User Manual

Tektronix

TVS600 Series
Waveform Analyzers
(TVS621, TVS625, TVS641 & TVS645)

070-9283-01

This document applies for firmware version 1.5
and above.

www.valuetronics.com

Copyright © Tektronix, Inc. 1995. All rights reserved. Licensed software products are owned by Tektronix or its suppliers and are protected by United States copyright laws and international treaty provisions.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, or subparagraphs (c)(1) and (2) of the Commercial Computer Software – Restricted Rights clause at FAR 52.227-19, as applicable.

Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supercedes that in all previously published material. Specifications and price change privileges reserved.

Printed in the U.S.A.

Tektronix, Inc., P.O. Box 1000, Wilsonville, OR 97070–1000

TEKTRONIX and TEK are registered trademarks, and IntelliFrame is a trademark of Tektronix, Inc.

www.valuetronics.com

WARRANTY

Tektronix warrants that the products that it manufactures and sells will be free from defects in materials and workmanship for a period of three (3) years from the date of shipment. If a product proves defective during this warranty period, Tektronix, at its option, either will repair the defective product without charge for parts and labor, or will provide a replacement in exchange for the defective product.

In order to obtain service under this warranty, Customer must notify Tektronix of the defect before the expiration of the warranty period and make suitable arrangements for the performance of service. Customer shall be responsible for packaging and shipping the defective product to the service center designated by Tektronix, with shipping charges prepaid. Tektronix shall pay for the return of the product to Customer if the shipment is to a location within the country in which the Tektronix service center is located. Customer shall be responsible for paying all shipping charges, duties, taxes, and any other charges for products returned to any other locations.

This warranty shall not apply to any defect, failure or damage caused by improper use or improper or inadequate maintenance and care. Tektronix shall not be obligated to furnish service under this warranty a) to repair damage resulting from attempts by personnel other than Tektronix representatives to install, repair or service the product; b) to repair damage resulting from improper use or connection to incompatible equipment; c) to repair any damage or malfunction caused by the use of non-Tektronix supplies; or d) to service a product that has been modified or integrated with other products when the effect of such modification or integration increases the time or difficulty of servicing the product.

THIS WARRANTY IS GIVEN BY TEKTRONIX IN LIEU OF ANY OTHER WARRANTIES, EXPRESS OR IMPLIED. TEKTRONIX AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TEKTRONIX' RESPONSIBILITY TO REPAIR OR REPLACE DEFECTIVE PRODUCTS IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY. TEKTRONIX AND ITS VENDORS WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER TEKTRONIX OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

www.valuetronics.com

Tektronix

EC Declaration of Conformity

We

Tektronix Holland N.V.
Marktweg 73A
8444 AB Heerenveen
The Netherlands

declare under sole responsibility that the

TVS600 Series Waveform Analyzers

meet the intent of Directive 89/336/EEC for Electromagnetic Compatibility. Compliance was demonstrated to the following specifications as listed in the official Journal of the European Communities:

EN 50081–1 Emissions:

EN 55011	Radiated, Class A
EN 55011	Conducted, Class A
EN 60555–2	Power Harmonics

EN 50082–1 Immunity:

IEC 801–2	Electrostatic Discharge
IEC 801–3	RF Radiated
IEC 801–4	Fast Transients

www.valuetronics.com

Table of Contents

List of Figures . . . . . **xi**
List of Tables . . . . . **xiv**
General Safety Summary . . . . . **xvii**
About This Manual xxi
Related Manuals xxi

Getting Started

Getting Started **1–1**
Product Description 1–1
Accessories 1–1
Configuration 1–2
Installation 1–5
Software Installation 1–8
Incoming Inspection Procedure **1–13**
Connect the VXIbus Test System 1–15
Self Tests 1–15
Functional Tests 1–17
Self Cal 1–20

Operating Basics

Operating Basics **2–1**
Connectors and Indicators **2–3**
Functional Overview **2–7**
System Operation 2–7
Acquisition 2–9
Vertical Range and Offset 2–19
Triggering 2–21
Calculations 2–31
Fast Data Channel 2–50
Data Transfer Formats 2–55
Instrument I/O **2–63**
VXIbus Interface 2–63
RS-232C Port 2–69
Tutorial **2–71**
Example 1 — Instrument Setup 2–72
Example 2 — Acquiring a Signal 2–74
Example 3 — Averaging and Enveloping a Signal 2–75
Example 4 — Performing Basic Calculations 2–76
Example 5 — Performing Advanced Calculations 2–78
Example 6 — Saving and Recalling Settings 2–79
Example 7 — Using Status and Events 2–81

www.valuetronics.com

Syntax and Commands

Command Syntax 3–1
SCPI Commands and Queries 3–1
IEEE 488.2 Common Commands 3–5
Constructed Mnemonics 3–6

Command Groups 3–9
Auto-Advance Commands 3–9
Abort Commands 3–9
Arm Commands 3–10
Average Commands 3–10
Calculate Commands 3–10
Calibration Commands 3–13
Data Commands 3–13
Format Commands 3–14
Function Commands 3–14
Initiate Commands 3–15
Input Commands 3–15
Memory Commands 3–16
Output Commands 3–16
Roscillator Commands 3–17
Sense Commands 3–17
Status Commands 3–18
Sweep Commands 3–19
System Commands 3–20
Test Commands 3–21
Trace Commands 3–21
Trigger Commands 3–22
Voltage Commands 3–24
IEEE 488.2 Commands 3–24

Commands 3–27
Overview 3–27

AADVance Subsystem 3–29
AADVance
AADVance? 3–30
AADVance:COUNt
AADVance:COUNt? 3–31
AADVance:RECord:COUNt
AADVance:RECord:COUNt? 3–32
AADVance:RECord:STARt
AADVance:RECord:STARt? 3–33

ARM Subsystem 3–35
ARM:DEFine? (Query Only) 3–35
ARM:SOURce
ARM:SOURce? 3–36

AVERage Subsystem 3–39
AVERage
AVERage? 3–39
AVERage:COUNt
AVERage:COUNt? 3–40

www.valuetronics.com

AVERage:TYPE
AVERage:TYPE? 3–41
CALCulate Subsystem 3–43
CALCulate:AAMList
CALCulate:AAMList? 3–44
CALCulate:AAMList:STATe
CALCulate:AAMList:STATe? 3–45
CALCulate:DATA? (Query Only) 3–46
CALCulate:DATA:PREamble? (Query Only) 3–47
CALCulate:DERivative:STATe
CALCulate:DERivative:STATe? 3–48
CALCulate:FEED1
CALCulate:FEED1? 3–49
CALCulate:FEED2
CALCulate:FEED2? 3–51
CALCulate:FEED2:CONTEXT
CALCulate:FEED2:CONTEXT? 3–52
CALCulate:FILTer:FREQuency[:TYPE]
CALCulate:FILTer:FREQuency[:TYPE]? 3–54
CALCulate:FILTer:FREQuency:CENTer
CALCulate:FILTer:FREQuency:CENTer? 3–55
CALCulate:FILTer:FREQuency:HPASs
CALCulate:FILTer:FREQuency:HPASs? 3–57
CALCulate:FILTer:FREQuency:LPASs
CALCulate:FILTer:FREQuency:LPASs? 3–58
CALCulate:FILTer:FREQuency:SPAN
CALCulate:FILTer:FREQuency:SPAN? 3–59
CALCulate:FILTer:FREQuency:SREJection
CALCulate:FILTer:FREQuency:SREJection? 3–60
CALCulate:FILTer:FREQuency:STARt
CALCulate:FILTer:FREQuency:STARt? 3–61
CALCulate:FILTer:FREQuency:STATe
CALCulate:FILTer:FREQuency:STATe? 3–63
CALCulate:FILTer:FREQuency:STOP
CALCulate:FILTer:FREQuency:STOP? 3–64
CALCulate:FILTer:FREQuency:TWIDth
CALCulate:FILTer:FREQuency:TWIDth? 3–65
CALCulate:FORMat
CALCulate:FORMat? 3–66
CALCulate:IMMediate
CALCulate:IMMediate? 3–68
CALCulate:INTegral:STATe
CALCulate:INTegral:STATe? 3–69
CALCulate:PATH
CALCulate:PATH? 3–70
CALCulate:PATH:EXPRession
CALCulate:PATH:EXPRession? 3–71
CALCulate:SMOothing
CALCulate:SMOothing? 3–73
CALCulate:SMOothing:POINts
CALCulate:SMOothing:POINts? 3–74
CALCulate:TRANsform:FREQuency:STATe
CALCulate:TRANsform:FREQuency:STATe? 3–75

www.valuetronics.com

CALCulate:TRANsform:FREQuency:WINDow
CALCulate:TRANsform:FREQuency:WINDow? 3–76
CALCulate:WMList
CALCulate:WMList? 3–78
CALCulate:WMList:STATe
CALCulate:WMList:STATe? 3–82
CALCulate:WMParameter:HIGH
CALCulate:WMParameter:HIGH? 3–83
CALCulate:WMParameter:HMEThod
CALCulate:WMParameter:HMEThod? 3–84
CALCulate:WMParameter:LOW
CALCulate:WMParameter:LOW? 3–85
CALCulate:WMParameter:LMEThod
CALCulate:WMParameter:LMEThod? 3–86
CALCulate:WMParameter:HREFerence
CALCulate:WMParameter:HREFerence? 3–88
CALCulate:WMParameter:HREFerence:RELative
CALCulate:WMParameter:HREFerence:RELative? 3–89
CALCulate:WMParameter:LREFerence
CALCulate:WMParameter:LREFerence? 3–90
CALCulate:WMParameter:LREFerence:RELative
CALCulate:WMParameter:LREFerence:RELative? 3–91
CALCulate:WMParameter:MREFerence
CALCulate:WMParameter:MREFerence? 3–92
CALCulate:WMParameter:MREFerence:HYSTeresis
CALCulate:WMParameter:MREFerence:HYSTeresis? 3–93
CALCulate:WMParameter:MREFerence:RELative
CALCulate:WMParameter:MREFerence:RELative? 3–94
CALCulate:WMParameter:RMEThod
CALCulate:WMParameter:RMEThod? 3–95

CALibration Subsystem 3–97
CALibration
CALibration? 3–97
CALibration:RESults? (Query Only) 3–98
CALibration:RESults:VERBose? (Query Only) 3–99

FORMat Subsystem 3–101
FORMat
FORMat? 3–101
FORMat:CALCulate
FORMat:CALCulate? 3–102
FORMat:TRACe:AATS
FORMat:TRACe:AATS? 3–103
FORMat:BORDer
FORMat:BORDer? 3–104

FUNCtion and DATA Subsystems 3–105
FUNCtion[:ON]
FUNCtion[:ON]? 3–106
FUNCtion[:ON]:ALL 3–107
FUNCtion[:ON]:COUNt? (Query Only) 3–108
FUNCtion:OFF
FUNCtion:OFF? 3–109
FUNCtion:OFF:ALL 3–110
FUNCtion:OFF:COUNt? (Query Only) 3–111

www.valuetronics.com

FUNCtion:CONCurrent
FUNCtion:CONCurrent? 3–112
FUNCtion:STATe
FUNCtion:STATe? 3–113
DATA? (Query Only) 3–114
DATA:PREamble? (Query Only) 3–115
INITiate and ABORt Subsystems 3–117
INITiate 3–117
INITiate:CONTinuous
INITiate:CONTinuous? 3–118
INITiate:COUNt
INITiate:COUNt? 3–119
ABORt 3–121
INPut Subsystem 3–123
INPut:COUPling
INPut:COUPling? 3–124
INPut:FILTer
INPut:FILTer? 3–125
INPut:FILTer:FREQuency
INPut:FILTer:FREQuency? 3–126
INPut:IMPedance
INPut:IMPedance? 3–127
INPut:PROTection:STATe
INPut:PROTection:STATe? 3–128
MEMory Subsystem 3–131
MEMory:DATA
MEMory:DATA? 3–131
MEMory:NSTates? (Query Only) 3–133
MEMory:STATe:CATalog? (Query Only) 3–134
MEMory:STATe:DEFine? (Query Only) 3–135
OUTPut Subsystem 3–137
OUTPut:ECLTrg<n>
OUTPut:ECLTrg<n>? 3–137
OUTPut:ECLTrg<n>:POLarity
OUTPut:ECLTrg<n>:POLarity? 3–138
OUTPut:ECLTrg<n>:SOURce
OUTPut:ECLTrg<n>:SOURce? 3–139
OUTPut:PCOMpensate
OUTPut:PCOMpensate? 3–140
OUTPut:PCOMpensate:FUNCtion
OUTPut:PCOMpensate:FUNCtion? 3–141
OUTPut:REFerence
OUTPut:REFerence? 3–142
OUTPut:REFerence:FUNCtion
OUTPut:REFerence:FUNCtion? 3–143
OUTPut:TTLTrg<n>
OUTPut:TTLTrg<n>? 3–144
OUTPut:TTLTrg<n>:POLarity
OUTPut:TTLTrg<n>:POLarity? 3–145
OUTPut:TTLTrg<n>:SOURce
OUTPut:TTLTrg<n>:SOURce? 3–146
ROSCillator Subsystem 3–149

www.valuetronics.com

ROSCillator:SOURce
ROSCillator:SOURce? 3–150

STATus Subsystem 3–151
STATus:OPERation? (Query Only) 3–151
STATus:OPERation:CONDition? (Query Only) 3–153
STATus:OPERation:ENABle
STATus:OPERation:ENABle? 3–154
STATus:OPERation:NTRansition
STATus:OPERation:NTRansition? 3–155
STATus:OPERation:PTRansition
STATus:OPERation:PTRansition? 3–156
STATus:OPERation:QENable:NTRansition
STATus:OPERation:QENable:NTRansition? 3–157
STATus:OPERation:QENable:PTRansition
STATus:OPERation:QENable:PTRansition? 3–158
STATus:PRESet 3–159
STATus:QUEStionable? (Query Only) 3–160
STATus:QUEStionable:CONDition? (Query Only) 3–161
STATus:QUEStionable:ENABle
STATus:QUEStionable:ENABle? 3–162
STATus:QUEStionable:NTRansition
STATus:QUEStionable:NTRansition? 3–163
STATus:QUEStionable:PTRansition
STATus:QUEStionable:PTRansition? 3–164
STATus:QUEStionable:QENable:NTRansition
STATus:QUEStionable:QENable:NTRansition? 3–165
STATus:QUEStionable:QENable:PTRansition
STATus:QUEStionable:QENable:PTRansition? 3–166
STATus:SESR:QENable
STATus:SESR:QENable? 3–168

SWEep Subsystem 3–169
SWEep:OFFSet:POINts
SWEep:OFFSet:POINts? 3–170
SWEep:OFFSet:TIME
SWEep:OFFSet:TIME? 3–171
SWEep:OREFerence:LOCation
SWEep:OREFerence:LOCation? 3–173
SWEep:POINts
SWEep:POINts? 3–174
SWEep:TIME
SWEep:TIME? 3–176
SWEep:TINTerval
SWEep:TINTerval? 3–177

SYSTem Subsystem 3–179
SYSTem:COMMunicate:SERial:BAUD
SYSTem:COMMunicate:SERial:BAUD? 3–179
SYSTem:COMMunicate:SERial:CONTrol:DCD
SYSTem:COMMunicate:SERial:CONTrol:DCD? 3–180
SYSTem:COMMunicate:SERial:CONTrol:RTS
SYSTem:COMMunicate:SERial:CONTrol:RTS? 3–181
SYSTem:COMMunicate:SERial:ECHO
SYSTem:COMMunicate:SERial:ECHO? 3–182

www.valuetronics.com

SYSTem:COMMunicate:SERial:ERESponse
SYSTem:COMMunicate:SERial:ERESponse? 3–183
SYSTem:COMMunicate:SERial:LBUFfer
SYSTem:COMMunicate:SERial:LBUFfer? 3–184
SYSTem:COMMunicate:SERial:PACE
SYSTem:COMMunicate:SERial:PACE? 3–185
SYSTem:COMMunicate:SERial:PARity
SYSTem:COMMunicate:SERial:PARity? 3–186
SYSTem:COMMunicate:SERial:PRESet 3–187
SYSTem:COMMunicate:SERial:SBITs
SYSTem:COMMunicate:SERial:SBITs? 3–188
SYSTem:ERRor? (Query Only) 3–189
SYSTem:ERRor:ALL? (Query Only) 3–190
SYSTem:ERRor:CODE? (Query Only) 3–191
SYSTem:ERRor:CODE:ALL? (Query Only) 3–192
SYSTem:ERRor:COUNt? (Query Only) 3–193
SYSTem:PROTect
SYSTem:PROTect? 3–193
SYSTem:SECurity:IMMediate 3–194
SYSTem:SET
SYSTem:SET? 3–195
SYSTem:VERSion? (Query Only) 3–196

TEST Subsystem 3–197

TEST
TEST? 3–197
TEST:RESults? (Query Only) 3–198
TEST:RESults:VERBose? (Query Only) 3–199

TRACe Subsystem 3–201

TRACe? (Query Only) 3–202
TRACe:PREamble? (Query Only) 3–203
TRACe:CATalog? (Query Only) 3–205
TRACe:COPY 3–206
TRACe:FEED? (Query Only) 3–207
TRACe:LIST
TRACe:LIST? 3–208
TRACe:POINts? (Query Only) 3–209

TRIGger[:A] Subsystem 3–211

TRIGger:ATRigger
TRIGger:ATRigger? 3–211
TRIGger:COUPling
TRIGger:COUPling? 3–212
TRIGger:COUPling:<preset> 3–213
TRIGger:DEFine? (Query Only) 3–214
TRIGger:DELay
TRIGger:DELay? 3–215
TRIGger:FILTer[:LPASs]
TRIGger:FILTer[:LPASs]? 3–216
TRIGger:FILTer:HPASs
TRIGger:FILTer:HPASs? 3–217
TRIGger:FILTer:NREJect
TRIGger:FILTer:NREJect? 3–218

www.valuetronics.com

TRIGger:HOLDoff:TIME
TRIGger:HOLDoff:TIME? 3–219
TRIGger:LEVel
TRIGger:LEVel? 3–220
TRIGger:SLOPe
TRIGger:SLOPe? 3–221
TRIGger:SOURce
TRIGger:SOURce? 3–222
TRIGger:TYPE
TRIGger:TYPE? 3–223
TRIGger:B Subsystem 3–225
TRIGger:B:COUPling
TRIGger:B:COUPling? 3–225
TRIGger:B:COUPling:<preset> 3–226
TRIGger:SEQuence2:DEFine? (Query Only) 3–227
TRIGger:B:DELay
TRIGger:B:DELay? 3–228
TRIGger:B:ECOunt
TRIGger:B:ECOunt? 3–229
TRIGger:B:FILTer[:LPASs]
TRIGger:B:FILTer[:LPASs]? 3–230
TRIGger:B:FILTer:HPASs
TRIGger:B:FILTer:HPASs? 3–231
TRIGger:B:FILTer:NREJect
TRIGger:B:FILTer:NREJect? 3–232
TRIGger:B:LEVel
TRIGger:B:LEVel? 3–233
TRIGger:B:SLOPe
TRIGger:B:SLOPe? 3–234
TRIGger:B:SOURce
TRIGger:B:SOURce? 3–235
TRIGger:PULSe Subsystem 3–237
TRIGger:PULSe:CLASs
TRIGger:PULSe:CLASs? 3–237
TRIGger:PULSe:GLITch:POLarity
TRIGger:PULSe:GLITch:POLarity? 3–238
TRIGger:PULSe:GLITch:QUALify
TRIGger:PULSe:GLITch:QUALify? 3–239
TRIGger:PULSe:GLITch:WIDTh
TRIGger:PULSe:GLITch:WIDTh? 3–240
TRIGger:PULSe:SOURce
TRIGger:PULSe:SOURce? 3–241
TRIGger:PULSe:THReshold
TRIGger:PULSe:THReshold? 3–242
TRIGger:PULSe:WIDTh:HLIMit
TRIGger:PULSe:WIDTh:HLIMit? 3–243
TRIGger:PULSe:WIDTh:LLIMit
TRIGger:PULSe:WIDTh:LLIMit? 3–244
TRIGger:PULSe:WIDTh:POLarity
TRIGger:PULSe:WIDTh:POLarity? 3–245
TRIGger:PULSe:WIDTh:QUALify
TRIGger:PULSe:WIDTh:QUALify? 3–246
VOLTage Subsystem 3–247

www.valuetronics.com

VOLTage:RANGe[:UPPer]
VOLTage:RANGe[:UPPer]? 3–248
VOLTage:RANGe:LOWer
VOLTage:RANGe:LOWer? 3–249
VOLTage:RANGe:OFFSet
VOLTage:RANGe:OFFSet? 3–251
VOLTage:RANGe:PTPeak
VOLTage:RANGe:PTPeak? 3–253
IEEE 488.2 Common Commands 3–255
*CAL? (Query Only) 3–255
*CLS 3–256
*ESE
*ESE? 3–257
*ESR? (Query Only) 3–258
*IDN? (Query Only) 3–259
*LRN? (Query Only) 3–260
*OPC
*OPC? 3–261
*OPT? (Query Only) 3–261
*PUD
*PUD? 3–262
*RCL 3–263
*RST 3–264
*SAV 3–264
*SRE
*SRE? 3–265
*STB? (Query Only) 3–266
*TRG 3–267
*TST? (Query Only) 3–268
*WAI 3–269

Status and Events

Status and Events 4–1
Registers 4–1
Queues 4–8
Status and Event Reporting Process 4–10
Synchronization Methods 4–11
Error Messages 4–12

Appendices

Appendix A: Specifications A–1
Product Description A–1
Specification Tables A–2
Appendix B: Algorithms B–1
Measurement Variables B–1
Measurement Algorithms B–4
Differentiation Algorithm B–13
Integration Algorithm B–13
Smooth Algorithm B–14
Appendix C: ASCII Character Chart C–1

www.valuetronics.com

Appendix D: SCPI Conformance Information D–1

Glossary and Index

www.valuetronics.com

List of Figures

Figure 1–1: Logical Address Switches 1–3
Figure 1–2: Module Retainer Screws and Ejector Mechanism 1–6
Figure 1–3: Example VXIbus Test System for the Incoming Inspection Procedure 1–15
Figure 1–4: Time Reference Test Setup for Functional Tests 1–17
Figure 1–5: Voltage Reference Test Setup for Functional Tests 1–19
Figure 2–1: TVS600 Front Panel 2–3
Figure 2–2: Instrument model showing root level nodes 2–8
Figure 2–3: Digital acquisition showing sampling and digitizing steps 2–9
Figure 2–4: Digitizer configuration 2–9
Figure 2–5: Digital sampling showing the sample interval, trigger event, and pretrigger samples 2–10
Figure 2–6: Positioning the waveform record relative to the trigger point 2–12
Figure 2–7: Real-time acquisition 2–13
Figure 2–8: The acquisition cycle 2–14
Figure 2–9: Comparison between auto-advance and INIT acquisition looping 2–15
Figure 2–10: Setting vertical range and offset of input channels 2–20
Figure 2–11: The Arm/Trigger cycle 2–21
Figure 2–12: Slope and level define the trigger event 2–23
Figure 2–13: Trigger holdoff time ensures valid triggering 2–24
Figure 2–14: Delayed runs after main 2–26
Figure 2–15: Delayed triggerable 2–26
Figure 2–16: How the delayed triggers work 2–27
Figure 2–17: Trigger holdoff time with trigger delay time 2–28
Figure 2–18: Parameters for pulse triggering 2–29
Figure 2–19: PATH definition for SCPI calculation model 2–32
Figure 2–20: Zero phase reference point in FFT phase records 2–41
Figure 2–21: How aliased frequencies corrupt an FFT transform . . . 2–42
Figure 2–22: Windowing the FFT time domain record 2–43
Figure 2–23: FFT windows and bandpass characteristics 2–45
Figure 2–24: Parameters for the four digital filters 2–47
Figure 2–25: Two methods of setting BPASs and NOTCh filters 2–48
Figure 2–26: Rejection level and transition slope for the digital filter 2–49

www.valuetronics.com

Figure 2–27: Binary transfer format 2–56
Figure 2–28: VXIbus Connectors P1 and P2 2–65
Figure 2–29: Pin assignments for the SERIAL INTERFACE (RS-232) connector 2–70
Figure 2–30: Initial Equipment Setup for the Tutorial 2–72
Figure 2–31: Standard Event and Status Byte Registers 2–81
Figure 3–1: Example of SCPI Subsystem Hierarchy Tree 3–1
Figure 3–2: Example of Abbreviating a Command 3–3
Figure 3–3: Example of Chaining Commands and Queries 3–4
Figure 3–4: Example of Omitting Root and Lower-level Nodes in Chained Message 3–4
Figure 3–5: Instrument Model Showing Root-level Nodes and Subsystems 3–27
Figure 3–6: AADVance Subsystem Hierarchy 3–29
Figure 3–7: AADVance Subsystem Functional Model 3–29
Figure 3–8: ARM Subsystem Hierarchy 3–35
Figure 3–9: AVERage Subsystem Hierarchy 3–39
Figure 3–10: AVERage Subsystem Functional Model 3–39
Figure 3–11: CALCulate Subsystem Hierarchy 3–43
Figure 3–12: CALCulate Subsystem Functional Model 3–43
Figure 3–13: CALibration Subsystem Hierarchy 3–97
Figure 3–14: FORMat Subsystem Hierarchy 3–101
Figure 3–15: FUNCtion and DATA Hierarchy 3–105
Figure 3–16: FUNCtion and DATA Functional Model 3–105
Figure 3–17: INITiate and ABORt Subsystem Hierarchy 3–117
Figure 3–18: INPut Subsystem Hierarchy 3–123
Figure 3–19: INPut Subsystem Functional Model 3–123
Figure 3–20: MEMory Subsystem Hierarchy 3–131
Figure 3–21: OUTPut Subsystem Hierarchy 3–137
Figure 3–22: ROSCillator Subsystem 3–149
Figure 3–23: ROSCillator Subsystem Functional Model 3–149
Figure 3–24: STATus Subsystem Hierarchy 3–151
Figure 3–25: SWEep Subsystem Hierarchy 3–169
Figure 3–26: SWEep Subsystem Functional Model 3–169
Figure 3–27: SYSTem Subsystem Hierarchy 3–179
Figure 3–28: TEST Subsystem Hierarchy 3–197
Figure 3–29: TRACe Subsystem Hierarchy 3–201
Figure 3–30: Functions of the TRACe Subsystem 3–201
Figure 3–31: TRIGger:A (SCPI SEQuence1) Subsystem Hierarchy . 3–211

www.valuetronics.com

Figure 3–32: TRIGger:B (SCPI SEQuence2) Subsystem Hierarchy . 3–225
Figure 3–33: TRIGger:PULSe Subsystem Hierarchy 3–237
Figure 3–34: VOLTage Subsystem Hierarchy 3–247
Figure 3–35: VOLTage Subsystem Functional Model 3–247
Figure 3–36: IEEE 488.2 Common Command Syntax 3–255
Figure 4–1: SCPI & IEEE 488.2 Status and Event Registers 4–2
Figure 4–2: Status and Event Reporting Process 4–10
Figure B–1: MCross Calculations B–4
Figure B–2: Fall time B–7
Figure B–3: Rise Time B–11
Figure B–4: Transfer function H(f) for an ideal bandpass filter B–15
Figure B–5: Transfer function for an ideal lowpass filter B–16
Figure B–6: Using a rectangular window to truncate the data from Figure B–5 to a finite number of points B–17
Figure B–7: Lowpass filter transfer function obtained by truncating the impulse response to just a few points B–18
Figure B–8: Using more points in the Lowpass filter results in a steeper transition at the cutoff frequency B–18
Figure B–9: Using many more points in the Lowpass filter results in a quicker transition but a minimum attenuation of 21 dB B–19
Figure B–10: Kaiser window with 200 points and b = 1, 5 and 20 . . . B–20
Figure B–11: Compare this result with Figure B–9 with the same number of points but a rectangular window B–21
Figure B–12: Filter specifications for a lowpass filter B–21
Figure B–13: Filter specifications for a bandpass filter B–23
Figure B–14: Record resulting from convolving the filter impulse response with the waveform record B–24
Figure B–15: Filter test signal with a 125 MHz signal modulating a 10 MHz signal B–25
Figure B–16: Test signal after being filtered with a lowpass filter . . . B–25
Figure B–17: View of the filtered record showing the first 5% of the filtered data B–26

www.valuetronics.com

List of Tables

Table 1–1: Factory Default RS-232 Settings 1–14
Table 2–1: Trigger Sources and Compatibility 2–22
Table 2–2: Measurement Definitions 2–35
Table 2–3: Measurement Parameters 2–37
Table 2–4: Commands for Fast Data Channel and Associated Functions 2–50
Table 2–5: Waveform Preambles and Their Formats 2–58
Table 2–6: Parameter Definitions for Preamble Elements 2–61
Table 2–7: Trigger output lines and their default assignments 2–64
Table 2–8: Left Slot P1 Pinout 2–66
Table 2–9: Left Slot P2 Pinout 2–67
Table 2–10: Right Slot P2 Pinout 2–68
Table 3–1: Parameter Types Used in Syntax Descriptions 3–2
Table 3–2: BNF Symbols and Meanings 3–6
Table 3–3: Auto-advance Commands 3–9
Table 3–4: Abort Commands 3–9
Table 3–5: Arm Commands 3–10
Table 3–6: Average Commands 3–10
Table 3–7: Calculate Commands 3–10
Table 3–8: Calibration Commands 3–13
Table 3–9: Data Commands 3–13
Table 3–10: Format Commands 3–14
Table 3–11: Function Commands 3–14
Table 3–12: Initiate Commands 3–15
Table 3–13: Input Commands 3–15
Table 3–14: Memory Commands 3–16
Table 3–15: Output Commands 3–16
Table 3–16: Reference Oscillator Commands 3–17
Table 3–17: Status Commands 3–18
Table 3–18: Sweep Commands 3–19
Table 3–19: System Commands 3–20
Table 3–20: Test Commands 3–21
Table 3–21: Trace Commands 3–21
Table 3–22: Trigger Commands 3–22
Table 3–23: Voltage Commands 3–24

www.valuetronics.com

Table 3–24: IEEE 488.2 Common Commands 3–25
Table 3–25: Waveform Measurement Definitions 3–78
Table 3–26: The Operation Status Register 3–151
Table 3–27: The Questionable Status Register 3–160
Table 3–28: Effects of :PRESet on Serial Port Parameters 3–187
Table 3–29: The Standard Event Status Register 3–258
Table 3–30: The Status Byte Register 3–266
Table 4–1: The Status Byte Register 4–3
Table 4–2: The Standard Event Status Register 4–4
Table 4–3: The Operation Status Register 4–5
Table 4–4: Control Registers for the Operation Status Register 4–6
Table 4–5: The Questionable Status Register 4–7
Table 4–6: Control Registers for the Questionable Status Register . 4–7
Table 4–7: Commands Associated with the Status Queue 4–8
Table 4–8: Command Error Messages (Bit 5 in Standard Event Status Register) 4–12
Table 4–9: Execution Error Messages (Bit 4 in Standard Event Status Register) 4–13
Table 4–10: Device Dependent Error Messages (Bit 3 in Standard Event Status Register) 4–14
Table 4–11: System Events 4–14
Table 4–12: Execution Warning Messages (Bit 3 in Standard Event Status Register) 4–14
Table A–1: Comparison of Product Features A–1
Table A–2: Signal Acquisition System A–2
Table A–3: Timebase System A–5
Table A–4: Trigger System A–6
Table A–5: Front Panel Connectors A–9
Table A–6: VXI Interface A–11
Table A–7: Power Distribution and Data Handling A–12
Table A–8: Environmental A–12
Table A–9: Certifications and Compliances A–13
Table A–10: Mechanical A–14
Table C–1: ASCII Code Chart C–1
Table D–1: SCPI Conformance Information D–1

www.valuetronics.com

General Safety Summary

Review the following safety precautions to avoid injury and prevent damage to this product or any products connected to it.

Only qualified personnel should perform service procedures.

While using this product, you may need to access other parts of the system. Read the *General Safety Summary* in other system manuals for warnings and cautions related to operating the system.

Injury Precautions

Avoid Electric Overload

To avoid electric shock or fire hazard, do not apply a voltage to a terminal that is outside the range specified for that terminal.

Avoid Electric Shock

To avoid injury or loss of life, do not connect or disconnect probes or test leads while they are connected to a voltage source.

Ground the Product

This product is indirectly grounded through the grounding conductor of the mainframe power cord. To avoid electric shock, the grounding conductor must be connected to earth ground. Before making connections to the input or output terminals of the product, ensure that the product is properly grounded.

Do Not Operate Without Covers

To avoid electric shock or fire hazard, do not operate this product with covers or panels removed.

Use Proper Fuse

To avoid fire hazard, use only the fuse type and rating specified for this product.

Do Not Operate in Wet/Damp Conditions

To avoid electric shock, do not operate this product in wet or damp conditions.

Do Not Operate in an Explosive Atmosphere

To avoid injury or fire hazard, do not operate this product in an explosive atmosphere.

www.valuetronics.com

Product Damage Precautions

Provide Proper Ventilation

To prevent product overheating, provide proper ventilation.

Do Not Operate With Suspected Failures

If you suspect there is damage to this product, have it inspected by qualified service personnel.

Safety Terms and Symbols

Terms in This Manual

These terms may appear in this manual:

WARNING. *Warning statements identify conditions or practices that could result in injury or loss of life.*

CAUTION. *Caution statements identify conditions or practices that could result in damage to this product or other property.*

Terms on the Product

These terms may appear on the product:

DANGER indicates an injury hazard immediately accessible as you read the marking.

WARNING indicates an injury hazard not immediately accessible as you read the marking.

CAUTION indicates a hazard to property including the product.

Symbols on the Product

The following symbols may appear on the product:

DANGER
High Voltage

Protective Ground
(Earth) Terminal

ATTENTION
Refer to Manual

Double
Insulated

www.valuetronics.com

Certifications and Compliances

Safety Certification of Plug-in or VXI Modules

For modules (plug-in or VXI) that are safety certified by Underwriters Laboratories, UL Listing applies only when the module is installed in a UL Listed product.

For modules (plug-in or VXI) that have cUL or CSA approval, the approval applies only when the module is installed in a cUL or CSA approved product.

Compliances

Consult the product specifications for Overvoltage Category and Safety Class.

Overvoltage Category

Overvoltage categories are defined as follows:

CAT III: Distribution level mains, fixed installation

CAT II: Local level mains, appliances, portable equipment

CAT I: Signal level, special equipment or parts of equipment, telecommunication, electronics

www.valuetronics.com

Preface

This manual describes the capabilities of the TVS600 Series Waveform Analyzers and how to use them in a programming environment. The TVS600 Series Waveform Analyzers, also referred to as the waveform analyzer, are controlled through the use of SCPI (Standard Commands for Programmable Instruments) derived commands and IEEE 488.2 Common Commands. This manual describes how to use these commands to configure the waveform analyzer and access information generated by it or stored within it.

About This Manual

This manual is composed of the following sections:

- *Getting Started* shows you how to configure and install your waveform analyzer and provides an Incoming Inspection procedure.
- *Operating Basics* describes the front panel connectors, provides a functional overview of product functions and presents a Tutorial on programming the waveform analyzer.
- *Syntax and Commands* defines the syntax used in command descriptions, presents a list of all command subsystems, and presents detailed descriptions of all programming commands.
- *Status and Events* describes how the status and Events Reporting system operates and presents a list of all system errors.
- *Appendices* provides additional information including the Specifications, Calculation algorithms, and SCPI conformance information.

Related Manuals

The following documents are also available for the TVS600 Series Waveform Analyzers:

- The *TVS600 Series Waveform Analyzers Reference* (Tektronix part number 070-9284-XX) provides an alphabetical listing of the programming commands. This manual is a standard accessory.
- The *TVS600 Series Waveform Analyzers Service Manual* (Tektronix part number 070-9285-XX) describes how to service the instrument to the module level. This optional manual must be ordered separately.

www.valuetronics.com

Getting Started

This chapter provides information that will help you get started using your TVS600 Waveform Analyzer. This Chapter describes the waveform analyzer and its options. This chapter also shows you how to configure and install the waveform analyzer and the VXI*plug&play* software included with the product. This chapter concludes with the *Incoming Inspection* procedure which verifies basic operation and functionality.

Product Description

The TVS600 Series Waveform Analyzers are a family of C-size, VXI modules that provide high-speed signal acquisition, real-time measurements and Fast Data Channel data transfer. Key features include:

- Fully programmable with an extensive SCPI command set and message based interface
- Fast rearm capability with auto-advance acquisition mode
- Sample intervals as short as 200 ps/point with real time sampling
- Acquisition modes such as auto-advance, envelope, and average
- A full compliment of internal triggering modes, including main, delayed, edge and pulse triggering
- Trigger sources from input channels, front panel External Trigger, and VXI backplane triggers
- Standard acquisition memory that allows 15,000 samples to be taken in realtime acquisition mode and 30,000 samples in extended-realtime acquisition mode.
- Very fast calculation system for measurements, waveform math, and waveform transforms.

Accessories

This section lists the standard and optional accessories available for the TVS600 Series Waveform Analyzers.

In addition to the options described in this section, Tektronix offers maintenance options that cover calibration and repair services. Contact your Tektronix representative for details.

www.valuetronics.com

Standard Accessories

The following accessories are shipped with the waveform analyzer:

- *TVS600 Series Waveform Analyzers User Manual* (Tektronix part number 070-9283-XX)
- *TVS600 Series Waveform Analyzers Reference Manual* (Tektronix part number 070-9284-XX)
- TVS600 Series VXI*plug&play* Instrument Drivers (Tektronix part number 063-1874-XX)

Optional Accessories

The following accessories are available for use with the waveform analyzer:

- *TVS600 Series Waveform Analyzers Service Manual* (Tektronix part number 070-9285-XX)

Configuration

You must configure the VXIbus module and the VXIbus mainframe before installing the module. To configure the waveform analyzer, set its logical address on the VXIbus. To configure your VXIbus mainframe, you set the Bus Grant and Interrupt Acknowledge jumpers. This section describes how to perform the necessary configuration.

Setting the Logical Address

Every module within a VXIbus system must have a unique logical address; no two modules can have the same address. On the waveform analyzer, you rotate two switches on the rear panel to select the logical address. Refer to Figure 1–1 for the switch locations. You can select a static address or Dynamic Auto Configuration. The default address is FF hexadecimal, which selects Dynamic Auto Configuration.

www.valuetronics.com

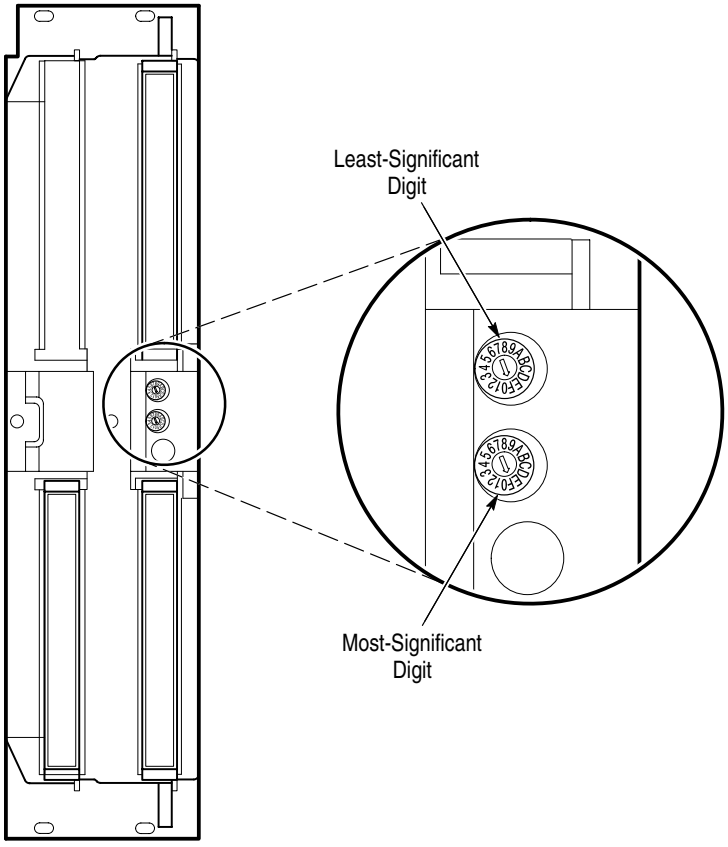


Figure 1–1: Logical Address Switches

Static Logical Address Static logical address selections set the address to a fixed value. The range for static addresses is from 01 to FE hexadecimal (1 to 254 decimal). A static logical address ensures that the waveform analyzer address remains fixed for compatibility with systems that require a specific address value. Remember that each device within your system must have a unique address to avoid communication problems. The factory default address is FF hexadecimal, which selects Dynamic Auto Configuration.

Dynamic Auto Configuration. With Dynamic Auto Configuration selected (hexadecimal FF or decimal 255), the system resource manager automatically sets the address to the next available value in your system. For example, if you already have devices set to addresses 01 and 02, the resource manager might automatically assign address 03 to the waveform analyzer at power on.

www.valuetronics.com

Configuring the VXIbus Mainframe

This section describes how to install the waveform analyzer into a Tektronix VXIbus mainframe. If you are installing the waveform analyzer into a different mainframe, refer to the instruction manual for that mainframe for any pertinent installation or capacity information.

Voltage, Current, and Cooling Requirements. You will find voltage, current, and cooling requirements for the waveform analyzer in *Appendix A*, *Specifications* at the following locations:

- Voltage and current requirements, see Table A–7 on page A–12
- Cooling requirements, see Table A–8 on page A–12

These requirements also appear on the left cover of the waveform analyzer. Be sure your mainframe can supply adequate current and cooling to the waveform analyzer and the other modules you plan to install into the same mainframe.

WARNING. *Shock hazards exist due to high currents within the mainframe compartment. Do not change configuration of the Bus Grant and Interrupt Acknowledge jumpers unless you are qualified to do so. Consult your VXI mainframe manual for safety warnings and configuration information.*

Jumper Settings. Many VXIbus mainframes contain daisy-chain jumper straps that you must configure before installing the waveform analyzer. The jumper straps, located beside the P1 connectors, set up the Bus Grant (BG0-BG3) and Interrupt Acknowledge (IACK) signals. If you are using a Tektronix mainframe, the names of the jumper straps (BG0-BG3 and IACK) are often printed on the circuit board facing the front of the mainframe. Access these jumpers from the front of the mainframe.

Some VXIbus mainframes, such as the Tektronix VX1410 Intelliframe™, have an auto-configurable backplane with no mechanical jumpers. You do not need to set jumpers on these VXIbus mainframes.

If your VXIbus mainframe has jumper straps, set the IACK and BG0–BG3 jumpers for the waveform analyzer as shown below:

- Remove the jumper straps for the left-most slot in which you will install the waveform analyzer (retain the strap for future configurations).
- Keep the jumpers for the right-most slot installed in the mainframe.

For example, if you want to install the waveform analyzer into the third and fourth mainframe slots, remove all jumper straps for the third slot and install them into the fourth slot.

www.valuetronics.com

Installation

This section describes how to install the waveform analyzer into a VXIbus mainframe and how to install the product software. For hardware configuration information refer to *Configuration* on page 1–2.

Hardware Installation

You may install the waveform analyzer into any empty slot in the mainframe except Slot 0. Be sure to set the logical address before installation (see *Setting the Logical Address* on page 1–3).

CAUTION. *If you install the waveform analyzer into a D-size mainframe, be sure to connect the P1 and P2 connectors of the module to the P1 and P2 connectors on the mainframe. Electrical damage will result when connecting the P1 and P2 connectors on the module to the P2 and P3 connectors on the mainframe.*

To avoid damage, look for bent pins on P1 and P2 before installation.

Use the following installation procedure and Figure 1–2 to install the waveform analyzer into the mainframe:

1. On the mainframe, set the power ON/STANDBY switch to OFF.
2. Insert the waveform analyzer into the mainframe top and bottom module guides and push it partially into the mainframe (Figure 1–2). Then slide the waveform analyzer into the mainframe as far as it will go without forcing it.
3. Be sure the front panel is flush with the front of the mainframe chassis. If so, use a flat-blade screwdriver to install the top and bottom retainer screws. Alternately tighten the screws, applying only a few turns at a time to fully seat the module. If it is not flat, remove the module and check for mechanical problems with the VXIbus connector or the module enclosure.

www.valuetronics.com

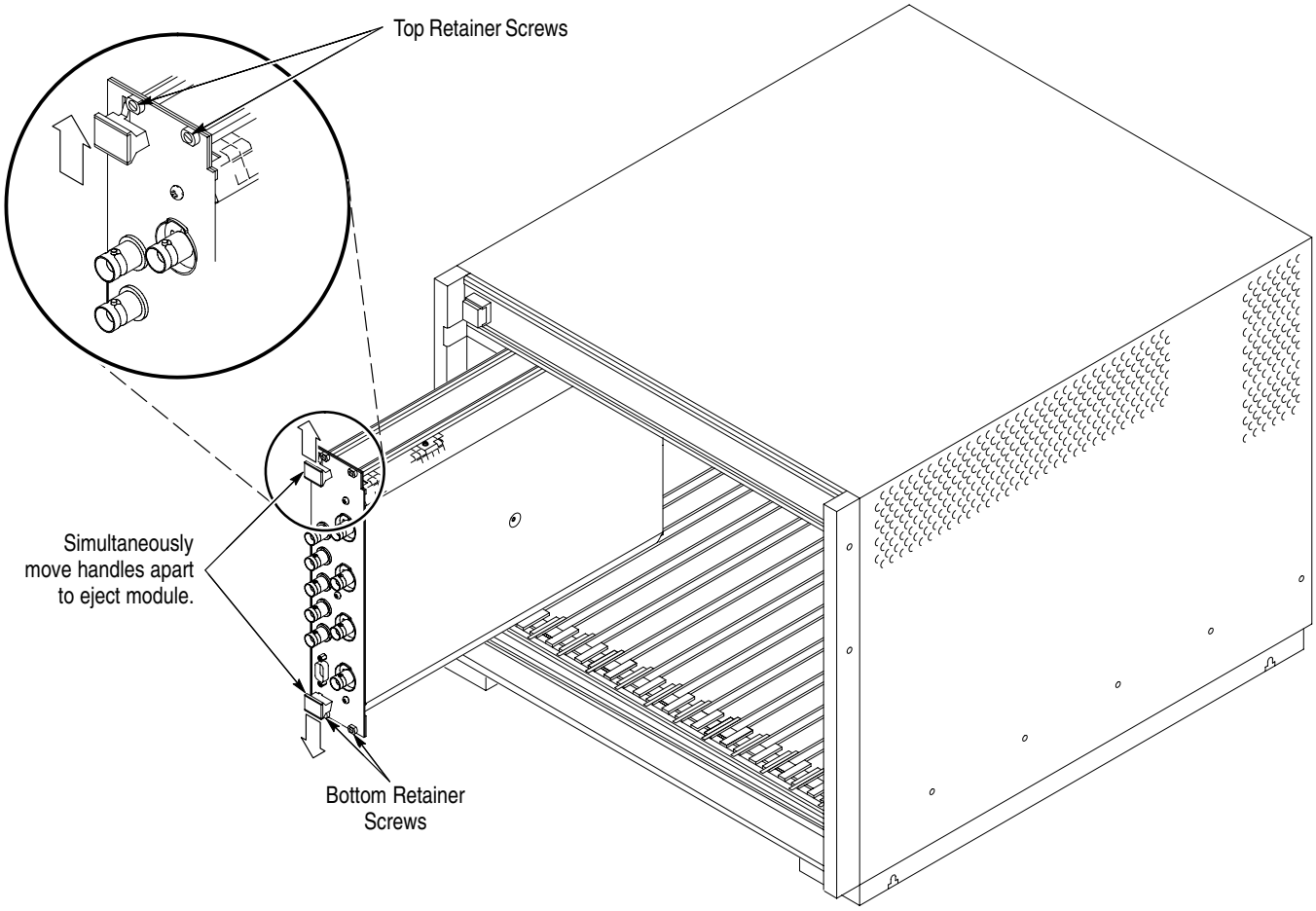


Figure 1–2: Module Retainer Screws and Ejector Mechanism

Removal from VXIbus Mainframe

Use the following procedure to remove the waveform analyzer from a Tektronix VXIbus mainframe. If you are using a different mainframe, you may need to modify this procedure. Refer to your mainframe manual for instructions.

1. On the mainframe, set the power ON/STANDBY switch to OFF.
2. Using a flat-blade screwdriver, loosen the top and bottom retainer screws (Figure 1–2).
3. Grasp both handles of the waveform analyzer. At the same time, move the top handle upward and the bottom handle downward to eject the waveform analyzer.
4. Pull the waveform analyzer out of the mainframe.

www.valuetronics.com

Power-On Procedure

This section describes how to check that your waveform analyzer powers up properly. Be certain that the waveform analyzer is properly configured before applying power. Refer to *Configuration* on page 1–2.

1. Before applying power to your waveform analyzer and VXIbus system, check the following items:
   - Ensure that all VXIbus modules are properly installed.
   - Check that all connected signal sources are set to an appropriate output level to avoid damaging inputs on your waveform analyzer.
   - Power up your controller, if it is external to the VXIbus mainframe.
2. Apply power to your VXIbus mainframe.

   During power on, the waveform analyzer performs a self test to verify functionality. The self test requires approximately five seconds to complete. The front-panel ARM'D and TRIG'D indicators blink during the self test. After testing successfully completes, the front panel READY indicator should be on (green when lit).

***NOTE**. The READY indicator does not light if the power-on self test fails.*

   If your waveform analyzer does not pass the power on self test (READY indicator does not light), power off the VXIbus mainframe and check that all modules are fully seated in the VXIbus mainframe. If the problem persists, remove the waveform analyzer and check that its address setting does not conflict with another module. If failures continue, the module might require service.

3. Once the power-on self tests are complete, the waveform analyzer recalls the settings that were active when the waveform analyzer was powered off. There is one exception: input protection is set ON, its reset value. Power-on settings are stored in nonvolatile memory.

Most parameters have a default value that you can restore by sending the `*RST` command.

www.valuetronics.com

Software Installation

This section describes how to install the VXI*plug&play* WIN Framework software that accompanies the TVS600 Series Waveform Analyzers User manual (this manual). The product software includes the TVS600 Soft Front Panel application and TVS600 device driver support library.

Description of the TVS600 VXI*plug&play* Software

The TVS600 VXI*plug&play* software provides a virtual front panel and a driver that supports several programming environments. The TVS600 Soft Front Panel (TKTVS600.EXE) is a tool that you can use to confirm that the waveform analyzer is connected to the VXIbus. You can also access and control some of its acquisition and measurement features. For detailed descriptions of the TVS600 Soft Front Panel and the device drivers, refer to the Windows on-line help documents present with these applications.

The TVS600 VXI*plug&play* software includes a driver function library that provides limited control of many instrument operations. The device driver supports the following programming environments:

- National Instruments LabWindows/CVI for Windows
- National Instruments LabVIEW for Windows
- Microsoft Visual BASIC
- Microsoft Visual C
- Borland Turbo C
- Hewlett Packard HP-VEE

Other programming languages may also be used if they are compatible with Windows 16 bit DLLs.

Driver Source Code. The driver library includes source code for the driver to help you better understand driver functions and to allow you to compile them in your programming environment. The source code conforms to ASCII C which may provide portability to other computer environments.

On-line Help. The driver library includes two Windows help documents: one for the Soft Front Panel (GITVS600.HLP) and a separate one for the device drivers (TKTVS600.HLP). Before proceeding, review the release notes included in the README.TXT file located in [drive]:\VXIPNP\WIN\TKTVS600.

www.valuetronics.com

Requirements for the VXI*plug&play* Software

The following system requirements should be met before installing the TVS600 VXI*plug&play* software.

- WIN Framework Version 3.0 or better (VISA.DLL). This software is provided by the manufacturer of VXI*plug&play* WIN Framework Ver. 3 compliant controllers and is required to run the Soft Front Panel software and the driver library.
- 3 Mbyte of disk space
- Windows 3.1

Installing the TVS600 VXI*plug&play* Software

The TVS600 VXI*plug&play* software, included with your waveform analyzer, is located on two DOS format, high-density, 3 1/2 inch floppy disks. For a list of the installed files, refer to *Installed Files* on page 1–10.

To install the VXI*plug&play* software from Windows, perform the following steps:

1. Insert disk 1 in a 3 1/2 inch disk drive.
2. Start the Program Manager and select Run in the File menu.
3. In the field below Command Line:, enter [drive]\setup. [drive] represents the actual drive name where the VXI*plug&play* disk resides. If you use the A drive, then the command to enter is A:\setup.
4. Click on OK to start the setup program. Setup loads all files on disk 1 to your C drive, unless you specify another, then prompts you to insert disk 2. If you previously installed a Tektronix VXI*plug&play* disk on your system, then you might not be prompted for disk 2.
5. When installation completes you are prompted to either run the soft front panel application or exit to Windows. If you have not previously installed the WIN Framework software, then you should exit to Windows. Install WIN Framework software before using the VXI*plug&play* software.

Setup creates a VXIPNP product group and installs the Soft Front Panel application icon named TKTVS600 FRONT PANEL. For a list of the installed files, refer to *Installed Files* on page 1–10.

Release Notes

The installation disk installs software release notes in an ASCII file named README.TXT. The README file contains additional installation and operation information that supercedes other product documentation. Installation places the README file at C:\VXIPNP\WIN\TKTVS600\README.TXT.

To view the README file, open the Notepad Windows accessory and select the appropriate path to open the README document.

www.valuetronics.com

Using the TVS600 VXI*plug&play* Software

The product software includes the TVS600 Soft Front Panel application and a library of driver functions. Documentation for the TVS600 Soft Front Panel is included as a Windows Help document, which is available once you start the application. Documentation for the library functions is contained in a Windows Help file named TKTVS600.HLP.

To start the TVS600 Soft Front Panel software, open the VXIPNP product group and double click on the TKTVS600 FRONT PANEL icon. At start up, the software finds the installed TVS600 modules and asks you to pick one or all for connection. If you select All, then all TVS600 waveform analyzers are connected though only one waveform analyzer. is active and in communication with the soft front panel. For instructions on using the soft front panel, select Help –> Front Panel Operation.

Installed Files

The setup.exe command installs all files within the default directory path <drive>:\VXIPNP\WIN unless you specify another path. If you previously installed VXI*plug&play* software and specified a different location for files, the setup script will find that path (vpnppath) in your autoexec.bat and use it instead of the default.

TVS600 Soft Front Panel Files

- \TKTVS600\TKTVS600.EXE — SFP executable
- \TKTVS600\GUTVS600.UIR — SFP main panel
- \TKTVS600\STARTUP.UIR — SFP startup panel
- \TKTVS600\GITVS600.HLP — SFP help file

TVS600 CVI Driver Files

- \TKTVS600\TKTVS600.C — Driver source
- \TKTVS600\TKTVS600.H — Driver header
- \TKTVS600\TKTVS600.FP — Driver function panel
- \TKTVS600\TKTVS600.HLP — Driver Help file
- \TKTVS600\TKTVS600.DOC — Driver help document (text format)
- \BIN\TKTVS600.DLL — Driver dynamic link library
- \TKTVS600\TKTVS600.LIB — Microsoft Visual C (MSVC) import library

www.valuetronics.com

Note that the function library TKTVS600.LIB is not compatible with Borland C. You can create a separate import library that is compatible by using the Borland implib utility command: `implib TKTVS600.DLL`.

■ \TKTVS600\TKTVS600.DEF	DLL module definition file
■ \TKTVS600\TKTVS600.MAK	DLL MSVC Make file

Other Files.

■ C:\VXIPNP\KBASE\TKTVS600.KB	VXI*plug&play* knowledge base for all TVS600 products
■ \TKTVS600\TKTVS600.BAS	Visual Basic support
■ \TKTVS600\DSTVS600.HLP	On-line data sheet
■ \TKTVS600\README.TXT	Release notes

Soft Front Panel Support Files

If the CVI executable is not already installed on your system, it will be installed as listed. The Windows WIN.INI file is changed in this case to add the entry: cvirt3=C:\VXIPNP\WIN\BIN\CVI\CVIRT3.EXE. Note, if CVI is already installed, then setup.exe does not install the following files.

\BIN\CVI\CVIRT3.EXE	CVI executable
\BIN\CVI\BIN\CVIRT3.RSC	CVI data file
\BIN\CVI\BIN\MSGRT3.TXT	CVI data file
\BIN\CVI\FONTS\NISYSTEM.TTF	CVI font
\BIN\CVI\FONTS\NI7SEG.TTF	CVI font

www.valuetronics.com

Incoming Inspection Procedure

This section contains instructions for performing the *Incoming Inspection Procedure*. This procedure verifies that the waveform analyzer is operating correctly after shipment.

If the waveform analyzer fails any test within this section, the module may need service.

Description

The *Incoming Inspection Procedure* is divided into four parts:

- *Connect the VXIbus Test System* on page 1–15 provides instructions for setting up an example test system for this procedure
- *Self Tests* on page 1–15 provides instructions for performing the internal self tests
- *Functional Tests* on page 1–17 measures the time and amplitude reference signals at the REFERENCE OUTPUT connector
- *Self Cal* on page 1–20 provides instructions for performing internal self calibration

Purpose

This procedure verifies that the waveform analyzer is operating correctly.

Test Equipment

The *Incoming Inspection Procedure* requires the following test equipment:

- VXIbus mainframe, such as the Tektronix VX1410 IntelliFrame™
- Slot 0 controller, such as the National Instruments VXIpc-486 Series Model 566
- Computer peripherals for suggested Slot 0 controller: keyboard, monitor and, mouse
- Talk/listen software
- One coaxial cable with BNC connectors
- One dual banana to BNC adapter
- Frequency counter (measures 10 MHz at <0.0025% accuracy), such as the Hewlett Packard 5314A
- Digital multimeter (measures +8 V at 0.25% accuracy), such as the Fluke 8842A

www.valuetronics.com

You can perform these tests using any system components that allow you to send commands to the waveform analyzer.

About the System Setup

The *Incoming Inspection Procedure* is designed for a VXIbus system that contains an embedded Slot 0 controller. During the procedure you will communicate with the waveform analyzer using talk/listen software such as TVS600 Soft Front Panel software, a standard accessory. Figure 1–3 on page 1–15 shows a typical setup.

The procedure can be performed using other Slot 0 controllers and talk/listen software. If you use an alternate setup, you might need to reformat the commands to work with your talk/listen software.

You may choose to perform the *Incoming Inspection Procedure* using the SERIAL INTERFACE connector on the front panel of the waveform analyzer. To do so, connect a terminal or computer COM port directly to the SERIAL INTERFACE connector. For terminal connection, use a 9-pin to 25-pin serial interface cable (Tektronix part number 012-1380-XX). A similar cable might work to connect your computer. On the computer, run talk/listen software that provides PC terminal emulation.

The waveform analyzer recalls the last RS-232 settings from memory at power-on. Table 1–1 lists the factory default settings. Use the SYSTem:COMMunicate:SERial commands to modify the RS-232 parameters.

Table 1–1: Factory Default RS-232 Settings

Parameter	Default Setting
Baud rate	9600
Stop bits	1
Parity	None
DCD	Off
Echo	On
LBUF	On
Pace	XON
RTS	On
ERES	On

www.valuetronics.com

Connect the VXIbus Test System

Perform the following steps to connect a VXIbus test system similar to the example shown in Figure 1–3. If you use a different system controller and slot 0 controller, you may need to reformat the commands used in this procedure.

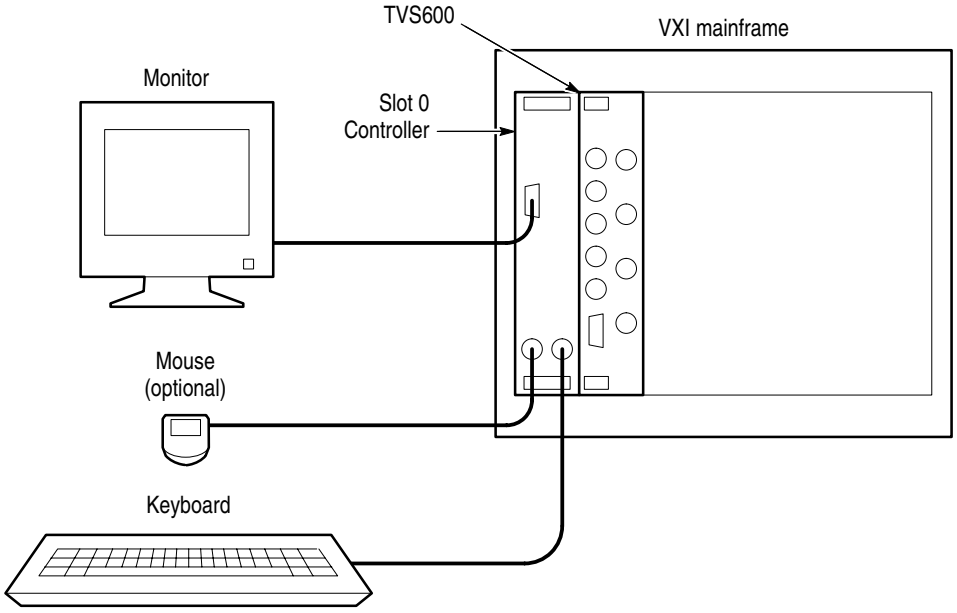


Figure 1–3: Example VXIbus Test System for the Incoming Inspection Procedure

1. Perform the *Power-On Procedure* located on page 1–7.
2. Allow a 20-minute warmup. Then perform the *Self Tests* procedure that begins on page 1–15.

Self Tests

The *Self Tests* use internal routines to verify that the waveform analyzer is functional. No test equipment is required.

1. Send the following command to execute the internal self test routines:

   TEST

2. Wait for the self tests to complete.

   - When running self tests, the READY, ACCESSED, ARM'D, and TRIG'D indicators blink
   - These tests take approximately 20 minutes to complete

www.valuetronics.com

3. Send the following query to check the self test results:

   `TEST:RES?`

4. Read the self test results.

   - A 0 result indicates all tests passed successfully
   - A –1 result indicates the self tests are still in progress; wait five minutes and send the `TEST:RES?` query again to read the test results
   - A 1000 to 2999 result indicates self test failures and a need for service on the failed module

5. Proceed to the *Functional Tests* to continue the *Incoming Inspection Procedure*.

www.valuetronics.com

Functional Tests

The following procedures test the internal time and voltage references of the waveform analyzer. You will need a frequency counter, digital multimeter, coaxial cable with BNC connectors, and a dual-banana to BNC adapter to perform the *Functional Tests*. See the equipment requirements on page 1–13.

Measure Time Reference

This procedure tests the accuracy of the internal time reference (10 MHz ± 1 kHz).

1. Connect a coaxial cable from the frequency counter input to the REFERENCE OUTPUT connector (see Figure 1–4).
2. Send the following command to initialize the waveform analyzer:

   ```
   *RST
   ```

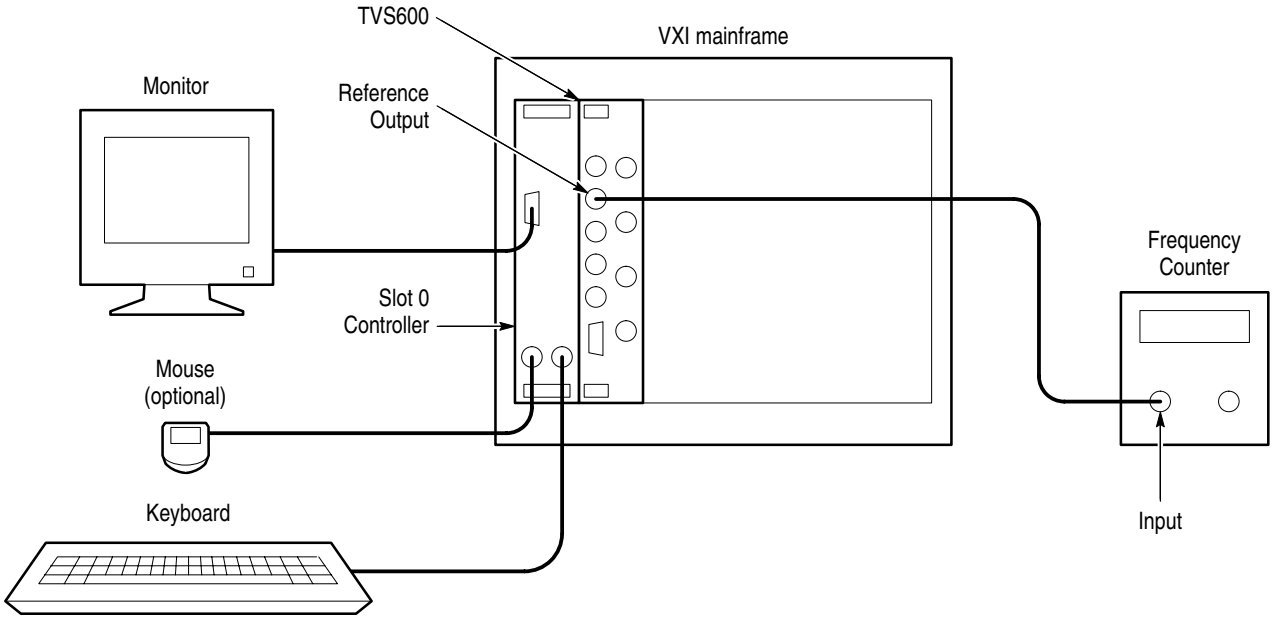


Figure 1–4: Time Reference Test Setup for Functional Tests

www.valuetronics.com

3. Select the following settings for the frequency counter:

Mode	Frequency
Trigger	Internal
Attenuation	X1

4. Send the following command to turn on the time reference:

   ```
   OUTP:REF:FUNC CLOC;:OUTP:REF ON
   ```

5. Check the frequency counter display. The frequency must be between 9,999,000 Hz and 10,001,000 Hz.

6. Disconnect the frequency counter from the REFERENCE OUTPUT connector.

7. Proceed to *Measure Voltage Reference* to continue the *Incoming Inspection Procedure*.

www.valuetronics.com

Measure Voltage Reference

This procedure tests the accuracy of the internal voltage reference (+8 V ± 1%).

1. Use a coaxial cable and dual-banana to BNC adapter to connect the digital multimeter input to the REFERENCE OUTPUT connector (see Figure 1–5).

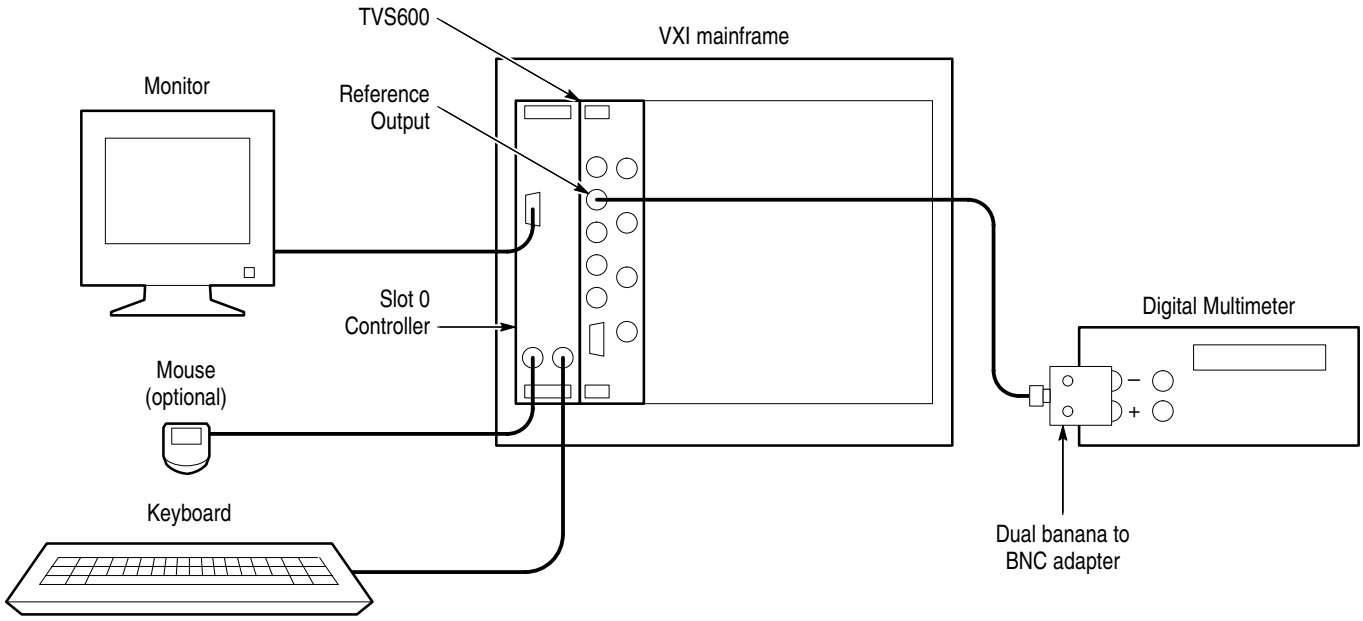


Figure 1–5: Voltage Reference Test Setup for Functional Tests

2. Send the following command to initialize the waveform analyzer:

   ```
   *RST
   ```

3. Select the following digital multimeter control settings:

Mode	DC Volts
Scale	20

4. Send the following command to turn on the voltage reference:

   ```
   OUTP:REF:FUNC VOLT;:OUTP:REF ON
   ```

5. Check the digital multimeter display. The voltage must be between +7.92 V and +8.08 V.
6. Disconnect the digital multimeter from the REFERENCE OUTPUT connector.
7. Proceed to *Self Cal* to continue the *Incoming Inspection Procedure*.

www.valuetronics.com

Self Cal

The *Self Cal* uses internal routines and the internal time and voltage reference to generate data such as gain and offset values, to optimize the waveform analyzer performance at the current ambient temperature. The data is stored in memory for use until you perform another self cal. No test equipment is required.

1. Send the following command to execute the internal self cal routines:

   ```
   CAL
   ```

2. Wait for the self cal to complete.

   - When running self cal, the READY, ACCESSED, ARM'D, and TRIG'D indicators blink
   - The self cal takes 15 to 30 minutes to complete

3. Send the following query to check the self cal results:

   ```
   CAL:RES?
   ```

4. Read the self cal results.

   - A 0 result indicates all tests passed successfully
   - A –1 result indicates the self cal is still in progress; wait five minutes and send the `CAL:RES?` query again to read the test results
   - A 2000 to 2999 result indicates self cal failures and a need for service on the failed module.

This completes the *Incoming Inspection Procedure*. If all tests passed, the waveform analyzer is ready for use. If any test failed, refer the module for service.

www.valuetronics.com

Operating Basics

This section contains information that can help you operate the TVS600 Waveform Analyzer more effectively. The section includes the following topics:

- *Connectors and Indicators,* on page 2–3, describes how to use the front panel connectors and what information the indicators convey.
- *Functional Overview,* on page 2–7, describes how the the main functions of the waveform analyzer operate and how to use the related commands to control these functions.
- *Instrument I/O,* on page 2–63, describes how the VXIbus and RS-232 interfaces are configured.
- *Tutorial,* on page 2–71, presents a step-by-step tutorial that shows you how to use the command set to configure the waveform analyzer to acquire waveforms and perform many of its main functions.

For information on configuring and installing your waveform analyzer, refer to *Getting Started.*

www.valuetronics.com

Connectors and Indicators

Figure 2–1 shows the connectors and indicators on the front panel of a four-channel waveform analyzer. The two-channel model looks and operates the same, but without the CH 3 and CH 4 inputs. Descriptions of each connector and indicator follow the illustration.

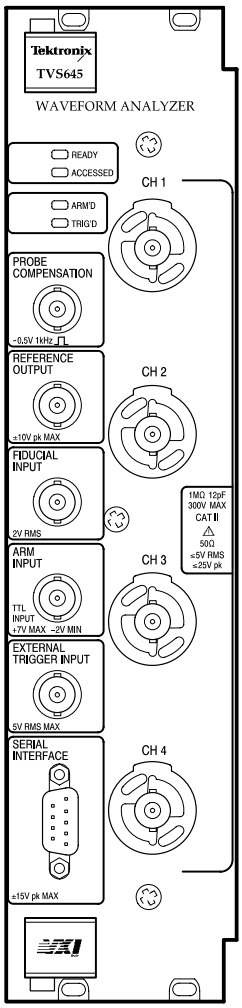


Figure 2–1: TVS600 Front Panel

CH1, CH2, CH3, and CH4 Channel Inputs. These BNC input connectors drive the vertical channel amplifiers and their dedicated digitizers. If your instrument is a TVS641 or TVS645, it has four input channels as shown in Figure 2–1. Otherwise, the TVS621 and TVS625 provide only the top two channels. Each

www.valuetronics.com

channel supports TekProbe Level 1 and Level 2 probes which offer many features, including signal offset.

The signal inputs can be set for 1 MΩ or 50 Ω input impedance. Coupling selections are DC, AC, and Ground. The Ground setting connects the internal amplifier to ground, but presents an open circuit impedance (≥ 500 kΩ) to your signal.

Every input channel has automatic input protection when set to 50 Ω coupling. When coupling is set to 50 Ω and an input signal exceeds the input range of an input channel, the coupling for that channel is set to 1 MΩ to protect the attenuator and digitizer.

For more information, refer to the description of the INPut commands starting on page 3–123.

READY Indicator. The green LED lights continuously after the waveform analyzer powers up and completes power-on diagnostics successfully. During normal operation, READY blinks when an error occurs that generates a status message.

ACCESSED Indicator. The yellow LED blinks on and then off under the following conditions:

- Each time communication with the waveform analyzer occurs
- When the Slot 0 resource manager asserts the Module Identification (MODID) line

ARM'D Indicator. The green LED lights when the waveform analyzer is armed and ready to accept a trigger signal.

TRIG'D Indicator. The green LED lights briefly when a trigger occurs for the current acquisition. The TRIG'D Indicator lights continuously whenever trigger events occur more often than three per second.

PROBE COMPENSATION. The BNC output provides a 1 kHz square wave signal for adjusting probe compensation. The signal amplitude is 0.5 V peak-to-peak into a 1 MΩ load. To enable the compensation signal, send the command OUTP:PCOM ON.

REFERENCE OUTPUT. The BNC output provides access to two internal references; the DC calibrator reference voltage or the time base clock. The precision calibrator reference voltage (VOLT) is +8.0 V. The time base clock (CLOC) is a 10 MHz square wave. Amplitude is ≥ 1 V into a 50 Ω load. To select a reference signal, send the command OUTP:REF:FUNC [CLOC|VOLT]. To enable the

www.valuetronics.com

selected signal to the REFERENCE OUTPUT connector, send the command OUTP:REF ON.

FIDUCIAL INPUT. The BNC input provides a way to add a signal component to the Channel 1 input signal. Adding a common timing signal to Channel 1 on several instruments provides a way to improve cross timing between multiple instruments. The input range is ±1 V. The input resistance presents 0.01 μF in series with 50 Ω.

ARM INPUT. The BNC input allows you to arm the acquisition system by grounding the center lead. The ARM input is level sensitive and is not latched. An internal pull up resistor connected to +5 V maintains a high level until you ground the input. You must maintain a ground state on the center lead until the Trigger event occurs.

EXTERNAL TRIGGER INPUT. The BNC input provides a connection for an external trigger source. The input has 50 Ω impedance and is DC coupled only. Trigger signals as large as ±5 V (DC + Peak AC) may be applied.

SERIAL INTERFACE. The subminiature D connector provides a serial interface for controlling the waveform analyzer and reading acquired data. Refer to page 2–29 for the RS-232 pin assignments. You can configure the serial interface with the commands in the SYST:COMM:SER subsystem.

www.valuetronics.com

Functional Overview

This section describes how the main subsystems of the waveform analyzer operate and the commands you use to control them. The initial discussion describes how the subsystems operate together. Following the overview each of the following topics is described in detail:

- *Acquisition*, page 2–9
- *Vertical Range and Offset*, page 2–19
- *Triggering*, page 2–21
- *Calculations and Measurements,* page 2–31
- *Fast Data Channel,* page 2–50
- *Data Transfers,* page 2–55

System Operation

The waveform analyzer is a multi-channel, high-speed digitizing VXIbus module. In addition to acquiring high-resolution waveforms, the waveform analyzer performs waveform transformations and automated measurements. The transfer of acquired data to the VXIbus controller can use the traditional word serial protocol or the Fast Data Channel, which provides a very high transfer rate.

Control of all these instrument functions is accomplished with the software command set. The waveform analyzer commands follow the SCPI 1995.0 standard for a sensing instrument. Figure 2–2 shows the organization of the waveform analyzer following the SCPI model.

In Figure 2–2, signals enter at the left through the INPut channels and measurement data exits at the right with the TRACe commands. The identifying number for each channel is used to identify corresponding operations in other blocks. For example, to set the input impedance for CH 1 you use the INPut1 command and to set the vertical range of CH1 you use the VOLTage1 command. Likewise, you configure the CH 2 input with the INPut2 and VOLTage2 commands. The four CALCulate function blocks, which perform calculations on any selected input, are an exception to this model.

The TRIGger block in Figure 2–2 controls when to acquire data by setting the trigger event. You can control when trigger events are allowed with the ARM function and its independent front-panel connection. Once the TRIGger block is armed, you can trigger on a TRIGgerA event followed by any defined TRIGgerB

www.valuetronics.com

event. Alternately, you can trigger on different pulse parameters based on settings in the TRIGger:PULSe subsystem.

The discussions following Figure 2–2 describe how to control the major instrument functions and which commands to use.

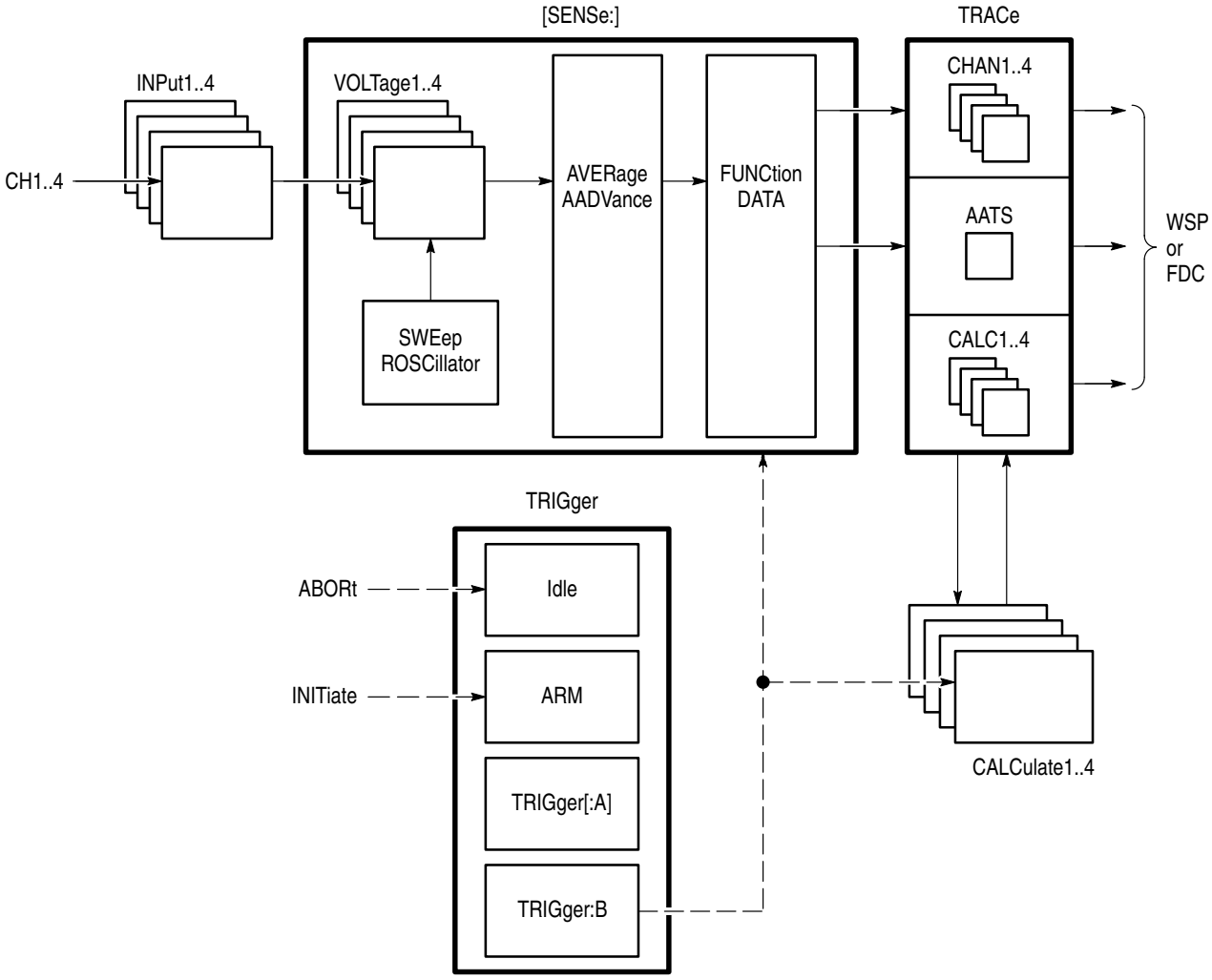


Figure 2–2: Instrument model showing root level nodes

www.valuetronics.com

Acquisition

Acquisition is the process of sampling an analog input signal, converting it into digital data, and assembling it into a waveform record. The waveform analyzer samples the voltage level of the input signal at regular time intervals. The signal parts within the vertical range of the amplifier are digitized. See Figure 2–3.

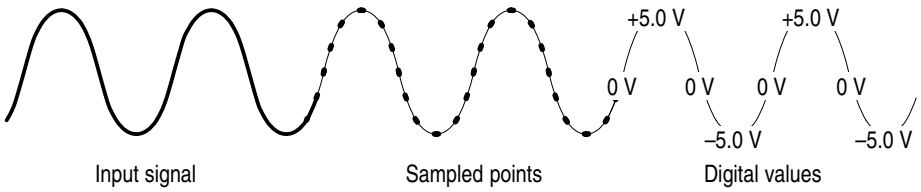


Figure 2–3: Digital acquisition showing sampling and digitizing steps

The sampled points are stored in memory as a waveform record. The waveform record includes the signal source, trigger point location, and horizontal and vertical scaling. Once the signal is acquired, you can use the CALCulate subsystem to perform measurements or to transform the digital waveform record. The waveform analyzer has other types of acquisition, such as Averaging and Enveloping, that produce enhanced waveform records. Refer to *Modes of Acquisition* on page 2–12 and *Averaging and Enveloping* on page 2–18.

Acquisition Hardware

Each channel has a dedicated input amplifier and digitizer as shown in Figure 2–4.

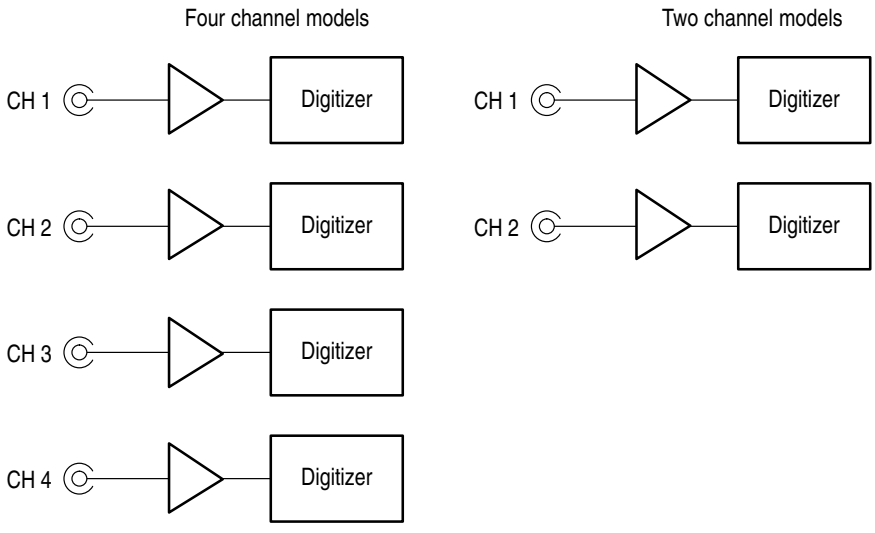


Figure 2–4: Digitizer configuration

www.valuetronics.com

Acquisition Parameters

The waveform analyzer uses a set of common parameters to acquire waveforms from all active channels. These common parameters include the sample interval, record length, trigger point, and the number of samples acquired before (pretrigger) and after (posttrigger) the trigger point. Figure 2–5 shows the trigger point and the pretrigger samples taken before the trigger point.

Sample Interval. The sample interval is the precise time between sample points taken during acquisition. Figure 2–5 shows the sample interval between two waveform samples. The shorter the sample interval, the more detail you acquire for a particular waveform feature. However, short sample intervals reduce the duration of the total waveform record. If you wish to capture many cycles of a waveform, you will likely need to use a longer sample interval. Use the command SWEep:TINTerval to set the sample interval. The sample interval is the reciprocal (1/x) of the sample rate.

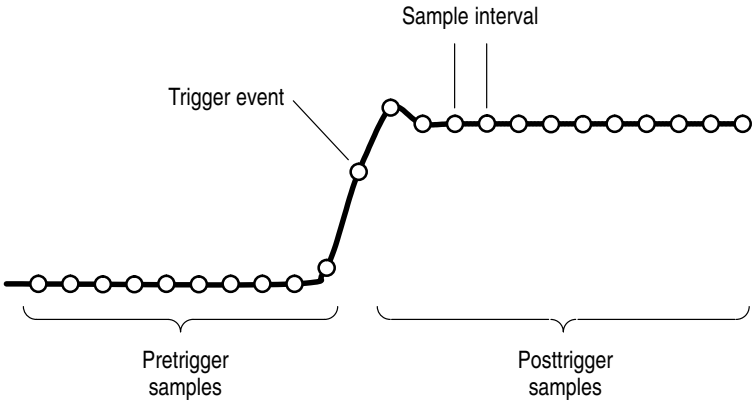


Figure 2–5: Digital sampling showing the sample interval, trigger event, and pretrigger samples

Record Length. The number of samples required to fill a waveform record is the record length. Use the command SWeep:POINts to set or query the number of samples in the waveform record. All channels share the record length setting. You can set the record length from 256 to 15,000 for real-time acquisition. A record length of 30,000 is available in the extended realtime mode. The time window or duration of the waveform record is equal to the Record Length * Sample Interval.

You can also set the record length as a time period with SWEep:TIME. The relationship between SWEep:POINts and SWEep:TIME is as follows:

```
SWEep:TIME = SWEep:TINTerval*SWEep:POINts
```


www.valuetronics.com

SWEep:TINTerval sets the sample interval. For more information, refer to *Sample Interval* on page 2–10 and *Acquisition Modes* on page 2–12.

Trigger Point. The trigger point marks the time zero in a waveform record. All waveform samples are located in time with respect to the trigger point. You can define the trigger point as a simple level and slope crossing, as a number of such trigger events, or upon detection of a defined type of pulse. Figure 2–5 shows a simple level and slope based trigger point. For information on the trigger system, refer to the discussion *Triggering* on page 2–21.

Waveform Record Position. You can position the waveform record relative to the trigger point to acquire pretrigger and posttrigger samples. The trigger point can occur anywhere within the waveform record. Note that all concurrent acquisitions share the record position settings.

The commands that position the waveform record are SWEep:OFFSet and SWEep:OREFerence. Figure 2–6 shows how these commands position the waveform record relative to the trigger point. You can set SWEep:OFFSet as a number of record points (:POINts) or as a period of time (:TIME).

For example, to acquire 50% pretrigger sample points in a 1024 point waveform record, set :OREFence:LOCation to zero and set :OFFSet:POINts to –512. Now the trigger is horizontally centered in the record; half the samples are in the pretrigger region and half in the posttrigger region.

The trigger circuit, with its event count, time delay, and holdoff capabilities is controlled independently from the record positioning functions of the SWEEp commands.

For more information, refer to the SWEep commands starting on page 3–169. For trigger information, refer to *Triggering* on page 2–21.

www.valuetronics.com

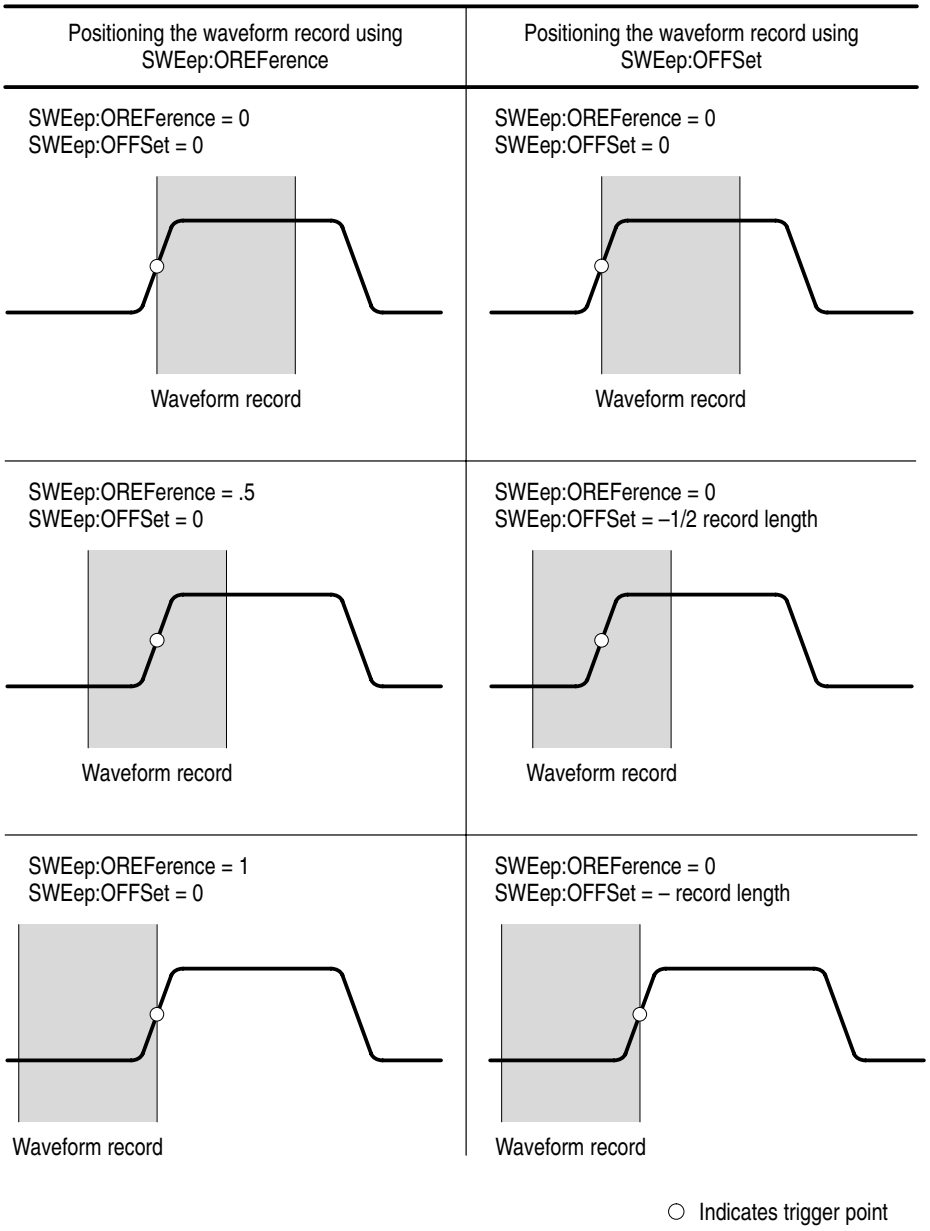


Figure 2–6: Positioning the waveform record relative to the trigger point

Modes of Acquisition

The waveform analyzer provides two modes of acquisition: real time and extended real time. Real-time acquisition is the normal mode after the reset command, *RST. The waveform analyzer switches from real-time to extended real-time acquisition when the sample interval setting is 100 ns or longer. Extended real time acquisition provides up to 30,000 point record lengths. The real-time and extended real-time acquisition modes may be used with the

www.valuetronics.com

auto-advance acquisition cycle, which provides very fast rearm time. Figure 2–8 on page 2–14 shows the auto-advance acquisition cycle.

Real Time Acquisition. With real-time acquisition, the waveform record is filled serially by the digitizers based on a single trigger event. Figure 2–7 shows how real-time acquisition acquires samples in a linear fashion. Because real-time acquisition requires only one trigger event, it is effective for capturing rare or non-repeating events.

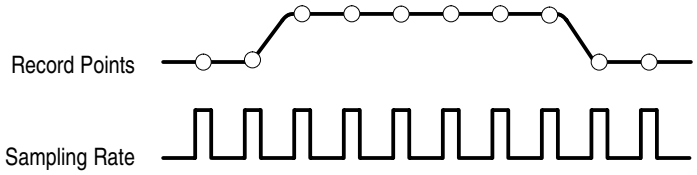


Figure 2–7: Real-time acquisition

Extended Real-Time Acquisition. With extended real-time acquisition, you can acquire a waveform record up to 30,000 points. The waveform record fills serially with only one trigger event. Extended real-time acquisition is available whenever the sample interval is 100 ns or longer, no explicit command is necessary. However, you do need to explicitly set the record length to 30000 after switching to extended real time acquisition if you want the longest record.

Auto-Advance Acquisition. Auto-advance acquisition provides the fastest acquisition rearm time by skipping certain system processing until all waveforms are acquired. Figure 2–8 shows the Auto-advance acquisition loop. For more information see *Auto-Advance Acquisition* on page 2–15.

www.valuetronics.com

Acquisition Cycle

The acquisition cycle follows common steps, but you can select from two types of acquisition looping. The acquisition cycle proceeds through the steps shown in Figure 2–8.

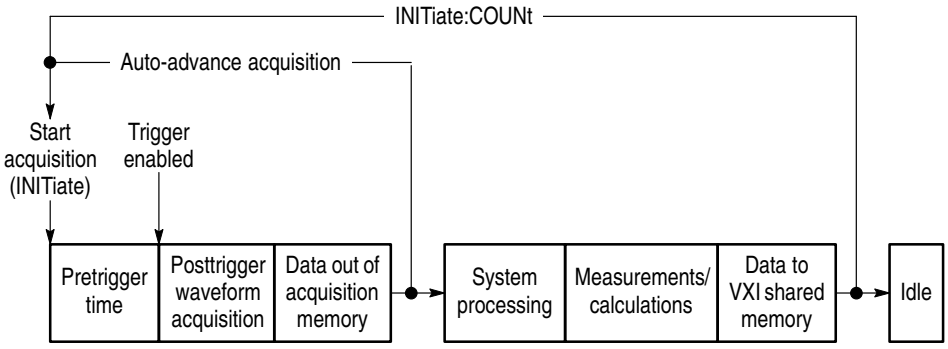


Figure 2–8: The acquisition cycle

A brief description of each acquisition block follows:

- The INIT command is always required to *start acquisition*. INIT puts the acquisition system in the active state awaiting an arm or trigger signal.
- During the *pretrigger time*, the acquisition system initializes itself and the digitizer acquires any pretrigger record points specified with the SWEep commands.
- Before the *trigger enabled* event occurs, any specified ARM event must occur and all required pretrigger samples must be acquired. Once enabled, the next valid TRIGger:A event is accepted.
- During the *posttrigger waveform acquisition* period, the active channel digitizers acquire all posttrigger waveform samples needed to fill the defined waveform records.
- In the *data out of acquisition memory* block, the digitizers transfer acquired data out of acquisition memory and into DSP memory.

As shown in Figure 2–8, acquisition normally proceeds at this point with system processing and any specified measurements or calculations. After processing completes, the resultant waveform records are moved to VXIbus shared memory for transfer to your system controller.

Acquisition Looping. The waveform analyzer provides two different ways to loop or cycle through a specified number of acquisitions. You might choose acquisition looping in order to capture a series of waveform records or measurements over a period of time. The two methods are auto-advance acquisition mode (AADV commands) and INIT looping.

www.valuetronics.com

Auto-advance provides the fastest cycle time but does not perform and return measurement results until all acquisitions complete. Figure 2–9 compares the acquisition cycle time between auto-advance acquisition and INIT looping.

When auto-advance acquisition is active, system processing starts on the auto-advance waveforms when the last acquisition loop completes. With the INIT:COUNt looping mode shown in Figure 2–8, all processing completes and measurement results return before another acquisition cycle begins. In single-shot acquisition (no looping enabled), the acquisition system returns to the Idle state after acquisition completes.

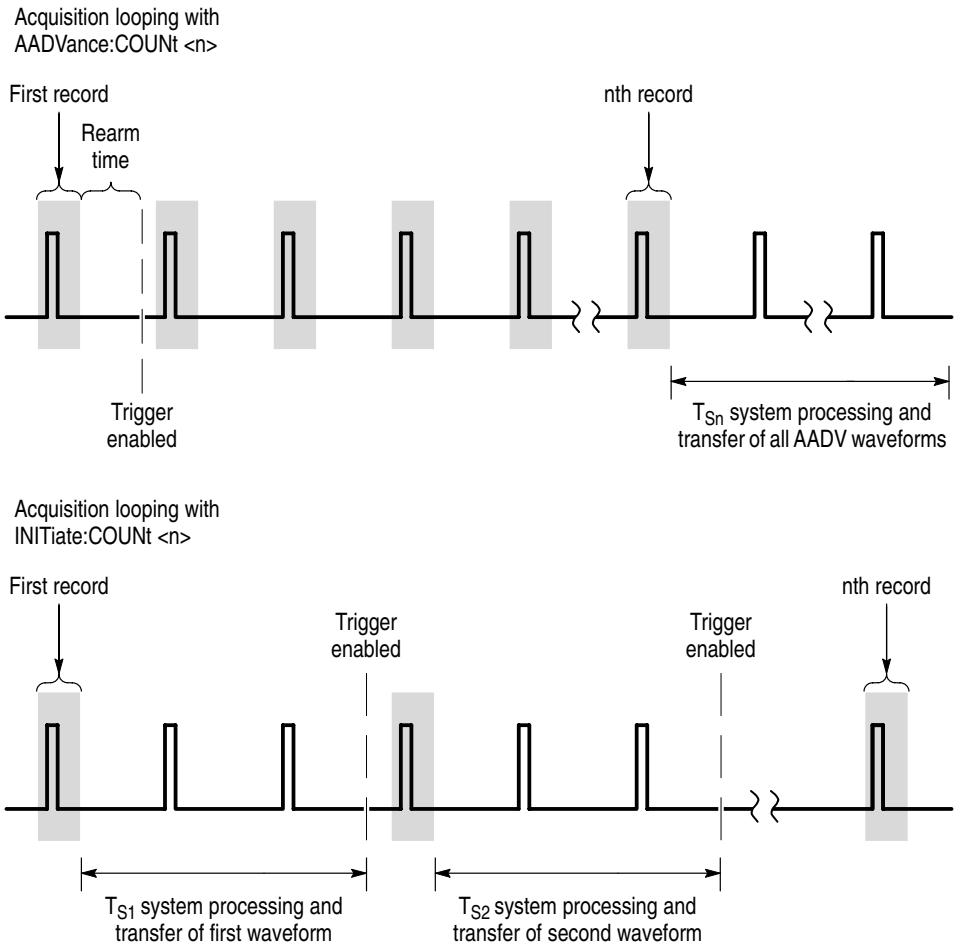


Figure 2–9: Comparison between auto-advance and INIT acquisition looping

Auto-Advance Acquisition

Auto-advance acquisition acquires a sequence of waveform records with minimal delay between acquisitions. As shown in Figure 2–8, auto-advance acquisition provides the fastest acquisition cycle time because the system processing, measurement, and data transfer tasks are delayed until acquisition completes.

www.valuetronics.com

Another feature of auto-advance acquisition is that each waveform record includes a timestamp which records the time between consecutive waveform records. Timestamp resolution is 125 ns.

Many of the basic settings for auto-advance acquisition are the same as for normal acquisition. You enable channels, define the record length and sample interval.

Several additional steps are necessary to configure auto-advance acquisition. First, enable the auto-advance mode with the command AADVance:STATe ON. Next, specify the number of waveform records to acquire with the command AADVance:COUNt <n>. You can set :COUNt to fill acquisition memory with waveform records by giving :COUNt the value MAX or 0 (zero).

The Timestamp. During auto-advance acquisition, the time between consecutive waveform records is recorded in a timestamp record. The timestamp record has the trace name AATS and is available with the transfer commands TRACe?, TRACe:COPY, etc. The record contains a sequence of times in seconds, from t0 to tn, separated by commas. Set the format of the AATS record to ASCII or 32-bit REAL numbers with the command FORM:TRAC:AATS.

Fast Data Channel and Auto-Advance Acquisition

The Fast Data Channel (FDC) provides a fast transfer protocol for moving waveform records and other data between the waveform analyzer and a VXIbus controller. The waveform analyzer includes an FDC driver that provides the instrument side of the protocol. To use FDC, you must do the following tasks:

- Ensure that controller used provides support for the FDC protocol. (The VXIplug&play VISA I/O interface provides FDC support.)
- Open two sessions with the waveform analyzer, one for sending ASCII commands and another for FDC data transfer.
- Configure the FDC session to use the FDC protocol (by setting VISA attributes — see step 10 in the example).
- Use the ASCII session to set up the waveform analyzer, acquire the data, and initiate its transfer.

When configuring the ASCII and FDC sessions, you must perform setup steps in proper order:

- The number of active channels and the record length must be set before you enter the number of waveform records to acquire with AADVance:COUNt. The waveform analyzer automatically computes the maximum (MAX)

www.valuetronics.com

AADV:COUNt value based on the number of active channels and the record length.

- You must make all acquisition settings and configure the FDC configuration before issuing the INITiate command to begin acquisition.

The following example shows the steps necessary to set up an auto-advance acquisition with FDC data transfer, observing the principles listed above. The example assumes you want to acquire CH1 and CH2.

1. Enable the channels to acquire with the command FUNC:ON “XTIM:VOLT 1”,“XTIM:VOLT 2”.
2. Set the individual channel parameters, such as voltage range.
3. Set the trigger parameters.
4. Set the waveform record length with SWE:POIN and the sample interval with SWE:TINT.
5. Enable auto-advance acquisition with the command AADV:STAT ON.
6. Specify the number of waveform records to acquire with the command AADV:COUN command. The value MAX acquires enough waveform records to fill DSP memory.
7. Specify the number of waveform records to transfer over the FDC with the command AADV:REC:COUN. The value MAX selects all acquisitions, from :STARt on, for transfer.
8. Specify the list of signal sources to transfer with the command TRAC:LIST. You may have an active channel but not select it for transfer. In this example, use TRAC:LIST CHAN1,CHAN2. The value AATS selects the timestamp record for the auto-advance waveform records that you will acquire.
9. You can choose to skip the transfer of the first acquisition records for both channels if, for example, you think the device under test might initially be unstable. To skip just the first two waveform records, specify the number of the waveform record to begin FDC transfers using the command AADV:REC:STAR 3 which skips over records 1 and 2.

www.valuetronics.com

10. Configure the FDC session to use the FDC protocol by setting the following VISA attributes using the function:

    viSetAttribute (vi_session, attr_name, attr_value)

    Attribute Name
    Attribute Value

    VI_ATTR_FDC_CHNL
    1

    VI_ATTR_FDC_MODE
    VI_FDC_STREAM

    VI_ATTR_IO_PROT
    VI_FDC

11. Start acquisition on the waveform analyzer with the INITiate command. Acquisition begins.

12. Use the VISA function viRead (fdc_session, buf, buf_size, return_count) to transfer the acquisition from the waveform analyzer to the character array buffer of the controller.

For more information, refer to *Auto-Advance Acquisition* on page 2–15, *Acquisition Cycle* on page 2–14, and *Fast Data Channel* on page 2–50. Additionally, refer to the command descriptions in Chapter 3, *Syntax and Commands*.

Average and Envelope Waveforms

Averaging and enveloping acquisition modes combine a number of acquisitions in one waveform record. Averaging accumulates an average value for each record point over many acquisitions to provide higher vertical resolution. Enveloping finds the highest and lowest sample values over many acquisitions to produce a waveform record of alternating maximum and minimum values. The commands in the AVERage subsystem control averaging. All active channels are affected by the AVERage setting. Averaging and enveloping occur in the acquisition system before waveform records are passed to the CALC blocks. An averaged waveform can often produce a more accurate measurement.

To enable averaging, set AVERage ON and specify the number of waveform records to average with AVERage:COUNt. Use the command AVERage:TYPE to select averaging (SCALar) or enveloping (ENVelope).

For more information, refer to the command descriptions in Chapter 3, *Syntax and Commands*.

www.valuetronics.com

Vertical Range and Offset

The vertical range and offset for the waveform analyzer input channels are individually controlled. The controlling commands are in the VOLTage subsystem, described on page 3–247. To set the vertical scaling for an input channel you actually set the vertical range and the offset of the channel digitizer. The offset value sets the voltage level that is at the middle of the digitizer range.

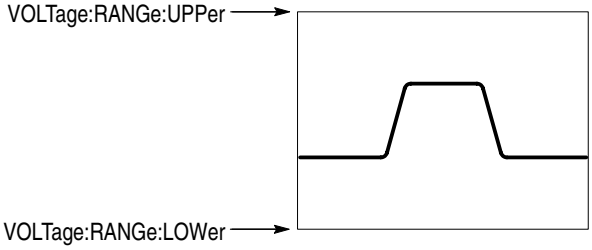
The waveform analyzer provides two methods of setting the vertical range and offset: RANGe and OFFSet, and UPPer and LOWer. Figure 2–10 shows how these two methods are used and presents several offset examples. Note in the example that it is the input vertical range or "window" of the digitizer that is moved by vertical OFFset and not the DC level of the input signal. Applying a negative OFFset moves the vertical range down. Likewise, applying a positive OFFset moves the vertical range up. You can use the UPPer and LOWer parameters to set the range to absolute vertical values if you prefer.

Valid waveform data are in the range +32256, for waveform values that equal VERTical:RANGe UPPer, to –32256, for waveform values that equal VERTical:RANGe LOWer.

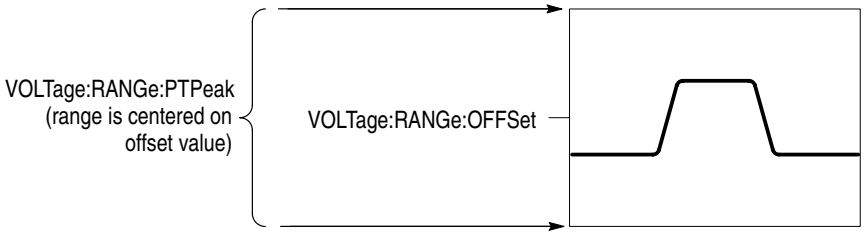
Parts of a waveform that exceed the range are clipped. Parts that exceed the most positive range limit (UPPer) are acquired as Overrange points and are assigned the value +32767. Parts that exceed the most negative range limit (LOWer) are acquired as Underrange points and are assigned the value –32767.

www.valuetronics.com

Setting vertical range with upper and lower limits



Setting vertical range with offset and peak-to-peak range



Examples:

VOLTage:RANGe:PTPeak 5 V

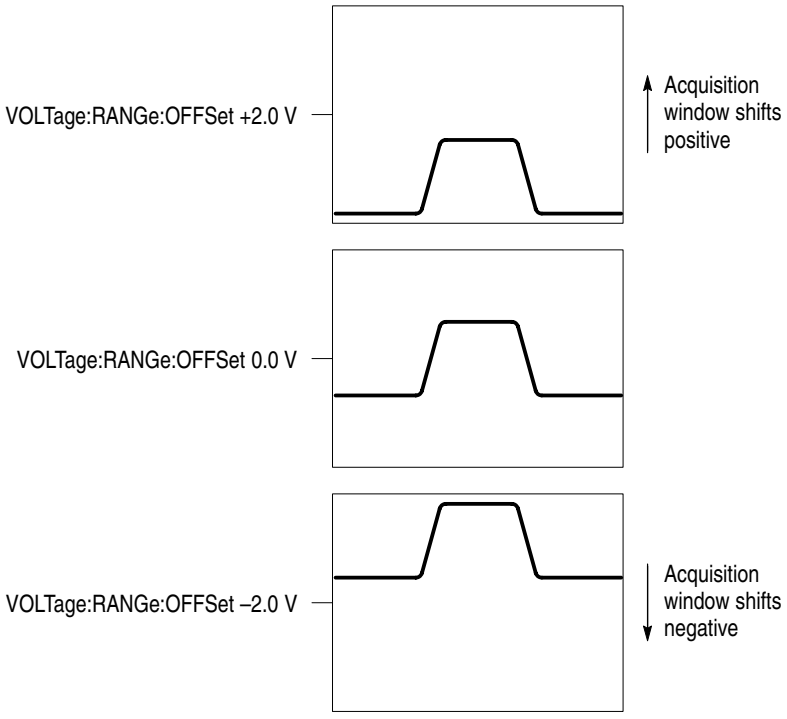


Figure 2–10: Setting vertical range and offset of input channels

www.valuetronics.com

Triggering

Triggering determines when the waveform analyzer acquires a waveform record. The trigger system parameters, such as trigger level and slope, determine at what point on a waveform the trigger event occurs. Figure 2–11 shows the waveform analyzer Arm/Trigger cycle that all acquisitions follow. The illustrated cycle follows the SCPI standard for triggering.

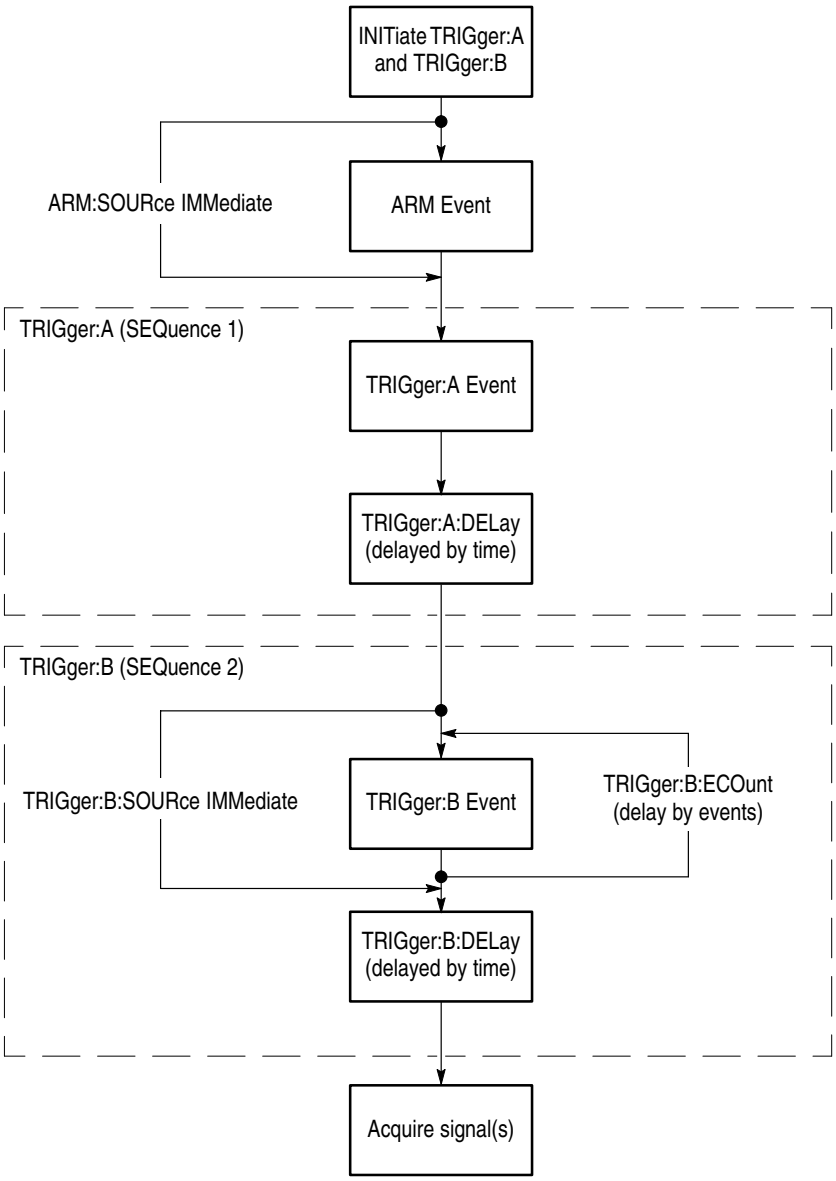


Figure 2–11: The Arm/Trigger cycle

www.valuetronics.com

The TRIGger:A and TRIGger:B delay capabilities shown in Figure 2–11 provide many delayed trigger modes.

The Trigger Event

The trigger event establishes the time-zero point (t_0) in the waveform record. The time locations of all points in the waveform record are relative to the trigger point. Once the waveform analyzer detects a trigger event, it ignores triggers until the acquisition completes and trigger holdoff expires. You can define a trigger event as a simple edge trigger, a delayed trigger, or a pulse trigger. Regardless of the type of triggering, you need to choose a trigger source.

Trigger Sources

The trigger source is the signal monitored for a trigger event. Table 2–1 lists the possible trigger sources and their compatibility with each trigger type.

Table 2–1: Trigger Sources and Compatibility

Trigger Source	Description	Trigger Type		
		EDGE TRIG:A	EDGE TRIG:B	TRIG :PULSe
Input channels CH1, CH2, CH3 CH4 (depending on model)	Provide triggering from signals connected to the input channels. Parameters include coupling (AC/DC), level, slope, and trigger filters. Trigger filters (:LPASs, HPASs, and NREJ) are available to eliminate low or high frequency elements. The channel you select as a trigger source need not be acquired.	✓	✓	✓
External Trigger	Selects the input connector EXTERNAL TRIGGER INPUT on the waveform analyzer front panel. The input is DC coupled and has a maximum range of ±1 V. The available trigger settings are slope and level.	✓	✓	
VXIbus Triggers	These digital trigger signals from the VXI bus are available to all VXI modules. The waveform analyzer can trigger off and drive eight TTL logic trigger lines (TTLTrg0 – TTLTrg7) and two ECL logic trigger lines (ECLTrg0 and ECLTrg1). The TTLTrg lines have a 12.5 Mhz clock speed and the ECLTrg lines 62.5 Mhz. The waveform analyzer can source the TTLTrg lines with any valid trigger source. Refer to the OUTPut:TTLTrg commands starting on page 3–144.	✓		
IMMediate	Effectively bypasses the TRIGger:B event detector.		✓	

www.valuetronics.com

Trigger Coupling

Trigger coupling determines which frequency components of an analog trigger signal are passed to the trigger system. The VXIbus trigger sources are digital signals and do not provide coupling selections. The External trigger provides DC coupling only. TRIGger:A and TRIGger:B provide a variety of trigger coupling selections with the TRIGger:A and TRIGger:B COUPling commands, when using the input signals.

Aside from the basic AC and DC coupling, the waveform analyzer offers several coupling preset commands that also control trigger filtering and noise rejection. The coupling selections are as follows:

- DC passes all of the input signal. In other words, it passes both AC and DC components to the trigger circuit.
- AC passes only the alternating components of an input signal. It removes the DC component from the trigger signal.
- HFReject removes the high frequency portion of the triggering signal. Only the low frequency components pass on to the triggering system to start an acquisition. High frequency rejection attenuates signals above 50 kHz.
- LFReject removes the low frequency portion of the triggering signal. Only the high frequency components pass on to the triggering system to start an acquisition. Low frequency rejection attenuates signals below 50 kHz.
- ACNReject provides noise rejection while rejecting the DC portion of the signal.
- DCNReject provides noise rejection but passes all other portions of the signal.

Slope and Level

The slope and level settings determine the analog parameters for the trigger point for edge triggering. Figure 2–12 shows the commands that control each parameter. The trigger level must be within the signal range to cause a trigger event. The slope setting allows you to capture the rising or falling edge of a signal.

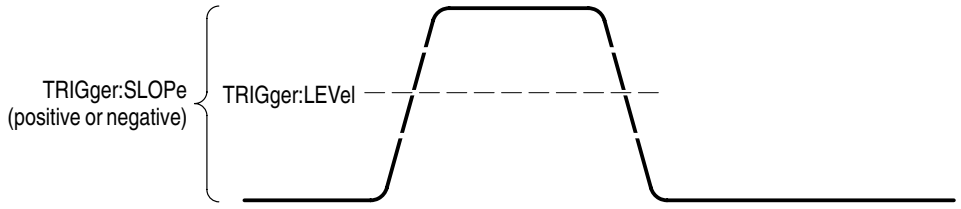


Figure 2–12: Slope and level define the trigger event

www.valuetronics.com

Trigger Position

The trigger position defines where on the waveform record the trigger occurs. It lets you properly align and measure data within records. The part of the record that occurs before the trigger is the pretrigger portion. The part that occurs after the trigger is the posttrigger portion.

For information on setting the trigger position refer to *Waveform Record Position* on page 2–11.

Trigger Holdoff

When the waveform analyzer recognizes a trigger event, it disables the trigger system until acquisition and trigger holdoff complete. Trigger holdoff starts when the trigger event occurs as shown in Figure 2–13. As shown in the illustration, you can set the holdoff time to skip signal pulses that would cause false triggering. Without trigger holdoff, pulses two, three, and four would be valid trigger events assuming sufficient time for acquisition and rearming. The desired trigger event is the first rising edge of each group of pulses. Note that the waveform record has 50% of its samples before the trigger point.

The holdoff range is from 250 ns (minimum) to 12 seconds (maximum).

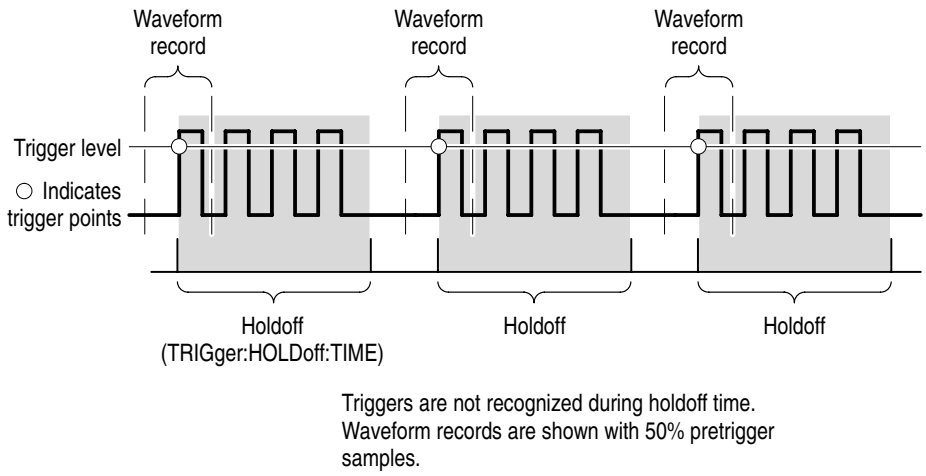


Figure 2–13: Trigger holdoff time ensures valid triggering

www.valuetronics.com

Trigger Types

The waveform analyzer provides two standard trigger types: edge and pulse. Use the command TRIGger:TYPE to select one. A brief definition of each type follows:

- **EDGE** is the default trigger type. An edge trigger event occurs when the trigger source passes through a specified voltage level in the specified direction (the trigger slope).
- **PULSe** is a special-purpose trigger based on the shape and duration of pulses on the trigger signal. The two classes of pulse triggers are glitch and width. Pulse triggering uses parameters separate from those used with Edge triggering.

Automatic Trigger Mode

The trigger mode determines how the waveform analyzer behaves in the absence of a trigger event. The waveform analyzer provides two trigger modes, normal and automatic. Use the command TRIGger:ATRigger to enable automatic triggering.

- **Normal** trigger mode enables the waveform analyzer to acquire a waveform only when it is triggered. If no trigger occurs, the waveform analyzer does not acquire a waveform.
- **Automatic** trigger mode enables the waveform analyzer to acquire a waveform even if a trigger event does not occur. Auto mode uses a 500 ms timer that starts after INIT starts acquisition. If the trigger circuit does not detect a trigger event before the time expires, the waveform analyzer forces a trigger.

Edge Triggering

The waveform analyzer can trigger on the rising or falling edge of a waveform. An edge trigger event occurs when the trigger source passes through a specified voltage level in a specified direction (the trigger slope). Edge triggering parameters control the TRIGger:A and TRIGger:B triggers. Use the command TRIGger:TYPE to select EDGE or PULSe triggering.

For information on edge trigger parameters, refer to *Slope and Level* page 2–23. For information on delayed triggering, refer to *Delayed Triggering* on page 2–26.

www.valuetronics.com

Delayed Triggering

The waveform analyzer provides delayed trigger functionality with the delay capabilities of TRIGger:A and TRIGger:B. TRIGger:A provides a time delay and TRIGger:B provides a delay by number of edge events and by a time delay. The time delay specifies a period to wait after the trigger event to start acquisition.

There are two ways to delay the acquisition of waveforms: delayed runs after main and delayed triggerable. Delayed triggerable uses the TRIGger:B system. Delayed runs after main looks for a TRIGger:A event, then waits a user-defined time, and then starts acquiring. See Figure 2–14.

Figure 2–14: Delayed runs after main

Delayed triggerable mode looks for a TRIGger:A event and then, makes one of the three types of delayed triggerable acquisitions. Figure 2–15 shows the delayed trigger modes and the sequence for each delayed mode.

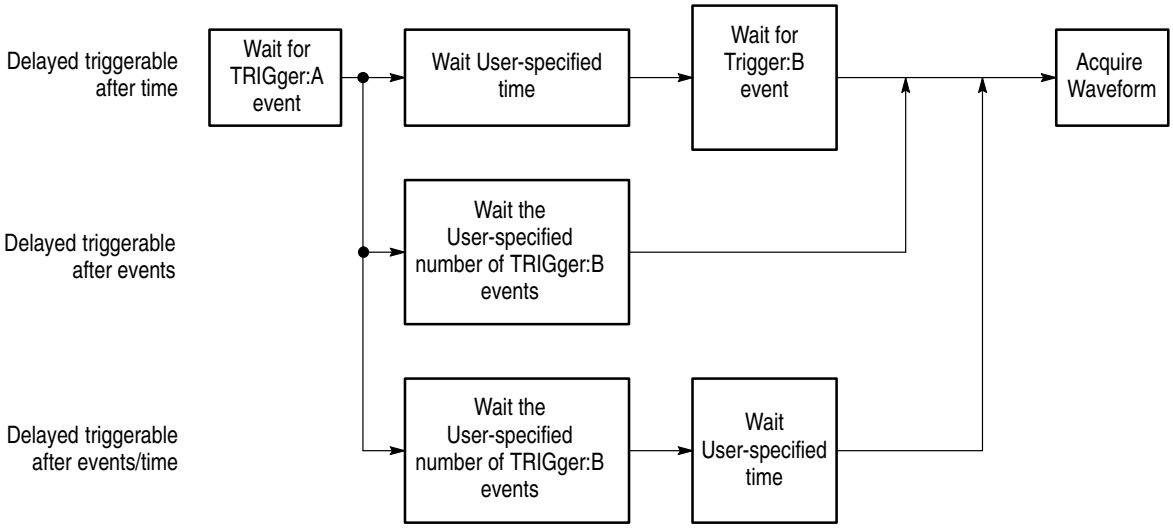


Figure 2–15: Delayed triggerable

Figure 2–16 shows the traditional delayed trigger modes and the commands used to set these modes.

www.valuetronics.com

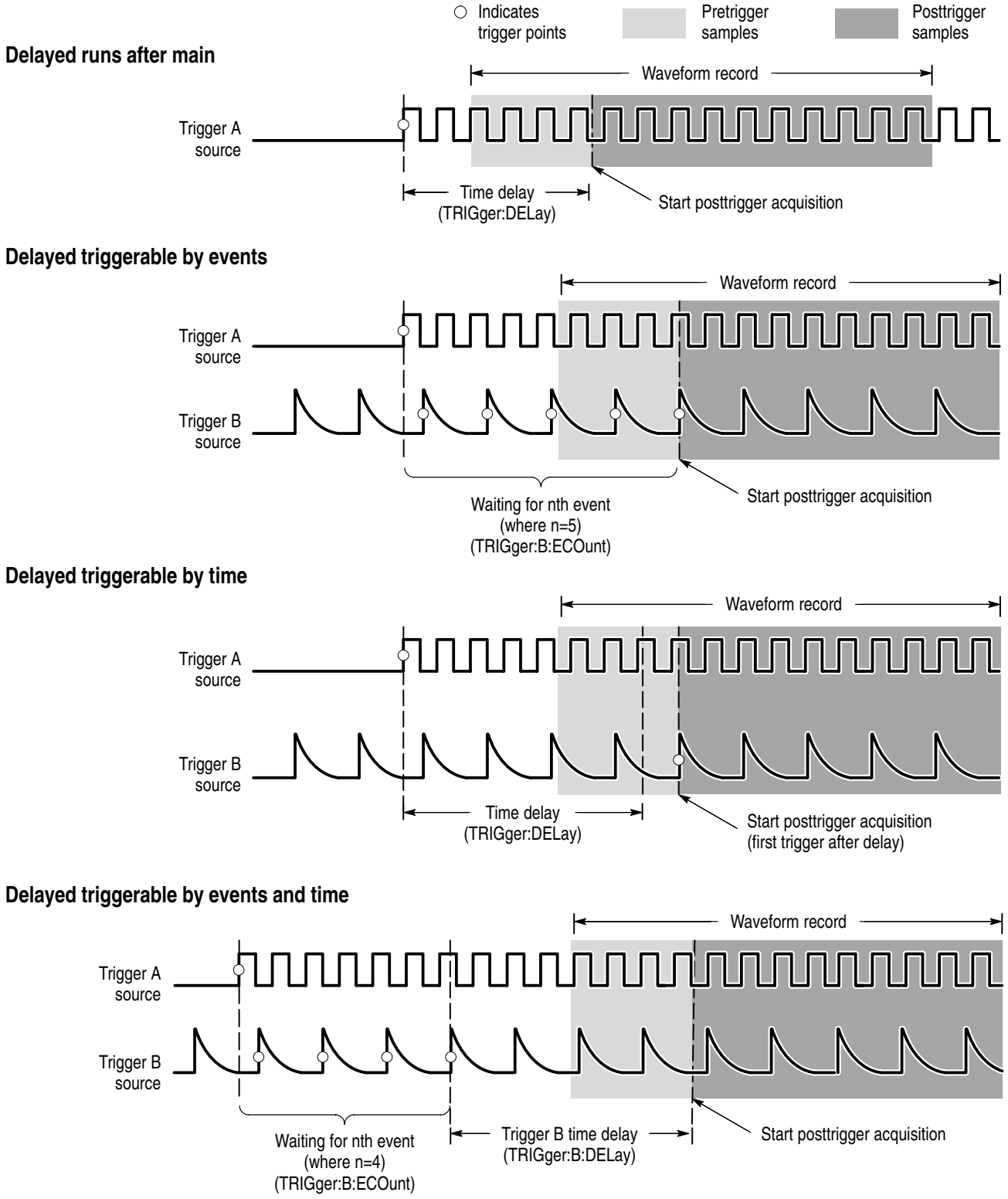


Figure 2–16: How the delayed triggers work

www.valuetronics.com

Interaction Between Delay and Holdoff. Trigger delay determines how long after a trigger event to start acquisition. Trigger holdoff sets how long after one trigger another event another can occur. Figure 2–17 shows the relationship between trigger delay and trigger holdoff. Note that trigger holdoff must include the trigger delay time to be effective.

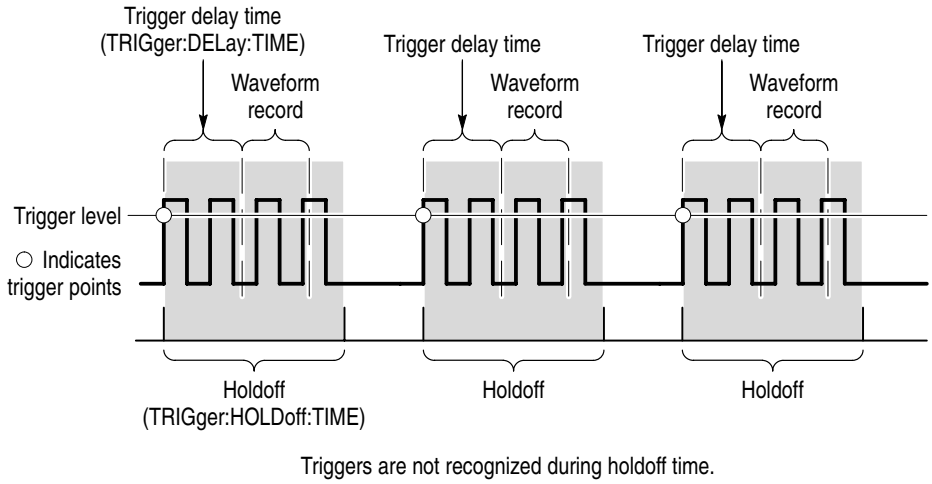


Figure 2–17: Trigger holdoff time with trigger delay time

Pulse Triggering

The waveform analyzer can trigger on various pulse types using a special set of TRIGger:PULSe parameters. When a qualified pulse event occurs, acquisition is started just as it would from an edge trigger. Use the TRIGger[:A]:TYPE command to select pulse triggering.

There are two classes of pulse triggering available, which you select with the command TRIGger:PULSe:CLASs. The classes of PULSe triggering are as follows:

- **GLITch** triggering occurs when the trigger source detects a pulse narrower (or wider) in width than some specified time. It can trigger on glitches of either polarity.
- **WIDTh** triggering occurs when the trigger source detects a pulse that is inside or, optionally, outside some specified time range (defined by the upper limit and lower limit). The waveform analyzer can trigger on positive or negative width pulses.

Figure 2–18 shows the PULSe trigger parameters and the commands used to set GLITch and WIDTh parameters. Both classes of PULSe triggering have unique parameters except for the vertical level parameter, TRIGger:PULSe:THReshold, which they share. THReshold determines the width of a pulse which is then compared against qualifying parameters.

www.valuetronics.com

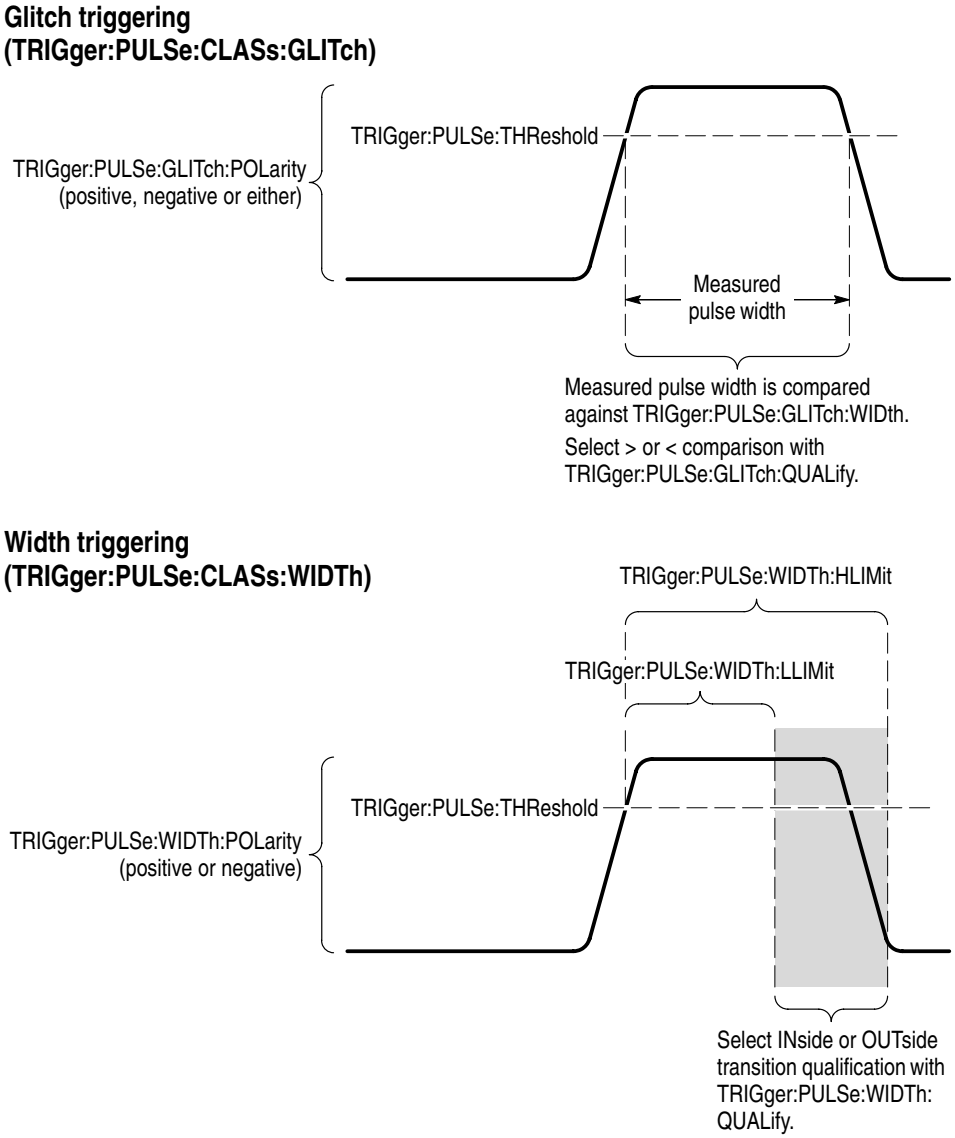


Figure 2–18: Parameters for pulse triggering

The qualifying parameters for GLITch triggering are as follows:

- **POLarity** sets what pulse polarity can qualify the trigger event. The choices are POSitive, NEGative, or BOTH. The selection BOTH accepts either positive or negative pulses.
- **WIDth** sets the comparison pulse width used to qualify the measured pulse width. The THReshold level helps determine the measured pulse.
- **QUALify** sets whether the measured width must be greater than (GT) or less than (LT) the WIDth parameter to generate a trigger event.

www.valuetronics.com

The qualifying parameters for WIDTh triggering are as follows:

- **POLarity** sets what pulse polarity can qualify the trigger event. The choices are POSitive or NEGative.
- **WIDth:HLIMit** sets the maximum extent of the range used to qualify the measured pulse width. The THReshold level helps determine the measured pulse.
- **WIDth:LLIMit** sets the minimum extent of the range used to qualify the measured pulse width. The THReshold level helps determine the measured pulse.
- **QUALify** sets whether the last edge of the pulse must be INside or OUTside the region defined by the WIDth parameters to generate a trigger event.

VXIbus Triggering

The waveform analyzer can trigger on the VXI bus TTL and ECL trigger lines. These are digital signals carried by the P2 backplane. They may be generated by any VXIbus module, including the waveform analyzer. These lines provide an easy way to trigger several modules from one VXI source. The TTLTrg lines have a 12.5 Mhz clock speed and the ECLTrg lines 62.5 Mhz. No coupling or filtering is possible due to the digital nature of these signals.

Use the command TRIGger:[A]:SOURce to select from the signals TTLTrg0 to TTLTrg7 and ECLTrg0 to ECLTrg1. The TTL signals conform to the TTL logic standard. Likewise, the ECL signals conform to the ECL logic standards.

The waveform analyzer can source the TTLTrg lines with any valid trigger source. Use the commands OUTPut:TTLTrg and OUTPut:ECLTrg to configure and drive the VXI trigger lines. For more information, refer to the OUTPut:TTLTrg commands starting on page 3–144.

www.valuetronics.com

Calculations

This section describes the waveform analyzer CALCulation subsystem. The CALCulate subsystem provides an extensive set of measurement, mathematical, transformation and filtering functions. This section begins with a discussion of the two models or methods of programming the CALCulate subsystem. Then, each of the following types of calculation is discussed:

- Measurements of signal parameters, such as rise time
- Waveform mathematics for inverting, adding, subtracting, and multiplying waveform records
- Waveform Integration, which creates the integral of a waveform record (see page 2–39)
- Waveform Differentiation, which creates the derivative of a waveform record (see page 2–39)
- Fast Fourier Transforms (FFT) which produce an amplitude versus frequency waveform record (see page 2–40)
- Digital filtering used to remove certain frequency components from a waveform record

Overview of CALC Models

The calculation system provides two methods for defining calculations:

- With the standard SCPI model, you define a signal source and a series of functions to execute on the source.
- With the expression model you describe a calculation with an algebraic expression using signal sources as variables.

Only one of the two CALC models may be active at a time. Both models allow you to use all CALC functions, such as FILTer and TRANsform. The expression model also allows calculations that operate on more than one source at a time, such as when adding the waveforms in CH 1 and CH 2. In general, the SCPI model allows only calculations that operate on one source at a time. (Measurement calculations using two waveforms, such as phase or delay, are exceptions to the one source limitation.) The two models are discussed in detail under the topics *SCPI Calculation Model* and *Expression Calculation Model* in this section.

Four CALC Blocks. The waveform analyzer provides four separate CALC blocks, each with separately defined CALC settings. This means that for each CALC block you will use you must set the measurement parameters in each CALC block or accept the default settings. For example, to perform the same measurement on the CH 1 and CH 2 inputs, you first define the measurement in two CALC blocks with a command like the following command:

www.valuetronics.com

```
CALC1:WML FTIM,RTIM;:CALC2:WML FTIM,RTIM
```

Two CALC blocks can perform measurements on the same source. This feature allows you to perform a calculation, such as a measurement, in more than one way for every acquisition. For example, you could perform the same measurement in two CALC blocks, but use the FILTer function to remove high-frequency components from the waveform before the measurement in one of the CALC blocks.

For information on setting parameters for calculation functions, refer to the individual command descriptions in section 3 and to discussions of the CALC functions in this section of the manual.

SCPI Calculation Model

The waveform analyzer supports the standard SCPI calculation model. In the SCPI CALCulate model, you use the CALCulate:PATH command to define a series of functions to perform on a single source. Figure 2–19 shows how the CALC:PATH command sets the data flow through a set of functions.

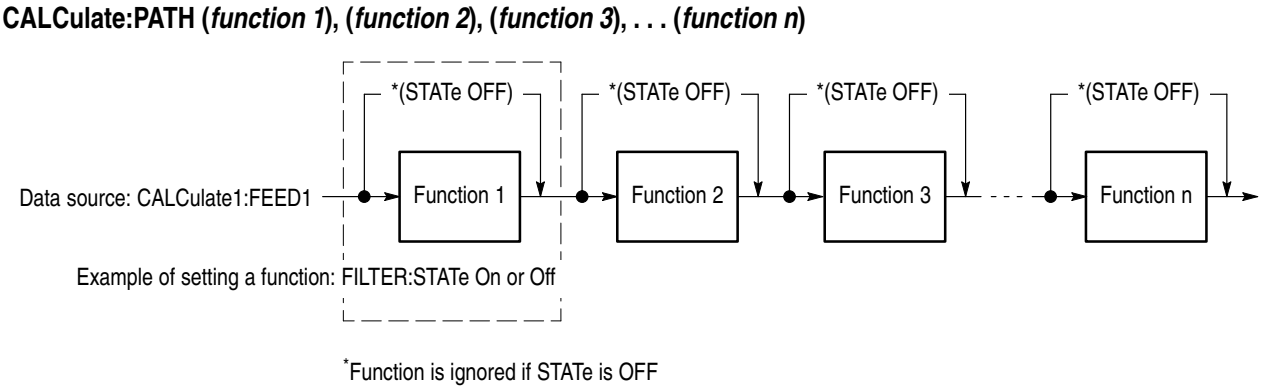


Figure 2–19: PATH definition for SCPI calculation model

Functions included in the CALC:PATH are performed only when their individual STATe is set to ON. Functions whose STATe is set to OFF are not executed and any data passed to them is routed unchanged to the next function in the PATH.

When defining SCPI model calculations that measure parameters based on two waveforms (such as Gain and Delay), the functions you set using CALC:PATH only apply to the data source selected with FEED1. The source selected by FEED2 is unaffected by the functions you select.

NOTE. *SCPI model calculations are disabled when you define an expression model calculation with CALC:PATH:EXPR.*

www.valuetronics.com

CALC Data Source. You must define a single data source for each CALC block using the CALC:FEED1 command. See Figure 2–19. The allowed sources are the function strings “XTIM:VOLT <n>” that represent the input channels. The command CALC1:FEED1 “XTIM:VOLT 1” connects the CH 1 waveform record to the CALC1 block. You may also specify the FEED1 source as CHAN<n>. More than one CALC block can operate on a source. For more information, refer to the CALC:FEED1 command in section 3 of this manual.

You may also define a second data source for each CALC block using the CALC:FEED2 command. The allowed sources are the same as for CALC:FEED1; for example, the command CALC1:FEED2 “XTIM:VOLT 2” connects the CH 2 waveform record to the CALC1 block. SCPI-model calculations that make dual-waveform measurements (such as when measuring Gain or Delay) can make use of the second data source.

Expression Model

The waveform analyzer supports an expression based model for defining calculations. With the expression model, you define a calculation involving one or more functions using an algebraic expression. The expression syntax follows standard C language syntax. For example, the following expression calculates the derivative of the difference between the CH 1 and CH 2 input signals.

```
CALC1:PATH:EXPR (DER(CHAN1-CHAN2))
```

The source to operate on is defined in the expression (CHAN1 and CHAN2 in the example) and not with a FEED command as in the SCPI model. The sources must be in acquisition memory to provide the expression with useful data. If CHAN 1 is not acquired, the above expression would produce an error during acquisition and calculation.

EXPRession Syntax. The syntax for CALCulate expressions is defined in the following BNF description:

```
<expr>       ::=        <factor> {[+|-|*|/] <factor>}
<factor>     ::=        [-<number>|{+}<number>]
                        |<ch_expr>
                        |AATS
                        |<func>"("<expr>[,<expr>]")"
<ch_expr>    ::=        <chan>{"["<nrx>"]"}
<chan>       ::=        [WMP1:|WMP2:|WMP3:|WMP4:]
                        CHAN1|CHAN2|CHAN3|CHAN4
<func>       ::=        AAMLIST|AC|AMPLitude|AREA|CARea|CMEan|
                        CRMS|CROss|DC|DELay|DERivative|FFT|
                        FILTer|FORMat|FREQuency|FTIMe|GAIN|HIGH|
                        INTegral|LOW|MAXimum|MEAN|MID|MINimum|
                        NCROss|NDUTycycle|NWIDth|PCROss|PERiod|
```


www.valuetronics.com

```
PHAse|PWIDth|PTPeak|RMS|RTIMe|
SDEViation|SMOothing|TRANsform|TTRig|
WMList]
```

An example of the expression syntax follows:

```
CALC1:PATH:EXPR (FORM(TRAN(FILT(CHAN1))))
```

This command performs frequency filtering on CHAN1, then performs an FFT transform and finally formats the result, perhaps in a logarithmic magnitude format. The settings for CALC:FILTer, :TRANsform, and :FORMat functions must be completed in separate command statements prior to starting acquisition. The settings might include enabling the low pass filter with CALC:FILT:FREQ:LPAS, setting the TRANsform WINDow type, and setting the result format for the resulting FFT waveform record with CALC:FORM MLOG.

The following example produces a waveform composed of data points that are the ratio of the deviation from the MEAN over the measured AMPLitude.

```
CALC2:PATH:EXPR ((CHAN1-MEAN(CHAN1))/AMPL(CHAN1))
```

Expressions for Auto-Advance Records. Expressions for auto-advance acquisition are similar to those for normal acquisition. You can specify a set of measurements to perform on all auto-advance waveform records with the command CALC1:AAMList. An example follows:

```
CALC1:PATH:EXPR (AAMList(CHAN1))
```

If AAMList is set to measure RTIMe (rise time), the expression returns a vector of RTIMe measurements for all auto-advance acquisitions (as delimited by AADV:REC:START,COUNT).

You can also specify individual measurements or functions to perform on a single record. In the following example, MEAN is measured only on auto-advance record number five.

```
CALC1:PATH:EXPR (MEAN(CHAN1[5]))
```

Measurements

The waveform analyzer provides voltage, time, area, and statistical waveform measurements. This section describes these measurements and how to use them. For algorithm information, see *Appendix B: Algorithms*.

Table 2–2 lists the available measurements in the waveform analyzer.

www.valuetronics.com

Table 2–2: Measurement Definitions

Measurement Command	Definition
AMPLitude	Voltage measurement. The High value less the Low value measured over the entire waveform. This result is not usually equal to the peak-to-peak value because of the way High and Low are determined. *AMPLitude = HIGH – LOW*
AREA	Voltage over time measurement. The area over the entire waveform in volt-seconds. Area measured above ground is positive; area below ground is negative.
CARea (Cycle area)	Voltage over time measurement. The area over the first cycle in the waveform in volt-seconds. Area measured above ground is positive; area below ground is negative.
CMEan (Cycle mean)	Voltage measurement. The arithmetic mean over the first cycle in the waveform.
CRMS (Cycle RMS)	Voltage measurement. The true Root Mean Square voltage over the first cycle in the waveform.
CROSs	Timing measurement. The time of the Nth positive- or negative-going crossing of the MREF (mid-reference) level, where N is set by CALC<n>:WMP:EDGE. Time is relative the trigger point of the acquisition.
DELay	Timing measurement. The time between the MidRef crossings of a reference and a target waveform.
FREQuency	Timing measurement for the first cycle in the waveform. The reciprocal of the PERiod. Measured in Hertz (Hz).
FTIMe (Fall time)	Timing measurement. Time taken for the falling edge of the first pulse in the waveform to fall from a HREF value (default = 90%) to a LREF value (default =10%) of its final value.
GAIN	Ratio measurement. An amplitude measurement of a target waveform is divided by an amplitude measurement of a reference waveform.
HIGH	The value used as 100% whenever HREF, MREF, and LREF values are needed (as in fall time and rise time measurements). Set to absolute value or calculated as the peak value or histogram, which finds the most common value. See *Methods of Setting HIGH and LOW* on page 2–38.
LOW	The value used as 0% whenever HREF, MREF, and LREF values are needed (as in fall time and rise time measurements). Set to absolute value or calculated as the peak value or histogram, which finds the most common value. See *Methods of Setting HIGH and LOW* on page 2–38.
MAXimum	Voltage measurement. The most positive peak voltage measured over the entire waveform.
MEAN or DC	Voltage measurement. The arithmetic mean over the entire waveform. Use MEAN or DC to select this measurement.
MID	The value half way between MAXimum and MINimum values. Units are in current vertical units. $MID = \frac{MAXimum - MINimum}{2}$
MINimum	Voltage measurement. The most negative peak voltage measured over the entire waveform.
NCROSs	Timing measurement. The time of the Nth negative-going crossing of the MREF (mid-reference) level, where N is set by CALC<n>:WMP:EDGE. Time is relative the trigger point of the acquisition.

www.valuetronics.com

Table 2–2: Measurement Definitions (Cont.)

Measurement Command	Definition
NDUTycycle (Negative Duty Cycle)	Timing measurement of the first cycle in the waveform. The ratio of the negative pulse width to the signal period, expressed as a percentage. $NDUTycycle = \frac{NegativeWidth}{Period} \times 100\%$
NWIDth (Negative Width)	Timing measurement of the first pulse in the waveform. The distance (time) between MREF (default 50%) amplitude points of a negative pulse.
PERiod	Timing measurement. Time for the first complete signal cycle to complete in the waveform. The reciprocal of frequency. Measured in seconds.
PDUTycycle (Positive Duty Cycle)	Timing measurement of the first cycle in the waveform. The ratio of the positive pulse width to the signal period, expressed as a percentage. $PDUTycycle = \frac{PositiveWidth}{Period} \times 100\%$
PCROSs	Timing measurement. The time of the Nth positive–going crossing of the MREF (mid-reference) level, where N is set by CALC<n>:WMP:EDGE. Time is relative the trigger point of the acquisition.
Phase	Timing measurement. The amount one waveform leads or lags another in time. Expressed in degrees, where 360° comprise one waveform cycle.
PTPeak (Peak to Peak)	Voltage measurement. The absolute difference between the MAXimum and MINimum amplitude in the entire waveform.
PWIDth (Positive Width)	Timing measurement of the first pulse in the waveform. The distance (time) between MREF (default 50%) amplitude points of a positive pulse.
RMS or AC	Voltage measurement. The true Root Mean Square voltage over the entire waveform. Use RMS or AC to select this measurement.
RTIMe (Rise time)	Timing measurement. Time for the leading edge of the first pulse to rise from the LREF value (default = 10%) to the HREF value (default = 90%).
σ **SDEViation (Standard deviation)**	Deviation from the MEAN measurement. The square root of the arithmetic mean of the squares of each measurement deviation from the measured MEAN. Result is in current vertical units.
TTRIG	Timing measurement. The time difference between the main and the delay triggers. Value returned is independent of channel number. Value returned is valid only when the delay trigger source is not set to immediate.

You can specify measurements to perform using either the SCPI model or the expression model. The models are discussed starting on page 2–32. The parameters used to make measurements are listed in Table 2–3.

Measurement Parameters

The automated measurements have parameters, such as LREF, MREF, and HREF, that determine how to make the various measurements. You can set the parameters with the CALCulate:WMParameter commands or use their default values. Table 2–3 lists the measurement parameters along with a description of

www.valuetronics.com

each. Following the table are discussions on the methods of setting parameters such as absolute or relative reference levels.

Table 2–3: Measurement Parameters

Measurement Parameter	Reset Value	Description
EDGE	1	Sets the edge(s) used for cross, ncross, pcross, and delay measurements.
HIGH	0.0	Sets the high or most positive level in current vertical units. Used for amplitude and time measurements when CALC:WMP:HMEThod is set to ABSolute.
LOW	0.0	Sets the low or most minimal level in current vertical units. Used for amplitude and time measurements when CALC:WMP:LMEThod is set to ABSolute.
HREFerence	0.0	Sets the high reference or distal level in current vertical units. Used for rise time (RTIM) and fall time (FTIM) measurements when CALC:WMP:RMEThod is set to ABSolute.
HREFerence:RELative	0.9 (90%)	Sets the high reference or distal level as a percentage of the current value for HIGH. Used for rise time (RTIM) and fall time (FTIM) measurements when CALC:WMP:RMEThod is set to RELative.
LREFerence	0.0	Sets the low reference or proximal level in current vertical units. Used for rise time (RTIM) and fall time (FTIM) measurements when CALC:WMP:RMEThod is set to ABSolute.
LREFerence:RELative	0.1 (10%)	Sets the low reference or proximal level as a percentage of the current value for HIGH. Used for rise time (RTIM) and fall time (FTIM) measurements when CALC:WMP:RMEThod is set to RELative.
MREFerence	0.0	Sets the middle reference or mesial level in current vertical units. Used for timing measurements when CALC:WMP:RMEThod is set to ABSolute.
MREFerence:HYSTeresis	0.05 (5%)	Sets the hysteresis value as a percentage of the current value for AMPLitude. Reduces effects of noise on measurements.
MREFerence:RELative	0.5 (50%)	Sets the middle reference or mesial level as a percentage of the current value for HIGH. Used for timing measurements when CALC:WMP:RMEThod is set to RELative.
SLOPe	POS	Sets the direction, positive- or negative-going, of the edge(s) used for cross and delay measurements.

www.valuetronics.com

Methods of Setting HIGH and LOW. The waveform analyzer provides four ways to setting the reference levels HIGH and LOW which are used in many amplitude and time measurements. The methods are as follows:

ABSolute — specifies that HIGH and LOW are set to absolute amplitude levels with CALCulate:WMParameter:HIGH and :LOW.

AUTO — selects the MODE method of setting HIGH and LOW when the histogram function is able to detect a consistent level above and below MID. Otherwise, the PEAK method is used. AUTO method is effective when you are not certain what type of waveform to expect.

MODE — selects the levels for HIGH and LOW based on a peak histogram function which looks for the most common value above and below MID. This method ignores short term aberrations, such as overshoot and ringing, on a digital logic waveform.

PEAK — specifies that HIGH and LOW are set to the highest and lowest values in the waveform record. This method is useful for a sinewave or triangle waveform but not for digital pulses.

You can choose to set HIGH and LOW using different methods. For more information on setting HIGH and LOW, refer to *Appendix B: Algorithms*.

Methods of Setting Reference Levels. Two methods are available for setting the reference levels LREF, MREF, and HREF. The methods are ABSolute and RELative which you pick with the command CALC:WMP:RMEThod. ABSolute lets you set the reference levels as absolute vertical levels. RELative lets you set them as a ratio or percentage of the value for the measurement parameter HIGH.

For more information on setting reference levels, refer to *Appendix B: Algorithms*.

Waveform Math

The waveform analyzer provides a means to mathematically manipulate your waveforms. You can add (+), subtract (–), multiply (*) or divide (/) two waveforms or a waveform and a scalar. The scalar can be the result of another measurement such as MEAN in the following example:

```
CALC1:PATH:EXPR (CHAN1 - (MEAN (CHAN1)))
```

All other CALC functions are compatible with the math operators. You can use the minus operator (–) to negate a scalar value. For more information on expression syntax, refer to *Expression Model* on page 2–33.

www.valuetronics.com

Waveform Differentiation

The calculation capabilities of the waveform analyzer include waveform differentiation. This capability allows you to compute a derivative math waveform that indicates the instantaneous rate of change of the waveform acquired. Derivative waveforms are useful in the measurement of the slew rate of amplifiers.

Use the command CALC<n>:DERivative:STATe to enable differentiation for a particular CALC block. For example, when CALC2:DER:STATe is ON, the CALC2 block will perform differentiation on the CALC2 source.

The resultant waveform is referred to as CALC<n> where <n> is the CALC block that performed the differentiation. Each of the four CALC blocks can perform differentiation when enabled. Use TRACe? CALC<n> to retrieve the resultant waveform.

The math waveform, derived from the sampled waveform, is computed based on the following equation:

$$Y_n = (X_{(n+1)} - X_n)\frac{1}{T}$$

Where: X is the source waveform
Y is the derivative math waveform
T is the time between samples

Since the resultant math waveform is a derivative waveform, its vertical units are volts/second (its horizontal units are seconds). The source signal is differentiated over its entire record length; therefore, the math waveform record length equals that of the source waveform.

Waveform Integration

The calculation capabilities of the waveform analyzer include waveform integration. This capability allows you to compute an integral math waveform that is an integrated version of the acquired waveform.

Use the command CALC<n>:INTegral:STATe to enable integration for a particular CALC block. For example, when CALC2:INT:STATe is ON, the CALC2 block will perform integration on the CALC2 source.

The resultant waveform is referred to as CALC<n> where <n> is the CALC block that performed the calculation. Each of the four CALC blocks can perform integration when enabled. Use TRACe? CALC<n> to retrieve the resultant waveform record.

The integral waveform record, derived from the sampled waveform, is computed based on the following equation:

$$y(n) = scale \sum_{i=1}^{n} \frac{x(i) + x(i-1)}{2} T$$

www.valuetronics.com

Where: x(i) is the source waveform
y(n) is a point in the integral waveform
scale is the output scale factor
T is the time between samples (sample interval)

Since the resultant waveform record is an integral waveform, its vertical units are volt-seconds (its horizontal units are seconds). The source signal is integrated over its entire record length; therefore, the waveform record length equals that of the source waveform.

Transforms (FFT)

The calculation capabilities of the waveform analyzer include taking the Fast Fourier Transform (FFT) of a waveform record. The FFT allows you to transform an amplitude versus time waveform into one that plots the amplitudes of the various discrete frequencies the waveform contains.

The FFT computes the frequency content of a waveform you specify as a CALC block source. The transform of the source waveform is based on the following equation:

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{\frac{j2\pi nk}{N}} \quad for: k = 0 \ to \ \frac{N}{2}$$

Where: x(n) is a point in the time domain record
X(k) is a point in the frequency domain record
n is the index to the time domain record
k is the index to the frequency domain record
N is the length of the time domain record
j is the square root of −1

The resulting waveform is a record consisting of N/2 complex coefficients representing values in the frequency domain. The horizontal scale for FFT waveforms is expressed in frequency with the first point of the waveform record representing zero frequency (DC).

Setting the Record Format. You can convert the complex data in the resultant transform record into magnitude or phase data using the CALC:FORMat function. The following lines set the waveform analyzer to perform a TRANsform on CH 1 and format the result as phase data:

```
CALC1:FORM PHAS
CALC1:PATH:EXPR (FORM(TRAN(CHAN1)))
```


www.valuetronics.com

Zero Phase Reference Point. The zero phase reference point for an FFT phase waveform is in the middle of the FFT time domain record regardless of the waveform record length. Figure 2–20 shows this placement.

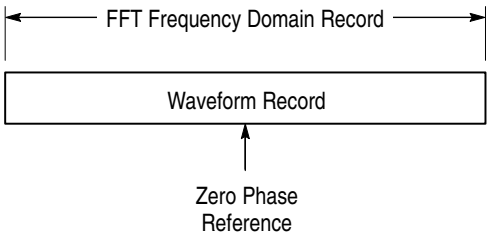


Figure 2–20: Zero phase reference point in FFT phase records

FFT Frequency Range and Resolution. The waveform analyzer can return either the magnitude or phase angle of the FFT frequency domain record. The resolution between the discrete frequencies in this waveform is determined by the following equation:

$$\Delta F = \frac{Sample\ Rate}{Input\ Length}$$

Where:

ΔF is the frequency resolution.
Sample Rate is the sample rate of the source waveform.
Input Length is the length of the source waveform record.

The sample rate also determines the range these frequencies span; they span from 0 to ½ the sample rate of the waveform record. (The value of ½ the sample rate is often referred to as the Nyquist frequency or point.) For example, a sample rate of 20 Megasamples per second would yield an FFT with a range of 0 to 10 MHz.

Undersampling (Aliasing). Aliasing occurs when the waveform analyzer acquires a source waveform with frequency components outside of the frequency range for the current sample rate. In an FFT waveform record, the actual higher frequency components are undersampled, and therefore, they appear as lower frequency aliases that "fold back" around the Nyquist point (½ the sample rate). See Figure 2–21.

- Source waveforms with fast edge transition times create many high frequency harmonics. These harmonics typically decrease in amplitude as their frequency increases.
- Sample the source signal at rates that are at least 2X that of the highest frequency component having significant amplitude.

www.valuetronics.com

- If necessary, filter the input to limit its bandwidth to frequencies below the Nyquist frequency.

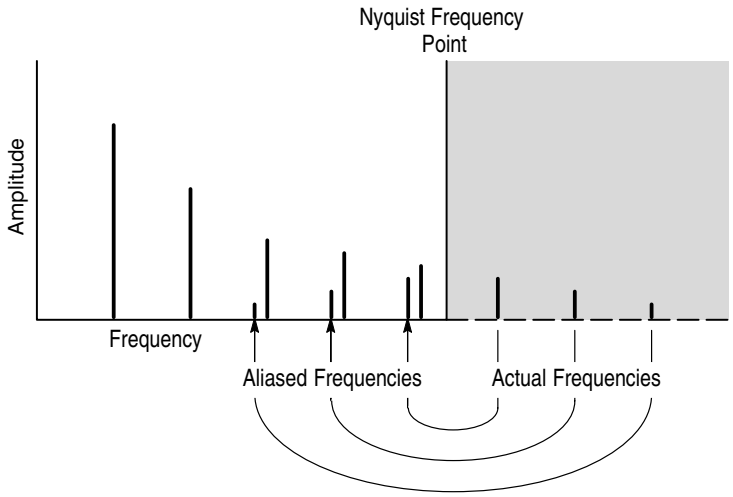


Figure 2–21: How aliased frequencies corrupt an FFT transform

FFT Windows. An FFT window acts like a bandpass filter between the time domain record and the FFT frequency domain record. The shape of the window controls the ability of the FFT to resolve (separate) frequencies and to accurately measure the amplitude of those frequencies. Figure 2–22 shows how a window is used in the transform process.

You can select from six windows for the transform with the command CALC:TRAN:FREQ:WIND. Each window provides benefits and losses. For example, if you select a window to provide better frequency resolution you give up a degree of amplitude accuracy. In general, choose a window that can just resolve the frequencies you want to measure. The available windows and their characteristics are as follows:

RECTangular — Best type for resolving a narrow band of frequencies but worst for accurate amplitude of those frequencies. Good for measuring nonrepetitive signals and frequency components near DC.

HAMMing — Very good window for resolving frequencies that are very close to the same value with somewhat improved amplitude accuracy over the rectangular window.

HANNing — Very good window for measuring amplitude accuracy but degraded for resolving frequencies.

BHARris — Widest bandwidth and lowest side lobes. Best for viewing a broad spectrum.

www.valuetronics.com

BLACkman — Best window for measuring the amplitude of frequencies but worst at resolving frequencies.

TRIangular — Window with the least attenuation of side lobes.

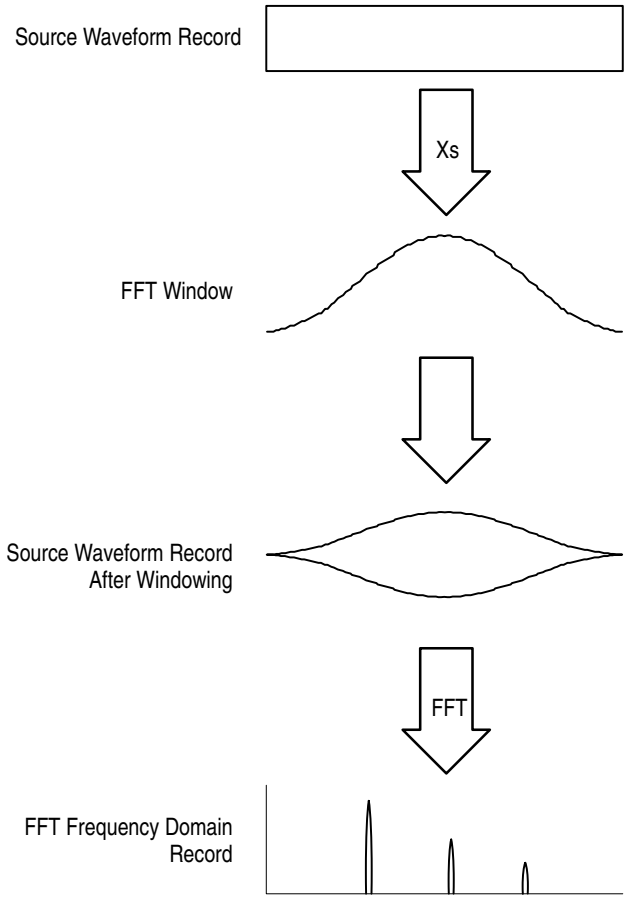


Figure 2–22: Windowing the FFT time domain record

Figure 2–23 shows each window, its bandpass characteristic, bandwidth, and highest side lobe. When choosing a window, consider the following characteristics:

- The narrower the central lobe for a given window, the better it can resolve a frequency.
- The lower the lobes on the side of each central lobe are, the better the amplitude accuracy of the frequency measured in the FFT using that window.
- Hamming, Hanning, Blackman, and BHarris are all bell-shaped windows that taper the waveform record at the ends. The Hanning and BHarris windows taper the data at the end of the record to zero; therefore, they are generally

www.valuetronics.com

better choices to eliminate leakage. However, be certain to position the most interesting parts of the signal in the center region of the waveform record.

- If the Hanning window merges frequencies, try the Hamming window before settling on the rectangular window. Depending on the distance of the frequencies you are trying to measure from the fundamental, the Hamming window sometimes resolves frequencies better than the Hanning.

www.valuetronics.com

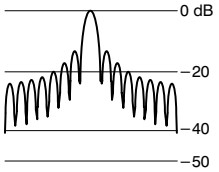
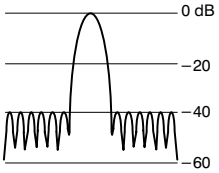
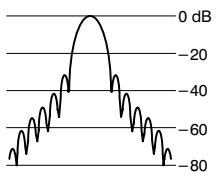
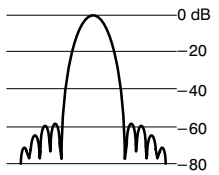
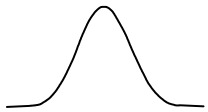
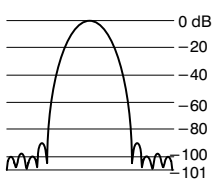
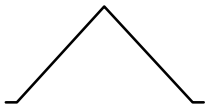
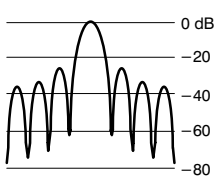
FFT Window Type	Bandpass Filter	–3 dB Bandwidth	Highest Side Lobe
Rectangular Window		0.89	–13 dB
Hamming Window		1.30	–43 dB
Hanning Window		1.44	–32 dB
Blackman Window		1.68	–58 dB
 BHarris Window		1.90	–92 dB
 Triangle Window		1.28	–27 dB

Figure 2–23: FFT windows and bandpass characteristics

Leakage. Leakage results when the time domain waveform delivered to the TRANsform (FFT) function contains a non-integer number of waveform cycles. Since there are fractions of cycles in such records, there are discontinuities at the ends of the record. These discontinuities cause energy from each discrete

www.valuetronics.com

frequency to “leak” over on to adjacent frequencies. The result is amplitude error when measuring those frequencies.

Digital Filtering

The waveform analyzer provides a configurable digital filter to perform post-acquisition frequency filtering. There are four types of filters available:

BPASs — rejects frequencies outside the defined frequency range.

HPASs — rejects frequencies below the defined frequency range.

LPASS — rejects frequencies above the defined frequency range.

NOTCh — rejects frequencies within the defined frequency range.

Use the command CALC:FILT:FREQ[:TYPE] to select a digital filter. Once you have selected a filter and configured it (see the following discussion) you must enable the filter with CALC:FILT:FREQ:STATe when using the standard SCPI model.

Filter Parameters. Each digital filter function has a set of parameters that determine its effective frequency range and roll off. The LPAS and HPASs filters have specific cutoff parameters. Other parameters affect the roll off or transition region. Figure 2–24 shows the main parameters that control the four digital filters and the commands used to set them. Note that the parameters set the –6dB point of attenuation.

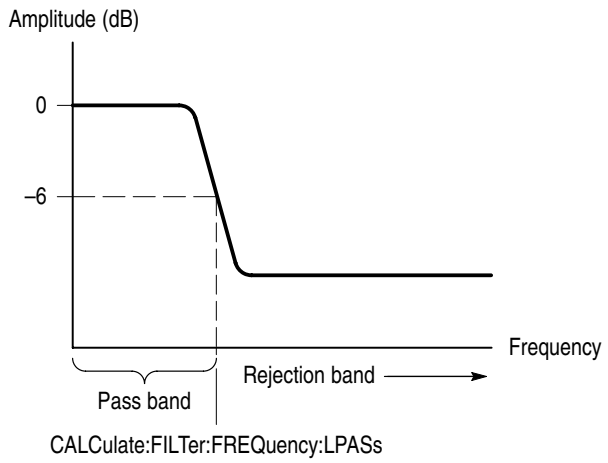
The NOTCh and BPASs bandpass filters are set with either of two command pairs: CENTer and SPAN or STARt and STOP. Figure 2–25 shows how these command pairs effectively control the same parameters. Setting a parameter of one pair affects the settings of the other pair. For example, if you could set the BPAS filter parameter STARt to 50 MHz and STOP to 100 MHz using the following command.

```
CALC1:FILT:FREQ BPAS; FREQ:STAR 50E6; STOP 100E6
```

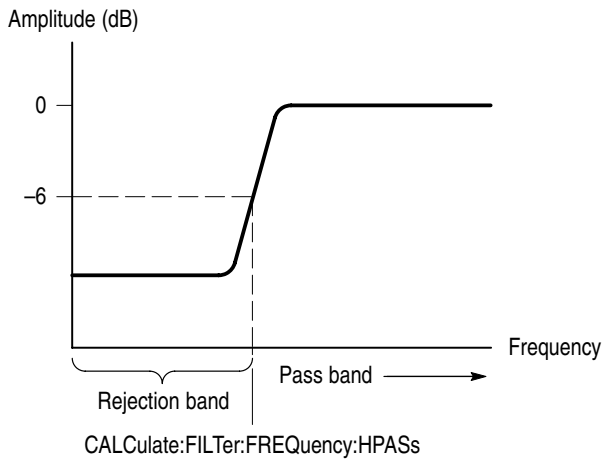
A query of CENTer and SPAN will return 75 MHz and 50 MHz, respectively. If you then set SPAN to 100 MHz and leave CENTer at 75 MHz, the STARt value changes to 25 MHz and the STOP value changes to 125 MHz. The new SPAN value is spread on either side of CENTer requiring a change in the ends of the pass band.

www.valuetronics.com

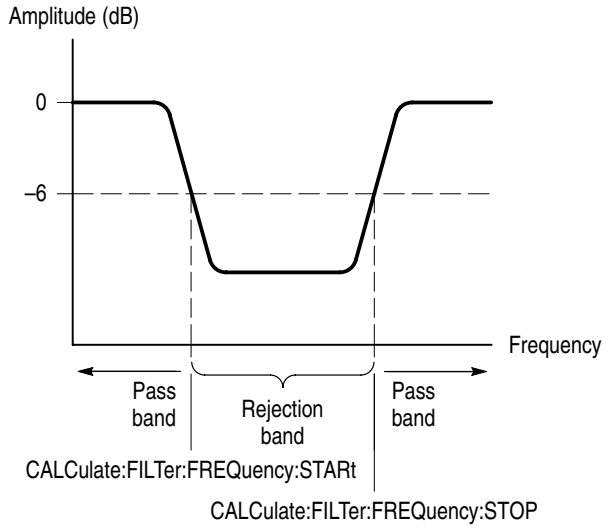
Low pass filter



High pass filter



Notch filter



Bandpass filter

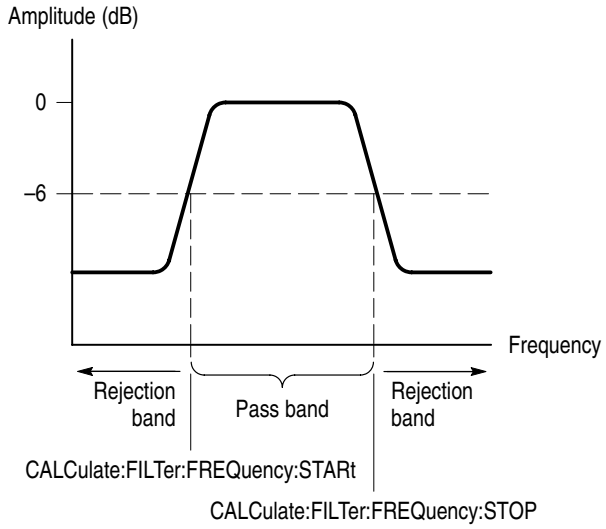


Figure 2–24: Parameters for the four digital filters

www.valuetronics.com

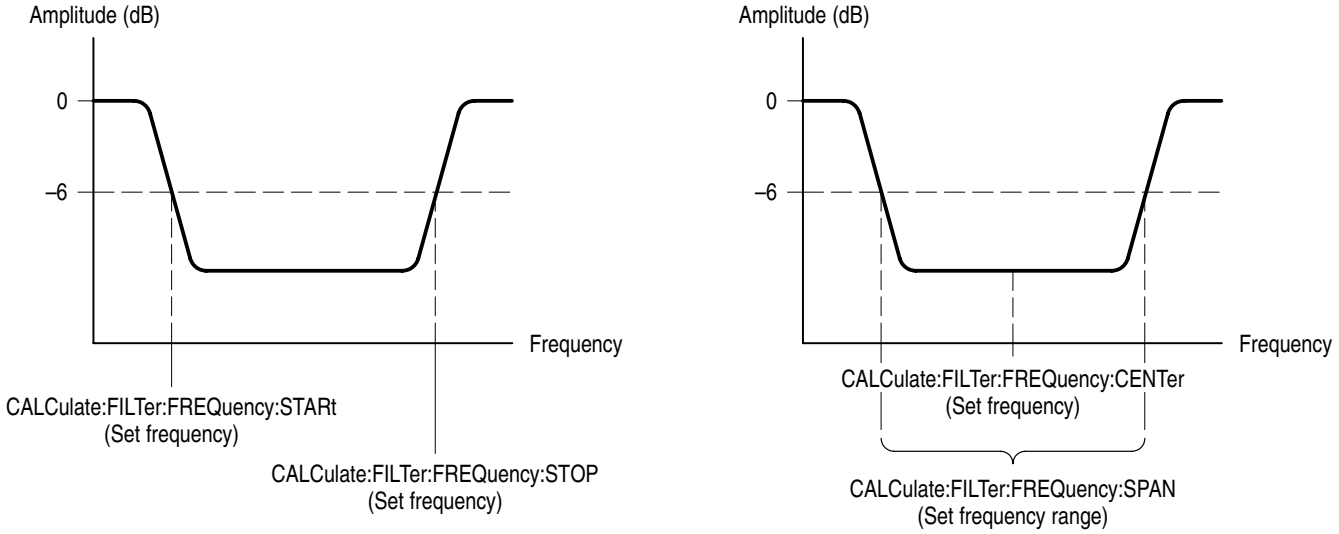


Figure 2–25: Two methods of setting BPASs and NOTCh filters

Transition Settings. The roll off or transition region between the pass band and the rejection band for all filter functions is controlled by the parameters SREJection and TWIDth. Figure 2–26 shows these parameters. SREJection controls the degree of attenuation and is specified in dB. The minimum setting is 15 dB and the maximum is 100 dB. TWIDth determines the rate of roll off or transition width. TWIDth is specified as a value between 0 and 1 which is derived using the following relation:

TWIDth = Transition Frequency * 2 * SWEep:TINTerval

The transition frequency is the actual range of frequencies over which the filter transitions from the pass band to the rejection band. SWEep:TINTerval is the current sample interval. The smaller the value of TWIDth the smaller the transition region and steeper the transition slope. To set TWIDth to 0.05 you would use the command

```
CALC1:FILT:FREQ:TWID 0.05
```


www.valuetronics.com

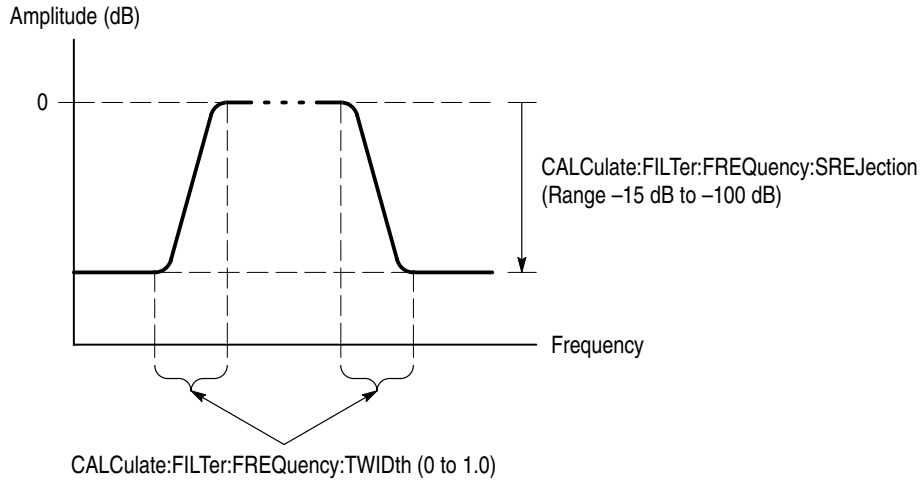


Figure 2–26: Rejection level and transition slope for the digital filter

For more information on the digital filter, refer to the Filter discussion in *Appendix B, Algorithms*.

www.valuetronics.com

Fast Data Channel

The Fast Data Channel (FDC) provides a fast transfer protocol for moving waveform records and other data between the waveform analyzer and a VXIbus controller. The waveform analyzer includes an FDC driver that provides the instrument side of the protocol.

The controller must initiate FDC transfers; therefore, you must ensure that your controller has an FDC driver to communicate correctly with the waveform analyzer FDC channel. Refer to your VXI system communications software manual to learn how to initiate an FDC channel on your controller. See ***Note*** below.

NOTE*. The VXIplug&play VISA I/O interface provides FDC support. The sections* Fast Data Channel *and* Auto-Advance Acquisition *on page 2–16 discusses setting up an FDC session using this interface. In particular, see step 10 on page 2–18 of that section for use of the VISA commands to configure the FDC session.*

Table 2–4 lists the commands that control the instrument side of the FDC protocol.

Table 2–4: Commands for Fast Data Channel and Associated Functions

Command	Description
TRACe:COPY FDC1,<source>	Copies waveform records or CALCulate block results to the outgoing Fast Data Channel (FDC1). You can specify one or more sources in a comma separated list. One selection is AATS which specifies all auto-advance waveform records. Alternatively, use TRAC:OUT:FDC to specify a list of items to transfer then use TRAC:COPY,LIST to transfer all the list of items.
TRACe:OUTput:FDC	Defines a list of waveform records, calculations, and the output of auto-advance acquisition to transfer with the FDC.

The following example command transfers CH 1 and CH 2 waveform records plus the results of the CALC1 and CALC2 blocks. (The transfer occurs immediately upon sending the command; it does not wait for acquisition or calculations to complete.)

```
TRAC:COPY FDC1,CHAN1,CHAN2,CALC1,CALC2
```


www.valuetronics.com

***NOTE**. Refer to* Fast Data Channel *and* Auto-Advance Acquisition *on page 2–16 for an example of setting the waveform analyzer for auto-advance acquisition with FDC waveform transfers.*

FDC Code Example

The following C-code example sets up an FDC channel, configures the TVS600, transfers a 512-point waveform, and closes the FDC channel. The program assumes the controller uses the VXI Plug&Play Interface to control the FDC session; users of other interfaces supplying FDC support should modify the program to support those interfaces.

The program simply exits if an error occurs. You should add statements to the program so it closes all sessions before exiting the program after an error.

The transferred waveform data is stored in a buffer (waveformBuf) as binary bytes characters. For a 512-point record, the received data includes 1024 bytes plus a few bytes of header information. The example program ends with the appropriate statements to close all sessions.

***NOTE**. This program was written in MicroSoft C QuickWin. It must be compiled with Microsoft Visual C++ as a QuickWin application. It must be compiled using the large memory model.*

www.valuetronics.com

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "c:\vxipnp\win\include\visa.h"

#define WAVEFORM_BUF_SIZE 2048 */Buffer sized for 512 pt record */

void main() {
   ViStatus   status; /* Return value from VISA.DLL functions */
   ViSession  visaRM; /* VISA resource manager session handle */
   ViSession  tvs600; /* Session handle for command channel */
   ViSession  fdc;    /* Session handle for FDC channel */
   ViChar tvs600Desc[256]; /* Descriptor used to address TVS600 */
   ViChar waveformBuf[WAVEFORM_BUF_SIZE]; /* Local waveform memory */
   ViUInt32   retCnt; /* Returns actual number of bytes transferred */

   /*****************************************************************/
   /*                                                               */
   /*  Establish Communication with the TVS600                      */
   /*                                                               */
   /*****************************************************************/

   /* Set the TVS600 descriptor */
   /* Customized following line with the logical address of TVS600. */
   /* The address here is 2. */
   strcpy(tvs600Desc, "VXI::2::INSTR");

   /* Open session with the VISA default resource manager */
   status = viOpenDefaultRM(&visaRM);
   if (status) return;

   /* Open a session with the tvs600 to send ASCII commands */
   status = viOpen(visaRM, tvs600Desc, VI_NULL, VI_NULL, &tvs600);
   if (status != VI_SUCCESS) return;

   /* Open a session with the tvs600 to receive FDC data */
   status = viOpen(visaRM, tvs600Desc, VI_NULL, VI_NULL, &fdc);
   if (status != VI_SUCCESS) return;
```

Example continued on following page.

www.valuetronics.com

Example continued from previous page

```
/*****************************************************************/
/*                                                               */
/*  Set up the FDC session attributes                            */
/*                                                               */
/*****************************************************************/
/* Selects FDC channel 1. */
/* The TVS600 outputs data on FDC channel 1.  */
   status = viSetAttribute(fdc, VI_ATTR_FDC_CHNL, 1);
   if (status != VI_SUCCESS) return;

/* The TVS600 uses a stream channel. A stream channel can */
/* send a series of waveforms more efficiently.              */
status = viSetAttribute(fdc, VI_ATTR_FDC_MODE, VI_FDC_STREAM);
if (status != VI_SUCCESS) return;

/* Select the FDC protocol for this session */
status = viSetAttribute(fdc, VI_ATTR_IO_PROT, VI_FDC);
if (status != VI_SUCCESS) return;

/* Set the FDC session timeout to 10 seconds */
status = viSetAttribute(fdc, VI_ATTR_TMO_VALUE, 10000);
if (status != VI_SUCCESS) return;

/*****************************************************************/
/*                                                               */
/*  Set up the TVS600 acquisition                                */
/*                                                               */
/*****************************************************************/
/* Initialize the TVS600 to a known state */
if (viPrintf(tvs600, "ABORT\n") != VI_SUCCESS) return;
if (viPrintf(tvs600, "*RST\n")  != VI_SUCCESS) return;

/* Set up channel 1 for the acquisition: */
/* Ch 1 50 ohm, 512 pts record, 100 MS/s, transfer Ch 1 only */
if (viPrintf(tvs600, "INP:IMP 50\n")      != VI_SUCCESS) return;
if (viPrintf(tvs600, "VOLT1:RANG 0.5\n")  != VI_SUCCESS) return;
if (viPrintf(tvs600, "FUNC:ON CHAN1\n")   != VI_SUCCESS) return;
if (viPrintf(tvs600, "FUNC:OFF CHAN2\n")  != VI_SUCCESS) return;
if (viPrintf(tvs600, "DATA:LIST CHAN1\n") != VI_SUCCESS) return;
if (viPrintf(tvs600, "SWE:POIN 512\n")    != VI_SUCCESS) return;
if (viPrintf(tvs600, "SWE:TINT 10nS\n")   != VI_SUCCESS) return;
if (viPrintf(tvs600, "INIT\n")            != VI_SUCCESS) return;
```

Example continues on following page

www.valuetronics.com

Example continued from previous page

```
    /*****************************************************************/
    /*                                                               */
    /*  Start transfer of TVS600 acquired data using FDC.            */
    /*                                                               */
    /*****************************************************************/
    status = viRead(fdc, waveformBuf, WAVEFORM_BUF_SIZE, &retCnt);
    if (status != VI_SUCCESS) return;

    /*****************************************************************/
    /*                                                               */
    /*  Close all sessions to deallocate resources and memory        */
    /*                                                               */
    /*****************************************************************/
    /* Close the command session */
    status = viClose(tvs600);
    if (status != VI_SUCCESS) return;
    /* Close the FDC session */
    status = viClose(fdc);
    if (status != VI_SUCCESS) return;
    /* Close the VISA resource manager */
    status = viClose(visaRM);
    if (status != VI_SUCCESS) return;
}
```


www.valuetronics.com

Data Transfer Formats

The waveform analyzer allows you to set the format of transferred data. Acquired data can be in ASCII integer or 16 bit binary formats. CALC data is always floating point, but may be ASCII or 32 REAL type. The primary format commands are in the FORMat subsystem.

The defined binary formats send the high byte (most significant) followed by lower bytes by default. You can change the byte order with FORMat:BORDer to either NORM (high byte/low byte) or SWAP (low byte/high byte).

Acquired Data Formats

The waveform analyzer uses signed, 16-bit integers to represent each data point in a waveform record. The waveform data are transferred as either decimal ASCII integers (ASCII,0) or 16 bit binary numbers (INT,16). You choose a format with the FORMat command. Data values range from –32767 to 32767. The commands DATA? and TRAC? return waveform data to the requesting interface. To get scaling information for the waveform, retrieve the waveform preamble for each waveform with the command TRACe:PREamble?. Refer to page 2–57 for a listing of format for the various waveform preambles.

ASCII Waveform Data. ASCII waveform data is represented by signed integer values. Each data value requires two to seven characters. For each data value, one to five characters to represent the value, a character to represent a minus sign for negative values, and a comma to separate data points.

An example ASCII waveform record may look like this:

```
0,-23,-64,-47,-15,2,87,324,989,1952,29847,. . .
```

The number of data bytes returned is determined by the length of the waveform record and the number of characters in each data value.

Binary Waveform Data. Binary waveform data is represented by 16 bit integer values and is used for normal trace acquisition. The data ranges from –32767 to 32767. Every data value is represented by two data bytes as shown in the following example:

www.valuetronics.com

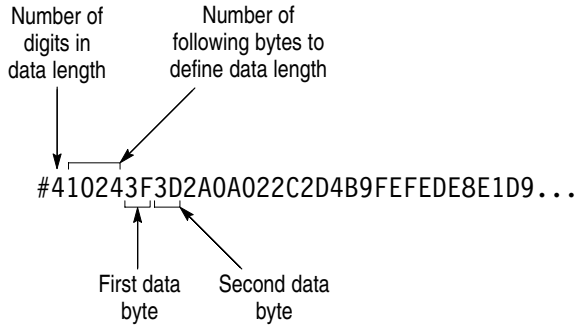


Figure 2–27: Binary transfer format

You can reverse the order of the bytes with the command FORMat:BORder.

Overrange and Underrange Values. When the input signal exceeds the range of a channel, special values are stored in the waveform record. Overrange points are stored as +32767 and underrange points as –32767.

CALCulate and AATS Data Formats

The waveform analyzer uses 32-bit, floating point values for calculated data including the AATS time stamp record. This data may be transferred as either ASCII (ASC,0) or binary (REAL,32) floating point values. You specify the data format for the four CALC blocks with the FORMat:CALC<n> commands and the time stamp format with the FORM:TRAC:AATS command.

ASCII CALC Data. ASCII data from a CALC block can be transferred by ASCII 32-bit, floating point values. Each data value includes a minus sign (if negative), a fraction, followed by an exponent. Commas separate the data points.

An example ASCII waveform record may look like this:

```
2E-6,5.34E-6,2.21E-4,4.65E-2,5.6E1,7.91E2,. . .
```

The number of data bytes returned is determined by the length of the waveform record and the number of characters in each data value.

Binary CALC Data. Binary data from a CALC block can be transferred as IEEE Standard 32-bit floating point values.

A 32 bit binary, floating point number is formatted as follows:

<sign 1 bit><exponent 8 bits><fraction 23 bits>

Figure 2–27 shows the basic binary transfer format. The 32-bit numbers are sent as four bytes. The first or high byte will contain the sign bit plus 7 exponent bits. The second byte contains the remaining 1 exponent bit and 7 fraction bits.

www.valuetronics.com

You can reverse the order of the bytes with the command FORMat:BORder. However, a similar swapping for 16 bit words is not provided.

Overrange, Underrange, and Null Values. When the calculated values exceed the numerical limits, special values are stored in the calculation record as defined by IEEE 754–1985. Different values are stored for ASCII,0 and REAL,32 data types as follows:

- Overrange
  - ASCII,0 +9.9e+37
  - REAL,32 +∞
- Underrange
  - ASCII,0 –9.9e+37
  - REAL,32 –∞
- Null
  - ASCII,0 +9.91e+37
  - REAL,32 NaN

Waveform Preambles

Each waveform that you transfer has an associated waveform preamble. The waveform preamble is a separate data record that contains information about the waveform such as the horizontal scale, the vertical scale, and other settings in place when the waveform was created. The format of the waveform preamble is determined by the settings of the FORMat and the CALC:FORMat commands. Table 2–5 shows the possible waveform preamble formats. Table 2–6 on page 2–61 defines many of the keywords used in the waveform preambles.

www.valuetronics.com

Table 2–5: Waveform Preambles and Their Formats

Preamble Type	Preamble Format
CALCulate Data (binary)	DIF(VERS 1995.0 SCOP PRE) IDEN(NAME "CALC1" INST(NAME "TVS641" ID "B010100")) ENC(FORM IFP32) DIM=X(TYPE IMPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "S") DIM=Y(TYPE EXPL SIZE <nr1> UNIT "V") DATA(CURV(CTYP NONE))
CALCulate Data (ASCII)	DIF(VERS 1995.0 SCOP PRE) IDEN(NAME "CALC1" INST(NAME "TVS641" ID "B010100")) ENC(FORM ASC NVAL 9.91E+37 ORAN 9.9E+37 URAN -9.9E+37) DIM=X(TYPE IMPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "S") DIM=Y(TYPE EXPL SIZE <nr1> UNIT "V") DATA(CURV(CTYP NONE))
SENSe Data (binary)	DIF(VERS 1995.0 SCOP PRE) IDEN(NAME "CHAN1" INST(NAME "TVS641" ID "B010100")) ENC(FORM INT16 NVAL -32768 ORAN 32767 URAN -32767) DIM=X(TYPE IMPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "S") DIM=Y(TYPE EXPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "V") DATA(CURV(CTYP NONE))
SENSe Data (ASCII)	DIF(VERS 1995.0 SCOP PRE) IDEN(NAME "CHAN1" INST(NAME "TVS641" ID "B010100")) ENC(FORM ASC NVAL -32768 ORAN 32767 URAN -32767) DIM=X(TYPE IMPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "S") DIM=Y(TYPE EXPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "V") DATA(CURV(CTYP NONE))
SENSe Data (binary auto-advance)	DIF(VERS 1995.0 SCOP PRE) IDEN(NAME "CHAN1" INST(NAME "TVS641" ID "B010100")) ENC(FORM INT16 NVAL -32768 ORAN 32767 URAN -32767) DIM=REC(TYPE IMPL SIZE <nr1> UNIT "") DIM=X(TYPE IMPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "S") DIM=Y(TYPE EXPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "V") DATA(CURV(CTYP NONE))
SENSe Data (ASCII auto-advance)	DIF(VERS 1995.0 SCOP PRE) IDEN(NAME "CHAN1" INST(NAME "TVS641" ID "B010100")) ENC(FORM ASC NVAL -32768 ORAN 32767 URAN -32767) DIM=REC(TYPE IMPL SIZE <nr1> UNIT "") DIM=X(TYPE IMPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "S") DIM=Y(TYPE EXPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "V") DATA(CURV(CTYP NONE))

www.valuetronics.com

Table 2–5: Waveform Preambles and Their Formats (Cont.)

Preamble Type	Preamble Format
SENSe Data (binary envelope)	DIF(VERS 1995.0 SCOP PRE) IDEN(NAME "CHAN1" INST(NAME "TVS641" ID "B010100")) ENC(FORM INT16 NVAL -32768 ORAN 32767 URAN -32767) DIM=X(TYPE IMPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "S") DIM=UPP(TYPE EXPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "V") DIM=LOW(TYPE EXPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "V") TRAC=TU(IND(LAB X) DEP(LAB UPP)) TRAC=TL(IND(LAB X) DEP(LAB LOW)) VIEW=ENVELOPE(ENV(UPP TU LOW TL)) DATA(CURV(CTYP NONE))
SENSe Data (ASCII envelope)	DIF(VERS 1995.0 SCOP PRE) IDEN(NAME "CHAN1" INST(NAME "TVS641" ID "B010100")) ENC(FORM ASC NVAL -32768 ORAN 32767 URAN -32767) DIM=X(TYPE IMPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "S") DIM=UPP(TYPE EXPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "V") DIM=LOW(TYPE EXPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "V") TRAC=TU(IND(LAB X) DEP(LAB UPP)) TRAC=TL(IND(LAB X) DEP(LAB LOW)) VIEW=ENVELOPE(ENV(UPP TU LOW TL)) DATA(CURV(CTYP NONE))
CALCulate Data (binary envelope)	DIF(VERS 1995.0 SCOP PRE) IDEN(NAME "CALC1" INST(NAME "TVS641" ID "B010100")) ENC(FORM IFP32) DIM=X(TYPE IMPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "S") DIM=UPP(TYPE EXPL SIZE <nr1> UNIT "V") DIM=LOW(TYPE EXPL SIZE <nr1> UNIT "V") TRAC=TU(IND(LAB X) DEP(LAB UPP)) TRAC=TL(IND(LAB X) DEP(LAB LOW)) VIEW=ENVELOPE(ENV(UPP TU LOW TL)) DATA(CURV(CTYP NONE))
CALCulate Data (ASCII envelope)	DIF(VERS 1995.0 SCOP PRE) IDEN(NAME "CALC1" INST(NAME "TVS641" ID "B010100")) ENC(FORM ASC NVAL 9.91E+37 ORAN 9.9E+37 URAN -9.9E+37) DIM=X(TYPE IMPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "S") DIM=UPP(TYPE EXPL SIZE <nr1> UNIT "V") DIM=LOW(TYPE EXPL SIZE <nr1> UNIT "V") TRAC=TU(IND(LAB X) DEP(LAB UPP)) TRAC=TL(IND(LAB X) DEP(LAB LOW)) VIEW=ENVELOPE(ENV(UPP TU LOW TL)) DATA(CURV(CTYP NONE))

www.valuetronics.com

Table 2–5: Waveform Preambles and Their Formats (Cont.)

Preamble Type	Preamble Format
CALCulate Data (binary complex)	DIF(VERS 1995.0 SCOP PRE) IDEN(NAME "CALC1" INST(NAME "TVS641" ID "B010100")) ENC(FORM IFP32) DIM=X(TYPE IMPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "S") DIM=REAL(TYPE EXPL SIZE <nr1> UNIT "V") DIM=IMAG(TYPE EXPL SIZE <nr1> UNIT "V") TRAC=TR(IND(LAB X) DEP(LAB REAL)) TRAC=TI(IND(LAB X) DEP(LAB IMAG)) VIEW=COMPLEX(RCOM(REAL TR IMAG TI)) DATA(CURV(CTYP NONE))
CALCulate Data (ASCII complex)	DIF(VERS 1995.0 SCOP PRE) IDEN(NAME "CALC1" INST(NAME "TVS641" ID "B010100")) ENC(FORM ASC NVAL 9.91E+37 ORAN 9.9E+37 URAN -9.9E+37) DIM=X(TYPE IMPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "S") DIM=REAL(TYPE EXPL SIZE <nr1> UNIT "V") DIM=IMAG(TYPE EXPL SIZE <nr1> UNIT "V") TRAC=TR(IND(LAB X) DEP(LAB REAL)) TRAC=TI(IND(LAB X) DEP(LAB IMAG)) VIEW=COMPLEX(RCOM(REAL TR IMAG TI)) DATA(CURV(CTYP NONE))
CALCulate Data (binary polar)	DIF(VERS 1995.0 SCOP PRE) IDEN(NAME "CALC1" INST(NAME "TVS641" ID "B010100")) ENC(FORM IFP32) DIM=X(TYPE IMPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "S") DIM=MAGN(TYPE EXPL SIZE <nr1> UNIT "V") DIM=PHAS(TYPE EXPL SIZE <nr1> UNIT "RAD") TRAC=TM(IND(LAB X) DEP(LAB MAGN)) TRAC=TP(IND(LAB X) DEP(LAB PHAS)) VIEW=POLAR(PCOM(MAGN TM PHAS TP)) DATA(CURV(CTYP NONE))
CALCulate Data (ASCII polar)	DIF(VERS 1995.0 SCOP PRE) IDEN(NAME "CALC1" INST(NAME "TVS641" ID "B010100")) ENC(FORM ASC NVAL 9.91E+37 ORAN 9.9E+37 URAN -9.9E+37) DIM=X(TYPE IMPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT "S") DIM=MAGN(TYPE EXPL SIZE <nr1> UNIT "V") DIM=PHAS(TYPE EXPL SIZE <nr1> UNIT "RAD") TRAC=TM(IND(LAB X) DEP(LAB MAGN)) TRAC=TP(IND(LAB X) DEP(LAB PHAS)) VIEW=POLAR(PCOM(MAGN TM PHAS TP)) DATA(CURV(CTYP NONE))

www.valuetronics.com

Table 2–6: Parameter Definitions for Preamble Elements

Preamble Element	Description
CTYPe	Indicates the checksum type.
CURVe	Indicates the data that follows is dimensioned.
DATA	Begins the block containing the actual waveform data. When the preamble is requested, the DATA block is empty.
DEPendent	Indicates a dependent DIMension that constitutes the TRACe.
DIF	Data Interchange Format. Always followed by the SCPI DIF version and the block type, such as SCOPe PREamble.
DIM=LOW()	Defines the type, scale, offset, size, and units for the lower envelope of the vertical dimension of envelope waveform data.
DIM=UPP()	Defines the type, scale, offset, size, and units for the upper envelope of the vertical dimension of envelope waveform data.
DIM=X()	Defines the type, scale, offset, size and units for the horizontal dimension of the waveform data.
DIM=Y()	Defines the type, scale, offset, size, and units for the vertical dimension of non-envelope waveform data.
ENCode	Describes the FORMat used to encode the associated waveform data. The FORMat may be one of the following: ■ ASC 8 bit ASCII characters ■ IFP32 32 bit binary floating point ■ INT16 16 bit binary integer
EXPLicit	Values for the dimension are present in DATA(CURVe).
IDENtify	Identifies the source of the data and the instrument used to acquire the data.
IMPLicit	Values for this dimension are derived from a linear function, y=mx + b.
LABel	Uniquely identifies the DIMension statement that applies to a specified trace data block. Only labels defined by DIM blocks, such as X, Y, UPP, or LOW, can appear in a LABel statement.
NVALue	Defines the numeric value of NaN, Not a Number in the data block.
OFFSet	Describes an offset value to be added to data in DATA(CURVe) after the SCALe factor is applied.
ORANge	Defines the numeric value of over range values in the data block.
SCALe	Describes the scaling factor to be applied to the data in DATA(CURVe).
SIZE	Defines the number of data points in the DATA(CURVe) waveform.

www.valuetronics.com

Table 2–6: Parameter Definitions for Preamble Elements (Cont.)

Preamble Element	Description
TRACe	Describes relationships between DIMensions. TRACe statements define a name for the correlation between vertical and horizontal DIMensions.
TYPE	Indicates if the dimension is implicit or explicit.
UNIT	Describes the correct units for the data, such as Seconds, Volts, and Hz.
URANge	Defines the numeric value of under range values in the data block.
VIEW	Describes how the defined TRACes are to be interpreted. Using an envelope waveform as an example, the VIEW block defines which TRACe is the upper vertical information and which is the lower.

www.valuetronics.com

Instrument I/O

The waveform analyzer provides VXIbus and RS-232C communication ports for instrument control. Additionally, you can use a GPIB controller and GPIB Slot 0 card to control VXIbus instruments. Each control interface is described here in order. For information on the Fast Data Channel, refer to the *Fast Data Channel* discussion on page 2–50.

VXIbus Interface

The waveform analyzer complies with the VXIbus System Specification in the following ways:

- The waveform analyzer supports VXIbus System Specification revision 1.4.
- The waveform analyzer is a Message-Based Servant, which supports VXIbus configuration and communication registers.
- The waveform analyzer supports Word Serial Protocol and Fast Data Channel data transfers. And it responds to the SCPI and IEEE 488.2 Common Commands listed in the *Syntax and Commands* section of this manual.
- The waveform analyzer is a programmable interrupter for levels 1–7, capable of asserting interrupts and performing interrupt acknowledge sequences.

TTL and ECL Trigger Buses

The waveform analyzer uses the VXIbus TTLTRG and ECLTRG buses to export and import signals to and from other modules in the system. The waveform analyzer supports the SYNC trigger protocol; exported signals are broadcast on a TTLTRG or ECLTRG line and do not require acknowledgement from an acceptor module.

The available waveform analyzer sources for TTLTRG and ECLTRG lines are are as follows:

- ARM — a valid ARM event occurs which enables the TRIG:A circuit
- ATR — a valid trigger event from the TRIGger A subsystem
- BTR —a valid trigger event from the TRIGger B subsystem
- OPC — the signal indicating the active command is complete. It is derived from the Operation Complete bit in the Standard Event Status Register.

www.valuetronics.com

Table 2–7 shows the TTLTRG and ECLTRG lines and their default assignments.

Table 2–7: Trigger output lines and their default assignments

TTLTRg Line Number	Default Assignments
TTLTRG0	ARM signal
TTLTRG1	TRIGger:A event
TTLTRG2	TRIGger:B event
TTLTRG3	OPC
TTLTRG4	ARM signal
TTLTRG5	TRIGger:A event
TTLTRG6	TRIGger:B event
TTLTRG7	OPC
ECLTRG0	TRIGger:B event
ECLTRG1	TRIGger:A event

When importing triggers, the waveform analyzer can enable any one of the TTLTRG or ECLTRG lines to be the trigger input. You can select rising- or falling-edge polarity.

VXIbus Pin Out

The VXIbus connectors are shown in Figure 2–28. Tables 2–8 through 2–10 list the pin assignments for each connector. Connectors are identified (left or right) when viewing the TVS600 Waveform Analyzer from the front panel. Refer to Figure 2–28 for connector locations. For detailed information regarding the signals on the pins, refer to the VMEbus and VXIbus standards referenced above.

www.valuetronics.com

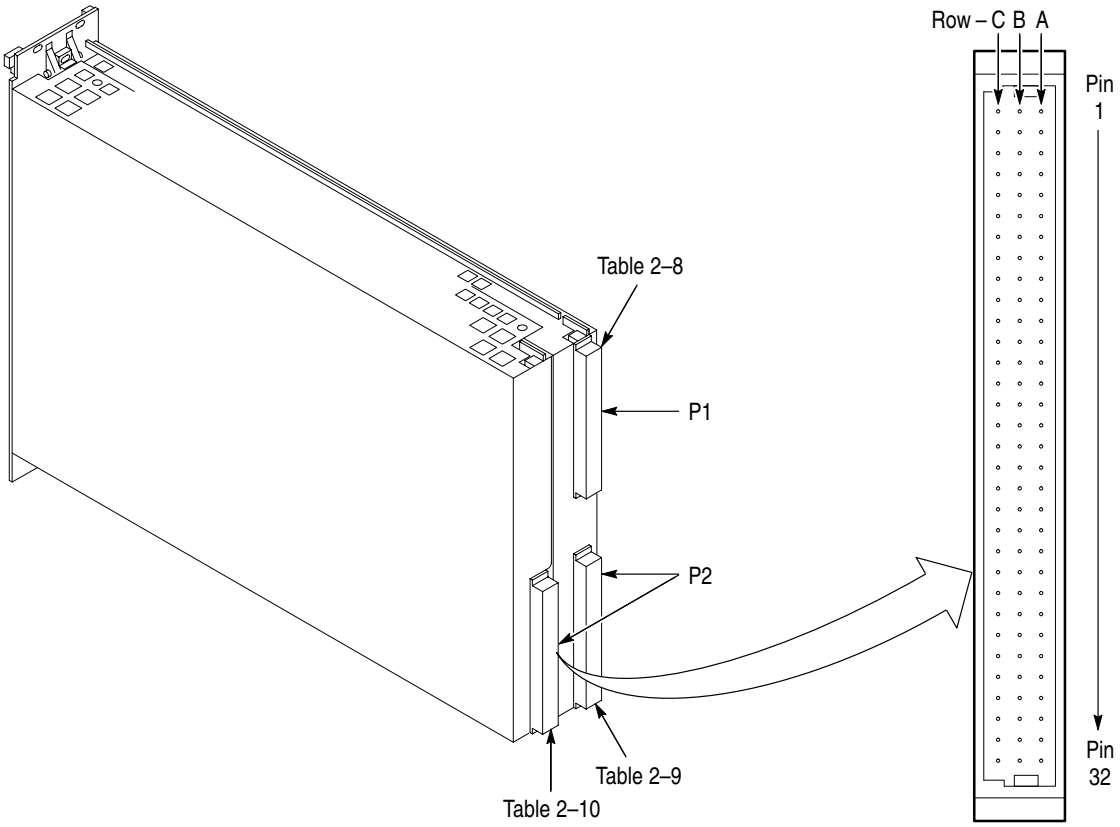


Figure 2–28: VXIbus Connectors P1 and P2

***NOTE**. In the following tables, "NC" means "No Connection."*

www.valuetronics.com

Table 2–8: Left Slot P1 Pinout

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	D00	NC	D08
2	D01	NC	D09
3	D02	ACFAIL	D10
4	D03	BG0IN	D11
5	D04	BG0OUT	D12
6	D05	BG1IN	D13
7	D06	BG1OUT	D14
8	D07	BG2IN	D15
9	GND	BG2OUT	GND
10	NC	BG3IN	SYSFAIL~
11	GND	BG3OUT	BERR~
12	DS1	NC	SYSRESET~
13	DS0	NC	NC
14	WRITE	NC	AM5
15	GND	NC	A23
16	DTACK~	AM0	A22
17	GND	AM1	A21
18	AS	NC	A20
19	GND	AM3	A19
20	IACK	GND	A18
21	IACKIN	NC	A17
22	IACKOUT~	NC	A16
23	AM4	GND	A15
24	A07	IRQ7	A14
25	A06	IRQ6	A13
26	A05	IRQ5~	A12
27	A04	IRQ4~	A11
28	A03	IRQ3~	A10
29	A02	IRQ2~	A09
30	A01	IRQ1~	A08
31	–12 V	+5 VSTDBY	+12 V
32	+5 V	+5 V	+5 V

www.valuetronics.com

Table 2–9: Left Slot P2 Pinout

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	NC	+5 V	NC
2	–2 V	GND	NC
3	NC	NC	GND
4	GND	NC	–5.2 V
5	NC	NC	NC
6	NC	NC	NC
7	–5.2 V	NC	GND
8	NC	NC	NC
9	NC	NC	NC
10	GND	NC	GND
11	NC	NC	NC
12	NC	GND	NC
13	–5.2 V	+5 V	–2 V
14	NC	NC	NC
15	NC	NC	NC
16	GND	NC	GND
17	NC	NC	NC
18	NC	NC	NC
19	–5.2 V	NC	–5.2 V
20	NC	NC	NC
21	NC	NC	NC
22	GND	GND	GND
23	NC	NC	NC
24	NC	NC	NC
25	+5 V	NC	GND
26	NC	NC	NC
27	NC	NC	NC
28	GND	NC	GND
29	NC	NC	NC
30	VMOD1D	NC	GND
31	GND	GND	NC
32	NC	+5 V	NC

www.valuetronics.com

Table 2–10: Right Slot P2 Pinout

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	ECLTRG0	+5 V	CLK10+
2	NC	GND	CLK10–
3	ECLTRG1	NC	GND
4	GND	NC	NC
5	NC	NC	NC
6	NC	NC	NC
7	NC	NC	GND
8	NC	NC	NC
9	NC	NC	NC
10	GND	NC	GND
11	NC	NC	NC
12	NC	GND	NC
13	NC	+5 V	NC
14	NC	NC	NC
15	NC	NC	NC
16	GND	NC	GND
17	NC	NC	NC
18	NC	NC	NC
19	NC	NC	NC
20	NC	NC	NC
21	NC	NC	NC
22	GND	GND	GND
23	TTLTRG0~	NC	TTLTRG1~
24	TTLTRG2~	NC	TTLTRG3~
25	+5 V	NC	GND
26	TTLTRG4~	NC	TTLTRG5~
27	TTLTRG6~	NC	TTLTRG7~
28	GND	NC	GND
29	NC	NC	NC
30	NC	NC	GND
31	GND	GND	NC
32	NC	+5 V	NC

www.valuetronics.com

RS-232C Port

The SERIAL INTERFACE located on the front panel is an RS-232C port. You can use the RS-232C port to control the waveform analyzer. All commands and queries are accepted over this serial interface. The command interpreter responds to the interface that issues a query whether it is the VXIbus or the SERIAL INTERFACE. The RS-232C interface uses a standard 9-pin D type connector. Figure 2–29 shows the pin numbers for the connector and the signals assigned to each pin.

Serial Interface Commands

The serial interface commands are in the SYSTem:COMMunicate:SERial subsystem.

Three sets of settings, available with the SYSTem:COMMunicate:SERial:PRESet command, support standard uses such as a display terminal or computer connection.

- ALL sets baud to 9600, RTS control ON, and echo, parity and pace (XON/XOFF) handshaking off.
- RAW sets echo and pace (XON/XOFF) handshaking off. This mode is appropriate for a computer interface.
- TERMinal sets echo on and automatically sends all status messages to the serial interface. This mode is appropriate for a basic display terminal.

For more information, refer to the serial interface command definitions starting on page 3–179.

www.valuetronics.com

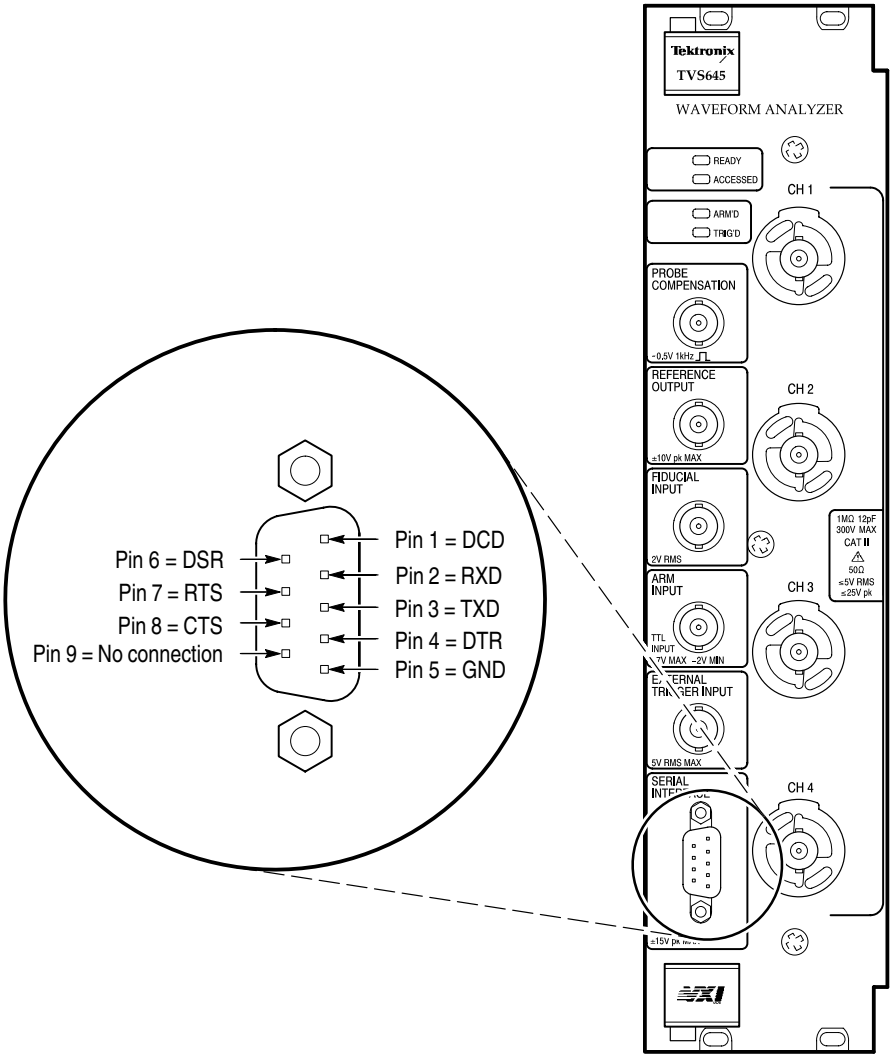


Figure 2–29: Pin assignments for the SERIAL INTERFACE (RS-232) connector

www.valuetronics.com

Tutorial

The examples presented here show you how to use the command set to perform many useful tasks. Each example builds on what you learned in the previous examples to enhance your understanding of the key features of your waveform analyzer. Each example describes briefly what you will accomplish in the example then provides step-by-step instructions to complete the task.

Required Equipment

You will need the following equipment to run these examples:

- A signal generator capable of supplying a 10 MHz square wave.
- Two probes or BNC cables to connect the TVS600 input channels to the signal generator output.
- Signal splitter for BNC or probes to supply the square wave signal to two input channels. If using probes, you may need an adapter to properly connect the probes to the generator output. Refer to the manual for your probes to choose the correct adapter, if any are needed.

Host System Requirements

Before starting this Tutorial, make certain that your VXIbus system is installed and you have a talker/listener program running. Install your waveform analyzer and verify communication between it and your VXIbus controller before starting this tutorial. For instructions on how to install your waveform analyzer, refer to page 1–5.

You will need a way to send commands to the waveform analyzer. You can use the talker/listener program that accompanies the communications card in your controller. Refer to documentation that accompanies the card for instructions on starting the talker/listener application. Alternately, you can use the command feature of the TVS600 Soft Front Panel software shipped with your system. It is installed as part of the *Installation* mentioned earlier. If you wish, you can enter commands with a terminal connected to the RS-232 connector on the waveform analyzer front panel. For RS232 set up information, refer to page 2–69.

***NOTE**. All examples use the Word Serial protocol for communications.*

www.valuetronics.com

Example 1 — Instrument Setup

This example describes the waveform analyzer settings you will typically make before starting to acquire signals. Correct initial set up is very important to ensure that you acquire good data. In this example, you will use the commands INPut, VOLTage, SWEep, and TRIGger. You will actually acquire a signal in Example 2 based on the settings you perform in this example.

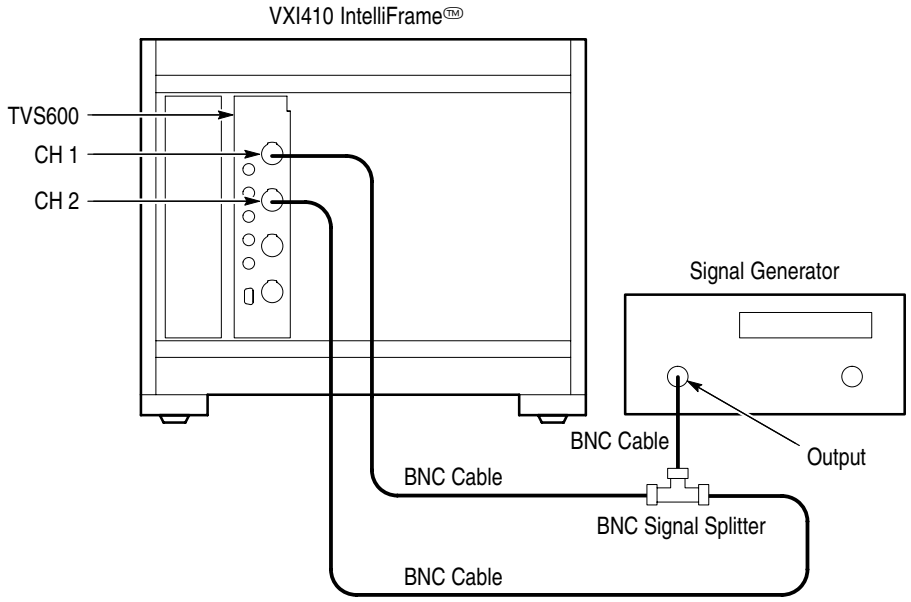


Figure 2–30: Initial Equipment Setup for the Tutorial

1. Connect Channel 1 waveform analyzer input to the signal generator output using a probe or BNC cable and the signal splitter.
2. Configure the signal generator as follows:
   - Output level 4 Vp-p
   - Waveform type Sine Wave
   - Frequency 10 MHz
   - Offset 0 V
   - Enable output
3. Send the command `*RST` to reset the waveform analyzer.

   The *RST command resets instrument settings to their default values. Resetting the instrument is often the quickest way to set the instrument to a

www.valuetronics.com

known state before starting to set up a new measurement. Communication port settings and data security commands are not affected by *RST.

4. Set signal coupling for CH 1 to DC by sending the command `INPut1:COUPling DC`.

   The INPut1:COUPling command sets the coupling to DC, which passes all signal components. Other commands in the INPUT subsystem control input impedance (50 Ω or 1 MΩ) and a low-pass filter (20 MHz or 250 MHz).

   You can specify the identical command using the minimal spelling INP:COUP DC. To ensure portability of your test programs, use the short form of the commands. The short form is used through the remainder of this tutorial.

5. Set the input voltage range for the CH 1 input by sending the command `VOLT1:RANG:PTP 5`.

   The VOLT1:RANG:PTP command sets the full-scale range for the selected channel to 5 V. The voltage range setting should be just larger than the signal you wish to acquire to ensure the best accuracy and resolution of acquired data. Places where the input signal exceeds the range setting are recorded as overrange or underrange values in the waveform record.

6. Now set the time base by sending the command `SWE:TINT 2E-9`.

   With this SWE:TINT command you set the time between samples, the sample interval, to 2 ns. With the default record length of 1024 points, you will acquire just over 20 cycles of the 10 MHz input signal. To derive this value, multiply the record length of 1024 intervals/record by 2 ns/interval to get 2048 ns record duration.

7. Set the trigger level to 1 V by sending the command `TRIG:LEV 1`.

   The TRIGger:LEVel setting applies to the source INTernal1, which is the CH 1 input. TRIGger:SOURce was set to INTernal1 by the *RST command sent in Example 1. You could set the TRIGger:SOURce to the CH 2 input by specifying INT2. Note that when you do not specify :A or :B in TRIGger commands, TRIGger:A is set.

   will

8. Enable the input CH 1 by sending the command `FUNC "XTIM:VOLT 1"`.

   The command FUNC “XTIM:VOLT 1” enables, or connects, the CH 1 signal from the INPut subsystem to the acquisition hardware, which is controlled by the SENSe subsystem. Note that the double quotes are required to send the text string.

The commands you have used in this example are necessary for most acquisitions. In other cases, you might adjust for a DC offset with the command

www.valuetronics.com

VOLT<n>:RANG:OFFS or place the trigger point in the middle of the acquisition record with the command SWE:OREF:LOC 0.5. The <n> in the VOLT<n> command is replaced by a channel number, such as VOLT2 or VOLT1.

You have completed Example 1 and should proceed to Example 2 where you will acquire a waveform.

Example 2 — Acquiring a Signal

This example shows you how to start acquiring a signal based on the setup described in Example 1. You will initiate acquisition and retrieve the acquisition record. The example ends with a discussion of the steps necessary to acquire signals on multiple channels.

The hardware setup is the same as that used in Example 1.

1. Perform the initial set up described in Example 1.
2. Start acquisition by sending the command `INITiate`.

   The INITiate command starts the acquisition system. After initiation, the system looks for any defined ARM condition then looks for a Trigger A event.

   You defined the Trigger A event to be a positive transition through the 1 V level. If you had specified an ARM source, such as the front panel ARM INPUT, then the acquisition system would wait for the ARM signal before looking for a trigger event.
3. Get the acquisition record just acquired from CH 1 by sending the command `DATA? "XTIM:VOLT 1"`.

   Due to the *RST sent earlier, the acquisition record is returned as a series of 1024 ASCII numbers separated by commas. You can change the data format to binary with the FORMat command.
4. To acquire a second channel, simply set the CH 2 vertical range with `VOLT2:RANG:PTP 5` and enable the channel with the command `FUNC "XTIM:VOLT 2"`. Then send the INIT command to start acquisition.

   The settings for the time base (SWEep) and trigger (TRIGger A) are shared by all input channels.

The sequence of commands you used in this example is necessary to insure that signal acquisition starts correctly. Note that the DATA? command waits for completion of acquisition or calculations before it starts to transfer the resultant data record or measurement value.

You have completed Example 2 and should proceed to Example 3 where you will acquire an averaged signal and an enveloped signal in real time.

www.valuetronics.com

Example 3 — Averaging and Enveloping a Signal

This example shows you how to acquire an averaged signal and an enveloped signal. Averaging removes noise from your signal which improves the dynamic range of the acquired waveform and the accuracy of any measurements you perform on it. Enveloping creates a waveform that contains the maximum and minimum values at each sample point over a number of acquisitions.

1. Reset the waveform analyzer with the *RST command and then perform the initial setup in Example 1.
2. Enable signal averaging by sending the command `AVER ON`.

   Averaging acquires a specified number of waveforms and averages them together to produce an averaged waveform.
3. Set the number of acquisitions to average by sending the command `AVER:COUN 16`.
4. To acquire an averaged waveform send the initiate command `INIT`.
5. To transfer the averaged waveform out of the instrument, use the same DATA command you used in Example 2, `DATA? "XTIM:VOLT 1"`.
6. To configure the instrument to acquire an envelope waveform send the command `AVER:TYPE ENV`.
7. Start envelope acquisition with the command INIT.
8. Now get the envelope waveform with the command `DATA? "XTIM:VOLT 1"`.

   Envelope is a special type of averaging acquisition. The default type of averaging is SCALar. Enveloping saves the maximum and minimum values acquired for each sample point during the 16 acquisitions (AVER:COUN 16) you specified in step 3. The result is an acquisition record containing 512 pairs of interleaved maximum and minimum points. The waveform record still contains 1024 points and covers the same time period as a normal acquisition record.

You have completed Example 3 and should proceed to Example 4. In Example 4 you will use the CALCulate system to perform a rise time measurement and integrate a signal in real time.

www.valuetronics.com

Example 4 — Performing Basic Calculations

This example shows you how to perform two types of SCPI calculations. First, you will configure the CALCulate system to measure the rise time on the CH 1 signal and retrieve the result. Then you will integrate a signal and retrieve the waveform record of the integrated signal.

1. Perform the steps in Example 1 in preparation to acquire CH 1.
2. Specify the CH 1 signal as the input to the CALCulate1 block by sending the command `CALC1:FEED "XTIM:VOLT 1"`.

   The FEED command specifies the source that the CALCulate1 block will operate on. The source here is the CH 1 signal vertically scaled by the VOLTage1 block. Hence, the "source" or FEED is the VOLTage 1 block. The waveform analyzer instruments have four CALC blocks that operate on any of the input channels. For example, we could perform the rise time measurement by specifying CALC2:FEED "XTIMe:VOLTage 1". You may find it easier to keep track of operations when you match CALC blocks to the input channel numbers.
3. Set up the CALC1 block to perform the rise time measurement by sending the command `CALC1:WML RTIM`.

   RTIM specifies the rise time measurement. You can specify several measurements at once by adding commands after RTIM separated by commas. For instance, the command CALC1:WML RTIM,PTP,MEAN specifies rise time, peak-to-peak and mean measurements. The parameters proximal, mesial, and distal used by the measurement system, are controlled by CALCulate:WMParameter commands. The defaults are used here.
4. Enable the measurement list for CALC1 block by sending the command `CALC1:WML:STAT ON`.

   This command activates the CALC1 block so it will perform its list of measurements, RTIM in this example. You must set the STATe of a CALC block to ON before the CALC block will perform any measurements.
5. Initiate the acquisition process as you did in the previous Examples.

   Active measurements, RTIM in this instance, are performed when the acquisition process ends. If you are averaging a waveform, the measurements are performed when all acquisitions have been averaged.
6. Retrieve the rise time result by sending the command `CALC1:DATA?`.

   CALC1:DATA? returns all results from the CALC1 block. In this case, the rise time value is returned as a floating point number in ASCII format. When you send CALC1:DATA? all results from calculations and measurements defined for the CALC1 block are returned as a comma separated list.

www.valuetronics.com

7. Set the CALC2 block to operate on the CH 1 signal by sending the command `CALC2:FEED "XTIM:VOLT 1"`.

   Note that the CALC1 block is still configured to perform the rise time measurement (RTIM).

8. Set the CALC2 block to integrate the CH 1 signal by sending the command `CALC:INT:STAT ON`.

9. Initiate the acquisition process as you previously did.

10. Retrieve the integrated waveform result by sending the command `CALC2:DATA?`.

    CALC2:DATA? returns all results from the CALC2 block. In this case the returned result is the integral waveform calculated on the CH 1 input signal. If you send the command CALC1:DATA? you will get the rise time measurement performed by the CALC1 block on this last acquisition. Note that all data returned from a CALC block are composed of ASCII characters separated by commas.

You have completed Example 4 and should proceed to Example 5. In Example 5 you will define a calculation using an algebraic expression.

www.valuetronics.com

Example 5 — Performing Advanced Calculations

This example introduces the expression model for defining calculations. The expression model provides a powerful tool for measurement and signal processing. You will find that setting up a measurement or calculation is easier with the expression model than with the standard SCPI model. Note that the expression model may not be available on other digitizer products with a SCPI command set.

In this example, you will calculate a waveform that is the CH 1 signal minus its mean value.

1. Reset your waveform analyzer and perform the initial set up in Example 1.
2. Specify the CH 1 signal as the input to the CALCulate1 block by sending the command `CALCulate1:FEED "XTIMe:VOLTage 1"`.
3. Set the CALC1 block to calculate the signal minus its mean by sending the command `CALC1:PATH:EXP (CHAN1-MEAN(CHAN1))`.

   This command sets the CALC1 block to first compute the arithmetic mean (or DC level) of the CH 1 acquisition record. Then it subtracts the mean value from each sample value in the record. Incidentally, this calculation cannot be defined using the standard calculation model defined by SCPI.

   Note that, when using the expression model, you do not need to set the LIST, STATe, or FEED for the chosen CALC block.
4. Initiate the acquisition process as you previously did.
5. Retrieve the calculated waveform by sending the command `CALC1:DATA?`.

You have completed Example 5 and should proceed to Example 6. In Example 6 you will save the current settings of the waveform analyzer into an on-board settings location. Then you will recall the setting.

www.valuetronics.com

Example 6 — Saving and Recalling Settings

This example shows you how to save the current waveform analyzer settings in an on-board location and then recall the settings. You will configure CH 1 and CH 2 to acquire signals and perform measurements on those signals.

1. Perform the initial set up described in Example 1 to enable CH 1.
2. Connect the signal generator to CH 2. CH 1 should already be connected. Set the CH 2 vertical range and enabling the channel with the command chain `VOLT2:RANG:PTP 5;FUNC "XTIM 2"`.

   A command chain is a series of commands separated by semicolons (;). You can send commands and queries as command chains.
3. Configure the CALC1 block to measure the rise time on CH 1 by sending the CALC1 source and measurement commands `CALC1:FEED "XTIM:VOLT 1";CALC1:WML RTIM`.
4. Configure the CALC2 block to measure the peak-to-peak value on CH 2 by sending the CALC2 source and measurement commands `CALC2:FEED "XTIM:VOLT 2";CALC2:WML PTP`.
5. Enable the measurement list for both CALC blocks by sending the command `CALC1:WML:STAT ON;CALC2:WML:STAT ON`.
6. Initiate the acquisition process as you did earlier.
7. Retrieve the two measurement results by sending the command `CALC1:DATA?;CALC2:DATA?`.

   The data returned by these chained queries are separated by a semicolon. You do not need to test your settings before saving them, though it is always good to verify your setup before making a measurement.
8. Save the current waveform analyzer instrument settings in location 1 by sending the command `*SAV 1`.

   All acquisition and calculation settings are saved in the settings location number 1. You can reuse a complex test setup by saving your final instrument configuration in a settings location. Instrument settings are not saved that control communications interfaces and input protection. You may store ten instrument settings in on-board memory that is retained when power is off.
9. In order to test the restore capability, reset the instrument by sending the command `*RST`.
10. Check that CH 1 signal feed is now disabled by sending the command `FUNC?`.

www.valuetronics.com

With the instrument reset, you should get the reply "", which is an ASCII null string that indicates no channels are enabled. If channels were enabled you would get a string such as “XTIM:VOLT 1”,”XTIM:VOLT 2”, which indicates CH 1 and CH 2 are enabled.

11. Now restore your settings from location 1 by sending the command `*RCL 1`.

12. Check that CH 1 and CH 2 signal feed is now enabled by sending the command `FUNC?`.

13. You can test the retrieved settings by initiating acquisition and retrieving the the new measurement results as you did in step 7.

You have completed Example 6 and should proceed to Example 7. In Example 7 you will work with the waveform analyzer status and event reporting system.

www.valuetronics.com

Example 7 — Using Status and Events

This example shows you how to use the status and event commands to determine the status of your waveform analyzer. Command, execution, and other system errors plus other system events are stored in a memory buffer on-board. You can use the event information to better control the waveform analyzer and enhance overall system performance.

***NOTE**. If you are entering commands over the RS-232 port, send the command SYST:COMM:SER:ERES OFF. This command disables automatic return of events from the error/event queue to the RS-232 port. Automatic echoing of events occurs only on the RS-232 port.*

1. Reset your waveform analyzer and perform the initial set up in Example 1 to enable CH 1.
2. Enable all events in the Standard Event Status Register (SESR) by sending the command *ESE 256. Figure 2–31 shows the SESR. You can enable any or all events in the SESR so they register in the Status Byte Register.
3. Create a command error by sending the incorrect command `CALC1:XXX?`.

   This syntax error for the root node CALC causes a command error. Figure 2–31 shows that when a command error occurs, its bit is set to one in the SESR.

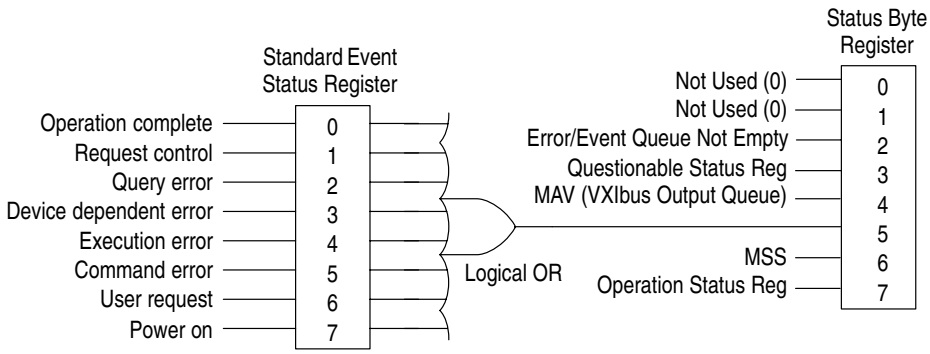


Figure 2–31: Standard Event and Status Byte Registers

4. Check the Status Byte Register (SBR) for the command error by sending the query `*STB?`.

   This query returns the status byte from the SBR. The status byte contains a decimal number that indicates errors or other significant events. Figure 2–31 shows the bit assignments. Your response should have bit five set for a

www.valuetronics.com

Command error. Unlike the other waveform analyzer status registers, the SBR output simply mirrors its inputs so it is not cleared when you read it.

5. To get specific information on the type of error, read the SESR by sending the command `*ESR?`.

   The returned decimal number has bit five set and possibly others. Note that reading the SESR, or any other event register, clears it.

6. To get the actual error message from the error/event queue send the command `SYST:ERR?`.

   The response should be the error `Execution Error -113, "Undefined header"`. Reading an event message from the event/error queue removes it from the queue. By repeating the SYST:ERR? query, or by sending the command SYST:ERR:ALL?, you can read all buffered events.

   Note that you can read the next event from the event/error queue without first checking the Standard Event Status Register or the Status Byte Register.

7. Generate a command error by sending the command `CALC1:WML FTM`

   This command has an incorrect spelling for the parameter FTIM (fall time).

8. Try clearing the events by sending the reset command `*RST`.

9. To check if the error message is still in the error/event queue send the command SYST:ERR?.

   You should get the error `Execution Error -141, "Invalid character data"`. To ensure that only current events and errors are in the event queue you should always clear the errors before you start a new measurement or test.

10. To clear the event registers and all messages from the error/event queue, send the command `*CLS`.

    Only the *CLS command, or shutting off power, can clear the status registers and error/event queue.

11. Check for the command error that you created in step 7 by sending the command `*ESR?`.

    You will get a response of "0" unless you have sent other commands since the last reset.

You have completed Example 7, which is the last of the Tutorial examples. For more information on specific commands refer to the command descriptions in section 3. For more detailed information on the subsystems of the waveform analyzer, refer to the *Functional Overview* in section 2.

www.valuetronics.com

Command Syntax

This section contains information on the Standard Commands for Programmable Instruments (SCPI) and IEEE 488.2 Common Commands you can use to program your waveform analyzer.

SCPI Commands and Queries

SCPI is a standard created by a consortium that provides guidelines for remote programming of instruments. These guidelines provide a consistent programming environment for instrument control and data transfer. This environment uses defined programming messages, instrument responses, and data format across all SCPI instruments, regardless of manufacturer. The waveform analyzer uses a command language based on the SCPI standard.

The SCPI language is based on a hierarchical or tree structure (see Figure 3–1) that represents a subsystem. The top level of the tree is the root node; it is followed by one or more lower-level nodes.

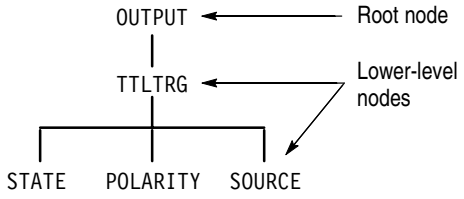


Figure 3–1: Example of SCPI Subsystem Hierarchy Tree

You can create commands and queries from these subsystem hierarchy trees. Commands specify actions for the instrument to perform. Queries return measurement data and information about parameter settings.

Creating Commands

SCPI commands are created by stringing together the nodes of a subsystem hierarchy and separating each node by a colon.

In Figure 3–1, OUTPUT is the root node and TTLTRG, STATE, POLARITY, and SOURCE are lower-level nodes. To create a SCPI command, start with the root node OUTPUT and move down the tree structure adding nodes until you reach the end of a branch. Most commands and some queries have parameters; you must include a value for these parameters. If you specify a parameter value that is out of range, the parameter will be set to a default value. The command descriptions, which start on page 3–27, list the valid values for all parameters.

www.valuetronics.com

For example, OUTPUT:TTLTRG1:STATE ON is a valid SCPI command created from the hierarchy tree in Figure 3–1.

Creating Queries

To create a query, start at the root node of a tree structure, move down to the end of a branch, and add a question mark. OUTPUT:TTLTrg:STATe? is an example of a valid SCPI query using the hierarchy tree in Figure 3–1.

Parameter Types

Every parameter in the command and query descriptions is of a specified type. The parameters are enclosed in brackets, such as <pattern>. The parameter type is listed after the parameter and is enclosed in parentheses, for example, (discrete). Some parameter types are defined specifically for the waveform analyzer command set and some are defined by ANSI/IEEE 488.2-1987 (see Table 3–1).

Table 3–1: Parameter Types Used in Syntax Descriptions

Parameter Type	Description	Example
binary	Binary numbers	#B0110
binary block[1]	A specified length of binary data	#512234xxxxx . . . where 5 indicates that the following 5 digits (12234) specify the length of the data in bits; xxxxx ... indicates the binary data
boolean	Boolean numbers or values	ON or 1 OFF or 0
discrete	A list of specific values	HIGH, LOW, MID, PRBS23
hexadecimal[2]	Hexadecimal numbers (0–9, A, B, C, D, E, F)	#HAA, #H1
NR1[2,3] numeric	Integers	0, 1, 15, –1
NR2[2] numeric	Decimal numbers	1.2, 3.141516, –6.5
NR3[2] numeric	Floating point numbers	3.1415E–9, –16.1E5
NRf[2] numeric	Flexible decimal number that may be type NR1, NR2 or NR3	See NR1, NR2, NR3 examples
string[4]	Alphanumeric characters (must be within quotation marks)	“Testing 1, 2, 3”

1 **Defined in ANSI/IEEE 488.2 as “Definite Length Arbitrary Block Response Data.”**

2 **An ANSI/IEEE 488.2–1992-defined parameter type.**

3 **Some commands and queries will accept a hexadecimal value even though the parameter type is defined as NR1.**

4 **Defined in ANSI/IEEE 488.2 as “String Response Data.”**

www.valuetronics.com

Abbreviating Commands, Queries, and Parameters

You can abbreviate most SCPI commands, queries, and parameters to an accepted short form. This manual shows these short forms as a combination of upper and lower case letters. The upper case letters indicate the accepted short form of a command. As shown in Figure 3–2, you can create a short form by using only the upper case letters. The accepted short form and the long form are equivalent and request the same action of the instrument.

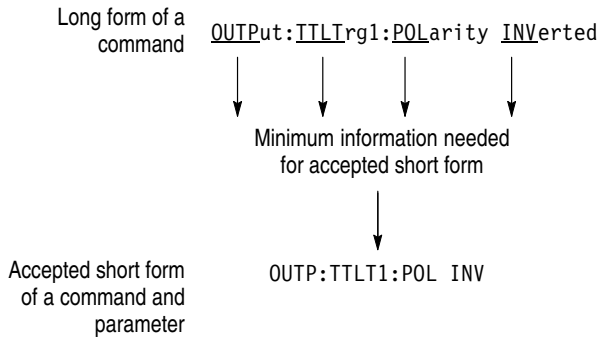


Figure 3–2: Example of Abbreviating a Command

NOTE. *The numeric part of a command or query must always be included in the accepted short form. In Figure 3–2, the "1" of "TTLTRG1" is always included in the command or query.*

Chaining Commands and Queries

You can chain several commands or queries together into a single message. To create a chained message, first create a command or query, add a semicolon (;), and then add more commands or queries and semicolons until you are done. If the command following a semicolon is a root node, precede it with a colon (:). Figure 3–3 illustrates a chained message consisting of several commands and queries. The single chained message should end in a command or query, not a semicolon. Responses to any queries in your message are separated by semicolons.

www.valuetronics.com

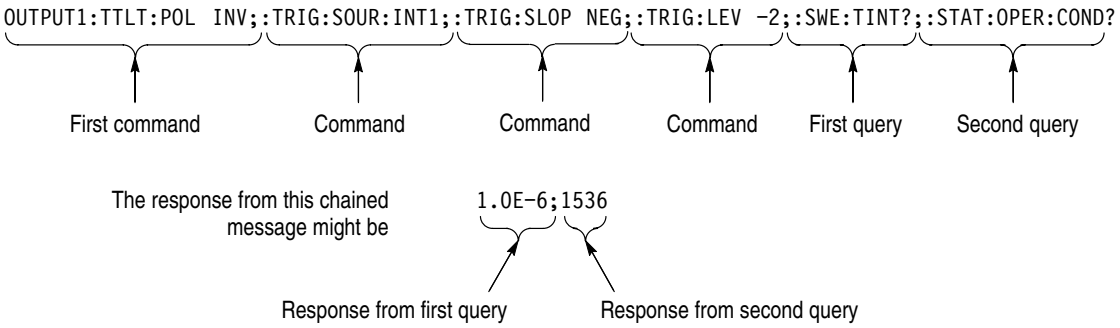


Figure 3–3: Example of Chaining Commands and Queries

If a command or query has the same root and lower-level nodes as the previous command or query, you can omit these nodes. In Figure 3–4, the second command has the same root node (TRIG) as the first command, so these nodes can be omitted.

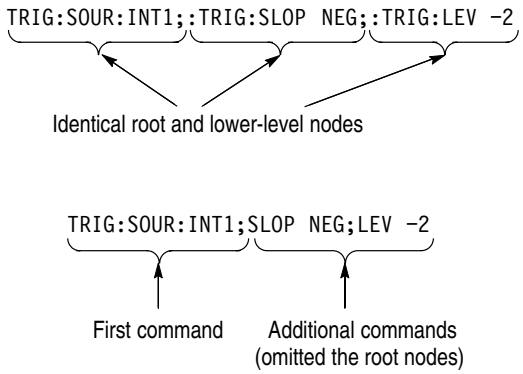


Figure 3–4: Example of Omitting Root and Lower-level Nodes in Chained Message

General Rules

Here are some general rules for using SCPI commands, queries, and parameters:

- You can use single (‘ ’) or double (“ ”) quotation marks for quoted strings, but you cannot use both types of quotation marks for the same string.

  correct: “This string uses quotation marks correctly.”

  correct: ‘This string also uses quotation marks correctly.’

  incorrect: “This string does not use quotation marks correctly.’

www.valuetronics.com

- You can use upper case, lower case, or a mixture of both cases for all commands, queries, and parameters.

  ```
  OUTPUT1:TTLTRG:POLARITY INVERTED
  ```

  is the same as

  ```
  output1:ttltrg:polarity inverted
  ```

  and

  ```
  OUTPUT1:ttltrg:polarity INVERTED
  ```

- No embedded spaces are allowed between or within nodes.

  correct: `OUTPUT1:TTLTRG:POLARITY INVERTED`

  incorrect: `OUTPUT1: TTLTRG: POLARITY INV ERTED`

IEEE 488.2 Common Commands

Description

ANSI/IEEE Standard 488.2 defines the codes, formats, protocols, and usage of common commands and queries used on the interface between the controller and the instruments. The waveform analyzer complies with this standard.

Command and Query Structure

The syntax for an IEEE 488.2 common command is an asterisk (*) followed by a command and, optionally, a space and parameter value. The syntax for an IEEE 488.2 common query is an asterisk (*) followed by a query and a question mark. All of the common commands and queries are listed in the last part of the *Syntax and Commands* section. The following are examples of common commands:

- `*ESE 16`
- `*CLS`

The following are examples of common queries:

- `*ESR?`
- `*IDN?`

www.valuetronics.com

Backus-Naur Form Definition

This manual may describe commands and queries using the Backus-Naur Form (BNF) notation. Table 3–2 defines the standard BNF symbols:

Table 3–2: BNF Symbols and Meanings

Symbol	Meaning
< >	Defined element
::=	Is defined as
|	Exclusive OR
{ }	Group; one element is required
[ ]	Optional; can be omitted
. . .	Previous element(s) may be repeated
( )	Comment

Message Terminators

This manual uses <EOM> (End of message) to represent a message terminator.

Symbol	Meaning
<EOM>	Message terminator

The end-of-message terminator may be the END message (EOI asserted concurrently with the last data byte), the ASCII code for line feed (LF) sent as the last data byte, or both. The waveform analyzer always terminates messages with LF and EOI. It allows white space before the terminator.

Constructed Mnemonics

Some header mnemonics specify one of a range of mnemonics. For example, a channel mnemonic can be either INP1, INP2, INP3, or INP4. You use these mnemonics in the command just as you do any other mnemonic. For example, there is a INP1:FILT command, and there is also an INP2:FILT command. In the command descriptions, this list of choices is abbreviated as INP<n>.

www.valuetronics.com

Block Arguments

Several waveform analyzer commands use a block argument form:

Symbol	Meaning
<NZDig>	A non-zero digit character, in the range 1–9
<Dig>	A digit character, in the range 0–9
<DChar>	A character with the hex equivalent of 00 through FF hexadecimal (0 through 255 decimal)
<Block>	A block of data bytes, defined as: <Block> ::= { #<NZDig><Dig>[<Dig>...][<DChar>...] | #0[<DChar>...]<terminator> }

<NZDig> specifies the number of <Dig> elements that follow. Taken together, the <Dig> elements form a decimal integer that specifies how many <DChar> elements follow.

www.valuetronics.com

Command Groups

This section lists waveform analyzer commands by functional groups. A question symbol surrounded by brackets [?] identifies commands that also have a query form.

Auto-Advance Commands

Commands in the AADVance subsystem control how auto-advance acquisition records are acquired and transferred to a VXIbus controller.

Table 3–3: Auto-advance Commands

Header	Description
AADVance[?]	Sets the state of the auto-advance acquisition mode.
:COUNT	Sets the number of records to acquire in the auto-advance acquisition mode.
:RECord	
:COUNt[?]	Sets the number of auto-advance acquisition records to transfer.
:STARt[?]	Sets the number of the first auto-advance acquisition record to transfer.

Abort Commands

Commands in the ABORt subsystem operate with the ARM and TRIGger subsystems to stop signal acquisition.

Table 3–4: Abort Commands

Header	Description
ABORt	Stops all acquisition and measurements and returns the arm/trigger subsystem to the idle state.

www.valuetronics.com

Arm Commands

Commands in the ARM Subsystem operate with the TRIGger, INITiate, and ABORt subsystems to trigger acquisitions.

Table 3–5: Arm Commands

Header	Description
ARM	
:DEFine?	Returns the predefined SEQuence1 alias.
:SOURce[?]	Sets the source that will arm the acquisition system.

Average Commands

Commands in the AVERage subsystem controls the averaging function.

Table 3–6: Average Commands

Header	Description
AVERage	Sets whether the waveform analyzer performs averaging.
:COUNt[?]	Sets the number of acquisition records to average.
:TYPE[?]	Sets the type of waveform averaging to perform.

Calculate Commands

Commands in the CALCulate subsystems process and perform measurements on acquisition records.

Table 3–7: Calculate Commands

Header	Description
CALCulate<n>	
:AAMList[?]	Sets the list of measurements to perform on auto-advance waveform records.
:STATe[?]	Sets whether to perform waveform measurement(s) on acquisition records captured with Auto Advance acquisition.
:DATA?	Returns the results of waveform processing and measurement functions.

www.valuetronics.com

Table 3–7: Calculate Commands (Cont.)

Header	Description
:PREamble?	Returns the data preamble for acquisition record processing and measurement functions.
:DERivative	
:STATe[?]	Sets whether to calculate a post-acquisition derivative on the selected channel.
:FEED1[?]	Sets the source of data for the specified CALCulate block.
:FEED2[?]	Sets a second source of data for the specified CALCulate block. Used when taking dual-waveform measurements, such as gain or delay.
:CONTEXT[?]	Sets the measurement parameter block (calc_block) used to characterize the feed2 waveform.
:FILTer	
:FREQuency	Sets the type of FREQuency filtering to perform on an acquisition record.
:CENTer[?]	Sets the center frequency of the bandpass or notch filter.
:HPASs[?]	Sets the limit frequency below which the filter attenuates all frequency components at 6 dB.
:LPASs[?]	Sets the limit frequency above which the filter attenuates all frequency components at 6 dB.
:SPAN[?]	Sets or queries the frequency range to be used by the bandpass and notch filters.
:SREJection[?]	Sets the level of rejection or attenuation for frequency components in the defined stop band.
:STARt[?]	Sets the 6 dB start, or lower limit, frequency of the bandpass and notch filters.
:STATe[?]	Sets whether frequency filtering will be performed on acquisition records.
:STOP[?]	Sets the 6 dB stop, or upper limit, frequency of the bandpass and notch filters.
:TWIDth[?]	Sets the slope of roll off for the post-acquisition filter.
:FORMat[?]	Sets whether to process the acquisition record to produce a new format with complex results.
:IMMediate[?]	Sets whether the specified CALCulate block will reprocess SENSe data without reacquiring new data.
:INTegral	
:STATe[?]	Sets whether to process the acquisition record to produce an integral acquisition record.

www.valuetronics.com

Table 3–7: Calculate Commands (Cont.)

Header	Description
:PATH[?]	Sets a list of CALCulate functions to execute in the order listed.
:EXPRession[?]	Sets a measurement expression using the standard syntax of the C programming language.
:SMOothing[?]	Sets whether to perform smoothing on an acquisition record.
:POINts[?]	Sets the number of adjacent points to average in the acquisition record.
:TRANsform	
:FREQuency	
:STATe[?]	Sets or queries whether to perform a Fast Fourier Transform (FFT) on the specified acquisition record.
:WINDow[?]	Sets the type of data windowing (or shaping) to use prior to the FFT transformation.
:WMList[?]	Sets the list of waveform measurements to perform.
:STATe[?]	Sets whether the waveform measurement list for the specified CALC block will execute after the next acquisition.
:WMParameter	
:EDGE[?]	Specifies by number which edge in the waveform record to use for taking cross, ncross, and pcross measurements.
:HIGH[?]	Sets the high (most positive) level used for time and amplitude measurements.
:HMEThod[?]	Sets the method for calculating the high (most positive) value for time and amplitude waveform measurements.
:LOW[?]	Sets the low (most negative) level used for time and amplitude measurements.
:LMEThod[?]	Sets the method for calculating the low (most negative) value for time and amplitude waveform measurements.
:HFREFerence[?]	Sets the high reference (distal) level in vertical units for time and amplitude measurements.
:RELative[?]	Sets the high reference (distal) level used for time and amplitude measurements.
:LREFerence[?]	Sets the low reference (proximal) level in vertical units for time and amplitude measurements.
:RELative[?]	Sets the low reference (proximal) level used for time and amplitude measurements.
:MREFerence[?]	Sets the middle reference (mesial) level in vertical units for time and amplitude measurements.

www.valuetronics.com

Table 3–7: Calculate Commands (Cont.)

Header	Description
:HYSTeresis[?]	Sets the middle reference (mesial) hysteresis used for time and amplitude measurements.
:RELative[?]	Sets the middle reference (mesial) level used for time and amplitude measurements.
:RMEThod[?]	Sets the method for calculating the reference (high, middle, low) values for time and amplitude measurements.
:SLOPe[?]	Sets the direction, positive or negative, for the waveform edges used in delay measurements.

Calibration Commands

Commands in the CALibration subsystem run the waveform analyzer self-calibration functions.

Table 3–8: Calibration Commands

Header	Description
CALibration[?]	Executes all self-calibration functions.
:RESults?	Returns the results code for the last calibration performed.
:VERBose?	Returns an ASCII string describing the results of the last calibration performed.

Data Commands

Commands in the DATA subsystem provide a means of accessing the data produced by the sense functions.

Table 3–9: Data Commands

Header	Description
DATA?	Returns the results for the specified function or for all sense functions that are enabled.
:PREamble?	Returns the data preamble for the specified function or for all enabled sense functions.

www.valuetronics.com

Format Commands

Commands in the FORMat subsystem set the format of acquisition record data and measurement data transferred out of the waveform analyzer.

Table 3–10: Format Commands

Header	Description
FORMat[?]	Sets the type of encoding used to transfer acquisition records acquired with the SENSe subsystem.
:CALCulate[n][?]	Sets the type of data format used to transfer CALCulate data.
:TRACe:<tracename>[?]	Sets the type of data format used to transfer TRACe data.
:BORDer[?]	Sets the byte order used to transfer binary data.

Function Commands

Commands in the FUNCtion subsystem control sense functions.

Table 3–11: Function Commands

Header	Description
FUNCtion	Sets which sense functions are enabled.
:ALL	Enables all sense functions.
:COUNt?	Returns the number of enabled sense functions.
:OFF[?]	Sets which sense functions are disabled.
:ALL	Disables all sense functions at once without the side effects of sending the *RST command.
:COUNt?	Returns the number of disabled sense functions.
:CONCurrent?	Sets whether more than one sense function can be enabled at a time.
:STATe[?]	Sets the state of a specified sense function.

www.valuetronics.com

Initiate Commands

Commands in the INITiate subsystem operate with the ARM and TRIGger subsystems to start signal acquisition.

Table 3–12: Initiate Commands

Header	Description
INITiate	Starts waveform analyzer acquisitions and measurements.
:CONTinuous[?]	Sets or queries whether the acquisition loop repeats continuously.
:COUNt[?]	Sets or queries the number of times to repeat the arm/trigger acquisition loop.

Input Commands

Commands in the INPut subsystem control input parameters that include coupling, filtering, impedance, and protection.

Table 3–13: Input Commands

Header	Description
INPut<n>	
:COUPling[?]	Sets the type of signal coupling for the specified input channel.
:FILTer[?]	Sets whether the low pass filter is on or off.
:FREQuency[?]	Sets the frequency limit of the low pass filter.
:IMPedance[?]	Sets the input impedance.
:PROTection	
:STATe[?]	Sets the state of the input protection circuitry for all input channels.

www.valuetronics.com

Memory Commands

Commands in the MEMory subsystem store and retrieve instrument settings.

Table 3–14: Memory Commands

Header	Description
MEMory	
:DATA[?]	Sets the instrument settings (or state) for the ten on-board nonvolatile memory locations.
:NSTates?	Returns the number of instrument settings (states) that can be stored in the waveform analyzer.
:STATe	
:CATalog?	Returns the list of predefined names for the on-board stored settings.
:DEFine?	Returns the register number of a specified instrument settings location in the waveform analyzer.

Output Commands

Commands in the OUTPut subsystem route signals to the VXIbus trigger lines and enable the probe compensation and reference signals.

Table 3–15: Output Commands

Header	Description
OUTPut	
:ECLTrg<n>[?]	Sets whether the instrument should drive (source) the specified VXIbus ECL trigger line when a trigger event occurs.
:POLarity[?]	Sets or queries the drive polarity for each VXIbus ECL trigger line.
:SOURce[?]	Sets or queries the drive source for each VXIbus ECL trigger line.
:PCOMpensate[:STATE][?]	Sets whether the probe compensation signal is output on the connector PROBE COMPENSATION.
:FUNCtion[?]	Sets or queries which compensate signal is output on the connector PROBE COMPENSATION.
:REFerence[:STATE][?]	Sets whether a reference signal is output on the front-panel connector REFERENCE OUTPUT.

www.valuetronics.com

Table 3–15: Output Commands (Cont.)

Header	Description
:FUNCtion[?]	Sets or queries which reference signal is output on the connector REFERENCE OUTPUT.
:TTLTrg<n>[?]	Sets whether the instrument should drive (source) the specified VXIbus TTL trigger lines.
:POLarity[?]	Sets or queries the drive polarity for each VXIbus TTL trigger line.
:SOURce?	Sets or queries the drive source for each VXIbus TTL trigger line.

Roscillator Commands

Commands in the ROSCillator subsystem control the source of the reference oscillator (clock) for the :SWEep subsystem.

Table 3–16: Reference Oscillator Commands

Header	Description
ROSCillator	
:SOURce[?]	Sets the source of the 10 MHz clock reference for the acquisition system.

Sense Commands

For a listing of the SENSe commands refer to their root level names in this section:

- AADVance, page 3–9
- AVERage, page 3–10
- DATA, page 3–13
- FUNCtion, page 3–14
- ROSCillator, page 3–17
- SWEep, page 3–19
- VOLTage, page 3–19

www.valuetronics.com

Status Commands

Commands in the STATus subsystem, along with several IEEE 488.2 Common Commands, control the status and event reporting system.

Table 3–17: Status Commands

Header	Description
STATus	
:OPERation?	Returns the contents of the Operation Status Register as a decimal number.
:CONDition?	Returns the contents of the Operation Status Condition Register (OSCR).
:ENABle[?]	Sets the contents of the Operation Status Enable Register (OSER).
:NTRansition[?]	Sets the contents of the Operation Negative Transition Register (ONTR).
:PTRansition[?]	Sets the contents of the Operation Positive Transition Register (OPTR).
:QENable[?]	
:NTRansition[?]	Sets the contents of the Negative Transition Queue Enable Register (NTQER) for the Operation Status Register.
:PTRansition[?]	Sets the contents of the Positive Transition Queue Enable Register (PTQER) for the Operation Status Register.
:PRESet	Presets the SCPI Enable and Transition registers, and the Status Queue enable registers.
:QUEStionable?	Returns the contents of the Questionable Status Register as a decimal number.
:CONDition?	Returns the contents of the Questionable Status Condition Register (QSCR).
:ENABle[?]	Sets the contents of the Questionable Status Enable Register (QSER).
:NTRansition[?]	Sets the contents of the Questionable Negative Transition Register (QNTR).
:PTRansition[?]	Sets the contents of the Questionable Positive Transition Register (QPTR).
:QENable	
:NTRansition[?]	Sets the contents of the Negative Transition Queue Enable Register (NTQER) for the Questionable Status Register.

www.valuetronics.com

Table 3–17: Status Commands (Cont.)

Header	Description
:PTRansition[?]	Sets the contents of the Positive Transition Queue Enable Register (PTQER) for the Questionable Status Register.
:SESR	
:QENable	Sets the contents of the Event Status Enable Register (ESER).

Sweep Commands

Commands in the SWEep subsystem control the acquisition timebase for all VOLTage[n] acquisitions.

Table 3–18: Sweep Commands

Header	Description
SWEep	
:OFFSet	
:POINts[?]	Sets the position in points of the acquisition record relative to the trigger point.
:TIME[?]	Sets the position in time (seconds) of the acquisition record relative to the trigger point.
:OREFerence	
:LOCation[?]	Sets the location of the reference point in an acquisition record.
:POINts[?]	Sets the number of data points in an acquisition record.
:TIME[?]	Sets the time span or duration of the acquisition record.
:TINTerval[?]	Sets the time interval between acquired data points.

www.valuetronics.com

System Commands

Commands in the SYSTem subsystem program utility functions and return version information about the waveform analyzer.

Table 3–19: System Commands

Header	Description
SYSTem	
:COMMunicate	
:SERial	
:BAUD?	Sets the baud rate of the front panel RS-232 port.
:CONTrol	
:DCD[?]	Sets whether the instrument is sensitive to the DCD line.
:RTS[?]	Sets the operation of the RTS and CTS lines.
:ECHO[?]	Sets whether incoming characters are echoed back.
:ERESponse[?]	Sets whether error messages are automatically returned.
:LBUFfer[?]	Sets the state of the character buffer.
:PACE[?]	Sets whether software flow control (XON/XOFF) is enabled.
:PARity[?]	Sets the type of parity for the front panel RS-232 port.
:PRESet	
[:ALL]	Configures RS-232 port parameters to default values.
:RAW	Configures the RS-232 port parameters for use with a computer.
:TERMinal	Configures the RS-232 port parameters for use with a terminal.
:SBITs[?]	Sets the number of stop bits sent with each character.
:VERBose[?]	
:ERRor?	Returns the next entry from the waveform analyzer Status Queue.
:ALL?	Returns the list of all events stored in the waveform analyzer Status Queue.
:CODE?	Returns the next event code stored in the waveform analyzer Status Queue.
:ALL?	Returns the list of all event codes stored in the waveform analyzer Status Queue.
:COUNt?	Returns the number of unread events in the Status Queue.
:PROTect[?]	Sets whether protection for a group of sensitive instrument commands is enabled.

www.valuetronics.com

Table 3–19: System Commands (Cont.)

Header	Description
:SECurity	
:IMMediate	Immediately destroys all measurement data and stored instrument settings.
:SET[?]	Sets the internal state of the instrument as a binary data block.
:VERSion?	Returns the SCPI version supported by the waveform analyzer.

Test Commands

Commands in the TEST subsystem execute the internal self-tests of the waveform analyzer module.

Table 3–20: Test Commands

Header	Description
TEST	Executes all internal self-tests once.
[:ALL][?]	Executes all internal self-tests once. The query returns the test results.
:RESults?	Returns the failure code for the last self-test command that was executed.
:VERBose?	Returns a failure code as a string describing the last executed self-test command and the test results.

Trace Commands

Commands in the TRACe subsystem store and retrieve acquisition and measurement results.

Table 3–21: Trace Commands

Header	Description
TRACe?	Transfers acquisition records or measurement results to your VXIbus controller.
:PREamble?	Transfers the data preamble for acquisition records or measurement results to your VXIbus controller.

www.valuetronics.com

Table 3–21: Trace Commands (Cont.)

Header	Description
:CATalog?	Returns list of the predefined trace names in your waveform analyzer.
:COPY	Copies acquisition or measurement data to the outgoing Fast Data Channel (FDC).
:FEED?	Returns the source of data for pre-defined trace names.
:LIST[?]	Sets or queries the list of traces to transfer.
:POINts?	Returns the number of sample points in the acquisition record or CALCulate block record.

Trigger Commands

Commands in the TRIGger subsystem operate with the ARM, INITiate and ABORt subsystems to trigger acquisitions.

Table 3–22: Trigger Commands

Header	Description
TRIGger	
:ATRigger[:STATe][?]	Sets whether to generate an automatic trigger.
:COUPling[?]	Sets the source of the A trigger to AC or DC coupled.
:<preset>	Sets trigger coupling, filtering, and hysteresis.
:DELay[?]	Sets the trigger delay for the trigger A circuit.
:FILTer	Sets the state of the 50 kHz low pass trigger filter.
:HPASs[:STATe][?]	Sets the state of the 50 kHz high pass trigger filter.
:HOLDoff	
:TIME[?]	Sets or queries the trigger holdoff time.
:HYSTeresis	
:SELect[?]	Sets how far the trigger A signal must fall below or rise above TRIGger:LEVel to detect an edge.
:LEVel[?]	Sets the trigger level for the TRIGger subsystem.
:SLOPe[?]	Sets whether triggering occurs on the positive-going or negative-going edge of the trigger source.
:SOURce[?]	Sets the source of the trigger signal for the trigger A circuit.

www.valuetronics.com

Table 3–22: Trigger Commands (Cont.)

Header	Description
:TYPE[?]	Sets the type of triggering to use for the next acquisition.
:DEFine?	Returns the predefined SEQuence1 alias, A.
:B	
:COUPling[?]	Sets the source of the B trigger to AC or DC coupled.
:<preset>	Sets B trigger coupling, filtering, and hysteresis.
:DELay[?]	Sets the trigger delay for the trigger B circuit.
:ECOunt[?]	Sets the number of B trigger events to count before starting acquisition.
:FILTer	Sets the state of the 50 kHz, low-pass filter for the B trigger circuit.
:HPASs[:STATe][?]	Sets the state of the 50 kHz high pass filter for the B trigger circuit.
:HYSTeresis[?]	
:SELect[?]	Sets how far the trigger B signal must fall below or rise above TRIGger:B:LEVel to detect an edge.
:LEVel[?]	Sets the trigger level for the TRIGger:B subsystem.
:SLOPe[?]	Sets whether triggering occurs on the positive-going or negative-going edge of the trigger B source.
:SOURce[?]	Sets the source of the trigger signal for the trigger B circuit.
:SEQuence2	
:DEFine?	Returns the predefined SEQuence2 alias, B.
:PULSe	
:CLASs[?]	Sets the class of pulse triggering to use for the next acquisition.
:GLITch	
:POLarity	Sets the polarity of the event pulse for pulse glitch triggering.
:QUALify	Sets the type of time qualification for pulse glitch triggering.
:WIDth	Sets the width of the pulse used for pulse glitch triggering.
:SOURce	Sets the source of the trigger signal used for all pulse triggering.
:THReshold	Sets the voltage threshold used for pulse triggering.

www.valuetronics.com

Table 3–22: Trigger Commands (Cont.)

Header	Description
:WIDth	
:HLIMit	Sets the high or longest valid pulse width to qualify for pulse triggering.
:LLIMit	Sets the lower or minimum valid pulse width to qualify for pulse triggering.
:POLarity	Sets the polarity of the pulse used for pulse width triggering.
:QUALify	Sets the type of time qualification for pulse width triggering.

Voltage Commands

Commands in the VOLTage subsystem control parameters that relate to the input voltage range of the waveform analyzer.

Table 3–23: Voltage Commands

Header	Description
VOLTage<n>	
:RANGe	
[:UPPer][?]	Sets the most positive end of the amplifier voltage range.
:LOWer[?]	Sets the most negative end of the amplifier voltage range.
:OFFSet[?]	Sets the voltage offset.
:PTPeak[?]	Sets the peak-to-peak (full-scale) voltage range.

IEEE 488.2 Commands

The waveform analyzer supports the following IEEE 488.2 common commands.

www.valuetronics.com

Table 3–24: IEEE 488.2 Common Commands

Header	Description
*CAL?	Initiates internal calibration and returns a failure code.
*CLS	Clears the SCPI and IEEE 488.2 event registers and the Status Queue.
*ESE[?]	Sets the Service Request Enable Register (SRER).
*ESR?	Returns the contents of the Standard Event Status Register.
*IDN?	Returns the waveform analyzer identification message.
*LRN?	Returns the current state of the waveform analyzer as a sequence of ASCII settings.
*OPC[?]	Synchronizes command execution with the controller.
*OPT?	Returns the options installed in the instrument.
*PUD[?]	Sets the protected user data stored in the waveform analyzer.
*RCL	Recalls the specified instrument setting from non-volatile memory.
*RST	Resets instrument settings to a default state.
*SAV	Saves the current instrument settings in the non-volatile memory.
*SRE[?]	Sets the Service Request Enable Register (SRER).
*STB?	Returns the contents of the Status Byte Register.
*TST?	Initiates an internal self-test and returns a failure code.
*WAI?	Synchronizes command execution with the system controller.

www.valuetronics.com

Commands

This section describes each command and query in the waveform analyzer. The commands are organized by subsystem groups and the commands in each group are in alphabetical order. In Figure 3–5, each block is a root node and the commands within a block are subsystems. For example, SENSe is a root node and AVERage is a subsystem of the SENSe node.

Overview

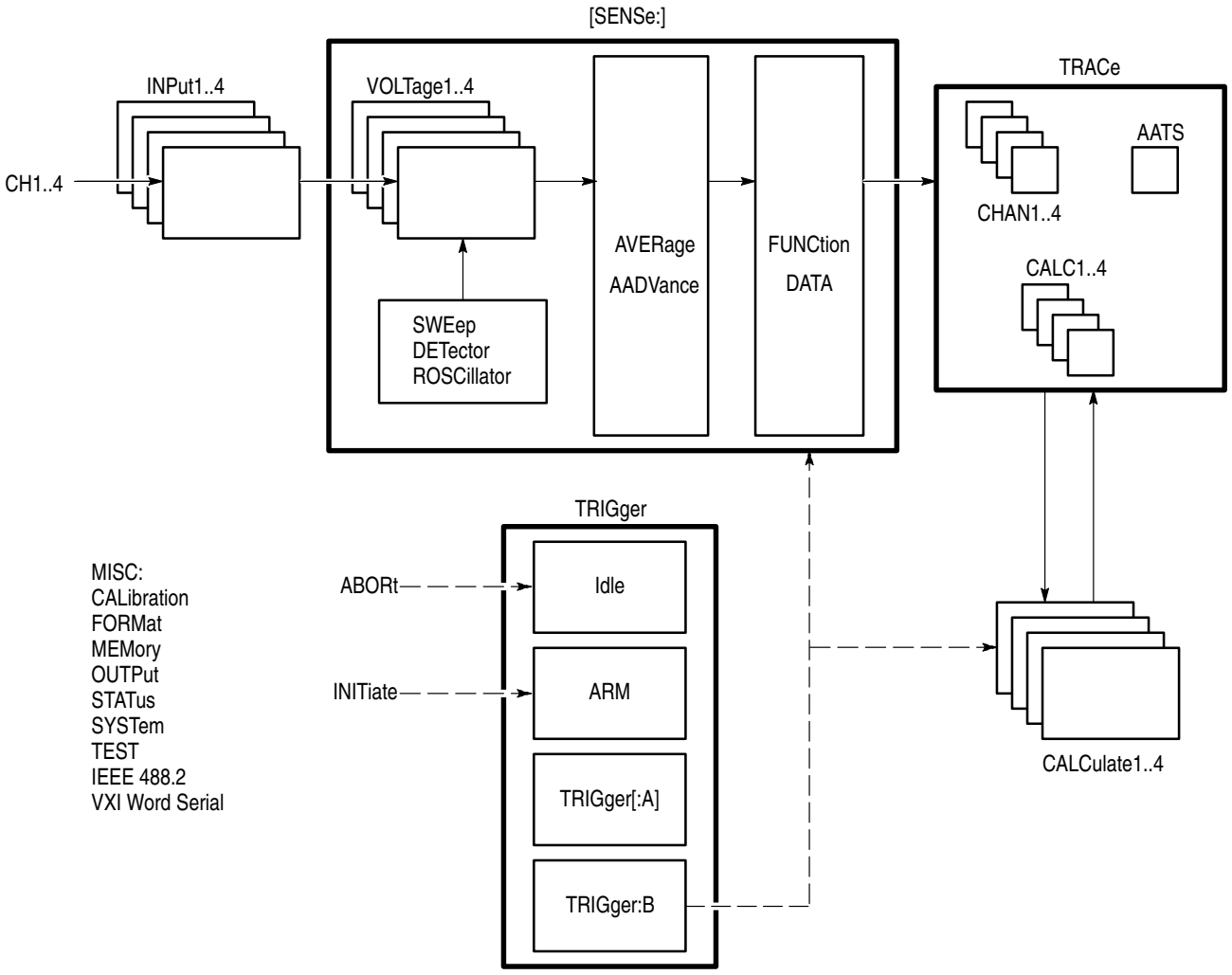


Figure 3–5: Instrument Model Showing Root-level Nodes and Subsystems

www.valuetronics.com

AADVance Subsystem

This section describes the commands in the [SENSe:]AADVance subsystem. These commands control how auto-advance acquisition records are acquired and transferred to a VXIbus controller.

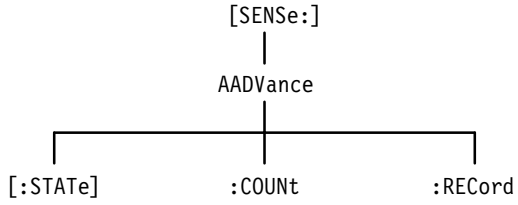


Figure 3–6: AADVance Subsystem Hierarchy

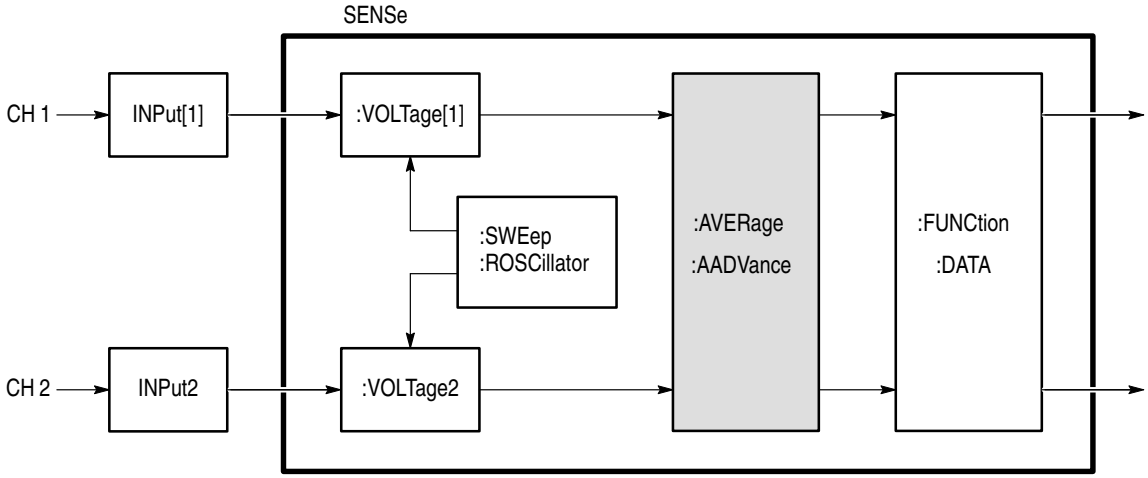


Figure 3–7: AADVance Subsystem Functional Model

www.valuetronics.com

AADVance
AADVance?

Sets or queries the state of the auto-advance acquisition mode. In the auto-advance mode the waveform analyzer acquires a sequence of data records for each active channel. The delay between one acquisition record and the next one is very short and is due only to the minimal re-arm time and any trigger holdoff you set. Use the command AADVance:COUNt to set the number of records to acquire. You cannot average acquisitions with [SENSe:]AVERage when using the auto-advance acquisition mode.

The auto-advance mode of acquisition affects all enabled channels (XTIM:VOLT <n>). You cannot acquire one channel in the auto-advance mode and acquire another with the normal acquisition mode.

Syntax

```
[SENSe:]AADVance[:STATe] <boolean>
[SENSe:]AADVance[:STATe]?
```

Parameters

<boolean>	Query Response
<NRf>	<NF1>
1 or ON	1
0 or OFF	0

Reset Value

0

Errors and Events

None

Dependencies

Enabling auto-advance acquisition disables [SENSe:]AVERage acquisition, ENVelope or SCALar type.

Examples

Command: `AADV ON`

Query: `AADV?`

Response: 1

Related Commands

AADVance:COUNt
AVERage

www.valuetronics.com

AADVance:COUNt
AADVance:COUNt?

Sets or queries the number of records to acquire in the auto-advance acquisition mode. The maximum number of Auto-advance records that you can acquire depends on the record length and the number of active channels. A setting of zero (0) acquires enough records to fill the DSP memory, regardless of the current acquisition system settings. MAXimum acquires enough records to fill DSP memory based on the current acquisition settings. The distinction is that with a change such as the number of active channels, the MAXimum COUNt setting is not adjusted, whereas the zero COUNt setting adjusts to ensure memory is just filled within the new conditions.

To determine the current value for MAX, first set all acquisition parameters. Set AADV:COUN to MAX. Finally, query with AADV:COUN? to return the current value for MAX.

Syntax

```
[SENSe:]AADVance:COUNt <count>
[SENSe:]AADVance:COUNt?
```

Parameters

<count>		Query Response
<NRf>		<NR1>
0	Fill acquisition memory	
1 to MAX	Acquire number	
MINimum		1
MAXimum		<depends on configuration>

Reset Value

1

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set count to an illegal value.

Dependencies

None

Examples

Command: `AADV:COUN 100`

Query: `AADV:COUN?`

Response: `100`

Related Commands

AADVance

www.valuetronics.com

AADVance:RECord:COUNt
AADVance:RECord:COUNt?

Sets or queries the number of auto-advance acquisition records to transfer in response to the commands DATA?, TRACe?, TRACe:COPY?, or TRACe:LIST?. The maximum :COUNt value depends on the number of acquired records and the value of AADVance:RECord:STARt. A setting of zero (0) selects all acquisition records, beginning with :STARt, for transfer regardless of the current acquisition system settings. MAXimum selects all acquisition records for transfer based on the current acquisition settings. If you change a setting, such as the number of active channels, the MAXimum setting will not be adjusted, but a setting of zero will automatically adjust for the new acquisition settings.

Syntax

```
[SENSe:]AADVance:RECord:COUNt <count>
[SENSe:]AADVance:RECord:COUNt?
```

Parameters

<count>		Query Response
<NRf>		<NR1>
0	Transfer all records	
1 to MAX	Transfer number of records	
MINimum		1
MAXimum		<depends on configuration>

Reset Value

1

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set count to an illegal value.

Dependencies

None

Examples

Command: `AADV:REC:COUN 100`

Query: `AADV:REC:COUN?`

Response: `100`

Related Commands

AADVance
AADVance:COUNt
AADVance:RECord:STARt

www.valuetronics.com

AADVance:RECord:STARt
AADVance:RECord:STARt?

Sets or queries the number of the first auto-advance waveform record to transfer in response to the commands DATA?, TRACe?, and TRACe:COPY?. The maximum :STARt value depends on the number of acquired records. A setting of zero (0) selects the last waveform record for transfer. Negative settings select waveform records referenced from the last one, the zero record. For example, the –1 record is the second from last and the –2 record is third from last.

Note that auto-advance acquisition always starts with record number one.

Syntax

```
[SENSe:]AADVance:RECord:STARt <start>
[SENSe:]AADVance:RECord:STARt?
```

Parameters

<start>		Query Response
<NRf>		<NR1>
–MAX + 1 to –1	Number before last record	
0	Last record	
1 to MAX	Number from first record	
MINimum		1
MAXimum		<depends on configuration>

Reset Value

1

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set start to an illegal value.

Dependencies

None

Examples

Command: `AADV:REC:STAR 100`

Query: `AADV:REC:STAR?`

Response: `100`

Related Commands

AADVance
AADVance:COUNt
AADVance:RECord:COUNt

www.valuetronics.com

ARM Subsystem

This section describes the commands in the ARM subsystem. These commands operate with the TRIGger, INITiate, and ABORt subsystems to trigger acquisitions.

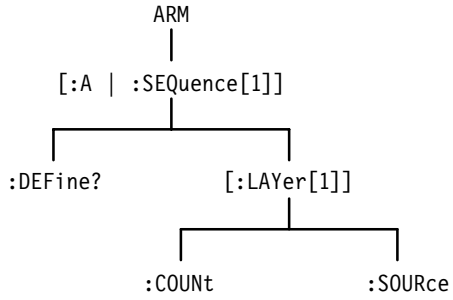


Figure 3–8: ARM Subsystem Hierarchy

ARM:DEFine? (Query Only)

Returns the predefined SEQuence1 alias. :A is a pre-defined alias for :SEQuence[1]. The commands ARM:DEFine? and TRIGger:DEFine? are aliases which produce the same result.

Syntax `ARM[:SEQuence[1]]:DEFine?`

Parameters

<sequence_alias>	Query Response
Not Applicable	<string> "A"

Reset Value "A"

Errors and Events None

Dependencies None

Examples Query: `ARM:DEF?`

www.valuetronics.com

Response: "A"

Related Commands

TRIGger:DEFine?
TRIGger:SEQuence2:DEFine?

ARM:SOURce
ARM:SOURce?

Sets or queries the source that will arm the acquisition system. You can specify only one source at a time and it is shared by all acquired channels. Setting the arm source to BUS configures the event detector to accept and arm on either the *TRG or the VXIbus word serial <Trigger> command. The ECLTrg and TTLTrg sources provide normal and inverted access to the standard ECLT and TTLT signals on the VXI P2 bus. TTLTrg arms when the TTLT line is low and ITLTrg arms when the TTLT line is high. The ECLTrg arms when the ECLT is high and ICLTrg arms when ECLT is low.

EXTernal is the front-panel BNC connector labeled Arm Input. Setting the arm source to IMMediate bypasses event detection and immediately arms the acquisition system.

Syntax

```
ARM[:A][:LAYer[1]]:SOURce <arm_source>
ARM[:A][:LAYer[1]]:SOURce?
```

Parameters

<arm_source>	Query Response
BUS	BUS
ECLTrg0	ECLT0
ECLTrg1	ECLT1
EXTernal	EXT
IECLTrg0	IECLTrg0
IECLTrg1	IECLTrg1
ITTLTrg0	ITTLTrg0
ITTLTrg1	ITTLTrg1
.	.
.	.
.	.
ITTLTrg7	ITTLTrg7
IMMediate (No source needed)	IMM
TTLTrg0	TTLT0
TTLTrg1	TTLT1
.	.
.	.
.	.
TTLTrg7	TTLT7

www.valuetronics.com

Reset Value IMM

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to set arm source to INTernal or any other invalid value.

Execution Error –212, “Arm ignored”
Sent *TRG or the VXIbus word serial <Trigger> command when ARM:SOURce is not set to BUS or when the instrument is not waiting at the ARM Event Detection layer.

Execution Error –215, “Arm deadlock”
Attempted to query the instrument when arm source is set to BUS and before sending *TRG or the VXIbus word serial <Trigger> command.

Dependencies None

Examples

Command: `ARM:SOUR TTLTRG0`

Query: `ARM:SOUR?`

Response: `TTLT0`

Related Commands TRIGger:SOURce

www.valuetronics.com

AVERage Subsystem

This section describes the commands in the [SENSe:]AVERage subsystem. See Figure 3–9. These commands control averaging in the SENSe block, which occurs prior to processing by the CALCulate block. See Figure 3–10.

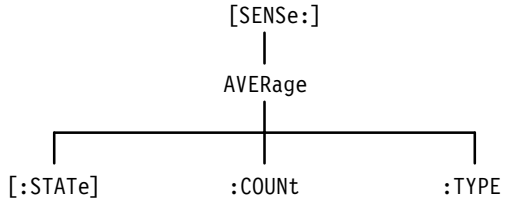


Figure 3–9: AVERage Subsystem Hierarchy

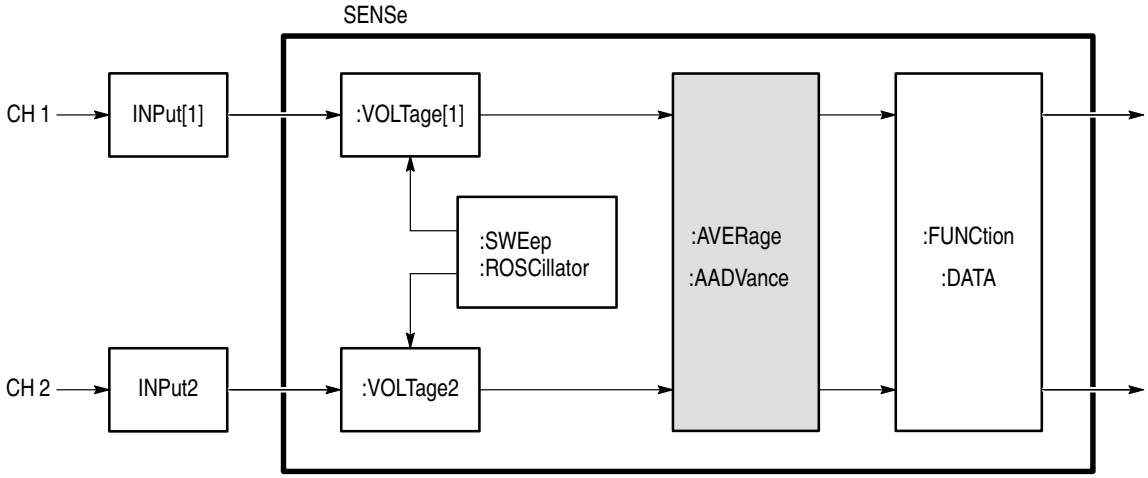


Figure 3–10: AVERage Subsystem Functional Model

AVERage
AVERage?

Sets or queries whether the waveform analyzer performs averaging during acquisition. The AVERage setting affects all channels. When averaging is enabled, the values returned from [SENSe:]DATA? are averaged. The raw, unaveraged data is not available.

www.valuetronics.com

Syntax

```
[SENSe:]AVERage[:STATe] <boolean>
[SENSe:]AVERage[:STATe]?
```

Parameters

<boolean>	Query Response
<NRf> N ≠ 0 or ON 0 or OFF	<NF1> 1 0

Reset Value 0

Errors and Events None

Dependencies Enabling averaging sets AADVance to OFF.

Examples Command: `AVER ON`

Query: `AVER?`

Response: 1

Related Commands AVERage:COUNt
AVERage:TYPE
AADVance

AVERage:COUNt
AVERage:COUNt?

Sets or queries the number of acquisition records to average. Averaging reduces signal noise by approximately 3 dB for each power of 2 increase (2, 4, 8, 16) in the value of COUNt. For example, a COUNt setting of 8 will result in 3 dB less noise than a COUNt of 4. The COUNt setting affects all active channels.

Syntax

```
[SENSe:]AVERage:COUNt <count>
[SENSe:]AVERage:COUNt?
```

Parameters

<count>	Query Response
<NRf> 2 ≤ N ≤ 4096 MINimum MAXimum	<NR1> 2 4096

www.valuetronics.com

Reset Value 2

Errors and Events Execution Error –222, "Data out of range"
Attempted to set count to an illegal value.

Dependencies None

Examples Command: `AVER:COUN 16`

Query: `AVER:COUN?`

Response: `16`

Related Commands AVERage
AVERage:TYPE

AVERage:TYPE
AVERage:TYPE?

Sets or queries the type of waveform averaging to perform. SCALar, the default, averages each new sample point with the corresponding point the previous acquisition record. ENVelope generates a waveform record of interleaved MAX/MIN sample points. The AVERage:TYPE setting affects all channels.

Syntax

```
[SENSe:]AVERage:TYPE <type>
[SENSe:]AVERage:TYPE?
```

Parameters

<type>	Query Response
ENVelope	ENV
SCALar	SCAL

Reset Value SCAL

Errors and Events Execution Error –224, "Illegal parameter value"
Attempted to set type to an illegal value.

Dependencies None

Examples Command: `AVER:TYPE ENV`

www.valuetronics.com

Query: `AVER:TYPE?`

Response: `ENV`

Related Commands

AVERage
AVERage:COUNt

www.valuetronics.com

CALCulate Subsystem

This section describes the commands in the CALCulate subsystems which process and perform measurements on acquisition records. These calculations are typically performed immediately after the waveform analyzer completes acquisition of the source. You can use CALCulate:IMMediate to reprocess and measure an existing acquisition record. Figures 3–11 and 3–12 show the command subblocks in the CALCulate subsystem.

For a description of how to process waveforms and make measurements with the commands in the CALCulate subsystem, refer to *Calculations* on page 2–31.

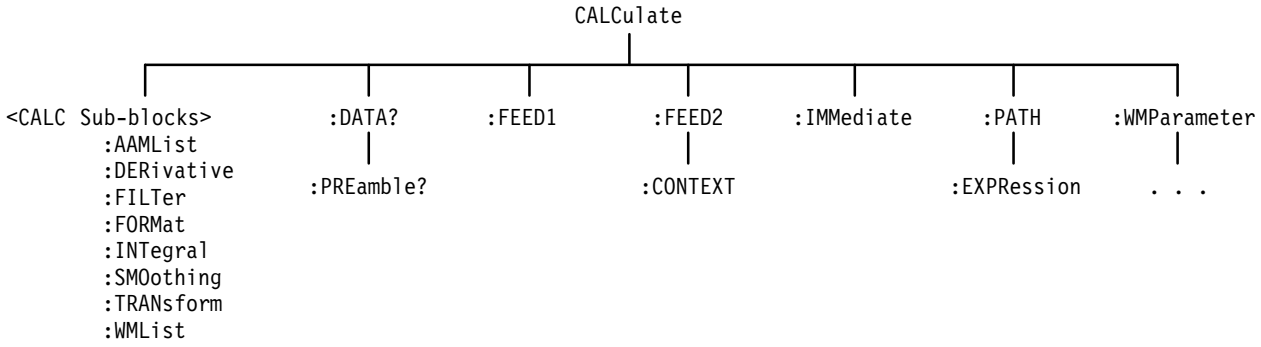


Figure 3–11: CALCulate Subsystem Hierarchy

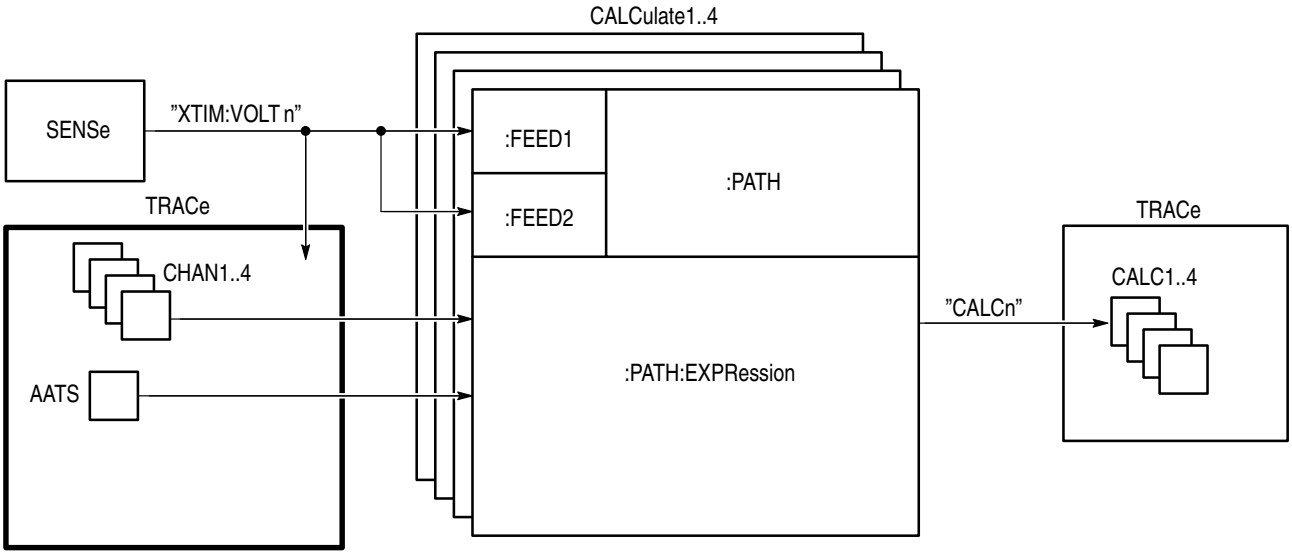


Figure 3–12: CALCulate Subsystem Functional Model

www.valuetronics.com

CALCulate:AAMList
CALCulate:AAMList?

Sets or queries the list of measurements to perform on the selected auto-advance acquisition records. You can specify up to 50 measurements by separating them with commas. Before you can make auto-advance measurements, you must enable the auto-advance measurement system with the command CALCulate:AAMList:STATe.

Specify the measurement methods and reference values with the CALC:WMParameter commands.

Syntax

```
CALCulate<n>:AAMList <list>
CALCulate<n>:AAMList?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

[1] **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<list>	Query Response
AMPLitude	AMPL
AREA	AREA
CARea	CAR
CMEan	CME
CRMS	CRMS
CROSs	CROSs
FREQuency	FREQ
FTIMe	FTIM
GAIN	GAIN
HIGH	HIGH
LOW	LOW
MAXimum	MAX
MEAN | DC	MEAN
MID	MID
MINimum	MIN
NCRoss	NCR
NDUTycycle	NDUT
NWIDth	NWID

www.valuetronics.com

<list> (Cont.)	Query Response (Cont.)
PCRossr	PCR
PERiod	PER
PDUTycycle	PDUT
PHAse	PHA
PTPeak	PTP
PWIDth	PWID
RMS | AC	RMS
RTIMe	RTIM
SDEViation	SDEV
TTRig	TTR

Reset Value MEAN

Errors and Events Execution Error –141, "Invalid character data"
Attempted to program an illegal waveform measurement.

Command Error 108, "Parameter not allowed"
Attempted to assign more than 50 measurements to the list.

Dependencies None

Examples Command: `CALC1:AAML RTIM,FTIM,PWID`

Query: `CALC1:AAML?`

Response: `RTIM,FTIM,PWID`

Related Commands CALCulate:AAMList:STATe

CALCulate:AAMList:STATe
CALCulate:AAMList:STATe?

Sets or queries whether to perform waveform measurement(s) on acquisition records captured with auto-advance acquisition. You specify Auto Advance measurements with the CALCulate:AAMList command.

Syntax

```
CALCulate<n>:AAMList:STATe <boolean>
CALCulate<n>:AAMList:STATe?
```


www.valuetronics.com

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<boolean>	Query Response
<NRf> 1 or ON 0 or OFF	<NF1> 1 0

Reset Value 0

Errors and Events None

Dependencies The setting for :STATe is ignored when you use the command CALC:PATH:EXPRession.

Examples Command: `CALC:AAML:STAT ON`

Query: `CALC:AAML:STAT?`

Response: 1

Related Commands CALCulate:AAMList
ADDV ON|OFF

CALCulate:DATA? (Query Only)

This query returns the results of waveform processing and measurement functions performed on acquired waveform records. The default format of the returned data is ASCII. However, you can set the data format with the FORMat:CALCulate command.

Results are returned only when pending acquisitions and calculations are complete. The synchronizing commands *WAI, *OPC, and *OPC? are not required unless you wish to synchronize the transfer differently.

www.valuetronics.com

The CALCulate and TRACe subsystems use consistent naming, such that CALC1:DATA? is equivalent to TRAC? CALC1.

Syntax `CALCulate<n>:DATA?`

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

Reset Value Not applicable

Errors and Events Execution Error –230, "Data corrupt or stale."
Attempted to query data that is invalid, incomplete or stale.

Dependencies None

Examples Query: `CALC1:DATA?`

Response: `<arb_block_data>`

Related Commands [SENSe:]DATA?
[SENSe:]DATA:PREamble?
CALCulate:DATA:PREamble?
TRACe?
TRACe:PREamble?

CALCulate:DATA:PREamble? (Query Only)

Returns the data preamble for acquisition record processing and measurement functions. Results are returned only when pending acquisitions and calculations are complete. The default format of the returned data is ASCII. However, you can set the data format with the FORMat:CALCulate command. The synchronizing commands *WAI, *OPC, and *OPC? are not required unless you wish to synchronize the transfer differently.

The CALCulate and TRACe subsystems use consistent naming, such that CALC1:DATA? is equivalent to TRAC? CALC1.

www.valuetronics.com

Syntax `CALCulate<n>:DATA:PREamble?`

Parameters

<n>[1]	Query Response
1 2 3 4	NA

[1] **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

Reset Value Not applicable

Errors and Events Execution Error –230, "Data corrupt or stale."
Attempted to query the preamble for data that is invalid, incomplete or stale.

Dependencies None

Examples Query: `CALC1:DATA:PRE?`

Response: `(DIF Expression)`

Related Commands [SENSe:]DATA?
[SENSe:]DATA:PREamble?
CALCulate:DATA?
TRACe?
TRACe:PREamble?

CALCulate:DERivative:STATe
CALCulate:DERivative:STATe?

Sets or queries whether to calculate a post-acquisition derivative on the selected channel. When ON, the waveform analyzer calculates the derivative for every point in an acquisition record. The result is a CALC<n> data record with the same number of points as the original acquisition record.

When you perform a calculation with the command CALC:PATH:EXPRession, the value of :STATe is ignored.

Syntax `CALCulate<n>:DERivative:STATe<boolean>`
`CALCulate<n>:DERivative:STATe?`

www.valuetronics.com

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<boolean> (DERivative state)	Query Response
<NRf> 1 or ON 0 or OFF	<NR1> 1 0

Reset Value 0

Errors and Events None

Dependencies None

Examples Command: `CALC1:DER:STAT ON`

Query: `CALC1:DER:STAT?`

Response: 1

Related Commands CALCulate:INTegral:STATe

CALCulate:FEED1
CALCulate:FEED1?

Sets or queries the source of data for the specified CALCulate block. The calculations you set for a CALC block are performed on the source chosen with this command. The source may be from any channel or SENSe function, it is not tied to the CALC block number. For example, the FEED1 for CALC2 could be "XTIM:VOLT 1" or "XTIM:VOLT 3." Additionally, more than one CALC block can operate on the same source.

Trace names and function strings may be used interchangeably for the source parameter <sense_func>. To follow standard SCPI practice and ensure program

www.valuetronics.com

compatibility with other products, use the XTIM:VOLT function string instead of the CHAN trace name. The query form always returns the function strings.

When you perform a calculation with the command CALC:PATH:EXPRession, the value of :FEED1 is ignored.

When specifying calculations that require more than one waveform, such as gain, phase, or delay, :FEED1 specifies the reference waveform and :FEED2 specifies the target waveform.

Syntax

```
CALCulate<n>:FEED1 <sense_func>
CALCulate<n>:FEED1?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<sense_func> (CALC block source)			Query Response
<string>			<string>
“XTIMe:VOLTage[:DC] 1”	or	CHAN1	“XTIM:VOLT 1”
“XTIMe:VOLTage[:DC] 2”	or	CHAN2	“XTIM:VOLT 2”
“XTIMe:VOLTage[:DC] 3”	or	CHAN3	“XTIM:VOLT 3”
“XTIMe:VOLTage[:DC] 4”	or	CHAN4	“XTIM:VOLT 4”
“ ”	or	NONE	“ ”

Reset Value

"" (NONE)

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to set FEED1 to an invalid trace name.

Execution Error –224, “Illegal parameter value”
Attempted to set FEED1 to an invalid sense function.

Execution Error –241, “Hardware missing”
Attempted to set feed to “XTIM:VOLT 3” or “XTIM:VOLT 4” when the instrument is configured with two channels.

Dependencies

None

www.valuetronics.com

Examples

Command: `CALC1:FEED1 "XTIM:VOLT 1"`

Query: `CALC1:FEED1?`

Response: `"XTIM:VOLT 1"`

Related Commands

CALCulate:PATH

CALCulate:FEED2
CALCulate:FEED2?

Sets or queries a second source of data for the specified CALCulate block. The second source is used when measurements that are based on two waveforms, such as gain, phase, and delay, are specified using the CALCulate:WMList command.

The source may be from any channel or SENSe function, it is not tied to the CALC block number. For example, the FEED2 for CALC2 could be "XTIM:VOLT 1" or "XTIM:VOLT 3." Additionally, more than one CALC block can operate on the same source.

Trace names and function strings may be used interchangeably for the source parameter <sense_func>. To follow standard SCPI practice and ensure program compatibility with other products, use the XTIM:VOLT function string instead of the CHAN trace name. The query form always returns the function strings.

When you perform a calculation with the command CALC:PATH:EXPRession, the value of :FEED2 is ignored.

When defining SCPI model calculations that measure parameters based on two waveforms (such as Gain and Delay), :FEED1 specifies the reference waveform and :FEED2 specifies the target waveform. Also, the functions you set using CALC:PATH only apply to the data source selected with FEED1. The source selected by FEED2 is unaffected by the functions you select.

Syntax

```
CALCulate<n>:FEED2 <sense_func>
CALCulate<n>:FEED2?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

www.valuetronics.com

<sense_func> (CALC block source)			Query Response
<string>			<string>
“XTIMe:VOLTage[:DC] 1”	or	CHAN1	“XTIM:VOLT 1”
“XTIMe:VOLTage[:DC] 2”	or	CHAN2	“XTIM:VOLT 2”
“XTIMe:VOLTage[:DC] 3”	or	CHAN3	“XTIM:VOLT 3”
“XTIMe:VOLTage[:DC] 4”	or	CHAN4	“XTIM:VOLT 4”
“ ”	or	NONE[1]	“” (empty string returned)

1 **Specifying NONE causes :FEED2 to default to the same source as used for :FEED1. If such is the case, the same waveform will be used for taking the gain, phase, or delay measurement.**

Reset Value "" (NONE)

Errors and Events Execution Error –141, “Invalid character data”
Attempted to set FEED2 to an invalid trace name.

Execution Error –224, “Illegal parameter value”
Attempted to set FEED2 to an invalid sense function.

Execution Error –241, “Hardware missing”
Attempted to set feed to “XTIM:VOLT 3” or “XTIM:VOLT 4” when the instrument is configured with two channels.

Dependencies None

Examples Command: `CALC1:FEED2 "XTIM:VOLT 1"`

Query: `CALC1:FEED2?`

Response: `"XTIM:VOLT 1"`

Related Commands CALCulate:PATH

CALCulate:FEED2:CONTEXT
CALCulate:FEED2:CONTEXT?

Sets or queries the measurement parameter block, one of CALC1–CALC4, used to characterize the FEED2 data.

Any of the four CALC blocks, CALC1–CALC4, may be used interchangeably for the context parameter <calc_block>.

www.valuetronics.com

When you perform a calculation with the command CALC:PATH:EXPRession, the value of :FEED2:CONTEXT is ignored.

Syntax

```
CALCulate<n>:FEED2:CONTEXT <calc_block>
CALCulate<n>:FEED2:CONTEXT?
```

Parameters

<calc_block> (CALC block source)	Query Response
CALC1	CALC1
CALC2	CALC2
CALC3	CALC3
CALC4	CALC4

Reset Values

Context	Value
CALC1:FEED2:CONTEXT	CALC1
CALC2:FEED2:CONTEXT	CALC2
CALC3:FEED2:CONTEXT	CALC3
CALC4:FEED2:CONTEXT	CALC4

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to set FEED2:CONTEXT to an invalid parameter value.

Dependencies

None

Examples

Command: `CALC1:FEED2:CONTEXT CALC4`

Query: `CALC1:FEED2:CONTEXT?`

Response: `CALC4`

Related Commands

CALCulate:FEED1

CALCulate:FEED2

www.valuetronics.com

CALCulate:FILTer:FREQuency[:TYPE]
CALCulate:FILTer:FREQuency[:TYPE]?

Sets or queries the type of FREQuency filtering to perform on an acquisition record.

The available filter types are as follows:

- BPASs—rejects frequency components outside the defined frequency range.
- NOTCh—rejects frequency components within a defined frequency range.
- HPASs—rejects frequency components below a specified frequency.
- LPASs—rejects frequency components above a specified frequency.

When you set :TYPE to bandpass or notch, set the filter parameters with the :FILTer:FREQuency commands STARt, STOP, CENTer, and SPAN. When you set :TYPE to high pass or low pass, set the filter parameters with the commands HPASs or LPASs. For an overview of the measurement system, refer to the *Calculations* discussion on page 2–31.

Syntax

```
CALCulate<n>:FILTer[:GATE]:FREQuency[:TYPE] <type>
CALCulate<n>:FILTer[:GATE]:FREQuency[:TYPE]?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<type> (Frequency filter)	Query Response
BPASs	BPAS
HPASs	HPAS
LPASs	LPAS
NOTCh	NOTC

Reset Value

BPAS

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to set type to an illegal value.

www.valuetronics.com

Dependencies None

Examples Command: `CALC1:FILT:FREQ NOTCH`

Query: `CALC1:FILT:FREQ?`

Response: `NOTC`

Related Commands CALCulate:FILTer:FREQuency:CENTer
CALCulate:FILTer:FREQuency:HPASs
CALCulate:FILTer:FREQuency:LPASs
CALCulate:FILTer:FREQuency:SPAN
CALCulate:FILTer:FREQuency:STARt
CALCulate:FILTer:FREQuency:STATe
CALCulate:FILTer:FREQuency:STOP

CALCulate:FILTer:FREQuency:CENTer
CALCulate:FILTer:FREQuency:CENTer?

Sets or queries the center frequency of the bandpass or notch filter. Use the command :FREQuency:SPAN to set the frequency range for :NOTCH and BPASS frequency filters. For example, if you set :CENTer to 10 MHz and :SPAN to 2 MHz, then the :NOTCH filter will have a range of 9 MHz to 11 MHz. For an overview of the measurement system, refer to the *Calculations* discussion on page 2–31.

Syntax

```
CALCulate<n>:FILTer[:GATE]:FREQuency:CENTer <center>
CALCulate<n>:FILTer[:GATE]:FREQuency:CENTer?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

www.valuetronics.com

(Center frequency of SPAN) [1]	Query Response
<NRf>	<NR3>

[1] **The default multiplier for the parameter is HZ for hertz. You can also use the multipliers KHZ for kilohertz, MHZ for megahertz, and GHZ for gigahertz.**

Reset Value 250.0E+6

Errors and Events Execution Error –222, "Data out of range"
Attempted to set center to an illegal value.

Dependencies You can set the frequency range for the frequency filters :NOTCH and :BPASS with either the command pair :CENTer and :SPAN or the command pair :STARt and :STOP. Because both command pairs control the same parameters, changing either command pair will affect the other.

Examples Command: `CALC1:FILT:FREQ:CENT 10E6`

Query: `CALC1:FILT:FREQ:CENT?`

Response: `10.0E+6`

Related Commands CALCulate:FILTer:FREQuency:SPAN

www.valuetronics.com

CALCulate:FILTer:FREQuency:HPASs
CALCulate:FILTer:FREQuency:HPASs?

Sets or queries the limit frequency (<cutoff>) below which the filter attenuates all frequency components at 6 dB. The :HPASs frequency is ignored when you select the bandpass, notch or low-pass filter. Use the command CALCulate:FILTer:FREQuency HPASs to enable the high-pass filter.

Syntax

```
CALCulate<n>:FILTer[:GATE]:FREQuency:HPASs <cutoff>
CALCulate<n>:FILTer[:GATE]:FREQuency:HPASs?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<cutoff> (Limit frequency)[1]	Query Response
<NRf>	<NR3>

1 **The default multiplier for the <cutoff> parameter is HZ for hertz. You can also use the multipliers KHZ for kilohertz, MHZ for megahertz, and GHZ for gigahertz**

Reset Value

250.0E+6

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set the high pass cutoff frequency to an illegal value.

Dependencies

None

Examples

Command: `CALC1:FILT:FREQ:HPAS 10E6`

Query: `CALC1:FILT:FREQ:HPAS?`

Response: `10.0E+6`

Related Commands

CALCulate:FILTer:FREQuency:LPASs

www.valuetronics.com

CALCulate:FILTer:FREQuency:LPASs
CALCulate:FILTer:FREQuency:LPASs?

Sets or queries the limit frequency (<cutoff>) above which the filter attenuates all frequency components at 6 dB. The :LPASs frequency is ignored when filter type is set to bandpass, notch or high pass. Use the command CALCulate:FILTer:FREQuency LPASs to enable the low-pass filter.

Syntax

```
CALCulate<n>:FILTer[:GATE]:FREQuency:LPASs <cutoff>
CALCulate<n>:FILTer[:GATE]:FREQuency:LPASs?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<cutoff> (Limit frequency)[1]	Query Response
<NRf>	<NR3>

1 **The default multiplier for the <cutoff> parameter is HZ for hertz. You can also use the multipliers KHZ for kilohertz, MHZ for megahertz, and GHZ for gigahertz**

Reset Value 250.0E+6

Errors and Events Execution Error –222, “Data out of range”
Attempted to set the low-pass cutoff frequency to an illegal value.

Dependencies None

Examples Command: `CALC1:FILT:FREQ:LPAS 10E6`

Query: `CALC1:FILT:FREQ:LPAS?`

Response: `10.0E+6`

Related Commands CALCulate:FILTer:FREQuency:HPASs

www.valuetronics.com

CALCulate:FILTer:FREQuency:SPAN
CALCulate:FILTer:FREQuency:SPAN?

Sets or queries the frequency range to be used by the bandpass and notch filters. After you define the range with :SPAN, you position the range within the available spectrum using the command :FREQuency:CENTer. For example, if you set :SPAN to 2 MHz and :CENTer to 10 MHz, then the :NOTCH filter will have a range of 9 MHz to 11 MHz.

The frequency span is ignored when you select the high-pass (:HPASs) or low-pass (:LPASs) filter.

Syntax

```
CALCulate<n>:FILTer[:GATE]:FREQuency:SPAN <span>
CALCulate<n>:FILTer[:GATE]:FREQuency:SPAN?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

(Frequency range)[1]	Query Response
<NRf>	<NR3>

1 **The default multiplier for the parameter is HZ for hertz. You can also use the multipliers KHZ for kilohertz, MHZ for megahertz, and GHZ for gigahertz.**

Reset Value

100.0E+6

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set span to an illegal value.

Dependencies

You can set the frequency range for the frequency filters :NOTCH and :BPASS with either the command pair :CENTer and :SPAN or the command pair :STARt and :STOP. Because both command pairs control the same parameters, changing either command pair will affect the other.

Examples

Command: `CALC1:FILT:FREQ:SPAN 10E3`

www.valuetronics.com

Query: `CALC1:FILT:FREQ:SPAN?`

Response: `10.0E+3`

Related Commands CALCulate:FILTer:FREQuency:CENTer

CALCulate:FILTer:FREQuency:SREJection
CALCulate:FILTer:FREQuency:SREJection?

Sets or queries the level of rejection or attenuation for frequency components in the defined stop band for the:FREQuency filters. :SREJEction sets the level of attenuation in dB for the filters :NOTCh, :BPASs, :LPASs and :HPASs. You can use :FREQuency:TWIDth to set the slope or roll off of the filter.

If you enter too large a value for :SREJEction, the function will generate an execution error and terminate the filter process. If an execution error occurs, increase the record length or increase the value of :TWIDth. For an overview discussion of the waveform analyzer digital filter, refer to the Calculations discussion on page 2–31.

Syntax

```
CALCulate<n>:FILTer[:GATE]:FREQuency:SREJection <level>
CALCulate<n>:FILTer[:GATE]:FREQuency:SREJection?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<level> (Frequency rejection in dB)	Query Response
<NRf> 15 ≤ N ≤ 100 MINimum MAXimum	<NR2> 15.0 100.0

Reset Value 60.0

www.valuetronics.com

Errors and Events Execution Error –222, "Data out of range"
Attempted to set rejection to an illegal value.

Dependencies None

Examples Command: `CALC1:FILT:FREQ:SREJ 75`

Query: `CALC1:FILT:FREQ:SREJ?`

Response: `75.0`

Related Commands CALCulate:FILTer:FREQuency:TWIDth

CALCulate:FILTer:FREQuency:STARt
CALCulate:FILTer:FREQuency:STARt?

Sets or queries the 6 dB start, or lower limit, frequency of the bandpass and notch filters. :STARt and :STOP define the frequency range for the filters :FREQuency:BPASs and :FREQuency:NOTCh. The commands :FREQuency:CENTer and :FREQuency:SPAN provide an alternate way to set the filter range. For an overview discussion of the waveform analyzer digital filter, refer to the Calculations discussion on page 2–31.

The :STARt frequency is ignored when you select the high-pass or low-pass filter.

Syntax

```
CALCulate<n>:FILTer[:GATE]:FREQuency:STARt <start>
CALCulate<n>:FILTer[:GATE]:FREQuency:STARt?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

www.valuetronics.com

<start> (Beginning frequency of range)[1]	Query Response
<NRf>	<NR3>

1 **The default multiplier for the parameter <start> is HZ for hertz. You can also use the multipliers KHZ for kilohertz, MHZ for megahertz, and GHZ for gigahertz.**

Reset Value 200.0E+6

Errors and Events Execution Error –222, "Data out of range"
Attempting to set start to an illegal value will generate execution error.

Dependencies You can set the frequency range for the frequency filters :NOTCH and :BPASS with either the command pair :CENTer and :SPAN or the command pair :STARt and :STOP. Because both command pairs control the same parameters, changing either command pair will affect the other.

Examples Command: `CALC1:FILT:FREQ:STAR 9.99E6`

Query: `CALC1:FILT:FREQ:STAR?`

Response: `9.99E+6`

Related Commands CALCulate:FILTer:FREQuency:STOP
CALCulate:FILTer:FREQuency:CENTer
CALCulate:FILTer:FREQuency:SPAN

www.valuetronics.com

CALCulate:FILTer:FREQuency:STATe
CALCulate:FILTer:FREQuency:STATe?

Sets or queries whether the selected CALCulate block performs frequency filtering on acquisition records.

If you define an expression with CALC:PATH:EXPRession, then the value of STATe setting is ignored.

Syntax

```
CALCulate<n>:FILTer[:GATE]:FREQuency:STATe<boolean>
CALCulate<n>:FILTer[:GATE]:FREQuency:STATe?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<boolean> (Set filtering on or off)	Query Response
<NRf> 1 or ON 0 or OFF	<NR1> 1 0

Reset Value 0

Errors and Events None

Dependencies None

Examples Command: `CALC1:FILT:FREQ:STAT ON`

Query: `CALC1:FILT:FREQ:STAT?`

Response: 1

Related Commands CALCulate:FILTer:FREQuency[:TYPE]

www.valuetronics.com

CALCulate:FILTer:FREQuency:STOP
CALCulate:FILTer:FREQuency:STOP?

Sets or queries the 6 dB stop, or upper limit, frequency of the bandpass and notch filters. :STARt and :STOP define the frequency range for the filters :FREQuency:BPASs and :FREQuency:NOTCh. The commands :FREQuency:CENTer and :FREQuency:SPAN provide an alternate way to set the filter range.

The STOP frequency is ignored when filter type is set to high pass or low pass.

Syntax

```
CALCulate<n>:FILTer[:GATE]:FREQuency:STOP <stop>
CALCulate<n>:FILTer[:GATE]:FREQuency:STOP?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<stop> (Ending frequency of range)[1]	Query Response
<NRf>	<NR3>

1 **The default multiplier for the parameter <stop> is HZ for hertz. You can also use the multipliers KHZ for kilohertz, MHZ for megahertz, and GHZ for gigahertz.**

Reset Value

300.0E+6

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set stop to an illegal value.

Dependencies

You can set the frequency range for the frequency filters :NOTCH and :BPASS with either the command pair :CENTer and :SPAN or the command pair :STARt and :STOP. Because both command pairs control the same parameters, changing either command pair will affect the other.

Examples

Command: `CALC1:FILT:FREQ:STOP 10.01E6`

Query: `CALC1:FILT:FREQ:STOP?`

www.valuetronics.com

Response: 10.01E+6

Related Commands

CALCulate:FILTer:FREQuency:STARt
CALCulate:FILTer:FREQuency:CENTer
CALCulate:FILTer:FREQuency:SPAN

CALCulate:FILTer:FREQuency:TWIDth
CALCulate:FILTer:FREQuency:TWIDth?

Sets or queries the slope of roll off for the post-acquisition filter. The filter slope is specified as a ratio of the absolute transition width (in Hz) to Nyquist frequency defined as 1 / (2 × <sample interval>). A low value for transition width produces a steep slope for the filter while a large value produces a gradual slope or roll off. You can also specify :TWIDth in percent.

If you enter too small a value for :TWIDth, the function will generate too many filter coefficients resulting in an execution error and termination of the filter process. For more information, refer to the filter discussion on page 2–31.

Syntax

```
CALCulate<n>:FILTer[:GATE]:FREQuency:TWIDth <width>
CALCulate<n>:FILTer[:GATE]:FREQuency:TWIDth?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<width> (Rate of transition in percent)	Query Response
<NRf> 0.0004 ≤ N ≤ 1.0 MINimum MAXimum	<NR2> 0.0004 1.0

Reset Value

0.1

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set transition width to an illegal value.

www.valuetronics.com

Dependencies None

Examples Command: `CALC1:FILT:FREQ:TWID 0.2`

Query: `CALC1:FILT:FREQ:TWID?`

Response: `0.2`

Related Commands CALCulate:FILTer:FREQuency:SREJection

CALCulate:FORMat
CALCulate:FORMat?

Sets or queries wether to process the acquisition record to produce a new format with complex results. The available format types are:

- COMPlex—produce an xy pair for each sample point.
- MLINear—calculate the square root of the sum of the squares of x and y.
- M LOGarithmic—calculate the $\log_{10}$ of the sum of the square of x and y.
- NONE—produces no change in the acquisition record data.
- PHASe—calculates the arctan of y over x.
- POLar—calculates a pair of points r1 and r2, where r1 is the square root of the sum of the squares of x and y and r2 is the arctan of y over x.

Setting format to NONE effectively disables the function :FORMat.

When the source waveform is not complex and format is set to something other than NONE, the value for y in x +jy is assumed to be zero.

Syntax

```
CALCulate<n>:FORMat <format>
CALCulate<n>:FORMat?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

www.valuetronics.com

<format>	Query Response
COMPlex	COMP
MLINear	MLIN
MLOGarithmic	MLOG
NONE	NONE
PHASe	PHAS
POLar	POL

Reset Value None

Errors and Events Execution Error –141, "Invalid character data"
Attempted to set format to an illegal value.

Dependencies None

Examples Command: `CALC1:FORM MLOG`

Query: `CALC1:FORM?`

Response: `MLOG`

Related Commands CALCulate:TRANsform:FREQuency

www.valuetronics.com

CALCulate:IMMediate
CALCulate:IMMediate?

Sets or queries whether the specified CALCulate block will reprocess SENSe data without reacquiring new data. The calculation is performed immediately and, if the query form is sent, the resulting data is returned afterward. The query form CALC:IMM? is semantically equivalent to CALC:IMM;DATA?. The format of the resulting data is determined by the FORMat subsystem.

This command sets the ACQ_OPC pending flag. *WAI, *OPC, or *OPC? may be used to synchronize the command.

Syntax

```
CALCulate<n>:IMMediate
CALCulate<n>:IMMediate?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<data>	Query Response
Not Applicable	Defined by FORMat:CALCulate

Reset Value

Not applicable

Errors and Events

The only errors are those associated with the defined measurements or calculations. Most calculation errors appear only at execution time rather than when you define a measurement list. The most common errors are Execution Error –260, “Expression error” and the Questionable Event errors: 2090, “Calculate1 questionable”; 2100, “Calculate2 questionable”; 2110, “Calculate3 questionable”; 2120, “Calculate4 questionable”.

Dependencies

None

Examples

Command: `CALC1:IMM`

Query: `CALC1:IMM?`

www.valuetronics.com

Response: <data>

Related Commands CALCulate:DATA?

CALCulate:INTegral:STATe
CALCulate:INTegral:STATe?

Sets or queries whether to process the acquisition record to produce an integral acquisition record.

The value of STATe is ignored when CALC:PATH:EXPRession is defined.

Syntax

```
CALCulate<n>:INTegral:STATe <boolean>
CALCulate<n>:INTegral:STATe?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<boolean> (Integral processing on or off)	Query Response
<NRf> 1 or ON 0 or OFF	<NR1> 1 0

Reset Value 0

Errors and Events None

Dependencies None

Examples Command: CALC1:INT:STAT ON

Query: CALC1:INT:STAT?

Response: 1

www.valuetronics.com

Related Commands CALCulate:DERivative:STATe

CALCulate:PATH
CALCulate:PATH?

Sets or queries a list of CALCulate functions to execute in the order listed upon completion of acquisition. You specify the :PATH as a list of functions separated by commas. The CALCulate subsystem performs the calculations in sequential order on the source you define with CALCulate:FEED. CALCulate:PATH supports simple post-processing measurements. When a measurement cannot be expressed in a simple linear fashion, use the command CALC:PATH:EXPRession.

Syntax

```
CALCulate<n>:PATH <path>
CALCulate<n>:PATH?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<path> (Calculate subblocks to execute)	Query Response
AAMList	AAML
DERivative	DER
FILTer	FILT
FORMat	FORM
INTegral	INT
SMOothing	SMO
TRANsform[1]	TRAN
WMList	WML

1 **TRANsform refers to the CALCulate:TRANsform:FREQuency commands.**

Reset Value SMO, DER, INT, FILT, TRAN, FORM, WML

Errors and Events Command Error 108, “Parameter not allowed”
Attempted to assign more than 12 functions to PATH.

www.valuetronics.com

Execution Error –141, “Invalid character data”
Attempted to program an invalid function.

Other calculation errors cannot be detected until execution time.

Dependencies None

Examples Command: `CALC1:PATH FILT,WML`

Query: `CALC1:PATH?`

Response: `FILT,WML`

Related Commands CALCulate:AAMList
CALCulate:DERivative
CALCulate:FILTer
CALCulate:FORMat
CALCulate:INTegral
CALCulate:SMOothing
CALCulate:TRANsform:FREQuency
CALCulate:WMList

CALCulate:PATH:EXPRession
CALCulate:PATH:EXPRession?

Sets or queries a measurement expression using the standard syntax of the C programming language. When you define an :EXPRession it will be calculated in place of any calculation defined with the CALCulate:PATH command. The :EXPRession command allows you to perform calculations that cannot be expressed in a linear or sequential form.

Commands that control individual subblock operation also apply when the subblock is called by the CALC:PATH:EXPRession. For instance, whether you use CALC:PATH or CALC:PATH:EXPRession to call the high pass filter (CALC:FILT:FREQ:HPAS), you must still use CALC:FILT:FREQ:HPASs <cutoff> to set the lower limit frequency.

Sources to :EXPRession must be specified as trace names instead of as data aliases, such as “XTIM:VOLT 1”. The settings of CALCulate:PATH and CALCulate:FEED are ignored, as are the :STATe commands of the subblocks.

Syntax

```
CALCulate<n>:PATH:EXPRession <path_expression>
CALCulate<n>:PATH:EXPRession?
```


www.valuetronics.com

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<path_expression> (C syntax expression)	Query Response
<expression>	<expression> (standard C language syntax)

Reset Value

( )

Errors and Events

Execution Error –241, "Hardware missing"
Attempted to define a path expression that requires CHAN3 or CHAN4 when the instrument is configured with two channels.

The :CALCulate subblock cannot detect most errors until execution time. If errors do occur, they are described with the failed :CALCulate function or with the description of any other failed system function.

Dependencies

When CALC:PATH:EXPRession is defined, the settings of CALCulate:PATH and CALCulate:FEED are ignored, as are the :STATe commands of the subblocks.

Examples

Command: `CALC1:PATH:EXPR (RTIM(CHAN1+CHAN2))`

Query: `CALC1:PATH:EXPR?`

Response: `(RTIM(CHAN1+CHAN2))`

Related Commands

CALCulate:PATH

www.valuetronics.com

CALCulate:SMOothing
CALCulate:SMOothing?

Sets or queries whether to perform smoothing on an acquisition record. Smoothing replaces each data point with the average for a specified number of adjacent data points. To set the number of data points to average, use :SMOothing:POINTs.

Syntax

```
CALCulate<n>:SMOothing[:STATe]<boolean>
CALCulate<n>:SMOothing[:STATe]?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<boolean> (Set smoothing on or off)	Query Response
<NRf> 1 or ON 0 or OFF	<NR1> 1 0

Reset Value

0

Errors and Events

None

Dependencies

None

Examples

Command: `CALC1:SMO ON`

Query: `CALC1:SMO?`

Response: 1

Related Commands

CALCulate:SMOothing:POINts

www.valuetronics.com

CALCulate:SMOothing:POINts
CALCulate:SMOothing:POINts?

Sets or queries the number of adjacent points to average in the acquisition record when smoothing is enabled. Use CALCulate:SMOothing to enable smoothing for a particular CALCulate block.

The maximum number of data points to average is limited to one fourth the length of the acquisition record.

Syntax

```
CALCulate<n>:SMOothing:POINts <points>
CALCulate<n>:SMOothing:POINts?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<points>	Query Response
<NRf> 2 ≤ N ≤ 65536 MINimum MAXimum	<NR1> 2 <record length>

Reset Value 2

Errors and Events Execution Error –222, "Data out of range"
Attempted to set points to an illegal value.

Dependencies None

Examples Command: `CALC1:SMO:POIN 8`

Query: `CALC1:SMO:POIN?`

Response: 8

www.valuetronics.com

Related Commands CALCulate:SMOothing

CALCulate:TRANsform:FREQuency:STATe
CALCulate:TRANsform:FREQuency:STATe?

Sets or queries whether to perform a Fast Fourier Transform (FFT) on the specified acquisition record to produce an equivalent frequency representation. The FFT function produces a complex array of pairs of real and imaginary data points. The number of data pairs in the resultant record is half the number of data points in the original record.

You can convert the complex data produced by the FFT function into magnitude or phase data using the CALC:FORMat block. Use the command :TRANsform:FREQuency:WINDow to select the type of data windowing (or shaping) used prior to the transformation.

Syntax

```
CALCulate<n>:TRANsform:FREQuency:STATe<boolean>
CALCulate<n>:TRANsform:FREQuency:STATe?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

[1] **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<boolean> (Set FFT transform on or off)	Query Response
<NRf> 1 or ON 0 or OFF	<NR1> 1 0

Reset Value 0

Errors and Events None

Dependencies None

Examples Command: `CALC1:TRAN:FREQ:STAT ON`

www.valuetronics.com

Query: `CALC1:TRAN:FREQ:STAT?`

Response: 1

Related Commands CALCulate:TRANsform:FREQuency:WINDow
CALCulate:FORMat

CALCulate:TRANsform:FREQuency:WINDow
CALCulate:TRANsform:FREQuency:WINDow?

Sets or queries the type of data windowing (or shaping) to use prior to the FFT transformation. The FFT window acts as a bandpass filter. The window types and there typical use are as follow:

- BHARris — Widest pass band and lowest side lobes. Best for viewing a broad spectrum.
- BLACkman — Best window for measuring the amplitude of frequencies but worst at resolving frequencies.
- HAMMing — Very good window for resolving frequencies that are very close to the same value with somewhat improved amplitude accuracy over the rectangular window.
- HANNing — Very good window for measuring amplitude accuracy but degraded for resolving frequencies.
- RECTangular — Best type of window for resolving frequencies that are very close to the same value but worst for the accuracy of amplitude for those frequencies. Best type for measuring the frequency spectrum of nonrepetitive signals and measuring frequency components near DC.
- TRIangular — Least attenuation of side lobes. The triangular window is the convolution of two rectangles half the width of the window.

For more information on the FFT function and the use of window filters, refer to the Calculations discussion on page 2–31.

Syntax

```
CALCulate<n>:TRANsform:FREQuency:WINDow <window>
CALCulate<n>:TRANsform:FREQuency:WINDow?
```


www.valuetronics.com

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<window>	Query Response
BHARris	BHAR
BLACkman	BLAC
HAMMing	HAMM
HANNing	HANN
RECTangular	RECT
TRIangular	TRI

Reset Value BHAR

Errors and Events Execution Error –141, "Invalid character data"
Attempted to set window to an illegal value.

Dependencies None

Examples Command: `CALC1:TRAN:FREQ:WIND HAMM`

Query: `CALC1:TRAN:FREQ:WIND?`

Response: `HAMM`

Related Commands CALCulate:TRANsform:FREQuency:STATe

www.valuetronics.com

CALCulate:WMList
CALCulate:WMList?

Sets or queries the list of waveform measurements to perform. Table 3–25 lists and describes the available measurements. All selected measurements are performed on the source chosen for the selected CALCulate block. Use the command CALCulate:WMList:STATe to enable the measurements. Refer to Appendix B for the measurement algorithms.

Measurement methods and reference values are specified by CALC:WMParameter commands. Each CALC block has a separate set of parameters.

Table 3–25: Waveform Measurement Definitions

Measurement	Definition
AMPLitude	Amplitude — the difference between HIGH and LOW. Current vertical units.
AREA	Area — the area across the entire acquisition record. Area above ground is positive and area below ground is negative. Current horizontal units-seconds.
CARea	Cycle Area — the area across the first cycle of the acquisition record. Area above ground is positive; area below ground is negative. Current horizontal units-seconds.
CMEan	Cycle Mean — the arithmetic mean of the first cycle in the acquisition record. Current vertical units.
CRMS	Cycle RMS — the root mean square value for the deviation from ground of each point in the first cycle of the acquisition record. Current vertical units.
CROSs	Time at Crossing — the time relative to the trigger point of the specified crossing. The crossing is specified according to its sequence in the waveform record as a Nth crossing, with positive values for N referenced from the start and negative values referenced from the end of the waveform record. Units are seconds.
DELay	Delay — the time between the MidRef crossing on the specified edge of a reference waveform and the MidRef crossing on the specified edge a target waveform. Units are seconds.
FREQuency	Frequency — the reciprocal of the period of the first cycle in the acquisition record. Measured in Hz.
FTIMe	Fall Time — the time between the HREFerence crossing and the LREFerence crossing of the first falling edge of the acquisition record. Units are seconds.
Gain	Gain — the ratio of the amplitude of the target waveform to the amplitude of the reference waveform. Reference waveforms with zero amplitude return an error message. Gain has no units.

www.valuetronics.com

Table 3–25: Waveform Measurement Definitions (Cont.)

Measurement	Definition
HIGH	High — the value used as 100% for calculating the HREFerence, MREFerence, and LREFerence. Current vertical units.
LOW	Low — the value used as 0% for calculating the HREFerence, MREFerence, and LREFerence. Current vertical units.
MAXimum	Maximum — the most positive value in the acquisition record. Current vertical units.
MEAN | DC	Mean or DC — the arithmetic mean of the entire acquisition record. Current vertical units.
MID	Middle — the mid-point between MINimum and MAXimum values. Current vertical units.
MINimum	Minimum — the most negative value in the acquisition record. Current vertical units.
NCROSs	Time at N^{th} Negative Crossing — the time relative to the trigger point of the specified negative crossing. The crossing is specified according to its sequence in the waveform record as a Nth negative crossing, with positive values for N referenced from the start and negative values referenced from the end of the waveform record. Positive crossings are ignored. Units are seconds.
NDUTycycle	Negative Dutycycle — the ratio between NWIDth and the PERiod of the acquisition record. No units.
NWIDth	Negative Width — the width of the first negative pulse in the acquisition record. Measured in seconds.
PCROSs	Time at N^{th} Positive Crossing — the time relative to the trigger point of the specified positive crossing. Specified according to its sequence in the waveform record as a Nth positive crossing, with positive values for N referenced from the start and negative values referenced from the end of the waveform record. Negative crossings are ignored. Units are seconds.
PDUTycycle	Positive Dutycycle — the ratio between PWIDth and the PERiod of the acquisition record. No units.
PERiod	Period — the width of the first cycle in the acquisition record. Measured in seconds.
Phase	Phase — the lead or lag in degrees between the MidRef crossing on the specified edge of a target waveform and the MidRef crossing on the specified edge a reference waveform. Phase is positive when target leads reference waveform; this measurement uses period of the target waveform when computing degrees of phase.
PTPeak	Peak To Peak — the difference between MAXimum and MINimum. Current vertical units.
PWIDth	Positive Width — the width of the first positive pulse in the acquisition record. Measured in seconds.

www.valuetronics.com

Table 3–25: Waveform Measurement Definitions (Cont.)

Measurement	Definition
RMS	RMS — the root mean square value for the deviation from ground of each point in the complete acquisition record. Current vertical units.
RTIMe	Rise Time — the time between the LREFerence crossing and the HREFerence crossing of the first rising edge of the acquisition record. Measured in seconds.
SDEViation | AC	Standard Deviation or AC — the root mean square value for the deviation from the arithmetic mean of each point in the acquisition record. Current vertical units.
TTRig	Trigger-to-trigger time — the time between the trigger event in the main acquistion and the trigger event of the delayed acquisition. Measured in seconds.

Syntax

```
CALCulate<n>:WMList <list>
CALCulate<n>:WMList?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

[1] **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<list>	Query Response
AMPLitude	AMPL
AREA	AREA
CARea	CAR
CMEan	CME
CRMS	CRMS
CROSs	CROSs
FREQuency	FREQ
FTIMe	FTIM
GAIN	GAIN
HIGH	HIGH
LOW	LOW
MAXimum	MAX
MEAN | DC	MEAN
MID	MID
MINimum	MIN

www.valuetronics.com

<list> (Cont.)	Query Response (Cont.)
NCRoss	NCR
NDUTycycle	NDUT
NONE	NONE
NWIDth	NWID
PCRoss	PCR
PERiod	PER
PDUTycycle	PDUT
PHAse	PHA
PTPeak	PTP
PWIDth	PWID
RMS | AC	RMS
RTIMe	RTIM
SDEViation	SDIV
TTRig	TTR

Reset Value MEAN

Errors and Events Execution Error –141, "Invalid character data"
Attempted to program an illegal waveform measurement.

Dependencies None

Examples Command: `CALC1:WML RTIM,FTIM,PWID`

Query: `CALC1:WML?`

Response: `RTIM,FTIM,PWID`

Related Commands CALCulate:WMList:STATe

www.valuetronics.com

CALCulate:WMList:STATe
CALCulate:WMList:STATe?

Sets or queries whether the waveform measurement list for the specified CALC block will execute after the next acquisition.

If you define a calculation with CALC:PATH:EXPRession, the calculations defined by CALCulate:WMList will not execute, regardless of the setting for :WMList:STATE.

Syntax

```
CALCulate<n>:WMList:STATe <boolean>
CALCulate<n>:WMList:STATe?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<boolean>	Query Response
<NRf> 1 or ON 0 or OFF	<NF1> 1 0

Reset Value 0

Errors and Events None

Dependencies None

Examples Command: `CALC1:WML:STAT ON`

Query: `CALC1:WML:STAT?`

Response: 1

Related Commands CALCulate:WMList

www.valuetronics.com

CALCulate:WMParameter:HIGH
CALCulate:WMParameter:HIGH?

Sets or queries the high (most positive) level used for time and amplitude measurements. This setting is used only when CALC:WMP:HMEThod is set to ABSolute. For more information, refer to the discussion of CALC:WMP:HME-Thod. The units are those currently used for vertical magnitude.

Syntax

```
CALCulate<n>:WMParameter:HIGH <high>
CALCulate<n>:WMParameter:HIGH?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<high> (Absolute HIGH level)	Query Response
<NRf>	<NR3>

Reset Value

0.0E+0

Errors and Events

Execution Error –222, “Data out of range”
Attempt to set HIGH to an illegal value.

Dependencies

None

Examples

Command: `CALC1:WMP:HIGH 10.0`

Query: `CALC1:WMP:HIGH?`

Response: `10.0E+0`

Related Commands

CALCulate:WMParameter:HMEThod
CALCulate:WMParameter:LOW

www.valuetronics.com

CALCulate:WMParameter:HMEThod
CALCulate:WMParameter:HMEThod?

Sets or queries the method for calculating the HIGH (most positive) value for time and amplitude waveform measurements.

ABSolute — specifies that HIGH is set to the value of CALCulate:WMParameter:HIGH.

AUTO — selects the MODE method of setting HIGH when the histogram function is able to detect a consistent level above MID. Otherwise, the PEAK method is used. This method is effective when you are not certain what type of waveform to expect.

MODE — selects the level for HIGH based on a peak histogram function which looks for a greater than average number of data points at a level above MID. This method is useful to ignore spurious peaks on a digital logic waveform, such as a TTL clock signal.

PEAK — specifies that HIGH is set to the highest amplitude data point in the acquisition record. This method is useful for a sinewave or triangle waveform.

Syntax

```
CALCulate<n>:WMParameter:HMEThod <method>
CALCulate<n>:WMParameter:HMEThod?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<method> (Select how to set HIGH)	Query Response
ABSolute	ABS
AUTO	AUTO
MODE	MODE
PEAK	PEAK

Reset Value MODE

www.valuetronics.com

Errors and Events Execution Error –141, "Invalid character data"
Attempted to set method to an illegal value.

Dependencies None

Examples Command: `CALC1:WMP:HMET ABS`

Query: `CALC1:WMP:HMET?`

Response: `ABS`

Related Commands CALCulate:WMParameter:LMEThod
CALCulate:WMParameter:HIGH

CALCulate:WMParameter:LOW
CALCulate:WMParameter:LOW?

Sets or queries the low (most negative) level used for time and amplitude measurements. This setting is used only when CALC:WMP:LMEThod is set to ABSolute. For more information, refer to the discussion of CALC:WMP:HME-Thod. The units are those currently used for vertical magnitude.

Syntax

```
CALCulate<n>:WMParameter:LOW <low>
CALCulate<n>:WMParameter:LOW?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

[1] **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<low> (Absolute LOW level)	Query Response
<NRf>	<NR3>

Reset Value 0.0E+0

www.valuetronics.com

Errors and Events Execution Error –222, "Data out of range"
Attempted to set LOW to an illegal value.

Dependencies None

Examples Command: `CALC1:WMP:LOW -10.0`

Query: `CALC1:WMP:LOW?`

Response: `-10.0E+0`

Related Commands CALCulate:WMParameter:HIGH

CALCulate:WMParameter:LMEThod
CALCulate:WMParameter:LMEThod?

Sets or queries the method for calculating the LOW (most negative) value for time and amplitude waveform measurements.

ABSolute — specifies that LOW is set to the value of CALCulate:WMParameter:LOW.

AUTO — selects the MODE method of setting LOW when the histogram function is able to detect a consistent level below MID. Otherwise, the PEAK method is used. The AUTO method is effective when you are not certain what type of waveform to expect.

MODE — selects the level for LOW based on a peak histogram function which looks for a greater than average number of data points at a level below MID. This method is useful to ignore spurious peaks on a digital logic waveform, such as a TTL clock signal.

PEAK — specifies that LOW is set to the lowest amplitude data point in the acquisition record. This method is useful for a sinewave or triangle waveform.

Syntax

```
CALCulate<n>:WMParameter:LMEThod <method>
CALCulate<n>:WMParameter:LMEThod?
```


www.valuetronics.com

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<method> (Select how to set LOW)	Query Response
ABSolute	ABS
AUTO	AUTO
MODE	MODE
PEAK	PEAK

Reset Value MODE

Errors and Events Execution Error –141, "Invalid character data"
Attempted to set method to an illegal value.

Dependencies None

Examples Command: `CALC1:WMP:LMET ABS`

Query: `CALC1:WMP:LMET?`

Response: `ABS`

Related Commands CALCulate:WMParameter:HMEThod
CALCulate:WMParameter:LOW

www.valuetronics.com

CALCulate:WMParameter:HREFerence
CALCulate:WMParameter:HREFerence?

Sets or queries the high reference (distal) level in vertical units for time and amplitude measurements. This setting is used when you have set CALC:WMP:RMEThod to ABSolute.

Syntax

```
CALCulate<n>:WMParameter:HREFerence[:ABSolute] <href>
CALCulate<n>:WMParameter:HREFerence[:ABSolute]?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<href> (Set absolute value of HREF)	Query Response
<NRf>	<NR3>

Reset Value

0.0E+0

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set the high reference to an illegal value.

Dependencies

None

Examples

Command: `CALC1:WMP:HREF 8`

Query: `CALC1:WMP:HREF?`

Response: `8.0E+0`

Related Commands

CALCulate:WMParameter:HREFerence:RELative
CALCulate:WMParameter:RMEThod

www.valuetronics.com

CALCulate:WMParameter:HREFerence:RELative
CALCulate:WMParameter:HREFerence:RELative?

Sets or queries the high reference (distal) level used for time and amplitude measurements. The level can be expressed as a ratio or percent of the current value for CALC:WMP:HIGH. This setting is used when you have set CALC:WMP:RMEThod to RELative.

Syntax

```
CALCulate<n>:WMParameter:HREFerence:RELative <href>
CALCulate<n>:WMParameter:HREFerence:RELative?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<href> (Set relative value of HREF)	Query Response
<NRf> 0.0 ≤ N ≤ 1.0[1] MINimum MAXimum	<NR2> 0.0 1.0

1 **The units must be specified as PCT when setting HREFerence:RELative as a percent of the value HIGH.**

Reset Value 0.9

Errors and Events Execution Error –222, “Data out of range”
Attempted to set the high reference to an illegal value.

Dependencies None

Examples Command: `CALC1:WMP:HREF:REL 95 PCT`

Query: `CALC1:WMP:HREF:REL?`

Response: `0.95`

www.valuetronics.com

Related Commands CALCulate:WMParameter:HREFerence

CALCulate:WMParameter:LREFerence
CALCulate:WMParameter:LREFerence?

Sets or queries the low reference (proximal) level in vertical units for time and amplitude measurements. This setting is used only when CALC:WMP:RME-Thod is set to ABSolute.

Syntax

```
CALCulate<n>:WMParameter:LREFerence[:ABSolute] <lref>
CALCulate<n>:WMParameter:LREFerence[:ABSolute]?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<lref> (Set absolute value of LREF)	Query Response
<NRf>	<NR2>

Reset Value 0.0E+0

Errors and Events Execution Error –222, “Data out of range”
Attempted to set the low reference to an illegal value.

Dependencies None

Examples Command: `CALC1:WMP:LREF -8`

Query: `CALC1:WMP:LREF?`

Response: `-8.0E+0`

Related Commands CALCulate:WMParameter:LREFerence:RELative
CALCulate:WMParameter:RMEThod

www.valuetronics.com

CALCulate:WMParameter:LREFerence:RELative
CALCulate:WMParameter:LREFerence:RELative?

Sets or queries the low reference (proximal) level used for time and amplitude measurements. The level can be expressed as a ratio or percent of the current value for CALC:WMP:LOW. This setting is used only when CALC:WMP:RMEThod is set to RELative.

Syntax

```
CALCulate<n>:WMParameter:LREFerence:RELative <lref>
CALCulate<n>:WMParameter:LREFerence:RELative?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

[1] **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<lref> (Set relative value of LREF)	Query Response
<NRf> 0.0 ≤ N ≤ 1.0[1] MINimum MAXimum	<NR2> 0.0 1.0

[1] **The units must be specified as PCT when setting LREFerence:RELative as a percent of the value LOW.**

Reset Value 0.1

Errors and Events Execution Error –222, “Data out of range”
Attempted to set the low reference to an illegal value.

Dependencies None

Examples Command: `CALC1:WMP:LREF:REL 5 PCT`

Query: `CALC1:WMP:LREF:REL?`

Response: `0.5`

www.valuetronics.com

Related Commands CALCulate:WMParameter:LREFerence
CALCulate:WMParameter:RMEThod

CALCulate:WMParameter:MREFerence
CALCulate:WMParameter:MREFerence?

Sets or queries the middle reference (mesial) level in vertical units for time and amplitude measurements. The MREFerence setting is effective only when CALC:WMP:RMEThod is set to ABSolute.

Syntax

```
CALCulate<n>:WMParameter:MREFerence[:ABSolute] <mref>
CALCulate<n>:WMParameter:MREFerence[:ABSolute]?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<mref> (Set absolute value of MREF)	Query Response
<NRf>	<NR3>

Reset Value 0.0E+0

Errors and Events Execution Error –222, "Data out of range"
Attempted to set the middle reference to an illegal value.

Dependencies None

Examples Command: `CALC1:WMP:MREF 1`

Query: `CALC1:WMP:MREF?`

Response: `1.0E+0`

www.valuetronics.com

Related Commands CALCulate:WMParameter:MREFerence:RELative
CALCulate:WMParameter:RMEThod

CALCulate:WMParameter:MREFerence:HYSTeresis
CALCulate:WMParameter:MREFerence:HYSTeresis?

Sets or queries the middle reference (mesial) hysteresis used for time and amplitude measurements. The hysteresis can be expressed as a ratio or percent of the current value for MREFerence. The signal must transition beyond the vertical range defined by :MREFerence:HYSTeresis before another crossing of :MREFerence can be accepted for measurements such as :PERiod.

This setting is used only when CALC:WMP:RMEThod is set to RELative.

Syntax

```
CALCulate<n>:WMParameter:MREFerence:HYSTeresis <hyst>
CALCulate<n>:WMParameter:MREFerence:HYSTeresis?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<hyst> (Set relative value of HYST)	Query Response
<NRf> 0.0 ≤ N ≤ 0.5[1] MINimum MAXimum	<NR2> 0.0 0.5

1 **The units must be specified as PCT when setting MREFerence:HYSTeresis as a percent of the value MREF.**

Reset Value 0.05

Errors and Events Execution Error –222, “Data out of range”
Attempted to set the middle reference hysteresis to an illegal value.

Dependencies None

www.valuetronics.com

Examples

Command: `CALC1:WMP:MREF:HYST 10 PCT`

Query: `CALC1:WMP:MREF:HYST?`

Response: `0.1`

Related Commands

CALCulate:WMParameter:MREFerence

CALCulate:WMParameter:MREFerence:RELative
CALCulate:WMParameter:MREFerence:RELative?

Sets or queries the middle reference (mesial) level used for time and amplitude measurements. The level can be expressed as a ratio or percent of the current value for HIGH. This setting is used only when CALC:WMP:RMEThod is set to RELative.

Syntax

```
CALCulate<n>:WMParameter:MREFerence:RELative <mref>
CALCulate<n>:WMParameter:MREFerence:RELative?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<mref> (Set relative value of MREF)	Query Response
<NRf> 0.0 ≤ N ≤ 1.0[1] MINimum MAXimum	<NR2> 0.0 1.0

1 **The units must be specified as PCT when setting MREFerence:RELative as a percent of the value HIGH.**

Reset Value

0.5

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set the middle reference to an illegal value.

www.valuetronics.com

Dependencies None

Examples Command: `CALC1:WMP:MREF:REL 55 PCT`

Query: `CALC1:WMP:MREF:REL?`

Response: `0.55`

Related Commands CALCulate:WMParameter:MREFerence
CALCulate:WMParameter:RMEThod

CALCulate:WMParameter:RMEThod
CALCulate:WMParameter:RMEThod?

Sets or queries the method for calculating the reference (high, middle, low) values for time and amplitude measurements. Selecting ABSolute lets you set the measurement parameters at absolute vertical levels. RELative lets you set them as a ratio or percentage of the parameter to the vertical amplitude of the record data.

Syntax

```
CALCulate<n>:WMParameter:RMEThod <method>
CALCulate<n>:WMParameter:RMEThod?
```

Parameters

<n>[1]	Query Response
1 2 3 4	NA

1 **The calculate block number <n> selects one of four calculate blocks. If you omit <n>, the default is calculate block 1.**

<method> (Sets reference method)	Query Response
ABSolute	ABS
RELative	REL

Reset Value REL

Errors and Events Execution Error –222, "Data out of range"
Attempted to set method to an illegal value.

www.valuetronics.com

Dependencies None

Examples Command: `CALC1:WMP:RMET ABS`

Query: `CALC1:WMP:RMET?`

Response: `ABS`

Related Commands CALCulate:WMParameter:HREFerence
CALCulate:WMParameter:LREFerence
CALCulate:WMParameter:MREFerence

www.valuetronics.com

CALibration Subsystem

This section describes the commands in the CALibration subsystem. These commands run the waveform analyzer self-calibration functions.

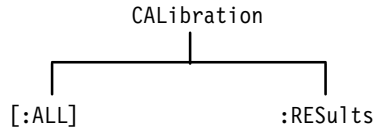


Figure 3–13: CALibration Subsystem Hierarchy

CALibration
CALibration?

Executes all self-calibration functions. If a failure occurs, calibration will immediately stop. The query form runs the self-calibration functions and returns the numeric identifier of the first failed function. A value of zero indicates there were no failures. The command form executes the functions but returns no results code.

Calibration functions have numeric identifiers in the range 2000 to 2999. CALibration sets the CAL_OPC pending flag. Use the command *WAI, *OPC, or *OPC? to synchronize the response with completion of self calibration.

To calibrate probes, use the command OUTPut:PCOMpensate which enables the probe calibration signal to the front panel.

Syntax

```
CALibration[:ALL]
CALibration[:ALL]?
```

Parameters

<boolean>	Query Response
Not applicable	<NR1> 0 No failures 2000 ≤ N ≤ 2999

Reset Value Not applicable

Errors and Events None

www.valuetronics.com

Dependencies None

Examples Command: `CAL`

Query: `CAL?`

Response: `0`

Related Commands CALibration:RESults?
CALibration:RESults:VERBose?

CALibration:RESults? (Query Only)

Returns the results code for the last calibration performed. The result code is the numeric identifier for the first function to fail. A value of zero is returned when there are no failures and a value of –1 indicates a calibration is in progress.

Calibration functions have numeric identifiers in the range 2000 to 2999.

Because this query is not synchronized with instrument operations, you may want to use *WAI, *OPC, or *OPC? to wait for the CAL_OPC pending flag to clear before sending the CAL:RES? query.

Syntax `CALibration:RESults[:CODE]?`

Parameters

	Query Response
Not applicable	<NR1> –1 Calibration in progress 0 No failures 2000 ≤ N ≤ 2999

Reset Value Not applicable

Errors and Events None

Dependencies None

Examples Query: `CAL:RES?`

Response: `0`

www.valuetronics.com

Related Commands CALibration
CALibration:RESults:VERBose?

CALibration:RESults:VERBose? (Query Only)

Returns an ASCII string describing the results of the last calibration performed. The returned string will include the numeric identifier for the first function that failed. When there are no failures, the query returns a zero value and a description of the last executed function. When a failure occurs, the query returns the number of the first failed function followed by detailed results information. If the query returns a value of –1, calibration is in progress.

Calibration functions have numeric identifiers in the range 2000 to 2999.

Because this query is not synchronized with instrument operations so you may want to use *WAI, *OPC, or *OPC? to wait for the CAL_OPC pending flag to clear before sending the CAL:RES? query.

Syntax

```
CALibration:RESults:VERBose?
```

Parameters

	Query Response
Not applicable	<NR1>,<string> <NR1>:results code –1 (Calibration in progress) 0 (No failures) 2000 ≤ N ≤ 2999 <string>:verbose results (error specific)

Reset Value Not applicable

Errors and Events None

Dependencies None

Examples Query: `CAL:RES:VERB?`

Response: `"2001,CAL_1 max=444 min=222 . . ."`

www.valuetronics.com

Related Commands CALibration[:ALL]
CALibration:RESults?

www.valuetronics.com

FORMat Subsystem

This section describes the commands in the FORMat subsystem. They are used to set the format of acquisition record data and measurement data transferred out of the waveform analyzer. The primary data transfer commands are CALCulate:DATA?, [SENse:]DATA?, and TRACe[:DATA]?.

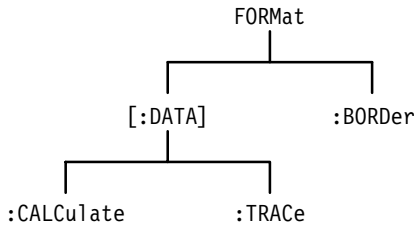


Figure 3–14: FORMat Subsystem Hierarchy

FORMat
FORMat?

Sets or queries the type of encoding used to transfer acquisition records acquired with the SENSe subsystem. The format applies to transfer through most external interfaces. FDC transfers are always in binary format.

The number of significant digits returned by an ASCII data query is not selectable. A setting of zero selects a format with a varying number of significant digits depending on the origin of the data. Refer to page 2–55 for a definition of the data formats for <type>.

Syntax

```
FORMat[:DATA] <type>[,<length>]
FORMat[:DATA]?
```

Parameters

<type>[,<length>]	Query Response
ASCii[,0]	ASC,0
INTeger[,16]	INT,16

Reset Value ASC,0

www.valuetronics.com

Errors and Events Execution Error –141, "Invalid character data"
Attempted to set format to an illegal value.

Dependencies None

Examples Command: `FORM INT`

Query: `FORM?`

Response: `INT,16`

Related Commands [SENSe:]DATA?
TRACe[:DATA]?

FORMat:CALCulate
FORMat:CALCulate?

Sets or queries the type of data format used to transfer data produced by a CALC block. The format applies to transfer through any external interface. The REAL type corresponds to a 32 bit Float type number in the C programming language.

The number of significant digits returned by an ASCII data query is not selectable. A setting of zero selects a format with a varying number of significant digits depending on the origin of the data. Refer to page 2–55 for a definition of the data formats for <type>.

Syntax

```
FORMat[:DATA]:CALCulate[n] <type>[,<length>]
FORMat[:DATA]:CALCulate[n]?
```

Parameters

<type>[,<length>]	Query Response
ASCii[,0]	ASC,0
REAL,32	REAL,32

Reset Value ASC,0

Errors and Events Execution Error –224, "Illegal parameter value"
Attempted to set format to an illegal value.

Dependencies None

www.valuetronics.com

Examples

Command: `FORM:CALC2 REAL,32`

Query: `FORM:CALC2?`

Response: `REAL,32`

Related Commands

CALCulate:DATA?
TRACe[:DATA]?

FORMat:TRACe:AATS
FORMat:TRACe:AATS?

Sets or queries the type of data format used to transfer the timestamp trace produced by Auto Advance acquisition. The format applies to transfer through any external interface. The REAL type corresponds to a Float type number in the C programming language.

The number of significant digits returned by an ASCII data query is not selectable. A setting of zero selects a format with a varying number of significant digits depending on the value of the data point. Refer to page 2–55 for a definition of the data formats for <type>.

Syntax

```
FORMat[:DATA]:TRACe:AATS <type>[,<length>]
FORMat[:DATA]:TRACe:AATS?
```

Parameters

<type>[,<length>]	Query Response
ASCii[,0]	ASC,0
REAL,32	REAL,32

Reset Value

ASC,0

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to set format to an illegal value.

Dependencies

None

Examples

Command: `FORM:TRAC:AATS REAL,32`

Query: `FORM:TRAC:AATS?`

Response: `REAL,32`

www.valuetronics.com

Related Commands TRACe[:DATA]?

FORMat:BORDer
FORMat:BORDer?

Sets or queries the byte order used to transfer binary data through all external interfaces. The FORMat selections REAL and INTeger produce binary data. This command has no effect when FORMat is set to ASCii.

Syntax

```
FORMat:BORDer <order>
FORMat:BORDer?
```

Parameters

<order>		Query Response
NORMal	(high byte–low byte)	NORM
SWAPped	(low byte–high byte)	SWAP

Reset Value NORM

Errors and Events Execution Error –141, “Invalid character data”
Attempted to set byte order to an illegal value.

Dependencies None

Examples Command: `FORM:BORD SWAP`

Query: `FORM:BORD?`

Response: `SWAP`

Related Commands [SENSe:]DATA?
CALCulate:DATA?
TRACe[:DATA]?

www.valuetronics.com

FUNCtion and DATA Subsystems

This section describes the commands in the [SENSe:]FUNCtion and [SENSe:]DATA subsystems. These commands control sense functions and provide a means of accessing the data produced by the sense functions.

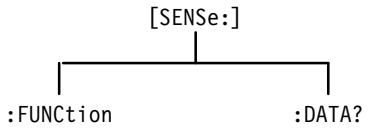


Figure 3–15: FUNCtion and DATA Hierarchy

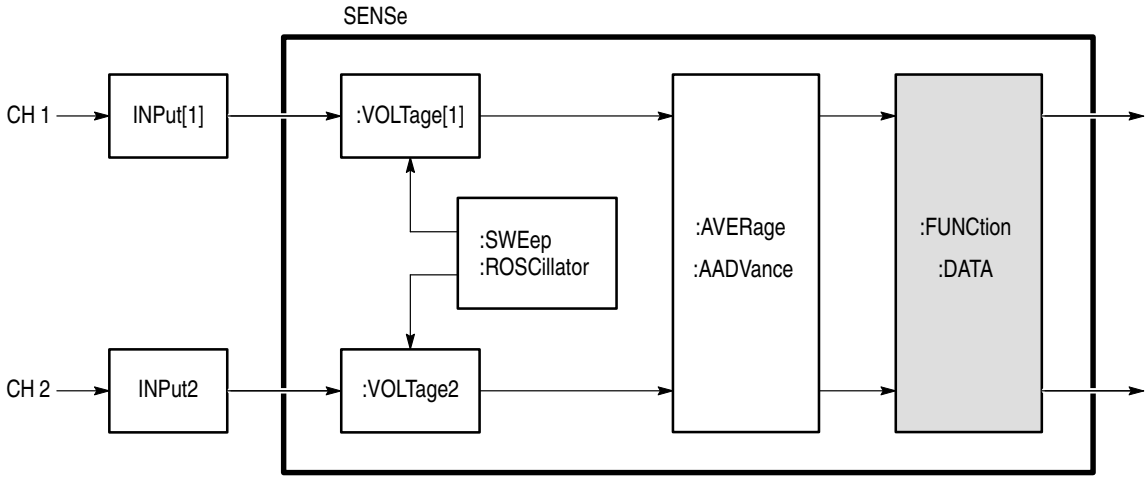


Figure 3–16: FUNCtion and DATA Functional Model

www.valuetronics.com

FUNCtion[:ON]
FUNCtion[:ON]?

Sets or queries which sense functions are enabled. XTIM:VOLT n functions acquire data from the input channel <n>. You can enable only one function at a time when FUNCtion:CONCurrent is OFF. Set FUNCtion:CONCurrent to ON if you want to enable more functions.

Channel names and function strings may be used interchangeably for parameters. This is not standard SCPI practice and does not provide horizontal compatibility.

The query FUNCtion[:ON]? always returns a comma-separated list of active sense functions. The functions are returned as a list of ASCII strings ordered by ascending channel number.

Syntax

```
[SENSe:]FUNCtion[:ON] <function>{,<function>}
[SENSe:]FUNCtion[:ON]?
```

Parameters

<function>	Query Response
	<string>
"XTIMe:VOLTage[:DC] 1" or CHAN1	"XTIM:VOLT 1"
"XTIMe:VOLTage[:DC] 2" or CHAN2	"XTIM:VOLT 2"
"XTIMe:VOLTage[:DC] 3" or CHAN3	"XTIM:VOLT 3"
"XTIMe:VOLTage[:DC] 4" or CHAN4	"XTIM:VOLT 4"

Reset Value

" " (null string)

Errors and Events

Execution Error –224, "Illegal parameter value"
Attempted to enable an invalid sense function.

Execution Error –241, "Hardware missing"
Attempted to enable "XTIM:VOLT 3" or "XTIM:VOLT 4" when the instrument is configured with two channels.

Execution Error –221, "Settings conflict."
Attempted to enable multiple sense functions when FUNCtion:CONCurrent is OFF.

Dependencies

Enabling a function when FUNCtion:CONCurrent is OFF will turn off the currently enabled function.

This command determines the number of records returned by "DATA?".

Examples

Command: `FUNC "XTIM:VOLT 1"`

www.valuetronics.com

Query: `FUNC?`

Response: `"XTIM:VOLT 1"`

Related Commands

FUNCtion:OFF
FUNCtion:CONCurrent
FUNCtion[:ON]:ALL
DATA?

FUNCtion[:ON]:ALL

Enables all sense functions. XTIM:VOLT n functions acquire data from the input channel <n>. You must first set FUNCtion:CONCurrent to ON before sending FUNCtion:ALL to enable all functions. To enable individual functions, use FUNCtion:[:ON].

Syntax

`[SENSe:]FUNCtion[:ON]:ALL`

Data Types

None

Reset Value

Not applicable

Errors and Events

Execution Error –221, “Settings conflict.”
Attempted to execute this command when FUNCtion:CONCurrent is OFF.

Dependencies

This command determines the number of records returned by “DATA?”.

Examples

Command: `FUNC:ALL`

Related Commands

FUNCtion:OFF:ALL
FUNCtion:CONCurrent

www.valuetronics.com

FUNCtion[:ON]:COUNt? (Query Only)

Returns the number of enabled sense functions. This command is useful when FUNCtion:CONCurrent is ON because it tells you how many responses to expect when sending a FUNC[:ON]? or DATA? query.

Syntax

```
[SENSe:]FUNCtion[:ON]:COUNt?
```

Parameters

<count>	Query Response
Not Applicable	<NR1>

Reset Value

0

Errors and Events

None

Dependencies

None

Examples

Query: `FUNC:COUN?`

Response: 1

Related Commands

FUNCtion:OFF:COUNt?
FUNCTion:CONCurrent

www.valuetronics.com

FUNCtion:OFF
FUNCtion:OFF?

Sets or queries which sense functions are disabled. Only active functions provide data when you send the query DATA?. To disable all functions at once, use FUNCtion:OFF:ALL.

The query FUNCtion:OFF? always returns a comma-separated list of the disabled sense functions. The functions are returned as a list of ASCII strings ordered by ascending channel number.

Channel names and function strings may be used interchangeably for parameters. This is not standard SCPI practice and does not provide horizontal compatibility.

Syntax

```
[SENSe:]FUNCtion:OFF <function>{,<function>}
[SENSe:]FUNCtion:OFF?
```

Parameters

<function>	Query Response
	<string>
"XTIMe:VOLTage[:DC] 1" or CHAN1	"XTIM:VOLT 1"
"XTIMe:VOLTage[:DC] 2" or CHAN2	"XTIM:VOLT 2"
"XTIMe:VOLTage[:DC] 3" or CHAN3	"XTIM:VOLT 3"
"XTIMe:VOLTage[:DC] 4" or CHAN4	"XTIM:VOLT 4"

Reset Value

All functions

Errors and Events

Execution Error –224, "Illegal parameter value"
Attempted to disable an invalid sense function.

Execution Error –221, "Settings conflict."
Attempted to disable a function when FUNCtion:CONCurrent is OFF.

Dependencies

This command determines the number of acquisition records returned by the query DATA?.

Examples

Command: `FUNC:OFF "XTIM:VOLT 1"`

Query: `FUNC:OFF?`

Response: `"XTIM:VOLT 1"`

www.valuetronics.com

Related Commands

FUNCtion[:ON]
FUNCtion:OFF:ALL
FUNCtion:CONCurrent

FUNCtion:OFF:ALL

Disables all sense functions at once without the side effects of sending the reset command *RST. Only active functions provide data when you send the query DATA?. To disable a single function, use FUNCtion:OFF.

Syntax

```
[SENSe:]FUNCtion:OFF:ALL
```

Data Types

None

Reset Value

Not applicable

Errors and Events

Execution Error –221, “Settings conflict.”
Attempted to execute this command when FUNCtion:CONCurrent is OFF.

Dependencies

This command determines the number of records returned by the query DATA?.

Examples

Command: `FUNC:OFF:ALL`

Related Commands

FUNCtion[:ON]:ALL

www.valuetronics.com

FUNCtion:OFF:COUNt? (Query Only)

Returns the number of disabled sense functions. Use FUNCtion[:ON] to enable functions.

Syntax

```
[SENSe:]FUNCtion:OFF:COUNt?
```

Parameters

<count>	Query Response
Not Applicable	<NR1>

Reset Value

4

Errors and Events

None

Dependencies

None

Examples

Query: `FUNC:OFF:COUN?`

Response: 1

Related Commands

FUNCtion[:ON]:COUNt?

www.valuetronics.com

FUNCtion:CONCurrent
FUNCtion:CONCurrent?

Sets or queries whether more than one sense function can be enabled at a time. When you set FUNCtion:CONCurrent from OFF to ON, the state of individual functions does not change. All disabled functions remain off and if one function was enabled, it remains on. While FUNCtion:CONCurrent is ON, you can enable any and all provided functions.

When FUNCtion:CONCurrent is OFF, and you define a new function with FUNCtion[:ON], the specified function is enabled and all others are disabled.

Syntax

```
[SENSe:]FUNCtion:CONCurrent <boolean>
[SENSe:]FUNCtion:CONCurrent?
```

Parameters

<boolean>	Query Response
<NRf>	<NR1>
1 or ON	1
0 or OFF	0

Reset Value

1

Errors and Events

None

Dependencies

Setting FUNCtion:CONCurrent from ON to OFF will enable XTIM:VOLT 1 and disable all other functions.

Examples

Command: `FUNC:CONC ON`

Query: `FUNC:CONC?`

Response: 1

Related Commands

FUNCtion[:ON]
FUNCtion:OFF

www.valuetronics.com

FUNCtion:STATe
FUNCtion:STATe?

Sets or queries the state of a specified sense function. It operates like the FUNCtion[:ON] and FUNCtion:OFF commands which allow you to enable or disable a function.

Channel names and function strings may be used interchangeably for parameters. This is not standard SCPI practice and does not provide horizontal compatibility.

Syntax

```
[SENSe:]FUNCtion:STATe <function>, <boolean>
[SENSe:]FUNCtion:STATe? <function>
```

Parameters

<function>	Query Response
"XTIMe:VOLTage[:DC] 1" or CHAN1 "XTIMe:VOLTage[:DC] 2" or CHAN2 "XTIMe:VOLTage[:DC] 3" or CHAN3 "XTIMe:VOLTage[:DC] 4" or CHAN4	Not Applicable

<boolean>	Query Response
<NRf> 1 or ON 0 or OFF	<NR1> 1 0

Reset Value

0

Errors and Events

Execution Error –224, "Illegal parameter value"
Attempted to to enable or disable an invalid SENSe function.

Execution Error –241, "Hardware missing"
Attempted to enableor disable "XTIM:VOLT 3" or "XTIM:VOLT 4" when the instrument is configured with two channels.

Dependencies

This command determines the number of records returned by the query DATA?.

Examples

Command: `FUNC:STAT "XTIM:VOLT 1", ON`

Query: `FUNC:STAT? "XTIM:VOLT 1"`

Response: 1

www.valuetronics.com

Related Commands

FUNCtion[:ON]
FUNCtion:OFF
FUNCtion:CONCurrent

DATA? (Query Only)

This query returns the results for the specified function or for all sense functions that are enabled. If auto-advance acquisition is on, then DATA? returns all acquisition records, beginning with AADV:RECord:STARt, for the specified channel. The format of the data is determined by the FORMat subsystem.

When you specify a sense function, DATA? returns only the results of the specified function. When no sense function is specified, DATA? returns the results of all sense functions that are enabled. The data is returned in ascending channel order. You can access individual auto-advance records with commands from the TRACe and CALCulate subsystems.

DATA? does not return data until pending acquisitions or calculations are complete. You can use the commands *WAI, *OPC, or *OPC? to synchronize the response in a different manner.

Channel names and function strings may be used interchangeably for parameters. This is not standard SCPI practice and does not provide horizontal compatibility.

Syntax

```
[SENSe:]DATA? [<function>]
```

Parameters

<function>	Query Response
"XTIMe:VOLTage[:DC] 1" or CHAN1 "XTIMe:VOLTage[:DC] 2" or CHAN2 "XTIMe:VOLTage[:DC] 3" or CHAN3 "XTIMe:VOLTage[:DC] 4" or CHAN4	Not Applicable

<data>	Query Response
Not Applicable	Defined by FORMat[:DATA]

Reset Value

Not applicable

Errors and Events

Execution Error –224, "Illegal parameter value"
Attempted to query data from an invalid sense function.

Execution Error –230, "Data corrupt or stale."
Attempted to query data that is invalid, incomplete or stale.

www.valuetronics.com

Execution Error –241, “Hardware missing”
Attempted to query data from “XTIM:VOLT 3” or “XTIM:VOLT 4” when the instrument is configured with two channels.

Dependencies

The input channel number is used by the SENSe and TRACe subsystems
For example, the command DATA? “XTIM:VOLT 1” is equivalent to TRAC[:DATA]? CHAN1 and both refer to the input channel 1.

Examples

Query: `DATA?`

Response: `<data>`

Related Commands

FUNCtion[:ON]
FUNCtion:CONCurrent
FORMat[:DATA]
DATA:PREamble?

DATA:PREamble? (Query Only)

This query returns the data preamble for the specified function or for all enabled sense functions. If Auto Advance is on, then one preamble for each enabled channel is returned.

When you specify a sense function, DATA:PREamble? returns only the preamble of the specified function. When you do not specify a sense function is specified, DATA:PREamble? returns the preambles of all enabled sense functions. The preambles are returned in ascending channel order.

DATA:PREamble? does not return data until pending acquisitions or calculations are complete. You can use the commands *WAI, *OPC, or *OPC? to synchronize the response in a different manner.

Channel names and function strings may be used interchangeably for parameters. This is not standard SCPI practice and does not provide horizontal compatibility.

Syntax

`[SENSe:]DATA:PREamble? [<function>]`

Parameters

<function>	Query Response
“XTIMe:VOLTage[:DC] 1” or CHAN1 “XTIMe:VOLTage[:DC] 2” or CHAN2 “XTIMe:VOLTage[:DC] 3” or CHAN3 “XTIMe:VOLTage[:DC] 4” or CHAN4	Not Applicable

www.valuetronics.com

<preamble>	Query Response
Not applicable	DIF expression

Reset Value

Not applicable

Errors and Events

Execution Error –224, “Illegal parameter value”
Attempted to query data from an invalid sense function.

Execution Error –230, “Data corrupt or stale.”
Attempted to query data that is invalid, incomplete or stale.

Execution Error –241, “Hardware missing”
Attempted to query data from “XTIM:VOLT 3” or “XTIM:VOLT 4” when the instrument is configured with two channels.

Dependencies

The input channel number is used by the SENSe and TRACe subsystems For example, the command DATA? “XTIM:VOLT 1” is equivalent to TRAC[:DATA]? CHAN1 and both refer to the input CH 1.

Examples

Query:

```
DATA:PRE?
```

Response:

```
DIF(VERS 1995.0 SCOP PRE)
IDEN(NAME "CHAN1" INST(NAME "TVS645" ID "B010100"))
ENC(FORM ASC NVAL -32768 ORAN 32767 URAN -32767)
DIM=X(TYPE IMPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT
"S")
DIM=Y(TYPE EXPL SCAL <nr3> OFFS <nr3> SIZE <nr1> UNIT
"V")
DATA(CURV(CTYP NONE))
```

Related Commands

FUNCtion[:ON]
FUNCtion:CONCurrent
FORMat[:DATA]
DATA?

www.valuetronics.com

INITiate and ABORt Subsystems

This section describes the commands in the INITiate and ABORt subsystems. These commands operate with the ARM and TRIGger subsystems to start signal acquisition.

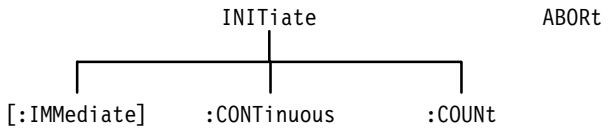


Figure 3–17: INITiate and ABORt Subsystem Hierarchy

INITiate

Starts the acquisition and measurement systems. INITiate causes the arm/trigger state-machine to exit the idle state and be ready for any defined ARM or TRIGger condition. This command is essential to start waveform analyzer acquisitions and measurements.

INITiate sets the ACQ_OPC pending flag. You may want to use *WAI, *OPC, or *OPC? to synchronize the command.

Syntax `INITiate[:IMMediate]`

Data Types None

Reset Value Not applicable

Errors and Events Execution Error –213, “Init ignored”
Sent INITiate when the instrument was not in the idle state.

Dependencies None

Examples Command: `INIT`

Related Commands ABORt

www.valuetronics.com

INITiate:CONTinuous
INITiate:CONTinuous?

Sets or queries whether the acquisition loop repeats continuously, including any defined arm or trigger conditions. Use this command only when FDC is employed to transfer data. INITiate:CONT may be difficult to interrupt once initiated.

***NOTE**. Your waveform analyzer may not respond after receiving INIT:CONT. To end acquisition and return control, send the VXIbus word serial command <Abort Normal Operation> then INIT:CONT OFF.*

When INIT:CONTinuous is set to ON and you send a self-synchronizing command, such as a TRACe? query, the parser does not execute subsequent commands until the current command (INIT:CONT) completes and an OPC occurs. OPC cannot occur until INIT:CONT terminates. When a defined trigger event or arm event does not occur, INIT:CONT does not terminate. Unless you have sent a data query or other self-synchronizing command, you can recover control by sending the *RST command.

When you send INIT:CONT, acquisition continues until the instrument receives an *RST, or VXI word serial <Abort Normal Operation> or <End Normal Operation> command and INIT:CONT OFF. The command *RST is effective only when you have not sent a self-synchronizing command, such as TRACe?. You can use the command ABORt to stop the current acquisition and automatically restart continuous acquisition.

The setting for ARM:COUNt is ignored when INIT:CONTinuous is set to ON.

Syntax

```
INITiate:CONTinuous <boolean>
INITiate:CONTinuous?
```

Parameters

<boolean>	Query Response
<NRf> 1 or ON 0 or OFF	<NR1> 1 0

Reset Value 0

Errors and Events None

www.valuetronics.com

Dependencies Setting INIT:CONTinuous to ON performs an implicit INITiate[:IMMediate] and, if the Fast Data Channel is active, sends a continuous stream of measurement results.

Examples Command: `INIT:CONT ON`

Query: `INIT:CONT?`

Response: 1

Related Commands INITiate
INITiate:COUNt
ABORt

INITiate:COUNt
INITiate:COUNt?

Sets or queries the number of times to repeat the arm/trigger acquisition loop. This command is most useful when the FDC is employed to transfer data.

For acquisition looping using FDC, you might set COUNT to a value of 10. The instrument cycles through ten arm/trigger loops, acquiring signals from all active channels during each pass through the loop. The specified waveform records and measurements are transferred at the end of each cycle. Upon completion of the tenth loop the arm/trigger state machine returns to the idle state. Without FDC, only the waveform record acquired on the last cycle and measurements on it are available.

If you have defined a Fast Data Channel, setting COUNt to a value other than one may cause the waveform analyzer to send multiple measurement results, one for each acquisition.

Syntax

```
INITiate:COUNt <count>
INITiate:COUNt?
```

Parameters

<count>	Query Response
<NRf> 1 ≤ N ≤ 10000 MINimum MAXimum	<NR1> 1 10000

Reset Value 1

www.valuetronics.com

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set count to an illegal value.

Dependencies

When INIT:CONTinuous is set to ON, INITiate:COUNt is ignored.

Examples

Command: `INIT:COUN 10`

Query: `INIT:COUN?`

Response: `10`

Related Commands

INITiate
INITiate:CONTinuous
ABORt

www.valuetronics.com

ABORt

Stops all acquisition and measurements and returns the arm/trigger subsystem to the idle state. Any remaining data is corrupt. When INITiate:CONTinuous is set to ON, the current acquisition is terminated and acquisition is restarted.

Syntax `ABORt`

Data Types None

Reset Value Not applicable

Errors and Events None

Dependencies Execution Error –230, "Data corrupt or stale."
Sent the command DATA? after sending ABORT.

Examples Command: `ABOR`

Related Commands INITiate

www.valuetronics.com

INPut Subsystem

This section describes each command and query in the INPut subsystem. The INPut subsystem controls the parameters shown in Figure 3–18. Figure 3–19 shows the part of the waveform analyzer controlled by the INPut commands. The input channel number, defined at the probe connector, is shared as the parameter <n> for the INPut<n> and VOLTage<n> commands. Only the four channel TVS641 accepts INPut3 and INPut4 commands.

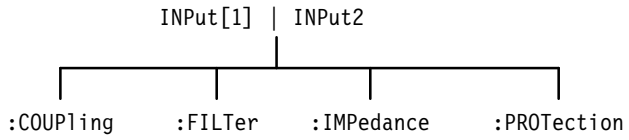


Figure 3–18: INPut Subsystem Hierarchy

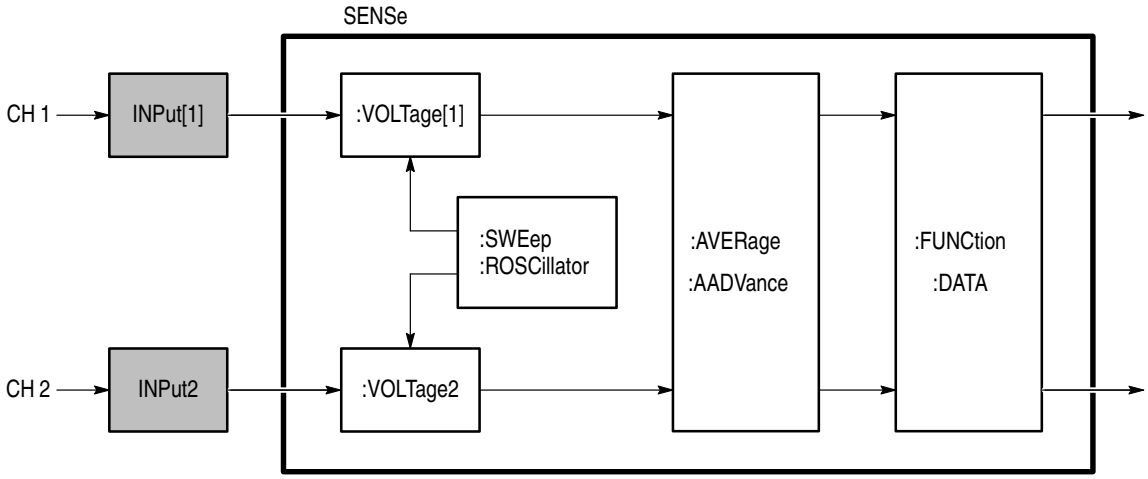


Figure 3–19: INPut Subsystem Functional Model

www.valuetronics.com

INPut:COUPling
INPut:COUPling?

Sets or queries the type of signal coupling for the specified input channel. When AC coupled and INPut:IMPedance is set to 1 MΩ, frequencies below 10 Hz are attenuated. When AC coupled and INPut:IMPedance is set to 50 Ω, frequencies below 200 kHz are attenuated. Setting COUPling to GROund connects the input of the amplifier to ground and presents a high impedance to the incoming signal.

Syntax

```
INPut<n>:COUPling <coupling>
INPut<n>:COUPling?
```

Parameters

<n> (Channel Number)[1]	Query Response
1 2 3 4	NA

1 **The input channel number <n> is used by the commands INPut<n> and VOLTage<n>. If you omit <n>, the default is channel 1.**

<coupling>	Query Response
AC	AC
DC	DC
GROund	GRO

Reset Value DC

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to set coupling to an illegal value.

Execution Error –241, “Hardware missing”
Attempted to program INPut3 or INPut4 on an instrument configured with only two channels.

Dependencies None

Examples

Command: `INP1:COUP AC`

Query: `INP1:COUP?`

Response: `AC`

www.valuetronics.com

Related Commands None

INPut:FILTer
INPut:FILTer?

Sets or queries whether the low-pass analog noise filter is on or off for the specified input channel. INPut:FILTer must be off to use the full bandwidth of the waveform analyzer. To set the limit frequency of the low pass filter, use the command INPut:FILTer:FREQuency.

Syntax

```
INPut<n>:FILTer[:LPASs][:STATe] <boolean>
INPut<n>:FILTer[:LPASs][:STATe]?
```

Parameters

<n> (Channel Number)[1]	Query Response
1 2 3 4	NA

1 **The input channel number <n> is used by the commands INPut<n> and VOLTage<n>. If you omit <n>, the default is channel 1.**

<boolean>	Query Response
<NRf> 1 or ON 0 or OFF	<NF1> 1 0

Reset Value 0 (OFF)

Errors and Events Execution Error –241, "Hardware missing"
Attempted to program INPut3 or INPut4 on an instrument configured with only two channels.

Dependencies None

Examples Command: `INP1:FILT ON`

Query: `INP1:FILT?`

Response: `1`

www.valuetronics.com

Related Commands INPut:FILTer:FREQuency

INPut:FILTer:FREQuency
INPut:FILTer:FREQuency?

Sets or queries the frequency limit of the low-pass analog noise filter for the specified input channel. You must enable the low pass filter with the command INPut:FILTer before the filter is effective. The filter selections are 20 MHz and 250 MHz. To use the full bandwidth of the waveform analyzer set INPut:FILTer to OFF.

Syntax

```
INPut<n>:FILTer[:LPASs]:FREQuency <frequency>
INPut<n>:FILTer[:LPASs]:FREQuency?
```

Parameters

<n> (Channel Number)[1]	Query Response
1 2 3 4	NA

1 **The input channel number <n> is used by the commands INPut<n> and VOLTage<n>. If you omit <n>, the default is channel 1.**

<frequency> (Limit of Low Pass Filter) [1]	Query Response
<NRf> 20E6 250E6[2] MINimum MAXimum	<NF3> 20.0E+6 250.0E+6 20.0E+6 250.0E+6

1 **The default multiplier for the <frequency> parameter is HZ. You can also use KHZ or MHZ.**

2 **When using the 250 MHz filter on the TVS621 or TVS641 input channels, the effective system bandwidth is typically 180 MHz.**

Reset Value 250.0E+6

Errors and Events Execution Error –222, “Data out of range”
Attempted to set low pass frequency to an illegal value.

www.valuetronics.com

Execution Error –241, “Hardware missing”
Attempted to program INPut3 or INPut4 on an instrument configured with only two channels.

Dependencies None

Examples Command: `INP1:FILT:FREQ 20E6`

Query: `INP1:FILT:FREQ?`

Response: `20.0E+6`

Related Commands INPut:FILTer

INPut:IMPedance
INPut:IMPedance?

Sets or queries the input impedance of the specified input channel. The impedance selections are 50 Ω and 1 MΩ. When input protection is enabled, an overload of the 50 Ω input automatically sets impedance to 1 MΩ.

Syntax

```
INPut<n>:IMPedance <impedance>
INPut<n>:IMPedance?
```

Parameters

<n> (Channel Number)[1]	Query Response
1 2 3 4	NA

1 **The input channel number <n> is used by the commands INPut<n> and VOLTage<n>. If you omit <n>, the default is channel 1.**

<impedance> (Channel impedance)[1]	Query Response
<NRf> 50 1E6 MINimum MAXimum	<NF3> 50.0E+0 1.0E+6 50.0E+0 1.0E+6

1 **The default multiplier for the <impedance> parameter is OHM. You can also use KOHM or MOHM.**

www.valuetronics.com

Reset Value 1.0E+6

Errors and Events Execution Error –222, "Data out of range"
Attempted to set impedance to an illegal value.

Execution Error –241, "Hardware missing"
Attempted to program INPut3 or INPut4 on an instrument configured with only two channels.

Dependencies None

Examples Command: `INP1:IMP 1E6`

Response: `INP1:IMP?`

Query: `1.0E+6`

Related Commands INPut:PROTection:STATe

INPut:PROTection:STATe
INPut:PROTection:STATe?

Sets or queries the state of the input protection circuitry for *all* input channels. When protection is enabled (ON), an input overload will automatically set the input channel impedance to 1 MΩ in order to protect the input.

Because the waveform analyzer stores the INPut:PROTection:STATe in nonvolatile RAM, this setting is recalled at power-up. However, it is not saved with a stored settings, and it is not changed by the reset command *RST. The factory default is to enable input protection.

CAUTION. *Use caution when disabling input protection. Damage may occur to the instrument input channels if the overload condition persists or is extreme.*

Syntax

```
INPut:PROTection:STATe <boolean>
INPut:PROTection:STATe?
```


www.valuetronics.com

Parameters

<boolean>	Query Response
<NRf> 1 or ON 0 or OFF	<NF1> 1 0

Reset Value

Not Applicable

Errors and Events

Execution Error –203, "Command protected"
Attempted to change the protection without first removing the restriction with the SYSTem:PROTect command.

Device Specific Error –310, "System error"
An input overload generates a device-specific error whether or not input protection is enabled.

Dependencies

Although other INPut commands separately control each channel, input protection for all channels is enabled or disabled at the same time. Disabling the protection for one channel automatically disables the protection for all channels.

Examples

Command: `INP:PROT:STAT ON`

Query: `INP:PROT:STAT?`

Response: 1

Related Commands

INPut:IMPedance
SYStem:PROTect

www.valuetronics.com

MEMory Subsystem

This section describes the commands in the MEMory subsystem. These commands store and retrieves instrument settings. The waveform analyzer can store the current instrument state in any of ten nonvolatile memory locations or in a file on your controller.

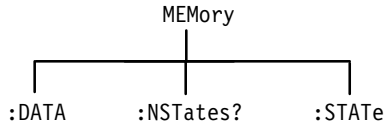


Figure 3–20: MEMory Subsystem Hierarchy

MEMory:DATA
MEMory:DATA?

Sets or queries the instrument settings (or state) for the ten on-board nonvolatile memory locations. With the query form, you can save the current settings (SAV0) or one or more of the on board settings (SAV1–10) to a file on your controller. Later, you can load these settings into the on board storage locations to run a suite of measurements based on the different instrument configurations. Use the command *RCL to use an on board instrument setting.

You cannot load a settings file into SAV0 because it is the current or active instrument settings. You can use the command SYSTem:SET to modify the current settings with a settings file.

Settings data is transferred in binary format, designated <arbitrary_block_data>, that is unique to the waveform analyzer.

Syntax

```
MEMory:DATA <setting>, <data>
MEMory:DATA? <setting>
```

Parameters

<source>	Query Response
SAV0 SAV1 SAV2 . . SAV10	Not applicable

www.valuetronics.com

<data>	Query Response
<arbitrary_block_data>	<arbitrary_block_data>

Reset Value

Not applicable

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to set or query an invalid setting number.

Execution Error –233, “Invalid version”
Attempted to load a block of binary settings that have a different version number than the instrument.

Dependencies

None

Examples

Command: `MEM:DATA STAT1,<arb_block_data>`

Query: `MEM:DATA? STAT1`

Response: `<arb_block_data>`

Related Commands

MEMory:DATA
MEMory:NSTates?
MEMory:STATe:CATalog?
MEMory:STATe:DEFine?
SYSTem:SET
*RCL
*SAV

www.valuetronics.com

MEMory:NSTates? (Query Only)

Returns the number of instrument settings (states) that can be stored in the waveform analyzer. Settings locations can be used or filled with the command *SAV and MEMory:DATA. Once stored on board, settings can be recalled with the command *RCL.

Syntax

```
MEMory:NSTates?
```

Parameters

<states>	Query Response
Not applicable	<NR1> 10

Reset Value

10

Errors and Events

None

Dependencies

None

Examples

Query: `MEM:NST?`

Response: 10

Related Commands

MEMory:DATA?
MEMory:STATe:CATalog?
MEMory:STATe:DEFine?
SYSTem:SET
*RCL
*SAV

www.valuetronics.com

MEMory:STATe:CATalog? (Query Only)

Returns the list of predefined names for the on-board stored settings. The list is composed of names as quoted strings separated by commas. Settings locations can be used or filled with the commands *SAV and MEMory:DATA. Once stored on board, settings can be recalled with the command *RCL.

Syntax

```
MEMory:STATe:CATalog?
```

Parameters

None

Reset Value

Not applicable

Errors and Events

None

Dependencies

None

Examples

Query:

```
MEM:STAT:CAT?
```

Response:

```
"SAV1,SAV2,SAV3,. . . ,SAV10"
```

Related Commands

MEMory:DATA
MEMory:NSTates?
MEMory:STATe:CATalog?
SYSTem:SET
*RCL
*SAV

www.valuetronics.com

MEMory:STATe:DEFine? (Query Only)

Returns the register number of a specified instrument settings location in the waveform analyzer. Settings locations can be used or filled with the commands *SAV and MEMory:DATA. Once stored on board, settings can be recalled with the command *RCL.

Syntax

```
MEMory:STATe:DEFine? <setting>
```

Parameters

<setting>	Query Response
SAV0	<NR1>
SAV1	1
SAV2	2
.	3
.	.
.	.
SAV10	10

Reset Value

Not applicable

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to query an invalid setting.

Dependencies

None

Examples

Query: `MEM:STAT:DEF? SAV3`

Response: 3

Related Commands

MEMory:DATA?
MEMory:NSTates?
MEMory:STATe:CATalog?
SYSTem:SET
*RCL
*SAV

www.valuetronics.com

OUTPut Subsystem

This section describes the commands in the OUTPut subsystem. These commands route signals to the VXIbus trigger lines and enable the probe compensation signals and reference signals.

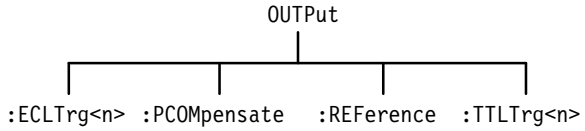


Figure 3–21: OUTPut Subsystem Hierarchy

OUTPut:ECLTrg<n>
OUTPut:ECLTrg<n>?

Sets or queries whether the instrument should drive (source) the specified VXIbus ECL trigger line when a trigger event occurs. You can choose inverted output polarity on the ECL trigger lines.

The waveform analyzer lets you to define the source to drive each VXIbus ECL trigger line. You can source and sense an ECL trigger line.

Syntax

```
OUTPut:ECLTrg<n>[:STATe] <boolean>
OUTPut:ECLTrg<n>[:STATe]?
```

Parameters

<n> (number of VXIbus trigger signal)	Query Response
<NR1> 0 or 1	<NR1>

<boolean>	Query Response
<NRf> 1 or ON 0 or OFF	<NR1> 1 0

Reset Value 0

www.valuetronics.com

Errors and Events None

Dependencies None

Examples Command: `OUTP:ECLTO ON`

Query: `OUTP:ECLTO?`

Response: 1

Related Commands OUTPUt:ECLTrg<n>:SOURce
OUTPUt:ECLTrg<n>:POLarity

OUTPut:ECLTrg<n>:POLarity
OUTPut:ECLTrg<n>:POLarity?

Sets or queries the drive polarity for the specified VXIbus ECL trigger line. Normally, the ECL trigger lines are asserted in the logical high state. With INVerted polarity the trigger lines are asserted in the logical low state.

You can enable each ECL trigger line separately with the command OUTPut:ECLTrg<n>. Use the command OUTPut:ECLTrg<n>:SOURce to control the source driver for the ECL trigger lines.

Syntax

```
OUTPut:ECLTrg<n>:POLarity<polarity>
OUTPut:ECLTrg<n>:POLarity?
```

Parameters

<n> (Line number)	Query Response
0 1	NA

<polarity>	Query Response
NORMal (active high) INVerted (active low)	NORM INV

Reset Value NORM

Errors and Events Execution Error –141, “Invalid character data”
Attempted to set polarity to an illegal value.

www.valuetronics.com

Dependencies

Examples

Command: `OUTP:ECLTO:POL INV`

Query: `OUTP:ECLTO:POL?`

Response: `INV`

Related Commands

OUTPut:ECLTrg<n>
OUTPut:ECLTrg<n>:SOURce

OUTPut:ECLTrg<n>:SOURce
OUTPut:ECLTrg<n>:SOURce?

Sets or queries the waveform analyzer source used to drive each VXIbus ECL trigger line. The ARMed, ATRigger, and BTRigger sources drive the chosen trigger line when the associated event occurs in the instrument. The OPC source drives the trigger line when the active command completes. The waveform analyzer can simultaneously source and sense an ECL trigger line.

Syntax

```
OUTPut:ECLTrg<n>:SOURce <source>
OUTPut:ECLTrg<n>:SOURce?
```

Parameters

<n> (Line number)	Query Response
0 1	NA

<source>	Query Response
ARMed	ARM
ATRigger	ATR
BTRigger	BTR
OPC	OPC

Reset Value

ATR (ECLTrg1)
BTR (ECLTrg0)

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to set source to an illegal value.

www.valuetronics.com

Dependencies None

Examples Command: `OUTP:ECLTO:SOUR BTR`

Query: `OUTP:ECLTO:SOUR?`

Response: `BTR`

Related Commands OUTPut:ECLTrg<n>
OUTPut:ECLTrg<n>:POLarity

OUTPut:PCOMpensate
OUTPut:PCOMpensate?

Sets or queries whether the probe compensation signal is output on the front-panel connector PROBE COMPENSATION. Use the command OUTPut:PCOMpensate:FUNCtion to select the type of compensation signal. You can select either CLOCk at 1 kHz and 500 mV peak-to-peak amplitude or VOLT with a 500 mV DC level.

When compensating probes, use the BNC to probe tip adapter recommended by the probe manufacturer and connect the probe ground lead to signal ground.

Syntax

```
OUTPut:PCOMpensate[:STATe] <boolean>
OUTPut:PCOMpensate[:STATe]?
```

Parameters

<boolean>	Query Response
<NRf>	<NR1>
1 or ON	1
0 or OFF	0

Reset Value 0

Errors and Events None

Dependencies None

Examples Command: `OUTP:PCOM ON`

Query: `OUTP:PCOM?`

www.valuetronics.com

Response: 1

Related Commands OUTPut:PCOMpensate:FUNC

OUTPut:PCOMpensate:FUNCtion
OUTPut:PCOMpensate:FUNCtion?

Sets or queries the source for the probe compensation signal which is output on the front-panel connector PROBE COMPENSATION. Use the command OUTPut:PCOMpensate to enable the signal. You can select either CLOCk at 1 kHz and 500 mV peak-to-peak amplitude or VOLT with a 500 mV DC level.

When compensating probes, use the BNC to probe tip adapter recommended by the probe manufacturer and connect the probe ground lead to signal ground.

Syntax

```
OUTPut:PCOMpensate:FUNCtion <function>
OUTPut:PCOMpensate:FUNCtion?
```

Parameters

<function>	Query Response
CLOCk	CLOC
VOLTage	VOLT

Reset Value CLOC

Errors and Events Execution Error –141, "Invalid character data"
Attempted to set function to an illegal value.

Dependencies None

Examples Command: `OUTP:PCOM:FUNC VOLT`

Query: `OUTP:PCOM:FUNC?`

Response: `VOLT`

Related Commands OUTPut:PCOMpensate

www.valuetronics.com

OUTPut:REFerence
OUTPut:REFerence?

Sets or queries whether a reference signal is output on the front-panel connector REFERENCE OUTPUT. The reference signal is intended for use during instrument verification. You can direct either of two signals to the REFERENCE OUTPUT: the internal 10 MHz reference clock with a 1 V peak-to-peak amplitude or the 8.0 V DC level. Use OUTPut:REFerence:FUNCtion to select which signal appears on the output. The reference signal is turned off when you start an acquisition or reset the instrument.

Syntax

```
OUTPut:REFerence[:STATe] <boolean>
OUTPut:REFerence[:STATe]?
```

Parameters

<boolean>	Query Response
<NRf>	<NR1>
1 or ON	1
0 or OFF	0

Reset Value

0

Errors and Events

None

Dependencies

None

Examples

Command: `OUTP:REF ON`

Query: `OUTP:REF?`

Response: 1

Related Commands

OUTPut:REFerence:FUNCtion

www.valuetronics.com

OUTPut:REFerence:FUNCtion
OUTPut:REFerence:FUNCtion?

Sets or queries which reference signal is output on the connector REFERENCE OUTPUT. The reference signal is intended for use during instrument verification. You can direct either of two signals to the REFERENCE OUTPUT: the internal 10 MHz reference clock with a 1 V peak-to-peak amplitude or the 8.0 V DC level. Use OUTPut:REFerence to enable the signal output. The reference signal output is disabled when you start acquisition or reset the instrument.

Syntax

```
OUTPut:REFerence:FUNCtion <function>
OUTPut:REFerence:FUNCtion?
```

Parameters

<function>	Query Response
CLOCk	CLOC
VOLTage	VOLT

Reset Value

CLOC

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to set the calibration reference function to an illegal value.

Dependencies

None

Examples

Command: `OUTP:REF:FUNC VOLT`

Query: `OUTP:REF:FUNC?`

Response: `VOLT`

Related Commands

OUTPut:REFerence

www.valuetronics.com

OUTPut:TTLTrg<n>
OUTPut:TTLTrg<n>?

Sets or queries whether the instrument should drive (source) the specified VXIbus TTL trigger lines when a trigger event occurs. You can enable each TTL trigger line separately. Use the command OUTPut:TTLTrg<n>:SOURce to pick the source for each TTL trigger line. Use the command OUTPut:TTLTrg<n>:POLarity to set the polarity of the output.

Syntax

```
OUTPut:TTLTrg<n>[:STATe] <boolean>
OUTPut:TTLTrg<n>[:STATe]?
```

Parameters

<n> (number of TTLTrg signal)	Query Response
<NR1> 0,1, 2, 3, 4, 5, 6, or 7	<NR1>

<boolean>	Query Response
<NRf> 1 or ON 0 or OFF	<NR1> 1 0

Reset Value 0

Errors and Events None

Dependencies None

Examples Command: `OUTP:TTLT0 ON`

Query: `OUTP:TTLT0?`

Response: 1

Related Commands OUTPut:TTLTrg<n>:POLarity
OUTPut:TTLTrg<n>:SOURce?

www.valuetronics.com

OUTPut:TTLTrg<n>:POLarity
OUTPut:TTLTrg<n>:POLarity?

Sets or queries the drive polarity for each VXI TTLTrg trigger line. The TTL trigger lines are normally active low. With INVerted polarity (active high), a controller can detect when all cards in a mainframe have asserted a trigger line.

Use the command OUTPut:TTLTrg<n>:SOURce to control the source driver for the TTL trigger lines. Use the command OUTPut:TTLTrg<n> to enable each TTLTrg driver.

Syntax

```
OUTPut:TTLTrg<n>:POLarity<polarity>
OUTPut:TTLTrg<n>:POLarity?
```

Parameters

<n> (number of TTLTrg signal)	Query Response
<NR1> 0,1, 2, 3, 4, 5, 6, or 7	<NR1>

<polarity>	Query Response
NORMal (active low) INVerted (active high)	NORM INV

Reset Value

NORM

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to set polarity to an illegal value.

Dependencies

Examples

Command: `OUTP:TTLT0:POL INV`

Query: `OUTP:TTLT0:POL?`

Response: `INV`

Related Commands

OUTPut:TTLTrg
OUTPut:TTLTrg<n>:SOURce?

www.valuetronics.com

OUTPut:TTLTrg<n>:SOURce
OUTPut:TTLTrg<n>:SOURce?

Sets or queries the waveform analyzer source for each VXI TTL trigger line. The waveform analyzer drives the VXIbus trigger lines only when they are enabled and a valid trigger event occurs. The available waveform analyzer sources are as follows:

- ARM — a valid ARM event occurs which enables the TRIG:A circuit
- ATR — a valid trigger event from the TRIGger A subsystem
- BTR — a valid trigger event from the TRIGger B subsystem
- OPC — the signal indicating the active command is complete. It is derived from the Operation Complete bit in the Standard Event Status Register.

Use the command OUTPut:TTLTrg<n> to enable each VXI TTLTrg line.

The waveform analyzer can simultaneously source and sense TTLTrg lines.

Syntax

```
OUTPut:TTLTrg<n>:SOURce <source>
OUTPut:TTLTrg<n>:SOURce?
```

Parameters

<n> (number of TTLTrg signal)	Query Response
<NR1> 0,1, 2, 3, 4, 5, 6, or 7	<NR1>

<source>	Query Response
ARMed	ARM
ATRigger	ATR
BTRigger	BTR
OPC	OPC

Reset Value

ARM (TTLTrg0)
ATR (TTLTrg1)
BTR (TTLTrg2)
OPC (TTLTrg3)
ARM (TTLTrg4)
ATR (TTLTrg5)
BTR (TTLTrg6)
OPC (TTLTrg7)

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to set source to an illegal value.

www.valuetronics.com

Dependencies None

Examples Query: `OUTP:TTLTO:SOUR ARM`

Query:`OUTP:TTLTO:SOUR?`

Response: `ARM`

Related Commands OUTPut:TTLTrg
OUTPut:TTLTrg<n>:POLarity

www.valuetronics.com

ROSCillator Subsystem

This section describes the commands in the SENSe:ROSCillator subsystem. These commands control the source of the reference oscillator (clock) for the :SWEep subsystem.

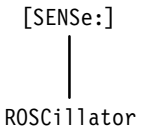


Figure 3–22: ROSCillator Subsystem

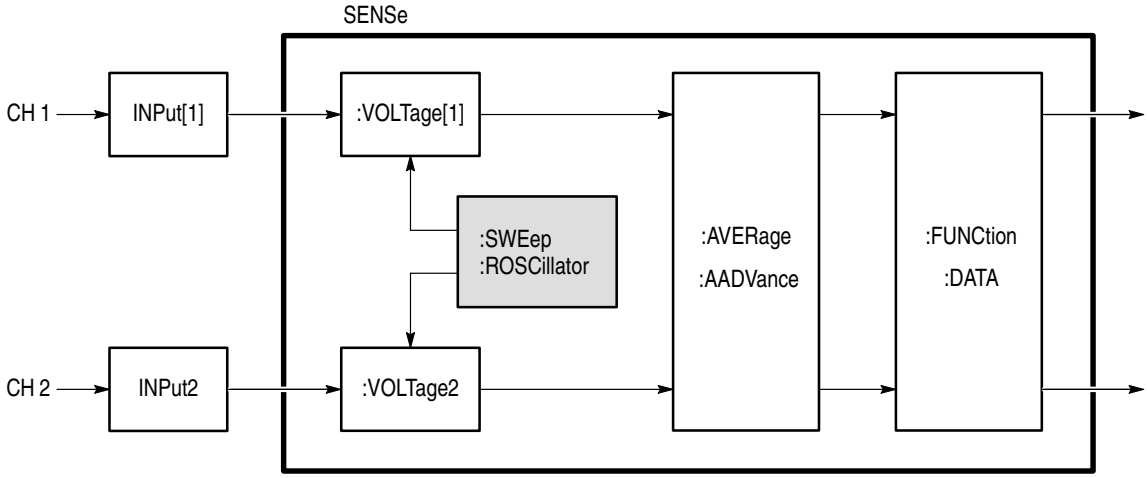


Figure 3–23: ROSCillator Subsystem Functional Model

www.valuetronics.com

ROSCillator:SOURce
ROSCillator:SOURce?

Sets or queries the source of the 10 MHz clock reference for the acquisition system. The choices are CLK10 which specifies the VXIbus 10 MHz reference, and INTernal which specifies the internal waveform analyzer oscillator.

Syntax

```
[SENSe:]ROSCillator:SOURce <source>
[SENSe:]ROSCillator:SOURce?
```

Parameters

<source>	Query Response
CLK10 INTernal	CLK10 INT

Reset Value

INT

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to set source to an illegal value.

Dependencies

None

Examples

Command: `ROSC:SOUR CLK10`

Query: `ROSC:SOUR?`

Response: `CLK10`

Related Commands

SWEep:TINTerval

www.valuetronics.com

STATus Subsystem

This section describes the commands in the STATus subsystem. These commands, along with several IEEE 488.2 Common Commands, control the status and event reporting system. The STATus subsystem provides a way to determine the state of the waveform analyzer and control what events can interrupt the system controller. For an overview of the status and event reporting system, refer to page 4–1.

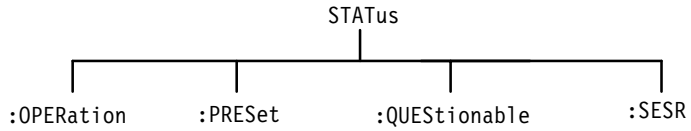


Figure 3–24: STATus Subsystem Hierarchy

STATus:OPERation? (Query Only)

Returns the contents of the Operation Status Register as a decimal number. The Operation Status Register, described in Table 3–26, identifies normal events, such as acquisition, that are in progress. Use the Operation Status Enable Register to determine which Operation Status events can set the Operation bit (bit 7) in the Status Byte Register.

Note that reading this register does not clear it.

Table 3–26: The Operation Status Register

Bit	Decimal Value	Function
0	1	**Calibrating** shows that a calibration routine is in progress.
1–3		Not used.
4	16	**Measuring/Acquiring** shows that measurement or acquisition is in progress.
5	32	**Waiting for Trigger** shows that the acquisition system is armed and waiting for a trigger event.
6	64	**Waiting for Arm** shows that the acquisition system has been initialized with INIT and is waiting to be armed.
7		Not used.
8	256	**Testing** shows that a self test routine is in progress.

www.valuetronics.com

Table 3–26: The Operation Status Register (Cont.)

Bit	Decimal Value	Function
9	512	**CH 1 Probe** shows that a probe is attached to the Channel 1 input.
10	1024	**CH 2 Probe** shows that a probe is attached to the Channel 2 input.
11	2048	**CH 3 Probe** shows that a probe is attached to the Channel 3 input.
12	4096	**CH 4 Probe** shows that a probe is attached to the Channel 4 input.
13–15		Not used.

Syntax

```
STATus:OPERation[:EVENt]?
```

Parameters

<event>	Query Response
Not applicable	<NR1>

Reset Value

Not applicable

Errors and Events

None

Dependencies

None

Examples

Query: `STAT:OPER?`

Response: `512`
This response means that the only registered Operation is a probe connected to input Channel 1.

Related Commands

STATus:OPERation:CONDition?
STATus:OPERation:ENABle
STATus:OPERation:PTRansition
STATus:OPERation:NTRansition
STATus:OPERation:QENable:NTRansition
STATus:OPERation:QENable:PTRansition

www.valuetronics.com

STATus:OPERation:CONDition? (Query Only)

Returns the contents of the Operation Status Condition Register (OSCR). The OSCR bits correspond to the Operation Status Register bits. The query response from the Operation Status Condition Register gives the current state of the Operation event lines prior to the Operation Transition Registers. The query returns the current setting as a 16 bit number whose bits correspond to the Operation Status Register bits. Refer to Table 3–26 on page 3–151 for a definition of the events associated with the Operation Status Register bits. For an overview of the status and event reporting system, refer to page 4–1

Note that reading this register does not clear it and unused bits always return zero.

Syntax

```
STATus:OPERation:CONDition?
```

Parameters

<condition>	Query Response
Not applicable	<NR1>

Reset Value

Not applicable

Errors and Events

None

Dependencies

None

Examples

Query: `STAT:OPER:COND?`

Response: 1

Related Commands

STATus:OPERation?
STATus:OPERation:ENABle

www.valuetronics.com

STATus:OPERation:ENABle
STATus:OPERation:ENABle?

Sets or queries the contents of the Operation Status Enable Register (OSER). The OSER allows you to individually disable any of the Operation events from setting bit 7 in the Status Byte Register. The query returns the current setting as a 16 bit number whose bits correspond to the Operation Status Register bits. Refer to Table 3–26 on page 3–151 for a definition of the events associated with the Operation Status Register bits. Setting unused enable bits does not generate an error, they are ignored. For an overview of the status and event reporting system, refer to page 4–1.

STATus:PRESet sets all register bits to zero (0) which disables all Operation events.

Syntax

```
STATus:OPERation:ENABle <mask>
STATus:OPERation:ENABle?
```

Parameters

<mask>	Query Response
<NRf> | <Non-decimal numeric> 0 ≤ N ≤ #HFFFF	<NR1>

Reset Value

Not applicable

Errors and Events

Execution Error –222, "Data out of range"
Attempted to set the enable register to an illegal value.

Dependencies

None

Examples

Command: `STAT:OPER:ENAB #H0010`
This command, (decimal 16), enables only the Acquiring/Measuring event.

Query: `STAT:OPER:ENAB?`

Response: `10`

Related Commands

STATus:OPERation?
STATus:OPERation:CONDition?
STATus:PRESet

www.valuetronics.com

STATus:OPERation:NTRansition
STATus:OPERation:NTRansition?

Sets or queries the contents of the Operation Negative Transition Register (ONTR). When you set an event bit true (1) in the ONTR, the ONTR reports an event in the Operation Status Register when the Operation Status Condition Register event line changes from true to false (1 to 0). The query returns the current setting as a 16 bit number whose bits correspond to the Operation Status Register bits. Refer to Table 3–26 on page 3–151 for a definition of the events associated with the Operation Status Register bits. The command STATus:OPERation:PTRansition provides similar control for positive event transitions. For an overview of the status and event reporting system, refer to page 4–1.

STATus:PRESet sets all register bits to zero (0) which disables reporting negative event transitions. Setting unused enable bits does not generate an error, they are ignored.

Syntax

```
STATus:OPERation:NTRansition <mask>
STATus:OPERation:NTRansition?
```

Parameters

<mask>	Query Response
<NRf> | <Non-decimal numeric> 0 ≤ N ≤ #HFFFF	<NR1>

Reset Value

Not applicable

Errors and Events

Execution Error –222, "Data out of range"
Attempted to set the transition register to an illegal value.

Dependencies

None

Examples

Command: `STAT:OPER:NTR #H0010`
This command, (decimal 16), sets the Operation Negative Transition Register to set the Acquiring/Measuring event bit when the current measurement or acquisition completes.

Query: `STAT:OPER:NTR?`

Response: `10`

www.valuetronics.com

Related Commands

STATus:OPERation?
STATus:OPERation:CONDition?
STATus:OPERation:PTRansition
STATus:PRESet

STATus:OPERation:PTRansition
STATus:OPERation:PTRansition?

Sets or queries the contents of the Operation Positive Transition Register (OPTR). When you set an event bit true (1) in the OPTR, the OPTR reports an event in the Operation Status Register when the Operation Status Condition Register event line changes from false to true (0 to 1). The query returns the current setting as a 16 bit number whose bits correspond to the Operation Status Register bits. Refer to Table 3–26 on page 3–151 for a definition of the events associated with the Operation Status Register bits. The command STATus:OPERation:NTRansition provides similar control for negative event transitions. For an overview of the status and event reporting system, refer to page 4–1.

STATus:PRESet sets all register bits to one (1) which enables reporting positive event transitions. Setting unused enable bits does not generate an error, they are ignored.

Syntax

```
STATus:OPERation:PTRansition <mask>
STATus:OPERation:PTRansition?
```

Parameters

<mask>	Query Response
<NRf> | <Non-decimal numeric> 0 ≤ N ≤ #HFFFF	<NR1>

Reset Value

Not applicable

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set the transition register to an illegal value.

Dependencies

None

Examples

Command: `STAT:OPER:PTR #H0010`
With this parameter value, (decimal 16), the Operation Positive Transition Register sets *only* the Acquiring/Measuring event bit

www.valuetronics.com

when measurement or acquisition starts. All other positive Operation events are disabled.

Query: `STAT:OPER:PTR?`

Response: 10

Related Commands

STATus:OPERation?
STATus:OPERation:CONDition?
STATus:OPERation:NTRansition
STATus:PRESet

STATus:OPERation:QENable:NTRansition
STATus:OPERation:QENable:NTRansition?

Sets or queries the contents of the Negative Transition Queue Enable Register (NTQER) for the Operation Status Register. When you set an event bit true (1) in the NTQER, the corresponding event the Operation Status Condition Register generates a message in the Status Queue when the event line changes from true to false (1 to 0). The NTQER bits correspond to the Operation Status Register bits. Refer to Table 3–26 on page 3–151 for a definition of the events associated with the Operation Status Register bits. The command STATus:OPERation:QENable:PTRansition provides similar control for positive event transitions. For an overview of the status and event reporting system, refer to page 4–1.

STATus:PRESet sets all register bits to zero (0) which disables the reporting of negative event transitions. Setting unused enable bits does not generate an error, they are ignored.

Syntax

```
STATus:OPERation:QENable:NTRansition <mask>
STATus:OPERation:QENable:NTRansition?
```

Parameters

<mask>	Query Response
<NRf> | <Non-decimal numeric> 0 ≤ N ≤ #HFFFF	<NR1>

Reset Value

Not applicable

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set the queue enable register to an illegal value.

www.valuetronics.com

Dependencies None

Examples Command: `STAT:OPER:QEN:NTR #H0001`

Query: `STAT:OPER:QEN:NTR?`

Response: 1

Related Commands STATus:OPERation?
STATus:OPERation:QENable:PTRansition
STATus:OPERation:CONDition?
STATus:OPERation:NTRansition
STATus:PRESet

STATus:OPERation:QENable:PTRansition
STATus:OPERation:QENable:PTRansition?

Sets or queries the contents of the Positive Transition Queue Enable Register (PTQER) for the Operation Status Register. When you set an event bit true (1) in the PTQER, the corresponding event the Operation Status Condition Register generates a message in the Status Queue when the event line changes from false to true (0 to 1). The PTQER bits correspond to the Operation Status Register bits. Refer to Table 3–26 on page 3–151 for a definition of the events associated with the Operation Status Register bits. The command STATus:OPERation:QENable:NTRansition provides similar control for negative event transitions. For an overview of the status and event reporting system, refer to page 4–1.

STATus:PRESet sets all register bits to zero (0) which disables reporting positive event transitions. Setting unused enable bits does not generate an error, they are ignored.

Syntax

```
STATus:OPERation:QENable:PTRansition <mask>
STATus:OPERation:QENable:PTRansition?
```

Parameters

<mask>	Query Response
<NRf> | <Non-decimal numeric> 0 ≤ N ≤ #HFFFF	<NR1>

Reset Value Not applicable

www.valuetronics.com

Errors and Events Execution Error –222, "Data out of range"
Attempted to set the queue enable register to an illegal value.

Dependencies None

Examples Command: `STAT:OPER:QEN:PTR #H0001`

Query: `STAT:OPER:QEN:PTR?`

Response: 1

Related Commands STATus:OPERation?
STATus:OPERation:QENable:NTRansition
STATus:OPERation:CONDition?
STATus:OPERation:NTRansition
STATus:PRESet

STATus:PRESet

Presets the SCPI Enable and Transition registers, and the Status Queue enable registers. The Operation and Questionable Enable Registers are preset to zero (all events disabled). Refer to the register command descriptions for the effects of STATus:PRESet on specific registers. This command does not clear SCPI Status registers or 488.2 enable and event registers. For an overview of the status and event reporting system, refer to page 4–1.

Syntax `STATus:PRESet`

Parameters None

Reset Value Not applicable

Errors and Events None

Dependencies None

Examples Command: `STAT:PRES`

Related Commands STATus:OPERation:ENABle
STATus:OPERation:NTRansition

www.valuetronics.com

STATus:OPERation:PTRansition
STATus:OPERation:QENable:NTRansition
STATus:OPERation:QENable:PTRansition
STATus:QUEStionable:ENABle
STATus:QUEStionable:NTRansition
STATus:QUEStionable:PTRansition
STATus:QUEStionable:QENable:NTRansition
STATus:QUEStionable:QENable:PTRansition
STATus:SESR:QENable
*CLS

STATus:QUEStionable? (Query Only)

Returns the contents of the Questionable Status Register as a decimal number. The Questionable Status Register, defined in Table 3–27, identifies operations whose results are questionable. Acquiring a waveform when the waveform analyzer is in need of calibration can generate an event on bit 8. Use the Operation Status Enable Register to determine which Questionable events can set the Questionable bit (bit 3) in the Status Byte Register. For an overview of the status and event reporting system, refer to page 4–1.

Reading this register clears it. Unused and reserved bits always return zero.

Table 3–27: The Questionable Status Register

Bit	Decimal Value	Function
0–7		Not used.
8	256	**Calibration** indicates that calibration is required due a change of greater than 5° C in ambient temperature since the last calibration.
9	512	**Calculate1** indicates that source data contained a value that was overrange or underrange, making the results of the CALC1 block questionable.
10	1024	**Calculate2** indicates that source data contained a value that was overrange or underrange, making the results of the CALC2 block questionable.
11	2048	**Calculate3** indicates that source data contained a value that was overrange or underrange, making the results of the CALC3 block questionable.
12	4096	**Calculate4** indicates that source data contained a value that was overrange or underrange, making the results of the CALC4 block questionable.
13–15		Not used.

www.valuetronics.com

Syntax `STATus:QUEStionable[:EVENt]?`

Parameters

<event>	Query Response
Not applicable	<NR1>

Reset Value Not applicable

Errors and Events None

Examples Query: `STAT:QUES?`

Response: 1

Related Commands STATus:QUEStionable:CONDition?
STATus:QUEStionable:ENABle

STATus:QUEStionable:CONDition? (Query Only)

Returns the contents of the Questionable Status Condition Register (QSCR). The QSCR bits correspond to the Questionable Status Register bits. The query response from the Questionable Status Condition Register gives the current state of the Questionable event lines prior to the Questionable Transition Registers. The query returns the current setting as a 16 bit number whose bits correspond to the Questionable Status Register bits. Refer to Table 3–27 on page 3–160 for a definition of the events associated with the Questionable Status Register bits. For an overview of the status and event reporting system, refer to page 4–1.

Note that reading this register does not clear it and unused bits always return zero.

Syntax `STATus:QUEStionable:CONDition?`

Parameters

<condition>	Query Response
Not applicable	<NR1>

Reset Value Not applicable

Errors and Events None

www.valuetronics.com

Dependencies None

Examples Query: `STAT:QUES:COND?`

Response: 1

Related Commands STATus:QUEStionable?
STATus:QUEStionable:ENABle

STATus:QUEStionable:ENABle
STATus:QUEStionable:ENABle?

Sets or queries the contents of the Questionable Status Enable Register (QSER). The QSER allows you to individually disable any of the Questionable events from setting bit 3 in the Status Byte Register. The query returns the current setting as a 16 bit number whose bits correspond to the Questionable Status Register bits. Refer to Table 3–27 on page 3–160 for a definition of the events associated with the Questionable Status Register bits. For an overview of the status and event reporting system, refer to page 4–1.

STATus:PRESet sets all register bits to zero (0) which disables all Questionable events. Setting unused enable bits does not generate an error, they are ignored.

Syntax
```
STATus:QUEStionable:ENABle <mask>
STATus:QUEStionable:ENABle?
```

Parameters

<mask>	Query Response
<NRf> | <Non-decimal numeric> 0 ≤ N ≤ #HFFFF	<NR1>

Reset Value Not applicable

Errors and Events Execution Error –222, “Data out of range”
Attempted to set the enable register to an illegal value.

Dependencies None

Examples Command: `STAT:QUES:ENAB #H0001`

Query: `STAT:QUES:ENAB?`

www.valuetronics.com

Response: 1

Related Commands

STATus:PRESet
STATus:QUEStionable?
STATus:QUEStionable:CONDition?

STATus:QUEStionable:NTRansition
STATus:QUEStionable:NTRansition?

Sets or queries the contents of the Questionable Negative Transition Register (QNTR). When you set an event bit true (1) in the QNTR, the QNTR reports an event in the Questionable Status Register when the Questionable Status Condition Register event line changes from true to false (1 to 0). The query returns the current setting as a 16 bit number whose bits correspond to the Questionable Status Register bits. Refer to Table 3–27 on page 3–160 for a definition of the events associated with the Questionable Status Register bits. The command STATus:QUEStionable:PTRansition provides similar control for positive event transitions. For an overview of the status and event reporting system, refer to page 4–1

STATus:PRESet sets all register bits to zero (0) which disables reporting negative event transitions. Setting unused enable bits does not generate an error, they are ignored.

Syntax

```
STATus:QUEStionable:NTRansition <mask>
STATus:QUEStionable:NTRansition?
```

Parameters

<mask>	Query Response
<NRf> | <Non-decimal numeric> 0 ≤ N ≤ #HFFFF	<NR1>

Reset Value

Not applicable

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set the transition register to an illegal value.

Dependencies

None

www.valuetronics.com

Examples

Command: `STAT:QUES:NTR #H0100`
With this command, (decimal 256), the Questionable Negative Transition Register sets the Calibration event bit when the calibration completes.

Query: `STAT:QUES:NTR?`

Response: `100`

Related Commands

STATus:QUEStionable?
STATus:QUEStionable:CONDition?
STATus:QUEStionable:PTRansition
STATus:PRESet

STATus:QUEStionable:PTRansition
STATus:QUEStionable:PTRansition?

Sets or queries the contents of the Questionable Positive Transition Register (QPTR). When you set an event bit true (1) in the QPTR, the QPTR reports an event in the Questionable Status Register when the Questionable Status Condition Register event line changes from false to true (0 to 1). The query returns the current setting as a 16 bit number whose bits correspond to the Questionable Status Register bits. Refer to Table 3–27 on page 3–160 for a definition of the events associated with the Questionable Status Register bits. The command STATus:QUEStionable:NTRansition provides similar control for negative event transitions. For an overview of the status and event reporting system, refer to page 4–1

STATus:PRESet sets all register bits to one (1) which enables reporting positive event transitions. Setting unused enable bits does not generate an error, they are ignored.

Syntax

```
STATus:QUEStionable:PTRansition <mask>
STATus:QUEStionable:PTRansition?
```

Parameters

<mask>	Query Response
<NRf> | <Non-decimal numeric> 0 ≤ N ≤ #HFFFF	<NR1>

Reset Value

Not applicable

www.valuetronics.com

Errors and Events

Execution Error –222, "Data out of range"
Attempted to set the transition register to an illegal value.

Dependencies

None

Examples

Command: `STAT:QUES:PTR #H0100`
With this command, (decimal 256), the Questionable Positive Transition Register sets the Calibration event bit when calibration becomes necessary. All other positive events in the Questionable Condition Register are ignored.

Query: `STAT:QUES:PTR?`

Response: `100`

Related Commands

STATus:QUEStionable?
STATus:QUEStionable:CONDition?
STATus:QUEStionable:NTRansition
STATus:PRESet

STATus:QUEStionable:QENable:NTRansition
STATus:QUEStionable:QENable:NTRansition?

Sets or queries the contents of the Negative Transition Queue Enable Register (NTQER) for the Questionable Status Register. When you set an event bit true (1) in the NTQER, the corresponding event the Questionable Status Condition Register generates a message in the Status Queue when the event line changes from true to false (1 to 0). The query returns the current setting as a 16 bit number whose bits correspond to the Questionable Status Register bits. Refer to Table 3–27 on page 3–160 for a definition of the events associated with the Questionable Status Register bits. The command STATus:QUEStionable:QENable:PTRansition provides similar control for positive event transitions. For an overview of the status and event reporting system, refer to page 4–1.

STATus:PRESet sets all register bits to zero (0) which disables reporting negative event transitions. Setting unused enable bits does not generate an error, they are ignored.

Syntax

```
STATus:QUEStionable:QENable:NTRansition <mask>
STATus:QUEStionable:QENable:NTRansition?
```


www.valuetronics.com

Parameters

<mask>	Query Response
<NRf> | <Non-decimal numeric> 0 ≤ N ≤ #HFFFF	<NR1>

Reset Value Not applicable

Errors and Events Execution Error –222, "Data out of range"
Attempted to set the queue enable register to an illegal value.

Dependencies None

Examples Command: `STAT:QUES:QEN:NTR #H0100`

Query: `STAT:QUES:QEN:NTR?`

Response: `100`

Related Commands STATus:QUEStionable?
STATus:QUEStionable:QENable:PTRansition
STATus:QUEStionable:CONDition?
STATus:QUEStionable:NTRansition
STATus:PRESet

STATus:QUEStionable:QENable:PTRansition
STATus:QUEStionable:QENable:PTRansition?

Sets or queries the contents of the Positive Transition Queue Enable Register (PTQER) for the Questionable Status Register. When you set an event bit true (1) in the PTQER, the corresponding event the Questionable Status Condition Register generates a message in the Status Queue when the event line changes from false to true (0 to 1). The query returns the current setting as a 16 bit number whose bits correspond to the Questionable Status Register bits. Refer to Table 3–27 on page 3–160 for a definition of the events associated with the Questionable Status Register bits. The command STATus:QUEStionable:QENable:NTRansition provides similar control for negative event transitions. For an overview of the status and event reporting system, refer to page 4–1.

STATus:PRESet sets all register bits to zero (0) which disables reporting positive event transitions. Setting unused enable bits does not generate an error, they are ignored.

www.valuetronics.com

Syntax

```
STATus:QUEStionable:QENable:PTRansition <mask>
STATus:QUEStionable:QENable:PTRansition?
```

Parameters

<mask>	Query Response
<NRf> | <Non-decimal numeric> 0 ≤ N ≤ #HFFFF	<NR1>

Reset Value

Not applicable

Errors and Events

Execution Error –222, "Data out of range"
Attempted to set the queue enable register to an illegal value.

Dependencies

None

Examples

Command: `STAT:QUES:QEN:PTR #H0100`

Query: `STAT:QUES:QEN:PTR?`

Response: `100`

Related Commands

STATus:QUEStionable?
STATus:QUEStionable:QENable:NTRansition
STATus:QUEStionable:CONDition?
STATus:QUEStionable:NTRansition
STATus:PRESet

www.valuetronics.com

STATus:SESR:QENable
STATus:SESR:QENable?

Sets or returns the contents of the Event Status Enable Register (ESER). When you set an event bit true (1) in the ESER, the corresponding event the Standard Event Status Register generates a message in the Status Queue when the event line changes from false to true (0 to 1).

STATus:PRESet sets the register bits to hexadecimal 4C (0).

Syntax

```
STATus:SESR:QENable <mask>
STATus:SESR:QENable?
```

Parameters

<mask>	Query Response
<NRf> | <Non-decimal numeric> 0 ≤ N ≤ #HFFFF	None

Reset Value

Not applicable

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set the queue enable register to an illegal value.

Dependencies

None

Examples

Command: `STAT:SESR:QEN #H7C`

Query: `STAT:SESR:QEN?`

Response: `124`
This response is the decimal equivalent of hexadecimal 7C.

Related Commands

*ESE
*ESR
STATus:PRESet

www.valuetronics.com

SWEep Subsystem

This section describes the commands in the [SENSe:]SWEep subsystem. These commands control the acquisition timebase for all VOLTage[n] acquisitions.

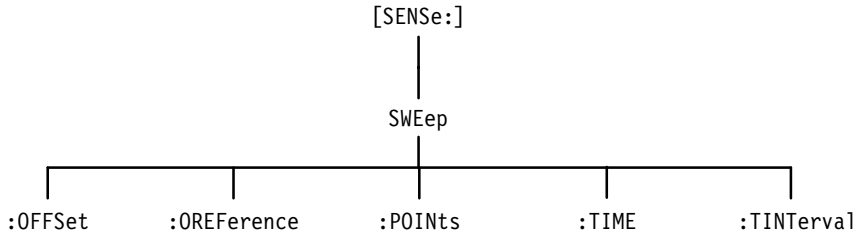


Figure 3–25: SWEep Subsystem Hierarchy

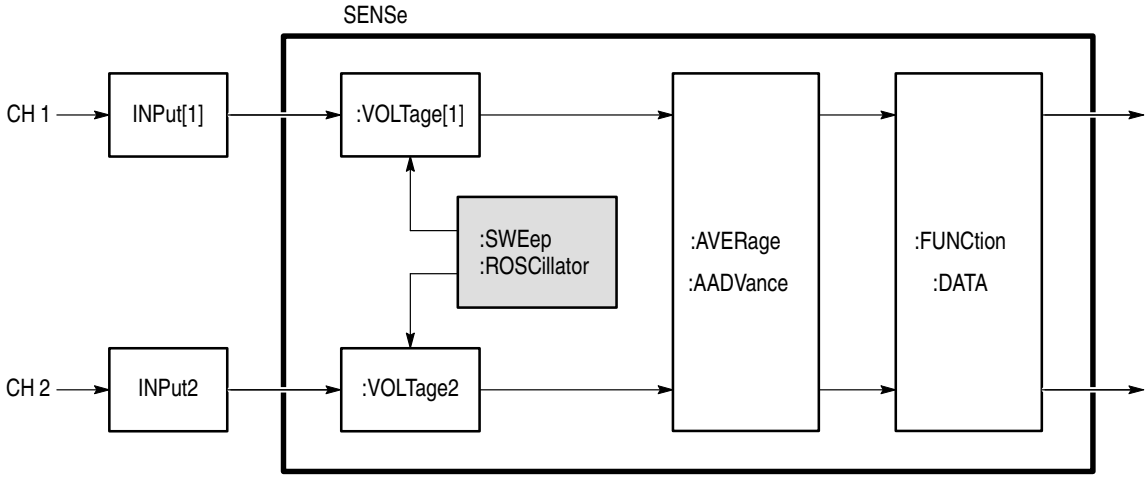


Figure 3–26: SWEep Subsystem Functional Model

www.valuetronics.com

SWEep:OFFSet:POINts
SWEep:OFFSet:POINts?

Sets or queries the position of the waveform record relative to the trigger point. SWEep:OFFSet:POINts defines the number of data points between the offset reference point and the trigger point. You set the offset reference with SWEep:OREFerence:LOCation. All channels use the same sweep offset.

You can think of the offset reference point as a "handle" that you place on a specific data point in the waveform record. You position the "handle" to move the waveform record relative to the trigger point. The command OFFSet:POINts specifies how far and in what direction to move the "handle." Setting OFFSet:POINts to a negative value positions the offset reference (the handle) before the trigger point. A positive value positions the offset reference after the trigger point.

For example, to set the trigger point in the middle of the waveform record, you would first set the offset reference point to the first record point (SWE:OFFS:POIN 0.0). Then you would set SWE:OFFS:POIN to minus one half the record length (set with SWEep:POINTs). If the record length is 1024, then to specify the half way point you would use SWE:OFFS:POIN –512.

The range for SWEep:OFFSet:POINts depends on the record length and the location of the offset reference point within the waveform record. Note that the trigger point must always be within the waveform record. You can set offset as a time interval with the command OFFSet:TIME.

The following equation defines the first point of the waveform record:

$$PT1_{time} = (\text{SWEep:OFFSet:POINts} \times \text{SWEep:TINTerval}) - (\text{SWEep:OREFerence:LOCation} \times \text{SWEep:TIME})$$

For more information, refer to the *Acquisition* discussion on page 2–9.

Syntax

```
[SENSe:]SWEep:OFFSet:POINts <offset_points>
[SENSe:]SWEep:OFFSet:POINts?
```

Parameters

<offset_points>	Query Response
<NRf>	<NR1>
MINimum –	(SWEep:OREFerence:LOCation × SWEep:POINts) – SWEep:POINts
MAXimum	(SWEep:OREFerence:LOCation × SWEep:POINts)

Reset Value

0.0E+0

www.valuetronics.com

Errors and Events

Execution Error –222, "Data out of range"
Attempted to set offset points to an illegal value.

Dependencies

Changing SWEep:OFFSet:POINts also changes SWEep:OFFSet:TIME as they both control the parameter SWEep:OFFSet.

Examples

Command: `SWE:OFFS:POIN -100`

Query: `SWE:OFF:POIN?`

Response: `-100`

Related Commands

SWEep:OREFerence:LOCation
SWEep:OFFSet:TIME
SWEep:POINts
SWEep:TINTerval

SWEep:OFFSet:TIME
SWEep:OFFSet:TIME?

Sets or queries the offset between the offset reference point and the trigger point. You define OFFSet:TIME as a specific period. You assign a certain data point in the record to be the offset reference point (SWEep:OREFerence:LOCation). A negative value for offset positions the offset reference point before the trigger point. All channels use the same sweep offset.

The range for SWEep:OFFSet:TIME depends on the record length and the location of the offset reference point within the waveform record. Because the trigger point must always be in the waveform record, the first point of the record can not be placed more than one full record length before the trigger point. You can set offset as a number of data points with the command SWEep:OFFSet:POINts.

The following equation defines the first point of the waveform record:

$PT1_{time}$ = SWEep:OFFSet:TIME –
(SWEep:OREFerence:LOCation * SWEep:TIME)

For more information on setting offset, refer to the discussion of SWEep:OFFSet:POINts on page 3–170 and the Acquisition discussion on page 2–9.

Syntax

```
[SENSe:]SWEep:OFFSet:TIME <offset_time>
[SENSe:]SWEep:OFFSet:TIME?
```


www.valuetronics.com

Parameters

<offset_time>[1]	Query Response
<NRf>	<NR3>
MINimum	(SWEep:OREFerence:LOCation × SWEep:TIME) – (SWEep:POINts × SWEep:TINTerval)
MAXimum	(SWEep:OREFerence:LOCation × SWEep:TIME)

1 **The default multiplier for <offset_time> is S for seconds. You can also use the multipliers MS for milliseconds, US for microseconds, NS for nanoseconds, and PS for picoseconds.**

Reset Value

0.0E+0

Errors and Events

Execution Error –222, "Data out of range"
Attempted to set offset time to an illegal value.

Dependencies

Changing SWEep:OFFSet:TIME also changes SWEep:OFFSet:POINts as they both control the parameter SWEep:OFFSet.

Examples

Command: `SWE:OFFSet:TIME -100E-6`

Query: `SWE:OFFSet:TIME?`

Response: `-100.0E-6`

Related Commands

SWEep:OREFerence:LOCation
SWEep:OFFSet:POINts
SWEep:TINTerval
SWEep:POINts

www.valuetronics.com

SWEep:OREFerence:LOCation
SWEep:OREFerence:LOCation?

Sets or queries the location of the offset reference point in a waveform record. When :OFFSet:TIME and :OFFSet:POINts are at zero, the reference point and trigger point are at the same data point. For example, if you set :OREFerence:LOCation to 0.5 (or 50%) and :OFFSet:TIME to zero, then the reference point is at the middle of the waveform record as is the trigger point. Therefore, half of the acquired data points are acquired before the trigger point and the remainder occur after the trigger point.

An offset reference value of 0.0 selects the first point of the record; a value of 0.5 selects the mid point; and a value of 1.0 selects the last point. All concurrent acquisitions have the same offset reference point.

You can use either of the commands SWEep:OFFSet:TIME or SWEep:OFFSet:POINts to position the waveform record relative to the trigger point. To configure an acquisition in terms of percent of pre-trigger data points, set SWEep:OFFSet:TIME to 0.0 and then set SWEep:OREFerence:LOCation to the desired value (for example, 0.5 for 50% pre-trigger points). To configure an acquisition in terms of time (or points) relative to the trigger point, set SWEep:OREFerence:LOCation to 0.0 and then set SWEep:OFFSet:TIME (or SWEep:OFFSet:POINts) to position the record. By using :OFFSet in this way, you can acquire all data points before the trigger or all data points after the trigger point.

For more information, refer to the discussion on *Acquisition* located on page 2–9.

Syntax

```
[SENSe:]SWEep:OREFerence:LOCation <oref_location>
[SENSe:]SWEep:OREFerence:LOCation?
```

Parameters

<oref_location>[1]	Query Response
No Units: <NRf> 0.0 ≤ N ≤ 1.0 MINimum MAXimum	 <NR2> 0.0 1.0
Units of Percent (PCT): <NRf> 0 ≤ N ≤ 100 MINimum (0 PCT) MAXimum (100 PCT)	 <NR2> 0 1.0

1 **The default has no units. However, you can set the reference location as a percentage by including the unit multiplier PCT as in 50 PCT.**

www.valuetronics.com

Reset Value 0.0

Errors and Events Execution Error –222, "Data out of range"
Attempted to set offset reference location to an illegal value.

Dependencies None

Examples Command: `SWE:OREF:LOC 0.5`

Query: `SWE:OREF:LOC?`

Response: `0.5`

Related Commands SWEep:OFFSet:TIME
SWEep:OFFSet:POINts

SWEep:POINts
SWEep:POINts?

Sets or queries the number of data points in a waveform record. The number of data points in a waveform record is its record length. The minimum record length is 256 data points and the maximum is 15,000 real time (RT) acquisition and 30,000 for extended real time (ERT) acquisition. All active channels share the record length setting.

The maximum record length for calculation or measurement functions (using a CALCulate block) or for auto-advance acquisition mode is 30,000.

Syntax

```
[SENSe:]SWEep:POINts <record_length>
[SENSe:]SWEep:POINts?
```


www.valuetronics.com

Parameters

<record_length> (Number of data points)	Query Response
<NRf>	<NR1>
256	256
512	512
1024	1024
2048	2048
4096	4096
8192	8192
15000	15000
30000 (ERT mode only)	30000
MINimum	256
MAXimum RT mode	15000
MAXimum ERT mode	30000

Reset Value

1024

Errors and Events

Execution Error –222, "Data out of range"
Attempted to set record length to an illegal value or to a value > 15,000 while in real time acquisition or to a value > 30,000 while in extended real time acquisition.

Dependencies

Changing SWEep:POINts changes SWEep:TIME if SWEep:TINTerval remains constant. The relationship is
SWEep:TINTerval = SWEep:TIME / SWEep:POINts.

Examples

Command: `SWE:POIN 1000`

Query: `SWE:POIN?`

Response: `1024`

Related Commands

SWEep:TINTerval
SWEep:TIME

www.valuetronics.com

SWEep:TIME
SWEep:TIME?

Sets or queries the time span or duration of the waveform record. All channels share the same time span. Set SWEep:POINts for the desired record length before setting the record duration with SWEep:TIME. Issuing these two commands sets the sample interval (SWEep:TINTerval) to the nearest legal value.

When using SWEep:POINts and SWEep:TIME to configure the acquisition, be careful to program SWEep:POINts before SWEep:TIME, because SWEep:POINts may modify SWEep:TIME.

SWEep:TIME is not returned in the response to the query *LRN?, since it can be derived as follows: SWEepTIME = SWEep:POINts * SWEep:TINTerval

Syntax

```
[SENSe:]SWEep:TIME <time_span>
[SENSe:]SWEep:TIME?
```

Parameters

<time_span> [1]	Query Response
<NRf> 51.2E–9 ≤ N ≤ 6000 (seconds)	<NR3>
MINimum	256 × sample interval
MAXimum	6000.0E+0

1 **The default multiplier for <time_span> is S for seconds. You can also use the multipliers MS for milliseconds, US for microseconds, NS for nanoseconds, and PS for picoseconds.**

Reset Value

1.024E–6

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set time span to an illegal value.

Execution Error –241, “Hardware missing”
Attempted to set time span to 51.2E–9 when the instrument is a TVS621 or TVS641.

Dependencies

Changing SWEep:TIME changes SWEep:TINTerval if SWEep:POINts remains constant. The relation is SWEep:TINTerval = SWEep:TIME / (SWEep:POINts).

Examples

Command: `SWE:TIME 1E-3`

www.valuetronics.com

Query: `SWE:TIME?`

Response: `1.024E-3`

Related Commands

SWEep:POINts
SWEep:TINTerval

SWEep:TINTerval
SWEep:TINTerval?

Sets or queries the time interval between acquired data points. The reciprocal of the time interval gives the effective sample rate. All channels are acquired with the same time interval.

The point at which the instrument transitions from real time (RT) acquisition to extended real time (ERT) acquisition mode is independent of instrument configuration or settings.

RT < 100 ns ≥ ERT

Syntax

```
[SENSe:]SWEep:TINTerval <time_interval>
[SENSe:]SWEep:TINTerval?
```

Parameters

<time_interval> (seconds)[1]	Query Response
<NRf>	<NR3>
200E–12 (TVS625, TVS645 only)	200.0E–12
400E–12 (TVS625, TVS645 only)	400.0E–12
1E–9	1.0E–9
2E–9	2.0E–9
4E–9	4.0E–9
10E–9	10.0E–9
20E–9	20.0E–9
40E–9	40.0E–9
100E–9	100.0E–9
.	.
.	.
.	.
200E–3	200.0E–3
MINimum	depends on model
MAXimum	200.0E–3

1 **The default multiplier for <time_interval> is S for seconds. You can also use the multipliers MS for milliseconds, US for microseconds, NS for nanoseconds, and PS for picoseconds.**

www.valuetronics.com

Reset Value 1.0E–9

Errors and Events Execution Error –222, “Data out of range”
Attempted to set time interval to an illegal value.

Execution Error –241, “Hardware missing”
Attempted to set time interval to 200E–12 or 400E–12 when the instrument is a TVS621 or TVS641.

Dependencies Changing SWEep:TINTerval changes SWEep:TIME. The relation is SWEep:TINTerval = (SWEep:TIME / SWEep:POINts).

Changing SWEep:TIMTerval to < 100 ns changes SWEep:POINts to 15,000 if previously set to 30,000. The acquisition mode changes from extended real time to real time mode below 100 ns/sample.

Examples Command: `SWE:TINT 1E-6`

Query: `SWE:TINT?`

Response: `1.0E-6`

Related Commands SWEep:POINts
SWEep:TIME

www.valuetronics.com

SYSTem Subsystem

This section describes each command and query in the SYSTem subsystem. These commands program utility functions and return version information about the waveform analyzer.

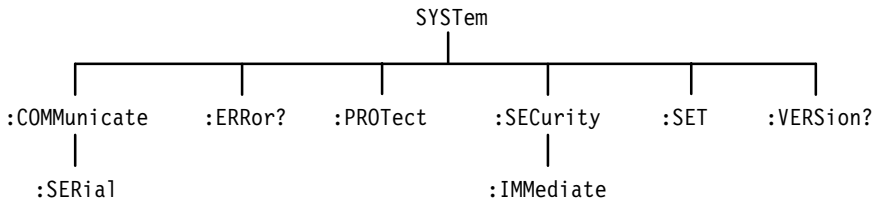


Figure 3–27: SYSTem Subsystem Hierarchy

SYSTem:COMMunicate:SERial:BAUD
SYSTem:COMMunicate:SERial:BAUD?

Sets or queries the baud rate of the front panel RS-232 port. The baud rate is the data transfer rate in bits per second for a serial transfer. Both transmit and receive baud rates are set by this command. Refer to Table 1–1 on page 1–14 for the factory setting.

When you set :SERial:BAUD, the new value is stored in non-volatile RAM and is recalled at power up. However, the baud rate is not saved with stored settings (*LRN) and it is not reset to a default value by *RST.

Syntax

```
SYSTem:COMMunicate:SERial:BAUD <baud_rate>
SYSTem:COMMunicate:SERial:BAUD?
```

Parameters

<baud_rate>	Query Response
<NRf>	<NR1>
300	300
600	600
1200	1200
2400	2400
4800	4800
9600	9600
19200	19200
MINimum	300
MAXimum	19200

www.valuetronics.com

Reset Value Not applicable

Errors and Events Execution Error –222, "Data out of range"
Attempted to set the baud rate to an illegal value.

Dependencies None

Examples Command: `SYST:COMM:SER:BAUD 4800`

Query: `SYST:COMM:SER:BAUD?`

Response: `4800`

Related Commands SYSTem:COMMunicate:SERial:PARity
SYSTem:COMMunicate:SERial:SBITs

SYSTem:COMMunicate:SERial:CONTrol:DCD
SYSTem:COMMunicate:SERial:CONTrol:DCD?

Sets or queries whether the instrument is sensitive to the front panel RS-232 DCD line. The setting ON specifies that the port ignores data unless the DCD line is high. This mode is useful when you have a modem connected. The setting OFF means the DCD line is ignored. Refer to Table 1–1 on page 1–14 for the factory setting.

When you set :SERial:CONTrol:DCD, the new value is stored in non-volatile RAM and is recalled at power up. However, the control mode is not saved with stored settings (*LRN) and it is not reset to a default value by *RST.

Syntax

```
SYSTem:COMMunicate:SERial:CONTrol:DCD <boolean>
SYSTem:COMMunicate:SERial:CONTrol:DCD?
```

Parameters

<boolean>	Query Response
<NRf> 1 or ON 0 or OFF	<NR1> 1 0

Reset Value Not applicable

Errors and Events None

www.valuetronics.com

Dependencies None

Examples Command: `SYST:COMM:SER:CONT:DCD ON`

Query: `SYST:COMM:SER:CONT:DCD?`

Response: 1

Related Commands SYSTem:COMMunicate:SERial:CONTrol:RTS
SYSTem:COMMunicate:SERial:PACE

SYSTem:COMMunicate:SERial:CONTrol:RTS
SYSTem:COMMunicate:SERial:CONTrol:RTS?

Sets or queries the operation of the front panel RS-232 RTS and CTS lines. The setting OFF sets the outgoing RTS line low and ignores the incoming CTS line. The setting ON sets the outgoing RTS line high and ignores the incoming CTS line. The setting IBFull or RFR enables the two lines for hardware flow control. Refer to Table 1–1 on page 1–14 for the factory setting.

The waveform analyzer does not support DTR/DSR flow control. The instrument holds the DTR line high and ignores the DSR line.

When you set :SERial:CONTrol:RTS, the new value is stored in non-volatile RAM and is recalled at power up. However, the control mode is not saved with stored settings (*LRN) and it is not reset to a default value by *RST.

Syntax

```
SYSTem:COMMunicate:SERial:CONTrol:RTS <flow_control>
SYSTem:COMMunicate:SERial:CONTrol:RTS?
```

Parameters

<flow_control>	Query Response
OFF	OFF
ON	ON
IBFull | RFR	IBF

Reset Value Not applicable

Errors and Events Execution Error –141, “Invalid character data”
Attempted to set the hardware flow control to an illegal value.

Dependencies None

www.valuetronics.com

Examples

Command: `SYST:COMM:SER:CONT:RTS IBF`

Query: `SYST:COMM:SER:CONT:RTS?`

Response: `IBF`

Related Commands

SYSTem:COMMunicate:SERial:CONTrol:DCD
SYSTem:COMMunicate:SERial:PACE

SYSTem:COMMunicate:SERial:ECHO
SYSTem:COMMunicate:SERial:ECHO?

Sets or queries whether incoming characters are echoed back to the front panel RS-232 port. This mode is useful when you are typing commands at a connected terminal because you can see what characters the instrument received. Refer to Table 1–1 on page 1–14 for the factory setting.

When you set :SERial:ECHO, the new value is stored in non-volatile RAM and is recalled at power up. However, the echo mode is not saved with stored settings (*LRN) and it is not reset to a default value by *RST.

Syntax

```
SYSTem:COMMunicate:SERial:ECHO <boolean>
SYSTem:COMMunicate:SERial:ECHO?
```

Parameters

<boolean>	Query Response
<NRf> 1 or ON 0 or OFF	<NR1> 1 0

Reset Value

Not applicable

Errors and Events

None

Dependencies

None

Examples

Command: `SYST:COMM:SER:ECHO OFF`

Query: `SYST:COMM:SER:ECHO?`

Response: `0`

www.valuetronics.com

Related Commands SYSTem:COMMunicate:SERial:PRESet:RAW
SYSTem:COMMunicate:SERial:PRESet:TERMinal

SYSTem:COMMunicate:SERial:ERESponse
SYSTem:COMMunicate:SERial:ERESponse?

Sets or queries whether error messages are automatically returned to the front panel RS-232 port. When OFF, error messages are stored in the RS-232 Error Queue. The OFF mode is appropriate when you have a computer connected to the RS-232 connector. When ON, messages are immediately displayed. The ON mode is most appropriate when a simple display terminal is connected to the front panel RS-232. Refer to Table 1–1 on page 1–14 for the factory setting.

When you set :SERial:ERESponse, the new value is stored in non-volatile RAM and is recalled at power up. However, the error response mode is not saved with stored settings (*LRN) and it is not reset to a default value by *RST.

Syntax

```
SYSTem:COMMunicate:SERial:ERESponse <boolean>
SYSTem:COMMunicate:SERial:ERESponse?
```

Parameters

<boolean>	Query Response
<NRf>	<NR1>
1 or ON	1
0 or OFF	0

Reset Value Not applicable

Errors and Events None

Dependencies None

Examples

Command: `SYST:COMM:SER:ERES OFF`

Query: `SYST:COMM:SER:ERES?`

Response: `0`

Related Commands SYSTem:COMMunicate:SERial:PRESet:RAW
SYSTem:COMMunicate:SERial:PRESet:TERMinal

www.valuetronics.com

SYSTem:COMMunicate:SERial:LBUFfer
SYSTem:COMMunicate:SERial:LBUFfer?

Sets or queries the state of the character buffer for the front panel RS-232 port. When ON, all input characters are queued until a newline (^J) or return (^M) character is received; then the buffer contents are released to the command parser. The following line editing commands let you modify the input before entering a newline or return character:

- ^H (backspace),
- ^U (line-delete)
- ^D (end-of-file).

When OFF, characters are immediately passed to the parser, line editing is not possible, and you must enter a newline (^J) character to terminate a command line. When transferring binary data, always set the line buffer to OFF. When buffering is on, return characters (^M) are converted to newlines (^J) and outgoing newlines (^J) are converted to return-newlines (^M^J). The conversion of termination characters causes significant problems when transferring binary data.

When you set :SERial:LBUFfer, the new value is stored in non-volatile RAM and is recalled at power up. However, the buffer mode is not saved with stored settings (*LRN) and it is not reset to a default value by *RST. Refer to Table 1–1 on page 1–14 for the factory setting.

Syntax

```
SYSTem:COMMunicate:SERial:LBUFfer <boolean>
SYSTem:COMMunicate:SERial:LBUFfer?
```

Parameters

<boolean>	Query Response
<NRf>	<NR1>
1 or ON	1
0 or OFF	0

Reset Value Not applicable

Errors and Events None

Dependencies None

Examples Command: `SYST:COMM:SER:LBUF OFF`

www.valuetronics.com

Query: SYST:COMM:SER:LBUF?

Response: 0

Related Commands SYSTem:COMMunicate:SERial:PRESet:RAW
SYSTem:COMMunicate:SERial:PRESet:TERMinal

SYSTem:COMMunicate:SERial:PACE
SYSTem:COMMunicate:SERial:PACE?

Sets or queries whether software flow control (XON/XOFF) is enabled for the front panel RS-232 port. Both transmit and receive values are set by this command. Refer to Table 1–1 on page 1–14 for the factory setting.

When you set :SERial:PACE, the new value is stored in non-volatile RAM and is recalled at power up. However, the flow control mode is not saved with stored settings (*LRN) and it is not reset to a default value by *RST.

You should not use software flow control when transferring binary data.

Syntax

```
SYSTem:COMMunicate:SERial:PACE <flow_control>
SYSTem:COMMunicate:SERial:PACE?
```

Parameters

<flow_control>	Query Response
NONE XON	NONE XON

Reset Value Not applicable

Errors and Events Execution Error –141, “Invalid character data”
Attempted to set the software flow control to an illegal value.

Dependencies None

Examples Command: SYST:COMM:SER:PACE XON

Query: SYST:COMM:SER:PACE?

Response: XON

www.valuetronics.com

Related Commands

SYSTem:COMMunicate:SERial:CONTrol:DCD
SYSTem:COMMunicate:SERial:CONTrol:RTS
SYSTem:COMMunicate:SERial:PRESet:RAW
SYSTem:COMMunicate:SERial:PRESet:TERMinal

SYSTem:COMMunicate:SERial:PARity
SYSTem:COMMunicate:SERial:PARity?

Sets or queries the type of parity for the front panel RS-232 port. Parity provides a minimum level of data security by appending one data bit to each transmitted character to achieve an even or odd number of "1" digits. Both transmit and receive values are set by this command. Refer to Table 1–1 on page 1–14 for the factory setting.

When you set :SERial:PARity, the new value is stored in non-volatile RAM and is recalled at power up. However, the parity mode is not saved with stored settings (*LRN) and it is not reset to a default value by *RST.

Syntax

```
SYSTem:COMMunicate:SERial:PARity[:TYPE] <type>
SYSTem:COMMunicate:SERial:PARity[:TYPE]?
```

Parameters

<type> (Select parity type)	Query Response
EVEN	EVEN
NONE	NONE
ODD	ODD

Reset Value

Not applicable

Errors and Events

Execution Error –141, "Invalid character data"
Attempted to set the parity to an illegal value.

Dependencies

None

Examples

Command: `SYST:COMM:SER:PAR EVEN`

Query: `SYST:COMM:SER:PAR?`

Response: `EVEN`

Related Commands

SYSTem:COMMunicate:SERial:BAUD
SYSTem:COMMunicate:SERial:SBITs

www.valuetronics.com

SYSTem:COMMunicate:SERial:PRESet

These commands provide three :SERial settings that configure the RS-232 port parameters for typical transfer modes. The default node sets all RS-232 settings to a known state. :RAW and :TERMinal change only a subset of the RS-232 parameters. The :RAW preset is most appropriate when you connect a computer to the RS-232 port. TERMinal should be used when you connect a display terminal.

The command "T" followed by a newline (^J) will configure the RS-232 port for terminal operation when it is currently in RAW mode. There are no associated queries.

The Table 3–28 specifies the preset values for each PRESet command.

Table 3–28: Effects of :PRESet on Serial Port Parameters

SERial Port Parameter	[:ALL]	:RAW	:TERMinal
BAUD	9600	NC[1]	NC[1]
CONTrol:DCD	OFF	NC[1]	NC[1]
CONTrol:RTS	ON	NC[1]	NC[1]
ECHO	OFF	OFF	ON
ERESponse	OFF	OFF	ON
LBUFfer	OFF	OFF	ON
PACE	NONE	NONE	XON
PARity	NONE	NC[1]	NC[1]
SBITs	1	NC[1]	NC[1]

[1] **The entry NC means the value is not changed.**

Syntax

```
SYSTem:COMMunicate:SERial:PRESet[:ALL]
SYSTem:COMMunicate:SERial:PRESet:RAW
SYSTem:COMMunicate:SERial:PRESet:TERMinal
```

Parameters

Not applicable	Query Response
See Syntax	Not applicable

Reset Value Not applicable

Errors and Events None

www.valuetronics.com

Dependencies None

Examples Command: `SYST:COMM:SER:PRES:TERM`

Related Commands SYSTem:COMMunicate:SERial Commands

SYSTem:COMMunicate:SERial:SBITs
SYSTem:COMMunicate:SERial:SBITs?

Sets or queries the number of stop bits sent with each character transmitted on the front panel RS-232 port. Both transmit and receive values are set by this command. Refer to Table 1–1 on page 1–14 for the factory setting.

When you set :SERial:SBITs, the new value is stored in non-volatile RAM and is recalled at power up. However, the number of stop bits is not saved with stored settings (*LRN) and it is not reset to a default value by *RST.

Syntax

```
SYSTem:COMMunicate:SERial:SBITs <stop_bits>
SYSTem:COMMunicate:SERial:SBITs?
```

Parameters

<stop_bits>	Query Response
<NRf>	<NR1>
1 or MINimum	1
2 or MAXimum	2

Reset Value Not applicable

Errors and Events Execution Error –222, “Data out of range”
Attempted to set the number of stop bits to an illegal value.

Dependencies None

Examples Command: `SYST:COMM:SER:SBIT 2`

Query: `SYST:COMM:SER:SBIT?`

Response: `2`

Related Commands SYSTem:COMMunicate:SERial:BAUD
SYSTem:COMMunicate:SERial:PARity

www.valuetronics.com

SYSTem:ERRor? (Query Only)

This query returns the next entry from the waveform analyzer Status Queue. The event message contains the event number and a text description of the event. Reading an event removes it from the queue. Errors and events are cleared from the Status Queue at power up, upon receipt of the command *CLS, and upon reading the last item from the event queue.

Refer to the Status and Events discussion on page 4–1 for a detailed description of the Status Queue and a complete list of system events.

Syntax

```
SYSTem:ERRor[:NEXT]?
```

Parameters

<event>	Query Response
Not applicable	<NR1>, <string>

Reset Value

Not applicable

Errors and Events

Device Specific Error –350, “Queue Overflow”
The error/event queue overflowed due to the execution errors of other commands. The SYST:ERR? query cannot generate an error.

Dependencies

None

Examples

Query: `SYST:ERR?`

Response: `-221, "Settings conflict"`

Related Commands

STATus:OPERation:QENable:NTRansition
STATus:OPERation:QENable:PTRansition
STATus:QUEStionable:QENable:NTRansition
STATus:QUEStionable:QENable:PTRansition
STATus:PRESet
STATus:SESR:QENable
SYSTem:ERRor:ALL?
*CLS

www.valuetronics.com

SYSTem:ERRor:ALL? (Query Only)

Returns the list of all events stored in the waveform analyzer Status Queue. Each event message contains the event number and a text description of the event. Events are separated by commas. Errors and events are cleared from the Status Queue at power up, upon receipt of the command *CLS, and upon reading the last item from the event queue.

Refer to the Status and Events discussion on page 4–1 for a detailed description of the Status Queue and a complete list of system events.

Syntax

```
SYSTem:ERRor:ALL?
```

Parameters

<event>	Query Response
Not applicable	<NR1>, <string>

Reset Value

Not applicable

Errors and Events

0, “No error”
Attempted to read an event when the queue is empty.

Device Specific Error –350, “Queue Overflow”
The Status Queue has overflowed or run out of memory.

Dependencies

None

Examples

Query:

```
SYST:QUE:ALL?
```

Response:

```
-221, "Settings conflict",-224,"Illegal parameter
value"
```

Related Commands

SYSTem:ERRor?

www.valuetronics.com

SYSTem:ERRor:CODE? (Query Only)

Returns the next event codes stored in the waveform analyzer Status Queue. The returned error code does not include the associated event message. Errors and events are cleared from the Status Queue at power up, upon receipt of the command *CLS, and upon reading the last item from the event queue. To return all error codes use SYSTem:ERRor:CODE:ALL?. Use SYSTem:ERRor? to return an error code and its event message.

Refer to the Status and Events discussion on page 4–1 for a detailed description of the Status Queue and a complete list of system events.

Syntax

```
SYSTem:ERRor:CODE[:NEXT]?
```

Parameters

<code>	Query Response
Not applicable	<NR1>

Reset Value

Not applicable

Errors and Events

0, “No error”
Attempted to read an event when the queue is empty.

Device Specific Error –350, “Queue Overflow”
The Status Queue has overflowed or run out of memory.

Dependencies

None

Examples

Query:

```
SYST:ERR:CODE?
```

Response:

```
-221
```

Related Commands

SYSTem:ERRor?
SYSTem:ERRor:CODE:ALL?

www.valuetronics.com

SYSTem:ERRor:CODE:ALL? (Query Only)

Returns the list of all event codes stored in the waveform analyzer Status Queue. The returned error codes do not include their associated event messages. Event codes are separated by commas. Use SYSTem:ERRor:CODE? to return one error code at a time. Use SYSTem:ERRor:ALL? to return all error codes and event messages.

Refer to the Status and Events discussion on page 4–1 for a detailed description of the Status Queue and a complete list of system events.

Syntax

```
SYSTem:ERRor:CODE:ALL?
```

Parameters

<code>	Query Response
Not applicable	<NR1>

Reset Value

Not applicable

Errors and Events

0, “No error”
Attempted to read an event when the queue is empty.

Device Specific Error –350, “Queue Overflow”
The Status Queue has overflowed or run out of memory.

Dependencies

None

Examples

Query: `SYST:ERR:CODE:ALL?`

Response: `-221,-224`

Related Commands

SYSTem:ERRor?
SYSTem:ERRor:CODE

www.valuetronics.com

SYSTem:ERRor:COUNt? (Query Only)

Returns the number of unread events in the Status Queue. The Status Queue is not modified by this command. Use SYSTem:ERRor? to return the errors and events from the Status Queue. Errors and events are cleared from the Status Queue at power up, upon receipt of the command *CLS, and upon reading the last item from the event queue.

Syntax `SYSTem:ERRor:COUNt?`

Parameters

<count>	Query Response
Not applicable	<NR1>

Reset Value Not applicable

Errors and Events None

Dependencies None

Examples Query: `SYST:ERR:COUN?`

Response: 3

Related Commands SYSTem:ERRor:ALL?
SYSTem:ERRor:CODE:ALL?

SYSTem:PROTect
SYSTem:PROTect?

Sets or queries whether protection for a group of sensitive instrument commands is enabled. When protection is enabled (set ON, the default), the following settings cannot be changed without first setting SYSTem:PROTect to OFF:

- INPut:PROTection:STATe
- SYSTem:SECurity:IMMediate
- TEST:LOG:CLEar
- *PUD

www.valuetronics.com

SYSTem:PROTect is enabled, set to ON, when you reset (*RST) or power up the instrument.

Syntax

```
SYSTem:PROTect
SYSTem:PROTect?
```

Parameters

<boolean>	Query Response
<NRf> 1 or ON 0 or OFF	<NR1> 1 0

Reset Value 1

Errors and Events None

Dependencies None

Examples Command: `SYST:PROT ON`

Query: `SYST:PROT?`

Response: 1

Related Commands INPut:PROTection:STATe
SYSTem:SECurity:IMMediate
TEST:LOG:CLEar
*PUD

SYSTem:SECurity:IMMediate

Immediately destroys all measurement data and stored instrument settings. The instrument writes over measurement data in acquisition memory three times with a fixed bit pattern. Current settings are initialized to their *RST or factory default values but critical calibration constants are retained. Set SYSTem:PROTect to OFF before using the command SYSTem:SECurity:IMMediate.

Syntax

```
SYSTem:SECurity:IMMediate
```

Parameters None

www.valuetronics.com

Reset Value Not applicable

Errors and Events Execution Error –203, "Command protected"
Attempted to execute this command without first removing the protection with the command SYSTem:PROTect.

Dependencies All measurement data and instrument settings are lost upon execution.

Examples Command: `SYST:SEC:IMM`

Related Commands SYSTem:PASSword
*RST

SYSTem:SET
SYSTem:SET?

Sets or queries the internal state of the instrument as a binary data block. The format of the binary data is instrument specific and should not be modified if you intend to use it to set the instrument state. Binary settings may not be compatible with other models or firmware versions of the TVS600 Series Waveform Analyzers. Use the *LRN? query to get an ASCII version of the instrument settings.

Syntax
```
SYSTem:SET <settings>
SYSTem:SET?
```

Parameters

<settings>	Query Response
<def_len_block> or <indef_len_block>	<def_len_block>

Reset Value Not applicable

Errors and Events Execution Error –233, "Invalid version"
Attempted to load a block of binary settings that have a version number different from that of the instrument.

Dependencies None

www.valuetronics.com

Examples

Command: `SYST:SET #45000 . . .`

Query: `SYST:SET?`

Response: `#45000 . . .`

Related Commands

*LRN?
*RCL
*SAV

SYSTem:VERSion? (Query Only)

Returns the SCPI version supported by the waveform analyzer. You will need this version number if you call Tektronix for product or application support.

Syntax

`SYSTem:VERSion?`

Parameters

<version>	Query Response
Not applicable	<NR2> 1995.0

Reset Value

Not applicable

Errors and Events

None

Dependencies

None

Examples

Query: `SYST:VERS?`

Response: `1995.0`

Related Commands

*IDN?

www.valuetronics.com

TEST Subsystem

This section describes the commands in the TEST subsystem. These commands execute the internal self-tests of the waveform analyzer module and return the pass or fail status.

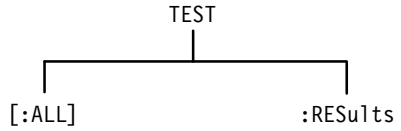


Figure 3–28: TEST Subsystem Hierarchy

TEST
TEST?

Executes all internal self-tests once. If a failure occurs, the test will stop. The query form returns the number of the first test that fails or zero if there were no failures. The command form executes the same tests but does not return a result value.

Use the command *WAI, *OPC, or *OPC? to delay execution of the TEST? command until the self test completes. Completion of the tests clears the TESTing flag in the Operation Status register.

Upon a failure, you should refer the instrument to your Service provider for repair. For a listing of the failure codes, refer to the *TVS600 Service Manual*.

Syntax

```
TEST[:ALL]
TEST[:ALL]?
```

Parameters

None	Query Response	
Not applicable	<NR1>	
	0	(no failure)
	1000 – 1999	(self test failure)
	2000 – 2999	(calibration failure)

Reset Value Not applicable

Errors and Events None

www.valuetronics.com

Dependencies

Examples

Command: `TEST`

Query: `TEST?`

Response: `0`

Related Commands

TEST:RESults?
TEST:RESults:VERBose?

TEST:RESults? (Query Only)

Returns the failure code for the last self-test command that was executed. The returned failure code identifies the first test that failed. See the Parameters table for the possible responses. Use the command *WAI, *OPC, or *OPC? to delay execution of the TEST:RESults? command until the self test completes. Completion of the tests clears the TESTing flag in the Operation Status register.

Syntax

`TEST:RESults[:CODE]?`

Parameters

<failure_code>	Query Response	
Not applicable	<NR1>	
	–1	(test in progress)
	0	(no failure)
	1000 – 1999	(Self Test failure)
	2000 – 2999	(Calibration failure)

Reset Value

Not applicable

Errors and Events

None

Dependencies

None

Examples

Query: `TEST:RES?`

Response: `0`

Related Commands

TEST
TEST:RESults:VERBose?

www.valuetronics.com

TEST:RESults:VERBose? (Query Only)

Returns a failure code as a string describing of the last executed self-test command and the results of the self test. The returned failure code describes the first test that failed. See the Parameters table for the possible responses. Use the command *WAI, *OPC, or *OPC? to delay execution of the TEST:RESults? command until the self test completes. Completion of the tests clears the TESTing flag in the Operation Status register.

Syntax

```
TEST:RESults:VERBose?
```

Parameters

<test_results>	Query Response
Not applicable	<string> error number: <NR1> –1 (test in progress) 0 (no failure) 1000 – 1999 (Self Test failure) 2000 – 2999 (Calibration failure) verbose message: (error specific)

Reset Value

Not applicable

Errors and Events

None

Dependencies

None

Examples

Query: `TEST:RES:VERB?`

Response: `1001,"TST_1 max=444 min=222 . . ."`

Related Commands

TEST
TEST:RESults?

www.valuetronics.com

TRACe Subsystem

This section describes the commands in the TRACe subsystem which store and retrieve acquisition and measurement results. The SENSE subsystem command DATA? provides another way to transfer acquisition records and CALC:DATA? provides an alternate way to transfer the results of calculations.

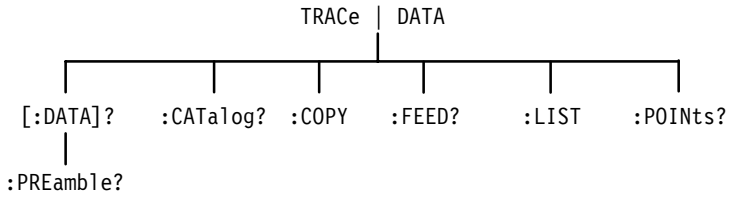


Figure 3–29: TRACe Subsystem Hierarchy

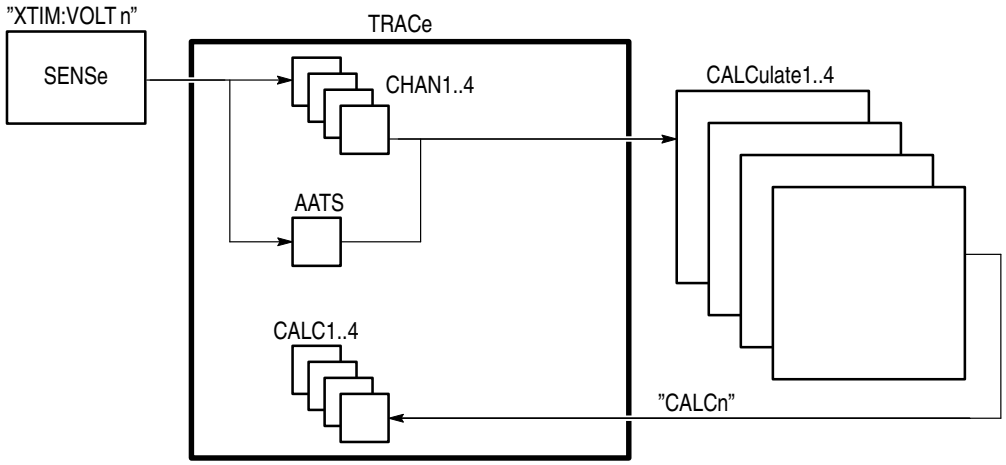


Figure 3–30: Functions of the TRACe Subsystem

www.valuetronics.com

TRACe? (Query Only)

Transfers acquisition records or measurement results from the waveform analyzer to your VXIbus controller. You set the data format for acquisition records and measurement results using the FORMat subsystem described on page 3–101. The query does not return data until pending acquisitions or calculations complete. Hence, the synchronizing commands *WAI, *OPC, and *OPC? are not required.

The [SENSe:]DATA and CALC:DATA commands provide equivalent functions to the commands in the TRACe subsystem. For example, [SENSe:]DATA? "XTIM:VOLT 1" is equivalent to TRAC? CHAN1 and CALC1:DATA? is equivalent to TRAC? CALC1. The feed controls for CHAN1 to CHAN4 and CALC1 to CALC4 are always enabled.

The source AATS is the Auto Advance Time Stamp record. The record contains a sequence of times in seconds, from $t_{:STARt}$ to tn. The value $t_{:STARt}$ is the record specified with the AADV:REC:STAR command. Set the format of the AATS record to ASCII or 32 bit REAL numbers with the command FORM:TRAC:AATS. FDC transfers are binary only. The feed control for AATS is always enabled. When Auto Advance is on, use the commands AADVance:RECord:STARt and :COUNt to specify the records to return.

For more information, refer to the *Data Transfer Format* discussion on page 2–55.

Syntax

```
TRACe[:DATA]? <source>
DATA:DATA? <source>
```

Parameters

<source>	Query Response
CHAN1 CHAN2 CHAN3 CHAN4 AATS CALC1 CALC2 CALC3 CALC4	Not Applicable

<data>	Query Response
Not Applicable	Defined by FORMat:TRACe

Reset Value

Not applicable

www.valuetronics.com

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to query data from an invalid trace name.

Execution Error –241, “Hardware missing”
Attempted to query data from CHAN3 or CHAN4 when the instrument is configured with two channels.

Execution Error –230, “Data corrupt or stale.”
Attempted to query data that is invalid, incomplete or stale.

Dependencies

None

Examples

Query: `TRAC? CHAN1`

Response: `<arb_block_data>`

Related Commands

[SENSe:]DATA?
[SENSe:]DATA:PREamble?
CALCulate:DATA?
CALCulate:DATA:PREamble?
TRACe:PREamble?

TRACe:PREamble? (Query Only)

Transfers the data preamble for acquisition records or measurement results from the waveform analyzer to your VXIbus controller. The data preamble includes scaling and length information for the associated data record. The query does not return data until pending acquisitions or calculations complete. Hence, the synchronizing commands *WAI, *OPC, and *OPC? are not required.

The [SENSe:]DATA:PRE and CALC:DATA:PRE commands provide equivalent functions to the command TRACe:PRE. For example, [SENSe:]DATA:PRE? “XTIM:VOLT 1” is equivalent to TRAC? CHAN1 and CALC1:DATA:PRE? is equivalent to TRAC? CALC1. The feed controls for CHAN1 to CHAN4 and CALC1 to CALC4 are always enabled.

The source AATS is the Auto Advance Time Stamp record which is not supported by the SENSe subsystem. The feed control for AATS is always enabled. If Auto Advance is on, use CHAN1 to CHAN4 to specify which preamble to return.

For more information, refer to the Data Transfer Format discussion on page 2–55.

www.valuetronics.com

Syntax

```
TRACe[:DATA]:PREamble? <source>
DATA:DATA:PREamble? <source>
```

Parameters

<source>	Query Response
CHAN1 CHAN2 CHAN3 CHAN4 AATS CALC1 CALC2 CALC3 CALC4	Not Applicable

<preamble>	Query Response
Not Applicable	DIF Expression

Reset Value

Not applicable

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to query the preamble for an invalid trace name.

Execution Error –230, “Data corrupt or stale.”
Attempted to query the preamble of data that is invalid, incomplete or stale.

Execution Error –241, “Hardware missing”
Attempted to query the preamble for CHAN3 or CHAN4 when the instrument is configured with two channels.

Dependencies

None

Examples

Query: `TRAC:PRE? CHAN1`

Response: (DIF Expression)

Related Commands

[SENSe:]DATA?
[SENSe:]DATA:PREamble?
CALCulate:DATA?
CALCulate:DATA:PREamble?
TRACe?

www.valuetronics.com

TRACe:CATalog? (Query Only)

Returns list of the predefined trace names in your waveform analyzer. The trace names are quoted strings separated by commas. CHAN1 to 4 refer to SENSe functions “XTIM:VOLT n”. CALC1 to 4 refer to the results data from the four CALCulate blocks. AATS refers to the Auto Advance timestamp record.

Syntax

```
TRACe:CATalog?
DATA:CATalog?
```

Data Types None

Reset Value Not applicable

Errors and Events None

Dependencies None

Examples

Query: `TRAC:CAT?`

Response: `"CHAN1","CHAN2","CHAN3","CHAN4","AATS","CALC1","CALC2","CALC3","CALC4"`
Note that CHAN3 and CHAN4 are not defined for two channel instruments.

Related Commands

TRACe?
TRACe:PREamble?

www.valuetronics.com

TRACe:COPY

Copies acquisition or measurement data to the outgoing Fast Data Channel (FDC). The CHAN1 to CHAN4 and CALC1 to CALC4 commands refer to the corresponding input channel or CALC block result. LIST includes all traces defined with the command TRACe:LIST. Traces with no data are copied as null blocks.

For information, refer to the Fast Data Channel discussion on page 2–50.

Syntax

```
TRACe:COPY <destination>, <source>
DATA:COPY <destination>, <source>
```

Parameters

<destination>	Query Response
FDC1	Not Applicable

<source>	Query Response
CHAN1 CHAN2 CHAN3 CHAN4 AATS CALC1 CALC2 CALC3 CALC4 LIST	Not Applicable

Reset Value

Not applicable

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to copy data to or from an invalid trace name.

Execution Error –221, “Settings conflict.”
Attempted to transfer data through the outgoing FDC channel when the channel is not active.

Execution Error –241, “Hardware missing”
Attempted to copy data from CHAN3 or CHAN4 when the instrument is configured with two channels.

Dependencies

None

www.valuetronics.com

Examples Command: `TRAC:COPY FDC1, CHAN1`

Related Commands MEMory:COPY

TRACe:FEED? (Query Only)

Returns the source of data for the pre-defined trace names.

Syntax

```
TRACe:FEED? <source>
DATA:FEED? <source>
```

Parameters

<source>	Query Response
CHAN1	“XTIM:VOLT 1”
CHAN2	“XTIM:VOLT 2”
CHAN3	“XTIM:VOLT 3”
CHAN4	“XTIM:VOLT 4”
AATS	”AADV”
CALC1	“CALC1”
CALC2	“CALC2”
CALC3	“CALC3”
CALC4	“CALC4”

Reset Value Not applicable

Errors and Events Execution Error –141, “Invalid character data”
Attempted to query the feed for an invalid trace name.

Dependencies None

Examples Query: `TRAC:FEED? CHAN1`

Response: `"XTIM:VOLT 1"`

Related Commands TRACe?
TRACe:PREamble?

www.valuetronics.com

TRACe:LIST
TRACe:LIST?

Sets or queries the list of traces to transfer through the VXI Fast Data Channel at the completion of each INITiate command or when LIST is specified as the source for the TRACe:COPY command.

If a trace contains no trace data, a null block is copied for that trace and no error is generated.

Syntax

```
TRACe:LIST <list>
TRACe:LIST?
DATA:LIST <list>
DATA:LIST?
```

Parameters

<list>	Query Response
CHAN1	CHAN1
CHAN2	CHAN2
CHAN3	CHAN3
CHAN4	CHAN4
AATS	AATS
CALC1	CALC1
CALC2	CALC2
CALC3	CALC3
CALC4	CALC4
NONE	NONE

Reset Value

NONE

Errors and Events

108 Command Error 108, “Parameter not allowed”
Attempted to assign more than 10 traces to the list.

Execution Error –141, “Invalid character data”
Attempted to include an invalid trace name in the list.

Dependencies

None

Examples

Command: `TRAC:LIST CHAN1,CALC1`

Query: `TRAC:LIST?`

Response: `CHAN1,CALC1`

www.valuetronics.com

Related Commands TRACE:COPY

TRACe:POINts? (Query Only)

Returns the number of sample points in the acquisition record or CALCulate block record. The record length is set differently for each source:

- CHAN<n> is set by SWEep:POINts.
- AATS (Auto-advance Time Stamp) is set by AADVance:COUNt.
- CALC<n> depends on the record length of the source waveform(s) and the measurement or calculation selected.
- Auto Advance is set by AADVance:RECord:STARt and AADVance:RECord:COUNt.

If the record contains no data, a length of zero is returned.

Syntax

```
TRACe:POINts? <source>
DATA:POINts? <source>
```

Parameters

<source>	Query Response
CHAN1 CHAN2 CHAN3 CHAN4 AATS CALC1 CALC2 CALC3 CALC4	<NR1>

Reset Value Not applicable

Errors and Events Execution Error –141, "Invalid character data"
Attempted to query the record length of an invalid trace name.

Execution Error –241, "Hardware missing"
Attempted to query the record length of CHAN3 or CHAN4 when the instrument is configured with two channels.

Dependencies None

www.valuetronics.com

Examples

Query: `TRAC:POIN? CHAN1`

Response: `1024`

Related Commands

TRACe?
TRACE:PREamble?
SWEep:POINts

www.valuetronics.com

TRIGger[:A] Subsystem

This section describes the commands in the TRIGger[:A] subsystem. These commands operate with the ARM, INITiate, TRIGger:B and ABORt subsystems to trigger acquisitions. The defined alias for the SCPI trigger :SEQuence[1] is A.

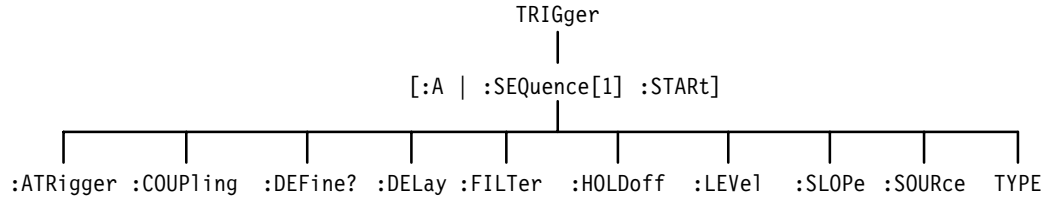


Figure 3–31: TRIGger:A (SCPI SEQuence1) Subsystem Hierarchy

TRIGger:ATRigger
TRIGger:ATRigger?

Sets or queries whether to generate an automatic trigger when the defined trigger does not occur within 500 ms. You should set TRIGger:ATRigger to OFF when acquiring signals of less than 2 Hz to avoid spurious automatic triggers. Automatic triggering works with all trigger sources.

Syntax

```
TRIGger[:A]:ATRigger[:STATe] <boolean>
TRIGger[:A]:ATRigger[:STATe]?
```

Parameters

<boolean>	Query Response
<NRf>	<NR1>
1 or ON	1
0 or OFF	0

Reset Value 0

Errors and Events None

Dependencies None

Examples Command: `TRIG:ATR ON`

www.valuetronics.com

Query: `TRIG:ATR?`

Response: 1

Related Commands TRIGger:SOURce

TRIGger:COUPling
TRIGger:COUPling?

Sets or queries whether the source of the A trigger is AC or DC coupled. AC coupling removes any DC component from the signal. DC coupling passes all frequencies equally. This command is effective only when you set TRIGger:SOURce to an INTernal source. Note that the external trigger is limited to DC coupling only.

You may want to review the command TRIGger:COUPling:<preset> which lets you set trigger coupling and filtering with one command. However, that command does not adhere to the SCPI standard.

Syntax

```
TRIGger[:A]:COUPling <trigger_coupling>
TRIGger[:A]:COUPling?
```

Parameters

<trigger_coupling>	Query Response
AC DC	AC DC

Reset Value DC

Errors and Events None

Dependencies Changing trigger coupling from DC to AC sets TRIGger:FILTer[:LPASs] to OFF and may cause TRIGger:LEVel to change if the level is out of range for AC coupling.Changing trigger coupling from AC to DC sets TRIGger:FILTer:HPASs to OFF.

When TRIGger:A and TRIGger:B share the same SOURce, changing COUPling for one will change COUPling for the other.

Examples Command: `TRIG:COUP AC`

Query: `TRIG:COUP?`

www.valuetronics.com

Response: AC

Related Commands

TRIGger:COUPling:<preset>
TRIGger:FILTer[:LPASs]
TRIGger:FILTer:HPASs
TRIGger:FILTer:NREJect
TRIGger:SOURce

TRIGger:COUPling:<preset>

Sets trigger coupling and filtering with one command. It should not be used in applications that must work on a variety of SCPI instruments. For a description of how this command affects the four underlying SCPI commands, see *Dependencies* on this page.

This command is effective only when you set TRIGger:SOURce to INTernal. Note that the external trigger provides DC coupling only.

Syntax

```
TRIGger[:A]:COUPling:AC
TRIGger[:A]:COUPling:ACNReject
TRIGger[:A]:COUPling:DC
TRIGger[:A]:COUPling:DCNReject
TRIGger[:A]:COUPling:HFReject
TRIGger[:A]:COUPling:LFReject
```

Parameters

None

Reset Value

Not applicable

Errors and Events

None

Dependencies

The following table describes the interactions between TRIGger:COUPling:<preset> and its four underlying SCPI commands.

<preset>	COUPling	FILT[:LPASs]	FILT:HPAS	FILT:NREJect
AC	AC	OFF	OFF	OFF
ACNReject	AC	OFF	OFF	ON
DC	DC	OFF	OFF	OFF
DCNReject	DC	OFF	OFF	ON

www.valuetronics.com

<preset>	COUPling	FILT[:LPASs]	FILT:HPAS	FILT:NREJect
HFReject	DC	ON	OFF	OFF
LFReject	AC	OFF	ON	OFF

Changing trigger coupling from DC to AC may cause TRIGger:LEVel to change if level is out of range for AC coupling.

When TRIGger:A and TRIGger:B share the same SOURce, changing COUPling for one will change COUPling for the other.

Examples

Command: `TRIG:COUP:ACNR`

Related Commands

TRIGger:COUPling
TRIGger:FILTer[:LPASs]
TRIGger:FILTer:HPASs
TRIGger:FILTer:NREJect

TRIGger:DEFine? (Query Only)

This query returns the predefined SEQuence1 alias, A. ARM:DEFine? and TRIGger:DEFine? return the same information.

Syntax

`TRIGger[:SEQuence[1]]:DEFine?`

Parameters

<sequence_alias>	Query Response
Not applicable	<string> “A”

Reset Value

“A”

Errors and Events

None

Dependencies

None

Examples

Query: `TRIG:DEF?`

Response: `"A"`

www.valuetronics.com

Related Commands

ARM:DEFine?
TRIGger:SEQuence2:DEFine?

TRIGger:DELay
TRIGger:DELay?

Sets or queries the trigger delay for the Trigger A circuit. Trigger delay sets the time after the trigger event to start acquisition. TRIGger:DELay is always positive which delays the start of acquisition a specified time after the trigger event. Use SWEep:OFFSet:TIME to acquire pretrigger data. The mininum real delay is 16 ns. Setting TRIGger:DELay to 0 indicates no delay and bypasses the delay counter.

Refer to the *Acquisition* discussion on page 2–9 and the *Triggering* discussion on page 2–21.

Syntax

```
TRIGger[:A]:DELay <delay_time>
TRIGger[:A]:DELay?
```

Parameters

<delay_time>[1]	Query Response
<NRf> 16 ns ≤ N ≤ 250 s (in 4 ns steps)	<NR3>
MINimum	0.0E+0
MAXimum	250.0E+0

1 **The default for the parameter <delay_time> is seconds (S). You can also use the multipliers MS for milliseconds, US for microseconds, and NS for nanoseconds.**

Reset Value

0.0E+0

Errors and Events

Execution Error –222, "Data out of range"
Attempted to set trigger delay to an illegal value.

Dependencies

Setting TRIGger:DELay to a value greater than 0 will cause TRIGger:B:DELay to be set to 0. Only one trigger delay may be active.

Setting TRIGger:DELay to a value greater than 0 when TRIGger:B:SOURce is set to INTernal<1234> or EXTernal, will set TRIGger:B:ECOunt to 1.

Examples

Command: `TRIG:DEL 10E-6`

Query: `TRIG:DEL?`

www.valuetronics.com

Response: 10.0E-6

Related Commands

SWEep:OFFSet:TIME
TRIGger:B:DELay
TRIGger:B:ECOunt

TRIGger:FILTer[:LPASs]
TRIGger:FILTer[:LPASs]?

Sets or queries the state of the 50 kHz low pass trigger filter. Components of the trigger signal above 50 kHz are attenuated when the LPASs filter is on. It may only be used when DC coupled and with TRIGger:SOURce set to INTernal.

The INPut:FILTer is separate from the trigger filter and is located before the trigger pickoff in the signal path.

Syntax

```
TRIGger[:A]:FILTer[:LPASs][:STATe] <boolean>
TRIGger[:A]:FILTer[:LPASs][:STATe]?
```

Parameters

<boolean>	Query Response
<NRf>	<NF1>
N ≠ 0 or ON	1
0 or OFF	0

Reset Value

0

Errors and Events

None

Dependencies

Enabling the low pass filter will set TRIGger:COUPling to DC and set TRIGger:FILTer:NREJect to OFF. When TRIGger:A and TRIGger:B share the same SOURce, changing FILTer for one will change FILTer for the other.

Examples

Command: TRIG:FILT ON

Query: TRIG:FILT?

Response: 1

www.valuetronics.com

Related Commands

TRIGger:COUPling
TRIGger:FILTer:HPASs
INPut:FILTer

TRIGger:FILTer:HPASs
TRIGger:FILTer:HPASs?

Sets or queries the state of the 50 kHz high pass trigger filter. The HPASs filter attenuates components of the trigger signal below 50 kHz. HPASs can only be used when AC coupled and with TRIGger:SOURce set to INTernal.

The INPut:FILTer is separate from the trigger filter and is located before the trigger pickoff in the signal path.

Syntax

```
TRIGger[:A]:FILTer:HPASs[:STATe] <boolean>
TRIGger[:A]:FILTer:HPASs[:STATe]?
```

Parameters

<boolean>	Query Response
<NRf> N ≠ 0 or ON 0 or OFF	<NR1> 1 0

Reset Value

0

Errors and Events

None

Dependencies

Enabling the high pass filter will set TRIGger:COUPling to AC and set TRIGger:FILTer:NREJect to OFF. When TRIGger:A and TRIGger:B share the same SOURce, changing FILTer for one will change FILTer for the other.

Examples

Command: `TRIG:FILT:HPAS ON`

Query: `TRIG:FILT:HPAS?`

Response: 1

Related Commands

TRIGger:COUPling
TRIGger:FILTer[:LPASs]
INPut:FILTer

www.valuetronics.com

TRIGger:FILTer:NREJect
TRIGger:FILTer:NREJect?

Sets or queries whether or not the noise reject filter is enabled. This filter provides a means of rejecting noise on the trigger signal. Only one trigger filter (i.e., LPAS, HPAS, or NREJ) may be enabled at a time. This command is effective only when you set TRIGger:SOURce to INTernal.

Syntax

```
TRIGger[:A]:FILTer:NREJect <boolean>
TRIGger[:A]:FILTer:NREJect?
```

Parameters

<boolean>	Query Response
<NRf>	<NR1>
1 or ON	1
0 or OFF	0

Reset Value

0

Errors and Events

none

Dependencies

When TRIGger:A:SOURce and TRIGger:B:SOURce use the same signal, changing TRIG:FILT:NREJ also changes TRIG:B:FILT:NREJ.

Examples

Command: `TRIG:FILT:NREJ ON`

Query: `TRIG:FILT:NREJ?`

Response: 1

Related Commands

TRIGger:LEVel

www.valuetronics.com

TRIGger:HOLDoff:TIME
TRIGger:HOLDoff:TIME?

Sets or queries the trigger holdoff time. Trigger holdoff determines how long after one trigger event the event detector will ignore valid triggers. You can use holdoff to trigger at a particular point in a recurring sequence of pulses.

Syntax

```
TRIGger[:A]:HOLDoff:TIME <holdoff_time>
TRIGger[:A]:HOLDoff:TIME?
```

Parameters

<holdoff_time>[1]	Query Response
<NRf> 250 ns ≤ N ≤ 12 s (in 8 ns steps) MINimum MAXimum	<NR3> 250.0E–9 12.0E+0

[1] **The default for the parameter <holdoff_time> is seconds (S). You can also use the multipliers MS for milliseconds, US for microseconds, and NS for nanoseconds.**

Reset Value

250.0E–9

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set holdoff time to an illegal value.

Dependencies

None

Examples

Command: `TRIG:HOLD:TIME 10E-6`

Query: `TRIG:HOLD:TIME?`

Response: `10.0E-6`

Related Commands

None

www.valuetronics.com

TRIGger:LEVel
TRIGger:LEVel?

Sets or queries the trigger level for the TRIGger[:A] subsystem. The trigger level is a specific voltage through which the trigger signal must pass to be recognized as a trigger event and start acquisition. The trigger level should be within the range of the signal in order to guarantee a trigger event. TRIGger:LEVel is effective only when you set TRIGger:SOURce to INTernal or EXTernal. Note that the front panel TRIGD LED lights briefly when a trigger event occurs.

Attaching a probe modifies the minimum and maximum values as it does the setting for VOLTage:RANGe:PTPeak. Multiply the minimum and maximum limits by the attenuation factor of the probe to determine the new maximum and minimum values.

If you experience or expect an unstable trigger point due to noise on the trigger signal, use the noise reject feature of the trigger circuit to reduce the affects of noise. With the command TRIGger:FILTer:NREJect you can reduce the effects of noise on the internal trigger sources.

Syntax

```
TRIGger[:A]:LEVel <trigger_level>
TRIGger[:A]:LEVel?
```

Parameters

<trigger_level>[1]	Query Response
<NRf> External: –1.0 ≤ N ≤ 1.0 (2 mV steps) MINimum MAXimum	<NR3> –1.0E+0 1.0E+0
Internal DC Coupled: absolute maximum[2]: –200.0 ≤ N ≤ 200.0 limited by ± VOLT:RANG:PTP steps 0.002 * VOLT:RANG:PTP MINimum[2] MAXimum[2]	 VOLT:RANG:OFFS ± VOLT:RANG:PTP VOLT:RANG:OFFS – VOLT:RANG:PTP VOLT:RANG:OFFS + VOLT:RANG:PTP
Internal AC Coupled: absolute maximum[2]: –100.0 ≤ N ≤ 100.0 limited by ± VOLT:RANG:PTP steps 0.002 * VOLT:RANG:PTP MINimum[2] MAXimum[2]	 ± VOLT:RANG:PTP – VOLT:RANG:PTP + VOLT:RANG:PTP

1 The default for the parameter <trigger_level> is V for volts. You can also use the multipliers MV for millivolts and UV for microvolts.

2 When you connect a probe, the maximum limits increase just as the vertical range increases for the input. For example, with a 1 V vertical range, connecting a 10X probe increases the vertical range and trigger level range to 10 V.

www.valuetronics.com

Reset Value 0.0E+0

Errors and Events Execution Error –222, "Data out of range"
Attempted to set trigger level to an illegal value.

Dependencies None

Examples Command: `TRIG:LEV 1`

Query: `TRIG:LEV?`

Response: `1.0E+0`

Related Commands TRIGger:SOURce
VOLTage:RANGe:PTPeak
VOLTage:RANGe:OFFSet

TRIGger:SLOPe
TRIGger:SLOPe?

Sets or queries whether triggering occurs on the positive-going or negative-going edge of the trigger source. This command has effect only when TRIGger:SOURce is set to INTernal or EXTernal. The other trigger sources are digital signals.

Syntax

```
TRIGger[:A]:SLOPe <trigger_slope>
TRIGger[:A]:SLOPe?
```

Parameters

<trigger_slope>	Query Response
POSitive	POS
NEGative	NEG

Reset Value POS

Errors and Events Execution Error –141, "Invalid character data"
Attempted to set slope to an illegal value.

Dependencies None

www.valuetronics.com

Examples

Command: `TRIG:SLOP NEGATIVE`

Query: `TRIG:SLOP?`

Response: `NEG`

Related Commands

TRIGger:LEVel

TRIGger:SOURce
TRIGger:SOURce?

Sets or queries the source of the trigger signal for the trigger A circuit. You can specify only one source at a time.

Setting source to INTernal1 selects the signal from INPut1 and [SENSe:]VOLTage1 as the trigger source. The ECLTrg and TTLTrg sources are digital trigger signals from the VXI P2 backplane. The EXTernal signal is from the front-panel BNC connector labeled External Trigger Input.

Many commands in the TRIGger subsystem are dependent on the trigger source you select. For instance, the TRIGger:LEVel setting is ignored when you set the trigger source to TTLTrg0. The dependencies for each command are listed with the command description.

Syntax

```
TRIGger[:A]:SOURce <trigger_source>
TRIGger[:A]:SOURce?
```

Parameters

<trigger_source>	Query Response
ECLTrg0	ECLT0
ECLTrg1	ECLT1
EXTernal	EXT
INTernal1	INT1
INTernal2	INT2
INTernal3[1]	INT3
INTernal4[1]	INT4
TTLTrg0	TTLT0
. . .	. . .
TTLTrg7	TTLT7

[1] **The selections INTernal3 and INTernal4 are not valid with the TVS621 and TVS625.**

Reset Value

INT1

www.valuetronics.com

Errors and Events

Execution Error –141, "Invalid character data"
Attempted to set trigger source to IMMediate or another illegal value.

Execution Error –241, "Hardware missing"
Attempted to set trigger source to INTernal3 or INTernal4 when the instrument has only two channels.

Dependencies

Changing the trigger source from INTernal to EXTernal or from one INTernal source to another INTernal source with a different VOLTage:RANGe setting may change the TRIGger:LEVel setting to accommodate a new vertical range.

Examples

Command: `TRIG:SOUR INTERNAL1`

Query: `TRIG:SOUR?`

Response: `INT1`

Related Commands

ARM:SOURce
TRIGger:B:SOURce

TRIGger:TYPE
TRIGger:TYPE?

Sets or queries the type of triggering to use for the next acquisition. The available types are as follows:

- EDGE specifies the default triggering. A trigger event occurs when a signal passes through a specified voltage level on the specified rising or falling slope. The standard SCPI trigger commands control this mode.
- PULse generates a trigger event when a specified pulse is detected. It is controlled by commands in the TRIGger[:A]:PULSe subsystem.

Syntax

```
TRIGger[:A]:TYPE <trigger_type>
TRIGger[:A]:TYPE?
```

Parameters

<trigger_type>	Query Response
EDGE	EDGE
PULSe	PULS

Reset Value

EDGE

www.valuetronics.com

Errors and Events

Execution Error –141, "Invalid character data"
Attempted to set trigger type to an illegal value.

Dependencies

Setting TRIGger:TYPE to a value other than EDGE will set TRIGger:B:SOURce to IMMediate.

Examples

Command: `TRIG:TYPE PULS`

Query: `TRIG:TYPE?`

Response: `PULS`

Related Commands

None

www.valuetronics.com

TRIGger:B Subsystem

This section describes the commands in the TRIGger:B subsystem. These commands operate with the ARM, INITiate, TRIGger[:A] and ABORt subsystems to trigger acquisitions. The TRIGger:B subsystem is an alias for the SCPI SEQuence2 trigger block. The defined alias for the SCPI trigger :SEQuence[2] is :B.

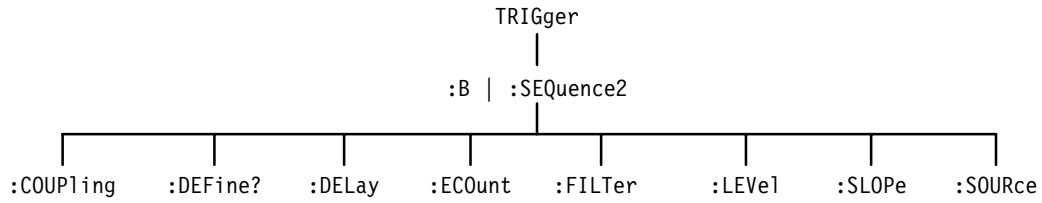


Figure 3–32: TRIGger:B (SCPI SEQuence2) Subsystem Hierarchy

TRIGger:B:COUPling
TRIGger:B:COUPling?

Sets or queries whether the source of the B trigger is AC or DC coupled. AC coupling removes any DC component from the signal. DC coupling passes all frequencies equally. This command is effective only when you set TRIGger:SOURce to an INTernal source. Note that the external trigger is limited to DC coupling only.

You may want to review the command TRIGger:COUPling:<preset> which lets you set trigger coupling and filtering with one command. However, that command does not adhere to the SCPI standard.

Syntax

```
TRIGger:B:COUPling <trigger_coupling>
TRIGger:B:COUPling?
```

Parameters

<trigger_coupling>	Query Response
AC	AC
DC	DC

Reset Value DC

www.valuetronics.com

Errors and Events

Execution Error –141, "Invalid character data"
Attempted to set coupling to an illegal value.

Dependencies

Changing B trigger coupling from DC to AC sets TRIGger:B:FILTer[:LPASs] to OFF and may cause TRIGger:B:LEVel to change if the level is out range for AC coupling.

Changing B trigger coupling from AC to DC sets TRIGger:B:FILTer:HPASs to OFF.

Examples

Command: `TRIG:B:COUP AC`

Query: `TRIG:B:COUP?`

Response: `AC`

Related Commands

TRIGger:B:COUPling:<preset>
TRIGger:B:FILTer[:LPASs]
TRIGger:B:FILTer:HPASs
TRIGger:B:FILTer:NREJect
TRIGger:B:SOURce

TRIGger:B:COUPling:<preset>

This command provides a traditional way of setting the B trigger coupling and filtering in one step. It should not be used in applications that must work on a variety of SCPI instruments. For a description of how this command affects the four underlying SCPI commands, see *Dependencies* for this command.

This command is effective only when you set TRIGger:B:SOURce to INTernal. Note that the external trigger provides DC coupling only. There is no associated query for this command.

Syntax

```
TRIGger:B:COUPling:AC
TRIGger:B:COUPling:ACNReject
TRIGger:B:COUPling:DC
TRIGger:B:COUPling:DCNReject
TRIGger:B:COUPling:HFReject
TRIGger:B:COUPling:LFReject
```

Parameters

None

Reset Value

Not applicable

www.valuetronics.com

Errors and Events None

Dependencies The following table describes the interactions between TRIGger:B:COUPling:<preset> and its four underlying SCPI commands.

<preset>	COUPling	FILT[:LPASs]	FILT:HPASs	FILT:NREJect
AC	AC	OFF	OFF	OFF
ACNReject	AC	OFF	OFF	ON
DC	DC	OFF	OFF	OFF
DCNReject	DC	OFF	OFF	ON
HFReject	DC	ON	OFF	OFF
LFReject	AC	OFF	ON	OFF

Changing trigger coupling from DC to AC may cause TRIGger:B:LEVel to change if level is out of range for AC coupling.

Examples Command: `TRIG:B:COUP:ACNR`

Related Commands TRIGger:COUPling
TRIGger:FILTer[:LPASs]
TRIGger:FILTer:HPASs
TRIGger:FILTer:NREJect

TRIGger:SEQuence2:DEFine? (Query Only)

This query returns the predefined SEQuence2 alias, B.

Syntax `TRIGger:SEQuence2:DEFine?`

Parameters

<sequence_alias>	Query Response
Not Applicable	<string> "B"

Reset Value "B"

Errors and Events None

www.valuetronics.com

Dependencies None

Examples Query: `TRIG:SEQ2:DEF?`

Response: `"B"`

Related Commands ARM:DEFine?
TRIGger:DEFine?

TRIGger:B:DELay
TRIGger:B:DELay?

Sets or queries the trigger delay for the Trigger B circuit. Trigger delay sets the time after the Trigger B event to start acquisition. TRIGger:B:DELay is always positive which delays the start of acquisition a specified time after the trigger event. The minimum real delay is 16 ns. Setting TRIGger:DELay to 0 indicates no delay and bypasses the delay counter. Use SWEep:OFFSet:TIME to acquire pretrigger data.

Refer to the Acquisition discussion on page 2–9 and the Triggering discussion on page 2–21.

Syntax

```
TRIGger:B:DELay <delay_time>
TRIGger:B:DELay?
```

Parameters

<delay_time>[1]	Query Response
<NRf> 16 ns ≤ N ≤ 250 s (in 4 ns steps)	<NR3>
MINimum	0.0E+0
MAXimum	250.0E+0

1 **The default for the parameter <delay_time> is seconds (S). You can also use the multipliers MS for milliseconds, US for microseconds, and NS for nanoseconds.**

Reset Value 0.0E+0

Errors and Events Execution Error –222, “Data out of range”
Attempted to set B trigger delay to an illegal value.

Dependencies Setting TRIGger:B:DELay to a value greater than 0 will cause TRIGger[:A]:DELay to be set to 0. Only one of the delays may operate at a time.

www.valuetronics.com

Examples

Command: `TRIG:B:DEL 10E-6`

Query: `TRIG:B:DEL?`

Response: `10.0E-6`

Related Commands

SWEep:OFFSet:TIME
TRIGger[:A]:DELay
TRIGger:B:ECOunt

TRIGger:B:ECOunt
TRIGger:B:ECOunt?

Sets or queries the number of B trigger events to count before starting acquisition. The number of B trigger events is ignored when you set TRIGger:B:SOURce to IMMediate.

Syntax

```
TRIGger:B:ECOunt <delay_events>
TRIGger:B:ECOunt?
```

Parameters

<delay_events>	Query Response
<NRf> 1 ≤ N ≤ 10,000,000 MINimum MAXimum	<NR1> 1 10000000

Reset Value

1

Errors and Events

Execution Error –222, "Data out of range"
Attempted to set B trigger events to an illegal value.

Dependencies

Setting TRIGger:B:ECOunt to a value greater than 1 when TRIGger:B:SOURce is set to INTernal or EXTernal sets TRIGger[:A]:DELay to 0.

Examples

Command: `TRIG:B:ECO 100`

Query: `TRIG:B:ECO?`

Response: `100`

www.valuetronics.com

Related Commands

TRIGger[:A]:DELay
TRIGger:B:DELay

TRIGger:B:FILTer[:LPASs]
TRIGger:B:FILTer[:LPASs]?

Sets or queries the state of the 50 kHz, low-pass filter for the B trigger circuit. Components of the trigger signal above 50 kHz are attenuated when the LPASs filter is on. It may only be used when DC coupled and with TRIGger:SOURce set to INTernal. Only one trigger filter (i.e., LPAS, HPAS, or NREJ) may be enabled at a time.

The INPut:FILTer is separate from the trigger filter and is located before the trigger pickoff in the signal path.

Syntax

```
TRIGger:B:FILTer[:LPASs][:STATe] <boolean>
TRIGger:B:FILTer[:LPASs][:STATe]?
```

Parameters

<boolean>	Query Response
<NRf>	<NR1>
1 or ON	1
0 or OFF	0

Reset Value

0

Errors and Events

None

Dependencies

Enabling the low pass filter will set TRIGger:B:COUPling to DC and set TRIGger:B:FILTer:NREJect to OFF.

When TRIGger:A:SOURce and TRIGger:B:SOURce are set to the same signal, changing LPAS for one will change LPAS for the other.

Examples

Command: `TRIG:B:FILT ON`

Query: `TRIG:B:FILT?`

Response: 1

www.valuetronics.com

Related Commands

TRIGger:B:COUPling
TRIGger:B:FILTer:HPASs
INPut:FILTer

TRIGger:B:FILTer:HPASs
TRIGger:B:FILTer:HPASs?

Sets or queries the state of the 50 kHz high pass filter for the B trigger circuit. The HPASs filter attenuates components of the trigger signal below 50 kHz. HPASs can only be used when AC coupled and with TRIGger:SOURce set to INTernal. Only one trigger filter (i.e., LPAS, HPAS, or NREJ) may be enabled at a time.

The INPut:FILTer is separate from the trigger filter and is located before the trigger pickoff in the signal path.

Syntax

```
TRIGger:B:FILTer:HPASs[:STATe] <boolean>
TRIGger:B:FILTer:HPASs[:STATe]?
```

Parameters

<boolean>	Query Response
<NRf>	<NR1>
1 or ON	1
0 or OFF	0

Reset Value

0

Errors and Events

None

Dependencies

Enabling the high pass filter sets TRIGger:B:COUPling to AC and TRIGger:B:FILTer:NREJect to OFF.

When TRIGger:A:SOURce and TRIGger:B:SOURce are set to the same signal, changing HPAS for one will change HPAS for the other.

Examples

Command: `TRIG:B:FILT:HPAS ON`

Query: `TRIG:B:FILT:HPAS?`

Response: 1

www.valuetronics.com

Related Commands

TRIGger:B:COUPling
TRIGger:B:FILTer[:LPASs]
TRIGger:B:FILTer:NREJect
INPut:FILTer

TRIGger:B:FILTer:NREJect
TRIGger:B:FILTer:NREJect?

Sets or queries whether or not the noise reject filter is enabled. This filter provides a means of rejecting noise on the trigger signal. Only one trigger filter (i.e., LPAS, HPAS, or NREJ) may be enabled at a time. This command is effective only when you set TRIGger:SOURce to INTernal

Syntax

```
TRIGger:B:FILTer:NREJect <boolean>
TRIGger:B:FILTer:NREJect?
```

Parameters

<boolean>	Query Response
<NRf>	<NR1>
1 or ON	1
0 or OFF	0

Reset Value

0

Errors and Events

None

Dependencies

Enabling NREJ sets HPAS and LPAS trigger filters to OFF.

When TRIGger:A:SOURce and TRIGger:B:SOURce are set to the same signal, changing NREJ for one will change NREJ for the other.

Examples

Command: `TRIG:B:FILT:NREJ ON`

Query: `TRIG:B:FILT:NREJ?`

Response: 1

Related Commands

TRIGger:B:LEVel

www.valuetronics.com

TRIGger:B:LEVel
TRIGger:B:LEVel?

Sets or queries the trigger level for the TRIGger:B subsystem. The trigger level is a specific voltage through which the trigger signal must pass to be recognized as a trigger event and start acquisition. The trigger level should be within the range of the signal in order to guarantee a trigger event. TRIGger:LEVel is effective only when you set TRIGger:SOURce to INTernal or EXTernal. Note that the front panel TRIGD LED lights briefly when a trigger event occurs.

Attaching a probe modifies the minimum and maximum values as it does the setting for VOLTage:RANGe:PTPeak. Multiply the minimum and maximum limits by the attenuation factor of the probe to determine the new maximum and minimum values.

If you experience or expect an unstable trigger point due to noise on the trigger signal, use the noise reject feature of the trigger circuit to reduce the affects of noise. With the command TRIGger:B:FILTer:NREJect you can reduce the effects of noise on the internal trigger sources.

Syntax

```
TRIGger:B:LEVel <trigger_level>
TRIGger:B:LEVel?
```

Parameters

<trigger_level>[1]	Query Response
<NRf> External: –1.0 ≤ N ≤ 1.0 (2 mV steps) MINimum MAXimum	<NR3> –1.0E+0 1.0E+0
Internal DC Coupled: absolute maximum[2]: –200.0 ≤ N ≤ 200.0 limited by ± VOLT:RANG:PTP steps 0.002 * VOLT:RANG:PTP MINimum MAXimum	 VOLT:RANG:OFFS ± VOLT:RANG:PTP VOLT:RANG:OFFS – VOLT:RANG:PTP VOLT:RANG:OFFS + VOLT:RANG:PTP
Internal AC Coupled: absolute maximum: –100.0 ≤ N ≤ 100.0 limited by ± VOLT:RANG:PTP steps 0.002 * VOLT:RANG:PTP MINimum MAXimum	 ± VOLT:RANG:PTP – VOLT:RANG:PTP + VOLT:RANG:PTP

1 The default for the parameter <trigger_level> is V for volts. You can also use the multipliers MV for millivolts and UV for microvolts.

2 When you connect a probe, the maximum limits increase just as the vertical range increases for the input. For example, with a 1 V vertical range, connecting a 10X probe increases the vertical range and trigger level range to 10 V.

www.valuetronics.com

Reset Value 0.0E+0

Errors and Events Execution Error –222, "Data out of range"
Attempted to set B trigger level to an illegal value.

Dependencies None

Examples Command: `TRIG:B:LEV 1`

Query: `TRIG:B:LEV?`

Response: `1.0E+0`

Related Commands TRIGger:B:FILTer:NREJect
TRIGger:B:SOURce
VOLTage:RANGe:PTPeak
VOLTage:RANGe:OFFSet

TRIGger:B:SLOPe
TRIGger:B:SLOPe?

Sets or queries whether triggering occurs on the positive-going or negative-going edge of the trigger B signal source. This command has effect only when TRIGger:B:SOURce is set to INTernal or EXTernal. The other trigger sources are digital signals.

Syntax
```
TRIGger:B:SLOPe <trigger_slope>
TRIGger:B:SLOPe?
```

Parameters

<trigger_slope>	Query Response
POSitive	POS
NEGative	NEG

Reset Value POS

Errors and Events Execution Error –141, "Invalid character data"
Attempted to set slope to an illegal value.

Dependencies None

www.valuetronics.com

Examples

Command: `TRIG:B:SLOP NEGATIVE`

Query: `TRIG:B:SLOP?`

Response: `NEG`

Related Commands

TRIGger:B:LEVel

TRIGger:B:SOURce
TRIGger:B:SOURce?

Sets or queries the source of the trigger signal for the trigger B circuit. You can specify only one source at a time.

Setting source to INTernal1 selects the signal from INPut1 and [SENSe:]VOLTage1 as the trigger source. The EXTernal signal is from the front-panel BNC connector labeled External Trigger Input. Setting the source to IMMediate will bypass event detection in the trigger B circuit forcing a trigger B event.

Many commands in the TRIGger subsystem are dependent on the trigger source you select. For instance, the TRIGger:B:COUPling setting is ignored when you set the trigger source to EXTernal. The dependencies for each command are listed with the command description.

Syntax

```
TRIGger:B:SOURce <trigger_source>
TRIGger:B:SOURce?
```

Parameters

<trigger_source>	Query Response
EXTernal	EXT
IMMediate	IMM
INTernal1	INT1
INTernal2	INT2
INTernal3	INT3
INTernal4	INT4

Reset Value

IMM

Errors and Events

Execution Error –141, “Invalid character data”
Attempted to set the trigger source to an illegal value.

Execution Error –221, “Settings conflict.”
Attempted to set trigger source to a value other than IMMediate when TRIGger:TYPE is set to PULSe.

www.valuetronics.com

Execution Error –241, “Hardware missing”
Attempted to set the trigger source to INTernal3 or INTernal4 when the instrument has only two channels.

Dependencies

Changing the trigger B source from INTernal to EXTernal or from one INTernal source to another INTernal source with a different VOLTage:RANGe setting may change the TRIGger:B:LEVel setting to accommodate a new vertical range.

Setting TRIGger:B:SOURce to INTernal[1, 2, 3, 4] or EXTernal when TRIGger:B:ECOunt is greater than 1 sets TRIGger[:A]:DELay to 0.

Examples

Command: `TRIG:B:SOUR INTERNAL1`

Query: `TRIG:B:SOUR?`

Response: `INT1`

Related Commands

ARM:SOURce
TRIGger[:A]:SOURce

www.valuetronics.com

TRIGger:PULSe Subsystem

This section describes the commands in the TRIGger:PULSe subsystem. The pulse trigger commands provide the capability to trigger when a pulse occurs that is outside specified parameters.

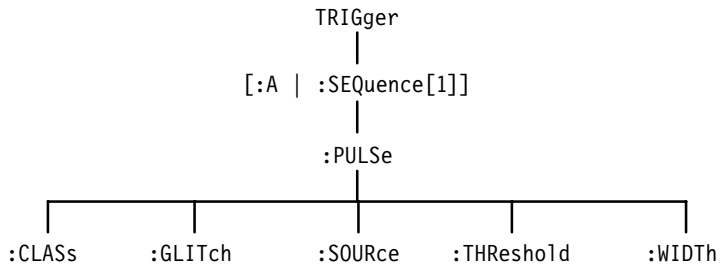


Figure 3–33: TRIGger:PULSe Subsystem Hierarchy

TRIGger:PULSe:CLASs
TRIGger:PULSe:CLASs?

Sets or queries the class of pulse triggering to use for the next acquisition. The signal monitored for pulse triggering is the trigger A source. The trigger classes are as follows:

- GLITch — the trigger event occurs when the instrument detects a pulse that is either narrower or wider than a specified duration or width.
- WIDTh — the trigger event occurs when the instrument detects a pulse that is either inside or outside a specified duration or width.

This command has effect only when you set TRIGger:TYPE to PULSe. The defined alias for the SCPI trigger :SEQuence[1] is A.

Syntax

```
TRIGger[:A]:PULSe:CLASs <pulse_class>
TRIGger[:A]:PULSe:CLASs?
```

Parameters

<pulse_class>	Query Response
GLITch	GLIT
WIDTh	WIDT

Reset Value GLITch

www.valuetronics.com

Errors and Events Execution Error –224, "Illegal parameter value"
Attempted to set class to an illegal value.

Dependencies None

Examples Command: `TRIG:PULS:CLAS WIDTH`

Query: `TRIG:PULS:CLAS?`

Response: `WIDT`

Related Commands TRIGger:TYPE

TRIGger:PULSe:GLITch:POLarity
TRIGger:PULSe:GLITch:POLarity?

Sets or queries the polarity of the event pulse for pulse glitch triggering. Before this command will be effective, you must set TRIGger:TYPE to PULSe and TRIGger:PULSe:CLASs to GLITch. Refer to the discussion of TRIGger:PULSe:CLASs on page 3–237.

Syntax

```
TRIGger[:A]:PULSe:GLITch:POLarity <polarity>
TRIGger[:A]:PULSe:GLITch:POLarity?
```

Parameters

<polarity>	Query Response
EITHer	EITH
NEGative	NEG
POSitive	POS

Reset Value POS

Errors and Events Execution Error –224, "Illegal parameter value"
Attempted to set polarity to an illegal value.

Dependencies None

Examples Command: `TRIG:PULS:GLIT:POL NEG`

Query: `TRIG:PULS:GLIT:POL?`

www.valuetronics.com

Response: NEG

Related Commands

TRIGger:TYPE
TRIGger:PULSe:CLASs

TRIGger:PULSe:GLITch:QUALify
TRIGger:PULSe:GLITch:QUALify?

Sets or queries the type of time qualification for pulse glitch triggering. LT specifies that the trigger event occurs when the glitch width is less than the specified time. GT specifies that the trigger event occurs when the glitch width is greater than the specified time.

Before this command will be effective, you must set TRIGger:TYPE to PULSe and TRIGger:PULSe:CLASs to GLITch. Refer to the discussion of TRIGger:PULSe:CLASs on page 3–237. The defined alias for the SCPI trigger :SEQuence[1] is A.

Syntax

```
TRIGger[:A]:PULSe:GLITch:QUALify <type>
TRIGger[:A]:PULSe:GLITch:QUALify?
```

Parameters

<type>	Query Response
GT	GT
LT	LT

Reset Value

LT

Errors and Events

Execution Error –224, “Illegal parameter value”
Attempted to set qualify to an illegal value.

Dependencies

None

Examples

Command: TRIG:PULS:GLIT:QUAL GT

Query: TRIG:PULS:GLIT:QUAL?

Response: GT

www.valuetronics.com

Related Commands

TRIGger:TYPE
TRIGger:PULSe:CLASs
TRIGger:PULSe:GLITch:WIDTh

TRIGger:PULSe:GLITch:WIDTh
TRIGger:PULSe:GLITch:WIDTh?

Sets or queries the width of the pulse used for pulse glitch triggering. Use the command TRIGger:PULSe:GLITch:QUALify to set whether the trigger event is for pulses less than or greater than the WIDTh you specify.

Before this command will be effective, you must set TRIGger:TYPE to PULSe and TRIGger:PULSe:CLASs to GLITch. Refer to the discussion of TRIGger:PULSe:CLASs on page 3–237. The defined alias for the SCPI trigger :SEQuence[1] is A.

Syntax

```
TRIGger[:A]:PULSe:GLITch:WIDTh <width>
TRIGger[:A]:PULSe:GLITch:WIDTh?
```

Parameters

<width>[1]	Query Response
<NRf>	<NR3>
MINimum	1.0E–9
MAXimum	1.0E+0

1 **The default for the parameter <delay_time> is seconds (S). You can also use the multipliers MS for milliseconds, US for microseconds, NS for nanoseconds, and PS for picoseconds.**

Reset Value

2.0E–9

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set width to an illegal value.

Dependencies

None

Examples

Command: `TRIG:PULS:GLIT:WIDT 1.0`

Query: `TRIG:PULS:GLIT:WIDT?`

Response: `1.0E+0`

www.valuetronics.com

Related Commands

TRIGger:TYPE
TRIGger:PULSe:CLASs
TRIGger:PULSe:GLITch:QUALify

TRIGger:PULSe:SOURce
TRIGger:PULSe:SOURce?

Sets or queries the source of the trigger signal used for all pulse triggering. To activate pulse triggering, you must set TRIGger:TYPE to PULSe.

Setting source to INTernal1 selects the signal from INPut1 and [SENSe:]VOLTage1 to be the trigger source. The defined alias for the SCPI trigger :SEQuence[1] is A.

Syntax

```
TRIGger[:A]:PULSe:SOURce <source>
TRIGger[:A]:PULSe:SOURce?
```

Parameters

<source>	Query Response
INTernal[1]	INT
INTernal2	INT2
INTernal3	INT3
INTernal4	INT4

Reset Value

INT

Errors and Events

Execution Error –224, “Illegal parameter value”
Attempted to set source to an illegal value.

Dependencies

None

Examples

Command: `TRIG:PULS:SOUR INT3`

Query: `TRIG:PULS:SOUR?`

Response: `INT3`

Related Commands

TRIGger:TYPE

www.valuetronics.com

TRIGger:PULSe:THReshold
TRIGger:PULSe:THReshold?

Sets or queries the voltage threshold used for pulse triggering. The threshold is the voltage level that the pulse trigger source must cross to mark the beginning and end of a pulse. To set positive or negative pulse polarity use the command TRIG:PULS:WIDT:POL or TRIG:GLIT:WIDT:POL, depending on the selected type of pulse triggering.

Syntax

```
TRIGger[:A]:PULSe:THReshold <level>
TRIGger[:A]:PULSe:THReshold?
```

Parameters

<level>[1]	Query Response
<NRf>	<NR3>
Internal DC Coupled: absolute maximum[2]: –200.0 ≤ N ≤ 200.0 limited by ± VOLT:RANG:PTP steps 0.002 * VOLT:RANG:PTP	VOLT:RANG:OFFS ± VOLT:RANG:PTP
MINimum[2]	VOLT:RANG:OFFS – VOLT:RANG:PTP
MAXimum[2]	VOLT:RANG:OFFS + VOLT:RANG:PTP

1 The default for the parameter <level> is V for volts. You can also use the multipliers MV for millivolts and UV for microvolts.

2 When you connect a probe, the maximum limits increase just as the vertical range increases for the input. For example, with a 1 V vertical range, connecting a 10X probe increases the vertical range and trigger level range to 10 V.

Reset Value

0.0E+0

Errors and Events

Execution Error –222, "Data out of range"
Attempted to set threshold to an illegal value.

Dependencies

None

Examples

Command: `TRIG:PULS:THR 1.0`

Query: `TRIG:PULS:THR?`

Response: `1.0E+0`

Related Commands

TRIGger:TYPE

www.valuetronics.com

TRIGger:PULSe:WIDTh:HLIMit
TRIGger:PULSe:WIDTh:HLIMit?

Sets or queries the high or longest valid pulse width for to qualify for pulse triggering. After a pulse on the trigger source begins, it cannot take longer than :WIDTh:HLIMit to cross the TRIGger:PULSe:THReshold level. HLIMit must be longer than :LLIMit. To activate pulse width triggering, set TRIGger:TYPE to PULSe and TRIGger:PULSe:CLASs to WIDTh.

Syntax

```
TRIGger[:A]:PULSe:WIDTh:HLIMit <hlimit>
TRIGger[:A]:PULSe:WIDTh:HLIMit?
```

Parameters

<hlimit>[1]	Query Response
<NRf>	<NR3>
MAXimum	1.0E–9
MINimum	1.0E+0

1 **The default for the parameter <hlimit> is seconds (S). You can also use the multipliers MS for milliseconds, US for microseconds, NS for nanoseconds, and PS for picoseconds.**

Reset Value

4.0E–9

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set high limit to an illegal value.

Dependencies

None

Examples

Command: `TRIG:PULS:WIDT:HLIM 1.0`

Query: `TRIG:PULS:WIDT:HLIM?`

Response: `1.0E+0`

Related Commands

TRIGger:TYPE
TRIGger:PULSe:CLASs
TRIGger:PULSe:WIDTh:LLIMit

www.valuetronics.com

TRIGger:PULSe:WIDTh:LLIMit
TRIGger:PULSe:WIDTh:LLIMit?

Sets or queries the lower or minimum valid pulse width to qualify for pulse triggering. The pulse on the trigger source must take longer than the :WIDTh value to cross the TRIGger:PULSe:THReshold level. To activate this command, set TRIGger:TYPE to PULSe and TRIGger:PULSe:CLASs to WIDTh.

Syntax

TRIGger[:A]:PULSe:WIDTh:LLIMit <llimit>
TRIGger[:A]:PULSe:WIDTh:LLIMit?

<llimit>[1]	Query Response
<NRf>	<NR3>
MAXimum	1.0E–9
MINimum	1.0E+0

1 **The default for the parameter <llimit> is seconds (S). You can also use the multipliers MS for milliseconds, US for microseconds, NS for nanoseconds, and PS for picoseconds.**

Reset Value

2.0E–9

Errors and Events

Execution Error –222, “Data out of range”
Attempted to set lower limit to an illegal value.

Dependencies

None

Examples

Command: `TRIG:PULS:WIDT:LLIM 0.1`

Query: `TRIG:PULS:WIDT:LLIM?`

Response: `100.0E-3`

Related Commands

TRIGger:TYPE
TRIGger:PULSe:CLASs
TRIGger:PULSe:WIDTh:HLIMit

www.valuetronics.com

TRIGger:PULSe:WIDTh:POLarity
TRIGger:PULSe:WIDTh:POLarity?

Sets or queries the polarity of the pulse used for pulse width triggering. Use the command TRIGger:PULSe:WIDTh:QUALify to set whether the trigger event is for pulses inside or outside the limits (LLIMit and HLIMit).

This command is active only when you set TRIGger:TYPE to PULSe and TRIGger:PULSe:CLASs to WIDTh.

Syntax

```
TRIGger[:A]:PULSe:WIDTh:POLarity <polarity>
TRIGger[:A]:PULSe:WIDTh:POLarity?
```

Parameters

<polarity>	Query Response
NEGative	NEG
POSitive	POS

Reset Value

POS

Errors and Events

Execution Error –224, “Illegal parameter value”
Attempted to set polarity to an illegal value.

Dependencies

None

Examples

Command: `TRIG:PULS:WIDT:POL NEG`

Query: `TRIG:PULS:WIDT:POL?`

Response: `NEG`

Related Commands

TRIGger:TYPE
TRIGger:PULSe:CLASs

www.valuetronics.com

TRIGger:PULSe:WIDTh:QUALify
TRIGger:PULSe:WIDTh:QUALify?

Sets or queries the type of time qualification for pulse width triggering. The type IN triggers when the measured pulse width is within the specified lower and higher limits. OUT triggers when the pulse width is outside the specified lower and upper limits.

This command is active only when you set TRIGger:TYPE to PULSe and TRIGger:PULSe:CLASs to WIDTh.

Syntax

```
TRIGger[:A]:PULSe:WIDTh:QUALify <type>
TRIGger[:A]:PULSe:WIDTh:QUALify?
```

Parameters

<type>	Query Response
IN OUT	IN OUT

Reset Value

IN

Errors and Events

Execution Error –224, “Illegal parameter value”
Attempted to set qualify to an illegal value.

Dependencies

None

Examples

Command: `TRIG:PULS:WIDT:QUAL OUT`

Query: `TRIG:PULS:WIDT:QUAL?`

Response: `OUT`

Related Commands

TRIGger:TYPE
TRIGger:PULSe:CLASs
TRIGger:PULSe:WIDTh:LLIMit
TRIGger:PULSe:WIDTh:HLIMit

www.valuetronics.com

VOLTage Subsystem

This section describes each command and query in the [SENSe:]VOLTage subsystem. Figure 3–34 shows the command tree for the VOLTage subsystem. Figure 3–35 shows the part of the waveform analyzer controlled by the VOLTage commands. The input channel number, defined at the probe connector, is shared as the parameter <n> for the INPut<n> and VOLTage<n> commands. Only the four channel TVS641 will accept VOLTage3 and VOLTage4 commands.

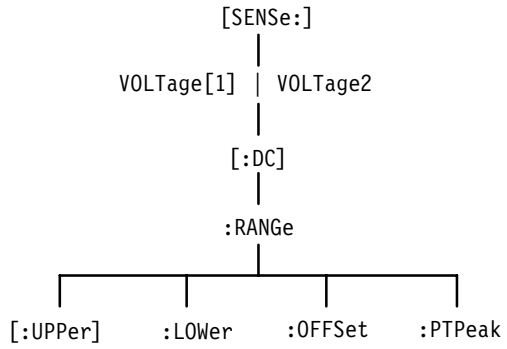


Figure 3–34: VOLTage Subsystem Hierarchy

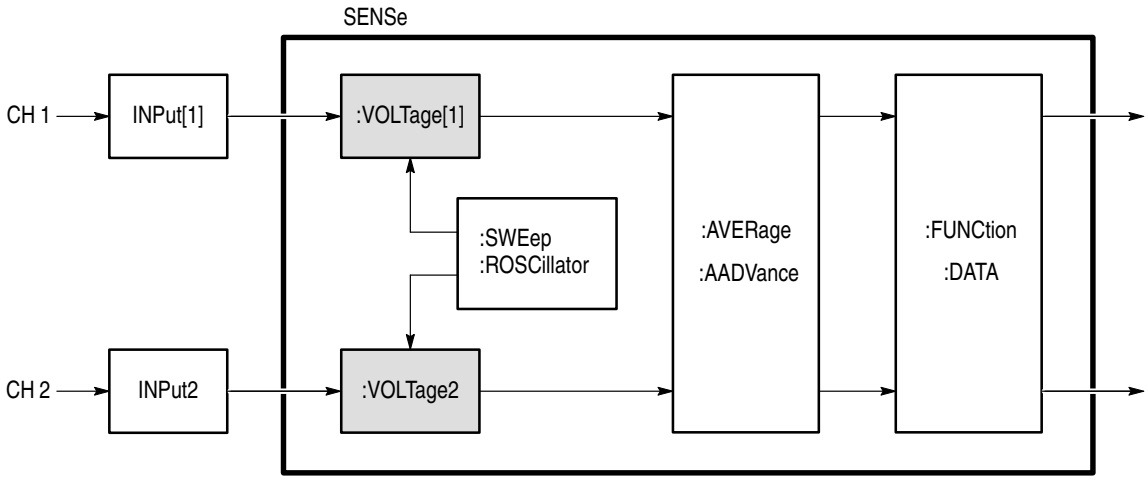


Figure 3–35: VOLTage Subsystem Functional Model

www.valuetronics.com

VOLTage:RANGe[:UPPer]
VOLTage:RANGe[:UPPer]?

Sets or queries the most positive end of the amplifier voltage range. You should set the amplifier voltage range to match the signal amplitude you expect to acquire with the digitizer. The string :UPPer may be omitted as shown in the examples that follow.

The range of permitted values for <upper> depends on the attached probe. Attached probes modify the full-scale range by the attenuation factor of the probe. For example, a 10X probe will cause the 100 mV range to become a 1 V range. This in turn causes the minimum and maximum offset values to change from ±1 V to ±10 V.

Syntax

```
[SENSe:]VOLTage<n>][:DC]:RANGe[:UPPer] <upper>
[SENSe:]VOLTage<n>[:DC]:RANGe[:UPPer]?
```

Parameters

<n> (Channel number)[1]	Query Response
1 2 3 4	NA

1 **The input channel number <n> is used by the commands INPut<n> and VOLTage<n>. If you omit <n>, the default is channel 1.**

<upper> (Upper limit of voltage range)[1]	Query Response
<NRf>	<NR3>

1 **The default multiplier for the <upper> parameter is V for volts. You can also use MV for millivolts.**

Reset Value

0.5E+0

Errors and Events

Execution Error –222, “Data out of range”
You attempted to set upper to an illegal value.

Execution Error –241, “Hardware missing”
You attempted to program VOLTage3 or VOLTage4 on an instrument configured with only two channels.

Dependencies

VOLTage:RANGe[:UPPer] and VOLTage:RANGe:LOWer operate as a pair. Changing one will not cause the other to change. However, a change in the

www.valuetronics.com

voltage range will change VOLTage:RANGe:PTPeak and VOLTage:RANGe:OFFSet so they describe the new voltage range.

VOLTage:RANGe:LOWer is not returned in the response to *LRN? because the voltage range is returned as VOLT:RANG:PTP and VOLT:RANG:OFFS.

Examples

Command: `VOLT1:RANG 2`

Query: `VOLT1:RANG?`

Response: `2.0E+0`

Related Commands

VOLTage:RANGe:LOWer
VOLTage:RANGe:OFFSet
VOLTage:RANGe:PTPeak

VOLTage:RANGe:LOWer
VOLTage:RANGe:LOWer?

Sets or queries the most negative end of the amplifier voltage range. You should set the amplifier voltage range to match the signal amplitude you expect to acquire with the digitizer.

The range of permitted values for <lower> depends on the attached probe. An attached probe modifies the full-scale range by the attenuation factor of the probe. For example, a 10X probe causes the 100 mV range to become a 1 V range and the minimum and maximum offset values to change from ±1 V to ±10 V.

Syntax

```
[SENSe:]VOLTage<n>[:DC]:RANGe:LOWer <lower>
[SENSe:]VOLTage<n>[:DC]:RANGe:LOWer?
```

Parameters

<n> (Channel number)[1]	Query Response
1 2 3 4	NA

1 **The input channel number <n> is used by the commands INPut<n> and VOLTage<n>. If you omit <n>, the default is channel 1.**

www.valuetronics.com

<lower> (Lower limit of voltage range)[1]	Query Response
<NRf>	<NR3>

1 The default multiplier for the <lower> parameter is V for volts. You can also use MV for millivolts.

Reset Value –0.5E+0

Errors and Events Execution Error –222, "Data out of range"
You attempted to set low pass frequency to an illegal value.

Execution Error –241, "Hardware missing"
You attempted to program VOLTage3 or VOLTage4 on an instrument configured with only two channels.

Dependencies VOLTage:RANGe[:UPPer] and VOLTage:RANGe:LOWer operate as a pair. Changing one will not cause the other to change. However, a change in the voltage range will change VOLTage:RANGe:PTPeak and VOLTage:RANGe:OFFSet so they describe the new voltage range.

VOLTage:RANGe:LOWer is not returned in the response to *LRN? because the voltage range is returned as VOLT:RANG:PTP and VOLT:RANG:OFFS.

Examples Command: `VOLT1:RANG:LOW -2`

Query: `VOLT1:RANG:LOW?`

Response: `-2.0E+0`

Related Commands VOLTage:RANGe[:UPPer]
VOLTage:RANGe:OFFSet
VOLTage:RANGe:PTPeak

www.valuetronics.com

VOLTage:RANGe:OFFSet
VOLTage:RANGe:OFFSet?

Sets or queries the voltage offset of the specified input amplifier. The offset value is subtracted from the input signal which allows you to move the signal up or down into the amplifier vertical range. A positive value removes positive DC offset from the signal and a negative offset value removes negative DC offset. A common use of offset is to subtract the DC component of a signal so you can acquire the AC portion at higher vertical resolution. Measurement results do not change when you change offset.

The range of permitted values and the smallest increment for offset depend on the setting of the peak-to-peak range (VOLTage:RANGe:PTPeak) and on the attached probe. An attached probe modifies the full-scale range by the attenuation factor of the probe. For example, a 10X probe causes the 100 mV range to become a 1 V range and the minimum and maximum offset values to change from ±1 V to ±10 V.

Syntax

```
[SENSe:]VOLTage<n>[:DC]:RANGe:OFFSet <offset>
[SENSe:]VOLTage<n>[:DC]:RANGe:OFFSet?
```

Parameters

<n> (Channel number)[1]	Query Response
1 2 3 4	NA

1 **The input channel number <n> is used by the commands INPut<n> and VOLTage<n>. If you omit <n>, the default is channel 1.**

<offset> (Value subtracted from signal)[1]	Query Response
<NRf>	<NR3>
PTPeak 10 mV – 1 V ±1.0 (in 1 mV steps) MINimum MAXimum	 –1.0E+0 1.0E+0
PTPeak 1.01 V – 10 V ±10.0 (in 10 mV steps) MINimum MAXimum	 –10.0E+0 10.0E+0

www.valuetronics.com

<offset> (Value subtracted from signal)[1]	Query Response
PTPeak 10.1 V – 100 V ±100.0 (in 100 mV steps) MINimum MAXimum	 –100.0E+0 100.0E+0

1 **The default multiplier for the <offset> parameter is V for volts. You can also use MV for millivolts.**

Reset Value 0.0E+0

Errors and Events Execution Error –222, "Data out of range"
You attempted to set offset to an illegal value.

Execution Error –241, "Hardware missing"
You attempted to program VOLTage3 or VOLTage4 on an instrument configured with only two channels.

Dependencies VOLTage:RANGe:PTPeak and VOLTage:RANGe:OFFSet operate as a pair. Changing the voltage range, changes VOLTage:RANGe[:UPPer] and VOLTage:RANGe:LOWer so they describe the new voltage range.

Examples Command: `VOLT1:RANG:OFFS 1`

Query: `VOLT1:RANG:OFFS?`

Response: `1.0E+0`

Related Commands VOLTage:RANGe:PTPeak
VOLTage:RANGe:LOWer
VOLTage:RANGe[:UPPer]

www.valuetronics.com

VOLTage:RANGe:PTPeak
VOLTage:RANGe:PTPeak?

Sets or queries the peak-to-peak (full-scale) voltage range of the specified input amplifier. The range of permitted values and the smallest increment for offset depend on the setting of the peak-to-peak range (VOLTage:RANGe:PTPeak). For specific values, see the listing of <range> values under *Parameters*.

The values permitted for <range> also depend on the attached probe. An attached probe modifies the full-scale range by the attenuation factor of the probe. For example, a 10X probe changes the 10 mV range to the 100 mV range.

Syntax

```
[SENSe:]VOLTage<n>[:DC]:RANGe:PTPeak <range>
[SENSe:]VOLTage<n>[:DC]:RANGe:PTPeak?
```

Parameters

<n> (Channel number)[1]	Query Response
1 2 3 4	NA

1 **The input channel number <n> is used by the commands INPut<n> and VOLTage<n>. If you omit <n>, the default is channel 1.**

<range> (Range for vertical amplifier)[1]	Query Response
<NRf>	<NR3>
10 mV – 20 mV (100 µV steps) 20 mV – 50 mV (200 µV steps) 50 mV – 100 mV (500 µV steps) 100 mV – 200 mV (1 mV steps) 200 mV – 500 mV (2 mV steps) 500 mV – 1 V (5 mV steps) 1 V – 2 V (10 mV steps) 2 V – 5 V (20 mV steps) 5 V – 10 V (50 mV steps) 10 V – 20 V (100 mV steps) 20 V – 50 V (200 mV steps) 50 V – 100 V (500 mV steps)	
MINimum MAXimum	10.0E–3 100.0E+0

1 **The default multiplier for the <offset> parameter is V for volts. You can also use MV for millivolts.**

Reset Value 1.0E+0

www.valuetronics.com

Errors and Events

Execution Error –222, "Data out of range"
You attempted to set range to an illegal value.

Execution Error –241, "Hardware missing"
You attempted to program VOLTage3 or VOLTage4 on an instrument configured with only two channels.

Dependencies

VOLTage:RANGe:PTPeak and VOLTage:RANGe:OFFSet together define what part of the input signal to acquire. Changing the voltage range with VOLTage:RANGe:PTPeak does change VOLTage:RANGe[:UPPer] and VOLTage:RANGe:LOWer, because together they provide an alternate way to query, or define, the voltage range for acquisition.

Examples

Command: `VOLT1:RANG:PTP 100`

Query: `VOLT1:RANG:PTP?`

Response: `100.0E+0`

Related Commands

VOLTage:RANGe:OFFSet
VOLTage:RANGe:[UPPer]
VOLTage:RANGe:LOWer

www.valuetronics.com

IEEE 488.2 Common Commands

This section describes the IEEE 488.2 common commands. Figure 3–36 shows the common syntax for these commands. The program mnemonics are described in alphabetical order in this section.

```
*<program mnemonic>[?]
```

Figure 3–36: IEEE 488.2 Common Command Syntax

*CAL? (Query Only)

Initiates internal calibration and returns a failure code. If more than one failure occurs, the response includes only the number of the first failure. You can obtain additional test results with the command CALibration:RESult:VERBose?.

*CAL? and CALibration? perform the same function.

Syntax `*CAL?`

Parameters

<failure_code>	Query Response	
Not applicable	<NR1>	
	0	(No failures)
	2000 to 2999	(Calibration failure)

Reset Value Not applicable

Errors and Events None

Dependencies None

Examples Query: `*CAL?`

Response: `0`

Related Commands CALibration
CALibration:RESults:VERBose?

www.valuetronics.com

*CLS

Clears, or nulls, the SCPI and IEEE 488.2 event registers and the Status Queue. However, the SCPI and IEEE 488.2 enable registers are not cleared. For more information on the Status Queue and the event registers, refer to the Status and Events discussion on page 4–1.

Syntax `*CLS`

Parameters None

Reset Value Not applicable

Errors and Events None

Dependencies None

Examples Command: `*CLS`

Related Commands STATus:PRESet

www.valuetronics.com

*ESE
*ESE?

Sets or queries the Service Request Enable Register (SRER). The SRER determines which events in the Standard Event Status Register can set the Event Status bit (bit 5) in the Status Byte Register. The bits in the SRER correspond to the bits in the Standard Event Status Register, which are defined in Table 3–29 on page 3–258.

The STATus:PRESet command does not change the Service Request Enable Register. For more information on the Status and Events reporting system, refer to the Status and Events discussion on page 4–1.

Syntax

```
*ESE <mask>
*ESE?
```

Parameters

<mask>	Query Response
<NRf> | <Non-decimal number> 0 ≤ N ≤ #HFF	<NR1>

Reset Value

Not applicable

Errors and Events

None

Dependencies

None

Examples

Command: `*ESE #H40`
This command enables only the User Request bit (bit6) in the Standard Event Status Register. No other events can set the Event Status bit in the Status Byte Register.

Query: `*ESE?`

Response: `64`

Related Commands

*ESR?

www.valuetronics.com

*ESR? (Query Only)

Returns the contents of the Standard Event Status Register as a decimal number. More than one event may be reported at the same time. The Status Queue contains a chronological record of the events. Table 3–29 describes the bit assignments in the Standard Event Status Register. Reading this register clears it. For more information on the Status and Events reporting system, refer to the Status and Events discussion on page 4–1.

Table 3–29: The Standard Event Status Register

Bit	Decimal Value	Function
0	1	**Operation Complete** shows that the operation is complete. This bit is set when all pending operations complete following an *OPC command.
1	2	**Request Control** (not used)
2	4	**Query Error** shows that the waveform analyzer attempted to read the Output Queue when no data was present or pending, or that data in the Output Queue was lost.
3	8	**Device Dependent Error** shows that a device error occurred. Table 4–10 on page 4–14 lists the device error messages.
4	16	**Execution Error** shows that an error occurred while the waveform analyzer was executing a command or query. Table 3–2 on page 4–13 lists the execution error messages.
5	32	**Command Error** shows that an error occurred while the waveform analyzer was parsing a command or query. Table 4–8 on page 4–12 lists the command error messages.
6	64	**User Request** indicates that a probe ID button was pressed.
7	128	**Power On** shows that the waveform analyzer was powered on.

Syntax

```
*ESR?
```

Parameters

<event>	Query Response
Not applicable	<NR1>

Reset Value

Not applicable

Errors and Events

None

Dependencies

Reading this register clears it.

www.valuetronics.com

Examples

Query: `*ESR?`

Response: `40`

Related Commands

*ESE

*IDN? (Query Only)

Returns the waveform analyzer identification message. The fourth field of the identification message includes both the SCPI and firmware version numbers.

Syntax

`*IDN?`

Parameters

<id_message>	Query Response
Not applicable	Manufacturer, model number, serial number, SCPI and firmware version numbers (syntax defined by IEEE 488.2)

Reset Value

Not applicable

Errors and Events

None

Dependencies

None

Examples

Query: `*IDN?`

Response: `TEKTRONIX,TVS641,B010100,SCPI:95.0 FVER:1.0.25`

Related Commands

SYSTem:SNUMber?
SYStem:VERSion?

www.valuetronics.com

*LRN? (Query Only)

Returns the current state of the waveform analyzer as a sequence of ASCII settings. You can store and return these settings as a group to the waveform analyzer to place it in a known state. Command headers and parameters in the returned settings are in the short form.

Syntax

```
*LRN?
```

Parameters

<ascii_setting>	Query Response
Not applicable	<Program Message Unit> {,<Program Message Unit>}

Reset Value

Not applicable

Errors and Events

None

Dependencies

None

Examples

Query:

```
*LRN?
```

Response:

```
INP1:COUP DC;FILT 1;
FILT:FREQ 20.0E+6;
IMP 50.0E+0;:INP2 ....
```

Related Commands

SYSTem:SET
*RCL
*SAV

www.valuetronics.com

*OPC
*OPC?

Synchronizes command execution with the controller. *OPC causes an Operation Complete event to be generated when the active command finishes. *OPC? causes an ASCII "1" to be placed in the output queue when the command completes.

Syntax

```
*OPC
*OPC?
```

Parameters None

Reset Value Not applicable

Errors and Events None

Dependencies None

Examples

Command: `*OPC`

Query: `*OPC?`

Response: 1

Related Commands *WAI

*OPT? (Query Only)

Returns the options installed in the instrument as a comma-separated list. The options are identified by their option designators, such as "10" for Option 10. A zero response indicates no option is installed.

Syntax

```
*OPT?
```

Parameters

<options>	Query Response
Not applicable	<arbitrary ascii response data> {10|0}

www.valuetronics.com

Reset Value Not applicable

Errors and Events None

Dependencies None

Examples Query: `*OPT?`

Response: `10`

Related Commands *IDN?

*PUD
*PUD?

Sets or queries the protected user data stored in the waveform analyzer. The user data can not exceed 1023 characters. You can use *PUD to provide a corporate identification or name for your waveform analyzer. You must set SYSTem:PROTect to OFF before attempting to modify the user data with *PUD. The query form may be used at any time.

Syntax

```
*PUD <data>
*PUD?
```

Parameters

<data>	Query Response
<string> (up to 1023 characters)	<string>

Reset Value Not applicable

Errors and Events Execution Error –223, “Too much data”
Attempted to set PUD to a string longer than 1023 characters.

Execution Error –203, “Command protected”
Attempted to set PUD without first removing the restriction with the SYSTem:PROTect command.

Dependencies None

www.valuetronics.com

Examples

Command: `*PUD "WA 4-1"`

Query: `*PUD?`

Response: `"WA 4-1"`

Related Commands

SYSTem:PASSword

*RCL

Recalls the specified instrument setting from one of the ten non-volatile memory locations in the waveform analyzer. You save settings to these locations with the *SAV command.

Syntax

`*RCL <location>`

Parameters

<location>	Query Response
<NRf> 1 to10	Not applicable

Reset Value

Not applicable

Errors and Events

Execution Error –221, “Settings conflict.”
Attempted to recall an instrument settings from an empty memory location.

Execution Error –224, “Illegal parameter value”
Attempted to recall an instrument setting from an illegal memory location.

Dependencies

None

Examples

Command: `*RCL 4`

Related Commands

SYSTem:SET
*LRN?
*SAV

www.valuetronics.com

*RST

Resets instrument settings to a default state. Each command description includes a reset value if appropriate. Many settings in the SYSTem and STATus subsystems are not reset by this command.

Syntax `*RST`

Parameters None

Reset Value Not applicable

Errors and Events None

Dependencies None

Examples Command: `*RST`

Related Commands STATus:PRESet
*CLS

*SAV

Saves the current instrument settings in one of the ten non-volatile memory locations in the waveform analyzer. You can recall these settings at a later time with the *RCL command.

Syntax `*SAV <location>`

Parameters

<location>	Query Response
<NRf> 1 to10	Not applicable

Reset Value Not applicable

Errors and Events None

www.valuetronics.com

Dependencies None

Examples Command: `*SAV 4`

Related Commands SYSTem:SET
*LRN?
*RCL

*SRE
*SRE?

Sets or queries the Service Request Enable Register (SRER). The SRER determines which events in the Status Byte Register can generate an interrupt to the system controller. The SRER bits correspond to the bits in the Status Byte Register, which are described in Table 3–30 on page 3–266.

STATus:PRESet does not change the Service Request Enable Register. For more information on the Status and Events reporting system, refer to the Status and Events discussion on page 4–1.

Syntax

```
*SRE <mask>
*SRE?
```

Parameters

<mask>	Query Response
<NRf> | <Non-decimal numeric> 0 ≤ N ≤ #HFF	<NR1>

Reset Value Not applicable

Errors and Events None

Dependencies None

Examples Command: `*SRE #H60`

Query: `*SRE?`

Response: `96`

Related Commands *STB?

www.valuetronics.com

*STB? (Query Only)

Returns the contents of the Status Byte Register as a decimal sum of all set events. You can get a chronological list of events from the Status Queue using the SYSTem:ERRor? query. Table 3–30 describes the bit assignments in the Status Byte Register. For a complete description of the status and events reporting system, see page 4–1.

Table 3–30: The Status Byte Register

Bit	Decimal Value	Function
0–1	–	Not used.
2	4	**Error/Event Queue not empty** indicates that the error/event queue contains information and is waiting to be read.
3	8	**Questionable Event Status** indicates that the quality of result data or of an operation is questionable.
4	16	**Message Available (MAV)** shows that data is available in the Output Queue.
5	32	**Event Status Bit** indicates that one or more events have occurred in the Standard Event Status Register.
6	64	**Master Summary Status (MSS)** is a summary bit that indicates other bits in the Status Byte Register are set.
7 (MSB)	128	**Operation Event Register** indicates that the waveform analyzer is busy performing a normal operation such as acquiring a waveform.

Syntax

```
*STB?
```

Parameters

<event>	Query Response
Not applicable	<NR1>

Reset Value

Not applicable

Errors and Events

None

Dependencies

Reading this register does not clear it.

Examples

Query: `*STB?`

www.valuetronics.com

Response: 96
This response indicates the MAV and the MSS bits are set.

Related Commands *SRE

*TRG

Arms the waveform analyzer to accept triggers from the VXIbus. Before using *TRG, you must start the Arm/Trigger system with the INITiate command and use ARM:SOURce to select BUS.

Syntax *TST?

Parameters

<failure_code>	Query Response
Not applicable	<NR1> 0 | 1000 ≤ N ≤ 2999

Reset Value Not applicable

Errors and Events Execution Error –212, “Arm ignored”
Sent the *TRG command or the VXIbus word serial command <Trigger> when ARM:SOURce was not set to BUS or when the instrument was not in the “Waiting for Arm” state.

Dependencies None

Examples Query: *TRG

Related Commands INITIATE
ARM:SOURCE

www.valuetronics.com

*TST? (Query Only)

Initiates an internal self-test and returns a failure code. If more than one failure occurred, only the test number of the first failure is reported. A value of zero indicates there were no failures. This query is only valid when looping is disabled. Detailed test results are available with the TEST:RESult:VERBose? command.

*TST? and TEST? perform the same function.

Syntax

```
*TST?
```

Parameters

<failure_code>	Query Response
Not applicable	<NR1> 0 (no failure) 1000 – 1999 (Self-test failure) 2000 – 2999 (Calibration failure)

Reset Value

Not applicable

Errors and Events

Execution Error –221, “Settings conflict.”
Executed *TST? when TEST:CONTrol:LOOP is ON.

Dependencies

None

Examples

Query: `*TST?`

Response: 0

Related Commands

TEST
TEST:RESults:VERBose?

www.valuetronics.com

*WAI

Synchronizes command execution with the system controller. *WAI prevents the waveform analyzer from executing further commands until the active command completes execution.

Syntax `*WAI`

Parameters None

Reset Value Not applicable

Errors and Events None

Dependencies None

Examples Command: `*WAI`

Related Commands *OPC

www.valuetronics.com

Status and Events

The Status and Event reporting system reports asynchronous events and errors that occur in the TVS600. This system consists of status registers and message queues that you access through the command language. Use these status registers and message queues to determine when an error has occurred or when a function completes.

This section describes each status register and message queue, the Status and Event reporting process, and how to synchronize programming. This section ends with a list of the system messages.

Registers

The registers in the Status and Event reporting system fall into three functional groups:

- Status Registers contain information about errors and normal operations in the waveform analyzer. The status registers latch their event status and are cleared when read.
- Enable Registers determine whether selected types of events are reported to the Status Registers and the Event Queue. The enable registers must be cleared by setting with the appropriate values.
- Condition Registers provide access to the event status lines before they enter the Enable and Status Registers. Only the OPERation and QUEStionable SCPI registers have Condition registers.

A set of commands for each register allow you to read its status and set the enable register. Figure 4–1 illustrates how the registers are connected and gives the commands that control them.

www.valuetronics.com

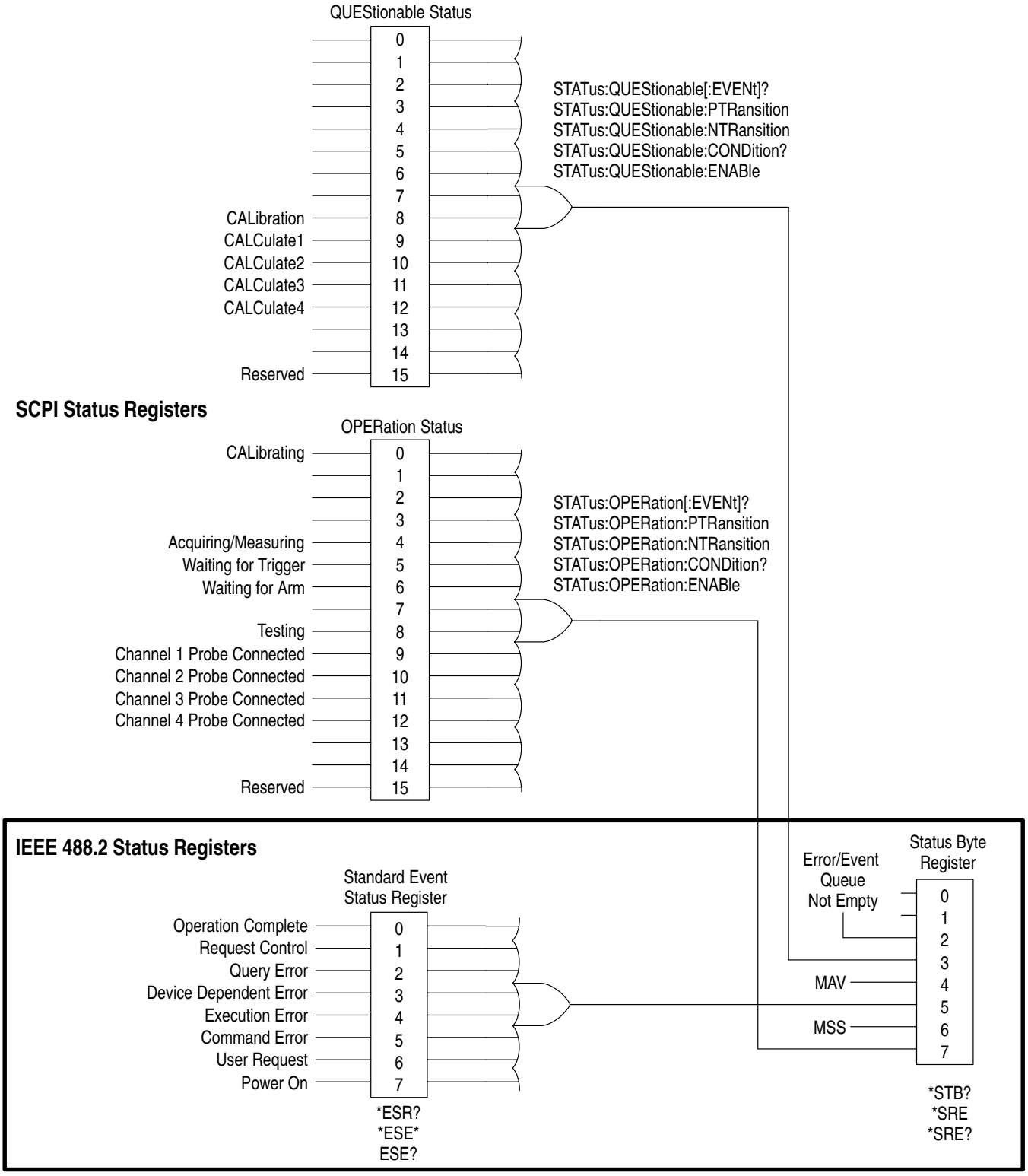


Figure 4–1: SCPI & IEEE 488.2 Status and Event Registers

www.valuetronics.com

Status Byte Register

The Status Byte Register, defined in Table 4–1, summarizes information from other registers and indicates when message data is in the Status Byte or Message queue. Refer to Figure 4–1 to see how the other status registers connect to the Status Byte Register. To read the contents of the Status Byte Register, use the *STB? query. The response is the sum of the decimal values for all true (1) bits. The *STB? query *does not* clear the Status Byte Register.

Table 4–1: The Status Byte Register

Bit	Decimal Value	Function
0–1	–	Not used.
2	4	**Error/Event Queue not empty** indicates that the error/event queue contains information and is waiting to be read.
3	8	**Questionable Event Status** indicates that the quality of result data or of an operation is questionable.
4	16	**Message Available (MAV)** shows that data is available in the Output Queue.
5	32	**Event Status Bit** indicates that one or more events have occurred in the Standard Event Status Register.
6	64	**MSS** (Master Summary Status) is a summary bit that indicates other bits in the Status Byte Register are set.
7 (MSB)	128	**Operation Event Register** indicates that the waveform analyzer is busy performing a normal operation such as acquiring a waveform.

Service Request Enable Register

The Service Request Enable Register (SRER) controls which bits in the Status Byte Register generate a service request. The bits in the SRER correspond to those in the Status Byte Register defined in Table 4–1. To set bits in the Service Request Enable Register, use the *SRE command. Set a SRER bit off (0) to disable its event from generating an interrupt. To see which bits are disabled, use the *SRE? query. The *SRE query returns the decimal sum for all set bits.

If, for example, the *SRE? query returns a value of 48, bits 4 and 5 are set in the Service Request Enable Register. All other bits are zero (0) and their events are disabled. Any event that sets the Message Available bit 4 or the Event Status bit 5 in the Status Byte Register will generate an interrupt. Other masked events, such as Questionable Status events, are prevented from generating an interrupt. To disable all interrupts except the Event Status bit 5, you would use the command *SRE 32.

Standard Event Status Register

The Standard Event Status Register,defined in Table 4–2, records eight types of events that may occur in the waveform analyzer. To read the contents of the Standard Event Status Register, use the *ESR? query. The response is the sum of the decimal values for all true (1) bits. The *ESR? query clears all bits in the

www.valuetronics.com

Standard Event Status Register. The Event Status Enable Register allows you to disable bits for specific events. Additionally, you can select which events cause an event message in the Status Queue with the command STATus:SESR:QENable.

Table 4–2: The Standard Event Status Register

Bit	Decimal Value	Function
0	1	**Operation Complete** shows that the operation is complete. This bit is set when all pending operations complete following an *OPC command.
1	2	**Request Control** (not used)
2	4	**Query Error** shows that the waveform analyzer attempted to read the Output Queue when no data was present or pending, or that data in the Output Queue was lost.
3	8	**Device Dependent Error** shows that a device error occurred. Table 4–10 on page 4–14 lists the device error messages.
4	16	**Execution Error** shows that an error occurred while the waveform analyzer was executing a command or query. Table 3–2 on page 4–13 lists the execution error messages.
5	32	**Command Error** shows that an error occurred while the waveform analyzer was parsing a command or query. Table 4–8 on page 4–12 lists the command error messages.
6	64	**User Request** indicates that a probe ID button was pressed.
7	128	**Power On** shows that the waveform analyzer was powered on.

The following example assumes that all bits have been enabled using the Event Status Enable Register (see the next topic). If an *ESR? query returns a value of 128, bit 7 is set (1) which indicates the instrument is in the initial power-on state and no other event bits are set.

Event Status Enable Register

The Event Status Enable Register (ESER) controls which events are summarized in bit 5 of the Status Byte Register. The bits in the ESER correspond to those in the Standard Event Status Register, which are defined in Table 4–2. Use the *ESE command to set bits in the Event Status Enable Register. Set an ESER bit off (0) to disable its event in the Standard Event Status Register. To see which bits are disabled, use the *ESE? query. The *ESE? query returns the decimal sum for all set bits.

For example, if the *ESE? query returns a value of 255, all bits are true (1) indicating that all events set the event status bit (bit 5) of the Status Byte Register. To disable Execution Errors, you could send the command *ESE 239, which disables only bit 4, Execution Errors.

www.valuetronics.com

Operation Status Register

TheOperation Status Register (OSR), defined in Table 4–3, identifies normal waveform analyzer events that are still in progress, such as acquisition in progress and waiting for a trigger. When an Operation event sets a bit true, the summary output sets bit 7 true in the Status Byte Register.

To read the contents of the OSR, use the STATus:OPERation? query. The response is the sum of the decimal values for all set (1) bits. Reading the OSR clears all bits.

Table 4–3: The Operation Status Register

Bit	Decimal Value	Function
0	1	**Calibrating** shows that a calibration routine is in progress.
1–3		Not used.
4	16	**Measuring/Acquiring** shows that measurement or acquisition is in progress.
5	32	**Waiting for Trigger** shows that the acquisition system is armed and waiting for a trigger event.
6	64	**Waiting for Arm** shows that the acquisition system has been initialized with INIT and is waiting to be armed.
7		Not used.
8	256	**Testing** shows that a self test routine is in progress.
9	512	**CH 1 Probe** shows that a probe is attached to the Channel 1 input.
10	1024	**CH 2 Probe** shows that a probe is attached to the Channel 2 input.
11	2048	**CH 3 Probe** shows that a probe is attached to the Channel 3 input.
12	4096	**CH 4 Probe** shows that a probe is attached to the Channel 4 input.
13–15		Not used.

Table 4–4 describes the control registers for the OSR. The control registers allow you to determine what is reported with the STATus:OPERation? query and what event can set the Operation bit (bit 7) in the Status Byte Register.

www.valuetronics.com

Table 4–4: Control Registers for the Operation Status Register

Control Register	Description	Control Commands	Affects of STATus:PRESet
Operation Status	Records the status of normal operating events. The Positive and Negative Transition registers will affect which events are stored in the Operation Status register.	STATus:OPERation?	None
Operation Condition	Provides the current state of Operation event lines prior to the Transition registers.	STATus:OPERation:CONDition?	None
Positive Transition	Reports an event in the Operation Status Register when the Condition Register event changes from false to true (0 to 1).	STATus:OPERation:PTRansition STATus:OPERation:PTRansition?	All 1's (reports all events on positive transitions)
Negative Transition	Reports an event in the Operation Status Register when the Condition Register event changes from true to false (1 to 0).	STATus:OPERation:NTRansition STATus:OPERation:NTRansition?	All 0's (reports no events on negative transitions)
Operation Enable	Disables events in the Operation Status Register from setting the summary bit 7 in the Status Byte Register. Setting a bit to one (1) enables that event, and setting it to zero (0) disables it.	STATus:OPERation:ENABle STATus:OPERation:ENABle?	All 0's (all events are disabled)
Status Queue Enable Positive Transition	Determines if transitions from false to true (0 to 1) on event lines in the Operation Condition register will generate a message in the Status Queue.	STATus:OPERation: QENable:PTRansition STATus:OPERation: QENable::PTRansition?	All 0's (all events are disabled)
Status Queue Enable Negative Transition	Determines if transitions from true to false (1 to 0) on event lines in the Operation Condition register will generate a message in the Status Queue.	STATus:OPERation: QENable:NTRansition STATus:OPERation: QENable::NTRansition?	All 0's (all events are disabled)

Questionable Status Register

The Questionable Status Register, (QSR) defined in Table 4–5, identifies operations whose results are questionable. To read the contents of the QSR, use the STATus:QUEStionable? query. The response is the sum of the decimal values for all set (1) bits. Reading the QSR clears all bits. Use the Questionable Status Enable Register to enable or disable specific events. In addition, the Questionable Status Condition Register provides access to the current state of these events.

An example of a questionable condition is when the waveform analyzer is due for self calibration because of changes in the ambient temperature. Because the waveform analyzer is in need of calibration, any data you acquire will be of questionable quality.

www.valuetronics.com

Table 4–5: The Questionable Status Register

Bit	Decimal Value	Function
0–7		Not used.
8	256	**Calibration** indicates that calibration is required due to the change in ambient temperature.
9	512	**Calculate1** indicates that source data contained a value that was overrange or underrange, making the results of the CALC1 block questionable.
10	1024	**Calculate2** indicates that source data contained a value that was overrange or underrange, making the results of the CALC2 block questionable.
11	2048	**Calculate3** indicates that source data contained a value that was overrange or underrange, making the results of the CALC3 block questionable.
12	4096	**Calculate4** indicates that source data contained a value that was overrange or underrange, making the results of the CALC4 block questionable.
13–15		Not used.

Table 4–6 describes the control registers for the QSR. The control registers allow you to determine what is reported with the STATus:QUEStionable? query and what event sets the Questionable bit (bit 3) in the Status Byte Register.

Table 4–6: Control Registers for the Questionable Status Register

Control Register	Description	Control Commands	Affects of STATus:PRESet
Questionable Status	Records the status of normal operating events. The Positive and Negative Transition registers will affect which events are stored in the Questionable Status Register.	STATus:QUEStionable?	None
Questionable Condition	Provides the current state of Questionable event lines prior to the Transition registers.	STATus:QUEStionable:CONDition?	None
Positive Transition	Reports an event in the Questionable Status Register when the Condition Register event changes from false to true (0 to 1).	STATus:QUEStionable:PTRansition STATus:QUEStionable:PTRansition?	All 1's (reports all events on positive transitions)
Negative Transition	Reports an event in the Questionable Status Register when the Condition Register event changes from true to false (1 to 0).	STATus:QUEStionable:NTRansition STATus:QUEStionable:NTRansition?	All 0's (reports no events on negative transitions)

www.valuetronics.com

Table 4–6: Control Registers for the Questionable Status Register (Cont.)

Control Register	Description	Control Commands	Affects of STATus:PRESet
Questionable Enable	Disables events in the Questionable Status Register from setting the summary bit 3 in the Status Byte Register. Setting a bit to one (1) enables that event, and setting it to zero (0) disables it.	STATus:QUEStionable:ENABle STATus:QUEStionable:ENABle?	All 0's (all events are disabled)
Status Queue Enable Positive Transition	Determines if transitions from false to true (0 to 1) on event lines in the Questionable Condition register will generate a message in the Status Queue.	STATus:QUEStionable: QENable:PTRansition STATus:QUEStionable: QENable::PTRansition?	All 0's (all events are disabled)
Status Queue Enable Negative Transition	Determines if transitions from true to false (1 to 0) on event lines in the Questionable Condition register will generate a message in the Status Queue.	STATus:QUEStionable: QENable:NTRansition STATus:QUEStionable: QENable::NTRansition?	All 0's (all events are disabled)

Queues

The waveform analyzer has two message queues. The Status Queue stores error and event messages that result from incorrect commands or waveform analyzer operations. When you send the STATus:QUEue? query, messages are stored temporarily in the second message queue, the Output Queue. The Output queue sets the Message Wating (MAV) bit four the Status Byte Register and waits for a VXIbus read command.

The Status Queue

The Status and Event reporting system stores event messages in the Status Queue. The Status Queue stores events, mainly errors, until the queue memory is filled. Events are stored in first-in, first-out order. When the queue overflows, the last event message is replaced with the device specific error

```
-350, "Too many errors".
```

To retrieve the first (and oldest) event message, send the command SYSTem:ERRor?. The returned event message contains the event number and a text description of the event such as error 350 above. Reading an event removes it from the queue. The commands described in Table 4–7 control the Status Queue and the responses to your event query.

Table 4–7: Commands Associated with the Status Queue

Command	Description
SYSTem:ERRor?	Returns the next event including the error code and text.
SYSTem:ERRor:ALL?	Returns all events including the error code and text.

www.valuetronics.com

Table 4–7: Commands Associated with the Status Queue (Cont.)

Command	Description
SYSTem:ERRor:CODE?	Returns only the error code for the next event.
SYSTem:ERRor:CODE:ALL?	Returns only the error codes for all events.
SYSTem:ERRor:COUNt?	Returns the number of events stored in the Status Queue.

You can control which register events send a message to the Status Queue. The following registers have commands that control event queueing:

- Standard Event Status Register provides the command STATus:SESR:QENable so you can disable queuing for one or more types of events.
- Operation Status Register provides the commands STATus:OPERation:QENable:PTRansition and :NTRansition so you can disable queueing for positive or negative transitions of event lines. Event lines are monitored at the Operation Condition Register.
- Questionable Status Register provides the commands STATus:QUEStionable:QENable:PTRansition and :NTRansition so you can disable queueing for positive or negative transitions of event lines. Event lines are monitored at the Questionable Condition Register.

The Output Queue

The waveform analyzer temporarily stores query responses in the Output Queue. When the Output queue has a message, it sets bit 4 (MAV) in the Status Byte Register which you can read with the command *STB.

The Output Queue is emptied each time you send a new command or query message after it receives an End Of Message (EOM). Therefore, if the controller does not read a query response from the Output Queue before it sends the next command (or query), it will lose the response to the previous query.

www.valuetronics.com

Status and Event Reporting Process

Figure 4–2 shows how to use the Status and Event Reporting system. In the explanation that follows, numbers in parentheses refer to the circled numbers in Figure 4–2.

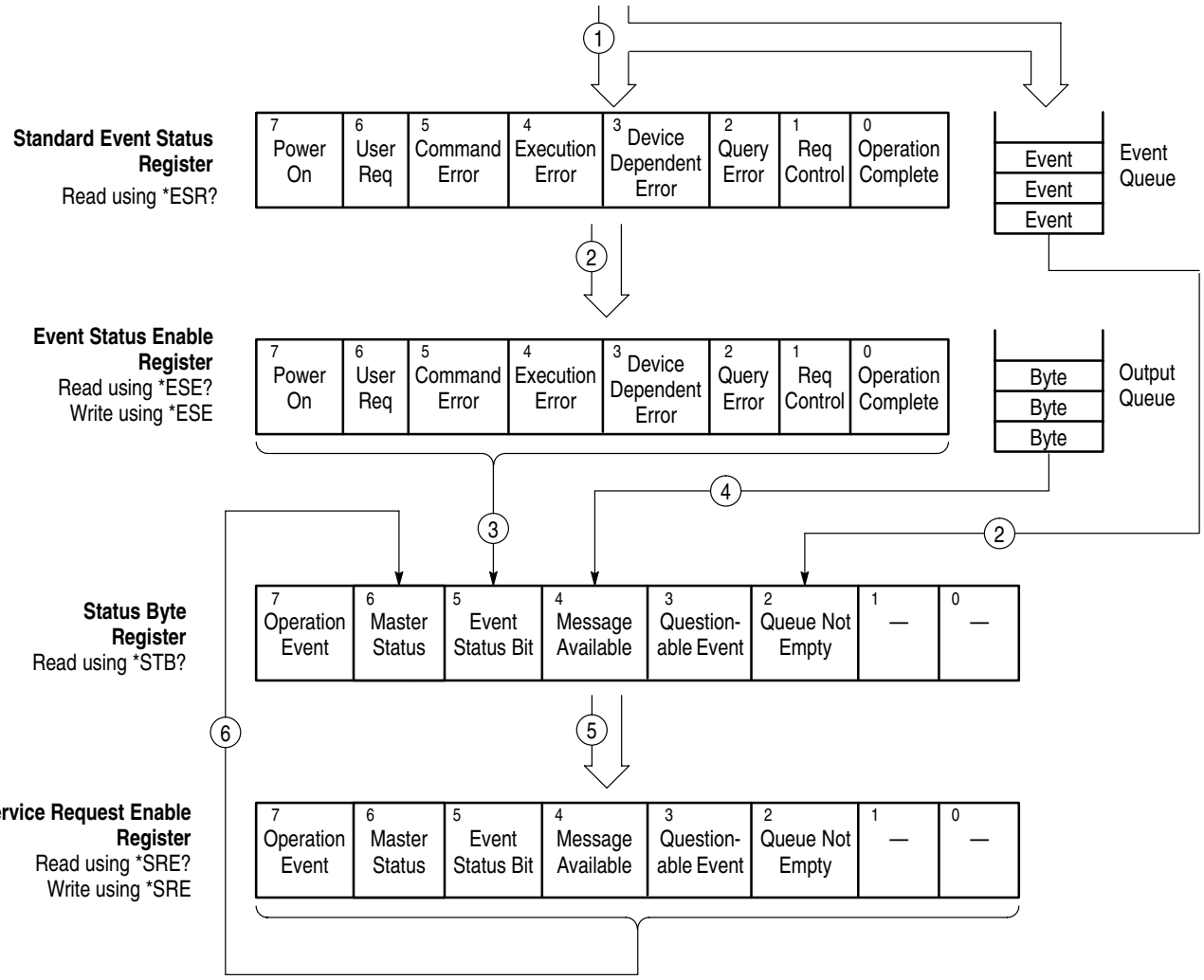


Figure 4–2: Status and Event Reporting Process

When an event occurs the appropriate bit in the Standard Event Status Register is set to one and the event is recorded in the Event Queue (1). If the corresponding bit in the Event Status Enable Register is also enabled (2), then the event status bit in the Status Byte Register is set to one (3). When an event enters the Event Queue, it sets the Queue Not Empty bit in the Status Byte Register (2).

www.valuetronics.com

When the Status Reporting System sends output to the Output Queue (for example, a response to a query), it sets the message available bit in the Status Byte Register to one (4).

When a bit in the Status Byte Register is set to one and the corresponding bit in the Service Request Enable Register is enabled (5), the master status summary bit in the Status Byte Register is set to one (6).

Synchronization Methods

Although most commands are completed soon after being received by the waveform analyzer, some commands start processes requiring a longer period. For example, after you send the INITiate command, you must wait until acquisition completes before you give another command or query.

Sometimes the result of an operation depends on the result of an earlier operation (the first operation must be completed before the next one is initiated). One example is performing a calculation after a waveform acquisition. The status and event reporting system provides this capability.

Using the *OPC? Query

Use the *OPC? query to synchronize commands. The *OPC? query places a 1 in the Output Queue once an operation is complete. A time out could occur if you try to read the output queue before any data is in it.

The same command sequence using the *OPC? query for synchronization looks like this:

```
/* Set up a chained message */
INITiate;*OPC?
```


www.valuetronics.com

Error Messages

The waveform analyzer generates error messages in response to events caused by commands or queries. Each type of event sets a specific bit in the Standard Event Status Register. Thus, each message is associated with a specific Standard Event Status Register bit. In the message tables that follow, the associated Standard Event Status Register bit is specified in the table title. Not shown in the tables are secondary messages giving more detail about the cause of the error or the meaning of the message. These secondary messages are shown for each command and query in *Syntax and Commands*.

Table 4–8 shows the error messages generated by improper command syntax. Check to see that the command is properly formatted and that it follows the rules in *Syntax and Commands*.

Table 4–8: Command Error Messages (Bit 5 in Standard Event Status Register)

Code	Message
100	Command error
101	Invalid character
102	Syntax error
103	Invalid separator
104	Data type error
105	Get not allowed
106	Invalid program data separator
108	Parameter not allowed
109	Missing parameter
110	Command header error
111	Header separator error
112	Mnemonic too long
113	Undefined header
118	Query not allowed
120	Numeric data error
121	Invalid char in number
123	Exponent too large
124	Too many digits
128	Numeric data not allowed
130	Suffix error
131	Invalid suffix
134	Suffix too long

www.valuetronics.com

Table 4–8: Command Error Messages (Bit 5 in Standard Event Status Register) (Cont.)

Code	Message
138	Suffix not allowed
140	Character data error
141	Invalid character data
144	Character data too long
148	Character data not allowed
150	String data error
151	Invalid string data
158	String data not allowed
160	Block data error
161	Invalid block data
168	Block data not allowed

Table 3–2 lists the execution error messages that can occur during execution of a command.

Table 4–9: Execution Error Messages (Bit 4 in Standard Event Status Register)

Code	Message
200	Execution error
220	Parameter error
221	Settings conflict
222	Data out of range
223	Too much data
224	Illegal parameter value
230	Data corrupt or stale
240	Hardware error
241	Hardware missing
250	Mass storage error
252	Missing mass storage
252	Missing media
253	Corrupt media
254	Media full
255	Directory full

www.valuetronics.com

Table 4–9: Execution Error Messages (Bit 4 in Standard Event Status Register) (Cont.)

Code	Message
256	File name not found
257	File name error
258	Media protected

Table 4–10 lists the device dependent error messages that can occur during waveform analyzer operation.

Table 4–10: Device Dependent Error Messages (Bit 3 in Standard Event Status Register)

Code	Message
300	Device specific error
310	System error
361	Autoscan failed

Table 4–11 lists the system events.

Table 4–11: System Events

Code	Message
401	Power on[1]
402	Operation complete[2]

1 Sets bit 7 in the Standard Event Status Register.

2 Sets bit 0 in the Standard Event Status Register.

Table 4–12 lists the execution warnings that can occur during execution of a command.

Table 4–12: Execution Warning Messages (Bit 3 in Standard Event Status Register)

Code	Message
500	Execution warnings

www.valuetronics.com

Appendix A: Specifications

This chapter provides a complete description of the waveform analyzer specifications. *Product Description* (below) is a general description of the instrument. The *Specification Tables*, which begin on page A–2, contain the complete specifications for the waveform analyzer.

Product Description

The TVS600 Series Waveform Analyzers are a family of C-size, double-wide VXI modules suitable for use in a variety of test and measurement applications and systems. Many key features are listed below:

- Four standard configurations with full-featured, 1 MΩ/50 Ω inputs: TVS641 and TVS645 have four channels, TVS621 and TVS625 have two channels.
- A maximum realtime digitizing rate up to 5 GSample/second with an analog bandwidth up to 1 GHz. See Table A–1 for details.

Table A–1: Comparison of Product Features

Product	Input Channels	Maximum Sample Rate	Analog Bandwidth
TVS645	4	5 GSample/second	1 GHz
TVS641	4	1 GSample/second	250 MHz
TVS625	2	5 GSample/second	1 GHz
TVS621	2	1 GSample/second	250 MHz

- A maximum record length of 30,000 samples with 8-bit vertical resolution.
- Full programmability using a SCPI command set.
- Acquisition modes such as sample, envelope, and average.
- A full complement of internal triggering modes such as edge and pulse, plus VXI backplane and external trigger sources.

www.valuetronics.com

Specification Tables

This section contains tables that list the specifications for the waveform analyzer. All specifications are guaranteed unless noted "typical." Specifications that are marked with the ✓ symbol are checked in the *Performance Verification* chapter of the Service Manual (an optional accessory).

The performance limits in this specification are valid with these conditions:

- The waveform analyzer must have been calibrated/adjusted at an ambient temperature between +20° C and +30° C.
- The waveform analyzer must be in an environment with temperature, altitude, humidity, and vibration within the operating limits described in these specifications.
- The waveform analyzer must have had a warm-up period of at least 20 minutes.
- The waveform analyzer must have had its signal-path-compensation routine (self cal) last executed after at least a 20 minute warm-up period at an ambient temperature within ±5° C of the current ambient temperature.

Table A–2: Signal Acquisition System

Name	Description	
✓ Accuracy, DC Gain	±1.5% for full scale ranges from 20 mV to 100 V ±2.0% for full scale ranges <19.9 mV	
✓ Accuracy, DC Voltage Measurement	±(1.5% of input signal + 1% of full scale range) with instrument temperature within 5° C of Setting-Cal temperature for input ranges ≥50 mV full scale	
✓ Accuracy, Delta DC Voltage Measurement	±(1.5% of input signal + 0.1% of full scale range) with instrument temperature within 5° C of Setting-Cal temperature	
✓ Accuracy, Offset[1]	*Full Scale Range Setting*	*Offset Accuracy*
	10 mV – 1 V	±[(0.2% × | offset |) + 1.5 mV + (6% × full scale range)]
	1.01 V – 10 V	±[(0.25% × | offset |) + 15 mV + (6% × full scale range)]
	10.1 V – 100 V	±[(0.25% × | offset |) + 150 mV + (6% × full scale range)]

[1] **Net offset is the nominal voltage level at the waveform analyzer input that corresponds to the center of the A/D Converter dynamic range. Offset accuracy describes the precision of the net offset voltage.**

www.valuetronics.com

Table A–2: Signal Acquisition System (Cont.)

Name	Description		
✓ Analog Bandwidth, DC–50 Ω Coupled or DC–1 MΩ Coupled	*Full Scale Range Setting*	*Bandwidth*[2]	
	10.1 V – 100 V	DC – 500 MHz (TVS625 and TVS645) DC – 250 MHz (TVS621 and TVS641)	
	100 mV – 10 V	DC – 1 GHz (TVS625 and TVS645) DC – 250 MHz (TVS621 and TVS641)	
	50 mV – 99.8 mV	DC – 900 MHz (TVS625 and TVS645) DC – 250 MHz (TVS621 and TVS641)	
	20 mV – 49.8 mV	DC – 600 MHz (TVS625 and TVS645) DC – 250 MHz (TVS621 and TVS641)	
	10 mV – 19.8 mV	DC – 500 MHz (TVS625 and TVS645) DC – 250 MHz (TVS621 and TVS641)	
Bandwidth, Analog, Selections	*Products*	*Bandwidth Selections*	
	TVS625 and TVS645	20 MHz, 250 MHz, and FULL	
	TVS621 and TVS641	20 MHz, 180 MHz, and FULL	
Calculated Rise Time, typical[3]	*Full Scale Range Setting*	*TVS625 and TVS645*	*TVS621 and TVS641*
Typical full-bandwidth rise times are shown in the chart to the right	10.1 V – 100 V	900 ps	1.8 ns
	100 mV – 10 V	450 ps	1.8 ns
	50 mV – 99 mV	500 ps	1.8 ns
	20 mV – 49.9 mV	750 ps	1.8 ns
	10 mV – 19.9 mV	900 ps	1.8 ns
Crosstalk (Channel Isolation)	≥300:1 at 100 MHz and ≥100:1 at the rated bandwidth for the channel's sensitivity (Full Scale Range) setting, for any two channels having equal sensitivity settings		
✓ Delay Between Channels, Full Bandwidth	≤100 ps with equal Full Scale Range and Coupling settings		
Digitized Bits, Number of	8 bits		
Frequency Limit, Upper, 20 MHz Bandwidth Limited, typical	20 MHz		
Frequency Limit, Upper, 250 MHz Bandwidth Limited, typical	*Products*	*Bandwidth*	
	TVS625 and TVS645	250 MHz	
	TVS621 and TVS641	180 MHz	

2 The limits given are for the ambient temperature range of 0° C to +30° C. Reduce the upper bandwidth frequencies by 5 MHz for each °C above +30° C. The bandwidth must be set to FULL.

3 Rise time (rounded to the nearest 50 ps) is calculated from the bandwidth when Full Bandwidth is selected. It is defined by the following formula:

Rise Time (ns) = 450 ÷ BW (MHz)

www.valuetronics.com

Table A–2: Signal Acquisition System (Cont.)

Name	Description	
Input Channels, Number of	*Product*	*Channels*
	TVS645	Four
	TVS641	Four
	TVS625	Two
	TVS621	Two
Input Coupling	DC, AC, or GND[4]	
Input Impedance, DC–1 MΩ Coupled	1 MΩ ±0.5% in parallel with 10 pF ±3 pF	
Input Impedance Selections	1 MΩ or 50 Ω	
Input Resistance, DC–50 Ω Coupled	50 Ω ±1%	
Input VSWR, DC–50 Ω Coupled	≤1.3:1 from DC – 500 MHz, ≤1.5:1 from 500 MHz – 1 GHz	
Input Voltage, Maximum, DC–1 MΩ, AC–1 MΩ, or GND Coupled	The greater of ±300 Vrms or DC, derated at 20 dB/decade above 1 MHz CAT II (see *Overvoltage Category Descriptions* on page A–13 for more information)	
Input Voltage, Maximum, DC–50 Ω or AC–50 Ω Coupled	5 V_{RMS}, with peaks ≤ ±25 V	
Lower Frequency Limit, AC Coupled, typical	≤10 Hz when AC–1 MΩ Coupled; ≤200 kHz when AC–50 Ω Coupled[5]	
Random Noise	*Bandwidth Selection*	*RMS Noise*
	Full	≤(350 μV + 0.5% of the Full Scale Range setting)
	250 MHz	≤(165 μV + 0.5% of the Full Scale Range setting)
	20 MHz	≤(75 μV + 0.5% of the Full Scale Range setting)
Range, Offset	*Full Scale Range Setting*	*Offset Range*
	10 mV – 1 V	±1 V
	1.01 V – 10 V	±10 V
	10.1 V – 100 V	±100 V
Range, Sensitivity (Full Scale Range), All Channels	10 mV to 100 V[6]	

4 GND input coupling disconnects the input connector from the attenuator and connects a ground reference to the input of the attenuator.

5 The AC Coupled Lower Frequency Limits are reduced by a factor of 10 when 10X passive probes are used.

6 The sensitivity ranges are 10 mV to 100 V full scale, switching in a 1–2–5 sequence of coarse settings. Between these coarse settings, you can adjust the sensitivity with a resolution equal to 1% of the more sensitive coarse setting. For example, between the 500 mV and 1 V ranges, the sensitivity can be set with 5 mV resolution.

www.valuetronics.com

Table A–2: Signal Acquisition System (Cont.)

Name	Description				
Step Response Settling Errors, typical[7]	*Full Scale Range Setting*	*± Step Response*	*Maximum Settling Error (%) at 20 ns*	*100 ns*	*20 ms*
	10 mV – 1 V	≤2 V	0.5%	0.2%	0.1%
	1.01 V – 10 V	≤20 V	1.0%	0.5%	0.2%
	10.1 V – 100 V	≤200 V	1.0%	0.5%	0.2%

Table A–3: Timebase System

Name	Description	
✓ Accuracy, Long Term Sample Rate and Delay Time	±100 ppm over any interval ≥1 ms	
Range, Extended Realtime Sampling Rate	5 S/s to 10 MS/s in a 1–2.5–5 sequence	
Range, Realtime Sampling Rate	*Products*	*Limits*
	TVS625 and TVS645	20 MS/s to 5 GS/s on all channels simultaneously in a 1–2.5–5 sequence
	TVS621 and TVS641	20 MS/s to 1 GS/s on all channels simultaneously in a 1–2.5–5 sequence
Record Length	256, 512, 1024, 2048, 4096, 8192, 15,000 30,000 (extended realtime sampling mode only)	
Time Stamping	125 ns resolution 0.1% variance	

7 **The Full Bandwidth settling errors are typically less than the percentages from the table.**

www.valuetronics.com

Table A–4: Trigger System

Name	Description	
✓ Accuracy (Time) for Pulse Glitch or Pulse Width Triggering	*Time Range*	*Accuracy*
	1 ns to 1 µs	±(20% of Setting + 0.5 ns)
	1.02 µs to 1 s	±(204.5 ns + 0.01% of Setting)
✓ Accuracy (DC) for External Trigger Level	±(5% + 150 mV) for signals having rise and fall times ≥20 ns	
✓ Accuracy (DC) for Internal Trigger Level, DC Coupled	±[(2% × | Setting) | + 0.03 of Full Scale Range + Offset Accuracy)] for signals having rise and fall times ≥20 ns	
Holdoff, Variable Main Trigger, typical[1]	For all sampling rates, the minimum holdoff is 250 ns and the maximum holdoff is 12 s; the minimum resolution is 8 ns for settings ≤1.2 µs	
Input, External Trigger, typical	50 Ω input resistance; ±5 V (DC + peak AC) maximum safe input voltage; DC coupled only	
Range, Delayed Trigger Time[2]	16 ns to 250 s	
Range, Events Delay	1 to 10,000,000	
Range (Time) for Pulse Glitch and Pulse Width Triggering	1 ns to 1 s	
Range, Trigger Level	*Source*	*Range*
	Any Channel	±100% of full scale range
	External Input	±1 V
Range, Trigger Point Position	Minimum: 0 Maximum: 30,000	
Resolution, Trigger Level	0.2% of full scale for any Channel source and 2 mV for the External Input source	
Resolution, Trigger Position	One sample interval at all sample rates	
Sensitivities, Pulse-Type Trigger and Events Delay, DC Coupled, typical	10% of full scale, from DC to 500 MHz, for Full Scale Range settings >100 mV and ≤10 V at the BNC input	
Sensitivities, Pulse-Type Trigger Width and Glitch, typical	10% of full scale, for Full Scale Range settings >100 mV and ≤10 V at the BNC input	

1 **Main Trigger is controlled with the TRIGger:A commands.**

2 **Delayed Trigger is controlled with the TRIGger:B commands.**

www.valuetronics.com

Table A–4: Trigger System (Cont.)

Name	Description			
Sensitivity, Edge-Type Trigger, DC Coupled[3]	The minimum signal levels required for stable edge triggering of an acquisition when the source is DC-coupled.			
	Products	*Trigger Source*	*Sensitivity*	
	TVS625 and TVS645	Any Channel	3.5% of Full Scale Range from DC to 50 MHz, increasing to 10% of Full Scale Range at 1 GHz	
	TVS621 and TVS641	Any Channel	3.5% of Full Scale Range from DC to 50 MHz, increasing to 10% of Full Scale Range at 250 MHz	
	TVS621, TVS625, TVS641, and TVS645	External	25 mV from DC to 50 MHz, increasing to 50 mV at 100 MHz	
Sensitivity, Edge-Type Trigger, Not DC Coupled, typical	*Trigger Coupling*		*Typical Signal Level for Stable Triggering*	
	AC		Same as the DC-coupled limits for frequencies above 60 Hz; attenuates signals below 60 Hz	
	High Frequency Reject		One and one-half times the DC-coupled limits from DC to 30 kHz; attenuates signals above 30 kHz	
	Low Frequency Reject		One and one-half times the DC-coupled limits for frequencies above 80 kHz; attenuates signals below 80 kHz	
	Noise Reject		Three times the DC-coupled limits	
Time, Minimum Pulse or Rearm, and Minimum Transition Time, for Pulse-Type Triggering, typical	For Full Scale Range settings >100 mV and ≤10 V at the BNC input			
	Pulse Class	*Minimum Pulse Width*	*Minimum Rearm Width*	
	Glitch	1 ns	2 ns + 5% of Glitch Width Setting	
	Width	1 ns	2 ns + 5% of Width Upper Limit Setting	
Time, Minimum Pulse or Rearm, for Events Delay Triggering, typical	The following chart shows the minimum values for input range settings >100 mV and ≤10 V at the BNC input			
	Triggering Type	*Minimum Pulse Width*	*Minimum Rearm Time*	*Minimum Time Between Channels*[4]
	Events Delay	1 ns (for either + or – pulsewidths)	N/A	2 ns

[3] **Delayed Trigger has the same specifications as Main Trigger.**

[4] **For Events Delay, the time is the minimum time between a main and delayed event that will be recognized if more than one channel is used.**

www.valuetronics.com

Table A–4: Trigger System (Cont.)

Name	Description	
Trigger Position Error, Edge Triggering, typical	*Acquisition Mode*	*Trigger Position Error*[5]
	Sample, Average	±(1 Sample Interval + 1 ns)
	Envelope	±(2 Sample Intervals + 2 ns)

[5] **The trigger position errors are typically less than the values given here. These values are for triggering signals having a slew rate at the trigger point of ≥5% of full scale/ns.**

www.valuetronics.com

Table A–5: Front Panel Connectors

Name	Description		
Arm Input	This input provides external arming capability with a BNC connector		
	Characteristic	*Limits*	
	Arming Threshold Voltage	≤0.8 V	
	Input Voltage Range	0 to 5 V_{pk}, TTL-compatible (arms on a switch closure to ground; internal pull-up resistor to +5 volts is provided)	
	Latency	10 µs	
	Minimum Pulsewidth	10 µs	
Fiducial Input, typical[1]	This input provides fiducial input capability with a BNC connector; the polarity of the signal acquired is inverted with respect to the input		
	Characteristic	*Limits*	
	Fiducial Input/CH 1 HF Gain Ratio	–6 ±25% (CH 1 set to 1 V Full Scale Range)	
	Input Impedance	0.01 µF in series with 50 Ω	
	Input LF Attenuation	Attenuates signals below 100 MHz (highpass time constant of 5 ns)	
	Input Sensitivity	*CH 1 Full Scale Range*	*Fiducial Full Scale Range*
		10 mV to 1 V	6 times the CH 1 Full Scale Range setting
		1.01 V to 10 V	0.6 times the CH 1 Full Scale Range setting
		10.1 V to 100 V	0.06 times the CH 1 Full Scale Range setting
	Input Voltage Range	±1 V	
	Maximum Input	2 V_{RMS}	
	Rise Time	*Products*	*Rise Time*
		TVS625 and TVS645	≤2.5 ns (10% to 90%)
		TVS621 and TVS641	≤3 ns (10% to 90%)

1 **The FIDUCIAL Input is designed for short-duration (≤3 ns) fast rise time (≤2 ns) pulse signals.**

www.valuetronics.com

Table A–5: Front Panel Connectors (Cont.)

Name	Description	
✓ Output, Reference	*Characteristic*	*Limits*
	Output Voltage	8 V ±1%
	Internal Frequency Reference	Frequency is 10 MHz ±0.015% Amplitude is ≥1 V_{p-p} into 50 Ω
Probe Compensation, Output Frequency, typical	1 kHz ±25%	
✓ Probe Compensation, Output Voltage	0.5 V (base-top) ±1% into a ≥50 Ω load	
Serial Interface	This front panel-mounted 9-pin D connector provides a serial interface with the following pin assignments: 1 DCD 2 RXD 3 TXD 4 DTR 5 GND 6 DSR 7 RTS 8 CTS 9 No Connection	

www.valuetronics.com

Table A–6: VXI Interface

Name	Description	
Addressing	Dynamic autoconfigure or set manually	
Inputs, ECLTRG	Either of the two ECLTRG lines may be individually selected to arm or trigger an acquisition	
Inputs, TTLTRG	Any of the eight TTLTRG lines may be individually selected to arm or trigger an acquisition	
Interface Type	Message based (I4)	
Interrupts	Programmable interrupter level 1–7	
Outputs, ECLTRG	Either of the two ECLTRG lines can be driven by the following signals: ARM — The waveform analyzer is armed and waiting for a trigger ATR — Main trigger event has occurred BTR — Delayed trigger event has occurred OPC — Operation pending complete	
Outputs, TTL	Each of the TTLTRG lines (TTLTRG0*–TTLTRG7*) can be driven by the following signals: ARM — The waveform analyzer is armed and waiting for a trigger ATR — Main trigger event has occurred BTR — Delayed trigger event has occurred OPC — Operation pending complete	
Outputs, TTLTRG, Logic Levels	Based on the VXIbus Specification RULE B.6.17	
	Characteristic	*Limits*
	Vout(HI)	Determined by the mainframe termination; the TTL outputs are open collector
	Vout(LO)	≤0.6 V when sinking 48 mA
Outputs, TTLTRG, Logic Polarity	Normal polarity: Negative TRUE; high-to-low transition indicates the event occurred Inverted polarity: Positive TRUE; low-to-high transition indicates the event occurred	
Protocols	Word Serial (WSP) Fast Data Channel FDC TEK V2.1	
VXI Interface	Complies with revision 1.4	

www.valuetronics.com

Table A–7: Power Distribution and Data Handling

Name	Description		
Current Requirements, TVS641 and TVS645, typical	*Voltage*	*DC Current*	*Dynamic Current*
	+12 V	1.3 A	0.45 A
	+5 V	11.0 A	0.8 A
	–5.2 V	4.6 A	0.09 A
	–12 V	1.0 A	0.4 A
Current Requirements, TVS621 and TVS625, typical	*Voltage*	*DC Current*	*Dynamic Current*
	+12 V	1.0 A	0.45 A
	+5 V	8.0 A	0.7 A
	–5.2 V	2.7 A	0.05 A
	–12 V	0.8 A	0.4 A
Nonvolatile Memory Retention Time, typical[1]	Battery life is ≥5 years		
Power Requirements, typical	*Products*	*Power Requirements*	
	TVS641 and TVS645	106.5 Watts	
	TVS621 and TVS625	75.6 Watts	

Table A–8: Environmental

Name	Description
Airflow Resistance	≤0.178 mm H_2O air pressure with 6.6 l/s airflow
Altitude, Operating and Nonoperating	Operating: to 15,000 feet (4570 m) Nonoperating: to 40,000 feet (12,190 m)
Humidity, Operating and Nonoperating	To 95% relative humidity at or below +30° C; to 45% relative humidity up to +50° C
Temperature, Operating and Nonoperating	Operating: 0° C to +50° C for exterior air when operated in a mainframe with 15° C internal temperature rise, and airflow of 0.75 mm H_2O air pressure @ 2 l/s Nonoperating: –40° C to +71° C

1 **The time that reference waveforms, stored setups, and calibration constants are retained when there is no power to the waveform analyzer.**

www.valuetronics.com

Table A–9: Certifications and Compliances

Name	Description
Certifications	Underwriters Laboratories listed to CAN/CSA-C22.2 No.1010.1-92. Underwriters Laboratories listed to Standard UL3111-1 for Electrical and Electronic Measuring and Testing Equipment.
EC Declaration of Conformity	Meets intent of Directive 89/336/EEC for Electromagnetic Compatibility and Low Voltage Directive 73/23/ECC for Product Safety. Compliance was demonstrated to the following specifications as listed in the Official Journal of the European Communities: EMC Directive 89/336/EEC: EN 50081-1 Emissions: EN 55011 Class A Radiated and Conducted Emissions EN 60555-2 AC Power Line Harmonic Emissions EN 50082-1 Immunity: IEC 801-2 Electrostatic Discharge Immunity IEC 801-3 RF Electromagnetic Field Immunity IEC 801-4 Electrical Fast Transient/Burst Immunity Low Voltage Directive 73/23/EEC: EN 61010-1 Safety requirements for electrical equipment for measurement, control, and laboratory use
FCC Compliance	Emissions comply with FCC Code of Federal Regulations 47, Part 15, Subpart B, Class A Limits
Overvoltage Category Descriptions CAT I: Signal levels in special equipment or parts of equipment, telecommunications, electronics. CAT II: Local-level mains, appliances, portable equipment. CAT III: Distribution-level mains, fixed installation.	CAT II (see *Input Voltage, Maximum, DC–1 MΩ, AC–1 MΩ, or GND Coupled* on page A–4)
Pollution Degree 2	Do not operate in environments where conductive pollutants may be present.
Safety Certification of Plug-in or VXI Modules	For modules (plug-in or VXI) that are safety certified by Underwriters Laboratories, UL Listing applies only when the module is installed in a UL Listed product. For modules (plug-in or VXI) that have cUL or CSA approval, the approval applies only when the module is installed in a cUL or CSA approved product.

www.valuetronics.com

Table A–10: Mechanical

Name	Description	
Construction Material	Chassis parts constructed of aluminum alloy; front panel constructed of plastic laminate; circuit boards constructed of glass laminate; cabinet is aluminum	
Weight	*Products*	*Weight*
	TVS641 and TVS645	2.6 kg (5 lbs 12 oz)
	TVS621 and TVS625	2.5 kg (5 lbs 8 oz)
	All TVS 600, when packaged for domestic shipment	7.7 kg (17 lbs)
Overall Dimensions	Height: 262 mm (10.3 in) Width: 61 mm (2.4 in) Depth: 368 mm (14.5 in)	

www.valuetronics.com

Appendix B: Algorithms

The waveform analyzer can take many automatic measurements and perform a variety of other calculations. By knowing how they make these calculations, you may better understand how to use your waveform analyzer and how to interpret the results.

Measurement Variables

The waveform analyzer uses a variety of variables in its calculations. This section discusses each variable and how to set it.

High and Low

High is the value used as the 100% level in measurements such as fall time and rise time. For example, if you request the 10% to 90% rise time, then the waveform analyzer calculates 10% and 90% as percentages with High representing 100%.

Low is the value used as the 0% level in measurements such as fall time and rise time.

The exact meaning of High and Low depends on which calculation method you choose. To set the method used to determine High you use the CALCulate:WMParameter:HMEThod command. To set the method used to determine Low you use the CALCulate:WMParameter:LMEThod command. The methods are PEAK, MODE, AUTO, and ABSolute.

PEAK defines the 0% and the 100% waveform levels as the lowest amplitude (most negative) and the highest amplitude (most positive) samples. The PEAK method is useful for measuring frequency, width, and period for many types of signals. PEAK is sensitive to waveform ringing and spikes, however, and does not always measure accurately rise time, fall time, overshoot, and undershoot.

MODE attempts to find the highest density of points above and below the waveform midpoint. It attempts to ignore ringing and spikes when determining the 0% and 100% levels. This method works well when measuring square waves and pulse waveforms.

The waveform analyzer calculates the histogram-based High and Low values as follows:

1. It makes a histogram of the record with 256 bins.

www.valuetronics.com

2. It splits the histogram into two sections at the halfway point between Min and Max (also called Mid).

3. The level with the most points in the upper histogram is the High value, and the level with the most points in the lower histogram is the Low value. (Choose the levels where the histograms peak for High and Low.)

   If Mid gives the largest peak value within the upper or lower histogram, then return the Mid value for both High and Low (this is probably a very low amplitude waveform).

   If more than one histogram level (bin) has the maximum value, choose the bin farthest from Mid.

This algorithm does not work well for two-level waveforms with greater than about 100% overshoot.

AUTO attempts to use the MODE method but will switch to the PEAK method if the histogram does not show an obvious consistent high level. For example, the MODE histogram operating on a triangle wave would not find significant High and Low levels so AUTO would select PEAK. On a square wave AUTO would pick the MODE method.

ABSolute uses the absolute value set with the commands CALCulate:WMParameter:HIGH or CALCulate:WMParameter:LOW.

Measurement Reference Levels

You can set the various reference levels used to take the automated measurements. You can choose to set reference levels in absolute vertical units or in relative units of percent or ratio. Use the command CALCulate:WMParameter:RMEThod to choose the reference method. The reference levels are as follows:

HREFerence the waveform high reference or distal level. Used in fall time and rise time calculations. In the RELative mode, you use the command CALCulate:WMParameter:HREFerence:RELative to set it from 0% to 100%, with a reset value of 90%. You can set it to a voltage level with the command CALCulate:WMParameter:HREFerence.

MREFerence the waveform middle reference or mesial level. In the RELative mode, you use the command CALCulate:WMParameter:MREFerence:RELative to set MREFerence from 0% to 100%, with a reset value of 50%. You can set it to a voltage level with the command CALCulate:WMParameter:MREFerence. You can also specify a hysteresis value for the MREFerence that reduces the effects of noise on measurements. The HYSTeresis value is a percent or ratio of the AMPLitude value.

www.valuetronics.com

LREFerence the waveform low reference or proximal level. Used in fall time and rise time calculations. In the RELative mode, you use the command CALCulate:WMParameter:LREFerence:RELative to set it from 0% to 100%, with a reset value of 10%. You can set it to a voltage level with the command CALCulate:WMParameter:LREFerence.

Other Variables

The waveform analyzer also measures several values itself that it uses to help calculate measurements.

RecordLength is the number of data points in the time base.

Hysteresis reduces the affects of noise on measurements by providing a guard band of 5% (the default) of the waveform amplitude, above and below the midpoint value. Hysteresis can be set in the range 0% to 50%. It is used in MCross1, MCross2, and MCross3 calculations.

For example, once a crossing has been measured in a negative direction, the waveform data must fall below 5% of the amplitude from the MidRef point before the measurement system is armed and ready for a positive crossing. Similarly, after a positive MidRef crossing, waveform data must exceed 5% of the amplitude before a negative crossing can be measured. Hysteresis is useful when you are measuring noisy signals, because it allows the waveform analyzer to ignore minor fluctuations in the signal.

MCross Calculations

MCross1, MCross2, and MCross3 refer to the first, second, and third MidRef cross times, respectively. (See Figure B–1.)

The polarity of the crossings does not matter for these variables, but the crossings alternate in polarity; that is, MCross1 could be a positive or negative crossing, but if MCross1 is a positive crossing, MCross2 will be a negative crossing.

The waveform analyzer calculates these values as follows:

1. Find the first MidRefCrossing in the waveform record. This is MCross1.
2. Continuing from MCross1, find the next MidRefCrossing in the waveform record (or the gated region) of the opposite polarity of MCross1. This is MCross2.
3. Continuing from MCross2, find the next MidRefCrossing in the waveform record (or the gated region) of the same polarity as MCross1. This is MCross3.

www.valuetronics.com

MCross1Polarity is the polarity of first crossing (no default). It can be rising or falling.

StartCycle is the starting time for cycle measurements. It is a floating-point number with values between 0.0 and (RecordLength – 1.0), inclusive.

StartCycle = MCross1

EndCycle is the ending time for cycle measurements. It is a floating-point number with values between 0.0 and (RecordLength – 1.0), inclusive.

EndCycle = MCross3

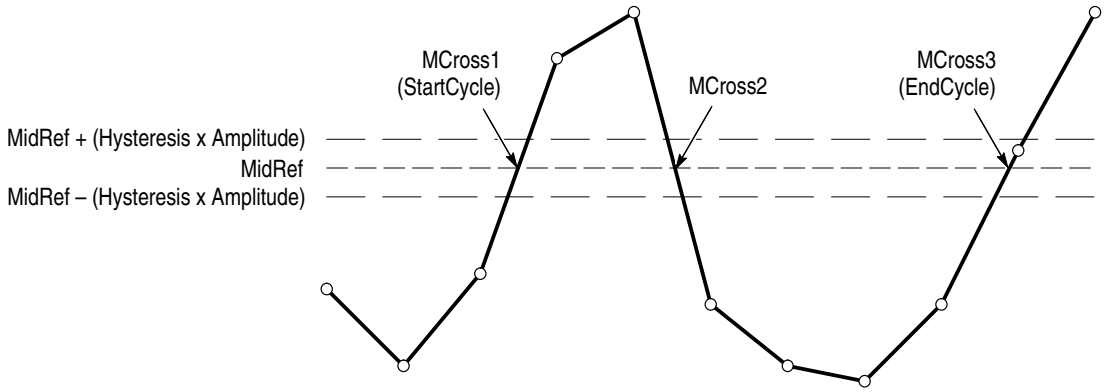


Figure B–1: MCross Calculations

Waveform[<0.0 ... RecordLength–1.0>] — holds the acquired data.

Measurement Algorithms

The automated measurements are defined and calculated as follows.

Amplitude Amplitude = High – Low

www.valuetronics.com

Area

The arithmetic area for one waveform. Remember that one waveform is not necessarily equal to one cycle. For cyclical data you may prefer to use the cycle area rather than the arithmetic area.

if Start = End then return the (interpolated) value at Start.

Otherwise,

$$Area = \int_{Start}^{End} Waveform(t)\mathrm{dt}$$

For details of the integration algorithm, see page B–13.

Cross

Timing measurement. The time relative to the trigger point at which the crossing that you specify occurs. The CROSs measurement searches for the N^{th} occurrences of an edge; during the search it counts edges of either polarity.

1. Searching from Start to End of waveform record, find the first transition — either negative-going or positive-going — through MREF (middle ref).
2. Continue the search process until the N^{th} crossing is found (user specifies N).
3. Cross = Time@crossing, where Time@trigger = 0.

   Pretrigger crossings return negative times; posttrigger crossings return positive times.

Positive values for N force the search at the start of the waveform record; Negative values and zero at the end.

Cycle Area

Amplitude (voltage) measurement. The area over one waveform cycle. For non-cyclical data, you might prefer to use the Area measurement.

If StartCycle = EndCycle then return the (interpolated) value at StartCycle.

$$CycleMean = \int_{StartCycle}^{EndCycle} Waveform(t)\mathrm{dt}$$

For details of the integration algorithm, see page B–13.

Cycle Mean

Amplitude (voltage) measurement. The mean over one waveform cycle. For non-cyclical data, you might prefer to use the Mean measurement.

If StartCycle = EndCycle then return the (interpolated) value at StartCycle.

www.valuetronics.com

$$CycleMean = \frac{\int_{StartCycle}^{EndCycle} Waveform(t)\mathrm{dt}}{(EndCycle - StartCycle) \times SampleInterval}$$

For details of the integration algorithm, see page B–13.

Cycle RMS

The true Root Mean Square voltage over one cycle.

If StartCycle = EndCycle then CycleRMS = Waveform[Start].

Otherwise,

$$CycleRMS = \sqrt{\frac{\int_{StartCycle}^{EndCycle} (Waveform(t))^2\mathrm{dt}}{(EndCycle - StartCycle) \times SampleInterval}}$$

For details of the integration algorithm, see page B–13.

Delay

Timing measurement. The amount of time between the *MidRef* crossings of two different traces or two different places on the same trace.

Delay measurements are actually a group of measurements. To get a specific delay measurement, you must specify the target and reference crossing polarities and the reference search direction.

Gain

Ratio of two amplitudes measurement. An amplitude measurement (see *Amplitude* on page B−4) is taken of the reference and target waveforms.

$$Gain = \frac{Amplitude_{target}}{Amplitude_{refernece}}$$

Fall Time

Timing measurement. The time taken for the falling edge of a pulse to drop from a HighRef value (default = 90%) to a LowRef value (default = 10%).

Figure B–2 shows a falling edge with the two crossings necessary to calculate a Fall measurement.

1. Searching from Start to End, find the first sample in the measurement zone greater than HighRef.
2. From this sample, continue the search to find the first (negative) crossing of HighRef. The time of this crossing is THF. (Use linear interpolation if necessary.)

www.valuetronics.com

3. From THF, continue the search, looking for a crossing of LowRef. Update THF if subsequent HighRef crossings are found. When a LowRef crossing is found, it becomes TLF. (Use linear interpolation if necessary.)
4. FallTime = TLF – THF

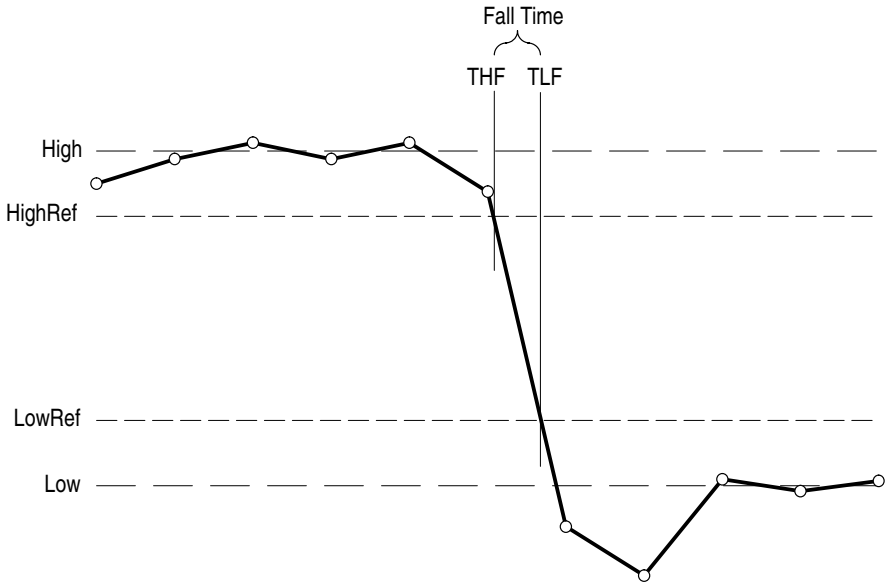


Figure B–2: Fall time

Frequency

Timing measurement. The reciprocal of the period. Measured in Hertz (Hz) where 1 Hz = 1 cycle per second.

If Period = 0 or is otherwise bad, return an error.

Frequency = 1/Period

High

100% (highest) voltage reference value. (See *High and Low* on page B–1.)

Using the min-max measurement technique:

High = Max

Low

0% (lowest) voltage reference value calculated. (See *High and Low* on page B–1.)

Using the min-max measurement technique:

Low = Min

www.valuetronics.com

Maximum

Amplitude (voltage) measurement. The maximum voltage. Typically the most positive peak voltage.

Examine all Waveform[ ] samples from Start to End inclusive, and set Max equal to the greatest magnitude Waveform[ ] value found.

Mean

The arithmetic mean for one waveform. Remember that one waveform is not necessarily equal to one cycle. For cyclical data you may prefer to use the cycle mean rather than the arithmetic mean.

$$Mean = \sum_{i=0}^{i=RecordLength-1} \frac{Waveform\ [i]}{RecordLength}$$

Minimum

Amplitude (voltage) measurement. The minimum amplitude. Typically the most negative peak voltage.

Examine all Waveform[ ] samples from Start to End inclusive, and set Min equal to the smallest magnitude Waveform[ ] value found.

Negative Duty Cycle

Timing measurement. The ratio of the negative pulse width to the signal period expressed as a percentage.

NegativeWidth is defined in **Negative Width**, below.

If Period = 0 or undefined then return an error.

$$NegativeDutyCycle = \frac{NegativeWidth}{Period} \times 100\%$$

Negative Width

Timing measurement. The distance (time) between MidRef (default = 50%) amplitude points of a negative pulse.

If MCross1Polarity = '–'

then

NegativeWidth = (MCross2 – MCross1)

else

NegativeWidth = (MCross3 – MCross2)

Ncross

Timing measurement. The time relative to the trigger point at which the negative-going crossing that you specify occurs. The NCROSs measurement searches for the N^{th} occurrence of an edge; during the search it counts only negative edges.

www.valuetronics.com

1. Searching from Start to End of waveform record, find the first negative-going transition through MREF (middle reference).
2. Continue the search process until the N^{th} negative-going crossing is found (user specifies N).
3. NCross = Time@ncrossing, whereTime@trigger = 0.

   Pretrigger crossings return negative times; posttrigger crossings return positive times.

Positive values for N force the search at the start of the waveform record; Negative values and zero, at the end.

Peak to Peak

Amplitude measurement. The absolute difference between the maximum and minimum amplitude.

PeaktoPeak = Max – Min

Pcross

Timing measurement. The time relative to the trigger point at which the positive-going crossing that you specify occurs. The PCROSs measurement searches for the N^{th} occurrence of an edge; during the search it counts only positive edges.

1. Searching from Start to End of waveform record, find the first positive-going transition through MREF (middle reference).
2. Continue the search process until the N^{th} positive-going crossing is found (user specifies N).
3. PCross = Time@pcrossing, whereTime@trigger = 0.

   Pretrigger crossings return negative times; posttrigger crossings return positive times.

Positive values for N force the search at the start of the waveform record; Negative values and zero, at the end.

Period

Timing measurement. Time taken for one complete signal cycle. The reciprocal of frequency. Measured in seconds.

Period = MCross3 – MCross1

Phase

Timing measurement. The amount of phase shift, expressed in degrees of the target waveform cycle, between the *MidRef* crossings of two different waveforms. Waveforms measured should be of the same frequency or one waveform should be a harmonic of the other.

www.valuetronics.com

Phase is a dual waveform measurement; that is, it is measured from a target waveform to a reference waveform. To get a specific phase measurement, you must specify the target and reference sources.

Phase is determined in the following manner:

1. The first *MidRefCrossing (MCross1Target)* and third *(MCross3*) in the source (target) waveform are found.
2. The period of the target waveform is calculated (see *Period* above).
3. The first *MidRefCrossing (MCross1Ref)* in the reference waveform crossing in the same direction (polarity) as that found *MCross1Target* for the target waveform is found.
4. The phase is determined by the following:

$$Phase = \frac{MCross1Ref - MCross1Target}{Period} \times 360$$

If the target waveform leads the reference waveform, phase is positive; if it lags, negative.

Positive Duty Cycle

Timing measurement. The ratio of the positive pulse width to the signal period, expressed as a percentage.

PositiveWidth is defined in **Positive Width**, following.

If Period = 0 or undefined then return an error.

$$PositiveDutyCycle = \frac{PositiveWidth}{Period} \times 100\%$$

Positive Width

Timing measurement. The distance (time) between MidRef (default = 50%) amplitude points of a positive pulse.

If MCross1Polarity = '+'

then

PositiveWidth = (MCross2 – MCross1)

else

PositiveWidth = (MCross3 – MCross2)

Rise Time

Timing measurement. Time taken for the leading edge of a pulse to rise from a LowRef value (default = 10%) to a HighRef value (default = 90%).

Figure B–3 shows a rising edge with the two crossings necessary to calculate a Rise Time measurement.

www.valuetronics.com

1. Searching from Start to End, find the first sample in the measurement zone less than LowRef.
2. From this sample, continue the search to find the first (positive) crossing of LowRef. The time of this crossing is the low rise time or TLR. (Use linear interpolation if necessary.)
3. From TLR, continue the search, looking for a crossing of HighRef. Update TLR if subsequent LowRef crossings are found. If a HighRef crossing is found, it becomes the high rise time or THR. (Use linear interpolation if necessary.)
4. RiseTime = THR – TLR

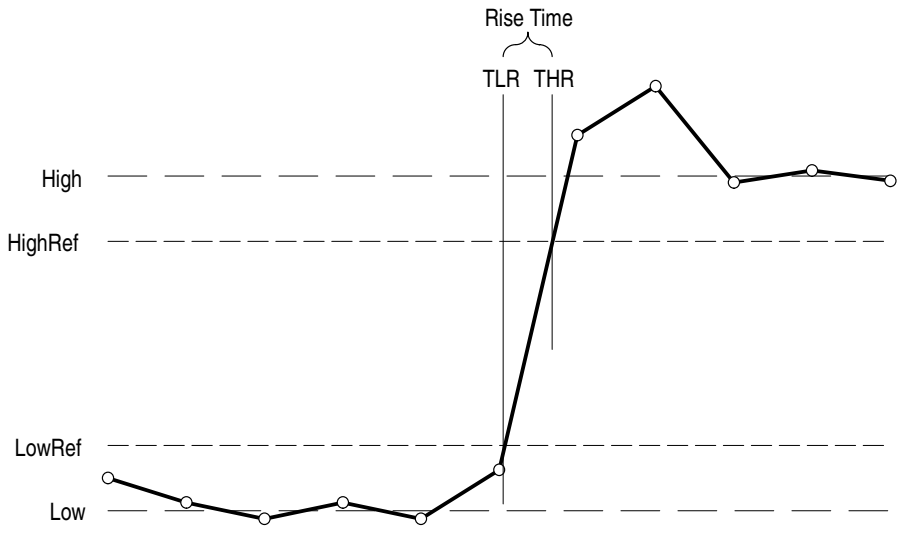


Figure B–3: Rise Time

RMS: Amplitude (voltage) measurement. The true Root Mean Square voltage.

If Start = End then RMS = the (interpolated) value at Waveform[Start].

Otherwise,

$$RMS = \sqrt{\frac{\int_{Start}^{End} (Waveform(t))^2 dt}{(End - Start) \times SampleInterval}}$$

For details of the integration algorithm, see *Integration Algorithm* in this section.

www.valuetronics.com

TTrig Timing measurement. The time difference between the main and the delay triggers.

TTrig = Time@delaytrig – Time@maintrigger

Value returned is independent of channel number. Value returned is valid only when the delay trigger source is not set to immediate.

www.valuetronics.com

Differentiation Algorithm

The differentiation algorithm used by the waveform analyzer is as follows:

$$Diff(w(n)) = 0$$
$$\quad for\ n = 0$$
$$Diff(w(n)) = [w(n+1) - w(n-1)]/(2T)$$
$$\quad for\ 1 \le n \le (R-1)$$
$$Diff(w(n)) = [w(R-1) - w(R-2)]/T$$
$$\quad for\ n = (R-1)$$

where:

n = index into the record of data points
$w(n)$ = input sampled data point
T = time interval between successive samples
R = record length

Integration Algorithm

The integration algorithm used by the waveform analyzer is as follows:

$$Intg(w(n)) = 0$$
$$\quad for\ n = 0$$
$$Intg(w(n)) = \left[1/2\ w(0) + \sum_{m=1}^{n-1} w(m) + 1/2\ w(n) \right] \times T$$
$$\quad for\ 1 \le n \le R$$

where:

n = index into record of data points
$w(n)$ = input sampled data point
T = time interval between successive samples
R = record length in points

www.valuetronics.com

Smooth Algorithm

The smoothing algorithm used by the waveform analyzer is as follows:

$$Smooth(w(n)) = (1/s)\left[\sum_{m=0}^{n+h} w(m) + (h - n) \times w(0)\right]$$

$$for\ n < h$$

$$Smooth(w(n)) = (1/s)\left[\sum_{m=n-h}^{n+h} w(m)\right]$$

$$for\ h \le n \le R-1-h$$

$$Smooth(w(n)) = (1/s)\left[\sum_{m=n-h}^{R-1} w(m) + (R-1-n) \times w(R-1)\right]$$

$$for\ n > R-1-h$$

where:

n = index into record of data points
$w(n)$ = input sampled data point
s = smoothing interval in samples; the second argument
h = half interval: (s – 1)/2 rounded down
R = record length in points

The smoothed waveform is derived by computing the average value of the corresponding point of the original waveform and a certain number of points of the original waveform on either side of the corresponding point. The number of points on either side is derived from the smoothing interval, which you set with the command CALC:SMO:POIN.

Near the ends of the waveform, nonexistent points beyond the ends of the waveform are required for averaging. The nonexistent points are assumed to be the value of the corresponding end points. This method of extending the waveform is arbitrary, so the results within a smoothing interval of the ends of the waveform must be interpreted accordingly.

www.valuetronics.com

Digital Filter Algorithms

This section describes how the digital filter of the waveform analyzer operates. The commands in the CALCulate:FILTer subsystem control the digital filter.

An Ideal Filter

The filter functions in the waveform analyzer instrument allow lowpass, highpass, bandpass and notch filters to be applied to any acquired set of data. A perfect filter would have unity transmission (with linear phase response) in the pass band, infinite attenuation in the stop band and abruptly change from pass to stop band. The transfer function for an ideal bandpass filter is depicted in Figure B–4.

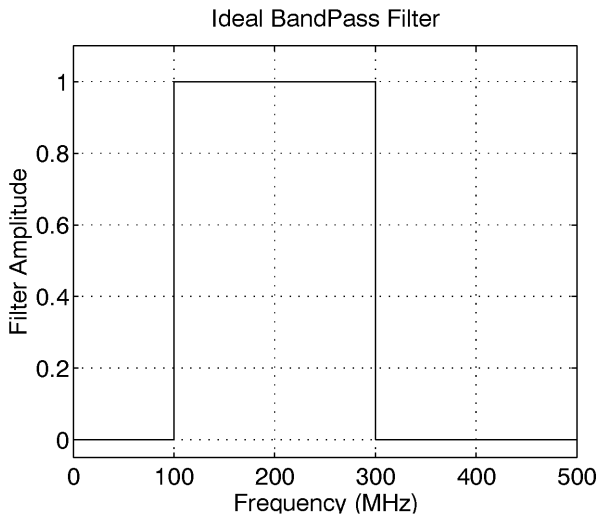


Figure B–4: Transfer function H(f) for an ideal bandpass filter

When you use a filter in the waveform analyzer, the frequency response of the desired filter is inverse Fourier transformed to calculate the response of the filter for a time domain impulse and this impulse response is convolved with the waveform data as shown in the following equation:

$$h(t) = \mathcal{F}^{-1}\{H(f)\}$$

$$output\ wfm = (input\ wfm) * h(t)$$

These equations are mathematically correct, however, it is impossible to implement them. For any ideal filter, which has abrupt changes in the transfer function, the impulse response extends for all time. Clearly an infinitely long impulse response cannot be convolved with the waveform data.

www.valuetronics.com

Rectangular Window

One possible solution for dealing with the infinitely long impulse function, h(t) is to reduce it to a manageable length. The simplest technique is to use the central points of the filter and throw away the remaining points. What this does is apply a time domain rectangular window filter to the impulse response h(t). Figure B–5 shows the transfer function for an ideal lowpass filter. Figure B–6 shows the time domain impulse for the lowpass filter and depicts one possible rectangular window which selects the central filter points.

Truncating the infinitely long impulse response of the filter to a finite length results in a filter frequency response that is no longer ideal. In the time domain the filter impulse response has been multiplied by a window w(t):

$$new\ h(t) = h(t) \cdot w(t)$$

To see how the frequency domain transfer function for the filter, H(f), has changed it is necessary to transfer the above equation to the frequency domain. Multiplication in the time domain corresponds to convolution in the frequency domain.

$$W(f) = \mathcal{T}\{w(t)\}$$

$$New\ H(f) = H(f) * W(f)$$

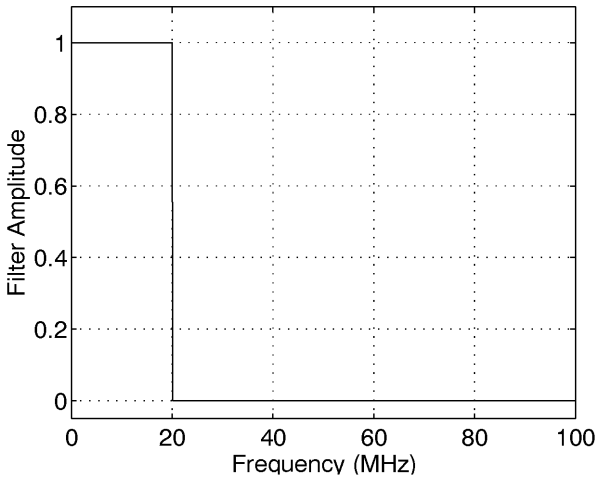


Figure B–5: Transfer function for an ideal lowpass filter

www.valuetronics.com

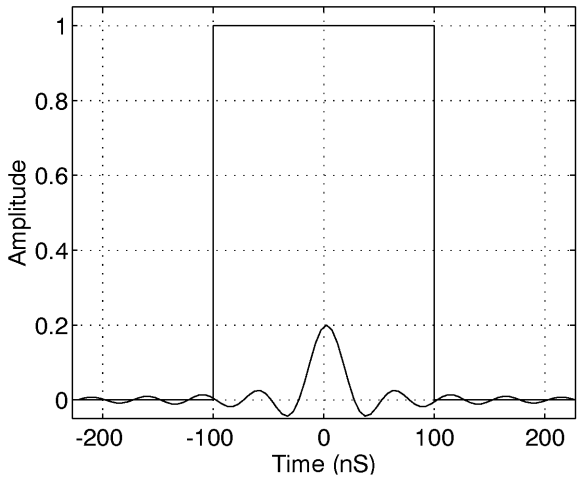


Figure B–6: Using a rectangular window to truncate the data from Figure B–5 to a finite number of points

The frequency domain convolution of H(f) with W(f) has three effects: the filter edges are no longer abrupt, the pass band transmission is no longer exactly unity, and the stop band attenuation is no longer infinite. Figures B–7 through B–9 show the original, ideal filter and the resultant filter with three different lengths of rectangular windows. Note that these plots use a dB scale for the filter.

As more points are used in the filter (corresponding to a longer window) the transition becomes sharper and sharper. However, in this example, the worst case attenuation of the filter in the stop band stays fixed at about –21 dB. Examining Figures B–7 through B–9 carefully, you can see that the peak amplitude of the ripple in the stop band (i.e., minimum attenuation) remains fixed at about –21 dB. As more points are used, the filter becomes sharper but this side lobe remains 21 dB down. For other filter shapes, such as bandpass and notch filters, the worst case stop band attenuation can be as low as –15 dB. The limitation on the stop band attenuation is the major drawback of using a rectangular window.

www.valuetronics.com

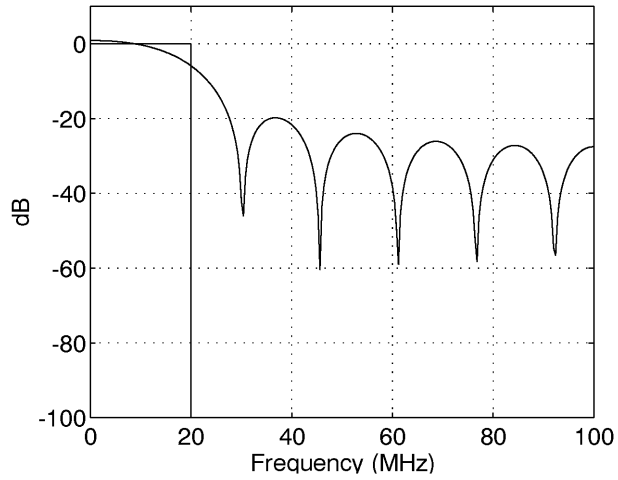


Figure B–7: Lowpass filter transfer function obtained by truncating the impulse response to just a few points

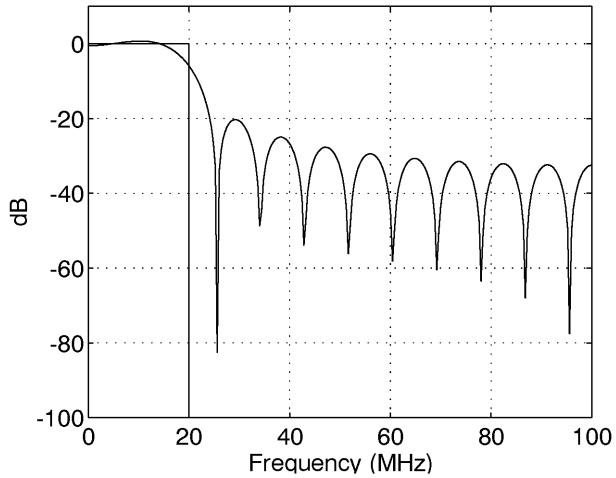


Figure B–8: Using more points in the Lowpass filter results in a steeper transition at the cutoff frequency

www.valuetronics.com

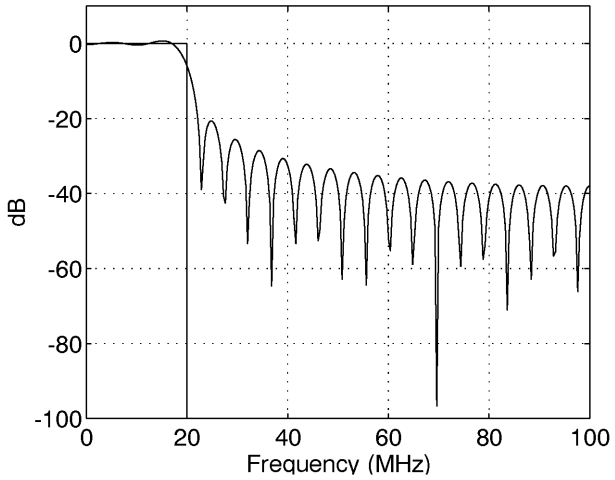


Figure B–9: Using many more points in the Lowpass filter results in a quicker transition but a minimum attenuation of 21 dB

Kaiser Window

When the filter response is truncated with a rectangular window the minimum attenuation in the stop band is at best 21 dB. In order to achieve greater attenuation in the stop band a non–rectangular window must be applied to the filter data.

There are many choices for non–rectangular windows. Common windows include Bartlett, Hamming, Hanning, and Blackman. The filter in the waveform analyzer employs a Kaiser window. This window was chosen because it offers a range of possible window shapes, and thus different stop band attenuations. For a window that is M+1 points long, the Kaiser window is defined as follows:

$$w[n] = \begin{cases} \dfrac{I_0\left[\beta\left(1-\left[\frac{(n-M/2)}{M/2}\right]^2\right)^{1/2}\right]}{I_0\left[\beta\right]} & 0 \le n \le M \\ 0 & otherwise \end{cases}$$

I_0 represents the zero order modified Bessel function of the first kind. β is a parameter that ranges from 0 to infinity. The larger the value of β, the more the window tapers at the edges. When β=0 the Kaiser window reduces to a rectangular window. Figure B–10 shows three Kaiser windows with 200 points in the window and a β of 1, 5 and 20.

www.valuetronics.com

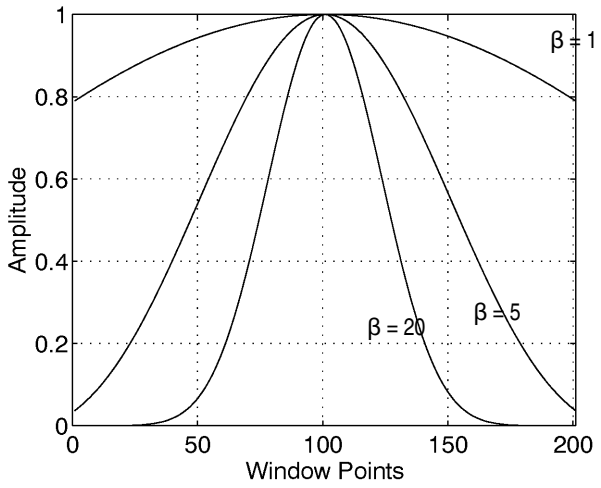


Figure B–10: Kaiser window with 200 points and β = 1, 5 and 20

For larger values of β, the Kaiser window tapers off slowly towards the edges of the window. Using the same number of data points and taking a Fourier transform of a Kaiser window and a rectangular window, the transform of the Kaiser window is broader than the rectangular window but the side lobes are much farther down.

As stated in the last section, the frequency domain transfer function of the filter is given by convolving the transfer function of the ideal filter with that of the window.

$$W(f) = \mathcal{T}\{w(t)\}$$

$$New\ H(f) = H(f) * W(f)$$

If a Kaiser window is used with the same number of points as a rectangular window, then the transition width will not be as narrow but the minimum stop band attenuation will be much greater than the 21 dB achieved with a rectangular window. To graphically see this effect, refer to Figure B–11 which shows a lowpass filter obtained with a Kaiser window with β=2.65. Compare the transfer function of this filter with that in Figure B–9 where the same number of points were used but with a rectangular window.

By setting β high enough, the stop band attenuation can be increased. The cost for this increase is a wider filter transition region which can be countered by using more points in the filter. As pointed out previously, the greater the number of points in the filter, the narrower the filter transition region. However, there is a limit to the number of points in the filter. As described more fully in the section on edge effects, the number of points in the filter is limited to a maximum of 10% of the record length.

www.valuetronics.com

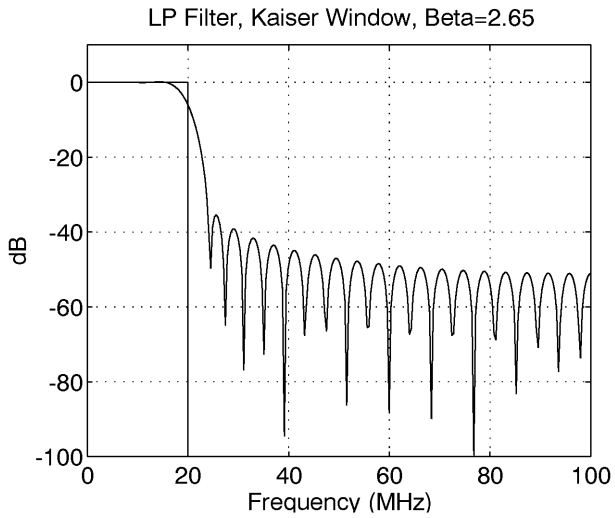


Figure B–11: Compare this result with Figure B–9 with the same number of points but a rectangular window

Defining Filter Specifications

The waveform analyzer filter is specified in a manner which may be unfamiliar to those used to working with analog filters. One difference is that cutoff and start/stop frequencies are specified as the –6 dB point, not the –3 dB point. Figure B–12 shows waveform analyzer specifications for a lowpass filter.

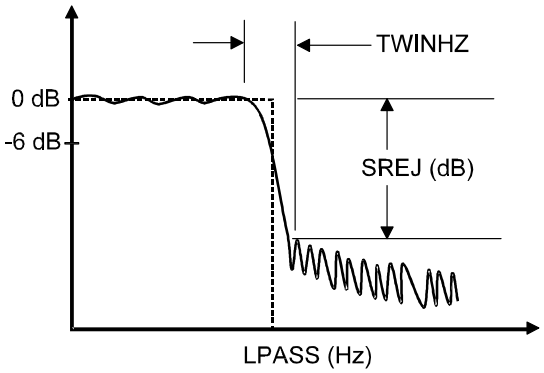


Figure B–12: Filter specifications for a lowpass filter

The cutoff frequency of the filter is specified in Hertz with the LPAS command. As an example, to set a lowpass cutoff frequency of 20 MHz, use the command:

```
CALC1:FILT:FREQ:LPAS 20E6
```

LPAS specifies a lowpass filter and 20E6 sets the cutoff frequency (at –6 dB).

www.valuetronics.com

The stop band attenuation or stop band rejection is set with the SREJ command. The SREJ is given in dB. The minimum attenuation is 15 dB and the maximum is 100 dB (the default value is 60 dB). As an example, to set the stop band attenuation to 40 dB use the command:

```
CALC1:FILT:FREQ:SREJ 40
```

The final specification of the filter is the relative filter transition width, TWID. The TWID is directly related to the TWIDHZ depicted in Figure B–12. TWIDHZ is a measure of how quickly the filter response changes from pass band to stop band. The TWID is specified relative to the Nyquist frequency for the acquisition record. The Nyquist frequency is defined as:

$$F_{NYQ} = \frac{1}{2 \cdot TINT}$$

Where TINT is the sample interval. Equivalently, the Nyquist frequency is also equal to 1/2 of the sampling rate.

The TWID is then defined as:

$$TWID = \frac{TWIDHZ}{F_{NYQ}} = TWIDHZ \cdot 2 \cdot TINT$$

Valid ranges for TWID are between 0 and 1. The smaller the value, the narrower the transition region. The default value is 0.1. As an example, to set the TWID to 0.05 use the command:

```
CALC1:FILT:FREQ:TWID 0.05
```

Define highpass filter in the same manner as the lowpass filter. The only difference is that it is necessary to specify a HPAS frequency. As an example, to set the highpass frequency to 100 MHz, use the command:

```
CALC1:FILT:FREQ:HPAS 100E6
```

Figure B–13 depicts the specifications for a band pass filter. The same specifications are used for the notch filter, but the two filters have swapped pass band and stop band regions. Like the lowpass filter, a bandpass filter has a specification for SREJ and TWID. SREJ and TWID are equal on both sides of the bandpass region. For the bandpass and notch filters it is necessary to specify a start and stop frequency. This is done with the STAR and STOP commands. As an example to set up a bandpass filter with a start frequency of 50 MHz and a stop frequency of 75 MHz use the command:

```
CALC1:FILT:FREQ:BPAS; STAR 50E6; STOP 75E6
```

Instead of specifying a STAR and STOP frequency, it is possible to specify CENT and SPAN frequencies.

www.valuetronics.com

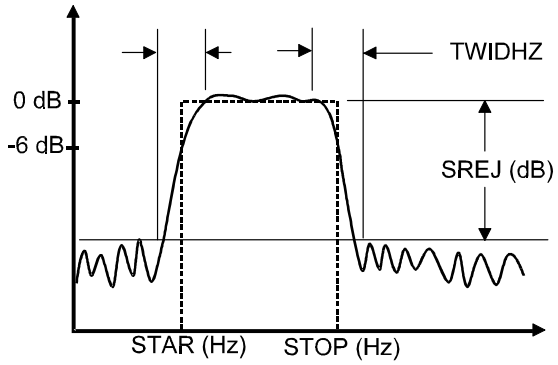


Figure B–13: Filter specifications for a bandpass filter

Filter Length Limitations and Edge Effects

After the TWID and the SREJ are specified, the β for the window is calculated from the following expressions:

$$SATT = SREJ + 6.0206$$

$$\beta = \begin{cases} 0.1102(SATT - 8.7) & SATT > 50 \\ 0.58422(SATT - 21)^{0.4} + 0.07886(SATT - 21) & 21 \le SATT \le 50 \\ 0 & SATT < 21 \end{cases}$$

The value 6.0206 is added to SREJ to ensure that the additive ripple from the notch and bandpass filters stay within specifications. Adding 6.0206 to SREJ is equivalent to dividing the ripple specification by two.

The number of points in the filter are calculated from the expression:

$$NFILT = \frac{SATT - 8}{2.285 \cdot \pi \cdot TWID} + 1$$

Actually the waveform analyzer algorithm uses the smallest odd integer greater than the NFILT calculated in the above expression (if NFILT=75.1 then 77 would be used). If there is a high level of stop band attenuation, SREJ, or if the relative filter transition width, TWID, is set very small then there may be a large number of points in the filter. For example, if SREJ =80dB and the TWID=.01 then the number of points in the filter is 1087. The input waveform itself may be less than 1087 points long and there are edge effects, which often preclude the use of such a long filter.

The output waveform is calculated by convolving the input waveform with the impulse response of the filter:

$$output\ wfm = (input\ wfm) * h(t)$$

www.valuetronics.com

If the length of the original data is N and the length of the filter is NFILT, then the result of the convolution is a record N+NFILT–1 points long. See Figure B–14. From this record, (NFILT–1)/2 points are cut off each end to return a record which is N points long, the same size as the original record. However, the filter specifications are not guaranteed throughout the length of the new data record. For (NFILT–1)/2 points on either end of the new record, the data is questionable. This is due to edge effects in the convolution when the filter record is not fully within the data record.

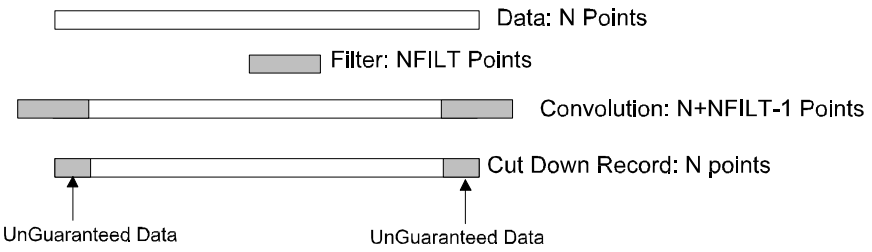


Figure B–14: Record resulting from convolving the filter impulse response with the waveform record

An example will help illustrate the edge effects. Figure B–15 shows a test signal created from a 1 V amplitude, 10 MHz sine wave and a 0.5 V, 125 MHz sine wave. This record is 500 points long, TINT=800 ps and the Nyquist frequency F_{nyq}=625 MHz. To filter out the high frequency signal, a lowpass filter was applied with LPASS = 62.5 MHz, TWID=0.1 (TWIDHZ=62.5 MHZ) and SREJ=26 dB. This resulted in a filter with β=1.51 and 53 points in the filter. Figure B–16 shows the result of applying the lowpass filter to the data. The filter did a good job of cutting out the high frequency components. To illustrate the edge effects, Figure B–17 shows a close up view of the left end of the filtered data and the 10 MHz sine signal. It is clear in that the filtered data does not initially track the source 10 MHz sine signal.

On each end of the record, (NFILT–1)/2 data points of the filtered data are not guaranteed to be within the specification of the filter. This is an undesirable situation. To limit this effect, the waveform analyzer digital filter algorithm limits the number of filter points to be a maximum of 10% of the acquisition record length. With this constraint, in the worst case condition only 5% of the data on either end of the filtered record is not guaranteed to be within the filter specification. Hence, all measurements should be on the central 90% of the data record.

To insure that your measurement does not include bad data, the waveform analyzer sets the (NFILT–1)/2 data points on either end of the filtered record to NULLs. The output waveforms of CALC blocks (which include the filter function) are always floating point numbers. NULL points are defined as

www.valuetronics.com

9.910000E+37 for ASCII format, and IEEE NAN (Not A Number) for REAL,32 format.

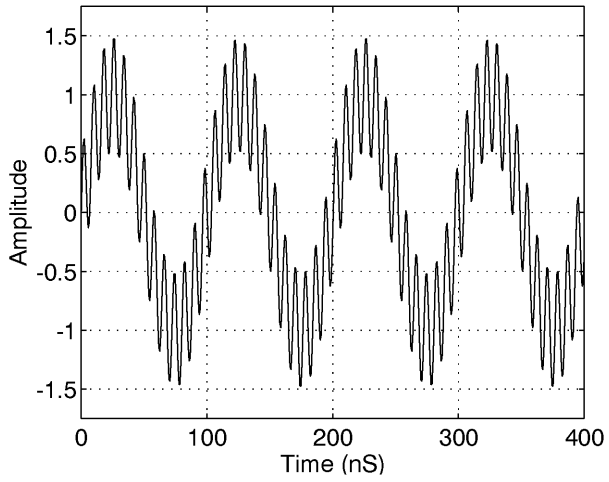


Figure B–15: Filter test signal with a 125 MHz signal modulating a 10 MHz signal

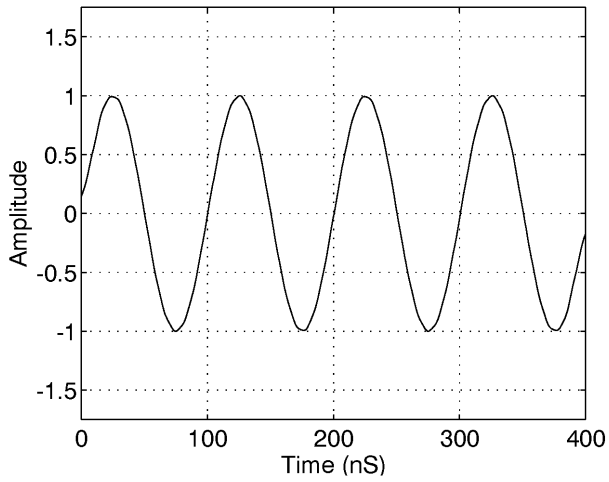


Figure B–16: Test signal after being filtered with a lowpass filter

www.valuetronics.com

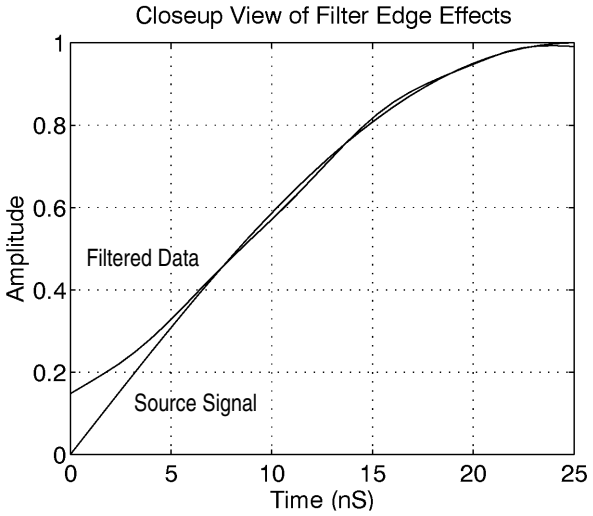


Figure B–17: View of the filtered record showing the first 5% of the filtered data

Filter Performance

Stop Band Attenuation. The stop band attenuation, or stop band rejection, is set by the SREJ parameter. The attenuation in the stop band is at least the value given by SREJ. Typically the minimum attenuation occurs at the start of the stop band. Further into the stop band the attenuation is typically several to tens of dB greater than SREJ.

Pass Band Ripple. The ripple in the pass band is not explicitly set through the filter commands. For the Kaiser window algorithm used in the waveform analyzer the pass band ripple is directly related to the stop band attenuation:

$$PassBand\ Ripple\ (dB)\ =\ 20log_{10}\left[1\ +\ 10^{\frac{-SREJ}{20}}\right]$$

For example, if the SREJ is set to 40dB then the pass band ripple will be less than 0.0864 dB; if SREJ is set to 60dB, pass band ripple will be less than 0.00868 dB.

Filter Cutoff and Roll Off. The digital filters implemented in the waveform analyzer are different than traditional analog filters. One important distinction is that the STAR, STOP, LPAS, and HPAS frequencies are not the –3 dB cutoff frequencies for the filter but are instead –6 dB cutoff frequencies.

Analog filter designers often characterize filters by the roll off rate beyond a cutoff frequency. For example, analog filters are characterized with a roll off of 20 dB/decade, 40 dB/decade, etc. Unfortunately, these familiar analog terms do not apply to digital filters. Unlike analog filters, digital filters do not continue to

www.valuetronics.com

drop off above a cutoff frequency. Instead, the filter response drops rapidly in the transition region and then flattens out somewhat in the stop band. In the transition region, the roll off cannot be well approximated as a constant roll off per decade of frequency (such as 40 dB/decade).

The Kaiser window filter technique does not provide a constant dB/decade roll off in the transition region. In fact, in the transition region, the Kaiser window technique only specifies that the transfer function will decrease from the pass band level to the stop band attenuation. What the Kaiser window does guarantee is the specifications in the pass band and stop band:

$$Passband\ Ripple \leq 20log_{10}\left[1 + 10^{\frac{-SREJ}{20}}\right] \quad f \leq LPAS - \left(\frac{TWID}{4 \cdot TINT}\right)$$

$$Stopband\ Rejection \geq SREJ \qquad f \leq LPAS + \left(\frac{TWID}{4 \cdot TINT}\right)$$

Similar specifications are achieved for highpass, bandpass and notch filters. For notch filters, be sure that (STOP– START) is greater than TWIDHZ or else no guarantee is made about the attenuation in any portion of the notch region.

Group Delay. The digital filters have linear phase in the pass band. The group delay, which is the derivative of the phase, is therefore constant in the pass band.

Practically speaking, this means that if you have a signal which is made up of many frequency components, the relative phase of these frequency components are preserved in the filter.

Error Conditions

There are two main causes of errors from the digital filter code. One of the sources of error is a filter specification that generates too many filter coefficients. The other class of errors is from cutoff frequencies that violate certain constraints.

Too Many Filter Coefficients. If the stop band attenuation SREJ and/or the relative filter transition width TWID is set to too high, then the number of points required by the filter may exceed 10% of the acquired record length. Since the digital filter implementation limits the number of coefficients to 10% of the record length, waveform analyzer reports an error, and performs no filtering.

Suppose, for example, you acquire a record with 1024 points at 1 GSample/second acquisition rate. You set the lowpass filter to a cutoff frequency of 200 MHz, a stop band attenuation, SREJ, of 80 dB and relative filter transition width, TWID, of 0.05. Such a filter requires 219 points, which is more than 102 points (10% of the data record), and the following error is reported:

```
2100,"CalculateN questionable; digital filter error - filter
specs require too many coefficients"
```


www.valuetronics.com

When this error occurs, there are a number of things that can be done to allow the filter generation to succeed. One change you could do is to change the value for TWID (and equivalently TWIDHZ) and keep the remaining filter parameters fixed. This value of TWID is calculated from:

$$TWID = \frac{SREJ - 1.9794}{7.1785 \cdot (MaxNPTS)}$$

Where MaxNPTS in this case is set to 102. The maximum possible value for TWID is 1 (for realistic filters it is desirable to have TWID less than 0.2). If the specified SREJ is too large then the value of TWID calculated in the above expression may exceed 1. In this case, it is not possible to design a filter that meets the SREJ specifications and uses less than the maximum possible number of points.

Another change you could do is to change the value of SREJ and keep the remaining filter parameters fixed. This value of SREJ is calculated from:

$$SREJ = 7.1785 \cdot TWID \cdot (MaxNPTS) + 1.9794$$

The minimum value for SREJ is 15 dB. If the TWID was initially chosen too small then the above formula may predict a value for SREJ which is less than 15 dB. In this case, it is impossible to create a filter which meets the specification for TWID and which uses less than the maximum possible number of points.

Another change would be to acquire the data using a longer record length. Since the filter can have more points if the input data record length is larger, you can use a tighter specified filter with a longer record length. In our example, going to a record length of 4096 would have allowed the filter to operate without error.

Incorrect Cutoff Frequencies. The basic problem is that you can't have the cutoff frequency for low or highpass filters close to 0 or Nyquist. For bandpass and notch filters, there is the additional constraint that start and stop frequencies can't be too close to each other. The rules are as follows:

- Insure that the cutoff frequency (LPAS or HPAS) minus half the transition width is greater than 0.
- Insure that the cutoff frequency (LPAS or HPAS) plus half the transition width is less than Nyquist.

For bandpass/notch filters, insure that START minus half the transition width is greater than 0, STOP plus half the transition width is less than Nyquist, and START plus half the transition width is less than STOP minus half the transition width.

If these constraints are violated, the following error messages appear:

www.valuetronics.com

```
2100,"CalculateN questionable; digital filter error - lowpass
filter cutoff invalid", or

2100,"CalculateN questionable; digital filter error -
highpass filter cutoff invalid", or

2100,"CalculateN questionable; digital filter error -
bandpass/notch filter start/stop invalid"
```

Note that for both major sources of error, errors are detected and reported when the calculate block is executed, not when filter parameters are defined or when the filter expression is defined. The errors are delayed because the sample rate (which defines Nyquist) and record length (which sets the maximum number of filter coefficients) may change after you define the filter parameters.

When these errors occur, the waveform analyzer CALC block copies its input waveform to its output and performs *no* filtering.

General Guidelines

Edge effects from the filter can set up to 5% of the data on either end of the filtered record to NULL values: ASCII, 9.910000E+37 or binary, IEEE NAN. If you are going to filter the data, make sure all acquisitions have a sufficient number of points on either side of the data of interest.

If you apply a filter to a set of data and get an error, you can either specify a filter with less stringent specifications or you acquire the data a second time using a longer record length. The maximum length of the filter is 10% of the data record, so if you use a longer record you have more points available in the filter and you can use a higher specified filter.

To increase the number of points in your data, use the same sample rate but a longer record length. See the commands SWEep:POINts and SWEep:TIME to set the record length. Make sure you acquire additional points before and after the data of interest. Do not try to obtain more points in your data record by simply increasing the sampling rate. Increasing the sampling rate provides more points but it also increases the Nyquist sampling rate which can make the desired filter more difficult to achieve.

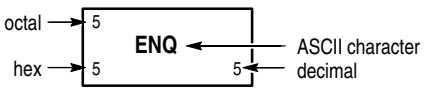
www.valuetronics.com

Appendix C: ASCII Character Chart

Table C–1: ASCII Code Chart

B7 B6 B5 BITS B4 B3 B2 B1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		NUMBERS SYMBOLS		UPPER CASE		LOWER CASE	
0 0 0 0	0 NUL 0 0	20 DLE 10 16	40 SP 20 32	60 0 30 48	100 @ 40 64	120 P 50 80	140 ` 60 96	160 p 70 112
0 0 0 1	1 SOH 1 1	21 DC1 11 17	41 ! 21 33	61 1 31 49	101 A 41 65	121 Q 51 81	141 a 61 97	161 q 71 113
0 0 1 0	2 STX 2 2	22 DC2 12 18	42 " 22 34	62 2 32 50	102 B 42 66	122 R 52 82	142 b 62 98	162 r 72 114
0 0 1 1	3 ETX 3 3	23 DC3 13 19	43 # 23 35	63 3 33 51	103 C 43 67	123 S 53 83	143 c 63 99	163 s 73 115
0 1 0 0	4 EOT 4 4	24 DC4 14 20	44 $ 24 36	64 4 34 52	104 D 44 68	124 T 54 84	144 d 64 100	164 t 74 116
0 1 0 1	5 ENQ 5 5	25 NAK 15 21	45 % 25 37	65 5 35 53	105 E 45 69	125 U 55 85	145 e 65 101	165 u 75 117
0 1 1 0	6 ACK 6 6	26 SYN 16 22	46 & 26 38	66 6 36 54	106 F 46 70	126 V 56 86	146 f 66 102	166 v 76 118
0 1 1 1	7 BEL 7 7	27 ETB 17 23	47 ' 27 39	67 7 37 55	107 G 47 71	127 W 57 87	147 g 67 103	167 w 77 119
1 0 0 0	10 BS 8 8	30 CAN 18 24	50 (28 40	70 8 38 56	110 H 48 72	130 X 58 88	150 h 68 104	170 x 78 120
1 0 0 1	11 HT 9 9	31 EM 19 25	51) 29 41	71 9 39 57	111 I 49 73	131 Y 59 89	151 i 69 105	171 y 79 121
1 0 1 0	12 LF A 10	32 SUB 1A 26	52 * 2A 42	72 : 3A 58	112 J 4A 74	132 Z 5A 90	152 j 6A 106	172 z 7A 122
1 0 1 1	13 VT B 11	33 ESC 1B 27	53 + 2B 43	73 ; 3B 59	113 K 4B 75	133 [5B 91	153 k 6B 107	173 { 7B 123
1 1 0 0	14 FF C 12	34 FS 1C 28	54 , 2C 44	74 < 3C 60	114 L 4C 76	134 \ 5C 92	154 l 6C 108	174 ¦ 7C 124
1 1 0 1	15 CR D 13	35 GS 1D 29	55 – 2D 45	75 = 3D 61	115 M 4D 77	135] 5D 93	155 m 6D 109	175 } 7D 125
1 1 1 0	16 SO E 14	36 RS 1E 30	56 . 2E 46	76 > 3E 62	116 N 4E 78	136 ^ 5E 94	156 n 6E 110	176 ~ 7E 126
1 1 1 1	17 SI F 15	37 US 1F 31	57 / 2F 47	77 ? 3F 63	117 O 4F 79	137 – 5F 95	157 o 6F 111	177 RUBOUT (DEL) 7F 127

KEY



Tektronix

REF: ANSI STD X3.4-1977
IEEE STD 488.1-1987
ISO STD 646-2973

www.valuetronics.com

Appendix D: SCPI Conformance Information

All commands in the TVS600 Series of waveform analyzers conform to SCPI Version 1995.0. Table D–1 lists all commands supported by the waveform analyzers. The columns at right show whether a command is defined in the SCPI 1995.0 Standard or not.

Table D–1: SCPI Conformance Information

Command				Defined in SCPI 1995.0	Not Defined In SCPI 1995.0
AADVance	[:STATe][?]				✓
	:COUNt[?]				✓
	:RECord	:COUNt[?]			✓
		:STARt[?]			✓
ABORt				✓	
ARM	[:A][:LAYer[1]]	:DEFine?		✓	
		:SOURce[?]		✓	
AVERage	[:STATe][?]			✓	
	:COUNt[?]			✓	
	:TYPE[?]			✓	
CALCulate[n]	:AAMList[?]				✓
		:STATe[?]			✓
	:DATA?			✓	
		:PREamble?		✓	
	:DERivative	:STATe[?]		✓	
	:FEED[?]			✓	
	:FILTer	:FREQuency	[:TYPE][?]	✓	
			:CENTer[?]	✓	
			:HPASs[?]		✓
			:LPASs[?]		✓
			:SPAN[?]	✓	
			:SREJection[?]		✓
			:STARt[?]	✓	
			:STATe[?]	✓	
			:STOP[?]	✓	

www.valuetronics.com

Table D–1: SCPI Conformance Information (Cont.)

Command				Defined in SCPI 1995.0	Not Defined In SCPI 1995.0
			:TWIDth[?]		✓
	:FORMat[?]			✓	
	:IMMediate[?]			✓	
	:INTegral	:STATe[?]		✓	
	:PATH[?]			✓	
		:EXPRession[?]		✓	
	:SMOothing	[:STATe][?]			✓
		:POINts[?]		✓	
	:TRANsform	:FREQuency	:STATe[?]	✓	
			:WINDow[?]	✓	
	:WMList[?]				✓
		:STATe[?]			✓
	:WMParameter	:HIGH[?]			✓
		:HMEThod[?]			✓
		:LOW[?]			✓
		:LMEThod[?]			✓
		:HREFerence	[:ABSolute][?]		✓
			:RELative[?]		✓
		:LREFerence	[:ABSolute][?]		✓
			:RELative[?]		✓
		:MREFerence	[:ABSolute][?]		✓
			:HYSTeresis[?]		✓
			:RELative[?]		✓
		:RMEThod[?]			✓
CALibration	:ALL[?]			✓	
	:RESults	[:CODE]?			✓
		:VERBose?			✓
DATA?				✓	
	:PREamble?			✓	
FORMat	[:DATA][?]			✓	
		:CALCulate[n][?]			✓
		:TRACe	AATS[?]		✓

www.valuetronics.com

Table D–1: SCPI Conformance Information (Cont.)

Command				Defined in SCPI 1995.0	Not Defined In SCPI 1995.0
	:BORDer[?]			✓	
FUNCtion	[:ON][?]			✓	
		:ALL		✓	
		:COUNt?		✓	
	:OFF[?]			✓	
		:ALL		✓	
		:COUNt?		✓	
	:CONCurrent[?]			✓	
	:STATe?			✓	
	:STATe				✓
INITiate	[:IMMediate]			✓	
	:CONTinuous[?]			✓	
	:COUNt[?]				✓
INPut[n]	:COUPling[?]			✓	
	:FILTer	[:LPASs]	[:STATe][?]	✓	
			:FREQuency[?]	✓	
	:IMPedance[?]			✓	
	:PROTection	:STATe[?]			✓
MEMory	:COPY	[:NAME]		✓	
	:DATA[?]			✓	
	:NSTates?			✓	
	:STATe	:CATalog?		✓	
		:DEFine?		✓	
OUTPut	:ECLTrg[n]	[:STATE][?]		✓	
		:SOURce?		✓	
	:PCOMpensate	[:STATE][?]			✓
	:REFerence	[:STATE][?]			✓
		:FUNCtion[?]			✓
	:TTLTrg[n]	[:STATE][?]		✓	
		:POLarity[?]		✓	
		:SOURce?		✓	
ROSCillator	:SOURce[?]			✓	

www.valuetronics.com

Table D–1: SCPI Conformance Information (Cont.)

Command					Defined in SCPI 1995.0	Not Defined In SCPI 1995.0
STATus	:OPERation?				✓	
		:CONDition?			✓	
		:ENABle[?]			✓	
		:NTRansition[?]			✓	
		:PTRansition[?]			✓	
		:QENable	:NTRansition[?]			✓
			:PTRansition[?]			✓
	:PRESet				✓	
	:QUEStionable	[EVENt]?			✓	
		:CONDition?			✓	
		:ENABle[?]			✓	
		:NTRansition[?]			✓	
		:PTRansition[?]			✓	
		:QENable	:NTRansition[?]			✓
			:PTRansition[?]			✓
	:SESR	:QENable				✓
SWEep	:OFFSet	:POINts[?]			✓	
		:TIME[?]			✓	
	:OREFerence	:LOCation[?]			✓	
	:POINts[?]				✓	
	:TIME[?]				✓	
	:TINTerval[?]				✓	
SYSTem	:COMMunicate	:SERial	:BAUD?		✓	
			:CONTrol	:DCD[?]		✓
				:RTS[?]	✓	
			:ECHO[?]			✓
			:ERESponse[?]			✓
			:LBUFfer[?]			✓
			:PACE[?]		✓	
			:PARity[?]		✓	
			:PRESet	[:ALL]		✓
				:RAW		✓

www.valuetronics.com

Table D–1: SCPI Conformance Information (Cont.)

Command					Defined in SCPI 1995.0	Not Defined In SCPI 1995.0
				:TERMinal		✓
			:SBITs[?]		✓	
	:ERRor?				✓	
		:ALL?				✓
		:CODE[?]				✓
			:ALL?			✓
		:COUNt?				✓
	:PROTect[?]					✓
	:SECurity	:IMMediate			✓	
	:SET[?]				✓	
	:VERSion?				✓	
TEST	[:ALL][?]					✓
	:RESults	[:CODE]?				✓
		:VERBose?				✓
TRACe	[:DATA]?				✓	
		:PREamble?			✓	
	:CATalog?				✓	
	:COPY				✓	
	:FEED?				✓	
	:LIST[?]					✓
	:POINts?				✓	
TRIGger	[:A]	:ATRigger	[:STATe][?]		✓	
		:COUPling[?]			✓	
		:COUPling	:<preset>			✓
		:DEFine?			✓	
		:DELay[?]			✓	
		:FILTer	[:LPASs]	[:STATe][?]	✓	
			:HPASs	[:STATe][?]	✓	
		:HOLDoff	:TIME[?]			✓
		:HYSTeresis	:SELect[?]			✓
		:PULSe				✓
			:CLASs[?]			✓

www.valuetronics.com

Table D–1: SCPI Conformance Information (Cont.)

Command					Defined in SCPI 1995.0	Not Defined In SCPI 1995.0
			:GLITch	:POLarity[?]		✓
				:QUALify[?]		✓
				:WIDTh[?]		✓
			:SOURce[?]			✓
			:THReshold[?]			✓
			:WIDTh	:LLIMit[?]		✓
				:POLarity[?]		✓
				:QUALify[?]		✓
				:ULIMit[?]		✓
		:LEVel[?]			✓	
		:SLOPe[?]			✓	
		:SOURce[?]			✓	
		:TYPE[?]			✓	
	:B	:COUPling[?]			✓	
		:COUPling	:<preset>			✓
		:DEFine?			✓	
		:DELay[?]			✓	
		:ECOunt[?]			✓	
		:FILTer	[:LPASs]	[:STATe][?]	✓	
			:HPASs	[:STATe][?]	✓	
		:HYSTeresis	:SELect[?]			✓
		:LEVel[?]			✓	
		:SLOPe[?]			✓	
		:SOURce[?]			✓	
VOLTage[n]	[:DC]	:RANGe	[:UPPer][?]		✓	
			:LOWer[?]		✓	
			:OFFSet[?]		✓	
			:PTPeak[?]		✓	
IEEE 488.2 Common Commands						
*CAL?					✓	
*CLS					✓	
*ESE[?]					✓	

www.valuetronics.com

Table D–1: SCPI Conformance Information (Cont.)

Command	Defined in SCPI 1995.0	Not Defined In SCPI 1995.0
*ESR?	✓	
*IDN?	✓	
*LRN?	✓	
*OPC[?]	✓	
*OPT?	✓	
*PUD[?]	✓	
*RCL	✓	
*RST	✓	
*SAV	✓	
*SRE[?]	✓	
*STB?	✓	
*TST?	✓	
*WAI?	✓	

www.valuetronics.com

Glossary

AC coupling
A type of signal transmission that blocks the DC component of a signal but uses the dynamic (AC) component. Useful for observing an AC signal that is normally riding on a DC signal.

Accuracy
The closeness of the indicated value to the true value.

Acquisition
The process of sampling signals from input channels, digitizing the samples into data points, and assembling the data points into a waveform record. The waveform record is stored in memory. The trigger marks time zero in that process.

Acquisition interval
The time duration of the waveform record divided by the record length. The TVS600 Waveform Analyzer stores one data point for every acquisition interval.

Acquisition record
An array of definite length containing digital values that represent an analog input signal. The digital values are derived by measuring the voltage level of the input signal at precisely timed intervals (the sample interval).

ADC
Analog to digital converter. A circuit device that converts an analog signal, such as a temperature sensor, into a digital value.

Aliasing
A false representation of a signal due to insufficient sampling of high frequencies or fast transitions. A condition that occurs when a waveform analyzer digitizes at an effective sampling rate that is too slow to reproduce the input signal. The waveform stored by the waveform analyzer may have a lower frequency than the actual input signal.

Amplitude
The High waveform value less the Low waveform value.

Area
Measurement of the waveform area taken over the entire waveform or the gated region. Expressed in volt-seconds. Area above ground is positive; area below ground is negative.

ASCII
Acronym for the American Standard Code for Information Interchange.

www.valuetronics.com

Controllers transmit commands to the waveform analyzer using ASCII character encoding.

Attenuation
The degree the amplitude of a signal is reduced when it passes through an attenuating device such as a probe or attenuator. That is, the ratio of the input measure to the output measure. For example, a 10X probe will attenuate, or reduce, the input voltage of a signal by a factor of 10.

Automatic trigger mode
A trigger mode that causes the waveform analyzer to automatically acquire if triggerable events are not detected within a specified time period.

Average acquisition mode
In this mode, the waveform analyzer acquires a waveform that is the averaged result of several acquisitions. Averaging reduces the apparent noise. The waveform analyzer acquires data as in the sample mode and then averages it according to a specified number of averages.

Bandwidth
The highest frequency signal the waveform analyzer can acquire with no more than 3 dB (× .707) attenuation of the original (reference) signal.

Burst width
A timing measurement of the duration of a burst.

Channel
One type of input used for signal acquisition. The waveform analyzer has four channels.

Class 2
A term for ECL-level signals on the VXI Local Bus.

Command
Specifies an action for the instrument to perform.

Common Commands
See IEEE 488.2 Common Commands.

Controller
A computer or other device that sends commands to and accepts responses from the waveform analyzer.

Coupling
The association of two or more circuits or systems in such a way that power or information can be transferred from one to the other. You can couple the input signal to the trigger and vertical systems several different ways.

Cycle area
A measurement of waveform area taken over one cycle. Expressed in volt-seconds. Area above ground is positive; area below ground is negative.

www.valuetronics.com

Cycle mean
An amplitude (voltage) measurement of the arithmetic mean over one cycle.

Cycle RMS
The true Root Mean Square voltage over one cycle.

DC coupling
A mode that passes both AC and DC signal components to the circuit. Available for both the trigger system and the vertical system.

Delay time
The time between the trigger event and the acquisition of data.

Digitizing
The process of converting a continuous analog signal such as a waveform to a set of discrete numbers representing the amplitude of the signal at specific points in time. Digitizing is composed of two steps: sampling and quantizing.

DSP (Digital Signal Processor)
The DSP coordinates operation of the acquisition system and performs waveform transforms, calculations, and measurements,

Edge Trigger
Triggering occurs when the waveform analyzer detects the source passing through a specified voltage level in a specified direction (the trigger slope).

Envelope acquisition mode
A mode in which the waveform analyzer acquires a waveform that shows the variation extremes of several acquisitions.

Chained Commands and Queries
Commands and queries grouped together into a single message to be sent to the instrument. Each command and query in the chained message must be separated by a semicolon (;).

EOM
A generic acronym referring to the end-of-message terminator. The end-of-message terminator can be either an EOI or the ASCII code for line feed (LF).

ERT sampling
Extended real time sampling. See Extended Real-time sampling for the definition.

Event Status Enable Register
Controls which events are summarized in the event status bit (bit 5) of the Status Byte Register.

Extended Real-time sampling (ERT)
A sampling mode where the waveform analyzer samples fast enough to completely fill a waveform record from a single trigger event and continues

www.valuetronics.com

until Acquisition memory is full. Use ERT sampling to acquire very long acquisition records for a single channel.

Fall time
A measurement of the time it takes for trailing edge of a pulse to fall from a HighRef value (typically 90%) to a LowRef value (typically 10%) of its amplitude.

Frequency
A timing measurement that is the reciprocal of the period. Measured in Hertz (Hz) where 1 Hz = 1 cycle per second.

Ground (GND) coupling
Coupling option that disconnects the input signal from the vertical system.

GHZ
Gigahertz, 10^9 cycles per second

GPIB
Acronym for General Purpose Interface Bus, the common name for the communications interface system defined in IEEE Std 488.

High
The value used as 100% in automated measurements (whenever high ref, mid ref, and low ref values are needed as in fall time and rise time measurements). May be calculated using either the min/max or the histogram method. With the min/max method (most useful for general waveforms), it is the maximum value found. With the histogram method (most useful for pulses), it refers to the most common value found above the mid point. See *Appendix B: Algorithms* for details.

Holdoff, trigger
A specified amount of time after a trigger signal that elapses before the trigger circuit will accept another trigger signal. Trigger holdoff helps ensure a stable acquisition.

HZ
Hertz, cycles per second

IEEE
Acronym for the Institute for Electrical and Electronic Engineers.

IEEE 488.21 Common Commands
The set of commands and queries defined by the ANSI/IEEE Standard 488.2.

KHZ
Kilohertz, 10^3 cycles per second

Logical address
A specific, unique address setting for modules in a VXIbus system.

www.valuetronics.com

Low
The value used as 0% in automated measurements (whenever high ref, mid ref, and low ref values are needed as in fall time and rise time measurements). May be calculated using either the min/max or the histogram method. With the min/max method (most useful for general waveforms), it is the minimum value found. With the histogram method (most useful for pulses), it refers to the most common value found below the mid point. See *Appendix B: Algorithms* for details.

Maximum
Amplitude (voltage) measurement of the maximum amplitude. Typically the most positive peak voltage.

Mean
Amplitude (voltage) measurement of the arithmetic mean over the entire waveform.

Mb/s
Megabits per second.

MHZ
Megahertz, 10^6 cycles per second

Minimum
Amplitude (voltage) measurement of the minimum amplitude. Typically the most negative peak voltage.

Negative duty cycle
A timing measurement representing the ratio of the negative pulse width to the signal period, expressed as a percentage.

Negative overshoot measurement
Amplitude (voltage) measurement.

$$NegativeOvershoot = \frac{Low - Min}{Amplitude} \times 100\%$$

Negative width
A timing measurement of the distance (time) between two amplitude points — falling-edge *MidRef* (default 50%) and rising-edge *MidRef* (default 50%) — on a negative pulse.

Normal trigger mode
A mode on which the waveform analyzer does not acquire a waveform record unless a valid trigger event occurs. It waits for a valid trigger event before acquiring waveform data.

Output Queue
Stores query responses from the instrument.

www.valuetronics.com

Peak-to-Peak
Amplitude (voltage) measurement of the absolute difference between the maximum and minimum amplitude.

Period
A timing measurement of the time covered by one complete signal cycle. It is the reciprocal of frequency and is measured in seconds.

Positive duty cycle
A timing measurement of the ratio of the positive pulse width to the signal period, expressed as a percentage.

Positive overshoot
Amplitude (voltage) measurement.

$$PositiveOvershoot = \frac{Max - High}{Amplitude} \times 100\%$$

Positive width
A timing measurement of the distance (time) between two amplitude points — rising-edge *MidRef* (default 50%) and falling-edge *MidRef* (default 50%) — on a positive pulse.

Posttrigger
The specified portion of the waveform record that contains data acquired after the trigger event.

Pretrigger
The specified portion of the waveform record that contains data acquired before the trigger event.

Probe
A waveform analyzer input device.

Probe compensation
Adjustment that improves low-frequency response of a probe.

Pulse trigger
A trigger mode in which triggering occurs if the waveform analyzer finds a pulse, of the specified polarity, with a width between, or optionally outside, the user-specified lower and upper time limits.

Quantizing
The process of converting an analog input that has been sampled, such as a voltage, to a digital value.

Query
A message sent to the instrument that returns information about the state of the instrument.

www.valuetronics.com

Real-time sampling
A sampling mode where the waveform analyzer samples fast enough to completely fill a waveform record from a single trigger event. Use real-time sampling to capture single-shot or transient events.

Record length
The specified number of samples in a waveform.

Reference memory
Memory in a waveform analyzer used to store settings. The waveform analyzer saves the data even when the waveform analyzer is turned off or unplugged.

Rise time
The time it takes for a leading edge of a pulse to rise from a *LowRef* value (typically 10%) to a *HighRef* value (typically 90%) of its amplitude.

RMS
Amplitude (voltage) measurement of the true Root Mean Square voltage.

RT Acquisition
A sampling mode where the waveform analyzer samples fast enough to completely fill a waveform record from a single trigger event. Use real-time sampling to capture single-shot or transient events.

Sample acquisition mode
The waveform analyzer creates a record point by saving the first sample during each acquisition interval. That is the default mode of the acquisition.

Sample interval
The time interval between successive samples in a time base. For real-time digitizers, the sample interval is the reciprocal of the sample rate.

Sampling
The process of capturing an analog input, such as a voltage, at a discrete point in time and holding it constant so that it can be quantized.

SCPI
An acronym for Standard Commands for Programmable Instruments. A standard that provides guidelines for remote programming of instruments.

:Sequence[1]
The first trigger system in the SCPI standard model for triggering. TRIGger:A provides Sequence1 functions.

Slope
The direction at a point on a waveform. You can calculate the direction by computing the sign of the ratio of change in the vertical quantity (Y) to the change in the horizontal quantity. The two values are rising and falling.

www.valuetronics.com

Slot 0
The location in a VXIbus mainframe for a controller or resource manager module.

Stale data
Requesting data with a command such as TRACe when that channel was not previously acquired results in Error 230: Data corrupt or stale.

Standard Event Status Register
Records many types of events that occur in the instrument such as execution error and operation complete.

Status Byte Register
Summarizes information from other registers in the Status and Event Reporting System.

Subsystem Hierarchy Tree
A graphical representation of a subsystem of commands and queries.

System Error and Event Queue
Stores error and event messages.

TEKSecure
A Tektronix custom command that initializes settings and waveform memories. This command overwrites any previously stored data.

TTLTRG*
The TTL-level trigger bus on the VXIbus backplane.

Tek Secure
This feature erases all waveform and setup memory locations (setup memories are replaced with the factory setup). Then it checks each location to verify erasure. This feature finds use where this TVS600 Waveform Analyzer is used to gather security sensitive data, such as is done for research or development projects.

Time base
The set of parameters that let you define the time and horizontal axis attributes of a waveform record. The time base determines when and how long to acquire record points.

Trigger
An event that marks time zero in the waveform record. It results in the acquisition of a waveform record.

Trigger level
The vertical level the trigger signal must cross to generate a trigger (on edge trigger mode).

www.valuetronics.com

VXIbus
A standardized backplane and system specification for modular instrumentation.

VXI Local Bus
Lines in the VXIbus backplane for direct communication between adjacent modules.

Waveform
The shape or form (visible representation) of a signal.

Waveform interval
The time interval between record points.

www.valuetronics.com

Index

A

AADVance, 3–30
AADVance Subsystem, 3–29–3–34
AADVance:COUNt, 3–31
AADVance:RECord:COUNt, 3–32
AADVance:RECord:STARt, 3–33
AAMList
  auto-advance measurement list, 3–44
  use in calculation expressions, 2–34
Abbreviating Commands, Queries, and Parameters, 3–3
ABORt, 3–121
ABORt Subsystem, 3–117
About the System Setup, 1–14
AC
  INPut coupling, 3–124
  input coupling, 2–4
  Measurement, 2–36
  measurement, 3–80
  trigger coupling discussion, 2–23
  TRIGger:A coupling, 3–212
  TRIGger:B coupling, 3–225
AC coupling, Glossary–1
AC line voltage, trigger input, 2–22
ACCESSED indicator, 2–4
Accessories, 1–1–1–2
  Optional, 1–1–1–2
  Standard, 1–1–1–2
Accuracy, *Glossary–1*
ACNReject, trigger coupling discussion, 2–23
Acquiring waveforms, starting acquisition, 3–117
Acquisition, 2–9–2–18, Glossary–1
  Abort command, 3–121
  acquisition cycle described, 2–14–2–16
  acquisition parameters, 2–10
  averaging during, 2–18
  enveloping during, 2–18
  extended real-time mode, 2–13
  Initiate command, 3–117
  Interval, Glossary–1
  looping during, 2–14
  looping with INIT:CONT, 3–118
  modes of acquisition, 2–12–2–14
  postrigger points, 2–11
  pretrigger points, 2–11
  principles of digitizing waveforms, 2–9
  real time mode, 2–13
  record length, 2–10
  record position, 2–11
  refresh cycle, 2–14
  start acquisition, 3–117
  trigger point, 2–11
  waveform record position, 2–11
Acquisition Clock, setting source, 3–150
Acquisition record, Definition of, *Glossary–1*
ADC, Definition of, *Glossary–1*
Address, setting the logical address, 1–2
Algebraic syntax, for use defining calculations, 2–33
Algorithms, B–1–B–30
  waveform function, *B–13*
Aliasing, Glossary–1
  on FFT waveforms, 2–41
AMPlitude, measurement, 3–78
Amplitude, 2–35, Glossary–1
Applications, derivative waveforms, 2–39
AREA, 2–35
  measurement, 3–78
Area, Glossary–1
ARM INPUT connector, 2–5
ARM Subsystem, 3–35–3–38
ARM:DEFine?, 3–35
ARM:SOURce, 3–36
ARM'D indicator, 2–4
ASCII, *Glossary–1*
ASCII data format
  for acquired waveforms, 2–55
  for calculated data, 2–56
Attenuation, Glossary–2
Auto-advance acquisition
  command descriptions, 3–29–3–34
  in relation to the acquisition cycle, 2–14
  overview, 2–15–2–17
  setting data format, 3–103
  setting measurement list AAMList, 3–44
  timestamp record, 2–15
  use with Fast Data Channel, 2–16
Auto-configuring the Logical Address, 1–3
Automated measurements, 2–34–2–36
Automatic trigger mode, 2–25, Glossary–2
Auxiliary trigger, 2–22
AVERage, 3–39
Average acquisition mode, Glossary–2
  overview, 2–18–2–20
AVERage Subsystem, 3–39
AVERage:COUNt, 3–40
AVERage:TYPE, 3–41

www.valuetronics.com

B

backplane. *See* VXIbus
Backus-Naur Form, *3–6*
Bandwidth, Glossary–2
BAUD, RS-232 parameter, 3–179
Bharris window, command, 3–76
Binary data format
  for acquired waveforms, 2–55–2–56
  for calculated data, 2–56–2–57
Blackman window, 2–43
  command, 3–76
Blackman-Harris window, 2–42
Block, command argument, *3–7*
BNF (Backus-Naur form), *3–6*
Bus Grant, setting mainframe jumper, 1–4
Byte order, used in data transfer, 3–104
Byte order in transfers, 3–104

C

CAL, used in Incoming Inspection Procedure, 1–20
*CAL?, 3–255
CALC waveform, FFT. *See* FFT math waveform
CALCulate
  derivative. *See* Derivative math waveform
  Differentiation, 2–39
  Integration, 2–39
CALCulate Subsystem, 3–43–3–96
  CALC model overview, 2–31–2–32
CALCulate:AAMList, 3–44
CALCulate:AAMList:STATe, 3–45
CALCulate:DATA?, 3–46
CALCulate:DATA:PREamble?, 3–47
CALCulate:DERivative:STATe, 3–48
CALCulate:FEED, 3–49
CALCulate:FEED2, 3–51
CALCulate:FEED2:CONTEXT, 3–52
CALCulate:FILTer:FREQuency, 3–54
CALCulate:FILTer:FREQuency:CENTer, 3–55
CALCulate:FILTer:FREQuency:HPASs, 3–57
CALCulate:FILTer:FREQuency:LPASs, 3–58
CALCulate:FILTer:FREQuency:SPAN, 3–59
CALCulate:FILTer:FREQuency:SREJection, 3–60
CALCulate:FILTer:FREQuency:STARt, 3–61
CALCulate:FILTer:FREQuency:STATe, 3–63
CALCulate:FILTer:FREQuency:STOP, 3–64
CALCulate:FILTer:FREQuency:TWIDth, 3–65
CALCulate:FORMat, 3–66
CALCulate:IMMediate, 3–68
CALCulate:INTegral:STATe, 3–69
CALCulate:PATH, 3–70
  use to define calculations, 2–32–2–33
CALCulate:PATH:EXPRession, 3–71
CALCulate:SMOothing, 3–73
CALCulate:SMOothing:POINts, 3–74
CALCulate:TRANsform:FREQuency:STATe, 3–75
CALCulate:TRANsform:FREQuency:WINDow, 3–76
CALCulate:WMList, 3–78
CALCulate:WMParameter:HIGH, 3–83
CALCulate:WMParameter:HMEThod, 3–84
CALCulate:WMParameter:HREFerence, 3–88
CALCulate:WMParameter:HREFerence:RELative, 3–89
CALCulate:WMParameter:LMEThod, 3–86
CALCulate:WMParameter:LOW, 3–85
CALCulate:WMParameter:LREFerence, 3–90
CALCulate:WMParameter:LREFerence:RELative, 3–91
CALCulate:WMParameter:MREFerence, 3–92
CALCulate:WMParameter:MREFerence:HYSTeresis, 3–93
CALCulate:WMParameter:MREFerence:RELative, 3–94
CALCulate:WMParameter:RMEThod, 3–95
Calculated data formats, 2–56–2–57
Calculations, 2–31–2–49
  CALC block overview, 2–31–2–32
  command descriptions, 3–43–3–96
  data sources, 2–33
  derivative waveform, 2–39
  digital filtering, 2–46–2–49
  expression based model, 2–33–2–34
  expression for auto-advance records, 2–34
  Fast Fourier Transforms, 2–40–2–46
  FFT waveform leakage, 2–45
  FFT windows, 2–42
  FFT windows illustrated, 2–45
  for auto-advance acquisition, 3–44
  integral waveforms, 2–39–2–40
  measurement parameters, 2–36
  parameters for measurements, 2–36
  PATH:EXPRession syntax defined, 2–33
  SCPI calculation model, 2–32–2–33
  setting a second source, 3–51
  setting the context, 3–52
  setting the result format, 3–102
  setting the source, 3–49
  transforms (FFT), 2–40–2–46
  waveform math operators, 2–38
Calibrating, Operation Status Register bit, 4–5
Calibrating OSR, 3–151, 3–160, 4–7

www.valuetronics.com

CALibration, 3–97
CALibration Subsystem, command listing, 3–97
CALibration:RESults?, 3–98
CALibration:RESults:VERBose?, 3–99
CARea, 2–35
  measurement, 3–78
CENTer, filter command, 3–55
CH 1 to 4 Probe, Operation Status Register bits, 4–5
CH connectors, 2–3
Chaining Commands and Queries, 3–3
Channel, Glossary–2
  Trigger input, 2–22
Channel enable, FUNCtion command, 3–106
Channels, digitizer configuration, 2–9
Circuit loading, Glossary–2
*CLS, 3–256
CMEan, measurement, 3–78
Command
  Block argument, *3–7*
  Rules for forming, *3–1*
  Syntax, *3–1*
Command Error Messages, 4–12
Command Groups, 3–9–3–26
Command synchronization, 4–11
Commands
  Command Groups, 3–9–3–26
  listing of commands, 3–27
  overview of subsystems, 3–27
  structure of IEEE 488.2 commands, 3–5
Compensating probes
  enabling the PROBE COMPENSATION signal, 3–140
  selecting the PROBE COMPENSATION source, 3–141
Condition Register, 4–1
  Operation, 4–6
  Questionable, 4–7
Configuration, 1–2–1–4
  environmental requirements, 1–4
  hardware installation, 1–5
  mainframe jumpers, 1–4
  setting the logical address, 1–2
  software installation, 1–8–1–12
  VXI mainframe, 1–4
  Warning, 1–4
Configuring the VXIbus Mainframe, 1–4
Connector, VXIbus, 2–64
Connectors
  description, 2–3
  Rear, description of, 2–64
Constructed Mnemonics, 3–6
Continuous acquisition, 3–118
Controller, Definition of, *Glossary–2*
Cooling Requirements, 1–4
COUPling
  TRIGger:A, 3–212
  TRIGger:A <presets>, 3–213
  TRIGger:B, 3–225
Coupling
  Ground, Glossary–4
  setting input coupling, 3–124
  trigger, 2–23–2–30
  TRIGger:B:<preset>, 3–226
Creating commands, 3–1
Creating queries, 3–2
CRMS, measurement, 3–78
CROSs, measurement, 3–78
Cross, defined, 2–35
CVI Executable Files, 1–11
CVI library, 1–10
Cycle Area, measurement, 3–78
Cycle area, Glossary–2
Cycle Mean, measurement, 3–78
Cycle mean, 2–35, Glossary–3
Cycle RMS, 2–35, Glossary–3
  measurement, 3–78

D

Data formats
  for acquired waveforms, 2–55–2–56
  for calculated data, 2–56–2–57
  for waveform preambles, 2–57–2–62
DATA Subsystem, 3–105
Data transfer
  setting byte order, 3–104
  TRACe command, 3–202
  using the Fast Data Channel, 2–16
Data transfer formats, 2–55–2–62
DATA?, 3–114
DATA:PREamble?, 3–115
DC
  INPut coupling, 3–124
  input coupling, 2–4
  measurement, 3–79
  trigger coupling discussion, 2–23
  TRIGger:A coupling, 3–212
  TRIGger:B coupling, 3–225
DC coupling, Glossary–3
DCD, RS-232 parameter, 3–180
DCNReject, trigger coupling discussion, 2–23
DELay, measurement, 3–78, 3–79, 3–80
Delay
  TRIGger:A by time, 3–215
  TRIGger:B by events, 3–229

www.valuetronics.com

TRIGger:B by time, 3–228
Delay time, Glossary–3
Delayed Runs After Main, 2–26–2–30
Delayed trigger, 2–26–2–30
commands controlling delay, 2–27
Delayed Triggerable, 2–26–2–30
DERivative, 3–48
Derivative math waveform, record length of, 2–39
Derivative waveform, 2–39
applications, 2–39
derivation of, 2–39
enabling with STATe, 3–48
Description, Product, 1–1
Device Dependent Error Messages, 4–14
Differentiation, waveform, 2–39
Digital filtering
controlling commands, 2–47
overview of types, 2–46
parameters, 2–46
setting BPASs and NOTCh, 2–48
setting the transition width, 2–48
Digitizing, Glossary–3
Digitizing waveforms, 2–9
Duty cycle, Glossary–5, Glossary–6
Dynamic Auto Configuration, 1–3

E

ECHO, RS-232 parameter, 3–182
ECLTrg:POLarity<n>, 3–138
ECLTrg:SOURce<n>, 3–139
Edge trigger, 2–25, Glossary–3
Edge Trigger A, 2–22
Edge Trigger B, 2–22
Enable Registers
Event Status Enable Register, 4–4
Operation Enable Register, 4–6
Questionable Enable Register, 4–8
Questionable Status Queue Enable Register, 4–8
Service Request Enable Register, 4–3
Status Queue Enable Register, 4–6
Envelope acquisition mode, Glossary–3
overview, 2–18–2–20
EOM, *Glossary–3*
Erase memory function, 3–194
ERESponse, RS-232 parameter, 3–183
Error, Using the Status Queue, 4–8
Error Messages, 4–12
execution warning messages, 4–14
Error messages, system events, 4–14
ERRor?, query for system errors, 3–189
ERT sampling, definition of, *Glossary–3*
*ESE, 3–257, 4–4
*ESR?, 3–258, 4–3, 4–5
description of Standard Event Status Register, 4–3
Event delay, TRIGger:B:ECOunt command, 3–229
Event Reporting Process, 4–10
Event Reporting System, 4–1
Event Status Enable Register, 4–4, Glossary–3
Events, using to delay triggering, 2–26
Examples of commands, 2–71–2–82
Execution Error Messages, 4–13
Execution Warning Messages, 4–14
EXPRession, CALC command, 3–71
Expression model for calculations, 2–33–2–34
Expression syntax defined, 2–33
Extended Real-time sampling, definition of, *Glossary–3*
Extended real-time acquisition, 2–13
External Trigger, source compatibility, 2–22
EXTERNAL TRIGGER INPUT connector, 2–5

F

Factory Default RS-232 Settings, 1–14
Fall Time, measurement, 3–78
Fall time, Glossary–4
Fast Data Channel
overview of use, 2–50–2–54
specify waveforms to transfer, 3–208
use with auto-advance acquisition, 2–16
waveform transfer command, 3–206
Fast Fourier Transforms, description, 2–40–2–46
FEED
specifying CALC block source, 3–49
specifying CALC context parameter, 3–52
FEED2, specifying CALC block source, 3–51
FFT frequency domain record, 2–40
FFT waveform, 2–40
aliasing, 2–41
bandpass characteristics of windows, 2–45
derivation of, 2–40
frequency range, 2–41
frequency resolution, 2–41
illustration of windows, 2–45
selecting an FFT window, 2–42–2–45
setting data format, 3–66
TRANsform command, 3–75
undersampling, 2–41
FIDUCIAL INPUT connector, 2–5
FILTer, 3–230
See also CALC:FILTer commands
TRIGger:A high pass, 3–217
TRIGger:A low pass, 3–216

www.valuetronics.com

TRIGger:B high pass, 3–231
TRIGger:B low pass, 3–230
FORMat, 3–101
CALC results command, 3–66
FORMat Subsystem, 3–101
FORMat:BORDer, 3–104
FORMat:CALCulate, *3–102*
FORMat:TRACe:AATS, 3–103
Formats, for data transfer, 2–55–2–62
Four channel configuration, 2–9
FREQuency, 2–35
input filter setting, 3–126
measurement, 3–78
selecting filter type, 3–54
Frequency, Glossary–4
Front panel
connectors, 2–3
illustration of, 2–3
indicators, 2–3
SERIAL INTERFACE pin out, 2–69
Front panel connectors
ARM INPUT, 2–5
CH1 to CH4, 2–3
EXTERNAL TRIGGER INPUT, 2–5
FIDUCIAL INPUT, 2–5
PROBE COMPENSATION, 2–4
REFERENCE OUTPUT, 2–4
SERIAL INTERFACE, 2–5
Front Panel Indicators, during power on, 1–7
Front panel indicators
ACCESSED, 2–4
ARM'D, 2–4
READY, 2–4
TRIG'D, 2–4
FTIMe, 2–35
measurement, 3–78
FUNCtion, 3–106
FUNCtion Subsystem, 3–105
FUNCtion:CONCurrent, 3–112
FUNCtion:OFF, 3–109
FUNCtion:OFF:ALL, 3–110
FUNCtion:OFF:COUNt?, 3–111
FUNCtion:STATe, 3–113
FUNCtion[:ON]:ALL, 3–107
FUNCtion[:ON]:COUNt?, 3–108
Functional Overview, 2–7–2–8
Functional Tests, used in Incoming Inspection Procedure, 1–17–1–19

G

GAIN, measurement, 3–78
Gain, 2–35
General Rules, for using SCPI commands, 3–4
Getting Started, 1–1–1–12
GHZ, *Glossary–4*
Glitch trigger, 2–28–2–31
GPIB, *Glossary–4*
GROund, INPut coupling, 3–124
Ground, input coupling mode, 2–4
Ground coupling, Glossary–4

H

Halting acquisition, 3–121
Hamming window, 2–42, 3–76
Hanning window, 2–42, 3–76
Hardware installation, 1–5
illustration of installation, 1–6
Power-On Procedure, 1–7
HFReject, trigger coupling discussion, 2–23
Hierarchy Tree, 3–1
HIGH, 2–35
measurement, 3–79
measurement parameter, 3–83
methods of setting, 2–38
High, Glossary–4
HMEThod, measurement parameter, 3–84
Holdoff
range, 2–24
trigger types, 2–24
Holdoff, trigger, Glossary–4
HOLDoff:TIME, 3–219
HPASs, high pass filter command, 3–57
HREFerence, measurement parameter, 3–88
HZ, *Glossary–4*

I

IACK, setting mainframe jumper, 1–4
*IDN?, 3–259
IEEE, *Glossary–4*
IEEE 488.2 Common Commands, 3–5
command descriptions, 3–255
IEEE Std 488.2-1987, *3–6*
IMMediate
ARM:SOURce, 3–36
CALC command, 3–68
IMMediate Trigger, 2–22
Impedance, input channel setting, 3–127
Incoming Inspection Procedure, 1–13–1–20
Functional Tests, 1–17
measure time reference, 1–17

www.valuetronics.com

measure voltage reference, 1–19
Self Cal, 1–20
Self Tests, 1–15
System setup, 1–14
Test Equipment, 1–13
Indicators
ACCESSED, 2–4
ARM'D, 2–4
description, 2–3
READY, 2–4
TRIG'D, 2–4
Initial Set Up, 1–2–1–4
INITiate, 3–117
acquisition cycle, 2–14
INITiate Subsystem, 3–117
INITiate:CONTinuous, 3–118
INITiate:COUNt, 3–119
Input
automatic input protection, 2–4
Coupling, 2–4
impedance, 2–4
setting vertical offset, 3–251
setting vertical range, 3–253
triggering from input channels, 2–22
Input connectors
ARM input, 2–5
description, 2–3
EXTERNAL TRIGGER input, 2–5
FIDUCIAL INPUT, 2–5
PROBE COMPENSATION, 2–4
REFERENCE OUTPUT, 2–4
SERIAL INTERFACE, 2–5
Input enable, FUNCtion command, 3–106
Input Filter, setting input filtering, 3–125
INPut Subsystem, 3–123
INPut:COUPling, 3–124
INPut:FILTer, 3–125
INPut:FILTer:FREQuency, 3–126
INPut:IMPedance, 3–127
INPut:PROTection:STATe, 3–128
Installation, 1–5–1–7
Caution, 1–5
hardware, 1–5
illustration of hardware, 1–6
in D-size mainframe, 1–5
incoming inspection procedure, 1–13
mainframe configuration, 1–4
mainframe jumpers, 1–4
of VXIplug&play Software, 1–9
Power-On Procedure, 1–7
removal from mainframe, 1–6
software installation, 1–8–1–12
Installed Files, 1–10
Installing the VXIplug&play software, 1–9
INTegral, 3–69
Integral math waveform, 2–39
derivation of, 2–39
Integral waveform
enabling with STATe, 3–69
record length of, 2–40
Integration, waveform, 2–39
interrupts, 2–63

J

Jumper Settings, 1–4

L

LBUFfer, RS-232 parameter, 3–184
Leakage, in FFT waveforms, 2–45
LEVel
TRIGger:A, 3–220
TRIGger:B, 3–233
Level, Trigger, 2–23–2–24
LFReject, trigger coupling discussion, 2–23
Library, of software functions, 1–10
LMEThod, measurement parameter, 3–86
Loading settings, 3–195
Logical Address
setting, 1–2
setting auto-configuration address, 1–3
setting static address, 1–3
Logical Address Switches, illustration of, 1–3
Looping, during acquisition, 2–14
LOW, 2–35
measurement, 3–79
measurement parameter, 3–85
methods of setting, 2–38
Low, Glossary–5
LPASs, low pass filter command, 3–58
LREFerence, measurement parameter, 3–90
*LRN?, 3–260

M

Mainframe, configuring, 1–4
Manuals, related, xxi
Math waveform, integral. *See* Integral math waveform
MAXimum, 2–35
measurement, 3–79
Maximum, Glossary–5
MEAN, 2–35
measurement, 3–79

www.valuetronics.com

Mean, Glossary–5
Measure Time Reference, 1–17
Measure Voltage Reference, 1–19
Measurement
  See also Calculations
  AC, 2–36
  Amplitude, 2–35, Glossary–1
  Area, 2–35, Glossary–1
  Burst width, Glossary–2
  CALC block overview, 2–31–2–49
  Cross, 2–35
  Cycle area, 2–35, Glossary–2
  Cycle mean, 2–35, Glossary–3
  Cycle RMS, 2–35, Glossary–3
  definition of measurements, 2–34–2–36
  Duty cycle, Glossary–5, Glossary–6
  Fall time, 2–35
  Frequency, 2–35, Glossary–4
  Gain, 2–35
  High, 2–35, Glossary–4
  Low, 2–35, Glossary–5
  Maximum, 2–35, Glossary–5
  Mean, 2–35, Glossary–5
  Minimum, 2–35, Glossary–5
  Ncross, 2–35
  Negative duty cycle, 2–36
  Negative width, 2–36
  Overshoot, Glossary–6
  Pcross, 2–36
  Peak to peak, 2–36, Glossary–6
  Period, 2–36, Glossary–6
  Phase, 2–36
  Positive duty cycle, 2–36
  Positive width, 2–36
  Propagation delay, 2–35
  Rise time, 2–36, Glossary–7
  RMS, 2–36, Glossary–7
  setting parameters, 2–36–2–38
  setting reference levels, 2–38
  Standard deviation, 2–36
  Time between triggers, 2–36
  Undershoot, Glossary–5
  Width, Glossary–5, Glossary–6
Measurements, 2–34–2–49
  See also Calculations
  Algorithms, B–1–B–30
  definition of measurements, 2–34–2–36
  definitions of, 3–78–3–81
Measuring & Acquiring, Operation Status Register bit, 4–5
Memory erase function, 3–194
MEMory Subsystem, 3–131
MEMory:DATA, 3–131
MEMory:NSTates?, 3–133
MEMory:STATe:CATalog?, 3–134
MEMory:STATe:DEFine?, 3–135
Message Terminators, 3–6
Messages, Error, 4–12
Methods of setting HIGH and LOW, 2–38
Methods of setting Reference levels, 2–38
MHZ, *Glossary–4*, *Glossary–5*
MID, measurement, 3–79
MINimum, 2–35
  measurement, 3–79
Minimum, Glossary–5
Modes, of triggering, 2–25
Modes of Acquisition, 2–12–2–14
  extended real time, 2–13
  real time, 2–13
MREFerence, measurement parameter, 3–92

N

NCROSs, measurement, 3–79
Ncross, defined, 2–35
NDUTycycle, 2–36
  measurement, 3–79
Negative Dutycycle, measurement, 3–79
Negative Transition
  Operation Status Register, 4–6
  Questionable Status Register, 4–7
Negative Width, measurement, 3–79
Noise reject
  TRIGger:A, 3–218, 3–232
  triggering options, 2–23
Nominal Traits, Defined, *A–2*
Normal trigger mode, 2–25, Glossary–5
NWIDth, 2–36
  measurement, 3–79
Nyquist frequency, for FFT waveforms, 2–41

O

OFFSet, setting input offset, 3–251
Offset, vertical, 2–19
*OPC, 3–261
OPC bit, 3–258, 4–4
*OPC?, 4–11
  using to synchronize commands, 4–11
Operating Basics, 2–1–2–2
Operation overview, 2–7–2–8
Operation Status, 4–6
OPERation status register, query command, 3–151
Operation Status Register, 4–5
  bit definitions, 3–151, 4–5

www.valuetronics.com

Optional Accessories, 1–2
Out of range data, 2–56
Output Queue, 4–9
Output queue, 4–9
Output Queues, 4–8–4–9
OUTPut Subsystem, 3–137
OUTPut:ECLTrg:SOURce<n>?, 3–139
OUTPut:ECLTrg<n>, 3–137
OUTPut:ECLTrg<n>?, 3–137
OUTPut:ELTrg:POLarity<n>, 3–138
OUTPut:PCOMpensate, 3–140
OUTPut:PCOMpensate:FUNCtion, 3–141
OUTPut:REFerence, 3–142
OUTPut:REFerence:FUNCtion, 3–143
OUTPut:TTLTrg<n>, 3–144
OUTPut:TTLTrg<n>:POLarity, 3–145
OUTPut:TTLTrg<n>:SOURce, 3–146
Overshoot, Glossary–6

P

P1 Connectors, Description of, 2–64
P2 Connectors, Description of, 2–64
PACE, RS-232 parameter, 3–185
Parameter Types Used in Syntax Descriptions, 3–2
Parameters
  for digital filters, 2–46–2–48
  measurement, 2–36
  of acquisition, 2–10
PARity, RS-232 parameter, 3–186
PATH
  CALC command, 3–70
  using to define a SCPI calculation, 2–32–2–34
PATH:EXPRession, CALC command, 3–71
PCROSs, measurement, 3–79
Pcross, defined, 2–36
PDUTycycle, measurement, 3–79
Peak To Peak, measurement, 3–79
Peak to peak, Glossary–6
Performance Characteristics, Nominal. *See Nominal Traits*
PERiod, 2–36
  measurement, 3–79
Period, Glossary–6
Phase, defined, 2–36
Plug&play Software, 1–8–1–12
Polarity, setting for Pulse glitch triggering, 2–29
Position, of trigger point, 2–24
Positioning the waveform record, horizontal position relative to trigger point, 2–12
Positive Dutycycle, measurement, 3–79
Positive Transition
  Operation Status Register, 4–6
  Questionable Status Register, 4–7
Positive Width, measurement, 3–79
Posttrigger, Glossary–6
Power-On Procedure, 1–7
  front panel indicators during, 1–7
Preface, xxi
PRESet, RS-232 parameter, 3–187
Pretrigger, Glossary–6
PROBE COMPENSATION
  enabling, 3–140
  selecting source signal, 3–141
PROBE COMPENSATION connector, 2–4
Probes
  Compensation, Glossary–6
  Definition, Glossary–6
  PROBE COMPENSATION connector, 2–4
  TekProbe connections, 2–3
Product Accessories, 1–1–1–2
Product Description, 1–1
Propagation delay, 2–35
PTPeak, 2–36
  measurement, 3–79
*PUD, 3–262
Pulse Trigger, sources, 2–22
Pulse trigger, 2–25, 2–28
Pulse Triggering
  Glitch, 2–28–2–31
  Width, 2–28–2–31
PWIDth, 2–36
  measurement, 3–79

Q

Qualify, setting for Pulse glitch triggering, 2–29
Quantizing, Glossary–6
Questionable Enable, 4–8
Questionable Status, 4–7
QUEStionable status register, query command, 3–160
Questionable Status Register, 4–6
Queue
  error, 4–8
  Output, 4–9
  Status, 4–8
Queues, Output queue, 4–9

www.valuetronics.com

R

Range
  setting vertical range, 3–253
  vertical input, 2–19
*RCL, 3–263
READY indicator, 2–4
Real time acquisition, 2–13
Real-time sampling, *Glossary–7*, Glossary–7
Record length, Glossary–7
  derivative math waveforms, 2–39
  discussion of, 2–10
  integral waveforms, 2–40
Record positioning, 2–11
Rectangular window, 2–42, 3–76
Reference Clock, setting source, 3–150
Reference levels, for measurements, 2–38
Reference memory, Glossary–7
REFERENCE OUTPUT
  enabling with OUTPut command, 3–142
  in Incoming Inspection Procedure, 1–17
  setting output signal, 3–142, 3–143
REFERENCE OUTPUT connector, 2–4
Register
  definition of status registers, 4–1
  Event Status Enable, 4–4
  Operation Condition, 4–6
  Operation Status, 4–6
  Operation Status Register, 4–5
  Questionable Status Register, 4–6
  Service Request Enable, 4–3
  Standard Event Status, 4–3
  status, 4–1
  Status Byte, 4–3
Related Manuals, xxi
Release Notes, for Plug & Play Software, 1–9
Removal from VXIbus Mainframe, 1–6
Retainer screws, use for installation, 1–5
Rise Time, 3–80
Rise time, Glossary–7
RMEThod, measurement parameter, 3–95
RMS, 2–36, Glossary–7
  measurement, 3–80
Root level nodes, diagram, 2–8
ROSCillator Subsystem, 3–149
ROSCillator:SOURce, 3–150
RS-232 interface
  factory settings, 1–14
  front panel connector, 2–5
RS-232 port, 2–69
*RST, 3–264
RTIMe, 2–36
  rise time measurement, 3–80
RTS, RS-232 parameter, 3–181
Rules, command forming, *3–1*

S

Sample acquisition mode, Glossary–7
Sample interval, Glossary–7
  discussion of, 2–10
Sampling, Glossary–7
*SAV, 3–264
Saving settings, MEMory command, 3–131
SBITs, RS-232 parameter, 3–188
SCPI
  subsystem hierarchy tree, 3–1
  system diagram, 2–8
  version, 3–196
SCPI CALCulate Model, 2–32–2–33
SCPI commands and queries syntax, 3–1–3–5
  abbreviating, 3–3
  chaining commands, 3–3
  creating commands, 3–1
  creating queries, 3–2
  general rules, 3–4
  parameter types, 3–2
SDEViation, 2–36
  measurement, 3–80
Self Cal, used in Incoming Inspection Procedure, 1–20
Self Test, executing, 3–197
Self Tests, used in Incoming Inspection Procedure, 1–15–1–16
SERIAL INTERFACE, 2–69
Serial Interface, factory settings, 1–14
SERIAL INTERFACE connector, 2–5
SERial:BAUD, 3–179
SERial:CONTrol:DCD, 3–180
SERial:CONTrol:RTS, 3–181
SERial:ECHO, 3–182
SERial:ERESponse, 3–183
SERial:LBUFfer, 3–184
SERial:PACE, 3–185
SERial:PARity, 3–186
SERial:PRESet, 3–187
SERial:SBITs, 3–188
Service Request Enable Register, 4–3
Setting the Logical Address, 1–2
Setting up CALC blocks, 2–31
Settings
  loading, 3–195
  mainframe jumpers, 1–4
Settings onboard, MEMory command, 3–131
SLOPe
  TRIGger:A, 3–221

www.valuetronics.com

TRIGger:B, 3–234
Slope, Glossary–7
Slope, Trigger, 2–23–2–24
SMOothing, CALC command, 3–73
Software Installation, 1–8–1–12
list of installed files, 1–10
Windows Installation, 1–9
Software library, 1–10
SOURce
TRIGger:A, 3–222
TRIGger:B, 3–235
TRIGger:PULSe, 3–241
Source, for CALC blocks, 2–32
SPAN, filter command, 3–59
Specification, *A–1–A–14*
banner, 1–1
*SRE, 3–265, 4–3
SREJection, filter command, 3–60
Stale data, Glossary–8
Standard Accessories, 1–1–1–2
Standard Deviation, measurement, 3–80
Standard Event Status Register, 4–3
STARt, filter command, 3–61
Start acquisition, 3–117
STATe
auto-advance measurement (AAMList) enable, 3–45
Derivative enable, 3–48
enabling input protection, 3–128
filter enable, 3–63
input channel enable, 3–113
Integral enable, 3–69
Transform (FFT) enable, 3–75
Static Logical Address, 1–3
Status and Event Reporting Process, 4–10
Status and Event Reporting System, 4–1
Status and Events, 4–1–4–14
Queues, 4–8
Service Request Enable Register, 4–3
Status Byte Register, 4–3
Status Registers, 4–1–4–14
Status Byte Register, 4–3
Status Queue, 4–8
Status Queue Enable Negative Transition, 4–8
Status Queue Enable Positive Transition, 4–8
STATus Subsystem, command descriptions, 3–151
STATus:OPERation?, 3–151, 4–5, 4–6
STATus:OPERation:CONDition?, 3–153, 4–6
STATus:OPERation:ENABle, 3–154, 4–6
STATus:OPERation:NTRansition, 3–155, 4–6
STATus:OPERation:PTRansition, 3–156, 4–6
STATus:OPERation:QENable:NTRansition, 3–157, 4–6
STATus:OPERation:QENable:PTRansition, 3–158, 4–6
STATus:PRESet, 3–159
STATus:QUEStionable?, 3–160, 4–6, 4–7
STATus:QUEStionable:CONDition?, 3–161, 4–7
STATus:QUEStionable:ENABle, 3–162, 4–8
STATus:QUEStionable:NTRansition, 3–163, 4–7
STATus:QUEStionable:PTRansition, 3–164, 4–7
STATus:QUEStionable:QENable:NTRansition, 3–165, 4–8
STATus:QUEStionable:QENable:PTRansition, 3–166, 4–8
STATus:QUEue?, 4–8
STATus:SESR:QENable, 3–168
*STB?, 3–266, 4–3
STOP, filter command, 3–64
Stopping acquisition, 3–121
Sweep
setting OFFSet, 2–11
setting OREFerence, 2–11
setting record length, 2–10
SWEep Subsystem, command descriptions, 3–169
SWEep:OFFSet:POINts, 3–170–3–171
SWEep:OFFSet:TIME, 3–171–3–172
SWEep:OREFerence:LOCation, 3–173–3–174
SWEep:POINts, 3–174–3–175
SWEep:TIME, 3–176–3–177
SWEep:TINTerval, 3–177–3–178
Synchronization Methods, 4–11
Synchronizing commands, 4–11
*OPC?, 4–11
Syntax, Command, *3–1*
System, block diagram, 2–8
System Errors, 4–8
System Events, 4–14
System Operation, system architecture, 2–7
System Setup, for incoming inspection, 1–14
SYSTem Subsystem, command descriptions, 3–179
SYSTem:COMMunicate:SERial:BAUD, 3–179–3–180
SYSTem:COMMunicate:SERial:CONTrol:DCD, 3–180–3–181
SYSTem:COMMunicate:SERial:CONTrol:RTS, 3–181–3–182
SYSTem:COMMunicate:SERial:ECHO, 3–182–3–183
SYSTem:COMMunicate:SERial:LBUFfer, 3–184–3–185
SYSTem:COMMunicate:SERial:PACE, 3–185–3–186
SYSTem:COMMunicate:SERial:PARity, 3–186
SYSTem:COMMunicate:SERial:PRESet, 3–187–3–188
SYSTem:COMMunicate:SERial:SBITs, 3–188
SYSTem:COMMunicate:SERial:VERBose, 3–183
SYSTem:ERRor?, 3–189
SYSTem:ERRor:ALL?, 3–190
SYSTem:ERRor:CODE?, 3–191
SYSTem:ERRor:CODE:ALL?, 3–192
SYSTem:ERRor:COUNt?, 3–193

www.valuetronics.com

SYSTem:PROTect, 3–193
SYSTem:SECurity:IMMediate, 3–194
SYSTem:SET, 3–195
SYSTem:VERSion?, 3–196

T

Taking measurements. *See* Calculations
Tek Secure, Glossary–8
TekProbe, probe scaling, 2–3
TEKSecure, *Glossary–8*
Terminators, message, 3–6
TEST, 3–197
Test Equipment, for Incoming Inspection Procedure, 1–13
TEST Subsystem, command descriptions, 3–197
TEST:RESults?, 3–198
TEST:RESults:VERBose?, 3–199
Testing, Operation Status Register bit, 4–5
Time base, Glossary–8
Time Reference, verification, 1–17
Timestamp, in auto-advance acquisition, 2–16
Trace format, setting, 3–101
TRACe Subsystem, command descriptions, 3–201
TRACe?, 3–202
TRACe:AATS, 3–103
TRACe:CATalog?, 3–205
TRACe:COPY, 3–206
TRACe:FEED?, 3–207
TRACe:LIST, 3–208
TRACe:POINts?, 3–209
TRACe:PREamble?, 3–203
Transfer waveform
  TRACe command, 3–202
  using Fast Data Channel, 2–50–2–54
Transfer waveform preamble, TRACe:PREamble command, 3–203
Transferring data, setting byte order, 3–104
Transferring waveforms, using the Fast Data Channel, 2–16
Transform (FFT), 2–40
*TRG, 3–267
TRIangular window, 2–43
Triangular window, command, 3–76
TRIG'D indicator, 2–4
Trigger, Glossary–8
  AC Line Voltage, 2–22
  ARM subsystem commands, 3–35
  Auxiliary, 2–22
  Delayed, 2–26–2–30
  Edge, Glossary–3
  Level, 2–23–2–24, Glossary–8
  Pulse, 2–25, 2–28
  Slope, 2–23–2–24
trigger, VXIbus backplane, 2–63
Trigger event, definition of, 2–22
Trigger point, discussion of, 2–11
TRIGger:A:COUPling:<preset>, 3–213
TRIGger:ATRigger, 3–211
TRIGger:B Subsystem, command descriptions, 3–225
TRIGger:B:COUPling, 3–225
TRIGger:B:COUPling:<preset>, 3–226
TRIGger:B:DELay, 3–228
TRIGger:B:ECOunt, 3–229
TRIGger:B:FILTer:HPASs, 3–231
TRIGger:B:FILTer[:LPASs], 3–230
TRIGger:B:LEVel, 3–233
TRIGger:B:SLOPe, 3–234
TRIGger:B:SOURce, 3–235
TRIGger:COUPling, 3–212
TRIGger:DEFine?, 3–214
TRIGger:DELay, 3–215
TRIGger:FILTer:HPASs, 3–217
TRIGger:FILTer:NREJect, 3–218, 3–232
TRIGger:FILTer[:LPASs], 3–216
TRIGger:HOLDoff:TIME, 3–219
TRIGger:LEVel, 3–220
TRIGger:PULSe Subsystem, command descriptions, 3–237
TRIGger:PULSe:CLASs, 3–237
TRIGger:PULSe:GLITch:POLarity, 3–238
TRIGger:PULSe:GLITch:QUALify, 3–239
TRIGger:PULSe:GLITch:WIDTh, 3–240
TRIGger:PULSe:SOURce, 3–241
TRIGger:PULSe:THReshold, 3–242
TRIGger:PULSe:WIDTh:HLIMit, 3–243
TRIGger:PULSe:WIDTh:LLIMit, 3–244
TRIGger:PULSe:WIDTh:POLarity, 3–245
TRIGger:PULSe:WIDTh:QUALify, 3–246
TRIGger:SEQuence2:DEFine?, 3–227
TRIGger:SLOPe, 3–221
TRIGger:SOURce, 3–222
TRIGger:TYPE, 3–223
TRIGger[:A] Subsystem, command descriptions, 3–211
Triggering, 2–21–2–30
  automatic trigger mode, 2–25
  block diagram of Arm/Trigger cycle, 2–21
  commands controlling delay, 2–27
  delay by events, 2–26
  delay by time, 2–26
  Delayed, 2–26
  Edge, 2–25
  External triggering, 2–22
  holdoff modes, 2–24

www.valuetronics.com

IMMediate, 2–22
input channels, 2–22
interaction between delay and holdoff, 2–28
noise reject coupling, 2–23
position in waveform record, 2–24
Pulse glitch, 2–28–2–31
Pulse polarity, 2–29
Pulse width, 2–28–2–31
qualification for Pulse triggering, 2–29
Sources, 2–22
table of triggers and sources, 2–22
trigger coupling, 2–23–2–30
trigger events, 2–22
trigger types, 2–25
VXIbus trigger overview, 2–30
VXIbus triggers, 2–22
Width for Pulse triggering, 2–29
*TST?, 3–268
TTLTrg<n>, 3–144
TTLTrg<n>:POLarity, 3–145
TTLTrg<n>:SOURce, 3–146
Ttrig, defined, 2–36
Tutorial, 2–71–2–82
Acquiring a Signal, 2–74
Averaging and Enveloping a Signal, 2–75
host system requirements, 2–71–2–82
Instrument Setup, 2–72–2–74
Performing Advanced Calculations, 2–78
Performing Basic Calculations, 2–76–2–77
Saving and Recalling Settings, 2–79–2–80
Using Status and Events, 2–81–2–82
TVS600 Plug & Play Software, 1–10
TWIDth, filter parameter command, 3–65
Two channel configuration, 2–9

U

Undershoot, Glossary–5

V

Verification, Incoming Inspection Procedure, 1–13
Vertical offset, discussion on setting, 2–19
Vertical range, discussion on setting, 2–19
Voltage Reference, verification, 1–19
VOLTage Subsystem
command descriptions, 3–247
overrange and underrange points, 2–19
overview, 2–19–2–20
VOLTage:RANGe:LOWer, 3–249
vertical system discussion, 2–19
VOLTage:RANGe:OFFSet, 3–251
VOLTage:RANGe:PTPeak, 3–253
VOLTage:RANGe[:UPPer], 3–248
vertical system discussion, 2–19
VXIbus
device type, 2–63
interface description. *See* VXIbus
interrupts, 2–63
pinout, 2–64
protocol, 2–63
specification, 2–63
trigger overview, 2–30
triggers, 2–63
VXIbus ECL trigger, OUTPut command, 3–137
VXIbus Mainframe, configuring, 1–4
VXIbus Trigger, source compatibility, 2–22
VXIbus TTL trigger, OUTPut command, 3–144
VXIplug&play software, installing, 1–9

W

*WAI, 3–269
Waiting for Arm, Operation Status Register bit, 4–5
Waiting for Trigger, Operation Status Register bit, 4–5
Waveform, Glossary–9
data transfer formats, 2–55–2–56
Interval, Glossary–9
Math, 2–38
preamble formats, 2–57–2–62
record position relative to the trigger, 2–11
sampling discussion, 2–9
setting record length, 2–10
setting transfer format, 3–101
transfer command, 3–202
Waveform calculations, 2–31–2–49
Waveform differentiation, 2–39
Waveform FFTs, 2–40–2–46
Waveform integration, 2–39
Waveform math, 2–38
Waveform preamble, transfer command, 3–203
Waveform record
FFT, 2–40
FFT source, 2–40
Waveform transfer
data formats, 2–55–2–62
using the Fast Data Channel, 2–50–2–54
Waveforms, waveform function algorithms, *B–13*
Width, Glossary–5, Glossary–6
setting for Pulse glitch triggering, 2–29
Width trigger, 2–28–2–31
WINDow, transform window selection, 3–76
Window
Bharris, 3–76

www.valuetronics.com

Blackman, 2–42, 2–43, 3–76
Blackman-Harris, 2–42
characteristics of, 2–43
for FFT waveforms, 2–42
Hamming, 2–42, 3–76
Hanning, 2–42, 3–76
rectangular, 2–42, 3–76
triangular, 2–42, 2–43, 3–76
Windows, FFT windows illustrated, 2–45
Windows Installation, 1–9
WMList, measurement selection command, 3–78

Z

Zero phase reference point, 2–41

www.valuetronics.com