Ctrl-down arrow)

For an introduction to the "selected plot" concept and its related commands, refer to "Making your first measurements" on page 33.

The **Ctrl-up/down arrow** commands can be very helpful for organizing displays with multiple plots. Holding down the Ctrl key when you press an arrow key will not change the currently selected plot, but will instead move the selected plot up and down in the legend table. Since the trace colors in TimeLab are assigned based on each plot's position in the legend table, this feature can be useful if you don't like the color of a particular plot, or if a plot farther down the table is obscuring data in an earlier plot. (The **Ctrl-up/down arrow** commands are frequently handy when using **Trace→Show estimated instrument noise (F2)**, for instance.)

X zoom in ( ])

X zoom out ([ )

Y zoom in (})

Y zoom out ({)

The bracket and shift-bracket keys are useful for navigating zoomed phase- and frequency-difference traces on a laptop or other PC without a three-button scrolling mouse. These menu options can be thought of as placeholders for their respective keyboard shortcuts. Specifically, after a zoomed region has been selected by dragging with the left mouse button, these commands allow you to increase and decrease the “magnification factor” without returning to the unzoomed display.

On a desktop PC, a combination of scroll-wheel and middle-button input is normally the best way to navigate within phase/frequency traces. The scroll wheel expands and shrinks the magnification factor, while the middle mouse button allows you to “scrub” the zoomed trace data left and right.

These commands have no effect in zoomed phase noise or AM noise measurement views, or in Allan deviation or other statistical views. For more about TimeLab’s zoom functionality, refer to “Navigating zoomed graphs” on page 61.

Decrease font size ((or Ctrl-mouse wheel)

Increase font size ( ) or Ctrl-mouse wheel)

Like the **X zoom/Y zoom** commands, the font-size control options are primarily meant as placeholders for their respective “hotkeys,” the left and right parentheses. They provide easy access to a choice of several discrete font sizes. Also, as with most Web browsers and other newer Windows applications, scrolling with the mouse wheel while holding down the Ctrl key will expand or shrink the text within TimeLab’s display window.

Refresh (F5)

This command triggers a recalculation of the graph extents for Allan deviation and related log-log statistical plots when these measurement views are selected. Alternatively, when the phase noise or AM noise measurement view is selected, the extents for all visible noise plots will be reset to their user-defined graph limits.

If a trace-refresh or data acquisition operation is already in progress at the time the command is issued in a deviation measurement view, only the graph extents are recomputed. Otherwise, the bin contents will be recomputed first in order to ensure that the graph extents are sized correctly.

Legend Menu

The **Legend** menu contains only one command, **Legend→Select (d)**. This command opens a dialog box in which you can choose which attributes are displayed for each visible plot in the legend table beneath the main graph. Some properties are flagged for display in the legend table by default, while others will not be displayed until you select them explicitly.

Each TimeLab acquisition driver can contribute its own entries to the **Legend→Select** dialog based on the property values in the plots that it generates. When a plot is acquired or loaded from a .TIM file, TimeLab checks for the presence of any displayable properties that it hasn't yet encountered. Any unfamiliar properties are logged in **TIMELAB.INI** and added to the selection menu.

As a result, the content of the **Legend→Select** menu may vary from one TimeLab installation to the next. (Note that after you select a property for display in the legend table, its value will be blank for all plots that don't include that particular property.)

As with other data stored in **TIMELAB.INI**, reinstalling TimeLab does not erase the list of known **Legend→Select** menu entries. These entries can be cleared only by manually editing or deleting **TIMELAB.INI**, or by using **File→Reset all parameters, options, and settings at next startup**. Refer to the latter command for more information about **TIMELAB.INI**.

The list of currently-defined .TIM file properties appears below. Many of these options are 'private' to a specific hardware driver and are not eligible for display on the legend table. They are listed here for reference only. A few of the properties are defined by TimeLab itself and will appear on the **Legend→Select** menu by default in all installations.

Acquired	Number of acquired phase data points used for stability and deviation plots
ADC Clock	Specified, estimated, or measured acquisition clock rate in Hz
ADC Format	Data format used by ADCs
ADC Temperatures	Average ADC temperature with individual deviations
Add'l AM Gain	Optional user-specified gain value added to AM noise traces
Add'l PN Gain	Optional user-specified gain value added to phase noise traces
Address	Address of instrument used to acquire data
Attenuate Instrument Spurs	True if known instrument spurs are suppressed automatically
Auto Clock	True if the clock frequency was determined automatically
Auto DUT	True if the DUT frequency is considered to be the frequency at the instrument input jack
Auto Input	True if the input frequency was measured automatically
Auto Ref	True if the reference frequency was measured automatically
Automatic Configuration	True if the TimeLab driver should infer various measurement parameters based on the incoming data
Autoscale Deviation	True if statistical deviation graph should automatically accommodate the extents of all visible traces
Autoscale Noise	True if noise graph should automatically accommodate the extents of all visible traces
Autosense Rate	True if the TimeLab driver should automatically measure the sample interval
Beatnote	Amplitude in dBm of calibration beatnote
Bin Density	Determines the number and distribution of tau intervals in statistical deviation measurements
Bin Threshold	Minimum number of data points required to display the statistical deviation at a given tau
BIST Result	Result of last built-in self test operation
BW Index	Index or other identifier for a given measurement bandwidth
Cancel CM Drift	True if common-mode drift (e.g., due to ADC clock warmup) is corrected in software
Ch 0 Auto	True if Ch 0 frequency should be measured automatically
Ch 0 Freq	Specified input frequency for channel 0
Ch 0 Label	User-assignable label for source connected to instrument channel 0
Ch 1 Auto	True if Ch 1 frequency should be measured automatically
Ch 1 Enabled	Enabled/disabled status for channel 1

Ch 1 Freq	Specified input frequency for channel 1
Ch 1 Freq Mode	Input frequency measurement mode for channel 1
Ch 1 Label	User-assignable label for source connected to instrument channel 1
Ch 1 Noise Ref	Reference for DUT channel 1 (Noise)
Ch 1 Rounding	Input frequency rounding parameter for channel 1
Ch 1 Specified Frequency	Input frequency specification parameter for channel 1
Ch 1 Stability Ref	Reference for DUT channel 1 (Stability)
Ch 1 Type	Input type for channel 1
Ch 2 Auto	True if Ch 2 frequency should be measured automatically
Ch 2 Enabled	Enabled/disabled status for channel 2
Ch 2 Freq	Specified input frequency for channel 2
Ch 2 Freq Mode	Input frequency measurement mode for channel 2
Ch 2 Label	User-assignable label for source connected to instrument channel 2
Ch 2 Noise Ref	Reference for DUT channel 2 (Noise)
Ch 2 Rounding	Input frequency rounding parameter for channel 2
Ch 2 Specified Frequency	Input frequency specification parameter for channel 2
Ch 2 Stability Ref	Reference for DUT channel 2 (Stability)
Ch 2 Type	Input type for channel 2
Ch 3 Auto	True if Ch 3 frequency should be measured automatically
Ch 3 Enabled	Enabled/disabled status for channel 3
Ch 3 Freq	Specified input frequency for channel 3
Ch 3 Freq Mode	Input frequency measurement mode for channel 3
Ch 3 Label	User-assignable label for source connected to instrument channel 3
Ch 3 Noise Ref	Reference for DUT channel 3 (Noise)
Ch 3 Rounding	Input frequency rounding parameter for channel 3
Ch 3 Specified Frequency	Input frequency specification parameter for channel 3
Ch 3 Stability Ref	Reference for DUT channel 3 (Stability)
Ch 3 Type	Input type for channel 3
Ch 4 Auto	True if Ch 4 frequency should be measured automatically
Ch 4 Enabled	Enabled/disabled status for channel 4
Ch 4 Freq	Specified input frequency for channel 4
Ch 4 Freq Mode	Input frequency measurement mode for channel 4
Ch 4 Label	User-assignable label for source connected to instrument channel 4
Ch 4 Noise Ref	Reference for DUT channel 4 (Noise)
Ch 4 Rounding	Input frequency rounding parameter for channel 4
Ch 4 Specified Frequency	Input frequency specification parameter for channel 4
Ch 4 Stability Ref	Reference for DUT channel 4 (Stability)
Ch 4 Type	Input type for channel 4
Ch 5 Label	User-assignable label for source connected to instrument channel 5
Channel	Identifies the (zero-based) channel number associated with a multichannel acquisition
Channel Attenuators	Comma-separated attenuation values in dB for each channel
Clock Nominal	True if the clock frequency is a nominal value that may be refined during self-calibration
Clock Oscillator Offset	Margin offset frequency in Hz for clock oscillator
Clock Signal Generator	Address of clock signal generator
Clock Signal Monitor	Address of clock signal monitor
Clock Stabilizer	True if clock duty cycle stabilization is enabled
CM Servo	Common-mode servo functionality for ADC inputs
Comment Prefix	ASCII text comment delimiter
Counter Connection Type	Interface used for acquisition from TIC or frequency counter
Cross ID	Binding identifier for cross-statistical measurement plots
Cross Stability Metrics	True if cross statistics should be applied to stability measurements
Data Format	Indicates the format of the incoming data (e.g., decimal, hex, binary)
Data Port	Instrument data port
Data Type	Indicates the type of the incoming data (e.g., phase differences, frequency differences, or absolute frequency)
Decimation	Hardware decimation ratio
Device ID	Instrument serial number or other identifier
Dither Level	Level of dithering applied to reduce spurious responses in input signals
Driver	Identifies the TimeLab .TLL driver used for acquisition
Driver version	Version of TimeLab driver used for data acquisition
Duration	Measurement duration
Duration Type	Measurement duration units
DUT Freq	Frequency of device under test (DUT) provided by user, prior to any conversion, multiplication, or division
EFC	Acquisition clock EFC DAC value
Elapsed	Elapsed time in the current measurement or measurement stage
ENBW	Equivalent noise bandwidth in Hz of recorded phase data
ENBW Factor	Equivalent noise bandwidth of recorded phase data, expressed as a fraction of the data rate
ENBW Index	Index or other identifier for a given noise-equivalent measurement bandwidth
EOS Character	End-of-string character for data source
Ext Converter	External frequency divider or mixer configuration
Ext LO	Specified frequency in MHz of local oscillator used for external mixing
Ext N	Specified division factor (N) used for external frequency division
FFT 0	Combination of physical ADC channel(s) contributing data to FFT ch 0

FFT 1	Combination of physical ADC channel(s) contributing data to FFT ch 1
FFT Win #	Index or other identifier for FFT window function used for noise measurement
FFT Window	FFT window function used for noise measurement
Field #	Numeric field position from which data was extracted
Filename	Filename under which the .TIM file associated with each plot was most recently saved
FPGA Bitfile	Name of the .bit file used to configure the FPGA prior to acquisition
FPGA Temperature	Approximate FPGA core temperature reported by on-chip sensor
FPGA Version	FPGA firmware version
Frequency	Incoming data consists of absolute frequency readings
Frequency Difference	Incoming data consists of fractional frequency differences
FRQ Max Offset	Maximum offset frequency for spectrum display
FRQ Max Scale	dBm value at top of spectrum display
FRQ Min Offset	Minimum offset frequency for spectrum display
FRQ Min Scale	dBm value at bottom of spectrum display
HP 5313xA Mode	True if the TimeLab driver should ignore statistics data from an HP 53131A/53132A counter
I/O Block Size	I/O buffer size in bytes
I/O Blocks	Number of I/O blocks in ring buffer
Imported From	Name of file or device from which data was imported
Incoming Sample Interval	Time in seconds between successive data points from the instrument
Input Amplitude	Approximate input amplitude in dBm
Input Dither	True if input dithering is enabled (certain instruments only)
Input Estimated	True if the input frequency is a low-precision estimate that should be rounded for display purposes
Input Freq	Frequency of signal applied to primary input jack, often rounded for display
Input Precision	Round input frequencies to the specified # of digits to the right of the decimal point
Input Splitter	True if input channel amplitudes should be adjusted by 3 dB to correct for internal or external RF splitter
Input Splitter (DUT)	DUT splitter
Input Splitter (Ref)	Reference splitter
Instrument	Instrument model used to acquire data
Interface	Configuration details for the GPIB adapter, serial or LAN adapter, or other physical interface
IP Address	IP address or hostname of instrument used to acquire data
L(f) Factor	Additional offset applied when converting dBc/Hz readings into L(f) SSB phase noise
L(f) Max Scale	dBc/Hz value at top of SSB noise graph
L(f) Min Scale	dBc/Hz value at bottom of SSB noise graph
LNA dB	Gain provided by LNA stage
Mask Margin	Amount by which the mask test has passed (positive) or failed (negative)
Mask Result	Result obtained by testing the currently-displayed mask against each visible trace
Max Sigma	Sigma value at top of statistical deviation graph
Max Tau (s)	Maximum tau value in seconds for statistical deviation graph (which will be rounded up to the nearest decade)
Measure AM Noise	True if measurement contains AM noise data
Measure Frequency Spectrum	True if measurement contains two-sided frequency spectrum data
Measure PM Noise	True if measurement contains phase noise data
Measure Stability	True if measurement contains phase/frequency stability data
Min Sigma	Sigma value at bottom of statistical deviation graph
Min Tau (s)	Minimum tau value in seconds for statistical deviation graph (which will be rounded down to the nearest decade)
MJD	Modified Julian Date when the measurement was triggered
Multichannel Data Port	Instrument data port(s) used to acquire each measurement channel
Multichannel ENBW	Equivalent noise bandwidth in Hz for each channel in a multichannel measurement
Multichannel Input dBm	Approximate input amplitude in dBm for each channel in a multichannel measurement
Multichannel Input Estimation	True for input channels whose frequencies are low-precision estimates that should be rounded for display purposes
Multichannel Input Hz	DUT input frequency in Hz for each channel in a multichannel measurement
Multichannel Input Precision	Round input frequencies in each measurement channel to the specified # of digits to the right of the decimal point
Multichannel Ref dBm	Approximate reference amplitude in dBm for each channel in a multichannel measurement
Multichannel Ref Estimation	True for reference channels whose frequencies are low-precision estimates that should be rounded for display purposes
Multichannel Ref Hz	Reference frequency in Hz for each channel in a multichannel measurement
Multichannel Ref Precision	Round reference input frequencies in each measurement channel to the specified # of digits to the right of the decimal point
Multichannel Sample Interval	Time in seconds between successive data points in each channel of a multichannel measurement
Narrowband Mode	Narrow acquisition bandwidth option

New t0	Most recent tau-zero interval specified for resampled phase data
Notes	Additional notes that apply to a measurement, as specified at acquisition time or edited with Edit->Trace Properties
Optional Filters	True if optional filters in use
Options	Option codes associated with instrument
Output Decimation	Final decimation ratio for output samples
Output Sample Rate	Rate at which data samples are generated by the TimeLab driver
Overlap %	Overlap percentage between successive FFT buffers
Overlapped	True if overlapped FFT processing is enabled for faster measurement
Pathname	Pathname under which the .TIM file associated with each plot was most recently saved
Phase	Incoming data consists of phase-difference values
Phase Record	Maximum allocated size of in-memory phase data record used for stability and deviation plots
Pilot	True if pilot tone in use
Pilot Freq	Pilot tone frequency in Hz
PN Max Offset	Maximum offset for SSB noise graph (which will be rounded up to the nearest decade)
PN Min Offset	Minimum offset for SSB noise graph (which will be rounded down to the nearest decade)
Port	IP port of instrument used to acquire data
Port Configuration	Configuration string for RS-232 COM port or other interface, if required
Precision	Precision of incoming floating-point data (e.g., 0=single, 1=double)
Prologix Compatibility Mode	True if serial ports with FTDI D2XX drivers should be treated as Prologix GPIB-USB adapters
Property Page	Selected property page number
Query Instrument ID	True if TimeLab should request the instrument's ID string
Rate Index	Index or other identifier for a given sample interval (device-specific)
Read Existing Data	True if any existing text from the file or device should be skipped prior to acquiring new data
Ref Amplitude	Approximate reference amplitude in dBm
Ref Ch	Physical channel(s) acting as reference
Ref Estimated	True if the reference frequency is a low-precision estimate that should be rounded for display purposes
Ref Freq	Frequency of signal applied to primary reference input jack, often rounded for display
Ref Precision	Round reference input frequencies to the specified # of digits to the right of the decimal point
Remaining	Remaining time in the current measurement or measurement stage
Rescale Factor	Most recent user-supplied rescale factor for phase data
Residual FM	Displays blue vertical cursors in the phase noise graph which can be moved by left/right-clicking with Control held down. The residual FM between these limits of integration will be displayed in units of Hz.
Rev	Detailed information available for plots acquired with certain instruments, including firmware version and/or options
RMS Integrated Noise (Degs)	Displays blue vertical cursors in the phase noise graph which can be moved by left/right-clicking with Control held down. RMS jitter between these limits of integration will be displayed in degrees.
RMS Integrated Noise (Rads)	Displays blue vertical cursors in the phase noise graph which can be moved by left/right-clicking with Control held down. RMS jitter between these limits of integration will be displayed in radians.
RMS Time Jitter	Displays blue vertical cursors in the phase noise graph which can be moved by left/right-clicking with Control held down. RMS jitter between these limits of integration will be displayed in units of seconds.
Role	Selected instrument configuration for a given measurement role
Sample Interval	Time in seconds between successive data points
Sample Rate	Incoming raw data rate from instrument in samples per second, where applicable
Scale Factor	Numeric scale factor used to convert acquired data, e.g., TI readings to seconds or frequency readings to Hz
Scale Jitter to DUT	True if jitter and carrier/noise values should be adjusted to compensate for external frequency conversion
Scale PN to DUT	True if phase noise values should be increased by 20 * log10 (DUT frequency / input frequency)
Scale Stability to DUT	True if stability measurements should be adjusted to compensate for external frequency conversion
Secondary stability plots	True if a copy of the secondary phase record for a cross ADEV measurement should be stored in a separate plot; false if only one plot should be generated
Segment Table	User-specified tag field from segment table used for noise acquisition
Setup String	ASCII text string(s) transmitted to device prior to acquisition
Source A	User-assignable label for first variance source in stability measurement
Source B	User-assignable label for second variance source in stability measurement
Specified Input Frequency	Exact input frequency in Hz provided by user

Specified Reference Frequency	Exact reference frequency in Hz provided by user
Spot Cursor	Displays a red vertical cursor in the graph window which can be moved by left-clicking. The value at the selected column will be displayed in the legend table.
Spur Min Hz	Minimum offset frequency at which spurs will be marked for reporting or removal
Spur Threshold	Minimum amplitude relative to average nearby trace amplitude at which a discrete spectral line will be recognized as a spur
SSB Carrier/Noise	Displays blue vertical cursors in the phase noise graph which can be moved by left/right-clicking with Control held down. The SSB carrier/noise ratio between these limits of integration will be displayed in units of dB.
Stability Ch	Input channel expression(s) for stability measurement
Stability Channel Count	Number of concurrent stability measurements associated with acquisition
Stability Channel ID	Field # or name used to identify channel in incoming data
Stop Condition	Termination condition for measurement
Syx Estimator	Estimator function used to render cross-spectrum FFT bins on log plots
Syx Estimator #	Index or other identifier for cross-spectrum estimator function
Time/Date	Time and date when the acquisition was initiated
Timestamp	Incoming data consists of absolute timestamps in seconds
Trace	Trace caption text, as specified at acquisition time or edited with Edit->Trace Properties
Trace History	Number of historical deviation traces displayed. More recent traces appear darker
Trace Window	Number of phase data points per historical deviation trace. This will equal the phase record size when Trace History is set to 1.
TSC Measurement Type	Noise plot type requested from 51XXA test set or TSERVE
TSC Phase Data Rate	Phase data rate from TSC 512X-compatible timing analyzer
Unwrapped	True if phase data has been unwrapped by the instrument driver
Unwrapped Phase	Incoming data consists of monotonic phase-difference values that do not contain phase wraps
USB Version	USB firmware version
Use Input Rate	True if driver's data rate matches the incoming data rate from the instrument
Use Internal Clock Generator	True if internal clock generator should be enabled
User-Assigned Command	User-assignable command string
Warnings Treated as Errors	True if warnings during data acquisition should terminate the measurement
Wrap Period	Wraparound period in seconds for incoming timestamps

Measurement Menu

The **Measurement** menu selection determines how TimeLab displays the data in any plots that have been acquired or loaded. One measurement type may be selected at a time.

As discussed on page 38, TimeLab plots (and their associated .TIM files) use independent data records for different measurement classes. Some instruments like the TimePod 5330A and PhaseStation 53100A can acquire data for all measurements supported by TimeLab, but most are limited to specific measurement types. For instance, acquisitions made with a frequency counter or TIC may be viewed with the **Allan deviation (**a**)**, **Modified Allan deviation (**m**)**, **Hadamard deviation (**h**)**, **Time deviation (**t**)**, **MTIE (**M**), Phase difference (**p**),** or **Frequency difference (**f**)** selections on the menu, but not **Phase noise (**P**)** or **AM noise (**A**).**

Regardless of the instrument used, all supported measurement records are acquired and processed simultaneously by TimeLab. After launching a PhaseStation 53100A acquisition with all of the available measurements selected in the acquisition dialog, for example, you can switch between all of the entries in the **Measurements** menu to view your data as it arrives.

Plots that don't include the necessary data for the selected measurement type will not be rendered. When the TimeLab window is empty despite the presence of one or more loaded or acquired plots, a brief explanatory message will be displayed in the color associated with each plot to indicate why nothing is being drawn.

See "Making Measurements" on page 43 for more information on the measurement types supported by TimeLab.

Masks Menu

The **Masks** menu is almost entirely user-configurable. It consists of a list of mask definitions for optional use in TimeLab. Masks are useful in production test applications where fast, reliable, and repeatable pass/fail judgments are required. You can also use mask testing for proof-of-performance verification of the 53100A and other instruments supported by TimeLab.

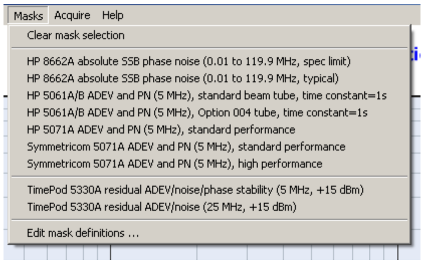

Masks
Acquire
Help
Clear mask selection
HP 8662A absolute SSB phase noise (0.01 to 119.9 MHz, spec limit)
HP 8662A absolute SSB phase noise (0.01 to 119.9 MHz, typical)
HP 5061A/B ADEV and PN (5 MHz), standard beam tube, time constant=1s
HP 5061A/B ADEV and PN (5 MHz), Option 004 tube, time constant=1s
HP 5071A ADEV and PN (5 MHz), standard performance
Symmetricom 5071A ADEV and PN (5 MHz), standard performance
Symmetricom 5071A ADEV and PN (5 MHz), high performance
TimePod 5330A residual ADEV/noise/phase stability (5 MHz, +15 dBm)
TimePod 5330A residual ADEV/noise (25 MHz, +15 dBm)
Edit mask definitions ...

Figure 55: User-defined mask entries

At startup, Timelab constructs the **Masks** menu based on the contents of the user-editable file **masks.txt**.   The folder's location will vary from one Windows version to the next. For example, in a typical Windows 7 installation on drive C: it can be found at **C:\Users\Public\Documents\TimeLab\Masks** at the command prompt, or **Libraries→Public Documents→TimeLab→Masks** in Explorer.

Generally, the easiest way to read and modify **masks.txt** is by selecting the last entry on the **Masks** menu, **Mask→Edit mask definitions**. This option will open **masks.txt** in Windows Notepad. You can then create your own mask definitions, edit existing masks, remove masks from the menu, or simply review the mask-testing features and capabilities in the current version of TimeLab.

Important: To preserve any user-created mask definitions, your existing **masks.txt** file will not be overwritten when a new copy of TimeLab is installed. The only exceptions to this rule occur when new features are introduced that render the mask file format incompatible with newer versions of TimeLab. In such cases, the TimeLab setup program will make a backup copy of **masks.txt** in the same directory, under the name **masks_backup.txt**. Keep in mind that **you will then need to transfer any user-created mask definitions from masks_backup.txt to the new masks.txt file.**

In addition to the example user mask definitions in **masks.txt**, TimeLab also provides standard performance test masks for the PhaseStation 53100A and TimePod 5330A instruments, as well as numerous predefined masks corresponding to telecom industry standards. These mask definitions are stored in a separate file, **standard_masks.txt**, which is overwritten with the latest version at installation time.   Consequently, user-created mask definitions should not be added to **standard_masks.txt**.

Clear mask selection

The first command on the **Masks** menu simply deselects any currently-selected masks, removing its limit line(s) from the applicable measurement views.

User-defined mask entries

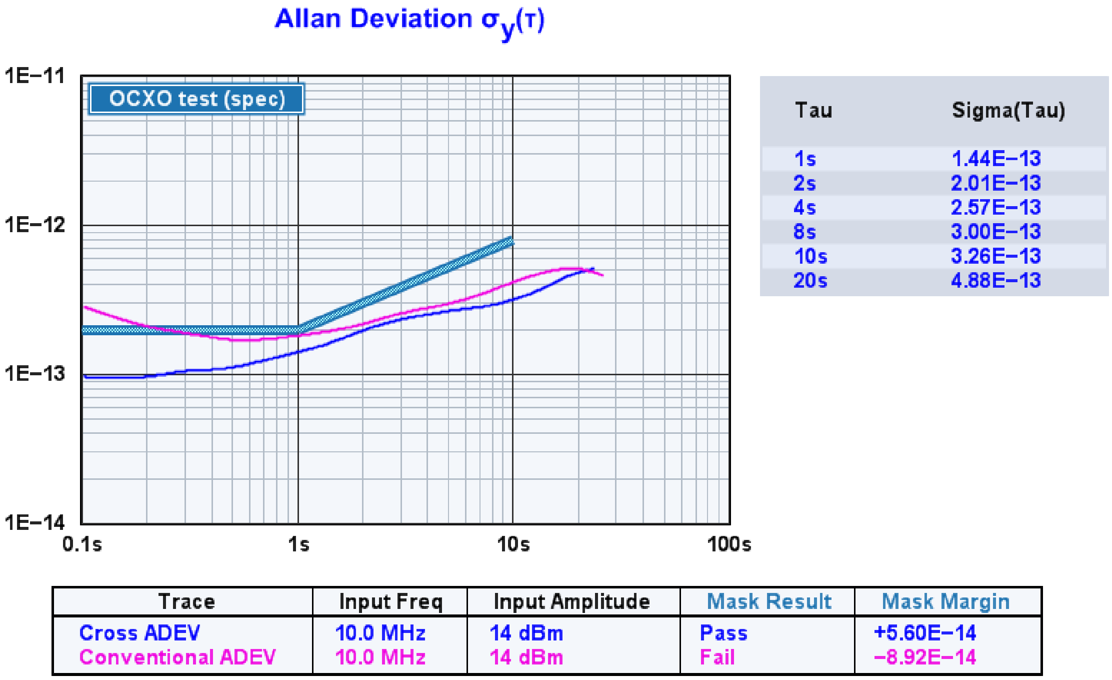
Menu entries appearing between **Mask→Clear mask definitions** and **Mask→Edit mask definitions** are defined in **masks.txt** and **standard_masks.txt**. Up to five masks may be selected at a time. Mask titles and limit lines will be visible only in measurement views corresponding to valid limit lines in a selected mask.

To see the results of the selected mask test, you'll need to select one of these views from the **Measurement** menu *and* enable **Legend→Mask Result**, **Legend→Mask Margin**, or both. Pass/fail results and margins will then appear in the legend table below the graph, updated in real time as the measurement progresses.

For detailed instructions and hints, carefully review the comments in **standard_masks.txt**. The latest mask definition features and syntax for the current TimeLab release will be documented as comments in this file. (Unlike **masks.txt**, the latest version of **standard_masks.txt** is always installed by the setup program.)

Edit mask definitions . . .

As noted above, the last command on the **Masks** menu opens **masks.txt** in Notepad for editing and review.



Tau	Sigma(Tau)
1s	1.44E−13
2s	2.01E−13
4s	2.57E−13
8s	3.00E−13
10s	3.26E−13
20s	4.88E−13

Trace	Input Freq	Input Amplitude	Mask Result	Mask Margin
Cross ADEV	10.0 MHz	14 dBm	Pass	+5.60E−14
Conventional ADEV	10.0 MHz	14 dBm	Fail	−8.92E−14

Figure 56: Mask test with automatic pass/fail evaluation

Scripts Menu

TimeLab includes an embedded JavaScript engine[13] that can run automated test scripts using the PhaseStation 53100A and TimePod 5330A. (Script-based access to other instruments such as time-interval counters is not currently supported.) Numerous API functions are provided to support a variety of applications, ranging from performance verification procedures included with TimeLab to user-developed test scripts. The **Scripts** menu helps you edit, run, and otherwise manage your collection of test scripts.

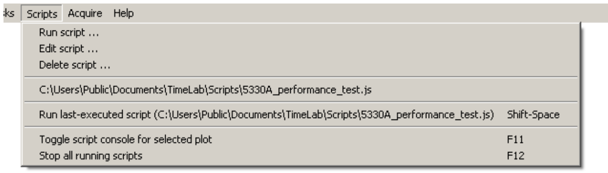

ks
Scripts
Acquire
Help
Run script ...
Edit script ...
Delete script ...
C:\Users\Public\Documents\TimeLab\Scripts\5330A_performance_test.js
Run last-executed script (C:\Users\Public\Documents\TimeLab\Scripts\5330A_performance_test.js) Shift-Space
Toggle script console for selected plot F11
Stop all running scripts F12

Figure 57: Scripts menu

Creating a new test script

The easiest way to create a new test script is to set up your measurement in the 53100A acquisition dialog just as you normally would. Then, rather than selecting **Start measurement**, click the **Save script** button. The Windows file dialog that appears will prompt you to save the current set of measurement parameters to a JavaScript (.js) file. When executed later with **Scripts→Run script**, the script will launch a measurement with the same acquisition parameters and options.

As an alternative, you can use **Scripts→Edit script** to create modified copies of the scripts that are included with TimeLab, such as **53100A_performance_test.js** and **5330A_ADC_test.js**. These programs are more elaborate than the elementary scripts created by **Save script**, but they were all originally created through that function. They exercise virtually all of the capabilities of the TimeLab script API as well as those of the instrument itself.

The script folder's location will vary from one Windows version to the next. For example, in a typical Windows 7 installation on drive C: it can be found at **C:\Users\Public\Documents\TimeLab\ Scripts** at the command prompt, or **Libraries→Public Documents→TimeLab→Scripts** in Explorer.

Examples

In addition to the performance test scripts, TimeLab includes two other examples showing how to accomplish various tasks under script control. **53100A_3_cornered_hat.js** demonstrates how to launch an automated 3-cornered hat stability measurement that will run for 12 hours, periodically saving the results while the test is running, while **53100A_command_line_example.js** illustrates one way to create a complete automated test procedure by launching TimeLab to run a script from an external shell, application, or environment with specified parameters and termination behavior. Review the comments in these scripts to learn how to configure, run, and adapt them.

[13] https://v8.dev

For a detailed listing of script API functions, refer to the JavaScript API Function Reference section on page 145.

Run script …

To launch a test script, select **Scripts→Run script . . .** and choose the desired script's JavaScript file (.js) in the file dialog that appears. TimeLab will load the script into memory and execute its "global scope" to initialize any variables declared outside of functions or event handlers. Control then passes to the script's mandatory **EventRunScript()** function.

Once launched, the script will continue to run until one of the following conditions is true:

- A script exception or other fatal error occurs

- The **ScriptEnd()** API function is called by the script prior to returning from an event handler

- The script returns from an event handler function such as **EventRunScript()** or **EventAcqDone()** with no plots associated with it, either because the user closed the plot(s) manually or because the event handler did not launch any acquisitions. (However, if the script has requested persistent timer service with **TimeSetTimer()**, it will be allowed to continue running.)

- The script calls **FileExitApplication()**

- The **Scripts→Stop all running scripts (**F12**)** command is issued by the user

- The user closes the plot manually, or presses the **Home** key to close all loaded plots

- The user exits from TimeLab

Any errors encountered during either loading or execution will be reported in the status line at the bottom of the TimeLab window.

Edit script …

This command simply launches an instance of Notepad on the specified .js file, allowing you to read, modify, and/or copy the script.

Delete script …

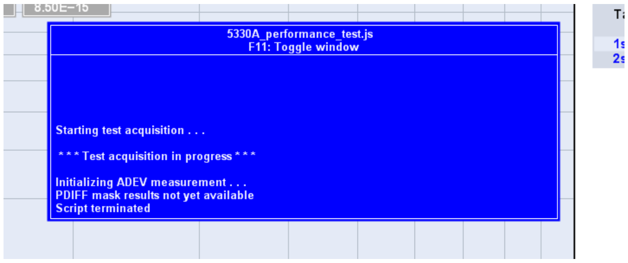
To delete a test script, select **Scripts→Delete script . . .** and specify the .js file to delete. The .js file will be deleted without further prompting.

Run last-executed script (Shift-Space)

Using Shift-Space to re-run the most recently executed script is analogous to pressing the Space bar by itself to bring up the most recently-used acquisition dialog. The script will be relaunched without further prompting from TimeLab.

Toggle script console for selected plot (F11)

When a measurement is started under script control, TimeLab will display an overlay containing any error or status messages issued by the script. This console overlay is shown by default whenever the measurement is otherwise visible on the graph.



```
5330A_performance_test.js
F11: Toggle window

Starting test acquisition . . .

 * * * Test acquisition in progress * * *

Initializing ADEV measurement . . .
PDIFF mask results not yet available
Script terminated
```

Figure 58: The script console

As the script console overlay tends to obscure the data being plotted, the F11 key provides a convenient way to hide or show the overlay for the currently-selected plot.

Stop all running scripts (F12)

Stops any scripts that are currently running, as well as any measurements associated with those script(s).

Acquire Menu

TimeLab's principal mission is to serve as the user interface for the PhaseStation™ 53100A Phase Noise Test Set from Jackson Labs Technologies, but support for many other instruments is also provided. The **Acquire** menu includes a list of all instruments currently recognized by TimeLab, followed by a small number of commands that manage acquisition operations.

When selected, each entry on the **Acquire** menu that corresponds to a given instrument will bring up an *acquisition dialog.* Acquisition dialogs are used to configure the instrument and initiate data collection. Every acquisition in TimeLab runs in its own thread, so you can load or acquire other plots while data collection is in progress. Acquired data can be saved at any time, even during the measurement; likewise, all options on the **Trace** and **Display** menus apply equally to existing plots and those that are currently being acquired.

While acquisition dialogs are straightforward to work with, their layout and functionality varies greatly from one instrument to the next. Each dialog displays extensive help text that's refreshed as the mouse pointer enters each field. This "mouseover" text is the principal source of hardware-specific documentation in TimeLab, since it's always up to date with the latest changes made to each hardware driver. It should be read carefully when using a given instrument or dialog option for the first time.

Jackson Labs PhaseStation / Microchip 53100A

The 53100A instruments use real-time direct sampling techniques from the Software-Defined Radio (SDR) field to measure frequency stability, phase noise, and AM noise at levels substantially beyond the reach of traditional counters and timing analyzers. Depending on the measurement options selected in the 53100A acquisition dialog, the unit can perform any (or all) measurements supported by TimeLab.

The dialog fields used to select and configure the 53100A's supported measurement roles are discussed separately in their own appendix on page 131, *PhaseStation 53100A Measurement Roles.*

Miles Design TimePod and Microsemi 3120A

The TimePod 5330A and Microsemi 3120A have been replaced by the PhaseStation 53100A, but continue to be fully supported by TimeLab. These legacy instruments offer similar performance capabilities at a limited range of carrier and offset frequencies, and provides a subset of the 53100A's connectivity and configuration options.

Acquire from counter in Talk-Only mode

This option allows you to acquire frequency, phase/TI, or timestamp readings from almost any GPIB- or RS232-connected counter that supports a "Talk Only" mode, in which ASCII readings are emitted autonomously with no interaction from a host controller. RS232-based timing analyzers such as the picPET are also supported by the Talk-Only option.

TimeLab supports GPIB connectivity through adapters from National Instruments or Prologix, including Agilent/Keysight GPIB adapters with NI488.2 compatibility enabled.

Acquire from live ASCII file

Similar to the Talk-Only acquisition option described above, this option allows you to specify the location of a text file containing frequency, phase/TI, or timestamp data that's being written by another process. TimeLab will open the file in read-only mode, process all existing data in the file if requested, and then continue to fetch data until the acquisition terminates. Only one reading per line is processed, but any numeric field within the line can be specified for processing.

HP 53131A/53132A/53181A and 53220A/53230A

HP 5335A and 5370A/B

HP 5371A/5372A

Philips/Fluke PM6680

Picotest/Array U6200A series

Stanford Research SR620

Wavecrest DTS-2050/2070 series

These instruments are high-performance counters and timing analyzers whose acquisition capabilities are very similar from TimeLab's perspective. Frequency and phase-difference (TI) readings are supported by each.   Depending on the instrument's I/O capabilities, available connectivity options may include the following:

- National Instruments NI488.2-compatible GPIB adapters
- Serial (COM) ports connected directly to the instrument
- Serial (COM) ports associated with Prologix GPIB-USB adapters
- IP addresses supported with Prologix GPIB-ETHERNET adapters
- Direct connections to TCP/IP-based instruments at the specified address

HP counters with serial interfaces connected via null-modem cables should use **Acquire→Acquire from counter in Talk-Only mode** instead.

Symmetricom 5115A / 5120A / 5125A (Frequency stability)

Symmetricom 5115A / 5120A / 5125A (Phase noise)

These digital phase noise analyzers provide advanced features and specifications similar to the TimePod and PhaseStation instruments. However, separate options in the **Acquisition** menu must be used to obtain data for xDEV and phase/frequency-difference plots and the Fourier spectrum used for phase noise plots. Phase/frequency data will be acquired continuously for the duration of the measurement, while phase noise acquisition will retrieve a snapshot of the analyzer's current phase noise trace, noise-floor estimate, and spur table.

As with other TimeLab acquisition processes, these acquisitions may be launched concurrently.

Note that the earliest versions of these test sets were developed and marketed by Timing Solutions Corporation, so they are often referred to as the "TSC 512x" family of instruments." Later instruments were produced with Symmetricom and Microsemi branding. The TimeLab drivers support all of these variations under the Symmetricom nomenclature.

This acquisition option is also useful for requesting phase noise and AM noise data from the TSERVE remote access server. See *Measurement example: Phase noise plot acquisition* on page 218.

Stop/repeat acquisition (Space)

This context-sensitive command associates the most common acquisition "verbs" with a single easy-to-remember keyboard shortcut.

- If deferred-acquisition mode is active and at least one acquisition is pending , the Space bar terminates all active acquisitions (after prompting).
- Otherwise, if the selected plot is associated with an acquisition in progress, the Space bar will terminate that acquisition (after prompting).
- If one or more acquisitions is running but the selected plot is not among them, the Space bar will offer to terminate the acquisition that was most recently started.
- If no acquisitions were running, the Space bar will bring up the most-recently-used acquisition dialog.

Because TimeLab measurement sessions often involve repeated acquisitions with the same instrument, the Space bar shortcut will usually “do what you mean.” It will either stop collecting data if an acquisition is in progress, or start a new acquisition if data is not currently being acquired.

Abort and retrigger selected acquisition (Ctrl-a)

Keep and retrigger selected acquisition (Ctrl-k)

Like **Acquire→Stop/repeat acquisition (**Space**)**, these commands are normally accessed via their respective keyboard shortcuts. Often used when you accidentally disturb the equipment or remember an omitted step in the measurement procedure, Ctrl-a will restart the selected plot’s measurement immediately with no further interaction. Ctrl-k behaves similarly, but it will not discard the acquired data before relaunching the acquisition.

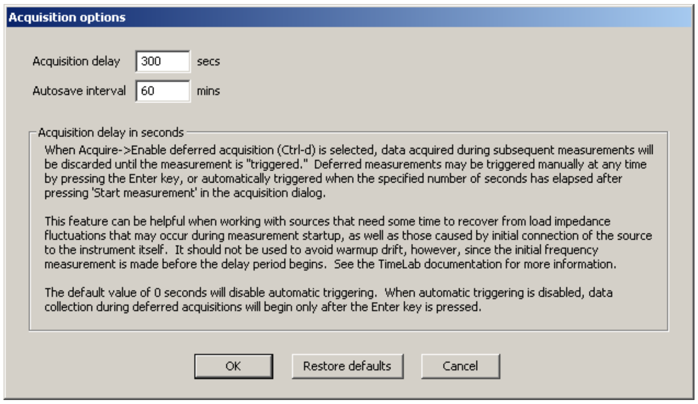
These commands operate as follows:

- If no plots are loaded, either command will bring back the most-recently-used acquisition dialog, as if the Space bar shortcut were used.
- Otherwise, if the selected plot is associated with an acquisition in progress, that acquisition will be terminated. (Any other acquisitions in progress will not be affected.)
- If the **Abort (**Ctrl-a**)** command was issued, the selected plot will be closed and its data discarded. A new acquisition will be started in its place, using the driver associated with the now-deleted plot. The new plot will retain the same position in the legend table.
- If the **Keep (**Ctrl-k**)** command was issued, the selected plot will *not* be closed. A new acquisition will be launched with its instrument driver. The resulting plot will appear in the next available slot in the legend table.

When applied to an acquisition in progress, *both commands work immediately without prompting for confirmation* if **File→Warn before retriggering measurement in progress** is unchecked. They're assigned to control keys to make them harder to press accidentally.

Acquisition options (Ctrl-F7**)**

This command brings up a dialog box that configures TimeLab's autosave and deferred-acquisition functionality.



Acquisition options

Acquisition delay: 300 secs

Autosave interval: 60 mins

Acquisition delay in seconds

When Acquire->Enable deferred acquisition (Ctrl-d) is selected, data acquired during subsequent measurements will be discarded until the measurement is "triggered." Deferred measurements may be triggered manually at any time by pressing the Enter key, or automatically triggered when the specified number of seconds has elapsed after pressing 'Start measurement' in the acquisition dialog.

This feature can be helpful when working with sources that need some time to recover from load impedance fluctuations that may occur during measurement startup, as well as those caused by initial connection of the source to the instrument itself. It should not be used to avoid warmup drift, however, since the initial frequency measurement is made before the delay period begins. See the TimeLab documentation for more information.

The default value of 0 seconds will disable automatic triggering. When automatic triggering is disabled, data collection during deferred acquisitions will begin only after the Enter key is pressed.

Figure 59

When deferred-acquisition mode has been toggled on by the **Acquire→Enable deferred acquisition (**Ctrl-d**)** command, data from subsequent measurements will be discarded until the measurement is "triggered" manually with **Trigger deferred acquisition(s) (**Enter**)** or by cancelling deferred execution with a second **Enable deferred acquisition (**Ctrl-d**)** command. The **Acquisition delay** field in the dialog box above provides an additional way to trigger a deferred measurement. By setting a nonzero **Acquisition delay** value in seconds, a deferred measurement will trigger itself automatically once a specified number of seconds has elapsed after initiation. With the default delay value of zero seconds, manual triggering is necessary in order to begin collecting data. Refer to the **Enable deferred acquisition (**Ctrl-d**)** command description for more information.

The autosave feature helps guard against data loss due to forced operating system updates, power failures, or other unexpected events that may occur during long-term measurements. When the **Autosave interval** field in the dialog box has been set to a value greater than or equal to 1 minute, the .TIM files associated with all subsequent measurements can be periodically saved to disk at the specified interval, just as if the user had issued a **File→Save image or .TIM file (**s**)** command manually.

The autosave interval can be specified at any time before or after the beginning of acquisition. However, only .TIM files can be automatically saved during the measurement, not image files. Additionally, each .TIM file to be automatically saved must be saved manually at least once in order to assign it a valid filename. Measurements in progress that have not been saved manually at least once, or that are running with **Autosave interval** set to a value less than one minute, will not be automatically saved.

Enable deferred acquisition (Ctrl-d)

Trigger deferred acquisition(s) (Enter)

Deferred acquisition mode is toggled with the **Enable deferred acquisition (**Ctrl-d**)** command. In this mode, TimeLab will allow acquisitions to be launched as usual. However, each new measurement's incoming data will be discarded until all pending measurements are *triggered.* In deferred-acquisition mode, triggering occurs at the expiration of a time interval specified in the **Acquisition options (**Ctrl-F7**)** dialog, when the **Trigger deferred acquisition(s) (**Enter**)** command is issued, or when another **Enable deferred acquisition (**Ctrl-d**)** command is issued to return to immediate-acquisition mode. Data will subsequently be collected normally for the duration of the measurement(s).

Deferred-acquisition mode can be useful when you wish to launch multiple concurrent acquisitions – perhaps using different types of instruments with varying latency or setup time requirements – while synchronizing the beginning of data collection as closely as possible across all measurements. Another common application for deferred acquisition is to wait for a given stabilization period to elapse after the instrument's attenuators and filters are configured at the beginning of the measurement process. Load impedance will be relatively constant at all times when using the 'N' female REFERENCE and DUT input jacks on the 53100A, due to the inherent isolation characteristics of the built-in active splitters. However, multichannel measurements will expose the reference and DUT sources to varying load conditions at the SMA-F channel input jacks during the startup phase, so deferred acquisition can be a useful tool in these cases.

Note that TimeLab doesn't specify the trigger latency associated with a group of deferred measurements. Triggered measurements made with time interval counters will normally start collecting valid data within one sample period, but this isn't guaranteed.

Also note that the DUT frequency is initially measured when the measurement begins, not when it is triggered. Attempting to measure an oscillator undergoing startup drift in a narrow measurement bandwidth will usually yield drift warning messages and questionable data regardless of the use of deferred acquisition. Always ensure that the measurement bandwidth is sufficiently wide to keep the DUT signal well within its passband during the deferral period, as well as the remaining duration of the measurement itself.

Help Menu

User guide (F1)

This command launches the system's default Web browser to display http://www.miles.io/timelab/readme.htm (or the equivalent beta page). Links on this page will allow you to download the latest version of TimeLab and access the latest edition of the *PhaseStation 53100A User's Manual* (this file).

When you install TimeLab, a local copy of the PhaseStation 53100A manual is also installed on your hard drive.

About TimeLab

This command displays an "About" box showing the program version, compilation date, and any supported command-line parameters.

Debug mode

TimeLab contains some debugging/development-specific features that are not documented or supported for general use, and are likely to be removed or deprecated in the future. For example, **Measurement→F)requency spectrum** is intended for debugging, rather than as a supported feature, because the TimePod firmware isn't optimized for general spectrum-analysis work and the 53100A doesn't support it at all. These debugging features will be inaccessible if **Help→Debug mode** is turned off.

It's not recommended that you enable this "debug mode" under most circumstances, as it may disable important safeguards against measurement errors and cause other features to function in unexpected ways.

Check for updates (Ctrl-u)

By default, TimeLab will automatically check the Miles Design web site on a weekly basis and inform you if a newer version is available for download. Updates to TimeLab are always free of charge. You can use the **Help→Check for updates (Ctrl-u)** command to reschedule or disable these notifications.

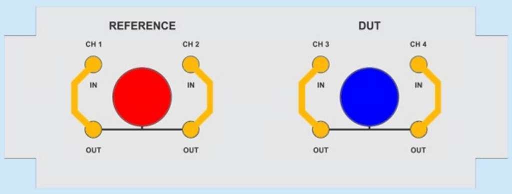
Automatic update checks are never performed during JavaScript program execution.

PhaseStation 53100A Measurement Roles

The PhaseStation 53100A differs from earlier direct-digital instruments in that all four of its internal ADC channel inputs are user-accessible. For simplicity's sake, the 53100A is typically operated in one of several predefined "measurement roles," in which the individual channel inputs and jumper positions are assumed to be configured specifically for the selected task with their per-channel controls populated accordingly, or in a user-defined role in which the channel controls may be freely assigned. Not all of the information needed to take advantage of these capabilities can be accommodated in the acquisition dialog help text, so we'll look at the details here.

Color-keyed diagrams for each role appear below in which **red** corresponds to external reference source connections, **blue** to connections made to the device(s) under test, and **orange** to the SMA jumper positions associated with the role. All of the remaining jacks that do not appear on the diagram for a given measurement role may be left unconnected.

Measure single DUT with single external reference

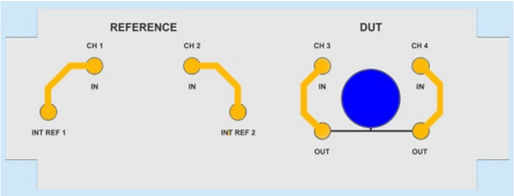


This is the default measurement role for 53100A instruments without the dual 100 MHz internal reference option (Option IR). It is based on the factory-default jumper positions shown above. In this configuration, a single measurement will be recorded when the **Start measurement** button is pressed. Depending on which measurement type(s) are selected, the measurement will include a phase data record based on the phase differences at channels 3 and 1, as well as cross-correlated phase noise and/or AM noise records based on independent acquisition of the signals present at channels 3-1 and 4-2.

The DUT and reference frequencies displayed in the legend table are rounded to the nearest 100 kHz by default. The frequency- and phase-difference plots will represent the measured fractional frequency difference between the actual DUT signal frequency and the rounded DUT frequency shown in the legend table. The frequency count chart will display the frequency of the signal at the first DUT input (3) with accuracy determined by the reference source at channel 1, which is assumed to be a multiple of 100 kHz.

Application note AN 53100A-1, *Oscillator Measurement and Calibration with the PhaseStation 53100A* provides a practical introduction to the single DUT/single reference measurement role, based on a wide variety of real-world test scenarios.

Measure single DUT with dual internal references



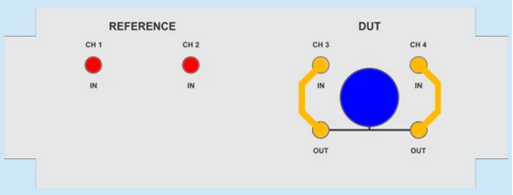
53100A instruments equipped with optional dual internal reference oscillators (Option IR) are configured as shown above by default. These instruments will default to the dual internal reference role if no other role has been selected.

In this role, phase noise and AM noise measurements between channels 3-1 and 4-2 will benefit from cross-spectral suppression of reference noise as well as other instrument noise contributions. Stability measurements, including frequency- and phase-difference plots, HDEV, TDEV, MDEV, and MTIE, will be based on the phase differences between channels 3 and 1 as in the single-reference case. However, the Allan deviation measurement will be performed with phase records from both 3-1 and 4-2 channel pairs, unless the **Always use cross ADEV when possible** box is unchecked in the **Additional options** page.

Also as in the default measurement role, DUT and reference frequencies displayed in the legend table will be rounded to the nearest 100 kHz. The frequency count chart will display the true frequency of the signal at the first DUT input (3) at maximum available precision. Frequency count chart accuracy is determined by the reference source at channel 1, given the assumption that the reference frequency is in fact an exact multiple of 100 kHz.

Refer to the Option IR appendix on page 221 for detailed information on dual-reference measurements. Application Note 53100A-2 is also recommended.

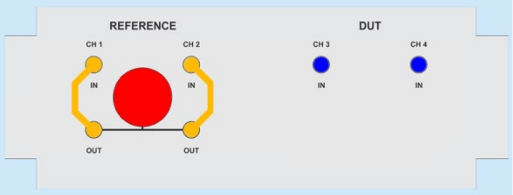
Measure single DUT with dual external references



All 53100A instruments (including those equipped with optional dual internal reference oscillators) can be configured for use with a pair of user-supplied external references by connecting the reference sources to the channel 1 and 2 inputs as indicated above. All of the considerations described above apply, noting that the two external references do not need to operate at the same frequency.

Operationally, the dual internal reference and dual external reference measurement roles are identical. They differ only in the appearance of the graphical connection diagram shown in the acquisition dialog box. Please refer to Application Note 53100A-2 for important additional guidelines regarding external reference selection and usage, as well as the *Choosing reference sources* section on page 28 and the Option IR appendix on page 221.

Measure two DUTs with single external reference



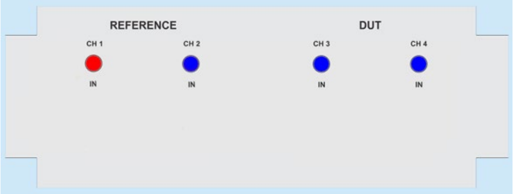
This measurement role will yield two separate stability plots, one based on the phase differences between channel 3 and channel 1 and the other based on the phase differences between channel 4 and channel 1. No noise measurement will be performed in this case.

For each of the two plots, the DUT and reference frequencies displayed in that plot's legend table row will be rounded to the nearest 100 kHz. The frequency-difference traces will represent the fractional frequency differences between the measured DUT signals and the displayed DUT frequencies as usual. The frequency count charts for both plots will inherit the accuracy of the reference oscillator at channel 1, based on the assumption that the reference frequency is an exact multiple of 100 kHz.

Note that because channel 1 acts as a common reference for both stability plots, it's not necessary for the factory default reference-channel jumper configuration to be used as shown above. The reference can be supplied directly to CH 1 IN if desired, with CH 2 IN left unconnected. (One advantage to using the N-channel splitter input, however, is that the reference will not experience significant load-pulling effects due to attenuator and filter selection as the measurement is initialized.)

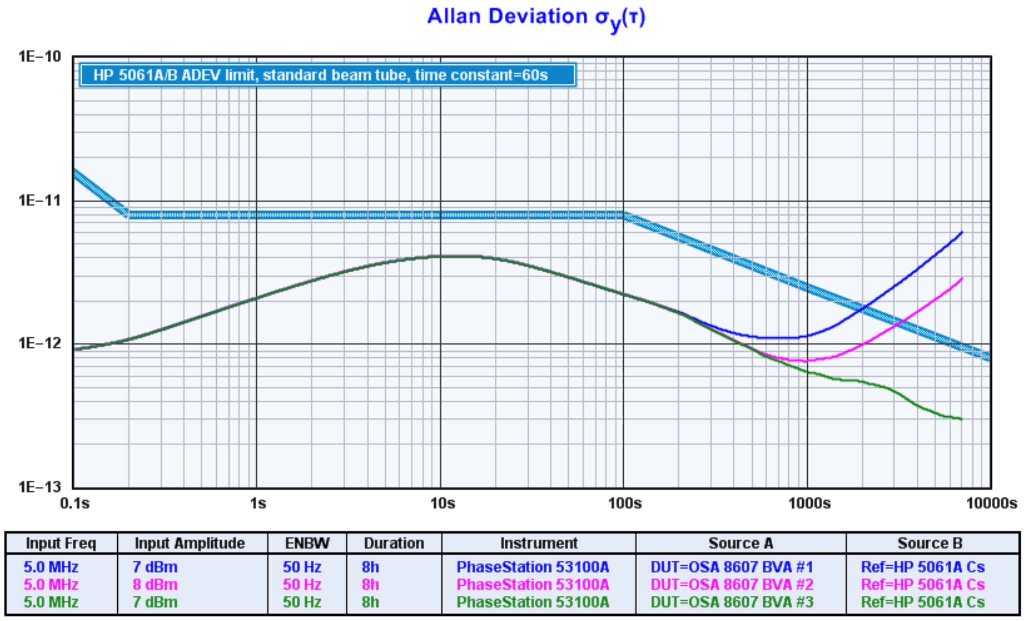
Additionally, if the instrument is equipped with the internal reference option, you may connect CH 1 IN to its adjacent INT REF output jack instead of using an external reference.

Measure three DUTs with single external reference

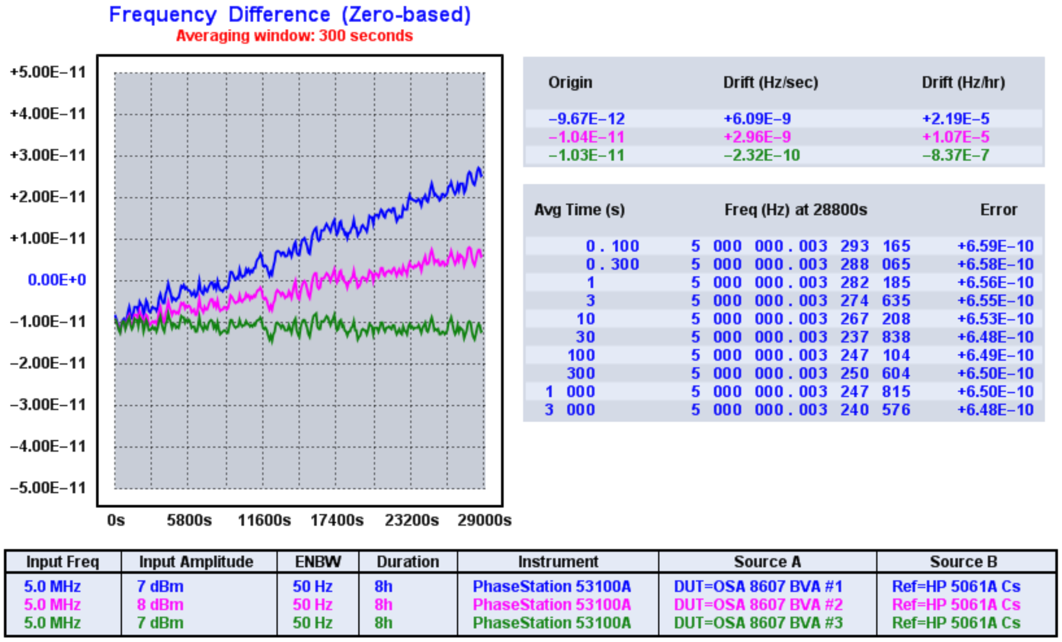


This measurement role is identical to the previous one, except that the CH 2 reference input jack is pressed into service as a third DUT input channel. Three stability plots (and no noise plots) will be generated when the **Start measurement** button is pressed.

As an example, ADEV and frequency-difference plots in which three high-performance crystal oscillators were measured simultaneously against a single cesium-beam frequency standard appear below.
All three plots were generated by the same measurement using the **Measure three DUTs with single external reference** role. The HP 5061A frequency standard was connected to the CH 1 IN jack, while the three Oscilloquartz 8607 BVA oscillators were connected to CH 2 IN, CH 3 IN, and CH 4 IN.



Input Freq	Input Amplitude	ENBW	Duration	Instrument	Source A	Source B
5.0 MHz	7 dBm	50 Hz	8h	PhaseStation 53100A	DUT=OSA 8607 BVA #1	Ref=HP 5061A Cs
5.0 MHz	8 dBm	50 Hz	8h	PhaseStation 53100A	DUT=OSA 8607 BVA #2	Ref=HP 5061A Cs
5.0 MHz	7 dBm	50 Hz	8h	PhaseStation 53100A	DUT=OSA 8607 BVA #3	Ref=HP 5061A Cs



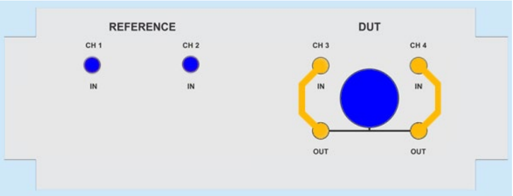
Origin	Drift (Hz/sec)	Drift (Hz/hr)
−9.67E−12	+6.09E−9	+2.19E−5
−1.04E−11	+2.96E−9	+1.07E−5
−1.03E−11	−2.32E−10	−8.37E−7

Avg Time (s)	Freq (Hz) at 28800s	Error
0 . 100	5 000 000 . 003 293 165	+6.59E−10
0 . 300	5 000 000 . 003 288 065	+6.58E−10
1	5 000 000 . 003 282 185	+6.56E−10
3	5 000 000 . 003 274 635	+6.55E−10
10	5 000 000 . 003 267 208	+6.53E−10
30	5 000 000 . 003 237 838	+6.48E−10
100	5 000 000 . 003 247 104	+6.49E−10
300	5 000 000 . 003 250 604	+6.50E−10
1 000	5 000 000 . 003 247 815	+6.50E−10
3 000	5 000 000 . 003 240 576	+6.48E−10

Input Freq	Input Amplitude	ENBW	Duration	Instrument	Source A	Source B
5.0 MHz	7 dBm	50 Hz	8h	PhaseStation 53100A	DUT=OSA 8607 BVA #1	Ref=HP 5061A Cs
5.0 MHz	8 dBm	50 Hz	8h	PhaseStation 53100A	DUT=OSA 8607 BVA #2	Ref=HP 5061A Cs
5.0 MHz	7 dBm	50 Hz	8h	PhaseStation 53100A	DUT=OSA 8607 BVA #3	Ref=HP 5061A Cs

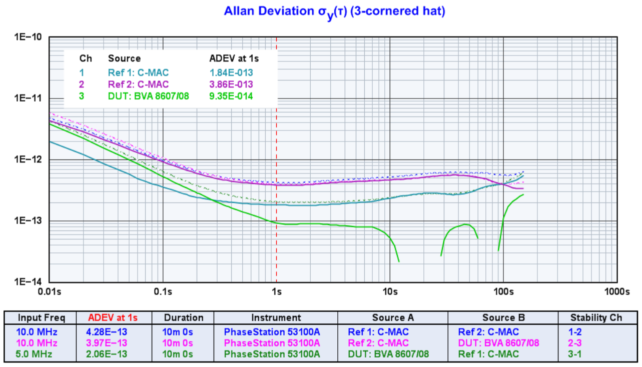
Figure 60: Stability plots of three crystal oscillators measured simultaneously against a cesium-beam standard

Measure three independent DUTs (three-cornered hat)



This measurement role is similar to **Measure single DUT with dual external references** and the other single-DUT roles, n that it is capable of performing a cross-correlated phase noise and/or AM noise measurement of a single device under test that is connected to channels 3 and 4 via an internal or external RF splitter. The phase noise and AM noise contributions from the two independent references connected to channels 1 and 2 – whether internal or external – will be averaged out of the measurement over time.

However, three stability plots are generated in this role rather than only one. In these stability plots, unlike the AM and phase noise measurements, the three connected devices are treated as individual test sources. Each source is represented as a DUT in one of the plots, and each also serves as a reference source for one of the other plots. The resulting trio of mutually-dependent stability plots can be rendered by TimeLab In the usual fashion, appearing as three independent traces, or they can be rendered in a special display mode using the **Trace→Show separated xDEV variances (N-cornered hat) (Ctrl-h)** function (Page 99.) In this mode, TimeLab will attempt to display the stability characteristics of each device by itself, rather than showing traditional stability measurements based on two contributing sources.



Ch	Source	ADEV at 1s
1	Ref 1: C-MAC	1.84E-013
2	Ref 2: C-MAC	3.86E-013
3	DUT: BVA 8607/08	9.35E-014

Input Freq	ADEV at 1s	Duration	Instrument	Source A	Source B	Stability Ch
10.0 MHz	4.28E−13	10m 0s	PhaseStation 53100A	Ref 1: C-MAC	Ref 2: C-MAC	1-2
10.0 MHz	3.97E−13	10m 0s	PhaseStation 53100A	Ref 2: C-MAC	DUT: BVA 8607/08	2-3
5.0 MHz	2.06E−13	10m 0s	PhaseStation 53100A	DUT: BVA 8607/08	Ref 1: C-MAC	3-1

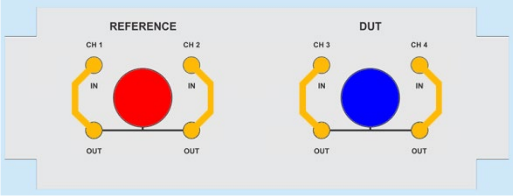
Figure 61: Three-cornered hat display

Note that any phase noise/AM noise data obtained in this measurement role will be associated only with the first of the three stability plots. The remaining two plots will not contain any noise data.

As with the other dual-reference measurements, please refer to Application Note 53100A-2 for a discussion of three-cornered hat measurements in TimeLab with detailed examples.

If a three-cornered hat measurement is being performed in order to achieve the lowest possible measurement floor for a single device, consider using one of the dual-reference measurement roles instead. A single .tim file contains all of the information necessary to display a cross ADEV measurement, eliminating the need to save all three files separately. Additionally, cross ADEV measurements improve the 53100A's own instrument floor over time, rather than only its references. See the **Cross Allan Deviation Measurements** section on page 139 for more details.

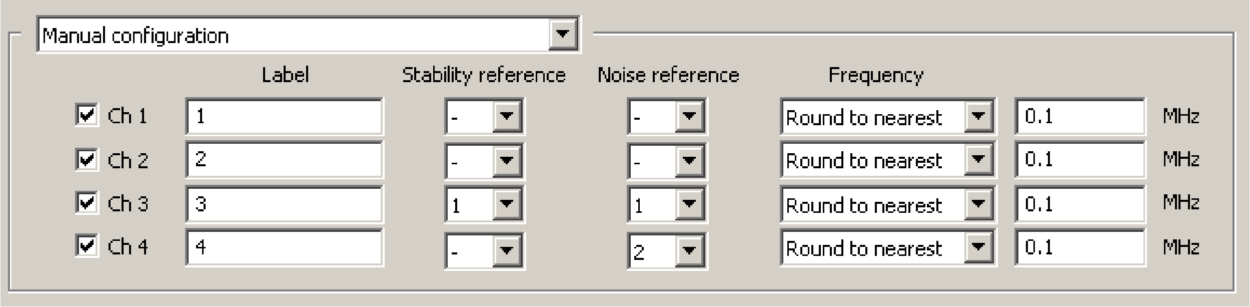
Measure single DUT with cross ADEV statistics



Selecting this role will cause the 53100A to perform a single-DUT stability measurement as in **Measure single DUT with single external reference**, using data from both the 3-1 and 4-2 channel pairs for cross ADEV computations even if the **Always use cross ADEV when possible** box is unchecked in the **Additional options** page.

Since this box is checked by default, it's normally unnecessary to select this role manually. See the **Cross Allan Deviation Measurements** section on page 139 for more details.

Manual configuration



Manual configuration

	Label	Stability reference	Noise reference	Frequency		
☑ Ch 1	1	-	-	Round to nearest	0.1	MHz
☑ Ch 2	2	-	-	Round to nearest	0.1	MHz
☑ Ch 3	3	1	1	Round to nearest	0.1	MHz
☑ Ch 4	4	-	2	Round to nearest	0.1	MHz

Figure 62: Manual configuration controls

As shown above, selecting the **Manual configuration** measurement role will enable all of the dialog fields associated with channel assignment and frequency specification that are normally grayed out when working with predefined measurement roles. Any changes you make to these fields will be saved for recall whenever the **Manual configuration** role is selected in the future, as long as the **Restore defaults** button has not been pressed since the changes were made.

Most 53100A users will not need to configure these fields manually, as the predefined roles should cover the most common measurement scenarios. Cases in which manual configuration should be considered include:

- You need to connect DUT input jacks to reference sources, or vice versa

- You need to perform noise measurements with nonstandard channel input connections

- You wish to display DUT and/or reference frequencies in the legend table with more than the default 100-kHz rounding precision

- The nominal frequency of your reference source is not a multiple of 100 kHz

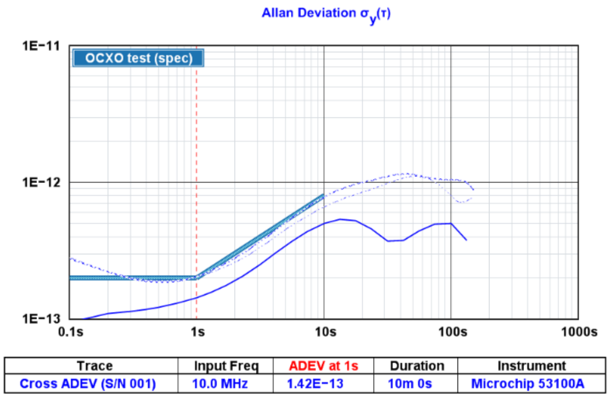
- You wish to specify precise DUT or reference input frequencies for any other reason

In these scenarios, the **Manual configuration** role provides the required flexibility at some cost in complexity. As with other fields in the acquisition dialog, detailed online help can be obtained by hovering over any of the controls associated with manual role configuration.

Note that cross ADEV measurements may be performed in the **Manual configuration** role only if two conditions are met. **Always use cross ADEV when possible** must be checked in the **Additional options** page, and both channels 3 and 4 must be assigned unique stability reference channels (typically 1 and 2 respectively.)

Cross Allan Deviation Measurements

Stability measurements performed by the 53100A normally rely on phase difference measured at a single ADC channel pair, typically based on a DUT connected to port 3 and a reference source connected to port 1. Measurements in this class include the frequency-difference and phase-difference “strip chart” plots, as well as modified Allan deviation (MDEV), time deviation (TDEV), Hadamard deviation (HDEV), and maximum time interval error (MTIE) graphs.



Trace	Input Freq	ADEV at 1s	Duration	Instrument
Cross ADEV (S/N 001)	10.0 MHz	1.42E−13	10m 0s	Microchip 53100A

Figure 63

Allan deviation (ADEV) measurements made with the 53100A, however, can take advantage of all four ADC channels. Just as phase noise and AM noise measurements benefit from instrument- and reference-noise improvement through cross-spectral averaging over time, cross Allan deviation measurements can help to overcome the stability limitations imposed by both the reference oscillators and the 53100A’s own internal ADCs. Cross Allan deviation is not a separate measurement type with its own statistical basis; it is simply the result of computing the ADEV bin variances based on two independent phase measurements taken simultaneously, rather than squaring the phase differences obtained from a single measurement.

By default, the 53100A performs cross ADEV measurements in all single-DUT measurement roles in which both DUT channels 3 and 4 have stability reference channels assigned to them. These roles include **Measure single DUT with dual external references**, **Measure single DUT with dual internal references**, and **Measure single DUT with cross ADEV statistics**. Stability measurements made in all of these cases will include cross ADEV plots as long as the **Always use cross ADEV when possible** checkbox remains selected on the **Additional options** page of the 53100A’s acquisition dialog. If this box is unchecked, cross ADEV measurements are performed only when the **Measure single DUT with cross ADEV statistics** role is explicitly selected. (The **Manual configuration** role can perform cross ADEV measurements only under certain conditions; refer to its description in the **PhaseStation 53100A Measurement Roles** section for details.)

When cross ADEV measurements are performed, two independent phase records are acquired during the measurement rather than only one. Both of these records are associated with the resulting plot and are stored in a single .TIM file. TimeLab uses data from both of these records when rendering a cross ADEV plot. The record associated with the 3-1 DUT-reference combination is considered the *primary* record, while the 4-2 channel pair contributes to the *secondary* phase record.

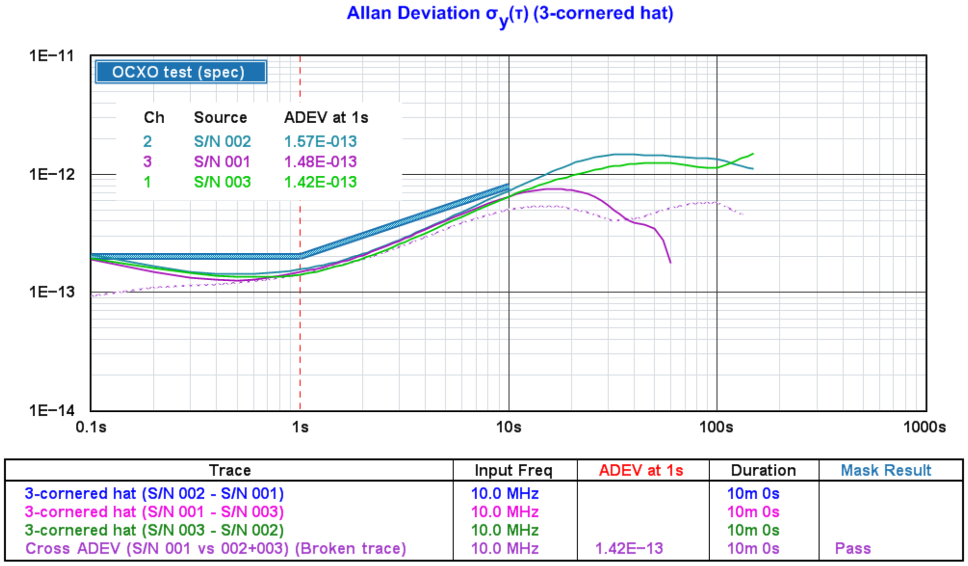
You can inspect the conventional Allan deviation traces for each record separately with the F6 hotkey (**Trace→Show original traces in computed xDEV displays**). When this function is turned on, as shown in

Figure 62 above, two dashed (broken) traces will appear in addition to the composite cross ADEV trace, assuming that cross ADEV displays have been enabled with **Trace→Show cross ADEV traces when available (**Ctrl-j**)**. The trace corresponding to the primary (3-1) channel pair is slightly thicker than the other trace which represents the secondary (4-2) channel pair.

The improvement in measurement floor performance can be substantial when both ADC channel pairs contribute. Figure 62 depicts a short-term stability measurement of a high-performance oscillator using two other examples of the same part as independent references. The two broken traces indicate what would be observed if the **Always use cross ADEV when possible** option had been unchecked at the time the measurement data was acquired. (In fact, the heavier of the two broken traces also reflects what would be seen if **Trace→Show cross ADEV traces when available (**Ctrl-j**)** were used to toggle off the cross ADEV display.)

Two distinct advantages of cross ADEV measurement can be observed here. First, the use of all four ADC channels rather than only two improves the performance dramatically between t=0.1s and t=1.0s. The conventional ADEV traces from the high-performance oscillator under test reveal the 53100A's own performance floor in this region, while in the cross-ADEV trace, uncorrelated noise from the two independent ADC channel pairs has been averaged out of the measurement over time.

Second, as noted earlier, the averaging process also helps to cancel uncorrelated noise from the two independent reference oscillators. At taus beyond 1s where instrument noise doesn't come into play, the result is reminiscent of what would be obtained from a three-cornered hat measurement (Figure 64).



Ch	Source	ADEV at 1s
2	S/N 002	1.57E-013
3	S/N 001	1.48E-013
1	S/N 003	1.42E-013

Trace	Input Freq	ADEV at 1s	Duration	Mask Result
3-cornered hat (S/N 002 - S/N 001)	10.0 MHz		10m 0s	
3-cornered hat (S/N 001 - S/N 003)	10.0 MHz		10m 0s	
3-cornered hat (S/N 003 - S/N 002)	10.0 MHz		10m 0s	
Cross ADEV (S/N 001 vs 002+003) (Broken trace)	10.0 MHz	1.42E−13	10m 0s	Pass

Figure 64

Given sufficient measurement time, it's likely that the violet traces would approach each other closely at taus greater than 1s, but only the cross ADEV technique can remove the instrument noise at shorter-term taus. Compare the violet traces in Figure 64 with the two dark blue traces in Figure 63 to see this effect. The three-cornered hat solution remains limited by the 53100A's noise floor below t=1s, and it also does not support limit-line evaluation. Finally, three-cornered hat measurements also require three different .TIM files to be saved and reloaded for later inspection, making them less user-friendly than the cross ADEV approach in scenarios where only one DUT needs to be measured.

Although this technique is capable of exceptional performance, there are also some subtle conditions and caveats associated with cross ADEV measurements. Residual measurements of quiet DUTs (or the instrument itself) may show substantially more variance than a conventional ADEV measurement would. Cross ADEV plots are not subject to the fragmentation often seen in three-cornered hat measurements, but in cases where the actual device performance is similar to or better than that of the instrument itself, cross ADEV plots can display numerous divots and valleys at taus where the measurement has not yet had time to settle. This effect is unlikely to be of concern when measuring real-world frequency standards and sources, even at the most demanding performance levels, **but may be a consideration in two-port residual ADEV measurements where it risks underrepresenting the DUT's true stability performance.** Discretion is required when interpreting these plots.

Trace→Draw xDEV traces with spline interpolation (i) may be used if desired to smooth cross ADEV traces, but be aware of the potential for artifacts here as well. While the accuracy at taus corresponding to the actual bin locations is unaffected by spline interpolation, it may exaggerate the apparent trace amplitude elsewhere. Overshoot can be expected near high-variance degenerate bins when spline interpolation is enabled. Once again, longer measurements yield smoother traces.

In all other respects, cross ADEV measurements are similar to those obtained in the other measurement roles. When a single reference source is used, the primary and secondary phase- and frequency-difference records will be virtually identical to each other except for differences in instrument noise (although the secondary phase record is accessible only when the **Additional options→Create secondary-channel plots** box is checked in the 53100A acquisition dialog.) Phase noise and AM noise measurements are unaffected by the cross ADEV process or any other aspects of stability measurements. They will be performed as usual, based on the 3-1 and 4-2 channel pairs by default.

TimeLab's cross ADEV implementation can be thought of as a four-channel application of the Groslambert Covariance (GCov) methodology.[14] [15] As a general guideline, traditional three-cornered hat tests should be performed only when three concurrent measurements of independent sources are actually needed. When only one source needs to be measured against either one or two references, the cross ADEV role will almost always return superior results in less measurement time.

[14] Vernotte and Lantz, Three-Cornered Hat and Groslambert Covariance: A first attempt to assess the uncertainty domains

[15] Stein, The Allan Variance – Challenges and Opportunities

Two-Port Residual Measurements

Residual noise and stability measurements with the 53100A can be conveniently performed in the **Measure single DUT with single external reference** or **Measure single DUT with cross ADEV statistics** role. Typically a two-way 0° RF splitter is used to feed a signal from a common reference source to both the REFERENCE jack and the input of the device under test. The device's output, in turn, is connected to the 53100A's DUT input jack. To the extent the device contributes its own phase drift, phase noise, or AM noise to the stimulus signal passing through it, the impact will be evident on the respective measurement traces.

Residual measurements are an integral part of the 53100A production and maintenance processes. Each 53100A ships with a test report showing the result of a two-hour residual measurement with 5 MHz and +15 dBm applied to the DUT and REFERENCE input ports via a Mini-Circuits ZFSC-2-1 splitter. The following notes will be helpful when conducting your own proof-of-performance measurements.

- At 50 Hz ENBW, residual ADEV at 5 MHz should be in the high E-14s at t=1s. Thermal effects may push the slope out somewhat, especially in an unstable environment or when warmup time has been insufficient. A look at the **p)hase difference** graph will often show a consistent phase slope for the first 30-60 minutes of a measurement made after a cold start.

- Typical results near 5E-15 at t=1s are achievable using the cross ADEV role at 5 Hz ENBW. This technique can improve the short-term measurement floor by up to 10x in cases where the 53100A's ADC noise is either limiting the available performance or is being measured itself.

- Residual PN at 5 MHz should be below -145 dBc/Hz at 10 Hz after a few minutes.

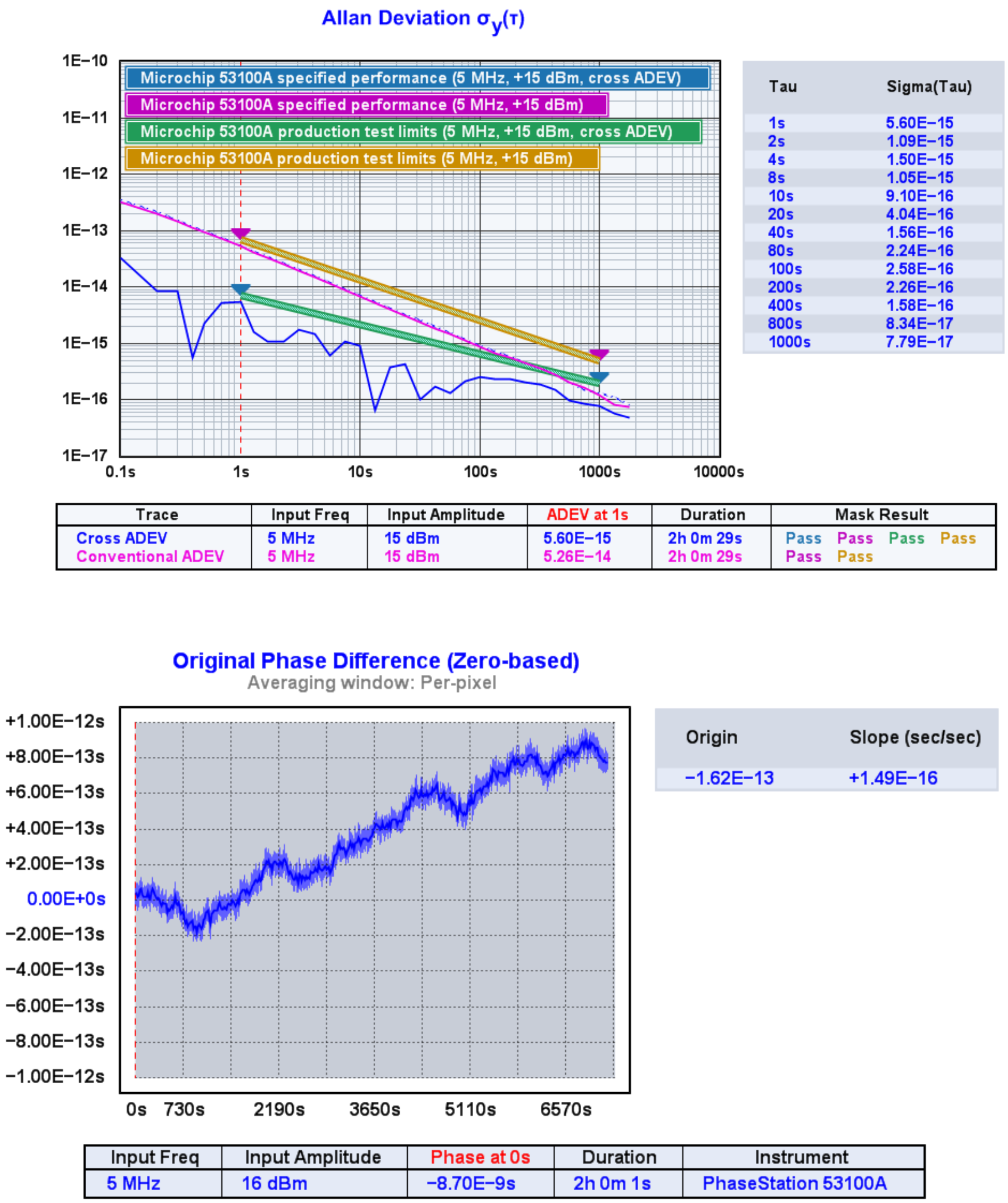
- With a strong, clean input signal at 5 to 10 MHz, the residual PN floor in the 10 kHz – 1 MHz decades will usually end up near -180 dBc/Hz after less than two hours. Expect lower performance near either end of the supported frequency range. When possible, both proof-of-performance tests and actual two-port device measurements should be made in the 5-10 MHz range where the 53100A's own residual performance is optimal.

- If the PC can support overlapped FFT processing, consider enabling this option by checking the **Additional options→Overlapped acquisition** box in the 53100A acquisition dialog. Very low close-in noise levels will require much less measurement time when overlapping is enabled.

- Phase hits and large outliers in the frequency-difference trace are never normal in a residual test with a clean source signal. They should be investigated if they recur with no obvious explanation. Spurs in phase noise plots should not exceed -100 dBc in a 5 MHz residual measurement, and will normally be much lower.

- High-quality double-shielded cables and coaxial interconnects are vital for low-noise, high-stability measurements. Cabling used in residual tests should be kept as short as possible.

Some typical residual instrument performance test results are reproduced below.



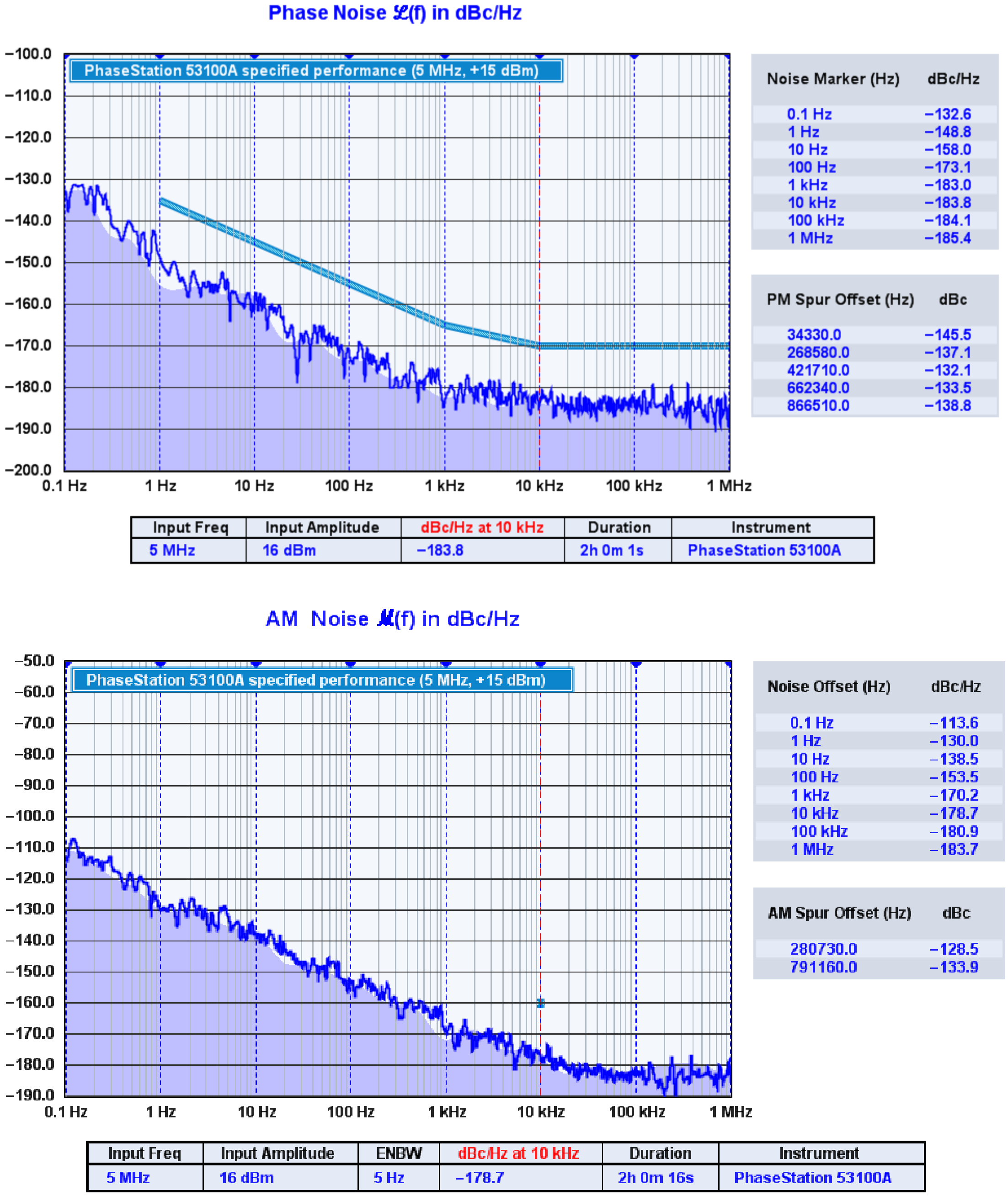
Tau	Sigma(Tau)
1s	5.60E−15
2s	1.09E−15
4s	1.50E−15
8s	1.05E−15
10s	9.10E−16
20s	4.04E−16
40s	1.56E−16
80s	2.24E−16
100s	2.58E−16
200s	2.26E−16
400s	1.58E−16
800s	8.34E−17
1000s	7.79E−17

Trace	Input Freq	Input Amplitude	ADEV at 1s	Duration	Mask Result
Cross ADEV	5 MHz	15 dBm	5.60E−15	2h 0m 29s	Pass Pass Pass Pass
Conventional ADEV	5 MHz	15 dBm	5.26E−14	2h 0m 29s	Pass Pass

Origin	Slope (sec/sec)
−1.62E−13	+1.49E−16

Input Freq	Input Amplitude	Phase at 0s	Duration	Instrument
5 MHz	16 dBm	−8.70E−9s	2h 0m 1s	PhaseStation 53100A

Figure 65: 5 MHz residual ADEV and phase drift



Noise Marker (Hz)	dBc/Hz
0.1 Hz	−132.6
1 Hz	−148.8
10 Hz	−158.0
100 Hz	−173.1
1 kHz	−183.0
10 kHz	−183.8
100 kHz	−184.1
1 MHz	−185.4

PM Spur Offset (Hz)	dBc
34330.0	−145.5
268580.0	−137.1
421710.0	−132.1
662340.0	−133.5
866510.0	−138.8

Input Freq	Input Amplitude	dBc/Hz at 10 kHz	Duration	Instrument
5 MHz	16 dBm	−183.8	2h 0m 1s	PhaseStation 53100A

Noise Offset (Hz)	dBc/Hz
0.1 Hz	−113.6
1 Hz	−130.0
10 Hz	−138.5
100 Hz	−153.5
1 kHz	−170.2
10 kHz	−178.7
100 kHz	−180.9
1 MHz	−183.7

AM Spur Offset (Hz)	dBc
280730.0	−128.5
791160.0	−133.9

Input Freq	Input Amplitude	ENBW	dBc/Hz at 10 kHz	Duration	Instrument
5 MHz	16 dBm	5 Hz	−178.7	2h 0m 16s	PhaseStation 53100A

Figure 66: 5 MHz residual phase noise and AM noise

Note that the 53100A's Allan deviation performance limits are specified for both conventional and cross ADEV measurement roles. The best achievable ADEV performance floor is obtained in cross ADEV mode. Refer to the **Cross Allan Deviation Measurements** section on page 139 for more information and examples.

JavaScript API Function Reference

TimeLab includes an embedded JavaScript engine[16] that can run automated test scripts using the TimePod 5330A and PhaseStation 53100A. The API functions listed below are useful in a variety of applications ranging from performance verification procedures to user-developed test scripts.

Usage examples for many of these API functions can be found in the **53100A_performance_test.js** script included with TimeLab, as well as **5330A_performance_test.js** and **5330A_ADC_test.js**.

In addition to the user performance test scripts, TimeLab includes two other examples showing how to accomplish various tasks under script control. **53100A_3_cornered_hat.js** demonstrates how to launch an automated 3-cornered hat stability measurement that will run for 12 hours, periodically saving the results while the test is running, while **53100A_command_line_example.js** illustrates one way to create a complete automated test procedure by launching TimeLab to run a script from an external shell, application, or environment with specified parameters and termination behavior.

Review the comments in these scripts to learn how to configure, run, and adapt them. For more information on creating, editing, and running test scripts, refer to the **Scripts** menu description on page 120.

AcqChannel([Number user_value,] Number channel)

> Returns the position in the legend table beneath the graph, from 0 to 8, of the plot associated with the script instance's specified acquisition channel. Since plots being acquired under script control may be moved up and down in the legend table by the user like any other loaded plots, **AcqChannel()** is needed to obtain the value needed when calling **ScriptBindToPlot()** in multichannel acquisitions.
>
> The **channel** parameter should range from 0 to one less than the value returned by **AcqNumChannels()**. If two parameters are passed, the first is the **user_value** that was originally passed to **AcqStartAcquisition()** when the plot at the desired index was created.
>
> Scripts that perform only standard single-channel acquisitions will not need to use this function.

AcqCheckOptions(String option_name [{, String option_name}])

> This function is reserved for compatibility with Symmetricom 3120A test scripts. It always returns TRUE in scripts associated with TimePod 5330A and PhaseStation 53100A measurements.

[16] https://v8.dev

AcqClearParams()

Each script instance maintains a dictionary of acquisition parameters that can be used to override any or all of the acquisition dialog parameters when **AcqStartAcquisition()** is called. **AcqClearParams()** clears this dictionary. It is normally called before issuing the **AcqSetParam()** call(s) needed to configure the measurement.

Acquisition parameters that the script does not explicitly set with **AcqSetParam()** will assume their default values when the measurement begins.

See **AcqNumParam()**, **AcqStrParam()** and **AcqSetParam()** for more information.

AcqFreqMHz([Number avg_secs])

This function retrieves the fractional frequency error from the count chart in the frequency-difference measurement view. The fractional frequency error is based on the value displayed in the legend table's **Input Freq** field, which is usually (but not necessarily) equal to the DUT frequency.

The averaging window specified by **avg_secs** begins at the most recently-received data point at the right end of the trace and extends backward in time. Omitting **avg_secs** or specifying a value of 0 will return the average fractional frequency error for the entire phase record.

AcqFreqErr([Number avg_secs])

Same as **AcqFreqErr** but returns the absolute frequency count in MHz over the specified window.

AcqGetAMSpurTable(Array offset_array, Array amplitude_array)

Populates **offset_array** and **amplitude_array** with a list of all detected AM spurs in the current measurement, returning the total number of AM spurs.

AcqGetPMSpurTable(Array offset_array, Array amplitude_array)

Populates **offset_array** and **amplitude_array** with a list of all detected PM spurs in the current measurement, returning the total number of PM spurs.

AcqGetXDEVBins(String selection, Array tau_array, Array sigma_array)

AcqGetXDEVChart(Array tau_array, Array sigma_array)

Populates **tau_array** and **sigma_array** with a list of sigma(tau) values from all valid bins in the specified statistical measurement, or from the most recently rendered sigma(tau) chart, depending on which function is called. The selection parameter passed to **AcqGetXDEVBins()** should be a string beginning with ‘a’, ‘h’, ‘m’, or ‘t’, corresponding to

Allan deviation, Hadamard deviation, modified Allan deviation, or time deviation, respectively.

The **AcqGetXDEVBins()** function differs from **AcqGetXDEVChart()** in that **AcqGetXDEVChart()** returns results only for the 1, 3, 10 tau multiples beginning at t=1s that were displayed in the most-recently-visible numeric table for the measurement. Before calling **AcqGetXDEVChart()**, you should use **MeasurementSelectView()** or **MeasurementDeferSelectView()** to display the statistical measurement for which the chart contents should be returned. Additionally, **TraceShowNumTab(true)** should be called to ensure that the numeric table is visible, and the script must allow time for at least one frame to be rendered. **AcqGetXDEVBins()** does not have any of these constraints.

AcqNumAcquiredPoints()

Returns the number of acquired points in the phase data record.

AcqNumChannels()

Returns the number of acquisition channels in the measurement associated with the script instance.

In scripts that perform only standard single-channel acquisitions, this function always returns 1.

AcqParam(String key)

AcqStrParam(String key)

AcqNumParam(String key)

Each script instance maintains a dictionary of acquisition parameters that can be used to override any or all of the default acquisition dialog parameters when **AcqStartAcquisition()** is called. **AcqStrParam()** and **AcqNumParam()** return string or numeric values, respectively, for the specified acquisition parameter.

If an existing plot is associated with the script instance, these functions will return the specific value associated with the plot. Otherwise, if the script has not yet started a measurement, **AcqStrParam()** and **AcqNumParam()** will return the value from the dictionary of initial parameters that will be passed to **AcqStartAcquisition()**. In either case, an attempt to look up an unrecognized parameter value will return an empty string.

The **AcqParam()** function is equivalent to **AcqStrParam()**, and is deprecated as a function name. New scripts should use **AcqStrParam()** or **AcqNumParam()** instead, depending on whether the parameter should be evaluated as a numeric value or as an arbitrary string.

See **AcqSetParam()** and **AcqClearParams()** for more information.

AcqReadAMNoiseTrace(Number offset_Hz)

AcqReadPMNoiseTrace(Number offset_Hz)

These functions return the current AM noise or phase noise level in dBc/Hz at the specified offset from the carrier. A value greater than 0 indicates that data is unavailable, either because the AM noise or phase noise trace has not yet been displayed or because noise data from the FFT segment containing the specified offset frequency is not yet available.

Values returned by these functions are based on the most recent AM noise or phase noise trace rendered.

AcqReadAMNoiseFloor(Number offset_Hz)

AcqReadPMNoiseFloor(Number offset_Hz)

These functions return the estimated AM noise or phase noise floor in dBc/Hz at the specified offset from the carrier. A value greater than 0 indicates that data is unavailable, either because the AM noise or phase noise trace has not yet been displayed or because noise data from the FFT segment containing the specified offset frequency is not yet available.

Values returned by these functions are based on the most recent AM noise or phase noise trace rendered. The noise floor values undergo spline smoothing at a relatively low resolution, and should be treated as rough estimates rather than definitive limits.

AcqSetParam(String key, String value)

Each script instance maintains a dictionary of acquisition parameters that can be used to override any or all of the acquisition dialog parameters when **AcqStartAcquisition()** is called. **AcqSetParam()** updates the stored value for the specified acquisition parameter.

The best way to obtain a list of measurement parameter names and values for a given instrument is to use the **Save Measurement Script** button in the **Utility** tab of the acquisition dialog to create a script that will contain explicit **AcqSetParam()** statements for each supported parameter. Almost all parameters will have obvious counterparts in the instrument's acquisition dialog; you can also refer to the **Legend** menu section on page 112 for individual descriptions.

See **AcqStrParam()**, **AcqNumParam()** and **AcqClearParams()** for more information.

AcqStartAcquisition(String menu_entry, Boolean skip_dialog [, Number user_value])

Starts a new measurement based on the script instance's current set of acquisition parameters.

The menu_entry parameter should correspond *exactly* to the name of the instrument as it appears on the **Acquire** menu, including any ellipsis (...) that follows it. Currently *Miles Design TimePod ...*, *Jackson Labs PhaseStation …*, and *Microchip 53100A …* are the only officially-supported instruments for scripted measurements, so one of these strings must be supplied as the menu_entry value.

With a script-initiated measurement, the initial acquisition parameters are the same as the default values in a fresh TimeLab installation. These default parameters are then overwritten with any entries that appear in the script instance's acquisition-parameter dictionary, as described in AcqNumParam(), AcqStrParam(), and AcqSetParam(). Unlike a manual measurement, script-based measurements do not write their parameters back to the driver's .INI file. As a result, the dialog defaults that will appear in the next user-initiated acquisition will not reflect the parameters used for the last scripted acquisition.

Next, if skip_dialog is false, TimeLab will present the acquisition dialog to the user, allowing any parameters to be changed before the measurement begins. In most cases, skip_dialog should be set to true to allow the script to run without further user intervention. This will be the case for measurement scripts created with the **Save script** button in the acquisition dialog.

The AcqStartAcquisition() function will return true if the acquisition is initiated successfully, or false if an error occurred. Reasons for failure might include the lack of an available slot in the legend table for the plot(s) created by the acquisition, the use of an unrecognized menu_entry name, the specification of an invalid acquisition parameter value with AcqSetParam(), or (if skip_dialog is false) the user's selection of the **Cancel** button.

Once the measurement begins collecting data, the script's EventAcqTriggered() event handler (if any) will be called. Prior to this stage, the script's EventAcqSetup() handler will be called if one is defined. Because AcqStartAcquisition() returns immediately after launching the acquisition thread, measurement-specific parameters such as **Input Freq** will not be available to the script until EventAcqTriggered() is called.

Optionally, a third parameter may be passed, consisting of an arbitrary numeric user_value which will be made available to subsequent event handlers.

AcqStopAcquisition()

Immediately stops any acquisition associated with the running script instance and calls the script's EventAcqDone() handler (if any). Unless a fatal error occurs, the plot(s) will not be closed and the script will continue to run until ScriptEnd() is called.

AcqSetupEnd()

Applications that include EventAcqSetup() handlers must call AcqSetupEnd() when their setup phase is complete. Until AcqSetupEnd() is called, the 53100A driver will remain in an endless loop awaiting further setup commands from the script.

DisplayFontSize([Number points])

Sets or returns the size of the main display font size used in the TimeLab window. This is the same value that is adjusted by the commands on the **Display** menu.

If no parameter is specified, the current font size is returned without alteration.

DisplayNoiseMarkers([String marker_list])

Returns, and optionally clears or specifies, a comma-separated list of noise marker Fourier offset frequencies for the **Measurement→Phase Noise** and **Measurement→AM Noise** views.

If the marker_list argument is empty, the noise marker list will be cleared. Otherwise, the submitted string is treated as a list of offset frequencies that will replace the existing list (if any.) In all cases, including when the argument is omitted entirely, the previous set of noise marker offset frequencies is returned as a comma-separated list.

DisplayOverlayMode([Boolean overlay_mode])

Equivalent to the **Display→Browse plots one at a time** (b) or **Display→Overlay all loaded plots** (o) command, depending on the overlay_mode parameter.

The overlay_mode parameter is optional; if it is not provided by the script, the function simply returns true in **Overlay** mode or false in **Browse** mode. If a new overlay_mode value is specified, the function returns the previous display mode.

DisplayPlotVisibility(Number selection [, Boolean status])

Returns the visibility status of the plot at the specified position in the legend table beneath the graph. True indicates that the plot is visible, while false indicates that its visibility has been turned off by the user or by a previous call to this function.

If the optional status argument is provided, the selected plot's visibility state will be set to that value before returning the previous visibility state.

DisplayShowIntTraces([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Display→Show integrated PN traces**.

To return the current flag value, call **DisplayShowIntTraces** with no argument.

DisplayShowNumTab([Boolean status])

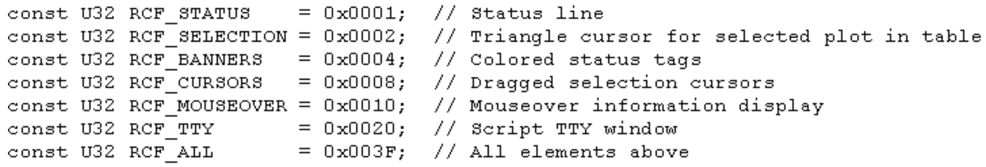
Sets and/or retrieves the **status** of the flag controlled by **Display→Numeric table (**Ctrl-n**)**. Refer to this command for more information.

To return the current flag value, call **DisplayShowNumTab()** with no argument.

DisplayRefresh([Number mask])

Forces an immediate display refresh that includes the elements specified by **mask**. If no parameter is passed, all display elements are refreshed.

Currently-defined display elements include the following:



```
const U32 RCF_STATUS    = 0x0001;  // Status line
const U32 RCF_SELECTION = 0x0002;  // Triangle cursor for selected plot in table
const U32 RCF_BANNERS   = 0x0004;  // Colored status tags
const U32 RCF_CURSORS   = 0x0008;  // Dragged selection cursors
const U32 RCF_MOUSEOVER = 0x0010;  // Mouseover information display
const U32 RCF_TTY       = 0x0020;  // Script TTY window
const U32 RCF_ALL       = 0x003F;  // All elements above
```

DisplaySpotTau(Number secs)

Sets the location of the spot cursor on Allan deviation and related log-log statistical plots to the desired tau interval in seconds, returning the previous cursor location.

DisplaySpotTime(Number secs)

Sets the location of the spot cursor on phase- and frequency-difference plots to the desired X-axis value in seconds, returning the previous cursor location.

DisplaySpotOffset(Number Hz)

Sets the location of the spot cursor on phase noise and AM noise plots to the desired Fourier offset in Hz, returning the previous cursor location.

DisplaySpotLowerOffset(Number Hz)

Sets the location of the lower integration limit cursor on phase noise plots to the desired Fourier offset in Hz, returning the previous cursor location.

DisplaySpotUpperOffset(Number Hz)

Sets the location of the upper integration limit cursor on phase noise plots to the desired Fourier offset in Hz, returning the previous cursor location.

DisplayYAxisHz([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Display→Frequency Y axis labels in Hz**.

To return the current flag value, call **DisplayYAxisHz** with no argument**.**

EventAcqDone([success][,user])

This is an optional user-supplied function. **EventAcqDone()** is called when an acquisition ends, either because the requested measurement duration expired, because an error occurred, or because it was stopped manually by the user. The optional **success** parameter is a Boolean value that indicates whether the hardware driver reported any errors during the measurement. The optional **user** parameter receives the optional user variable passed to **AcqStartAcquisition()**.

See **EventRunScript()** for further information.

EventAcqSetup([user])

This is an optional user-supplied function. **EventAcqSetup()** is called after a measurement running on a PhaseStation 53100A has connected to the target instrument, but before it enters the pretrigger state in preparation for data acquisition. The optional **user** parameter receives the optional user variable passed to **AcqStartAcquisition()**.

During the setup phase, a number of additional setup-related API functions are available. The user setup handler typically uses these functions to communicate with the 53100A to query operational parameters, validate correct operation, and identify signal frequencies and amplitudes at the input jacks. The setup handler may also need to communicate with other equipment in certain applications.

In order to proceed to the pretrigger phase, the setup handler must exit by calling **AcqSetupEnd()**. Most scripts other than diagnostic applications will not need to implement a setup handler.

EventAcqTriggered([user])

This is an optional user-supplied function. **EventAcqTriggered()** is called when a measurement initiated by **AcqStartAcquisition()** begins collecting data. Typically this happens a few seconds after **AcqStartAcquisition()** is called by the user-provided **EventRunScript()** function, unless the deferred acquisition options on the **Acquire** menu have been used to postpone data collection. The optional **user** parameter receives the optional user variable passed to **AcqStartAcquisition()**.

Any errors that occur between **EventRunScript()** and **EventAcqTriggered()** automatically terminate script execution.

See **EventRunScript()** for further information.

EventRunScript()

This is a mandatory user-supplied function that must appear in every TimeLab script. It is called when the user issues **Scripts→Run script** or a similar command, and serves as the entry point for script execution.

Typically, **EventRunScript()** does little more than set up the appropriate acquisition parameters and call **AcqStartAcquisition()**, returning immediately afterwards. Subsequently, the script's **EventAcqTriggered()** handler is called by TimeLab when the acquisition begins collecting data after the optional **EventAcqSetup()** phase. **EventAcqTriggered(),** in turn, may use **TimeSetTimer()** to arrange for periodic callback service to **EventTimer()**. Scripts that need to perform extensive measurement supervision and control operations normally do so within their **EventTimer()** handlers, while final result evaluation and report generation is done within **EventAcqDone()**.

For a detailed look at the operation and overall life cycle of test scripts, refer to the **53100A_performance_test.js** script provided with TimeLab, as well as the other API function descriptions in this section.

EventTimer()

This is an optional user-supplied function. **EventTimer()** is called periodically at a rate determined by the parameter passed to **TimeSetTimer().**

Because TimeLab's scripting system relies on cooperative multitasking, most non-trivial scripts should carry out the majority of their processing work in their **EventTimer()** handler while their measurement(s) are in progress. For detailed examples of timer usage, refer to **53100A_performance_test.js** and **5330A_ADC_test.js**. Also see **EventRunScript()** for further information.

FileCloseAllChannels()

Closes all plot(s) acquired in the most recent measurement executed by the script, stopping any acquisition(s) that may still be in progress. For conventional single-channel measurements, **FileCloseAllChannels()** is equivalent to calling **FileClosePlot()** with no argument.

As an example, **53100A_performance_test.js** uses **FileCloseAllChannels()** in its **EventAcqDone()** handler to discard the two plots generated during the preliminary warmup acquisition that the script performs before running the actual test. (Because the warmup acquisition has already terminated at this point, **EventAcqDone()** is not reissued.) Simply calling **FileClosePlot()** here would have closed only the first of the two plots.

FileCloseAllPlots()

Equivalent to calling **FileClosePlot()** on all loaded plots. For example, **5330A_ADC_test.js** uses **FileCloseAllPlots()** to ensure that enough slots are available in the legend table to hold the four plots that it acquires for various ADC combinations.

FileClosePlot([Number selection])

If the optional **selection** parameter is specified, **FileClosePlot()** closes the plot at the specified position in the legend table, from 0 to 8. Since plots being acquired under script control may be moved up and down in the legend table by the user like any other loaded plots, **AcqChannel()** is needed to obtain the value passed to this function if the intent is to close any of the plot(s) that are associated with a measurement launched by the script.

Alternatively, **selection** may be omitted. In this case, the function closes the plot associated with the first channel of the most recent measurement executed by the script.

In both cases, calling **FileClosePlot()** on a plot associated with a given measurement will stop any acquisition(s) in progress that were initiated by that measurement. The script's **EventAcqDone()** handler, if any, will be called for any acquisitions that were terminated by **FileClosePlot()**.

FileExecute[Wait](String path_to_file [, String params])

FileExecute() attempts to execute or otherwise open the specified file using the "open" verb with the Win32 ShellExecute() API function. If ShellExecute() returns successfully, **FileExecute()** returns true; otherwise, if an error occurred, it returns false.

An alternative version of this function, **FileExecuteWait()**, pauses execution of TimeLab until the spawned process exits. **FileExecuteWait()** uses the Win32 CreateProcess and WaitForSingleObject() APIs rather than ShellExecute(), but otherwise behaves similarly.

The **path_to_file** parameter should be a fully-qualified pathname. Optionally, a string consisting of one or more parameters may be passed to ShellExecute() via the **params** argument.

As an example, **FileExecute()** is used by **53100A_performance_test.js** to view the generated HTML report page on the system's Web browser.

FileExitApplication([code])

When called, this function causes the TimeLab process to exit to the command shell or other launching process with the specified exit **code**. If no parameter is provided, the returned exit code is 0.

Process termination occurs when the JavaScript function that called **FileExitApplication()** returns. It is not necessary to clean up resources by calling TimeSetTimer(0) or ScriptEnd(); the calling function may return immediately after calling **FileExitApplication()** to request termination.

See the **EventTimer()** handler in **53100A_command_line_example.js** for a detailed usage example.

FileExists(String path, [String subdir,] String filename)

Returns a Boolean value of true if the specified file can be opened for reading, false otherwise.

By default, if **path** is empty, the user's Documents folder is treated as the path. This path must already exist. The **subdir** parameter is optional; it represents a subdirectory relative to **path** that contains the specified file. Unlike **path**, the **subdir** folder will be created automatically if it does not already exist.

This function can be used to poll for signals from external or host processes that create empty files with specific names. See the **EventTimer()** handler in **53100A_command_line_example.js** for a detailed usage example.

FileExportASCIIData(String descriptor, String path, [String subdir,] String filename)

This function writes phase-difference samples from the phase data record, sigma(tau) data from the ADEV, MDEV, HDEV, TDEV, or MTIE bins, or AM/PM noise or spur data to the specified ASCII text file, depending on which measurement has most recently been selected for display with **MeasurementSelectView()**. In most cases, the exported data comes from the plot that is currently associated with (bound to) the measurement script.

The **descriptor** parameter is a case-insensitive text string, in which the presence or absence of certain characters influences the composition of the exported data as outlined below.

Phase/frequency difference views: Each phase-difference or absolute frequency sample may be preceded by an optional timestamp. This will be the case if the **descriptor** string contains the characters "M", indicating that Modified Julian Date timestamps should be written to the file prior to each exported sample, or "T", indicating that each data point should be prefaced by its sample time in seconds relative to the beginning of the phase record. If the string contains a comma, the timestamp and sample values will be separated by a comma followed by a single space; otherwise, they will be separated by a single space.

Allan deviation and other stability measurements: The output file will consist of a series of individual bin records, each of which appears on its own line with the tau and sigma(tau) values. By default these values are separated by a single space, but if the **descriptor** parameter contains a comma, a comma followed by a single space will be used.

If a three-cornered hat display is currently visible at the time **FileExportASCIIData** is called, there will be a total of four columns in each exported line, consisting of the tau value followed by the separated deviation figures for each unique source in the measurement. If the **descriptor** parameter contains a “#” or “;” character, that character will appear at the beginning of a comment in the first line of the file which will contain the word “Tau” followed by the names of the sources associated with each column. Otherwise, no comment line will be written. Note that any degenerate bin values in three-cornered hat displays will be replaced with the placeholder “NaN” (Not a Number).

Phase noise and AM noise measurements: By default, each line in the exported file will consist of the Fourier offset frequency, a separator substring, and the L(f) or M(f) value in dBc/Hz at that offset frequency. As with other data types, the separator will be either a comma followed by a space if **descriptor** contains a comma, or a single space by itself if not.

If the **descriptor** string associated with an export operation in the phase noise view contains an “i” character, at least one integrated noise type has been selected for display in the legend table, and **Display→Show integrated PN traces** is enabled, then one or more files containing integrated phase noise data will be written instead of a single file containing offset and L(f) values. As before, a two-column format with optional comma separators is used, where the offset frequency appears in the first column and the integrated noise value appears in the second column. Each filename will consist of the portion of the specified filename prior to any suffix, followed by “_xxx” where xxx is the name of the selected integrated phase noise type (SSB_CNR_dB, RMS_rads, RMS_degs, resid_FM_Hz, or RMS_jitter_s) if more than one integrated noise type is visible, followed by the suffix portion of the specified filename.

Finally, if the **descriptor** contains the letter “s”, the spur chart for an AM or phase noise measurement will be exported instead of the noise data, again with each entry consisting of the offset frequency followed by the spur amplitude at that offset frequency.

By default, if **path** is empty, the user’s Documents folder is treated as the destination path. This path must already exist. The **subdir** parameter is optional; it represents a subdirectory relative to **path** in which the destination file will be saved. Unlike **path**, the **subdir** folder will be created automatically if it does not already exist.

The **FileExportASCIIData()** function returns a string containing the exported file's fully-qualified pathname, or that of the last file written in the case where multiple integrated noise files are exported. Any errors that occur will not be reported to the script. They will result in a JavaScript exception which will terminate the script with an appropriate error message.

To export data from multiple plots in a multichannel measurement, you must use **ScriptBindToPlot()** to associate the script with the channel whose data is to be exported. See **53100A_data_export_example.js** for an example. The only exception to this is the three-cornered hat case described above, where all visible three-cornered hat traces will be exported regardless of the file associated with the script.

FileImageURI()

Returns a .GIF image of the current TimeLab display in the form of an HTML data URI.

FileLoad(String path, [String subdir,] String filename)

Reads the specified .TIM file into memory and allocates a plot for it, just as if the plot had been created by calling **AcqStartAcquisition**(). On success, the fully-qualified pathname of the loaded file is returned.

Typically, **DisplayRefresh()** should be called after **FileLoad()** in order to display the newly-loaded file and make the measurement data available to other JavaScript API functions. An exception is thrown if the file could not be loaded or if the script is already bound to an existing plot. **ScriptUnbind()** must be called prior to **FileLoad()** in this case.

By default, if **path** is empty, the user's Documents folder is treated as the path. This path must already exist. The **subdir** parameter is optional; it represents a subdirectory relative to **path** that contains the specified file. Unlike **path**, the **subdir** folder will be created automatically if it does not already exist.

FileNumLoadedPlots()

Returns the number of currently-loaded plots, from 0 to 9.

FileNumUnsavedPlots()

Returns the number of currently-loaded plots, from 0 to 9, that have not been saved since being acquired or modified.

FileSave(String path, [String subdir,] String filename)

Based on the (case-insensitive) suffix of **filename**, **FileSave()** saves either a screen image in .TGA, .GIF, .BMP, .PCX, or .PNG format, or a .TIM file representing the measurement currently associated with the script.

By default, if **path** is empty, the user's Documents folder is treated as the destination path. This path must already exist. The **subdir** parameter is optional; it represents a subdirectory relative to **path** in which the destination file will be saved. Unlike **path**, the **subdir** folder will be created automatically if it does not already exist.

The **FileSave()** function returns a string containing the saved file's fully-qualified pathname. Any errors that occur will not be reported to the script. They will result in a JavaScript exception which will terminate the script with an appropriate error message.

To save .TIM data from multiple plots in a multichannel measurement, you must use **ScriptBindToPlot()** to associate the script with channels other than the first one. See **53100A_performance_test.js** for an example.

FileSaveText(String path, [String subdir,] String filename, String contents)

The **FileSaveText()** function saves the ASCII **contents** string to **filename**. As with **FileSave()** above, the user's Documents folder is treated as the destination folder if **path** is empty. Any explicitly-specified **path** must already exist. The **subdir** parameter is optional; it represents a subdirectory relative to **path** in which the destination file will be saved. Unlike **path**, the **subdir** folder will be created automatically if it does not already exist.

This function returns a string containing the saved file's fully-qualified pathname. Any errors that occur will not be reported to the script. They will result in a JavaScript exception which will terminate the script with an appropriate error message.

As an example, 53100A_performance_test.js uses **FileSaveText()** to save its HTML report.

LegendSelect([String field_name] [,] [Boolean status])

Controls and/or reports the visibility of a given parameter (**field_name**) in the legend table beneath the graph.

This function's arguments may consist of a single Boolean value, in which case it will select or deselect *all* legend fields in a manner similar to the **Legend→All** or **Legend→None** menu commands; a single String value, which will simply return the visibility state of the specified **field_name** without changing it; or both String and Boolean parameters, which will show (**status**==true) or hide (**status**==false) the specified **field_name** in the table and return its previous state.

MaskResultMargin([Number mask_num])

Returns a Number value representing the margin by which the measurement associated with the script is passing (if positive) or failing (if negative) the mask test selected by **MaskSelect()**. The returned value is based on the most recently rendered frame, so it is dependent on both the current measurement view *and* the specified mask. Refer to the

Masks menu description on page 118 for more information about TimeLab's mask-test functionality.

Up to five masks may be selected at once. The optional **mask_num** parameter ranges from 0 to 4, defaulting to 0 if not specified.

Important: Keep in mind that JavaScript execution does not disable or inhibit the rest of the TimeLab user interface. While test masks are easy to create and use in TimeLab, script-based mask evaluation is complicated by the need to establish the correct measurement view, allow enough time for the display to update, and wait for valid results to become available for the X-axis range covered by the mask, all while allowing for the user's ability to switch measurement views, move plots up and down in the legend table, or alter other settings manually at any time.

Typically the best way to coordinate the necessary script actions while allowing for manual user intervention is to use an **EventTimer()** handler to check the mask results and the current measurement view selection at the same time, then make any changes (such as selecting the next measurement view for evaluation) at the very end of the timer handler. Script authors are *strongly* encouraged to use the **EventTimer()** handler in **53100A_performance_test.js** as a model for their own mask-test procedures.

MaskResultValid([Number mask_num])

Returns true if valid data is available from **MaskResultMargin()**, or false if the mask test was not ready for evaluation for any reason. For example, ADEV mask results will not be available if the measurement has not yet run long enough to return valid data at the tau corresponding to the left endpoint of the mask limit line, or if the user has turned visibility off for the plot associated with the script. Always check **MaskResultValid()** prior to calling **MaskResultMargin()**.

The optional **mask_num** parameter ranges from 0 to 2, defaulting to 0 if not specified.

See **MaskResultMargin()** for other notes on scripted mask tests.

MaskSelect([String mask_name [, Number mask_num])

This function selects a mask for use with **MaskResultMargin()** and **MaskResultValid()**, returning the previously-selected mask if any.

The **mask_num** parameter ranges from 0 to 2, defaulting to 0 if not specified.

Specifying an empty string ("") for **mask_name** will deselect the mask in the specified slot, or deselect all masks if **mask_num** is not specified.

If **mask_name** is omitted, the function returns the currently-selected mask name for slot 0. The function will return an empty string if no mask was selected.

Mask names should be specified *exactly* as they appear in the **masks.txt** file accessed via **Masks→Edit mask definitions** (which also determines the names under which they are added to the **Masks** menu). See **MaskResultMargin()** for other notes on scripted mask tests.

MeasurementSelectView(String hotkey)

MeasurementDeferSelectView(String hotkey)

These functions accept a **hotkey** parameter that consists of the (case-sensitive) shortcut key for the desired measurement view as it appears in the **Measurement** menu. For example, **MeasurementSelectView("P")** will emulate the 'P' keyboard shortcut (**Measurement→Phase noise (**P**)**).

If your script is performing mask tests with multiple measurement types, it's a good idea to call **MeasurementDeferSelectView()** as the last action taken before returning from **EventTimer().** Upon the next invocation of **EventTimer()**, your script should call **MeasurementView()** and evaluate the masks or other test parameters for that view, regardless of any changes to the measurement view that may have taken place between timer ticks. Script authors are encouraged to use the **EventTimer()** handler in **53100A_performance_test.js** as a model for test procedures that use multiple measurement types. Otherwise, it may be necessary to call **DisplayRefresh()** after changing the measurement view manually, in order to allow the script to access the most recent available data.

The difference between **MeasurementSelectView()** and **MeasurementDeferSelectView()** is that the latter will not take effect until after the event handler returns. This distinction is important when multiple instances of the same script may be executed at once. Essentially, deferring the view selection until after all script instances with the same **EventTimer()** interval have been serviced leaves a single script instance "in charge" of the view, reducing opportunities for confusion.

MeasurementView()

Returns the current measurement view as a String containing the hotkey for the view as it appears in the **Measurement** menu. For example, when viewing a phase noise plot, **MeasurementView()** will return "P", the shortcut key for **Measurement→Phase noise**.

See **MeasurementSelectView()** for additional notes on measurement view selection.

Print(…)

Displays one or more user-specified values in the script console window associated with the currently-bound plot. Multiple arguments may be separated by commas; a line feed will be inserted after each argument. (To avoid this, you can use string concatenation rather than multiple **Print()** arguments.)

Numerous usage examples for the Print() function appear in **53100_performance_test.js** and **5330A_ADC_test.js**.

ScriptBindToPlot(Number selection)

Associates the script instance with the plot at the specified position in the legend table, from 0 to 8.

Most script authors will not need to use ScriptBindToPlot(). It is required only in scripts that support multichannel plots, such as **53100A_performance_test.js** and **5330A_ADC_test.js**. A script can be associated ("bound") to only one plot at a time, and most operations performed by a script implicitly involve the currently bound plot. When a measurement initiated by a script specifies multiple stability channels, as in the TimePod and PhaseStation validation scripts mentioned, it's necessary to use ScriptBindToPlot() to access the plots that represent all channels of the measurement.

By default, scripts are bound to channel 0 of a multichannel acquisition. It's a good idea to restore the default channel-0 binding with ScriptBindToPlot(AcqChannel(0)) before returning from any function or event handler that uses ScriptBindToPlot().

Since plots in multichannel measurements may be moved up and down in the legend table by the user, ScriptBindToPlot() is normally used together with the AcqChannel() function. See the two performance test scripts mentioned above for further comments and usage examples.

ScriptBoundToPlot()

Returns the position in the legend table beneath the graph, from 0 to 8, of the plot to which the script is currently bound.

Most script authors will not need to use ScriptBoundToPlot(). It is required only in scripts that support multichannel plots, such as **53100A_performance_test.js** and **5330A_ADC_test.js**.

ScriptEnd()

Marks the script for cleanup by the TimeLab runtime JavaScript engine. Scripts which are associated with loaded plots must be terminated with a call to ScriptEnd() when finished. Otherwise, they will continue to run (and consume resources) until one of the following conditions is true:

- A script exception or other fatal error occurs

- The **Scripts→Stop all running scripts (**F12**)** command is issued by the user

- The user exits from TimeLab

ScriptFilename(Boolean full_path)

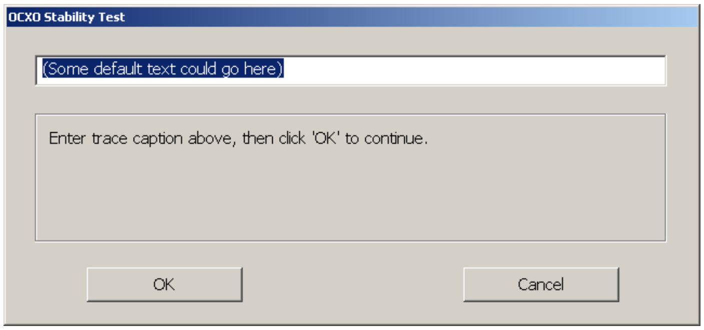
If **full_path** is true, ScriptFilename() returns a String containing the fully-qualified path to the script's JavaScript (.js) source file. Otherwise, it returns only the filename itself.

ScriptGetString(String heading, String message, Object user_text, String b1_text [, String b2_text [, String b3_txt]])

Presents a modal dialog box to the user that accepts freeform text entry. An example appears below.

```
var trace_caption = { value:"(Some default text could go here)" };

if (ScriptGetString(   "OCXO Stability Test",
                       "Enter trace caption above, then click 'OK' to continue.",
                        trace_caption,
                       "OK",
                       "Cancel") != 0)
        {
        ScriptEnd();
        return;
        }
```

When executed, the result is a dialog box of the form


OCXO Stability Test

(Some default text could go here)

Enter trace caption above, then click 'OK' to continue.

Figure 67: ScriptGetString() dialog

Text for one, two, or three buttons can be specified as the **b1_text**, **b2_text**, and/or **b3_text** argument(s) to **ScriptGetString()**. The returned value will equal the numeric index of the button selected by the user, beginning with 0 for the **b1_text** button.

All text fields accept up to 1024 bytes. Note that the **user_text** argument is an Object containing a String property called **value**, not a String in itself. The **value** property is used both as the default text used to initialize the edit control and for reception of the returned text. (This is necessary because Strings are always passed by copy in JavaScript, while Objects are passed by reference and may be modified by the called function.)

ScriptHostFilename([bool full_path])

Returns a String containing the TimeLab executable filename and version information. Either the full pathname or only the filename portion may be requested, depending in the value of the optional **full_path** parameter.

ScriptLastStatusMessage()

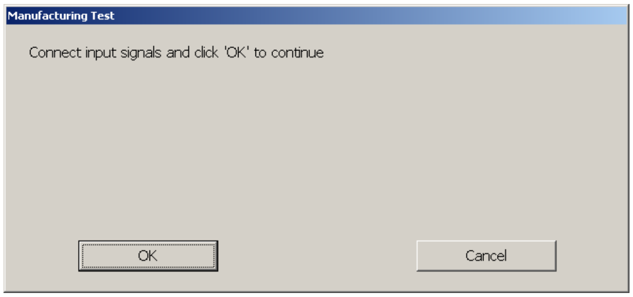
Returns a String containing the most recent status message displayed by TimeLab. This function may be useful for logging and report generation.

ScriptMessageBox(String heading, String message, String b1_text [, String b2_text [, String b3_txt]])

Presents a modal dialog box to the user that accepts button presses. An example appears below.

```
if (ScriptMessageBox(  "Manufacturing Test",
                       "Connect input signals and click 'OK' to continue.",
                       "OK",
                       "Cancel") != 0)
       {
       ScriptEnd();
       return;
       }
```

When executed, the result is a dialog box of the form



Manufacturing Test

Connect input signals and click 'OK' to continue

Figure 68: ScriptMessageBox() dialog

Text for one, two, or three buttons can be specified as the **b1_text**, **b2_text**, and/or **b3_text** argument(s) to **ScriptMessageBox()**. The returned value will equal the numeric index of the button selected by the user, beginning with 0 for the **b1_text** button.

ScriptMinsToHMS(Number minutes)

Returns a String representing the minutes argument as hours, minutes, and seconds.

Example: The following statement

```
Print(ScriptMinsToHMS(10.5));
```

will display:

```
10m 30s
```

ScriptNumParam(Number n)

Returns a Number corresponding to the floating-point numeric value of the *n*th (zero-based) command line parameter following the JavaScript (.js) program name on the TimeLab command line. An exception will be thrown if the requested parameter is not present or is not representable as a numeric value.

See **53100A_command_line_example.js** for a detailed usage example.

ScriptNumVal(String format_string, Number value)

Returns a String consisting of the specified format_string with an embedded numeric value.

In operation, ScriptNumVal() passes its format_string and floating-point value arguments to the standard C sprintf() function, returning the resulting formatted string.

Example: The following statement

```
Print(ScriptNumVal("Pi = %.2lf", 3.1415926535));
```

will display:

```
Pi = 3.14
```

Additional usage examples appear in both **53100A_performance_test.js** and **5330A_ADC_test.js**. Refer to the sprintf() documentation in the C runtime library reference for a detailed discussion of the format and precision specifiers supported for floating-point values.

ScriptParamCount()

Returns a Number value representing the number of command-line parameters following the JavaScript (.js) program name on the TimeLab command line.

The result will be zero if no script was specified on the TimeLab command line, or if no subsequent parameter strings were passed on the command line.

See **53100A_command_line_example.js** for a detailed usage example.

ScriptPlaySound(String filename)

Plays the specified sound file. Sound files for use with this function are typically stored in the .wav format.

If **filename** contains a backslash character, TimeLab assumes it consists of a fully-qualified path name and plays the sound from the filename exactly as specified. Otherwise, filenames without backslash characters are played from the TimeLab installation directory.

ScriptSetWindowPos([String text])

Returns the current position and size of the TimeLab window on the Windows desktop, optionally setting a new position and size if the **text** argument is provided.

As an example, the command

```
   var prev_layout = ScriptSetWindowPos("0x0, 1, (-1,-1), (-1,-1), (10,10,1410,970)");
```

will move the TimeLab window to occupy the region on the desktop whose upper-left and lower-right corners are (10,10) and (1410,970) respectively. A descriptor string for the previous window position will be returned.

Fields in the descriptor string other than the last four are reserved for system use and should not normally be altered.

ScriptSetWindowTitle([String text])

Returns the current TimeLab window title string, optionally setting a new window title if the **text** argument is provided.

ScriptSetTTYHeader(Number linenum [, String text])

Sets and/or retrieves the text displayed at the top of the script TTY window that is associated with the plot(s) bound to the script instance.

The **linenum** argument specifies which of the first two TTY lines will be retrieved or updated; it must equal 0 or 1. The **text** argument is optional. If present, it will replace the existing text for the specified TTY line. Regardless, the function will return the contents of the specified line at the time it was called.

Note that the script TTY is associated with the most recent plot that has been acquired by the script or otherwise bound to it. (All channels in a multichannel acquisition share the same TTY.) The **Print** function will write its output to the TimeLab status line if no TTY is associated with the script, but **ScriptTTYHeader()** will fail with an exception. In most cases, you should not attempt to write to the TTY before **AcqStartAcquisition()** has been called.

ScriptShowTTY([Boolean show])

Shows or hides the script TTY window that is associated with the plot(s) bound to the script instance, returning the previous visibility state. If the **show** argument is omitted, the function will return the current visibility state without changing it.

The script TTY window becomes visible by default upon successful execution of **AcqStartAcquisition()**. However, because the TTY can obscure a large portion of the TimeLab graph display area, simple scripts that report little or no status information may wish to hide the TTY window. The user will still have the option to select **Scripts→Toggle script console for selected plot (**F11**)** to show the TTY manually.

Unlike the other TTY functions, **ScriptShowTTY()** may be called at any time. The visibility state that it establishes will be inherited by any subsequent plots that the script acquires.

ScriptStrParam(Number n)

Returns a String corresponding to the contents of the *n*th (zero-based) command line parameter following the JavaScript (.js) program name on the TimeLab command line. An exception will be thrown if the requested parameter is not present.

See **53100A_command_line_example.js** for a detailed usage example.

ScriptUnbind()

Removes the association between the script instance and any plot to which it is bound.

SetupDetectedAmpldBfs

SetupDetectedValid

SetupDetectedFreqHz

SetupDetectSignals

SetupMeasuredAmpldBm

SetupMeasuredFreqHz

SetupMeasureSignals

SetupNumQuery

SetupSetDefaultAttenuation

SetupStrQuery

These setup API functions are supported only by the PhaseStation 53100A, and only when called from within an **EventAcqSetup** handler. They are currently reserved for internal use.

TimeHoursSinceTrigger()

TimeMinsSinceTrigger()

TimeSecsSinceTrigger()

These three functions return the elapsed time since the measurement currently associated with the script was triggered, if the measurement is still in progress. Otherwise, if the measurement is no longer in progress, the overall duration of the acquisition is returned.

The values returned are floating-point Numbers that equivalently represent the elapsed time in terms of hours, minutes or seconds.

TimeSetTimer(Number msec [, Boolean persist])

This function arranges for TimeLab to call the script's **EventTimer()** handler at regular intervals of **msec** milliseconds. Timer callbacks are made **msec** milliseconds after the previous call to the timer handler returned – meaning that if a timer handler runs for more than one service interval, missed intervals will not be made up.

Because TimeLab's scripting system relies on cooperative multitasking, most non-trivial scripts should carry out the majority of their processing work in their **EventTimer()** handler while their measurement(s) are in progress. For detailed examples of timer usage, refer to **53100A_performance_test.js** and **5330A_ADC_test.js** as well as **53100A_command_line_example.js** and **53100A_3_cornered_hat.js**.

Prior to termination, a script that uses timer service should call **TimeSetTimer(0)** to discontinue further callbacks.

By default, scripts are terminated automatically if a timer event occurs with no acquisition in progress. If the **persist** parameter is present and equal to **true**, automatic termination will not occur. This may result in script resource leakage if the user closes the script's plot(s) manually without terminating the script itself.

TraceAvgWindow([Number seconds])

Sets and/or retrieves the duration in **seconds**, from 0.01 to 10000.0, of the averaging window used for phase- and frequency-difference plots. An argument of 0 **seconds** disables averaging entirely.

The function returns the previous window duration. It may be called with no argument to return the current value without altering it.

See the **Trace→Averaging window for phase/frequency traces (**g**)** command for important information on trace averaging.

TraceFreqZero([Boolean status])

Sets and/or retrieves the **status** of the flag used by **Trace→Phase/frequency traces begin at zero (**z**)** when frequency-difference measurements are selected for viewing. Refer to this command for more information.

To return the current flag value for the frequency-difference view, call **TraceFreqZero()** with no argument.

TraceMarkSpurs([Boolean status])

Sets and/or retrieves the **status** of the spur-visibility flag controlled by **Trace→Mark spurs in noise traces (**Ctrl-m**)**. Refer to this command for more information.

To return the current spur-visibility flag value, call **TraceMarkSpurs()** with no argument.

TraceOverlayAMPM([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Trace→Show AM noise in PN view (**F8**)**. Refer to this command for more information.

To return the current flag value, call **TraceOverlayAMPM()** with no argument.

TracePhaseFreqAutoY([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Trace→Phase/frequency Y axis unlocked in zoom mode (**y**)**. Refer to this command for more information.

To return the current flag value, call **TracePhaseFreqAutoY()** with no argument.

TracePhaseFreqResid([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Trace→Show linear phase/frequency residual (**r**)**. Refer to this command for more information.

To return the current flag value, call **TracePhaseFreqResid()** with no argument.

TracePhaseFreqTrend([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Trace→Show linear phase/frequency trend (**Ctrl-t**)**. Refer to this command for more information.

To return the current flag value, call **TracePhaseFreqTrend()** with no argument.

TracePhaseFreqZero([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Trace→Phase/frequency traces begin at zero (**z**)**. Refer to this command for more information.

To return the current flag value for the phase- or frequency-difference view, call **TracePhaseFreqZero()** with no argument.

TracePhaseZero([Boolean status])

Sets and/or retrieves the **status** of the flag used by **Trace→Phase/frequency traces begin at zero (**z**)** when phase-difference measurements are selected for viewing. Refer to this command for more information.

To return the current flag value for the phase-difference view, call **TracePhaseZero()** with no argument.

TraceShowCorrGain([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Trace→Show correlation gain for selected noise trace (**Ctrl-g**)**. Refer to this command for more information.

To return the current flag value, call **TraceShowCorrGain()** with no argument.

TraceShowCrossStatistics([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Trace→Show cross ADEV traces when available (**Ctrl-j**)**. Refer to this command for more information.

To return the current flag value, call **TraceShowCrossStatistics()** with no argument.

TraceShowHatStatistics([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Trace→Show separated variances (N-cornered hat) when available (**Ctrl-h**)**. Refer to this command for more information.

To return the current flag value, call **TraceShowHatStatistics()** with no argument.

TraceShowImag([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Trace→Show imaginary part of cross spectrum (**Ctrl-F3**)**. Refer to this command for more information.

To return the current flag value, call **TraceShowImag()** with no argument.

TraceShowNoiseFloor([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Trace→Show estimated instrument noise (**F2**)**. Refer to this command for more information.

To return the current flag value, call TraceShowNoiseFloor() with no argument.

TraceShowOriginal([Boolean status])

Sets and/or retrieves the status of the flag controlled by **Trace→Show original traces in computed xDEV displays (**F6**)**. Refer to this command for more information.

To return the current flag value, call TraceShowOriginal() with no argument.

TraceShowRaw([Boolean status])

Sets and/or retrieves the status of the flag controlled by **Trace→Show raw PN channel trace(s) (**Ctrl-r**)**. Refer to this command for more information.

To return the current flag value, call TraceShowRaw() with no argument.

TraceShowSlopes([Boolean status])

Sets and/or retrieves the status of the flag controlled by **Trace→Show FFT segment filter slope(s) (**Ctrl-i**)**. Refer to this command for more information.

To return the current flag value, call TraceShowSlopes() with no argument.

TraceSmoothNoise([Boolean status])

Sets and/or retrieves the status of the flag controlled by **Trace→Smooth noise traces (**Ctrl-w**)**. Refer to this command for more information.

To return the current flag value, call TraceSmoothNoise() with no argument.

TraceSuppressSpurs([Boolean status])

Sets and/or retrieves the status of the spur-suppression flag controlled by **Trace→Suppress spurs in noise traces (**Ctrl-s**)**. Refer to this command for more information.

To return the current spur-suppression flag value, call TraceSuppressSpurs() with no argument.

TraceThickness([Number status])

Sets and/or retrieves the status of the flag controlled by **Trace→Toggle trace thickness for current measurement (**T**)**. Refer to this command for more information.

If a value other than 1 or 0 is passed, all traces will be rendered with heavy lines. (Boolean true and false values may also be used.) To return the current flag value, call TraceThickness() with no argument.

TraceTickMarks([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Trace→Tick marks (**k**)**. Refer to this command for more information.

To return the current flag value, call **TraceTickMarks()** with no argument.

TraceXDEVBars([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Trace→Show xDEV error bars (**Ctrl-e**)**. Refer to this command for more information.

To return the current flag value, call **TraceXDEVBars()** with no argument.

TraceXDEVClipBW([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Trace→Clip xDEV traces by noise bandwidth (**Ctrl-b**)**. Refer to this command for more information.

To return the current flag value, call **TraceXDEVClipBW()** with no argument.

TraceXDEVClipConfidence([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Trace→Clip xDEV traces by confidence (**Ctrl-v**)**. Refer to this command for more information.

To return the current flag value, call **TraceXDEVClipConfidence()** with no argument.

TraceXDEVSpline([Boolean status])

Sets and/or retrieves the **status** of the flag controlled by **Trace→Draw xDEV traces with spline interpolation (**i**)**. Refer to this command for more information.

To return the current flag value, call **TraceXDEVSpline()** with no argument.

Accessing the PhaseStation and TimePod Remotely

Introduction to STREAM

TimeLab includes a Windows console application, **STREAM.EXE**, which can be used to provide remote access to measurement data acquired by a PhaseStation 53100A or TimePod 5330A. Continuous streams of phase-difference or frequency measurements acquired by **STREAM.EXE** can be written to a shared file or streamed continuously via TCP/IP over a local- or wide-area network.

Important note: As of October 2019, **STREAM** is deprecated for use with the PhaseStation 53100A. A new application, **TSERVE**, has been added to the TimeLab distribution. Like **STREAM**, **TSERVE** supports continuous streaming of phase-difference and frequency measurements. **TSERVE** also adds support for remote control commands that allow phase noise and AM noise measurements to be configured, started, stopped, and monitored by networked clients. The **TSERVE** command language provides a high degree of compatibility with the Microsemi/Symmetricom/TSC 51XXA series of phase noise test sets. Refer to Introduction to TSERVE to learn how to configure, launch, and access the **TSERVE** remote measurement server.

Launching STREAM

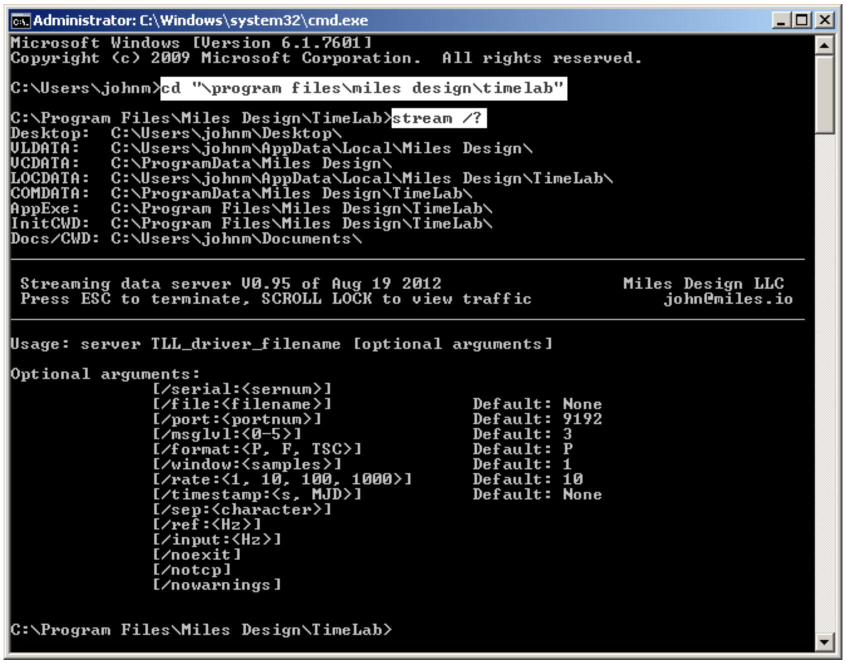
STREAM.EXE uses relatively little CPU time or RAM compared to the full-fledged TimeLab application. As a console application, it displays no graphics at all. Consequently it can run on almost any Windows PC, including energy-efficient servers or netbooks.

You can launch **STREAM.EXE** from a Windows desktop shortcut, but it's preferable to run it from a command prompt ("DOS box") instead. Because the program's user interface is based on command-line parameters rather than GUI controls, you'll find it easier to work at the command prompt during the familiarization process. After you've determined what options and parameters are needed in your application, you can create a batch file or shortcut to launch **STREAM.EXE** with the same parameters.

STREAM.EXE is installed in the same directory as the rest of the TimeLab package. Unless you specified a different location during installation, you can find it by opening a DOS session and changing the current directory to **c:\Program Files\Miles Design\TimeLab**. To do this, select *Start → Run…* in Windows and enter *cmd* as the name of the program to run. Once the DOS box appears, drag its lower edge to expand the window to a more comfortable size. Then, enter the commands

```
cd "\program files\miles design\timelab"

stream /?
```

As shown below, the /? option causes **STREAM.EXE** to display a brief list of the command-line options available in the current release. (Black on white text indicates typed commands.)



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\johnm>cd "\program files\miles design\timelab"

C:\Program Files\Miles Design\TimeLab>stream /?
Desktop:  C:\Users\johnm\Desktop\
VLDATA:   C:\Users\johnm\AppData\Local\Miles Design\
VCDATA:   C:\ProgramData\Miles Design\
LOCDATA:  C:\Users\johnm\AppData\Local\Miles Design\TimeLab\
COMDATA:  C:\ProgramData\Miles Design\TimeLab\
AppExe:   C:\Program Files\Miles Design\TimeLab\
InitCWD:  C:\Program Files\Miles Design\TimeLab\
Docs/CWD: C:\Users\johnm\Documents\

───────────────────────────────────────────────────────────────────────────────

 Streaming data server V0.95 of Aug 19 2012                  Miles Design LLC
 Press ESC to terminate, SCROLL LOCK to view traffic             john@miles.io

───────────────────────────────────────────────────────────────────────────────

Usage: server TLL_driver_filename [optional arguments]

Optional arguments:
              [/serial:<sernum>]
              [/file:<filename>]              Default: None
              [/port:<portnum>]               Default: 9192
              [/msglvl:<0-5>]                 Default: 3
              [/format:<P, F, TSC>]           Default: P
              [/window:<samples>]             Default: 1
              [/rate:<1, 10, 100, 1000>]      Default: 10
              [/timestamp:<s, MJD>]           Default: None
              [/sep:<character>]
              [/ref:<Hz>]
              [/input:<Hz>]
              [/noexit]
              [/notcp]
              [/nowarnings]


C:\Program Files\Miles Design\TimeLab>
```

Figure 69: Running STREAM.EXE from the MS-DOS command line prompt

Using STREAM

Because **STREAM.EXE** performs only phase/frequency stability measurements, its command-line options are relatively limited. Only one parameter is mandatory: you must specify the hardware driver's filename as the first parameter on the command line. Currently, only the TimePod and PhaseStation drivers are supported, so all **STREAM.EXE** command lines must begin as follows:

stream timepod.tll

stream phasestation.tll

Additional options, described below, may be specified as needed.

/serial:<sernum>

When more than one TimePod or PhaseStation is connected to the server PC, you may need to use the /serial option to associate a given **STREAM.EXE** instance with a specific instrument. Each instance of **STREAM.EXE** supports only one acquisition at a time, but you can launch as many instances as desired.

Example: *stream timepod.tll /serial:55908.304876*

The command above will launch an instance of **STREAM.EXE** for use with the TimePod whose serial number is 55908.304876. All other parameters will assume their default values, configuring the server to provide phase-difference data on TCP/IP port 9192 at a rate of 10 readings per second.

/file:<filename>

Specifies the name of a file to which the phase-difference or frequency readings will be written. If the specified file already exists, it will be overwritten. Typically the file is shared with other applications that can access it in read-only mode while it's being written by **STREAM.EXE**.

The data written to the file may be 'followed' from within TimeLab by selecting **Acquire→Acquire from live ASCII file***.* You can also monitor the data written to the file with a command such as *tail –f <filename>* from Cygwin. Finally, pressing the Scroll Lock key will cause **STREAM.EXE** to print a copy of all lines written to the file to the stdout device (i.e., the DOS console itself).

Example: *stream timepod.tll /notcp /file:%HOMEPATH%\Documents\mydata.txt*

The command above writes a continuous stream of phase-difference readings to the file mydata.txt in the current user's Documents folder. You can then use TimeLab or another application to 'follow' the file. Even if the acquisition fails or otherwise encounters problems, you can use the appropriate **File→Import** option to bring the data into TimeLab at any time.

Additionally, the /notcp option Is used in this example to keep **STREAM.EXE** from attempting to open a network port.

/logfile:<filename>

Specifies the pathname for a logfile which will record status, warning, and error messages. If the /logfile parameter contains an absolute path specification, the log file is placed at that location. Otherwise, if /logfile specifies a filename by itself, the log file is placed in the folder indicated by the first status message written to the Windows console after startup, typically the user's Documents directory.

The default logfile name used when no /logfile parameter is provided is **server.log**.

/port:<portnum>

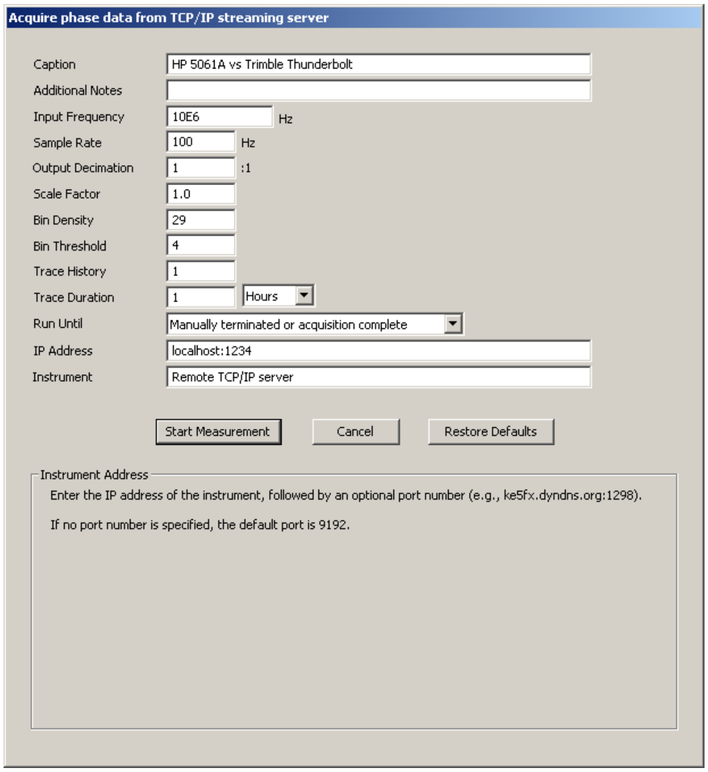
Specifies the TCP/IP port on which **STREAM.EXE** will broadcast incoming phase-difference or frequency readings. By default, **STREAM.EXE** transmits data on port 9192.

To allow remote clients to access **STREAM.EXE**, you may need to configure your firewall or NAT gateway to open this port. Instructions for doing so can be obtained from the equipment manufacturer.

When using software written to capture phase data from the Symmetricom/TSC 5115A/5120A/5125A and compatible instruments, you should specify /port:1298 as well as /format:TSC on the **STREAM.EXE** command line. With TimeLab's **Acquire→Acquire from TCP/IP streaming server** option, only the default phase-difference data format (/format:P) is supported. You can specify any desired port number as long as you also enter it in the acquisition dialog as shown below.

Example: *stream timepod.tll /port:1234 /rate:100*

This command will broadcast phase-difference data on port 1234 at a rate of 100 readings per second. You can receive the data on any machine on your LAN that can access the server, including the server PC itself. In TimeLab, the **Acquire→Acquire from TCP/IP streaming server** option could be used as shown in figure 70 below.



Caption	HP 5061A vs Trimble Thunderbolt
Additional Notes	
Input Frequency	10E6 Hz
Sample Rate	100 Hz
Output Decimation	1 :1
Scale Factor	1.0
Bin Density	29
Bin Threshold	4
Trace History	1
Trace Duration	1 Hours
Run Until	Manually terminated or acquisition complete
IP Address	localhost:1234
Instrument	Remote TCP/IP server

Instrument Address

Enter the IP address of the instrument, followed by an optional port number (e.g., ke5fx.dyndns.org:1298).

If no port number is specified, the default port is 9192.

Figure 70: Connecting to a TCP/IP streaming server

/msglvl:<0-5>

Specifies the "verbosity" of informational, warning, and error messages that **STREAM.EXE** will display on the server console. /msglvl:0 will display all available status messages and notices, including internal driver diagnostic messages, while /msglvl:5 will display only fatal error messages. The default message level is 3.

Messages displayed at the server console are also written to the logfile. See the /logfile option for more information.

/format:<P, F, TSC>

Specifies the type of measurement data that will be broadcast via TCP/IP or logged to the specified file.

STREAM.EXE writes phase-difference data by default (/format:P), but you can also record absolute frequency readings with /format:F or TSC 51xx-style phase values with /format:TSC. The latter option emits phase-difference data that has been inverted and scaled by the input frequency.

From TimeLab's perspective, one reason to use **Acquire→Acquire from TCP/IP streaming server** with **STREAM.EXE** instead of the legacy TSC acquisition driver (**Acquire→Symmetricom 5115A / 5120 A / 5125A (Frequency stability)**) is that any warning or error messages from the TimePod or PhaseStation driver will be forwarded to the TimeLab client(s) by **STREAM.EXE** for display in their status lines. In /format:TSC mode only the numeric phase data is transmitted, for compatibility reasons.

See the /input and /ref descriptions below for important additional notes.

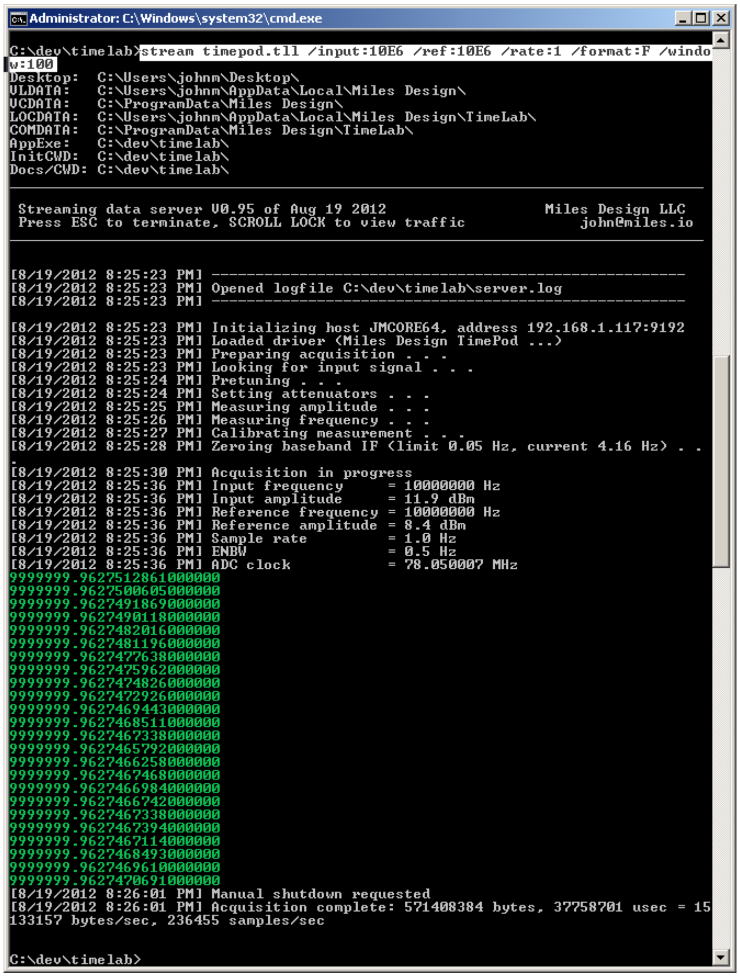
/window:<samples>

When the /format:F option is used to acquire a stream of absolute frequency measurements, each frequency reading is normally derived by subtracting the previous phase-difference reading from each incoming one and dividing the result by the reciprocal of the sampling interval. The resulting frequency difference is applied to the input frequency that was provided explicitly with the /input option or estimated when the measurement began, yielding an absolute frequency in hertz. This value, in turn, is logged to the specified file, sent to network clients via TCP/IP, and/or written to the local console window for diagnostic purposes if the Scroll Lock light is on.

It's possible to achieve much more accurate frequency readings if the frequency-difference value is computed over longer periods of time. The default /window:1 value causes frequency readings to be derived from adjacent pairs of phase-difference readings, as described above. If you specify a larger value with /window:*n*, the frequency differences are computed from phase readings separated by *n* sample intervals (or the number of intervals since the acquisition began, whichever is less.)

In the example below, an attempt was made to measure the actual output frequency from a DDS whose nominal frequency was set to 10 MHz. The use of /rate:1 yielded one reading per second at the minimum supported measurement bandwidth of 0.5 Hz. The DDS clock was derived from the same 10 MHz reference that was supplied to the TimePod, making this a true "residual frequency" measurement.

Example: *stream timepod.tll /input:10E6 /ref:10E6 /rate:1 /format:F /window:100*



```
C:\dev\timelab>stream timepod.tll /input:10E6 /ref:10E6 /rate:1 /format:F /windo
w:100
Desktop:  C:\Users\johnm\Desktop\
VLDATA:   C:\Users\johnm\AppData\Local\Miles Design\
VCDATA:   C:\ProgramData\Miles Design\
LOCDATA:  C:\Users\johnm\AppData\Local\Miles Design\TimeLab\
COMDATA:  C:\ProgramData\Miles Design\TimeLab\
AppExe:   C:\dev\timelab\
InitCWD:  C:\dev\timelab\
Docs/CWD: C:\dev\timelab\

 Streaming data server V0.95 of Aug 19 2012                   Miles Design LLC
 Press ESC to terminate, SCROLL LOCK to view traffic             john@miles.io


[8/19/2012 8:25:23 PM] -------------------------------------------------------
[8/19/2012 8:25:23 PM] Opened logfile C:\dev\timelab\server.log
[8/19/2012 8:25:23 PM] -------------------------------------------------------

[8/19/2012 8:25:23 PM] Initializing host JMCORE64, address 192.168.1.117:9192
[8/19/2012 8:25:23 PM] Loaded driver (Miles Design TimePod ...)
[8/19/2012 8:25:23 PM] Preparing acquisition . . .
[8/19/2012 8:25:23 PM] Looking for input signal . . .
[8/19/2012 8:25:24 PM] Pretuning . . .
[8/19/2012 8:25:24 PM] Setting attenuators . . .
[8/19/2012 8:25:25 PM] Measuring amplitude . . .
[8/19/2012 8:25:26 PM] Measuring frequency . . .
[8/19/2012 8:25:27 PM] Calibrating measurement . . .
[8/19/2012 8:25:28 PM] Zeroing baseband IF (limit 0.05 Hz, current 4.16 Hz) . .
.
[8/19/2012 8:25:30 PM] Acquisition in progress
[8/19/2012 8:25:36 PM] Input frequency     = 10000000 Hz
[8/19/2012 8:25:36 PM] Input amplitude     = 11.9 dBm
[8/19/2012 8:25:36 PM] Reference frequency = 10000000 Hz
[8/19/2012 8:25:36 PM] Reference amplitude = 8.4 dBm
[8/19/2012 8:25:36 PM] Sample rate         = 1.0 Hz
[8/19/2012 8:25:36 PM] ENBW                = 0.5 Hz
[8/19/2012 8:25:36 PM] ADC clock           = 78.050007 MHz
9999999.9627512861000000
9999999.9627500605000000
9999999.9627491869000000
9999999.9627490118000000
9999999.9627482016000000
9999999.9627481196000000
9999999.9627477638000000
9999999.9627475962000000
9999999.9627474826000000
9999999.9627472926000000
9999999.9627469443000000
9999999.9627468511000000
9999999.9627467338000000
9999999.9627465792000000
9999999.9627466258000000
9999999.9627467468000000
9999999.9627466984000000
9999999.9627466742000000
9999999.9627467338000000
9999999.9627467394000000
9999999.9627467114000000
9999999.9627468493000000
9999999.9627469610000000
9999999.9627470691000000
[8/19/2012 8:26:01 PM] Manual shutdown requested
[8/19/2012 8:26:01 PM] Acquisition complete: 571408384 bytes, 37758701 usec = 15
133157 bytes/sec, 236455 samples/sec


C:\dev\timelab>
```

Figure 71: Using Scroll Lock to echo measurement data

Here, the frequency readings in green were displayed by turning Scroll Lock on to view the data in real time as it was acquired. After only 24 seconds, the results have converged to within about 3 parts in 10^{-15} of the expected value of 9999999.962747097 MHz. Real-world results will vary, but in any similar test you can expect a trend towards increased accuracy as the frequency-measurement interval grows towards its requested maximum.

See the /input and /ref option descriptions below for additional notes.

/rate:<1, 10, 100, 1000>

By default, **STREAM.EXE** configures the instrument driver to make 10 measurements per second. The baseband measurement bandwidth is half of this interval, or 5 Hz by default.   The 53100A drivers limits the minimum bandwidth to 5 Hz, even when /rate:1 is used.

You can use the /rate option to select 1 reading per second to minimize network bandwidth and/or disk space for use with extremely stable sources. Conversely, measurements of drift-prone sources may require /rate:100 or /rate:1000 to avoid acquisition failures.

Note that TimePod measurements made at 1 reading per second (0.5 Hz BW) are likely to fail if insufficient warmup time is allowed for the DUT, the reference source, or the instrument itself.

/timestamp:<s, MJD>

/sep:<character>

These related options cause **STREAM.EXE** to timestamp each phase-difference or frequency reading sent to files or network clients.

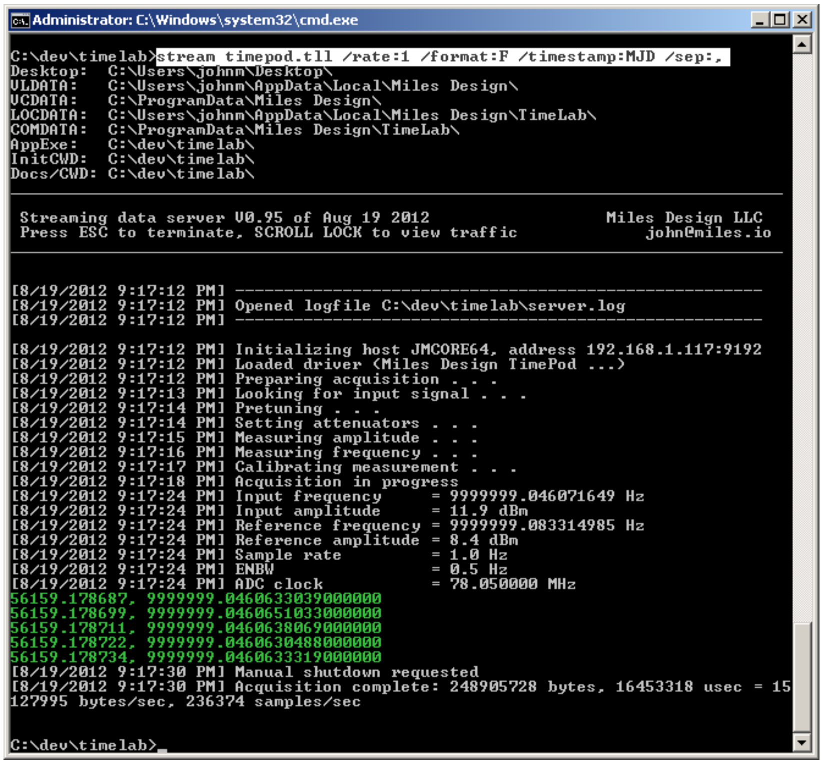
By default, no timestamps are emitted. When /timestamp:MJD is specified, the Modified Julian Date (MJD) associated with each phase-difference or frequency value will be written as the first numeric field on each line, followed by a space character (ASCII 32), then the measurement value itself. MJD timestamps are generated with six digits of decimal precision, resulting in roughly 10 unique MJD values per second.

You can also use /timestamp:s to emit a simple numeric timestamp at one-microsecond precision. These timestamps begin at 0.000000 for the first reading acquired.

In either case, the difference between successive timestamps will always be equal to the reciprocal of the /rate parameter, whose default is 10 readings per second.

The /sep option may be used to specify an additional separator character that will appear before the space between the timestamp and measurement value. If the client requires comma-separated values, for example, /sep:, would be useful. Any numeric value used as the /sep: argument is treated as an ASCII code; e.g. /sep:9 generates a tab character.

Example: *stream timepod.tll /rate:1 /format:F /timestamp:MJD /sep:,*



```
C:\dev\timelab>stream timepod.tll /rate:1 /format:F /timestamp:MJD /sep:,
Desktop:  C:\Users\johnm\Desktop\
VLDATA:   C:\Users\johnm\AppData\Local\Miles Design\
VCDATA:   C:\ProgramData\Miles Design\
LOCDATA:  C:\Users\johnm\AppData\Local\Miles Design\TimeLab\
COMDATA:  C:\ProgramData\Miles Design\TimeLab\
AppExe:   C:\dev\timelab\
InitCWD:  C:\dev\timelab\
Docs/CWD: C:\dev\timelab\

 Streaming data server V0.95 of Aug 19 2012                Miles Design LLC
 Press ESC to terminate, SCROLL LOCK to view traffic           john@miles.io


[8/19/2012 9:17:12 PM] ------------------------------------------------------
[8/19/2012 9:17:12 PM] Opened logfile C:\dev\timelab\server.log
[8/19/2012 9:17:12 PM] ------------------------------------------------------

[8/19/2012 9:17:12 PM] Initializing host JMCORE64, address 192.168.1.117:9192
[8/19/2012 9:17:12 PM] Loaded driver (Miles Design TimePod ...)
[8/19/2012 9:17:12 PM] Preparing acquisition . . .
[8/19/2012 9:17:13 PM] Looking for input signal . . .
[8/19/2012 9:17:14 PM] Pretuning . . .
[8/19/2012 9:17:14 PM] Setting attenuators . . .
[8/19/2012 9:17:15 PM] Measuring amplitude . . .
[8/19/2012 9:17:16 PM] Measuring frequency . . .
[8/19/2012 9:17:17 PM] Calibrating measurement . . .
[8/19/2012 9:17:18 PM] Acquisition in progress
[8/19/2012 9:17:24 PM] Input frequency     = 9999999.046071649 Hz
[8/19/2012 9:17:24 PM] Input amplitude     = 11.9 dBm
[8/19/2012 9:17:24 PM] Reference frequency = 9999999.083314985 Hz
[8/19/2012 9:17:24 PM] Reference amplitude = 8.4 dBm
[8/19/2012 9:17:24 PM] Sample rate         = 1.0 Hz
[8/19/2012 9:17:24 PM] ENBW                = 0.5 Hz
[8/19/2012 9:17:24 PM] ADC clock           = 78.050000 MHz
56159.178687, 9999999.046063303900000
56159.178699, 9999999.046065103300000
56159.178711, 9999999.046063806900000
56159.178722, 9999999.046063048800000
56159.178734, 9999999.046063331900000
[8/19/2012 9:17:30 PM] Manual shutdown requested
[8/19/2012 9:17:30 PM] Acquisition complete: 248905728 bytes, 16453318 usec = 15
127995 bytes/sec, 236374 samples/sec


C:\dev\timelab>_
```

Figure 72: Using the /timestamp and /sep options

/ref:<Hz>

/input:<Hz>

When acquiring data directly from a 53100A in most of the predefined measurement roles, TimeLab rounds the reference and input frequencies to the nearest 100 kHz. If you wish to obtain accurate frequency counts in TimeLab's **Measurement→Frequency difference (f)** view in measurements where one or both of these frequencies is not a multiple of 100 kHz, you can use the /ref and /input options to specify the nominal reference and/or input frequencies explicitly.

See the preceding demonstration of the /window option for an example.

/autoupdate

The /autoupdate function is useful in scenarios where **STREAM.EXE** may need to be recompiled, upgraded, or otherwise replaced over a network without manual intervention at the server console. When /autoupdate is present on the command line, **STREAM.EXE** will periodically check for the presence of a file called **STREAM.EX1** in the same directory as the executable. If **STREAM.EX1** is ever found, **STREAM.EXE** will terminate with exit(2), reporting the message "Server terminated due to release of new version."

This feature depends on the use of a batch file or script to launch **STREAM.EXE**. Upon detecting exit code 2 when **STREAM.EXE** terminates, the batch file or script should overwrite **STREAM.EXE** with a copy of **STREAM.EX1**, delete **STREAM.EX1**, and then relaunch the new **STREAM.EXE** process with the same set of command-line parameters. When any other exit code is returned by **STREAM.EXE**, the batch file or script can exit normally or take other actions.

Two examples of /autoupdate batch files are included in the TimeLab directory. **SERVE.BAT** is a generic launcher for **STREAM.EXE** that specifies the phasestation.tll and /autoupdate options on behalf of the user, passing along any other command-line options as well. **TSC.BAT** demonstrates the use of the /format:TSC and /port:1298 options along with /autoupdate.

SERVE.BAT appears below in its entirety.

```
@echo off

rem
rem Batch file to launch TCP streaming data server with 53100A driver
rem and automatic restart capability
rem

:start

rem
rem If stream.ex1 exists in the directory, rename it to stream.exe and run it
rem (passing any command-line arguments that were originally used with
rem serve.bat)
rem

if not exist stream.ex1 goto serve
copy stream.ex1 stream.exe >nul
del stream.ex1
:serve
if not exist stream.exe goto bail

stream phasestation.tll /autoupdate %*

rem
rem If stream server exits with code 2, it means that the /autoupdate option
rem detected a new copy of stream.ex1 in the directory.  Go back and launch
rem the new version without any manual intervention
rem

if errorlevel 2 goto start
:bail
```

/noexit

By default, **STREAM.EXE** will disconnect any attached clients, close its destination file (if any), and terminate with exit(1) under a variety of abnormal conditions, including loss of input or reference signals, out-of-range signal frequencies or amplitude levels, or other acquisition errors. To force the server to restart itself instead of terminating after errors that may be transient or otherwise nonfatal in nature, use the /noexit option.

Attached clients will still be disconnected when the server restarts, but they may reconnect if desired, assuming the server is able to start a new acquisition. If a destination file was specified with the /file option, it will remain open and the new acquisition process will continue to update it.

Note that certain conditions such as memory allocation failures and loss of the USB connection are still considered fatal even with /noexit, and will cause **STREAM.EXE** to terminate.

/notcp

You can use /notcp to keep **STREAM.EXE** from opening any network ports or otherwise attempting to act as a TCP/IP server. See the description of the /file option for an example.

/nowarnings

When /nowarnings is used, certain conditions such as excessive reference/input frequency and amplitude drift that would normally result in warning messages will generate error messages instead. Subsequently, **STREAM.EXE** will disconnect any attached clients, close its destination file (if any), and terminate with exit(1).

If /noexit is used in conjunction with /nowarnings, the server will attempt to restart itself under these circumstances rather than terminating. See the /noexit option description for further details.

Introduction to TSERVE

Using the TSERVE server application provided with TimeLab, phase noise and stability measurements with the PhaseStation 53100A can be configured, initiated, and recorded under the control of a remote Telnet client or other ASCII terminal application. This section describes how to run TSERVE and how to access the services it provides. Compatibility with other industry-standard phase noise test instrumentation is also demonstrated.

Launching TSERVE

TSERVE.EXE is a Windows console application. It is installed as part of the standard TimeLab software package, and is suitable for use with any 32-bit or 64-bit Windows PC that is capable of running 53100A acquisitions in the full TimeLab GUI. Since TSERVE does not display graphics, provide extensive user interaction, or work with large data files, most dual-core PCs should be capable of running it reliably.

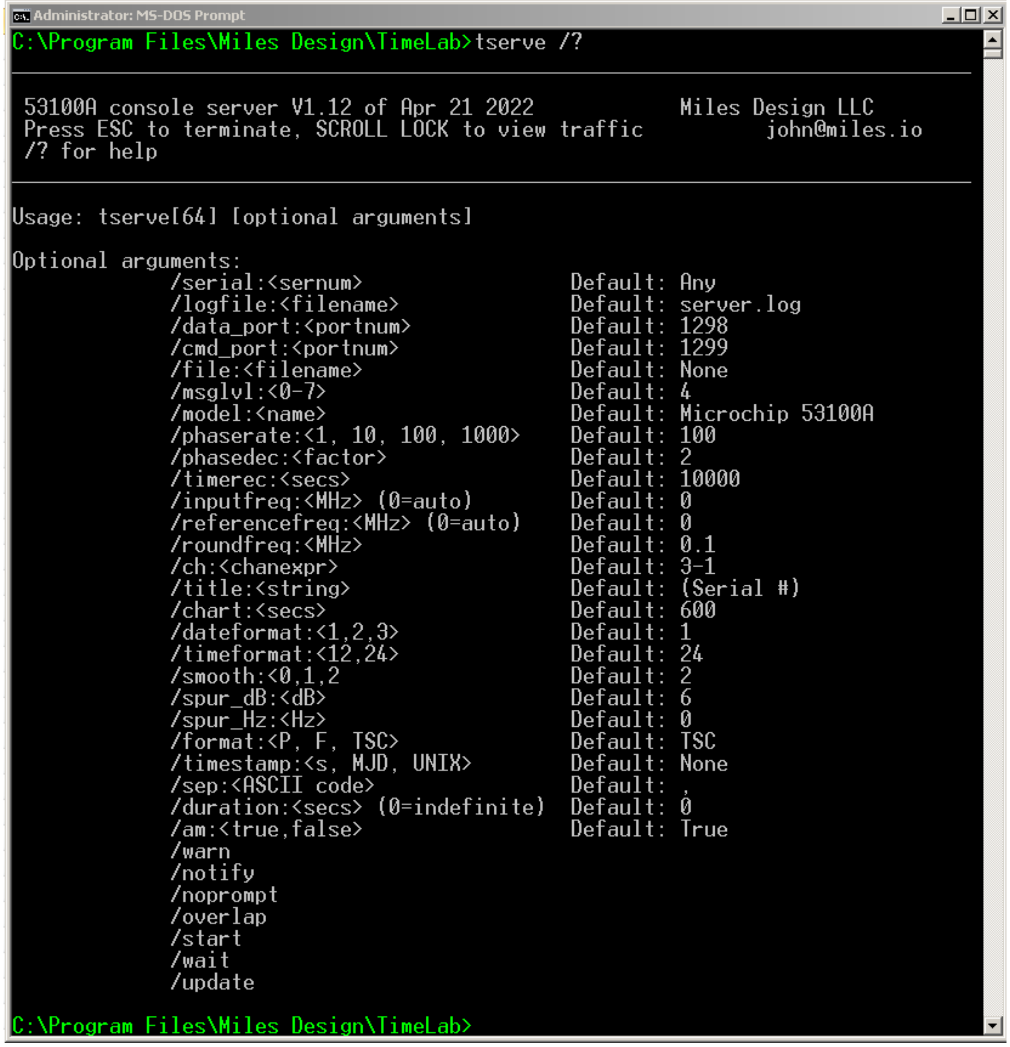
TSERVE can be launched from a Windows desktop shortcut or run manually at a command prompt ("DOS box.") Particularly during the familiarization process, you may find it easier to work with TSERVE at the command prompt. After you've determined what options and parameters are needed in your application, you can create a batch file or shortcut to launch TSERVE with the same parameters.

Unless you specified a different location during installation, you can find the program by opening a command prompt and changing the current directory to **c:\Program Files\Miles Design\TimeLab**.
To do this, select *Start* → *Run* in Windows and enter *cmd* as the name of the program to run. Once the DOS box appears, you can drag its lower edge to expand the window to a more comfortable size.
Then, enter the commands

cd "\program files\miles design\timelab"

tserve /?

As shown in Figure 73 below, the */?* option causes TSERVE to display a brief list of the command-line options available in the current release. (Throughout this document, black on white text indicates typed commands.)



```
C:\Program Files\Miles Design\TimeLab>tserve /?

 ----------------------------------------------------------------------------

 53100A console server V1.12 of Apr 21 2022               Miles Design LLC
 Press ESC to terminate, SCROLL LOCK to view traffic             john@miles.io
 /? for help

 ----------------------------------------------------------------------------

Usage: tserve[64] [optional arguments]

Optional arguments:
             /serial:<sernum>                Default: Any
             /logfile:<filename>             Default: server.log
             /data_port:<portnum>            Default: 1298
             /cmd_port:<portnum>             Default: 1299
             /file:<filename>                Default: None
             /msglvl:<0-7>                   Default: 4
             /model:<name>                   Default: Microchip 53100A
             /phaserate:<1, 10, 100, 1000>   Default: 100
             /phasedec:<factor>              Default: 2
             /timerec:<secs>                 Default: 10000
             /inputfreq:<MHz> (0=auto)       Default: 0
             /referencefreq:<MHz> (0=auto)   Default: 0
             /roundfreq:<MHz>                Default: 0.1
             /ch:<chanexpr>                  Default: 3-1
             /title:<string>                 Default: (Serial #)
             /chart:<secs>                   Default: 600
             /dateformat:<1,2,3>             Default: 1
             /timeformat:<12,24>             Default: 24
             /smooth:<0,1,2                  Default: 2
             /spur_dB:<dB>                   Default: 6
             /spur_Hz:<Hz>                   Default: 0
             /format:<P, F, TSC>             Default: TSC
             /timestamp:<s, MJD, UNIX>       Default: None
             /sep:<ASCII code>               Default: ,
             /duration:<secs> (0=indefinite) Default: 0
             /am:<true,false>                Default: True
             /warn
             /notify
             /noprompt
             /overlap
             /start
             /wait
             /update

C:\Program Files\Miles Design\TimeLab>
```

Figure 73

Note that both 32-bit and native 64-bit versions are installed on 64-bit Windows PCs. These can be launched by running TSERVE or TSERVE64, respectively. There is no functional difference between the 32-bit and 64-bit executables, but TSERVE64 may provide more reliable acquisition on slower systems due to its use of 64-bit processing.

TSERVE command line options and defaults

When TSERVE or TSERVE64 is launched without any command-line arguments, the program will listen for Telnet connections at port 1299.[17] Once a measurement has begun, ASCII phase data will be transmitted to any client(s) connected to port 1298. Parameters appropriate for emulation of the industry-standard Symmetricom/Microsemi 5115A/5120A/5125A test sets will be used by default. In this mode of operation, TSERVE will transmit phase data samples in the TSC format, which consists of

[17] Although the command port is referred to as a "Telnet" port in this document, no Telnet-specific codes or protocol elements are required. Most ASCII terminals and other communications applications should be usable with TSERVE.

negative time interval values in seconds multiplied by the nominal input frequency. This default behavior can be modified with various command-line options, described below in alphabetical order.

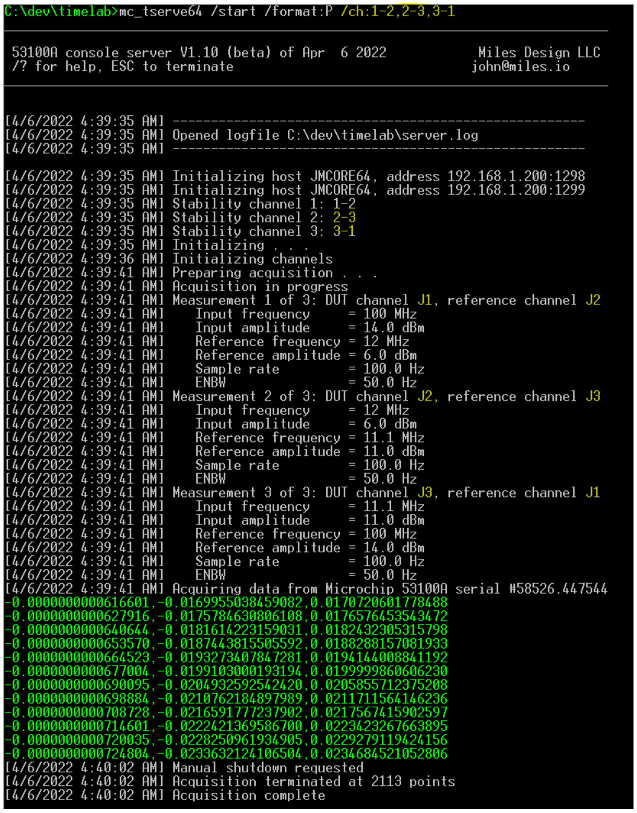
/am:<true | false>

By default, TSERVE performs AM noise measurements as well as phase noise measurements, as in the TimeLab GUI application. AM measurements can be disabled to reduce CPU workload by specifying */am:false* on the command line. The **show amspectrum** and **show amspurs** commands will display the error message *AM measurements are not enabled* in this case.

/ch:<chanexpr>

Specifies the channel configuration to be used for stability measurements.

By default, the channel expression is **3-1**, indicating that a DUT connected to the 53100A's channel 3 input (or the DUT input splitter that drives it) should be measured using the reference connected to the channel 1 input or REFERENCE input splitter. It is possible to return data for up to 4 channel pairs, sampled simultaneously.   For example, data for a three-cornered hat measurement of oscillators connected to channels 1, 2, and 3 could be obtained by specifying */ch:1-2,2-3,3-1* on the command line as shown in Figure 74 below.



```
C:\dev\timelab>mc_tserve64 /start /format:P /ch:1-2,2-3,3-1
─────────────────────────────────────────────────────────────────────────────

 53100A console server V1.10 (beta) of Apr  6 2022            Miles Design LLC
 /? for help, ESC to terminate                               john@miles.io

─────────────────────────────────────────────────────────────────────────────


[4/6/2022 4:39:35 AM] -----------------------------------------------------
[4/6/2022 4:39:35 AM] Opened logfile C:\dev\timelab\server.log
[4/6/2022 4:39:35 AM] -----------------------------------------------------

[4/6/2022 4:39:35 AM] Initializing host JMCORE64, address 192.168.1.200:1298
[4/6/2022 4:39:35 AM] Initializing host JMCORE64, address 192.168.1.200:1299
[4/6/2022 4:39:35 AM] Stability channel 1: 1-2
[4/6/2022 4:39:35 AM] Stability channel 2: 2-3
[4/6/2022 4:39:35 AM] Stability channel 3: 3-1
[4/6/2022 4:39:35 AM] Initializing . . .
[4/6/2022 4:39:36 AM] Initializing channels
[4/6/2022 4:39:41 AM] Preparing acquisition . . .
[4/6/2022 4:39:41 AM] Acquisition in progress
[4/6/2022 4:39:41 AM] Measurement 1 of 3: DUT channel J1, reference channel J2
[4/6/2022 4:39:41 AM]    Input frequency     = 100 MHz
[4/6/2022 4:39:41 AM]    Input amplitude     = 14.0 dBm
[4/6/2022 4:39:41 AM]    Reference frequency = 12 MHz
[4/6/2022 4:39:41 AM]    Reference amplitude = 6.0 dBm
[4/6/2022 4:39:41 AM]    Sample rate         = 100.0 Hz
[4/6/2022 4:39:41 AM]    ENBW                = 50.0 Hz
[4/6/2022 4:39:41 AM] Measurement 2 of 3: DUT channel J2, reference channel J3
[4/6/2022 4:39:41 AM]    Input frequency     = 12 MHz
[4/6/2022 4:39:41 AM]    Input amplitude     = 6.0 dBm
[4/6/2022 4:39:41 AM]    Reference frequency = 11.1 MHz
[4/6/2022 4:39:41 AM]    Reference amplitude = 11.0 dBm
[4/6/2022 4:39:41 AM]    Sample rate         = 100.0 Hz
[4/6/2022 4:39:41 AM]    ENBW                = 50.0 Hz
[4/6/2022 4:39:41 AM] Measurement 3 of 3: DUT channel J3, reference channel J1
[4/6/2022 4:39:41 AM]    Input frequency     = 11.1 MHz
[4/6/2022 4:39:41 AM]    Input amplitude     = 11.0 dBm
[4/6/2022 4:39:41 AM]    Reference frequency = 100 MHz
[4/6/2022 4:39:41 AM]    Reference amplitude = 14.0 dBm
[4/6/2022 4:39:41 AM]    Sample rate         = 100.0 Hz
[4/6/2022 4:39:41 AM]    ENBW                = 50.0 Hz
[4/6/2022 4:39:41 AM] Acquiring data from Microchip 53100A serial #58526.447544
-0.000000000616601,-0.0169955038459082,0.0170720601778488
-0.000000000627916,-0.0175784630806108,0.0176576453543472
-0.000000000640644,-0.0181614223159031,0.0182432305315798
-0.000000000653570,-0.0187443815505592,0.0188288157081933
-0.000000000664523,-0.0193273407847281,0.0194144008841192
-0.000000000677004,-0.0199103000193194,0.0199999860606230
-0.000000000690095,-0.0204932592542420,0.0205855712375208
-0.000000000698884,-0.0210762184897989,0.0211711564146236
-0.000000000708728,-0.0216591777237902,0.0217567415902597
-0.000000000714601,-0.0222421369586700,0.0223423267663895
-0.000000000720035,-0.0228250961934905,0.0229279119424156
-0.000000000724804,-0.0233632124106504,0.0234684521052806
[4/6/2022 4:40:02 AM] Manual shutdown requested
[4/6/2022 4:40:02 AM] Acquisition terminated at 2113 points
[4/6/2022 4:40:02 AM] Acquisition complete
```

Figure 74

In Figure 74, the */start* option has been used to initiate a measurement in the absence of any remote client connections, */ch:1-2,2-3,3-1* is used to request separate measurements of three channel pairs, and */format:P* requests phase-difference data in seconds. The Scroll Lock key has been pressed to display a subset of the outgoing data in the server console. Each column represents the phase slope over time that corresponds to the frequency of the DUT channels (1, 2, and 3 respectively) measured against the reference sources on channels 2, 3, and 1.

Phase noise and AM noise measurements in TSERVE are always performed using the default channel configuration for noise measurements in TimeLab; specifically, the DUT that drives channels 3 and 4 through an internal or external splitter is cross-correlated against the single or dual reference(s) present at channels 1 and 2.

/chart:<secs>

Specifies the maximum length of the data record returned by TSERVE in response to the **show phasediff** and **show freqdiff** commands.

By default, these commands will return the most recent 600 seconds of data. Values between 10 and 86400 seconds are accepted.

/cmd_port:<port #>

Specifies the TCP/IP port to be used for command input from a Telnet client or other ASCII terminal program.

The default port for Telnet command access is 1299. If the port number is 0, no command port will be opened, and TSERVE's Telnet server functionality will not be available.

Up to three clients at a time may log on to the command port. Note that each client can control the measurement independently; TSERVE does not attempt to arbitrate or otherwise manage control authority between clients. To allow remote clients to initiate and control PhaseStation 53100A measurements made by TSERVE, you may need to configure your firewall or NAT gateway to open the specified command port. Instructions for doing so can be obtained from the equipment manufacturer.

/data_port:<port #>

After a measurement has been initiated through the use of the */start* command-line option or the **start** Telnet command, TSERVE will begin broadcasting a continuous stream of frequency or phase-difference data on this TCP/IP port.

The stream consists of a series of double-precision real values in standard ASCII. Data is provided at the rate given by the */phaserate* and */phasedec* parameters (100 per second by default), using the format specified by the */format* parameter. One reading per line is emitted, with each line terminated with a CR+LF pair. Optionally, each reading may be prefixed by a timestamp; see the */timestamp* and */sep* options for further information.

The default port for phase/frequency data streaming is 1298. If the port number is 0, TSERVE will not attempt to open a streaming data output port.

Up to eight clients at a time may receive data via the streaming port. TSERVE ignores any commands or other input received. To allow remote clients to receive data from TSERVE, you may need to configure your firewall or NAT gateway to open the specified data port. Instructions for doing so can be obtained from the equipment manufacturer.

/dateformat:<1,2,3>

Sets the format used by the **show date** command.

1 = dd (abbreviated name of month) yyyy
2 = mm/dd/yyyy
3 = dd/mm/yyyy

The default date format is 1.

/duration:<seconds>

Specifying a nonzero duration with this option will cause TSERVE to exit after the specified number of seconds.

/file:<filename>

Specifies the name of an optional ASCII text file to which streaming phase-difference or frequency readings will be written. If the specified file already exists, it will be overwritten. Typically the file is shared with other applications that can access it in read-only mode while it's being written by TSERVE.

The data written to the file may be 'followed' from within TimeLab by selecting **Acquire→Acquire from live ASCII file**. You can also monitor the data written to the file with a command such as *tail –f <filename>* from Cygwin. Finally, pressing the **Scroll Lock** key will cause TSERVE to display the lines written to the file on the stdout device (i.e., the DOS console itself) at a maximum rate of 10 readings per second.

Example: *tserve64 /data_port:0 /cmd_port:0 /start /format:P /file:%HOMEPATH%\Documents\mydata.txt*

The command above writes a continuous stream of phase-difference readings to the file **mydata.txt** in the current user's Documents folder. No TCP/IP ports are opened for either data streaming or command input. You can then use TimeLab or another application to 'follow' the file. Even if the acquisition fails or otherwise encounters problems, you can use the appropriate **File→Import** option to bring the data into TimeLab at any time.

In addition to specifying the output file on the TSERVE command line, connected Telnet clients can also open and close the output file remotely with the **open** and **close** commands.

/format:<P, F, TSC>

Specifies the type of measurement data that will be broadcast via the TCP/IP data port and/or logged to the specified file.

TSERVE writes Symmetricom 51XXA-style phase-difference data by default (*/format:TSC*). This format consists of phase-difference values that have been inverted and multiplied by the input frequency. You can also generate conventional (unscaled) phase differences with */format:P* or absolute frequency readings with */format:F*.

Optionally, any warning or error messages can be forwarded to the connected TimeLab client(s) by TSERVE for display in the status line. This capability requires the use of the */notify* option on the TSERVE command line. If */notify* is not used, data sent to the TCP/IP streaming clients and/or shared file will always consist of numeric readings in the specified format.

/inputfreq:<MHz>

By default, the DUT and reference input frequencies are measured at acquisition startup time and rounded to the nearest 100 kHz. These nominal frequencies are reported by the **show inputs** command and used as the basis for phase-difference readings. If either frequency does not accurately reflect the nominal frequency of the corresponding input source, the discrepancy will appear as a nonzero slope in successive phase-difference readings. If the phase slope is due to insufficient DUT frequency precision, the */inputfreq* option can be used to specify the precise DUT input frequency to be used instead of the estimated value. Alternatively, */roundfreq* can be used to force the nominal input frequency to be logged at higher precision.

For example, if a DUT with a nominal frequency of 10.23 MHz is being measured and it is desirable to minimize the slope of the phase-difference readings , either */inputfreq:10.23* or */roundfreq:0.01* could be used.

Note that the accuracy of the absolute frequency readings obtained with the */format:F* switch depends only on the reference source, regardless of the nominal input frequency reported by **show inputs**. Consequently, it is unaffected by the */inputfreq* or */roundfreq* options. As a result, the */inputfreq* and */roundfreq* options are not required in most TSERVE measurements. However, note that if the nominal reference frequency is not a multiple of 100 kHz, it is necessary to use */referencefreq* to avoid inaccurate results regardless of the selected output format. See the */referencefreq* option for more information.

/logfile:<filename>

Specifies the pathname for a logfile which will record status, warning, and error messages. If the */logfile* parameter contains an absolute path specification, the log file is placed at that location. Otherwise, if */logfile* specifies a filename by itself, the log file is placed in the user's Documents directory. The location of the log file is displayed when TSERVE is launched.

The default logfile name used when no */logfile* parameter is provided is **server.log**.

/model:<name>

Specifies the instrument name that will be returned by the **show version** command. The instrument name is also used to present a logon banner ("Welcome to the ____") when a new connection Is made to the Telnet command port.

The default model name is *Microchip 53100A*. The specified model name has no effect on any aspect of operation except as described above. A different model name may be required for compatibility with legacy software that expects to communicate with (e.g.) a Symmetricom 5125A:

Example: *tserve64 /model:"Symmetricom 5125A" /start*

/msglvl:<0-7>

Specifies the "verbosity" of informational, warning, and error messages that TSERVE will display on the server console. */msglvl:0* will display all available status messages and notices including internal driver diagnostics, while */msglvl:7* will display only fatal error messages. The default message level is 4.

Messages displayed at the server console are also written to the logfile. See the */logfile* option for more information.

/noprompt

By default, clients attached to the Telnet command port will receive a command prompt containing the TSERVE host IP address. Issuing the **prompt off** command or specifying the */noprompt* argument on the TSERVE command line will prevent the prompt from being displayed.

/notify

Optionally, any warning or error messages that occur during data acquisition can be forwarded to clients via the TCP/IP data port used for phase/frequency streaming. To enable this capability, include the */notify* option on the TSERVE command line.

If */notify* is not used, data sent to the TCP/IP streaming clients and/or shared file will always consist of numeric readings in the specified format.

/overlap

For maximum reliability on slower host PCs, overlapped FFT processing during phase noise measurement is turned off by default in TSERVE. Specifying the */overlap* option can significantly improve the speed required for low-noise measurements to converge, at the risk of possible data overruns.

Specifying the */overlap* option on the TSERVE command line is equivalent to checking the *Overlapped acquisition* checkbox in the *Additional options* page of the 53100A acquisition dialog.

/phasedec

/phaserate

These options determine the default sample rate for frequency and phase-difference data streamed to TCP/IP clients connected to the data port.

When a client initially establishes a Telnet connection to the TSERVE command port, it inherits various global settings including the */phaserate* parameter. In this case, the */phaserate* parameter becomes the **phaserate** value for measurements initiated by that client. This value can be displayed with the **show phaserate** command or modified with the **set phaserate** command at any time prior to starting a measurement. Using either of these techniques, you can set the phase rate to 1 reading per second to conserve network bandwidth and/or disk space for use with extremely stable sources. Conversely, measurements of drift-prone sources may require phase data rates of 100 or 1000 readings per second to avoid acquisition failures.

The choice of phase data rate also determines the baseband measurement bandwidth for frequency and phase stability. In TSERVE, the equivalent noise bandwidth (ENBW) is half of the */phaserate* interval, or 50 Hz by default. (Note that the PhaseStation 53100A driver limits the minimum bandwidth to 5 Hz, even when */phaserate:1* is used.)

The */phasedec* parameter provides some additional flexibility by allowing you to specify a larger-than-normal decimation ratio for the post-detection phase data. Its default value is 2, corresponding to the default *Output decimation* property on the *Additional options* page of the PhaseStation 53100A acquisition dialog. The nominal phase rate options of 1, 10, 100, and 1000 readings per second are based on the default */phasedec* value of 2. Setting */phasedec* to larger values reduces the overall rate at which frequency and phase-difference data is streamed to TSERVE clients and any output file specified with the */file* parameter, and also determines the tau0 interval for the **show**

adev command. Regardless of any changes to */phasedec, t*he equivalent noise bandwidth is still determined by the **phaserate** value that was in effect when the measurement began.

If not overridden by a command-line argument, the */phaserate* value defaults to 100 samples per second. Note that some legacy Symmetricom/Microsemi 51XXA applications may require a phase data rate of 1000 samples per second. Unlike these test sets, any changes made to parameters such as **phaserate**, **tau0**, and **referencefreq** will not affect either queried or streamed data from the measurement in progress. As a result, the data rate must either be set from the command line when TSERVE is started, or by issuing a **set phaserate** command prior to the **start** command.

/referencefreq:<MHz>

At measurement startup time, TSERVE estimates the frequency of the signal at the 53100A's REFERENCE input by rounding it to the nearest 100 kHz. This provides an authoritative basis for calculation of the applied DUT input frequency, assuming that the reference frequency is indeed a multiple of 100 kHz. If this is not the case, you should use the */referencefreq* option to specify the nominal reference frequency explicitly.

For example, if a 10.5 MHz oscillator is used as a reference source, the */referencefreq* parameter does not need to be specified. However, if a 10.23 MHz oscillator is used as the reference, it is necessary to specify */referencefreq:10.23* to avoid inaccurate results.

The nominal reference frequency can be changed by remote clients with **set referencefreq**. Whether it is measured or specified, the reference frequency is inherited by Telnet command clients at connection time and used as the **referencefreq** parameter during subsequent measurements. Any changes made with **set referencefreq** while a measurement is in progress will have no effect on frequency-count or streamed frequency/phase-difference readings reported during the current measurement.

/roundfreq:<MHz>

By default, the DUT and reference input frequencies are measured at acquisition startup time and rounded to the nearest 100 kHz. These nominal frequencies are reported by the **show inputs** command and used as the basis for phase-difference readings. If either frequency does not accurately reflect the nominal frequency of the corresponding input source, the discrepancy will appear as a nonzero slope in successive phase-difference readings. If the phase slope is due to insufficient DUT frequency precision, the */roundfreq* option can be used to force the nominal input frequency to be logged at higher precision. Alternatively, */inputfreq* can be used to specify the precise DUT input frequency to be used instead of the estimated value.

For example, if a DUT with a nominal frequency of 10.23 MHz is being measured and it is desirable to minimize the slope of the phase-difference readings , either */inputfreq:10.23* or */roundfreq:0.01* could be used. The **show inputs** command will report an input frequency of 10.23 MHz in either case.

Note that the accuracy of the absolute frequency readings obtained with the */format:F* switch depends only on the reference source, regardless of the nominal input frequency reported by **show inputs**. Consequently, it is unaffected by the */inputfreq* or */roundfreq* options. As a result, the */inputfreq* and */roundfreq* options are not required in most TSERVE measurements. However, note that if the nominal reference frequency is not a multiple of 100 kHz, it is necessary to use */referencefreq* to avoid inaccurate results regardless of the selected output format. There is no corresponding way to increase the reference frequency's rounding precision; the nominal reference frequency must be specified explicitly with */referencefreq* if 100-kHz precision is insufficient.

/ sep:<character>

/ timestamp:<s, MJD, UNIX>

These related options cause TSERVE to add a timestamp to each phase-difference or frequency reading that is written to a shared file or streamed to network clients.

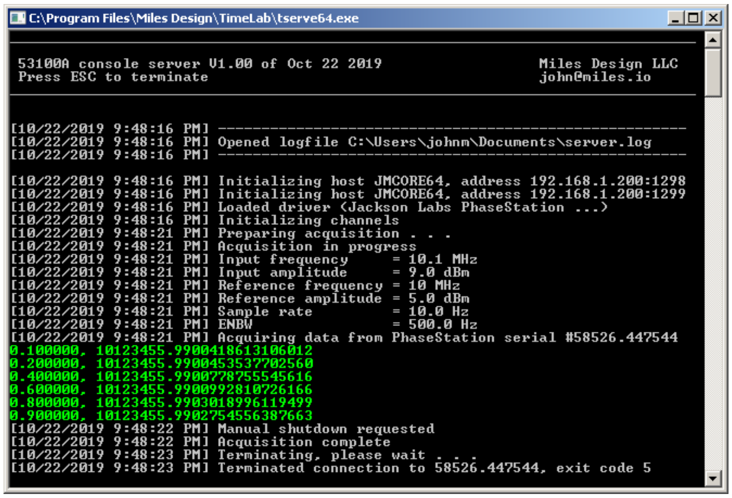
By default, no timestamps are emitted. When */timestamp:MJD* is specified, the Modified Julian Date (MJD) associated with each phase-difference or frequency value will be written as the first numeric field on each line, followed by a space character (ASCII 32), then the measurement value itself. MJD timestamps are generated with six digits of decimal precision, resulting in roughly 10 unique MJD values per second.

You can also use */timestamp:s* to emit a simple numeric timestamp at one-microsecond precision. These timestamps begin at 0.000000 for the first reading acquired. Finally, */timestamp:UNIX* can be used to request numeric timestamps at one-millisecond precision relative to the standard Unix epoch (January 1st, 1970 at 00:00:00 UTC.)

In all cases, the difference between successive timestamps will be equal to the reciprocal of the */phaserate* parameter, whose default is 100 readings per second, scaled further by any specified */phasedec* ratio beyond its default value of 2:1. Refer to the description of the */phaserate* parameter for more information.

The */sep* option may be used to specify an additional separator character that will appear before the space between the timestamp and measurement value. If the client requires comma-separated values, for example, */sep:,* would be useful. Any nonzero numeric value used as the */sep:* argument is treated as an ASCII code; e.g. */sep:9* generates a tab character.

Example: *tserve64 /start /phaserate:1000 /phasedec:200 /format:F /timestamp:s /sep:,*



```
 53100A console server V1.00 of Oct 22 2019                 Miles Design LLC
 Press ESC to terminate                                     john@miles.io

[10/22/2019 9:48:16 PM] ----------------------------------------------------
[10/22/2019 9:48:16 PM] Opened logfile C:\Users\johnm\Documents\server.log
[10/22/2019 9:48:16 PM] ----------------------------------------------------

[10/22/2019 9:48:16 PM] Initializing host JMCORE64, address 192.168.1.200:1298
[10/22/2019 9:48:16 PM] Initializing host JMCORE64, address 192.168.1.200:1299
[10/22/2019 9:48:16 PM] Loaded driver (Jackson Labs PhaseStation ...)
[10/22/2019 9:48:16 PM] Initializing channels
[10/22/2019 9:48:21 PM] Preparing acquisition . . .
[10/22/2019 9:48:21 PM] Acquisition in progress
[10/22/2019 9:48:21 PM] Input frequency     = 10.1 MHz
[10/22/2019 9:48:21 PM] Input amplitude     = 9.0 dBm
[10/22/2019 9:48:21 PM] Reference frequency = 10 MHz
[10/22/2019 9:48:21 PM] Reference amplitude = 5.0 dBm
[10/22/2019 9:48:21 PM] Sample rate         = 10.0 Hz
[10/22/2019 9:48:21 PM] ENBW                = 500.0 Hz
[10/22/2019 9:48:21 PM] Acquiring data from PhaseStation serial #58526.447544
0.100000, 10123455.9900418613106012
0.200000, 10123455.9900453537702560
0.400000, 10123455.9900778755545616
0.600000, 10123455.9900992810726166
0.800000, 10123455.9903018996119499
0.900000, 10123455.9902754556387663
[10/22/2019 9:48:22 PM] Manual shutdown requested
[10/22/2019 9:48:22 PM] Acquisition complete
[10/22/2019 9:48:23 PM] Terminating, please wait . . .
[10/22/2019 9:48:23 PM] Terminated connection to 58526.447544, exit code 5
```

Figure 75

Note that without the */phasedec:200* parameter, the timestamps shown in green in Figure 75 above would have advanced at the rate of 0.001 second per reading rather than 0.1 second. This effect would be apparent to any connected streaming-data clients but would not necessarily be visible in the server console, since the readings obtained with the Scroll Lock monitor function are limited to 10 readings per second.

/serial:<sernum>

When more than one 53100A is connected to the server PC, you may need to use the */serial* option to associate a given TSERVE instance with a specific instrument. Each instance of TSERVE supports only one acquisition at a time, but you can launch as many instances as desired.

Example: *tserve64 /serial:55908.304876 /cmd_port:1234 /data_port:1235*

The command above will launch an instance of TSERVE for use with the 53100A whose serial number is 55908.304876. To avoid addressing conflicts, each TSERVE instance must specify unique TCP/IP ports for command and data traffic.

/smooth:<0,1,2>

The */smooth* option determines the filtering operations applied to phase noise data returned by the **show spectrum** command.

0 = No smoothing or spur suppression
1 = Spurs are removed from the trace but no smoothing is applied
2 = Spurs are removed from the trace and smoothing is applied (default)

In most cases the smoothing parameter should be left at its default value of 2. Spurs that are present in the phase noise spectrum have undergone the same 1-Hz normalization process as the noise data itself, so they will rarely appear at their true amplitudes. Use the **show spurs** command to retrieve the offsets and amplitudes of detected spurs.

/spur_dB:<dB>

This parameter corresponds to the *Spur threshold* field in the *Additional options* page of the PhaseStation 53100A acquisition dialog. It determines the amplitude above the local average phase noise level at which a discrete spectral line is recognized as a coherent spur. The default spur detection threshold is 6.0 dB.

/spur_Hz:<Hz>

This parameter corresponds to the *Spur min offset* field in the *Additional options* page of the PhaseStation 53100A acquisition dialog. Spurs with offset frequencies below this value in Hz will not be removed from the plot or added to the spur chart. This may be helpful when one or more spurs close to the carrier are being erroneously flagged. The default minimum spur offset is 0 Hz, which disables the feature.

/start

Causes TSERVE to begin an acquisition immediately after launch.

If */start* is not specified on the command line, no measurement will begin until a client logs onto the Telnet command port and issues a **start** command.

/timeformat:<12,24>

Sets the format used by the **show time** command. */timeformat:12* specifies 12-hour time while */timeformat:24* specifies 24-hour time.

/timerec:<s>

Determines the size of the internal phase record used for ADEV calculation.

ADEV calculations performed by TSERVE are based on a 1-2-4 distribution of tau bins in each decade, using a minimum threshold of 4 samples per bin. Assuming that the */phasedec* parameter is left at its default value of 2, ADEV results will be returned for tau intervals ranging from the minimum t_0 value given by the reciprocal of the */phaserate* parameter (100 seconds by default, yielding t_0=0.01 seconds) to a maximum of 1/5 of the */timerec* parameter (10000 seconds by default, yielding t_{max}=2000 seconds).

Note that */phasedec* parameter values larger than the default (2) will increase the minimum tau interval while leaving the maximum interval unchanged. For example, if TSERVE was launched with */timerec:100* and */phasedec:20*, ADEV values from t=0.1s to t=20s will be available. Also note that since TSERVE must scroll the phase record in memory during long-term measurements, increasing the */timerec* and */phaserate* parameter values simultaneously may result in measurement termination due to USB buffer starvation. Should this occur, you may find it helpful to lower the */phaserate* value or increase the */phasedec* value proportionately when increasing the phase record size. (Both */phaserate* and */phasedec* affect the t_0 interval, but a reduction in */phaserate* will also lower the measurement bandwidth.)

The */timerec* parameter also determines the maximum size of the phase record in .TIM files written by the **save** command. Refer to the **save** command description for more information.

/timestamp:<s, MJD, UNIX>

See */sep*.

/title:<string>

Establishes the title text returned by the **show title** command.

By default, the title is the reserved string **(Serial #),** which causes the serial number of the connected instrument to be returned by **show title**. Refer to the **show title** command description for more information.

/update

The */update* function is useful in scenarios where **TSERVE[64].EXE** may need to be recompiled, upgraded, or otherwise replaced over a network without manual intervention at the server console. When */update* is present on the command line, TSERVE will periodically check for the presence of a file called **TSERVE[64].EX1** in the same directory as the executable. If **TSERVE[64].EX1** is ever found, the program will terminate with exit(2), reporting the message "Server terminated due to release of new version."

This feature depends on the use of a batch file or script to launch **TSERVE[64].EXE**. Upon detecting exit code 2, the batch file or script should overwrite the program executable with a copy of **TSERVE[64].EX1**, then delete the **TSERVE[64].EX1** file and relaunch the new **TSERVE[64].EXE** process with the same set of command-line parameters. When any other exit code is returned by the program, the batch file or script can exit normally or take other actions.

As an example, **TSC.BAT** is provided in the TimeLab installation directory to demonstrate the use of the */update* and */model* options. Any additional command-line options specified by the user are passed along to the program as well. **TSC.BAT** appears below in its entirety.

```
@echo off

rem
rem Batch file to emulate a Symmetricom 5125A with the PhaseStation 53100A
rem

:start

rem
rem If tserve64.ex1 exists in the directory, rename it to tserve64.exe and run it
rem (passing any command-line arguments that were originally used with tsc.bat)
rem

if not exist tserve64.ex1 goto serve
copy tserve64.ex1 tserve64.exe >nul
del tserve64.ex1
:serve
if not exist tserve64.exe goto bail

tserve64 /model:"Symmetricom 5125A" /update %*

rem
rem If stream server exits with code 2, it means that the /autoupdate option
rem detected a new copy of tserve64.ex1 in the directory.  Go back and launch
rem the new version without any manual intervention
rem

if errorlevel 2 goto start
:bail
```

/wait

If the */wait* option is specified on the TSERVE command line, the program will prompt for a keypress before terminating, except when the */update* mechanism has been triggered by the appearance of a newer executable version. See the description of the */update* option below for more information.

Typically the */wait* option is used to keep Windows from discarding useful status or error information by closing the console window immediately upon termination. It is not necessary when TSERVE is launched directly at an MS-DOS prompt, but may be helpful when launching from a desktop shortcut.

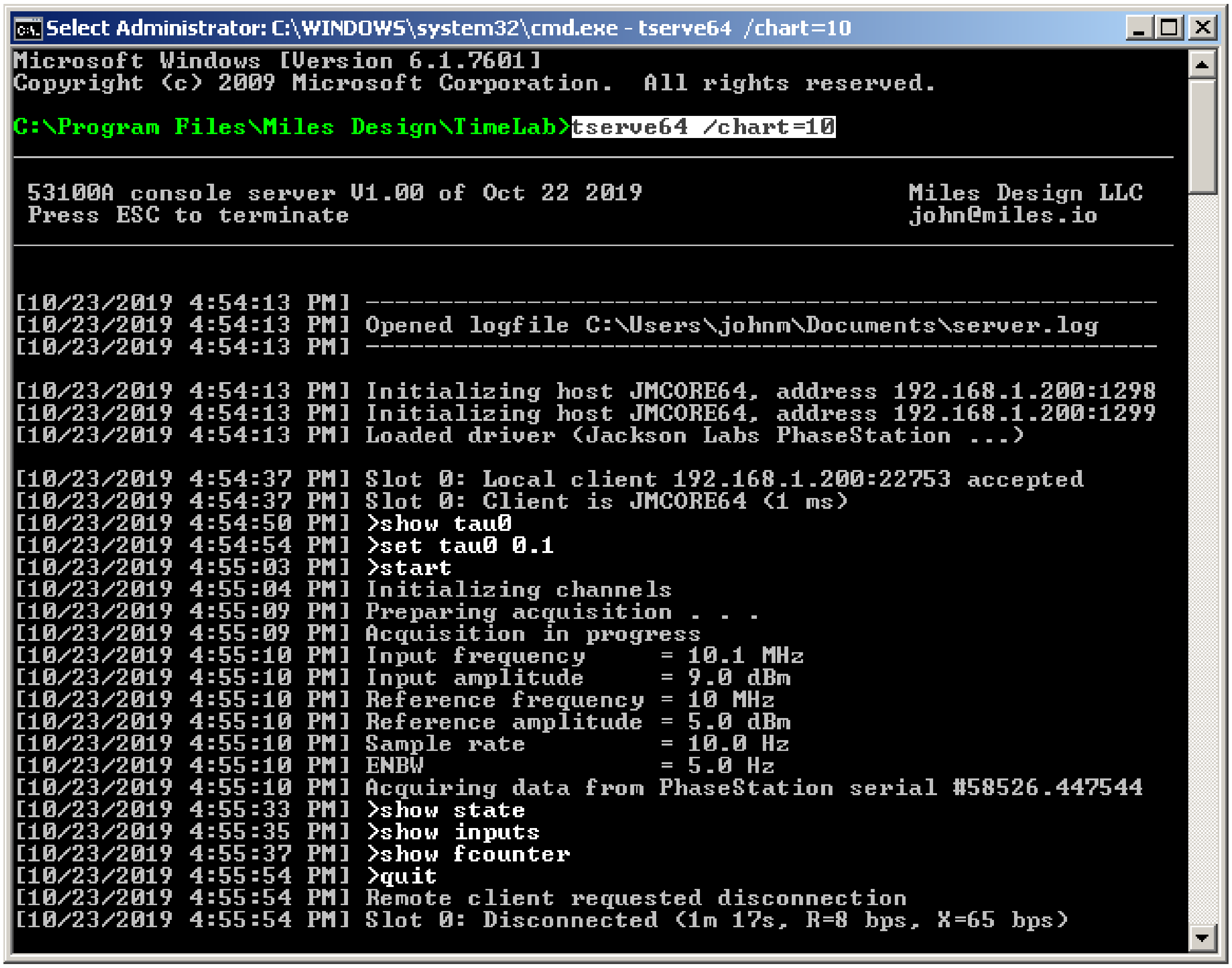
/warn

By default, a TSERVE-hosted measurement will stop If any significant changes in signal amplitude or frequency are observed at either the DUT or REFERENCE input ports. Using the */warn* option will allow the measurement to continue running when non-fatal anomalies are encountered. This option is equivalent to unchecking the *All warnings fatal* option in the *Additional options* page of the PhaseStation 53100A acquisition dialog.

When a measurement stops due to an error condition, any streaming data connections will remain open, and any Telnet or terminal clients will remain connected. Clients attached to the command port can query the status of the measurement with the **show state** and **show message** commands, while clients connected to the streaming data port will receive error and warning messages if the */notify* option was specified on the TSERVE command line.

TSERVE remote commands

A variety of commands and queries may be issued by a Telnet or other ASCII terminal client connected to the TSERVE command port. These commands will appear in the console TTY display as shown in Figure 76 below, and will also be written to the server log file.

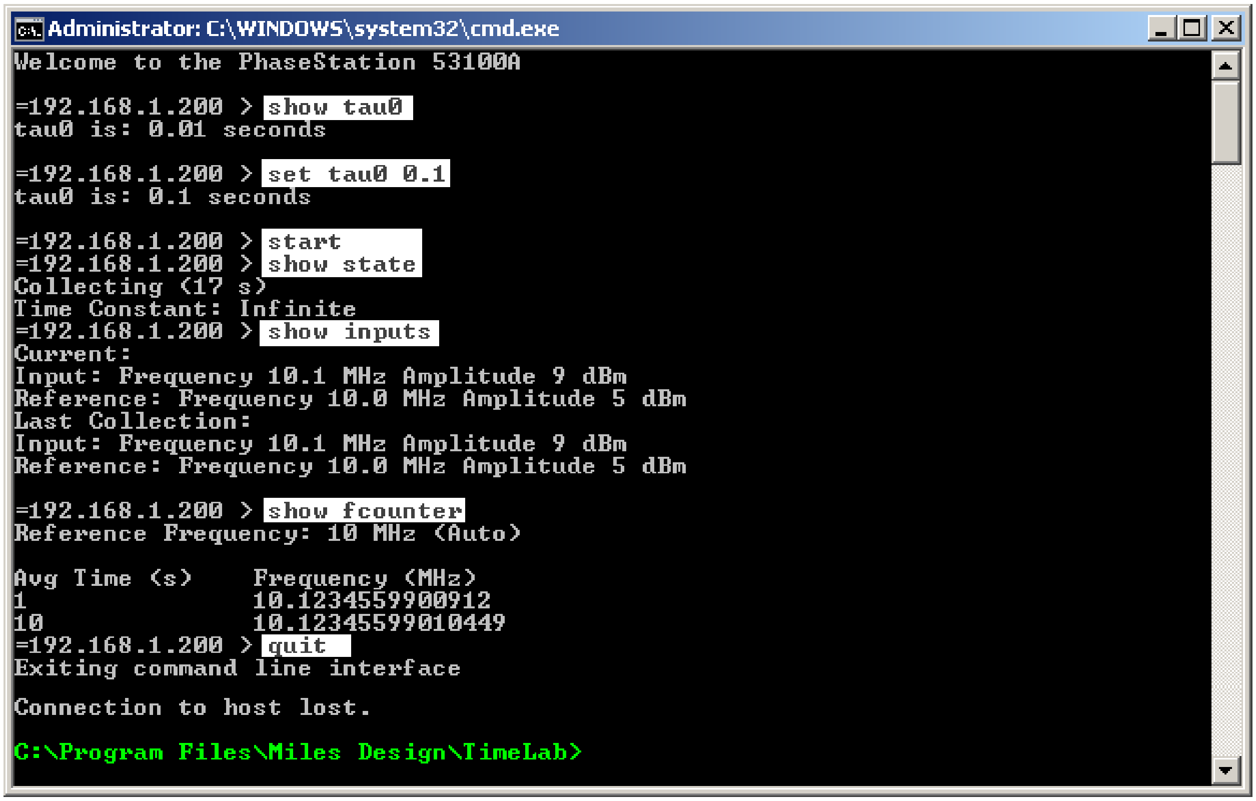

Select Administrator: C:\WINDOWS\system32\cmd.exe - tserve64 /chart=10

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Program Files\Miles Design\TimeLab>tserve64 /chart=10
______________________________________________________________________________

 53100A console server V1.00 of Oct 22 2019                       Miles Design LLC
 Press ESC to terminate                                           john@miles.io
______________________________________________________________________________


[10/23/2019 4:54:13 PM] ------------------------------------------------------
[10/23/2019 4:54:13 PM] Opened logfile C:\Users\johnm\Documents\server.log
[10/23/2019 4:54:13 PM] ------------------------------------------------------

[10/23/2019 4:54:13 PM] Initializing host JMCORE64, address 192.168.1.200:1298
[10/23/2019 4:54:13 PM] Initializing host JMCORE64, address 192.168.1.200:1299
[10/23/2019 4:54:13 PM] Loaded driver (Jackson Labs PhaseStation ...)

[10/23/2019 4:54:37 PM] Slot 0: Local client 192.168.1.200:22753 accepted
[10/23/2019 4:54:37 PM] Slot 0: Client is JMCORE64 (1 ms)
[10/23/2019 4:54:50 PM] >show tau0
[10/23/2019 4:54:54 PM] >set tau0 0.1
[10/23/2019 4:55:03 PM] >start
[10/23/2019 4:55:04 PM] Initializing channels
[10/23/2019 4:55:09 PM] Preparing acquisition . . .
[10/23/2019 4:55:09 PM] Acquisition in progress
[10/23/2019 4:55:10 PM] Input frequency     = 10.1 MHz
[10/23/2019 4:55:10 PM] Input amplitude     = 9.0 dBm
[10/23/2019 4:55:10 PM] Reference frequency = 10 MHz
[10/23/2019 4:55:10 PM] Reference amplitude = 5.0 dBm
[10/23/2019 4:55:10 PM] Sample rate         = 10.0 Hz
[10/23/2019 4:55:10 PM] ENBW                = 5.0 Hz
[10/23/2019 4:55:10 PM] Acquiring data from PhaseStation serial #58526.447544
[10/23/2019 4:55:33 PM] >show state
[10/23/2019 4:55:35 PM] >show inputs
[10/23/2019 4:55:37 PM] >show fcounter
[10/23/2019 4:55:54 PM] >quit
[10/23/2019 4:55:54 PM] Remote client requested disconnection
[10/23/2019 4:55:54 PM] Slot 0: Disconnected (1m 17s, R=8 bps, X=65 bps)
```

Figure 76

The TSERVE example session shown above was launched from an MS-DOS prompt. (Black-on-white characters indicate text entered by the user at the prompt.) The 64-bit version of TSERVE was run by typing **tserve64**, followed by the */chart=10* command-line option.

Shortly after the server was started, a connection was established from another DOS box on the same PC as shown in Figure 77 below, using the command **telnet 192.168.1.200 1299**:



```
Welcome to the PhaseStation 53100A

=192.168.1.200 > show tau0
tau0 is: 0.01 seconds

=192.168.1.200 > set tau0 0.1
tau0 is: 0.1 seconds

=192.168.1.200 > start
=192.168.1.200 > show state
Collecting (17 s)
Time Constant: Infinite
=192.168.1.200 > show inputs
Current:
Input: Frequency 10.1 MHz Amplitude 9 dBm
Reference: Frequency 10.0 MHz Amplitude 5 dBm
Last Collection:
Input: Frequency 10.1 MHz Amplitude 9 dBm
Reference: Frequency 10.0 MHz Amplitude 5 dBm

=192.168.1.200 > show fcounter
Reference Frequency: 10 MHz (Auto)

Avg Time (s)    Frequency (MHz)
1               10.1234559900912
10              10.12345599010449
=192.168.1.200 > quit
Exiting command line interface

Connection to host lost.

C:\Program Files\Miles Design\TimeLab>
```

Figure 77

In this example, several different commands and queries were transmitted to TSERVE from the Telnet client application, including the **start** command to begin data acquisition. These user-entered commands appear in black-on-white text in the Telnet client window and are displayed in bold white text at the server console. After the user issued the final **quit** command at 4:55:54 PM, the Telnet session ended. The server (and the measurement acquisition) continued to run, awaiting further connections.

An alphabetical list of supported commands and queries appears below. Except as noted, TSERVE's remote command syntax is compatible with the Microsemi/Symmetricom/TSC 5125A and other phase noise test sets in the 51XXA series. Applications and test scripts written for these instruments should run on a PhaseStation 53100A with little or no adaptation.

Note that multiple commands may be entered on one line by separating them with semicolons.

beep

Plays a sound on the TSERVE host PC.

close

Closes the output stream data file most recently opened with the **open** command or */file:* command-line option.

exit

See **quit.**

help [<command>]

Returns a list of commands with brief usage information. If a command name is specified, only the help text for that command is returned.

history

Returns a list of previously-entered commands.

logout

See **quit.**

measurelinear

Computes the slope and intercept coefficients of the current phase-difference chart. After a **measurelinear** command has been issued, a subsequent **removelinear on** command will cause the linear trendline to be removed from the chart data returned by **show phasediff.**

If **measurelinear** is executed while **removelinear on** is in effect, the stored coefficients will be updated for use by subsequent **show phasediff** commands. It is not necessary to use **removelinear off** before issuing a **measurelinear** command.

open <filename>

Opens an ASCII text file to which streaming phase-difference or frequency readings will be written. If the specified file already exists, it will be overwritten. Typically the file is shared with other applications that can access it in read-only mode while it's being written by TSERVE.

The data written to the file may be 'followed' from within TimeLab by selecting **Acquire→Acquire from live ASCII file**. You can also monitor the data written to the file with a command such as *tail –f <filename>* from Cygwin. Finally, pressing the **Scroll Lock** key will cause TSERVE to display the lines written to the file on the stdout device (i.e., the DOS console itself) at a maximum rate of 10 readings per second.

Example: *tserve64 /data_port:0 /cmd_port:0 /start /format:P /file:%HOMEPATH%\Documents\mydata.txt*

The command above writes a continuous stream of phase-difference readings to the file **mydata.txt** in the current user's Documents folder. No TCP/IP ports are opened for either data streaming or command input. You can then use TimeLab or another application to 'follow' the file. Even if the acquisition fails or otherwise encounters problems, you can use the appropriate **File→Import** option to bring the data into TimeLab at any time.

The output file may be flushed and closed at any time by issuing a **close** command or by **open**ing a different file. In addition to the **open** command, a stream output file can also be opened at TSERVE startup time by specifying the */file: <filename>* command-line option.

pause <chart>

pause phasediff and **pause freq** will inhibit updates to the phase difference and frequency difference charts, respectively. Updates may be resumed with **resume phasediff** or **resume freq**. (The inconsistent chart nomenclature is required for compatibility with the TSC 51XXA command language.)

prompt <on|off>

The **prompt off** and **prompt on** commands will disable and enable the Telnet command prompt, respectively. The prompt is enabled by default. **prompt off** is equivalent to the */noprompt* command-line option.

quit

Closes the Telnet connection. The TSERVE server will continue running, as will any measurement in progress.

removelinear <on|off>

When **removelinear on** is in effect, the linear trendline most recently measured by the **measurelinear** command will be removed from subsequent phase-difference charts returned by the **show phasediff** command. Use **removelinear off** to disable linear-trend subtraction.

reset

restorefactorydefaults

When the Telnet server thread is created by TSERVE at startup time, copies of the default values for the **phaserate**, **tau0**, **referencefreq**, **inputfreq, prompt**, **dateformat**, **timeformat**, and **title** properties are associated with the server instance, taking into account any changes made to the program defaults at the TSERVE command line. These properties subsequently apply to any measurements initiated by either Telnet clients or by a */start* parameter on the TSERVE command line. Clients may alter these server property values by issuing the appropriate **set** commands. Changes made to the server property values will apply to acquisitions that are subsequently initiated by that client or any others.

Issuing a **reset** or **restorefactorydefaults** command will immediately stop any measurement in progress. TSERVE will continue running and any clients will remain connected, but the server default properties will be restored to the program default or command-line values, just as if TSERVE had been terminated and restarted. The **reset** and **restorefactorydefaults** commands behave identically.

resume <chart>

See **pause**.

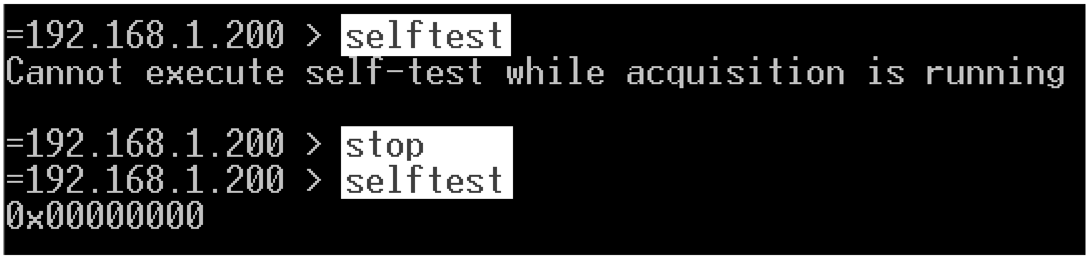
save <filename.TIM>

Saves the current phase record and phase noise trace data to the specified .TIM file. The saved file may subsequently be loaded for inspection and editing by the TimeLab GUI application.

The length of the phase record in the saved .TIM file is determined by the /*timerec* parameter specified on the TSERVE command line, which defaults to 10000 seconds. Note that ADEV results displayed in TimeLab may not necessarily match those reported by the **show adev** command at the time the file is saved. The .TIM file contains up to */timerec* seconds of incoming phase data beginning at the last **start** command, while TSERVE continues to update its existing ADEV bin contents incrementally when new data points beyond the */timerec* record size cause the phase record to scroll.

selftest

Executes a built-in self test (BIST) routine on the PhaseStation. As shown below (Figure 78), the **selftest** command is available only when no measurement is currently in progress.



```
=192.168.1.200 > selftest
Cannot execute self-test while acquisition is running

=192.168.1.200 > stop
=192.168.1.200 > selftest
0x00000000
```

Figure 78

selftest returns a 32-bit hexadecimal number whose individual bits represent various fault indications. A result of 0x00000000 indicates that no issues were found. Nonzero bits are interpreted as follows:

```
const U32 BIST_OK                 = 0x00000000;  // All bits clear = no faults detected
const U32 BIST_POCXO_COLD         = 0x00000001;  // Clock OCXO not warmed up (< 68C)
const U32 BIST_POCXO_HOT          = 0x00000002;  // Clock OCXO over max temperature (75C)
const U32 BIST_POCXO_UNLOCKED     = 0x00000004;  // ADC clock PLL unlocked
const U32 BIST_FPGA_HOT           = 0x00000008;  // FPGA over max temperature (100C)
const U32 BIST_FPGA_COLD          = 0x00000010;  // FPGA under min temperature (0C)
const U32 BIST_PREREG_OVERVOLT    = 0x00000020;  // Controller input voltage > 13.9V
const U32 BIST_PREREG_UNDERVOLT   = 0x00000040;  // Controller input voltage < 13.2V
const U32 BIST_POWER_HIGH         = 0x00000080;  // Controller input power > 15W
const U32 BIST_POWER_LOW          = 0x00000100;  // Controller input power < 11W
const U32 BIST_1P0_OVERVOLT       = 0x00000200;  // 1.0V bus > 1.05V
const U32 BIST_1P0_UNDERVOLT      = 0x00000400;  // 1.0V bus < 0.95V
const U32 BIST_1P8_OVERVOLT       = 0x00000800;  // 1.8V bus > 1.89V
const U32 BIST_1P8_UNDERVOLT      = 0x00001000;  // 1.8V bus < 1.71V
const U32 BIST_PSRAM_FAULT        = 0x00002000;  // PSRAM FIFO could not be enabled
const U32 BIST_FPGA_UNCONFIGURED  = 0x00004000;  // FPGA could not be configured
const U32 BIST_BULK_READ_ERROR    = 0x00008000;  // Couldn't stream input data
const U32 BIST_CH0_PRN_FAULT      = 0x00010000;  // Invalid data read from ADC channel 0 (J3)
const U32 BIST_CH1_PRN_FAULT      = 0x00020000;  // Invalid data read from ADC channel 1 (J1)
const U32 BIST_CH2_PRN_FAULT      = 0x00040000;  // Invalid data read from ADC channel 2 (J4)
const U32 BIST_CH3_PRN_FAULT      = 0x00080000;  // Invalid data read from ADC channel 3 (J2)
const U32 BIST_MB_NOT_FOUND       = 0x04000000;  // Main board disconnected
const U32 BIST_FPAN_NOT_FOUND     = 0x08000000;  // Front panel disconnected
const U32 BIST_AUTH_FAILURE       = 0x10000000;  // Authentication failure
const U32 BIST_DEVICE_IN_USE      = 0x20000000;  // Device is currently in use
const U32 BIST_USB_ERROR          = 0x40000000;  // Non-fatal USB communication error(s) occurred
const U32 BIST_CONNECT_FAILURE    = 0x80000000;  // Failed to connect to device
```

It is normal for bit 1 to be set (0x00000001) if the **selftest** command is issued immediately after power-up while the front-panel STATUS indicator is still blue.

set <. . .>

When the Telnet server thread is created by TSERVE at startup time, copies of the default values for the **phaserate**, **tau0**, **referencefreq**, **prompt**, **dateformat**, **timeformat**, and **title** properties are associated with the server, taking into account any changes made to the program defaults at the TSERVE command line. These properties subsequently apply to any measurements initiated by either Telnet clients or by a */start* parameter on the TSERVE command line. Clients may alter these server property values by issuing the appropriate **set** commands. Changes made to the server property values will apply to acquisitions that are subsequently initiated by that client or any others.

The following **set** commands are supported by TSERVE:

- **set dateformat <1|2|3>**

  Sets the date format used by the **show date** command.

  1 dd (abbreviated name of month) yyyy
  2 mm/dd/yyyy
  3 dd/mm/yyyy

  The default date format is 1, unless overridden by the */dateformat* command-line parameter.

- **set phaserate <rate>**

  See the description of the */phaserate* command-line parameter for more information.

- **set referencefreq <auto|MHz>**

  See the description of the */referencefreq* command-line parameter for more information.

- **set inputfreq <auto|MHz>**

  See the description of the */inputfreq* command-line parameter for more information.

- **set tau0 <value>**

  Sets the tau-zero period for the **show adev** command in seconds. Values of 0.001, 0.01, 0.1, and 1 second are supported.

  Note that changing the **tau0** value will recompute the **phaserate** property, and vice versa. See the description of the */phaserate* command-line parameter for more information. As with **phaserate**, changes to **tau0** are scaled by the */phasedec* parameter if a non-default value is provided on the command line.

- **set timeformat <24|12>**

  Sets the time format used by the **show date** command. The default time format is 24, unless overridden by the */timeformat* command-line parameter.

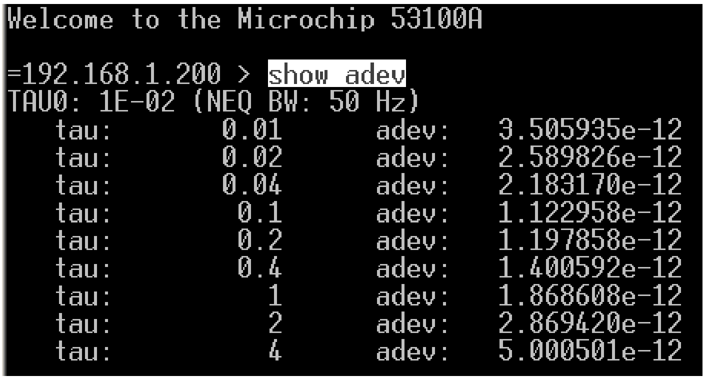
- **set title <title>**
  Establishes the title text returned by the **show title** command. Unless changed by the */title* command-line parameter, the default title is the reserved string **(Serial #).** When **show title** is issued during a measurement, the reserved string will be replaced by the serial number of the connected instrument. Other title strings will be reported verbatim.

  Titles containing spaces must be enclosed in double quotes (" "). See the **show title** command description below for more information.

show <. . .>

The **show** command queries the current value of a property or the results of the current measurement in progress. The following **show** queries are supported by TSERVE:

- **show adev**
  Returns the current Allan deviation chart, as shown in the example below (Figure 79):



```
Welcome to the Microchip 53100A

=192.168.1.200 > show adev
TAU0: 1E-02 (NEQ BW: 50 Hz)
   tau:        0.01      adev:   3.505935e-12
   tau:        0.02      adev:   2.589826e-12
   tau:        0.04      adev:   2.183170e-12
   tau:         0.1      adev:   1.122958e-12
   tau:         0.2      adev:   1.197858e-12
   tau:         0.4      adev:   1.400592e-12
   tau:           1      adev:   1.868608e-12
   tau:           2      adev:   2.869420e-12
   tau:           4      adev:   5.000501e-12
```

Figure 79

ADEV calculations performed by TSERVE are based on a 1-2-4 distribution of tau bins in each decade, using a minimum threshold of 4 samples per bin. Assuming that the */phasedec* parameter is set to its default value of 2, results will be returned for tau intervals ranging from the minimum t_0 value given by the reciprocal of the */phaserate* parameter (100 seconds by default, yielding t_0=0.01 seconds) to a maximum of 1/5 of the */timerec* parameter (10000 seconds by default, yielding t_{max}=2000 seconds).

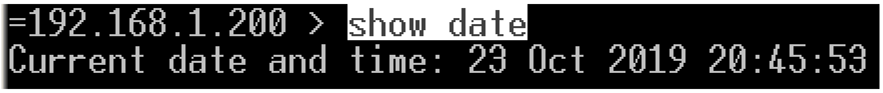
See the */timerec*, */phasedec*, and */phaserate* parameters for more information.

- **show amspectrum**
- **show amspurs**
  These commands function identically to **show spectrum** and **show spurs**, but for the AM noise rather than phase noise. See **show spectrum** and **show spurs** for more information.

If */am:false* was specified on the TSERVE command line, these commands will return an error message indicating that no AM measurement data is available.

- **show date**

  Returns the current date/time information as reported by the TSERVE host OS, as shown in the example below (Figure 80):



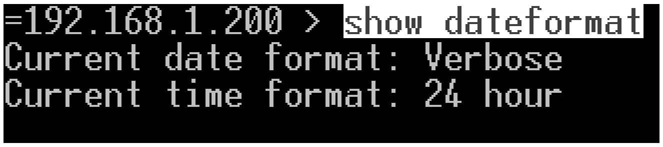
```
=192.168.1.200 > show date
Current date and time: 23 Oct 2019 20:45:53
```

Figure 80

The date and time formats are determined by the **dateformat** and **timeformat** properties, respectively.

- **show dateformat**
- **show timeformat**

  Both of these commands return the date and time formats, as established by the **dateformat** and **timeformat** properties (Figure 81):

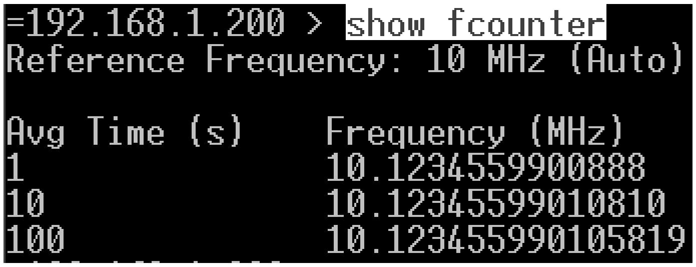


```
=192.168.1.200 > show dateformat
Current date format: Verbose
Current time format: 24 hour
```

Figure 81

- **show fcounter**

  Returns the current reference frequency and frequency count chart at all available averaging times up to 1000 seconds (Figure 82):



```
=192.168.1.200 > show fcounter
Reference Frequency: 10 MHz (Auto)

Avg Time (s)    Frequency (MHz)
1               10.1234559900888
10              10.12345599010810
100             10.123455990105819
```

Figure 82

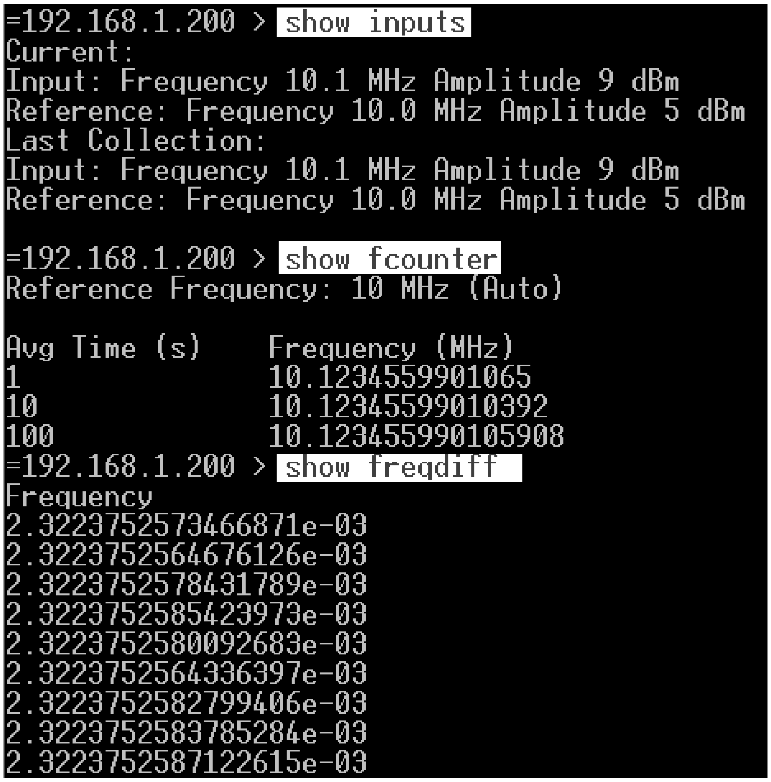
The display precision is 13 digits to the right of the decimal point for 1-second averaging times at all DUT input frequencies. At DUT input frequencies of 5 MHz and above, 15 digits are displayed for averaging times longer than 10 seconds. 14 digits are displayed under all other conditions.

As with the TSC 51XXA phase noise test sets, accurate frequency counts with reference frequencies other than multiples of 100 kHz require the actual reference frequency to be specified. This can be done by a Telnet client with the **set referencefreq** command, or by using the */referencefreq* command-line option on the server.

Unlike the 51XXA instruments, **set referencefreq** must be issued before the measurement begins in order to provide correct frequency counts. Any changes made during the measurement will not take effect until the next measurement begins.

- **show freqdiff**
- **show freq**

Returns the current frequency difference chart (Figure 83). Both forms (**freqdiff** and **freq**) are accepted.



```
=192.168.1.200 > show inputs
Current:
Input: Frequency 10.1 MHz Amplitude 9 dBm
Reference: Frequency 10.0 MHz Amplitude 5 dBm
Last Collection:
Input: Frequency 10.1 MHz Amplitude 9 dBm
Reference: Frequency 10.0 MHz Amplitude 5 dBm

=192.168.1.200 > show fcounter
Reference Frequency: 10 MHz (Auto)

Avg Time (s)    Frequency (MHz)
1               10.1234559901065
10              10.12345599010392
100             10.123455990105908
=192.168.1.200 > show freqdiff
Frequency
2.3223752573466871e-03
2.3223752564676126e-03
2.3223752578431789e-03
2.3223752585423973e-03
2.3223752580092683e-03
2.3223752564336397e-03
2.3223752582799406e-03
2.3223752583785284e-03
2.3223752587122615e-03
```

Figure 83

The values reported by this command comprise a historical “strip chart” record of the frequency of the signal at the DUT input jack, assuming that the reference frequency is either a multiple of 100 kHz or has been set correctly with either **set referencefreq** or */referencefreq.* These DUT frequency readings are returned in the form of fractional frequency differences relative to the rounded input frequency reported by **show inputs**. Entries for the past *n-1* seconds are displayed, ordered by age (oldest to newest), where *n* defaults to 600 seconds. The value of *n* can be configured through the use of the */chart* command-line parameter.

In the example above, the DUT signal was generated by a DDS synthesizer tuned to a nominal frequency of 10.123456 MHz. This value will be reported by the **show inputs** query as 10.1 MHz unless */inputfreq*, */roundfreq*, or **set inputfreq** has been used to establish a more precise nominal DUT input frequency. The readings from **show fcounter** are more accurate as well as more precise than **show inputs**, displaying consistent error that reveals the synthesizer's limited tuning precision. Multiplying 10.1 MHz by one of the reported frequency difference readings near 2.322…E-3 and adding the result back to 10.1 MHz yields a similarly-precise absolute frequency count.

- **show inputs**

  Returns amplitude readings and rounded or specified frequency readings for the signals at the 53100A's REFERENCE and DUT input ports, as shown in Figure 83 above.

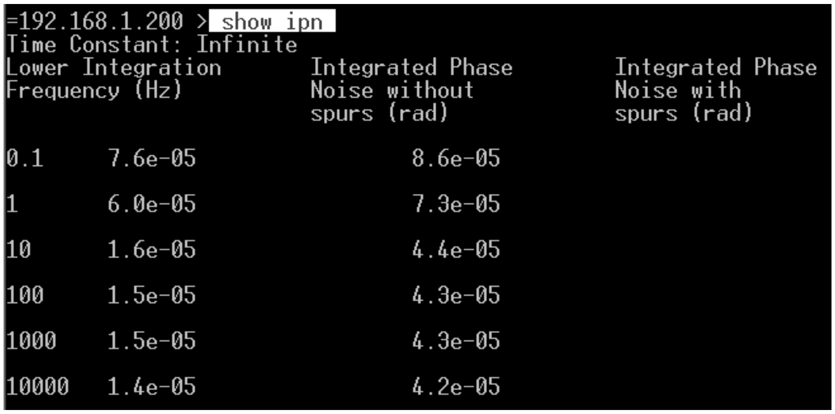
  Unlike the TSC 51XXA implementation of this command, both the 'Current' and 'Last Collection' values reflect the state of the measurement in progress. If no measurement is running at the time **show inputs** is issued, the message *No measurement in progress* will be returned.

- **show inputfreq**

  Returns the nominal DUT input frequency used for phase-difference calculations and the **show inputs** command. See the description of the */inputfreq* command-line parameter for more information.

- **show ipn**

  Returns a chart showing integrated phase noise in units of radians over various portions of the Fourier spectrum, with and without the influence of coherent spurs.



```
=192.168.1.200 > show ipn
Time Constant: Infinite
Lower Integration       Integrated Phase        Integrated Phase
Frequency (Hz)          Noise without           Noise with
                        spurs (rad)             spurs (rad)

0.1     7.6e-05                 8.6e-05

1       6.0e-05                 7.3e-05

10      1.6e-05                 4.4e-05

100     1.5e-05                 4.3e-05

1000    1.5e-05                 4.3e-05

10000   1.4e-05                 4.2e-05
```

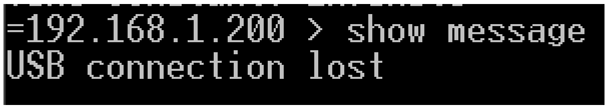
Figure 84

Each row in the chart reports the jitter at limits of integration between the given offset shown at left and the maximum offset supported by the PhaseStation hardware, normally 1 MHz. Lower integration limits from 0.01 Hz to 10000 Hz are supported. In the example shown above (Figure 84), the measurement has been running long enough to provide data at a minimum lower integration limit of 0.1 Hz.

Columns are tab-separated for compatibility with TSC 51XXA applications.

- **show message**

  Returns the most recent error or warning message from the PhaseStation hardware driver (Figure 85).

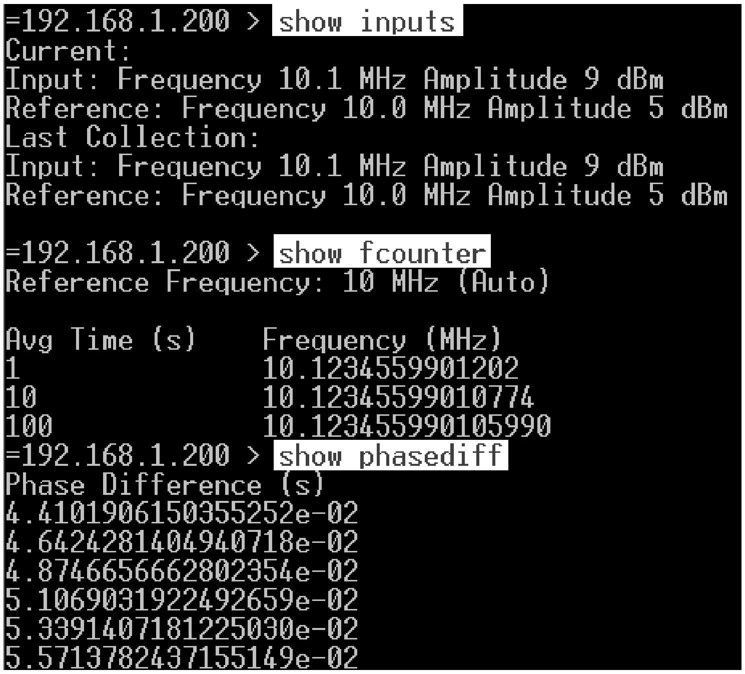


```
=192.168.1.200 > show message
USB connection lost
```

Figure 85

- **show phasediff**

  Returns the current phase difference chart (Figure 86).



```
=192.168.1.200 > show inputs
Current:
Input: Frequency 10.1 MHz Amplitude 9 dBm
Reference: Frequency 10.0 MHz Amplitude 5 dBm
Last Collection:
Input: Frequency 10.1 MHz Amplitude 9 dBm
Reference: Frequency 10.0 MHz Amplitude 5 dBm

=192.168.1.200 > show fcounter
Reference Frequency: 10 MHz (Auto)

Avg Time (s)    Frequency (MHz)
1               10.1234559901202
10              10.12345599010774
100             10.123455990105990
=192.168.1.200 > show phasediff
Phase Difference (s)
4.4101906150355252e-02
4.6424281404940718e-02
4.8746656662802354e-02
5.1069031922492659e-02
5.3391407181225030e-02
5.5713782437155149e-02
```

Figure 86

The values reported by this command comprise a historical “strip chart” record of the difference in phase between the nominal DUT input frequency reported by **show inputs** and the actual frequency at the DUT input jack, assuming that the reference frequency is either a multiple of 100 kHz or has been set correctly with either **set referencefreq** or */referencefreq.* Entries for the past *n* seconds are displayed, ordered by age (oldest to newest), where *n* defaults to 600 seconds. The value of *n* can be configured through the use of the */chart* command-line parameter.

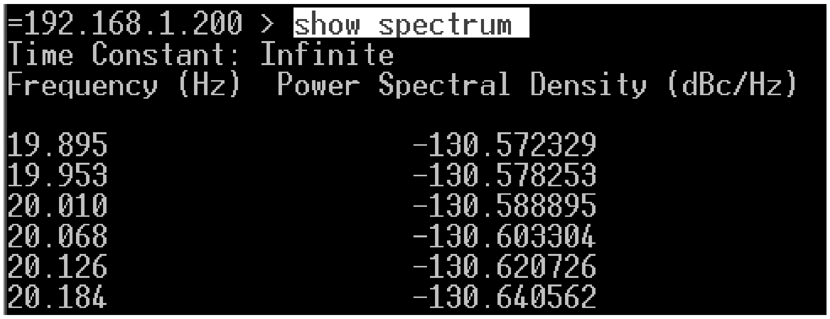
The phase difference chart's overall slope is a representation of the fractional difference between the nominal and actual frequencies. In the example above, as in the earlier **show freqdiff** example, the reported DUT input frequency is 10.1 MHz while the actual frequency from the test source is just under 10.123456 MHz. The readings above advance by about 2.322 milliseconds per second, corresponding to the fractional frequency differences returned by the **show freqdiff** command. Continuing the example, if the nominal input frequency had been specified with **set inputfreq 10.123456**, */inputfreq:10.123456*, or */roundfreq:0.000001*, the slope of the phase differences would be almost zero.

See the **measurelinear** and **removelinear** commands for information on calculating and removing the linear trend from the phase-difference chart.

- **show phaserate**
  Returns the phase data rate set by the */phaserate* command-line parameter or the most recent **set phaserate** command. See the description of the */phaserate* command-line parameter for more information.

- **show referencefreq**
  Returns the reference frequency used for frequency counter and phase/frequency difference calculations. See the description of the */referencefreq* command-line parameter for more information.

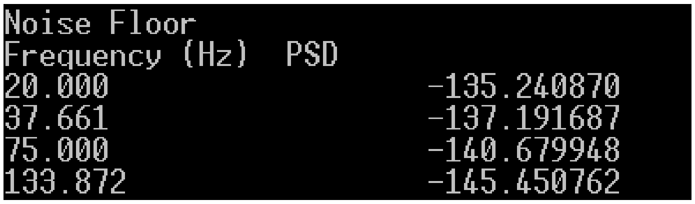
- **show spectrum**
  Returns a list of offset frequencies and dBc/Hz values corresponding to the phase noise spectrum in the measurement in progress (Figure 87):



```
=192.168.1.200 > show spectrum
Time Constant: Infinite
Frequency (Hz)  Power Spectral Density (dBc/Hz)

19.895                    -130.572329
19.953                    -130.578253
20.010                    -130.588895
20.068                    -130.603304
20.126                    -130.620726
20.184                    -130.640562
```

Figure 87

Following the PSD data, the current instrument noise floor estimate will be returned (Figure 88):



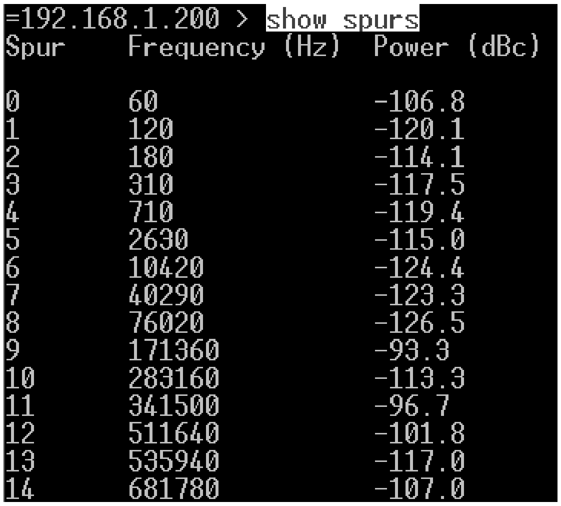
```
Noise Floor
Frequency (Hz)  PSD
20.000                  -135.240870
37.661                  -137.191687
75.000                  -140.679948
133.872                 -145.450762
```

Figure 88

If no measurement is currently in progress, the **show spectrum** query will have no effect.

- **show spurs**

  Returns a list of identified coherent spurs in the phase noise spectrum (Figure 89).



```
=192.168.1.200 > show spurs
Spur    Frequency (Hz)  Power (dBc)

0       60              -106.8
1       120             -120.1
2       180             -114.1
3       310             -117.5
4       710             -119.4
5       2630            -115.0
6       10420           -124.4
7       40290           -123.3
8       76020           -126.5
9       171360          -93.3
10      283160          -113.3
11      341500          -96.7
12      511640          -101.8
13      535940          -117.0
14      681780          -107.0
```

Figure 89

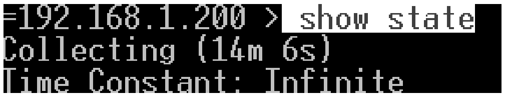
All identified spurs are removed from the PSD plot returned by **show spectrum**. Spurs that are not classified as instrument artifacts will appear in the table returned by **show spurs** at their correct (i.e., non-normalized) amplitudes in dBc.

See the descriptions of the */spur_Hz* and */spur_dB* command-line parameters for more information.

- **show state**

  Returns the current measurement status and elapsed time (Figure 90).

  Ready No measurement has been started
  Initializing The measurement has not yet begun to return data
  Collecting The measurement is currently in progress



```
=192.168.1.200 > show state
Collecting (14m 6s)
Time Constant: Infinite
```

Figure 90

Since TSERVE does not support the **timeconstant** property from the TSC 51XXA command language, **show state** will always report **Time Constant: Infinite**.

- **show tau0**

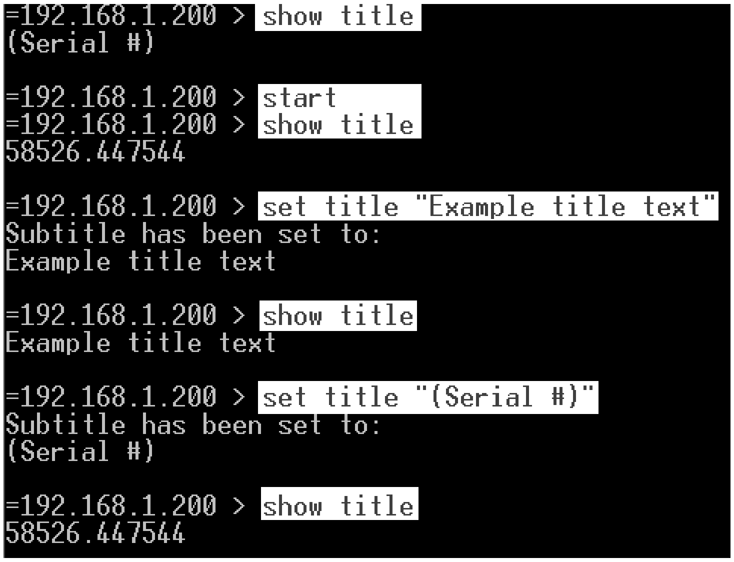
  Returns the current tau0 property value.

- **show timeformat**

  See **show dateformat**.

- **show title**

  Returns the title most recently set by the **set title** command or the */title* command-line option. As shown in Figure 91 below, the **show title** query response depends on the measurement state as well as the title text itself.



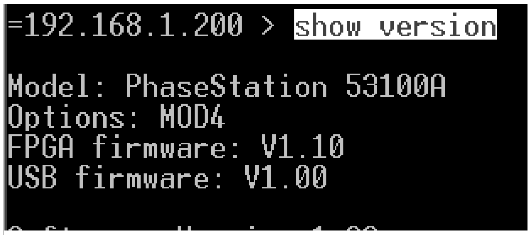
```
=192.168.1.200 > show title
(Serial #)

=192.168.1.200 > start
=192.168.1.200 > show title
58526.447544

=192.168.1.200 > set title "Example title text"
Subtitle has been set to:
Example title text

=192.168.1.200 > show title
Example title text

=192.168.1.200 > set title "(Serial #)"
Subtitle has been set to:
(Serial #)

=192.168.1.200 > show title
58526.447544
```

Figure 91

The default title text is the reserved string **(Serial #)**. If a **show title** query is issued during a measurement and the default title has not been changed, the reserved string will be replaced with the serial number of the PhaseStation that is running the measurement.

- **show version**

  Returns software and hardware version information (Figure 92).



```
=192.168.1.200 > show version

Model: PhaseStation 53100A
Options: MOD4
FPGA firmware: V1.10
USB firmware: V1.00
```

Figure 92

shutdown

This command can be used to request immediate shutdown of the TSERVE server process, disconnecting all connected clients. To disconnect the current client without exiting TSERVE, use the **quit** command instead.

start

stop

These commands can be used by any connected client to initiate a new measurement or terminate one that is already in progress.

In many applications, it may be more convenient to start a measurement at the time TSERVE is launched and allow it to run until the server itself is stopped. This can be accomplished with the */start* command-line parameter. Other command-line parameters can be used in lieu of the **set <property>** commands when necessary.

The above approach is analogous to pressing the START button on a physical test set. Streaming clients can connect to the data port and acquire data from the PhaseStation 53100A without the need for any Telnet clients or other terminal applications to connect to the command port.

Compatibility notes

- The following 51XXA commands and queries are not supported by TSERVE:

  button <number> ***calinputs*** ***control <take | yield>***
  print [<options>] ***set print <options>*** ***set timeconstant <value>***
  show mac ***show printformats*** ***show printoptions*** ***show screens***

- As discussed elsewhere, commands such as **set referencefreq** that affect both current and subsequent measurements on the Symmetricom/Microsemi 51XXA phase noise test sets will affect only subsequent measurements in TSERVE.

- The **show inputs** command returns both "Current" and "Last Collection" values on the 51XXA test sets. These values are not correctly maintained by the 51XXA test set firmware under certain conditions, such as when the previous input signals are disconnected and different ones reconnected. TSERVE does not attempt to emulate the 51XXA's behavior in this regard. Instead, both the "Current" and "Last Collection" values reflect the rounded frequency and amplitude estimates for the measurement currently in progress. If no measurement is currently being performed, the message *No measurement in progress* is returned to the client.

- The streaming phase-difference values returned by TSERVE are based on the rounded or specified input frequency reported by the **show inputs** query. This is also true of the **show phasediff** and **show freqdiff** chart entries. On the 51XXA test sets, however, these values are referred to a more precise internal frequency measurement that is not made available to remote-access clients. Consequently, only the **show fcounter** command can be used to obtain accurate frequency counts on the 51XXA test sets, while TimeLab's own frequency count chart in the **Measurement→Frequency difference (f)** view can be used with phase data from TSERVE if the correct rounded input frequency is entered in the acquisition dialog.

- On the 51XXA test sets, the **show adev** command returns separate Allan deviation charts for all supported t_0 intervals. TSERVE returns data only for the t_0 interval corresponding to the most recent **set phaserate** or **set tau0** commands.

- The default **phaserate** for the 51XXA test sets is 1000 samples per second. TSERVE's default **phaserate** is 100 samples per second, unless changed with the */phaserate* command-line option.

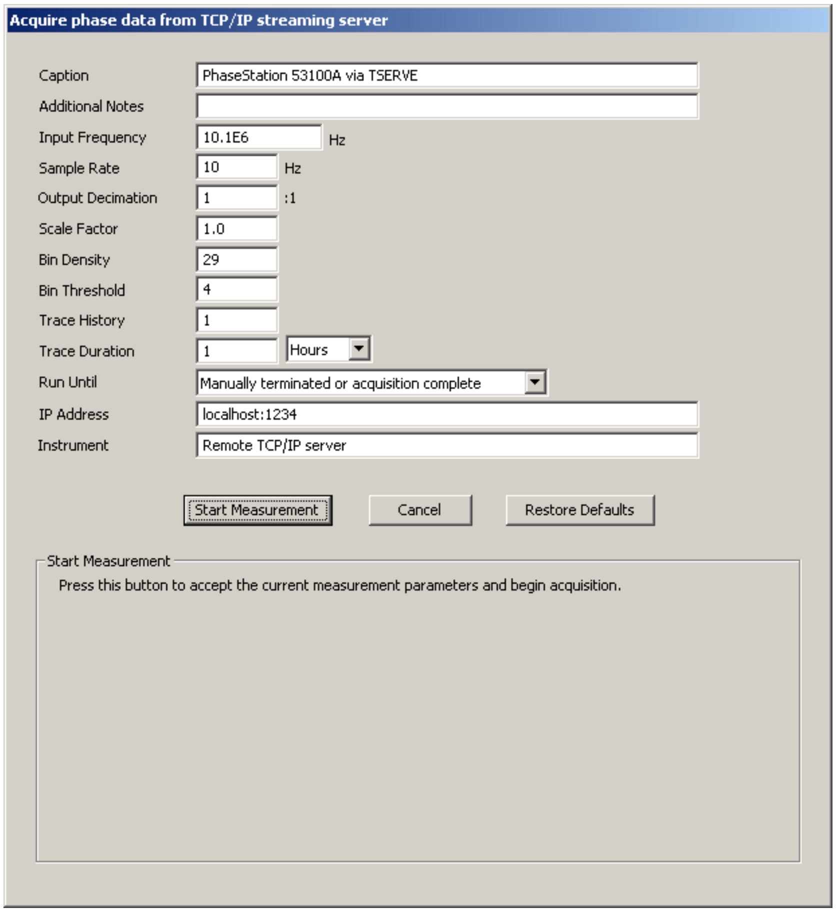
- The 51XXA test sets transmit measurement-related error messages to the Telnet client asynchronously, sometimes waiting until the next character arrives from the remote client before sending the message. As a rule, TSERVE transmits text on the Telnet command port only in response to specific commands. The **show message** command is provided to allow the remote user or application to determine if the most recent measurement encountered an error or warning condition. An empty string indicates that no errors or warnings have occurred.

Measurement example: Phase data streaming

When using software that was originally written to capture phase data from the Symmetricom / Microsemi 5115A/5120A/5125A and compatible instruments, you will normally want to allow TSERVE to use its default */port* value (1298) and default */format* value (TSC). However, when using TimeLab's **Acquire→Acquire from TCP/IP streaming server** option, only the phase-difference data format (*/format:P*) is supported.

Example: *tserve64 /data_port:1234 /phaserate:100 /phasedec:20 /format:P /start*

As an example, the command above will begin capturing phase-difference data at a rate of 100 samples per second as soon as TSERVE is launched. If desired, Telnet commands may be issued via port 1299 as usual, but this isn't required since the */start* option has been specified on the TSERVE command line.



Acquire phase data from TCP/IP streaming server

Caption	PhaseStation 53100A via TSERVE
Additional Notes	
Input Frequency	10.1E6 Hz
Sample Rate	10 Hz
Output Decimation	1 :1
Scale Factor	1.0
Bin Density	29
Bin Threshold	4
Trace History	1
Trace Duration	1 Hours
Run Until	Manually terminated or acquisition complete
IP Address	localhost:1234
Instrument	Remote TCP/IP server

Start Measurement

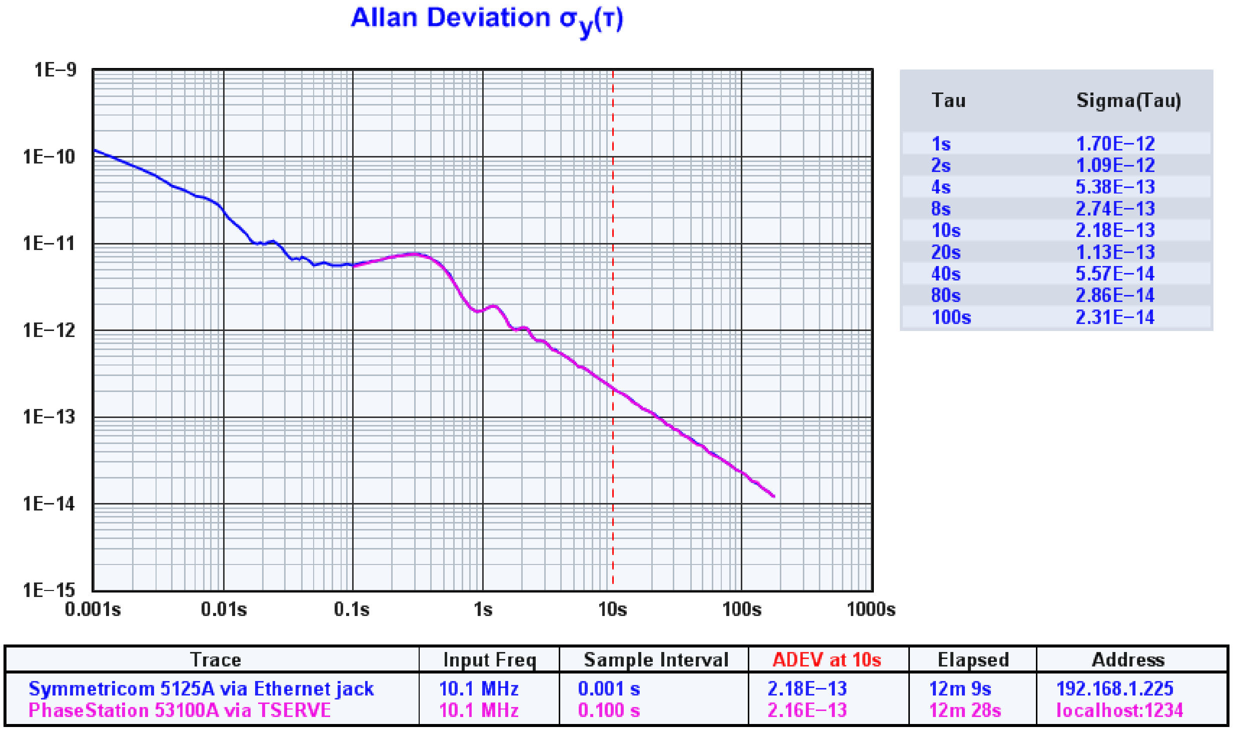
Press this button to accept the current measurement parameters and begin acquisition.

Figure 93

Internally, the data will be captured at 200 samples per second with a noise-equivalent bandwidth of 50 Hz. Phase data would normally be recorded with 2:1 decimation after detection, for a final data rate of 100 samples per second. However, the use of the */phasedec:20* option results in a final data rate of only 10 samples per second. This is the rate at which the phase-difference readings associated with */format:P* will be streamed from port 1234.

You can receive the streaming phase data on any machine that can access the server on your network, including the server PC itself. Either **Acquire→Symmetricom 5115A / 5120A / 5125A (Frequency stability)** or the more general **Acquire→Acquire from TCP/IP streaming server** option can be used to retrieve phase data from TSERVE. In Figure 93 above, the latter has been selected. The Input Frequency field in the acquisition dialog has been populated with the 100-kHz rounded value that would be reported by the **show inputs** command, while Sample Rate is set to the final data rate of 10 samples per second. With an Output Decimation value of 1, no additional decimation will be performed by TimeLab.

TimeLab can receive data concurrently from multiple instruments. When the PhaseStation 53100A measurement described above was run alongside a Symmetricom 5125A test set connected to the same signal sources, the result shown in Figure 94 was obtained:



Tau	Sigma(Tau)
1s	1.70E−12
2s	1.09E−12
4s	5.38E−13
8s	2.74E−13
10s	2.18E−13
20s	1.13E−13
40s	5.57E−14
80s	2.86E−14
100s	2.31E−14

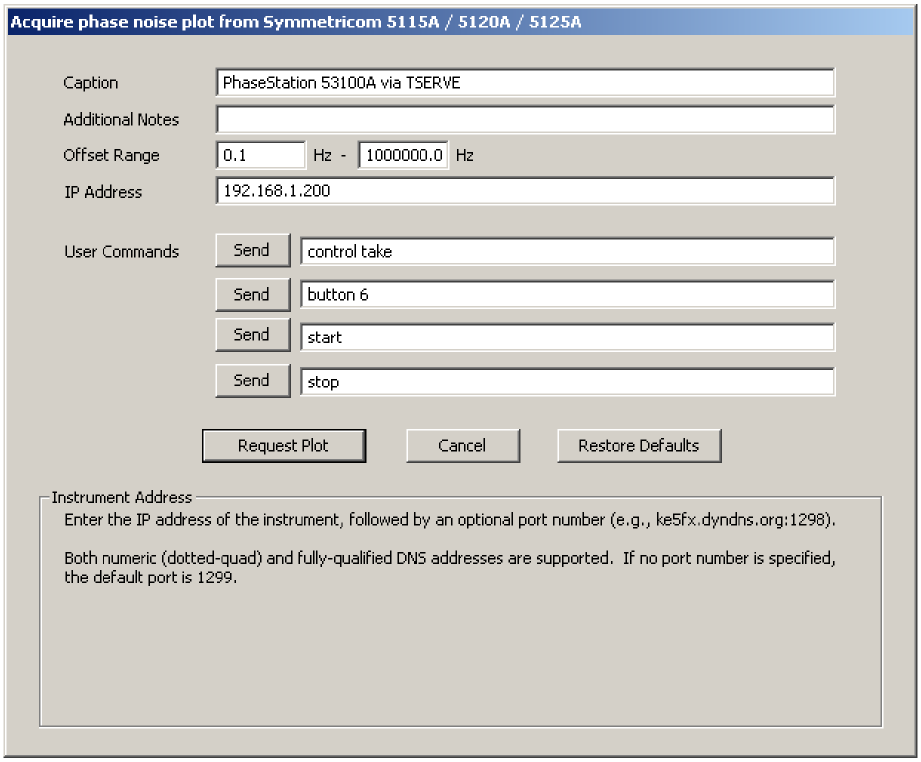
Trace	Input Freq	Sample Interval	ADEV at 10s	Elapsed	Address
Symmetricom 5125A via Ethernet jack	10.1 MHz	0.001 s	2.18E−13	12m 9s	192.168.1.225
PhaseStation 53100A via TSERVE	10.1 MHz	0.100 s	2.16E−13	12m 28s	localhost:1234

Figure 94

For this exercise, the Symmetricom 5125A's **phaserate** property was left at its default value of 1000 samples per second. Consequently, its Allan deviation trace begins at t_0=0.001 second, while the PhaseStation measurement begins at t_0=0.1 second. The results are otherwise virtually identical.

Measurement example: Phase noise plot acquisition

TimeLab can also retrieve and display phase noise and AM noise data from the PhaseStation 53100A through the use of the **Acquire→Symmetricom 5115A / 5120A / 5125A (Phase noise)** acquisition option. Unlike the streaming phase data acquisition process, obtaining noise data from either TSERVE or an actual 51XXA test set requires the program to connect to the server's Telnet command port.



Acquire phase noise plot from Symmetricom 5115A / 5120A / 5125A

Caption: PhaseStation 53100A via TSERVE

Additional Notes:

Offset Range: 0.1 Hz - 1000000.0 Hz

IP Address: 192.168.1.200

User Commands:
control take
button 6
start
stop

Instrument Address

Enter the IP address of the instrument, followed by an optional port number (e.g., ke5fx.dyndns.org:1298).

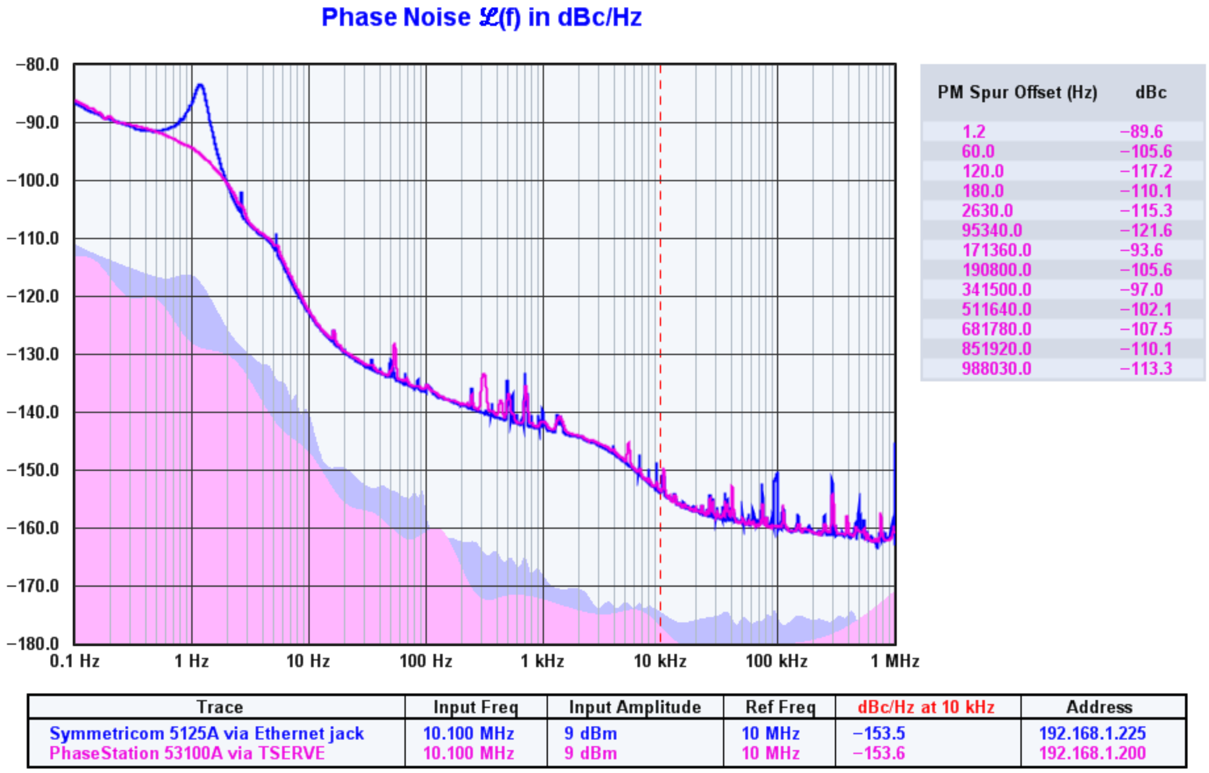
Both numeric (dotted-quad) and fully-qualified DNS addresses are supported. If no port number is specified, the default port is 1299.

Figure 95

Because TimeLab can obtain most of the required measurement parameters and properties directly from the server, relatively few configuration steps are needed in the 51XXA phase noise acquisition dialog. It's necessary only to ensure that the correct IP address and optional port number are specified, in this case either **192.168.1.200** as used above in Figure 95, or the equivalent **localhost** alias used earlier for phase data capture. No port number is required as long as the default command port assignment (1299) has not been changed with the */cmd_port* command-line parameter.

For convenience, the phase noise acquisition dialog offers four general-purpose command buttons that are predefined with useful defaults that allow front-panel control commands to be transmitted to an actual TSC 51XXA phase noise test set. (Only the **start** and **stop** buttons are supported by TSERVE.)

Figure 96 shows the phase noise plots that were simultaneously measured during the one-hour streaming data acquisition in the previous example.



PM Spur Offset (Hz)	dBc
1.2	−89.6
60.0	−105.6
120.0	−117.2
180.0	−110.1
2630.0	−115.3
95340.0	−121.6
171360.0	−93.6
190800.0	−105.6
341500.0	−97.0
511640.0	−102.1
681780.0	−107.5
851920.0	−110.1
988030.0	−113.3

Trace	Input Freq	Input Amplitude	Ref Freq	dBc/Hz at 10 kHz	Address
Symmetricom 5125A via Ethernet jack	10.100 MHz	9 dBm	10 MHz	−153.5	192.168.1.225
PhaseStation 53100A via TSERVE	10.100 MHz	9 dBm	10 MHz	−153.6	192.168.1.200

Figure 96

Once again, the results match closely. The most notable difference is that the PhaseStation has classified the sharp peak near 1.2 Hz as a spur, removing it from the plot and adding it to the spur chart at right. Adding */spur_Hz=2* to the TSERVE command line would have resulted in an even-closer match by forcing the PhaseStation's spur-detection routine to disregard the peak.

Appendix: 53100A Dual 100 MHz Reference (Option IR)

Option IR adds two independent 100 MHz reference outputs to the Microchip 53100A. These reference sources support fully self-contained noise and stability measurements at a higher level of performance than virtually any single-channel references can provide, and can also serve as spectrally-pure 100 MHz signal sources in other applications.   The reference outputs are directly accessible at the front-panel INT REF 1 and INT REF 2 jacks.

Front and rear panel layouts

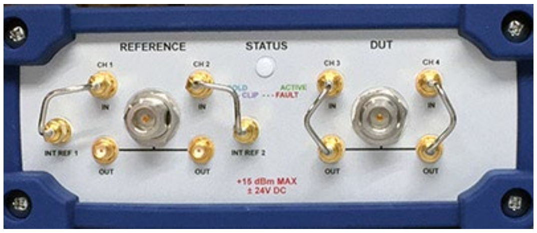


Figure 97

Test sets equipped with Option IR are delivered with the channel-1 and channel-2 input jacks strapped to INT REF 1 and INT REF 2 by default, rather than to the built-in reference splitter outputs (Figure 97).   In this configuration, the N-F front-panel REFERENCE input jack may be driven by an external 10 MHz “house clock” to discipline one or both of the internal references, or simply left unconnected for true standalone operation.

The rear panel includes the additional TTL / 5V CMOS level 1 PPS IN jack shown in Figure 98. Like the front-panel REFERENCE jack, an external disciplining source may be connected to the 1 PPS IN jack. More information on selection and connection of external reference sources appears later in this section.

The INT REF 1 and INT REF 2 outputs provide reference signals at 100 MHz as long as power is connected to the 53100A, regardless of whether external 10 MHz or 1PPS sources are present. At the same time, all four of the 53100A’s input channels remain user-accessible in both Option IR and standard units, to support applications that require direct connection of multichannel sources or external references.

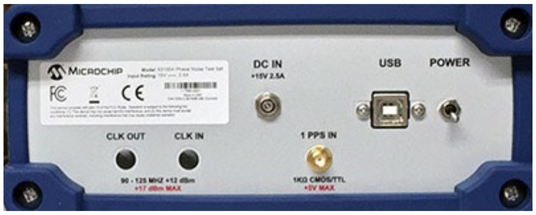


Figure 98

Optional internal atomic standard (STD)

The Option IR reference module may itself be equipped with an optional board-level atomic frequency standard, which can serve as an additional disciplining source in measurement applications that require long-term frequency stability but that do not have access to high quality external standards. Reference frequency stability at intervals longer than approximately 1000 seconds is improved up to 10x in units equipped with the IR/STD option.

Block diagram

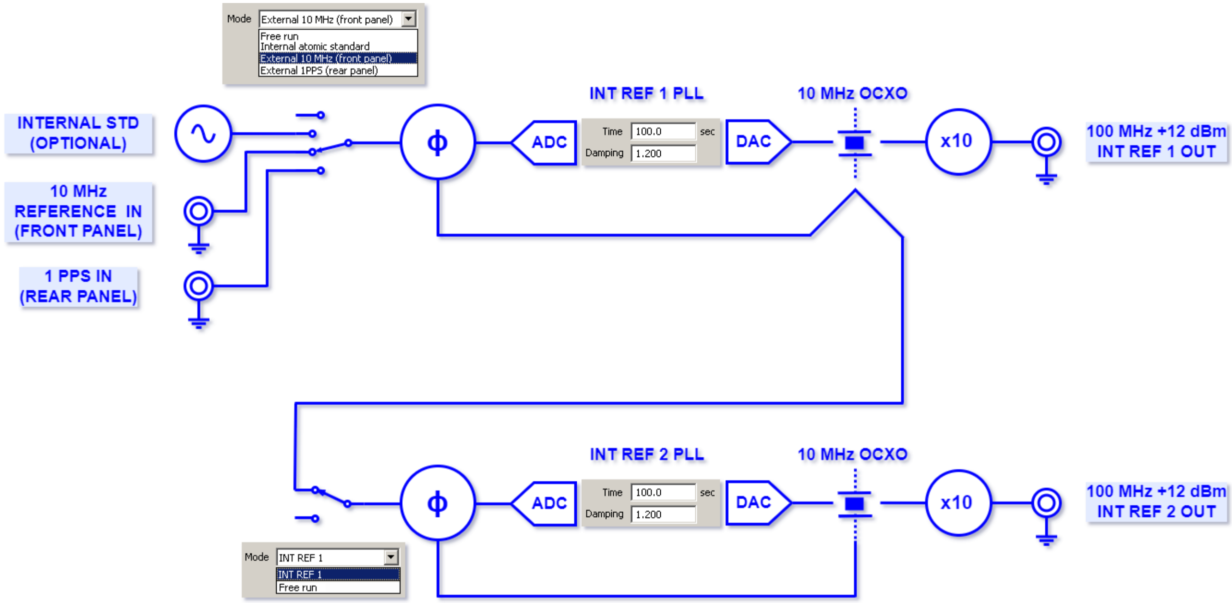


Figure 99

Figure 99 illustrates the basic architecture of the Option IR internal reference module. Each of the two independent channels has its own 10 MHz oven-controlled crystal oscillator (OCXO) which is followed by a low-noise frequency multiplier that generates the 100 MHz output for that channel.

While capable of free-running operation, the 10 MHz OCXO that drives the INT REF 1 output is most often phase-locked to an external or internal 10 MHz standard to maintain frequency accuracy and long term stability, with its PLL acting as a "cleanup" loop for any short-term noise that may be present. Similarly, the INT REF 2 OCXO can either be phase-locked to the INT REF 1 OCXO or allowed to free-run. By default, INT REF 2 is locked to INT REF 1 with a 100-second time constant, allowing the two channels' short-term noise to remain uncorrelated while maintaining overall frequency accuracy at longer intervals.

With this approach, for example, the 53100A can measure the 1-Hz phase noise of ultrastable 5 MHz OCXOs at levels near -130 dBc/Hz even though the 1-Hz phase noise at the INT REF 1 and INT REF 2 outputs is closer to -120 dBc/Hz. At the same time, Allan deviation measurements near 2E-13 @ t=1s are practical. In areas of the graph where the INT REF 1 and INT REF 2 sources are uncorrelated, the real-world measurement floor for a 53100A equipped with Option IR is often determined more by acquisition time than by the reference sources' own specifications.

Why 100 MHz?

Compared to traditional 5 MHz or 10 MHz references, the use of 100 MHz reference sources offers substantial reductions in PN measurement time when working with high-quality VHF crystal oscillators. Especially at offsets above 1 kHz, lower-frequency references require many more FFT averages to achieve the necessary noise floor improvement. Unfortunately, even the best 100 MHz OCXOs lack the stability to make good reference sources for measurements of high-quality HF oscillators.

Option IR is designed to overcome both limitations. As noted above, the two INT REF output signals originate with independent 10 MHz oven-controlled crystal oscillators (OCXOs) with excellent short-term stability and phase noise characteristics. These oscillators are followed by 100 MHz analog PLLs that are based on JFET frequency multipliers rather than traditional dividers. As a result, the OCXOs' superior phase stability is maintained at offsets below 1 kHz without compromising the broadband noise floor. Integrated jitter at the 100 MHz outputs is typically better than 40 fs in the critical 1 Hz - 1 kHz range and below 10 fs from 1 kHz to 1 MHz.

Software support

The TimeLab acquisition dialog for the Microchip 53100A includes a separate page of settings and controls for units equipped with Option IR. This page is accessed by pressing the *Internal Reference* button seen at right (Figure 100) after selecting **Acquire→Microchip 53100A** in the TimeLab main menu.

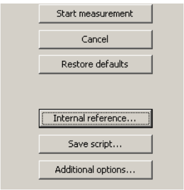
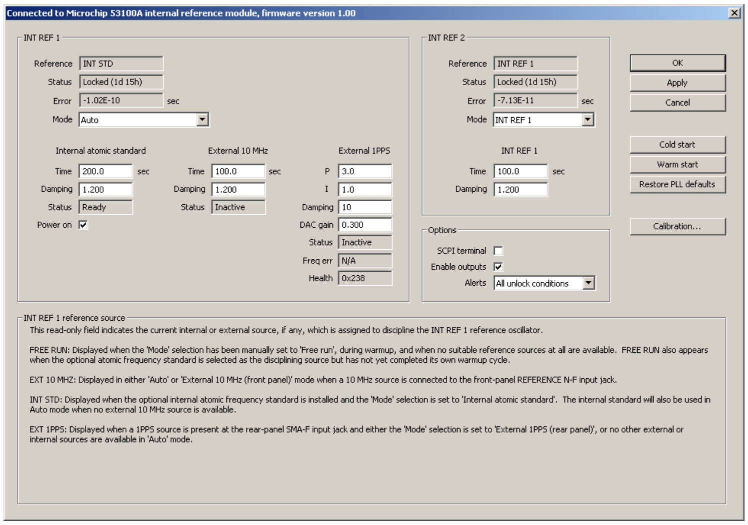


Figure 100

As shown in Figure 101 below, this dialog page includes controls to adjust PLL properties, configure lock source preferences, and calibrate the two OCXOs and the optional internal atomic standard.



Connected to Microchip 53100A internal reference module, firmware version 1.00

INT REF 1

Reference: INT STD
Status: Locked (1d 15h)
Error: -1.02E-10 sec
Mode: Auto

Internal atomic standard
Time: 200.0 sec
Damping: 1.200
Status: Ready
Power on ☑

External 10 MHz
Time: 100.0 sec
Damping: 1.200
Status: Inactive

External 1PPS
P: 3.0
I: 1.0
Damping: 10
DAC gain: 0.300
Status: Inactive
Freq err: N/A
Health: 0x238

INT REF 2

Reference: INT REF 1
Status: Locked (1d 15h)
Error: -7.13E-11 sec
Mode: INT REF 1

INT REF 1
Time: 100.0 sec
Damping: 1.200

Options
SCPI terminal ☐
Enable outputs ☑
Alerts: All unlock conditions

INT REF 1 reference source

This read-only field indicates the current internal or external source, if any, which is assigned to discipline the INT REF 1 reference oscillator.

FREE RUN: Displayed when the 'Mode' selection has been manually set to 'Free run', during warmup, and when no suitable reference sources at all are available. FREE RUN also appears when the optional atomic frequency standard is selected as the disciplining source but has not yet completed its own warmup cycle.

EXT 10 MHZ: Displayed in either 'Auto' or 'External 10 MHz (front panel)' mode when a 10 MHz source is connected to the front-panel REFERENCE N-F input jack.

INT STD: Displayed when the optional internal atomic frequency standard is installed and the 'Mode' selection is set to 'Internal atomic standard'. The internal standard will also be used in Auto mode when no external 10 MHz source is available.

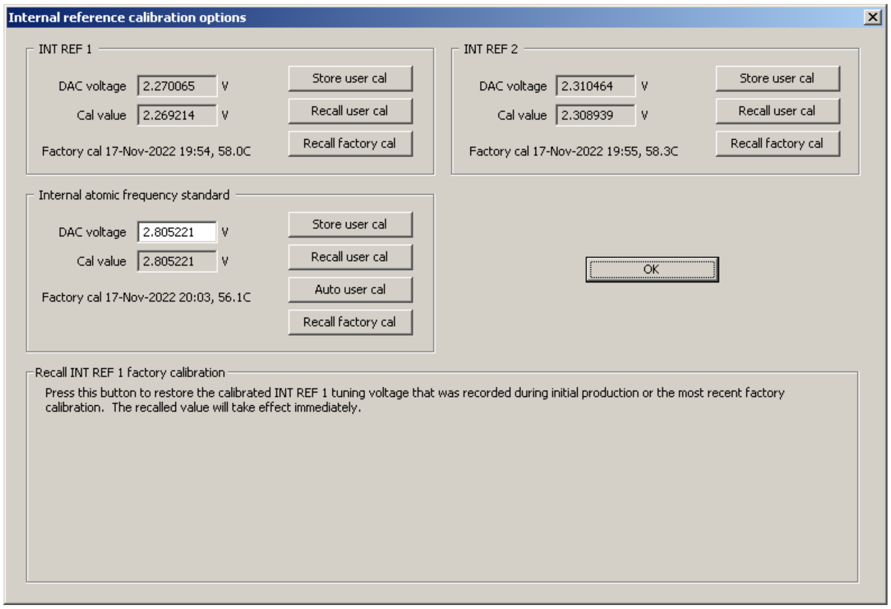
EXT 1PPS: Displayed when a 1PPS source is present at the rear-panel SMA-F input jack and either the 'Mode' selection is set to 'External 1PPS (rear panel)', or no other external or internal sources are available in 'Auto' mode.

Figure 101

The Option IR configuration dialog is divided into two separate sections governing the INT REF 1 and INT 2 references and a third section containing controls common to both. The two INT REF sections are similar, except that INT REF 1 can be locked to the optional internal atomic standard or to an external 10 MHz or 1 PPS source, while INT REF 2 can be disciplined only by INT REF 1. Buttons are also provided at right to reset the internal reference module's onboard CPU (*Cold start*), force immediate relocking in both channels (*Warm start*), and restore the values that determine the PLL properties in both channels to their factory default settings (*Restore PLL defaults*).

User and factory calibration controls

The *Calibration* button opens an additional dialog page containing user calibration options for the internal OCXO sources and optional atomic standard (Figure 102).



Internal reference calibration options

INT REF 1

DAC voltage 2.270065 V

Cal value 2.269214 V

Factory cal 17-Nov-2022 19:54, 58.0C

INT REF 2

DAC voltage 2.310464 V

Cal value 2.308939 V

Factory cal 17-Nov-2022 19:55, 58.3C

Internal atomic frequency standard

DAC voltage 2.805221 V

Cal value 2.805221 V

Factory cal 17-Nov-2022 20:03, 56.1C

Recall INT REF 1 factory calibration

Press this button to restore the calibrated INT REF 1 tuning voltage that was recorded during initial production or the most recent factory calibration. The recalled value will take effect immediately.

Figure 102

This dialog box provides access to the calibration constant parameters that apply when a given OCXO is in free-running operation, unlocked to any internal or external source. These calibration constants represent tuning voltages applied via microvolt-resolution DACs to the OCXO frequency-control inputs during unlocked operation. When an OCXO is in free-running (unlocked) operation, its calibration constant can be edited manually if desired and updated by pressing *Store user cal*. The oscillator's tuning range and sensitivity in hertz per volt appears in the mouseover help text for that channel's *DAC voltage* field. In units equipped with the optional internal atomic frequency standard, a third calibration constant determines its tuning voltage as well.

A key point is that the OCXO calibration constant is not used at all when its corresponding oscillator is locked to an internal or external standard. The OCXO's tuning DAC is driven by its respective software PLL during closed-loop operation, with the DAC voltage displayed as a read-only value. At any time, the tuning DAC voltage for a given channel can be saved to nonvolatile memory with the channel's *Store user cal* button. This will establish a new "user calibration" value for the channel. Assuming the OCXO was locked to a given calibration source at the time *Store user cal* was pressed, its frequency during subsequent free-running operation will be very close to that of the source.

It's also important to note that the optional internal atomic frequency standard is not automatically steered by an external source. (In fact, an external 10 MHz source connected to the front-panel **REFERENCE** jack will take precedence over the internal atomic standard in locking the INT REF 1 OCXO when the channel's *Mode* selection is set to the default *Auto* value.) Instead, the internal standard can be calibrated by connecting an external 10 MHz or 1PPS signal temporarily and pressing the *Auto user cal* button once the INT REF 1 OCXO has successfully phase-locked to the external source. This will start an autocalibration routine that typically takes 3-5 minutes to match the frequency of the external signal to a fractional accuracy of less than 1E-10. If successful, the resulting DAC voltage will be stored as the internal atomic standard's user calibration constant. The external standard may then be disconnected if desired.

Finally, the most recent factory calibration constant for INT REF 1, INT REF 2, or the internal atomic standard may be recalled at any time by pressing the appropriate *Recall factory cal* button. The factory calibration constant will replace the user calibration constant (if any) for that channel. It will remain in effect until a new user calibration constant is stored with *Store user cal* or a previously-stored one is recalled with *Recall user cal*. The factory calibration constants are always available for selection, and cannot be overwritten in normal operation.

User calibration instructions

1. Ensure that power has been connected for at least 2 hours prior to calibration, and that no measurement is currently in progress

2. Connect a calibrated frequency source, such as the Microchip 5071B primary frequency standard, to the front-panel REFERENCE jack

3. Click **Acquire→Microchip 53100A**, then select the *Internal reference* control page.

4. Ensure that INT REF 1's *Mode* control is set to either *Auto* or *External 10 MHz (front panel)*, and that the INT REF 2 *Mode* is set to *INT REF 1*. (These are the same defaults established by the *Restore PLL defaults* button.) Wait for the *Status* fields in both channels to reach the *Locked* state.

5. Ensure that the *Time* fields for both INT REF 1 and INT REF 2 are set to at least 10 seconds for best results. (*Restore PLL defaults* will set both fields to 100 seconds.)

6. Press the *Calibration* button and observe the *DAC voltage* fields for the INT REF 1 and INT REF 2 channels. If the external 10 MHz source was recently connected or if any loop parameters have recently been changed, these readings can be expected to fluctuate significantly as the control loops settle. For best results, wait until the DAC voltages remain within +/- 10 microvolts for several seconds before proceeding.

7. After the DAC voltages have stabilized, press the *Store user cal* buttons for both the INT REF 1 and INT REF 2 references. Observe that the *Cal value* fields capture the current DAC voltages, and that the calibration status line changes to "User cal <date> <time> <temperature>." Calibration of the channel-1 and channel-2 OCXOs is now complete.

(Option IR/STD only) To calibrate the internal atomic standard, simply connect the external 10 MHz standard as described above and press the *Store user cal* button in the control area for the internal atomic frequency standard. This procedure will take approximately 3-5 minutes. OCXO calibration does not need to be performed prior to atomic standard calibration.

Firmware updates

The Option IR module runs continuously under the control of its own microprocessor, even when the 53100A's rear-panel power switch is turned off. The latest firmware for the option module accompanies each TimeLab software release alongside the firmware for the 53100A itself. To install new firmware, open the *Additional options* dialog box from the 53100A acquisition dialog and click the *Update firmware* button, then select the appropriate Intel .hex file with the file-selection dialog box that appears.

It is not necessary to power-cycle or otherwise reboot the 53100A manually after installing new firmware, although phase lock will be lost during the cold-start cycle that is automatically initiated. As with any other cold-start cycle, please wait at least 30-60 minutes before attempting to make high-performance measurements after updating.

Online help

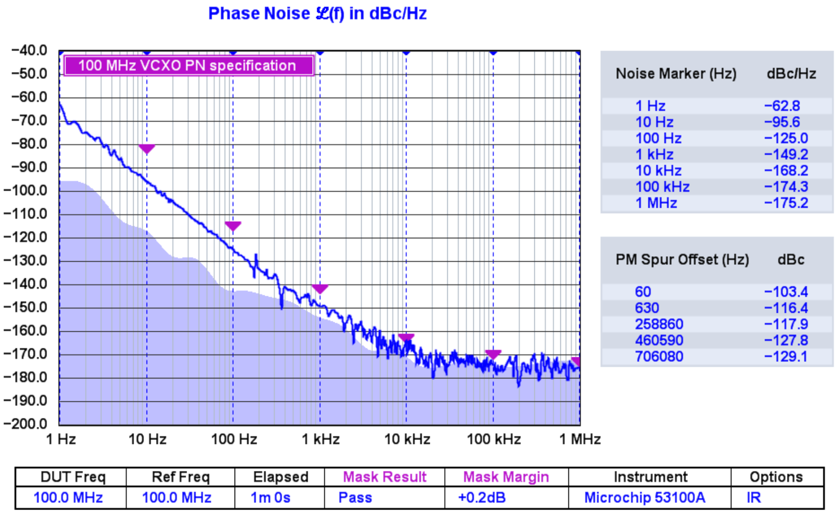
As with other acquisition dialog pages in TimeLab, individual controls and features in the *Internal reference* and *Calibration* dialogs may vary from one software release to the next. Detailed context-sensitive "mouseover" help text is provided for all dialog fields, and serves as the primary documentation for these controls. Please refer to the help text for the latest user information and operating tips.

Application examples

The following measurements share one aspect in common: they were carried out with no external equipment or reference sources other than a Microchip 53100A equipped with Option IR.

Phase noise of 100 MHz VCXO (1-minute and 30-minute runs)

In the first example, the 53100A was used to evaluate a high-performance VHF crystal oscillator module for compliance with its phase noise specifications. A one-minute test returned the following result (Figure 103):



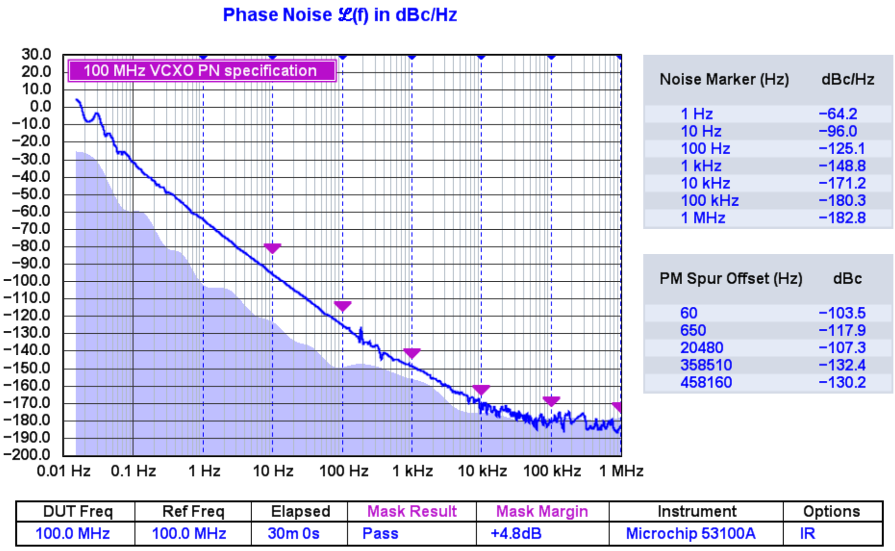
Noise Marker (Hz)	dBc/Hz
1 Hz	−62.8
10 Hz	−95.6
100 Hz	−125.0
1 kHz	−149.2
10 kHz	−168.2
100 kHz	−174.3
1 MHz	−175.2

PM Spur Offset (Hz)	dBc
60	−103.4
630	−116.4
258860	−117.9
460590	−127.8
706080	−129.1

DUT Freq	Ref Freq	Elapsed	Mask Result	Mask Margin	Instrument	Options
100.0 MHz	100.0 MHz	1m 0s	Pass	+0.2dB	Microchip 53100A	IR

Figure 103

The observed close-in noise is excellent for this class of device, with substantial headroom available at offsets below 10 kHz. Specifications at the higher-frequency offsets are met as well, but only barely. Along with the high instrument floor estimate, significant trace variance at offsets beyond a few kHz suggests that the cross-correlation measurement hasn't had enough time to converge fully. Running for 30 minutes instead of only one minute gives us a better clue to the oscillator's true performance, as well as that of the 53100A.



Noise Marker (Hz)	dBc/Hz
1 Hz	−64.2
10 Hz	−96.0
100 Hz	−125.1
1 kHz	−148.8
10 kHz	−171.2
100 kHz	−180.3
1 MHz	−182.8

PM Spur Offset (Hz)	dBc
60	−103.5
650	−117.9
20480	−107.3
358510	−132.4
458160	−130.2

DUT Freq	Ref Freq	Elapsed	Mask Result	Mask Margin	Instrument	Options
100.0 MHz	100.0 MHz	30m 0s	Pass	+4.8dB	Microchip 53100A	IR

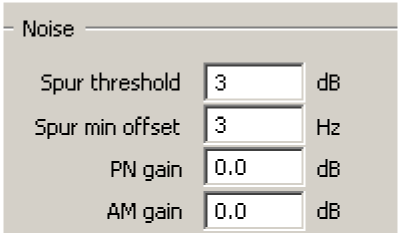
Figure 104

In Figure 104 above, the trace still hasn't converged fully beyond 10 kHz after 30 minutes -- and likely will not do so regardless of measurement duration, given that it is already well below the -175 dBc/Hz typical performance figures noted in the 53100A's own documentation -- but it is safe to conclude that the DUT's broadband noise floor is in the vicinity of -180 dBc/Hz.[18]

Most of the 53100A's acquisition parameters – and all of the Option IR loop parameters – were left at their default values for this measurement, apart from specifying the 1-minute and 30-minute measurement durations. Parameter changes made for this demonstration are listed below.

Set appropriate spur amplitude and offset thresholds

These options can be found on the *Additional options* page of the 53100A acquisition dialog (Figure 105). Especially if you use continuous limit-line masks for PN pass/fail evaluation, it's helpful to reduce the *Spur threshold* from 6 dB to 3 dB to keep small unrecognized spurs from artificially failing the PN test.



Noise

Spur threshold	3	dB
Spur min offset	3	Hz
PN gain	0.0	dB
AM gain	0.0	dB

Figure 105

False spur identification at close-in offsets can be avoided by setting a minimum offset for spur detection. (This effect is more likely to occur when testing sources with very low close-in noise, rather than the VCXO in this example.)

[18] Specifically, the oscillator has a carrier output power of +12 dBm and source impedance near 50 ohms. At this power level, a theoretically-ideal instrument could measure the PN noise at levels down to -177 dBm/Hz - 12 dBm + 3 dB = -186 dBc/Hz, allowing 3 dB of additional noise for the analyzer's own load impedance. Some caution is warranted as cross-correlating analyzers are subject to artifacts that can underestimate the true noise level when sources are operating near the thermal limit. However, measurement time constraints and very low-level instrument spurs usually determine the 53100A's performance floor before these effects become apparent.

Enable overlapped acquisition

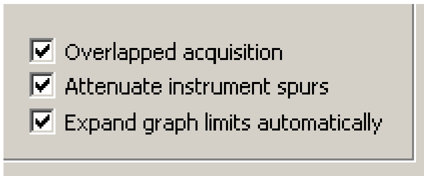

☑ Overlapped acquisition
☑ Attenuate instrument spurs
☑ Expand graph limits automatically

Figure 106

This will help low-level phase noise tests pass sooner, often within less than a minute (Figure 106). Try disabling AM noise measurements if USB overruns occur. Also, use the *System power* option to ensure that high performance power plan is in effect during acquisition.

Both of these fields are on the *Additional options* page of the main 53100A acquisition dialog.

Use the abs(I) cross spectrum estimator

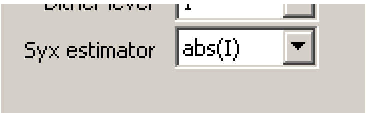
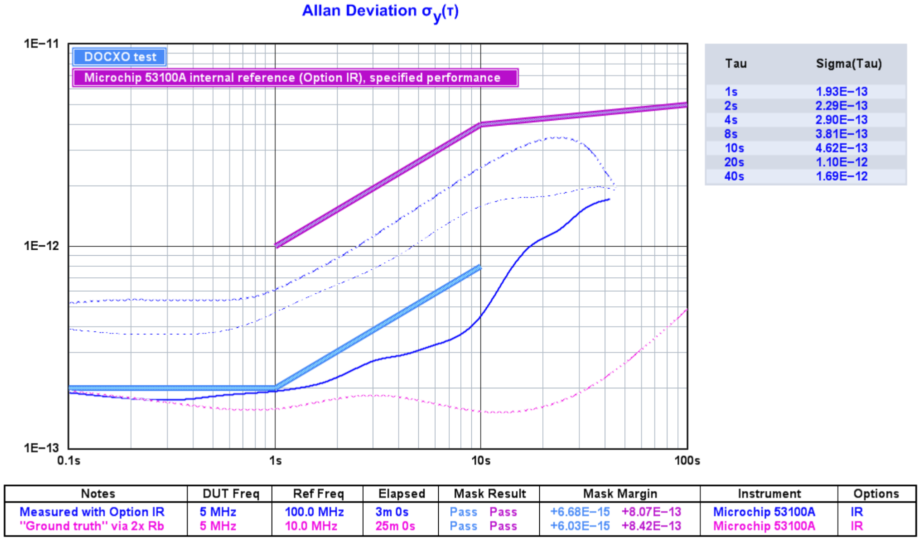

Syx estimator abs(I)

Figure 107

This improves test time in challenging measurements where the DUT noise floor is close to the instrument floor (Figure 107, also on the *Additional options* page). The trace may appear 'uglier' at first with numerous false spurs that may take a minute or two to resolve. However, the instrument noise contribution will be 3 dB lower, and there will be a corresponding improvement in measurement convergence time.

Short-term stability of ultrastable 5 MHz DOCXO

In our second scenario, the goal is to verify that a double-oven OCXO meets its short-term Allan deviation specifications of 2E-13 @ t=0.1s, 2E-13 @ t=1s, and 8E-13 @ t=10s. Can the 53100A's internal reference module make this measurement, given that its own absolute ADEV specifications are several times higher? Yes, as Figure 108 reveals – as long as a few prerequisites are understood.



Tau	Sigma(Tau)
1s	1.93E−13
2s	2.29E−13
4s	2.90E−13
8s	3.81E−13
10s	4.62E−13
20s	1.10E−12
40s	1.69E−12

Notes	DUT Freq	Ref Freq	Elapsed	Mask Result	Mask Margin	Instrument	Options
Measured with Option IR	5 MHz	100.0 MHz	3m 0s	Pass Pass	+6.68E−15 +8.07E−13	Microchip 53100A	IR
"Ground truth" via 2x Rb	5 MHz	10.0 MHz	25m 0s	Pass Pass	+6.03E−15 +8.42E−13	Microchip 53100A	IR

Figure 108

Select a measurement role with cross ADEV support

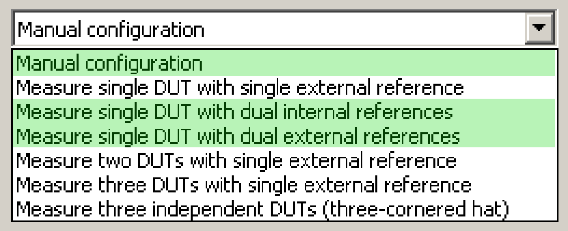
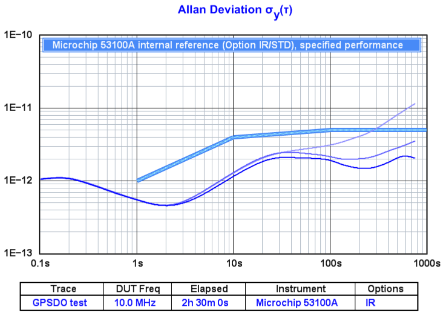

Manual configuration
Manual configuration
Measure single DUT with single external reference
Measure single DUT with dual internal references
Measure single DUT with dual external references
Measure two DUTs with single external reference
Measure three DUTs with single external reference
Measure three independent DUTs (three-cornered hat)

Figure 109

Cross Allan deviation is not a unique measurement type in itself, but rather a way to perform traditional Allan deviation measurements using multiple channels and/or reference sources in a manner reminiscent of more familiar cross-correlated phase noise measurement techniques. Just as with cross-correlated PN measurements, both reference stability and instrument ADC noise performance undergo progressive improvement over time by averaging the results of two identical measurements carried out simultaneously. While the Microchip 53100A can make cross ADEV measurements using a pair of external or internal references, the technique is an especially-good fit for Option IR instruments.

Referring to the main page of the 53100A'a acquisition dialog, any of the predefined measurement role options highlighted in Figure 109 above can be used to perform cross ADEV measurements. The required secondary phase data record is automatically created whenever channels 3 and 4 have separate *Stability reference* channels assigned to them. This will be the case whenever one of the predefined dual-reference roles is chosen, as well as when *Manual configuration* is used to set up a measurement that meets the same conditions. (Note that *Always use cross ADEV when possible* must also be checked on the *Additional options* page of the acquisition dialog. This checkbox is selected by default.)

In TimeLab itself, make sure that **Trace→Show cross ADEV traces when available (Ctrl-j)** is enabled. For the screenshot example in Figure 108, we've also selected **Trace→Show original traces in computed xDEV displays (F6)** to highlight the improvement in the stability measurement floor achieved with cross ADEV. The heavy and light dotted traces in **blue** represent the measurements that would have been obtained by measuring the DUT against either the INT REF 1 or INT REF 2 reference source by itself. (Also for the sake of comparison, the **magenta** trace shows the actual performance of the DOCXO under test as measured with a pair of lab-grade rubidium frequency standards serving as reference sources.)



Trace	DUT Freq	Elapsed	Instrument	Options
GPSDO test	10.0 MHz	2h 30m 0s	Microchip 53100A	IR

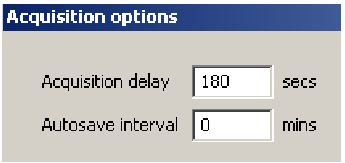
Figure 110

Consider using deferred acquisition if necessary

Given adequate warmup time and an appropriate PLL time constant, drift is negligible when INT REF 1 is locked to an internal or external 10 MHz standard. When one or both OCXOs is allowed to free-run, however, temperature variations associated with the onset of USB data acquisition can induce a small amount of frequency drift in the early part of the measurement. Challenging ADEV measurements such as the one in this example may take longer than expected to settle as a result.

As with other temperature-related effects, initial drift can often be diagnosed by using **Edit→Trace properties (e)** to set the *Trace history* parameter to 2 or 3 (Figure 110). Here, the darker ADEV traces correspond to later segments of the measurement's phase record.

One way to avoid this issue is to tell TimeLab to discard the initial subset of measurement data by using **Acquire→Acquisition options** to specify a delay as shown in Figure 111. Subsequently, select the **Acquire→Enable deferred acquisition** option to enable the delay. The measurement will begin as usual, but by selecting an appropriate *Acquisition delay* parameter, no data will actually be recorded until temperature equilibrium has been established.



Acquisition options

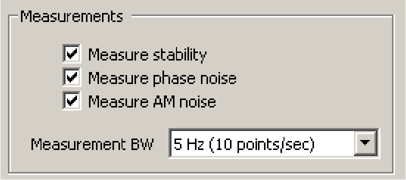
Acquisition delay 180 secs

Autosave interval 0 mins

Figure 111

Reduce the measurement bandwidth

The DOCXO that we measured in this example would not have passed its ADEV test at the default 50 Hz (100 points/sec) measurement bandwidth due to the unusually tight performance requirement at t=0.1s. It was necessary to select 5 Hz instead (Figure 112).



Measurements

☑ Measure stability

☑ Measure phase noise

☑ Measure AM noise

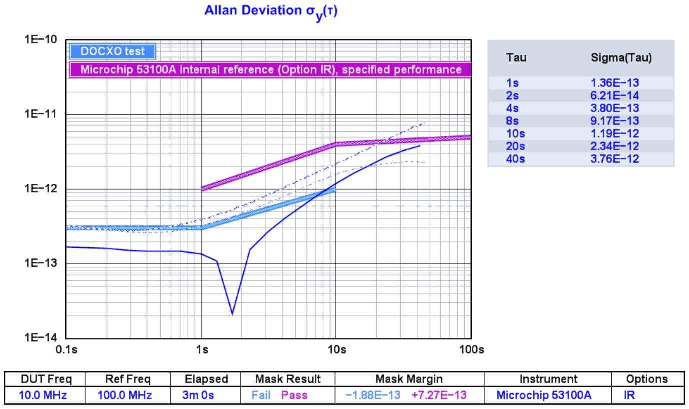
Measurement BW 5 Hz (10 points/sec)

Figure 112

Note that the choice of measurement bandwidth does not affect noise measurements, only stability measurements.

Ensure references are disciplined appropriately

To achieve the lowest ADEV measurement floor near t=1s, it may be tempting to configure INT REF 1 and/or INT REF 2 in free-running mode. Doing so can keep the external or internal standard from contributing to short-term reference instability, but the resulting drift may give rise to artifacts that take a very long time to resolve, such as the large divot in Figure 113 that's caused by leaving both references unlocked.



Tau	Sigma(Tau)
1s	1.36E−13
2s	6.21E−14
4s	3.80E−13
8s	9.17E−13
10s	1.19E−12
20s	2.34E−12
40s	3.76E−12

DUT Freq	Ref Freq	Elapsed	Mask Result	Mask Margin	Instrument	Options
10.0 MHz	100.0 MHz	3m 0s	Fail Pass	−1.88E−13 +7.27E−13	Microchip 53100A	IR

Figure 113

While both cross ADEV and phase noise measurements rely on statistical independence of INT REF 1 and INT REF 2 at the taus and offset frequencies of interest, the choice of lock bandwidth is more likely to be dictated by the maximum ADEV tau of interest than by the minimum PN offset requirement. INT REF 2's lock bandwidth will have a noticeable impact on the cross ADEV measurement floor at taus exceeding 5% to 10% of the lock time constant in seconds.

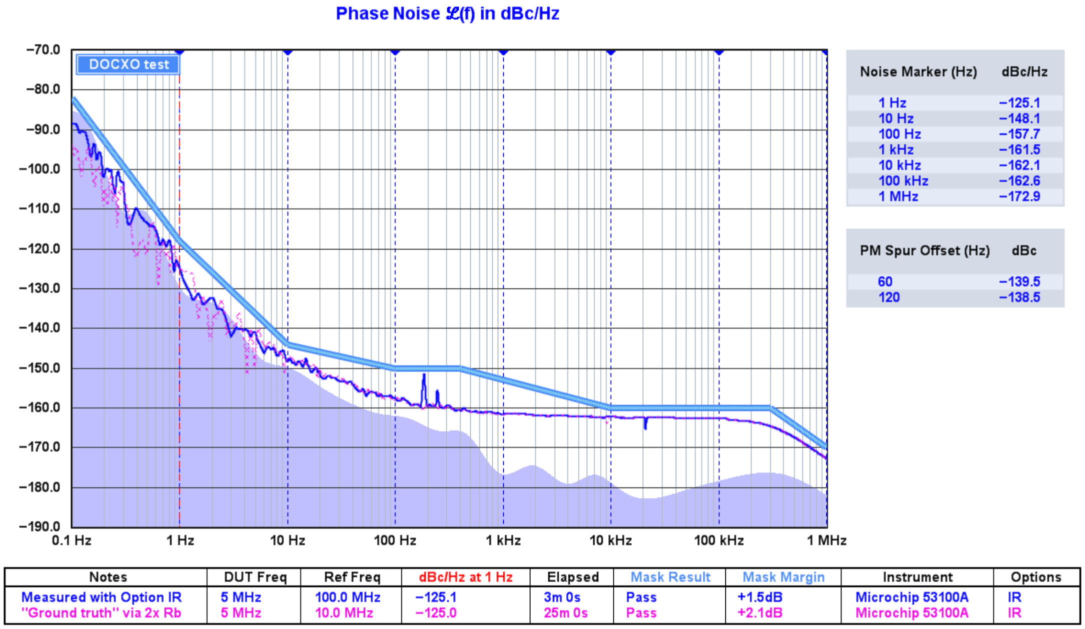
Consequently, in the absence of an external 10 MHz reference with excellent short-term stability that can be used as a disciplining source for INT REF 1, a better strategy for measuring low ADEV levels at taus exceeding a few seconds can be to operate INT REF 1 with a longer-than-usual time constant – e.g., 200 seconds or more, particularly if locking to the internal atomic standard – while locking INT REF 2 to INT REF 1 with a similar time constant. INT REF 1 can also be allowed to free-run while INT REF 2 is locked to it. (All of that being said, the result in Figure 108 was obtained with the factory-default lock parameters.)

On the other hand, when a high-quality 10 MHz reference such as a maser or ultrastable OCXO is available, consider using short time constants on the order of 1 second for both INT REF 1 *and* INT REF 2. Stability measurements will inherit both the short-term and long-term characteristics of your external reference in this case, while phase noise measurements at offsets beyond a few hertz will benefit from the independent low-noise internal OCXOs as usual.

As a further note, external 1PPS references should be avoided in demanding applications such as this example. Residual ADEV from the 1PPS source is approximately 1E-11 at t=30s with the default 1PPS disciplining parameters. The 10 MHz loops have significantly better residual performance, are much easier to tune, and lock much more rapidly.

Close-in phase noise of ultrastable 5 MHz DOCXO

The phase noise of the DOCXO was also measured during the previous test, and is now evaluated for compliance with specified limits (Figure 114):



Noise Marker (Hz)	dBc/Hz
1 Hz	−125.1
10 Hz	−148.1
100 Hz	−157.7
1 kHz	−161.5
10 kHz	−162.1
100 kHz	−162.6
1 MHz	−172.9

PM Spur Offset (Hz)	dBc
60	−139.5
120	−138.5

Notes	DUT Freq	Ref Freq	dBc/Hz at 1 Hz	Elapsed	Mask Result	Mask Margin	Instrument	Options
Measured with Option IR	5 MHz	100.0 MHz	−125.1	3m 0s	Pass	+1.5dB	Microchip 53100A	IR
"Ground truth" via 2x Rb	5 MHz	10.0 MHz	−125.0	25m 0s	Pass	+2.1dB	Microchip 53100A	IR

Figure 114

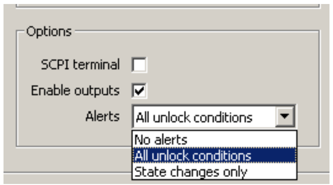
As before, the **blue** trace represents the measurement made by the standalone 53100A equipped with Option IR in 3 minutes, while the **magenta** trace represents the same DUT measured against a pair of high performance rubidium standards over 25 minutes. Both measurements are sufficient to confer a passing grade on the DUT. In particular, the self-contained Option IR references were able to measure the oscillator's 1 Hz phase noise at -125 dBc/Hz -- a figure achieved by a relatively small fraction of devices on the market – while additionally confirming performance below -170 dBc/Hz at 1 MHz.

Additional tips for high-performance measurements with Option IR

Allow sufficient stabilization time after applying power or changing loop parameters

As noted earlier, the Option IR module is active whenever the 53100A is connected to its power supply, regardless of the power switch setting. The firmware enforces a 3-minute warmup period after power application before any lock attempts are made. This represents a minimum requirement. It is strongly recommended that the reference module be allowed to warm up for at least 30-60 minutes before making measurements with a 53100A that has just been connected to its power supply. When freshly powered up, the INT REF 1 OCXO may require multiple attempts to lock to an external 10 MHz source or to the optional internal atomic standard – and if INT REF 2 is locked to INT REF 1, it may need some extra time as well.

Observe lock status before and during the measurement


Options

SCPI terminal □

Enable outputs ☑

Alerts All unlock conditions

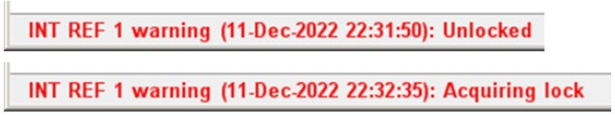
No alerts
All unlock conditions
State changes only

Figure 115

Prior to starting a measurement, you can monitor the lock status for both INT REF 1 and INT REF 2 by clicking on the *Internal reference* button in the 53100A acquisition dialog to bring up the Option IR configuration dialog box shown in Figure 115. The read-only *Reference*, *Status*, and *Error* fields in the configuration dialog will be updated periodically to reflect the current INT REF 1 and INT REF 2 lock states. Lock status messages may be observed in the SCPI terminal, which can be toggled on and off with the corresponding checkbox in the *Options* area.

Additionally, the tuning voltages for any OCXOs undergoing phase locking are also updated in real time on the *Calibration* page. Stable lock conditions exist when the least significant digits of the tuning voltages are no longer changing rapidly.

During the measurement itself, there are two ways to keep an eye on the reference's lock status. First, whenever operating conditions such as loss-of-lock, OCXO and atomic standard warmup, and 1PPS holdover are detected by the reference module's onboard controller, they will be reported by default in the status line at the bottom of the main TimeLab window. Refer to the help text for the *Alerts* control in the *Options* area of the Option IR configuration dialog box (Figure 101) for more information about the conditions that generate these messages. The most common alerts from the internal reference module will resemble Figure 116 below, indicating that the reference module is attempting to return to a phase-locked condition after excessive drift, external signal loss, or other hardware or environmental issue has interrupted the disciplining process. Timestamps help determine when the most recent interruption or other event occurred in the course of a long (and possibly unattended) measurement.



INT REF 1 warning (11-Dec-2022 22:31:50): Unlocked

INT REF 1 warning (11-Dec-2022 22:32:35): Acquiring lock

Figure 116

For more specific information about the operating status of the Option IR module while a measurement is in progress, you can access the *Internal reference* and *Calibration* dialogs in much the same way as you would when the 53100A is idle. Select **Acquire→Microchip 53100A** in TimeLab, just as if you were preparing to initiate a new measurement. The device selection menu entry for the 53100A that's currently performing the measurement will be marked *(In use)*, with all of the other controls in the acquisition dialog box except for the *Cancel* and *Internal reference* buttons disabled. Selecting *Internal reference* will close the acquisition dialog and bring up the Option IR configuration dialog in its place. All options and controls in this dialog box, as well as those on the *Calibration* page, are available for use while the 53100A is collecting data. Note, however, that any Option IR dialog boxes will automatically be closed when the measurement ends, cancelling any operations such as user calibration of the optional atomic frequency standard that may be in progress at the time.

Access to the Option IR dialog boxes at measurement time can be invaluable for monitoring and diagnostic purposes. Of course, any attempt to change disciplining sources, make significant adjustments to the PLL parameters, or select different calibration states may have adverse effects on the measurement in progress. These operations are best performed before initiating the measurement.

Use high-grade cables to connect the DUT

The use of cables with low shield resistance, as measured with an ohmmeter from shell to shell, is especially important for PN measurements that are expected to reach -170 dBc/Hz and below.

Shield DUT from environmental effects

Devices that are exposed to the open air on a PCB or breadboard will be vulnerable to effects such as HVAC circulation and local convection currents that make it difficult to pass demanding ADEV tests. They can also exhibit higher-than-necessary levels of close-in phase noise. Both environmental and RF isolation can be improved by enclosing the DUT in an insulated housing for test purposes (Figure 117). This was necessary in order to achieve the results obtained here for both the 10 MHz DOCXO and the 100 MHz VCXO.

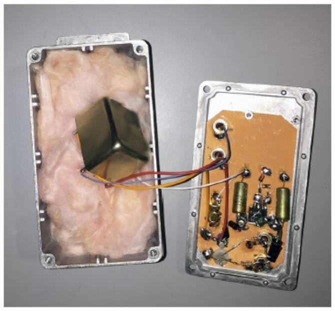


Figure 117

Appendix: Legal and regulatory notices

Federal Communications Commission Statement

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna
- Increase the separation between the equipment and receiver
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected
- Consult the dealer or an experienced radio/TV technician for help

Changes or modifications not expressly approved by the party responsible for compliance may void the user's authority to operate the equipment.

EC Declaration of Conformity

Manufacturer: Jackson Labs Technologies, Inc., 10080 Alta Drive , Suite 110, Las Vegas, NV 89145, USA

Product: PhaseStation 53100A Phase Noise Test Set

Description: Small/portable electronic instrument for laboratory use by qualified personnel, powered by external DC supply with UL/CUL/TUV/CB/CE approvals

Directives:

2006/95/EC The Low Voltage Directive and its amending directives

2004/108/EC The Electromagnetic Compatibility Directive and its amending directives

2002/95/EC Restrictions on Hazardous Substances and its amending directives

Standards:

EN 61010-1:2001 Safety Requirements for Electrical Equipment for Measurement, Control and Laboratory Use – Part 1: General Requirements

EN 61326-1:2001 Electrical Requirements for Electrical Equipment for Measurement, Control and Laboratory Use – Part 1: General Requirements

Signature: I hereby certify that this equipment has been designed and manufactured in accordance with the above referenced Standards, and complies with all essential requirements of the Directives.

John Miles
Engineer, Jackson Labs Technologies, Inc.
July 11, 2019

Performance Certification and Validation

Jackson Labs Technologies, Inc. certifies that this product met its published specifications at time of delivery. Because measurements made by the PhaseStation 53100A Phase Noise Test Set are based on differences between the external signals applied to the DUT and REFERENCE ports, no specific periodic calibration procedures are required. Furthermore, in the absence of errors reported by the measurement software, no specific maintenance procedures are necessary, including preventative or periodic maintenance.

The procedures outlined in the default mask definition file (**default_masks.txt**) will permit customer verification of all specifications using commonly available test signals. Users requiring annual performance certification should contact http://www.jackson-labs.com/index.php/support for factory-certified performance verification options.

Warranty

This Jackson Labs Technologies, Inc. hardware product is warranted against defects in material and workmanship for a period of 1 (one) year from date of delivery. During the warranty period Jackson Labs Technologies, Inc. will, at its discretion, either repair or replace products that prove to be defective. Jackson Labs Technologies, Inc. does not warrant that the operation for the software, firmware, or hardware shall be uninterrupted or error free even if the product is operated within its specifications. For warranty service, this product must be returned to Jackson Labs Technologies, Inc. or a service facility designated by Jackson Labs Technologies, Inc. Customer shall prepay shipping charges (and shall pay all duties and taxes) for products returned to Jackson Labs Technologies, Inc. for warranty service. Except for products returned to Customer from another country, Jackson Labs Technologies, Inc. shall pay for return of products to Customer. If Jackson Labs Technologies, Inc. is unable, within a reasonable time, to repair or replace any product to condition as warranted, the Customer shall be entitled to a refund of the purchase price upon return of the product to Jackson Labs Technologies, Inc.

Limitation of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by the Customer, Customer-supplied software or interfacing, unauthorized modification or misuse, opening of the instruments enclosure or removal of the instruments panels, operation outside of the environmental or electrical specifications for the product, or improper site preparation and maintenance. **JACKSON LABS TECHNOLOGIES, INC. SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE**. **No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Jackson Labs Technologies, Inc. products are not intended for use in medical, life saving, or life sustaining applications.**

Exclusive Remedies

THE REMEDIES PROVIDED HEREIN ARE THE CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES. JACKSON LABS TECHNOLOGIES, INC. SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

Software License: TimeLab

http://www.miles.io/timelab/readme.htm

Software License: FFTSS

http://www.ssisc.org/fftss/index.html.en

NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Software License: FIDLIB

http://uazu.net

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Software License: V8 JavaScript Engine

https://v8.dev