

Magnova® SCPI Manual

December 19, 2024

Document Version 0.1

Note: This is a preliminary version and may be updated periodically.

Contents

1 Overview	16
1.1 Syntax	16
1.1.1 Commands and Queries	16
1.1.2 Parameters and Return Values	17
1.1.3 Block Data	17
1.1.4 Abbreviation	18
1.1.5 Notes	18
1.2 Interfaces	18
1.2.1 Common	18
1.2.2 RESTful API	18
2 Examples	21
2.1 Simple instrument commands and queries using USBTMC	21
2.2 Waveform Plot using raw sockets	22
2.3 Waveform Plot using the REST API	23
2.4 FFT Plot using raw sockets	24
2.5 Measurement using raw sockets	25
3 Commands and Queries	26
3.1 Common	26
*CLS	26
*ESE	26
*ESE?	26
*ESR?	26
*IDN?	26
*OPC	26
*OPC?	26
*RST	26
*SRE	26
*SRE?	26
*STB	26
*STB?	26
*TST?	26
*WAI	26
3.2 Control	27
:RUN	27
:STOP	27
:SINGle	27
:FORCe	27
:AUTO	27
:AUTO?	27
:CLear	27
:SEquence:STOP	27
:SEquence:WAIT?	27
3.3 :ACquire	28
:ACquire:MDEPth	28
:ACquire:MDEPth?	28
:ACquire:TEXPansion	28
:ACquire:TEXPansion?	28
:ACquire:ROLLmode	28
:ACquire:ROLLmode?	29
:ACquire:HRESolution	29
:ACquire:HRESolution?	29
:ACquire:RCLock	29
:ACquire:RCLock?	30
:ACquire:RCOut	30
:ACquire:RCOut?	30

	:ACQuire:MODE	30
	:ACQuire:MODE?	30
	:ACQuire:AVERAge	30
	:ACQuire:AVERAge?	30
	:ACQuire:SRATE?	31
3.4	:TIMebase	32
	:TIMebase:REFerence	32
	:TIMebase:REFerence?	32
	:TIMebase:OFFSet	32
	:TIMebase:OFFSet?	32
	:TIMebase:SCALe	32
	:TIMebase:SCALe?	32
3.5	:CHANnel	33
	:CHANnel:STATe	33
	:CHANnel:STATe?	33
	:CHANnel[N]:STATe	33
	:CHANnel[N]:STATe?	33
	:CHANnel[N]:TERMination	33
	:CHANnel[N]:TERMination?	33
	:CHANnel[N]:COUPling	34
	:CHANnel[N]:COUPling?	34
	:CHANnel[N]:DESKew	34
	:CHANnel[N]:DESKew?	34
	:CHANnel[N]:DIVisor	34
	:CHANnel[N]:DIVisor?	34
	:CHANnel[N]:UNIT	34
	:CHANnel[N]:UNIT?	35
	:CHANnel[N]:OFFSet	35
	:CHANnel[N]:OFFSet?	35
	:CHANnel[N]:SCALe	35
	:CHANnel[N]:SCALe?	35
3.6	:CHANnel[N]:DATA	36
	:CHANnel[N]:DATA:SOURce	36
	:CHANnel[N]:DATA:SOURce?	36
	:CHANnel[N]:DATA:TYPE	36
	:CHANnel[N]:DATA:TYPE?	36
	:CHANnel[N]:DATA:RECORD	36
	:CHANnel[N]:DATA:RECORD?	36
	:CHANnel[N]:DATA:TDELta?	36
	:CHANnel[N]:DATA:STIME?	37
	:CHANnel[N]:DATA:ETIME?	37
	:CHANnel[N]:DATA:SSTart?	37
	:CHANnel[N]:DATA:SLENgth?	37
	:CHANnel[N]:DATA:VSTart?	37
	:CHANnel[N]:DATA:VLENgth?	37
	:CHANnel[N]:DATA:SAMPles?	37
	:CHANnel[N]:DATA:PACKed?	38
	3.6.1 Analog Data Samples	38
	3.6.2 Analog Data Packed	39
3.7	:DIGital	40
	:DIGital:STATe	40
	:DIGital:RST	40
	:DIGital:STATe?	40
	:DIGital[N]:STATe	40
	:DIGital[N]:STATe?	40
	:DIGital:THReshoLd[0to7/8to15]	40
	:DIGital:THReshoLd[0to7/8to15]?	40
3.8	:TRIGger	41

:TRIGger:TYPE	41
:TRIGger:TYPE?	41
:TRIGger:AOUT	42
:TRIGger:AOUT?	42
3.9 :TRIGger:EDGE	43
:TRIGger:EDGE:SOURce	43
:TRIGger:EDGE:SOURce?	43
:TRIGger:EDGE:SLOPe	43
:TRIGger:EDGE:SLOPe?	43
:TRIGger:EDGE:LEVeL	43
:TRIGger:EDGE:LEVeL?	43
:TRIGger:EDGE:LHYSteresis	44
:TRIGger:EDGE:LHYSteresis?	44
3.10 :TRIGger:NEDGE	45
:TRIGger:NEDGE:SOURce	45
:TRIGger:NEDGE:SOURce?	45
:TRIGger:NEDGE:SLOPe	45
:TRIGger:NEDGE:SLOPe?	45
:TRIGger:NEDGE:LEVeL	45
:TRIGger:NEDGE:LEVeL?	45
:TRIGger:NEDGE:IDLe	45
:TRIGger:NEDGE:IDLe?	46
:TRIGger:NEDGE:COUNt	46
:TRIGger:NEDGE:COUNt?	46
:TRIGger:NEDGE:LHYSteresis	46
:TRIGger:NEDGE:LHYSteresis?	46
3.11 :TRIGger:TIMEout	47
:TRIGger:TIMEout:SOURce	47
:TRIGger:TIMEout:SOURce?	47
:TRIGger:TIMEoutGe:SLOPe	47
:TRIGger:TIMEout:SLOPe?	47
:TRIGger:TIMEout:LEVeL	47
:TRIGger:TIMEout:LEVeL?	47
:TRIGger:TIMEout:TIME	47
:TRIGger:TIMEout:TIME?	48
:TRIGger:TIMEout:LHYSteresis	48
:TRIGger:TIMEout:LHYSteresis?	48
3.12 :TRIGger:DELaY	49
:TRIGger:DELaY:SOURce1	49
:TRIGger:DELaY:SOURce1?	49
:TRIGger:DELaY:SLOPe1	49
:TRIGger:DELaY:SLOPe1?	49
:TRIGger:DELaY:LEVeL1	49
:TRIGger:DELaY:LEVeL1?	49
:TRIGger:DELaY:SOURce2	49
:TRIGger:DELaY:SOURce2?	50
:TRIGger:DELaY:SLOPe2	50
:TRIGger:DELaY:SLOPe2?	50
:TRIGger:DELaY:LEVeL2	50
:TRIGger:DELaY:LEVeL2?	50
:TRIGger:DELaY:TYPE	50
:TRIGger:DELaY:TYPE?	51
:TRIGger:DELaY:THReshoLd	51
:TRIGger:DELaY:THReshoLd?	51
:TRIGger:DELaY:DURation:LOWer	51
:TRIGger:DELaY:DURation:LOWer?	51
:TRIGger:DELaY:DURation:UPPer	51
:TRIGger:DELaY:DURation:UPPer?	51

:TRIGger:DElay:LHYSteresis	52
:TRIGger:DElay:LHYSteresis?	52
3.13 :TRIGger:WINDow	53
:TRIGger:WINDow:SOURce	53
:TRIGger:WINDow:SOURce?	53
:TRIGger:WINDow:TYPe	53
:TRIGger:WINDow:TYPe?	53
:TRIGger:WINDow:LEVe11	53
:TRIGger:WINDow:LEVe11?	53
:TRIGger:WINDow:LEVe12	53
:TRIGger:WINDow:LEVe12?	54
:TRIGger:WINDow:TIMing	54
:TRIGger:WINDow:TIMing?	54
:TRIGger:WINDow:THReSho1d	54
:TRIGger:WINDow:THReSho1d?	54
:TRIGger:WINDow:DURation:LOWer	54
:TRIGger:WINDow:DURation:LOWer?	54
:TRIGger:WINDow:DURation:UPPer	55
:TRIGger:WINDow:DURation:UPPer?	55
:TRIGger:WINDow:LHYSteresis	55
:TRIGger:WINDow:LHYSteresis?	55
3.14 :TRIGger:PULSe	56
:TRIGger:PULSe:SOURce	56
:TRIGger:PULSe:SOURce?	56
:TRIGger:PULSe:POLarity	56
:TRIGger:PULSe:POLarity?	56
:TRIGger:PULSe:LEVe1	56
:TRIGger:PULSe:LEVe1?	56
:TRIGger:PULSe:TIMing	57
:TRIGger:PULSe:TIMing?	57
:TRIGger:PULSe:THReSho1d	57
:TRIGger:PULSe:THReSho1d?	57
:TRIGger:PULSe:DURation:LOWer	57
:TRIGger:PULSe:DURation:LOWer?	57
:TRIGger:PULSe:DURation:UPPer	57
:TRIGger:PULSe:DURation:UPPer?	58
:TRIGger:PULSe:LHYSteresis	58
:TRIGger:PULSe:LHYSteresis?	58
3.15 :TRIGger:SLOPe	59
:TRIGger:SLOPe:SOURce	59
:TRIGger:SLOPe:SOURce?	59
:TRIGger:SLOPe:TYPe	59
:TRIGger:SLOPe:TYPe?	59
:TRIGger:SLOPe:LEVe11	59
:TRIGger:SLOPe:LEVe11?	59
:TRIGger:SLOPe:LEVe12	59
:TRIGger:SLOPe:LEVe12?	60
:TRIGger:SLOPe:TIMing	60
:TRIGger:SLOPe:TIMing?	60
:TRIGger:SLOPe:THReSho1d	60
:TRIGger:SLOPe:THReSho1d?	60
:TRIGger:SLOPe:DURation:LOWer	60
:TRIGger:SLOPe:DURation:LOWer?	60
:TRIGger:SLOPe:DURation:UPPer	61
:TRIGger:SLOPe:DURation:UPPer?	61
:TRIGger:SLOPe:LHYSteresis	61
:TRIGger:SLOPe:LHYSteresis?	61
3.16 :TRIGger:INTerval	62

:TRIGger:INTerval:SOURce	62
:TRIGger:INTerval:SOURce?	62
:TRIGger:INTerval:POLarity	62
:TRIGger:INTerval:POLarity?	62
:TRIGger:INTerval:LEVel	62
:TRIGger:INTerval:LEVel?	62
:TRIGger:INTerval:TIMing	63
:TRIGger:INTerval:TIMing?	63
:TRIGger:INTerval:THReshold	63
:TRIGger:INTerval:THReshold?	63
:TRIGger:INTerval:DURation:LOWer	63
:TRIGger:INTerval:DURation:LOWer?	63
:TRIGger:INTerval:DURation:UPPer	63
:TRIGger:INTerval:DURation:UPPer?	64
:TRIGger:INTerval:LHYSteresis	64
:TRIGger:INTerval:LHYSteresis?	64
3.17 :TRIGger:SHOLd	65
:TRIGger:SHOLd:CLOCK:SOURce	65
:TRIGger:SHOLd:CLOCK:SOURce?	65
:TRIGger:SHOLd:CLOCK:SLOPe	65
:TRIGger:SHOLd:CLOCK:SLOPe?	65
:TRIGger:SHOLd:CLOCK:LEVel	65
:TRIGger:SHOLd:CLOCK:LEVel?	65
:TRIGger:SHOLd:CLOCK:HYSteresis	65
:TRIGger:SHOLd:CLOCK:HYSteresis?	66
:TRIGger:SHOLd:DATA:SOURce	66
:TRIGger:SHOLd:DATA:SOURce?	66
:TRIGger:SHOLd:DATA:LEVel	66
:TRIGger:SHOLd:DATA:LEVel?	66
:TRIGger:SHOLd:DATA:HYSteresis	66
:TRIGger:SHOLd:DATA:HYSteresis?	66
:TRIGger:SHOLd:TYPe	66
:TRIGger:SHOLd:TYPe?	67
:TRIGger:SHOLd:STIME	67
:TRIGger:SHOLd:STIME?	67
:TRIGger:SHOLd:HTIME	67
:TRIGger:SHOLd:HTIME?	67
3.18 :TRIGger:RUNT	68
:TRIGger:RUNT:SOURce	68
:TRIGger:RUNT:SOURce?	68
:TRIGger:RUNT:POLarity	68
:TRIGger:RUNT:POLarity?	68
:TRIGger:RUNT:LEVel1	68
:TRIGger:RUNT:LEVel1?	68
:TRIGger:RUNT:LEVel2	68
:TRIGger:RUNT:LEVel2?	69
:TRIGger:RUNT:TIMing	69
:TRIGger:RUNT:TIMing?	69
:TRIGger:RUNT:THReshold	69
:TRIGger:RUNT:THReshold?	69
:TRIGger:RUNT:DURation:LOWer	69
:TRIGger:RUNT:DURation:LOWer?	69
:TRIGger:RUNT:DURation:UPPer	70
:TRIGger:RUNT:DURation:UPPer?	70
:TRIGger:RUNT:LHYSteresis	70
:TRIGger:RUNT:LHYSteresis?	70
3.19 :TRIGger:PATTern	71
:TRIGger:PATTern:TYPe	71

:TRIGger:PATtern:TYPE?	71
:TRIGger:PATtern:TIMing	71
:TRIGger:PATtern:TIMing?	71
:TRIGger:PATtern:THReshold	72
:TRIGger:PATtern:THReshold?	72
:TRIGger:PATtern:DURation:LOWer	72
:TRIGger:PATtern:DURation:LOWer?	72
:TRIGger:PATtern:DURation:UPPer	72
:TRIGger:PATtern:DURation:UPPer?	72
:TRIGger:PATtern:[Channel]:CLOCK	72
:TRIGger:PATtern:[Channel]:CLOCK?	72
:TRIGger:PATtern:[Channel]:PATtern	73
:TRIGger:PATtern:[Channel]:PATtern?	73
:TRIGger:PATtern:[Channel]:THReshold:LOWer	73
:TRIGger:PATtern:[Channel]:THReshold:LOWer?	73
:TRIGger:PATtern:[Channel]:THReshold:UPPer	73
:TRIGger:PATtern:[Channel]:THReshold:UPPer?	73
:TRIGger:PATtern:CLOCK:SOURce	73
:TRIGger:PATtern:CLOCK:SOURce?	74
:TRIGger:PATtern:CLOCK:SLOPe	74
:TRIGger:PATtern:CLOCK:SLOPe?	74
:TRIGger:PATtern:CLOCK:LEVeL	74
:TRIGger:PATtern:CLOCK:LEVeL?	74
:TRIGger:PATtern:CLOCK:HYSteresis	74
:TRIGger:PATtern:CLOCK:HYSteresis?	74
3.20 :TRIGger:DECode	75
:TRIGger:DECode:GRoup	75
:TRIGger:DECode:GRoup?	75
3.21 :TRIGger:DECode:I2C	76
:TRIGger:DECode:I2C:EVEnt	76
:TRIGger:DECode:I2C:EVEnt?	76
:TRIGger:DECode:I2C:ADDRess:NFORmat	76
:TRIGger:DECode:I2C:ADDRess:NFORmat?	77
:TRIGger:DECode:I2C:ADDRess:LENGth	77
:TRIGger:DECode:I2C:ADDRess:LENGth?	77
:TRIGger:DECode:I2C:ADDRess:DIRection	77
:TRIGger:DECode:I2C:ADDRess:DIRection?	77
:TRIGger:DECode:I2C:ADDRess:WORD	77
:TRIGger:DECode:I2C:ADDRess:WORD?	78
:TRIGger:DECode:I2C:DATA:NFORmat	78
:TRIGger:DECode:I2C:DATA:NFORmat?	78
:TRIGger:DECode:I2C:DATA:POSition	78
:TRIGger:DECode:I2C:DATA:POSition?	78
:TRIGger:DECode:I2C:DATA:OFFSet	78
:TRIGger:DECode:I2C:DATA:OFFSet?	79
:TRIGger:DECode:I2C:DATA:LENGth	79
:TRIGger:DECode:I2C:DATA:LENGth?	79
:TRIGger:DECode:I2C:DATA:BYTE[ByteN]	79
:TRIGger:DECode:I2C:DATA:BYTE[ByteN]?	79
3.22 :TRIGger:DECode:SPI	80
:TRIGger:DECode:SPI:LANE	80
:TRIGger:DECode:SPI:LANE?	80
:TRIGger:DECode:SPI:ALIGnment	80
:TRIGger:DECode:SPI:ALIGnment?	80
:TRIGger:DECode:SPI:DATA:POSition	80
:TRIGger:DECode:SPI:DATA:POSition?	80
:TRIGger:DECode:SPI:DATA:NFORmat	81
:TRIGger:DECode:SPI:DATA:NFORmat?	81

:TRIGger:DECode:SPI:DATA:WORDS:OFFSet	81
:TRIGger:DECode:SPI:DATA:WORDS:OFFSet?	81
:TRIGger:DECode:SPI:DATA:WORDS:LENGth	81
:TRIGger:DECode:SPI:DATA:WORDS:LENGth?	81
:TRIGger:DECode:SPI:DATA:WORD[WordN]	81
:TRIGger:DECode:SPI:DATA:WORD[WordN]?	82
:TRIGger:DECode:SPI:DATA:BITS:OFFSet	82
:TRIGger:DECode:SPI:DATA:BITS:OFFSet?	82
:TRIGger:DECode:SPI:DATA:BITS:LENGth	82
:TRIGger:DECode:SPI:DATA:BITS:LENGth?	82
:TRIGger:DECode:SPI:DATA:BITS	82
:TRIGger:DECode:SPI:DATA:BITS?	82
3.23 :TRIGger:DECode:UART	83
:TRIGger:DECode:UART:EVENT	83
:TRIGger:DECode:UART:EVENT?	83
:TRIGger:DECode:UART:DATA:POSition	83
:TRIGger:DECode:UART:DATA:POSition?	83
:TRIGger:DECode:UART:DATA:OFFSet	83
:TRIGger:DECode:UART:DATA:OFFSet?	84
:TRIGger:DECode:UART:DATA:LENGth	84
:TRIGger:DECode:UART:DATA:LENGth?	84
:TRIGger:DECode:UART:DATA:PARity	84
:TRIGger:DECode:UART:DATA:PARity?	84
:TRIGger:DECode:UART:DATA:NFORmat	84
:TRIGger:DECode:UART:DATA:NFORmat?	85
:TRIGger:DECode:UART:DATA:WORD[WordN]	85
:TRIGger:DECode:UART:DATA:WORD[WordN]?	85
3.24 :TRIGger:DECode:PARAllel	86
:TRIGger:DECode:PARAllel:NFORmat	86
:TRIGger:DECode:PARAllel:NFORmat?	86
:TRIGger:DECode:PARAllel:WORD[WordN]	86
:TRIGger:DECode:PARAllel:WORD[WordN]?	86
3.25 :MEASurement	87
:MEASurement:STATE	87
:MEASurement:STATE?	87
:MEASurement:DISPlay	87
:MEASurement:DISPlay?	87
:MEASurement:GATE	87
:MEASurement:GATE?	87
:MEASurement:ATYPE	88
:MEASurement:ATYPE?	88
:MEASurement:AVERage	88
:MEASurement:AVERage?	88
:MEASurement:RLEVeLs:CHANneL[AnaLogN]:BASE	88
:MEASurement:RLEVeLs:CHANneL[AnaLogN]:BASE?	88
:MEASurement:RLEVeLs:CHANneL[AnaLogN]:VPP:LOWer	88
:MEASurement:RLEVeLs:CHANneL[AnaLogN]:VPP:LOWer?	89
:MEASurement:RLEVeLs:CHANneL[AnaLogN]:VPP:MID	89
:MEASurement:RLEVeLs:CHANneL[AnaLogN]:VPP:MID?	89
:MEASurement:RLEVeLs:CHANneL[AnaLogN]:VPP:UPPer	89
:MEASurement:RLEVeLs:CHANneL[AnaLogN]:VPP:UPPer?	89
:MEASurement:RLEVeLs:CHANneL[AnaLogN]:VAMPP:LOWer	89
:MEASurement:RLEVeLs:CHANneL[AnaLogN]:VAMP:LOWer?	89
:MEASurement:RLEVeLs:CHANneL[AnaLogN]:VAMP:MID	89
:MEASurement:RLEVeLs:CHANneL[AnaLogN]:VAMP:MID?	90
:MEASurement:RLEVeLs:CHANneL[AnaLogN]:VAMP:UPPer	90
:MEASurement:RLEVeLs:CHANneL[AnaLogN]:VAMP:UPPer?	90
:MEASurement:RLEVeLs:CHANneL[AnaLogN]:MANuaL:LOWer	90

:MEASurement:RLEVels:CHANnel[AnalogN]:MANual:LOWer?	90
:MEASurement:RLEVels:CHANnel[AnalogN]:MANual:MID	90
:MEASurement:RLEVels:CHANnel[AnalogN]:MANual:MID?	90
:MEASurement:RLEVels:CHANnel[AnalogN]:MANual:UPPer	90
:MEASurement:RLEVels:CHANnel[AnalogN]:MANual:UPPer?	91
:MEASurement:CLEar	91
:MEASurement:[Type]:ALL?	91
:MEASurement:[Type]:CURRent?	91
:MEASurement:[Type]:AVERAge?	91
:MEASurement:[Type]:MAXimum?	91
:MEASurement:[Type]:MINimum?	91
:MEASurement:[Type]:DEVIation?	92
:MEASurement:[Type]:COUNT?	92
:MEASurement:[Type]:ADD	92
:MEASurement:[Type]:REMOve	92
3.25.1 All measurement statistics	92
3.25.2 Measurement Types	92
3.26 :REFerence	94
:REFerence:STATE	94
:REFerence:STATE?	94
:REFerence:CLEar	94
:REFerence[N]:STATE	94
:REFerence[N]:STATE?	94
:REFerence[N]:SOURce	94
:REFerence[N]:SOURce?	94
:REFerence[N]:OFFSet	95
:REFerence[N]:OFFSet?	95
:REFerence[N]:SCALE	95
:REFerence[N]:SCALE?	95
:REFerence[N]:REFresh	95
:REFerence[N]:ADD	95
:REFerence[N]:REMOve	95
3.27 :MATH	96
:MATH:STATE	96
:MATH:STATE?	96
:MATH:CLEar	96
:MATH[N]:STATE	96
:MATH[N]:STATE?	96
:MATH[N]:UNIT	96
:MATH[N]:UNIT?	96
:MATH[N]:EXPRession	97
:MATH[N]:EXPRession?	97
:MATH[N]:OFFSet	97
:MATH[N]:OFFSet?	97
:MATH[N]:SCALE	97
:MATH[N]:SCALE?	97
:MATH[N]:ADD	97
:MATH[N]:REMOve	97
3.28 :FFT	98
:FFT:STATE	98
:FFT:STATE?	98
:FFT:SCALE	98
:FFT:SCALE?	98
:FFT:FREQUency:CENTer	98
:FFT:FREQUency:CENTer?	99
:FFT:FREQUency:SPAN	99
:FFT:FREQUency:SPAN?	99
:FFT:FREQUency:STARt	99

: FFT:FREQuency:START?	99
: FFT:FREQuency:STOP	99
: FFT:FREQuency:STOP?	99
: FFT[N]:STATe	99
: FFT[N]:STATe?	100
: FFT[N]:SOURce	100
: FFT[N]:SOURce?	100
: FFT[N]:WINDow	100
: FFT[N]:WINDow?	100
: FFT[N]:TRACe	101
: FFT[N]:TRACe?	101
: FFT[N]:AVERAge	101
: FFT[N]:AVERAge?	101
: FFT[N]:OFFSet	101
: FFT[N]:OFFSet?	101
: FFT[N]:SCALe	101
: FFT[N]:SCALe?	102
3.29 : FFT[N]:DATA	103
: FFT[N]:DATA:SCALe	103
: FFT[N]:DATA:SCALe?	103
: FFT[N]:DATA:BFRequency?	103
: FFT[N]:DATA:SFRequency?	103
: FFT[N]:DATA:BINS	103
: CHANnel[N]:DATA:PACKed?	104
3.29.1 FFT Data Bins	104
3.29.2 FFT Data Packed	104
3.30 : FFT:PTABle	105
: FFT:PTABle:STATe	105
: FFT:PTABle:STATe?	105
: FFT:PTABle[PeakN]:STATe	105
: FFT:PTABle[PeakN]:STATe?	105
: FFT:PTABle[PeakN]:SOURce	105
: FFT:PTABle[PeakN]:SOURce?	105
: FFT:PTABle[PeakN]:FRANge	106
: FFT:PTABle[PeakN]:FRANge?	106
: FFT:PTABle[PeakN]:TYPE	106
: FFT:PTABle[PeakN]:TYPE?	106
: FFT[N]:PTABle[PeakN]:EXCURsion	106
: FFT[N]:PTABle[PeakN]:EXCURsion?	106
: FFT[N]:PTABle[PeakN]:THReshoLd	106
: FFT[N]:PTABle[PeakN]:THReshoLd?	107
: FFT[N]:PTABle[PeakN]:PEAKs	107
3.30.1 FFT Peak Data	108
3.31 : FFT:SPECTrogram	109
: FFT:SPECTrogram:STATe	109
: FFT:SPECTrogram:STATe?	109
: FFT:SPECTrogram:SOURce	109
: FFT:SPECTrogram:SOURce?	109
: FFT:SPECTrogram:THReshoLd:LOWer	109
: FFT:SPECTrogram:THReshoLd:LOWer?	109
: FFT:SPECTrogram:THReshoLd:UPPer	109
: FFT:SPECTrogram:THReshoLd:UPPer?	110
: FFT:SPECTrogram:CGRading	110
: FFT:SPECTrogram:CGRading?	110
3.32 : FFT:CURSor	111
: FFT:CURSor:STATe	111
: FFT:CURSor:STATe?	111
: FFT:CURSor:SOURce	111

:FFT:CURSor:SOURce?	111
:FFT:CURSor:TRACking	111
:FFT:CURSor:TRACking?	111
:FFT:CURSor:[X/Y]:STATe	112
:FFT:CURSor:[X/Y]:STATe?	112
:FFT:CURSor:[X/Y]:COUPling	112
:FFT:CURSor:[X/Y]:COUPling?	112
:FFT:CURSor:[A/B]:STATe	112
:FFT:CURSor:[A/B]:STATe?	112
:FFT:CURSor:[A/B]:[X/Y]:STATe	112
:FFT:CURSor:[A/B]:[X/Y]:STATe?	113
:FFT:CURSor:[A/B]:[X/Y]:POSition	113
:FFT:CURSor:[A/B]:[X/Y]:POSition?	113
3.33 :FFT:MARKer	114
:FFT:MARKer:STATe	114
:FFT:MARKer:STATe?	114
:FFT:MARKer:SOURce	114
:FFT:MARKer:SOURce?	114
:FFT:MARKer:CLEar	114
:FFT:MARKer[MarkerN]:STATe	114
:FFT:MARKer[MarkerN]:STATe?	114
:FFT:MARKer[MarkerN]:FREQuency	115
:FFT:MARKer[MarkerN]:FREQuency?	115
:FFT:MARKer[MarkerN]:AMPLitude	115
:FFT:MARKer[MarkerN]:AMPLitude?	115
:FFT:MARKer[MarkerN]:ADD	115
:FFT:MARKer[MarkerN]:REMOve	115
3.34 :DECode	116
:DECode:STATe	116
:DECode:STATe?	116
:DECode:CLEar	116
:DECode[N]:STATe	116
:DECode[N]:STATe?	116
:DECode[N]:NFORmat	116
:DECode[N]:NFORmat?	117
:DECode[N]:TABLe	117
:DECode[N]:TABLe?	117
:DECode[N]:ADD	117
:DECode[N]:REMOve	117
3.35 :DECode[N]:I2C	118
:DECode[N]:I2C:CLOCK:SOURce	118
:DECode[N]:I2C:CLOCK:SOURce?	118
:DECode[N]:I2C:CLOCK:SLOPe	118
:DECode[N]:I2C:CLOCK:SLOPe?	118
:DECode[N]:I2C:CLOCK:THReshoLd:LOWer	118
:DECode[N]:I2C:CLOCK:THReshoLd:LOWer?	118
:DECode[N]:I2C:CLOCK:THReshoLd:UPPer	118
:DECode[N]:I2C:CLOCK:THReshoLd:UPPer?	119
:DECode[N]:I2C:DATA:SOURce	119
:DECode[N]:I2C:DATA:SOURce?	119
:DECode[N]:I2C:DATA:ACTive	119
:DECode[N]:I2C:DATA:ACTive?	119
:DECode[N]:I2C:DATA:THReshoLd:LOWer	119
:DECode[N]:I2C:DATA:THReshoLd:LOWer?	119
:DECode[N]:I2C:DATA:THReshoLd:UPPer	119
:DECode[N]:I2C:DATA:THReshoLd:UPPer?	120
3.36 :DECode[N]:SPI	121
:DECode[N]:SPI:TIMEout	121

:DECode[N]:SPI:TIMEout?	121
:DECode[N]:SPI:BITS	121
:DECode[N]:SPI:BITS?	121
:DECode[N]:SPI:BORDER	121
:DECode[N]:SPI:BORDER?	121
:DECode[N]:SPI:CSElect:SOURce	121
:DECode[N]:SPI:CSElect:SOURce?	122
:DECode[N]:SPI:CSElect:ACTive	122
:DECode[N]:SPI:CSElect:ACTive?	122
:DECode[N]:SPI:CSElect:THReshoId:LOWer	122
:DECode[N]:SPI:CSElect:THReshoId:LOWer?	122
:DECode[N]:SPI:CSElect:THReshoId:UPPer	122
:DECode[N]:SPI:CSElect:THReshoId:UPPer?	122
:DECode[N]:SPI:CLOCK:SOURce	122
:DECode[N]:SPI:CLOCK:SOURce?	123
:DECode[N]:SPI:CLOCK:SLOPe	123
:DECode[N]:SPI:CLOCK:SLOPe?	123
:DECode[N]:SPI:CLOCK:THReshoId:LOWer	123
:DECode[N]:SPI:CLOCK:THReshoId:LOWer?	123
:DECode[N]:SPI:CLOCK:THReshoId:UPPer	123
:DECode[N]:SPI:CLOCK:THReshoId:UPPer?	123
:DECode[N]:SPI:MOSi:SOURce	123
:DECode[N]:SPI:MOSi:SOURce?	124
:DECode[N]:SPI:MOSi:ACTive	124
:DECode[N]:SPI:MOSi:ACTive?	124
:DECode[N]:SPI:MOSi:THReshoId:LOWer	124
:DECode[N]:SPI:MOSi:THReshoId:LOWer?	124
:DECode[N]:SPI:MOSi:THReshoId:UPPer	124
:DECode[N]:SPI:MOSi:THReshoId:UPPer?	124
:DECode[N]:SPI:MISo:SOURce	124
:DECode[N]:SPI:MISo:SOURce?	125
:DECode[N]:SPI:MISo:ACTive	125
:DECode[N]:SPI:MISo:ACTive?	125
:DECode[N]:SPI:MISo:THReshoId:LOWer	125
:DECode[N]:SPI:MISo:THReshoId:LOWer?	125
:DECode[N]:SPI:MISo:THReshoId:UPPer	125
:DECode[N]:SPI:MISo:THReshoId:UPPer?	125
3.37 :DECode[N]:UART	126
:DECode[N]:UART:BRATe	126
:DECode[N]:UART:BRATe?	126
:DECode[N]:UART:PARity	126
:DECode[N]:UART:PARity?	126
:DECode[N]:UART:SBIT	126
:DECode[N]:UART:SBIT?	127
:DECode[N]:UART:BITS	127
:DECode[N]:UART:BITS?	127
:DECode[N]:UART:BORDER	127
:DECode[N]:UART:BORDER?	127
:DECode[N]:UART:RTX:SOURce	127
:DECode[N]:UART:RTX:SOURce?	127
:DECode[N]:UART:RTX:ACTive	128
:DECode[N]:UART:RTX:ACTive?	128
:DECode[N]:UART:RTX:THReshoId:LOWer	128
:DECode[N]:UART:RTX:THReshoId:LOWer?	128
:DECode[N]:UART:RTX:THReshoId:UPPer	128
:DECode[N]:UART:RTX:THReshoId:UPPer?	128
3.38 :DECode[N]:PARAllel	129
:DECode[N]:PARAllel:CLOCK:SOURce	129

:DECode[N]:PARAllel:CLOCK:SOURce?	129
:DECode[N]:PARAllel:CLOCK:SLOPe	129
:DECode[N]:PARAllel:CLOCK:SLOPe?	129
:DECode[N]:PARAllel:CLOCK:THReshold:LOWer	129
:DECode[N]:PARAllel:CLOCK:THReshold:LOWer?	129
:DECode[N]:PARAllel:CLOCK:THReshold:UPPer	129
:DECode[N]:PARAllel:CLOCK:THReshold:UPPer?	130
:DECode[N]:PARAllel:BITS	130
:DECode[N]:PARAllel:BITS?	130
:DECode[N]:PARAllel:BIT[BitN]:SOURce	130
:DECode[N]:PARAllel:BIT[BitN]:SOURce?	130
:DECode[N]:PARAllel:BIT[BitN]:ACTive	130
:DECode[N]:PARAllel:BIT[BitN]:ACTive?	130
:DECode[N]:PARAllel:BIT[BitN]:THReshold:LOWer	130
:DECode[N]:PARAllel:BIT[BitN]:THReshold:LOWer?	131
:DECode[N]:PARAllel:BIT[BitN]:THReshold:UPPer	131
:DECode[N]:PARAllel:BIT[BitN]:THReshold:UPPer?	131
3.39 :CURSor	132
:CURSor:STATE	132
:CURSor:STATE?	132
:CURSor:SOURce	132
:CURSor:SOURce?	132
:CURSor:TDELta	132
:CURSor:TDELta?	132
:CURSor:TRACKing	133
:CURSor:TRACKing?	133
:CURSor:[X/Y]:STATE	133
:CURSor:[X/Y]:STATE?	133
:CURSor:[X/Y]:COUPLing	133
:CURSor:[X/Y]:COUPLing?	133
:CURSor:[A/B]:STATE	133
:CURSor:[A/B]:STATE?	134
:CURSor:[A/B]:[X/Y]:STATE	134
:CURSor:[A/B]:[X/Y]:STATE?	134
:CURSor:[A/B]:[X/Y]:POSition	134
:CURSor:[A/B]:[X/Y]:POSition?	134
3.40 :FGENERator	135
:FGENERator:STATE	135
:FGENERator:STATE?	135
:FGENERator:LOAD	135
:FGENERator:LOAD?	135
3.41 :FGENERator:WAVEform	136
:FGENERator:WAVEform:SHAPE	136
:FGENERator:WAVEform:SHAPE?	136
:FGENERator:WAVEform:FREQUency	136
:FGENERator:WAVEform:FREQUency?	136
:FGENERator:WAVEform:PERiod	136
:FGENERator:WAVEform:PERiod?	137
:FGENERator:WAVEform:AMPLitude	137
:FGENERator:WAVEform:AMPLitude?	137
:FGENERator:WAVEform:RMS	137
:FGENERator:WAVEform:RMS?	137
:FGENERator:WAVEform:OFFSet	137
:FGENERator:WAVEform:OFFSet?	137
:FGENERator:WAVEform:LEVeL:HIGH	137
:FGENERator:WAVEform:LEVeL:HIGH?	138
:FGENERator:WAVEform:LEVeL:LOW	138
:FGENERator:WAVEform:LEVeL:LOW?	138

3.42	:FGENERator:WAVEform:RECTangle	139
	:FGENERator:WAVEform:RECTangle:DUTY	139
	:FGENERator:WAVEform:RECTangle:DUTY?	139
3.43	:FGENERator:WAVEform:PULSe	140
	:FGENERator:WAVEform:PULSe:DUTY	140
	:FGENERator:WAVEform:PULSe:DUTY?	140
	:FGENERator:WAVEform:PULSe:RTIME	140
	:FGENERator:WAVEform:PULSe:RTIME?	140
	:FGENERator:WAVEform:PULSe:FTIME	140
	:FGENERator:WAVEform:PULSe:FTIME?	140
3.44	:FGENERator:WAVEform:RAMP	141
	:FGENERator:WAVEform:RAMP:SYMMetry	141
	:FGENERator:WAVEform:RAMP:SYMMetry?	141
3.45	:FGENERator:MODIfication	142
	:FGENERator:MODIfication:TYPe	142
	:FGENERator:MODIfication:TYPe?	142
3.46	:FGENERator:MODIfication:MODulation	143
	:FGENERator:MODIfication:MODulation:TYPe	143
	:FGENERator:MODIfication:MODulation:TYPe?	143
	:FGENERator:MODIfication:MODulation:SHApe	143
	:FGENERator:MODIfication:MODulation:SHApe?	143
	:FGENERator:MODIfication:MODulation:FREQuency	144
	:FGENERator:MODIfication:MODulation:FREQuency?	144
	:FGENERator:MODIfication:MODulation:AM:DEPTh	144
	:FGENERator:MODIfication:MODulation:AM:DEPTh?	144
	:FGENERator:MODIfication:MODulation:DSSCAM:DEPTh	144
	:FGENERator:MODIfication:MODulation:DSSCAM:DEPTh?	144
	:FGENERator:MODIfication:MODulation:FM:FDEVIation	144
	:FGENERator:MODIfication:MODulation:FM:FDEVIation?	144
	:FGENERator:MODIfication:MODulation:PM:PDEVIation	145
	:FGENERator:MODIfication:MODulation:PM:PDEVIation?	145
3.47	:FGENERator:MODIfication:SWEEp	146
	:FGENERator:MODIfication:SWEEp:TYPe	146
	:FGENERator:MODIfication:SWEEp:TYPe?	146
	:FGENERator:MODIfication:SWEEp:DIRection	146
	:FGENERator:MODIfication:SWEEp:DIRection?	146
	:FGENERator:MODIfication:SWEEp:TIME:UP	146
	:FGENERator:MODIfication:SWEEp:TIME:UP?	146
	:FGENERator:MODIfication:SWEEp:TIME:DOWN	147
	:FGENERator:MODIfication:SWEEp:TIME:DOWN?	147
	:FGENERator:MODIfication:SWEEp:FREQuency:START	147
	:FGENERator:MODIfication:SWEEp:FREQuency:START?	147
	:FGENERator:MODIfication:SWEEp:FREQuency:STOP	147
	:FGENERator:MODIfication:SWEEp:FREQuency:STOP?	147
	:FGENERator:MODIfication:SWEEp:HOLD:START	147
	:FGENERator:MODIfication:SWEEp:HOLD:START?	147
	:FGENERator:MODIfication:SWEEp:HOLD:STOP	148
	:FGENERator:MODIfication:SWEEp:HOLD:STOP?	148
3.48	:FGENERator:MODIfication:BURSt	149
	:FGENERator:MODIfication:BURSt:TTYPE	149
	:FGENERator:MODIfication:BURSt:TTYPE?	149
	:FGENERator:MODIfication:BURSt:SPHase	149
	:FGENERator:MODIfication:BURSt:SPHase?	149
	:FGENERator:MODIfication:BURSt:CYCLes	149
	:FGENERator:MODIfication:BURSt:CYCLes?	149
	:FGENERator:MODIfication:BURSt:TDELay	149
	:FGENERator:MODIfication:BURSt:TDELay?	150
	:FGENERator:MODIfication:BURSt:PERiod	150

:FGENERator:MODIfication:BURSt:PERIod?	150
:FGENERator:MODIfication:BURSt:ILTYpe	150
:FGENERator:MODIfication:BURSt:ILTYpe?	150
:FGENERator:MODIfication:BURSt:ILEVeL	150
:FGENERator:MODIfication:BURSt:ILEVeL?	150
:FGENERator:MODIfication:BURSt:TRIGger	150
3.49 :BPLot	151
:BPLot:STATe	151
:BPLot:STATe?	151
:BPLot:FSCale	151
:BPLot:FSCale?	151
:BPLot:VMODE	151
:BPLot:VMODE?	151
:BPLot:ASCale	152
:BPLot:ASCale?	152
:BPLot:RSCale	152
:BPLot:RSCale?	152
:BPLot:FREQuency:CENTer	152
:BPLot:FREQuency:CENTer?	152
:BPLot:FREQuency:SPAN	153
:BPLot:FREQuency:SPAN?	153
:BPLot:FREQuency:START	153
:BPLot:FREQuency:START?	153
:BPLot:FREQuency:STOP	153
:BPLot:FREQuency:STOP?	153
:BPLot[N]:STATe	153
:BPLot[N]:STATe?	153
:BPLot[N]:OUT	154
:BPLot[N]:OUT?	154
:BPLot[N]:OFFSet	154
:BPLot[N]:OFFSet?	154
:BPLot[N]:SCALe	154
:BPLot[N]:SCALe?	154
:BPLot[N]:POFFSet	155
:BPLot[N]:POFFSet?	155
:BPLot[N]:PSCALe	155
:BPLot[N]:PSCALe?	155
3.50 :BPLot:SWEEp	156
:BPLot[N]:IN	156
:BPLot[N]:IN?	156
:BPLot:SWEEp:POINts	156
:BPLot:SWEEp:POINts?	156
:BPLot:SWEEp:PDISTribution	156
:BPLot:SWEEp:PDISTribution?	156
:BPLot:SWEEp:FREQuency:START	156
:BPLot:SWEEp:FREQuency:START?	157
:BPLot:SWEEp:FREQuency:STOP	157
:BPLot:SWEEp:FREQuency:STOP?	157
:BPLot:SWEEp:VSCale	157
:BPLot:SWEEp:VSCale?	157
:BPLot:SWEEp:VOFFset	157
:BPLot:SWEEp:VOFFset?	157
:BPLot:SWEEp:VAMPLitude	158
:BPLot:SWEEp:VAMPLitude?	158
3.51 :BPLot[N]:DATA	159
:BPLot[N]:DATA:ASCale	159
:BPLot[N]:DATA:ASCale?	159
:BPLot[N]:DATA:POINts?	159

:BPLOT[N]:DATA:PACKED?	159
3.51.1 Bode Plot Points	160
3.51.2 Bode Plot Points Packed	160
3.52 :HISTORY	161
:HISTORY:STATE	161
:HISTORY:STATE?	161
:HISTORY:RECORD	161
:HISTORY:RECORD?	161
:HISTORY:TOTAL?	161
:HISTORY:NEXT	161
:HISTORY:PREVIOUS	161
:HISTORY:PLAY	161
:HISTORY:STOP	161
:HISTORY:REPEAT	162
:HISTORY:REPEAT?	162
3.53 :DISPLAY	163
:DISPLAY:BRIGHTNESS	163
:DISPLAY:BRIGHTNESS?	163
:DISPLAY:SCREENSHOT?	163
3.54 :DISPLAY:WAVEFORM	164
:DISPLAY:WAVEFORM:INTERPOLATION	164
:DISPLAY:WAVEFORM:INTERPOLATION?	164
:DISPLAY:WAVEFORM:INTENSITY	164
:DISPLAY:WAVEFORM:INTENSITY?	164
:DISPLAY:WAVEFORM:COLOR	164
:DISPLAY:WAVEFORM:COLOR?	165
:DISPLAY:WAVEFORM:PERSISTENCE	165
:DISPLAY:WAVEFORM:PERSISTENCE?	165
3.55 :DISPLAY:GRID	166
:DISPLAY:GRID:DETAIL	166
:DISPLAY:GRID:DETAIL?	166
:DISPLAY:GRID:INTENSITY	166
:DISPLAY:GRID:INTENSITY?	166
:DISPLAY:GRID:MLABEL	166
:DISPLAY:GRID:MLABEL?	167
3.56 :ZOOM	168
:ZOOM:STATE	168
:ZOOM:STATE?	168
:ZOOM:HORIZONTAL:POSITION	168
:ZOOM:HORIZONTAL:POSITION?	168
:ZOOM:HORIZONTAL:SCALE	168
:ZOOM:HORIZONTAL:SCALE?	168
:ZOOM:VERTICAL:POSITION	168
:ZOOM:VERTICAL:POSITION?	169
:ZOOM:VERTICAL:SCALE	169
:ZOOM:VERTICAL:SCALE?	169

1 Overview

This document provides a comprehensive overview of the SCPI (Standard Commands for Programmable Instruments) commands and queries supported by the Magnova BMO oscilloscope. SCPI is an established standard for controlling and communicating with test and measurement instruments, enabling users to automate complex measurements and integrate devices into larger testing systems.

SCPI builds upon the IEEE 488.1 and IEEE 488.2 standards, which define the physical, electrical, and protocol layers for communication with programmable instruments. By adhering to these standards, SCPI ensures compatibility across a wide range of instruments, facilitating seamless integration into automated testing environments.

In addition to listing the supported commands, this document provides detailed descriptions of each command, including their syntax and functionality. The commands are organized based on relevancy to typical use cases, though the importance of specific commands may naturally differ depending on individual user needs and applications. Programming examples are included to demonstrate how these commands can be effectively utilized over USB and Ethernet using various protocols such as USBTMC, raw sockets, HiSLIP (not yet unlocked as of firmware version 1.2.1), and a REST API, making it straightforward for users to implement them in real-world applications.

By following this guide, users can unlock the full potential of the Magnova BMO's programmable interface, streamlining measurement workflows, enhancing data accuracy, and reducing manual intervention. This document is intended for engineers, researchers, and developers seeking to leverage SCPI for automation in laboratories, manufacturing environments, and educational settings.

1.1 Syntax

A small summary of the syntax is provided in the following section.

1.1.1 Commands and Queries

SCPI commands and queries on the Magnova BMO oscilloscope follow a structured syntax. Each command or query comprises a header and, optionally, a list of parameters. The header begins with either a colon (:) for device-specific commands or an asterisk (*) for common commands. Path components within the header are separated by colons (:), and a question mark (?) at the end of the header signifies a query. Parameters, if present, are separated by commas (,) and follow a space after the header. Importantly, there must be no spaces between the parameters themselves. Commands can also be combined into a composite command by separating them with semicolons (;); however, only the result of the last query in the sequence is returned.

Examples

- To enable the first analog channel:
:CHANne11:STATe ON
- To query whether the first analog channel is enabled:
:CHANne11:STATe?
- To query packed waveform data from the second analog channel:
:CHANne12:DATA:PACKed? SCReen,V
- Composition of multiple commands:
:CHANne11:STATe ON; :TRIGger:TYPE EDGe

1.1.2 Parameters and Return Values

The following types are used in both commands and return values.

- **Discrete**

For discrete types, the possible values are explicitly listed and follow the same naming convention as the commands and queries themselves. For example:

- ON
- SCReen

- **Integer**

An integer, which may be constrained to a specific range if specified; it is important to note that a decimal in this context will result in an error.

- **Real**

A real number, which may be constrained to a specific range if specified.

- **ASCII**

A string of ASCII characters.

- **Block Data**

See 1.1.3.

Some parameters specify possible units. This allows users to optionally specify the value in engineering notation, including the unit. For example:

- 50mV
- 1.25dBm

The usage of a unit is always optional. However, when a unit is used, it must be one of the specified units. Return values never use a unit. For example:

- 0.05
- 1.25

1.1.3 Block Data

Queries that return large amounts of data are formatted using the *definite-length binary block format* as defined by the IEEE 488.2 standard.

Field	Count (bytes)	Description	Example
#	1	Hash character indicating block start	#
N = Digits	1	Length of byte count (number of digits)	5 (ASCII)
M = Byte Count	N	Number of data bytes	12345 (ASCII)
Binary Data	M	The actual binary data	Binary sequence

Unless explicitly specified, binary data uses little-endian byte order.

Depending on the API used to send SCPI commands and evaluate return values, the processing of the *definite-length binary block format* is typically handled automatically. However, the little-endian byte order may need to be explicitly specified.

The concrete block data definition in this document includes a **JSON/XML Tag**. When the RESTful API is used, the binary layout is ignored and the data is returned in fields with their respective names.

1.1.4 Abbreviation

The names of commands and parameter values are provided in their long form but can be abbreviated by using only their short form (upper case parts). For example:

- :CHAN1:STAT ON
- :CHAN2:DATA:PACK? SCR,V

When abbreviating, the abbreviation must be complete and consistent. Partial abbreviations are **not** valid. For instance:

- :CHANn1:STAT ON (invalid due to incomplete abbreviation)
- :CHAN2:DATA:PACK? SCR_{ee},V (this also applies to parameters)

It is important to note the special case of numbers at the end of a name. These numbers remain unchanged regardless of whether the long or short form is used.

1.1.5 Notes

All commands operate synchronously unless explicitly stated otherwise.

1.2 Interfaces

This section outlines the available communication interfaces that serve as transport mediums for SCPI.

1.2.1 Common

The following common communication interfaces are currently supported:

- **Raw sockets**
Raw sockets provide a lightweight communication method that allows direct transmission of SCPI commands over a network using TCP/IP without additional protocol overhead. Ideal for simple setups where minimal latency is required.
See Section 2.2 for an example on how to use raw sockets.
- **USBTCM**
The USB Test and Measurement Class protocol enables communication with SCPI devices via USB. It provides a standardized way to control instruments and transfer data using USB connections.
See Section 2.1 for an example on how to use USBTCM.
- **HiSLIP**
The High-Speed LAN Instrument Protocol is a network-based protocol designed for SCPI communication. It offers improved performance and capabilities compared to traditional raw sockets and is suited for modern, high-speed instruments.
Note: Not yet unlocked as of firmware version 1.2.1.

1.2.2 RESTful API

In addition to the commonly used and well-documented interfaces, a RESTful API (Representational State Transfer) is also available. REST is an architectural style for designing networked applications that uses standard HTTP methods (such as GET, POST, PUT, DELETE) to perform operations on resources. This stateless communication approach ensures that each request from a client to the server is independent and contains all the necessary information, making it versatile for various applications and easy to integrate into different systems.

See Section 2.3 for an example on how to use the REST API.

Note: All provided REST API requests that return **application/json** can also return **application/xml** if requested.

Version Get the device firmware version.

GET	/version <i>Get the device firmware version</i>
Parameter	
<i>no parameter</i>	
Response	application/json
200 ok	
	<pre>{ "Major": 1, "Minor": 1, "Revision": 7 }</pre>

Screenshot Take a screenshot of the currently displayed device screen.

GET	/screenshot <i>Get a screenshot</i>
Parameter	
<i>no parameter</i>	
Response	image/png
200 ok	

SCPI Interact with the SCPI interface.

GET	/scpi <i>Get an alphabetically sorted list of available commands and queries</i>
Parameter	
<i>no parameter</i>	
Response	application/json
200 ok	
	<pre>{ "*CLS", "*ESE", "...", "CHANne1:STATe", "CHANne1:SCALe", "... }</pre>

POST	/scpi <i>Execute a command</i>
Parameter	
<i>no parameter</i>	
Body	application/json
<pre> :CHANel1:STAT ON </pre>	
Response	
application/json	
200	ok
404	error: command not found
400	error: invalid syntax
<pre> "Invalid command syntax" </pre>	
400	error: invalid parameters
<pre> "Invalid command parameters" </pre>	

POST	/scpi <i>Get the result of executing a query</i>
Parameter	
<i>no parameter</i>	
Body	application/json
<pre> :CHANel1:STAT? </pre>	
Response	
application/json	
200	ok
<pre> "ON" </pre>	
404	error: command not found
400	error: invalid syntax
<pre> "Invalid command syntax" </pre>	
400	error: invalid parameters
<pre> "Invalid command parameters" </pre>	

2 Examples

The following section provides several examples of using SCPI commands to remotely control the instrument via various protocols in Python. See also <https://github.com/Batronix/MagnovaExamples> for more examples.

2.1 Simple instrument commands and queries using USBTMC

In this example, the instrument's unique identifier is queried, the first analog channel is enabled, and the acquisition is started.

```
#!/usr/bin/python3

import pyvisa

# Open the instrument using a USBTMC connection
rm = pyvisa.ResourceManager()
instr = rm.open_resource('USB0::0x19B2::0x0030::[YOUR_SERIAL_NUMBER]::INSTR')

# Query the unique identifier
print(instr.query('*IDN?'))

# Enable the first analog channel
instr.write("CHAN1:STAT ON")

# Start the acquisition
instr.write("RUN")
```

2.2 Waveform Plot using raw sockets

In this example, waveform data is queried in packed format, requiring the metadata to be separated from the waveform samples.

```
#!/usr/bin/python3

import matplotlib.pyplot as plt
import numpy as np
import pyvisa
import struct

# Open the instrument using a raw TCP/IP connection with required newline termination
rm = pyvisa.ResourceManager()
instr = rm.open_resource('TCPIP::[YOUR_INSTRUMENT_IP]::5025::SOCKET')
instr.read_termination = '\n'

# Query the packed waveform data
data = instr.query_binary_values('CHAN1:DATA:PACK? ALL,RAW', datatype='B')

# Define the struct format for the 32-byte metadata
metadata_format = 'fffIIfFI' # Corresponds to TimeDelta, StartTime, EndTime, etc.
metadata_size = struct.calcsize(metadata_format)

# Extract the first 32 bytes of metadata from the data
metadata_bytes = bytes(data[:metadata_size]) # Convert the metadata part to a byte string
metadata = struct.unpack(metadata_format, metadata_bytes)

# Assign metadata to meaningful variable names
(TimeDelta, StartTime, EndTime, SampleStart, SampleLength,
 VerticalStart, VerticalLength, SampleCount) = metadata

# Print parsed metadata
print(f"Metadata")
print(f"  TimeDelta      = {TimeDelta}")
print(f"  StartTime      = {StartTime}")
print(f"  EndTime        = {EndTime}")
print(f"  SampleStart    = {SampleStart}")
print(f"  SampleLength   = {SampleLength}")
print(f"  VerticalStart   = {VerticalStart}")
print(f"  VerticalLength = {VerticalLength}")
print(f"  SampleCount    = {SampleCount}")

# Extract samples starting from byte 33 onwards (after metadata) and convert into 16-bit uint
waveform_data_bytes = data[metadata_size:]
waveform_data = np.frombuffer(bytes(waveform_data_bytes), dtype=np.uint16)

# Normalize sample values based on VerticalStart and VerticalLength
amplitude_data = VerticalStart + (waveform_data / (SampleLength - 1)) * VerticalLength

# Create the timebase
timebase = np.linspace(StartTime, EndTime, len(amplitude_data), endpoint=True)

plt.plot(timebase, amplitude_data)
plt.xlabel('Time in s')
plt.ylabel('Amplitude in V')
plt.title('Waveform')
plt.show()
```

2.3 Waveform Plot using the REST API

In this example, waveform data is queried in packed format using the default parameters ALL and V. Unlike the raw socket example, the metadata is stored in separate JSON fields.

```
#!/usr/bin/python3

import matplotlib.pyplot as plt
import numpy as np
import requests

# Query the packed data with a POST to the REST API and interpret the response as json
data = requests.post("http://[YOUR_INSTRUMENT_IP]:8080/scpi", json="CHAN1:DATA:PACK?").json()

TimeDelta = data["TimeDelta"]
StartTime = data["StartTime"]
EndTime = data["EndTime"]
SampleCount = data["SampleCount"]

# Print metadata
print(f"Metadata")
print(f"  TimeDelta      = {TimeDelta}")
print(f"  StartTime       = {StartTime}")
print(f"  EndTime         = {EndTime}")
print(f"  SampleCount     = {SampleCount}")

# The samples can be directly fetched from the response json object
y = data["Samples"]

# Create the timebase
x = np.linspace(data["StartTime"], data["EndTime"], len(y), endpoint=True)

plt.plot(x, y)
plt.xlabel('Time in s')
plt.ylabel('Amplitude in V')
plt.title('Waveform')
plt.show()
```


2.4 FFT Plot using raw sockets

In this example, FFT bins are queried in packed format using the default parameter DBM.

```
#!/usr/bin/python3

import matplotlib.pyplot as plt
import numpy as np
import pyvisa
import struct

# Open the instrument using a raw TCP/IP connection with required newline termination
rm = pyvisa.ResourceManager()
instr = rm.open_resource('TCPIP::[YOUR_INSTRUMENT_IP]::5025::SOCKET')
instr.read_termination = '\n'

# Query the packed waveform data
data = instr.query_binary_values('FFT1:DATA:PACK?', datatype='B')

# Define the struct format for the 12-byte metadata
metadata_format = 'ffI' # Corresponds to BinFrequency, StopFrequency and BinCount
metadata_size = struct.calcsize(metadata_format)

# Extract the first 12 bytes of metadata from the data
metadata_bytes = bytes(data[:metadata_size]) # Convert the metadata part to a byte string
metadata = struct.unpack(metadata_format, metadata_bytes)

# Assign metadata to meaningful variable names
BinFrequency, StopFrequency, BinCount = metadata

# Print parsed metadata
print(f"Metadata")
print(f"  BinFrequency   = {BinFrequency}")
print(f"  StopFrequency  = {StopFrequency}")
print(f"  BinCount       = {BinCount}")

# Extract bins starting from byte 13 onwards (after metadata) and convert into 32-bit floats
bins_bytes = data[metadata_size:]
bins = np.frombuffer(bytes(bins_bytes), dtype=np.float32)

# Create the frequency base
frequency_base = np.linspace(0, StopFrequency, len(bins), endpoint=True)

plt.plot(frequency_base, bins)
plt.xlabel('Frequency in Hz')
plt.ylabel('Amplitude in dBm')
plt.title('FFT')
plt.show()
```

2.5 Measurement using raw sockets

In this example, the VAMP measurement is queried for CHAN1.

```
#!/usr/bin/python3

import matplotlib.pyplot as plt
import numpy as np
import pyvisa
import struct

# Open the instrument using a raw TCP/IP connection with required newline termination
rm = pyvisa.ResourceManager()
instr = rm.open_resource('TCPIP::[YOUR_INSTRUMENT_IP]::5025::SOCKET')
instr.read_termination = '\n'

# Query all statistics for the measurement
data = instr.query_binary_values('MEAS:VAMP:ALL? CHAN1', datatype='B')

# Define the struct format for the statistics
statistics_format = 'dddddI'
statistics_size = struct.calcsize(statistics_format)

# Unpack the statistics
statistics = struct.unpack(statistics_format, bytes(data))

# Assign statistics to meaningful variable names
(Current, Average, Maximum, Minimum, Deviation, Count) = statistics

# Print parsed statistics
print(f"Statistics")
print(f"  Current      = {Current}")
print(f"  Average      = {Average}")
print(f"  Maximum      = {Maximum}")
print(f"  Minimum      = {Minimum}")
print(f"  Deviation    = {Deviation}")
print(f"  Count        = {Count}")
```

3 Commands and Queries

The following sections describe the various available SCPI commands and queries. For an in-depth explanation of how these features work, please consult the Batronix Magnova® User Manual.

- **Placeholders**

Brackets ([and]) indicate placeholders that must be replaced with specific values, as outlined in the corresponding sections. For example, :CHAN would be replaced with :CHAN2 to reference the second analog channel. Detailed rules for replacing placeholders are provided in the relevant sections. When multiple values are specified separated by / , one must be selected.

3.1 Common

The common commands declared mandatory by *IEEE 488.2*.

***CLS**

***ESE**

***ESE?**

***ESR?**

***IDN?**

***OPC**

***OPC?**

***RST**

***SRE**

***SRE?**

***STB**

***STB?**

***TST?**

***WAI**

3.2 Control

This section focuses on controlling the state of waveform acquisition, including managing synchronization with the acquisition of new records.

:RUN

:STOP

:SINGle

:FORCe

:AUTO <State>

Parameters

State

ON or 1

OFF or 0

:AUTO?

Returns

ON

OFF

:CLEAr

:SEQUence:STOP

:SEQUence:WAIT? <Waveforms>

Parameters

Waveforms

Integer

The number of records to wait for. Default 1

Returns

The actual number of records acquired while waiting

3.3 :ACquire

This section covers the general configuration of waveform acquisition.

:ACquire:MDEPth <Depth>

Returns the memory depth per analog channel and digital channel group. The digital channels 0-7 and 8-15 each form a group.

Parameters

Depth

AUTO

AFASt

Integer in pts

:ACquire:MDEPth?

Sets the memory depth per analog channel and digital channel group. The digital channels 0-7 and 8-15 each form a group.

Returns

AUTO

AFASt

Integer in pts

Changes to other parameters that explicitly or implicitly activate channels also modify the memory depth per channel.

:ACquire:TEXPansion <Expansion>

Parameters

Expansion

ON or 1

OFF or 0

:ACquire:TEXPansion?

Returns

ON

OFF

:ACquire:ROLLmode <Mode>

Parameters

Mode

ON or 1

OFF or 0

:ACQuire:ROLLmode?

Returns

ON**OFF**

:ACQuire:HRESolution <Resolution>

Parameters

Resolution**12****12.5****13****13.5****14****14.5****15****15.5****16**

:ACQuire:HRESolution?

Returns

12**12.5****13****13.5****14****14.5****15****15.5****16**

:ACQuire:RCLock <Clock>

Parameters

Clock**INTernal****EXTernal**

:ACQUIRE:RCLOCK?

Returns

INTERNAL**EXTERNAL**

:ACQUIRE:RCOUT <State>

Parameters

State**ON or 1****OFF or 0**

:ACQUIRE:RCOUT?

Returns

ON**OFF**

:ACQUIRE:MODE <Mode>

Parameters

Mode**SAMPLE****PDETECT****AVERAGE**

:ACQUIRE:MODE?

Returns

SAMPLE**PDETECT****AVERAGE**

:ACQUIRE:AVERAGE <Average>

Parameters

Average**Integer**

:ACQUIRE:AVERAGE?

Returns

Integer

:ACQuire:SRATe?**Returns****Integer in pts/s**

3.4 :TIMEbase

This section covers the configuration of the timebase.

:TIMEbase:REFerence <Reference>

Parameters

Reference

CENTer

LEFT

RIGHT

TRIGger

:TIMEbase:REFerence?

Returns

CENTer

LEFT

RIGHT

TRIGger

:TIMEbase:OFFSet <Offset>

Parameters

Offset

Real in s

:TIMEbase:OFFSet?

Returns

Real in s

:TIMEbase:SCALE <Scale>

Parameters

Scale

Real in s

:TIMEbase:SCALE?

Returns

Real in s

3.5 :CHANneL

This section covers the configuration of analog channels.

- [N] has to be replaced with an analog channel index, for example: CHANneL1

:CHANneL:STATE <State>

Parameters

State

ON or 1

OFF or 0

:CHANneL:STATE?

Returns

ON

OFF

:CHANneL[N]:STATE <State>

Parameters

State

ON or 1

OFF or 0

:CHANneL[N]:STATE?

Returns

ON

OFF

:CHANneL[N]:TERMination <Termination>

Parameters

Termination

ON or 1

OFF or 0

:CHANneL[N]:TERMination?

Returns

ON

OFF

:CHANnel[N]:COUpling <Coupling>

Parameters

Coupling

AC

DC

:CHANnel[N]:COUpling?

Returns

AC

DC

:CHANnel[N]:DESKew <Delay>

Parameters

Delay

Real in s

:CHANnel[N]:DESKew?

Returns

Real in s

:CHANnel[N]:DIVisor <Divisor>

Parameters

Divisor

Numerator/Denominator For example 1/100 or 20/5

:CHANnel[N]:DIVisor?

Returns

Numerator/Denominator For example 1/100 or 20/5

:CHANnel[N]:UNIT <Unit>

Parameters

Unit

V

A

:CHANne1[N]:UNIT?

Returns

V

A

:CHANne1[N]:OFFSet <Offset>

Parameters

Offset

Real in V or A

:CHANne1[N]:OFFSet?

Returns

Real in V or A

:CHANne1[N]:SCALE <Scale>

Parameters

Scale

Real in V or A

:CHANne1[N]:SCALE?

Returns

Real in V or A

3.6 :CHANneL[N]:DATA

This section focuses on the transfer of analog waveform data.

- [N] has to be replaced with an analog channel index, for example: CHANneL1

:CHANneL[N]:DATA:SOURce <Source>

Parameters

Source

SCReen

ALL

:CHANneL[N]:DATA:SOURce?

Returns

SCReen

ALL

:CHANneL[N]:DATA:TYPE <Type>

Parameters

Type

V

RAW

:CHANneL[N]:DATA:TYPE?

Returns

V

RAW

:CHANneL[N]:DATA:RECORD <Record>

Parameters

Record

Integer

Negative to start at the back, for example -1 is the last record

:CHANneL[N]:DATA:RECORD?

Returns

Integer

Negative to start at the back, for example -1 is the last record

:CHANneL[N]:DATA:TDELta?

Returns

Real in s

:CHANnel[N]:DATA:STIME?

Returns

Real in s

:CHANnel[N]:DATA:ETIME?

Returns

Real in s

:CHANnel[N]:DATA:SStart?

Returns

Integer

:CHANnel[N]:DATA:SLENGth?

Returns

Integer

:CHANnel[N]:DATA:VStart?

Returns

Real in V

:CHANnel[N]:DATA:VLENGth?

Returns

Real in V

:CHANnel[N]:DATA:SAMPles?

See :CHANnel[N]:DATA:SOURce, :CHANnel[N]:DATA:TYPE and :CHANnel[N]:DATA:RECORD for parameterization.

Returns

Sample data

See 3.6.1. The metadata can be queried from the various :CHANnel[N]:DATA sub-commands.

:CHANneł[N]:DATA:PACKed? <Source>, <Type>, <Record>

:CHANneł[N]:DATA:SOURce, :CHANneł[N]:DATA:TYPE and :CHANneł[N]:DATA:RECORD are ignored and specified through the parameters instead.

Parameters

Source

SCReen

ALL

Default

Type

V

RAW

Default

Record

Integer

See RECOrd. Default -1

Returns

Packed sample data

See 3.6.2.

3.6.1 Analog Data Samples

All data in this section has a *Block Data* header (see 1.1.3).

Analog waveform samples in V N is implicitly defined through the *Block Data* header

Field	Count (bytes)	Data Type	Json/XML Name
Samples	4 * N	4-byte floating-point	Samples

Analog waveform samples as raw N is implicitly defined through the *Block Data* header

Field	Count (bytes)	Data Type	Json/XML Name
Samples	2 * N	2-byte unsigned integer	Samples

3.6.2 Analog Data Packed

All data in this section has a *Block Data* header (see 1.1.3).

Analog waveform samples in V including a metadata header

Field	Count (bytes)	Data Type	Json/XML Name
Time between samples	4	4-byte floating-point	TimeDelta
Time of the first sample	4	4-byte floating-point	StartTime
Time of the last sample	4	4-byte floating-point	EndTime
N = Number of samples	4	4-byte unsigned integer	SampleCount
Samples	4 * N	4-byte floating-point	Samples

Analog waveform samples as raw including a metadata header

Field	Count (bytes)	Data Type	Json/XML Name
Time between samples	4	4-byte floating-point	TimeDelta
Time of the first sample	4	4-byte floating-point	StartTime
Time of the last sample	4	4-byte floating-point	EndTime
Offset to the valid part of the sample value	4	4-byte unsigned integer	SampleStart
Length of the valid part of the sample value	4	4-byte unsigned integer	SampleLength
Voltage associated with the start of the valid sample value	4	4-byte floating-point	VerticalStart
Voltage associated with the length of the valid sample value	4	4-byte unsigned integer	VerticalLength
N = Number of samples	4	4-byte unsigned integer	SampleCount
Samples	2 * N	2-byte unsigned integer	Samples

3.7 :DIGital

This section covers the configuration of digital channels.

- [N] has to be replaced with a digital channel index, for example: DIGital5

:DIGital:STATE <State>

Parameters

State

ON or 1

OFF or 0

:DIGital:RST

:DIGital:STATE?

Returns

ON

OFF

:DIGital[N]:STATE <State>

Parameters

State

ON or 1

OFF or 0

:DIGital[N]:STATE?

Returns

ON

OFF

:DIGital:THReshold[0to7/8to15] <Threshold>

Parameters

Threshold

Real in V

:DIGital:THReshold[0to7/8to15]?

Returns

Real in V

3.8 :TRIGger

This section covers the selection and configuration of the trigger.

:TRIGger:TYPE <Type>

Parameters

Type

EDGE
NEDGE
TIMEout
DELay
WINDow
PULSe
SLOPe
INTerval
SHOLd
RUNT
PATtern
DECode

:TRIGger:TYPE?

Returns

EDGE
NEDGE
TIMEout
DELay
WINDow
PULSe
SLOPe
INTerval
SHOLd
RUNT
PATtern
DECode

:TRIGger:AOUT <State>**Parameters****State****ON or 1****OFF or 0**

:TRIGger:AOUT?**Returns****ON****OFF**

3.9 :TRIGger:EDGE

This section covers the edge trigger.

:TRIGger:EDGE:SOURce <Source>

Parameters

Source

[Channel]

:TRIGger:EDGE:SOURce?

Returns

[Channel]

:TRIGger:EDGE:SLOPe <Slope>

Parameters

Slope

RISing

FALLing

BOTH

ALternate

:TRIGger:EDGE:SLOPe?

Returns

RISing

FALLing

BOTH

ALternate

:TRIGger:EDGE:LEVEl <Level>

Parameters

Level

Real in V or A

:TRIGger:EDGE:LEVEl?

Returns

Real in V or A

:TRIGger:EDGE:LHYSteresis <Hysteresis>**Parameters****Hysteresis****Real in %**

:TRIGger:EDGE:LHYSteresis?**Returns****Real in %**

3.10 :TRIGger:NEDGE

This section covers the nth-edge trigger.

:TRIGger:NEDGE:SOURce <Source>

Parameters

Source

[Channel]

:TRIGger:NEDGE:SOURce?

Returns

[Channel]

:TRIGger:NEDGE:SLOPe <Slope>

Parameters

Slope

RISing

FALLing

:TRIGger:NEDGE:SLOPe?

Returns

RISing

FALLing

:TRIGger:NEDGE:LEVel <Level>

Parameters

Level

Real in V or A

:TRIGger:NEDGE:LEVel?

Returns

Real in V or A

:TRIGger:NEDGE:IDLe <Idle>

Parameters

Idle

Idle in s

:TRIGger:NEDGe:IDLe?

Returns

Real in s

:TRIGger:NEDGe:COUNT <Count>

Parameters

Count

Integer

:TRIGger:NEDGe:COUNT?

Returns

Integer

:TRIGger:NEDGe:LHYSteresis <Hysteresis>

Parameters

Hysteresis

Real in %

:TRIGger:NEDGe:LHYSteresis?

Returns

Real in %

3.11 :TRIGger:TIMEout

This section covers the timeout trigger.

:TRIGger:TIMEout:SOURce <Source>

Parameters

Source

[Channel]

ACLIne is not valid

:TRIGger:TIMEout:SOURce?

Returns

[Channel]

:TRIGger:TIMEoutGe:SLOPe <Slope>

Parameters

Slope

RISing

FALLing

:TRIGger:TIMEout:SLOPe?

Returns

RISing

FALLing

:TRIGger:TIMEout:LEVel <Level>

Parameters

Level

Real in V or A

:TRIGger:TIMEout:LEVel?

Returns

Real in V or A

:TRIGger:TIMEout:TIME <Time>

Parameters

Time

Real in s

:TRIGger:TIMEout:TIME?

Returns

Real in s

:TRIGger:TIMEout:LHYSteresis <Hysteresis>

Parameters

Hysteresis

Real in %

:TRIGger:TIMEout:LHYSteresis?

Returns

Real in %

3.12 :TRIGger:DELay

This section covers the delay trigger.

:TRIGger:DELay:SOURce1 <Source>

Parameters

Source

[Channel]

:TRIGger:DELay:SOURce1?

Returns

[Channel]

:TRIGger:DELay:SLOPe1 <Slope>

Parameters

Slope

RISing

FALLing

:TRIGger:DELay:SLOPe1?

Returns

RISing

FALLing

:TRIGger:DELay:LEVeL1 <Level>

Parameters

Level

Real in V or A

:TRIGger:DELay:LEVeL1?

Returns

Real in V or A

:TRIGger:DELay:SOURce2 <Source>

Parameters

Source

[Channel]

:TRIGger:DElay:SOURce2?

Returns

[Channel]

:TRIGger:DElay:SLOPe2 <Slope>

Parameters

Slope

RISing

FALLing

:TRIGger:DElay:SLOPe2?

Returns

RISing

FALLing

:TRIGger:DElay:LEVel2 <Level>

Parameters

Level

Real in V or A

:TRIGger:DElay:LEVel2?

Returns

Real in V or A

:TRIGger:DElay:TYPE <Type>

Parameters

Type

LTHan

GTHan

INSide

OUTSide

:TRIGger:DELay:TYPE?

Returns

LTHan

GTHan

INSide

OUTSide

:TRIGger:DELay:THReshold <Threshold>

Parameters

Threshold

Real in s

:TRIGger:DELay:THReshold?

Returns

Real in s

:TRIGger:DELay:DURation:LOWer <Duration>

Parameters

Duration

Real in s

:TRIGger:DELay:DURation:LOWer?

Returns

Real in s

:TRIGger:DELay:DURation:UPPer <Duration>

Parameters

Duration

Real in s

:TRIGger:DELay:DURation:UPPer?

Returns

Real in s

:TRIGger:DELay:LHYSteresis <Hysteresis>**Parameters****Hysteresis****Real in %**

:TRIGger:DELay:LHYSteresis?**Returns****Real in %**

3.13 :TRIGger:WINDow

This section covers the window trigger.

:TRIGger:WINDow:SOURce <Source>

Parameters

Source

CHANnel[AnalogN]

:TRIGger:WINDow:SOURce?

Returns

CHANnel[AnalogN]

:TRIGger:WINDow:TYPE <Slope>

Parameters

Slope

ENTer

LEAVe

:TRIGger:WINDow:TYPE?

Returns

ENTer

LEAVe

:TRIGger:WINDow:LEVel1 <Level>

Parameters

Level

Real in V or A

:TRIGger:WINDow:LEVel1?

Returns

Real in V or A

:TRIGger:WINDow:LEVel2 <Level>

Parameters

Level

Real in V or A

:TRIGger:WINDow:LEVel2?

Returns

Real in V or A

:TRIGger:WINDow:TIMing <Timing>

Parameters

Timing

NONE

LTHan

GTHan

INSide

OUTSide

:TRIGger:WINDow:TIMing?

Returns

NONE

LTHan

GTHan

INSide

OUTSide

:TRIGger:WINDow:THReshold <Threshold>

Parameters

Threshold

Real in s

:TRIGger:WINDow:THReshold?

Returns

Real in s

:TRIGger:WINDow:DURation:LOWer <Duration>

Parameters

Duration

Real in s

:TRIGger:WINDow:DURation:LOWer?

Returns

Real in s

:TRIGger:WINDow:DURation:UPPer <Duration>

Parameters

Duration

Real in s

:TRIGger:WINDow:DURation:UPPer?

Returns

Real in s

:TRIGger:WINDow:LHYSteresis <Hysteresis>

Parameters

Hysteresis

Real in %

:TRIGger:WINDow:LHYSteresis?

Returns

Real in %

3.14 :TRIGger:PULSe

This section covers the pulse trigger.

:TRIGger:PULSe:SOURce <Source>

Parameters

Source

[Channel]

:TRIGger:PULSe:SOURce?

Returns

[Channel]

:TRIGger:PULSe:POLarity <Polarity>

Parameters

Polarity

POSitive

NEGative

:TRIGger:PULSe:POLarity?

Returns

POSitive

NEGative

:TRIGger:PULSe:LEVel <Level>

Parameters

Level

Real in V or A

:TRIGger:PULSe:LEVel?

Returns

Real in V or A

:TRIGger:PULSe:TIMing <Timing>

Parameters

Timing

LTHan

GTHan

INSide

OUTSide

:TRIGger:PULSe:TIMing?

Returns

LTHan

GTHan

INSide

OUTSide

:TRIGger:PULSe:THReshold <Threshold>

Parameters

Threshold

Real in s

:TRIGger:PULSe:THReshold?

Returns

Real in s

:TRIGger:PULSe:DURation:LOWer <Duration>

Parameters

Duration

Real in s

:TRIGger:PULSe:DURation:LOWer?

Returns

Real in s

:TRIGger:PULSe:DURation:UPPer <Duration>

Parameters

Duration

Real in s

:TRIGger:PULSe:DURation:UPPer?

Returns

Real in s

:TRIGger:PULSe:LHYSteresis <Hysteresis>

Parameters

Hysteresis

Real in %

:TRIGger:PULSe:LHYSteresis?

Returns

Real in %

3.15 :TRIGger:SLOPe

This section covers the slope trigger.

:TRIGger:SLOPe:SOURce <Source>

Parameters

Source

CHANnel[AnalogN]

:TRIGger:SLOPe:SOURce?

Returns

CHANnel[AnalogN]

:TRIGger:SLOPe:TYPe <Slope>

Parameters

Slope

RISing

FALLing

:TRIGger:SLOPe:TYPe?

Returns

RISing

FALLing

:TRIGger:SLOPe:LEVel1 <Level>

Parameters

Level

Real in V or A

:TRIGger:SLOPe:LEVel1?

Returns

Real in V or A

:TRIGger:SLOPe:LEVel2 <Level>

Parameters

Level

Real in V or A

:TRIGger:SLOPe:LEVel2?

Returns

Real in V or A

:TRIGger:SLOPe:TIMing <Timing>

Parameters

Timing

LTHan

GTHan

INSide

OUTSide

:TRIGger:SLOPe:TIMing?

Returns

LTHan

GTHan

INSide

OUTSide

:TRIGger:SLOPe:THReshold <Threshold>

Parameters

Threshold

Real in s

:TRIGger:SLOPe:THReshold?

Returns

Real in s

:TRIGger:SLOPe:DURation:LOWer <Duration>

Parameters

Duration

Real in s

:TRIGger:SLOPe:DURation:LOWer?

Returns

Real in s

:TRIGger:SLOPe:DURation:UPPer <Duration>

Parameters

Duration

Real in s

:TRIGger:SLOPe:DURation:UPPer?

Returns

Real in s

:TRIGger:SLOPe:LHYSteresis <Hysteresis>

Parameters

Hysteresis

Real in %

:TRIGger:SLOPe:LHYSteresis?

Returns

Real in %

3.16 :TRIGger:INTerval

This section covers the interval trigger.

:TRIGger:INTerval:SOURce <Source>

Parameters

Source

[Channel]

:TRIGger:INTerval:SOURce?

Returns

[Channel]

:TRIGger:INTerval:POLarity <Polarity>

Parameters

Polarity

POSitive

NEGative

:TRIGger:INTerval:POLarity?

Returns

POSitive

NEGative

:TRIGger:INTerval:LEVel <Level>

Parameters

Level

Real in V or A

:TRIGger:INTerval:LEVel?

Returns

Real in V or A

:TRIGger:INTerval:TIMing <Timing>

Parameters

Timing

LTHan

GTHan

INSide

OUTSide

:TRIGger:INTerval:TIMing?

Returns

LTHan

GTHan

INSide

OUTSide

:TRIGger:INTerval:THReshold <Threshold>

Parameters

Threshold

Real in s

:TRIGger:INTerval:THReshold?

Returns

Real in s

:TRIGger:INTerval:DURation:LOWer <Duration>

Parameters

Duration

Real in s

:TRIGger:INTerval:DURation:LOWer?

Returns

Real in s

:TRIGger:INTerval:DURation:UPPer <Duration>

Parameters

Duration

Real in s

:TRIGger:INTerval:DURation:UPPer?

Returns

Real in s

:TRIGger:INTerval:LHYSteresis <Hysteresis>

Parameters

Hysteresis

Real in %

:TRIGger:INTerval:LHYSteresis?

Returns

Real in %

3.17 :TRIGger:SHOLd

This section covers the setup-hold trigger.

:TRIGger:SHOLd:CLOCK:SOURce <Source>

Parameters

Source

[Channel]

:TRIGger:SHOLd:CLOCK:SOURce?

Returns

[Channel]

:TRIGger:SHOLd:CLOCK:SLOPe <Slope>

Parameters

Slope

RISing

FALLing

:TRIGger:SHOLd:CLOCK:SLOPe?

Returns

RISing

FALLing

:TRIGger:SHOLd:CLOCK:LEVeL <Level>

Parameters

Level

Real in V or A

:TRIGger:SHOLd:CLOCK:LEVeL?

Returns

Real in V or A

:TRIGger:SHOLd:CLOCK:HYSTeresis <Hysteresis>

Parameters

Hysteresis

Real in %

:TRIGger:SHOLd:CLOCK:HYSteresis?

Returns

Real in %

:TRIGger:SHOLd:DATA:SOURce <Source>

Parameters

Source

[Channel]

:TRIGger:SHOLd:DATA:SOURce?

Returns

[Channel]

:TRIGger:SHOLd:DATA:LEVel <Level>

Parameters

Level

Real in V or A

:TRIGger:SHOLd:DATA:LEVel?

Returns

Real in V or A

:TRIGger:SHOLd:DATA:HYSteresis <Hysteresis>

Parameters

Hysteresis

Real in %

:TRIGger:SHOLd:DATA:HYSteresis?

Returns

Real in %

:TRIGger:SHOLd:TYPE <Type>

Parameters

Type

HOLD

SETup

SHOLd

:TRIGger:SHOLd:TYPe?

Returns

HOLD

SETup

SHOLd

:TRIGger:SHOLd:STIME <Time>

Parameters

Time

Real in s

:TRIGger:SHOLd:STIME?

Returns

Real in s

:TRIGger:SHOLd:HTIME <Time>

Parameters

Time

Real in s

:TRIGger:SHOLd:HTIME?

Returns

Real in s

3.18 :TRIGger:RUNT

This section covers the runt trigger.

:TRIGger:RUNT:SOURce <Source>

Parameters

Source

CHANnel[AnalogN]

:TRIGger:RUNT:SOURce?

Returns

CHANnel[AnalogN]

:TRIGger:RUNT:POLarity <Polarity>

Parameters

Polarity

POSitive

NEGative

:TRIGger:RUNT:POLarity?

Returns

POSitive

NEGative

:TRIGger:RUNT:LEVEl1 <Level>

Parameters

Level

Real in V or A

:TRIGger:RUNT:LEVEl1?

Returns

Real in V or A

:TRIGger:RUNT:LEVEl2 <Level>

Parameters

Level

Real in V or A

:TRIGger:RUNT:LEVel2?

Returns

Real in V or A

:TRIGger:RUNT:TIMing <Timing>

Parameters

Timing

NONe

LTHan

GTHan

INSide

OUTSide

:TRIGger:RUNT:TIMing?

Returns

NONe

LTHan

GTHan

INSide

OUTSide

:TRIGger:RUNT:THReshold <Threshold>

Parameters

Threshold

Real in s

:TRIGger:RUNT:THReshold?

Returns

Real in s

:TRIGger:RUNT:DURation:LOWer <Duration>

Parameters

Duration

Real in s

:TRIGger:RUNT:DURation:LOWer?

Returns

Real in s

:TRIGger:RUNT:DURation:UPPer <Duration>

Parameters

Duration

Real in s

:TRIGger:RUNT:DURation:UPPer?

Returns

Real in s

:TRIGger:RUNT:LHYSteresis <Hysteresis>

Parameters

Hysteresis

Real in %

:TRIGger:RUNT:LHYSteresis?

Returns

Real in %

3.19 :TRIGger:PATtern

This section covers the pattern trigger.

:TRIGger:PATtern:TYPE <Timing>

Parameters

Timing

AND

OR

NAND

NOR

:TRIGger:PATtern:TYPE?

Returns

ND

OR

NAND

NOR

:TRIGger:PATtern:TIMing <Timing>

Parameters

Timing

NOne

LTHan

GTHan

INSide

OUTSide

:TRIGger:PATtern:TIMing?

Returns

NOne

LTHan

GTHan

INSide

OUTSide

:TRIGger:PATtern:THReshold <Threshold>

Parameters

Threshold

Real in s

:TRIGger:PATtern:THReshold?

Returns

Real in s

:TRIGger:PATtern:DURation:LOWer <Duration>

Parameters

Duration

Real in s

:TRIGger:PATtern:DURation:LOWer?

Returns

Real in s

:TRIGger:PATtern:DURation:UPPer <Duration>

Parameters

Duration

Real in s

:TRIGger:PATtern:DURation:UPPer?

Returns

Real in s

:TRIGger:PATtern:[Channel]:CLOCK <State>

Parameters

State

ON or 1

OFF or 0

:TRIGger:PATtern:[Channel]:CLOCK?

Returns

ON

OFF

:TRIGger:PATtern:[Channel]:PATtern <Pattern>

Parameters

Pattern

IGNore

HIGH

LOW

:TRIGger:PATtern:[Channel]:PATtern?

Returns

IGNore

HIGH

LOW

:TRIGger:PATtern:[Channel]:THReshold:LOWer <Duration>

Parameters

Duration

Real in V or A

:TRIGger:PATtern:[Channel]:THReshold:LOWer?

Returns

Real in V or A

:TRIGger:PATtern:[Channel]:THReshold:UPPer <Duration>

Parameters

Duration

Real in V or A

:TRIGger:PATtern:[Channel]:THReshold:UPPer?

Returns

Real in V or A

:TRIGger:PATtern:CLOCk:SOURce <Source>

Parameters

Source

[Channel]

CLine is not valid

:TRIGger:PATtern:CLOCK:SOURce?

Returns

[Channel]

:TRIGger:PATtern:CLOCK:SLOPe <Slope>

Parameters

Slope

RISing

FALLing

:TRIGger:PATtern:CLOCK:SLOPe?

Returns

RISing

FALLing

:TRIGger:PATtern:CLOCK:LEVel <Level>

Parameters

Level

Real in V or A

:TRIGger:PATtern:CLOCK:LEVel?

Returns

Real in V or A

:TRIGger:PATtern:CLOCK:HYSTeresis <Hysteresis>

Parameters

Hysteresis

Real in %

:TRIGger:PATtern:CLOCK:HYSTeresis?

Returns

Real in %

3.20 :TRIGger:DECode

This section covers the selection of the decode group as trigger source and the configuration of bus-specific trigger settings.

:TRIGger:DECode:GR0up <GR0up>

Parameters

GR0up

DECode[DecodeN]

:TRIGger:DECode:GR0up?

Returns

DECode[DecodeN]

3.21 :TRIGger:DECode:I2C

This section covers the I2C decode trigger.

- [ByteN] has to be replaced with a byte index, for example: BYTE4

:TRIGger:DECode:I2C:EVENT <Event>

Parameters

Event

START
STOP
REStart
ACK
NACK
ADDRESS
DATA
ADATA

:TRIGger:DECode:I2C:EVENT?

Returns

START
STOP
REStart
ACK
NACK
ADDRESS
DATA
ADATA

:TRIGger:DECode:I2C:ADDRESS:NFORMAT <Format>

Parameters

Format

DEC
HEX
BIN
ASCII

:TRIGger:DECode:I2C:ADDRess:NFORmat?

Returns

DEC
HEX
BIN
ASCii

:TRIGger:DECode:I2C:ADDRess:LENGth <Length>

Parameters

Length

AUTo
7
10

:TRIGger:DECode:I2C:ADDRess:LENGth?

Returns

AUTo
7
10

:TRIGger:DECode:I2C:ADDRess:DIRectioN <Direction>

Parameters

Direction

READ
WRITe
RWRite

:TRIGger:DECode:I2C:ADDRess:DIRectioN?

Returns

READ
WRITe
RWRite

:TRIGger:DECode:I2C:ADDRess:WORD <Word>

Parameters

Word

Integer/String

In :TRIGger:DECode:I2C:ADDRess:NFORmat

:TRIGger:DECode:I2C:ADDRess:WORD?

Returns

Integer/String

In :TRIGger:DECode:I2C:ADDRess:NFORmat

:TRIGger:DECode:I2C:DATA:NFORmat <Format>

Parameters

Format**DEC****HEX****BIN****ASCii**

:TRIGger:DECode:I2C:DATA:NFORmat?

Returns

DEC**HEX****BIN****ASCii**

:TRIGger:DECode:I2C:DATA:POStion <Position>

Parameters

Position**OFFSet****ANYWhere**

:TRIGger:DECode:I2C:DATA:POStion?

Returns

OFFSet**ANYWhere**

:TRIGger:DECode:I2C:DATA:OFFSet <Offset>

Parameters

Offset**Integer in bytes**

:TRIGger:DECode:I2C:DATA:OFFSet?

Returns

Integer in bytes

:TRIGger:DECode:I2C:DATA:LENGth <Length>

Parameters

Length

Integer in bytes

:TRIGger:DECode:I2C:DATA:LENGth?

Returns

Integer in bytes

:TRIGger:DECode:I2C:DATA:BYTE[ByteN] <Byte>

Parameters

Byte

Integer/String

In :TRIGger:DECode:I2C:DATA:NFormat

:TRIGger:DECode:I2C:DATA:BYTE[ByteN]?

Returns

Integer/String

In :TRIGger:DECode:I2C:DATA:NFormat

3.22 :TRIGger:DECode:SPI

This section covers the SPI decode trigger.

- [WordN] has to be replaced with a word index, for example: WORD1

:TRIGger:DECode:SPI:LANE <Lane>

Parameters

Lane

MOSi

MISo

:TRIGger:DECode:SPI:LANE?

Returns

MOSi

MISo

:TRIGger:DECode:SPI:ALIGnment <Alignment>

Parameters

Alignment

BIT

WORD

:TRIGger:DECode:SPI:ALIGnment?

Returns

BIT

WORD

:TRIGger:DECode:SPI:DATA:POSition <Position>

Parameters

Position

OFFSet

ANYWhere

:TRIGger:DECode:SPI:DATA:POSition?

Returns

OFFSet

ANYWhere

:TRIGger:DECode:SPI:DATA:NFORmat <Format>**Parameters****Format****DEC****HEX****BIN****ASCii**

:TRIGger:DECode:SPI:DATA:NFORmat?**Returns****DEC****HEX****BIN****ASCii**

:TRIGger:DECode:SPI:DATA:WORDs:OFFSet <Offset>**Parameters****Offset****Integer in words**

:TRIGger:DECode:SPI:DATA:WORDs:OFFSet?**Returns****Integer in words**

:TRIGger:DECode:SPI:DATA:WORDs:LENGth <Length>**Parameters****Length****Integer in words**

:TRIGger:DECode:SPI:DATA:WORDs:LENGth?**Returns****Integer in words**

:TRIGger:DECode:SPI:DATA:WORD[WordN] <Word>**Parameters****Word****Integer/String**

In :TRIGger:DECode:SPI:DATA:NFORmat

:TRIGger:DECode:SPI:DATA:WORD[WordN]?

Returns

Integer/String

In :TRIGger:DECode:SPI:DATA:NFormat

:TRIGger:DECode:SPI:DATA:BITS:OFFSet <Offset>

Parameters

Offset**Integer in bits**

:TRIGger:DECode:SPI:DATA:BITS:OFFSet?

Returns

Integer in bits

:TRIGger:DECode:SPI:DATA:BITS:LENGth <Length>

Parameters

Length**Integer in bits**

:TRIGger:DECode:SPI:DATA:BITS:LENGth?

Returns

Integer in bits

:TRIGger:DECode:SPI:DATA:BITS <Bits>

Parameters

Bits**Integer/String**

In :TRIGger:DECode:SPI:DATA:NFormat

:TRIGger:DECode:SPI:DATA:BITS?

Returns

Integer/String

In :TRIGger:DECode:SPI:DATA:NFormat

3.23 :TRIGger:DECode:UART

This section covers the UART decode trigger.

- [WordN] has to be replaced with a word index, for example: WORD3

:TRIGger:DECode:UART:EVENT <Event>

Parameters

Event

FStart
IFStart
FPCheck
IFCompletion
DATA

:TRIGger:DECode:UART:EVENT?

Returns

FStart
IFStart
FPCheck
IFCompletion
DATA

:TRIGger:DECode:UART:DATA:POSition <Position>

Parameters

Position

OFFSet
ANYWhere

:TRIGger:DECode:UART:DATA:POSition?

Returns

OFFSet
ANYWhere

:TRIGger:DECode:UART:DATA:OFFSet <Offset>

Parameters

Offset

Integer in words

:TRIGger:DECode:UART:DATA:OFFSet?

Returns

Integer in words

:TRIGger:DECode:UART:DATA:LENGth <Length>

Parameters

Length

Integer in words

:TRIGger:DECode:UART:DATA:LENGth?

Returns

Integer in words

:TRIGger:DECode:UART:DATA:PARity <Parity>

Parameters

Parity

IGNore

MATCh

MISMatch

:TRIGger:DECode:UART:DATA:PARity?

Returns

IGNore

MATCh

MISMatch

:TRIGger:DECode:UART:DATA:NFORmat <Format>

Parameters

Format

DEC

HEX

BIN

ASCii

:TRIGger:DECode:UART:DATA:NFORmat?

Returns

DEC

HEX

BIN

ASCIi

:TRIGger:DECode:UART:DATA:WORD[WordN] <Word>

Parameters

Word

Integer/String

In :TRIGger:DECode:UART:DATA:NFORmat

:TRIGger:DECode:UART:DATA:WORD[WordN]?

Returns

Integer/String

In :TRIGger:DECode:UART:DATA:NFORmat

3.24 :TRIGger:DECode:PARAllel

This section covers the parallel decode trigger.

- [WordN] has to be replaced with a word index, for example: WORD2

:TRIGger:DECode:PARAllel:NFORmat <Format>

Parameters

Format

DEC
HEX
BIN
AScii

:TRIGger:DECode:PARAllel:NFORmat?

Returns

DEC
HEX
BIN
AScii

:TRIGger:DECode:PARAllel:WORD[WordN] <Word>

Parameters

Word

Integer/String

In :TRIGger:DECode:PARAllel:NFORmat

:TRIGger:DECode:PARAllel:WORD[WordN]?

Returns

Integer/String

In :TRIGger:DECode:PARAllel:NFORmat

3.25 :MEASurement

This section covers the general measurement configuration, as well as adding, removing and configuring specific measurements. Measurement statistics can also be queried.

- [AnalogN] has to be replaced with an analog channel index, for example: CHANne11
- [Channel] has to be replaced with an analog or digital channel, for example: CHANne11 or DIGi-tal7
- [Type] has to be replaced with one of the measurements at the end of this section, see 3.25.2.

:MEASurement:STATE <State>

Parameters

State

ON or 1

OFF or 0

:MEASurement:STATE?

Returns

ON

OFF

:MEASurement:DISPlay <Display>

Parameters

Display

COMPact

DETAiled

:MEASurement:DISPlay?

Returns

COMPact

DETAiled

:MEASurement:GATE <Gate>

Parameters

Gate

SCReen

AREa

:MEASurement:GATE?

Returns

SCReen

AREa

:MEASurement:ATYPE <Type>

Parameters

Type

FIXed

RUNNing

:MEASurement:ATYPE?

Returns

FIXed

RUNNing

:MEASurement:AVERage <Average>

Parameters

Average

Integer

:MEASurement:AVERage?

Returns

Integer

:MEASurement:RLEVels:CHANnel[AnalogN]:BASE <Level>

Parameters

Level

VPP

VAMP

MANual

:MEASurement:RLEVels:CHANnel[AnalogN]:BASE?

Returns

VPP

VAMP

MANual

:MEASurement:RLEVels:CHANnel[AnalogN]:VPP:LOWer <Level>

Parameters

Level

Real in %

:MEASurement:RLEVels:CHANnel[AnalogN]:VPP:LOWer?

Returns

Real in %

:MEASurement:RLEVels:CHANnel[AnalogN]:VPP:MID <Level>

Parameters

Level

Real in %

:MEASurement:RLEVels:CHANnel[AnalogN]:VPP:MID?

Returns

Real in %

:MEASurement:RLEVels:CHANnel[AnalogN]:VPP:UPPer <Level>

Parameters

Level

Real in %

:MEASurement:RLEVels:CHANnel[AnalogN]:VPP:UPPer?

Returns

Real in %

:MEASurement:RLEVels:CHANnel[AnalogN]:VAMPP:LOWer <Level>

Parameters

Level

Real in %

:MEASurement:RLEVels:CHANnel[AnalogN]:VAMP:LOWer?

Returns

Real in %

:MEASurement:RLEVels:CHANnel[AnalogN]:VAMP:MID <Level>

Parameters

Level

Real in %

:MEASurement:RLEvels:CHANnel[AnalogN]:VAMP:MID?

Returns

Real in %

:MEASurement:RLEvels:CHANnel[AnalogN]:VAMP:UPPer <Level>

Parameters

Level

Real in %

:MEASurement:RLEvels:CHANnel[AnalogN]:VAMP:UPPer?

Returns

Real in %

:MEASurement:RLEvels:CHANnel[AnalogN]:MANual:LOWer <Level>

Parameters

Level

Real in %

:MEASurement:RLEvels:CHANnel[AnalogN]:MANual:LOWer?

Returns

Real in V

:MEASurement:RLEvels:CHANnel[AnalogN]:MANual:MID <Level>

Parameters

Level

Real in V

:MEASurement:RLEvels:CHANnel[AnalogN]:MANual:MID?

Returns

Real in V

:MEASurement:RLEvels:CHANnel[AnalogN]:MANual:UPPer <Level>

Parameters

Level

Real in V

:MEASurement:RLEvels:CHANnel[AnalogN]:MANual:UPPer?

Returns

Real in V

:MEASurement:CLEar

:MEASurement:[Type]:ALL? <Type>

Parameters

Type

[Channel]

[Channel],[Channel]

For Delay and Phase

Returns

Packed statistics

See 3.25.1.

:MEASurement:[Type]:CURRent? <Type>

Parameters

Type

[Channel]

[Channel],[Channel]

For Delay and Phase

:MEASurement:[Type]:AVERAge? <Type>

Parameters

Type

[Channel]

[Channel],[Channel]

For Delay and Phase

:MEASurement:[Type]:MAXimum? <Type>

Parameters

Type

[Channel]

[Channel],[Channel]

For Delay and Phase

:MEASurement:[Type]:MINimum? <Type>

Parameters

Type

[Channel]

[Channel],[Channel]

For Delay and Phase

:MEASurement: [Type]:DEViation? <Type>

Parameters

Type

[Channel]

[Channel], [Channel]

For Delay and Phase

:MEASurement: [Type]:COUNT? <Type>

Parameters

Type

[Channel]

[Channel], [Channel]

For Delay and Phase

:MEASurement: [Type]:ADD

:MEASurement: [Type]:REMove

3.25.1 All measurement statistics

All data in this section has a *Block Data* header (see 1.1.3).

The measurement statistics

Field	Count (bytes)	Data Type	Json/XML Name
Current	8	8-byte floating-point	Current
Average	8	8-byte floating-point	Average
Maximum	8	8-byte floating-point	Maximum
Minimum	8	8-byte floating-point	Minimum
Deviation	8	8-byte floating-point	Deviation
Count	4	4-byte integer	Count

3.25.2 Measurement Types

The available measurement types.

- **VMIN** Vmin
- **VMAX** Vmax
- **VPP** Vpp
- **VBASE** Vbase
- **VTOP** Vtop
- **VAMP** Vamp
- **VLOWER** Vlower
- **VUPPER** Vupper

• VMID	Vmid
• VAVG	Vavg
• VRMS	Vrms
• VSDeviation	Std. Dev
• VPPReshoot	+Preshoot
• VNPPReshoot	-Preshoot
• VPOVershoot	+Overshoot
• VNOVershoot	-Overshoot
• VPAREa	+Area
• VNAREa	-Area
• VAARea	Area
• VNAAREa	Area
• VPPAREa	+Per. Area
• VNPAREa	-Per. Area
• VAPAREa	Per. Area
• VNAPAREa	Per. Area
• HRTIME	Trise
• HFTIME	Tfall
• HPERIOD	Period
• HFREQ	Freq
• HHWIDTH	+Width
• HLWIDTH	-Width
• HHDUTY	+Duty
• HLDUTY	-Duty
• RECOUNT	+Edge#
• FECOUNT	-Edge#
• PPCOUNT	+Pulse#
• NPCOUNT	-Pulse#
• HDELAY: [RISing/FALLing] : [RISing/FALLing] : [NEARest/FORward/BACKward] [+/-][+/-]Delay[[None]/>/<]	
• HPHase: [RISing/FALLing] : [RISing/FALLing] : [NEARest/FORward/BACKward] [+/-][+/-]Phase[[None]/>/<]	

3.26 :REference

This section covers adding, removing and configuring references.

- [N] has to be replaced with a reference index, for example: REference2
- References are automatically created when a setting is changed for a non-existent channel.

:REference:STATe <State>

Parameters

State

ON or 1

OFF or 0

:REference:STATe?

Returns

ON

OFF

:REference:CLEAr

:REference[N]:STATe <State>

Parameters

State

ON or 1

OFF or 0

:REference[N]:STATe?

Returns

ON

OFF

:REference[N]:SOURce <Source>

Parameters

Source

CHANnel[AnalogN]

:REference[N]:SOURce?

Returns

CHANnel[AnalogN]

:REFeRence[N]:OFFSet <Offset>

Parameters

Offset

Real in V or A

:REFeRence[N]:OFFSet?

Returns

Real in V or A

:REFeRence[N]:SCALE <Scale>

Parameters

Scale

Real in V or A

:REFeRence[N]:SCALE?

Returns

Real in V or A

:REFeRence[N]:REFResh

:REFeRence[N]:ADD

:REFeRence[N]:REMOve

3.27 :MATH

This section covers adding, removing and configuring math channels.

- [N] has to be replaced with a math channel index, for example: MATH1
- Math channels are automatically added when a setting is changed for a non-existent channel.

:MATH:STATe <State>

Parameters

State

ON or 1

OFF or 0

:MATH:STATe?

Returns

ON

OFF

:MATH:CLEAr

:MATH[N]:STATe <State>

Parameters

State

ON or 1

OFF or 0

:MATH[N]:STATe?

Returns

ON

OFF

:MATH[N]:UNIT <Unit>

Parameters

Unit

String

:MATH[N]:UNIT?

Returns

String

:MATH[N]:EXPRession <Expression>

Parameters

Expression**String**

Mathematical expression

:MATH[N]:EXPRession?

Returns

String

Mathematical expression

:MATH[N]:OFFSet <Offset>

Parameters

Offset**Real in any unit**

:MATH[N]:OFFSet?

Returns

Real

:MATH[N]:SCALE <Scale>

Parameters

Scale**Real in any unit**

:MATH[N]:SCALE?

Returns

Real

:MATH[N]:ADD

:MATH[N]:REMOve

3.28 :FFT

This section covers the configuration of general FFT settings and FFT channels.

- [N] has to be replaced with an FFT channel index, for example: FFT2
- [AnaLogN] has to be replaced with an analog channel index, for example: CHANneL1

:FFT:STATE <State>

Parameters

State

ON or 1

OFF or 0

:FFT:STATE?

Returns

ON

OFF

:FFT:SCALE <Scale>

Parameters

Scale

DBM

DBV

DBMV

DBUV

V

:FFT:SCALE?

Returns

DBM

DBV

DBMV

DBUV

V

:FFT:FREQUENCY:CENTER <Center>

Parameters

Center

Real in Hz

:FFT:FREQuency:CENTer?

Returns

Real in Hz

**:FFT:FREQuency:SPAN **

Parameters

Span

Real in Hz

:FFT:FREQuency:SPAN?

Returns

Real in Hz

:FFT:FREQuency:START <Start>

Parameters

Start

Real in Hz

:FFT:FREQuency:START?

Returns

Real in Hz

:FFT:FREQuency:STOP <Stop>

Parameters

Stop

Real in Hz

:FFT:FREQuency:STOP?

Returns

Real in Hz

:FFT[N]:STATe <State>

Parameters

State

ON or 1

OFF or 0

:FFT[N]:STATE?

Returns

ON

OFF

:FFT[N]:SOURCE <Source>

Parameters

Source

CHANnel[AnalogN]

:FFT[N]:SOURCE?

Returns

CHANnel[AnalogN]

:FFT[N]:WINDOW <Window>

Parameters

Window

RECTangle

HANN

HAMming

BLACKman

FLATtop

GAUSSian

KBESsel

:FFT[N]:WINDOW?

Returns

RECTangle

HANN

HAMming

BLACKman

FLATtop

GAUSSian

KBESsel

:FFT[N]:TRACe <Trace>**Parameters****Trace****SPECTrum****AVERAge****MINHold****MAXHold**

:FFT[N]:TRACe?**Returns****SPECTrum****AVERAge****MINHold****MAXHold**

:FFT[N]:AVERAge <Average>**Parameters****Average****Integer**

:FFT[N]:AVERAge?**Returns****Integer**

:FFT[N]:OFFSet <Offset>**Parameters****Offset****Real in unit**

See :FFT:SCALE for units

:FFT[N]:OFFSet?**Returns****Real in unit**

See :FFT:SCALE for units

:FFT[N]:SCALE <Scale>**Parameters****Scale****Real in unit**

See :FFT:SCALE for units

:FFT[N]:SCALE?**Returns****Real in unit**

See :FFT:SCALE for units

3.29 :FFT[N]:DATA

This section focuses on the transfer of FFT bins.

- [N] has to be replaced with an FFT channel index, for example: FFT2

:FFT[N]:DATA:SCALE <Scale>

Parameters

Scale

DBM
DBV
DBMV
DBUV
V

:FFT[N]:DATA:SCALE?

Returns

DBM
DBV
DBMV
DBUV
V

:FFT[N]:DATA:BFrequency?

Returns

Real in Hz

:FFT[N]:DATA:SFrequency?

Returns

Real in Hz

:FFT[N]:DATA:BINS

See :FFT[N]:DATA:SCALE for parameterization.

Returns

Bin data

See 3.29.1. The metadata can be queried from :FFT[N]:DATA:BFrequency and :FFT[N]:DATA:SFrequency.

:CHANnel[N]:DATA:PACKed? <Scale>

:FFT[N]:DATA:SCALE is ignored and specified through the parameter instead.

Parameters

Scale

- DBM Default
- DBV
- DBMV
- DBUV
- V

Returns

- Packed bin data** See 3.29.2.

3.29.1 FFT Data Bins

All data in this section has a *Block Data* header (see 1.1.3).

The FFT bins N is implicitly defined through the *Block Data* header

Field	Count (bytes)	Data Type	Json/XML Name
Bins	4 * N	4-byte floating-point	Bins

3.29.2 FFT Data Packed

All data in this section has a *Block Data* header (see 1.1.3).

The FFT bins including a metadata header

Field	Count (bytes)	Data Type	Json/XML Name
The bin frequency	4	4-byte floating-point	BinFrequency
Frequency of the last bin	4	4-byte floating-point	StopFrequency
N = Number of bins	4	4-byte unsigned integer	BinCount
Bins	4 * N	4-byte floating-point	Bins

3.30 :FFT:PTABLe

This section covers the configuration of the FFT peak table.

- [N] has to be replaced with an FFT channel index, for example: FFT3
- [PeakN] has to be replaced with a peak table index, for example: PTABLe1

:FFT:PTABLe:STATE <State>

Parameters

State

ON or 1

OFF or 0

:FFT:PTABLe:STATE?

Returns

ON

OFF

:FFT:PTABLe[PeakN]:STATE <State>

Parameters

State

ON or 1

OFF or 0

:FFT:PTABLe[PeakN]:STATE?

Returns

ON

OFF

:FFT:PTABLe[PeakN]:SOURCE <Source>

Parameters

Source

FFT[N]

:FFT:PTABLe[PeakN]:SOURCE?

Returns

FFT[N]

:FFT:PTABLE[PeakN]:FRANge <Range>

Parameters

Range

COMPLete

SPAN

:FFT:PTABLE[PeakN]:FRANge?

Returns

COMPLete

SPAN

:FFT:PTABLE[PeakN]:TYPE <Type>

Parameters

Type

MINimum

MAXimum

:FFT:PTABLE[PeakN]:TYPE?

Returns

MINimum

MAXimum

:FFT[N]:PTABLE[PeakN]:EXCURsion <Excursion>

Parameters

Excursion

Real in unit

See :FFT:SCALE for units

:FFT[N]:PTABLE[PeakN]:EXCURsion?

Returns

Real in unit

See :FFT:SCALE for units

:FFT[N]:PTABLE[PeakN]:THRESHold <Threshold>

Parameters

Threshold

Real in unit

See :FFT:SCALE for units

:FFT[N]:PTABLE[PeakN]:THReshold?

Returns

Real in unit

See :FFT:SCALE for units

:FFT[N]:PTABLE[PeakN]:PEAKs

Returns

Peak data

See 3.30.1.

3.30.1 FFT Peak Data

All data in this section has a *Block Data* header (see 1.1.3).

The FFT peaks N is implicitly defined through the *Block Data* header

Field	Count (bytes)	Data Type	Json/XML Name
Peaks (Frequency / Amplitude)	$2 * 8 * N$	8-byte floating-point / 8-byte floating-point	Frequency / Amplitude

3.31 :FFT:SPECTrogram

This section covers the configuration of the FFT spectrogram.

- [N] has to be replaced with an FFT channel index, for example: FFT4

:FFT:SPECTrogram:STATE <State>

Parameters

State

ON or 1

OFF or 0

:FFT:SPECTrogram:STATE?

Returns

ON

OFF

:FFT:SPECTrogram:SOURce <Source>

Parameters

Source

FFT[N]

:FFT:SPECTrogram:SOURce?

Returns

FFT[N]

:FFT:SPECTrogram:THReshoId:LOWer <Threshold>

Parameters

Threshold

Real in unit

See :FFT:SCALE for units

:FFT:SPECTrogram:THReshoId:LOWer?

Returns

Real in unit

See :FFT:SCALE for units

:FFT:SPECTrogram:THReshoId:UPPer <Threshold>

Parameters

Threshold

Real in unit

See :FFT:SCALE for units

:FFT:SPECTrogram:THReshoId:UPPer?**Returns****Real in unit**

See :FFT:SCALe for units

:FFT:SPECTrogram:CGRading <Grading>**Parameters****Grading****HEATmap7****HEATmap5****GRADient2**

:FFT:SPECTrogram:CGRading?**Returns****HEATmap7****HEATmap5****GRADient2**

3.32 :FFT:CURSOR

This section covers the configuration of the FFT cursor settings.

- [N] has to be replaced with an FFT channel index, for example: FFT2

:FFT:CURSOR:STATE <State>

Parameters

State

ON or 1

OFF or 0

:FFT:CURSOR:STATE?

Returns

ON

OFF

:FFT:CURSOR:SOURCE <Source>

Parameters

Source

FFT[N]

:FFT:CURSOR:SOURCE?

Returns

FFT[N]

:FFT:CURSOR:TRACKING <Tracking>

Parameters

Tracking

ON or 1

OFF or 0

:FFT:CURSOR:TRACKING?

Returns

ON

OFF

:FFT:CURSor:[X/Y]:STATE <State>

Parameters

State

ON or 1

OFF or 0

:FFT:CURSor:[X/Y]:STATE?

Returns

ON

OFF

:FFT:CURSor:[X/Y]:COUPling <Coupling>

Parameters

Coupling

ON or 1

OFF or 0

:FFT:CURSor:[X/Y]:COUPling?

Returns

ON

OFF

:FFT:CURSor:[A/B]:STATE <State>

Parameters

State

ON or 1

OFF or 0

:FFT:CURSor:[A/B]:STATE?

Returns

ON

OFF

:FFT:CURSor:[A/B]:[X/Y]:STATE <State>

Parameters

State

ON or 1

OFF or 0

:FFT:CURSor:[A/B]:[X/Y]:STATe?

Returns

ON

OFF

:FFT:CURSor:[A/B]:[X/Y]:POStion <Position>

Parameters

Position

Real in unit

See :FFT:SCALe for units

:FFT:CURSor:[A/B]:[X/Y]:POStion?

Returns

Real in unit

See :FFT:SCALe for units

3.33 :FFT:MARKer

This section covers the configuration of the FFT markers.

- [N] has to be replaced with an FFT channel index, for example: FFT3
- [MarkerN] has to be replaced with a marker index, for example: MARKer4
- Markers are automatically added when a setting is changed for a non-existent marker.

:FFT:MARKer:STATe <State>

Parameters

State

ON or 1

OFF or 0

:FFT:MARKer:STATe?

Returns

ON

OFF

:FFT:MARKer:SOURce <Source>

Parameters

Source

FFT[N]

:FFT:MARKer:SOURce?

Returns

FFT[N]

:FFT:MARKer:CLEar

:FFT:MARKer[MarkerN]:STATe <State>

Parameters

State

ON or 1

OFF or 0

:FFT:MARKer[MarkerN]:STATe?

Returns

ON

OFF

:FFT:MARKer[MarkerN]:FREQuency <Frequency>

Parameters

Frequency

Real in Hz

:FFT:MARKer[MarkerN]:FREQuency?

Returns

Real in Hz

:FFT:MARKer[MarkerN]:AMPLitude <Amplitude>

Parameters

Amplitude

Real in unit

See :FFT:SCALE for units

:FFT:MARKer[MarkerN]:AMPLitude?

Returns

Real in unit

See :FFT:SCALE for units

:FFT:MARKer[MarkerN]:ADD

:FFT:MARKer[MarkerN]:REMove

3.34 :DECode

This section covers adding, removing and configuring decode instances.

- [N] has to be replaced with a decode instance index, for example: DECode2
- Decode instances are automatically created when a setting is changed for a non-existent instance.

:DECode:STAtE <State>

Parameters

State

ON or 1

OFF or 0

:DECode:STAtE?

Returns

ON

OFF

:DECode:CLEAr

:DECode[N]:STAtE <State>

Parameters

State

ON or 1

OFF or 0

:DECode[N]:STAtE?

Returns

ON

OFF

:DECode[N]:NFORmat <Format>

Parameters

Format

BIN

DEC

HEX

ASCIi

:DECode [N] :NFORmat?

Returns

BIN

DEC

HEX

ASCii

:DECode [N] :TABLE <Table>

Parameters

Table

ON or 1

OFF or 0

:DECode [N] :TABLE?

Returns

ON

OFF

:DECode [N] :ADD

:DECode [N] :REMove

3.35 :DECode [N] :I2C

This section covers the configuration of the I2C bus for decode instances.

- [N] has to be replaced with a decode instance index, for example: DECode2
- Decode instances are automatically created when a setting is changed for a non-existent instance.

:DECode [N] :I2C:CLOCK:SOURce <Source>

Parameters

Source

[Channel]]

:DECode [N] :I2C:CLOCK:SOURce?

Returns

[Channel]

:DECode [N] :I2C:CLOCK:SLOPe <Slope>

Parameters

Slope

RISing

FALLing

:DECode [N] :I2C:CLOCK:SLOPe?

Returns

RISing

FALLing

:DECode [N] :I2C:CLOCK:THReshoId:LOWer <Threshold>

Parameters

Threshold

Real in V or A

:DECode [N] :I2C:CLOCK:THReshoId:LOWer?

Returns

Real in V or A

:DECode [N] :I2C:CLOCK:THReshoId:UPPer <Threshold>

Parameters

Threshold

Real in V or A

:DECode [N] :I2C:CLOCK:THReshold:UPPer?

Returns

Real in V or A

:DECode [N] :I2C:DATA:SOURce <Source>

Parameters

Source

[Channel]

:DECode [N] :I2C:DATA:SOURce?

Returns

[Channel]

:DECode [N] :I2C:DATA:ACTive <Active>

Parameters

Active

HIGH

LOW

:DECode [N] :I2C:DATA:ACTive?

Returns

HIGH

LOW

:DECode [N] :I2C:DATA:THReshold:LOWer <Threshold>

Parameters

Threshold

Real in V or A

:DECode [N] :I2C:DATA:THReshold:LOWer?

Returns

Real in V

:DECode [N] :I2C:DATA:THReshold:UPPer <Threshold>

Parameters

Threshold

Real in V or A

:DECode [N] :I2C:DATa:THResho1d:UPPer?

Returns

Real in V

3.36 :DECode [N] :SPI

This section covers the configuration of the SPI bus for decode instances.

- [N] has to be replaced with a decode instance index, for example: DECode2
- Decode instances are automatically created when a setting is changed for a non-existent instance.

:DECode [N] :SPI:TIMEout <Timeout>

Parameters

Timeout

Real in s

:DECode [N] :SPI:TIMEout?

Returns

Real in s

:DECode [N] :SPI:BITS <Bits>

Parameters

Bits

Integer

:DECode [N] :SPI:BITS?

Returns

Integer

:DECode [N] :SPI:BORDER <Order>

Parameters

Order

MSBFirst

LSBFirst

:DECode [N] :SPI:BORDER?

Returns

MSBFirst

LSBFirst

:DECode [N] :SPI:CSElect:SOURce <Source>

Parameters

Source

[Channel]

:DECode [N] :SPI:CSElect:SOURce?

Returns

[Channel]

:DECode [N] :SPI:CSElect:ACTive <Active>

Parameters

Active

HIGH

LOW

:DECode [N] :SPI:CSElect:ACTive?

Returns

HIGH

LOW

:DECode [N] :SPI:CSElect:THReshold:LOWer <Threshold>

Parameters

Threshold

Real in V or A

:DECode [N] :SPI:CSElect:THReshold:LOWer?

Returns

Real in V

:DECode [N] :SPI:CSElect:THReshold:UPPer <Threshold>

Parameters

Threshold

Real in V

:DECode [N] :SPI:CSElect:THReshold:UPPer?

Returns

Real in V

:DECode [N] :SPI:CLOCK:SOURce <Source>

Parameters

Source

[Channel]

:DECode [N] :SPI:CLOCK:SOURce?

Returns

[Channel]

:DECode [N] :SPI:CLOCK:SLOPe <Slope>

Parameters

Slope

RISing

FALLing

:DECode [N] :SPI:CLOCK:SLOPe?

Returns

RISing

FALLing

:DECode [N] :SPI:CLOCK:THReshoId:LOWer <Threshold>

Parameters

Threshold

Real in V or A

:DECode [N] :SPI:CLOCK:THReshoId:LOWer?

Returns

Real in V or A

:DECode [N] :SPI:CLOCK:THReshoId:UPPer <Threshold>

Parameters

Threshold

Real in V or A

:DECode [N] :SPI:CLOCK:THReshoId:UPPer?

Returns

Real in V or A

:DECode [N] :SPI:MOSi:SOURce <Source>

Parameters

Source

[Channel]

:DECode [N] :SPI:MOSi:SOURce?

Returns

[Channel]

:DECode [N] :SPI:MOSi:ACTive <Active>

Parameters

Active

HIGH

LOW

:DECode [N] :SPI:MOSi:ACTive?

Returns

HIGH

LOW

:DECode [N] :SPI:MOSi:THReshoId:LOWer <Threshold>

Parameters

Threshold

Real in V or A

:DECode [N] :SPI:MOSi:THReshoId:LOWer?

Returns

Real in V

:DECode [N] :SPI:MOSi:THReshoId:UPPer <Threshold>

Parameters

Threshold

Real in V or A

:DECode [N] :SPI:MOSi:THReshoId:UPPer?

Returns

Real in V

:DECode [N] :SPI:MISo:SOURce <Source>

Parameters

Source

[Channel]

:DECode [N] :SPI:MISo:SOURce?

Returns

[Channel]

:DECode [N] :SPI:MISo:ACTive <Active>

Parameters

Active

HIGH

LOW

:DECode [N] :SPI:MISo:ACTive?

Returns

HIGH

LOW

:DECode [N] :SPI:MISo:THReshoId:LOWer <Threshold>

Parameters

Threshold

Real in V or A

:DECode [N] :SPI:MISo:THReshoId:LOWer?

Returns

Real in V

:DECode [N] :SPI:MISo:THReshoId:UPPer <Threshold>

Parameters

Threshold

Real in V or A

:DECode [N] :SPI:MISo:THReshoId:UPPer?

Returns

Real in V

3.37 :DECode [N] :UART

This section covers the configuration of the UART bus for decode instances.

- [N] has to be replaced with a decode instance index, for example: DECode2
- Decode instances are automatically created when a setting is changed for a non-existent instance.

:DECode [N] :UART:BRATe <Rate>

Parameters

Rate

Integer in bps

:DECode [N] :UART:BRATe?

Returns

Integer in bps

:DECode [N] :UART:PARity <Parity>

Parameters

Parity

NONe

EVEN

ODD

MARK

SPACe

:DECode [N] :UART:PARity?

Returns

NONe

EVEN

ODD

MARK

SPACe

:DECode [N] :UART:SBIT <Bit>

Parameters

Bit

1

1.5

2

:DECode [N] :UART:SBIT?

Returns

1
1.5
2

:DECode [N] :UART:BITS <Rate>

Parameters

Rate

Integer

:DECode [N] :UART:BITS?

Returns

Integer

:DECode [N] :UART:BORDER <Order>

Parameters

Order

MSBFirst
LSBFirst

:DECode [N] :UART:BORDER?

Returns

MSBFirst
LSBFirst

:DECode [N] :UART:RTX:SOURce <Source>

Parameters

Source

[Channel]

:DECode [N] :UART:RTX:SOURce?

Returns

[Channel]

:DECode [N]:UART:RTX:ACTive <Active>

Parameters

Active

HIGH

LOW

:DECode [N]:UART:RTX:ACTive?

Returns

HIGH

LOW

:DECode [N]:UART:RTX:THResho1d:LOWer <Threshold>

Parameters

Threshold

Real in V or A

:DECode [N]:UART:RTX:THResho1d:LOWer?

Returns

Real in V

:DECode [N]:UART:RTX:THResho1d:UPPer <Threshold>

Parameters

Threshold

Real in V or A

:DECode [N]:UART:RTX:THResho1d:UPPer?

Returns

Real in V

3.38 :DECode[N]:PARAllel

This section covers the configuration of the parallel bus for decode instances.

- [N] has to be replaced with a decode instance index, for example: DECode2
- Decode instances are automatically created when a setting is changed for a non-existent instance.

:DECode[N]:PARAllel:CLOCK:SOURce <Source>

Parameters

Source

[Channel]]

:DECode[N]:PARAllel:CLOCK:SOURce?

Returns

[Channel]

:DECode[N]:PARAllel:CLOCK:SLOPe <Slope>

Parameters

Slope

RISing

FALLing

:DECode[N]:PARAllel:CLOCK:SLOPe?

Returns

RISing

FALLing

:DECode[N]:PARAllel:CLOCK:THReshold:LOWer <Threshold>

Parameters

Threshold

Real in V or A

:DECode[N]:PARAllel:CLOCK:THReshold:LOWer?

Returns

Real in V or A

:DECode[N]:PARAllel:CLOCK:THReshold:UPPer <Threshold>

Parameters

Threshold

Real in V or A

:DECode [N]:PARAllel:CLOCK:THReshold:UPPer?

Returns

Real in V or A

:DECode [N]:PARAllel:BITS <Bits>

Parameters

Bits

Integer

:DECode [N]:PARAllel:BITS?

Returns

Integer

:DECode [N]:PARAllel:BIT[BitN]:SOURce <Source>

Parameters

Source

[Channel]

:DECode [N]:PARAllel:BIT[BitN]:SOURce?

Returns

[Channel]

:DECode [N]:PARAllel:BIT[BitN]:ACTive <Active>

Parameters

Active

HIGH

LOW

:DECode [N]:PARAllel:BIT[BitN]:ACTive?

Returns

HIGH

LOW

:DECode [N]:PARAllel:BIT[BitN]:THReshold:LOWer <Threshold>

Parameters

Threshold

Real in V or A

:DECode[N]:PARAllel:BIT[BitN]:THReshold:LOWer?

Returns

Real in V

:DECode[N]:PARAllel:BIT[BitN]:THReshold:UPPer <Threshold>

Parameters

Threshold

Real in V or A

:DECode[N]:PARAllel:BIT[BitN]:THReshold:UPPer?

Returns

Real in V

3.39 :CURSor

This section covers the cursor configuration.

- [AnalogN] has to be replaced with an analog channel index, for example: CHANnel1

:CURSor:STAtE <State>

Parameters

State

ON or 1

OFF or 0

:CURSor:STAtE?

Returns

ON

OFF

:CURSor:SOURce <Source>

Parameters

Source

CHANnel[AnalogN]

:CURSor:SOURce?

Returns

CHANnel[AnalogN]

:CURSor:TDELta <State>

Parameters

State

TIME

FREQuency

BOTH

:CURSor:TDELta?

Returns

TIME

FREQuency

BOTH

:CURSor:TRACking <Tracking>

Parameters

Tracking

ON or 1

OFF or 0

:CURSor:TRACking?

Returns

ON

OFF

:CURSor:[X/Y]:STATe <State>

Parameters

State

ON or 1

OFF or 0

:CURSor:[X/Y]:STATe?

Returns

ON

OFF

:CURSor:[X/Y]:COUPling <Coupling>

Parameters

Coupling

ON or 1

OFF or 0

:CURSor:[X/Y]:COUPling?

Returns

ON

OFF

:CURSor:[A/B]:STATe <State>

Parameters

State

ON or 1

OFF or 0

:CURSor:[A/B]:STATe?

Returns

ON

OFF

:CURSor:[A/B]:[X/Y]:STATe <State>

Parameters

State

ON or 1

OFF or 0

:CURSor:[A/B]:[X/Y]:STATe?

Returns

ON

OFF

:CURSor:[A/B]:[X/Y]:POStion <Position>

Parameters

Position

Real in V, A or s

:CURSor:[A/B]:[X/Y]:POStion?

Returns

Real in V, A or s

3.40 :FGENERator

This section covers the configuration of general function generator settings.

:FGENERator:STATE <State>

Parameters

State

ON or 1

OFF or 0

:FGENERator:STATE?

Returns

ON

OFF

:FGENERator:LOAD <Load>

Parameters

Load

HIZ

500HM

:FGENERator:LOAD?

Returns

HIZ

500HM

3.41 :FGENERator:WAVEform

This section covers the configuration of function generator waveform settings.

:FGENERator:WAVEform:SHAPE <Shape>

Parameters

Shape

SINe
RECTangle
PULSe
RAMP
NOISe
ARBitrary
DC

:FGENERator:WAVEform:SHAPE?

Returns

SINe
RECTangle
PULSe
RAMP
NOISe
ARBitrary
DC

:FGENERator:WAVEform:FREQuency <Frequency>

Parameters

Frequency

Real in Hz

:FGENERator:WAVEform:FREQuency?

Returns

Real in Hz

:FGENERator:WAVEform:PERiod <Frequency>

Parameters

Frequency

Real in s

:FGENERator:WAVEform:PERiod?

Returns

Real in s

:FGENERator:WAVEform:AMPLitude <Frequency>

Parameters

Frequency

Real in V

:FGENERator:WAVEform:AMPLitude?

Returns

Real in V

:FGENERator:WAVEform:RMS <Frequency>

Parameters

Frequency

Real in Vrms

:FGENERator:WAVEform:RMS?

Returns

Real in Vrms

:FGENERator:WAVEform:OFFSet <Frequency>

Parameters

Frequency

Real in V

:FGENERator:WAVEform:OFFSet?

Returns

Real in V

:FGENERator:WAVEform:LEVel:HIGH <Level>

Parameters

Level

Real in V

:FGENERator:WAVEform:LEVel:HIGH?

Returns

Real in V

:FGENERator:WAVEform:LEVel:LOW <Level>

Parameters

Level

Real in V

:FGENERator:WAVEform:LEVel:LOW?

Returns

Real in V

3.42 :FGENERator:WAVEform:RECTangle

This section covers the configuration of function generator waveform rectangle settings.

:FGENERator:WAVEform:RECTangle:DUTY <Duty>

Parameters

Duty

Real in %

:FGENERator:WAVEform:RECTangle:DUTY?

Returns

Real in %

3.43 :FGENERator:WAVEform:PULSE

This section covers the configuration of function generator waveform pulse settings.

:FGENERator:WAVEform:PULSE:DUTY <Duty>

Parameters

Duty

Real in %

:FGENERator:WAVEform:PULSE:DUTY?

Returns

Real in %

:FGENERator:WAVEform:PULSE:RTIME <Time>

Parameters

Time

Real in s

:FGENERator:WAVEform:PULSE:RTIME?

Returns

Real in s

:FGENERator:WAVEform:PULSE:FTIME <Time>

Parameters

Time

Real in s

:FGENERator:WAVEform:PULSE:FTIME?

Returns

Real in s

3.44 :FGENERator:WAVEform:RAMP

This section covers the configuration of function generator waveform ramp settings.

:FGENERator:WAVEform:RAMP:SYMMetry <Symmetry>

Parameters

Symmetry

Real in %

:FGENERator:WAVEform:RAMP:SYMMetry?

Returns

Real in %

3.45 :FGENERator:MODification

This section covers the configuration of function generator modification settings.

:FGENERator:MODification:TYPE <Type>

Parameters

Type

NONE

MODulation

SWEep

BURSt

:FGENERator:MODification:TYPE?

Returns

NONE

MODulation

SWEep

BURSt

3.46 :FGENERator:MODification:MODulation

This section covers the configuration of function generator modulation modification settings.

:FGENERator:MODification:MODulation:TYPE <Type>

Parameters

Type

AM
DSSCAM
FM
PM

:FGENERator:MODification:MODulation:TYPE?

Returns

AM
DSSCAM
FM
PM

:FGENERator:MODification:MODulation:SHAPE <Shape>

Parameters

Shape

SINe
RECTangle
TRIangle
RUP
RD0wn
NOISe
ARBitrary

:FGENERator:MODification:MODulation:SHAPE?

Returns

SINe
RECTangle
TRIangle
RUP
RD0wn
NOISe
ARBitrary

:FGENERator:MODification:MODulation:FREQuency <Frequency>

Parameters

Frequency

Real in Hz

:FGENERator:MODification:MODulation:FREQuency?

Returns

Real in Hz

:FGENERator:MODification:MODulation:AM:DEPTH <Depth>

Parameters

Depth

Real in %

:FGENERator:MODification:MODulation:AM:DEPTH?

Returns

Real in %

:FGENERator:MODification:MODulation:DSSCAM:DEPTH <Depth>

Parameters

Depth

Real in %

:FGENERator:MODification:MODulation:DSSCAM:DEPTH?

Returns

Real in %

:FGENERator:MODification:MODulation:FM:FDEVIation <Deviation>

Parameters

Deviation

Real in Hz

:FGENERator:MODification:MODulation:FM:FDEVIation?

Returns

Real in Hz

:FGENerator:MODification:MODulation:PM:PDEViation <Deviation>

Parameters

Deviation

Real in degrees

:FGENerator:MODification:MODulation:PM:PDEViation?

Returns

Real in degrees

3.47 :FGENERator:MODification:SWEEp

This section covers the configuration of function generator sweep modification settings.

:FGENERator:MODification:SWEEp:TYPE <Type>

Parameters

Type

LINear

LOGarithmic

:FGENERator:MODification:SWEEp:TYPE?

Returns

LINear

LOGarithmic

:FGENERator:MODification:SWEEp:DIRection <Direction>

Parameters

Direction

UP

DOWN

BOTH

:FGENERator:MODification:SWEEp:DIRection?

Returns

UP

DOWN

BOTH

:FGENERator:MODification:SWEEp:TIME:UP <Time>

Parameters

Time

Real in s

:FGENERator:MODification:SWEEp:TIME:UP?

Returns

Real in s

:FGENERator:MODification:SWEEp:TIME:DOWN <Time>

Parameters

Time

Real in s

:FGENERator:MODification:SWEEp:TIME:DOWN?

Returns

Real in s

:FGENERator:MODification:SWEEp:FREQuency:START <Start>

Parameters

Start

Real in Hz

:FGENERator:MODification:SWEEp:FREQuency:START?

Returns

Real in Hz

:FGENERator:MODification:SWEEp:FREQuency:STOP <Stop>

Parameters

Stop

Real in Hz

:FGENERator:MODification:SWEEp:FREQuency:STOP?

Returns

Real in Hz

:FGENERator:MODification:SWEEp:HOLD:START <Start>

Parameters

Start

Real in s

:FGENERator:MODification:SWEEp:HOLD:START?

Returns

Real in s

:FGENerator:MODification:SWEep:HOLD:STOP <Stop>

Parameters

Stop

Real in s

:FGENerator:MODification:SWEep:HOLD:STOP?

Returns

Real in s

3.48 :FGENERator:MODification:BURSt

This section covers the configuration of function generator burst modification settings.

:FGENERator:MODification:BURSt:TTYPe <Type>

Parameters

Type

AUTomatic

MANual

:FGENERator:MODification:BURSt:TTYPe?

Returns

AUTomatic

MANual

:FGENERator:MODification:BURSt:SPHase <Phase>

Parameters

Phase

Real in degrees

:FGENERator:MODification:BURSt:SPHase?

Returns

Real in degrees

:FGENERator:MODification:BURSt:CYCLes <Cycles>

Parameters

Cycles

Integer

:FGENERator:MODification:BURSt:CYCLes?

Returns

Integer

:FGENERator:MODification:BURSt:TDELay <Delay>

Parameters

Delay

Real in s

:FGENERator:MODification:BURSt:TDElay?

Returns

Real in s

:FGENERator:MODification:BURSt:PERiod <Period>

Parameters

Period

Real in s

:FGENERator:MODification:BURSt:PERiod?

Returns

Real in s

:FGENERator:MODification:BURSt:ILTYpe <Type>

Parameters

Type

FP0int

MANual

:FGENERator:MODification:BURSt:ILTYpe?

Returns

FP0int

MANual

:FGENERator:MODification:BURSt:ILEVel <Period>

Parameters

Period

Real in %

:FGENERator:MODification:BURSt:ILEVel?

Returns

Real in %

:FGENERator:MODification:BURSt:TRIGger

3.49 :BPLot

This section covers the configuration of the general bode plot settings and bode plot instances.

- [N] has to be replaced with a bode plot instance index, for example: BPLot3
- [AnaLogN] has to be replaced with an analog channel index, for example: CHANneL1

:BPLot:STATE <State>

Parameters

State

ON or 1

OFF or 0

:BPLot:STATE?

Returns

ON

OFF

:BPLot:FSCale <Scale>

Parameters

Scale

LINear

LOGarithmic

:BPLot:FSCale?

Returns

LINear

LOGarithmic

:BPLot:VMODE <Mode>

Parameters

Mode

RATio

AMPLitude

:BPLot:VMODE?

Returns

RATio

AMPLitude

:BPLot:AScale <Scale>

Parameters

Scale

DBM

DBV

DBMV

DBUV

V

:BPLot:AScale?

Returns

DBM

DBV

DBMV

DBUV

V

:BPLot:RScale <Scale>

Parameters

Scale

LINear

LOGarithmic

:BPLot:RScale?

Returns

LINear

LOGarithmic

:BPLot:FREQuency:CENTer <Center>

Parameters

Center

Real in Hz

:BPLot:FREQuency:CENTer?

Returns

Real in Hz

**:BPLot:FREQuency:SPAN **

Parameters

Span

Real in Hz

:BPLot:FREQuency:SPAN?

Returns

Real in Hz

:BPLot:FREQuency:START <Start>

Parameters

Start

Real in Hz

:BPLot:FREQuency:START?

Returns

Real in Hz

:BPLot:FREQuency:STOP <Stop>

Parameters

Stop

Real in Hz

:BPLot:FREQuency:STOP?

Returns

Real in Hz

:BPLot[N]:STATE <State>

Parameters

State

ON or 1

OFF or 0

:BPLot[N]:STATE?

Returns

ON

OFF

:BPLot[N]:OUT <Out>

Parameters

Out

CHANnel[AnalogN]

:BPLot[N]:OUT?

Returns

CHANnel[AnalogN]

:BPLot[N]:OFFSet <Offset>

Parameters

Offset

Real in unit

See :FFT:SCALE for units

Real in ratio

Real in logarithmic ratio

:BPLot[N]:OFFSet?

Returns

Real in unit

See :FFT:SCALE for units

Real in ratio

Real in logarithmic ratio

:BPLot[N]:SCALE <Scale>

Parameters

Scale

Real in unit

See :BPLot:AScale for units

Real in ratio

Real in logarithmic ratio

:BPLot[N]:SCALE?

Returns

Real in unit

See :BPLot:AScale for units

Real in ratio

Real in logarithmic ratio

:BPLot[N]:POFFSet <Offset>

Parameters

Offset

Real in degrees

:BPLot[N]:POFFSet?

Returns

Real in degrees

:BPLot[N]:PSCALE <Scale>

Parameters

Scale

Real in degrees

:BPLot[N]:PSCALE?

Returns

Real in degrees

3.50 :BPLot:SWEep

This section focuses on the configuration of bode plot sweep settings.

- [N] has to be replaced with a bode plot instance index, for example: BPLot3

:BPLot[N]:IN <In>

Parameters

In

CHANnel[AnalogN]

:BPLot[N]:IN?

Returns

CHANnel[AnalogN]

:BPLot:SWEep:POINts <Points>

Parameters

Points

Integer

:BPLot:SWEep:POINts?

Returns

Integer

:BPLot:SWEep:PDIStribution <Distribution>

Parameters

Distribution

LINear

LOGarithmic

:BPLot:SWEep:PDIStribution?

Returns

LINear

LOGarithmic

:BPLot:SWEep:FREQuency:STARt <Start>

Parameters

Start

Real in Hz

:BPLot:SWEEp:FREQuency:STArT?

Returns

Real in Hz

:BPLot:SWEEp:FREQuency:STOP <Stop>

Parameters

Stop

Real in Hz

:BPLot:SWEEp:FREQuency:STOP?

Returns

Real in Hz

:BPLot:SWEEp:VSCale <Scale>

Parameters

Scale

DBM

DBV

DBMV

DBUV

V

:BPLot:SWEEp:VSCale?

Returns

DBM

DBV

DBMV

DBUV

V

:BPLot:SWEEp:VOFFset <Offset>

Parameters

Offset

Real in unit

See :BPLot:SWEEp:VSCale for units

:BPLot:SWEEp:VOFFset?

Returns

Real in unit

See :BPLot:SWEEp:VSCale for units

:BPLot:SWEEP:VAMPLitude <Amplitude>**Parameters****Amplitude****Real in unit**

See :BPLot:SWEEP:VScale for units

:BPLot:SWEEP:VAMPLitude?**Returns****Real in unit**

See :BPLot:SWEEP:VScale for units

3.51 :BPLot[N]:DATA

This section focuses on the transfer of bode plot data.

- [N] has to be replaced with a bode plot instance index, for example: BPLot3

:BPLot[N]:DATA:AScale <Scale>

Parameters

Scale

DBM
 DBV
 DBMV
 DBUV
 V

:BPLot[N]:DATA:AScale?

Returns

DBM
 DBV
 DBMV
 DBUV
 V

:BPLot[N]:DATA:POINTs?

Returns

Point data See 3.51.1.

:BPLot[N]:DATA:PACKed? <Scale>

Parameters

Scale

DBM
 DBV
 DBMV
 DBUV
 V

Returns

Packed point data See 3.51.2.

3.51.1 Bode Plot Points

All data in this section has a *Block Data* header (see 1.1.3).

The Bode Plot points N is implicitly defined through the *Block Data* header

Field	Count (bytes)	Data Type	Json/XML Name
Points (Frequency / Amplitude / Ratio / Phase)	4 * N	4-byte floating-point / 4-byte floating-point / 4-byte floating-point / 4-byte floating-point	Frequency / Amplitude / Ratio / Phase

3.51.2 Bode Plot Points Packed

All data in this section has a *Block Data* header (see 1.1.3).

The Bode Plot points including a metadata header

Field	Count (bytes)	Data Type	Json/XML Name
N = Number of points	4	4-byte unsigned integer	PointCount
Points (Frequency / Amplitude / Ratio / Phase)	4 * N	4-byte floating-point / 4-byte floating-point / 4-byte floating-point / 4-byte floating-point	Frequency / Amplitude / Ratio / Phase

3.52 :HISTory

This section covers the configuration of the history.

:HISTory:STATe <State>

Parameters

State

ON or 1

OFF or 0

:HISTory:STATe?

Returns

ON

OFF

:HISTory:RECOrd <Record>

Parameters

Record

Integer starting at 1

:HISTory:RECOrd?

Returns

Integer starting at 1

:HISTory:TOTAL?

Returns

Integer

:HISTory:NEXT

:HISTory:PREVious

:HISTory:PLAY <Play>

Parameters

Play

Integer between -5 and 5

Default 1, specifies the direction and speed of the playback

:HISTory:STOP

:HISTory:REPeat <Repeat>**Parameters****Repeat****ON or 1****OFF or 0**

:HISTory:REPeat?**Returns****ON****OFF**

3.53 :DISPlay

This section covers the configuration of the display settings.

:DISPlay:BRIGhtness <Brightness>

Parameters

Brightness

Integer between 0 and 100

:DISPlay:BRIGhtness?

Returns

Integer between 0 and 100

:DISPlay:SCReenshot? <Format>

Parameters

Format

BMP

PNG

Returns

Screenshot data

The data in the specified format can be saved directly to a file

3.54 :DISPlay:WAVeform

This section covers the configuration of the display waveform settings.

:DISPlay:WAVeform:INTERpolation <Interpolation>

Parameters

Interpolation

NONE
HOLD
LINEar
SINC

:DISPlay:WAVeform:INTERpolation?

Returns

NONE
HOLD
LINEar
SINC

:DISPlay:WAVeform:INTENSITY <Intensity>

Parameters

Intensity

Integer between 0 and 100

:DISPlay:WAVeform:INTENSITY?

Returns

Integer between 0 and 100

:DISPlay:WAVeform:COLor <Color>

Parameters

Color

INTensity
INVerted
HEATmap7
HEATmap5
GRADient2

:DISPlay:WAVeform:COLor?

Returns

INTensity**INVerted****HEATmap7****HEATmap5****GRADient2**

:DISPlay:WAVeform:PERSistence <Persistence>

Parameters

Persistence**OFF****INFinite****Real in s**

:DISPlay:WAVeform:PERSistence?

Returns

OFF**INFinite****Real in s**

3.55 :DISPlay:GRID

This section covers the configuration of the display grid settings.

:DISPlay:GRID:DETail <Detail>

Parameters

Detail

NONe
SIMPlE
NORMaL
FINe

:DISPlay:GRID:DETail?

Returns

NONe
SIMPlE
NORMaL
FINe

:DISPlay:GRID:INTensity <Intensity>

Parameters

Intensity

LOW
HIGH

:DISPlay:GRID:INTensity?

Returns

LOW
HIGH

:DISPlay:GRID:MLABel <Label>

Parameters

Label

HIDDEN
ONCHange
VISible

:DISPlay:GRID:MLABel?**Returns****HIDDEN****ONCHange****VISible**

3.56 :ZOOM

This section covers the configuration of the zoom.

:ZOOM:STATe <State>

Parameters

State

ON or 1

OFF or 0

:ZOOM:STATe?

Returns

ON

OFF

:ZOOM:HORizontal:POSition <Offset>

Parameters

Offset

Real in %

:ZOOM:HORizontal:POSition?

Returns

Real in %

:ZOOM:HORizontal:SCALE <Offset>

Parameters

Offset

Real

:ZOOM:HORizontal:SCALE?

Returns

Real

:ZOOM:VERTical:POSition <Offset>

Parameters

Offset

Real in %

:ZOOM:VERTical:POSition?

Returns

Real in %

:ZOOM:VERTical:SCALe <Offset>

Parameters

Offset

Real

:ZOOM:VERTical:SCALe?

Returns

Real