



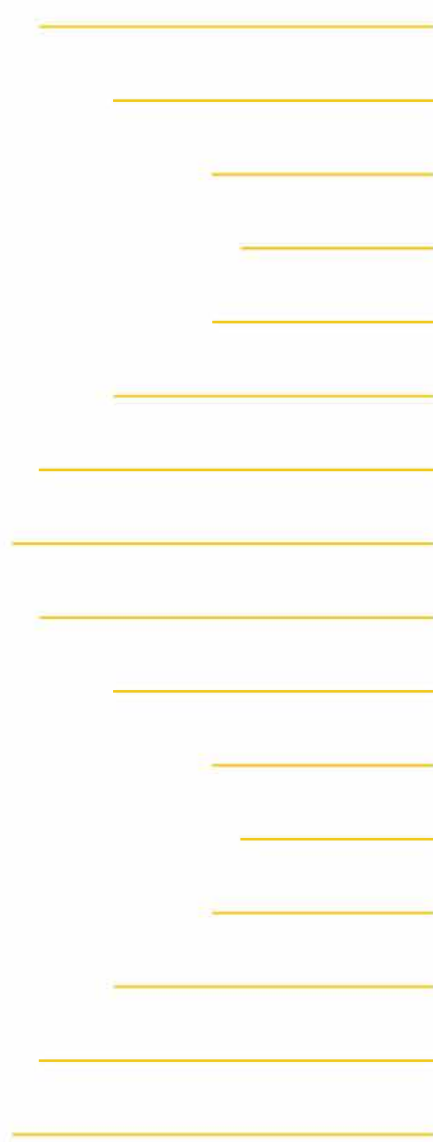
RIGOL

DHO1000/DHO4000 Series

Digital Oscilloscope

Programming Guide

Nov. 2022



Guaranty and Declaration

Copyright

© 2022 RIGOL TECHNOLOGIES CO., LTD. All Rights Reserved.

Trademark Information

RIGOL® is the trademark of RIGOL TECHNOLOGIES CO., LTD.

Notices

- RIGOL products are covered by P.R.C. and foreign patents, issued and pending.
- RIGOL reserves the right to modify or change parts of or all the specifications and pricing policies at the company's sole decision.
- Information in this publication replaces all previously released materials.
- Information in this publication is subject to change without notice.
- RIGOL shall not be liable for either incidental or consequential losses in connection with the furnishing, use, or performance of this manual, as well as any information contained.
- Any part of this document is forbidden to be copied, photocopied, or rearranged without prior written approval of RIGOL.

Product Certification

RIGOL guarantees that this product conforms to the national and industrial standards in China as well as the ISO9001:2015 standard and the ISO14001:2015 standard. Other international standard conformance certifications are in progress.

| Section | Description | Page |
|---------|------------------------------|------|
| | List of Figures..... | XX |
| 1 | Document Overview..... | 1 |
| 2 | SCPI Command Overview..... | 3 |
| 3 | Command System..... | 6 |
| 3.1 | :CLEAr..... | 6 |
| 3.2 | :RUN..... | 6 |
| 3.3 | :STOP..... | 7 |
| 3.4 | :SINGLE..... | 8 |
| 3.5 | :TFORce..... | 8 |
| 3.6 | :AUToset Commands..... | 9 |
| 3.6.1 | :AUToset..... | 9 |
| 3.6.2 | :AUToset:PEAK..... | 10 |
| 3.6.3 | :AUToset:OPENch..... | 10 |
| 3.6.4 | :AUToset:OVERlap..... | 11 |
| 3.6.5 | :AUToset:KEEPcoup..... | 12 |
| 3.6.6 | :AUToset:LOCK..... | 12 |
| 3.6.7 | :AUToset:ENABle..... | 13 |
| 3.7 | :ACQuire Commands..... | 14 |
| 3.7.1 | :ACQuire:AVERages..... | 14 |
| 3.7.2 | :ACQuire:BITS..... | 15 |
| 3.7.3 | :ACQuire:MDEPth..... | 16 |
| 3.7.4 | :ACQuire:MEMDepth?..... | 17 |
| 3.7.5 | :ACQuire:TYPE..... | 18 |
| 3.7.6 | :ACQuire:SRATe?..... | 19 |
| 3.7.7 | :ACQuire:ULTRa:MODE..... | 19 |
| 3.7.8 | :ACQuire:ULTRa:TIMEout..... | 21 |
| 3.7.9 | :ACQuire:ULTRa:MAXFrame..... | 21 |
| 3.8 | :BUS<n> Commands..... | 22 |
| 3.8.1 | :BUS<n>:MODE..... | 22 |
| 3.8.2 | :BUS<n>:DISPlay..... | 23 |
| 3.8.3 | :BUS<n>:FORMat..... | 23 |
| 3.8.4 | :BUS<n>:EVENT..... | 24 |
| 3.8.5 | :BUS<n>:LABel..... | 25 |

| | | |
|----------|---------------------------|----|
| 3.8.6 | :BUS<n>:DATA? | 25 |
| 3.8.7 | :BUS<n>:EEXPort | 26 |
| 3.8.8 | :BUS<n>:POSition | 27 |
| 3.8.9 | :BUS<n>:THReshold | 28 |
| 3.8.10 | :BUS<n>:PARAllel | 30 |
| 3.8.10.1 | :BUS<n>:PARAllel:BUS | 30 |
| 3.8.10.2 | :BUS<n>:PARAllel:CLK | 30 |
| 3.8.10.3 | :BUS<n>:PARAllel:SLOPe | 31 |
| 3.8.10.4 | :BUS<n>:PARAllel:WIDTh | 32 |
| 3.8.10.5 | :BUS<n>:PARAllel:BITX | 32 |
| 3.8.10.6 | :BUS<n>:PARAllel:SOURce | 33 |
| 3.8.10.7 | :BUS<n>:PARAllel:ENDian | 34 |
| 3.8.10.8 | :BUS<n>:PARAllel:POLarity | 35 |
| 3.8.11 | :BUS<n>:RS232 | 35 |
| 3.8.11.1 | :BUS<n>:RS232:TX | 35 |
| 3.8.11.2 | :BUS<n>:RS232:RX | 36 |
| 3.8.11.3 | :BUS<n>:RS232:POLarity | 37 |
| 3.8.11.4 | :BUS<n>:RS232:PARity | 37 |
| 3.8.11.5 | :BUS<n>:RS232:ENDian | 38 |
| 3.8.11.6 | :BUS<n>:RS232:BAUD | 39 |
| 3.8.11.7 | :BUS<n>:RS232:DBITs | 40 |
| 3.8.11.8 | :BUS<n>:RS232:SBITs | 40 |
| 3.8.12 | :BUS<n>:IIC | 41 |
| 3.8.12.1 | :BUS<n>:IIC:SCLK:SOURce | 41 |
| 3.8.12.2 | :BUS<n>:IIC:SDA:SOURce | 42 |
| 3.8.12.3 | :BUS<n>:IIC:EXCHange | 42 |
| 3.8.12.4 | :BUS<n>:IIC:ADDResS | 43 |
| 3.8.13 | :BUS<n>:SPI | 44 |
| 3.8.13.1 | :BUS<n>:SPI:SCLK:SOURce | 44 |
| 3.8.13.2 | :BUS<n>:SPI:SCLK:SLOPe | 44 |
| 3.8.13.3 | :BUS<n>:SPI:MISO:SOURce | 45 |
| 3.8.13.4 | :BUS<n>:SPI:MOSI:SOURce | 46 |
| 3.8.13.5 | :BUS<n>:SPI:POLarity | 46 |
| 3.8.13.6 | :BUS<n>:SPI:MISO:POLarity | 47 |
| 3.8.13.7 | :BUS<n>:SPI:MOSI:POLarity | 48 |
| 3.8.13.8 | :BUS<n>:SPI:DBITs | 49 |
| 3.8.13.9 | :BUS<n>:SPI:ENDian | 49 |

| | | |
|-----------|---------------------------------|----|
| 3.8.13.10 | :BUS<n>:SPI:MODE..... | 50 |
| 3.8.13.11 | :BUS<n>:SPI:TIMEout:TIME..... | 51 |
| 3.8.13.12 | :BUS<n>:SPI:SS:SOURce..... | 51 |
| 3.8.13.13 | :BUS<n>:SPI:SS:POLarity..... | 52 |
| 3.8.14 | :BUS<n>:CAN..... | 53 |
| 3.8.14.1 | :BUS<n>:CAN:SOURce..... | 53 |
| 3.8.14.2 | :BUS<n>:CAN:STYPe..... | 54 |
| 3.8.14.3 | :BUS<n>:CAN:BAUD..... | 55 |
| 3.8.14.4 | :BUS<n>:CAN:SPOint..... | 55 |
| 3.8.14.5 | :BUS<n>:CAN:FDBaud..... | 56 |
| 3.8.14.6 | :BUS<n>:CAN:FDSPoint..... | 57 |
| 3.8.15 | :BUS<n>:FLEXray..... | 57 |
| 3.8.15.1 | :BUS<n>:FLEXray:BAUD..... | 58 |
| 3.8.15.2 | :BUS<n>:FLEXray:SOURce..... | 58 |
| 3.8.15.3 | :BUS<n>:FLEXray:SPOint..... | 59 |
| 3.8.15.4 | :BUS<n>:FLEXray:STYPe..... | 60 |
| 3.8.15.5 | :BUS<n>:FLEXray:CHANnel..... | 60 |
| 3.8.16 | :BUS<n>:LIN..... | 61 |
| 3.8.16.1 | :BUS<n>:LIN:PARity..... | 61 |
| 3.8.16.2 | :BUS<n>:LIN:SOURce..... | 62 |
| 3.8.16.3 | :BUS<n>:LIN:STANdard..... | 63 |
| 3.8.17 | :BUS<n>:IIS..... | 63 |
| 3.8.17.1 | :BUS<n>:IIS:SOURce:CLOCK..... | 63 |
| 3.8.17.2 | :BUS<n>:IIS:SOURce:DATA..... | 64 |
| 3.8.17.3 | :BUS<n>:IIS:SOURce:WSElect..... | 65 |
| 3.8.17.4 | :BUS<n>:IIS:ALIGnment..... | 65 |
| 3.8.17.5 | :BUS<n>:IIS:CLOCK:SLOPe..... | 66 |
| 3.8.17.6 | :BUS<n>:IIS:RWIDth..... | 67 |
| 3.8.17.7 | :BUS<n>:IIS:RECewidth..... | 67 |
| 3.8.17.8 | :BUS<n>:IIS:WSLow..... | 68 |
| 3.8.17.9 | :BUS<n>:IIS:ENDian..... | 69 |
| 3.8.17.10 | :BUS<n>:IIS:POLarity..... | 69 |
| 3.8.18 | :BUS<n>:M1553..... | 70 |
| 3.8.18.1 | :BUS<n>:M1553:SOURce..... | 70 |
| 3.9 | :CHANnel<n> Commands..... | 71 |
| 3.9.1 | :CHANnel<n>:BWLimit..... | 71 |
| 3.9.2 | :CHANnel<n>:COUPling..... | 72 |

| | | |
|---------|------------------------------------|----|
| 3.9.3 | :CHANnel<n>:DISPlay..... | 73 |
| 3.9.4 | :CHANnel<n>:INVert..... | 73 |
| 3.9.5 | :CHANnel<n>:OFFSet..... | 74 |
| 3.9.6 | :CHANnel<n>:TCALibrate..... | 75 |
| 3.9.7 | :CHANnel<n>:SCALe..... | 76 |
| 3.9.8 | :CHANnel<n>:IMPedance..... | 77 |
| 3.9.9 | :CHANnel<n>:BVOLTage..... | 78 |
| 3.9.10 | :CHANnel<n>:CStArt..... | 78 |
| 3.9.11 | :CHANnel<n>:PROBe..... | 79 |
| 3.9.12 | :CHANnel<n>:PROBe:DELay..... | 80 |
| 3.9.13 | :CHANnel<n>:PROBe:BIAS..... | 80 |
| 3.9.14 | :CHANnel<n>:PROBe:CALibration..... | 81 |
| 3.9.15 | :CHANnel<n>:PROBe:DEMag..... | 82 |
| 3.9.16 | :CHANnel<n>:LABel:SHOW..... | 82 |
| 3.9.17 | :CHANnel<n>:LABel:CONtent..... | 83 |
| 3.9.18 | :CHANnel<n>:UNITs..... | 83 |
| 3.9.19 | :CHANnel<n>:VERNier..... | 84 |
| 3.9.20 | :CHANnel<n>:POSition..... | 85 |
| 3.10 | :COUNter Commands..... | 86 |
| 3.10.1 | :COUNter:CURRent?..... | 86 |
| 3.10.2 | :COUNter:ENABle..... | 86 |
| 3.10.3 | :COUNter:SOURce..... | 87 |
| 3.10.4 | :COUNter:MODE..... | 88 |
| 3.10.5 | :COUNter:NDIGits..... | 88 |
| 3.10.6 | :COUNter:TOTAlize:ENABle..... | 89 |
| 3.10.7 | :COUNter:TOTAlize:CLear..... | 90 |
| 3.11 | :CURSor Commands..... | 90 |
| 3.11.1 | :CURSor:MODE..... | 90 |
| 3.11.2 | :CURSor:MEASure:INDicator..... | 91 |
| 3.11.3 | :CURSor:MANual:TYPE..... | 91 |
| 3.11.4 | :CURSor:MANual:SOURce..... | 92 |
| 3.11.5 | :CURSor:MANual:CAX..... | 93 |
| 3.11.6 | :CURSor:MANual:CAY..... | 93 |
| 3.11.7 | :CURSor:MANual:CBX..... | 94 |
| 3.11.8 | :CURSor:MANual:CBY..... | 95 |
| 3.11.9 | :CURSor:MANual:AXValue?..... | 95 |
| 3.11.10 | :CURSor:MANual:AYValue?..... | 96 |

| | | |
|---------|------------------------------|-----|
| 3.11.11 | :CURSor:MANual:BXValue?..... | 96 |
| 3.11.12 | :CURSor:MANual:BYValue?..... | 97 |
| 3.11.13 | :CURSor:MANual:XDELta?..... | 98 |
| 3.11.14 | :CURSor:MANual:IXDelta?..... | 98 |
| 3.11.15 | :CURSor:MANual:YDELta?..... | 99 |
| 3.11.16 | :CURSor:TRACk:SOURce1..... | 99 |
| 3.11.17 | :CURSor:TRACk:SOURce2..... | 100 |
| 3.11.18 | :CURSor:TRACk:CAX..... | 100 |
| 3.11.19 | :CURSor:TRACk:CBX..... | 101 |
| 3.11.20 | :CURSor:TRACk:CAY..... | 102 |
| 3.11.21 | :CURSor:TRACk:CBY..... | 102 |
| 3.11.22 | :CURSor:TRACk:AXValue?..... | 103 |
| 3.11.23 | :CURSor:TRACk:AYValue?..... | 103 |
| 3.11.24 | :CURSor:TRACk:BXValue?..... | 104 |
| 3.11.25 | :CURSor:TRACk:BYValue?..... | 104 |
| 3.11.26 | :CURSor:TRACk:XDELta?..... | 105 |
| 3.11.27 | :CURSor:TRACk:YDELta?..... | 105 |
| 3.11.28 | :CURSor:TRACk:IXDelta?..... | 106 |
| 3.11.29 | :CURSor:TRACk:MODE..... | 106 |
| 3.11.30 | :CURSor:XY:AX..... | 107 |
| 3.11.31 | :CURSor:XY:BX..... | 108 |
| 3.11.32 | :CURSor:XY:AY..... | 108 |
| 3.11.33 | :CURSor:XY:BY..... | 109 |
| 3.11.34 | :CURSor:XY:AXValue?..... | 110 |
| 3.11.35 | :CURSor:XY:AYValue?..... | 110 |
| 3.11.36 | :CURSor:XY:BXValue?..... | 110 |
| 3.11.37 | :CURSor:XY:BYValue?..... | 111 |
| 3.11.38 | :CURSor:XY:XDELta?..... | 111 |
| 3.11.39 | :CURSor:XY:YDELta?..... | 112 |
| 3.12 | :DISPlay Commands..... | 112 |
| 3.12.1 | :DISPlay:CLear..... | 112 |
| 3.12.2 | :DISPlay:TYPE..... | 113 |
| 3.12.3 | :DISPlay:GRADing:TIME..... | 114 |
| 3.12.4 | :DISPlay:WBRightness..... | 115 |
| 3.12.5 | :DISPlay:GRID..... | 115 |
| 3.12.6 | :DISPlay:GBRightness..... | 116 |
| 3.12.7 | :DISPlay:DATA?..... | 116 |

| | | |
|---------|--------------------------------|-----|
| 3.12.8 | :DISPlay:RULers..... | 117 |
| 3.12.9 | :DISPlay:COLor..... | 118 |
| 3.12.10 | :DISPlay:WHOLd..... | 118 |
| 3.13 | :DVM Commands..... | 119 |
| 3.13.1 | :DVM:CURRent?..... | 119 |
| 3.13.2 | :DVM:ENABle..... | 119 |
| 3.13.3 | :DVM:SOURce..... | 120 |
| 3.13.4 | :DVM:MODE..... | 121 |
| 3.14 | IEEE488.2 Common Commands..... | 121 |
| 3.14.1 | *IDN?..... | 121 |
| 3.14.2 | *RST..... | 122 |
| 3.14.3 | *CLS..... | 122 |
| 3.14.4 | *ESE..... | 123 |
| 3.14.5 | *ESR?..... | 123 |
| 3.14.6 | *OPC..... | 124 |
| 3.14.7 | *RCL..... | 125 |
| 3.14.8 | *SAV..... | 125 |
| 3.14.9 | *SRE..... | 126 |
| 3.14.10 | *STB?..... | 126 |
| 3.14.11 | *WAI..... | 127 |
| 3.14.12 | *TST?..... | 127 |
| 3.15 | :LAN Commands..... | 128 |
| 3.15.1 | :LAN:DHCP..... | 128 |
| 3.15.2 | :LAN:AUTOip..... | 129 |
| 3.15.3 | :LAN:GATeway..... | 129 |
| 3.15.4 | :LAN:DNS..... | 130 |
| 3.15.5 | :LAN:MAC?..... | 131 |
| 3.15.6 | :LAN:DSERver?..... | 131 |
| 3.15.7 | :LAN:MANual..... | 132 |
| 3.15.8 | :LAN:IPADdress..... | 132 |
| 3.15.9 | :LAN:SMASk..... | 133 |
| 3.15.10 | :LAN:STATus?..... | 134 |
| 3.15.11 | :LAN:VISA?..... | 135 |
| 3.15.12 | :LAN:MDNS..... | 135 |
| 3.15.13 | :LAN:HOST:NAME..... | 136 |
| 3.15.14 | :LAN:DESCRiption..... | 137 |
| 3.15.15 | :LAN:APPLY..... | 137 |

| | | |
|---------|-----------------------------------|-----|
| 3.16 | :MASK Commands..... | 138 |
| 3.16.1 | :MASK:ENABle..... | 138 |
| 3.16.2 | :MASK:SOURce..... | 138 |
| 3.16.3 | :MASK:OPERate..... | 139 |
| 3.16.4 | :MASK:X..... | 140 |
| 3.16.5 | :MASK:Y..... | 140 |
| 3.16.6 | :MASK:CREate..... | 141 |
| 3.16.7 | :MASK:RESet..... | 141 |
| 3.16.8 | :MASK:FAILED?..... | 142 |
| 3.16.9 | :MASK:PASSed?..... | 142 |
| 3.16.10 | :MASK:TOTAL?..... | 143 |
| 3.16.11 | :MASK:OUTPut:ENABle..... | 143 |
| 3.16.12 | :MASK:OUTPut:EVENT..... | 144 |
| 3.16.13 | :MASK:OUTPut:TIME..... | 144 |
| 3.17 | :MATH<n> Commands..... | 145 |
| 3.17.1 | :MATH<n>:DISPlay..... | 145 |
| 3.17.2 | :MATH<n>:OPERator..... | 146 |
| 3.17.3 | :MATH<n>:SOURce1..... | 146 |
| 3.17.4 | :MATH<n>:SOURce2..... | 147 |
| 3.17.5 | :MATH<n>:LSOURce1..... | 148 |
| 3.17.6 | :MATH<n>:LSOURce2..... | 149 |
| 3.17.7 | :MATH<n>:SCALe..... | 150 |
| 3.17.8 | :MATH<n>:OFFSet..... | 150 |
| 3.17.9 | :MATH<n>:INVert..... | 151 |
| 3.17.10 | :MATH<n>:RESet..... | 152 |
| 3.17.11 | :MATH<n>:EXPand..... | 152 |
| 3.17.12 | :MATH<n>:WAVetype..... | 153 |
| 3.17.13 | :MATH<n>:FFT:SOURce..... | 154 |
| 3.17.14 | :MATH<n>:FFT:WINDow..... | 154 |
| 3.17.15 | :MATH<n>:FFT:UNIT..... | 155 |
| 3.17.16 | :MATH<n>:FFT:SCALe..... | 156 |
| 3.17.17 | :MATH<n>:FFT:OFFSet..... | 157 |
| 3.17.18 | :MATH<n>:FFT:HSCale..... | 157 |
| 3.17.19 | :MATH<n>:FFT:HCENter..... | 158 |
| 3.17.20 | :MATH<n>:FFT:FREQuency:START..... | 159 |
| 3.17.21 | :MATH<n>:FFT:FREQuency:END..... | 159 |
| 3.17.22 | :MATH<n>:FFT:SEARch:ENABle..... | 160 |

| | | |
|---------|------------------------------------|-----|
| 3.17.23 | :MATH<n>:FFT:SEARch:NUM..... | 161 |
| 3.17.24 | :MATH<n>:FFT:SEARch:THReshold..... | 161 |
| 3.17.25 | :MATH<n>:FFT:SEARch:EXCursion..... | 162 |
| 3.17.26 | :MATH<n>:FFT:SEARch:ORDER..... | 163 |
| 3.17.27 | :MATH<n>:FFT:SEARch:RES?..... | 163 |
| 3.17.28 | :MATH<n>:FILTer:TYPE..... | 164 |
| 3.17.29 | :MATH<n>:FILTer:W1..... | 165 |
| 3.17.30 | :MATH<n>:FILTer:W2..... | 166 |
| 3.17.31 | :MATH<n>:SENSitivity..... | 167 |
| 3.17.32 | :MATH<n>:DISTance..... | 168 |
| 3.17.33 | :MATH<n>:THReshold1..... | 169 |
| 3.17.34 | :MATH<n>:THReshold2..... | 169 |
| 3.17.35 | :MATH<n>:THReshold3..... | 170 |
| 3.17.36 | :MATH<n>:THReshold4..... | 171 |
| 3.17.37 | :MATH<n>:WINDow:TITLe?..... | 172 |
| 3.17.38 | :MATH<n>:LABel:SHOW..... | 173 |
| 3.18 | :MEASure Commands..... | 173 |
| 3.18.1 | :MEASure:SOURce..... | 173 |
| 3.18.2 | :MEASure:ITEM..... | 174 |
| 3.18.3 | :MEASure:CLear..... | 175 |
| 3.18.4 | :MEASure:AMSource..... | 176 |
| 3.18.5 | :MEASure:STATistic:COUNT..... | 177 |
| 3.18.6 | :MEASure:STATistic:DISPlay..... | 177 |
| 3.18.7 | :MEASure:STATistic:RESet..... | 178 |
| 3.18.8 | :MEASure:STATistic:ITEM..... | 178 |
| 3.18.9 | :MEASure:SETup:MAX..... | 180 |
| 3.18.10 | :MEASure:SETup:MID..... | 181 |
| 3.18.11 | :MEASure:SETup:MIN..... | 181 |
| 3.18.12 | :MEASure:SETup:PSA..... | 182 |
| 3.18.13 | :MEASure:SETup:PSB..... | 183 |
| 3.18.14 | :MEASure:SETup:DSA..... | 184 |
| 3.18.15 | :MEASure:SETup:DSB..... | 184 |
| 3.18.16 | :MEASure:THReshold:SOURce..... | 185 |
| 3.18.17 | :MEASure:THReshold:TYPE..... | 186 |
| 3.18.18 | :MEASure:THReshold:DEFault..... | 186 |
| 3.18.19 | :MEASure:AREA..... | 187 |
| 3.18.20 | :MEASure:INDicator..... | 188 |

| | | |
|---------|--|-----|
| 3.18.21 | :MEASure:COUNter:ENABle..... | 188 |
| 3.18.22 | :MEASure:COUNter:SOURce..... | 189 |
| 3.18.23 | :MEASure:COUNter:VALue?..... | 189 |
| 3.18.24 | :MEASure:AMP:TYPE..... | 190 |
| 3.18.25 | :MEASure:AMP:MANual:TOP..... | 190 |
| 3.18.26 | :MEASure:AMP:MANual:BASE..... | 191 |
| 3.18.27 | :MEASure:CATegory..... | 192 |
| 3.19 | :POWer Commands..... | 192 |
| 3.19.1 | :POWer:TYPE..... | 192 |
| 3.19.2 | :POWer:CURRent:SOURce..... | 193 |
| 3.19.3 | :POWer:VOLTage:SOURce..... | 194 |
| 3.19.4 | :POWer:REFLevel:METHod..... | 194 |
| 3.19.5 | :POWer:REFLevel:ABSolute:HIGH..... | 195 |
| 3.19.6 | :POWer:REFLevel:ABSolute:LOW..... | 196 |
| 3.19.7 | :POWer:REFLevel:ABSolute:MID..... | 196 |
| 3.19.8 | :POWer:REFLevel:PERCent:HIGH..... | 197 |
| 3.19.9 | :POWer:REFLevel:PERCent:LOW..... | 197 |
| 3.19.10 | :POWer:REFLevel:PERCent:MID..... | 198 |
| 3.19.11 | :POWer:QUALity:DISPlay..... | 199 |
| 3.19.12 | :POWer:QUALity:FREQ:REFerence..... | 199 |
| 3.19.13 | :POWer:QUALity:STATistics:RESult?..... | 200 |
| 3.19.14 | :POWer:RIPple:SOURce..... | 201 |
| 3.19.15 | :POWer:RIPple:DISPlay..... | 201 |
| 3.19.16 | :POWer:RIPple:STATistics:RESult?..... | 202 |
| 3.19.17 | :POWer:STATistics:RESet..... | 202 |
| 3.20 | :QUICK Command..... | 203 |
| 3.20.1 | :QUICK:OPERation..... | 203 |
| 3.21 | :RECord Commands..... | 204 |
| 3.21.1 | :RECord:WRECord:ENABle..... | 204 |
| 3.21.2 | :RECord:ENABle..... | 204 |
| 3.21.3 | :RECord:WRECord:OPERate..... | 205 |
| 3.21.4 | :RECord:STARt..... | 205 |
| 3.21.5 | :RECord:WRECord:FRAMes..... | 206 |
| 3.21.6 | :RECord:FRAMes..... | 207 |
| 3.21.7 | :RECord:WRECord:FRAMes:MAX..... | 207 |
| 3.21.8 | :RECord:WRECord: FMAX?..... | 208 |
| 3.21.9 | :RECord:WRECord:FINTErval..... | 208 |

| | | |
|---------|-------------------------------------|-----|
| 3.21.10 | :RECORD:WRECORD:PROMPT..... | 209 |
| 3.21.11 | :RECORD:WREPLAY:FCURRENT..... | 210 |
| 3.21.12 | :RECORD:CURRENT..... | 210 |
| 3.21.13 | :RECORD:WREPLAY:FCURRENT:TIME?..... | 211 |
| 3.21.14 | :RECORD:WREPLAY:FSTART..... | 211 |
| 3.21.15 | :RECORD:WREPLAY:FEND..... | 212 |
| 3.21.16 | :RECORD:WREPLAY:FMAX?..... | 213 |
| 3.21.17 | :RECORD:WREPLAY:FINTEVAL..... | 213 |
| 3.21.18 | :RECORD:WREPLAY:MODE..... | 214 |
| 3.21.19 | :RECORD:WREPLAY:DIRECTION..... | 214 |
| 3.21.20 | :RECORD:WREPLAY:OPERATE..... | 215 |
| 3.21.21 | :RECORD:PLAY..... | 216 |
| 3.21.22 | :RECORD:WREPLAY:BACK..... | 216 |
| 3.21.23 | :RECORD:WREPLAY:NEXT..... | 217 |
| 3.21.24 | :RECORD:WREPLAY:PLAY..... | 217 |
| 3.22 | :REFERENCE Commands..... | 218 |
| 3.22.1 | :REFERENCE:SOURCE..... | 218 |
| 3.22.2 | :REFERENCE:VSCALE..... | 218 |
| 3.22.3 | :REFERENCE:VOFFSET..... | 219 |
| 3.22.4 | :REFERENCE:RESET..... | 220 |
| 3.22.5 | :REFERENCE:CURRENT..... | 220 |
| 3.22.6 | :REFERENCE:SAVE..... | 221 |
| 3.22.7 | :REFERENCE:COLOR..... | 221 |
| 3.22.8 | :REFERENCE:LABEL:ENABLE..... | 222 |
| 3.22.9 | :REFERENCE:LABEL:CONTENT..... | 223 |
| 3.23 | :SAVE and :LOAD Commands..... | 223 |
| 3.23.1 | :SAVE:IMAGE:INVERT..... | 223 |
| 3.23.2 | :SAVE:IMAGE:COLOR..... | 224 |
| 3.23.3 | :SAVE:IMAGE:FORMAT..... | 225 |
| 3.23.4 | :SAVE:IMAGE:HEADER..... | 225 |
| 3.23.5 | :SAVE:IMAGE:DATA?..... | 226 |
| 3.23.6 | :SAVE:IMAGE..... | 227 |
| 3.23.7 | :SAVE:SETUP..... | 227 |
| 3.23.8 | :SAVE:WAVEFORM..... | 228 |
| 3.23.9 | :SAVE:MEMORY:WAVEFORM..... | 229 |
| 3.23.10 | :SAVE:STATUS?..... | 230 |
| 3.23.11 | :SAVE:OVERLAP..... | 230 |

| | | |
|---------|-----------------------------------|-----|
| 3.23.12 | :SAVE:PREFix..... | 231 |
| 3.23.13 | :LOAD:SETup..... | 232 |
| 3.24 | :SEARch Commands..... | 232 |
| 3.24.1 | :SEARch:COUNT?..... | 232 |
| 3.24.2 | :SEARch:STATe..... | 233 |
| 3.24.3 | :SEARch:MODE..... | 233 |
| 3.24.4 | :SEARch:EVENT..... | 234 |
| 3.24.5 | :SEARch:VALue?..... | 235 |
| 3.24.6 | :SEARch:EDGE:SLOPe..... | 235 |
| 3.24.7 | :SEARch:EDGE:SOURce..... | 236 |
| 3.24.8 | :SEARch:EDGE:THReshold..... | 236 |
| 3.24.9 | :SEARch:PULSe:POLarity..... | 237 |
| 3.24.10 | :SEARch:PULSe:QUALifier..... | 238 |
| 3.24.11 | :SEARch:PULSe:SOURce..... | 238 |
| 3.24.12 | :SEARch:PULSe:UWIDth..... | 239 |
| 3.24.13 | :SEARch:PULSe:LWIDth..... | 240 |
| 3.24.14 | :SEARch:PULSe:THReshold..... | 240 |
| 3.25 | :NAVigate Commands..... | 241 |
| 3.25.1 | :NAVigate:ENABLE..... | 241 |
| 3.25.2 | :NAVigate:MODE..... | 241 |
| 3.25.3 | :NAVigate:TIME:SPeed..... | 242 |
| 3.25.4 | :NAVigate:TIME:PLAY..... | 243 |
| 3.25.5 | :NAVigate:TIME:END..... | 244 |
| 3.25.6 | :NAVigate:TIME:START..... | 244 |
| 3.25.7 | :NAVigate:TIME:NEXT..... | 245 |
| 3.25.8 | :NAVigate:TIME:BACK..... | 245 |
| 3.25.9 | :NAVigate:SEARch:END..... | 245 |
| 3.25.10 | :NAVigate:SEARch:START..... | 246 |
| 3.25.11 | :NAVigate:SEARch:NEXT..... | 246 |
| 3.25.12 | :NAVigate:SEARch:BACK..... | 247 |
| 3.25.13 | :NAVigate:FRAME:DISPlay:MODE..... | 247 |
| 3.25.14 | :NAVigate:FRAME:END:FRAME..... | 249 |
| 3.25.15 | :NAVigate:FRAME:START:FRAME..... | 249 |
| 3.25.16 | :NAVigate:FRAME:END..... | 250 |
| 3.25.17 | :NAVigate:FRAME:START..... | 250 |
| 3.25.18 | :NAVigate:FRAME:NEXT..... | 251 |
| 3.25.19 | :NAVigate:FRAME:BACK..... | 251 |

| | | |
|---------|------------------------------------|-----|
| 3.25.20 | :NAVigate:FRAME:PLAY..... | 252 |
| 3.26 | :SYSTem Commands..... | 253 |
| 3.26.1 | :SYSTem:AOUTput..... | 253 |
| 3.26.2 | :SYSTem:BEEPer..... | 253 |
| 3.26.3 | :SYSTem:DATE..... | 254 |
| 3.26.4 | :SYSTem:ERRor[:NEXT]?..... | 255 |
| 3.26.5 | :SYSTem:RCLock..... | 255 |
| 3.26.6 | :SYSTem:GPIB..... | 256 |
| 3.26.7 | :SYSTem:PON..... | 256 |
| 3.26.8 | :SYSTem:PSTatus..... | 257 |
| 3.26.9 | :SYSTem:OPTion:INSTall..... | 258 |
| 3.26.10 | :SYSTem:OPTion:UNINStall..... | 258 |
| 3.26.11 | :SYSTem:OPTion:STATus?..... | 259 |
| 3.26.12 | :SYSTem:RAMount?..... | 260 |
| 3.26.13 | :SYSTem:RESet..... | 261 |
| 3.26.14 | :SYSTem:SETup..... | 261 |
| 3.26.15 | :SYSTem:TIME..... | 262 |
| 3.26.16 | :SYSTem:LOCKed..... | 263 |
| 3.26.17 | :SYSTem:MODules?..... | 263 |
| 3.26.18 | :SYSTem:AUToscale..... | 264 |
| 3.26.19 | :SYSTem:GAMount?..... | 264 |
| 3.26.20 | :SYSTem:STIME..... | 265 |
| 3.26.21 | :SYSTem:VERSion?..... | 265 |
| 3.26.22 | :SYSTem:DGSTatus?..... | 266 |
| 3.27 | :TIMebase Commands..... | 266 |
| 3.27.1 | :TIMebase:DELay:ENABle..... | 266 |
| 3.27.2 | :TIMebase:DELay:OFFSet..... | 267 |
| 3.27.3 | :TIMebase:DELay:SCALE..... | 268 |
| 3.27.4 | :TIMebase[:MAIN][:OFFSet]..... | 268 |
| 3.27.5 | :TIMebase[:MAIN]:SCALE..... | 269 |
| 3.27.6 | :TIMebase:MODE..... | 270 |
| 3.27.7 | :TIMebase:HREFerence:MODE..... | 271 |
| 3.27.8 | :TIMebase:HREFerence:POSition..... | 272 |
| 3.27.9 | :TIMebase:VERNier..... | 272 |
| 3.27.10 | :TIMebase:HOTKeys..... | 273 |
| 3.27.11 | :TIMebase:ROLL..... | 274 |
| 3.27.12 | :TIMebase:XY:ENABle..... | 274 |

| | | |
|-----------|------------------------------|-----|
| 3.27.13 | :TImEbase:XY:X..... | 275 |
| 3.27.14 | :TImEbase:XY:Y..... | 275 |
| 3.27.15 | :TImEbase:XY:Z..... | 276 |
| 3.28 | :TRIGger Commands..... | 277 |
| 3.28.1 | :TRIGger:MODE..... | 277 |
| 3.28.2 | :TRIGger:COUPling..... | 278 |
| 3.28.3 | :TRIGger:STATus?..... | 278 |
| 3.28.4 | :TRIGger:SWEEp..... | 279 |
| 3.28.5 | :TRIGger:HOLDoff..... | 280 |
| 3.28.6 | :TRIGger:NREJect..... | 281 |
| 3.28.7 | :TRIGger:POSition?..... | 281 |
| 3.28.8 | :TRIGger:EDGE..... | 282 |
| 3.28.8.1 | :TRIGger:EDGE:SOURce..... | 282 |
| 3.28.8.2 | :TRIGger:EDGE:SLOPe..... | 282 |
| 3.28.8.3 | :TRIGger:EDGE:LEVel..... | 283 |
| 3.28.9 | :TRIGger:PULSe..... | 284 |
| 3.28.9.1 | :TRIGger:PULSe:SOURce..... | 284 |
| 3.28.9.2 | :TRIGger:PULSe:POLarity..... | 284 |
| 3.28.9.3 | :TRIGger:PULSe:WHEN..... | 285 |
| 3.28.9.4 | :TRIGger:PULSe:UWIDth..... | 286 |
| 3.28.9.5 | :TRIGger:PULSe:LWIDth..... | 287 |
| 3.28.9.6 | :TRIGger:PULSe:LEVel..... | 287 |
| 3.28.10 | :TRIGger:SLOPe Commands..... | 288 |
| 3.28.10.1 | :TRIGger:SLOPe:SOURce..... | 288 |
| 3.28.10.2 | :TRIGger:SLOPe:POLarity..... | 289 |
| 3.28.10.3 | :TRIGger:SLOPe:WHEN..... | 289 |
| 3.28.10.4 | :TRIGger:SLOPe:TUPPer..... | 290 |
| 3.28.10.5 | :TRIGger:SLOPe:TLOWer..... | 291 |
| 3.28.10.6 | :TRIGger:SLOPe:WINDow..... | 292 |
| 3.28.10.7 | :TRIGger:SLOPe:ALEVel..... | 292 |
| 3.28.10.8 | :TRIGger:SLOPe:BLEVel..... | 293 |
| 3.28.11 | :TRIGger:VIDeo Commands..... | 294 |
| 3.28.11.1 | :TRIGger:VIDeo:SOURce..... | 294 |
| 3.28.11.2 | :TRIGger:VIDeo:POLarity..... | 294 |
| 3.28.11.3 | :TRIGger:VIDeo:MODE..... | 295 |
| 3.28.11.4 | :TRIGger:VIDeo:LINE..... | 296 |
| 3.28.11.5 | :TRIGger:VIDeo:STANDard..... | 297 |

| | | |
|-----------|--------------------------------|-----|
| 3.28.11.6 | :TRIGger:VIDeo:LEVel..... | 298 |
| 3.28.12 | :TRIGger:PATtern..... | 299 |
| 3.28.12.1 | :TRIGger:PATtern:PATtern..... | 299 |
| 3.28.12.2 | :TRIGger:PATtern:SOURce..... | 300 |
| 3.28.12.3 | :TRIGger:PATtern:LEVel..... | 300 |
| 3.28.13 | :TRIGger:DURation..... | 301 |
| 3.28.13.1 | :TRIGger:DURation:SOURce..... | 301 |
| 3.28.13.2 | :TRIGger:DURation:TYPE..... | 302 |
| 3.28.13.3 | :TRIGger:DURation:WHEN..... | 303 |
| 3.28.13.4 | :TRIGger:DURation:TUPPer..... | 304 |
| 3.28.13.5 | :TRIGger:DURation:TLOWer..... | 304 |
| 3.28.13.6 | :TRIGger:DURation:LEVel..... | 305 |
| 3.28.14 | :TRIGger:TIMeout..... | 306 |
| 3.28.14.1 | :TRIGger:TIMeout:SOURce..... | 306 |
| 3.28.14.2 | :TRIGger:TIMeout:SLOPe..... | 306 |
| 3.28.14.3 | :TRIGger:TIMeout:TIME..... | 307 |
| 3.28.14.4 | :TRIGger:TIMeout:LEVel..... | 308 |
| 3.28.15 | :TRIGger:RUNT..... | 309 |
| 3.28.15.1 | :TRIGger:RUNT:SOURce..... | 309 |
| 3.28.15.2 | :TRIGger:RUNT:POLarity..... | 309 |
| 3.28.15.3 | :TRIGger:RUNT:WHEN..... | 310 |
| 3.28.15.4 | :TRIGger:RUNT:WUPPer..... | 311 |
| 3.28.15.5 | :TRIGger:RUNT:WLOWer..... | 311 |
| 3.28.15.6 | :TRIGger:RUNT:ALEVel..... | 312 |
| 3.28.15.7 | :TRIGger:RUNT:BLEVel..... | 313 |
| 3.28.16 | :TRIGger:WINDows..... | 314 |
| 3.28.16.1 | :TRIGger:WINDows:SOURce..... | 314 |
| 3.28.16.2 | :TRIGger:WINDows:SLOPe..... | 314 |
| 3.28.16.3 | :TRIGger:WINDows:POSition..... | 315 |
| 3.28.16.4 | :TRIGger:WINDows:TIME..... | 316 |
| 3.28.16.5 | :TRIGger:WINDows:ALEVel..... | 316 |
| 3.28.16.6 | :TRIGger:WINDows:BLEVel..... | 317 |
| 3.28.17 | :TRIGger:DELay..... | 318 |
| 3.28.17.1 | :TRIGger:DELay:SA..... | 318 |
| 3.28.17.2 | :TRIGger:DELay:ASLop..... | 318 |
| 3.28.17.3 | :TRIGger:DELay:SB..... | 319 |
| 3.28.17.4 | :TRIGger:DELay:BSLop..... | 320 |

| | | |
|-----------|------------------------------|-----|
| 3.28.17.5 | :TRIGger:DElay:TYPE..... | 320 |
| 3.28.17.6 | :TRIGger:DElay:TUPPer..... | 321 |
| 3.28.17.7 | :TRIGger:DElay:TLOWer..... | 322 |
| 3.28.17.8 | :TRIGger:DElay:ALEvel..... | 323 |
| 3.28.17.9 | :TRIGger:DElay:BLEvel..... | 323 |
| 3.28.18 | :TRIGger:SHOLd..... | 324 |
| 3.28.18.1 | :TRIGger:SHOLd:DSRC..... | 324 |
| 3.28.18.2 | :TRIGger:SHOLd:CSRC..... | 324 |
| 3.28.18.3 | :TRIGger:SHOLd:SLOPe..... | 325 |
| 3.28.18.4 | :TRIGger:SHOLd:PATtern..... | 326 |
| 3.28.18.5 | :TRIGger:SHOLd:TYPE..... | 326 |
| 3.28.18.6 | :TRIGger:SHOLd:STIME..... | 327 |
| 3.28.18.7 | :TRIGger:SHOLd:HTIME..... | 328 |
| 3.28.18.8 | :TRIGger:SHOLd:DLEvel..... | 328 |
| 3.28.18.9 | :TRIGger:SHOLd:CLEvel..... | 329 |
| 3.28.19 | :TRIGger:NEDGe..... | 330 |
| 3.28.19.1 | :TRIGger:NEDGe:SOURce..... | 330 |
| 3.28.19.2 | :TRIGger:NEDGe:SLOPe..... | 330 |
| 3.28.19.3 | :TRIGger:NEDGe:IDLE..... | 331 |
| 3.28.19.4 | :TRIGger:NEDGe:EDGE..... | 332 |
| 3.28.19.5 | :TRIGger:NEDGe:LEvel..... | 332 |
| 3.28.20 | :TRIGger:RS232..... | 333 |
| 3.28.20.1 | :TRIGger:RS232:SOURce..... | 333 |
| 3.28.20.2 | :TRIGger:RS232:LEvel..... | 333 |
| 3.28.20.3 | :TRIGger:RS232:POLarity..... | 334 |
| 3.28.20.4 | :TRIGger:RS232:WHEN..... | 335 |
| 3.28.20.5 | :TRIGger:RS232:DATA..... | 335 |
| 3.28.20.6 | :TRIGger:RS232:BAUD..... | 336 |
| 3.28.20.7 | :TRIGger:RS232:WIDTh..... | 337 |
| 3.28.20.8 | :TRIGger:RS232:STOP..... | 337 |
| 3.28.20.9 | :TRIGger:RS232:PARity..... | 338 |
| 3.28.21 | :TRIGger:IIC..... | 338 |
| 3.28.21.1 | :TRIGger:IIC:SCL..... | 338 |
| 3.28.21.2 | :TRIGger:IIC:CLEvel..... | 339 |
| 3.28.21.3 | :TRIGger:IIC:SDA..... | 340 |
| 3.28.21.4 | :TRIGger:IIC:DLEvel..... | 340 |
| 3.28.21.5 | :TRIGger:IIC:WHEN..... | 341 |

| | | |
|------------|-----------------------------|-----|
| 3.28.21.6 | :TRIGger:IIC:AWIDth..... | 342 |
| 3.28.21.7 | :TRIGger:IIC:ADDResS..... | 343 |
| 3.28.21.8 | :TRIGger:IIC:DIRectIon..... | 343 |
| 3.28.21.9 | :TRIGger:IIC:DBYTes..... | 344 |
| 3.28.21.10 | :TRIGger:IIC:DATA..... | 344 |
| 3.28.21.11 | :TRIGger:IIC:CURRbit..... | 345 |
| 3.28.21.12 | :TRIGger:IIC:CODE..... | 346 |
| 3.28.22 | :TRIGger:SPI..... | 346 |
| 3.28.22.1 | :TRIGger:SPI:CLK..... | 346 |
| 3.28.22.2 | :TRIGger:SPI:SCL..... | 347 |
| 3.28.22.3 | :TRIGger:SPI:CLEVel..... | 348 |
| 3.28.22.4 | :TRIGger:SPI:SLOPe..... | 348 |
| 3.28.22.5 | :TRIGger:SPI:MISO..... | 349 |
| 3.28.22.6 | :TRIGger:SPI:SDA..... | 349 |
| 3.28.22.7 | :TRIGger:SPI:DLEVel..... | 350 |
| 3.28.22.8 | :TRIGger:SPI:WHEN..... | 351 |
| 3.28.22.9 | :TRIGger:SPI:CS..... | 351 |
| 3.28.22.10 | :TRIGger:SPI:SLEVel..... | 352 |
| 3.28.22.11 | :TRIGger:SPI:MODE..... | 353 |
| 3.28.22.12 | :TRIGger:SPI:TIMEout..... | 353 |
| 3.28.22.13 | :TRIGger:SPI:WIDTh..... | 354 |
| 3.28.22.14 | :TRIGger:SPI:DATA..... | 354 |
| 3.28.22.15 | :TRIGger:SPI:CURRbit..... | 355 |
| 3.28.22.16 | :TRIGger:SPI:CODE..... | 356 |
| 3.28.23 | :TRIGger:CAN..... | 356 |
| 3.28.23.1 | :TRIGger:CAN:BAUD..... | 356 |
| 3.28.23.2 | :TRIGger:CAN:SOURce..... | 357 |
| 3.28.23.3 | :TRIGger:CAN:STYPe..... | 358 |
| 3.28.23.4 | :TRIGger:CAN:WHEN..... | 358 |
| 3.28.23.5 | :TRIGger:CAN:SPOInt..... | 360 |
| 3.28.23.6 | :TRIGger:CAN:EXTended..... | 361 |
| 3.28.23.7 | :TRIGger:CAN:DEFine..... | 361 |
| 3.28.23.8 | :TRIGger:CAN:DWIDth..... | 362 |
| 3.28.23.9 | :TRIGger:CAN:DATA..... | 363 |
| 3.28.23.10 | :TRIGger:CAN:CURRbit..... | 364 |
| 3.28.23.11 | :TRIGger:CAN:CODE..... | 364 |
| 3.28.23.12 | :TRIGger:CAN:LEVel..... | 365 |

| | | |
|------------|----------------------------------|-----|
| 3.28.23.13 | :TRIGger:CAN:FDBaud..... | 366 |
| 3.28.23.14 | :TRIGger:CAN:FDSPoint..... | 366 |
| 3.28.24 | :TRIGger:LIN..... | 367 |
| 3.28.24.1 | :TRIGger:LIN:SOURce..... | 367 |
| 3.28.24.2 | :TRIGger:LIN:LEVel..... | 368 |
| 3.28.24.3 | :TRIGger:LIN:STANdard..... | 368 |
| 3.28.24.4 | :TRIGger:LIN:BAUD..... | 369 |
| 3.28.24.5 | :TRIGger:LIN:SAMPlepoint..... | 370 |
| 3.28.24.6 | :TRIGger:LIN:WHEN..... | 370 |
| 3.28.24.7 | :TRIGger:LIN:ERRor..... | 371 |
| 3.28.24.8 | :TRIGger:LIN:ID..... | 372 |
| 3.28.24.9 | :TRIGger:LIN:DATA..... | 372 |
| 3.28.24.10 | :TRIGger:LIN:CURRbit..... | 373 |
| 3.28.24.11 | :TRIGger:LIN:CODE..... | 374 |
| 3.28.25 | :TRIGger:IIS..... | 374 |
| 3.28.25.1 | :TRIGger:IIS:SOURce:CLOCK..... | 374 |
| 3.28.25.2 | :TRIGger:IIS:CLEVel..... | 375 |
| 3.28.25.3 | :TRIGger:IIS:SOURce:WSElect..... | 376 |
| 3.28.25.4 | :TRIGger:IIS:SLEVel..... | 376 |
| 3.28.25.5 | :TRIGger:IIS:SOURce:DATA..... | 377 |
| 3.28.25.6 | :TRIGger:IIS:DLEVel..... | 377 |
| 3.28.25.7 | :TRIGger:IIS:CLOCK:SLOPe..... | 378 |
| 3.28.25.8 | :TRIGger:IIS:AUDio..... | 379 |
| 3.28.25.9 | :TRIGger:IIS:WHEN..... | 379 |
| 3.28.25.10 | :TRIGger:IIS:UWIDth..... | 380 |
| 3.28.25.11 | :TRIGger:IIS:WIDTh..... | 381 |
| 3.28.25.12 | :TRIGger:IIS:ALIGNment..... | 381 |
| 3.28.25.13 | :TRIGger:IIS:DATA..... | 382 |
| 3.28.25.14 | :TRIGger:IIS:DMIN..... | 383 |
| 3.28.25.15 | :TRIGger:IIS:DMAX..... | 383 |
| 3.28.25.16 | :TRIGger:IIS:CODE..... | 384 |
| 3.28.26 | :TRIGger:FLEXray..... | 385 |
| 3.28.26.1 | :TRIGger:FLEXray:SOURce..... | 385 |
| 3.28.26.2 | :TRIGger:FLEXray:LEVel..... | 385 |
| 3.28.26.3 | :TRIGger:FLEXray:BAUD..... | 386 |
| 3.28.26.4 | :TRIGger:FLEXray:CH..... | 387 |
| 3.28.26.5 | :TRIGger:FLEXray:WHEN..... | 387 |

| | | |
|------------|-------------------------------|-----|
| 3.28.26.6 | :TRIGger:FLEXray:POS..... | 388 |
| 3.28.26.7 | :TRIGger:FLEXray:FRAMe..... | 389 |
| 3.28.26.8 | :TRIGger:FLEXray:SYMBol..... | 389 |
| 3.28.26.9 | :TRIGger:FLEXray:ERRor..... | 390 |
| 3.28.26.10 | :TRIGger:FLEXray:DEFine..... | 391 |
| 3.28.26.11 | :TRIGger:FLEXray:CYCComp..... | 391 |
| 3.28.26.12 | :TRIGger:FLEXray:MAXCy..... | 392 |
| 3.28.26.13 | :TRIGger:FLEXray:MINCy..... | 393 |
| 3.28.26.14 | :TRIGger:FLEXray:IDCMp..... | 393 |
| 3.28.26.15 | :TRIGger:FLEXray:MINid..... | 394 |
| 3.28.26.16 | :TRIGger:FLEXray:MAXid..... | 395 |
| 3.28.27 | :TRIGger:M1553..... | 396 |
| 3.28.27.1 | :TRIGger:M1553:SOURce..... | 396 |
| 3.28.27.2 | :TRIGger:M1553:WINDow..... | 396 |
| 3.28.27.3 | :TRIGger:M1553:ALEVel..... | 397 |
| 3.28.27.4 | :TRIGger:M1553:BLEVel..... | 398 |
| 3.28.27.5 | :TRIGger:M1553:POLarity..... | 399 |
| 3.28.27.6 | :TRIGger:M1553:WHEN..... | 399 |
| 3.28.27.7 | :TRIGger:M1553:SYNC..... | 400 |
| 3.28.27.8 | :TRIGger:M1553:ERRor..... | 401 |
| 3.28.27.9 | :TRIGger:M1553:DATComp..... | 401 |
| 3.28.27.10 | :TRIGger:M1553:DATValue..... | 402 |
| 3.28.27.11 | :TRIGger:M1553:DMAX..... | 403 |
| 3.28.27.12 | :TRIGger:M1553:DMIN..... | 403 |
| 3.28.27.13 | :TRIGger:M1553:DRTA..... | 404 |
| 3.28.27.14 | :TRIGger:M1553:DBIT..... | 405 |
| 3.28.27.15 | :TRIGger:M1553:CODE..... | 405 |
| 3.29 | :WAVEform Commands..... | 406 |
| 3.29.1 | :WAVEform:SOURce..... | 408 |
| 3.29.2 | :WAVEform:MODE..... | 408 |
| 3.29.3 | :WAVEform:FORMat..... | 409 |
| 3.29.4 | :WAVEform:POINts..... | 410 |
| 3.29.5 | :WAVEform:DATA?..... | 410 |
| 3.29.6 | :WAVEform:XINCrement?..... | 412 |
| 3.29.7 | :WAVEform:XORigin?..... | 413 |
| 3.29.8 | :WAVEform:XREFerence?..... | 413 |
| 3.29.9 | :WAVEform:YINCrement?..... | 414 |

| | | |
|---------|---------------------------------------|-----|
| 3.29.10 | :WAVeform:YORigin?..... | 415 |
| 3.29.11 | :WAVeform:YREFerence?..... | 415 |
| 3.29.12 | :WAVeform:START..... | 416 |
| 3.29.13 | :WAVeform:STOP..... | 417 |
| 3.29.14 | :WAVeform:PREamble?..... | 417 |
| 4 | Programming Examples..... | 419 |
| 4.1 | Programming Preparations..... | 419 |
| 4.2 | LabVIEW Programming Example..... | 419 |
| 4.3 | Visual Basic Programming Example..... | 423 |
| 4.4 | Visual C++ Programming Example..... | 425 |

List of Figures

| | | |
|------------|--|-----|
| Figure 3.1 | Parameter Definitions in NORMAL Mode | 407 |
| Figure 3.2 | Parameter Definitions in RAW Mode | 407 |

1 Document Overview

This manual introduces how to program and control DHO1000 and DHO4000 series via the remote interface by using the SCPI (Standard Commands for Programmable Instruments) commands. DHO1000 and DHO4000 series can communicate with PC through USB, LAN, or GPIB interface (with the USB to GPIB interface module provided by RIGOL).



TIP

For the newest version of this manual, download it

Publication Number

PGA34100-1110


Software Version

00.01.00

Software upgrade might change or add product features. Please acquire the latest version of the manual to upgrade the software.

Format Conventions in this Manual

1. Key


The front panel key is denoted by the menu key icon. For example,  indicates the "DEFAULT" key.

2. Menu

The menu item is denoted by the format of "Menu Name (Bold) + Character Shading" in the manual. For example, **Setup** indicates the "Setup" sub-menu under the "Utility" function menu. You can click or tap **Setup** to access the "Setup" menu.

3. Operation Procedures

The next step of the operation is denoted by ">" in the manual. For example, 

> **Storage** indicates first clicking or tapping  and then clicking or tapping **Storage**.

4. The front/rear panel connector is denoted by "Brackets + Connector Name (Bold)", for example, [**AUX OUT**].

Content Conventions in this Manual

DHO4000 series digital oscilloscope includes the following models. Unless otherwise specified, this manual takes DHO4804 as an example to illustrate the basic operation methods of DHO4000 series.

| Model | Max. Analog Bandwidth | Analog Channels |
|---------|-----------------------|-----------------|
| DHO4204 | 200 MHz | 4 |
| DHO4404 | 400 MHz | 4 |
| DHO4804 | 800 MHz | 4 |

DHO1000 series digital oscilloscope includes the following models. Unless otherwise specified, this manual takes the 4-channel model DHO1104 as an example to illustrate the basic operation methods of DHO1000 series.

| Model | Max. Analog Bandwidth | Analog Channels |
|---------|-----------------------|-----------------|
| DHO1072 | 70 MHz | 2 |
| DHO1074 | 70 MHz | 4 |
| DHO1102 | 100 MHz | 2 |
| DHO1104 | 100 MHz | 4 |
| DHO1202 | 200 MHz | 2 |
| DHO1204 | 200 MHz | 4 |

2 SCPI Command Overview

SCPI (Standard Commands for Programmable Instruments) is a standardized instrument programming language that is built upon the existing standard IEEE 488.1 and IEEE 488.2 and conforms to various standards, such as the floating point operation rule in IEEE 754 standard, ISO 646 7-bit coded character set for information interchange (equivalent to ASCII programming). The SCPI commands provide a hierarchical tree structure, and consist of multiple subsystems. Each command subsystem consists of one root keyword and one or more sub-keywords.

Syntax

The command line usually starts with a colon; the keywords are separated by colons, and following the keywords are the parameter settings available. The command ending with a quotation mark indicates querying a certain function and returns the query results. The keywords of the command and the first parameter are separated by a space.

For example,

```
:ACQUIRE:TYPE <type>
```

```
:ACQUIRE:TYPE?
```

ACQUIRE is the root keyword of the command, **TYPE** is the second-level keyword. The command line starts with a colon ":", and different levels of keywords are also separated by colons. <type> indicates a settable parameter. The command ending with a quotation mark "?" indicates querying a certain function. The command keywords **:ACQUIRE:TYPE** and the parameter <type> are separated by a space.

In some commands with parameters, "," is often used to separate multiple parameters. For example,

```
:SYSTEM:DATE <year>,<month>,<day>
```

Symbol Description

The following symbols are not sent with the commands.

1. Braces { }

The contents in the braces can contain one or multiple parameters. These parameters can be omitted or used for several times. Parameters are usually separated by the vertical bar "|". When using the command, you must select one of the parameters.

2. Vertical Bar |

The vertical bar is used to separate multiple parameters. When using the command, you must select one of the parameters.

3. Square Brackets []

The contents in the square brackets can be omitted.

4. Angle Brackets < >

The parameter enclosed in the angle brackets must be replaced by an effective value.

Parameter Type

1. Bool

The parameter can be set to ON, OFF, 1, or 0. For example,

```
:SYSTem:BEEPer <bool>
```

```
:SYSTem:BEEPer?
```

Wherein, <bool> can be set to {{1|ON}}{0|OFF}}. The query returns 1 or 0.

2. Discrete

The parameter can be any of the values listed. For example,

```
:SYSTem:PSTatus <sat>
```

```
:SYSTem:PSTatus?
```

Wherein,

- <sat> can be set to DEFault|OPEN.
- The query returns an abbreviated form: DEF or OPEN.

3. Integer

Unless otherwise specified, the parameter can be any integer (NR1 format) within the effective value range.



CAUTION

Do not set the parameter to a decimal, otherwise, errors will occur.

For example,

```
:DISPlay:GBrightness <brightness>
```

```
:DISPlay:GBrightness?
```

Wherein, <brightness> can be set to an integer ranging from 1 to 100. The query returns an integer ranging from 1 to 100.

4. Real

The parameter can be any real number within the effective value range, and this command accepts parameter input in decimal (NR2 format) and scientific notation (NR3 format). For example,

```
:TRIGger:TIMEout:TIME <time>
```

:TRIGger:TIMEout:TIME?

Wherein, *<time>* can be set to any real number ranging from 1.6E-8 (that is, 16 ns) to 1E+1 (that is, 10 s). The query returns a real number in scientific notation.

5. ASCII String

The parameter can be the combinations of ASCII characters. For example,

:LAN:GATeway <string>

Wherein, *<string>* can be set to

```
192.168.1.1
```

Command Abbreviation

All the commands are case-insensitive. They can all be in upper case or in lower case. If abbreviation is used, you must input all the capital letters in the command. For example,

:DISPlay:GBRightness?

can be abbreviated to

:DISP:GBR?

3 Command System

This chapter introduces the syntax, functions, parameters, and usage of each DHO4000 and DHO1000 command.



CAUTION

1. Unless otherwise specified, the descriptions in this manual take DHO4804 as an example to illustrate the commands of the DHO4000 series.
2. Unless otherwise specified, the descriptions in this manual take DHO1204 as an example to illustrate the commands of the DHO1000 series.
3. For the parameter setting command (time, frequency, amplitude, etc.), the digital oscilloscope can only recognize the numbers, unable to recognize the unit sent together with them. The unit of the parameter is a default one. For the default units of various parameters, refer to the descriptions for the specified command.

3.1 :CLEAr

Syntax

:CLEAr

Description

Clears all the waveforms on the screen.

This command functions the same as the front-panel key  .

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A



3.2 :RUN

Syntax

:RUN

Description

The :RUN command starts running the oscilloscope

This command functions the same as clicking or tapping the icon  in the quick operation bar at the upper-right part of the screen; or press  on the front panel.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A



3.3 :STOP

Syntax

:STOP

Description

The :STOP command stops running the oscilloscope.

This command functions the same as clicking or tapping the icon  in the quick operation bar at the upper-right part of the screen; or press  on the front panel.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A


3.4 :SINGle

Syntax

:SINGle

Description

Performs a single trigger. Sets the trigger mode of the oscilloscope to "Single". This command functions the same as sending the *:TRIGger:SWEEp SINGle* command.

This command functions the same as the front-panel key  .

Parameter

N/A

Remarks

- In the single trigger mode, the oscilloscope performs a single trigger when the trigger conditions are met and then it stops.
- When the waveform recording function is enabled or the recorded waveforms are played back, this command is invalid.
- For the single trigger, you can use the *:TFORce* command to generate one trigger by force.

Return Format

N/A

Example

N/A


3.5 :TFORce

Syntax

:TFORce

Description

Generates a trigger signal forcefully. This command is only applicable to the normal and single trigger modes. Refer to the *:TRIGger:SWEEp* command.

This command functions the same as the  key in the trigger control area of the front panel.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A


3.6 :AUToset Commands

The **:AUToset** commands are used to perform waveform auto setting operations.

3.6.1 :AUToset

Syntax**:AUToset****Description**

Enables the waveform auto setting function. The oscilloscope will automatically adjust the vertical scale, horizontal time base, and trigger mode according to the input signal to realize optimal waveform display.

This command functions the same as the front-panel key .

Parameter

N/A

Remarks

- When the AUTO function is disabled, this command is invalid. For details, refer to *:SYSTem:AUToscale*.
- When the pass/fail test is enabled, the AUTO function runs normally, but the pass/fail test function is forced to be disabled.
- When the waveform recording function is enabled, the AUTO function runs normally, but the recording or playing function is forced to be disabled.

Return Format

N/A

Example

N/A

3.6.2 :AUToset:PEAK**Syntax**`:AUToset:PEAK <bool>``:AUToset:PEAK?`**Description**

Sets or queries whether the peak-peak priority setting is enabled.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 1 ON |

Remarks

This function is intended for the shifted signal. If there is a large deviation, you can view the signal waveform in priority when you enable the function.

Return Format

The query returns 0 or 1.

Example

```
:AUToset:PEAK OFF /*Disables the peak-peak priority setting.*/
:AUToset:PEAK? /*The query returns 0.*/
```

3.6.3 :AUToset:OPENch**Syntax**`:AUToset:OPENch <bool>``:AUToset:OPENch?`**Description**

Sets or queries whether to only test the enabled channel when performing the AUTO operation.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

- If you select "0|OFF", the system will test the four analog channels (CH1-CH4) in sequence when performing the AUTO operation. If no signal is found on the channel, then the channel is disabled. If a signal is found on the channel, adjust the channel to an optimal scale to show the signal.
- If you select "1|ON", the system will only test the enabled channels when performing the AUTO operation.

Return Format

The query returns 0 or 1.

Example

```
:AUToset:OPENch ON /*Sets to test only the enabled channel(s) for
AUTO operation.*/
:AUToset:OPENch? /*The query returns 1.*/
```

3.6.4 :AUToset:OVERlap

Syntax

```
:AUToset:OVERlap <bool>
```

```
:AUToset:OVERlap?
```

Description

Sets or queries the waveform display mode.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 1 ON |

Remarks

- **1|ON:** enables the waveform overlay display. Waveforms of different channels will be displayed in the same position of the screen.

- **0|OFF:** disables the waveform overlay display. Waveforms of different channels will be displayed on the screen from top to bottom in sequence.

Return Format

The query returns 0 or 1.

Example

```
:AUToset:OVERlap OFF /*Disables the waveform overlay display.*/
:AUToset:OVERlap? /*The query returns 0.*/
```

3.6.5 :AUToset:KEEPcoup

Syntax

```
:AUToset:KEEPcoup <bool>
```

```
:AUToset:KEEPcoup?
```

Description

Sets or queries whether to enable keeping coupling.

Parameter

| Name | Type | Range | Default |
|--------|------|-------------------|---------|
| <bool> | Bool | {{1 ON}}{ 0 OFF}} | 0 OFF |

Remarks

- **1|ON:** Enables keeping coupling. When enabled, the system performs auto setting operation. The settings for the channel coupling remain unchanged.
- **0|OFF:** Disables keeping coupling. When disabled, the channel is, by default, DC coupled.

Return Format

The query returns 0 or 1.

Example

```
:AUToset:KEEPcoup ON /*Sets to enable keeping coupling.*/
:AUToset:KEEPcoup? /*The query returns 1.*/
```

3.6.6 :AUToset:LOCK

Syntax

```
:AUToset:LOCK <bool>
```

`:AUToset:LOCK?`

Description

Sets or queries the on/off status of the AUTO function.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

- **1|ON:** locks the AUTO key; disables the AUTO function.
- **0|OFF:** unlocks the AUTO key; enables the AUTO function.

You can also send the `:AUToset:ENable` command to enable or disable the AUTO function.

Return Format

The query returns 0 or 1.

Example

```
:AUToset:LOCK ON /*Locks the AUTO key; disables the AUTO
function.*/
:AUToset:LOCK? /*The query returns 1.*/
```

3.6.7 :AUToset:ENable

Syntax

`:AUToset:ENable <bool>`

`:AUToset:ENable?`



Description

Sets or queries whether to enable the AUTO function.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 1 ON |

Remarks

- **1|ON:** enables the AUTO function. It functions the same as enabling the front-panel key .
- **0|OFF:** disables the AUTO function. It functions the same as disabling the front-panel key .

You can also send the `:AUToset:LOCK` command to enable or disable the AUTO function.

Return Format

The query returns 0 or 1.

Example

```
:AUToset:ENable OFF /*Disables the AUTO function.*/
:AUToset:ENable? /*The query returns 0.*/
```

3.7 :ACQUIRE Commands

The **:ACQUIRE** commands are used to set the memory depth of the oscilloscope, the acquisition mode, the average times, as well as query the current sample rate.

3.7.1 :ACQUIRE:AVERAGES

Syntax

```
:ACQUIRE:AVERAGES <count>
```

```
:ACQUIRE:AVERAGES?
```

Description

Sets or queries the number of averages in the average acquisition mode.

Parameter

| Name | Type | Range | Default |
|---------|---------|--|---------|
| <count> | Integer | 2 ⁿ (n is an integer, and its range is from 1 to 16). | 2 |

Remarks

- You can send the `:ACQUIRE:TYPE` command to set the acquisition mode.

- In the average acquisition mode, greater number of averages can lower the noise and increase the vertical resolution; but will also slow the response of the displayed waveform to the waveform changes.
- The number of averages must be in the Nth power of 2. When the value is not in the Nth power of 2, a value that is smaller than the one you input and the closest to the N power-of-2 increments will be input automatically. For example, if you input 9 with the numeric keypad, the average count will be input 8 automatically.

Return Format

The query returns an integer ranging from 2 to 65536.

Example

```
:ACquire:AVERages 128 /*Sets the average times to 128.*/  
:ACquire:AVERages? /*The query returns 128.*//
```

3.7.2 :ACquire:BITS

Syntax

```
:ACquire:BITS <bit>
```

```
:ACquire:BITS?
```

Description

Sets or queries the resolution bits supported by the oscilloscope in high-resolution mode.

Parameter

| Name | Type | Range | Default |
|-------|----------|---------|---------|
| <bit> | Discrete | {14 16} | 14 |

Remarks

This oscilloscope supports 14 bits and 16 bits in high-resolution mode; 12 bits in other modes.

Return Format

The query returns the resolution bits in integer. Its unit is bit.

Example

```
:ACquire: BITS 16 /*Sets the resolution in high-resolution mode  
to 16 bits.*/  
:ACquire: BITS? /*The query returns 16.*//
```

3.7.3 :ACQUIRE:MDEPTH

Syntax

```
:ACQUIRE:MDEPTH <mdep>
```

```
:ACQUIRE:MDEPTH?
```

Description

Sets or queries the memory depth of the oscilloscope (i.g. the number of waveform points that can be stored through the sampling in a single trigger). The default unit is pts.

Parameter

| Name | Type | Range | Default |
|--------|----------|--|---------|
| <mdep> | Discrete | {AUTO 1k 10k 100k 1M 10M 25M 50M 100M 125M 200M 250M 500M 1000 10000 100000 1000000 10000000 25000000 50000000 100000000 125000000 200000000 250000000 500000000 1e3 1e4 1e5 1e6 1e7 2.5e7 5e7 1e8 125e6 1.25e8 2e8 250e6 2.5e8 5e8} | 10k |

Remarks

When you select the "Auto" mode, the oscilloscope selects the memory depth automatically according to the current sample rate.

- When only one of the four channels is enabled:
 - For DHO4000 series, the available memory depths are AUTO, 1k, 10k, 100k, 1M, 10M, 25M, 50M, 100M, 125M, 200M, 250M, and 500M (option).
 - For DHO1000 series, the available memory depths are AUTO, 1k, 10k, 100k, 1M, 10M, 25M, 50M, and 100M (option).
- When any two of the four channels are enabled:
 - For DHO4000 series, the available memory depths are AUTO, 1k, 10k, 100k, 1M, 10M, 25M, 50M, 100M, 125M, and 250M (option).

- For DHO1000 series, the available memory depths are AUTO, 1k, 10k, 100k, 1M, 10M, 25M, and 50M (option).
- When three channels or four channels are enabled (only applicable to four-channel models):
 - For DHO4000 series, the available memory depths are AUTO, 1k, 10k, 100k, 1M, 10M, 25M, 50M, 100M (option), and 125M (option).
 - For DHO1000 series, the available memory depths are AUTO, 1k, 10k, 100k, 1M, 10M, and 25M (option).

Modifying the memory depth will affect the sample rate. To query the current sample rate, run the `:ACQUIRE:SRATE?` command.

Return Format

The query returns the memory depth in scientific notation.

Example

```
:ACQUIRE:MDEPTH 1M /*Sets the memory depth to 1M.*/  
:ACQUIRE:MDEPTH? /*The query returns 1.000E+6.*/
```

3.7.4 :ACQUIRE:MEMDEPTH?

Syntax

```
:ACQUIRE:MEMDEPTH?
```

Description

Queries the memory depth of the oscilloscope.

Parameter

N/A

Remarks

This command exists for backwards compatibility. Use the command `:ACQUIRE:MDEPTH`.

Return Format

The query returns the memory depth in strings.

Example

```
:ACQUIRE:MDEPTH? /*The query returns 1.000E+6.*/
```

3.7.5 :ACquire:TYPE

Syntax

:ACquire:TYPE <type>

:ACquire:TYPE?

Description

Sets or queries the acquisition mode of the oscilloscope.

Parameter

| Name | Type | Range | Default |
|--------|----------|--|---------|
| <type> | Discrete | {NORMal PEAK AVERages HRESolution ULTRa} | NORMal |

Remarks

- NORMal:** In this mode, the oscilloscope samples the signal at a specified fixed time interval to rebuild the waveform. For most of the waveforms, using this mode can produce the optimal display effects.
- AVERages:** In this mode, the oscilloscope averages the waveforms from multiple samples to reduce the random noise of the input signal and improve the vertical resolution. Greater number of averages can lower the noises and increase the vertical resolution, but will also slow the response of the displayed waveform to the waveform changes.
- PEAK:** indicates the peak detection. In this mode, the oscilloscope samples the maximum and minimum value of the signal at the fixed sampling interval to acquire the signal envelope or the narrow pulses that might be lost. In this mode, signal aliasing can be prevented, but the noise displayed would be larger.
- HRESolution:** indicates high resolution. The oscilloscope will average the adjacent sample points of the sample waveform to lower the random noises of the input signals and display much more smoother waveforms. If the sample rate of the digital converter is greater than the storage rate of the acquisition memory, this mode is often adopted.

- **ULTRa (UltraAcquire):** divides the oscilloscope's memory into segments and fills a memory segment for each individual trigger event. In this mode, it provides a high waveform capture rate and minimizes the dead time between trigger events.

Return Format

The query returns NORM, PEAK, AVER, ULTR or HRES.

Example

```
:ACquire:TYPE AVERages /*Sets the acquisition mode to AVERages.*/
:ACquire:TYPE? /*The query returns AVER.*/
```

3.7.6 :ACquire:SRATE?

Syntax

```
:ACquire:SRATE?
```

Description

Queries the current sample rate. The default unit is Sa/s.

Parameter

N/A

Remarks

- Sample rate indicates the frequency of the signal sampling, i.g. the number of waveform points sampled per second.
- The sample rate and memory depth will change accordingly in accordance with the horizontal time base. To set the memory depth, send the `:ACquire:MDEPTH` command. To set the horizontal time base, send the `:TIMEbase[:MAIN]:SCALE` command.

Return Format

The query returns the sample rate in scientific notation.

Example

```
:ACquire:SRATE? /*The query returns 1.00000E+6.*/
```

3.7.7 :ACquire:ULTRa:MODE

Syntax

```
:ACquire:ULTRa:MODE <mode>
```

```
:ACquire:ULTRa:MODE?
```


Description

Sets or queries the display mode of Ultra Acquire.

Parameter

| Name | Type | Range | Default |
|--------|----------|---|---------|
| <mode> | Discrete | {ADJacent OVERlay WATERfall PERSpective MOSaic} | - |

Remarks

- **ADJacent:** indicates the adjacent display. Waveform segments are shown in adjacent display, with each segment shown next to the previous segment. In this mode, a maximum of 100 frames of waveforms can be displayed on the screen at a time.
- **OVERlay:** indicates the overlay display. All the captured waveform segments are overwritten to display as one single segment of the waveform. In this mode, a maximum of 100 frames of waveforms can be displayed on the screen at a time.
- **WATERfall:** indicates the waterfall display. It displays the captured waveform segments in a cascaded display order. In this mode, a maximum of 100 frames of waveforms can be displayed on the screen at a time.
- **PERSpective:** indicates the perspective display. The waveform segments are displayed in the ladder-like form, with each segment being arranged above another with a certain perspective (angle), moving up like a rising slope. In this mode, a maximum of 100 frames of waveforms can be displayed on the screen at a time.
- **MOSaic:** indicates the mosaic display. The whole waveform view is divided into several blocks, and each waveform segment is displayed in each block in sequence. In this mode, a maximum of 80 frames of waveforms can be displayed on the screen at a time.

Return Format

The query returns ADJ, OVER, WAT, PERS, or MOS.

Example

```
:ACquire:ULTRa:MODE ADJacent /*Sets the Ultra Acquire display
mode to ADJacent.*/
:ACquire:ULTRa:MODE? /*The query returns ADJ.*/
```

3.7.8 :ACquire:ULTRa:TIMEout**Syntax**

```
:ACquire:ULTRa:TIMEout < tmo>
```

```
:ACquire:ULTRa:TIMEout?
```

Description

Sets or queries the timeout value of Ultra Acquire.

Parameter

| Name | Type | Range | Default |
|-------|------|-------------|---------|
| <tmo> | Real | 1 us to 1 s | 1.00 ms |

Remarks

N/A

Return Format

The query returns a real number in scientific notation.

Example

```
:ACquire:ULTRa:TIMEout 0.1 /*Sets the Ultra Acquire timeout
value to 0.1 s.*/
:ACquire:ULTRa:TIMEout? /*The query returns 1E-1.*/
```

3.7.9 :ACquire:ULTRa:MAXFrame**Syntax**

```
:ACquire:ULTRa:MAXFrame < frame>
```

```
:ACquire:ULTRa:MAXFrame?
```

Description

Sets or queries the maximum number of frames that can be set for Ultra Acquire sampling mode.

Parameter

| Name | Type | Range | Default |
|---------|---------|-----------------------------------|---------|
| <frame> | Integer | 1 to the maximum number of frames | - |

Remarks

N/A

Return Format

The query returns the maximum number of frames in integer.

Example

```
:ACquire:ULTRa:MAXFrame 100 /*Sets the maximum number of frames
to 100.*/
:ACquire:ULTRa:MAXFrame? /*The query returns 100.*/
```

3.8 :BUS<n> Commands

The **:BUS<n>** commands are used to execute the decoding-related settings and operations.

3.8.1 :BUS<n>:MODE

Syntax

```
:BUS<n>:MODE <mode>
```

```
:BUS<n>:MODE?
```

Description

Sets or queries the decoding type of the specified decoding bus.

Parameter

| Name | Type | Range | Default |
|--------|----------|--|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <mode> | Discrete | {PARAllel RS232 SPI IIC IIS LIN CAN FLEXray M1553} | PARAllel |

Remarks

Parallel, RS232, SPI, I2C, and CAN decodings are standard configurations for the DHO4000 series oscilloscope. Other decodings are optional configurations. Only when the specified option is installed, can this command be available.

The DHO1000 series does not support FlexRay, I2S, and M1553 decoding types. Except them, other decoding types are standard configurations.

Return Format

The query returns PAR, RS232, SPI, IIC, IIS, LIN, CAN, FLEX, or M1553.

Example

```
:BUS1:MODE SPI /*Sets the type of Bus 1 to SPI.*/
:BUS1:MODE? /*The query returns SPI.*/
```

3.8.2 :BUS<n>:DISPlay

Syntax

```
:BUS<n>:DISPlay <bool>
```

```
:BUS<n>:DISPlay?
```

Description

Enables or disables the specified decoding bus; or queries the on/off display status of the specified decoding bus.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:BUS1:DISPlay ON /*Enables the decoding bus.*/
:BUS1:DISPlay? /*The query returns 1.*/
```

3.8.3 :BUS<n>:FORMat

Syntax

```
:BUS<n>:FORMat <format>
```

```
:BUS<n>:FORMat?
```

Description

Sets or queries the format of decoding data on the specified decoding bus.

Parameter

| Name | Type | Range | Default |
|----------|----------|---------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <format> | Discrete | {HEX ASCii DEC BIN} | HEX |

Remarks

- **Hex:** indicates Hexadecimal;
- **ASCii:** indicates ASCII;
- **DEC:** indicates Decimal;
- **BIN:** indicates Binary.

Return Format

The query returns HEX, ASC, DEC, or BIN.

Example

```
:BUS1:FORMat HEX /*Sets the display format of the bus to HEX.*/
:BUS1:FORMat? /*The query returns HEX.*/
```

3.8.4 :BUS<n>:EVENT**Syntax**

```
:BUS<n>:EVENT <bool>
```

```
:BUS<n>:EVENT?
```

Description

Enables or disables the event table of the specified decoding bus; or queries the on/off status of the specified decoding bus event table.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

Before using the command, enable the specified decoding bus.

Return Format

The query returns 1 or 0.

Example

```
:BUS1:EVENT ON /*Enables the event table of the specified
decoding bus.*/
:BUS1:EVENT? /*The query returns 1.*/
```

3.8.5 :BUS<n>:LABel**Syntax**

```
:BUS<n>:LABel <bool>
```

```
:BUS<n>:LABel?
```

Description

Enables or disables the label of the specified decoding bus; or queries the on/off display status of the label of the specified decoding bus.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bool> | Bool | {{1 ON}}{0 OFF}} | 1 ON |

Remarks

Before using the command, enable the specified decoding bus.

Return Format

The query returns 1 or 0.

Example

```
:BUS1:LABel ON /*Enables the label of the specified decoding bus.*/
:BUS1:LABel? /*The query returns 1.*/
```

3.8.6 :BUS<n>:DATA?**Syntax**

```
:BUS<n>:DATA?
```

Description

Reads the data from the decoding event table.

Parameter

| Name | Type | Range | Default |
|------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |

Remarks

N/A

Return Format

The query returns the data in the decoding event table with the following formats.

```
#9000000086PARALLEL
Time,Data,
-2.47us,0,
-2.444us,1,
-1.448us,0,
-446ns,1,
551.6ns,0,
1.554us,1,
```

Wherein, "#9000000086" is the TMC data block header, which is followed by the data in the event table. The 9-digit data following #9 in the data block header indicates the number of bytes of the effective data. "PARALLEL" indicates the decoding type. The available decoding type can also be RS232, I2C, SPI, LIN, and etc. The data are separated by a comma, and will automatically switch to the next line according to the data information in the decoding list. The data value is related to the numeral system that you have set.

**CAUTION**

You can save all the data (except TMC data block header and decoding type, e.g. #9000000086PARALLEL) as the "*.csv" file and view the data in the form of a list.

Example

N/A

3.8.7 :BUS<n>:EEXPort**Syntax**

```
:BUS<n>:EEXPort <path>
```

Description

Exports the decoding information from the specified decoding bus event table in CSV form.

Parameter

| Name | Type | Range | Default |
|--------|--------------|-------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <path> | ASCII String | Refer to <i>Remarks</i> | - |

Remarks

- <path> includes the file storage location and the filename with a suffix. If the specified storage location already contains a file with the same filename, the original file will be overwritten.
- When the operating status of the instrument is STOP (set it by sending the *:STOP* command), you can export the time and corresponding decoding data from the current event table.
- This command is valid when the display of the event table is enabled. You can enable the display of the event table by sending the *:BUS<n>:EVENTt* command.
- The stored "*.csv" file can be opened and edited in Excel.

Return Format

N/A

Example

```
:BUS1:EEEXPort C:/123.csv /*Exports the decoding information from
the bus event table and saves it to the local Disk C, with the
filename 123.csv.*/*
```

3.8.8 :BUS<n>:POSITION**Syntax**

```
:BUS<n>:POSITION <pos>
```

```
:BUS<n>:POSITION?
```

Description

Sets or queries the vertical position of the bus on the screen.

Parameter

| Name | Type | Range | Default |
|-------|----------|-------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <pos> | Integer | -250 to 250 | 0 |

Remarks

N/A

Return Format

The query returns an integer ranging from -250 to 250.

Example

```
:BUS1:POSition 200 /*Sets the vertical position of the bus to
200.* /
:BUS1:POSition? /*The query returns 200.* /
```

3.8.9 :BUS<n>:THReshold**Syntax**

```
:BUS<n>:THReshold <value>,<type>
```

```
:BUS<n>:THReshold? <type>
```

Description

Sets or queries the threshold of the specified decoding source on the specified decoding bus.

Parameter

| Name | Type | Range | Default |
|---------|----------|--|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <value> | Real | (-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet) | 0 |
| <type> | Discrete | {PAL TX RX SCL SDA CS CLK MISO MOSI LIN CAN CANSUB1 FLEX 1553 PALCLK DATA WS I2SCLK CH1 CH2 CH3 CH4} | - |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

- **PAL:** indicates the bus source of Parallel decoding.
- **PALCLK:** indicates the clock source of Parallel decoding. Only when the source is enabled, can you set the threshold.
- **TX:** indicates the TX channel source of RS232 decoding.
- **RX:** indicates the RX channel source of RS232 decoding. Only when the RX source is enabled, can you set the threshold.
- **SCL:** indicates the clock source of I2C decoding.
- **SDA:** indicates the clock source of I2C decoding.
- **DATA:** indicates the data source of I2S decoding.
- **WS:** indicates the audio channel of I2S decoding.
- **I2SClk:** indicates the clock source of I2S decoding.
- **CS:** indicates the source of the CS line of SPI decoding.
- **CLK:** indicates the clock source of SPI decoding.
- **MISO:** indicates the MISO data source of SPI decoding.
- **MOSI:** indicates the MOSI data source of SPI decoding.
- **LIN:** indicates the bus source of LIN decoding.
- **CAN|CANSub1:** indicates the source of CAN decoding.
- **1553:** indicates the channel source of M1553 decoding.
- **FLEX:** indicates the source of FlexRay decoding.
- **CH1|CH2|CH3|CH4:** CH1-CH4, available for all the decoding types.

The DHO1000 series does not support FLEX, 1553, and I2S decodings.

Return Format

The query returns the threshold of the specified decoding source in scientific notation.

Example

```
:BUS1:THReshold 2.4,PAL /*Sets the threshold of the Parallel  
decoding source to 2.4 V.*/  
:BUS1:THReshold? PAL /*The query returns 2.400000E0.*/
```

3.8.10 :BUS<n>:PARAllel

The :BUS<n>:PARAllel commands are used to set relevant parameters for Parallel decoding.

3.8.10.1 :BUS<n>:PARAllel:BUS

Syntax

```
:BUS<n>:PARAllel:BUS <source>
```

```
:BUS<n>:PARAllel:BUS?
```

Description

Sets or queries the current source of the Parallel decoding bus.

Parameter

| Name | Type | Range | Default |
|----------|----------|--|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 USER} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, or USER.

Example

```
:BUS1:PARAllel:BUS CHANnel1 /*Sets the current source of the
Parallel decoding bus to CHANnel1.*/
:BUS1:PARAllel:BUS? /*The query returns CHAN1.*/
```

3.8.10.2 :BUS<n>:PARAllel:CLK

Syntax

```
:BUS<n>:PARAllel:CLK <source>
```

```
:BUS<n>:PARAllel:CLK?
```

Description

Sets or queries the clock source of the Parallel decoding.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 OFF} | OFF |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, or OFF.

Example

```
:BUS1:PARallel:CLK CHANnel2 /*Sets the clock source of the
Parallel decoding to CHANnel2.*/
:BUS1:PARallel:CLK? /*The query returns CHAN2.*/
```

3.8.10.3 :BUS<n>:PARallel:SLOPe**Syntax**

```
:BUS<n>:PARallel:SLOPe <slope>
```

```
:BUS<n>:PARallel:SLOPe?
```

Description

Sets or queries the edge type of the clock channel when being sampled by Parallel decoding on the data channel.

Parameter

| Name | Type | Range | Default |
|---------|----------|--------------------------|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <slope> | Discrete | {POSitive NEGative BOTH} | POSitive |

Remarks

If no clock channel is selected, the instrument will sample when the channel data hopping occurs during the decoding.

Return Format

The query returns POS, NEG, or BOTH.

Example

```
:BUS1:PARAllel:SLOPe BOTH /*Sets the Parallel decoding to
sample on any edge of the clock channel.*/
:BUS1:PARAllel:SLOPe? /*The query returns BOTH.*/
```

3.8.10.4 :BUS<n>:PARAllel:WIDTh**Syntax**

```
:BUS<n>:PARAllel:WIDTh <wid>
```

```
:BUS<n>:PARAllel:WIDTh?
```

Description

Sets or queries the data width of the Parallel decoding, i.g. the number of bits per frame.

Parameter

| Name | Type | Range | Default |
|-------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <wid> | Integer | 1 to 4 | 1 |

Remarks

- Send this command to set the data width of the bus first, then send the `:BUS<n>:PARAllel:BITX` and `:BUS<n>:PARAllel:SOURce` command to select the bit and set the channel source for the bit.
- Only when the bus source is set to User, can this command be valid.

Return Format

The query returns an integer ranging from 1 to 4.

Example

```
:BUS1:PARAllel:WIDTh 4 /*Sets the data width of Parallel
decoding to 4.*/
:BUS1:PARAllel:WIDTh? /*The query returns 4.*/
```

3.8.10.5 :BUS<n>:PARAllel:BITX**Syntax**

```
:BUS<n>:PARAllel:BITX <bit>
```

```
:BUS<n>:PARAllel:BITX?
```

Description

Sets or queries the data bit that the parallel bus requires to set for the channel source.

Parameter

| Name | Type | Range | Default |
|-------|----------|-----------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bit> | Integer | 0 to (data width - 1) | 0 |

Remarks

- The data width is set by the `:BUS<n>:PARAllel:WIDTh` command.
- After selecting the desired bit, send the `:BUS<n>:PARAllel:SOURce` command to set the channel source for the bit.

Return Format

The query returns the current data bits in integer. Its unit is Hz.

Example

```
:BUS1:PARAllel:BITX 2 /*Sets the current bit to 2.*/
:BUS1:PARAllel:BITX? /*The query returns 2.*/
```

3.8.10.6 :BUS<n>:PARAllel:SOURce**Syntax**

```
:BUS<n>:PARAllel:SOURce <SRC>
```

```
:BUS<n>:PARAllel:SOURce?
```

Description

Sets or queries the channel source of the currently selected data bit.

Parameter

| Name | Type | Range | Default |
|-------|----------|---------------------------------------|-----------------------------|
| <n> | Discrete | {1 2 3 4} | - |
| <src> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | Related to the selected bit |

Remarks

Before sending this command, send the `:BUS<n>:PARAllel:BITX` command to select the desired data bit.

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:BUS1:PARAllel:SOURce CHANnel2 /*Sets the channel source of the
current bit to CHANnel2.*/
:BUS1:PARAllel:SOURce? /*The query returns CHAN2.*/
```

3.8.10.7 :BUS<n>:PARAllel:ENDian**Syntax**

```
:BUS <n>:PARAllel:ENDian <endian>
```

```
:BUS <n>:PARAllel:ENDian?
```

Description

Sets or queries the endian of Parallel decoding on the specified decoding bus.

Parameter

| Name | Type | Range | Default |
|----------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <endian> | Discrete | {MSB LSB} | MSB |

Remarks

- **MSB:** indicates Most Significant Bit transmission sequence, i.g. the highest bit of the data is transmitted first.
- **LSB:** indicates Least Significant Bit transmission sequence, i.g. the lowest bit of the data is transmitted first.

Return Format

The query returns MSB or LSB.

Example

```
:BUS1:PARAllel:ENDian LSB /*Sets the endian of Parallel decoding
to LSB.*/
:BUS1:PARAllel:ENDian? /*The query returns LSB.*/
```

3.8.10.8 :BUS<n>:PARallel:POLarity

Syntax

```
:BUS<n>:PARallel:POLarity <pol>
```

```
:BUS<n>:PARallel:POLarity?
```

Description

Sets or queries the data polarity of Parallel decoding.

Parameter

| Name | Type | Range | Default |
|-------|----------|---------------------|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <pol> | Discrete | {NEGative POSitive} | POSitive |

Remarks

- **NEGative:** indicates negative polarity.
- **POSitive:** indicates positive polarity.

Return Format

The query returns NEG or POS.

Example

```
:BUS1:PARallel:POLarity NEGative /*Sets the data polarity of
Parallel decoding to Negative.*/
:BUS1:PARallel:POLarity? /*The query returns NEG.*/
```

3.8.11 :BUS<n>:RS232

The :BUS<n>:RS232 commands are used to set relevant parameters for RS232 decoding.

3.8.11.1 :BUS<n>:RS232:TX

Syntax

```
:BUS<n>:RS232:TX <source>
```

```
:BUS<n>:RS232:TX?
```

Description

Sets or queries the Tx source of RS232 decoding on the specific bus.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 OFF} | CHANnel1 |

Remarks

The Tx and Rx sources cannot be set to OFF at the same time. The Rx source can be set by using the `:BUS<n>:RS232:RX` command.

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, or OFF.

Example

```
:BUS1:RS232:TX CHANnel2 /*Sets the Tx source of RS232 decoding
to CHANnel2.*/
:BUS1:RS232:TX? /*The query returns CHAN2.*/
```

3.8.11.2 :BUS<n>:RS232:RX**Syntax**

```
:BUS<n>:RS232:RX <source>
```

```
:BUS<n>:RS232:RX?
```

Description

Sets or queries the Rx source of RS232 decoding on the specific bus.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 OFF} | OFF |

Remarks

The Tx and Rx sources cannot be set to OFF at the same time. The Tx source can be set by using the `:BUS<n>:RS232:TX` command.

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, or OFF.

Example

```
:BUS1:RS232:RX CHANnel2 /*Sets the Rx source of RS232 decoding
to CHANnel2.*/
:BUS1:RS232:RX? /*The query returns CHAN2.*/
```

3.8.11.3 :BUS<n>:RS232:POLarity**Syntax**

```
:BUS<n>:RS232:POLarity <pol>
```

```
:BUS<n>:RS232:POLarity?
```

Description

Sets or queries the polarity of RS232 decoding on the specified bus.

Parameter

| Name | Type | Range | Default |
|-------|----------|---------------------|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <pol> | Discrete | {POSitive NEGative} | NEGative |

Remarks

- **POSitive:** indicates positive polarity. High level is logic "1" and low level is logic "0".
- **NEGative:** indicates negative polarity. High level is logic "0" and low level is logic "1".

Return Format

The query returns POS or NEG.

Example

```
:BUS1:RS232:POLarity POSitive /*Sets the polarity of RS232
decoding to POSitive.*/
:BUS1:RS232:POLarity? /*The query returns POS.*/
```

3.8.11.4 :BUS<n>:RS232:PARity**Syntax**

```
:BUS<n>:RS232:PARity <parity>
```

```
:BUS<n>:RS232:PARity?
```

Description

Sets or queries the odd-even check mode of data transmission in RS232 decoding.

Parameter

| Name | Type | Range | Default |
|----------|----------|-----------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <parity> | Discrete | {NONE ODD EVEN} | NONE |

Remarks

- **None:** indicates that there is no parity bit in data transmission.
- **ODD:** indicates the odd parity bit. The total count of occurrences of 1 in the data bit and check bit is an odd number. For example, if 0x55 (01010101) is transmitted, 1 shall be added to the check bit.
- **Even:** indicates the even parity bit. The total count of occurrences of 1 in the data bit and check bit is an even number. For example, if 0x55 (01010101) is transmitted, 0 shall be added to the check bit.

Return Format

The query returns NONE, ODD, or EVEN.

Example

```
:BUS1:RS232:PARity ODD /*Sets the odd-even check mode of
data transmission in RS232 decoding to ODD.*/
:BUS1:RS232:PARity? /*The query returns ODD.*/
```

3.8.11.5 :BUS<n>:RS232:ENDian**Syntax**

```
:BUS<n>:RS232:ENDian <endian>
```

```
:BUS<n>:RS232:ENDian?
```

Description

Sets or queries the endian of data transmission in RS232 decoding.

Parameter

| Name | Type | Range | Default |
|----------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <endian> | Discrete | {MSB LSB} | LSB |

Remarks

- **LSB:** indicates Least Significant Bit transmission sequence, i.g. the lowest bit of the data is transmitted first.
- **MSB:** indicates Most Significant Bit transmission sequence, i.g. the highest bit of the data is transmitted first.

Return Format

The query returns LSB or MSB.

Example

```
:BUS1:RS232:ENDian MSB /*Sets the transmission order of
RS232 decoding to MSB.*/
:BUS1:RS232:ENDian? /*The query returns MSB.*/
```

3.8.11.6 :BUS<n>:RS232:BAUD**Syntax**

```
:BUS<n>:RS232:BAUD <baud>
```

```
:BUS<n>:RS232:BAUD?
```

Description

Sets or queries the baud rate of data transmission in RS232 decoding. The default unit is bps.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <baud> | Integer | 1 bps to 20 Mbps | 9600 bps |

Remarks

If the baud rate is set to a value with "M", then "A" should be added at the end of the value. For example, if you send 5M, you need to send 5MA.

Return Format

The query returns an integer ranging from 1 to 20M.

Example

```
:BUS1:RS232:BAUD 4800 /*Sets the baud rate of data
transmission in RS232 decoding to 4800 bps.*/
:BUS1:RS232:BAUD? /*The query returns 4800.*/
```

3.8.11.7 :BUS<n>:RS232:DBITs**Syntax**

```
:BUS<n>:RS232:DBITs <bits>
```

```
:BUS<n>:RS232:DBITs?
```

Description

Sets or queries the data width of RS232 decoding.

Parameter

| Name | Type | Range | Default |
|--------|----------|-------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bits> | Discrete | {5 6 7 8 9} | 8 |

Remarks

N/A

Return Format

The query returns 5, 6, 7, 8, or 9.

Example

```
:BUS1:RS232:DBITs 7 /*Sets the data width of RS232 decoding to
7.*/
:BUS1:RS232:DBITs? /*The query returns 7.*/
```

3.8.11.8 :BUS<n>:RS232:SBITs**Syntax**

```
:BUS<n>:RS232:SBITs <stop bits>
```

```
:BUS<n>:RS232:SBITs?
```

Description

Sets or queries the stop bits of each frame of data in RS232 decoding.

Parameter

| Name | Type | Range | Default |
|-------------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <stop bits> | Discrete | {1 1.5 2} | 1 |

Remarks

N/A

Return Format

The query returns 1, 1.5, or 2.

Example

```
:BUS1:RS232:SBITS 2 /*Sets the stop bits of RS232 decoding to 2.*/  
:BUS1:RS232:SBITS? /*The query returns 2.*/*
```

3.8.12 :BUS<n>:IIC

The :BUS<n>:IIC commands are used to set relevant parameters for I2C decoding.

3.8.12.1 :BUS<n>:IIC:SCLK:SOURce**Syntax**

```
:BUS<n>:IIC:SCLK:SOURce <source>
```

```
:BUS<n>:IIC:SCLK:SOURce?
```

Description

Sets or queries the clock source of I2C decoding.

Parameter

| Name | Type | Range | Default |
|----------|----------|---------------------------------------|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:BUS1:IIC:SCLK:SOURce CHANnel2 /*Sets the clock source of I2C
decoding to CHANnel2.*/
:BUS1:IIC:SCLK:SOURce? /*The query returns CHAN2.*/
```

3.8.12.2 :BUS<n>:IIC:SDA:SOURce**Syntax**

```
:BUS<n>:IIC:SDA:SOURce <source>
```

```
:BUS<n>:IIC:SDA:SOURce?
```

Description

Sets or queries the data source of the I2C decoding.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:BUS1:IIC:SDA:SOURce CHANnel2 /*Sets the data source of I2C
decoding to CHANnel2.*/
:BUS1:IIC:SDA:SOURce? /*The query returns CHAN2.*/
```

3.8.12.3 :BUS<n>:IIC:EXCHange**Syntax**

```
:BUS<n>:IIC:EXCHange <bool>
```

```
:BUS<n>:IIC:EXCHange?
```

Description

Sets to exchange the clock channel source and the data channel source of I2C decoding on the specified bus; queries whether the clock channel source and the data channel source of I2C decoding on the specified bus have been exchanged.

Parameter

| Name | Type | Range | Default |
|--------|----------|-------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bool> | Bool | {{1 ON}} {0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 0 or 1.

Example

```
:BUS1:IIC:EXCHange ON /*Sets to exchange the clock channel source
and the data channel source.*/
:BUS1:IIC:EXCHange? /*The query returns 1.*/
```

3.8.12.4 :BUS<n>:IIC:ADDRESS**Syntax**

```
:BUS<n>:IIC:ADDRESS <addr>
```

```
:BUS<n>:IIC:ADDRESS?
```

Description

Sets or queries the address mode of I2C decoding.

Parameter

| Name | Type | Range | Default |
|--------|----------|-------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <addr> | Discrete | {NORMal RW} | NORMal |

Remarks

- **NORMal:** indicates that the address width does not include the R/W bit.
- **RW:** indicates that the address width includes the R/W bit.

Return Format

The query returns NORM or RW.

Example

```
:BUS1:IIC:ADDRESS RW /*Sets the address of I2C decoding to
include the R/W bit.*/
:BUS1:IIC:ADDRESS? /*The query returns RW.*/
```

3.8.13 :BUS<n>:SPI

The :BUS<n>:SPI commands are used to set relevant parameters for SPI decoding.

3.8.13.1 :BUS<n>:SPI:SCLK:SOURce**Syntax**

```
:BUS<n>:SPI:SCLK:SOURce <source>
```

```
:BUS<n>:SPI:SCLK:SOURce?
```

Description

Sets or queries the clock source of SPI decoding.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:BUS1:SPI:SCLK:SOURce CHANnel2 /*Sets the clock source of SPI
decoding to CHANnel2.*/
:BUS1:SPI:SCLK:SOURce? /*The query returns CHAN2.*/
```

3.8.13.2 :BUS<n>:SPI:SCLK:SLOPe**Syntax**

```
:BUS<n>:SPI:SCLK:SLOPe <slope>
```

```
:BUS<n>:SPI:SCLK:SLOPe?
```

Description

Sets or queries the clock edge type of the SPI decoding.

Parameter

| Name | Type | Range | Default |
|---------|----------|---------------------|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <slope> | Discrete | {POSitive NEGative} | POSitive |

Remarks

N/A

Return Format

The query returns POS or NEG.

Example

```
:BUS1:SPI:SCLK:SLOPe NEGative /*Sets the clock edge type of SPI
decoding to Negative.*/
:BUS1:SPI:SCLK:SLOPe? /*The query returns NEG.*/
```

3.8.13.3 :BUS<n>:SPI:MISO:SOURce**Syntax**

```
:BUS<n>:SPI:MISO:SOURce <source>
```

```
:BUS<n>:SPI:MISO:SOURce?
```

Description

Sets or queries the MISO data source of SPI decoding on the specified bus.

Parameter

| Name | Type | Range | Default |
|----------|----------|--|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 OFF} | CHANnel2 |

Remarks

The source specified in this command and the *:BUS<n>:SPI:MOSI:SOURce* command cannot be set to OFF at the same time.

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, or OFF.

Example

```
:BUS1:SPI:MISO:SOURce CHANnel2 /*Sets the MISO data source of
SPI decoding to CHANnel2.*/
:BUS1:SPI:MISO:SOURce? /*The query returns CHAN2.*/
```

3.8.13.4 :BUS<n>:SPI:MOSI:SOURce**Syntax**

```
:BUS<n>:SPI:MOSI:SOURce <source>
```

```
:BUS<n>:SPI:MOSI:SOURce?
```

Description

Sets or queries the MOSI data source of SPI decoding on the specified bus.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 OFF} | OFF |

Remarks

The source specified in this command and the `:BUS<n>:SPI:MISO:SOURce` command cannot be set to OFF at the same time.

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, or OFF.

Example

```
:BUS1:SPI:MOSI:SOURce CHANnel2 /*Sets the MOSI data source of
SPI decoding to CHANnel2.*/
:BUS1:SPI:MOSI:SOURce? /*The query returns CHAN2.*/
```

3.8.13.5 :BUS<n>:SPI:POLarity**Syntax**

```
:BUS<n>:SPI:POLarity <polarity>
```

```
:BUS<n>:SPI:POLarity?
```

Description

Sets or queries the polarity of the SPI decoding on the specified decoding bus.

Parameter

| Name | Type | Range | Default |
|------------|----------|------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <polarity> | Discrete | {HIGH LOW} | HIGH |

Remarks

- **HIGH:** indicates positive polarity. The high level is 1, and low level is 0.
- **LOW:** indicates negative polarity. The high level is 0, and low level is 1.

Return Format

The query returns HIGH or LOW.

Example

```
:BUS1:SPI:POLarity HIGH /*Sets the polarity of the SPI
decoding to Positive.*/
:BUS1:SPI:POLarity? /*The query returns HIGH.*/
```

3.8.13.6 :BUS<n>:SPI:MISO:POLarity**Syntax**

```
:BUS<n>:SPI:MISO:POLarity <polarity>
:BUS<n>:SPI:MISO:POLarity?
```

Description

Sets or queries the polarity of MISO data line of SPI decoding.

Parameter

| Name | Type | Range | Default |
|------------|----------|------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <polarity> | Discrete | {HIGH LOW} | HIGH |

Remarks

- **HIGH:** positive polarity. It indicates that high level is 1, and low level is 0.
- **LOW:** negative polarity. It indicates that low level is 1, and high level is 0.

This command exists for backwards compatibility. Use the command `:BUS<n>:SPI:POLarity`.

Return Format

The query returns HIGH or LOW.

Example

```
:BUS1:SPI:MISO:POLarity HIGH /*Sets the polarity of MISO data
line to Positive.*/
:BUS1:SPI:MISO:POLarity? /*The query returns HIGH.*/
```

3.8.13.7 :BUS<n>:SPI:MOSI:POLarity

Syntax

```
:BUS<n>:SPI:MOSI:POLarity <polarity>
```

```
:BUS<n>:SPI:MOSI:POLarity?
```

Description

Sets or queries the polarity of MOSI data line of SPI decoding.

Parameter

| Name | Type | Range | Default |
|------------|----------|------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <polarity> | Discrete | {HIGH LOW} | HIGH |

Remarks

- **HIGH:** positive polarity. It indicates that high level is 1, and low level is 0.
- **LOW:** negative polarity. It indicates that low level is 1, and high level is 0.

This command exists for backwards compatibility. Use the command `:BUS<n>:SPI:POLarity`.

Return Format

The query returns HIGH or LOW.

Example

```
:BUS1:SPI:MOSI:POLarity HIGH /*Sets the polarity of MOSI data
line to Positive.*/
:BUS1:SPI:MOSI:POLarity? /*The query returns HIGH.*/
```

3.8.13.8 :BUS<n>:SPI:DBITs

Syntax

```
:BUS<n>:SPI:DBITs <width>
```

```
:BUS<n>:SPI:DBITs?
```

Description

Sets or queries the data width of SPI decoding.

Parameter

| Name | Type | Range | Default |
|---------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <width> | Integer | 4 to 32 | 8 |

Remarks

N/A

Return Format

The query returns an integer ranging from 4 to 32.

Example

```
:BUS1:SPI:DBITs 10      /*Sets the data width of SPI decoding to
10.*/
:BUS1:SPI:DBITs?       /*The query returns 10.*/
```

3.8.13.9 :BUS<n>:SPI:ENDian

Syntax

```
:BUS<n>:SPI:ENDian <endian>
```

```
:BUS<n>:SPI:ENDian?
```

Description

Sets or queries the endian of data transmission in SPI decoding.

Parameter

| Name | Type | Range | Default |
|----------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <endian> | Discrete | {MSB LSB} | MSB |

Remarks

- **MSB:** indicates Most Significant Bit transmission sequence, i.g. the highest bit of the data is transmitted first.
- **LSB:** indicates Least Significant Bit transmission sequence, i.g. the lowest bit of the data is transmitted first.

Return Format

The query returns MSB or LSB.

Example

```
:BUS1:SPI:ENDian LSB /*Sets the endian of data transmission
in SPI decoding to LSB.*/
:BUS1:SPI:ENDian? /*The query returns LSB.*/
```

3.8.13.10 :BUS<n>:SPI:MODE**Syntax**

```
:BUS<n>:SPI:MODE <mode>
```

```
:BUS<n>:SPI:MODE?
```

Description

Sets or queries the decode mode of SPI decoding.

Parameter

| Name | Type | Range | Default |
|--------|----------|--------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <mode> | Discrete | {CS TIMEout} | TIMEout |

Remarks

- **CS:** indicates chip select. It contains a chip select line (CS). You can perform frame synchronization according to CS.
- **TIMEout:** indicates timed out. You can perform frame synchronization according to the timeout.

At this time, you can send the `:BUS<n>:SPI:TIMEout:TIME` command to set the timeout value.

Return Format

The query returns CS or TIM.

Example

```
:BUS1:SPI:MODE CS      /*Sets the decode mode of SPI decoding to
CS.* /
:BUS1:SPI:MODE?       /*The query returns CS.* /
```

3.8.13.11 :BUS<n>:SPI:TIMEout:TIME**Syntax**

```
:BUS<n>:SPI:TIMEout:TIME <time>
:BUS<n>:SPI:TIMEout:TIME?
```

Description

Sets or queries the timeout value of SPI decoding on the specified bus. The unit is s.

Parameter

| Name | Type | Range | Default |
|--------|----------|--------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <time> | Real | 8 ns to 10 s | 1μs |

Remarks

- The timeout must be greater than the maximum clock pulse width and less than the idle time between frames.
- This setting command is only valid in timeout mode (To set or query the mode, send the *:BUS<n>:SPI:MODE* command).

Return Format

The query returns the timeout value in scientific notation.

Example

```
:BUS1:SPI:TIMEout:TIME 0.000005 /*Sets the timeout value to 5
μs.* /
:BUS1:SPI:TIMEout:TIME?         /*The query returns
5.000000E-6.* /
```

3.8.13.12 :BUS<n>:SPI:SS:SOURce**Syntax**

```
:BUS<n>:SPI:SS:SOURce <source>
```


`:BUS<n>:SPI:SS:SOURce?`

Description

Sets or queries the source of the CS line of SPI decoding on the specified bus.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel3 |

Remarks

This setting command is only valid in timeout mode (To set or query the mode, send the `:BUS<n>:SPI:MODE` command).

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:BUS1:SPI:SS:SOURce CHANnel2 /*Sets the source of CS line of
SPI decoding to CHANnel2.*/
:BUS1:SPI:SS:SOURce? /*The query returns CHAN2.*/
```

3.8.13.13 :BUS<n>:SPI:SS:POLarity

Syntax

`:BUS<n>:SPI:SS:POLarity <polarity>`

`:BUS<n>:SPI:SS:POLarity?`

Description

Sets or queries the polarity of CS line of SPI decoding on the specified bus.

Parameter

| Name | Type | Range | Default |
|------------|----------|------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <polarity> | Discrete | {HIGH LOW} | LOW |

Remarks

- **HIGH:** indicates that the oscilloscope samples data of the source channel of data line on the specified edge of the clock signal when the CS signal is high level.
- **LOW:** indicates that the oscilloscope samples data of the source channel of data line on the specified edge of the clock signal when the CS signal is low level.

This setting command is only valid in timeout mode (To set or query the mode, send the `:BUS<n>:SPI:MODE` command).

Return Format

The query returns HIGH or LOW.

Example

```
:BUS1:SPI:SS:POLarity HIGH /*Sets the polarity of CS line of
SPI decoding to HIGH.*/
:BUS1:SPI:SS:POLarity? /*The query returns HIGH.*/
```

3.8.14 :BUS<n>:CAN

The `:BUS<n>:CAN` commands are used to set relevant parameters for CAN decoding.

3.8.14.1 :BUS<n>:CAN:SOURce**Syntax**

```
:BUS<n>:CAN:SOURce <source>
```

```
:BUS<n>:CAN:SOURce?
```

Description

Sets or queries the source of CAN decoding.

Parameter

| Name | Type | Range | Default |
|----------|----------|---------------------------------------|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:BUS1:CAN:SOURce CHANnel2 /*Sets the source of CAN decoding to
CHANnel2.*/
:BUS1:CAN:SOURce? /*The query returns CHAN2.*/
```

3.8.14.2 :BUS<n>:CAN:STYPe**Syntax**

```
:BUS<n>:CAN:STYPe <stype>
```

```
:BUS<n>:CAN:STYPe?
```

Description

Sets or queries the signal type of CAN decoding.

Parameter

| Name | Type | Range | Default |
|---------|----------|--------------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <stype> | Discrete | {TX RX CANH CANL DIFFerential} | CANL |

Remarks

- **TX:** indicates the Transmit signal from the CAN bus transceiver.
- **RX:** indicates the Receive signal from the CAN bus transceiver.
- **CANH:** indicates the actual CAN_H differential bus signal.
- **CANL:** indicates the actual CAN_L differential bus signal.
- **DIFFerential:** indicates the CAN differential bus signal connected to an analog channel by using a differential probe. Connect the differential probe's positive lead to the CAN_H bus signal and connect the negative lead to the CAN_L bus signal.

Return Format

The query returns TX, RX, CANH, CANL, or DIFF.

Example

```
:BUS1:CAN:SType TX /*Sets the signal type of CAN decoding
to TX.*/
:BUS1:CAN:SType? /*The query returns TX.*/
```

3.8.14.3 :BUS<n>:CAN:BAUD**Syntax**

```
:BUS<n>:CAN:BAUD <baud>
```

```
:BUS<n>:CAN:BAUD?
```

Description

Sets or queries the signal rate of CAN decoding. The unit is bps.

Parameter

| Name | Type | Range | Default |
|--------|----------|-----------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <baud> | Integer | 10kbps to 5Mbps | 1Mbps |

Remarks

If the baud rate is set to a value with "M", then "A" should be added at the end of the value. For example, if you send 5M, you need to send 5MA.

Return Format

The query returns an integer ranging from 10k to 5M.

Example

```
:BUS1:CAN:BAUD 120000 /*Sets the signal rate of CAN decoding to
120000 bps*/
:BUS1:CAN:BAUD? /*The query returns 120000.*/
```

3.8.14.4 :BUS<n>:CAN:SPOint**Syntax**

```
:BUS<n>:CAN:SPOint <spoint>
```

```
:BUS<n>:CAN:SPOint?
```

Description

Sets or queries the sample point position of CAN decoding (expressed in %).

Parameter

| Name | Type | Range | Default |
|----------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <spoint> | Integer | 10 to 90 | 50 |

Remarks

The sample point is within the range of the bit time. The oscilloscope samples the bit level at the sample point. The sample point position is expressed as the ratio of "time from the bit start to the sample point" to "bit time", in %.

Return Format

The query returns an integer ranging from 10 to 90.

Example

```
:BUS1:CAN:SPOint 70 /*Sets the sample point position of CAN
decoding to 70%.*/
:BUS1:CAN:SPOint? /*The query returns 70.*/
```

3.8.14.5 :BUS<n>:CAN:FDBaud**Syntax**

```
:BUS<n>:CAN:FDBaud <baud>
```

```
:BUS<n>:CAN:FDBaud?
```

Description

Sets or queries the signal rate of CAN-FD decoding on the specified decoding bus. The unit is bps.

Parameter

| Name | Type | Range | Default |
|--------|----------|-------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <baud> | Integer | 1 Mbps to 10 Mbps | 1 Mbps |

Remarks

If the baud rate is set to a value with "M", then "A" should be added at the end of the value. For example, if you send 5M, you need to send 5MA.

The DHO1000 series oscilloscope does not support this command. This command is optional for the DHO4000 series oscilloscope.

Return Format

The query returns an integer ranging from 1M to 10M.

Example

```
:BUS1:CAN:FDBaud 120000 /*Sets the signal rate of CAN-FD
decoding to 120000 bps.*/
:BUS1:CAN:FDBaud? /*The query returns 120000.*/
```

3.8.14.6 :BUS<n>:CAN:FDSPoint**Syntax**

```
:BUS<n>:CAN:FDSPoint <spoint>
```

```
:BUS<n>:CAN:FDSPoint?
```

Description

Sets or queries the sample point position of the CAN-FD decoding on the specified bus (expressed in %).

Parameter

| Name | Type | Range | Default |
|----------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <spoint> | Integer | 10 to 90 | 50 |

Remarks

The sample point is within the range of the bit time. The oscilloscope samples the bit level at the sample point. The sample point position is expressed as the ratio of "time from the bit start to the sample point" to "bit time", in %.

The DHO1000 series oscilloscope does not support this command. This command is optional for the DHO4000 series oscilloscope.

Return Format

The query returns an integer ranging from 10 to 90.

Example

```
:BUS1:CAN:FDSPoint 70 /*Sets the sample point position of
CAN-FD decoding to 70%.*/
:BUS1:CAN:FDSPoint? /*The query returns 70.*/
```

3.8.15 :BUS<n>:FLEXray

The :BUS<n>:FLEXray commands are used to set the relevant parameters for FLEXray decoding.

**NOTE**

The DHO1000 series oscilloscope does not support :BUS<n>:FLEXray commands.

The :BUS<n>:FLEXray commands are optional configurations for the DHO4000 series oscilloscope.

3.8.15.1 :BUS<n>:FLEXray:BAUD**Syntax**

```
:BUS<n> : FLEXray : BAUD <baud>
```

```
:BUS<n> : FLEXray : BAUD?
```

Description

Sets or queries the signal rate of FlexRay decoding. The default unit is bps.

Parameter

| Name | Type | Range | Default |
|--------|----------|----------------------------|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <baud> | Discrete | {2500000 5000000 10000000} | 10000000 |

Remarks

N/A

Return Format

The query returns 2500000, 5000000, or 10000000.

Example

```
:BUS1:FLEXray:BAUD 2500000 /*Sets the signal rate of
FlexRay decoding to 2500000 bps.*/
:BUS1:FLEXray:BAUD? /*The query returns 2500000.*/
```

3.8.15.2 :BUS<n>:FLEXray:SOURce**Syntax**

```
:BUS<n> : FLEXray : SOURce <source>
```

```
:BUS<n> : FLEXray : SOURce?
```

Description

Sets or queries the source of FlexRay decoding.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:BUS1:FLEXray:SOURce CHANnel2 /*Sets the source channel of
FlexRay decoding to CHANnel2.*/
:BUS1:FLEXray:SOURce? /*The query returns CHAN2.*/
```

3.8.15.3 :BUS<n>:FLEXray:SPOint**Syntax**

```
:BUS<n>:FLEXray:SPOint <spoint>
```

```
:BUS<n>:FLEXray:SPOint?
```

Description

Sets or queries the sample point position of FlexRay decoding (expressed in %).

Parameter

| Name | Type | Range | Default |
|----------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <spoint> | Integer | 10 to 90 | 50 |

Remarks

The sample point is within the range of the bit time. The oscilloscope samples the bit level at the sample point. The sample point position is expressed as the ratio of "time from the bit start to the sample point" to "bit time", in %.

Return Format

The query returns an integer ranging from 10 to 90.

Example

```
:BUS1:FLEXray: SPoint 70 /*Sets the sample point position
of FlexRay decoding to 70%.*/
:BUS1:FLEXray: SPoint? /*The query returns 70%.*/
```

3.8.15.4 :BUS<n>:FLEXray:STYPe**Syntax**

```
:BUS<n>:FLEXray:STYPe <stype>
```

```
:BUS<n>:FLEXray:STYPe?
```

Description

Sets or queries the signal type of FlexRay decoding on the specified bus.

Parameter

| Name | Type | Range | Default |
|---------|----------|------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <stype> | Discrete | {BP BM RT} | BP |

Remarks

N/A

Return Format

The query returns BP, BM, or RT.

Example

```
:BUS1:FLEXray:STYPe BM /*Sets the signal type of FlexRay
decoding to BM.*/
:BUS1:FLEXray:STYPe? /*The query returns BM.*/
```

3.8.15.5 :BUS<n>:FLEXray:CHANnel**Syntax**

```
:BUS <n>:FLEXray:CHANnel <ch>
```

```
:BUS <n>:FLEXray:CHANnel?
```

Description

Sets or queries the selected source channel of FlexRay decoding on the specified bus.

Parameter

| Name | Type | Range | Default |
|------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <ch> | Discrete | {A B} | A |

Remarks

N/A

Return Format

The query returns A or B.

Example

```
:BUS1:FLEXray CHANnel B.          /*Sets the source channel of
FlexRay decoding to B.*/
:BUS1:FLEXray:CHANnel?           /*The query returns B.*/
```

3.8.16 :BUS<n>:LIN

The :BUS<n>:LIN commands are used to set relevant parameters for LIN decoding.

NOTE

The :BUS<n>:LIN commands are standard configurations for the DHO1000 series oscilloscope.

The :BUS<n>:LIN commands are optional configurations for the DHO4000 series oscilloscope.

**3.8.16.1 :BUS<n>:LIN:PARity****Syntax**

```
:BUS<n>:LIN:PARity <bool>
```

```
:BUS<n>:LIN:PARity?
```

Description

Sets or queries whether the LIN decoding on the specified decoding bus includes the parity bit.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bool> | Bool | {{1 ON} {0 OFF}} | 0 OFF |

Remarks

- **1|ON:** includes the parity bit.
- **0|OFF:** does not include the parity bit.

Return Format

The query returns 0 or 1.

Example

```
:BUS1:LIN:PARity ON /*Sets the parity bit to be
included in LIN decoding.*/
:BUS1:LIN:PARity? /*The query returns 1.*/
```

3.8.16.2 :BUS<n>:LIN:SOURce**Syntax**

```
:BUS<n>:LIN:SOURce <source>
```

```
:BUS<n>:LIN:SOURce?
```

Description

Sets or queries the source of LIN bus.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:BUS1:LIN:SOURce CHANnel2 /*Sets the source of LIN bus to
CHANnel2.*/
:BUS1:LIN:SOURce? /*The query returns CHAN2.*/
```

3.8.16.3 :BUS<n>:LIN:STANdard

Syntax

```
:BUS<n>:LIN:STANdard <value>
```

```
:BUS<n>:LIN:STANdard?
```

Description

Sets or queries the version of LIN bus.

Parameter

| Name | Type | Range | Default |
|---------|----------|-----------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <value> | Discrete | {V1X V2X MIXed} | MIXed |

Remarks

N/A

Return Format

The query returns V1X, V2X, or MIX.

Example

```
:BUS1:LIN:STANdard V2X          /*Sets the LIN bus version to
V2X.*/
:BUS1:LIN:STANdard?            /*The query returns V2X.*/
```

3.8.17 :BUS<n>:IIS

The :BUS<n>:IIS commands are used to set relevant parameters for I2S decoding.

NOTE

The DHO1000 series oscilloscope does not support :BUS<n>:IIS commands.

The :BUS<n>:IIS commands are optional configurations for the DHO4000 series oscilloscope.

3.8.17.1 :BUS<n>:IIS:SOURce:CLOCK

Syntax

```
:BUS<n>:IIS:SOURce:CLOCK <source>
```

```
:BUS<n>:IIS:SOURce:CLOCK?
```

Description

Sets or queries the clock source of the I2S decoding.



Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:BUS1:IIS:SOURce:CLOCK CHANnel2 /*Sets the clock source of
the I2S decoding to CHANnel2.*/
:BUS1:IIS:SOURce:CLOCK? /*The query returns CHAN2.*/
```

3.8.17.2 :BUS<n>:IIS:SOURce:DATA**Syntax**

```
:BUS<n>:IIS:SOURce:DATA <source>
```

```
:BUS<n>:IIS:SOURce:DATA?
```

Description

Sets or queries the data source of the I2S decoding.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel3 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:BUS1:IIS:SOURce:DATA CHANnel2 /*Sets the data source of
the I2S decoding to CHANnel2.*/
:BUS1:IIS:SOURce:DATA? /*The query returns CHAN2.*/
```

3.8.17.3 :BUS<n>:IIS:SOURce:WSElect**Syntax**

```
:BUS<n>:IIS:SOURce:WSElect <source>
```

```
:BUS<n>:IIS:SOURce:WSElect?
```

Description

Sets or queries the audio channel of the I2S trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel2 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:BUS1:IIS:SOURce:WSElect CHANnel2 /*Sets the audio channel to
CHANnel2.*/
:BUS1:IIS:SOURce:WSElect? /*The query returns CHAN2.*/
```

3.8.17.4 :BUS<n>:IIS:ALIGnment**Syntax**

```
:BUS<n>:IIS:ALIGnment <align>
```

```
:BUS<n>:IIS:ALIGnment?
```

Description

Sets or queries the the alignment mode of I2S decoding on the specified bus.

Parameter

| Name | Type | Range | Default |
|---------|----------|-------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <align> | Discrete | {IIS RJ LJ} | IIS |

Remarks

- **IIS:** data transmission (MSB first) begins at the second edge of the WS transition.
- **RJ:** data transmission (MSB first) is right-justified to the WS transition.
- **LJ:** data transmission (MSB first) begins at the edge of the WS transition.

Return Format

The query returns IIS, RJ, or LJ.

Example

```
:BUS1:IIS:ALIGNment RJ          /*Sets the alignment mode of the
I2S decoding to RJ.*/
:BUS1:IIS:ALIGNment?           /*The query returns RJ.*/
```

3.8.17.5 :BUS<n>:IIS:CLOCK:SLOPe**Syntax**

```
:BUS<n> : IIS : CLOCK : SLOPe <slope>
```

```
:BUS<n> : IIS : CLOCK : SLOPe?
```

Description

Sets or queries the clock edge type of the I2S decoding.

Parameter

| Name | Type | Range | Default |
|---------|----------|---------------------|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <slope> | Discrete | {NEGative POSitive} | POSitive |

Remarks

N/A

Return Format

The query returns POS or NEG.

Example

```
:BUS1:IIS:CLOCK:SLOPe NEGative /*Sets the clock edge of I2S
decoding to NEGative.*/
:BUS1:IIS:CLOCK:SLOPe? /*The query returns NEG.*/
```

3.8.17.6 :BUS<n>:IIS:RWIDth**Syntax**

```
:BUS<n>:IIS:RWIDth <val>
```

```
:BUS<n>:IIS:RWIDth?
```

Description

Sets or queries the word size of the I2S decoding.

Parameter

| Name | Type | Range | Default |
|-------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <val> | Integer | 4 to 32 | 4 |

Remarks

N/A

Return Format

The query returns an integer ranging from 4 to 32.

Example

```
:BUS1:IIS:RWIDth 5 /*Sets the word size of I2S
decoding to 5.*/
:BUS1:IIS:RWIDth? /*The query returns 5.*/
```

3.8.17.7 :BUS<n>:IIS:RECewidth**Syntax**

```
:BUS<n>:IIS:RECewidth <val>
```

```
:BUS<n>:IIS:RECewidth? <val>
```

Description

Sets or queries the receive width of I2S decoding on the specified bus.

Parameter

| Name | Type | Range | Default |
|-------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <val> | Integer | 4 to 32 | 4 |

Remarks

N/A

Return Format

The query returns an integer ranging from 4 to 32.

Example

```
:BUS1:IIS:RECewidth 5 /*Sets the receive width of I2S
decoding to 5.*/
:BUS1:IIS:RECewidth? /*The query returns 5.*/
```

3.8.17.8 :BUS<n>:IIS:WSLow**Syntax**

```
:BUS<n>:IIS:WSLow <WS>
```

```
:BUS<n>:IIS:WSLow?
```

Description

Sets or queries the audio polarity of I2S decoding on the specified bus.

Parameter

| Name | Type | Range | Default |
|------|----------|--------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <ws> | Discrete | {LEFT RIGHT} | - |

Remarks

- **LEFT:** indicates that the WS Low is Left.
- **RIGHT:** indicates that the WS Low is Right.

Return Format

The query returns LEFT or RIGH.

Example

```
:BUS1:IIS:WSLow LEFT /*Sets the audio polarity of I2S decoding
to Left.*/
:BUS1:IIS:WSLow? /*The query returns LEFT.*/
```

3.8.17.9 :BUS<n>:IIS:ENDian**Syntax**

```
:BUS<n>:IIS:ENDian <endian>
```

```
:BUS<n>:IIS:ENDian?
```

Description

Sets or queries the bit order of data of I2S decoding on the specified bus.

Parameter

| Name | Type | Range | Default |
|----------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <endian> | Discrete | {MSB LSB} | MSB |

Remarks

- **MSB:** indicates Most Significant Bit transmission sequence, i.g. the highest bit of the data is transmitted first.
- **LSB:** indicates Least Significant Bit transmission sequence, i.g. the lowest bit of the data is transmitted first.

Return Format

The query returns MSB or LSB.

Example

```
:BUS1:IIS:ENDian LSB /*Sets the endian of I2S decoding to
LSB.*/
:BUS1:IIS:ENDian? /*The query returns LSB.*/
```

3.8.17.10 :BUS<n>:IIS:POLarity**Syntax**

```
:BUS<n>:IIS:POLarity <pol>
```

```
:BUS<n>:IIS:POLarity?
```

Description

Sets or queries the data polarity of I2S decoding on the specified bus.

Parameter

| Name | Type | Range | Default |
|-------|----------|---------------------|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <pol> | Discrete | {POSitive NEGative} | POSitive |

Remarks

- **POSitive:** indicates positive polarity.
- **NEGative:** indicates negative polarity.

Return Format

The query returns POS or NEG.

Example

```
:BUS1:IIS:POLarity NEGative /*Sets the data polarity of I2S
decoding to Negative.*/
:BUS1:IIS:POLarity? /*The query returns NEG.*/
```

3.8.18 :BUS<n>:M1553

The :BUS<n>:M1553 commands are used to set relevant parameters for M1553 decoding.

**NOTE**

The DHO1000 series oscilloscope does not support : BUS<n>:M1553 commands.

The : BUS<n>:M1553 commands are optional configurations for the DHO4000 series oscilloscope.

3.8.18.1 :BUS<n>:M1553:SOURce**Syntax**

```
:BUS<n>:M1553:SOURce <source>
```

```
:BUS<n>:M1553:SOURce?
```

Description

Sets or queries the source of the M1553 decoding.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:BUS1:M1553:SOURce CHANnel2 /*Sets the source of the M1553
decoding to CHANnel2.*/
:BUS1:M1553:SOURce? /*The query returns CHAN2.*/
```

3.9 :CHANnel<n> Commands

The **:CHANnel<n>** commands are used to set or query the bandwidth limit, coupling, vertical scale, vertical offset, and other vertical system parameters of the analog channel.

3.9.1 :CHANnel<n>:BWLimit

Syntax

```
:CHANnel<n>:BWLimit <val>
```

```
:CHANnel<n>:BWLimit?
```

Description

Sets or queries the bandwidth limit of the specified channel.

Parameter

| Name | Type | Range | Default |
|-------|----------|-------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <val> | Discrete | Refer to <i>Remarks</i> | OFF |

Remarks

- **1 MΩ:** {OFF|ON|20M|250M}, with the unit Hz.

- **50 Ω:** {OFF|ON|20M|250M}, with the unit Hz.

When set to OFF, the bandwidth limit is disabled.

Neither the 50 Ω input impedance nor the 250 MHz bandwidth under 1 MΩ is supported by the DHO1000 series.

Return Format

- **1 MΩ:** The query returns 20M, 250M, ON or OFF.
- **50 Ω:** The query returns 20M, 250M, ON, or OFF.

Example

```
:CHANnel1:BWLimit 20M /*Enables the 20MHz bandwidth limit.*/
:CHANnel1:BWLimit? /*The query returns 20M.*/
```

3.9.2 :CHANnel<n>:COUPLing

Syntax

```
:CHANnel<n>:COUPLing <coupling>
```

```
:CHANnel<n>:COUPLing?
```

Description

Sets or queries the coupling mode of the specified channel.

Parameter

| Name | Type | Range | Default |
|------------|----------|-------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <coupling> | Discrete | {AC DC GND} | DC |

Description

- **AC:** the DC components of the signal under test are blocked.
- **DC:** both DC and AC components of the signal under test can pass through the channel.
- **GND:** both DC and AC components of the signal under test are blocked.

Return Format

The query returns AC, DC, or GND.

Example

```
:CHANnel1:COUPling AC /*Selects the AC coupling mode.*/
:CHANnel1:COUPling? /*The query returns AC.*/
```

3.9.3 :CHANnel<n>:DISPlay**Syntax**

```
:CHANnel<n>:DISPlay <bool>
```

```
:CHANnel<n>:DISPlay?
```

Description

Enables or disables the specified channel; or queries the on/off status of the specified channel.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bool> | Bool | {{1 ON} {0 OFF}} | 1 ON |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:CHANnel1:DISPlay ON /*Enables CH1.*/
:CHANnel1:DISPlay? /*The query returns 1.*/
```

3.9.4 :CHANnel<n>:INVert**Syntax**

```
:CHANnel<n>:INVert <bool>
```

```
:CHANnel<n>:INVert?
```

Description

Turns on or off the waveform invert for the specified channel; or queries the on/off status of the waveform invert for the specified channel.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

When the waveform invert is turned off, the waveform is displayed normally; when the waveform invert is turned on, the voltage values of the displayed waveform are inverted.

Return Format

The query returns 1 or 0.

Example

```
:CHANnel1:INVert ON /*Enables the waveform invert for CH1.*/
:CHANnel1:INVert? /*The query returns 1.*/
```

3.9.5 :CHANnel<n>:OFFSet**Syntax**

```
:CHANnel<n>:OFFSet <offset>
```

```
:CHANnel<n>:OFFSet?
```

Description

Sets or queries the vertical offset of the specified channel. The default unit is V.

Parameter

| Name | Type | Range | Default |
|----------|----------|-------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <offset> | Real | Refer to <i>Remarks</i> | 0 V |

Remarks

The vertical offset of the specified channel is related to the vertical scale and the input impedance.

- For the DHO4000 series, when the input impedance is 1 MΩ:
 - ±0.5 V (<500 μV/div)
 - ±1 V (500 μV/div to 65 mV/div)

- ± 10 V (65.1 mV/div to 270 mV/div)
- ± 20 V (270.1 mV/div to 2.75 V/div)
- ± 100 V (2.7501 V/div to 10 V/div)
- For the DHO4000 series, when the input impedance is 50 Ω :
 - ± 1 V (≤ 135 mV/div)
 - ± 4 V (> 135 mV/div)
- For the DHO1000 series, when the input impedance is 1 M Ω :
 - ± 1 V (≤ 65 mV/div)
 - ± 10 V (65.1 mV/div to 270 mV/div)
 - ± 20 V (270.1 mV/div to 2.75 V/div)
 - ± 100 V (2.7501 V/div to 10 V/div)
- The DHO1000 series does not support the 50 Ω input impedance.

Return Format

The query returns the vertical offset in scientific notation.

Example

```
:CHANnel1:OFFSet 0.01 /*Sets the vertical offset of CH1 to 10
mV.*/
:CHANnel1:OFFSet? /*The query returns 1E-2.*/
```

3.9.6 :CHANnel<n>:TCALibrate

Syntax

```
:CHANnel<n>:TCALibrate <val>
```

```
:CHANnel<n>:TCALibrate?
```

Description

Sets or queries the delay calibration time (used to calibrate the zero offset of the corresponding channel) of the specified channel. The unit is s.

Parameter

| Name | Type | Range | Default |
|-------|----------|-------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <val> | Real | -100 ns to 100 ns | 0 s |

Remarks

When the horizontal time base is greater than 10 μ s, the parameter <val> cannot be set.

Return Format

The query returns the delay calibration time in scientific notation.

Example

```
:CHANnel1:TCALibrate 0.00000002 /*Sets the delay calibration time
to 20 ns.*/
:CHANnel1:TCALibrate? /*The query returns 2.000000E-8.*/
```

3.9.7 :CHANnel<n>:SCALE**Syntax**

```
:CHANnel<n>:SCALE <scale>
```

```
:CHANnel<n>:SCALE?
```

Description

Sets or queries the vertical scale of the specified channel.

Parameter

| Name | Type | Range | Default |
|---------|----------|-------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <scale> | Real | Refer to <i>Remarks</i> | 50 mV |

Remarks

The vertical scale of the specified channel is related to the input impedance and the probe ratio.

- For the DHO4000 series, when the input impedance is 1 M Ω and the probe ratio is 1X, the range of <scale> is from 100 μ V/div to 10 V/div.
- For the DHO4000 series, when the input impedance is 50 Ω and the probe ratio is 1X, the range of <scale> is from 100 μ V/div to 1 V/div.
- For the DHO1000 series, when the input impedance is 1 M Ω and the probe ratio is 1X, the range of <scale> is from 500 μ V/div to 10 V/div.
- The DHO1000 series does not support the 50 Ω input impedance.

Return Format

The query returns the vertical scale in scientific notation. The unit is V.

Example

```
:CHANnel1:SCALe 0.1/*Sets the vertical scale of CH1 to 0.1 V/div.*/
:CHANnel1:SCALe? /*The query returns 1E-1.*/
```

3.9.8 :CHANnel<n>:IMPedance**Syntax**

```
:CHANnel<n>:IMPedance <impedance>
```

```
:CHANnel<n>:IMPedance?
```

Description

Sets or queries the input impedance of the specified analog channel.

Parameter

| Name | Type | Range | Default |
|-------------|----------|--------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <impedance> | Discrete | {OMEG FIFTy} | OMEG |

Remarks

- **OMEG:** 1M Ω . This indicates that the input impedance of the oscilloscope is rather high, and the current flowing from the circuit under test to the oscilloscope can be ignored.
- **FIFTy:** 50 Ω . The oscilloscope shall match a device whose output impedance is 50 Ω .

The DHO1000 series does not support the 50 Ω input impedance.

Return Format

The query returns OMEG or FIFT.

Example

```
:CHANnel1:IMPedance OMEG /*Sets the input impedance of CH1 to 1
M $\Omega$ .*/
:CHANnel1:IMPedance? /*The query returns OMEG.*/
```

3.9.9 :CHANnel<n>:BVOLtage

Syntax

```
:CHANnel<n>:BVOLtage <vol>
```

```
:CHANnel<n>:BVOLtage?
```

Description

Sets or queries the bias voltage of the probe being connected to the specified channel. The default unit is V.

Parameter

| Name | Type | Range | Default |
|-------|----------|-----------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <vol> | Real | Affected by the probe model | 0 V |

Remarks

- This function is used to adjust the signal under test that exceeds the input dynamic range of the probe amplifier to an appropriate range to ensure the integrity of the signal under test.
- This command is only valid when an active probe is correctly connected.
- This command is compatible with the specified command in MSO4000. It functions the same as the `:CHANnel<n>:PROBe:BIAS` command in this manual.

Return Format

The query returns the bias voltage of the probe in scientific notation.

Example

```
:CHANnel1:BVOLtage 0 /*Sets the bias voltage of probe currently
connected to CH1 to 0 V.*/
:CHANnel1:BVOLtage? /*The query returns 0.000000.*/
```

3.9.10 :CHANnel<n>:CStart

Syntax

```
:CHANnel<n>:CStart
```

Description

Starts calibration for the active probe currently connected to the specified analog channel.

Parameter

| Name | Type | Range | Default |
|------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |

Remarks

This command is only valid when an active probe is correctly connected.

Return Format

N/A

Example

N/A

3.9.11 :CHANnel<n>:PROBe

Syntax

```
:CHANnel<n>:PROBe <atten>
```

```
:CHANnel<n>:PROBe?
```

Description

Sets or queries the probe ratio of the specified analog channel.

Parameter

| Name | Type | Range | Default |
|---------|----------|--|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <atten> | Discrete | {0.001 0.002 0.005 0.01 0.02 0.05 0.1 0.2 0.5 1 2 5 10 20 50 100 200 500 1000 2000 5000 10000 20000 50000} | 1 |

Remarks

- Sets the probe ratio. That is, multiply the acquired signal by a specified number (not affect the actual amplitude of the signal).
- The set probe ratio affects the settable range of the current vertical scale.

Return Format

The query returns 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000, 20000, or 50000.

Example

```
:CHANnel1:PROBe 10 /*Sets the probe ratio of CH1 to 10X.*/
:CHANnel1:PROBe? /*The query returns 10.*/
```

3.9.12 :CHANnel<n>:PROBe:DELAy**Syntax**

```
:CHANnel<n>:PROBe:DELAy <delay>
```

```
:CHANnel<n>:PROBe:DELAy?
```

Description

Sets or queries the probe delay time of the specified channel.

Parameter

| Name | Type | Range | Default |
|---------|----------|-------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <delay> | Real | -100 ns to 100 ns | 0 s |

Remarks

- To avoid measurement result errors arising from the transmission delay of the probe cable, the oscilloscope provides the probe delay adjustment function for the active probe.
- This command is only valid when an active probe is correctly connected.

Return Format

The query returns the probe delay time in scientific notation.

Example

```
:CHANnel1:PROBe:DELAy 0.00000001 /*Sets the probe delay time of
CH1 to 10 ns.*/
:CHANnel1:PROBe:DELAy? /*The query returns 1E-8.*/
```

3.9.13 :CHANnel<n>:PROBe:BIAS**Syntax**

```
:CHANnel<n>:PROBe:BIAS <bias>
```

```
:CHANnel<n>:PROBe:BIAS?
```

Description

Sets or queries the probe bias voltage for the specified channel.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bias> | Real | Affected by the probe model. | 0 V |

Remarks

- This function is used to adjust the signal under test that exceeds the input dynamic range of the probe amplifier to an appropriate range to ensure the signal integrity.
- This command is only valid when an active probe is correctly connected.

Return Format

The query returns the probe bias voltage in scientific notation.

Example

N/A

3.9.14 :CHANnel<n>:PROBe:CALibration

Syntax

```
:CHANnel<n>:PROBe:CALibration
```

Description

Performs the probe calibration.

Parameter

| Name | Type | Range | Default |
|------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |

Remarks

N/A

Return Format

N/A

Example

```
:CHANnel1:PROBe:CALibration /*Performs the probe calibration for CH1.*/
```

3.9.15 :CHANnel<n>:PROBe:DEMag**Syntax**

```
:CHANnel<n>:PROBe:DEMag
```

Description

Starts to demagnetize the current probe that is connected to the specified channel.

Parameter

| Name | Type | Range | Default |
|------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |

Remarks

This command is only valid when a current probe is correctly connected.

Return Format

N/A

Example

```
:CHANnel1:PROBe:DEMag /*Starts to demagnetize the current probe that is connected to CH1.*/
```

3.9.16 :CHANnel<n>:LABel:SHOW**Syntax**

```
:CHANnel<n>:LABel:SHOW <bool>
```

```
:CHANnel<n>:LABel:SHOW?
```

Description

Sets or queries whether to display the label of the specified channel.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bool> | Bool | {{1 ON}}{0 OFF}} | - |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:CHANnel1:LABel:SHOW ON          /*Displays the label of the
specified channel.*/
:CHANnel1:LABel:SHOW?           /*The query returns 1.*/
```

3.9.17 :CHANnel<n>:LABel:CONTent**Syntax**

```
:CHANnel<n>:LABel:CONTent <str>
```

```
:CHANnel<n>:LABel:CONTent?
```

Description

Sets or queries the label of the specified channel.

Parameter

| Name | Type | Range | Default |
|-------|--------------|---|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <str> | ASCII String | The label can contain English letters and numbers, as well as some symbols. | - |

Remarks

N/A

Return Format

The query returns the label of the specified channel in strings.

Example

```
:CHANnel1:LABel:CONTent ch1      /*Sets the label of Channel 1 to
ch1.*/
:CHANnel1:LABel:CONTent? /*The query returns ch1.*/
```

3.9.18 :CHANnel<n>:UNITs**Syntax**

```
:CHANnel<n>:UNITs <units>
```


:CHANnel <n> :UNITs?

Description

Sets or queries the amplitude display unit of the specified analog channel.

Parameter

| Name | Type | Range | Default |
|---------|----------|-------------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <units> | Discrete | {WATT AMPere VOLTage UNKNown} | VOLTage |

Remarks

N/A

Return Format

The query returns VOLT, WATT, AMP, or UNKN.

Example

```
:CHANnel1:UNITs VOLTage /*Sets the amplitude display unit of CH1
to VOLTage.*/
:CHANnel1:UNITs? /*The query returns VOLT.*/
```

3.9.19 :CHANnel<n>:VERNier

Syntax

:CHANnel <n> :VERNier <bool>

:CHANnel <n> :VERNier?

Description

Enables or disables the fine adjustment of the vertical scale of the specified channel; or queries the on/off status of the fine adjustment of the vertical scale of the specified channel.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:CHANnel1:VERNier ON /*Enables the fine adjustment of the
vertical scale of CH1.*/
:CHANnel1:VERNier? /*The query returns 1.*/
```

3.9.20 :CHANnel<n>:POSition

Syntax

```
:CHANnel<n>:POSition <offset>
```

```
:CHANnel<n>:POSition?
```

Description

Sets or queries the bias voltage of the specified channel.

Parameter

| Name | Type | Range | Default |
|----------|----------|-------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <offset> | Real | Refer to <i>Remarks</i> | 0 |

Remarks

The range of the bias voltage of the specified channel is related to the vertical scale and the input impedance.

- For the DHO4000 series, when the input impedance is 1 M Ω :
 - ± 0.5 V (<500 μ V/div)
 - ± 1 V (500 μ V/div to 65 mV/div)
 - ± 10 V (65.1 mV/div to 270 mV/div)
 - ± 20 V (270.1 mV/div to 2.75 V/div)
 - ± 100 V (2.7501 V/div to 10 V/div)
- For the DHO4000 series, when the input impedance is 50 Ω :
 - ± 1 V (\leq 135 mV/div)
 - ± 4 V (>135 mV/div)
- For the DHO1000 series, when the input impedance is 1 M Ω :

- ± 1 V (≤ 65 mV/div)
 - ± 10 V (65.1 mV/div to 270 mV/div)
 - ± 20 V (270.1 mV/div to 2.75 V/div)
 - ± 100 V (2.7501 V/div to 10 V/div)
- The DHO1000 series does not support the 50 Ω input impedance.

Return Format

The query returns the bias voltage of the specified channel in scientific notation.

Example

```
:CHANnel1:POSition 10 /*Sets the bias voltage of CH1 to 10 V.*/  
:CHANnel1:POSition? /*The query returns 1.000000E+01.*/
```

3.10 :COUNter Commands

3.10.1 :COUNter:CURRent?

Syntax

```
:COUNter:CURRent?
```

Description

Queries the measurement value of the frequency counter.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the current measurement value of the frequency counter in scientific notation.

Example

```
N/A
```

3.10.2 :COUNter:ENABLE

Syntax

```
:COUNter:ENABLE <bool>
```

```
:COUNter:ENABLE?
```

Description

Enables or disables the frequency counter; or queries the on/off status of the frequency counter.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:COUNTER:ENABLE ON /*Enables the frequency counter.*/
:COUNTER:ENABLE? /*The query returns 1.*/
```

3.10.3 :COUNTER:SOURce**Syntax**

```
:COUNTER:SOURce <source>
```

```
:COUNTER:SOURce?
```

Description

Sets or queries the source of the frequency counter.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 EXT} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, or EXT.

Example

```
:COUNter:SOURce CHANnel2      /*Sets the source of the frequency
counter to CHANnel2.*/
:COUNter:SOURce?              /*The query returns CHAN2.*/
```

3.10.4 :COUNter:MODE**Syntax**

```
:COUNter:MODE <mode>
```

```
:COUNter:MODE?
```

Description

Sets or queries the mode of the frequency counter.

Parameter

| Name | Type | Range | Default |
|--------|----------|-----------------------------|-----------|
| <mode> | Discrete | {FREQuency PERiod TOTalize} | FREQuency |

Remarks

- **FREQuency:** indicates the Frequency measurement.
- **PERiod:** indicates the Period measurement.
- **TOTalize:** indicates the Totalize measurement.

Return Format

The query returns **FREQ**, **PER**, or **TOT**.

Example

```
:COUNter:MODE PERiod      /*Sets the mode of the frequency counter
to PERiod.*/
:COUNter:MODE?            /*The query returns PER.*/
```

3.10.5 :COUNter:NDIGits**Syntax**

```
:COUNter:NDIGits <val>
```

```
:COUNter:NDIGits?
```

Description

Sets or queries the resolution of the frequency counter.

Parameter

| Name | Type | Range | Default |
|-------|---------|--------|---------|
| <val> | Integer | 3 to 6 | 4 |

Remarks

The resolution setting is only available for "Period" and "Frequency", and unavailable for "Totalize".

Return Format

The query returns an integer ranging from 3 to 6.

Example

```
:COUNTER:NDIGits 4 /*Sets the resolution of the frequency
counter to 4.*/
:COUNTER:NDIGits? /*The query returns 4.*/
```

3.10.6 :COUNTER:TOTALize:ENABLE**Syntax**

```
:COUNTER:TOTALize:ENABLE <bool>
```

```
:COUNTER:TOTALize:ENABLE?
```

Description

Enables or disables the statistical function of the frequency counter; or queries the on/off status of the statistical function of the frequency counter.

Parameter

| Name | Type | Range | Default |
|--------|------|--------------------|---------|
| <bool> | Bool | {{1 ON}} {{0 OFF}} | 0 OFF |

Remarks

The statistical function is only available for "Period" and "Frequency", but it is unavailable for "Totalize".

Return Format

The query returns 1 or 0.

Example

```
:COUNTER:TOTALize:ENABLE ON /*Enables the statistical function
of the frequency counter.*/
:COUNTER:TOTALize:ENABLE? /*The query returns 1.*/
```

3.10.7 :COUNter:TOTAlize:CLEAr

Syntax

```
:COUNter:TOTAlize:CLEAr
```

Description

Clears the total count.

Parameter

N/A

Remarks

Available when "Totalize", "Frequency", or "Period" is selected under "Measure".

Return Format

N/A

Example

N/A

3.11 :CURSor Commands

The :CURSor commands are used to measure the X axis values (e.g. Time) and Y axis values (e.g. Voltage) of the waveform on the screen.

3.11.1 :CURSor:MODE

Syntax

```
:CURSor:MODE <mode>
```

```
:CURSor:MODE?
```

Description

Sets or queries the mode of the cursor measurement.

Parameter

| Name | Type | Range | Default |
|--------|----------|-----------------------|---------|
| <mode> | Discrete | {OFF MANual TRACK XY} | OFF |

Remarks

- **OFF:** disables the cursor measurement function.

- **MANual:** enables the manual mode of cursor measurement.
- **TRACk:** enables the track mode of cursor measurement.
- **XY:** enables the XY mode of cursor measurement. It is only valid when you select "XY" mode.

Return Format

The query returns OFF, MAN, TRAC, or XY.

Example

```
:CURSor:MODE MANual /*Selects the manual mode of cursor
measurement.*/
:CURSor:MODE? /*The query returns MAN.*/
```

3.11.2 :CURSor:MEASure:INDicator

Syntax

```
:CURSor:MEASure:INDicator <bool>
```

```
:CURSor:MEASure:INDicator?
```

Description

Sets or queries the on/off status of the indicator for the measurement function.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 0 or 1.

Example

```
:CURSor:MEASure:INDicator? ON /*Sets the indicator for the
measurement function to ON.*/
:CURSor:MEASure:INDicator? /*The query returns 1.*/
```

3.11.3 :CURSor:MANual:TYPE

Syntax

```
:CURSor:MANual:TYPE <type>
```


`:CURSor:MANual:TYPE?`

Description

Sets or queries the cursor type in the manual mode of cursor measurement.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <type> | Discrete | {TIME AMPLitude} | TIME |

Remarks

- **TIME:** indicates X cursor, which is often used to measure the time parameters.
- **AMPLitude:** indicates Y cursor, which is often used to measure the voltage parameters.

Return Format

The query returns TIME or AMPL.

Example

```
:CURSor:MANual:TYPE AMPLitude      /*Sets the cursor type to
AMPLitude.*/
:CURSor:MANual:TYPE?                 /*The query returns AMPL.*/
```

3.11.4 :CURSor:MANual:SOURce

Syntax

`:CURSor:MANual:SOURce <source>`

`:CURSor:MANual:SOURce?`

Description

Sets or queries the channel source of the manual mode of cursor measurement.

Parameter

| Name | Type | Range | Default |
|----------|----------|--|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 MATH1 MATH2 MATH3 MATH4 NONE} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, MATH1, MATH2, MATH3, MATH4, or NONE.

Example

```
:CURSor:MANual:SOURce CHANnel2 /*Sets the channel source to
CHANnel2.*/
:CURSor:MANual:SOURce? /*The query returns CHAN2.*/
```

3.11.5 :CURSor:MANual:CAX**Syntax**

```
:CURSor:MANual:CAX <ax>
```

```
:CURSor:MANual:CAX?
```

Description

Sets or queries the horizontal position of Cursor A in the manual mode of cursor measurement.

Parameter

| Name | Type | Range | Default |
|------|------|-------------------------|---------|
| <ax> | Real | Refer to <i>Remarks</i> | - |

Remarks

The range of the horizontal position of Cursor A is determined by the current horizontal scale and position.

Return Format

The query returns the horizontal position of Cursor A scientific notation. The unit is s.

Example

```
:CURSor:MANual:CAX 0.00000001 /*Sets the horizontal position of
Cursor A to 10 ns.*/
:CURSor:MANual:CAX? /*The query returns 1.000000E-8.*/
```

3.11.6 :CURSor:MANual:CAY**Syntax**

```
:CURSor:MANual:CAY <ay>
```

```
:CURSor:MANual:CAY?
```

Description

Sets or queries the vertical position of Cursor A in the manual mode of cursor measurement.

Parameter

| Name | Type | Range | Default |
|------|------|-------------------------|---------|
| <ay> | Real | Refer to <i>Remarks</i> | - |

Remarks

The range of the vertical position of Cursor A is determined by the current vertical scale and position.

Return Format

The query returns the vertical position of Cursor A in scientific notation. The unit is V.

Example

```
:CURSOR:MANual:CAY 0.1 /*Sets the vertical position of Cursor A
to 0.1 V.*/
:CURSOR:MANual:CAY? /*The query returns 1.000000E-1.*/
```

3.11.7 :CURSOR:MANual:CBX**Syntax**

```
:CURSOR:MANual:CBX <bx>
```

```
:CURSOR:MANual:CBX?
```

Description

Sets or queries the horizontal position of Cursor B in the manual mode of cursor measurement.

Parameter

| Name | Type | Range | Default |
|------|------|-------------------------|---------|
| <bx> | Real | Refer to <i>Remarks</i> | - |

Remarks

The range of the horizontal position of Cursor B is determined by the current horizontal scale and position.

Return Format

The query returns the horizontal position of Cursor B in scientific notation. The unit is s.

Example

```
:CURSor:MANual:CBX 0.00000001 /*Sets the horizontal position of
Cursor B to 10 ns.*/
:CURSor:MANual:CBX? /*The query returns 1.000000E-8.*/
```

3.11.8 :CURSor:MANual:CBY**Syntax**

```
:CURSor:MANual:CBY <by>
```

```
:CURSor:MANual:CBY?
```

Description

Sets or queries the vertical position of Cursor B in the manual mode of cursor measurement.

Parameter

| Name | Type | Range | Default |
|------|------|-------------------------|---------|
| <by> | Real | Refer to <i>Remarks</i> | - |

Remarks

The range of the vertical position of Cursor B is determined by the current vertical scale and position.

Return Format

The query returns the vertical position of Cursor B in scientific notation. The unit is V.

Example

```
:CURSor:MANual:CBY 0.1 /*Sets the vertical position of Cursor B
to 0.1 V.*/
:CURSor:MANual:CBY? /*The query returns 1.000000E-1.*/
```

3.11.9 :CURSor:MANual:AXValue?**Syntax**

```
:CURSor:MANual:AXValue?
```

Description

Queries the X value at Cursor A in the manual mode of cursor measurement. The unit is determined by the horizontal unit selected for the currently corresponding channel.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the X value at Cursor A in scientific notation.

Example

N/A

3.11.10 :CURSor:MANual:AYValue?

Syntax

```
:CURSor:MANual:AYValue?
```

Description

Queries the Y value at Cursor A in the manual mode of cursor measurement. The unit is determined by the currently selected vertical unit.

Parameter

N/A

Remarks

- The returned value is the same as the measurement value in the Cursor interface. Therefore, the unit is related to the vertical unit. When the vertical unit of cursor is set to Source, the unit of the returned value is the same as vertical unit of the channel.
- No value is returned when the cursor measurement value is invalid.

Return Format

The query returns the Y value at Cursor A in scientific notation.

Example

N/A

3.11.11 :CURSor:MANual:BXValue?

Syntax

```
:CURSor:MANual:BXValue?
```

Description

Queries the X value at Cursor B in the manual mode of cursor measurement. The unit is determined by the currently selected horizontal unit.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the X value at Cursor B in scientific notation.

Example

N/A

3.11.12 :CURSOR:MANual:BYValue?

Syntax

```
:CURSOR:MANual:BYValue?
```

Description

Queries the Y value at Cursor B in the manual mode of cursor measurement. The unit is determined by the currently selected vertical unit.

Parameter

N/A

Remarks

- The returned value is the same as the measurement value in the Cursor interface. Therefore, the unit is related to the vertical unit. When the vertical unit of cursor is set to Source, the unit of the returned value is the same as vertical unit of the channel.
- No value is returned when the cursor measurement value is invalid.

Return Format

The query returns the Y value at Cursor B in scientific notation.

Example

N/A

3.11.13 :CURSOR:MANual:XDELta?

Syntax

:CURSOR:MANual:XDELta?

Description

Queries the difference (ΔX) between the X value at Cursor A and the X value at Cursor B in the manual mode of cursor measurement. The unit is determined by the currently selected horizontal unit.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the current difference in scientific notation.

Example

N/A

3.11.14 :CURSOR:MANual:IXDELta?

Syntax

:CURSOR:MANual:IXDELta?

Description

Queries the reciprocal ($1/\Delta X$) of the absolute difference between the X value at Cursor A and the X value at Cursor B in the manual mode of cursor measurement. The unit is determined by the currently selected horizontal unit.

Parameter

N/A

Remarks

N/A

Return Format

The query returns $1/\Delta X$ in scientific notation.

Example

N/A

3.11.15 :CURSOR:MANual:YDELta?

Syntax

```
:CURSOR:MANual:YDELta?
```

Description

Queries the difference (ΔY) between the Y value at Cursor A and the Y value at Cursor B in the manual mode of cursor measurement. The unit is determined by the currently selected vertical unit.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the current difference in scientific notation.

Example

```
N/A
```

3.11.16 :CURSOR:TRACK:SOURce1

Syntax

```
:CURSOR:TRACK:SOURce1 <source>
```

```
:CURSOR:TRACK:SOURce1?
```

Description

Sets or queries the channel source of Cursor A in the track mode of cursor measurement.

Parameter

| Name | Type | Range | Default |
|----------|----------|--|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 MATH1 MATH2 MATH3 MATH4 NONE} | CHANnel1 |

Remarks

When no channel is enabled, sending this command will enable the corresponding channel.

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, MATH1, MATH2, MATH3, MATH4, or NONE.

Example

```
:CURSOR:TRACK:SOURce1 CHANnel2 /*Sets the channel source to
CHANnel2.*/
:CURSOR:TRACK:SOURce1? /*The query returns CHAN2.*/
```

3.11.17 :CURSOR:TRACK:SOURce2**Syntax**

```
:CURSOR:TRACK:SOURce2 <source>
:CURSOR:TRACK:SOURce2?
```

Description

Sets or queries the channel source of Cursor B in the track mode of cursor measurement.

Parameter

| Name | Type | Range | Default |
|----------|----------|--|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 MATH1 MATH2 MATH3 MATH4 NONE} | CHANnel1 |

Remarks

When no channel is enabled, sending this command will enable the corresponding channel.

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, MATH1, MATH2, MATH3, MATH4, or NONE.

Example

```
:CURSOR:TRACK:SOURce2 CHANnel2 /*Sets the channel source to
CHANnel2.*/
:CURSOR:TRACK:SOURce2? /*The query returns CHAN2.*/
```

3.11.18 :CURSOR:TRACK:CAX**Syntax**

```
:CURSOR:TRACK:CAX <ax>
:CURSOR:TRACK:CAX?
```

Description

Sets or queries the horizontal position of Cursor A in the track mode of cursor measurement.

Parameter

| Name | Type | Range | Default |
|------|------|-------------------------|---------|
| <ax> | Real | Refer to <i>Remarks</i> | - |

Remarks

The range of the horizontal position of Cursor A is determined by the current horizontal scale and position.

Return Format

The query returns the horizontal position of Cursor A scientific notation. The unit is s.

Example

```
:CURSOR:TRACK:CAX 1.000000E-8 /*Sets the horizontal position of
Cursor A to 10 ns.*/
:CURSOR:TRACK:CAX? /*The query returns 1.000000E-8.*/
```

3.11.19 :CURSOR:TRACK:CBX**Syntax**

```
:CURSOR:TRACK:CBX <bx>
```

```
:CURSOR:TRACK:CBX?
```

Description

Sets or queries the horizontal position of Cursor B in the track mode of cursor measurement.

Parameter

| Name | Type | Range | Default |
|------|------|-------------------------|---------|
| <bx> | Real | Refer to <i>Remarks</i> | - |

Remarks

The range of the horizontal position of Cursor B is determined by the current horizontal scale and position.

Return Format

The query returns the horizontal position of Cursor B in scientific notation. The unit is s.

Example

```
:CURSOR:TRACK:CBX 1.000000E-8 /*Sets the horizontal position of
Cursor B 10 ns.*/
:CURSOR:TRACK:CBX? /*The query returns 1.000000E-8.*/
```

3.11.20 :CURSOR:TRACK:CAY**Syntax**

```
:CURSOR:TRACK:CAY <ay>
```

```
:CURSOR:TRACK:CAY?
```

Description

Sets or queries the vertical position of Cursor A in the track mode of cursor measurement.

Parameter

| Name | Type | Range | Default |
|------|------|-------------------------|---------|
| <ay> | Real | Refer to <i>Remarks</i> | - |

Remarks

The range of the vertical position of Cursor A is determined by the current vertical scale and position.

Return Format

The query returns the vertical position of Cursor A in scientific notation. The unit is V.

Example

```
:CURSOR:TRACK:CAY 0.1 /*Sets the vertical position of Cursor A
to 0.1 V.*/
:CURSOR:TRACK:CAY? /*The query returns 1.000000E-1.*/
```

3.11.21 :CURSOR:TRACK:CBY**Syntax**

```
:CURSOR:TRACK:CBY <by>
```

```
:CURSOR:TRACK:CBY?
```

Description

Sets or queries the vertical position of Cursor B in the track mode of cursor measurement.

Parameter

| Name | Type | Range | Default |
|------|------|-------------------------|---------|
| <by> | Real | Refer to <i>Remarks</i> | - |

Remarks

The range of the vertical position of Cursor B is determined by the current vertical scale and position.

Return Format

The query returns the vertical position of Cursor B in scientific notation. The unit is V.

Example

```
:CURSor:TRACk:CBY 0.1 /*Sets the vertical position of Cursor B
to 0.1 V.*/
:CURSor:TRACk:CBY? /*The query returns 1.000000E-1.*/
```

3.11.22 :CURSor:TRACk:AXValue?**Syntax**

```
:CURSor:TRACk:AXValue?
```

Description

Queries the X value at Cursor A in the track mode of cursor measurement. The unit is determined by the amplitude unit selected for the currently corresponding channel.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the X value at Cursor A in scientific notation.

Example

N/A

3.11.23 :CURSor:TRACk:AYValue?**Syntax**

```
:CURSor:TRACk:AYValue?
```

Description

Queries the Y value at Cursor A in the track mode of cursor measurement. The unit is the same as that selected for the current channel.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the Y value at Cursor A in scientific notation.

Example

N/A

3.11.24 :CURSOR:TRACK:BXValue?

Syntax

```
:CURSOR:TRACK:BXValue?
```

Description

Queries the X value at Cursor B in the track mode of cursor measurement. The unit is determined by the amplitude unit selected for the currently corresponding channel.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the X value at Cursor B in scientific notation.

Example

N/A

3.11.25 :CURSOR:TRACK:BYValue?

Syntax

```
:CURSOR:TRACK:BYValue?
```

Description

Queries the Y value at Cursor B in the track mode of cursor measurement. The unit is the same as that selected for the current channel.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the Y value at Cursor B in scientific notation.

Example

N/A

3.11.26 :CURSOR:TRACK:XDELta?

Syntax

```
:CURSOR:TRACK:XDELta?
```

Description

Queries the difference (ΔX) between the X value at Cursor A and the X value at Cursor B in the track mode of cursor measurement.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the current difference in scientific notation.

Example

N/A

3.11.27 :CURSOR:TRACK:YDELta?

Syntax

```
:CURSOR:TRACK:YDELta?
```

Description

Queries the difference (ΔY) between the Y value at Cursor A and the Y value at Cursor B in the track mode of cursor measurement. The unit is the same as that selected for the current channel.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the current difference in scientific notation.

Example

N/A

3.11.28 :CURSOR:TRACK:IXDelta?

Syntax

:CURSOR:TRACK:IXDelta?

Description

Queries the reciprocal ($1/\Delta X$) of the absolute difference between the X value at Cursor A and the X value at Cursor B in the track mode of cursor measurement. The default unit is Hz.

Parameter

N/A

Remarks

N/A

Return Format

The query returns $1/\Delta X$ in scientific notation.

Example

N/A

3.11.29 :CURSOR:TRACK:MODE

Syntax

:CURSOR:TRACK:MODE <mode>

:CURSOR:TRACK:MODE?

Description

Sets or queries the axis in the track mode of cursor measurement.

Parameter

| Name | Type | Range | Default |
|--------|----------|-------|---------|
| <mode> | Discrete | {Y X} | - |

Remarks

N/A

Return Format

The query returns Y or X.

Example

```
:CURSOR:TRACK:MODE X /*Sets the axis in the track mode of cursor
measurement to X-axis.*/
:CURSOR:TRACK:MODE? /*The query returns X.*/
```

3.11.30 :CURSOR:XY:AX

Syntax

:CURSOR:XY:AX <X>

:CURSOR:XY:AX?

Description

Sets or queries the horizontal position of Cursor A in the XY cursor measurement mode.

Parameter

| Name | Type | Range | Default |
|------|------|--|---------|
| <x> | Real | Related to the current vertical scale and position | - |

Remarks

N/A

Return Format

The query returns the horizontal position of Cursor A in scientific notation.

Example

```
:CURSOR:XY:AX 0.1 /*Sets the horizontal position of Cursor A to
100 mV.*/
:CURSOR:XY:AX? /*The query returns 1.000000E-1.*/
```

3.11.31 :CURSOR:XY:BX**Syntax**

```
:CURSOR:XY:BX <x>
```

```
:CURSOR:XY:BX?
```

Description

Sets or queries the horizontal position of Cursor B in the XY cursor measurement mode.

Parameter

| Name | Type | Range | Default |
|------|------|--|---------|
| <x> | Real | Related to the current vertical scale and position | - |

Remarks

N/A

Return Format

The query returns the horizontal position of Cursor B in scientific notation.

Example

```
:CURSOR:XY:BX 0.1 /*Sets the horizontal position of Cursor B to
100 mV.*/
:CURSOR:XY:BX? /*The query returns 1.000000E-1.*/
```

3.11.32 :CURSOR:XY:AY**Syntax**

```
:CURSOR:XY:AY <y>
```

```
:CURSOR:XY:AY?
```

Description

Sets or queries the vertical position of Cursor A in the XY cursor measurement mode.

Parameter

| Name | Type | Range | Default |
|------|------|--|---------|
| <y> | Real | Related to the current vertical scale and position | - |

Remarks

N/A

Return Format

The query returns the vertical position of Cursor A in scientific notation.

Example

```
:CURSor:XY:AY 0.1 /*Sets the vertical position of Cursor A to
100 mV.*
:CURSor:XY:AY? /*The query returns 1.000000E-1.*
```

3.11.33 :CURSor:XY:BY**Syntax**

```
:CURSor:XY:BY <y>
```

```
:CURSor:XY:BY?
```

Description

Sets or queries the vertical position of Cursor B in the XY cursor measurement mode.

Parameter

| Name | Type | Range | Default |
|------|------|--|---------|
| <y> | Real | Related to the current vertical scale and position | - |

Remarks

N/A

Return Format

The query returns the vertical position of Cursor B in scientific notation.

Example

```
:CURSor:XY:BY 0.1 /*Sets the vertical position of Cursor B to
100 mV.*
:CURSor:XY:BY? /*The query returns 1.000000E-1.*
```

3.11.34 :CURSOR:XY:AXValue?

Syntax

:CURSOR:XY:AXValue?

Description

Queries the X value at Cursor A in the XY cursor measurement mode.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the X value at Cursor A in scientific notation.

Example

N/A

3.11.35 :CURSOR:XY:AYValue?

Syntax

:CURSOR:XY:AYValue?

Description

Queries the X value at Cursor A in the XY cursor measurement mode.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the Y value at Cursor A in scientific notation.

Example

N/A

3.11.36 :CURSOR:XY:BXValue?

Syntax

:CURSOR:XY:BXValue?

Description

Queries the X value at Cursor B in the XY cursor measurement mode.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the X value at Cursor B in scientific notation.

Example

N/A

3.11.37 :CURSor:XY:BYValue?

Syntax

:CURSor:XY:BYValue?

Description

Queries the Y value at Cursor B in the XY cursor measurement mode.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the Y value at Cursor B in scientific notation.

Example

N/A

3.11.38 :CURSor:XY:XDELta?

Syntax

:CURSor:XY:XDELta?

Description

Queries the difference (ΔX) between the X value at Cursor A and the X value at Cursor B in the XY cursor measurement.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the current difference in scientific notation.

Example

N/A

3.11.39 :CURSor:XY:YDELta?

Syntax`:CURSor:XY:YDELta?`**Description**

Queries the difference (ΔY) between the Y value at Cursor A and the Y value at Cursor B in the XY cursor measurement. The unit is the same as that selected for the current channel.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the current difference in scientific notation.

Example

N/A

3.12 :DISPlay Commands

The **:DISPlay** commands can be used to set the displayed type of the waveform, persistence time, intensity, grid type, grid brightness, etc.

3.12.1 :DISPlay:CLEar

Syntax`:DISPlay:CLEar`


Description

Clears all the waveforms on the screen.

Parameter

N/A

Remarks

- If the oscilloscope is in the "RUN" state, new waveforms will continue being displayed after being cleared.
- You can also send the *:CLEAr* command to clear all the waveforms on the screen.
- This command functions the same as the front-panel key  .

Return Format

N/A

Example

N/A

3.12.2 :DISPlay:TYPE

Syntax

```
:DISPlay:TYPE <type>
```

```
:DISPlay:TYPE?
```

Description

Sets or queries the display type of the waveforms on the screen.

Parameter

| Name | Type | Range | Default |
|--------|----------|-----------|---------|
| <type> | Discrete | {VECTors} | VECTors |

Remarks

VECTors: The sample points are connected by lines and displayed. In most cases, this mode can provide the most vivid waveform for you to view the steep edge of the waveform (such as square waveform).

Return Format

The query returns VECT.

Example

```
:DISPlay:TYPE VECTors /*Sets the display type to VECTors.*/  
:DISPlay:TYPE? /*The query returns VECT.*/
```

3.12.3 :DISPlay:GRADing:TIME**Syntax**

```
:DISPlay:GRADing:TIME <time>
```

```
:DISPlay:GRADing:TIME?
```

Description

Sets or queries the persistence time. The default unit is s.

Parameter

| Name | Type | Range | Default |
|--------|----------|-------------------------------------|---------|
| <time> | Discrete | {MIN 0.1 0.2 0.5 1 2 5 10 INFinite} | MIN |

Remarks

- **MIN:** sets the persistence time to its minimum value to view how the waveform changes at a high refresh rate.
- **specified value (e.g. 0.1, 0.2, 0.5, 1, 2, 5, 10):** sets the persistence time to any of the above specific value to observe glitches that change relatively slowly or glitches with low occurrence probability.
- **INFinite:** In this mode, the oscilloscope displays the waveform newly acquired without clearing the waveforms acquired formerly. It can be used to measure noise and jitter and to capture incidental events.

Return Format

The query returns MIN, 0.1, 0.2, 0.5, 1, 2, 5, 10, or INF.

Example

```
:DISPlay:GRADing:TIME 0.1 /*Sets the persistence time to 100 ms.*/  
:DISPlay:GRADing:TIME? /*The query returns 0.1.*/
```

3.12.4 :DISPlay:WBRightness

Syntax

```
:DISPlay:WBRightness <brightness>
```

```
:DISPlay:WBRightness?
```

Description

Sets or queries the brightness of the waveform on the screen, expressed in percentage.

Parameter

| Name | Type | Range | Default |
|--------------|---------|----------|---------|
| <brightness> | Integer | 1 to 100 | 50 |

Remarks

N/A

Return Format

The query returns an integer ranging from 1 to 100.

Example

```
:DISPlay:WBRightness 50 /*Sets the waveform brightness to 50%.*/  
:DISPlay:WBRightness? /*The query returns 50.*/
```

3.12.5 :DISPlay:GRID

Syntax

```
:DISPlay:GRID <grid>
```

```
:DISPlay:GRID?
```

Description

Sets or queries the display type of the screen grid.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <grid> | Discrete | {FULL HALF NONE} | FULL |

Remarks

- **FULL:** turns the background grid and coordinates on.

- **HALF:** turns the background grid off and turns the coordinate on.
- **NONE:** turns the background grid and coordinate off.

Return Format

The query returns FULL, HALF, or NONE.

Example

```
:DISPlay:GRID NONE /*Turns the background grid and coordinates
off.*/
:DISPlay:GRID? /*The query returns NONE.*/
```

3.12.6 :DISPlay:GBrightness

Syntax

```
:DISPlay:GBrightness <brightness>
```

```
:DISPlay:GBrightness?
```

Description

Sets or queries the brightness of the screen grid, expressed in percentage.

Parameter

| Name | Type | Range | Default |
|--------------|---------|----------|---------|
| <brightness> | Integer | 0 to 100 | 50 |

Remarks

N/A

Return Format

The query returns an integer ranging from 0 to 100.

Example

```
:DISPlay:GBrightness 60 /*Sets the screen grid brightness to
60%.*/
:DISPlay:GBrightness? /*The query returns 60.*/
```

3.12.7 :DISPlay:DATA?

Syntax

```
:DISPlay:DATA? [<type>]
```

Description

Queries the bitmap data stream of the currently displayed image.

Parameter

| Name | Type | Range | Default |
|--------|----------|---------------|---------|
| <type> | Discrete | {BMP PNG JPG} | BMP |

Remarks

The read data format is TMC header + binary data stream of the screenshot + terminator. The TMC header is in #NXXXXXX format; wherein, # is the TMC header identifier; N following # represents the number of digits (in the decimal integer) that follow; the length of the binary data stream of the screenshot is expressed in ASCII strings, and the terminator represents the ending of communication. For example, the data read for one time is #9000387356. 9 indicates the number of digits (in the decimal integer) that follow, and "000387356" indicates the length of the binary data stream, that is, the number of bytes to be transmitted.

Return Format

The query returns the binary data stream of the screenshot in specified format.

Example

N/A

3.12.8 :DISPlay:RULers

Syntax

```
:DISPlay:RULers <bool>
```

```
:DISPlay:RULers?
```

Description

Enables or disables the display of the scale ruler; or queries the on/off status of the scale ruler.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 1 ON |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:DISPlay:RULers ON      /*Enables the display of the scale ruler.*/
:DISPlay:RULers?       /*The query returns 1.*/
```

3.12.9 :DISPlay:COLor**Syntax**

```
:DISPlay:COLor <bool>
```

```
:DISPlay:COLor?
```

Description

Enables or disables the color grade display; or queries the on/off status of the color grade display.

Parameter

| Name | Type | Range | Default |
|--------|------|-------------------|---------|
| <bool> | Bool | {{1 ON}}{{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:DISPlay:COLor ON      /*Enables the color grade display.*/
:DISPlay:COLor?       /*The query returns 1.*/
```

3.12.10 :DISPlay:WHOLd**Syntax**

```
:DISPlay:WHOLd <bool>
```

```
:DISPlay:WHOLd?
```

Description

Sets to enable or disable the waveform freezing function; queries whether to enable or disable the waveform freezing function.

Parameter

| Name | Type | Range | Default |
|--------|------|-------------------|---------|
| <bool> | Bool | {{1 ON}}{{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:DISPlay:WHOLd ON /*Enables the Waveform Freeze.*/  
:DISPlay:WHOLd? /*The query returns 1.*/
```

3.13 :DVM Commands

3.13.1 :DVM:CURRent?

Syntax

```
:DVM:CURRent?
```

Description

Queries the current voltage value under test.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

```
N/A
```

3.13.2 :DVM:ENABLE

Syntax

```
:DVM:ENABLE <bool>
```

```
:DVM:ENABLE?
```

Description

Enables or disables the digital voltmeter; or queries the on/off status of the digital voltmeter.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:DVM:ENABle ON      /*Enables the digital voltmeter.*/
:DVM:ENABle?       /*The query returns 1.*/
```

3.13.3 :DVM:SOURce**Syntax**

```
:DVM:SOURce <source>
```

```
:DVM:SOURce?
```

Description

Sets or queries the source of the digital voltmeter.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:DVM:SOURce CHANnel1 /*Sets the source of DVM to CHANnel1.*/
:DVM:SOURce?         /*The query returns CHAN1.*/
```

3.13.4 :DVM:MODE

Syntax

```
:DVM:MODE <mode>
```

```
:DVM:MODE?
```

Description

Sets or queries the mode of digital voltmeter.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <mode> | Discrete | {ACRMs DC DCRMs} | ACRMs |

Remarks

- **ACRMs:** displays the root-mean-square value of the acquired data, with the DC component removed.
- **DC:** displays the root- average-square value of the acquired data.
- **DCRMs:** displays the root-mean-square value of the acquired data.

Return Format

The query returns ACRM, DC, or DCRM.

Example

```
:DVM:MODE DC /*Sets the mode of the digital voltmeter to DC.*/
:DVM:MODE? /*The query returns DC.*/
```

3.14 IEEE488.2 Common Commands

3.14.1 *IDN?

Syntax

```
*IDN?
```

Description

Queries the ID string of the instrument.

Parameter

N/A

Remarks

N/A

Return Format

The query returns RIGOL TECHNOLOGIES,<model>,<serial number>,<software version>.

- **<model>**: indicates the model number of the instrument.
- **<serial number>**: indicates the serial number of the instrument.
- **<software version>**: indicates the software version of the instrument.

Example

N/A

3.14.2 *RST

Syntax***RST****Description**

Restores the instrument to its factory default settings.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.14.3 *CLS

Syntax***CLS****Description**

Clears all the event registers, and also clears the error queue.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.14.4 *ESE**Syntax*****ESE** <maskargument>***ESE?****Description**

Sets or queries the enable register of the standard event register set.

Parameter

| Name | Type | Range | Default |
|----------------|---------|----------|---------|
| <maskargument> | Integer | 0 to 255 | 0 |

Remarks

N/A

Return Format

The query returns an integer. The integer equals to the binary-weighted sum of all the bits set in the register.

Example

```
*ESE 16 /*Enables Bit 4 (16 in decimal) in the register.*/
*ESE? /*The query returns the enable value of the register
16.*/
```

3.14.5 *ESR?**Syntax*****ESR?**

Description

Queries and clears the event register of the standard event status register.

Parameter

N/A

Remarks

Bit 1 and Bit 6 in the standard event status register are not used and are always treated as 0; therefore, the range of the returned value is a decimal number corresponding to a binary number X0XXXX0X (X is 1 or 0).

Return Format

The query returns an integer. The integer equals to the binary-weighted sum of all the bits set in the register.

Example

N/A

3.14.6 *OPC

Syntax

*OPC

*OPC?

Description

The *OPC command sets bit 0 (Operation Complete, OPC) in the standard event status register to 1 after the current operation is finished. The OPC? command queries whether the current operation is finished.

Parameter

N/A

Remarks

N/A

Return Format

The query returns 1 after the current operation is finished; otherwise, the query returns 0.

Example

N/A

3.14.7 *RCL

Syntax

*RCL

Description

Recalls instrument settings from the specified non-volatile memory. The previous saved settings through the *SAV command will be overwritten.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.14.8 *SAV

Syntax

*SAV <value>

Description

Saves the current instrument state to the selected register.

Parameter

| Name | Type | Range | Default |
|---------|---------|---------|---------|
| <value> | Integer | 0 to 49 | 0 |

Remarks

N/A

Return Format

N/A

Example

```
*SAV 1 /*Saves the current instrument state to Register 1.*/
```

3.14.9 *SRE

Syntax

*SRE <maskargument>

*SRE?

Description

Sets or queries the enable register of the status byte register set.

Parameter

| Name | Type | Range | Default |
|----------------|---------|----------|---------|
| <maskargument> | Integer | 0 to 255 | 0 |

Remarks

This command queries the standard event enable register value. Bit 3 and Bit 6 in the standard event status register are not used and are always treated as 0. Therefore, the range of <maskargument> is a decimal number corresponding to a binary number X0XX0XXX (X is 1 or 0).

Return Format

The query returns an integer. The integer equals to the binary-weighted sum of all the bits set in the register.

Example

```
*SRE 16 /*Enables Bit 4 (16 in decimal) in the register.*/
*SRE? /*The query returns the enable value of the register 16.*/
```

3.14.10 *STB?

Syntax

*STB?

Description

Queries the event register for the status byte register. After executing the command, the value in the status byte register is cleared.

Parameter

N/A

Remarks

Bit 0 and Bit 1 in the status byte register are not used and are always treated as 0; therefore, the range of the returned value is a decimal number corresponding to a binary number X0XXXX0X (X is 1 or 0).

Return Format

The query returns an integer. The integer equals to the binary-weighted sum of all the bits set in the register.

Example

N/A

3.14.11 *WAI

Syntax

*WAI

Description

Waits for all the pending operations to complete before executing any additional commands.

Parameter

N/A

Remarks

This operation command does not have any functions, only to be compatible with other devices.

Return Format

N/A

Example

N/A

3.14.12 *TST?

Syntax

*TST?

Description

Performs a self-test and queries the self-test result.

Parameter

N/A

Remarks

N/A

Return Format

The query returns a decimal integer.

Example

N/A

3.15 :LAN Commands

The :LAN commands are used to set and query the LAN parameters.

3.15.1 :LAN:DHCP

Syntax

:LAN:DHCP <bool>

:LAN:DHCP?

Description

Turns on or off the DHCP configuration mode; or queries the on/off status of the current DHCP configuration mode.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 1 ON |

Remarks

- When the three IP configuration types (DHCP, Auto IP, and Static IP) are all turned on, the priority of the parameter configuration from high to low is "DHCP", "Auto IP", and "Static IP". The three IP configuration types cannot be all turned off at the same time.
- When DHCP is valid, the DHCP server in the current network will assign the network parameters (such as the IP address) for the oscilloscope.
- After the :LAN:APPLY command is executed, the configuration type can take effect immediately.

Return Format

The query returns 1 or 0.

Example

```
:LAN:DHCP OFF /*Disables DHCP configuration mode.*/
:LAN:DHCP? /*The query returns 0.*/
```

3.15.2 :LAN:AUTOip**Syntax**

```
:LAN:AUTOip <bool>
```

```
:LAN:AUTOip?
```

Description

Turns on or off the Auto IP configuration mode; or queries the on/off status of the current Auto IP configuration mode.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 1 ON |

Remarks

When the auto IP mode is valid, disable DHCP manually. You can self-define the gateway and DNS address for the oscilloscope.

Return Format

The query returns 1 or 0.

Example

```
:LAN:AUTOip OFF /*Disables the Auto IP configuration
mode.*/
:LAN:AUTOip? /*The query returns 0.*/
```

3.15.3 :LAN:GATeway**Syntax**

```
:LAN:GATeway <string>
```

```
:LAN:GATeway?
```

Description

Sets or queries the default gateway.

Parameter

| Name | Type | Range | Default |
|----------|--------------|-------------------------|---------|
| <string> | ASCII String | Refer to <i>Remarks</i> | - |

Remarks

- The format of <string> is nnn.nnn.nnn.nnn. The range of the first section of "nnn" is from 0 to 223 (except 127), and the ranges of the other three sections of "nnn" are from 0 to 255.
- When you use this command, the IP configuration mode should be Auto IP or Static IP mode.

Return Format

The query returns the current gateway in strings.

Example

```
:LAN:GATeway 192.168.1.1 /*Sets the default gateway to
192.168.1.1.*/
:LAN:GATeway? /*The query returns 192.168.1.1.*/
```

3.15.4 :LAN:DNS**Syntax**

:LAN:DNS <string>

:LAN:DNS?

Description

Sets or queries the DNS address.

Parameter

| Name | Type | Range | Default |
|----------|--------------|-------------------------|---------|
| <string> | ASCII String | Refer to <i>Remarks</i> | - |

Remarks

- The format of <string> is nnn.nnn.nnn.nnn. The range of the first section of "nnn" is from 0 to 223 (except 127), and the ranges of the other three sections of "nnn" are from 0 to 255.

- When you use this command, the IP configuration mode should be Auto IP or Static IP mode.

Return Format

The query returns the current DNS address in strings.

Example

```
:LAN:DNS 192.168.1.1 /*Sets the DNS address to  
192.168.1.1.*/  
:LAN:DNS? /*The query returns 192.168.1.1.*/
```

3.15.5 :LAN:MAC?

Syntax

```
:LAN:MAC?
```

Description

Queries the MAC address of the instrument.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the MAC address in strings. For example, 00:19:AF:00:11:22.

Example

```
N/A
```

3.15.6 :LAN:DSERver?

Syntax

```
:LAN:DSERver?
```

Description

Queries the address of the DHCP server.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the address of the DHCP server in strings.

Example

N/A

3.15.7 :LAN:MANual**Syntax**

```
:LAN:MANual <bool>
```

```
:LAN:MANual?
```

Description

Turns on or off the static IP configuration mode; or queries the on/off status of the static IP configuration mode.

Parameter

| Name | Type | Range | Default |
|--------|------|--------------------|---------|
| <bool> | Bool | {{1 ON}} {{0 OFF}} | 0 OFF |

Remarks

When the static IP mode is valid, disable DHCP and Auto IP manually. You can self-define the network parameters of the oscilloscope, such as IP address, subnet mask, gateway, and DNS address. For the setting of the IP address, refer to the [:LAN:IPADdress](#) command. For the setting of the subnet mask, refer to the [:LAN:SMASk](#) command. For the setting of the gateway, refer to the [:LAN:GATeway](#) command. For the setting of DNS, refer to the [:LAN:DNS](#) command.

Return Format

The query returns 1 or 0.

Example

```
:LAN:MANual ON /*Enables the static IP configuration mode.*/
:LAN:MANual? /*The query returns 1.*/
```

3.15.8 :LAN:IPADdress**Syntax**

```
:LAN:IPADdress <string>
```

```
:LAN:IPADdress?
```

Description

Sets or queries the IP address of the instrument.

Parameter

| Name | Type | Range | Default |
|----------|--------------|-------------------------|---------|
| <string> | ASCII String | Refer to <i>Remarks</i> | - |

Remarks

- The format of <string> is nnn.nnn.nnn.nnn. The range of the first section of "nnn" is from 0 to 223 (except 127), and the ranges of the other three sections of "nnn" are from 0 to 255.
- When you use the command, the IP configuration mode should be static IP. Besides, the DHCP and auto IP should be disabled.

Return Format

The query returns the current IP address in strings.

Example

```
:LAN:IPADdress 192.168.1.10 /*Sets the IP address to
192.168.1.10.*/
:LAN:IPADdress? /*The query returns 192.168.1.10.*/
```

3.15.9 :LAN:SMASk**Syntax**

:LAN:SMASk <string>

:LAN:SMASk?

Description

Sets or queries the subnet mask.

Parameter

| Name | Type | Range | Default |
|----------|--------------|-------------------------|---------|
| <string> | ASCII String | Refer to <i>Remarks</i> | - |

Remarks

- The format of <string> is nnn.nnn.nnn.nnn. The range of the section "nnn" is from 0 to 255.
- When you use the command, the IP configuration mode should be static IP. Besides, the DHCP and auto IP should be disabled.

Return Format

The query returns the current subnet mask in strings.

Example

```
:LAN:SMASK 255.255.255.0 /*Sets the subnet mask to  
255.255.255.0.*/  
:LAN:SMASK? /*The query returns 255.255.255.0.*/
```

3.15.10 :LAN:STATus?

Syntax

:LAN:STATus?

Description

Queries the current network configuration status.

Parameter

N/A

Remarks

- **UNLINK:** not connected.
- **CONNECTED:** the network is successfully connected.
- **INIT:** the instrument is acquiring an IP address.
- **IPCONFLICT:** there is an IP address conflict.
- **BUSY:** please wait...
- **CONFIGURED:** the network configuration has been successfully configured.
- **DHCPFAILED:** the DHCP configuration has failed.
- **INVALIDIP:** invalid IP.
- **IPLOSE:** IP lost.

Return Format

The query returns UNLINK, CONNECTED, INIT, IPCONFLICT, BUSY, CONFIGURED, DHCPFAILED, INVALIDIP, or IPLOSE.

Example

N/A

3.15.11 :LAN:VISA?**Syntax**

:LAN:VISA? [*<type>*]

Description

Queries the VISA address of the instrument.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <type> | Discrete | {USB LXI SOCKET} | - |

Remarks

This command contains a parameter "type" and it is used to set or query the address type. By default, it returns the LXI address.

Return Format

The query returns the VISA address in strings.

Example

N/A

3.15.12 :LAN:MDNS**Syntax**

:LAN:MDNS *<bool>*

:LAN:MDNS?

Description

Enables or disables mDNS; or queries the mDNS status.

Parameter

| Name | Type | Range | Default |
|--------|------|--------------------|---------|
| <bool> | Bool | {{1 ON}} {{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:LAN:MDNS ON /*Enables mDNS.*/
:LAN:MDNS? /*The query returns 1.*/
```

3.15.13 :LAN:HOST:NAME**Syntax**

```
:LAN:HOST:NAME <name>
```

```
:LAN:HOST:NAME?
```

Description

Sets or queries the host name.

Parameter

| Name | Type | Range | Default |
|--------|--------------|---|---------|
| <name> | ASCII String | The label can contain English letters and numbers, as well as some symbols. | - |

Remarks

N/A

Return Format

The query returns the host name in ASCII strings.

Example

N/A

3.15.14 :LAN:DESCRiption

Syntax

:LAN:DESCRiption <name>

:LAN:DESCRiption?

Description

Sets or queries the description.

Parameter

| Name | Type | Range | Default |
|--------|--------------|---|---------|
| <name> | ASCII String | The label can contain English letters and numbers, as well as some symbols. | - |

Remarks

N/A

Return Format

The query returns the description in ASCII strings.

Example

N/A

3.15.15 :LAN:APPLY

Syntax

:LAN:APPLY

Description

Applies the network configuration.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.16 :MASK Commands

The :MASK commands are used to set or query the pass/fail test related parameters.

3.16.1 :MASK:ENABLE

Syntax

```
:MASK:ENABLE <bool>
```

```
:MASK:ENABLE?
```

Description

Enables or disables the pass/fail test function; or queries the on/off status of the pass/fail test function.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

The pass/fail test function is invalid in the following conditions: when the horizontal time base mode is ROLL; when in the delayed sweep mode (Zoom).

Return Format

The query returns 1 or 0.

Example

```
:MASK:ENABLE ON /*Enables the pass/fail test function.*/
:MASK:ENABLE? /*The query returns 1.*/
```

3.16.2 :MASK:SOURce

Syntax

```
:MASK:SOURce <source>
```

```
:MASK:SOURce?
```

Description

Sets or queries the source of the pass/fail test.

Parameter

| Name | Type | Range | Default |
|----------|----------|---------------------------------------|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

When you use the command to set the disabled channel, the disabled channel will be enabled automatically.

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:MASK:SOURce CHANnel2 /*Sets the source of the pass/fail test to
CHANnel2.*/
:MASK:SOURce? /*The query returns CHAN2.*/
```

3.16.3 :MASK:OPERate**Syntax**

```
:MASK:OPERate <oper>
```

```
:MASK:OPERate?
```

Description

Starts or stops the pass/fail test; or queries the operating status of the pass/fail test.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------|---------|
| <oper> | Discrete | {RUN STOP} | STOP |

Remarks

Before running this command, send the *:MASK:ENABLE* command to enable the pass/fail test function.

Return Format

The query returns RUN or STOP.

Example

```
:MASK:OPERate RUN /*Starts the pass/fail test.*/
:MASK:OPERate? /*The query returns RUN.*/
```


3.16.4 :MASK:X

Syntax

```
:MASK:X <X>
```

```
:MASK:X?
```

Description

Sets or queries the horizontal adjustment parameter of the pass/fail test mask. The default unit is div.

Parameter

| Name | Type | Range | Default |
|------|------|-------------------|----------|
| <x> | Real | 0.01 div to 2 div | 0.24 div |

Remarks

N/A

Return Format

The query returns the current horizontal adjustment parameter in scientific notation.

Example

```
:MASK:X 0.28 /*Sets the horizontal adjustment parameter to 0.28
div.*/
:MASK:X? /*The query returns 2.800000E-1.*/
```

3.16.5 :MASK:Y

Syntax

```
:MASK:Y <y>
```

```
:MASK:Y?
```

Description

Sets or queries the vertical adjustment parameter of the pass/fail test mask. The default unit is div.

Parameter

| Name | Type | Range | Default |
|------|------|-------------------|----------|
| <y> | Real | 0.04 div to 2 div | 0.48 div |

Remarks

N/A

Return Format

The query returns the current vertical adjustment parameter in scientific notation.

Example

```
:MASK:Y 0.36 /*Sets the vertical adjustment parameter to 0.36
div.*/
:MASK:Y? /*The query returns 3.600000E-1.*/
```

3.16.6 :MASK:CREate

Syntax

:MASK:CREate

Description

Creates the pass/fail test mask with the currently set horizontal and vertical adjustment parameters.

Parameter

N/A

Remarks

This command is only valid when the pass/fail test function is enabled and not in the running state.

Return Format

N/A

Example

```
N/A
```

3.16.7 :MASK:RESet

Syntax

:MASK:RESet

Description

Resets the number of frames that passed and failed the pass/fail test, as well as the total number of frames.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.16.8 :MASK:FAILED?

Syntax`:MASK:FAILED?`**Description**

Queries the total number of failed frames in the pass/fail test results.

Parameter

N/A

Remarks

N/A

Return Format

The query returns an integer.

Example

N/A

3.16.9 :MASK:PASSED?

Syntax`:MASK:PASSED?`**Description**

Queries the total number of passed frames in the pass/fail test results.

Parameter

N/A

Remarks

N/A

Return Format

The query returns an integer.

Example

N/A

3.16.10 :MASK:TOTal?**Syntax**`:MASK:TOTal?`**Description**

Queries the total number of frames in the pass/fail test results.

Parameter

N/A

Remarks

N/A

Return Format

The query returns an integer.

Example

N/A

3.16.11 :MASK:OUTPut:ENABLE**Syntax**`:MASK:OUTPut:ENABLE <bool>``:MASK:OUTPut:ENABLE?`**Description**

Sets or queries the output on/off status of the rear-panel AUX OUT connector.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

- If enabled, in the Utility menu, "AUX Out" is automatically set to "PassFail". When a successful or failed event is detected, a pulse will be output from the [AUX OUT] connector.

- If disabled, in the Utility menu, "AUX Out" is automatically set to "TrigOut". The output of the [AUX OUT] connector is irrelevant with the pass/fail test.

Return Format

The query returns 0 or 1.

Example

```
:MASK:OUTPut:ENABle ON /*Enables the Aux output.*/
:MASK:OUTPut:ENABle? /*The query returns 1.*/
```

3.16.12 :MASK:OUTPut:EVENT

Syntax

```
:MASK:OUTPut:EVENT <item>
```

```
:MASK:OUTPut:EVENT?
```

Description

Sets or queries the output event.

Parameter

| Name | Type | Range | Default |
|--------|----------|-------------|---------|
| <item> | Discrete | {FAIL PASS} | FAIL |

Remarks

N/A

Return Format

The query returns FAIL or PASS.

Example

```
:MASK:OUTPut:EVENT PASS /*Sets the output event to PASS.*/
:MASK:OUTPut:EVENT? /*The query returns PASS.*/
```

3.16.13 :MASK:OUTPut:TIME

Syntax

```
:MASK:OUTPut:TIME <time>
```

```
:MASK:OUTPut:TIME?
```

Description

Sets or queries the output pulse time.

Parameter

| Name | Type | Range | Default |
|--------|------|-----------------|-----------|
| <time> | Real | 100 ns to 10 ms | 1 μ s |

Remarks

N/A

Return Format

The query returns the pulse time in scientific notation.

Example

```
:MASK:OUTPut:TIME 0.000003 /*Sets the pulse time to 3  $\mu$ s.*/
:MASK:OUTPut:TIME? /*The query returns 3.000000E-6.*/
```

3.17 :MATH<n> Commands

The :MATH<n> commands are used to set various math operation functions of the channel waveforms.

3.17.1 :MATH<n>:DISPlay

Syntax

```
:MATH<n>:DISPlay <bool>
```

```
:MATH<n>:DISPlay?
```

Description

Enables or disables the math operation function; or queries the on/off status of the math operation function.

Parameter

| Name | Type | Range | Default |
|--------|----------|-----------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bool> | Bool | {{1 ON}}{0 OFF} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:MATH1:DISPlay ON /*Enables the math operation of Math1.*/
:MATH1:DISPlay? /*The query returns 1.*/
```

3.17.2 :MATH<n>:OPERator**Syntax**

```
:MATH<n>:OPERator <opt>
```

```
:MATH<n>:OPERator?
```

Description

Sets or queries the operator of math operation.

Parameter

| Name | Type | Range | Default |
|-------|----------|--|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <opt> | Discrete | {ADD SUBTract MULTiply DIVision AND OR XOR NOT FFT INTG DIFF SQRT LG LN EXP ABS LPASs HPASs BPASs BSTop AXB} | ADD |

Remarks

N/A

Return Format

The query returns ADD, SUBT, MULT, DIV, AND, OR, XOR, NOT, FFT, INTG, DIFF, SQRT, LG, LN, EXP, ABS, LPAS, HPAS, BPAS, BST, or AXB.

Example

```
:MATH1:OPERator INTG /*Sets the math operator of Math1 to
Integrate.*/
:MATH1:OPERator? /*The query returns INTG.*/
```

3.17.3 :MATH<n>:SOURce1**Syntax**

```
:MATH<n>:SOURce1 <source>
```

```
:MATH<n>:SOURce1?
```

Description

Sets or queries the source or Source A of arithmetic operation/function operation/filter operation.

Parameter

| Name | Type | Range | Default |
|----------|----------|-------------------------|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | Refer to <i>Remarks</i> | CHANnel1 |

Remarks

- For arithmetic operation, this command is used to set Source A.
- For function operation and filter operation, only use this command to set the source.
- When n=1, the range of the parameter <source> is {CHANnel1|CHANnel2|CHANnel3|CHANnel4|REF1|REF2|REF3|REF4|REF5|REF6|REF7|REF8|REF9|REF10}.

When n=2, the range of the parameter <source> is {CHANnel1|CHANnel2|CHANnel3|CHANnel4|REF1|REF2|REF3|REF4|REF5|REF6|REF7|REF8|REF9|REF10|MATH1}.

When n=3, the range of the parameter <source> is {CHANnel1|CHANnel2|CHANnel3|CHANnel4|REF1|REF2|REF3|REF4|REF5|REF6|REF7|REF8|REF9|REF10|MATH1|MATH2}.

When n=4, the range of the parameter <source> is {CHANnel1|CHANnel2|CHANnel3|CHANnel4|REF1|REF2|REF3|REF4|REF5|REF6|REF7|REF8|REF9|REF10|MATH1|MATH2|MATH3}.

Return Format

The query returns MATH1, MATH2, MATH3, CHAN1, CHAN2, CHAN3, CHAN4, REF1, REF2, REF3, REF4, REF5, REF6, REF7, REF8, REF9, or REF10.

Example

```
:MATH1:SOURce1 CHANnel3 /*Sets Source A of the arithmetic
operation to CHANnel3.*/
:MATH1:SOURce1? /*The query returns CHAN3.*/
```

3.17.4 :MATH<n>:SOURce2**Syntax**

```
:MATH<n> : SOURce2 <source>
```

```
:MATH<n> : SOURce2?
```

Description

Sets or queries Source B of arithmetic operation.

Parameter

| Name | Type | Range | Default |
|----------|----------|-------------------------|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | Refer to <i>Remarks</i> | CHANnel1 |

Remarks

This command is only available for arithmetic operation (containing two sources).

When n=1, the range of the parameter <source> is {CHANnel1|CHANnel2|CHANnel3|CHANnel4|REF1|REF2|REF3|REF4|REF5|REF6|REF7|REF8|REF9|REF10}.

When n=2, the range of the parameter <source> is {CHANnel1|CHANnel2|CHANnel3|CHANnel4|REF1|REF2|REF3|REF4|REF5|REF6|REF7|REF8|REF9|REF10|MATH1}.

When n=3, the range of the parameter <source> is {CHANnel1|CHANnel2|CHANnel3|CHANnel4|REF1|REF2|REF3|REF4|REF5|REF6|REF7|REF8|REF9|REF10|MATH1|MATH2}.

When n=4, the range of the parameter <source> is {CHANnel1|CHANnel2|CHANnel3|CHANnel4|REF1|REF2|REF3|REF4|REF5|REF6|REF7|REF8|REF9|REF10|MATH1|MATH2|MATH3}.

Return Format

The query returns MATH1, MATH2, MATH3, CHAN1, CHAN2, CHAN3, CHAN4, REF1, REF2, REF3, REF4, REF5, REF6, REF7, REF8, REF9, or REF10.

Example

```
:MATH1:SOURce2 CHANnel3 /*Sets Source B of the arithmetic
operation to CHANnel3.*/
:MATH1:SOURce2? /*The query returns CHAN3.*/
```

3.17.5 :MATH<n>:LSOURCE1**Syntax**

```
:MATH<n>:LSOURCE1 <source>
```

```
:MATH<n>:LSOURCE1?
```

Description

Sets or queries Source A of the logic operation.

Parameter

| Name | Type | Range | Default |
|------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

The logic operations include A&&B, A||B, A^B, and !A.

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:MATH1:LSOURCE1 CHANnel3 /*Sets Source A of the logic operation
to CHANnel3.*/
:MATH1:LSOURCE1? /*The query returns CHAN3.*/
```

3.17.6 :MATH<n>:LSOURCE2

Syntax

```
:MATH<n>:LSOURCE2 <source>
```

```
:MATH<n>:LSOURCE2?
```

Description

Sets or queries Source B of the logic operation.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

- The logic operations include A&&B, A||B, A^B, and !A.
- This command is only available for the logic operation that contains two sources.
It is used to set Source B.

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:MATH1:LSOURCE2 CHANnel4 /*Sets Source B of the logic operation
to CHANnel4.*/
:MATH1:LSOURCE2? /*The query returns CHAN4.*/
```

3.17.7 :MATH<n>:SCALE**Syntax**

```
:MATH<n>:SCALE <scale>
```

```
:MATH<n>:SCALE?
```

Description

Sets or queries the vertical scale of the operation results. The unit is related to the currently selected operator and the unit selected by the source.

Parameter

| Name | Type | Range | Default |
|---------|----------|-------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <scale> | Real | Refer to <i>Remarks</i> | - |

Remarks

- The setting range of the vertical scale is related to the currently selected operator and the scale of the source channel. For integration and differentiation operations, the actual range of <scale> is also related to the current horizontal time base.
- This command is invalid for logic operation and FFT operation.

Return Format

The query returns the vertical scale of the current operation results in scientific notation.

Example

```
:MATH1:SCALE 0.2 /*Sets the vertical scale to 200 mV.*/
:MATH1:SCALE? /*The query returns 2.000000E-1.*/
```

3.17.8 :MATH<n>:OFFSet**Syntax**

```
:MATH<n>:OFFSet <offset>
```

:MATH<n>:OFFSet?

Description

Sets or queries the vertical offset of the operation results. The unit is related to the currently selected operator and the unit selected by the source.

Parameter

| Name | Type | Range | Default |
|----------|----------|----------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <offset> | Real | -1 GV to +1 GV | 0.00 V |

Remarks

This command is invalid for logic operation and FFT operation.

Return Format

The query returns the vertical offset of the current operation results in scientific notation.

Example

```
:MATH1:OFFSet 8 /*Sets the vertical offset to 8 V.*/
:MATH1:OFFSet? /*The query returns 8.000000E0.*/
```

3.17.9 :MATH<n>:INVert

Syntax

:MATH<n>:INVert <bool>

:MATH<n>:INVert?

Description

Enables or disables the inverted display of the operation results; or queries the on/off status of the inverted display of the operation results.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

This command is invalid for FFT operation and logical operation.

Return Format

The query returns 1 or 0.

Example

```
:MATH1:INVert ON /*Enables the inverted display.*/
:MATH1:INVert? /*The query returns 1.*/
```

3.17.10 :MATH<n>:RESet**Syntax**

:MATH<n>:RESet

Description

After you send this command, the instrument will adjust the vertical scale of the operation results to an optimal value based on the currently selected operator and the horizontal time base of the source.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.17.11 :MATH<n>:EXPand**Syntax**

:MATH<n>:EXPand <exp>

:MATH<n>:EXPand?

Description

Sets or queries the vertical expansion type of math operation.

Parameter

| Name | Type | Range | Default |
|------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |

| Name | Type | Range | Default |
|-------|----------|--------------|---------|
| <exp> | Discrete | {GND CENTer} | GND |

Remarks

- **CENTER:** when the vertical scale is changed, the waveform will be expanded or compressed around the screen center.
- **GND:** when the vertical scale is changed, the waveform will be expanded or compressed around the signal ground level position.

Return Format

The query returns GND or CENTER.

Example

```
:MATH1:EXPand CENTer /*Sets the vertical expansion type of math
operation of CH1 to CENTER.*/
:MATH1:EXPand? /*The query returns CENTER.*/
```

3.17.12 :MATH<n>:WAVetype

Syntax

```
:MATH<n>:MATH<n>:WAVetype <type>
```

```
:MATH<n>:WAVetype?
```

Description

Sets or queries the waveform type of math operation.

Parameter

| Name | Type | Range | Default |
|--------|----------|-------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <type> | Discrete | {MAIN ZOOM} | MAIN |

Remarks

- **MAIN:** indicates the main time base region.
- **ZOOM:** indicates the zoomed time base region.

Only when the zoom function is enabled, can the Zoom waveform type be enabled. Zoom is not supported for the FFT operation.

Return Format

The query returns MAIN or ZOOM.

Example

```
:MATH1:WAVetype ZOOM /*Sets the waveform type of math operation to ZOOM.*/
:MATH1:WAVetype? /*The query returns ZOOM.*/
```

3.17.13 :MATH<n>:FFT:SOURce**Syntax**

```
:MATH<n>:FFT:SOURce <source>
```

```
:MATH<n>:FFT:SOURce?
```

Description

Sets or queries the channel source of FFT operation.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:MATH1:FFT:SOURce CHANnel3 /*Sets the channel source of FFT operation to CHANnel3.*/
:MATH1:FFT:SOURce? /*The query returns CHAN3.*/
```

3.17.14 :MATH<n>:FFT:WINDow**Syntax**

```
:MATH<n>:FFT:WINDow <window>
```

```
:MATH<n>:FFT:WINDow?
```

Description

Sets or queries the window function of FFT operation.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <window> | Discrete | {RECTangle BLACkman HANNing HAMMing FLATtop TRlangle} | HANNing |

Remarks

- The window function can effectively reduce the spectrum leakage effect.
- Different window functions are applicable to measurements of different waveforms. You need to select the window function according to the different waveforms to be measured and their characteristics.

Return Format

The query returns RECT, BLAC, HANN, HAMM, FLAT, or TRI.

Example

```
:MATH1:FFT:WINDow BLACkman /*Sets the window function of FFT
operation to Blackman-Harris.*/
:MATH1:FFT:WINDow? /*The query returns BLAC.*/
```

3.17.15 :MATH<n>:FFT:UNIT**Syntax**

```
:MATH<n>:FFT:UNIT <unit>
```

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

Description

Sets or queries the vertical unit of FFT operation results.

Parameter

| Name | Type | Range | Default |
|--------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <unit> | Discrete | {VRMS DB} | DB |

Remarks

N/A

Return Format

The query returns VRMS or DB.

Example

```
:MATH1:FFT:UNIT VRMS /*Sets the vertical unit of FFT operation
results to Vrms.*/
:MATH1:FFT:UNIT? /*The query returns VRMS.*/
```

3.17.16 :MATH<n>:FFT:SCALE**Syntax**

```
:MATH<n>:FFT:SCALE <scale>
```

```
:MATH<n>:FFT:SCALE?
```

Description

Sets or queries the vertical unit of FFT operation results.

Parameter

| Name | Type | Range | Default |
|---------|----------|-------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <scale> | Real | Refer to <i>Remarks</i> | - |

Remarks

- When the unit is set to dB, the range of <scale> is from 1 ndB to 5 GdB. The default value is 20 dB.
- When the unit is set to V_{rms} , the range of <scale> is from 1 nV_{rms} to 5 GV_{rms}. The default value is 10 V_{rms}.

You can run the `:MATH<n>:FFT:UNIT` command to configure or query the current unit.

Return Format

The query returns the current vertical scale in scientific notation.

Example

```
:MATH1:FFT:SCALE 0.3 /*Sets the vertical scale of the FFT
operation results to 300 mdB.*/
:MATH1:FFT:SCALE? /*The query returns 3.000000E-1.*/
```

3.17.17 :MATH<n>:FFT:OFFSet

Syntax

```
:MATH<n>:FFT:OFFSet <offset>
```

```
:MATH<n>:FFT:OFFSet?
```

Description

Sets or queries the vertical offset of FFT operation results.

Parameter

| Name | Type | Range | Default |
|----------|----------|-------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <offset> | Real | Refer to <i>Remarks</i> | 0 dB |

Remarks

- When the unit is set to dB, the range of <offset> is from -1 GdB to 1 GdB. The default value is 0 dB.
- When the unit is set to V_{rms} , the range of <offset> is from -1 GV_{rms} to 1 GV_{rms} . The default value is 0 V_{rms} .

You can run the `:MATH<n>:FFT:UNIT` command to configure or query the current unit.

Return Format

The query returns the current vertical offset in scientific notation.

Example

```
:MATH1:FFT:OFFSet 0.3 /*Sets the vertical offset of the FFT
operation results to 300 mdB.*/
:MATH1:FFT:OFFSet? /*The query returns 3.000000E-1.*/
```

3.17.18 :MATH<n>:FFT:HSCale

Syntax

```
:MATH<n>:FFT:HSCale <hsc>
```

```
:MATH<n>:FFT:HSCale?
```

Description

Sets or queries the frequency range of FFT operation results. The default unit is Hz.

Parameter

| Name | Type | Range | Default |
|-------|----------|----------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <hsc> | Real | 10 Hz to 1 GHz | 10 MHz |

Remarks

You can reduce the frequency range to observe the details of the spectrum.

Modifying the frequency range of the FFT operation results will affect the value of the center frequency. You can run the `:MATH<n>:FFT:HCENter` command to query or modify the center frequency.

Return Format

The query returns the current frequency range in scientific notation.

Example

```
:MATH1:FFT:HSCale 500000 /*Sets the frequency range of the FFT
operation results to 500 kHz.*/
:MATH1:FFT:HSCale? /*The query returns 5.000000E+5.*/
```

3.17.19 :MATH<n>:FFT:HCENter**Syntax**

```
:MATH<n>:FFT:HCENter <cent>
```

```
:MATH<n>:FFT:HCENter?
```

Description

Sets or queries the center frequency of FFT operation results, i.g. the frequency relative to the horizontal center of the screen.

Parameter

| Name | Type | Range | Default |
|--------|----------|---------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <cent> | Real | 5 Hz to 1 GHz | 5 MHz |

Remarks

Modifying the center frequency of the FFT operation results will affect the value of the frequency range. You can run the `:MATH<n>:FFT:HSCale` command to query or modify the frequency range.

Return Format

The query returns the current center frequency in scientific notation. The unit is Hz.

Example

```
:MATH1:FFT:HCEnter 10000000 /*Sets the center frequency of the
FFT operation results to 10 MHz.*/
:MATH1:FFT:HCEnter? /*The query returns 1.000000E+7.*/
```

3.17.20 :MATH<n>:FFT:FREQuency:STARt**Syntax**

```
:MATH<n>:FFT:FREQuency:STARt <value>
```

```
:MATH<n>:FFT:FREQuency:STARt?
```

Description

Sets or queries the start frequency of FFT operation results.

Parameter

| Name | Type | Range | Default |
|---------|----------|-----------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <value> | Real | 0 Hz to (Stop Freq - 10 Hz) | 0 Hz |

Remarks

The range of the start frequency of FFT operation is related to the stop frequency. You can run the `:MATH<n>:FFT:FREQuency:END` command to query or configure the stop frequency.

Return Format

The query returns the start frequency of the operation results in scientific notation. The unit is Hz.

Example

```
:MATH1:FFT:FREQuency:STARt 10000000 /*Sets the start frequency of
the FFT operation results to 10 MHz.*/
:MATH1:FFT:FREQuency:STARt? /*The query returns 1.000000E
+7.*/
```

3.17.21 :MATH<n>:FFT:FREQuency:END**Syntax**

```
:MATH<n>:FFT:FREQuency:END <value>
```

```
:MATH<n>:FFT:FREQuency:END?
```

Description

Sets or queries the stop frequency of FFT operation results.

Parameter

| Name | Type | Range | Default |
|---------|----------|-------------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <value> | Real | (Start Freq + 10 Hz) to 1 GHz | 10 MHz |

Remarks

The range of the stop frequency of FFT operation is related to the start frequency. You can run the `:MATH<n>:FFT:FREQUENCY:START` command to query or configure the start frequency.

Return Format

The query returns the stop frequency of the operation results in scientific notation. The unit is Hz.

Example

```
:MATH1:FFT:FREQUENCY:END 10000000 /*Sets the stop frequency of
the FFT operation results to 10 MHz.*/
:MATH1:FFT:FREQUENCY:END? /*The query returns 1.000000E
+7.*/
```

3.17.22 :MATH<n>:FFT:SEARCH:ENABLE**Syntax**

```
:MATH<n>:FFT:SEARCH:ENABLE <bool>
```

```
:MATH<n>:FFT:SEARCH:ENABLE?
```

Description

Enables or disables the FFT peak search; or queries the on/off status of the FFT peak search function.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:MATH1:FFT:SEARCh:ENABle ON      /*Enables the FFT peak search.*/
:MATH1:FFT:SEARCh:ENABle?      /*The query returns 1.*/
```

3.17.23 :MATH<n>:FFT:SEARCh:NUM**Syntax**

```
:MATH<n>:FFT:SEARCh:NUM <num>
```

```
:MATH<n>:FFT:SEARCh:NUM?
```

Description

Sets or queries the maximum number of the FFT peak search.

Parameter

| Name | Type | Range | Default |
|-------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <num> | Integer | 1 to 15 | 5 |

Remarks

N/A

Return Format

The query returns an integer ranging from 1 to 15.

Example

```
:MATH1:FFT:SEARCh:NUM 10      /*Sets the maximum number of the
FFT peak search to 10.*/
:MATH1:FFT:SEARCh:NUM?      /*The query returns 10.*/
```

3.17.24 :MATH<n>:FFT:SEARCh:THReshold**Syntax**

```
:MATH<n>:FFT:SEARCh:THReshold <thres>
```

```
:MATH<n>:FFT:SEARCh:THReshold?
```

Description

Sets or queries the threshold of the FFT peak search.

Parameter

| Name | Type | Range | Default |
|---------|----------|--|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <thres> | Real | Related to the vertical scale and vertical offset of FFT operation | 5.5 dB |

Remarks

N/A

Return Format

The query returns the threshold in scientific notation.

Example

```
:MATH1:FFT:SEARCh:THReshold 0.5 /*Sets the threshold of the
FFT peak search to 500 mdB.*/
:MATH1:FFT:SEARCh:THReshold? /*The query returns
5.000000E-1.*/
```

3.17.25 :MATH<n>:FFT:SEARCh:EXCursion**Syntax**

```
:MATH<n>:FFT:SEARCh:EXCursion <excur>
```

```
:MATH<n>:FFT:SEARCh:EXCursion?
```

Description

Sets or queries the excursion of the FFT peak search.

Parameter

| Name | Type | Range | Default |
|---------|----------|--------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <excur> | Real | 0 to (8 x VerticalScale) | 1.8 dB |

Remarks

VerticalScale indicates the vertical scale of FFT.

Return Format

The query returns the excursion in scientific notation.

Example

```
:MATH1:FFT:SEARCh:EXCursion 0.5 /*Sets the excursion of the
FFT peak search to 500 mdB.*/
:MATH1:FFT:SEARCh:EXCursion? /*The query returns
5.000000E-1.*/
```

3.17.26 :MATH<n>:FFT:SEARCh:ORDER**Syntax**

```
:MATH<n>:FFT:SEARCh:ORDER <order>
```

```
:MATH<n>:FFT:SEARCh:ORDER?
```

Description

Sets or queries the sequence of the FFT peak search results.

Parameter

| Name | Type | Range | Default |
|---------|----------|----------------------|----------|
| <n> | Discrete | {1 2 3 4} | - |
| <order> | Discrete | {AMPorder FREQorder} | AMPorder |

Remarks

N/A

Return Format

The query returns AMP or FREQ.

Example

```
:MATH1:FFT:SEARCh:ORDER AMPorder /*Sets the sequence of the FFT
peak search results to AMPorder.*/
:MATH1:FFT:SEARCh:ORDER? /*The query returns AMP.*/
```

3.17.27 :MATH<n>:FFT:SEARCh:RES?**Syntax**

```
:MATH<n>:FFT:SEARCh:RES?
```

Description

Queries the peak search results table.

Parameter

| Name | Type | Range | Default |
|------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |

Remarks

N/A

Return Format

The query returns the peak search results table in strings.

Example

```
:MATH1:FFT:SEARCH:RES? /*The query returns the peak search
results table in strings.*/
1,2.50000MHz,-24.98dBV
2,3.50000MHz,-27.84dBV
3,4.50000MHz,-30.04dBV
4,5.50125MHz,-31.5dBV
5,6.50125MHz,-32.34dBV
```

3.17.28 :MATH<n>:FILTER:TYPE**Syntax**

```
:MATH<n>:FILTER:TYPE <type>
```

```
:MATH<n>:FILTER:TYPE?
```

Description

Sets or queries the filter type.

Parameter

| Name | Type | Range | Default |
|--------|----------|---------------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <type> | Discrete | {LPASs HPASs BPASs BStop} | LPASs |

Remarks

The oscilloscope provides 4 practical filters (Low Pass Filter, High Pass Filter, Band Pass Filter, and Band Stop Filter), which can filter the specified frequencies in the signal by setting the bandwidth. You can use the `:MATH<n>:FFT:SOURce` command to set or query the channel source of the filter.

- **LPASs:** indicates low pass filter, which only allows the signals whose frequencies are smaller than the current cut-off frequency to pass.

- **HPASs:** indicates high pass filter, which only allows the signals whose frequencies are greater than the current cut-off frequency to pass.
- **BPASs:** indicates band pass filter, which only allows the signals whose frequencies are greater than the current cut-off frequency 1 and smaller than the current cut-off frequency 2 to pass.

Note: The cut-off frequency 1 must be smaller than the cut-off frequency 2.

- **BSTop:** indicates band stop filter, which only allows the signals whose frequencies are smaller than the current cut-off frequency 1 or greater than the current cut-off frequency 2 to pass.

Note: The cut-off frequency 1 must be smaller than the cut-off frequency 2.

Return Format

The query returns LPAS, HPAS, BPAS, or BST.

Example

```
:MATH1:FILTER:TYPE LPASs /*Sets the filter type to Low Pass
Filter.*/*
:MATH1:FILTER:TYPE? /*The query returns LPAS.*/*
```

3.17.29 :MATH<n>:FILTER:W1

Syntax

```
:MATH<n>:FILTER:W1 <freq1>
```

```
:MATH<n>:FILTER:W1?
```

Description

Sets or queries the cut-off frequency of Low Pass Filter/High Pass Filter; or the cut-off frequency 1 of Band Pass Filter/Band Stop Filter. The default unit is Hz.

Parameter

| Name | Type | Range | Default |
|---------|----------|-------------------------|--|
| <n> | Discrete | {1 2 3 4} | - |
| <freq1> | Real | Refer to <i>Remarks</i> | Related to the filter type LPASs BPASs BSTop: |

| Name | Type | Range | Default |
|------|------|-------|--|
| | | | 0.005×screen sample rate HPASs: 0.1×screen sample rate |

Remarks

- When the filter type is set to LPASs (Low Pass Filter) or HPASs (High Pass Filter), you need to set one cut-off frequency. At this time, the range of <freq1> is from (0.005×screen sample rate) to (0.1×screen sample rate), at a step of (0.005×screen sample rate). Wherein, screen sample rate = 100/horizontal time base
- When the filter type is set to BPASs (Band Pass Filter) or BStop (Band Stop Filter), you need to set two cut-off frequencies. Run this command to set the cut-off frequency 1, and run the `:MATH<n>:FILTer:W2` command to set the cut-off frequency 2. At this time, the range of <freq1> is from (0.005×screen sample rate) to (0.095×screen sample rate), at a step of (0.005×screen sample rate). Wherein, screen sample rate = 100/horizontal time base

Note: The cut-off frequency 1 must be smaller than the cut-off frequency 2.

Return Format

The query returns the current cut-off frequency or cut-off frequency 1 in scientific notation.

Example

```
:MATH1:FILTer:W1 1000000 /*Sets the cut-off frequency of Low
Pass Filter to 1 MHz.*/
:MATH1:FILTer:W1? /*The query returns 1.000000E+6.*/
```

3.17.30 :MATH<n>:FILTer:W2

Syntax

```
:MATH<n>:FILTer:W2 <freq2>
```

```
:MATH<n>:FILTer:W2?
```

Description

Sets or queries the cut-off frequency 2 of Band Pass Filter/Band Stop Filter. The default unit is Hz.

Parameter

| Name | Type | Range | Default |
|---------|----------|-------------------------|------------------------|
| <n> | Discrete | {1 2 3 4} | - |
| <freq2> | Real | Refer to <i>Remarks</i> | 0.1×screen sample rate |

Remarks

When the filter type is set to BPASs (Band Pass Filter) or BStop (Band Stop Filter), you need to set two cut-off frequencies. Run the `:MATH<n>:FILTer:W1` command to set the cut-off frequency 1, and run this command to set the cut-off frequency 2. At this time, the range of <freq2> is from (0.01×screen sample rate) to (0.1×screen sample rate), at a step of (0.005×screen sample rate). Wherein, screen sample rate = 100/horizontal time base

Note: The cut-off frequency 2 must be greater than the cut-off frequency 1.

Return Format

The query returns the current cut-off frequency 2 in scientific notation.

Example

```
:MATH1:FILTer:W2 1500000 /*Sets the cut-off frequency 2 of
Band Pass Filter to 1.5 MHz.*/
:MATH1:FILTer:W2? /*The query returns 1.500000E+6.*/
```

3.17.31 :MATH<n>:SENSitivity

Syntax

```
:MATH<n>:SENSitivity <sens>
```

```
:MATH<n>:SENSitivity?
```

Description

Sets or queries the sensitivity of the logic operation. The default unit is div.

Parameter

| Name | Type | Range | Default |
|------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |

| Name | Type | Range | Default |
|--------|------|-------------------|----------|
| <sens> | Real | 100 mdiv to 1 div | 300 mdiv |

Remarks

N/A

Return Format

The query returns the sensitivity of the logic operation in scientific notation.

Example

```
:MATH1:SENSitivity 0.2 /*Sets the sensitivity of the logic
operation to 0.2 div.*/
:MATH1:SENSitivity? /*The query returns 2.000000E-1.*/
```

3.17.32 :MATH<n>:DISTance**Syntax**

```
:MATH<n>:DISTance <dist>
```

```
:MATH<n>:DISTance?
```

Description

Sets or queries the smoothing window width of differential operation.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <dist> | Integer | 1 to 1,000 | - |

Remarks

N/A

Return Format

The query returns an integer ranging from 1 to 1,000.

Example

```
:MATH1:DISTance 20 /*Sets the smoothing window width of
differential operation to 20.*/
:MATH1:DISTance? /*The query returns 20.*/
```

3.17.33 :MATH<n>:THReshold1

Syntax

```
:MATH<n>:THReshold1 <thre>
```

```
:MATH<n>:THReshold1?
```

Description

Sets or queries threshold level of Analog Channel 1 in the logic operation. The default unit is V.

Parameter

| Name | Type | Range | Default |
|--------|----------|---|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <thre> | Real | (-4 × VerticalScale - VerticalOffset) to (4 × VerticalScale - VerticalOffset) | 0 V |

Remarks

- This command is only available for the logic operations A&&B, A||B, A^B, and !A.
- VerticalScale indicates the vertical scale of Analog Channel 1. VerticalOffset indicates the vertical offset of Analog Channel 1. The step value is VerticalScale/10.

Return Format

The query returns the threshold level of of Analog Channel 1 in scientific notation.

Example

```
:MATH1:THReshold1 0.8 /*Sets the threshold level of Analog
Channel 1 in logic operation to 800 mV.*/
:MATH1:THReshold1? /*The query returns 8.000000E-1.*/
```

3.17.34 :MATH<n>:THReshold2

Syntax

```
:MATH<n>:THReshold2 <thre>
```

```
:MATH<n>:THReshold2?
```

Description

Sets or queries threshold level of Analog Channel 2 in the logic operation. The default unit is V.

Parameter

| Name | Type | Range | Default |
|--------|----------|---|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <thre> | Real | (-4 × VerticalScale - VerticalOffset) to (4 × VerticalScale - VerticalOffset) | 0 V |

Remarks

- This command is only available for the logic operations A&&B, A||B, A^B, and !A.
- VerticalScale indicates the vertical scale of Analog Channel 2. VerticalOffset indicates the vertical offset of Analog Channel 2. The step value is VerticalScale/10.

Return Format

The query returns the threshold level of of Analog Channel 2 in scientific notation.

Example

```
:MATH1:THReshold2 0.8 /*Sets the threshold level of Analog
Channel 2 in logic operation to 800 mV.*/
:MATH1:THReshold2? /*The query returns 8.000000E-1.*/
```

3.17.35 :MATH<n>:THReshold3**Syntax**

```
:MATH<n>:THReshold3 <thre>
```

```
:MATH<n>:THReshold3?
```

Description

Sets or queries threshold level of Analog Channel 3 in the logic operation. The default unit is V.

Parameter

| Name | Type | Range | Default |
|--------|----------|---|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <thre> | Real | (-4 × VerticalScale - VerticalOffset) to (4 × VerticalScale - VerticalOffset) | 0 V |

Remarks

- This command is only available for the logic operations A&&B, A||B, A^B, and !A.
- VerticalScale indicates the vertical scale of Analog Channel 3. VerticalOffset indicates the vertical offset of Analog Channel 3. The step value is VerticalScale/10.

Return Format

The query returns the threshold level of of Analog Channel 3 in scientific notation.

Example

```
:MATH1:THReshold3 0.8 /*Sets the threshold level of Analog
Channel 3 in logic operation to 800 mV.*/
:MATH1:THReshold3? /*The query returns 8.000000E-1.*/
```

3.17.36 :MATH<n>:THReshold4**Syntax**

```
:MATH<n>:THReshold4 <thre>
```

```
:MATH<n>:THReshold4?
```

Description

Sets or queries threshold level of Analog Channel 4 in the logic operation. The default unit is V.

Parameter

| Name | Type | Range | Default |
|------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |

| Name | Type | Range | Default |
|--------|------|---|---------|
| <thre> | Real | (-4 × VerticalScale - VerticalOffset) to (4 × VerticalScale - VerticalOffset) | 0 V |

Remarks

- This command is only available for the logic operations A&&B, A||B, A^B, and !A.
- VerticalScale indicates the vertical scale of Analog Channel 4. VerticalOffset indicates the vertical offset of Analog Channel 4. The step value is VerticalScale/10.

Return Format

The query returns the threshold level of of Analog Channel 4 in scientific notation.

Example

```
:MATH1:THReshold4 0.8 /*Sets the threshold level of Analog
Channel 4 in logic operation to 800 mV.*/
:MATH1:THReshold4? /*The query returns 8.000000E-1.*/
```

3.17.37 :MATH<n>:WINDow:TITLe?**Syntax**

```
:MATH<n>:WINDow:TITLe?
```

Description

Queries the title of the specified math operation window.

Parameter

| Name | Type | Range | Default |
|------|----------|-----------|---------|
| <n> | Discrete | {1 2 3 4} | - |

Remarks

N/A

Return Format

The query returns the title of the specified math operation window in strings.

Example

```
:MATH1:WINDow:TITLe? /*The query returns Math1 CH1*CH1 Scale:0U
Sa:2GSa/s.*/
```

3.17.38 :MATH<n>:LABel:SHOW**Syntax**

```
:MATH<n>:LABel:SHOW <bool>
```

```
:MATH<n>:LABel:SHOW?
```

Description

Sets or queries whether to display the waveform label of the specified operation.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <n> | Discrete | {1 2 3 4} | - |
| <bool> | Bool | {{1 ON}}{0 OFF}} | - |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:MATH1:LABel:ENABle ON /*Enables the display of the
label.*/
:MATH1:LABel:ENABle? /*The query returns 1.*/
```

3.18 :MEASure Commands

The :MEASure commands are used to set and query the measurement parameters.

3.18.1 :MEASure:SOURce**Syntax**

```
:MEASure:SOURce <source>
```

```
:MEASure:SOURce?
```

Description

Sets or queries the channel source of the current measurement parameter.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 MATH1 MATH2 MATH3 MATH4} | CHANnel1 |

Remarks

This command has the same function as the *:MEASure:SETup:DSA* and *:MEASure:SETup:PSA* commands.

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, MATH1, MATH2, MATH3, or MATH4.

Example

```
:MEASure:SOURce CHANnel2 /*Sets the channel source of the
measurement parameter to CHANnel2.*/
:MEASure:SOURce? /*The query returns CHAN2.*/
```

3.18.2 :MEASure:ITEM**Syntax**

```
:MEASure:ITEM <item> [, <src> [, <src> ]]
```

```
:MEASure:ITEM? <item> [, <src> [, <src> ]]
```

Description

Measures any waveform parameter of the specified source, or queries the statistical results of any waveform parameter of the specified source.

Parameter

| Name | Type | Range | Default |
|--------|----------|---|---------|
| <item> | Discrete | {VMAX VMIN VPP VTOP VBASE VAMP VAVG VRMS OVERshoot PREShoot MARea MPARea PERiod FREQuency RTIME FTIME PWIDth NWIDth PDUTy NDUTy TVMAX TVMIN PSLewrate NSLewrate VUPPer VMID VLOWer VARiance PVRMs PPULses NPULses PEDGes NEDGes RRDelay RFDelay | - |

| Name | Type | Range | Default |
|-------|----------|---|---------|
| | | FRDelay FFDelay RRPHase RFPHase FRPHase FFPHase ACRMs} | |
| <src> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 MATH1 MATH2 MATH3 MATH4} | - |

Remarks

- The parameter [,<src>[,<src>]] is used to set the source of the parameter under measurement.
- If the measurement parameter is a single source, you only need to set one source. If this parameter is omitted, then the source is by default, selected by the *:MEASure:SOURce* command.
- If the measurement parameter is a dual channel source, observe the following rules to determine the source that you've selected. That is, if the parameter <src> is omitted, the first source is, by default, the one that you've selected in the last sent command (*:MEASure:SOURce*, *:MEASure:SETup:PSA*, or *:MEASure:SETup:DSA*); the second source is, by default, the one that you've selected in the last sent command (*:MEASure:SETup:PSB* or *:MEASure:SETup:DSB*).

Return Format

The query returns the current measurement value in scientific notation.

Example

```
:MEASure:ITEM OVERshoot,CHANnel2 /*Enables the overshoot
measurement of CH2.*/
:MEASure:ITEM? OVERshoot,CHANnel2 /*The query returns
8.888889E-3.*/
```

3.18.3 :MEASure:CLEar

Syntax

```
:MEASure:CLEar
```

Description

Clears all the enabled measurement items.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.18.4 :MEASure:AMSource**Syntax**

```
:MEASure:AMSource <chan>
```

```
:MEASure:AMSource?
```

Description

Sets the source and displays all measurement values of the set source; or queries the channel source(s) of the all measurement function.

Parameter

| Name | Type | Range | Default |
|--------|----------|---|---------|
| <chan> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 OFF} | OFF |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, or OFF.

Example

```
:MEASure:AMSource CHANnel1 /*Sets the source to CHANnel1.*/
:MEASure:AMSource? /*The query returns CHAN1.*/
```

3.18.5 :MEASure:STATistic:COUNT

Syntax

```
:MEASure:STATistic:COUNT <val>
```

```
:MEASure:STATistic:COUNT?
```

Description

Sets or queries the statistics count.

Parameter

| Name | Type | Range | Default |
|-------|---------|--------------|---------|
| <val> | Integer | 2 to 100,000 | 1,000 |

Remarks

N/A

Return Format

The query returns an integer ranging from 2 to 100,000.

Example

```
:MEASure:STATistic:COUNT 1000 /*Sets the statistics count to
1,000.*/
:MEASure:STATistic:COUNT? /*The query returns 1000.*/
```

3.18.6 :MEASure:STATistic:DISPlay

Syntax

```
:MEASure:STATistic:DISPlay <bool>
```

```
:MEASure:STATistic:DISPlay?
```

Description

Enables or disables the statistical function; or queries the status of the statistical function.

Parameter

| Name | Type | Range | Default |
|--------|------|--------------------|---------|
| <bool> | Bool | {{1 ON}} {{0 OFF}} | 0 OFF |

Remarks

When the statistical function is enabled, the instrument makes statistics of the measurement results for at most 10 measurement items that are turned on last time and displays the statistical results.

Return Format

The query returns 1 or 0.

Example

```
:MEASure:STATistic:DISPlay ON /*Enables the statistical  
function.*/  
:MEASure:STATistic:DISPlay? /*The query returns 1.*/
```

3.18.7 :MEASure:STATistic:RESet

Syntax

```
:MEASure:STATistic:RESet
```

Description

Clears the history statistics data and makes statistics again.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

```
N/A
```

3.18.8 :MEASure:STATistic:ITEM

Syntax

```
:MEASure:STATistic:ITEM <item>[,<src>[,<src>]]
```

```
:MEASure:STATistic:ITEM?<type>,<item>[,<src>[,<src>]]
```

Description

Enables the statistical function of any waveform parameter of the specified source, or queries the statistical results of any waveform parameter of the specified source.

Parameter

| Name | Type | Range | Default |
|--------|----------|--|---------|
| <item> | Discrete | {VMAX VMIN VPP VTOP VBASE VAMP VAVG VRMS OVERshoot PREShoot MARea MPARea PERiod FREQuency RTIME FTIME PWIDth NWIDth PDUTy NDUTy TVMAX TVMIN PSLewrate NSLewrate VUPPer VMID VLOWer VARiance PVRMs PPULses NPULses PEDGes NEDGes RRDelay RFDelay FRDelay FFDelay RRPHase RFPHase FRPHase FFPHase ACRMs} | - |
| <src> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 MATH1 MATH2 MATH3 MATH4} | - |
| <type> | Discrete | {MAXimum MINimum CURRent AVERages DEVIation CNT} | - |

Remarks

- The parameter [,<src> [,<src>]] is used to set the source of the parameter under measurement.
- If the measurement parameter is a single source, you only need to set one source. If this parameter is omitted, then the source is, by default, the one that you've selected in the last sent command
(*:MEASure:SOURce*, *:MEASure:SETup:PSA*, or *:MEASure:SETup:DSA*).
- If the measurement parameter is a dual channel source, observe the following rules to determine the source that you've selected. That is, if the parameter <src> is omitted, the first source is, by default, the one that you've selected in the last sent command (*:MEASure:SOURce*, *:MEASure:SETup:PSA*, or *:MEASure:SETup:DSA*); the second source is, by default, the one that you've

selected in the last sent command (*:MEASure:SETup:PSB* or *:MEASure:SETup:DSB*).

Return Format

The query returns the statistical results in scientific notation.

Example

```
:MEASure:STATistic:ITEM VPP,CHANnel2 /*Enables the statistical
function of the peak-peak value of CH2.*/
:MEASure:STATistic:ITEM? MAXimum,VPP /*The query returns
9.120000E-1.*/
```

3.18.9 :MEASure:SETup:MAX

Syntax

:MEASure:SETup:MAX <value>

:MEASure:SETup:MAX?

Description

Sets or queries the threshold level upper limit of the analog channel in auto measurement.

Parameter

| Name | Type | Range | Default |
|---------|---------|-------------------------|---------|
| <value> | Integer | Refer to <i>Remarks</i> | - |

Remarks

The range of the threshold level upper limit is related to the current threshold middle value. You can send the *:MEASure:SETup:MID* command to set or query the threshold middle value of the current analog channel in auto measurement.

- When the threshold type is percentage, its range is from (threshold middle value + 1%) to 100%.
- When the threshold type is absolute, its range changes with the probe ratio. Its max. range is from -100 MV to 100 MV; and its min. range is from -20 V to 20 V.
- When the set upper limit is smaller than the current threshold middle value, a message "Set at lower limit" will be displayed, and the threshold middle value will not be modified automatically.

Return Format

The query returns an integer. When the threshold type is absolute, the default unit of the returned value is V.

Example

```
:MEASure:SETup:MAX 95 /*Sets the upper limit of the threshold
level to 95%.*/
:MEASure:SETup:MAX? /*The query returns 95.*/
```

3.18.10 :MEASure:SETup:MID**Syntax**

```
:MEASure:SETup:MID <value>
```

```
:MEASure:SETup:MID?
```

Description

Sets or queries the threshold level middle value of the analog channel in auto measurement.

Parameter

| Name | Type | Range | Default |
|---------|---------|-------------------------|---------|
| <value> | Integer | Refer to <i>Remarks</i> | - |

Remarks

The set middle value must be smaller than the currently set upper limit and greater than the currently set lower limit.

You can send the *:MEASure:SETup:MAX* and *:MEASure:SETup:MIN* commands to set or query the threshold level upper limit and lower limit of the current analog channel in auto measurement.

Return Format

The query returns an integer. When the threshold type is absolute, the default unit of the returned value is V.

Example

```
:MEASure:SETup:MID 89 /*Sets the middle value of the threshold
level to 89%.*/
:MEASure:SETup:MID? /*The query returns 89.*/
```

3.18.11 :MEASure:SETup:MIN**Syntax**

```
:MEASure:SETup:MIN <value>
```

:MEASure:SETup:MIN?

Description

Sets or queries the threshold level lower limit of the analog channel in auto measurement.

Parameter

| Name | Type | Range | Default |
|---------|---------|-------------------------|---------|
| <value> | Integer | Refer to <i>Remarks</i> | - |

Remarks

The range of the threshold level lower limit is related to the current threshold middle value. You can send the `:MEASure:SETup:MID` command to set or query the threshold middle value of the current analog channel in auto measurement.

- When the threshold type is percentage, its range is from 0% to (threshold middle value - 1%).
- When the threshold type is absolute, its range changes with the probe ratio. Its max. range is from -100 MV to 100 MV; and its min. range is from -20 V to 20 V.
- When the set lower limit is greater than the current threshold middle value, a message "Set at upper limit" will be displayed, and the threshold middle value will not be modified automatically.

Return Format

The query returns an integer. When the threshold type is absolute, the default unit of the returned value is V.

Example

```
:MEASure:SETup:MIN 53 /*Sets the lower limit of the threshold
level to 53%.*/
:MEASure:SETup:MIN? /*The query returns 53.*/
```

3.18.12 :MEASure:SETup:PSA

Syntax

:MEASure:SETup:PSA <source>

:MEASure:SETup:PSA?

Description

Sets or queries Source A in the phase or delay measurement.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 MATH1 MATH2 MATH3 MATH4} | CHANnel1 |

Remarks

This command has the same function as the *:MEASure:SOURce* and *:MEASure:SETup:DSA* commands.

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, MATH1, MATH2, MATH3, or MATH4.

Example

```
:MEASure:SETup:PSA CHANnel1 /*Sets Source A of the phase
measurement to CHANnel1.*/
:MEASure:SETup:PSA? /*The query returns CHAN1.*/
```

3.18.13 :MEASure:SETup:PSB**Syntax**

```
:MEASure:SETup:PSB <source>
```

```
:MEASure:SETup:PSB?
```

Description

Sets or queries Source B in the phase or delay measurement.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 MATH1 MATH2 MATH3 MATH4} | CHANnel1 |

Remarks

This command has the same function as the *:MEASure:SETup:DSB* command.

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, MATH1, MATH2, MATH3, or MATH4.

Example

```
:MEASure:SETup:PSB CHANnel2 /*Sets Source B of the phase
measurement to CHANnel2.*/
:MEASure:SETup:PSB? /*The query returns CHAN2.*/
```

3.18.14 :MEASure:SETup:DSA**Syntax**

```
:MEASure:SETup:DSA <source>
```

```
:MEASure:SETup:DSA?
```

Description

Sets or queries Source A in the phase or delay measurement.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 MATH1 MATH2 MATH3 MATH4} | CHANnel1 |

Remarks

This command has the same function as the *:MEASure:SOURce* and *:MEASure:SETup:PSA* commands.

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, MATH1, MATH2, MATH3, or MATH4.

Example

```
:MEASure:SETup:DSA CHANnel1 /*Sets Source A of the delay
measurement to CHANnel1.*/
:MEASure:SETup:DSA? /*The query returns CHAN1.*/
```

3.18.15 :MEASure:SETup:DSB**Syntax**

```
:MEASure:SETup:DSB <source>
```

```
:MEASure:SETup:DSB
```

Description

Sets or queries Source B in the phase or delay measurement.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 MATH1 MATH2 MATH3 MATH4} | CHANnel1 |

Remarks

This command has the same function as the `:MEASure:SETup:PSB` command.

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, MATH1, MATH2, MATH3, or MATH4.

Example

```
:MEASure:SETup:DSB CHANnel2 /*Sets Source B of the delay
measurement to CHANnel2.*/
:MEASure:SETup:DSB? /*The query returns CHAN2.*/
```

3.18.16 :MEASure:THReshold:SOURce**Syntax**

```
:MEASure:THReshold:SOURce <source>
```

```
:MEASure:THReshold:SOURce?
```

Description

Sets or queries the threshold source.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 MATH1 MATH2 MATH3 MATH4} | CHANnel1 |

Remarks

Modifying the threshold will affect the measurement results of time, delay and phase parameters.

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, MATH1, MATH2, MATH3, or MATH4.

Example

```
:MEASure:THReshold:SOURce CHANnel2 /*Sets the threshold source
to CHANnel2.*/
:MEASure:THReshold:SOURce? /*The query returns CHAN2.*/
```

3.18.17 :MEASure:THReshold:TYPE**Syntax**

```
:MEASure:THReshold:TYPE <type>
```

```
:MEASure:THReshold:TYPE?
```

Description

Sets or queries the measurement threshold type.

Parameter

| Name | Type | Range | Default |
|--------|----------|--------------------|---------|
| <type> | Discrete | {PERCent ABSolute} | PERCent |

Remarks

N/A

Return Format

The query returns PERC or ABS.

Example

```
:MEASure:THReshold:TYPE ABSolute /*Sets the threshold type to
ABSolute.*/
:MEASure:THReshold:TYPE? /*The query returns ABS.*/
```

3.18.18 :MEASure:THReshold:DEFAult**Syntax**

```
:MEASure:THReshold:DEFAult
```

Description

Sets the threshold level of the analog channel in auto measurement to a default value.

Parameter

N/A

Remarks

For the default threshold level in absolute, its upper limit and lower limit are + (vertical scale x 3) and -(vertical scale x 3), respectively.

Return Format

N/A

Example

N/A

3.18.19 :MEASure:AREA**Syntax**

```
:MEASure:AREA <area>
```

```
:MEASure:AREA?
```

Description

Sets or queries the type of the measurement range.

Parameter

| Name | Type | Range | Default |
|--------|----------|-------------|---------|
| <area> | Discrete | {MAIN ZOOM} | MAIN |

Remarks

- **MAIN:** indicates that the measurement range is within the main time base region.
- **ZOOM:** indicates that the measurement range is within the zoomed time base region. Note that only when you enable the delayed sweep function first, can "Zoom" be enabled.

Return Format

The query returns MAIN, ZOOM.

Example

```
:MEASure:AREA ZOOM /*Sets the type of the measurement
range to ZOOM.*/
:MEASure:AREA? /*The query returns ZOOM.*/
```


3.18.20 :MEASure:INDicator

Syntax

```
:MEASure:INDicator <bool>
```

```
:MEASure:INDicator?
```

Description

Sets or queries the on/off status of the measurement auto cursor.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:MEASure:INDicator ON /*Sets the measurement auto cursor to be
on.*/
:MEASure:INDicator? /*The query returns 1.*/
```

3.18.21 :MEASure:COUNter:ENABLE

Syntax

```
:MEASure:COUNter:ENABLE <bool>
```

```
:MEASure:COUNter:ENABLE?
```

Description

Sets or queries the on/off status of the frequency counter.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:MEASure:COUNter:ENABle ON      /*Enables the frequency counter.*/
:MEASure:COUNter:ENABle?       /*The query returns 1.*/
```

3.18.22 :MEASure:COUNter:SOURce**Syntax**

```
:MEASure:COUNter:SOURce <source>
```

```
:MEASure:COUNter:SOURce?
```

Description

Sets or queries the measurement source for the frequency counter.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 EXT} | CHANnel1 |

Remarks

The measurement source of the frequency counter is CH1-CH4 and EXT.

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, or EXT.

Example

```
:MEASure:COUNter:SOURce CHANnel4 /*Sets the measurement source
for the frequency counter to CHANnel4.*/
:MEASure:COUNter:SOURce?         /*The query returns CHAN4.*/
```

3.18.23 :MEASure:COUNter:VALue?**Syntax**

```
:MEASure:COUNter:VALue?
```

Description

Queries the measurement results of the frequency counter.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the measurement results of the frequency counter in scientific notation.

Example

```
:MEASure:COUNter:VALue? /*The query returns 9.999996E-04.*/
```

3.18.24 :MEASure:AMP:TYPE**Syntax**

```
:MEASure:AMP:TYPE <val>
```

```
:MEASure:AMP:TYPE?
```

Description

Sets or queries the amplitude method.

Parameter

| Name | Type | Range | Default |
|-------|----------|---------------|---------|
| <val> | Discrete | {AUTO MANual} | MANual |

Remarks

- **AUTO:** indicates the Auto method.
- **MANual:** indicates the Manual method.

Return Format

The query returns AUTO or MAN.

Example

```
:MEASure:AMP:TYPE MANual /*Sets the amplitude method to MANual.*/
:MEASure:AMP:TYPE? /*The query returns MAN.*/
```

3.18.25 :MEASure:AMP:MANual:TOP**Syntax**

```
:MEASure:AMP:MANual:TOP <val>
```

```
:MEASure:AMP:MANual:TOP?
```

Description

Sets or queries the amplitude top value type for the manual amplitude method.

Parameter

| Name | Type | Range | Default |
|-------|----------|--------------------|-----------|
| <val> | Discrete | {HISTogram MAXMin} | HISTogram |

Remarks

- **HISTogram:** indicates the histogram type.
- **MAXMin:** indicates the Max-Min type.

Return Format

The query returns HIST or MAXM.

Example

```
:MEASure:AMP:MANual:TOP MAXMin /*Sets the amplitude top value
type for the manual amplitude method to MAXMin.*/
:MEASure:AMP:MANual:TOP? /*The query returns MAXM.*/
```

3.18.26 :MEASure:AMP:MANual:BASE**Syntax**

```
:MEASure:AMP:MANual:BASE <val>
```

```
:MEASure:AMP:MANual:BASE?
```

Description

Sets or queries the amplitude base value type for the manual amplitude method.

Parameter

| Name | Type | Range | Default |
|-------|----------|--------------------|-----------|
| <val> | Discrete | {HISTogram MAXMin} | HISTogram |

Remarks

- **HISTogram:** indicates the histogram type.
- **MAXMin:** indicates the Max-Min type.

Return Format

The query returns HIST or MAXM.

Example

```
:MEASure:AMP:MANual:BASE MAXMin /*Sets the amplitude base value
type for the manual amplitude method to MAXMin.*/
:MEASure:AMP:MANual:BASE? /*The query returns MAXM.*/
```

3.18.27 :MEASure:CATegory**Syntax**

```
:MEASure:CATegory <val>
```

```
:MEASure:CATegory?
```

Description

Sets or queries the measurement type.

Parameter

| Name | Type | Range | Default |
|-------|---------|--------|---------|
| <val> | Integer | 0 to 2 | 0 |

Remarks

0: horizontal; 1: vertical; 2: other.

Return Format

The query returns an integer ranging from 0 to 2.

Example

```
:MEASure:CATegory 1 /*Sets the measurement type to
Vertical.*/
:MEASure:CATegory? /*The query returns 1.*/
```

3.19 :POWER Commands

The :POWER commands are used to set and query relevant parameters for the power analysis function.

**NOTE**

The DHO1000 series oscilloscope does not support :POWER commands.

The :POWER commands are optional configurations for the DHO4000 series oscilloscope.

3.19.1 :POWER:TYPE**Syntax**

```
:POWER:TYPE <type>
```

:POWER:TYPE?

Description

Sets or queries the power analysis type.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------|---------|
| <type> | Discrete | {QUALity RIPPlE} | QUALity |

Remarks

- **QUALity**: tests the quality of AC input signal.
- **RIPPlE**: tests the ripple of the DC output voltage.

Return Format

The query returns QUAL or RIPP.

Example

```
:POWER:TYPE RIPPlE          /*Sets the power analysis type to
RIPPlE.*/*
:POWER:TYPE?                /*The query returns RIPP.*/*
```

3.19.2 :POWER:CURRENT:SOURce

Syntax

:POWER:CURRENT:SOURce <source>

:POWER:CURRENT:SOURce?

Description

Sets or queries the current source of the power quality.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:POWer:CURRent:SOURce CHANnel2      /*Sets the current source to
CHANnel2.*/
:POWer:CURRent:SOURce?              /*The query returns CHAN2.*/
```

3.19.3 :POWer:VOLTage:SOURce**Syntax**

```
:POWer:VOLTage:SOURce <source>
```

```
:POWer:VOLTage:SOURce?
```

Description

Sets or queries the voltage source of the power quality.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:POWer:VOLTage:SOURce CHANnel2      /*Sets the voltage source of
the power quality to CHANnel2.*/
:POWer:VOLTage:SOURce?              /*The query returns CHAN2.*/
```

3.19.4 :POWer:REFLevel:METHod**Syntax**

```
:POWer:REFLevel:METHod <method>
```

```
:POWer:REFLevel:METHod?
```

Description

Sets or queries the reference level type of power quality.

Parameter

| Name | Type | Range | Default |
|----------|----------|--------------------|---------|
| <method> | Discrete | {ABSolute PERCent} | PERCent |

Remarks

N/A

Return Format

The query returns ABS or PERC.

Example

```
:POWer:REFLevel:METhod ABSolute /*Sets the reference level
type of power quality to ABSolute.*/
:POWer:REFLevel:METhod? /*The query returns ABS.*/
```

3.19.5 :POWer:REFLevel:ABSolute:HIGH**Syntax**

```
:POWer:REFLevel:ABSolute:HIGH <value>
```

```
:POWer:REFLevel:ABSolute:HIGH?
```

Description

Sets or queries the upper limit of the absolute value of the reference level of the power quality.

Parameter

| Name | Type | Range | Default |
|---------|------|----------------|---------|
| <value> | Real | [-100MV,100MV] | 300 mV |

Remarks

Upper limit > middle value > lower limit

Return Format

The query returns the upper limit of the absolute value of the reference level of the power quality. The unit is V.

Example

```
:POWer:REFLevel:ABSolute:HIGH 50mV /*Sets the upper limit of the
absolute value of the reference level of the power quality to 50
mV.*/
:POWer:REFLevel:ABSolute:HIGH? /*The query returns 5.000000E-2.*/
```


3.19.6 :POWER:REFLevel:ABSolute:LOW

Syntax

```
:POWER:REFLevel:ABSolute:LOW <value>
```

```
:POWER:REFLevel:ABSolute:LOW?
```

Description

Sets or queries the lower limit of the absolute value of the reference level of the power quality.

Parameter

| Name | Type | Range | Default |
|---------|------|----------------|---------|
| <value> | Real | [-100MV,100MV] | -300 mV |

Remarks

Upper limit > middle value > lower limit

Return Format

The query returns the lower limit of the absolute value of the reference level of the power quality. The unit is V.

Example

```
:POWER:REFLevel:ABSolute:LOW -50mV /*Sets the lower limit of the
absolute value of the reference level of the power quality to -50
mV.*/
:POWER:REFLevel:ABSolute:LOW? /*The query returns -5.000000E-2.*/
```

3.19.7 :POWER:REFLevel:ABSolute:MID

Syntax

```
:POWER:REFLevel:ABSolute:MID <value>
```

```
:POWER:REFLevel:ABSolute:MID?
```

Description

Sets or queries the middle absolute value of the reference level of power quality.

Parameter

| Name | Type | Range | Default |
|---------|------|----------------|---------|
| <value> | Real | [-100MV,100MV] | 0 |

Remarks

Upper limit > middle value > lower limit

Return Format

The query returns the middle absolute value of the reference level of power quality.
The unit is V.

Example

```
:POWer:REFLevel:ABSolute:MID 0 /*Sets the middle absolute value of
the reference level of power quality to 0.*/
:POWer:REFLevel:ABSolute:MID? /*The query returns 0.000000.*/
```

3.19.8 :POWer:REFLevel:PERCent:HIGH**Syntax**

```
:POWer:REFLevel:PERCent:HIGH <value>
```

```
:POWer:REFLevel:PERCent:HIGH?
```

Description

Sets or queries the reference level percentage upper limit of power quality.

Parameter

| Name | Type | Range | Default |
|---------|---------|----------|---------|
| <value> | Integer | 0 to 100 | 90 |

Remarks

Upper limit > middle value > lower limit

Return Format

The query returns an integer ranging from 0 to 100.

Example

```
:POWer:REFLevel:PERCent:HIGH 20 /*Sets the reference level
percentage upper limit of power quality to 20%.*/
:POWer:REFLevel:PERCent:HIGH? /*The query returns 20.*/
```

3.19.9 :POWer:REFLevel:PERCent:LOW**Syntax**

```
:POWer:REFLevel:PERCent:LOW <value>
```

```
:POWer:REFLevel:PERCent:LOW?
```

Description

Sets or queries the reference level percentage lower limit of power quality.

Parameter

| Name | Type | Range | Default |
|---------|---------|----------|---------|
| <value> | Integer | 0 to 100 | 10 |

Remarks

Upper limit > middle value > lower limit

Return Format

The query returns an integer ranging from 0 to 100.

Example

```
:POWer:REFLevel:PERCent:LOW 20 /*Sets the reference level
percentage lower limit of power quality to 20%.*/
:POWer:REFLevel:PERCent:LOW? /*The query returns 20.*/
```

3.19.10 :POWer:REFLevel:PERCent:MID**Syntax**

```
:POWer:REFLevel:PERCent:MID <value>
```

```
:POWer:REFLevel:PERCent:MID?
```

Description

Sets or queries the middle percentage value of the reference level of power quality.

Parameter

| Name | Type | Range | Default |
|---------|---------|----------|---------|
| <value> | Integer | 0 to 100 | 50 |

Remarks

Upper limit > middle value > lower limit

Return Format

The query returns an integer ranging from 0 to 100.

Example

```
:POWer:REFLevel:PERCent:MID 20 /*Sets the middle percentage
value of the reference level of power quality to 20%.*/
:POWer:REFLevel:PERCent:MID? /*The query returns 20.*/
```

3.19.11 :POWER:QUALITY:DISPlay

Syntax

```
:POWER:QUALITY:DISPlay <bool>
```

```
:POWER:QUALITY:DISPlay?
```

Description

Sets or queries the display status of the power quality analysis results.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:POWER:QUALITY:DISPlay ON /*Enables the display of the power
quality analysis results.*/
:POWER:RIPPLE:DISPlay? /*The query returns 1.*/
```

3.19.12 :POWER:QUALITY:FREQ:REference

Syntax

```
:POWER:QUALITY:FREQ:REference <source>
```

```
:POWER:QUALITY:FREQ:REreference?
```

Description

Sets or queries the frequency reference source of power quality.

Parameter

| Name | Type | Range | Default |
|----------|----------|-------------------|---------|
| <source> | Discrete | {VOLTage CURRENT} | VOLTage |

Remarks

N/A

Return Format

The query returns VOLT or CURR.

Example

```
:POWer:QUALity:FREQ:REFerence CURRent /*Sets the frequency
reference source of power quality to CURRent.*/
:POWer:QUALity:FREQ:REFerence? /*The query returns
CURR.*/
```

3.19.13 :POWer:QUALity:STATistics:RESult?**Syntax**

```
:POWer:QUALity:STATistics:RESult?
```

Description

Queries the statistical results of the power quality analysis.

Remarks

N/A

Return Format

The query returns the statistical results of the power quality analysis.

Example

```
:POWer:QUALity:STATistics:RESult? /*The query returns the
statistical results of the power quality analysis in strings.*/
[["Name", "SourceA", "SourceB", "Current", "Average", "Maximum", "Minimum",
"Deviation", "Count"]
["Ref_Freq", "1", "0", "999.50kHz", "1.0000MHz", "1.0010MHz", "998.50kHz",
"572.43Hz", "271"]
["Vrms", "1", "0", "88.273mV", "88.247mV", "88.344mV", "88.151mV", "71.570u
V", "271"]
["Irms", "2", "0", "89.039mA", "89.036mA", "89.286mA", "88.678mA", "86.316u
A", "271"]
["Real_P", "38", "0", "4.5289mW", "4.5237mW", "4.5548mW", "4.4843mW", "8.31
39uW", "271"]
["Apparent_P", "1", "2", "7.8598mVA", "7.8572mVA", "7.8810mVA", "7.8231mVA
", "6.6072uVA", "271"]
["Reactive_P", "1", "2", "6.4237mVAR", "6.4243mVAR", "6.4486mVAR", "6.3988
mVAR", "6.6072uVAR", "271"]
["P_Factor", "1", "2", "576.21m", "575.74m", "578.52m", "572.63m", "690.53u
", "271"]
["Phase_Angle", "1", "2", "54.81degree", "54.84degree", "55.06degree", "54
.65degree", "58.463degree", "271"]
["Imp", "1", "2", "991.39ohm", "991.14ohm", "994.82ohm", "988.43ohm", "598.
02ohm", "271"]
["V_CrestFactor", "1", "0", "1.3798", "1.3801", "1.3845", "1.3773", "598.02
u", "271"]
["I_CrestFactor", "2", "0", "1.3703", "1.3708", "1.3838", "1.3646", "2.5605
m", "271"]]
```

3.19.14 :POWER:RIPPLE:SOURce

Syntax

```
:POWER:RIPPLE:SOURce <source>
```

```
:POWER:RIPPLE:SOURce?
```

Description

Sets or queries the ripple signal source.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:POWER:RIPPLE:SOURce CHANnel1      /*Sets the ripple signal
source to CHANnel1.*/
:POWER:RIPPLE:SOURce?                /*The query returns CHAN1.*/
```

3.19.15 :POWER:RIPPLE:DISPlay

Syntax

```
:POWER:RIPPLE:DISPlay <bool>
```

```
:POWER:RIPPLE:DISPlay?
```

Description

Sets or queries the on/off status of the display of the ripple analysis result.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:POWer:RIPPlE:DISPlay ON /*Enables the display of the ripple analysis result.*/  
:POWer:RIPPlE:DISPlay? /*The query returns 1.*/
```

3.19.16 :POWer:RIPPlE:STATistics:RESult?

Syntax

```
:POWer:RIPPlE:STATistics:RESult?
```

Description

Queries the statistical results of the ripple analysis.

Remarks

N/A

Return Format

The query returns the statistical results of the ripple analysis.

Example

```
:POWer:RIPPlE:STATistics:RESult? /*The query returns the statistical results of the ripple analysis in strings.*/  
[["Name", "SourceA", "SourceB", "Current", "Average", "Maximum", "Minimum", "Deviation", "Count"]  
["Meas_Vpp", "1", "0", "249.79mV", "249.82mV", "250.25mV", "249.45mV", "183.10uV", "57"]]
```

3.19.17 :POWer:STATistics:RESet

Syntax

```
:POWer:STATistics:RESet
```

Description

Resets the statistics.

Remarks

N/A

Return Format

N/A

Example

N/A

3.20 :QUICK Command

The **:QUICK** command is used to set and query the relevant parameters for quick operation shortcut keys.

3.20.1 :QUICK:OPERATION

Syntax

```
:QUICK:OPERATION <type>
```

```
:QUICK:OPERATION?
```

Description

Sets or queries the type of the shortcut keys.

Parameter

| Name | Type | Range | Default |
|--------|----------|--|---------|
| <type> | Discrete | {SIMage SWAVe SSETup AMEasure SRESet RECORD SSAVE} | SIMage |

Remarks

- **SIMage**: indicates the screen image.
- **SWAVe**: indicates the waveform saving.
- **SSETup**: indicates the setup saving.
- **AMEasure**: indicates all measurement.
- **SRESet**: indicates statistics reset.
- **RECORD**: indicates waveform recording.
- **SSAVE**: indicates saving group.

Return Format

The query returns SIM, SWAV, SSET, AME, REC, SSAV, or SRES.

Example

```
:QUICK:OPERATION SWAVe /*Sets the type of the shortcut key to
"save waveform".*/
:QUICK:OPERATION? /*The query returns SWAV.*/
```


3.21 :RECORD Commands

3.21.1 :RECORD:WRECORD:ENABLE

Syntax

```
:RECORD:WRECORD:ENABLE <bool>
```

```
:RECORD:WRECORD:ENABLE?
```

Description

Enables or disables the waveform recording function; or queries the on/off status of the waveform recording function.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 0 or 1.

Example

```
:RECORD:WRECORD:ENABLE ON /*Enables the waveform recording
function.*/
:RECORD:WRECORD:ENABLE? /*The query returns 1.*/
```

3.21.2 :RECORD:ENABLE

Syntax

```
:RECORD:ENABLE <bool>
```

```
:RECORD:ENABLE?
```

Description

Enables or disables the waveform recording function; or queries the on/off status of the waveform recording function.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

This command exists for backwards compatibility. Use the command `:RECORD:WRECORD:ENABLE`.

Return Format

The query returns 0 or 1.

Example

```
:RECORD:ENABLE ON /*Enables the waveform recording function.*/
:RECORD:ENABLE? /*The query returns 1.*/
```

3.21.3 :RECORD:WRECORD:OPERate**Syntax**

```
:RECORD:WRECORD:OPERate <operate>
```

```
:RECORD:WRECORD:OPERate?
```

Description

Sets to start the waveform recording, or queries whether the waveform recording starts or stops.

Parameter

| Name | Type | Range | Default |
|-----------|----------|------------|---------|
| <operate> | Discrete | {RUN STOP} | STOP |

Remarks

N/A

Return Format

The query returns RUN or STOP.

Example

```
:RECORD:WRECORD:OPERate RUN /*Sets to start recording
waveforms.*/
:RECORD:WRECORD:OPERate? /*The query returns RUN.*/
```

3.21.4 :RECORD:START**Syntax**

```
:RECORD:START <bool>
```

```
:RECORD:START?
```

Description

Sets to start the waveform recording, or queries whether the waveform recording starts or stops.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

This command exists for backwards compatibility. Use the command *:RECORD:WRECORD:OPERate*.

Return Format

The query returns 1 or 0.

Example

```
:RECORD:START ON /*Sets to start recording the waveforms.*/
:RECORD:START? /*The query returns 1.*/
```

3.21.5 :RECORD:WRECORD:FRAMES**Syntax**

```
:RECORD:WRECORD:FRAMES <value>
```

```
:RECORD:WRECORD:FRAMES?
```

Description

Sets or queries the number of frames for waveform recording.

Parameter

| Name | Type | Range | Default |
|---------|---------|--|---------|
| <value> | Integer | 1 to the maximum number of frames that can be recorded currently | 1,000 |

Remarks

N/A

Return Format

The query returns an integer ranging from 1 to the maximum number of frames that can be recorded currently.

Example

```
:RECORD:WRECORD:FRAMES 300 /*Sets the number of recorded frames
to 300.*/
:RECORD:WRECORD:FRAMES? /*The query returns 300.*/
```

3.21.6 :RECORD:FRAMES**Syntax**

```
:RECORD:FRAMES <value>
```

```
:RECORD:FRAMES?
```

Description

Sets or queries the number of frames for waveform recording.

Parameter

| Name | Type | Range | Default |
|---------|---------|--|---------|
| <value> | Integer | 1 to the maximum number of frames that can be recorded currently | 1,000 |

Remarks

This command exists for backwards compatibility. Use the command *:RECORD:WRECORD:FRAMES*.

Return Format

The query returns an integer ranging from 1 to the maximum number of frames that can be recorded currently.

Example

```
:RECORD:FRAMES 300 /*Sets the number of recorded frames to 300.*/
:RECORD:FRAMES? /*The query returns 300.*/
```

3.21.7 :RECORD:WRECORD:FRAMES:MAX**Syntax**

```
:RECORD:WRECORD:FRAMES:MAX
```

Description

Sets the number of recorded frames to the maximum number of frames.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

```
:RECORD:WRECORD:FRAMES:MAX /*Sets the number of recorded frames to the maximum number of frames.*/
```

3.21.8 :RECORD:WRECORD: FMAX?

Syntax

```
:RECORD:WRECORD: FMAX?
```

Description

Queries the maximum number of frames that can be recorded currently.

Parameter

N/A

Remarks

N/A

Return Format

The query returns an integer. The maximum number of frames that can be recorded currently is determined by the current memory depth.

Example

N/A

3.21.9 :RECORD:WRECORD:FINTEerval

Syntax

```
:RECORD:WRECORD:FINTEerval <interval>
```

```
:RECORD:WRECORD:FINTEerval?
```

Description

Sets or queries the time interval between frames in waveform recording.

Parameter

| Name | Type | Range | Default |
|------------|------|--------------|---------|
| <interval> | Real | 10 ns to 1 s | 10 ns |

Remarks

N/A

Return Format

The query returns the time interval in scientific notation. The unit is s.

Example

```
:RECORD:WRECORD:FINterval 1 /*Sets the time interval between frames
in waveform recording to 1 s.*/
:RECORD:WRECORD:FINterval? /*The query returns 1.000000E0.*/
```

3.21.10 :RECORD:WRECORD:PROMpt**Syntax**

```
:RECORD:WRECORD:PROMpt <bool>
```

```
:RECORD:WRECORD:PROMpt?
```

Description

Sets or queries the on/off status of the beeper when the recording is completed.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 1 ON |

Remarks

N/A

Return Format

The query returns 0 or 1.

Example

```
:RECORD:WRECORD:PROMpt ON /*Enables the beeper when the
recording is completed.*/
:RECORD:WRECORD:PROMpt? /*The query returns 1.*/
```

3.21.11 :RECORD:WREPLAY:FCURRENT

Syntax

```
:RECORD:WREPLAY:FCURRENT <value>
```

```
:RECORD:WREPLAY:FCURRENT?
```

Description

Sets or queries the current frame in waveform playing.

Parameter

| Name | Type | Range | Default |
|---------|---------|--|---------------------------------------|
| <value> | Integer | 1 to the maximum number of frames recorded | The maximum number of frames recorded |

Remarks

N/A

Return Format

The query returns an integer.

Example

```
:RECORD:WREPLAY:FCURRENT 300 /*Sets the current frame for
waveform playing to 300.*/
:RECORD:WREPLAY:FCURRENT? /*The query returns 300.*/
```

3.21.12 :RECORD:CURRENT

Syntax

```
:RECORD:CURRENT <value>
```

```
:RECORD:CURRENT?
```

Description

Sets or queries the current frame in waveform playing.

Parameter

| Name | Type | Range | Default |
|---------|---------|--|-----------------------|
| <value> | Integer | 1 to the maximum number of frames recorded | The maximum number of |

| Name | Type | Range | Default |
|------|------|-------|-----------------|
| | | | frames recorded |

Remarks

This command exists for backwards compatibility. Use the command `:RECORD:WREPLAY:FCURRENT`.

Return Format

The query returns an integer.

Example

```
:RECORD:CURRENT 300 /*Sets the current frame for waveform
playing to 300.*/
:RECORD:CURRENT? /*The query returns 300.*/
```

3.21.13 :RECORD:WREPLAY:FCURRENT:TIME?**Syntax**

```
:RECORD:WREPLAY:FCURRENT:TIME?
```

Description

Queries the time stamp of the current frame in waveform playing.

Parameter

N/A

Remarks

N/A

Return Format

Queries the time stamp of the current frame in strings in waveform playing.

Example

N/A

3.21.14 :RECORD:WREPLAY:FSTART**Syntax**

```
:RECORD:WREPLAY:FSTART <start>
```

```
:RECORD:WREPLAY:FSTART?
```


Description

Sets or queries the start frame in waveform playback.

Parameter

| Name | Type | Range | Default |
|---------|---------|---|---------|
| <start> | Integer | 1 to the maximum number of frames that can be played back currently | - |

Remarks

N/A

Return Format

The query returns the start frame in integer.

Example

```
:RECORD:WREPlay:FStart 10 /*Sets the start frame in waveform
playing to 10.*/
:RECORD:WREPlay:FStart? /*The query returns 10.*/
```

3.21.15 :RECORD:WREPlay:FEND**Syntax**

```
:RECORD:WREPlay:FEND <end>
```

```
:RECORD:WREPlay:FEND?
```

Description

Sets or queries the end frame in waveform playback.

Parameter

| Name | Type | Range | Default |
|-------|---------|--|---------|
| <end> | Integer | 1 to the maximum number of frames recorded | - |

Remarks

N/A

Return Format

The query returns the end frame in integer.

Example

```
:RECORD:WREPlay:FEND 346 /*Sets the end frame for waveform
playing to 346.*/
:RECORD:WREPlay:FEND? /*The query returns 346.*/
```

3.21.16 :RECORD:WREPlay:FMAX?**Syntax**

```
:RECORD:WREPlay:FMAX?
```

Description

Queries the maximum number of frames that can be played back currently.

Parameter

N/A

Remarks

N/A

Return Format

The query returns an integer ranging from 0 to the number of frames that have been recorded currently.

Example

N/A

3.21.17 :RECORD:WREPlay:FINterval**Syntax**

```
:RECORD:WREPlay:FINterval <interval>
```

```
:RECORD:WREPlay:FINterval?
```

Description

Sets or queries the time interval between frames in waveform playback.

Parameter

| Name | Type | Range | Default |
|------------|------|-------------|---------|
| <interval> | Real | 1 ms to 1 s | - |

Remarks

N/A

Return Format

The query returns the time interval in scientific notation. The unit is s.

Example

```
:RECORD WREPlay:FINTerval 1 /*Sets the time interval between frames
in waveform playback to 1 s.*/
:RECORD:WREPlay:FINTerval? /*The query returns 1.000000E0.*/
```

3.21.18 :RECORD:WREPlay:MODE**Syntax**

```
:RECORD:WREPlay:MODE <mode>
```

```
:RECORD:WREPlay:MODE?
```

Description

Sets the waveform playback mode to Repeat or Single; queries the waveform playback mode.

Parameter

| Name | Type | Range | Default |
|--------|----------|-----------------|---------|
| <mode> | Discrete | {REPeat SINGle} | SINGLE |

Remarks

N/A

Return Format

The query returns REP or SING.

Example

```
:RECORD:WREPlay:MODE REP /*Sets the playback mode to REP.*/
:RECORD:WREPlay:MODE? /*The query returns REP.*/
```

3.21.19 :RECORD:WREPlay:DIRection**Syntax**

```
:RECORD:WREPlay:DIRection <direction>
```

```
:RECORD:WREPlay:DIRection?
```

Description

Sets or queries the playback direction in waveform playing.

Parameter

| Name | Type | Range | Default |
|-------------|----------|--------------------|---------|
| <direction> | Discrete | {FORWARD BACKWARD} | FORWARD |

Remarks

N/A

Return Format

The query returns FORW or BACK.

Example

```
:RECORD:WREPLAY:DIRrection BACK /*Sets the direction of waveform
playback to BACK.*/
:RECORD:WREPLAY:DIRrection? /*The query returns BACK.*/
```

3.21.20 :RECORD:WREPLAY:OPERate**Syntax**

```
:RECORD:WREPLAY:OPERate <operate>
```

```
:RECORD:WREPLAY:OPERate?
```

Description

Enables or disables the waveform playing function; or queries the on/off status of the waveform playing function.

Parameter

| Name | Type | Range | Default |
|-----------|----------|------------|---------|
| <operate> | Discrete | {RUN STOP} | STOP |

Remarks

N/A

Return Format

The query returns RUN or STOP.

Example

```
:RECORD:WREPLAY:OPERate RUN /*Sets to play the waveforms.*/
:RECORD:WREPLAY:OPERate? /*The query returns RUN.*/
```

3.21.21 :RECORD:PLAY

Syntax

```
:RECORD:PLAY <bool>
```

```
:RECORD:PLAY?
```

Description

Enables or disables the waveform playing function; or queries the on/off status of the waveform playing function.

Parameter

| Name | Type | Range | Default |
|--------|------|-------------------|---------|
| <bool> | Bool | {{1 ON}}{{0 OFF}} | 0 OFF |

Remarks

This command exists for backwards compatibility. Use the command *:RECORD:WREPLAY:OPERate*.

Return Format

The query returns 1 or 0.

Example

```
:RECORD:PLAY ON /*Sets to play the waveforms.*/
:RECORD:PLAY? /*The query returns 1.*/
```

3.21.22 :RECORD:WREPLAY:BACK

Syntax

```
:RECORD:WREPLAY:BACK
```

Description

Plays back the previous frame of waveforms manually.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

```
:RECORD:WREPlay:BACK /*Plays back the previous frame of waveforms manually.*/*
```

3.21.23 :RECORD:WREPlay:NEXT**Syntax**

```
:RECORD:WREPlay:NEXT
```

Description

Plays back the next frame of waveforms manually.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

```
:RECORD:WREPlay:NEXT /*Plays back the next frame of waveforms manually.*/*
```

3.21.24 :RECORD:WREPlay:PLAY**Syntax**

```
:RECORD:WREPlay:PLAY <val>
```

Description

Sets to play from the start frame or end frame manually.

Parameter

| Name | Type | Range | Default |
|-------|----------|---------------|---------|
| <val> | Discrete | {FFIRst FEND} | FFIRst |

Remarks

- **FFIRst:** starts from the start frame.
- **FEND:** starts from the end frame.

Return Format

N/A

Example

```
:RECORD:WREPlay: PLAY FEND /*Sets to play from end frame manually.*/
```

3.22 :REference Commands

The **:REference** commands are used to set the reference waveform parameters.

3.22.1 :REference:SOURce

Syntax

```
:REference:SOURce <ref>,<chan>
```

```
:REference:SOURce? <ref>
```

Description

Sets or queries the source of the specified reference channel.

Parameter

| Name | Type | Range | Default |
|--------|----------|---|----------|
| <ref> | Discrete | {1 2 3 4 5 6 7 8 9 10} | - |
| <chan> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 MATH1 MATH2 MATH3 MATH4} | CHANnel1 |

Remarks

Only the currently enabled channel can be selected as the source of the specified reference channel.

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, MATH1, MATH2, MATH3, or MATH4.

Example

```
:REference:SOURce 1,CHANnel1 /*Sets the source of the reference  
channel 1 to CHANnel1.*/  
:REference:SOURce? 1 /*The query returns CHAN1.*/
```

3.22.2 :REference:VSCale

Syntax

```
:REference:VSCale <ref>,<scale>
```

```
:REference:VSCale? <ref>
```

Description

Sets or queries the vertical scale of the specified reference channel.

Parameter

| Name | Type | Range | Default |
|---------|----------|-------------------------|---------|
| <ref> | Discrete | {1 2 3 4 5 6 7 8 9 10} | - |
| <scale> | Real | Refer to <i>Remarks</i> | 50 mV |

Remarks

The range of the parameter <scale> is related to the probe ratio setting.

When the probe ratio is 1X, the value of <scale> ranges from 100 μ V to 10 V. When the probe ratio is 10X, the value of <scale> ranges from 1 mV to 100 V.

This command is only available when the reference waveform of the specified reference channel has been saved.

Return Format

The query returns the vertical scale in scientific notation.

Example

```
:REfERENCE:VSCale 1,2 /*Sets the vertical scale of reference
channel 1 to 2 V.*/
:REfERENCE:VSCale? 1 /*The query returns 2.000000E0.*/
```

3.22.3 :REfERENCE:VOFFset**Syntax**

```
:REfERENCE:VOFFset <ref>,<offset>
```

```
:REfERENCE:VOFFset? <ref>
```

Description

Sets or queries the vertical position of the specified reference channel.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|---------|
| <ref> | Discrete | {1 2 3 4 5 6 7 8 9 10} | - |
| <offset> | Real | (-10 \times RefVerticalScale) to (10 \times RefVerticalScale) | 0 V |

Remarks

RefVerticalScale indicates the vertical scale of the currently set reference channel.

Return Format

The query returns the vertical position in scientific notation.

Example

```
:REfERENCE:VOFFset 1,0.5 /*Sets the vertical offset of reference
channel 1 to 500 mV.* /
:REfERENCE:VOFFset? 1 /*The query returns 5.000000E-1.* /
```

3.22.4 :REfERENCE:RESet**Syntax**

```
:REfERENCE:RESet <ref>
```

Description

Resets the vertical scale and vertical offset of the specified reference channel to the defaults.

Parameter

| Name | Type | Range | Default |
|-------|----------|------------------------|---------|
| <ref> | Discrete | {1 2 3 4 5 6 7 8 9 10} | - |

Remarks

N/A

Return Format

N/A

Example

N/A

3.22.5 :REfERENCE:CURRent**Syntax**

```
:REfERENCE:CURRent <ref>
```

Description

Sets the current reference channel.

Parameter

| Name | Type | Range | Default |
|-------|----------|------------------------|---------|
| <ref> | Discrete | {1 2 3 4 5 6 7 8 9 10} | 1 |

Remarks

N/A

Return Format

N/A

Example

N/A

3.22.6 :REference:SAVE**Syntax**

:REference:SAVE <ref>

Description

Saves the waveform of the specified reference channel to the internal memory as the reference waveform.

Parameter

| Name | Type | Range | Default |
|-------|----------|------------------------|---------|
| <ref> | Discrete | {1 2 3 4 5 6 7 8 9 10} | - |

Remarks

N/A

Return Format

N/A

Example

N/A

3.22.7 :REference:COLor**Syntax**

:REference:COLor <ref>, <color>

:REference:COLor? <ref>

Description

Sets or queries the color of the specified reference channel.

Parameter

| Name | Type | Range | Default |
|---------|----------|------------------------------|---------|
| <ref> | Discrete | {1 2 3 4 5 6 7 8 9 10} | - |
| <color> | Discrete | {GRAY GREen BLUE RED ORANge} | - |

Remarks

N/A

Return Format

The query returns GRAY, GRE, BLUE, RED, or ORAN.

Example

```
:REfERENCE:COLor 1,GREen /*Sets the display color of the
reference channel 1 to GREen.*/
:REfERENCE:COLor? 1 /*The query returns GRE.*/
```

3.22.8 :REfERENCE:LABel:ENABLE**Syntax**

```
:REfERENCE:LABel:ENABle <bool>
```

```
:REfERENCE:LABel:ENABle?
```

Description

Enables or disables the label display of all the reference channels; or queries the on/off label display status of all the reference channels.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:REfERENCE:LABel:ENABle ON          /*Enables the label display of
all the reference channels.*/
:REfERENCE:LABel:ENABle?          /*The query returns 1.*/
```

3.22.9 :REfERENCE:LABel:CONTent**Syntax**

```
:REfERENCE:LABel:CONTent <ref>,<str>
```

```
:REfERENCE:LABel:CONTent? <ref>
```

Description

Sets or queries the label of the specified reference channel.

Parameter

| Name | Type | Range | Default |
|-------|--------------|---|---------|
| <ref> | Discrete | {1 2 3 4 5 6 7 8 9 10} | - |
| <str> | ASCII String | The label can contain English letters and numbers, as well as some symbols. | - |

Remarks

N/A

Return Format

The query returns the label of the specified reference channel in strings.

Example

```
:REfERENCE:LABel:CONTent 1,REF1    /*Sets the label of
reference channel 1 to REF1.*/
:REfERENCE:LABel:CONTent? 1        /*The query returns REF1.*/
```

3.23 :SAVE and :LOAD Commands**3.23.1 :SAVE:IMAGe:INVert****Syntax**

```
:SAVE:IMAGe:INVert <bool>
```

```
:SAVE:IMAGe:INVert?
```

Description

Enables or disables the invert function when saving the image; or queries whether the invert function is enabled when saving the image.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:SAVE:IMAGE:INVert ON          /*Enables the invert function when
saving the image.*/
:SAVE:IMAGE:INVert?           /*The query returns 1.*/
```

3.23.2 :SAVE:IMAGE:COLor**Syntax**

```
:SAVE:IMAGE:COLor <color>
```

```
:SAVE:IMAGE:COLor?
```

Description

Sets the image color for image saving to Color or Gray; or queries image color for image saving.

Parameter

| Name | Type | Range | Default |
|---------|----------|--------------|---------|
| <color> | Discrete | {COLor GRAY} | COLor |

Remarks

N/A

Return Format

The query returns COL or GRAY.

Example

```
:SAVE:IMAGe:COLor GRAY /*Sets the image color for image
saving to GRAY.*/
:SAVE:IMAGe:COLor? /*The query returns GRAY.*/
```

3.23.3 :SAVE:IMAGe:FORMat**Syntax**

```
:SAVE:IMAGe:FORMat <format>
```

```
:SAVE:IMAGe:FORMat?
```

Description

Sets or queries in what format is the image saved.

Parameter

| Name | Type | Range | Default |
|----------|----------|---------------|---------|
| <format> | Discrete | {PNG BMP JPG} | - |

Remarks

N/A

Return Format

The query returns PNG, BMP, or JPG.

Example

```
:SAVE:IMAGe:FORMat PNG /*Sets the storage format to PNG.*/
:SAVE:IMAGe:FORMat? /*The query returns PNG.*/
```

3.23.4 :SAVE:IMAGe:HEADer**Syntax**

```
:SAVE:IMAGe:HEADer <bool>
```

```
:SAVE:IMAGe:HEADer?
```

Description

Sets or queries whether to display the image header.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | - |

Remarks

- **1|ON:** enables to display the image header. If you select "ON", the instrument model and the image creation date will be displayed in the header of the image when you save the image file.
- **0|OFF:** disables to display the image header.

Return Format

The query returns 1 or 0.

Example

```
:SAVE:IMAGE:HEADer ON /*Enables the display of the image header.*/  
:SAVE:IMAGE:HEADer? /*The query returns 1.*/
```

3.23.5 :SAVE:IMAGE:DATA?

Syntax

```
:SAVE:IMAGe:DATA?
```

Description

Queries the bitmap data stream of the currently displayed image.

Parameter

N/A

Remarks

The read data format is TMC header + binary data stream of the screenshot + terminator. The TMC header is in #NXXXXXX format; wherein, # is the TMC header identifier; N following # represents the number of digits (in the decimal integer) that follow; the length of the binary data stream of the screenshot is expressed in ASCII strings, and the terminator represents the ending of communication. For example, the data read for one time is #9000387356. 9 indicates the number of digits (in the decimal integer) that follow, and "000387356" indicates the length of the binary data stream, that is, the number of bytes to be transmitted.

Return Format

The query returns the binary data stream.

Example

```
:SAVE:IMAGe:DATA? /*The query returns the binary data stream.*/
```

3.23.6 :SAVE:IMAGe

Syntax

:SAVE:IMAGe <path>

Description

Stores the contents displayed on the screen into the internal or external memory in image format. The supported image format is PNG, Windows Bitmap, and JPEG.

Parameter

| Name | Type | Range | Default |
|--------|--------------|-------------------------|---------|
| <path> | ASCII String | Refer to <i>Remarks</i> | - |

Remarks

<path> includes the file storage location and the filename with a suffix.

- The path of the local disk is C:/; and the path of the external storage device can be D:/, E:/...
- The suffix of the filename can be .bmp, .png, or .jpg.
- Refer to *:SAVE:OVERlap*. When you send the command :SAVE:OVERlap ON, if the specified storage location already contains a file with the same filename, the original file will be overwritten.
- The filename can contain letters, numbers, and other non-Chinese characters. The length of the filename shall not exceed 16 characters.

Return Format

N/A

Example

```
:SAVE:IMAGe D:/123.png /*Stores the contents displayed on the
screen into the external storage device Disk D, with the filename
123.png.*/
```

3.23.7 :SAVE:SETup

Syntax

:SAVE:SETup <path>

Description

Saves the current setup parameters of the oscilloscope to the internal or external memory as a file.

Parameter

| Name | Type | Range | Default |
|--------|--------------|-------------------------|---------|
| <path> | ASCII String | Refer to <i>Remarks</i> | - |

Remarks

<path> includes the file storage location and the filename with a suffix.

- The path of the local disk is C:/; and the path of the external storage device can be D:/, E:/...
- The suffix of the filename is "*.stp".
- Refer to *:SAVE:OVERlap*. When you send the command :SAVE:OVERlap ON, if the specified storage location already contains a file with the same filename, the original file will be overwritten.
- The filename can contain letters, numbers, and other non-Chinese characters. The length of the filename shall not exceed 16 characters.

Return Format

N/A

Example

```
:SAVE:SETup D:/123.stp /*Stores the current setup parameters of
the oscilloscope into the external storage device Disk D, with the
filename 123.stp.*/*
```

3.23.8 :SAVE:WAVEform**Syntax**

```
:SAVE:WAVEform <path>
```

Description

Saves the waveform data to the internal or external memory as a file.

Parameter

| Name | Type | Range | Default |
|--------|--------------|-------------------------|---------|
| <path> | ASCII String | Refer to <i>Remarks</i> | - |

Remarks

<path> includes the file storage location and the filename with a suffix.

- The path of the local disk is C:/; and the path of the external storage device can be D:/, E:/...
- The suffix of the filename is "*.bin" or "*.csv".
- Refer to *:SAVE:OVERlap*. When you send the command :SAVE:OVERlap ON, if the specified storage location already contains a file with the same filename, the original file will be overwritten.
- The filename can contain letters, numbers, and other non-Chinese characters. The length of the filename shall not exceed 16 characters.

Return Format

N/A

Example

```
:SAVE:WAVEform D:/123.csv /*Saves the waveform file to the
external storage device Disk D, with the filename 123.csv.*/
```

3.23.9 :SAVE:MEMory:WAVEform

Syntax

```
:SAVE:MEMory:WAVEform <path>
```

Description

Saves the memory waveform data of the oscilloscope to the specified path as a file.

Parameter

| Name | Type | Range | Default |
|--------|--------------|-------------------------|---------|
| <path> | ASCII String | Refer to <i>Remarks</i> | - |

Remarks

<path> includes the file storage location and the filename with a suffix.

- The path of the local disk is C:/; and the path of the external storage device can be D:/, E:/...
- The suffix of the filename is "*.bin", "*.csv", or "*.wfm".
- Refer to *:SAVE:OVERlap*. When you send the command :SAVE:OVERlap ON, if the specified storage location already contains a file with the same filename, the original file will be overwritten.
- The filename can contain letters, numbers, and other non-Chinese characters. The length of the filename shall not exceed 16 characters.

Return Format

N/A

Example

```
:SAVE:MEMory:WAVEform D:/123.bin /*Saves the memory waveform data of the oscilloscope to Disk D, with the filename 123.bin.*/
```

3.23.10 :SAVE:STATus?

Syntax

```
:SAVE:STATus?
```

Description

Queries the status of the memory.

Parameter

N/A

Remarks

N/A

Return Format

The query returns 0 or 1 (when the saving operation is completed).

Example

N/A

3.23.11 :SAVE:OVERlap

Syntax

```
:SAVE:OVERlap <bool>
```

:SAVE:OVERlap?

Description

Sets or queries whether to overwrite the existing file when saving a file.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | - |

Remarks

- **1|ON:** enables to overwrite the existing file. The existing file with the same name as the file to be saved will be overwritten.
- **0|OFF:** disables to overwrite the existing file.

Return Format

The query returns 1 or 0.

Example

```
:SAVE:OVERlap ON /*Enables to overwrite the existing file.*/
:SAVE:OVERlap? /*The query returns 1.*/
```

3.23.12 :SAVE:PREFix

Syntax

:SAVE:PREFix <name>

:SAVE:PREFix?

Description

Sets or queries the saved prefix of the filename.

Parameter

| Name | Type | Range | Default |
|--------|--------------|-------------------------|---------|
| <name> | ASCII String | Refer to <i>Remarks</i> | - |

Remarks

The filename mentioned in this command shall not contain the suffix. Its prefix can contain letters, numbers, and other non-Chinese characters. The length of the prefix shall not exceed 16 characters.

Return Format

The query returns the saved prefix of the filename in strings.

Example

```
:SAVE:PREFix Rigol /*Sets the saved prefix of the filename to
Rigol.*/
:SAVE:PREFix? /*The query returns Rigol.*/
```

3.23.13 :LOAD:SETup**Syntax**

```
:LOAD:SETup <path>
```

Description

Loads the setup file of the oscilloscope from the specified path.

Parameter

| Name | Type | Range | Default |
|--------|--------------|-------------------------|---------|
| <path> | ASCII String | Refer to <i>Remarks</i> | - |

Remarks

<path> includes the file storage location and the filename with a suffix.

- The path of the local disk is C:/; and the path of the external storage device can be D:/, E:/...
- The suffix of the filename to be loaded is "*.stp".

Return Format

N/A

Example

```
:LOAD:SETup D:/123.stp /*Loads the setup file "123.stp" from
the external storage device Disk D.*/
```

3.24 :SEARch Commands**3.24.1 :SEARch:COUNT?****Syntax**

```
:SEARch:COUNT?
```

Description

Queries the total number of the search events.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the total number of the search events in integer.

Example

N/A

3.24.2 :SEARch:STATe**Syntax**

```
:SEARch:STATe <bool>
```

```
:SEARch:STATe?
```

Description

Enables or disables the search function; or queries the on/off status of the search function.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Return Format

The query returns 1 or 0.

Example

```
:SEARch:STATe ON /*Enables the search function.*/
:SEARch:STATe? /*The query returns 1.*/
```

3.24.3 :SEARch:MODE**Syntax**

```
:SEARch:MODE <value>
```

```
:SEARch:MODE?
```

Description

Sets the search type.

Parameter

| Name | Type | Range | Default |
|---------|----------|--------------|---------|
| <value> | Discrete | {EDGE PULSe} | EDGE |

Remarks

- **EDGE:** selects "Edge" as the search type.
- **PULSe:** selects "Pulse" as the search type.

Return Format

The query returns EDGE or PULS.

Example

```
:SEARCH:MODE PULSe /*Selects the search type to "PULSe".*/
:SEARCH:MODE? /*The query returns PULS.*/
```

3.24.4 :SEARCH:EVENT**Syntax**

```
:SEARCH:EVENT <value>
```

```
:SEARCH:EVENT?
```

Description

Sets to navigate a search event.

Parameter

| Name | Type | Range | Default |
|---------|---------|--|---------|
| <value> | Integer | 0 to (the number of searched events – 1) | 0 |

Remarks

N/A

Return Format

The query returns an integer.

Example

```
:SEARCh:EVENT 1 /*Sets to navigate to Search Event 1.*/
:SEARCh:EVENT? /*The query returns 1.*/
```

3.24.5 :SEARCh:VALue?**Syntax**

```
:SEARCh:VALue? <X>
```

Description

Queries the time value for the specified Line No. in the marktable.

Parameter

| Name | Type | Range | Default |
|------|---------|-------|---------|
| <x> | Integer | - | - |

Remarks

<x> denotes the line number of the marktable.

Return Format

The query returns a time value.

Example

N/A

3.24.6 :SEARCh:EDGE:SLOPe**Syntax**

```
:SEARCh:EDGE:SLOPe <slope>
```

```
:SEARCh:EDGE:SLOPe?
```

Description

Sets or queries the edge for the "Edge" search type.

Parameter

| Name | Type | Range | Default |
|---------|----------|----------------------------|----------|
| <slope> | Discrete | {POSitive NEGative EITHer} | POSitive |

Remarks

- **POSitive:** indicates the rising edge.

- **NEGative:** indicates the falling edge.
- **EITHer:** indicates the rising edge or the falling edge.

Return Format

The query returns POS, NEG, or EITH.

Example

```
:SEARCh:EDGE:SLOPe NEGative /*Sets the edge type to NEGative.*/
:SEARCh:EDGE:SLOPe? /*The query returns NEG.*/
```

3.24.7 :SEARCh:EDGE:SOURce

Syntax

```
:SEARCh:EDGE:SOURce <source>
```

```
:SEARCh:EDGE:SOURce?
```

Description

Sets or queries the source for the "Edge" search type.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:SEARCh:EDGE:SOURce CHANnel1 /*Sets the source for the Edge
search type to CHANnel1.*/
:SEARCh:EDGE:SOURce? /*The query returns CHAN1.*/
```

3.24.8 :SEARCh:EDGE:THReshold

Syntax

```
:SEARCh:EDGE:THReshold <thre>
```

```
:SEARCh:EDGE:THReshold?
```

Description

Sets or queries the threshold for the "Edge" search type.

Parameter

| Name | Type | Range | Default |
|--------|------|---|---------|
| <thre> | Real | (-4.5 x VerticalScale - Offset) to (4.5 x VerticalScale - Offset) | 0.000 V |

Remarks

N/A

Return Format

The query returns the threshold for the edge search type in scientific notation.

Example

```
:SEARCH:EDGE:THreshold 0.01 /*Sets the threshold to 0.01 V.*/
:SEARCH:EDGE:THreshold? /*The query returns 1.000000E-2.*/
```

3.24.9 :SEARch:PULSe:POLarity

Syntax

```
:SEARch:PULSe:POLarity <polarity>
```

```
:SEARch:PULSe:POLarity?
```

Description

Sets or queries the polarity for the "Pulse" search type.

Parameter

| Name | Type | Range | Default |
|------------|----------|---------------------|----------|
| <polarity> | Discrete | {POSitive NEGative} | POSitive |

Remarks

N/A

Return Format

The query returns POS or NEG.

Example

```
:SEARch:PULSe:POLarity POSitive /*Sets the polarity for the Pulse search type to POSitive.*/
:SEARch:PULSe:POLarity? /*The query returns POS.*/
```

3.24.10 :SEARCh:PULSe:QUALifier

Syntax

```
:SEARCh:PULSe:QUALifier <qualifier>
```

```
:SEARCh:PULSe:QUALifier?
```

Description

Sets or queries the search condition for the "Pulse" search type.

Parameter

| Name | Type | Range | Default |
|-------------|----------|----------------------|---------|
| <qualifier> | Discrete | {GREater LESS GLESS} | GREater |

Remarks

- **GREater:** the positive/negative pulse width of the input signal is greater than the specified pulse width.
- **LESS:** the positive/negative pulse width of the input signal is smaller than the specified pulse width.
- **GLESS:** the positive/negative pulse width of the input signal is greater than the specified lower limit of pulse width and smaller than the specified upper limit of pulse width.

Return Format

The query returns GRE, LESS, or GLE.

Example

```
:SEARCh:PULSe:QUALifier LESS /*Sets the search condition for the
"Pulse" search type to LESS.*/
:SEARCh:PULSe:QUALifier? /*The query returns LESS.*/
```

3.24.11 :SEARCh:PULSe:SOURce

Syntax

```
:SEARCh:PULSe:SOURce <source>
```

```
:SEARCh:PULSe:SOURce?
```

Description

Sets or queries the source for the "Pulse" search type.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:SEARCH:PULSE:SOURce CHANnel1 /*Sets the source for the Pulse
search type to CHANnel1.*/
:SEARCH:PULSE:SOURce? /*The query returns CHAN1.*/
```

3.24.12 :SEARCH:PULSE:UWIDTH**Syntax**

```
:SEARCH:PULSE:UWIDTH <width>
```

```
:SEARCH:PULSE:UWIDTH?
```

Description

Sets or queries the upper limit of the pulse width for the "Pulse" search type.

Parameter

| Name | Type | Range | Default |
|---------|------|----------------|---------|
| <width> | Real | 800 ps to 10 s | 2 us |

Remarks

N/A

Return Format

The query returns the upper limit of the pulse width in scientific notation.

Example

```
:SEARCH:PULSE:UWIDTH 1 /*Sets the upper limit of the pulse width
for the Pulse search type to 1 s.*/
:SEARCH:PULSE:UWIDTH? /*The query returns 1.000000E0.*/
```

3.24.13 :SEARCh:PULSe:LWIDth

Syntax

```
:SEARCh:PULSe:LWIDth <width>
```

```
:SEARCh:PULSe:LWIDth?
```

Description

Sets or queries the lower limit of the pulse width for the "Pulse" search type.

Parameter

| Name | Type | Range | Default |
|---------|------|----------------|---------|
| <width> | Real | 800 ps to 10 s | 1 us |

Remarks

N/A

Return Format

The query returns the lower limit of the pulse width in scientific notation.

Example

```
:SEARCh:PULSe:LWIDth 0.2 /*Sets the lower limit of the pulse
width for the Pulse search type to 200 ms.*/
:SEARCh:PULSe:LWIDth? /*The query returns 2.000000E-1.*/
```

3.24.14 :SEARCh:PULSe:THReshold

Syntax

```
:SEARCh:PULSe:THReshold <thre>
```

```
:SEARCh:PULSe:THReshold?
```

Description

Sets or queries the threshold for the "Pulse" search type.

Parameter

| Name | Type | Range | Default |
|--------|------|--|---------|
| <thre> | Real | (-5 × VerticalScale - OFFSet) to (5 × VerticalScale - OFFSet) | 0.000 V |

Remarks

N/A

Return Format

The query returns the threshold for the pulse search type in scientific notation.

Example

```
:SEARCh:PULSe:THReshold 0.01          /*Sets the threshold to 10
mV.*/
:SEARCh:PULSe:THReshold?              /*The query returns
1.000000E-2.*/
```

3.25 :NAVigate Commands

3.25.1 :NAVigate:ENABLE

Syntax

```
:NAVigate:ENABLE <bool>
```

```
:NAVigate:ENABLE?
```

Description

Sets or queries the on/off status of the Navigation function.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:NAVigate:ENABLE ON          /*Enables the Navigation function.*/
:NAVigate:ENABLE?           /*The query returns 1.*/
```

3.25.2 :NAVigate:MODE

Syntax

```
:NAVigate:MODE <mode>
```

```
:NAVigate:MODE?
```

Description

Sets or queries the navigation mode.

Parameter

| Name | Type | Range | Default |
|--------|----------|---------------------|---------|
| <mode> | Discrete | {TIME SEARCh FRAME} | TIME |

Remarks

- **TIME:** indicates the time navigation. It is available only when in "YT" time mode.
- **SEARCh:** indicates event search navigation. After searching events using the Search function, you can select the "Search Event" mode in Navigation menu to navigate to specific events in the marktable.
- **FRAME:** indicates frame segment navigation. This mode is available only in UltraAcquire acquisition mode. When UltraAcquire is enabled, the Mode is automatically set to "Frame Segment" which cannot be modified.

The navigation function is available only when the oscilloscope is in "STOP" state (acquisition stopped). You can send the `:STOP` command to set the oscilloscope to STOP status.

Return Format

The query returns TIME, SEARCh, or FRAME.

Example

```
:NAVigate:MODE TIME /*Sets the navigation mode to TIME.*/
:NAVigate:MODE? /*The query returns TIME.*/
```

3.25.3 :NAVigate:TIME:SPEEd

Syntax

```
:NAVigate:TIME:SPEEd <speed>
```

```
:NAVigate:TIME:SPEEd?
```

Description

Sets the speed of playing the waveforms in time navigation mode.

Parameter

| Name | Type | Range | Default |
|---------|----------|-------------------|---------|
| <speed> | Discrete | {HIGH NORMAl LOW} | NORMAl |

Remarks

- **HIGH:** indicates playing the waveforms at a high speed.
- **NORMAL:** indicates playing the waveforms at a normal speed.
- **LOW:** indicates playing the waveforms at a low speed.

Return Format

The query returns HIGH, NORMAL, or LOW.

Example

```
:NAVigate:TIME:SPEEd LOW /*Sets to play the waveforms in time
navigation mode at a low speed.*/
:NAVigate:TIME:SPEEd? /*The query returns LOW.*/
```

3.25.4 :NAVigate:TIME:PLAY**Syntax**

```
:NAVigate:TIME:PLAY <bool>
```

```
:NAVigate:TIME:PLAY?
```

Description

Sets or queries whether to play the waveforms in time navigation mode.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

- **1|ON:** starts playing the waveforms.
- **0|OFF:** stops playing the waveforms.

Return Format

The query returns 0 or 1.

Example

```
:NAVigate:TIME:PLAY ON /*Sets to play the waveforms in time
navigation.*/
:NAVigate:TIME:PLAY? /*The query returns 1.*/
```


3.25.5 :NAVigate:TIME:END

Syntax

```
:NAVigate:TIME:END
```

Description

Sets to navigate to the end waveforms (waveforms ending at the rightmost end of screen) in time navigation mode.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.25.6 :NAVigate:TIME:START

Syntax

```
:NAVigate:TIME:START
```

Description

Sets to navigate to the start waveform (waveform starting at the leftmost end of screen) in time navigation mode.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.25.7 :NAVigate:TIME:NEXT

Syntax

:NAVigate:TIME:NEXT

Description

Sets to play forward the waveform in time navigation mode.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.25.8 :NAVigate:TIME:BACK

Syntax

:NAVigate:TIME:BACK

Description

Sets to play backward the waveform in time navigation mode.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.25.9 :NAVigate:SEARch:END

Syntax

:NAVigate:SEARch:END

Description

Sets to navigate to the last event.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.25.10 :NAVigate:SEARch:START

Syntax

:NAVigate:SEARch:START

Description

Sets to navigate to the first event.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.25.11 :NAVigate:SEARch:NEXT

Syntax

:NAVigate:SEARch:NEXT

Description

Sets to navigate to the next event.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.25.12 :NAVigate:SEARch:BACK

Syntax`:NAVigate:SEARch:BACK`**Description**

Sets to navigate to the previous event.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.25.13 :NAVigate:FRAMe:DISPlay:MODE

Syntax`:NAVigate:FRAMe:DISPlay:MODE <mode>``:NAVigate:FRAMe:DISPlay:MODE?`**Description**

Sets or queries the display mode in Frame Segment navigation.

Parameter

| Name | Type | Range | Default |
|--------|----------|---|---------|
| <mode> | Discrete | {ADJacent OVERlay WATERfall PERSpective MOSaic} | - |

Remarks

- **ADJacent:** indicates the adjacent display. Waveform segments are shown in adjacent display, with each segment shown next to the previous segment. In this mode, a maximum of 100 frames of waveforms can be displayed on the screen at a time.
- **OVERlay:** indicates the overlay display. All the captured waveform segments are overwritten to display as one single segment of the waveform. In this mode, a maximum of 100 frames of waveforms can be displayed on the screen at a time.
- **WATERfall:** indicates the waterfall display. It displays the captured waveform segments in a cascaded display order. In this mode, a maximum of 100 frames of waveforms can be displayed on the screen at a time.
- **PERSpective:** indicates the perspective display. The waveform segments are displayed in the ladder-like form, with each segment being arranged above another with a certain perspective (angle), moving up like a rising slope. In this mode, a maximum of 100 frames of waveforms can be displayed on the screen at a time.
- **MOSaic:** indicates the mosaic display. The whole waveform view is divided into several blocks, and each waveform segment is displayed in each block in sequence. In this mode, a maximum of 80 frames of waveforms can be displayed on the screen at a time.

Return Format

The query returns ADJ, OVER, WAT, PERS, or MOS.

Example

```
:NAVIGate:FRAME:DISPlay:MODE ADJacent /*Sets the display mode to
ADJacent in Frame Segment navigation.*/
:NAVIGate:FRAME:DISPlay:MODE? /*The query returns ADJ.*/
```

3.25.14 :NAVigate:FRAME:END:FRAME

Syntax

```
:NAVigate:FRAME:END:FRAME <frame>
```

```
:NAVigate:FRAME:END:FRAME?
```

Description

Sets or queries the end frame in Frame Segment navigation mode.

Parameter

| Name | Type | Range | Default |
|---------|---------|-------|---------|
| <frame> | Integer | - | - |

Remarks

The range of the end frame is from Start Frame to the maximum number of the frames acquired in UltraAcquire acquisition mode. You can send the `:ACQUIRE:ULTRA:MAXFRAME` command to query the maximum number of frames that can be set for UltraAcquire acquisition mode.

When you click or tap the Play key, it plays from the "Start Frame". The number of frames that can be displayed on the current screen page is (End Frame - Start Frame + 1). For example, if you set Start Frame to 3 and End Frame to 9, it will play from the 3rd frame and displays 7 frames on one screen page.

Return Format

The query returns an integer.

Example

```
:NAVigate:FRAME:END:FRAME 8 /*Sets the end frame in Frame Segment
navigation to 8.*/
:NAVigate:FRAME:END:FRAME? /*The query returns 8.*/
```

3.25.15 :NAVigate:FRAME:START:FRAME

Syntax

```
:NAVigate:FRAME:START:FRAME <frame>
```

```
:NAVigate:FRAME:START:FRAME?
```

Description

Sets or queries the start frame in Frame Segment navigation.

Parameter

| Name | Type | Range | Default |
|---------|---------|-------|---------|
| <frame> | Integer | - | - |

Remarks

When you click or tap the Play key, it plays from the "Start Frame". The number of frames that can be displayed on the current screen page is (End Frame - Start Frame + 1). For example, if you set Start Frame to 3 and End Frame to 9, it will play from the 3rd frame and displays 7 frames on one screen page.

Return Format

The query returns an integer.

Example

```
:NAVigate:FRAME:START:FRAME 3 /*Sets the start frame in Frame
Segment navigation to 3.*/
:NAVigate:FRAME:START:FRAME? /*The query returns 3.*/
```

3.25.16 :NAVigate:FRAME:END**Syntax**

```
:NAVigate:FRAME:END
```

Description

Sets to play the final page in Frame Segment navigation mode.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.25.17 :NAVigate:FRAME:START**Syntax**

```
:NAVigate:FRAME:START
```

Description

Sets to play the first page in Frame Segment navigation mode.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.25.18 :NAVigate:FRAME:NEXT

Syntax

:NAVigate:FRAME:NEXT

Description

Sets to play the next page in Frame Segment navigation mode.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.25.19 :NAVigate:FRAME:BACK

Syntax

:NAVigate:FRAME:BACK

Description

Sets to play the previous page in Frame Segment navigation mode.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.25.20 :NAVigate:FRAME:PLAY

Syntax`:NAVigate:FRAME:PLAY <bool>``:NAVigate:FRAME:PLAY?`**Description**

Sets or queries whether to play the waveforms in Frame Segment navigation mode.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

- **1|ON:** starts playing the waveforms.
- **0|OFF:** stops playing the waveforms.

During playing in Frame Segment navigation mode, you are not allowed to set the Start Frame and the End Frame.

Return Format

The query returns 1 or 0.

Example

```
:NAVigate:FRAME:PLAY ON /*Sets to play the waveforms in Frame
Segment navigatio mode.*/
:NAVigate:FRAME:PLAY? /*The query returns 1.*/
```

3.26 :SYSTem Commands

The :SYSTem commands are used to set the system-related parameters.

3.26.1 :SYSTem:AOUTput

Syntax

```
:SYSTem:AOUTput <auxoutput>
```

```
:SYSTem:AOUTput?
```

Description

Sets or queries the type of the signal output from the rear-panel [AUX OUT] connector.

Parameter

| Name | Type | Range | Default |
|-------------|----------|--------------|---------|
| <auxoutput> | Discrete | {TOUT PFAil} | TOUT |

Remarks

- **TOUT:** after you select this type, the oscilloscope initiates a trigger and then a signal which reflects the current capture rate of the oscilloscope can be output from the [AUX OUT] connector.
- **PFAil:** after you select this type, a pulse signal will be output from the [AUX OUT] connector once the oscilloscope detects a successful or failed event.

Return Format

The query returns TOUT or PFA.

Example

```
:SYSTem:AOUTput PFAil /*Sets the signal type to PFAil.*/
:SYSTem:AOUTput? /*The query returns PFA.*/
```

3.26.2 :SYSTem:BEEPer

Syntax

```
:SYSTem:BEEPer <bool>
```

```
:SYSTem:BEEPer?
```

Description

Turns on or off the beeper or queries the on/off status of the beeper.

Parameter

| Name | Type | Range | Default |
|--------|------|--------------------|---------|
| <bool> | Bool | {{1 ON}} {{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:SYSTem:BEEPer ON /*Enables the beeper.*/
:SYSTem:BEEPer? /*The query returns 1.*/
```

3.26.3 :SYSTem:DATE**Syntax**

```
:SYSTem:DATE <year>,<month>,<day>
```

```
:SYSTem:DATE?
```

Description

Sets or queries the system date.

Parameter

| Name | Type | Range | Default |
|---------|---------|-------------------------|---------|
| <year> | Integer | 1900 to 2100 | - |
| <month> | Integer | 1 to 12 | - |
| <day> | Integer | 1 to 31 (28, 29, or 30) | - |

Remarks

N/A

Return Format

The query returns the system date in strings. The year, month, and date are separated by commas.

Example

```
:SYSTem:DATE 2017,10,17 /*Sets the system date to
2017/10/17.*/
:SYSTem:DATE? /*The query returns 2017,10,17.*/
```

3.26.4 :SYSTem:ERRor[:NEXT]?

Syntax

```
:SYSTem:ERRor[:NEXT]?
```

Description

Queries and clears the error queue message.

Parameter

N/A

Remarks

N/A

Return Format

The query is in <Message Number>, <Message Content> format. Wherein, <Message Number> is an integer; <Message Content> is a double-quoted ASCII string. For example, -113,"Undefined header; command cannot be found".

Example

N/A

3.26.5 :SYSTem:RCLock

Syntax

```
:SYSTem:RCLock <clock>
```

```
:SYSTem:RCLock?
```

Description

Sets or queries the type of the 10M reference clock.

Parameter

| Name | Type | Range | Default |
|---------|----------|----------------------|---------|
| <clock> | Discrete | {OFF COUtput CINPut} | - |

Remarks

- **OFF:** disables the 10M reference clock.
- **COUtput:** indicates the clock output. The oscilloscope outputs the internal 10 MHz clock signal from the rear-panel **[10 MHz REF OUT]** connector.
- **CINPut:** indicates the clock input. The oscilloscope receives the external 10 MHz signal input from the rear-panel **[10 MHz REF IN]** connector as the clock signal.

Return Format

The query returns OFF, COUT, or CINP.

Example

```
:SYSTem:RCLock CINPut /*Sets the type of the reference clock to
CINPut.*/
:SYSTem:RCLock? /*The query returns CINP.*/
```

3.26.6 :SYSTem:GPIB**Syntax**

```
:SYSTem:GPIB <adr>
```

```
:SYSTem:GPIB?
```

Description

Sets or queries the GPIB address.

Parameter

| Name | Type | Range | Default |
|-------|---------|---------|---------|
| <adr> | Integer | 1 to 30 | 1 |

Remarks

N/A

Return Format

The query returns an integer ranging from 1 to 30.

Example

```
:SYSTem:GPIB 2 /*Sets the GPIB address to 2.*/
:SYSTem:GPIB? /*The query returns 2.*/
```

3.26.7 :SYSTem:PON**Syntax**

```
:SYSTem:PON <power_on>
```

```
:SYSTem:PON?
```

Description

Sets or queries the configuration type recalled by the oscilloscope when it is powered on again after power-off.

Parameter

| Name | Type | Range | Default |
|------------|----------|------------------|---------|
| <power_on> | Discrete | {LATest DEFAult} | DEFAult |

Remarks

N/A

Return Format

The query returns LAT or DEF.

Example

```
:SYSTem:PON LATest /*Sets the oscilloscope to recall Last
value after it is powered on again.*/
:SYSTem:PON? /*The query returns LAT.*/
```

3.26.8 :SYSTem:PSTatus**Syntax**

```
:SYSTem:PSTatus <sat>
```

```
:SYSTem:PSTatus?
```

Description

Sets or queries the power status of the oscilloscope.

Parameter

| Name | Type | Range | Default |
|-------|----------|----------------|---------|
| <sat> | Discrete | {DEFAult OPEN} | OPEN |

Remarks

- **DEFAult:** after the oscilloscope is powered on, you need to press the Power key on the front panel to start the oscilloscope.
- **OPEN:** after the oscilloscope is powered on, it starts directly. You do not have to press the Power key.

Return Format

The query returns DEF or OPEN.

Example

```
:SYSTem:PSTatus DEFAult /*Sets the power status to DEFAult.*/
:SYSTem:PSTatus? /*The query returns DEF.*/
```

3.26.9 :SYSTem:OPTion:INSTall

Syntax

```
:SYSTem:OPTion:INSTall <license>
```

Description

Installs an option.

Parameter

| Name | Type | Range | Default |
|-----------|--------------|-------------------------|---------|
| <license> | ASCII String | Refer to <i>Remarks</i> | - |

Remarks

- To install the option, first purchase the required option to obtain the key, and then use the key to obtain the option license according to the following steps.
 - Log in to the RIGOL. Click **SERVICE CENTRE > License Activation** to enter the "Registered product license code" interface.
 - In the interface, input the correct key, serial number (click or tap **Utility > About** to obtain the serial number of the instrument), and the **verification code**. Click Generate to obtain the download link of the option license file. If you need to use the option license file, click to download the file to the specified directory of the USB storage device.
- The license is a fixed length of strings. Each instrument has a unique license.

Return Format

N/A

Example

```
:SYSTem:OPTion:INSTall HDO4000-  
RLU-05@A7DEC6C1E10D42EE8E3AF0728C3D272F507E646EB54B9C97E6CCBA98468A4  
6A863FED814C24D47B8B40C894B1822660B94852E6778392281A20B54B4E723E3FD
```

3.26.10 :SYSTem:OPTion:UNINStall

Syntax

```
:SYSTem:OPTion:UNINStall
```

Description

Uninstalls all the official options.

Parameter

N/A

Remarks

After the option has been uninstalled, you need to restart the instrument.

Return Format

N/A

Example

N/A

3.26.11 :SYSTem:OPTion:STATus?

Syntax

:SYSTem:OPTion:STATus? <type>

Description

Queries whether an option is activated or not.

Parameter

| Name | Type | Range | Default |
|--------|----------|--|---------|
| <type> | Discrete | {RLU BND COMP EMBD AUTO FLEX AUDio AERO UPA BW2T4 BW2T8 BW4T8 BW7T10 BW7T20 BW10T20} | - |

Remarks

- **RLU:** memory depth upgrade option.
- **BND:** function and application bundle option. It includes AUTO, AERO, FLEX, AUDio, and UPA options.
- **COMP:** PC serial bus trigger and analysis option (RS232/UART).
- **EMBD:** embedded serial bus trigger and analysis option (I2C and SPI).
- **AUTO:** auto serial bus trigger and analysis (CAN-FD, LIN).
- **FLEX:** FlexRay serial bus trigger and analysis option (FlexRay).

- **AUDio:** audio serial bus trigger and analysis option (I2S).
- **AERO:** MIL-STD-1553 bus trigger and analysis option (MIL-STD-1553).
- **UPA:** power analysis option.
- **BW7T10|BW7T20|BW10T20:** bandwidth upgrade options supported by the DHO1000 series. BW7T10 indicates upgrading the bandwidth from 70 MHz to 100 MHz; BW7T20 indicates upgrading the bandwidth from 70 MHz to 200 MHz; BW10T20 indicates upgrading the bandwidth from 100 MHz to 200 MHz.
- **BW2T4|BW2T8|BW4T8:** bandwidth upgrade options supported by the DHO4000 series. BW2T4 indicates upgrading the bandwidth from 200 MHz to 400 MHz; BW2T8 indicates upgrading the bandwidth from 200 MHz to 800 MHz; BW4T8 indicates upgrading the bandwidth from 400 MHz to 800 MHz.

Return Format

The query returns 0 or 1.

- 0: indicates that the option is not installed.
- 1: indicates that the official option has been installed.

Example

N/A

3.26.12 :SYSTem:RAMount?

Syntax

:SYSTem:RAMount?

Description

Queries the number of analog channels of the current instrument.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the number of analog channels of the current instrument in integer. For this oscilloscope, the query returns a fixed value 4.

Example

N/A

3.26.13 :SYSTem:RESet**Syntax**`:SYSTem:RESet`**Description**

Resets the system to power on.

Parameter

N/A

Remarks

N/A

Return Format

N/A

Example

N/A

3.26.14 :SYSTem:SETup**Syntax**`:SYSTem:SETup <setup_data>``:SYSTem:SETup?`**Description**

Sends or reads the data stream of the system setup file.

Parameter

| Name | Type | Range | Default |
|--------------|--------|-------------------------|---------|
| <setup_data> | Binary | Refer to <i>Remarks</i> | - |

Remarks

- <setup_data> is a binary data block, which consists of the TMC data block header and setup data.
 - The format of the TMC data block header is #NX...X. Wherein, # is the start identifier of the data stream; the N-digit data "X...X" (N≤9) following the start identifier indicate the length of the data stream (the number of bytes).

For example, 9000002506

Wherein, N is 9, 000002506 following it represents that the data stream contains 2506 bytes of effective data.

- The setup data are expressed in ASCII format.
- When sending the command, directly place the data stream after the command string, then complete the whole sending process in one time. When reading the data stream, ensure that there is enough buffer space to receive the data stream; otherwise, errors might occur in reading the program.

Return Format

N/A

Example

N/A

3.26.15 :SYSTem:TIME

Syntax

```
:SYSTem:TIME <hours>,<minutes>,<seconds>
```

```
:SYSTem:TIME?
```

Description

Sets or queries the system time.

Parameter

| Name | Type | Range | Default |
|-----------|---------|---------|---------|
| <hours> | Integer | 0 to 23 | - |
| <minutes> | Integer | 0 to 59 | - |
| <seconds> | Integer | 0 to 59 | - |

Remarks

There is a certain delay between the return time value and the set time value due to the command response time and other factors.

Return Format

The query returns the system time in strings.

Example

```
:SYSTem:TIME 16,10,17 /*Sets the system time to 16:10:17.*/
:SYSTem:TIME? /*The query returns 16:10:17.*/
```

3.26.16 :SYSTem:LOCKed

Syntax

```
:SYSTem:LOCKed <bool>
```

```
:SYSTem:LOCKed?
```

Description

Enables or disables the front-panel key operation and touch screen operation; queries whether the front-panel key operation and touch screen operation are locked.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:SYSTem:LOCKed ON /*Disables the front-panel key operation
and touch screen operation.*/
:SYSTem:LOCKed? /*The query returns 1.*/
```

3.26.17 :SYSTem:MODules?

Syntax

```
:SYSTem:MODules?
```

Description

Queries the hardware modules.

Parameter

N/A

Remarks

N/A

Return Format

The query returns 1,0,0,0,0. The first figure indicates LA; the second figure indicates DG; and the others are not defined currently. 1 indicates available, and 0 indicates not available.

Example

N/A

3.26.18 :SYSTem:AUToscale**Syntax**`:SYSTem:AUToscale <bool>``:SYSTem:AUToscale?`**Description**

Enables or disables the function of the Auto menu; or queries the on/off status of the Auto menu.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 1 ON |

Remarks

- You can send this command or use the menu key to disable the function of the **AUTO** key.
- After the function of the **AUTO** key is disabled, you cannot perform the Auto Scale operation.

Return Format

The query returns 1 or 0.

Example

```
:SYSTem:AUToscale ON /*Enables the function of the front-panel
AUTO key.*/
:SYSTem:AUToscale? /*The query returns 1.*/
```

3.26.19 :SYSTem:GAMount?**Syntax**`:SYSTem:GAMount?`**Description**

Queries the number of grids in the horizontal direction of the screen.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the number of grids in integer. For this oscilloscope, the query returns a fixed value 10.

Example

N/A

3.26.20 :SYSTem:STIME**Syntax**

```
:SYSTem:STIME <bool>
```

```
:SYSTem:STIME?
```

Description

Sets or queries whether to display the system time.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 1 ON |

Remarks

N/A

Return Format

The query returns 0 or 1.

Example

```
:SYSTem:STIME ON /*Sets to display the system time.*/
:SYSTem:STIME? /*The query returns 1.*/
```

3.26.21 :SYSTem:VERSIon?**Syntax**

```
:SYSTem:VERSIon?
```

Description

Queries the version number of the SCPI used by the system.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the SCPI version number in strings.

Example

```
:SYSTem:VERSion? /*The query returns 3.0.*/
```

3.26.22 :SYSTem:DGSTatus?

Syntax

```
:SYSTem:DGSTatus?
```

Description

Queries whether the DG module exists.

Parameter

N/A

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:SYSTem:DGSTatus? /*The query returns 1 if the instrument has installed the DG module; otherwise, it returns 0.*/
```

3.27 :TIMebase Commands

The **:TIMebase** commands are used to set the horizontal system, for example, enable the delayed sweep, set the horizontal time base mode, etc.

3.27.1 :TIMebase:DELay:ENABLE

Syntax

```
:TIMebase:DELay:ENABLE <bool>
```

```
:TIMebase:DELay:ENABLE?
```

Description

Turns on or off the delayed sweep; or queries the on/off status of the delayed sweep.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

Delayed sweep can be used to enlarge a length of waveform horizontally to view waveform details.

Return Format

The query returns 1 or 0.

Example

```
:TIMebase:DELAy:ENABle ON /*Enables the delayed sweep.*/
:TIMebase:DELAy:ENABle? /*The query returns 1.*/
```

3.27.2 :TIMebase:DELAy:OFFSet**Syntax**

```
:TIMebase:DELAy:OFFSet <offset>
```

```
:TIMebase:DELAy:OFFSet?
```

Description

Sets or queries the offset of the delayed time base.

Parameter

| Name | Type | Range | Default |
|----------|------|--|---------|
| <offset> | Real | -(LeftTime - DelayRange/2) to (RightTime - DelayRange/2) | 0 |

Remarks

LeftTime = 5 × MainScale - MainOffset

RightTime = 5 × MainScale + MainOffset

DelayRange = 10 × DelayScale

Wherein, MainScale indicates the current main time base scale, MainOffset indicates the current main time base offset, and

DelayScale indicates the current delay time base scale.

Return Format

The query returns the offset of the delayed time base in scientific notation.

Example

```
:TIMebase:DElay:OFFSet 0.000002 /*Sets the offset of the delayed
time base to 2 μs.*/
:TIMebase:DElay:OFFSet? /*The query returns 2.000000E-6.*/
```

3.27.3 :TIMebase:DElay:SCALE**Syntax**

```
:TIMebase:DElay:SCALE <scale>
```

```
:TIMebase:DElay:SCALE?
```

Description

Sets or queries the scale of the delayed time base. The default unit is s/div.

Parameter

| Name | Type | Range | Default |
|---------|------|-------------------------|---------|
| <scale> | Real | Refer to <i>Remarks</i> | - |

Remarks

- The maximum value of the parameter <scale> is the current scale of the main time base.
- The delayed time base scale can only be the maximum value or the values acquired by reducing the maximum value at 1-2-5 step. If the minimum value calculated according to the above formula is not one of the settable values, take the minimum settable value that is greater than the minimum value calculated.
- The default value of the delayed time base scale is determined by the scale in Main mode and its value is the next scale value in Main mode.

Return Format

The query returns the scale of the delayed time base in scientific notation.

Example

```
:TIMebase:DElay:SCALE 0.00000005 /*Sets the scale of the delayed
time base to 50 ns/div.*/
:TIMebase:DElay:SCALE? /*The query returns 5.000000E-8.*/
```

3.27.4 :TIMebase[:MAIN][:OFFSet]**Syntax**

```
:TIMebase[:MAIN][:OFFSet] <offset>
```

:TIMebase[:MAIN][:OFFSet]?

Description

Sets or queries the offset of the main time base. The default unit is s.

Parameter

| Name | Type | Range | Default |
|----------|------|-------------------------|---------|
| <offset> | Real | Refer to <i>Remarks</i> | 0 |

Remarks

- In RUN state, the offset range of the main time base is as follows:
 - MainLeftTime = -5 x MainScale
 - When MainScale ≤ 10 ms, MainRightTime = 1 s
 - When 10 ms < MainScale < 10 s, MainRightTime = 100 x MainScale
 - When MainScale < 200 s and MainScale ≥ 10 s, MainRightTime = 1 ks
 - When MainScale ≥ 200s, MainRightTime = 5 x MainScale

MainRightTime indicates the maximum offset value of the main time base;
MainLeftTime indicates the minimum offset value of the main time base;
MainScale is the current scale of the main time base.
- In Stop state, the offset range of the main time base is the same as that of the memory sampling.

Return Format

The query returns the offset of the main time base in scientific notation.

Example

```
:TIMebase:MAIN:OFFSet 0.0002 /*Sets the offset of the main time
base to 200 μs.*/
:TIMebase:MAIN:OFFSet? /*The query returns 2.000000E-4.*/
```

3.27.5 :TIMebase[:MAIN]:SCALE

Syntax

:TIMebase[:MAIN]:SCALE <scale>

:TIMebase[:MAIN]:SCALE?

Description

Sets or queries the main time base scale.

Parameter

| Name | Type | Range | Default |
|---------|------|-------------------------|---------|
| <scale> | Real | Refer to <i>Remarks</i> | 2µs/div |

Remarks

The range of <scale> is related to the bandwidth of the oscilloscope. Its range affects the horizontal time base mode.

Return Format

The query returns the main time base scale in scientific notation.

Example

```
:TIMebase:MAIN:SCALE 0.0002 /*Sets the main time base scale to
200 µs/div.*/
:TIMebase:MAIN:SCALE? /*The query returns 2.000000E-4.*/
```

3.27.6 :TIMebase:MODE

Syntax

```
:TIMebase:MODE <mode>
```

```
:TIMebase:MODE?
```

Description

Sets or queries the horizontal time base mode.

Parameter

| Name | Type | Range | Default |
|--------|----------|----------------|---------|
| <mode> | Discrete | {MAIN XY ROLL} | MAIN |

Remarks

- **MAIN:** the current time base mode. When you send the command `:TIMebase:MODE MAIN`, the time base is set to YT mode.
- **XY:** indicates the XY mode. After sending the command `:TIMebase:MODE XY`, you can send the query command `:TIMebase:MODE?` and the query command returns MAIN.
- **ROLL:** When you send the command `:TIMebase:MODE ROLL`, the time base is set to Roll mode.

Return Format

N/A

Example

```
:TIMebase:MODE ROLL /*Sets the horizontal time base mode to
ROLL.*/
:TIMebase:MODE? /*The query returns ROLL.*/
```

3.27.7 :TIMebase:HREFerence:MODE**Syntax**

```
:TIMebase:HREFerence:MODE <href>
```

```
:TIMebase:HREFerence:MODE?
```

Description

Sets or queries the horizontal reference mode.

Parameter

| Name | Type | Range | Default |
|--------|----------|--------------------------|---------|
| <href> | Discrete | {CENTer LB RB TRIG USER} | CENTer |

Remarks

- **CENTer:** when the horizontal time base is modified, the waveform displayed will be expanded or compressed horizontally relative to the screen center.
- **LB:** when the horizontal time base is modified, the waveform displayed will be expanded or compressed relative to the left border of the screen.
- **RB:** when the horizontal time base is modified, the waveform displayed will be expanded or compressed relative to the right border of the screen.
- **TRIG:** when the horizontal time base is modified, the waveform displayed will be expanded or compressed horizontally relative to the trigger position.
- **USER:** when the horizontal time base is modified, the waveform displayed will be expanded or compressed horizontally relative to the user-defined reference position.

Return Format

The query returns CENT, LB, RB, TRIG, or USER.

Example

```
:TIMebase:HREFerence:MODE TRIG /*Sets the horizontal
reference mode to trigger position.*/
:TIMebase:HREFerence:MODE? /*The query returns TRIG.*/
```

3.27.8 :TIMebase:HREFerence:POSition**Syntax**

```
:TIMebase:HREFerence:POSition <pos>
```

```
:TIMebase:HREFerence:POSition?
```

Description

Sets or queries the user-defined reference position when the waveforms are expanded or compressed horizontally.

Parameter

| Name | Type | Range | Default |
|-------|---------|-------------|---------|
| <pos> | Integer | -500 to 500 | 0 |

Remarks

N/A

Return Format

The query returns an integer ranging from -500 to 500.

Example

```
:TIMebase:HREFerence:POSition 60 /*Sets the user-defined
reference position to 60.*/
:TIMebase:HREFerence:POSition? /*The query returns
60.*/
```

3.27.9 :TIMebase:VERNier**Syntax**

```
:TIMebase:VERNier <bool>
```

```
:TIMebase:VERNier?
```

Description

Enables or disables the fine adjustment function of the horizontal scale; or queries the on/off status of the fine adjustment function of the horizontal scale.

Parameter

| Name | Type | Range | Default |
|--------|------|--------------------|---------|
| <bool> | Bool | {{1 ON}} {{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 1 or 0.

Example

```
:TImEbase:VERNier ON /*Sets the fine adjustment function
of the horizontal scale to ON.*/
:TImEbase:VERNier? /*The query returns 1.*/
```

3.27.10 :TImEbase:HOTKeys**Syntax**

```
:TImEbase:HOTKeys <action>
```

Description

Sets the running status.

Parameter

| Name | Type | Range | Default |
|----------|----------|-------------------|---------|
| <action> | Discrete | {STOP RUN SINGle} | - |

Remarks

- **STOP:** stops the measurement.
- **RUN:** runs the measurement.
- **SINGle:** indicates the single trigger measurement.

Return Format

N/A

Example

```
:TImEbase:HOTKeys RUN /*Sets the operating status to RUN.*/
```

3.27.11 :TIMebase:ROLL

Syntax

```
:TIMebase:ROLL <value>
```

```
:TIMebase:ROLL?
```

Description

Sets or queries the status of the ROLL mode.

Parameter

| Name | Type | Range | Default |
|---------|---------|-------|---------|
| <value> | Integer | {0 1} | 1 |

Remarks

- **0**: disables the Auto ROLL mode.
- **1**: enables the Auto ROLL mode. The system enters the ROLL mode automatically when the time base is greater than or equal to 50 ms.

Return Format

The query returns 0 or 1.

Example

```
:TIMebase:ROLL 0 /*Disables the ROLL mode.*/
:TIMebase:ROLL? /*The query returns 0.*/
```

3.27.12 :TIMebase:XY:ENABLE

Syntax

```
:TIMebase:XY:ENABle <bool>
```

```
:TIMebase:XY:ENABle?
```

Description

Enables or disables the XY mode; or queries the on/off status of the XY mode.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

N/A

Return Format

The query returns 0 or 1.

Example

```
:TIMebase:XY:ENABle OFF /*Disables the XY mode.*/
:TIMebase:XY:ENABle? /*The query returns 0.*/
```

3.27.13 :TIMebase:XY:X**Syntax**

```
:TIMebase:XY:X <S>
```

```
:TIMebase:XY:X?
```

Description

Sets or queries the source channel of X coordinate when the horizontal time base mode is XY.

Parameter

| Name | Type | Range | Default |
|------|----------|---|----------|
| <S> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TIMebase:XY:X CHANnel3 /*Sets the source channel of X
coordinate to CHANnel3.*/
:TIMebase:XY:X? /*The query returns CHAN3.*/
```

3.27.14 :TIMebase:XY:Y**Syntax**

```
:TIMebase:XY:Y <S>
```

```
:TIMebase:XY:Y?
```


Description

Sets or queries the channel source of the Y coordinate when the horizontal time base mode is XY.

Parameter

| Name | Type | Range | Default |
|------|----------|---|----------|
| <S> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel2 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TIMebase:XY:Y CHANnel3 /*Sets the channel source of the Y
coordinate to CHANnel3.*/
:TIMebase:XY:Y? /*The query returns CHAN3.*/
```

3.27.15 :TIMebase:XY:Z**Syntax**

```
:TIMebase:XY:Z <S>
```

```
:TIMebase:XY:Z?
```

Description

Sets or queries Source Z in XY mode.

Parameter

| Name | Type | Range | Default |
|------|----------|--|---------|
| <S> | Discrete | {NONE CHANnel1 CHANnel2 CHANnel3 CHANnel4} | - |

Remarks

N/A

Return Format

The query returns NONE, CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TIMebase:XY:Z CHANnel3          /*Sets Source Z in XY mode to
CHANnel3.*/
:TIMebase:XY:Z?                  /*The query returns CHAN3.*/
```

3.28 :TRIGger Commands

The **:TRIGger** commands are used to set the trigger source type, trigger input edge type and trigger delay as well as generate a trigger event.

3.28.1 :TRIGger:MODE

Syntax

```
:TRIGger:MODE <mode>
```

```
:TRIGger:MODE?
```

Description

Sets or queries the trigger type.

Parameter

| Name | Type | Range | Default |
|--------|----------|---|---------|
| <mode> | Discrete | {EDGE PULSe SLOPe VIDeo PATTern DURation TIMEout RUNT WINDow DELay SETup NEDGe RS232 IIC SPI CAN FLEXray LIN IIS M1553} | EDGE |

Remarks

The LIN, FlexRay, IIS, and M1553 trigger functions of the DHO4000 series oscilloscope are options. These trigger commands are only available when the specified trigger option is installed. The DHO1000 series does not support FlexRay, IIS, and M1553 trigger types. Except them, other trigger types are standard configurations.

Return Format

The query returns EDGE, PULS, SLOP, VID, PATT, DUR, TIM, RUNT, WIND, DEL, SET, NEDG, RS232, IIC, SPI, CAN, FLEX, LIN, IIS, or M1553.

Example

```
:TRIGger:MODE SLOPe          /*Selects the Slope trigger.*/
:TRIGger:MODE?              /*The query returns SLOP.*/
```

3.28.2 :TRIGger:COUPLing

Syntax

```
:TRIGger:COUPLing <couple>
```

```
:TRIGger:COUPLing?
```

Description

Selects or queries the trigger coupling type.

Parameter

| Name | Type | Range | Default |
|----------|----------|---------------------------|---------|
| <couple> | Discrete | {AC DC LFReject HFReject} | DC |

Remarks

This command is only available for the Edge trigger in which the analog channel is selected as the source.

- **AC:** blocks any DC components to pass the trigger path.
- **DC:** allows DC and AC components to pass the trigger path.
- **LFReject:** blocks the DC components and rejects the low frequency components to pass the trigger path.
- **HFReject:** rejects the high frequency components to pass the trigger path.

Return Format

The query returns AC, DC, LFR, or HFR.

Example

```
:TRIGger:COUPLing LFReject /*Sets the trigger coupling type to
low frequency rejection.*/
:TRIGger:COUPLing? /*The query returns LFR.*/
```

3.28.3 :TRIGger:STATus?

Syntax

```
:TRIGger:STATus?
```

Description

Queries the current trigger status.

Parameter

N/A

Remarks

N/A

Return Format

The query returns TD, WAIT, RUN, AUTO, or STOP.

Example

N/A

3.28.4 :TRIGger:SWEEp

Syntax

```
:TRIGger:SWEEp < sweep >
```

```
:TRIGger:SWEEp?
```

Description

Sets or queries the trigger mode.

Parameter

| Name | Type | Range | Default |
|---------|----------|----------------------|---------|
| <sweep> | Discrete | {AUTO NORMal SINGle} | AUTO |

Remarks

- **AUTO:** Auto trigger. The waveforms are displayed no matter whether the trigger conditions are met.
- **NORMal:** Normal trigger. The waveforms are displayed when trigger conditions are met. If the trigger conditions are not met, the oscilloscope displays the original waveforms and waits for another trigger.
- **SINGle:** Single trigger. The oscilloscope waits for a trigger, displays the waveforms when the trigger conditions are met, and then stops.

Return Format

The query returns AUTO, NORM, or SING.

Example

```
:TRIGger:SWEep NORMal /*Selects the normal trigger mode.*/
:TRIGger:SWEep? /*The query returns NORM.*/
```

3.28.5 :TRIGger:HOLDoff**Syntax**

```
:TRIGger:HOLDoff <value>
:TRIGger:HOLDoff?
```

Description

Sets or queries the trigger holdoff time. The default unit is s.

Parameter

| Name | Type | Range | Default |
|---------|------|--------------|---------|
| <value> | Real | 8 ns to 10 s | 8 ns |

Remarks

- Trigger holdoff can be used to stably trigger complex waveforms (such as pulse waveform). Holdoff time indicates the time that the oscilloscope waits for re-arming the trigger module. The oscilloscope will not trigger before the holdoff time expires.
- Holdoff time is not available for Video trigger, Timeout trigger, Setup&Hold trigger, Nth Edge trigger, RS232 trigger, I2C trigger, SPI trigger, CAN trigger, FlexRay trigger, LIN trigger, I2S trigger, or 1553B trigger.
- The DHO1000 series oscilloscope does not support FlexRay, I2S, and 1553B trigger types.

Return Format

The query returns the trigger holdoff time in scientific notation.

Example

```
:TRIGger:HOLDoff 0.0000002 /*Sets the trigger holdoff time to 200
ns.*/
:TRIGger:HOLDoff? /*The query returns 2.000000E-7.*/
```

3.28.6 :TRIGger:NREJect

Syntax

```
:TRIGger:NREJect <bool>
```

```
:TRIGger:NREJect?
```

Description

Turns on or off noise rejection; or queries the on/off status of noise rejection.

Parameter

| Name | Type | Range | Default |
|--------|------|------------------|---------|
| <bool> | Bool | {{1 ON}}{0 OFF}} | 0 OFF |

Remarks

- Noise rejection reduces the possibility of the Noise trigger.
- This command is only available when the source is an analog channel or EXT.

Return Format

The query returns 1 or 0.

Example

```
:TRIGger:NREJect ON /*Enables the noise rejection function.*/
:TRIGger:NREJect? /*The query returns 1.*/
```

3.28.7 :TRIGger:POSition?

Syntax

```
:TRIGger:POSition?
```

Description

Queries the waveform trigger position relative to the corresponding position in the internal memory.

Parameter

N/A

Remarks

N/A

Return Format

The query returns the waveform trigger position relative to the corresponding position in the internal memory in scientific notation.

Example

```
:TRIGger:POSition? /*The query returns 0.000E+00.*/
```

3.28.8 :TRIGger:EDGE**3.28.8.1 :TRIGger:EDGE:SOURce****Syntax**

```
:TRIGger:EDGE:SOURce <source>
```

```
:TRIGger:EDGE:SOURce?
```

Description

Sets or queries the trigger source of Edge trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|--|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 ACLIne EXT} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, ACL, or EXT.

Example

```
:TRIGger:EDGE:SOURce CHANnel1 /*Sets the trigger source to  
CHANnel1.*/  
:TRIGger:EDGE:SOURce? /*The query returns CHAN1.*/
```

3.28.8.2 :TRIGger:EDGE:SLOPe**Syntax**

```
:TRIGger:EDGE:SLOPe <slope>
```

```
:TRIGger:EDGE:SLOPe?
```

Description

Sets or queries the edge type of Edge trigger.

Parameter

| Name | Type | Range | Default |
|---------|----------|--------------------------|----------|
| <slope> | Discrete | {POSitive NEGative RFAL} | POSitive |

Remarks

- **POSitive:** indicates the rising edge.
- **NEGative:** indicates the falling edge.
- **RFAL:** indicates the rising or falling edge.

Return Format

The query returns POS, NEG, or RFAL.

Example

```
:TRIGger:EDGE:SLOPe NEGative /*Sets the edge type to NEGative.*/
:TRIGger:EDGE:SLOPe? /*The query returns NEG.*/
```

3.28.8.3 :TRIGger:EDGE:LEVEL**Syntax**

```
:TRIGger:EDGE:LEVEl <level>
```

```
:TRIGger:EDGE:LEVEl?
```

Description

Sets or queries the trigger level of Edge trigger. The unit is the same as that of current amplitude of the selected source.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Analog channel: (-4.5 x VerticalScale - OFFSet) to (4.5 x VerticalScale - OFFSet) | 0 V |

Remarks

- For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

- Only when the selected source is an analog channel or external trigger, can this setting command be valid.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:EDGE:LEVel 0.16 /*Sets the trigger level to 160 mV.*/
:TRIGger:EDGE:LEVel? /*The query returns 1.600000E-1.*/
```

3.28.9 :TRIGger:PULSe

3.28.9.1 :TRIGger:PULSe:SOURce

Syntax

```
:TRIGger:PULSe:SOURce <source>
```

```
:TRIGger:PULSe:SOURce?
```

Description

Sets or queries the trigger source of Pulse trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:PULSe:SOURce CHANnel1 /*Sets the trigger source to
CHANnel1.*/
:TRIGger:PULSe:SOURce? /*The query returns CHAN1.*/
```

3.28.9.2 :TRIGger:PULSe:POLarity

Syntax

```
:TRIGger:PULSe:POLarity <polarity>
```

```
:TRIGger:PULSe:POLarity?
```

Description

Sets or queries the polarity of Pulse trigger.

Parameter

| Name | Type | Range | Default |
|------------|----------|---------------------|----------|
| <polarity> | Discrete | {POSitive NEGative} | POSitive |

Remarks

N/A

Return Format

The query returns POS or NEG.

Example

```
:TRIGger:PULSe:POLarity NEGative /*Sets the polarity of Pulse
trigger to NEGative.*/
:TRIGger:PULSe:POLarity?          /*The query returns NEG.*/
```

3.28.9.3 :TRIGger:PULSe:WHEN**Syntax**

```
:TRIGger:PULSe:WHEN <when>
```

```
:TRIGger:PULSe:WHEN?
```

Description

Sets or queries the trigger condition of Pulse trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|----------------------|---------|
| <when> | Discrete | {GREater LESS GLESs} | GREater |

Remarks

- **GREater:** triggers when the positive/negative pulse width of the input signal is greater than the specified pulse width.
- **LESS:** triggers when the positive/negative pulse width of the input signal is smaller than the specified pulse width.

- **GLEsSs:** triggers when the positive/negative pulse is greater than than the specified lower limit of pulse width and smaller than the specified upper limit of pulse width.

Return Format

The query returns GRE, LESS, or GLEs.

Example

```
:TRIGger:PULSe:WHEN LESS /*Sets the trigger condition to LESS.*/
:TRIGger:PULSe:WHEN? /*The query returns LESS.*/
```

3.28.9.4 :TRIGger:PULSe:UWIDth

Syntax

```
:TRIGger:PULSe:UWIDth <width>
```

```
:TRIGger:PULSe:UWIDth?
```

Description

Sets or queries the pulse upper limit of the Pulse trigger. The default unit is s.

Parameter

| Name | Type | Range | Default |
|---------|------|---------------------------|-----------|
| <width> | Real | Pulse lower limit to 10 s | 2 μ s |

Remarks

This command is only available when the trigger condition is set to LESS or GLEsSs. To set or query the trigger condition of the Pulse trigger, send the `:TRIGger:PULSe:WHEN` command.

When the trigger condition is set to GLEsSs, if the set pulse upper limit value is smaller than the lower limit, the lower limit will be automatically changed. You can send the `:TRIGger:PULSe:LWIDth` command to set or query the pulse lower limit value of the Pulse trigger.

Return Format

The query returns the pulse upper limit in scientific notation.

Example

```
:TRIGger:PULSe:UWIDth 0.000003 /*Sets the pulse upper limit to 3
 $\mu$ s.*/
:TRIGger:PULSe:UWIDth? /*The query returns 3.000000E-6.*/
```

3.28.9.5 :TRIGger:PULSe:LWIDth

Syntax

```
:TRIGger:PULSe:LWIDth <width>
```

```
:TRIGger:PULSe:LWIDth?
```

Description

Sets or queries the pulse lower limit of the Pulse trigger. The default unit is s.

Parameter

| Name | Type | Range | Default |
|---------|------|---------------------|-----------|
| <width> | Real | 1 ns to upper limit | 1 μ s |

Remarks

This command is only available when the trigger condition is set to GREater or GLESS. To set or query the trigger condition of the Pulse trigger, send the `:TRIGger:PULSe:WHEN` command.

When the trigger condition is set to GLESS, if the set pulse lower limit value is greater than the upper limit, the upper limit will be automatically changed. You can send the `:TRIGger:PULSe:UWIDth` command to set or query the pulse upper limit value of the Pulse trigger.

Return Format

The query returns the pulse lower limit in scientific notation.

Example

```
:TRIGger:PULSe:LWIDth 0.000003 /*Sets the pulse lower limit of
the Pulse trigger to 3  $\mu$ s.*/
:TRIGger:PULSe:LWIDth? /*The query returns 3.000000E-6.*/
```

3.28.9.6 :TRIGger:PULSe:LEVel

Syntax

```
:TRIGger:PULSe:LEVel <level>
```

```
:TRIGger:PULSe:LEVel?
```

Description

Sets or queries the trigger level of Pulse trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Analog channel: (-4.5 x VerticalScale - OFFSet) to (4.5 x VerticalScale - OFFSet) | 0 V |

Remarks

- For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.
- Only when the selected source is an analog channel or external trigger, can this setting command be valid.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:PULSe:LEVel 0.16 /*Sets the trigger level to 160 mV.*/
:TRIGger:PULSe:LEVel? /*The query returns 1.600000E-1.*/
```

3.28.10 :TRIGger:SLOPe Commands

3.28.10.1 :TRIGger:SLOPe:SOURce

Syntax

```
:TRIGger:SLOPe:SOURce <channel>
```

```
:TRIGger:SLOPe:SOURce?
```

Description

Sets or queries the trigger source of Slope trigger.

Parameter

| Name | Type | Range | Default |
|-----------|----------|---------------------------------------|----------|
| <channel> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:SLOPe:SOURce CHANnel2 /*Sets the trigger source to
CHANnel2.*/
:TRIGger:SLOPe:SOURce? /*The query returns CHAN2.*/
```

3.28.10.2 :TRIGger:SLOPe:POLarity**Syntax**

```
:TRIGger:SLOPe:POLarity <polarity>
```

```
:TRIGger:SLOPe:POLarity?
```

Description

Sets or queries the edge type of Slope trigger.

Parameter

| Name | Type | Range | Default |
|------------|----------|---------------------|----------|
| <polarity> | Discrete | {POSitive NEGative} | POSitive |

Remarks

- **POSitive:** triggers on the rising edge.
- **NEGative:** triggers on the falling edge.

Return Format

The query returns POS or NEG.

Example

```
:TRIGger:SLOPe:POLarity POSitive /*Sets the polarity of Slope
trigger to POSitive.*/
:TRIGger:SLOPe:POLarity? /*The query returns POS.*/
```

3.28.10.3 :TRIGger:SLOPe:WHEN**Syntax**

```
:TRIGger:SLOPe:WHEN <when>
```

```
:TRIGger:SLOPe:WHEN?
```

Description

Sets or queries the trigger condition of Slope trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|----------------------|---------|
| <when> | Discrete | {GREater LESS GLEsS} | GREater |

Remarks

- **GREater:** the positive slope time of the input signal is greater than the specified time.
- **LESS:** the positive slope time of the input signal is smaller than the specified time.
- **GLEsS:** the positive slope time of the input signal is greater than the specified lower time limit and smaller than the specified upper time limit.

Return Format

The query returns GRE, LESS, or GLEs.

Example

```
:TRIGger:SLOPe:WHEN LESS /*Sets the trigger condition to LESS.*/
:TRIGger:SLOPe:WHEN? /*The query returns LESS.*/
```

3.28.10.4 :TRIGger:SLOPe:TUPPer**Syntax**

```
:TRIGger:SLOPe:TUPPer <time>
```

```
:TRIGger:SLOPe:TUPPer?
```

Description

Sets or queries the upper time limit value of the Slope trigger. The default unit s.

Parameter

| Name | Type | Range | Default |
|--------|------|---------------------|-----------|
| <time> | Real | Lower limit to 10 s | 2 μ s |

Remarks

This command is only available when the trigger condition is set to LESS or GLEs. To set or query the trigger condition of the Slope trigger, send the `:TRIGger:SLOPe:WHEN` command.

When the trigger condition is set to GLEsSs, if the set upper time limit value is smaller than the lower limit, the lower limit will be automatically changed. You can use the `:TRIGger:SLOPe:TLOWer` command to set or query the lower time limit value of the Slope trigger.

Return Format

The query returns the upper time limit in scientific notation.

Example

```
:TRIGger:SLOPe:TUPPer 0.000003 /*Sets the upper time limit to 3
µs.*/
:TRIGger:SLOPe:TUPPer? /*The query returns 3.000000E-6.*/
```

3.28.10.5 :TRIGger:SLOPe:TLOWer

Syntax

```
:TRIGger:SLOPe:TLOWer <time>
```

```
:TRIGger:SLOPe:TLOWer?
```

Description

Sets or queries the lower time limit value of the Slope trigger. The default unit is s.

Parameter

| Name | Type | Range | Default |
|--------|------|---------------------|---------|
| <time> | Real | 1 ns to upper limit | 1 µs |

Remarks

This command is only available when the trigger condition is set to GREater or GLEsSs. To set or query the trigger condition of the Slope trigger, send the `:TRIGger:SLOPe:WHEN` command.

When the trigger condition is set to GLEsSs, if the set lower time limit value is greater than the upper limit, the upper limit will be automatically changed. You can send the `:TRIGger:SLOPe:TUPPer` command to set or query the upper time limit value of the Slope trigger.

Return Format

The query returns the lower time limit in scientific notation.

Example

```
:TRIGger:SLOPe:TLOWer 0.000000020 /*Sets the lower time limit
to 20 ns.*/
:TRIGger:SLOPe:TLOWer? /*The query returns 2.000000E-8.*/
```


3.28.10.6 :TRIGger:SLOPe:WINDow

Syntax

```
:TRIGger:SLOPe:WINDow <window>
```

```
:TRIGger:SLOPe:WINDow?
```

Description

Sets or queries the vertical window type of Slope trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|-------------|---------|
| <window> | Discrete | {TA TB TAB} | TA |

Remarks

- **TA:** only adjusts the upper limit of the trigger level.
- **TB:** only adjust the lower limit of the trigger level.
- **TAB:** adjusts the upper and lower limits of the trigger level at the same time.

Return Format

The query returns TA, TB, or TAB.

Example

```
:TRIGger:SLOPe:WINDow TB /*Sets the vertical window type to
TB.* /
:TRIGger:SLOPe:WINDow? /*The query returns TB.* /
```

3.28.10.7 :TRIGger:SLOPe:ALEVel

Syntax

```
:TRIGger:SLOPe:ALEVel <level>
```

```
:TRIGger:SLOPe:ALEVel?
```

Description

Sets or queries the upper limit of the trigger level of Slope trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|--|---------|
| <level> | Real | Lower limit of the trigger level to (4.5 x VerticalScale - Offset) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the upper limit of the trigger level in scientific notation.

Example

```
:TRIGger:SLOPe:ALEVel 0.16 /*Sets the upper limit of the trigger
level to 160 mV.*/
:TRIGger:SLOPe:ALEVel? /*The query returns 1.600000E-1.*/
```

3.28.10.8 :TRIGger:SLOPe:BLEVel**Syntax**

```
:TRIGger:SLOPe:BLEVel </level>
```

```
:TRIGger:SLOPe:BLEVel?
```

Description

Sets or queries the lower limit of the trigger level of Slope trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | (-4.5 x VerticalScale - VerticalOffset) to upper limit of the trigger level | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the lower limit of the trigger level in scientific notation.

Example

```
:TRIGger:SLOPe:BLEVel 0.16 /*Sets the lower limit of the trigger
level to 160 mV.*/
:TRIGger:SLOPe:BLEVel? /*The query returns 1.600000E-1.*/
```

3.28.11 :TRIGger:VIDeo Commands**3.28.11.1 :TRIGger:VIDeo:SOURce****Syntax**

```
:TRIGger:VIDeo:SOURce <source>
```

```
:TRIGger:VIDeo:SOURce?
```

Description

Sets or queries the trigger source of Video trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:VIDeo:SOURce CHANnel2 /*Sets the trigger source to
CHANnel2.*/
:TRIGger:VIDeo:SOURce? /*The query returns CHAN2.*/
```

3.28.11.2 :TRIGger:VIDeo:POLarity**Syntax**

```
:TRIGger:VIDeo:POLarity <polarity>
```

```
:TRIGger:VIDeo:POLarity?
```

Description

Sets or queries the video polarity of Video trigger.

Parameter

| Name | Type | Range | Default |
|------------|----------|---------------------|----------|
| <polarity> | Discrete | {POSitive NEGative} | POSitive |

Remarks

N/A

Return Format

The query returns POS or NEG.

Example

```
:TRIGger:VIDeo:POLarity NEGative /*Sets the video polarity to
NEGative.*/
:TRIGger:VIDeo:POLarity? /*The query returns NEG.*/
```

3.28.11.3 :TRIGger:VIDeo:MODE**Syntax**

```
:TRIGger:VIDeo:MODE <mode>
```

```
:TRIGger:VIDeo:MODE?
```

Description

Sets or queries the sync type of Video trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|----------------------------------|---------|
| <mode> | Discrete | {ODDfield EVENfield LINE ALINes} | ALINes |

Remarks

- **ODDfield:** indicates the odd field. The oscilloscope triggers on the rising edge of the first ramp waveform in the odd field. It is available when the video standard is NTSC, PAL/SECAM, or 1080i.
- **EVENfield:** indicates the even field. The oscilloscope triggers on the rising edge of the first ramp waveform in the even field. It is available when the video standard is NTSC, PAL/SECAM, or 1080i.

- **LINE:** for NTSC and PAL/SECAM video standards, the oscilloscope triggers on the specified line in the odd or even field. For 480p, 576p, 720p, 480p and 1080i video standards, the oscilloscope triggers on the specified line.
- **ALINes:** triggers on all the horizontal sync pulses.

Return Format

The query returns ODDF, EVEN, LINE, or ALIN.

Example

```
:TRIGger:VIDeo:MODE ODDField /*Sets the sync type to ODDField.*/
:TRIGger:VIDeo:MODE? /*The query returns ODDF.*/
```

3.28.11.4 :TRIGger:VIDeo:LINE

Syntax

```
:TRIGger:VIDeo:LINE <line>
```

```
:TRIGger:VIDeo:LINE?
```

Description

Sets or queries the line number when the sync type of Video trigger is set to Line.

Parameter

| Name | Type | Range | Default |
|--------|---------|-------------------------|---------|
| <line> | Integer | Refer to <i>Remarks</i> | 1 |

Remarks

- **PAL/SECAM:** 1 to 625
- **NTSC:** 1 to 525
- **480P:** 1 to 525
- **576P:** 1 to 625
- **720P60:** 1 to 750
- **720P50:** 1 to 750
- **720P30:** 1 to 750
- **720P25:** 1 to 750
- **720P24:** 1 to 750

- **1080P60:** 1 to 1125
- **1080P50:** 1 to 1125
- **1080P30:** 1 to 1125
- **1080P25:** 1 to 1125
- **1080P24:** 1 to 1125
- **1080I60:** 1 to 1125
- **1080I50:** 1 to 1125

Return Format

The query returns an integer.

Example

```
:TRIGger:VIDeo:LINE 100 /*Sets the line number to 100.*/
:TRIGger:VIDeo:LINE? /*The query returns 100.*/
```

3.28.11.5 :TRIGger:VIDeo:STANdard

Syntax

```
:TRIGger:VIDeo:STANdard <standard>
```

```
:TRIGger:VIDeo:STANdard?
```

Description

Sets or queries the video standard of Video trigger.

Parameter

| Name | Type | Range | Default |
|------------|----------|--|---------|
| <standard> | Discrete | {PALSecam NTSC 480P 576P 720P60 720P50 720P30 720P25 720P24 1080P60 1080P50 1080P30 1080P25 1080P24 1080I60 1080I50} | NTSC |

Remarks

| Video Standard | Frame Frequency (Frame) | Sweep Function | TV Scan Line |
|----------------|-------------------------|------------------|--------------|
| PALSecam | 25 | Interlaced Scan | 625 |
| NTSC | 30 | Interlaced Scan | 525 |
| 480P | 60 | Progressive Scan | 525 |
| 576P | 50 | Progressive Scan | 625 |

| Video Standard | Frame Frequency (Frame) | Sweep Function | TV Scan Line |
|----------------|-------------------------|------------------|--------------|
| 720P60 | 60 | Progressive Scan | 750 |
| 720P50 | 50 | Progressive Scan | 750 |
| 720P30 | 30 | Progressive Scan | 750 |
| 720P25 | 25 | Progressive Scan | 750 |
| 720P24 | 24 | Progressive Scan | 750 |
| 1080P60 | 60 | Progressive Scan | 1125 |
| 1080P50 | 50 | Progressive Scan | 1125 |
| 1080P30 | 30 | Progressive Scan | 1125 |
| 1080P25 | 25 | Progressive Scan | 1125 |
| 1080P24 | 24 | Progressive Scan | 1125 |
| 1080I60 | 60 | Interlaced Scan | 1125 |
| 1080I50 | 50 | Interlaced Scan | 1125 |

Return Format

The query returns PALS, NTSC, 480P, 576P, 720P60, 720P50, 720P30, 720P25, 720P24, 1080P60, 1080P50, 1080P30, 1080P25, 1080P24, 1080I60, or 1080I50.

Example

```
:TRIGger:VIDeo:STANdard NTSC /*Sets the video standard to
NTSC.*/
:TRIGger:VIDeo:STANdard? /*The query returns NTSC.*/
```

3.28.11.6 :TRIGger:VIDeo:LEVel

Syntax

```
:TRIGger:VIDeo:LEVel <level>
```

```
:TRIGger:VIDeo:LEVel?
```

Description

Sets or queries the trigger level of Video trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | (-4.5 x VerticalScale - OFFSet) to (4.5 x VerticalScale - OFFSet) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:VIDeo:LEVel 0.16 /*Sets the trigger level to 160 mV.*/
:TRIGger:VIDeo:LEVel? /*The query returns 1.600000E-1.*/
```

3.28.12 :TRIGger:PATtern**3.28.12.1 :TRIGger:PATtern:PATtern****Syntax**

```
:TRIGger:PATtern:PATtern <pch1>[,<pch2>[,<pch3>[,<pch4>]]]
```

```
:TRIGger:PATtern:PATtern?
```

Description

Sets or queries the channel pattern of Pattern trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|-------------|---------|
| <pch1> | Discrete | {H L X R F} | X |
| <pch2> | Discrete | {H L X R F} | X |
| <pch3> | Discrete | {H L X R F} | X |
| <pch4> | Discrete | {H L X R F} | X |

Remarks

- The parameter "<pch1> to <pch4>" sets the patterns of the analog channels "CHANnel1 to CHANnel4".
- In the parameter range, H indicates high level (higher than the threshold level of the channel), L indicates low level (lower than the threshold level of the channel), and X indicates omitting the channel (This channel is not used as a part of the pattern. When all the channels are set to X, the oscilloscope will not trigger.) R indicates rising edge, and F indicates falling edge.
- In the pattern, you can only specify one edge (rising edge or falling edge). If one edge item is currently defined and then another edge item is defined in

another channel in the pattern, then a prompt message "Invalid input" is displayed. Then, the latter defined edge item will be replaced by X.

Return Format

The query returns the currently set pattern of all the channels. The channels are separated by commas.

Example

```
:TRIGger:PATtern:PATtern H,R,L,X /*Sets the patterns of "CHANnel1
to CHANnel4" to H,R,L,X.*/
:TRIGger:PATtern:PATtern? /*The query returns H,R,L,X.*/
```

3.28.12.2 :TRIGger:PATtern:SOURce

Syntax

```
:TRIGger:PATtern:SOURce <source>
```

```
:TRIGger:PATtern:SOURce?
```

Description

Sets or queries the trigger source of Pattern trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:PATtern:SOURce CHANnel2 /*Sets the trigger source to
CHANnel2.*/
:TRIGger:PATtern:SOURce? /*The query returns CHAN2.*/
```

3.28.12.3 :TRIGger:PATtern:LEVel

Syntax

```
:TRIGger:PATtern:LEVel <source>,<level>
```

```
:TRIGger:PATtern:LEVel? <level>
```

Description

Sets or queries the trigger level of the specified channel in Pattern trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|----------|----------|--|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |
| <level> | Real | (-4.5 x VerticalScale - OFFSet) to (4.5 x VerticalScale - OFFSet) | 0 |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:PATtern:LEVel CHANnel2,0.16 /*Sets the trigger level of
CHANnel2 to 160 mV.*/
:TRIGger:PATtern:LEVel? CHANnel2 /*The query returns
1.600000E-1.*/
```

3.28.13 :TRIGger:DURation**3.28.13.1 :TRIGger:DURation:SOURce****Syntax**

```
:TRIGger:DURation:SOURce <source>
```

```
:TRIGger:DURation:SOURce?
```

Description

Sets or queries the trigger source of Duration trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:DURation:SOURce CHANnel2 /*Sets the trigger source to
CHANnel2.*/
:TRIGger:DURation:SOURce? /*The query returns CHAN2.*/
```

3.28.13.2 :TRIGger:DURation:TYPE**Syntax**

```
:TRIGger:DURation:TYPE <pch1>[,<pch2>[,<pch3>[,<pch4>]]]
```

```
:TRIGger:DURation:TYPE?
```

Description

Sets or queries the channel pattern of Duration trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|---------|---------|
| <pch1> | Discrete | {H L X} | X |
| <pch2> | Discrete | {H L X} | X |
| <pch3> | Discrete | {H L X} | X |
| <pch4> | Discrete | {H L X} | X |

Remarks

- The parameter "<pch1> to <pch4>" sets the patterns of the analog channels "CHANnel1 to CHANnel4".
- In the parameter range, H indicates high level (higher than the threshold level of the channel), L indicates low level (lower than the threshold level of the channel), and X indicates omitting the channel (This channel is not used as a part of the pattern. When all the channels are set to X, the oscilloscope will not trigger.)

Return Format

The query returns the currently set pattern of all the channels. The channels are separated by commas.

Example

```
:TRIGger:DURation:TYPE L,X,H,L /*Sets the patterns of "CHANnel1 to
CHANnel4" to L,X,H,L.*/
:TRIGger:DURation:TYPE? /*The query returns L,X,H,L.*/
```

3.28.13.3 :TRIGger:DURation:WHEN

Syntax

```
:TRIGger:DURation:WHEN <when>
:TRIGger:DURation:WHEN?
```

Description

Sets or queries the trigger condition of Duration trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------------------------|---------|
| <when> | Discrete | {GREater LESS GLESs UNGLess} | GREater |

Remarks

- **GREater:** triggers when the set duration time of the pattern is greater than the preset time.
- **LESS:** triggers when the set duration time of the pattern is smaller than the preset time.
- **GLESs:** triggers when the set duration time of the pattern is smaller than the preset upper time limit and greater than the preset lower time limit.
- **UNGLess:** triggers when the set duration time of the pattern is greater than the preset upper time limit or smaller than the preset lower time limit.

Return Format

The query returns GRE, LESS, GLES, or UNGL.

Example

```
:TRIGger:DURation:WHEN LESS /*Sets the trigger condition to
LESS.*/
:TRIGger:DURation:WHEN? /*The query returns LESS.*/
```

3.28.13.4 :TRIGger:DURation:TUPPer

Syntax

```
:TRIGger:DURation:TUPPer <time>
:TRIGger:DURation:TUPPer?
```

Description

Sets or queries the upper limit of the duration time of Duration trigger. The default unit is s.

Parameter

| Name | Type | Range | Default |
|--------|------|-----------------|-----------|
| <time> | Real | 1.01 ns to 10 s | 1 μ s |

Remarks

This command is only available when the trigger condition is set to LESS, GLEs, or UNGLess. To set or query the trigger condition of the Duration trigger, send the *:TRIGger:DURation:WHEN* command.

When the trigger condition is set to GLEs or UNGLess, if the set upper limit of the duration time value is smaller than the lower limit, the lower limit will be automatically changed. You can send the *:TRIGger:DURation:TLOWer* command to set or query the lower limit of the duration time value of the Duration trigger.

Return Format

The query returns the upper limit of the duration time in scientific notation.

Example

```
:TRIGger:DURation:TUPPer 0.000003 /*Sets the upper limit of the
duration time to 3  $\mu$ s.*/
:TRIGger:DURation:TUPPer? /*The query returns
3.000000E-6.*/
```

3.28.13.5 :TRIGger:DURation:TLOWer

Syntax

```
:TRIGger:DURation:TLOWer <time>
:TRIGger:DURation:TLOWer?
```

Description

Sets or queries the lower limit of the duration time of Duration trigger. The default unit is s.

Parameter

| Name | Type | Range | Default |
|--------|------|---------------|-----------|
| <time> | Real | 1 ns to 9.9 s | 1 μ s |

Remarks

This command is only available when the trigger condition is set to GREater, GLESS, or UNGLess. To set or query the trigger condition of the Duration trigger, send the *:TRIGger:DURation:WHEN* command.

When the trigger condition is set to GLESS or UNGLess, if the set lower limit of the duration time value is greater than the upper limit, the upper limit will be automatically changed. You can send the *:TRIGger:DURation:TUPPer* command to set or query the upper limit of the duration time value of the Duration trigger.

Return Format

The query returns the lower limit of the duration time in scientific notation.

Example

```
:TRIGger:DURation:TLOWer 0.000003 /*Sets the lower limit of the
duration time to 3  $\mu$ s.*/
:TRIGger:DURation:TLOWer? /*The query returns
3.000000E-6.*/
```

3.28.13.6 :TRIGger:DURation:LEVel**Syntax**

```
:TRIGger:DURation:LEVel <source>,<level>
```

```
:TRIGger:DURation:LEVel?<source>
```

Description

Sets or queries the trigger level of the specified channel in Duration trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |
| <level> | Real | Analog channel: (-4.5 x VerticalScale - OFFSet) to (4.5 x VerticalScale - OFFSet) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:DURation:LEVel CHANnel2,0.16 /*Sets the trigger level of
CH2 to 160 mV.*/
:TRIGger:DURation:LEVel? CHANnel2 /*The query returns
1.600000E-1.*/
```

3.28.14 :TRIGger:TIMEout

3.28.14.1 :TRIGger:TIMEout:SOURce

Syntax

```
:TRIGger:TIMEout:SOURce <source>
```

```
:TRIGger:TIMEout:SOURce?
```

Description

Sets or queries the trigger source of Timeout trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:TIMEout:SOURce CHANnel2 /*Sets the trigger source to
CHANnel2.*/
:TRIGger:TIMEout:SOURce? /*The query returns CHAN2.*/
```

3.28.14.2 :TRIGger:TIMEout:SLOPe

Syntax

```
:TRIGger:TIMEout:SLOPe <slope>
```

:TRIGger:TIMEout:SLOPe?

Description

Sets or queries the edge type of Timeout trigger.

Parameter

| Name | Type | Range | Default |
|---------|----------|---------------------------|----------|
| <slope> | Discrete | {POSitive NEGative RFALI} | POSitive |

Remarks

- **POSitive:** starts timing when the rising edge of the input signal passes through the trigger level.
- **NEGative:** starts timing when the falling edge of the input signal passes through the trigger level.
- **RFALI:** starts timing when any edge of the input signal passes through the trigger level.

Return Format

The query returns POS, NEG, or RFAL.

Example

```
:TRIGger:TIMEout:SLOPe NEGative /*Sets the edge type to
NEGative.*/
:TRIGger:TIMEout:SLOPe? /*The query returns NEG.*/
```

3.28.14.3 :TRIGger:TIMEout:TIME

Syntax

:TRIGger:TIMEout:TIME <time>

:TRIGger:TIMEout:TIME?

Description

Sets or queries the timeout value of Timeout trigger. The default unit is s.

Parameter

| Name | Type | Range | Default |
|--------|------|--------------|-----------|
| <time> | Real | 1 ns to 10 s | 1 μ s |

Remarks

N/A

Return Format

The query returns the timeout value in scientific notation.

Example

```
:TRIGger:TIMEout:TIME 0.002 /*Sets the timeout value to 2 ms.*/
:TRIGger:TIMEout:TIME? /*The query returns 2.000000E-3.*/
```

3.28.14.4 :TRIGger:TIMEout:LEVel**Syntax**

```
:TRIGger:TIMEout:LEVel <level>
```

```
:TRIGger:TIMEout:LEVel?
```

Description

Sets or queries the trigger level of Timeout trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Analog channel: (-4.5 x VerticalScale - OFFSet) to (4.5 x VerticalScale - OFFSet) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:TIMEout:LEVel 0.16 /*Sets the trigger level to 160 mV.*/
:TRIGger:TIMEout:LEVel? /*The query returns 1.600000E-1.*/
```

3.28.15 :TRIGger:RUNT

3.28.15.1 :TRIGger:RUNT:SOURce

Syntax

```
:TRIGger:RUNT:SOURce <source>
```

```
:TRIGger:RUNT:SOURce?
```

Description

Sets or queries the trigger source of Runt trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:RUNT:SOURce CHANnel2 /*Sets the trigger source to  
CHANnel2.*/  
:TRIGger:RUNT:SOURce? /*The query returns CHAN2.*/
```

3.28.15.2 :TRIGger:RUNT:POLarity

Syntax

```
:TRIGger:RUNT:POLarity <polarity>
```

```
:TRIGger:RUNT:POLarity?
```

Description

Sets or queries the polarity of Runt trigger.

Parameter

| Name | Type | Range | Default |
|------------|----------|---------------------|----------|
| <polarity> | Discrete | {POSitive NEGative} | POSitive |

Remarks

- **POSitive:** indicates the positive polarity. The oscilloscope triggers on the positive polarity of Runt trigger.
- **NEGative:** indicates the negative polarity. The oscilloscope triggers on the negative polarity of Runt trigger.

Return Format

The query returns POS or NEG.

Example

```
:TRIGger:RUNT:POLarity NEGative /*Sets the polarity of Runt
trigger to NEGative.*/
:TRIGger:RUNT:POLarity? /*The query returns NEG.*/
```

3.28.15.3 :TRIGger:RUNT:WHEN**Syntax**

```
:TRIGger:RUNT:WHEN <when>
```

```
:TRIGger:RUNT:WHEN?
```

Description

Sets or queries the trigger conditions of Runt trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|--------------------------|---------|
| <when> | Discrete | {NONE GREater LESS GLEs} | NONE |

Remarks

- **NONE:** indicates not setting the trigger condition of Runt trigger.
- **GREater:** triggers when the runt pulse width is greater than the lower limit of pulse width.
- **LESS:** triggers when the runt pulse width is smaller than the upper limit of pulse width.
- **GLEs:** triggers when the runt pulse width is greater than the lower limit and smaller than the upper limit of pulse width.

The lower limit of the pulse width must be smaller than the upper limit.

Return Format

The query returns NONE, GRE, LESS, or GLES.

Example

```
:TRIGger:RUNT:WHEN LESS /*Sets the trigger condition of Runt
trigger to LESS.*/
:TRIGger:RUNT:WHEN? /*The query returns LESS.*/
```

3.28.15.4 :TRIGger:RUNT:WUPPer**Syntax**

```
:TRIGger:RUNT:WUPPer <width>
```

```
:TRIGger:RUNT:WUPPer?
```

Description

Sets or queries the upper limit of the pulse width of Runt trigger. The default unit is s.

Parameter

| Name | Type | Range | Default |
|---------|------|-----------------|-----------|
| <width> | Real | 1.01 ns to 10 s | 2 μ s |

Remarks

This command is only available when the trigger condition is set to LESS or GLEs. To set or query the trigger condition of the Runt trigger, send the `:TRIGger:RUNT:WHEN` command.

When the trigger condition is set to GLEs, if the set upper limit of the pulse width is smaller than the lower limit, the lower limit will be automatically changed. You can send the `:TRIGger:RUNT:WLOWer` command to set or query the lower limit of the pulse width of Runt trigger.

Return Format

The query returns the upper limit of the pulse width in scientific notation.

Example

```
:TRIGger:RUNT:WUPPer 0.02 /*Sets the upper limit of the pulse
width to 20 ms.*/
:TRIGger:RUNT:WUPPer? /*The query returns 2.000000E-2.*/
```

3.28.15.5 :TRIGger:RUNT:WLOWer**Syntax**

```
:TRIGger:RUNT:WLOWer <width>
```

:TRIGger:RUNT:WLOWer?

Description

Sets or queries the lower limit of the pulse width of Runt trigger. The default unit is s.

Parameter

| Name | Type | Range | Default |
|---------|------|---------------|-----------|
| <width> | Real | 1 ns to 9.9 s | 1 μ s |

Remarks

This command is only available when the trigger condition is set to GREater or GLESS. To set or query the trigger condition of the Runt trigger, send the *:TRIGger:RUNT:WHEN* command.

When the trigger condition is set to GLESS, if the set lower limit of the pulse width is greater than the lower limit, the upper limit will be automatically changed. You can send the *:TRIGger:RUNT:WUPPer* command to set or query the upper limit of the pulse width of Runt trigger.

Return Format

The query returns the lower limit of the pulse width in scientific notation.

Example

```
:TRIGger:RUNT:WLOWer 0.01 /*Sets the lower limit of the pulse
width to 10 ms.*/
:TRIGger:RUNT:WLOWer? /*The query returns 1.000000E-2.*/
```

3.28.15.6 :TRIGger:RUNT:ALEVel

Syntax

:TRIGger:RUNT:ALEVel <level>

:TRIGger:RUNT:ALEVel?

Description

Sets or queries the upper limit of the trigger level of Runt trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|--|---------|
| <level> | Real | Lower limit of the trigger level to (4.5 x VerticalScale - Offset) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the upper limit of the trigger level in scientific notation.

Example

```
:TRIGger:RUNT:ALEVel 0.16 /*Sets the upper limit of the trigger
level to 160 mV.*/
:TRIGger:RUNT:ALEVel? /*The query returns 1.600000E-1.*/
```

3.28.15.7 :TRIGger:RUNT:BLEVel**Syntax**

```
:TRIGger:RUNT:BLEVel <level>
```

```
:TRIGger:RUNT:BLEVel?
```

Description

Sets or queries the lower limit of the trigger level of Runt trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | (-4.5 x VerticalScale - Offset) to upper limit of the trigger level | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the lower limit of the trigger level in scientific notation.

Example

```
:TRIGger:RUNT:BLEVel 0.16 /*Sets the lower limit of the trigger
level to 160 mV.*/
:TRIGger:RUNT:BLEVel? /*The query returns 1.600000E-1.*/
```

3.28.16 :TRIGger:WINDows

3.28.16.1 :TRIGger:WINDows:SOURce

Syntax

```
:TRIGger:WINDows:SOURce <source>
```

```
:TRIGger:WINDows:SOURce?
```

Description

Sets or queries the trigger source of Window trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:WINDows:SOURce CHANnel2 /*Sets the trigger source to  
CHANnel2.*/  
:TRIGger:WINDows:SOURce? /*The query returns CHAN2.*/
```

3.28.16.2 :TRIGger:WINDows:SLOPe

Syntax

```
:TRIGger:WINDows:SLOPe <type>
```

```
:TRIGger:WINDows:SLOPe?
```

Description

Sets or queries the edge type of Windows trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|---------------------------|----------|
| <type> | Discrete | {POSitive NEGative RFALI} | POSitive |

Remarks

- **POSitive:** triggers on the rising edge of the input signal when the voltage level is greater than the preset high trigger level.
- **NEGative:** triggers on the falling edge of the input signal when the voltage level is smaller than the preset low trigger level.
- **RFALL:** triggers on either the rising or falling edge of the input signal when the voltage level meets the preset trigger level.

Return Format

The query returns POS, NEG, or RFAL.

Example

```
:TRIGger:WINDows:SLOPe NEGative /*Sets the edge type of Windows
trigger to NEGative.*/
:TRIGger:WINDows:SLOPe? /*The query returns NEG.*/
```

3.28.16.3 :TRIGger:WINDows:POSition**Syntax**

```
:TRIGger:WINDows:POSition <pos>
```

```
:TRIGger:WINDows:POSition?
```

Description

Sets or queries the trigger position of Window trigger.

Parameter

| Name | Type | Range | Default |
|-------|----------|-------------------|---------|
| <pos> | Discrete | {EXIT ENTER TIME} | ENTER |

Remarks

- **EXIT:** triggers when the input signal exits the specified trigger level range.
- **ENTER:** triggers when the input signal enters the specified trigger level range.
- **TIME:** triggers when the accumulated hold time after the trigger signal enters the specified trigger level range is equal to the window time.

Return Format

The query returns EXIT, ENT, or TIME.

Example

```
:TRIGger:WINDows:POSition ENTer /*Sets the trigger position to
ENT.*/
:TRIGger:WINDows:POSition? /*The query returns ENT.*/
```

3.28.16.4 :TRIGger:WINDows:TIME**Syntax**

```
:TRIGger:WINDows:TIME <time>
```

```
:TRIGger:WINDows:TIME?
```

Description

Sets or queries the window time of Window trigger.

Parameter

| Name | Type | Range | Default |
|--------|------|--------------|-----------|
| <time> | Real | 1 ns to 10 s | 1 μ s |

Remarks

N/A

Return Format

The query returns the window time in scientific notation.

Example

```
:TRIGger:WINDows:TIME 0.002 /*Sets the window time to 2 ms.*/
:TRIGger:WINDows:TIME? /*The query returns 2.000000E-3.*/
```

3.28.16.5 :TRIGger:WINDows:ALEVEL**Syntax**

```
:TRIGger:WINDows:ALEVEL <level>
```

```
:TRIGger:WINDows:ALEVEL?
```

Description

Sets or queries the upper limit of the trigger level of Window trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Lower limit to (4.5 x VerticalScale - OFFSet) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the upper limit of the trigger level in scientific notation.

Example

```
:TRIGger:WINDows:ALEVel 0.16 /*Sets the upper limit of the
trigger level to 160 mV.*/
:TRIGger:WINDows:ALEVel? /*The query returns 1.600000E-1.*/
```

3.28.16.6 :TRIGger:WINDows:BLEVel**Syntax**

```
:TRIGger:WINDows:BLEVel </level>
```

```
:TRIGger:WINDows:BLEVel?
```

Description

Sets or queries the lower limit of the trigger level of Window trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|--|---------|
| <level> | Real | (-4.5 x VerticalScale - OFFSet) to upper limit | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the lower limit of the trigger level in scientific notation.

Example

```
:TRIGger:WINDows:BLEVel 0.05 /*Sets the lower limit of the
trigger level to 50 mV.*/
:TRIGger:WINDows:BLEVel? /*The query returns 5.000000E-2.*/
```

3.28.17 :TRIGger:DElay**3.28.17.1 :TRIGger:DElay:SA****Syntax**

```
:TRIGger:DElay:SA <source>
```

```
:TRIGger:DElay:SA?
```

Description

Sets or queries the trigger source of Source A in Delay trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:DElay:SA CHANnel2 /*Sets trigger source A to
CHANnel2.*/
:TRIGger:DElay:SA? /*The query returns CHAN2.*/
```

3.28.17.2 :TRIGger:DElay:ASLop**Syntax**

```
:TRIGger:DElay:ASLop <slope>
```

```
:TRIGger:DElay:ASLop?
```

Description

Sets or queries the edge type of Edge A in Delay trigger.

Parameter

| Name | Type | Range | Default |
|---------|----------|---------------------|----------|
| <slope> | Discrete | {POSitive NEGative} | POSitive |

Remarks

N/A

Return Format

The query returns POS or NEG.

Example

```
:TRIGger:DElay:ASLop NEGative /*Sets the edge type of Edge A to
NEGative.*/
:TRIGger:DElay:ASLop? /*The query returns NEG.*/
```

3.28.17.3 :TRIGger:DElay:SB**Syntax**

```
:TRIGger:DElay:SB <source>
```

```
:TRIGger:DElay:SB?
```

Description

Sets or queries the trigger source of Source B in Delay trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|--|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel2 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:DElay:SB CHANnel4 /*Sets trigger source B to
CHANnel4.*/
:TRIGger:DElay:SB? /*The query returns CHAN4.*/
```

3.28.17.4 :TRIGger:DElay:BSLop

Syntax

```
:TRIGger:DElay:BSLop <slope>
```

```
:TRIGger:DElay:BSLop?
```

Description

Sets or queries the edge type of Edge B in Delay trigger.

Parameter

| Name | Type | Range | Default |
|---------|----------|---------------------|----------|
| <slope> | Discrete | {POSitive NEGative} | POSitive |

Remarks

N/A

Return Format

The query returns POS or NEG.

Example

```
:TRIGger:DElay:BSLop NEGative /*Sets the edge type of Edge B to
NEGative.*/
:TRIGger:DElay:BSLop? /*The query returns NEG.*/
```

3.28.17.5 :TRIGger:DElay:TYPE

Syntax

```
:TRIGger:DElay:TYPE <type>
```

```
:TRIGger:DElay:TYPE?
```

Description

Sets or queries the trigger condition of the Delay trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|---------------------------|---------|
| <type> | Discrete | {GREater LESS GLESS GOUT} | GREater |

Remarks

- **GREater:** triggers when the time difference (ΔT) between the specified edges of Source A and Source B is greater than the preset time limit.
- **LESS:** triggers when the time difference (ΔT) between the specified edges of Source A and Source B is smaller than the preset time limit.
- **GLEs:** triggers when the time difference (ΔT) between the specified edges of Source A and Source B is greater than the lower limit of the preset time and smaller than the upper limit of the preset time.
- **GOUT:** triggers when the time difference (ΔT) between the specified edges of Source A and Source B is smaller than the lower limit of the preset time or greater than the upper limit of the preset time.

Return Format

The query returns GRE, LESS, GLEs, or GOUT.

Example

```
:TRIGger:DElay:TYPe GOUT /*Sets the trigger condition to GOUT.*/
:TRIGger:DElay:TYPe? /*The query returns GOUT.*/
```

3.28.17.6 :TRIGger:DElay:TUPPer**Syntax**

```
:TRIGger:DElay:TUPPer <time>
```

```
:TRIGger:DElay:TUPPer?
```

Description

Sets or queries the upper limit of delay time of the Delay trigger. The default unit is s.

Parameter

| Name | Type | Range | Default |
|--------|------|-----------------|-----------|
| <time> | Real | 1.01 ns to 10 s | 2 μ s |

Remarks

This command is only available when the trigger condition is set to LESS, GLEs, or GOUT. To set or query the trigger condition of the Delay trigger, send the `:TRIGger:DElay:TYPe` command.

When the trigger condition is set to GLEsS or GOuT, if the set upper limit of the delay time is smaller than the lower limit, the lower limit will be automatically changed. You can send the `:TRIGger:DElay:TLOWer` command to set or query the lower limit of the delay time of the Delay trigger.

Return Format

The query returns the upper limit of delay time in scientific notation.

Example

```
:TRIGger:DElay:TUPPer 0.002 /*Sets the upper limit of delay time
to 2 ms.*/
:TRIGger:DElay:TUPPer? /*The query returns 2.000000E-3.*/
```

3.28.17.7 :TRIGger:DElay:TLOWer

Syntax

```
:TRIGger:DElay:TLOWer <time>
```

```
:TRIGger:DElay:TLOWer?
```

Description

Sets or queries the lower limit of delay time of the Delay trigger. The default unit is s.

Parameter

| Name | Type | Range | Default |
|--------|------|---------------|---------|
| <time> | Real | 1 ns to 9.9 s | 1µs |

Remarks

This command is only available when the trigger condition is set to GREater, GLEsS, or GOuT. To set or query the trigger condition of the Delay trigger, send the `:TRIGger:DElay:TYPE` command.

When the trigger condition is set to GLEsS or GOuT, if the set lower limit of the delay time is greater than the upper limit, the upper limit will be automatically changed. You can send the `:TRIGger:DElay:TUPPer` command to set or query the upper limit of the delay time of the Delay trigger.

Return Format

The query returns the lower limit of delay time in scientific notation.

Example

```
:TRIGger:DElay:TLOWer 0.002 /*Sets the lower limit of delay time
to 2 ms.*/
:TRIGger:DElay:TLOWer? /*The query returns 2.000000E-3.*/
```

3.28.17.8 :TRIGger:DElay:ALEvel

Syntax

```
:TRIGger:DElay:ALEvel <level>
```

```
:TRIGger:DElay:ALEvel?
```

Description

Sets or queries the threshold level of Source A of Delay trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | (-4.5 x VerticalScale - Offset) to (4.5 x VerticalScale - Offset) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the threshold level of Source A in scientific notation.

Example

```
:TRIGger:DElay:ALEvel 0.16 /*Sets the threshold level of Source
A to 160 mV.*/
:TRIGger:DElay:ALEvel? /*The query returns 1.600000E-1.*/
```

3.28.17.9 :TRIGger:DElay:BLEvel

Syntax

```
:TRIGger:DElay:BLEvel <level>
```

```
:TRIGger:DElay:BLEvel?
```

Description

Sets or queries the threshold level of Source B of Delay trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | (-4.5 x VerticalScale - Offset) to (4.5 x VerticalScale - Offset) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the threshold level of Source B in scientific notation.

Example

```
:TRIGger:DElay BLEvel 0.05 /*Sets the threshold level of Source
B to 50 mV.*/
:TRIGger:DElay:BLEvel? /*The query returns 5.000000E-2.*/
```

3.28.18 :TRIGger:SHOLd**3.28.18.1 :TRIGger:SHOLd:DSRC****Syntax**

```
:TRIGger:SHOLd:DSRC <source>
```

```
:TRIGger:SHOLd:DSRC?
```

Description

Sets or queries the data source of Setup&Hold trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel2 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:SHOLd:DSRC CHANnel1 /*Sets the data source to
CHANnel1.*/
:TRIGger:SHOLd:DSRC? /*The query returns CHAN1.*/
```

3.28.18.2 :TRIGger:SHOLd:CSRC**Syntax**

```
:TRIGger:SHOLd:CSRC <source>
```

:TRIGger:SHOLd:CSRC?

Description

Sets or queries the clock source of Setup&Hold trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---------------------------------------|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:SHOLd:CSRC CHANnel2 /*Sets the clock source to
CHANnel2.*/
:TRIGger:SHOLd:CSRC? /*The query returns CHAN2.*/
```

3.28.18.3 :TRIGger:SHOLd:SLOPe

Syntax

:TRIGger:SHOLd:SLOPe <slope>

:TRIGger:SHOLd:SLOPe?

Description

Sets or queries the edge type of Setup&Hold trigger.

Parameter

| Name | Type | Range | Default |
|---------|----------|---------------------|----------|
| <slope> | Discrete | {POSitive NEGative} | POSitive |

Remarks

N/A

Return Format

The query returns POS or NEG.

Example

```
:TRIGger:SHOLd:SLOPe NEGative /*Sets the edge type to
NEGative.*/
:TRIGger:SHOLd:SLOPe? /*The query returns NEG.*/
```

3.28.18.4 :TRIGger:SHOLd:PATtern**Syntax**

```
:TRIGger:SHOLd:PATtern <pattern>
```

```
:TRIGger:SHOLd:PATtern?
```

Description

Sets or queries the data type of Setup&Hold trigger.

Parameter

| Name | Type | Range | Default |
|-----------|----------|-------|---------|
| <pattern> | Discrete | {H L} | H |

Remarks

- **H:** indicates high level.
- **L:** indicates low level.

Return Format

The query returns H or L.

Example

```
:TRIGger:SHOLd:PATtern L /*Sets data type to L.*/
:TRIGger:SHOLd:PATtern? /*The query returns L.*/
```

3.28.18.5 :TRIGger:SHOLd:TYPE**Syntax**

```
:TRIGger:SHOLd:TYPE <type>
```

```
:TRIGger:SHOLd:TYPE?
```

Description

Sets or queries the trigger condition of Setup/Hold trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|----------------------|---------|
| <type> | Discrete | {SETup HOLD SETHold} | SETup |

Remarks

- **SETup:** the oscilloscope triggers when the setup time is smaller than the specified setup time.
- **HOLD:** the oscilloscope triggers when the hold time is smaller than the specified hold time.
- **SETHold:** the oscilloscope triggers when the setup time or hold time is smaller than the specified time value.

Return Format

The query returns SET, HOLD, or SETH.

Example

```
:TRIGger:SHOLd:TYPE SETHold /*Sets the trigger condition to
SETHold.*/
:TRIGger:SHOLd:TYPE? /*The query returns SETH.*/
```

3.28.18.6 :TRIGger:SHOLd:STIME**Syntax**

```
:TRIGger:SHOLd:STIME <time>
```

```
:TRIGger:SHOLd:STIME?
```

Description

Sets or queries the setup time of Setup&Hold trigger. The default unit is s.

Parameter

| Name | Type | Range | Default |
|--------|------|--------------|-----------|
| <time> | Real | 1 ns to 10 s | 2 μ s |

Remarks

- Setup time indicates the time that the data remain stable and unchanged before the specified clock edge arrives.

- This command is only available when the hold type is SETup or SETHOLd.

Return Format

The query returns the setup time value in scientific notation.

Example

```
:TRIGger:SHOLd:STIME 0.002 /*Sets the setup time to 2 ms.*/
:TRIGger:SHOLd:STIME? /*The query returns 2.000000E-3.*/
```

3.28.18.7 :TRIGger:SHOLd:HTIME

Syntax

```
:TRIGger:SHOLd:HTIME <time>
:TRIGger:SHOLd:HTIME?
```

Description

Sets or queries the hold time of Setup&Hold trigger. The default unit is s.

Parameter

| Name | Type | Range | Default |
|--------|------|--------------|-----------|
| <time> | Real | 1 ns to 10 s | 1 μ s |

Remarks

- Hold time indicates the time that the data remain stable and unchanged after the specified clock edge arrives.
- This command is only available when the hold type is HOLD or SETHOLd.

Return Format

The query returns the hold time value in scientific notation.

Example

```
:TRIGger:SHOLd:HTIME 0.002 /*Sets the hold time to 2 ms.*/
:TRIGger:SHOLd:HTIME? /*The query returns 2.000000E-3.*/
```

3.28.18.8 :TRIGger:SHOLd:DLEVel

Syntax

```
:TRIGger:SHOLd:DLEVel <level>
:TRIGger:SHOLd:DLEVel?
```

Description

Sets or queries the trigger level of the data source. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Analog channel: (-4.5 x VerticalScale - Offset) to (4.5 x VerticalScale - Offset) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the trigger level of the data source in scientific notation.

Example

```
:TRIGger:SHOLd:DLEVel 0.16 /*Sets the trigger level of the data
source to 160 mV.*/
:TRIGger:SHOLd:DLEVel? /*The query returns 1.600000E-1.*/
```

3.28.18.9 :TRIGger:SHOLd:CLEVel**Syntax**

```
:TRIGger:SHOLd:CLEVel <level>
:TRIGger:SHOLd:CLEVel?
```

Description

Sets or queries the trigger level of the clock source. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Analog channel: (-4.5 x VerticalScale - Offset) to (4.5 x VerticalScale - Offset) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the trigger level of the clock source in scientific notation.

Example

```
:TRIGger:SHOLd:CLEVel 0.05 /*Sets the trigger level of the
clock source to 50 mV.*/
:TRIGger:SHOLd:CLEVel? /*The query returns 5.000000E-2.*/
```

3.28.19 :TRIGger:NEDGE**3.28.19.1 :TRIGger:NEDGE:SOURce****Syntax**

```
:TRIGger:NEDGE:SOURce <source>
```

```
:TRIGger:NEDGE:SOURce?
```

Description

Sets or queries the trigger source of the Nth Edge trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:NEDGE:SOURce CHANnel2 /*Sets the trigger source to
CHANnel2.*/
:TRIGger:NEDGE:SOURce? /*The query returns CHAN2.*/
```

3.28.19.2 :TRIGger:NEDGE:SLOPe**Syntax**

```
:TRIGger:NEDGE:SLOPe <slope>
```

```
:TRIGger:NEDGE:SLOPe?
```

Description

Sets or queries the edge type of the Nth Edge trigger.

Parameter

| Name | Type | Range | Default |
|---------|----------|---------------------|----------|
| <slope> | Discrete | {POSitive NEGative} | POSitive |

Remarks

- **POSitive:** indicates that the oscilloscope triggers on the rising edge of the input signal when the voltage level meets the preset trigger level.
- **NEGative:** indicates that the oscilloscope triggers on the falling edge of the input signal when the voltage level meets the preset trigger level.

Return Format

The query returns POS or NEG.

Example

```
:TRIGger:NEDGE:SLOPe NEGative /*Sets the edge type to
NEGative */
:TRIGger:NEDGE:SLOPe? /*The query returns NEG.*/
```

3.28.19.3 :TRIGger:NEDGE:IDLE**Syntax**

```
:TRIGger:NEDGE:IDLE <time>
```

```
:TRIGger:NEDGE:IDLE?
```

Description

Sets or queries the idle time of the Nth Edge trigger. The default unit is s.

Parameter

| Name | Type | Range | Default |
|--------|------|---------------|-----------|
| <time> | Real | 16 ns to 10 s | 1 μ s |

Remarks

N/A

Return Format

The query returns the idle time in scientific notation.

Example

```
:TRIGger:NEDGe:IDLE 0.002 /*Sets the idle time to 2 ms.*/
:TRIGger:NEDGe:IDLE? /*The query returns 2.000000E-3.*/
```

3.28.19.4 :TRIGger:NEDGe:EDGE**Syntax**

```
:TRIGger:NEDGe:EDGE <edge>
```

```
:TRIGger:NEDGe:EDGE?
```

Description

Sets or queries the number of edges of the Nth Edge trigger.

Parameter

| Name | Type | Range | Default |
|--------|---------|------------|---------|
| <edge> | Integer | 1 to 65535 | 1 |

Remarks

N/A

Return Format

The query returns an integer ranging from 1 to 65535.

Example

```
:TRIGger:NEDGe:EDGE 20 /*Sets the number of edges to 20.*/
:TRIGger:NEDGe:EDGE? /*The query returns 20.*/
```

3.28.19.5 :TRIGger:NEDGe:LEVel**Syntax**

```
:TRIGger:NEDGe:LEVel <level>
```

```
:TRIGger:NEDGe:LEVel?
```

Description

Sets or queries the trigger level of the Nth Edge trigger. The unit is the same as that of current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|--|---------|
| <level> | Real | (-4.5 x VerticalScale - Offset) to (4.5 x VerticalScale - Offset) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:NEDGE:LEVel 0.16 /*Sets the trigger level to 160 mV.*/
:TRIGger:NEDGE:LEVel? /*The query returns 1.600000E-1.*/
```

3.28.20 :TRIGger:RS232**3.28.20.1 :TRIGger:RS232:SOURce****Syntax**

```
:TRIGger:RS232:SOURce <source>
```

```
:TRIGger:RS232:SOURce?
```

Description

Sets or queries the trigger source of RS232 trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---------------------------------------|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:RS232:SOURce CHANnel2 /*Sets the trigger source to
CHANnel2.*/
:TRIGger:RS232:SOURce? /*The query returns CHAN2.*/
```

3.28.20.2 :TRIGger:RS232:LEVel**Syntax**

```
:TRIGger:RS232:LEVel <level>
```

```
:TRIGger:RS232:LEVel?
```

Description

Sets or queries the trigger level of RS232 trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Analog channel: (-4.5 x VerticalScale - Offset) to (4.5 x VerticalScale - Offset) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:RS232:LEVel 0.16 /*Sets the trigger level to 160 mV.*/
:TRIGger:RS232:LEVel? /*The query returns 1.600000E-1.*/
```

3.28.20.3 :TRIGger:RS232:POLarity**Syntax**

```
:TRIGger:RS232:POLarity <polarity>
```

```
:TRIGger:RS232:POLarity?
```

Description

Sets or queries the pulse polarity of RS232 trigger.

Parameter

| Name | Type | Range | Default |
|------------|----------|---------------------|----------|
| <polarity> | Discrete | {POSitive NEGative} | POSitive |

Remarks

N/A

Return Format

The query returns POS or NEG.

Example

```
:TRIGger:RS232:POLarity POSitive /*Sets the pulse polarity of
RS232 trigger to POSitive.*/
:TRIGger:RS232:POLarity? /*The query returns POS.*/
```

3.28.20.4 :TRIGger:RS232:WHEN**Syntax**

```
:TRIGger:RS232:WHEN <when>
```

```
:TRIGger:RS232:WHEN?
```

Description

Sets or queries the trigger condition of RS232 trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|---------------------------|---------|
| <when> | Discrete | {START ERRor CERRor DATA} | START |

Remarks

- **START:** triggers at the start of a frame.
- **ERRor:** triggers when an error frame is found.
- **CERRor:** triggers when a check error is found.
- **DATA:** triggers on the last bit of the preset data bits.

Return Format

The query returns STAR, ERR, CERR, or DATA.

Example

```
:TRIGger:RS232:WHEN ERRor /*Sets the trigger condition to
ERRor.*/
:TRIGger:RS232:WHEN? /*The query returns ERR.*/
```

3.28.20.5 :TRIGger:RS232:DATA**Syntax**

```
:TRIGger:RS232:DATA <data>
```

```
:TRIGger:RS232:DATA?
```

Description

Sets or queries the data value of RS232 trigger when the trigger condition is "Data".

Parameter

| Name | Type | Range | Default |
|--------|---------|--------------|---------|
| <data> | Integer | 0 to 2^n-1 | 0 |

Remarks

In the expression 2^n-1 , n indicates the current data width, and its available value can be 5, 6, 7, and 8. You can send the `:TRIGger:RS232:WIDTh` command to set or query the data width.

Return Format

The query returns an integer.

Example

```
:TRIGger:RS232:DATA 10 /*Sets the data value to 10.*/
:TRIGger:RS232:DATA? /*The query returns 10.*/
```

3.28.20.6 :TRIGger:RS232:BAUD**Syntax**

```
:TRIGger:RS232:BAUD <baud>
```

```
:TRIGger:RS232:BAUD?
```

Description

Sets or queries the baud rate of RS232 trigger. The default unit is bps.

Parameter

| Name | Type | Range | Default |
|--------|---------|------------------|----------|
| <baud> | Integer | 1 bps to 20 Mbps | 9600 bps |

Remarks

If the baud rate is set to a value with "M", then "A" should be added at the end of the value. For example, if you send 5M, you need to send 5MA.

Return Format

The query returns an integer ranging from 1 bps to 20 Mbps.

Example

```
:TRIGger:RS232:BAUD 4800 /*Sets the baud rate to 4800 bps.*/
:TRIGger:RS232:BAUD? /*The query returns 4800.*/
```

3.28.20.7 :TRIGger:RS232:WIDTH

Syntax

```
:TRIGger:RS232:WIDTH <width>
```

```
:TRIGger:RS232:WIDTH?
```

Description

Sets or queries the data width of RS232 trigger when the trigger condition is "Data".

Parameter

| Name | Type | Range | Default |
|---------|----------|-----------|---------|
| <width> | Discrete | {5 6 7 8} | 8 |

Remarks

N/A

Return Format

The query returns 5, 6, 7, or 8.

Example

```
:TRIGger:RS232:WIDTH 6 /*Sets the data width to 6.*/
:TRIGger:RS232:WIDTH? /*The query returns 6.*/
```

3.28.20.8 :TRIGger:RS232:STOP

Syntax

```
:TRIGger:RS232:STOP <bit>
```

```
:TRIGger:RS232:STOP?
```

Description

Sets or queries the stop bits of RS232 trigger.

Parameter

| Name | Type | Range | Default |
|-------|----------|-----------|---------|
| <bit> | Discrete | {1 1.5 2} | 1 |

Remarks

N/A

Return Format

The query returns 1, 1.5, or 2.

Example

```
:TRIGger:RS232:STOP 2 /*Sets the stop bits to 2.*/
:TRIGger:RS232:STOP? /*The query returns 2.*/
```

3.28.20.9 :TRIGger:RS232:PARity**Syntax**

```
:TRIGger:RS232:PARity <parity>
```

```
:TRIGger:RS232:PARity?
```

Description

Sets or queries the check mode of RS232 trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|-----------------|---------|
| <parity> | Discrete | {EVEN ODD NONE} | NONE |

Remarks

N/A

Return Format

The query returns EVEN, ODD, or NONE.

Example

```
:TRIGger:RS232:PARity EVEN /*Sets the check mode to EVEN.*/
:TRIGger:RS232:PARity? /*The query returns EVEN.*/
```

3.28.21 :TRIGger:IIC**3.28.21.1 :TRIGger:IIC:SCL****Syntax**

```
:TRIGger:IIC:SCL <source>
```

```
:TRIGger:IIC:SCL?
```

Description

Sets or queries the source channel of the clock line of I2C trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:IIC:SCL CHANnel2 /*Sets the clock source to CHANnel2.*/
:TRIGger:IIC:SCL? /*The query returns CHAN2.*/
```

3.28.21.2 :TRIGger:IIC:CLEVel**Syntax**

```
:TRIGger:IIC:CLEVel <level>
```

```
:TRIGger:IIC:CLEVel?
```

Description

Sets or queries the trigger level of the clock line in I2C trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Analog channel: (-4.5 x VerticalScale - Offset) to (4.5 x VerticalScale - Offset) | 0 V |

Remarks

For VerticalScale, refer to the *:CHANnel<n>:SCALE* command. For OFFSet, refer to the *:CHANnel<n>:OFFSet* command.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:IIC:CLEVel 0.16 /*Sets the trigger level to 160 mV.*/
:TRIGger:IIC:CLEVel? /*The query returns 1.600000E-1.*/
```


3.28.21.3 :TRIGger:IIC:SDA

Syntax

```
:TRIGger:IIC:SDA <source>
```

```
:TRIGger:IIC:SDA?
```

Description

Sets or queries the source channel of the data line of I2C trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel2 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:IIC:SDA CHANnel2 /*Sets the data source to CHANnel2.*/
:TRIGger:IIC:SDA? /*The query returns CHAN2.*/
```

3.28.21.4 :TRIGger:IIC:DLEVel

Syntax

```
:TRIGger:IIC:DLEVel <level>
```

```
:TRIGger:IIC:DLEVel?
```

Description

Sets or queries the trigger level of the data line in I2C trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Analog channel: (-4.5 x VerticalScale - Offset) to (4.5 x VerticalScale - Offset) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:IIC:DLEVel 0.16 /*Sets the trigger level to 160 mV.*/
:TRIGger:IIC:DLEVel? /*The query returns 1.600000E-1.*/
```

3.28.21.5 :TRIGger:IIC:WHEN**Syntax**

```
:TRIGger:IIC:WHEN <when>
```

```
:TRIGger:IIC:WHEN?
```

Description

Sets or queries the trigger condition of I2C trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|--|---------|
| <when> | Discrete | {START REStArt STOP NACKnowledge ADDRess DATA ADATa} | START |

Remarks

- **START:** indicates that the oscilloscope triggers when SCL is high level and SDA transitions from high level to low level.
- **REStArt:** indicates that the oscilloscope triggers when another start condition occurs before a stop condition.
- **STOP:** indicates that the oscilloscope triggers when SCL is high level and SDA transitions from low level to high level.
- **NACKnowledge:** indicates missing acknowledgment. The oscilloscope triggers when SDA is high level during the acknowledgment of the SCL bit.
- **ADDRess:** indicates that the oscilloscope searches for the specified address and triggers on the read/write bit.

- **DATA:** indicates that the oscilloscope searches for the specified data on the data line (SDA) and triggers on the clock line (SCL) of the jumping edge of the last bit of the data.
- **ADATa:** indicates that the oscilloscope searches for the specified address and data, and then triggers when both the address and data meet the conditions.

Return Format

The query returns STAR, REST, STOP, NACK, ADDR, DATA, or ADAT.

Example

```
:TRIGger:IIC:WHEN REStart /*Sets the trigger condition to
REStart.*/
:TRIGger:IIC:WHEN? /*The query returns REST.*/
```

3.28.21.6 :TRIGger:IIC:AWIDth

Syntax

```
:TRIGger:IIC:AWIDth <bits>
```

```
:TRIGger:IIC:AWIDth?
```

Description

Sets or queries the address width of I2C trigger when the trigger condition is "ADDRess" or "ADATa".

Parameter

| Name | Type | Range | Default |
|--------|----------|----------|---------|
| <bits> | Discrete | {7 8 10} | 7 |

Remarks

N/A

Return Format

The query returns 7, 8, or 10.

Example

```
:TRIGger:IIC:AWIDth 10 /*Sets the address width to 10.*/
:TRIGger:IIC:AWIDth? /*The query returns 10.*/
```

3.28.21.7 :TRIGger:IIC:ADDRess

Syntax

```
:TRIGger:IIC:ADDRess <address>
```

```
:TRIGger:IIC:ADDRess?
```

Description

Sets or queries the address of I2C trigger when the trigger condition is "ADDRess" or "ADATa".

Parameter

| Name | Type | Range | Default |
|-----------|---------|------------------------|---------|
| <address> | Integer | 0 to 2 ⁿ -1 | 0 |

Remarks

In the expression 2ⁿ-1, n indicates the current address width. Its range is from 0 to 127, 0 to 255, or 0 to 1,023.

Return Format

The query returns the address in integer.

Example

```
:TRIGger:IIC:ADDRess 100 /*Sets the address to 100.*/
:TRIGger:IIC:ADDRess? /*The query returns 100.*/
```

3.28.21.8 :TRIGger:IIC:DIRection

Syntax

```
:TRIGger:IIC:DIRection <direction>
```

```
:TRIGger:IIC:DIRection?
```

Description

Sets or queries the data direction of I2C trigger when the trigger condition is "ADDRess" or "ADATa".

Parameter

| Name | Type | Range | Default |
|-------|----------|---------------------|---------|
| <dir> | Discrete | {READ WRITe RWRite} | WRITe |

Remarks

This command is unavailable when the address width is set to 8.

Return Format

The query returns READ, WRIT, or RWR.

Example

```
:TRIGger:IIC:DIRection RWRite /*Sets the data direction to
RWRite.*/
:TRIGger:IIC:DIRection? /*The query returns RWR.*/
```

3.28.21.9 :TRIGger:IIC:DBYTes**Syntax**

```
:TRIGger:IIC:DBYTes <bytes>
```

```
:TRIGger:IIC:DBYTes?
```

Description

Sets or queries of the data bytes of I2C trigger when the trigger condition is "DATA" or "ADATa".

Parameter

| Name | Type | Range | Default |
|---------|------|--------|---------|
| <bytes> | Real | 1 to 5 | 1 |

Remarks

N/A

Return Format

The query returns the data bytes in scientific notation.

Example

```
:TRIGger:IIC:DBYTes 3 /*Sets the data bytes to 3 when the
trigger condition is "DATA" or "ADATa".*/
:TRIGger:IIC:DBYTes? /*The query returns 3.*/
```

3.28.21.10 :TRIGger:IIC:DATA**Syntax**

```
:TRIGger:IIC:DATA <data>
```

```
:TRIGger:IIC:DATA?
```

Description

Sets or queries the data value of I2C trigger when the trigger condition is "DATA" or "ADATa".

Parameter

| Name | Type | Range | Default |
|--------|---------|-------------------------|---------|
| <data> | Integer | 0 to 2 ⁴⁰ -1 | 0 |

Remarks

The settable range of <data> is affected by the data bytes. You can send the *:TRIGger:IIC:DBYtes* command to set the data bytes. The maximum byte length can be set to 5, i.g. 40-bit binary data. Therefore, the range of <data> is from 0 to 2⁴⁰-1.

Return Format

The query returns an integer.

Example

```
:TRIGger:IIC:DATA 64 /*Sets the data value to 64.*/
:TRIGger:IIC:DATA? /*The query returns 64.*/
```

3.28.21.11 :TRIGger:IIC:CURRbit**Syntax**

```
:TRIGger:IIC:CURRbit <currbit>
:TRIGger:IIC:CURRbit?
```

Description

Sets or queries the current bit of the I2C trigger data.

Parameter

| Name | Type | Range | Default |
|-----------|---------|---------|---------|
| <currbit> | Integer | 0 to 39 | 0 |

Remarks

After configuring the settings for this command, you can send the *:TRIGger:IIC:CODE* command to set or modify the set bit data.

Return Format

The query returns an integer ranging from 0 to 39.

Example

```
:TRIGger:IIC:CURRbit 8 /*Sets the current bit of I2C trigger
data to 8. That is, the oscilloscope triggers on the 9th bit of
I2C trigger data.*/
:TRIGger:IIC:CURRbit? /*The query returns 8.*/
```

3.28.21.12 :TRIGger:IIC:CODE**Syntax**

```
:TRIGger:IIC:CODE <code>
```

```
:TRIGger:IIC:CODE?
```

Description

Sets or queries the data value of a certain bit of I2C trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|-----------|---------|
| <code> | Discrete | {0 1 255} | 255 |

Remarks

When <code> is set to 255, it indicates the data value can be any value.

After sending the *:TRIGger:IIC:CURRbit* command to set the specified bit, you can send this command to query or modify the value of the specified data bit.

Return Format

The query returns 0, 1, or 255.

Example

```
:TRIGger:IIC:CODE 0 /*Sets the data value to 0.*/
:TRIGger:IIC:CODE? /*The query returns 0.*/
```

3.28.22 :TRIGger:SPI**3.28.22.1 :TRIGger:SPI:CLK****Syntax**

```
:TRIGger:SPI:CLK <source>
```

```
:TRIGger:SPI:CLK?
```

Description

Sets or queries the channel source of the clock line of SPI trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:SPI:CLK CHANnel3 /*Sets the channel source of the data
line of SPI trigger to CHANnel3.*/
:TRIGger:SPI:CLK? /*The query returns CHAN3.*/
```

3.28.22.2 :TRIGger:SPI:SCL**Syntax**

```
:TRIGger:SPI:SCL <source>
```

```
:TRIGger:SPI:SCL?
```

Description

Sets or queries the channel source of the clock line of SPI trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

This command exists for backwards compatibility. Use the command *:TRIGger:SPI:CLK*.

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:SPI:SCL CHANnel1 /*Sets the channel source of the clock
line to CHANnel1.*/
:TRIGger:SPI:SCL? /*The query returns CHAN1.*/
```


3.28.22.3 :TRIGger:SPI:CLeVel

Syntax

```
:TRIGger:SPI:CLeVel <level>
```

```
:TRIGger:SPI:CLeVel?
```

Description

Sets or queries the trigger level of the clock channel of SPI trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Analog channel: (-4.5 x VerticalScale - OFFSet) to (4.5 x VerticalScale - OFFSet) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALe` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:SPI:CLeVel 0.16 /*Sets the trigger level to 160 mV.*/
:TRIGger:SPI:CLeVel? /*The query returns 1.600000E-1.*/
```

3.28.22.4 :TRIGger:SPI:SLOPe

Syntax

```
:TRIGger:SPI:SLOPe <slope>
```

```
:TRIGger:SPI:SLOPe?
```

Description

Sets or queries the type of the clock edge of SPI trigger.

Parameter

| Name | Type | Range | Default |
|---------|----------|---------------------|----------|
| <slope> | Discrete | {POSitive NEGative} | POSitive |

Remarks

- **POSitive:** samples the data on the rising edge of the clock.
- **NEGative:** samples the data on the falling edge of the clock.

Return Format

The query returns POS or NEG.

Example

```
:TRIGger:SPI:SLOPe POSitive /*Sets the clock edge to POSitive.*/
:TRIGger:SPI:SLOPe? /*The query returns POS.*/
```

3.28.22.5 :TRIGger:SPI:MISO**Syntax**

```
:TRIGger:SPI:MISO <source>
```

```
:TRIGger:SPI:MISO?
```

Description

Sets or queries the channel source of the data line of SPI trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel2 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, and CHAN4.

Example

```
:TRIGger:SPI:MISO CHANnel3 /*Sets the channel source of the data
line of SPI trigger to CHANnel3.*/
:TRIGger:SPI:MISO? /*The query returns CHAN3.*/
```

3.28.22.6 :TRIGger:SPI:SDA**Syntax**

```
:TRIGger:SPI:SDA <source>
```

```
:TRIGger:SPI:SDA?
```

Description

Sets or queries the channel source of the data line of SPI trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel2 |

Remarks

This command exists for backwards compatibility. Use the command *:TRIGger:SPI:MISO*.

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:SPI:SDA CHANnel2 /*Sets the channel source of the data
line to CHANnel2.*/
:TRIGger:SPI:SDA? /*The query returns CHAN2.*/
```

3.28.22.7 :TRIGger:SPI:DLEVel**Syntax**

```
:TRIGger:SPI:DLEVel <level>
```

```
:TRIGger:SPI:DLEVel?
```

Description

Sets or queries the trigger level of the data channel of SPI trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Analog channel: (-4.5 x VerticalScale - OFFSet) to (4.5 x VerticalScale - OFFSet) | 0 V |

Remarks

For VerticalScale, refer to the *:CHANnel<n>:SCALE* command. For OFFSet, refer to the *:CHANnel<n>:OFFSet* command.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:SPI:DLEVel 0.16 /*Sets the trigger level to 160 mV.*/
:TRIGger:SPI:DLEVel? /*The query returns 1.600000E-1.*/
```

3.28.22.8 :TRIGger:SPI:WHEN**Syntax**

```
:TRIGger:SPI:WHEN <when>
```

```
:TRIGger:SPI:WHEN?
```

Description

Sets or queries the trigger condition of SPI trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|--------------|---------|
| <when> | Discrete | {CS TIMEout} | CS |

Remarks

When the trigger condition is "TIMEout", you can run the *:TRIGger:SPI:TIMEout* command to set the timeout value.

Return Format

The query returns CS or TIM.

Example

```
:TRIGger:SPI:WHEN TIMEout /*Sets the trigger condition to
TIMEout.*/
:TRIGger:SPI:WHEN? /*The query returns TIM.*/
```

3.28.22.9 :TRIGger:SPI:CS**Syntax**

```
:TRIGger:SPI:CS <source>
```

```
:TRIGger:SPI:CS?
```

Description

Sets or queries the source channel of the CS line when the trigger condition of SPI is set to CS.

Parameter

| Name | Type | Range | Default |
|----------|----------|---------------------------------------|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel3 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:SPI:CS CHANnel2 /*Sets the source channel of the CS
line to CHANnel2 when the trigger condition of SPI is set to CS.*/
:TRIGger:SPI:CS? /*The query returns CHAN2.*/
```

3.28.22.10 :TRIGger:SPI:SLEVel**Syntax**

```
:TRIGger:SPI:SLEVel <level>
```

```
:TRIGger:SPI:SLEVel?
```

Description

Sets or queries the trigger level of the CS channel of SPI trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Analog channel: (-4.5 x VerticalScale - OFFSet) to (4.5 x VerticalScale - OFFSet) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:SPI:SLEVel 0.16 /*Sets the trigger level to 160 mV.*/
:TRIGger:SPI:SLEVel? /*The query returns 1.600000E-1.*/
```

3.28.22.11 :TRIGger:SPI:MODE

Syntax

```
:TRIGger:SPI:MODE <mode>
```

```
:TRIGger:SPI:MODE?
```

Description

Sets or queries the CS mode of SPI trigger when the trigger condition is "CS".

Parameter

| Name | Type | Range | Default |
|--------|----------|------------|---------|
| <mode> | Discrete | {HIGH LOW} | LOW |

Remarks

N/A

Return Format

The query returns HIGH or LOW.

Example

```
:TRIGger:SPI:MODE LOW /*Sets the CS mode to LOW.*/
:TRIGger:SPI:MODE? /*The query returns LOW.*/
```

3.28.22.12 :TRIGger:SPI:TIMEout

Syntax

```
:TRIGger:SPI:TIMEout <time>
```

```
:TRIGger:SPI:TIMEout?
```

Description

Sets or queries the timeout value when the trigger condition of SPI trigger is "Timeout". The default unit is s.

Parameter

| Name | Type | Range | Default |
|--------|------|--------------|-----------|
| <time> | Real | 16 ns to 1 s | 1 μ s |

Remarks

N/A

Return Format

The query returns the timeout value in scientific notation.

Example

```
:TRIGger:SPI:TIMEout 0.001 /*Sets the timeout value to 1 ms.*/
:TRIGger:SPI:TIMEout? /*The query returns 1.000000E-3.*/
```

3.28.22.13 :TRIGger:SPI:WIDTH**Syntax**

```
:TRIGger:SPI:WIDTH <width>
```

```
:TRIGger:SPI:WIDTH?
```

Description

Sets or queries the data width of data channel in SPI trigger.

Parameter

| Name | Type | Range | Default |
|---------|---------|---------|---------|
| <width> | Integer | 4 to 32 | 8 |

Remarks

N/A

Return Format

The query returns an integer ranging from 4 to 32.

Example

```
:TRIGger:SPI:WIDTH 10 /*Sets the data width to 10.*/
:TRIGger:SPI:WIDTH? /*The query returns 10.*/
```

3.28.22.14 :TRIGger:SPI:DATA**Syntax**

```
:TRIGger:SPI:DATA <data>
```

```
:TRIGger:SPI:DATA?
```

Description

Sets or queries the data value of SPI trigger.

Parameter

| Name | Type | Range | Default |
|--------|---------|-------------------------|---------|
| <data> | Integer | 0 to 2 ³² -1 | 0 |

Remarks

The range of the parameter <data> is related to the current data width. You can send the `:TRIGger:SPI:WIDTh` command to set or query the data width. The available maximum data width is 32. Therefore, the range of <data> is from 0 to 2³²-1.

Return Format

The query returns an integer.

Example

```
:TRIGger:SPI:DATA 5 /*Sets the data value to 5.*/
:TRIGger:SPI:DATA? /*The query returns 5.*/
```

3.28.22.15 :TRIGger:SPI:CURRbit**Syntax**

```
:TRIGger:SPI:CURRbit <currbit>
```

```
:TRIGger:SPI:CURRbit?
```

Description

Sets or queries the current bit of the SPI trigger data.

Parameter

| Name | Type | Range | Default |
|-----------|---------|---------|---------|
| <currbit> | Integer | 0 to 39 | 0 |

Remarks

After configuring the settings for this command, you can send the `:TRIGger:SPI:CODE` command to set or modify the set bit data.

Return Format

The query returns an integer ranging from 0 to 39.

Example

```
:TRIGger:SPI:CURRbit 8 /*Sets the current bit of SPI trigger
data to 8. That is, the oscilloscope triggers on the 9th bit of
SPI trigger data.*/
:TRIGger:SPI:CURRbit? /*The query returns 8.*/
```


3.28.22.16 :TRIGger:SPI:CODE**Syntax**

```
:TRIGger:SPI:CODE <code>
```

```
:TRIGger:SPI:CODE?
```

Description

Sets or queries the data value of a certain bit of SPI trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|-----------|---------|
| <code> | Discrete | {0 1 255} | 255 |

Remarks

When <code> is set to 255, it indicates the data value can be any value.

After sending the *:TRIGger:SPI:CURRbit* command to set the specified bit, you can send this command to query or modify the value of the specified data bit.

Return Format

The query returns 0, 1, or 255.

Example

```
:TRIGger:SPI:CODE 0 /*Sets the data value to 0.*/
:TRIGger:SPI:CODE? /*The query returns 0.*/
```

3.28.23 :TRIGger:CAN

The :TRIGger:CAN commands are used to set relevant parameters for the CAN trigger.

3.28.23.1 :TRIGger:CAN:BAUD**Syntax**

```
:TRIGger:CAN:BAUD <baud>
```

```
:TRIGger:CAN:BAUD?
```

Description

Sets or queries the signal rate of CAN trigger. The unit is bps.

Parameter

| Name | Type | Range | Default |
|--------|---------|-------------------|---------|
| <baud> | Integer | 10 kbps to 5 Mbps | 1 Mbps |

Remarks

If the baud rate is set to a value with "M", then "A" should be added at the end of the value. For example, if you send 5M, you need to send 5MA.

Return Format

The query returns an integer ranging from 10 kbps to 5 Mbps.

Example

```
:TRIGger:CAN:BAUD 125000 /*Sets the signal rate to 125000 bps.*/
:TRIGger:CAN:BAUD? /*The query returns 125000.*/
```

3.28.23.2 :TRIGger:CAN:SOURce**Syntax**

```
:TRIGger:CAN:SOURce <source>
```

```
:TRIGger:CAN:SOURce?
```

Description

Sets or queries the trigger source of CAN trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:CAN:SOURce CHANnel2 /*Sets the trigger source to
CHANnel2.*/
:TRIGger:CAN:SOURce? /*The query returns CHAN2.*/
```

3.28.23.3 :TRIGger:CAN:STYPe

Syntax

```
:TRIGger:CAN:STYPe <stype>
```

```
:TRIGger:CAN:STYPe?
```

Description

Sets or queries the signal type of CAN trigger.

Parameter

| Name | Type | Range | Default |
|---------|----------|-------------------------|---------|
| <stype> | Discrete | {H L RXTX DIFFerential} | H |

Remarks

- **H:** indicates the actual CAN_H differential bus signal.
- **L:** indicates the actual CAN_L differential bus signal.
- **RXTX:** indicates the Receive or Transmit signal from the CAN bus transceiver.
- **DIFFerential:** indicates the CAN differential bus signal connected to an analog channel by using a differential probe. Connect the differential probe's positive lead to the CAN_H bus signal and connect the negative lead to the CAN_L bus signal.

Return Format

The query returns H, L, RXTX, or DIFF.

Example

```
:TRIGger:CAN:STYPe L /*Sets the signal type to CAN_L
differential bus signal.*/
:TRIGger:CAN:STYPe? /*The query returns L.*/
```

3.28.23.4 :TRIGger:CAN:WHEN

Syntax

```
:TRIGger:CAN:WHEN <cond>
```

```
:TRIGger:CAN:WHEN?
```

Description

Sets or queries the trigger condition of CAN trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|---|---------|
| <cond> | Discrete | {SOF EOF IDRemote OVERload IDFrame DATAframe IDData ERFrame ERANswer ERCheck ERFormat ERRandom ERBit} | SOF |

Remarks

- **SOF:** indicates start of frame. It indicates that the oscilloscope triggers at the start of a data frame.
- **EOF:** indicates end of frame. It indicates that the oscilloscope triggers at the end of a data frame.

Frame Type

- **IDRemote:** indicates remote ID. It indicates that the oscilloscope triggers on the remote frame with the specified ID.
- **OVERload:** indicates overload frame. It indicates that the oscilloscope triggers on the CAN overload frames.
- **IDFrame:** indicates frame ID. It indicates that the oscilloscope triggers on the data frames with the specified ID.
- **DATAframe:** indicates frame data. It indicates that the oscilloscope triggers on the data frames with specified data.
- **IDData:** indicates Data & ID. It indicates that the oscilloscope triggers on the data frames with the specified ID and data.

Frame Error

- **ERFrame:** indicates frame error. It indicates that the oscilloscope triggers on the error frame.
- **ERANswer:** indicates answer error. It indicates that the oscilloscope triggers on the answer error frame.
- **ERCheck:** indicates check error. It indicates that the oscilloscope triggers on the check error frame.

- **ERFormat:** indicates format error. It indicates that the oscilloscope triggers on the format error frame.
- **ERRandom:** indicates random error. It indicates that the oscilloscope triggers on the random error frame, such as the format error frame, answer error frame, etc.
- **ERBit:** indicates bit fill. It indicates that the oscilloscope triggers on the error frame with the bit fill.

Return Format

The query returns SOF, EOF, IDR, OVER, IDFR, DAT, IDD, ERFR, ERAN, ERCH, ERF, ERR, or ERB.

Example

```
:TRIGger:CAN:WHEN EOF      /*Sets the trigger condition to EOF.*/
:TRIGger:CAN:WHEN?        /*The query returns EOF.*/
```

3.28.23.5 :TRIGger:CAN:SPOint

Syntax

```
:TRIGger:CAN:SPOint <spoint>
```

```
:TRIGger:CAN:SPOint?
```

Description

Sets or queries the sample point position of CAN trigger (expressed in %).

Parameter

| Name | Type | Range | Default |
|----------|---------|----------|---------|
| <spoint> | Integer | 10 to 90 | 50 |

Remarks

The sample point is within the range of the bit time. The oscilloscope samples the bit level at the sample point. The sample point position is expressed as the ratio of "time from the bit start to the sample point" to "bit time", in %.

Return Format

The query returns an integer ranging from 10 to 90.

Example

```
:TRIGger:CAN:SPoint 60 /*Sets the sample point position of
CAN trigger to 60%.*
:TRIGger:CAN:SPoint? /*The query returns 60.*
```

3.28.23.6 :TRIGger:CAN:EXTended**Syntax**

```
:TRIGger:CAN:EXTended <bool>
```

```
:TRIGger:CAN:EXTended?
```

Description

Enables or disables the extended ID when the trigger condition of CAN trigger is "Remote ID" or "Frame ID"; queries whether the extended ID is enabled when the trigger condition of CAN trigger is "Remote ID" or "Frame ID".

Parameter

| Name | Type | Range | Default |
|--------|------|--------------------|---------|
| <bool> | Bool | {{1 ON}} {{0 OFF}} | 0 OFF |

Remarks

- **0|OFF:** disables the extended ID.
- **1|ON:** enables the extended ID.

To set or query the trigger condition of CAN trigger, send the *:TRIGger:CAN:WHEN* command.

Return Format

The query returns 0 or 1.

Example

```
:TRIGger:CAN:EXTended ON /*Enables the extended ID.*
:TRIGger:CAN:EXTended /*The query returns 1.*
```

3.28.23.7 :TRIGger:CAN:DEFine**Syntax**

```
:TRIGger:CAN:DEFine <type>
```

```
:TRIGger:CAN:DEFine?
```

Description

Sets Define to ID or Data when the trigger condition of CAN trigger is set to Data or ID; queries Define to ID or Data when the trigger condition of CAN trigger is set to Data or ID.

Parameter

| Name | Type | Range | Default |
|--------|----------|-----------|---------|
| <type> | Discrete | {DATA ID} | DATA |

Remarks

- **DATA:** sets Define to Data.
- **ID:** sets Define to ID.

To set or query the trigger condition of CAN trigger, send the *:TRIGger:CAN:WHEN* command.

Return Format

The query returns DATA or ID.

Example

```
:TRIGger:CAN:DEFine ID /*Sets Define to ID.*/
:TRIGger:CAN:DEFine? /*The query returns ID.*/
```

3.28.23.8 :TRIGger:CAN:DWIDth**Syntax**

```
:TRIGger:CAN:DWIDth <data>
```

```
:TRIGger:CAN:DWIDth?
```

Description

Sets or queries the data width of CAN trigger when the trigger condition is "DATAframe" or "IDData".

Parameter

| Name | Type | Range | Default |
|--------|---------|--------|---------|
| <data> | Integer | 1 to 8 | 1 |

Remarks

N/A

Return Format

The query returns an integer ranging from 1 to 8.

Example

```
:TRIGger:CAN:DWIDth 5 /*Sets the data width of CAN trigger to 5
when the trigger condition is "DATAframe" or "IDData".*/
:TRIGger:CAN:DWIDth? /*The query returns 5.*/
```

3.28.23.9 :TRIGger:CAN:DATA

Syntax

```
:TRIGger:CAN:DATA <data>
```

```
:TRIGger:CAN:DATA?
```

Description

Sets or queries the data value of CAN trigger.

Parameter

| Name | Type | Range | Default |
|--------|---------|-------------------------|---------|
| <data> | Integer | 0 to 2 ⁴⁰ -1 | 0 |

Remarks

The settable range of <data> is affected by the data bytes. You can send the *:TRIGger:IIC:DBYtes* command to set the data bytes. The maximum byte length can be set to 5, i.g. 40-bit binary data. Therefore, the range of <data> is from 0 to 2⁴⁰-1.

This command is valid when the trigger condition is set to "Frame Data" or "Data & ID" (send *:TRIGger:CAN:WHEN* command to set or query).

- When the trigger condition is "Frame Data", the setting command is used to set the data value.
- When the trigger condition is "Data & ID", the value to be set by sending the setting command is based on the *:TRIGger:CAN:DEFine* command.
 - When Define is set to "ID", the setting command is used to set the ID value.
 - When Define is set to "Data", the setting command is used to set the data value.

Return Format

The query returns an integer.

Example

```
:TRIGger:CAN:DATA 100 /*Sets the data value of CAN trigger to
100.*/
:TRIGger:CAN:DATA? /*The query returns 100.*/
```

3.28.23.10 :TRIGger:CAN:CURRbit**Syntax**

```
:TRIGger:CAN:CURRbit <currbit>
```

```
:TRIGger:CAN:CURRbit?
```

Description

Sets or queries the current bit of the CAN trigger data.

Parameter

| Name | Type | Range | Default |
|-----------|---------|---------|---------|
| <currbit> | Integer | 0 to 39 | 0 |

Remarks

After configuring the settings for this command, you can send the *:TRIGger:CAN:CODE* command to set or modify the set bit data.

Return Format

The query returns an integer ranging from 0 to 39.

Example

```
:TRIGger:CAN:CURRbit 8 /*Sets the current bit of CAN trigger
data to 8. That is, the oscilloscope triggers on the 9th bit of
CAN trigger data.*/
:TRIGger:SPI:CURRbit? /*The query returns 8.*/
```

3.28.23.11 :TRIGger:CAN:CODE**Syntax**

```
:TRIGger:CAN:CODE <code>
```

```
:TRIGger:CAN:CODE?
```

Description

Sets or queries the data value of a certain bit of CAN trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|-----------|---------|
| <code> | Discrete | {0 1 255} | 255 |

Remarks

When <code> is set to 255, it indicates the data value can be any value.

After sending the `:TRIGger:CAN:CURRbit` command to set the specified bit, you can send this command to query or modify the value of the specified data bit.

Return Format

The query returns 0, 1, or 255.

Example

```
:TRIGger:CAN:CODE 0 /*Sets the data value to 0.*/
:TRIGger:CAN:CODE? /*The query returns 0.*/
```

3.28.23.12 :TRIGger:CAN:LEVel**Syntax**

```
:TRIGger:CAN:LEVel <level>
```

```
:TRIGger:CAN:LEVel?
```

Description

Sets or queries the trigger level of CAN trigger. Its unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Analog channel: (-4.5 x VerticalScale - Offset) to (4.5 x VerticalScale - Offset) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:CAN:LEVel 0.16 /*Sets the trigger level to 160 mV.*/
:TRIGger:CAN:LEVel? /*The query returns 1.600000E-1.*/
```

3.28.23.13 :TRIGger:CAN:FDBaud**Syntax**

```
:TRIGger:CAN:FDBaud <fdbaud>
```

```
:TRIGger:CAN:FDBaud?
```

Description

Sets or queries the CAN-FD baud rate of CAN trigger. The unit is bps.

Parameter

| Name | Type | Range | Default |
|----------|---------|-------------------|---------|
| <fdbaud> | Integer | 1 Mbps to 10 Mbps | 1 Mbps |

Remarks

If the baud rate is set to a value with "M", then "A" should be added at the end of the value. For example, if you send 5M, you need to send 5MA.

The DHO1000 series oscilloscope does not support this command. This command is optional for the DHO4000 series oscilloscope.

Return Format

The query returns an integer ranging from 1M to 10M.

Example

```
:TRIGger:CAN:FDBaud 120000 /*Sets the CAN-FD baud rate of
CAN trigger to 120000 bps.*/
:TRIGger:CAN:FDBaud? /*The query returns 120000.*/
```

3.28.23.14 :TRIGger:CAN:FDSPoint**Syntax**

```
:TRIGger:CAN:FDSPoint <fdspoint>
```

```
:TRIGger:CAN:FDSPoint?
```

Description

Sets or queries the FD sample position of CAN trigger (expressed in percentage).

Parameter

| Name | Type | Range | Default |
|------------|---------|----------|---------|
| <fdspoint> | Integer | 10 to 90 | 50 |

Remarks

The sample point is within the range of the bit time. The oscilloscope samples the bit level at the sample point. The sample point position is expressed as the ratio of "time from the bit start to the sample point" to "bit time", in %.

The DHO1000 series oscilloscope does not support this command. This command is optional for the DHO4000 series oscilloscope.

Return Format

The query returns an integer ranging from 10 to 90.

Example

```
:TRIGger:CAN:FDSPoint 70 /*Sets the FD sample position of
CAN trigger to 70%.*/
:TRIGger:CAN:FDSPoint? /*The query returns 70.*/
```

3.28.24 :TRIGger:LIN

The :TRIGger:LIN commands are used to set relevant parameters for the LIN trigger.

**NOTE**

The :TRIGger:LIN commands are standard configurations for the DHO1000 series oscilloscope. The :TRIGger:LIN commands are optional configurations for the DHO4000 series oscilloscope.

3.28.24.1 :TRIGger:LIN:SOURce**Syntax**

```
:TRIGger:LIN:SOURce <source>
```

```
:TRIGger:LIN:SOURce?
```

Description

Sets or queries the trigger source of LIN trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:LIN:SOURce CHANnel2 /*Sets the trigger source to
CHANnel2.*/
:TRIGger:LIN:SOURce? /*The query returns CHAN2.*/
```

3.28.24.2 :TRIGger:LIN:LEVel**Syntax**

```
:TRIGger:LIN:LEVel <level>
```

```
:TRIGger:LIN:LEVel?
```

Description

Sets or queries the trigger level of LIN trigger. Its unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Analog channel: (-4.5 x VerticalScale - OFFSet) to (4.5 x VerticalScale - OFFSet) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:LIN:LEVel 0.16 /*Sets the trigger level to 160 mV.*/
:TRIGger:LIN:LEVel? /*The query returns 1.600000E-1.*/
```

3.28.24.3 :TRIGger:LIN:STANdard**Syntax**

```
:TRIGger:LIN:STANdard <std>
```

```
:TRIGger:LIN:STANdard?
```

Description

Sets or queries the protocol version of LIN trigger.

Parameter

| Name | Type | Range | Default |
|-------|----------|--------------|---------|
| <std> | Discrete | {1X 2X BOTH} | BOTH |

Remarks

N/A

Return Format

The query returns 1X, 2X, or BOTH.

Example

```
:TRIGger:LIN:STANdard 2X /*Sets the protocol version of
LIN trigger to 2X.*/
:TRIGger:LIN:STANdard? /*The query returns 2X.*/
```

3.28.24.4 :TRIGger:LIN:BAUD**Syntax**

```
:TRIGger:LIN:BAUD <baud>
```

```
:TRIGger:LIN:BAUD?
```

Description

Sets or queries the baud rate of LIN trigger. The default unit is bps.

Parameter

| Name | Type | Range | Default |
|--------|---------|-------------------|----------|
| <baud> | Integer | 1 kbps to 20 Mbps | 9600 bps |

Remarks

If the baud rate is set to a value with "M", then "A" should be added at the end of the value. For example, if you send 5M, you need to send 5MA.

Return Format

The query returns an integer ranging from 1 kbps to 20 Mbps.

Example

```
:TRIGger:LIN:BAUD 19200 /*Sets the baud rate of LIN trigger
to 19.2 kbps.*/
:TRIGger:LIN:BAUD? /*The query returns 19200.*/
```

3.28.24.5 :TRIGger:LIN:SAMPlepoint

Syntax

```
:TRIGger:LIN:SAMPlepoint <value>
```

```
:TRIGger:LIN:SAMPlepoint?
```

Description

Sets or queries the sample position of LIN trigger.

Parameter

| Name | Type | Range | Default |
|---------|---------|----------|---------|
| <value> | Integer | 10 to 90 | 50 |

Remarks

The sample position is expressed as the ratio of "time from the bit start to the sample point" to "bit time", in %.

Return Format

The query returns an integer ranging from 10 to 90.

Example

```
:TRIGger:LIN:SAMPlepoint 40 /*Sets the sample point
position of LIN trigger to 40%.*/
:TRIGger:LIN:SAMPlepoint? /*The query returns 40.*/
```

3.28.24.6 :TRIGger:LIN:WHEN

Syntax

```
:TRIGger:LIN:WHEN <when>
```

```
:TRIGger:LIN:WHEN?
```

Description

Sets or queries the trigger condition of LIN trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|---|-----------|
| <when> | Discrete | {SYNCbreak ID DATA IDData SLEep WAKEup ERRor} | SYNCbreak |

Remarks

- **SYNCbreak:** triggers on the last bit of the sync field.
- **ID:** triggers when the frames with the specified ID are found.
- **DATA:** triggers when the data that meet the preset conditions are found.
- **IDData:** triggers when the frames with the specified ID and data that meet the preset conditions are both found.
- **SLEep:** triggers when the sleep frame is found.
- **WAKEup:** triggers when the wakeup frame is found.
- **ERRor:** triggers on the specified type of error frame.

Return Format

The query returns SYNC, ID, DATA, IDD, SLE, WAK, or ERR.

Example

```
:TRIGger:LIN:WHEN SYNCbreak /*Sets the trigger condition to
SYNCbreak.*/
:TRIGger:LIN:WHEN? /*The query returns SYNC.*/
```

3.28.24.7 :TRIGger:LIN:ERRor**Syntax**

```
:TRIGger:LIN:ERRor <value>
```

```
:TRIGger:LIN:ERRor?
```

Description

Sets or queries the error type of LIN trigger when the trigger condition is "Data".

Parameter

| Name | Type | Range | Default |
|---------|----------|-----------------|---------|
| <value> | Discrete | {SYNC ID CHECK} | SYNC |

Remarks

- **SYNC:** indicates Sync error.
- **ID:** indicates Even Odd error.
- **CHECK:** indicates Check Sum error.

Return Format

The query returns SYNC, ID, or CHEC.

Example

```
:TRIGger:LIN:ERRor ID          /*Sets the LIN trigger error type to
ID.*/*
:TRIGger:LIN:ERRor?           /*The query returns ID.*/*
```

3.28.24.8 :TRIGger:LIN:ID**Syntax**

```
:TRIGger:LIN:ID <id>
```

```
:TRIGger:LIN:ID?
```

Description

Sets or queries the ID value of LIN trigger when the trigger condition is "Data & ID".

Parameter

| Name | Type | Range | Default |
|------|---------|---------|---------|
| <id> | Integer | 0 to 63 | 0 |

Remarks

N/A

Return Format

The query returns an integer ranging from 0 to 63.

Example

```
:TRIGger:LIN:ID 4             /*Sets the ID value of LIN trigger
to 4.*/*
:TRIGger:LIN:ID?             /*The query returns 4.*/*
```

3.28.24.9 :TRIGger:LIN:DATA**Syntax**

```
:TRIGger:LIN:DATA <data>
```

```
:TRIGger:LIN:DATA?
```

Description

Sets or queries the data value of LIN trigger when the trigger condition is "Data".

Parameter

| Name | Type | Range | Default |
|--------|---------|-------------------------|---------|
| <data> | Integer | Refer to <i>Remarks</i> | 0 |

Remarks

The range of the data value of LIN trigger is related to the value of data bytes. The maximum number of bytes can be set to 8, i.g. 64-bit binary data. Therefore, the range of <data> is from 0 to $2^{64}-1$.

Return Format

The query returns an integer ranging from 0 to $2^{64}-1$.

Example

```
:TRIGger:LIN:DATA 100 /*Sets the data value of LIN trigger to
100 when the trigger condition is "Data".*/
:TRIGger:LIN:DATA? /*The query returns 100.*/
```

3.28.24.10 :TRIGger:LIN:CURRbit**Syntax**

```
:TRIGger:LIN:CURRbit <currbit>
```

```
:TRIGger:LIN:CURRbit?
```

Description

Sets or queries the current bit of the LIN trigger data.

Parameter

| Name | Type | Range | Default |
|-----------|---------|---------|---------|
| <currbit> | Integer | 0 to 39 | 0 |

Remarks

After configuring the settings for this command, you can send the *:TRIGger:LIN:CODE* command to set or modify the set bit data.

Return Format

The query returns an integer ranging from 0 to 39.

Example

```
:TRIGger:LIN:CURRbit 8 /*Sets the current bit of LIN trigger
data to 8. That is, the oscilloscope triggers on the 9th bit of
LIN trigger data.*/
:TRIGger:SPI:CURRbit? /*The query returns 8.*/
```

3.28.24.11 :TRIGger:LIN:CODE

Syntax

```
:TRIGger:LIN:CODE <code>
```

```
:TRIGger:LIN:CODE?
```

Description

Sets or queries the data value of a certain bit of LIN trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|-----------|---------|
| <code> | Discrete | {0 1 255} | 255 |

Remarks

When <code> is set to 255, it indicates the data value can be any value.

After sending the *:TRIGger:LIN:CURRbit* command to set the specified bit, you can send this command to query or modify the value of the specified data bit.

Return Format

The query returns 0, 1, or 255.

Example

```
:TRIGger:LIN:CODE 0 /*Sets the data value to 0.*/
:TRIGger:LIN:CODE? /*The query returns 0.*/
```

3.28.25 :TRIGger:IIS

The :TRIGger:IIS commands are used to set relevant parameters for the I2S trigger.



NOTE

The DHO1000 series oscilloscope does not support :TRIGger:IIS commands.

The :TRIGger:IIS commands are optional configurations for the DHO4000 series oscilloscope.

3.28.25.1 :TRIGger:IIS:SOURce:CLOCK

Syntax

```
:TRIGger:IIS:SOURce:CLOCK <source>
```

```
:TRIGger:IIS:SOURce:CLOCK?
```

Description

Sets or queries the clock source of the I2S trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:IIS:SOURce:CLOCK CHANnel2      /*Sets clock source to
CHANnel2.*/
:TRIGger:IIS:SOURce:CLOCK?              /*The query returns CHAN2.*/
```

3.28.25.2 :TRIGger:IIS:CLEVel**Syntax**

```
:TRIGger:IIS:CLEVel <level>
```

```
:TRIGger:IIS:CLEVel?
```

Description

Sets or queries the trigger level of the clock line source in I2S trigger. The unit is V.

Parameter

| Name | Type | Range | Default |
|---------|------|--|---------|
| <level> | Real | (-4.5 x VerticalScale - Offset) to (4.5 x VerticalScale - Offset) | 0 V |

Remarks

N/A

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:IIS:CLEVel 0.01      /*Sets the trigger level of I2S
trigger to 0.01 V.*/
:TRIGger:IIS:CLEVel?          /*The query returns 1.000000E-2.*/
```

3.28.25.3 :TRIGger:IIS:SOURce:WSElect**Syntax**

```
:TRIGger:IIS:SOURce:WSElect <SOURCE>
```

```
:TRIGger:IIS:SOURce:WSElect?
```

Description

Sets or queries the audio channel of the I2S trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---------------------------------------|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel2 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:IIS:SOURce:WSElect CHANnel2 /*Sets the audio channel
to CHANnel2.*/
:TRIGger:IIS:SOURce:WSElect? /*The query returns
CHAN2.*/
```

3.28.25.4 :TRIGger:IIS:SLEVel**Syntax**

```
:TRIGger:IIS:SLEVel </level>
```

```
:TRIGger:IIS:SLEVel?
```

Description

Sets or queries the trigger level of the frame clock line source in I2S trigger. The unit is V.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | (-4.5 x VerticalScale - Offset) to (4.5 x VerticalScale - Offset) | - |

Remarks

N/A

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:IIS:SLEVel 0.01 /*Sets the trigger level to 10 mV.*/
:TRIGger:IIS:SLEVel? /*The query returns 1.000000E-2.*/
```

3.28.25.5 :TRIGger:IIS:SOURce:DATA**Syntax**

```
:TRIGger:IIS:SOURce:DATA <source>
```

```
:TRIGger:IIS:SOURce:DATA?
```

Description

Sets or queries the data source of the I2S trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---------------------------------------|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel3 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:IIS:SOURce:DATA CHANnel2 /*Sets data source to
CHANnel2.*/
:TRIGger:IIS:SOURce:DATA? /*The query returns CHAN2.*/
```

3.28.25.6 :TRIGger:IIS:DLEVel**Syntax**

```
:TRIGger:IIS:DLEVel <level>
```

```
:TRIGger:IIS:DLEVel?
```

Description

Sets or queries the trigger level of the data line source in I2S trigger. The unit is V.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | (-4.5 x VerticalScale - Offset) to (4.5 x VerticalScale - Offset) | - |

Remarks

N/A

Return Format

The query returns the trigger level of data line source in scientific notation.

Example

```
:TRIGger:IIS:DLEVel 0.16 /*Sets the trigger level of data line
source to 160 mV.*/
:TRIGger:IIS:DLEVel? /*The query returns 1.60000E-1.*/
```

3.28.25.7 :TRIGger:IIS:CLOCK:SLOPe**Syntax**

```
:TRIGger:IIS:CLOCK:SLOPe <slope>
```

```
:TRIGger:IIS:CLOCK:SLOPe?
```

Description

Sets or queries the type of the clock edge of I2S trigger.

Parameter

| Name | Type | Range | Default |
|---------|----------|---------------------|----------|
| <slope> | Discrete | {NEGative POSitive} | POSitive |

Remarks

N/A

Return Format

The query returns NEG or POS.

Example

```
:TRIGger:IIS:CLOCK:SLOPe NEGative /*Sets the clock edge to
NEGative.*/
:TRIGger:IIS:CLOCK:SLOPe? /*The query returns NEG.*/
```

3.28.25.8 :TRIGger:IIS:AUDio

Syntax

```
:TRIGger:IIS:AUDio <audio>
```

```
:TRIGger:IIS:AUDio?
```

Description

Sets or queries the audio state of I2S trigger.

Parameter

| Name | Type | Range | Default |
|---------|----------|---------------------|---------|
| <audio> | Discrete | {RIGHT LEFT EITHER} | LEFT |

Remarks

- **RIGHT:** data of the right channel.
- **LEFT:** data of the left channel.
- **EITHER:** data of either of the channel.

Return Format

The query returns RIGH, LEFT, or EITH.

Example

```
:TRIGger:IIS:AUDio RIGHT /*Sets the I2S audio to RIGHT.*/
:TRIGger:IIS:AUDio? /*The query returns RIGH.*/
```

3.28.25.9 :TRIGger:IIS:WHEN

Syntax

```
:TRIGger:IIS:WHEN <operator>
```

```
:TRIGger:IIS:WHEN?
```

Description

Sets or queries the trigger condition of I2S trigger.

Parameter

| Name | Type | Range | Default |
|------------|----------|--|---------|
| <operator> | Discrete | {EQUAL NOTEQUAL LESSthan GREaterthan INRange OUTRange} | EQUAL |

Remarks

- **EQUal:** triggers when the channel's data value equals the set value.
- **NOTequal:** triggers when the channel's data value does not equal the set value.
- **LESSthan:** triggers when the channel's data value is smaller than the set value.
- **GREaterthan:** triggers when the channel's data value is greater than the set value.
- **INRange:** triggers when the channel's data value is smaller than the set upper limit value and greater than the set lower limit value.
- **OUTRange:** triggers when the channel's data value is greater than the set upper limit value or smaller than the set lower limit value.

Return Format

The query returns EQU, NOT, LESS, GRE, INR, or OUTR.

Example

```
:TRIGger:IIS:WHEN NOTequal /*Sets the trigger condition of
I2S trigger to NOTequal.*/
:TRIGger:IIS:WHEN? /*The query returns NOT.*/
```

3.28.25.10 :TRIGger:IIS:UWIDth**Syntax**

```
:TRIGger:IIS:UWIDth <uwidth>
```

```
:TRIGger:IIS:UWIDth?
```

Description

Sets or queries the user width of I2S trigger.

Parameter

| Name | Type | Range | Default |
|----------|---------|-------------------------|---------|
| <uwidth> | Integer | Refer to <i>Remarks</i> | - |

Remarks

The user width ranges from 4 to width of I2S trigger. To set or query the width of I2S trigger, send the *:TRIGger:IIS:WIDTh* command.

Return Format

The query returns an integer ranging from 4 to 32.

Example

```
:TRIGger:IIS:UWIDth 10 /*Sets the user width of I2S trigger to
10.*/
:TRIGger:IIS:UWIDth? /*The query returns 10.*/
```

3.28.25.11 :TRIGger:IIS:WIDTh**Syntax**

```
:TRIGger:IIS:WIDTh <uwidth>
:TRIGger:IIS:WIDTh?
```

Description

Sets or queries the width of I2S trigger.

Parameter

| Name | Type | Range | Default |
|----------|---------|---------|---------|
| <uwidth> | Integer | 4 to 32 | 4 |

Remarks

N/A

Return Format

The query returns an integer ranging from 4 to 32.

Example

```
:TRIGger:IIS:WIDTh 10 /*Sets the width of I2S trigger to 10.*/
:TRIGger:IIS:WIDTh? /*The query returns 10.*/
```

3.28.25.12 :TRIGger:IIS:ALIGnment**Syntax**

```
:TRIGger:IIS:ALIGnment <setting>
:TRIGger:IIS:ALIGnment?
```

Description

Sets or queries the alignment mode of the I2S trigger.

Parameter

| Name | Type | Range | Default |
|-----------|----------|-------------|---------|
| <setting> | Discrete | {LJ RJ IIS} | IIS |

Remarks

- **LJ:** data transmission (MSB first) begins at the edge of the WS transition.
- **RJ:** data transmission (MSB first) is right-justified to the WS transition.
- **IIS:** data transmission (MSB first) begins at the second edge of the WS transition.

Return Format

The query returns LJ, RJ, or IIS.

Example

```
:TRIGger:IIS:ALIGNment LJ          /*Sets the alignment mode of
I2S trigger to LJ.*/
:TRIGger:IIS:ALIGNment?           /*The query returns LJ.*/
```

3.28.25.13 :TRIGger:IIS:DATA**Syntax**

```
:TRIGger:IIS:DATA <data>
```

```
:TRIGger:IIS:DATA?
```

Description

Sets or queries the data value of I2S trigger when the trigger condition is "=" or "≠".

Parameter

| Name | Type | Range | Default |
|--------|---------|-----------------|---------|
| <data> | Integer | 0 to $2^{32}-1$ | 0 |

Remarks

The settable range of <data> is affected by the byte length. The maximum byte length can be set to 4, i.g. 32-bit binary data. Therefore, the range of <data> is from 0 to $2^{32}-1$.

Return Format

The query returns an integer ranging from 0 to $2^{32}-1$.

Example

```
:TRIGger:IIS:DATA 10 /*Sets the data value of I2S trigger to 10
when the trigger condition is "=" or "!=".*/
:TRIGger:IIS:DATA? /*The query returns 10.*/
```

3.28.25.14 :TRIGger:IIS:DMIN**Syntax**

```
:TRIGger:IIS:DMIN <datamin>
```

```
:TRIGger:IIS:DMIN?
```

Description

Sets or queries the specified bit for the data min. of I2S trigger.

Parameter

| Name | Type | Range | Default |
|-----------|---------|-------------------------|---------|
| <datamin> | Integer | Refer to <i>Remarks</i> | 0 |

Remarks

The range of the parameter is related to the user width of I2S trigger set by the *:TRIGger:IIS:UWIDTH* command.

After running this command, you can send the *:TRIGger:IIS:CODE* command to set or query the specified bit of the data min. value.

Return Format

The query returns an integer ranging from 0 to 39.

Example

```
:TRIGger:IIS:DMIN 3 /*Sets the current bit for the data min. of
I2S trigger to 3.*/
:TRIGger:IIS:DMIN? /*The query returns 3.*/
```

3.28.25.15 :TRIGger:IIS:DMAX**Syntax**

```
:TRIGger:IIS:DMAX <datamax>
```

```
:TRIGger:IIS:DMAX?
```

Description

Sets or queries the data max. value of the I2S trigger.

Parameter

| Name | Type | Range | Default |
|-----------|---------|-------------------------|---------|
| <datamax> | Integer | Refer to <i>Remarks</i> | 0 |

Remarks

The range of the parameter is related to the user width of I2S trigger set by the `:TRIGger:IIS:UWIDTH` command.

After running this command, you can send the `:TRIGger:IIS:CODE` command to set or query the specified bit of the data max. value.

Return Format

The query returns an integer ranging from 0 to 39.

Example

```
:TRIGger:IIS:DMAX 3 /*Sets the data max. value to 3.*/
:TRIGger:IIS:DMAX? /*The query returns 3.*/
```

3.28.25.16 :TRIGger:IIS:CODE**Syntax**

```
:TRIGger:IIS:CODE <code>
```

```
:TRIGger:IIS:CODE?
```

Description

Sets or queries the data value of a certain bit of I2S trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|-----------|---------|
| <code> | Discrete | {0 1 255} | 255 |

Remarks

When <code> is set to 255, it indicates the data value can be any value.

After sending the `:TRIGger:IIS:DMAX` or `:TRIGger:IIS:DMIN` command to set the specified bit, you can send this command to query or modify the value of the specified data bit.

Return Format

The query returns 0, 1, or 255.

Example

```
:TRIGger:IIS:CODE 0 /*Sets the data value to 0.*/
:TRIGger:IIS:CODE? /*The query returns 0.*/
```

3.28.26 :TRIGger:FLEXray

The :TRIGger:FLEXray commands are used to set the relevant parameters for the FLEXray trigger.

**NOTE**

The DHO1000 series oscilloscope does not support :TRIGger:FLEXray commands.

The :TRIGger:FLEXray commands are optional configurations for the DHO4000 series oscilloscope.

3.28.26.1 :TRIGger:FLEXray:SOURce**Syntax**

```
:TRIGger:FLEXray:SOURce <source>
:TRIGger:FLEXray:SOURce?
```

Description

Sets or queries the trigger source of FlexRay trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:FLEXray:SOURce CHANnel2 /*Sets the trigger source of
FlexRay trigger to CHANnel2.*/
:TRIGger:FLEXray:SOURce? /*The query returns CHAN2.*/
```

3.28.26.2 :TRIGger:FLEXray:LEVel**Syntax**

```
:TRIGger:FLEXray:LEVel </evel>
```

:TRIGger:FLEXray:LEVel?

Description

Sets or queries the trigger level of FlexRay trigger. Its unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Analog channel: (-4.5 x VerticalScale - OFFSet) to (4.5 x VerticalScale - OFFSet) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the trigger level in scientific notation.

Example

```
:TRIGger:FLEXray:LEVel 0.16 /*Sets the trigger level to 160
mV.*/
:TRIGger:FLEXray:LEVel? /*The query returns 1.600000E-1.*/
```

3.28.26.3 :TRIGger:FLEXray:BAUD

Syntax

:TRIGger:FLEXray:BAUD <baud>

:TRIGger:FLEXray:BAUD?

Description

Sets or queries the signal rate of FlexRay trigger. The default unit is bps.

Parameter

| Name | Type | Range | Default |
|--------|----------|----------------------------|----------|
| <baud> | Discrete | {2500000 5000000 10000000} | 10000000 |

Remarks

N/A

Return Format

The query returns 2500000, 5000000, or 10000000.

Example

```
:TRIGger:FLEXray:BAUD 5000000 /*Sets the signal rate to
5000000 bps.*/
:TRIGger:FLEXray:BAUD? /*The query returns 5000000.*/
```

3.28.26.4 :TRIGger:FLEXray:CH**Syntax**

```
:TRIGger:FLEXray:CH <ch>
```

```
:TRIGger:FLEXray:CH?
```

Description

Sets or queries the channel of FlexRay trigger.

Parameter

| Name | Type | Range | Default |
|------|----------|-------|---------|
| <ch> | Discrete | {A B} | A |

Remarks

N/A

Return Format

The query returns A or B.

Example

```
:TRIGger:FLEXray:CH B /*Sets the channel of FlexRay trigger
to Channel B.*/
:TRIGger:FLEXray:CH? /*The query returns B.*/
```

3.28.26.5 :TRIGger:FLEXray:WHEN**Syntax**

```
:TRIGger:FLEXray:WHEN <cond>
```

```
:TRIGger:FLEXray:WHEN?
```

Description

Sets or queries the trigger condition of FLEXray trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|--------------------------|---------|
| <cond> | Discrete | {FRAME SYMBOL ERROR TSS} | FRAME |

Remarks

- **FRAME:** triggers on the frame of the FlexRay bus.
- **SYMBOL:** triggers on the Channel Idle Delimiter (CID), Collision Avoidance Symbol (CAS), Media Access Test Symbol (MTS), and Wakeup Pattern (WUP) of the FlexRay bus.
- **ERROR:** triggers when an error occurs to the FlexRay bus, including Head CRC Err, Tail CRC Err, Decode Err, and Random Err.
- **TSS:** triggers on the transmission start sequence of the FlexRay bus.

Return Format

The query returns FRAM, SYMB, ERR, or TSS.

Example

```
:TRIGger:FLEXray:WHEN TSS          /*Sets the trigger condition to
TSS.*/
:TRIGger:FLEXray:WHEN?              /*The query returns TSS.*/
```

3.28.26.6 :TRIGger:FLEXray:POS**Syntax**

```
:TRIGger:FLEXray:POS <pos>
```

```
:TRIGger:FLEXray:POS?
```

Description

Sets or queries the position of FlexRay trigger when the trigger condition is Position.

Parameter

| Name | Type | Range | Default |
|-------|----------|-------------------|---------|
| <pos> | Discrete | {TSS FSS FES DTS} | TSS |

Remarks

- **TSS:** Transmit Start Sequence.

- **FSS:** Frame Start Sequence.
- **FES:** Frame End Sequence.
- **DTS:** Dynamic Trailing Sequence.

Return Format

The query returns TSS, FSS, FES, or DTS.

Example

```
:TRIGger:FLEXray:POS TSS          /*Sets the position of FlexRay
trigger to TSS when the trigger condition is Position.*/
:TRIGger:FLEXray:POS?             /*The query returns TSS.*/
```

3.28.26.7 :TRIGger:FLEXray:FRAMe

Syntax

```
:TRIGger:FLEXray:FRAMe? <frame>
```

```
:TRIGger:FLEXray:FRAMe?
```

Description

Sets or queries the frame type of FlexRay trigger.

Parameter

| Name | Type | Range | Default |
|---------|----------|----------------------|---------|
| <frame> | Discrete | {NULL SYNC STAR ANY} | NULL |

Remarks

N/A

Return Format

The query returns NULL, SYNC, STAR, or ANY.

Example

```
:TRIGger:FLEXray:FRAMe SYNC       /*Sets the frame type of
FlexRay trigger to SYNC.*/
:TRIGger:FLEXray:FRAMe?           /*The query returns SYNC.*/
```

3.28.26.8 :TRIGger:FLEXray:SYMBOL

Syntax

```
:TRIGger:FLEXray:SYMBOL <symbol>
```

```
:TRIGger:FLEXray:SYMBOL?
```

Description

Sets or queries the symbol type of FlexRay trigger when the trigger condition is set to Symbol.

Parameter

| Name | Type | Range | Default |
|----------|----------|-----------|---------|
| <symbol> | Discrete | {CAS WUS} | CAS |

Remarks

N/A

Return Format

The query returns CAS or WUS.

Example

```
:TRIGger:FLEXray:SYMBOL WUS          /*Sets the symbol type of
FlexRay trigger to WUS when the trigger condition is set to
Symbol.*/
:TRIGger:FLEXray:SYMBOL?              /*The query returns WUS.*/
```

3.28.26.9 :TRIGger:FLEXray:ERRor**Syntax**

```
:TRIGger:FLEXray:ERRor <err>
```

```
:TRIGger:FLEXray:ERRor?
```

Description

Sets or queries the error type of FlexRay trigger when the trigger condition is Error.

Parameter

| Name | Type | Range | Default |
|-------|----------|---------------------------|---------|
| <err> | Discrete | {HEAD TRAIler DECode ANY} | HEAD |

Remarks

N/A

Return Format

The query returns HEAD, TRAIler, DECode, or ANY.

Example

```
:TRIGger:FLEXray:ERRor DECode          /*Sets the error type of
FlexRay trigger to DECode when the trigger condition is Error.*/
:TRIGger:FLEXray:ERRor?                /*The query DECode.*/
```

3.28.26.10 :TRIGger:FLEXray:DEFine**Syntax**

```
:TRIGger:FLEXray:DEFine? <type>
```

```
:TRIGger:FLEXray:DEFine?
```

Description

Sets or queries to use ID or Cyc Count to define the trigger condition of FlexRay trigger when the trigger condition is set to Frame.

Parameter

| Name | Type | Range | Default |
|--------|----------|------------|---------|
| <type> | Discrete | {CYCLe ID} | - |

Remarks

- **CYCLe:** indicates Cyc Count.
- **ID:** indicates Frame ID.

Return Format

The query returns CYCL or ID.

Example

```
:TRIGger:FLEXray:DEFine? CYCLe        /*Sets to use Cyc Count to
define the trigger condition of FlexRay trigger when the trigger
condition is set to Frame.*/
:TRIGger:FLEXray:DEFine?              /*The query returns CYCL.*/
```

3.28.26.11 :TRIGger:FLEXray:CYCComp**Syntax**

```
:TRIGger:FLEXray:CYCComp <cycmax>
```

```
:TRIGger:FLEXray:CYCComp?
```

Description

Sets or queries the CYC comparison condition of FlexRay trigger when the trigger condition is set to Frame.

Parameter

| Name | Type | Range | Default |
|----------|----------|-----------------------------|---------|
| <cycmax> | Discrete | {EQU NOT GRE LESS INR OUTR} | EQU |

Remarks

- **EQU:** =
- **NOT:** ≠
- **GRE:** >
- **LESS:** <
- **INR:** ><
- **OUTR:** <>

Return Format

The query returns EQU, NOT, GRE, LESS, INR, or OUTR.

Example

```
:TRIGger:FLEXray:CYCComp NOT /*Sets the CYC comparison
condition of FlexRay trigger to NOT.*/
:TRIGger:FLEXray:CYCComp? /*The query returns NOT.*/
```

3.28.26.12 :TRIGger:FLEXray:MAXCy**Syntax**

```
:TRIGger:FLEXray:MAXCy <cycmax>
```

```
:TRIGger:FLEXray:MAXCy?
```

Description

Sets or queries the upper limit of cycle count of FlexRay trigger.

Parameter

| Name | Type | Range | Default |
|----------|---------|---------|---------|
| <cycmax> | Integer | 0 to 63 | 0 |

Remarks

The upper limit of cycle count of FlexRay trigger configured by this command should be greater than or equal to the cycle count lower limit configured by

the `:TRIGger:FLEXray:MINCy` command, otherwise, the lower limit will be changed with the modification of the upper limit.

Return Format

The query returns an integer ranging from 0 to 63.

Example

```
:TRIGger:FLEXray:MAXCy 10 /*Sets the upper limit of cycle
count of FlexRay trigger to 10.*/
:TRIGger:FLEXray:MAXCy? /*The query returns 10.*/
```

3.28.26.13 :TRIGger:FLEXray:MINCy

Syntax

```
:TRIGger:FLEXray:MINCy <cymin>
```

```
:TRIGger:FLEXray:MINCy?
```

Description

Sets or queries the lower limit of cycle count of FlexRay trigger.

Parameter

| Name | Type | Range | Default |
|---------|---------|---------|---------|
| <cymin> | Integer | 0 to 63 | 0 |

Remarks

The lower limit of cycle count of FlexRay trigger configured by this command should be smaller than or equal to the cycle count upper limit configured by the `:TRIGger:FLEXray:MAXCy` command, otherwise, the upper limit will be changed with the modification of the lower limit.

Return Format

The query returns an integer ranging from 0 to 63.

Example

```
:TRIGger:FLEXray:MINCy 10 /*Sets the lower limit of cycle
count of FlexRay trigger to 10.*/
:TRIGger:FLEXray:MINCy? /*The query returns 10.*/
```

3.28.26.14 :TRIGger:FLEXray:IDCMP

Syntax

```
:TRIGger:FLEXray:IDCMP <idcomp>
```

```
:TRIGger:FLEXray:IDCMP?
```

Description

Sets or queries the ID comparison of FlexRay trigger when the trigger condition is set to Frame or Symbol.

Parameter

| Name | Type | Range | Default |
|----------|----------|--|---------|
| <idcomp> | Discrete | {EQUAL NOTequal GREaterthan LESSthan INRange OUTRange} | EQUAL |

Remarks

- **EQUAL:** =
- **NOTequal:** ≠
- **GREaterthan:** >
- **LESSthan:** <
- **INRange:** ><
- **OUTRange:** <>

Return Format

The query returns EQU, NOT, GRE, LESS, INR, or OUTR.

Example

```
:TRIGger:FLEXray:IDCmp GREaterthan /*Sets the ID comparison
condition of FlexRay trigger to GREaterthan.*/
:TRIGger:FLEXray:IDCmp? /*The query returns GRE.*/
```

3.28.26.15 :TRIGger:FLEXray:MINid**Syntax**

```
:TRIGger:FLEXray:MINid <datamin>
```

```
:TRIGger:FLEXray:MINid?
```

Description

Sets or queries the ID Min value of FlexRay trigger when the trigger condition is set to Frame or Symbol.

Parameter

| Name | Type | Range | Default |
|-----------|---------|------------|---------|
| <datamin> | Integer | 0 to 1,023 | 0 |

Remarks

N/A

Return Format

The returns an integer ranging from 0 to 1,023.

Example

```
:TRIGger:FLEXray:MINid 10 /*Sets the ID Min value of
FlexRay trigger to 10 when the trigger condition is set to Frame
or Symbol.*/
:TRIGger:FLEXray:MINid? /*The query returns 10.*/
```

3.28.26.16 :TRIGger:FLEXray:MAXid**Syntax**

```
:TRIGger:FLEXray:MAXid <datamax>
```

```
:TRIGger:FLEXray:MAXid?
```

Description

Sets or queries the ID upper limit of FlexRay trigger.

Parameter

| Name | Type | Range | Default |
|-----------|---------|-----------|---------|
| <datamax> | Integer | 0 to 1023 | 0 |

Remarks

N/A

Return Format

The returns an integer ranging from 0 to 1023.

Example

```
:TRIGger:FLEXray:MAXid 100 /*Sets the ID upper limit of
FlexRay trigger to 100.*/
:TRIGger:FLEXray:MAXid? /*The query returns 10.*/
```


3.28.27 :TRIGger:M1553

The :TRIGger:M1553 commands are used to set relevant parameters for the M1553 trigger.



NOTE

The DHO1000 series oscilloscope does not support :TRIGger:M1553 commands.

The :TRIGger:M1553 commands are optional configurations for the DHO4000 series oscilloscope.

3.28.27.1 :TRIGger:M1553:SOURce

Syntax

```
:TRIGger:M1553:SOURce <source>
```

```
:TRIGger:M1553:SOURce?
```

Description

Sets or queries the trigger source of M1553 trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4} | CHANnel1 |

Remarks

N/A

Return Format

The query returns CHAN1, CHAN2, CHAN3, or CHAN4.

Example

```
:TRIGger:M1553:SOURce CHANnel2 /*Sets the trigger source to
CHANnel2.*/
:TRIGger:M1553:SOURce? /*The query returns CHAN2.*/
```

3.28.27.2 :TRIGger:M1553:WINDow

Syntax

```
:TRIGger:M1553:WINDow <window>
```

```
:TRIGger:M1553:WINDow?
```

Description

Sets or queries the type of the trigger level of M1553 trigger.

Parameter

| Name | Type | Range | Default |
|----------|----------|-------------|---------|
| <window> | Discrete | {TA TB TAB} | TA |

Remarks

- **TA:** Level A. Only adjusts the upper limit of the trigger level; the lower limit of the trigger level remains unchanged.
- **TB:** Level B. Only adjusts the lower limit of the trigger level; the upper limit of the trigger level remains unchanged.
- **TAB:** adjusts the upper and lower trigger level synchronously. The trigger level deviation (the difference between the upper limit and lower limit) remains unchanged.

Return Format

The query returns TA, TB, or TAB.

Example

```
:TRIGger:M1553:WINDow TAB /*Sets the type of the trigger level of
M1553 trigger to TAB.*/
:TRIGger:M1553:WINDow? /*The query returns TAB.*/
```

3.28.27.3 :TRIGger:M1553:ALEVel**Syntax**

```
:TRIGger:M1553:ALEVel <level>
```

```
:TRIGger:M1553:ALEVel?
```

Description

Sets or queries the upper limit of the trigger level of M1553 trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|---|---------|
| <level> | Real | Lower limit to (4.5 x VerticalScale - OFFSet) | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the upper limit of the trigger level in scientific notation.

Example

```
:TRIGger:M1553:ALEVEL 0.16 /*Sets the upper limit of the
trigger level to 160 mV.*/
:TRIGger:M1553:ALEVEL? /*The query returns 1.600000E-1.*/
```

3.28.27.4 :TRIGger:M1553:BLEVel**Syntax**

```
:TRIGger:M1553:BLEVel </level>
```

```
:TRIGger:M1553:BLEVel?
```

Description

Sets or queries the lower limit of the trigger level of Delay trigger. The unit is the same as that of the current amplitude.

Parameter

| Name | Type | Range | Default |
|---------|------|--|---------|
| <level> | Real | (-4.5 x VerticalScale - OFFSet) to upper limit | 0 V |

Remarks

For VerticalScale, refer to the `:CHANnel<n>:SCALE` command. For OFFSet, refer to the `:CHANnel<n>:OFFSet` command.

Return Format

The query returns the lower limit of the trigger level in scientific notation.

Example

```
:TRIGger:M1553:BLEVel 0.05 /*Sets the lower limit of the trigger
level to 50 mV.*/
:TRIGger:M1553:BLEVel? /*The query returns 5.000000E-2.*/
```

3.28.27.5 :TRIGger:M1553:POLarity**Syntax**

```
:TRIGger:M1553:POLarity <polarity>
```

```
:TRIGger:M1553:POLarity?
```

Description

Sets or queries the polarity of M1553 trigger.

Parameter

| Name | Type | Range | Default |
|------------|----------|---------------------|----------|
| <polarity> | Discrete | {POSitive NEGative} | POSitive |

Return Format

The query returns POS or NEG.

Example

```
:TRIGger:M1553:POLarity POSitive /*Sets the polarity of M1553
trigger to POSitive.*/
:TRIGger:M1553:POLarity? /*The query returns POS.*/
```

3.28.27.6 :TRIGger:M1553:WHEN**Syntax**

```
:TRIGger:M1553:WHEN <when>
```

```
:TRIGger:M1553:WHEN?
```

Description

Sets or queries the trigger condition of M1553 trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|-----------------------------------|-----------|
| <when> | Discrete | {SYNCbreak DATA CMD STATus ERRor} | SYNCbreak |

Remarks

- **SYNCbreak:** triggers on the specified sync type.
- **DATA:** triggers on the specified data word.
- **CMD:** triggers on the specified remote terminal address.
- **STATus:** triggers on the RTA and the remaining 11 bits.
- **ERRor:** triggers on the specified error type.

Return Format

The query returns SYNC, DATA, CMD, STAT, or ERR.

Example

```
:TRIGger:M1553:WHEN CMD /*Sets the trigger condition to
CMD.* /
:TRIGger:M1553:WHEN? /*The query returns CMD.* /
```

3.28.27.7 :TRIGger:M1553:SYNC**Syntax**

```
:TRIGger:M1553:SYNC <sync>
```

```
:TRIGger:M1553:SYNC?
```

Description

Sets or queries the sync type of M1553 trigger when the trigger condition is Sync.

Parameter

| Name | Type | Range | Default |
|--------|----------|-------------------|---------|
| <sync> | Discrete | {DATA STATus ALL} | - |

Remarks

- **DATA:** indicates data frame sync.
- **STATus:** indicates commands/status sync.
- **ALL:** indicates the all frame sync.

Return Format

The query returns DATA, STATus, or ALL.

Example

```
:TRIGger:M1553:SYNC DATA /*Sets the sync type of M1553 trigger
to DATA.*/
:TRIGger:M1553:SYNC? /*The query returns DATA.*/
```

3.28.27.8 :TRIGger:M1553:ERRor**Syntax**

```
:TRIGger:M1553:ERRor <err>
```

```
:TRIGger:M1553:ERRor?
```

Description

Sets or queries the M1553 trigger error type.

Parameter

| Name | Type | Range | Default |
|-------|----------|------------|---------|
| <err> | Discrete | {SYNC ERR} | SYNC |

Remarks

N/A

Return Format

The query returns SYNC or ERR.

Example

```
:TRIGger:M1553:ERRor ERR /*Sets the M1553 trigger error
type to ERR.*/
:TRIGger:M1553:ERRor? /*The query returns ERR.*/
```

3.28.27.9 :TRIGger:M1553:DATComp**Syntax**

```
:TRIGger:M1553:DATComp <datacomp>
```

```
:TRIGger:M1553:DATComp?
```

Description

Sets or queries the comparison type of M1553 trigger when the trigger condition is "DATA".

Parameter

| Name | Type | Range | Default |
|------------|----------|--|---------|
| <datacomp> | Discrete | {EQUAL NOTequal GREaterthan LESSthan INRange OUTRange} | EQUAL |

Remarks

- **EQUAL:** =
- **NOTequal:** ≠
- **GREaterthan:** >
- **LESSthan:** <
- **INRange:** > <
- **OUTRange:** < >

Return Format

The query returns EQU, NOT, GRE, LESS, INR, or OTR.

Example

```
:TRIGger:M1553:DATComp NOTequal /*Sets the comparison type
of M1553 trigger to NOTequal when the trigger condition is
"DATA".*/
:TRIGger:M1553:DATComp? /*The query returns NOT.*/
```

3.28.27.10 :TRIGger:M1553:DATValue**Syntax**

```
:TRIGger:M1553:DATValue <data>
```

```
:TRIGger:M1553:DATValue?
```

Description

Sets or queries the data value of 1553 trigger.

Parameter

| Name | Type | Range | Default |
|--------|---------|------------|---------|
| <data> | Integer | 0 to 65535 | 0 |

Remarks

N/A

Return Format

The query returns an integer ranging from 0 to 65535.

Example

```
:TRIGger:M1553:DATValue 100 /*Sets the data value of M1553
trigger to 100.*/
:TRIGger:M1553:DATValue? /*The query returns 100.*/
```

3.28.27.11 :TRIGger:M1553:DMAX**Syntax**

```
:TRIGger:M1553:DMAX <datamax>
```

```
:TRIGger:M1553:DMAX?
```

Description

Sets or queries the bit of data max. of M1553 trigger.

Parameter

| Name | Type | Range | Default |
|-----------|---------|---------|---------|
| <datamax> | Integer | 0 to 19 | 0 |

Remarks

After configuring the specified bit, you can run the *:TRIGger:M1553:CODE* command to set or query the value of the specified bit.

Return Format

The query returns an integer ranging from 0 to 19.

Example

```
:TRIGger:M1553:DMAX 10 /*Sets the bit of data max. of M1553
trigger to 10.*/
:TRIGger:M1553:DMAX? /*The query returns 10.*/
```

3.28.27.12 :TRIGger:M1553:DMIN**Syntax**

```
:TRIGger:M1553:DMIN <datamin>
```

```
:TRIGger:M1553:DMIN?
```

Description

Sets or queries the bit of data min. of M1553 trigger.

Parameter

| Name | Type | Range | Default |
|-----------|---------|---------|---------|
| <datamin> | Integer | 0 to 19 | 0 |

Remarks

The bit starts from the zero bit of the Binary format and ends at the last bit of the Hex format. The spaces between strings of Binary and Hex format are ignored.

After configuring the specified bit, you can run the `:TRIGger:M1553:CODE` command to set or query the value of the specified bit.

Return Format

The query returns an integer ranging from 0 to 19.

Example

```
:TRIGger:M1553:DMIN 10 /*Sets the bit of data min. of M1553
trigger to 10.*/
:TRIGger:M1553:DMIN? /*The query returns 10.*/
```

3.28.27.13 :TRIGger:M1553:DRTA**Syntax**

```
:TRIGger:M1553:DRTA <data>
```

```
:TRIGger:M1553:DRTA?
```

Description

Sets or queries the data value of M1553 trigger.

Parameter

| Name | Type | Range | Default |
|--------|---------|--------|---------|
| <data> | Integer | 0 to 6 | 0 |

Remarks

N/A

Return Format

The query returns an integer ranging from 0 to 6.

Example

```
:TRIGger:M1553:DRTA 4 /*Sets the data value of M1553 trigger to
4.*/
:TRIGger:M1553:DRTA? /*The query returns 4.*/
```

3.28.27.14 :TRIGger:M1553:DBIT

Syntax

```
:TRIGger:M1553:DBIT < databit>
```

```
:TRIGger:M1553:DBIT?
```

Description

Sets or queries the bit of bit time of M1553 trigger when the trigger condition is set to "RAT+11Bit".

Parameter

| Name | Type | Range | Default |
|-----------|---------|---------|---------|
| <databit> | Integer | 0 to 13 | 0 |

Remarks

After configuring the specified bit, you can run the `:TRIGger:M1553:CODE` command to set or query the value of the specified bit.

Return Format

The query returns an integer ranging from 0 to 13.

Example

```
:TRIGger:M1553:DBIT 10 /*Sets the bit of the bit time of M1553
trigger to 10.*/
:TRIGger:M1553:DBIT? /*The query returns 10.*/
```

3.28.27.15 :TRIGger:M1553:CODE

Syntax

```
:TRIGger:M1553:CODE < code>
```

```
:TRIGger:M1553:CODE?
```

Description

Sets or queries the data value of a certain bit of M1553 trigger.

Parameter

| Name | Type | Range | Default |
|--------|----------|-----------|---------|
| <code> | Discrete | {0 1 255} | 255 |

Remarks

When <code> is set to 255, it indicates the data value can be any value.

- When you modify the data max. value of M1553 trigger, first send the `:TRIGger:M1553:DMAX` command to set the bit to be modified, and then send this command to modify the value of the bit.
- When you modify the data min. value of M1553 trigger, first send the `:TRIGger:M1553:DMIN` command to set the bit to be modified, and then send this command to modify the value of the bit.
- When you modify the bit time of M1553 trigger, first send the `:TRIGger:M1553:DBIT` command to set the bit to be modified, and then send this command to modify the value of the bit.

Return Format

The query returns 0, 1, or 255.

Example

```
:TRIGger:M1553:CODE 1 /*Sets the data value of a certain bit of  
M1553 trigger to 1.*/  
:TRIGger:M1553:CODE? /*The query returns 1.*/
```

3.29 :WAVeform Commands

The **:WAVeform** commands are used to read waveform data and relevant settings. The `:WAVeform:MODE` command is used to set the reading mode of waveform data. In different modes, the definitions for the parameters are different, as shown in [Figure 3.1](#) and [Figure 3.2](#).

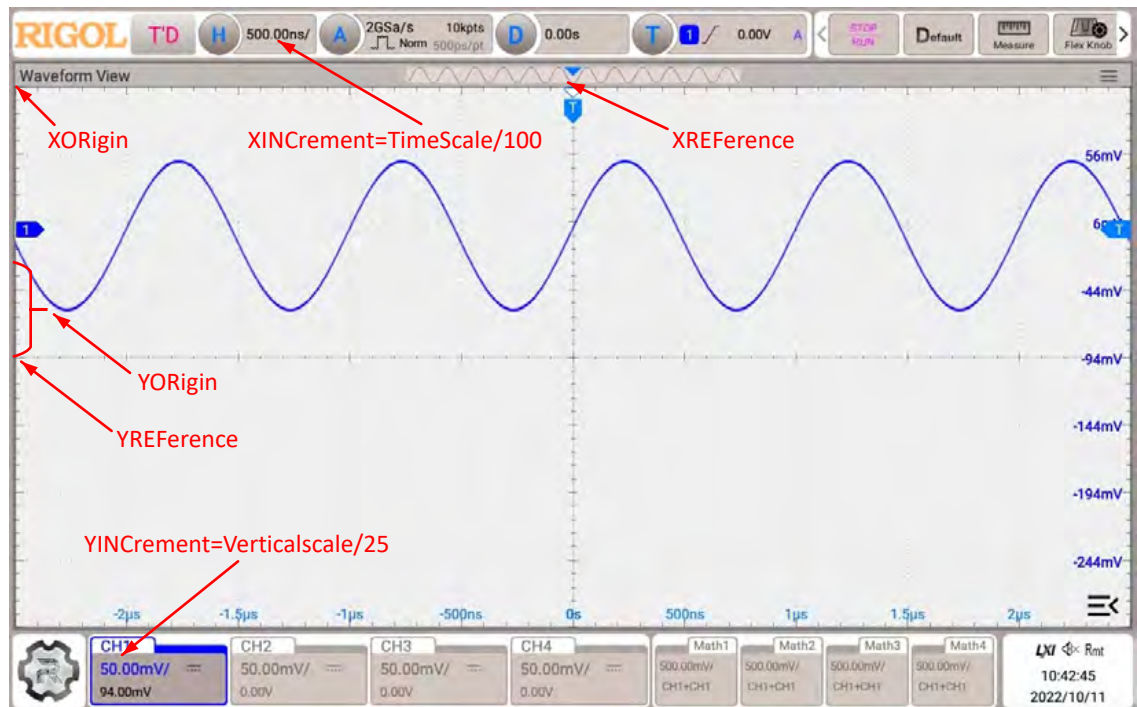


Figure 3.1 Parameter Definitions in NORMAL Mode

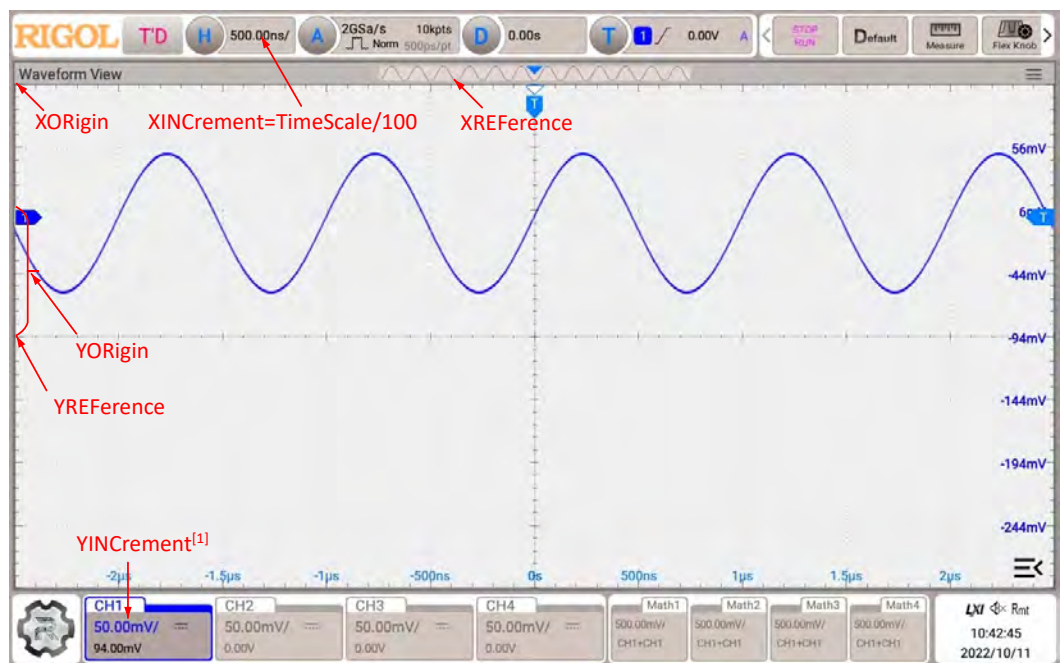


Figure 3.2 Parameter Definitions in RAW Mode



NOTE

[1]: In RAW mode, YINcrement and Verticalscale of the memory waveforms are related to the currently selected Verticalscale.

3.29.1 :WAVeform:SOURce

Syntax

```
:WAVeform:SOURce <source>
```

```
:WAVeform:SOURce?
```

Description

Sets or queries the source channel of waveform data reading.

Parameter

| Name | Type | Range | Default |
|----------|----------|---|----------|
| <source> | Discrete | {CHANnel1 CHANnel2 CHANnel3 CHANnel4 MATH1 MATH2 MATH3 MATH4} | CHANnel1 |

Remarks

When the channel source is set to MATH1-MATH4, *:WAVeform:MODE* can only select the NORMAL mode.

Return Format

The query returns CHAN1, CHAN2, CHAN3, CHAN4, MATH1, MATH2, MATH3, or MATH4.

Example

```
:WAVeform:SOURce CHANnel2 /*Sets the channel source to
CHANnel2.*/
:WAVeform:SOURce? /*The query returns CHAN2.*/
```

3.29.2 :WAVeform:MODE

Syntax

```
:WAVeform:MODE <mode>
```

```
:WAVeform:MODE?
```

Description

Sets or queries the mode of the *:WAVeform:DATA?* command in reading data.

Parameter

| Name | Type | Range | Default |
|--------|----------|----------------------|---------|
| <mode> | Discrete | {NORMAL MAXimum RAW} | NORMAL |

Remarks

- **NORMAL:** reads the waveform data currently displayed on the screen.
- **MAXimum:** reads the waveform data displayed on the screen when the oscilloscope is in the Run state; reads the waveform data from the internal memory when the oscilloscope is in the Stop state.
- **RAW:** reads the waveform data from the internal memory. Note: The data in the internal memory can only be read when the oscilloscope is in the Stop state. You are not allowed to operate the instrument when it is reading data.
- When the channel source is set to MATH, only the NORMAL mode is valid.

Return Format

The query returns NORM, MAX, or RAW.

Example

```
:WAVeform:MODE RAW /*Sets the reading mode of waveform data to
RAW.*/
:WAVeform:MODE? /*The query returns RAW.*/
```

3.29.3 :WAVeform:FORMat**Syntax**

```
:WAVeform:FORMat <format>
```

```
:WAVeform:FORMat?
```

Description

Sets or queries the return format of the waveform data.

Parameter

| Name | Type | Range | Default |
|----------|----------|-------------------|---------|
| <format> | Discrete | {WORD BYTE ASCii} | BYTE |

Remarks

- **WORD:** Each waveform point occupies two bytes (16 bits).
- **BYTE:** Each waveform point occupies one byte (8 bits).
- **ASCii:** The query returns the actual voltage value of each waveform point in scientific notation; and the voltage values are separated by commas.

Return Format

The query returns WORD, BYTE, or ASC.

Example

```
:WAVeform:FORMat WORD /*Sets the returned format of waveform data
to WORD.*/
:WAVeform:FORMat? /*The query returns WORD.*/
```

3.29.4 :WAVeform:POINTs**Syntax**

```
:WAVeform:POINTs <point>
```

```
:WAVeform:POINTs?
```

Description

Sets or queries the number of the waveform points to be read in the current mode.

Parameter

| Name | Type | Range | Default |
|---------|---------|-------------------------|---------|
| <point> | Integer | Refer to <i>Remarks</i> | - |

Remarks

The range of <point> is related to the current reading mode of the waveform data. You can send the *:WAVeform:MODE* command to set or query the reading mode of the waveform data.

- **NORMal:** 1 to 1,000
- **RAW:** 1 to the current maximum memory depth
- **MAXimum:** In RUN state: 1 to the number of effective points on the current screen; In STOP state: 1 to the number of effective points on the current memory

Return Format

The query returns the number of waveform points in integer.

Example

```
:WAVeform:POINTs 100 /*Sets the number of waveform points to be
read to 100.*/
:WAVeform:POINTs? /*The query returns 100.*/
```

3.29.5 :WAVeform:DATA?**Syntax**

```
:WAVeform:DATA?
```

Description

Reads the waveform data.

Parameter

N/A

Remarks

Procedures of reading the waveform data on the screen:

```
:WAV:SOUR CHAN1 /*Sets the channel source to CHANnel1.*/
:WAV:MODE NORMAl /*Sets the waveform reading mode to NORMAl.*/
:WAV:FORM BYTE /*Sets the return format of the waveform data to
BYTE.*/
:WAV:DATA? /*Reads the waveform data on the screen.*/
```

Procedures of reading the waveform data from the internal memory:

```
:STOP /*Sets the instrument to STOP state (you can only read the
waveform data from the internal memory when the oscilloscope is in
STOP state).*/
:WAV:SOUR CHAN1 /*Sets the channel source to CHANnel1.*/
:WAV:MODE RAW /*Sets the waveform reading mode to RAW.*/
:WAV:FORM BYTE /*Sets the return format of the waveform data to
BYTE.*/
:WAV:STAR 1 /*Sets the start point of waveform data reading to
the first waveform point.*/
:WAVeform:STOP 120000 /*Sets the stop point of waveform data
reading to the 120000th waveform point (last point).*/
```

Return Format

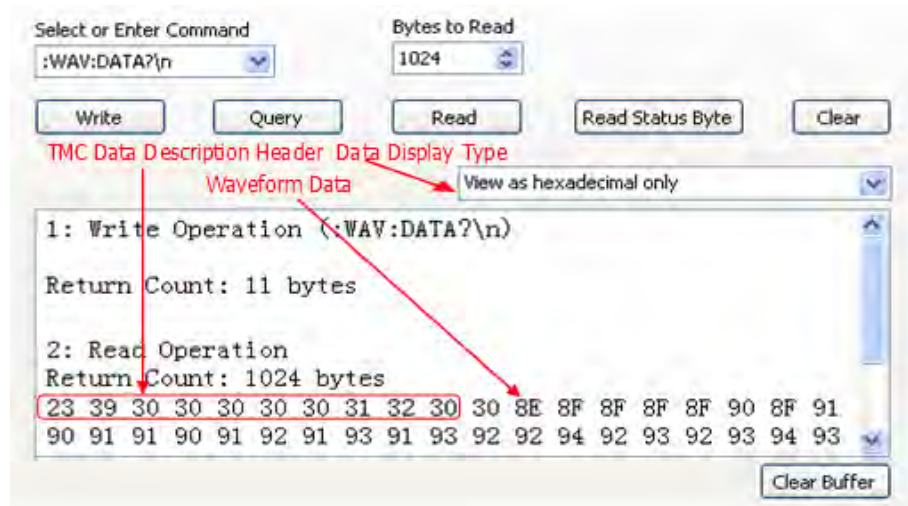
- The return format is related to the return format of the currently selected waveform data (*:WAVeform:FORMat*).

WORD or BYTE format: The read data format is TMC header + waveform data points + end identifier. The TMC header is in #NXXXXXX format; wherein, # is the TMC header identifier; N following # represents the length of the waveform data; the length of the waveform data points is expressed in ASCII strings, and the terminator represents the ending of communication. For example, the data read for one time is #9000001000. It indicates that 9 bytes are used to describe the data length. 000001000 indicates the length of waveform data, i.g. 1,000 bytes.

AScii format: The read data format is waveform data points + end identifier. The waveform data point query returns the actual voltage value of each waveform point in scientific notation; and the voltage values are separated by commas.

- When the waveform data in the internal memory are read in batches, the waveform data returned each time might be the data in one area of the internal memory. In "WORD" or "BYTE" return format, each returned data in blocks contain the TMC data block header. Waveform data in two adjacent data blocks are consecutive.
- The figure below shows the waveform data that have been read (in BYTE format). First, select "View as hexadecimal only" from the drop-down list at the right side. Then, the waveform data that have been read are displayed in hexadecimal format. The first 11 bytes denote the "TMC data block header", and beginning

from the 12th byte (8E) are the waveform data. You can convert the waveform data read to the voltage value of each point of the waveform by using the formula " $(0x8E - YORigin - YREference) \times YINCrement$ ". For the definitions of the parameters in this formula, refer to *Related Command*.



Related Command

:WAVeform:MODE

:WAVeform:YINCrement?

:WAVeform:YINCrement?

:WAVeform:YORigin?

3.29.6 :WAVeform:XINCrement?

Syntax

`:WAVeForm:XINCrement?`

Description

Queries the time interval between two neighboring points of the currently selected channel source in the X direction.

Parameter

N/A

Remarks

The returned value is related to the current data reading mode:

- In NORMal mode, XINCrement = TimeScale/100.
- In RAW mode, XINCrement = 1/SampleRate.
- In MAX mode, XINCrement = TimeScale/100 when the oscilloscope is in the Run state; XINCrement = 1/SampleRate when the oscilloscope is in the Stop state.

The unit is related to the current channel source:

Return Format

The query returns the time difference in scientific notation.

Example

N/A

3.29.7 :WAVeform:XORigin?

Syntax

:WAVeform:XORigin?

Description

Queries the start time of the waveform data of the currently selected channel source in the X direction.

Parameter

N/A

Remarks

The returned value is related to the current data reading mode:

- In NORMal mode, the query returns the start time of the waveform data displayed on the screen.
- In RAW mode, the query returns the start time of the waveform data in the internal memory.
- In MAX mode, the query returns the start time of the waveform data displayed on the screen when the instrument is in the RUN state; the query returns the start time of the waveform data in the internal memory when the instrument is in the Stop state.

The unit is related to the current channel source.

Return Format

The query returns the time value in scientific notation.

Example

N/A

3.29.8 :WAVeform:XREFerence?

Syntax

:WAVeform:XREFerence?

Description

Queries the reference time of the waveform points of the currently selected channel source in the X direction.

Parameter

N/A

Remarks

N/A

Return Format

The query returns 0 (namely the first waveform point on the screen or in the internal memory).

Example

N/A

3.29.9 :WAVeform:YINCrement?

Syntax

:WAVeform:YINCrement?

Description

Queries the unit voltage value of the current source channel Y in the Y direction.

Parameter

N/A

Remarks

The returned value is related to the current data reading mode:

- In NORMal mode, YINCrement = VerticalScale/25.
- In RAW mode, YINCrement and Verticalscale of the memory waveforms are related to the currently selected Verticalscale.
- In MAX mode, YINCrement = VerticalScale/25 when the instrument is in the RUN state; YINCrement is related to the VerticalScale of the internal waveform and the currently selected VerticalScale when the instrument is in the Stop state.

Return Format

The query returns the unit voltage value in scientific notation.

Example

N/A

3.29.10 :WAVeform:YORigin?

Syntax

:WAVeForm:YORigin?

Description

Queries the vertical offset relative to the vertical reference position of the currently selected channel source in the Y direction.

Parameter

N/A

Remarks

The returned value is related to the current data reading mode:

- In NORMal mode, YORigin = VerticalOffset/YINCrement.
- In RAW mode, YORigin is related to the VerticalScale of the memory waveforms and the currently selected VerticalScale.
- In MAX mode, YORigin = VerticalOffset/YINCrement when the instrument is in the RUN state; YORigin is related to the VerticalScale of the internal waveform and the currently selected VerticalScale when the instrument is in the Stop state.

Return Format

The query returns an integer.

Example

N/A

3.29.11 :WAVeform:YREFerence?

Syntax

:WAVeForm:YREFerence?

Description

Queries the vertical reference position of the currently selected channel source in the Y direction.

Parameter

N/A

Remarks

The value of YREference is related to the configuration of the `:WAVeform:FORMat` command. The reference position is different for different return formats of waveform data.

Return Format

The query returns an integer.

Example

N/A

3.29.12 :WAVeform:START**Syntax**

```
:WAVeform:START <sta>
```

```
:WAVeform:START?
```

Description

Sets or queries the start position of waveform data reading.

Parameter

| Name | Type | Range | Default |
|-------|---------|-------------------------|---------|
| <sta> | Integer | Refer to <i>Remarks</i> | 1 |

Remarks

When reading the waveform data from the internal memory, the actual settable ranges of the start point and stop point of a reading operation are related to the memory depth of the oscilloscope and the return format of the waveform data currently selected.

- In Normal mode, the range is from 1 to 1,000.
- In Max mode, when the oscilloscope is in RUN state, its range is from 1 to 1,000; when the oscilloscope is in STOP state, its range is from 1 to current maximum memory depth.
- In Raw mode, the range is from 1 to the current maximum memory depth.

Return Format

The query returns an integer.

Example

```
:WAVeform:START 100 /*Sets the start point to 100.*/
:WAVeform:START? /*The query returns 100.*/
```

3.29.13 :WAVeform:STOP

Syntax

```
:WAVeform:STOP <stop>
```

```
:WAVeform:STOP?
```

Description

Sets or queries the stop position of waveform data reading.

Parameter

| Name | Type | Range | Default |
|--------|---------|-------------------------|---------|
| <stop> | Integer | Refer to <i>Remarks</i> | 1,000 |

Remarks

When reading the waveform data in the internal memory, the actual settable ranges of the start point and stop point of a reading operation are related to the memory depth of the oscilloscope and the return format of the waveform data currently selected.

- In Normal mode, the range is from 1 to 1,000.
- In Max mode, when the oscilloscope is in RUN state, its range is from 1 to 1,000; when the oscilloscope is in STOP state, its range is from 1 to current maximum memory depth.
- In Raw mode, the range is from 1 to the current maximum memory depth.

Return Format

The query returns an integer.

Example

```
:WAVeform:STOP 500 /*Sets the stop point to 500.*/
:WAVeform:STOP? /*The query returns 500.*/
```

3.29.14 :WAVeform:PREamble?

Syntax

```
:WAVeform:PREamble?
```

Description

Queries all the waveform parameters.

Parameter

N/A

Remarks

N/A

Return Format

The query returns 10 waveform parameters, separated by commas.

<format>,<type>,<points>,<count>,<xincrement>,<xorigin>,<xreference>,<yincrement>,<yorigin>,<yreference>

Wherein,

<**format**>: indicates 0 (BYTE), 1 (WORD), or 2 (ASC).

<**type**>: indicates 0 (NORMal), 1 (MAXimum), or 2 (RAW).

<**points**>: an integer ranging from 1 to 50,000,000.

<**count**>: indicates the number of averages in the average sample mode. The value of <count> parameter is 1 in other modes.

<**xincrement**>: indicates the time difference between two neighboring points in the X direction.

<**xorigin**>: indicates the start time of the waveform data in the X direction.

<**xreference**>: indicates the reference time of the waveform data in the X direction.

<**yincrement**>: indicates the step value of the waveforms in the Y direction.

<**yorigin**>: indicates the vertical offset relative to the "Vertical Reference Position" in the Y direction.

<**yreference**>: indicates the vertical reference position in the Y direction.

Example

```
:WAVEform:PREamble?/*The query returns  
0,0,1000,1,1.000000E-8,-5.000000E-6,0.000000E-12,4.000000E-03,0,128.  
*/
```

4 Programming Examples

This chapter illustrates how to control the instrument by programming in LabVIEW, Visual Basic, and Visual C++. These examples are programmed based on Virtual Instrument Software Architecture (VISA) library.

4.1 Programming Preparations

Before programming, you need to prepare the following tasks:

You can log in to download the software. Then install the software according to the installation wizard. After Ultra Sigma is installed successfully, NI-VISA library will be completely installed automatically. In this manual, the default installation path is C:\Program Files\IVI Foundation\VISA.

In the manual, the instrument communicates with the PC via the USB interface. Connect the USB Device interface on the rear panel of the instrument to the PC by using the USB cable. After the instrument is properly connected to the PC, power on the instrument to start it.

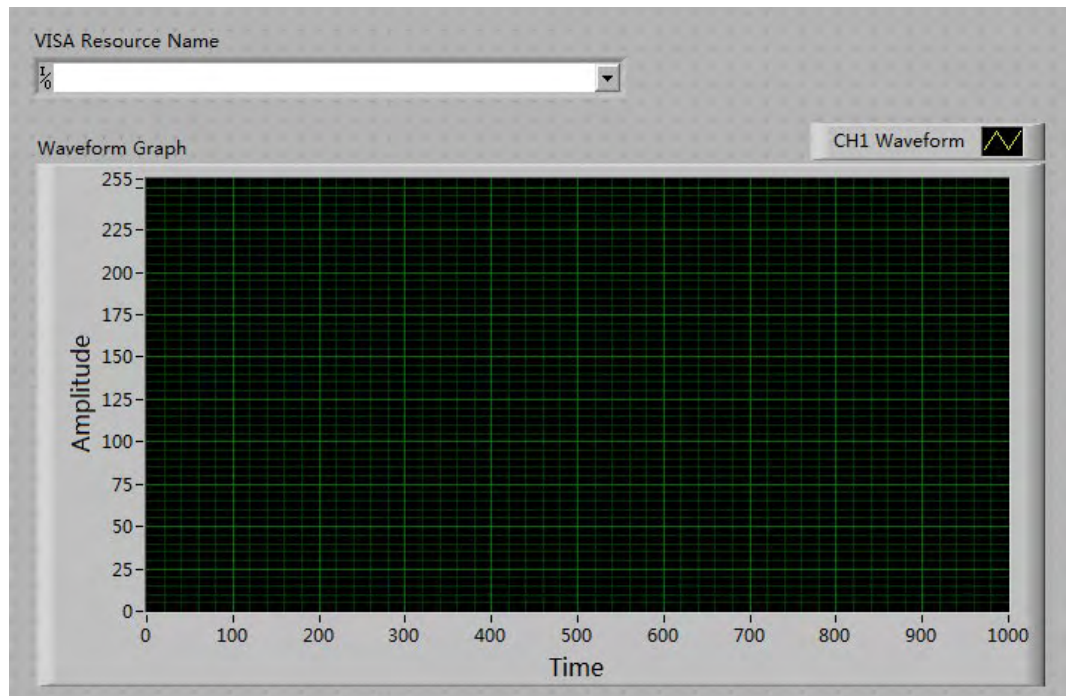
The following parts will make a detailed introduction about the programming examples in LabVIEW, Visual Basic, and Visual C++.

4.2 LabVIEW Programming Example

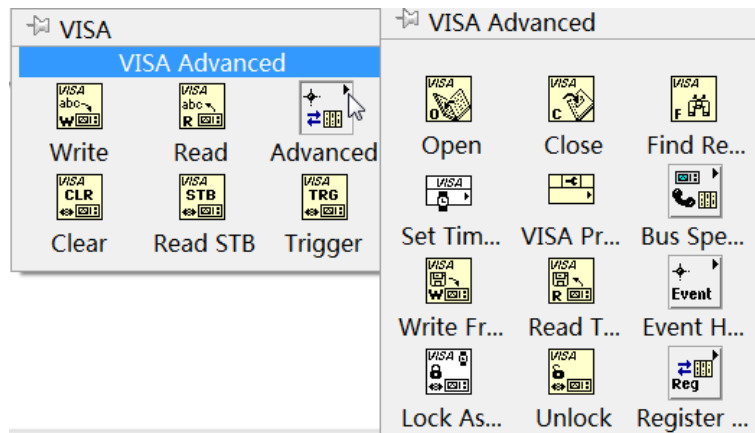
Program used in this example: LabVIEW2010

Function realized in this example: reading the waveform data of CH1 on the screen.

1. Run LabVIEW, and then create a VI file named LabVIEW_Demo.
2. Add controls and create the front panel as shown in the figure below.



3. Open the Block Diagram panel. Click **Instrument I/O > VISA**. Add the following functions: VISA Open, VISA Read, VISA Write, and VISA Close.

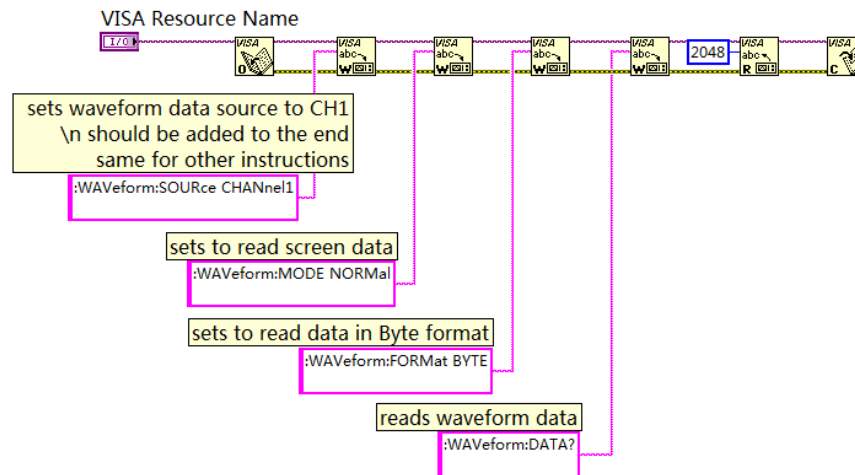


4. Connect the VISA resource name with the VISA Open. Then, connect the VISA resource name outputs of all the functions with the VISA resource name and connect the error output with the error input, as shown in the figure below.

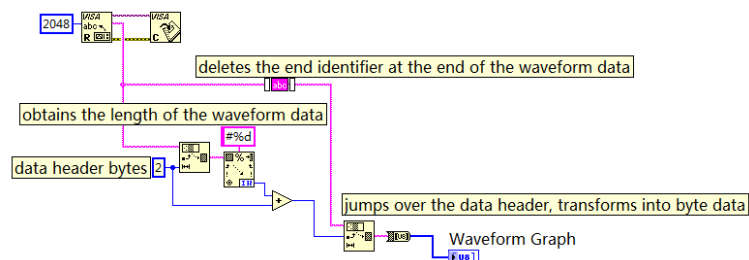


5. Add string constant in the write buffer areas of the VISA Write function and input the following instructions in the figure below. Waveform data is read through the

VISA Read function which requires users to input the total number of bytes to be read. In this example, the total number of bytes of waveform data to be read is less than 2048. Use the VISA Close function to close the VISA resource after the VISA operation is finished.

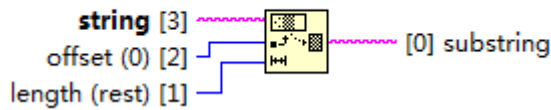


6. The data format is TMC data block header + waveform data + terminator. The TMC header is in #NXXXXXX format; wherein, # is the TMC header identifier; N following # represents the length of the waveform data; the length of the waveform data points is expressed in ASCII strings, and the terminator represents the ending of communication. For example, the data read for one time is #9000001000XXXX. It indicates that 9 bytes are used to describe the data length. 000001000 indicates the length of waveform data, i.g. 1,000 bytes. Use the following block diagram to obtain the number of bytes that the TMC header occupies. Ignore the TMC header and delete the terminator at the end of the waveform data, and transfer the waveform data to the byte data and display it on the waveform diagram controls.



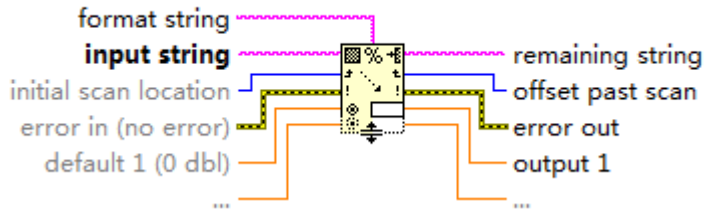
The available functions used in the above block diagram as as follows:

String Subset



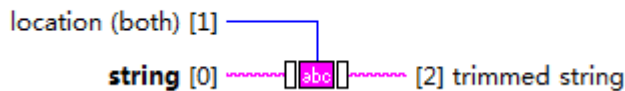
Used to obtain the TMC header "#N". After obtaining the number of bytes that the TMC header occupies, ignore the data header to obtain the waveform data strings

Scan From String



Used to obtain the waveform data length bytes

Trim Whitespace.vi (4803)



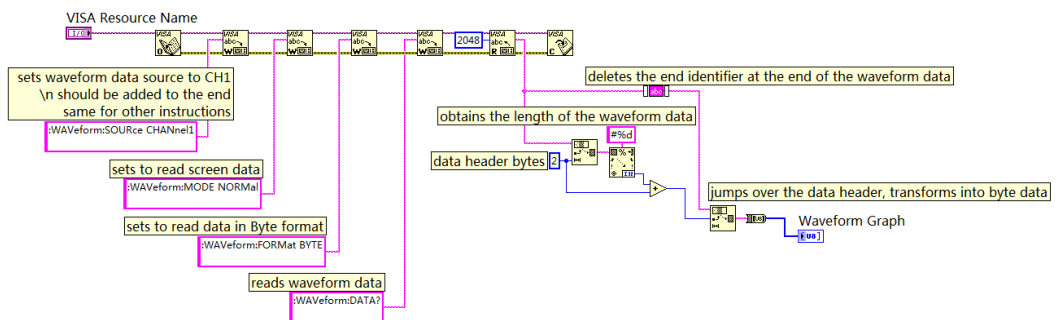
Used to delete the terminator at the end of the waveform data

String To Byte Array

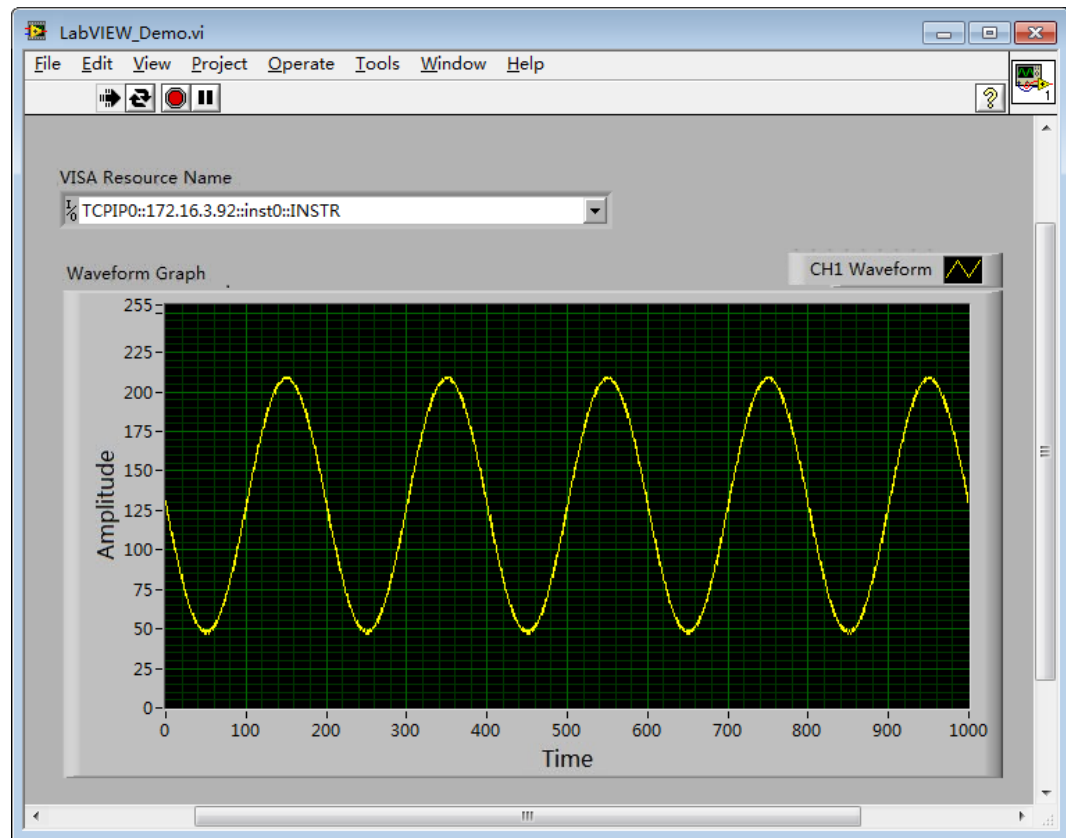


Used to transfer waveform data strings to the byte group

7. The complete program block diagram is as shown in the figure below:



8. Select the device resource from the VISA Resource Name drop-down list and run the program.



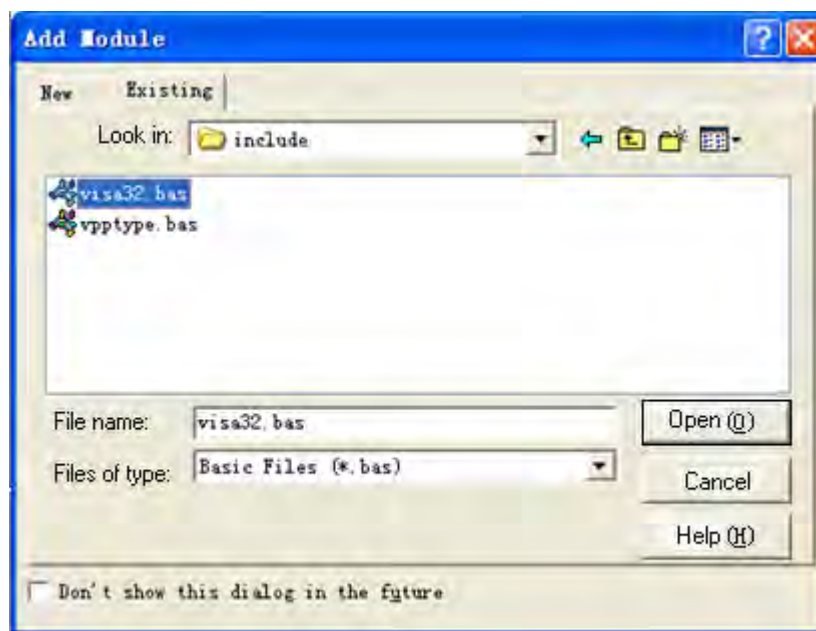
4.3 Visual Basic Programming Example

Program used in this example: Visual Basic 6.0

Function realized in this example: control the on/off state of any channel.

Enter the Visual Basic 6.0 programming environment, and perform the following procedures.

1. Build a standard application program project (Standard EXE), and name it "Demo".
2. Click **Project > Add Module** to open the Add Module dialog box. In the dialog box, click the Existing tab to search for the **visa32.bas** file in the include folder under the NI-VISA installation path and add the file.



- In the Demo dialog box, add four buttons to represent CH1 to CH4 respectively. Add four Labels (Label1(0), Label1(1), Label1(2), Label1(3)) to represent the statuses of CH1 to CH4 respectively (when the channel is enabled, it displays the color of the channel; when the channel is disabled, it displays gray), as shown in the figure below.



- Click **Project > Project1 Properties** to open the Project1 – Project Properties dialog box. In the dialog box, click on the General tab and select **Form1** from the drop-down list under Startup Object.
- Double-click CH1 to enter the programming environment. Add the following codes to control CH1-CH4. The codes of CH1 are as shown below; the codes of the other channels are similar.

```
Dim defrm As Long
Dim vi As Long
Dim strRes As String * 200
Dim list As Long
Dim nmatches As Long
Dim matches As String * 200 'Reserve the obtained device number
Dim s32Disp As Integer
' Obtain the usb resource of visa
Call viOpenDefaultRM(defrm)
Call viFindRsrc(defrm, "USB?*\"", list, nmatches, matches)
' Turn on the instrument
Call viOpen(defrm, matches, 0, 0, vi)
```

```

' Send a command to query the status of CH1
Call viVPrintf(vi, ":CHAN1:DISP?" + Chr$(10), 0)
' Obtain the status of CH1
Call viVScanf(vi, "%t", strRes)
s32Disp = CInt(strRes)
If (s32Disp = 1) Then
' Send the setting command
Call viVPrintf(vi, ":CHAN1:DISP 0" + Chr$(10), 0)
Label1(0).ForeColor = &H808080 'Gray
Else
Call viVPrintf(vi, ":CHAN1:DISP 1" + Chr$(10), 0)
Label1(0).ForeColor = &HFFFF& 'Yellow
End If
' Close the resource
Call viClose(vi)
Call viClose(defrm)

```

6. Save and run the project to obtain a single exe program for demo. When the instrument is correctly connected to the PC, you can control the on/off status of any channel.

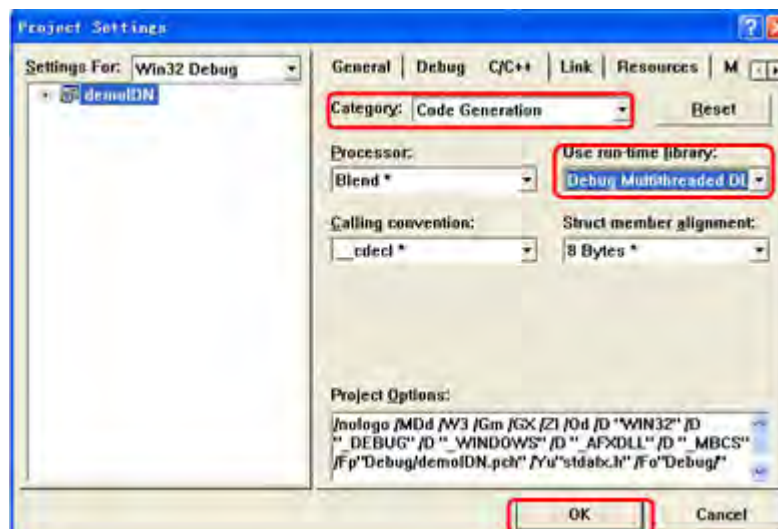
4.4 Visual C++ Programming Example

Program used in this example: Visual C++ 6.0

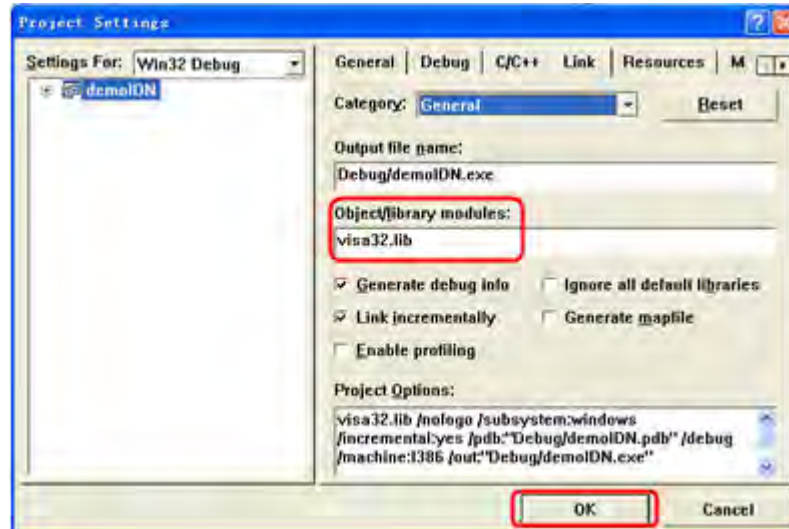
Function realized in this example: search for the instrument address, connect to the instrument, send commands, and read return values.

Enter the Visual C++ 6.0 programming environment, and perform the following procedures.

1. Create a MFC project based on a dialog box.
2. Click **Project > Settings** to open the **Project Setting** dialog box. In the dialog box, click the **C/C++** tab, select **Code Generation** from the drop-down list under **Category**. Choose **Debug Multithreaded DLL** from the drop-down list under **Use run-time library**. Click **OK** to close the dialog box.



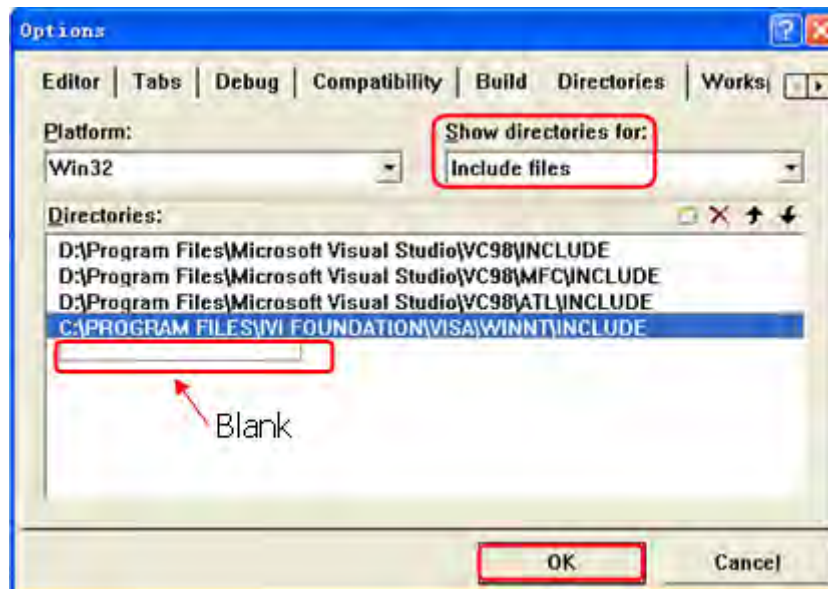
- Click **Project > Settings** to open the **Project Setting** dialog box. In the dialog box, click the **Link** tab, add "visa32.lib" under **Object/library modules**, then click **OK** to close the dialog box.



- Click **Tools > Options** to open the Options dialog box. Then click the **Directories** tab.

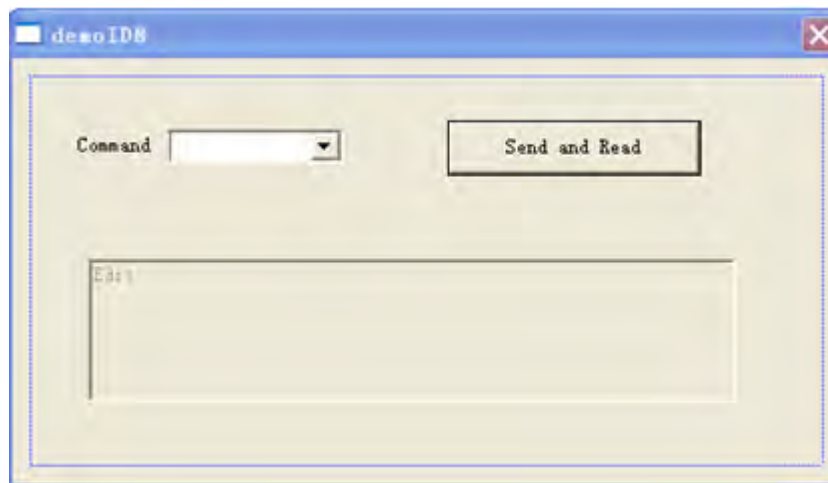
Select **Include files** from the drop-down list under **Show directories for**. Double click the empty space under **Directories** to enter the specified path of Include files: C:\Program Files\IVI Foundation\VISA\WinNT\include. Click **OK** to close the dialog box.

Select **Library files** from the drop-down list under **Show directories for**. Double click the empty space under **Directories** to enter the specified path of Library files: C:\Program Files\IVI Foundation\VISA\WinNT\lib\msc. Click **OK** to close the dialog box.



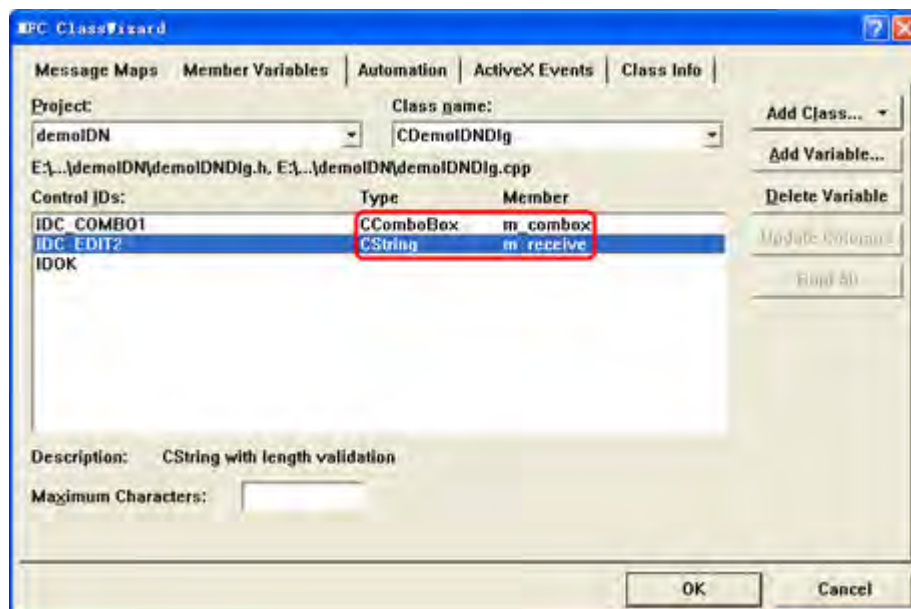
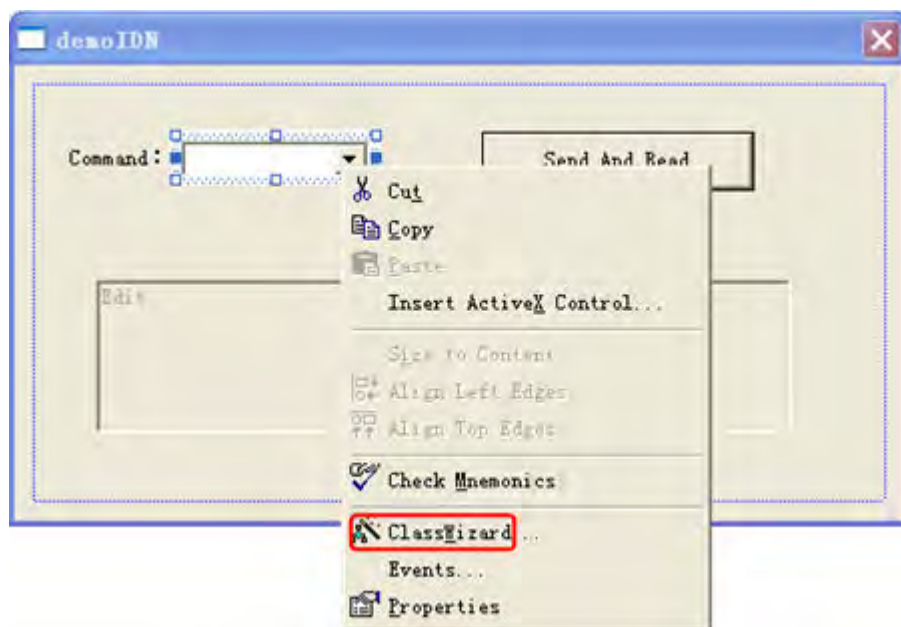
Note: By now, VISA library has been added.

5. Add the **Text**, **Combo Box**, **Button**, and **Edit Box** controls. The layout interface for adding controls is as follows:



6. Modify the control attributes.
 - a. Name **Text** as "Command".
 - b. Open the **Data** item in the **Combo Box** attribute and input the following command *IDN? manually.
 - c. Open the **General** item in the **Edit Box** attribute and select **Disabled**.
 - d. Name **Button** as **Send and Read**.

7. Add the variables `m_combox` and `m_receive` to the **Com Box** and **Edit Box** controls respectively.



8. Add codes.

Double-click **Send and Read** to enter the programming environment. Declare the `#include <visa.h>` of the VISA library in the header file and then add the following codes:

```
ViSession defaultRM, vi;
char buf [256] = {0};
CString s, strTemp;
char* stringTemp;

ViChar buffer [VI_FIND_BUFLLEN];
```

```
ViRsrc matches="buffer";
ViUInt32 nmatches;
ViFindList list;

viOpenDefaultRM (&defaultRM);
//Acquire the USB resource of VISA
viFindRsrc(defaultRM, "USB?*",&list,&nmatches, matches);
viOpen (defaultRM,matches,VI_NULL,VI_NULL,&vi);

//Send the command received
m_combox.GetLBText(m_combox.GetCurSel(),strTemp);
strTemp = strTemp + "\n";
stringTemp = (char *) (LPCTSTR)strTemp;
viPrintf (vi,stringTemp);

//Read the results
viScanf (vi, "%t\n", &buf);

//Display the results
UpdateData (TRUE);
m_receive = buf;
UpdateData (FALSE);
viClose (vi);
viClose (defaultRM);
```

9. Save, compile, and run the project to obtain a single exe file. When the instrument is correctly connected to the PC, enter a command (for example, *IDN?) and click **Send and Read** to execute the command. Then, the reading results will be returned.

