SDS 系列
数字示波器

编程手册

CN11I



深圳市鼎阳科技股份有限公司
SIGLENT TECHNOLOGIES CO., LTD.

目录

1 版本声明 1

1.1 CN11I 版本 1

1.2 CN11H 版本 1

1.3 CN11G 版本 2

1.4 CN11F 版本 3

2 机型支持 4

3 编程概述 5

3.1 建立通信 5

NI-VISA 的安装 5

连接仪器 8

3.2 远程控制的实现 9

用户自定义程序 9

通过 NI-MAX 发送 SCPI 命令 9

通过 Telnet 发送 SCPI 命令 9

通过 Socket 发送 SCPI 命令 10

4 SCPI 语言简介 11

4.1 命令和查询格式 11

4.2 命令缩写 11

4.3 命令符号 12

4.4 参数类型 12

5 命令与查询 14

5.1 IEEE488.2 通用命令 14

*IDN 14

*OPC 14

*RST 15

5.2 Root(:)命令系统 16

:AUToset 16

:PRINt 16
:FORMat:DATA 16
5.3 ACQuire 命令系统 18
:ACQuire:AMODe 18
:ACQuire:CSWeep 18
:ACQuire:INTerpolation 19
:ACQuire:MMANagement 19
:ACQuire:MODE 20
:ACQuire:MDEPth 20
:ACQuire:NUMACq 22
:ACQuire:POINts 22
:ACQuire:RESolution 23
:ACQuire:SEQuence 23
:ACQuire:SEQuence:COUNt 23
:ACQuire:SRATe 24
:ACQuire:TYPE 24
:ACQuire:XY:DISPlay 25
5.4 BODE 命令系统 26
:BODE 26
:BODE:SOURce:INPut 26
:BODE:SOURce:OUTPut<x> 27
:BODE:SOURce:GAIN 27
:BODE:SWEep:TYPE 28
:BODE:SWEep:MODE 28
:BODE:SWEep:COUNt 29
:BODE:SWEep:POINt 29
:BODE:SWEep:SEGMent 29
:BODE:FREQuency:MODE 30
:BODE:FREQuency:STARt 30
:BODE:FREQuency:STOP 31
:BODE:FREQuency:CENTer 31
:BODE:FREQuency:SPAN 32

:BODE:WGEN:AMPLitude 32
:BODE:WGEN:OFFSet 33
:BODE:WGEN:REFLevel 33
:BODE:WGEN:UNIT 33
:BODE:WGEN:LOAD 34
:BODE:WGEN:INTerface 34
:BODE:WGEN:IPADdress 35
:BODE:OPERate 35
:BODE:DATA 36
:BODE:DATA:SAVE 36
:BODE:MEASure:P<n> 37
5.5 CHANnel 命令系统 38
:CHANnel:REFerence 38
:CHANnel<n>:BWLimit 38
:CHANnel<n>:COUPling 39
:CHANnel<n>:IMPedance 39
:CHANnel<n>:INVert 40
:CHANnel<n>:LABel 40
:CHANnel<n>:LABel:TEXT 41
:CHANnel<n>:OFFSet 41
:CHANnel<n>:PROBe 42
:CHANnel<n>:SCALe 43
:CHANnel<n>:SKEW 43
:CHANnel<n>:SWITch 44
:CHANnel<n>:UNIT 44
:CHANnel<n>:VISible 45
:CHANnel<n>:ZSCale 45
:CHANnel<n>:ZOFFset 46
5.6 COUNter 命令系统 47
:COUNter 47
:COUNter:CURRent 47
:COUNter:LEVel 48

:COUNter:MODE....48
:COUNter:SOURce....49
:COUNter:STATistics....49
:COUNter:STATistics:RESet....50
:COUNter:STATistics:VALue....50
:COUNter:TOTalizer:GATE....51
:COUNter:TOTalizer:GATE:LEVel....51
:COUNter:TOTalizer:GATE:SLOPe....52
:COUNter:TOTalizer:GATE:TYPE....52
:COUNter:TOTalizer:RESet....53
:COUNter:TOTalizer:SLOPe....53
5.7 CURSor 命令系统....54
通用命令....54
单组光标命令....56
多光标命令....60
5.8 DECode 命令系统....85
:DECode....85
:DECode:LIST....85
:DECode:LIST:LINE....86
:DECode:LIST:SCRoll....86
:DECode:LIST<n>:RESult....87
:DECode:BUS<n>....87
:DECode:BUS<n>:COPY....88
:DECode:BUS<n>:FORMat....88
:DECode:BUS<n>:PROTocol....88
:DECode:BUS<n>:RESult....89
:DECode:BUS<n>:IIC....90
:DECode:BUS<n>:SPI....93
:DECode:BUS<n>:UART....103
:DECode:BUS<n>:CAN....109
:DECode:BUS<n>:LIN....113
:DECode:BUS<n>:FLEXray....115

:DECode:BUS<n>:CANFd 117
:DECode:BUS<n>:IIS 119
:DECode:BUS<n>:M1553 126
:DECode:BUS<n>:SENT 129
:DECode:BUS<n>:MANChester 134
5.9 DIGital 命令系统（选件） 141
:DIGital 141
:DIGital:ACTive 141
:DIGital:BUS<n>:DISPlay 142
:DIGital:BUS<n>:DEFault 142
:DIGital:BUS<n>:FORMat 142
:DIGital:BUS<n>:MAP 143
:DIGital:D<d> 144
:DIGital:HEIGht 144
:DIGital:LABel<d> 145
:DIGital:POINts 145
:DIGital:POSition 145
:DIGital:SKEW 146
:DIGital:SRATe 146
:DIGital:THReshold<n> 147
5.10 DISPlay 命令系统 148
:DISPlay:AXIS 148
:DISPlay:AXIS:MODE 148
:DISPlay:AXIS:POSition 149
:DISPlay:BACKlight 149
:DISPlay:CLEar 150
:DISPlay:COLor 150
:DISPlay:GRATicule 150
:DISPlay:GRIDstyle 151
:DISPlay:HIDemenu 151
:DISPlay:INTensity 152
:DISPlay:MENU 152

:DISPlay:MENU:HIDE 152
:DISPlay:PERSistence 153
:DISPlay:TRANsparence 154
:DISPlay:TYPE 154
5.11 DVM 命令系统 155
:DVM 155
:DVM:ALARm 155
:DVM:ARANge 156
:DVM:CURRent 156
:DVM:HOLD 156
:DVM:MODE 157
:DVM:SOURce 157
5.12 EYE 命令系统 159
:EYE 159
:EYE:SIGNal 159
:EYE:SOURce 160
:EYE:LEVel 160
:EYE:HYSTeresis 161
:EYE:CLOCk 161
:EYE:CLOCk:MODE 162
:EYE:CLOCk:SRATe 162
:EYE:CLOCk:FONCe 163
:EYE:CLOCk:FOPLL:CUToff 163
:EYE:CLOCk:SOPLL 163
:EYE:CLOCk:SOPLL:JTF:CUToff 164
:EYE:CLOCk:SOPLL:JTF:LBANdwidth 164
:EYE:CLOCk:SOPLL:JTF:PEAKing 165
:EYE:CLOCk:SOPLL:OJTF:CUToff 165
:EYE:CLOCk:SOPLL:OJTF:LBANdwidth 166
:EYE:CLOCk:SOPLL:OJTF:DAMPing 166
:EYE:OVERlay 167
:EYE:OPERate 167

:EYE:RUN 168
:EYE:QUICkview 168
:EYE:MEASure 169
:EYE:MEASure:STATistics 169
:EYE:MEASure:STATistics:RESet 169
:EYE:MEASure:STATistics:HISTOGram 170
:EYE:MEASure:STATistics:AIMLimit 170
:EYE:MEASure:STATistics:MAXCount 171
:EYE:MEASure:P<n> 171
:EYE:MEASure:P<n>:TYPE 171
:EYE:MEASure:P<n>:VALue 173
:EYE:MEASure:P<n>:STATistics 173
:EYE:MEASure:P<n>:SHIStory 174
:EYE:MEASure:CLEar 174
:EYE:MTESt 174
:EYE:MTESt:TYPE 175
:EYE:MTESt:MASK:LOAD 175
:EYE:MTESt:OPERate 176
:EYE:MTESt:COUNt 177
:EYE:MTESt:FUNCtion:BUZZer 177
:EYE:MTESt:FUNCtion:SOF 177
5.13 FUNCtion 命令系统 179
:FUNCtion:FFTDisplay 179
:FUNCtion:GVALue 179
:FUNCtion<x> 180
:FUNCtion<x>:AVERage:NUM 180
:FUNCtion<x>:DELay:DELay 181
:FUNCtion<x>:DIFF:DX 181
:FUNCtion<x>:ENVelope:POINts 182
:FUNCtion<x>:ERES:BITS 183
:FUNCtion<x>:FFT:AUToset 183
:FUNCtion<x>:FFT:HCENter 183

:FUNCtion<x>:FFT:HSCale 184
:FUNCtion<x>:FFT:SPAN 184
:FUNCtion<x>:FFT:LOAD 185
:FUNCtion<x>:FFT:MODE 185
:FUNCtion<x>:FFT:POINts 186
:FUNCtion<x>:FFT:RESET 187
:FUNCtion<x>:FFT:RLEVel 187
:FUNCtion<x>:FFT:SCALe 188
:FUNCtion<x>:FFT:SEARch 189
:FUNCtion<x>:FFT:SEARch:EXCursion 189
:FUNCtion<x>:FFT:SEARch:MARKer<n> 190
:FUNCtion<x>:FFT:SEARch:MARKer<n>:SHOW 190
:FUNCtion<x>:FFT:SEARch:MON 191
:FUNCtion<x>:FFT:SEARch:PORDer 191
:FUNCtion<x>:FFT:SEARch:RESult 191
:FUNCtion<x>:FFT:SEARch:TABLe 192
:FUNCtion<x>:FFT:SEARch:TABLe:DELTa 193
:FUNCtion<x>:FFT:SEARch:TABLe:FREQuency 193
:FUNCtion<x>:FFT:SEARch:THReshold 194
:FUNCtion<x>:FFT:UNIT 194
:FUNCtion<x>:FFT:WINDow 195
:FUNCtion<x>:FFTPhase:UNWRap 196
:FUNCtion<x>:FFTPhase:UNWRap:THReshold 196
:FUNCtion<x>:FFTPhase:SQUelch 197
:FUNCtion<x>:FFTPhase:SQUelch:THReshold 197
:FUNCtion<x>:FILTer:TYPe 198
:FUNCtion<x>:FILTer:HFRequency 198
:FUNCtion<x>:FILTer:LFRequency 199
:FUNCtion<x>:INTegrate:GATE 199
:FUNCtion<x>:INTegrate:OFFSet 200
:FUNCtion<x>:INTErpolate:COEF 200
:FUNCtion<x>:INVert 201
:FUNCtion<x>:LABel 201

:FUNCtion<x>:LABel:TEXT 202
:FUNCtion<x>:MAXHold:SWeeps 202
:FUNCtion<x>:MINHold:SWeeps 203
:FUNCtion<x>:OPERation 203
:FUNCtion<x>:POSition 205
:FUNCtion<x>:SCALe 205
:FUNCtion<x>:SOURce<n> 206
5.14 GATE 命令系统 207
:GATE:<channel>:SWITch 207
:GATE:<channel>:X<n> 207
:GATE:<channel>:XDELta 208
:GATE:<channel>:REFerence 208
:GATE:<channel>:SYNChronization 209
5.15 HISTORy 命令系统 210
:HISTORy 210
:HISTORy:FRAMe 210
:HISTORy:INTERval 211
:HISTORy:LIST 211
:HISTORy:PLAY 212
:HISTORy:TIME 212
5.16 JITTer 命令系统 213
:JITTer 213
:JITTer:SOURce 213
:JITTer:LEVel 214
:JITTer:HYSTeresis 214
:JITTer:SIGNal 215
:JITTer:CLOCk 215
:JITTer:CLOCk:MODE 215
:JITTer:CLOCk:SRATe 216
:JITTer:CLOCk:FONCe 217
:JITTer:CLOCk:FOPLL:CUToff 217
:JITTer:CLOCk:SOPLL 217

:JITTer:CLOCk:SOPLL:JTF:CUToff 218
:JITTer:CLOCk:SOPLL:JTF:LBANdwidth 218
:JITTer:CLOCk:SOPLL:JTF:PEAKing 219
:JITTer:CLOCk:SOPLL:OJTF:CUToff 219
:JITTer:CLOCk:SOPLL:OJTF:LBANdwidth 220
:JITTer:CLOCk:SOPLL:OJTF:DAMPing 220
:JITTer:DISPlay 221
:JITTer:QVIew 221
:JITTer:MEASure 221
:JITTer:MEASure:STATistics:RESet 222
:JITTer:MEASure:P<n> 222
:JITTer:MEASure:P<n>:TYPE 222
:JITTer:MEASure:P<n>:VALue 224
:JITTer:MEASure:P<n>:STATistics 225
:JITTer:MEASure:P<n>:SHIStory 225
:JITTer:MEASure:CLEar 226
:JITTer:PMODe 226
:JITTer:PREPeats 226
:JITTer:RUN 227
5.17 MEASure 命令系统 228
:MEASure 228
:MEASure:ADVanced:CLEar 228
:MEASure:ADVanced:LINenumber 228
:MEASure:ADVanced:P<n> 229
:MEASure:ADVanced:P<n>:SOURce<x> 229
:MEASure:ADVanced:P<n>:STATistics 230
:MEASure:ADVanced:P<n>:SHIStory 231
:MEASure:ADVanced:P<n>:TYPE 231
:MEASure:ADVanced:P<n>:VALue 235
:MEASure:ADVanced:STATistics 235
:MEASure:ADVanced:STATistics:AIMLimit 236
:MEASure:ADVanced:STATistics:HISTOGram 236

:MEASure:ADVanced:STATistics:MAXCount 237
:MEASure:ADVanced:STATistics:RECord 237
:MEASure:ADVanced:STATistics:RESet 238
:MEASure:ADVanced:STYLe 238
:MEASure:ASTRategy 238
:MEASure:ASTRategy:BASE 239
:MEASure:ASTRategy:TOP 239
:MEASure:DTIMe<n>:EDGE1 240
:MEASure:DTIMe<n>:EDGE2 240
:MEASure:DTIMe<n>:SLOPe1 241
:MEASure:DTIMe<n>:SLOPe2 241
:MEASure:DTIMe<n>:THReshold1 242
:MEASure:DTIMe<n>:THReshold2 242
:MEASure:GATE 243
:MEASure:GATE:GA 243
:MEASure:GATE:GB 244
:MEASure:MODE 244
:MEASure:RDISplay 245
:MEASure:SIMPle:CLEar 245
:MEASure:SIMPle:ITEM 246
:MEASure:SIMPle:SOURce 246
:MEASure:SIMPle:VALue 247
:MEASure:THReshold:SOURce 248
:MEASure:THReshold:TYPE 248
:MEASure:THReshold:ABSolute 249
:MEASure:THReshold:PERCent 249
5.18 MEMory 命令系统 251
:MEMory<m>:HORizontal:POSition 251
:MEMory<m>:HORizontal:SCALe 251
:MEMory<m>:HORizontal:SYNC 252
:MEMory<m>:IMPort 252
:MEMory<m>:LABel 253

:MEMory<m>:LABel:TEXT 253
:MEMory<m>:SWITch 254
:MEMory<m>:VERTical:POSition 254
:MEMory<m>:VERTical:SCALe 255
5.19 METEr 命令系统 256
MMETer 256
READ 256
CONFigure 257
MEASure 262
SENSe 266
5.20 MTESt 命令系统 271
:MTESt 271
:MTESt:COUNt 271
:MTESt:FUNCtion:BUZZer 271
:MTESt:FUNCtion:COF 272
:MTESt:FUNCtion:FTH 272
:MTESt:FUNCtion:SOF 273
:MTESt:IDISplay 273
:MTESt:MASK:CREate 274
:MTESt:MASK:LOAD 274
:MTESt:OPERate 275
:MTESt:RESet 275
:MTESt:SOURce 276
:MTESt:TYPE 276
5.21 RECall 命令系统 277
:RECall:FDEFault 277
:RECall:PROJect 277
:RECall:REFerence 278
:RECall:SERase 278
:RECall:SETup 279
5.22 REF 命令系统 280
:REF<r>:LABel 280

:REF<r>:LABel:TEXT 280
:REF<r>:DATA 281
:REF<r>:DATA:SOURce 281
:REF<r>:DATA:SCALe 282
:REF<r>:DATA:POSition 282
5.23 SAVE 命令系统 283
:SAVE:BINary 283
:SAVE:CSV 284
:SAVE:DEFault 285
:SAVE:IMAGe 285
:SAVE:PROJect 286
:SAVE:MATLab 286
:SAVE:REFerence 287
:SAVE:SETup 288
:SAVE:SFILe 289
5.24 SEARch 命令系统 290
:SEARch 290
:SEARch:MODE 290
:SEARch:COUNT 291
:SEARch:EVENt 291
:SEARch:COPY 291
:SEARch:EDGE 292
:SEARch:SLOPe 294
:SEARch:PULSe 299
:SEARch:INTerval 303
:SEARch:RUNT 308
5.25 SYSTem 命令系统 314
:SYSTem:BUZZer 314
:SYSTem:CLOCk 314
:SYSTem:COMMunicate:LAN:GATeway 315
:SYSTem:COMMunicate:LAN:IPADdress 315
:SYSTem:COMMunicate:LAN:MAC 316

:SYSTem:COMMunicate:LAN:SMASk 316
:SYSTem:COMMunicate:LAN:TYPE 316
:SYSTem:COMMunicate:VNCPort 317
:SYSTem:DATE 317
:SYSTem:EDUMode 318
:SYSTem:LANGuage 319
:SYSTem:MENU 319
:SYSTem:NSTorage 320
:SYSTem:NSTorage:CONNect 320
:SYSTem:NSTorage:DISConnect 321
:SYSTem:NSTorage:STATus 321
:SYSTem:PON 321
:SYSTem:REBoot 322
:SYSTem:REMote 322
:SYSTem:SELFCal 322
:SYSTem:SHUTdown 323
:SYSTem:SSAVer 323
:SYSTem:TIME 323
:SYSTem:TOUCh 324
5.26 TIMebase 命令系统 325
:TIMebase:DELay 325
:TIMebase:REFerence 325
:TIMebase:REFerence:POSition 326
:TIMebase:SCALe 326
:TIMebase:WINDow 327
:TIMebase:WINDow:DELay 327
:TIMebase:WINDow:SCALe 328
5.27 TPPA 命令系统 329
:TPPA:DUT<a> 329
:TPPA:DUT<a>:GROup<b> 329
:TPPA:DUT<a>:GROup<b>:CONFig:WIRing 330
:TPPA:DUT<a>:GROup<b>:CONFig:VSOurce<c> 330

:TPPA:DUT<a>:GROup<b>:CONFig:ISOurce<c> 331
:TPPA:DUT<a>:GROup<b>:CONFig:EQUalifier 332
:TPPA:DUT<a>:GROup<b>:CONFig:LPFilter:CFRequency 332
:TPPA:DUT<a>:GROup<b>:MEASure:PQUality 333
:TPPA:DUT<a>:GROup<b>:MEASure:PQUality:PHASor 334
:TPPA:DUT<a>:GROup<b>:MEASure:RIPPle 334
:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic 335
:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:FREQuency 335
:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:THDStandard 336
:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:STANdard 336
:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:TABLe 337
:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:BAR 337
:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic 338
:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:FREQuency 339
:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:THDStandard 339
:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:STANdard 340
:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:TABLe 340
:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:BAR 341
:TPPA:DUT<a>:MINDuration 341
:TPPA:DUT<a>:EFFiciency 342
:TPPA:AUToset 342
5.28 TRIGger 命令系统 343
:TRIGger:ACTion:SAVE:DIRectory 343
:TRIGger:ACTion:SAVE:FILename 343
:TRIGger:ACTion:SAVE:WMODe 344
:TRIGger:ACTion:SAVE:WAVeform 344
:TRIGger:ACTion:SAVE:WAVeform:FORMat 345
:TRIGger:ACTion:SAVE:IMAGe 345
:TRIGger:ACTion:SAVE:IMAGe:FORMat 346
:TRIGger:ACTion:OPERation 346
:TRIGger:FREQuency 347
:TRIGger:MODE 347

:TRIGger:RUN 348
:TRIGger:STATus 348
:TRIGger:STOP 348
:TRIGger:TYPE 348
:TRIGger:EDGE 349
:TRIGger:SLOPe 355
:TRIGger:PULSe 363
:TRIGger:VIDeo 370
:TRIGger:WINDow 376
:TRIGger:INTerval 383
:TRIGger:DROPout 391
:TRIGger:RUNT 397
:TRIGger:PATTern 405
:TRIGger:QUALified 411
:TRIGger:DELay 416
:TRIGger:NEDGe 420
:TRIGger:SHOLd 426
:TRIGger:IIC 431
:TRIGger:SPI 438
:TRIGger:UART 447
:TRIGger:CAN 455
:TRIGger:LIN 459
:TRIGger:FLEXray（选配） 466
:TRIGger:CANFd（选配） 470
:TRIGger:IIS（选配） 476
:TRIGger:SENT（选配） 484
5.29 WAVeform 命令系统 486
:WAVeform:ACCelerate 486
:WAVeform:BYTeorder 486
:WAVeform:SOURce 487
:WAVeform:STARt 487
:WAVeform:INTerval 488

:WAVeform:POINt 488
:WAVeform:MAXPoint 489
:WAVeform:WIDTh 489
:WAVeform:PREamble 490
:WAVeform:DATA 493
:WAVeform:SEQuence 496
5.30 WGEN 命令系统 498
ARbWaVe 498
BaSic_WaVe 499
OUTPut 500
SToreList 501
SYNC 502
VOLTPRT 502
6 编程实例 503
6.1 基于 VISA 编程 503
VC++实例 503
VB 实例 511
MATLAB 实例 517
LabVIEW 实例 519
C#实例 522
6.2 基于 Socket 编程 539
Python 实例 539
C 实例 542
6.3 常用命令实例 544
模拟通道波形重构实例 544
数字通道波形重构实例 549
数学函数 FFT 波形重构实例 552
分段采集波形重构实例 555
截屏实例 564

1 版本声明

本章节将介绍已发布的编程手册版本中对编程指令的修改内容。

1.1 CN11I 版本

此版本修正文档内指令大小写错误、含占位符的指令描述合并减少冗余描述。指令修改详见下表。

命令	描述
:BODE:SWEep:COUNt	新增，查询波特图已完成的扫描次数
:BODE:DATA:SAVE	新增，保存波特图配置和数据
:BODE:SWEep:MODE	更改，新增参数支持设置连续扫描次数

1.2 CN11H 版本

此版本新增部分分析功能指令、自动保存指令、新增数学运算的指令，下面显示更详细描述。

命令	描述
:ACQuire:XY:DISPlay	新增，设置 XY 分屏显示
:CHANnel<n>:ZSCale	新增，设置 Zoom 的垂直档位
:CHANnel<n>:ZOFFset	新增，设置 Zoom 的垂直偏移
:CURSor:CLEar	新增，多光标清除当前所有光标
:CURSor:RESet	新增，多光标恢复默认设置
:DECode:BUS<n>:CAN:SIGNal	新增，设置解码 CAN 信号类型
:DECode:BUS<n>:CAN:CANLSource	新增，设置解码 CAN_L 信源
:DECode:BUS<n>:CAN:CANLThreshold	新增，设置解码 CAN_L 阈值
:FUNCtion<x>:DELay:DELay	新增，设置延迟算子延迟值
:FUNCtion<x>:ENVelope:POINts	新增，设置包络算子最大点数
:FUNCtion<x>:FFT:UNIT	更改，参数新增{DEGRee | RADian | SECond}
:FUNCtion<x>:FFT:WINDow	更改，参数新增{BHARris | GAUSsian}
:FUNCtion<x>:FFTPhase 命令	新增，FFT 相位谱子命令系统
:FUNCtion<x>:OPERation	更改，参数新增{ENVelope | TAN | ATAN | FFTPhase | DELay | EXPRess}
:MEASure:ADVanced:STATistics:RECord	新增，设置测量统计中记录测量结果数目
:MEMory<m>:HORizontal:SYNC	更改，参数新增{MGRoup1|MGRoup2}

命令	描述
:RECall:PROJect	新增，调出测量工程
:SAVE:BINary	更改，参数新增{ALL}
:SAVE:CSV	更改，参数新增{ALL}
:SAVE:MATLab	更改，参数新增{ALL}
:SAVE:PROJect	新增，保存测量工程
:SAVE:SFILe	新增，设置多通道是否保存成单文件
:TRIGger:ACTion 命令	新增，自动保存子命令系统
:WAVeform:ACCelerate	新增，设置 WAV 加速功能开关
:BODE 命令	新增，波特图命令系统
:EYE 命令	新增，眼图命令系统
:GATE 命令	新增，分析门限命令系统
:JITTer 命令	新增，抖动分析命令系统
:TPPA 命令	新增，三相电指令系统，暂不支持获取测量结果

1.3 CN11G 版本

此版本修正了各命令的缩写格式、部分命令格式、示例中的多余空格及中英文符号错误。更新了实例章节中波形重构实例代码。下面显示了新增命令和更改命令的更详细描述。

命令	描述
:DISPlay:AXIS:POSition	新增，设置垂直轴标签的显示位置
:DISPlay:HIDemenu	新增，隐藏右侧菜单
:FUNCtion<x>:FFT:SEARch:MARKer<n>	新增，设置 FFT 标记的频率
:FUNCtion<x>:FFT:SEARch:MARKer<n>:SHOW	新增，开启 FFT 指定标记
:FUNCtion<x>:FFT:SEARch:MON	新增，设置 FFT 标记谐波或峰值
:FUNCtion<x>:FFT:SEARch:PORDer	新增，设置 FFT 峰值排序
:FUNCtion<x>:FFT:SEARch:TABLe	新增，开启 FFT 搜索表格
:FUNCtion<x>:FFT:SEARch:TABLe:DELTa	新增，开启 FFT 搜索表格增量显示
:FUNCtion<x>:FFT:SEARch:TABLe:FREQuency	新增，开启 FFT 搜索表格频率显示
:MEASure:ADVanced:P<n>:SHIStory	新增，返回高级测量项的历史测量数据
:MEASure:DTIMe<n>:EDGE1	新增，设置测量项 DeltaTime 开始沿
:MEASure:DTIMe<n>:EDGE2	新增，设置测量项 DeltaTime 截止沿

命令	描述
:MEASure:DTIMe<n>:SLOPe1	新增，设置测量项 DeltaTime 开始沿类型
:MEASure:DTIMe<n>:SLOPe2	新增，设置测量项 DeltaTime 截止沿类型
:MEASure:DTIMe<n>:THReshold1	新增，设置测量项 DeltaTime 开始沿阈值
:MEASure:DTIMe<n>:THReshold2	新增，设置测量项 DeltaTime 截止沿阈值
:MEASure:RDISplay	新增，设置测量结果样式
多光标命令	新增，多光标指令系统，区别于单组光标
COUNter:CURRent	新增，返回计数器测量值
:COUNter:STATistics:VALue	新增，返回计数器统计信息
:DECode:LIST<n>:RESult	新增，返回解码列表结果，区别于 :DECode:BUS<n>:RESult
:WAVeform:BYTeorder	新增，设置传输波形数据字的字节顺序
:DECode:BUS<n>:PROTocol :TRIGger:TYPE	更改，参数 ARINC429 更新为 A429
:MEASure:SIMPle:ITEM :MEASure:ADVanced:P<n>:TYPE	更改，新增参数 ULOWer、DTIMe1-4
:MEASure:ADVanced:P<n>:SOURce1 :MEASure:SIMPle:SOURce :MEASure:THReshold:SOURce	更改，延迟测量新增参数 F<x>；新增参数 ZF<x>、ZM<m>；
:SAVE:IMAGe	更改，新增可选参数{MON|MOFF}

1.4 CN11F 版本

SDS 系列编程指导的首个中文版本，和英文版本使用相同的版本号。该文档适用于的大部分机型，有关详细信息，请参阅下一章“机型支持”。

2 机型支持

本文档中列出的命令可用于如下表所示的 SIGLENT 数字示波器系列。

下表内容根据机型初始发布的日期排列。

型号	支持的版本号
SDS5000X	0.9.0 及更高版本
SDS2000X Plus	1.3.5R3 及更高版本
SDS6000 Pro/SDS6000A	1.1.7.0 及更高版本
SHS800X/SHS1000X	1.1.9 及更高版本
SDS2000X HD	1.2.0.2 及更高版本
SDS6000L	1.0.1.0 及更高版本
SDS1000X HD	1.1.0.2 及更高版本
SDS7000A	1.0.7.0 及更高版本
SDS800X HD	1.1.3.1 及更高版本
SDS3000X HD	1.0.3.0 及更高版本
SDS5000X HD	1.1.4.5 及更高版本
Module Scope(MS032/052/102A)	1.1.3.1 及更高版本

3 编程概述

用户可以通过使用示波器的 USB 和 LAN 端口，并结合 NI-VISA 和程序语言，远程控制示波器。基于 LAN 端口，SDS 支持 VXI-11、Sockets 和 Telnet 通信协议。本节介绍了如何建立 SDS 系列示波器和计算机之间的通信，同时介绍如何远程控制示波器。

3.1 建立通信

NI-VISA 的安装

在编程之前，请确保正确安装 NI-VISA 软件的最新版本。

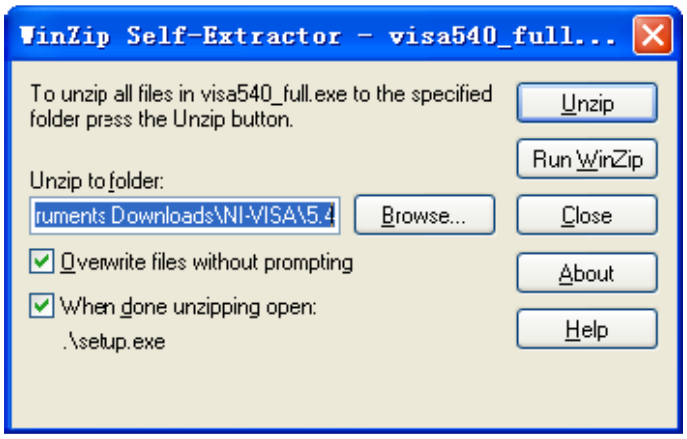
NI-VISA 是用于计算机与设备之间通信的通信库。NI 软件有两种有效 VISA 安装包：完整版和运行引擎版（Run-Time Engine）。完整版包括 NI 设备驱动和 NI MAX 工具，其中 NI MAX 是用于控制设备的用户界面。虽然驱动和 NI MAX 都很有用，但是它们不用于远程控制。运行引擎版（Run-Time Engine）是一个比完整版更小的文件，它主要用于远程控制。

你可以在 NI 官网（NI-VISA Download - NI）上下载最新的 NI-VISA 运行引擎或完整版。它们的安装步骤基本相同。

按照下列步骤安装 NI-VISA（示例使用 NI-VISA5.4 完整版，更新版本的安装步骤类似）：

a. 下载合适版本的 NI-VISA（推荐运行引擎版）

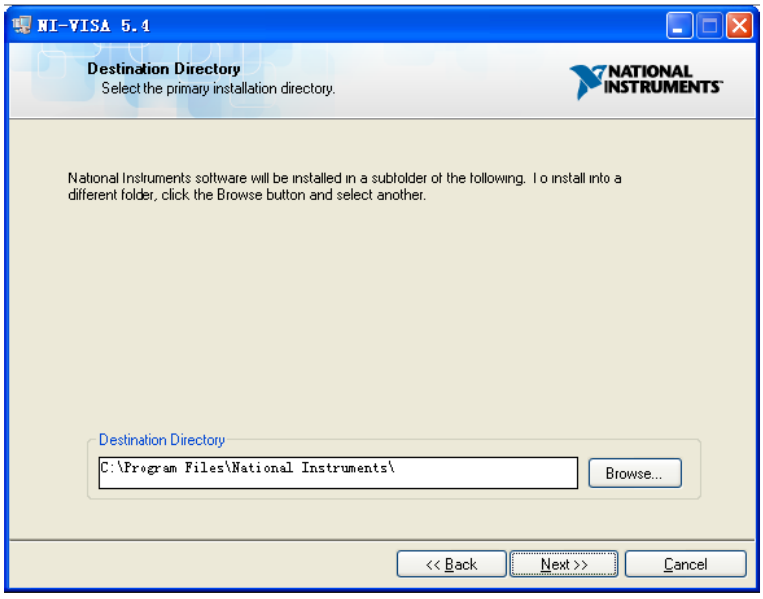
b. 双击 visa540_full.exe，弹出对话框如下：



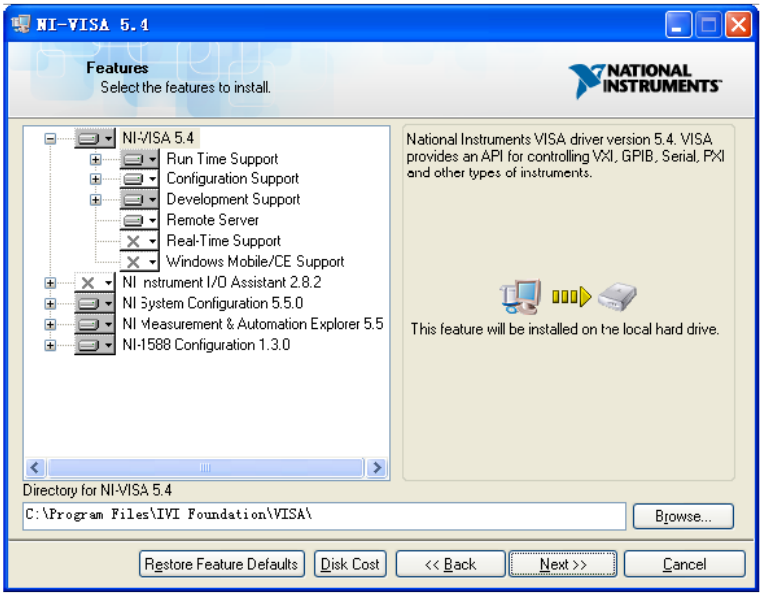
c. 点击 Unzip 解压文件，当解压完成后，安装程序将自动执行。若你的计算机需要安装.NET Framework4，则在安装过程会自动安装.NET Framework4。



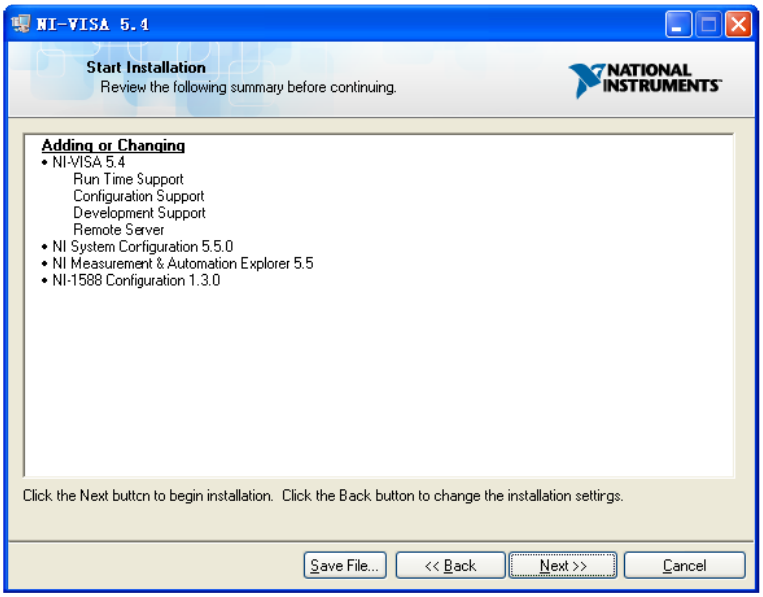
d. NI-VISA 安装对话框如上图所示，点击 Next 开始安装过程。



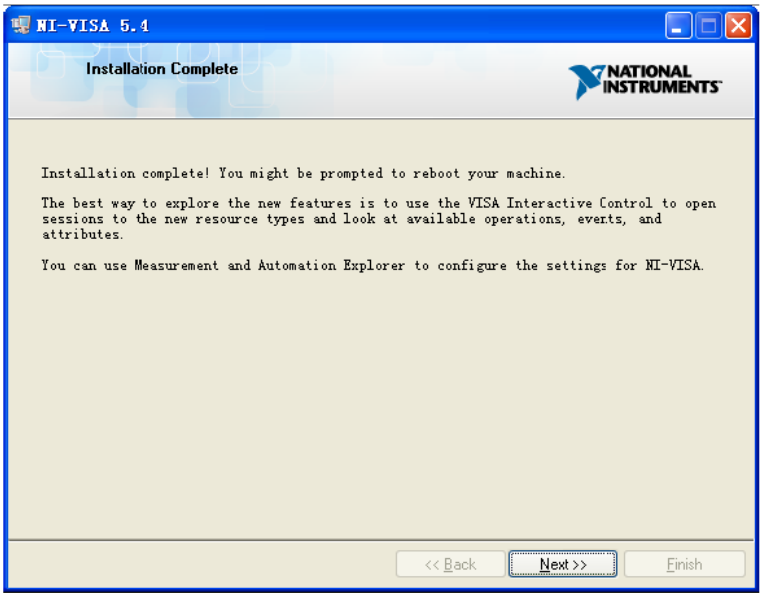
e. 设置安装路径，默认路径为“C:\Program Files\National Instruments\”。你也可以修改安装路径。点击 Next，对话框如下图所示。



f. 点击 Next 两次，在许可协议对话框下，选择“I accept the above 2 License Agreement(s).”并点击 Next，对话框如下图显示：



g. 点击 Next 开始安装：

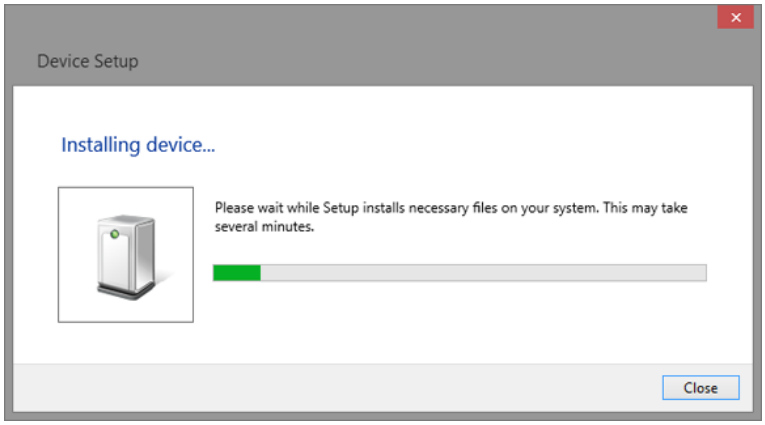


h. 安装完成后，重启电脑。

连接仪器

根据具体机型，示波器能够通过 USB 或 LAN 接口连接计算机。

使用 USB 线将示波器的 USB Device 端口和计算机的 USB Host 端口连接起来。假设你的计算机已经启动，打开示波器后，桌面将弹出“设备安装”界面，并自动安装设备驱动，如下图所示：



等待安装完成，然后进行下一步。

3.2 远程控制的实现

用户自定义程序

用户可通过计算机发送 SCPI 命令实现编程和控制示波器。相关内容，请查阅“编程实例”中的介绍。

通过 NI-MAX 发送 SCPI 命令

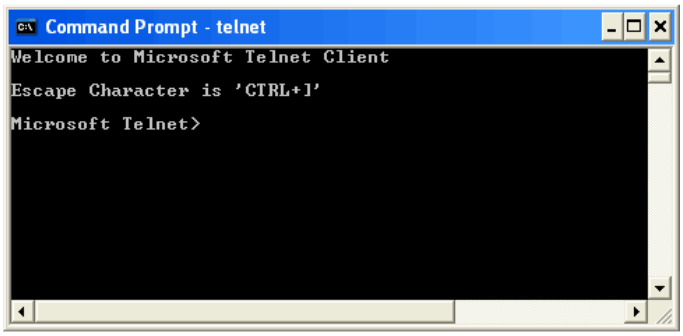
NI-MAX 是由 NI 公司创建和维护的程序。它为 VXI、LAN、USB、GPIB 和串行通信提供基础的远程控制接口。用户可以通过 NI-MAX 发送 SCPI 命令远程控制示波器。

通过 Telnet 发送 SCPI 命令

Telnet 提供一种通过 LAN 端口与示波器通信的方式。Telnet 协议支持从计算机向示波器发送 SCPI 命令，该方式类似于通过 USB 与示波器通信。发送和接受信息是交互的：一次只能发送一个命令。Windows 操作系统使用命令提示符样式接口作为 Telnet 客户端。

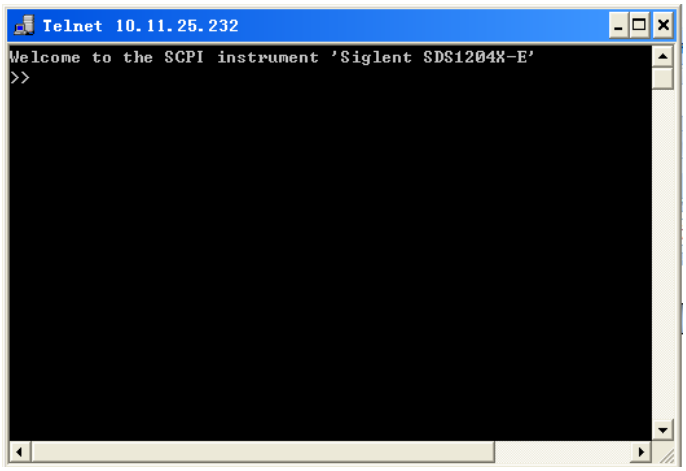
步骤如下：

1. 在计算机桌面，点击开始>所有程序>附件>命令提示符
2. 在命令提示符窗口，输入 Telnet
3. 按下 Enter 键。将弹出 Telnet 显示窗

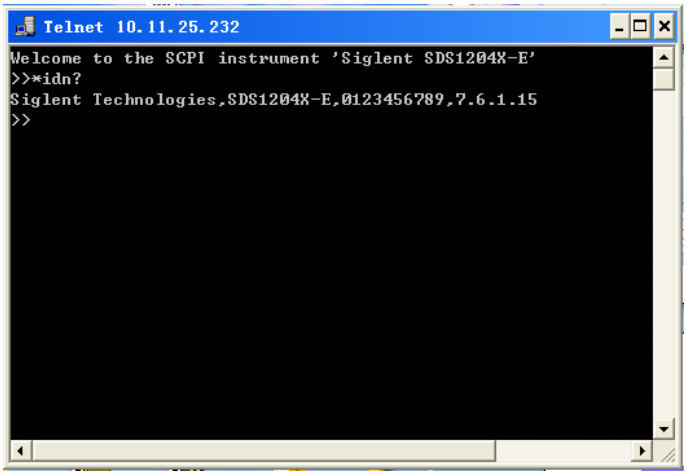


4. 在 Telnet 命令行，键入：*open XXX.XXX.XXX.XXX 5024*

其中 XXX.XXX.XXX.XXX 是指设备的 IP 地址；5024 是端口。通信成功后你会看到和下面类似的响应内容：



5. 在“>>”提示符后，输入 SCPI 命令例如*IDN?。该命令将返回公司名、机器型号、序列号和固件版本号。



6. 同时按下 Ctrl+] 键将退出 SCPI 会话。
7. 通过在提示符出键入 quit 或关闭 Telnet 窗口来关闭设备的连接并退出 Telnet。

通过 Socket 发送 SCPI 命令

Socket 接口可以在不安装其他库的情况下通过 LAN 端口控制示波器。它可以减少编程的复杂度。

SOCKET 地址	IP 地址+端口号
IP 地址	SDS IP 地址
端口号	5025

请查看“基于 Socket 编程”章节了解更多信息。

4 SCPI 语言简介

4.1 命令和查询格式

本节列出并描述仪器识别的远程控制命令和查询语句。所有命令和查询都可以在本地或远程状态下执行。在“命令格式”中，本文给出了该命令的长格式和短格式。

SCPI 命令为树状层次结构，包括多个子系统，每个子系统由一个根关键字和一个或数个层次关键字构成。命令通常以冒号“:”开始。关键字之间用冒号“:”分隔，关键字后面跟随可选的参数设置。命令行后面添加问号“?”，表示对此功能进行查询。命令关键字和第一个参数之间以空格分开，多个参数的，参数之间用逗号“,”分隔。

例如：

:ACQuire:TYPE <type>

:ACQuire:TYPE?

ACQuire 是命令的根关键字，TYPE 是第二级关键字。命令行以冒号“:”开始，同时用冒号“:”将各级关键字分开，<type>表示可设置的参数。问号“?”表示查询。命令关键字:ACQuire:TYPE 和参数<type>之间用空格分开。

在一些带参数的命令中，通常用逗号“,”分隔多个参数，例如：

:SAVE:CSV <path>,<source>,<state>

4.2 命令缩写

每个命令都有一个长格式和一个短格式。请注意，在本文档中短格式部分用大写字符显示，其余部分用小写字符显示。所有命令对大小写不敏感，你可以全部采用大写或小写。但是如果要缩写，必须完整且仅仅输入命令格式中的大写字母。例如：

:CHANnel1:SCALe?

可缩写成：

:CHAN1:SCAL?

4.3 命令符号

下面是用在命令中的符号：

<> 三角括号

包含用于占位的字符，三角括号中的参数必须用一个有效值来替换。

:= 定义为

冒号后面跟着等号，它用于分隔占位符和占位符的描述，占位符的描述是指用于替换命令中占位符的参数类型和范围。例如，<A>:= <B>表示在任何包含<A>的语句中，<A>都可以用<B>替换

{} 大括号

列出了可供选择的参数，参数项之间通常用竖线（|）分隔，表示“或”。使用命令时，只能选择其中一个元素。例如，{ON|OFF} 表示只能选择 ON 或 OFF，而不能同时选择两者。

[] 方括号

方括号中的内容是可省略的。

… 省略号

表示其左侧和右侧可以重复多次

4.4 参数类型

枚举型（Enumeration）

参数取值为所列举的值。将这些参数输入为不带引号的文本单词。与关键字一样，枚举参数遵循相同的约定，其中大写部分是必需的，小写部分是可选的。例如：

:ACQuire:TYPE <type>

其中<type>可设置为：{NORMal|PEAK|AVERage|ERES}。

整型（Integer）

除非另有说明，参数在有效值范围内可以是任意整数（NR1 格式）。例如：

:DISPlay:GRATicule <value>

其中<value>可设置为：[0,100]之间的整数。

浮点型（float）

参数在有效值范围内可以是指定精度的实数。该命令接受小数（NR2 格式）和科学计数（NR3 格式）格式的参数输入。例如：

:CHANnel1:SKEW <value>

其中<value>可设置为：[-1.00E-07, 1.00E-07] 之间的 NR3 格式实数。

二进制数据（Binary Block）

定长块响应数据允许任何类型的设备相关数据作为一系列 8 位二进制数据字节通过系统接口传输。这对于发送大量数据或 8 位扩展 ASCII 码特别有用。以#字符开头，后跟一个非零数字，表示十进制整数中的位数；在非零数字之后是十进制整数，表示正在发送的 8 位数据字节的数量，紧接着是实际数据。例如：传输 5000 字节的数据

#45000<5000 bytes of data><NL>

4 表示位数，5000 是要传输的字节数，<5000 bytes of data>是实际数据。

ASCII 字符串（ASCII String）

参数为用双引号（"）括起来的 ASCII 字符组合。以下是带引号的字符示例："This is A quoted string"。本文档将该参数表示为<qstring>。使用带引号的字符串时，请遵守以下规则：

1、 使用双引号字符打开和关闭字符串。例如："this is a valid string"。

2、 遵循前面的规则，就可以在字符串中混合使用单引号。例如："this is an 'acceptable' string"。

3、 不能在字符串中重复包含双引号。例如："here is a"" mark"。它将被识别为"here is a "。

4、 字符串可以有大小写字符，但是最终将会被自动转换为大写。

5、 嵌入带引号字符串中的回车或换行将被识别为字符串。

以下是一些无效字符串：

- "Invalid string argument'（引号的类型不同）
- "here is a "mark""（双引号内有重复的双引号）

5 命令与查询

5.1 IEEE488.2 通用命令

IEEE 标准定义的通用命令适用于查询设备的基本信息和执行基本操作。这些命令通常以“*”开头以及命令的关键字长度为 3 个字符。

*IDN

描述	*IDN?是用于查询设备的 ID。返回由四个不同的字段组成，提供有关设备商、设备型号、序列号和固件版本的信息。
命令格式	*IDN?
返回格式	<设备商>,<机型>,<序列号>,<固件版本> <设备商>:= “Siglent Technologies” <机型>:= 机器型号 <序列号>:= 14 位 ASCII 字符组成的编号，此序列号标注产品的唯一性 <固件版本>:= 与软件版本信息相关的编号
示例	查询版本信息： **IDN?* 返回值： *Siglent Technologies,SDS6204 H12 Pro,SDS62DDD6R0323,14.29.08.1.4.8.3* (每个版本可能不同)

*OPC

描述	当示波器完成所有待处理命令后，向输出缓冲区返回 1。*OPC? 查询可以在重叠的命令之间使用，以确保一个命令在下一个命令开始之前完成。
命令格式	*OPC?
返回格式	1
示例	查询状态： **OPC?* 返回值： *1*

*RST

描述	*RST 重置示波器到默认设置。此命令等同于前面板上的 Default 按键。
命令格式	*RST
示例	重置示波器: **RST*
关联命令	:RECall:FDEFault :RECall:SETup :SAVE:DEFault :SAVE:SETup

5.2 Root(:)命令系统

:Root 命令子系统用于查询示波器的基本信息或执行基本操作。

:AUToset

描述	根据信号的幅度、频率等信息自动设置示波器的触发、垂直、水平参数，快捷地显示信号。不建议在低频事件（<100Hz）中使用 Autoset。
命令格式	:AUToset
示例	自动设置示波器参数： *:AUToset* *AUT*

:PRINt

描述	返回指定格式的截屏数据。
命令格式	:PRINt? <type>[,<format>]
参数说明	<type>:= {BMP|PNG} ● BMP：bmp 格式数据 ● PNG：png 格式数据 <format>:= {NORMal|INVerted} ● NORMal：正常颜色，默认值 ● INVerted：反色
返回格式	<bin> <bin>:= 指定格式的二进制数据
示例	见“截屏实例”

:FORMat:DATA

描述	设置或查询 NR3 格式返回值的有效位数。当前默认有效位数为 3 位。
命令格式	:FORMat:DATA <option>[,<digit>] :FORMat:DATA?
参数说明	<option>:= {SINGle|DOUBle|CUSTom}

	● SINGle：单精度类型，7 位有效位数 ● DOUBle：双精度类型，14 位有效位数 ● CUSTom：自定义有效位数，需指定<digit> <digit>:= 整型 NR1 格式，范围［1,64］
返回格式	CUSTom,<digit>
示例	设置命令返回值精度为 5 位数： *:FORMat:DATA CUSTom,5* *FORM:DATA CUST,5* 查询命令返回值精度： *FORM:DATA?* 返回值： *CUSTom,5*

5.3 ACQuire 命令系统

:ACQUIRE 命令子系统控制采集波形的方式。这些命令可以设置或查询采集和存储的参数。

:ACQuire:AMODe

描述	设置或查询示波器的采集模式。此命令可以提供高速波形捕获速率，以帮助捕获信号异常。
命令格式	:ACQuire:AMODe <mode> :ACQuire:AMODe?
参数说明	<mode>:= {FAST|SLOW} ● FAST：快采。快速采集波形 ● SLOW：慢采。慢速采集波形
返回格式	{FAST|SLOW}
示例	设置快采模式： *:ACQuire:AMODe FAST* *ACQ:AMOD FAST* 查询当前采集模式： *ACQ:AMOD?* 返回值： *FAST*

:ACQuire:CSWeep

描述	设置清除示波器的扫描并重新开始采集，此设置等价于前面板上的 Clear Sweeps 按钮。
命令格式	:ACQuire:CSWeep
示例	清除采集扫描： *:ACQuire:CSWeep* *ACQ:CSW*

:ACQuire:INTerpolation

描述	设置或查询示波器的内插方式。
命令格式	:ACQuire:INTerpolation <state> :ACQuire:INTerpolation?
参数说明	<state>:= {ON|OFF} ● ON（Sinc）：Sinc 内插方式 ● OFF（X）：线性内插方式
返回格式	{ON|OFF}
示例	设置 Sinc 内插方式： *:ACQuire:INTerpolation ON* *ACQ:INT ON* 查询当前内插方式： *ACQ:INT?* 返回值： *ON*

:ACQuire:MMANagement

描述	设置或查询示波器的存储模式。
命令格式	:ACQuire:MMANagement <mem_mode> :ACQuire:MMANagement?
参数说明	<mem_mode>:= {AUTO|FSRate|FMDepth} ● AUTO：自动。保持最大采样率，并根据时基自动设置存储深度和采样率。 ● FSRate：固定采样率。保持指定的采样速率，并根据时基自动设置存储深度。 ● FMDepth：固定存储深度。保持指定的存储深度，并根据时基自动设置采样速率。
返回格式	{AUTO|FSRate|FMDepth}
示例	设置固定存储深度存储模式： *:ACQuire:MMANagement FMDepth* *ACQ:MMAN FMD* 查询当前存储模式： *ACQ:MMAN?* 返回值： *FMDepth*

:ACQuire:MODE

描述	设置或查询示波器的模式。
命令格式	:ACQuire:MODE <mode_type> :ACQuire:MODE?
参数说明	<mode_type>:= {YT|XY|ROLL} ● YT：该模式绘制幅度(Y)与时间(T)的关系。 ● XY：该模式绘制通道 X 与通道 Y 的关系，通常称为利萨如曲线。 ● ROLL：该模式为滚动模式，绘制幅度(Y)与时间(T)的关系，但从显示屏的右侧开始写入波形。这类似于“带状图”记录，非常适合每秒钟发生几次的慢速事件。
返回格式	{YT|XY|ROLL}
示例	设置示波器的 YT 模式： *:ACQuire:MODE YT* *ACQ:MODE YT* 问询示波器的当前模式： *ACQ:MODE?* 返回值： *YT*

:ACQuire:MDEPth

描述	设置或查询示波器的最大存储深度。
命令格式	:ACQuire:MDEPth <memory_size> :ACQuire:MDEPth?
参数说明	<memory_size>:= 存储深度。因机型而异，有关详细信息，请参见下表：

机型	存储深度
SDS5000X	单通道 {250k|1.25M|2.5M|12.5M|25M|125M|250M}
	双通道 {125k|625k|1.25M|6.25M|12.5M|62.5M|125M}
SDS2000X Plus	单通道 {20k|200k|2M|20M|200M}
	双通道 {10k|100k|1M|10M|100M}
SDS6000 Pro SDS6000A	1G 机型单通道 {1.25k|5k|25k|50k|250k|500k|2.5M|5M|12.5M|125M|250M}

	1G 机型双通道 {1.25k|2.5k|12.5k|25k|125k|250k|1.25M|2.5M|12.5M|62.5M|125M}
	2G 机型 {2.5k|5k|25k|50k|250k|500k|2.5M|5M|12.5M|25M|50M|125M|250M|250M|500M}
SDS6000L	{2.5k|5k|25k|50k|250k|500k|2.5M|5M|12.5M|25M|50M|125M|250M|250M|500M}
SHS800X SHS1000X	单通道 {12k|120k|1.2M|12M}
	双通道 {6k|60k|600k|6M}
SDS2000X HD	单通道 {20k|200k|2M|20M|200M}
	双通道 {10k|100k|1M|10M|100M}
SDS800X HD SDS1000X HD	单通道 {10k|100k|1M|10M|100M}
	双通道 {10k|100k|1M|10M|50M}
	四通道 {10k|100k|1M|10M|25M}
SDS7000A	{1k|5k|10k|50k|100k|500k|1M|5M|10M|50M|100M|500M|1G}
SDS3000X HD	单通道 {2k|10k|20k|100k|200k|1M|10M|20M|100M|200M|400M}
	双通道 {2k|10k|20k|100k|200k|1M|10M|20M|100M|200M}
	多通道 {2k|10k|20k|100k|200k|1M|10M|20M|100M}
SDS5000X HD	单通道 {2.5k|25k|250k|2.5M|25M|62.5M|250M|500M|1G|2.5G}
	双通道 {2.5k|25k|250k|2.5M|25M|62.5M|250M|500M|1G}
	多通道 {1.25k|12.5k|125k|1.25M|12.5M|25M|62.5M|125M|250M|500M}

注意:

- 关于单通道和双通道模式的定义，请参考用户手册。

	● 打开数字通道或将采集类型设置为平均/ERES 或将采集模式设置为滚动，将限制存储深度。
返回格式	<memory_size>
示例	设置 SDS5000X 系列示波器的存储深度为 125 Mpts： *:ACQuire:MDEPth 125M* *ACQ:MDEP 125M* 查询存储深度： *ACQ:MDEP?* 返回值： *125M*
关联命令	:ACQuire:MODE :ACQuire:TYPE

:ACQuire:NUMACq

描述	查询已采集到的波形帧数。当任何影响波形采集的操作发生时，如水平或垂直参数改变，该值重置为零。
命令格式	:ACQuire:NUMACq?
返回格式	整型 NR1 格式
示例	查询已采集到的 350 帧波形： *:ACQuire:NUMACq?* *ACQ:NUMAC?* 返回值： *350*

:ACQuire:POINts

描述	查询屏幕上当前波形的采样点数。
命令格式	:ACQuire:POINts?
返回格式	浮点型 NR3 格式
示例	查询返回当前采样点数： *:ACQuire:POINts?* *ACQ:POIN?* 返回值： *1.25E+08*

:ACQuire:RESolution

描述	设置或查询 SDS2000X Plus 示波器的 ADC 分辨率。
命令格式	:ACQuire:RESolution <bit> :ACQuire:RESolution?
参数说明	<bit>:= {8Bits|10Bits}
返回格式	{8Bits|10Bits}
示例	设置 ADC 分辨率为 10Bits: *:ACQuire:RESolution 10Bits* *ACQ:RES 10B* 查询当前 ADC 分辨率: *ACQ:RES?* 返回值: *10Bits*

:ACQuire:SEQuence

描述	设置或查询分段采集开关状态。
命令格式	:ACQuire:SEQuence <state> :ACQuire:SEQuence?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置分段采集开启: *:ACQuire:SEQuence ON* *ACQ:SEQ ON* 查询当前分段采集开关状态: *ACQ:SEQ?* 返回值: *ON*
关联命令	:ACQuire:SEQuence:COUNt

:ACQuire:SEQuence:COUNt

描述	设置或查询分段采集段数。最大分段数可能受到示波器存储深度的限制。
命令格式	:ACQuire:SEQuence:COUNt <count> :ACQuire:SEQuence:COUNt?

参数说明	<count>:= 数值，整型 NR1 格式，该值的范围因机型和当前时基而异，详情请参见用户手册。
返回格式	整型 NR1 格式
示例	设置分段采集段数为 5 段： *:ACQuire:SEQuence:COUNt 5* *ACQ:SEQ:COUN 5* 查询当前分段采集段数： *ACQ:SEQ:COUN?* 返回值： *5*
关联命令	:ACQuire:SEQuence

:ACQuire:SRATe

描述	设置或查询固定采样率模式下的采样速率。
命令格式	:ACQuire:SRATe <rate> :ACQuire:SRATe?
参数说明	<rate>:= 数值，浮点型 NR3 格式，如果设定值大于可设定值，它将自动匹配可设定值。
返回格式	浮点型 NR3 格式
示例	设置采样率为 5Gsa/s： *:ACQuire:SRATe 5.00E9* *ACQ:SRAT 5.00E9* 查询当前采样率： *ACQ:SRAT?* 返回值： *5.00E+09*

:ACQuire:TYPE

描述	设置或查询示波器的采集方式。
命令格式	:ACQuire:TYPE <type> :ACQuire:TYPE?
参数说明	<type>:= {NORMal|PEAK|AVERage[,<times>]|ERES[,<bits>]} ● NORMal：普通。普通模式 ● PEAK：峰值检测。峰值检测模式

	● AVERage：平均。设置平均次数需指定<times> ● ERES：增强分辨率。设置增强位需指定<bits> <times>:= {4|16|32|64|128|256|512|1024|2048|4096|8192} <bits>:= {0.5|1.0|1.5|2.0|2.5|3.0|3.5|4.0} **注意**：分段采集开启时，平均模式和增强分辨率模式不可用。 （:ACQuire:SEQuence ON）
返回格式	{NORMal|PEAK|AVERage[,<times>]|ERES[,<bits>]}
示例	设置采集方式为平均，平均次数为 16： *:ACQuire:TYPE AVERage,16* *ACQ:TYPE AVER,16* 查询当前采集方式： *ACQ:TYPE?* 返回值： *AVERage,16*

:ACQuire:XY:DISPlay

描述	设置或查询 XY 模式下的分屏显示。 **注意**：仅 SDS800X HD、SDS1000X HD 支持
命令格式	:ACQuire:XY:DISPlay <type> :ACQuire:XY:DISPlay?
参数说明	<type>:= {EXCLusive|SPLit}
返回格式	{EXCLusive|SPLit}
示例	设置 XY 分屏： *:ACQuire:XY:DISPlay SPLit* *ACQ:XY:DISP SPL* 查询当前 XY 显示类型： *ACQ:XY:DISP?* 返回值： *SPLit*

5.4 BODE 命令系统

:BODE 命令子系统控制波特图功能的设置与查询。

:BODE

描述	设置或查询波特图开关状态。
命令格式	:BODE <state> :BODE?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置波特图开启： *:BODE ON* *BODE ON* 查询当前波特图开关状态： *BODE?* 返回值： *ON*

:BODE:SOURce:INPut

描述	设置或查询波特图的 DUT 输入信源。
命令格式	:BODE:SOURce:INPut <source> :BODE:SOURce:INPut?
参数说明	<source>:= {C<n>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式
返回格式	{C<n>}
示例	设置 C1 作为波特图 DUT 输入源： *:BODE:SOURce:INPut C1* *BODE:SOUR:INP C1* 查询波特图 DUT 输入源： *BODE:SOUR:INP?* 返回值： *C1*
关联命令	:BODE:SOURce:OUTPut<x>

:BODE:SOURce:OUTPut<x>

描述	设置或查询波特图 DUT 输出的信源。
命令格式	:BODE:SOURce:OUTPut<x> <source> :BODE:SOURce:OUTPut<x>?
参数说明	<x>:= {1|2|3}，DUT 输出通道序号 <source>:= {C<n>|NONE} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● NONE:= 无
返回格式	{C<n>|NONE}
示例	设置 C2 作为波特图 DUT 输出 1 的信源： *:BODE:SOURce:OUTPut1 C2* *BODE:SOUR:OUTP1 C2* 查询波特图 DUT 输出 1 的信源： *BODE:SOUR:OUTP1?* 返回值： *C2*
关联命令	:BODE:SOURce:INPut

:BODE:SOURce:GAIN

描述	设置或查询波特图的通道增益模式。
命令格式	:BODE:SOURce:GAIN <mode> :BODE:SOURce:GAIN?
参数说明	<mode>:= {AUTO|HOLD}
返回格式	{AUTO|HOLD}
示例	设置特图的通道增益模式为保持： *:BODE:SOURce:GAIN HOLD* *BODE:SOUR:GAIN HOLD* 查询当前特图的通道增益模式： *BODE:SOUR:GAIN?* 返回值： *HOLD*

:BODE:SWEep:TYPE

描述	设置或查询波特图的扫描类型。
命令格式	:BODE:SWEep:TYPE <type> :BODE:SWEep:TYPE?
参数说明	<type>:= {SIMPle|SEGMent[,<file>]} ● SIMPle：恒定幅度 ● SEGMent：可变幅度。设置可变幅度配置文件需指定<file> <file>:= {FILEA|FILEB|FILEC|FILED}
返回格式	{SIMPle|SEGMent[,{FILEA|FILEB|FILEC|FILED}]}
示例	设置波特图扫描类型为可变幅度，配置文件选择 D： *:BODE:SWEep:TYPE SEGMent,FILED* *BODE:SWE:TYPE SEGM,FILED* 查询当前波特图扫描类型： *BODE:SWEep:TYPE?* 返回值： *SEGMent,FILED*

:BODE:SWEep:MODE

描述	设置或查询波特图扫描/测量模式。
命令格式	:BODE:SWEep:MODE <mode>[,<count>] :BODE:SWEep:MODE?
参数说明	<mode>:= {CONTinue|SINGle} ● CONTinue：连续扫描。设置扫描次数需指定<count> ● SINGle：单次扫描 <count>:= 整型 NR1 格式
返回格式	{CONTinue[,NR1]|SINGle}
示例	设置波特图扫描/测量模式为单次： *:BODE:SWEep:MODE SINGle* *BODE:SWE:MODE SING* 查询当前波特图扫描/测量模式： *BODE:SWE:MODE?* 返回值： *SINGle*

:BODE:SWEep:COUNt

描述	查询波特图已完成的扫描次数。
命令格式	:BODE:SWEep:COUNt?
返回格式	整型 NR1 格式
示例	例如波特图连续扫描了 20 次，发送命令问询： *:BODE:SWEep:COUNt?* *BODE:SWE:COUN?* 返回值： *20*

:BODE:SWEep:POINt

描述	设置或查询波特图的扫描点数。
命令格式	:BODE:SWEep:POINt <num> :BODE:SWEep:POINt?
参数说明	<num>:= 整型 NR1 格式
返回格式	整型 NR1 格式
示例	设置波特图的扫描点数为 100： *:BODE:SWEep:POINt 100* *BODE:SWE:POIN 100* 查询当前波特图的扫描点数： *BODE:SWE:POIN?* 返回值： *100*

:BODE:SWEep:SEGMent

描述	设置波特图可变幅度下扫描结点的频率和幅值。
命令格式	:BODE:SWEep:SEGMent <freq1>,<ampl1>,...,<freqn>,<ampln>
参数说明	<freq1>,<ampl1>,...,<freqn>,<ampln> <freq1>:= 第一个结点的频率值，浮点型 NR3 格式 <ampl1>:= 第一个结点的幅值，浮点型 NR3 格式 ... <freqn>:= 第 n 个结点的频率值，浮点型 NR3 格式 <ampln>:= 第 n 个结点的幅值，浮点型 NR3 格式

	注意：设置参数必须成对输入
示例	设置波特图可变幅度下 4 个结点，频点和幅值分别为[5kHz,2V]、[10kHz,4V]、[15kHz,3V]、[20kHz,4V]： *:BODE:SWEep:SEGMent 5e+3,2,10e+3,4,15e+3,3,20e+3,4* *BODE:SWE:SEGM 5e+3,2,10e+3,4,15e+3,3,20e+3,4*
关联命令	:BODE:SWEep:TYPE

:BODE:FREQuency:MODE

描述	设置或查询波特图频率模式。
命令格式	:BODE:FREQuency:MODE <mode> :BODE:FREQuency:MODE?
参数说明	<mode>:= {LINear|DECade}
返回格式	{LINear|DECade}
示例	设置波特图频率模式为对数： *:BODE:FREQuency:MODE DECade* *BODE:FREQ:MODE DEC* 查询当前波特图频率模式： *BODE:FREQ:MODE?* 返回值： *DECade*

:BODE:FREQuency:STARt

描述	设置或查询波特图对数模式下的起始频率。
命令格式	:BODE:FREQuency:STARt <value> :BODE:FREQuency:STARt?
参数说明	<value>:= 浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置波特图起始频率为 1MHz： *:BODE:FREQuency:STARt 1.00E+06* *BODE:FREQ:STAR 1.00E+06* 查询波特图起始频率： *BODE:FREQ:STAR?* 返回值：

	1.00E+06
关联命令	:BODE:FREQuency:MODE :BODE:FREQuency:STOP

:BODE:FREQuency:STOP

描述	设置或查询波特图对数模式下的终止频率。
命令格式	:BODE:FREQuency:STOP <value> :BODE:FREQuency:STOP?
参数说明	<value>:= 浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置波特图终止频率为 100MHz: *:BODE:FREQuency:STOP 1.00E+08* *BODE:FREQ:STOP 1.00E+08* 查询波特图终止频率: *BODE:FREQ:STOP?* 返回值: *1.00E+08*
关联命令	:BODE:FREQuency:MODE :BODE:FREQuency:STARt

:BODE:FREQuency:CENTer

描述	设置或查询波特图线性模式下的中心频率。
命令格式	:BODE:FREQuency:CENTer <value> :BODE:FREQuency:CENTer?
参数说明	<value>:= 浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置波特图中心频率为 10MHz: *:BODE:FREQuency:CENTer 1.00E+07* *BODE:FREQ:CENT 1.00E+07* 查询波特图中心频率: *BODE:FREQ:CENT?* 返回值: *1.00E+07*

关联命令	:BODE:FREQuency:MODE :BODE:FREQuency:SPAN

:BODE:FREQuency:SPAN

描述	设置或查询波特图线性模式下的扫宽。
命令格式	:BODE:FREQuency:SPAN <value> :BODE:FREQuency:SPAN?
参数说明	<value>:= 浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置波特图扫宽为 10MHz： *:BODE:FREQuency:SPAN 1.00E+07* *BODE:FREQ:SPAN 1.00E+07* 查询波特图扫宽： *BODE:FREQ:SPAN?* 返回值： *1.00E+07*
关联命令	:BODE:FREQuency:MODE :BODE:FREQuency:CENTer

:BODE:WGEN:AMPLitude

描述	设置或查询波特图信号源输出幅度。
命令格式	:BODE:WGEN:AMPLitude <value> :BODE:WGEN:AMPLitude?
参数说明	<value>:= 浮点型 NR3 格式，数值范围[0,24]
返回格式	浮点型 NR3 格式
示例	设置波特图信号源幅度为 4V： *:BODE:WGEN:AMPLitude 4* *BODE:WGEN:AMPL 4* 查询波特图信号源幅度： *BODE:WGEN:AMPL?* 返回值： *4.00E+00*

:BODE:WGEN:OFFSet

描述	设置或查询波特图信号源偏移量。
命令格式	:BODE:WGEN:OFFSet <value> :BODE:WGEN:OFFSet?
参数说明	<value>:= 浮点型 NR3 格式，数值范围[幅度/2-24,24-幅度/2]
返回格式	浮点型 NR3 格式
示例	设置波特图信号源偏移量为 2V： *:BODE:WGEN:OFFSet 2* *BODE:WGEN:OFFS 2* 查询波特图信号源偏移量： *BODE:WGEN:OFFS?* 返回值： *2.00E+00*

:BODE:WGEN:REFLevel

描述	设置或查询波特图信号源参考电平，仅在单位为 Arb dB 时有效。
命令格式	:BODE:WGEN:REFLevel <value> :BODE:WGEN:REFLevel?
参数说明	<value>:= 浮点型 NR3 格式，数值范围[0.001,100]
返回格式	浮点型 NR3 格式
示例	设置波特图信号源参考电平为 2V： *:BODE:WGEN:REFLevel 2* *BODE:WGEN:REFL 2* 查询波特图信号源参考电平： *BODE:WGEN:REFL?* 返回值： *2.00E+00*
关联命令	:BODE:WGEN:UNIT

:BODE:WGEN:UNIT

描述	设置或查询波特图信号源幅度单位。
命令格式	:BODE:WGEN:UNIT <unit> :BODE:WGEN:UNIT?

参数说明	<unit>:= {VPP|VRMS|DBV|DBU|DBM|ADB}
返回格式	{VPP|VRMS|DBV|DBU|DBM|ADB}
示例	设置波特图信号源单位为 Arb dB： *:BODE:WGEN:UNIT ADB* *BODE:WGEN:UNIT ADB* 查询波特图信号源参考电平： *BODE:WGEN:UNIT?* 返回值： *ADB*
关联命令	:BODE:WGEN:AMPLitude :BODE:WGEN:OFFSet :BODE:WGEN:REFLevel

:BODE:WGEN:LOAD

描述	设置或查询波特图信号源负载。
命令格式	:BODE:WGEN:LOAD <load> :BODE:WGEN:LOAD?
参数说明	<load>:= 整型 NR1 格式，该值范围为[50,1000000] **注意**：100 kΩ 以上时，表示为高阻 Hi-Z
返回格式	整型 NR1 格式
示例	设置波特图信号源负载为 75Ω： *:BODE:WGEN:LOAD 75* *BODE: WGEN:LOAD 75* 查询当前波特图信号源负载： *BODE: WGEN:LOAD?* 返回值： *75*

:BODE:WGEN:INTerface

描述	设置或查询波特图信号源接口类型。
命令格式	:BODE:WGEN:INTerface <type> :BODE:WGEN:INTerface?
参数说明	<type>:= {INNer|USB|LAN}

返回格式	{INNer|USB|LAN}
示例	设置波特图信号源连接类型为 USB： *:BODE:WGEN:INTerface USB* *BODE:WGEN:INT USB* 查询波特图信号源连接类型： *BODE:WGEN:INT?* 返回值： *USB*

:BODE:WGEN:IPADdress

描述	设置或查询波特图信号源 LAN 接口的 IP 地址。
命令格式	:BODE:WGEN:IPADdress <ip> :BODE:WGEN:IPADdress?
参数说明	<ip>:= 带引号的 ASCII 文本字符串
返回格式	带引号的 ASCII 文本字符串
示例	设置波特图信号源 IP 地址为”10.12.255.2”： *:BODE:WGEN:IPADdress "10.12.255.2"* *BODE:WGEN:IPAD "10.12.255.2"* 查询波特图信号源 IP 地址： *BODE:WGEN:IPAD?* 返回值： *"10.12.255.2"*
关联命令	:BODE:WGEN:INTerface

:BODE:OPERate

描述	设置或查询波特图运行开关状态。
命令格式	:BODE:OPERate <state> :BODE:OPERate?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置波特图运行开启： *:BODE:OPERate ON* *BODE:OPER ON* 查询当前波特图运行状态：

	BODE:OPER? 返回值： *ON*

:BODE:DATA

描述	查询波特图数据。
命令格式	:BODE:DATA?
返回格式	<freq1>,<ampl11>,<pha11>[,<ampl12>,<pha12>,<ampl13>,<pha13>]\n <freq2>,<ampl21>,<pha21>[,<ampl22>,<pha22>,<ampl23>,<pha23>]\n ...\n <freqn>,<ampln1>,<phan1> [<ampln2>,<phan2>,<ampln3>,<phan3>]\n <freq>:= 频率点，浮点型 <ampl>:= DUT 输出幅值，浮点型。如 DUT 未开启，则不返回数值 <pha>:= DUT 输出相位，浮点型。如 DUT 未开启，则不返回数值
示例	DUT1 开启，DUT2、DUT3 未开启，查询当前波特图数据： *BODE:DATA?* 返回值： *49999250,6.07444187e-06,49.4160138* *49999333.3,6.63886343e-06,144.049428*
关联命令	:BODE:OPERate

:BODE:DATA:SAVE

描述	保存波特图的配置和数据。
命令格式	:BODE:DATA:SAVE <path>
参数说明	<path>:= 带引号的 ASCII 文本字符串，包含文件存储路径和带后“.csv”缀的文件名。可指定存储路径类型如下表： **存储路径类型** | **示例** 内部 | “local/SIGLENT/test.csv” 网络存储器 | “net_storage/test.csv” U 盘 | “U-disk0/test.csv” **注意**： ● 保存到内部时，存储路径必须包含 local。 ● 保存到外部时，如未指定存储路径类型，将默认存储到 U-disk0。

	文件格式不是由文件扩展名自动确定，您需要选择一个扩展名与所选文件格式一致的文件。
示例	将波特图的配置和数据存储至 U 盘的 SIGLENT 文件夹中，文件名为 bode.csv： *:BODE:DATA:SAVE "U-disk0/SIGLENT/bode.csv"* *BODE:DATA:SAVE "U-disk0/SIGLENT/bode.csv"*
关联命令	:BODE:DATA

:BODE:MEASure:P<n>

描述	设置波特图指定测量项的信源和测量类型。 查询波特图指定测量项的测量值。
命令格式	:BODE:MEASure:P<n> <source>,<type> :BODE:MEASure:P<n>?
参数说明	<n>:= 测量项序号，整型 NR1 格式，取值范围[1,6] <source>:= {C<m>} ● C<m>:= 模拟通道，<m>:= 序号，整型 NR1 格式 <type>:= {UFRequency|LFRequency|BW|GM|PM|CF|NONE} 波特图测量参数表

参数	描述
UFRequency（上限截止频率）	增益曲线中相对最高增益点衰减-3 dB 的频率且较高频率的频率值。
LFRequency（下限截止频率）	增益曲线中相对最高增益点衰减-3 dB 的频率且较低频率的频率值。
BW（带宽）	上限截止频率减去下限截止频率的频率值。
GM（增益裕度）	当相位达到 0°时增益距离 0 dB 的增益值。
PM（相位裕度）	当增益降为 0 dB 时相位距离 0°的相位值。
CF（穿越频率）	增益穿过 0 dB 的频率值。

返回格式	浮点型 NR3 格式
示例	设置波特图第一个测量项为 C2 的上限截止频率： *:BODE:MEASure:P1 C2,UFRequency* *BODE:MEAS:P1 C2,UFR* 查询波特图第一个测量项的测量值： *BODE:MEAS:P1?* 返回值： *5.00E+07*

5.5 CHANnel 命令系统

:CHANnel 命令子系统控制模拟通道的参数。通道可独立调节，用于垂直偏移、探头、耦合、带宽限制、反相等更多功能。命令中可对指定的模拟通道进行设置。

:CHANnel:REFerence

描述	设置或查询扩展策略中的垂直参考策略。
命令格式	:CHANnel:REFerence <type> :CHANnel:REFerence?
参数说明	<type>:= {OFFSet|POSition} ● OFFSet：偏移固定。当垂直档位改变时，垂直偏移保持固定。随着垂直档位的改变，波形围绕屏幕的主 X 轴扩展/收缩。 ● POSition：位置固定。当垂直档位改变时，垂直偏移保持固定在网格位置。随着垂直档位的改变，波形围绕屏幕上的固定位置扩展/收缩。
返回格式	{OFFSet|POSition}
示例	设置垂直参考为偏移固定： *:CHANnel:REFerence OFFSet* *CHAN:REF OFFS* 查询当前垂直参考策略： *CHAN:REF?* 返回值： *OFFSet*

:CHANnel<n>:BWLimit

描述	设置或查询指定输入通道的带宽限制。
命令格式	:CHANnel<n>:BWLimit <bwlimit> :CHANnel<n>:BWLimit?
参数说明	<n>:= 通道序号，整型 NR1 格式 <bwlimit>:= {20M|200M|FULL} ● 20M：20M 带宽限制 ● 200M：200M 带宽限制 ● FULL：全带宽
返回格式	{20M|200M|FULL}

示例	设置 C1 的带宽限制为 20M： *:CHANnel1:BWLimit 20M* *:CHAN1:BWL 20M* 查询 C1 的带宽限制： *CHAN1:BWL?* 返回值： *20M*

:CHANnel<n>:COUPling

描述	设置或查询指定输入通道的耦合方式。
命令格式	:CHANnel<n>:COUPling < coupling_mode> :CHANnel<n>:COUPling?
参数说明	<n>:= 通道序号，整型 NR1 格式 <coupling_mode>:= {AC|DC|GND} ● AC：交流耦合 ● DC：直流耦合 ● GND：接地
返回格式	{AC|DC|GND}
示例	设置 C1 的通道耦合为 AC： *:CHANnel1:COUPling AC* *CHAN1:COUP AC* 查询 C1 的通道耦合： *CHAN1:COUP?* 返回值： *AC*

:CHANnel<n>:IMPedance

描述	设置或查询指定输入通道的阻抗。
命令格式	:CHANnel<n>:IMPedance < impedance> :CHANnel<n>:IMPedance?
参数说明	<n>:= 通道序号，整型 NR1 格式 <impedance>:= {ONEMeg|FIFTy} ● ONEMeg：1MΩ ● FIFTy：50Ω

	注意：当阻抗设置为 FIFTy(50Ω)，垂直档位最大值为 1V。
返回格式	{ONEMeg|FIFTy}
示例	设置 C1 的通道阻抗为 50Ω： *:CHANnel1:IMPedance FIFTy* *CHAN1:IMP FIFT* 查询 C1 的通道阻抗： *CHAN1:IMP?* 返回值： *FIFTy*
关联命令	:CHANnel<n>:SCALe

:CHANnel<n>:INVert

描述	设置或查询指定输入通道的反相状态。
命令格式	:CHANnel<n>:INVert <state> :CHANnel<n>:INVert?
参数说明	<n>:= 通道序号，整型 NR1 格式 <state>:= {ON|OFF} ● ON：打开指定的通道反相 ● OFF：关闭指定的通道反相
返回格式	{ON|OFF}
示例	设置 C1 的反相状态为 ON： *:CHANnel1:INVert ON* *CHAN1:INV ON* 查询 C1 的反相状态： *CHAN1:INV?* 返回值： *ON*

:CHANnel<n>:LABel

描述	设置或查询指定输入通道的标签状态。
命令格式	:CHANnel<n>:LABel < state> :CHANnel<n>:LABel?
参数说明	<n>:= 通道序号，整型 NR1 格式

	<state>:= {ON|OFF} ● ON：打开指定的通道标签 ● OFF：关闭指定的通道标签
返回格式	{ON|OFF}
示例	设置 C1 的标签状态为 ON： *:CHANnel1:LABel ON* *CHAN1:LAB ON* 查询 C1 的标签状态： *CHAN1:LAB?* 返回值： *ON*
关联命令	:CHANnel<n>:LABel:TEXT

:CHANnel<n>:LABel:TEXT

描述	设置或查询指定输入通道的标签字符。
命令格式	:CHANnel<n>:LABel:TEXT <qstring> :CHANnel<n>:LABel:TEXT?
参数说明	<n>:= 通道序号，整型 NR1 格式 <qstring>:= 带引号的 ASCII 文本字符串，字符串的长度限制为 20
返回格式	带引号的 ASCII 文本字符串
示例	设置 C1 的标签内容为“VOUT”： *:CHANnel1:LABel:TEXT "VOUT"* *CHAN1:LAB:TEXT "VOUT"* 查询 C1 的标签内容： *CHAN1:LAB:TEXT?* 返回值： *"VOUT"*
关联命令	:CHANnel<n>:LABel

:CHANnel<n>:OFFSet

描述	设置或查询指定输入通道的垂直偏移。取值范围和垂直档位相关。
命令格式	:CHANnel<n>:OFFSet <offset_value> :CHANnel<n>:OFFSet?

参数说明	<n>:= 通道序号，整型 NR1 格式 <offset_value>:= 垂直偏移数值，浮点型 NR3 格式，数值范围与垂直档位有关
返回格式	浮点型 NR3 格式
示例	设置 C2 的垂直偏移为 -3.8V： *:CHANnel2:OFFSet -3.8E+00* *CHAN2:OFFS -3.8E+00* 查询 C2 的垂直偏移： *CHAN2:OFFS?* 返回值： *-3.8E+00*
关联命令	:CHANnel<n>:SCALe

:CHANnel<n>:PROBe

描述	设置或查询指定输入通道的探头衰减系数。
命令格式	:CHANnel<n>:PROBe <attenuation>[,<value>] :CHANnel<n>:PROBe?
参数说明	<n>:= 通道序号，整型 NR1 格式 <attenuation>:= {DEFault|VALue} ● DEFault：默认系数 1X ● VALue：自定义系数，需指定<value> <value>:= 探头衰减率，浮点型 NR3 格式，其范围为 [1E-6, 1E6]
返回格式	浮点型 NR3 格式
示例	设置 C1 的探头衰减系数为 100： *:CHANnel1:PROBe VALue,1.00E+02* *CHAN1:PROB VAL,1.00E+02* 查询 C1 的探头衰减系数： *CHAN1:PROB?* 返回值： *-1E+02*
关联命令	:CHANnel<n>:SCALe :CHANnel<n>:OFFSet

:CHANnel<n>:SCALe

描述	设置或查询指定输入通道的垂直档位。取值范围和探头衰减系数成正比。
命令格式	:CHANnel<n>:SCALe <scale> :CHANnel<n>:SCALe?
参数说明	<n>:= 通道序号，整型 NR1 格式 <scale>:= 垂直档位值，浮点型 NR3 格式 **注意**：垂直档位和机型相关，不同机型档位范围不一致。
返回格式	浮点型 NR3 格式
示例	设置 C1 的垂直档位为 50mV/div： *:CHANnel1:SCALe 5.00E-02* *CHAN1:SCAL 5.00E-02* 查询 C1 的垂直档位： *CHAN1:SCAL?* 返回值： *5E-02*
关联命令	:CHANnel<n>:PROBe

:CHANnel<n>:SKEW

描述	设置或查询指定输入通道的时滞。
命令格式	:CHANnel<n>:SKEW <skew_value> :CHANnel<n>:SKEW?
参数说明	<n>:= 通道序号，整型 NR1 格式 <skew_value>:= 通道时滞，浮点型 NR3 格式，其取值范围为 [-1.00E-07, 1.00E-07]
返回格式	浮点型 NR3 格式
示例	设置 C1 的通道时滞为 1.52 ns： *:CHANnel1:SKEW 1.52E-09* *CHAN1:SKEW 1.52E-09* 查询 C1 的通道时滞： *CHAN1:SKEW?* 返回值： *1.52E-09*

:CHANnel<n>:SWITch

描述	设置或查询指定输入通道的开关状态。
命令格式	:CHANnel<n>:SWITch <state> :CHANnel<n>:SWITch?
参数说明	<n>:= 通道序号，整型 NR1 格式 <state>:= {OFF|ON}
返回格式	{OFF|ON}
示例	设置 C2 的开关状态为 ON： *:CHANnel2:SWITch ON* *CHAN2:SWIT ON* 查询 C2 的开关状态： *CHAN2:SWIT?* 返回值： *ON*

:CHANnel<n>:UNIT

描述	设置或查询指定输入通道的垂直单位。
命令格式	:CHANnel<n>:UNIT <unit> :CHANnel<n>:UNIT?
参数说明	<n>:= 通道序号，整型 NR1 格式 <unit>:= {V|A} **注意**：该指令会更改指定通道的单位，会影响测量结果、光标、通道档位和触发电平等功能的单位。
返回格式	{V|A}
示例	设置 C2 的垂直单位为 A： *:CHANnel2:UNIT A* *CHAN2:UNIT A* 查询 C2 的垂直单位： *CHAN2:UNIT?* 返回值： *A*

:CHANnel<n>:VISible

描述	设置或查询指定输入通道的波形显示状态。
命令格式	:CHANnel<n>:VISible <display_state> :CHANnel<n>:VISible?
参数说明	<n>:= 通道序号，整型 NR1 格式 <display_state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置 C2 的波形显示状态为 ON： *:CHANnel2:VISible ON* *CHAN2:VIS ON* 查询 C2 的垂直偏移： *CHAN2:VIS?* 返回值： *ON*

:CHANnel<n>:ZSCale

描述	设置或查询指定 Zoom 通道的垂直档位。取值范围最大值为对应模拟通道的垂直档位，最小值和机型相关，不同机型不一致。
命令格式	:CHANnel<n>:ZSCale <scale> :CHANnel<n>:ZSCale?
参数说明	<n>:= 通道序号，整型 NR1 格式 <scale>:= 垂直档位值，浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置 Z1 的垂直档位为 50mV/div： *:CHANnel1:ZSCale 5.00E-02* *CHAN1:ZSC 5.00E-02* 查询 C1 的垂直档位： *CHAN1:ZSC?* 返回值： *5E-02*
关联命令	:CHANnel<n>:SCALe

:CHANnel<n>:ZOFFset

描述	设置或查询指定 Zoom 通道的垂直偏移。取值范围由对应模拟通道和 Zoom 的垂直档位共同决定。
命令格式	:CHANnel<n>:ZOFFset <offset_value> :CHANnel<n>:ZOFFset?
参数说明	<n>:= 通道序号，整型 NR1 格式 <offset_value>:= 垂直偏移数值，浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	假设 C1 垂直档位为 1V/div，Z1 的垂直档位为 50mV/div，以下命令将 Z1 的垂直偏移设置为 100mV： *:CHANnel1:ZOFFset 1.00E-01* *CHAN1:ZOFF 1.00E-01* 查询 C2 的垂直偏移： *CHAN1:ZOFF?* 返回值： *1.00E-01*
关联命令	:CHANnel<n>:ZSCale

5.6 COUNter 命令系统

:COUNter 命令子系统控制计数器功能，可测量当前信号的频率、周期，并对当前信号计数。仅对模拟通道输入信号的相关参数进行测量。

:COUNter

描述	设置或查询计数器开关状态。
命令格式	:COUNter <state> :COUNter?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置计数器开启： *:COUNter ON* *COUN ON* 查询当前计数器开关状态： *COUN?* 返回值： *ON*

:COUNter:CURRent

描述	查询计数器的当前值。
命令格式	:COUNter:CURRent?
返回格式	浮点型 NR3 格式 **注意**：计数器测量精度具有 7 位。默认返回 3 位，如需更高精度值，请先使用指令设置有效位数:FORMat:DATA
示例	查询当前计数器电平： *COUN:CURR?* 返回值： *1.00E+3*

:COUNter:LEVel

描述	设置或查询计数电平。
命令格式	:COUNter:LEVel <value> :COUNter:LEVel?
参数说明	<value>:=计数电平值，浮点型 NR3 格式。因机型而异，取值范围请参考下表： **机型** / **计数电平取值范围** SDS7000A SDS5000X HD / [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L / [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD SDS3000X HD / [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置计数器电平为 0.5V： *:COUNter:LEVel 5.00E-1* *COUN:LEV 5.00E-1* 查询当前计数器电平： *COUN:LEV?* 返回值： *5.00E-1*

:COUNter:MODE

描述	设置或查询计数器的类型。
命令格式	:COUNter:MODE <type> :COUNter:MODE?
参数说明	<type>:= {FREQuency|PERiod|TOTalizer} ● FREQuency：一段时间内的频率平均值 ● PERiod：一段时间内的频率平均值的倒数 ● TOTalizer：累计计数的值

返回格式	{FREQuency|PERiod|TOTalizer}
示例	设置计数器模式为频率： *:COUNter:MODE FREQuency* *COUN:MODE FREQ* 查询当前计数器类型： *COUN:MODE?* 返回值： *FREQ*

:COUNter:SOURce

描述	设置或查询指定计数器通道的源。
命令格式	:COUNter:SOURce <source> :COUNter:SOURce?
参数说明	<source>:= {C<n>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式
返回格式	{C<n>}
示例	设置 C1 作为计数器的源： *:COUNter:SOURce C1* *COUN:SOUR C1* 查询计数器的源： *COUN:SOUR?* 返回值： *C1*

:COUNter:STATistics

描述	设置或查询计数器统计功能的开关状态，此功能仅在计数器模式为频率和周期时有效。
命令格式	:COUNter:STATistics <state> :COUNter:STATistics?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置计数器统计功能开启： *:COUNter:STATistics ON*

	COUN:STAT ON 查询当前计数器统计功能的开关状态： *COUN:STAT?* 返回值： *ON*

:COUNter:STATistics:RESet

描述	重置计数器统计，仅在计数器模式为周期和频率下有效。
命令格式	:COUNter:STATistics:RESet
示例	设置计数器开启： *:COUNter:STATistics:RESet* *COUN:STAT:RES*
关联命令	:COUNter:STATistics

:COUNter:STATistics:VALue

描述	查询当前计数器的统计信息，此功能仅在计数器模式为频率和周期时有效。
命令格式	:COUNter:STATistics:VALue?
返回格式	<current>,<mean>,<min>,<max>,<stdev>,<count> <current>:= 统计当前值，浮点型 NR3 格式。 <mean>:= 统计平均值，浮点型 NR3 格式。 <min>:= 统计最小值，浮点型 NR3 格式。 <max>:= 统计最大值，浮点型 NR3 格式。 <stdev>:= 计标准差值，浮点型 NR3 格式。 <count>:= 统计次数，整型 NR1 格式。 **注意**： ● 当计数器统计状态处于 OFF 时，此指令返回“OFF” ● 计数器测量精度具有 7 位。默认返回 3 位，如需更高精度值，请先使用指令设置有效位数:FORMat:DATA
示例	当前计数器模式为“频率”时，查询统计信息： *COUN:STAT:VAL?* 返回值： *1.00E+03,1.00E+03,1.00E+03,1.00E+03,1.52E-02,312*

:COUNter:TOTalizer:GATE

描述	设置或查询计数器统计门限的开关状态。
命令格式	:COUNter:TOTalizer:GATE <state> :COUNter:TOTalizer:GATE?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置计数器统计门限功能开启： *:COUNter:TOTalizer:GATE ON* *COUN:TOT:GATE ON* 查询当前计数器统计门限的开关状态： *COUN:TOT:GATE?* 返回值： *ON*

:COUNter:TOTalizer:GATE:LEVel

描述	设置或查询计数器统计门限电平。
命令格式	:COUNter:TOTalizer:GATE:LEVel <value> :COUNter:TOTalizer:GATE:LEVel?
参数说明	<value>:= 计数器统计门限电平值，浮点型 NR3 格式。因机型而异，取值范围请参考下表： **机型** | **计数器门限电平取值范围** SDS7000A SDS5000X HD | [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L | [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD SDS3000X HD | [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式

示例	设置计数器电平为 0.5V： *:COUNter:TOTalizer:GATE:LEVel 5.00E-1* *COUN:TOT:GATE:LEV 5.00E-1* 查询当前计数器电平： *COUN:TOT:GATE:LEV?* 返回值： *5.00E-1*

:COUNter:TOTalizer:GATE:SLOPe

描述	设置或查询计数器门限源的斜率/极性。当门限类型为电平设置极性，类型为边沿则是斜率。
命令格式	:COUNter:TOTalizer:GATE:SLOPe <slope> :COUNter:TOTalizer:GATE:SLOPe?
参数说明	<slope>:= {RISing|FALLing} ● RISing：门限源斜率为上升沿/门限源极性为正极性 ● FALLing：门限源斜率为下降沿/门限源极性为负极性
返回格式	{RISing|FALLing}
示例	设置计数器门限为边沿时，其源的斜率为上升沿： *:COUNter:TOTalizer:GATE:SLOPe RISing* *COUN:TOT:GATE:SLOP RIS* 查询当前计数器门限斜率： *COUN:TOT:GATE:SLOP?* 返回值： *RISing*

:COUNter:TOTalizer:GATE:TYPE

描述	设置或查询计数门限类型。
命令格式	:COUNter:TOTalizer:GATE:TYPE <style> :COUNter:TOTalizer:GATE:TYPE?
参数说明	<type>:= {LEVel|AEDGe} ● LEVel：计数门限类型为电平 ● AEDGe：计数门限类型为边沿之后
返回格式	{LEVel|AEDGe}

示例	设置计数门限类型为电平： *:COUNter:TOTalizer:GATE:TYPE LEVel* *COUN:TOT:GATE:TYPE LEV* 查询当前计数器门限类型： *COUN:TOT:GATE:TYPE?* 返回值： *LEVel*

:COUNter:TOTalizer:RESet

描述	重置计数器累加功能的结果。
命令格式	:COUNter:TOTalizer:RESet
示例	重置计数器累加功能。 *:COUNter:TOTalizer:RESet* *COUN:TOT:RES*

:COUNter:TOTalizer:SLOPe

描述	设置或查询计数源的斜率。
命令格式	:COUNter:TOTalizer:SLOPe <slope> :COUNter:TOTalizer:SLOPe?
参数说明	<slope>:= {RISing|FALLing} ● RISing：计数源斜率为上升沿 ● FALLing：计数源斜率为下降沿
返回格式	{RISing|FALLing}
示例	设置计数源斜率为上升沿： *:COUNter:TOTalizer:SLOPe RISing* *COUN:TOT:SLOP RIS* 查询计数源的斜率： *COUN:TOT:SLOP?* 返回值： *RISing*

5.7 CURSor 命令系统

:CURSor 命令子系统控制光标测量功能，可进行手动光标、追踪光标和测量光标设置。Siglent 示波器支持两种形式的光标：单组光标和多光标。两种形式光标的命令系统有所差异，请根据光标形式使用正确的指令。

通用命令

单组光标、多光表的通用功能包含：光标功能开关、光标样式、光标参考策略，可使用本节命令设置查询。

:CURSor

描述	设置或查询光标的开关状态。
命令格式	:CURSor <state> :CURSor?
参数说明	<state>:= {OFF|ON}
返回格式	{OFF|ON}
示例	设置光标的开关状态为 ON： *:CURSor ON* *CURS ON* 查询光标的开关状态： *CURS?* 返回值： *ON*

:CURSor:TAGStyle

描述	设置或查询光标样式。
命令格式	:CURSor:TAGStyle <style> :CURSor:TAGStyle?
参数说明	<type>:= {FIXed|FOLLowing} ● FIXed：固定光标样式 ● FOLLowing：跟随光标样式
返回格式	{FIXed|FOLLowing}
示例	设置光标样式为固定：

	:CURSor:TAGStyle FIXed *CURS:TAGS FIXed* 查询当前光标样式： *CURS:TAGS?* 返回值： *FIXed*

:CURSor:XREFerence

描述	设置或查询 X 光标参考的类型。
命令格式	:CURSor:XREFerence <type> :CURSor:XREFerence?
参数说明	<type>:= {DELay|POSition} ● DELay：延时固定。在水平档位变化时，X 光标的值保持不变 ● POSition：位置固定。在水平档位变化时，X 光标按屏幕上固定网格的位置保持不变
返回格式	<type>:= {DELay|POSition}
示例	设置 X 光标的水平参考位置为延时固定： *:CURSor:XREFerence DELay* *CURS:XREF DELay* 查询光标模式： *CURS:XREF?* 返回值： *DELay*

:CURSor:YREFerence

描述	设置或查询 Y 光标参考的类型。
命令格式	:CURSor:YREFerence <type> :CURSor:YREFerence?
参数说明	<type>:= {OFFSet|POSition} ● OFFSet：偏移固定。垂直档位变化时，Y 光标的值保持不变 ● POSition：位置固定。在垂直档位变化时，Y 光标按屏幕上固定网格的位置保持不变
返回格式	<type>:= {OFFSet|POSition}
示例	设置 Y 光标的水平参考位置为偏移固定：

	:CURSor:YREFerence OFFSet *CURS:YREF OFFSet* 查询光标模式： *CURS:YREF?* 返回值： *OFFSet*

单组光标命令

单组光标仅支持 4 根光标线：X1，X2，Y1，Y2。各光标模式下，通过本节命令设置查询光标参数。

:CURSor:IXDelta

描述	查询返回光标 1/(X2-X1)当前值。
命令格式	:CURSor:IXDelta?
返回格式	浮点型 NR3 格式
示例	查询光标 1/(X2-X1)的当前值： *CURS:IXD?* 返回值： *3.8E-05*
关联命令	:CURSor:X<n> :CURSor:XDELta

:CURSor:MITem

描述	当光标模式为测量光标时，设置或查询测量项。
命令格式	:CURSor:MITem <type>,<source1>[,<source2>] :CURSor:MITem?
参数说明	<type>:= 参数同测量类型，详见“表 5- 2 测量参数表”。 <source1>:= {C<n>|Z<n>|F<x>|M<m>|D<d>|ZD<d>|REF<r>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● Z<n>:= 缩放波形，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式，取值范围 [1,4] ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式，取值范围 [0,15] ● ZD<d>:= 缩放数字通道，<d>:= 序号，整型 NR1 格式，取值范围 [0,15] ● REF<r>:= 参考波形，<r>:= {A|B|C|D}，参考波形序号

	<source2>:= {C<n>}。当高级测量项为通道延迟，需要指定第二个源 **注意**：只要当 Zoom 处于开启状态，选择 Z<n>作为源才有效
返回格式	<type>,<source1>[,<source2>]
示例	当高级测量打开时，设置光标测量项为 PKPK(C2)： *:CURSor:MITem PKPK,C2* *CURS:MIT PKPK,C2* 查询当前光标测量项： *CURS:MIT?* 返回值： *PKPK,C2*

:CURSor:MODE

描述	设置或查询光标的模式，当模式为手动时需要设置显示类型。
命令格式	:CURSor:MODE <type> :CURSor:MODE?
参数说明	<type>:= {TRACk|MANual[,<mode>]|MEASure} ● MANual：手动光标。此模式下需指定<mode>，可选择光标类型为：水平、垂直、水平+垂直。 ● TRACk：追踪光标。光标类型自动被设置为“水平+垂直”。此时只需要调节水平光标，垂直光标将自动附着到与水平光标交汇的信源波形上。 ● MEASure：测量光标。当在参数测量中对指定的测量项开启测量光标时，光标自动标示出测量项的测量方法。 <mode>:= {X|Y|XY}
返回格式	{TRACk|MANual[,{X|Y|XY}]|MEASure}
示例	将光标模式设置为手动光标，光标类型设置为 X： *:CURSor:MODE MANual,X* *CURS:MODE MAN,X* 查询光标模式： *CURS:MODE?* 返回值： *MANual,X*

:CURSor:SOURce<n>

描述	手动光标下，设置或查询光标的通道源；追踪光标下，设置或查询光标 X1/X2 的通道源。
命令格式	:CURSor:SOURce<n> <source> :CURSor:SOURce<n>?
参数说明	<n>:= {1|2}，对应追踪光标 X 的序号。手动光标下可省略 <source>:= {C<n>|Z<n>|F<x>|M<m>|REF<r>|DIGital|HISTOGram} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● Z<n>:= 缩放波形，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式 ● REF<r>:= 参考波形，<r>:= {A|B|C|D}，参考波形序号 ● DIGital：数字通道 ● HISTOGram：波形直方图 **注意**：只能选择已打开的通道作为通道源。当光标模式为追踪时，不能设置源为 HISTOGram 或 DIGital。
返回格式	{C<n>|Z<n>|F<x>|M<m>|REF<r>|DIGital|HISTOGram}
示例	设置手动光标的通道源为 C1： *:CURSor:SOURce C1* *CURS:SOUR C1* 查询手动光标的通道源： *CURS:SOUR?* 返回值： *C1*
关联命令	:CURSor:MODE

:CURSor:X<n>

描述	设置或查询光标 X 的位置。
命令格式	:CURSor:X<n> <value> :CURSor:X<n>?
参数说明	<n>:= {1|2}，对应光标线的序号 <value>:= 数值，浮点型 NR3 格式，取值范围 [-水平格数/2*当前时基+水平延时，水平格数/2*当前时基+水平延时]
返回格式	浮点型 NR3 格式

示例	设置光标 X1 的当前位置为 1us： *:CURSor:X1 1.00E-6* *CURS:X1 1.00E-6* 查询光标 X1 的当前值： *CURS:X1?* 返回值： *1.00E-06*
关联命令	:CURSor:IXDelta :CURSor:XDELta

:CURSor:XDELta

描述	查询光标 X2 和 X1 当前位置的水平差。
命令格式	:CURSor:XDELta?
返回格式	浮点型 NR3 格式
示例	查询光标 X2 和 X1 当前位置的水平差： *CURS:XDEL?* 返回值： *-2E-06*
关联命令	:CURSor:X<n> :CURSor:IXDelta

:CURSor:Y<n>

描述	设置或查询光标 Y 的位置。
命令格式	:CURSor:Y<n> <value> :CURSor:Y<n>?
参数说明	<n>:={1|2}，对应光标线的序号 <value>:= 数值，浮点型 NR3 格式，取值范围 [-垂直格数/2*垂直档位+垂直偏移，垂直格数/2*垂直档位+垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置光标 Y1 的当前位置为 12V： *:CURSor:Y1 1.20E+01* *CURS:Y1 1.20E+01* 查询光标 Y1 的当前值： *CURS:Y1?*

	返回值： *1.20E+01*
关联命令	:CURSor:YDELta

:CURSor:YDELta

描述	查询光标 Y2 和 Y1 当前位置的垂直差。
命令格式	:CURSor:YDELta?
返回格式	浮点型 NR3 格式
示例	查询光标 Y2 和 Y1 当前位置的垂直差： *CURS:YDEL?* 返回值： *-2.4E+01*

多光标命令

多光标系统支持 5 种光标模式：手动 X，手动 Y，追踪，测量，XY。各光标模式下支持多根光标线，各光标线可设置不同参数。通过本节命令设置查询光标参数。

:CURSor:MANual:X<n>

描述	设置或查询指定手动 X 光标的开关状态。
命令格式	:CURSor:MANual:X<n> <state> :CURSor:MANual:X<n>?
参数说明	<n>:=手动 X 光标的序号，整型 NR1 格式，该值范围为［1,8］ <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置手动光标 X2 的开关状态为 ON： *:CURSor:MANual:X2 ON* *CURS:MAN:X2 ON* 查询手动光标 X2 的开关状态： *CURS:MAN:X2?* 返回值： *ON*
关联命令	:CURSor

:CURSor:MANual:X<n>:COLor

描述	设置或查询指定手动 X 光标的颜色。
命令格式	:CURSor:MANual:X<n>:COLor <color> :CURSor:MANual:X<n>:COLor?
参数说明	<n>:= 手动 X 光标的序号，整型 NR1 格式，该值范围为 [1,8] <color>:= {DEFault|DELTa|CUSTom[,<string>]} ● DEFault：默认为光标源的颜色 ● DELTa：同步为参考光标的颜色 ● CUSTom：自定义光标颜色，需指定<string> <string>:= 带引号的 16 进制 RGB 颜色代码。
返回格式	带引号的 16 进制 RGB 颜色代码
示例	设置手动光标 X2 的颜色为光标源的颜色： *:CURSor:MANual:X2:COLor DEFault* *CURS:MAN:X2:COL DEF* 查询手动光标 X2 的颜色： *CURS:MAN:X2:COL?* 返回值： *"#ffff00"*
关联命令	:CURSor:MANual:X<n> :CURSor:MANual:X<n>:SOURce :CURSor:MANual:X<n>:DTCursor

:CURSor:MANual:X<n>:DFOLlow

描述	设置或查询指定手动光标 X 与参考光标固定相对位置的开关状态。
命令格式	:CURSor:MANual:X<n>:DFOLlow <state> :CURSor:MANual:X<n>:DFOLlow?
参数说明	<n>:= 手动 X 光标的序号，整型 NR1 格式，该值范围为 [1,8] <state>:= {ON|OFF} **注意**：当参考光标源不为 NONE 时，才可设置
返回格式	{ON|OFF}
示例	设置手动光标 X2 与参考光标相对位置固定： *:CURSor:MANual:X2:DFOLlow ON* *CURS:MAN:X2:DFOL ON* 查询手动光标 X2 与参考光标相对位置固定状态：

	CURS:MAN:X2:DFOL? 返回值： *ON*
关联命令	:CURSor:MANual:X<n>:DTCursor

:CURSor:MANual:X<n>:DTCursor

描述	设置或查询指定手动 X 光标的参考光标。
命令格式	:CURSor:MANual:X<n>:DTCursor <source> :CURSor:MANual:X<n>:DTCursor?
参数说明	<n>:= 手动 X 光标的序号，整型 NR1 格式，该值范围为［1,8］ <source>:= {MX<m>|TRACK<t>|NONE} ● MX<m>:= 手动 X 光标，<m>:= 序号，整型 NR1 格式，该值范围为［1,8］ ● TRACK<t>:= 追踪光标，<t>:= 序号，整型 NR1 格式，该值范围为［1,8］ ● NONE:= 不设置参考光标
返回格式	{MX<m>|TRACK<t>|NONE}
示例	设置手动光标 X2 的参考光标为 MX1： *:CURSor:MANual:X2:DTCursor MX1* *CURS:MAN:X2:DTC MX1* 查询手动光标 X2 的参考光标： *CURS:MAN:X2:DTC?* 返回值： *MX1*
关联命令	:CURSor:MANual:X<n> :CURSor:TRACk<n>

:CURSor:MANual:X<n>:DVALue

描述	设置或查询指定手动 X 光标相对于参考光标的位置。
命令格式	:CURSor:MANual:X<n>:DVALue <value> :CURSor:MANual:X<n>:DVALue?
参数说明	<n>:=手动 X 光标的序号，整型 NR1 格式，该值范围为［1,8］ <value>:= 相对于参考光标的位置值，浮点型 NR3 格式。
返回格式	浮点型 NR3 格式
示例	设置手动光标 X2 相对于参考光标的位置为 1us：

	:CURSor:MANual:X2:DVALue 1.00E-06 *CURS:MAN:X2:DVAL 1.00E-06* 查询手动光标 X2 相对于参考光标的位置： *CURS:MAN:X2:DVAL?* 返回值： *1.00E-06*
关联命令	:CURSor:MANual:X<n>:DFOLlow :CURSor:MANual:X<n>:DTCursor

:CURSor:MANual:X<n>:LABel

描述	设置或查询指定手动 X 光标的标签。
命令格式	:CURSor:MANual:X<n>:LABel <state> :CURSor:MANual:X<n>:LABel?
参数说明	<n>:= 手动 X 光标的序号，整型 NR1 格式，该值范围为 [1,8] <state>:= {DEFault|DELTa|CUSTom[,<string>]} ● DEFault：默认标签为空 ● DELTa：同步为参考光标的标签 ● CUSTom：自定义标签，需指定<string> <string>:= 带引号的 ASCII 文本字符串，字符串的长度限制为 20。
返回格式	带引号的 ASCII 文本字符串。
示例	设置手动光标 X2 的标签为默认标签： *:CURSor:MANual:X2:LABel DEFault* *CURS:MAN:X2:LAB DEF* 查询手动光标 X2 的标签： *CURS:MAN:X2:LAB?* 返回值： *""*
关联命令	:CURSor:MANual:X<n> :CURSor:MANual:X<n>:DTCursor

:CURSor:MANual:X<n>:POSition

描述	设置或查询指定手动 X 光标的位置。
命令格式	:CURSor:MANual:X<n>:POSition <value> :CURSor:MANual:X<n>:POSition?

参数说明	<n>:=手动 X 光标的序号，整型 NR1 格式，该值范围为［1,8］ <value>:= 数值，浮点型 NR3 格式，取值范围［-水平格数/2*当前时基+水平延时，水平格数/2*当前时基+水平延时］
返回格式	浮点型 NR3 格式
示例	设置手动光标 X2 的当前位置为 1us： *:CURSor:MANual:X2:POSition 1.00E-06* *CURS:MAN:X2:POS 1.00E-06* 查询手动光标 X2 的当前位置： *CURS:MAN:X2:POS?* 返回值： *1.00E-06*
关联命令	:CURSor:MANual:X<n>

:CURSor:MANual:X<n>:SOURce

描述	设置或查询指定手动 X 光标的信源。
命令格式	:CURSor:MANual:X<n>:SOURce <source> :CURSor:MANual:X<n>:SOURce?
参数说明	<n>:=手动 X 光标的序号，整型 NR1 格式，该值范围为［1,8］ <source>:= {C<n>|Z<n>|F<x>|ZF<x>|M<m>|ZM<m>|DIGital|ZDIGital |HISTOGram} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● Z<n>:= 缩放波形，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● ZF<x>:= 缩放数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式 ● ZM<m>:= 缩放内存波形，<m>:= 序号，整型 NR1 格式 ● DIGital：数字通道 ● HISTOGram：波形直方图
返回格式	{C<n>|Z<n>|F<x>|ZF<x>|M<m>|ZM<m>|DIGital|ZDIGital |HISTOGram}
示例	设置手动光标 X2 的信源为 C1： *:CURSor:MANual:X2:SOURce C1* *CURS:MAN:X2:SOUR C1* 查询手动光标 X2 信源： *CURS:MAN:X2:SOUR?* 返回值： *C1*

关联命令	:CURSor:MANual:X<n>

:CURSor:MANual:Y<n>

描述	设置或查询指定手动 Y 光标的开关状态。
命令格式	:CURSor:MANual:Y<n> <state> :CURSor:MANual:Y<n>?
参数说明	<n>:=手动 Y 光标的序号，整型 NR1 格式，该值范围为［1,8］ <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置手动光标 Y2 的开关状态为 ON： *:CURSor:MANual:Y2 ON* *CURS:MAN:Y2 ON* 查询手动光标 Y2 的开关状态： *CURS:MAN:Y2?* 返回值： *ON*
关联命令	:CURSor

:CURSor:MANual:Y<n>:COLor

描述	设置或查询指定手动 Y 光标的颜色。
命令格式	:CURSor:MANual:Y<n>:COLor <color> :CURSor:MANual:Y<n>:COLor?
参数说明	<n>:=手动 Y 光标的序号，整型 NR1 格式，该值范围为［1,8］ <color>:= {DEFault|DELTa|CUSTom[,<string>]} ● DEFault：默认为光标源的颜色 ● DELTa：同步为参考光标的颜色 ● CUSTom：自定义光标颜色，需指定<string> <string>:= 带引号的 16 进制 RGB 颜色代码。
返回格式	带引号的 16 进制 RGB 颜色代码
示例	设置手动光标 Y2 的颜色为光标源的颜色： *:CURSor:MANual:Y2:COLor DEFault* *CURS:MAN:Y2:COL DEF* 查询手动光标 Y2 的颜色： *CURS:MAN:Y2:COL?*

	返回值： *"#ffff00"*
关联命令	:CURSor:MANual:Y<n> :CURSor:MANual:Y<n>:SOURce :CURSor:MANual:Y<n>:DTCursor

:CURSor:MANual:Y<n>:DFOLlow

描述	设置或查询指定手动 Y 光标与参考光标固定相对位置的开关状态。
命令格式	:CURSor:MANual:Y<n>:DFOLlow <state> :CURSor:MANual:Y<n>:DFOLlow?
参数说明	<n>:=手动 Y 光标的序号，整型 NR1 格式，该值范围为［1,8］ <state>:= {ON|OFF} **注意**：当参考光标源不为 NONE 时，才可设置
返回格式	{ON|OFF}
示例	设置手动光标 Y2 与参考光标相对位置固定： *:CURSor:MANual:Y2:DFOLlow ON* *CURS:MAN:Y2:DFOL ON* 查询手动光标 Y2 与参考光标相对位置固定状态： *CURS:MAN:Y2:DFOL?* 返回值： *ON*
关联命令	:CURSor:MANual:Y<n>:DTCursor

:CURSor:MANual:Y<n>:DTCursor

描述	设置或查询指定手动 Y 光标的的参考光标。
命令格式	:CURSor:MANual:Y<n>:DTCursor <source> :CURSor:MANual:Y<n>:DTCursor?
参数说明	<n>:=手动 Y 光标的序号，整型 NR1 格式，该值范围为［1,8］ <source>:= {MY<m>|NONE} ● MY<m>:= 手动 Y 光标，<m>:= 序号，整型 NR1 格式，该值范围为［1,8］ ● NONE:= 不设置参考光标
返回格式	{MY<m>|NONE}

示例	设置手动光标 Y2 的参考光标为 MY1： *:CURSor:MANual:Y2:DTCursor MY1* *CURS:MAN:Y2:DTC MY1* 查询手动光标 Y2 的参考光标： *CURS:MAN:Y2:DTC?* 返回值： *MY1*
关联命令	:CURSor:MANual:Y<n>

:CURSor:MANual:Y<n>:DVALue

描述	设置或查询指定手动 Y 光标相对于参考光标的位置。
命令格式	:CURSor:MANual:Y<n>:DVALue <value> :CURSor:MANual:Y<n>:DVALue?
参数说明	<n>:=手动 Y 光标的序号，整型 NR1 格式，该值范围为 [1,8] <value>:= 相对于参考光标的位置值，浮点型 NR3 格式。
返回格式	浮点型 NR3 格式
示例	设置手动光标 Y2 相对于参考光标的位置为 1V： *:CURSor:MANual:Y2:DVALue 1.00E+00* *CURS:MAN:Y2:DVAL 1.00E+00* 查询手动光标 Y2 相对于参考光标的位置： *CURS:MAN:Y2:DVAL?* 返回值： *1.00E+00*
关联命令	:CURSor:MANual:Y<n>:DTCursor :CURSor:MANual:Y<n>:DFOLlow

:CURSor:MANual:Y<n>:LABel

描述	设置或查询指定手动 Y 光标的标签。
命令格式	:CURSor:MANual:Y<n>:LABel <state> :CURSor:MANual:Y<n>:LABel?
参数说明	<n>:= 手动 Y 光标的序号，整型 NR1 格式，该值范围为 [1,8] <state>:= {DEFault|DELTa|CUSTom[,<string>]} ● DEFault：默认标签为空 ● DELTa：同步为参考光标的标签

	● CUSTom：自定义标签，需指定<string> <string>:= 带引号的 ASCII 文本字符串，字符串的长度限制为 20。
返回格式	{DEFault|DELTa|CUSTom[,<string>]}
示例	设置手动光标 Y2 的标签为默认标签： *:CURSor:MANual:Y2:LABel DEFault* *CURS:MAN:Y2:LAB DEF* 查询手动光标 Y2 的标签： *CURS:MAN:Y2:LAB?* 返回值： *DEFault*
关联命令	:CURSor:MANual:Y<n> :CURSor:MANual:Y<n>:DTCursor

:CURSor:MANual:Y<n>:POSition

描述	设置或查询指定手动 Y 光标的位置。
命令格式	:CURSor:MANual:Y<n>:POSition <value> :CURSor:MANual:Y<n>:POSition?
参数说明	<n>:=手动 Y 光标的序号，整型 NR1 格式，该值范围为 [1,8] <value>:= 数值，浮点型 NR3 格式，取值范围[-垂直格数/2*垂直档位+垂直偏移，垂直格数/2*垂直档位+垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置手动光标 Y2 的当前位置为 1V： *:CURSor:MANual:Y2:POSition 1.00E+00* *CURS:MAN:Y2:POS 1.00E+00* 查询手动光标 Y2 的当前位置： *CURS:MAN:Y2:POS?* 返回值： *1.00E+00*
关联命令	:CURSor:MANual:Y<n>

:CURSor:MANual:Y<n>:SOURce

描述	设置或查询指定手动 Y 光标的信源。
命令格式	:CURSor:MANual:Y<n>:SOURce <source>

	:CURSor:MANual:Y<n>:SOURce?
参数说明	<n>:= 手动 Y 光标的序号，整型 NR1 格式，该值范围为 [1,8] <source>:= {C<n>|Z<n>|F<x>|ZF<x>|M<m>|ZM<m>|HISTOGram} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● Z<n>:= 缩放波形，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● ZF<x>:= 缩放数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式 ● ZM<m>:= 缩放内存波形，<m>:= 序号，整型 NR1 格式 ● HISTOGram：波形直方图
返回格式	{C<n>|Z<n>|F<x>|ZF<x>|M<m>|ZM<m>|DIGital|ZDIGital |HISTOGram}
示例	设置手动光标 Y2 的信源为 C1： *:CURSor:MANual:Y2:SOURce C1* *CURS:MAN:Y2:SOUR C1* 查询手动光标 Y2 信源： *CURS:MAN:Y2:SOUR?* 返回值： *C1*
关联命令	:CURSor:MANual:Y<n>

:CURSor:MEASure<n>

描述	设置或查询指定测量光标 MEA 的开关状态。
命令格式	:CURSor:MEASure<n> <state> :CURSor:MEASure<n>?
参数说明	<n>:= 测量光标 MEA 的序号，整型 NR1 格式，该值范围为 [1,4] <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置测量光标 MEA2 的开关状态为 ON： *:CURSor:MEASure2 ON* *CURS:MEAS2 ON* 查询测量光标 MEA2 的开关状态： *CURS:MEAS2?* 返回值： *ON*
关联命令	:CURSor

:CURSor:MEASure<n>:COLor

描述	设置或查询指定策略光标 MEA 的颜色。
命令格式	:CURSor:MEASure<n>:COLor <color> :CURSor:MEASure<n>:COLor?
参数说明	<n>:= 测量光标 MEA 的序号，整型 NR1 格式，该值范围为 [1,4] <color>:= {DEFault|CUSTom[,<string>]} ● DEFault：默认为光标源的颜色 ● CUSTom：自定义光标颜色，需指定<string> <string>:= 带引号的 16 进制 RGB 颜色代码。
返回格式	带引号的 16 进制 RGB 颜色代码
示例	设置测量光标 MEA2 的颜色为光标源的颜色： *:CURSor:MEASure2:COLor DEFault* *CURS:MEAS2:COL DEF* 查询测量光标 MEA2 的光标颜色： *CURS:MEAS2:COL?* 返回值： *"#ffff00"*
关联命令	:CURSor:MEASure<n>

:CURSor:MEASure<n>:LABel

描述	设置或查询指定测量光标 MEA 的标签。
命令格式	:CURSor:MEASure<n>:LABel <state> :CURSor:MEASure<n>:LABel?
参数说明	<n>:= 追踪光标 TX 的序号，整型 NR1 格式，该值范围为 [1,8] <state>:= {DEFault|CUSTom[,<string>]} ● DEFault：默认标签为空 ● CUSTom：自定义标签，需指定<string> <string>:= 带引号的 ASCII 文本字符串，字符串的长度限制为 20。
返回格式	带引号的 ASCII 文本字符串
示例	设置测量光标 MEA2 的标签为默认标签： *:CURSor:MEASure2:LABel DEFault* *CURS:MEAS2:LAB DEF* 查询测量光标 MEA2 的标签： *CURS:MEAS2:LAB?* 返回值：

	"MEA2"
关联命令	:CURSor:MEASure<n>

:CURSor:MEASure<n>:MITem

描述	设置或查询指定测量光标 MEA 的测量项。
命令格式	:CURSor:MEASure<n>:MITem <type>,<source1>[,<source2>] :CURSor:MEASure<n>:MITem?
参数说明	<n>:= 测量光标序号，整型 NR1 格式，取值范围 [1,4] <type>:= 参数同测量类型，详见"表 5- 2 测量参数表"。 <source1>:= {C<n>|Z<n>|F<x>|M<m>|D<d>|ZD<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● Z<n>:= 缩放波形，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式，取值范围 [1,4] ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式，取值范围 [0,15] ● ZD<d>:= 缩放数字通道，<d>:= 序号，整型 NR1 格式，取值范围 [0,15] <source2>:= {C<n>}。当高级测量项为通道延迟，需要指定第二个源 **注意**：只要当 Zoom 处于开启状态，选择 Z<n>作为源才有效
返回格式	<type>,<source1>[,<source2>]
示例	当高级测量打开时，设置测量光标 MEA2 的测量项为 PKPK(C2)： *:CURSor:MEASure2:MITem PKPK,C2* *CURS:MEAS2:MIT PKPK,C2* 查询测量光标 MEA2 的测量项： *CURS:MEAS2:MIT?* 返回值： *PKPK,C2*
关联命令	:CURSor:MEASure<n> :MEASure:ADVanced:P<n>:TYPE

:CURSor:TRACk<n>

描述	设置或查询指定追踪光标 TX 的开关状态。
命令格式	:CURSor:TRACk<n> <state>

	:CURSor:TRACk<n>?
参数说明	<n>:=追踪光标 TX 的序号，整型 NR1 格式，该值范围为［1,8］ <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置追踪光标 TX2 的开关状态为 ON： *:CURSor:TRACk2 ON* *CURS:TRAC2 ON* 查询追踪光标 TX2 的开关状态： *CURS:TRAC2?* 返回值： *ON*
关联命令	:CURSor

:CURSor:TRACk<n>:COLor

描述	设置或查询指定追踪光标 TX 的颜色。
命令格式	:CURSor:TRACk<n>:COLor <color> :CURSor:TRACk<n>:COLor?
参数说明	<n>:=追踪光标 TX 的序号，整型 NR1 格式，该值范围为［1,8］ <color>:= {DEFault|DELTa|CUSTom[,<string>]} ● DEFault：默认为光标源的颜色 ● DELTa：同步为参考光标的颜色 ● CUSTom：自定义光标颜色，需指定<string> <string>:= 带引号的 16 进制 RGB 颜色代码。
返回格式	带引号的 16 进制 RGB 颜色代码
示例	设置追踪光标 TX2 的颜色为光标源的颜色： *:CURSor:TRACk2:COLor DEFault* *CURS:TRAC2:COL DEF* 查询追踪光标 TX2 的颜色： *CURS:TRAC2:COL?* 返回值： *"#ffff00"*
关联命令	:CURSor:TRACk<n> :CURSor:TRACk<n>:SOURce :CURSor:TRACk<n>:DTCursor

:CURSor:TRACk<n>:DFOLlow

描述	设置或查询指定追踪光标 TX 与参考光标固定相对位置的开关状态。
命令格式	:CURSor:TRACk<n>:DFOLlow <state> :CURSor:TRACk<n>:DFOLlow?
参数说明	<n>:=追踪光标 TX 的序号，整型 NR1 格式，该值范围为［1,8］ <state>:= {ON|OFF} **注意**：当参考光标源不为 NONE 时，才可设置
返回格式	{ON|OFF}
示例	设置追踪光标 TX2 与参考光标相对位置固定： *:CURSor:TRACk2:DFOLlow ON* *CURS:TRAC2:DFOL ON* 查询追踪光标 TX2 与参考光标相对位置固定状态： *CURS:TRAC2:DFOL?* 返回值： *ON*
关联命令	:CURSor:TRACk<n>:DTCursor

:CURSor:TRACk<n>:DTCursor

描述	设置或查询指定追踪光标 TX 的参考光标。
命令格式	:CURSor:TRACk<n>:DTCursor <source> :CURSor:TRACk<n>:DTCursor?
参数说明	<n>:=追踪光标 TX 的序号，整型 NR1 格式，该值范围为［1,8］ <source>:= {MX<m>|TRACK<t>|NONE} ● MX<m>:= 手动 X 光标，<m>:= 手动光标和追踪光标的序号，整型 NR1 格式，该值范围为［1,8］ ● TRACK<t>:= 追踪光标，<t>:= 序号，整型 NR1 格式，该值范围为［1,8］ ● NONE:= 不设置参考光标
返回格式	{MX<m>>|TRACK<t>|NONE}
示例	设置追踪光标 TX2 的参考光标为 MX1： *:CURSor:TRAC2:DTCursor MX1* *CURS:TRAC2:DTC MX1* 查询追踪光标 TX2 的参考光标： *CURS:TRAC2:DTC?* 返回值：

	MX1
关联命令	:CURSor:MANual:X<n> :CURSor:TRACk<n>

:CURSor:TRACk<n>:DVALue

描述	设置或查询指定追踪光标 TX 相对于参考光标的位置。
命令格式	:CURSor:TRACk<n>:DVALue <value> :CURSor:TRACk<n>:DVALue?
参数说明	<n>:=追踪光标 TX 的序号，整型 NR1 格式，该值范围为 [1,8] <value>:= 相对于参考光标的位置值，浮点型 NR3 格式。
返回格式	浮点型 NR3 格式
示例	设置追踪光标 TX2 相对于参考光标的位置为 1us： *:CURSor:TRACk2:DVALue 1.00E-06* *CURS:TRAC2:DVAL 1.00E-06* 查询追踪光标 TX2 相对于参考光标的位置： *CURS:TRAC2:DVAL?* 返回值： *1.00E-06*
关联命令	:CURSor:TRACk<n>:DTCursor :CURSor:TRACk<n>:DFOLlow

:CURSor:TRACk<n>:LABel

描述	设置或查询指定追踪光标 TX 的标签。
命令格式	:CURSor:TRACk<n>:LABel <state> :CURSor:TRACk<n>:LABel?
参数说明	<n>:= 追踪光标 TX 的序号，整型 NR1 格式，该值范围为 [1,8] <state>:= {DEFault|DELTa|CUSTom[,<string>]} ● DEFault：默认标签为空 ● DELTa：同步为参考光标的标签 ● CUSTom：自定义标签，需指定<string> <string>:= 带引号的 ASCII 文本字符串，字符串的长度限制为 20。
返回格式	带引号的 ASCII 文本字符串。
示例	设置追踪光标 TX2 的标签为默认标签：

	:CURSor:TRACk2:LABel DEFault *CURS:TRAC2:LAB DEF* 查询追踪光标 TX2 的标签： *CURS:TRAC2:LAB?* 返回值： *“”*
关联命令	:CURSor:TRACk<n> :CURSor:TRACk<n>:DTCursor

:CURSor:TRACk<n>:POSition

描述	设置或查询指定追踪光标 TX 的水平位置。
命令格式	:CURSor:TRACk<n>:POSition <value> :CURSor:TRACk<n>:POSition?
参数说明	<n>:= 追踪光标 TX 的序号，整型 NR1 格式，该值范围为 [1,8] <value>:= 数值，浮点型 NR3 格式，取值范围 [-水平格数/2*当前时基+水平延时，水平格数/2*当前时基+水平延时]
返回格式	浮点型 NR3 格式
示例	设置追踪光标 TX2 的当前位置为 1us： *:CURSor:TRACk2:POSition 1.00E-06* *CURS:TRAC2:POS 1.00E-06* 查询追踪光标 TX2 的当前位置： *CURS:TRAC2:POS?* 返回值： *1.00E-06*
关联命令	:CURSor:TRACk<n>

:CURSor:TRACk<n>:SOURce

描述	设置或查询指定追踪光标 TX 的信源。
命令格式	:CURSor:TRACk<n>:SOURce <source> :CURSor:TRACk<n>:SOURce?
参数说明	<n>:= 追踪光标 TX 的序号，整型 NR1 格式，该值范围为 [1,8] <source>:= {C<n>|Z<n>|F<x>|ZF<x>|M<m>|ZM<m>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● Z<n>:= 缩放波形，<n>:= 序号，整型 NR1 格式

	● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● ZF<x>:= 缩放数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式 ● ZM<m>:= 缩放内存波形，<m>:= 序号，整型 NR1 格式
返回格式	{C<n>|Z<n>|F<x>|ZF<x>|M<m>|ZM<m>}
示例	设置追踪光标 TX2 的信源为 C1： *:CURSor:TRACk2:SOURce C1* *CURS:TRAC2:SOUR C1* 查询追踪光标 TX2 信源： *CURS:TRAC2:SOUR?* 返回值： *C1*
关联命令	:CURSor:TRACk<n>

:CURSor:TRACk<n>:VALue

描述	查询指定追踪光标 TX 的垂直方向位置。
命令格式	:CURSor:TRACk<n>:VALue?
参数说明	<n>:= 追踪光标 TX 的序号，整型 NR1 格式，该值范围为［1,8］
返回格式	浮点型 NR3 格式
示例	查询追踪光标 TX2 的垂直位置： *CURS:TRAC2:VAL?* 返回值： *1.00E+00*
关联命令	:CURSor:TRACk<n>:POSition

:CURSor:XY:X<n>

描述	设置或查询指定 XY 模式 X 光标的开关状态。
命令格式	:CURSor:XY:X<n> <state> :CURSor:XY:X<n>?
参数说明	<n>:= XY 模式 X 光标的序号，整型 NR1 格式，该值范围为［1,2］ <state>:= {ON|OFF} **注意**：

	● XY 开启后，才可以设置 XY 模式光标 ● XY 关闭后，XY 模式光标自动关闭
返回格式	{ON|OFF}
示例	设置 XY 模式光标 X2 的开关状态为 ON: *:CURSor:XY:X2 ON* *CURS:XY:X2 ON* 查询 XY 模式光标 X2 的开关状态: *CURS:XY:X2?* 返回值: *ON*
关联命令	:CURSor :ACQuire:MODE

:CURSor:XY:X<n>:COLor

描述	设置或查询指定 XY 模式 X 光标的颜色。
命令格式	:CURSor:XY:X<n>:COLor <color> :CURSor:XY:X<n>:COLor?
参数说明	<n>:= XY 模式 X 光标的序号，整型 NR1 格式，该值范围为［1,2］ <color>:= {DEFault|CUSTom[,<string>]} ● DEFault：默认为光标源的颜色 ● CUSTom：自定义光标颜色，需指定<string> <string>:= 带引号的 16 进制 RGB 颜色代码。
返回格式	带引号的 16 进制 RGB 颜色代码
示例	设置 XY 模式光标 X2 的颜色为光标源的颜色: *:CURSor:XY:X2:COLor DEFault* *CURS:XY:X2:COL DEF* 查询 XY 模式光标 X2 的颜色: *CURS:XY:X2:COL?* 返回值: *"#ffff00"*
关联命令	:CURSor:XY:X<n> :CURSor:XY:X<n>:DTCursor

:CURSor:XY:X2:DFOLlow

描述	设置或查询指定 XY 模式 X2 光标与参考光标 XY_X1 固定相对位置的开关状态。
命令格式	:CURSor:XY:X2:DFOLlow <state> :CURSor:XY:X2:DFOLlow?
参数说明	<state>:= {ON|OFF} **注意**：当参考光标源不为 NONE 时，才可设置
返回格式	{ON|OFF}
示例	设置 XY 模式光标 X2 与参考光标 XY_X1 相对位置固定： *:CURSor:XY:X2:DFOLlow ON* *CURS:XY:X2:DFOL ON* 查询 XY 模式光标 X2 与参考光标 XY_X1 相对位置固定状态： *CURS:XY:X2:DFOL?* 返回值： *ON*
关联命令	:CURSor:XY:X2:DTCursor

:CURSor:XY:X2:DTCursor

描述	设置或查询指定 XY 模式 X2 光标的参考光标。当前参考光标仅可为 XY_X1。
命令格式	:CURSor:XY:X2:DTCursor <source> :CURSor:XY:X2:DTCursor?
参数说明	<source>:= {XY_X1|NONE} ● XY_X1: XY_X1 光标 ● NONE:不设置参考光标
返回格式	{XY_X1|NONE}
示例	设置 XY 模式光标 X2 的参考光标为 XY_X1： *:CURSor:XY:X2:DTCursor XY_X1* *CURS:XY:X2:DTC XY_X1* 查询 XY 模式光标 X2 的参考光标源： *CURS:XY:X2:DTC?* 返回值： *XY_X1*
关联命令	:CURSor:XY:X<n>

:CURSor:XY:X2:DVALue

描述	设置或查询指定 XY 模式 X2 光标相对于参考光标 XY_X1 的位置。
命令格式	:CURSor:XY:X2:DVALue <value> :CURSor:XY:X2:DVALue?
参数说明	<value>:= 相对于参考光标的位置值，浮点型 NR3 格式。
返回格式	浮点型 NR3 格式
示例	设置 XY 模式光标 X2 相对于参考光标 XY_X1 的位置为 1V： *:CURSor:XY:X2:DVALue 1.00E+00* *CURS:XY:X2:DVAL 1.00E+00* 查询 XY 模式光标 X2 相对于参考光标的位置： *CURS:XY:X2:DVAL?* 返回值： *1.00E+00*
关联命令	:CURSor:XY:X2:DTCursor :CURSor:XY:X2:DFOLlow

:CURSor:XY:X<n>:LABel

描述	设置或查询指定 XY 模式 X 光标的标签。
命令格式	:CURSor:XY:X<n>:LABel <state> :CURSor:XY:X<n>:LABel?
参数说明	<n>:= XY 模式 X 光标的序号，整型 NR1 格式，该值范围为 [1,2] <state>:= {DEFault|DELTa|CUSTom[,<string>]} ● DEFault：默认标签为空 ● DELTa：同步为参考光标的标签 ● CUSTom：自定义标签，需指定<string> <string>:= 带引号的 ASCII 文本字符串，字符串的长度限制为 20。
返回格式	带引号的 ASCII 文本字符串
示例	设置 XY 模式光标 X2 的标签为默认标签： *:CURSor:XY:X2:LABel DEFault* *CURS:XY:X2:LAB DEF* 查询 XY 模式光标 X2 的标签： *CURS:XY:X2:LAB?* 返回值： *"XY"*

关联命令	:CURSor:XY:X<n> :CURSor:XY:X<n>:DTCursor

:CURSor:XY:X<n>:POSition

描述	设置或查询指定 XY 模式 X 光标的位置。
命令格式	:CURSor:XY:X<n>:POSition <value> :CURSor:XY:X<n>:POSition?
参数说明	<n>:=XY 模式 X 光标的序号，整型 NR1 格式，该值范围为 [1,2] <value>:= 数值，浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置 XY 模式光标 X2 的当前位置为 1V： *:CURSor:XY:X2:POSition 1.00E+00* *CURS:XY:X2:POS 1.00E+00* 查询 XY 模式光标 X2 的当前位置： *CURS:XY:X2:POS?* 返回值： *1.00E+00*
关联命令	:CURSor:XY:X<n>

:CURSor:XY:Y<n>

描述	设置或查询指定 XY 模式 Y 光标的开关状态。
命令格式	:CURSor:XY:Y<n> <state> :CURSor:XY:Y<n>?
参数说明	<n>:=XY 模式 Y 光标的序号，整型 NR1 格式，该值范围为 [1,2] <state>:= {ON|OFF} **注意**： ● XY 开启后，才可以设置 XY 模式光标 ● XY 关闭后，XY 模式光标自动关闭
返回格式	{ON|OFF}
示例	设置 XY 模式光标 Y2 的开关状态为 ON： *:CURSor:XY:Y2 ON* *CURS:XY:Y2 ON* 查询 XY 模式光标 Y2 的开关状态：

	CURS:XY:Y2? 返回值： *ON*
关联命令	:CURSor :ACQuire:MODE

:CURSor:XY:Y<n>:COLor

描述	设置或查询指定 XY 模式 Y 光标的颜色。
命令格式	:CURSor:XY:Y<n>:COLor <color> :CURSor:XY:Y<n>:COLor?
参数说明	<n>:=XY 模式 Y 光标的序号，整型 NR1 格式，该值范围为［1,2］ <color>:= {DEFault|DELTa|CUSTom[,<string>]} ● DEFault：默认为光标源的颜色 ● DELTa：同步为参考光标的颜色 ● CUSTom：自定义光标颜色，需指定<string> <string>:= 带引号的 16 进制 RGB 颜色代码。
返回格式	带引号的 16 进制 RGB 颜色代码
示例	设置 XY 模式光标 Y2 的颜色为光标源的颜色： *:CURSor:XY:Y2:COLor DEFault* *CURS:XY:Y2:COL DEF* 查询 XY 模式光标 Y2 的颜色： *CURS:XY:Y2:COL?* 返回值： *"#ffff00"*
关联命令	:CURSor:XY:Y<n> :CURSor:XY:Y<n>:DTCursor

:CURSor:XY:Y2:DFOLlow

描述	设置或查询指定 XY 模式 Y2 光标与参考光标 XY_Y1 固定相对位置的开关状态。
命令格式	:CURSor:XY:Y2:DFOLlow <state> :CURSor:XY:Y2:DFOLlow?
参数说明	<state>:= {ON|OFF}

	注意：当参考光标源不为 NONE 时，才可设置
返回格式	{ON|OFF}
示例	设置 XY 模式光标 Y2 与参考光标 XY_Y1 相对位置固定： *:CURSor:XY:Y2:DFOLlow ON* *CURS:XY:Y2:DFOL ON* 查询 XY 模式光标 Y2 与参考光标 XY_Y1 相对位置固定状态： *CURS:XY:Y2:DFOL?* 返回值： *ON*
关联命令	:CURSor:XY:Y2:DTCursor

:CURSor:XY:Y2:DTCursor

描述	设置或查询指定 XY 模式 Y2 光标的参考光标。当前参考光标仅可为 XY_Y1。
命令格式	:CURSor:XY:Y2:DTCursor <source> :CURSor:XY:Y2:DTCursor?
参数说明	<source>:= {XY_Y1|NONE} ● XY_Y1: XY 模式 Y1 光标 ● NONE:不设置参考光标
返回格式	{XY_Y1|NONE}
示例	设置 XY 模式光标 Y2 的参考光标为 XY_Y1： *:CURSor:XY:Y2:DTCursor XY_Y1* *CURS:XY:Y2:DTC XY_Y1* 查询 XY 模式光标 Y2 的参考光标： *CURS:XY:Y2:DTC?* 返回值： *XY_Y1*
关联命令	:CURSor:XY:Y<n>

:CURSor:XY:Y2:DVALue

描述	设置或查询指定 XY 模式 Y2 光标相对于参考光标 XY_Y1 的位置。
命令格式	:CURSor:XY:Y2:DVALue <value> :CURSor:XY:Y2:DVALue?
参数说明	<value>:= 相对于参考光标的位置值，浮点型 NR3 格式。

返回格式	浮点型 NR3 格式
示例	设置 XY 模式光标 Y2 相对于参考光标 XY_Y1 的位置为 1V： *:CURSor:XY:Y2:DVALue 1.00E+00* *CURS:XY:Y2:DVAL 1.00E+00* 查询 XY 模式光标 Y2 相对于参考光标的位置： *CURS:XY:Y2:DVAL?* 返回值： *1.00E+00*
关联命令	:CURSor:XY:Y2:DTCursor :CURSor:XY:Y2:DFOLlow

:CURSor:XY:Y<n>:LABel

描述	设置或查询指定 XY 模式 Y 光标的标签。
命令格式	:CURSor:XY:Y<n>:LABel <state> :CURSor:XY:Y<n>:LABel?
参数说明	<n>:= XY 模式 Y 光标的序号，整型 NR1 格式，该值范围为 [1,2] <state>:= {DEFault|DELTa|CUSTom[,<string>]} ● DEFault：默认标签为空 ● DELTa：同步为参考光标的标签 ● CUSTom：自定义标签，需指定<string> <string>:= 带引号的 ASCII 文本字符串，字符串的长度限制为 20。
返回格式	带引号的 ASCII 文本字符串
示例	设置 XY 模式光标 Y2 的标签为默认标签： *:CURSor:XY:Y2:LABel DEFault* *CURS:XY:Y2:LAB DEF* 查询 XY 模式光标 Y2 的标签： *CURS:XY:Y2:LAB?* 返回值： *"XY"*
关联命令	:CURSor:XY:Y<n> :CURSor:XY:Y<n>:DTCursor

:CURSor:XY:Y<n>:POSition

描述	设置或查询指定 XY 模式 Y 光标的位置。

命令格式	:CURSor:XY:Y<n>:POSition <value> :CURSor:XY:Y<n>:POSition?
参数说明	<n>:=XY 模式 Y 光标的序号，整型 NR1 格式，该值范围为 [1,2] <value>:= 数值，浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置 XY 模式光标 Y2 的当前位置为 1V： *:CURSor:XY:Y2:POSition 1.00E+00* *CURS:XY:Y2:POS 1.00E+00* 查询 XY 模式光标 Y2 的当前位置： *CURS:XY:Y2:POS?* 返回值： *1.00E+00*
关联命令	:CURSor:XY:Y<n>

:CURSor:CLEar

描述	多光标清除当前所有光标，即关闭所有光标线。
命令格式	:CURSor:CLEar
示例	多光标清除当前所有光标： *:CURSor:CLEar* *CURS:CLE*

:CURSor:RESet

描述	多光标恢复默认设置，即：MX1-MX2 打开并建立参考关系，MY1-MY2 打开并建立参考关系，其余光标关闭。
命令格式	:CURSor:RESet
示例	多光标恢复默认设置： *:CURSor:RESet* *CURS:RES*

5.8 DECode 命令系统

:DECode 命令子系统控制示波器的解码功能。

:DECode

描述	设置或查询解码功能的开关状态。
命令格式	:DECode <state> :DECode?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置解码功能开启： *:DECode ON* *DEC ON* 查询当前解码功能开关状态： *DEC?* 返回值： *ON*

:DECode:LIST

描述	设置或查询解码列表状态。
命令格式	:DECode:LIST <state> :DECode:LIST?
参数说明	<state>:= {OFF|D1|D2} ● OFF：解码列表关闭 ● D1：解码总线 1 ● D2：解码总线 2
返回格式	{OFF|D1|D2}
示例	设置解码列表为解码总线 1 的结果： *:DECode:LIST D1* *DEC:LIST D1* 查询当前解码列表显示状态： *DEC:LIST?* 返回值： *D1*

:DECode:LIST:LINE

描述	设置或查询解码列表显示的行数。
命令格式	:DECode:LIST:LINE <value> :DECode:LIST:LINE?
参数说明	<value>:= 整型 NR1 格式，该值范围为 [1,7]
返回格式	整型 NR1 格式
示例	设置解码列表显示行数为 6： *:DECode:LIST:LINE 6* *DEC:LIST:LINE 6* 查询解码列表显示行数： *DEC:LIST:LINE?* 返回值： *6*

:DECode:LIST:SCRoll

描述	设置或查询解码列表选定行。
命令格式	:DECode:LIST:SCRoll <value> :DECode:LIST:SCRoll?
参数说明	<value>:= 整型 NR1 格式
返回格式	整型 NR1 格式
示例	设置选定解码列表第三行： *:DECode:LIST:SCRoll 3* *DEC:LIST:SCR 3* 查询当前解码列表选定行： *DEC:LIST:SCR?* 返回值： *3*
关联命令	:DECode:LIST :DECode:LIST:LINE

:DECode:LIST<n>:RESult

描述	查询指定解码列表的结果。
命令格式	:DECode:LIST<n>:RESult?
参数说明	<n>:= {1|2}，对应总线的列表序号
返回格式	<ascii_text>:= ASCII 形式数据。用逗号分隔，每一帧用换行符分隔。数据内容与命令":DECode:BUS<n>:FORMat"设置的格式有关。第一行为头部描述信息。
示例	查询解码列表 1 的解码结果： *DEC:LIST1:RES?* 返回值： *UART,\sTime,\sRx,\sRx\sErr,\sTx,\sTx\sErr\s* *1,-93.7501us,0x4C,\s,\s,\s* *2,-93.7501us,\s,\s,0x4C,\s* *3,-4.20950us,0x45,\s,\s,\s* *4,-4.20950us,\s,\s,0x45,\s* *5,85.4178us,,Less\sLength,\s,\s* *6,85.4178us,\s,\s,,Less\sLength*

:DECode:BUS<n>

描述	设置或查询解码总线的显示状态。
命令格式	:DECode:BUS<n> <state> :DECode:BUS<n>?
参数说明	<n>:= {1|2}，总线序号 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启解码总线 1： *:DECode:BUS1 ON* *DEC:BUS1 ON* 查询解码总线 1 的状态： *DEC:BUS1?* 返回值： *ON*
关联命令	:DECode

:DECode:BUS<n>:COPY

描述	设置解码设置和触发设置同步。
命令格式	:DECode:BUS<n>:COPY <operation>
参数说明	<n>:= {1|2}，总线序号 <operation>:= {FROMtrigger|TOTRigger} ● FROMtrigger：从触发复制。将当前触发设置同步到解码设置中 ● TOTRigger：复制到触发。将当前解码设置同步到触发设置中
示例	设置将触发器配置复制到解码总线： *:DECode:BUS1:COPY FROMtrigger* *DEC:BUS1:COPY FROM*

:DECode:BUS<n>:FORMat

描述	设置或查询解码总线的数据格式。
命令格式	:DECode:BUS<n>:FORMat <formal> :DECode:BUS<n>:FORMat?
参数说明	<n>:= {1|2}，总线序号 <format>:= {BINary|DECimal|HEX|ASCii}
返回格式	{BINary|DECimal|HEX|ASCii}
示例	设置解码总线 1 的数据格式为 HEX： *:DECode:BUS1:FORMat HEX* *DEC:BUS1:FORM HEX* 查询解码总线 1 的数据格式： *DEC:BUS1:FORM?* 返回值： *HEX*

:DECode:BUS<n>:PROTocol

描述	设置或查询解码总线的协议。
命令格式	:DECode:BUS<n>:PROTocol <protocol> :DECode:BUS<n>:PROTocol?
参数说明	<n>:= {1|2}，总线序号

	<protocol>:= {IIC|SPI|UART|CAN|LIN|FLEXray|CANFd|IIS|M1553|SENT|MANchester|ARINC429|USB20}
返回格式	{IIC|SPI|UART|CAN|LIN|FLEXray|CANFd|IIS|M1553|SENT|MANchester|A429|USB20}
示例	设置解码总线 1 的协议为 IIC： *:DECode:BUS1:PROTocol IIC* *DEC:BUS1:PROT IIC* 查询解码总线 1 的协议： *DEC:BUS1:PROT?* 返回值： *IIC*

:DECode:BUS<n>:RESult

描述	查询解码总线的结果。当屏幕上有大量解码数据时，解码总线的显示空间有限，返回的结果也会不完整；如果需要获得完整的解码数据，请使用命令 :DECode:LIST<n>:RESult。
命令格式	:DECode:BUS<n>:RESult?
参数说明	<n>:= {1|2}，总线序号
返回格式	<ascii_text>:= ASCII 形式数据。用逗号分隔，每一帧用分号分隔并自动换行。数据内容与命令“:DECode:BUS<n>:FORMat”设置的格式有关。第一行数据是头部描述信息。
示例	查询解码总线 1 的结果： *DEC:BUS1:RES?* 返回值： *lin,sync,id,parity,data,checksum;* *0x55,0x06,0,0x54\s0x5F,0x46;*

:DECode:BUS<n>:IIC

:DECode:BUS<n>:IIC:RWBit

描述	设置或查询 IIC 总线解码结果是否包含读写位。
命令格式	:DECode:BUS<n>:IIC:RWBit <state> :DECode:BUS<n>:IIC:RWBit?
参数说明	<n>:= {1|2}，总线序号 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置 IIC 协议解码总线 1 的解码结果包含读写位： *:DECode:BUS1:IIC:RWBit ON* *DEC:BUS1:IIC:RWB ON* 查询当前 IIC 协议解码总线 1 的解码结果是否包含读写位： *DEC:BUS1:IIC:RWB?* 返回值： *ON*

:DECode:BUS<n>:IIC:SCLSource

描述	设置或查询 IIC 总线的 SCL（串行时钟线）信源。
命令格式	:DECode:BUS<n>:IIC:SCLSource <source> :DECode:BUS<n>:IIC:SCLSource?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 IIC 协议解码总线 1 的 SCL 源为 C1： *:DECode:BUS1:IIC:SCLSource C1* *DEC:BUS1:IIC:SCLS C1* 查询当前 IIC 协议解码总线 1 的 SCL 源： *DEC:BUS1:IIC:SCLS?* 返回值： *C1*

关联命令	:DECode:BUS<n>:IIC:SCLThreshold :DECode:BUS<n>:IIC:SDASource

:DECode:BUS<n>:IIC:SCLThreshold

描述	设置或查询 IIC 总线的 SCL（串行时钟线）阈值。
命令格式	:DECode:BUS<n>:IIC:SCLThreshold <value> :DECode:BUS<n>:IIC:SCLThreshold?
参数说明	<n>:= {1|2}，总线序号 <value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 IIC 协议解码总线 1 的 SCL 阈值为 1V： *:DECode:BUS1:IIC:SCLThreshold 1.00E+00* *DEC:BUS1:IIC:SCLT 1.00E+00* 查询当前 IIC 协议解码总线 1 的 SCL 阈值： *DEC:BUS1:IIC:SCLT?* 返回值： *1.00E+00*
关联命令	:DECode:BUS<n>:IIC:SCLSource

:DECode:BUS<n>:IIC:SDASource

描述	设置或查询 IIC 总线的 SDA（串行数据线）信源。
命令格式	:DECode:BUS<n>:IIC:SDASource <source> :DECode:BUS<n>:IIC:SDASource?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 IIC 协议解码总线 1 的 SDA 源为 C1： *:DECode:BUS1:IIC:SDASource C1* *DEC:BUS1:IIC:SDAS C1* 查询当前 IIC 协议解码总线 1 的 SDA 源： *DEC:BUS1:IIC:SDAS?* 返回值： *C1*
关联命令	:DECode:BUS<n>:IIC:SDAThreshold :DECode:BUS<n>:IIC:SCLSource

:DECode:BUS<n>:IIC:SDAThreshold

描述	设置或查询 IIC 总线的 SDA（串行数据线）阈值。
命令格式	:DECode:BUS<n>:IIC:SDAThreshold <value> :DECode:BUS<n>:IIC:SDAThreshold?
参数说明	<n>:= {1|2}，总线序号 <value>:= 浮点型 NR3 格式。因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]

	SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式	
示例	设置 IIC 协议解码总线 1 的 SDA 阈值为 1V: *:DECode:BUS1:IIC:SDAThreshold 1.00E+00* *DEC:BUS1:IIC:SDAT 1.00E+00* 查询当前 IIC 协议解码总线 1 的 SDA 阈值: *DEC:BUS1:IIC:SDAT?* 返回值: *1.00E+00*	
关联命令	:DECode:BUS<n>:IIC:SDASource	

:DECode:BUS<n>:SPI

:DECode:BUS<n>:SPI:BITorder

描述	设置或查询 SPI 总线的比特流格式。
命令格式	:DECode:BUS<n>:SPI:BITorder <order> :DECode:BUS<n>:SPI:BITorder?
参数说明	<n>:= {1|2}，总线序号 <order>:= {LSB|MSB} ● LSB：最低有效位在先 ● MSB：最高有效位在先
返回格式	{LSB|MSB}
示例	设置 SPI 协议解码总线 1 的比特流格式为 LSB: *:DECode:BUS1:SPI:BITorder LSB* *DEC:BUS1:SPI:BIT LSB* 查询当前 SPI 协议解码总线 1 的比特流格式: *DEC:BUS1:SPI:BIT?* 返回值: *LSB*

:DECode:BUS<n>:SPI:CLKSource

描述	设置或查询 SPI 总线的 CLK（时钟）信源。
命令格式	:DECode:BUS<n>:SPI:CLKSource <source> :DECode:BUS<n>:SPI:CLKSource?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 SPI 协议解码总线 1 的 CLK 信源为 C1： *:DECode:BUS1:SPI:CLKSource C1* *DEC:BUS1:SPI:CLKS C1* 查询当前 SPI 协议解码总线 1 的 CLK 信源： *DEC:BUS1:SPI:CLKS?* 返回值： *C1*
关联命令	:DECode:BUS<n>:SPI:CLKThreshold

:DECode:BUS<n>:SPI:CLKThreshold

描述	设置或查询 SPI 总线的 CLK（时钟）阈值。
命令格式	:DECode:BUS<n>:SPI:CLKThreshold <value> :DECode:BUS<n>:SPI:CLKThreshold?
参数说明	<n>:= {1|2}，总线序号 <value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]

	SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式	
示例	设置 SPI 协议解码总线 1 的 CLK 阈值为 1V： *:DECode:BUS1:SPI:CLKThreshold 1.00E+00* *DEC:BUS1:SPI:CLKT 1.00E+00* 查询当前 SPI 协议解码总线 1 的 CLK 阈值： *DEC:BUS1:SPI:CLKT?* 返回值： *1.00E+00*	
关联命令	:DECode:BUS<n>:SPI:CLKSource	

:DECode:BUS<n>:SPI:CSSource

描述	设置或查询 SPI 总线的 CS（片选）信源。
命令格式	:DECode:BUS<n>:SPI:CSSource <source> :DECode:BUS<n>:SPI:CSSource?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 SPI 协议解码总线 1 的 CS 信源为 C1： *:DECode:BUS1:SPI:CSSource C1* *DEC:BUS1:SPI:CSS C1* 查询当前 SPI 协议解码总线 1 的 CS 信源： *DEC:BUS1:SPI:CSS?* 返回值： *C1*

:DECode:BUS<n>:SPI:CSThreshold

描述	设置或查询 SPI 总线的 CS（片选）阈值。
命令格式	:DECode:BUS<n>:SPI:CSThreshold <value> :DECode:BUS<n>:SPI:CSThreshold?
参数说明	<n>:= {1|2}，总线序号 <value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 SPI 协议解码总线 1 的 CS 阈值为 1V： *:DECode:BUS1:SPI:CSThreshold 1.00E+00* *DEC:BUS1:SPI:CST 1.00E+00* 查询当前 SPI 协议解码总线 1 的 CS 阈值： *DEC:BUS1:SPI:CST?* 返回值： *1.00E+00*
关联命令	:DECode:BUS<n>:SPI:CLKSource

:DECode:BUS<n>:SPI:CSTYpe

描述	设置或查询 SPI 总线的片选类型。
命令格式	:DECode:BUS<n>:SPI:CSTYpe <type> :DECode:BUS<n>:SPI:CSTYpe?

参数说明	<n>:={1|2}，总线序号 <type>:={NCS|CS|TIMeout[,<time>]} ● CS：高有效。片选为高时成帧，此时需要为片选信号指定源和阈值电平。此时，片选信号需要在屏幕内有完整的上升沿才视为有效。 ● NCS：低有效。片选为低时成帧，此时需要为片选信号指定源和阈值电平。此时，片选信号需要在屏幕内有完整的下降沿才视为有效。 ● TIMeout：时钟超时。此时不需要为片选信号指定源和阈值电平，但要指定<time>。此时间为示波器搜索到将要触发的数据码型前，时钟信号必须为空闲状态（不跳变）的最小时间。该设置适用于无片选信号的 SPI 总线，或通道数不够的情况（如两通道示波器）。 <time>:= 数值，浮点型 NR3 格式，该值范围为 [1.00E-07,5.00E-03]。
返回格式	{NCS|CS|TIMeout[,<time>]}
示例	设置 SPI 协议解码总线 1 的片选类型为 CS： *:DECode:BUS1:SPI:CSTYpe CS* *DEC:BUS1:SPI:CSTY CS* 查询当前 SPI 协议解码总线 1 的片选类型： *DEC:BUS1:SPI:CSTY?* 返回值： *CS*

:DECode:BUS<n>:SPI:DLENgth

描述	设置或查询 SPI 总线的数据位宽。
命令格式	:DECode:BUS<n>:SPI:DLENgth <value> :DECode:BUS<n>:SPI:DLENgth?
参数说明	<n>:={1|2}，总线序号 <value>:= 整型 NR1 格式，该值范围为 [4,32]
返回格式	整型 NR1 格式
示例	设置 SPI 协议解码总线 1 的数据位宽为 5： *:DECode:BUS1:SPI:DLENgth 5* *DEC:BUS1:SPI:DLEN 5* 查询当前 SPI 协议解码总线 1 的数据位宽： *DEC:BUS1:SPI:DLEN?* 返回值： *5*

:DECode:BUS<n>:SPI:LATChedge

描述	设置或查询 SPI 总线的 CLK（时钟）采样边沿。
命令格式	:DECode:BUS<n>:SPI:LATChedge <slope> :DECode:BUS<n>:SPI:LATChedge?
参数说明	<n>:= {1|2}，总线序号 <slope>:= {RISing|FALLing}
返回格式	{RISing|FALLing}
示例	设置 SPI 协议解码总线 1 的 CLK 采样边沿为上升沿： *:DECode:BUS1:SPI:LATChedge RISing* *DEC:BUS1:SPI:LATC RIS* 查询 SPI 协议的解码总线 1 的 CLK 采样边沿： *DEC:BUS1:SPI:LATC?* 返回值： *RISing*

:DECode:BUS<n>:SPI:MISOSource

描述	设置或查询 SPI 总线的 MISO 信源。
命令格式	:DECode:BUS<n>:SPI:MISOSource <source> :DECode:BUS<n>:SPI:MISOSource?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>|DIS} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式 ● DIS:= 禁用
返回格式	{C<n>|D<d>|DIS}
示例	设置 SPI 协议解码总线 1 的 MISO 信源为 C1： *:DECode:BUS1:SPI:MISOSource C1* *DEC:BUS1:SPI:MISOS C1* 查询当前 SPI 协议解码总线 1 的 MISO 信源： *DEC:BUS1:SPI:MISOS?* 返回值： *C1*
关联命令	:DECode:BUS<n>:SPI:MISOThreshold

:DECode:BUS<n>:SPI:MISOThreshold

描述	设置或查询 SPI 总线的 MISO 阈值。
命令格式	:DECode:BUS<n>:SPI:MISOThreshold <value> :DECode:BUS<n>:SPI:MISOThreshold?
参数说明	<n>:= {1|2}，总线序号 <value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 SPI 协议解码总线 1 的 MISO 阈值为 1V： *:DECode:BUS1:SPI:MISOThreshold 1.00E+00* *DEC:BUS1:SPI:MISOT 1.00E+00* 查询当前 SPI 协议解码总线 1 的 MISO 阈值： *DEC:BUS1:SPI:MISOT?* 返回值： *1.00E+00*
关联命令	:DECode:BUS<n>:SPI:MISOSource

:DECode:BUS<n>:SPI:MOSISource

描述	设置或查询 SPI 总线的 MOSI 信源。
命令格式	:DECode:BUS<n>:SPI:MOSISource <source> :DECode:BUS<n>:SPI:MOSISource?

参数说明	<n>:={1|2}，总线序号 <source>:={C<n>|D<d>|DIS} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式 ● DIS:= 禁用
返回格式	{C<n>|D<d>|DIS}
示例	设置 SPI 协议解码总线 1 的 MOSI 信源为 C1： *:DECode:BUS1:SPI:MOSISource C1* *DEC:BUS1:SPI:MOSIS C1* 查询当前 SPI 协议解码总线 1 的 MOSI 信源： *DEC:BUS1:SPI:MOSIS?* 返回值： *C1*
关联命令	:DECode:BUS<n>:SPI:MOSIThreshold

:DECode:BUS<n>:SPI:MOSIThreshold

描述	设置或查询 SPI 总线的 MOSI 阈值。
命令格式	:DECode:BUS<n>:SPI:MOSIThreshold <value> :DECode:BUS<n>:SPI:MOSIThreshold?
参数说明	<n>:={1|2}，总线序号 <value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 SPI 协议解码总线 1 的 MOSI 阈值为 1V： *:DECode:BUS1:SPI:MOSIThreshold 1.00E+00* *DEC:BUS1:SPI:MOSIT 1.00E+00* 查询当前 SPI 协议解码总线 1 的 MOSI 阈值： *DEC:BUS1:SPI:MOSIT?* 返回值： *1.00E+00*
关联命令	:DECode:BUS<n>:SPI:MOSISource

:DECode:BUS<n>:SPI:NCSSource

描述	设置或查询 SPI 总线的 NCS 信源。
命令格式	:DECode:BUS<n>:SPI:NCSSource <source> :DECode:BUS<n>:SPI:NCSSource?
参数说明	<n>:={1|2}，总线序号 <source>:={C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 SPI 协议解码总线 1 的 NCS 信源为 C1： *:DECode:BUS1:SPI:NCSSource C1* *DEC:BUS1:SPI:NCSS C1* 查询当前 SPI 协议解码总线 1 的 NCS 信源： *DEC:BUS1:SPI:NCSS?* 返回值： *C1*
关联命令	:DECode:BUS<n>:SPI:NCSThreshold

:DECode:BUS<n>:SPI:NCSThreshold

描述	设置或查询 SPI 总线的 NCS 阈值。
命令格式	:DECode:BUS<n>:SPI:NCSThreshold <value> :DECode:BUS<n>:SPI:NCSThreshold?
参数说明	<n>:={1|2}，总线序号

<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式 浮点型 NR3 格式

示例

设置 SPI 协议解码总线 1 的 NCS 阈值为 1V：

:DECode:BUS1:SPI:NCSThreshold 1.00E+00

DEC:BUS1:SPI:NCST 1.00E+00

查询当前 SPI 协议解码总线 1 的 NCS 阈值：

DEC:BUS1:SPI:NCST?

返回值：

1.00E+00

关联命令 :DECode:BUS<n>:SPI:NCSSource

:DECode:BUS<n>:UART

:DECode:BUS<n>:UART:BAUD

描述	设置或查询 UART 总线的波特率。
命令格式	:DECode:BUS<n>:UART:BAUD <baud> :DECode:BUS<n>:UART:BAUD?
参数说明	<n>:= {1|2}，总线序号 <baud>:= {600bps|1200bps|2400bps|4800bps|9600bps|19200bps|38400bps| 57600bps|115200bps|CUSTom[,<value>]} <value>:= 整型 NR1 格式，该值范围为 [300,20000000]
返回格式	整型 NR1 格式
示例	设置 UART 协议解码总线 1 的波特率为 9600bps： *:DECode:BUS1:UART:BAUD 9600bps* *DEC:BUS1:UART:BAUD 9600bps* 查询当前 UART 协议解码总线 1 的波特率： *DEC:BUS1:UART:BAUD?* 返回值： *9600bps*

:DECode:BUS<n>:UART:BITorder

描述	设置或查询 UART 总线的比特流格式。
命令格式	:DECode:BUS<n>:UART:BITorder <order> :DECode:BUS<n>:UART:BITorder?
参数说明	<n>:= {1|2}，总线序号 <order>:= {LSB|MSB} ● LSB：最低有效位在先 ● MSB：最高有效位在先
返回格式	{LSB|MSB}
示例	设置 UART 协议解码总线 1 的比特流格式为 LSB： *:DECode:BUS1:UART:BITorder LSB* *DEC:BUS1:UART:BIT LSB*

	查询当前 UART 协议解码总线 1 的比特流格式： *DEC:BUS1:UART:BIT?* 返回值： *LSB*

:DECode:BUS<n>:UART:DLENgth

描述	设置或查询 UART 总线的数据长度。
命令格式	:DECode:BUS<n>:UART:DLENgth <value> :DECode:BUS<n>:UART:DLENgth?
参数说明	<n>:= {1|2}，总线序号 <value>:= 整型 NR1 格式，该值范围为 [5,8]
返回格式	整型 NR1 格式
示例	设置 UART 协议解码总线 1 的数据长度为 5： *:DECode:BUS1:UART:DLENgth 5* *DEC:BUS1:UART:DLEN 5* 查询当前 UART 协议解码总线 1 的数据长度： *DEC:BUS1:UART:DLEN?* 返回值： *5*

:DECode:BUS<n>:UART:IDLE

描述	设置或查询 UART 总线的空闲电平。
命令格式	:DECode:BUS<n>:UART:IDLE <idle> :DECode:BUS<n>:UART:IDLE?
参数说明	<n>:= {1|2}，总线序号 <idle>:= {LOW|HIGH} ● LOW：空闲电平为低电平 ● HIGH：空闲电平为高电平
返回格式	{LOW|HIGH}
示例	设置 UART 协议解码总线 1 的空闲电平为低电平： *:DECode:BUS1:UART:IDLE LOW* *DEC:BUS1:UART:IDLE LOW*

	查询当前 UART 协议解码总线 1 的空闲电平： *DEC:BUS1:UART:IDLE?* 返回值： *LOW*

:DECode:BUS<n>:UART:PARity

描述	设置或查询 UART 总线的奇偶校验。
命令格式	:DECode:BUS<n>:UART:PARity <parity> :DECode:BUS<n>:UART:PARity?
参数说明	<n>:= {1|2}，总线序号 <parity>:= {NONE|ODD|EVEN|MARK|SPACe} ● NONE：无校验 ● ODD：奇校验 ● EVEN：偶校验 ● MARK：1 校验 ● SPACe：0 校验（UART 数据为 9 位时，第 9 位可视为 1 校验或 0 校验）
返回格式	{NONE|ODD|EVEN|MARK|SPACe}
示例	设置 UART 协议解码总线 1 的奇偶校验为无： *:DECode:BUS1:UART:PARity NONE* *DEC:BUS1:UART:PAR NONE* 查询当前 UART 协议解码总线 1 的奇偶校验： *DEC:BUS1:UART:PAR?* 返回值： *NONE*

:DECode:BUS<n>:UART:RXSource

描述	设置或查询 UART 总线的 RX 信源。
命令格式	:DECode:BUS<n>:UART:RXSource <source> :DECode:BUS<n>:UART:RXSource?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>|DIS} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式 ● DIS:= 禁用

返回格式	{C<n>|D<d>|DIS}
示例	设置 UART 协议解码总线 1 的 RX 信源为 C1： *:DECode:BUS1:UART:RXSource C1* *DEC:BUS1:UART:RXS C1* 查询当前 UART 协议解码总线 1 的 RX 信源： *DEC:BUS1:UART:RXS?* 返回值： *C1*

:DECode:BUS<n>:UART:RXTHreshold

描述	设置或查询 UART 总线的 RX 阈值。
命令格式	:DECode:BUS<n>:UART:RXTHreshold <value> :DECode:BUS<n>:UART:RXTHreshold?
参数说明	<n>:= {1|2}，总线序号 <value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 UART 协议解码总线 1 的 RX 阈值为 1V： *:DECode:BUS1:UART:RXTHreshold 1.00E+00* *DEC:BUS1:UART:RXTH 1.00E+00* 查询当前 UART 协议解码总线 1 的 RX 阈值： *DEC:BUS1:UART:RXTH?* 返回值：

	1.00E+00
关联命令	:DECode:BUS<n>:UART:RXSource

:DECode:BUS<n>:UART:STOP

描述	设置或查询 UART 总线的停止位数。
命令格式	:DECode:BUS<n>:UART:STOP <bit> :DECode:BUS<n>:UART:STOP?
参数说明	<n>:= {1|2}，总线序号 <bit>:= {1|1.5|2}
返回格式	{1|1.5|2}
示例	设置 UART 协议解码总线 1 停止位数为 1： *:DECode:BUS1:UART:STOP 1* *DEC:BUS1:UART:STOP 1* 查询当前 UART 协议解码总线 1 的停止位数： *DEC:BUS1:UART:STOP?* 返回值： *1*

:DECode:BUS<n>:UART:TXSource

描述	设置或查询 UART 总线的 TX 信源。
命令格式	:DECode:BUS<n>:UART:TXSource <source> :DECode:BUS<n>:UART:TXSource?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>|DIS} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式 ● DIS:= 禁用
返回格式	{C<n>|D<d>|DIS}
示例	设置 UART 协议解码总线 1 的 TX 信源为 C1： *:DECode:BUS1:UART:TXSource C1* *DEC:BUS1:UART:TXS C1* 查询当前 UART 协议解码总线 1 的 TX 信源：

	DEC:BUS1:UART:TXS? 返回值： *C1*
关联命令	:DECode:BUS<n>:UART:

:DECode:BUS<n>:UART:TXTHreshold

描述	设置或查询 UART 总线的 TX 阈值。
命令格式	:DECode:BUS<n>:UART:TXTHreshold <value> :DECode:BUS<n>:UART:TXTHreshold?
参数说明	<n>:= {1|2}，总线序号 <value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 UART 协议解码总线 1 的 TX 阈值为 1V： *:DECode:BUS1:UART:TXTHreshold 1.00E+00* *DEC:BUS1:UART:TXTH 1.00E+00* 查询当前 UART 协议解码总线 1 的 TX 阈值： *DEC:BUS1:UART:TXTH?* 返回值： *1.00E+00*
关联命令	:DECode:BUS<n>:UART:TXSource

:DECode:BUS<n>:CAN

:DECode:BUS<n>:CAN:BAUD

描述	设置或查询 CAN 总线的波特率。
命令格式	:DECode:BUS<n>:CAN:BAUD <baud> :DECode:BUS<n>:CAN:BAUD?
参数说明	<n>:= {1|2}，总线序号 <baud>:= {5kbps|10kbps|20kbps|50kbps|100kbps|125kbps|250kbps|500kbps|800kbps|1Mbps|CUSTom[,<value>]} <value>:= 整型 NR1 格式，该值范围为 [5000,1000000]
返回格式	整型 NR1 格式
示例	设置 CAN 协议解码总线 1 的波特率为 10kbps： *:DECode:BUS1:CAN:BAUD 10kbps* *DEC:BUS1:CAN:BAUD 10kbps* 查询当前 CAN 协议解码总线 1 的波特率： *DEC:BUS1:CAN:BAUD?* 返回值： *10kbps*

:DECode:BUS<n>:CAN:CANLSource

描述	设置或查询 CAN 总线的 CAN_L 信源。
命令格式	:DECode:BUS<n>:CAN:CANLSource <source> :DECode:BUS<n>:CAN:CANLSource?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 CAN 协议解码总线 1 的 CAN_L 信源为 C1： *:DECode:BUS1:CAN:CANLSource C1* *DEC:BUS1:CAN:CANLS C1* 查询当前 CAN 协议解码总线 1 的信源：

	DEC:BUS1:CAN:CANLS? 返回值： *C1*
关联命令	:DECode:BUS<n>:CAN:CANLThreshold

:DECode:BUS<n>:CAN:CANLThreshold

描述	设置或查询 CAN 总线的 CAN_L 阈值。
命令格式	:DECode:BUS<n>:CAN:CANLThreshold <value> :DECode:BUS<n>:CAN:CANLThreshold?
参数说明	<n>:= {1|2}，总线序号 <value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 CAN 协议解码总线 1 的 CAN_L 阈值为 1V： *:DECode:BUS1:CAN:CANLThreshold 1.00E+00* *DEC:BUS1:CAN:CANLT 1.00E+00* 查询当前 CAN 协议解码总线 1 的 CAN_L 阈值： *DEC:BUS1:CAN:CANLT?* 返回值： *1.00E+00*
关联命令	:DECode:BUS<n>:CAN:CANLSource

:DECode:BUS<n>:CAN:SIGNal

描述	设置或查询 CAN 总线的信号类型。
命令格式	:DECode:BUS<n>:CAN:SIGNal <type> :DECode:BUS<n>:CAN:SIGNal?
参数说明	<n>:= {1|2}，总线序号 <type>:= {CANH|CANL|DIFH} ● CANH：差分总线的 CAN_H 信号 ● CANL：差分总线的 CAN_L 信号 ● DIFH：差分信号 CAN_H-CAN_L
返回格式	{CANH|CANL|DIFH}
示例	设置 CAN 协议解码总线 1 的信号类型为 CAN_H： *:DECode:BUS1:CAN:SIGNal CANH* *DEC:BUS1:CAN:SIGN CANH* 查询当前 CAN 协议解码总线 1 的信源： *DEC:BUS1:CAN:SIGN?* 返回值： *CANH*
关联命令	:DECode:BUS<n>:CAN:CANLSource :DECode:BUS<n>:CAN:SOURce

:DECode:BUS<n>:CAN:SOURce

描述	设置或查询 CAN 总线的 CAN_H 信源。
命令格式	:DECode:BUS<n>:CAN:SOURce <source> :DECode:BUS<n>:CAN:SOURce?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 CAN 协议解码总线 1 的 CAN_H 信源为 C1： *:DECode:BUS1:CAN:SOURce C1* *DEC:BUS1:CAN:SOUR C1* 查询当前 CAN 协议解码总线 1 的 CAN_H 信源： *DEC:BUS1:CAN:SOUR?* 返回值：

	C1
关联命令	:DECode:BUS<n>:CAN:THReshold

:DECode:BUS<n>:CAN:THReshold

描述	设置或查询 CAN 总线的 CAN_H 阈值。
命令格式	:DECode:BUS<n>:CAN:THReshold <value> :DECode:BUS<n>:CAN:THReshold?
参数说明	<n>:= {1|2}，总线序号 <value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD / [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X / [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD / [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置 CAN 协议解码总线 1 的 CAN_H 阈值为 1V： *:DECode:BUS1:CAN:THReshold 1.00E+00* *DEC:BUS1:CAN:THR 1.00E+00* 查询当前 CAN 协议解码总线 1 的 CAN_H 阈值： *DEC:BUS1:CAN:THR?* 返回值： *1.00E+00*
关联命令	:DECode:BUS<n>:CAN:SOURce

:DECode:BUS<n>:LIN

:DECode:BUS<n>:LIN:BAUD

描述	设置或查询 LIN 总线的波特率。
命令格式	:DECode:BUS<n>:LIN:BAUD <baud> :DECode:BUS<n>:LIN:BAUD?
参数说明	<n>:= {1|2}，总线序号 <baud>:= {600bps|1200bps|2400bps|4800bps|9600bps|19200bps|CUSTom[,<value>]} <value>:= 整型 NR1 格式，该值范围为 [300,20000000]
返回格式	整型 NR1 格式
示例	设置 LIN 协议解码总线 1 的波特率为 9600bps： *:DECode:BUS1:LIN:BAUD 9600bps* *DEC:BUS1:LIN:BAUD 9600bps* 查询当前 LIN 协议解码总线 1 的波特率： *DEC:BUS1:LIN:BAUD?* 返回值： *9600bps*

:DECode:BUS<n>:LIN:SOURce

描述	设置或查询 LIN 总线的信源。
命令格式	:DECode:BUS<n>:LIN:SOURce <source> :DECode:BUS<n>:LIN:SOURce?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 LIN 协议解码总线 1 的信源为 C1： *:DECode:BUS1:LIN:SOURce C1* *DEC:BUS1:LIN:SOUR C1* 查询当前 LIN 协议解码总线 1 的信源：

	DEC:BUS1:LIN:SOUR? 返回值： *C1*
关联命令	:DECode:BUS<n>:LIN:THReshold

:DECode:BUS<n>:LIN:THReshold

描述	设置或查询 LIN 总线的阈值。
命令格式	:DECode:BUS<n>:LIN:THReshold <value> :DECode:BUS<n>:LIN:THReshold?
参数说明	<n>:= {1|2}，总线序号 <value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 LIN 协议解码总线 1 的阈值为 1V： *:DECode:BUS1:LIN:THReshold 1.00E+00* *DEC:BUS1:LIN:THR 1.00E+00* 查询当前 LIN 协议解码总线 1 的阈值： *DEC:BUS1:LIN:THR?* 返回值： *1.00E+00*
关联命令	:DECode:BUS<n>:LIN:SOURce

:DECode:BUS<n>:FLEXray

:DECode:BUS<n>:FLEXray:BAUD

描述	设置或查询 FlexRay 总线的波特率。
命令格式	:DECode:BUS<n>:FLEXray:BAUD <baud> :DECode:BUS<n>:FLEXray:BAUD?
参数说明	<n>:= {1|2}，总线序号 <baud>:= {2500kbps|5Mbps|10Mbps|CUSTom[,<value>]} <value>:= 整型 NR1 格式，该值范围为 [1000000,20000000]
返回格式	整型 NR1 格式
示例	设置 FlexRay 协议解码总线 1 的波特率为 5Mbps： *:DECode:BUS1:FLEXray:BAUD 5Mbps* *DEC:BUS1:FLEX:BAUD 5Mbps* 查询当前 FlexRay 协议解码总线 1 的波特率： *DEC:BUS1:FLEX:BAUD?* 返回值： *5Mbps*

:DECode:BUS<n>:FLEXray:SOURce

描述	设置或查询 FlexRay 总线的信源。
命令格式	:DECode:BUS<n>:FLEXray:SOURce <source> :DECode:BUS<n>:FLEXray:SOURce?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 FlexRay 协议解码总线 1 的信源为 C1： *:DECode:BUS1:FLEXray:SOURce C1* *DEC:BUS1:FLEX:SOUR C1* 查询当前 LIN 协议解码总线 1 的信源： *DEC:BUS1:FLEX:SOUR?*

	返回值： *C1*
关联命令	:DECode:BUS<n>:FLEXray:THReshold

:DECode:BUS<n>:FLEXray:THReshold

描述	设置或查询 FlexRay 总线的阈值。
命令格式	:DECode:BUS<n>:FLEXray:THReshold <value> :DECode:BUS<n>:FLEXray:THReshold?
参数说明	<n>:= {1|2}，总线序号 <value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 FlexRay 协议解码总线 1 的阈值为 1V： *:DECode:BUS1:FLEXray:THReshold 1.00E+00* *DEC:BUS1:FLEX:THR 1.00E+00* 查询当前 FlexRay 协议解码总线 1 的阈值： *DEC:BUS1:FLEX:THR?* 返回值： *1.00E+00*
关联命令	:DECode:BUS<n>:FLEXray:SOURce

:DECode:BUS<n>:CANFd

:DECode:BUS<n>:CANFd:BAUDData

描述	设置或查询 CAN FD 总线的数据波特率。
命令格式	:DECode:BUS<n>:CANFd:BAUDData <baud> :DECode:BUS<n>:CANFd:BAUDData?
参数说明	<n>:= {1|2}，总线序号 <baud>:= {500kbps|1Mbps|2Mbps|5Mbps|8Mbps|10Mbps|CUSTom[,<value>]} <value>:= 整型 NR1 格式，该值范围为 [100000,10000000]
返回格式	整型 NR1 格式
示例	设置 CAN FD 协议解码总线 1 的数据波特率为 500kbps: *:DECode:BUS1:CANFd:BAUDData 500kbps* *DEC:BUS1:CANF:BAUDD 500kbps* 查询当前 CAN FD 协议解码总线 1 的数据波特率: *DEC:BUS1:CANF:BAUDD?* 返回值: *500kbps*

:DECode:BUS<n>:CANFd:BAUDNominal

描述	设置或查询 CAN FD 总线的标准波特率。
命令格式	:DECode:BUS<n>:CANFd:BAUDNominal <baud> :DECode:BUS<n>:CANFd:BAUDNominal?
参数说明	<n>:= {1|2}，总线序号 <baud>:= {10kbps|25kbps|50kbps|100kbps|250kbps|1Mbps|CUSTom[,<value>]} <value>:= 整型 NR1 格式，该值范围为 [10000,1000000]
返回格式	整型 NR1 格式
示例	设置 CAN FD 协议解码总线 1 的标准波特率为 50kbps: *:DECode:BUS1:CANFd:BAUDNominal 50kbps* *DEC:BUS1:CANF:BAUDN 50kbps* 查询当前 CAN FD 协议解码总线 1 的标准波特率: *DEC:BUS1:CANF:BAUDN?*

	返回值： *50kbps*

:DECode:BUS<n>:CANFd:SOURce

描述	设置或查询 CAN FD 总线的信源。
命令格式	:DECode:BUS<n>:CANFd:SOURce <source> :DECode:BUS<n>:CANFd:SOURce?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 CAN FD 协议解码总线 1 的信源为 C1： *:DECode:BUS1:CANFd:SOURce C1* *DEC:BUS1:CANF:SOUR C1* 查询当前 CAN FD 协议解码总线 1 的信源： *DEC:BUS1:CANF:SOUR?* 返回值： *C1*
关联命令	:DECode:BUS<n>:CANFd:THReshold

:DECode:BUS<n>:CANFd:THReshold

描述	设置或查询 CAN FD 总线的阈值。
命令格式	:DECode:BUS<n>:CANFd:THReshold <value> :DECode:BUS<n>:CANFd:THReshold?
参数说明	<n>:= {1|2}，总线序号 <value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]

	SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式	
示例	设置 CAN FD 协议解码总线 1 阈值为 1V： *:DECode:BUS1:CANFd:THReshold 1.00E+00* *DEC:BUS1:CANF:THR 1.00E+00* 查询当前 CAN FD 协议解码总线 1 阈值： *DEC:BUS1:CANF:THR?* 返回值： *1.00E+00*	
关联命令	:DECode:BUS<n>:CANFd:SOURce	

:DECode:BUS<n>:IIS

:DECode:BUS<n>:IIS:ANNotate

描述	设置或查询 IIS 总线的注释通道。
命令格式	:DECode:BUS<n>:IIS:ANNotate <type> :DECode:BUS<n>:IIS:ANNotate?
参数说明	<n>:= {1|2}，总线序号 <type>:= {ALL|LEFT|RIGHt}
返回格式	{ALL|LEFT|RIGHt}
示例	设置 IIS 协议解码总线 1 的注释通道为所有通道： *:DECode:BUS1:IIS:ANNotate ALL* *DEC:BUS1:IIS:ANN ALL* 查询当前 IIS 协议解码总线 1 的注释通道： *DEC:BUS1:IIS:ANN?* 返回值： *ALL*
关联命令	:DECode:BUS<n>:IIS:LCH

:DECode:BUS<n>:IIS:AVARiant

描述	设置或查询 IIS 总线的音频格式。
命令格式	:DECode:BUS<n>:IIS:AVARiant <type> :DECode:BUS<n>:IIS:AVARiant?
参数说明	<n>:= {1|2}，总线序号 <type>:= {I2S|LJ|RJ} ● I2S：音频格式为 Audio-I2S ● LJ：音频格式为 Audio-LJ ● RJ：音频格式为 Audio-RJ
返回格式	{I2S|LJ|RJ}
示例	设置 IIS 协议解码总线 1 的音频格式为 RJ： *:DECode:BUS1:IIS:AVARiant RJ* *DEC:BUS1:IIS:AVAR RJ* 查询当前 IIS 协议解码总线 1 的音频格式： *DEC:BUS1:IIS:AVAR?* 返回值： *RJ*

:DECode:BUS<n>:IIS:BCLKSource

描述	设置或查询 IIS 总线的时钟（BCLK）源。
命令格式	:DECode:BUS<n>:IIS:BCLKSource <source> :DECode:BUS<n>:IIS:BCLKSource?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 IIS 协议解码总线 1 的时钟源为 C1： *:DECode:BUS1:IIS:BCLKSource C1* *DEC:BUS1:IIS:BCLKS C1* 查询当前 IIS 协议解码总线 1 的时钟源： *DEC:BUS1:IIS:BCLKS?* 返回值： *C1*
关联命令	:DECode:BUS<n>:IIS:BCLKThreshold

:DECode:BUS<n>:IIS:BCLKThreshold

描述	设置或查询 IIS 总线的时钟（BCLK）源阈值。
命令格式	:DECode:BUS<n>:IIS:BCLKThreshold <value> :DECode:BUS<n>:IIS:BCLKThreshold?
参数说明	<n>:= {1|2}，总线序号 value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD / [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L / [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD / [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置 IIS 协议解码总线 1 的时钟源阈值为 1V： *:DECode:BUS1:IIS:BCLKThreshold 1.00E+00* *DEC:BUS1:IIS:BCLKT 1.00E+00* 查询当前 IIS 协议解码总线 1 的时钟源阈值： *DEC:BUS1:IIS:BCLKT?* 返回值： *1.00E+00*
关联命令	:DECode:BUS<n>:IIS:BCLKSource

:DECode:BUS<n>:IIS:BITorder

描述	设置或查询 IIS 总线的比特流格式。
命令格式	:DECode:BUS<n>:IIS:BITorder <order> :DECode:BUS<n>:IIS:BITorder?
参数说明	<n>:= {1|2}，总线序号 <order>:= {LSB|MSB} ● LSB：最低有效位在先 ● MSB：最高有效位在先

返回格式	{LSB|MSB}
示例	设置 IIS 协议解码总线 1 的比特流格式为 LSB： *:DECode:BUS1:IIS:BITorder LSB* *DEC:BUS1:IIS:BIT LSB* 查询当前 IIS 协议解码总线 1 的比特流格式： *DEC:BUS1:IIS:BIT?* 返回值： *LSB*

:DECode:BUS<n>:IIS:DLENgth

描述	设置或查询 IIS 总线的数据位数。
命令格式	:DECode:BUS<n>:IIS:DLENgth <value> :DECode:BUS<n>:IIS:DLENgth?
参数说明	<n>:= {1|2}，总线序号 <value>:= 数据位数，整型 NR1 格式，该值范围为 [1,32]
返回格式	整型 NR1 格式
示例	设置 IIS 协议解码总线 1 的数据位数为 5： *:DECode:BUS1:IIS:DLENgth 5* *DEC:BUS1:IIS:DLEN 5* 查询当前 IIS 协议解码总线 1 的数据位数： *DEC:BUS1:IIS:DLEN?* 返回值： *5*
关联命令	:DECode:BUS<n>:IIS:SBIT

:DECode:BUS<n>:IIS:DSource

描述	设置或查询 IIS 总线的数据源。
命令格式	:DECode:BUS<n>:IIS:DSource <source> :DECode:BUS<n>:IIS:DSource?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}

示例	设置 IIS 协议解码总线 1 的数据源为 C1： *:DECode:BUS1:IIS:DSource C1* *DEC:BUS1:IIS:DS C1* 查询当前 IIS 协议解码总线 1 的数据源： *DEC:BUS1:IIS:DS?* 返回值： *C1*
关联命令	:DECode:BUS<n>:IIS:DTHReshold

:DECode:BUS<n>:IIS:DTHReshold

描述	设置或查询 IIS 总线的数据源阈值。
命令格式	:DECode:BUS<n>:IIS:DTHReshold <value> :DECode:BUS<n>:IIS:DTHReshold?
参数说明	<n>:= {1|2}，总线序号 value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 IIS 协议解码总线 1 的数据源阈值为 1V： *:DECode:BUS1:IIS:DTHReshold 1.00E+00* *DEC:BUS1:IIS:DTHR 1.00E+00* 查询当前 IIS 协议解码总线 1 的数据源阈值： *DEC:BUS1:IIS:DTHR?* 返回值： *1.00E+00*
关联命令	:DECode:BUS<n>:IIS:

:DECode:BUS<n>:IIS:LATChedge

描述	设置或查询 IIS 总线的时钟采样边沿。
命令格式	:DECode:BUS<n>:IIS:LATChedge <slope> :DECode:BUS<n>:IIS:LATChedge?
参数说明	<n>:= {1|2}，总线序号 <slope>:= {RISing|FALLing} ● RISing：上升沿。数据在时钟上升沿锁存 ● FALLing：下降沿。数据在时钟下降沿锁存
返回格式	{RISing|FALLing}
示例	设置 IIS 协议解码总线 1 的时钟采样边沿为上升沿： *:DECode:BUS1:IIS:LATChedge RISing* *DEC:BUS1:IIS:LATC RIS* 查询当前 IIS 协议解码总线 1 的时钟采样边沿： *DEC:BUS1:IIS:LATC?* 返回值： *RISing*

:DECode:BUS<n>:IIS:LCH

描述	设置或查询 IIS 总线的左声道极性。
命令格式	:DECode:BUS<n>:IIS:LCH <left> :DECode:BUS<n>:IIS:LCH?
参数说明	<n>:= {1|2}，总线序号 <left>:= {LOW|HIGH} ● LOW：低电平。WS 为低时选择左声道，WS 为高时选择右声道 ● HIGH：高电平。WS 为低时选择右声道，WS 为高时选择左声道
返回格式	{LOW|HIGH}
示例	设置 IIS 协议解码总线 1 的左声道极性为低电平： *:DECode:BUS1:IIS:LCH LOW* *DEC:BUS1:IIS:LCH LOW* 查询当前 IIS 协议解码总线 1 的左声道极性： *DEC:BUS1:IIS:LCH?* 返回值： *LOW*

:DECode:BUS<n>:IIS:SBIT

描述	设置或查询 IIS 总线的起始位。
命令格式	:DECode:BUS<n>:IIS:SBIT <value> :DECode:BUS<n>:IIS:SBIT?
参数说明	<n>:= {1|2}，总线序号 <value>:= 数值，整型 NR1 格式，该值范围为 [1,31]
返回格式	整型 NR1 格式
示例	设置 IIS 协议解码总线 1 的起始位为 1： *:DECode:BUS1:IIS:SBIT 1* *DEC:BUS1:IIS:SBIT 1* 查询 IIS 协议解码总线 1 的起始位： *DEC:BUS1:IIS:SBIT?* 返回值： *1*
关联命令	:DECode:BUS<n>:IIS:DLENgth

:DECode:BUS<n>:IIS:WSSource

描述	设置或查询 IIS 总线的帧时钟（WS）源。
命令格式	:DECode:BUS<n>:IIS:WSSource <source> :DECode:BUS<n>:IIS:WSSource?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 IIS 协议解码总线 1 的帧时钟源为 C1： *:DECode:BUS1:IIS:WSSource C1* *DEC:BUS1:IIS:WSS C1* 查询当前 IIS 协议解码总线 1 的帧时钟源： *DEC:BUS1:IIS:WSS?* 返回值： *C1*
关联命令	:DECode:BUS<n>:IIS:WSTHreshold

:DECode:BUS<n>:IIS:WSTHreshold

描述	设置或查询 IIS 总线的帧时钟（WS）阈值。
命令格式	:DECode:BUS<n>:IIS:WSTHreshold <value> :DECode:BUS<n>:IIS:WSTHreshold?
参数说明	<n>:={1|2}，总线序号 value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD / [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L / [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD / [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置 IIS 协议解码总线 1 的帧时钟阈值为 1V： *:DECode:BUS1:IIS:WSTHreshold 1.00E+00* *DEC:BUS1:IIS:WSTH 1.00E+00* 查询当前 IIS 协议解码总线 1 的帧时钟阈值： *DEC:BUS1:IIS:WSTH?* 返回值： *1.00E+00*
关联命令	:DECode:BUS<n>:IIS:WSSource

:DECode:BUS<n>:M1553

:DECode:BUS<n>:M1553:LTHReshold

描述	设置或查询 M1553 总线的阈值低电平。
命令格式	:DECode:BUS<n>:M1553:LTHReshold <value> :DECode:BUS<n>:M1553:LTHReshold?
参数说明	<n>:={1|2}，总线序号 <value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

注意：低电平不能大于高电平（高电平设置指令：:DECode:BUS<n>:M1553:UTHReshold）。

返回格式	浮点型 NR3 格式
示例	设置 M1553 协议解码总线 1 的阈值低电平为 1V： *:DECode:BUS1:M1553:LTHReshold 1.00E+00* *DEC:BUS1:M1553:LTHR 1.00E+00* 查询当前 M1553 协议解码总线 1 的阈值低电平： *DEC:BUS1:M1553:LTHR?* 返回值： *1.00E+00*
关联命令	:DECode:BUS<n>:M1553:SOURce :DECode:BUS<n>:M1553:UTHReshold

:DECode:BUS<n>:M1553:SOURce

描述	设置或查询 M1553 总线的信源。
命令格式	:DECode:BUS<n>:M1553:SOURce <source> :DECode:BUS<n>:M1553:SOURce?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式
返回格式	{C<n>}
示例	设置 M1553 协议解码总线 1 的信源为 C1： *:DECode:BUS1:M1553:SOURce C1* *DEC:BUS1:M1553:SOUR C1* 查询当前 M1553 协议解码总线 1 的信源：

	DEC:BUS1:M1553:SOUR? 返回值： *C1*
关联命令	:DECode:BUS<n>:M1553:LTHReshold :DECode:BUS<n>:M1553:UTHReshold

:DECode:BUS<n>:M1553:UTHReshold

描述	设置或查询 M1553 总线的阈值高电平。
命令格式	:DECode:BUS<n>:M1553:UTHReshold <value> :DECode:BUS<n>:M1553:UTHReshold?
参数说明	<n>:= {1|2}，总线序号 value>:= 阈值电平，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD / [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L / [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD / [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移] **注意**：高电平不能小于低电平（低电平设置指令：:DECode:BUS<n>:M1553:LTHReshold）。
返回格式	浮点型 NR3 格式
示例	设置 M1553 协议解码总线 1 的阈值高电平为 2V： *:DECode:BUS1:M1553:UTHReshold 2.00E+00* *DEC:BUS1:M1553:UTHR 2.00E+00* 查询当前 M1553 协议解码总线 1 的阈值高电平： *DEC:BUS1:M1553:UTHR?* 返回值： *2.00E+00*
关联命令	:DECode:BUS<n>:M1553:SOURce :DECode:BUS<n>:M1553:LTHReshold

:DECode:BUS<n>:SENT

:DECode:BUS<n>:SENT:SOURce

描述	设置或查询 SENT 总线的信源。
命令格式	:DECode:BUS<n>:SENT:SOURce <source> :DECode:BUS<n>:SENT:SOURce?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 SENT 协议解码总线 1 的信源为 C1： *:DECode:BUS1:SENT:SOURce C1* *DEC:BUS1:SENT:SOUR C1* 查询当前 SENT 协议解码总线 1 的信源： *DEC:BUS1:SENT:SOUR?* 返回值： *C1*
关联命令	:DECode:BUS<n>:SENT:THReshold

:DECode:BUS<n>:SENT:THReshold

描述	设置或查询 SENT 总线的阈值。
命令格式	:DECode:BUS<n>:SENT:THReshold <value> :DECode:BUS<n>:SENT:THReshold?
参数说明	<n>:= {1|2}，总线序号 value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 SENT 协议解码总线 1 的阈值为 1V： *:DECode:BUS1:SENT:THReshold 1.00E+00* *DEC:BUS1:SENT:THR 1.00E+00* 查询当前 SENT 协议解码总线 1 的阈值： *DEC:BUS1:SENT:THR?* 返回值： *1.00E+00*
关联命令	:DECode:BUS<n>:SENT:SOURce

:DECode:BUS<n>:SENT:FORMat

描述	设置或查询 SENT 总线的消息格式。
命令格式	:DECode:BUS<n>:SENT:FORMat <format> :DECode:BUS<n>:SENT:FORMat?
参数说明	<n>:= {1|2}，总线序号 <format>:= {NIBBles|FSIGnal|SSERial|ESERial}
返回格式	{NIBBles|FSIGnal|SSERial|ESERial}
示例	设置 SENT 协议解码总线 1 的消息格式为 NIBBles： *:DECode:BUS1:SENT:FORMat NIBBles* *DEC:BUS1:SENT:FORM NIBB* 查询 SENT 协议解码总线 1 的消息格式： *DEC:BUS1:SENT:FORM?* 返回值： *NIBBles*

:DECode:BUS<n>:SENT:CLOCk

描述	设置或查询 SENT 总线的时钟周期。
命令格式	:DECode:BUS<n>:SENT:CLOCk <value> :DECode:BUS<n>:SENT:CLOCk?
参数说明	<n>:= {1|2}，总线序号 <value>:= 数值，浮点型 NR3 格式，该值范围为 [500E-09,300E-06]
返回格式	浮点型 NR3 格式
示例	设置 SENT 协议解码总线 1 的时钟周期为 1us：

	:DECode:BUS1:SENT:CLOCk 1.00E-06 *DEC:BUS1:SENT:CLOC 1.00E-06* 查询当前 SENT 协议解码总线 1 的时钟周期： *DEC:BUS1:SENT:CLOC?* 返回值： *1.00E-06*

:DECode:BUS<n>:SENT:TOLerance

描述	设置或查询 SENT 总线时钟周期的容差。
命令格式	:DECode:BUS<n>:SENT:TOLerance <value> :DECode:BUS<n>:SENT:TOLerance?
参数说明	<n>:= {1|2}，总线序号 <value>:= 百分比数值，整型 NR1 格式，该值范围为 [1,25]
返回格式	整型 NR1 格式
示例	设置 SENT 协议解码总线 1 的容差为 5%： *:DECode:BUS1:SENT:TOLerance 5* *DEC:BUS1:SENT:TOL 5* 查询当前 SENT 协议解码总线 1 的容差： *DEC:BUS1:SENT:TOL?* 返回值： *5*

:DECode:BUS<n>:SENT:IDLE

描述	设置或查询 SENT 总线的空闲状态。
命令格式	:DECode:BUS<n>:SENT:IDLE <idle> :DECode:BUS<n>:SENT:IDLE?
参数说明	<n>:= {1|2}，总线序号 <idle>:= {LOW|HIGH} ● LOW：低电平 ● LOW：高电平
返回格式	{LOW|HIGH}
示例	设置 SENT 协议解码总线 1 的空闲状态为低电平： *:DECode:BUS1:SENT:IDLE LOW*

	DEC:BUS1:SENT:IDLE LOW 查询当前 SENT 协议解码总线 1 的空闲状态： *DEC:BUS1:SENT:IDLE?* 返回值： *LOW*

:DECode:BUS<n>:SENT:LENGth

描述	设置或查询 SENT 总线的半字节数。
命令格式	:DECode:BUS<n>:SENT:LENGth <value> :DECode:BUS<n>:SENT:LENGth?
参数说明	<n>:= {1|2}，总线序号 <value>:= 数值，整型 NR1 格式，该值范围为 [3,8]
返回格式	整型 NR1 格式
示例	设置 SENT 协议解码总线 1 的半字节数为 5： *:DECode:BUS1:SENT:LENGth 5* *DEC:BUS1:SENT:LENG 5* 查询当前 SENT 协议解码总线 1 的半字节数： *DEC:BUS1:SENT:LENG?* 返回值： *5*

:DECode:BUS<n>:SENT:CRC

描述	设置或查询 SENT 总线的 CRC 校验格式
命令格式	:DECode:BUS<n>:SENT:CRC <state> :DECode:BUS<n>:SENT:CRC?
参数说明	<n>:= {1|2}，总线序号 <state>:= {OFF|ON} ● ON：设置为 2010 CRC 格式 ● OFF：设置为 2008 CRC 格式
返回格式	{OFF|ON}
示例	设置 SENT 协议解码总线 1 的 CRC 校验格式为 2010 CRC： *:DECode:BUS1:SENT:CRC ON* *DEC:BUS1:SENT:CRC ON*

	查询 SENT 协议解码总线 1 的 CRC 校验格式： *DEC:BUS1:SENT:CRC?* 返回值： *ON*

:DECode:BUS<n>:SENT:PPULse

描述	设置或查询 SENT 总线的可变暂停位状态。
命令格式	:DECode:BUS<n>:SENT:PPULse <state> :DECode:BUS<n>:SENT:PPULse?
参数说明	<state>:= {OFF|ON}
返回格式	{OFF|ON}
示例	设置 SENT 协议解码总线 1 的可变暂停位状态为开： *:DECode:BUS1:SENT:PPULse ON* *DEC:BUS1:SENT:PPUL ON* 查询当前 SENT 协议解码总线 1 的可变暂停位状态： *DEC:BUS1:SENT:PPUL?* 返回值： *ON*

:DECode:BUS<n>:MANChester

:DECode:BUS<n>:MANChester:SOURce

描述	设置或查询 MANChester 总线的信源。
命令格式	:DECode:BUS<n>:MANChester:SOURce <source> :DECode:BUS<n>:MANChester:SOURce?
参数说明	<n>:= {1|2}，总线序号 <source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 MANChester 协议解码总线 1 的信源为 C1： *:DECode:BUS1:MANChester:SOURce C1* *DEC:BUS1:MANC:SOUR C1* 查询当前 MANChester 协议解码总线 1 的信源： *DEC:BUS1:MANC:SOUR?* 返回值： *C1*
关联命令	:DECode:BUS<n>:MANChester:THReshold

:DECode:BUS<n>:MANChester:THReshold

描述	设置或查询 MANChester 总线的阈值。
命令格式	:DECode:BUS<n>:MANChester:THReshold <value> :DECode:BUS<n>:MANChester:THReshold?
参数说明	<n>:= {1|2}，总线序号 value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS2000X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 MANChester 协议解码总线 1 的阈值为 1V： *:DECode:BUS1:MANChester:THReshold 1.00E+00* *DEC:BUS1:MANC:THR 1.00E+00* 查询当前 MANChester 协议解码总线 1 的阈值： *DEC:BUS1:MANC:THR?* 返回值： *1.00E+00*
关联命令	:DECode:BUS<n>:MANChester:SOURce

:DECode:BUS<n>:MANChester:BAUD

描述	设置或查询 MANChester 总线的波特率。
命令格式	:DECode:BUS<n>:MANChester:BAUD <baud> :DECode:BUS<n>:MANChester:BAUD?
参数说明	<n>:= {1|2}，总线序号 <value>:= 数值，整型 NR1 格式，该值范围为 [500,5000000]
返回格式	整型 NR1 格式
示例	设置 MANChester 协议解码总线 1 的波特率为 9600bps： *:DECode:BUS1:MANChester:BAUD 9600* *DEC:BUS1:MANC:BAUD 9600* 查询当前 MANChester 协议解码总线 1 的波特率： *DEC:BUS1:MANC:BAUD?* 返回值： *9600*

:DECode:BUS<n>:MANChester:POLarity

描述	设置或查询 MANChester 总线的极性。
命令格式	:DECode:BUS<n>:MANChester:PULarity <polar> :DECode:BUS<n>:MANChester:POLarity?
参数说明	<n>:= {1|2}，总线序号 <polar>:= {RISing|FALLing} ● RISing：上升沿，1 表示上升沿为逻辑 1 ● FALLing：下降沿，1 表示下降沿为逻辑 1
返回格式	{RISing|FALLing}

示例	设置 MANChester 协议解码总线 1 的极性为上升沿： *:DECode:BUS1:MANChester:POLarity RISing* *DEC:BUS1:MANC:POL RIS* 查询当前 MANChester 协议解码总线 1 的极性： *DEC:BUS1:MANC:POL?* 返回值： *RISing*

:DECode:BUS<n>:MANChester:IDLE

描述	设置或查询 MANChester 总线的空闲状态。
命令格式	:DECode:BUS<n>:MANChester:IDLE <idle> :DECode:BUS<n>:MANChester:IDLE?
参数说明	<n>:= {1|2}，总线序号 <idle>:= {LOW|HIGH} ● LOW：低电平 ● LOW：高电平
返回格式	{LOW|HIGH}
示例	设置 MANChester 协议解码总线 1 的空闲状态为低电平： *:DECode:BUS1:MANChester:IDLE LOW* *DEC:BUS1:MANC:IDLE LOW* 查询当前 MANChester 协议解码总线 1 的空闲状态： *DEC:BUS1:MANC:IDLE?* 返回值： *LOW*

:DECode:BUS<n>:MANChester:IBITs

描述	设置或查询 MANChester 总线的帧间隔。
命令格式	:DECode:BUS<n>:MANChester:IBITs <value> :DECode:BUS<n>:MANChester:IBITs?
参数说明	<n>:= {1|2}，总线序号 <value>:= 数值，整型 NR1 格式，该值范围为 [2,32]
返回格式	整型 NR1 格式
示例	设置 MANChester 协议解码总线 1 的帧间隔为 5 位： *:DECode:BUS1:MANChester:IBITs 5* *DEC:BUS1:MANC:IBIT 5*

	查询当前 MANChester 协议解码总线 1 的帧间隔： *DEC:BUS1:MANC:IBIT?* 返回值： *5*

:DECode:BUS<n>:MANChester:STARt

描述	设置或查询 MANChester 总线的起始沿。
命令格式	:DECode:BUS<n>:MANChester:STARt <value> :DECode:BUS<n>:MANChester:STARt?
参数说明	<n>:={1|2}，总线序号 <value>:= 数值，整型 NR1 格式，该值范围为 [1,32]
返回格式	整型 NR1 格式
示例	设置 MANChester 协议解码总线 1 的起始沿为 5： *:DECode:BUS1:MANChester:STARt 5* *DEC:BUS1:MANC:STAR 5* 查询当前 MANChester 协议解码总线 1 的起始沿： *DEC:BUS1:MANC:STAR?* 返回值： *5*

:DECode:BUS<n>:MANChester:SSIZe

描述	设置或查询 MANChester 总线的同步字段长度。
命令格式	:DECode:BUS<n>:MANChester:SSIZe <value> :DECode:BUS<n>:MANChester:SSIZe?
参数说明	<n>:={1|2}，总线序号 <value>:= 数值，整型 NR1 格式，该值范围为 [0,32]
返回格式	整型 NR1 格式
示例	设置 MANChester 协议解码总线 1 的同步字段长度为 5： *:DECode:BUS1:MANChester:SSIZe 5* *DEC:BUS1:MANC:SSIZ 5* 查询当前 MANChester 协议解码总线 1 的同步字段长度： *DEC:BUS1:MANC:SSIZ?* 返回值： *5*

:DECode:BUS<n>:MANChester:HSIZe

描述	设置或查询 MANChester 总线的头部字段长度。
命令格式	:DECode:BUS<n>:MANChester:HSIZe <value> :DECode:BUS<n>:MANChester:HSIZe?
参数说明	<n>:={1|2}，总线序号 <value>:= 数值，整型 NR1 格式，该值范围为 [0,32]
返回格式	整型 NR1 格式
示例	设置 MANChester 协议解码总线 1 的头部字段长度为 5： *:DECode:BUS1:MANChester:HSIZe 5* *DEC:BUS1:MANC:HSIZ 5* 查询当前 MANChester 协议解码总线 1 的头部字段长度： *DEC:BUS1:MANC:HSIZ?* 返回值： *5*
关联命令	:DECode:BUS<n>:MANChester:DISPlay

:DECode:BUS<n>:MANChester:TSIZe

描述	设置或查询 MANChester 总线的尾部字段长度。
命令格式	:DECode:BUS<n>:MANChester:TSIZe <value> :DECode:BUS<n>:MANChester:TSIZe?
参数说明	<n>:={1|2}，总线序号 <value>:= 数值，整型 NR1 格式，该值范围为 [0,32]
返回格式	整型 NR1 格式
示例	设置 MANChester 协议解码总线 1 的尾部字段长度为 5： *:DECode:BUS1:MANChester:TSIZe 5* *DEC:BUS1:MANC:TSIZ 5* 查询当前 MANChester 协议解码总线 1 的尾部字段长度： *DEC:BUS1:MANC:TSIZ?* 返回值： *5*
关联命令	:DECode:BUS<n>:MANChester:DISPlay

:DECode:BUS<n>:MANChester:WSIZe

描述	设置或查询 MANChester 总线的字大小。
命令格式	:DECode:BUS<n>:MANChester:WSIZe <value> :DECode:BUS<n>:MANChester:WSIZe?
参数说明	<n>:={1|2}，总线序号 <value>:= 数值，整型 NR1 格式，该值范围为 [2,8]
返回格式	整型 NR1 格式
示例	设置 MANChester 协议解码总线 1 的字大小为 5： *:DECode:BUS1:MANChester:WSIZe 5* *DEC:BUS1:MANC:WSIZ 5* 查询当前 MANChester 协议解码总线 1 的字大小： *DEC:BUS1:MANC:WSIZ?* 返回值： *5*
关联命令	:DECode:BUS<n>:MANChester:DISPlay

:DECode:BUS<n>:MANChester:DSIZe

描述	设置或查询 MANChester 总线的数据字长度。
命令格式	:DECode:BUS<n>:MANChester:DSIZe <value> :DECode:BUS<n>:MANChester:DSIZe?
参数说明	<n>:={1|2}，总线序号 <value>:= 数值，整型 NR1 格式，该值范围为 [1,255]
返回格式	整型 NR1 格式
示例	设置 MANChester 协议解码总线 1 的数据字长度为 5： *:DECode:BUS1:MANChester:DSIZe 5* *DEC:BUS1:MANC:DSIZ 5* 查询当前 MANChester 协议解码总线 1 的数据字长度： *DEC:BUS1:MANC:DSIZ?* 返回值： *5*
关联命令	:DECode:BUS<n>:MANChester:DISPlay

:DECode:BUS<n>:MANChester:DISPlay

描述	设置或查询 MANChester 总线的显示格式。
命令格式	:DECode:BUS<n>:MANChester:DISPlay <format> :DECode:BUS<n>:MANChester:DISPlay?
参数说明	<n>:= {1|2}，总线序号 <format>:= {WORD|BIT}
返回格式	{WORD|BIT}
示例	设置 MANChester 协议解码总线 1 的显示格式为字： *:DECode:BUS1:MANChester:DISPlay WORD* *DEC:BUS1:MANC:DISP WORD* 查询当前 MANChester 协议解码总线 1 的显示格式： *DEC:BUS1:MANC:DISP?* 返回值： *WORD*

:DECode:BUS<n>:MANChester:BITorder

描述	设置或查询 MANChester 总线的比特流格式。
命令格式	:DECode:BUS<n>:MANChester:BITorder <order> :DECode:BUS<n>:MANChester:BITorder?
参数说明	<n>:= {1|2}，总线序号 <order>:= {LSB|MSB}
返回格式	{LSB|MSB}
示例	设置 MANChester 协议解码总线 1 的比特流格式为 MSB： *:DECode:BUS1:MANChester:BITorder MSB* *DEC:BUS1:MANC:BIT MSB* 查询当前 MANChester 协议解码总线 1 的比特流格式： *DEC:BUS1:MANC:BIT?* 返回值： *MSB*

5.9 DIGital 命令系统（选件）

: DIGital 命令子系统控制数字通道开关状态，对数字通道进行阈值、时滞等设置。

:DIGital

描述	设置或查询示波器的数字通道开关状态。
命令格式	:DIGital <state> :DIGital?
参数说明	<state>:= {ON|OFF} ● ON：开启数字通道 ● OFF：关闭数字通道
返回格式	{ON|OFF}
示例	设置数字通道开启： *:DIGital ON* *DIG ON* 查询数字通道状态： *DIG?* 返回值： *ON*

:DIGital:ACTive

描述	设置或查询当前激活的数字通道。
命令格式	:DIGital:ACTive <digital> :DIGital:ACTive?
参数说明	<digital>:= {D<d>} ● D<d>:= 数字通道，<d>:= 通道序号，整型 NR1 格式
返回格式	{D<d>}
示例	设置数字通道 D5 为当前选中通道： *:DIGital:ACTive D5* *DIG:ACT D5* 问询数字通道当前选中通道： *DIG:ACT?* 返回值： *D5*

:DIGital:BUS<n>:DISPlay

描述	设置或查询示波器的数字通道总线的显示状态。
命令格式	:DIGital:BUS<n>:DISPlay <state> :DIGital:BUS<n>:DISPlay?
参数说明	<n>:= {1|2}，总线序号 <state>:= {ON|OFF} ● ON：显示数字通道总线 ● OFF：关闭数字通道总线
返回格式	{ON|OFF}
示例	设置打开数字通道总线 1： *:DIGital:BUS1:DISPlay ON* *DIG:BUS1:DISP ON* 查询数字通道总线 1 的显示状态： *DIG:BUS1:DISP?* 返回值： *ON*

:DIGital:BUS<n>:DEFault

描述	重置示波器的数字通道总线比特位顺序。
命令格式	:DIGital:BUS<n>:DEFault
参数说明	<n>:= {1|2}，总线序号
示例	重置数字通道总线 1 比特位顺序： *:DIGital:BUS1:DEFault* *DIG:BUS1:DEF*
关联命令	:DIGital:BUS<n>:MAP

:DIGital:BUS<n>:FORMat

描述	设置或查询示波器的数字通道总线的显示格式。
命令格式	:DIGital:BUS<n>:FORMat <format> :DIGital:BUS<n>:FORMat?
参数说明	<n>:= {1|2}，总线序号 <format>:= { BINary|DECimal|UDECimal|HEX}

	● BINary：以二进制格式显示数据通道总线数据 ● DECimal：以十进制格式显示数字通道总线数据 ● UDECimal：以无符号十进制格式显示数字通道总线数据 ● HEX：以十六进制格式显示数字通道总线数据
返回格式	{BINary|DECimal|UDECimal|HEX}
示例	设置以 16 进制显示数字通道总线 1 的数据： *:DIGital:BUS1:FORMat HEX* *DIG:BUS1:FORM HEX* 查询当前数字通道总线 1 的显示格式： *DIG:BUS1:FORM?* 返回值： *HEX*

:DIGital:BUS<n>:MAP

描述	设置数字总线位宽及各位对应的通道；查询按 LSB 顺序返回当前数字总线数据组成。
命令格式	:DIGital:BUS<n>:MAP <source>[...[,<source>]] :DIGital:BUS<n>:MAP?
参数说明	<n>:= {1|2}，总线序号 <source>:= {D<d>} ● D<d>:= 数字通道，<d>:= 通道序号，整型 NR1 格式 **注意**: ● 此命令会同步设置数字总线的位宽，位宽由参数的数量决定。 ● DIGital:BUS<n>:DEFault 将根据当前数字总线位宽将位序列重置为 D0-D15。
返回格式	<source>[...[,<source>]]
示例	设置数字通道总线 1 的比特位顺序为 D0、D3、D7、D15： *:DIGital:BUS1:MAP D0,D3,D7,D15* *DIG:BUS1:MAP D0,D3,D7,D15* 查询当前数字通道总线 1 的比特位顺序。 *DIG:BUS1:MAP?* 返回值： *D0,D3,D7,D15*
关联命令	:DIGital:BUS<n>:DEFault

:DIGital:D<d>

描述	设置或查询数字通道中当前指定通道的开关状态。
命令格式	:DIGital:D<d> <state> :DIGital:D<d>?
参数说明	<d>:= 通道序号，整型 NR1 格式 <state>:= {ON|OFF} ● ON：打开指定的数字通道 ● OFF：关闭指定的数字通道
返回格式	{ON|OFF}
示例	关闭数字通道 D5： *:DIGital:D5 OFF* *DIG:D5 OFF* 查询数字通道 D5 的开关状态： *DIG:D5?* 返回值： *OFF*
关联命令	:DIGital

:DIGital:HEIGht

描述	设置或查询数字通道的显示高度。
命令格式	:DIGital:HEIGht <value> :DIGital:HEIGht?
参数说明	<value>:= 垂直高度值，浮点型 NR3 格式，数值范围 [4.00E+00,8.00E+00]。
返回格式	浮点型 NR3 格式
示例	设置数字通道的显示高度为 6div： *:DIGital:HEIGht 6.00E+00* *DIG:HEIG 6.00E+00* 查询数字通道的显示高度： *DIG:HEIG?* 返回值： *6.00E+00*
关联命令	:DIGital:POSition

:DIGital:LABel<d>

描述	设置或查询指定数字通道的标签字符。
命令格式	:DIGital:LABel<d> <qstring> :DIGital:LABel<d>?
参数说明	<d>:= 通道序号，整型 NR1 格式。 <qstring >:= 带引号的 ASCII 文本字符串，字符串的长度限制为 8。
返回格式	带引号的 ASCII 文本字符串，字符串的长度限制为 8
示例	设置 D15 的标签内容为“IIC_DATA”： *:DIGital:LABel15 "IIC_DATA"* *DIG:LAB15 "IIC_DATA"* 查询 D15 的标签内容： *DIG:LAB15?* 返回值： *"IIC_DATA"*

:DIGital:POINts

描述	查询数字通道当前的采样点数。
命令格式	:DIGital:POINts?
返回格式	浮点型 NR3 格式
示例	查询数字通道的显示点数： *DIG:POIN?* 返回值： *6.25E+02*
关联命令	:DIGital:SRATe

:DIGital:POSition

描述	设置或查询数字通道的显示位置。
命令格式	:DIGital:POSition <value> :DIGital:POSition?
参数说明	<value>:= 垂直位置值，浮点型 NR3 格式。当波形区域未被压缩时，数字波形从波形区域顶部到底部移动的格数。 **注意**：该值的范围随着数字通道开启的数量变化而变化。

返回格式	浮点型 NR3 格式
示例	当数字通道高度为 4div 时，设置位置为 4div： *:DIGital:POSition 4.00E+00* *DIG:POS 4.00E+00* 查询数字通道显示的位置： *DIG:POS?* 返回值： *4.00E+00*
关联命令	:DIGital:HEIGht

:DIGital:SKEW

描述	设置或查询数字通道的时滞。
命令格式	:DIGital:SKEW <value> :DIGital:SKEW?
参数说明	<value>:= 浮点型 NR3 格式，取值范围 [-1.00E-07,1.00E-07]。
返回格式	浮点型 NR3 格式
示例	当数字通道时滞为 100ns： *:DIGital:SKEW 1.00E-07* *DIG:SKEW 1.00E-07* 查询数字通道的时滞： *DIG:SKEW?* 返回值： *1.00E-07*

:DIGital:SRATe

描述	查询数字通道当前的采样率。
命令格式	:DIGital:SRATe?
返回格式	浮点型 NR3 格式
示例	查询数字通道的采样率： *DIG:SRAT?* 返回值： *1.25E+09*

:DIGital:THReshold<n>

描述	设置或查询数字通道组的阈值。
命令格式	:DIGital:THReshold<n> <type> :DIGital:THReshold<n>?
参数说明	<n>:= {1|2} ● 1：通道组 1，包含：D0-D7 ● 2：通道组 2，包含：D8-D15 <type >:= {TTL|CMOS|LVCMOS33|LVCMOS25|CUSTom[,<value>]} <value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：
返回格式	{TTL|CMOS|LVCMOS33|LVCMOS25|CUSTom[,<value>]}
示例	设置数字通道组 1 的阈值电平为 CMOS： *:DIGital:THReshold1 CMOS* *DIG:THR1 CMOS* 查询数字通道组 1 的阈值电平： *DIG:THR1?* 返回值： *CMOS*

机型	范围
SDS7000A SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS5000X HD SDS3000X HD SDS2000X Plus SDS2000X HD	[-1E+01,1E+01]
SDS1000X HD SDS800X HD	[-8E-1,8E-1]

5.10 DISPlay 命令系统

:DISPlay 命令子系统控制波形和屏幕显示。

:DISPlay:AXIS

描述	设置或查询轴标签显示状态。
命令格式	:DISPlay:AXIS <state> :DISPlay:AXIS?
参数说明	<state>:= {ON|OFF} ● ON：开启轴标签 ● OFF：关闭轴标签
返回格式	{ON|OFF}
示例	设置开启轴标签： *:DISPlay:AXIS ON* *DISP:AXIS ON* 查询当前轴标签状态： *DISP:AXIS?* 返回值： *ON*

:DISPlay:AXIS:MODE

描述	设置或查询轴标签显示模式。
命令格式	:DISPlay:AXIS:MODE <mode> :DISPlay:AXIS:MODE?
参数说明	<mode>:= {FIXed|MOVing} ● FIXed：固定模式。轴的位置保持固定，而坐标随着波形的移动而更新 ● MOVing：跟随模式。移动波形时，轴的位置随波形移动，而坐标保持不变
返回格式	{FIXed|MOVing}
示例	设置轴标签模式为固定模式： *:DISPlay:AXIS:MODE FIXed* *DISP:AXIS:MODE FIXed* 查询当前轴标签显示模式： *DISP:AXIS:MODE?*

	返回值： *FIXed*

:DISPlay:AXIS:POSition

描述	设置或查询轴标签垂直轴的显示位置。
命令格式	:DISPlay:AXIS:POSition <pos> :DISPlay:AXIS:POSition?
参数说明	<pos>:= {LEFT|MIDDle|RIGHt} ● LEFT：垂直轴位于屏幕左侧 ● MIDDle：垂直轴位于屏幕中间 ● RIGHt：垂直轴位于屏幕右侧
返回格式	{LEFT|MIDDle|RIGHt}
示例	设置轴标签垂直轴显示位置为右侧： *:DISPlay:AXIS:POSition RIGHt* *DISP:AXIS:POS RIGHt* 查询当前轴标签垂直轴的显示位置： *DISP:AXIS:POS?* 返回值： *RIGHt*

:DISPlay:BACKlight

描述	设置或查询当前示波器的屏幕亮度。
命令格式	:DISPlay:BACKlight <value> :DISPlay:BACKlight?
参数说明	<value>:= 百分比数值，整型 NR1 格式，取值范围 [0,100]
返回格式	整型 NR1 格式
示例	设置示波器屏幕亮度为 100%： *:DISPlay:BACKlight 100* *DISP:BACK 100* 查询当前示波器屏幕亮度： *DISP:BACK?* 返回值： *100*

:DISPlay:CLEar

描述	清除当前屏幕内显示的波形。
命令格式	:DISPlay:CLEar
示例	清除当前屏幕内显示的波形： *:DISPlay:CLEar* *DISP:CLE*
关联命令	:ACQuire:CSWeep

:DISPlay:COLor

描述	设置或查询色温状态。
命令格式	:DISPlay:COLor <state> :DISPlay:COLor?
参数说明	<state>:= {ON|OFF} ● ON：开启色温 ● OFF：关闭色温
返回格式	{ON|OFF}
示例	设置色温开启： *:DISPlay:COLor ON* *DISP:COL ON* 查询当前色温状态： *DISP:COL?* 返回值： *ON*

:DISPlay:GRATicule

描述	设置或查询当前示波器的网格亮度。
命令格式	:DISPlay:GRATicule <value> :DISPlay:GRATicule?
参数说明	<value>:= 百分比数值，整型 NR1 格式，取值范围 [0,100]
返回格式	整型 NR1 格式
示例	设置示波器网格亮度为 50%： *:DISPlay:GRATicule 50*

	DISP:GRAT 50 查询当前网格亮度： *DISP:GRAT?* 返回值： *50*

:DISPlay:GRIDstyle

描述	设置或查询网格的显示模式。
命令格式	:DISPlay:GRIDstyle <type> :DISPlay:GRIDstyle?
参数说明	<mode>:= {FULL|LIGHt|NONE} ● FULL：完整网格。显示 8 行、10 列组成的网格 ● LIGHt：轻网格。将屏幕均分成四部分 ● NONE：无网格。无网格显示
返回格式	{FULL|LIGHt|NONE}
示例	设置网格模式为轻网格： *:DISPlay:GRIDstyle LIGHt* *DISP:GRID LIGH* 查询当前网格模式： *DISP:GRID?* 返回值： *LIGHt*

:DISPlay:HIDemenu

描述	设置隐藏右侧菜单。
命令格式	:DISPlay:HIDemenu
示例	设置隐藏右侧菜单： *:DISPlay:HIDemenu* *DISP:HID*
关联命令	:DISPlay:MENU:HIDE

:DISPlay:INTensity

描述	设置或查询当前示波器波形亮度。
命令格式	:DISPlay:INTensity <value> :DISPlay:INTensity?
参数说明	<value>:= 百分比数值，整型 NR1 格式，取值范围 [0,100]
返回格式	整型 NR1 格式
示例	设置示波器波形亮度为 75%： *:DISPlay:INTensity 75* *DISP:INT 75* 查询当前网格亮度： *DISP:INT?* 返回值： *75*

:DISPlay:MENU

描述	设置或查询菜单的显示样式。
命令格式	:DISPlay:MENU <type> :DISPlay:MENU?
参数说明	<mode>:= {EMBedded|FLOating} ● EMBedded：内嵌 ● FLOating：悬浮
返回格式	{EMBedded|FLOating}
示例	设置菜单的显示样式为悬浮： *:DISPlay:MENU FLOating* *DISP:MENU FLO* 查询当前菜单的显示样式： *DISP:MENU?* 返回值： *FLOating*

:DISPlay:MENU:HIDE

描述	设置或查询隐藏菜单的时间。
命令格式	:DISPlay:MENU:HIDE <time> :DISPlay:MENU:HIDE?

参数说明	<time>:= {OFF|3S|5S|10S|30S|60S} ● OFF：关闭自动隐藏 ● 3S|5S|10S|30S|60S：菜单等待 3S/5S/10S/30S/60S 后隐藏
返回格式	{OFF|3S|5S|10S|30S|60S}
示例	设置隐藏菜单时间为 10S： *:DISPlay:MENU:HIDE 10S* *DISP:MENU:HIDE 10S* 查询当前隐藏菜单时间： *DISP:MENU:HIDE?* 返回值： *10S*
关联命令	:DISPlay:HIDemenu

:DISPlay:PERSistence

描述	设置或查询余辉的显示时间。
命令格式	:DISPlay:PERSistence <time> :DISPlay:PERSistence?
参数说明	<time>:= 余辉显示时间。因机型而异，取值范围请参考下表：

机型	余辉显示时间
SDS7000A SDS6000 Pro SDS6000A SDS6000L SDS5000X HD SDS3000X HD SDS2000X HD	{OFF|INFinite|100MS|200MS|500MS|1S|5S|10S|30S}
SDS2000X Plus SHS800X SHS1000X SDS1000X HD SDS800X HD	{OFF|INFinite|1S|5S|10S|30S}

返回格式	{OFF|INFinite|100MS|200MS|500MS|1S|5S|10S|30S}
示例	设置余辉显示时间为 10S： *:DISPlay:PERSistence 10S* *DISP:PERS 10S* 查询当前余辉显示时间： *DISP:PERS?*

	返回值： *10S*

:DISPlay:TRANsparence

描述	设置或查询信息显示框的透明度，如光标信息栏。仅适用于 SHS800X/ SHS1000X
命令格式	:DISPlay:TRANsparence <value> :DISPlay:TRANsparence?
参数说明	<value>:= 百分比数值，整型 NR1 格式，取值范围 [0,100]
返回格式	整型 NR1 格式
示例	设置信息显示框透明度为 80%： *:DISPlay:TRANsparence 80* *DISP:TRAN 80* 查询信息显示框的透明度： *DISP:TRAN?* 返回值： *80*

:DISPlay:TYPE

描述	设置或查询当前波形的显示类型。
命令格式	:DISPlay:TYPE <type> :DISPlay:TYPE?
参数说明	<type>:= {VECTor|DOT} ● VECTor：矢量。采样点之间通过连线的方式显示 ● DOT：点显。直接显示原始采样点
返回格式	{VECTor|DOT}
示例	设置波形显示类型为矢量： *:DISPlay:TYPE VECTor* *DISP:TYPE VECT* 查询当前波形显示类型： *DISP:TYPE?* 返回值： *VECTor*

5.11 DVM 命令系统

:DVM 子系统命令控制数字电压表（DVM）功能。此功能可用于测量参数，如直流和交流振幅。

:DVM

描述	设置或查询 DVM 开关状态。
命令格式	:DVM <state> :DVM?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置 DVM 开启： *:DVM ON* *DVM ON* 查询当前 DVM 开关状态： *DVM?* 返回值： *ON*

:DVM:ALARm

描述	设置或查询 DVM 超量程报警开关状态。启用时，如果信号幅度超过屏幕范围，将发出警报。
命令格式	:DVM:ALARm <state> :DVM:ALARm?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启 DVM 的过载警报： *:DVM:ALARm ON* *DVM:ALAR ON* 查询 DVM 过载警报的开关状态： *DVM:ALAR?* 返回值： *ON*

:DVM:ARANge

描述	设置或查询 DVM 自动量程开关状态。
命令格式	:DVM:ARANge <state> :DVM:ARANge?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启 DVM 自动量程： *:DVM:ARANge ON* *DVM:ARAN ON* 查询 DVM 自动量程的开关状态： *DVM:ARAN?* 返回值： *ON*

:DVM:CURRent

描述	查询 DVM 当前模式下测量值。当前默认有效位数为 3 位。
命令格式	:DVM:CURRent?
返回格式	浮点型 NR3 格式
示例	查询 DVM 当前模式下的有效测量值： *DVM:CURR?* 返回值： *0.98E+00*

:DVM:HOLD

描述	设置或查询 DVM 保持的开关状态。启用后，测量显示值将保持不变，后台测量仍会持续。
命令格式	:DVM:HOLD <state> :DVM:HOLD?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启 DVM 保持： *:DVM:HOLD ON*

	DVM:HOLD ON 查询 DVM 保持的开关状态： *DVM:HOLD?* 返回值： *ON*

:DVM:MODE

描述	设置或查询 DVM 测量模式。
命令格式	:DVM:MODE <mode> :DVM:MODE?
参数说明	<mode>:= {DCavg|DCRMs|ACRMs|PKPK|AMPLitude} ● DCavg：直流平均值。波形数据的算数平均数 ● DCRMs：直流均方根。直流耦合下，所有数据的均方根值 ● ACRMs：交流均方根。交流流耦合下，所有数据的均方根值 ● PKPK：峰峰值。波形数据中最大值与最小值的差值 ● AMPLitude：振幅。顶端值与底端值的差值，如果不是双峰信号，则是最大值与最小值之差
返回格式	{DCavg|DCRMs|ACRMs|PKPK|AMPLitude}
示例	设置 DVM 测量模式为振幅： *:DVM:MODE AMPLitude* *DVM:MODE AMPL* 查询 DVM 的测量模式： *DVM:MODE?* 返回值： *AMPLitude*

:DVM:SOURce

描述	设置或查询 DVM 测量的信源（模拟通道）。
命令格式	:DVM:SOURce <source> :DVM:SOURce?
参数说明	<source>:= {C<n>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式
返回格式	{C<n>}
示例	设置 DVM 的测量的信源为 C2：

:DVM:SOURce C2

DVM:SOUR C2

查询 DVM 的测量源：

DVM:SOUR?

返回值：

C2

5.12 EYE 命令系统

:EYE 命令子系统控制示波器的眼图功能。

:EYE

描述	设置或查询眼图开关状态。
命令格式	:EYE <state> :EYE?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置眼图为开启： *:EYE ON* *EYE ON* 查询当前眼图开关状态： *EYE?* 返回值： *ON*

:EYE:SIGNal

描述	设置或查询眼图信号类型。
命令格式	:EYE:SIGNal <type> :EYE:SIGNal?
参数说明	<type>:= {NRZ|100BASETX|100BASET1|1000BASET}
返回格式	{NRZ|100BASETX|100BASET1|1000BASET}
示例	将眼图信号类型设置为 NRZ： *:EYE:SIGNal NRZ* *EYE:SIGN NRZ* 查询当前眼图信号类型： *EYE:SIGN?* 返回值： *NRZ*

:EYE:SOURce

描述	设置或查询眼图信源。
命令格式	:EYE:SOURce <source> :EYE:SOURce?
参数说明	<source>:= {C<n>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式
返回格式	{C<n>}
示例	设置眼图的信源为 C2： *:EYE:SOURce C2* *EYE:SOUR C2* 查询当前眼图的信源： *EYE:SOUR?* 返回值： *C2*

:EYE:LEVel

描述	设置或查询眼图信号电平。
命令格式	:EYE:LEVel <level1>,<level2>,<level3>,<level4> :EYE:LEVel?
参数说明	<level>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A / [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000L / [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] 不同信号类型，需设置不同的电平参数： ● NRZ：设置<level1> ● 100BASE-TX/100BASE-T1：设置<level1>,<level2> ● 1000BASE-T：设置<level1>,<level2>,<level3>,<level4>
返回格式	浮点型 NR3 格式
示例	将眼图 1000BASE-T 信号 4 个电平设为 1V,2V,3V,4V： *:EYE:LEVel 1,2,3,4* *EYE:LEV 1,2,3,4* 查询眼图信号电平：

	EYE:LEV? 返回值： *1.00E+00,2.00E+00,3.00E+00,4.00E+00*
关联命令	:EYE:SIGNal

:EYE:HYSTeresis

描述	设置或查询眼图电平迟滞。
命令格式	:EYE:HYSTeresis <value> :EYE:HYSTeresis?
参数说明	<value>:= 浮点型 NR3 格式，取值范围 [0.05,2]
返回格式	浮点型 NR3 格式
示例	将眼图信号迟滞设置为 100mdiv： *:EYE:HYSTeresis 1E-01* *EYE:HYST 1E-01* 查询当眼图信号迟滞： *EYE:HYST?* 返回值： *1.00E-01*

:EYE:CLOCk

描述	设置或查询眼图时钟恢复的类型。
命令格式	:EYE:CLOCk <type> :EYE:CLOCk?
参数说明	<type>:= {FIXed|FOPLL|SOPLL} ● FIXed：恒定频率 ● FOPLL：一阶锁相环 ● SOPLL：二阶锁相环
返回格式	{FIXed|FOPLL|SOPLL}
示例	将眼图时钟恢复的类型设置为二阶锁相环： *:EYE:CLOCk SOPLL* *EYE:CLOC SOPLL* 查询当前眼图时钟恢复的类型： *EYE:CLOC?*

	返回值： *SOPLL*

:EYE:CLOCk:MODE

描述	设置或查询眼图时钟恢复下的速率模式、查找模式。
命令格式	:EYE:CLOCk:MODE <rate_mode> :EYE:CLOCk:MODE?
参数说明	<rate_mode>:= {AUTO[,<find_mode>]|MANual} ● AUTO：自动查找速率，设置查找模式需指定<find_mode> ● MANual：手动查找速率，需指定数据速率 <find_mode>:= {FIRSt|EVERy} ● FIRSt：按第一次数据速率恢复时钟 ● EVERy：按每一次数据速率恢复时钟
返回格式	{AUTO[,<find_mode>]|MANual}
示例	将眼图时钟恢复下的速率设置为自动且查找模式设为每一次： *:EYE:CLOCk:MODE AUTO,EVERy* *EYE:CLOC:MODE AUTO,EVER* 查询眼图时钟恢复下的速率模式和查找模式： *EYE:CLOC:MODE?* 返回值： *AUTO,EVERy*
关联命令	:EYE:CLOCk:SRATe

:EYE:CLOCk:SRATe

描述	设置或查询眼图手动恢复时钟的数据速率。
命令格式	:EYE:CLOCk:SRATe <baud> :EYE:CLOCk:SRATe?
参数说明	<baud>:= 浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	将眼图手动恢复时钟的数据速率设置为 1Gbps： *:EYE:CLOCk:SRATe 1E+09* *EYE:CLOC:SRAT 1E+09*

	查询眼图时钟恢复下手动模式的数据速率： *EYE:CLOC:SRAT?* 返回值： *1.00E+09*
关联命令	:EYE:CLOCk:MODE

:EYE:CLOCk:FONCe

描述	眼图自动按第一次查找的数据速率恢复时钟时，可设置重新查找。
命令格式	:EYE:CLOCk:FONCe
示例	重新查找： *:EYE:CLOCk:FONCe* *EYE:CLOC:FONC*
关联命令	:EYE:CLOCk:MODE

:EYE:CLOCk:FOPLL:CUToff

描述	设置或查询眼图一阶锁相环的截止因子。
命令格式	:EYE:CLOCk:FOPLL:CUToff <factor> :EYE:CLOCk:FOPLL:CUToff?
参数说明	<factor>:= 整型 NR1 格式
返回格式	整型 NR1 格式
示例	将眼图一阶锁相环的截止因子设置为 1667： *:EYE:CLOCk:FOPLL:CUToff 1667* *EYE:CLOC:FOPLL:CUT 1667* 查询眼图时钟恢复下锁相环的截止因子： *EYE:CLOC:FOPLL:CUT?* 返回值： *1667*
关联命令	:EYE:CLOCk

:EYE:CLOCk:SOPLL

描述	设置或查询眼图二阶锁相环的传递函数类型。

命令格式	:EYE:CLOCk:SOPLL <type> :EYE:CLOCk:SOPLL?
参数说明	<type>:= {JTF|OJTF}
返回格式	{JTF|OJTF}
示例	将眼图二阶锁相环的传递函数设置为 JTF: *:EYE:CLOCk:SOPLL JTF* *EYE:CLOC:SOPLL JTF* 查询当前眼图二阶锁相环的传递函数类型: *EYE:CLOC:SOPLL?* 返回值: *JTF*

:EYE:CLOCk:SOPLL:JTF:CUToff

描述	设置或查询眼图二阶锁相环 JTF 的截止因子。
命令格式	:EYE:CLOCk:SOPLL:JTF:CUToff <factor> :EYE:CLOCk:SOPLL:JTF:CUToff?
参数说明	<factor>:= 整型 NR1 格式。
返回格式	整型 NR1 格式
示例	将眼图时钟恢复二阶锁相环 JTF 的截止因子设置为 1667: *:EYE:CLOCk:SOPLL:JTF:CUToff 1667* *EYE:CLOC:SOPLL:JTF:CUT 1667* 查询眼图时钟恢复二阶锁相环 JTF 的截止因子: *EYE:CLOC:SOPLL:JTF:CUT?* 返回值: *1667*
关联命令	:EYE:CLOCk:SOPLL

:EYE:CLOCk:SOPLL:JTF:LBANdwidth

描述	设置或查询眼图二阶锁相环 JTF 的环路带宽。
命令格式	:EYE:CLOCk:SOPLL:JTF:LBANdwidth <band> :EYE:CLOCk:SOPLL:JTF:LBANdwidth?
参数说明	<band>:= 浮点型 NR3 格式。
返回格式	浮点型 NR3 格式

示例	当眼图时钟恢复二阶锁相环的数据速率为 1G 时，将 JTF 的环路带宽设置为 10MHz： *:EYE:CLOCk:SOPLL:JTF:LBANdwidth 1E+07* *EYE:CLOC:SOPLL:JTF:LBAN 1E+07* 查询眼图时钟恢复二阶锁相环 JTF 的环路带宽： *EYE:CLOC:SOPLL:JTF:LBAN?* 返回值： *1.00E+07*
关联命令	:EYE:CLOCk:SRATe :EYE:CLOCk:SOPLL:JTF:CUToff

:EYE:CLOCk:SOPLL:JTF:PEAKing

描述	设置或查询眼图时钟恢复二阶锁相环 JTF 的峰值。
命令格式	:EYE:CLOCk:SOPLL:JTF:PEAKing <value> :EYE:CLOCk:SOPLL:JTF:PEAKing?
参数说明	<value>:= 浮点型 NR3 格式。
返回格式	浮点型 NR3 格式
示例	将眼图时钟恢复二阶锁相环 JTF 的峰值设置为 2dB： *:EYE:CLOCk:SOPLL:JTF:PEAKing 2* *EYE:CLOC:SOPLL:JTF:PEAK 2* 查询眼图时钟恢复二阶锁相环 JTF 的峰值： *EYE:CLOC:SOPLL:JTF:PEAK?* 返回值： *2.00E+00*
关联命令	:EYE:CLOCk:SOPLL

:EYE:CLOCk:SOPLL:OJTF:CUToff

描述	设置或查询眼图时钟恢复二阶锁相环 OJTF 的截止因子。
命令格式	:EYE:CLOCk:SOPLL:OJTF:CUToff <factor> :EYE:CLOCk:SOPLL:OJTF:CUToff?
参数说明	<factor>:= 整型 NR1 格式。
返回格式	整型 NR1 格式
示例	将眼图时钟恢复二阶锁相环 OJTF 的截止因子设置为 1667：

	:EYE:CLOCk:SOPLL:OJTF:CUToff 1667 *EYE:CLOC:SOPLL:OJTF:CUT 1667* 查询眼图时钟恢复二阶锁相环 OJTF 的截止因子： *EYE:CLOC:SOPLL:OJTF:CUT?* 返回值： *1667*
关联命令	:EYE:CLOCk:SOPLL

:EYE:CLOCk:SOPLL:OJTF:LBANdwidth

描述	设置或查询眼图时钟恢复二阶锁相环 OJTF 的环路带宽。
命令格式	:EYE:CLOCk:SOPLL:OJTF:LBANdwidth <band> :EYE:CLOCk:SOPLL:OJTF:LBANdwidth?
参数说明	<band>:= 浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	当眼图时钟恢复二阶锁相环的数据速率为 1G 时，将 OJTF 的环路带宽设置为 10MHz： *:EYE:CLOCk:SOPLL:OJTF:LBANdwidth 1.00E+07* *EYE:CLOC:SOPLL:OJTF:LBAN 1.00E+07* 查询眼图时钟恢复二阶锁相环 OJTF 的环路带宽： *EYE:CLOC:SOPLL:OJTF:LBAN?* 返回值： *1.00E+07*
关联命令	:EYE:CLOCk:SRATe :EYE:CLOCk:SOPLL:OJTF:CUToff

:EYE:CLOCk:SOPLL:OJTF:DAMPing

描述	设置或查询眼图时钟恢复二阶锁相环 OJTF 的阻尼系数。
命令格式	:EYE:CLOCk:SOPLL:OJTF:DAMPing <value> :EYE:CLOCk:SOPLL:OJTF:DAMPing?
参数说明	<value>:= 浮点型 NR3 格式。
返回格式	浮点型 NR3 格式
示例	将眼图时钟恢复二阶锁相环 OJTF 的阻尼系数设置为 2：

	:EYE:CLOCk:SOPLL:OJTF:DAMPing 2 *EYE:CLOC:SOPLL:OJTF:DAMP 2* 查询眼图时钟恢复二阶锁相环 OJTF 的阻尼系数： *EYE:CLOC:SOPLL:OJTF:DAMP?* 返回值： *2.00E+00*
关联命令	:EYE:CLOCk:SOPLL

:EYE:OVERlay

描述	设置或查询眼图的波形叠加方式。仅在时钟自动恢复，且查找模式为 FIRSt 时才有效。
命令格式	:EYE:OVERlay <way> :EYE:OVERlay?
参数说明	<way>:= {LAST|ALL} ● LAST：仅显示最后一次 ● ALL：叠加所有
返回格式	{LAST|ALL}
示例	将眼图叠加方式设置为 ALL： *:EYE:OVERlay ALL* *EYE:OVERlay ALL* 查询眼图叠加方式： *EYE:OVER?* 返回值： *ALL*
关联命令	:EYE:CLOCk:MODE

:EYE:OPERate

描述	设置或查询眼图运行开关。
命令格式	:EYE:OPERate <state> :EYE:OPERate?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	将眼图运行开关设置为开：

	:EYE:OPERate ON *EYE:OPER ON* 查询眼图当前运行开关： *EYE:OPER?* 返回值： *ON*
关联命令	:EYE

:EYE:RUN

描述	设置或查询眼图运行状态。
命令格式	:EYE:RUN <state> :EYE:RUN?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	将眼图运行状态设置为运行： *:EYE:RUN ON* *EYE:RUN ON* 查询眼图当前运行状态： *EYE:RUN?* 返回值： *ON*
关联命令	:EYE:OPERate

:EYE:QUICkview

描述	眼图快速设置。自动查找信号电平，按恒定频率自动恢复时钟，并开启眼图常用测量项。
命令格式	:EYE:QUICkview
示例	快速设置眼图： *:EYE:QUICkview* *EYE:QUIC*

:EYE:MEASure

描述	设置或查询眼图的测量开关。
命令格式	:EYE:MEASure <state> :EYE:MEASure?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	将眼图测量功能设置为开启： *:EYE:MEASure ON* *EYE:MEAS ON* 查询眼图测量开关状态： *EYE:MEAS?* 返回值： *ON*

:EYE:MEASure:STATistics

描述	设置或查询眼图测量统计开关。
命令格式	:EYE:MEASure:STATistics <state> :EYE:MEASure:STATistics?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	将眼图测量统计开关设置为开启： *:EYE:MEASure:STATistics ON* *EYE:MEAS:STAT ON* 查询眼图测量统计开关状态： *EYE:MEAS:STAT?* 返回值： *ON*

:EYE:MEASure:STATistics:RESet

描述	重置眼图测量统计。
命令格式	:EYE:MEASure:STATistics:RESet
示例	重置眼图测量统计：

	:EYE:MEASure:STATistics:RESet *EYE:MEAS:STAT:RES*
关联命令	:EYE:MEASure:STATistics

:EYE:MEASure:STATistics:HISTOGram

描述	设置或查询眼图测量统计直方图开关。
命令格式	:EYE:MEASure:STATistics:HISTOGram <state> :EYE:MEASure:STATistics:HISTOGram?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置眼图测量统计直方图为开启： *:EYE:MEASure:STATistics:HISTOGram ON* *EYE:MEAS:STAT:HISTOG ON* 查询眼图测量统计直方图开关状态： *EYE:MEAS:STAT:HISTOGram?* 返回值： *ON*

:EYE:MEASure:STATistics:AIMLimit

描述	设置或查询眼图测量统计的最大 AIM 次数。
命令格式	:EYE:MEASure:STATistics:AIMLimit <num> :EYE:MEASure:STATistics:AIMLimit?
参数说明	<num>:= AIM 次数，整型 NR1 格式，该值取值范围和机型相关。
返回格式	整型 NR1 格式
示例	设置眼图测量统计的 AIM 限制值为 100： *:EYE:MEASure:STATistics:AIMLimit 100* *EYE:MEAS:STAT:AIML 100* 查询眼图测量统计的 AIM 限制值： *EYE:MEAS:STAT:AIML?* 返回值： *100*

:EYE:MEASure:STATistics:MAXCount

描述	设置或查询眼图测量统计的最大统计次数。
命令格式	:EYE:MEASure:STATistics:MAXCount <num> :EYE:MEASure:STATistics:MAXCount?
参数说明	<num>:= 最大统计次数，整型 NR1 格式。取值范围 [0,1024]，其中 0 表示统计次数无限制。
返回格式	整型 NR1 格式
示例	设置眼图测量统计的最大次数为 1024： *:EYE:MEASure:STATistics:MAXCount 1024* *EYE:MEAS:STAT:MAXC 1024* 查询眼图测量统计的最大次数： *EYE:MEAS:STAT:MAX?* 返回值： *1024*

:EYE:MEASure:P<n>

描述	设置或查询眼图指定测量项的开关状态。
命令格式	:EYE:MEASure:P<n> <state> :EYE:MEASure:P<n>?
参数说明	<n>:= 测量项序号，整型 NR1 格式，取值范围 [1,6] <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置眼图第一个测量项为开启： *:EYE:MEASure:P1 ON* *EYE:MEAS:P1 ON* 查询眼图第一个测量项的开关状态： *EYE:MEAS:P1?* 返回值： *ON*

:EYE:MEASure:P<n>:TYPE

描述	设置或查询眼图指定测量项的源和类型。

命令格式	:EYE:MEASure:P<n>:TYPE <eyeindex>,<type> :EYE:MEASure:P<n>:TYPE?
参数说明	<n>:= 测量项序号，整型 NR1 格式，取值范围［1,6］ <eyeindex>:= {COMMon|EYE0|EYE1|EYE2|EYE3} <type>:={ETIe|EPOWer|EVPP|EWIDth|EHEight|EONe|EZERo|ERISe|EFALl|ECRossing|EQFactor|EAMPlitude} 眼图测量参数表

参数	描述
ETIe（时间间隔误）	实际信号有效沿时间点和理想信号有效沿时间点的差值。
EPOWer（平均功率）	整个数据流的平均值，“1”所占的比例越高，该参数越大。
EVPP（峰峰值）	波形数据中最大值与最小值的差值。
EWIDth（眼宽）	眼图在水平方向上张开的宽度。
EHEight（眼高）	双峰信号的最大平顶值。
EONe（1 电平）	眼图的“1”电平值。
EZERo（0 电平）	眼图的“0”电平值。
ERISe（上升时间）	20%~80%上升时间。
EFALl（下降时间）	80%~20%下降时间。
ECRossing（眼图交叉比）	交叉点到 0 电平的幅度与眼幅值的比值。
EQFactor（Q 因子）	眼幅值与 1 电平和 0 电平上的噪声幅度的比值。
EAMPlitude（眼幅值）	眼图的幅度，1 电平与 0 电平的差值。

返回格式	{COMMon|EYE0|EYE1|EYE2|EYE3},{ETIe|EPOWer|EVPP|EWIDth|EHEight|EONe|EZERo|ERISe|EFALl|ECRossing|EQFactor|EAMPlitude}
示例	设置眼图测量第一个测量项为 EYE0 的眼宽： *:EYE:MEASure:P1:TYPE EYE0,EWIDth* *EYE:MEAS:P1:TYPE EYE0,EWID* 查询眼图测量第一个测量项的源和类型： *EYE:MEAS:P1:TYPE?* 返回值： *EYE0,EWIDth*
关联命令	:EYE:MEASure:P<n>

:EYE:MEASure:P<n>:VALue

描述	查询当前眼图指定测量项的值。
命令格式	:EYE:MEASure:P<n>:VALue?
参数说明	<n>:= 测量项序号，整型 NR1 格式，取值范围［1,6］
返回格式	浮点型 NR3 格式
示例	查询第一个测量项的值： *EYE:MEAS:P1:VAL?* 返回值： *2.733E+00*
关联命令	:EYE:MEASure:P<n>:TYPE

:EYE:MEASure:P<n>:STATistics

描述	查询眼图指定测量项的统计信息。
命令格式	:EYE:MEASure:P<n>:STATistics? <type>
参数说明	<n>:= 测量项序号，整型 NR1 格式，取值范围［1,6］ <type>:= {ALL|CURRent|MEAN|MAXimum|MINimum|STDev|COUNt} ● ALL：所有统计参数 ● CURRent：当前值 ● MEAN：平均值 ● MAXimum：最大值 ● MINimum：最小值 ● STDev：标准差 ● COUNt：统计次数
返回格式	浮点型 NR3 格式
示例	查询眼图第一个测量项的当前值： *:EYE:MEASure:P1:STATistics? CURR* *EYE:MEAS:P1:STAT? CURR* 返回值： *2.733E+00*
关联命令	:EYE:MEASure:STATistics

:EYE:MEASure:P<n>:SHIStory

描述	查询眼图指定测量项的历史统计数据。仅在最大统计次数有限制时有效，无限制时则返回统计当前值。
命令格式	:EYE:MEASure:P<n>:SHIStory? [<count>]
参数说明	<n>:= 测量项序号，整型 NR1 格式，取值范围 [1,6] <count>:= 历史统计数据的个数，整型 NR1 格式，取值范围受最大统计次数限制。 **注意**：不指定<count>时，返回所有历史数据。
返回格式	Count=<value>,<value1>,<value2>,...,<valueN> <value>:= 统计数据的个数，整型 NR1 格式 <valueN>:= 统计数据值，浮点型 NR3 格式
示例	当最大统计次数为 1024 时，查询眼图 P1 当前值前 1 个的测量数据： *:EYE:MEASure:P1:SHIStory? 1* *EYE:MEAS:P1:SHIS? 1* 返回值： *Count=1,-3.333973E-11*
关联命令	:EYE:MEASure:STATistics:MAXCount

:EYE:MEASure:CLEar

描述	清除眼图所有的测量项。
命令格式	:EYE:MEASure:CLEar
示例	清除眼图所有的测量项： *:EYE:MEASure:CLEar* *EYE:MEASV:CLE*
关联命令	:EYE:MEASure:P<n> :EYE:MEASure:P<n>:TYPE

:EYE:MTESt

描述	设置或查询眼图模板测试开关状态。
命令格式	:EYE:MTESt <state> :EYE:MTESt?

参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置眼图模板测试开关为开启： *:EYE:MTESt ON* *EYE:MTES ON* 查询眼图模板测试开关状态： *EYE:MTES?* 返回值： *ON*

:EYE:MTESt:TYPE

描述	设置或查询眼图模板测试类型。
命令格式	:EYE:MTESt:TYPE <type> :EYE:MTESt:TYPE?
参数说明	<type>:= {ALL_IN|ALL_OUT|ANY_IN|ANY_OUT} ● ALL_IN：所有波形点均处于模板内时判断为通过，只要有 1 个波形点不在模板区域即判断为失败 ● ALL_OUT：所有波形点均处于模板外时判断为通过，只要有 1 个波形点进入模板区域即判断为失败 ● ANY_IN：只要有 1 个波形点处于模板内则判断为通过，否则判断为失败 ● ANY_OUT：只要有 1 个波形点处于模板外则判断为通过，否则判断为失败
返回格式	{ALL_IN|ALL_OUT|ANY_IN|ANY_OUT}
示例	设置眼图模板测类型为 ALL_IN： *:EYE:MTESt:TYPE ALL_IN* *EYE:MTES:TYPE ALL_IN* 查询眼图模板测类型： *EYE:MTES:TYPE?* 返回值： *ALL_IN*

:EYE:MTESt:MASK:LOAD

描述	眼图模板测试加载内部或外部存储的模板。
命令格式	:EYE:MTESt:MASK:LOAD <location>
参数说明	<location>:={BUILtin,<name>|EXTernal,<path>}

- BUILtin:= 内置模板。
  <name>:= 带引号的文件名字符串，当前可选项{"USB2.0_T1"|"USB2.0_T2"|"USB2.0_T3"|"USB2.0_T4"|"USB2.0_T5"|"USB2.0_T6"|"100BASE_TX"}
- EXTernal:= 通过指定路径<path>加载模板。
  <path>:= 带引号的路径名字符串，扩展名为".smsk"；用户可以根据需要从内部、网络存储器或 U 盘调用模板

存储路径类型	示例
内部	"local/SIGLENT/test.smsk"
网络存储器	"net_storage/SIGLENT/test.smsk"
U 盘	"U-disk0/SIGLENT/test.smsk"

注意：文件格式不是由文件扩展名自动确定，您需要选择一个扩展名与所选文件格式一致的文件。

示例

眼图模板测试加载内部名称为"USB2.0_T1"的模板：

:EYE:MTESt:MASK:LOAD BUILtin,"USB2.0_T1"

EYE:MTES:MASK:LOAD BUIL,"USB2.0_T1"

眼图模板测试加载外部 U-disk0 中 SIGLENT 目录下文件名为 TEST.smsk"的模板：

:EYE:MTES:MASK:LOAD EXTernal,"U-disk0/SIGLENT/TEST.msk"

EYE:MTES:MASK:LOAD EXT,"U-disk0/SIGLENT/TEST.msk"

:EYE:MTESt:OPERate

描述	设置或查询眼图下模板测试的操作开关。
命令格式	:EYE:MTESt:OPERate <state> :EYE:MTESt:OPERate?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置眼图下模板测试的操作开关为开启： *:EYE:MTESt:OPERate ON* *EYE:MTES:OPER ON* 查询当前眼图下模板测试的操作开关： *EYE:MTES:OPER?* 返回值： *ON*

:EYE:MTESt:COUNt

描述	查询眼图模板测试的结果。
命令格式	:EYE:MTESt:COUNt?
返回格式	FAIL,<num>,PASS,<num>,TOTAL,<num> <num>:= 测试数据值，整型 NR1 格式
示例	查询眼图模板测试的结果： *:EYE:MTES:COUN?* 返回值： *FAIL,38176,PASS,5617,TOTAL,43793*
关联命令	:EYE:MTESt:OPERate

:EYE:MTESt:FUNCtion:BUZZer

描述	设置或查询眼图下模板测试蜂鸣器开关状态。开启蜂鸣器时，检测到故障帧发声提示。
命令格式	:EYE:MTESt:FUNCtion:BUZZer <state> :EYE:MTESt:FUNCtion:BUZZer?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置眼图模板测试的蜂鸣器为开启： *:EYE:MTESt:FUNCtion:BUZZer ON* *EYE:MTES:FUNC:BUZZ ON* 查询眼图模板测试蜂鸣器的开关状态： *EYE:MTES:FUNC:BUZZ?* 返回值： *ON*

:EYE:MTESt:FUNCtion:SOF

描述	设置或查询眼图下模板测试中失败即停的开关状态。开启失败即停时，当检测到故障帧则停止采集。
命令格式	:EYE:MTESt:FUNCtion:SOF <state> :EYE:MTESt:FUNCtion:SOF?
参数说明	<state>:= {ON|OFF}

返回格式	{ON|OFF}
示例	设置眼图模板测试的失败即停为开启： *:EYE:MTESt:FUNCtion:SOF ON* *EYE:MTES:FUNC:SOF ON* 查询眼图模板测试中失败即停的开关状态： *EYE:MTES:FUNC:SOF?* 返回值： *ON*

5.13 FUNCtion 命令系统

:FUNCtion 命令子系统命令控制示波器中的数学函数。

:FUNCtion:FFTDisplay

描述	设置或查询 FFT 显示模式。
命令格式	:FUNCtion:FFTDisplay <mode> :FUNCtion:FFTDisplay?
参数说明	<mode>:= {SPLit|FULL|EXCLusive} ● SPLit：分屏。时域波形和频域波形分屏显示，时域波形在上半屏，频域波形在下半屏。在分屏模式下，如果打开波形缩放，则缩放波形和频域波形共享下半屏显示 ● FULL：全屏。时域波形和频域波形同屏显示 ● EXCLusive：频谱仪。仅显示频域波形
返回格式	{SPLit|FULL|EXCLusive}
示例	设置 FFT 显示模式为分屏： *:FUNCtion:FFTDisplay SPLit* *FUNC:FFTD SPL* 查询 FFT 显示模式： *FUNC:FFTD?* 返回值： *SPLit*

:FUNCtion:GVALue

描述	设置或查询数学函数积分门限阈值。
命令格式	:FUNCtion:GVALue <valueA>,<valueB> :FUNCtion:GVALue?
参数说明	<valueA/B>:= 数值，浮点型 NR3 格式，取值范围 [-水平格数/2*当前时基+水平延时，水平格数/2*当前时基+水平延时] **注意**：GA 的值不能大于 GB 的值。如果设置的值大于 GB，将自动设置为与 GB 相同的值；GB 比 GA 小时同理自动设置为 GA 的值。
返回格式	<valueA>,<valueB>:= 数值，浮点型 NR3 格式

示例	设置积分门限为-100ns 到 100ns： *:FUNCtion:GVALue -1.00E-07,1.00E-07* *FUNC:GVAL -1.00E-07,1.00E-07* 查询积分门限： *FUNC:GVAL?* 返回值： *-1.00E-07,1.00E-7*
关联命令	:FUNCtion<x>:INTegrate:GATE

:FUNCtion<x>

描述	设置或查询数学函数开关状态。
命令格式	:FUNCtion<x> <state> :FUNCtion<x>?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启函数 1： *:FUNCtion1 ON* *FUNC1 ON* 查询函数 1 的开关状态： *FUNC1?* 返回值： *ON*

:FUNCtion<x>:AVERage:NUM

描述	设置或查询平均算子的平均次数。
命令格式	:FUNCtion<x>:AVERage:NUM <num> :FUNCtion<x>:AVERage:NUM?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <num>:= 平均次数。因机型而异，取值范围请参考下表：

机型	<num>
SDS7000A SDS6000 Pro SDS6000A	{4|16|32|64|128|256|512|1024|2048|4096|8192}

	SDS6000L SDS5000X HD SDS3000X HD SDS2000X HD	
	SDS2000X Plus SDS1000X HD SDS800X HD	{4|16|32|64|128|256|512|1024}
返回格式	{4|16|32|64|128|256|512|1024|2048|4096|8192}	
示例	设置函数 2 平均算子的平均次数为 128： *:FUNCtion2:AVERage:NUM 128* *FUNC2:AVER:NUM 128* 查询函数 2 平均算子的平均次数： *FUNC2:AVER:NUM?* 返回值： *128*	

:FUNCtion<x>:DELay:DELay

描述	设置或查询数学函数延时算子的延时值。
命令格式	:FUNCtion<x>:DELay:DELay <value> :FUNCtion<x>:DELay:DELay?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <value>:= 延时值，浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置函数 1 延时算子的延时值为 1us： *:FUNCtion1:DELay:DELay 1e-6* *FUNC1:DEL:DEL 1e-06* 查询函数 1 延时算子的延时值： *FUNC1:DEL:DEL?* 返回值： *1e-06*

:FUNCtion<x>:DIFF:DX

描述	设置或查询数学函数导数的步长。
命令格式	:FUNCtion<x>:DIFF:DX <dx>

	:FUNCtion<x>:DIFF:DX?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <dx>:= 步长值，整型 NR1 格式
返回格式	整型 NR1 格式
示例	设置函数 1 导数算子的步长为 52： *:FUNCtion1:DIFF:DX 52* *FUNC1:DIFF:DX 52* 查询函数 1 导数算子的步长： *FUNC1:DIFF:DX?* 返回值： *52*

:FUNCtion<x>:ENVelope:POINts

描述	设置或查询包络算子最大点数。
命令格式	:FUNCtion<x>:ENVelope:POINts <point> :FUNCtion<x>:ENVelope:POINts?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <point>:= 最大点数。因机型而异，取值范围请参考下表：

机型	<point>
SDS7000A	{4k|8k|16k|32k|64k|128k|256k|512k|1M|2M|4M|8M|16M|32M}
SDS6000 Pro SDS6000A SDS6000L SDS5000X HD	{4k|8k|16k|32k|64k|128k|256k|512k|1M|2M|4M|8M}
SDS3000X HD	{4k|8k|16k|32k|64k|128k|256k|512k|1M|2M|4M}

返回格式	<point>
示例	设置函数 1 包络算子的最大点数为 2M： *:FUNCtion1:ENVelope:POINts 2M* *FUNC1:ENV:POIN 2M* 查询函数 1 包络算子的最大点数： *FUNC1:ENV:POIN?* 返回值： *2M*

:FUNCtion<x>:ERES:BITS

描述	设置或查询数学函数增强分辨率的增强位数。
命令格式	:FUNCtion<x>:ERES:BITS <bits> :FUNCtion<x>:ERES:BITS?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <bits>:= {0.5|1.0|1.5|2.0|2.5|3.0}
返回格式	{0.5|1.0|1.5|2.0|2.5|3.0}
示例	设置函数 1 增强分辨率算子的增强位数为 3.0： *:FUNCtion1:ERES:BITS 3.0* *FUNC1:ERES:BITS 3.0* 查询函数 1 增强分辨率算子的增强位数： *FUNC1:ERES:BITS?* 返回值： *3.0*

:FUNCtion<x>:FFT:AUToset

描述	自动设置 FFT 波形水平和垂直参数，快捷地显示 FFT 波形。
命令格式	:FUNCtion<x>:FFT:AUToset <mode>
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <mode>:= {SPAN|PEAK|NORMal} ● SPAN：全频率范围 ● PEAK：峰值居中 ● NORMal：中心设置为基频，频率范围设置为 FFT 采样率的一半
示例	设置函数 1FFT 波形在屏幕上的显示位置为峰值居中： *:FUNCtion1:FFT:AUToset PEAK* *FUNC1:FFT:AUT PEAK*

:FUNCtion<x>:FFT:HCENter

描述	设置或查询 FFT 的中心频率。
命令格式	:FUNCtion<x>:FFT:HCENter :FUNCtion<x>:FFT:HCENter?

参数说明	<x>:= 数学函数序号，整型 NR1 格式 := 数值，浮点型 NR3 格式 **注意**：取值范围受到 FFT 采样率影响。
返回格式	浮点型 NR3 格式
示例	设置函数 1 的 FFT 中心频率为 2MHz： *:FUNCtion1:FFT:HCENter 2.00E+06* *FUNC1:FFT:HCEN 2.00E+06* 查询函数 1 的 FFT 的中心频率： *FUNC1:FFT:HCEN?* 返回值： *2.00E+06*

:FUNCtion<x>:FFT:HSCale

描述	查询 FFT 的水平档位
命令格式	:FUNCtion<x>:FFT:HSCale?
参数说明	<x>:= 数学函数序号，整型 NR1 格式
返回格式	<scale>:= 数值，浮点型 NR3 格式
示例	查询函数 1 的 FFT 的水平档位： *FUNC2:FFT:HSC?* 返回值： *1.00E+06*
关联命令	:FUNCtion<x>:FFT:HCENter :FUNCtion<x>:FFT:SPAN

:FUNCtion<x>:FFT:SPAN

描述	设置或查询 FFT 的频率范围。
命令格式	:FUNCtion<x>:FFT:SPAN :FUNCtion<x>:FFT:SPAN?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 := 数值，浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置函数 1 的 FFT 频率范围为 20MHz：

	:FUNCtion1:FFT:SPAN 2.00E+07 *FUNC1:FFT:SPAN 2.00E+07* 查询函数 1 的 FFT 频率范围： *FUNC1:FFT:SPAN?* 返回值： *2.00E+07*
关联命令	:FUNCtion<x>:FFT:HCENter

:FUNCtion<x>:FFT:LOAD

描述	设置或查询 FFT 的外部负载。
命令格式	:FUNCtion<x>:FFT:LOAD <load> :FUNCtion<x>:FFT:LOAD?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <load>:= 数值，整型 NR1 格式，取值范围 [1,1000000] **注意**：只有当 FFT 的单位为 dBm 时，才可以设置外部负载。
返回格式	整型 NR1 格式
示例	设置函数 1 的 FFT 的外部负载为 50Ω： *:FUNCtion1:FFT:LOAD 50* *FUNC1:FFT:LOAD 50* 查询函数 1 的 FFT 的外部负载： *FUNC1:FFT:LOAD?* 返回值： *50*
关联命令	:FUNCtion<x>:FFT:UNIT

:FUNCtion<x>:FFT:MODE

描述	设置或查询 FFT 的采集模式。
命令格式	:FUNCtion<x>:FFT:MODE <mode> :FUNCtion<x>:FFT:MODE?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <mode>:= {NORMal|MAXHold|AVERage[,<num>]} ● NORMal：普通。直接显示每帧 FFT 运算结果。

	● MAXHold：最大保持。显示历史计算结果中的最大幅度结果，适用于检测非连续波，如偶发脉冲信号，或跳频信号。 ● AVERage：平均。减小信号上叠加的随机噪声成分的影响，需指定<num>。 <num>:= 平均次数，整型 NR1 格式，取值范围 [4,1024]
返回格式	{NORMal|MAXHold|AVERage[,<num>]}
示例	设置函数 1 FFT 的采集模式为普通： *:FUNCtion1:FFT:MODE NORMal* *FUNC1:FFT:MODE NORM* 查询函数 1 FFT 的采集模式： *FUNC1:FFT:MODE?* 返回值： *NORMal*

:FUNCtion<x>:FFT:POINts

描述	设置或查询 FFT 最大点数。
命令格式	:FUNCtion<x>:FFT:POINts <point> :FUNCtion<x>:FFT:POINts?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <point>:= 最大点数。因机型而异，取值范围请参考下表：

机型	<point>
SDS7000A	{1k|2k|4k|8k|16k|32k|64k|128k|256k|512k|1M|2M|4M|8M|16M|32M}
SDS6000 Pro SDS6000A SDS6000L SDS5000X HD	{1k|2k|4k|8k|16k|32k|64k|128k|256k|512k|1M|2M|4M|8M}
SDS3000X HD	{1k|2k|4k|8k|16k|32k|64k|128k|256k|512k|1M|2M|4M}
SDS500X SDS2000X HD SDS2000X Plus SDS1000X HD SDS800X HD	{1k|2k|4k|8k|16k|32k|64k|128k|256k|512k|1M|2M}
SHS800X SHS1000X	{1k|2k|4k|8k|16k|32k|64k|128k|256k|512k|1M}

返回格式	<point>
示例	设置函数 1 FFT 的最大点数为 2M： *:FUNCtion1:FFT:POINts 2M* *FUNC1:FFT:POIN 2M* 查询函数 1 FFT 的最大点数： *FUNC1:FFT:POIN?* 返回值： *2M*

:FUNCtion<x>:FFT:RESET

描述	当 FFT 采集模式为平均时，设置 FFT 重新开始计数。
命令格式	:FUNCtion<x>:FFT:RESET
参数说明	<x>:= 数学函数序号，整型 NR1 格式
示例	设置 FFT 重新开始计数： *:FUNCtion2:FFT:RESET* *FUNC2:FFT:RESET*
关联命令	:FUNCtion<x>:FFT:MODE

:FUNCtion<x>:FFT:RLEVel

描述	设置或查询 FFT 的参考电平。
命令格式	:FUNCtion<x>:FFT:RLEVel <level> :FUNCtion<x>:FFT:RLEVel?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <level>:= 参考电平值，浮点型 NR3 格式；该值和 FFT 源的探头系数相关，各机型取值范围请参考下表：

探头系数	单位：dBVrms	单位：Vrms	单位：dBm
1E6X	[-40,200]	[1E-2,1E10]	[-27,213]
1E5X	[-60,180]	[1E-3,1E9]	[-47,193]
1E4X	[-80,160]	[1E-4,1E8]	[-67,173]
1000X	[-100,140]	[1E-5,1E7]	[-87,153]
100X	[-120,120]	[1E-6,1E6]	[-107,133]
10X	[-140,100]	[1E-7,1E5]	[-127,113]

1X	[-160,80]	[1E-8,1E4]	[-147,93]
0.1X	[-180,60]	[1E-9,1E3]	[-167,73]
0.01X	[-200,40]	[1E-10,1E2]	[-187,53]
1E-3X	[-220,20]	[1E-11,10]	[-207,33]
1E-4X	[-240,0]	[1E-12,1]	[-227,13]
1E-5X	[-260,-20]	[1E-13,1E-1]	[-247,-7]
1E-6X	[-280,-40]	[1E-14,1E-2]	[-267,-27]

注意：:FUNCtion<x>:FFT:SCALe 越小，参考电平值的精度就越高。

返回格式	浮点型 NR3 格式
示例	当 FFT 单位为 dBVrms 时，设置函数 1FFT 的参考电平为 10dBV： *:FUNCtion1:FFT:RLEVel 1.00E+01* *FUNC1:FFT:RLEV 1.00E+01* 查询函数 1FFT 的参考电平： *FUNC1:FFT:RLEV?* 返回值： *1.00E+01*
关联命令	:CHANnel<n>:PROBe :FUNCtion<x>:FFT:SCALe

:FUNCtion<x>:FFT:SCALe

描述	设置或查询 FFT 的垂直档位。
命令格式	:FUNCtion<x>:FFT:SCALe <scale> :FUNCtion<x>:FFT:SCALe?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <scale>:= 垂直档位，浮点型 NR3 格式；各单位取值范围请参考下表：

单位	范围
dBVrms	[1.00E-01, 2.00E+01]
Vrms	[1.00E-03, 1.00E+01]
dBm	[1.00E-01, 2.00E+01]

返回格式	浮点型 NR3 格式
示例	当 FFT 单位为 dBVrms 时，设置函数 1 FFT 的垂直档位为 20dB： *:FUNCtion1:FFT:SCALe 2.00E+01* *FUNC1:FFT:SCAL 2.00E+01*

	查询函数 1 FFT 的参考电平： *FUNC2:FFT:SCAL?* 返回值： *2.00E+01*
关联命令	:CHANnel<n>:PROBe

:FUNCtion<x>:FFT:SEARch

描述	设置或查询 FFT 的搜索工具类型。
命令格式	:FUNCtion<x>:FFT:SEARch <type> :FUNCtion<x>:FFT:SEARch?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <type>:= {OFF|PEAK|MARKer}
返回格式	{OFF|PEAK|MARKer}
示例	设置函数 1 FFT 的搜索工具类型为标记： *:FUNCtion1:FFT:SEARch MARKer* *FUNC1:FFT:SEAR MARK* 查询函数 1 FFT 的搜索工具类型： *FUNC1:FFT:SEAR?* 返回值： *MARKer*
关联命令	:FUNCtion<x>:FFT:SEARch:THReshold :FUNCtion<x>:FFT:SEARch:EXCursion

:FUNCtion<x>:FFT:SEARch:EXCursion

描述	设置或查询 FFT 搜索工具的峰值偏移。
命令格式	:FUNCtion<x>:FFT:SEARch:EXCursion <value> :FUNCtion<x>:FFT:SEARch:EXCursion?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <value>:= 偏移值，浮点型 NR3 格式，数值范围随相应的单位而变化，当 FFT 单位为 dBVrms 时，值的范围为 [0,1.60E+02]
返回格式	浮点型 NR3 格式
示例	设置函数 1 FFT 的搜索工具的峰值偏移为 20dB： *:FUNCtion1:FFT:SEARch:EXCursion 2.00E+01*

	FUNC1:FFT:SEAR:EXC 2.00E+01 查询函数 1 FFT 的搜索工具的峰值偏移： *FUNC1:FFT:SEAR:EXC?* 返回值： *2.00E+01*
关联命令	:FUNCtion<x>:FFT:SEARch:THReshold

:FUNCtion<x>:FFT:SEARch:MARKer<n>

描述	设置 FFT 指定标记的频率。
命令格式	:FUNCtion<x>:FFT:SEARch:MARKer<n> <freq>
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <n>:= 标记序号，整型 NR1 格式，取值范围 [1,8] <freq>:= {NPeak|NAMPlitude|<value>} ● NPeak：标记下一峰值 ● NAMPlitude：标记下一幅度 <value>:= 标记的频率值，浮点型 NR3 格式
示例	设置函数 1 FFT 标记 1 的频率为 1 MHz： *:FUNCtion1:FFT:SEARch:MARKer1 1.00E+06* *FUNC1:FFT:SEAR:MARK1 1.00E+06*
关联命令	:FUNCtion<x>:FFT:SEARch :FUNCtion<x>:FFT:SEARch:MARKer<n>:SHOW

:FUNCtion<x>:FFT:SEARch:MARKer<n>:SHOW

描述	设置 FFT 指定标记的开关状态。
命令格式	:FUNCtion<x>:FFT:SEARch:MARKer<n>:SHOW <state>
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <n>:= 标记序号，整型 NR1 格式，取值范围 [1,8] <state>:= {ON|OFF}
示例	设置关闭函数 1 FFT 搜索标记 2： *:FUNCtion1:FFT:SEARch:MARKer2:SHOW ON* *FUNC1:FFT:SEAR:MARK2:SHOW ON*
关联命令	:FUNCtion<x>:FFT:SEARch

:FUNCtion<x>:FFT:SEARch:MON

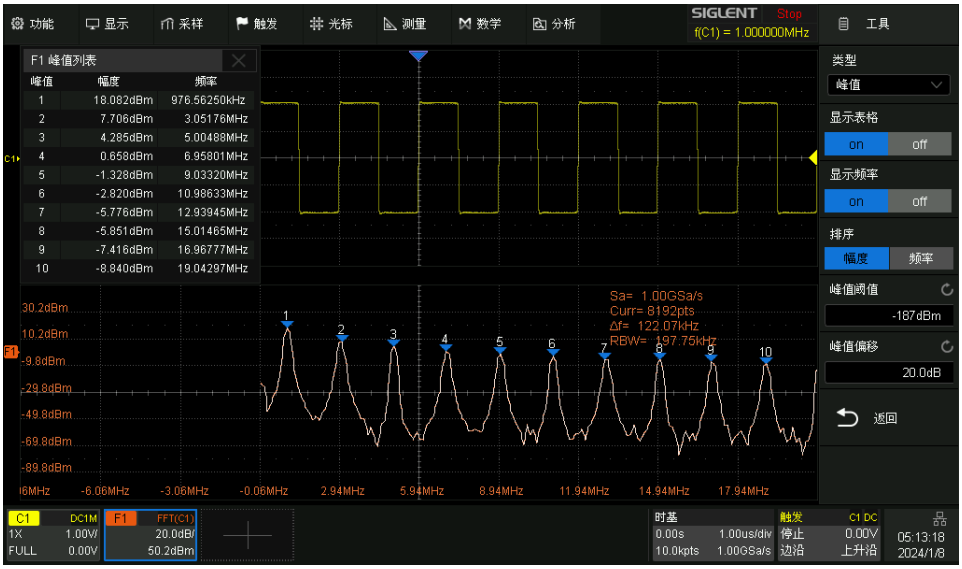
描述	设置 FFT 标记的方式。
命令格式	:FUNCtion<x>:FFT:SEARch:MON <type>
参数说明	<x>:= 数学函数序号，整型 NR1 格式。 <type>:= {PEAK|HARMonics} ● PEAK：标记峰值 ● HARMonics：标记谐波
示例	设置函数 1 FFT 标记峰值： *:FUNCtion1:FFT:SEARch:MON PEAK* *FUNC1:FFT:SEAR:MON PEAK*

:FUNCtion<x>:FFT:SEARch:PORDer

描述	设置或查询 FFT 峰值的排序方式。
命令格式	:FUNCtion<x>:FFT:SEARch:PORDer <type> :FUNCtion<x>:FFT:SEARch:PORDer?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <type>:= {AMPLitude|FREQuency}
返回格式	{AMPLitude|FREQuency}
示例	设置函数 1 FFT 峰值的按幅值排序： *:FUNCtion1:FFT:SEARch:PORDer AMPLitude* *FUNC1:FFT:SEAR:PORDer AMPLitude* 查询函数 1 FFT 峰值的排序方式： *FUNC1:FFT:SEAR:PORD?* 返回值： *AMPLitude*
关联命令	:FUNCtion<x>:FFT:SEARch

:FUNCtion<x>:FFT:SEARch:RESult

描述	查询 FFT 当前搜索列表的结果，只包含搜索编号、频率和幅度信息。
命令格式	:FUNCtion<x>:FFT:SEARch:RESult?
参数说明	<x>:= 数学函数序号，整型 NR1 格式
返回格式	<type>,<no>,<freq>,<ampl>;

	● <type>:= {Markers|Peaks} ● <no>:= 峰值标号，整型 NR1 格式 ● <freq>:= 峰值频率，浮点型 NR3 格式 ● <ampl>:= 峰值幅度，浮点型 NR3 格式
示例	查询返回下图函数 1 FFT 的峰值结果：  *FUNC1:FFT:SEAR:RES?* 返回值： *Peaks,1,9.536743E+02,2.231755E+00;2,3.099442E+03,-8.056905E+00;3,5.006790E+03,-1.151463E+01;4,6.914139E+03,-1.514894E+01;5,9.059906E+03,-1.694874E+01;6,1.096725E+04,-1.847880E+01;7,1.311302E+04,-2.107302E+01;8,1.502037E+04,-2.107302E+01;9,1.692772E+04,-2.264706E+01;10,1.907349E+04,-2.361992E+01;*
关联命令	:FUNCtion<x>:FFT:SEARch:THReshold :FUNCtion<x>:FFT:SEARch:EXCursion :FUNCtion<x>:FFT:UNIT

:FUNCtion<x>:FFT:SEARch:TABLe

描述	设置或查询 FFT 搜索表格的开关状态。
命令格式	:FUNCtion<x>:FFT:SEARch:TABLe <state> :FUNCtion<x>:FFT:SEARch:TABLe?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <state>:= {ON|OFF}
返回格式	{ ON|OFF }
示例	设置打开函数 1 FFT 的搜索表格：

	:FUNCtion1:FFT:SEARch:TABLe ON *FUNC1:FFT:SEAR:TABL ON* 查询函数 1 FFT 的搜索表格的状态： *FUNC1:FFT:SEAR:TABL?* 返回值： *ON*
关联命令	:FUNCtion<x>:FFT:SEARch

:FUNCtion<x>:FFT:SEARch:TABLe:DELTa

描述	设置或查询 FFT 搜索表格中增量的显示状态。
命令格式	:FUNCtion<x>:FFT:SEARch:TABLe:DELTa <state> :FUNCtion<x>:FFT:SEARch:TABLe:DELTa?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置函数 1 FFT 的搜索表格显示增量： *:FUNCtion1:FFT:SEARch:TABLe:DELTa ON* *FUNC1:FFT:SEAR:TABL:DELT ON* 查询函数 1 FFT 的搜索表格增量显示状态： *FUNC1:FFT:SEAR:TABL:DELT?* 返回值： *ON*
关联命令	:FUNCtion<x>:FFT:SEARch:TABLe

:FUNCtion<x>:FFT:SEARch:TABLe:FREQuency

描述	设置或查询 FFT 搜索表格中频率的显示状态。
命令格式	:FUNCtion<x>:FFT:SEARch:TABLe:FREQuency <state> :FUNCtion<x>:FFT:SEARch:TABLe:FREQuency?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置函数 1 FFT 的搜索表格显示频率： *:FUNCtion1:FFT:SEARch:TABLe:FREQuency ON*

	FUNC1:FFT:SEAR:TABL:FREQ ON 查询函数 1 FFT 的搜索表格频率显示状态： *FUNC1:FFT:SEAR:TABL:FREQ?* 返回值： *ON*
关联命令	:FUNCtion<x>:FFT:SEARch:TABLe

:FUNCtion<x>:FFT:SEARch:THReshold

描述	设置或查询 FFT 搜索工具的峰值阈值。
命令格式	:FUNCtion<x>:FFT:SEARch:THReshold <value> :FUNCtion<x>:FFT:SEARch:THReshold?
参数说明	<x>:= 数学函数序号，整型 NR1 格式。 <value>:= 阈值，浮点型 NR3 格式，数值范围随相应的单位而变化，当 FFT 单位为 dBVrms 时，值的范围为 [-1.60E+02, 8.00E+01]。
返回格式	浮点型 NR3 格式
示例	设置函数 1 FFT 搜索工具的峰值阈值为-100dB： *:FUNCtion1:FFT:SEARch:THReshold -1.00E+02* *FUNC1:FFT:SEAR:THR -1.00E+02* 查询函数 1 FFT 搜索工具的峰值阈值： *FUNC1:FFT:SEAR:THR?* 返回值： *-1.00E+02*
关联命令	:FUNCtion<x>:FFT:SEARch:EXCursion

:FUNCtion<x>:FFT:UNIT

描述	设置或查询 FFT 幅度谱或 FFT 相位谱的单位。
命令格式	:FUNCtion<x>:FFT:UNIT <unit> :FUNCtion<x>:FFT:UNIT?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <unit>:= {DBVrms|Vrms|DBm|DEGRee|RADian|SECond}
返回格式	{DBVrms|Vrms|DBm|DEGRee|RADian|SECond}
示例	设置函数 1 FFT 幅度谱的单位为 DBm： *:FUNCtion1:FFT:UNIT DBm*

	FUNC1:FFT:UNIT DBm 查询函数 1 FFT 幅度谱的单位： *FUNC1:FFT:UNIT?* 返回值： *DBm*

:FUNCtion<x>:FFT:WINDow

描述	设置或查询 FFT 的窗口类型。
命令格式	:FUNCtion<x>:FFT:WINDow <window> :FUNCtion<x>:FFT:WINDow?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <window>:= {RECTangle|BLACkman|HANNing|HAMMing|FLATtop|BHARris|GAUSsian} ● RECTangle：矩形窗。适用于瞬态信号，以及时间记录中有整数周期的信号 ● BLACkman：布莱克曼窗。与矩形窗口相比，BLACkman 降低了时间分辨率，但由于次级波瓣较低，它提高了检测较小脉冲的能力（提供了最小的频谱泄漏） ● HANNing：汉宁窗。可用于频率分解和通用测试。它有利于分辨两个接近的频率，或进行频率测量 ● HAMMing：海明窗。 ● FLATtop：平顶窗。是对频率峰值进行精确振幅测量的最佳方法 ● BHARris：布莱克曼-哈里斯窗。通过牺牲频率分辨率换取旁瓣性能，适合需要高精度频谱幅值分析的场景 ● GAUSsian：高斯窗。无负值，主瓣和旁瓣特性由σ参数控制。适合短时傅里叶变换（STFT）等时频分析，或具有高斯包络的信号（如脉冲信号）
返回格式	{RECTangle|BLACkman|HANNing|HAMMing|FLATtop|BHARris|GAUSsian}
示例	设置函数 1 FFT 的窗口类型为平顶窗： *:FUNCtion1:FFT:WINDow FLATtop* *FUNC1:FFT:WIND FLAT* 查询函数 1 的 FFT 的窗口类型： *FUNC1:FFT:WIND?* 返回值： *FLATtop*

:FUNCtion<x>:FFTPhase:UNWRap

描述	设置或查询 FFT 相位谱相位展开开关状态。
命令格式	:FUNCtion<x>:FFTPhase:UNWRap <state> :FUNCtion<x>:FFTPhase:UNWRap?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置函数 1 FFT 相位谱相位展开开启： *:FUNCtion1:FFTPhase:UNWRap ON* *FUNC1:FFTP:UNWR ON* 查询函数 1 FFT 相位谱相位展开开关状态： *FUNC1:FFTP:UNWR?* 返回值： *ON*

:FUNCtion<x>:FFTPhase:UNWRap:THReshold

描述	设置或查询 FFT 相位谱相位展开阈值。相位谱单位为度和弧度时有效。
命令格式	:FUNCtion<x>:FFTPhase:UNWRap:THReshold <value> :FUNCtion<x>:FFTPhase:UNWRap:THReshold?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <value>:= 浮点型 NR3 格式，相位展开阈值范围为[1,359]
返回格式	浮点型 NR3 格式
示例	设置函数 1 FFT 相位谱相位展开阈值为 90°： *:FUNCtion1:FFTPhase:UNWRap:THReshold 90* *FUNC1:FFTP:UNWR:THR 90* 查询函数 1 FFT 相位谱相位展开阈值： *FUNC1:FFTP:UNWR:THR?* 返回值： *9.00E+01*
关联命令	:FUNCtion<x>:FFTPhase:UNWRap :FUNCtion<x>:FFT:UNIT

:FUNCtion<x>:FFTPhase:SQUelch

描述	设置或查询 FFT 相位谱噪声相位抑制开关状态。
命令格式	:FUNCtion<x>:FFTPhase:SQUelch <state> :FUNCtion<x>:FFTPhase:SQUelch?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置函数 1 FFT 相位谱的噪声相位抑制开启： *:FUNCtion1:FFTPhase:SQUelch ON* *FUNC1:FFTP:SQU ON* 查询函数 1 FFT 相位谱的噪声相位抑制开关状态： *FUNC1:FFTP:SQU?* 返回值： *ON*

:FUNCtion<x>:FFTPhase:SQUelch:THReshold

描述	设置或查询 FFT 相位谱噪声相位抑制阈值。
命令格式	:FUNCtion<x>:FFTPhase:SQUelch:THReshold <value> :FUNCtion<x>:FFTPhase:SQUelch:THReshold?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <value>:= 浮点型 NR3 格式，噪声抑制阈值范围为[0,1E+03]
返回格式	浮点型 NR3 格式
示例	设置函数 1 FFT 相位谱的噪声相位抑制阈值为 10V： *:FUNCtion1:FFTPhase:SQUelch:THReshold 10* *FUNC1:FFTP:SQU:THR 10* 查询函数 1 FFT 相位谱的噪声相位抑制阈值： *FUNC1:FFTP:SQU:THR?* 返回值： *1.00E+01*
关联命令	:FUNCtion<x>:FFTPhase:SQUelch

:FUNCtion<x>:FILTer:TYPe

描述	设置或查询数学函数滤波的滤波类型。
命令格式	:FUNCtion<x>:FILTer:TYPe <type> :FUNCtion<x>:FILTer:TYPe?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <type>:= {LPASs|HPASs|BPASs|BREJect} ● LPASs：低通。允许信号中的低频和直流分量通过 ● HPASs：高通。允许信号中的高频分量通过 ● BPASs：带通。允许指定频段内的信号分量通过 ● BREJect：带阻。允许指定频段外的信号分量通过
返回格式	{LPASs|HPASs|BPASs|BREJect}
示例	设置函数 1 滤波算子的滤波类型为高通： *:FUNCtion1:FILTer:TYPE HPASs* *FUNC1:FILT:TYPE HPAS* 查询函数 1 滤波算子的滤波类型： *FUNC1:FILT:TYPE?* 返回值： *HPASs*

:FUNCtion<x>:FILTer:HFRequency

描述	当滤波算子的滤波类型为带通或带阻时，设置或查询滤波算子的上限截止频率。
命令格式	:FUNCtion<x>:FILT:HFRequency <value> :FUNCtion<x>:FILT:HFRequency?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <value>:= 数值，浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置函数 1 滤波算子的上限截止频率为 100MHz： *:FUNCtion1:FILTer:HFRequency 1.00E+08* *FUNC1:FILT:HFR 1.00E+08* 查询函数 1 滤波算子的上限截止频率： *FUNC1:FILT:HFR?* 返回值： *1.00E+08*
关联命令	:FUNCtion<x>:FILTer:TYPe :FUNCtion<x>:FILTer:LFRequency

:FUNCtion<x>:FILTer:LFRequency

描述	当滤波算子的滤波类型为带通或带阻时，设置或查询滤波算子的下限截止频率；当滤波类型为低通或高通时，则设置或查询截止频率。
命令格式	:FUNCtion<x>:FILT:LFRequency <value> :FUNCtion<x>:FILT:LFRequency?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <value>:= 数值，浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置函数 1 滤波算子的下限截止频率为 100MHz： *:FUNCtion1:FILTer:LFRequency 1.00E+08* *FUNC1:FILT:LFR 1.00E+08* 查询函数 1 滤波算子的下限截止频率： *FUNC1:FILT:LFR?* 返回值： *1.00E+08*
关联命令	:FUNCtion<x>:FILTer:HFRequency

:FUNCtion<x>:INTegrate:GATE

描述	设置或查询数学函数积分门限开关状态。
命令格式	:FUNCtion<x>:INTegrate:GATE <state> :FUNCtion<x>:INTegrate:GATE?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启函数 1 积分算子的门限： *:FUNCtion1:INTegrate:GATE ON* *FUNC1:INT:GATE ON* 查询函数 1 积分算子门限的开关状态： *FUNC1:INT:GATE?* 返回值： *ON*
关联命令	:FUNCtion:GVALue

:FUNCtion<x>:INTegrate:OFFSet

描述	设置或查询积分算子的偏移。
命令格式	:FUNCtion<x>:INTegrate:OFFSet <offset> :FUNCtion<x>:INTegrate:OFFSet?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <offset>:= 数值，浮点型 NR3 格式，取值范围 [-100*像素,100*像素]
返回格式	浮点型 NR3 格式
示例	设置函数 1 积分算子的偏移为 100mV： *:FUNCtion1:INTegrate:OFFSet 1.00E-01* *FUNC1:INT:OFFS 1.00E-01* 查询函数 1 积分算子的偏移： *FUNC1:INT:OFFS?* 返回值： *1.00E-01*
关联命令	:CHANnel<n>:PROBe

:FUNCtion<x>:INTErpolate:COEF

描述	设置或查询数学函数插值的插值系数。
命令格式	:FUNCtion<x>:INTErpolate:COEF <coef> :FUNCtion<x>:INTErpolate:COEF?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <coef>:= {2|5|10|20}
返回格式	{2|5|10|20}
示例	设置函数 1 插值算子的插值系数为 10： *:FUNCtion1:INTErpolate:COEF 10* *FUNC1:INTE:COEF 10* 查询函数 1 插值算子的插值系数： *FUNC1:INTE:COEF?* 返回值： *10*

:FUNCtion<x>:INVert

描述	设置或查询数学函数反相开关状态。
命令格式	:FUNCtion<x>:INVert <state> :FUNCtion<x>:INVert?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启函数 1 的反相： *:FUNCtion1:INVert ON* *FUNC1:INV ON* 查询函数 1 反相开关状态： *FUNC1:INV?* 返回值： *ON*

:FUNCtion<x>:LABel

描述	设置或查询数学函数标签开关状态。
命令格式	:FUNCtion<x>:LABel <state> :FUNCtion<x>:LABel?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启函数 1 的标签： *:FUNCtion1:LABel ON* *FUNC1:LAB ON* 查询函数 1 的标签开关状态： *FUNC1:LAB?* 返回值： *ON*
关联命令	:FUNCtion<x>:LABel:TEXT

:FUNCtion<x>:LABel:TEXT

描述	设置或查询指定数学函数的标签字符。
命令格式	:FUNCtion<x>:LABel:TEXT <qstring> :FUNCtion<x>:LABel:TEXT?
参数说明	<n>:= 数学函数序号，整型 NR1 格式 <qstring>:= 带引号的 ASCII 文本字符串，字符串的长度限制为 20
返回格式	带引号的 ASCII 文本字符串
示例	设置函数 1 标签内容为“MATH”： *:FUNCtion1:LABel:TEXT "MATH"* *FUNC1:LAB:TEXT "MATH"* 查询函数 1 标签内容： *FUNC1:LAB:TEXT?* 返回值： *"MATH"*
关联命令	:FUNCtion<x>:LABel

:FUNCtion<x>:MAXHold:SWeeps

描述	设置或查询数学函数最大保持的扫描限制。
命令格式	:FUNCtion<x>:MAXHold:SWeeps <value> :FUNCtion<x>:MAXHold:SWeeps?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <value>:= 数值，整型 NR1 格式。取值范围 [0,2147483646]，其中 0 表示无限制
返回格式	整型 NR1 格式
示例	设置函数 1 最大保持算子的扫描限制为 100： *:FUNCtion1:MAXHold:SWeeps 100* *FUNC1:MAXH:SW 100* 查询函数 1 最大保持算子的扫描限制： *FUNC1:MAXH:SW?* 返回值： *100*

:FUNCtion<x>:MINHold:SWeeps

描述	设置或查询数学函数最小保持的扫描限制。
命令格式	:FUNCtion<x>:MINHold:SWeeps <value> :FUNCtion<x>:MINHold:SWeeps?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <value>:= 数值，整型 NR1 格式。取值范围 [1,2147483646]，其中 0 表示无限制
返回格式	整型 NR1 格式
示例	设置函数 1 最小保持算子的扫描限制为 100： *:FUNCtion1:MINHold:SWeeps 100* *FUNC1:MINH:SW 100* 查询函数 1 最小保持算子的扫描限制： *FUNC1:MINH:SW?* 返回值： *100*

:FUNCtion<x>:OPERation

描述	设置或查询数学函数的算子。
命令格式	:FUNCtion<x>:OPERation <operation> :FUNCtion<x>:OPERation?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <operation>:= {ADD|SUBTract|MULTiply|DIVision|INTegrate|DIFF|FFT|SQRT|ERES|AVERage|ABSolute|SIGN|IDENtity|NEGation|EXP|TEN|LN|LOG|INTErpolate|MAXHold|MINHold|FILTer|ENVelope|TAN|ATAN|FFTPhase|DELay|EXPRess[,<qstring>]} ● 参数说明见下表，设置表达式时需指定<qstring> <qstring>:=带引号的 ASCII 文本字符串，字符串的长度限制为 20

表 5- 1 数学算子表

参数描述	单算子字符串	表达式字符
加法	ADD	+
减法	SUBTract	-
乘法	MULTiply	*
除法	DIVision	/
FFT 幅度谱	FFT	FFTMag()或 FFT()

导数	DIFF	d()/dt
积分	INTegrate -> INTEGrate	S()dt
开方	SQRT	Sqrt()
平均值	AVERage	Average()
增强分辨率	ERES	ERES()
绝对值	ABSolute	Abs()
符号	SIGN	Sign()
恒等	IDENtity	Identity()
相反	NEGation	-()
指数(e)	EXP	Exp()
指数(10)	TEN	Exp10()
自然对数	LN	Ln()
常用对数	LOG -> LG	Lg()
插值	INTErpolate -> INTERpolate	Intrp()
最大保持	MAXHold	MaxHold()
最小保持	MINHold	MinHold()
滤波	FILTer	Filter()
延时	DELay	Delay()
包络	ENVelope	Envelope()
FFT 相位谱	FFTPhase	FFTPhs()
正切	TAN	Tan()
反正切	ATAN	Atan(,)

返回格式 {ADD|SUBTract|MULTiply|DIVision|INTegrate|DIFF|FFT|SQRT|ERES|AVERage|ABSolute|SIGN|IDENtity|NEGation|EXP|TEN|LN|LOG|INTErpolate|MAXHold|MINHold|FILTer|ENVelope|TAN|ATAN|FFTPhase|DELay|EXPRess}

示例 设置算子为 ERES(C1)的导数，可以通过 2 个数学函数嵌套实现 F1=ERES(C1)、F2=FFT(F1)：

:FUNCtion1:SOURce1 C1

:FUNCtion1:OPERation ERES

:FUNCtion2:OPERation FFT

也可以通过 1 个数学函数表达式实现：

:FUNCtion1:OPERation EXPRess,"FFT(ERES(C1))"

:FUNCtion<x>:POSition

描述	设置或查询数学函数的垂直位移。
命令格式	:FUNCtion<x>:POSition <offset> :FUNCtion<x>:POSition?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <offset>:= 垂直位移，浮点型 NR3 格式，取值范围 [-1E+5,1E+5] **注意**：垂直位移值的取值范围不随函数源垂直位移改变，但和算子类型相关。
返回格式	浮点型 NR3 格式
示例	设置函数 1 的垂直位置为 1V： *:FUNCtion1:POSition 1.00E+00* *FUNC1:POS 1.00E+00* 查询函数 1 的垂直位置： *FUNC1:POS?* 返回值： *1.00E+00*
关联命令	:FUNCtion<x>:OPERation

:FUNCtion<x>:SCALe

描述	设置或查询数学函数的垂直档位。
命令格式	:FUNCtion<x>:SCALe <scale> :FUNCtion<x>:SCALe?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <scale>:= 垂直档位，浮点型 NR3 格式，取值范围 [5E-4,1E+4] **注意**：垂直档位的取值范围与函数源档位范围相关，积分或导数的档位与时基相关。
返回格式	浮点型 NR3 格式
示例	设置函数 1 的垂直位档位为 10V： *:FUNCtion1:SCALe 1.00E+01* *FUNC1:SCAL 1.00E+01* 查询函数 1 的垂直位置： *FUNC1:SCAL?* 返回值： *1.00E+01*

关联命令	:CHANnel<n>:SCALe

:FUNCtion<x>:SOURce<n>

描述	设置或查询指定数学函数的信源 A 或信源 B。
命令格式	:FUNCtion<x>:SOURce<n> <source> :FUNCtion<x>:SOURce<n>?
参数说明	<x>:= 数学函数序号，整型 NR1 格式 <n>:= {1|2}，信源序号 <source>:= {C<n>|Z<n>|F<x>|M<m>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● Z<n>:= 缩放波形，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式 **注意**: ● 当 Zoom 开启时，选择 Z<n>作为源才有效 ● 函数不能以自身为源，例：F2=F2+F2
返回格式	{C<n>|Z<n>|F<x>|M<m>}
示例	设置函数 1 的信源 A 为 C1： *:FUNCtion1:SOURce1 C1* *FUNC1:SOUR1 C1* 查询函数 1 的信源 A： *FUNC1:SOUR1?* 返回值： *C1*

5.14 GATE 命令系统

:GATE 命令子系统控制分析门限功能，只有模拟通道&数字通道、内存波形支持分析门限。

:GATE:<channel>:SWITch

描述	设置或查询指定通道的分析门限开关状态。
命令格式	:GATE:<channel>:SWITch <state> :GATE:<channel>:SWITch?
参数说明	<channel>:= {ANALog|MEMory<n>} ● ANALog:= 模拟和数字通道 ● MEMory<n>:= 内存波形 <n>:= 内存波形序号，整型 NR1 格式 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置内存波形 1 的分析门限开关为开启： *:GATE:MEMory1:SWITch ON* *GATE:MEM1:SWIT ON* 查询内存波形 1 的分析门限开关状态： *GATE:MEM1:SWIT?* 返回值： *ON*

:GATE:<channel>:X<n>

描述	设置或查询指定通道分析门限光标 A/B 的位置。
命令格式	:GATE:<channel>:X<n> <value> :GATE:<channel>:X<n>?
参数说明	<channel>:= {ANALog|MEMory<n>} ● ANALog:= 模拟和数字通道 ● MEMory<n>:= 内存波形 <n>:= 内存波形序号，整型 NR1 格式 <n>:= {1|2}，门限光标序号 <value>:= 门限光标 A 的值，浮点型 NR3 格式

返回格式	浮点型 NR3 格式
示例	设置内存波形 1 分析门限光标 A 的位置为-1us： *:GATE:MEMory1:X1 -1.00E-06* *GATE:MEM1:X1 -1.00E-06* 查询内存波形 1 分析门限光标 A 的位置： *GATE:MEM1:X1?* 返回值： *-1.00E-06*

:GATE:<channel>:XDELta

描述	设置或查询指定通道分析门限光标 B-A 的值。
命令格式	:GATE:<channel>:XDELta <value> :GATE:<channel>:XDELta?
参数说明	<channel>:= {ANALog|MEMory<n>} ● ANALog:= 模拟和数字通道 ● MEMory<n>:= 内存波形 <n>:= 内存波形序号，整型 NR1 格式 <value>:= 门限光标 B-A 值，浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置内存波形 1 分析门限光标 B-A 的值为 1us： *:GATE:MEMory1:XDELta 1.00E-06* *GATE:MEM1:XDELta 1.00E-06* 查询内存波形 1 分析门限光标 B-A 的值： *GATE:MEM1:XDELta?* 返回值： *1.00E-06*

:GATE:<channel>:REFerence

描述	设置或查询指定通道的分析门限参考策略。
命令格式	:GATE:<channel>:REFerence <type> :GATE:<channel>:REFerence?
参数说明	<channel>:= {ANALog|MEMory<n>} ● ANALog:= 模拟和数字通道

	● MEMory<n>:= 内存波形 <n>:= 内存波形序号，整型 NR1 格式 <type>:= {DELay|POSition}
返回格式	{DELay|POSition}
示例	设置内存波形 1 分析门限的参考策略为延时： *:GATE:MEMory1:REFerence DELay* *GATE:MEM1:REF DEL* 查询内存波形 1 分析门限的参考策略： *GATE:MEM1:REF?* 返回值： *DELay*

:GATE:<channel>:SYNChronization

描述	设置或查询指定通道的分析门限同步源。
命令格式	:GATE:<channel>:SYNChronization <source> :GATE:<channel>:SYNChronization?
参数说明	<channel>:= {ANALog|MEMory<n>} <source>:= {OFF|ANALog|MEMory<n>} ● ANALog:= 数字和模拟通道 ● MEMory<n>:= 内存波形 <n>:= 内存波形序号，整型 NR1 格式 ● OFF:= 无同步源
返回格式	{OFF|ANALog|MEMory<n>}
示例	设置内存波形 1 分析门限光标的同步源为模拟和数字： *:GATE:MEMory1:SYNChronization ANALog* *GATE:MEM1:SYNC ANAL* 查询内存波形 1 分析门限的同步源： *GATE:MEM1:SYNC?* 返回值： *ANALog*

5.15 HISTORy 命令系统

:HISTORy 命令子系统控制波形记录功能和历史波形播放功能。

:HISTORy

描述	设置或查询历史波形的开关状态。
命令格式	:HISTORy <state> :HISTORy?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启历史波形： *:HISTORy ON* *HISTOR ON* 查询历史波形开关状态： *HISTOR?* 返回值： *ON*

:HISTORy:FRAMe

描述	设置或查询历史波形当前帧号。
命令格式	:HISTORy:FRAMe <value> :HISTORy:FRAMe?
参数说明	<value>:= 帧号，整型 NR1 格式 **注意**：该值的取值范围和存储深度有关，一帧中有更多的点，意味着总帧数也随着变少。
返回格式	整型 NR1 格式
示例	设置历史波形显示第 10 帧： *:HISTORy:FRAMe 10* *HISTOR:FRAM 10* 查询历史波形当前帧号： *HISTOR:FRAM?* 返回值： *10*

:HISTORy:INTERval

描述	设置或查询历史波形的播放间隔。
命令格式	:HISTORy:INTERval <value> :HISTORy:INTERval?
参数说明	<value>:= 播放间隔值，浮点型 NR3 格式，取值范围 [1.00E-06,1]
返回格式	浮点型 NR3 格式
示例	设置历史波形的播放间隔为 1ms: *:HISTORy:INTERval 1.00E-03* *HISTOR:INTER 1.00E-03* 查询历史波形的播放间隔为: *HISTOR:INTER?* 返回值: *1.00E-03*

:HISTORy:LIST

描述	设置或查询历史波形列表的开关状态。
命令格式	:HISTORy:LIST <state> :HISTORy:LIST?
参数说明	<state>:= {OFF|ON[,<type>]} <type>:= {TIME|DELTa} ● TIME：采样时间。列表按采样时间显示 ● DELTa：采样间隔。列表按采样间隔显示
返回格式	{OFF|ON[,<TIME|DELTa>]}
示例	设置开启历史波形列表，选择采样时间显示: *:HISTORy:LIST ON,TIME* *HISTOR:LIST ON,TIME* 查询历史波形列表开关状态: *HISTOR:LIST?* 返回值: *ON,TIME*

:HISTORy:PLAY

描述	设置或查询历史波形的播放状态。
命令格式	:HISTORy:PLAY <state> :HISTORy:PLAY?
参数说明	<state>:= {BACKWards|PAUSe|FORWards} ● BACKWards：向后播放。从当前帧向前回放直到第一帧波形 ● PAUSe：停止播放。停止回放 ● FORWards：向前播放。从当前帧向后回放直到最后一帧波形
返回格式	{BACKWards|PAUSe|FORWards}
示例	设置历史波形的播放状态为向后播放： *:HISTORy:PLAY BACKWards* *HISTOR:PLAY BACKW* 查询历史波形的播放状态： *HISTOR:PLAY?* 返回值： *BACKWards*

:HISTORy:TIME

描述	查询当前历史帧的时间戳。
命令格式	:HISTORy:TIME?
返回格式	<time>:= 时:分:秒.微秒；整型 NR1 格式
示例	查询当前帧的时间戳： *HISTOR:TIME?* 返回值： *07:48:09.253827*
关联命令	:HISTORy:FRAMe

5.16 JITTer 命令系统

:Jitter 命令子系统控制示波器的抖动分析功能。

:JITTer

描述	设置或查询抖动分析开关状态。
命令格式	:JITTer <state> :JITTer?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置抖动分析为开启： *:JITTer ON* *JITT ON* 查询当前抖动分析开关状态： *JITT?* 返回值： *ON*

:JITTer:SOURce

描述	设置或查询抖动分析信源。
命令格式	:JITTer:SOURce <source> :JITTer:SOURce?
参数说明	<source>:= {C<n>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式
返回格式	{C<n>}
示例	设置抖动分析的信源为 C2： *:JITTer:SOURce C2* *JITT:SOUR C2* 查询抖动分析的信源： *JITT:SOUR?* 返回值： *C2*

:JITTer:LEVel

描述	设置或查询抖动分析信号电平。
命令格式	:JITTer:LEVel <value> :JITTer:LEVel?
参数说明	<value>:=计数电平值，浮点型 NR3 格式。因机型而异，取值范围请参考下表： **机型** / **计数电平取值范围** SDS7000A / [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000L / [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置抖动分析信号电平为 0.5V： *:JITTer:LEVel 5.00E-1* *JITT:LEV 5.00E-1* 查询抖动分析信号电平： *JITT:LEV?* 返回值： *5.00E-1*

:JITTer:HYSTeresis

描述	设置或查询抖动分析电平迟滞。
命令格式	:JITTer:HYSTeresis <value> :JITTer:HYSTeresis?
参数说明	<value>:= 浮点型 NR3 格式，取值范围[0.05,2]
返回格式	浮点型 NR3 格式
示例	设置抖动分析的电平迟滞设置为 100mdiv： *:JITTer:HYSTeresis 1E-01* *JITT:HYST 1E-01* 查询抖动分析的电平迟滞： *JITT:HYST?* 返回值： *1.00E-01*

:JITTer:SIGNal

描述	设置或查询抖动分析信号类型。
命令格式	:JITTer:SIGNal <type> :JITTer:SIGNal?
参数说明	<type>:= {DATA|CLOCk}
返回格式	{DATA|CLOCk}
示例	设置抖动分析的信号类型为数据： *:JITTer:SIGNal DATA* *JITT:SIGN DATA* 查询抖动分析的信号类型： *JITT:SIGN?* 返回值： *DATA*

:JITTer:CLOCk

描述	设置或查询抖动分析时钟恢复的类型。
命令格式	:JITTer:CLOCk <type> :JITTer:CLOCk?
参数说明	<type>:= {FIXed|FOPLL|SOPLL} ● FIXed：恒定频率 ● FRPLL：一阶锁相环 ● SOPLL：二阶锁相环
返回格式	{FIXed|FOPLL|SOPLL}
示例	设置抖动分析时钟恢复类型为二阶锁相环： *:JITTer:CLOCk SOPLL* *JITT:CLOC SOPLL* 查询抖动分析时钟恢复的类型： *JITT:CLOC?* 返回值： *SOPLL*

:JITTer:CLOCk:MODE

描述	设置或查询抖动分析时钟恢复下的速率模式、查找模式。

命令格式	:JITTer:CLOCk:MODE <rate_mode> :JITTer:CLOCk:MODE?
参数说明	<rate_mode>:= {AUTO[,<find_mode>]|MANual} ● AUTO：自动查找速率，设置查找模式需指定<find_mode> ● MANual：手动查找速率，需指定数据速率 <find_mode>:= {FIRSt|EVERy} ● FIRSt：按第一次数据速率恢复时钟 ● EVERy：按每一次数据速率恢复时钟
返回格式	{AUTO[,<find_mode>]|MANual}
示例	将抖动分析时钟恢复下的速率设置为自动且查找模式设为每一次： *:JITTer:CLOCk:MODE AUTO,EVERy* *JITT:CLOC:MODE AUTO,EVER* 查询抖动分析时钟恢复下的速率模式和查找模式： *JITT:CLOC:MODE?* 返回值： *AUTO,EVERy*
关联命令	:JITTer:CLOCk:SRATe

:JITTer:CLOCk:SRATe

描述	设置或查询抖动分析手动恢复时钟的数据速率。
命令格式	:JITTer:CLOCk:SRATe <baud> :JITTer:CLOCk:SRATe?
参数说明	<baud>:= 浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	将抖动分析手动恢复时钟的数据速率设置为 1Gbps： *:JITTer:CLOCk:SRATe 1E+09* *JITT:CLOC:SRAT 1E+09* 查询眼图时钟恢复下手动模式的数据速率： *JITT:CLOC:SRAT?* 返回值： *1.00E+09*
关联命令	:JITTer:CLOCk:MODE

:JITTer:CLOCk:FONCe

描述	抖动分析自动按第一次查找的数据速率恢复时钟时，可设置重新查找。
命令格式	:JITTer:CLOCk:FONCe
示例	重新查找： *:JITTer:CLOCk:FONCe* *JITT:CLOC:FONC*
关联命令	:JITTer:CLOCk:MODE

:JITTer:CLOCk:FOPLL:CUToff

描述	设置或查询抖动分析一阶锁相环的截止因子。
命令格式	:JITTer:CLOCk:FOPLL:CUToff <factor> :JITTer:CLOCk:FOPLL:CUToff?
参数说明	<factor>:= 整型 NR1 格式
返回格式	整型 NR1 格式
示例	将抖动分析一阶锁相环的截止因子设置为 1667： *:JITTer:CLOCk:FOPLL:CUToff 1667* *JITT:CLOC:FOPLL:CUT 1667* 查询抖动分析时钟恢复下锁相环的截止因子： *JITT:CLOC:FOPLL:CUT?* 返回值： *1667*
关联命令	:JITTer:CLOCk

:JITTer:CLOCk:SOPLL

描述	设置或查询抖动分析二阶锁相环的传递函数类型。
命令格式	:JITTer:CLOCk:SOPLL <type> :JITTer:CLOCk:SOPLL?
参数说明	<type>:= {JTF|OJTF}
返回格式	{JTF|OJTF}
示例	将抖动分析二阶锁相环的传递函数设置为 JTF： *:JITTer:CLOCk:SOPLL JTF* *JITT:CLOC:SOPLL JTF*

	查询当前抖动分析二阶锁相环的传递函数类型： *JITT:CLOC:SOPLL?* 返回值： *JTF*

:JITTer:CLOCk:SOPLL:JTF:CUToff

描述	设置或查询抖动分析二阶锁相环 JTF 的截止因子。
命令格式	:JITTer:CLOCk:SOPLL:JTF:CUToff <factor> :JITTer:CLOCk:SOPLL:JTF:CUToff?
参数说明	<factor>:= 整型 NR1 格式。
返回格式	整型 NR1 格式
示例	将抖动分析时钟恢复二阶锁相环 JTF 的截止因子设置为 1667： *:JITTer:CLOCk:SOPLL:JTF:CUToff 1667* *JITT:CLOC:SOPLL:JTF:CUT 1667* 查询抖动分析时钟恢复二阶锁相环 JTF 的截止因子： *JITT:CLOC:SOPLL:JTF:CUT?* 返回值： *1667*
关联命令	:JITTer:CLOCk:SOPLL

:JITTer:CLOCk:SOPLL:JTF:LBANdwidth

描述	设置或查询抖动分析二阶锁相环 JTF 的环路带宽。
命令格式	:JITTer:CLOCk:SOPLL:JTF:LBANdwidth <band> :JITTer:CLOCk:SOPLL:JTF:LBANdwidth?
参数说明	<band>:= 浮点型 NR3 格式。
返回格式	浮点型 NR3 格式
示例	当抖动分析时钟恢复二阶锁相环的数据速率为 1G 时，将 JTF 的环路带宽设置为 10MHz： *:JITTer:CLOCk:SOPLL:JTF:LBANdwidth 1E+07* *JITT:CLOC:SOPLL:JTF:LBAN 1E+07* 查询抖动分析时钟恢复二阶锁相环 JTF 的环路带宽： *JITT:CLOC:SOPLL:JTF:LBAN?* 返回值： *1.00E+07*

关联命令	:JITTer:CLOCk:SRATe :JITTer:CLOCk:SOPLL:JTF:CUToff

:JITTer:CLOCk:SOPLL:JTF:PEAKing

描述	设置或查询抖动分析时钟恢复二阶锁相环 JTF 的峰值。
命令格式	:JITTer:CLOCk:SOPLL:JTF:PEAKing <value> :JITTer:CLOCk:SOPLL:JTF:PEAKing?
参数说明	<value>:= 浮点型 NR3 格式。
返回格式	浮点型 NR3 格式
示例	将抖动分析时钟恢复二阶锁相环 JTF 的峰值设置为 2dB: *:JITTer:CLOCk:SOPLL:JTF:PEAKing 2* *JITT:CLOC:SOPLL:JTF:PEAK 2* 查询抖动分析时钟恢复二阶锁相环 JTF 的峰值: *JITT:CLOC:SOPLL:JTF:PEAK?* 返回值: *2.00E+00*
关联命令	:EYE:CLOCk:SOPLL

:JITTer:CLOCk:SOPLL:OJTF:CUToff

描述	设置或查询抖动分析时钟恢复二阶锁相环 OJTF 的截止因子。
命令格式	:JITTer:CLOCk:SOPLL:OJTF:CUToff <factor> :JITTer:CLOCk:SOPLL:OJTF:CUToff?
参数说明	<factor>:= 整型 NR1 格式。
返回格式	整型 NR1 格式
示例	将抖动分析时钟恢复二阶锁相环 OJTF 的截止因子设置为 1667: *:JITT:CLOCk:SOPLL:OJTF:CUToff 1667* *JITT:CLOC:SOPLL:OJTF:CUT 1667* 查询抖动分析时钟恢复二阶锁相环 OJTF 的截止因子: *JITT:CLOC:SOPLL:OJTF:CUT?* 返回值: *1667*
关联命令	:EYE:CLOCk:SOPLL

:JITTer:CLOCk:SOPLL:OJTF:LBANdwidth

描述	设置或查询抖动分析时钟恢复二阶锁相环 OJTF 的环路带宽。
命令格式	:JITTer:CLOCk:SOPLL:OJTF:LBANdwidth <band> :JITTer:CLOCk:SOPLL:OJTF:LBANdwidth?
参数说明	<band>:= 浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	当抖动分析时钟恢复二阶锁相环的数据速率为 1G 时，将 OJTF 的环路带宽设置为 10MHz： *:JITTer:CLOCk:SOPLL:OJTF:LBANdwidth 1.00E+07* *JITT:CLOC:SOPLL:OJTF:LBAN 1.00E+07* 查询抖动分析时钟恢复二阶锁相环 OJTF 的环路带宽： *JITT:CLOC:SOPLL:OJTF:LBAN?* 返回值： *1.00E+07*
关联命令	:EYE:CLOCk:SRATe :EYE:CLOCk:SOPLL:OJTF:CUToff

:JITTer:CLOCk:SOPLL:OJTF:DAMPing

描述	设置或查询抖动分析时钟恢复二阶锁相环 OJTF 的阻尼系数。
命令格式	:JITTer:CLOCk:SOPLL:OJTF:DAMPing <value> :JITTer:CLOCk:SOPLL:OJTF:DAMPing?
参数说明	<value>:= 浮点型 NR3 格式。
返回格式	浮点型 NR3 格式
示例	将抖动分析时钟恢复二阶锁相环 OJTF 的阻尼系数设置为 2： *:JITTer:CLOCk:SOPLL:OJTF:DAMPing 2* *JITT:CLOC:SOPLL:OJTF:DAMP 2* 查询抖动分析时钟恢复二阶锁相环 OJTF 的阻尼系数： *JITT:CLOC:SOPLL:OJTF:DAMP?* 返回值： *2.00E+00*
关联命令	:JITTer:CLOCk:SOPLL

:JITTer:DISPlay

描述	设置或查询抖动分析显示模式。
命令格式	:JITTer:DISPlay <mode> :JITTer:DISPlay?
参数说明	<mode>:= {WAVE|EYE} ● WAVE：通道波形 ● EYE：通道眼图
返回格式	{WAVE|EYE}
示例	设置抖动分析的显示模式为波形： *:JITTer:DISPlay WAVE* *JITT:DISP WAVE* 查询抖动分析的显示模式： *JITT:DISP?* 返回值： *WAVE*

:JITTer:QVIew

描述	抖动分析快速设置。自动查找信号电平，按恒定频率自动恢复时钟。
命令格式	:JITTer:QVIew
示例	抖动分析快速设置： *:JITTer:QVIew* *JITT:QVI*

:JITTer:MEASure

描述	设置或查询抖动分析的测量开关。
命令格式	:JITTer:MEASure <state> :JITTer:MEASure?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	将抖动分析测量功能设置为开启： *:JITTer:MEASure ON* *JITT:MEAS ON*

	查询抖动分析测量开关状态： *JITT:MEAS?* 返回值： *ON*

:JITTer:MEASure:STATistics:RESet

描述	重置抖动分析测量统计。
命令格式	:JITTer:MEASure:STATistics:RESet
示例	重置抖动分析测量统计： *:JITTer:MEASure:STATistics:RESet* *JITT:MEAS:STAT:RES*
关联命令	:JITTer:MEASure:P<n>:STATistics

:JITTer:MEASure:P<n>

描述	设置或查询抖动分析指定测量项的开关状态。
命令格式	:JITTer:MEASure:P<n> <state> :JITTer:MEASure:P<n>?
参数说明	<n>:= 测量项序号，整型 NR1 格式，取值范围 [1,6] <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置抖动分析第一个测量项为开启： *:JITTer:MEASure:P1 ON* *JITT:MEAS:P1 ON* 查询抖动分析第一个测量项的开关状态： *JITT:MEAS:P1?* 返回值： *ON*

:JITTer:MEASure:P<n>:TYPE

描述	设置或查询抖动分析指定测量项的源和类型。
命令格式	:JITTer:MEASure:P<n>:TYPE <type> :JITTer:MEASure:P<n>:TYPE?

参数说明

<n>:= 测量项序号，整型 NR1 格式，取值范围［1,6］

<type>:={JPRD|JFReq|JPWidth|JNWidth|JPDuty|JNDuty|JCPRd|JCPWidth|JCNWidth|JCPDuty|JCNDuty|JTIe|RJ|DJ|DCD|DDJ|PJ|TJ}

抖动分析测量参数表：

参数	描述
JPRD（周期）	信号周期，仅适用于时钟信号。在测量统计中取“标准差”即为时钟的周期抖动。
JFReq（频率）	信号频率，仅适用于时钟信号。
JPWidth（正脉宽）	上升沿与其后相邻的下降沿之间的时间差。
JNWidth（正脉宽）	下降沿与其后相邻的上升沿之间的时间差。
JPDuty（正占空比）	正脉宽与周期的比值，仅适用于时钟信号。
JNDuty（负占空比）	负脉宽与周期的比值，仅适用于时钟信号。
JCPRd（相邻周期抖动）	相邻两个周期的时间差，一般取测量统计中的“标准差”，仅适用于时钟信号。
JCPWidth（正脉宽周期抖动）	相邻两个正脉宽的时间差，仅适用于时钟信号。
JCNWidth（负脉宽周期抖动）	相邻两个负脉宽的时间差，仅适用于时钟信号。
JCPDuty（正占空比周期抖动）	相邻两个周期的正占空比之差。
JCNDuty（负占空比周期抖动）	相邻两个周期的负占空比之差。
JTIe	信号边沿到参考时钟边沿的时间差。
RJ	随机抖动，主要来自于热噪声、晶振的机械噪声等随机噪声。RJ 的 PDF（概率密度函数）服从高斯分布，测量样本值越多，测量值分布范围越宽，理论上当测量样本无限多时，测量值分布范围趋近于无穷大，因此它是无界（Unbounded）的，其大小通常用标准偏差σ来表征。此外由于 RJ 是随机的，因此它与信号不相关，与时间也是不相关的。
DJ	确定性抖动，其分布是有界的（Bounded），大小一般用峰峰值表征。
DCD	占空比相关抖动，主要来自于上升沿和下降沿的不对称。时钟的占空比不是 50%、上升沿和下降沿不相等以及判决电平的误差，都会引入 DCD。

	DDJ	数据相关抖动，主要来自于 ISI（码间串扰）。由于数字信号传输信道频率响应的非理想特性，当不同的码型通过信道时，会产生大小不一致的上升/下降沿，从而产生不同的过零点位置。DDJ 的 PDF 表现为若干离散的线分布。
	PJ	周期性抖动，主要由板上的周期信号干扰引起，包括电源纹波、时钟串扰等。周期性干扰直接形成对被测信号的相位调制。
	TJ（TJ@BER）	基于误码率估算的总抖动，来自于公式： $$TJ = DJ + 2Q_{BER} * \sigma_{RJ}$$ 其中 Q_{BER}的取值根据不同的误码率有所不同。在误码率=10^{-12}时，$Q_{BER} = 7.05$
返回格式	{JPRD|JFReq|JPWidth|JNWidth|JPDuty|JNDuty|JCPRd|JCPWidth|JCNWidth|JCPDuty|JCNDuty|JTIe|RJ|DJ|DCD|DDJ|PJ|TJ}	
示例	设置抖动分析测量第一个测量项为 TIE： *:JITTer:MEASure:P1:TYPE JTIE* *JITT:MEAS:P1:TYPE JTIE* 查询抖动分析测量第一个测量项的源和类型： *JITT:MEAS:P1:TYPE?* 返回值： *JTIe*	
关联命令	:JITTer:MEASure:P<n>	

:JITTer:MEASure:P<n>:VALue

描述	查询当前抖动分析指定测量项的值。
命令格式	:JITTer:MEASure:P<n>:VALue?
参数说明	<n>:= 测量项序号，整型 NR1 格式，取值范围［1,6］
返回格式	浮点型 NR3 格式
示例	查询第一个测量项的值： *JITTer:MEAS:P1:VAL?* 返回值： *-3.487E-12*
关联命令	:JITTer:MEASure:P<n>:TYPE

:JITTer:MEASure:P<n>:STATistics

描述	查询抖动分析指定测量项的统计信息。
命令格式	:JITTer:MEASure:P<n>:STATistics? <type>
参数说明	<n>:= 测量项序号，整型 NR1 格式，取值范围 [1,6] <type>:= {ALL|CURRent|MEAN|MAXimum|MINimum|STDev|COUNt} ● ALL：所有统计参数 ● CURRent：当前值 ● MEAN：平均值 ● MAXimum：最大值 ● MINimum：最小值 ● STDev：标准差 ● COUNt：统计次数
返回格式	浮点型 NR3 格式
示例	查询抖动分析第一个测量项的当前值： *:JITTer:MEASure:P1:STATistics? CURR* *JITT:MEAS:P1:STAT? CURR* 返回值： *-6.121E-12*

:JITTer:MEASure:P<n>:SHIStory

描述	查询抖动分析指定测量项的历史统计数据。仅在最大统计次数有限制时有效，无限制时则返回统计当前值。
命令格式	:JITTer:MEASure:P<n>:SHIStory? [<count>]
参数说明	<n>:= 测量项序号，整型 NR1 格式，取值范围 [1,6] <count>:= 历史统计数据的个数，整型 NR1 格式，取值范围受最大统计次数限制。 **注意**：不指定<count>时，返回所有历史数据。
返回格式	Count=<value>,<value1>,<value2>,...,<valueN> <value>:= 统计数据的个数，整型 NR1 格式 <valueN>:= 统计数据值，浮点型 NR3 格式
示例	当最大统计次数为 1024 时，查询抖动分析 P1 当前值前 1 个的测量数据： *:JITTer:MEASure:P1:SHIStory? 1* *JITT:MEAS:P1:SHIS? 1* 返回值： *Count=1,-3.333973E-11*

:JITTer:MEASure:CLEar

描述	清除抖动分析所有的测量项。
命令格式	:JITTer:MEASure:CLEar
示例	清除抖动分析所有的测量项： *:JITTer:MEASure:CLEar* *JITT:MEASV:CLE*
关联命令	:JITTer:MEASure:P<n> :JITTer:MEASure:P<n>:TYPE

:JITTer:PMODe

描述	设置或查询抖动分析分解数据码型长度计算方式。仅信号类型为数据时有效。
命令格式	:JITTer:PMODe <mode> :JITTer:PMODe?
参数说明	<mode>:= {AUTO[,<MINRepeats>]|MANual[,<MINRepeats>,<Lenth>]} ● AUTO：自动模式，根据输入信号自动设置码型，需指定<MINRepeats> ● MANual：手动模式，由用户设置码型，需指定<MINRepeats>,<Lenth> <MINRepeats>:=最小码型数量，整型 NR1 格式 <Lenth>:=码型长度，整型 NR1 格式
返回格式	{AUTO|MANual},<MINRepeats>,<Lenth>
示例	设置抖动分析的码型长度计算方式为自动、最小码型数量 100： *:JITTer:PMODe AUTO,100* *JITT:PMODe AUTO,100* 查询抖动分析的码型长度计算方式： *JITT:PMOD?* 返回值： *AUTO,100,2*

:JITTer:PREPeats

描述	查询抖动分析分解数据码型数量。
命令格式	:JITTer:PREPeats?
返回格式	整型 NR1 格式

示例	查询抖动分析码型数量： *JITT:PREPeats?* 返回值： *256*

:JITTer:RUN

描述	设置或查询抖动分析运行状态。
命令格式	:JITTer:RUN <state> :JITTer:RUN?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置抖动分析的运行状态为运行： *:JITTer:RUN ON* *JITT:RUN ON* 查询抖动分析的运行状态： *JITT:RUN?* 返回值： *ON*

5.17 MEASure 命令系统

:MEASURE 命令子系统控制自动测量。

:MEASure

描述	设置或查询测量开关状态。
命令格式	:MEASure <state> :MEASure?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启测量: *:MEASure ON* *MEAS ON* 查询测量开关状态: *MEAS?* 返回值: *ON*

:MEASure:ADVanced:CLEar

描述	清除所有的高级测量项。
命令格式	:MEASure:ADVanced:CLEar
示例	清除所有的高级测量项: *:MEASure:ADVanced:CLEar* *MEAS:ADV:CLE*
关联命令	:MEASure:ADVanced:P<n> :MEASure:ADVanced:P<n>:TYPE

:MEASure:ADVanced:LINenumber

描述	设置或查询高级测量 M2 模式下显示测量项总数。
命令格式	:MEASure:ADVanced:LINenumber <value> :MEASure:ADVanced:LINenumber?

参数说明	<value>:= 显示测量项的总数，整型 NR1 格式，取值范围 [1,12]
返回格式	整型 NR1 格式
示例	设置高级测量项显示的总数为 12： *:MEASure:ADVanced:LINenumber 12* *MEAS:ADV:LIN 12* 查询高级测量项显示的总数： *MEAS:ADV:LIN?* 返回值： *12*
关联命令	:MEASure:MODE

:MEASure:ADVanced:P<n>

描述	设置或查询指定高级测量项的开关状态。
命令格式	:MEASure:ADVanced:P<n> <state> :MEASure:ADVanced:P<n>?
参数说明	<n>:= 测量项序号，整型 NR1 格式，取值范围 [1,12] <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启第一个高级测量项： *:MEASure:ADVanced:P1 ON* *MEAS:ADV:P1 ON* 查询第一个高级测量项的开关状态： *MEAS:ADV:P1?* 返回值： *ON*
关联命令	:MEASure:ADVanced:P<n>:TYPE :MEASure:ADVanced:P<n>:SOURce<x>

:MEASure:ADVanced:P<n>:SOURce<x>

描述	设置或查询指定高级测量项的信源 A 或信源 B。测量类型为延迟测量项时，需设置信源 B。
命令格式	:MEASure:ADVanced:P<n>:SOURce<x> <source> :MEASure:ADVanced:P<n>:SOURce<x>?

参数说明	<n>:= 测量项序号，整型 NR1 格式，取值范围 [1,12] <n>:={1|2}，测量信源序号 <source>:={C<n>|Z<n>|F<x>|ZF<x>|M<m>|ZM<m>|D<d>|ZD<d>|REF<r>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● Z<n>:= 缩放波形，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● ZF<x>:= 缩放数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式 ● ZM<m>:= 缩放内存波形，<m>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式，取值范围 [0,15] ● ZD<d>:= 缩放数字通道，<d>:= 序号，整型 NR1 格式，取值范围 [0,15] ● REF<r>:= 参考波形，<r>:={A|B|C|D}，参考波形序号 **注意**: ● 当 Zoom 开启时，Z<n>作为源才有效 ● 通道延迟测量仅支持 C<n>、F<x>
返回格式	{C<n>|Z<n>|F<x>|ZF<x>|M<m>|ZM<m>|D<d>|ZD<d>|REF<r>}
示例	设置第一个高级测量项的信源 A 为 C1： *:MEASure:ADVanced:P1:SOURce1 C1* *MEAS:ADV:P1:SOUR1 C1* 查询第一个高级测量项的信源 A： *MEAS:ADV:P1:SOUR1?* 返回值： *C1*

:MEASure:ADVanced:P<n>:STATistics

描述	查询指定高级测量项的统计信息。
命令格式	:MEASure:ADVanced:P<n>:STATistics? <type>
参数说明	<n>:= 测量项序号，整型 NR1 格式，取值范围 [1,12] <type>:={ALL|CURRent|MEAN|MAXimum|MINimum|STDev|COUNt} ● ALL：所有统计参数 ● CURRent：当前值 ● MEAN：平均值 ● MAXimum：最大值 ● MINimum：最小值 ● STDev：标准差 ● COUNt：统计次数

返回格式	浮点型 NR3 格式
示例	查询高级测量第一个测量项的当前值： *MEAS:ADV:P1:STAT? CURR* 返回值： *6.7E-02*
关联命令	:MEASure:ADVanced:STATistics

:MEASure:ADVanced:P<n>:SHIStory

描述	查询指定高级测量项的历史统计数据。仅在最大统计次数有限制时有效，无限制时则返回统计当前值。
命令格式	:MEASure:ADVanced:P<n>:SHIStory? [<count>]
参数说明	<count>:= 历史统计数据的个数，整型 NR1 格式，取值范围受最大统计次数限制。 **注意**： ● 不指定<count>时，返回所有历史数据。
返回格式	Count=<value>,<value1>,<value2>,...,<valueN> <value>:= 统计数据的个数，整型 NR1 格式 <valueN>:= 统计数据值，浮点型 NR3 格式
示例	当最大统计次数为 1024 时，查询 P1 当前值前 5 个的测量数据： *:MEASure:ADVanced:P1:SHIStory? 5* *MEAS:ADV:P1:SHIS? 5* 返回值： *Count=5,4.223529E+00,4.221176E+00,4.221176E+00,* *4.214118E+00,4.223529E+00,*
关联命令	:MEASure:ADVanced:STATistics:MAXCount

:MEASure:ADVanced:P<n>:TYPE

描述	设置查询指定高级测量项的测量类型。
命令格式	:MEASure:ADVanced:P<n>:TYPE <parameter>
参数说明	<n>:= 测量项序号，整型 NR1 格式，取值范围 [1,12] <parameter>:=

{PKPK|MAX|MIN|AMPL|TOP|BASE|LEVELX|CMEAN|MEAN|STDEV|VSTD|RMS|CRMS|MEDIAN|CMEDIAN|OVSN|ULOWer|FPRE|OVSP|RPRE|PER|FREQ|TMAX|TMIN|PWID|NWID|DUTY|NDUTY|WID|NBWID|DELAY|TIMEL|RISE|FALL|RISE10T90|FALL90T10|CCJ|PAREA|NAREA|AREA|ABSAREA|CYCLES|REDGES|FEDGES|EDGES|PPULSES|NPULSES|PHA|SKEW|FRR|FRF|FFR|FFF|LRR|LRF|LFR|LFF|PACArea|NACArea|ACArea|ABSACArea|PSLOPE|NSLOPE|TSR|TSF|THR|THF|DTIMe1|DTIMe2|DTIMe3|DTIMe4}

表 5- 2 测量参数表

参数	描述
PKPK（峰峰值）	最大值和最小值之间的差值
MAX（最大值）	波形数据中幅度的最大值
MIN（最小值）	波形数据中幅度的最小值
AMPL（幅值）	顶端值与底端值的差值，如果不是双峰信号，则是最大值与最小值之差
TOP（顶端值）	双峰信号的最大平顶值
BASE（底端值）	双峰信号的最小平顶值
LEVELX（L@T）	触发点的电压值
CMEAN (周期平均值)	第一个周期的算术平均值
MEAN（平均值）	波形数据的算术平均值
STDEV（标准差）	所有数据的标准偏差
VSTD（周期标准差）	第一个周期内波形数据的标准偏差
RMS（均方根）	所有数据的均方根值
CRMS（周期均方根）	第一个周期内波形数据的均方根值
MEDIAN（中位数）	波形数据的中位数
CMEDIAN (周期中位数)	第一个周期内波形数据的中位数
OVSN（下降过激）	过激是边沿转换后的失真，以幅度的百分比表示
FPRE（下降前激）	前激是边沿转换前的失真，以幅度的百分比表示
OVSP（上升过激）	过激是边沿转换后的失真，以幅度的百分比表示
RPRE（上升前激）	前激是边沿转换前的失真，以幅度的百分比表示
ULOWer（高低值）	测量阈值的高值到低值之间差值
PER（周期）	两个连续、同极性边沿的中值交叉点之间的时间
FREQ（频率）	周期的倒数
TMAX（最大值时间）	信号最大值出现的时间

TMIN（最小值时间）	信号最小值出现的时间
PWID（正脉宽）	从脉冲上升沿的中值处到紧接着的一个下降沿的中值处之间的时间差
NWID（负脉宽）	从脉冲下降沿的中值处到紧接着的一个上升沿的中值处之间的时间差
DUTY（正占空比）	正脉宽与周期的比值
NDUTY（负占空比）	负脉宽与周期的比值
WID（正脉冲宽度）	过第一个上升沿中值的点与过最后一个下降沿中值的点间的时间
NBWID（负脉冲宽度）	过第一个下降沿中值的点与过最后一个上升沿中值的点间的时间
DELAY（延时）	触发点到第一个沿的中值处之间的时间差
TIMEL（T@M）	触发点到每个上升沿的中值点的时间
RISE（上升时间）	信号幅度从阈值电平低值上升至阈值电平高值所经历的时间
FALL（下降时间）	信号幅度从阈值电平高值下降至阈值电平低值所经历的时间
RISE10T90（10-90%上升时间）	信号幅度从 10%上升至 90%所经历的时间
FALL90T10（90-10%下降时间）	信号幅度从 90%下降至 10%所经历的时间
CCJ（相邻周期抖动）	两个相邻周期的差
PAREA（直流正面积）	屏幕内波形，值大于零的部分的面积之和
NAREA（直流负面积）	屏幕内波形，值小于零的部分的面积之和
AREA（直流有效面积）	屏幕内波形的有效面积
ABSAREA（直流绝对面积）	屏幕内波形的绝对面积
CYCLES（周期数）	屏幕内波形的周期数
EDGES（边沿总数）	屏幕内波形的边沿总数
REDGES（上升沿个数）	屏幕内波形的上升沿个数
FEDGES（下降沿个数）	屏幕内波形的下降沿个数
PPULSES（正脉冲数）	屏幕内波形的正脉冲数
NPULSES（负脉冲数）	屏幕内波形的负脉冲数
PHA（相位）	通道 A 和通道 B 的第一个上升沿的中值点间的

	相位差。该参数只有在能测量到通道 A 和通道 B 的完整周期时才有效
SKEW（时滞）	通道 A 边沿与最近的通道 B 边沿的时间差
FRR（FRFR）	通道 A 的第一个上升沿的中值点和通道 B 的第一个上升沿的中值点间的距离
FRF(FRFF)	通道 A 的第一个上升沿的中值点和通道 B 的第一个下降沿的中值点间的距离
FFR(FFFR)	通道 A 的第一个下降沿的中值点和通道 B 的第一个上升沿的中值点间的距离
FFF(FFFF)	通道 A 的第一个下降沿的中值点和通道 B 的第一个下降沿的中值点间的距离
LRR(FRLR)	通道 A 的第一个上升沿的中值点和通道 B 的最后一个上升沿的中值点间的距离
LRF(FRLF)	通道 A 的第一个上升沿的中值点和通道 B 的最后一个下降沿的中值点间的距离
LFR(FFLR)	通道 A 的第一个下降沿的中值点和通道 B 的最后一个上升沿的中值点间的距离
LFF(FFLF)	通道 A 的第一个下降沿的中值点和通道 B 的最后一个下降沿的中值点间的距离
PACArea（交流正面积）	屏幕内波形，值大于平均值的部分的面积之和
NACArea（交流负面积）	屏幕内波形，值小于平均值的部分的面积之和
ACArea（交流有效面积）	屏幕内交流分量的有效面积
ABSACArea（交流绝对面积）	屏幕内交流分量的绝对面积
PSLOPE（上升沿斜率）	上升沿阈值高电平与低电平之差除以上升时间
NSLOPE（下降沿斜率）	下降沿阈值高电平与低电平之差除以下降时间
TSR	数据相位时钟上升沿的建立时间
TSF	数据相位时钟下降沿的建立时间
THR	数据相位时钟上升沿的保持时间
THF	数据相位时钟下降沿的保持时间
DTIMe1-DTIMe4	指定的两个沿之间的时间差

返回格式 {PKPK|MAX|MIN|AMPL|TOP|BASE|LEVELX|CMEAN|MEAN|STDEV|VSTD|RMS|CRMS|MEDIAN|CMEDIAN|OVSN|FPRE|OVSP|ULOW|RPRE|PER|FREQ|TMAX|TMIN|PWID|NWID|DUTY|NDUTY|WID|NBWID|DELAY|TIMEL|RISE|FALL|RISE10T90|FALL90T10|CCJ|PAREA|NAREA|AREA|ABSAREA|CYCLES|REDGES|FEDGE

	S|EDGES|PPULSES|NPULSES|PHA|SKEW|FRR|FRF|FFR|FFF|LRR|LRF|LFR|LFF|PACArea|NACArea|ACArea|ABSACArea|PSLOPE|NSLOPE|TSR|TSF|THR|THF|DTIMe1|DTIMe2|DTIMe3|DTIMe4}
示例	设置高级测量第一个测量项测量类型为最大值： *:MEASure:ADVanced:P1:TYPE MAX* *MEAS:ADV:P1:TYPE MAX* 查询高级测量第一个测量项的当前值： *MEAS:ADV:P1:TYPE?* 返回值： *MAX*
关联命令	:MEASure:ADVanced:P<n>

:MEASure:ADVanced:P<n>:VALue

描述	查询当前指定高级测量项的值。
命令格式	:MEASure:ADVanced:P<n>:VALue?
参数说明	<n>:= 测量项序号，整型 NR1 格式，取值范围 [1,12]
返回格式	浮点型 NR3 格式
示例	查询第一个测量项的值： *MEAS:ADV:P1:VAL?* 返回值： *4.033E+00*
关联命令	:MEASure:ADVanced:P<n>:TYPE

:MEASure:ADVanced:STATistics

描述	设置或查询高级测量统计的开关状态。
命令格式	:MEASure:ADVanced:STATistics <state> :MEASure:ADVanced:STATistics?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启高级测量统计： *:MEASure:ADVanced:STATistics ON* *MEAS:ADV:STAT ON* 查询高级测量统计的开关状态：

	MEAS:ADV:STAT? 返回值： *ON*
关联命令	:MEASure:ADVanced:P<n>:STATistics

:MEASure:ADVanced:STATistics:AIMLimit

描述	设置或查询高级测量统计的 AIM 次数。
命令格式	:MEASure:ADVanced:STATistics:AIMLimit <value> :MEASure:ADVanced:STATistics:AIMLimit?
参数说明	<value>:= AIM 次数，整型 NR1 格式，该值取值范围和机型相关
返回格式	整型 NR1 格式
示例	设置测量统计的 AIM 限制值为 100： *:MEASure:ADVanced:STATistics:AIMLimit 100* *MEAS:ADV:STAT:AIML 100* 查询测量统计的 AIM 限制值： *MEAS:ADV:STAT:AIML?* 返回值： *100*
关联命令	:MEASure:ADVanced:P<n>:STATistics

:MEASure:ADVanced:STATistics:HISTOGram

描述	设置或查询高级测量统计直方图开关状态。
命令格式	:MEASure:ADVanced:STATistics:HISTOGram <state> :MEASure:ADVanced:STATistics:HISTOGram?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启测量统计直方图： *:MEASure:ADVanced:STATistics:HISTOGram ON* *MEAS:ADV:STAT:HISTOG ON* 查询测量统计直方图开关状态： *MEAS:ADV:STAT:HISTOG?* 返回值： *ON*

关联命令	:MEASure:ADVanced:STATistics

:MEASure:ADVanced:STATistics:MAXCount

描述	设置或查询高级测量的最大统计次数。
命令格式	:MEASure:ADVanced:STATistics:MAXCount <value> :MEASure:ADVanced:STATistics:MAXCount?
参数说明	<value>:= 最大统计次数，整型 NR1 格式。取值范围 [0,1024]，其中 0 表示统计次数无限制。
返回格式	整型 NR1 格式
示例	设置测量统计的最大次数为 1024： *:MEASure:ADVanced:STATistics:MAXCount 1024* *MEAS:ADV:STAT:MAXC 1024* 查询测量统计的最大次数： *MEAS:ADV:STAT:MAXC?* 返回值： *1024*
关联命令	:MEASure:ADVanced:STATistics

:MEASure:ADVanced:STATistics:RECord

描述	设置或查询测量统计设置下的记录测量次数。
命令格式	:MEASure:ADVanced:STATistics:RECord <num> :MEASure:ADVanced:STATistics:RECord?
参数说明	<num>:= 数值，整型 NR1 格式。范围因机型而异，有关详细信息可参阅用户手册
返回格式	整型 NR1 格式
示例	设置高级测量统计记录测量结果的数目为 2000： *:MEASure:ADVanced:STATistics:RECord 2000* *MEAS:ADV:STAT:REC 2000* 查询高级测量统计记录测量结果的数目： *MEAS:ADV:STAT:REC?* 返回值： *2000*

:MEASure:ADVanced:STATistics:RESet

描述	重置高级测量的统计结果。
命令格式	:MEASure:ADVanced:STATistics:RESet
示例	重置高级测量的统计结果： *:MEASure:ADVanced:STATistics:RESet* *MEAS:ADV:STAT:RES*
关联命令	:MEASure:ADVanced:STATistics

:MEASure:ADVanced:STYLe

描述	设置或查询高级测量的显示模式。
命令格式	:MEASure:ADVanced:STYLe <type> :MEASure:ADVanced:STYLe?
参数说明	<type>:= {M1|M2} ● M1：垂直显示测量统计数据（含直方图） ● M2：水平显示测量统计数据，无法显示统计直方图
返回格式	{M1|M2}
示例	设置高级测量的显示模式为 M1： *:MEASure:ADVanced:STYLe M1* *MEAS:ADV:STYLe M1* 查询高级测量的显示模式： *MEAS:ADV:STYLe?* 返回值： *M1*

:MEASure:ASTRategy

描述	设置或查询测量的幅值计算策略。
命令格式	:MEASure:ASTRategy <type> :MEASure:ASTRategy?
参数说明	<type>:= {AUTO|MANual} ● AUTO：自动。输入信号自动选择幅度计算策略以保证测量值的准确 ● MANual：手动。手动选择顶端值和底端值计算策略（直方图或最值）
返回格式	{AUTO|MANual}

示例	设置测量的幅值计算策略为手动： *:MEASure:ASTRategy MANual* *MEAS:ASTR MAN* 查询测量的幅值计算策略： *MEAS:ASTR?* 返回值： *MANual*

:MEASure:ASTRategy:BASE

描述	当幅值计算策略为手动时，设置或查询底端值计算策略。
命令格式	:MEASure:ASTRategy:BASE <type> :MEASure:ASTRategy:BASE?
参数说明	<type>:= {HISTogram|MIN} ● HISTogram：直方图。对小于峰峰值 1/2 的值进行统计，概率最大的值认定为底端值 ● MIN：最小值。将波形最小值认定为底端值
返回格式	{HISTogram|MIN}
示例	设置测量的底端值计算策略为直方图： *:MEASure:ASTRategy:BASE HISTogram* *MEAS:ASTR:BASE HIST* 查询测量的底端值计算策略： *MEAS:ASTR:BASE?* 返回值： *HISTogram*

:MEASure:ASTRategy:TOP

描述	当幅值计算策略为手动时，设置或查询顶端值计算策略。
命令格式	:MEASure:ASTRategy:TOP <type> :MEASure:ASTRategy:TOP?
参数说明	<type>:= {HISTogram|MAX} ● HISTogram：直方图。对大于峰峰值 1/2 的值进行统计，概率最大的值认定为顶端值 ● MAX：最大值。将波形最大值认定为顶端值
返回格式	{HISTogram|MAX}

示例	设置测量的顶端值计算策略为直方图： *:MEASure:ASTRategy:TOP HISTogram* *MEAS:ASTR:TOP HIST* 查询测量的顶端值计算策略： *MEAS:ASTR:TOP?* 返回值： *HISTogram*

:MEASure:DTIMe<n>:EDGE1

描述	设置或查询指定△Time 的开始沿。
命令格式	:MEASure:DTIMe<n>:EDGE1 <value> :MEASure:DTIMe<n>:EDGE1?
参数说明	<n>:= △Time 序号，整型 NR1 格式。取值范围 [1,4] <value>:= 开始沿，整型 NR1 格式。取值范围受最大 AIM 次数限制。设置为 -1 时，表示最后一个沿
返回格式	整型 NR1 格式
示例	设置△Time1 的开始沿为最后一个沿： *:MEASure:DTIMe1:EDGE1 -1* *MEAS:DTIM1:EDGE1 -1* 查询△Time1 的开始沿： *MEAS:DTIM1:EDGE1?* 返回值： *-1*
关联命令	:MEASure:ADVanced:STATistics:AIMLimit :MEASure:DTIMe<n>:EDGE2

:MEASure:DTIMe<n>:EDGE2

描述	设置或查询指定△Time 的截止沿。
命令格式	:MEASure:DTIMe<n>:EDGE2 <value> :MEASure:DTIMe<n>:EDGE2?
参数说明	<n>:= △Time 序号，整型 NR1 格式。取值范围 [1,4] <value>:= 截止沿，整型 NR1 格式。取值范围受最大 AIM 次数限制。设置为 -1 时，表示最后一个沿
返回格式	整型 NR1 格式
示例	设置△Time1 的截止沿为最后一个沿：

	:MEASure:DTIMe1:EDGE2 -1 *MEAS:DTIM1:EDGE2 -1* 查询△Time1 的截止沿： *MEAS:DTIM1:EDGE2?* 返回值： *-1*
关联命令	:MEASure:ADVanced:STATistics:AIMLimit :MEASure:DTIMe<n>:EDGE1

:MEASure:DTIMe<n>:SLOPe1

描述	设置或查询指定△Time 开始沿类型。
命令格式	:MEASure:DTIMe<n>:SLOPe1 <slope_type> :MEASure:DTIMe<n>:SLOPe1?
参数说明	<n>:= △Time 序号，整型 NR1 格式。取值范围 [1,4] <slope_type>:={RISing|FALLing}
返回格式	{RISing|FALLing}
示例	设置△Time1 的开始沿类型为上升沿： *:MEASure:DTIMe1:SLOPe1 RISing* *MEAS:DTIM1:SLOP1 RIS* 查询△Time1 的开始沿类型： *MEAS:DTIM1:SLOP1?* 返回值： *RISing*
关联命令	:MEASure:DTIMe<n>:SLOPe2

:MEASure:DTIMe<n>:SLOPe2

描述	设置或查询指定△Time 的截止沿类型。
命令格式	:MEASure:DTIMe<n>:SLOPe2 <slope_type> :MEASure:DTIMe<n>:SLOPe2?
参数说明	<n>:= △Time 序号，整型 NR1 格式。取值范围 [1,4] <slope_type>:={RISing|FALLing}
返回格式	{RISing|FALLing}
示例	设置△Time1 的截止沿类型为上升沿： *:MEASure:DTIMe1:SLOPe2 RISing*

	MEAS:DTIM1:SLOP2 RIS 查询△Time1 的截止沿类型： *MEAS:DTIM1:SLOP2?* 返回值： *RISing*
关联命令	:MEASure:DTIMe<n>:SLOPe1

:MEASure:DTIMe<n>:THReshold1

描述	设置或查询指定△Time 的开始沿阈值。
命令格式	:MEASure:DTIMe<n>:THReshold1 <type> :MEASure:DTIMe<n>:THReshold1?
参数说明	<n>:= △Time 序号，整型 NR1 格式。取值范围 [1,4] <type>:= {UPPer|MIDDle|LOWer}
返回格式	{UPPer|MIDDle|LOWer}
示例	设置△Time1 的开始沿阈值为高值： *:MEASure:DTIMe1:THReshold1 UPPer* *MEAS:DTIM1:THR1 UPP* 查询△Time1 的开始沿阈值： *MEAS:DTIM1:THR1?* 返回值： *UPPer*
关联命令	:MEASure:DTIMe<n>:THReshold2

:MEASure:DTIMe<n>:THReshold2

描述	设置或查询指定△Time 的截止沿阈值。
命令格式	:MEASure:DTIMe<n>:THReshold2 <type> :MEASure:DTIMe<n>:THReshold2?
参数说明	<n>:= △Time 序号，整型 NR1 格式。取值范围 [1,4] <type>:= {UPPer|MIDDle|LOWer}
返回格式	{UPPer|MIDDle|LOWer}
示例	设置△Time1 的截止沿阈值为高值： *:MEASure:DTIMe1:THReshold2 UPPer* *MEAS:DTIM1:THR2 UPP*

	查询△Time1 的截止沿阈值： *MEAS:DTIM1:THR2?* 返回值： *UPPer*
关联命令	:MEASure:DTIMe<n>:THReshold1

:MEASure:GATE

描述	设置或查询测量门限开关状态。
命令格式	:MEASure:GATE <state> :MEASure:GATE?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启测量门限： *:MEASure:GATE ON* *MEAS:GATE ON* 查询测量门限开关状态： *MEAS:GATE?* 返回值： *ON*
关联命令	:MEASure:GATE:GA :MEASure:GATE:GB

:MEASure:GATE:GA

描述	设置或查询测量门限 A 的位置。
命令格式	:MEASure:GATE:GA <value> :MEASure:GATE:GA?
参数说明	<value>:= 门限 A 的位置值，浮点型 NR3 格式，取值范围 [-水平格数/2*当前时基+水平延时，水平格数/2*当前时基+水平延时] **注意**：GA 的值不能大于 GB 的值。如果设置的值大于 GB，它将自动设置为与 GB 相同的值。
返回格式	浮点型 NR3 格式
示例	设置测量门限 GA 为-100ns： *:MEASure:GATE:GA -1.00E-07* *MEAS:GATE:GA -1.00E-07*

	查询测量门限 GA： *MEAS:GATE:GA?* 返回值： *-1.00E-07*
关联命令	:MEASure:GATE :MEASure:GATE:GB

:MEASure:GATE:GB

描述	设置或查询测量门限 B 的位置。
命令格式	:MEASure:GATE:GB <value> :MEASure:GATE:GB?
参数说明	<value>:= 门限 B 的位置值，浮点型 NR3 格式，取值范围 [-水平格数/2*当前时基+水平延时，水平格数/2*当前时基+水平延时] **注意**：GB 的值不能小于 GA 的值。如果设置的值小于 GA，它将自动设置为与 GA 相同的值。
返回格式	浮点型 NR3 格式
示例	设置测量门限 GB 为 100ns： *:MEASure:GATE:GB 1.00E-07* *MEAS:GATE:GB 1.00E-07* 查询测量门限 GB： *MEAS:GATE:GB?* 返回值： *1.00E-07*
关联命令	:MEASure:GATE :MEASure:GATE:GA

:MEASure:MODE

描述	设置或查询测量类型。
命令格式	:MEASure:MODE <type> :MEASure:MODE?
参数说明	<type>:= {SIMPle|ADVanced} ● SIMPle：基本测量。显示指定源的所有测量项的测量值，测量项支持垂直测量、水平测量和混合测量。

	● ADVanced：高级测量。显示不同源、不同测量项的测量值。该模式下也可显示统计结果、直方图。
返回格式	{SIMPle|ADVanced}
示例	设置测量类型为基本测量： *:MEASure:MODE SIMPle* *MEAS:MODE SIMP* 查询测量类型： *MEAS:MODE?* 返回值： *SIMPle*

:MEASure:RDISplay

描述	设置或查询测量结果的显示样式。
命令格式	:MEASure:RDISplay <type> :MEASure:RDISplay?
参数说明	<type>:= {EMBedded|FLOating} ● EMBedded：测量结果内嵌进屏幕，会压缩波形区域。 ● FLOating：测量结果悬浮于屏幕，不会压缩波形区域。
返回格式	{EMBedded|FLOating}
示例	设置测量结果显示样式为悬浮： *:MEASure:RDISplay FLOating* *MEAS:RDIS FLO* 查询测量类型： *MEAS:RDIS?* 返回值： *FLOating*

:MEASure:SIMPle:CLEar

描述	清除所有的基本测量项。
命令格式	:MEASure:SIMPle:CLEar
示例	清除所有的基本测量项： *:MEASure:SIMPle:CLEar* *MEAS:SIMP:CLE*

关联命令	:MEASure:SIMPle:ITEM

:MEASure:SIMPle:ITEM

描述	设置基本测量项的测量类型。
命令格式	:MEASure:SIMPle:ITEM <parameter>,<state>
参数说明	<parameter>:= {PKPK|MAX|MIN|AMPL|TOP|BASE|LEVELX|CMEAN|MEAN|STDEV|VSTD|RMS|CRMS|MEDIAN|CMEDIAN|OVSN|FPRE|OVSP|RPRE|ULOWer|PER|FREQ|TMAX|TMIN|PWID|NWID|DUTY|NDUTY|WID|NBWID|DELAY|TIMEL|RISE|FALL|RISE20T80|FALL80T20|CCJ|PAREA|NAREA|AREA|ABSAREA|CYCLES|REDGES|FEDGES|EDGES|PPULSES|NPULSES|PACArea|NACArea|ACArea|ABSACArea} <state>:= {ON|OFF} **注意**：相关参数描述见“表 5- 2 测量参数表”。
返回格式	{PKPK|MAX|MIN|AMPL|TOP|BASE|LEVELX|CMEAN|MEAN|STDEV|VSTD|RMS|CRMS|MEDIAN|CMEDIAN|OVSN|FPRE|OVSP|RPRE|ULOWer|PER|FREQ|TMAX|TMIN|PWID|NWID|DUTY|NDUTY|WID|NBWID|DELAY|TIMEL|RISE|FALL|RISE20T80|FALL80T20|CCJ|PAREA|NAREA|AREA|ABSAREA|CYCLES|REDGES|FEDGES|EDGES|PPULSES|NPULSES|PACArea|NACArea|ACArea|ABSACArea}
示例	设置测量中开启最大值： *:MEASure:SIMPle:ITEM MAX,ON* *MEAS:SIMP:ITEM MAX,ON*
关联命令	:MEASure:SIMPle:VALue

:MEASure:SIMPle:SOURce

描述	设置或查询基本测量的源。
命令格式	:MEASure:SIMPle:SOURce <source> :MEASure:SIMPle:SOURce?
参数说明	<source>:= {C<n>|Z<n>|F<x>|ZF<x>|M<m>|ZM<m>|D<d>|ZD<d>|REF<r>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● Z<n>:= 缩放波形，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● ZF<x>:= 缩放数学函数，<x>:= 序号，整型 NR1 格式

	● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式 ● ZM<m>:= 缩放内存波形，<m>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式 ● ZD<d>:= 缩放数字通道，<d>:= 序号，整型 NR1 格式 ● REF<r>:= 参考波形，<r>:= {A|B|C|D}，参考波形序号 **注意**：当 Zoom 开启时，Z<n>、ZF<x>和 ZM<m>作为源才有效。
返回格式	{C<n>|Z<n>|F<x>|ZF<x>|M<m>|ZM<m>|D<d>|ZD<d>|REF<r>}
示例	设置基本测量源为 C1： *:MEASure:SIMPle:SOURce C1* *MEAS:SIMP:SOUR C1* 查询基本测量源： *MEAS:SIMP:SOUR?* 返回值： *C1*

:MEASure:SIMPle:VALue

描述	查询基本测量中指定测量项的值。
命令格式	:MEASure:SIMPle:VALue? <type>
参数说明	<type>:= {PKPK|MAX|MIN|AMPL|TOP|BASE|LEVELX|CMEAN|MEAN|STDEV|VSTD|RMS|CRMS|MEDIAN|CMEDIAN|OVSN|FPRE|OVSP|RPRE|ULOWer|PER|FREQ|TMAX|TMIN|PWID|NWID|DUTY|NDUTY|WID|NBWID|DELAY|TIMEL|RISE|FALL|RISE20T80|FALL80T20|CCJ|PAREA|NAREA|AREA|ABSAREA|CYCLES|REDGES|FEDGES|EDGES|PPULSES|NPULSES|PACArea|NACArea|ACArea|ABSACArea|ALL} **注意**： ● 相关参数描述见“表 5- 2 测量参数表”。 ● ALL 查询所有基本测量项的值。
返回格式	浮点型 NR3 格式
示例	查询基本测量的最大值： *MEAS:SIMP:VAL? MAX* 返回值： *2.000E+00*
关联命令	:MEASure:SIMPle:ITEM

:MEASure:THReshold:SOURce

描述	设置或查询测量阈值的源。
命令格式	:MEASure:THReshold:SOURce <source> :MEASure:THReshold:SOURce?
参数说明	<source>:= {C<n>|Z<n>|F<x>|ZF<x>|M<m>|ZM<m>|REF<r>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● Z<n>:= 缩放波形，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● ZF<x>:= 缩放数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式 ● ZM<m>:= 缩放内存波形，<m>:= 序号，整型 NR1 格式 ● REF<r>:= 参考波形，<r>:= {A|B|C|D}，参考波形序号 **注意**：Zoom 开启时，Z<n>、ZF<x>和 ZM<m>作为源才有效。
返回格式	{C<n>|Z<n>|F<x>|ZF<x>|M<m>|ZM<m>|REF<r>}
示例	设置测量阈值的源为 C1： *:MEASure:THReshold:SOURce C1* *MEAS:THR:SOUR C1* 查询测量阈值的源： *MEAS:THR:SOUR?* 返回值： *C1*

:MEASure:THReshold:TYPE

描述	设置或查询测量阈值的类型。
命令格式	:MEASure:THReshold:TYPE <type> :MEASure:THReshold:TYPE?
参数说明	<type>:= {PERCent|ABSolute} ● PERCent：百分比。按照波形的百分比进行设置。低值和高值的设置范围为 1%~99%，其中低值不得高于中值和高值。 ● ABSolute：绝对值。按照垂直方向范围进行设置。绝对阈值取决于通道档位、垂直位移、探头系数。在设置绝对阈值之前，应首先设置这些值。低值和高值限制在屏幕上的数值范围之内。如果任一绝对阈值高于或低于最小或最大波形值，测量可能就会无效。
返回格式	{PERCent|ABSolute}

示例	设置测量阈值的类型为百分比： *:MEASure:THReshold:TYPE PERCent* *MEAS:THR:TYPE PERC* 查询测量阈值的类型： *MEAS:THR:TYPE?* 返回值： *PERCent*

:MEASure:THReshold:ABSolute

描述	当阈值类型设置为绝对值时，设置或查询测量阈值。
命令格式	:MEASure:THReshold:ABSolute <high>,<mid>,<low> :MEASure:THReshold:ABSolute?
参数说明	<high>,<mid>,<low>:= 测量阈值的高/中/低值，浮点型 NR3 格式，取值范围和当前阈值源的垂直档位和垂直偏移相关
返回格式	<high>,<mid>,<low>
示例	设置绝对类型的测量阈值为 3V、1V、-1.5V： *:MEASure:THReshold:ABSolute 3.00,1.00,-1.50* *MEAS:THR:ABS 3.00,1.00,-1.50* 查询绝对类型的测量阈值： *MEAS:THR:ABS?* 返回值： *3.00 E+00,1.00 E+00,-1.50E+00*
关联命令	:MEASure:THReshold:TYPE

:MEASure:THReshold:PERCent

描述	当阈值类型设置为百分比时，设置或查询测量阈值。
命令格式	:MEASure:THReshold:PERCent <high>,<mid>,<low> :MEASure:THReshold:PERCent?
参数说明	<high>,<mid>,<low>:= 测量阈值的高/中/低百分比，整型 NR1 格式 <high>:= 取值范围 [3,99] <mid>:= 取值范围 [2,98] <low>:= 取值范围 [1,97]
返回格式	<high>,<mid>,<low>

示例	设置百分比类型的测量阈值为 80%、45%、10%： *:MEASure:THReshold:PERCent 80,45,10* *MEAS:THR:PERC 80,45,10* 查询百分比类型的测量阈值： *MEAS:THR:PERC?* 返回值： *80,45,10*
关联命令	:MEASure:THReshold:TYPE

5.18 MEMory 命令系统

:MEMory 命令子系统控制内存波形的保存、调出以及参数调整。

:MEMory<m>:HORizontal:POSition

描述	设置或查询内存波形的水平位置（偏移）。
命令格式	:MEMory<m>:HORizontal:POSition <val> :MEMory<m>:HORizontal:POSition?
参数说明	<m>:= 内存波形序号，整型 NR1 格式 <val>:= 水平位置值，浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置内存波形 2 的水平位置为 10us： *:MEMory2:HORizontal:POSition 1.00E-05* *MEM2:HOR:POS 1.00E-05* 查询内存波形 2 的水平位置： *MEM2:HOR:POS?* 返回值： *1.00E-05*

:MEMory<m>:HORizontal:SCALe

描述	设置或查询内存波形的水平档位。
命令格式	:MEMory<m>:HORizontal:SCALe <value> :MEMory<m>:HORizontal:SCALe?
参数说明	<m>:= 内存波形序号，整型 NR1 格式 <value>:= 水平档位，浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置内存波形 2 的水平档位为 100 ns/div： *:MEMory2:HORizontal:SCALe 1.00E-07* *MEM2:HOR:SCAL 1.00E-07* 查询内存波形 2 的水平档位： *MEM2:HOR:SCAL?* 返回值： *1.00E-07*

:MEMory<m>:HORizontal:SYNC

描述	设置或查询内存波形同步源。
命令格式	:MEMory<m>:HORizontal:SYNC <group> :MEMory<m>:HORizontal:SYNC?
参数说明	<m>:= 内存波形序号，整型 NR1 格式 <group>:={OFF|MTIMebase|MGRoup1|MGRoup2} ● OFF:= 关闭同步功能 ● MTIMebase:= 与主时基同步 ● MGRoup1:= 与内存波形组 1 同步 ● MGRoup2:= 与内存波形组 2 同步
返回格式	{OFF|MTIMebase|MGRoup1|MGRoup2}
示例	设置内存波形 2 与内存波形组 1 同步： *:MEMory2:HORizontal:SYNC MGRoup1* *MEM2:HOR:SYNC MGR1* 查询内存波形 2 的同步组： *MEM2:HOR:SYNC?* 返回值： *MGRoup1*

:MEMory<m>:IMPort

描述	指定源导入内存波形。
命令格式	:MEMory<m>:IMPort <source>
参数说明	<m>:= 内存波形序号，整型 NR1 格式 <source>:={C<n>|Z<n>|F<x>|M<m>|<path>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● Z<n>:= 缩放波形，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式 ● <path>:= 带引号的路径字符串，文件扩展名为“.bin”
返回格式	{C<n>|Z<n>|F<x>| M<m>|<path>}
示例	将 C2 的波形数据导入内存波形 2： *:MEMory2:IMPort C2* *MEM2:IMP C2*

:MEMory<m>:LABel

描述	设置或查询内存波形标签开关状态。
命令格式	:MEMory<m>:LABel <state> :MEMory<m>:LABel?
参数说明	<m>:= 内存波形序号，整型 NR1 格式 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启内存波形 2 的标签： *:MEMory2:LABel ON* *MEM2:LAB ON* 查询内存波形 2 的标签开关状态： *MEM2:LAB?* 返回值： *ON*

:MEMory<m>:LABel:TEXT

描述	设置或查询内存波形的标签字符。
命令格式	:MEMory<m>:LABel:TEXT <qstring> :MEMory<m>:LABel:TEXT?
参数说明	<m>:= 内存波形序号，整型 NR1 格式 <qstring>:= 带引号的 ASCII 文本字符串，字符串的长度限制为 20
返回格式	带引号的 ASCII 文本字符串
示例	设置内存波形 2 的标签内容为“MATH”： *:MEMory2:LABel:TEXT "MATH"* *MEM2:LAB:TEXT "MATH"* 查询内存波形 2 的标签内容： *MEM2:LAB:TEXT?* 返回值： *"MATH"*

:MEMory<m>:SWITch

描述	设置或查询内存波形显示开关状态。
命令格式	:MEMory<m>:SWITch <state> :MEMory<m>:SWITch?
参数说明	<m>:= 内存波形序号，整型 NR1 格式 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启内存波形 2： *:MEMory2:SWITch ON* *MEM2:SWIT ON* 查询内存波形 2 的开关状态： *MEM2:SWIT?* 返回值： *ON*

:MEMory<m>:VERTical:POSition

描述	设置或查询内存波形的垂直位置（偏移）。
命令格式	:MEMory<m>:VERTical:POSition <offset> :MEMory<m>:VERTical:POSition?
参数说明	<m>:= 内存波形序号，整型 NR1 格式 <offset>:= 垂直位置值，浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置内存波形 2 的垂直位置为 100mV： *:MEMory2:VERTical:POSition 1.00E-01* *MEM2:VERT:POS 1.00E-01* 查询内存波形 2 的垂直位置： *MEM2:VERT:POS?* 返回值： *1.00E-01*

:MEMory<m>:VERTical:SCALe

描述	设置或查询内存波形的垂直档位。
命令格式	:MEMory<m>:VERTical:SCALe <scale> :MEMory<m>:VERTical:SCALe?
参数说明	<m>:= 内存波形序号，整型 NR1 格式 <scale>:= 垂直档位，浮点型 NR3 格式
返回格式	浮点型 NR3 格式
示例	设置内存波形 2 的垂直档位为 10V： *:MEMory2:VERTical:SCALe 1.00E+01* *MEM2:VERT:SCAL 1.00E+01* 查询内存波形 2 的垂直档位： *MEM2:VERT:SCAL?* 返回值： *1.00E+01*

5.19 METEr 命令系统

METEr 命令子系统仅适用于手持式数字示波表 SHS800X/SHS1000X 的万用表功能，支持万用表配置和测量结果查询。

MMETer

描述	设置万用表开关状态。
命令格式	MMETer <switch>
参数说明	<switch>:= {ON|OFF}
示例	设置万用表开启： *MMETer ON*

READ

描述	查询测量值。
命令语法	READ?
返回格式	MM_VALUE <value>
示例	查询测量值： *READ?* 返回值： *MM_VALUE 0.00V*

CONFigure

CONFigure 命令使示波器恢复默认测量配置值，是配置测量的最简便方法。这些命令不会自动开始测量，配置默认值之后，还可以在正式启动测量之前修改测量属性。

CONFigure

描述	查询当前测量功能和测量值。当前测量功能名称以短格式返回，例如 ACV（交流电压测量）。
命令格式	CONFigure?
返回格式	{DCV|ACV|RES|DIODE|CONTINUITY|CAP|DCI|ACI}
示例	查询当前测量功能和测量值： *CONFigure?* 返回值： *DCV -04.mV*

CONFigure:CONTinuity

描述	设置通断测量下的所有测量参数和触发参数为默认值。 READ? 和 MEASure:CONTinuity? 查询测量电阻值。如果电阻大于 600Ω，界面显示“Overload”，并返回“Overload”。
命令格式	CONFigure:CONTinuit
示例	设置进行连通性测量： *CONF:CONT* 查询测量值： *READ?* 返回值： *Overload*

CONFigure:CURRent:AC

描述	设置交流电流测量下的所有测量参数和触发参数为默认值，可通过传入参数指定量程。 ● 指定为自动量程时，调整量程根据输入信号方便地为每个测量选择量程。自动调整量程可能需要更多的时间进行量程选择。进行快速测量时，使用固定量程。

	● 如果输入信号值大于指定量程，界面显示“Overload”，并返回“Overload”。 ● 使用 READ?读取测量值。
命令格式	CONFigure:CURRent:AC <range>
参数说明	<range>:= {60mA|600mA|6A|10A|AUTO|MIN|MAX|DEF} ● 60mA|600mA|6A|10A：设置固定量程为 60mA|600mA|6A|10A ● AUTO：自动量程 ● MIN：最小量程 ● MAX：最大量程 ● DEF：默认设置为自动量程
示例	设置使用 6A 的量程测量交流电流值： *CONF:CURR:AC 6A* 查询当前测量值： *READ?* 返回值： *+4.32133675E-04*

CONFigure:CURRent:DC

描述	设置直流电流测量下的所有测量参数和触发参数为默认值，可通过传入参数指定量程。
命令格式	CONFigure:CURRent:DC <range>
参数说明	<range>:= {60mA|600mA|6A|10A|AUTO|MIN|MAX|DEF} ● 60mA|600mA|6A|10A：设置固定量程为 60mA|600mA|6A|10A ● AUTO：自动量程 ● MIN：最小量程 ● MAX：最大量程 ● DEF：默认量程，设置为自动量程
示例	设置使用 6A 的量程测量直流电流值： *CONF:CURR:DC 6A* 查询当前测量值： *READ?* 返回值： *+4.32133675E-04*

CONFigure:DIODe

描述	设置二极管测量下的所有测量参数和触发参数为默认值。 ● 二极管测量的量程固定为 2 VDC。 ● READ? 和 MEASure:DIODe? 查询测量电压。如果超出量程 2V，界面显示“Overload”，并返回“Overload”。 ● 使用 READ? 读取测量值。
命令格式	CONFigure:DIOD
示例	设置进行二极管测量： *CONF:DIOD* 查询当前测量值： *READ?* 返回值： *Overload*

CONFigure:RESistance

描述	设置二线电阻测量下的所有测量参数和触发参数为默认值，可通过传入参数指定量程。 ● 指定为自动量程时，调整量程根据输入信号方便地为每个测量选择量程。自动调整量程可能需要更多的时间进行量程选择。进行快速测量时，使用固定量程。 ● 如果输入信号值大于指定量程，界面显示“Overload”，并返回“Overload”。 ● 使用 READ?读取测量值。
命令格式	CONFigure:RESistance <range>
参数说明	<range>:= {600|6k|60k|600k|6M|60M|AUTO|MIN|MAX|DEF} ● 600|6k|60k|600k|6M|60M：设置固定量程为 600|6k|60k|600k|6M|60M ● AUTO：自动量程 ● MIN：最小量程 ● MAX：最大量程 ● DEF：默认量程，设置为自动量程
示例	设置使用 600Ω 的量程测量二线电阻值： *CONF:RES 600* 查询当前测量值： *READ?* 返回值： *+6.71881065E+01*

CONFigure[:VOLTage]:AC

描述	设置交流电压测量下的所有测量参数和触发参数为默认值，可通过传入参数指定量程。 ● 指定为自动量程时，调整量程根据输入信号方便地为每个测量选择量程。自动调整量程可能需要更多的时间进行量程选择。进行快速测量时，使用固定量程。 ● 如果输入信号值大于指定量程，界面显示“Overload”，并返回“Overload”。 ● 使用 READ?读取测量值。
命令格式	CONFigure[:VOLTage]:AC <range>
参数说明	<range>:= 交流电压测量量程，各机型可设置量程范围请参考下表： **机型** | **量程范围** SHS800X | {60mV|600mV|6V|60V|600V|AUTO|MIN|MAX|DEF} SHS1000X | {60mV|600mV|6V|60V|600V|750V|AUTO|MIN|MAX|DEF}
示例	设置使用 60V 的量程测量交流电压值： *CONF:VOLT:AC 60* 查询当前测量值： *READ?* 返回值： *+2.43186951E-02*

CONFigure[:VOLTage]:DC

描述	设置直流电压测量下的所有测量参数和触发参数为默认值，可通过传入参数指定量程。 ● 指定为自动量程时，调整量程根据输入信号方便地为每个测量选择量程。自动调整量程可能需要更多的时间进行量程选择。进行快速测量时，使用固定量程。 ● 如果输入信号值大于指定量程，界面显示“Overload”，并返回“Overload”。 ● 使用 READ?读取测量值。
命令格式	CONFigure[:VOLTage]:DC <range>
参数说明	<range>:= 交流电压测量量程，各机型可设置量程范围请参考下表： **机型** | **量程范围** SHS800X | {60mV|600mV|6V|60V|600V|AUTO|MIN|MAX|DEF} SHS1000X | {60mV|600mV|6V|60V|600V|1000V|AUTO|MIN|MAX|DEF}
示例	设置使用 60V 的量程测量直流电压值：

	CONF:VOLT:DC 60 查询当前测量值： *READ?* 返回值： *+2.43186951E-02*

CONFigure:CAPacitance

描述	设置电容测量下的所有测量参数和触发参数为默认值，可通过传入参数指定量程。 ● 如果输入信号值大于指定量程，界面显示“Overload”，并返回“Overload”。 ● 使用 READ?读取测量值。
命令格式	CONFigure:CAPacitance
示例	设置使用 4 uF 的量程测量电容值： *CONF:CAP* 查询当前测量值： *READ?* 返回值： *+7.26141264E-10*

MEASure

MEASure 命令采用默认的测量参数进行测量，是获取测量值最简便的方法。MEASure 可以用来设置测量模式和档位，但不能更改其他参数的默认值。测量后返回测量值。

注意：一个 MEASure 查询在功能上相当于发送 CONFigure，随后立即进行 READ?。区别在于 CONFigure 命令支持在 CONFigure 和 READ?之间更改参数。

MEASure:CONTinuity

描述	设置通断测量下的所有测量参数和触发参数为其默认值后进行一次测量，并返回测量值。 ● 连通性测量量程固定为 600Ω（二线电阻测量）。 ● 如果蜂鸣器开启，测量值小于或等于量程阈值时，蜂鸣器发声，界面显示当前测量值。 ● READ? 和 MEASure:DIODe? 也可查询电阻值。如果超出量程 600Ω，界面显示“Overload”，并返回“Overload”。
命令格式	MEASure:CONTinuity?
返回格式	<value>
示例	进行连通性测量并查询测量值： *MEAS:CONT?* 返回值： *+9.84739065E+02*

MEASure:CURRent:AC

描述	设置交流电流测量下的所有测量参数和触发参数为默认值后进行一次测量，并返回测量值。此外，还可通过传入参数指定量程。 ● 指定为自动量程时，调整量程根据输入信号方便地为每个测量选择量程。自动调整量程可能需要更多的时间进行量程选择。进行快速测量时，使用固定量程。 ● 如果输入信号值大于指定量程，界面显示“Overload”，并返回“Overload”。
命令格式	MEASure:CURRent:AC? <range>
参数说明	<range>:= {60mA|600mA|6A|10A|AUTO} ● 60mA|600mA|6A|10A：设置固定量程为 60mA|600mA|6A|10A ● AUTO：自动量程

返回格式	<value>
示例	设置使用 6A 的量程测量交流电流值，并查询当前交流电流值： *MEAS:CURR:AC? 6* 返回值： *+4.32133675E-04*

MEASure:CURRent:DC

描述	设置交流电流测量下的所有测量参数和触发参数为默认值后进行一次测量，并返回测量值。此外，还可通过传入参数指定量程。
命令格式	MEASure:CURRent:DC? <range>
参数说明	<range>:= {60mA|600mA|6A|10A|AUTO} ● 60mA|600mA|6A|10A：设置固定量程为 60mA|600mA|6A|10A ● AUTO：自动量程
返回格式	<value>
示例	设置使用 6A 的量程测量直流电流值，并查询直流电流值测量值： *MEAS:CURR:AC? 6* 返回值： *+4.32133675E-04*

MEASure:DIODe

描述	设置二极管测量下的所有测量参数和触发参数为默认值后进行一次测量，并返回测量值。 ● 二极管测量模式量程固定为 2 VDC。 ● READ? 和 MEASure:DIODe? 查询返回测得的电压。如果超出量程 2V，界面显示“Overload”，并返回“Overload”。
命令格式	MEASure:DIODe?
返回格式	<value>
示例	设置并查询二极管测量值： *MEAS:DIOD?* 返回值： *+9.84733701E-01*

MEASure:RESistance

描述	设置二线电阻测量下的所有测量参数和触发参数为默认值后进行一次测量，并返回测量值。此外，还可通过传入参数指定量程。 ● 指定为自动量程时，调整量程根据输入信号方便地为每个测量选择量程。自动调整量程可能需要更多的时间进行量程选择。进行快速测量时，使用固定量程。 ● 如果输入信号值超出指定量程，界面显示“Overload”，并返回“Overload”。
命令格式	MEASure:RESistance? <range>
参数说明	<range>:= {600|6k|60k|600k|6M|60M} ● 600|6k|60k|600k|6M|60M：设置固定量程为 600|6k|60k|600k|6M|60M ● 量程默认为自动量程
返回格式	<value>
示例	设置使用 600Ω 的量程测量二线电阻值，并查询测量值： *MEAS:RES? 600* 返回值： *+6.71881065E+01*

MEASure[:VOLTage]:AC

描述	设置交流电压测量下的所有测量参数和触发参数为默认值后进行一次测量，并返回测量值。此外，还可通过传入参数指定量程。 ● 指定为自动量程时，调整量程根据输入信号方便地为每个测量选择量程。自动调整量程可能需要更多的时间进行量程选择。进行快速测量时，使用固定量程。 ● 如果输入信号值超出指定量程，界面显示“Overload”，并返回“Overload”。
命令格式	MEASure:VOLTage:AC? <range>
参数说明	<range>:= 交流电压测量量程，各机型可设置量程范围请参考下表： **机型** | **量程范围** SHS800X | {60mV|600mV|6V|60V|600V } SHS1000X | {60mV|600 mV|6V|60V|600V|1000V(DC)/750V(AC)}
返回格式	<value>
示例	使用 600V 的量程测量交流电压值，并查询测量值： *MEAS:VOLT:AC? 600* 返回值： *+2.43186951E-02*

MEASure[:VOLTage]:DC

描述	设置直流电压测量下的所有测量参数和触发参数为默认值后进行一次测量，并返回测量值。此外，还可通过传入参数指定量程。 ● 指定为自动量程时，调整量程根据输入信号方便地为每个测量选择量程。自动调整量程可能需要更多的时间进行量程选择。进行快速测量时，使用固定量程。 ● 如果输入信号值超出指定量程，界面显示“Overload”，并返回“Overload”。
命令格式	MEASure:VOLTage:DC? <range>
参数说明	<range>:= 直流电压测量量程，各机型可设置量程范围请参考下表： **机型** | **量程范围** SHS800X | {60mV|600mV|6V|60V|600V } SHS1000X | {60mV|600 mV|6V|60V|600V|1000V(DC)/750V(AC)}
返回格式	<value>
示例	使用 600V 的量程测量交流电压值，并查询测量值： *MEAS:VOLT:AC? 600* 返回值： *+2.43186951E-02*

MEASure:CAPacitance

描述	设置电容测量下测量参数和触发参数为默认值，进行返回测量值。如果输入信号值超出指定量程，界面显示“Overload”，并返回“Overload”。
命令格式	MEASure:CAPacitance?
返回格式	<value>
示例	设置并查询电容测量值： *MEAS:CAP?* 返回值： *+7.26141264E-10*

SENSe

[SENSe:]CURRent:AC:NULL

描述	设置交流电流测量下相对值功能的开关状态。交流电流和直流电流相对值功能互相独立，需要分别进行设置。 出厂重置或 CONFigure 功能之后将关闭相对值功能。
命令格式	[SENSe:]CURRent:AC:NULL <state>
参数说明	<state>:= {ON|OFF}
示例	设置交流电流测量： *CONF:CURR:AC* 设置交流电流测量相对值功能开启： *CURR:AC:NULL ON* 查询测量值： *READ?* 返回值： *MM_VALUE 0.00V*

[SENSe:]CURRent:DC:NULL

描述	设置直流电流测量下相对值功能的开关状态。交流电流和直流电流相对值功能互相独立，需要分别进行设置。 出厂重置或 CONFigure 功能之后仪器禁用相对值功能。
命令格式	[SENSe:]CURRent:DC:NULL <state>
参数说明	<state>:= {ON|OFF}
示例	设置直流电流测量： *CONF:CURR:DC* 设置直流电流测量相对值功能开启： *CURR:DC:NULL ON* 查询测量值： *READ?* 返回值： *MM_VALUE 0.00V*

[SENSe:]CURRent:AC:SELEct

描述	设置交流电流测量的单位为 mA 或 A。交流电流和直流电流单位互相独立，需要分别进行设置。
命令格式	[SENSe:]CURRent:AC:SELEct <unit>
参数说明	<unit>:= {MA|A} ● MA：设置交流电流测量的单位为 mA ● A：设置交流电流测量的单位为 A
示例	设置交流电流测量： *CONF:CURR:AC* 设置交流电流测量的单位为 mA： *CURR:AC:SELE MA*

[SENSe:]CURRent:DC:SELEct

描述	设置直流电流测量的单位为 mA 或 A。交流电流和直流电流单位互相独立，需要分别进行设置。
命令格式	[SENSe:]CURRent:DC:SELEct <unit>
参数说明	<unit>:= {MA|A} ● MA：设置直流电流测量的单位为 mA ● A：设置直流电流测量的单位为 A
示例	设置交流电流测量： *CONF:CURR:DC* 设置交流电流测量的单位为 mA： *CURR:DC:SELE MA*

[SENSe:]RESistance:NULL

描述	设置电阻测量下相对值功能的开关状态。 出厂重置或 CONFigure 功能之后仪器禁用相对值功能。
命令格式	[SENSe:]RESistance:NULL <state>
参数说明	<state>:= {ON|OFF}
示例	设置二线电阻测量，测量 1.5KΩ 的测量电阻： *CONF:RES* 设置电阻测量相对值功能开启：

	RES:NULL ON 查询测量值： *READ?* 返回值： *0*

[SENSe:]VOLTage:AC:NULL

描述	设置交流电压测量下相对值功能的开关状态。交流电压和直流电压相对值功能互相独立，需要分别进行设置。 出厂重置或 CONFigure 功能之后仪器禁用相对值功能。
命令格式	[SENSe:]VOLTage:AC:NULL <state>
参数说明	<state>:= {ON|OFF}
示例	设置交流电压测量，测量 1.5V 的交流电压： *CONF:VOLT:AC* 设置交流电压测量相对值功能开启： *VOLT:AC:NULL ON* 查询测量值： *READ?* 返回值： *MM_VALUE 00.04V*

[SENSe:]VOLTage:DC:NULL

描述	设置直流电压测量下相对值功能的开关状态。交流电压和直流电压相对值功能互相独立，需要分别进行设置。 出厂重置或 CONFigure 功能之后仪器禁用相对值功能。
命令格式	[SENSe:]VOLTage:DC:NULL <state>
参数说明	<state>:= {ON|OFF}
示例	设置直流电压测量，测量 1.5V 的直流电压： *CONF:VOLT:AC* 设置直流电压测量相对值功能开启： *VOLT:DC:NULL ON* 查询测量值： *READ?*

	返回值： *MM_VALUE 00.04V*

[SENSe:]VOLTage:AC:SELEct

描述	设置交流电压测量的单位为 mV 或 V。
命令格式	[SENSe:]VOLTage:AC:SELEct <unit>
参数说明	<unit>:= {MV|V} ● MV：设置交流电压测量的单位为 mV ● V：设置交流电压测量的单位为 V
示例	设置交流电压测量： *CONF:VOLT:AC* 设置交流电压测量的单位为 V： *VOLT:AC:SELE V*

[SENSe:]VOLTage:DC:SELEct

描述	设置直流电压测量的单位为 mV 或 V。
命令格式	[SENSe:]VOLTage:DC:SELEct <unit>
参数说明	<unit>:= {MV|V} ● MV：设置直流电压测量的单位为 mV ● V：设置直流电压测量的单位为 V
示例	设置直流电压测量： *CONF:VOLT:DC* 设置直流电压测量的单位为 V： *VOLT:DC:SELE V*

[SENSe:]CAPacitance:NULL

描述	设置电容测量下相对值功能的开关状态。
命令格式	[SENSe:]CAPacitance:NULL <state>
参数说明	<state>:= {ON|OFF}
示例	设置电容测量： *CONF:CAP*

设置电容测量相对值功能开启：

CAP:NULL ON

查询测量值：

READ?

返回值：

MM_VALUE 0.00nF

5.20 MTESt 命令系统

:MTESt 命令子系统控制模板测试功能，涉及模板创建与保存、测试类型的选取等。

:MTESt

描述	设置或查询模板测试开关状态。
命令格式	:MTESt <state> :MTESt?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启模板测试： *:MTESt ON* *MTES ON* 查询模板测试开关状态： *MTES?* 返回值： *ON*

:MTESt:COUNt

描述	查询模板测试的结果。
命令格式	:MTESt:COUNt?
返回格式	FAIL,<num>,PASS,<num>,TOTAL,<num> <num>:= 测试数据值，整型 NR1 格式
示例	查询模板测试的结果： *MTES:COUN?* 返回值： *FAIL,38176,PASS,5617,TOTAL,43793*
关联命令	:MTESt:OPERate

:MTESt:FUNCtion:BUZZer

描述	设置或查询模板测试蜂鸣器开关状态。开启蜂鸣器时，检测到故障帧发声提示。

命令格式	:MTESt:FUNCtion:BUZZer <state> :MTESt:FUNCtion:BUZZer?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启模板测试的蜂鸣器： *:MTESt:FUNCtion:BUZZer ON* *MTES:FUNC:BUZZ ON* 查询蜂鸣器的开关状态： *MTES:FUNC:BUZZ?* 返回值： *ON*

:MTESt:FUNCtion:COF

描述	设置或查询模板测试失败截图的开关状态。开启失败截图时，当检测到故障帧则会截图保存，保存默认路径为“SIGLENT/”。
命令格式	:MTESt:FUNCtion:COF <state> :MTESt:FUNCtion:COF?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启模板测试的失败截图： *:MTESt:FUNCtion:COF ON* *MTES:FUNC:COF ON* 查询模板测试失败截图的开关状态： *MTES:FUNC:COF?* 返回值： *ON*

:MTESt:FUNCtion:FTH

描述	设置或查询模板测试中存储失败帧的开关状态。开启存储失败帧时，当检测到故障帧则会存储到内存中，可在历史波形中查看失败帧。
命令格式	:MTESt:FUNCtion:FTH <state> :MTESt:FUNCtion:FTH?
参数说明	<state>:= {ON|OFF}

返回格式	{ON|OFF}
示例	设置开启模板测试的存储失败帧： *:MTESt:FUNCtion:FTH ON* *MTES:FUNC:FTH ON* 查询模板测试中存储失败帧的开关状态： *MTES:FUNC:FTH?* 返回值： *ON*
关联命令	:MTESt:OPERate

:MTESt:FUNCtion:SOF

描述	设置或查询模板测试中失败即停的开关状态。开启失败即停时，当检测到故障帧则停止采集。
命令格式	:MTESt:FUNCtion:SOF <state> :MTESt:FUNCtion:SOF?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启模板测试的失败即停： *:MTESt:FUNCtion:SOF ON* *MTES:FUNC:SOF ON* 查询模板测试中失败即停的开关状态： *MTES:FUNC:SOF?* 返回值： *ON*

:MTESt:IDISplay

描述	设置或查询模板测试中显示信息的开关状态。
命令格式	:MTESt:IDISplay <state> :MTESt:IDISplay?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启模板测试的显示信息： *:MTESt:IDISplay ON*

	MTES:IDIS ON 查询模板测试中显示信息的开关状态： *MTES:IDIS?* 返回值： *ON*
关联命令	:MTESt:COUNt

:MTESt:MASK:CREate

描述	根据指定的水平 X 和垂直 Y 数值，以波形为中心创建模板。
命令格式	:MTESt:MASK:CREate <XMARgin>,<YMARgin>
参数说明	<XMARgin>:= 水平 X 的数值，浮点型 NR2 格式 <YMARgin>:= 垂直 Y 的数值，浮点型 NR2 格式
示例	按照水平 X = 0.8，垂直 Y = 0.08 创建规则模板： *:MTESt:MASK:CREate 0.8,0.08* *MTES:MASK:CRE 0.8,0.08*

:MTESt:MASK:LOAD

描述	模板测试加载内部或外部存储的模板。
命令格式	:MTESt:MASK:LOAD <location>
参数说明	<location>:= {INTernal,<num>|EXTernal,<path>} ● INTernal:= 早期版本支持以索引号保存模板到内部，将参数指定为 INTernal，可以加载早期保存的模板。 <num>:= {1|2|3|4} ● EXTernal:=通过指定路径<path>加载模板。 <path>:= 带引号的路径名字符串，扩展名为“.msk”或“.smsk”；用户可以根据需要从内部、网络存储器或 U 盘调用模板

存储路径类型	示例
内部	“local/SIGLENT/test.msk”
网络存储器	“net_storage/SIGLENT/test.smsk”
U 盘	“U-disk0/SIGLENT/test.msk” “U-disk0/SIGLENT/test.smsk”

注意：文件格式不是由文件扩展名自动确定，您需要选择一个扩展名与所选文

	件格式一致的文件。
示例	设置加载内部文件编号为 1 的模板： *:MTESt:MASK:LOAD INTernal,1* *MTES:MASK:LOAD INT,1* *:MTES:MASK:LOAD EXTernal,"local/SIGLENT/SDS0001.msk"* *MTES:MASK:LOAD EXT,"local/SIGLENT/SDS0001.msk"* 设置加载外部 U-disk0 中 SIGLENT 目录下文件名为“TEST.msk”的模板： *:MTESt:MASK:LOAD EXTernal,"SIGLENT/TEST.msk"* *MTES:MASK:LOAD EXT,"SIGLENT/TEST.msk"*

:MTESt:OPERate

描述	设置或查询模板测试中操作开关状态。
命令格式	:MTESt:OPERate <state> :MTESt:OPERate?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启模板测试的操作： *:MTESt:OPERate ON* *MTES:OPER ON* 查询模板测试中操作的开关状态： *MTES:OPER?* 返回值： *ON*

:MTESt:RESet

描述	重置模板测试。
命令格式	:MTESt:RESet
示例	重置模板测试： *:MTESt:RESet* *MTES:RES*
关联命令	:MTESt:OPERate

:MTESt:SOURce

描述	设置或查询模板测试的源。
命令格式	:MTESt:SOURce <source> :MTESt:SOURce?
参数说明	<source>:= {C<n>|Z<n>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● Z<n>:= 缩放波形，<n>:= 序号，整型 NR1 格式 **注意**：Zoom 开启时，Z<n>作为源才有效。
返回格式	{C<n>|Z<n>}
示例	设置模板测试源为 C1： *:MTESt:SOURce C1* *MTES:SOUR C1* 查询模板测试源： *MTES:SOUR?* 返回值： *C1*

:MTESt:TYPE

描述	设置或查询模板测试类型。
命令格式	:MTESt:TYPE <type> :MTESt:TYPE?
参数说明	<type>:= {ALL_IN|ALL_OUT|ANY_IN|ANY_OUT} ● ALL_IN：所有波形点均处于模板内时判断为通过，只要有 1 个波形点不在模板区域即判断为失败 ● ALL_OUT：所有波形点均处于模板外时判断为通过，只要有 1 个波形点进入模板区域即判断为失败 ● ANY_IN：只要有 1 个波形点处于模板内则判断为通过，否则判断为失败 ● ANY_OUT：只要有 1 个波形点处于模板外则判断为通过，否则判断为失败
返回格式	{ALL_IN|ALL_OUT|ANY_IN|ANY_OUT}
示例	设置模板测类型为 ALL_IN： *:MTESt:TYPE ALL_IN* *MTES:TYPE ALL_IN* 查询模板测类型： *MTES:TYPE?* 返回值： *ALL_IN*

5.21 RECall 命令系统

:RECall 命令子系统控制示波器的设置和波形数据的调用。

:RECall:FDEFault

描述	调用出厂设置。
命令格式	:RECall:FDEFault
示例	调用出厂设置： *:RECall:FDEFault* *REC:FDEF*
关联命令	:RECall:SETup

:RECall:PROJect

描述	从内部或外部存储器中调出工程文件。
命令格式	:RECall:PROJect <path>
参数说明	<path>:= 带引号的 ASCII 文本字符串，。可指定调出路径类型如下表： **存储路径类型** / **示例** 内部 / “local/SIGLENT/Project/Project.spro” 网络存储器 / “net_storage/SIGLENT/Project.spro” U 盘 / “U-disk0/SIGLENT/Project.spro” **注意**：如果未指定存储路径类型，将默认从 U-disk0 中调出配置文件。
示例	调出 U 盘 SIGLENT 文件夹下工程目录为 Project 的工程，仅需加载 Project.spro 文件： *:RECall:PROJect "U-disk0/SIGLENT/Project/Project.spro"* *REC:PROJ "U-disk0/SIGLENT/Project/Project.spro"*
关联命令	:SAVE:PROJect

:RECall:REFerence

描述	从内部或外部存储器中调用参考波形文件，并复制到指定的参考波形中。
命令格式	:RECall:REFerence <location>,<path>
参数说明	<location>:= {REF<r>} ● REF<r>:= 参考波形，<r>:= {A|B|C|D} <path>:= 带引号的 ASCII 文本字符串，包含文件存储路径和带后缀“.ref”的文件名。可指定调出路径类型如下表： **存储路径类型** / **示例** 内部 / “local/SIGLENT/test.ref” 网络存储器 / “net_storage/SIGLENT/test.ref” U 盘 / “U-disk0/SIGLENT/test.ref” “U-disk1/SIGLENT/test.ref” **注意**： ● 文件格式不是由文件扩展名自动确定，您需要选择一个扩展名与所选文件格式一致的文件。 ● 如果未指定存储路径类型，将默认从 U-disk0 中调出配置文件。
示例	从外部 U-disk0 中调用 SIGLENT 目录下的波形文件 math.ref，并应用到 REFD 中： *:RECall:REFerence REFD,"U-disk0/SIGLENT/math.ref"* *REC:REF REFD,"U-disk0/SIGLENT/math.ref"*
关联命令	:SAVE:REFerence

:RECall:SERase

描述	删除用户存储在示波器内部的文件，包括参考波形文件、设置文件、模板文件、自定义默认设置文件、复制通道波形到 AWG 时生成的波形文件。
命令格式	:RECall:SERase
示例	删除用户存储在示波器内部的文件： *:RECall:SERase* *REC:SER*

:RECall:SETup

<table>
<tr><td>描述</td><td>从内部或外部存储器调出配置文件。</td></tr>
<tr><td>命令格式</td><td>:RECall:SETup <state></td></tr>
<tr><td>参数说明</td><td><state>:= {INTernal,<num>|EXTernal,<path>}
<num>:= 整型 NR1 格式
<path>:= 带引号的 ASCII 文本字符串，包含文件存储路径和带后缀“.xml”的文件名。可指定调出路径类型如下表：
<table>
<tr><th>存储路径类型</th><th>示例</th></tr>
<tr><td>内部</td><td>“local/SIGLENT/test.xml”</td></tr>
<tr><td>网络存储器</td><td>“net_storage/SIGLENT/test.xml”</td></tr>
<tr><td>U 盘</td><td>“U-disk0/SIGLENT/test.xml”
“U-disk1/SIGLENT/test.xml”</td></tr>
</table>
注意：
● 文件格式不是由文件扩展名自动确定，您需要选择一个扩展名与所选文件格式一致的文件。
● 如果未指定存储路径类型，将默认从 U-disk0 中调出配置文件。</td></tr>
<tr><td>示例</td><td>从内部文件“SDS00001.xml”中调出配置：
:RECall:SETup INTernal,1
REC:SET INT,1
从外部文件“SIGLENT/default.xml”中调出配置：
:RECall:SETup EXTernal,"U-disk0/SIGLENT/default.xml"
REC:SET EXT,"SIGLENT/default.xml"</td></tr>
<tr><td>关联命令</td><td>:RECall:FDEFault
:SAVE:SETup</td></tr>
</table>

5.22 REF 命令系统

:REF 命令子系统控制参考波形。

:REF<r>:LABel

描述	设置或查询指定参考波形标签开关状态。
命令格式	:REF<r>:LABel <state> :REFA:LAB?
参数说明	<r>:= {A|B|C|D} <state>:= {ON|OFF}
返回格式	{ON/OFF}
示例	设置参考波形 REFA 标签开启： *:REFA:LABel ON* *REFA:LAB ON* 查询当前参考波形 REFA 标签状态： *REFA:LAB?* 返回值： *ON*
关联命令	:REF<r>:LABel:TEXT

:REF<r>:LABel:TEXT

描述	设置或查询指定参考波形的标签字符。
命令格式	:REF<r>:LABel:TEXT <qstring> :REFA:LAB?
参数说明	<r>:= {A|B|C|D} <qstring>:= 带引号的 ASCII 文本字符串，字符串的长度限制为 20
返回格式	<qstring>
示例	设置 REFA 的标签为 REFA： *:REFA:LABel:TEXT "REFA"* *REFA:LAB:TEXT "REFA"* 查询 REFA 标签： *REFA:LAB:TEXT?* 返回值：

	"REFA"
关联命令	:REF<r>:LABel

:REF<r>:DATA

描述	设置参考波形的调用及保存。
命令格式	:REF<r>:DATA <operation>
参数说明	<r>:= {A|B|C|D} <operation>:= {LOAD|UNLoad|SAVE,<source>} ● LOAD：REF 波形文件存在时，开启 REF 波形显示 ● UNLoad：关闭 REF 波形显示 ● SAVE：保存指定源到参考波形中 <source>:= {C<n>|F<x>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
示例	REFA 波形文件存在时，开启显示： *:REFA:DATA LOAD* *REFA:DATA LOAD*

:REF<r>:DATA:SOURce

描述	查询当前参考波形的信源。
命令格式	:REF<r>:DATA: SOURce?
参数说明	<r>:= {A|B|C|D}
返回格式	<source>:= {C<n>|F<x>|D<d>} ● <n>:= 模拟通道序号，整型 NR1 格式 ● <x>:= 数学函数序号，整型 NR1 格式 ● <d>:= 数字通道序号，整型 NR1 格式
示例	查询返回 REFA 的信源： *:REFA:DATA:SOUR?* 返回值： *C1*

:REF<r>:DATA:SCALe

描述	设置或查询参考波形的垂直档位。此命令仅参考波形已存储且打开时可用。
命令格式	:REF<r>:DATA:SCALe <value> :REF<r>:DATA:SCALe?
参数说明	<r>:= {A|B|C|D} <value>:= 数值，浮点型 NR3 格式。如果设定值大于可设定值，它将自动匹配可设定值，设置范围与参考源相同。
返回格式	浮点型 NR3 数据
示例	设置 REFA 的垂直档位为 0.1V： *:REFA:DATA:SCALe 1.00E-01* *REFA:DATA:SCAL 1.00E-01* 查询 REFA 垂直档位： *REFA:DATA:SCAL?* 返回值： *1.00E-01*
关联命令	:REF<r>:DATA:POSition

:REF<r>:DATA:POSition

描述	设置或查询参考通道的垂直偏移。此命令仅参考波形已存储且打开时可用。
命令格式	:REF<r>:DATA:POSition <value> :REF<r>:DATA:POSition?
参数说明	<r>:= {A|B|C|D} <value>:= 数值，浮点型 NR3 格式。如果设定值大于可设定值，它将自动匹配可设定值，设置范围与参考源相同。
返回格式	浮点型 NR3 格式
示例	设置 REFA 的垂直偏移为 0.2V： *:REFA:DATA:POSition 2.00E-01* *REFA:DATA:POS 2.00E-01"* 查询 REFA 的垂直偏移： *REFA:DATA:POS?* 返回值： *2.00E-01*
关联命令	:REF<r>:DATA:SCALe

5.23 SAVE 命令系统

:SAVE 命令子系统控制示波器进行存储设置、波形数据、图片等到内部或外部存储器。

:SAVE:BINary

描述	将屏幕显示的指定通道波形以 bin 格式文件保存到内部或外部存储器中。
命令格式	:SAVE:BINary <path>,<source>
参数说明	<path>:= 带引号的 ASCII 文本字符串，包含文件存储路径和带后缀“.bin”的文件名。可指定存储路径类型如下表： **存储路径类型** | **示例** 内部 | “local/SIGLENT/test.bin” 网络存储器 | “net_storage/test.bin” U 盘 | “U-disk0/test.bin” <source>:= {C<n>|Z<n>|F<x>|M<m>|D0_D15|DIGital|ZD0_ZD15|ZDIGital|ALL} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● Z<n>:= 缩放波形，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式 ● D0_D15：数字通道，按位存储数据 ● DIGital：数字通道，按总线存储数据 ● ZD0_ZD15：数字通道缩放波形，按位存储数据 ● ZDIGital：数字通道缩放波形，按总线存储数据 ● ALL：保存所有开启通道。当多通道保存成单文件，文件名称为<path>；当各通道分开保存时，文件名以<path>路径为前缀，自动补全通道名。 **注意**： ● 保存到内部时，存储路径必须包含 local。 ● 保存到外部时，如未指定存储路径类型，将默认存储到 U-disk0。 ● 文件格式不是由文件扩展名自动确定，您需要选择一个扩展名与所选文件格式一致的文件。
示例	将 C1 波形存储至 U-disk0 文件夹 SIGLENT 中，文件名为 channel1.bin： *:SAVE:BINary "U-disk0/SIGLENT/channel1.bin",C1* *SAVE:BIN "U-disk0/Siglent/c1.bin",C1*
关联命令	:SAVE:SFILe

:SAVE:CSV

描述	将屏幕显示的指定通道波形以 csv 格式文件保存到内部或外部存储器中。
命令格式	:SAVE:CSV <path>,<source>,<state>
参数说明	<path>:= 带引号的 ASCII 文本字符串，包含文件存储路径和带后“.csv”缀的文件名。可指定存储路径类型如下表： **存储路径类型** | **示例** 内部 | “local/SIGLENT/test.csv” 网络存储器 | “net_storage/test.csv” U 盘 | “U-disk0/test.csv” <source>:= {C<n>|Z<n>|F<x>|M<m>|D0_D15|DIGital|ZD0_ZD15|ZDIGital|ALL} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● Z<n>:= 缩放波形，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式 ● D0_D15：数字通道，按位存储数据 ● DIGital：数字通道，按总线存储数据 ● ZD0_ZD15：数字通道缩放波形，按位存储数据 ● ZDIGital：数字通道缩放波形，按总线存储数据 ● ALL：保存所有开启通道。当多通道保存成单文件，文件名称为<path>；当各通道分开保存时，文件名以<path>路径为前缀，自动补全通道名。 <state>:= {OFF|ON} ● ON：开启参数保存，文件中将包含垂直、水平等参数设置信息 ● OFF：关闭参数保存 **注意**: ● 保存到内部时，存储路径必须包含 local。 ● 保存到外部时，如未指定存储路径类型，将默认存储到 U-disk0。 ● 文件格式不是由文件扩展名自动确定，您需要选择一个扩展名与所选文件格式一致的文件。
示例	将 C1 波形存储至内部文件夹 SIGLENT 中，文件名为 channel1.csv，且开启参数保存： *:SAVE:CSV "local/SIGLENT/channel1.csv",C1,ON*
关联命令	:SAVE:SFILe

:SAVE:DEFault

描述	保存用户自定义设置或者出厂设置作为默认设置。
命令格式	:SAVE:DEFault <set>
参数说明	<set>:= {CUSTom|FACTory} ● CUSTom：用户自定义的当前设置。 ● FACTory：出厂设置
示例	将用户自定义的当前设置保存为默认设置： *:SAVE:DEFault CUSTom* *SAVE:DEF CUST*
关联命令	:RECall:SETup

:SAVE:IMAGe

描述	将屏幕截图以 bmp、jpg 或 png 格式文件保存到内部或外部存储器中。
命令格式	:SAVE:IMAGe <path>,<type>,<invert>[,<menu>]
参数说明	<path>:= 带引号的 ASCII 文本字符串，包含文件存储路径和带后缀“.bmp”、“.jpg”或“.png”的文件名。可指定存储路径类型如下表： **存储路径类型** | **示例** 内部 | “local/SIGLENT/test.bmp” 网络存储器 | “net_storage/test.jpg” U 盘 | “U-disk0/test.png” <type>:= {BMP|JPG|PNG} <invert>:= {OFF|ON} ● ON：开启反色功能，图片底色色调为浅色，适合打印。 ● OFF：保存与仪器显示相同的图片。 <menu>:= {MOFf|MON} ● MON：截图包含菜单，存储后图片带有菜单信息。 ● MOFf：截图不包含菜单，存储后图片隐藏菜单信息：顶部菜单、右侧菜单以及右下角时间信息。 **注意**： ● 保存到内部时，存储路径必须包含 local。 ● 保存到外部时，如未指定存储路径类型，将默认存储到 U-disk0。 ● 文件格式不是由文件扩展名自动确定，您需要选择一个扩展名与所选文

	件格式一致的文件。
示例	将截图存储至外部存储器 U-disk0 的 SIGLENT 文件夹中，文件名为 screen.bmp，且开启反色功能和隐藏菜单： *:SAVE:IMAGe "U-disk0/SIGLENT/screen.bmp",BMP,ON,MOFf* *SAVE:IMAG "U-disk0/SIGLENT/screen.bmp",BMP,ON,MOF*
关联命令	:PRINt

:SAVE:PROJect

描述	将当前波形数据和使用的所有配置保存成工程文件到内部或外部存储器中。
命令格式	:SAVE:PROJect <path>
参数说明	<path>:= 带引号的 ASCII 文本字符串。可指定存储路径类型如下表： **存储路径类型** | **示例** 内部 | "local/SIGLENT/Project" 网络存储器 | "net_storage/SIGLENT/Project" U 盘 | "U-disk0/SIGLENT/Project" **注意**： ● 保存到内部时，存储路径必须包含 local。 ● 保存到外部时，如未指定存储路径类型，将默认存储到 U-disk0。
示例	将当前波形数据以及配置保存至内部 SIGLENT 文件夹下的 Project 文件夹中： *:SAVE:PROJect "local/SIGLENT/Project"* *SAVE:PROJ "local/SIGLENT/Project"*

:SAVE:MATLab

描述	将屏幕显示的指定通道波形以 mat 格式文件保存到内部或外部存储器中。
命令格式	:SAVE:MATLab <path>,<source>
参数说明	<path>:= 带引号的 ASCII 文本字符串，包含文件存储路径和带后缀".mat"的文件名。可指定存储路径类型如下表： **存储路径类型** | **示例** 内部 | "local/SIGLENT/test.mat" 网络存储器 | "net_storage/test.mat" U 盘 | "U-disk0/test.mat"

	<source>:= {C<n>|Z<n>|F<x>|M<m>|D0_D15|DIGital|ZD0_ZD15|ZDIGital|ALL} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● Z<n>:= 缩放波形，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式 ● D0_D15：数字通道，按位存储数据 ● DIGital：数字通道，按总线存储数据 ● ZD0_ZD15：数字通道缩放波形，按位存储数据 ● ZDIGital：数字通道缩放波形，按总线存储数据 ● ALL：保存所有开启通道。当多通道保存成单文件，文件名称为<path>；当各通道分开保存时，文件名以<path>路径为前缀，自动补全通道名。 **注意**： ● 保存到内部时，存储路径必须包含 local。 ● 保存到外部时，如未指定存储路径类型，将默认存储到 U-disk0。 ● 文件格式不是由文件扩展名自动确定，您需要选择一个扩展名与所选文件格式一致的文件。
示例	将 C1 波形存储至外部存储器 U-disk0 的 SIGLENT 文件夹中，文件名为 channel.mat： *:SAVE:MATLab "U-disk0/SIGLENT/channel.mat",C1* *SAVE:MATL "U-disk0/SIGLENT/channel.mat",C1*
关联命令	:SAVE:SFILe

:SAVE:REFerence

描述	将指定通道波形以 ref 格式文件保存到内部或外部存储器作参考波形。
命令格式	:SAVE:REFerence <path>,<source>
参数说明	<path>:= 带引号的 ASCII 文本字符串，包含文件存储路径和带后缀“.ref”的文件名。可指定存储路径类型如下表： **存储路径类型** | **示例** 内部 | “local/SIGLENT/test.ref” 网络存储器 | “net_storage/test.ref” U 盘 | “U-disk0/test.ref” <source>:= {C<n>|F<x>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式

	● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式 **注意**: ● 保存到内部时，存储路径必须包含 local。 ● 保存到外部时，如未指定存储路径类型，将默认存储到 U-disk0。 ● 文件格式不是由文件扩展名自动确定，您需要选择一个扩展名与所选文件格式一致的文件。
示例	将 C1 波形数据作为参考波形存储至内部 SIGLENT 文件夹中，文件名为 channel.ref： *:SAVE:REFerence "local/SIGLENT/channel.ref",C1* *SAVE:REF "local/SIGLENT/channel.ref",C1*
关联命令	:RECall:REFerence

:SAVE:SETup

描述	将当前设置保存到外部或内部存储器中。
命令格式	:SAVE:SETup <setup_num>
参数说明	<setup_num>:= {INTernal,<num>|EXTernal,<path>} <num>:= 保存设置序列号，整型 NR1 数据 <path>:= 带引号的 ASCII 文本字符串，包含文件存储路径和带后缀“.xml”文件名。可指定存储路径类型如下表：

存储路径类型	示例
内部	“local/SIGLENT/default.xml”
网络存储器	“net_storage/SIGLENT/default.xml”
U 盘	“U-disk0/SIGLENT/default.xml”

	注意: ● 保存到内部时，存储路径必须包含 local。 ● 保存到外部时，如未指定存储路径类型，将默认存储到 U-disk0。 ● 文件格式不是由文件扩展名自动确定，您需要选择一个扩展名与所选文件格式一致的文件。
示例	有两种方式将当前设置保存到内部文件”SDS00001.xml”中： *:SAVE:SETup INTernal,1* *SAVE:SET INT,1* *:SAVE:SETup EXTernal,"local/SDS00001.xml"* *:SAVE:SET EXT,"local/SDS00001.xml"*

关联命令	:SAVE:DEFault :RECall:SETup

:SAVE:SFILe

描述	保存全部通道数据时，设置或查询多通道是否保存成单文件
命令格式	:SAVE:SFILe <state> :SAVE:SFILe?
参数说明	<state>:= {YES|NO}
返回格式	{YES|NO}
示例	设置多通道保存成单文件功能为开启： *:SAVE:SFILe YES* *SAVE:SFIL YES* 查询当前多通道保存成单文件功能开关状态： *SAVE:SFIL?* 返回值： *YES*
关联命令	:SAVE:BINary :SAVE:CSV :SAVE:MATLab

5.24 SEARch 命令系统

:SEARch 命令子系统控制示波器搜索功能。

:SEARch

描述	设置或查询搜索功能的开关状态。
命令格式	:SEARch <state> :SEARch?
参数说明	<state>:= {ON|OFF}
返回格式	{ON/OFF}
示例	设置搜索功能开启： *:SEARch ON* *SEAR ON* 查询当前搜索功能开关状态： *SEAR?* 返回值： *ON*

:SEARch:MODE

描述	设置或查询当前搜索模式。
命令格式	:SEARch:MODE <mode> :SEARch:MODE?
参数说明	<mode>:= {EDGE|SLOPe|PULSE|INTerval|RUNT} ● EDGE：边沿搜索 ● SLOPe：斜率搜索 ● PULSE：脉宽搜索 ● INTerval：间隔搜索 ● RUNT：欠幅搜索
返回格式	{EDGE|SLOPe|PULSE|INTerval|RUNT}
示例	设置搜索模式为边沿搜索： *:SEARch:MODE EDGE* *SEAR:MODE EDGE* 查询搜索模式：

	SEAR:MODE? 返回值： *EDGE*

:SEARch:COUNT

描述	查询当前屏幕内搜索事件总数。
命令格式	:SEARch:COUNT?
参数说明	<value>:= 数值，整型 NR1 格式
返回格式	整型 NR1 格式
示例	查询当前屏幕内搜索事件总数： *SEARch:COUNt?* *SEAR:COUN?* 返回值： *10*

:SEARch:EVENt

描述	查询示波器采集停止时，定位的事件帧号。
命令格式	:SEARch:EVENt?
参数说明	<value>:= 数值，整型 NR1 格式
返回格式	整型 NR1 格式
示例	查询当前定位事件帧号： *SEARch:EVENt?* *SEAR:EVEN?* 返回值： *5*

:SEARch:COPY

描述	设置搜索设置和触发设置同步。
命令格式	:SEARch:COPY <operation>
参数说明	<operation>:= {FROMtrigger|TOTRigger|CANCel} ● FROMtrigger：从触发复制。将当前触发设置同步到搜索设置中

	● TOTRigger：复制到触发。将当前搜索设置同步到触发设置中 ● CANCel：取消复制。取消上一次同步操作，还原到同步前的配置
示例	设置将当前触发设置同步到搜索设置中： *:SEARch:COPY FROMtrigger* *SEAR:COPY FROM*

:SEARch:EDGE

:SEARch:EDGE:SOURce

描述	设置或查询边沿搜索的信源。
命令格式	:SEARch:EDGE:SOURce <source > :SEARch:EDGE:SOURce?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置边沿搜索的信源为 C1： *:SEARch:EDGE:SOURce C1* *SEAR:EDGE:SOUR C1* 查询边沿搜索的信源： *SEAR:EDGE:SOUR?* 返回值： *C1*
关联命令	:SEARch:EDGE:LEVel

:SEARch:EDGE:SLOPe

描述	设置或查询边沿搜索的斜率。
命令格式	:SEARch:EDGE:SLOPe <slope_type> :SEARch:EDGE:SLOPe?
参数说明	<slope_type>:= {RISing|FALLing|ALTernate} ● RISing：上升沿 ● FALLing：下降沿 ● ALTernate：任意沿
返回格式	{RISing|FALLing|ALTernate}

示例	设置边沿搜索的斜率为上升沿： *:SEARch:EDGE:SLOPe RISing* *SEAR:EDGE:SLOP RIS* 查询边沿搜索的斜率： *SEAR:EDGE:SLOP?* 返回值： *RISing*

:SEARch:EDGE:LEVel

描述	设置或查询边沿搜索的搜索电平。
命令格式	:SEARch:EDGE:LEVel <level_value> :SEARch:EDGE:LEVel?
参数说明	<level_value>:= 数值，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置边沿搜索的搜索电平为 0.5 V： *:SEARch:EDGE:LEVel 5.00E-01* *SEAR:EDGE:LEV 5.00E-01* 查询边沿搜索的搜索电平： *SEAR:EDGE:LEV?* 返回值： *5.00E-01*
关联命令	:SEARch:EDGE:SOURce

:SEARch:SLOPe

:SEARch:SLOPe:SOURce

描述	设置或查询斜率搜索的信源。
命令格式	:SEARch:SLOPe:SOURce <source> :SEARch:SLOPe:SOURce?
参数说明	<source>:= C<n> <n>:= 模拟通道序号，整型 NR1 格式
返回格式	C<n>
示例	设置斜率搜索的信源为 C2： *:SEARch:SLOPe:SOURce C2* *SEAR:SLOP:SOUR C2* 查询斜率搜索的信源： *SEAR:SLOP:SOUR?* 返回值： *C2*

:SEARch:SLOPe:SLOPe

描述	设置或查询斜率搜索的斜率。
命令格式	:SEARch:SLOPe:SLOPe <slope_type> :SEARch:SLOPe:SLOPe?
参数说明	<slope_type>:= {RISing|FALLing} ● RISing：上升沿 ● FALLing：下降沿
返回格式	{RISing|FALLing}
示例	设置斜率搜索的斜率为上升沿： *:SEARch:SLOPe:SLOPe RISing* *SEAR:SLOP:SLOP RIS* 查询斜率搜索的斜率： *SEAR:SLOP:SLOP?* 返回值： *RISing*

:SEARch:SLOPe:HLEVel

描述	设置或查询斜率搜索的高电平。
命令格式	:SEARch:SLOPe:HLEVel <high_level_value> :SEARch:SLOPe:HLEVel?
参数说明	<high_level_value>:= 数值，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **注意**：高电平不能小于低电平（低电平设置指令:SEARch:SLOPe:LLEVel）。
返回格式	浮点型 NR3 格式
示例	设置斜率搜索的高电平为 0.5V： *:SEARch:SLOPe:HLEVel 5.00E-01* *SEAR:SLOP:HLEV 5.00E-01* 查询斜率搜索的高电平： *SEAR:SLOP:HLEV?* 返回值： *5.00E-01*
关联命令	:SEARch:SLOPe:LLEVel

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

:SEARch:SLOPe:LLEVel

描述	设置或查询斜率搜索的低电平。
命令格式	:SEARch:SLOPe:LLEVel <low_level_value> :SEARch:SLOPe:LLEVel?

参数说明	<low_level_value>:= 数值，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： <table><tr><th>机型</th><th>范围</th></tr><tr><td>SDS7000A SDS5000X HD</td><td>[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]</td></tr><tr><td>SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X</td><td>[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]</td></tr><tr><td>SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD</td><td>[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]</td></tr></table> **注意**：低电平不能大于高电平（高电平设置指令:SEARch:SLOPe:HLEVel）。
返回格式	浮点型 NR3 格式
示例	设置斜率搜索的低电平为 -0.5V： *:SEARch:SLOPe:LLEVel -5.00E-01* *SEAR:SLOP:LLEV -5.00E-01* 查询斜率搜索的低电平： *SEAR:SLOP:LLEV?* 返回值： *-5.00E-01*
关联命令	:SEARch:SLOPe:HLEVel

:SEARch:SLOPe:LIMit

描述	设置或查询斜率搜索限制条件类型。
命令格式	:SEARch:SLOPe:LIMit <type> :SEARch:SLOPe:LIMit?
参数说明	<type>:= {LESSthan|GREATerthan|INNer|OUTer} ● LESSthan（<=）：小于时间值。 ● GREATerthan（>=）：大于时间值。 ● INNer（[--,--]）：时间值范围内。 ● OUTer（--][--）：时间值范围外。

返回格式	{LESSthan|GREATerthan|INNer|OUTer}
示例	设置斜率搜索的限制条件类型为小于时间值： *:SEARch:SLOPe:LIMit LESSthan* *SEAR:SLOP:LIM LESS* 查询斜率搜索的限制条件类型： *SEAR:SLOP:LIM?* 返回值： *LESSthan*
关联命令	:SEARch:SLOPe:TUPPer :SEARch:SLOPe:TLOWer

:SEARch:SLOPe:TUPPer

描述	设置或查询斜率搜索限制条件的上限值。
命令格式	:SEARch:SLOPe:TUPPer <value> :SEARch:SLOPe:TUPPer?
参数说明	<value>:= 数值，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： <table><tr><th>机型</th><th>范围</th></tr><tr><td>SDS7000A</td><td>[1.00E-09,2.00E+01]</td></tr><tr><td>SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD</td><td>[-2.00E-09,2.00E+01]</td></tr><tr><td>SHS800X SHS1000X</td><td>[2.00E-09,4.20E+00]</td></tr></table> **注意**： ● 上限值不能小于下限值（下限值设置指令:SEARch:SLOPe:TLOWer）。 ● 当限制类型为“>=”时，此命令不可用。
返回格式	浮点型 NR3 格式
示例	设置斜率搜索限制条件的上限值为 30ns： *:SEARch:SLOPe:TUPPer 3.00E-08*

	SEAR:SLOP:TUPP 3.00E-08 查询斜率搜索限制条件的上限值： *SEAR:SLOP:TUPP?* 返回值： *3.00E-08*
关联命令	:SEARch:SLOPe:LIMit :SEARch:SLOPe:TLOWer

:SEARch:SLOPe:TLOWer

描述	设置或查询斜率搜索限制条件的下限值。
命令格式	:SEARch:SLOPe:TLOWer <value> :SEARch:SLOPe:TLOWer?
参数说明	<value>:= 数值，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A	[1.00E-09,2.00E+01]
SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-2.00E-09,2.00E+01]
SHS800X SHS1000X	[2.00E-09,4.20E+00]

注意：

- 下限值不能大于上限值（上限值设置指令:SEARch:SLOPe:TUPPer）。
- 当限制类型为“<=”时，此命令不可用。

返回格式	浮点型 NR3 格式
示例	设置斜率搜索限制条件的下限值为 10ns： *:SEARch:SLOPe:TLOWer 1.00E-08* *SEAR:SLOP:TLOW 1.00E-08* 查询斜率搜索限制条件的下限值：

	SEAR:SLOP:TLOW? 返回值： *1.00E-08*
关联命令	:SEARch:SLOPe:LIMit :SEARch:SLOPe:TUPPer

:SEARch:PULSe

:SEARch:PULSe:SOURce

描述	设置或查询脉宽搜索的信源。
命令格式	:SEARch:PULSe:SOURce <source> :SEARch:PULSe:SOURce?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置脉宽搜索的信源为 C2： *:SEARch:PULSe:SOURce C2* *SEAR:PULS:SOUR C2* 查询脉宽搜索的信源： *SEAR:PULS:SOUR?* 返回值： *C2*

:SEARch:PULSe:POLarity

描述	设置或查询脉宽搜索的极性。
命令格式	:SEARch:PULSe:POLarity <polarity> :SEARch:PULSe:POLarity?
参数说明	<polarity_type>:= {POSitive|NEGative} ● POSitive：正脉冲 ● NEGative：负脉冲
返回格式	{POSitive|NEGative}
示例	设置脉宽搜索的极性为正脉冲： *:SEARch:PULSe:POLarity POSitive*

	SEAR:PULS:POL POS 查询脉宽搜索的极性： *SEAR:PULS:POL?* 返回值： *POSitive*

:SEARch:PULSe:LEVel

描述	设置或查询脉宽搜索的搜索电平。
命令格式	:SEARch:PULSe:LEVel <level_value> :SEARch:PULSe:LEVel?
参数说明	<level_value>:= 数值，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD / [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X / [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD / [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置脉宽搜索的搜索电平为 0.5V： *:SEARch:PULSe:LEVel 5.00E-01* *SEAR:PULS:LEV 5.00E-01* 查询脉宽搜索的搜索电平： *SEAR:PULS:LEV?* 返回值： *5.00E-01*
关联命令	:SEARch:PULSe:SOURce

:SEARch:PULSe:LIMit

描述	设置或查询脉宽搜索的限制条件类型。
命令格式	:SEARch:PULSe:LIMit <type> :SEARch:PULSe:LIMit?
参数说明	<type>:= {LESSthan|GREATerthan|INNer|OUTer} ● LESSthan（<=）：小于时间值。 ● GREATerthan（>=）：大于时间值。 ● INNer（[--,--]）：时间值范围内。 ● OUTer（--][--）：时间值范围外。
返回格式	{LESSthan|GREATerthan|INNer|OUTer}
示例	设置脉宽搜索的限制条件类型为时间值范围内： *:SEARch:PULSe:LIMit INNer* *SEAR:PULS:LIM INN* 查询脉宽搜索的限制条件类型： *SEAR:PULS:LIM?* 返回值： *INNer*
关联命令	:SEARch:PULSe:TUPPer :SEARch:PULSe:TLOWer

:SEARch:PULSe:TUPPer

描述	设置或查询脉宽搜索限制条件的上限值。
命令格式	:SEARch:PULSe:TUPPer <value> :SEARch:PULSe:TUPPer?
参数说明	<value>:= 数值，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A	[1.00E-09,2.00E+01]
SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD	[-2.00E-09,2.00E+01]

SDS1000X HD SDS800X HD	
SHS800X SHS1000X	[2.00E-09,4.20E+00]

注意:

- 上限值不能小于下限值（下限值设置指令:SEARch:SLOPe:TLOWer）。
- 当限制类型为“>=”时，此命令不可用。

返回格式	浮点型 NR3 格式
示例	设置脉宽搜索限制条件的上限值为 30ns: *:SEARch:PULSe:TUPPer 3.00E-08* *SEAR:PULS:TUPP 3.00E-08* 查询脉宽搜索限制条件的上限值: *SEAR:PULS:TUPP?* 返回值: *3.00E-08*
关联命令	:SEARch:PULSe:LIMit :SEARch:PULSe:TLOWer

:SEARch:PULSe:TLOWer

描述	设置或查询脉宽搜索限制条件的下限值。
命令格式	:SEARch:PULSe:TLOWer <value> :SEARch:PULSe:TLOWer?
参数说明	<value>:= 数值，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表:

机型	范围
SDS7000A	[1.00E-09,2.00E+01]
SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-2.00E-09,2.00E+01]

SHS800X SHS1000X	[2.00E-09,4.20E+00]

注意:

- 下限值不能大于上限值（上限值设置指令:SEARch:SLOPe:TUPPer）。
- 当限制类型为“<=”时，此命令不可用。

返回格式	浮点型 NR3 格式
示例	设置脉宽搜索限制条件的下限值为 10ns: *:SEARch:PULSe:TLOWer 1.00E-08* *SEAR:PULS:TLOW 1.00E-08* 查询脉宽搜索限制条件的下限值: *SEAR:PULS:TLOW?* 返回值: *1.00E-08*
关联命令	:SEARch:PULSe:LIMit :SEARch:PULSe:TUPPer

:SEARch:INTerval

:SEARch:INTerval:SOURce

描述	设置或查询间隔搜索的信源。
命令格式	:SEARch:INTerval:SOURce <source> :SEARch:INTerval:SOURce?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置间隔搜索的信源为 C1: *:SEARch:INTerval:SOURce C1* *SEAR:INT:SOUR C1* 查询间隔搜索的信源: *SEAR:INT:SOUR?* 返回值: *C1*

:SEARch:INTerval:SLOPe

描述	设置或查询间隔搜索的斜率。
命令格式	:SEARch:INTerval:SLOPe <slope_type> :SEARch:INTerval:SLOPe?
参数说明	<slope_type>:= {RISing|FALLing} ● RISing：上升沿 ● FALLing：下降沿
返回格式	{RISing|FALLing}
示例	设置间隔搜索的斜率为上升沿： *:SEARch:INTerval:SLOPe RISing* *SEAR:INT:SLOP RIS* 查询间隔搜索的斜率： *SEAR:INT:SLOP?* 返回值： *RISing*

:SEARch:INTerval:LEVel

描述	设置或查询间隔搜索的搜索电平。
命令格式	:SEARch:INTerval:LEVel <level_value> :SEARch:INTerval:LEVel?
参数说明	<level_value>:= 数值，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

	SDS1000X HD SDS800X HD	
返回格式	浮点型 NR3 格式	
示例	设置间隔搜索的搜索电平为 0.5V： *:SEARch:INTerval:LEVel 5.00E-01* *SEAR:INT:LEV 5.00E-01* 查询间隔搜索的搜索电平： *SEAR:INT:LEV?* 返回值： *5.00E-01*	

:SEARch:INTerval:LIMit

描述	设置或查询间隔搜索的限制条件类型。
命令格式	:SEARch:INTerval:LIMit <type> :SEARch:INTerval:LIMit?
参数说明	<type>:= {LESSthan|GREATerthan|INNer|OUTer} ● LESSthan（<=）：小于时间值。 ● GREATerthan（>=）：大于时间值。 ● INNer（[--,--]）：时间值范围内。 ● OUTer（--][--）：时间值范围外。
返回格式	{LESSthan|GREATerthan|INNer|OUTer}
示例	设置间隔搜索的限制条件类型为时间值范围内： *:SEARch:INTerval:LIMit INNer* *SEAR:INT:LIM INN* 查询间隔搜索的限制条件类型： *SEAR:INT:LIM?* 返回值： *INNer*
关联命令	:SEARch:INTerval:TUPPer :SEARch:INTerval:TLOWer

:SEARch:INTerval:TUPPer

描述	设置或查询间隔搜索限制条件的上限值。
命令格式	:SEARch:INTerval:TUPPer <value> :SEARch:INTerval:TUPPer?
参数说明	<value>:= 数值，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： <table><tr><th>机型</th><th>范围</th></tr><tr><td>SDS7000A</td><td>[1.00E-09,2.00E+01]</td></tr><tr><td>SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD</td><td>[-2.00E-09,2.00E+01]</td></tr><tr><td>SHS800X SHS1000X</td><td>[2.00E-09,4.20E+00]</td></tr></table> **注意**： ● 上限值不能小于下限值（下限值设置指令:SEARch:INTerval:TLOWer）。 ● 当限制类型为“>=”时，此命令不可用。
返回格式	浮点型 NR3 格式
示例	设置间隔搜索限制条件的上限值为 30ns： *:SEARch:INTerval:TUPPer 3.00E-08* *SEAR:INT:TUPP 3.00E-08* 查询间隔搜索限制条件的上限值： *SEAR:INT:TUPP?* 返回值： *3.00E-08*
关联命令	:SEARch:INTerval:LIMit :SEARch:INTerval:TLOWer

:SEARch:INTerval:TLOWer

描述	设置或查询间隔搜索限制条件的下限值。
命令格式	:SEARch:INTerval:TLOWer <value> :SEARch:INTerval:TLOWer?
参数说明	<value>:= 数值，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： <table><tr><th>机型</th><th>范围</th></tr><tr><td>SDS7000A</td><td>[1.00E-09,2.00E+01]</td></tr><tr><td>SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD</td><td>[-2.00E-09,2.00E+01]</td></tr><tr><td>SHS800X SHS1000X</td><td>[2.00E-09,4.20E+00]</td></tr></table> **注意**： ● 下限值不能大于上限值（上限值设置指令:SEARch:INTerval:TUPPer）。 ● 当限制类型为“<=”时，此命令不可用。
返回格式	浮点型 NR3 格式
示例	设置间隔搜索限制条件的下限值为 10ns： *:SEARch:INTerval:TLOWer 1.00E-08* *SEAR:INT:TLOW 1.00E-08* 查询间隔搜索限制条件的下限值： *SEAR:INT:TLOW?* 返回值： *1.00E-08*
关联命令	:SEARch:INTerval:LIMit :SEARch:INTerval:TUPPer

:SEARch:RUNT

:SEARch:RUNT:SOURce

描述	设置或查询欠幅搜索的信源。
命令格式	:SEARch:RUNT:SOURce <source> :SEARch:RUNT:SOURce?
参数说明	<source>:= {C<n>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置欠幅搜索的信源为 C1： *:SEARch:RUNT:SOURce C1* *SEAR:RUNT:SOUR C1* 查询欠幅搜索的信源： *SEAR:RUNT:SOUR?* 返回值： *C1*

:SEARch:RUNT:POLarity

描述	设置或查询欠幅搜索的极性。
命令格式	:SEARch:RUNT:POLarity <polarity> :SEARch:RUNT:POLarity?
参数说明	<polarity_type>:= {POSitive|NEGative} ● POSitive：正脉冲 ● NEGative：负脉冲
返回格式	{POSitive|NEGative}
示例	设置欠幅搜索的极性为正脉冲： *:SEARch:RUNT:POLarity POSitive* *SEAR:RUNT:POL POS* 查询欠幅搜索的极性： *SEAR:RUNT:POL?* 返回值： *POSitive*

:SEARch:RUNT:HLEVel

描述	设置或查询欠幅搜索的高电平。
命令格式	:SEARch:RUNT:HLEVel <high_level_value> :SEARch:RUNT:HLEVel?
参数说明	<high_level_value>:= 数值，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **注意**：高电平不能小于低电平（低电平设置指令:SEARch:RUNT:LLEVel）。
返回格式	浮点型 NR3 格式
示例	设置欠幅搜索的高电平为 0.5V： *:SEARch:RUNT:HLEVel 5.00E-01* *SEAR:RUNT:HLEV 5.00E-01* 查询欠幅搜索的高电平： *SEAR:RUNT:HLEV?* 返回值： *5.00E-01*
关联命令	:SEARch:RUNT:LLEVel

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

:SEARch:RUNT:LLEVel

描述	设置或查询欠幅搜索的低电平。
命令格式	:SEARch:RUNT:LLEVel <low_level_value> :SEARch:RUNT:LLEVel?
参数说明	<low_level_value>:= 数值，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **注意**：低电平不能大于高电平（高电平设置指令:SEARch:RUNT:HLEVel）。
返回格式	浮点型 NR3 格式
示例	设置欠幅搜索的低电平为-0.5V： *:SEARch:RUNT:LLEVel -5.00E-01* *SEAR:RUNT:LLEV -5.00E-01* 查询欠幅搜索的低电平： *SEAR:RUNT:LLEV?* 返回值： *-5.00E-01*
关联命令	:SEARch:RUNT:HLEVel

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

:SEARch:RUNT:LIMit

描述	设置或查询欠幅搜索的限制条件类型。
命令格式	:SEARch:RUNT:LIMit <type> :SEARch:RUNT:LIMit?
参数说明	<type>:= {LESSthan|GREATerthan|INNer|OUTer} ● LESSthan（<=）：小于时间值。 ● GREATerthan（>=）：大于时间值。 ● INNer（[--,--]）：时间值范围内。 ● OUTer（--][--）：时间值范围外。
返回格式	{LESSthan|GREATerthan|INNer|OUTer}
示例	设置欠幅搜索的限制条件类型为时间值范围内： *:SEARch:RUNT:LIMit INNer* *SEAR:RUNT:LIM INN* 查询欠幅搜索的限制条件类型： *SEAR:RUNT:LIM?* 返回值： *INNer*
关联命令	:SEARch:RUNT:TUPPer :SEARch:RUNT:TLOWer

:SEARch:RUNT:TUPPer

描述	设置或查询欠幅搜索限制条件的上限值。
命令格式	:SEARch:RUNT:TUPPer <value> :SEARch:RUNT:TUPPer?
参数说明	<value>:= 数值，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A	[1.00E-09,2.00E+01]
SDS6000 Pro SDS6000A SDS6000L SDS5000X HD SDS5000X SDS3000X HD SDS2000X Plus	[-2.00E-09,2.00E+01]

SDS2000X HD SDS1000X HD SDS800X HD	
SHS800X SHS1000X	[2.00E-09,4.20E+00]

注意:

- 上限值不能小于下限值（下限值设置指令:SEARch:RUNT:TLOWer）。
- 当限制类型为“>=”时，此命令不可用。

返回格式	浮点型 NR3 格式
示例	设置欠幅搜索限制条件的上限值为 30ns: *:SEARch:RUNT:TUPPer 3.00E-08* *SEAR:RUNT:TUPP 3.00E-08* 查询欠幅搜索限制条件的上限值: *SEAR:RUNT:TUPP?* 返回值: *3.00E-08*
关联命令	:SEARch:RUNT:LIMit :SEARch:RUNT:TLOWer

:SEARch:RUNT:TLOWer

描述	设置或查询幅度搜索限制条件的下限值。
命令格式	:SEARch:RUNT:TLOWer <value> :SEARch:RUNT:TLOWer?
参数说明	<value>:= 数值，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表:

机型	范围
SDS7000A	[1.00E-09,2.00E+01]
SDS6000 Pro SDS6000A SDS6000L SDS5000X HD SDS5000X SDS3000X HD SDS2000X Plus	[-2.00E-09,2.00E+01]

SDS2000X HD SDS1000X HD SDS800X HD	
SHS800X SHS1000X	[2.00E-09,4.20E+00]

注意:

- 下限值不能大于上限值（上限值设置指令:SEARch:RUNT:TUPPer）。
- 当限制类型为“<=”时，此命令不可用。

返回格式	浮点型 NR3 格式
示例	设置欠幅搜索限制条件的下限值为 10ns: *:SEARch:RUNT:TLOWer 1.00E-08* *SEAR:RUNT:TLOW 1.00E-08* 查询欠幅搜索限制条件的下限值: *SEAR:RUNT:TLOW?* 返回值: *1.00E-08*
关联命令	:SEARch:RUNT:LIMit :SEARch:RUNT:TUPPer

5.25 SYSTem 命令系统

:SYSTem 命令子系统控制示波器的系统功能。

:SYSTem:BUZZer

描述	设置或查询蜂鸣器的开关状态。
命令格式	:SYSTem:BUZZer <state> :SYSTem:BUZZer?
参数说明	<state>:= {ON|OFF}
返回格式	{ON/OFF}
示例	设置蜂鸣器开启： *:SYSTem:BUZZer ON* *SYST:BUZZ ON* 查询蜂鸣器开关状态： *SYST:BUZZ?* 返回值： *ON*

:SYSTem:CLOCk

描述	设置或查询示波器时钟源和内部 10MHz 时钟输出状态。
命令格式	:SYSTem:CLOCk <source> :SYSTem:CLOCk?
参数说明	<source>:= {EXT|IN_ON|IN_OFF} ● EXT：选择外部时钟源，内部 10MHz 输出自动关闭 ● IN_ON：选择内部时钟源，启用 10MHz 输出 ● IN_OFF：选择内部时钟源，禁用 10MHz 输出
返回格式	{EXT|IN_ON|IN_OFF}
示例	设置示波器时钟源为内部，打开 10MHz 输出： *:SYSTem:CLOCk IN_ON* *SYST:CLOCK IN_ON* 查询时钟源状态： *SYST:CLOC?* 返回值： *IN_ON*

:SYSTem:COMMunicate:LAN:GATeway

描述	设置或查询示波器局域网的网关。
命令格式	:SYSTem:COMMunicate:LAN:GATeway <qstring> :SYSTem:COMMunicate:LAN:GATeway?
参数说明	<qstring>:= 带引号的 ASCII 字符串。
返回格式	<qstring>
示例	设置示波器内部网络接口的网关为“10.12.0.1”: *:SYSTem:COMMunicate:LAN:GATeway "10.12.0.1"* *SYST:COMM:LAN:GAT "10.12.0.1"* 查询示波器内部网络接口的网关: *SYST:COMM:LAN:GAT?* 返回值: *"10.12.0.1"*
关联命令	:SYSTem:COMMunicate:LAN:IPADdress :SYSTem:COMMunicate:LAN:SMASk :SYSTem:COMMunicate:LAN:TYPE

:SYSTem:COMMunicate:LAN:IPADdress

描述	设置或查询示波器局域网的 IP 地址。
命令格式	:SYSTem:COMMunicate:LAN:IPADdress <qstring> :SYSTem:COMMunicate:LAN:IPADdress?
参数说明	<qstring>:= 带引号的 ASCII 字符串。
返回格式	<qstring>
示例	设置示波器内部网络接口的 IP 地址为“10.12.255.229”: *:SYSTem:COMMunicate:LAN:IPADdress "10.12.255.229"* *SYST:COMM:LAN:IPAD "10.12.255.229"* 查询示波器内部网络接口的 IP 地址: *SYST:COMM:LAN:IPAD?* 返回值: *"10.12.255.229"*
关联命令	:SYSTem:COMMunicate:LAN:IPADdress :SYSTem:COMMunicate:LAN:SMASk :SYSTem:COMMunicate:LAN:TYPE

:SYSTem:COMMunicate:LAN:MAC

描述	查询示波器的 MAC 地址。
命令格式	:SYSTem:COMMunicate:LAN:MAC?
返回格式	<byte1>:<byte2>:<byte3>:<byte4>:<byte5>:<byte6 <byte>:= 十六进制数值
示例	查询示波器的 MAC 地址： *SYST:COMM:LAN:MAC?* 返回值： *00:01:D2:0C:00:A0*

:SYSTem:COMMunicate:LAN:SMASk

描述	设置或查询示波器局域网的子网掩码。
命令格式	:SYSTem:COMMunicate:LAN:SMASk <qstring> :SYSTem:COMMunicate:LAN:SMASk?
参数说明	<qstring>:= 带引号的 ASCII 字符串。
返回格式	<qstring>
示例	设置示波器内部网络接口的子网掩码为“255.255.0.0”： *:SYSTem:COMMunicate:LAN:SMASk "255.255.0.0"* *SYST:COMM:LAN:SMAS "255.255.0.0"* 查询示波器内部网络接口的子网掩码： *SYST:COMM:LAN:SMAS?* 返回值： *"255.255.0.0"*
关联命令	:SYSTem:COMMunicate:LAN:GATeway :SYSTem:COMMunicate:LAN:IPADdress :SYSTem:COMMunicate:LAN:TYPE

:SYSTem:COMMunicate:LAN:TYPE

描述	设置或查询示波器局域网配置的方式。
命令格式	:SYSTem:COMMunicate:LAN:TYPE <state> :SYSTem:COMMunicate:LAN:TYPE?
参数说明	<state>:= {STATIC|DHCP}

	● STATIC：手动配置网络设置。 ● DHCP：从具有 DHCP 服务器的局域网上自动获取示波器的 IP 地址、子网掩码和网关。
返回格式	{STATIC|DHCP}
示例	设置示波器局域网配置方式为 DHCP： *:SYSTem:COMMunicate:LAN:TYPE DHCP* *SYST:COMM:LAN:TYPE DHCP* 查询示波器局域网配置方式： *SYST:COMM:LAN:TYPE?* 返回值： *DHCP*
关联命令	:SYSTem:COMMunicate:LAN:GATeway :SYSTem:COMMunicate:LAN:IPADdress :SYSTem:COMMunicate:LAN:SMASk

:SYSTem:COMMunicate:VNCPort

描述	设置或查询示波器 VNC 端口号。
命令格式	:SYSTem:COMMunicate:VNCPort <value> :SYSTem:COMMunicate:VNCPort?
参数说明	<value>:= 端口号，整型 NR1 格式，范围 [5900,5999]
返回格式	整型 NR1 格式
示例	设置示波器的 VNC 端口号为 5903： *:SYSTem:COMMunicate:VNCPort 5903* *SYST:COMM:VNCP 5903* 查询示波器的 VNC 端口号： *SYST:COMM:VNCP?* 返回值： *5903*

:SYSTem:DATE

描述	设置或查询示波器系统日期。
命令格式	:SYSTem:DATE <date> :SYSTem:DATE?

参数说明	<date>:= 整型 NR1 格式。8 位有效数字从高位到低位，前 4 位代表年份，接下来 2 位代表月份，最后 2 位代表日期。
返回格式	整型 NR1 格式
示例	设置示波器的系统时间为 2019 年 8 月 19 日： *:SYSTem:DATE 20190819* *SYST:DATE 20190819* 查询示波器的系统时间： *SYST:DATE?* 返回值： *20190819*
关联命令	:SYSTem:TIME

:SYSTem:EDUMode

描述	设置或查询示波器教育模式开关。教育模式下可禁用 Auto Setup、Measure、Cursors 功能。
命令格式	:SYSTem:EDUMode <func>,<lock> :SYSTem:EDUMode? :SYSTem:EDUMode? <func>
参数说明	<func>:= {AUTOSet|MEASure|CURSor} <lock>:= {ON|OFF} ● ON：启用所选功能 ● OFF：禁用所选功能
返回格式	查询时未指定<func>，则返回： AUTOSet,<lock>;MEASure,<lock>;CURSor,<lock> 查询时指定<func>，则返回： {ON|OFF}
示例	示波器禁用 Auto Setup 功能： *:SYSTem:EDUMode AUTOSet,OFF* *SYST:EDUM AUTOS,OFF* 查询示波器 Auto Setup 功能锁定状态: *SYST:EDUM? AUTOS* 返回值： *OFF*

:SYSTem:LANGuage

描述	设置或查询示波器显示语言。
命令格式	:SYSTem:LANGuage <language> :SYSTem:LANGuage?
参数说明	<language>:= {SCHinese|TCHinese|ENGLish|FRENch|JAPanese|KORean|DEUTsch|ESPan|RUSSian|ITALiana|PORTuguese}
返回格式	{SCHinese|TCHinese|ENGLish|FRENch|JAPanese|KORean|DEUTsch|ESPan|RUSSian|ITALiana|PORTuguese}
示例	设置示波器的显示语言为英语： *:SYSTem:LANGuage ENGLish* *SYST:LANG ENGL* 查询示波器的显示语言： *SYST:LANG?* 返回值： *ENGLish*

:SYSTem:MENU

描述	设置或查询菜单开关状态。该命令仅适用于有菜单开关按键 Menu 的示波器。
命令格式	:SYSTem:MENU <state> :SYSTem:MENU?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	开启示波器菜单： *:SYSTem:MENU ON* *SYST:MENU ON* 查询示波器菜单状态： *SYST:MENU?* 返回值： *ON*

:SYSTem:NSTorage

描述	设置或查询挂载网络存储器的参数。
命令格式	:SYSTem:NSTorage <path>,<user>,<pwd>,<anon>,<auto_con>,<rem_path>,<rem_user>,<rem_pwd> :SYSTem:NSTorage?
参数说明	<path>:= 带引号的 ASCII 文本字符串，表示服务器挂载路径 <user>:= 带引号的 ASCII 文本字符串，表示用户名 <pwd>:= 带引号的 ASCII 文本字符串，表示用户密码，格式为带引号的字符串 <anon>:={1|0}，匿名标志，1 代表 ON，0 代表 OFF <auto_con>:={1|0}，自动连接标志，1 代表 ON，0 代表 OFF <rem_path>:={1|0}，记住路径标志，1 代表 ON，0 代表 OFF <rem_user>:={1|0}，记住用户名标志，1 代表 ON，0 代表 OFF <rem pwd>:={1|0}，记住密码标志，1 代表 ON，0 代表 OFF
返回格式	<path>,<user>,<pwd>,<anon>,<auto_con>,<rem_path>,<rem_user>,<rem_pwd> **注意**：出于安全考虑，密码将始终返回“***”。
示例	设置连接服务器参数信息： *:SYSTem:NSTorage "//10.12.255.239/nfs","","",0,0,1,0,0* *SYST:NST "//10.12.255.239/nfs","","",0,0,1,0,0* 查询连接服务器参数信息： *SYST:NST?* 返回值： *"//10.12.255.239/nfs","","***",0,0,1,0,0*

:SYSTem:NSTorage:CONNect

描述	设置示波器连接网络存储器。
命令格式	:SYSTem:NSTorage:CONNect
示例	设置示波器连接服务器： *:SYSTem:NSTorage:CONNect* *SYST:NST:CONN*

:SYSTem:NSTorage:DISConnect

描述	设置示波器断开网络存储器。
命令格式	:SYSTem:NSTorage:DISConnect
示例	设置示波器断开服务器连接： *:SYSTem:NSTorage:DISCONNect* *SYST:NST:DISC*

:SYSTem:NSTorage:STATus

描述	查询示波器网络存储器的连接状态。
命令格式	:SYSTem:NSTorage:STATus?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	查询示波器与服务器的连接状态： *SYST:NST:STAT?* 返回值： *OFF*

:SYSTem:PON

描述	设置或查询上电开机开关状态。如果该功能启用，上电后示波器将自动开机。
命令格式	:SYSTem:PON <state> :SYSTem:PON?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置上电开机开启： *:SYSTem:PON ON* *SYST:PON ON* 查询上电开机状态： *SYST:PON?* 返回值： *ON*

:SYSTem:REBoot

描述	重启示波器。
命令格式	:SYSTem:REBoot
示例	重启示波器： *:SYSTem:REBoot* *SYST:REB*
关联命令	:SYSTem:SHUTdown

:SYSTem:REMote

描述	设置或查询远程控制状态。当远程控制开启时，触摸屏、前面板按键和其他外设都会被禁用，示波器界面提示"Remote"。
命令格式	:SYSTem:REMote <state> :SYSTem:REMote?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置远程控制开启： *:SYSTem:REMote ON* *SYST:REM ON* 查询远程控制状态： *SYST:REM?* 返回值： *ON*

:SYSTem:SELFCal

描述	设置或查询示波器自校正状态。
命令格式	:SYSTem:SELFCal :SYSTem:SELFCal?
返回格式	{DOING|DONE}
示例	设置示波器开始自校正： *:SYSTem:SELFCal* *SYST:SELFC* 查询示波器自校正是否完成：

	SYST:SELFC? 返回值： *DONE*

:SYSTem:SHUTdown

描述	示波器关机。
命令格式	:SYSTem:SHUTdown
示例	关闭示波器： *:SYSTem:SHUTdown* *SYST:SHUT*
关联命令	:SYSTem:REBoot

:SYSTem:SSAVer

描述	设置或查询示波器屏幕保护状态。在预设时间到达后将自动关闭显示器输出。
命令格式	:SYSTem:SSAVer <time> :SYSTem:SSAVer?
参数说明	<time>:= {OFF|1MIN|5MIN|10MIN|30MIN|60MIN}
返回格式	{OFF|1MIN|5MIN|10MIN|30MIN|60MIN}
示例	设置示波器屏幕保护时间为 10min： *:SYSTem:SSAVer 10MIN* *SYST:SSAV 10MIN* 查询示波器屏幕保护时间： *SYST:SSAV?* 返回值： *10MIN*

:SYSTem:TIME

描述	设置或查询示波器当前时间（24 小时制）。
命令格式	:SYSTem:TIME <time> :SYSTem:TIME?
参数说明	<time>:= 整型 NR1 格式。6 位有效数字从高位到低位，前 2 位代表时，中间

	2 位代表分，最后 2 位代表秒
返回格式	<time>
示例	设置示波器当前时间为 08:10:40： *:SYSTem:TIME 081040* *SYST:TIME 081040* 查询示波器当前时间： *SYST:TIME?* 返回值： *081040*
关联命令	:SYSTem:DATE

:SYSTem:TOUCh

描述	设置或查询触摸屏开关状态。
命令格式	:SYSTem:TOUCh <state> :SYSTem:TOUCh?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置触摸功能开启： *:SYSTem:TOUCh ON* *SYST:TOUC ON* 查询触摸功能状态： *SYST:TOUC?* 返回值： *ON*

5.26 TIMebase 命令系统

:Timebase 命令子系统控制示波器水平系统功能，可以控制主窗口和 Zoom 缩放窗口的水平档位、水平延时等参数。

:TIMebase:DELay

描述	设置或查询水平触发延时。
命令格式	:TIMebase:DELay <delay_value> :TIMebase:DELay?
参数说明	<delay_value>:= 数值，浮点型 NR3 格式。
返回格式	浮点型 NR3 格式
示例	设置水平触发延时为 10us： *:TIMebase:DELay 1.00E-05* *TIM:DEL 1.00E-05* 查询水平触发延时： *TIM:DEL?* 返回值： *1.00E-05*
关联命令	:TIMebase:SCALe

:TIMebase:REFerence

描述	设置或查询水平扩展参考策略。
命令格式	:TIMebase:REFerence <type> :TIMebase:REFerence?
参数说明	<type>:= {DELay|POSition} ● DELay：延时固定。水平档位变化时，水平延时值不变。在该模式下可设置水平参考点位置，水平延时值以参考点为中心保持不变。 ● POSition：位置固定。在水平档位变化时，水平延时按屏幕上固定网格的位置保持不变。
返回格式	{DELay|POSition}
示例	设置水平扩展策略为延时固定： *:TIMebase:REFerence DELay* *TIM:REF DEL*

	查询水平扩展策略： *TIM:REF?* 返回值： *DELay*

:TIMebase:REFerence:POSition

描述	扩展策略为延时固定时，设置或查询水平参考中心位置。
命令格式	:TIMebase:REFerence:POSition <value> :TIMebase:REFerence:POSition?
参数说明	<value>:= 百分比数值，整型 NR1 格式，该值范围为 [0,100]
返回格式	整型 NR1 格式
示例	设置水平参考中心为 20%： *:TIMebase:REFerence:POSition 20* *TIM:REF:POS 20* 查询水平参考中心位置： *TIM:REF:POS?* 返回值： *20*
关联命令	:TIMebase:REFerence

:TIMebase:SCALe

描述	设置或查询主窗口时基。由于扩展点数的限制，设置时基从大到小时，示波器会自动匹配当前可以设置的最小时基。
命令格式	:TIMebase:SCALe <value> :TIMebase:SCALe?
参数说明	<value>:= 数值，浮点型 NR3 格式 **注意**：时基的范围因机型而异，详情请参考数据手册。
返回格式	浮点型 NR3 格式
示例	设置主窗口时基为 100ns/div： *:TIMebase:SCALe 1.00E-07* *TIM:SCAL 1.00E-07* 查询主窗口时基：

	TIM:SCAL? 返回值： *1.00E-07*
关联命令	:TIMebase:DELay

:TIMebase:WINDow

描述	设置或查询 Zoom 窗口的开关状态。
命令格式	:TIMebase:WINDow <state> :TIMebase:WINDow?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置 Zoom 窗口开启： *:TIMebase:WINDow ON* *TIM:WIND ON* 查询 Zoom 窗口状态： *TIM:WIND?* 返回值： *ON*
关联命令	:TIMebase:WINDow:DELay :TIMebase:WINDow:SCALe

:TIMebase:WINDow:DELay

描述	设置或查询 Zoom 窗口水平延时。
命令格式	:TIMebase:WINDow:DELay <delay_value> :TIMebase:WINDow:DELay?
参数说明	<delay_value>:= 数值，浮点型 NR3 格式 **注意**： ● 主窗口时基和水平延时共同决定 Zoom 窗口的水平延时范围，Zoom 窗口范围必须在主窗口范围内。 ● Zoom 窗口延时设置值超出范围时，将会自动设置至最接近的可设置值。
返回格式	浮点型 NR3 格式
示例	设置 Zoom 窗口延时为 100ns/div：

	:TIMebase:WINDow:DELay 1.00E-03 *TIM:WIND:DEL 1.00E-03* 查询 Zoom 窗口延时： *TIM:WIND:DEL?* 返回值： *1.00E-03*
关联命令	:TIMebase:WINDow:SCALe :TIMebase:SCALe :TIMebase:DELay

:TIMebase:WINDow:SCALe

描述	设置或查询 Zoom 窗口时基。
命令格式	:TIMebase:WINDow:SCALe <scale_value> :TIMebase:WINDow:SCALe
参数说明	<scale_value>:= 数值，浮点型 NR3 格式 **注意**：Zoom 窗口的时基不能大于主窗口的时基，否则将自动匹配主窗口时基。
返回格式	浮点型 NR3 格式
示例	设置 Zoom 窗口时基为 1ms/div： *:TIMebase:WINDow:SCALe 1.00E-03* *TIM:WIND:SCAL 1.00E-03* 查询 Zoom 窗口时基： *TIM:WIND:SCAL?* 返回值： *1.00E-03*
关联命令	:TIMebase:WINDow:DELay :TIMebase:SCALe :TIMebase:DELay

5.27 TPPA 命令系统

:TPPA 命令子系统用于控制三相电分析功能。

:TPPA:DUT<a>

描述	设置或查询三相电分析待测设备开关状态。
命令格式	:TPPA:DUT<a> <state> :TPPA:DUT<a>?
参数说明	<a>:= {1|2}，待测设备序号 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置三相电分析 DUT1 开启： *:TPPA:DUT1 ON* *TPPA:DUT1 ON* 查询三相电分析 DUT1 开关状态： *TPPA:DUT1?* 返回值： *ON*

:TPPA:DUT<a>:GROup<b>

描述	设置或查询三相电分析输入输出开关状态。
命令格式	:TPPA:DUT<a>:GROup<b> <state> :TPPA:DUT<a>:GROup<b>?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启三相电分析 DUT1 的输入： *:TPPA:DUT1:GROup1 ON* *TPPA:DUT1:GRO1 ON*

	查询三相电分析 DUT1 的输入开关状态： *TPPA:DUT1:GRO1?* 返回值： *ON*

:TPPA:DUT<a>:GROup<b>:CONFig:WIRing

描述	设置或查询三相电分析布线类型。
命令格式	:TPPA:DUT<a>:GROup<b>:CONFig:WIRing <type> :TPPA:DUT<a>:GROup<b>:CONFig:WIRing?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <type>:={P1W2[,<line>] | P2W3 | P3W3V2I2[,<line>] | P3W3V3I3[,<line>] | P3W4} ● P1W2：可设置 1V1I，可选线路类型<line>:={AC|DC} ● P2W3：可设置 2V2I ● P3W3：可设置 2V2I，3V3I。2V2I 下可选线路类型 <line>:={ABCB|ACBC|CABA}；3V3I 下可选线路类型 <line>:={CONVert|NCONvert} ● P3W4：可设置 3V4I
返回格式	{P1W2[,<line>] | P2W3 | P3W3V2I2[,<line>] | P3W3V3I3[,<line>] | P3W4}
示例	设置三相电分析 DUT1 的输入信号的布线类型为 P1W2，线路类型为 AC： *:TPPA:DUT1:GROup1:CONFig:WIRing P1W2,AC* *TPPA:DUT1:GRO1:CONF:WIR P1W2,AC* 查询三相电分析 DUT1 的输入信号的布线类型： *TPPA:DUT1:GRO1:CONF:WIR?* 返回值： *P1W2,AC*

:TPPA:DUT<a>:GROup<b>:CONFig:VSOurce<c>

描述	设置或查询三相电分析各相位电压通道。
命令格式	:TPPA:DUT<a>:GROup<b>:CONFig:VSOurce<c> <source> :TPPA:DUT<a>:GROup<b>:CONFig:VSOurce<c>?

参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <c>:= {1|2|3}，相位序号 <source>:= {C<n>|F<x>|M<m>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式
返回格式	{C<n>|F<x>|M<m>}
示例	设置三相电分析 DUT1 的输入信号相位 A 电压通道为 C1： *:TPPA:DUT1:GROup1:CONFig:VSOurce1 C1* *:TPPA:DUT1:GRO1:CON:VSO1 C1* 查询三相电分析 DUT1 的输入信号相位 A 电压通道： *TPPA:DUT1:GRO1:CON:VSO1?* 返回值： *C1*

:TPPA:DUT<a>:GROup<b>:CONFig:ISOurce<c>

描述	设置或查询三相电分析各相位电流通道。
命令格式	:TPPA:DUT<a>:GROup<b>:CONFig:ISOurce<c> <source> :TPPA:DUT<a>:GROup<b>:CONFig:ISOurce<c>?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <c>:= {1|2|3}，相位序号 <source>:= {C<n>|F<x>|M<m>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式
返回格式	{C<n>|F<x>|M<m>}
示例	设置三相电分析 DUT1 的输入信号相位 A 电流通道为 C1：

	:TPPA:DUT1:GROup1:CONFig:ISOurce1 C1 *:TPPA:DUT1:GRO1:CON:ISO1 C1* 查询三相电分析 DUT1 的输入信号相位 A 电流通道： *TPPA:DUT1:GRO1:CON:ISO1?* 返回值： *C1*

:TPPA:DUT<a>:GROup<b>:CONFig:EQUalifier

描述	设置或查询三相电分析边缘限定通道。
命令格式	:TPPA:DUT<a>:GROup<b>:CONFig:EQUalifier <source> :TPPA:DUT<a>:GROup<b>:CONFig:EQUalifier?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <source>:= {C<n>|F<x>|M<m>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● M<m>:= 内存波形，<m>:= 序号，整型 NR1 格式
返回格式	{C<n>|F<x>|M<m>}
示例	设置三相电分析 DUT1 的输入信号边缘限定通道为 C1： *:TPPA:DUT1:GROup1:CONFig:EQUalifier C1* *:TPPA:DUT1:GRO1:CON:EQU C1* 查询三相电分析 DUT1 的输入信号边缘限定通道： *TPPA:DUT1:GRO1:CON:EQU?* 返回值： *C1*

:TPPA:DUT<a>:GROup<b>:CONFig:LPFilter:CFRequency

描述	设置或查询三相电分析低通滤波器截止频率。
命令格式	:TPPA:DUT<a>:GROup<b>:CONFig:LPFilter:CFRequency <value> :TPPA:DUT<a>:GROup<b>:CONFig:LPFilter:CFRequency?
参数说明	<a>:= {1|2}，待测设备序号

	<b>:= {1|2} ● 1: 输入信号 ● 2: 输出信号 <value>:= 数值，浮点型 NR3 格式，该值的范围为[1E2,1E6]
返回格式	浮点型 NR3 格式
示例	设置三相电分析 DUT1 的输入低通滤波器截止频率为 1kHz: *:TPPA:DUT1:GROup1:CONFig:LPFilter:CFRequency 1.00E+03* *TPPA:DUT1:GRO1:CONF:LPF:CFR 1.00E+03* 查询三相电分析 DUT1 的输入低通滤波器截止频率: *TPPA:DUT1:GRO1:CONF:LPF:CFR?* 返回值: *1.00E+03*

:TPPA:DUT<a>:GROup<b>:MEASure:PQUality

描述	设置或查询三相电分析电能质量测量项开关状态。
命令格式	:TPPA:DUT<a>:GROup<b>:MEASure:PQUality <state> :TPPA:DUT<a>:GROup<b>:MEASure:PQUality?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1: 输入信号 ● 2: 输出信号 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启三相电分析 DUT1 的输入电能质量测量项: *:TPPA:DUT1:GROup1:MEASure:PQUality ON* *TPPA:DUT1:GRO1:MEAS:PQU ON* 查询三相电分析 DUT1 的输入电能质量开关状态: *TPPA:DUT1:GRO1:MEAS:PQU?* 返回值: *ON*

:TPPA:DUT<a>:GROup<b>:MEASure:PQUality:PHASor

描述	设置或查询三相电分析电能质量向量图开关状态。
命令格式	:TPPA:DUT<a>:GROup<b>:MEASure:PQUality:PHASor ＜state> :TPPA:DUT<a>:GROup<b>:MEASure:PQUality:PHASor?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启三相电分析 DUT1 的输入电能质量向量图： *:TPPA:DUT1:GROup1:MEASure:PQUality:PHASor ON* *TPPA:DUT1:GRO1:MEAS:PQU:PHA ON* 查询三相电分析 DUT1 的输入电能质量向量图开关状态： *TPPA:DUT1:GRO1:MEAS:PQU:PHA?* 返回值： *ON*

:TPPA:DUT<a>:GROup<b>:MEASure:RIPPle

描述	设置或查询三相电分析纹波分析测量项开关状态。
命令格式	:TPPA:DUT<a>:GROup<b>:MEASure:RIPPle ＜state> :TPPA:DUT<a>:GROup<b>:MEASure:RIPPle?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启打开三相电分析 DUT1 的输入纹波分析测量项： *:TPPA:DUT1:GROup1:MEASure:RIPPle ON* *TPPA:DUT1:GRO1:MEAS:RIPP ON* 查询三相电分析 DUT1 的输入菜单纹波分析开关状态： *TPPA:DUT1:GRO1:MEAS:RIPP?*

	返回值： *ON*

:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic

描述	设置或查询三相电分析电流谐波测量项开关状态。
命令格式	:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic ＜state> :TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启三相电分析 DUT1 的输入电流谐波测量项： *:TPPA:DUT1:GROup1:MEASure:IHARmonic ON* *TPPA:DUT1:GRO1:MEAS:IHAR ON* 查询三相电分析 DUT1 的输入电流谐波开关状态： *TPPA:DUT1:GRO1:MEAS:IHAR?* 返回值： *ON*

:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:FREQuency

描述	设置或查询三相电分析电流谐波线路频率。
命令格式	:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:FREQuency <value> :TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:FREQuency?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <value>:= {AUTO_HZ|HZ50|HZ60|HZ400}
返回格式	{AUTO_HZ|HZ50|HZ60|HZ400}
示例	设置三相电分析 DUT1 的输入电流谐波线路频率为 50Hz：

	:TPPA:DUT1:GROup1:MEASure:IHARmonic:FREQuency HZ50 *TPPA:DUT1:GRO1:MEAS:IHAR:FREQ HZ50* 查询三相电分析 DUT1 的输入电流谐波线路频率： *TPPA:DUT1:GRO1:MEAS:IHAR:FREQ?* 返回值： *HZ50*

:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:THDStandard

描述	设置或查询三相电分析电流谐波总谐波失真标准类型
命令格式	:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:THDStandard <type> :TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:THDStandard?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <type>:= {IEC|CSA}
返回格式	{IEC|CSA}
示例	设置三相电分析 DUT1 的输入电流谐波的总谐波失真标准为 IEC： *:TPPA:DUT1:GROup1:MEASure:IHARmonic:THDStandard IEC* *TPPA:DUT1:GRO1:MEAS:IHAR:THDS IEC* 查询三相电分析 DUT1 的输入电流谐波的总谐波失真标准： *TPPA:DUT1:GRO1:MEAS:IHAR:THDS?* 返回值： *IEC*

:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:STANdard

描述	设置或查询三相电分析电流谐波标准类型
命令格式	:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:STANdard <type> :TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:STANdard?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号

	<type>:= {IEC_61000_3_2_A|IEC_61000_3_2_B|IEC_61000_3_2_C|IEC_61000_3_2_D}
返回格式	{IEC_61000_3_2_A|IEC_61000_3_2_B|IEC_61000_3_2_C|IEC_61000_3_2_D}
示例	设置三相电分析 DUT1 的输入电流谐波标准为 IEC_61000_3_2_A： *:TPPA:DUT1:GROup1:MEASure:IHARmonic:STANdard IEC_61000_3_2_A* *TPPA:DUT1:GRO1:MEAS:IHAR:STAN IEC_61000_3_2_A* 查询三相电分析 DUT1 的输入电流谐波标准： *TPPA:DUT1:GRO1:MEAS:IHAR:STAN?* 返回值： *IEC_61000_3_2_A*

:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:TABLe

描述	设置或查询三相电分析电流谐波测量结果表格开关状态
命令格式	:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:TABLe <state> :TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:TABLe?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启三相电分析 DUT1 的输入电流谐波测量结果表格： *:TPPA:DUT1:GROup1:MEASure:IHARmonic:TABLe ON* *TPPA:DUT1:GRO1:MEAS:IHAR:TABL ON* 查询三相电分析 DUT1 的输入电流谐波测量结果表格开关状态： *TPPA:DUT1:GRO1:MEAS:IHAR:TABL?* 返回值： *ON*

:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:BAR

描述	设置或查询三相电分析电流谐波测量结果柱状图开关状态
命令格式	:TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:BAR <state> :TPPA:DUT<a>:GROup<b>:MEASure:IHARmonic:BAR?

参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启三相电分析 DUT1 的输入电流谐波测量结果柱状图： *:TPPA:DUT1:GROup1:MEASure:IHARmonic:BAR ON* *TPPA:DUT1:GRO1:MEAS:IHAR:BAR ON* 查询三相电分析 DUT1 的输入电流谐波测量结果柱状图开关状态： *TPPA:DUT1:GRO1:MEAS:IHAR:BAR?* 返回值： *ON*

:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic

描述	设置或查询三相电分析电压谐波测量项开关状态。
命令格式	:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic <state> :TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启三相电分析 DUT1 的输入电压谐波测量项： *:TPPA:DUT1:GROup1:MEASure:VHARmonic ON* *TPPA:DUT1:GRO1:MEAS:VHAR ON* 查询三相电分析 DUT1 的输入电压谐波开关状态： *TPPA:DUT1:GRO1:MEAS:VHAR?* 返回值： *ON*

:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:FREQuency

描述	设置或查询三相电分析电压谐波线路频率。
命令格式	:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:FREQuency <value> :TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:FREQuency?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <value>:= {AUTO_HZ|HZ50|HZ60|HZ400}
返回格式	{AUTO_HZ|HZ50|HZ60|HZ400}
示例	设置三相电分析 DUT1 的输入电压谐波线路频率为 50Hz： *:TPPA:DUT1:GROup1:MEASure:VHARmonic:FREQuency HZ50* *TPPA:DUT1:GRO1:MEAS:VHAR:FREQ HZ50* 查询三相电分析 DUT1 的输入电压谐波线路频率： *TPPA:DUT1:GRO1:MEAS:VHAR:FREQ?* 返回值： *HZ50*

:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:THDStandard

描述	设置或查询三相电分析电压谐波总谐波失真标准类型
命令格式	:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:THDStandard <type> :TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:THDStandard?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <type>:= {IEC|CSA}
返回格式	{IEC|CSA}
示例	设置三相电分析 DUT1 的输入电压谐波的总谐波失真标准为 IEC： *:TPPA:DUT1:GROup1:MEASure:VHARmonic:THDStandard IEC* *TPPA:DUT1:GRO1:MEAS:VHAR:THDS IEC* 查询三相电分析 DUT1 的输入电压谐波的总谐波失真标准： *TPPA:DUT1:GRO1:MEAS:VHAR:THDS?*

	返回值： *IEC*

:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:STANdard

描述	设置或查询三相电分析电压谐波标准类型
命令格式	:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:STANdard <type> :TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:STANdard?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <type>:= {IEC_61000_3_2_A|IEC_61000_3_2_B|IEC_61000_3_2_C|IEC_61000_3_2_D}
返回格式	{IEC_61000_3_2_A|IEC_61000_3_2_B|IEC_61000_3_2_C|IEC_61000_3_2_D}
示例	设置三相电分析 DUT1 的输入电压谐波标准为 IEC_61000_3_2_A： *:TPPA:DUT1:GROup1:MEASure:VHARmonic:STANdard IEC_61000_3_2_A* *TPPA:DUT1:GRO1:MEAS:VHAR:STAN IEC_61000_3_2_A* 查询三相电分析 DUT1 的输入电压谐波标准： *TPPA:DUT1:GRO1:MEAS:VHAR:STAN?* 返回值： *IEC_61000_3_2_A*

:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:TABLe

描述	设置或查询三相电分析电压谐波测量结果表格开关状态
命令格式	:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:TABLe <state> :TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:TABLe?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <state>:= {ON|OFF}
返回格式	{ON|OFF}

示例	设置开启三相电分析 DUT1 的输入电压谐波测量结果表格： *:TPPA:DUT1:GROup1:MEASure:VHARmonic:TABLe ON* *TPPA:DUT1:GRO1:MEAS:VHAR:TABL ON* 查询三相电分析 DUT1 的输入电压谐波测量结果表格开关状态： *TPPA:DUT1:GRO1:MEAS:VHAR:TABL?* 返回值： *ON*

:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:BAR

描述	设置或查询三相电分析电压谐波测量结果柱状图开关状态
命令格式	:TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:BAR <state> :TPPA:DUT<a>:GROup<b>:MEASure:VHARmonic:BAR?
参数说明	<a>:= {1|2}，待测设备序号 <b>:= {1|2} ● 1：输入信号 ● 2：输出信号 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启三相电分析 DUT1 的输入电压谐波测量结果柱状图： *:TPPA:DUT1:GROup1:MEASure:VHARmonic:BAR ON* *TPPA:DUT1:GRO1:MEAS:VHAR:BAR ON* 查询三相电分析 DUT1 的输入电压谐波测量结果柱状图开关状态： *TPPA:DUT1:GRO1:MEAS:VHAR:BAR?* 返回值： *ON*

:TPPA:DUT<a>:MINDuration

描述	设置或查询三相电分析最小分析长度
命令格式	:TPPA:DUT<a>:MINDuration <type> :TPPA:DUT<a>:MINDuration?
参数说明	<a>:= {1|2}，待测设备序号 <type>:= {TIMe[,<time>]|ICCount[,<period>]|OCCount[,<period>]} ● TIMe：时间，需设定时间<time> ● ICCount：输入周期数，需指定周期数<period>

	● OCCount：输出周期数，需指定周期数<period> <time>:= 浮点型 NR3 格式，范围[5E-09,20] <period>:= 整型 NR1 格式，范围[1,40]
返回格式	{TIMeI[,<time>]|ICCount[,<period>]|OCCount[,<period>]}
示例	设置三相电分析 DUT1 的最小分析长度持续时间为 1s： *:TPPA:DUT1:MINDuration TIMe,1.00* *TPPA:DUT1:MIND TIMe,1.00* 查询三相电分析 DUT1 的最小分析长度类型： *TPPA:DUT1:MIND?* 返回值： *TIMe,1.00E+00*

:TPPA:DUT<a>:EFFiciency

描述	设置或查询三相电分析效率分析测量项开关状态
命令格式	:TPPA:DUT<a>:EFFiciency <state> :TPPA:DUT<a>:EFFiciency?
参数说明	<a>:= {1|2}，待测设备序号 <state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置开启三相电分析 DUT1 的效率分析测量项： *:TPPA:DUT1:EFFiciency ON* *TPPA:DUT1: EFF ON* 查询三相电分析 DUT1 的效率分析测量项开关状态： *TPPA:DUT1: EFF?* 返回值： *ON*

:TPPA:AUToset

描述	三相电分析自动设置。根据已开启 DUT 的最小分析长度设置时基，并自动设置通道的垂直参数。
命令格式	:TPPA:AUToset
示例	自动设置示波器三相电分析参数： *TPPA:AUToset* *TPPA:AUT*

5.28 TRIGger 命令系统

:TRIGger 命令子系统控制示波器的触发模式、所有触发类型参数和自动保存参数。

:TRIGger:ACTion:SAVE:DIRectory

描述	设置或查询自动保存的目录。
命令格式	:TRIGger:ACTion:SAVE:DIRectory <path> :TRIGger:ACTion:SAVE:DIRectory?
参数说明	<path>:= 带引号的 ASCII 文本字符串，包含文件存储路径，最后一级目录会自动补全为"Autosave"。可指定存储路径类型如下表： **存储路径类型** | **示例** 内部 | "local" 网络存储器 | "net_storage" U 盘 | "U-disk0" **注意**： ● 保存到内部时，存储路径必须包含 local。 ● 保存到外部时，如未指定存储路径类型，将默认存储到 U-disk0。
返回格式	ASCII 文本字符串，含最后一级"Autosave"
示例	将自动保存目录修改为 U-disk0/SIGLENT/Autosave，路径仅需写为"U-disk0/SIGLENT"： *:TRIGger:ACTion:SAVE:DIRectory "U-disk0/SIGLENT"* *TRIG:ACT:SAVE:DIR "U-disk0/SIGLENT"* 查询自动保存目录： *TRIG:ACT:SAVE:DIR?* 返回值： *U-disk0/SIGLENT/Autosave*
关联命令	:TRIGger:ACTion:SAVE:FILename

:TRIGger:ACTion:SAVE:FILename

描述	设置或查询自动保存文件名
命令格式	:TRIGger:ACTion:SAVE:FILename <name> :TRIGger:ACTion:SAVE:FILename?

参数说明	<name>:= 带引号的 ASCII 文本字符串
返回格式	ASCII 文本字符串
示例	将自动保存文件名修改为 123： *:TRIGger:ACTion:SAVE:FILename "123"* *TRIG:ACT:SAVE:FIL "123"* 查询自动保存文件名： *TRIG:ACT:SAVE:FIL?* 返回值： *123*
关联命令	:TRIGger:ACTion:SAVE:DIRectory

:TRIGger:ACTion:SAVE:WMODe

描述	设置或查询自动保存写入模式
命令格式	:TRIGger:ACTion:SAVE:WMODe <mode> :TRIGger:ACTion:SAVE:WMODe?
参数说明	<mode>:= {WRAP|FILL}
返回格式	{WRAP|FILL}
示例	将自动保存写入模式设置为覆盖： *:TRIGger:ACTion:SAVE:WMODe WRAP* *TRIG:ACT:SAVE:WMOD WRAP* 查询自动保存写入模式： *TRIG:ACT:SAVE:WMOD?* 返回值： *WRAP*

:TRIGger:ACTion:SAVE:WAVeform

描述	设置或查询自动保存波形开关状态
命令格式	:TRIGger:ACTion:SAVE:WAVeform <state> :TRIGger:ACTion:SAVE:WAVeform?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	开启自动保存波形数据： *:TRIGger:ACTion:SAVE:WAVeform ON*

	TRIG:ACT:SAVE:WAV ON 查询自动保存波形数据状态： *TRIG:ACT:SAVE:WAV?* 返回值： *ON*
关联命令	:TRIGger:ACTion:SAVE:WAVeform:FORMat

:TRIGger:ACTion:SAVE:WAVeform:FORMat

描述	设置或问询自动保存波形数据格式
命令格式	:TRIGger:ACTion:SAVE:WAVeform:FORMat <format> :TRIGger:ACTion:SAVE:WAVeform:FORMat?
参数说明	<format>:= {BINary|CSV|MATLab}
返回格式	{BINary|CSV|MATLab}
示例	将自动保存波形数据格式设置为 Binary： *:TRIGger:ACTion:SAVE:WAVeform:FORMat BINary* *TRIG:ACT:SAVE:WAV:FORM BIN* 查询自动保存波形数据格式： *TRIG:ACT:SAVE:WAV:FORM?* 返回值： *BINary*
关联命令	:TRIGger:ACTion:SAVE:WAVeform

:TRIGger:ACTion:SAVE:IMAGe

描述	设置或查询自动保存图片开关状态
命令格式	:TRIGger:ACTion:SAVE:IMAGe <state> :TRIGger:ACTion:SAVE:IMAGe?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	开启自动保存图片： *:TRIGger:ACTion:SAVE:IMAGe ON* *TRIG:ACT:SAVE:IMAG ON* 查询自动保存图片状态： *TRIG:ACT:SAVE:IMAG?*

	返回值： *ON*
关联命令	:TRIGger:ACTion:SAVE:IMAGe:FORMat

:TRIGger:ACTion:SAVE:IMAGe:FORMat

描述	设置或问询自动保存图片格式
命令格式	:TRIGger:ACTion:SAVE:IMAGe:FORMat <format> :TRIGger:ACTion:SAVE:IMAGe:FORMat?
参数说明	<format>:= {BMP|JPG|PNG}
返回格式	{BMP|JPG|PNG}
示例	将自动保存图片格式设置为 PNG： *:TRIGger:ACTion:SAVE:IMAGe:FORMat PNG* *TRIG:ACT:SAVE:IMAG:FORM PNG* 查询自动保存图片格式： *TRIG:ACT:SAVE:IMAG:FORM?* 返回值： *PNG*
关联命令	:TRIGger:ACTion:SAVE:IMAGe

:TRIGger:ACTion:OPERation

描述	设置或问询自动保存运行状态
命令格式	:TRIGger:ACTion:OPERation <state> :TRIGger:ACTion:OPERation?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	开启自动保存： *:TRIGger:ACTion:OPERation ON* *TRIG:ACT:OPER ON* 查询自动保存运行状态： *TRIG:ACT:OPER?* 返回值： *ON*

:TRIGger:FREQuency

描述	查询硬件频率计。如果频率有效，则返回以 Hz 为单位的频率值，默认精度为 3 位，最高精度为 7 位。使用命令":FORMat:DATA"来设置返回数值精度。
命令格式	:TRIGger:FREQuency?
参数说明	<value>:= 浮点型 NR3 格式
返回格式	<value>
示例	查询硬件频率计的值： *:FORMat:DATA CUSTom,7* *TRIG:FREQ?* 返回值： *1.234561E+04*
关联命令	:FORMat:DATA

:TRIGger:MODE

描述	设置或查询触发模式。
命令格式	:TRIGger:MODE <mode> :TRIGger:MODE?
参数说明	<mode>:= {SINGle|NORMal|AUTO|FTRIG} ● AUTO：在自动触发模式中，如果超过定时值仍未找到满足触发条件的波形，示波器将进行强制采集一帧波形数据，在示波器上显示。 ● NORMal：只有满足触发条件时才会进行触发和采集；不满足条件时保持上一次波形显示，等待下一次触发。 ● SINGle：当输入的信号满足触发条件时，示波器即进行捕获并将波形稳定显示在屏幕上。此后，即使再有满足条件的信号，示波器也不予理会。需要进行再次捕获须重新进行单次设置。 ● FTRIG：无论输入的信号是否满足触发条件，强制触发一帧。
返回格式	{SINGle|NORMal|AUTO|FTRIG}
示例	设置示波器触发模式为 SINGle： *:TRIGger:MODE SINGle* *TRIG:MODE SING* 查询示波器触发模式： *TRIG:MODE?* 返回值： *SINGle*

:TRIGger:RUN

描述	设置示波器状态为 Run，并保持当前触发模式。
命令格式	:TRIGger:RUN
示例	如果示波器当前触发模式为 SINGle，且已触发一帧为 Stop 状态，设置示波器状态为 Run 后，将保持为 SINGle 模式重新采集一帧： *:TRIGger:RUN* *TRIG:RUN*
关联命令	:TRIGger:STOP

:TRIGger:STATus

描述	查询示波器当前触发状态。
命令格式	:TRIGger:STATus?
返回格式	<status>:= {Arm|Ready|Auto|Trig'd|Stop|Roll}
示例	查询示波器当前触发状态： *:TRIG:STAT?* 返回值： *Stop*
关联命令	:TRIGger:MODE

:TRIGger:STOP

描述	设置示波器状态为 Stop。此命令等同于前面板上的 Run|Stop 按键的 Stop。
命令格式	:TRIGger:STOP
示例	设置示波器触发状态为 Stop： *:TRIGger:STOP* *TRIG:STOP*
关联命令	:TRIGger:RUN

:TRIGger:TYPE

描述	设置或查询触发类型。
命令格式	:TRIGger:TYPE <type>

	:TRIGger:TYPE?
参数说明	<type>:= {EDGE|PULSE|SLOPe|INTerval|PATTern|RUNT|WINDow|DROPout|VIDeo| QUALified|NEDGe|DELay|SHOLd|IIC|SPI|UART|LIN|CAN|FLEXray|CANFd| IIS|M1553|SENT|A429}
返回格式	{EDGE|PULSE|SLOPe|INTerval|PATTern|RUNT|WINDow|DROPout|VIDeo| QUALified|NEDGe|DELay|SHOLd|IIC|SPI|UART|LIN|CAN|FLEXray|CANFd| IIS|M1553|SENT|A429}
示例	设置触发类型为边沿触发： *:TRIGger:TYPE EDGE* *TRIG:TYPE EDGE* 查询当前触发类型： *TRIG:TYPE?* 返回值： *EDGE*

:TRIGger:EDGE

:TRIGger:EDGE:COUPling

描述	设置或查询边沿触发的耦合方式。
命令格式	:TRIGger:EDGE:COUPling <mode> :TRIGger:EDGE:COUPling?
参数说明	<mode>:= {DC|AC|LFREJect|HFREJect} ● DC：直流耦合。通过信号的所有分量。 ● AC：交流耦合。抑制信号的直流分量，截止频率详见数据手册。 ● LFREJect：低频抑制。相当于高通滤波器，截止频率详见数据手册。 ● HFREJect：高频抑制。相当于低通滤波器，截止频率详见数据手册。
返回格式	{DC|AC|LFREJect|HFREJect}
示例	设置边沿触发的耦合方式为直流耦合： *:TRIGger:EDGE:COUPling DC* *TRIG:EDGE:COUP DC* 查询当前边沿触发的耦合方式： *TRIG:EDGE:COUP?* 返回值： *DC*

:TRIGger:EDGE:HLDEVent

描述	设置或查询边沿触发的触发释抑事件数。
命令格式	:TRIGger:EDGE:HLDEVent <value> :TRIGger:EDGE:HLDEVent?
参数说明	<value>:= 整型 NR1 格式，该值范围为 [1,100000000]
返回格式	整型 NR1 格式
示例	设置边沿触发的触发释抑事件数为 3： *:TRIGger:EDGE:HLDEVent 3* *TRIG:EDGE:HLDEV 3* 查询当前边沿触发的触发释抑事件数： *TRIG:EDGE:HLDEV?* 返回值： *3*
关联命令	:TRIGger:EDGE:HOLDoff

:TRIGger:EDGE:HLDTime

描述	设置或查询边沿触发的触发释抑时间。
命令格式	:TRIGger:EDGE:HLDTime <value> :TRIGger:EDGE:HLDTime?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS6000 Pro SDS6000A SDS6000L SDS5000X HD SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[8.00E-09,3.00E+01]
SHS800X SHS1000X	[80.00E-09,1.5E+00]

返回格式	浮点型 NR3 格式

示例	设置边沿触发的触发释抑时间为 15ns： *:TRIGger:EDGE:HLDTime 1.50E-08* *TRIG:EDGE:HLDT 1.50E-08* 查询当前边沿触发的触发释抑时间： *TRIG:EDGE:HLDT?* 返回值： *1.50E-08*
关联命令	:TRIGger:EDGE:HOLDoff

:TRIGger:EDGE:HOLDoff

描述	设置或查询边沿触发的触发释抑类型。
命令格式	:TRIGger:EDGE:HOLDoff <holdoff_type> :TRIGger:EDGE:HOLDoff?
参数说明	<holdoff_type>:= {OFF|EVENts|TIME} ● OFF：关闭触发释抑 ● EVENts：事件。满足触发条件的次数 ● TIME：时间。指从触发之后到下一次重新启用触发电路之前示波器等待的时间
返回格式	{OFF|EVENts|TIME}
示例	设置边沿触发的触发释抑关闭： *:TRIGger:EDGE:HOLDoff OFF* *TRIG:EDGE:HOLD OFF* 查询当前边沿触发的触发释抑类型： *TRIG:EDGE:HOLD?* 返回值： *OFF*
关联命令	:TRIGger:EDGE:HLDEVent :TRIGger:EDGE:HLDTime :TRIGger:EDGE:HSTart

:TRIGger:EDGE:HSTart

描述	设置或查询边沿触发的触发释抑启动条件。
命令格式	:TRIGger:EDGE:HSTart <start_holdoff> :TRIGger:EDGE:HSTart?
参数说明	<start_holdoff>:= {LAST_TRIG|ACQ_START}

	● LAST_TRIG：上次触发。开始计算触发释抑的位置为上次的触发时间点。 ● ACQ_START：采集开始。开始计算触发释抑的位置为第一个满足触发条件的时间。
返回格式	{LAST_TRIG|ACQ_START}
示例	设置边沿触发的触发释抑启动条件为上次触发： *:TRIGger:EDGE:HSTart LAST_TRIG* *TRIG:EDGE:HST LAST_TRIG* 查询边沿触发的触发释抑启动条件： *TRIG:EDGE:HST?* 返回值： *LAST_TRIG*
关联命令	:TRIGger:EDGE:HOLDoff

:TRIGger:EDGE:IMPedance

描述	设置或查询边沿触发的触发源阻抗，该命令仅在触发源为 EXT 或 EXT/5 时可用。
命令格式	:TRIGger:EDGE:IMPedance <ohm> :TRIGger:EDGE:IMPedance?
参数说明	<ohm>:= {ONEMeg|FIFTy}
返回格式	{ONEMeg|FIFTy}
示例	设置边沿触发的触发源阻抗为 50ohm： *:TRIGger:EDGE:IMPedance FIFTy* *TRIG:EDGE:IMP FIFT* 查询边沿触发的触发源阻抗： *TRIG:EDGE:IMP?* 返回值： *FIFTy*
关联命令	:TRIGger:EDGE:SOURce

:TRIGger:EDGE:LEVel

描述	设置或查询边沿触发的触发电平。
命令格式	:TRIGger:EDGE:LEVel <level_value> :TRIGger:EDGE:LEVel?
参数说明	<level_value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见

	下表： **机型** / **范围** SDS7000A SDS5000X HD / [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X / [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD / [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置边沿触发的触发电平为 0.5V： *:TRIGger:EDGE:LEVel 5.00E-01* *TRIG:EDGE:LEV 5.00E-01* 查询当前边沿触发的触发电平： *TRIG:EDGE:LEV?* 返回值： *5.00E-01*
关联命令	:TRIGger:EDGE:SOURce

:TRIGger:EDGE:NREJect

描述	设置或查询边沿触发的噪声抑制开关状态。
命令格式	:TRIGger:EDGE:NREJect <state> :TRIGger:EDGE:NREJect?
参数说明	<state>:= {OFF|ON}
返回格式	{OFF|ON}
示例	设置边沿触发的噪声抑制状态为开启： *:TRIGger:EDGE:NREJect ON* *TRIG:EDGE:NREJ ON* 查询当前边沿触发的噪声抑制状态： *TRIG:EDGE:NREJ?*

	返回值： *ON*
关联命令	:TRIGger:EDGE:SOURce

:TRIGger:EDGE:SLOPe

描述	设置或查询边沿触发的斜率类型。
命令格式	:TRIGger:EDGE:SLOPe <slope_type> :TRIGger:EDGE:SLOPe?
参数说明	<slope_type>:= {RISing|FALLing|ALTernate}
返回格式	{RISing|FALLing|ALTernate}
示例	设置边沿触发的斜率类型为上升沿： *:TRIGger:EDGE:SLOPe RISing* *TRIG:EDGE:SLOP RIS* 查询当前边沿触发的斜率类型： *TRIG:EDGE:SLOP?* 返回值： *RISing*

:TRIGger:EDGE:SOURce

描述	设置或查询边沿触发的触发源。
命令格式	:TRIGger:EDGE:SOURce <source> :TRIGger:EDGE:SOURce?
参数说明	<source>:= {C<n>|D<d>|EX|EX5|LINE} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>|EX|EX5|LINE}
示例	设置边沿触发的触发源为 C1： *:TRIGger:EDGE:SOURce C1* *TRIG:EDGE:SOUR C1* 查询当前边沿触发的触发源： *TRIG:EDGE:SOUR?* 返回值： *C1*
关联命令	:TRIGger:EDGE:LEVel

:TRIGger:SLOPe

:TRIGger:SLOPe:COUPling

描述	设置或查询斜率触发的耦合方式。
命令格式	:TRIGger:SLOPe:COUPling <mode> :TRIGger:SLOPe:COUPling?
参数说明	<mode>:= {DC|AC|LFREJect|HFREJect} ● DC：直流耦合。通过信号的所有分量。 ● AC：交流耦合。抑制信号的直流分量，截止频率详见数据手册。 ● LFREJect：低频抑制。相当于高通滤波器，截止频率详见数据手册。 ● HFREJect：高频抑制。相当于低通滤波器，截止频率详见数据手册。
返回格式	{DC|AC|LFREJect|HFREJect}
示例	设置斜率触发的耦合方式为直流耦合： *:TRIGger:SLOPe:COUPling DC* *TRIG:SLOP:COUP DC* 查询当前斜率触发的耦合方式： *TRIG:SLOP:COUP?* 返回值： *DC*

:TRIGger:SLOPe:HLDEVent

描述	设置或查询斜率触发的触发释抑事件数。
命令格式	:TRIGger:SLOPe:HLDEVent <value> :TRIGger:SLOPe:HLDEVent?
参数说明	<value>:= 整型 NR1 格式，该值范围为 [1,100000000]
返回格式	整型 NR1 格式
示例	设置斜率触发的触发释抑事件数为 3： *:TRIGger:SLOPe:HLDEVent 3* *TRIG:SLOP:HLDEV 3* 查询当前斜率触发的触发释抑事件数： *TRIG:SLOP:HLDEV?* 返回值： *3*
关联命令	:TRIGger:SLOPe:HOLDoff

:TRIGger:SLOPe:HLDTime

描述	设置或查询斜率触发的触发释抑时间。
命令格式	:TRIGger:SLOPe:HLDTime <value> :TRIGger:SLOPe:HLDTime?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS6000 Pro SDS6000A SDS6000L SDS5000X HD SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD [8.00E-09,3.00E+01]
返回格式	浮点型 NR3 格式
示例	设置斜率触发的触发释抑时间为 15ns： *:TRIGger:SLOPe:HLDTime 1.50E-08* *TRIG:SLOP:HLDT 1.50E-08* 查询当前斜率触发的触发释抑时间： *TRIG:SLOP:HLDT?* 返回值： *1.50E-08*
关联命令	:TRIGger:SLOPe:HOLDoff

:TRIGger:SLOPe:HLEVel

描述	设置或查询斜率触发的触发高电平。
命令格式	:TRIGger:SLOPe:HLEVel <high_level_value> :TRIGger:SLOPe:HLEVel?
参数说明	<high_level_value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]

SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

注意：高电平不能小于低电平（低电平设置指令:TRIGger:SLOPe:LLEVel）。

返回格式	浮点型 NR3 格式
示例	设置斜率触发的触发高电平为 0.5V： *:TRIGger:SLOPe:HLEVel 5.00E-01* *TRIG:SLOP:HLEV 5.00E-01* 查询当前斜率触发的触发高电平： *TRIG:SLOP:HLEV?* 返回值： *5.00E-01*
关联命令	:TRIGger:SLOPe:LLEVel

:TRIGger:SLOPe:HOLDoff

描述	设置或查询斜率触发的触发释抑类型。
命令格式	:TRIGger:SLOPe:HOLDoff <holdoff_type> :TRIGger:SLOPe:HOLDoff?
参数说明	<holdoff_type>:= {OFF|EVENts|TIME} ● OFF：关闭触发释抑 ● EVENts：事件。满足触发条件的次数 ● TIME：时间。指从触发之后到下一次重新启用触发电路之前示波器等待的时间
返回格式	{OFF|EVENts|TIME}
示例	设置斜率触发的触发释抑类型为关闭： *:TRIGger:SLOPe:HOLDoff OFF* *TRIG:SLOP:HOLD OFF* 查询当前斜率触发的触发释抑类型：

	TRIG:SLOP:HOLD? 返回值： *OFF*
关联命令	:TRIGger:SLOPe:HLDEVent :TRIGger:SLOPe:HLDTime :TRIGger:SLOPe:HSTart

:TRIGger:SLOPe:HSTart

描述	设置或查询斜率触发的触发释抑启动条件。
命令格式	:TRIGger:SLOPe:HSTart <start_holdoff> :TRIGger:SLOPe:HSTart?
参数说明	<start_holdoff>:= {LAST_TRIG|ACQ_START} ● LAST_TRIG：上次触发。开始计算触发释抑的位置为上次的触发时间点。 ● ACQ_START：采集开始。开始计算触发释抑的位置为第一个满足触发条件的时间。
返回格式	{LAST_TRIG|ACQ_START}
示例	设置斜率触发的触发释抑启动条件为上次触发： *:TRIGger:SLOPe:HSTart LAST_TRIG* *TRIG:SLOP:HST LAST_TRIG* 查询斜率触发的触发释抑启动条件： *TRIG:SLOP:HST?* 返回值： *LAST_TRIG*
关联命令	:TRIGger:SLOPe:HOLDoff

:TRIGger:SLOPe:LIMit

描述	设置或查询斜率触发的限制条件类型。
命令格式	:TRIGger:SLOPe:LIMit <type> :TRIGger:SLOPe:LIMit?
参数说明	<type>:= {LESSthan|GREATerthan|INNer|OUTer} ● LESSthan（<=）：小于时间值。 ● GREATerthan（>=）：大于时间值。 ● INNer（[--,--]）：时间值范围内。 ● OUTer（--] [--）：时间值范围外。

返回格式	{LESSthan|GREATerthan|INNer|OUTer}
示例	设置斜率触发的限制条件类型为小于时间值： *:TRIGger:SLOPe:LIMit LESSthan* *TRIG:SLOP:LIM LESS* 查询当前斜率触发的限制条件类型： *TRIG:SLOP:LIM?* 返回值： *LESSthan*
关联命令	:TRIGger:SLOPe:TLOWer :TRIGger:SLOPe:TUPPer

:TRIGger:SLOPe:LLEVel

描述	设置或查询斜率触发的触发低电平。
命令格式	:TRIGger:SLOPe:LLEVel <low_level_value> :TRIGger:SLOPe:LLEVel?
参数说明	<low_level_value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

注意：低电平不能大于高电平（高电平设置指令:TRIGger:SLOPe:HLEVel）。

返回格式	浮点型 NR3 格式
示例	设置斜率触发的触发低电平为-0.5V： *:TRIGger:SLOPe:LLEVel -5.00E-01*

	TRIG:SLOP:LLEV -5.00E-01 查询当前斜率触发的触发低电平： *TRIG:SLOP:LLEV?* 返回值： *-5.00E-01*
关联命令	:TRIGger:SLOPe:HLEVel

:TRIGger:SLOPe:NREJect

描述	设置或查询斜率触发的噪声抑制开关状态。
命令格式	:TRIGger:SLOPe:NREJect <state> :TRIGger:SLOPe:NREJect?
参数说明	<state>:= {OFF|ON}
返回格式	{OFF|ON}
示例	设置斜率触发的噪声抑制状态为开启： *:TRIGger:SLOPe:NREJect ON* *TRIG:SLOP:NREJ ON* 查询当前斜率触发的噪声抑制状态： *TRIG:SLOP:NREJ?* 返回值： *ON*

:TRIGger:SLOPe:SLOPe

描述	设置或查询斜率触发的斜率类型。
命令格式	:TRIGger:SLOPe:SLOPe <slope_type> :TRIGger:SLOPe:SLOPe?
参数说明	<slope_type>:= {RISing|FALLing}
返回格式	{RISing|FALLing}
示例	设置斜率触发的斜率类型为上升沿： *:TRIGger:SLOPe:SLOPe RISing* *TRIG:SLOP:SLOP RIS* 查询当前斜率触发的斜率类型： *TRIG:SLOP:SLOP?* 返回值： *RISing*

:TRIGger:SLOPe:SOURce

描述	设置或查询斜率触发的触发源。
命令格式	:TRIGger:SLOPe:SOURce <source> :TRIGger:SLOPe:SOURce?
参数说明	<source>:= {C<n>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式
返回格式	{C<n>}
示例	设置斜率触发的触发源为 C2： *:TRIGger:SLOPe:SOURce C2* *TRIG:SLOP:SOUR C2* 查询当前斜率触发的触发源： *TRIG:SLOP:SOUR?* 返回值： *C2*

:TRIGger:SLOPe:TLOWer

描述	设置或查询斜率触发限制条件的下限值。
命令格式	:TRIGger:SLOPe:TLOWer <value> :TRIGger:SLOPe:TLOWer?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A	[1.00E-09,2.00E+01]
SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-2.00E-09,2.00E+01]
SHS800X SHS1000X	[2.00E-09,4.20E+00]

注意：

- 下限值不能大于上限值（上限值设置指令:TRIGger:SLOPe:TUPPer）。

	● 当限制类型为“<=”时，此命令不可用。
返回格式	浮点型 NR3 格式
示例	设置斜率触发的下限值为 10ns： *:TRIGger:SLOPe:TLOWer 1.00E-08* *TRIG:SLOP:TLOW 1.00E-08* 查询当前斜率触发的下限值： *TRIG:SLOP:TLOW?* 返回值： *1.00E-08*
关联命令	:TRIGger:SLOPe:LIMit :TRIGger:SLOPe:TUPPer

:TRIGger:SLOPe:TUPPer

描述	设置或查询斜率触发限制条件的上限值。
命令格式	:TRIGger:SLOPe:TUPPer <value> :TRIGger:SLOPe:TUPPer?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A	[1.00E-09,2.00E+01]
SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-2.00E-09,2.00E+01]
SHS800X SHS1000X	[2.00E-09,4.20E+00]

	注意： ● 上限值不能小于下限值（下限值设置指令:TRIGger:SLOPe:TLOWer）。 ● 当限制类型为“>=”时，此命令不可用。
返回格式	浮点型 NR3 格式
示例	设置斜率触发限制条件的上限值为 30ns： *:TRIGger:SLOPe:TUPPer 3.00E-08*

	TRIG:SLOP:TUPP 3.00E-08 查询当前斜率触发限制条件的上限值： *TRIG:SLOP:TUPP?* 返回值： *3.00E-08*
关联命令	:TRIGger:SLOPe:LIMit :TRIGger:SLOPe:TLOWer

:TRIGger:PULSe

:TRIGger:PULSe:COUPling

描述	设置或查询脉宽触发的耦合方式。
命令格式	:TRIGger:PULSe:COUPling <mode> :TRIGger:PULSe:COUPling?
参数说明	<mode>:= {DC|AC|LFREJect|HFREJect} ● DC：直流耦合。通过信号的所有分量。 ● AC：交流耦合。抑制信号的直流分量，截止频率详见数据手册。 ● LFREJect：低频抑制。相当于高通滤波器，截止频率详见数据手册。 ● HFREJect：高频抑制。相当于低通滤波器，截止频率详见数据手册。
返回格式	{DC|AC|LFREJect|HFREJect}
示例	设置脉宽触发的耦合方式为直流耦合： *:TRIGger:PULSe:COUPling DC* *TRIG:PULS:COUP DC* 查询脉宽触发的耦合方式： *TRIG:PULS:COUP?* 返回值： *DC*

:TRIGger:PULSe:HLDEVent

描述	设置或查询脉宽触发的触发释抑事件数。
命令格式	:TRIGger:PULSe:HLDEVent <value> :TRIGger:PULSe:HLDEVent?
参数说明	<value>:= 整型 NR1 格式，该值范围为 [1,100000000]
返回格式	整型 NR1 格式
示例	设置脉宽触发的触发释抑事件数为 3： *:TRIGger:PULSe:HLDEVent 3*

	TRIG:PULS:HLDEV 3 查询当前脉宽触发的触发释抑事件数： *TRIG:PULS:HLDEV?* 返回值： *3*
关联命令	:TRIGger:PULSe:HOLDoff

:TRIGger:PULSe:HLDTime

描述	设置或查询脉宽触发的触发释抑时间。
命令格式	:TRIGger:PULSe:HLDTime <value> :TRIGger:PULSe:HLDTime?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[8.00E-09,3.00E+01]

返回格式	浮点型 NR3 格式
示例	设置脉宽触发的触发释抑时间为 15ns： *:TRIGger:PULSe:HLDTime 1.50E-08* *TRIG:PULS:HLDT 1.50E-08* 查询当前脉宽触发的触发释抑时间： *TRIG:PULS:HLDT?* 返回值： *1.50E-08*
关联命令	:TRIGger:PULSe:HOLDoff

:TRIGger:PULSe:HOLDoff

描述	设置或查询脉宽触发的触发释抑类型。

命令格式	:TRIGger:PULSe:HOLDoff <holdoff_type> :TRIGger:PULSe:HOLDoff?
参数说明	<holdoff_type>:= {OFF|EVENts|TIME} ● OFF：关闭触发释抑 ● EVENts：事件。满足触发条件的次数 ● TIME：时间。指从触发之后到下一次重新启用触发电路之前示波器等待的时间
返回格式	{OFF|EVENts|TIME}
示例	设置脉宽触发的触发释抑类型为关闭： *:TRIGger:PULSe:HOLDoff OFF* *TRIG:PULS:HOLD OFF* 查询当前脉宽触发的触发释抑类型： *TRIG:PULS:HOLD?* 返回值： *OFF*
关联命令	:TRIGger:PULSe:HLDEVent :TRIGger:PULSe:HLDTime :TRIGger:PULSe:HSTart

:TRIGger:PULSe:HSTart

描述	设置或查询脉宽触发的触发释抑启动条件。
命令格式	:TRIGger:PULSe:HSTart <start_holdoff> :TRIGger:PULSe:HSTart?
参数说明	<start_holdoff>:= {LAST_TRIG|ACQ_START} ● LAST_TRIG：上次触发。开始计算触发释抑的位置为上次的触发时间点。 ● ACQ_START：采集开始。开始计算触发释抑的位置为第一个满足触发条件的时间。
返回格式	{LAST_TRIG|ACQ_START}
示例	设置脉宽触发的触发释抑启动条件为上次触发： *:TRIGger:PULSe:HSTart LAST_TRIG* *TRIG:PULS:HST LAST_TRIG* 查询脉宽触发的触发释抑启动条件： *TRIG:PULS:HST?* 返回值： *LAST_TRIG*
关联命令	:TRIGger:PULSe:HOLDoff

:TRIGger:PULSe:LEVel

描述	设置或查询脉宽触发的触发电平。
命令格式	:TRIGger:PULSe:LEVel <level_value> :TRIGger:PULSe:LEVel?
参数说明	<level_value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD — [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X — [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD — [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置脉宽触发的触发电平为 0.5V： *:TRIGger:PULSe:LEVel 5.00E-01* *TRIG:PULS:LEV 5.00E-01* 查询当前脉宽触发的触发电平： *TRIG:PULS:LEV?* 返回值： *5.00E-01*
关联命令	:TRIGger:PULSe:SOURce

:TRIGger:PULSe:LIMit

描述	设置或查询脉宽触发的限制条件类型。
命令格式	:TRIGger:PULSe:LIMit <type> :TRIGger:PULSe:LIMit?

参数说明	<type>:= {LESSthan|GREATerthan|INNer|OUTer} ● LESSthan（<=）：小于时间值。 ● GREATerthan（>=）：大于时间值。 ● INNer（[--,--]）：时间值范围内。 ● OUTer（--] [--）：时间值范围外。
返回格式	{LESSthan|GREATerthan|INNer|OUTer}
示例	设置脉宽触发的限制条件类型为时间值范围内： *:TRIGger:PULSe:LIMit INNer* *TRIG:PULS:LIM INN* 查询当前脉宽触发的限制条件类型： *TRIG:PULS:LIM?* 返回值： *INNer*
关联命令	:TRIGger:PULSe:TLOWer :TRIGger:PULSe:TUPPer

:TRIGger:PULSe:NREJect

描述	设置或查询脉宽触发的噪声抑制开关状态。
命令格式	:TRIGger:PULSe:NREJect <state> :TRIGger:PULSe:NREJect?
参数说明	<state>:= {OFF|ON}
返回格式	{OFF|ON}
示例	设置脉宽触发的噪声抑制状态为开启： *:TRIGger:PULSe:NREJect ON* *TRIG:PULS:NREJ ON* 查询当前脉宽触发的噪声抑制状态： *TRIG:PULS:NREJ?* 返回值： *ON*

:TRIGger:PULSe:POLarity

描述	设置或查询脉宽触发的极性。
命令格式	:TRIGger:PULSe:POLarity <polarity_type> :TRIGger:PULSe:POLarity?
参数说明	<polarity_type>:= {POSitive|NEGative}

返回格式	{POSitive|NEGative}
示例	设置脉宽触发的极性为正脉冲： *:TRIGger:PULSe:POLarity POSitive* *TRIG:PULS:POL POS* 查询当前脉宽触发的极性： *TRIG:PULS:POL?* 返回值： *POSitive*

:TRIGger:PULSe:SOURce

描述	设置或查询脉宽触发的触发源。
命令格式	:TRIGger:PULSe:SOURce <source> :TRIGger:PULSe:SOURce?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置脉宽触发的触发源为 C2： *:TRIGger:PULSe:SOURce C2* *TRIG:PULS:SOUR C2* 查询当前脉宽触发的触发源： *TRIG:PULS:SOUR?* 返回值： *C2*

:TRIGger:PULSe:TLOWer

描述	设置或查询脉宽触发限制条件的下限值。
命令格式	:TRIGger:PULSe:TLOWer <value> :TRIGger:PULSe:TLOWer?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A	[1.00E-09,2.00E+01]
SDS6000 Pro SDS6000A SDS6000L	[-2.00E-09,2.00E+01]

SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	
SHS800X SHS1000X	[2.00E-09,4.20E+00]

注意:

- 下限值不能大于上限值（上限值设置指令:TRIGger:PULSe:TUPPer）。
- 当限制类型为“<=”时，此命令不可用。

返回格式	浮点型 NR3 格式
示例	设置脉宽触发的下限值为 10ns: *:TRIGger:PULSe:TLOWer 1.00E-08* *TRIG:PULS:TLOW 1.00E-08* 查询当前脉宽触发的下限值: *TRIG:PULS:TLOW?* 返回值: *1.00E-08*
关联命令	:TRIGger:PULSe:LIMit :TRIGger:PULSe:TUPPer

:TRIGger:PULSe:TUPPer

描述	设置或查询脉宽触发限制条件的上限值。
命令格式	:TRIGger:PULSe:TUPPer <value> :TRIGger:PULSe:TUPPer?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表:

机型	范围
SDS7000A	[1.00E-09,2.00E+01]
SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD	[-2.00E-09,2.00E+01]

SDS1000X HD SDS800X HD	
SHS800X SHS1000X	[2.00E-09,4.20E+00]

注意:

- 上限值不能小于下限值（下限值设置指令:TRIGger:PULSe:TLOWer）。
- 当限制类型为“>=”时，此命令不可用。

返回格式	浮点型 NR3 格式
示例	设置脉宽触发的限制条件上限值为 30ns: *:TRIGger:PULSe:TUPPer 3.00E-08* *TRIG:PULS:TUPP 3.00E-08* 查询当前脉宽触发的限制条件上限值: *TRIG:PULS:TUPP?* 返回值: *3.00E-08*
关联命令	:TRIGger:PULSe:LIMit :TRIGger:PULSe:TLOWer

:TRIGger:VIDeo

:TRIGger:VIDeo:FCNT

描述	设置或查询视频触发的自定义标准场数。
命令格式	:TRIGger:VIDeo:FCNT <field_cnt> :TRIGger:VIDeo:FCNT?
参数说明	<field_cnt>:= {1|2|4|8}
返回格式	{1|2|4|8}
示例	设置视频触发的自定义标准场数为 8: *:TRIGger:VIDeo:FCNT 8* *TRIG:VID:FCNT 8* 查询当前视频触发的自定义标准场数: *TRIG:VID:FCNT?* 返回值: *8*
关联命令	:TRIGger:VIDeo:STANdard

:TRIGger:VIDeo:FIELd

描述	设置或查询视频触发的同步触发场数，仅标准类型为 NTSC、PAL、1080i/50 或 1080i/60 时有效。
命令格式	:TRIGger:VIDeo:FIELd <field> :TRIGger:VIDeo:FIELd?
参数说明	<field>:= {1|2}
返回格式	{1|2}
示例	设置视频触发的同步触发场数为 2： *:TRIGger:VIDeo:FIELd 2* *TRIG:VID:FIEL 2* 查询当前视频触发的同步触发场数： *TRIG:VID:FIEL?* 返回值： *2*
关联命令	:TRIGger:VIDeo:STANdard :TRIGger:VIDeo:SYNC

:TRIGger:VIDeo:FRATe

描述	设置或查询视频触发自定义标准下的帧速率。
命令格式	:TRIGger:VIDeo:FRATe <frate> :TRIGger:VIDeo:FRATe?
参数说明	<field>:= {25Hz|30Hz|50Hz|60Hz}
返回格式	{25Hz|30Hz|50Hz|60Hz}
示例	设置视频触发自定义标准下的帧速率为 50Hz： *:TRIGger:VIDeo:FRATe 50Hz* *TRIG:VID:FRAT 50Hz* 查询当前视频触发自定义标准下的帧速率： *TRIG:VID:FRAT?* 返回值： *50Hz*
关联命令	:TRIGger:VIDeo:STANdard

:TRIGger:VIDeo:INTerlace

描述	设置或查询视频触发自定义标准下的交错比例。
命令格式	:TRIGger:VIDeo:INTerlace <interlace> :TRIGger:VIDeo:INTerlace?
参数说明	<field>:= {1|2|4|8}
返回格式	{1|2|4|8}
示例	设置视频触发自定义标准下的交错比例为 8:1： *:TRIGger:VIDeo:INTerlace 8* *TRIG:VID:INT 8* 查询当前视频触发自定义标准下的交错比例： *TRIG:VID:INT?* 返回值： *8*
关联命令	:TRIGger:VIDeo:STANdard

:TRIGger:VIDeo:LCNT

描述	设置或查询视频触发自定义标准下的行数。
命令格式	:TRIGger:VIDeo:LCNT <line_cnt> :TRIGger:VIDeo:LCNT?
参数说明	<line_cnt>:= 整型 NR1 格式，该值范围为 [300,2000]。 如果行数设置为 800，那么交错比例、场数、触发于行数、触发于场数的正确关系如下表：

选择行数	交错	选择场数	触发于行	触发于场
800	1:1	1	800	1
800	2:1	1/2/4/8	400	1/1~2/1~4/1~8
800	4:1	1/2/4/8	300	1/1~2/1~4/1~8
800	8:1	1/2/4/8	100	1/1~2/1~4/1~8

返回格式	整型 NR1 格式
示例	设置视频触发自定义标准下的行数为 500： *:TRIGger:VIDeo:LCNT 500* *TRIG:VID:LCNT 500* 查询当前视频触发自定义标准下的行数： *TRIG:VID:LCNT?* 返回值： *500*

关联命令	:TRIGger:VIDeo:STANdard

:TRIGger:VIDeo:LEVel

描述	设置或查询视频触发的触发电平。
命令格式	:TRIGger:VIDeo:LEVel <level_value> :TRIGger:VIDeo:LEVel?
参数说明	<level_value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD / [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X / [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD / [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置视频触发的触发电平为 0.5V： *:TRIGger:VIDeo:LEVel 5.00E-01* *TRIG:VID:LEV 5.00E-01* 查询当前视频触发的触发电平： *TRIG:VID:LEV?* 返回值： *5.00E-01*

:TRIGger:VIDeo:LINE

描述	设置或查询视频触发非自定义标准下的同步触发行数。
命令格式	:TRIGger:VIDeo:LINE <line> :TRIGger:VIDeo:LINE?

参数说明	<line>:= 整型 NR1 格式 下表展示了所有视频标准（自定义除外）的行和字段之间的对应关系： **视频标准** | **场 1** | **场 2** NTSC | [1, 263] | [1, 262] PAL | [1, 313] | [1, 312] HDTV 720P/50, 720P/60 | [1, 750] | HDTV 1080P/50, 1080P/60 | [1, 1125] | HDTV 1080i/50, 1080i/60 | [1, 563] | [1, 562]
返回格式	整型 NR1 格式
示例	设置视频触发非自定义标准下的同步触发行数为 2： *:TRIGger:VIDeo:LINE 2* *TRIG:VID:LINE 2* 查询当前视频触发非自定义标准下的同步触发行数： *TRIG:VID:LINE?* 返回值： *2*
关联命令	:TRIGger:VIDeo:STANdard :TRIGger:VIDeo:SYNC

:TRIGger:VIDeo:SOURce

描述	设置或查询视频触发的触发源。
命令格式	:TRIGger:VIDeo:SOURce <source> :TRIGger:VIDeo:SOURce?
参数说明	<source>:= {C<n>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式
返回格式	{C<n>}
示例	设置视频触发的触发源为 C2： *:TRIGger:VIDeo:SOURce C2* *TRIG:VID:SOUR C2* 查询当前视频触发的触发源： *TRIG:VID:SOUR?* 返回值： *C2*

:TRIGger:VIDeo:STANdard

描述	设置或查询视频触发的标准类型。
命令格式	:TRIGger:VIDeo:STANdard <standard> :TRIGger:VIDeo:STANdard?
参数说明	<standard>:={NTSC|PAL|P720L50|P720L60|P1080L50|P1080L60|I1080L50|I1080L60|CUSTom}
返回格式	{NTSC|PAL|P720L50|P720L60|P1080L50|P1080L60|I1080L50|I1080L60|CUSTom}
示例	设置视频触发的标准类型为 NTSC： *:TRIGger:VIDeo:STANdard NTSC* *TRIG:VID:STAN NTSC* 查询当前视频触发的标准类型： *TRIG:VID:STAN?* 返回值： *NTSC*

:TRIGger:VIDeo:SYNC

描述	设置或查询视频触发的同步模式。
命令格式	:TRIGger:VIDeo:SYNC <sync> :TRIGger:VIDeo:SYNC?
参数说明	<sync>:= {SELect|ANY} ● SELect：选择。可选择指定场和指定行对信号进行触发 ● ANY：任意。视频可在满足条件的任意行上进行触发
返回格式	{SELect|ANY}
示例	设置视频触发的同步类型为选择： *:TRIGger:VIDeo:SYNC SELect* *TRIG:VID:SYNC SEL* 查询当前视频触发的同步类型： *TRIG:VID:SYNC?* 返回值： *SELect*
关联命令	:TRIGger:VIDeo:STANdard :TRIGger:VIDeo:LINE :TRIGger:VIDeo:FIELd

:TRIGger:WINDow

:TRIGger:WINDow:CLEVel

描述	设置或查询窗口触发的相对中心电平。
命令格式	:TRIGger:WINDow:CLEVel <value> :TRIGger:WINDow:CLEVel?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD — [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X — [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD — [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置窗口触发的相对中心电平为 0.5V： *:TRIGger:WINDow:CLEVel 5.00E-01* *TRIG:WIND:CLEV 5.00E-01* 查询当前窗口触发的中心电平： *TRIG:WIND:CLEV?* 返回值： *5.00E-01*
关联命令	:TRIGger:WINDow:DLEVel

:TRIGger:WINDow:COUPling

描述	设置或查询窗口触发的耦合方式。
命令格式	:TRIGger:WINDow:COUPling <mode> :TRIGger:WINDow:COUPling?
参数说明	<mode>:= {DC|AC|LFREJect|HFREJect} ● DC：直流耦合。通过信号的所有分量。 ● AC：交流耦合。抑制信号的直流分量，截止频率详见数据手册。 ● LFREJect：低频抑制。相当于高通滤波器，截止频率详见数据手册。 ● HFREJect：高频抑制。相当于低通滤波器，截止频率详见数据手册。
返回格式	{DC|AC|LFREJect|HFREJect}
示例	设置窗口触发的耦合方式为直流耦合： *:TRIGger:WINDow:COUPling DC* *TRIG:WIND:COUP DC* 查询当前窗口触发的耦合方式： *TRIG:WIND:COUP?* 返回值： *DC*

:TRIGger:WINDow:DLEVel

描述	设置或查询窗口触发的相对电平范围。
命令格式	:TRIGger:WINDow:DLEVel <value> :TRIGger:WINDow:DLEVel?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置窗口触发的相对电平范围为 0.5V： *:TRIGger:WINDow:DLEVel 5.00E-01* *TRIG:WIND:DLEV 5.00E-01* 查询当前窗口触发的中心电平： *TRIG:WIND:DLEV?* 返回值： *5.00E-01*
关联命令	:TRIGger:WINDow:CLEVel

:TRIGger:WINDow:HLDEVent

描述	设置或查询窗口触发的触发释抑事件数。
命令格式	:TRIGger:WINDow:HLDEVent <value> :TRIGger:WINDow:HLDEVent?
参数说明	<value>:= 整型 NR1 格式，该值范围为 [1,100000000]
返回格式	整型 NR1 格式
示例	设置窗口触发的触发释抑事件数为 3： *:TRIGger:WINDow:HLDEVent 3* *TRIG:WIND:HLDEV 3* 查询当前窗口触发的触发释抑事件数： *TRIG:WIND:HLDEV?* 返回值： *3*
关联命令	:TRIGger:WINDow:HOLDoff

:TRIGger:WINDow:HLDTime

描述	设置或查询窗口触发的触发释抑时间。
命令格式	:TRIGger:WINDow:HLDTime <value> :TRIGger:WINDow:HLDTime?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS6000 Pro SDS6000A SDS6000L	[8.00E-09,3.00E+01]

	SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	
返回格式	浮点型 NR3 格式	
示例	设置窗口触发的触发释抑时间为 15ns： *:TRIGger:WINDow:HLDTime 1.50E-08* *TRIG:WIND:HLDT 1.50E-08* 查询当前窗口触发的触发释抑时间： *TRIG:WIND:HLDT?* 返回值： *1.50E-08*	
关联命令	:TRIGger:WINDow:HOLDoff	

:TRIGger:WINDow:HLEVel

描述	设置或查询窗口触发的绝对高电平。
命令格式	:TRIGger:WINDow:HLEVel <value> :TRIGger:WINDow:HLEVel?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

注意：高电平不能小于低电平（低电平设置指令:TRIGger:WINDow:LLEVel）。

返回格式	浮点型 NR3 格式
示例	设置窗口触发的绝对高电平为 0.5V： *:TRIGger:WINDow:HLEVel 5.00E-01* *TRIG:WIND:HLEV 5.00E-01* 查询当前窗口触发的绝对高电平： *TRIG:WIND:HLEV?* 返回值： *5.00E-01*
关联命令	:TRIGger:WINDow:LLEVel

:TRIGger:WINDow:HOLDoff

描述	设置或查询窗口触发的触发释抑类型。
命令格式	:TRIGger:WINDow:HOLDoff <holdoff_type> :TRIGger:WINDow:HOLDoff?
参数说明	<holdoff_type>:= {OFF|EVENts|TIME} ● OFF：关闭触发释抑 ● EVENts：事件。满足触发条件的次数 ● TIME：时间。指从触发之后到下一次重新启用触发电路之前示波器等待的时间
返回格式	{OFF|EVENts|TIME}
示例	设置窗口触发的触发释抑类型为关闭： *:TRIGger:WINDow:HOLDoff OFF* *TRIG:WIND:HOLD OFF* 查询当前窗口触发的触发释抑类型： *TRIG:WIND:HOLD?* 返回值： *OFF*
关联命令	:TRIGger:WINDow:HLDEVent :TRIGger:WINDow:HLEVel :TRIGger:WINDow:HSTart

:TRIGger:WINDow:HSTart

描述	设置或查询窗口触发的触发释抑启动条件。
命令格式	:TRIGger:WINDow:HSTart <start_holdoff> :TRIGger:WINDow:HSTart?

参数说明	<start_holdoff>:= {LAST_TRIG|ACQ_START} ● LAST_TRIG：上次触发。开始计算触发释抑的位置为上次的触发时间点。 ● ACQ_START：采集开始。开始计算触发释抑的位置为第一个满足触发条件的时间。
返回格式	{LAST_TRIG|ACQ_START}
示例	设置窗口触发的触发释抑启动条件为上次触发： *:TRIGger:WINDow:HSTart LAST_TRIG* *TRIG:WIND:HST LAST_TRIG* 查询窗口触发的触发释抑启动条件： *TRIG:WIND:HST?* 返回值： *LAST_TRIG*
关联命令	:TRIGger:WINDow:HOLDoff

:TRIGger:WINDow:LLEVel

描述	设置或查询窗口触发的绝对低电平。
命令格式	:TRIGger:WINDow:LLEVel <value> :TRIGger:WINDow:LLEVel?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

注意：
低电平不能大于高电平（高电平设置指令 :TRIGger:WINDow:HLEVel）。

返回格式	浮点型 NR3 格式
示例	设置窗口触发的绝对低电平为-0.5V： *:TRIGger:WINDow:LLEVel -5.00E-01* *TRIG:WIND:LLEV -5.00E-01* 查询当前窗口触发的绝对低电平： *TRIG:WIND:LLEV?* 返回值： *-5.00E-01*
关联命令	:TRIGger:SLOPe:HLEVel

:TRIGger:WINDow:NREJect

描述	设置或查询窗口触发的噪声抑制开关状态。
命令格式	:TRIGger:WINDow:NREJect <state> :TRIGger:WINDow:NREJect?
参数说明	<state>:= {OFF|ON}
返回格式	{OFF|ON}
示例	设置窗口触发的噪声抑制开关状态为开启： *:TRIGger:WINDow:NREJect ON* *TRIG:WIND:NREJ ON* 查询当前窗口触发的噪声抑制开关状态： *TRIG:WIND:NREJ?* 返回值： *ON*

:TRIGger:WINDow:SOURce

描述	设置或查询窗口触发的触发源。
命令格式	:TRIGger:WINDow:SOURce <source> :TRIGger:WINDow:SOURce?
参数说明	<source>:= {C<n>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式
返回格式	{C<n>}
示例	设置窗口触发的触发源为 C2： *:TRIGger:WINDow:SOURce C2* *TRIG:WIND:SOUR C2* 查询当前窗口触发的触发源：

	TRIG:WIND:SOUR? 返回值： *C2*

:TRIGger:WINDow:TYPE

描述	设置或查询窗口触发的窗口类型。
命令格式	:TRIGger:WINDow:TYPE <type> :TRIGger:WINDow:TYPE?
参数说明	<type>:= {ABSolute|RELative} ● ABSolute：绝对。在绝对窗口下，可单独调节高、低触发电平。 ● RELative：相对。在相对窗口下，可同时移动高、低触发电平，或放大/缩小两电平间的垂直距离，但不能单独调节高、低触发电平的垂直位移。
返回格式	{ABSolute|RELative}
示例	设置窗口触发的窗口类型为绝对： *:TRIGger:WINDow:TYPE ABSolute* *TRIG:WIND:TYPE ABS* 查询当前窗口触发的窗口类型： *TRIG:WIND:TYPE?* 返回值： *ABSolute*

:TRIGger:INTerval

:TRIGger:INTerval:COUPling

描述	设置或查询间隔触发的耦合方式。
命令格式	:TRIGger:INTerval:COUPling <mode> :TRIGger:INTerval:COUPling?
参数说明	<mode>:= {DC|AC|LFREJect|HFREJect} ● DC：直流耦合。通过信号的所有分量。 ● AC：交流耦合。抑制信号的直流分量，截止频率详见数据手册。 ● LFREJect：低频抑制。相当于高通滤波器，截止频率详见数据手册。 ● HFREJect：高频抑制。相当于低通滤波器，截止频率详见数据手册。
返回格式	{DC|AC|LFREJect|HFREJect}
示例	设置间隔触发的耦合方式为直流耦合：

	:TRIGger:INTerval:COUPling DC *TRIG:INT:COUP DC* 查询当前间隔触发的耦合方式： *TRIG:INT:COUP?* 返回值： *DC*

:TRIGger:INTerval:HLDEVent

描述	设置或查询间隔触发的触发释抑事件数。
命令格式	:TRIGger:INTerval:HLDEVent <value> :TRIGger:INTerval:HLDEVent?
参数说明	<value>:= 整型 NR1 格式，该值范围为 [1,100000000]
返回格式	整型 NR1 格式
示例	设置间隔触发的触发释抑事件数为 3： *:TRIGger:INTerval:HLDEVent 3* *TRIG:INT:HLDEV 3* 查询当前间隔触发的触发释抑事件数： *TRIG:INT:HLDEV?* 返回值： *3*
关联命令	:TRIGger:INTerval:HOLDoff

:TRIGger:INTerval:HLDTime

描述	设置或查询间隔触发的触发释抑时间。
命令格式	:TRIGger:INTerval:HLDTime <value> :TRIGger:INTerval:HLDTime?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD	[8.00E-09,3.00E+01]

	SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	
返回格式	浮点型 NR3 格式	
示例	设置间隔触发的触发释抑时间为 15ns： *:TRIGger:INTerval:HLDTime 1.50E-08* *TRIG:INT:HLDT 1.50E-08* 查询当前间隔触发的触发释抑时间： *TRIG:INT:HLDT?* 返回值： *1.50E-08*	
关联命令	:TRIGger:EDGE:HOLDoff	

:TRIGger:INTerval:HOLDoff

描述	设置或查询间隔触发的触发释抑类型。
命令格式	:TRIGger:INTerval:HOLDoff <holdoff_type> :TRIGger:INTerval:HOLDoff?
参数说明	<holdoff_type>:= {OFF|EVENts|TIME} ● OFF：关闭触发释抑 ● EVENts：事件。满足触发条件的次数 ● TIME：时间。指从触发之后到下一次重新启用触发电路之前示波器等待的时间
返回格式	{OFF|EVENts|TIME}
示例	设置间隔触发的触发释抑类型为关闭： *:TRIGger:INTerval:HOLDoff OFF* *TRIG:INT:HOLD OFF* 查询当前间隔触发的触发释抑类型： *TRIG:INT:HOLD?* 返回值： *OFF*
关联命令	:TRIGger:INTerval:HLDEVent :TRIGger:INTerval:HLDTime :TRIGger:INTerval:HSTart

:TRIGger:INTerval:HSTart

描述	设置或查询间隔触发的触发释抑启动条件。
命令格式	:TRIGger:INTerval:HSTart <start_holdoff> :TRIGger:INTerval:HSTart?
参数说明	<start_holdoff>:= {LAST_TRIG|ACQ_START} ● LAST_TRIG：上次触发。开始计算触发释抑的位置为上次的触发时间点。 ● ACQ_START：采集开始。开始计算触发释抑的位置为第一个满足触发条件的时间。
返回格式	{LAST_TRIG|ACQ_START}
示例	设置间隔触发的触发释抑启动条件为上次触发： *:TRIGger:INTerval:HSTart LAST_TRIG* *TRIG:INT:HST LAST_TRIG* 查询当前间隔触发的触发释抑启动条件： *TRIG:INT:HST?* 返回值： *LAST_TRIG*
关联命令	:TRIGger:INTerval:HOLDoff

:TRIGger:INTerval:LEVel

描述	设置或查询间隔触发的触发电平。
命令格式	:TRIGger:INTerval:LEVel <level_value> :TRIGger:INTerval:LEVel?
参数说明	<level_value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

	SDS2000X HD SDS1000X HD SDS800X HD	
返回格式	浮点型 NR3 格式	
示例	设置间隔触发的触发电平为 0.5V： *:TRIGger:INTerval:LEVel 5.00E-01* *TRIG:INT:LEV 5.00E-01* 查询当前间隔触发的触发电平： *TRIG:INT:LEV?* 返回值： *5.00E-01*	
关联命令	:TRIGger:INTerval:SOURce	

:TRIGger:INTerval:LIMit

描述	设置或查询间隔触发的限制条件类型。
命令格式	:TRIGger:INTerval:LIMit <type> :TRIGger:INTerval:LIMit?
参数说明	<type>:= {LESSthan|GREATerthan|INNer|OUTer} ● LESSthan（<=）：小于时间值。 ● GREATerthan（>=）：大于时间值。 ● INNer（[--,--]）：时间值范围内。 ● OUTer（--][--）：时间值范围外。
返回格式	{LESSthan|GREATerthan|INNer|OUTer}
示例	设置间隔触发的限制条件类型为时间值范围内： *:TRIGger:INTerval:LIMit INNer* *TRIG:INT:LIM INN* 查询当前间隔触发的限制条件类型： *TRIG:INT:LIM?* 返回值： *INNer*
关联命令	:TRIGger:INTerval:TLOWer :TRIGger:INTerval:TUPPer

:TRIGger:INTerval:NREJect

描述	设置或查询间隔触发的噪声抑制开关状态。
命令格式	:TRIGger:INTerval:NREJect <state> :TRIGger:INTerval:NREJect?
参数说明	<state>:= {OFF|ON}
返回格式	{OFF|ON}
示例	设置间隔触发的噪声抑制状态为开启： *:TRIGger:INTerval:NREJect ON* *TRIG:INT:NREJ ON* 查询当前间隔触发的噪声抑制状态： *TRIG:INT:NREJ?* 返回值： *ON*

:TRIGger:INTerval:SLOPe

描述	设置或查询间隔触发的斜率类型。
命令格式	:TRIGger:INTerval:SLOPe <slope_type> :TRIGger:INTerval:SLOPe?
参数说明	<state>:= {RISing|FALLing}
返回格式	{RISing|FALLing}
示例	设置间隔触发的斜率类型为上升沿： *:TRIGger:INTerval:SLOPe RISing* *TRIG:INT:SLOP RIS* 查询当前间隔触发的斜率类型： *TRIG:INT:SLOP?* 返回值： *RISing*

:TRIGger:INTerval:SOURce

描述	设置或查询间隔触发的触发源。
命令格式	:TRIGger:INTerval:SOURce <source> :TRIGger:INTerval:SOURce?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式

返回格式	{C<n>|D<d>}
示例	设置间隔触发的触发源为 C1： *:TRIGger:INTerval:SOURce C1* *TRIG:INT:SOUR C1* 查询当前间隔触发的触发源： *TRIG:INT:SOUR?* 返回值： *C1*

:TRIGger:INTerval:TLOWer

描述	设置或查询间隔触发限制条件的下限值。
命令格式	:TRIGger:INTerval:TLOWer <value> :TRIGger:INTerval:TLOWer?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A	[1.00E-09,2.00E+01]
SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-2.00E-09,2.00E+01]
SHS800X SHS1000X	[2.00E-09,4.20E+00]

注意：

- 下限值不能大于上限值（上限值设置指令:TRIGger:INTerval:TUPPer）。
- 当限制类型为“<=”时，此命令不可用。

返回格式	浮点型 NR3 格式
示例	设置间隔触发的下限值为 10ns： *:TRIGger:INTerval:TLOWer 1.00E-08* *TRIG:INT:TLOW 1.00E-08* 查询当前间隔触发的下限值： *TRIG:INT:TLOW?*

	返回值： *1.00E-08*
关联命令	:TRIGger:INTerval:LIMit :TRIGger:INTerval:TUPPer

:TRIGger:INTerval:TUPPer

描述	设置或查询间隔触发限制条件的上限值。
命令格式	:TRIGger:INTerval:TUPPer <value> :TRIGger:INTerval:TUPPer?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A / [1.00E-09,2.00E+01] SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD / [-2.00E-09,2.00E+01] SHS800X SHS1000X / [2.00E-09,4.20E+00] **注意**： ● 上限值不能小于下限值（下限值设置令:TRIGger:INTerval:TLOWer）。 ● 当限制类型为“>=”时，此命令不可用。
返回格式	浮点型 NR3 格式
示例	设置间隔触发的上限值为 30ns： *:TRIGger:INTerval:TUPPer 3.00E-08* *TRIG:INT:TUPP 3.00E-08* 查询间隔触发的上限值： *TRIG:INT:TUPP?* 返回值： *3.00E-08*
关联命令	:TRIGger:INTerval:LIMit :TRIGger:INTerval:TLOWer

:TRIGger:DROPout

:TRIGger:DROPout:COUPling

描述	设置或查询超时触发的耦合方式。
命令格式	:TRIGger:DROPout:COUPling <mode> :TRIGger:DROPout:COUPling?
参数说明	<mode>:= {DC|AC|LFREJect|HFREJect} ● DC：直流耦合。通过信号的所有分量。 ● AC：交流耦合。抑制信号的直流分量，截止频率详见数据手册。 ● LFREJect：低频抑制。相当于高通滤波器，截止频率详见数据手册。 ● HFREJect：高频抑制。相当于低通滤波器，截止频率详见数据手册。
返回格式	{DC|AC|LFREJect|HFREJect}
示例	设置超时触发的耦合方式为直流耦合： *:TRIGger:DROPout:COUPling DC* *TRIG:DROP:COUP DC* 查询当前超时触发的耦合方式： *TRIG:DROP:COUP?* 返回值： *DC*

:TRIGger:DROPout:HLDEVent

描述	设置或查询超时触发的触发释抑事件数。
命令格式	:TRIGger:DROPout:HLDEVent <value> :TRIGger:DROPout:HLDEVent?
参数说明	<value>:= 整型 NR1 格式，该值范围为 [1,100000000]
返回格式	整型 NR1 格式
示例	设置超时触发的触发释抑事件数为 3： *:TRIGger:DROPout:HLDEVent 3* *TRIG:DROP:HLDEV 3* 查询当前超时触发的触发释抑事件数： *TRIG:DROP:HLDEV?* 返回值： *3*
关联命令	:TRIGger:DROPout:HOLDoff

:TRIGger:DROPout:HLDTime

描述	设置或查询超时触发的触发释抑时间。
命令格式	:TRIGger:DROPout:HLDTime <value> :TRIGger:DROPout:HLDTime?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型**：SDS7000A, SDS6000 Pro, SDS6000A, SDS6000L, SDS5000X, SDS3000X HD, SDS2000X Plus, SDS2000X HD, SDS1000X HD, SDS800X HD **范围**：[8.00E-09,3.00E+01]
返回格式	浮点型 NR3 格式
示例	设置超时触发的触发释抑时间为 15ns： *:TRIGger:DROPout:HLDTime 1.50E-08* *TRIG:DROP:HLDT 1.50E-08* 查询当前超时触发的触发释抑时间： *TRIG:DROP:HLDT?* 返回值： *1.50E-08*
关联命令	:TRIGger:DROPout:HOLDoff

:TRIGger:DROPout:HOLDoff

描述	设置或查询超时触发的触发释抑类型。
命令格式	:TRIGger:DROPout:HOLDoff <holdoff_type> :TRIGger:DROPout:HOLDoff?
参数说明	<holdoff_type>:= {OFF|EVENts|TIME} ● OFF：关闭触发释抑 ● EVENts：事件。满足触发条件的次数 ● TIME：时间。指从触发之后到下一次重新启用触发电路之前示波器等待的时间

返回格式	{OFF|EVENts|TIME}
示例	设置超时触发的触发释抑类型为关闭： *:TRIGger:DROPout:HOLDoff OFF* *TRIG:DROP:HOLD OFF* 查询当前超时触发的触发释抑类型： *TRIG:DROP:HOLD?* 返回值： *OFF*
关联命令	:TRIGger:DROPout:HLDEVent :TRIGger:DROPout:HLDTime :TRIGger:DROPout:HSTart

:TRIGger:DROPout:HSTart

描述	设置或查询超时触发的触发释抑启动条件。
命令格式	:TRIGger:DROPout:HSTart <start_holdoff> :TRIGger:DROPout:HSTart?
参数说明	<start_holdoff>:= {LAST_TRIG|ACQ_START} ● LAST_TRIG：上次触发。开始计算触发释抑的位置为上次的触发时间点。 ● ACQ_START：采集开始。开始计算触发释抑的位置为第一个满足触发条件的时间。
返回格式	{LAST_TRIG|ACQ_START}
示例	设置超时触发的触发释抑启动条件为上次触发： *:TRIGger:DROPout:HSTart LAST_TRIG* *TRIG:DROP:HST LAST_TRIG* 查询当前超时触发的触发释抑启动条件： *TRIG:DROP:HST?* 返回值： *LAST_TRIG*
关联命令	:TRIGger:DROPout:HOLDoff

:TRIGger:DROPout:LEVel

描述	设置或查询超时触发的触发电平。
命令格式	:TRIGger:DROPout:LEVel <level_value> :TRIGger:DROPout:LEVel?

参数说明	<level_value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD — [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X — [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD — [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置超时触发的触发电平为 0.5V： *:TRIGger:DROPout:LEVel 5.00E-01* *TRIG:DROP:LEV 5.00E-01* 查询当前超时触发的触发电平： *TRIG:DROP:LEV?* 返回值： *5.00E-01*
关联命令	:TRIGger:DROPout:SOURce

:TRIGger:DROPout:NREJect

描述	设置或查询超时触发的噪声抑制开关状态。
命令格式	:TRIGger:DROPout:NREJect <state> :TRIGger:DROPout:NREJect?
参数说明	<state>:= {OFF|ON}
返回格式	{OFF|ON}
示例	设置超时触发的噪声抑制状态为开启： *:TRIGger:DROPout:NREJect ON* *TRIG:DROP:NREJ ON* 查询当前超时触发的噪声抑制状态：

	TRIG:DROP:NREJ? 返回值： *ON*

:TRIGger:DROPout:SLOPe

描述	设置或查询超时触发的斜率类型。
命令格式	:TRIGger:DROPout:SLOPe <slope_type> :TRIGger:DROPout:SLOPe?
参数说明	<state>:= {RISing|FALLing}
返回格式	{RISing|FALLing}
示例	设置超时触发的斜率类型为上升沿： *:TRIGger:DROPout:SLOPe RISing* *TRIG:DROP:SLOP RIS* 查询当前超时触发的斜率类型： *TRIG:DROP:SLOP?* 返回值： *RISing*

:TRIGger:DROPout:SOURce

描述	设置或查询超时触发的触发源。
命令格式	:TRIGger:DROPout:SOURce <source> :TRIGger:DROPout:SOURce?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置超时触发的触发源为 C2： *:TRIGger:DROPout:SOURce C2* *TRIG:DROP:SOUR C2* 查询当前超时触发的触发源： *TRIG:DROP:SOUR?* 返回值： *C2*

:TRIGger:DROPout:TIME

描述	设置或查询超时触发的超时时间。
命令格式	:TRIGger:DROPout:TIME <time> :TRIGger:DROPout:TIME?
参数说明	<time>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A / [1.00E-09,2.00E+01] SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD / [-2.00E-09,2.00E+01] SHS800X SHS1000X / [2.00E-09,4.20E+00]
返回格式	浮点型 NR3 格式
示例	设置超时触发的超时时间为 10ns： *:TRIGger:DROPout:TIME 1.00E-08* *TRIG:DROP:TIME 1.00E-08* 查询当前超时触发的触发时间： *TRIG:DROP:TIME?* 返回值： *1.00E-08*

:TRIGger:DROPout:TYPE

描述	设置或查询超时触发的超时类型。
命令格式	:TRIGger:DROPout:TYPE <type> :TRIGger:DROPout:TYPE?
参数说明	<type>:= {EDGE|STATe} ● EDGE：边沿 ● STATe：状态

返回格式	{EDGE|STATe}
示例	设置超时触发的超时类型为边沿： *:TRIGger:DROPout:TYPE EDGE* *TRIG:DROP:TYPE EDGE* 查询当前超时触发的超时类型： *TRIG:DROP:TYPE?* 返回值： *EDGE*

:TRIGger:RUNT

:TRIGger:RUNT:COUPling

描述	设置或查询欠幅触发的耦合方式。
命令格式	:TRIGger:RUNT:COUPling <mode> :TRIGger:RUNT:COUPling?
参数说明	<mode>:= {DC|AC|LFREJect|HFREJect} ● DC：直流耦合。通过信号的所有分量。 ● AC：交流耦合。抑制信号的直流分量，截止频率详见数据手册。 ● LFREJect：低频抑制。相当于高通滤波器，截止频率详见数据手册。 ● HFREJect：高频抑制。相当于低通滤波器，截止频率详见数据手册。
返回格式	{DC|AC|LFREJect|HFREJect}
示例	设置欠幅触发的耦合方式为直流耦合： *:TRIGger:RUNT:COUPling DC* *TRIG:RUNT:COUP DC* 查询当前欠幅触发的耦合方式： *TRIG:RUNT:COUP?* 返回值： *DC*

:TRIGger:RUNT:HLDEVent

描述	设置或查询欠幅触发的触发释抑事件数。
命令格式	:TRIGger:RUNT:HLDEVent <value> :TRIGger:RUNT:HLDEVent?
参数说明	<value>:= 整型 NR1 格式，该值范围为 [1,100000000]

返回格式	整型 NR1 格式
示例	设置欠幅触发的触发释抑事件数为 3： *:TRIGger:RUNT:HLDEVent 3* *TRIG:RUNT:HLDEV 3* 查询当前欠幅触发的触发释抑事件数： *TRIG:RUNT:HLDEV?* 返回值： *3*
关联命令	:TRIGger:RUNT:HOLDoff

:TRIGger:RUNT:HLDTime

描述	设置或查询欠幅触发的触发释抑时间。
命令格式	:TRIGger:RUNT:HLDTime <value> :TRIGger:RUNT:HLDTime?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[8.00E-09,3.00E+01]

返回格式	浮点型 NR3 格式
示例	设置欠幅触发的触发释抑时间为 15ns： *:TRIGger:RUNT:HLDTime 1.50E-08* *TRIG:RUNT:HLDT 1.50E-08* 查询当前欠幅触发的触发释抑时间： *TRIG:RUNT:HLDT?* 返回值： *1.50E-08*
关联命令	:TRIGger:RUNT:HOLDoff

:TRIGger:RUNT:HLEVel

描述	设置或查询欠幅触发的触发高电平。
命令格式	:TRIGger:RUNT:HLEVel <value> :TRIGger:RUNT:HLEVel?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD / [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X / [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD / [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏] **注意**：高电平不能小于低电平（低电平设置指令:TRIGger:RUNT:LLEVel）。
返回格式	浮点型 NR3 格式
示例	设置欠幅触发的高电平为 0.5V： *:TRIGger:RUNT:HLEVel 5.00E-01* *TRIG:RUNT:HLEV 5.00E-01* 查询当前欠幅触发的高电平： *TRIG:RUNT:HLEV?* 返回值： *5.00E-01*
关联命令	:TRIGger:RUNT:LLEVel

:TRIGger:RUNT:HOLDoff

描述	设置或查询欠幅触发的触发释抑类型。
命令格式	:TRIGger:RUNT:HOLDoff <holdoff_type> :TRIGger:RUNT:HOLDoff?

参数说明	<holdoff_type>:= {OFF|EVENts|TIME} ● OFF：关闭触发释抑 ● EVENts：事件。满足触发条件的次数 ● TIME：时间。指从触发之后到下一次重新启用触发电路之前示波器等待的时间
返回格式	{OFF|EVENts|TIME}
示例	设置欠幅触发的触发释抑类型为关闭： *:TRIGger:RUNT:HOLDoff OFF* *TRIG:RUNT:HOLD OFF* 查询当前欠幅触发的触发释抑类型： *TRIG:RUNT:HOLD?* 返回值： *OFF*
关联命令	:TRIGger:RUNT:HLDEVent :TRIGger:RUNT:HLDTime :TRIGger:RUNT:HSTart

:TRIGger:RUNT:HSTart

描述	设置或查询欠幅触发的触发释抑启动条件。
命令格式	:TRIGger:RUNT:HSTart <start_holdoff> :TRIGger:RUNT:HSTart?
参数说明	<start_holdoff>:= {LAST_TRIG|ACQ_START} ● LAST_TRIG：上次触发。开始计算触发释抑的位置为上次的触发时间点。 ● ACQ_START：采集开始。开始计算触发释抑的位置为第一个满足触发条件的时间。
返回格式	{LAST_TRIG|ACQ_START}
示例	设置欠幅触发的触发释抑启动条件为上次触发： *:TRIGger:RUNT:HSTart LAST_TRIG* *TRIG:RUNT:HST LAST_TRIG* 查询当前欠幅触发的触发释抑启动条件： *TRIG:RUNT:HST?* 返回值： *LAST_TRIG*
关联命令	:TRIGger:RUNT:HOLDoff

:TRIGger:RUNT:LIMit

描述	设置或查询欠幅触发的限制条件类型。
命令格式	:TRIGger:RUNT:LIMit <type> :TRIGger:RUNT:LIMit?
参数说明	<type>:= {LESSthan|GREATerthan|INNer|OUTer} ● LESSthan（<=）：小于时间值。 ● GREATerthan（>=）：大于时间值。 ● INNer（[--,--]）：时间值范围内。 ● OUTer（--][--）：时间值范围外。
返回格式	{LESSthan|GREATerthan|INNer|OUTer}
示例	设置欠幅触发的限制条件类型为时间值范围内： *:TRIGger:RUNT:LIMit INNer* *TRIG:RUNT:LIM INN* 查询当前欠幅触发的限制条件类型： *TRIG:RUNT:LIM?* 返回值： *INNer*
关联命令	:TRIGger:RUNT:TLOWer :TRIGger:RUNT:TUPPer

:TRIGger:RUNT:LLEVel

描述	设置或查询欠幅触发的触发低电平。
命令格式	:TRIGger:RUNT:LLEVel <value> :TRIGger:RUNT:LLEVel?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	

注意：低电平不能大于高电平（高电平设置指令:TRIGger:RUNT:HLEVel）。

返回格式	浮点型 NR3 格式
示例	设置欠幅触发的触发低电平为-0.5V： *:TRIGger:RUNT:LLEVel -5.00E-01* *TRIG:RUNT:LLEV -5.00E-01* 查询当前欠幅触发的触发低电平： *TRIG:RUNT:LLEV?* 返回值： *-5.00E-01*
关联命令	:TRIGger:RUNT:HLEVel

:TRIGger:RUNT:NREJect

描述	设置或查询欠幅触发的噪声抑制开关状态。
命令格式	:TRIGger:RUNT:NREJect <state> :TRIGger:RUNT:NREJect?
参数说明	<state>:= {OFF|ON}
返回格式	{OFF|ON}
示例	设置欠幅触发的噪声抑制状态为开启： *:TRIGger:RUNT:NREJect ON* *TRIG:RUNT:NREJ ON* 查询当前欠幅触发的噪声抑制开关状态： *TRIG:RUNT:NREJ?* 返回值： *ON*

:TRIGger:RUNT:POLarity

描述	设置或查询欠幅触发的极性。
命令格式	:TRIGger:RUNT:POLarity <polarity_type> :TRIGger:RUNT:POLarity?

参数说明	<polarity_type>:= {POSitive|NEGative}
返回格式	{POSitive|NEGative}
示例	设置欠幅触发的极性为正脉冲： *:TRIGger:RUNT:POLarity POSitive* *TRIG:RUNT:POL POS* 查询当前欠幅触发的极性： *TRIG:RUNT:POL?* 返回值： *POSitive*

:TRIGger:RUNT:SOURce

描述	设置或查询欠幅触发的触发源。
命令格式	:TRIGger:RUNT:SOURce <source> :TRIGger:RUNT:SOURce?
参数说明	<source>:= {C<n>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式
返回格式	{C<n>}
示例	设置欠幅触发的触发源为 C2： *:TRIGger:RUNT:SOURce C2* *TRIG:RUNT:SOUR C2* 查询当前欠幅触发的触发源： *TRIG:RUNT:SOUR?* 返回值： *C2*

:TRIGger:RUNT:TLOWer

描述	设置或查询欠幅触发限制条件的下限值。
命令格式	:TRIGger:RUNT:TLOWer <value> :TRIGger:RUNT:TLOWer?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A	[1.00E-09,2.00E+01]
SDS6000 Pro SDS6000A	[-2.00E-09,2.00E+01]

SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	
SHS800X SHS1000X	[2.00E-09,4.20E+00]

注意:

- 下限值不能大于上限值（上限值设置指令:TRIGger:RUNT:TUPPer）。
- 当限制类型为“<=”时，此命令不可用。

返回格式	浮点型 NR3 格式
示例	设置欠幅触发的下限值为 10ns: *:TRIGger:RUNT:TLOWer 1.00E-08* *TRIG:RUNT:TLOW 1.00E-08* 查询当前欠幅触发的下限值: *TRIG:RUNT:TLOW?* 返回值: *1.00E-08*
关联命令	:TRIGger:RUNT:LIMit :TRIGger:RUNT:TUPPer

:TRIGger:RUNT:TUPPer

描述	设置或查询欠幅触发限制条件的上限值。
命令格式	:TRIGger:RUNT:TUPPer <value> :TRIGger:RUNT:TUPPer?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表:

机型	范围
SDS7000A	[1.00E-09,2.00E+01]
SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus	[-2.00E-09,2.00E+01]

SDS2000X HD SDS1000X HD SDS800X HD	
SHS800X SHS1000X	[2.00E-09,4.20E+00]

注意:

- 上限值不能小于下限值（下限值设置指令:TRIGger:RUNT:TLOWer）。
- 当限制类型为“>=”时，此命令不可用。

返回格式	浮点型 NR3 格式
示例	设置欠幅触发的上限值为 30ns: *:TRIGger:RUNT:TUPPer 3.00E-08* *TRIG:RUNT:TUPP 3.00E-08* 查询当前欠幅触发的上限值: *TRIG:RUNT:TUPP?* 返回值: *3.00E-08*
关联命令	:TRIGger:RUNT:LIMit :TRIGger:RUNT:TLOWer

:TRIGger:PATTern

:TRIGger:PATTern:HLDEVent

描述	设置或查询码型触发的触发释抑事件数。
命令格式	:TRIGger:PATTern:HLDEVent <value> :TRIGger:PATTern:HLDEVent?
参数说明	<value>:= 整型 NR1 格式，该值范围为 [1,100000000]
返回格式	整型 NR1 格式
示例	设置码型触发的触发释抑事件数为 3: *:TRIGger:PATTern:HLDEVent 3* *TRIG:PATT:HLDEV 3* 查询当前码型触发的触发释抑事件数: *TRIG:PATT:HLDEV?* 返回值: *3*
关联命令	:TRIGger:PATTern:HOLDoff

:TRIGger:PATTern:HLDTime

描述	设置或查询码型触发的触发释抑时间。
命令格式	:TRIGger:PATTern:HLDTime <value> :TRIGger:PATTern:HLDTime?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS6000 Pro SDS6000A SDS6000L SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD / [8.00E-09,3.00E+01] SHS800X SHS1000X / [80.00E-09,1.5E+00]
返回格式	浮点型 NR3 格式
示例	设置码型触发的触发释抑时间为 15ns： *:TRIGger:PATTern:HLDTime 1.50E-08* *TRIG:PATT:HLDT 1.50E-08* 查询当前码型触发的触发释抑时间： *TRIG:PATT:HLDT?* 返回值： *1.50E-08*
关联命令	:TRIGger:PATTern:HOLDoff

:TRIGger:PATTern:HOLDoff

描述	设置或查询码型触发的触发释抑类型。
命令格式	:TRIGger:PATTern:HOLDoff <holdoff_type> :TRIGger:PATTern:HOLDoff?
参数说明	<holdoff_type>:= {OFF|EVENts|TIME} ● OFF：关闭触发释抑 ● EVENts：事件。满足触发条件的次数 ● TIME：时间。指从触发之后到下一次重新启用触发电路之前示波器等待的时间

返回格式	{OFF|EVENts|TIME}
示例	设置码型触发的触发释抑类型为关闭： *:TRIGger:PATTern:HOLDoff OFF* *TRIG:PATT:HOLD OFF* 查询当前码型触发的触发释抑类型： *TRIG:PATT:HOLD?* 返回值： *OFF*
关联命令	:TRIGger:PATTern:HLDEVent :TRIGger:PATTern:HLDTime :TRIGger:PATTern:HSTart

:TRIGger:PATTern:HSTart

描述	设置或查询码型触发的触发释抑启动条件。
命令格式	:TRIGger:PATTern:HSTart <start_holdoff> :TRIGger:PATTern:HSTart?
参数说明	<start_holdoff>:= {LAST_TRIG|ACQ_START} ● LAST_TRIG：上次触发。开始计算触发释抑的位置为上次的触发时间点。 ● ACQ_START：采集开始。开始计算触发释抑的位置为第一个满足触发条件的时间。
返回格式	{LAST_TRIG|ACQ_START}
示例	设置码型触发的触发释抑启动条件为上次触发： *:TRIGger:PATTern:HSTart LAST_TRIG* *TRIG:PATT:HST LAST_TRIG* 查询当前码型触发的触发释抑启动条件： *TRIG:PATT:HST?* 返回值： *LAST_TRIG*
关联命令	:TRIGger:PATTern:HOLDoff

:TRIGger:PATTern:INPut

描述	设置或查询码型触发源的逻辑状态。触发源包括模拟通道和数字通道。
命令格式	:TRIGger:PATTern:INPut <logic>[...[,<logic>]] :TRIGger:PATTern:INPut?

参数说明	<logic>:= {X|L|H} ● X：不关心，无效。 ● H：高电平有效。 ● L：低电平有效。 **注意**：参数按 C1-C<n>、D0-D15 的顺序配置到相应的信源。
返回格式	{X|L|H}
示例	设置 C1 为逻辑高、C2 为逻辑高、C3 为逻辑低、C4 为不关心、所有数字通道均为不关心： *:TRIGger:PATTern:INPut H,H,L,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X* *TRIG:PATT:INP H,H,L,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X* 查询码型触发源的逻辑状态： *TRIG:PATT:INP?* 返回值： *H,H,L,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X*

:TRIGger:PATTern:LEVel

描述	设置或查询码型触发源的逻辑电平。
命令格式	:TRIGger:PATTern:LEVel <source>,<value> :TRIGger:PATTern:LEVel? <source>
参数说明	<source>:= {C<n>} ● C<n>:= 模拟通道，<n>:= 通道序号，整型 NR1 格式 <value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置码型触发 C2 的逻辑电平为 0.5V： *:TRIGger:PATTern:LEVel C2,5.00E-01* *TRIG:PATT:LEV C2,5.00E-01* 查询当前码型触发 C2 的逻辑电平： *TRIG:PATT:LEV? C2* 返回值： *C2,5.00E-01*
关联命令	:TRIGger:PATTern:INPut

:TRIGger:PATTern:LIMit

描述	设置或查询码型触发的限制条件类型。
命令格式	:TRIGger:PATTern:LIMit <type> :TRIGger:PATTern:LIMit?
参数说明	<type>:= {LESSthan|GREATerthan|INNer|OUTer} ● LESSthan（<=）：小于时间值。 ● GREATerthan（>=）：大于时间值。 ● INNer（[--,--]）：时间值范围内。 ● OUTer（--][--）：时间值范围外。
返回格式	{LESSthan|GREATerthan|INNer|OUTer}
示例	设置码型触发的限制条件类型为时间值范围内： *:TRIGger:PATTern:LIMit INNer* *TRIG:PATT:LIM INN* 查询当前码型触发的限制条件类型： *TRIG:PATT:LIM?* 返回值： *INNer*
关联命令	:TRIGger:PATTern:TLOWer :TRIGger:PATTern:TUPPer

:TRIGger:PATTern:LOGic

描述	设置或查询码型触发的逻辑关系。
命令格式	:TRIGger:PATTern:LOGic <logic_type> :TRIGger:PATTern:LOGic?

参数说明	<logic_type>:= {AND|OR|NAND|NOR}
返回格式	{AND|OR|NAND|NOR}
示例	设置码型触发的逻辑关系为 AND： *:TRIGger:PATTern:LOGic AND* *TRIG:PATT:LOG AND* 查询当前码型触发的逻辑关系： *TRIG:PATT:LOG?* 返回值： *AND*

:TRIGger:PATTern:TLOWer

描述	设置或查询码型触发限制条件的下限值。
命令格式	:TRIGger:PATTern:TLOWer <value> :TRIGger:PATTern:TLOWer?
参数说明	<value>:= 浮点型 NR3 格式 **注意**： ● 下限值不能大于上限值（上限值设置指令:TRIGger:PATTern:TUPPer）。 ● 当限制类型为“<=”时，此命令不可用。
返回格式	浮点型 NR3 格式
示例	设置码型触发的下限值为 10ns： *:TRIGger:PATTern:TLOWer 1.00E-08* *TRIG:PATT:TLOW 1.00E-08* 查询当前码型触发的下限值： *TRIG:PATT:TLOW?* 返回值： *1.00E-08*
关联命令	:TRIGger:PATTern:LIMit :TRIGger:PATTern:TUPPer

:TRIGger:PATTern:TUPPer

描述	设置或查询码型触发限制条件的上限值。
命令格式	:TRIGger:PATTern:TUPPer <value> :TRIGger:PATTern:TUPPer?

参数说明	<value>:= 浮点型 NR3 格式 **注意**: ● 上限值不能小于下限值（下限值设置指令:TRIGger:PATTern:TLOWer）。 ● 当限制类型为“>=”时，此命令不可用。
返回格式	浮点型 NR3 格式
示例	设置码型触发的上限值为 30ns: *:TRIGger:PATTern:TUPPer 3.00E-08* *TRIG:PATT:TUPP 3.00E-08* 查询当前码型触发的上限值: *TRIG:PATT:TUPP?* 返回值: *3.00E-08*
关联命令	:TRIGger:PATTern:LIMit :TRIGger:PATTern:TLOWer

:TRIGger:QUALified

:TRIGger:QUALified:ELEVel

描述	设置或查询前提边沿触发的边沿触发电平。
命令格式	:TRIGger:QUALified:ELEVel <value> :TRIGger:QUALified:ELEVel?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表:

机型	范围
SDS7000A	[-4.26*垂直档位-垂直偏移，4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L	[-4.5*垂直档位-垂直偏移，4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移，4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式

示例	设置前提边沿触发的边沿触发电平为 0.5V： *:TRIGger:QUALified:ELEVel 5.00E-01* *TRIG:QUAL:ELEV 5.00E-01* 查询当前前提边沿触发的边沿触发电平： *TRIG:QUAL:ELEV?* 返回值： *5.00E-01*
关联命令	:TRIGger:QUALified:QLEVel

:TRIGger:QUALified:ESLope

描述	设置或查询前提边沿触发的边沿触发斜率类型。
命令格式	:TRIGger:QUALified:ESLope <type> :TRIGger:QUALified:ESLope?
参数说明	<type>:= {RISing|FALLing}
返回格式	{RISing|FALLing}
示例	设置前提边沿触发的边沿触发斜率类型为上升沿： *:TRIGger:QUALified:ESLope RISing* *TRIG:QUAL:ESL RIS* 查询当前前提边沿触发的边沿触发斜率类型： *TRIG:QUAL:ESL?* 返回值： *RISing*
关联命令	:TRIGger:QUALified:TYPE

:TRIGger:QUALified:ESource

描述	设置或查询前提边沿触发的边沿触发信源。
命令格式	:TRIGger:QUALified:ESource <source> :TRIGger:QUALified:ESource?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置前提边沿触发的边沿触发源为 C1： *:TRIGger:QUALified:ESource C1* *TRIG:QUAL:ES C1*

	查询前提边沿触发的边沿触发源： *TRIG:QUAL:ES?* 返回值： *C1*
关联命令	:TRIGger:QUALified:QSource

:TRIGger:QUALified:LIMit

描述	设置或查询前提边沿触发的限制条件类型。
命令格式	:TRIGger:QUALified:LIMit <type> :TRIGger:QUALified:LIMit?
参数说明	<type>:= {LESSthan|GREATerthan|INNer|OUTer} ● LESSthan（<=）：小于时间值。 ● GREATerthan（>=）：大于时间值。 ● INNer（[--,--]）：时间值范围内。 ● OUTer（--][--）：时间值范围外。
返回格式	{LESSthan|GREATerthan|INNer|OUTer}
示例	设置前提边沿触发的限制条件类型为时间值范围内： *:TRIGger:QUALified:LIMit INNer* *TRIG:QUAL:LIM INN* 查询前提边沿触发的限制条件类型： *TRIG:QUAL:LIM?* 返回值： *INNer*
关联命令	:TRIGger:QUALified:TLOWer :TRIGger:QUALified:TUPPer

:TRIGger:QUALified:QLEVel

描述	设置或查询前提边沿触发的前提信号电平。
命令格式	:TRIGger:QUALified:QLEVel <value> :TRIGger:QUALified:QLEVel?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]

	SDS6000 Pro SDS6000A SDS6000L	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
	SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式	
示例	设置前提边沿触发的前提信号电平为 0.5V： *:TRIGger:QUALified:QLEVel 5.00E-01* *TRIG:QUAL:QLEV 5.00E-01* 查询当前前提边沿触发的前提信号电平： *TRIG:QUAL:QLEV?* 返回值： *5.00E-01*	
关联命令	:TRIGger:QUALified:ELEVel	

:TRIGger:QUALified:QSource

描述	设置或查询前提边沿触发的前提信号信源。
命令格式	:TRIGger:QUALified:QSource <source> :TRIGger:QUALified:QSource?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置前提边沿触发的前提信号信源为 C1： *:TRIGger:QUALified:QSource C1* *TRIG:QUAL:QS C1* 查询前提边沿触发的前提信号信源： *TRIG:QUAL:QS?* 返回值： *C1*
关联命令	:TRIGger:QUALified:ESource

:TRIGger:QUALified:TLOWer

描述	设置或查询前提边沿触发限制条件的下限值。（仅当前提类型为“电平且限时”或“边沿且限时”有效）
命令格式	:TRIGger:QUALified:TLOWer <value> :TRIGger:QUALified:TLOWer?
参数说明	<value>:= 浮点型 NR3 格式 **注意**: ● 下限值不能大于上限值（上限值设置命令:TRIGger:QUALified:TUPPer）。 ● 当限制类型为“<=”时，此命令不可用。
返回格式	浮点型 NR3 格式
示例	设置前提边沿触发的下限值为 10ns： *:TRIGger:QUALified:TLOWer 1.00E-08* *TRIG:QUAL:TLOW 1.00E-08* 查询当前前提边沿触发的下限值： *TRIG:QUAL:TLOW?* 返回值： *1.00E-08*
关联命令	:TRIGger:QUALified:LIMit :TRIGger:QUALified:TUPPer

:TRIGger:QUALified:TUPPer

描述	设置或查询前提边沿触发限制条件的上限值。
命令格式	:TRIGger:QUALified:TUPPer <value> :TRIGger:QUALified:TUPPer?
参数说明	<value>:= 浮点型 NR3 格式 **注意**: ● 上限值不能小于下限值（下限值设置令:TRIGger:QUALified:TLOWer）。 ● 当限制类型为“>=”时，此命令不可用。
返回格式	浮点型 NR3 格式
示例	设置前提边沿触发的上限值为 30ns： *:TRIGger:QUALified:TUPPer 3.00E-08* *TRIG:QUAL:TUPP 3.00E-08* 查询当前前提边沿触发的上限值： *TRIG:QUAL:TUPP?*

	返回值： *3.00E-08*
关联命令	:TRIGger:QUALified:LIMit :TRIGger:QUALified:TLOWer

:TRIGger:QUALified:TYPE

描述	设置或查询前提边沿触发的前提类型。
命令格式	:TRIGger:QUALified:TYPE <type>[,<option>] :TRIGger:QUALified:TYPE?
参数说明	<type>:= {STATe|STATE_DLY|EDGE|EDGE_DLY} <option>:= {LOW|HIGH}，当前提类型为“STATe”或“STATE_DLY”时有效 <option>:= {RISing|FALLing}，当前提类型为“EDGE”或“EDGE_DLY”时有效
返回格式	<type>[,<option>]
示例	设置前提边沿触发的前提类型为边沿： *:TRIGger:QUALified:TYPE EDGE* *TRIG:QUAL:TYPE EDGE* 查询当前前提边沿触发的前提类型： *TRIG:QUAL:TYPE?* 返回值： *EDGE*

:TRIGger:DELay

:TRIGger:DELay:SOURce<n>

描述	设置或查询延时触发信源 A 的状态，或触发信源 B 的信源。
命令格式	:TRIGger:DELay:SOURce<n> <value>[...[,<value>]] :TRIGger:DELay:SOURce<n>?
参数说明	<n>:= {1|2}，触发信源序号。设置触发信源 A 斜率时可省略 <value>:= {X|L|H}，设置触发信源 A 状态时的可选项 ● X：代表不关心，无效。 ● H：代表高电平有效。 ● L：代表低电平有效。 **注意**：参数按 C1-C<n>、D0-D15 的顺序配置到相应的信源。

	<value>:={C<n>|D<d>}，设置触发信源 B 的信源时的可选项 ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	<value>[...[,<value>]]
示例	设置延时触发信源 A 的状态：通道 1 的逻辑输入为 H、通道 2 的逻辑输入为 L、通道 3 的逻辑输入为 L、通道 4 的逻辑输入为 X、所有数字通道的逻辑输入为 X： *:TRIGger:DELay:SOURce H,L,L,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X* *TRIG:DEL:SOUR H,L,L,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X* 查询当前延时触发信源 A 的状态： *TRIG:DEL:SOUR?* 返回值： *H,L,L,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X*

:TRIGger:DELay:SLOPe<n>

描述	设置或查询延时触发信源 A/B 的斜率类型。
命令格式	:TRIGger:DELay:SLOPe<n> <slope_type> :TRIGger:DELay:SLOPe<n>?
参数说明	<n>:={1|2}，触发信源序号。设置触发信源 A 斜率时可省略 <slope_type>:={RISing|FALLing}
返回格式	{RISing|FALLing}
示例	设置延时触发信源 A 的斜率类型为上升沿： *:TRIGger:DELay:SLOPe RISing* *TRIG:DEL:SLOP RIS* 查询当前延时触发信源 A 的斜率类型： *TRIG:DEL:SLOP?* 返回值： *RISing*

:TRIGger:DELay:LEVel<n>

描述	设置或查询延时触发信源 A/B 的触发电平。
命令格式	:TRIGger:DELay:LEVel<n> [<source>,]<value> :TRIGger:DELay:LEVel<n>?
参数说明	<n>:={1|2}，触发信源序号。设置触发信源 A 斜率时可省略

<source>:= {C<n>}，设置触发信源 A 的触发电平时需指定该参数

- C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式

<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L	[-4.5*垂直档位-垂直偏移，4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移，4.1*垂直档位-垂直偏移]

返回格式 [<source>,]<value>

示例 设置延时触发 C2 信源 A 的触发电平为 0.5V：

:TRIGger:DELay:LEVel C2,5.00E-01

TRIG:DEL:LEV C2,5.00E-01

查询当前延时触发 C2 信源 A 的触发电平：

TRIG:DEL:LEV? C2

返回值：

C2,5.00E-01

:TRIGger:DELay:LIMit

描述 设置或查询延时触发的限制条件类型。

命令格式

:TRIGger:DELay:LIMit <type>

:TRIGger:DELay:LIMit?

参数说明 <type>:= {LESSthan|GREATerthan|INNer|OUTer}

- LESSthan（<=）：小于时间值。
- GREATerthan（>=）：大于时间值。
- INNer（[--,--]）：时间值范围内。
- OUTer（--] [--）：时间值范围外。

返回格式 {LESSthan|GREATerthan|INNer|OUTer}

示例 设置延时触发的限制条件类型为小于时间值：

:TRIGger:DELay:LIMit LESSthan

TRIG:DEL:LIM LESS

	查询当前延时触发的限制条件类型： *TRIG:DEL:LIM?* 返回值： *LESSthan*
关联命令	:TRIGger:DELay:TLOWer :TRIGger:DELay:TUPPer

:TRIGger:DELay:TLOWer

描述	设置或查询延时触发限制条件的下限值。
命令格式	:TRIGger:DELay:TLOWer <value> :TRIGger:DELay:TLOWer?
参数说明	<value>:= 浮点型 NR3 格式 **注意**： ● 下限值不能大于上限值（上限值设置指令:TRIGger:DELay:TUPPer）。 ● 当限制类型为“<=”时，此命令不可用。
返回格式	浮点型 NR3 格式
示例	设置延时触发的下限值为 10ns： *:TRIGger:DELay:TLOWer 1.00E-08* *TRIG:DEL:TLOW 1.00E-08* 查询当前延时触发的下限值： *TRIG:DEL:TLOW?* 返回值： *1.00E-08*
关联命令	:TRIGger:DELay:LIMit :TRIGger:DELay:TUPPer

:TRIGger:DELay:TUPPer

描述	设置或查询延时触发限制条件的上限值。
命令格式	:TRIGger:DELay:TUPPer <value> :TRIGger:DELay:TUPPer?
参数说明	<value>:= 浮点型 NR3 格式 **注意**：

	● 上限值不能小于下限值（下限值设置指令:TRIGger:DELay:TLOWer）。 ● 当限制类型为“>=”时，此命令不可用。
返回格式	浮点型 NR3 格式
示例	设置延时触发的上限值为 30ns： *:TRIGger:DELay:TUPPer 3.00E-08* *TRIG:DEL:TUPP 3.00E-08* 查询当前延时触发的上限值： *TRIG:DEL:TUPP?* 返回值： *3.00E-08*
关联命令	:TRIGger:DELay:LIMit :TRIGger:DELay:TLOWer

:TRIGger:NEDGe

:TRIGger:NEDGe:SOURce

描述	设置或查询第 N 边沿触发的触发源。
命令格式	:TRIGger:NEDGe:SOURce <source> :TRIGger:NEDGe:SOURce?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置第 N 边沿触发的触发源为 C1： *:TRIGger:NEDGe:SOURce C1* *TRIG:NEDG:SOUR C1* 查询当前第 N 边沿触发的触发源： *TRIG:NEDG:SOUR?* 返回值： *C1*
关联命令	:TRIGger:NEDGe:LEVel

:TRIGger:NEDGe:SLOPe

描述	设置或查询第 N 边沿触发的斜率类型。

命令格式	:TRIGger:NEDGe:SLOPe <slope_type> :TRIGger:NEDGe:SLOPe?
参数说明	<slope_type>:= {RISing|FALLing}
返回格式	{RISing|FALLing}
示例	设置第 N 边沿触发的斜率类型为上升沿： *:TRIGger:NEDGe:SLOPe RISing* *TRIG:NEGD:SLOP RIS* 查询当前第 N 边沿触发的斜率类型： *TRIG:NEDG:SLOP?* 返回值： *RISing*

:TRIGger:NEDGe:IDLE

描述	设置或查询第 N 边沿触发的空闲时间。
命令格式	:TRIGger:NEDGe:IDLE <value> :TRIGger:NEDGe:IDLE?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** | **范围** SDS7000A SDS5000X SDS6000 Pro SDS6000A SDS6000L SDS3000X HD SDS2000X HD SDS1000X HD SDS800X HD [8.00E-09, 2.00E+01]
返回格式	浮点型 NR3 格式
示例	设置第 N 边沿触发的空闲时间为 15ns： *:TRIGger:NEDGe:IDLE 1.50E-08* *TRIG:NEDG:IDLE 1.5E-08* 查询当前第 N 边沿触发的空闲时间： *TRIG:NEDG:IDLE?* 返回值： *1.50E-08*

:TRIGger:NEDGe:EDGE

描述	设置或查询第 N 边沿触发的边沿数。
命令格式	:TRIGger:NEDGe:EDGE <value> :TRIGger:NEDGe:EDGE?
参数说明	<value>:= 整型 NR1 格式，该值范围为 [1,65535]
返回格式	整型 NR1 格式
示例	设置第 N 边沿触发的边沿数为 3： *:TRIGger:NEDGe:EDGE 3* *TRIG:NEDG:EDGE 3* 查询当前第 N 边沿触发的边沿数： *TRIG:NEDG:EDGE?* 返回值： *3*

:TRIGger:NEDGe:LEVel

描述	设置或查询第 N 边沿触发的触发电平。
命令格式	:TRIGger:NEDGe:LEVel <level_value> :TRIGger:NEDGe:LEVel?
参数说明	<level_value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD — [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L — [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X HD SDS1000X HD SDS800X HD — [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置第 N 边沿触发的触发电平为 0.5V： *:TRIGger:NEDGe:LEVel 5.00E-01* *TRIG:NEDG:LEV 5.00E-01*

	查询当前第 N 边沿触发的触发电平： *TRIG:NEDG:LEV?* 返回值： *5.00E-01*
关联命令	:TRIGger:NEDGe:SOURce

:TRIGger:NEDGe:HOLDoff

描述	设置或查询第 N 边沿触发的触发释抑类型。
命令格式	:TRIGger:NEDGe:HOLDoff <holdoff_type> :TRIGger:NEDGe:HOLDoff?
参数说明	<holdoff_type>:= {OFF|EVENts|TIME} ● OFF：关闭触发释抑 ● EVENts：事件。满足触发条件的次数 ● TIME：时间。指从触发之后到下一次重新启用触发电路之前示波器等待的时间
返回格式	{OFF|EVENts|TIME}
示例	设置第 N 边沿触发触发的触发释抑类型为关闭： *:TRIGger:NEDGe:HOLDoff OFF* *TRIG:NEDG:HOLD OFF* 查询当前第 N 边沿触发的触发释抑类型： *TRIG:NEDG:HOLD?* 返回值： *OFF*
关联命令	:TRIGger:NEDGe:HLDEVent :TRIGger:NEDGe:HLDTime :TRIGger:NEDGe:HSTart

:TRIGger:NEDGe:HLDTime

描述	设置或查询第 N 边沿触发的触发释抑时间。
命令格式	:TRIGger:NEDGe:HLDTime <value> :TRIGger:NEDGe:HLDTime?
参数说明	<value>:= 浮点型 NR3 格式。有关详细信息，请参见下表： **机型** | **范围** SDS7000A | {8.00E-09,3.00E+01}

	SDS6000 Pro SDS6000A SDS6000L SDS3000X HD SDS2000X HD SDS1000X HD SDS800X HD	
返回格式	浮点型 NR3 格式	
示例	设置第 N 边沿触发的触发释抑时间为 15ns： *:TRIGger:NEDGe:HLDTime 1.50E-08* *TRIG:NEDG:HLDT 1.50E-08* 查询当前第 N 边沿触发的触发释抑时间： *TRIG:NEDG:HLDT?* 返回值： *1.50E-08*	
关联命令	:TRIGger:NEDGe:HOLDoff	

:TRIGger:NEDGe:HLDEVent

描述	设置或查询第 N 边沿触发的触发释抑事件数。
命令格式	:TRIGger:NEDGe:HLDEVent <value> :TRIGger:NEDGe:HLDEVent?
参数说明	<value> := 整型 NR1 格式，该值范围为 [1,100000000]
返回格式	整型 NR1 格式
示例	设置第 N 边沿触发的触发释抑事件数为 3： *:TRIGger:NEDGe:HLDEvent 3* *TRIG:NEDG:HLDEV 3* 查询当前第 N 边沿触发的触发释抑事件数： *TRIG:NEDG:HLDEV?* 返回值： *3*
关联命令	:TRIGger:NEDGe:HOLDoff

:TRIGger:NEDGe:HSTart

描述	设置或查询第 N 边沿触发的触发释抑启动条件。

命令格式	:TRIGger:NEDGe:HSTart <start_holdoff> :TRIGger:NEDGe:HSTart?
参数说明	<start_holdoff>:= {LAST_TRIG|ACQ_START} ● LAST_TRIG：上次触发。开始计算触发释抑的位置为上次的触发时间点。 ● ACQ_START：采集开始。开始计算触发释抑的位置为第一个满足触发条件的时间。
返回格式	{LAST_TRIG|ACQ_START}
示例	设置第 N 边沿触发的触发释抑启动条件为上次触发： *:TRIGger:NEDGe:HSTart LAST_TRIG* *TRIG:NEDG:HST LAST_TRIG* 查询当前第 N 边沿触发的触发释抑启动条件： *TRIG:NEDG:HST?* 返回值： *LAST_TRIG*
关联命令	:TRIGger:NEDGe:HOLDoff

:TRIGger:NEDGe:NREJect

描述	设置或查询第 N 边沿触发的噪声抑制开关状态。
命令格式	:TRIGger:NEDGe:NREJect <state> :TRIGger:NEDGe:NREJect?
参数说明	<state>:= {OFF|ON}
返回格式	{OFF|ON}
示例	设置第 N 边沿触发的噪声抑制开启： *:TRIGger:NEDGe:NREJect ON* *TRIG:NEDG:NREJ ON* 查询当前第 N 边沿触发的噪声抑制状态： *TRIG:NEDG:NREJ?* 返回值： *ON*

:TRIGger:SHOLd

:TRIGger:SHOLd:TYPE

描述	设置或查询建立/保持触发的类型。
命令格式	:TRIGger:SHOLd:TYPE <type> :TRIGger:SHOLd:TYPE?
参数说明	<type>:= {SETup|HOLD}
返回格式	{SETup|HOLD}
示例	设置建立/保持触发的类型为建立： *:TRIGger:SHOLd:TYPE SETup* *TRIG:SHOL:TYPE SET* 查询当前建立/保持触发的类型： *TRIG:SHOL:TYPE?* 返回值： *SETup*

:TRIGger:SHOLd:CSource

描述	设置或查询建立/保持触发的时钟信源。
命令格式	:TRIGger:SHOLd:CSource <source> :TRIGger:SHOLd:CSource?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置建立/保持触发的时钟信源为 C1： *:TRIGger:SHOLd:CSource C1* *TRIG:SHOL:CS C1* 查询当前建立/保持触发的时钟信源： *TRIG:SHOL:CS?* 返回值： *C1*
关联命令	:TRIGger:SHOLd:CTHReshold

:TRIGger:SHOLd:CTHReshold

描述	设置或查询建立/保持触发的时钟阈值。
命令格式	:TRIGger:SHOLd:CTHReshold <value> :TRIGger:SHOLd:CTHReshold?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD — [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L — [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD — [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置建立/保持触发的时钟阈值为 1V： *:TRIGger:SHOLd:CTHReshold 1.50E+00* *TRIG:SHOL:CTHR 1.50E+00* 查询当前建立/保持触发的时钟阈值： *TRIG:SHOL:CTHR?* 返回值： *1.50E+00*
关联命令	:TRIGger:SHOLd:DSource

:TRIGger:SHOLd:SLOPe

描述	设置或查询建立/保持触发的斜率类型。
命令格式	:TRIGger:SHOLd:SLOPe <slope_type> :TRIGger:SHOLd:SLOPe?
参数说明	<state>:= {RISing|FALLing}
返回格式	{RISing|FALLing}
示例	设置建立/保持触发的斜率类型为上升沿： *:TRIGger:SHOLd:SLOPe RISing*

	TRIG:SHOL:SLOP RIS 查询当前建立/保持触发的斜率： *TRIG:SHOL:SLOP?* 返回值： *RISing*

:TRIGger:SHOLd:DSource

描述	设置或查询建立/保持触发的数据信源。
命令格式	:TRIGger:SHOLd:DSource <source> :TRIGger:SHOLd:DSource?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置建立/保持触发的数据信源为 C1： *:TRIGger:SHOLd:CSource C1* *TRIG:SHOL:CS C1* 查询当前建立/保持触发的数据信源： *TRIG:SHOL:CS?* 返回值： *C1*
关联命令	:TRIGger:SHOLd:CTHReshold

:TRIGger:SHOLd:DTHReshold

描述	设置或查询建立/保持触发的数据源阈值。
命令格式	:TRIGger:SHOLd:DTHReshold <value> :TRIGger:SHOLd:DTHReshold?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]

	SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式	
示例	设置建立/保持触发的数据源阈值为 1V： *:TRIGger:SHOLd:DTHReshold 1.50E+00* *TRIG:SHOL:DTHR 1.50E+00* 查询当前建立/保持触发的数据源阈值： *TRIG:SHOL:DTHR?* 返回值： *1.50E+00*	
关联命令	:TRIGger:SHOLd:DSource	

:TRIGger:SHOLd:LEVel

描述	设置或查询建立/保持触发的数据源电平状态。
命令格式	:TRIGger:SHOLd:LEVel <level_state> :TRIGger:SHOLd:LEVel?
参数说明	<level_value>:= {LOW|HIGH}
返回格式	{LOW|HIGH}
示例	设置建立/保持触发的数据源电平状态为高电平： *:TRIGger:SHOLd:LEVel HIGH* *TRIG:SHOL:LEV HIGH* 查询当前建立/保持触发的数据源电平状态： *TRIG:SHOL:LEV?* 返回值： *HIGH*

:TRIGger:SHOLd:LIMit

描述	设置或查询建立/保持触发的限制条件类型。
命令格式	:TRIGger:SHOLd:LIMit <type> :TRIGger:SHOLd:LIMit?

参数说明	<type>:= {LESSthan|GREATerthan|INNer|OUTer} ● LESSthan（<=）：小于时间值。 ● GREATerthan（>=）：大于时间值。 ● INNer（[--,--]）：时间值范围内。 ● OUTer（--] [--）：时间值范围外。
返回格式	{LESSthan|GREATerthan|INNer|OUTer}
示例	设置建立/保持触发的限制条件类型为小于时间值： *:TRIGger:SHOLd:LIMit LESSthan* *TRIG:SHOL:LIM LESS* 查询当前建立/保持触发的限制条件类型： *TRIG:SHOL:LIM?* 返回值： *LESSthan*
关联命令	:TRIGger:SHOLd:TLOWer :TRIGger:SHOLd:TUPPer

:TRIGger:SHOLd:TLOWer

描述	设置或查询建立/保持触发限制条件的下限值。
命令格式	:TRIGger:SHOLd:TLOWer <value> :TRIGger:SHOLd:TLOWer?
参数说明	<value>:= 浮点型 NR3 格式 **注意**： ● 下限值不能大于上限值（上限值设置指令:TRIGger:SHOLd:TUPPer）。 ● 当限制类型为“<=”时，此命令不可用。
返回格式	浮点型 NR3 格式
示例	设置建立/保持触发的下限值为 10ns： *:TRIGger:SHOLd:TLOWer 1.00E-08* *TRIG:SHOL:TLOW 1.00E-08* 查询当前建立/保持触发的下限值： *TRIG:SHOL:TLOW?* 返回值： *1.00E-08*
关联命令	:TRIGger:SHOLd:LIMit :TRIGger:SHOLd:TUPPer

:TRIGger:SHOLd:TUPPer

描述	设置或查询建立/保持触发限制条件的上限值。
命令格式	:TRIGger:SHOLd:TUPPer <value> :TRIGger:SHOLd:TUPPer?
参数说明	<value>:= 浮点型 NR3 格式 **注意**: ● 上限值不能小于下限值（下限值设置指令:TRIGger:SHOLd:TLOWer）。 ● 当限制类型为“>=”时，此命令不可用。
返回格式	浮点型 NR3 格式
示例	设置建立/保持触发的下限值为 30ns: *:TRIGger:SHOL:TUPPer 3.00E-08* *TRIG:SHOL:TUPP 3.00E-08* 查询建立/保持触发的下限值: *TRIG:SHOL:TUPP?* 返回值: *3.00E-08*
关联命令	:TRIGger:SHOLd:LIMit :TRIGger:SHOLd:TLOWer

:TRIGger:IIC

:TRIGger:IIC:ADDRess

描述	设置或查询 IIC 总线触发的地址。
命令格式	:TRIGger:IIC:ADDRess <addr> :TRIGger:IIC:ADDRess?
参数说明	<addr>:= 地址，整型 NR1 格式。该值范围为 [0,1024]，其中 1024 表示任意值 0xXX。
返回格式	整型 NR1 格式
示例	设置 IIC 总线触发的地址为 0x0a: *:TRIGger:IIC:ADDRess 10* *TRIG:IIC:ADDR 10* 查询当前 IIC 总线触发的地址: *TRIG:IIC:ADDR?* 返回值: *10*

关联命令	:TRIGger:IIC:CONDition

:TRIGger:IIC:ALENgth

描述	设置或查询 IIC 总线触发地址的长度。
命令格式	:TRIGger:IIC:ALENgth <length> :TRIGger:IIC:ALENgth?
参数说明	<length>:= {7BIT|10BIT}
返回格式	{7BIT|10BIT}
示例	设置 IIC 总线触发地址的长度为 10 bit: *:TRIGger:IIC:ALENgth 10BIT* *TRIG:IIC:ALEN 10BIT* 查询当前 IIC 总线触发地址的长度: *TRIG:IIC:ALEN?* 返回值: *10BIT*
关联命令	:TRIGger:IIC:CONDition

:TRIGger:IIC:CONDition

描述	设置或查询 IIC 总线触发的触发条件。
命令格式	:TRIGger:IIC:CONDition <condition> :TRIGger:IIC:CONDition?
参数说明	<condition>:= {STARt|STOP|RESTart|NACK|EEPRom|7ADDRess|10ADDRess|DLENgth}
返回格式	{STARt|STOP|RESTart|NACK|EEPRom|7ADDRess|10ADDRess|DLENgth}
示例	设置 IIC 总线触发的触发条件为 STOP: *:TRIGger:IIC:CONDition STOP* *TRIG:IIC:COND STOP* 查询当前 IIC 总线触发的触发条件: *TRIG:IIC:COND?* 返回值: *STOP*

:TRIGger:IIC:DAT2

描述	设置或查询 IIC 总线触发的数据 2。
命令格式	:TRIGger:IIC:DAT2 <data> :TRIGger:IIC:DAT2?
参数说明	<data>:= 数值，整型 NR1 格式。该值范围为 [0,256]，其中 256 表示为任意值 0xXX。
返回格式	整型 NR1 格式
示例	设置 IIC 总线触发的数据 2 为 0x0b： *:TRIGger:IIC:DAT2 11* *TRIG:IIC:DAT2 11* 查询当前 IIC 总线触发的数据 2： *TRIG:IIC:DAT2?* 返回值： *11*
关联命令	:TRIGger:IIC:CONDition

:TRIGger:IIC:DATA

描述	设置或查询 IIC 总线触发的数据 1。
命令格式	:TRIGger:IIC:DATA <data> :TRIGger:IIC:DATA?
参数说明	<data>:= 数值，整型 NR1 格式。该值范围为 [0,256]，其中 256 表示为任意值 0xXX。
返回格式	整型 NR1 格式
示例	设置 IIC 总线触发的数据 1 为 0x2A： *:TRIGger:IIC:DATA 42* *TRIG:IIC:DATA 42* 查询当前 IIC 总线触发的数据 1： *TRIG:IIC:DATA?* 返回值： *42*
关联命令	:TRIGger:IIC:CONDition :TRIGger:IIC:DAT2

:TRIGger:IIC:DLENgth

描述	设置或查询 IIC 总线触发的数据长度。
命令格式	:TRIGger:IIC:DLENgth <length> :TRIGger:IIC:DLENgth?
参数说明	<length>:= 整型 NR1 格式，该值范围为 [1,12]。
返回格式	整型 NR1 格式
示例	设置 IIC 总线触发的数据长度为 10： *:TRIGger:IIC:DLENgth 10* *TRIG:IIC:DLEN 10* 查询当前 IIC 总线触发的数据长度： *TRIG:IIC:DLEN?* 返回值： *10*
关联命令	:TRIGger:IIC:CONDition

:TRIGger:IIC:LIMit

描述	当 IIC 总线触发的条件为 EEPPOM 时，设置或查询限制条件类型。
命令格式	:TRIGger:IIC:LIMit <limit_type> :TRIGger:IIC:LIMit?
参数说明	<limit_type>:= {EQUal|GREaterthan|LESSthan}
返回格式	{EQUal|GREaterthan|LESSthan}
示例	设置 IIC 总线触发条件为 EEPPOM 时的限制条件类型为小于： *:TRIGger:IIC:LIMit LESSthan* *TRIG:IIC:LIM LESS* 查询当前 IIC 总线触发条件为 EEPPOM 时的限制条件类型： *TRIG:IIC:LIM?* 返回值： *LESSthan*
关联命令	:TRIGger:IIC:CONDition

:TRIGger:IIC:RWBit

描述	当 IIC 总线触发的条件为 7/10 地址&数据时，设置或查询读写位。
命令格式	:TRIGger:IIC:RWBit <type>

	:TRIGger:IIC:RWBit?
参数说明	<type>:= {WRITe|READ|ANY}
返回格式	{WRITe|READ|ANY}
示例	设置 IIC 总线触发的条件为 7/10 地址&数据时的读写位为读： *:TRIGger:IIC:RWBit READ* *TRIG:IIC:RWB READ* 查询当前 IIC 总线触发的条件为 7/10 地址&数据时的读写位： *TRIG:IIC:RWB?* 返回值： *READ*
关联命令	:TRIGger:IIC:CONDition

:TRIGger:IIC:SCLSource

描述	设置或查询 IIC 总线触发的 SCL 信源。
命令格式	:TRIGger:IIC:SCLSource <source> :TRIGger:IIC:SCLSource?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 IIC 总线触发的 SCL 信源为 C2： *:TRIGger:IIC:SCLSource C2* *TRIG:IIC:SCLS C2* 查询当前 IIC 总线触发的 SCL 信源： *TRIG:IIC:SCLS?* 返回值： *C2*
关联命令	:TRIGger:IIC:SCLThreshold :TRIGger:IIC:SDASource

:TRIGger:IIC:SCLThreshold

描述	设置或查询 IIC 总线触发的 SCL 阈值。
命令格式	:TRIGger:IIC:SCLThreshold <value> :TRIGger:IIC:SCLThreshold?

参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD — [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X — [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD — [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置 IIC 总线触发的 SCL 阈值为 1.5V： *:TRIGger:IIC:SCLThreshold 1.50E+00* *TRIG:IIC:SCLT 1.50E+00* 查询当前 IIC 总线触发的 SCL 阈值： *TRIG:IIC:SCLT?* 返回值： *1.50E+00*
关联命令	:TRIGger:IIC:SCLSource

:TRIGger:IIC:SDASource

描述	设置或查询 IIC 总线触发的 SDA 信源。
命令格式	:TRIGger:IIC:SDASource <source> :TRIGger:IIC:SDASource?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 IIC 总线触发的 SDA 信源为 C2： *:TRIGger:IIC:SDASource C2* *TRIG:IIC:SDAS C2*

	查询当前 IIC 总线触发的 SDA 信源： *TRIG:IIC:SDAS?* 返回值： *C2*
关联命令	:TRIGger:IIC:SCLSource :TRIGger:IIC:SDAThreshold

:TRIGger:IIC:SDAThreshold

描述	设置或查询 IIC 总线触发的 SDA 阈值。
命令格式	:TRIGger:IIC:SDAThreshold <value> :TRIGger:IIC:SDAThreshold?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD — [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X — [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD — [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置 IIC 总线触发的 SDA 阈值为 1.5V： *:TRIGger:IIC:SDAThreshold 1.50E+00* *TRIG:IIC:SDAT 1.50E+00* 查询当前 IIC 总线触发的 SDA 阈值： *TRIG:IIC:SDAT?* 返回值： *1.50E+00*
关联命令	:TRIGger:IIC:SDASource

:TRIGger:SPI

:TRIGger:SPI:BITorder

描述	设置或查询 SPI 总线触发的比特流格式。
命令格式	:TRIGger:SPI:BITorder <bit_order> :TRIGger:SPI:BITorder?
参数说明	<bit_order>:= {LSM|MSB} ● LSM：最低有效位在先 ● MSB：最高有效位在先
返回格式	{LSM|MSB}
示例	设置 SPI 总线触发的比特流格式为 LSB： *:TRIGger:SPI:BITorder LSB* *TRIG:SPI:BIT LSB* 查询当前 SPI 总线触发的比特流格式： *TRIG:SPI:BIT?* 返回值： *LSB*

:TRIGger:SPI:CLKSource

描述	设置或查询 SPI 总线触发的 CLK（时钟）信源。
命令格式	:TRIGger:SPI:CLKSource <source> :TRIGger:SPI:CLKSource?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 SPI 总线触发的 CLK 信源为 C2： *:TRIGger:SPI:CLKSource C2* *TRIG:SPI:CLKS C2* 查询 SPI 总线触发的 CLK 信源： *TRIG:SPI:CLKS?* 返回值： *C2*
关联命令	:TRIGger:SPI:CLKSource

:TRIGger:SPI:CLKThreshold

描述	设置或查询 SPI 总线触发的 CLK（时钟）阈值。
命令格式	:TRIGger:SPI:CLKThreshold <clk_threshold> :TRIGger:SPI:CLKThreshold?
参数说明	<clk_threshold>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD / [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X / [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD / [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置 SPI 总线触发的 CLK 阈值为 1.5V： *:TRIGger:SPI:CLKThreshold 1.50E+00* *TRIG:SPI:CLKT 1.50E+00* 查询当前 SPI 总线触发的 CLK 阈值： *TRIG:SPI:CLKT?* 返回值： *1.50E+00*
关联命令	:TRIGger:SPI:CLKSource

:TRIGger:SPI:CSSource

描述	设置或查询 SPI 总线触发的 CS（片选）信源。
命令格式	:TRIGger:SPI:CSSource <source> :TRIGger:SPI:CSSource?
参数说明	<source>:= {C<n>|D<d>}

	● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 SPI 总线触发的 CS 信源为 C2： *:TRIGger:SPI:CSSource C2* *TRIG:SPI:CSS C2* 查询当前 SPI 总线触发的 CS 信源： *TRIG:SPI:CSS?* 返回值： *C2*
关联命令	:TRIGger:SPI:CSThreshold

:TRIGger:SPI:CSThreshold

描述	设置或查询 SPI 总线触发的 CS（片选）阈值。
命令格式	:TRIGger:SPI:CSThreshold <threshold> :TRIGger:SPI:CSThreshold?
参数说明	<threshold>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 SPI 总线触发的 CS 阈值为 1.5V： *:TRIGger:SPI:CSThreshold 1.50E+00* *TRIG:SPI:CST 1.50E+00*

	查询当前 SPI 总线触发的 CS 阈值： *TRIG:SPI:CST?* 返回值： *1.50E+00*
关联命令	:TRIGger:SPI:CSSource

:TRIGger:SPI:CSTYpe

描述	设置或查询 SPI 总线触发的片选类型。
命令格式	:TRIGger:SPI:CSTYpe <type> :TRIGger:SPI:CSTYpe?
参数说明	<type>:= {NCS|CS|TIMeout[,<time>]} ● NCS：低有效。片选为低时成帧，此时需要为片选信号指定源和阈值电平。此时，片选信号需要在屏幕内有完整的下降沿才视为有效。 ● CS：高有效。片选为高时成帧，此时需要为片选信号指定源和阈值电平。此时，片选信号需要在屏幕内有完整的上升沿才视为有效。 ● TIMeout：时钟超时。此时不需要为片选信号指定源和阈值电平，但要指定<time>。此时间为示波器搜索到将要触发的数据码型前，时钟信号必须为空闲状态（不跳变）的最小时间。该设置适用于无片选信号的 SPI 总线，或通道数不够的情况（如两通道示波器）。 <time>:= 数值，浮点型 NR3 格式，该值范围为 [1.00E-07,5.00E-03]。
返回格式	{NCS|CS|TIMeout[,<time>]}
示例	设置 SPI 总线触发的片选类型为 CS： *:TRIGger:SPI:CSTYpe CS* *TRIG:SPI:CSTY CS* 查询当前 SPI 总线触发的片选类型： *TRIG:SPI:CSTY?* 返回值： *CS*

:TRIGger:SPI:DATA

描述	设置 SPI 总线触发的数据。
命令格式	:TRIGger:SPI:DATA <data>[,<data>[...[,<data>]]]
参数说明	<data>:= {0|1|X}

	注意: ● 参数的数量应与命令:TRIGger:SPI:DLENgth 设置的数据长度一致。 ● 参数按从高到低的顺序分配给每个位。
示例	当数据长度为 8 时，设置 SPI 总线触发的数据为 0x82: *:TRIGger:SPI:DATA 1,0,0,0,0,0,1,0* *TRIG:SPI:DATA 1,0,0,0,0,0,1,0*
关联命令	:TRIGger:SPI:DLENgth

:TRIGger:SPI:DLENgth

描述	设置或查询 SPI 总线触发的数据长度。
命令格式	:TRIGger:SPI:DLENgth <data_length> :TRIGger:SPI:DLENgth?
参数说明	<data_length>:= 整型 NR1 格式，该值范围为 [4,96]
返回格式	整型 NR1 格式
示例	设置 SPI 总线触发的数据长度为 10 bit: *:TRIGger:SPI:DLENgth 10* *TRIG:SPI:DLEN 10* 查询当前 SPI 总线触发的数据长度: *TRIG:SPI:DLEN?* 返回值: *10*

:TRIGger:SPI:LATChedge

描述	设置或查询 SPI 总线触发的 CLK（时钟）采样边沿。
命令格式	:TRIGger:SPI:LATChedge <slope> :TRIGger:SPI:LATChedge?
参数说明	<slope>:= {RISing|FALLing}
返回格式	{RISing|FALLing}
示例	设置 SPI 总线触发的 CLK（时钟）采样边沿为上升沿: *:TRIGger:SPI:LATChedge RISing* *TRIG:SPI:LATC RIS* 查询当前 SPI 总线触发的 CLK（时钟）采样边沿: *TRIG:SPI:LATC?*

	返回值： *RISing*

:TRIGger:SPI:MISOSource

描述	设置或查询 SPI 总线触发的 MISO 信源。
命令格式	:TRIGger:SPI:MISOSource <source> :TRIGger:SPI:MISOSource?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 SPI 总线触发的 MISO 信源为 C2： *:TRIGger:SPI:MISOSource C2* *TRIG:SPI:MISOS C2* 查询当前 SPI 总线触发的 MISO 信源： *TRIG:SPI:MISOS?* 返回值： *C2*
关联命令	:TRIGger:SPI:MISOThreshold

:TRIGger:SPI:MISOThreshold

描述	设置或查询 SPI 总线触发的 MISO 阈值。
命令格式	:TRIGger:SPI:MISOThreshold <value> :TRIGger:SPI:MISOThreshold?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

	SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	
返回格式	浮点型 NR3 格式	
示例	设置 SPI 总线触发的 MISO 阈值为 1.5V： *:TRIGger:SPI:MISOThreshold 1.50E+00* *TRIG:SPI:MISOT 1.50E+00* 查询当前 SPI 总线触发的 MISO 阈值： *TRIG:SPI:MISOT?* 返回值： *1.50E+00*	
关联命令	:TRIGger:SPI:MISOSource	

:TRIGger:SPI:MOSISource

描述	设置或查询 SPI 总线触发的 MOSI 信源。
命令格式	:TRIGger:SPI:MOSISource <source> :TRIGger:SPI:MOSISource?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 SPI 总线触发的 MOSI 信源为 C2： *:TRIGger:SPI:MOSISource C2* *TRIG:SPI:MOSIS C2* 查询当前 SPI 总线触发的 MOSI 信源： *TRIG:SPI:MOSIS?* 返回值： *C2*
关联命令	:TRIGger:SPI:MOSIThreshold

:TRIGger:SPI:MOSIThreshold

描述	设置或查询 SPI 总线触发的 MOSI 阈值。

命令格式	:TRIGger:SPI:MOSIThreshold <value> :TRIGger:SPI:MOSIThreshold?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD / [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X / [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD / [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置 SPI 总线触发的 MOSI 阈值为 1.5V： *:TRIGger:SPI:MOSIThreshold 1.50E+00* *TRIG:SPI:MOSIT 1.50E+00* 查询当前 SPI 总线触发的 MOSI 阈值： *TRIG:SPI:MOSIT?* 返回值： *1.50E+00*
关联命令	:TRIGger:SPI:MOSISource

:TRIGger:SPI:NCSSource

描述	设置或查询 SPI 总线触发的 NCS 信源。
命令格式	:TRIGger:SPI:NCSSource <source> :TRIGger:SPI:NCSSource?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 SPI 总线触发的 NCS 信源为 C2：

	:TRIGger:SPI:NCSSource D0 *:TRIG:SPI:NCSS D0* 查询当前 SPI 总线触发的 NCS 信源： *:TRIG:SPI:NCSS?* 返回值： *C2*
关联命令	:TRIGger:SPI:NCSThreshold

:TRIGger:SPI:NCSThreshold

描述	设置或查询 SPI 总线触发的 NCS 阈值。
命令格式	:TRIGger:SPI:NCSThreshold <value> :TRIGger:SPI:NCSThreshold?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD / [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X / [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD / [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置 SPI 总线触发的 NCS 阈值为 1.5V： *:TRIGger:SPI:NCSThreshold 1.50E+00* *TRIG:SPI:NCST 1.50E+00* 查询当前 SPI 总线触发的 NCS 阈值： *TRIG:SPI:NCST?* 返回值： *C2*
关联命令	:TRIGger:SPI:NCSSource

:TRIGger:SPI:TTYPe

描述	设置或查询 SPI 总线触发的触发源类型。
命令格式	:TRIGger:SPI:TTYPe <trigger_type> :TRIGger:SPI:TTYPe?
参数说明	<trigger_type>:= {MISO|MOSI}
返回格式	{MISO|MOSI}
示例	设置 SPI 总线触发的触发源类型为 MOSI： *:TRIGger:SPI:TTYPe MOSI* *TRIG:SPI:TTYP MOSI* 查询当前 SPI 总线触发的触发源类型： *TRIG:SPI:TTYP?* 返回值： *MOSI*

:TRIGger:UART

:TRIGger:UART:BAUD

描述	设置或查询 UART 总线触发的波特率。
命令格式	:TRIGger:UART:BAUD <baud> :TRIGger:UART:BAUD?
参数说明	<baud>:= {600bps|1200bps|2400bps|4800bps|9600bps|19200bps|38400bps|57600bps |115200bps|CUSTom[,<value>]} <value>:= 整型 NR1 格式，该值范围为 [300,20000000]
返回格式	整型 NR1 格式
示例	设置 UART 总线触发的波特率为 96000bps： *:TRIGger:UART:BAUD 9600bps* *TRIG:UART:BAUD 9600bps* 查询当前 UART 总线触发的波特率： *TRIG:UART:BAUD?* 返回值： *9600bps*

:TRIGger:UART:BITorder

描述	设置或查询 UART 总线触发的比特流格式。
命令格式	:TRIGger:UART:BITorder <order> :TRIGger:UART:BITorder?
参数说明	<order>:= {LSM|MSB} ● LSB：最低有效位在先 ● MSB：最高有效位在先
返回格式	{LSM|MSB}
示例	设置 UART 总线触发的比特流格式为 LSB： *:TRIGger:UART:BITorder LSB* *TRIG:UART:BIT LSB* 查询当前 UART 总线触发的比特流格式： *TRIG:UART:BIT?* 返回值： *LSB*

:TRIGger:UART:CONDition

描述	设置或查询 UART 总线触发的触发条件。
命令格式	:TRIGger:UART:CONDition <condition> :TRIGger:UART:CONDition?
参数说明	<condition>:= {STARt|STOP|DATA|ERRor} ● STARt：开始条件。在 RX/TX 上出现开始位时示波器触发。 ● STOP：停止条件。在 RX/TX 上出现停止位时示波器触发。 ● DATA：数据。可选择比较符并设置相应数据值来对信号进行触发。 ● ERRor：校验错误。示波器根据用户设置的奇偶校验类型对数据进行奇偶校验，如果校验值错误则触发。
返回格式	{STARt|STOP|DATA|ERRor}
示例	设置 UART 总线触发的触发条件为 STOP： *:TRIGger:UART:CONDition STOP* *TRIG:UART:COND STOP* 查询当前 UART 总线触发的触发条件： *TRIG:UART:COND?* 返回值： *STOP*

:TRIGger:UART:DATA

描述	设置或查询 UART 总线触发的数据。
命令格式	:TRIGger:UART:DATA <data> :TRIGger:UART:DATA?
参数说明	<data>:= 数值，整型 NR1 格式 **注意**: ● 数据的长度应与命令 ● :TRIGger:UART:DLENgth 设置的数据长度一致。 ● 如需设置为任意值 0xXX，则在最大值上加 1。如数据长度为 8 时，256 表示任意值。
返回格式	整型 NR1 格式
示例	设置 UART 总线触发的数据为 0x53: *:TRIGger:UART:DATA 83* *TRIG:UART:DATA 83* 查询当前 UART 总线触发的数据: *TRIG:UART:DATA?* 返回值: *83*
关联命令	:TRIGger:UART:CONDition :TRIGger:UART:DLENgth

:TRIGger:UART:DLENgth

描述	设置或查询 UART 总线触发的数据长度。
命令格式	:TRIGger:UART:DLENgth <value> :TRIGger:UART:DLENgth?
参数说明	< value>:= 数值，整型 NR1 格式，该值范围为 [5,8]
返回格式	整型 NR1 格式
示例	设置 UART 总线触发的数据长度为 8: *:TRIGger:UART:DLENgth 8* *TRIG:UART:DLEN 8* 查询当前 UART 总线触发的数据长度: *TRIG:UART:DLEN?* 返回值: *8*

:TRIGger:UART:IDLE

描述	设置或查询 UART 总线触发的空闲电平。
命令格式	:TRIGger:UART:IDLE <idle> :TRIGger:UART:IDLE?
参数说明	<idle>:= {LOW|HIGH} ● LOW：空闲电平为低电平 ● HIGH：空闲电平为高电平
返回格式	{LOW|HIGH}
示例	设置 UART 总线触发的空闲电平为低电平： *:TRIGger:UART:IDLE LOW* *TRIG:UART:IDLE LOW* 查询当前 UART 总线触发的空闲电平： *TRIG:UART:IDLE?* 返回值： *LOW*

:TRIGger:UART:LIMit

描述	当 UART 总线触发的条件为 DATA 时，设置或查询限制条件。
命令格式	:TRIGger:UART:LIMit <limit_type> :TRIGger:UART:LIMit?
参数说明	<limit_type>:= {EQUal|GREaterthan|LESSthan}
返回格式	{EQUal|GREaterthan|LESSthan}
示例	设置 UART 总线触发的条件为 DATA 时的限制条件为小于： *:TRIGger:UART:LIMit LESSthan* *TRIG:UART:LIM LESS* 查询当前 UART 总线触发的条件为 DATA 时的限制条件： *TRIG:UART:LIM?* 返回值： *LESSthan*
关联命令	:TRIGger:UART:CONDition

:TRIGger:UART:PARity

描述	设置或查询 UART 总线触发的奇偶校验。
命令格式	:TRIGger:UART:PARity <parity> :TRIGger:UART:PARity?
参数说明	<parity>:= {NONE|ODD|EVEN|MARK|SPACe} ● NONE：无校验 ● ODD：奇校验 ● EVEN：偶校验 ● MARK：1 校验 ● SPACe：0 校验（UART 数据为 9 位时，第 9 位可视为 1 校验或 0 校验）
返回格式	{NONE|ODD|EVEN|MARK|SPACe}
示例	设置 UART 总线触发的奇偶校验为奇校验： *:TRIGger:UART:PARity ODD* *TRIG:UART:PAR ODD* 查询当前 UART 总线触发的奇偶校验： *TRIG:UART:PAR?* 返回值： *ODD*

:TRIGger:UART:RXSource

描述	设置或查询 UART 总线触发的 RX 信源。
命令格式	:TRIGger:UART:RXSource <source> :TRIGger:UART:RXSource?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 UART 总线触发的 RX 信源为 C2： *:TRIGger:UART:RXSource C2* *TRIG:UART:RXS C2* 查询当前 UART 总线触发的 RX 信源： *TRIG:UART:RXS?* 返回值： *C2*
关联命令	:TRIGger:UART:

:TRIGger:UART:RXTHreshold

描述	设置或查询 UART 总线触发的 RX 阈值。
命令格式	:TRIGger:UART:RXTHreshold <value> :TRIGger:UART:RXTHreshold?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD — [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X — [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD — [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置 UART 总线触发的 RX 阈值为 1.5V： *:TRIGger:UART:RXTHreshold 1.50E+00* *TRIG:UART:RXTH 1.50E+00* 查询当前 UART 总线触发的 RX 阈值： *TRIG:UART:RXTH?* 返回值： *1.50E+00*
关联命令	:TRIGger:UART:RXSource

:TRIGger:UART:STOP

描述	设置或查询 UART 总线触发的停止位数。
命令格式	:TRIGger:UART:STOP <bit> :TRIGger:UART:STOP?
参数说明	<bit>:= {1|1.5|2}
返回格式	{1|1.5|2}

示例	设置 UART 总线触发的停止位数为 1： *:TRIGger:UART:STOP 1* *TRIG:UART:STOP 1* 查询当前 UART 总线触发的停止位数： *TRIG:UART:STOP?* 返回值： *1*

:TRIGger:UART:TTYPe

描述	设置或查询 UART 总线触发的触发源类型。
命令格式	:TRIGger:UART:TTYPe <trigger_type> :TRIGger:UART:TTYPe?
参数说明	<trigger_type>:= {RX|TX}
返回格式	{RX|TX}
示例	设置 UART 总线触发的触发源类型为 RX： *:TRIGger:UART:TTYPe RX* *TRIG:UART:TTYP RX* 查询当前 UART 总线触发的触发源类型： *TRIG:UART:TTYP?* 返回值： *RX*

:TRIGger:UART:TXSource

描述	设置或查询 UART 总线触发的 TX 信源。
命令格式	:TRIGger:UART:TXSource <source> :TRIGger:UART:TXSource?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 UART 总线触发的 TX 信源为 C2： *:TRIGger:UART:TXSource C2* *TRIG:UART:TXS C2* 查询当前 UART 总线触发的 TX 信源：

	TRIG:UART:TXS? 返回值： *C2*
关联命令	:TRIGger:UART:

:TRIGger:UART:TXTHreshold

描述	设置或查询 UART 总线触发的 TX 阈值。
命令格式	:TRIGger:UART:TXTHreshold <value> :TRIGger:UART:TXTHreshold?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD SDS1000X HD SDS800X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 UART 总线触发的 TX 阈值为 1.5V： *:TRIGger:UART:TXTHreshold 1.50E+00* *TRIG:UART:TXTH 1.50E+00* 查询当前 UART 总线触发的 TX 阈值： *TRIG:UART:TXTH?* 返回值： *1.50E+00*
关联命令	:TRIGger:UART:TXSource

:TRIGger:CAN

:TRIGger:CAN:BAUD

描述	设置或查询 CAN 总线触发的波特率。
命令格式	:TRIGger:CAN:BAUD <baud> :TRIGger:CAN:BAUD?
参数说明	<baud>:= {5kbps|10kbps|20kbps|50kbps|100kbps|125kbps|250kbps|500kbps|800kbps|1Mbps|CUSTom[,<value>]} <value>:= 整型 NR1 格式，该值范围为 [5000,1000000]
返回格式	整型 NR1 格式
示例	设置 CAN 总线触发的波特率为 20kbps: *:TRIGger:CAN:BAUD 20kbps* *TRIG:CAN:BAUD 20kbps* 查询当前 CAN 总线触发的波特率: *TRIG:CAN:BAUD?* 返回值: *20kbps*

:TRIGger:CAN:CONDition

描述	设置或查询 CAN 总线触发的触发条件。
命令格式	:TRIGger:CAN:CONDition <condition> :TRIGger:CAN:CONDition?
参数说明	<condition>:= {STARt|REMote|ID|ID_AND_DATA|ERRor} ● STARt：开始条件。示波器在帧开始时触发。 ● REMote：远程帧。示波器在具有指定 ID 的远程帧上触发。 ● ID：示波器在与指定 ID 匹配的数据帧上触发。 ● ID_AND_DATA：ID+数据。示波器在与指定 ID 和数据匹配的数据帧上触发。 ● ERRor：错误帧。示波器将在 CAN 信号的错误帧上触发。
返回格式	{STARt|REMote|ID|ID_AND_DATA|ERRor}
示例	设置 CAN 总线触发的触发条件为开始条件: *:TRIGger:CAN:CONDition STARt* *TRIG:CAN:COND STAR* 查询当前 CAN 总线触发的触发条件: *TRIG:CAN:COND?*

	返回值： *STARt*

:TRIGger:CAN:DAT2

描述	设置或查询 CAN 总线触发的数据 2。
命令格式	:TRIGger:CAN:DAT2 <data> :TRIGger:CAN:DAT2?
参数说明	<data>:= 数值，整型 NR1 格式。该值范围为 [0,256]，其中 256 表示为任意值 0xXX。
返回格式	整型 NR1 格式
示例	设置 CAN 总线触发的数据 2 为 0x49： *:TRIGger:CAN:DAT2 73* *TRIG:CAN:DAT2 73* 查询当前 CAN 总线触发的数据 2： *TRIG:CAN:DAT2?* 返回值： *73*
关联命令	:TRIGger:CAN:CONDition

:TRIGger:CAN:DATA

描述	设置或查询 CAN 总线触发的数据 1。
命令格式	:TRIGger:CAN:DATA <data> :TRIGger:CAN:DATA?
参数说明	<data>:= 数值，整型 NR1 格式。该值范围为 [0,256]，其中 256 表示为任意值 0xXX。
返回格式	整型 NR1 格式
示例	设置 CAN 总线触发的数据 1 为 0x43： *:TRIGger:CAN:DATA 67* *TRIG:CAN:DATA 67* 查询当前 CAN 总线触发的数据 1： *TRIG:CAN:DATA?* 返回值： *67*
关联命令	:TRIGger:CAN:CONDition

:TRIGger:CAN:ID

描述	设置或查询 CAN 总线触发的 ID。
命令格式	:TRIGger:CAN:ID <id> :TRIGger:CAN:ID?
参数说明	<id>:= 整型 NR1 格式。 当 ID 长度为 29 bit 时，该值范围为 [0,536870912]，其中 536870912 表示为任意值 0xXXXXXXXX；当 ID 长度为 11 bit 时，该值范围为 [0,2048]，其中 2048 表示为任意值 0xXXXX。
返回格式	整型 NR1 格式
示例	设置 CAN 总线触发的 ID 为 0x7819F51： *:TRIGger:CAN:ID 125935441* *TRIG:CAN:ID 125935441* 查询当前 CAN 总线触发的 ID： *TRIG:CAN:ID?* 返回值： *125935441*
关联命令	:TRIGger:CAN:CONDition

:TRIGger:CAN:IDLength

描述	设置或查询 CAN 总线触发的 ID 长度。
命令格式	:TRIGger:CAN:IDLength <id_length> :TRIGger:CAN:IDLength?
参数说明	<id_length>:= {11BITS|29BITS}
返回格式	{11BITS|29BITS}
示例	设置 CAN 总线触发的 ID 长度为 29 bit： *:TRIGger:CAN:IDLength 29BITS* *TRIG:CAN:IDL 29BITS* 查询当前 CAN 总线触发的 ID 长度： *TRIG:CAN:IDL?* 返回值： *29BITS*
关联命令	:TRIGger:CAN:CONDition

:TRIGger:CAN:SOURce

描述	设置或查询 CAN 总线触发的信源。
命令格式	:TRIGger:CAN:SOURce <source> :TRIGger:CAN:SOURce?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 CAN 总线触发的信源为 C2： *:TRIGger:CAN:SOURce C2* *TRIG:CAN:SOUR C2* 查询当前 CAN 总线触发的信源： *TRIG:CAN:SOUR?* 返回值： *C2*
关联命令	:TRIGger:CAN:THReshold

:TRIGger:CAN:THReshold

描述	设置或查询 CAN 总线触发的阈值。
命令格式	:TRIGger:CAN:THReshold <value> :TRIGger:CAN:THReshold?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

	SDS1000X HD SDS800X HD	
返回格式	浮点型 NR3 格式	
示例	设置 CAN 总线触发的阈值为 1.5V： *:TRIGger:CAN:THReshold 1.50E+00* *TRIG:CAN:THR 1.50E+00* 查询当前 CAN 总线触发的阈值： *TRIG:CAN:THR?* 返回值： *1.50E+00*	
关联命令	:TRIGger:CAN:SOURce	

:TRIGger:LIN

:TRIGger:LIN:BAUD

描述	设置或查询 LIN 总线触发的波特率。
命令格式	:TRIGger:LIN:BAUD <baud> :TRIGger:LIN:BAUD?
参数说明	<baud>:= {600bps|1200bps|2400bps|4800bps|9600bps|19200bps|CUSTom[,<value>]} <value>:= 整型 NR1 格式，该值范围为 [300,20000000]
返回格式	整型 NR1 格式
示例	设置 LIN 总线触发的波特率为 9600bps： *:TRIGger:LIN:BAUD 9600bps* *TRIG:LIN:BAUD 9600bps* 查询当前 LIN 总线触发的波特率： *TRIG:LIN:BAUD?* 返回值： *9600bps*

:TRIGger:LIN:CONDition

描述	设置或查询 LIN 总线触发的触发条件。
命令格式	:TRIGger:LIN:CONDition <condition> :TRIGger:LIN:CONDition?
参数说明	<condition>:= {BReak|ID|ID_AND_DATA|DATA_ERROR} ● BReak：间隔。示波器在帧开始时触发。 ● ID：示波器将在检测到其 ID 等于设定值的帧时触发。 ● ID_AND_DATA：ID+数据。示波器将在检测到其 ID 和数据等于设定值的帧时触发。 ● DATA_ERROR：数据错误。示波器将在 LIN 信号的错误帧上触发。
返回格式	{BReak|ID|ID_AND_DATA|DATA_ERROR}
示例	设置 LIN 总线触发的触发条件为 ID+数据： *:TRIGger:LIN:CONDition ID_AND_DATA* *TRIG:LIN:COND ID_AND_DATA* 查询当前 LIN 总线触发的触发条件： *TRIG:LIN:COND?* 返回值： *ID_AND_DATA*

:TRIGger:LIN:DAT2

描述	设置或查询 LIN 总线触发的数据 2。
命令格式	:TRIGger:LIN:DAT2 <data> :TRIGger:LIN:DAT2?
参数说明	<data>:= 数值，整型 NR1 格式。该值范围为 [0,256]，其中 256 表示为任意值 0xXX。
返回格式	整型 NR1 格式
示例	设置 LIN 总线触发的数据 2 为 0x4C： *:TRIGger:LIN:DAT2 76* *TRIG:LIN:DAT2 76* 查询当前 LIN 总线触发的数据 2： *TRIG:LIN:DAT2?* 返回值： *76*
关联命令	:TRIGger:LIN:CONDition :TRIGger:LIN:DATA

:TRIGger:LIN:DATA

描述	设置或查询 LIN 总线触发的数据 1。
命令格式	:TRIGger:LIN:DATA <data> :TRIGger:LIN:DATA?
参数说明	<data>:= 数值，整型 NR1 格式。该值范围为 [0,256]，其中 256 表示为任意值 0xXX。
返回格式	整型 NR1 格式
示例	设置 LIN 总线触发的数据 1 为 0x45： *:TRIGger:LIN:DATA 69* *TRIG:LIN:DATA 69* 查询当前 LIN 总线触发的数据 1： *TRIG:LIN:DATA?* 返回值： *69*
关联命令	:TRIGger:LIN:CONDition :TRIGger:LIN:DAT2

:TRIGger:LIN:ERRor:CHECksum

描述	当 LIN 总线触发的触发条件为错误时，设置或查询校验和检测状态。
命令格式	:TRIGger:LIN:ERRor:CHECksum <state> :TRIGger:LIN:ERRor:CHECksum?
参数说明	<state>:= {0|1} ● 0：关闭校验和检测 ● 1：开启校验和检测
返回格式	{0|1}
示例	设置当 LIN 总线触发的触发条件为错误时，开启校验和检测： *:TRIGger:LIN:ERRor:CHECksum 1* *TRIG:LIN:ERR:CHEC 1* 查询当前检验和检测状态： *TRIG:LIN:ERR:CHEC?* 返回值： *1*
关联命令	:TRIGger:LIN:CONDition

:TRIGger:LIN:ERRor:DLENgth

描述	当 LIN 总线触发的触发条件为错误且开启校验和检测时，设置或查询错误帧的数据长度。
命令格式	:TRIGger:LIN:ERRor:DLENgth <length> :TRIGger:LIN:ERRor:DLENgth?
参数说明	<length>:= 数值，整型 NR1 格式，该值范围为 [1,8]。
返回格式	整型 NR1 格式
示例	设置 LIN 总线触发错误帧的数据长度为 4 比特： *:TRIGger:LIN:ERRor:DLENgth 4* *TRIG:LIN:ERR:DLEN 4* 查询当前 LIN 总线触发错误帧的数据长度： *TRIG:LIN:ERR:DLEN?* 返回值： *4*
关联命令	:TRIGger:LIN:CONDition :TRIGger:LIN:ERRor:CHECksum

:TRIGger:LIN:ERRor:ID

描述	当 LIN 总线触发的触发条件为错误且检验和检测时，设置或查询错误帧的 ID。
命令格式	:TRIGger:LIN:ERRor:ID <id> :TRIGger:LIN:ERRor:ID?
参数说明	<id>:= 整型 NR1 格式，该值范围为 [0,63]。
返回格式	整型 NR1 格式
示例	设置 LIN 总线触发错误帧的 ID 为 0x2A： *:TRIGger:LIN:ERRor:ID 42* *TRIG:LIN:ERR:ID 42* 查询当前 LIN 总线触发错误帧的 ID： *TRIG:LIN:ERR:ID?* 返回值： *42*
关联命令	:TRIGger:LIN:CONDition :TRIGger:LIN:ERRor:CHECksum

:TRIGger:LIN:ERRor:PARity

描述	当 LIN 总线触发的触发条件为错误时，设置或查询报头奇偶校验的状态。
命令格式	:TRIGger:LIN:ERRor:PARity <state> :TRIGger:LIN:ERRor:PARity?
参数说明	<state>:= {0|1} ● 0：关闭报头奇偶校验 ● 1：开启报头奇偶校验
返回格式	{0|1}
示例	设置当 LIN 总线触发的触发条件为错误时，开启报头奇偶校验： *:TRIGger:LIN:ERRor:PARity 1* *TRIG:LIN:ERR:PAR 1* 查询当前报头奇偶校验状态： *TRIG:LIN:ERR:PAR?* 返回值： *1*
关联命令	:TRIGger:LIN:CONDition

:TRIGger:LIN:ERRor:SYNC

描述	当 LIN 总线触发的触发条件为错误时，设置或查询 LIN 总线触发的同步字节检测状态。
命令格式	:TRIGger:LIN:ERRor:SYNC <state> :TRIGger:LIN:ERRor:SYNC?
参数说明	<state>:= {0|1}
返回格式	{0|1}
示例	设置开启 LIN 总线触发的同步字节检测： *:TRIGger:LIN:ERRor:SYNC 1* *TRIG:LIN:ERR:SYNC 1* 查询当前 LIN 总线触发的同步字节检测状态： *TRIG:LIN:ERR:SYNC?* 返回值： *1*
关联命令	:TRIGger:LIN:CONDition

:TRIGger:LIN:ID

描述	当 LIN 总线触发的触发条件为 ID 时，设置或查询 ID。
命令格式	:TRIGger:LIN:ID <id> :TRIGger:LIN:ID?
参数说明	<id>:= 整型 NR1 格式。该值范围为 [0,64]，其中 64 表示为任意值 0xXX。
返回格式	整型 NR1 格式
示例	设置当 LIN 总线触发的触发条件为 ID 时的 ID 为 0x2B： *:TRIGger:LIN:ID 43* *TRIG:LIN:ID 43* 查询当前 ID： *TRIG:LIN:ID?* 返回值： *43*
关联命令	:TRIGger:LIN:CONDition

:TRIGger:LIN:SOURce

描述	设置或查询 LIN 总线触发的信源。
命令格式	:TRIGger:LIN:SOURce <source> :TRIGger:LIN:SOURce?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 LIN 总线触发的信源为 C2： *:TRIGger:LIN:SOURce C2* *TRIG:LIN:SOUR C2* 查询当前 LIN 总线触发的信源： *TRIG:LIN:SOUR?* 返回值： *C2*
关联命令	:TRIGger:LIN:THReshold

:TRIGger:LIN:STANdard

描述	当 LIN 总线触发的触发条件为错误且开启数据校验时，设置或查询 LIN 协议版本。
命令格式	:TRIGger:LIN:STANdard <version> :TRIGger:LIN:STANdard?
参数说明	<version>:= {0|1} ● 0：Rev1.3 ● 1：Rev2.x
返回格式	{0|1}
示例	设置当 LIN 总线触发的触发条件为错误且数据校验时 LIN 协议版本为 1.3： *:TRIGger:LIN:STANdard 0* *TRIG:LIN:STAN 0* 查询当前 LIN 总线触发的 LIN 协议标准： *TRIG:LIN:STAN?* 返回值： *0*
关联命令	:TRIGger:LIN:CONDition :TRIGger:LIN:THReshold

:TRIGger:LIN:THReshold

描述	设置或查询 LIN 总线触发的阈值。
命令格式	:TRIGger:LIN:THReshold <value> :TRIGger:LIN:THReshold?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

	SDS2000X HD SDS1000X HD SDS800X HD	
返回格式	浮点型 NR3 格式	
示例	设置 LIN 总线触发的阈值为 1.5V： *:TRIGger:LIN:THReshold 1.50E+00* *TRIG:LIN:THR 1.50E+00* 查询当前 LIN 总线触发的阈值： *TRIG:LIN:THR?* 返回值： *1.50E+00*	
关联命令	:TRIGger:LIN:SOURce	

:TRIGger:FLEXray（选配）

:TRIGger:FLEXray:BAUD

描述	设置或查询 FlexRay 总线触发的波特率。
命令格式	:TRIGger:FLEXray:BAUD <baud> :TRIGger:FLEXray:BAUD?
参数说明	<baud>:= {2500kbps|5Mbps|10Mbps|CUSTom[,<value>]} <value>:= 整型 NR1 格式，该值范围为 [1000000,20000000]
返回格式	整型 NR1 格式
示例	设置 FlexRay 总线触发的波特率为 2500kbps： *:TRIGger:FLEXray:BAUD 2500kbps* *TRIG:FLEX:BAUD 2500kbps* 查询当前 FlexRay 总线触发的波特率： *TRIG:FLEX:BAUD?* 返回值： *2500kbps*

:TRIGger:FLEXray:CONDition

描述	设置或查询 FlexRay 总线触发的触发条件。
命令格式	:TRIGger:FLEXray:CONDition <condition> :TRIGger:FLEXray:CONDition?
参数说明	<condition>:= {TSS|FRAMe|SYMBol|ERRor} ● TSS：起始。示波器在传输起始序列上触发。 ● FRAMe：帧。示波器将在总线的帧上触发。 ● SYMBol：符号。示波器将在 CID（信道空闲界定符）、CAS/MTS（冲突避免符/媒体访问测试符）和 WUS（唤醒特征符）上触发。 ● ERRor：错误。示波器将在总线错误时触发，包括帧起始错误、字节起始错误、帧结束错误、头部 CRC 错误、数据 CRC 错误。
返回格式	{TSS|FRAMe|SYMBol|ERRor}
示例	设置 FlexRay 总线触发的触发条件为 SYMBol： *:TRIGger:FLEXray:CONDition SYMBol* *TRIG:FLEX:COND SYMB* 查询当前 FlexRay 总线触发的触发条件： *TRIG:FLEX:COND?* 返回值： *SYMBol*

:TRIGger:FLEXray:FRAMe:COMPare

描述	设置或查询 FlexRay 总线触发的帧比较类型。
命令格式	:TRIGger:FLEXray:FRAMe:COMPare <type> :TRIGger:FLEXray:FRAMe:COMPare?
参数说明	<type >:= {ANY|EQUal|GREaterthan|LESSthan}
返回格式	{ANY|EQUal|GREaterthan|LESSthan}
示例	设置 FlexRay 总线触发的帧比较类型为小于： *:TRIGger:FLEXray:FRAMe:COMPare LESSthan* *TRIG:FLEX:FRAM:COMP LESS* 查询当前 FlexRay 总线触发的帧比较类型： *TRIG:FLEX:FRAM:COMP?* 返回值： *LESSthan*
关联命令	:TRIGger:FLEXray:CONDition

:TRIGger:FLEXray:FRAMe:CYCLe

描述	设置或查询 FlexRay 总线触发的周期数量。
命令格式	:TRIGger:FLEXray:FRAMe:CYCLe <cycle> :TRIGger:FLEXray:FRAMe:CYCLe?
参数说明	<cycle>:= 周期数，整型 NR1 格式，该值范围为 [0,63]
返回格式	整型 NR1 格式
示例	设置 FlexRay 总线触发的周期数量为 2： *:TRIGger:FLEXray:FRAMe:CYCLe 2* *TRIG:FLEX:FRAM:CYCL 2* 查询当前 FlexRay 总线触发的周期数量： *TRIG:FLEX:FRAM:CYCL?* 返回值： *2*
关联命令	:TRIGger:FLEXray:CONDition

:TRIGger:FLEXray:FRAMe:ID

描述	设置或查询 FlexRay 总线触发的 ID。
命令格式	:TRIGger:FLEXray:FRAMe:ID <id> :TRIGger:FLEXray:FRAMe:ID?
参数说明	<id>:= 数值，整型 NR1 格式。该值范围为 [0,2048]，其中 2048 表示为任意值 0xXXX。
返回格式	整型 NR1 格式
示例	设置 FlexRay 总线触发的 ID 为 0x701： *:TRIGger:FLEXray:FRAMe:ID 1793* *TRIG:FLEX:FRAM:ID 1793* 查询当前 FlexRay 总线触发的 ID： *TRIG:FLEX:FRAM:ID?* 返回值： *1973*
关联命令	:TRIGger:FLEXray:CONDition

:TRIGger:FLEXray:FRAMe:REPetition

描述	设置或查询 FlexRay 总线触发的重复系数。
命令格式	:TRIGger:FLEXray:FRAMe:REPetition <times> :TRIGger:FLEXray:FRAMe:REPetition?
参数说明	<times>:= {1|2|4|8|16|32|64}
返回格式	{1|2|4|8|16|32|64}
示例	设置 FlexRay 总线触发的重复系数为 8： *:TRIGger:FLEXray:FRAMe:REPetition 8* *TRIG:FLEX:FRAM:REP 8* 查询当前 FlexRay 总线触发的重复系数： *TRIG:FLEX:FRAM:REP?* 返回值： *8*
关联命令	:TRIGger:FLEXray:CONDition :TRIGger:FLEXray:FRAMe:COMPare

:TRIGger:FLEXray:SOURce

描述	设置或查询 FlexRay 总线触发的信源。
命令格式	:TRIGger:FLEXray:SOURce <source> :TRIGger:FLEXray:SOURce?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 FlexRay 总线触发的信源为 C2： *:TRIGger:FLEXray:SOURce C2* *TRIG:FLEX:SOUR C2* 查询当前 FlexRay 总线触发的信源： *TRIG:FLEX:SOUR?* 返回值： *C2*
关联命令	:TRIGger:FLEXray:THReshold

:TRIGger:FLEXray:THReshold

描述	设置或查询 FlexRay 总线触发的阈值。
命令格式	:TRIGger:FLEXray:THReshold <value> :TRIGger:FLEXray:THReshold?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表： **机型** / **范围** SDS7000A SDS5000X HD — [-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移] SDS6000 Pro SDS6000A SDS6000L — [-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移] SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD — [-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]
返回格式	浮点型 NR3 格式
示例	设置 FlexRay 总线触发的阈值为 1.5V： *:TRIGger:FLEXray:THReshold 1.50E+00* *TRIG:FLEX:THR 1.50E+00* 查询当前 FlexRay 总线触发的阈值： *TRIG:FLEX:THR?* 返回值： *1.50E+00*
关联命令	:TRIGger:FLEXray:SOURce

:TRIGger:CANFd（选配）

:TRIGger:CANFd:BAUDData

描述	设置或查询 CAN FD 总线触发的数据波特率。
命令格式	:TRIGger:CANFd:BAUDData <baud> :TRIGger:CANFd:BAUDData?
参数说明	<baud>:= {500kbps|1Mbps|2Mbps|5Mbps|8Mbps|10Mbps|CUSTom[,<value>]} <value>:= 整型 NR1 格式，该值范围为 [100000,10000000]
返回格式	整型 NR1 格式

示例	设置 CAN FD 总线触发的数据波特率为 500kbps： *:TRIGger:CANFd:BAUDData 500kbps* *TRIG:CANF:BAUDD 500kbps* 查询当前 CAN FD 总线触发的数据波特率： *TRIG:CANF:BAUDD?* 返回值： *500kbps*
关联命令	:TRIGger:CANFd:FTYPe :TRIGger:CANFd:BAUDNominal

:TRIGger:CANFd:BAUDNominal

描述	设置或查询 CAN FD 总线触发的标准波特率。
命令格式	:TRIGger:CANFd:BAUDNominal <baud> :TRIGger:CANFd:BAUDNominal?
参数说明	<baud>:= {10kbps|25kbps|50kbps|100kbps|250kbps|1Mbps|CUSTom[,<value>]} <value>:= 整型 NR1 格式，该值范围为 [10000,1000000]
返回格式	整型 NR1 格式
示例	设置 CAN FD 总线触发的标准波特率为 10kbps： *:TRIGger:CANFd:BAUDNominal 10kbps* *TRIG:CANF:BAUDN 10kbps* 查询当前 CAN FD 总线触发的标准波特率： *TRIG:CANF:BAUDN?* 返回值： *10kbps*
关联命令	:TRIGger:CANFd:FTYPe :TRIGger:CANFd:BAUDData

:TRIGger:CANFd:CONDition

描述	设置或查询 CAN FD 总线触发的触发条件。
命令格式	:TRIGger:CANFd:CONDition <condition> :TRIGger:CANFd:CONDition?
参数说明	<condition>:= {STARt|REMote|ID|ID_AND_DATA|ERRor}
返回格式	{STARt|REMote|ID|ID_AND_DATA|ERRor}

	● STARt：开始条件。示波器在帧开始时触发。 ● REMote：远程帧。示波器在具有指定 ID 的远程帧上触发。 ● ID：示波器在与指定 ID 匹配的数据帧上触发。 ● ID_AND_DATA：ID+数据。示波器在与指定 ID 和数据匹配的数据帧上触发。 ● ERRor：错误帧。示波器将在 CAN FD 信号的错误帧上触发。
示例	设置 CAN FD 总线触发的触发条件为 ID+数据： *:TRIGger:CANFd:CONDition ID_AND_DATA* *TRIG:CANF:COND ID_AND_DATA* 查询当前 CAN FD 总线触发的触发条件： *TRIG:CANF:COND?* 返回值： *ID_AND_DATA*

:TRIGger:CANFd:DAT2

描述	设置或查询 CAN FD 总线触发的数据 2。
命令格式	:TRIGger:CANFd:DAT2 <data> :TRIGger:CANFd:DAT2?
参数说明	<data>:= 数值，整型 NR1 格式。该值范围为 [0,256]，其中 256 表示为任意值 0xXX。
返回格式	整型 NR1 格式
示例	设置 CAN FD 总线触发的数据 2 为 0x3F： *TRIGger:CANFd:DAT2 63* *TRIG:CANF:DAT2 63* 查询当前 CAN FD 总线触发的数据 2： *TRIG:CANF:DAT2?* 返回值： *63*
关联命令	:TRIGger:CANFd:CONDition :TRIGger:CANFd:DATA

:TRIGger:CANFd:DATA

描述	设置或查询 CAN FD 总线触发的数据 1。
命令格式	:TRIGger:CANFd:DATA <data> :TRIGger:CANFd:DATA?

参数说明	<data>:= 数值，整型 NR1 格式。该值范围为 [0,256]，其中 256 表示为任意值 0xXX。
返回格式	整型 NR1 格式
示例	设置 CAN FD 总线触发的数据 1 为 0x2E: *:TRIGger:CANFd:DATA 46* *TRIG:CANF:DATA 46* 查询当前 CAN FD 总线触发的数据 1: *TRIG:CANF:DATA?* 返回值: *46*
关联命令	:TRIGger:CANFd:CONDition :TRIGger:CANFd:DAT2

:TRIGger:CANFd:FTYPe

描述	设置或查询 CAN FD 总线触发的帧类型。
命令格式	:TRIGger:CANFd:FTYPe <frame_type> :TRIGger:CANFd:FTYPe?
参数说明	<frame_type>:= {BOTH|CAN|CANFd}
返回格式	{BOTH|CAN|CANFd}
示例	设置 CAN FD 总线触发的帧类型为 CANFd: *:TRIGger:CANFd:FTYPe CANFd* *TRIG:CANF:FTYP CANF* 查询当前 CAN FD 总线触发的帧类型: *TRIG:CANF:FTYP?* 返回值: *CANFd*

:TRIGger:CANFd:ID

描述	当 CAN FD 总线触发的触发条件远程帧、ID 或 ID+数据时，设置或查询 ID。
命令格式	:TRIGger:CANFd:ID <id> :TRIGger:CANFd:ID?
参数说明	<id>:= 数值，整型 NR1 格式。 当 ID 长度为 29 bit 时，该值范围为 [0,536870912]，其中 536870912 表示为任意值 0xXXXXXXXX；当 ID 长度为 11 bit 时，该值范围为 [0,2048]，其中 2048 表示为任意值 0xXXXX。

返回格式	整型 NR1 格式
示例	设置 CAN FD 总线触发的 ID 为 0x56A710C： *:TRIGger:CANFd:ID 90861836* *TRIG:CANF:ID 90861836* 查询当前 CAN FD 总线触发的 ID： *TRIG:CANF:ID?* 返回值： *90861836*
关联命令	:TRIGger:CANFd:CONDition :TRIGger:CANFd:IDLength

:TRIGger:CANFd:IDLength

描述	设置或查询 CAN FD 总线触发的 ID 长度。
命令格式	:TRIGger:CANFd:IDLength <length> :TRIGger:CANFd:IDLength?
参数说明	<length>:= {11BITS|29BITS}
返回格式	{11BITS|29BITS}
示例	设置 CAN FD 总线触发的 ID 长度为 29 bit： *:TRIGger:CANFd:IDLength 29BITS* *TRIG:CANF:IDL 29BITS* 查询当前 CAN FD 总线触发的 ID 长度： *TRIG:CANF:IDL?* 返回值： *29BITS*
关联命令	:TRIGger:CANFd:CONDition :TRIGger:CANFd:ID

:TRIGger:CANFd:SOURce

描述	设置或查询 CAN FD 总线触发的信源。
命令格式	:TRIGger:CANFd:SOURce <source> :TRIGger:CANFd:SOURce?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式

返回格式	{C<n>ID<d>}
示例	设置 CAN FD 总线触发的信源为 C2： *:TRIGger:CANFd:SOURce C2* *TRIG:CANF:SOUR C2* 查询当前 CAN FD 总线触发的信源： *TRIG:CANF:SOUR?* 返回值： *C2*
关联命令	:TRIGger:CANFd:THReshold

:TRIGger:CANFd:THReshold

描述	设置或查询 CAN FD 总线触发的阈值。
命令格式	:TRIGger:CANFd:THReshold <threshold> :TRIGger:CANFd:THReshold?
参数说明	<threshold>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 CAN FD 总线触发的阈值为 1.5V： *:TRIGger:CANFd:THReshold 1.50E+00* *TRIG:CANF:THR 1.50E+00* 查询当前 CAN FD 总线触发的阈值： *TRIG:CANF:THR?* 返回值： *1.50E+00*
关联命令	:TRIGger:CANFd:SOURce

:TRIGger:IIS（选配）

:TRIGger:IIS:AVARiant

描述	设置或查询 IIS 总线触发的音频格式。
命令格式	:TRIGger:IIS:AVARiant <type> :TRIGger:IIS:AVARiant?
参数说明	<type>:= {I2S|LJ|RJ}
返回格式	{I2S|LJ|RJ}
示例	设置 IIS 总线触发的音频格式为 I2S： *:TRIGger:IIS:AVARiant I2S* *TRIG:IIS:AVAR I2S* 查询当前 IIS 总线触发的音频格式： *TRIG:IIS:AVAR?* 返回值： *I2S*

:TRIGger:IIS:BCLKSource

描述	设置或查询 IIS 总线触发的时钟源（BCLK）。
命令格式	:TRIGger:IIS:BCLKSource <source> :TRIGger:IIS:BCLKSource?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 IIS 总线触发的时钟源为 C2： *:TRIGger:IIS:BCLKSource C2* *TRIG:IIS:BCLKS C2* 查询当前 IIS 总线触发的时钟源： *TRIG:IIS:BCLKS?* 返回值： *C2*
关联命令	:TRIGger:IIS:BCLKThreshold

:TRIGger:IIS:BCLKThreshold

描述	设置或查询 IIS 总线触发的时钟阈值。
命令格式	:TRIGger:IIS:BCLKThreshold <value> :TRIGger:IIS:BCLKThreshold?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 IIS 总线触发的时钟阈值为 1.5V： *:TRIGger:IIS:BCLKThreshold 1.50E+00* *TRIG:IIS:BCLKT 1.50+00* 查询当前 IIS 总线触发的时钟阈值： *TRIG:IIS:BCLKT?* 返回值： *1.50E+00*
关联命令	:TRIGger:IIS:BCLKSource

:TRIGger:IIS:BITorder

描述	设置或查询 IIS 总线触发的比特流格式。
命令格式	:TRIGger:IIS:BITorder <order> :TRIGger:IIS:BITorder?
参数说明	<order>:= {LSM|MSB}
返回格式	{LSM|MSB}
示例	设置 IIS 总线触发的比特流格式为 MSB： *:TRIGger:IIS:BITorder MSB* *TRIG:IIS:BIT MSB*

	查询当前 IIS 总线触发的比特流格式： *TRIG:IIS:BIT?* 返回值： *MSB*

:TRIGger:IIS:CHANnel

描述	设置或查询 IIS 总线触发的通道。
命令格式	:TRIGger:IIS:CHANnel <channel> :TRIGger:IIS:CHANnel?
参数说明	<channel>:= {LEFT|RIGHT}
返回格式	{LEFT|RIGHT}
示例	设置 IIS 总线触发的通道为 RIGHT： *:TRIGger:IIS:CHANnel RIGHT* *TRIG:IIS:CHAN RIGHT* 查询当前 IIS 总线触发的通道： *TRIG:IIS:CHAN?* 返回值： *RIGHT*

:TRIGger:IIS:COMPare

描述	设置或查询 IIS 总线触发的比较类型。
命令格式	:TRIGger:IIS:COMPare <type> :TRIGger:IIS:COMPare?
参数说明	<type>:= {EQUal|GREaterthan|LESSthan}
返回格式	{EQUal|GREaterthan|LESSthan}
示例	设置 IIS 总线触发的比较类型为“<”： *:TRIGger:IIS:COMPare LESSthan* *TRIG:IIS:COMP LESS* 查询当前 IIS 总线触发的比较类型： *TRIG:IIS:COMP?* 返回值： *LESSthan*
关联命令	:TRIGger:IIS:CONDition

:TRIGger:IIS:CONDition

描述	设置或查询 IIS 总线触发的触发条件。
命令格式	:TRIGger:IIS:CONDition <condition> :TRIGger:IIS:CONDition?
参数说明	<condition>:= {DATA|MUTE|CLIP|GLITch|RISing|FALLing} ● DATA：数据。可选择比较符并设置相应数据值来对信号进行触发。 ● MUTE：对静音信号进行触发。音量小于设定值且持续时间达到设定值的信号，即为静音。 ● CLIP：对破音信号进行触发。音量大于设定值且持续时间达到设定值的信号，即为破音。 ● GLITch：毛刺。对音频信号的毛刺进行触发。 ● RISing：上升沿。在大于门限设置值的信号上触发。 ● FALLing：下降沿。在小于门限设置值的信号上触发。
返回格式	{DATA|MUTE|CLIP|GLITch|RISing|FALLing}
示例	设置 IIS 总线触发的触发条件为数据： *:TRIGger:IIS:CONDition DATA* *TRIG:IIS:COND DATA* 查询当前 IIS 总线触发的触发条件： *TRIG:IIS:COND?* 返回值： *DATA*

:TRIGger:IIS:DLENgth

描述	设置或查询 IIS 总线触发的数据位数。
命令格式	:TRIGger:IIS:DLENgth <value> :TRIGger:IIS:DLENgth?
参数说明	<value>:= 数值，整型 NR1 格式 **注意**：该值的范围与通道位数和起始位有关。如果通道位数为 32，起始位为 2，则范围为 [1,30]。
返回格式	整型 NR1 格式
示例	设置 IIS 总线触发的数据位数为 10： *:TRIGger:IIS:DLENgth 10* *TRIG:IIS:DLEN 10* 查询当前 IIS 总线触发的数据位数： *TRIG:IIS:DLEN?*

	返回值： *10*

:TRIGger:IIS:DSource

描述	设置或查询 IIS 总线触发的数据源。
命令格式	:TRIGger:IIS:DSource <source> :TRIGger:IIS:DSource?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 IIS 总线触发的数据源为 C2： *:TRIGger:IIS:DSource C2* *TRIG:IIS:DS C2* 查询当前 IIS 总线触发的数据源： *TRIG:IIS:DS?* 返回值： *C2*
关联命令	:TRIGger:IIS:DTHReshold

:TRIGger:IIS:DTHReshold

描述	设置或查询 IIS 总线触发的数据源阈值。
命令格式	:TRIGger:IIS:DTHReshold <value> :TRIGger:IIS:DTHReshold?
参数说明	<value>:= 阈值电平，浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

	SDS2000X Plus SDS2000X HD
返回格式	浮点型 NR3 格式
示例	设置 IIS 总线触发的数据源阈值为 1.5V： *:TRIGger:IIS:DTHReshold 1.50E+00* *TRIG:IIS:DTHR 1.50E+00* 查询当前 IIS 总线触发的数据源阈值： *TRIG:IIS:DTHR?* 返回值： *1.50E+00*
关联命令	:TRIGger:IIS:

:TRIGger:IIS:LATChedge

描述	设置或查询 IIS 总线触发的时钟采样边沿。
命令格式	:TRIGger:IIS:LATChedge <slope> :TRIGger:IIS:LATChedge?
参数说明	<slope>:= {RISing| FALLing } ● RISing：上升沿。数据在时钟上升沿锁存 ● FALLing：下降沿。数据在时钟下降沿锁存
返回格式	{RISing|FALLing}
示例	设置 IIS 总线触发的时钟采样边沿为上升沿： *:TRIGger:IIS:LATChegde RISing* *TRIG:IIS:LATC RIS* 查询当前 IIS 总线触发的时钟采样边沿： *TRIG:IIS:LATC?* 返回值： *RISing*

:TRIGger:IIS:LCH

描述	设置或查询 IIS 总线触发的左声道极性。
命令格式	:TRIGger:IIS:LCH <level> :TRIGger:IIS:LCH?
参数说明	<level>:= {LOW|HIGH} ● LOW：低电平。WS 为低时选择左声道，WS 为高时选择右声道

	● HIGH：高电平。WS 为低时选择右声道，WS 为高时选择左声道
返回格式	{LOW|HIGH}
示例	设置 IIS 总线触发的左声道极性为高电平： *:TRIGger:IIS:LCH HIGH* *TRIG:IIS:LCH HIGH* 查询当前 IIS 总线触发的左声道极性： *TRIG:IIS:LCH?* 返回值： *HIGH*

:TRIGger:IIS:VALue

描述	设置或查询 IIS 总线触发的数据值。
命令格式	:TRIGger:IIS:VALue <value> :TRIGger:IIS:VALue?
参数说明	<value>:= 数值，整型 NR1 格式 **注意**：如果数据值选择为 256（数据长度为 8），则数值将被忽略。
返回格式	整型 NR1 格式
示例	设置 IIS 总线触发的数据值为 0x56（数据长度为 8 时）： *:TRIGger:IIS:VALue 86* *TRIG:IIS:VAL 86* 查询当前 IIS 总线触发的数据值： *TRIG:IIS:VAL?* 返回值： *86*
关联命令	:TRIGger:IIS:CONDition :TRIGger:IIS:DLENgth

:TRIGger:IIS:WSSource

描述	设置或查询 IIS 总线触发的帧时钟（WS）源。
命令格式	:TRIGger:IIS:WSSource <source> :TRIGger:IIS:WSSource?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式

	● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>ID<d>}
示例	设置 IIS 总线触发的帧时钟源为 C2： *:TRIGger:IIS:WSSource C2* *TRIG:IIS:WSS C2* 查询当前 IIS 总线触发的帧时钟源： *TRIG:IIS:WSS?* 返回值： *C2*
关联命令	:TRIGger:IIS:WSTHreshold

:TRIGger:IIS:WSTHreshold

描述	设置或查询 IIS 总线触发的帧时钟（WS）阈值。
命令格式	:TRIGger:IIS:WSTHreshold <value> :TRIGger:IIS:WSTHreshold?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus SDS2000X HD	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

返回格式	浮点型 NR3 格式
示例	设置 IIS 总线触发的帧时钟阈值为 1.5V： *:TRIGger:IIS:WSTHreshold 1.50E+00* *TRIG:IIS:WSTH 1.50E+00* 查询当前 IIS 总线触发的帧时钟阈值： *TRIG:IIS:WSTH?* 返回值： *1.50E+00*
关联命令	:TRIGger:IIS:WSSource

:TRIGger:SENT（选配）

:TRIGger:SENT:SOURce

描述	设置或查询 SENT 总线触发的信源。
命令格式	:TRIGger:SENT:SOURce <source> :TRIGger:SENT:SOURce?
参数说明	<source>:= {C<n>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|D<d>}
示例	设置 SENT 总线触发的信源为 C2： *:TRIGger:SENT:SOURce C2* *TRIG:SENT:SOUR C2* 查询当前 SENT 总线触发的信源： *TRIG:SENT:SOUR?* 返回值： *C2*
关联命令	:TRIGger:SENT:THReshold

:TRIGger:SENT:THReshold

描述	设置或查询 SENT 总线触发的阈值。
命令格式	:TRIGger:SENT:THReshold <value> :TRIGger:SENT:THReshold?
参数说明	<value>:= 浮点型 NR3 格式。范围因机型而异，有关详细信息，请参见下表：

机型	范围
SDS7000A SDS5000X HD	[-4.26*垂直档位-垂直偏移, 4.26*垂直档位-垂直偏移]
SDS6000 Pro SDS6000A SDS6000L SHS800X SHS1000X	[-4.5*垂直档位-垂直偏移, 4.5*垂直档位-垂直偏移]
SDS5000X SDS3000X HD SDS2000X Plus	[-4.1*垂直档位-垂直偏移, 4.1*垂直档位-垂直偏移]

	SDS2000X HD	
返回格式	浮点型 NR3 格式	
示例	设置 SENT 总线触发的阈值为 1.5V： *:TRIGger:SENT:THReshold 1.50E+00* *TRIG:SENT:THR 1.50E+00* 查询当前 SENT 总线触发的阈值： *TRIG:SENT:THR?* 返回值： *1.50E+00*	
关联命令	:TRIGger:SENT:SOURce	

5.29 WAVeform 命令系统

:WAVeform 命令子系统用于读取波形数据及其相关设置。波形记录实际上包含在两个部分中：波形参数和波形数据，上位机必须使用两个单独的命令:WAVeform:PREamble 和:WAVeform:DATA 从示波器读取。波形数据是针对指定源中的每个点读取的实际数据。波形参数包含用于解释波形数据的信息，该信息包括所读取点的数量、所读取的数据的格式和所读取数据的类型。波形参数还包含档位、偏移等信息，因此字和字节数据可以转换为时间值和电压值。

:WAVeform:ACCelerate

描述	设置或查询波形预读数缓存开关状态。预读数缓存开启后，将从缓存中读取数据，提高波形读取速度，最大程度避免丢失触发事件。 **注意**：由于时序问题波形读取可能不是最新一帧，且影响波形刷新速率。
命令格式	:WAVeform:ACCelerate <state> :WAVeform:ACCelerate?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}
示例	设置波形预读数缓存开启： *:WAVeform:ACCelerate ON* *WAV:ACC ON* 查询波形预读数缓存开关状态： *WAV:ACC?* 返回值： *ON*

:WAVeform:BYTeorder

描述	设置或查询传输波形数据字的字节顺序。当波形数据按 16bit 传输时，需要设置字节顺序，默认为 LSB。
命令格式	:WAVeform:BYTeorder <order> :WAVeform:BYTeorder?
参数说明	<order>:= {MSB|LSB} ● MSB：高位优先 ● LSB：低位优先
返回格式	{MSB|LSB}

示例	设置示波器传输数据字的字节顺序为 MSB： *:WAVeform:BYTeorder MSB* *WAV:BYT MSB* 查询要从示波器传输波形数据的源： *WAV:BYT?* 返回值： *MSB*
关联命令	:WAVeform:WIDTh

:WAVeform:SOURce

描述	设置或查询传输波形数据的源。
命令格式	:WAVeform:SOURce <source> :WAVeform:SOURce?
参数说明	<source>:= {C<n>|F<x>|D<d>} ● C<n>:= 模拟通道，<n>:= 序号，整型 NR1 格式 ● F<x>:= 数学函数，<x>:= 序号，整型 NR1 格式 ● D<d>:= 数字通道，<d>:= 序号，整型 NR1 格式
返回格式	{C<n>|F<x>|D<d>}
示例	设置要从示波器传输 C1 波形数据： *:WAVeform:SOURce C1* *WAV:SOUR C1* 查询要从示波器传输波形数据的源： *WAV:SOUR?* 返回值： *C1*
关联命令	:WAVeform:DATA :WAVeform:PREamble

:WAVeform:STARt

描述	设置或查询波形数据的起始点。
命令格式	:WAVeform:STARt <value> :WAVeform:STARt?
参数说明	<value>:= 波形数据起始点，整型 NR1 格式

	注意：取值范围和波形采样点数及:WAVeform:POINt 的设置值有关。
返回格式	整型 NR1 格式
示例	当前波形采样点数为 400 kpts 时，设置起始点为 1000： *:WAVeform:STARt 1000* *WAV:STAR 1000* 查询读取波形的起始点： *WAV:STAR?* 返回值： *1000*
关联命令	:WAVeform:POINt

:WAVeform:INTerval

描述	设置或查询波形数据传输点间隔。
命令格式	:WAVeform:INTerval <value> :WAVeform:INTerval?
参数说明	<value>:= 点间隔，整型 NR1 格式 **注意**：取值范围和:WAVeform:STARt、:WAVeform:POINt 的设置值有关。
返回格式	整型 NR1 格式
示例	设置波形数据点间隔为 200： *:WAVeform:INTerval 200* *WAV:INT 200* 查询点之间的间隔： *WAV:INT?* 返回值： *200*
关联命令	:WAVeform:STARt :WAVeform:POINt

:WAVeform:POINt

描述	设置或查询波形数据传输的点数。
命令格式	:WAVeform:POINt <value> :WAVeform:POINt?

参数说明	<value>:= 点数，整型 NR1 格式 **注意**：取值范围和当前波形采样点数有关。
返回格式	整型 NR1 格式
示例	设置波形传输数据点数为 20000： *:WAVeform:POINt 20000* *WAV:POIN 20000* 查询波形传输数据点数： *WAV:POIN?* 返回值： *20000*
关联命令	:ACQuire:POINts

:WAVeform:MAXPoint

描述	受限于示波器内存，部分机型需要分片读取波形数据，此命令用于查询单片数据点数。
命令格式	:WAVeform:MAXPoint?
返回格式	<value> <value>:= 数值，整型 NR1 格式
示例	查询波形数据单片数据点数： *WAV:MAXP?* 返回值： *100000000*

:WAVeform:WIDTh

描述	设置或查询波形数据的传输格式。
命令格式	:WAVeform:WIDTh <type> :WAVeform:WIDTh?
参数说明	<type>:= {BYTE|WORD} ● BYTE：格式化的数据以 8 位字节的形式传输 ● WORD：格式化的数据将 16 位数据传输为两个字节，默认为 LSB **注意**：当读取波形数据大于 8bit 时，必须设置为 WORD 传输。

返回格式	{BYTE|WORD}
示例	设置波形数据传输格式为 BYTE： *:WAVeform:WIDTh BYTE* *WAV:WIDT BYTE* 查询当前波形数据传输的输出格式： *WAV:WIDT?* 返回值： *BYTE*

:WAVeform:PREamble

描述	查询:WAVeform:SOURce 指定源的波形参数。
命令格式	:WAVeform:PREamble?
返回格式	<bin> <bin>:= 二进制数据。头部为#9<9-Digits>，有关详细信息，请参阅“表 5- 3 波形参数说明”。
关联命令	:WAVeform:SOURce

表 5- 3 波形参数说明

注：“#9<9-Digits>”后的第一个字节为起始地址 0

地址	类型	长度 (byte)	描述说明
0~15	char	16	描述字符串，前 8 个字符为 WAVEDESC
16~31	char	16	模板字符串，前 7 个字符为 WAVEACE
32~33	short	2	COMM_TYPE，:WAVeform:WIDTh 设置的数据格式。0 为 BYTE，1 为 WORD。默认值为 0
34~35	short	2	COMM_ORDER，数据比特流格式。0 为 LSB，1 为 MSB。默认值为 0
36~39	long	4	wave_desc_length，波形参数长度，即:WAVeform:PREamble 返回数据块中除头部#9<9-Digits>剩余部分的长度。固定为 346 bytes
40~59	long	4*5	预留
60~63	long	4	WAVE_ARRAY_1，波形数据字节长度。 :WAVeform:WIDTh 设置的传输格式影响该值。仅适用于模拟通道

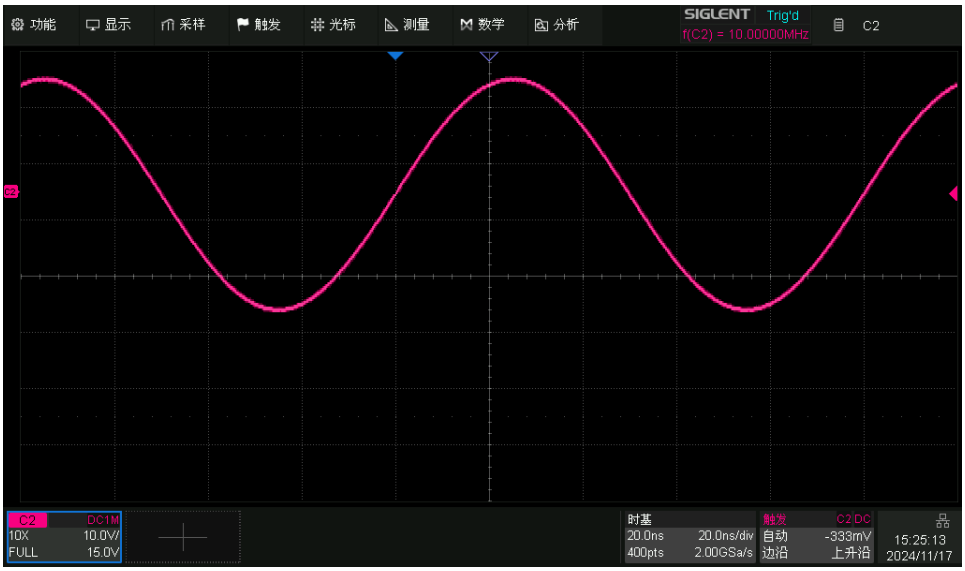
64~75	long	4*3	预留
76~91	char	16	设备名称字符串。固定为 Siglent SDS
92~95	long	4	预留
96~111	char	16	预留
112~115	long	4	预留
116~119	long	4	波形点数。仅适用于模拟通道。当读取分段采集波形数据时，表示单帧波形点数
120~131	long	4*3	预留
132~135	long	4	起始点。表示传输波形数据的起始点，与:WAVeform:STARt 返回值相同
136~139	long	4	数据间隔。表示数据点之间的间隔，与:WAVeform:INTerval 返回值相同
140~143	long	4	预留
144~147	long	4	read_frames，分段采集波形一次读取的帧数。可用于计算读取分段采集波形的总次数
148~151	long	4	sum_frames，分段采集波形的总帧数，用于计算读取分段采集波形的总次数
152~155	short	2*2	预留
156~159	float	4	垂直档位。不含探头系数的垂直档位值
160~163	float	4	垂直偏移。不含探头系数的垂直偏移值
164~167	float	4	码字值/div。不同机型的单位码字值不同
168~171	float	4	预留
172~173	short	2	ADC 位数
174~175	short	2	分段采集波形帧号，与:WAVeform:SEQuence 返回值的<value1>相同，默认值为 1
176~179	float	4	采样间隔，等于采样率倒数
180~187	long double	8	水平延迟。用于计算波形数据起始时间值。单位是 s
188~195	long double	8	预留
196~243	char	48	预留
244~291	char	48	预留

292~295	float	4	预留
296~311	struct	16	预留
312~315	float	4	预留
316~323	short	2*4	预留
324~325	short	2	水平时基，枚举值形式，详见“表 5- 4 时基枚举表”
326~327	short	2	通道耦合方式。0-DC,1-AC,2-GND
328~331	float	4	通道探头系数
332~333	short	2	垂直档位，枚举值形式。垂直档位值参考地址 156~159
334~335	short	2	通道带宽限制。0-OFF,1-20M,2-200M
336~343	float	4*2	预留
344~345	short	2	波形数据源。0-C1,1-C2,2-C3,3-C4,4-C5,5-C6,6-C7,7-C8

表 5- 4 时基枚举表

机型	索引值	时基	索引值	时基	索引值	时基
SDS5000X SDS2000X Plus SHS800X/SHS1000X SDS2000X HD SDS1000X HD SDS800X HD SDS3000X HD	0	200E-12	13	5E-6	26	100E-3
	1	500E-12	14	10E-6	27	200E-3
	2	1E-9	15	20E-6	28	500E-3
	3	2E-9	16	50E-6	29	1
	4	5E-9	17	100E-6	30	2
	5	10E-9	18	200E-6	31	5
	6	20E-9	19	500E-6	32	10
	7	50E-9	20	1E-3	33	20
	8	100E-9	21	2E-3	34	50
	9	200E-0	22	5E-3	35	100
	10	500E-9	23	10E-3	36	200
	11	1E-6	24	20E-3	37	500
	12	2E-6	25	50E-3	38	1000
机型	**索引-时基说明**					
SDS6000 Pro SDS6000A SDS6000L	最小时基为 100E-12，0-100E-12，上表时基索引值+1					
SDS7000A	最小时基为 50E-12，0-50E-12，1-100E-12，上表时基索引值+2					

:WAVeform:DATA

描述	查询:WAVeform:SOURce 指定源的波形数据。 **注意**：波形数据可能会含有“0x0A”，“0x0D”等字节，如传输终止符设置为这些字节如 LF(0x0A)时，数据流将会被截断。
命令格式	:WAVeform:DATA?
返回格式	<bin> <bin>:= 二进制数据。头部为#N<N-Digits>
示例	以下步骤演示通过波形数据重构波形。 读取模拟通道和数学函数波形（除 FFT）数据：  第一步：指定波形数据源，读取 8bit 波形数据 *:WAVeform:SOURce C2* *:WAVeform:DATA?* 返回信息： Data ··· Description 23 33 34 30 30 68 69 69 69 6A 6A 6A 6A 6B 6B 6A 6B 6B 6A ··· #3400..... 6A 6A 69 69 68 68 67 67 66 66 65 64 62 62 60 60 5F 5D 5C ··· ..... 5B 59 58 57 56 54 52 51 4F 4E 4C 4A 48 47 45 43 41 40 3D ··· ..... 3B 3A 38 36 34 32 31 2E 2C 2A 29 ..... 49 4B 4D 4E 50 51 ··· ..... 53 55 56 58 59 5A 5B 5D 5E 5F 60 62 63 63 65 65 66 66 67 ··· ..... 0A 0A ··· \n\n 数据头部#3400，以#字符开头，3 位十进制数 400 表示其后 400 个字节的数据(即图中 400pts)，最后两个字节 0A 0A 表示数据结束。 第二步：读取波形参数 *:WAVeform:PRE?* 波形参数解析请参阅“表 5- 3 波形参数说明”，解析后可以得到垂直档位为 10V/div，垂直偏移为 15V，水平时基为 2E-8s，水平延迟为 2E-8s，采样间隔为 500E-12s。

第三步：计算数据点的电压值

计算公式：垂直电压值（V）= 数据字节(码值) *（垂直档位/(码字值/div)）- 垂直偏移

参数	描述	上图实例
数据字节(码值)	数据字节为有符号数。如果机型垂直分辨率大于 8bit，每个数据点为 2 个字节，LSB 传输，左对齐低位补零。 对于 SDS6000 Pro，H12 和 H10 机型均为 12bit 数据，但 ADC 码字范围不同。	返回信息头部后的第 1 个字节 0x68 即为第 1 个数据点。转换成 10 进制为 104
垂直档位	:WAVeform:PREamble?返回的数据字节中，地址为 156~159 的值为原始垂直档位，地址为 328~331 的值为探头系数，二者乘积即为垂直档位。	10
垂直偏移	:WAVeform:PREamble?返回的数据字节中，地址为 160~163 的值为原始垂直偏移，地址为 328~331 的值为探头系数，二者乘积即为垂直偏移。	15
码字值/div	垂直方向上每大格的码字值。:WAVeform:PREamble?返回的数据字节中，地址为 164~167 的值	30

则第一个点的垂直电压值 = 104*(10/30) - 15 = 19.664V

第四步：计算数据点的时间值

计算公式：水平时间值（s）= 水平延迟-（水平时基*水平格数/2）+ 索引*采样间隔

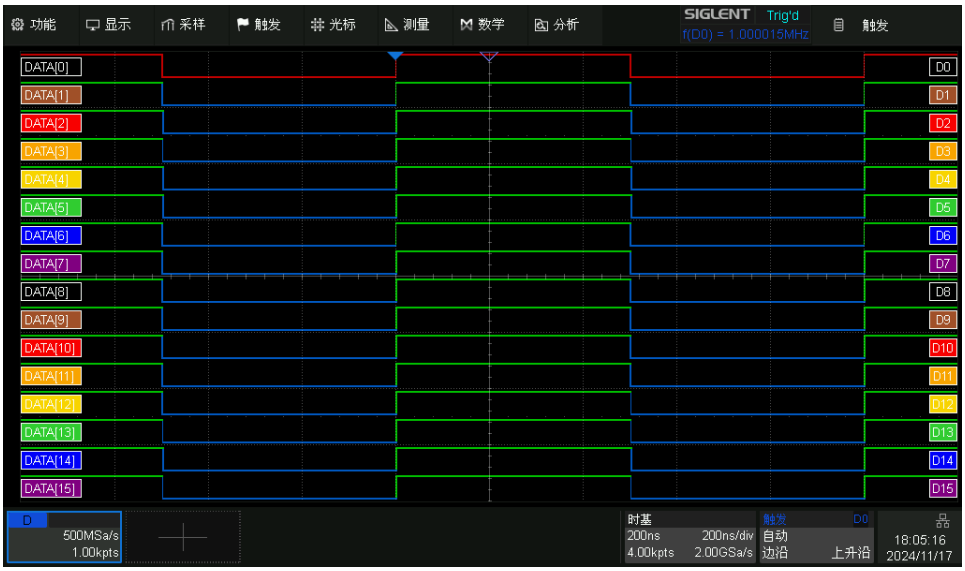
参数	描述	上图实例
水平时基	:WAVeform:PREamble?返回的数据字节中，地址为 324~325 的值，该值为时基枚举，查表获取时基数值	2E-8
水平延迟	:WAVeform:PREamble?返回的数据字节中，地址为 180~187 的值	2E-8
水平格数	水平方向网格数。	10
索引	数据点索引值，第 1 个点即为 0	--
采样间隔	:WAVeform:PREamble?返回的数据字节中，地址为 176~179 的值	5E-10

则第一个数据点的水平时间值 = (2E-08) - (2E-08*10/2) = -8E-08 = -80ns

第二个数据点的水平时间值 = -80ns + 0.2ns = -79.8ns

具体源代码请参考“模拟通道波形重构实例”章节。FFT 波形的示例请参考“数学函数 FFT 波形重构实例”。

读取数字通道波形数据：



第一步：指定波形数据源，读取波形数据

:WAVeform:SOURce D0

:WAVeform:DATA?

返回信息：

```
Data                                                    Description
--------------------------------------------------------  ----------------------
23 33 31 32 35 FF FF FF FF FF FF FF FF FF FF FF FF FF  #3125..................
FF FF FF FF FF FF 0F 00 00 00 00 00 00 00 00 00 00 00  ......................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C0  ......................
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ......................
FF FF FF FF FF FF FF FF FF FF FF FF 00 00 00 00 00 00  ......................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ......................
00 00 00 00 00 FC FF FF FF FF FF FF FF FF FF FF 0A 0A  ..............\n\n
```

数据头部#3125，以#字符开头，3 位十进制数 125 表示其后 125 个字节的数据(即图中 125*8=1000pts)，最后两个字节 0A 0A 表示数据结束。

第二步：读取波形参数

:WAVeform:PRE?

波形参数解析请参阅“表 5- 3 波形参数说明”，解析后可以得到水平时基为 2E-7s，水平延时为 2E-7s，采样间隔为 2E-9s。

第三步：计算数据点的逻辑电平

返回信息头部后的第 1 个字节 0xFF 即为前 8 个数据点。LSB 转换为二进制是 0b11111111。

第四步：计算数据点的时间值

计算公式：水平时间值（s）= 水平延时-（水平时基*水平格数/2）+索引*采样间隔

参数	描述	上图实例
水平时基	:WAVeform:PREamble?返回的数据字节中，地址为 324~325 的值，该值为时基枚举，查表获取时基数值	2E-7
水平延迟	:WAVeform:PREamble?返回的数据字节中，地址为 180~187 的值	2E-7
水平格数	水平方向网格数。	10
索引	数据点索引值，第 1 个点即为 0	--
采样间隔	:WAVeform:PREamble?返回的数据字节中，地址为 176~179 的值	2E-9

则第一个数据点的时间值 = (2E-07) - (2E-07*10/2) = -8E-07 = -800ns
第二个数据点的时间值 = -80ns + 2ns = -798ns
具体源代码请参考请参考“数字通道波形重构实例”章节。

关联命令	:WAVeform:STARt :WAVeform:INTerval :WAVeform:POINt :WAVeform:MAXPoint :WAVeform:WIDTh

:WAVeform:SEQuence

描述	读取分段采集波形时，设置或查询当前读取波形的帧号。仅在开启分段采集时有效。
命令格式	:WAVeform:SEQuence <value1>,<value2> :WAVeform:SEQuence?
参数说明	<value1>:= 分段采集波形的帧号，整型 NR1 格式。 用于:WAVeform:DATA?读取指定帧波形，设置为 0 时将返回所有帧数据。受限于示波器内存，当所有帧数据量超过限制时，需指定<value2>进行分片读取。 <value2>:= 读取所有帧数据时，分片读取的起始帧号。整型 NR1 格式。:WAVform:PREamble?返回数据字节中的 read_frmes(0x90-0x93)和 sum_frame(0x94-0x97)可用来计算读取分段采集波形的总次数。 **注意**: ● 当开启分段采集时，<value1>默认设置为 1；分段采集下打开历史帧后，<value1>默认设置为最后一帧；其他情况下，<value1>默认设置为 4294967295。

	● 设置范围和当前分段采集段数有关。
返回格式	<value1>,<value2>
示例	分段采集 5 帧波形后开启历史帧，每帧 10 kpts，总计 50 kpts，设置读取所有帧波形： *:WAVeform:SEQuence 0,1* *WAV:SEQ 0,1* 查询当前读取的帧号： *WAV:SEQ?* 返回值： *0,1* 读取分段采集波形的源代码，请参阅读取“分段采集波形重构实例”章节。

5.30 WGEN 命令系统

WGEN 命令子系统设置任意波形发生器的波形参数。当示波器支持任意波形发生器 SAG1021I 并获得许可，可以控制输出正弦波、方波、锯齿波、脉冲、直流、噪声、任意波。

WGEN 命令与 SDG 系列的命令相同，因此格式与其他命令子系统不一致。详情请参考 SDG 编程手册。

ARbWaVe

描述	设置或查询任意波的类型。
命令格式	<channel>:ARbWaVe INDEX,<index> <channel>:ARbWaVe NAME,<name> <channel>:ARbWaVe?
参数说明	<channel>:= {C1}，SAG1021I 和内置波形发生器仅支持一个输出通道 <index>:= 详见表 5-5，表示任意波的索引号 <name>:= 详见表 5-5，表示任意波的名称 **注意**：下表只是一个示例，索引号取决于具体的机型。“STL?”可用于查询索引和名称之间的精确映射关系。
返回格式	<channel>:ARWV INDEX,<index>,NAME,<name>
示例	设置 AWG 输出索引为 2 的任意波： *C1:ARWV INDEX,2* 查询 AWG 输出的任意波： *C1:ARWV?* 返回值： *C1:ARWV INDEX,2,NAME,StairUp* 设置 AWG 输出名称为 wave_1 的任意波： *C1:ARWV NAME,wave_1*
关联命令	STore,List

表 5-5 内置波形索引

<index>	<name>	<index>	<name>	<index>	<name>	<index>	<name>
0	Sine	12	Logfall	24	Gmonopuls	36	Triang
1	Noise	13	Logrise	25	Tripuls	37	Harris
2	StairUp	14	Sqrt	26	Cardiac	38	Bartlett
3	StairDn	15	Root3	27	Quake	39	Tan

4	Stairud	16	X^2	28	Chirp	40	Cot
5	Ppulse	17	X^3	29	Twotone	41	Sec
6	Npulse	18	Sinc	30	Snr	42	Csc
7	Trapezia	19	Gaussian	31	Hamming	43	Asin
8	Upramp	20	Dlorentz	32	Hanning	44	Acos
9	Dnramp	21	Haversine	33	Kaiser	45	Atan
10	Exp_fall	22	Lorentz	34	Blackman	46	Acot
11	Exp_rise	23	Gauspuls	35	Gausswin	47	Square

BaSic_WaVe

描述	设置或查询基本波形参数。
命令格式	<channel>:BaSic_WaVe <parameter>,<value > <channel>:BaSic_WaVe?
参数说明	<channel>:= {C1}，SAG1021I 和内置波形发生器仅支持一个输出通道 <parameter>:= 详见表 5- 6，表示参数名 <value>:= 详见表 5- 6，表示参数值
返回格式	<channel>:BSWV <parameter> <parameter>:= 当前波形的所有参数
示例	设置 AWG 输出三角波： *C1:BSWV WVTP,RAMP* 设置 AWG 输出频率为 2000Hz： *C1:BSWV FRQ,2000* 设置 AWG 输出幅度为 3Vpp： *C1:BSWV AMP,3* 查询 AWG 基本波形参数： *C1:BSWV?* 返回值： *C1:BSWV WVTP,SINE,FRQ,100HZ,PERI,0.01S,AMP,2V,OFST,0V,HLEV,1V,LLEV,-1V,PHSE,0*

表 5- 6 基本波形参数说明

<parameter>	<value>	描述
WVTP	<type>	:= {SINE|SQUARE|RAMP|PULSE|NOISE|ARB|DC|PRBS|IQ}如果命令未设置基本波形类型，WVPT 默认设置为当前波形。
FRQ	<frequency>	:= 频率。单位是赫兹“Hz”，可在数据手册查阅参数值的有效范围。当 WVTP 为噪声和直流，该值无效。

PERI	<period>	:= 周期。单位是秒“s”。可在数据手册查阅参数值的有效范围。当 WVTP 为噪声和直流，该值无效。
AMP	<amplitude>	:= 幅值。单位是伏特，峰峰值“Vpp”。可在数据手册查阅参数值的有效范围。当 WVTP 为噪声和直流，该参数无效。
OFST	<offset>	:= 偏置。单位是伏特“V”。可在数据手册查阅参数值的有效范围。当 WVTP 为噪声，该值无效。
SYM	<symmetry>	:= {0 至 100}。三角波的对称度。单位是“%”。仅当 WVTP 为三角波才能设置该参数。
DUTY	<duty>	:={0 至 100}。占空比。单位是“%”。该参数的值取决于频率。仅当 WVTP 是方波和脉冲才能设置该参数。
STDEV	<stdev>	:= 噪声的标准差。单位是伏特“V”。可在数据手册查阅参数值的有效范围。仅当 WVTP 是噪声时，才能设置。
MEAN	<mean>	:= 噪声的均值。单位是伏特“V”。可在数据手册查阅参数值的有效范围。仅当 WVTP 是噪声时，才能设置该参数。
WIDTH	<width>	:= 正脉宽。单位是秒“s”。可在数据手册查阅参数值的有效范围。仅当 WVTP 是脉冲波时，才能设置该参数。

OUTPut

描述	设置或查询 AWG 输出状态及输出参数。
命令格式	<channel>:OUTPut <state>,LOAD,<load>,PLRT<polarity> <channel>:OUTPut?
参数说明	<channel>:= {C1}，SAG1021I 和内置波形发生器仅支持一个输出通道 <state>:= {ON|OFF} <load>:= {50|HZ}，单位为 ohm <polarity>:= {NOR}，AWG 模块无法设置该参数，默认为 NOR
返回格式	<channel>:OUTP <state>,LOAD,<load>,PLRT,<polarity>
示例	开启 CH1： *C1:OUTP ON* 查询 CH1 输出状态： *C1:OUTP?* 返回值： *C1:OUTP ON,LOAD,HZ,PLRT,NOR* 设置 CH1 的负载为 50ohm： *C1:OUTP LOAD,50* 设置 CH1 的负载为高阻： *C1:OUTP LOAD,HZ*

SToreList

描述	查询已存储波形名称。若无存储波形，将返回“EMPTY”。
命令格式	StoreList? [<location>]
参数说明	<location>:= {BUILDIN|USER}
示例	查询 SDS2000X Plus 内置波形发生器中的内置任意波列表： *STL? BUILDIN* 返回值： *STL M10, ExpFal, M100, ECG14, M101, ECG15, M102, LFPulse, M103, Tens1, M104, Tens2, M105, Tens3, M106, Airy, M107, Besselj, M108, Bessely, M109, Dirichlet, M11, ExpRise, M110, Erf, M111, Erfc, M112, Erfclnv, M113, Erflnv, M114, Laguerre, M115, Legend, M116, Versiera, M117, Weibull, M118, LogNormal, M119, Laplace, M12, LogFall, M120, Maxwell, M121, Rayleigh, M122, Cauchy, M123, CosH, M124, CosInt, M125, CotH, M126, CscH, M127, SecH, M128, SinH, M129, SinInt, M13, LogRise, M130, TanH, M131, ACosH, M132, ASecH, M133, ASinH, M134, ATanH, M135, ACsch, M136, ACoth, M137, Bartlett, M138, BohmanWin, M139, ChebWin, M14, Sqrt, M140, FlattopWin, M141, ParzenWin, M142, TaylorWin, M143, TukeyWin, M144, Duty01, M145, Duty02, M146, Duty04, M147, Duty06, M148, Duty08, M149, Duty10, M15, Root3, M150, Duty12, M151, Duty14, M152, Duty16, M153, Duty18, M154, Duty20, M155, Duty22, M156, Duty24, M157, Duty26, M158, Duty28, M159, Duty30, M16, X^2, M160, Duty32, M161, Duty34, M162, Duty36, M163, Duty38, M164, Duty40, M165, Duty42, M166, Duty44, M167, Duty46, M168, Duty48, M169, Duty50, M17, X^3, M170, Duty52, M171, Duty54, M172, Duty56, M173, Duty58, M174, Duty60, M175, Duty62, M176, Duty64, M177, Duty66, M178, Duty68, M179, Duty70, M18, Sinc, M180, Duty72, M181, Duty74, M182, Duty76, M183, Duty78, M184, Duty80, M185, Duty82, M186, Duty84, M187, Duty86, M188, Duty88, M189, Duty90, M19, Gaussian, M190, Duty92, M191, Duty94, M192, Duty96, M193, Duty98, M194, Duty99, M195, demo1_375, M196, demo1_16k, M197, demo2_3k, M198, demo2_16k, M2, StairUp, M20, Dlorentz, M21, Haversine, M22, Lorentz, M23, Gauspuls, M24, Gmonopuls, M25, Tripuls, M26, Cardiac, M27, Quake, M28, Chirp, M29, Twotone, M3, StairDn, M30, SNR, M31, Hamming, M32, Hanning, M33, kaiser, M34, Blackman, M35, Gausswin, M36, Triangle, M37, BlackmanH, M38, Bartlett-Hann, M39, Tan, M4, StairUD, M40, Cot, M41, Sec, M42, Csc, M43, Asin, M44, Acos, M45, Atan, M46, Acot, M47, Square, M48, SineTra, M49, SineVer, M5, Ppulse, M50, AmpALT, M51, AttALT, M52, RoundHalf, M53, RoundsPM, M54, BlaseiWave, M55, DampedOsc, M56, SwingOsc, M57, Discharge, M58, Pahcur, M59, Combin, M6, Npulse, M60, SCR, M61, Butterworth, M62, Chebyshev1, M63, Chebyshev2, M64, TV, M65,*

	Voice, M66, Surge, M67, Radar, M68, Ripple, M69, Gamma, M7, Trapezia, M70, StepResp, M71, BandLimited, M72, CPulse, M73, CWPulse, M74, GateVibr, M75, LFMPulse, M76, MCNoise, M77, AM, M78, FM, M79, PFM, M8, Upramp, M80, PM, M81, PWM, M82, EOG, M83, EEG, M84, EMG, M85, Pulseilogram, M86, ResSpeed, M87, ECG1, M88, ECG2, M89, ECG3, M9, Dnramp, M90, ECG4, M91, ECG5, M92, ECG6, M93, ECG7, M94, ECG8, M95, ECG9, M96, ECG10, M97, ECG11, M98, ECG12, M99, ECG13

SYNC

描述	设置或查询外置 AWG 同步输出状态。
命令格式	<channel>:SYNC <state> <channel>:SYNC?
参数说明	<state>:= {ON|OFF} <channel>:= {C1}，SAG1021I 和内置波形发生器仅支持一个输出通道。
返回格式	<channel>:SYNC <state>,TYPE,<TYPE>
示例	设置开启外置 AWG 的同步输出： *C1:SYNC ON* 查询外置 AWG 的同步输出状态： *C1:SYNC?* 返回值： *C1:SYNC ON,TYPE,CH1*

VOLTPRT

描述	设置或查询过压保护状态。
命令格式	VOLTPRT <state> VOLTPRT?
参数说明	<state>:= {ON|OFF}
返回格式	{ON|OFF}

6 编程实例

本章例举了在 LabVIEW、Visual Basic、Visual C++等开发环境中如何使用命令实现示波器常用功能的编程实例。这些实例都是基于 VISA（Virtual Instrument Software Architecture）库或 Socket 编程实现的，通过这些实例，您可以开发更多的应用程序。

6.1 基于 VISA 编程

VC++实例

环境：Windows7 32 位系统，Visual Studio。

描述：分别通过 USBTMC 和 TCP/IP 访问示波器，使用 NI-VISA 执行写入和读取操作。

步骤：

1. 打开 Visual Studio 软件，新建一个 VC++ win32 项目。
2. 设置调用 NI-VISA 库的项目环境。此处给出两种设置方法，分别为静态和动态：

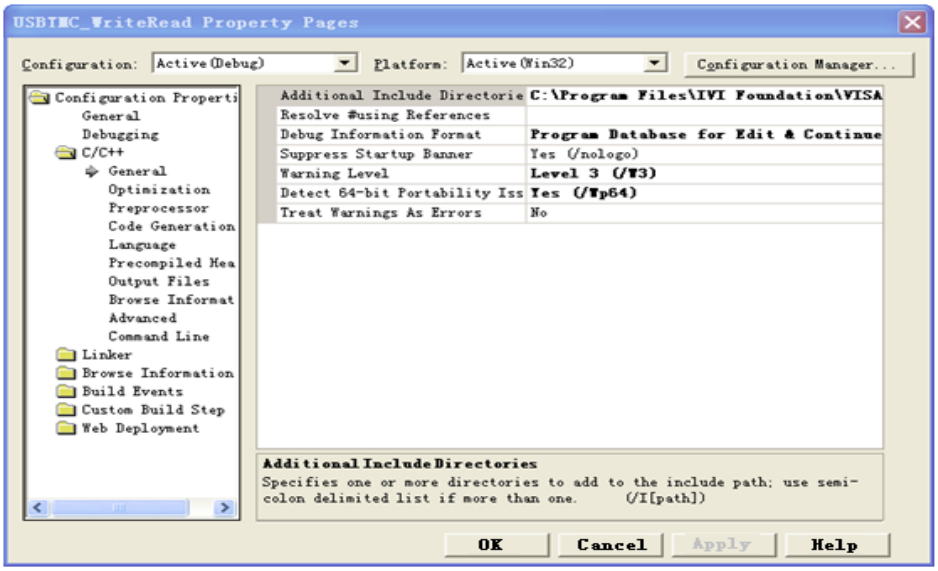
   a) 静态调用：

   在 NI-VISA 安装路径找: visa.h、visatype.h、 visa32.lib 文件，将它们复制到 VC++项目的根路径下并添加到项目中。在 projectname.cpp 文件上添加下列两行代码：

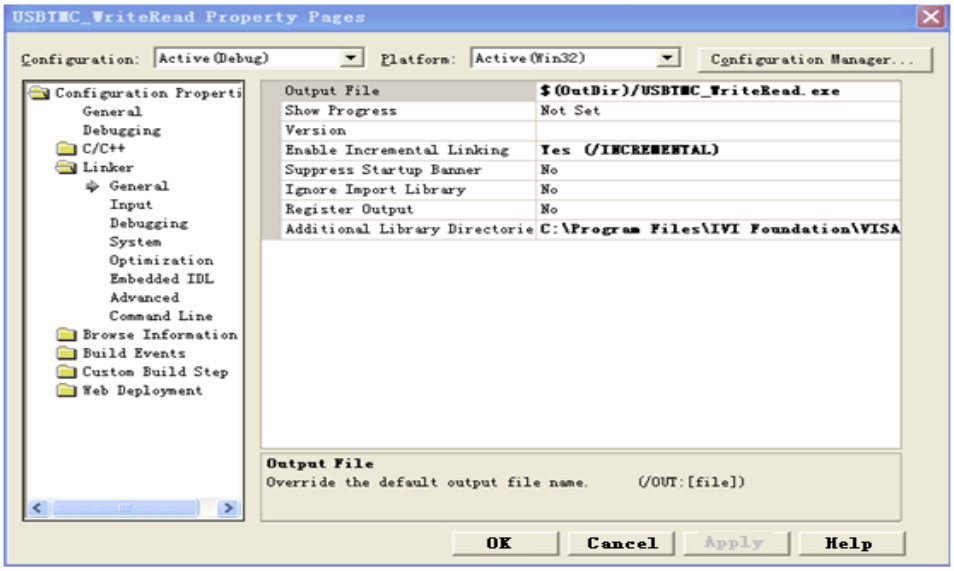
   ```
   #include "visa.h"
   #pragma comment(lib,"visa32.lib")
   ```

   b) 动态调用：

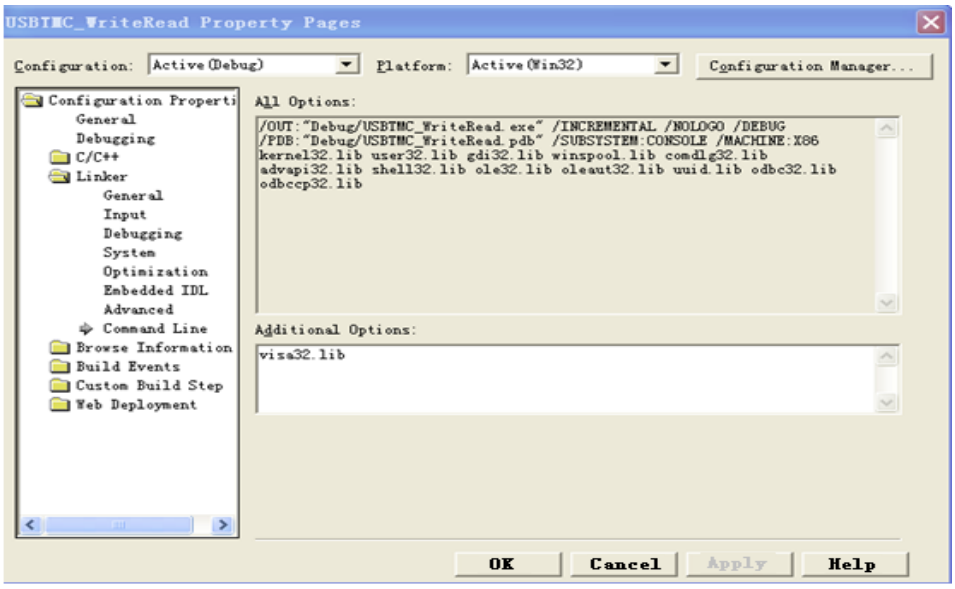
   点击“project>>properties”，在属性对话框左侧选择“C/C++->General”，将“Additional Include Directories”项的值设置为 NI-VISA 的安装路径（例如：C:\Program Files\IVI Foundation\VISA\WinNT\include），如下图所示：



在属性对话框左侧选择“Linker---General”，并将“Additional Library Directories”项的值设置为 NI-VISA 的安装路径。（例如：C:\Program Files\IVI Foundation\VISA\WinNT\lib\msc），如下图所示：



在属性对话框左侧选择“Linker---Command Line”，将“Additional”项的值设置为 visa32.lib，如下图所示：



在 projectname.cpp 文件上添加 visa.h 文件：

```
#include <visa.h>
```

3. 代码：

a) USBTMC：

```
Int             Usbtmc_test()
{
        /* This code demonstrates sending synchronous read & write commands */
        /* to an USB Test & Measurement Class (USBTMC) instrument using      */
        /* NI-VISA                                                           */
        /* The example writes the "*IDN?\n" string to all the USBTMC         */
        /* devices connected to the system and attempts to read back         */
        /* results using the write and read functions.                       */
        /* The general flow of the code is   */
        /*      Open Resource Manager        */
        /*      Open VISA Session to an Instrument                           */
        /*      Write the Identification Query Using viPrintf
*/
        /*      Try to Read a Response With viScanf
                */
```

```
        /*      Close the VISA Session          */
        /*****************************************************/
        ViSession defaultRM;
        ViSession instr;
        ViUInt32 numInstrs;
        ViFindList findList;
        ViUInt32 retCount;
        ViUInt32 writeCount;
        ViStatus status;
        char        instrResourceString[VI_FIND_BUFLEN];
        unsigned  char        buffer[100];
        char        stringinput[512];
        int         i;
        /** First we must call viOpenDefaultRM to get the manager
        * handle.   We will store this handle in defaultRM.*/
        status= ViOpenDefaultRM (&defaultRM);
        if (status<VI_SUCCESS)
        {
                printf ("Could not open a session to the VISA Resource
    Manager!\n");
                return      status;
        }
        /* Find all the USB TMC VISA resources in our system and store the   number
of resources in the system in numInstrs.                    */
        status = viFindRsrc (defaultRM, "USB?*INSTR", &findList, &numInstrs,
instrResourceString);
        if (status<VI_SUCCESS)
        {
                printf ("An error occurred while finding resources.\nHit enter to
    continue.");
```

```
                fflush(stdin);
                getchar();
                viClose (defaultRM);
                return      status;
        }
        /** Now we will open VISA sessions to all USB TMC instruments.
        * We must use the handle from viOpenDefaultRM and we must
        * also use a string that indicates which instrument to open.    This
        * is called the instrument descriptor.    The format for this string
        * can be found in the function panel by right clicking on the
        * descriptor parameter. After opening a session to the
        * device, we will get a handle to the instrument which we
        * will use in later VISA functions.    The AccessMode and Timeout
        * parameters in this function are reserved for future
        * functionality.    These two parameters are given the value VI_NULL.*/
        for (i= 0; i<numInstrs; i++)
        {
                if (i> 0)
{
        viFindNext (findList, instrResourceString);
}
        status = viOpen (defaultRM, instrResourceString, VI_NULL, VI_NULL, &instr);
                if (status<VI_SUCCESS)
                {
                                printf ("Cannot open a session to the
device %d.\n", i+1);
                                continue;
                }
                /* * At this point we now have a session open to the USB TMC
```

```
instrument.
                    * We will now use the viPrintf function to send the device the string
"*IDN?\n",
                    * asking for the device's identification.    */
                    char * cmmand ="*IDN?\n";
                    status = viPrintf    (instr, cmmand);
                    if (status<VI_SUCCESS)
                    {
                                        printf ("Error writing to the device %d.\n",
    i+1);
                                        status = viClose (instr);
                                        continue;
                    }
                    /** Now we will attempt to read back a response from the device
to
                    * the identification query that was sent.    We will use the viScanf
                    * function to acquire the data.
                    * After the data has been read the response is displayed.*/
                    status =    viScanf(instr, "%t", buffer);
                    if (status<VI_SUCCESS)
    {
    printf ("Error reading a response from the device %d.\n", i+1);
    }
        else
    {
                        printf ("\nDevice %d: %*s\n", i+1,retCount, buffer);
    }
                    status = viClose (instr);
        }
        /** Now we will close the session to the instrument using
```

```
                * viClose. This operation frees all system resources.                    */
                status = viClose (defaultRM);
        printf("Press 'Enter' to exit.");
                fflush(stdin);
                getchar();
                return 0;
}
```

b) TCP/IP:

```
int             TCP_IP_Test(char *pIP)
{
                char            outputBuffer[VI_FIND_BUFLEN];
                ViSession defaultRM, instr;
                ViStatus status;
                ViUInt32 count;
                ViUInt16 portNo;
                /* First we will need to open the default resource manager. */
                status = viOpenDefaultRM (&defaultRM);
                if (status<VI_SUCCESS)
                {
                        printf("Could not open a session to the VISA Resource Manager!\n");
                }
                /* Now we will open a session via TCP/IP device */
                char            head[256] ="TCPIP0::";
                char            tail[] ="::INSTR";
                char            resource [256];
                strcat(head,pIP);
                strcat(head,tail);
                status = viOpen (defaultRM, head, VI_LOAD_CONFIG, VI_NULL, &instr);
```

```
        if (status<VI_SUCCESS)
        {
                printf ("An error occurred opening the session\n");
                viClose(defaultRM);
        }
        status = viPrintf(instr, "*idn?\n");
        status = viScanf(instr, "%t", outputBuffer);
        if (status<VI_SUCCESS)
        {
                printf("viRead failed with error code: %x \n",status);
                viClose(defaultRM);
        }
    else
    {
                printf ("\ndata read from device: %*s\n", 0,outputBuffer);
    }
        status = viClose (instr);
        status = viClose (defaultRM);
    printf("Press 'Enter' to exit.");
        fflush(stdin);
        getchar();
                        return 0;
}
```

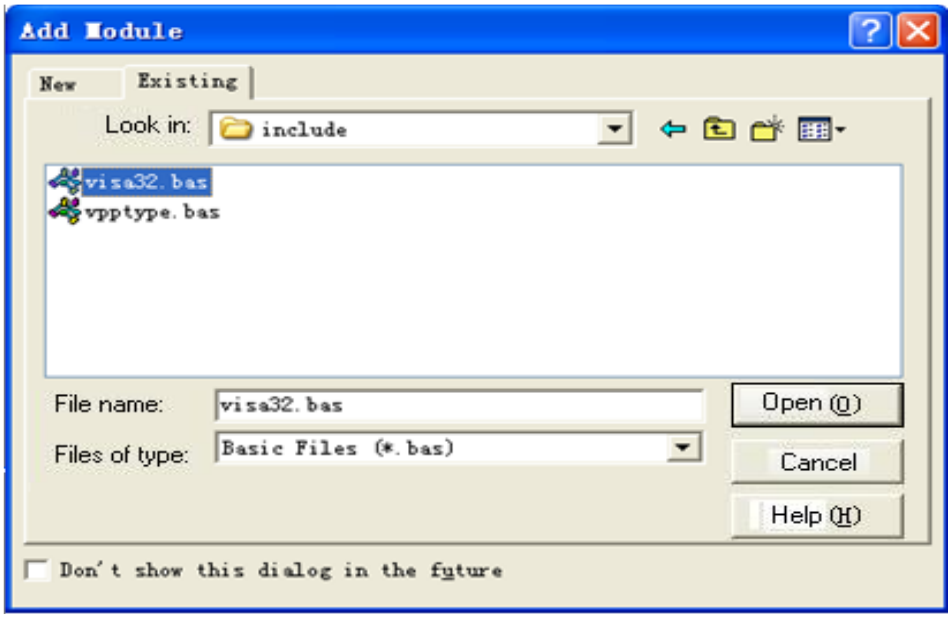
VB 实例

环境：Windows7 32 位系统，Microsoft Visual Basic 6.0。

描述：分别通过 USBTMC 和 TCP/IP 访问示波器，使用 NI-VISA 执行写入和读取操作。

步骤：

1. 打开 Visual Basic 软件，并新建一个标准的应用程序项目。
2. 设置项目环境中使用 NI-VISA 库。单击项目的现有标签>>添加模块，在 NI-VISA 安装路径下的 include 文件夹中搜索 visa32.bas 文件并添加该文件。这将使得 VISA 功能和 VISA 的数据类型在程序中使用。如下图所示：



3. 代码：

a) USBTMC：

```
Private Function Usbtmc_test() As Long
    ' This code demonstrates sending synchronous read & write commands
    ' to an USB Test & Measurement Class (USBTMC) instrument using
    ' NI-VISA
    ' The example writes the "*IDN?\n" string to all the USBTMC
    ' devices connected to the system and attempts to read back
    ' results using the write and read functions.
     ' The general flow of the code is
    '       Open Resource Manager
```

```
'       Open VISA Session to an Instrument
'       Write the Identification Query Using viWrite
'       Try to Read a Response With viRead
'       Close the VISA Session
Const MAX_CNT = 200

Dim defaultRM As Long
Dim instrsesn As Long
Dim numInstrs As Long
Dim findList As Long
Dim retCount As Long
Dim writeCount As Long
Dim status As Long
Dim instrResourceString As String * VI_FIND_BUFLEN
Dim buffer As String * MAX_CNT
Dim i As Integer
 ' First we must call viOpenDefaultRM to get the manager
' handle. We will store this handle in defaultRM.
status = viOpenDefaultRM(defaultRM)
If (status < VI_SUCCESS) Then
      Debug.Print "Could not open a session to the VISA Resource Manager!"
      Usbtmc_test = status
      Exit Function
End If

' Find all the USB TMC VISA resources in our system and store the
' number of resources in the system in numInstrs.
  status= ViFindRsrc(defaultRM,"USB?*INSTR",findList,numInstrs,instrResourceString)
If (status < VI_SUCCESS) Then
```

```
        Debug.Print "An error occurred while finding resources."
        viClose (defaultRM)
        Usbtmc_test = status
        Exit Function
    End If

     ' Now we will open VISA sessions to all USB TMC instruments.
     ' We must use the handle from viOpenDefaultRM and we must
    ' also use a string that indicates which instrument to open. This
    ' is called the instrument descriptor.    The format for this string
    ' can be found in the function panel by right clicking on the
    ' descriptor parameter. After opening a session to the
    ' device, we will get a handle to the instrument which we
    ' will use in later VISA functions.    The AccessMode and Timeout
    ' parameters in this function are reserved for future
    ' functionality.    These two parameters are given the value VI_NULL.
    For i = 0 To numInstrs
        If (i > 0) Then
            status = viFindNext(findList, instrResourceString)
        End If
            status = viOpen(defaultRM, instrResourceString, VI_NULL, VI_NULL, instrsesn)
        If (status < VI_SUCCESS) Then
            Debug.Print "Cannot open a session to the device ", i + 1
            GoTo NextFind
        End If

        ' At this point we now have a session open to the USB TMC instrument.
        ' We will now use the viWrite function to send the device the string "*IDN?",
        ' asking for the device's identification.
```

```
        status = viWrite(instrsesn, "*IDN?", 5, retCount)
        If (status < VI_SUCCESS) Then
                Debug.Print "Error writing to the device."
                status = viClose(instrsesn)
                GoTo NextFind
        End If

        ' Now we will attempt to read back a response from the device to
        ' the identification query that was sent.  We will use the viRead
        ' function to acquire the data.
        ' After the data has been read the response is displayed.
        status = viRead(instrsesn, buffer, MAX_CNT, retCount)
        If (status < VI_SUCCESS) Then
            Debug.Print "Error reading a response from the device.", i + 1
        Else
            Debug.Print i + 1, retCount, buffer
        End If
        status = viClose(instrsesn)
    Next i

    ' Now we will close the session to the instrument using
    ' viClose. This operation frees all system resources.
     status = viClose(defaultRM)
    Usbtmc_test = 0
End Function
```

b) TCP/IP:

```
Private Function TCP_IP_Test(ip As String) As Long
```

```
Dim outputBuffer As String * VI_FIND_BUFLEN
Dim defaultRM As Long
Dim instrsesn As Long
Dim status As Long
Dim count As Long

' First we will need to open the default resource manager.
status = viOpenDefaultRM (defaultRM)
If (status < VI_SUCCESS) Then
    Debug.Print "Could not open a session to the VISA Resource Manager!"
    TCP_IP_Test = status
    Exit Function
End If

' Now we will open a session via TCP/IP device
status = viOpen(defaultRM, "TCPIP0::" + ip + "::INSTR", VI_LOAD_CONFIG, VI_NULL, instrsesn)
 If (status < VI_SUCCESS) Then
    Debug.Print "An error occurred opening the session"
    viClose (defaultRM)
    TCP_IP_Test = status
    Exit Function
End If

status = viWrite(instrsesn, "*IDN?", 5, count)
 If (status < VI_SUCCESS) Then
    Debug.Print "Error writing to the device."
End If
status = viRead(instrsesn, outputBuffer, VI_FIND_BUFLEN, count)
```

```
    If (status < VI_SUCCESS) Then
        Debug.Print "Error reading a response from the device.", i + 1
    Else
        Debug.Print "read from device:", outputBuffer
    End If
    status = viClose(instrsesn)
    status = viClose(defaultRM)
    TCP_IP_Test = 0
End Function
```

MATLAB 实例

环境：Windows7 32 位系统 MATLAB R2010b。

描述：分别通过 USBTMC 和 TCP/IP 访问示波器，使用 NI-VISA 执行写入和读取操作。

步骤：

1. 打开 MATLAB 软件，并修改当前目录。在本示例中，当前目录修改为："D:\USBTMC_TCPIP_Demo"。
2. 点击文件>>新建>>脚本（File>>New>>Script）在 MATLAB 界面来创建一个空的 M 文档。
3. 代码：

   a) USBTMC：

```
function USBTMC_test()
% This code demonstrates sending synchronous read & write commands
% to an USB Test & Measurement Class (USBTMC) instrument using
% NI-VISA

%Create a VISA-USB object connected to a USB instrument
vu = visa('ni','USB0::0xF4EC::0xEE38::0123456789::INSTR');

%Open the VISA object created
fopen(vu);

%Send the string "*IDN?",asking for the device's identification.
fprintf(vu,'*IDN?');

%Request the data
outputbuffer = fscanf(vu);
disp(outputbuffer);

%Close the VISA object
fclose(vu);
```

```
delete(vu);
clear vu;

end
```

b) TCP/IP:

```
function TCP_IP_test( IPstr )
% This code demonstrates sending synchronous read & write commands
% to an TCP/IP instrument using NI-VISA

%Create a VISA-TCPIP object connected to an instrument
%configured with IP address.
vt = visa('ni',['TCPIP0::',IPstr,'::INSTR']);

%Open the VISA object created
fopen(vt);

%Send the string "*IDN?", asking for the device's identification.
fprintf(vt,'*IDN?');

%Request the data
outputbuffer = fscanf(vt);
disp(outputbuffer);

%Close the VISA object
fclose(vt);
delete(vt);
clear vt;

end
```

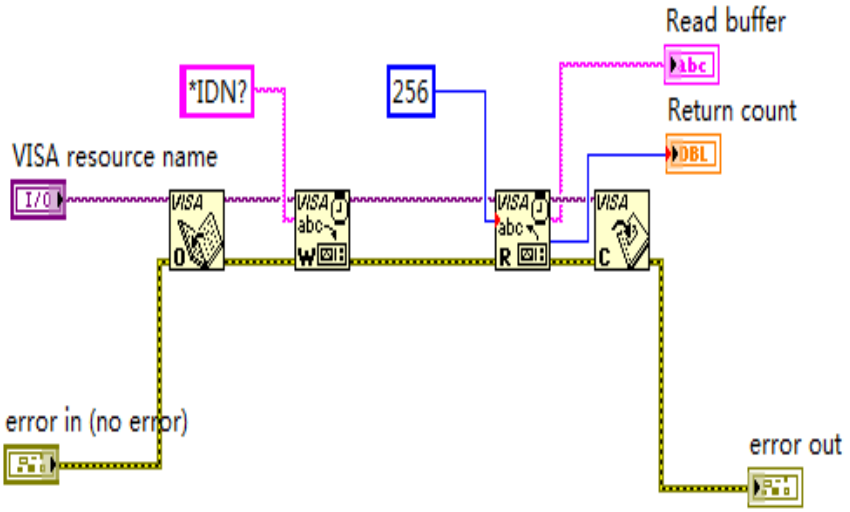
LabVIEW 实例

环境：Windows7 32 位系统，LabVIEW 2011。

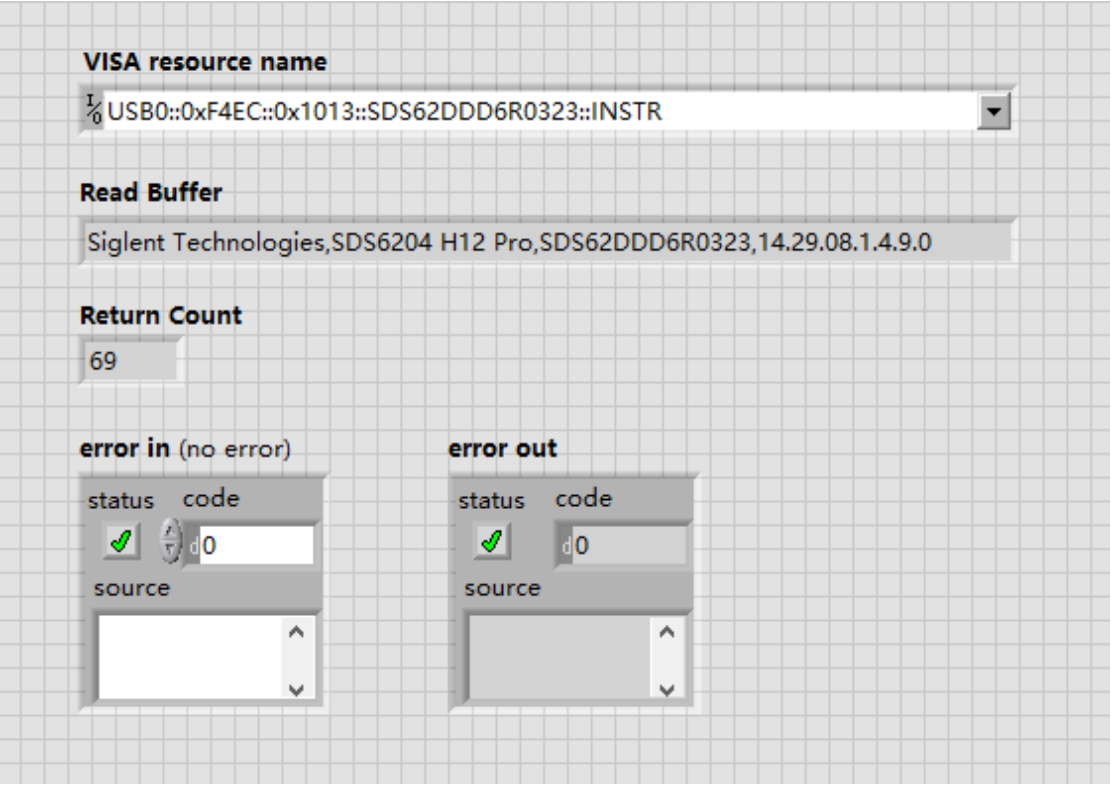
描述：分别通过 USBTMC 和 TCP/IP 访问示波器，使用 NI-VISA 执行写入和读取操作。

步骤：

1. 打开 LabVIEW 软件，并创建一个 VI 文件。
2. 添加控件。右键单击前面板接口，选择并加入 VISA 资源名称，错误输入，错误输出以及控制栏的一些指标。
3. 打开框图界面。右击 VISA 资源名称，并在弹出菜单的 VISA Palette 中选择和添加下列功能：**VISA Write**、**VISA Read**、**VISA Open** 和 **VISA Close**。
4. 如下图连接它们：



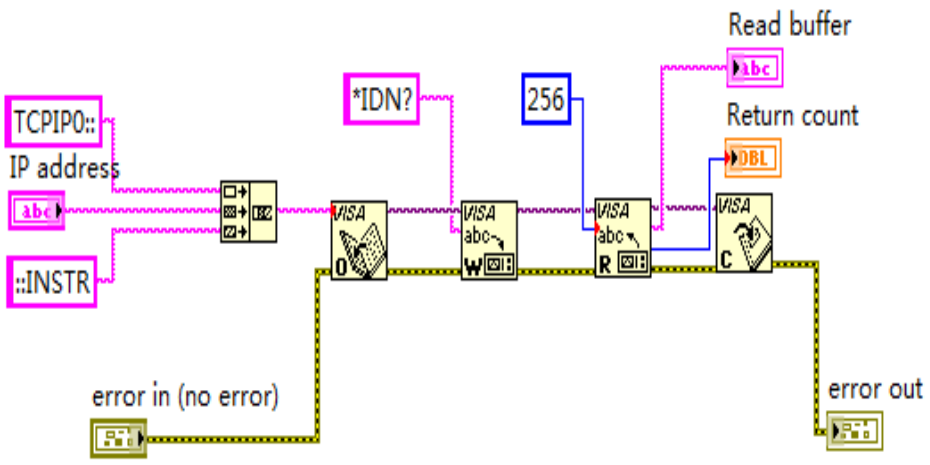
5. 从 VISA 资源名列表中选择设备资源并运行程序。



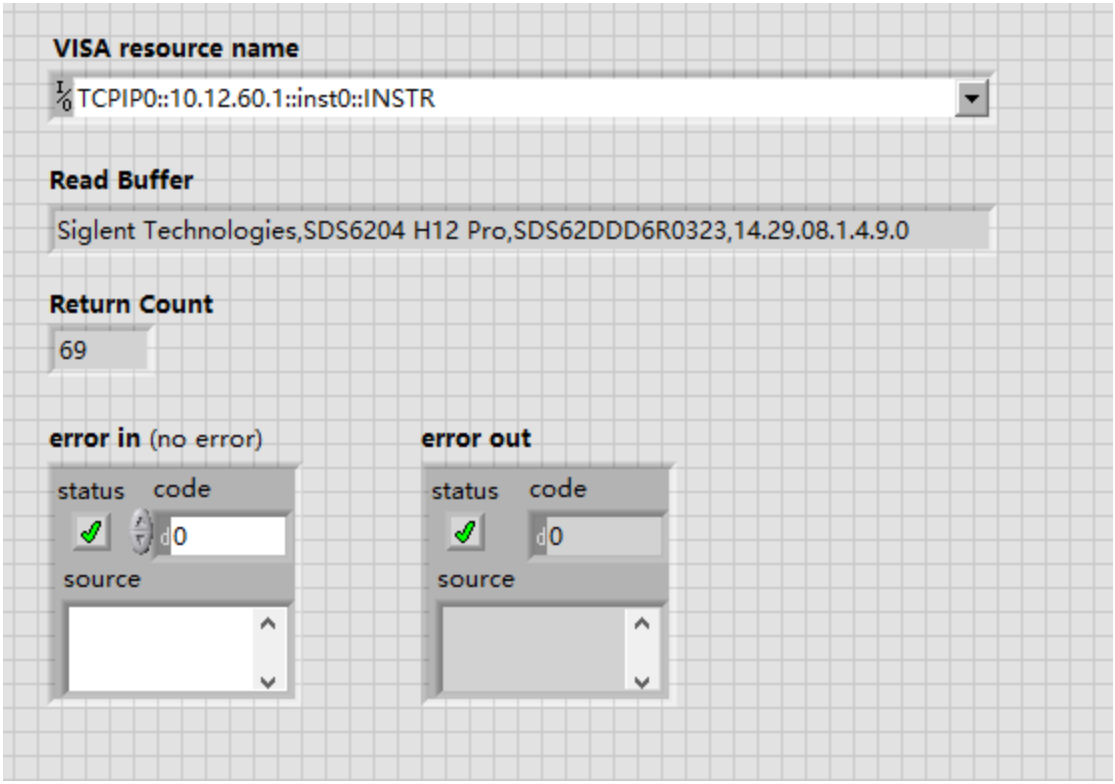
在此例中，VI 打开了一个 USBTMC 设备的 VISA 会话，并向设备写一个命令并回读响应值。当所有通信完成时，VI 将关闭 VISA 会话。

6. 通过 TCP/IP 与设备通信类似于 USBTMC。但是，你需要改变 VISA 写入和 VISA 读取功能来同步 I/O。LabVIEW 的默认值是异步的 I/O。右键单击该节点，然后从快捷菜单中选择同步 I/O 模式>>同步来写入或读取同步数据。

7. 按照下图连接它们：



8. 输入 IP 地址并运行程序。


VISA resource name
TCPIP0::10.12.60.1::inst0::INSTR
Read Buffer
Siglent Technologies,SDS6204 H12 Pro,SDS62DDD6R0323,14.29.08.1.4.9.0
Return Count
69
error in (no error)
status
code
0
source
error out
status
code
0
source

C#实例

环境：Windows7 32 位系统，Visual Studio 2008/2010

描述：分别通过 USBTMC 和 TCP/IP 访问示波器，使用 NI-VISA 执行写入和读取操作。

步骤：

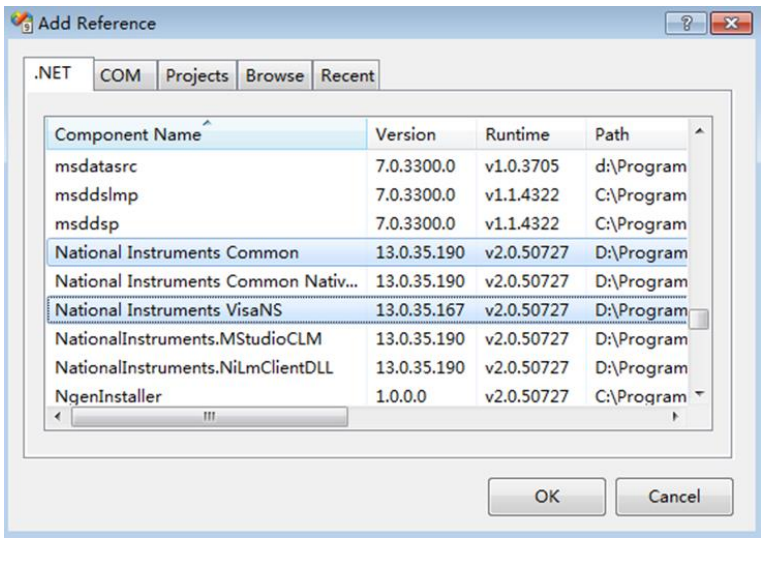
1. 打开 Visual Studio 软件，新建一个 C# 项目。
2. 将下面的代码剪切并粘贴到 C# 源文件中。
3. 编辑程序以使用示波器的 VISA 地址。
4. 添加引用。

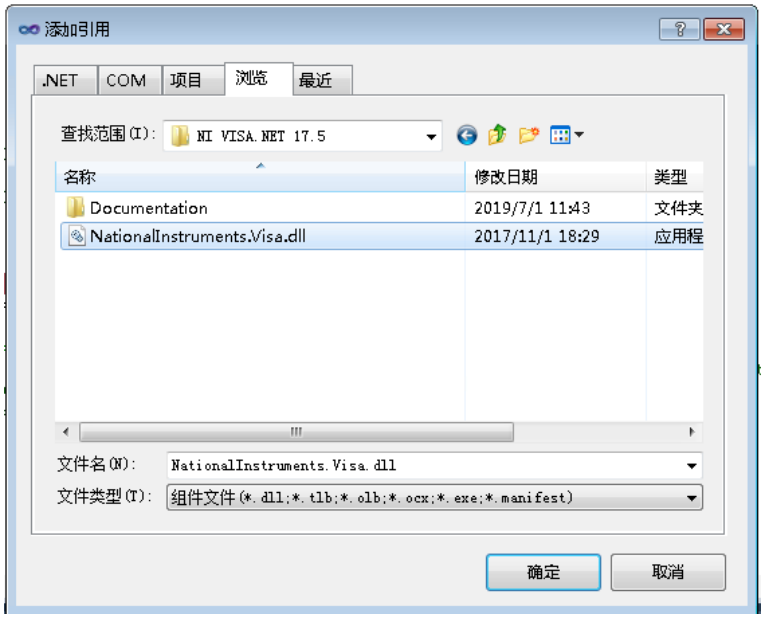
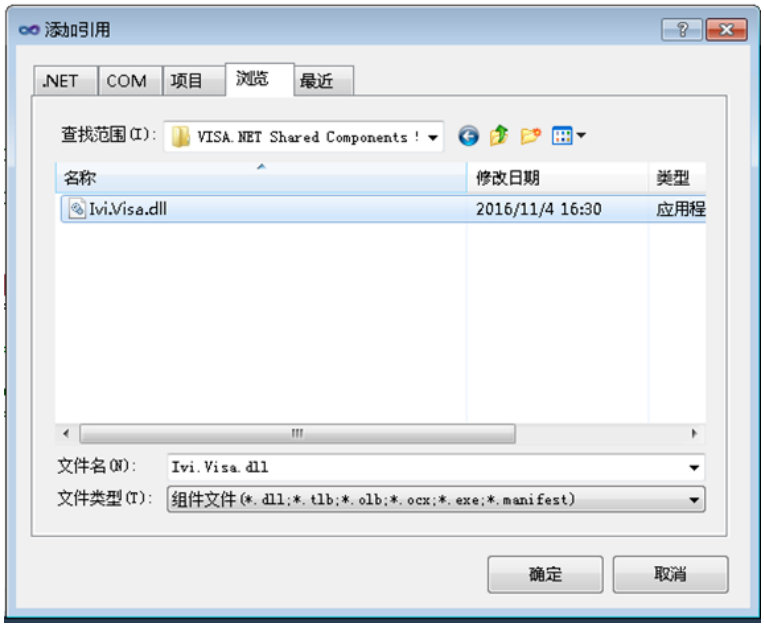
   添加 Ivi.Visa.dll 和 NationalInstruments.Visa.dll 到项目。

   如果您安装的 NI-VISA 版本过低（例如：5.4.0），则需添加 NationalInstruments.Common.dll 和 NationalInstruments.VisaNS.dll。（链接 NI-VISA .NET Library - NI 的第 11 项详细说明了 NI-VISA NET 和 NI-VISA NS 之间的一些主要差异）

   （**注意**：安装 NI-VISA 时，必须安装.NET Framework 3.5/4.0/4.5 支持的语言。）

   - 在 Microsoft Visual Studio 环境的“Solution Explorer”窗口中右键单击要修改的项目。
   - 选择添加引用。
   - 在“添加应用”对话框，选择“浏览”选项卡，然后导航到 NI-VISA 安装的文件夹。（例如：C:\Program Files (x86)\IVI Foundation\VISA\Microsoft.NET\..）
   - 选择下面的.dll 文件，然后单击“OK”。

在 NI-VISA NS (Visual Studio 2008) 添加引用	

在 NI-VISA.NET (Visual Studio 2010) 添加引用	 

5. NI-VISA NS 代码

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using NationalInstruments.VisaNS;
```

```
namespace TestVisa
{
class Program
{
static void Main(string[] args)
{
// Find all the USBTMC resources
string[]
usbRsrcStrings=ResourceManager.GetLocalManager().FindResources("USB?*INSTR");
if (usbRsrcStrings.Length <= 0)
{
Console.WriteLine("Cannot find USBTMC Device!");
return;
}

//Choose the first resource string to connect the device.
//You can input the address manually
//USBTMC:
//MessageBasedSession
mbSession=(MessageBasedSession)ResourceManager.GetLocalManager().Open("USB0::0
xF4EC::0xEE38::0123456789::INSTR");
/TCP IP:
//MessageBasedSession
mbSession=(MessageBasedSession)ResourceManager.GetLocalManager().Open("TCPIP0:
:192.168.1.100::INSTR");
MessageBasedSession
mbSession=(MessageBasedSession)ResourceManager.GetLocalManager().Open(usbRsrc
Strings[0]);
mbSession.Write("*IDN?");
string result = mbSession.ReadString();
mbSession.Dispose();
```

```
Console.WriteLine(result);
}
}
}
```

6. NI-VISA.NET 代码：

```
using System;
using System.Collections.Generic;
using System.Linq;
using NationalInstruments.Visa;
using Ivi.Visa;

namespace test_visa_csharp
{
    static class Program
            {
        static void Main()
        {
          TcpipSession section = new TcpipSession("TCPIP::10.12.255.135::inst0::INSTR");
          IMessageBasedFormattedIO io = section.FormattedIO;
          io.WriteLine("*IDN?");
          string result = io.ReadLine();
          section.Dispose();
          Console.WriteLine(result);
        }
    }
}
```

7. NI-VISA.NET 模拟通道波形重构实例：

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;
using NationalInstruments.Visa;
using Ivi.Visa;
using System.Runtime.InteropServices;
using System.Threading;



[StructLayout(LayoutKind.Sequential, Pack = 2)]
public struct RT_TIME_OLD
{
    public double seconds;    //8
    public char minutes;      //1
    public char hours;        //1
    public char days;         //1
    public char months;       //1
    public short year;        //2
    public short dummy;       //2
};

[StructLayout(LayoutKind.Sequential, Pack = 2)]
public struct WD_PARAM
{
    /* MANDATORY PART */
    [MarshalAs(UnmanagedType.ByValTStr, SizeConst = 16)]
    public string descriptor_name;          /* will contain "WAVEDESC"      char [16] */
    [MarshalAs(UnmanagedType.ByValTStr, SizeConst = 16)]
```

```
public string template_name;    //char [16]

public short comm_type;
public short comm_order;

/* DESCRIPTOR PART */
public Int32 wave_desc_length;    //4
public Int32 user_text_length;
public Int32 res_desc1;

public Int32 trig_time_array;
public Int32 ris_time_array;
public Int32 res_array1;


/* ARRAY PART */
public Int32 wave_array_1;
public Int32 wave_array_2;                /* this is 0 if not present */
public Int32 res_array2;
public Int32 res_array3;

[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 16)]
public string instrument_name;//char [16]

public UInt32 instrument_number;    //4

[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 16)]
public string trace_label;    //char [16]
```

```
public Int32 reserved_data_count;      /* Equal to internalDescriptor.data_count. */
/* Currently not documented to users.          */
/* Necessary because                           */
/* internalDescriptor.dat_count can't be      */
/* recomputed from other fields of             */
/* externalDescriptor;                         */
/* wrong internalDescriptor.data_count         */
/* is detrimental to automated testing         */
/* of wf? / wf and card:sto / rec.             */

/* The following variables describe the waveform type and the time at
      which the waveform was generated.*/

public Int32 wave_array_count;          /* actual nbr of data items in simple array
                                           = nominal + extra points required.    This may
                                           be necessary for expansion but should NOT
                                           be used for calculations --> use last_valid
                                           - first_valid for determining the number
                                           of points in computations and display */

public Int32 points_per_screen;    /* nominal number of points in the waveform */

public Int32 first_valid;           /* ;    count of number of points to skip
                                              ;; before first good point
                                              ;; FIRST_VALID_POINT = 0
                                              ;; for normal waveforms.                  */
```

```
/* index to first valid point in data array.
     Point 0 of the data array always maps to
     the first pixel before the left edge of the
     screen.   This means that if the waveform
     does not actual start at that edge, the data
     in the beginning of the array is invalid -
     however, no assumption is made that it is
     zero. For an unexpanded waveform, this
     is set to 0 */

public Int32 last_valid;              /* ;   index of last good data point
                                   ;; in record before padding (blanking)
                                   ;; was started.
                                   ;; LAST_VALID_POINT = WAVE_ARRAY_COUNT-1
                                   ;; except for aborted sequence
                                   ;; and rollmode acquisitions              */

public Int32 first_point;

public Int32 sparsing_factor;

public Int32 segment_no;

public Int32 subarray_count;       /* for Sequence, acquired segment count,
                                         between 0 and NOM_SUBARRAY_COUNT */

public Int32 sweeps_per_acq;       /* for Averages and Extrema:
                                              number of sweeps accumulated */
```

```
        public short points_per_pair;       /* for Peak Detect only */
        public short pair_offset;           /* for Peak Detect only */

        public float vertical_gain;         /* total gain of waveform, units per lsb */

        public float vertical_offset;       /* total vertical offset of waveform */

        public float code_per_div;          // float representation of the max integer (byte or
word) value used
        // corresponds to verFrameStop
        public float reserved;                                  // float representation of the min
integer (byte or word) value used
        // corresponds to verFrameStart

        public short nominal_bits;      /* estimated: Ch1, Ch2: 8; averaging: 12, etc */

        public short nom_subarray_count;        /* for Sequence, nominal segment count
                                                   else 1 */

        public float horizontal_interval; /* this corresponds to the sampling interval
                                             (ie. time/point) for time domain waveforms
                                             and freq/point for FFT's. It is the nominal
                                             time between successive points in the data
                                             In RIS, it is the equivalent sampling rate*/

        public double horizontal_offset;    /* this corresponds to trigger offset in time
                                               domain    for zero'th sweep of trigger,
```

```
                                        seconds from trigger to zero'th data point
                                        (ie. actual trigger delay) */

        public double pixel_offset;         /* from trigger to zero'th pixel of display
                                        segment in time domain. measured in
seconds
                                        (ie. nominal trigger delay) */

        [MarshalAs(UnmanagedType.ByValTStr, SizeConst = 48)]
        public string vertunit; //char [48]
        [MarshalAs(UnmanagedType.ByValTStr, SizeConst = 48)]
        public string horunit;//char [48]

        public float horiz_uncertainty;  // this correspond to the TDC resolution
        // in seconds.
        public RT_TIME_OLD trigger_time;          /* also for sequence waveforms */

        public float acq_duration;          /* time in seconds; for sequence & RIS */

        public short ca_record_type;        /* type of waveform (see enum above) */

        public short processing_done;       /* indication of whether any processing done
                                        see enum (or bit pattern) above */

        public short reserved5;

        public short ris_sweeps;            /* ; for RIS, the number of sweeps
                                        ;; else 1                                */
```

```
        /* the following information should be for history only */

        public short time_base;                 /* this is the enumerated time/div */
        public short vertical_coupling;

        public float probe_attenuation;

        public short fixed_vertical_gain;
        public short band_width_limit;

        public float vertical_vernier;                  /* needed for waveform display */
        public float acquisition_vertical_offset; /* divisions */

        public short wave_source;
};



namespace test_visa_csharp
{
    static class Program
    {
        /// <summary>
        /// </summary>
        [STAThread]
        static void Main()
        {
            /*
            //Connect Via TCPIP
```

```
            TcpipSession section = new TcpipSession("TCPIP::10.12.58.1::inst0::INSTR");
            IMessageBasedFormattedIO io = section.FormattedIO;
            io.WriteLine("*IDN?");
            string result = io.ReadLine();
            section.Dispose();
            Console.WriteLine(result);
            */
            //Connect Via TCPIP or USB
            var rmSession = new ResourceManager();
            MessageBasedSession mbSession =
(MessageBasedSession)rmSession.Open("TCPIP::10.12.60.1::inst0::INSTR");
            sds_initialize(mbSession);
            mbSession.RawIO.Write("TRIG:MODE SINGLE");
            bool state = WaitAcquisitionComplete(mbSession);
            if (state)
            {
                float[] c1_volt = sds_fetchwaveform(mbSession,"C1");
            }
            mbSession.Dispose();
        }

        public static void sds_initialize(MessageBasedSession mbSession)
        {
            mbSession.RawIO.Write("*IDN?");
            string result = mbSession.RawIO.ReadString();
            mbSession.RawIO.Write("CHDR OFF");
            Console.WriteLine(result);
        }
```

```
public static bool WaitAcquisitionComplete(MessageBasedSession mbSession)
{
    mbSession.RawIO.Write("TRIG:MODE?");
    string mode = mbSession.RawIO.ReadString();
    if (mode.Contains("SINGle"))
    {
        while (true)
        {
            mbSession.RawIO.Write("TRIG:STAT?");
            string result = mbSession.RawIO.ReadString();
            if (result.Contains("Stop"))
            {
                Console.WriteLine("Single Acquisition finished");
                return true;
            }
        }
    }
    else
    {
        while (true)
        {
            mbSession.RawIO.Write("INR?");
            string result = mbSession.RawIO.ReadString();
            Int16 state = Convert.ToInt16(result);
            if ((state & 0x01) == 1)
            {
                Console.WriteLine("Acquisition finished");
                return true;
            }
```

```
            }
        }
        return false;
    }

    public static float[] sds_fetchwaveform(MessageBasedSession mbSession,string channel)
    {
        string src_cmd = string.Format("WAV:SOUR {0}", channel);
        mbSession.RawIO.Write(src_cmd);
        mbSession.RawIO.Write("WAV:STAR 0");
        mbSession.RawIO.Write("WAV:PRE?");
        byte[] WaveParamBytes = mbSession.FormattedIO.ReadBinaryBlockOfByte();
        var wp = ConvertToWaveFormParam(WaveParamBytes);
        Console.WriteLine(wp.code_per_div);
        mbSession.Clear();

        mbSession.RawIO.Write("WAVeform:MAXPoint?");
        string result = mbSession.RawIO.ReadString();
        float one_piece_num = (float)Convert.ToSingle(result);
        mbSession.RawIO.Write("ACQ:POIN?");
        result = mbSession.RawIO.ReadString();
        int point = (int)Convert.ToSingle(result);

        Int16[] WaveDataInt16All = new Int16[point];
        sbyte[] WaveDataAll = new sbyte[point];
        float[] volt = new float[WaveDataAll.Length];

        if (wp.nominal_bits > 8)
        {
```

```
                    WaveDataInt16All = GetWaveData16bit(mbSession, point,
one_piece_num);
                    for (int i = 0; i < WaveDataInt16All.Length; i++)
                    {
                        volt[i] = (WaveDataInt16All[i] / wp.code_per_div * wp.vertical_gain -
wp.vertical_offset) * wp.probe_attenuation;
                    }
                }
                else
                {
                    WaveDataAll = GetWaveData8bit(mbSession, point, one_piece_num);
                    for (int i = 0; i < WaveDataAll.Length; i++)
                    {
                        volt[i] = (WaveDataAll[i] / wp.code_per_div * wp.vertical_gain -
wp.vertical_offset) * wp.probe_attenuation;
                    }
                }
                return volt;
            }

            public static sbyte[] GetWaveData8bit(MessageBasedSession mbSession, int
point, float one_piece_num)
            {
                sbyte[] WaveDataAll = new sbyte[point];

                int read_times = (int)System.Math.Ceiling((point / one_piece_num));
                mbSession.RawIO.Write("WAV:WIDT BYTE");
                DateTime dt1 = DateTime.Now;
                for (int i = 0; i < read_times; i++)
                {
                    int start = (int)(i * one_piece_num);
```

```
                string start_cmd = string.Format("WAVeform:STARt {0}", start);
                mbSession.RawIO.Write(start_cmd);
                DateTime dt3 = DateTime.Now;
                mbSession.RawIO.Write("WAV:DATA?");
                sbyte[] WaveDataBytes = mbSession.FormattedIO.ReadBinaryBlockOfSByte();
                Console.WriteLine(WaveDataBytes.Length);
                Array.Copy(WaveDataBytes, 0, WaveDataAll, start, WaveDataBytes.Length);
            }
            mbSession.Clear();
            DateTime dt2 = DateTime.Now;
            TimeSpan ts = dt2.Subtract(dt1);
            Console.WriteLine("used time = {0}ms", ts.TotalMilliseconds);
            Console.WriteLine("WaveDataAll.Length = {0}", WaveDataAll.Length);
            return WaveDataAll;
        }

        public static Int16[] GetWaveData16bit(MessageBasedSession mbSession, int point, float one_piece_num)
        {
            Int16[] WaveDataInt16All = new Int16[point];

            mbSession.RawIO.Write("WAV:WIDT WORD");
            mbSession.RawIO.Write("WAV:BYTEORDER MSB");
            int read_times = (int)System.Math.Ceiling((point / one_piece_num));
            DateTime dt1 = DateTime.Now;
            for (int i = 0; i < read_times; i++)
            {
                int start = (int)(i * one_piece_num);
```

```
                    string start_cmd = string.Format("WAVeform:STARt {0}", start);
                    mbSession.RawIO.Write(start_cmd);
                    mbSession.RawIO.Write("WAV:DATA?");
                    Int16[] WaveDataBytes =
mbSession.FormattedIO.ReadBinaryBlockOfInt16();
                    Array.Copy(WaveDataBytes, 0, WaveDataInt16All, start,
WaveDataBytes.Length);
                }
                mbSession.Clear();
                DateTime dt2 = DateTime.Now;
                TimeSpan ts = dt2.Subtract(dt1);
                Console.WriteLine("used time = {0}ms", ts.TotalMilliseconds);
                Console.WriteLine("WaveDataInt16All.Length = {0}",
WaveDataInt16All.Length);
                return WaveDataInt16All;
            }

            public static WD_PARAM ConvertToWaveFormParam(byte[] parambuff)
            {
                int struczise = Marshal.SizeOf(typeof(WD_PARAM));
                IntPtr ptemp = Marshal.AllocHGlobal(struczise);
                Marshal.Copy(parambuff, 0, ptemp, struczise);
                WD_PARAM wd = (WD_PARAM)Marshal.PtrToStructure(ptemp,
typeof(WD_PARAM));
                Marshal.FreeHGlobal(ptemp);
                return wd;
            }
        }
    }
```

6.2 基于 Socket 编程

Socket 是计算机网络中允许应用程序通过网络硬件和操作系统构建的标准网络协议进行通信的一种基本通信技术，通过固定端口号在设备和计算机之间进行双向通信。

注意：程序中的 SCPI 命令的字符串必须以“\n”字符（换行）作为结尾。

Python 实例

Python 提供一个底层网络模块，可以提供对 Socket 接口的访问，端口为 5025。可以为 Socket 接口编写 Python 脚本来执行各种测试和测量任务。

环境：Windows7 32 位系统，Python v2.7.5。

描述：打开 Socket，发送询问 SCPI 并循环执行 10 次，然后关闭 Socket。

下面是脚本的代码：

```
#!/usr/bin/env python
#-*- coding:utf-8 -*-
#------------------------------------------------------------------------------
# The short script is a example that open a socket, sends a query,
# print the return message and closes the socket.
#------------------------------------------------------------------------------
import socket      # for sockets
import sys    # for exit
import time # for sleep
#------------------------------------------------------------------------------
remote_ip = "10.12.255.209"    # should match the instrument's IP address
port = 5025 # the port number of the instrument service
count = 0
```

```
def SocketConnect():
    try:
        #create an AF_INET, STREAM socket (TCP)
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    except socket.error:
        print ('Failed to create socket.')
        sys.exit();
    try:
        #Connect to remote server
        s.connect((remote_ip , port))
    except socket.error:
        print ('failed to connect to ip ' + remote_ip)
    return s

def SocketQuery(Sock, cmd):
    try :
        #Send cmd string
        Sock.sendall(cmd)
        Sock.sendall(b'\n')
        time.sleep(1)
    except socket.error:
        #Send failed
        print ('Send failed')
        sys.exit()
    reply = Sock.recv(4096)
    return reply

def SocketClose(Sock):
    #close the socket
```

```
    Sock.close()
    time.sleep(.300)

def main():
    global remote_ip
    global port
    global count

    # Body: send the SCPI commands *IDN? 10 times and print the return message
    s = SocketConnect()
    for i in range(10):
        qStr = SocketQuery(s, b'*IDN?')
        print (str(count) + ":: " + str(qStr))
        count = count + 1
    SocketClose(s)
    input('Press "Enter" to exit')

if __name__ == '__main__':
    proc = main()
```

C 实例

```
int MySocket;
if((MySocket=socket(PF_INET,SOCK_STREAM,0))==-1)
{
    exit(1);
}
struct in_addr
{
    unsigned long s_addr;
};
struct sockaddr_in
{
    short int sin_family; // Address family
    unsigned short int sin_port; // Port number
    struct in_addr sin_addr; // Internet address
    unsigned char sin_zero[8]; // Padding
};
struct sockaddr_in MyAddress;

// Initialize the whole structure to zero
memset(&MyAddress,0,sizeof(struct sockaddr_in));
// Then set the individual fields
MyAddress.sin_family=PF_INET; // IPv4
MyAddress.sin_port=htons(5025); // Port number used by most instruments
MyAddress.sin_addr.s_addr=inet_addr(ntsddr_in));        // IP Address

// Establish TCP connection
if(connect(MySocket,(struct sockaddr*)&MyAddress,sizeof(struct sockaddr_in))==-1)
```

```
{
    exit(1);
}

// Send SCPI command
if(send(MySocket,ands,sizeof(t_addr(
{
    exit(1);
}

// Read response
char buffer[200];
int actual;
if((actual=recv(MySocket,&buffer[0],200,0))==-1)
{
    exit(1);
}
buffer[actual]= 0; // Add zero character (C string)
printf(d zero character (C string)],2

// Close socket
if(close(MySocket)==-1)
{
    exit(1);
}
```

6.3 常用命令实例

本节列出了常用命令的编程实例。

环境：Windows7 32 位系统、Python v3.6.5、pyvisa-1.9、Matplotlib-3.1.1。

注意:

使用 VISA 库时，以下设置需注意：

- 设置 I/O 缓冲区大小

  例如：对于命令:WAVeform:DATA，读取缓冲区的大小取决于数据点的数量。当需要分片读取波形时，每片大小因机型而异，可使用:WAVeform:MAXPoint 查询。

- 设置超时值

  超时值与网络速度或 USB 传输速度有关，请自行评估。默认值通常为 2s。

模拟通道波形重构实例

```
# Import modules.
# ----------------------------------------------------------
import visa
import pylab as pl
import struct
import math
import gc


# Global variables
# (Modify the following global variables according to the model).
# ----------------------------------------------------------
SDS_RSC = "TCPIP0::10.12.59.1::inst0::INSTR"
CHANNEL = "C2"
```

```
HORI_NUM = 10
tdiv_enum = [200e-12,500e-12, 1e-9,\
                2e-9, 5e-9, 10e-9, 20e-9, 50e-9, 100e-9, 200e-9, 500e-9, \
                1e-6, 2e-6, 5e-6, 10e-6, 20e-6, 50e-6, 100e-6, 200e-6, 500e-6, \
                1e-3, 2e-3, 5e-3, 10e-3, 20e-3, 50e-3, 100e-3, 200e-3, 500e-3, \
                1, 2, 5, 10, 20, 50, 100, 200, 500, 1000]


# ============================================================
# main_desc:Analyzing waveform parameters from data blocks
# ============================================================
def main_desc(recv):
    param_addr_type={"data_bytes":[0x3c,"i"],
                     "point_num":[0x74,'i'],
                     "fp":[0x84,'i'],
                     "sp":[0x88,'i'],
                     "vdiv":[0x9c,'f'],
                     "offset":[0xa0,'f'],
                     "code":[0xa4,'f'],
                     "adc_bit":[0xac,'h'],
                     "interval":[0xb0,'f'],
                     "delay":[0xb4,'d'],
                     "tdiv":[0x144,'h'],
                     "probe":[0x148,'f']}
    data_byte = {"i": 4, "f": 4, "h": 2, "d": 8}
    param_val ={}

    for key,addr_type in param_addr_type.items():
        addr_start = addr_type[0]
```

```
            format = addr_type[1]
            bytes = recv[addr_start:addr_start+data_byte[format]]
            param_val[key] = struct.unpack(format, bytes)[0]
        param_val["tdiv"] = tdiv_enum[param_val["tdiv"]]
        param_val["vdiv"] = param_val["vdiv"]*param_val["probe"]
        param_val["offset"] = param_val["offset"]*param_val["probe"]
        return param_val


# ============================================================
# Main program:
# ============================================================
def main_wf_data():
        _rm = visa.ResourceManager()
        sds = _rm.open_resource(SDS_RSC)
        sds.timeout = 2000    # default value is 2000(2s)
        sds.chunk_size = 20 * 1024 * 1024    # default value is 20*1024(20k bytes)

        # Get the channel waveform parameter data blocks and parse them
        sds.write(":WAVeform:STARt 0")
        sds.write("WAV:SOUR {}".format(CHANNEL))
        sds.write("WAV:PREamble?")
        recv_all = sds.read_raw()
        recv = recv_all[recv_all.find(b'#') + 11:]
        print(len(recv))
        param_dic = main_desc(recv)
        print(param_dic)

        # Get the waveform points and confirm the number of waveform slice reads
```

```
    points = param_dic["point_num"]
    one_piece_num = float(sds.query(":WAVeform:MAXPoint?").strip())
    read_times = math.ceil(points / one_piece_num)
    #Set the number of read points per slice, if the waveform points is greater than the maximum
number of slice reads
    if points > one_piece_num:
        sds.write(":WAVeform:POINt {}".format(one_piece_num))
    # Choose the format of the data returned
    sds.write(":WAVeform:WIDTh BYTE")
    if param_dic[“adc_bit”] > 8:
        sds.write(":WAVeform:WIDTh WORD")

    #Get the waveform data for each slice
    recv_byte = b''
    for i in range(0, read_times):
        start = i * one_piece_num
        #Set the starting point of each slice
        sds.write(":WAVeform:STARt {}".format(start))
        #Get the waveform data of each slice
        sds.write("WAV:DATA?")
        recv_rtn = sds.read_raw()
        #Splice each waveform data based on data block information
        block_start = recv_rtn.find(b'#')
        data_digit = int(recv_rtn[block_start + 1:block_start + 2])
        data_start = block_start + 2 + data_digit
        data_len = int(recv_rtn[block_start + 2:data_start])
        recv_byte += recv_rtn[data_start:data_start + data_len]
    # Unpack signed byte data.
```

```
    if param_dic[“adc_bit”] > 8:
        convert_data = struct.unpack("%dh"%points, recv_byte)
    else:
        convert_data = struct.unpack("%db"%points, recv_byte)
    del recv_byte
    gc.collect()
    #Calculate the voltage value and time value
    time_value = []
    volt_value = []
    for idx in range(0, len(convert_data)):
        volt_value.append(convert_data[idx]/param_dic["code"]*param_dic["vdiv"]-
param_dic["offset"])
        time_data = - (param_dic[“tdiv”] * HORI_NUM / 2) + idx * param_dic[“interval”] +
param_dic[“delay”]
        time_value.append(time_data)
    print(len(volt_value))
    #Draw Waveform
    pl.figure(figsize=(7, 5))
    pl.plot(time_value, volt_value, markersize=2, label=u"Y-T")
    pl.legend()
    pl.grid()
    pl.show()


    if __name__ == '__main__':
        main_wf_data()
```

数字通道波形重构实例

```
# Import modules.
# ----------------------------------------------------
import visa
import pylab as pl
import struct


# Global variables
# (Modify the following global variables according to the model).
# ----------------------------------------------------
SDS_RSC = "TCPIP0::10.12.59.1::inst0::INSTR"
CHANNEL = "D0"
HORI_NUM = 10
tdiv_enum = [100e-12, 200e-12, 500e-12, \
                    1e-9, 2e-9, 5e-9, 10e-9, 20e-9, 50e-9, 100e-9, 200e-9, 500e-9, \
                    1e-6, 2e-6, 5e-6, 10e-6, 20e-6, 50e-6, 100e-6, 200e-6, 500e-6, \
                    1e-3, 2e-3, 5e-3, 10e-3, 20e-3, 50e-3, 100e-3, 200e-3, 500e-3, \
                    1, 2, 5, 10, 20, 50, 100, 200, 500, 1000]


# ====================================================
# get_char_bit:Get each bit from a byte
# ====================================================
def get_char_bit(char,n):
    return (char >> n) & 1


# ====================================================
# main_desc:Analyzing waveform parameters from data blocks
```

```
# ============================================================
def main_desc(recv):
    param_addr_type = {"data_bytes": [0x3c, "i"],
                       "point_num": [0x74, 'i'],
                       "interval": [0xb0, 'f'],
                       "delay": [0xb4, 'd'],
                       "tdiv": [0x144, 'h']}
    data_byte = {"i": 4, "f": 4, "h": 2, "d": 8}
    param_val = {}

    for key, addr_type in param_addr_type.items():
        addr_start = addr_type[0]
        format = addr_type[1]
        bytes = recv[addr_start:addr_start + data_byte[format]]
        param_val[key] = struct.unpack(format, bytes)[0]
    param_val["tdiv"] = tdiv_enum[param_val["tdiv"]]
    return param_val


# ============================================================
# Main program:
# ============================================================
def main_digital_wf_data():
    _rm = visa.ResourceManager()
    sds = _rm.open_resource(SDS_RSC)
    # Get the digital channel waveform parameter data blocks and parse them
    sds.write("WAV:SOUR {}".format(CHANNEL))
    sds.write("WAV:PREamble?")
    recv_all = sds.read_raw()
```

```
        recv = recv_all[recv_all.find(b'#')+11:]
        param_val = main_desc(recv)
        print(param_val)
        # Get the waveform data
        sds.write("WAV:DATA?")
        recv_rtn = sds.read_raw()
        block_start = recv_rtn.find(b'#')
        data_digit = int(recv_rtn[block_start + 1:block_start + 2])
        data_start = block_start + 2 + data_digit
        data_len = int(recv_rtn[block_start + 2:data_start])
        data = recv_rtn[data_start:data_start + data_len]
        # Calculate the voltage value and time value
        volt_value = []
        for char in data:
            for i in range(0,8):
                volt_value.append(get_char_bit(char,i))
        print(len(volt_value))
        time_value = []
        for idx in range(0,len(volt_value)):
            time_data = - (param_val["tdiv"] * HORI_NUM / 2) + idx * param_val["interval"] +
param_val["delay"]
            time_value.append(time_data)
        # Draw Waveform
        pl.figure(figsize=(7,5))
        pl.ylim(-1,2)
        pl.plot(time_value,volt_value,markersize=2,label=u"Y-T")
        pl.legend()
        pl.grid()
```

```
    pl.show()


if __name__=='__main__':
    main_digital_wf_data()
```

数学函数 FFT 波形重构实例

```
# Import modules.
# ------------------------------------------------------
import visa
import pylab as pl
import struct
import math
import gc


    # Global variables
# ------------------------------------------------------
SDS_RSC = "TCPIP0::10.12.255.127::inst0::INSTR"
FUNC = "FUNC1"



    # ======================================================
# main_desc:Analyzing waveform parameters from data blocks
# ======================================================
def main_desc(recv):
    param_addr_type={"data_bytes":[0x3c,"i"],
                     "point_num":[0x74,'i'],
                     "fp":[0x84,'i'],
                     "sp":[0x88,'i'],
```

```
                    "vdiv":[0x9c,'f'],
                    "offset":[0xa0,'f'],
                    "code":[0xa4,'f'],
                    "adc_bit":[0xac,'h'],
                    "interval":[0xb0,'f'],
                    "delay":[0xb4,'d'],
                    "tdiv":[0x144,'h'],
                    "probe":[0x148,'f']}
    data_byte = {"i": 4, "f": 4, "h": 2, "d": 8}
    param_val ={}

    for key,addr_type in param_addr_type.items():
        addr_start = addr_type[0]
        format = addr_type[1]
        bytes = recv[addr_start:addr_start+data_byte[format]]
        param_val[key] = struct.unpack(format, bytes)[0]
    param_val["tdiv"] = tdiv_enum[param_val["tdiv"]]
    param_val["vdiv"] = param_val["vdiv"]*param_val["probe"]
    param_val["offset"] = param_val["offset"]*param_val["probe"]
    return param_val


# ============================================================
# Main program:
# ============================================================
def main_fft_data():
    _rm = visa.ResourceManager()
    sds = _rm.open_resource(SDS_RSC)
    # Get the channel waveform parameter data blocks and parse them
```

```
        sds.write("WAV:SOUR F1")
        sds.write("WAV:PREamble?")
        recv_all = sds.read_raw()
        recv = recv_all[recv_all.find(b'#') + 11:]
        param_val = main_desc(recv)
        display_len = int(param_val["delay"]/ param_val["interval"])+1
        unit = sds.query("{}:FFT:UNIT?".format(FUNC)).strip()      # {Vrms,DBm,DBVrms}
        if unit == "DBm":
            load = float(sds.query("{}:FFT:LOAD?".format(FUNC)).strip())
        mode = sds.query("{}:FFT:MODE?".format(FUNC)).strip()    #
    {NORMal|MAXHold|AVERage[,num]}
        # Get the waveform data
        sds.write("WAV:DATA?")
        recv_all = sds.read_raw().rstrip()
        block_start = recv_all.find(b'#')
        data_digit = int(recv_all[block_start + 1:block_start + 2])
        data_start = block_start + 2 + data_digit
        recv = recv_all[data_start:]
        print(len(recv))
        # Unpack data.
        volt_value = []
        freq_value = []
        len_data = int(len(recv) / 8)    ##采样定理 f/2
        print(len_data)
        print(recv[0:4])
        for i in range(0, len_data):
            data_rel = struct.unpack("f", recv[8 * i:8 * i + 4])
            data_imag = struct.unpack("f", recv[8 * i + 4:8 * i + 8])
```

```
            data_rel = list(data_rel)[0]
            data_imag = list(data_imag)[0]
            if mode == "NORMal":
                data_float = math.sqrt(pow(float(data_rel), 2) + pow(float(data_imag), 2))
            else:
                data_float = float(data_rel)
            if unit == "DBVrms":
                data_float = 20*math.log10(data_float)
            elif unit == "DBm":
                data_float = 10 * math.log10(data_float*data_float/load/1E-3)
            volt_value.append(data_float)
            freq_value.append(i*param_val["interval"])
        # Draw Waveform
        pl.figure(figsize=(7, 5))
        pl.plot(freq_value, volt_value, markersize=2)
        pl.legend()
        pl.grid()
        pl.show()


        if __name__ == '__main__':
        main_fft_data()
```

分段采集波形重构实例

```
# Import modules.
# -----------------------------------------------------------
import visa
```

```
import pylab as pl
import time as t
import math
import struct
import gc


# Global variables
# (Modify the following global variables according to the model).
# ----------------------------------------------------
CHANNEL = "C2"
HORI_NUM = 10
TDIV_ENUM = [100e-12, 200e-12, 500e-12, \
             1e-9, 2e-9, 5e-9, 10e-9, 20e-9, 50e-9, 100e-9, 200e-9, 500e-9, \
             1e-6, 2e-6, 5e-6, 10e-6, 20e-6, 50e-6, 100e-6, 200e-6, 500e-6, \
             1e-3, 2e-3, 5e-3, 10e-3, 20e-3, 50e-3, 100e-3, 200e-3, 500e-3, \
             1, 2, 5, 10, 20, 50, 100, 200, 500, 1000]


# ============================================================
# main_wf_desc:Analyzing waveform parameters from data blocks
# ============================================================
def main_wf_desc(recv):
    param_addr_type = {"width":[0x20,"h"],#01-16bit,00-8bit
                       "order":[0x22,"h"],#01-MSB,00-LSB
                       "data_bytes": [0x3c, "i"],
                       "one_frame_pts":[0x74, 'i'],#pts of single frame,maybe bigger than
12.5M
                       "fp": [0x84, 'i'],
                       "sp": [0x88, 'i'],
```

```
                    "sn":[0xae,"h"],
                    "read_frame":[0x90,"i"],#all sequence frames number return by this
command
                    "sum_frame":[0x94,"i"],#all sequence frames number acquired
                    "vdiv": [0x9c, 'f'],
                    "offset": [0xa0, 'f'],
                    "code": [0xa4, 'f'],
                    "adc_bit": [0xac, 'h'],
                    "interval": [0xb0, 'f'],
                    "delay": [0xb4, 'd'],
                    "tdiv": [0x144, 'h'],
                    "probe": [0x148, 'f']}
        data_byte = {"i": 4, "f": 4, "h": 2, "d": 8}
        param_val = {}
        for key, addr_type in param_addr_type.items():
            addr_start = addr_type[0]
            format = addr_type[1]
            bytes = recv[addr_start:addr_start + data_byte[format]]
            param_val[key] = struct.unpack(format, bytes)[0]
        param_val["tdiv"] = TDIV_ENUM[param_val["tdiv"]]
        param_val["vdiv"] = param_val["vdiv"] * param_val["probe"]
        param_val["offset"] = param_val["offset"] * param_val["probe"]
        return param_val


# ============================================================
# main_time_stamp_deal:Parsing timestamps from binary blocks
# ============================================================
def main_time_stamp_deal(time):
```

```
    seconds = time[0x00:0x08]          # type:long double
    minutes = time[0x08:0x09]          # type:char
    hours = time[0x09:0x0a]            # type:char
    days = time[0x0a:0x0b]       # type:char
    months = time[0x0b:0x0c]     # type:char
    year = time[0x0c:0x0e]       # type:short

    seconds = struct.unpack('d',seconds)[0]
    minutes = struct.unpack('c', minutes)[0]
    hours = struct.unpack('c', hours)[0]
    days = struct.unpack('c', days)[0]
    months = struct.unpack('c', months)[0]
    year = struct.unpack('h', year)[0]
    months = int.from_bytes(months, byteorder='big', signed=False)
    days = int.from_bytes(days, byteorder='big', signed=False)
    hours = int.from_bytes(hours, byteorder='big', signed=False)
    minutes = int.from_bytes(minutes, byteorder='big', signed=False)
    print("{}/{}/{},{}:{}:{}".format(year,months,days,hours,minutes,seconds))


# ============================================================
# Main program: Read data of all sequence frame.
# When total points num (single_frame_pts * frame_num) is
# bigger than 12.5Mpts, you have to read more than one time.
# Frames number and points number readed this time will be
# saved in the head parameter, see main_wf_desc.
# ============================================================
def main_all_frame(sds):
    sds.write(":WAVeform:SOURce {}".format(CHANNEL))
```

```
sds.write(":WAVeform:STARt 0")
sds.write(":WAVeform:POINt 0")
sds.write(":WAVeform:SEQUence 0,0")
sds.timeout = 2000 #default value is 2000(2s)
sds.chunk_size = 20*1024*1024 #default value is 20*1024(20k bytes)

sds.write(":WAVeform:PREamble?")
recv_all = sds.read_raw()
recv = recv_all[recv_all.find(b'#')+11:]
print(len(recv))
param_dic = main_wf_desc(recv)
read_times = math.ceil(param_dic["sum_frame"]/param_dic["read_frame"])
print("read_times=",read_times)
one_piece_num = float(sds.query(":WAVeform:MAXPoint?").strip())

for i in range(0,read_times):
    if i+1 == read_times: :#frame num of last read time
        read_frame = param_dic["sum_frame"] -(read_times-1)*param_dic["read_frame"]
    else:
        read_frame = param_dic["read_frame"]
    sds.write(":WAVeform:SEQUence {},{}".format(0,read_frame*i+1))
    sds.write(":WAVeform:PREamble?")
    recv_rtn = sds.read_raw()
    recv_desc = recv_rtn[recv_rtn.find(b'#')+11:]
    time_stamp = recv_desc[346:]

    if param_dic["adc_bit"] > 8:
```

```
            sds.write(":WAVeform:WIDTh WORD")
        sds.write(":WAVeform:DATA?")
        recv_rtn = sds.read_raw()
        block_start = recv_rtn.find(b'#')
        data_digit = int(recv_rtn[block_start + 1:block_start + 2])
        data_start = block_start + 2 + data_digit
        data_len = int(recv_rtn[block_start + 2:data_start])
        recv = recv_rtn[data_start:data_start + data_len]

        for j in range(0, param_dic["read_frame"]):
            time = time_stamp[16*j:16*(j+1)]#timestamp spends 16 bytes
            main_time_stamp_deal(time)
            if param_dic["adc_bit"] > 8:
                start = int(j * param_dic["one_frame_pts"]*2)
                end = int((j + 1) * param_dic["one_frame_pts"]*2)
                convert_data=struct.unpack("%dh"%param_dic["one_frame_pts"],
recv[start:end])
            else:
                start = int(j*param_dic["one_frame_pts"])
                end = int((j+1)* param_dic["one_frame_pts"])
                convert_data=struct.unpack("%db" %param_dic["one_frame_pts"],
recv[start:end])

            volt_value = []
            time_value = []
            for idx in range(0,len(convert_data)):
                volt_value.append(convert_data[idx] / param_dic["code"] * param_dic["vdiv"] -
param_dic["offset"])
```

```
                time_value.append(- (param_dic["tdiv"] * HORI_NUM / 2) + idx *
param_dic["interval"] + param_dic["delay"])

            print('Data convert finish,start to draw!')
            pl.figure(figsize=(7,5))
            pl.plot(time_value,volt_value,markersize=2,label=u"Y-T")
            pl.legend()
            pl.grid()
            pl.show()
            pl.close()
            del volt_value,time_value,convert_data
            gc.collect()
        del recv
        gc.collect()

# ============================================================
# Main program: Read data of single frame.
# ============================================================
def main_specify_frame(sds,frame_num):
        sds.write(":WAVeform:SOURce {}".format(CHANNEL))
        sds.write(":WAVeform:STARt 0")
        sds.write(":WAVeform:POINt 0")
        sds.write(":WAVeform:SEQUence {},{}".format(frame_num,0))
        sds.timeout = 2000    # default value is 2000(2s)
        sds.chunk_size = 20 * 1024 * 1024    # default value is 20*1024(20k bytes)

        sds.write(":WAVeform:PREamble?")
        recv_all = sds.read_raw()
```

```
print(len(recv_all))
recv = recv_all[recv_all.find(b'#')+11:]
time_stamp = recv[346:]
main_time_stamp_deal(time_stamp)
param_dic = main_wf_desc(recv)

one_piece_num = float(sds.query(":WAVeform:MAXPoint?").strip())
if param_dic["one_frame_pts"] > one_piece_num:
    sds.write(":WAVeform:POINt {}".format(one_piece_num))
if param_dic["adc_bit"] > 8:
    sds.write(":WAVeform:WIDTh WORD")

read_times = math.ceil(param_dic["one_frame_pts"] / one_piece_num)
data_recv = b''
for i in range(0, read_times):
    start = i * one_piece_num
    sds.write(":WAVeform:STARt {}".format(start))
    sds.write("WAV:DATA?")
    recv_rtn = sds.read_raw()
    block_start = recv_rtn.find(b'#')
    data_digit = int(recv_rtn[block_start + 1:block_start + 2])
    data_start = block_start + 2 + data_digit
    data_len = int(recv_rtn[block_start + 2:data_start])
    data_recv += recv_rtn[data_start: data_start + data_len]
print("len(data_recv)=", len(data_recv))
if param_dic["adc_bit"] > 8:
    convert_data = struct.unpack("%dh" % param_dic["one_frame_pts"], data_recv)
else:
```

```
            convert_data = struct.unpack("%db" % param_dic["one_frame_pts"], data_recv)


        volt_value = []
        time_value = []
        for idx in range(0, len(convert_data)):
            volt_value.append(convert_data[idx] / param_dic["code"] * param_dic["vdiv"] -
param_dic["offset"])
            time_value.append(-(param_dic["tdiv"] * HORI_NUM / 2) + idx * param_dic["interval"] +
param_dic["delay"])


        print('Data convert finish,start to draw!')
        pl.figure(figsize=(7, 5))
        pl.plot(time_value, volt_value, markersize=2, label=u"Y-T")
        pl.legend()
        pl.grid()
        pl.show()
        pl.close()
        del volt_value, time_value, data_recv
        gc.collect()


if __name__=='__main__':
        _rm = visa.ResourceManager()
        sds = _rm.open_resource("TCPIP0::10.12.59.1::inst0::INSTR")
        main_all_frame(sds)
        main_specify_frame(sds, 1)
        sds.close()
```

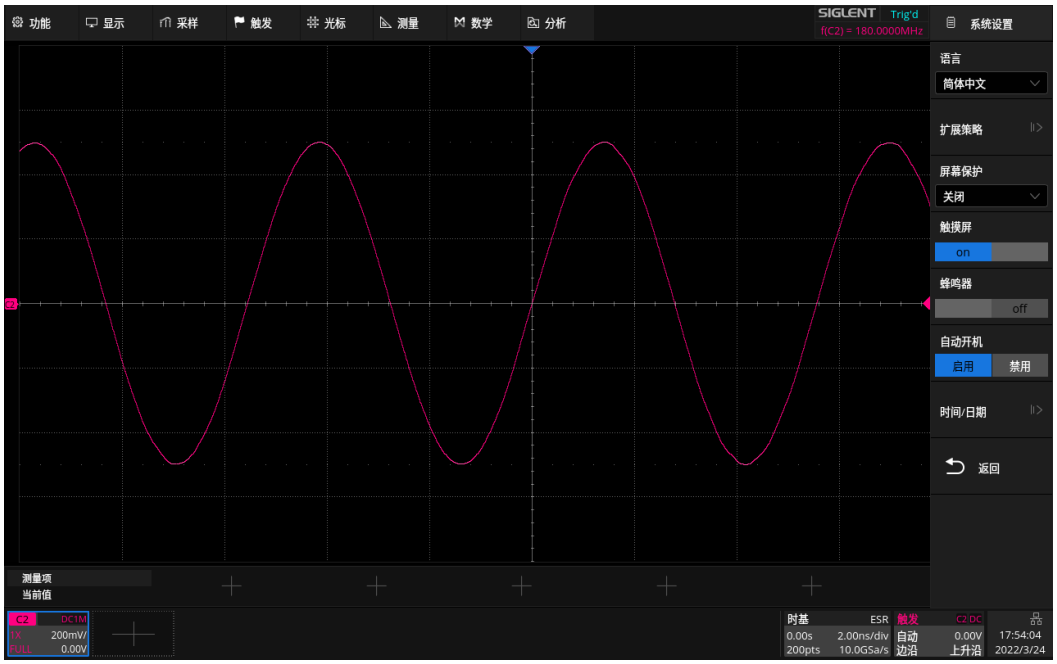
截屏实例

```
import visa


def main():
    _rm = visa.ResourceManager()
    sds = _rm.open_resource("USB0::0xF4EC::0xEE38::0123456789::INSTR")
    sds.chunk_size = 20*1024*1024 #default value is 20*1024(20k bytes)
    file_name = "F:\\SCDP.bmp"
    sds.write("PRIN? BMP")
    result_str = sds.read_raw()
    f = open(file_name,'wb')
    f.write(result_str)
    f.flush()
    f.close()

if __name__=='__main__':
    main()
```

可以打开如下所示截图：


功能
显示
采样
触发
光标
测量
数学
分析
SIGLENT
Trig'd
f(C2) = 180.0000MHz
系统设置
语言
简体中文
扩展策略
屏幕保护
关闭
触摸屏
on
蜂鸣器
off
自动开机
启用
禁用
时间/日期
返回
测量项
当前值
C2
DC1M
200mV/
0.00V
时基
ESR
0.00s
2.00ns/div
200pts
10.0GSa/s
触发
C2 DC
自动
0.00V
边沿
上升沿
17:54:04
2022/3/24

关于鼎阳

鼎阳科技（SIGLENT）是通用电子测试测量仪器领域的行业领军企业，A 股上市公司。

SIGLENT 已经成为全球知名的测试测量仪器品牌。2002 年，鼎阳科技创始人开始专注于示波器研发，2005 年成功研制出鼎阳第一款数字示波器。历经多年发展，鼎阳产品已扩展到数字示波器、手持示波表、频谱分析仪、函数/任意波形发生器、矢量网络分析仪、射频/微波信号源、台式万用表、直流电源、电子负载、精密源表等通用测试测量仪器产品。鼎阳科技是全球极少数能够同时研发、生产、销售数字示波器、信号发生器、频谱分析仪和矢量网络分析仪四大通用电子测试测量仪器主力产品的厂家之一，国家重点“小巨人”企业，同时也是国内主要竞争对手中极少数同时拥有这四大主力产品并且四大主力产品全线进入高端领域的厂家。公司总部位于深圳，在马来西亚槟城州设有生产基地，在美国俄勒冈、德国奥格斯堡、日本东京成立了子公司，在成都成立了分公司，产品远销全球 80 多个国家和地区。

联系我们

深圳市鼎阳科技股份有限公司

全国免费服务热线：400-878-0807

网址：www.siglent.com

声明

SIGLENT 鼎阳 是深圳市鼎阳科技股份有限公司的注册商标，事先未经过允许，不得以任何形式或通过任何方式复制本手册中的任何内容。

本资料中的信息代替原先的此前所有版本。技术数据如有变更，恕不另行通告。

技术许可

对于本文档中描述的硬件和软件，仅在得到许可的情况下才会提供，并且只能根据许可进行使用或复制。