UNI-T

MSO8000HDP 系列高分辨率示波器

20GSa/s | 8GHz | 12-bit | 4Gpts | 1,000,000wfm/s



编程手册 REV.1.0

2026 年 6 月

目录

引言 ---- 16
版权信息 ---- 16
商标信息 ---- 16
文档版本 ---- 16
声明 ---- 17
软件版本 ---- 17
SCPI 指令简介 ---- 18
指令格式 ---- 18
符号说明 ---- 18
参数说明 ---- 18
简写规则 ---- 19
数据返回 ---- 19
SCPI 指令详解 ---- 21
IEEE488.2 通用命令 ---- 21
* IDN ---- 21
* RST ---- 21
SYSTem 指令系统 ---- 22
:RUN ---- 22
:STOP ---- 22
:AUTO ---- 22
:SYSTem:SINGle ---- 22
:SYSTem:RTC ---- 22
:SYSTem:TFORce ---- 23
:SYSTem:CALibration ---- 23
:SYSTem:CLEar ---- 23
:SYSTem:LANGuage ---- 23
:SYSTem:PRTSc ---- 24
:SYSTem:TOUChlock ---- 24
:SYSTem:AUTO:TIMebase ---- 25
:SYSTem:AUTO:VERTical ---- 25
:SYSTem:AUTO:ACQuire ---- 26
:SYSTem:AUTO:TRIGge ---- 26
:SYSTem:AUTO:CHANel:COUPling ---- 27
:SYSTem:AUTO:CHANel:ACTive ---- 27
:SYSTem:AUTO:DISPlay:MODe ---- 27
ACQuire 指令系统 ---- 29

:ACQuire:STATus ---- 29
:ACQuire:TYPe ---- 29
:ACQuire:AVERage ---- 30
:ACQuire:CLKSrc ---- 30
:ACQuire:EXTLocked ---- 31
:ACQuire:FAST ---- 31
:ACQuire:STORage ---- 31
AWG 指令系统 ---- 33
:AWG<n>:LOAD ---- 33
:AWG<n>:ENABle ---- 33
:AWG<n>:MODE ---- 34
:AWG<n>:BASe:WAVe ---- 34
:AWG<n>:BASe:FREQuency ---- 35
:AWG<n>:BASe:ARBitrary ---- 35
:AWG<n>:BASe:PHASe ---- 36
:AWG<n>:BASe:AMPLitude ---- 36
:AWG<n>:BASe:OFFSet ---- 37
:AWG<n>:BASe:HIGH ---- 37
:AWG<n>:BASe:LOW ---- 38
:AWG<n>:PULSe:RISe ---- 38
:AWG<n>:PULSe:FALL ---- 38
:AWG<n>:BASe:DUTY ---- 39
:AWG<n>:BASe:SYMMetry ---- 39
:AWG<n>:MODUlate:MODe ---- 40
:AWG<n>:MODUlate:WAVe ---- 40
:AWG<n>:MODUlate:ARBitrary ---- 41
:AWG<n>:MODUlate:FREQuency ---- 41
:AWG<n>:FM:FREQuency:DEV ---- 42
:AWG<n>:MODUlate:DEPTh ---- 42
:AWG<n>:PM:PHASe:DEV ---- 42
:AWG<n>:SWEep:TYPe ---- 43
:AWG<n>:SWEep:DURatio ---- 43
:AWG<n>:SWEep:FREQuency:STARt ---- 44
:AWG<n>:SWEep:FREQuency:END ---- 44
:AWG<n>:SWEep:TRIGger:SOURce ---- 45
:AWG<n>:SWEep:TRIGger:ENABle ---- 45
:AWG<n>:SWEep:TRIGger:MANualtrig ---- 46
CHANnel 指令系统 ---- 47

:CHANnel<n>:SELect ----47
:CHANnel<n>:BWLimit ----47
:CHANnel<n>:COUPling ---- 48
:CHANnel<n>:DISPlay ---- 48
:CHANnel<n>:INVert ---- 49
:CHANnel<n>:OFFSet ----49
:CHANnel<n>:PROBe:MODel ---- 49
:CHANnel<n>:PROBe:SERailnumber ----50
:CHANnel<n>:PROBe:GAIN:CURRent ----50
:CHANnel<n>:PROBe:GAIN:CUSTom ---- 50
:CHANnel<n>:PROBe:BUTTon ---- 51
:CHANnel<n>:UNIT:ACTive ---- 51
:CHANnel<n>:UNIT ----52
:CHANnel<n>:UNIT:RATio ---- 52
:CHANnel<n>:SCALe ----53
:CHANnel<n>:FINe ---- 53
:CHANnel<n>:BIAS ---- 53
:CHANnel<n>:DELay ---- 54
:CHANnel<n>:COPY ----54
:CHANnel<n>:LABel:ENABle ---- 55
:CHANnel<n>:LABel ----55
:CHANnel<n>:ERES:ENABle ----55
:CHANnel<n>:ERES:BIT ----56
CURSor 指令系统 ----57
:CURSor:ACTive ----57
:CURSor:TRACewave ----57
:CURSor:TYPe ----58
:CURSor:SOURce ---- 58
:CURSor:MOVe:SYNC ---- 59
:CURSor:UNIT:VCURsor ----59
:CURSor:UNIT:HCURsor ---- 59
:CURSor:POSition:MODe ---- 60
:CURSor:CAX ----60
:CURSor:CBX ---- 61
:CURSor:CAY ---- 61
:CURSor:CBY ----62
:CURSor:AXValue ---- 62
:CURSor:BXValue ---- 62

:CURSor:AYValue ---- 63
:CURSor:BYValue ---- 63
:CURSor:DATA:MODe ---- 63
TIMebase 指令系统 ---- 64
:TIMebase:SCALe ---- 64
:TIMebase:DELay ---- 64
TRIGger 指令系统 ---- 65
:TRIGger:TYPe ---- 65
:TRIGger:HOLDoff ---- 65
:TRIGger:MODe ---- 66
:TRIGger:EDGe ---- 67
:TRIGger:SHOLd ---- 70
:TRIGger:PATTern ---- 75
:TRIGger:PWIDth ---- 77
:TRIGger:RUNT ---- 80
:TRIGger:SLOPe ---- 84
:TRIGger:TIMeout ---- 88
:TRIGger:VIDeo ---- 91
:TRIGger:NEDGe ---- 94
:TRIGger:BUS ---- 97
:TRIGger:RS232 ---- 98
:TRIGger:I2C ---- 100
:TRIGger:SPI ---- 103
:TRIGger:CAN ---- 105
:TRIGger:CAN_FD ---- 110
:TRIGger:LIN ---- 115
:TRIGger:FLEXRAY ---- 117
:TRIGger:AUDIOBUS ---- 124
:TRIGger:MIL_STD_1553 ---- 126
:TRIGger:ARINC429 ---- 129
:TRIGger:USB ---- 132
:TRIGger:SENT ---- 133
:TRIGger:SPMi ---- 136
:TRIGger:ETHernet ---- 138
:TRIGger:PSI5 ---- 142
WAVeform 指令系统 ---- 144
:WAVeform:SOURce ---- 144
:WAVeform:MODE ---- 144

:WAVeform:FORMat ---- 145
:WAVeform:STARt ---- 145
:WAVeform:POINts ---- 146
:WAVeform:FRAMe:ID ---- 146
:WAVeform:DATA ---- 146
MEASure 指令系统 ---- 148
:MEASure:QKMEasure:DISPlay ---- 152
:MEASure:QKMEasure:SOURce ---- 152
:MEASure:QKMEasure:DATA ---- 153
:MEASure:ALL:ACTive ---- 153
:MEASure:ULTRaacq ---- 154
:MEASure:GATe ---- 155
:MEASure:INDicator ---- 155
:MEASure:ITEM<n>:DISPlay ---- 155
:MEASure:STATistics:DISPlay ---- 156
:MEASure:STATistics:COUNt ---- 156
:MEASure:STATistics:RESet ---- 157
:MEASure:ITEM<n>:TYPe ---- 157
:MEASure:ITEM<n>:SOURce<ch> ---- 158
:MEASure:ITEM<n>:THRold:TYPe ---- 158
:MEASure:ITEM<n>:THRold:TOP ---- 159
:MEASure:ITEM<n>:THRold:MIDDle ---- 159
:MEASure:ITEM<n>:THRold:BOTTom ---- 160
:MEASure:ITEM<n>:VALue ---- 160
:MEASure:ITEM<n>:COUNter:MAX ---- 160
:MEASure:ITEM<n>:COUNter:MIN ---- 161
:MEASure:ITEM<n>:COUNter:AVG ---- 161
:MEASure:ITEM<n>:COUNter:DEV ---- 161
:MEASure:ITEM<n>:COUNter:POP ---- 162
:MEASure:ITEM<n>:HISTgram ---- 162
:MEASure:ITEM<n>:TRENd ---- 162
:MEASure:ITEM<n>:TRACk ---- 163
MATH 指令系统 ---- 164
:MATH<n>:ACTive ---- 164
:MATH<n>:LABel:ENABle ---- 164
:MATH<n>:LABel ---- 165
:MATH<n>:UNIT:ENABle ---- 165
:MATH<n>:UNIT ---- 165

:MATH<n>:TYPe ---- 166
:MATH<n>:VERTical:OFFSet ---- 166
:MATH<n>:VERTical:SCALe ---- 167
:MATH<n>:HORizontal:OFFSet ---- 167
:MATH<n>:HORizontal:SCALe ---- 168
:MATH<n>:SOURce<ch> ---- 168
:MATH<n>:OPERation ---- 168
:MATH<n>:FFT:WINDow ---- 169
:MATH<n>:FFT:TYPe ---- 170
:MATH<n>:FFT:COOR:UNIT ---- 170
:MATH<n>:FFT:PSD:UNIT ---- 171
:MATH<n>:FFT:POINt ---- 171
:MATH<n>:FFT:FREQuency:CENTer ---- 172
:MATH<n>:FFT:FREQuency:SPAN ---- 172
:MATH<n>:FFT:MARK:RDMode ---- 172
:MATH<n>:FFT:MARK:AUTO ---- 173
:MATH<n>:FFT:MARK:STMode ---- 173
:MATH<n>:FFT:MARK:RESult ---- 174
:MATH<n>:FFT:MARK:COUNt ---- 174
:MATH<n>:FFT:MARK:THReshold ---- 175
:MATH<n>:FFT:MARK:EXCursion ---- 175
:MATH<n>:FFT:MARK:FREQTOCENTer ---- 175
:MATH<n>:FFT:DATA ---- 176
:MATH<n>:EXPRession ---- 176
:MATH<n>:FILTer:TYPe ---- 177
:MATH<n>:FILTer:FREQuency<n>:STOP ---- 177
:MATH<n>:FILTer:CUSTom:ENABle ---- 178
CYMometer 指令系统 ---- 179
:CYMometer:ACTive ---- 179
:CYMometer:HISTgram ---- 179
:CYMometer:TRENd ---- 180
:CYMometer:FREQuency ---- 180
DVM 指令系统 ---- 181
:DVM:ACTive ---- 181
:DVM:SOURce ---- 181
:DVM:HISTgram ---- 182
:DVM:TRENd ---- 182
:DVM:MODe ---- 182

:DVM:AUTO ---- 183
:DVM:CURRent ---- 183
DISPlay 指令系统 ---- 184
:DISPlay:PERSist ---- 184
:DISPlay:MARK ---- 184
:DISPlay:MKVert ---- 185
:DISPlay:MKHori ---- 185
:DISPlay:GRID:STYLe ---- 185
:DISPlay:GRID:BRIGtness ---- 186
:DISPlay:WAVe:STYLe ---- 186
:DISPlay:WAVe:BRIGtness ---- 187
:DISPlay:SCReen:BRIGtness ---- 187
:DISPlay:SCReen:CONTrast ---- 188
FILe 指令系统 ---- 189
:FILe:WAVe:FORMat ---- 189
:FILe:WAVe:TXTEncoding ---- 189
:FILe:WAVe:SOURce ---- 190
:FILe:WAVe:GATe ---- 190
:FILe:WAVe:NAMe:FORMat ---- 190
:FILe:WAVe:NAMe ---- 191
:FILe:WAVe:PATH ---- 191
:FILe:WAVe:SAVe ---- 192
:FILe:PICTure:REGion ---- 192
:FILe:PICTure:COLor ---- 193
:FILe:PICTure:FORMat ---- 193
:FILe:PICTure:NAMe ---- 193
:FILe:PICTure:NAMe:FORMat ---- 194
:FILe:PICTure:PATH ---- 194
:FILe:PICTure:SAVe ---- 195
:FILe:SETTing:NAMe ---- 195
:FILe:SETTing:NAMe:FORMat ---- 195
:FILe:SETTing:PATH ---- 196
:FILe:SETTing:SAVe ---- 196
:FILe:SETTing:READFULLPATH ---- 197
PRINt 指令系统 ---- 198
:PRINt:REGion ---- 198
:PRINt:COLor ---- 198
:PRINt:ORIent ---- 199

LA 指令系统 ---- 200
:LA:ITEM<n>:DISPlay ---- 200
:LA:ITEM<n>:OFFSet ---- 200
:LA:ITEM<n>:LABel ---- 201
:LA:ITEM<n>:THReshold ---- 201
:LA:ITEM<n>:LEVel ---- 202
:LA:HEIGht ---- 202
:LA:AUTO:LOCate ---- 203
:LA:ALL:DISPlay ---- 203
SEGement 指令系统 ---- 204
:SEGement:ACTive ---- 204
:SEGement:RSTacq ---- 204
:SEGement:MODe ---- 204
:SEGement:FRAMe:COUNt ---- 205
:SEGement:SEQuent:TYPe ---- 205
:SEGement:FRAMe:STARt ---- 206
:SEGement:FRAMe:END ---- 206
:SEGement:FRAMe:CURRent ---- 207
:SEGement:FRAMe:REF ---- 207
:SEGement:REFactive ---- 207
:SEGement:CALLback ---- 208
XY 指令系统 ---- 209
:XY<n>:ACTive ---- 209
:XY<n>:SOURce:X ---- 209
:XY<n>:SOURce:Y ---- 210
:XY<n>:CURSor:ACTive ---- 210
:XY<n>:TYPe ---- 210
:XY<n>:DATA:TYPe ---- 211
:XY<n>:CURSor:CAX ---- 211
:XY<n>:CURSor:CBX ---- 212
:XY<n>:CURSor:CAY ---- 212
:XY<n>:CURSor:CBY ---- 213
:XY<n>:CURSor:AXValue ---- 213
:XY<n>:CURSor:BXValue ---- 213
:XY<n>:CURSor:AYValue ---- 214
:XY<n>:CURSor:BYValue ---- 214
:XY<n>:WAVe:AXValue ---- 214
:XY<n>:WAVe:BXValue ---- 215

:XY<n>:WAVe:AYValue ---- 215
:XY<n>:WAVe:BYValue ---- 215
:XY<n>:RADius:A ---- 215
:XY<n>:RADius:B ---- 216
:XY<n>:ANGLe:A ---- 216
:XY<n>:ANGLe:B ---- 216
ZOOM 指令系统 ---- 218
:ZOOM:ACTive ---- 218
:ZOOM:CENTer:X ---- 218
:ZOOM:CENTer:Y ---- 219
:ZOOM:SCALe:X ---- 219
:ZOOM:SCALe:Y ---- 219
JITTer 指令系统 ---- 221
:JITTer:ACTive ---- 221
:JITTer:SOURce ---- 221
:JITTer:SIGNal:TYPe ---- 222
:JITTer:DTALength ---- 222
:JITTer:THReshold ---- 222
:JITTer:HYSTeresis ---- 223
:JITTer:BITRate ---- 223
:JITTer:CLOCk:TYPe ---- 224
:JITTer:PLL:TYPe ---- 224
:JITTer:PLL:DIVisor:CUToff ---- 225
:JITTer:PLL:FREQ:CUToff ---- 225
:JITTer:PLL:FREQ:NATural ---- 225
:JITTer:PLL:FACTor ---- 226
:JITTer:EXTernal:SOURce ---- 226
:JITTer:EXTernal:THReshold ---- 227
:JITTer:EXTernal:HYSTeresis ---- 227
:JITTer:PARams:JITTer ---- 227
:JITTer:HISTogram:BINNum ---- 228
:JITTer:GRAPh ---- 228
:JITTer:DATA:JITTer ---- 229
:JITTer:EYEDiagram:ACTive ---- 229
:JITTer:EYEDiagram:SOURce ---- 230
:JITTer:EYEDiagram:FAST ---- 230
:JITTer:EYEDiagram:SLOPe ---- 231
:JITTer:EYEDiagram:POLarity ---- 231

:JITTer:EYEDiagram:THReshold ---- 232
:JITTer:EYEDiagram:HYSTeresis ---- 232
:JITTer:EYEDiagram:BITRate ---- 232
:JITTer:EYEDiagram:CLOCk:TYPe ---- 233
:JITTer:EYEDiagram:PLL:TYPe ---- 233
:JITTer:EYEDiagram:PLL:DIVisor ---- 234
:JITTer:EYEDiagram:PLL:CUTFrequency ---- 234
:JITTer:EYEDiagram:PLL:NATFrequency ---- 235
:JITTer:EYEDiagram:PLL:FACTor ---- 235
:JITTer:EYEDiagram:EXTernal:SOURce ---- 235
:JITTer:EYEDiagram:EXTernal:THReshold ---- 236
:JITTer:EYEDiagram:EXTernal:HYSTeresis ---- 236
:JITTer:EYEDiagram:PARams:ACTive ---- 237
:JITTer:EYEDiagram:GATe ---- 237
:JITTer:EYEMask:ACTive ---- 238
:JITTer:EYEMask:MODe ---- 238
:JITTer:EYEMask:STDMask ---- 239
:JITTer:EYEMask:STANdard:INDex ---- 239
:JITTer:EYEMask:COUNt ---- 239
:JITTer:EYEMask:DATA ---- 240
:JITTer:DATA:EYE ---- 240
PASS/FAIL 指令系统 ---- 242
:PF:ACTive ---- 242
:PF:VISible ---- 242
:PF:MODe ---- 243
:PF:SOURce ---- 243
:PF:LIMit:SOURce ---- 243
:PF:LIMit:VTOLerance ---- 244
:PF:LIMit:HTOLerance ---- 244
:PF:LIMit:CREate ---- 245
:PF:STANdard:TYPe ---- 245
:PF:OFFSet:VERTical ---- 245
:PF:OFFSet:HORizontal ---- 246
:PF:OFFSet:RSTVertical ---- 246
:PF:OFFSet:RSTHorizontal ---- 246
:PF:STANdard:LOCK ---- 247
:PF:STANdard:INDex ---- 247
:PF:STOP:TYPe ---- 247

:PF:WFMS:COUNt ----248
:PF:TIMe:DURation ----248
:PF:VIOLations:COUNt ---- 249
:PF:OPTion:SAVe ----249
:PF:OPTion:BEEP ----250
:PF:OPTion:PULSe ---- 250
:PF:OPTion:SCReen ---- 251
:PF:WAVe:FORMat ---- 251
:PF:SCReen:FORMat ---- 251
:PF:FILe:PATH ---- 252
:PF:FILe:NAMe ----252
:PF:FILe:NAMe:FORMat ---- 253
:PF:RUN ---- 253
:PF:DATA ----254
SEARch 指令系统 ----255
:SEARch<n>:ACTive ---- 255
:SEARch<n>:SOURce ----255
:SEARch<n>:TYPe ---- 256
:SEARch<n>:VISible ---- 256
:SEARch<n>:EVENt ---- 256
:SEARch<n>:EDGe:SLOPe ---- 257
:SEARch<n>:EDGe:LEVel ----257
:SEARch<n>:PULSe:POLarity ----258
:SEARch<n>:PULSe:CONDition ---- 258
:SEARch<n>:PULSe:TIMe:UPPer ---- 259
:SEARch<n>:PULSe:TIMe:LOWer ---- 259
:SEARch<n>:PULSe:LEVel ---- 260
:SEARch<n>:DATA ----260
POWer 指令系统 ----262
:PWAS:ADD ----262
:PWAS:DELete ----263
:POWer:MODe ----263
:POWer:SOURce:VOLTage ---- 263
:POWer:SOURce:CURRent ----264
:POWer:DATA ---- 264
:QUALity:FREQuency:REFerence ---- 265
:HARMonic:COUNt ----265
:HARMonic:FREQuency:REFerence ----266

:HARMonic:FREQuency:CONStant ----266
:HARMonic:SOURce ----267
:HARMonic:TYPe ---- 267
:SAFeoperationarea:STOPonfail ---- 268
:SAFeoperationarea:AXIS:XMAX ---- 268
:SAFeoperationarea:AXIS:XMIN ---- 269
:SAFeoperationarea:AXIS:YMAX ---- 269
:SAFeoperationarea:AXIS:YMIN ---- 269
:LOOPanalysis:SOURce:AWG ----270
:LOOPanalysis:IMPedance ---- 270
:LOOPanalysis:FREQuency:MODe ----271
:LOOPanalysis:FREQuency:STARt ---- 271
:LOOPanalysis:FREQuency:STOP ---- 272
:LOOPanalysis:SCANnum ---- 272
:LOOPanalysis:AMPMode ---- 272
:LOOPanalysis:AMPValue ---- 273
:LOOPanalysis:TRIGStatus ---- 273
:MODulation:SOURce ----274
:MODulation:HISTogram:TYPe ---- 274
:MODulation:TRENd:TYPe ----275
:MODulation:HISTogram:ADD ---- 276
:MODulation:TRENd:ADD ---- 276
:INRUsh:PEAKcurrent ---- 276
:INRUsh:RUN ---- 277
:EFFIciency:INPut:VOLTage ---- 277
:EFFIciency:INPut:CURRent ---- 277
:EFFIciency:OUTput:VOLTage ----278
:EFFIciency:OUTput:CURRent ---- 278
:EFFIciency:INPut:DIAGram ---- 278
:EFFIciency:OUTput:DIAGram ---- 279
:RDS:WAVEform ----279
:TIMEonoff:INPut:VOLTage ---- 279
:TIMEonoff:OUTput:VOLTage ---- 280
:TIMEonoff:TESTtype ----280
:TIMEonoff:Typeconvert ---- 281
:TIMEonoff:ACQTime ---- 281
:PSRR:AWGSource ---- 281
:PSRR:IMPedance ---- 282

:PSRR:FREQuency:MODe ---- 282
:PSRR:TRIGStatus ---- 283
:PSRR:FREQuency:STARt ---- 283
:PSRR::FREQuency:STOP ---- 284
:PSRR:SCANnumber ---- 284
:PSRR:AMPMode ---- 284
:PSRR:AMPLitude ---- 285
:PSRR:BODepicture ---- 285
:PSRR:RUN ---- 286
:SLEWrate:DIAgram:DVDT ---- 286
:SLEWrate:DIAgram:DIDT ---- 286
REF 指令系统 ---- 287
:REFerence<n>:ACTive ---- 287
:REFerence<n>:LABel:ENABle ---- 287
:REFerence<n>:LABel ---- 288
:REFerence<n>:VERTical:OFFSet ---- 288
:REFerence<n>:VERTical:SCALe ---- 288
:REFerence<n>:HORizontal:OFFSet ---- 289
:REFerence<n>:HORizontal:SCALe ---- 289
:REFerence<n>:FINe ---- 290
BUS 指令系统 ---- 291
:BUS<n>:ACTive ---- 291
:BUS<n>:TYPe ---- 291
:BUS<n>:FORMat ---- 292
:BUS<n>:POSition ---- 292
:BUS<n>:LABel:ENABle ---- 292
:BUS<n>:LABel ---- 293
:BUS<n>:DATA ---- 293
:BUS<n>:RS232 ---- 294
:BUS<n>:I2C ---- 298
:BUS<n>:SPI ---- 301
:BUS<n>:CAN ---- 307
:BUS<n>:CAN_FD ---- 310
:BUS<n>:LIN ---- 314
:BUS<n>:FLEXRAY ---- 318
:BUS<n>:AUDIOBUS ---- 321
:BUS<n>:MIL_STD_1553 ---- 328
:BUS<n>:A429 ---- 331

:BUS<n>:USB ---- 335
:BUS<n>:SENT ---- 336
:BUS<n>:SPMi ---- 341
:BUS<n>:ETHernet ---- 343
:BUS<n>:NRZ ---- 345
:BUS<n>:Manchester ---- 348
:BUS<n>:8B10B ---- 354
:BUS<n>:PSI5 ---- 356
:BUS<n>:SMBus ---- 359
AUX 指令系统 ---- 362
:AUX:INPut:TYPe ---- 362
:AUX:INPut:POLarity ---- 362
:AUX:OUTPut:TYPe ---- 363
:AUX:OUTPut:POLarity ---- 363
NETWork 指令系统 ---- 365
:NETWork:LAN:COUNt ---- 365
:NETWork:LAN:SELect ---- 365
:NETWork:LAN:RESet ---- 365
:NETWork:LAN:APPLy ---- 365
:NETWork:LAN:IPADdress ---- 366
:NETWork:LAN:SMASK ---- 366
:NETWork:LAN:GATEway ---- 367
:NETWork:LAN:DHCP ---- 367
:NETWork:LAN:MAC ---- 367

引言

本编程手册包括 MSO8000HDP 系列示波器有关的 SCPI 指令和相关说明。

版权信息

优利德科技（中国）股份有限公司版权所有。

如果原购买者自购买该产品之日起三年内，将该产品出售或转让给第三方，则保修期应为自原购买者从优利德或授权的优利德分销商购买该产品之日起三年内。探头及其他附件和保险丝等不受此保证的保护。如果在适用的保修期内证明产品有缺陷，优利德可自行决定是修复有缺陷的产品且不收部件和人工费用，或用同等产品（由优利德决定）更换有缺陷的产品。优利德用作保修用途的部件、模块和更换产品可能是全新的，或者经修理具有相当于新产品的性能。所有更换的部件、模块和产品将成为优利德的财产。

以下提到的“客户”是指据声明本保证所规定权利的个人或实体。为获得本保证承诺的服务，“客户”必须在适用的保修期内向优利德通报缺陷，并为服务的履行做适当安排。客户应负责将有缺陷的产品装箱并运送到优利德指定的维修中心，同时预付运费并提供原购买者的购买证明副本。如果产品要运送到优利德维修中心所在国范围内的地点，优利德应支付向客户送返产品的费用。如果产品送返到任何其他地点，客户应负责支付所有的运费、关税、税金及任何其他费用。

本保证不适用于由于意外、机器部件的正常磨损、在产品规定的范围之外使用或使用不当或者维护保养不当或不足而造成的任何缺陷、故障或损坏。优利德根据本保证的规定无义务提供以下服务:

- 修理由非优利德服务代表人员对产品进行安装、修理或维护所导致的损坏;
- 修理由于使用不当或与不兼容的设备连接造成的损坏;
- 修理由于使用不符合本说明书要求的电源而造成的任何损坏或故障;
- 维修已改动或者与其他产品集成的产品（如果这种改动或集成会增加产品维修的时间或难度）;

本保证由优利德针对本产品而订立，用于替代任何其他的明示或暗示的保证，优利德及其经销商拒绝对用于特殊目的的适销性或适用性做任何暗示的保证。对于违反本保证的情况，优利德负责修理或更换有缺陷产品是提供给客户的唯一和全部补救措施。无论优利德及其经销商是否被预先告知可能发生任何间接、特殊、偶然或必然的损坏，优利德及其经销商对这些损坏均概不负责。

商标信息

UNI-T 是优利德科技（中国）股份有限公司[UNI-TREND TECHNOLOGY(CHINA)CO.， LTD]的注册商标。

文档版本

MSO8000HDP-V1.0

声明

- 优利德产品受中国或其他国家专利权的保护，包括已取得或正在申请的专利。
- 优利德保留更改产品规格和价格的权利。
- 优利德保留所有权利。许可软件产品由优利德及其子公司或提供商所有，受国家版权法及国际条约规定的保护。本文中的信息将取代所有以前出版的资料中的信息。
- 技术数据如有变更，恕不另行通告。

软件版本

V1.00.0000

软件升级可能更改或增加产品功能，请关注优利德官方网站:**www.uni-trend.com.cn** 获取最新版本手册和固件信息。

SCPI 指令简介

SCPI（Standard Commands for Programmable Instruments，即可编程仪器标准命令集）是一种建立在现有标准 IEEE 488.1 和 IEEE 488.2 基础上，并遵循了 IEEE754 标准中浮点运算规则、ISO646 信息交换 7 位编码符号（相当于 ASCII 编程）等多种标准的标准化仪器编程语言。本节简介 SCPI 命令的格式、符号、参数和缩写规则。

指令格式

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。命令行通常以冒号":"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。命令关键字和第一个参数之间以空格分开。命令字符串必须以一个 <换行> (<NL>) 字符结尾。命令行后面添加问号"?"通常表示对此功能进行查询。

符号说明

下面三种符号不是 SCPI 命令中的内容，不随命令发送，但是通常用于辅助说明命令中的参数。

- **大括号 { }**

  大括号中通常包含多个可选参数，发送命令时必须选择其中一个参数。如: :ACQuire:TYPe <type> 命令,type 可选{NORMal | PEAK | HIGHres | AVERages | ENVelope}。

- **竖线 |**

  竖线用于分隔多个参数选项，发送命令时必须选择其中一个参数。

  如: :ACQuire:TYPe <type> { NORMal | PEAK | HIGHres | AVERages | ENVelope}命令。

- **三角括号 < >**

  三角括号中的参数必须用一个有效值来替换。例如:以:ACQuire:TYPe NORMal 的形式发送:ACQuire:TYPe <type>命令。

参数说明

本手册介绍的命令中所含的参数可以分为以下5种类型:布尔型、整数型、实型、离散型、ASCII 字符串。

■ **布尔型**

参数取值为“ON”(1) 或“OFF”(0)。例如: :AWG1:ENABle <enable> 命令中的<enable>为布尔型{ON | OFF}

■ **整数型**

除非另有说明，参数在有效值范围内可以取任意整数值。注意:此时，请不要设置参数为小数格式，否则将出现异常。例如: :DISPlay:SCReen:BRIGtness <value>命令中的参数<value>可取 5 到 100 范围内的任一整数。

■ **实型**

除非另有说明，参数在有效值范围内可以取任意值。

例如:对于 CH1，CHANnel1:OFFSet <offset>命令中的参数<offset>的取值为实型。

■ **离散型**

参数只能取指定的几个数值或字符。例如: :ACQuire:TYPe <type> {NORMal | PEAK | HIGHres | AVERages | ENVelope} 命令只能取参数：NORMal、PEAK、HIGHres、AVERages、ENVelope。

■ **ASCII 字符串**

字符串参数实际上可包含所有 ASCII 字符集。字符串必须以配对的引号开始和结尾；可以用单引号或双引号。引号分隔符也可以作为字符串的一部分，只需键入两次并且不在中间添加任何字符，例如设置 IP:SYST:COMM:LAN:IPAD "192.168.1.10"。

简写规则

所有命令对大小写都能识别，可以全部采用大写或小写。如果要缩写，必须输完命令格式中的所有大写字母。

数据返回

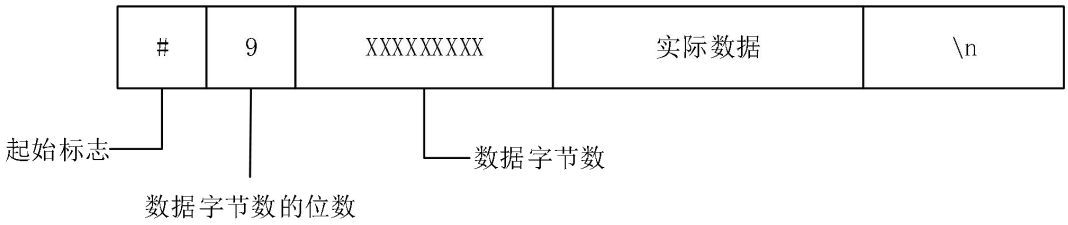
数据返回分为单个数据和批量数据返回。

单个数据返回相对应的参数类型，其中实型返回用科学计数法表示，返回频率和时间值时 E 前部分小数点后面保留十三位数据，E 部分保留三位数据，返回其它数值时 E 前部分小数点后面保留五位数据，E 部分保留三位数据；

批量数据返回格式为“数据块报头+数据块”，其中数据块报头具有如下的格式：#后面的第一位数字（9）表示数据块报头中剩下的数字位数；数据块报头中剩下的数字表示本次传输的数据字节数（不足 9 位在前面补 0），例如：发送 1000 字节的数据块报头为#9000001000：

数据块格式：

DATA 是数据流,其他为 ASCII 字符，如下图所示：<#9XXXXXXXXX + DATA + \n>



注意:当返回无效数据是 ASCII 类型数据时，用*表示；如果是实型数据类型，使用实型数据类型能表示的最大值数据返回。

SCPI 指令详解

IEEE488.2 通用命令

* IDN

功能描述：用于查询制造商名称、示波器型号、产品序列号和软件版本号。

指令：

IDN ?

返回：

制造商名称，示波器型号，产品序列号，由点号分隔的软件版本号。

示例：

**IDN?* /*查询返回 UNI-T, MSO8804HDP, AHD8888888888, V1.00.0000*/

* RST

功能描述：用于恢复出厂设置并清空所有的错误信息及发送接收队列缓冲。

指令：

*RST

SYSTem 指令系统

用于对示波器进行最基本的操作，主要包括运行控制、全键盘锁定、错误队列和系统设置数据的操作。

:RUN

功能描述：用于开始示波器波形采样工作，如需停止工作，需要执行:STOP 命令。

指令：

:RUN

:STOP

功能描述：用于停止示波器波形采样工作，如需恢复工作，需要执行:RUN 命令。

指令：

:STOP

:AUTO

功能描述：自动设置示波器将根据输入信号自动调整垂直档位、水平时基以及触发方式，使波形显示达到最佳状态注意，应用自动设置要求被测信号的频率不小于 20 Hz，占空比大于 1%，且幅度至少为 10 mVpp。

说明：当打开某些高级功能时，该命令不可用

指令：

:AUTO

:SYSTem:SINGle

功能描述：设置示波器触发模式为单次触发

指令：

:SYSTem:SINGle

:SYSTem:RTC

功能描述：设置或查询系统设置时间。

指令：

:SYSTem:RTC <value>

:SYSTem:RTC ?

参数：

<value>= 字符串型

返回：

查询返回系统时间。

示例：

:SYSTem:RTC "2026,4,18,17,8,57" /*设置系统时间为 2026/4/18/17/8/57*/

:SYSTem:RTC ? /*查询返回 2026,4,18,17,8,57*/

:SYSTem:TFORce

功能描述：强制产生一个触发信号。强制触发适用于“普通”和“单次”触发模式

指令：

:SYSTem:TFORce

:SYSTem:CALibration

功能描述：示波器自校正。

指令：

:SYSTem:CALibration

:SYSTem:CLEar

功能描述：清除示波器测量结果等。

指令：

:SYSTem:CLEar

:SYSTem:LANGuage

功能描述：设置或查询软件语言类型。

指令：

:SYSTem:LANGuage <language>

:SYSTem:LANGuage ?

参数：

<language>= 离散型{SIMPlifiedchinese | ENGLish | GERMan | FRENch | SPANish | ITALian}

- SIMPlifiedchinese:简体中文
- ENGLish:英文
- GERMan:德语
- FRENch:法语
- SPANish:西班牙语
- ITALian:意大利语

返回：

查询返回软件语言类型。

示例：

:SYSTem:LANGuage ENGLish /*设置语言为英文*/

:SYSTem:LANGuage ? /*查询返回 ENGLish*/

:SYSTem:PRTSc

功能描述：截图。

指令：

:SYSTem:PRTSc?

返回：

查询返回表示一张图片的字节数组。

读取的数据格式为头+波形数据点+结束符。头为#NXXXXXX 的形式，#为规定的头标志符，N 表示后面含有 N 个字节，以 ASCII 字符的形式描述波形数据点的长度，结束符用于表示通讯的终止。例如，一次读取的数据为#9000001000XXXX 表示 9 个字节描述数据的长度，000001000 表示波形数据的长度，即 1000 字节。

:SYSTem:TOUChlock

功能描述：设置或查询触摸锁开关状态。

指令：

:SYSTem:TOUChlock <active>

:SYSTem:TOUChlock ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:SYSTem:TOUChlock ON /*设置触摸锁打开*/

:SYSTem:TOUChlock ? /*查询返回 1*/

:SYSTem:AUTO:TIMebase

功能描述：用于自动设置时，水平时基档位属性是否保持当前状态。

指令：

:SYSTem:AUTO:TIMebase <status>

:SYSTem:AUTO:TIMebase ?

参数：

<status>= 离散型{AUTO/KEEP}

- AUTO：自动。自动设置时，水平时基档位属性为自动状态
- KEEP：保持。自动设置时，水平时基档位属性保持当前状态

返回：

查询返回“AUTO”、“KEEP”。

示例：

:SYSTem:AUTO:TIMebase KEEP /*设置水平时基档位属性保持当前状态*/

:SYSTem:AUTO:TIMebase? /*查询返回 KEEP*/

:SYSTem:AUTO:VERTical

功能描述：用于自动设置时，模拟通道垂直档位属性是否保持当前状态。

指令：

:SYSTem:AUTO:VERTical <status>

:SYSTem:AUTO:VERTical ?

参数：

<status>= 离散型{AUTO/KEEP}

- AUTO：自动。自动设置时，模拟通道垂直档位属性为自动状态
- KEEP：保持。自动设置时，模拟通道垂直档位属性保持当前状态

返回：

查询返回“AUTO”、“KEEP”。

示例：

:SYSTem:AUTO:VERTical KEEP /*设置模拟通道垂直档位属性保持当前状态*/

:SYSTem:AUTO:VERTical? /*查询返回 KEEP*/

:SYSTem:AUTO:ACQuire

功能描述：用于自动设置时，采集模式是否保持当前状态。

指令：

:SYSTem:AUTO:ACQuire <status>

:SYSTem:AUTO:ACQuire ?

参数：

<status>= 离散型{AUTO/KEEP}

- AUTO：自动。自动设置时，采集模式属性为自动状态
- KEEP：保持。自动设置时，采集模式属性为保持当前状态

返回：

查询返回“AUTO”、“KEEP”。

示例：

:SYSTem:AUTO:ACQuire KEEP /*设置采集模式属性为保持当前状态*/

:SYSTem:AUTO:ACQuire ? /*查询返回 KEEP*/

:SYSTem:AUTO:TRIGge

功能描述：用于自动设置时，触发模式、触发源是否保持当前状态。

指令：

;SYSTem:AUTO:TRIGger <status>

:SYSTem:AUTO:TRIGger ?

参数：

<status>= 离散型{AUTO/KEEP}

- AUTO：自动。自动设置时，触发模式、触发源属性为自动状态
- KEEP：保持。自动设置时，采集模式属性为保持当前状态

返回：

查询返回“AUTO”、“KEEP”。

示例：

:SYSTem:AUTO:TRIGger KEEP /*设置触发模式、触发源属性为保持当前状态*/

:SYSTem:AUTO:TRIGger? /*查询返回 KEEP*/

:SYSTem:AUTO:CHANel:COUPling

功能描述：用于自动设置时，通道耦合是否保持当前状态。

指令：

:SYSTem:AUTO:CHANel:COUPling <status>

:SYSTem:AUTO:CHANel:COUPling ?

参数：

<status>= 离散型{AUTO/KEEP}

- AUTO：自动。自动设置时，通道耦合属性为自动状态
- KEEP：保持。自动设置时，通道耦合属性为保持当前状态

返回：

查询返回“AUTO”、“KEEP”。

示例：

:SYSTem:AUTO:CHANel:COUPling KEEP /*设置通道耦合属性为保持当前状态*/

:SYSTem:AUTO:CHANel:COUPling? /*查询返回 KEEP*/

:SYSTem:AUTO:CHANel:ACTive

功能描述：用于自动设置时，通道打开或关闭是否保持当前状态。

指令：

:SYSTem:AUTO:CHANel:ACTive <status>

:SYSTem:AUTO:CHANel:ACTive ?

参数：

<status>= 离散型{AUTO/KEEP}

- AUTO：自动。自动设置时，通道打开或关闭属性为自动状态
- KEEP：保持。自动设置时，通道打开或关闭属性为保持当前状态

返回：

查询返回“AUTO”、“KEEP”。

示例：

:SYSTem:AUTO:CHANel:ACTive KEEP /*设置通道开关属性为保持当前状态*/

:SYSTem:AUTO:CHANel:ACTive? /*查询返回 KEEP*/

:SYSTem:AUTO:DISPlay:MODe

功能描述：用于自动设置时，通道是否堆叠显示。

指令：

:SYSTem:AUTO:DISPlay:MODe <status>

:SYSTem:AUTO:DISPlay:MODe ?

参数：

<status>= 离散型{AUTO/KEEP}

- AUTO：自动。自动设置时，通道自动堆叠显示
- KEEP：保持。自动设置时，通道保持平铺显示

返回：

查询返回“AUTO”、“KEEP”。

示例：

:SYSTem:AUTO:DISPlay:MODe KEEP /*设置通道为保持平铺显示*/

:SYSTem:AUTO:DISPlay:MODe ? /*查询返回 KEEP*/

ACQuire 指令系统

用于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:STATus

功能描述：查询示波器采集模式。

指令：

:ACQuire:STATus ?

返回：

查询返回“STOP”、“ARMED”、“READY”、“TRIGED”、“AUTO”、“ROLL”、“RESET”。

- STOP:停止
- ARMED:已配备
- READY:就绪
- TRIGED:已触发
- AUTO:自动
- ROLL:扫描
- RESET:重置

示例：

:ACQuire:STATus ? /*查询返回 READY*/

:ACQuire:TYPe

功能描述：设置或查询采集模式。

指令：

:ACQuire:TYPe <type>

:ACQuire:TYPe ?

参数：

<type>= 离散型{NORMal | PEAK | HIGHres | AVERage}

- NORMal:普通
- PEAK:峰值检测
- HIGHres:高分辨率
- AVERage:平均

返回：

查询返回“NORMal”、“PEAK”、“HIGHres”、“AVERage”。

示例：

:ACQuire:TYPe NORMal /*设置采样模式为普通*/

:ACQuire:TYPe ? /*查询返回 NORMal*/

:ACQuire:AVERage

功能描述：设置或查询平均采样模式下的平均次数。

指令：

:ACQuire:AVERage <count>

:ACQuire:AVERage ?

参数：

<count>= 整数型{2~65536}

返回：

查询返回一个表示平均次数的整数。

示例：

:ACQuire:AVERage 20 /*设置平均次数为 20 次*/

:ACQuire:AVERage ? /*查询返回 20*/

:ACQuire:CLKSrc

功能描述：设置或查询 10MHz 时钟信号来源。

指令：

:ACQuire:CLKSrc <clksrc>

:ACQuire:CLKSrc ?

参数：

<clksrc>= 离散型{INNer | OUTTer}

- INNer:内部
- OUTTer:外部

返回：

查询返回“INNer”、“OUTTer”。

示例：

:ACQuire:CLKSrc OUTTer /*设置 10MHz 时钟信号来源为“外部*/

::ACQuire:CLKSrc ? /*查询返回 OUTTer*/

:ACQuire:EXTLocked

功能描述：查询外部输入 10MHz 时钟频率是否锁定。

指令：

:ACQuire:EXTLocked ?

返回：

查询返回“1”、“0”。1:锁定 0:未锁定

:ACQuire:FAST

功能描述：设置或查询快采的开关状态。

指令：

:ACQuire:FAST <active>

:ACQuire:FAST ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

- ON:开启
- OFF:关闭

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:ACQuire:FAST ON /*设置快采打开*/

:ACQuire:FAST ? /*查询返回 1*/

:ACQuire:STORage

功能描述：设置或查询存储深度。

指令：

:ACQuire:STORage <type>

:ACQuire:STORage ?

参数：

<type>= 离散型，默认单位 Pts。

- 选择 AUTO 时，示波器根据当前的采样率自动选择存储深度。
- 全通道模式下：{100k | 1M | 10M | 100M | 1G | 2G | 4G}
- **返回：**

查询返回实际存储深度。

示例：

:ACQuire:STORage 100k /*设置存储深度 100kPts*/

:ACQuire:STORage ? /*查询返回 1.00000E+005*/

AWG 指令系统

用于设置示波器任意波形发生器的波形参数

:AWG<n>:LOAD

功能描述：设置或查询指定通道的负载阻抗。

指令：

:AWG<n>:LOAD <load>

:AWG<n>:LOAD ?

参数：

<load>= 离散型{HIGH | LOW}

- HIGH:1MΩ
- LOW:50Ω

返回：

查询返回“HIGH”、“LOW”。

示例：

:AWG1:LOAD HIGH /*设置通道 1 的负载阻抗为高阻*/

:AWG1:LOAD ? /*查询返回通道 1 的负载阻抗为 HIGH*/

:AWG<n>:ENABle

功能描述：设置或查询打开或关闭指定 AWG 通道的输出。

指令：

:AWG<n>:ENABle <enable>

:AWG<n>:ENABle ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:AWG1:ENABle ON /*设置通道 1 输出打开*/

:AWG1:ENABle ? /*查询返回 1*/

:AWG<n>:MODE

功能描述：设置或查询指定 AWG 通道的输出模式。

指令：

:AWG<n>:MODE <mode>

:AWG<n>:MODE ?

参数：

<mode>= 离散型{CONTinuous | MODulation | SWEep}

- CONTinuous:连续波
- MODulation:调制波
- SWEep:扫频

返回：

查询返回“CONTinuous”、“MODulation”、“SWEep”。

示例：

:AWG1:MODE MODulation /*设置 AWG1 的输出模式为调制波*/

:AWG1:MODE ? /*查询返回 MODulation*/

连续波

:AWG<n>:BASe:WAVe

功能描述：设置或查询指定通道的基本波形类型。

指令：

:AWG<n>:BASe:WAVe <type>

:AWG<n>:BASe:WAVe ?

参数：

<type>= 离散型{SINusoid | SQUare | PULSe | RAMP | NOISe | DC | SINC| EXPRise | EXPFall | LORentz | HAVersine | GAUSsian | ECG | ARBitrary}

- SINusoid:正弦波
- SQUare:方波
- PULSe:脉冲波
- RAMP:斜波
- NOISe:噪声
- DC:直流

- SINC:Sinc
- EXPRise:指数上升
- EXPFall:指数下降
- LORentz:洛伦兹
- HAVersine:半正矢
- GAUSsian:高斯
- ECG:心电图
- ARBitrary:任意波

返回：

查询返回“SINusoid”、“SQUare”、“PULSe”、“RAMP”、“NOISe”、“DC”、“SINC”、“EXPRise”、“EXPFall”、“LORentz”、“HAVersine”、“GAUSsian”、“ECG”、“ARBitrary”。

示例：

:AWG1:BASe:WAVe SINusoid /*设置通道 1 的基本波形为正弦波*/

:AWG1:BASe:WAVe ? /*查询返回通道 1 的基本波形为 SINusoid*/

:AWG<n>:BASe:FREQuency

功能描述：设置或查询指定通道的基本波形频率。

指令：

:AWG<n>:BASe:FREQuency <freq>

:AWG<n>:BASe:FREQuency ?

参数：

<freq>=实型，单位:uHz,mHz,Hz,kHz,MHz，不输入单位则默认单位为 Hz

返回：

查询返回指定通道的基本波形频率。

示例：

:AWG1:BASe:FREQuency 20 /*设置通道 1 的波形频率为 20Hz*/

:AWG1:BASe:FREQuency ? /*查询返回通道 1 的波形频率为 2.0000000000000E+001*/

:AWG<n>:BASe:ARBitrary

功能描述：设置或查询指定通道任意波文件读取路径。

指令：

:AWG<n>:BASe:ARBitrary <path>

:AWG<n>:BASe:ARBitrary ?

参数：

<path>= 字符串类型 AWG 任意波打开路径

返回：

查询返回指定通道当前任意波文件的路径。

示例：

:AWG1:BASe:ARBitrary "C:\\WaveForm\\Uni-t000.bin" /*设置通道 1 任意波文件读取路径为 C:\\WaveForm\\Uni-t000.bin*/

:AWG1:BASe:ARBitrary ? /*查询返回 C:\\WaveForm\\Uni-t000.bin*/

:AWG<n>:BASe:PHASe

功能描述：设置或查询指定通道的输出相位。

指令：

:AWG<n>:BASe:PHASe <phase>

:AWG<n>:BASe:PHASe ?

参数：

<phase>= 实型，表示相位，单位°，范围 0°~359.99°

返回：

查询返回指定通道的基本波形的相位。

示例：

:AWG1:BASe:PHASe 20 /*设置通道 1 的波形相位为 20°*/

:AWG1:BASe:PHASe ? /*查询返回通道 1 的波形相位为 2.00000E+001*/

:AWG<n>:BASe:AMPLitude

功能描述：设置或查询指定通道的输出幅度。

指令：

:AWG<n>:BASe:AMPLitude <amp>

:AWG<n>:BASe:AMPLitude ?

参数：

<amp>= 实型,表示电压，单位 mVpp,Vpp，不输入单位则默认单位为 Vpp。高阻输出范围:20mVpp ~ 6Vpp；低阻输出范围:10mVpp ~ 3Vpp

返回：

查询返回指定通道的基本波形的幅度

示例：

:AWG1:BASe:AMPLitude 2 /*设置通道 1 的波形幅度为 2Vpp*/

:AWG1:BASe:AMPLitude ? /*查询返回通道 1 的波形幅度为 2.00000E+000*/

:AWG<n>:BASe:OFFSet

功能描述：设置或查询指定通道的输出偏移。

指令：

:AWG<n>:BASe:OFFSet <offset>

:AWG<n>:BASe:OFFSet ?

参数：

<offset>= 实型,表示电压，单位 mV,V，不输入单位则默认单位为 V。高阻输出范围:-3V ~ 3V；低阻输出范围:-1.5V ~ 1.5V。

返回：

查询返回指定通道的基本波形的输出偏移。

示例：

:AWG1:BASe:OFFSet 2 /*设置通道 1 的波形偏置为 2V*/

:AWG1:BASe:OFFSet ? /*查询返回通道 1 的波形偏置为 2.00000E+000*/

:AWG<n>:BASe:HIGH

功能描述：设置或查询指定通道信号输出高值。

指令：

:AWG<n>:BASe:HIGH <voltage>

:AWG<n>:BASe:HIGH ?

参数：

<voltage>= 实型,表示电压，单位 mV,V，不输入单位则默认单位为 V。高阻输出范围:-3V ~ 3V；低阻输出范围:-1.5V ~ 1.5V。

返回：

查询返回指定通道信号输出高值。

示例：

:AWG1:BASe:HIGH 1 /*设置通道 1 的信号输出高值为 1V*/

:AWG1:BASe:HIGH ? /*查询返回通道 1 的信号输出高值为 1.00000E+000*/

:AWG<n>:BASe:LOW

功能描述：设置或查询指定通道信号输出低值。

指令：

:AWG<n>:BASe:LOW <voltage>

:AWG<n>:BASe:LOW ?

参数：

<voltage>= 实型,表示电压，单位 V，单位 mV,V，不输入单位则默认单位为 V。高阻输出范围:-3V ~ 3V；低阻输出范围:-1.5V ~ 1.5V。

返回：

查询返回指定通道信号输出低值。

示例：

:AWG1:BASe:LOW 1 /*设置通道 1 的信号输出低值为 1V*/

:AWG1:BASe:LOW ? /*查询返回通道 1 的信号输出低值为 1.00000E+000*/

:AWG<n>:PULSe:RISe

功能描述：设置或查询指定通道脉冲波的上升沿时间。

指令：

:AWG<n>:PULSe:RISe <time>

:AWG<n>:PULSe:RISe ?

参数：

<time>= 实型,单位 ns,us，默认单位为 s。范围 5ns~399.999us。

返回：

查询返回指定通道脉冲波的上升沿时间。

示例：

:AWG1:PULSe:RISe 10ns /*设置通道 1 的脉冲波上升时间为 10ns*/

:AWG1:PULSe:RISe ? /*查询返回 1.0000000000000E-008*/

说明：该指令仅在脉冲波下使用

:AWG<n>:PULSe:FALL

功能描述：设置或查询指定通道脉冲波的下降沿时间。

指令：

:AWG<n>:PULSe:FALL <time>

:AWG<n>:PULSe:FALL ?

参数：

<time>= 实型,单位 ns,us，默认单位为 s。范围 5ns~399.999us。

返回：

查询返回指定通道脉冲波的下降沿时间。

示例：

:AWG1:PULSe:FALL 10ns /*设置通道 1 的脉冲波下降时间为 10ns*/

:AWG1:PULSe:FALL ? /*查询返回 1.0000000000000E-008*/

说明：该指令仅在脉冲波下使用

:AWG<n>:BASe:DUTY

功能描述：设置或查询指定通道信号输出占空比。

指令：

:AWG<n>:BASe:DUTY <duty>

:AWG<n>:BASe:DUTY ?

参数：

<Duty>= 实型,表示占空比，单位%，范围 0~100。

返回：

查询返回指定通道信号输出占空比。

示例：

:AWG1:BASe:DUTY 50 /*设置通道 1 的信号输出占空比为 50%*/

:AWG1:BASe:DUTY ? /*查询返回通道 1 的信号输出占空比 5.00000E+001*/

说明：该指令仅在方波和脉冲波下使用

:AWG<n>:BASe:SYMMetry

功能描述：设置或查询指定通道信号输出对称性。

指令：

:AWG<n>:BASe:SYMMetry <duty>

:AWG<n>:BASe:SYMMetry ?

参数：

<duty>= 实型,表示占空比，单位%，范围 0~100。

返回：

查询返回指定通道信号输出对称性

示例：

:AWG1:BASe:SYMMetry 50 /*设置通道 1 的信号输出对称性为 50%*/

:AWG1:BASe:SYMMetry ? /*查询返回通道 1 的信号输出对称性 5.00000E+001/

说明：该指令仅在斜波下使用

调制波

:AWG<n>:MODUlate:MODe

功能描述：设置或查询指定通道信号调制模式。

指令：

:AWG<n>:MODUlate:MODe <mode>

:AWG<n>:MODUlate:MODe ?

参数：

<mode>= 离散型{AM | PM | FM}

返回：

查询返回“AM”、“PM”、“FM”。

示例：

:AWG1:MODUlate:MODe AM /*设置通道 1 信号为调幅模式*/

:AWG1:MODUlate:MODe ? /*查询返回 AM*/

:AWG<n>:MODUlate:WAVe

功能描述：设置或查询指定通道信号调制波形。

指令：

:AWG<n>:MODUlate:WAVe <type>

:AWG<n>:MODUlate:WAVe ?

参数：

<type>= 离散型{SINusoid | SQUare | RAMP | NOISe | ARBitrary}

- SINusoid:正弦波
- SQUare:方波
- RAMP:斜波

- NOISe:噪声
- ARBitrary:任意波

返回：

查询返回“SINusoid”、“SQUare”、“RAMP”、“NOISe”、“ARBitrary”。

示例：

:AWG1:MODUlate:WAVe SINusoid /*设置通道 1 信号调制波类型为正弦波*/

:AWG1:MODUlate:WAVe ? /*查询返回 SINusoid*/

:AWG<n>:MODUlate:ARBitrary

功能描述：设置或查询指定 AWG 通道调制波形为任意波时文件读取路径

指令：

:AWG<n>:MODUlate:ARBitrary <path>

:AWG<n>:MODUlate:ARBitrary ?

参数：

<path>= 字符串类型 AWG 任意波打开路径

返回：

查询返回指定 AWG 通道调制波形为任意波时文件路径。

示例：

:AWG1:MODUlate:ARBitrary "C:\\WaveForm\\Uni-t000.bin" /*设置通道 1 调制波形为任意波时文件读取路径为 C:\\WaveForm\\Uni-t000.bin*/

:AWG1:MODUlate:ARBitrary ? /*查询返回 C:\\WaveForm\\Uni-t000.bin*/

:AWG<n>:MODUlate:FREQuency

功能描述：设置或查询指定通道信号调制频率。

指令：

:AWG<n>:MODUlate:FREQuency <freq>

:AWG<n>:MODUlate:FREQuency ?

参数：

<Freq>= 实型，表示频率，单位 mHz,Hz,kHz，不输入单位则默认单位为 Hz。

返回：

查询返回指定通道信号的调制频率。

示例：

:AWG1:MODUlate:FREQuency 2000 /*设置通道 1 信号调制频率为 2KHz*/

:AWG1:MODUlate:FREQuency ? /*查询返回 2.0000000000000E+00*/

:AWG<n>:FM:FREQuency:DEV

功能描述：设置或查询指定通道波形的频率偏移。

指令：

:AWG<n>:FM:FREQuency DEV <freq>

:AWG<n>:FM:FREQuency ?

参数：

<Freq>= 实型，表示频率，单位 mHz,Hz,kHz，不输入单位则默认单位为 Hz。

返回：

查询返回指定通道波形的频率偏移。

示例：

:AWG1:FM:FREQuency:DEV 2 /*设置通道 1 频率偏移为 2Hz*/

:AWG1:FM:FREQuency:DEV ? /*查询返回通道 1 频率偏移为 2.0000000000000E+000*/

:AWG<n>:MODUlate:DEPTh

功能描述：设置或查询指定通道波形的频率偏移。

指令：

:AWG<n>:MODUlate:DEPTh <depth>

:AWG<n>:MODUlate:DEPTh ?

参数：

<depth>= 实型，表示调制深度，单位%。AM 调制深度为 1%~120%

返回：

查询返回指定通道信号的调制深度。

示例：

:AWG1:MODUlate:DEPTh 50 /*设置通道 1 信号调制深度为 50%*/

:AWG1:MODUlate:DEPTh ? /*查询返回 5.00000E+001*/

:AWG<n>:PM:PHASe:DEV

功能描述：设置或查询指定通道信号相位调制相偏。

指令：

:AWG<n>:PM:PHASe:DEV <value>

:AWG<n>:PM:PHASe:DEV ?

参数：

<value>= 实型，范围 0°~359.99°，单位 °（度）。

返回：

查询返回指定通道信号相位调制相偏。

示例：

:AWG1:PM:PHASe:DEV 50 /*设置通道 1 信号调制相偏为 50°*/

:AWG1:PM:PHASe:DEV ? /*查询返回 5.00000E+001*/

扫频

:AWG<n>:SWEep:TYPe

功能描述：设置或查询指定通道信号扫频类型。

指令：

:AWG<n>:SWEep:TYPe <type>

:AWG<n>:SWEep:TYPe ?

参数：

<type>= 离散型{LINear | LOGarithmic}

- LINear :线性
- LOGarithmic:对数

返回：

查询返回“LINear ”、“LOGarithmic”。

示例：

:AWG1:SWEep:TYPe LOGarithmic /*设置通道 1 扫频类型为对数*/

:AWG1:SWEep:TYPe ? /*查询返回 LOGarithmic*/

:AWG<n>:SWEep:DURatio

功能描述：设置或查询指定通道扫频时间。

指令：

:AWG<n>:SWEep:DURatio <time>

:AWG<n>:SWEep:DURatio ?

参数：

<time>= 实型，单位 ms，s，默认单位 s。

返回：

查询返回指定通道扫频时间。

示例：

:AWG1:SWEep:DURatio 1 /*设置通道 1 扫频时间 1s*/

:AWG1:SWEep:DURatio ? /*查询返回 1.0000000000000E+000*/

:AWG<n>:SWEep:FREQuency:STARt

功能描述：设置或查询指定通道扫频起始频率。

指令：

:AWG<n>:SWEep:FREQuency:STARt <freq>

:AWG<n>:SWEep:FREQuency:STARt ?

参数：

<freq>= 实型，单位 mHz、Hz、kHz、MHz，默认单位 Hz。

返回：

查询返回指定通道扫频起始频率。

示例：

:AWG1:SWEep:FREQuency:STARt 1 /*设置通道 1 扫频起始频率 1Hz*/

:AWG1:SWEep:FREQuency:STARt ? /*查询返回 1.0000000000000E+000*/

:AWG<n>:SWEep:FREQuency:END

功能描述：设置或查询指定通道扫频结束频率。

指令：

:AWG<n>:SWEep:FREQuency:END <freq>

:AWG<n>:SWEep:FREQuency:END ?

参数：

<freq>= 实型，单位 mHz、Hz、kHz、MHz，默认单位 Hz。

返回：

查询返回指定通道扫频结束频率。

示例：

:AWG1:SWEep:FREQuency:END 1 /*设置通道 1 扫频结束频率 1Hz*/

:AWG1:SWEep:FREQuency:END ? /*查询返回 1.0000000000000E+000*/

:AWG<n>:SWEep:TRIGger:SOURce

功能描述：设置或查询指定通道扫频触发源。

指令：

:AWG<n>:SWEep:TRIGger:SOURce <source>

:AWG<n>:SWEep:TRIGger:SOURce ?

参数：

<source>= 离散型{INSide | MANual | OUTSide}

- INSide:内部
- MANual:手动
- OUTSide:外部

返回：

查询返回指定通道扫频触发源。

示例：

:AWG1:SWEep:TRIGger:SOURce OUTSide /*设置通道 1 扫频触发源为外部*/

:AWG1:SWEep:TRIGger:SOURce ? /*查询返回 OUTSide*/

:AWG<n>:SWEep:TRIGger:ENABle

功能描述：设置或查询指定通道扫频触发输出的开启或关闭。

指令：

:AWG<n>:SWEep:TRIGger:ENABle <enable>

:AWG<n>:SWEep:TRIGger:ENABle ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:AWG1:SWEep:TRIGger:ENABle OFF /*设置通道 1 扫频触发输出关闭*/

:AWG1:SWEep:TRIGger:ENABle ? /*查询返回 0*/

:AWG<n>:SWEep:TRIGger:MANualtrig

功能描述：设置指定通道扫频触发源为“手动”时触发一次。

指令：

:AWG<n>:SWEep:TRIGger:MANualtrig

CHANnel 指令系统

用于设置或查询模拟通道的参数，通道可独立调节。

:CHANnel<n>:SELect

功能描述：设置或查询通道的选中状态。

指令：

:CHANnel<n>:SELect

:CHANnel<n>:SELect ?

参数：

<n>= 模拟通道数，范围 1~4

返回：

查询返回模拟通道的选中状态，返回“1”、“0”分别代表选中、未选中。

示例：

:CHANnel1:SELect /*设置通道 1 为选中状态*/

:CHANnel1:SELect? /*查询返回 1*/

:CHANnel<n>:BWLimit

功能描述：设置或查询通道的带宽限制。

指令：

:CHANnel<n>:BWLimit <bwlimit>

:CHANnel<n>:BWLimit ?

参数：

<bwlimit>= 离散型{8GHz | 6GHz | 4GHz | 2GHz | 1GHz | 500MHz | 20MHz}

- 8GHz:带宽限制为 8GHz
- 6GHz:带宽限制为 6GHz
- 4GHz:带宽限制为 4GHz
- 2GHz:带宽限制为 2GHz
- 1GHz:带宽限制为 1GHz
- 500MHz:带宽限制为 500MHz
- 20MHz:带宽限制为 20MHz

返回：

查询返回“8GHz”、“6GHz”、“4GHz”、“2GHz”、“1GHz”、“500MHz”、“20MHz”。

示例：

:CHANnel1:BWLimit 2GHz /*设置通道 1 带宽限制为 2GHz*/

:CHANnel1:BWLimit ? /*查询返回 2GHz*/

:CHANnel<n>:COUPling

功能描述：设置或查询通道输入的耦合方式为“DC50Ω”、“GND”。

指令：

:CHANnel<n>:COUPling <coupling>

:CHANnel<n>:COUPling ?

参数：

<coupling>= 离散型{DC50 | GND}

- DC50:耦合方式为直流低阻
- GND:耦合方式为接地

返回：

查询返回““DC50”、“GND”

示例：

:CHANnel1:COUPling DC50 /*设置通道 1 耦合方式为 DC50*/

:CHANnel1:COUPling ? /*查询返回 DC50*/

:CHANnel<n>:DISPlay

功能描述：关闭或打开通道的显示。

指令：

:CHANnel<n>:DISPlay <active>

:CHANnel<n>:DISPlay ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:CHANnel1:DISPlay ON /*打开通道 1 显示*/

:CHANnel1:DISPlay ? /*查询返回 1*/

:CHANnel<n>:INVert

功能描述：关闭或打开通道的反相显示。

指令：

:CHANnel<n>:INVert <invert>

:CHANnel<n>:INVert ?

参数：

<invert>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:CHANnel1:INVert ON /*打开通道 1 反相显示*/

:CHANnel1:INVert ? /*查询返回 1*/

:CHANnel<n>:OFFSet

功能描述：设置或查询指定通道波形显示的垂直偏移。

指令：

:CHANnel<n>:OFFSet <offset>

:CHANnel<n>:OFFSet ?

参数：

<offset>= 实型，-5.5~5.5，居中为 0，单位 div

返回：

查询返回指定通道的垂直偏移。

示例：

:CHANnel1:OFFSet 1 /*设置通道 1 垂直偏移 1div*/

:CHANnel1:OFFSet ? /*查询返回 1.00000E+000*/

:CHANnel<n>:PROBe:MODel

功能描述：查询探头型号。

指令：

:CHANnel<n>:PROBe:MODel ?

:CHANnel<n>:PROBe:SERailnumber

功能描述：查询探头序列号。

指令：

:CHANnel<n>:PROBe:SERailnumber ?

:CHANnel<n>:PROBe:GAIN:CURRent

功能描述：设置或查询探头衰减比。

指令：

:CHANnel<n>:PROBe:GAIN:CURRent <probe>

:CHANnel<n>:PROBe:GAIN:CURRent ?

参数：

<probe>= 离散型{1 | 5 | 10 | 100 | CUSTom}

- 1:探头衰减比为 1
- 5:探头衰减比为 5
- 10:探头衰减比为 10
- 100:探头衰减比为 100
- CUStom:自定义

返回：

查询返回探头衰减比。

示例：

:CHANnel1:PROBe:GAIN:CURRent 10 /*设置通道 1 探头衰减比为 10*/

:CHANnel1:PROBe:GAIN:CURRent ? /*查询返回 10*/

:CHANnel<n>:PROBe:GAIN:CUSTom

功能描述：设置或查询自定义探头倍率。

指令：

:CHANnel<n>:PROBe:GAIN:CUSTom <count>

:CHANnel<n>:PROBe:GAIN:CUSTom ?

参数：

<count>= 实型，设置范围 0.001X-2000X

返回：

查询返回自定义的探头倍率，以科学计数法表示。

示例：

:CHANnel1:PROBe:GAIN:CUSTom 10 /*设置通道 1 探头倍率为 10*/

:CHANnel1:PROBe:GAIN:CUSTom ? /*查询返回 1.00000E+001*/

:CHANnel<n>:PROBe:BUTTon

功能描述：设置或查询探头按钮类型。

指令：

:CHANnel<n>:PROBe:BUTTon <button>

:CHANnel<n>:PROBe:BUTTon ?

参数：

<button>= 离散型{HEADlight | RUNorstop | CLEar | FORCetrig | NOOPs}

- HEADlight:前灯
- RUNorstop:运行/停止
- CLEar:清除
- FORCetrig 强制触发:
- NOOPs:无操作

返回：

查询返回“HEADlight”、“RUNorstop”、“CLEar”、“FORCetrig”、“NOOPs”。

示例：

:CHANnel1:PROBe:BUTTon NOOPs /*设置通道 1 探头按钮类型为无操作*/

:CHANnel1:PROBe:BUTTon ? /*查询返回 NOOPs*/

:CHANnel<n>:UNIT:ACTive

功能描述：设置或查询备用单位开启状态。

指令：

:CHANnel<n>:UNIT:ACTive <active>

:CHANnel<n>:UNIT:ACTive ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:CHANnel1:UNIT:ACTive ON /*设置通道 1 备用单位打开*/

:CHANnel1:UNIT:ACTive ? /*查询返回 1*/

:CHANnel<n>:UNIT

功能描述：设置或查询备用单位。

指令：

:CHANnel<n>:UNIT <unit>

:CHANnel<n>:UNIT ?

参数：

<unit>= 字符串类型

返回：

查询返回备用单位字符串。

示例：

:CHANnel1:UNIT "A" /*设置通道 1 探头单位为 A*/

:CHANnel1:UNIT ? /*查询返回 A*/

:CHANnel<n>:UNIT:RATio

功能描述：设置或查询备用单位比率。

指令：

:CHANnel<n>:UNIT:RATio <value>

:CHANnel<n>:UNIT:RATio ?

参数：

<value>= 实型，设置范围 0.001~1000

返回：

查询返回备用单位比率值。

示例：

:CHANnel1:UNIT:RATio 3 /*设置通道 1 探头单位比率为 3A/V*/

:CHANnel1:UNIT:RATio ? /*查询返回 3*/

:CHANnel<n>:SCALe

功能描述：设置或查询指定通道波形显示的垂直档位。

指令：

:CHANnel<n>:SCALe <scale>

:CHANnel<n>:SCALe ?

参数：

<scale>= 实型，单位:自定义，不输入单位则默认单位为 V

- 当输入阻抗为 50Ω时，可调范围在 1mV ~1V

返回：

查询返回以科学计数形式的垂直档位值，默认单位 V。

示例：

:CHANnel1:SCALe 1V /*设置通道 1 波形显示的垂直档位为 1V*/

:CHANnel1:SCALe ? /*查询返回 1.00000E+000*/

:CHANnel<n>:FINe

功能描述：设置或查询指定通道的垂直档位细调是否打开。

指令：

:CHANnel<n>:FINe <active>

:CHANnel<n>:FINe ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:CHANnel1:FINe ON /*设置通道 1 的垂直档位细调打开*/

:CHANnel1:FINe ? /*查询返回 1*/

:CHANnel<n>:BIAS

功能描述：设置或查询偏置电压。

指令：

:CHANnel<n>:BIAS <bias>

:CHANnel<n>:BIAS ?

参数：

<bias>= 实型，单位:与通道垂直刻度单位一致，默认单位为 V

返回：

查询返回以科学计数形式的偏置电压值。

示例：

:CHANnel1:BIAS 1V /*设置通道 1 波形的偏置电压为 1V*/

:CHANnel1:BIAS ? /*查询返回 1.00000E+000*/

:CHANnel<n>:DELay

功能描述：设置或查询通道延迟。

指令：

:CHANnel<n>:DELay <time>

:CHANnel<n>:DELay ?

参数：

<time>= 实型，默认单位为 s，可设置范围为-100ns~100ns。

返回：

查询返回以科学计数形式的通道延迟值。

示例：

:CHANnel1:DELay 1ns /*设置通道 1 波形的通道延迟为 1ns*/

:CHANnel1:DELay ? /*查询返回 1.0000000000000E-009*/

说明：该指令仅在时基刻度小于 50ns/div 时可用

:CHANnel<n>:COPY

功能描述：设置该通道各项垂直设置复制到另一模拟通道源。

指令：

:CHANnel<n>:COPY <source>

参数：

<time>= 离散型{C1 | C2 | C3 | C4}

示例：

:CHANnel1:COPY C2 /*设置通道 1 的各项垂直设置复制到通道 2*/

说明：复制到的通道不能和被复制通道一样

:CHANnel<n>:LABel:ENABle

功能描述：设置或查询打开或关闭指定通道标签显示。

指令：

:CHANnel<n>:LABel:ENABle <enable>

:CHANnel<n>:LABel:ENABle ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:CHANnel1:LABel:ENABle ON /*设置通道 1 的垂直档位细调打开*/

:CHANnel1:LABel:ENABle ? /*查询返回 1*/

:CHANnel<n>:LABel

功能描述：设置或查询通道标签。

指令：

:CHANnel<n>:LABel <label>

:CHANnel<n>:LABel ?

参数：

<label>=通道名称字符串

返回：

查询返回通道名称字符串。

示例：

:CHANnel1:LABel ″abcd″ /*设置通道 1 标签为 abcd*/

:CHANnel1:LABel ? /*查询返回 abcd*/

:CHANnel<n>:ERES:ENABle

功能描述：设置或查询增强分辨率打开状态。

指令：

:CHANnel<n>:ERES:ENABle <type>

:CHANnel<n>:ERES:ENABle ?

参数：

<type>= 布尔型{ON | OFF}或{1 | 0}

- ON:开启
- OFF:关闭

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:CHANnel1:ERES:ENABle ON /*设置通道 1 增强分辨率打开*/

:CHANnel1:ERES:ENABle ? /*查询返回 1*/

:CHANnel<n>:ERES:BIT

功能描述：设置或查询增强分辨率增强位数

指令：

:CHANnel<n>:ERES:BIT <value>

:CHANnel<n>:ERES:BIT ?

参数：

<value>= 实型，范围 0.5~4，步进为 0.5

返回：

查询返回增强分辨率增强位数，以科学计数法表示。

示例：

:CHANnel1:ERES:BIT 1 /*设置通道 1 增强分辨率增强位数 1*/

:CHANnel1:ERES:BIT ? /*查询返回 1.00000E+000*/

CURSor 指令系统

用于设置或查询示波器的光标系统。

:CURSor:ACTive

功能描述：设置或查询光标是否打开。

指令：

:CURSor:ACTive <active>

:CURSor:ACTive ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:CURSor:ACTive ON /*设置打开光标*/

:CURSor:ACTive ? /*查询返回 1*/

:CURSor:TRACewave

功能描述：设置或查询光标模式的波形追踪。

指令：

:CURSor:TRACewave <active>

:CURSor:TRACewave ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:CURSor:TRACewave 1 /*设置光标波形追踪打开*/

:CURSor:TRACewave ? /*查询返回 1*/

:CURSor:TYPe

功能描述：设置或查询光标模式的光标类型。

指令：

:CURSor:TYPe <type>

:CURSor:TYPe ?

参数：

<type>= 离散型{AMPLitude | TIMe | SCReen | CLOSe}

- AMPLitude:幅度（水平）
- TIMe:时间（垂直）
- SCReen:屏幕（水平&垂直）
- CLOSe:关闭

返回：

查询返回“AMPLitude”、“TIMe”、“SCReen”、“CLOSe”。

示例：

:CURSor:TYPe AMPLitude /*设置光标类型为幅度*/

:CURSor:TYPe ? /*查询返回 AMPLitude*/

:CURSor:SOURce

功能描述：设置或查询光标模式的源。

指令：

:CURSor:SOURce <source>

:CURSor:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4}

返回：

查询返回“C1”、“C2”、“C3”、“C4”、“M1”、“M2”、“M3”、“M4”、“M5”、“M6”、“M7”、“M8”、“R1”、“R2”、“R3”、“R4”。

示例：

:CURSor:SOURce C1 /*设置光标测量源为通道 1*/

:CURSor:SOURce ? /*查询返回 C1*/

:CURSor:MOVe:SYNC

功能描述：设置或查询光标的同步移动开启或关闭

指令：

:CURSor:MOVe:SYNC <enable>

:CURSor:MOVe:SYNC ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:CURSor:MOVe:SYNC ON /*设置光标的同步移动开启*/

:CURSor:MOVe:SYNC ? /*查询返回 1*/

:CURSor:UNIT:VCURsor

功能描述：设置或查询垂直光标单位

指令：

:CURSor:UNIT:VCURsor <unit>

:CURSor:UNIT:VCURsor ?

参数：

<unit>= 离散型{SECond | DEGRee | PERCent}

- SECond:秒
- DEGRee:度
- PERCent:百分比

返回：

查询返回“SECond”、“DEGRee”、“PERCent”。

示例：

:CURSor:UNIT:VCURsor SECond /*设置垂直光标单位为秒*/

:CURSor:UNIT:VCURsor ? /*查询返回 SECond*/

:CURSor:UNIT:HCURsor

功能描述：设置或查询水平光标单位。

指令：

:CURSor:UNIT:HCURsor <unit>

:CURSor:UNIT:HCURsor ?

参数：

<unit>= 离散型{BASe | PERCent}

- BASe:基
- PERCent:百分比

说明：BASe 取决于光标所在通道的单位。

返回：

查询返回“BASe”、“PERCent”。

示例：

:CURSor:UNIT:HCURsor BASe /*设置水平光标单位为基*/

:CURSor:UNIT:HCURsor ? /*查询返回 BASe*/

:CURSor:POSition:MODe

功能描述：设置或查询光标位置显示类型。

指令：

:CURSor:POSition:MODe <mode>

:CURSor:POSition:MODe ?

参数：

<mode>= 离散型{ABSolute | RELative}

- ABSolute:绝对值
- RELative:相对值

返回：

查询返回“ABSolute”、“RELative”。

示例：

:CURSor:POSition:MODe ABSolute /*设置打开光标位置显示类型为绝对值显示*/

:CURSor:POSition:MODe ? /*查询返回 ABSolute*/

:CURSor:CAX

功能描述：设置或查询垂直光标 A 位置。

指令：

:CURSor:CAX <value>

:CURSor:CAX ?

参数：

<value>= 实型，光标位置类型为相对值显示时单位为 div，光标位置显示类型为绝对值时与实际水平坐标单位一致，以 s 为默认单位。

返回：

查询返回垂直光标线 A 的位置。

示例：

:CURSor:CAX 3 /*设置垂直光标 A 的位置 3div*/

:CURSor:CAX ? /*查询返回 3.00000E+000*/

:CURSor:CBX

功能描述：设置或查询垂直光标 B 位置。

指令：

:CURSor:CBX <value>

:CURSor:CBX ?

参数：

<value>= 实型，光标位置类型为相对值显示时单位为 div，光标位置显示类型为绝对值时与实际水平坐标单位一致，以 s 为默认单位。

返回：

查询返回垂直光标线 B 的位置。

示例：

:CURSor:CBX 3 /*设置垂直光标 B 的位置 3div*/

:CURSor:CBX ? /*查询返回 3.00000E+000*/

:CURSor:CAY

功能描述：设置或查询水平光标 A 位置。

指令：

:CURSor:CAY <value>

:CURSor:CAY ?

参数：

<value>= 实型，光标位置类型为相对值显示时单位为 div，光标位置显示类型为绝对值时与实际垂直坐标单位一致，以 V 为默认单位。

返回：

查询返回水平光标线 A 的位置。

示例：

:CURSor:CAY 3 /*设置水平光标 A 的位置 3div*/

:CURSor:CAY ? /*查询返回 3.00000E+000*/

:CURSor:CBY

功能描述：设置或查询水平光标 B 位置。

指令：

:CURSor:CBY <value>

:CURSor:CBY ?

参数：

<value>= 实型，光标位置类型为相对值显示时单位为 div，光标位置显示类型为绝对值时与实际垂直坐标单位一致，以 V 为默认单位。

返回：

查询返回水平光标线 B 的位置。

示例：

:CURSor:CBY 3 /*设置水平光标 B 的位置 3div*/

:CURSor:CBY ? /*查询返回 3.00000E+000*/

:CURSor:AXValue

功能描述：查询垂直光标 A 的测量值。

指令：

:CURSor:AXValue ?

返回：

按”时间值,电压值”的格式返回，采用科学计数法。

:CURSor:BXValue

功能描述：查询垂直光标 B 的测量值。

指令：

:CURSor:BXValue ?

返回：

按”时间值,电压值”的格式返回，采用科学计数法。

:CURSor:AYValue

功能描述：查询水平光标 A 的测量值。

指令：

:CURSor:AYValue ?

返回：

查询返回水平光标 A 的测量值，采用科学计数法表示。

:CURSor:BYValue

功能描述：查询水平光标 B 的测量值。

指令：

:CURSor:BYValue ?

返回：

查询返回水平光标 B 的测量值，采用科学计数法表示。

:CURSor:DATA:MODe

功能描述：设置或查询光标数据悬浮是否打开。

指令：

:CURSor:DATA:MODe <active>

:CURSor:DATA:MODe ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

- ON:悬浮
- OFF:固定

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:CURSor:DATA:MODe ON /*设置打开光标数据悬浮*/

:CURSor:DATA:MODe ? /*查询返回 1*/

TIMebase 指令系统

用于设置或查询示波器的水平时基系统

:TIMebase:SCALe

功能描述：设置或查询主时基档位。

指令：

:TIMebase:SCALe <scale>

:TIMebase:SCALe ?

参数：

<scale>= 实型，单位:ps,ns,us,ms,s,ks，不输入单位则默认单位为 s。设置范围:50ps~1ks。

返回：

查询以科学计数形式返回水平档位值,默认单位为 s。

示例：

:TIMebase:SCALe 1s /*设置水平时基档为 1s*/

:TIMebase:SCALe ? /*查询返回 1.0000000000000E+000*/

:TIMebase:DELay

功能描述：设置或查询延迟扫描。

指令：

:TIMebase:DELay <delay>

:TIMebase:DELay ?

参数：

<delay>= 实型，单位 fs,ps,ns,us,ms,s,ks，不输入单位则默认单位为 s

返回：

查询以科学计数形式返回延迟扫描。

示例：

:TIMebase:DELay 1s /*设置延迟扫描为 1s*/

:TIMebase:DELay ? /*查询返回 1.0000000000000E+000*/

TRIGger 指令系统

用于设置示波器的触发模式、所有触发类型参数和自动保存参数。

:TRIGger:TYPe

功能描述：设置或查询触发类型。

指令：

:TRIGger:TYPe <type>

:TRIGger:TYPe ?

参数：

<type>= 离散型{EDGe | PWIDth | VIDeo | PATTern | TIMeout | SETuphold | RUNt | TRANsition | NEDGe | SERial}

- EDGe:边沿
- PWIDth:脉宽
- VIDeo:视频
- PATTern:码型
- TIMeout:超时
- SETuphold:建立保持
- RUNt:欠幅
- TRANsition:斜率
- NEDGe:N 边沿
- SERial:总线

返回：

查询返回“EDGe”、“PWIDth”、“VIDeo”、“PATTern”、“TIMeout”、“SETuphold”、“RUNt”、“TRANsition”、“NEDGe”、“SERial”。

示例：

:TRIGger:TYPe EDGe /*设置触发类型为边沿触发*/

:TRIGger:TYPe ? /*查询返回 EDGe*/

:TRIGger:HOLDoff

功能描述：设置或查询触发释抑时间。

指令：

:TRIGger:HOLDoff <holdoff>

:TRIGger:HOLDoff ?

参数：

<holdoff>= 实型，单位:ns,us,ms,s，默认单位为 s

返回：

查询返回释抑时间，以科学计数法表示。

示例：

:TRIGger:HOLDoff 1s /*输出选择触发释抑 1s*/

:TRIGger:HOLDoff ? /*查询返回 1.0000000000000E+000*/

:TRIGger:MODe

功能描述：设置或查询触发模式。

指令：

:TRIGger:MODe <mode>

:TRIGger:MODe ?

参数：

<mode>= 离散型{AUTO | NORMal | SINGle}

- AUTO:自动
- NORMal:正常
- SINGle:单次

返回：

查询返回“AUTO”、“NORMal”、“SINGle”。

示例：

:TRIGger:MODe NORMal /*设置触发模式为正常模式*/

:TRIGger:MODe ? /*查询返回 NORMal*/

:TRIGger:EDGe

:TRIGger:EDGe:COUPling

功能描述：设置或查询触发耦合。

指令：

:TRIGger:EDGe:COUPling <coupling>

:TRIGger:EDGe:COUPling ?

参数：

<coupling>= 离散型{DC | AC | LFReject | HFReject | NREJect}

- DC:直流
- AC:交流
- LFReject:低频抑制
- HFReject:高频抑制
- NREJect:噪声抑制

返回：

查询返回“DC”、“AC”、“LFReject”、“HFReject”、“NREJect”。

示例：

:TRIGger:EDGe:COUPling DC /*设置触发耦合为 DC 耦合*/

:TRIGger:EDGe:COUPling ? /*查询返回 DC*/

:TRIGger:EDGe:SOURce

功能描述：设置或查询边沿触发源。

指令：

:TRIGger:EDGe:SOURce <source>

:TRIGger:EDGe:SOURce?

参数：

<source>= 离散型{C1 | C2 | C3 | C4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | Ext | Ext5 | AC | AUXin}

返回：

查询返回“C1”、“C2”、“C3”、“C4”、“D0”、“D1”、“D2”、“D3”、“D4”、“D5”、“D6”、“D7”、“D8”、“D9”、“D10”、“D11”、“D12”、“D13”、“D14”、“D15”、“Ext”、“Ext5”、“AC”、“AUXin”。

示例：

:TRIGger:EDGe:SOURce C1 /*设置边沿触发源为通道 1*/

:TRIGger:EDGe:SOURce ? /*查询返回 C1*/

:TRIGger:EDGe:SLOPe

功能描述：设置或查询边沿触发边沿类型。

指令：

:TRIGger:EDGe:SLOPe <type>

:TRIGger:EDGe:SLOPe ?

参数：

<type>= 离散型{RISe | FALL | ANY}

- RISe:上升沿
- FALL:下降沿
- ANY:任意沿

返回：

查询返回“RISe”、“FALL”、“ANY”。

示例：

:TRIGger:EDGe:SLOPe RISe /*设置边沿触发边沿类型为上升沿*/

:TRIGger:EDGe:SLOPe ? /*查询返回 RISe*/

:TRIGger:EDGe:LEVel

功能描述：设置或查询边沿触发的触发电平。

指令：

:TRIGger:EDGe:LEVel <level>

:TRIGger:EDGe:LEVel ?

参数：

<level>= 实型，单位:与通道垂直刻度单位一致，默认单位为 V。

返回：

查询以科学计数形式返回触发电平。

示例：

:TRIGger:EDGe:LEVel 1V /*设置边沿触发电平为 1V*/

:TRIGger:EDGe:LEVel ? /*查询返回 1.00000E+000*/

:TRIGger:EDGe:EXTern:IMPedance

功能描述：设置或查询外触发的输入阻抗。

指令：

:TRIGger:EDGe:EXTern:IMPedance <impedance >

:TRIGger:EDGe:EXTern:IMPedance ?

参数：

<impedance>= 离散型{HIGH | LOW}

- HIGH:1MΩ
- LOW:50Ω

返回：

查询返回外触发的输入阻抗。

示例：

:TRIGger:EDGe:EXTern:IMPedance HIGH /*设置外触发输入阻抗为 1MΩ*/

:TRIGger:EDGe:EXTern:IMPedance ? /*查询返回 HIGH*/

:TRIGger:EDGe:RESet

功能描述：设置边沿触发的触发电平复位。

指令：

:TRIGger:EDGe:RESet

:TRIGger:SHOLd

:TRIGger:SHOLd:DSOUrce

功能描述：设置或查询建立保持数据源。

指令：

:TRIGger:SHOLd:DSOUrce <source>

:TRIGger:SHOLd:DSOUrce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回“C1”、“C2”、“C3”、“C4”。

示例：

:TRIGger:SHOLd:DSOUrce C1 /*设置建立保持数据源为通道 1*/

:TRIGger:SHOLd:DSOUrce ? /*查询返回 C1*/

说明：数据源和时钟源不能相同！！！

:TRIGger:SHOLd:CSOUrce

功能描述：设置或查询建立保持时钟源。

指令：

:TRIGger:SHOLd:CSOUrce <source>

:TRIGger:SHOLd:CSOUrce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回“C1”、“C2”、“C3”、“C4”。

示例：

:TRIGger:SHOLd:CSOUrce C2 /*设置建立保持时钟源为通道 2*/

:TRIGger:SHOLd:CSOUrce ? /*查询返回 C2*/

说明：时钟源和数据源不能相同！！！

:TRIGger:SHOLd:TYPe

功能描述：设置或查询建立保持触发的时钟边沿。

指令：

:TRIGger:SHOLd:TYPe <type>

:TRIGger:SHOLd:TYPe ?

参数：

<type>= 离散型{RISe | FALL}

- RISe:上升
- FALL:下降

返回：

查询返回“RISe”、“FALL”。

示例：

:TRIGger:SHOLd:TYPe RISe /*设置建立保持触发的时钟边沿为上升沿*/

:TRIGger:SHOLd:TYPe ? /*查询返回 RISe*/

:TRIGger: SHOLd:CLEVel

功能描述：设置或查询建立保持触发的时钟电平。

指令：

:TRIGger:SHOLd:CLEVel <value>

:TRIGger:SHOLd:CLEVel?

参数：

<value>= 实型，单位:与通道垂直刻度单位一致，默认单位为 V。

返回：

查询返回建立保持触发的时钟电平。

示例：

:TRIGger:SHOLd:CLEVel 1V /*设置建立保持触发的时钟电平为 1V*/

:TRIGger:SHOLd:CLEVel ? /*查询返回 1.00000E+000*/

:TRIGger:SHOLd:DLEVel

功能描述：设置或查询建立保持触发的数据电平。

指令：

:TRIGger:SHOLd:DLEVel <value>

:TRIGger:SHOLd:DLEVel?

参数：

<value>= 实型，单位:与通道垂直刻度单位一致，默认单位为 V。

返回：

设置或查询建立保持触发的数据电平。

示例：

:TRIGger:SHOLd:DLEVel 1V /*设置建立保持触发的数据电平为 1V*/

:TRIGger:SHOLd:DLEVel ? /*查询返回 1.00000E+000*/

:TRIGger:SHOLd:PATTern

功能描述：设置或查询建立保持触发的码型条件。

指令：

:TRIGger:SHOLd:PATTern <type>

:TRIGger:SHOLd:PATTern ?

参数：

<type>= 离散型{HIGH | LOW}

- HIGH:高
- LOW:低

返回：

查询返回“HIGH”、“LOW”

示例：

:TRIGger:SHOLd:PATTern HIGH /*设置建立保持触发的码型条件为 HIGH*/

:TRIGger:SHOLd:PATTern ? /*查询返回 HIGH*/

:TRIGger:SHOLd:VIOLation

功能描述：设置或查询保持触发的触发条件。

指令：

:TRIGger:SHOLd:VIOLation <type>

:TRIGger:SHOLd:VIOLation ?

参数：

<type>= 离散型{SETup | HOLD | SETHold}

- SETup:建立
- HOLD:保持
- SETHold:建立保持

返回：

查询返回“SETup”、“HOLD”、“SETHold”。

示例：

:TRIGger:SHOLd:VIOLation SETup /*设置保持触发的触发条件为建立*/

:TRIGger:SHOLd:VIOLation ? /*查询返回 SETup*/

:TRIGger:SHOLd:SETTime

功能描述：设置或查询保持触发的建立时间。

指令：

:TRIGger:SHOLd:SETTime <value>

:TRIGger:SHOLd:SETTime ?

参数：

<value>= 实型，单位:ns,us,ms,s，不输入单位则默认单位为 s。

返回：

查询返回保持触发的建立时间。

示例：

:TRIGger:SHOLd:SETTime 1s /*设置保持触发的建立时间为 1s*/

:TRIGger:SHOLd:SETTime ? /*查询返回 1.0000000000000E+000*/

:TRIGger:SHOLd:HOLDtime

功能描述：设置或查询保持触发的保持时间。

指令：

:TRIGger:SHOLd:HOLDtime <value>

:TRIGger:SHOLd:HOLDtime ?

参数：

<value>= 实型，单位:ns,us,ms,s，不输入单位则默认单位为 s。

返回：

查询返回保持触发的保持时间。

示例：

:TRIGger:SHOLd:HOLDtime 1 /*设置保持触发的保持时间为 1s*/

:TRIGger:SHOLd:HOLDtime ? /*查询返回 1.0000000000000E+000*/

:TRIGger:PATTern

:TRIGger:PATTern:LOGic:ALLCondition

功能描述：设置码型触发逻辑资格的所有通道触发条件。

指令：

:TRIGger:PATTern:LOGic:ALLCondition <condition>

参数：

<condition>= 离散型{HIGH | LOW | ANY | RISe | FALL}

- HIGH:高
- LOW:低
- ANY:任意
- RISe :上升沿
- FALL:下降沿

示例：

:TRIGger:PATTern:LOGic:ALLCondition HIGH /*设置所有通道触发条件为 HIGH*/

:TRIGger:PATTern:LOGic:CONDition:C<n>

功能描述：设置或查询码型触发逻辑资格的单个模拟通道触发条件。

指令：

:TRIGger:PATTern:LOGic:CONDition:C<n> <type>

:TRIGger:PATTern:LOGic:CONDition:C<n> ?

参数：

<type>= 离散型{HIGH | LOW | ANY | RISe | FALL}

- HIGH:高
- LOW:低
- ANY:任意
- RISe :上升沿
- FALL:下降沿

返回：

查询返回“HIGH”、“LOW”、“ANY”、“RISE”、“FALL”。

示例：

:TRIGger:PATTern:LOGic:CONDition:C1 HIGH /*设置 C1 触发条件为 HIGH*/

:TRIGger:PATTern:LOGic:CONDition:C1 ? /*查询返回 HIGH*/

:TRIGger:PATTern:LOGic:GATe:C<n>

功能描述：设置或查询码型触发逻辑资格的单个模拟通道门限。

指令：

:TRIGger:PATTern:LOGic:GATe:C<n> <value>

:TRIGger:PATTern:LOGic:GATe:C<n> ?

参数：

<value>= 实型，单位:与通道垂直刻度单位一致，默认单位为 V。

返回：

查询返回码型触发逻辑资格的单个模拟通道门限。

示例：

:TRIGger:PATTern:LOGic:GATe:C1 1V /*设置 C1 触发门限为 1V*/

:TRIGger:PATTern:LOGic:GATe:C1 ? /*查询返回 1.00000E+000*/

:TRIGger:PWIDth

:TRIGger:PWIDth:CONDition

功能描述：设置或查询脉宽触发条件。

指令：

:TRIGger:PWIDth:CONDition <condition>

:TRIGger:PWIDth:CONDition ?

参数：

<condition>= 离散型{GREaterthan | LESSthan | RANGe}

- GREaterthan:大于
- LESSthan:小于
- RANGe:范围内

返回：

查询返回“GREaterthan”、“LESSthan”、“RANGe”

示例：

:TRIGger:PWIDth:CONDition GREaterthan /*设置脉宽触发条件为 GREaterthan*/

:TRIGger:PWIDth:CONDition ? /*查询返回 GREaterthan*/

:TRIGger:PWIDth:TIMe:LOWer

功能描述：设置或查询脉宽触发的脉宽下限。

指令：

:TRIGger:PWIDth:TIMe:LOWer <time>

:TRIGger:PWIDth:TIMe:LOWer ?

参数：

<time>= 实型，单位:ns,us,ms,s，不输入单位则默认单位为 s

返回：

查询返回以秒为单位，以科学计数法表示的脉宽下限值

示例：

:TRIGger:PWIDth:TIMe:LOWer 1 /*设置触发脉宽下限为 1s*/

:TRIGger:PWIDth:TIMe:LOWer ? /*查询返回 1.0000000000000E+000*/

:TRIGger:PWIDth:TIMe:UPPer

功能描述：设置或查询脉宽触发的脉宽上限。

指令：

:TRIGger:PWIDth:TIMe:UPPer <time>

:TRIGger:PWIDth:TIMe:UPPer ?

参数：

<time>= 实型，单位:ns,us,ms,s，不输入单位则默认单位为 s

返回：

查询返回以秒为单位，以科学计数法表示的脉宽上限值

示例：

:TRIGger:PWIDth:TIMe:UPPer 1 /*设置触发脉宽上限为 1s*/

:TRIGger:PWIDth:TIMe:UPPer ? /*查询返回 1.0000000000000E+000*/

:TRIGger:PWIDth:POLarity

功能描述：设置或查询脉宽触发的脉宽极性。

指令：

:TRIGger:PWIDth:POLarity <polarity>

:TRIGger:PWIDth:POLarity ?

参数：

<polarity>= 离散型{POSitive | NEGative}

- POSitive:正脉宽
- NEGative:负脉宽

返回：

查询返回“POSitive”、“NEGative”。

示例：

:TRIGger:PWIDth:POLarity POSitive /*设置脉宽触发极性为 POSitive*/

:TRIGger:PWIDth:POLarity ? /*查询返回查询返回 POSitive*/

:TRIGger:PWIDth:SOURce

功能描述：设置或查询脉宽触发的源。

指令：

:TRIGger:PWIDth:SOURce <source>

:TRIGger:PWIDth:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15}

返回：

查询返回“C1”、“C2”、“C3”、“C4”、“D0”、“D1”、“D2”、“D3”、“D4”、“D5”、“D6”、“D7”、“D8 ”、“D9”、“D10”、“D11”、“D12”、“D13”、“D14”、“D15 ”。

示例：

:TRIGger:PWIDth:SOURce C1 /*设置脉宽触发源为 C1*/

:TRIGger:PWIDth:SOURce ? /*查询返回 C1*/

:TRIGger:PWIDth:LEVel

功能描述：设置或查询脉宽触发的时钟电平。

指令：

:TRIGger:PWIDth:LEVel <value>

:TRIGger:PWIDth:LEVel ?

参数：

<value>= 实型，单位:与通道垂直刻度单位一致，默认单位为 V。

返回：

查询返回脉宽触发的时钟电平。

示例：

:TRIGger:PWIDth:LEVel 10mV /*设置脉宽触发的时钟电平为 10mV*/

:TRIGger:PWIDth:LEVel ? /*查询返回 1.00000E-002*/

:TRIGger:PWIDth:RESet

功能描述：设置脉宽触发的触发电平复位。

指令：

:TRIGger:PWIDth:RESet

:TRIGger:RUNT

:TRIGger:RUNT:SOURce

功能描述：设置或查询欠幅触发源。

指令：

:TRIGger:RUNT:SOURce <source>

:TRIGger:RUNT:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回“C1”、“C2”、“C3”、“C4”。

示例：

:TRIGger:RUNT:SOURce C1 /*设置欠幅触发源为 C1*/

:TRIGger:RUNT:SOURce ? /*查询返回 C1*/

:TRIGger:RUNT:CONDition

功能描述：设置或查询欠幅触发条件。

指令：

:TRIGger:RUNT:CONDition <type>

:TRIGger:RUNT:CONDition ?

参数：

<type>= 离散型{GREaterthan | LESSthan | RANGe | NRANge}

- GREaterthan :大于
- LESSthan:小于
- RANGe:范围内
- NRANge:范围外

返回：

查询返回“GREaterthan”、“LESSthan”、“RANGe”、“NRANge”。

示例：

:TRIGger:RUNT:CONDition GREaterthan /*设置欠幅触发条件为 GREaterthan*/

:TRIGger:RUNT:CONDition ? /*查询返回 GREaterthan*/

:TRIGger:RUNT:LEVel:UPPer

功能描述：设置或查询欠幅触发的高电平。

指令：

:TRIGger:RUNT:LEVel:UPPer <value>

:TRIGger:RUNT:LEVel:UPPer ?

参数：

<value>= 实型，单位:与通道垂直刻度单位一致，默认单位为 V。

返回：

查询返回欠幅触发的高电平。

示例：

:TRIGger:RUNT:LEVel:UPPer 1 /*设置欠幅触发高电平为 1V*/

:TRIGger:RUNT:LEVel:UPPer ? /*查询返回 1.00000E+000*/

:TRIGger:RUNT:LEVel:LOWer

功能描述：设置或查询欠幅触发的低电平。

指令：

:TRIGger:RUNT:LEVel:LOWer <value>

:TRIGger:RUNT:LEVel:LOWer ?

参数：

<value>= 实型，单位:与通道垂直刻度单位一致，默认单位为 V。

返回：

查询返回欠幅触发的低电平。

示例：

:TRIGger:RUNT:LEVel:LOWer 1 /*设置欠幅触发低电平为 1V*/

:TRIGger:RUNT:LEVel:LOWer ? /*查询返回 1.00000E+000*/

:TRIGger:RUNT:POLarity

功能描述：设置或查询欠幅触发的极性。

指令：

:TRIGger:RUNT:POLarity <polarity>

:TRIGger:RUNT:POLarity ?

参数：

<Polarity>= 离散型{POSitive | NEGative}

- POSitive:正极性
- NEGative:负极性

返回：

查询返回“POSitive”、“NEGative”。

示例：

:TRIGger:RUNT:POLarity POSitive /*设置欠幅触发极性为正极性*/

:TRIGger:RUNT:POLarity ? /*查询返回 POSitive*/

:TRIGger:RUNT:TIMe:UPPer

功能描述：设置或查询欠幅触发的欠幅时间上限。

指令：

:TRIGger:RUNT:TIMe:UPPer <time>

:TRIGger:RUNT:TIMe:UPPer ?

参数：

<time>= 实型，单位:ns,us,ms,s，不输入单位则默认单位为 s。

返回：

查询返回欠幅触发的欠幅时间上限。

示例：

:TRIGger:RUNT:TIMe:UPPer 1 /*设置欠幅触发的欠幅时间上限为 1s*/

:TRIGger:RUNT:TIMe:UPPer ? /*查询返回 1.0000000000000E+000*/

:TRIGger:RUNT:TIMe:LOWer

功能描述：设置或查询欠幅触发的欠幅时间下限。

指令：

:TRIGger:RUNT:TIMe:LOWer <time>

:TRIGger:RUNT:TIMe:LOWer ?

参数：

<time>= 实型，单位:ns,us,ms,s，不输入单位则默认单位为 s。

返回：

查询返回欠幅触发的欠幅时间下限。

示例：

:TRIGger:RUNT:TIMe:LOWer 1 /*设置欠幅触发的欠幅时间下限为 1s*/

:TRIGger:RUNT:TIMe:LOWer ? /*查询返回 1.0000000000000E+000*/

:TRIGger:SLOPe

:TRIGger:SLOPe:SOURce

功能描述：设置或查询斜率触发源。

指令：

:TRIGger:SLOPe:SOURce <source>

:TRIGger:SLOPe:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回“C1”、“C2”、“C3”、“C4”。

示例：

:TRIGger:SLOPe:SOURce C1 /*设置斜率触发源为通道 1*/

:TRIGger:SLOPe:SOURce ? /*查询返回 C1*/

:TRIGger:SLOPe:CONDition

功能描述：设置或查询斜率触发条件。

指令：

:TRIGger:SLOPe:CONDition <type>

:TRIGger:SLOPe:CONDition ?

参数：

<type>= 离散型{GREaterthan | LESSthan | RANGe}

- GREaterthan:大于
- LESSthan:小于
- RANGe:范围内

返回：

查询返回“GREaterthan”、“LESSthan”、“RANGe”。

示例：

:TRIGger:SLOPe:CONDition GREaterthan /*设置斜率触发条件为 GREaterthan*/

:TRIGger:SLOPe:CONDition ? /*查询返回 GREaterthan*/

:TRIGger:SLOPe:LEVel:UPPer

功能描述：设置或查询斜率触发的高电平。

指令：

:TRIGger:SLOPe:LEVel:UPPer <value >

:TRIGger:SLOPe:LEVel:UPPer ?

参数：

<value>= 实型，单位:与通道垂直刻度单位一致，默认单位为 V。

返回：

查询返回斜率触发的高电平。

示例：

:TRIGger:SLOPe:LEVel:UPPer 1 /*设置斜率触发高电平为 1V*/

:TRIGger:SLOPe:LEVel:UPPer ? /*查询返回 1.00000E+000*/

:TRIGger:SLOPe:LEVel:LOWer

功能描述：设置或查询斜率触发的低电平。

指令：

:TRIGger:SLOPe:LEVel:LOWer <value>

:TRIGger:SLOPe:LEVel:LOWer ?

参数：

<value>= 实型，单位:与通道垂直刻度单位一致，默认单位为 V。

返回：

查询返回斜率触发的低电平。

示例：

:TRIGger:SLOPe:LEVel:LOWer 1 /*设置斜率触发低电平为 1V*/

:TRIGger:SLOPe:LEVel:LOWer ? /*查询返回 1.00000E+000*/

:TRIGger:SLOPe:SLOPe

功能描述：设置或查询斜率触发的斜率类型。

指令：

:TRIGger:SLOPe:SLOPe <type>

:TRIGger:SLOPe:SLOPe ?

参数：

<type>= 离散型{RISe | FALL}

- RISe:上升
- FALL:下降

返回：

查询返回“RISe”、“FALL”。

示例：

:TRIGger:SLOPe:SLOPe RISe /*设置斜率触发类型为 RISe*/

:TRIGger:SLOPe:SLOPe ? /*查询返回 RISe*/

:TRIGger:SLOPe:TIMe:UPPer

功能描述：设置或查询斜率触发的斜率时间上限。

指令：

:TRIGger:SLOPe:TIMe:UPPer <time>

:TRIGger:SLOPe:TIMe:UPPer ?

参数：

<time>= 实型，单位:ns,us,ms,s，不输入单位则默认单位为 s。

返回：

查询返回斜率触发的斜率时间上限。

示例：

:TRIGger:SLOPe:TIMe:UPPer 1 /*设置斜率触发斜率时间上限为 1s*/

:TRIGger:SLOPe:TIMe:UPPer ? /*查询返回 1.0000000000000E+000*/

:TRIGger:SLOPe:TIMe:LOWer

功能描述：设置或查询斜率触发的斜率时间下限。

指令：

:TRIGger:SLOPe:TIMe:LOWer <time>

:TRIGger:SLOPe:TIMe:LOWer ?

参数：

<time>= 实型，单位:ns,us,ms,s，不输入单位则默认单位为 s。

返回：

查询返回斜率触发的斜率时间下限。

示例：

:TRIGger:SLOPe:TIMe:LOWer 1 /*设置斜率触发斜率时间下限为 1s*/

:TRIGger:SLOPe:TIMe:LOWer ? /*查询返回 1.0000000000000E+000*/

:TRIGger:TIMeout

:TRIGger:TIMeout:SOURce

功能描述：设置或查询超时触发源。

指令：

:TRIGger:TIMeout:SOURce <source>

:TRIGger:TIMeout:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15}

返回：

查询返回“C1”、“C2”、“C3”、“C4”、“D0”、“D1”、“D2”、“D3”、“D4”、“D5”、“D6”、“D7”、“D8 ”、“D9”、“D10”、“D11”、“D12”、“D13”、“D14”、“D15 ”。

示例：

:TRIGger:TIMeout:SOURce C1 /*设置超时触发源为通道 1*/

:TRIGger:TIMeout:SOURce ? /*查询返回 C1*/

:TRIGger:TIMeout:SLOPe

功能描述：设置或查询超时边沿类型。

指令：

:TRIGger:TIMeout:SLOPe <type>

:TRIGger:TIMeout:SLOPe ?

参数：

<type>= 离散型{POSitive | NEGative | ANY}

- POSitive:上升沿
- NEGative:下降沿
- ANY:任意沿

返回：

查询返回“POSitive”、“NEGative”、“ANY”。

示例：

:TRIGger:TIMeout:SLOPe POSitive /*设置超时触发类型为上升沿*/

:TRIGger:TIMeout:SLOPe ? /*查询返回 POSitive*/

TRIGger:TIMeout:LEVel

功能描述：设置或查询超时触发的触发电平。

指令：

:TRIGger:TIMeout:LEVel <level>

:TRIGger:TIMeout:LEVel ?

参数：

<level>= 实型，单位:与通道垂直刻度单位一致，默认单位为 V。

返回：

查询返回超时触发的触发电平。

示例：

:TRIGger:TIMeout:LEVel 1 /*设置超时触发电平为 1V*/

:TRIGger:TIMeout:LEVel ? /*查询返回 1.00000E+000*/

:TRIGger:TIMeout:TIMe

功能描述：设置或查询超时触发的持续时间。

指令：

:TRIGger:TIMeout:TIMe <time>

:TRIGger:TIMeout:TIMe ?

参数：

<time>= 实型，单位:ns,us,ms,s，不输入单位则默认单位为 s

返回：

查询返回超时触发的持续时间。

示例：

:TRIGger:TIMeout:TIMe 1 /*设置超时触发持续时间为 1s*/

:TRIGger:TIMeout:TIMe ? /*查询返回 1.0000000000000E+000*/

:TRIGger:TIMeout:RESet

功能描述：设置超时触发的触发电平复位。

指令：

:TRIGger:TIMeout:RESet

:TRIGger:VIDeo

:TRIGger:VIDeo:SOURce

功能描述：设置或查询视频触发源。

指令：

:TRIGger:VIDeo:SOURce <source>

:TRIGger:VIDeo:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回“C1”、“C2”、“C3”、“C4”。

示例：

:TRIGger:VIDeo:SOURce C1 /*设置视频触发源为通道 1*/

:TRIGger:VIDeo:SOURce ? /*查询返回 C1*/

:TRIGger:VIDeo:STANdard

功能描述：设置或查询视频标准类型。

指令：

:TRIGger:VIDeo:STANdard <standard>

:TRIGger:VIDeo:STANdard ?

参数：

<standard>= 离散型{PAL | NTSC | V525P60 | V625P60 | V720P24 | V720P25 | V720P30 | V720P50 | V720P60 | V1080I25 | V1080I30 | V1080P24 | C1080P25 | V1080P30 | V1080PSF24}

返回：

查询返回“PAL”、NTSC”、“V525P60”、“V625P60”、“V720P24”、“V720P25”、“V720P30”、“V720P50”、“V720P60”、“V1080I25”、“V1080I30”、“V1080P24”、“V1080P25”、“V1080P30”、“V1080PSF24”、。

示例：

:TRIGger:VIDeo:STANdard NTSC /*设置视频标准类型为 NTSC*/

:TRIGger:VIDeo:STANdard ? /*查询返回 NTSC*/

:TRIGger:VIDeo:SYNC

功能描述：设置或查询视频触发的同步方式。

指令：

:TRIGger:VIDeo:SYNC <sync>

:TRIGger:VIDeo:SYNC ?

参数：

<sync>= 离散型{ODD | EVEn | ALL | ALLLine | SPECified}

- ODD:奇数场
- EVEn:偶数场
- ALL:所有场
- ALLLine:所有行
- SPECified:指定行

返回：

查询返回视频触发的同步方式。

示例：

:TRIGger:VIDeo:SYNC ALL /*设置视频触发的同步方式为所有行*/

:TRIGger:VIDeo:SYNC ? /*查询返回所有行*/

:TRIGger:VIDeo:LINe

功能描述：设置或查询视频触发指定行的视频行数。

指令：

:TRIGger:VIDeo:LINe <line>

:TRIGger:VIDeo:LINe ?

参数：

<line>= 整数型，范围 1~625

返回：

查询返回视频触发指定行的视频行数。

示例：

:TRIGger:VIDeo:LINe 10 /*设置视频触发指定行的视频行数为 10*/

:TRIGger:VIDeo:LINe ? /*查询返回 10*/

:TRIGger:VIDeo:LINe:RESet

功能描述：设置视频触发中“同步”为“指定行”时对行数进行复位。

指令：

:TRIGger:VIDeo:LINe:RESet

:TRIGger:NEDGe

:TRIGger:NEDGe:SLOPe

功能描述：设置或查询 N 边沿触发的边沿类型。

指令：

:TRIGger:NEDGe:SLOPe <type>

:TRIGger:NEDGe:SLOPe ?

参数：

<type>= 离散型{RISe | FALL}

- RISe:上升沿
- FALL:下降沿

返回：

查询返回“RISe”、“FALL”。

示例：

:TRIGger:NEDGe:SLOPe RISe /*设置 N 边沿触发的边沿类型为 RISe*/

:TRIGger:NEDGe:SLOPe ? /*查询返回 RISe*/

:TRIGger:NEDGe:SOURce

功能描述：设置或查询 N 边沿触发的触发源。

指令：

:TRIGger:NEDGe:SOURce <source>

:TRIGger:NEDGe:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15}

返回：

查询返回“C1”、“C2”、“C3”、“C4”、“D0”、“D1”、“D2”、“D3”、“D4”、“D5”、“D6”、“D7”、“D8”、“D9”、“D10”、“D11”、“D12”、“D13”、“D14”、“D15”。

示例：

:TRIGger:NEDGe:SOURce C1 /*设置 N 边沿触发源为通道 1*/

:TRIGger:NEDGe:SOURce ? /*查询返回 C1*/

:TRIGger:NEDGe:DURation

功能描述：设置或查询 N 边沿触发的空闲时间。

指令：

:TRIGger:NEDGe:DURation <time>

:TRIGger:NEDGe:DURation ?

参数：

<time>= 实型，单位:ns,us,ms,s，不输入单位则默认单位为 s。设置范围:3.1ns~10s。

返回：

查询返回 N 边沿触发的空闲时间。

示例：

:TRIGger:NEDGe:DURation 1s /*设置 N 边沿触发空闲时间 1s*/

:TRIGger:NEDGe:DURation ? /*查询返回 1.0000000000000E+000*/

:TRIGger:NEDGe:LEVel

功能描述：设置或查询 N 边沿触发电平

.指令：

:TRIGger:NEDGe:LEVel <level>

:TRIGger:NEDGe:LEVel ?

参数：

<level>= 实型，单位:与通道垂直刻度单位一致，默认单位为 V

返回：

查询返回 N 边沿触发电平值。

示例：

:TRIGger:NEDGe:LEVel 1V /*设置 N 边沿触发电平为 1V*/

:TRIGger:NEDGe:LEVel ? /*查询返回 1.00000E+000*/

TRIGger:NEDGe:EDGe

功能描述：设置或查询 N 边沿触发的边沿数。

.指令：

:TRIGger:NEDGe:EDGe <count>

:TRIGger:NEDGe:EDGe ?

参数：

<count>= 整数型，设置范围为 1~65.536k

返回：

查询返回边沿数。

示例：

:TRIGger:NEDGe:EDGe 5 /*设置 N 边沿触发的边沿数为 5*/

:TRIGger:NEDGe:EDGe ? /*查询返回 5*/

:TRIGger:NEDGe:RESet

功能描述：设置 N 边沿触发的触发电平复位。

指令：

:TRIGger:NEDGe:RESet

总线

:TRIGger:BUS

功能描述：设置或查询总线触发的总线通道。

指令：

:TRIGger:BUS <type>

:TRIGger:BUS ?

参数：

<type>= 离散型{B1 | B2}

返回：

查询返回总线触发的总线通道。

示例：

:TRIGger:BUS B1 /*设置总线触发的触发通道为 B1*/

:TRIGger:BUS ? /*查询返回 B1*/

:TRIGger:RS232

:TRIGger:RS232:CONDition

功能描述：设置或查询 RS232 的触发条件。

指令：

:TRIGger:RS232:CONDition <type>

:TRIGger:RS232:CONDition ?

参数：

<type>= 离散型{STARt | END | ERRor | DATA}

- STARt:起始位
- END:停止位
- ERRor:校验错
- DATA:数据

返回：

查询返回 RS232 的触发条件。

示例：

:TRIGger:RS232:CONDition DATA /*设置 RS232 触发条件为 DATA*/

:TRIGger:RS232:CONDition ? /*查询返回 DATA*/

:TRIGger:RS232:DATA

功能描述：设置或查询 RS232 触发条件为数据时的数据值。

指令：

:TRIGger:RS232:DATA <value>

:TRIGger:RS232:DATA ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据，其范围与:BUS<n>:RS232:WIDTh 指令设置的值相关，范围为：0~2^n-1，n 为当前的数据宽度。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:RS232:DATA 01111111 /*设置 RS232 触发数据值为 0x7F*/

:TRIGger:RS232:DATA ? /*查询返回 01111111*/

:TRIGger:I2C

:TRIGger:I2C:CONDition

功能描述： 设置或查询 I2C 的触发条件。

指令：

:TRIGger:I2C:CONDition <type>

:TRIGger:I2C:CONDition ?

参数：

<type>= 离散型{STARt | RESTart | STOP | LOST | ADDRess | DATA | ADDAta}

- STARt:起始位
- RESTart:重启
- STOP:停止位
- LOST:响应失败
- ADDRess:地址
- DATA:数据
- ADDAta:地址和数据

返回：

查询返回 I2C 的触发条件。

示例：

:TRIGger:I2C:CONDition DATA /*设置 RS232 触发条件为 DATA*/

:TRIGger:I2C:CONDition ? /*查询返回 DATA*/

:TRIGger:I2C:DATA

功能描述： 设置或查询 I2C 触发条件为数据和地址数据时的数据值。

指令：

:TRIGger:I2C:DATA <value>

:TRIGger:I2C:DATA ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:I2C:DATA 10001110 /*设置 I2C 触发数据值为 0x8E*/

:TRIGger:I2C:DATA ? /*查询返回 10001110*/

:TRIGger:I2C:DIRection

功能描述：设置或查询 I2C 触发条件为地址和地址数据时的数据方向。

指令：

:TRIGger:I2C:DIRection <direction>

:TRIGger:I2C:DIRection ?

参数：

<direction>= 离散型{WRITe | READ}

- WRITe:写
- READ:读

返回：

查询返回 I2C 触发条件为地址和地址数据时的数据方向。

示例：

:TRIGger:I2C:DIRection WRITe /*设置 I2C 触发数据方向为 WRITe*/

:TRIGger:I2C:DIRection ? /*查询返回 WRITe*/

:TRIGger:I2C:ADDRess

功能描述：设置或查询 I2C 触发条件为地址或地址数据时的地址值。

指令：

:TRIGger:I2C:ADDRess <value>

:TRIGger:I2C:ADDRess ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:I2C:ADDRess 100101 /*设置 I2C 触发地址值为 0x25*/

:TRIGger:I2C:ADDRess ? /*查询返回 100101*/

:TRIGger:I2C:BYTe

功能描述：设置或查询解码数据字节数。

指令：

:TRIGger:I2C:BYTe <value>

:TRIGger:I2C:BYTe ?

参数：

<value>= 整数型 范围：0-5。

返回：

查询返回解码数据字节数。

示例：

:TRIGger:I2C:BYTe 2 /*设置 I2C 数据字节数为 2*/

:TRIGger:I2C:BYTe ? /*查询返回 2*/

:TRIGger:SPI

:TRIGger:SPI:CONDition

功能描述：设置或查询 SPI 触发条件。

指令：

:TRIGger:SPI:CONDition <condition>

:TRIGger:SPI:CONDition ?

参数：

<condition>= 离散型{START | DATA}

- START:起始
- DATA:数据

返回：

查询返回 SPI 触发条件。

示例：

:TRIGger:SPI:CONDition DATA /*设置 SPI 触发条件为 DATA*/

:TRIGger:SPI:CONDition ? /*查询返回 DATA*/

:TRIGger:SPI:DATA

功能描述：设置或查询 SPI 触发条件为数据时的数据值。

指令：

:TRIGger:SPI:DATA <value>

:TRIGger:SPI:DATA ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据，其范围与:TRIGger:SPI:FRAMecount 指令设置的帧数相关。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:SPI:DATA 10001 /*设置 SPI 触发数据值为 0x001*/

:TRIGger:SPI:DATA ? /*查询返回 10001*/

:TRIGger:SPI:FRAMecount

功能描述： 设置或查询 SPI 触发条件为数据时的帧数。

指令：

:TRIGger:SPI:FRAMecount <value>

:TRIGger:SPI:FRAMecount ?

参数：

<value>= 整数型

返回：

查询返回 SPI 触发条件为数据时的帧数。

示例：

:TRIGger:SPI:FRAMecount 2 /*设置 SPI 触发帧数为 2*/

:TRIGger:SPI:FRAMecount ? /*查询返回 2*/

:TRIGger:SPI:WIDTh

功能描述： 设置或查询 SPI 触发条件为数据时的数据位宽。

指令：

:TRIGger:SPI:WIDTh <value>

:TRIGger:SPI:WUDTh ?

参数：

<value>= 整数型 范围：1-64

返回：

查询 SPI 触发条件为数据时的数据位宽。

示例：

:TRIGger:SPI:WIDTh 2 /*设置查询 SPI 触发条件为数据时的数据位宽为 2*/

:TRIGger:SPI:WIDTh ? /*查询返回 2*/

:TRIGger:CAN

:TRIGger:CAN:CONDition

功能描述： 设置或查询 CAN 触发方式。

指令：

:TRIGger:CAN:CONDition <condition>

:TRIGger:CAN:CONDition ?

参数：

<condition>= 离散型{STARtframe | TYPeframe | ID | DATA | IDDAta | ENDFrame | ERRor}

- STARtframe:帧起始
- TYPeframe:帧类型
- ID:ID
- DATA:数据
- IDDAta:ID 和数据
- ENDFrame:帧结尾
- ERRor:错误

返回：

查询返回 CAN 触发方式。

示例：

:TRIGger:CAN:CONDition DATA /*设置 CAN 触发方式为 DATA*/

:TRIGger:CAN:CONDition ? /*查询返回 DATA*/

:TRIGger:CAN:DATA

功能描述： 设置或查询 CAN 触发条件为数据或 ID 和数据时的数据值。

指令：

:TRIGger:CAN:DATA <value>

:TRIGger:CAN:DATA ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:CAN:DATA 1010100 /*设置 CAN 触发数据值为 54h*/

:TRIGger:CAN:DATA ? /*查询返回 1010100*/

:TRIGger:CAN:BYTecount

功能描述：设置或查询 CAN 触发条件为数据或 ID 和数据时的数据字节数。

指令：

:TRIGger:CAN:BYTecount <value>

:TRIGger:CAN:BYTecount ?

参数：

<value>= 整数型 范围：1-8

返回：

查询返回 CAN 触发条件为数据或 ID 和数据时的数据字节数。

示例：

:TRIGger:CAN:BYTecount 2 /*设置 CAN 触发条件为数据或 ID 和数据时的数据字节数为 2*/

:TRIGger:CAN:BYTecount ? /*查询返回 2*/

:TRIGger:CAN:DIRection

功能描述：设置或查询 CAN 触发条件为 ID 或 ID 和数据时的帧方向。

指令：

:TRIGger:CAN:DIRection <direction>

:TRIGger:CAN:DIRection ?

参数：

<direction>= 离散型{READ | WRITe | BOTH}

- READ:读
- WRITe:写
- BOTH:读写

返回：

查询返回 CAN 触发条件为 ID 或 ID 和数据时的帧方向。

示例：

:TRIGger:CAN:DIRection WRITe /*设置 CAN 触发帧方向为 WRITe*/

:TRIGger:CAN:DIRection ? /*查询返回 WRITe*/

:TRIGger:CAN:IDSTandard

功能描述：设置或查询 CAN 触发条件为 ID 或 ID 和数据时的 ID 标准。

指令：

:TRIGger:CAN:IDSTandard <standard>

:TRIGger:CAN:IDSTandard ?

参数：

<standard>= 离散型{STANdard | EXTended | FDSTandard | FDEXtended}

- STANdard:标准
- EXTended:扩展

返回：

查询返回 CAN 触发条件为 ID 或 ID 和数据时的 ID 标准。

示例：

:TRIGger:CAN:IDSTandard EXTended /*设置 CAN 触发 ID 标准为 EXTended*/

:TRIGger:CAN:IDSTandard ? /* 查询返回 EXTended*/

:TRIGger:CAN:STANdardid

功能描述：设置或查询 CAN 触发条件为 ID 或 ID 和数据时的标准 ID。

指令：

:TRIGger:CAN:STANdardid <value>

:TRIGger:CAN:STANdardid ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:CAN:STANdardid 101000 /*设置 CAN 触发标准 ID 为 028h*/

:TRIGger:CAN:STANdardid ? /*查询返回 101000*/

:TRIGger:CAN:EXTendedid

功能描述：设置或查询 CAN 触发条件为 ID 或 ID 和数据时的扩展 ID。

指令：

:TRIGger:CAN:EXTendedid <value>

:TRIGger:CAN:EXTendedid ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:CAN:EXTendedid 101000 /*设置 CAN 触发扩展 ID 为 00028h*/

:TRIGger:CAN:EXTendedid ? /*查询返回 101000*/

:TRIGger:CAN:FTYPe

功能描述：设置或查询 CAN 触发条件为帧类型时的帧类型。

指令：

:TRIGger:CAN:FTYPe <type>

:TRIGger:CAN:FTYPe ?

参数：

<type>= 离散型{DATA | REMote | ERRor | OVERload}

- DATA:数据帧
- REMote:远程帧
- ERRor:错误帧
- OVERload:超载帧

返回：

查询返回 CAN 触发条件为帧类型时的帧类型。

示例：

:TRIGger:CAN:FTYPe DATA /*设置 CAN 触发帧类型为 DATA*/

:TRIGger:CAN:FTYPe ? /*查询返回 DATA*/

:TRIGger:CAN:ETYPe

功能描述：设置或查询 CAN 触发条件为错误时的错误包类型。

指令：

:TRIGger:CAN:ETYPe <type>

:TRIGger:CAN:ETYPe ?

参数：

<type>= 离散型{ANYerror | ACKLose | BITFillerror | CRCerror}

- ANYerror:所有错误
- ACKLose:丢失确认
- BITFillerror:位填充错误
- CRCerror:CRC 错误

返回：

查询返回 CAN 触发条件为错误时的错误包类型。

示例：

:TRIGger:CAN:ETYPe ACKLose /*设置 CAN 触发错误包类型为 ACKLose*/

:TRIGger:CAN:ETYPe ? /*查询返回 ACKLose*/

:TRIGger:CAN_FD

:TRIGger:CAN_FD:CONDition

功能描述： 设置或查询 CAN_FD 触发方式。

指令：

:TRIGger:CAN_FD:CONDition <condition>

:TRIGger:CAN_FD:CONDition ?

参数：

<condition>= 离散型{STARtframe | TYPeframe | ID | DATA | IDDAta | ENDFrame | ERRor }

- STARtframe:帧起始
- TYPeframe:帧类型
- ID:ID
- DATA:数据
- IDDAta:ID 和数据
- ENDFrame:帧结尾
- ERRor:错误

返回：

查询返回 CAN_FD 触发方式。

示例：

:TRIGger:CAN_FD:CONDition IDDAta /*设置 CAN_FD 触发方式为 IDDAta*/

:TRIGger:CAN_FD:CONDition ? /*查询返回 IDDAta*/

:TRIGger:CAN_FD:DATA

功能描述： 设置或查询 CAN_FD 触发条件为数据或 ID 和数据时的数据值。

指令：

:TRIGger:CAN_FD:DATA <value>

:TRIGger:CAN_FD:DATA ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:CAN_FD:DATA 1010100 /*设置 CAN_FD 数据值为 54h*/

:TRIGger:CAN_FD:DATA ? /*查询返回 1010100*/

:TRIGger:CAN_FD:IDSTandard

功能描述：设置或查询 CAN_FD 触发条件为 ID 或 ID 和数据时的 ID 标准。

指令：

:TRIGger:CAN_FD:IDSTandard <standard>

:TRIGger:CAN_FD:IDSTandard ?

参数：

<standard>= 离散型{STANdard | EXTended}

- STANdard:标准
- EXTended:扩展
- FDSTandard:FD 标准
- FDEXtended:FD 扩展

返回：

查询返回 CAN_FD 触发条件为 ID 或 ID 和数据时的 ID 标准。

示例：

:TRIGger:CAN_FD:IDSTandard EXTended /*设置 CAN_FD 触发 ID 标准为 EXTended*/

:TRIGger:CAN_FD:IDSTandard ? /*查询返回 EXTended*/

:TRIGger:CAN_FD:STANdardid

功能描述：设置或查询 CAN_FD 触发条件为 ID 或 ID 和数据时的标准 ID。

指令：

:TRIGger:CAN_FD:STANdardid <value>

:TRIGger:CAN_FD:STANdardid ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:CAN_FD:STANdardid 101000 /*设置 CAN_FD 触发标准 ID 为 0x28*/

:TRIGger:CAN_FD:STANdardid ? /*查询返回 101000*/

:TRIGger:CAN_FD:EXTendedid

功能描述：设置或查询 CAN_FD 触发条件为 ID 或 ID 和数据时的扩展 ID。

指令：

:TRIGger:CAN_FD:EXTendedid <value>

:TRIGger:CAN_FD:EXTendedid ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:CAN_FD:EXTendedid 1111 /*设置 CAN_FD 触发扩展 ID 为 0x0000000F*/

:TRIGger:CAN_FD:EXTendedid ? /*查询返回 1111*/

:TRIGger:CAN_FD:FDSTandardid

功能描述：设置或查询 CAN_FD 触发条件为 ID 或 ID 和数据时的 FD 标准 ID。

指令：

:TRIGger:CAN_FD:FDSTandardid <value>

:TRIGger:CAN_FD:FDSTandardid?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:CAN_FD:FDSTandardid 10001 /*设置 CAN_FD 触发的 FD 标准 ID 为 0x011*/

:TRIGger:CAN_FD:FDSTandardid ? /*查询返回 10001*/

:TRIGger:CAN_FD:FDEXtendedid

功能描述：设置或查询 CAN_FD 触发条件为 ID 或 ID 和数据时的 FD 扩展 ID。

指令：

:TRIGger:CAN_FD:FDEXtendedid <value>

:TRIGger:CAN_FD:FDEXtendedid ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:CAN_FD:FDEXtendedid 10001 /*设置 CAN_FD 触发的 FD 扩展 ID 为 0x011*/

:TRIGger:CAN_FD:FDEXtendedid ? /*查询返回 10001*/

:TRIGger:CAN_FD:FTYPe

功能描述：设置或查询 CAN_FD 触发条件为帧类型时的帧类型。

指令：

:TRIGger:CAN_FD:FTYPe <type>

:TRIGger:CAN_FD:FTYPe ?

参数：

<type>= 离散型{DATA | REMote | ERRor | OVERload}

- DATA:数据帧
- REMote:远程帧
- ERRor:错误帧
- OVERload:超载帧

返回：

查询返回 CAN_FD 触发条件为帧类型时的帧类型。

示例：

:TRIGger:CAN_FD:FTYPe DATA /*设置 CAN_FD 触发帧类型为 DATA*/

:TRIGger:CAN_FD:FTYPe ? /*查询返回 DATA*/

:TRIGger:CAN_FD:ETYPe

功能描述：设置或查询 CAN_FD 触发条件为错误时的错误包类型。

指令：

:TRIGger:CAN_FD:ETYPe <type>

:TRIGger:CAN_FD:ETYPe ?

参数：

<type>= 离散型{ANYerror | ACKLose | BITFillerror | CRCerror}

- ANYerror:所有错误
- ACKLose:丢失确认
- BITFillerror:位填充错误
- CRCerror:CRC 错误

返回：

查询返回 CAN_FD 触发条件为错误时的错误包类型。

示例：

:TRIGger:CAN_FD:ETYPe ACKLose /*设置 CAN_FD 触发错误包类型为 ACKLose*/

:TRIGger:CAN_FD:ETYPe ? /*查询返回 ACKLose*/

:TRIGger:CAN_FD:DLENgth

功能描述：设置或查询 CAN_FD 触发条件为数据或 ID 和数据时的数据字节数。

指令：

:TRIGger:CAN_FD:DLENgth <value>

:TRIGger:CAN_FD:DLENgth ?

参数：

<value>= 数据字节数，范围 1~8

返回：

查询返回 CAN_FD 触发条件为数据或 ID 和数据时的数据字节数。

示例：

:TRIGger:CAN_FD:DLENgth1 /*设置 CAN_FD 触发方式为数据时的数据字节为 1*/

:TRIGger:CAN_FD:DLENgth ? /*查询返回 1*/

:TRIGger:LIN

:TRIGger:LIN:CONDition

功能描述：设置或查询 LIN 触发方式。

指令：

:TRIGger:LIN:CONDition <condition>

:TRIGger:LIN:CONDition ?

参数：

<condition>= 离散型{STARtframe | ID | DATA | IDDAta | WAKeup | SLEep | SYERror | IDCRcerror | SUMCrcerror}

- STARtframe:帧起始
- ID:ID
- DATA:数据
- IDDAta:ID 和数据
- WAKeup:唤醒帧
- SLEep:休眠帧
- SYERror:同步错误
- IDCRcerror:ID 校验错
- SUMCrcerror:校验和错

返回：

查询返回 LIN 触发方式。

示例：

:TRIGger:LIN:CONDition IDDAta /*设置 LIN 触发方式为 IDDAta*/

:TRIGger:LIN:CONDition ? /*查询返回 IDDAta*/

:TRIGger:LIN:DATA

功能描述：设置或查询 LIN 触发条件为数据或 ID 和数据时的数据值。

指令：

:TRIGger:LIN:DATA <value>

:TRIGger:LIN:DATA ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:LIN:DATA 1010100 /*设置 LIN 数据值为 54h*/

:TRIGger:LIN:DATA ? /*查询返回 1010100*/

:TRIGger:LIN:ID

功能描述：设置或查询 LIN 触发条件为 ID 或 ID 和数据时的 ID。

指令：

:TRIGger:LIN:ID <value>

:TRIGger:LIN:ID ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:LIN:ID 101000 /*设置 LIN ID 为 0x28*/

:TRIGger:LIN:ID ? /*查询返回 101000*/

:TRIGger:FLEXRAY

:TRIGger:FR:CONDition

功能描述：设置或查询 FLEXRAY 触发方式。

指令：

:TRIGger:FR:CONDition <condition>

:TRIGger:FR:CONDition ?

参数：

<condition>= 离散型{HEADframe | INDicator | ID | CIRCulate | HEADer | DATA | IDDAta | TAILframe | ERRor}

- HEADframe:帧头
- INDicator:指示位
- ID:ID
- CIRCulate:循环数
- HEADer:标头字段
- DATA:数据
- IDDAta:ID 和数据
- TAILframe:帧结尾
- ERRor:错误

返回：

查询返回 FLEXRAY 触发方式。

示例：

:TRIGger:FR:CONDition DATA /*设置 TRIGger 的 FLEXRAY 触发方式为 DATA*/

:TRIGger:FR:CONDition ? /*查询返回 DATA*/

:TRIGger:FR:DATA:LOW

功能描述：设置或查询 FLEXRAY 触发条件为数据或 ID 和数据时的数据(低位)值。

指令：

:TRIGger:FR:DATA:LOW <value>

:TRIGger:FR:DATA:LOW ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据吗，其范围与:TRIGger:FR:DCOUnt 指令设置的值相关。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:FR:DATA:LOW 1010100 /*设置 FLEXRAY 触发数据(低位)值为 54h*/

:TRIGger:FR:DATA:LOW ? /*查询返回 1010100*/

:TRIGger:FR:DATA:HIGH

功能描述： 设置或查询 FLEXRAY 触发条件为数据或 ID 和数据时的数据(高位)值。

指令：

:TRIGger:FR:DATA:HIGH <value>

:TRIGger:FR:DATA:HIGH ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据吗，其范围与:TRIGger:FR:DCOUnt 指令设置的值相关。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:FR:DATA:HIGH 1010101 /*设置 FLEXRAY 触发数据(高位)值为 55h*/

:TRIGger:FR:DATA:HIGH ? /*查询返回 1010101*/

:TRIGger:FR:DCOUnt

功能描述： 设置或查询 FLEXRAY 触发条件为数据或 ID 和数据时的数据长度。

指令：

:TRIGger:FR:DCOUnt <value>

:TRIGger:FR:DCOUnt ?

参数：

<value>= 整数型

返回：

查询返回 FLEXRAY 触发条件为数据或 ID 和数据时的数据长度。

示例：

:TRIGger:FR:DCOUnt 3 /*设置 FLEXRAY 触发条件为数据或 ID 和数据时的数据长度 3*/

:TRIGger:FR:DCOUnt ? /*查询返回 3*/

:TRIGger:FR:ID

功能描述：设置或查询 FLEXRAY 触发条件为 ID 或 ID 和数据时的标准 ID。

指令：

:TRIGger:FR:ID <value>

:TRIGger:FR:ID ?

参数：

<value>= 整数型 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回 FR 触发条件为 ID 或 ID 和数据时的标准 ID。

示例：

:TRIGger:FR:ID 101000 /*设置 FR 触发条件为 ID 或 ID 和数据时的标准 ID 为 28h*/

:TRIGger:FR:ID ? /*查询返回 101000*/

:TRIGger:FR:INDicator

功能描述：设置或查询 FLEXRAY 触发条件为指示位时指示位类型。

指令：

:TRIGger:FR:INDicator <type>

:TRIGger:FR:INDicator ?

参数：

<value>= 离散型{NMFRame | PLFRame | EPFRame | SNFRame | STFRame}

- NMFRame:正常帧
- PLFRame:净荷帧
- EPFRame:空帧
- SNFRame:同步帧
- STFRame:启动帧

返回：

查询返回 FLEXRAY 触发条件为指示位时指示位类型。

示例：

:TRIGger:FR:INDicator NMFRame /*设置 FLEXRAY 触发数据值为 NMFRame*/

:TRIGger:FR:INDicator ? /*查询返回 NMFRame*/

:TRIGger:FR:CYCLedata

功能描述：设置或查询 FLEXRAY 触发条件为周期数时的循环数据。

指令：

:TRIGger:FR:CYCLedata <value>

:TRIGger:FR:CYCLedata ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:FR:CYCLedata 100000 /*设置 FLEXRAY 触发循环数据值为 0x20*/

:TRIGger:FR:CYCLedata ? /*查询返回 100000*/

:TRIGger:FR:EDFType

功能描述：设置或查询 FLEXRAY 触发条件为结尾帧时的结尾帧类型。

指令：

:TRIGger:FR:EDFType <type>

:TRIGger:FR:EDFType ?

参数：

<type>= 离散型{STATic | DYNamic | ALL}

- STATic:静态
- DYNamic:动态
- ALL:全部

返回：

查询返回 FLEXRAY 触发条件为结尾帧时的结尾帧类型。

示例：

:TRIGger:FR:EDFType DYNamic /*设置 FLEXRAY 触发结尾帧类型为 DYNamic*/

:TRIGger:FR:EDFType ? /*查询返回 DYNamic*/

:TRIGger:FR:ETYPe

功能描述：设置或查询 FLEXRAY 触发条件为错误时的错误包类型。

指令：

:TRIGger:FR:ETYPe <type>

:TRIGger:FR:ETYPe ?

参数：

<type>= 离散型{PGHeadcrc | PGTail | SEFRame | DEFRame | SYFRame | STFRame}

- PGHeadcrc:标头 CRC
- PGTail:帧尾
- SEFRame:空帧，静态
- DEFRame:空帧，动态
- SYFRame:同步帧
- STFRame:启动帧（无同步）

返回：

查询返回 FLEXRAY 触发条件为错误时的错误包类型。

示例：

:TRIGger:FR:ETYPe SEFRame /*设置 FLEXRAY 触发错误包类型为 SEFRame*/

:TRIGger:FR:ETYPe ? /*查询返回 SEFRame*/

:TRIGger:FR:BIAS

功能描述：设置和查询返回 FR 触发条件为数据的偏置开关状态。

指令：

:TRIGger:FR:BIAS <type>

:TRIGger:FR:BIAS ?

参数：

<type>= 布尔型{ON | OFF}或{1 | 0}

- ON:打开
- OFF:关闭

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:TRIGger:FR:BIAS ON /*设置 FR 触发条件为数据的偏置打开*/

:TRIGger:FR:BIAS ? /*查询返回 1*/

:TRIGger:FR:OFFSet

功能描述：设置或查询 FR 触发条件为数据的偏置值。

指令：

:TRIGger:FR:OFFSet <value>

:TRIGger:FR:OFFSet ?

参数：

<value>= 实型 范围：0-253。

返回：

查询返回 FR 触发条件为数据的偏置值。

示例：

:TRIGger:FR:OFFSet 1 /*设置 FR 触发条件为数据的偏置值为 1*/

:TRIGger:FR:OFFSet ? /*查询返回 1*/

:TRIGger:FR:INDicator:DATA

功能描述：设置或查询 FR 触发条件为标头字段的指示位值。

指令：

:TRIGger:FR:INDicator:DATA <value>

:TRIGger:FR:INDicator:DATA ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回 FR 触发条件为标头字段的指示位值。

示例：

:TRIGger:FR:INDicator:DATA 100 /*设置 FR 触发条件为标头字段的指示位值为 04h*/

:TRIGger:FR:INDicator:DATA ? /*查询返回 100*/

:TRIGger:FR:PAYLoad

功能描述：设置或查询 FR 触发条件为标头字段的静荷载长度。

指令：

:TRIGger:FR:PAYLoad <value>

:TRIGger:FR:PAYLoad ?

参数：

<value>=参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回 FR 触发条件为标头字段的静荷载长度。

示例：

:TRIGger:FR:PAYLoad 100 /*设置 FR 触发条件为标头字段的静荷载长度为 04h*/

:TRIGger:FR:PAYLoad ? /*查询返回 100*/

:TRIGger:FR:HEAD

功能描述：设置或查询 FR 触发条件为标头字段的标头 CRC 值。

指令：

:TRIGger:FR:HEAD <value>

:TRIGger:FR:HEAD ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回 FR 触发条件为标头字段的标头 CRC 值。

示例：

:TRIGger:FR:HEAD 100 /*设置 FR 触发条件为标头字段的标头 CRC 值为 04h*/

:TRIGger:FR:HEAD ? /*查询返回 100*/

:TRIGger:AUDIOBUS

:TRIGger:AB:CONDition

功能描述：设置或查询 AUDIOBUS 触发方式。

指令：

:TRIGger:AB:CONDition <type>

:TRIGger:AB:CONDition ?

参数：

<type>= 离散型{SYNC | DATA}

- SYNC:同步
- DATA:数据

返回：

查询返回 AUDIOBUS 触发方式。

示例：

:TRIGger:AB:CONDition DATA /*设置 AUDIOBUS 触发方式为 DATA*/

:TRIGger:AB:CONDition ? /*查询返回 DATA*/

:TRIGger:AB:DATA

功能描述：设置或查询 AUDIOBUS 触发条件为数据时的数据值。

指令：

:TRIGger:AB:DATA <value>

:TRIGger:AB:DATA ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:AB:DATA 1010100 /*设置 AUDIOBUS 触发数据值为 0x00000054*/

:TRIGger:AB:DATA ? /*查询返回 1010100*/

:TRIGger:AB:TYPe

功能描述：设置或查询 AB 触发条件为数据时的通道类型。

指令：

:TRIGger:AB:TYPe <TYPe>

:TRIGger:AB:TYPe ?

参数：

<type>= 离散型{LR | LEFT | RIGHT}。

返回：

查询返回 AB 触发条件为数据时的通道类型。

示例：

:TRIGger:AB:TYPe LEFT /*设置 AB 触发条件为数据时的通道类型为左声道*/

:TRIGger:AB:TYPe ? /*查询返回 LEFT*/

:TRIGger:MIL_STD_1553

:TRIGger:MS:CONDition

功能描述： 设置或查询 MIL_STD_1553 触发条件。

指令：

:TRIGger:MS:CONDition <condition>

:TRIGger:MS:CONDition ?

参数：

<condition>= 离散型{CMD | ERRor | DATA | SYNC}

- CMD:命令/状态字
- ERRor:错误
- DATA:数据
- SYNC:同步

返回：

查询返回 MIL_STD_1553 触发条件。

示例：

:TRIGger:MS:CONDition DATA /*设置 MIL_STD_1553 触发条件为 DATA*/

:TRIGger:MS:CONDition ? /*查询返回 DATA*/

:TRIGger:MS:ADDRess

功能描述： 设置或查询 MS 触发条件为命令/状态字时远程终端地址。

指令：

:TRIGger:MS:ADDRess <value>

:TRIGger:MS:ADDRess ?

参数：

<Value>= 整数型 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回 MS 触发条件为命令/状态字时远程终端地址。

示例：

:TRIGger:MS:ADDRess 100 /*设置 MS 触发条件为命令/状态字时远程终端地址为 4*/

:TRIGger:MS:ADDRess ? /*查询返回 100*/

:TRIGger:MS:DATA

功能描述： 设置或查询 MS 触发条件为命令/状态字或者数据时数据值。

指令：

:TRIGger:MS:DATA <value>

:TRIGger:MS:DATA ?

参数：

<Value>= 整数型 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回 MS 触发条件为命令/状态字或者数据时数据值。

示例：

:TRIGger:MS:DATA 100 /*设置 MS 触发条件为命令/状态字时或者数据时数据值为 4*/

:TRIGger:MS:DATA ? /*查询返回 100*/

:TRIGger:MS:PAIRty

功能描述： 设置或查询 MS 触发条件为命令/状态字或者数据时校验值。

指令：

:TRIGger:MS:PAIRty <value>

:TRIGger:MS:PAIRty ?

参数：

<Value>= 整数型,参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回 MS 触发条件为命令/状态字或者数据时校验值。

示例：

:TRIGger:MS:PAIRty 100 /*设置 MS 触发条件为命令/状态字时或者数据时校验值为 4*/

:TRIGger:MS:PAIRty ? /*查询返回 100*/

:TRIGger:MS:ERRor

功能描述： 设置或查询 MS 触发条件为错误时错误类型。

指令：

:TRIGger:MS:ERRor <type>

:TRIGger:MS:ERRor ?

参数：

<type>= 离散型{ODDevent | SYNC | MANChester | NOTContinue}

- ODDevent:奇偶校验
- SYNC:同步
- MANChester:曼彻斯特
- NOTContinue:非连续数据

返回：

查询返回 MS 触发条件为错误时错误类型。

示例：

:TRIGger:MS:ERRor SYNC /*设置 MS 触发条件为错误时错误类型为同步*/

:TRIGger:MS:ERRor ? /*查询返回 SYNC*/

:TRIGger:ARINC429

:TRIGger:A429:CONDition

功能描述：设置或查询 ARINC429 触发方式。

指令：

:TRIGger:A429:CONDition <condition>

:TRIGger:A429:CONDition ?

参数：

<condition>= 离散型{HDFRame | TLFRame | LABel | SDI | DATA | SSM | LDATa | ERror}

- HDFRame:帧起始
- TLFRame:帧结尾
- LABel:标签
- SDI:sdi
- DATA:数据
- SSM:标志和状态位
- LDATa:标签和数据
- ERror:错误

返回：

查询返回 ARINC429 触发方式。

示例：

:TRIGger:A429:CONDition DATA /*设置 ARINC429 触发方式为 DATA*/

:TRIGger:A429:CONDition ? /*查询返回 DATA*/

:TRIGger:A429:DATA

功能描述：设置或查询 ARINC429 触发条件为数据时的数据值。

指令：

:TRIGger:A429:DATA <value>

:TRIGger:A429:DATA ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:A429:DATA 1010100 /*设置 ARINC429 触发数据值为 0x54*/

:TRIGger:A429:DATA ? /*查询返回 1010100*/

:TRIGger:A429:SDI

功能描述：设置或查询 ARINC429 触发条件为源或目的标识符时的值。

指令：

:TRIGger:A429:SDI <value>

:TRIGger:A429:SDI ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:A429:SDI 11 /*设置 ARINC429 触发 SDI 值为 0x03*/

:TRIGger:A429:SDI ? /*查询返回 11*/

:TRIGger:A429:SSM

功能描述：设置或查询 ARINC429 触发条件为标志和状态位的值。

指令：

:TRIGger:A429:SSM <value>

:TRIGger:A429:SSM ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:A429:SSM 11 /*设置 ARINC429 触发 SSM 值为 0x03*/

:TRIGger:A429:SSM ? /*查询返回 11*/

:TRIGger:A429:LABel

功能描述：设置或查询 ARINC429 触发条件为标签时的标签值。

指令：

:TRIGger:A429:LABel <value>

:TRIGger:A429:LABel ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回 ARINC429 触发条件为标签时的标签值。

示例：

:TRIGger:A429:LABel 11 /*设置 ARINC429 触发标签值为 0x03*/

:TRIGger:A429:LABel ? /*查询返回 11*/

:TRIGger:A429:ERRor

功能描述：设置或查询 A429 触发条件为错误时错误类型。

指令：

:TRIGger:A429:ERRor <value>

:TRIGger:A429:ERRor ?

参数：

<type>= 离散型{CRC | WORD | GAP | ANY}

- CRC:检验位错误
- WORD:字错误
- GAP:帧间隙错误
- ANY:任意错误

返回：

询返回 A429 触发条件为错误时错误类型。

示例：

:TRIGger:A429:ERRor WORD /*设置 A429 触发条件为错误时错误类型为字错误*/

:TRIGger:A429:ERRor ? /*查询返回 WORD*/

:TRIGger:USB

:TRIGger:USB:CONDition

功能描述：设置或查询 USB 触发条件。

指令：

:TRIGger:USB:CONDition <value>

:TRIGger:USB:CONDition ?

参数：

<value>= 离散型{SYNC | RESet | PAUSe | RESume | EDPackage | TOKPackage | HDSPackage | DTAPackage | SPECial | ERRor}

- SYNC:同步
- RESet:复位
- PAUSe:暂停
- RESume:恢复
- EDPackage:包尾
- TOKPackage:令牌包
- HDSPackage:握手包
- DTAPackage:数据包
- SPECial:特殊
- ERRor:错误

返回：

查询返回 USB 触发条件。

示例：

:TRIGger:USB:CONDition PAUSe /*设置 USB 触发条件为暂停*/

:TRIGger:USB:CONDition ? /*查询返回 PAUSe*/

:TRIGger:SENT

:TRIGger:SENT:CONDition

功能描述： 设置或查询 SENT 触发条件。

指令：

:TRIGger:SENT:CONDition <condition>

:TRIGger:SENT:CONDition ?

参数：

<condition>= 离散型{SYNC | STATus | DATA | CRC | STD | STDC | ERRor}

- SYNC:同步
- STATus:状态
- DATA:数据
- CRC:校验错
- STD:状态+数据
- STDC:状态+数据+CRC
- ERRor:错误

返回：

查询返回 SENT 触发条件。

示例：

:TRIGger:SENT:CONDition DATA /*设置 SENT 触发条件为 DATA*/

:TRIGger:SENT:CONDition ? /*查询返回 DATA*/

:TRIGger:SENT:DATA

功能描述： 设置或查询 SENT 触发条件为数据时的数据值。

指令：

:TRIGger:SENT:DATA <value>

:TRIGger:SENT:DATA ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回二进制字符串数据。

示例：

:TRIGger:SENT:DATA 100 /*设置 SENT 触发数据值为 0x4h*/

:TRIGger:SENT:DATA ? /*查询返回 100*/

:TRIGger:SENT:STATus

功能描述：设置或查询 SENT 触发条件为状态时的状态值。

指令：

:TRIGger:SENT:STATus <value>

:TRIGger:SENT:STATus ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回 SENT 触发条件为状态时的状态值。

示例：

:TRIGger:SENT:STATus 11 /*设置 SENT 触发状态为 0x3h*/

:TRIGger:SENT:STATus ? /*查询返回 11*/

:TRIGger:SENT:DLEN

功能描述：设置或查询 SENT 触发条件为数据时的数据长度。

指令：

:TRIGger:SENT:DLEN <value>

:TRIGger:SENT:DLEN ?

参数：

<value>= 整数型 范围：1-6。

返回：

查询返回 SENT 触发条件为数据时的数据长度。

示例：

:TRIGger:SENT:DLEN 1 /*设置 SENT 触发数据长度为 1*/

:TRIGger:SENT:DLEN ? /*查询返回 1*/

:TRIGger:SENT:CRC

功能描述：设置或查询 SENT 触发条件为 CRC 时的 CRC 值。

指令：

:TRIGger:SENT:CRC <value>

:TRIGger:SENT:CRC ?

参数：

<value>= 参数为 0 或 1 表示的二进制字符串数据。

返回：

查询返回 SENT 触发条件为 CRC 时的 CRC 值。

示例：

:TRIGger:SENT:CRC 11 /*设置 SENT 触发条件为 CRC 时的 CRC 值 3h*/

:TRIGger:SENT:CRC ? /*查询返回 11*/

:TRIGger:SENT:ERR

功能描述：设置或查询 SENT 触发条件为错误时的错误选项值。

指令：

:TRIGger:SENT:ERR <type>

:TRIGger:SENT:ERR ?

参数：

<type>= 离散型{CRC | CONPuls}

- CRC:快速 CRC 错误
- CONPuls:连续脉冲错误

返回：

查询返回 SENT 触发条件为错误时的错误选项值。

示例：

:TRIGger:SENT:ERR CRC /*设置 SENT 触发条件为错误时的错误选项值为快速 CRC 错误*/

:TRIGger:SENT:ERR ? /*查询返回 CRC*/

:TRIGger:SPMi

:TRIGger:SPMi:CONDition

功能描述： 设置或查询 SPMi 触发条件。

指令：

:TRIGger:SPMi:CONDition <value>

:TRIGger:SPMi:CONDition ?

参数：

<value>= 离散型{EXREGWRITe | EXREGREAD | LONGExREGWRITe | LONGEXREGREAD | REGWRITe | REGREAD | MASTERWRITe | MASTERREAd | DDMASTERWRITe | DDMASTERREAd | REGWRITEZERo | RESet | DORMancy | STOP | WAKeup | IDENtify | BUSRIGHTTransfer | ERRor}

- EXREGWRITe:扩展寄存器写
- EXREGREAD:扩展寄存器读
- LONGExREGWRITe:扩展寄存器写（长）
- LONGEXREGREAD:扩展寄存器读（长）
- REGWRITe:寄存器写
- REGREAD:寄存器读
- MASTERWRITe:主机写
- MASTERREAd:主机读
- DDMASTERWRITe:DD 块主机读
- DDMASTERREAd:DD 块从机读
- REGWRITEZERo:寄存器写 0
- RESet:复位
- DORMancy:休眠
- STOP:停止
- WAKeup:唤醒
- IDENtify:身份识别
- BUSRIGHTTransfer:总线权移交
- ERRor:校验错

返回：

查询返回 SPMi 触发条件。

示例：

:TRIGger:SPMi:CONDition STOP /*设置 SPMi 触发条件为停止*/

:TRIGger:SPMi:CONDition ? /*查询返回 STOP*/

:TRIGger:ETHernet

:TRIGger:ETHernet:CONDition

功能描述：设置或查询触发方式。

指令：

:TRIGger:ETHernet:CONDition <type>

:TRIGger:ETHernet:CONDition ?

参数：

<type>= 离散型{HEADframe | IDLE | DSTAddress | SRCAddress | TYPE | DATA | CRCError}

- HEADframe:帧头
- IDLE:空闲
- DSTAddress:MAC 目的地址
- SRCAddress:MAC 源地址
- TYPE:类型
- DATA:数据
- CRCError:CRC 校验错

返回：

查询返回触发方式。

示例：

:TRIGger:ETHernet:CONDition IDLE /*设置触发方式为空闲*/

:TRIGger:ETHernet:CONDition ? /*查询返回 IDLE*/

:TRIGger:ETHernet:DSTMac

功能描述：设置或查询 MAC 目的地址。

指令：

:TRIGger:ETHernet:DSTMac <value>

:TRIGger:ETHernet:DSTMac ?

参数：

<value>= 十六进制字符串

返回：

查询返回 MAC 目的地址。

示例：

:TRIGger:ETHernet:DSTMac "8C:8C:AA:17:E4:03" /*设置 MAC 目的地址为"8C:8C:AA:17:E4:03*/

:TRIGger:ETHernet:DSTMac ? /*查询返回 8C:8C:AA:17:E4:03*/

:TRIGger:ETHernet:SRCMac

功能描述： 设置或查询 MAC 源地址。

指令：

:TRIGger:ETHernet:SRCMac <value>

:TRIGger:ETHernet:SRCMac ?

参数：

<value>= 十六进制字符串

返回：

查询返回 MAC 源地址。

示例：

:TRIGger:ETHernet:SRCMac "8C:8C:AA:17:E4:03" /*设置 MAC 源地址为"8C:8C:AA:17:E4:03*/

:TRIGger:ETHernet:SRCMac ? /*查询返回 8C:8C:AA:17:E4:03*/

:TRIGger:ETHernet:TYPMac

功能描述： 设置或查询 MAC 长度/类型。

指令：

:TRIGger:ETHernet:TYPMac <value>

:TRIGger:ETHernet:TYPMac ?

参数：

<value>= 整数型

返回：

查询返回 MAC 长度/类型。

示例：

:TRIGger:ETHernet:TYPMac 1234 /*设置 MAC 长度/类型为 1234*/

:TRIGger:ETHernet:TYPMac ? /*查询返回 1234*/

:TRIGger:ETHernet:DATA

功能描述：设置或查询数据值。

指令：

:TRIGger:ETHernet:DATA <value>

:TRIGger:ETHernet:DATA ?

参数：

<value>= 整数型

返回：

查询返回数据值。

示例：

:TRIGger:ETHernet:DATA 22 /*设置数据值 22*/

:TRIGger:ETHernet:DATA ? /*查询返回 22*/

:TRIGger:ETHernet:DATALength

功能描述：设置或查询字节长度。

指令：

:TRIGger:ETHernet:DATALength <value>

:TRIGger:ETHernet:DATALength ?

参数：

<value>= 整数型

返回：

查询返回字节长度。

示例：

:TRIGger:ETHernet:DATALength 2 /*设置字节长度 2*/

:TRIGger:ETHernet:DATALength ? /*查询返回 2*/

:TRIGger:ETHernet:DATAOffset

功能描述：设置或查询数据偏移。

指令：

:TRIGger:ETHernet:DATAOffset <value>

:TRIGger:ETHernet:DATAOffset ?

参数：

<value>= 整数型

返回：

查询返回数据偏移。

示例：

:TRIGger:ETHernet:DATAOffset 4 /*设置数据偏移 4*/

:TRIGger:ETHernet:DATAOffset ? /*查询返回 4*/

:TRIGger:PSI5

:TRIGger:PSI5:CONDition

功能描述：设置或查询 PSI5 触发方式。

指令：

:TRIGger:PSI5:CONDition <type>

:TRIGger:PSI5:CONDition ?

参数：

<type>= 离散型{STARt | DATAA | DATAB | CRC | PARity | IDBlock | STATus}

- STARt:帧起始
- DATAA:数据域 A
- DATAB:数据域 B
- CRC:CRC 校验
- PARity:奇偶校验
- IDBlock:ID 块
- STATus:传感器状态

返回：

查询返回触发方式。

示例：

:TRIGger:PSI5:CONDition DATAB /*设置触发方式为数据域 B*/

:TRIGger:PSI5:CONDition ? /*查询返回 DATAB*/

:TRIGger:PSI5:DATA

功能描述：设置或查询输数据

指令：

:TRIGger:PSI5:DATA<n><value>

:TRIGger:PSI5:DATA<n> ?

参数：

<n>= 1,2,3,4

- 1：表示触发方式为数据域 A 时的数据值
- 2：触发方式为数据域 B 时的数据值

- 3：触发方式为 ID 块时的数据值
- 4：触发方式为传感器状态时的数据值

<value>= 整数型

返回：

查询返回数据值。

示例：

:TRIGger:PSI5:DATA1 13 /*设置数据域 A 的数据值为 13*/

:TRIGger:PSI5:DATA1 ? /*查询返回 13*/

WAVeform 指令系统

用于读取示波器波形数据及其相关设置。

:WAVeform:SOURce

功能描述：设置或查询波形数据读取的通道源。

当通道源为 math 通道时，只有 ASCII 格式数据有效。

指令：

:WAVeform:SOURce <source>

:WAVeform:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8}

返回：

查询返回“C1”、“C2”、“C3”、“C4”、“M1”、“M2”、“M3”、“M4”、“M5”、“M6”、“M7”、“M8”。

示例：

:WAVeform:SOURce C1 /*设置波形数据读取的通道源为通道 1*/

:WAVeform:SOURce ? /*查询返回 C1*/

:WAVeform:MODE

功能描述：设置或查询读取波形数据的类型。

指令：

:WAVeform:MODE <mode>

:WAVeform:MODE ?

参数：

<mode>= 离散型{NORMal | RAW | RECord}

- NORMal:屏幕显示数据
- RAW:内存波形数据
- RECord:读取顺序模式波形

返回：

查询返回“NORMal”、“RAW”、“RECord”。

示例：

:WAVeform:MODE NORMal /*设置读取波形数据的类型为 NORMal*/

:WAVeform:MODE ? /*查询返回 NORMal*/

:WAVeform:FORMat

功能描述：设置或查询波形数据的返回格式。

指令：

:WAVeform:FORMat <format>

:WAVeform:FORMat ?

参数：

<format>=离散型{BYTE | ASCII | WORD}

- BYTE：单字节数据，如果单个波形点 AD 值占一个字节，只有 8 位 ADC 时有效，仅支持“C1”、“C2”、“C3”、“C4”，返回模拟通道波形数据。
- WORD：双字节数据，如果单个波形点 AD 值占两个字节，8 位 ADC 时，低 8 位有效，高 8 位为 0。
- ASCII：以科学计数形式返回各波形点的实际电压值，各电压值之间以逗号隔开且符合数据块格式。当通道源为 math 通道时，只有 ASCII 格式数据有效，返回屏幕显示的数学通道波形电压数据。

返回：

查询返回“BYTE”、“WORD”、“ASCII”。

示例：

:WAVeform:FORMat BYTE /*设置波形数据的返回格式为 BYTE*/

:WAVeform:FORMat ? /*查询返回 BYTE*/

:WAVeform:STARt

功能描述：设置或查询欲读取的波形起始位置。

指令：

:WAVeform:STARt <value>

:WAVeform:STARt ?

参数：

<value>= 整数型

返回：

查询返回欲读取的波形起始位置。

示例：

:WAVeform:STARt 0 /*设置读取波形起始位置为 0(即第一个点)*/

:WAVeform:STARt ? /*查询返回 0*/

:WAVeform:POINts

功能描述：设置或查询欲读取的波形点数。

指令：

:WAVeform:POINts <point>

:WAVeform:POINts ?

参数：

<point>= 整数型

返回：

查询返回欲读取的波形点数。

示例：

:WAVeform:POINts 10000 /*设置欲读取的波形点数为 10000*/

:WAVeform:POINts ? /*查询返回 10000*/

:WAVeform:FRAMe:ID

功能描述：设置或查询读取顺序模式波形时帧号。

指令：

:WAVeform:FRAMe:ID <value>

:WAVeform:FRAMe:ID ?

参数：

<value>= 整数型

返回：

查询返回读取顺序模式波形时帧号。

示例：

:WAVeform:FRAMe:ID 40 /*设置读取顺序模式波形第 40 帧*/

:WAVeform:FRAMe:ID ? /*查询返回 40*/

:WAVeform:DATA

功能描述：读取当前屏幕显示波形数据。

指令：

:WAVeform:DATA ?

返回：

查询返回波形数据。返回格式与当前选择的波形数据返回格式:WAVeform:FORMat 有关。

示例：

获得指定数据源的波形数据指令顺序执行如下：

获得屏幕波形数据流程

:WAVeform:SOURce C1 /*设置当前要查询波形数据的信号源为通道一*/

:WAVeform: MODE NORMal /*设置读取屏幕显示波形数据*/

:WAVeform:FORMat WORD /*波形数据的返回格式为双字模式*/

:WAVeform:STARt 1000 /*波形数据从第 1000 个点开始读取*/

:WAVeform:POINts 10000 /*设置欲读取的波形点数为 10000*/

:WAVeform:DATA ? /*获得波形数据*/

获得内存波形数据流程，当存储深度过大时，需分块读取，此流程只有在示波器停止状态下有效

:WAVeform:SOURce C1 /*设置当前要查询波形数据的信号源为通道一*/

:WAVeform: MODE RAW /*设置读取内存波形数据*/

:WAVeform:FORMat WORD /*波形数据的返回格式为双字模式*/

:WAVeform:STARt 1000 /*设置读取内存波形起始点为第 1000 个点*/

:WAVeform:POINts 5000 /*读取内存波形点数为 5000*/

循环发送读取波形指令以获取整个内存数据

:WAVeform:DATA ? /*获得内存中一块的波形数据*/

:WAVeform: STARt ? /*直到波形数据读取的起始位置等于-1 表示已到最后一个点*/

MEASure 指令系统

用于控制或查询示波器参数测量信息。

测量参数表

类型	参数	描述	多信源
垂直	最大值（MAX）	采样点的最大值，指波形显示的最高值	
垂直	最小值（MIN）	采样点的最小值，指波形显示的最低值	
垂直	峰峰值（PKPK）	最大值与最小值之间的差值	
垂直	顶值（TOP）	波形中点以上出现频率最高的值	
垂直	底值（BASE ）	波形中点以下出现频率最高的值	
垂直	中值（MID）	顶端值与底端值之间的平均值	
垂直	幅度（AMPL）	顶端值与底端值之间的差值	
垂直	平均值（AVE）	波形区域内所有采样点数值的算术平均值	
垂直	中位数（MEDI）	波形区域内所有波形数据的中位数	
垂直	有效值（RMS）	波形区域内所有波形数据电压值的均方根值	
垂直	方差（VAR）	波形区域内所有波形数据移除直流分量后的电压值方差	
垂垂	标准差（STDD）	波形区域内所有波形数据移除直流分量后的电压值均方根值	
垂直	正过冲（POST）	上升沿跃迁后的失真	
垂直	负过冲（NOST）	下降沿跃迁后的失真	
垂直	周期最大值（CMAX）	波形在一个周期内的最大值	

垂直	周期最小值（CMIN）	波形在一个周期内的最小值	
垂直	周期有效值（CRMS）	波形在一个周期内的均方根值	
垂直	周期方差（CVAR）	波形在一个周期内的方差	
垂直	周期标准差（CSTDD）	波形在一个周期内的标准差	
垂直	周期平均值（CAVE）	波形在一个周期内的平均值	
垂直	周期中位数（CMEDI）	波形在一个周期内的中位数	
垂直	周期峰峰值（CPK）	波形在一个周期内的峰峰值	
垂直	周期顶值（CTOP）	波形在一个周期内顶值	
垂直	周期底值（CBASE）	波形在一个周期内底值	
垂直	周期中值（CMID）	波形在一个周期内中间值	
垂直	周期幅度（CAMPL）	波形在一个周期内幅度	
垂直	正预冲（PPST）	波形的正预冲是指上升沿跃迁前的失真	
垂直	负预冲（NPST）	波形的负预冲是指下降沿跃迁前的失真	
水平	周期（PRD）	周期是完成一个重复周期所需的时间	
水平	频率（FREQ）	频率定义为周期的倒数	
水平	上升时间（RISE）	边沿从最低参考电平（RB）升到最高参考电平（RT）所需的时间	
水平	下降时间（FALL）	边沿从最高参考电平（RT）降到最低参考电平（RB）所需的时间	
水平	正脉宽（PWID）	上升沿的中间阀值到下一个下降沿的中间阈值的时间	
水平	正脉冲串宽度（PSWID）	从波形区域的左侧开始第一个上升沿的中间参考电平（RM）到最后一个下降沿中间参考电平（RM）经历的时间。RM 是幅值的 50%	

水平	负脉宽（NWID）	下降沿的中间阀值到下一个上升沿的中间阀值的时间	
水平	负脉冲串宽度（NSWID）	从波形区域的左侧开始第一个下降沿的中间参考电平（RM）到最后一个上升沿的中间参考电平（RM）经历的时间。RM 是幅值的 50%	
水平	正占空比（DUTY）	波形的正脉冲宽度和周期的比率	
水平	负占空比（NDTY）	波形的负脉冲宽度和周期的比率	
水平	时间@Max（TMAX）	从波形区域的左侧开始第一次出现波形最大值时的水平轴值	
水平	周期时间@MAX（CTMAX）	波形在一个周期内最大值时的水平轴值（通常为时间）。对于周期信号，最大值的位置在整个波形内可能会变动	
水平	时间@Min（TMIN）	从波形区域的左侧开始第一次出现波形最小值时的水平轴值	
水平	周期时间@MIN（CTMIN）	波形在一个周期内最小值时的水平轴值（通常为时间）。对于周期信号，最小值的位置在整个波形内可能会变动	
水平	相邻周期抖动（CCJ）	两个相邻周期波形的周期差值	
水平	上升时间@Lv（LRIS）	在指定电平处波形的上升时间	
水平	下降时间@Lv（LFAL）	在指定电平处波形的下降时间	
水平	周期@Lv（LPRD）	在指定电平处波形的周期时间	
水平	频率@Lv（LFRQ）	在指定电平处波形的频率	
水平	脉宽@Lv（LWID）	在指定电平处波形的脉冲宽度	
水平	占空比@Lv（LDUT）	在指定电平处波形的占空比	
水平	相位差@Lv（LPHA）	在指定电平处两个源第一个上升沿 50%处的相位差	√
水平	RRD@Lv（LRRD）	计算两个源第一个上升沿指定电平处的时间差	√

水平	FFD@Lv（LFFD）	计算两个源第一个下降沿指定电平处的时间差	√
水平	RFD@Lv（LRFD）	计算从第一个源上升沿到第二个源下降沿指定电平处的时间差	√
水平	FRD@Lv（LFRD）	计算从第一个源下降沿到第二个源上升沿指定电平处的时间差	√
水平	偏移（SKEW）	计算从第一个源的上升沿 50%处到第二个源的上升沿 50%处的时间差	√
水平	数据点数（WLEN）	参与测量的波形采样点的样本数目	
水平	周期数（CYCL）	周期波形在波形区域内的循环次数	
水平	上升沿数（RISES）	波形区域内的上升沿数量	
水平	下降沿数（FALLS）	波形区域内的下降沿数量	
水平	正脉冲数（POSPULSES）	波形区域内的正脉冲数量	
水平	负脉冲数（NEGPULSES）	波形区域内的负脉冲数量	
水平	建立时间（STPT）	从超过数据源上指定中间参考电平到最近一次超过时钟源上的指定中间参考电平的时间	√
水平	保持时间（HLDT）	从超过时钟源上的指定中间参考电平到最近一次超过数据源上的指定中间参考电平的时间	√
水平	稳态时间（SETPT）	从指定起始电平达到指定稳态电平区域的时间	√
其他	面积（AREA）	波形的面积是指测量波形和接地电平之间的面积	
其他	绝对面积（ABSAREA）	测量波形与接地电平之间的面积的绝对值之和，即将波形位于接地电平上方和下方的所有面积均取绝对值后进行累加。与代数面积不同，正负面积不会相互抵消，能反映波形整体偏离接地电平的总能量	
其他	正面积（POSAREA）	屏幕内所有大于 GND(地)的电压和时间乘积的代数和	√
其他	负面积（NEGAREA）	屏幕内所有小于 GND(地)的电压和时间乘积的代数和	√
其他	周期面积（CAR）	波形在一个周期内的面积	√

其他	周期绝对面积（CYCABSAREA）	波形在一个周期内的绝对面积	√
其他	周期正面积（CYCPOSAREA）	一个周期内所有大于 GND(地)的电压和时间乘积的代数和	√
其他	周期负面积（CYCNEGAREA）	一个周期内所有小于 GND(地)的电压和时间乘积的代数和	√
其他	上升沿斜率（RISESLOPE）	上升沿穿越的高参考电平（RT）与低参考电平（RB）的差值除以穿越经历的上升时间。RB 是幅值的 10%，RT 是幅值的 90%	
其他	下降沿斜率（FALLSLOPE）	下降沿穿越的高参考电平（RT）与低参考电平（RB）的差值除以穿越经历的下降时间。RT 是幅值的 90%，RB 是幅值的 10%	

参数快照

:MEASure:QKMEasure:DISPlay

功能描述： 设置或查询参数快照测量开关状态。

指令：

:MEASure:QKMEasure:DISPlay <active>

:MEASure:QKMEasure:DISPlay?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:MEASure:QKMEasure:DISPlay ON /*设置参数快照测量开启*/

:MEASure:QKMEasure:DISPlay ? /*查询返回 1*/

:MEASure:QKMEasure:SOURce

功能描述： 设置或查询参数快照测量源。

指令：

:MEASure:QKMEasure:SOURce <source>

:MEASure:QKMEasure:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回“C1”、“C2”、“C3”、“C4”。

示例：

:MEASure:QKMEasure:SOURce C1 /*设置参数快照测量源为通道 1*/

:MEASure:QKMEasure:SOURce ? /*查询返回 C1*/

:MEASure:QKMEasure:DATA

功能描述：查询参数快照中所有测量参数的结果。

其中参数快照包含以下测量参数，并按照如下排列：请参考测量参数表

"MAX", "MIN", "PKPK", "TOP", "BASE", "MID", "AMPL", "AVE", "MEDI", "RMS", "VAR", "STDD", "POST", "NOST", "PRD", "FREQ", "RISE", "FALL", "PWID", "PSWID", "NWID", "NSWID", "DUTY", "NDTY", "TMAX", "CTMAX", "TMIN", "CTMIN", "CMAX", "CMIN", "CRMS", "CVAR", "CSTDD", "CAVE", "CMEDI", "CPK", "CTOP", "CBASE", "CMID", "CAMPL", "CCJ", "AREA", "ABSAREA", "RISESLOPE", "FALLSLOPE", "WLEN", "CYCL", "RISES", "FALLS", "POSPULSES", "NEGPULSES", "PPST", "NPST"

指令：

:MEASure:QKMEasure:DATA ?

返回：

查询返回相应通道参数快照测量所有结果，符合数据块格式，以科学计数法表示，并按顺序以逗号按分隔排列，无效值使用实型数据最大值表示。

例如：“#90000004475.26667E-001,-5.20000E-001,1.04667E+000,5.07246E-001,...”

参数测量

:MEASure:ALL:ACTive

功能描述：设置所有测量项是否激活。

指令：

:MEASure:ALL:ACTive <active>

参数：

<active>= {ON | OFF}或{1 | 0}

示例：

:MEASure:ALL:ACTive ON /*设置所有测量项都激活*/

:MEASure:STOPMEASure

功能描述：设置或查询暂停态测量开启状态

指令：

:MEASure:STOPMEASure <value>

:MEASure:STOPMEASure ?

参数：

<value>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回暂停态测量开启状态

示例：

:MEASure:STOPMEASure ON /*设置暂停态测量开启*/

:MEASure:STOPMEASure ? /*查询返回 1*/

:MEASure:ULTRaacq

功能描述：设置快速开启垂直类型或水平类型的多个测量项。

指令：

:MEASure:ULTRaacq <mode>

参数：

<mode>= 离散型{VERTical | HORizontal}

- VERTical:垂直
- HORizontal:水平

示例：

:MEASure:ULTRaacq VERTical /*设置快速开启垂直类型的多个测量项*/

:MEASure:QKMEasure:SOURce ? /*查询返回 C1*/

:MEASure:GATe

功能描述：设置或查询测量门限。

指令：

:MEASure:GATe <type>

:MEASure:GATe ?

参数：

<type>= 离散型{SCReen | CURSor}

- SCReen:屏幕
- CURSor:光标

返回：

查询返回“SCReen”、“CURSor”。

示例：

:MEASure:GATe SCReen /*设置测量门限为屏幕*/

:MEASure:GATe ? /*查询返回 SCReen*/

:MEASure:INDicator

功能描述：设置或查询指示器。

指令：

:MEASure:INDicator <indicator>

:MEASure:INDicator ?

参数：

<indicator>= 离散型{OFF | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10}

返回：

查询返回“OFF”、“P1”、“P2”、“P3”、“P4”、“P5”、“P6”、“P7”、“P8”、“P9”、“P10”。

示例：

:MEASure:INDicator P1 /*设置 P1 的指示器打开*/

:MEASure:INDicator ? /*查询返回 P1*/

:MEASure:ITEM<n>:DISPlay

功能描述：设置或查询单个测量项是否激活。

指令：

:MEASure:ITEM<n>:DISPlay <active>

:MEASure:ITEM<n>:DISPlay ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:MEASure:ITEM1:DISPlay ON /*设置单个测量项 P1 打开*/

:MEASure:ITEM1:DISPlay ? /*查询返回 1*/

:MEASure:STATistics:DISPlay

功能描述：设置或查询统计是否打开。

指令：

:MEASure:STATistics:DISPlay <active>

:MEASure:STATistics:DISPlay ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:MEASure:STATistics:DISPlay ON /*设置统计打开*/

:MEASure:STATistics:DISPlay ? /*查询返回 1*/

:MEASure:STATistics:COUNt

功能描述：设置或查询测量统计次数。

指令：

:MEASure:STATistics:COUNt <value>

:MEASure:STATistics:COUNt ?

参数：

<value>= 实型

返回：

查询返回测量统计次数。

示例：

:MEASure:STATistics:COUNt 2000 /*设置测量统计次数为 2000*/

:MEASure:STATistics:COUNt ? /*查询返回 2000*/

:MEASure:STATistics:RESet

功能描述：设置统计复位。

指令：

:MEASure:STATistics:RESet

:MEASure:ITEM<n>:TYPe

功能描述：设置或查询测量项类型。

指令：

:MEASure:ITEM<n>:TYPe <type>

:MEASure:ITEM<n>:TYPe ?

参数：

<n>= 离散型{1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 }。表示第 n 项测量参数。

<type>= 离散型，表示测量参数类型，请参考测量参数表。

{MAX | MIN | PKPK | TOP | BASE | MID | AMPL | AVE | MEDI | RMS | VAR | STDD | POST | NOST | CMAX | CMIN | CRMS | CVAR | CSTDD | CAVE | CMEDI | CPK | CTOP | CBASE | CMID | CAMPL | PPST | NPST | PRD | FREQ | RISE | FALL | PWID | PSWID | NWID | NSWID | DUTY | NDTY | TMAX | CTMAX | TMIN | CTMIN | CCJ | LRIS | LFAL | LPRD | LFRQ | LWID | LDUT | LPHA | LRRD | LFFD | LRFD | LFRD | SKEW | WLEN | CYCL | RISES | FALLS | POSPULSES | NEGPULSES | STPT | HLDT | SETPT | AREA | ABSAREA | POSAREA | NEGAREA | CAR | CYCABSAREA | CYCPOSAREA | CYCNEGAREA | RISESLOPE | FALLSLOPE}以及参数运算{+ | - | * | /}

说明：参数运算仅在有两个及以上参数打开时有效。

返回：

查询返回"MAX"、"MIN"、"PKPK"、"TOP"、"BASE"、"MID"、"AMPL"、"AVE"、"MEDI"、"RMS"、"VAR"、"STDD"、"POST"、"NOST"、"CMAX"、"CMIN"、"CRMS"、"CVAR"、"CSTDD"、"CAVE"、"CMEDI"、"CPK"、"CTOP"、"CBASE"、"CMID"、"CAMPL"、"PPST"、"NPST"、"PRD"、"FREQ"、"RISE"、"FALL"、"PWID"、"PSWID"、"NWID"、"NSWID"、"DUTY"、"NDTY"、"TMAX"、"CTMAX"、"TMIN"、"CTMIN"、"CCJ"、"LRIS"、"LFAL"、"LPRD"、"LFRQ"、"LWID"、"LDUT"、"LPHA"、"LRRD"、"LFFD"、"LRFD"、"LFRD"、"SKEW"、"WLEN"、"CYCL"、"RISES"、"FALLS"、"POSPULSES"、"NEGPULSES"、"STPT"、"HLDT"、"SETPT"、

"AREA"、"ABSAREA"、"POSAREA"、"NEGAREA"、"CAR"、"CYCABSAREA"、"CYCPOSAREA"、"CYCNEGAREA"、"RISESLOPE"、"FALLSLOPE"以及参数运算。

示例：

:MEASure:ITEM3:TYPe "P2+P1" /*设置测量位 P3 的测量项为 P2+P1*/

:MEASure:ITEM3:TYPe ? /*查询返回 P2+P1*/

:MEASure:ITEM<n>:SOURce<ch>

功能描述：设置或查询测量位测量源。

指令：

:MEASure:ITEM<n>:SOURce<ch> <source>

:MEASure:ITEM<n>:SOURce<ch> ?

参数：

<n>= 离散型{1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 }。表示第 n 项测量参数。

<source>= 离散型{C1 | C2 | C3 | C4 | R1 | R2 | R3 | R4}

<ch>= 离散型{1 | 2}，表示信源 1 或信源 2。

如果测量参数是单信源则只需设置信源 1，信源 2 可以省略。

如果测量参数是多信源则需要设置信源 1 和信源 2，信源 2 不可以省略。

返回：

查询返回“C1”、“C2”、“C3”、“C4”、“R1”、“R2”、“R3”、“R4”。

示例：

:MEASure:ITEM1:SOURce1 C1 /*设置测量位 P1 的信源 1 为通道 1*/

:MEASure:ITEM1:SOURce1 ? /*查询返回 C1*/

:MEASure:ITEM<n>:THRold:TYPe

功能描述：设置或查询测量位参考阈值-寻顶方式，在特定测量项类型下才能设置成功。

指令：

:MEASure:ITEM<n>:THRold:TYPe <type>

:MEASure:ITEM<n>:THRold:TYPe ?

参数：

<n>= 离散型{1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 }。表示第 n 项测量参数。

<type>= 寻顶方式{BASetop | MINMax | ZERMin | ZERMax}

- BASetop:底值-顶值

- MINMax:最小-最大值
- ZERMin:零-最小值
- ZERMax:零-最大值

返回：

查询返回“BASetop”、“MINMax”、“ZERMin”、“ZERMax”。

示例：

:MEASure:ITEM1:THRold:TYPe BASetop /*设置测量位 P1 的参考阈值-寻顶方式为顶底值*/

:MEASure:ITEM1:THRold:TYPe ? /*查询返回 BASetop*/

:MEASure:ITEM<n>:THRold:TOP

功能描述：设置或查询测量项阈值高参考值，有效值 25%~95%，在特定测量项类型下才能设置成功。

指令：

:MEASure:ITEM<n>:THRold:TOP <top>

:MEASure:ITEM<n>:THRold:TOP ?

参数：

<n>= 离散型{1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 }。表示第 n 项测量参数。

<top>= 整型，高参考值

返回：

查询返回阈值。

示例：

:MEASure:ITEM1:THRold:TOP 50 /*设置测量位 P1 的参考阈值-顶的阈值为 50%*/

:MEASure:ITEM1:THRold:TOP ? /*查询返回 50*/

:MEASure:ITEM<n>:THRold:MIDDle

功能描述：设置或查询测量项阈值中参考值，有效值 15%~85%，在特定测量项类型下才能设置成功。

指令：

:MEASure:ITEM<n>:THRold:MIDDle <middle>

:MEASure:ITEM<n>:THRold:MIDDle ?

参数：

<n>= 离散型{1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 }。表示第 n 项测量参数。

<middle>= 整型，中参考值

返回：

查询返回阈值。

示例：

:MEASure:ITEM1:THRold:MIDDle 50 /*设置测量位 P1 的参考阈值-中的阈值为 50%*/

:MEASure:ITEM1:THRold:MIDDle ? /*查询返回 50*/

:MEASure:ITEM<n>:THRold:BOTTom

功能描述：设置或查询测量项阈值低参考值，有效值 5%~75%，在特定测量项类型下才能设置成功。

指令：

:MEASure:ITEM<n>:THRold:BOTTom <bottom>

:MEASure:ITEM<n>:THRold:BOTTom ?

参数：

<n>= 离散型{1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 }。表示第 n 项测量参数。

<bottom>= 整型，低参考值

返回：

查询返回阈值。

示例：

:MEASure:ITEM1:THRold:BOTTom 50 /*设置测量位 P1 的参考阈值-底的阈值为 50%*/

:MEASure:ITEM1:THRold:BOTTom ? /*查询返回 50*/

:MEASure:ITEM<n>:VALue

功能描述：查询测量项测量值。

指令：

:MEASure:ITEM<n>:VALue ?

参数：

<n>= 离散型{1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 }。表示第 n 项测量参数。

返回：

查询返回测量值。

:MEASure:ITEM<n>:COUNter:MAX

功能描述：查询测量项统计值-最大值。

指令：

:MEASure:ITEM<n>:COUNter:MAX ?

参数：

<n>= 离散型{1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 }。表示第 n 项测量参数。

返回：

查询返回测量项统计值的最大值。

:MEASure:ITEM<n>:COUNter:MIN

功能描述： 查询测量项统计值-最小值。

指令：

:MEASure:ITEM<n>:COUNter:MIN ?

参数：

<n>= 离散型{1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 }。表示第 n 项测量参数。

返回：

查询返回测量项统计值的最小值。

:MEASure:ITEM<n>:COUNter:AVG

功能描述： 查询测量项统计值-平均值。

指令：

:MEASure:ITEM<n>:COUNter:AVG ?

参数：

<n>= 离散型{1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 }。表示第 n 项测量参数。

返回：

查询返回测量项统计值的平均值。

:MEASure:ITEM<n>:COUNter:DEV

功能描述： 查询测量项统计值-标准差。

指令：

:MEASure:ITEM<n>:COUNter:DEV ?

参数：

<n>= 离散型{1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 }。表示第 n 项测量参数。

返回：

查询返回测量项统计值的标准差。

:MEASure:ITEM<n>:COUNter:POP

功能描述： 查询测量项统计-计数值。

指令：

:MEASure:ITEM<n>:COUNter:POP?

参数：

<n>= 离散型{1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 }。表示第 n 项测量参数。

返回：

查询返回测量项统计的计数值。

:MEASure:ITEM<n>:HISTgram

功能描述： 设置或查询直方图是否打开。

指令：

:MEASure:ITEM<n>:HISTgram <active>

:MEASure:ITEM<n>:HISTgram ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:MEASure:ITEM1:HISTgram ON /*设置直方图打开*/

:MEASure:ITEM1:HISTgram ? /*查询返回 1*/

:MEASure:ITEM<n>:TRENd

功能描述： 设置或查询趋势图是否打开。

指令：

:MEASure:ITEM<n>:TRENd <active>

:MEASure:ITEM<n>:TRENd ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:MEASure:ITEM1:TRENd ON /*设置趋势图打开*/

:MEASure:ITEM1:TRENd ? /*查询返回 1*/

:MEASure:ITEM<n>:TRACk

功能描述：设置或查询追踪是否打开。

指令：

:MEASure:ITEM<n>:TRACk <active>

:MEASure:ITEM<n>:TRACk ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:MEASure:ITEM1:TRACk ON /*设置追踪打开*/

:MEASure:ITEM1:TRACk ? /*查询返回 1*/

MATH 指令系统

用于设置或查询示波器数学通道。

:MATH<n>:ACTive

功能描述：设置或查询指定数学通道开启状态。

指令：

:MATH<n>:ACTive <active>

:MATH<n>:ACTive ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:MATH1:ACTive ON /*设置数学通道 1 打开*/

:MATH1:ACTive ? /*查询返回 1*/

:MATH<n>:LABel:ENABle

功能描述：设置或查询打开或关闭指定数学通道标签显示。

指令：

:MATH<n>:LABel:ENABle <enable>

:MATH<n>:LABel:ENABle ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:MATH1:LABel:ENABle ON /*设置数学通道 1 标签显示打开*/

:MATH1:LABel:ENABle ? /*查询返回 1*/

:MATH<n>:LABel

功能描述：设置或查询数学通道标签。

指令：

:MATH<n>:LABe <label>

:MATH<n>:LABe ?

参数：

<Label>= 通道名称字符串

返回：

查询返回通道名称字符串

示例：

:MATH1:LABel "abcd" /*设置数学通道 1 标签为 abcd*/

:MATH1:LABel ? /*查询返回 abcd*/

:MATH<n>:UNIT:ENABle

功能描述：设置或查询数学通道自定义单位开启状态。

指令：

:MATH<n>:UNIT:ENABle <enable>

:MATH<n>:UNIT:ENABle ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回"1"、"0"分别代表"ON"、"OFF"。

示例：

:MATH1:UNIT:ENABle ON /*设置数学通道 1 自定义单位打开*/

:MATH1:UNIT:ENABle ? /*查询返回 1*/

:MATH<n>:UNIT

功能描述：设置或查询指定数学通道显示单位名称，在自定义单位开启的时候才能设置成功。

指令：

:MATH<n>:UNIT <unit>

:MATH<n>:UNIT ?

参数：

<unit>= 单位字符串

返回：

查询返回指定数学通道的显示单位名称

示例：

:MATH1:UNIT A /*设置数学通道 1 单位为 A*/

:MATH1:UNIT ? /*查询返回 A*/

:MATH<n>:TYPe

功能描述：设置或查询数学通道数学运算类型。

指令：

:MATH<n>:TYPe <type>

:MATH<n>:TYPe ?

参数：

<type>= 离散型{BINary | FFT | FILTer | CUSTom | ADVanced}

- BINary:基本运算
- FFT:快速傅里叶变化
- FILTer:滤波
- CUSTom:自定义
- ADVanced:高级运算

返回：

查询返回“BINary”、“FFT”、“FILTer”、“CUStom”、“ADVanced”。

示例：

:MATH1:TYPe FFT /*设置数学通道 1 数学运算类型为 FFT*/

:MATH1:TYPe ? /*查询返回 FFT*/

:MATH<n>:VERTical:OFFSet

功能描述：设置或查询数学通道垂直位移。

指令：

:MATH<n>:VERTical:OFFSet <offset>

:MATH<n>:VERTical:OFFSet ?

参数：

<offset>= 实型，单位 div，范围为-4div-4div。

返回：

查询返回数学通道垂直位移。

示例：

:MATH1:VERTical:OFFSet 1 /*设置数学通道 1 垂直位移 1div*/

:MATH1:VERTical:OFFSet ? /*查询返回 1.00000E+000*/

:MATH<n>:VERTical:SCALe

功能描述：设置或查询数学通道垂直档位。

指令：

:MATH<n>:VERTical:SCALe <scale>

:MATH<n>:VERTical:SCALe ?

参数：

<scale>= 实型，单位:自定义单位，不输入单位则默认单位为 V。

返回：

查询返回数学通道垂直档位。

示例：

:MATH1:VERTical:SCALe 1 /*设置数学通道 1 垂直档位 1V*/

:MATH1:VERTical:SCALe ? /*查询返回 1.00000E+000*/

:MATH<n>:HORizontal:OFFSet

功能描述：设置或查询数学通道水平位移。

指令：

:MATH<n>:HORizontal:OFFSet <offset>

:MATH<n>:HORizontal:OFFSet ?

参数：

<offset>= 实型，单位 div，范围为 0div-10div。

返回：

查询返回数学通道水平位移。

示例：

:MATH1:HORizontal:OFFSet 1 /*设置数学通道 1 水平位移 1div*/

:MATH1:HORizontal:OFFSet ? /*查询返回 1.00000E+000*/

:MATH<n>:HORizontal:SCALe

功能描述：设置或查询数学通道水平档位。

指令：

:MATH<n>:HORizontal:SCALe <scale>

:MATH<n>:HORizontal:SCALe ?

参数：

<scale>= 实型，单位:默认单位为 s，范围:与主时基大小相关。

返回：

查询返回数学通道水平档位。

示例：

:MATH1:HORizontal:SCALe 1us /*设置数学通道 1 水平档位 1us*/

:MATH1:HORizontal:SCALe ? /*查询返回 1.0000000000000E-006*/

:MATH<n>:SOURce<ch>

功能描述：设置或查询数学通道运算源。

指令：

:MATH<n>:SOURce<ch> <source>

:MATH<n>:SOURce<ch> <source> ?

参数：

<ch>= {1 | 2},表示信源 1 或者信源 2

<source>=离散型{C1 | C2 | C3 | C4 | R1 | R2 | R3 | R4}

返回：

查询返回“C1”、“C2”、“C3”、“C4”、“R1”、“R2”、“R3”、“R4”。

示例：

:MATH1:SOURce1 C1 /*设置数学通道 1 运算源为通道 1*/

:MATH1:SOURce1 ? /*查询返回 C1*/

说明：若只有一个源可以写 source1，也可以直接写 source

基本运算

:MATH<n>:OPERation

功能描述：设置或查询数学通道基础运算类型。

指令：

:MATH<n>:OPERation <type>

:MATH<n>:OPERation ?

参数：

<type>= 离散型{ADD | SUBTract | MULTiply | DIVide}

- ADD:加
- SUBTract:减
- MULTiply:乘
- DIVide:除

返回：

查询返回“ADD”、“SUBTract”、“MULTiply”、“DIVide”。

示例：

:MATH1:OPERation ADD /*设置数学通道 1 基础运算类型为加*/

:MATH1:OPERation ? /*查询返回 ADD*/

FFT

:MATH<n>:FFT:WINDow

功能描述：运算类型为快速傅里叶变换（FFT）时，设置或查询数学通道窗函数。

指令：

:MATH<n>:FFT:WINDow <type>

:MATH<n>:FFT:WINDow ?

参数：

<type>= 离散型{RECTangle | HANN | HAMMing | BLACkman | FLATtop}

- RECTangle:矩形窗
- HANN:汉宁窗
- HAMMing:汉明窗
- BLACkman:布莱克曼窗
- FLATtop:平顶窗

返回：

查询返回“RECTangle”、“HANN”、“HAMMing”、“BLACkman”、“FLATtop”。

示例：

:MATH1:FFT:WINDow RECTangle /*设置数学通道 1 窗函数为矩形窗*/

:MATH1:FFT:WINDow ? /*查询返回 RECTangle*/

:MATH<n>:FFT:TYPe

功能描述：运算类型为快速傅里叶变换（FFT）时，设置或查询数学通道输出类型。

指令：

:MATH<n>:FFT:TYPe <type>

:MATH<n>:FFT:TYPe ?

参数：

<type>= 离散型{AMPLtd | POWer | PSD | REAL | IMAGinary | PHASe}

- AMPLtd :幅度谱
- POWer:功率谱
- PSD:Psd
- REAL:实部
- IMAGinary 虚部
- PHASe:相位谱

返回：

查询返回“AMPLtd”、“POWer”、“PSD”、“REAL”、“IMAGinary”、“PHASe”。

示例：

:MATH1:FFT:TYPe AMPLtd /*设置数学通道 1 输出类型为幅度谱*/

:MATH1:FFT:TYPe ? /*查询返回 AMPLtd*/

:MATH<n>:FFT:COOR:UNIT

功能描述：快速傅里叶变换（FFT）输出类型为幅度谱时，设置或查询数学通道 FFT（幅度谱）单位。

指令：

:MATH<n>:FFT:COOR:UNIT <type>

:MATH<n>:FFT:COOR:UNIT ?

参数：

<type>= 离散型{VRMS | DBM | DBUW | DBMV | DBUV | DBMA | DBUA}

返回：

查询返回“VRMS”、“DBM”、“DBUW”、“DBMV”、“DBUV”、“DBMA”、“DBUA”。

示例：

:MATH1:FFT:COOR:UNIT VRMS /*设置数学通道 1 的 FFT 单位为 VRMS*/

:MATH1:FFT:COOR:UNIT ? /*查询返回 VRMS*/

:MATH<n>:FFT:PSD:UNIT

功能描述：快速傅里叶变换（FFT）输出类型为 PSD 时，设置或查询数学通道 FFT（PSD）单位。

指令：

:MATH<n>:FFT:PSD:UNIT <type>

:MATH<n>:FFT:PSD:UNIT ?

参数：

<type>= 离散型{V2 | DBM | DBUW | DBMV | DBUV | DBMA | DBUA}

返回：

查询返回“V2”、“DBM”、“DBUW”、“DBMV”、“DBUV”、“DBMA”、“DBUA”。

示例：

:MATH1:FFT:PSD:UNIT DBM /*设置数学通道 1 的 FFT 单位为 DBM*/

:MATH1:FFT:PSD:UNIT ? /*查询返回 DBM*/

:MATH<n>:FFT:POINt

功能描述：运算类型为快速傅里叶变换（FFT）时，设置或查询数学通道扫描点数。

指令：

:MATH<n>:FFT:POINt <type>

:MATH<n>:FFT:POINt ?

参数：

<type>= 离散型{1k | 2k | 4k | 8k | 16k | 32k | 64k | 128k | 256k | 512k | 1M | 2M | 4M | 8M | 16M | 32M}，默认单位 Pts。

返回：

查询返回“1k”、“2k”、“4k” 、“8k”、“16k”、“32k”、“64k”，“128k”、“256k”、“512k”、“1M”、“2M”、“4M”、“8M”、“16M”、“32M”。

示例：

:MATH1:FFT:POINt 1k /*设置数学通道 1 扫描点数为 1KPts*/

:MATH1:FFT:POINt ? /*查询返回 1K*/

:MATH<n>:FFT:FREQuency:CENTer

功能描述：运算类型为快速傅里叶变换(FFT)时，设置或查询指定 FFT 的中心频率，默认单位 Hz。

指令：

:MATH<n>:FFT:FREQuency:CENTer <freq>

:MATH<n>:FFT:FREQuency:CENTer ?

参数：

<freq>= 实型，单位 mHz、Hz、kHz、MHz

返回：

查询返回中心频率，采用科学计数法表示。

示例：

:MATH1:FFT:FREQuency:CENTer 100 /*设置数学通道 1FFT 中心频率 100Hz*/

:MATH1:FFT:FREQuency:CENTer ? /*查询返回 1.0000000000000E+002*/

:MATH<n>:FFT:FREQuency:SPAN

功能描述：运算类型为快速傅里叶变换(FFT)时，设置或查询指定 FFT 的跨度，默认单位 Hz。

指令：

:MATH<n>:FFT:FREQuency:SPAN

:MATH<n>:FFT:FREQuency:SPAN ?

参数：

= 实型，单位 Hz、kHz、MHz

返回：

查询返回频率跨度，采用科学计数法表示。

示例：

:MATH1:FFT:FREQuency:SPAN 100000 /*设置数学通道 1FFT 频率跨度 100kHz*/

:MATH1:FFT:FREQuency:SPAN ? /*查询返回 1.0000000000000E+005*/

说明:频率跨度实际可设置范围与当前时基刻度有关。

:MATH<n>:FFT:MARK:RDMode

功能描述：设置或查询 FFT 标记功能中的读数。

指令：

:MATH<n>:FFT:MARK:RDMode <mode>

:MATH<n>:FFT:MARK:RDMode ?

参数：

<mode>= 离散型{ABSolute | DELTa}

- ABSolute:绝对值
- DELTa:增量

返回：

查询返回“ABSolute”、“DELTa”。

示例：

:MATH1:FFT:MARK:RDMode ABSolute /*设置数学通道 1FFT 读数为 ABSolute*/

:MATH1:FFT:MARK:RDMode ? /*查询返回 ABSolute*/

:MATH<n>:FFT:MARK:AUTO

功能描述：设置或查询 FFT 标记功能中自动标记的开启状态。

指令：

:MATH<n>:FFT:MARK:AUTO <enable>

:MATH<n>:FFT:MARK:AUTO ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:MATH1:FFT:MARK:AUTO ON /*设置数学通道 1 的 FFT 自动标记打开*/

:MATH1:FFT:MARK:AUTO ? /*查询返回 1*/

:MATH<n>:FFT:MARK:STMode

功能描述：设置或查询 FFT 标记的排序模式。

指令：

:MATH<n>:FFT:MARK:STMode <mode>

:MATH<n>:FFT:MARK:STMode ?

参数：

<mode>= 离散型{FREQuency | AMPLitude}

- FREQuency:频率

- AMPLitude:幅度

返回：

查询返回 FFT 标记的排序模式

示例：

:MATH1:FFT:MARK:STMode AMPLitude /*设置 FFT 标记的排序模式为幅度*/

:MATH1:FFT:MARK:STMode ? /*查询返回 AMPLitude*/

:MATH<n>:FFT:MARK:RESult

功能描述：设置或查询 FFT 标记功能中结果表的开启状态。

指令：

:MATH<n>:FFT:MARK:RESult <enable>

:MATH<n>:FFT:MARK:RESult ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:MATH1:FFT:MARK:RESult ON /*设置数学通道 1FFT 标记中结果表打开*/

:MATH1:FFT:MARK:RESult ? /*查询返回 1*/

:MATH<n>:FFT:MARK:COUNt

功能描述：设置或查询 FFT 标记功能中的标记数量。

指令：

:MATH<n>:FFT:MARK:COUNt <count>

:MATH<n>:FFT:MARK:COUNt ?

参数：

<count>= 整数型，范围 1~11。

返回：

查询返回 FFT 标记功能中的标记数量的整数。

示例：

:MATH1:FFT:MARK:COUNt 11 /*设置数学通道 1FFT 标记功能中标记数量为 11 次*/

:MATH1:FFT:MARK:COUNt ? /*查询返回 11*/

:MATH<n>:FFT:MARK:THReshold

功能描述：设置或查询 FFT 标记功能中的阈值,阈值范围-200~100。

指令：

:MATH<n>:FFT:MARK:THReshold <value>

:MATH<n>:FFT:MARK:THReshold ?

参数：

<value>= 整型，范围-200~100

返回：

查询返回 FFT 标记功能中的阈值，采用科学计数法。

示例：

:MATH1:FFT:MARK:THReshold 20 /*设置数学通道 1FFT 的标记阈值为 20*/

:MATH1:FFT:MARK:THReshold ? /*查询返回 2.00000E001*/

:MATH<n>:FFT:MARK:EXCursion

功能描述：设置或查询 FFT 标记功能中的振幅,振幅范围 0~299。

指令：

:MATH<n>:FFT:MARK:EXCursion <value>

:MATH<n>:FFT:MARK:EXCursion ?

参数：

<value>= 整型，范围 0~299

返回：

查询返回 FFT 标记功能中的振幅，采用科学计数法。

示例：

:MATH1:FFT:MARK:EXCursion 20 /*设置数学通道 1FFT 的标记振幅为 20*/

:MATH1:FFT:MARK:EXCursion ? /*查询返回 2.00000E001*/

:MATH<n>:FFT:MARK:FREQTOCENTer

功能描述：设置 FFT 标记功能以“峰值设为中心”。

指令：

:MATH<n>:FFT:MARK:FREQTOCENTer

:MATH<n>:FFT:DATA

功能描述：运算类型为快速傅里叶变换（FFT）并在标记结果表打开情况下，查询指定 FFT 的标记结果表结果。

指令：

:MATH<n>:FFT:DATA ?

参数：

<n>= 整型，范围 1~8

返回：

查询返回指定 FFT 的标记结果表所有结果值，符合数据块格式，以科学计数法表示，并按顺序以逗号按分隔排列，无效值使用实型数据最大值表示。

例如：“9000000350Freq(Hz),Value(dBm)

991.821289 Hz,-12.93852 dBm

11.367798 kHz,-66.563634 dBm

15.830994 kHz,-69.902813 dBm

26.435852 kHz,-73.254517 dBm

30.021667 kHz,-70.721245 dBm

30.632019 kHz,-74.100575 dBm

34.942627 kHz,-74.549815 dBm

35.934448 kHz,-77.051566 dBm

39.024353 kHz,-74.300219 dBm

40.054321 kHz,-75.016271 dBm

43.029785 kHz,-74.406271 dBm”

自定义

:MATH<n>:EXPRession

功能描述：利用自由组合表达式进行数学计算。

表达式格式可见示波器下 MATH 菜单下自定义类型公式编辑窗，<expression>属于 ASCII 字符串参数。

指令：

:MATH<n>:EXPRession <expression>

:MATH<n>:EXPRession ?

参数：

<expression>= 自定义运算字符串。

自定义运算仅支持模拟通道和参考通道。

示例：

*:MATH1:EXPRession "C1 * C2"* /*表示将通道一和二相乘*/

:MATH1:EXPRession ? /*查询返回 C1 * C2*/

说明：公式中所含函数表达式均需小写（通道字母固定大写）

滤波

:MATH<n>:FILTer:TYPe

功能描述：运算类型为 Filter 时，设置或查询滤波类型。

指令：

:MATH<n>:FILTer:TYPe <type>

:MATH<n>:FILTer:TYPe ?

参数：

<type>= 离散型{LOWPass | HIGHpass | BNDPass | BNDStop}

- LOWPass:低通
- HIGHpass:高通
- BNDPass:带通
- BNDStop:带阻

返回：

查询返回"LOWPass"、"HIGHpass"、"BNDPass"、"BNDStop"。

示例：

:MATH1:FILTer:TYPe BNDStop /*设置数学通道 1 的滤波类型为带阻*/

:MATH1:FILTer:TYPe ? /*查询返回 BNDStop*/

:MATH<n>:FILTer:FREQuency<n>:STOP

功能描述：运算类型为 Filter 时，设置或查询截止频率，默认单位 Hz。

指令：

:MATH<n>:FILTer:FREQuency<n>:STOP <freq>

:MATH<n>:FILTer:FREQuency<n>:STOP ?

参数：

<n>= 离散型{1 | 2}，表示截止频率 1 和截至频率 2

<freq>= 实型，单位 mHz、Hz、kHz、MHz

返回：

查询返回截至频率，采用科学计数法表示。

示例：

:MATH1:FILTer:FREQuency1:STOP 100 /*设置数学通道 1 的滤波截止频率 1 为 100Hz*/

:MATH1:FILTer:FREQuency1:STOP ? /*查询返回 1.0000000000000E+002*/

:MATH<n>:FILTer:CUSTom:ENABle

功能描述：运算类型为 Filter 时，设置或查询自定义滤波器的打开状态。

指令：

:MATH<n>:FILTer:CUSTom:ENABle <enable>

:MATH<n>:FILTer:CUSTom:ENABle ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:MATH1:FILTer:CUSTom:ENABle ON /*设置数学通道 1 自定义滤波器打开*/

:MATH1:FILTer:CUSTom:ENABle ? /*查询返回 1*/

CYMometer 指令系统

用于设置示波器频率计开关及查询测量值。

:CYMometer:ACTive

功能描述：设置或查询频率计开启状态。

指令：

:CYMometer:ACTive <active>

:CYMometer:ACTive ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:CYMometer:ACTive ON /*设置频率计打开*/

:CYMometer:ACTive ? /*查询返回 1*/

:CYMometer:HISTgram

功能描述：设置或查询频率计直方图是否打开。

指令：

:CYMometer:HISTgram <enable>

:CYMometer:HISTgram ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

- ON:打开
- OFF:关闭

返回：

查询返回 1 或 0，分别代表 ON 或 OFF。

示例：

:CYMometer:HISTgram ON /*设置打开频率计趋势图*/

:CYMometer:HISTgram ? /*查询返回 1*/

:CYMometer:TRENd

功能描述：设置或查询频率计趋势图是否打开。

指令：

:CYMometer:TRENd <enable>

:CYMometer:TRENd ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

- ON:打开
- OFF:关闭

返回：

查询返回 1 或 0，分别代表 ON 或 OFF。

示例：

:CYMometer:TRENd ON /*设置频率计趋势图打开*/

:CYMometer:TRENd ? /*查询返回 1*/

:CYMometer:FREQuency

功能描述：查询返回频率计的值。

指令：

:CYMometer:FREQuency ?

返回：

查询返回频率计测量值，以 Hz 为单位。

DVM 指令系统

用于设置示波器电压表参数及查询测量值。

:DVM:ACTive

功能描述：设置或查询电压表开启状态。

指令：

:DVM:ACTive <active>

:DVM:ACTive ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:DVM:ACTive ON /*设置电压表打开*/

:DVM:ACTive ? /*查询返回 1*/

:DVM:SOURce

功能描述：设置或查询电压表源。

指令：

:DVM:SOURce <source>

:DVM:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回“C1”、“C2”、“C3”、“C4”。

示例：

:DVM:SOURce C1 /*设置电压表源为 C1*/

:DVM:SOURce ? /*查询返回 C1*/

:DVM:HISTgram

功能描述：设置或查询电压表直方图是否打开。

指令：

:DVM:HISTgram <enable>

:DVM:HISTgram ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

- ON:打开
- OFF:关闭

返回：

查询返回 1 或 0，分别代表 ON 或 OFF。

示例：

:DVM:HISTgram ON /*设置打开频率计电压表*/

:DVM:HISTgram ? /*查询返回 1*/

:DVM:TRENd

功能描述：设置或查询电压表趋势图是否打开。

指令：

:DVM:TRENd <enable>

:DVM:TRENd ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

- ON:打开
- OFF:关闭

返回：

查询返回 1 或 0，分别代表 ON 或 OFF。

示例：

:DVM:TRENd ON /*设置打开电压表趋势图*/

:DVM:TRENd ? /*查询返回 1*/

:DVM:MODe

功能描述：设置或查询电压表模式。

指令：

:DVM:MODe <mode>

:DVM:MODe ?

参数：

<mode>= 离散型{DC | ACRMs | DCACrms}

返回：

查询返回“DC”、“ACRMs”、“DCACrms”。

示例：

:DVM:MODe DC /*设置电压表模式为 DC*/

:DVM:MODe ? /*查询返回 DC*/

:DVM:AUTO

功能描述：设置或查询电压表自动量程开启状态。

指令：

:DVM:AUTO <active>

:DVM:AUTO ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:DVM:AUTO ON /*设置电压表自动量程打开*/

:DVM:AUTO ? /*查询返回 1*/

:DVM:CURRent

功能描述：查询返回电压表的值。

指令：

:DVM:CURRent ?

返回：

查询返回电压表测量值，以 V 为单位。

DISPlay 指令系统

用于设置或查询示波器屏幕显示相关参数。

:DISPlay:PERSist

功能描述：设置或查询无限余辉状态。

指令：

:DISPlay:PERSist <persist>

:DISPlay:PERSist ?

参数：

<persist>= 离散型{CLOSe | AUTO | INFinity}

- CLOSe:关闭
- AUTO:自动
- INFinity:无限

返回：

查询返回无限余辉状态“CLOSe”、“AUTO”、“INFinity”。

示例：

:DISPlay:PERSist CLOSe /*设置无限余辉关闭*/

:DISPlay:PERSist ? /*查询返回 CLOSe */

:DISPlay:MARK

功能描述：设置或查询显示标记。

指令：

:DISPlay:MARK <mark>

:DISPlay:MARK?

参数：

<mark>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:DISPlay:MARK ON /*设置显示标记打开*/

:DISPlay:MARK ? /*查询返回 1 */

:DISPlay:MKVert

功能描述：设置或查询垂直标记位置。

指令：

:DISPlay:MKVert <mkvert>

:DISPlay:MKVert ?

参数：

<mkvert>= 离散型{LEFT | RIGHt}

- LEFT:左
- RIGHt:右

返回：

查询返回垂直标记位置“LEFT”、“RIGHt”。

示例：

:DISPlay:MKVert LEFT /*设置垂直标记位置为左*/

:DISPlay:MKVert ? /*查询返回 LEFT */

:DISPlay:MKHori

功能描述：设置或查询水平标记位置。

指令：

:DISPlay:MKHori <mkHori>

:DISPlay:MKHori ?

参数：

<mkHori>= 离散型{TOP | BOTTom}

- TOP:上
- BOTTom:下

返回：

查询返回水平标记位置“TOP”、“BOTTom”。

示例：

:DISPlay:MKHori TOP /*设置水平标记位置为上*/

:DISPlay:MKHori ? /*查询返回 TOP */

:DISPlay:GRID:STYLe

功能描述：设置或查询网格样式。

指令：

:DISPlay:GRID:STYLe <style>

:DISPlay:GRID:STYLe ?

参数：

<style>= 离散型{FULL | BRIef | NONe}

- FULL:满
- BRIef:简单
- NONe:无

返回：

查询返回网格样式“FULL”、“BRIef”、“NONe”。

示例：

:DISPlay:GRID:STYLe FULL /*设置网格样式为满*/

:DISPlay:GRID:STYLe ? /*查询返回 FULL*/

:DISPlay:GRID:BRIGtness

功能描述：设置或查询网格亮度。

指令：

:DISPlay:GRID:BRIGtness <value>

:DISPlay:GRID:BRIGtness ?

参数：

<value>= 整数型，范围：0~100。

返回：

查询返回网格亮度值。

示例：

:DISPlay:GRID:BRIGtness 50 /*设置网格亮度值 50*/

:DISPlay:GRID:BRIGtness ? /*查询返回 50*/

:DISPlay:WAVe:STYLe

功能描述：设置或查询波形样式。

指令：

:DISPlay:WAVe:STYLe <style>

:DISPlay:WAVe:STYLe ?

参数：

<style>= 离散型{VECTor | DOT}

- VECTor:矢量
- DOT:点

返回：

查询返回波形样式“VECTor”、“DOT”。

示例：

:DISPlay:WAVe:STYLe VECTor /*设置波形样式为矢量*/

:DISPlay:WAVe:STYLe ? /*查询返回 VECTor*/

:DISPlay:WAVe:BRIGtness

功能描述：设置或查询波形亮度。

指令：

:DISPlay:WAVe:BRIGtness <value>

:DISPlay:WAVe:BRIGtness ?

参数：

<value>= 整数型，范围：0~100。

返回：

查询返回波形亮度值。

示例：

:DISPlay:WAVe:BRIGtness 50 /*设置波形亮度值 50*/

:DISPlay:WAVe:BRIGtness ? /*查询返回 50*/

:DISPlay:SCReen:BRIGtness

功能描述：设置或查询屏幕亮度。

指令：

:DISPlay:SCReen:BRIGtness <value>

:DISPlay:SCReen:BRIGtness ?

参数：

<value>= 整数型，范围：5~100。

返回：

查询返回屏幕亮度值。

示例：

:DISPlay:SCReen:BRIGtness 50 /*设置屏幕亮度值 50*/

:DISPlay:SCReen:BRIGtness ? /*查询返回 50*/

:DISPlay:SCReen:CONTrast

功能描述：设置或查询屏幕对比度。

指令：

:DISPlay:SCReen:CONTrast <value>

:DISPlay:SCReen:CONTrast ?

参数：

<value>= 整数型，范围：50~100。

返回：

查询返回屏幕对比度值。

示例：

:DISPlay:SCReen:CONTrast 50 /*设置屏幕对比度值 50*/

:DISPlay:SCReen:CONTrast ? /*查询返回 50*/

FILe 指令系统

用于设置和查询示波器波形、截屏相关参数

波形

:FILe:WAVe:FORMat

功能描述：设置或查询波形保存格式。

指令：

:FILe:WAVe:FORMat <format>

:FILe:WAVe:FORMat ?

参数：

<format>= 离散型{BINary | TEXT | MATlab | EXCel | CSV | TSV | DAT}

返回：

查询返回波形保存格式“BINary”、“TEXT”、“MATlab”、“EXCel”、“CSV”、“TSV”、“DAT”。

示例：

:FILe:WAVe:FORMat TEXT /*设置波形保存格式 TEXT*/

:FILe:WAVe:FORMat ? /*查询返回 TEXT*/

:FILe:WAVe:TXTEncoding

功能描述：设置或查询当波形保存格式为文本时的文本编码格式。

指令：

:FILe:WAVe:TXTEncoding <encoding>

:FILe:WAVe:TXTEncoding ?

参数：

<encoding>= 离散型{ASCII | GB2312 | UTF8 | UTF32 | UNICode}

返回：

查询返回文本编码格式“ASCII”、“GB2312”、“UTF8”、“UTF32”、“UNICode”。

示例：

:FILe:WAVe:TXTEncoding ASCII /*设置文本编码格式 ASCII*/

:FILe:WAVe:TXTEncoding ? /*查询返回 ASCII*/

:FILe:WAVe:SOURce

功能描述：设置或查询波形保存的源。

指令：

:FILe:WAVe:SOURce <source>

:FILe:WAVe:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回波形保存源“C1”、“C2”、“C3”、“C4”。

示例：

:FILe:WAVe:SOURce C1 /*设置波形保存源 C1*/

:FILe:WAVe:SOURce ? /*查询返回 C1*/

:FILe:WAVe:GATe

功能描述：设置或查询波形保存的选通类型。

指令：

:FILe:WAVe:GATe <value>

:FILe:WAVe:GATe ?

参数：

<<value>= 离散型{SCReen | CURSor}

- SCReen:屏幕
- CURSor:光标

返回：

查询返回波形保存的选通类型。

示例：

:FILe:WAVe:GATe SCReen /*设置波形保存的选通类型为屏幕*/

:FILe:WAVe:GATe ? /*查询返回 SCReen*/

:FILe:WAVe:NAMe:FORMat

功能描述：设置或查询波形保存名称后缀。

指令：

:FILe:WAVe:NAMe:FORMat <value>

:FILe:WAVe:DATetime:ENABle ?

参数：

<value>= 离散型{ACCumulation | TIMe | DISabled}

- ACCumulation:累加
- TIMe:时间
- DISabled:禁用

返回：

查询返回名称后缀。

示例：

:FILe:WAVe:NAMe:FORMat TIMe /*设置波形保存名称后缀为时间*/

:FILe:WAVe:NAMe:FORMat ? /*查询返回 TIMe*/

:FILe:WAVe:NAMe

功能描述：设置或查询波形保存名称。

指令：

:FILe:WAVe:NAMe <name>

:FILe:WAVe:NAMe ?

参数：

<name>= 可自定义输入保存波形名称。若不输入名称则默认名称自动叠加序号。

返回：

查询返回波形名称。

示例：

:FILe:WAVe:NAMe "abcd" /*设置波形保存名称 abcd*/

:FILe:WAVe:NAMe ? /*查询返回 abcd */

:FILe:WAVe:PATH

功能描述：设置或查询波形保存路径。

指令：

:FILe:WAVe:PATH <path>

:FILe:WAVe:PATH ?

参数：

<path>= 保存的文件路径

返回：

查询返回波形路径。

示例：

:FILe:WAVe:PATH "D:\\WaveForm" /*设置波形保存路径 D:\\WaveForm*/

:FILe:WAVe:PATH ? /*查询返回 D:\\WaveForm*/

:FILe:WAVe:SAVe

功能描述：保存波形数据并返回是否保存成功。

指令：

:FILe:WAVe:SAVe ?

返回：

查询返回离散型{SUCCess | FAIL}。

- SUCCess:保存成功
- FAIL: 保存失败

截屏

:FILe:PICTure:REGion

功能描述：设置或查询图片保存区域。

指令：

:FILe:PICTure:REGion <region>

:FILe:PICTure:REGion ?

参数：

<region>= 离散型{WINDow | APPLication}。

- WINDow:栅格
- APPLication:屏幕

返回：

查询返回“WINDow”、“APPLication”。

示例：

:FILe:PICTure:REGion WINDow /*设置图片保存栅格区域*/

:FILe:PICTure:REGion ? /*查询返回 WINDow*/

:FILe:PICTure:COLor

功能描述：设置或查询图片保存颜色。

指令：

:FILe:PICTure:COLor <color>

:FILe:PICTure:COLor ?

参数：

<color>= 离散型{STANdard | BLACkwhite | REVerse}。

- STANdard:标准
- BLACkwhite:黑白
- REVerse:反色

返回：

查询返回“STANdard ”、“BLACkwhite”、“REVerse”。

示例：

:FILe:PICTure:COLor STANdard /*设置图片保存颜色为标准*/

:FILe:PICTure:COLor ? /*查询返回 STANdard */

:FILe:PICTure:FORMat

功能描述：设置或查询图片保存格式。

指令：

:FILe:PICTure:FORMat <format>

:FILe:PICTure:FORMat ?

参数：

<format>= 离散型{BMP | TIFF | GIF | PNG | JPEG}。

返回：

查询返回图片的保存格式。

示例：

:FILe:PICTure:FORMat JPEG /*设置图片保存格式为 JPEG*/

:FILe:PICTure:FORMat ? /*查询返回 JPEG*/

:FILe:PICTure:NAMe

功能描述：设置或查询图片保存名称。

指令：

:FILe:PICTure:NAMe <name>

:FILe:PICTure:NAMe ?

参数：

<name>= 字符串类型，可自定义输入保存波形名称。若不输入名称则默认名称自动叠加序号。

示例：

:FILe:PICTure:NAMe "abcd" /*设置图片保存名称 abcd*/

:FILe:PICTure:NAMe ? /*查询返回 abcd*/

:FILe:PICTure:NAMe:FORMat

功能描述：设置或查询截图名称后缀。

指令：

:FILe:PICTure:NAMe:FORMat <value>

:FILe:PICTure:NAMe:FORMat ?

参数：

<value>= 离散型{ACCumulation | TIMe | DISabled}

- ACCumulation:累加
- TIMe:时间
- DISabled:禁用

返回：

查询名称后缀。

示例：

:FILe:PICTure:NAMe:FORMat TIMe /*设置截图保存名称后缀为时间*/

:FILe:PICTure:NAMe:FORMat ? /*查询返回 TIMe*/

:FILe:PICTure:PATH

功能描述：设置或查询截图保存路径。

指令：

:FILe:PICTure:PATH <path>

:FILe:PICTure:PATH ?

参数：

<path>= 文档保存路径

返回：

查询返回截图路径。

示例：

:FILe:PICTure:PATH "D:\\Picture" /*设置截图保存路径 D:\\Picture*/

:FILe:PICTure:PATH ? /*查询返回 D:\\Picture*/

:FILe:PICTure:SAVe

功能描述：保存图片数据并返回是否保存成功。

指令：

:FILe:PICTure:SAVe ?

返回：

查询返回离散型{SUCCess | FAIL}。

- SUCCess:保存成功
- FAIL:保存失败

设置

:FILe:SETTing:NAMe

功能描述：设置或查询系统设置保存名称。

指令：

:FILe:SETTing:NAMe <name>

:FILe:SETTing:NAMe ?

参数：

<name>= 离散型

返回：

查询返回系统设置保存名称。

示例：

:FILe:SETTing:NAMe "abcd" /*设置系统设置保存名称为 abcd*/

:FILe:SETTing:NAMe ? /*查询返回 abcd*/

:FILe:SETTing:NAMe:FORMat

功能描述：设置或查询系统设置名称后缀。

指令：

:FILe:SETTing:NAMe:FORMat <value>

:FILe:SETTing:NAMe:FORMat ?

参数：

<value>= 离散型{ACCumulation | TIMe | DISabled }

- ACCumulation:累加
- TIMe:时间
- DISabled:禁用

返回：

查询系统设置名称后缀。

示例：

:FILe:SETTing:NAMe:FORMat TIMe /*设置系统设置保存名称后缀为时间*/

:FILe:SETTing:NAMe:FORMat ? /*查询返回 TIMe*/

:FILe:SETTing:PATH

功能描述：设置或查询系统设置的保存路径。

指令：

:FILe:SETTing:PATH <value>

:FILe:SETTing:PATH ?

参数：

<value>= 字符串类型

返回：

查询系统设置保存路径。

示例：

:FILe:SETTing:PATH "C:\\Users\\lenovo\\Documents\\U2 DSO\\Setting" /*设置系统设置保存路径为 C:\\Users\\lenovo\\Documents\\U2 DSO\\Setting*/

:FILe:SETTing:PATH ? /*查询返回 C:\\Users\\lenovo\\Documents\\U2 DSO\\Setting*/

:FILe:SETTing:SAVe

功能描述：保存系统设置并返回是否保存成功。

指令：

:FILe:SETTing:SAVe ?

返回：

查询返回 SUCCess:保存成功，FAIL:保存失败

示例：

:FILe:SETTing:SAVe ? /*保存系统设置并返回是否保存成功*/

:FILe:SETTing:READFULLPATH

功能描述：加载指定的系统设置文件。

指令：

:FILe:SETTing:READFULLPATH <path>

参数：

<path>= 字符串类型，设置文件的全路径

示例：

:FILe:SETTing:READFULLPATH "C:\\Users\\lenovo\\Documents\\U2 DSO\\Setting\\U2Factory.set"

/*读取 C:\\Users\\lenovo\\Documents\\U2 DSO\\Setting\\U2Factory.set 设置文件*/

PRINt 指令系统

用于设置和查询示波器打印系统相关参数

:PRINt:REGion

功能描述：设置或查询打印机打印区域。

指令：

:PRINt:REGion <region>

:PRINt:REGion ?

参数：

<region>= 离散型{SCReen | GRID}

- SCReen:屏幕
- GRID:栅格

返回：

查询返回“SCReen”、“GRID”。

示例：

:PRINt:REGion SCReen /*设置打印区域为屏幕*/

:PRINt:REGion ? /*查询返回 SCReen*/

:PRINt:COLor

功能描述：设置或查询打印机打印颜色。

指令：

:PRINt:COLor <color>

:PRINt:COLor ?

参数：

<color>= 离散型{STANdard | BLACkwhite | REVerse}。

- STANdard:标准
- BLACkwhite:黑白
- REVerse:反色

返回：

查询返回“STANdard ”、“BLACkwhite”、“REVerse”。

示例：

:PRINt:COLor STANdard /*设置打印颜色为“标准”*/

:PRINt:COLor ? /*查询返回 STANdard*/

:PRINt:ORIent

功能描述：设置或查询打印机打印方向。

指令：

:PRINt:ORIent <orient>

:PRINt:ORIent ?

参数：

<orient>= 离散型{HOR | VER}。

- HOR:横向
- VER:纵向

返回：

查询返回“HOR ”、“VER”。

示例：

:PRINt:ORIent HOR /*设置打印方向为横向*/

:PRINt:ORIent ? /*查询返回 HOR */

LA 指令系统

用于设置或查询示波器 LA 逻辑通道相关信息。

:LA:ITEM<n>:DISPlay

功能描述：设置或查询指定逻辑通道打开状态。

指令：

:LA:ITEM<n>:DISPlay <active>

:LA:ITEM<n>:DISPlay ?

参数：

<n>= 整数型，设置范围 0~15，表示逻辑通道 D0~D15。

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:LA:ITEM1:DISPlay ON /* 设置逻辑通道 1 打开*/

:LA:ITEM1:DISPlay ? /*查询返回 1*/

:LA:ITEM<n>:OFFSet

功能描述：设置或查询指定逻辑通道的垂直位置偏移。

指令：

:LA:ITEM<n>:OFFSet <offset>

:LA:ITEM<n>:OFFSet?

参数：

<n>= 整数型，设置范围 0~15，表示逻辑通道 D0~D15。

<offset>= 实型，设置范围-3.5div~4div。

返回：

查询返回指定逻辑通道的垂直位置偏移，以科学计数法表示。

示例：

:LA:ITEM1:OFFSet 3 /*设置逻辑通道 1 的垂直位置偏移 3div*/

:LA:ITEM1:OFFSet ? /*查询返回 3.00000E+000*/

:LA:ITEM<n>:LABel

功能描述：设置或查询指定逻辑通道的标签。

指令：

:LA:ITEM<n>:LABel <lable>

:LA:ITEM<n>:LABel ?

参数：

<n>= 整数型，设置范围 0~15，表示逻辑通道 D0~D15。

<lable>= 通道名称字符串。

返回：

查询返回指定逻辑通道的标签。

示例：

:LA:ITEM1:LABel "abcd" /*设置逻辑通道 1 的标签 abcd*/

:LA:ITEM1:LABel ? /*查询返回 abcd*/

:LA:ITEM<n>:THReshold

功能描述：设置或查询指定逻辑通道的阈值电平属性。

指令：

:LA:ITEM<n>:THReshold <threshold>

:LA:ITEM<n>:THReshold ?

参数：

<n>= 整数型，设置范围 0~15，表示逻辑通道 D0~D15。

<threshold>= 离散型{TTL | CMOS5000 | CMOS3300 | CMOS2500 | CMOS1800 | ECL | PECL | LVDS | USER}。

- TTL:TTL
- CMOS5000:5.0V COMS
- CMOS3300:3.3V COMS
- CMOS2500:2.5V COMS
- CMOS1800:1.8V COMS
- ECL:ECL
- PECL:PECL
- LVDS:LVDS
- USER:USER

返回：

查询返回“TTL”、“CMOS5000”、“CMOS3300”、“CMOS2500”、“CMOS1800”、“ECL”、“PECL”、“LVDS”、“USER”。

示例：

:LA:ITEM1:THReshold CMOS3300 /*设置逻辑通道 1 的阈值电平属性 CMOS3300*/

:LA:ITEM1:THReshold ? /*查询返回 CMOS3300*/

:LA:ITEM<n>:LEVel

功能描述：设置或查询指定逻辑通道的阈值电平。

指令：

:LA:ITEM<n>:LEVel <level>

:LA:ITEM<n>:LEVel ?

参数：

<n>= 整数型，设置范围 0~15，表示逻辑通道 D0~D15。

<level>= 实型，单位 mV、V，默认单位 V。

返回：

查询返回逻辑通道的阈值电平,采用科学计数法。

示例：

:LA:ITEM1:LEVel 3 /*设置逻辑通道 1 的阈值电平 3*/

:LA:ITEM1:LEVel ? /*查询返回 3.00000E+000 */

:LA:HEIGht

功能描述：设置或查询逻辑通道的高度。

指令：

:LA:HEIGht <height>

:LA:HEIGht ?

参数：

<height>= 离散型{SMALl | MEDium | LARGe}。

- SMALl:低
- MEDium:中
- LARGe:高

返回：

查询返回“SMALl”、“MEDium”、“LARGe”。

示例：

:LA:HEIGht LARGe /*设置逻辑通道的高度 LARGe*/

:LA:HEIGht ? /*查询返回 LARGe */

:LA:AUTO:LOCate

功能描述：设置逻辑通道自动排序。

指令：

:LA:AUTO:LOCate <auto>

参数：

<auto>= 布尔型{ON | OFF}或{1 | 0}。

示例：

:LA:AUTO:LOCate ON /*设置逻辑通道的自动排序*/

:LA:ALL:DISPlay

功能描述：设置所有逻辑通道打开。

指令：

:LA:ALL:DISPlay <active>

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}。

示例：

:LA:ALL:DISPlay ON /*设置所有逻辑通道打开*/

SEGement 指令系统

用于设置和查询示波器顺序模式相关参数。

:SEGement:ACTive

功能描述： 设置或查询顺序模式开启状态。

指令：

:SEGement:ACTive <active>

:SEGement:ACTive ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:SEGement:ACTive ON /*设置顺序模式打开*/

:SEGement:ACTive ? /*查询返回 1*/

:SEGement:RSTacq

功能描述： 设置顺序模式为重采。

指令：

:SEGement:RSTacq ON

示例：

:SEGement:RSTacq ON /*设置顺序模式重采*/

:SEGement:MODe

功能描述： 设置或查询顺序模式的工作模式。

指令：

:SEGement:MODe <mode>

:SEGement:MODe ?

参数：

<mode>= 离散型{SINGle | SEQuent}

- SINGle:单帧
- SEQuent:连续帧

返回：

查询返回“SINGle”、“SEQuent”。

示例：

:SEGement:MODe SEQuent /*设置顺序模式的工作模式为连续帧模式*/

:SEGement:MODe ? /*查询返回 SEQuent*/

:SEGement:FRAMe:COUNt

功能描述：设置或查询顺序模式的总帧数。

指令：

:SEGement:FRAMe:COUNt <count>

:SEGement:FRAMe:COUNt ?

参数：

<count>= 整数型

返回：

查询返回顺序模式的总帧数。

示例：

:SEGement:FRAMe:COUNt 100 /*设置顺序模式的的总帧数 100*/

:SEGement:FRAMe:COUNt ? /*查询返回 100*/

:SEGement:SEQuent:TYPe

功能描述：设置或查询顺序模式在连续帧下的波形显示类型。

指令：

:SEGement:SEQuent:TYPe <mode>

:SEGement:SEQuent:TYPe ?

参数：

<mode>= 离散型{ANGLe | STACk | SUPerimpose | SPLiced | NONe}

- ANGLe:45°
- STACk:堆叠
- SUPerimpose:叠加
- SPLiced:拼接

- NONe:无

返回：

查询返回“ANGLe”、“STACk”、“SUPerimpose”、“SPLiced”、“NONe”。

示例：

:SEGement:SEQuent:TYPe ANGLe /*设置顺序模式的在连续帧下的波形显示类型为 45°*/

:SEGement:SEQuent:TYPe ? /*查询返回 ANGLe*/

:SEGement:FRAMe:STARt

功能描述：设置或查询顺序模式在连续帧模式下的起始帧。

指令：

:SEGement:FRAMe:STARt <count>

:SEGement:FRAMe:STARt ?

参数：

<count>= 整数型

返回：

查询返回顺序模式在连续帧模式下的起始帧。

示例：

:SEGement:FRAMe:STARt 1 /*设置顺序模式的起始帧 1*/

:SEGement:FRAMe:STARt ? /*查询返回 1*/

:SEGement:FRAMe:END

功能描述：设置或查询顺序模式在连续帧模式下的结束帧。

指令：

:SEGement:FRAMe:END <count>

:SEGement:FRAMe:END ?

参数：

<count>= 整数型

返回：

查询返回顺序模式在连续帧模式下的结束帧。

示例：

:SEGement:FRAMe:END 15 /*设置顺序模式的结束帧 15*/

:SEGement:FRAMe:END ? /*查询返回 15*/

:SEGement:FRAMe:CURRent

功能描述：设置或查询顺序模式在单帧模式下的选定帧。

指令：

:SEGement:FRAMe:CURRent <count>

:SEGement:FRAMe:CURRent ?

参数：

<count>= 整数型

返回：

查询返回顺序模式在单帧模式下的选定帧。

示例：

:SEGement:FRAMe:CURRent 1 /*设置顺序模式的选定帧 1*/

:SEGement:FRAMe:CURRent ? /*查询返回 1*/

:SEGement:FRAMe:REF

功能描述：设置或查询顺序模式在单帧模式下的参考帧。

指令：

:SEGement:FRAMe:REF <count>

:SEGement:FRAMe:REF ?

参数：

<count>= 整数型

返回：

查询返回顺序模式在单帧模式下的参考帧。

示例：

:SEGement:FRAMe:REF 15 /*设置顺序模式的的参考帧 15*/

:SEGement:FRAMe:REF ? /*查询返回 15*/

:SEGement:REFactive

功能描述：设置或查询顺序模式下参考帧开启状态。

指令：

:SEGement:REFactive <active>

:SEGement:REFactive ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:SEGement:REFactive ON /*设置顺序模式参考帧打开*/

:SEGement:REFactive ? /*查询返回 1*/

:SEGement:CALLback

功能描述：设置或查询顺序模式在单帧模式下的回放状态。

指令：

:SEGement:CALLback <callback>

:SEGement:CALLback ?

参数：

<callback>= 布尔型{ON | OFF}或{1 | 0}

- ON:开始回放
- OFF:停止回放

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:SEGement:CALLback ON /*设置顺序模式回放*/

:SEGement:CALLback ? /*查询返回 1*/

XY 指令系统

用于设置和查询示波器 XY 李沙育(Lissajous)图形。

:XY<n>:ACTive

功能描述：设置或查询指定 XY 是否打开。

指令：

:XY<n>:ACTive <active>

:XY<n>:ACTive ?

参数：

<n>= 整型，范围 1~4

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:XY1:ACTive ON /*设置 XY1 打开*/

:XY1:ACTive ? /*查询返回 1*/

:XY<n>:SOURce:X

功能描述：设置或查询指定 XY 的源 X。

指令：

:XY<n>:SOURce:X <source>

:XY<n>:SOURce:X ?

参数：

<n>= 整型，范围 1~4

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回“C1”、“C2”、“C3”、“C4”。

示例：

:XY1:SOURce:X C1 /*设置 XY1 的源 X 为 C1*/

:XY1:SOURce:X ? /*查询返回 C1*/

:XY<n>:SOURce:Y

功能描述：设置或查询指定 XY 的源 Y。

指令：

:XY<n>:SOURce:Y <source>

:XY<n>:SOURce:Y ?

参数：

<n>= 整型，范围 1~4

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回“C1”、“C2”、“C3”、“C4”。

示例：

:XY1:SOURce:Y C2 /*设置 XY1 的源 Y 为 C2*/

:XY1:SOURce:Y ? /*查询返回 C2*/

:XY<n>:CURSor:ACTive

功能描述：设置或查询指定 XY 的光标是否打开。

指令：

:XY<n>:CURSor:ACTive <active>

:XY<n>:CURSor:ACTive ?

参数：

<n>= 整型，范围 1~4

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:XY1:CURSor:ACTive ON /*设置 XY1 的光标打开*/

:XY1:CURSor:ACTive ? /*查询返回 1*/

:XY<n>:TYPe

功能描述：设置或查询指定 XY 的标记类型。

指令：

:XY<n>:TYPe <type>

:XY<n>:TYPe ?

参数：

<n>= 整型，范围 1~4

<type>= 离散型{CURSor | WAVe}

- CURSor:光标
- WAVe:波形

返回：

查询返回“CURSor”、“WAVe”。

示例：

:XY1:TYPe CURSor /*设置 XY1 标记类型为光标*/

:XY1:TYPe ? /*查询返回 CURSor*/

:XY<n>:DATA:TYPe

功能描述：设置或查询指定 XY 的数据类型。

指令：

:XY<n>:DATA:TYPe <type>

:XY<n>:DATA:TYPe ?

参数：

<n>= 整型，范围 1~4

<type>= 离散型{RECTangle | POLar}

- RECTangle:矩形
- POLar:极性

返回：

查询返回“RECTangle”、“POLar”。

示例：

:XY1:DATA:TYPe POLar /*设置 XY1 数据类型为极性*/

:XY1:DATA:TYPe ? /*查询返回 POLar*/

:XY<n>:CURSor:CAX

功能描述：设置或查询指定 XY 标记在光标模式下垂直光标 A 位置。

指令：

:XY<n>:CURSor:CAX <value>

:XY<n>:CURSor:CAX ?

参数：

<n>= 整型，范围 1~4

<value>= 实型，以 div 为单位的实数,范围 0~10

返回：

查询返回指定 XY 标记在光标模式下垂直光标 A 位置。

示例：

:XY1:CURSor:CAX 2 /*设置 XY1 在光标模式下垂直光标 A 的位置为 2div*/

:XY1:CURSor:CAX ? /*查询返回 2.00000E+000*/

:XY<n>:CURSor:CBX

功能描述：设置或查询指定 XY 标记在光标模式下垂直光标 B 位置。

指令：

:XY<n>:CURSor:CBX <value>

:XY<n>:CURSor:CBX ?

参数：

<n>= 整型，范围 1~4

<value>= 实型，以 div 为单位的实数,范围 0~10

返回：

查询返回指定 XY 标记在光标模式下垂直光标 B 位置。

示例：

:XY1:CURSor:CBX 2 /*设置 XY1 在光标模式下垂直光标 B 的位置为 2div*/

:XY1:CURSor:CBX ? /*查询返回 2.00000E+000*/

:XY<n>:CURSor:CAY

功能描述：设置或查询指定 XY 标记在光标模式下水平光标 A 位置。

指令：

:XY<n>:CURSor:CAY <value>

:XY<n>:CURSor:CAY ?

参数：

<n>= 整型，范围 1~4

<value>= 实型，以 div 为单位的实数,范围-4~4

返回：

查询返回指定 XY 标记在光标模式下水平光标 A 位置。

示例：

:XY1:CURSor:CAY 2 /*设置 XY1 在光标模式下水平光标 A 的位置为 2div*/

:XY1:CURSor:CAY ? /*查询返回 2.00000E+000*/

:XY<n>:CURSor:CBY

功能描述：设置或查询指定 XY 标记在光标模式下水平光标 B 位置。

指令：

:XY<n>:CURSor:CBY <value>

:XY<n>:CURSor:CBY ?

参数：

<n>= 整型，范围 1~4

<value>= 实型，以 div 为单位的实数,范围-4~4

返回：

查询返回指定 XY 标记在光标模式下水平光标 B 位置。

示例：

:XY1:CURSor:CBY 2 /*设置 XY1 在光标模式下水平光标 B 的位置为 2div*/

:XY1:CURSor:CBY ? /*查询返回 2.00000E+000*/

:XY<n>:CURSor:AXValue

功能描述：查询指定 XY 标记在光标模式下垂直光标 A 的矩形数据测量值，单位与源 X 保持一致。

指令：

:XY<n>:CURSor:AXValue ?

参数：

<n>= 整型，范围 1~4

返回：

查询返回指定 XY 标记在光标模式下垂直光标 A 的矩形数据测量值，采用科学计数法,默认单位 V。

:XY<n>:CURSor:BXValue

功能描述：查询指定 XY 标记在光标模式下垂直光标 B 的矩形数据测量值，单位与源 X 保持一致。

指令：

:XY<n>:CURSor:BXValue ?

参数：

<n>= 整型，范围 1~4

返回：

查询返回指定 X 标记在光标模式下垂直光标 B 的矩形数据测量值，采用科学计数法,默认单位 V。

:XY<n>:CURSor:AYValue

功能描述：查询指定 XY 标记在光标模式下水平光标 A 的矩形数据测量值，单位与源 Y 保持一致。

指令：

:XY<n>:CURSor:AYValue ?

参数：

<n>= 整型，范围 1~4

返回：

查询返回指定 XY 标记在光标模式下水平光标 A 的矩形数据测量值，采用科学计数法,默认单位 V。

:XY<n>:CURSor:BYValue

功能描述：查询指定 XY 标记在光标模式下水平光标 B 的矩形数据测量值，单位与源 Y 保持一致。

指令：

:XY<n>:CURSor:BYValue ?

参数：

<n>= 整型，范围 1~4

返回：

查询返回指定 XY 标记在光标模式下水平光标 B 的矩形数据测量值，采用科学计数法,默认单位 V。

:XY<n>:WAVe:AXValue

功能描述：查询指定 XY 标记在波形模式下垂直光标 A 的矩形数据测量值，单位与源 X 保持一直。

指令：

:XY<n>:WAVe:AXValue ?

参数：

<n>= 整型，范围 1~4

返回：

查询返回指定 XY 标记在波形模式下垂直光标 A 的矩形数据测量值，采用科学计数法。

:XY<n>:WAVe:BXValue

功能描述：查询指定 XY 标记在波形模式下垂直光标 B 的矩形数据测量值，单位与源 X 保持一致。

指令：

:XY<n>:WAVe:BXValue ?

参数：

<n>= 整型，范围 1~4

返回：

查询返回指定 XY 标记在波形模式下垂直光标 B 的矩形数据测量值，采用科学计数法。

:XY<n>:WAVe:AYValue

功能描述：查询指定 XY 标记在波形模式下水平光标 A 的矩形数据测量值，单位与源 Y 保持一致。

指令：

:XY<n>:WAVe:AYValue ?

参数：

<n>= 整型，范围 1~4

返回：

查询返回指定 XY 标记在波形模式下水平光标 A 的矩形数据测量值，采用科学计数法。

:XY<n>:WAVe:BYValue

功能描述：查询指定 XY 标记在波形模式下水平光标 B 的矩形数据测量值，单位与源 Y 保持一致。

指令：

:XY<n>:WAVe:BYValue ?

参数：

<n>= 整型，范围 1~4

返回：

查询返回指定 XY 标记在波形模式下水平光标 B 的矩形数据测量值，采用科学计数法。

:XY<n>:RADius:A

功能描述：查询指定 XY 的源 X 和源 Y 在光标 A 处电压值组成的极坐标系半径，单位与源 X 保持一致。

指令：

:XY<n>:RADius:A ?

参数：

<n>= 整型，范围 1~4

返回：

查询返回指定 XY 的源 X 和源 Y 在光标 A 处电压值组成的极坐标系半径，采用科学计数法。

:XY<n>:RADius:B

功能描述：查询指定 XY 的源 X 和源 Y 在光标 B 处电压值组成的极坐标系半径，单位与源 X 保持一致。

指令：

:XY<n>:RADius:B ?

参数：

<n>= 整型，范围 1~4

返回：

查询返回指定 XY 的源 X 和源 Y 在光标 B 处电压值组成的极坐标系半径，采用科学计数法。

:XY<n>:ANGLe:A

功能描述：查询指定 XY 的源 X 和源 Y 在光标 A 处电压值组成的极坐标系夹角，单位度。

指令：

:XY<n>:ANGLe:A ?

参数：

<n>= 整型，范围 1~4

返回：

查询返回指定 XY 的源 X 和源 Y 在光标 A 处电压值组成的极坐标系夹角，采用科学计数法。

:XY<n>:ANGLe:B

功能描述：查询指定 XY 的源 X 和源 Y 在光标 B 处电压值组成的极坐标系夹角，单位度。

指令：

:XY<n>:ANGLe:B ?

参数：

<n>= 整型，范围 1~4

返回：

查询返回指定 XY 的源 X 和源 Y 在光标 B 处电压值组成的极坐标系夹角，采用科学计数法。

ZOOM 指令系统

用于设置和查询示波器 ZOOM 扩展相关信息。

:ZOOM:ACTive

功能描述：设置或查询扩展是否打开。

指令：

:ZOOM:ACTive <active>

:ZOOM:ACTive ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回扩展状态，“1”、“0”分别代表“ON”、“OFF”。

示例：

:ZOOM:ACTive ON /* 设置打开扩展*/

:ZOOM:ACTive ? /*查询返回 1*/

:ZOOM:CENTer:X

功能描述：设置或查询扩展中心点垂直位置。

指令：

:ZOOM:CENTer:X <value>

:ZOOM:CENTer:X ?

参数：

<value>= 实型，范围 0-10000

返回：

查询返回扩展中心点垂直位置。

示例：

:ZOOM:CENTer:X 5000 /* 设置打开扩展垂直中心点为 5000*/

:ZOOM:CENTer:X ? /*查询返回 5.00000E+003*/

:ZOOM:CENTer:Y

功能描述：设置或查询扩展中心点水平位置。

指令：

:ZOOM:CENTer:Y <value>

:ZOOM:CENTer:Y ?

参数：

<value>= 实型，范围-4000-4000

返回：

查询返回扩展中心点水平位置。

示例：

:ZOOM:CENTer:Y 0 /* 设置打开扩展水平中心点为 0*/

:ZOOM:CENTer:Y ? /*查询返回 0.00000E+000*/

:ZOOM:SCALe:X

功能描述：设置或查询扩展水平方向比例。

指令：

:ZOOM:SCALe:X <value>

:ZOOM:SCALe:X ?

参数：

<value>= 实型，范围 1-1000

返回：

查询返回扩展水平方向比例。

示例：

:ZOOM:SCALe:X 10 /*设置扩展水平方向比例为 10*/

:ZOOM:SCALe:X ? /*查询返回 1.00000E+001*/

:ZOOM:SCALe:Y

功能描述：设置或查询扩展垂直方向比例。

指令：

:ZOOM:SCALe:Y <value>

:ZOOM:SCALe:Y ?

参数：

<value>= 实型，范围 1-1000

返回：

查询返回扩展垂直方向比例。

示例：

:ZOOM:SCALe:Y 10 /*设置打开扩展水平中心点为 10*/

:ZOOM:SCALe:Y ? /*查询返回 1.00000E+001*/

JITTer 指令系统

用于设置或查询示波器的抖动分析功能。

:JITTer:ACTive

功能描述：设置或查询抖动分析是否打开。

指令：

:JITTer:ACTive <active>

:JITTer:ACTive ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回抖动分析打开状态，“1”、“0”分别代表“ON”、“OFF”。

示例：

:JITTer:ACTive ON /*设置打开抖动分析*/

:JITTer:ACTive ? /*查询返回 1*/

:JITTer:SOURce

功能描述：设置或查询抖动分析数据源。

指令：

:JITTer:SOURce <source>

:JITTer:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回“C1”、“C2”、“C3”、“C4”。

示例：

:JITTer:SOURce C1 /*设置抖动分析数据源为 C1*/

:JITTer:SOURce ? /*查询返回 C1*/

:JITTer:SIGNal:TYPe

功能描述：设置或查询抖动分析信号类型。

指令：

:JITTer:SIGNal:TYPe <type>

:JITTer:SIGNal:TYPe ?

参数：

<type>= 离散型{CUSTom | CLOCk}

- CUSTom:数据信号
- CLOCk:时钟信号

返回：

查询返回“CUSTom”、“CLOCk”。

示例：

:JITTer:SIGNal:TYPe CLOCk /*设置抖动分析信号类型为时钟信号*/

:JITTer:SIGNal:TYPe ? /*查询返回 CLOCk*/

:JITTer:DTALength

功能描述：设置或查询数据信号类型下的数据长度。

指令：

:JITTer:DTALength <value>

:JITTer:DTALength ?

参数：

<value>= 整数型，范围 0~2Gbit，默认单位 bit

返回：

查询返回数据信号类型下的数据长度。

示例：

:JITTer:DTALength 127 /*设置数据信号类型下的数据长度为 127bit*/

:JITTer:DTALength ? /*查询返回 127*/

:JITTer:THReshold

功能描述：设置或查询抖动分析的比较阈值,阈值范围 45%~55%。

指令：

:JITTer:THReshold <value>

:JITTer:THReshold ?

参数：

<value>= 实型

返回：

查询返回抖动分析的比较阈值。

示例：

:JITTer:THReshold 50 /*设置抖动分析的比较阈值为 50%*/

:JITTer:THReshold ? /*查询返回 5.00000E+001*/

:JITTer:HYSTeresis

功能描述：设置或查询抖动分析的迟滞。

指令：

:JITTer:HYSTeresis <value>

:JITTer:HYSTeresis ?

参数：

<value>= 实型

返回：

查询返回抖动分析的迟滞。

示例：

:JITTer:HYSTeresis 30 /*设置抖动分析的迟滞为 30%*/

:JITTer:HYSTeresis ? /*查询返回 3.00000E+001*/

:JITTer:BITRate

功能描述：设置或查询抖动分析的比特率。

指令：

:JITTer:BITRate <bit>

:JITTer:BITRate?

参数：

<value>= 实型，范围 0~5Gbps，默认单位 bps

返回：

查询返回抖动分析的比特率。

示例：

:JITTer:BITRate 20 /*设置抖动分析的比特率为 20*/

:JITTer:BITRate ? /*查询返回 2.00000E+001*/

:JITTer:CLOCk:TYPe

功能描述：设置或查询抖动分析的时钟恢复方法。

指令：

:JITTer:CLOCk:TYPe <type>

:JITTer:CLOCk:TYPe ?

参数：

<type>= 离散型{CONStant | PLL | EXTernal}

- CONStant:常频时钟
- PLL:锁相环
- EXTernal:外部时钟

返回：

查询返回“CONStant”、“PLL”、“EXTernal”。

示例：

:JITTer:CLOCk:TYPe PLL /*设置抖动分析的时钟恢复方法为 PLL*/

:JITTer:CLOCk:TYPe ? /*查询返回 PLL*/

:JITTer:PLL:TYPe

功能描述：设置或查询抖动分析的 PLL 模式。

指令：

:JITTer:PLL:TYPe <type>

:JITTer:PLL:TYPe ?

参数：

<type>= 离散型{GOLDen | SECondOrder}

- GOLDen:黄金分割频率锁相环
- SECondOrder:二阶锁相环

返回：

查询返回“GOLDen”、“SECondOrder”。

示例：

:JITTer:PLL:TYPe GOLDen /*设置抖动分析的 PLL 模式为 GOLDen*/

:JITTer:PLL:TYPe ? /*查询返回 GOLDen */

:JITTer:PLL:DIVisor:CUToff

功能描述：设置或查询抖动分析黄金分割频率锁相环的截止系数。

指令：

:JITTer:PLL:DIVisor:CUToff <value>

:JITTer:PLL:DIVisor:CUToff ?

参数：

<value>= 实型

返回：

查询返回抖动分析黄金分割频率锁相环的截止系数。

示例：

:JITTer:PLL:DIVisor:CUToff 100 /*设置抖动分析黄金分割频率锁相环的截止系数为 100 */

:JITTer:PLL:DIVisor:CUToff ? /*查询返回 1.00000E+002*/

:JITTer:PLL:FREQ:CUToff

功能描述：设置或查询抖动分析黄金分割频率锁相环的截止频率，默认单位 Hz。

指令：

:JITTer:PLL:FREQ:CUToff <value>

:JITTer:PLL:FREQ:CUToff ?

参数：

<value>= 实型，单位 mHz、Hz

返回：

查询返回抖动分析黄金分割频率锁相环的截止频率。

示例：

:JITTer:PLL:FREQ:CUToff 10 /*设置抖动分析黄金分割频率锁相环的截止频率为 10Hz */

:JITTer:PLL:FREQ:CUToff ? /*查询返回 1.0000000000000E+001 */

:JITTer:PLL:FREQ:NATural

功能描述：设置或查询抖动分析二阶锁相环的自然频率，默认单位 Hz。

指令：

:JITTer:PLL:FREQ:NATural <value>

:JITTer:PLL:FREQ:NATural ?

参数：

<value>= 实型，单位 mHz、Hz

返回：

设置或查询抖动分析二阶锁相环的自然频率，默认单位 Hz。

示例：

:JITTer:PLL:FREQ:NATural 10 /*设置抖动分析二阶锁相环的自然频率为 10Hz*/

:JITTer:PLL:FREQ:NATural ? /*查询返回 1.0000000000000E+001 */

:JITTer:PLL:FACTor

功能描述：设置或查询抖动分析二阶锁相环的阻尼因子。

指令：

:JITTer:PLL:FACTor <value>

:JITTer:PLL:FACTor ?

参数：

<value>= 实型

返回：

查询返回二阶锁相环的阻尼因子。

示例：

:JITTer:PLL:FACTor 0.6 /*设置抖动分析二阶锁相环的阻尼因子为 0.6*/

:JITTer:PLL:FACTor ? /*查询返回 6.00000E-001*/

:JITTer:EXTernal:SOURce

功能描述：设置或查询抖动分析外部时钟源。

指令：

:JITTer:EXTernal:SOURce <source>

:JITTer:EXTernal:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回抖动分析外部时钟源。

示例：

:JITTer:EXTernal:SOURce C2 /*设置抖动分析为外部时钟时的信源为 C2*/

:JITTer:EXTernal:SOURce ? /*查询返回 C2*/

:JITTer:EXTernal:THReshold

功能描述：设置或查询抖动分析外部时钟阈值。

指令：

:JITTer:EXTernal:THReshold <value>

:JITTer:EXTernal:THReshold ?

参数：

<value>= 整数型

返回：

查询返回抖动分析外部时钟阈值。

示例：

:JITTer:EXTernal:THReshold 45 /*设置抖动分析外部时钟阈值为 45%*/

:JITTer:EXTernal:THReshold ? /*查询返回 4.50000E+001*/

:JITTer:EXTernal:HYSTeresis

功能描述：设置或查询抖动分析外部时钟迟滞。

指令：

:JITTer:EXTernal:HYSTeresis <value>

:JITTer:EXTernal:HYSTeresis ?

参数：

<value>= 整数型

返回：

查询返回抖动分析外部时钟迟滞。

示例：

:JITTer:EXTernal:HYSTeresis 30 /*设置抖动分析外部时钟迟滞为 30%*/

:JITTer:EXTernal:HYSTeresis ? /*查询返回 3.00000E+001*/

:JITTer:PARams:JITTer

功能描述：设置或查询抖动分析抖动参数是否打开。

指令：

:JITTer:PARams:JITTer <active>

:JITTer:PARams:JITTer ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回抖动分析抖动参数打开状态，“1”、“0”分别代表“ON”、“OFF”。

示例：

:JITTer:PARams:JITTer ON /*设置抖动分析抖动参数打开*/

:JITTer:PARams:JITTer ? /*查询返回 1*/

:JITTer:HISTogram:BINNum

功能描述：设置或查询抖动分析直方图柱体数。

指令：

:JITTer:HISTogram:BINNum <value>

:JITTer:HISTogram:BINNum ?

参数：

<value>= 离散型{25 | 50 | 100 | 250 | 500 | 2000 | MAX}

返回：

查询返回抖动分析直方图柱体数。

示例：

:JITTer:HISTogram:BINNum 25 /*设置抖动分析直方图柱体数 25*/

:JITTer:HISTogram:BINNum ? /*查询返回 25*/

:JITTer:GRAPh

功能描述：设置抖动分析图形是否打开。

指令：

:JITTer:GRAPh <type> <enable>

参数：

<type>= 离散型{TRENd | SPECtrum | HISTogram | BATHtub | ALL}

- TRENd:趋势图
- SPECtrum:频谱图

- HISTogram:直方图
- BATHtub:浴盆曲线
- ALL:全部

<enable>= 布尔型{ON | OFF}或{1 | 0}

示例:

:JITTer:GRAPh TRENd ON /*设置抖动分析趋势图打开*/

:JITTer:DATA:JITTer

功能描述: 查询抖动分析参数结果，并按照如下排列:

“TIE”、“TJ”、“RJ”、“DJ”、“PJ”、“DDJ”、“DCD”、“ISI”

- TIE:时间间隔误差
- TJ:特定误码率下的总体抖动
- RJ:随机抖动
- DJ:确定性抖动
- PJ:周期性抖动
- DDJ:数据相关性抖动
- DCD:占空比失真
- ISI:码间干扰

指令:

:JITTer:DATA:JITTer ?

返回:

查询返回抖动分析参数结果值，符合数据块格式，以科学计数法表示，并按顺序以逗号按分隔排列，无效值使用实型数据最大值表示。

例如:“#90000001001.20000E+000,2.00000E+002,1.20000E+003.....”

:JITTer:EYEDiagram:ACTive

功能描述: 设置或者查询眼图否打开。

指令:

:JITTer:EYEDiagram:ACTive <active>

:JITTer:EYEDiagram:ACTive ?

参数:

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回眼图打开状态，“1”、“0”分别代表“ON”、“OFF”。

示例：

:JITTer:EYEDiagram:ACTive ON /*设置眼图打开*/

:JITTer:EYEDiagram:ACTive ? /*查询返回 1*/

:JITTer:EYEDiagram:SOURce

功能描述：设置或者查询眼图信源。

指令：

:JITTer:EYEDiagram:SOURce <source>

:JITTer:EYEDiagram:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8}

返回：

查询返回“C1”、“C2”、“C3”、“C4”、“M1”、“M2”、“M3”、“M4”、“M5”、“M6”、“M7”、“M8”。

示例：

:JITTer:EYEDiagram:SOURce C1 /*设置眼图信源为 C1*/

:JITTer:EYEDiagram:SOURce ? /*查询返回 C1*/

:JITTer:EYEDiagram:FAST

功能描述：设置或者查询快速眼图打开状态。

指令：

:JITTer:EYEDiagram:FAST <active>

:JITTer:EYEDiagram:FAST ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回快速眼图打开状态，“1”、“0”分别代表“ON”、“OFF”。

示例：

:JITTer:EYEDiagram:FAST 1 /*设置快速眼图打开*/

:JITTer:EYEDiagram:FAST ? /*查询返回 1*/

:JITTer:EYEDiagram:SLOPe

功能描述：设置或者查询眼图边沿类型。

指令：

:JITTer:EYEDiagram:SLOPe <type>

:JITTer:EYEDiagram:SLOPe ?

参数：

<type>= 离散型{RISe | FALL | ANY}

- RISe:上升沿
- FALL:下降沿
- ANY:任意

返回：

查询返回眼图边沿类型。

示例：

:JITTer:EYEDiagram:SLOPe ANY /*设置眼图边沿类型 ANY*/

:JITTer:EYEDiagram:SLOPe ? /*查询返回 ANY*/

:JITTer:EYEDiagram:POLarity

功能描述：设置或者查询眼图电平极性。

指令：

:JITTer:EYEDiagram:POLarity <type>

:JITTer:EYEDiagram:POLarity ?

参数：

<type>= 离散型{POSitive | NEGative | ANY}

- POSitive:正极性
- NEGative:负极性
- ANY:任意

返回：

查询返回眼图电平极性。

示例：

:JITTer:EYEDiagram:POLarity POSitive /*设置眼图电平极性为正极性*/

:JITTer:EYEDiagram:POLarity ? /*查询返回 POSitive*/

:JITTer:EYEDiagram:THReshold

功能描述：设置或查询眼图的比较阈值,阈值范围 45%~55%。

指令：

:JITTer:EYEDiagram:THReshold <value>

:JITTer:EYEDiagram:THReshold ?

参数：

<value>= 实型

返回：

查询返回眼图的比较阈值。

示例：

:JITTer:EYEDiagram:THReshold 50 /*设置眼图的比较阈值为 50%*/

:JITTer:EYEDiagram:THReshold ? /*查询返回 5.00000E+001*/

:JITTer:EYEDiagram:HYSTeresis

功能描述：设置或查询眼图的迟滞。

指令：

:JITTer:EYEDiagram:HYSTeresis <value>

:JITTer:EYEDiagram:HYSTeresis ?

参数：

<value>= 实型

返回：

查询返回眼图的迟滞。

示例：

:JITTer:EYEDiagram:HYSTeresis 30 /*设置眼图的迟滞为 30%*/

:JITTer:EYEDiagram:HYSTeresis ? /*查询返回 3.00000E+001*/

:JITTer:EYEDiagram:BITRate

功能描述：设置或查询眼图的比特率。

指令：

:JITTer:EYEDiagram:BITRate <bit>

:JITTer:EYEDiagram:BITRate?

参数：

<value>= 实型，范围 0~5Gbps，默认单位 bps

返回：

查询返回眼图的比特率。

示例：

:JITTer:EYEDiagram:BITRate 20 /*设置眼图的比特率为 20*/

:JITTer:EYEDiagram:BITRate ? /*查询返回 2.00000E+001*/

:JITTer:EYEDiagram:CLOCk:TYPe

功能描述：设置或查询眼图的时钟恢复方法。

指令：

:JITTer:EYEDiagram:CLOCk:TYPe <type>

:JITTer:EYEDiagram:CLOCk:TYPe ?

参数：

<type>= 离散型{CONStant | PLL | EXTernal}

- CONStant:常频时钟
- PLL:锁相环
- EXTernal:外部时钟

返回：

查询返回“CONStant”、“PLL”。

示例：

:JITTer:EYEDiagram:CLOCk:TYPe PLL /*设置眼图时钟恢复方法为 PLL*/

:JITTer:EYEDiagram:CLOCk:TYPe ? /*查询返回 PLL*/

:JITTer:EYEDiagram:PLL:TYPe

功能描述：设置或查询眼图的 PLL 模式。

指令：

:JITTer:EYEDiagram:PLL:TYPe <type>

:JITTer:EYEDiagram:PLL:TYPe ?

参数：

<type>= 离散型{GOLDen | SECondOrder}

- GOLDen:黄金分割频率锁相环

- SECondOrder:二阶锁相环

返回：

查询返回“GOLDen”、“SECondOrder”。

示例：

:JITTer:EYEDiagram:PLL:TYPe GOLDen /*设置眼图的 PLL 模式为 GOLDen */

:JITTer:EYEDiagram:PLL:TYPe ? /*查询返回 GOLDen */

:JITTer:EYEDiagram:PLL:DIVisor

功能描述：设置或查询眼图黄金分割频率锁相环的截止系数。

指令：

:JITTer:EYEDiagram:PLL:DIVisor <value>

:JITTer:EYEDiagram:PLL:DIVisor ?

参数：

<value>= 实型

返回：

查询返回眼图黄金分割频率锁相环的截止系数。

示例：

:JITTer:EYEDiagram:PLL:DIVisor 100 /*设置眼图黄金分割频率锁相环的截止系数为 100 */

:JITTer:EYEDiagram:PLL:DIVisor ? /*查询返回 1.00000E+002*/

:JITTer:EYEDiagram:PLL:CUTFrequency

功能描述：设置或查询眼图黄金分割频率锁相环的截止频率，默认单位 Hz。

指令：

:JITTer:EYEDiagram:PLL:CUTFrequency <value>

:JITTer:EYEDiagram:PLL:CUTFrequency ?

参数：

<value>= 实型，单位 mHz、Hz

返回：

查询返回眼图黄金分割频率锁相环的截止频率。

示例：

:JITTer:EYEDiagram:PLL:CUTFrequency 10 /*设置眼图黄金分割频率锁相环的截止频率为 10Hz */

:JITTer:EYEDiagram:PLL:CUTFrequency ? /*查询返回 1.0000000000000E+001 */

:JITTer:EYEDiagram:PLL:NATFrequency

功能描述：设置或查询眼图二阶锁相环的自然频率，默认单位 Hz。

指令：

:JITTer:EYEDiagram:PLL:NATFrequency <value>

:JITTer:EYEDiagram:PLL:NATFrequency ?

参数：

<value>= 实型，单位 mHz、Hz

返回：

查询返回眼图二阶锁相环的自然频率。

示例：

:JITTer:EYEDiagram:PLL:NATFrequency 10 /*设置眼图二阶锁相环的自然频率为 10Hz */

:JITTer:EYEDiagram:PLL:NATFrequency ? /*查询返回 1.0000000000000E+001 */

:JITTer:EYEDiagram:PLL:FACTor

功能描述：设置或查询眼图二阶锁相环的阻尼因子。

指令：

:JITTer:EYEDiagram:PLL:FACTor <value>

:JITTer:EYEDiagram:PLL:FACTor ?

参数：

<value>= 实型

返回：

查询返回二阶锁相环的阻尼因子。

示例：

:JITTer:EYEDiagram:PLL:FACTor 0.6 /*设置眼图二阶锁相环的阻尼因子为 0.6*/

:JITTer:EYEDiagram:PLL:FACTor ? /*查询返回 6.00000E-001*/

:JITTer:EYEDiagram:EXTernal:SOURce

功能描述：设置或查询眼图外部时钟源。

指令：

:JITTer:EYEDiagram:EXTernal:SOURce <source>

:JITTer:EYEDiagram:EXTernal:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回眼图外部时钟源。

示例：

:JITTer:EYEDiagram:EXTernal:SOURce C2 /*设置眼图外部时钟源为 C2*/

:JITTer:EYEDiagram:EXTernal:SOURce ? /*查询返回 C2*/

:JITTer:EYEDiagram:EXTernal:THReshold

功能描述：设置或查询眼图外部时钟阈值。

指令：

:JITTer:EYEDiagram:EXTernal:THReshold <value>

:JITTer:EYEDiagram:EXTernal:THReshold ?

参数：

<value>= 整数型

返回：

查询返回眼图外部时钟阈值。

示例：

:JITTer:EYEDiagram:EXTernal:THReshold 45 /*设置眼图外部时钟阈值为 45%*/

:JITTer:EYEDiagram:EXTernal:THReshold ? /*查询返回 4.50000E+001*/

:JITTer:EYEDiagram:EXTernal:HYSTeresis

功能描述：设置或查询眼图外部时钟迟滞。

指令：

:JITTer:EYEDiagram:EXTernal:HYSTeresis <value>

:JITTer:EYEDiagram:EXTernal:HYSTeresis ?

参数：

<value>= 整数型

返回：

查询返回眼图外部时钟迟滞。

示例：

:JITTer:EYEDiagram:EXTernal:HYSTeresis 10 /*设置眼图外部时钟迟滞为 10%*/

:JITTer:EYEDiagram:EXTernal:HYSTeresis ? /*查询返回 1.00000E+001*/

:JITTer:EYEDiagram:PARams:ACTive

功能描述：设置或查询眼图参数是否打开。

指令：

:JITTer:EYEDiagram:PARams:ACTive <active>

:JITTer:EYEDiagram:PARams:ACTive ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回眼图参数打开状态，“1”、“0”分别代表“ON”、“OFF”。

示例：

:JITTer:EYEDiagram:PARams:ACTive ON /*设置眼图参数打开*/

:JITTer:EYEDiagram:PARams:ACTive ? /*查询返回 1*/

:JITTer:EYEDiagram:GATe

功能描述：设置或查询眼图的选择门限

指令：

:JITTer:EYEDiagram:GATe <type>

:JITTer:EYEDiagram:GATe ?

参数：

<type>= 离散型{SCReen | CURSor}

- SCReen:屏幕
- CURSor:光标

返回：

查询返回眼图的选择门限。

示例：

:JITTer:EYEDiagram:GATe SCReen /*设置眼图的选择门限为屏幕*/

:JITTer:EYEDiagram:GATe ? /*查询返回 SCReen*/

:JITTer:EYEMask:ACTive

功能描述：设置或查询眼图模板是否打开。

指令：

:JITTer:EYEMask:ACTive <active>

:JITTer:EYEMask:ACTive ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

- ON:打开
- OFF:关闭

返回：

查询返回 1 或 0，分别代表 ON 或 OFF。

示例：

:JITTer:EYEMask:ACTive ON /*设置打开眼图模板*/

:JITTer:EYEMask:ACTive ? /*查询返回 1*/

:JITTer:EYEMask:MODe

功能描述：设置或查询眼图模板类型。

指令：

:JITTer:EYEMask:MODe <mode>

:JITTer:EYEMask:MODe ?

参数：

<mode>= 离散型{STDMask | CUSTomMask}

- STDMask:标准测试
- CUSTomMask:自定义测试

返回：

查询返回眼图模板类型。

示例：

:JITTer:EYEMask:MODe STDMask /*设置眼图模板类型为标准测试*/

:JITTer:EYEMask:MODe ? /*查询返回 STDMask*/

:JITTer:EYEMask:STDMask

功能描述：设置或查询眼图模板标准测试类型。

指令：

:JITTer:EYEMask:STDMask <type>

:JITTer:EYEMask:STDMask ?

参数：

<type>= 离散型{IEEE | USB | SERial | SATA | INFiniband | CAN}

返回：

查询返回眼图模板标准测试类型。

示例：

:JITTer:EYEMask:STDMask CAN /*设置眼图模板标准测试类型为 CAN*/

:JITTer:EYEMask:STDMask ? /*查询返回 CAN*/

:JITTer:EYEMask:STANdard:INDex

功能描述：设置或查询眼图模板标准测试模板索引。

指令：

:JITTer:EYEMask:STANdard:INDex <index>

:JITTer:EYEMask:STANdard:INDex ?

参数：

<index>= 整数型

返回：

查询返回眼图模板标准测试模板索引。

示例：

:JITTer:EYEMask:STANdard:INDex 1 /*设置眼图模板标准测试模板索引为 1*/

:JITTer:EYEMask:STANdard:INDex ? /*查询返回 1*/

:JITTer:EYEMask:COUNt

功能描述：设置或查询眼图模板测试总违例次数。

指令：

:JITTer:EYEMask:COUNt <count>

:JITTer:EYEMask:COUNt ?

参数：

<count>= 整数型

返回：

查询返回眼图模板测试总违例次数。

示例：

:JITTer:EYEMask:COUNt 1000 /*设置眼图模板测试总违例次数为 1000 次*/

:JITTer:EYEMask:COUNt ? /*查询返回 1000*/

:JITTer:EYEMask:DATA

功能描述：查询眼图模板测试数据。

指令：

:JITTer:EYEMask:DATA ?

返回：

返回眼图模板测试数据。

:JITTer:DATA:EYE

功能描述：查询眼图参数结果，并按照如下排列：

“ZeroLevel”、“OneLevel”、“EyeAmplitude”、“EyeHeight”、“EyeWidth”、“ExtinctionRatio”、“QFactor”、“EyeCrossRatio”

- ZeroLevel:0 电平
- OneLevel:1 电平
- EyeAmplitude:眼幅度
- EyeHeight:眼高
- EyeWidth:眼宽
- ExtinctionRatio:消光比
- QFactor:Q 因子
- EyeCrossRatio:眼交叉比

指令：

:JITTer:DATA:EYE ?

返回：

查询返回眼图参数结果值，符合数据块格式，以科学计数法表示，并按顺序以逗号按分隔排列，无效值使用实型数据最大值表示。

例如：“#90000001001.20000E+000,2.00000E+002,1.20000E+003.....”

PASS/FAIL 指令系统

用于设置或查询示波器的通过失败功能。

:PF:ACTive

功能描述：设置或查询通过失败测试是否打开。

指令：

:PF:ACTive <active>

:PF:ACTive ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回通过失败测试打开状态，“1”、“0”分别代表“ON”、“OFF”。

示例：

:PF:ACTive ON /*设置打开通过失败测试*/

:PF:ACTive ? /*查询返回 1*/

:PF:VISible

功能描述：设置或查询通过失败测试是否显示测试模板。

指令：

:PF:VISible <visible>

:PF:VISible ?

参数：

<visible>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回通过失败测试是否显示测试模板，“1”、“0”分别代表“ON”、“OFF”。

示例：

:PF:VISible ON /*设置显示通过失败测试显示模板*/

:PF:VISible ? /*查询返回 1*/

:PF:MODe

功能描述：设置或查询通过失败测试类型。

指令：

:PF:MODe <type>

:PF:MODe ?

参数：

<type>= 离散型{LIMitMode | STDMaskmode | CUSTommaskmode}

- LIMitMode:极限测试
- STDMaskmode:标准测试
- CUSTommaskmode：自定义

返回：

查询返回通过失败测试类型。

示例：

:PF:MODe LIMitMode /*设置通过失败测试类型为极限测试*/

:PF:MODe ? /*查询返回 LIMitMode*/

:PF:SOURce

功能描述：设置或查询通过失败测试数据源。

指令：

:PF:SOURce <source>

:PF:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4 }

返回：

查询返回通过失败测试数据源。

示例：

:PF:SOURce C1 /*设置通过失败测试数据源为 C1*/

:PF:SOURce ? /*查询返回 C1*/

:PF:LIMit:SOURce

功能描述：设置或查询通过失败极限测试参考源。

指令：

:PF:LIMit:SOURce <source>

:PF:LIMit:SOURce ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4 }

返回：

查询返回通过失败极限测试参考源。

示例：

:PF:LIMit:SOURce C1 /*设置通过失败极限测试参考源为 C1*/

:PF:LIMit:SOURce ? /*查询返回 C1*/

:PF:LIMit:VTOLerance

功能描述：设置或查询通过失败极限测试模板垂直容限，默认单位 mdiv。

指令：

:PF:LIMit:VTOLerance <value>

:PF:LIMit:VTOLerance?

参数：

<value>= 整数型，范围 1mdiv~1div

返回：

查询返回通过失败极限测试模板垂直容限。

示例：

:PF:LIMit:VTOLerance 100 /*设置通过失败极限测试模板垂直容限 100mdiv*/

:PF:LIMit:VTOLerance ? /*查询返回 100*/

:PF:LIMit:HTOLerance

功能描述：设置或查询通过失败极限测试模板水平容限，默认单位 mdiv。

指令：

:PF:LIMit:HTOLerance <value>

:PF:LIMit:HTOLerance ?

参数：

<value>= 整数型，单位范围 1mdiv~500mdiv

返回：

查询返回通过失败极限测试模板水平容限。

示例：

:PF:LIMit:HTOLerance 100 /*设置通过失败极限测试模板水平容限 100mdiv*/

:PF:LIMit:HTOLerance ? /*查询返回 100*/

:PF:LIMit:CREate

功能描述：创建通过失败极限测试模板。

指令：

:PF:LIMit:CREate

:PF:STANdard:TYPe

功能描述：设置或查询通过失败标准测试类型。

指令：

:PF:STANdard:TYPe <type>

:PF:STANdard:TYPe ?

参数：

<type>= 离散型{ANSIT | ITU | USB | ANSIX | IEEE}

- ANSIT:ANSI T1.102
- ITU:ITU-T
- USB:USB
- ANSIX：ANSI X3.263
- IEEE:IEEE 802.3

返回：

查询返回通过失败标准测试类型。

示例：

:PF:STANdard:TYPe ANSIT /*设置通过失败标准测试类型 ANSI*/

:PF:STANdard:TYPe ? /*查询返回 ANSIT*/

:PF:OFFSet:VERTical

功能描述：设置或查询自定义模板垂直偏移。

指令：

:PF:OFFSet:VERTical <value>

:PF:OFFSet:VERTical ?

参数：

<value>= 实型，默认单位为 V，可设置范围为-1.6V~1.6V。

返回：

查询返回自定义模板垂直偏移数值。

示例：

:PF:OFFSet:VERTical 1.6V /*设置自定义模板垂直偏移 1.6V*/

:PF:OFFSet:VERTical ? /*查询返回 1.60000E+000l*/

:PF:OFFSet:HORizontal

功能描述：设置或查询自定义模板水平偏移。

指令：

:PF:OFFSet:HORizontal <value>

:PF:OFFSet:HORizontal ?

参数：

<value>= 实型，默认单位为 s

返回：

查询返回自定义模板垂直偏移数值。

示例：

:PF:OFFSet:HORizontal 20ms /*设置自定义模板水平偏移 20ms*/

:PF:OFFSet:HORizontal ? /*查询返回 2.00000E-002*/

:PF:OFFSet:RSTVertical

功能描述：复位自定义模板垂直偏移。

指令：

:PF:OFFSet:RSTVertical

:PF:OFFSet:RSTHorizontal

功能描述：复位自定义模板水平偏移。

指令：

:PF:OFFSet:RSTHorizontal

:PF:STANdard:LOCK

功能描述：设置或查询通过失败标准测试模板是否锁定。

指令：

:PF:STANdard:LOCK <lock>

:PF:STANdard:LOCK ?

参数：

<lock>= 布尔型{ON | OFF}或{1 | 0}

- ON 或 1:锁定
- OFF 或 0:未锁定

返回：

查询返回通过失败标准测试模板锁定状态，“1”、“0”分别代表“ON”、“OFF”。

示例：

:PF:STANdard:LOCK ON /*设置通过失败标准测试模板锁定*/

:PF:STANdard:LOCK ? /*查询返回 1*/

:PF:STANdard:INDex

功能描述：设置或查询通过失败标准测试模板索引。

指令：

:PF:STANdard:INDex <index>

:PF:STANdard:INDex ?

参数：

<index>= 整数型

返回：

查询返回通过失败标准测试模板索引。

示例：

:PF:STANdard:INDex 10 /*设置通过失败标准测试模板索引为 10*/

:PF:STANdard:INDex ? /*查询返回 10*/

:PF:STOP:TYPe

功能描述：设置或查询通过失败测试结束条件。

指令：

:PF:STOP:TYPe <type>

:PF:STOP:TYPe ?

参数：

<type>= 离散型{WFMs | TIMe}

- WFMs:波形总数
- TIMe:总时间

返回：

查询返回通过失败测试结束条件。

示例：

:PF:STOP:TYPe WFMs /*设置通过失败测试结束条件为 WFMs*/

:PF:STOP:TYPe ? /*查询返回 WFMs*/

:PF:WFMS:COUNt

功能描述：设置或查询通过失败测试结束条件为波形时的波形总数。

指令：

:PF:WFMS:COUNt <value>

:PF:WFMS:COUNt ?

参数：

<value>= 整数型，范围 1-100000

返回：

查询返回通过失败测试结束条件为波形时的波形总数。

示例：

:PF:WFMS:COUNt 1000 /*设置通过失败测试的波形总数为 1000*/

:PF:WFMS:COUNt ? /*查询返回 1000*/

:PF:TIMe:DURation

功能描述：设置或查询通过失败测试结束条件为时间时的总时间，默认单位 s。

指令：

:PF:TIMe:DURation <value>

:PF:TIMe:DURation ?

参数：

<value>= 实型，范围 100ms-1Ms

返回：

查询返回通过失败测试结束条件为时间时的总时间。

示例：

:PF:TIMe:DURation 100ms /*设置通过失败测试的总时间为 100ms*/

:PF:TIMe:DURation ? /*查询返回 1.0000000000000E-001*/

:PF:VIOLations:COUNt

功能描述：设置或查询通过失败测试总违例次数。

指令：

:PF:VIOLations:COUNt <value>

:PF:VIOLations:COUNt ?

参数：

<value>= 整数型，范围 1-1000

返回：

查询返回通过失败测试总违例次数。

示例：

:PF:VIOLations:COUNt 10 /*设置通过失败测试的总违例次数为 10*/

:PF:VIOLations:COUNt ? /*查询返回 10*/

:PF:OPTion:SAVe

功能描述：设置或查询通过失败测试失败操作是否保存。

指令：

:PF:OPTion:SAVe <enable>

:PF:OPTion:SAVe ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

- ON:保存
- OFF:不保存

返回：

查询返回通过失败测试操作失败是否保存，“1”、“0”分别代表“ON”、“OFF”。

示例：

:PF:OPTion:SAVe ON /*设置通过失败测试失败保存*/

:PF:OPTion:SAVe ? /*查询返回 1*/

:PF:OPTion:BEEP

功能描述：设置或查询通过失败测试失败操作是否报警。

指令：

:PF:OPTion:BEEP <enable>

:PF:OPTion:BEEP ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

- ON:报警
- OFF:不报警

返回：

查询返回通过失败测试操作失败是否报警，“1”、“0”分别代表“ON”、“OFF”。

示例：

:PF:OPTion:BEEP ON /*设置通过失败测试失败报警*/

:PF:OPTion:BEEP ? /*查询返回 1*/

:PF:OPTion:PULSe

功能描述：设置或查询通过失败测试失败操作是否输出脉冲。

指令：

:PF:OPTion:PULSe <enable>

:PF:OPTion:PULSe ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

- ON:输出脉冲
- OFF:不打开脉冲

返回：

查询返回通过失败测试失败操作是否输出脉冲，“1”、“0”分别代表“ON”、“OFF”。

示例：

:PF:OPTion:PULSe ON /*设置通过失败测试失败输出脉冲*/

:PF:OPTion:PULSe ? /*查询返回 1*/

:PF:OPTion:SCReen

功能描述：设置或查询通过失败测试失败操作是否截屏。

指令：

:PF:OPTion:SCReen <enable>

:PF:OPTion:SCReen ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

- ON:截屏
- OFF:不截屏

返回：

查询返回通过失败测试失败操作是否截屏，“1”、“0”分别代表“ON”、“OFF”。

示例：

:PF:OPTion:SCReen ON /*设置通过失败测试失败打开截屏*/

:PF:OPTion:SCReen ? /*查询返回 1*/

:PF:WAVe:FORMat

功能描述：设置或查询通过失败测试保存的波形格式。

指令：

:PF:WAVe:FORMat <format>

:PF:WAVe:FORMat ?

参数：

<format>= 离散型{BINary | CSV}

返回：

查询返回通过失败测试保存的波形格式。

示例：

:PF:WAVe:FORMat BINary /*设置通过失败测试波形保存格式为 Binary*/

:PF:WAVe:FORMat ? /*查询返回 Binary*/

:PF:SCReen:FORMat

功能描述：设置或查询通过失败测试保存的截屏格式。

指令：

:PF:SCReen:FORMat <format>

:PF:SCReen:FORMat ?

参数：

<format>= 离散型{BMP | PNG | JPEG}

返回：

查询返回通过失败测试保存的截屏格式。

示例：

:PF:SCReen:FORMat BMP /*设置通过失败测试截屏格式为 BMP*/

:PF:SCReen:FORMat ? /*查询返回 BMP*/

:PF:FILe:PATH

功能描述：设置或查询通过失败测试结果保存路径。

指令：

:PF:FILe:PATH <path>

:PF:FILe:PATH ?

<path>= 字符串类型，文件保存路径

返回：

查询返回通过失败测试结果保存路径。

示例：

:PF:FILe:PATH "D:\\Picture" /*设置通过失败测试保存路径为 D:\\Picture*/

:PF:FILe:PATH ? /*查询返回 D:\\Picture*/

:PF:FILe:NAMe

功能描述：设置或查询通过失败测试结果保存名称。

指令：

:PF:FILe:NAMe <name>

:PF:FILe:NAMe ?

参数：

<name>= 单位字符串

返回：

查询返回通过失败测试结果保存名称。

示例：

:PF:FILe:NAMe "Uni-t001" /*设置通过失败测试保存名称为 Uni-t001*/

:PF:FILe:NAMe ? /*查询返回 Uni-t001*/

:PF:FILe:NAMe:FORMat

功能描述：设置或查询通过失败测试结果保存名称后缀。

指令：

:PF:FILe:NAMe:FORMat <active>

:PF:FILe:DATetime ?

参数：

<value>= 离散型{ACCumulation | TIMe}

- ACCumulation:累加
- TIMe:时间

返回：

查询返回名称后缀。

示例：

:PF:FILe:NAMe:FORMat TIMe /*设置通过失败测试保存名称后缀为时间*/

:PF:FILe:NAMe:FORMat ? /*查询返回 TIMe*/

:PF:RUN

功能描述：设置或查询通过失败测试运行状态。

指令：

:PF:RUN <run>

:PF:RUN ?

参数：

<run>= 布尔型{ON | OFF}或{1 | 0}

- ON:运行
- OFF:停止

返回：

查询返回通过失败测试运行状态，“1”、“0”分别代表“ON”、“OFF”。

示例：

:PF:RUN ON /*设置通过失败测试运行*/

:PF:RUN ? /*查询返回 1*/

:PF:DATA

功能描述：查询通过失败测试状态表结果。

指令：

:PF:DATA ?

返回：

查询返回通过失败测试结果值，符合数据块格式，并按顺序以逗号按分隔排列，无效值使用实型数据最大值表示。

例如：“#9000000147

Test Status,OFF,

Total Waveform,0,

Total Violations,0,

Testing Time,0 s,

Number of Hits,0,

Hits Per Segment:

1,0

2,0

3,0

4,0

5,0

6,0

7,0”

SEARch 指令系统

用于设置或查询示波器的搜索功能。

:SEARch<n>:ACTive

功能描述：设置或查询指定 search 是否打开。

指令：

:SEARch<n>:ACTive <active>

:SEARch<n>:ACTive ?

参数：

<n>= 整型，范围 0~10

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回指定 search 的打开状态，“1”、“0”分别代表“ON”、“OFF”。

示例：

:SEARch1:ACTive ON /*设置打开 search1*/

:SEARch1:ACTive ? /*查询返回 1*/

:SEARch<n>:SOURce

功能描述：设置或查询指定 search 的数据源。

指令：

:SEARch<n>:SOURce <source>

:SEARch<n>:SOURce ?

参数：

<n>= 整型，范围 0~10

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回指定 search 的数据源。

示例：

:SEARch1:SOURce C1 /*设置 search1 的数据源为 C1 */

:SEARch1:SOURce ? /*查询返回 C1 */

:SEARch<n>:TYPe

功能描述：设置或查询指定 search 的搜索类型。

指令：

:SEARch<n>:TYPe <type>

:SEARch<n>:TYPe ?

参数：

<n>= 整型，范围 0~10

<type>= 离散型{EDGe | PULSe}

- EDGe:边沿
- PULSe:脉宽

返回：

查询返回指定 search 的搜索类型。

示例：

:SEARch1:TYPe EDGe /*设置 search1 的搜索类型为 EDGe */

:SEARch1:TYPe ? /*查询返回 EDGe */

:SEARch<n>:VISible

功能描述：设置或查询指定 search 的标记是否打开。

指令：

:SEARch<n>:VISible <enable>

:SEARch<n>:VISible ?

参数：

<n>= 整型，范围 0~10

<enable>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回指定 search 的标记打开状态，“1”、“0”分别代表“ON”、“OFF”。

示例：

:SEARch1:VISible ON /*设置 search1 的标记打开 */

:SEARch1:VISible ? /*查询返回 1*/

:SEARch<n>:EVENt

功能描述：设置或查询指定 search 的事件列表是否打开。

指令：

:SEARch<n>:EVENt <enable>

:SEARch<n>:EVENt ?

参数：

<n>= 整型，范围 0~10

<enable>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回指定 search 的事件列表打开状态，“1”、“0”分别代表“ON”、“OFF”。

示例：

:SEARch1:EVENt ON /*设置 search1 的事件列表打开*/

:SEARch1:EVENt ? /*查询返回 1*/

:SEARch<n>:EDGe:SLOPe

功能描述：设置或查询指定 search 为边沿搜索时的边沿类型。

指令：

:SEARch<n>:EDGe:SLOPe <slope>

:SEARch<n>:EDGe:SLOPe ?

参数：

<n>= 整型，范围 0~10

<slope>= 离散型{RISe | FALL | ANY}

- RISe:上升沿
- FALL:下降沿
- ANY:任意沿

返回：

查询返回指定 search 为边沿搜索时的边沿类型。

示例：

:SEARch1:EDGe:SLOPe FALL /*设置 search1 边沿搜索的边沿类型为下降沿 */

:SEARch1:EDGe:SLOPe ? /*查询返回 FALL*/

:SEARch<n>:EDGe:LEVel

功能描述：设置或查询指定 search 为边沿搜索时的搜索电平，默认单位 V。

指令：

:SEARch<n>:EDGe:LEVel <level>

:SEARch<n>:EDGe:LEVel ?

参数：

<n>= 整型，范围 0~10

<level>= 实型，与通道垂直刻度单位一致，默认单位为 V

返回：

查询返回指定 search 为边沿搜索时的搜索电平。

示例：

:SEARch1:EDGe:LEVel 100mV /*设置 search1 边沿搜索的搜索电平为 100mV*/

:SEARch1:EDGe:LEVel ? /*查询返回 1.00000E-001*/

:SEARch<n>:PULSe:POLarity

功能描述：设置或查询指定 search 为脉宽搜索时的搜索极性。

指令：

:SEARch<n>:PULSe:POLarity <polarity>

:SEARch<n>:PULSe:POLarity ?

参数：

<n>= 整型，范围 0~10

<polarity>= 离散型{POSitive | NEGative}

- POSitive:正
- NEGative:负

返回：

查询返回指定 search 为脉宽搜索时的搜索极性。

示例：

:SEARch1:PULSe:POLarity NEGative /*设置 search1 脉宽搜索的搜索极性为 NEGative*/

:SEARch1:PULSe:POLarity ? /*查询返回 NEGative*/

:SEARch<n>:PULSe:CONDition

功能描述：设置或查询指定 search 为脉宽搜索时的搜索条件。

指令：

:SEARch<n>:PULSe:CONDition <condition>

:SEARch<n>:PULSe:CONDition ?

参数：

<n>= 整型，范围 0~10

<condition>= 离散型{GREAterthan | LESSthan | RANGe}

- GREAterthan:大于
- LESSthan:小于
- RANGe:范围内

返回：

查询返回指定 search 为脉宽搜索时的搜索条件。

示例：

:SEARch1:PULSe:CONDition LESSthan /*设置 search1 脉宽搜索的搜索条件为 LESSthan*/

:SEARch1:PULSe:CONDition ? /*查询返回 LESSthan*/

:SEARch<n>:PULSe:TIMe:UPPer

功能描述：设置或查询指定 search 为脉宽搜索时的脉冲宽度上限，默认单位 s。

指令：

:SEARch<n>:PULSe:TIMe:UPPer <width>

:SEARch<n>:PULSe:TIMe:UPPer ?

参数：

<n>= 整型，范围 0~10

<width>= 实型，单位 ns、us、ms、s

返回：

查询返回指定 search 为脉宽搜索时的脉冲宽度上限，采用科学计数法。

示例：

:SEARch1:PULSe:TIMe:UPPer 2us /*设置 search1 脉宽搜索的脉冲宽度上限为 2us*/

:SEARch1:PULSe:TIMe:UPPer ? /*查询返回 2.0000000000000E-006*/

:SEARch<n>:PULSe:TIMe:LOWer

功能描述：设置或查询指定 search 为脉宽搜索时的脉冲宽度下限，默认单位 s。

指令：

:SEARch<n>:PULSe:TIMe:LOWer <width>

:SEARch<n>:PULSe:TIMe:LOWer ?

参数：

<n>= 整型，范围 0~10

<width>= 实型，单位 ns、us、ms、s

返回：

查询返回指定 search 为脉宽搜索时的脉冲宽度下限，采用科学计数法。

示例：

:SEARch1:PULSe:TIMe:LOWer 1us /*设置 search1 脉宽搜索的脉冲宽度下限为 1us*/

:SEARch1:PULSe:TIMe:LOWer ? /*查询返回 1.0000000000000E-006*/

:SEARch<n>:PULSe:LEVel

功能描述：设置或查询指定 search 为脉宽搜索时的阈值电平，默认单位 V。

指令：

:SEARch<n>:PULSe:LEVel <level>

:SEARch<n>:PULSe:LEVel ?

参数：

<n>= 整型，范围 0~10

<level>= 实型，与通道垂直刻度单位一致，默认单位为 V

返回：

查询返回指定 search 为脉宽搜索时的阈值电平。

示例：

:SEARch1:PULSe:LEVel 100mV /*设置 search1 脉宽搜索的阈值电平为 100mV*/

:SEARch1:PULSe:LEVel ? /*查询返回 1.00000E-001*/

:SEARch<n>:DATA

功能描述：设查询指定 search 事件列表结果值。

指令：

:SEARch<n>:DATA ?

返回：

查询返回指定 search 事件列表所有结果值，符合数据块格式，以科学计数法表示，并按顺序以逗号按分隔排列，无效值使用实型数据最大值表示。

例如：“#9000000549index,type,position,delta,description,

0,Edge,-2.4000000000000E-006,0.0000000000000E+000,Edge,

1,Edge,-2.3000000000000E-006,1.0000000000000E-007,Edge,

2,Edge,-2.2000000000000E-006,1.0000000000000E-007,Edge,”

POWer 指令系统

用于设置或查询示波器的电源分析功能。

说明：MSO8000HDP 系列无内置高阻，开展电源分析需另购高阻适配器【型号：UT-OA-1M】

:PWAS:ADD

功能描述：添加一个指定类型的电源分析。

指令：

:PWAS:ADD <type>,<vsource>,<csource>

参数：

<type>= 离散型{QUALity | HARMonic | RIPPle | SWITchingloss | SAFeoperationarea | LOOPanalysis | MODulation | INRushcurrent | EFFicency | RDSon | TURNonoff | PSRR | SLEWrate}

- QUALity:电源质量
- HARMonic:谐波分析
- RIPPle:纹波分析
- SWITchingloss:开关损耗
- SAFeoperationarea:安全工作区
- LOOPanalysis:环路分析
- MODulation：调制分析
- INRushcurrent：浪涌电流
- EFFicency：电源效率
- RDSon：导通电阻
- TURNonoff：开启/关闭时间
- PSRR：电源抑制比
- SLEWrate：转换速率

<vsource>=电压源，离散型{C1 | C2 | C3 | C4}

<csource>=电流源，离散型{C1 | C2 | C3 | C4}

示例：

:PWAS:ADD QUALity,C1,C2 /*添加电源质量分析，电压源为 C1，电流源为 C2*/

说明：不支持同时添加多个电源分析项

:PWAS:DELete

功能描述：关闭电源分析。

指令：

:PWAS:DELete

:POWer:MODe

功能描述：查询 POWER 的电源分析类型。

指令：

:POWer:MODe ?

返回：

查询返回 POWER 的电源分析类型。

- QUALity:电源质量
- HARMonic:谐波分析
- RIPPle:纹波分析
- SWITchingloss:开关损耗
- SAFeoperationarea:安全工作区
- LOOPanalysis:环路分析
- MODulation：调制分析
- INRushcurrent：浪涌电流
- EFFicency：电源效率
- RDSon：导通电阻
- TURNonoff：开启/关闭时间
- PSRR：电源抑制比
- SLEWrate：转换速率

:POWer:SOURce:VOLTage

功能描述：设置或查询 POWER 的电压源。

指令：

:POWer:SOURce:VOLTage <source>

:POWer:SOURce:VOLTage ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 POWER 的电压源。

示例：

:POWer:SOURce:VOLTage C1 /*设置 POWER1 的电压源为 C1*/

:POWer:SOURce:VOLTage ? /*查询返回 C1*/

:POWer:SOURce:CURRent

功能描述：设置或查询 POWER 的电流源。

指令：

:POWer:SOURce:CURRent <source>

:POWer:SOURce:CURRent ?

参数：

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 POWER 的电流源。

示例：

:POWer:SOURce:CURRent C2 /*设置 POWER1 的电流源为 C2*/

:POWer:SOURce:CURRent ? /*查询返回 C2*/

:POWer:DATA

功能描述：查询 POWER 的测量结果值。

指令：

:POWer:DATA ?

返回：

查询返回 POWER 的测量结果值，符合数据块格式，以科学计数法表示，并按顺序以逗号按分隔排列，无效值使用实型数据最大值表示。

例如：“#9000000564QUALity,

Value,Average,Maxumum,Minimum,

1.78612E-003,1.99920E-003,2.47743E-003,1.53923E-003,

1.95003E+000,2.51101E+000,4.00265E+000,1.22652E+000,

1.79769E+308,1.79769E+308,1.79769E+308,1.79769E+308,

1.78612E-003,1.99920E-003,2.47743E-003,1.53923E-003,
1.95003E+000,2.51199E+000,4.00265E+000,1.22652E+000,
4.02983E-006,4.00127E-006,4.18383E-006,3.83827E-006,
3.19021E-006,4.01689E-006,6.13767E-006,2.36924E-006,
NaN,1.90541E-006,4.68634E-006,8.00995E-008,
1.26319E+000,1.01649E+000,1.65799E+000,6.45764E-001,
NaN,2.47287E+001,4.97770E+001,1.16113E+000,”

电源质量

:QUALity:FREQuency:REFerence

功能描述：设置或查询指定电源质量分析的频率参考。

指令：

:QUALity:FREQuency:REFerence <type>

:QUALity:FREQuency:REFerence ?

参数：

<type>= 离散型{VOLTage | CURRent}

- VOLTage:电压源
- CURRent:电流源

返回：

查询返回电源质量分析的频率参考。

示例：

:QUALity:FREQuency:REFerence CURRent /*设置 QUALity 的频率参考为电流源*/

:QUALity:FREQuency:REFerence ? /*查询返回 CURRent*/

谐波分析

:HARMonic:COUNt

功能描述：设置或查询指定谐波分析的谐波个数。

指令：

:HARMonic:COUNt <value>

:HARMonic:COUNt ?

参数：

<type>= 整数型，范围 40~200

返回：

查询返回谐波分析的谐波个数。

示例：

:HARMonic:COUNt 50 /*设置 HARMonic1 的谐波个数为 50*/

:HARMonic:COUNt ? /*查询返回 50*/

:HARMonic:FREQuency:REFerence

功能描述：设置或查询谐波分析的频率参考。

指令：

:HARMonic:FREQuency:REFerence <type>

:HARMonic:FREQuency:REFerence ?

参数：

<type>= 离散型{HARMonicsrc | VOLTage | CURRent | CONStant}

- HARMonicsrc:谐波源
- VOLTage:电压源
- CURRent:电流源
- CONStant:固定

返回：

查询返回谐波分析的频率参考。

示例：

:HARMonic:FREQuency:REFerence CURRent /*设置 HARMonic1 的频率参考为电流源*/

:HARMonic:FREQuency:REFerence ? /*查询返回 CURRent*/

:HARMonic:FREQuency:CONStant

功能描述：设置或查询谐波分析的频率参考为固定时的频率，默认单位 Hz。

指令：

:HARMonic:FREQuency:CONStant <value>

:HARMonic:FREQuency:CONStant ?

参数：

<value>= 实型，单位 Hz、KHz，可设置范围为 10Hz~100kHz。

返回：

查询返回谐波分析的频率参考为固定时的频率，以科学计数法返回。

示例：

:HARMonic:FREQuency:CONStant 120 /*设置 HARMonic1 的固定频率参考为 120Hz*/

:HARMonic:FREQuency:CONStant ? /*查询返回 1.2000000000000E+002*/

:HARMonic:SOURce

功能描述：设置或查询谐波分析的谐波源。

指令：

:HARMonic:SOURce <source>

:HARMonic:SOURce ?

参数：

<source>= 离散型{VOLTage | CURRent}

- VOLTage:电压
- CURRent:电流

返回：

查询返回谐波分析的谐波源。

示例：

:HARMonic:SOURce CURRent /*设置 HARMonic1 的谐波源为电流*/

:HARMonic:SOURce ? /*查询返回 CURRent*/

:HARMonic:TYPe

功能描述：设置或查询指定电源谐波分析的谐波类型。

指令：

:HARMonic:TYPe <value>

:HARMonic:TYPe ?

参数：

<value>= {ODD | EVEN | ALL}

- ODD:奇次
- EVEN:偶次
- ALL:全部

返回：

查询返回指定电源谐波分析的谐波类型。

示例：

:HARMonic:TYPe ALL /*设置电源谐波分析谐波类型为全部*/

:HARMonic:TYPe ? /*查询返回 ALL*/

安全工作区

:SAFeoperationarea:STOPonfail

功能描述：设置或查询安全工作区违例时是否停止。

指令：

:SAFeoperationarea:STOPonfail <active>

:SAFeoperationarea:STOPonfail ?

参数：

<active>= 布尔型{ON | OFF}或{1 | 0}

- ON:停止
- OFF:不停止

返回：

查询返回安全工作区违例时是否停止，“1”、“0”分别代表“ON”、“OFF”。

示例：

:SAFeoperationarea:STOPonfail ON /*设置 SAFeoperationarea1 违例时停止*/

:SAFeoperationarea:STOPonfail ? /*查询返回 1*/

:SAFeoperationarea:AXIS:XMAX

功能描述：设置或查询安全工作区 X 轴最大值，默认单位 V。

指令：

:SAFeoperationarea:AXIS:XMAX <value>

:SAFeoperationarea:AXIS:XMAX ?

参数：

<value>= 实型，范围-3.99V~10kV

返回：

查询返回安全工作区 X 轴最大值。

示例：

:SAFeoperationarea:AXIS:XMAX 5 /*设置 SAFeoperationarea1 的 X 轴最大值为 5V*/

:SAFeoperationarea:AXIS:XMAX ? /*查询返回 5.00000E+000*/

:SAFeoperationarea:AXIS:XMIN

功能描述：设置或查询安全工作区 X 轴最小值，默认单位 V。

指令：

:SAFeoperationarea:AXIS:XMIN <value>

:SAFeoperationarea:AXIS:XMIN ?

参数：

<value>= 实型，范围-10kV~7.99V

返回：

查询返回安全工作区 X 轴最小值。

示例：

:SAFeoperationarea:AXIS:XMIN 1 /*设置 SAFeoperationarea1 的 X 轴最小值为 1V*/

:SAFeoperationarea:AXIS:XMIN ? /*查询返回 1.00000E+000*/

:SAFeoperationarea:AXIS:YMAX

功能描述：设置或查询安全工作区 Y 轴最大值，默认单位 A。

指令：

:SAFeoperationarea:AXIS:YMAX <value>

:SAFeoperationarea:AXIS:YMAX ?

参数：

<value>= 实型，范围 10mA~10kA

返回：

查询返回安全工作区 Y 轴最大值。

示例：

:SAFeoperationarea:AXIS:YMAX 5 /*设置 SAFeoperationarea1 的 Y 轴最大值为 5A*/

:SAFeoperationarea:AXIS:YMAX ? /*查询返回 5.00000E+000*/

:SAFeoperationarea:AXIS:YMIN

功能描述：设置或查询安全工作区 Y 轴最小值，默认单位 A。

指令：

:SAFeoperationarea:AXIS:YMIN <value>

:SAFeoperationarea:AXIS:YMIN ?

参数：

<value>=实型，范围-10kA~5.29A

返回：

查询返回安全工作区 Y 轴最小值。

示例：

:SAFeoperationarea:AXIS:YMIN 1 /*设置 SAFeoperationarea1 的 Y 轴最小值为 1A*/

:SAFeoperationarea:AXIS:YMIN ? /*查询返回 1.00000E+000*/

环路分析

:LOOPanalysis:SOURce:AWG

功能描述：设置或查询环路分析的 AWG 源。

指令：

:LOOPanalysis:SOURce:AWG <source>

:LOOPanalysis:SOURce:AWG ?

参数：

<source>= 离散型{G1 | G2}

返回：

查询返回环路分析的 AWG 源。

示例：

:LOOPanalysis:SOURce:AWG G1 /*设置 LOOPanalysis1 的 AWG 源为 G1 */

:LOOPanalysis:SOURce:AWG ? /*查询返回 G1*/

:LOOPanalysis:IMPedance

功能描述：设置或查询环路分析的阻抗。

指令：

:LOOPanalysis:IMPedance <type>

:LOOPanalysis:IMPedance ?

参数：

<type>= 离散型{LOW | HIGH}

- LOW:50Ω

- HIGH:高阻

返回：

查询返回环路分析的阻抗。

示例：

:LOOPanalysis:IMPedance HIGH /*设置 LOOPanalysis1 阻抗为高阻 */

:LOOPanalysis:IMPedance ? /*查询返回 HIGH */

:LOOPanalysis:FREQuency:MODe

功能描述：设置或查询指定电源环路分析的频率模式。

指令：

:LOOPanalysis:FREQuency:MODe <mode>

:LOOPanalysis:FREQuency ?

参数：

<mode>= 离散型{CONTinuous | SINGle}

- CONTinuous:连续
- SINGle:单次

返回：

查询返回指定电源环路分析的频率模式。

示例：

:LOOPanalysis:FREQuency:MODe SINGle /*设置电源环路分析的频率模式为单次*/

:LOOPanalysis:FREQuency:MODe ? /*查询返回 SINGle*/

:LOOPanalysis:FREQuency:STARt

功能描述：设置或查询环路分析的开始频率，默认单位 Hz。

指令：

:LOOPanalysis:FREQuency:STARt <value>

:LOOPanalysis:FREQuency:STARt ?

参数：

<value>= 实型，单位 Hz、kHz、MHz

返回：

查询返回环路分析的开始频率，以科学计数法返回。

示例：

:LOOPanalysis:FREQuency:STARt 100 /*设置 LOOPanalysis1 的开始频率为 100Hz*/

:LOOPanalysis:FREQuency:STARt ? /*查询返回 1.0000000000000E+002*/

:LOOPanalysis:FREQuency:STOP

功能描述：设置或查询环路分析的停止频率，默认单位 Hz。

指令：

:LOOPanalysis:FREQuency:STOP <value>

:LOOPanalysis:FREQuency:STOP ?

参数：

<value>= 实型，单位 Hz、kHz、MHz

返回：

查询返回环路分析的停止频率，以科学计数法返回。

示例：

:LOOPanalysis:FREQuency:STOP 1MHz /*设置 LOOPanalysis1 的停止频率为 1MHz */

:LOOPanalysis:FREQuency:STOP ? /*查询返回 1.0000000000000E+006*/

:LOOPanalysis:SCANnum

功能描述：设置或查询环路分析的扫描点数。

指令：

:LOOPanalysis:SCANnum <value>

:LOOPanalysis:SCANnum ?

参数：

<value>= 整数型，范围 1~1k

返回：

查询返回环路分析的扫描点数。

示例：

:LOOPanalysis:SCANnum 30 /*设置 LOOPanalysis1 的扫描点数为 30 */

:LOOPanalysis:SCANnum ? /*查询返回 30*/

:LOOPanalysis:AMPMode

功能描述：设置或查询环路分析的幅度模式。

指令：

:LOOPanalysis:AMPMode <mode>

:LOOPanalysis:AMPMode ?

参数：

<mode>= 离散型{CONStant | VARiable}

- CONStant:恒定
- VARiable:可变

返回：

查询返回环路分析的幅度模式。

示例：

:LOOPanalysis:AMPMode CONStant /*设置 LOOPanalysis1 的幅度模式为恒定*/

:LOOPanalysis:AMPMode ? /*查询返回 CONStant*/

:LOOPanalysis:AMPValue

功能描述：设置或查询环路分析的幅度，默认单位 V。

指令：

:LOOPanalysis:AMPValue <value>

:LOOPanalysis:AMPValue ?

参数：

<value>= 实型，单位 mV、V

返回：

查询返回环路分析的幅度，以科学计数法返回。

示例：

:LOOPanalysis:AMPValue 1 /*设置 LOOPanalysis1 的幅度为 1V */

:LOOPanalysis:AMPValue ? /*查询返回 1.00000E+000*/

:LOOPanalysis:TRIGStatus

功能描述：设置或查询指定电源环路分析的触发状态检查。

指令：

:LOOPanalysis:TRIGStatus <check>

:LOOPanalysis:TRIGStatus ?

参数：

<check>= 布尔型{ON | OFF}或{1 | 0}

- OFF:不检测

- ON:检测

返回：

查询返回指定电源环路分析的触发状态检查。

示例：

:LOOPanalysis:TRIGStatus ON /*设置电源环路分析触发状态检查为检查*/

:LOOPanalysis:TRIGStatus ? /*查询返回 1*/

调制分析

:MODulation:SOURce

功能描述：设置或查询指定调制分析信源类型。

指令：

:MODulation:SOURce <type>

:MODulation:SOURce ?

参数：

<type>= 离散型{VOLTage | CURRent}

- VOLTage:电压源
- CURRent:电流源

返回：

查询返回指定调制分析信源类型。

示例：

:MODulation:SOURce VOLTage /*设置 MODUlation1 的调制信源为 VOLTage */

:MODulation:SOURce ? /*查询返回 VOLTage */

:MODulation:HISTogram:TYPe

功能描述：设置或查询调制分析直方图的类型。

指令：

:MODulation:HISTogram:TYPe <type>

:MODulation:HISTogram:TYPe ?

参数：

<type>= 离散型{PERiod | FREQuency | PDUTy | NDUTy | PWIDth | NWIDth | RISEtime | FALLtime}

- PERiod:周期

- FREQuency:频率
- PDUTy:正占空比
- NDUTy:负占空比
- PWIDth:正脉宽
- NWIDth:负脉宽
- RISEtime:上升时间
- FALLtime:下降时间

返回：

查询返回指定调制分析的直方图类型。

示例：

:MODulation:HISTogram:TYPe PERiod /*设置 MODUlation1 的调制信源为 PERiod*/

:MODulation:HISTogram:TYPe ? /*查询返回 PERiod */

:MODulation:TRENd:TYPe

功能描述：设置或查询调制分析趋势图的类型。

指令：

:MODulation:TRENd:TYPe <type>

:MODulation:TRENd:TYPe ?

参数：

<type>= 离散型{PERiod | FREQuency | PDuty | NDuty | PWidth | NWidth | RISEtime | FALLtime}

- PERiod:周期
- FREQuency:频率
- PDuty:正占空比
- NDuty:负占空比
- PWidth:正脉宽
- NWidth:负脉宽
- RISEtime:上升时间
- FALLtime:下降时间

返回：

查询返回调制分析的趋势图类型。

示例：

:MODulation:TRENd:TYPe PERiod /*设置 MODUlation1 的调制信源为 PERiod*/

:MODulation:TRENd:TYPe ? /*查询返回 PERiod */

:MODulation:HISTogram:ADD

功能描述：设置调制分析打开直方图。

指令：

:MODulation:HISTogram:ADD

参数：

<n>= 整型，范围 1~4

示例：

:MODulation:HISTogram:ADD /*打开直方图*/

:MODulation:TRENd:ADD

功能描述：设置调制分析打开趋势图。

指令：

:MODulation:TRENd:ADD

示例：

:MODulation:TRENd:ADD /*打开趋势图*/

浪涌电流

:INRUsh:PEAKcurrent

功能描述：设置或查询浪涌电流的峰值电流值。

指令：

:INRUsh:PEAKcurrent <value>

:INRUsh:PEAKcurrent ?

参数：

<value>= 实型 范围：-55V-55V 默认单位 V。

返回：

查询返回浪涌电流的峰值电流值。

示例：

:INRUsh:PEAKcurrent 1 /*设置浪涌电流的峰值电流值为 1V */

:INRUsh:PEAKcurrent ? /*查询返回 1.00000E+000*/

:INRUsh:RUN

功能描述：运行浪涌电流。

指令：

:INRUsh:RUN

示例：

:INRUsh:RUN /*运行浪涌电流 */

电源效率

:EFFIciency:INPut:VOLTage

功能描述：设置或查询电源效率的输入电压源。

指令：

:EFFIciency:INPut:VOLTage <type>

:EFFIciency:INPut:VOLTage ?

参数：

<type>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回电源效率的输入电压源。

示例：

:EFFIciency:INPut:VOLTage C1 /*设置电源效率的输入电压源 C1*/

:EFFIciency:INPut:VOLTage ? /*查询返回 C1 */

:EFFIciency:INPut:CURRent

功能描述：设置或查询电源效率的输入电流源。

指令：

:EFFIciency:INPut:CURRent <type>

:EFFIciency:INPut:CURRent ?

参数：

<type>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回电源效率的输入电流源。

示例：

:EFFIciency:INPut:CURRent C2 /*设置电源效率的输入电流源 C2*/

:EFFIciency:INPut:CURRent ? /*查询返回 C2*/

:EFFIciency:OUTput:VOLTage

功能描述：设置或查询电源效率的输出电压源。

指令：

:EFFIciency:OUTput:VOLTage <type>

:EFFIciency:OUTput:VOLTage ?

参数：

<type>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回电源效率的输出电压源。

示例：

:EFFIciency:OUTput:VOLTage C3 /*设置电源效率的输出电压源 C3*/

:EFFIciency:OUTput:VOLTage ? /*查询返回 C3*/

:EFFIciency:OUTput:CURRent

功能描述：设置或查询电源效率的输出电流源。

指令：

:EFFIciency:OUTput:CURRent <type>

:EFFIciency:OUTput:CURRent ?

参数：

<type>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回电源效率的输出电流源。

示例：

:EFFIciency:OUTput:CURRent C4 /*设置电源效率的输出电流源 C4*/

:EFFIciency:OUTput:CURRent ? /*查询返回 C4*/

:EFFIciency:INPut:DIAGram

功能描述：打开输入功率图。

指令：

:EFFIciency:INPut:DIAGram

示例：

:EFFIciency:INPut:DIAGram /*打开输入功率图*/

:EFFIciency:OUTput:DIAGram

功能描述：打开输出功率图。

指令：

:EFFIciency:OUTput:DIAGram

示例：

:EFFIciency:OUTput:DIAGram /*打开输出功率图*/

Rds（on）

:RDS:WAVEform

功能描述：打开 RDS 波形图。

指令：

:RDS:WAVEform

示例：

:RDS:WAVEform /*打开 RDS 波形图*/

开启/关闭时间

:TIMEonoff:INPut:VOLTage

功能描述：设置或查询开启/关闭时间的输入峰值电压值。

指令：

:TIMEonoff:INPut:VOLTage <value>

:TIMEonoff:INPut:VOLTage ?

参数：

<value>= 实型 范围 -55V-55V 单位 mV、V

返回：

查询返回开启/关闭时间的输入峰值电压值。

示例：

:TIMEonoff:INPut:VOLTage 1 /*设置开启/关闭时间的输入峰值电压值为 1V*/

:TIMEonoff:INPut:VOLTage ? /*查询返回 1.00000E+000*/

:TIMEonoff:OUTput:VOLTage

功能描述：设置或查询指定开启/关闭时间的输出峰峰值。

指令：

:TIMEonoff:OUTput:VOLTage <value>

:TIMEonoff:OUTput:VOLTage ?

参数：

<value>= 实型

返回：

查询返回指定开启/关闭时间的输出峰峰值。

示例：

:TIMEonoff:OUTput:VOLTage 5 /*设置开启/关闭时间的输出峰峰值为 5V*/

:TIMEonoff:OUTput:VOLTage ? /*查询返回 5.00000E+000*/

:TIMEonoff:TESTtype

功能描述：设置或查询开启/关闭时间的测试类型。

指令：

:TIMEonoff:TESTtype <type>

:TIMEonoff:TESTtype ?

参数：

<Type>= 离散型 {TRON | TROFF}

- TRON:开启时间
- TROFF:关闭时间

返回：

查询返回开启/关闭时间的测试类型。

示例：

:TIMEonoff:TESTtype TRON /*设置开启/关闭时间的测试类型为开启时间*/

:TIMEonoff:TESTtype ? /*查询返回 TRON */

:TIMEonoff:Typeconvert

功能描述：设置或查询开启/关闭时间的转换类型。

指令：

:TIMEonoff:Typeconvert <type>

:TIMEonoff:Typeconvert ?

参数：

<Type>= 离散型{DCDC | ACDC}

- DCDC:直流/直流
- ACDC:交流/直流

返回：

查询返回开启/关闭时间的转换类型。

示例：

:TIMEonoff:Typeconvert DCDC /*设置开启/关闭时间的转换类型为直流/直流*/

:TIMEonoff:Typeconvert ? /*查询返回 DCDC */

:TIMEonoff:ACQTime

功能描述：设置或查询开启/关闭时间的采集时间值。

指令：

:TIMEonoff:ACQTime <value>

:TIMEonoff:ACQTime ?

参数：

<value>= 实型 200ms-1000s 单位 ms,s

返回：

查询返回开启/关闭时间的采集时间值,以科学计数法返回。

示例：

:TIMEonoff:ACQTime 1 /*设置开启/关闭时间的采集时间值为 1s*/

:TIMEonoff:ACQTime ? /*查询返回 1.0000000000000E+000 */

PSRR

:PSRR:AWGSource

功能描述：设置或查询 PSRR 的 AWG 源。

指令：

:PSRR:AWGSource <source>

:PSRR:AWGSource ?

参数：

<source>= 离散型{G1 | G2}

返回：

查询返回 PSRR 的 AWG 源。

示例：

:PSRR:AWGSource G1 /*设置 PSRR 的 AWG 源为 G1 */

:PSRR:AWGSource ? /*查询返回 G1 */

:PSRR:IMPedance

功能描述：设置或查询 PSRR 的阻抗类型。

指令：

:PSRR:IMPedance <type>

:PSRR:IMPedance ?

参数：

<type>= 离散型{LOW | HIGH}

返回：

查询返回 PSRR 的阻抗类型。

示例：

:PSRR:IMPedance LOW /*设置 PSRR 的 AWG 源为 50Ω*/

:PSRR:IMPedance ? /*查询返回 LOW*/

:PSRR:FREQuency:MODe

功能描述：设置或查询指定 PSRR 的频率模式。

指令：

:PSRR:FREQuency:MODe <mode>

:PSRR:FREQuency:MODe ?

参数：

<mode>= 离散型 {CONTinuous | SINGle}

- CONTinuous:连续

- SINGle:单次

返回：

查询返回指定 PSRR 的频率模式。

示例：

:PSRR:FREQuency:MODe SINGle /*设置 PSRR 的频率模式为单次*/

:PSRR:FREQuency:MODe ? /*查询返回 SINGle*/

:PSRR:TRIGStatus

功能描述：设置或查询指定 PSRR 的触发状态检查。

指令：

:PSRR:TRIGStatus <check>

:PSRR::TRIGStatus ?

参数：

<check>= 布尔型{ON | OFF}或{1 | 0}

- ON:检测
- OFF:不检测

返回：

查询返回指定 PSRR 的触发状态检查。

示例：

:PSRR:TRIGStatus ON /*设置 PSRR 的触发状态检查为检测*/

:PSRR:TRIGStatus ? /*查询返回 1*/

:PSRR:FREQuency:STARt

功能描述：设置或查询 PSRR 的开始频率值。

指令：

:PSRR:FREQuency:STARt <value>

:PSRR:FREQuency:STARt ?

参数：

<value>= 实型 单位 mHz、Hz

返回：

查询返回指定 PSRR 的开始频率。

示例：

:PSRR:FREQuency:STARt 100 /*设置 PSRR 的开始频率为 100Hz*/

:PSRR:FREQuency:STARt ? /*查询返回 1.0000000000000E+002*/

:PSRR::FREQuency:STOP

功能描述：设置或查询 PSRR 的停止频率值。

指令：

:PSRR:FREQuency:STOP <value>

:PSRR:FREQuency:STOP ?

参数：

<value>= 实型 范围 单位 mHz、Hz

返回：

查询返回 PSRR 的停止频率。

示例：

:PSRR:FREQuency:STOP 1000_000 /*设置 PSRR 的停止频率为 1MHz*/

:PSRR:ENDFrequency ? /*查询返回 1.0000000000000E+006*/

:PSRR:SCANnumber

功能描述：设置或查询 PSRR 的扫描点个数。

指令：

:PSRR:SCANnumber <value>

:PSRR:SCANnumber ?

参数：

<value>= 实型 范围 1-1K

返回：

查询返回 PSRR 的扫描点个数。

示例：

:PSRR:SCANnumber 10 /*设置 PSRR 的扫描点个数为 10*/

:PSRR:SCANnumber ? /*查询返回 10 *

:PSRR:AMPMode

功能描述：设置或查询 PSRR 的幅度模式。

指令：

:PSRR:AMPMode <mode>

:PSRR:AMPMode ?

参数：

<mode>= 离散型{CONStant | VARiable}

- CONStant:恒定
- VARiable:可变

返回：

查询返回 PSRR 的幅度模式。

示例：

:PSRR:AMPMode CONStant /*设置 PSRR 的幅度模式为恒定*/

:PSRR:AMPMode ? /*查询返回 CONStant*/

:PSRR:AMPLitude

功能描述：设置或查询 PSRR 的幅度，默认单位 mV。

指令：

:PSRR:AMPLitude <value>

:PSRR:AMPLitude ?

参数：

<value>=实型，单位 V

返回：

查询返回 PSRR 的幅度，以科学计数法返回。

示例：

:PSRR:AMPLitude 1 /*设置 PSRR 的幅度为 1V*/

:PSRR:AMPLitude ? /*查询返回 1.00000E+000*/

:PSRR:BODepicture

功能描述：打开 PSRR 波特图。

指令：

:PSRR:BODE

示例：

:PSRR:BODE /*打开 PSRR 波特图*/

:PSRR:RUN

功能描述：运行 PSRR。

指令：

:PSRR:RUN

示例：

:PSRR:RUN /*运行 PSRR*/

转换速率

:SLEWrate:DIAgram:DVDT

功能描述：打开转换速率 dv/dt 图。

指令：

:SLEWrate:DIAgram:DVDT

示例：

:SLEWrate:DIAgram:DVDT /*打开转换速率 dv/dt 图*/

:SLEWrate:DIAgram:DIDT

功能描述：打开转换速率 di/dt 图。

指令：

:SLEWrate:DIAgram:DIDT

示例：

:SLEWrate:DIAgram:DIDT /*打开转换速率 di/dt 图*/

REF 指令系统

用于设置或查询示波器的参考通道功能。

:REFerence<n>:ACTive

功能描述：设置或查询指定参考通道是否打开。

指令：

:REFerence<n>:ACTive <active>，path

:REFerence<n>:ACTive ?

参数：

<n>= 整型，范围 0~4

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回指定参考通道的打开状态，“1”、“0”分别代表“ON”、“OFF”。

示例：

:REFerence1:ACTive ON,"C:\\Users\\Administrator\\Desktop\\Uni-t000.bin" /*设置打开 REFerence1*/

:REFerence1:ACTive ? /*查询返回 1*/

:REFerence<n>:LABel:ENABle

功能描述：设置或查询打开或关闭指定参考通道标签显示。

指令：

:REFerence<n>:LABel:ENABle <enable>

:REFerence<n>:LABel:ENABle ?

参数：

<enable>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:REFerence1:LABel:ENABle ON /*设置参考通道 1 标签显示打开*/

:REFerence1:LABel:ENABle ? /*查询返回 1*/

:REFerence<n>:LABel

功能描述：设置或查询指定参考通道的标签。

指令：

:REFerence<n>:LABel <lable>

:REFerence<n>:LABel ?

参数：

<n>= 整型，范围 0~4

<lable>= 字符串类型

返回：

查询返回指定参考通道的标签。

示例：

:REFerence1:LABel "QWER" /*设置 REF1 标签为 QWER*/

:REFerence1:LABel ? /*查询返回 QWER */

:REFerence<n>:VERTical:OFFSet

功能描述：设置或查询指定参考通道的垂直位移。

指令：

:REFerence<n>:VERTical:OFFSet <offset>

:REFerence<n>:VERTical:OFFSet ?

参数：

<n>= 整型，范围 0~4

<offset>= 实型，单位为 div，范围为-4div-4div

返回：

查询返回指定参考通道的垂直位移，采用科学计数法。

示例：

:REFerence1:VERTical:OFFSet 1.2 /*设置 REF1 垂直位移为 1.2*/

:REFerence1:VERTical:OFFSet ? /*查询返回 1.20000E+000*/

:REFerence<n>:VERTical:SCALe

功能描述：设置或查询指定参考通道的垂直档位。

指令：

:REFerence<n>:VERTical:SCALe <scale>

:REFerence<n>:VERTical:SCALe ?

参数：

<n>= 整型，范围 0~4

<scale>= 实型，单位为保存波形时的单位

返回：

查询返回指定参考通道的垂直档位，采用科学计数法。

示例：

:REFerence1:VERTical:SCALe 1 /*设置 REF1 垂直档位为 1*/

:REFerence1:VERTical:SCALe ? /*查询返回 1.00000E+000*/

:REFerence<n>:HORizontal:OFFSet

功能描述：设置或查询指定参考通道的水平位移。

指令：

:REFerence<n>:HORizontal:OFFSet <offset>

:REFerence<n>:HORizontal:OFFSet ?

参数：

<n>= 整型，范围 0~4

<offset>= 实型，单位为 div，范围为 0div-10div

返回：

查询返回指定参考通道的水平位移，采用科学计数法。

示例：

:REFerence1:HORizontal:OFFSet 1.2 /*设置 REF1 水平位移为 1.2*/

:REFerence1:HORizontal:OFFSet ? /*查询返回 1.20000E+000*/

:REFerence<n>:HORizontal:SCALe

功能描述：设置或查询指定参考通道的水平档位。

指令：

:REFerence<n>:HORizontal:SCALe <scale>

:REFerence<n>:HORizontal:SCALe ?

参数：

<n>= 整型，范围 0~4

<scale>= 实型，单位:us、ms、s

返回：

查询返回指定参考通道的水平档位，采用科学计数法。

示例：

:REFerence1:HORizontal:SCALe 1 /*设置 REF1 水平档位为 1*/

:REFerence1:HORizontal:SCALe ? /*查询返回 1.0000000000000E+000*/

:REFerence<n>:FINe

功能描述：设置或查询参考通道细调。

指令：

:REFerence<n>:FINe <value>

:REFerence<n>:FINe *?*

参数：

<value>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回指定参考通道细调打开状态。

示例：

:REFerenceR1:FINe ON /*设置 R1 打开细调*/

:REFerenceR1:FINe ? /*查询返回 1*/

BUS 指令系统

用于设置或查询示波器的解码功能。

:BUS<n>:ACTive

功能描述：设置或查询 BUS 是否打开。

指令：

:BUS<n>:ACTive <active>

:BUS<n>:ACTive ?

参数：

<n>= 整型，范围 1~2

<active>= 布尔型{ON | OFF}或{1 | 0}

返回：

查询返回指定 BUS 的打开状态，“1”、“0”分别代表“ON”、“OFF”。

示例：

:BUS1:ACTive 1 /*设置 BUS1 打开*/

:BUS1:ACTive ? /*查询返回 1*/

:BUS<n>:TYPe

功能描述：设置或查询指定 BUS 的解码类型。

指令：

:BUS<n>:TYPe <type>

:BUS<n>:TYPe ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{CLOSe | RS232 | I2C | SPI | CAN | CANFd | LIN | FLEXray | AUDioBus | MIL | ARINc429 USB | SENT | SPMi | ETHernet | NRZ | MANChester | COMMon8b10b | PSI5 | SMBus}

返回：

查询返回指定 BUS 的解码类型。

示例：

:BUS1:TYPe RS232 /*设置 BUS1 的解码类型为 RS232 */

:BUS1:TYPe ? /*查询返回 RS232*/

:BUS<n>:FORMat

功能描述：设置或查询指定 BUS 的解码显示格式。

指令：

:BUS<n>:FORMat <type>

:BUS<n>:FORMat ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{HEX | DEC | BINary | ASCII | AUTo}

返回：

查询返回指定 BUS 的解码显示格式。

示例：

:BUS1:FORMat HEX /*设置 BUS1 的解码显示格式为 HEX*/

:BUS1:FORMat ? /*查询返回 HEX*/

:BUS<n>:POSition

功能描述：设置或查询指定 BUS 的解码显示位置，默认单位 div。

指令：

:BUS<n>:POSition <value>

:BUS<n>:POSition ?

参数：

<n>= 整型，范围 1~2

<value>= 实型，-4div ~ 4div

返回：

查询返回指定 BUS 的解码显示位置，以科学计数法表示。

示例：

:BUS1:POSition 2 /*设置 BUS1 的解码显示位置为 2div*/

:BUS1:POSition ? /*查询返回 2.00000E+000*/

:BUS<n>:LABel:ENABle

功能描述：设置或查询打开或关闭指定总线通道标签显示。

指令：

:BUS<n>:LABel:ENABle <enable>

:BUS<n>:LABel:ENABle ?

参数：

<enable>=布尔型{ON | OFF}或{1 | 0}

返回：

查询返回“1”、“0”分别代表“ON”、“OFF”。

示例：

:BUS1:LABel:ENABle ON /*设置总线通道 1 标签显示打开*/

:BUS1:LABel:ENABle ? /*查询返回 1*/

:BUS<n>:LABel

功能描述：设置或查询指定 BUS 的标签。

指令：

:BUS<n>:LABel <lable>

:BUS<n>:LABel ?

参数：

<n>= 整型，范围 1~2

<lable>= 字符串类型

返回：

查询返回指定 BUS 的标签。

示例：

:BUS1:LABel "abcd" /*设置 BUS1 的标签为 abcd*/

:BUS1:LABel ? /*查询返回 abcd*/

:BUS<n>:DATA

功能描述：查询指定 BUS 的事件列表结果。

指令：

:BUS<n>:DATA ?

参数：

<n>= 整型，范围 1~2

返回：

查询返回指定 BUS 事件列表所有结果值，符合数据块格式，以科学计数法表示，并按顺序以逗号按分隔排列，无效值使用实型数据最大值表示。以 CAN 类型举例的返回格式如下：

“Index,Start Time,StandardID,ExpandID,DLC,Data,CRC,ACK,EOF,ERROR”

例如：“#9000000272CAN,

Index,Start Time,StandardID,ExpandID,DLC,Data,CRC,ACK,EOF,ERROR

1,-1.89960E-004,NaN,00 02 73 E6,0F,NaN,NaN,NaN,NaN,NaN

2,2.10040E-004,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN

3,3.10040E-004,01 8F,NaN,00,NaN,30 67,01,03,CRC

4,5.60040E-004,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN”

:BUS<n>:RS232

:BUS<n>:RS232:SOURce

功能描述：设置或查询 RS232 的解码信源。

指令：

:BUS<n>:RS232:SOURce <source>

:BUS<n>:RS232:SOURce ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 RS232 的解码信源。

示例：

:BUS1:RS232:SOURce C1 /*设置 BUS1 的 RS232 源为 C1*/

:BUS1:RS232:SOURce ? /*查询返回 C1*/

:BUS<n>:RS232:BAUDrate:CURRent

功能描述：设置或查询 RS232 的解码波特率，默认单位 bps。

指令：

:BUS<n>:RS232:BAUDrate:CURRent <value>

:BUS<n>:RS232:BAUDrate ?

参数：

<n>= 整型，范围 1~2

<value>= 离散型{CUSTom | 2400 | 4800 | 9600 | 192000 | 38400 | 57600 | 115200}

返回：

查询返回 RS232 的解码波特率。

示例：

:BUS1:RS232:BAUDrate:CURRent 115200 /*设置 BUS1 的 RS232 波特率为 115200bps*/

:BUS1:RS232:BAUDrate:CURRent ? /*查询返回 115200*/

:BUS<n>:RS232:BAUdrate:CUSTom

功能描述：设置或查询 RS232 的解码自定义波特率，默认单位 bps。

指令：

:BUS<n>:RS232:BAUdrate:CUSTom <value>

:BUS<n>:RS232:BAUdrate:CUSTom ?

参数：

<n>= 整型，范围 1~2

<value>= 整数型，范围 1kbps~1Gbps

返回：

查询返回 RS232 的解码自定义波特率。

示例：

:BUS1:RS232:BAUdrate:CUSTom 5500 /*设置 BUS1 的 RS232 自定义波特率为 5500bps*/

:BUS1:RS232:BAUdrate:CUSTom ? /*查询返回 5.50000E+003*/

:BUS<n>:RS232:LEVel

功能描述：设置或查询 RS232 的解码阈值，默认单位 V。

指令：

:BUS<n>:RS232:LEVel <value>

:BUS<n>:RS232:LEVel ?

参数：

<n>= 整型，范围 1~2

<value>= 实型，单位与通道垂直刻度单位一致，默认单位为 V

返回：

查询返回 RS232 的解码阈值。

示例：

:BUS1:RS232:LEVel 1 /*设置 BUS1 的 RS232 解码阈值为 1V*/

:BUS1:RS232:LEVel ? /*查询返回 1.00000E+000*/

:BUS<n>:RS232:ORDer

功能描述：设置或查询 RS232 的解码位顺序。

指令：

:BUS<n>:RS232:ORDer <order>

:BUS<n>:RS232:ORDer ?

参数：

<n>= 整型，范围 1~2

<order>= 离散型{MSB | LSB}

- MSB:最高有效位
- LSB:最低有效位

返回：

查询返回 RS232 的解码位顺序。

示例：

:BUS1:RS232:ORDer LSB /*设置 BUS1 的 RS232 解码位顺序为 LSB*/

:BUS1:RS232:ORDer ? /*查询返回 LSB*/

:BUS<n>:RS232:WIDTh

功能描述：设置或查询 RS232 的解码位宽。

指令：

:BUS<n>:RS232:WIDTh <width>

:BUS<n>:RS232:WIDTh ?

参数：

<n>= 整型，范围 1~2

<width>= 离散型{5 | 6 | 7 | 8}

返回：

查询返回 RS232 的解码位宽。

示例：

:BUS1:RS232:WIDTh 8 /*设置 BUS1 的 RS232 解码位宽为 8*/

:BUS1:RS232:WIDTh ? /*查询返回 8*/

:BUS<n>:RS232:STOP

功能描述：设置或查询 RS232 的解码停止位。

指令：

:BUS<n>:RS232:STOP <stop>

:BUS<n>:RS232:STOP ?

参数：

<n>= 整型，范围 1~2

<width>= 离散型{1 | 2}

返回：

查询返回 RS232 的解码停止位。

示例：

:BUS1:RS232:STOP 1 /*设置 BUS1 的 RS232 解码停止位为 1*/

:BUS1:RS232:STOP ? /*查询返回 1*/

:BUS<n>:RS232:PARity

功能描述：设置或查询 RS232 的解码奇偶校验位。

指令：

:BUS<n>:RS232:PARity <type>

:BUS<n>:RS232:PARity ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{NONe | ODD | EVEN}

- NONe:无校验
- ODD:奇校验
- EVEN:偶校验

返回：

查询返回 RS232 的解码奇偶校验位。

示例：

:BUS1:RS232:PARity ODD /*设置 BUS1 的 RS232 解码奇偶校验位为 ODD*/

:BUS1:RS232:PARity ? /*查询返回 ODD*/

:BUS<n>:RS232:POLarity

功能描述：设置或查询 RS232 的解码极性。

指令：

:BUS<n>:RS232:POLarity <type>

:BUS<n>:RS232:POLarity ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{NEGative | POSitive}

- NEGative:负极性
- POSitive:正极性

返回：

查询返回 RS232 的解码极性。

示例：

:BUS1:RS232:POLarity POSitive /*设置 BUS1 的 RS232 解码极性为 POSitive*/

:BUS1:RS232:POLarity ? /*查询返回 POSitive*/

:BUS<n>:I2C

:BUS<n>:I2C:SDA

功能描述：设置或查询 I2C 解码的数据源。

指令：

:BUS<n>:I2C:SDA <source>

:BUS<n>:I2C:SDA ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 I2C 解码的数据源。

示例：

:BUS1:I2C:SDA C1 /*设置 BUS1 的 I2C 解码数据源 C1*/

:BUS1:I2C:SDA ? /*查询返回 C1*/

:BUS<n>:I2C:SCL

功能描述：设置或查询 I2C 解码的时钟信源。

指令：

:BUS<n>:I2C:SCL <source>

:BUS<n>:I2C:SCL ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 I2C 解码的数据源。

示例：

:BUS1:I2C:SCL C2 /*设置 BUS1 的 I2C 解码时钟源 C2*/

:BUS1:I2C:SCL ? /*查询返回 C2*/

:BUS<n>:I2C:DLEVel

功能描述：设置或查询 I2C 解码的数据阈值，默认单位 V。

指令：

:BUS<n>:I2C:DLEVel <level>

:BUS<n>:I2C:DLEVel ?

参数：

<n>= 整型，范围 1~2

<level>= 实型，单位与通道垂直刻度单位一致，默认单位为 V，范围-8V~8V

返回：

查询返回 I2C 解码的数据阈值。

示例：

:BUS1:I2C:DLEVel 1 /*设置 BUS1 的 I2C 解码数据阈值为 1V*/

:BUS1:I2C:DLEVel ? /*查询返回 1.00000E+000*/

:BUS<n>:I2C:CLEVel

功能描述：设置或查询 I2C 解码的时钟阈值，默认单位 V。

指令：

:BUS<n>:I2C:CLEVel <level>

:BUS<n>:I2C:CLEVel ?

参数：

<n>= 整型，范围 1~2

<level>= 实型，单位与通道垂直刻度单位一致，默认单位为 V，范围-8V~8V

返回：

查询返回 I2C 解码的时钟阈值。

示例：

:BUS1:I2C:CLEVel 1 /*设置 BUS1 的 I2C 解码时钟阈值为 1V*/

:BUS1:I2C:CLEVel ? /*查询返回 1.00000E+000*/

:BUS<n>:I2C:WIDth

功能描述：设置或查询 I2C 解码的位宽。

指令：

:BUS<n>:I2C:WIDTh <width>

:BUS<n>:I2C:WIDTh ?

参数：

<n>= 整型，范围 1~2

<width>= 离散型{7 | 10}

返回：

查询返回 I2C 解码的位宽。

示例：

:BUS1:I2C:WIDTh 7 /*设置 BUS1 的 I2C 解码的位宽为 7bit*/

:BUS1:I2C:WIDTh ? /*查询返回 7*/

:BUS<n>:SPI

:BUS<n>:SPI:MOD

功能描述：设置或查询 SPI 解码模式。

指令：

:BUS<n>:SPI:MOD <type>

:BUS<n>:SPI:MOD ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{TIMEout | CS }

- TIMEout:超时
- CS:片选

返回：

查询返回 SPI 解码模式。

示例：

:BUS1:SPI:MOD CS /*设置 BUS1 的 SPI 解码模式为 CS*/

:BUS1:SPI:MOD ? /*查询返回 CS*/

:BUS<n>:SPI:TIMEout

功能描述：设置或查询 SPI 解码模式的空闲时间。

指令：

:BUS<n>:SPI:TIMEout <value>

:BUS<n>:SPI:TIMEout ?

参数：

<n>= 整型，范围 1~2

<value>= 实型，范围：50ns-10s。

返回：

查询返回 SPI 解码模式 TIMEout 的空闲时间。

示例：

:BUS1:SPI:TIMEout 1 /*设置解码模式空闲时间为 1s*/

:BUS1:SPI:TIMEout ? /*查询返回 1.0000000000000E+000*/

:BUS<n>:SPI:SOURce:SCL

功能描述：设置或查询 SPI 解码时钟信源。

指令：

:BUS<n>:SPI:SOURce:SCL <source>

:BUS<n>:SPI:SOURce:SCL ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 SPI 解码时钟信源。

示例：

:BUS1:SPI:SOURce:SCL C2 /*设置 BUS1 的 SPI 解码时钟源 C2*/

:BUS1:SPI:SOURce:SCL ? /*查询返回 C2*/

:BUS<n>:SPI:SOURce:BIT

功能描述：设置或查询 SPI 解码的字选择源。

指令：

:BUS<n>:SPI:SOURce:BIT <source>

:BUS<n>:SPI:SOURce:BIT ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 SPI 解码的字选择源。

示例：

:BUS1:SPI:SOURce:BIT C3 /*设置 BUS1 的 SPI 解码字选择源 C3*/

:BUS1:SPI:SOURce:BIT ? /*查询返回 C3*/

:BUS<n>:SPI:SOURce:DAT

功能描述：设置或查询 SPI 解码的数据源。

指令：

:BUS<n>:SPI:SOURce:DAT <source>

:BUS<n>:SPI:SOURce:DAT ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 SPI 解码的数据源。

示例：

:BUS1:SPI:SOURce:DAT C3 /*设置 BUS1 的 SPI 解码数据源 C3*/

:BUS1:SPI:SOURce:DAT ? /*查询返回 C3*/

:BUS<n>:SPI:LEVel:SCL

功能描述：设置或查询 SPI 解码的时钟阈值，默认单位 V。

指令：

:BUS<n>:SPI:LEVel:SCL <level>

:BUS<n>:SPI:LEVel:SCL ?

参数：

<n>= 整型，范围 1~2

<level>= 实型，单位与通道垂直刻度单位一致，默认单位为 V，范围-8V~8V

返回：

查询返回 SPI 解码的时钟阈值。

示例：

:BUS1:SPI:LEVel:SCL 1 /*设置 BUS1 的 SPI 解码时钟阈值为 1V*/

:BUS1:SPI:LEVel:SCL ? /*查询返回 1.00000E+000*/

:BUS<n>:SPI:LEVel:BIT

功能描述：设置或查询 SPI 解码的字选择阈值，默认单位 V。

指令：

:BUS<n>:SPI:LEVel:BIT <level>

:BUS<n>:SPI:LEVel:BIT ?

参数：

<n>= 整型，范围 1~2

<level>= 实型，单位与通道垂直刻度单位一致，默认单位为 V，范围-8V~8V

返回：

查询返回 SPI 解码的字选择阈值。

示例：

:BUS1:SPI:LEVel:BIT 1 /*设置 BUS1 的 SPI 解码字选择阈值为 1V*/

:BUS1:SPI:LEVel:BIT ? /*查询返回 1.00000E+000*/

:BUS<n>:SPI:LEVel:DATa

功能描述：设置或查询 SPI 解码的数据阈值，默认单位 V。

指令：

:BUS<n>:SPI:LEVel:DATa <level>

:BUS<n>:SPI:LEVel:DATa ?

参数：

<n>= 整型，范围 1~2

<level>= 实型，单位与通道垂直刻度单位一致，默认单位为 V，范围-8V~8V

返回：

查询返回 SPI 解码的数据阈值。

示例：

:BUS1:SPI:LEVel:DATa 1 /*设置 BUS1 的 SPI 解码 MOSI 阈值为 1V*/

:BUS1:SPI:LEVel:DATa ? /*查询返回 1.00000E+000*/

:BUS<n>:SPI:EDGe:SCL

功能描述：设置或查询 SPI 解码的时钟沿。

指令：

:BUS<n>:SPI:EDGe:SCL <type>

:BUS<n>:SPI:EDGe:SCL ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{FALL | RISe}

- FALL:下降沿
- RISe:上升沿

返回：

查询返回 SPI 解码的时钟沿。

示例：

:BUS1:SPI:EDGe:SCL RISe /*设置 BUS1 的 SPI 解码时钟沿为 RISe*/

:BUS1:SPI:EDGe:SCL ? /*查询返回 RISe*/

:BUS<n>:SPI:EDGe:BIT

功能描述：设置或查询 SPI 解码的字选择极性。

指令：

:BUS<n>:SPI:EDGe:BIT <type>

:BUS<n>:SPI:EDGe:BIT ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{LOW | HIGH}

- LOW:负极性
- HIGH:正极性

返回：

查询返回 SPI 解码的字选择极性。

示例：

:BUS1:SPI:EDGe:BIT LOW /*设置 BUS1 的 SPI 字选择极性为负极性*/

:BUS1:SPI:EDGe:BIT ? /*查询返回 LOW*/

:BUS<n>:SPI:POLarity:DATa

功能描述：设置或查询 SPI 解码的数据极性。

指令：

:BUS<n>:SPI:POLarity:DATa <type>

:BUS<n>:SPI:POLarity:DATa ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{NEGative | POSitive}

- NEGative:负极性
- POSitive:正极性

返回：

查询返回 SPI 解码的数据极性。

示例：

:BUS1:SPI:POLarity:DATa POSitive /*设置 BUS1 的 SPI 解码数据极性为 POSitive*/

:BUS1:SPI:POLarity:DATa ? /*查询返回 POSitive*/

:BUS<n>:SPI:FRAMeCount

功能描述：设置或查询 SPI 的数据位宽。

指令：

:BUS<n>:SPI:FRAMeCount <value>

:BUS<n>:SPI:FRAMeCount ?

参数：

<n>= 整型，范围 1~2

<value>= 实型 4-32

返回：

查询返回 SPI 的数据位宽。

示例：

:BUS1:SPI:FRAMeCount 8 /*设置 BUS1 的 RS232 解码位宽为 8*/

:BUS1:SPI:FRAMeCount ? /*查询返回 8*/

:BUS<n>:SPI:ORDer

功能描述：设置或查询 SPI 解码的位顺序。

指令：

:BUS<n>:SPI:ORDer <type>

:BUS<n>:SPI:ORDer ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{MSB | LSB}

- MSB:最高有效位
- LSB:最低有效位

返回：

查询返回 SPI 解码的位顺序。

示例：

:BUS1:SPI:ORDer LSB /*设置 BUS1 的 SPI 解码位顺序为 LSB*/

:BUS1:SPI:ORDer ? /*查询返回 LSB*/

:BUS<n>:CAN

:BUS<n>:CAN:SOURce

功能描述：设置或查询 CAN 解码信源。

指令：

:BUS<n>:CAN:SOURce <source>

:BUS<n>:CAN:SOURce ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 CAN 解码信源。

示例：

:BUS1:CAN:SOURce C1 /*设置 BUS1 的 CAN 解码源为 C1*/

:BUS1:CAN:SOURce ? /*查询返回 C1*/

:BUS<n>:CAN:STYPe

功能描述：设置或查询 CAN 解码信号类型。

指令：

:BUS<n>:CAN:STYPe <type>

:BUS<n>:CAN:STYPe ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{L | H}

- L:CAN_L
- H:CAN_H

返回：

查询返回 CAN 解码信号类型。

示例：

:BUS1:CAN:STYPe H /*设置 BUS1 的 CAN 解码信号类型为 CAN_H*/

:BUS1:CAN:STYPe ? /*查询返回 H*/

:BUS<n>:CAN:LEVel

功能描述：设置或查询 CAN 解码阈值，默认单位 V。

指令：

:BUS<n>:CAN:LEVel <level>

:BUS<n>:CAN:LEVel ?

参数：

<n>= 整型，范围 1~2

<level>= 实型，单位与通道垂直刻度单位一致，默认单位为 V，范围-12V~12V

返回：

查询返回 CAN 解码阈值。

示例：

:BUS1:CAN:LEVel 1 /*设置 BUS1 的 CAN 解码阈值为 1V*/

:BUS1:CAN:LEVel ? /*查询返回 1.00000E+000*/

:BUS<n>:CAN:BAUDrate:CURRent

功能描述：设置或查询 CAN 解码信号速率。

指令：

:BUS<n>:CAN:BAUDrate:CURRent <value>

:BUS<n>:CAN:BAUDrate:CURRent ?

参数：

<n>= 整型，范围 1~2

<value>= 离散型{CUSTom | 10000 | 19200 | 20000 | 33300 | 38400 | 50000 | 57600 | 62500 | 83300 | 100000 | 115200 | 125000 | 230400 | 250000 | 490800 | 500000 | 800000 | 921600 | 1000000 | 2000000 | 3000000 | 4000000 | 5000000}

返回：

查询返回 CAN 解码信号速率。

示例：

:BUS1:CAN:BAUDrate:CURRent 10000 /*设置 BUS1 的 CAN 解码信号速率为 10k*/

:BUS1:CAN:BAUDrate:CURRent ? /*查询返回 10000*/

:BUS<n>:CAN:SPOint

功能描述：设置或查询 CAN 解码采样点，默认单位%。

指令：

:BUS<n>:CAN:SPOint <value>

:BUS<n>:CAN:SPOint ?

参数：

<n>= 整型，范围 1~2

<value>= 整数型，范围 30%~90%

返回：

查询返回 CAN 采样点。

示例：

:BUS1:CAN:SPOint 30 /*设置 BUS1 的 CAN 解码采样点为 30%*/

:BUS1:CAN:SPOint ? /*查询返回 30*/

:BUS<n>:CAN_FD

:BUS<n>:CAN_FD:SOURce

功能描述：设置或查询 CAN_FD 解码信源。

指令：

:BUS<n>:CAN_FD:SOURce <source>

:BUS<n>:CAN_FD:SOURce ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 CAN_FD 解码信源。

示例：

:BUS1:CAN_FD:SOURce C1 /*设置 BUS1 的 CAN_FD 解码源为 C1*/

:BUS1:CAN_FD:SOURce ? /*查询返回 C1*/

:BUS<n>:CAN_FD:STYPe

功能描述：设置或查询 CAN_FD 解码信号类型。

指令：

:BUS<n>:CAN_FD:STYPe <type>

:BUS<n>:CAN_FD:STYPe ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{L | H}

返回：

查询返回 CAN_FD 解码信号类型。

示例：

:BUS1:CAN_FD:STYPe H /*设置 BUS1 的 CAN_FD 解码信号类型为 CAN_H*/

:BUS1:CAN_FD:STYPe ? /*查询返回 H*/

:BUS<n>:CAN_FD:LEVel

功能描述：设置或查询 CAN_FD 解码阈值，默认单位 V。

指令：

:BUS<n>:CAN_FD:LEVel <level>

:BUS<n>:CAN_FD:LEVel ?

参数：

<n>= 整型，范围 1~2

<level>= 实型，单位与通道垂直刻度单位一致，默认单位为 V，范围-12V~12V

返回：

查询返回 CAN_FD 解码阈值。

示例：

:BUS1:CAN_FD:LEVel 1 /*设置 BUS1 的 CAN_FD 解码阈值为 1V*/

:BUS1:CAN_FD:LEVel ? /*查询返回 1.00000E+000*/

:BUS<n>:CAN_FD:BAUDrate:SDCURrent

功能描述：设置或查询 CAN_FD 解码 SD 信号速率。

指令：

:BUS<n>:CAN_FD:BAUDrate:SDCURrent <value>

:BUS<n>:CAN_FD:BAUDrate:SDCURrent ?

参数：

<n>= 整型，范围 1~2

<value>= 离散型{CUSTom | 10000 | 19200 | 20000 | 33300 | 38400 | 50000 | 57600 | 62500 | 83300 | 100000 | 115200 | 125000 | 230400 | 250000 | 490800 | 500000 | 800000 | 921600 | 1000000 | 2000000 | 3000000 | 4000000 | 5000000}

返回：

查询返回 CAN_FD 解码 SD 信号速率。

示例：

:BUS1:CAN_FD:BAUDrate:SDCURrent 10000 /*设置 BUS1 的 CAN_FD 解码 SD 信号速率为 10k*/

:BUS1:CAN_FD:BAUDrate:SDCURrent ? /*查询返回 10000*/

:BUS<n>:CAN_FD:BAUDrate:SDCUStom

功能描述：设置或查询 CAN_FD 解码 SD 自定义信号速率，默认单位 bps。

指令：

:BUS<n>:CAN_FD:BAUDrate:SDCUStom <value>

:BUS<n>:CAN_FD:BAUDrate:SDCUStom ?

参数：

<n>= 整型，范围 1~2

<value>= 整数型，范围 10kbps~1Mbps

返回：

查询返回 CAN_FD 解码 SD 自定义信号速率，以科学计数法表示。

示例：

:BUS1:CAN_FD:BAUDrate:SDCUStom 10000 /*设置 CAN_FD 解码 SD 自定义信号速率为 10k*/

:BUS1:CAN_FD:BAUDrate:SDCUStom ? /*查询返回 1.00000E+004*/

:BUS<n>:CAN_FD:BAUDrate:FDCURrent

功能描述：设置或查询 CAN_FD 解码 FD 信号速率。

指令：

:BUS<n>:CAN_FD:BAUDrate:FDCURrent <value>

:BUS<n>:CAN_FD:BAUDrate:FDCURrent ?

参数：

<n>= 整型，范围 1~2

<value>= 离散型{CUSTom | 250000 | 500000 | 800000 | 1000000 | 1500000 | 2000000 | 3000000 | 4000000 | 5000000 | 6000000 | 7000000 | 8000000}

返回：

查询返回 CAN_FD 解码 FD 信号速率。

示例：

:BUS1:CAN_FD:BAUDrate:FDCURrent 1000000 /*设置 CAN_FD 解码 FD 信号速率为 1M*/

:BUS1:CAN_FD:BAUDrate:FDCURrent ? /*查询返回 1000000*/

:BUS<n>:CAN_FD:BAUDrate:FDCUStom

功能描述：设置或查询 CAN_FD 解码 FD 自定义信号速率，默认单位 bps。

指令：

:BUS<n>:CAN_FD:BAUDrate:FDCUStom <value>

:BUS<n>:CAN_FD:BAUDrate:FDCUStom ?

参数：

<n>= 整型，范围 1~2

<value>= 整数型，范围 1kbps~10Mbps

返回：

查询返回 CAN_FD 解码 FD 自定义信号速率，以科学计数法表示。

示例：

:BUS1:CAN_FD:BAUDrate:FDCUStom 1000000 /*设置 CAN_FD 解码 FD 自定义信号速率为 1M*/

:BUS1:CAN_FD:BAUDrate:FDCUStom ? /*查询返回 1.00000E+006*/

:BUS<n>:CAN_FD:SPOint

功能描述：设置或查询 CAN_FD 解码采样点，默认单位%。

指令：

:BUS<n>:CAN_FD:SPOint <value>

:BUS<n>:CAN_FD:SPOint ?

参数：

<n>= 整型，范围 1~2

<value>= 整数型，范围 30%~90%

返回：

查询返回 CAN_FD 采样点。

示例：

:BUS1:CAN_FD:SPOint 30 /*设置 BUS1 的 CAN 解码采样点为 30%*/

:BUS1:CAN_FD:SPOint ? /*查询返回 30*/

:BUS<n>:CAN_FD:DPOint

功能描述：设置或查询 CAN_FD 数据域采样点，默认单位%。

指令：

:BUS<n>:CAN_FD:DPOint <value>

:BUS<n>:CAN_FD:DPOint ?

参数：

<n>= 整型，范围 1~2

<value>= 整数型，范围 30%~90%

返回：

查询返回 CAN_FD 数据域采样点。

示例：

:BUS1:CAN_FD:DPOint 30 /*设置 BUS1 的 CAN 解码采样点为 30%*/

:BUS1:CAN_FD:DPOint ? /*查询返回 30*/

:BUS<n>:LIN

:BUS<n>:LIN:SOURce

功能描述：设置或查询 LIN 解码信源。

指令：

:BUS<n>:LIN:SOURce <source>

:BUS<n>:LIN:SOURce ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 LIN 解码信源。

示例：

:BUS1:LIN:SOURce C1 /*设置 BUS1 的 LIN 解码源为 C1*/

:BUS1:LIN:SOURce ? /*查询返回 C1*/

:BUS<n>:LIN:LEVel

功能描述：设置或查询 LIN 解码阈值，默认单位 V。

指令：

:BUS<n>:LIN:LEVel <level>

:BUS<n>:LIN:LEVel ?

参数：

<n>= 整型，范围 1~2

<level>= 实型，单位与通道垂直刻度单位一致，默认单位为 V，范围-12V~12V

返回：

查询返回 LIN 解码阈值。

示例：

:BUS1:LIN:LEVel 1 /*设置 BUS1 的 LIN 解码阈值为 1V*/

:BUS1:LIN:LEVel ? /*查询返回 1.00000E+000*/

:BUS<n>:LIN:POLarity

功能描述：设置或查询 LIN 解码极性。

指令：

:BUS<n>:LIN:POLarity <type>

:BUS<n>:LIN:POLarity ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{POSitive | NEGative}

- NEGative:负极性
- POSitive:正极性

返回：

查询返回 LIN 解码极性。

示例：

:BUS1:LIN:POLarity POSitive /*设置 BUS1 的 LIN 解码极性为 POSitive*/

:BUS1:LIN:POLarity ? /*查询返回 POSitive*/

:BUS<n>:LIN:BAUDrate:CURRent

功能描述：设置或查询 LIN 解码波特率。

指令：

:BUS<n>:LIN:BAUDrate:CURRent <value>

:BUS<n>:LIN:BAUDrate:CURRent ?

参数：

<n>= 整型，范围 1~2

<value>= 离散型{CUSTom | 2400 | 4800 | 9600 | 19200}

返回：

查询返回 LIN 解码波特率。

示例：

:BUS1:LIN:BAUDrate:CURRent 9600 /*设置 BUS1 的 LIN 解码波特率为 9600*/

:BUS1:LIN:BAUDrate:CURRent ? /*查询返回 9600*/

:BUS<n>:LIN:BAUDrate:CUSTom

功能描述：设置或查询 LIN 解码自定义波特率，默认单位 bps。

指令：

:BUS<n>:LIN:BAUDrate:CUSTom <value>

:BUS<n>:LIN:BAUDrate:CUSTom ?

参数：

<n>= 整型，范围 1~2

<value>= 整数型，范围 1kbps~200Mbps

返回：

查询返回 LIN 解码自定义波特率。

示例：

:BUS1:LIN:BAUDrate:CUSTom 10000 /*设置 BUS1 的 LIN 解码自定义波特率为 10kbps*/

:BUS1:LIN:BAUDrate:CUSTom ? /*查询返回 1.00000E+004*/

:BUS<n>:LIN:STANdard

功能描述：设置或查询 LIN 解码信号标准。

指令：

:BUS<n>:LIN:STANdard <type>

:BUS<n>:LIN:STANdard ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{V1 | V2 | Both}

- V1:LIN1.0
- V2:LIN2.0

- Both:Both

返回：

查询返回 LIN 解码信号标准。

示例：

:BUS1:LIN:STANdard V1 /*设置 BUS1 的 LIN 解码信号标准为 LIN1.0*/

:BUS1:LIN:STANdard ? /*查询返回 V1 /

:BUS<n>:LIN:SPOint

功能描述：设置或查询 LIN 解码采样点。

指令：

:BUS<n>:LIN:SPOint <value>

:BUS<n>:LIN:SPOint ?

参数：

<n>= 整型，范围 1~2

<value>= 实型，范围 50%-90%

返回：

查询返回 LIN 解码采样点。

示例：

:BUS1:LIN:SPOint 50 /*设置 BUS1 的 LIN 解码采样点为 50%*/

:BUS1:LIN:SPOint ? /*查询返回 50*/

:BUS<n>:LIN:CKPID

功能描述：设置或查询 LIN 解码 ID 包括奇偶位。

指令：

:BUS<n>:LIN:CKPID <type>

:BUS<n>:LIN:CKPID?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{Y | N}

- Y:是
- N:否

返回：

查询返回 LIN 解码是否包括奇偶位。

示例：

:BUS1:LIN:CKPID Y /*设置 BUS1 的 LIN 解码 ID 包括奇偶位为是*/

:BUS1:LIN:CKPID ? /*查询返回 Y*/

:BUS<n>:FLEXRAY

:BUS<n>:FR:SOURce

功能描述：设置或查询 FLEXRAY 解码信源。

指令：

:BUS<n>:FR:SOURce <source>

:BUS<n>:FR:SOURce ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 FLEXRAY 解码信源。

示例：

:BUS1:FR:SOURce C1 /*设置 BUS1 的 FLEXRAY 解码源为 C1*/

:BUS1:FR:SOURce ? /*查询返回 C1*/

:BUS<n>:FR:STYPe

功能描述：设置或查询 FLEXRAY 解码源类型。

指令：

:BUS<n>:FR:STYPe <type>

:BUS<n>:FR:STYPe ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{BP | BM}

返回：

查询返回 FLEXRAY 解码源类型。

示例：

:BUS1:FR:STYPe BP /*设置 BUS1 的 FLEXRAY 解码源类型为 BP*/

:BUS1:FR:STYPe ? /*查询返回 BP*/

:BUS<n>:FR:BAUDrate:CURRent

功能描述：设置或查询 FLEXRAY 解码波特率，默认单位 bps。

指令：

:BUS<n>:FR:BAUDrate:CURRent <value>

:BUS<n>:FR:BAUDrate:CURRent ?

参数：

<n>= 整型，范围 1~2

<value>= 离散型{CUSTom | 1000000 | 5000000 | 10000000}

返回：

查询返回 FLEXRAY 解码波特率。

示例：

:BUS1:FR:BAUDrate:CURRent 1000000 /*设置 BUS1 的 FLEXRAY 解码波特率为 1Mbps*/

:BUS1:FR:BAUDrate:CURRent ? /*查询返回 1000000*/

:BUS<n>:FR:BAUDrate:CUSTom

功能描述：设置或查询 FLEXRAY 解码自定义波特率，默认单位 bps。

指令：

:BUS<n>:FR:BAUDrate:CUSTom <value>

:BUS<n>:FR:BAUDrate:CUSTom ?

参数：

<n>= 整型，范围 1~2

<value>= 整数型，范围 1Mbps~10Mbps

返回：

查询返回 FLEXRAY 解码自定义波特率。

示例：

:BUS1:FR:BAUDrate:CUSTom 1000000 /*设置 BUS1 的 FLEXRAY 解码自定义波特率为 1Mbps*/

:BUS1:FR:BAUDrate:CUSTom ? /*查询返回 1.00000E+006*/

:BUS<n>:FR:LEVel

功能描述：设置或查询 FLEXRAY 解码阈值，默认单位 V。

指令：

:BUS<n>:FR:LEVel <level>

:BUS<n>:FR:LEVel ?

参数：

<n>= 整型，范围 1~2

<level>= 实型，单位与通道垂直刻度单位一致，默认单位为 V，范围-12V~12V

返回：

查询返回 FLEXRAY 解码阈值。

示例：

:BUS1:FR:LEVel 1 /*设置 BUS1 的 FLEXRAY 解码阈值为 1V*/

:BUS1:FR:LEVel ? /*查询返回 1.00000E+000*/

:BUS<n>:FR:CHANnel

功能描述：设置或查询 FLEXRAY 解码通道类型。

指令：

:BUS<n>:FR:CHANnel <type>

:BUS<n>:FR:CHANnel ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{A | B}

返回：

查询返回 FLEXRAY 解码通道类型。

示例：

:BUS1:FR:CHANnel A /*设置 BUS1 的 FLEXRAY 解码通道类型为 A*/

:BUS1:FR:CHANnel ? /*查询返回 A*/

:BUS<n>:AUDIOBUS

:BUS<n>:AB:TYPe

功能描述：设置或查询 AUDIOBUS 解码协议类型。

指令：

:BUS<n>:AB:TYPe <type>

:BUS<n>:AB:TYPe ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{I2S | LJ | RJ | TDM}

返回：

查询返回 AUDIOBUS 解码协议类型。

示例：

:BUS1:AB:TYPe I2S /*设置 BUS1 的 AUDIOBUS 解码协议类型为 I2S*/

:BUS1:AB:TYPe ? /*查询返回 I2S*/

:BUS<n>:AB:DPOLarity

功能描述：设置或查询 AUDIOBUS 解码数据极性。

指令：

:BUS<n>:AB:DPOLarity <type>

:BUS<n>:AB:DPOLarity ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{POSitive | NEGative}

- POSitive:正极性
- NEGative:负极性

返回：

查询返回 AUDIOBUS 解码数据极性。

示例：

:BUS1:AB:DPOLarity NEGative /*设置 BUS1 的 AUDIOBUS 解码数据极性为 NEGative*/

:BUS1:AB:DPOLarity ? /*查询返回 NEGative*/

:BUS<n>:AB:SPOLarity

功能描述：设置或查询 AUDIOBUS 解码同步极性。

指令：

:BUS<n>:AB:SPOLarity <type>

:BUS<n>:AB:SPOLarity ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{POSitive | NEGative}

- POSitive:正极性
- NEGative:负极性

返回：

查询返回 AUDIOBUS 解码同步极性。

示例：

:BUS1:AB:SPOLarity NEGative /*设置 BUS1 的 AUDIOBUS 解码同步极性为 NEGative*/

:BUS1:AB:SPOLarity ? /*查询返回 NEGative*/

:BUS<n>:AB:CEDGe

功能描述：设置或查询 AUDIOBUS 解码有效时钟沿。

指令：

:BUS<n>:AB:CEDGe <type>

:BUS<n>:AB:CEDGe?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{RISe | FALL}

- RISe:上升沿
- FALL:下降沿

返回：

查询返回 AUDIOBUS 解码有效时钟沿。

示例：

:BUS1:AB:CEDGe FALL /*设置 BUS1 的 AUDIOBUS 解码有效时钟沿为 FALL*/

:BUS1:AB:CEDGe ? /*查询返回 FALL*/、

:BUS<n>:AB:ORDer

功能描述：设置或查询 AUDIOBUS 解码位序。

指令：

:BUS<n>:AB:ORDer <type>

:BUS<n>:AB:ORDer ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{MSB | LSB}

- MSB:最高有效位
- LSB:最低有效位

返回：

查询返回 AUDIOBUS 解码位序。

示例：

:BUS1:AB:ORDer LSB /*设置 BUS1 的 AUDIOBUS 解码位序为 LSB*/

:BUS1:AB:ORDer ? /*查询返回 LSB*/

:BUS<n>:AB:SCTYpe

功能描述：设置或查询 AUDIOBUS 解码声道类型。

指令：

:BUS<n>:AB:SCTYpe <type>

:BUS<n>:AB:SCTYpe ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{LR | L | R}

- LR:左/右声道
- L:左声道
- R:右声道

返回：

查询返回 AUDIOBUS 解码声道类型。

示例：

:BUS1:AB:SCTYpe R /*设置 BUS1 的 AUDIOBUS 解码声道类型为右声道*/

:BUS1:AB:SCTYpe ? /*查询返回 R*/

:BUS<n>:AB:SCCount

功能描述：设置或查询 AUDIOBUS 解码每帧通道数。

指令：

:BUS<n>:AB:SCCount <value>

:BUS<n>:AB:SCCount ?

参数：

<n>= 整型，范围 1~2

<value>= 整数型，范围 4~32

返回：

查询返回 AUDIOBUS 解码每帧通道数。

示例：

:BUS1:AB:SCCount 4 /*设置 BUS1 的 AUDIOBUS 解码每帧通道数为 4*/

:BUS1:AB:SCCount ? /*查询返回 4*/

:BUS<n>:AB:DBCount

功能描述：设置或查询每通道数据位数。

指令：

:BUS<n>:AB:DBCount <value>

:BUS<n>:AB:DBCount ?

参数：

<n>= 整型，范围 1~2

<value>= 整数型

返回：

查询返回每通道数据位数。

示例：

:BUS1:AB:DBCount 2 /*设置 1 通道数据位数为 2*/

:BUS1:AB:DBCount ? /*查询返回 2*/

:BUS<n>:AB:CBCount

功能描述：设置或查询 AUDIOBUS 解码每声道时钟位数。

指令：

:BUS<n>:AB:CBCount <value>

:BUS<n>:AB:CBCount ?

参数：

<n>= 整型，范围 1~2

<value>= 整数型，范围 4~32

返回：

查询返回 AUDIOBUS 解码每声道时钟位数。

示例：

:BUS1:AB:CBCount 8 /*设置 BUS1 的 AUDIOBUS 解码每声道时钟位数为 8*/

:BUS1:AB:CBCount ? /*查询返回 8*/

:BUS<n>:AB:SOURce:SCL

功能描述：设置或查询 AUDIOBUS 解码时钟源。

指令：

:BUS<n>:AB:SOURce:SCL <source>

:BUS<n>:AB:SOURce:SCL ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 AUDIOBUS 解码时钟源。

示例：

:BUS1:AB:SOURce:SCL C4 /*设置 BUS1 的 AUDIOBUS 解码时钟源为 C4*/

:BUS1:AB:SOURce:SCL ? /*查询返回 C4*/

:BUS<n>:AB:SOURce:CS

功能描述：设置或查询 AUDIOBUS 解码字选择源。

指令：

:BUS<n>:AB:SOURce:CS <source>

:BUS<n>:AB:SOURce:CS ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 AUDIOBUS 解码字选择源。

示例：

:BUS1:AB:SOURce:CS C3 /*设置 BUS1 的 AUDIOBUS 解码字选泽源为 C3*/

:BUS1:AB:SOURce:CS ? /*查询返回 C3*/

:BUS<n>:AB:SOURce:SDA

功能描述：设置或查询 AUDIOBUS 解码数据源。

指令：

:BUS<n>:AB:SOURce:SDA <source>

:BUS<n>:AB:SOURce:SDA ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 AUDIOBUS 解码数据源。

示例：

:BUS1:AB:SOURce:SDA C2 /*设置 BUS1 的 AUDIOBUS 解码数据源为 C2*/

:BUS1:AB:SOURce:SDA ? /*查询返回 C2*/

:BUS<n>:AB:LEVel:SCL

功能描述：设置或查询 AUDIOBUS 解码时钟阈值，默认单位 V。

指令：

:BUS<n>:AB:LEVel:SCL <level>

:BUS<n>:AB:LEVel:SCL ?

参数：

<n>= 整型，范围 1~2

<level>= 实型，单位与通道垂直刻度单位一致，默认单位为 V，范围-20V~20V

返回：

查询返回 AUDIOBUS 解码时钟阈值。

示例：

:BUS1:AB:LEVel:SCL 1 /*设置 BUS1 的 AUDIOBUS 解码时钟阈值为 1V*/

:BUS1:AB:LEVel:SCL ? /*查询返回 1.00000E+000*/

:BUS<n>:AB:LEVel:CS

功能描述：设置或查询 AUDIOBUS 解码字选择阈值，默认单位 V。

指令：

:BUS<n>:AB:LEVel:CS <level>

:BUS<n>:AB:LEVel:CS ?

参数：

<n>= 整型，范围 1~2

<level>= 实型，单位与通道垂直刻度单位一致，默认单位为 V，范围-20V~20V

返回：

查询返回 AUDIOBUS 解码字选择阈值。

示例：

:BUS1:AB:LEVel:CS 1 /*设置 BUS1 的 AUDIOBUS 解码字选择阈值为 1V*/

:BUS1:AB:LEVel:CS ? /*查询返回 1.00000E+000*/

:BUS<n>:AB:LEVel:SDA

功能描述：设置或查询 AUDIOBUS 解码数据阈值，默认单位 V。

指令：

:BUS<n>:AB:LEVel:SDA <level>

:BUS<n>:AB:LEVel:SDA ?

参数：

<n>= 整型，范围 1~2

<level>= 实型，单位与通道垂直刻度单位一致，默认单位为 V，范围-20V~20V

返回：

查询返回 AUDIOBUS 解码数据阈值。

示例：

:BUS1:AB:LEVel:SDA 1 /*设置 BUS1 的 AUDIOBUS 解码数据阈值为 1V*/

:BUS1:AB:LEVel:SDA ? /*查询返回 1.00000E+000*/

:BUS<n>:AB:BITDelay

功能描述：设置或查询 AUDIOBUS 解码位延迟。

指令：

:BUS<n>:AB:BITDelay <value>

:BUS<n>:AB:BITDelay ?

参数：

<n>= 整型，范围 1~2

<value>= 整数型，范围 0~31

返回：

查询返回 AUDIOBUS 解码位延迟。

示例：

:BUS1:AB:BITDelay 10 /*设置 BUS1 的 AUDIOBUS 解码位延迟为 10*/

:BUS1:AB:BITDelay ? /*查询返回 10*/

:BUS<n>:MIL_STD_1553

:BUS<n>:MS:SOURce

功能描述：设置或查询 MIL_STD_1553 解码信源。

指令：

:BUS<n>:MS:SOURce <source>

:BUS<n>:MS:SOURce ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 MIL_STD_1553 解码信源。

示例：

:BUS1:MS:SOURce C4 /*设置 BUS1 的 MIL_STD_1553 解码信源为 C4*/

:BUS1:MS:SOURce ? /*查询返回 C4*/

:BUS<n>:MS:BAUDrate:CURRent

功能描述：设置或查询 MIL_STD_1553 解码波特率，默认单位为 bps。

指令：

:BUS<n>:MS:BAUDrate:CURRent <value>

:BUS<n>:MS:BAUDrate:CURRent ?

参数：

<n>= 整型，范围 1~2

<value>= 离散型{CUSTom | 1000000 | 10000000}

返回：

查询返回 MIL_STD_1553 解码波特率。

示例：

:BUS1:MS:BAUDrate:CURRent 1000000 /*设置 BUS1 的 MIL_STD_1553 解码波特率为 1Mbps*/

:BUS1:MS:BAUDrate:CURRent ? /*查询返回 1000000*/

:BUS<n>:MS:BAUDrate:CUSTom

功能描述：设置或查询 MIL_STD_1553 解码波特率，默认单位为 bps。

指令：

:BUS<n>:MS:BAUDrate:CUSTom <value>

:BUS<n>:MS:BAUDrate:CUSTom ?

参数：

<n>= 整型，范围 1~2

<value>= 整数型，范围 1kbps~20Mbps

返回：

查询返回 MIL_STD_1553 解码波特率。

示例：

:BUS1:MS:BAUDrate:CUSTom 1000000 /*设置 BUS1 的 MIL_STD_1553 解码波特率为 1Mbps*/

:BUS1:MS:BAUDrate:CUSTom ? /*查询返回 1.00000E+006*/

:BUS<n>:MS:LEVel:HIGH

功能描述：设置或查询 MIL_STD_1553 解码高阈值，默认单位 V。

指令：

:BUS<n>:MS:LEVel:HIGH <value>

:BUS<n>:MS:LEVel:HIGH ?

参数：

<n>= 整型，范围 1~2

<value>= 实型，范围-30V~30V

返回：

查询返回 MIL_STD_1553 解码高阈值。

示例：

:BUS1:MS:LEVel:HIGH 2 /*设置 BUS1 的 MIL_STD_1553 解码高阈值为 2V*/

:BUS1:MS:LEVel:HIGH ? /*查询返回 2.00000E+000*/

:BUS<n>:MS:LEVel:LOW

功能描述：设置或查询 MIL_STD_1553 解码低阈值,默认单位 V。

指令：

:BUS<n>:MS:LEVel:LOW <value>

:BUS<n>:MS:LEVel:LOW ?

参数：

<n>= 整型，范围 1~2

<value>= 实型，范围-30V~30V

返回：

查询返回 MIL_STD_1553 解码低阈值。

示例：

:BUS1:MS:LEVel:LOW 1 /*设置 BUS1 的 MIL_STD_1553 解码低阈值为 1V*/

:BUS1:MS:LEVel:LOW ? /*查询返回 1.00000E+000*/

:BUS<n>:MS:POLarity

功能描述：设置或查询 MIL_STD_1553 解码极性。

指令：

:BUS<n>:MS:POLarity <polarity>

:BUS<n>:MS:POLarity ?

参数：

<n>= 整型，范围 1~2

<polarit>= 离散型{POSitive | NEGative}

- POSitive:正极性
- NEGative:负极性

返回：

查询返回 MIL_STD_1553 解码极性。

示例：

:BUS1:MS:POLarity POSitive /*设置 BUS1 的 MIL_STD_1553 解码极性为正极性*/

:BUS1:MS:POLarity ? /*查询返回 POSitive*/

:BUS<n>:A429

:BUS<n>:A429:SOURce

功能描述：设置或查询 ARINC429 解码信源。

指令：

:BUS<n>:A429:SOURce <source>

:BUS<n>:A429:SOURce ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 ARINC429 解码信源。

示例：

:BUS1:A429:SOURce C4 /*设置 BUS1 的 ARINC429 解码信源为 C4*/

:BUS1:A429:SOURce ? /*查询返回 C4*/

:BUS<n>:A429:BAUDrate:CURRent

功能描述：设置或查询 ARINC429 解码波特率，默认单位为 bps。

指令：

:BUS<n>:A429:BAUDrate:CURRent <value>

:BUS<n>:A429:BAUDrate:CURRent ?

参数：

<n>= 整型，范围 1~2

<value>= 离散型{CUSTom | 12500 | 100000}

返回：

查询返回 ARINC429 解码波特率。

示例：

:BUS1:A429:BAUDrate:CURRent 12500 /*设置 BUS1 的 ARINC429 解码波特率为 12.5kbps*/

:BUS1:A429:BAUDrate:CURRent ? /*查询返回 12500*/

:BUS<n>:A429:BAUDrate:CUSTom

功能描述：设置或查询 ARINC429 自定义解码波特率，默认单位为 bps。

指令：

:BUS<n>:A429:BAUDrate:CUSTom <value>

:BUS<n>:A429:BAUDrate:CUSTom ?

参数：

<n>= 整型，范围 1~2

<value>= 整数型，范围 1kbps~10Mbps

返回：

查询返回 ARINC429 自定义解码波特率。

示例：

:BUS1:A429:BAUDrate:CUSTom 1000000 /*设置 BUS1 的 ARINC429 自定义解码波特率为 1Mbps*/

:BUS1:A429:BAUDrate:CUSTom ? /*查询返回 1.00000E+006*/

:BUS<n>:A429:LEVel:HIGH

功能描述：设置或查询 ARINC429 解码高阈值,默认单位 V。

指令：

:BUS<n>:A429:LEVel:HIGH <value>

:BUS<n>:A429:LEVel:HIGH ?

参数：

<n>= 整型，范围 1~2

<value>= 实型，范围-20V~20V

返回：

查询返回 ARINC429 解码高阈值。

示例：

:BUS1:A429:LEVel:HIGH 2 /*设置 BUS1 的 ARINC429 解码高阈值为 2V*/

:BUS1:A429:LEVel:HIGH ? /*查询返回 2.00000E+000*/

:BUS<n>:A429:LEVel:LOW

功能描述：设置或查询 ARINC429 解码低阈值，默认单位 V。

指令：

:BUS<n>:A429:LEVel:LOW <value>

:BUS<n>:A429:LEVel:LOW ?

参数：

<n>= 整型，范围 1~2

<value>= 实型，范围-20V~20V

返回：

查询返回 ARINC429 解码低阈值。

示例：

:BUS1:A429:LEVel:LOW 1 /*设置 BUS1 的 ARINC429 解码低阈值为 1V*/

:BUS1:A429:LEVel:LOW ? /*查询返回 1.00000E+000*/

:BUS<n>:A429:POLarity

功能描述：设置或查询 ARINC429 解码极性。

指令：

:BUS<n>:A429:POLarity <polarity>

:BUS<n>:A429:POLarity ?

参数：

<n>= 整型，范围 1~2

<polarity>= 离散型{POSitive | NEGative}

- POSitive:正极性
- NEGative:负极性

返回：

查询返回 ARINC429 解码极性。

示例：

:BUS1:A429:POLarity POSitive /*设置 BUS1 的 ARINC429 解码极性为 POSitive*/

:BUS1:A429:POLarity ? /*查询返回 POSitive*/

:BUS<n>:A429:LEVel:MODe

功能描述：设置或查询 ARINC429 解码模式。

指令：

:BUS<n>:A429:MODe <type>

:BUS<n>:A429:MODe ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{19 | 21 | 23}

返回：

查询返回 ARINC429 解码模式。

示例：

:BUS1:A429:MODe 21 /*设置 BUS1 的 ARINC429 解码模式为 21BIT*/

:BUS1:A429:MODe ? /*查询返回 21*/

:BUS<n>:USB

:BUS<n>:USB:SOURce<n>

功能描述：设置或查询解码信源。

指令：

:BUS<n>:USB:SOURce <value>

:BUS<n>:USB:SOURce ?

参数：

<n>= 整型，范围 1~2

<n>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回解码信源。

示例：

:BUS1:USB:SOURce C2 /*设置 BUS1 的 USB 解码信源为 C2*/

:BUS1:USB:SOURce ? /*查询返回 C2*/

:BUS<n>:USB:THReshold:HIGH

功能描述：设置或查询 USB 解码高阈值，默认单位 V。

指令：

:BUS<n>:USB:THReshold:HIGH <value>

:BUS<n>:USB:THReshold:HIGH ?

参数：

<n>= 整型，范围 1~2

<value>= 实数型，单位 mV、V

返回：

查询返回解码高阈值。

示例：

:BUS1:USB:THReshold:HIGH 2 /*设置解码高阈值为 2V*/

:BUS1:USB:THReshold:HIGH ? /*查询返回 2.00000E+000*/

:BUS<n>:USB:THReshold:LOW

功能描述：设置或查询 USB 解码低阈值，默认单位 V。

指令：

:BUS<n>:USB:THReshold:LOW <value>

:BUS<n>:USB:THReshold:LOW ?

参数：

<n>= 整型，范围 1~2

<value>= 实数型，单位 mV、V

返回：

查询返回解码低阈值。

示例：

:BUS1:USB:THReshold:LOW 1 /*设置解码低阈值为 1V*/

:BUS1:USB:THReshold:LOW ? /*查询返回 1.00000E+000*/

:BUS<n>:SENT

:BUS<n>:SENT:SOURce

功能描述：设置或查询 SENT 解码信源。

指令：

:BUS<n>:SENT:SOURce <source>

:BUS<n>:SENT:SOURce ?

参数：

<n>= 整型，范围 1~2

<source>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回 SENT 解码信源。

示例：

:BUS1:SENT:SOURce C2 /*设置 BUS1 的 SENT 解码信源为 C2*/

:BUS1:SENT:SOURce ? /*查询返回 C2*/

:BUS<n>:SENT:CLOCKtick:CURRent

功能描述：设置或查询 SENT 解码时钟周期，默认单位为 us。

指令：

:BUS<n>:SENT:CLOCLKtick:CURRent <value>

:BUS<n>:SENT:CLOCKtick:CURRent ?

参数：

<n>= 整型，范围 1~2

<value>= 离散型{CUSTom | 1 | 3 | 10 | 30 | 100 | 300}

返回：

查询返回 SENT 解码时钟周期。

示例：

:BUS1:SENT:CLOCKtick:CURRent 10 /*设置 BUS1 的 SENT 解码时钟周期为 10us*/

:BUS1:SENT:CLOCKtick:CURRent ? /*查询返回 10*/

:BUS<n>:SENT:CLOCKtick:CUSTom

功能描述：设置或查询 SENT 解码自定义时钟周期大小，默认单位为 us。

指令：

:BUS<n>:SENT:CLOCKtick:CUSTom <value>

:BUS<n>:SENT:CLOCKtick:CUSTom ?

参数：

<n>= 整型，范围 1~2

<value>= 整数型，范围 50ns-300us

返回：

查询返回 SENT 解码时钟周期，以科学计数法表示。

示例：

:BUS1:SENT:CLOCKtick:CUSTom 1.25E-4 /*设置 BUS1 的 SENT 解码时钟周期为 125us*/

:BUS1:SENT:CLOCKtick:CUSTom ? /*查询返回 1.25000E-004*/

:BUS<n>:SENT:POLarity

功能描述：设置或查询 SENT 解码极性。

指令：

:BUS<n>:SENT:POLarity <polarity>

:BUS<n>:SENT:POLarity ?

参数：

<n>= 整型，范围 1~2

<polarity>= 离散型{POSitive | NEGative}

- POSitive:正极性
- NEGative:负极性

返回：

查询返回 SENT 解码极性。

示例：

:BUS1:SENT:POLarity POSitive /*设置 BUS1 的 SENT 解码极性为 POSitive*/

:BUS1:SENT:POLarity ? /*查询返回 POSitive*/

:BUS<n>:SENT:THReshold

功能描述：设置或查询 SENT 解码阈值，默认单位 V。

指令：

:BUS<n>:SENT:THReshold <value>

:BUS<n>:SENT:THReshold ?

参数：

<n>= 整型，范围 1~2

<value>= 实型，范围-12V~12V

返回：

查询返回 SENT 解码阈值。

示例：

:BUS1:SENT:THReshold 1 /*设置 BUS1 的 SENT 解码阈值为 1V*/

:BUS1:SENT:THReshold ? /*查询返回 1.00000E+000*/

:BUS<n>:SENT:DLENgth

功能描述：设置或查询 SENT 解码数据长度。

指令：

:BUS<n>:SENT:DLENgth <value>

:BUS<n>:SENT:DLENgth ?

参数：

<n>= 整型，范围 1~2

<value>= 离散型{1 | 2 | 3 | 4 | 5 | 6}

返回：

查询返回 SENT 数据长度。

示例：

:BUS1:SENT:DLENgth 2 /*设置 BUS1 的 SENT 解码数据长度为 2Nibbles*/

:BUS1:SENT:DLENgth ? /*查询返回 2*/

:BUS<n>:SENT:PAUSe

功能描述：设置或查询 SENT 解码有无暂停位。

指令：

:BUS<n>:SENT:PAUSe <type>

:BUS<n>:SENT:PAUSe ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{NO | YES}

- NO:无
- YES:有

返回：

查询返回 SENT 解码暂停位。

示例：

:BUS1:SENT:PAUSe YES /*设置 BUS1 的 SENT 解码有暂停位*/

:BUS1:SENT:PAUSe ? /*查询返回 YES*/

:BUS<n>:SENT:MODe

功能描述：设置或查询 SENT 解码模式。

指令：

:BUS<n>:SENT:MODe <value>

:BUS<n>:SENT:MODe ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{FAST | SLOW}

- FAST:快速通道
- SLOW:慢速通道

返回：

查询返回 SENT 解码模式。

示例：

:BUS1:SENT:MODe FAST /*设置 BUS1 的 SENT 解码模式为 FAST*/

:BUS1:SENT:MODe ? /*查询返回 FAST*/

:BUS<n>:SENT:SEGment

功能描述：设置或查询 SENT 数据段格式。

指令：

:BUS<n>:SENT:SEGment <value>

:BUS<n>:SENT:SEGment ?

参数：

<n>= 整型，范围 1~2

<type>= 离散型{NIB | FAST}

- NIB:半字节
- FAST:快速通道

返回：

查询返回 SENT 数据段格式。

示例：

:BUS1:SENT:SEGment FAST /*设置 BUS1 的 SENT 数据段格式为 FAST*/

:BUS1:SENT:SEGment ? /*查询返回 FAST*/

:BUS<n>:SENT:EBXTol

功能描述：设置或查询 SENT 时钟容差。

指令：

:BUS<n>:SENT:EBXTol <value>

:BUS<n>:SENT:EBXTol ?

参数：

<n>= 整型，范围 1~2

<value>= 实型，范围 1%-30%

返回：

查询返回 SENT 时钟容差。

示例：

:BUS1:SENT:EBXTol 20 /*设置 BUS1 的 SENT 时钟容差为 20%*/

:BUS1:SENT:EBXTol ? /*查询返回 20*/

:BUS<n>:SPMi

:BUS<n>:SPMi:SOURce:SDA

功能描述：设置或查询解码数据源。

指令：

:BUS<n>:SPMi:SOURce:SDA <value>

:BUS<n>:SPMi:SOURce:SDA ?

参数：

<value>=离散型{C1 | C2 | C3 | C4}

返回：

查询返回解码数据源。

示例：

:BUS1:SPMi:SOURce:SDA C2 /*设置解码数据源为 C2*/

:BUS1:SPMi:SOURce:SDA ? /*查询返回 C2*/

:BUS<n>:SPMi:SOURce:SCL

功能描述：设置或查询解码时钟信源。

指令：

:BUS<n>:SPMi:SOURce:SCL <value>

:BUS<n>:SPMi:SOURce:SCL ?

参数：

<value>=离散型{C1 | C2 | C3 | C4}

返回：

查询返回解码时钟信源。

示例：

:BUS1:SPMi:SOURce:SCL C2 /*设置解码时钟信源为 C2*/

:BUS1:SPMi:SOURce:SCL ? /*查询返回 C2*/

:BUS<n>:SPMi:DLEVel

功能描述：设置或查询解码数据阈值,默认单位 V。

指令：

:BUS<n>:SPMi:DLEVel <value>

:BUS<n>:SPMi:DLEVel ?

参数：

<value>=离散型 单位 mV、V

返回：

查询返回解码数据阈值。

示例：

:BUS1:SPMi:DLEVel 1 /*设置解码数据阈值为 1V*/

:BUS1:SPMi:DLEVel ? /*查询返回 1.00000E+000*/

:BUS<n>:SPMi:CLEVel

功能描述：设置或查询解码时钟阈值,默认单位 V。

指令：

:BUS<n>:SPMi:CLEVel <value>

:BUS<n>:SPMi:CLEVel ?

参数：

<value>=离散型 单位 mV、V

返回：

查询返回解码数时钟阈值。

示例：

:BUS1:SPMi:CLEVel 1 /*设置解码时钟阈值为 1V*/

:BUS1:SPMi:CLEVel? /*查询返回 1.00000E+000*/

:BUS<n>:SPMi:VERSion

功能描述：设置或查询 SPMI 版本。

指令：

:BUS<n>:SPMi:VERSion <value>

:BUS<n>:SPMi:VERSion ?

参数：

<value>=离散型{SPMI1 | SPMI2}

- SPMI1:1.0
- SPMI2:2.0

返回：

查询返回 SPMI 版本。

示例：

:BUS1:SPMi:VERSion SMPI1 /*设置 SPMI 版本为 1.0 */

:BUS1:SPMi:VERSion? /*查询返回 SPMI1*/

:BUS<n>:ETHernet

:BUS<n>:ETHernet:TYPe

功能描述：设置或查询信号类型。

指令：

:BUS<n>:ETHernet:TYPe <value>

:BUS<n>:ETHernet:TYPe ?

参数：

<value>=离散型{SINGle | DIFFerence}

- SINGle:单端信号
- DIFFerence:差分信号

返回：

查询返回信号类型。

示例：

:BUS1:ETHernet:TYPe DIFFerence /*设置总线 1 信号类型为差分信号*/

:BUS1:ETHernet:TYPe ? /*查询返回 DIFFerence*/

:BUS<n>:ETHernet:SOURce

功能描述：设置或查询输入信号源通道。

指令：

:BUS<n>:ETHernet:SOURce<ch><value>

:BUS<n>:ETHernet:TYPe ?

参数：

<ch>= 离散型{1 | 2}

- 1:信源 1
- 2:信源 2

<value>= 离散型{C1 | C2| C3 | C4}

返回：

查询返回解码信源。

示例：

:BUS1:ETHernet:SOURce2 C2 /*设置总线 1 解码信源 2 为 C2*/

:BUS1:ETHernet:SOURce2 ? /*查询返回 C2*/

说明：配置以太网协议解码所使用的信号源输入通道，对应以太网物理层 D+/D- 差分信号采集

:BUS<n>:ETHernet:THReshold:HIGH

功能描述：设置或查询高阈值，默认单位 V。

指令：

:BUS<n>:ETHernet:THReshold:HIGH <value>

:BUS<n>:ETHernet:THReshold:HIGH ?

参数：

<value>= 实型，单位 mV、V

返回：

查询返回解码高阈值。

示例：

:BUS1:ETHernet:THReshold:HIGH 2 /*设置总线 1 解码高阈值为 2V*/

:BUS1:ETHernet:THReshold:HIGH ? /*查询返回 2.00000E+000*/

:BUS<n>:ETHernet:THReshold:LOW

功能描述：设置或查询低阈值，默认单位 V。

指令：

:BUS<n>:ETHernet:THReshold:LOW <value>

:BUS<n>:ETHernet:THReshold:LOW ?

参数：

<value>= 实型，单位 mV、V

返回：

查询返回解码低阈值。

示例：

:BUS1:ETHernet:THReshold:LOW 2 /*设置总线 1 解码低阈值为 2V*/

:BUS1:ETHernet:THReshold:LOW ? /*查询返回 2.00000E+000*/

:BUS<n>:NRZ

:BUS<n>:NRZ:SOURce

功能描述：设置或查询解码信源。

指令：

:BUS<n>:NRZ:SOURce <value>

:BUS<n>:NRZ:SOURce ?

参数：

<value>=离散型{C1 | C2 | C3 | C4}

返回：

查询返回解码信源。

示例：

:BUS1:NRZ:SOURce C2 /*设置总线 1 解码信源为 C2*/

:BUS1:NRZ:SOURce ? /*查询返回 C2*/

:BUS<n>:NRZ:THReshold

功能描述：设置或查询解码阈值,默认单位 V。

指令：

:BUS<n>:NRZ:THReshold <value>

:BUS<n>:NRZ:THReshold ?

参数：

<value>=实型，单位 mV、V

返回：

查询返回解码阈值。

示例：

:BUS1:NRZ:THReshold 12V /*设置总线 1 解码阈值为 12V*/

:BUS1:NRZ:THReshold ? /*查询返回 1.20000E+001*/

:BUS<n>:NRZ:LEVel

功能描述：设置或查询空闲电平。

指令：

:BUS<n>:NRZ:LEVel <value>

:BUS<n>:NRZ:LEVel ?

参数：

<value>= 离散型{Negative | Positive}

- Negative:低电平
- Positive:高电平

返回：

查询返回空闲电平。

示例：

:BUS1:NRZ:LEVel Positive /*设置总线 1 空闲电平为高电平*/

:BUS1:NRZ:LEVel ? /*查询返回 Positive*/

:BUS<n>:NRZ:ORDer

功能描述：设置或查询解码位顺序。

指令：

:BUS<n>:NRZ:ORDer <value>

:BUS<n>:NRZ:ORDerl ?

参数：

<value>= 离散型{MSB | LSB}

返回：

查询返回解码位顺序。

示例：

:BUS1:NRZ:ORDer LSB /*设置总线 1 解码位顺序位 LSB*/

:BUS1:NRZ:ORDer ? /*查询返回 LSB*/

:BUS<n>:NRZ:BAUDrate

功能描述：设置或查询波特率，默认单位 bps。

指令：

:BUS<n>:NRZ:BAUDrate <value>

:BUS<n>:NRZ:BAUDratel ?

参数：

<value>= 整数型

返回：

查询返回波特率。

示例：

:BUS1:NRZ:BAUDrate 9600 /*设置总线 1 波特率为 9600*/

:BUS1:NRZ:BAUDrate ? /*查询返回 9.60000E+003*/

:BUS<n>:Manchester

:BUS<n>:MANC:SOURce

功能描述：设置或查询解码信源。

指令：

:BUS<n>:MANC:SOURce <value>

:BUS<n>:MANC:SOURce ?

参数：

<value>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回解码信源。

示例：

:BUS1:MANC:SOURce C2 /*设置总线 1 解码信源为 C2*/

:BUS1:MANC:SOURce ? /*查询返回 C2*/

:BUS<n>:MANC:POLarity

功能描述：设置或查询极性。

指令：

:BUS<n>:MANC:POLarity <value>

:BUS<n>:MANC:POLarity ?

参数：

<value>=离散型{Rising | Falling}

- Rising:上升沿
- Falling:下降沿

返回：

查询返回极性。

示例：

:BUS1:MANC:POLarity Rising /*设置总线 1 极性为上升沿*/

:BUS1:MANC:POLarity ? /*查询返回 Rising*/

:BUS<n>:MANC:BAUDrate

功能描述：设置或查询波特率，默认单位 bps。

指令：

:BUS<n>:MANC:BAUDratey <value>

:BUS<n>:MANC:BAUDrate ?

参数：

<value>= 整数型

返回：

查询返回波特率。

示例：

:BUS1:MANC:BAUDrate 9600 /*设置总线 1 波特率为 9600bps*/

:BUS1:MANC:BAUDrate ? /*查询返回 9.60000E+003*/

:BUS<n>:MANC:LEVel

功能描述：设置或查询解码阈值,默认单位 V。

指令：

:BUS<n>:MANC:LEVel <value>

:BUS<n>:MANC:LEVel ?

参数：

<value>= 实型，单位 mV、V

返回：

查询返回解码阈值。

示例：

:BUS1:MANC:LEVel 10V /*设置总线 1 解码阈值为 10V*/

:BUS1:MANC:LEVel ? /*查询返回 1.00000E+001*/

:BUS<n>:MANC:STARt

功能描述：设置或查询开始索引。

指令：

:BUS<n>:MANC:STARt <value>

:BUS<n>:MANC:STARt ?

参数：

<value>= 实型，范围是 1~8

返回：

查询返回开始索引。

示例：

:BUS1:MANC:STARt 4 /*设置总线 1 开始索引为 4*/

:BUS1:MANC:STARt ? /*查询返回 4*/

:BUS<n>:MANC:TOLerance

功能描述：设置或查询容限。

指令：

:BUS<n>:MANC:TOLerance <value>

:BUS<n>:MANC:TOLerance ?

参数：

<value>= 实型，范围是 1~50

返回：

查询返回容限。

示例：

:BUS1:MANC:TOLerance 20 /*设置总线 1 容限 20% */

:BUS1:MANC:TOLerance ? /*查询返回 2.00000E+001*/

:BUS<n>:MANC:IDLe

功能描述：设置或查询空闲位。

指令：

:BUS<n>:MANC:IDLe <value>

:BUS<n>:MANC:IDLe ?

参数：

<value>= 实型，范围是 1.2~32

返回：

查询返回空闲位。

示例：

:BUS1:MANC:IDLe 1.2 /*设置总线 1 空闲位为 1.2bits */

:BUS1:MANC:IDLe ? /*查询返回 1.20000E+000*/

:BUS<n>:MANC:DATAVIEW

功能描述：设置或查询数据包示图。

指令：

:BUS<n>:MANC:DATAVIEW <value>

:BUS<n>:MANC:DATAVIEW ?

参数：

<value>= 离散型{OPEN | CLOSe}

- OPEN:打开数据包示图
- CLOSe:关闭数据包示图

返回：

查询返回数据包示图状态。

示例：

:BUS1:MANC:DATAVIEW OPEN /*设置总线 1 打开数据包示图*/

:BUS1:MANC:DATAVIEW ? /*查询返回 OPEN*/

:BUS<n>:MANC:SYNC

功能描述：设置或查询同步位。

指令：

:BUS<n>:MANC:SYNC <value>

:BUS<n>:MANC:SYNC ?

参数：

<value>= 整数型，范围是 0~32

返回：

查询返回同步位。

示例：

:BUS1:MANC:SYNC 2 /*设置总线 1 同步位为 2bits*/

:BUS1:MANC:SYNC ? /*查询返回 2*/

:BUS<n>:MANC:DLENgth

功能描述：设置或查询字数。

指令：

:BUS<n>:MANC:DLENgth <value>

:BUS<n>:MANC:DLENgth ?

参数：

<value>= 整数型，范围是 1~255

返回：

查询返回字数。

示例：

:BUS1:MANC:DLENgth 2 /*设置总线 1 字数为 2*/

:BUS1:MANC:DLENgth ? /*查询返回 2*/

:BUS<n>:MANC:WSIZe

功能描述：设置或查询字大小。

指令：

:BUS<n>:MANC:WSIZe <value>

:BUS<n>:MANC:WSIZe ?

参数：

<value>= 整数型

返回：

查询返回字大小。

示例：

:BUS1:MANC:WSIZe 2 /*设置总线 1 字大小为 2bits*/

:BUS1:MANC:WSIZe ? /*查询返回 2*/

:BUS<n>:MANC:HEAD

功能描述：设置或查询标头。

指令：

:BUS<n>:MANC:HEAD <value>

:BUS<n>:MANC:HEAD ?

参数：

<value>= 整数型，范围是 0~128

返回：

查询返回标头。

示例：

:BUS1:MANC:HEAD 2 /*设置总线 1 标头为 2bits*/

:BUS1:MANC:HEAD ? /*查询返回 2*/

:BUS<n>:MANC:TAIL

功能描述：设置或查询帧尾。

指令：

:BUS<n>:MANC:TAIL <value>

:BUS<n>:MANC:TAIL ?

参数：

<value>= 整数型

返回：

查询返回帧尾。

示例：

:BUS1:MANC:TAIL 2 /*设置总线 1 帧尾为 2bits*/

:BUS1:MANC:TAIL ? /*查询返回 2*/

:BUS<n>:MANC:ORDer

功能描述：设置或查询位序。

指令：

:BUS<n>:MANC:ORDer <value>

:BUS<n>:MANC:ORDer ?

参数：

<value>= 离散型{MSB | LSB}

返回：

查询返回位序。

示例：

:BUS1:MANC:ORDer LSB /*设置总线 1 设置位序为 LSB*/

:BUS1:MANC:ORDer ? /*查询返回 LSB*/

:BUS<n>:MANC:PARity

功能描述： 设置或查询解码奇偶校验位。

指令：

:BUS<n>:MANC:PARity <value>

:BUS<n>:MANC:PARity ?

参数：

<value>= 离散型{NONe | ODD | EVEN}

- NONe:无校验
- ODD:奇校验
- EVEN:偶校验

返回：

查询返回解码奇偶校验。

示例：

:BUS1:MANC:PARity NONe /*设置总线 1 解码奇偶校验为无*/

:BUS1:MANC:PARity ? /*查询返回 NONe*/

:BUS<n>:8B10B

:BUS<n>:8B10B:SOURce

功能描述： 设置或查询解码信源。

指令：

:BUS<n>:8B10B:SOURce <value>

:BUS<n>:8B10B:SOURce ?

参数：

<value>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回解码信源。

示例：

:BUS1:8B10B:SOURce C2 /*设置总线 1 解码信源为 C2*/

:BUS1:8B10B:SOURce ? /*查询返回 C2*/

:BUS<n>:8B10B:BAUDrate

功能描述：设置或查询波特率，默认单位 bps。

指令：

:BUS<n>:8B10B:BAUDrate <value>

:BUS<n>:8B10B:BAUDrate ?

参数：

<value>= 整数型

返回：

查询返回波特率。

示例：

:BUS1:8B10B:BAUDrate 9600 /*设置总线 1 波特率为 9600bps*/

:BUS1:8B10B:BAUDrate ? /*查询返回 9.60000E+003*/

:BUS<n>:8B10B:LEVel

功能描述：设置或查询解码阈值,默认单位 V。

指令：

:BUS<n>:8B10B:LEVel <value>

:BUS<n>:8B10B:LEVel ?

参数：

<value>= 实型，单位 mV、V

返回：

查询返回解码阈值。

示例：

:BUS1:8B10B:LEVel 10V /*设置总线 1 解码阈值为 10V*/

:BUS1:8B10B:LEVel ? /*查询返回 1.00000E+001*/

:BUS<n>:PSI5

:BUS<n>:PSI5:MODe

功能描述：设置或查询信号速率模式。

指令：

:BUS<n>:PSI5:MODe <value>

:BUS<n>:PSI5:MODe ?

参数：

<value>= 离散型{STANDard | FAST}

- STANDard:标准模式（236kbps）
- FAST:快速模式（189kbps）

返回：

查询返回信号速率模式。

示例：

:BUS1:PSI5:MODe FAST /*设置总线 1 信号速率模式为快速模式*/

:BUS1:PSI5:MODe ? /*查询返回 FAST*/

:BUS<n>:PSI5:SOURce

功能描述：设置或查询输入信号源通道。

指令：

:BUS<n>:PSI5:SOURce <value>

:BUS<n>:PSI5:SOURce ?

参数：

<value>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回信源。

示例：

:BUS1:PSI5:SOURce C2 /*设置总线 1 信源为 C2*/

:BUS1:PSI5:SOURce ? /*查询返回 C2*/

:BUS<n>:PSI5:THReshold

功能描述：设置或查询阈值，默认单位 V。

指令：

:BUS<n>:PSI5:THReshold <value>

:BUS<n>:PSI5:THReshold ?

参数：

<value>= 实型，单位 mV、V

返回：

查询返回解码阈值。

示例：

:BUS1:PSI5:THReshold 2 /*设置总线 1 解码阈值为 2V*/

:BUS1:PSI5:THReshold ? /*查询返回 2.00000E+000*/

:BUS<n>:PSI5:SMSG

功能描述：设置或查询串行消息。

指令：

:BUS<n>:PSI5:SMSG <value>

:BUS<n>:PSI5:SMSG ?

参数：

<value>= 离散型{OFF | ON}

- OFF:关闭串行消息
- ON:打开串行消息

返回：

查询返回串行消息。

示例：

:BUS1:PSI5:SMSG OFF /*设置总线 1 关闭串行消息为 OFF*/

:BUS1:PSI5:SMSG ? /*查询返回 OFF*/

:BUS<n>:PSI5:STATus

功能描述：设置或查询状态位。

指令：

:BUS<n>:PSI5:STATus <value>

:BUS<n>:PSI5:STATus ?

参数：

<value>= 离散型{BIT0| BIT1 | BIT2}

返回：

查询返回状态位。

示例：

:BUS1:PSI5:STATus BIT1 /*设置状态位为为 1*/

:BUS1:PSI5:STATus ? /*查询返回 1*/

:BUS<n>:PSI5:FCONtrol

功能描述：设置或查询帧控制。

指令：

:BUS<n>:PSI5:FCONtrol <value>

:BUS<n>:PSI5:FCONtrol ?

参数：

<value>= 离散型{BIT0| BIT1 | BIT2 | BIT3 | BIT4}

返回：

查询返回帧控制。

示例：

:BUS1:PSI5:FCONtrol BIT2 /*设置总线 1 帧控制为 2*/

:BUS1:PSI5:FCONtrol ? /*查询返回 BIT2*/

:BUS<n>:PSI5:DSIZe<ch>

功能描述：设置或查询输数据域。

指令：

:BUS<n>:PSI5:DSIZe<ch> <value>

:BUS<n>:PSI5:DSIZe<ch> ?

参数：

<ch>= 离散型{1 | 2}

- 1:数据 A
- 2:数据 B

<value>= 整数型

返回：

查询返回数据域。

示例：

:BUS1:PSI5:DSIZe1 13 /*设置总线 1 数据域 A 为 13*/

:BUS1:PSI5:DSIZe1 ? /*查询返回 13*/

:BUS<n>:SMBus

:BUS<n>:SMBus:SOURce:CLK

功能描述：设置或查询时钟源。

指令：

:BUS<n>:SMBus:SOURce:CLK <value>

:BUS<n>:SMBus:SOURce:CLK ?

参数：

<value>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回时钟源。

示例：

:BUS1:SMBus:SOURce:CLK C2 /*设置总线 1 时钟源为 C2*/

:BUS1:SMBus:SOURce:CLK ? /*查询返回 C2*/

:BUS<n>:SMBus:SOURce:DATA

功能描述：设置或查询数据源。

指令：

:BUS<n>:SMBus:SOURce:DATA <value>

:BUS<n>:SMBus:SOURce:DATA ?

参数：

<value>= 离散型{C1 | C2 | C3 | C4}

返回：

查询返回数据源。

示例：

:BUS1:SMBus:SOURce:DATA C2 /*设置总线 1 数据源为 C2*/

:BUS1:SMBus:SOURce:DATA ? /*查询返回 C2*/

:BUS<n>:SMBus:THReshold:CLK

功能描述：设置或查询时钟阈值,默认单位 V。

指令：

:BUS<n>:SMBus:THReshold:CLK <value>

:BUS<n>:SMBus:THReshold:CLK ?

参数：

<value>= 离散型，单位 mV、V

返回：

查询返回时钟阈值。

示例：

:BUS1:SMBus:THReshold:CLK 1 /*设置总线 1 时钟阈值为 1V*/

:BUS1:SMBus:THReshold:CLK ? /*查询返回 1.00000E+000*/

:BUS<n>:SMBus:THReshold:DATA

功能描述：设置或查询数据阈值,默认单位 V。

指令：

:BUS<n>:SMBus:THReshold:DATA <value>

:BUS<n>:SMBus:THReshold:DATA ?

参数：

<value>= 离散型，单位 mV、V

返回：

查询返回数据阈值。

示例：

:BUS1:SMBus:THReshold:DATA 1 /*设置总线 1 数据阈值为 1V*/

:BUS1:SMBus:THReshold:DATA ? /*查询返回 1.00000E+000*/

:BUS<n>:SMBus:PECByte

功能描述：设置或查询 PEC 字节启用状态。

指令：

:BUS<n>:SMBus:PECByte <value>

:BUS<n>:SMBus:PECByte ?

参数：

<value>=离散型{OPEN | CLOSe}

- OPEN:启用
- CLOSe:关闭

返回：

查询返回 PEC 字节启用状态。

示例：

:BUS1:SMBus:PECByte OPEN /*设置 PEC 字节启用状态为启用*/

:BUS1:SMBus:PECByte ? /*查询返回 OPEN*/

AUX 指令系统

用于设置或查询示波器的辅助输入/输出功能。

:AUX:INPut:TYPe

功能描述：设置或查询辅助输入类型。

指令：

:AUX:INPut:TYPe <type>

:AUX:INPut:TYPe ?

参数：

<type>= 离散型{CLOSe | TRIGger | AWGSync}

- CLOSe:关闭
- TRIGger:触发同步
- AWGSync:AWG 外触发

返回：

查询返回辅助输入类型。

示例：

:AUX:INPut:TYPe TRIGger /*设置辅助输入为触发同步*/

:AUX:INPut:TYPe ? /*查询返回 TRIGger*/

:AUX:INPut:POLarity

功能描述：设置或查询输入极性。

指令：

:AUX:INPut:POLarity <type>

:AUX:INPut:POLarity ?

参数：

<type>= 离散型{POSitive | NEGative}

- POSitive:正极性
- NEGative:负极性

返回：

查询返回输入极性。

示例：

:AUX:INPut:POLarity NEGative /*设置输入极性为负极性*/

:AUX:INPut:POLarity ? /*查询返回 NEGative*/

:AUX:OUTPut:TYPe

功能描述：设置或查询辅助输出类型。

指令：

:AUX:OUTPut:TYPe <type>

:AUX:OUTPut:TYPe ?

参数：

<type>= 离散型{CLOSe | TRIGger | AWGSync | PFTest}

- CLOSe:关闭
- TRIGger:触发同步
- AWGSync:AWG 触发
- PFTest:通过测试

返回：

查询返回辅助输出类型。

示例：

:AUX:OUTPut:TYPe TRIGger /*设置辅助输出为触发同步*/

:AUX:OUTPut:TYPe ? /*查询返回 TRIGger*/

:AUX:OUTPut:POLarity

功能描述：设置或查询输出极性。

指令：

:AUX:OUTPut:POLarity <type>

:AUX:OUTPut:POLarity ?

参数：

<type>= 离散型{POSitive | NEGative}

- POSitive:正极性
- NEGative:负极性

返回：

查询返回输出极性。

示例：

:AUX:OUTPut:POLarity POSitive /*设置输出极性为正极性*/

:AUX:OUTPut:POLarity ? /*查询返回 POSitive*/

NETWork 指令系统

用于设置或查询示波器以太网参数。

:NETWork:LAN:COUNt

功能描述：查询当前处于已连接状态的以太网网络适配器的个数。

指令：

:NETWork:LAN:COUNt ?

:NETWork:LAN:SELect

功能描述：设置或查询当前选中的以太网网络适配器。

指令：

:NETWork:LAN:SELect <index>

:NETWork:LAN:SELect ?

参数：

<index>= 整数型，index 小于或等于查询到的已连接的以太网网络适配器个数

返回：

查询返回输出极性。

示例：

:NETWork:LAN:SELect 1 /*设置选中第一个以太网网络适配器*/

:NETWork:LAN:SELect ? /*查询返回 1*/

:NETWork:LAN:RESet

功能描述：用于立即重设当前设置的网络参数到默认参数。

指令：

:NETWork:LAN:RESet

:NETWork:LAN:APPLy

功能描述：设置已选中的适配器立即生效当前设置的网络参数。

指令：

:NETWork:LAN:APPLy

:NETWork:LAN:IPADdress

功能描述：用于设置或查询已选中的适配器的 IP 地址，格式为 xxx.xxx.xxx.xxx。

指令：

:NETWork:LAN:IPADdress <adress>

:NETWork:LAN:IPADdress ?

参数：

<adress>= 字符串类型

返回：

查询返回 IP 地址。

示例：

:NETWork:LAN:IPADdress "192.168.137.10" /*设置 IP 地址为 192.168.137.10*/

:NETWork:LAN:IPADdress ? /*查询返回 192.168.137.10*/

:NETWork:LAN:SMASK

功能描述：设置或查询已选中的适配器的子网掩码，格式为 xxx.xxx.xxx.xxx。

指令：

:NETWork:LAN:SMASK <smask>

:NETWork:LAN:SMASK ?

参数：

<smask>= 字符串类型

返回：

查询返回子网掩码。

示例：

:NETWork:LAN:SMASK "255.255.255.0" /*设置子网掩码为 255.255.255.0*/

:NETWork:LAN:SMASK ? /*查询返回 255.255.255.0*/

:NETWork:LAN:GATEway

功能描述：设置或查询已选中的适配器的默认网关，格式为 xxx.xxx.xxx.xxx。

指令：

:NETWork:LAN:GATEway <gateway>

:NETWork:LAN:GATEway ?

参数：

<gateway>= 字符串类型

返回：

查询返回默认网关。

示例：

:NETWork:LAN:GATEway 192.168.137.1 /*设置默认网关为 192.168.137.1*/

:NETWork:LAN:GATEway ? /*查询返回 192.168.137.1*/

:NETWork:LAN:DHCP

功能描述：设置或查询已选中的适配器的配置模式（自动 IP）和（手动 IP）。

指令：

:NETWork:LAN:DHCP <type>

:NETWork:LAN:DHCP ?

参数：

<type>= 离散型{AUTO | MANual}

- AUTO:自动
- MANual:手动

返回：

查询返回动态配置模式。

示例：

:NETWork:LAN:DHCP AUTO /*设置配置模式为 AUTO*/

:NETWork:LAN:DHCP ? /*查询返回 AUTO*/

:NETWork:LAN:MAC

功能描述：用于查询 MAC 物理地址。

指令：

:NETWork:LAN:MAC ?

UNI-T

公司简介

优利德科技(中国)股份有限公司(以下简称为“公司”)成立于 2003 年，总部位于东莞松山湖，是一家集仪器仪表自主研发、生产、销售为一体的国家高新技术企业，主要包括通用仪表、专业仪表、温度及环境测试仪表、测试仪器四大产品线，广泛应用于电子、家用电器、机电设备、节能环保、轨道交通、汽车制造、冷暖通、建筑工程、5G 新基建、新能源、物联网、大数据中心、人工智能、电力建设及维护、高等教育和科学研究等领域。

公司拥有东莞、成都及常州三处研发中心。凭借深厚的技术积淀和较强的研发实力，公司累计参与四项国家标准的起草工作，获得三次中国专利优秀奖，连续两次被评为国家知识产权优势企业。同时，公司还获批设立了广东省博士工作站、博士后创新实践基地，并获得了“广东省仪器仪表工程技术研究中心”、广东省“创新型中小企业”、广东省“专精特新中小企业”、“广东省制造业单项冠军企业”、广东省专精特新“小巨人”企业、“省级企业技术中心”、国家级专精特新“小巨人”企业等认定。

截至 2024 年 12 月 31 日，公司累计获得专利 526 项，其中发明专利 96 项、实用新型专利 195 项. 外观设计专利 235 项;拥有软件著作权 33 项，核心技术 25 项，具备一定的技术领先优势。

企业规模

公司在东莞及河源合计拥有约 10 万平方米的生产基地，建立了先进、完备的产品生产和质量控制体系，合计设计年产能达到 1000 万台以上，确保为全球用户提供了持续稳定的产品供应保障。

公司理念

公司一直秉承为全球用户提供高质量、高安全性、高可靠性、高性价比的测试测量产品及综合解决方案，坚持以科技及人文为本，致力于成为世界一流的仪器仪表民族品牌。

销售网络

优利德测试仪器销售网络

联系我们

UNI-T 技术支持热线： 400-876-7822

声明：

UNI-T 是优利德科技（中国）股份有限公司的英文名称和商标。
本文档中的产品信息可不经通知而变更，有关 UNI-T 最新的产品、
应用、服务等方面的信息请访问 UNI-T 官方网站：www.uni-trend.com.cn