

LR8101 LR8102

HIOKI

通信コマンド取扱説明書

データロガー



JA

1 LR8100通信コマンド取扱説明書	10
1.1 はじめに	10
2 概要	11
2.1 機器の仕様、設定	11
2.2 コマンド概要	14
2.3 レジスタ	24
2.4 入出力バッファ	28
3 コマンド	29
3.1 データの種類	29
3.2 共通コマンド	30
*CLS	30
*ESR?	31
*IDN?	32
*OPC	33
*OPC?	34
*OPT?	35
*RST	36
*STB?	37
*TST?	38
*WAI	39
*ESR0?	40
3.3 実行処理	41
*ABORT	41
*ERRor?	42
*HEADer	43
*STARt	44
*STATUS?	45
*STOP	46
*NRMFlag?	47
*WAITNextsmp!	49
3.4 測定設定 (CONFigure)	50
*ADDComent	50
*ADDDate	51
*ATSAve	52
*AUTOFolder	54
*EXTRECSamp	55
*FILEName	56
*RECTime	57
*SAMPle	58
*SAMPKind	59
*SAVECalc	60
*SAVEDeci	61
*SAVEFormat	62
*SAVEKind	63
*SAVELen	64
*SAVEMode	65
*SAVEPri	66
*SAVEReg	67
*SAVESep	68
*SAVETime	69

!SAVEWave	70
!START	71
!STARTTime	72
!STOP	73
!STOPTime	74
!SYNC:SET	75
!SYNC:CHECK?	76
!THINData	77
!THINOut	78
3.5 入力チャンネル (MODUle)	79
!ADJUST?	79
!DATARate	80
!DFILter?	81
!FILTer	82
!IDN?	83
!INMOde	84
!PCOMode	86
!PCOStart	87
!PCOUnt	88
!PFILTer	89
!PINMOde	90
!PRANGe	91
!PRESet	92
!PSLOPe	93
!PSMooth	94
!PTHRe	95
!RANGe	96
!RJC	97
!SENSor	98
!STORE	99
!WIRE	100
3.6 スケーリング (SCALing)	101
!KIND	101
!OFFSet	102
!SCUPLOW	103
!SENSE	104
!SET	105
!UNIT	106
!VOLT	107
!VOUPLOW	108
3.7 コメント (COMMeNt)	109
!ALMCH	109
!CH	111
!TITLe	112
!MODule	113
3.8 トリガ (TRIGger)	114
!DETECTDate?	114
!DETECTTime?	115
!MANUal	116
!MODE	117

•PRETrig	118
•SET	119
•SOURce	120
•SSOURce	121
•TIMER	122
•TIMIng	123
•TMINTvl	124
•ANALog:START:KIND	125
•ANALog:STOP:KIND	125
•CALCulate:START:KIND	125
•CALCulate:STOP:KIND	125
•PULSe:START:KIND	125
•PULSe:STOP:KIND	125
•ANALog:START:LEVEL	127
•ANALog:STOP:LEVEL	127
•CALCulate:START:LEVEL	127
•CALCulate:STOP:LEVEL	127
•PULSe:START:LEVEL	127
•PULSe:STOP:LEVEL	127
•ANALog:START:LOWEr	129
•ANALog:STOP:LOWEr	129
•CALCulate:START:LOWEr	129
•CALCulate:STOP:LOWEr	129
•PULSe:START:LOWEr	129
•PULSe:STOP:LOWEr	129
•ANALog:START:SIDE	131
•ANALog:STOP:SIDE	131
•CALCulate:START:SIDE	131
•CALCulate:STOP:SIDE	131
•PULSe:START:SIDE	131
•PULSe:STOP:SIDE	131
•ANALog:START:SLOPe	133
•ANALog:STOP:SLOPe	133
•CALCulate:START:SLOPe	133
•CALCulate:STOP:SLOPe	133
•PULSe:START:SLOPe	133
•PULSe:STOP:SLOPe	133
•ANALog:START:UPPEr	135
•ANALog:STOP:UPPEr	135
•CALCulate:START:UPPEr	135
•CALCulate:STOP:UPPEr	135
•PULSe:START:UPPEr	135
•PULSe:STOP:UPPEr	135
•LOGic:START:PATTErn	137
•LOGic:STOP:PATTErn	137
•EXTErnal:START:KIND	138
•EXTErnal:STOP:KIND	138
3.9 警報 (ALARm)	139
•ACTive	139
•ARCD?	140

•ARCDNum?	141
•BEEP	142
•BURN	143
•FILTer	144
•HISTory	145
•HOLD	146
•SOURce	147
•ANALog:KIND	148
•CALCulate:KIND	148
•PULSe:KIND	148
•ANALog:LEVEL	150
•CALCulate:LEVEL	150
•PULSe:LEVEL	150
•ANALog:LOWEr	152
•CALCulate:LOWEr	152
•PULSe:LOWEr	152
•ANALog:SIDE	154
•CALCulate:SIDE	154
•PULSe:SIDE	154
•ANALog:SLOPe	156
•CALCulate:SLOPe	156
•PULSe:SLOPe	156
•ANALog:SLP2:TIME	158
•CALCulate:SLP2:TIME	158
•PULSe:SLP2:TIME	158
•ANALog:STIME	160
•CALCulate:STIME	160
•PULSe:STIME	160
•ANALog:UPPEr	162
•CALCulate:UPPEr	162
•PULSe:UPPEr	162
•LOGic:PATTern	164
3.10 システム (SYSTem)	165
•ADDComment	165
•ADDDate	166
•ADJDate?	167
•BEEP	168
•CALCSplit	169
•CHECK	170
•CHECK:ROMRam	170
•CHECK:MODule	171
•CHECK:IF:LAN1	172
•CHECK:IF:LAN2	173
•CHECK:MEDia:SD	174
•CHECK:MEDia:USB	174
•CLBDate?	175
•CLOCK:OUT	176
•COMMunicate:LAN:DHCP	177
•COMMunicate:LAN2:DHCP	177
•COMMunicate:LAN:HOSTname	179

•COMMunicate:LAN2:HOSTname	179
•COMMunicate:LAN:IPADdress	180
•COMMunicate:LAN2:IPADdress	180
•COMMunicate:LAN:SMASk	182
•COMMunicate:LAN2:SMASk	182
•COMMunicate:LAN:GATeway	184
•COMMunicate:LAN2:GATeway	184
•COMMunicate:LAN:CONTRol	186
•COMMunicate:LAN2:CONTRol	186
•COMMunicate:LAN:DNS	188
•COMMunicate:LAN2:DNS	188
•COMMunicate:LAN:UPDate	190
•COMMunicate:LAN2:UPDate	190
•COMMunicate:LAN2:SEND:IPADdress	191
•COMMunicate:LAN2:SEND:PORT	192
•COMMunicate:LAN2:SEND:ENDian	193
•COMMunicate:LAN2:SEND:FORMat	194
•COMMunicate:LAN2:SEND:SYNC	195
•DATAClear	196
•DATE	197
•DATETime	198
•DFORmat	199
•DSEParator	200
•FILEName	201
•LANGuage	202
•MARK	203
•SAVEPri	204
•STARt	205
•THINData	206
•THINOut	207
•TIME	208
•TIMEZone	209
•TMAXis	210
•EXT:IO1:KIND	211
•EXT:IO1:SLOPe:STARt	212
•EXT:IO1:SLOPe:STOP	213
•EXT:IO2:KIND	214
•EXT:IO2:SLOPe:STARt	215
•EXT:IO2:SLOPe:STOP	216
•EXT:IO3:KIND	217
•EXT:IO3:SLOPe:STARt	218
•EXT:IO3:SLOPe:STOP	219
•EXT:IO4:KIND	220
•EXTFILTer	221
•EXTSLOPe	222
•NTP:ADDRess	223
•NTP:CHECK?	224
•NTP:KIND	225
•NTP:SYNC	226
•NTP:STARt	227

•FTP:ADDRess	228
•FTP:AUTODel	229
•FTP:CERTIFICATE	230
•FTP:CHECK?	231
•FTP:DIR	232
•FTP:FILE:HOST	233
•FTP:FILE:IP	234
•FTP:FILE:TIME	235
•FTP:PASSword	236
•FTP:PASV	237
•FTP:PORT	238
•FTP:PROGress?	239
•FTP:SECURITY	240
•FTP:STATe?	241
•FTP:USE	242
•FTP:USER	243
•RTOuT	244
3.11 ストレージデータ (MEMory)	245
•ADATa?	245
•AFETch?	247
•AMAXPoint?	248
•APOINt	249
•AREAL?	250
•BDATa?	251
•BFETch?	253
•BREAL?	254
•CHSTore?	255
•FCHSTore?	256
•GETReal	257
•MAXPoint?	258
•POINt	259
•TAFETch?	260
•TARCH?	261
•TAREAL?	262
•TCHSTore?	263
•TFCHSTore?	264
•TOPPoint?	265
•TVFETch?	266
•TVRCH?	267
•TVREAL?	268
•VDATa?	269
•VFETch?	271
•VREAL?	272
3.12 画面表示 (DISPlay)	273
•MARK	273
•MARK?	274
•MARKJump?	275
3.13 演算 (CALCulate)	276
•MEASure	276
•MEAS:ANSWer?	277

!MEAS:FILE	278
!MEAS:INTegra	279
!MEAS:KIND	280
!MEAS:LEN	281
!MEAS:LEVEL	282
!MEAS:REG	283
!MEAS:SET	284
!MEAS:TARGet	286
!MEAS:TIME	287
!WAVE:KIND	288
!WAVE:STR	289
!WAVE:ARITHmetic:COEF:A	290
!WAVE:ARITHmetic:COEF:B	290
!WAVE:ARITHmetic:COEF:C	290
!WAVE:ARITHmetic:COEF:D	290
!WAVE:ARITHmetic:COEF:E	290
!WAVE:ARITHmetic:OPERator:A	292
!WAVE:ARITHmetic:OPERator:B	292
!WAVE:ARITHmetic:OPERator:C	292
!WAVE:ARITHmetic:OPERator:D	292
!WAVE:MOVE:POINT	294
!WAVE:RESet:BASE	295
!WAVE:RESet:INT	296
!WAVE:RESet:KIND	297
!WAVE:RESet:TIME	298
!WAVE:SOURce:SR1	299
!WAVE:SOURce:SR2	299
!WAVE:SOURce:SR3	299
!WAVE:SOURce:SR4	299
3.14 エラー (ERRor)	301
!BIT:ERRor?	301
!BIT:WARNing?	302
!BIT:WARNing:CLEAR	303
!LOG:ERRor?	304
!LOG:ERRor:CLEAR	305
!LOG:WARNing?	306
!LOG:WARNing:CLEAR	307
3.15 データ保存 (MEDia)	308
!SD:FINFo:SET?	308
!USB:FINFo:SET?	308
!SD:FREE?	309
!USB:FREE?	309
!SD:FORMat?	310
!USB:FORMat?	310
!SD:FLISt:SET?	311
!USB:FLISt:SET?	311
!SD:LOAD:SET	312
!USB:LOAD:SET	312
!SD:SAVE:DATA:MEM	314
!USB:SAVE:DATA:MEM	314

•SD:SAVE:DATA:CSV	314
•USB:SAVE:DATA:CSV	314
•SD:SAVE:DATA:MF4	314
•USB:SAVE:DATA:MF4	314
•SD:SAVE:SET	314
•USB:SAVE:SET	314
•SD:SAVE:A2L:LAN1	314
•USB:SAVE:A2L:LAN1	314
•SD:SAVE:A2L:LAN2	314
•USB:SAVE:A2L:LAN2	314
•SD:SAVE:CALC:CSV	314
•USB:SAVE:CALC:CSV	314
4 トラブルシューティング	317
4.1 有線LAN接続時	317

1.1 はじめに

- 本取扱説明書においてはコマンドに関する部分のみ扱っています。
- 本体の通信設定については、本体取扱説明書をご覧ください。
- **本取扱説明書で説明している通信コマンドを使った測定では、サンプリング同期したチャンネル数が750ch(1台あたり75chまで)のときに、最短50ミリ秒間隔でリアルタイムでデータの取得が可能です。50ミリ秒より早い間隔でリアルタイムでデータを取得したい場合は、専用のソフトウェアであるLogger Utilityを使用してください。**
- 本取扱説明書の内容につきましては万全を期していますが、ご不明な点や誤りなどお気づきのことがありましたら、本社カスタマーサポートまたは最寄りの支店までご連絡ください。
- 本取扱説明書は改善のため予告なしに記載事項を変更することがあります。
- 本取扱説明書を無断で転載、複製する事は禁止されています。
- Microsoft、Windows、Visual Basic、およびVisual Studioは米国 Microsoft Corporation の米国、日本およびその他の国における登録商標または商標です。
- その他の商品名、会社名は、一般に各社の商号、登録商標または商標です。

問い合わせ先

弊社ホームページにありますお問い合わせリンクから「使い方や仕様について確認したい」をご覧ください。

<https://www.hioki.co.jp/jp/contact/product-contact/>

更新履歴

- 初版 2023年12月

2.1 機器の仕様、設定

有線LANの仕様

本器ではLANケーブルでPCと接続できます。

ポート番号は、「本体取扱説明書 2.8 LANの設定と接続をする」で設定した番号が使用されます。

接続中に他のコンピュータから接続された場合、先に接続していたコンピュータを切断し、後から接続したコンピュータに接続します。

データリンク	IEEE802.3 Ethernet 1000BASE-T、100BASE-TX
伝送速度	1Gbps
プロトコル	TCP/IP

有線LANの設定

ネットワークへ接続する場合、設定項目を予めネットワークシステムの管理者(部署)から割り当ててもら
う必要があります。
必ず他の機器と重ならないようにします。以下の項目について管理者(部署)から設定を割り当ててもら
いメモしておきます。
通信中は設定を変更しないでください。

ホスト名(15文字まで)	XXXXXXXXXXXXXXXXXX
IPアドレス	XXX.XXX.XXX.XXX
サブネットマスク	XXX.XXX.XXX.XXX
デフォルトゲートウェイ (あり/なし)	XXX.XXX.XXX.XXX (ありの場合)
TCP/IPポート番号 (10進数4桁)	XXXX 本機のデフォルト値は、880Xです。 通信コマンドはポート番号の末尾、2を使用するため、 デフォルトの場合、8802を指定します。

既存のネットワークへは接続せず専用にネットワークを組む場合、管理者からアドレスを頂いても構い
ませんが、

管理者がいない場合は設定を一任される時があります。

この場合、以下のIPアドレスが推奨されています。

192.168.0.1～192.168.253.254

(192.168.0.0と192.168.0.255は使えません)

例えば、本体を192.168.1.2、コントローラ(パソコンなど)を192.168.1.1に割り当てます。

ロガーを複数台接続するときは192.168.1.3のように最後の数を順次増やします。

ホスト名は他と重ならない名前を適当に付けます。

その他の項目は、サブネットマスク:255.255.255.0、デフォルトゲートウェイ:OFF、ポート番号:8802などとするのが良いでしょう。

【設定例】

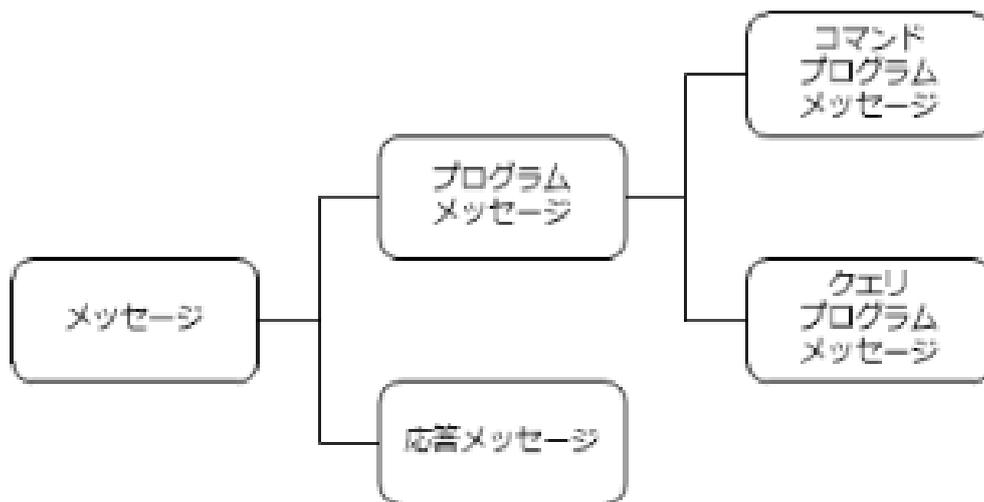
設定	本機(LR8100)	パソコン
ホスト名(15文字まで)	LR8100	PC
IPアドレス	192.168.1.2	192.168.1.1
サブネットマスク	255.255.255.0	255.255.255.0
デフォルトゲートウェイ	OFF	OFF
TCP/IPポート番号	880X	※1

※1:本器の設定が[880X]となっている場合は、パソコンの接続先ポートを[8802]にしてください。

2.2 コマンド概要

コマンド/メッセージ

通信機器と受信・送信されるデータをメッセージといいます。メッセージは以下の図のように分類されます。



プログラムメッセージ

コントローラから機器に送信するメッセージです。

応答メッセージ

機器からコントローラに送信するメッセージです。

このメッセージは、クエリー・プログラム・メッセージを受信し、
構文をチェックした時点で作成されます。

コマンド・プログラ
ム・メッセージ

機器の設定、リセットなどの機器を制御する命令です。

クエリー・プログラ
ム・メッセージ

機器の動作結果、測定結果、あるいは設定状態を問合せる命令です。

コマンド・プログラム・メッセージ、およびクエリー・プログラム・メッセージを総称してコマンドと呼びます。

(コマンドはHIOKI固有のSCPI言語です。)

コマンド・シンタックス

コマンドは大文字・小文字・その混合のいずれの場合も受け付けます。

コマンド名は、実行しようとする機能にできるだけ近いものが選ばれており、そのほとんどは短縮可能です。

コマンド名そのものを**ロングフォーム**といい、短縮したものを**ショートフォーム**といいます。

本取扱説明では、ショートフォームの部分を大文字で、小文字まで含めた部分をロングフォームとして記述しています。

本体からの応答メッセージは、大文字のロングフォームで返されます。

【例】

本取扱説明の表記 (コマンド名)	ショートフォーム	ロングフォーム
DISPlay	DISP	DISPLAY

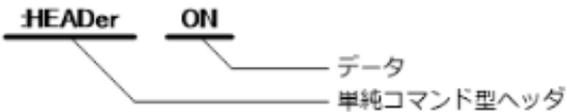
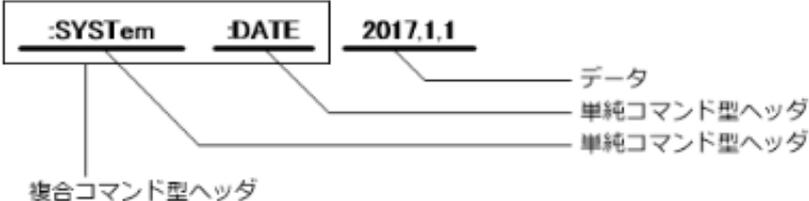
DiSpLayのように大文字と小文字が混ざっていても受け付けますが、DISPLA、DISPL、DIS はエラーとなります。

コマンド・プログラム・ヘッダ

ヘッダは、そのコマンドがどんな機能を持つかを示します。

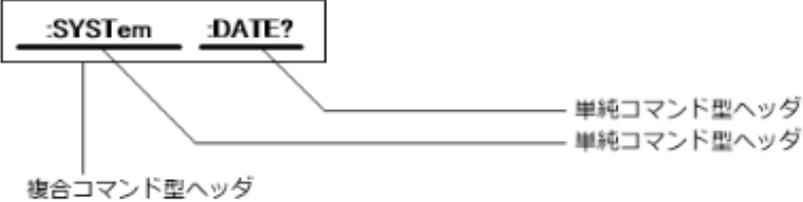
コマンドは必ずヘッダが必要で、「単純コマンド型」、「複合コマンド型」、および**「共通コマンド型」

**の3種類の形式があります。

コマンドの種類	説明
単純コマンド型	<p>1語で構成されたヘッダ</p> <p>【例】</p> 
複合コマンド型	<p>コロン「:」で区切られた複数の単純コマンド型ヘッダ</p> <p>【例】</p> 
共通コマンド型	<p>先頭の文字がアスタリスク「*」で始まり、IEEE488.2で規定されたコマンド</p> <p>【例】</p> <p style="text-align: center;">*RST</p>

クエリー・プログラム・ヘッダ

機器のコマンドに対する動作結果や設定を問合せのために使用し、プログラム・ヘッダの後に「?」をつけるとクエリーとして認識されます。

コマンドの種類	説明
単純コマンド型	<p>1語で構成されたヘッダ</p> <p>【例】</p> <p><u>HEADer?</u></p> 
複合コマンド型	<p>コロン「:」で区切られた複数の単純コマンド型ヘッダ</p> <p>【例】</p> <p><u>:SYSTem</u> <u>:DATE?</u></p> 
共通コマンド型	<p>先頭の文字がアスタリスク「*」で始まり、IEEE488.2で規定されたコマンド</p> <p>【例】</p> <p>*IDN?</p>

応答メッセージ

クエリーに対する応答メッセージは、プログラムメッセージと同様にヘッダ部とデータ部からなり、原則としてそのクエリーに対応するプログラムメッセージのフォーマットと同様のフォーマットで出力します。なお、ヘッダ部をなくすこともできます。

【例】

クエリー・プログラム・メッセージ

:SYSTem:TIME?

応答メッセージ

:SYSTEM:TIME 12,34,56(ヘッダの設定がONの場合)

12,34,56(ヘッダの設定がOFFの場合)

ターミネータとセパレータ

(1) メッセージターミネータ

メッセージターミネータとは、1回のメッセージ転送の区切りを意味します。
ただしターミネータ自身はメッセージに含まれません。

本体/通信ソフトの設定	ANSI文字コード(16進数)	意味	英語名称	デリミタの設定
LF	0Ah	改行	Line Feed	デリミタをLFに設定します。
CR+LF	0Dh 0Ah			デリミタをCR+LFに設定します。

(2) メッセージ単位セパレータ

セミコロン「;」はメッセージ単位セパレータとして使用され、1行に複数のメッセージを記述する場合に使用します。

`:SYSTEM:DATE 2017,1,1;:SYSTEM:TIME 12,34,56`

↑
メッセージ単位セパレータ

(3) ヘッダ・セパレータ

ヘッダとデータを持つメッセージにおいて、ヘッダとデータを分離するもの（ヘッダ・セパレータ）として、スペース「 」を使用します。

スペース「 」は実際のプログラムは表示しません。

`:SYSTEM:DATE 2017,1,1`

↑
スペース

(4) データ・セパレータ

カンマ「,」は、複数のデータ間を区切るもの（データ・セパレータ）として使用します。

`:SYSTEM:DATE 2017,1,1`

↑
コンマ

コマンド・ツリー

複数の複合コマンド型のメッセージを記述する場合、セミコロン（メッセージ単位セパレータ）「;」の次のヘッダにコロン「:」がないときは、原則としてヘッダが直前メッセージの最後のコロンより前のヘッダに引き続いているものとみなします。

この直前のヘッダをカレント・パスと呼ぶことにします。

【例1】

:SYSTem:DATE 2017,1,1;:SYSTem:TIME 12,34,56

【例2】

:SYSTem:DATE 2017,1,1;TIME 12,34,56

【例3】

:SYSTem:DATE 2017,1,1

（上記データを送信後、下記データを送信）

TIME 12,34,56

【例1】【例2】、および【例3】は、時刻を設定するメッセージです。

【例1】は、セミコロンの次にコロンがあるため、カレント・パスは**ルート**になります。従って次のコマンド検索はルートから行います。

【例2】は、「:SYSTem:DATE 2017,1,1」で**カレント・パス**が「:SYSTEM」になっているので、「TIME」の前の「:SYSTem:」を省略できます。

【例3】は、一度「:SYSTem:DATE 2017,1,1」を送信していますが、【例2】と同様に**カレント・パス**が「:SYSTEM」になっているので、「TIME」の前の「:SYSTem:」を省略できます。

コマンドの先端にあるコロン「:」は、コマンドの検索を**ルート**から行うことを表します。

データ形式

コマンドのデータは、文字データ、10進数値データ、および文字列データを使用します。

■文字データ

- (1) 先頭文字は、必ず英文字とします。
- (2) 先頭以外の文字は、英文字、数字、およびアンダースコア「_」だけを含みます。
- (3) 英文字は、送信側は大文字だけですが、受信側は大文字、小文字の両方を許容します。

【例】

:HEADER ON

■10進数値データ

- (1) 10進数の数値データ表現は、次に示す NR形式を使用します。
- (2) NR形式は、NR1、NR2、およびNR3の3種類で、これら3種類を全て含む形式をNRf形式と呼びます。

NRf形式	意味	例
NR1	整数	+15,-20,25
NR2	固定小数点	+1.23,-4.57,7.89
NR3	浮動小数点	+10.0E-3,-2.3E+3,5E+3

- (3) NR形式は、符号つき数値、符号なし数値の両方を受け付けます。符号のない数値は、正の数と判断します。
- (4) 数値の精度が本器の取扱い範囲を越えた場合は、四捨五入を行います。
- (5) 本器で数値データを受信する場合、NRf形式で受け付けますが、送信はNR1~NR3のいずれかを、コマンドに応じて送信します。

【例1】 NR1形式

:SYSTem:THINOut 10

【例2】 NR2形式

:CONFigure:SAMPlE 0.1

【例3】 NR3形式

:ALARm:ANALog:LEVEl ALM1,CH1_1,+1.0E-3

■文字列データ

- (1) 文字列データは、前後をクォーテーション・マークで囲みます。
- (2) 8ビット ASCII 文字からなるデータです。ただし、コメント設定等の一部コマンドでは、送受信共に Shift_JISとなります。
- (3) 本体で扱えない文字は、アンダーバー「_」に置き換わります。
- (4) クォーテーション・マークとして、送信側はダブル・クォート「"」だけですが、受信側はダブル・クォートとシングル・クォート「'」の両方を受け付けます。

【例】

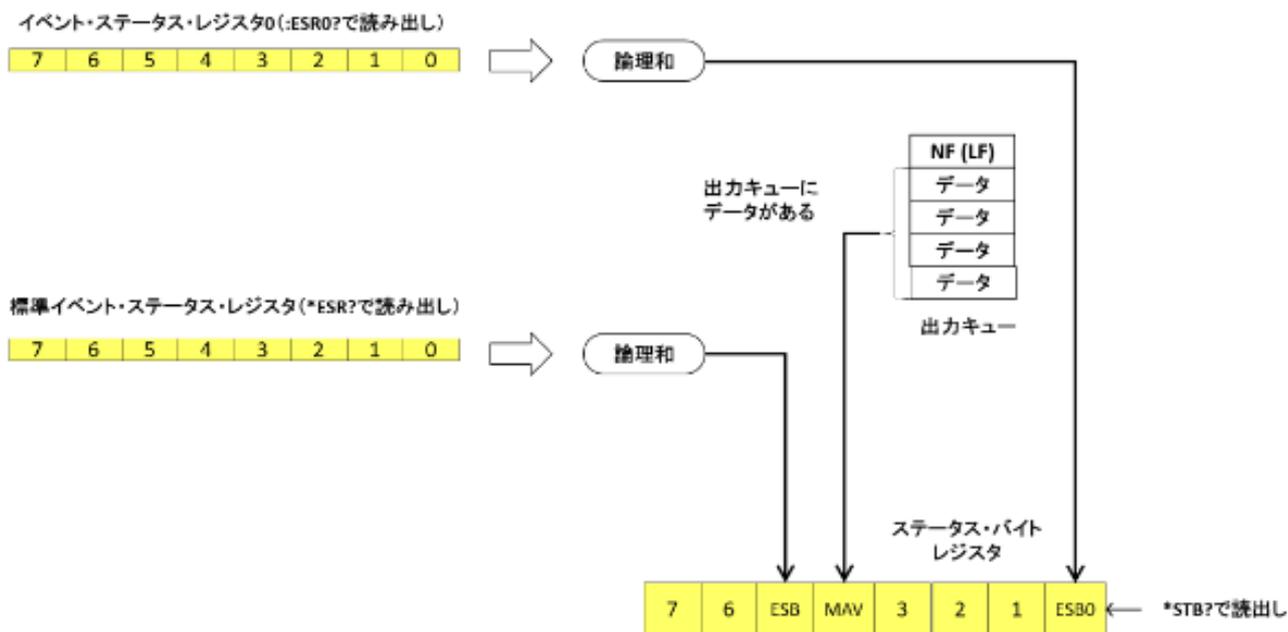
:COMMeNt:TITLe 'HIOKI'
:COMMeNt:TITLe "HIOKI"

2.3 レジスタ

ステータス・バイトとイベント・レジスタ

本器には、下図のようにIEEE488.2で規定されているレジスタ

- 標準イベント・ステータス・レジスタ
- ステータス・バイト・レジスタ
- イベント・ステータス・レジスタ0 を持っています。



ステータス・バイト

ステータス・バイトの各ビットは、そのビットに対応するイベント・レジスタのサマリ（論理和）となっています。

ステータス・バイト	内容
ビット7	未使用:0
ビット6	未使用:0
ビット5 (ESB)	イベント・サマリビット 標準イベント・ステータス・レジスタのサマリを表します。
ビット4 (MAV)	メッセージアベイラブル 出力キューにメッセージがあることを示します。
ビット3	未使用:0
ビット2	未使用:0
ビット1	未使用:0
ビット0 (ESB0)	イベント・サマリ・ビット0 イベント・ステータス・レジスタ0のサマリを表します。

ステータス・バイトの読み出しは、以下のコマンドを使用します。

***STB?**

標準イベント・ステータス・レジスタ (SESR)

このレジスタのサマリは、ステータス・バイトのビット5にセットされます。
標準イベント・ステータス・レジスタの内容がクリアされるのは、以下の場合です。

1. *CLSコマンドを受信
2. *ESR?クエリーで内容が読み出されたとき
3. 電源が再投入されたとき

標準イベント・ステータス・レジスタ (SESR)	内容
ビット7 (PON)	電源再投入 最後にこのレジスタが読み出されてから、電源が切断されたことがある。 電源投入時は1。
ビット6 (URQ)	ユーザ・リクエスト 未使用:0
ビット5 (CME)	コマンドエラー 受信したコマンドに誤りがある。文法上の誤り、意味上の誤り。
ビット4 (EXE)	実行エラー、レンジエラー、モードエラー 機器が実行中のコマンドに誤りがある。
ビット3 (DDE)	機器に依存したエラー 未使用:0
ビット2 (QYE)	クエリーエラー キューが空、データ喪失
ビット1 (RQC)	コントローラ権の要求(使用しない) 未使用:0
ビット0 (OPC)	動作の完了 *OPCコマンドに対してのみセットされる。

標準イベント・ステータス・レジスタの読み出しは以下のコマンドを使用します。

***ESR?**

イベント・ステータス・レジスタ0 (ESR0)

このレジスタのサマリは、ステータス・バイトのビット0にセットされます。
イベント・ステータス・レジスタ0がクリアされるのは以下の場合です。

1. *CLSコマンド受信
2. :ESR0?クエリーで内容が読み出されたとき
3. 電源が再投入されたとき

イベント・ステータス・レジスタ0 (ESR0)	内容
ビット7	未使用:0
ビット6	未使用:0
ビット5	未使用:0
ビット4	未使用:0
ビット3	未使用:0
ビット2	トリガ待ち完了(トリガがかかるとセットされる)
ビット1	START処理完了(STOPするとセットされる)
ビット0	エラー・ワーニングの発生

イベント・ステータス・レジスタ0の読み出しは以下のコマンドを使用します。

:ESR0?

2.4 入出力バッファ

入力バッファ

入力バッファは、200キロバイトです。
受信したメッセージは、このバッファに入力され、順次実行されます。
ただし、:ABORTコマンドは受信した時点で実行されます。

出力キュー

出力キューは、200キロバイトです。
応答メッセージは、ここに溜められ、コントローラから読み出されます。
応答メッセージが200キロバイトを超えた場合はクエリー・エラーとなります。
出力キューがクリアされるのは、以下の場合です。

1. コントローラが読み出したとき
2. 電源再投入
3. 次のメッセージの受信

3

コマンド

3.1 データの種類

記号	意味	例
A\$	文字列データ	OFF,ON
module\$	モジュールデータ	MODULE1 ~ MODULE10
ch\$	チャンネルデータ	CH1_1 ~ CH10_30/PLS1/ALM1 ~ ALM4/W1 ~ W30
pls\$	パルスチャンネルデータ	PLS1
alm\$	警報チャンネルデータ	ALM1 ~ ALM4
w\$	波形演算チャンネルデータ	W1 ~ W30
A,B,C	数値データ	10, -20, 1.5E+05, 0.1
A<NR1>	整数データ	+15, -20, 25
A<NR2>	固定小数点データ	+1.23, -4.57, 7.89
A<NR3>	浮動小数点データ	+10.0E-03, -2.3E+03, 5E+03

3.2 共通コマンド

ステータス・バイトと関連キュー(出力キューを除く)のクリア

*CLS

構文

コマンド

*CLS

解説

ステータス・バイト・レジスタの各ビットに対応したイベント・レジスタをクリアします。
ステータス・バイト・レジスタもクリアされます。

例

*CLS

注記

出力キューはクリアされないので、ステータスバイトのMAV(ビット4)は影響されません。

使用可能条件

標準イベント・ステータス・レジスタ (SESR) の読み出しとクリア

*ESR?

構文

クエリー

*ESR?

応答

A<NR1>

A = 0 ~ 255

解説

標準イベント・ステータス・レジスタ (SESR) の読み出しを行います。
SESRの内容をNR1で返し、SESRをクリアします。

例

*ESR?

(応答) *ESR 0 (ヘッダがONの場合)

注記

使用可能条件

機器ID (識別コード) の問い合わせ

*IDN?

構文

クエリー

*IDN?

応答

A\$,B\$,C\$,D\$

A\$ = メーカー名

B\$ = モデル名

C\$ = 製造番号

D\$ = ソフトウェアバージョン

解説

機器のIDの問い合わせを行います。

例

*IDN?

(応答) *IDN HIOKI,LR8102,221215007,V1.00 (ヘッダがONの場合)

注記

使用可能条件

全動作終了後にSESRのLSBをセットする

*OPC

構文

コマンド

*OPC

解説

送信されたコマンドのうち、*OPCコマンドより前のコマンド処理が終了した時点で、SESR (標準イベント・ステータス・レジスタ) のLSB (ビット0) をセットします。

例

A\$;OPC

(コマンドA\$の処理終了後OPCが実行されます。)

注記

処理終了待機対象コマンドは以下です。

- 測定のストップ(:STOP)
 - ホールドデータの取り込み(:MEMory:GETReal)
 - 機器の初期化(*RST)
- (測定のストップを待機する場合、:STOPコマンドを2回送信する必要があります。)

使用可能条件

全動作終了後にASCIIの1を応答する

*OPC?

構文

クエリー

*OPC?

応答

A<NR1>

A = 1

解説

送信されたコマンドのうち、*OPC?コマンドより前のコマンド処理が終了した時点で、ASCIIの1を応答します。

例

A\$;*OPC?

(コマンドA\$処理終了後、ASCIIの1を応答します。)(応答) *OPC 1 (ヘッダがONの場合)

注記

処理終了待機対象コマンドは以下です。

- 測定のストップ(:STOP)
- ホールドデータの取り込み(:MEMory:GETReal)
- 機器の初期化(*RST)
(測定のストップを待機する場合、:STOPコマンドを2回送信する必要があります。)

使用可能条件

機器のオプション装備の問い合わせ

*OPT?

構文

クエリー

*OPT?

応答

A1,A2,A3,...,A10<NR1>

Ax = 0 ~ 3

解説

搭載されているモジュールの種類を返します。

0: なし

1: M7100 15ch電圧・温度モジュール

3: M7102 30ch電圧・温度モジュール

例

*OPT?

(応答) *OPT 1,1,3,1,1,3,1,1,3,1 (ヘッダがONの場合)

注記

使用可能条件

機器の初期化

*RST

構文

コマンド

*RST

解説

機器を初期化します。

例

*RST

注記

LAN通信に関するものは初期化しません。
*RSTコマンドの処理には時間がかかります。

使用可能条件

ステータス・バイトを読み出す

*STB?

構文

クエリー

*STB?

応答

A<NR1>

A = 0 ~ 255

解説

ステータス・バイトを読み出します。

例

*STB?

(応答) *STB? 128 (ヘッダがONの場合)

注記

使用可能条件

ROM/RAMチェックの結果の問い合わせ

*TST?

構文

クエリー

*TST?

応答

A<NR1>

A = 0, 1

解説

本体のROM/RAMチェックの結果を、NR1数値で返します。

0: 正常

1: 異常

例

*TST?

(応答) *TST 0 (ヘッダがONの場合)

注記

使用可能条件

動作終了後、後続コマンドを実行する

*WAI

構文

コマンド

*WAI

解説

直前の処理終了後、後に続くコマンドを実行します。

例

A\$;*WAI;*IDN?

注記

処理終了待機対象コマンドは以下です。

- 測定のストップ(:STOP)
 - ホールドデータの取り込み(:MEMory:GETReal)
 - 機器の初期化(*RST)
- (測定のストップを待機する場合、:STOPコマンドを2回送信する必要があります。)

使用可能条件

イベント・ステータス・レジスタ0 (ESR0) の読み出しとクリア

:ESR0?

構文

クエリー

:ESR0?

応答

A<NR1>

A = 0 ~ 255

解説

イベント・ステータス・レジスタ0(ESR0)の読み出しを行います。
ESR0の内容をNR1で返し、ESR0をクリアします。

例

:ESR0?

(応答) :ESR0 0 (ヘッダがONの場合)

注記

使用可能条件

3.3 実行処理

強制終了

:ABORT

構文

コマンド

:ABORT

解説

機器の動作を強制的に終了させます。

例

:ABORT

注記

波形取り込み動作が完了しなくても、停止します。

:ABORT;*OPC?のように、他のコマンドと組み合わせて使用することはできません。

使用可能条件

本体エラー番号の問い合わせ

:ERRor?

構文

クエリー

:ERRor?

応答

A\$

A\$ = エラー番号

解説

本体で発生したエラーまたはワーニングの番号を返します。
エラーまたはワーニングについては、本体の取扱説明書を参照してください。

例

:ERRor?

(応答) :ERROR ERR_SY01 (ヘッダがONの場合)

注記

使用可能条件

ヘッダのON/OFFの設定と問い合わせ

:HEADer A\$

構文

コマンド

:HEADer A\$

クエリー

:HEADer?

応答

A\$

A\$ = OFF, ON

解説

クエリー・コマンドの応答データにヘッダをつけるかどうか、ON/OFFで設定します。

クエリー・コマンドの応答データにヘッダをつけるかどうか、ON/OFFで返します。

OFF: クエリー・コマンドの応答データにヘッダをつけません。

ON: クエリー・コマンドの応答データにヘッダをつけます。

例

:HEADer ON

:HEADer?

(応答) :HEADER ON (ヘッダがONの場合)

注記

使用可能条件

スタート処理

:START

構文

コマンド

:START

解説

波形取り込み動作を開始します。

例

:START

注記

使用可能条件

同期運転の設定がセカンダリの際は、測定開始できません。

ストレージ状態の問い合わせ

:STATUS?

構文

クエリー

:STATUS?

応答

A<NR1>

A = 0 ~ 63

解説

ストレージの状態を、NR1数値で返します。例えば、応答が3の場合、ストレージ中、およびスタート中であることを意味します。

Bit0: スタート中

Bit1: ストレージ中

Bit2: トリガ待ち

Bit3: プリトリガ待ち

Bit4: (予約)

Bit5: ファイルアクセス中

例

:STATUS?

(応答) :STATUS 3 (ヘッドがONの場合)

注記

使用可能条件

ストップ処理

:STOP

構文

コマンド

:STOP

解説

測定のストップ。
波形取り込み動作が完了した時点で停止します。

例

:STOP

注記

記録時間の設定により動作が異なります。
:STOPコマンド1回目
記録時間が連続記録の場合:停止しません。
記録時間が時間指定の場合:記録時間分の測定をしたあと停止します。
:STOPコマンド2回目
記録時間が連続記録の場合:測定を停止します。
記録時間が時間指定の場合:測定を停止します。
(:STOPコマンドの処理終了タイミングは、実際に測定が停止したタイミングになります。)

使用可能条件

同期運転の設定がセカンダリの時は、測定終了できません。

正規化処理

:NRMFlag?

構文

クエリー

:NRMFlag?

応答

A<NR1>

A = 0x00000000 ~ 0xffffffff

解説

正規化ビットの読み出しをおこないます。
値をNR1に16進数で返し、ビットをクリアします。

ビット番号	対象	内容
1	その他	以下の分類以外の正規化。
2	記録間隔	記録間隔の正規化。
3	外部入力端子(I/O 3)	外部トリガ設定がONの場合に、外部入力端子(I/O 3)の設定をトリガ入力に変更。
4	記録時間	使用するチャンネル数、記録間隔による記録時間の変更。
5	自動保存の波形データ分割時間	記録間隔による波形データ分割時間の変更。
6	自動保存の数値演算結果分割時間	記録間隔による数値演算結果分割時間の変更。
7	プリトリガ時間	記録間隔によるプリトリガ時間の変更。
8	モジュールのデータ更新間隔	記録間隔によるモジュールのデータ更新間隔の変更。
9	自動保存の波形データ保存形式	使用するチャンネル数、記録間隔による波形データ保存形式の変更。
10	繰り返し記録	インターバルトリガ設定をONにした場合、繰り返し記録設定をONに変更。
11	インターバルトリガ	繰り返し記録設定をOFFにした場合に、インターバルトリガ設定をOFFに変更。
12	自動保存のテキスト形式日付フォーマット	テキスト形式日付フォーマット設定をローカライズの形式に合わせる変更。
13	測定開始時刻、停止時刻	測定開始時刻と停止時刻を変更。

例

:NRMFlag?

(応答) :NRMFLAG 4 (ヘッダがONの場合)

注記

使用可能条件

ホールドしたデータ番号の問い合わせ

:WAITNextsmpl?

構文

クエリー

:WAITNextsmpl?

応答

A<NR1>

A = 0 ~ 最新のストレージ番号

A = -1 (測定中でない場合)

解説

:MEMory:xxFETch?系のコマンドと組み合わせることにより、更新された直後のデータを取得することができます。

組み合わせ可能なホールドデータ取得コマンドは以下の通りです。

:MEMory:VFETch?

:MEMory:AFETch?

:MEMory:BFETch?

:MEMory:TVFETch?

:MEMory:TAFETch?

例

:WAITNextsmpl?

(応答) :WAITNEXTSMPL 1000 (ヘッダがONの場合)

:MEMory:TVFETch? MODULE1

(応答) :MEMORY:TVFETCH +1.00000E-2, +2.00000E-2, +3.00000E-2 (ヘッダがONの場合)

注記

記録間隔が遅い場合(10s~)では、実行エラーとなります。

:MEMory:GETRealを送る必要はありません。

使用可能条件

3.4 測定設定(CONFigure)

自動保存のファイル名タイトルコメント付与の設定と問い合わせ

:CONFigure:ADDCOMMENT A\$

構文

コマンド

:CONFigure:ADDCOMMENT A\$

クエリー

:CONFigure:ADDCOMMENT?

応答

A\$

A\$ = OFF, ON

解説

自動保存ファイル名へのタイトルコメント付与有無を設定します。
現在の自動保存ファイル名へのタイトルコメント付与有無を返します。
OFF: タイトルコメントを付与しない(自動連番を付与する)
ON: タイトルコメントを付与する

例

:CONFigure:ADDCOMMENT ON

:CONFigure:ADDCOMMENT?

(応答) :CONFIGURE:ADDCOMMENT ON (ヘッダがONの場合)

注記

使用可能条件

自動保存のファイル名トリガ日付付与の設定と問い合わせ

:CONFigure:ADDDate A\$

構文

コマンド

:CONFigure:ADDDate A\$

クエリー

:CONFigure:ADDDate?

応答

A\$

A\$ = OFF, ON

解説

自動保存ファイル名へのトリガ日付付与有無を設定します。
現在の自動保存ファイル名へのトリガ日付付与有無を返します。
OFF: トリガ日付を付与しない(自動連番を付与する)
ON: トリガ日付を付与する

例

:CONFigure:ADDDate ON

:CONFigure:ADDDate?

(応答) :CONFIGURE:ADDDATE ON (ヘッダがONの場合)

注記

使用可能条件

自動保存の設定と問い合わせ

:CONFigure:ATSAve A\$("B\$")

構文

コマンド

:CONFigure:ATSAve A\$("B\$")

クエリー

:CONFigure:ATSAve?

応答

A\$("B\$")

A\$ = OFF, BIN, CSV, MF4, MEAS, BIN_MEAS, CSV_MEAS, MF4_MEAS

B\$ = ファイル名 (最大半角8文字分, A\$ = OFF時は省略)

解説

自動保存の波形データ、数値演算結果保存形式と、ファイル名を設定します。
現在の自動保存の波形データ、数値演算結果保存形式と、ファイル名を返します。
OFF: 波形データ:保存しない 数値演算結果:保存しない
BIN: 波形データ:バイナリ形式 数値演算結果:保存しない
CSV: 波形データ:テキスト形式 数値演算結果:保存しない
MF4: 波形データ:MDF4形式 数値演算結果:保存しない
MEAS: 波形データ:保存しない 数値演算結果:テキスト形式
BIN_MEAS: 波形データ:バイナリ形式 数値演算結果:テキスト形式
CSV_MEAS: 波形データ:テキスト形式 数値演算結果:テキスト形式
MF4_MEAS: 波形データ:MDF4形式 数値演算結果:テキスト形式

例

:CONFigure:ATSAve BIN,"AUTO"

:CONFigure:ATSAve?

(応答) :CONFIGURE:ATSAVE BIN,"AUTO" (ヘッダがONの場合)

注記

波形データ保存形式は:CONFigure:SAVEWaveコマンドでも設定可能
数値演算結果保存形式は:CONFigure:SAVECalcコマンドでも設定可能
ファイル名は:CONFigure:FILENAMEコマンドでも設定可能

使用可能条件

自動保存フォルダ分割機能の設定と問い合わせ

:CONFigure:AUTOFolder A\$

構文

コマンド

:CONFigure:AUTOFolder A\$

クエリー

:CONFigure:AUTOFolder?

応答

A\$

A\$ = OFF, DAY, WEEK, MONTH

解説

自動保存ファイル保存先フォルダの分割単位を設定します。
現在の自動保存ファイル保存先フォルダの分割単位を返します。
OFF: 分割なし
DAY: 1日
WEEK: 1週間
MONTH: 1か月

例

:CONFigure:AUTOFolder DAY

:CONFigure:AUTOFolder?

(応答) :CONFIGURE:AUTOFOLDER DAY (ヘッダがONの場合)

注記

使用可能条件

外部サンプリングの記録サンプル数の設定と問い合わせ

:CONFigure:EXTRECSamp A

構文

コマンド

:CONFigure:EXTRECSamp A

クエリー

:CONFigure:EXTRECSamp?

応答

A<NR1>

A = 1 ~ 1000000000

解説

外部サンプリングの記録サンプル数を設定します。
現在の外部サンプリングの記録サンプル数をNR1数値で返します。

例

:CONFigure:EXTRECSamp 10

:CONFigure:EXTRECSamp?

(応答) :CONFIGURE:EXTRECSAMP 10 (ヘッダがONの場合)

注記

使用可能条件

自動保存ファイル名の設定と問い合わせ

`:CONFigure:FILENAME "A$"`

構文

コマンド

`:CONFigure:FILENAME "A$"`

クエリー

`:CONFigure:FILENAME?`

応答

"A\$"

A\$ = ファイル名 (最大半角8文字分)

解説

自動保存時のファイル名を設定します。
現在の自動保存時のファイル名を返します。

例

`:CONFigure:FILENAME "ABC"`

`:CONFigure:FILENAME?`

(応答) `:CONFIGURE:FILENAME "ABC"` (ヘッダがONの場合)

注記

`:CONFigure:ATSAve`コマンドのパラメータB\$でも設定可能
最大文字数以上の文字列を入力した場合、最大文字数以降の文字は入力されません。

使用可能条件

記録時間の設定と問い合わせ

:CONFigure:RECTime day,hour,min,sec

構文

コマンド

:CONFigure:RECTime day,hour,min,sec

クエリー

:CONFigure:RECTime?

応答

day<NR1>,hour<NR1>,min<NR1>,sec<NR1>

day = 0 ~ 500 (日)

hour = 0 ~ 23 (時)

min = 0 ~ 59 (分)

sec = 0 ~ 59 (秒)

解説

記録時間を設定します。

現在の記録時間の設定を、NR1数値で返します。

全パラメータ0の場合は連続記録となります。

例

:CONFigure:RECTime 0,0,0,10

:CONFigure:RECTime?

(応答) :CONFIGURE:RECTIME 0,0,0,10 (ヘッダがONの場合)

注記

使用可能条件

記録間隔の設定と問い合わせ

:CONFigure:SAMPlE A

構文

コマンド

:CONFigure:SAMPlE A

クエリー

:CONFigure:SAMPlE?

応答

A<NR3> (小数点以下1桁)

A = 5.0E-3 ~ 3.6E+3 (秒)

解説

記録間隔を数値(単位s)で設定します。

現在の記録間隔値を数値(単位s)で返します。

例

:CONFigure:SAMPlE 1E-2

:CONFigure:SAMPlE?

(応答) :CONFIGURE:SAMPLE 1.0E-02 (ヘッダがONの場合)

注記

設定にない値を指定した場合、設定しようとした値よりも高いレンジが存在するときは、最も近いレンジに設定されます。

使用可能条件

記録モード(ファンクション)の設定と問い合わせ

:CONFigure:SAMPKind A\$

構文

コマンド

:CONFigure:SAMPKind A\$

クエリー

:CONFigure:SAMPKind?

応答

A\$

A\$ = NORMAl, EXT

解説

記録モード(ファンクション)を設定します。

現在の記録モード(ファンクション)を、文字データで返します。

NORMAl: 内部クロックに同期して、データを記録します。

EXT: 外部クロックに同期して、データを記録します。

例

:CONFigure:SAMPKind NORMAl

:CONFigure:SAMPKind?

(応答) :CONFIGURE:SAMPKIND NORMAL (ヘッダがONの場合)

注記

使用可能条件

自動保存の数値演算結果保存形式の設定と問い合わせ

:CONFigure:SAVECalc A\$

構文

コマンド

:CONFigure:SAVECalc A\$

クエリー

:CONFigure:SAVECalc?

応答

A\$

A\$ = OFF, CSV

解説

自動保存の数値演算結果保存形式を設定します。
現在の自動保存の数値演算結果保存形式の設定を返します。
OFF: 保持しない
CSV: テキスト形式

例

:CONFigure:SAVECalc CSV
:CONFigure:SAVECalc?
(応答) :CONFIGURE:SAVECALC CSV (ヘッダがONの場合)

注記

:CONFigure:ATSAveコマンドのパラメータA\$でも設定可能

使用可能条件

テキストファイル保存時の小数点記号の設定と問い合わせ

:CONFigure:SAVEDeci A\$

構文

コマンド

:CONFigure:SAVEDeci A\$

クエリー

:CONFigure:SAVEDeci?

応答

A\$

A\$ = PERIOD, COMMA

解説

テキスト形式データの小数点記号の設定をします。
現在のテキスト形式データの小数点記号の設定を、文字データで返します。
PERIOD: ピリオド
COMMA: カンマ

例

:CONFigure:SAVEDeci PERIOD
:CONFigure:SAVEDeci?
(応答) :CONFIGURE:SAVEDECI PERIOD (ヘッダがONの場合)

注記

区切り記号の設定がCOMMAの時は、小数点記号をCOMMAに設定できません。

使用可能条件

テキストファイル保存時の日付フォーマットの設定と問い合わせ

`:CONFigure:SAVEFormat A$`

構文

コマンド

`:CONFigure:SAVEFormat A$`

クエリー

`:CONFigure:SAVEFormat?`

応答

A\$

A\$ = COMMENT, SPLITMS

解説

テキスト形式データの日付フォーマットの設定をします。

現在のテキスト形式データの日付フォーマットの設定を、文字データで返します。

COMMENT: コメント(yy-MM-dd hh:mm:ss.0)

SPLITMS: 1秒未満分離(yyyy-MM-dd hh:mm:ss + ms)

例

`:CONFigure:SAVEFormat COMMENT`

`:CONFigure:SAVEFormat?`

(応答) `:CONFIGURE:SAVEFORMAT COMMENT` (ヘッダがONの場合)

注記

使用可能条件

自動保存ファイル分割機能の設定と問い合わせ

:CONFigure:SAVEKind A\$

構文

コマンド

:CONFigure:SAVEKind A\$

クエリー

:CONFigure:SAVEKind?

応答

A\$

A\$ = NORMAl, DIVide, REGUlarly

解説

自動保存ファイルの分割の設定をします

現在の自動保存ファイルの分割の設定を、文字データで返します。

NORMAl: 1回の記録のデータを1つのファイルへ保存

DIVide: 測定開始時から設定時間ごとにデータを分割して別ファイルへ保存

REGUlarly: 基準時刻を設定し、基準時刻から分割時間ごとにデータを別ファイルへ保存

例

:CONFigure:SAVEKind NORMAl

:CONFigure:SAVEKind?

(応答) :CONFIGURE:SAVEKIND NORMAL (ヘッダがONの場合)

注記

使用可能条件

自動保存ファイル分割の分割長の設定と問い合わせ

:CONFigure:SAVELen day,hour,min

構文

コマンド

:CONFigure:SAVELen day,hour,min

クエリー

:CONFigure:SAVELen?

応答

day<NR1>,hour<NR1>,min<NR1>

day = 0 ~ 30 (日)

hour = 0 ~ 23 (時)

min = 0 ~ 59 (分)

解説

自動保存ファイル分割機能の分割長の設定をします。
現在の自動保存ファイル分割の分割長の設定を、NR1数値で返します。

例

:CONFigure:SAVELen 0,0,10

:CONFigure:SAVELen?

(応答) :CONFIGURE:SAVELEN 0,0,10 (ヘッダがONの場合)

注記

使用可能条件

自動保存削除モードの設定と問い合わせ

:CONFigure:SAVEMode A\$

構文

コマンド

:CONFigure:SAVEMode A\$

クエリー

:CONFigure:SAVEMode?

応答

A\$

A\$ = FILEfull, REMove

解説

自動保存機能の削除モードの設定をします。

現在の自動保存機能の削除モードの設定を返します。

FILEfull: OFF:保存先メディアの空き容量が少なくなると保存を終了

REMOve: ON:保存先メディアの容量が少なくなったとき、最も古い波形ファイル(バイナリ、テキスト)を削除して保存

例

:CONFigure:SAVEMode FILEfull

:CONFigure:SAVEMode?

(応答) :CONFIGURE:SAVEMODE FILEFULL (ヘッダがONの場合)

注記

使用可能条件

自動保存優先保存先の設定と問い合わせ

:CONFigure:SAVEPri A\$

構文

コマンド

:CONFigure:SAVEPri A\$

クエリー

:CONFigure:SAVEPri?

応答

A\$

A\$ = SD, USB

解説

自動保存機能の優先保存先を設定をします。

現在の自動保存機能の優先保存先の設定を、文字データで返します。

SD: SDメモリカード

USB: USBメモリ

例

:CONFigure:SAVEPri SD

:CONFigure:SAVEPri?

(応答) :CONFIGURE:SAVEPRI SD (ヘッダがONの場合)

注記

使用可能条件

自動保存ファイル分割の基準時刻の設定と問い合わせ

:CONFigure:SAVEReg hour,min

構文

コマンド

:CONFigure:SAVEReg hour,min

クエリー

:CONFigure:SAVEReg?

応答

hour<NR1>, min<NR1>

hour = 0 ~ 23 (時)

min = 0 ~ 59 (分)

解説

自動保存ファイル分割機能の基準時刻の設定をします。
現在の自動保存ファイル分割の基準時刻の設定を、NR1数値で返します。

例

:CONFigure:SAVEReg 0,0

:CONFigure:SAVEReg?

(応答) :CONFIGURE:SAVEREG 0,0 (ヘッダがONの場合)

注記

使用可能条件

テキストファイル保存時の区切り記号の設定と問い合わせ

:CONFigure:SAVESep A\$

構文

コマンド

:CONFigure:SAVESep A\$

クエリー

:CONFigure:SAVESep?

応答

A\$

A\$ = COMMA, SPACE, TAB, SEMI

解説

テキスト形式データの区切り記号の設定をします。
現在のテキスト形式データの区切り記号の設定を、文字データで返します。
COMMA: カンマ
SPACE: スペース
TAB: タブ
SEMI: セミコロン

例

:CONFigure:SAVESep COMMA
:CONFigure:SAVESep?
(応答) :CONFIGURE:SAVESEP COMMA (ヘッダがONの場合)

注記

小数点記号の設定がCOMMAの時は、区切り記号をCOMMAに設定できません。

使用可能条件

自動保存ファイル分割の分割時間の設定と問い合わせ

:CONFigure:SAVETime A

構文

コマンド

:CONFigure:SAVETime A

クエリー

:CONFigure:SAVETime?

応答

A<NR1>

A = 1,2,5,10,15,20,30,60(1時間),120(2時間),180(3時間),240(4時間),360(6時間),480(8時間),720(12時間),1440(1日) (単位min)

解説

自動保存ファイル分割の分割時間を数値(単位min)で設定します。
現在の自動保存ファイル分割の分割時間を、NR1数値で返します。

例

:CONFigure:SAVETime 1

:CONFigure:SAVETime?

(応答) :CONFIGURE:SAVETIME 1 (ヘッダがONの場合)

注記

設定にない値を指定した場合、設定しようとした値よりも長い分割時間が存在するときは、最も近い分割時間に設定されます。

使用可能条件

自動保存の波形データ保存形式の設定と問い合わせ

:CONFigure:SAVEWave A\$

構文

コマンド

:CONFigure:SAVEWave A\$

クエリー

:CONFigure:SAVEWave?

応答

A\$

A\$ = OFF, BIN, CSV, MF4

解説

自動保存の波形データ保存形式を設定します。
現在の自動保存の波形データ保存形式の設定を返します。
OFF: 保存しない
BIN: バイナリ形式
CSV: テキスト形式
MF4: MF4形式

例

:CONFigure:SAVEWave BIN
:CONFigure:SAVEWave?
(応答) :CONFIGURE:SAVEWAVE BIN (ヘッダがONの場合)

注記

:CONFigure:ATSAveコマンドのパラメータA\$でも設定可能

使用可能条件

測定開始の設定と問い合わせ

:CONFigure:STARt A\$

構文

コマンド

:CONFigure:STARt A\$

クエリー

:CONFigure:STARt?

応答

A\$

A\$ = MANUAL, TIME

解説

測定開始の設定をします。
現在の測定開始の設定を返します。
MANUAL: 手動
TIME: 時刻

例

:CONFigure:STARt MANUAL
:CONFigure:STARt?
(応答) :CONFIGURE:START MANUAL (ヘッダがONの場合)

注記

測定中は、測定開始の設定はできません。

使用可能条件

測定開始時刻の設定と問い合わせ

`:CONFigure:STARTTime year,month,day,hour,minute`

構文

コマンド

`:CONFigure:STARTTime year,month,day,hour,minute`

クエリー

`:CONFigure:STARTTime?`

応答

`year<NR1>,month<NR1>,day<NR1>,hour<NR1>,minute<NR1>`

year = 21 ~ 37 (年)

month = 1 ~ 12 (月)

day = 1 ~ 31 (日)

hour = 0 ~ 23(時)

minute = 0 ~ 59(分)

解説

測定開始時刻の設定をします。

現在の測定開始時刻の設定を返します。

例

`:CONFigure:STARTTime 21,1,2,12,34`

`:CONFigure:STARTTime?`

(応答) `:CONFIGURE:STARTTIME 21,1,2,12,34` (ヘッダがONの場合)

注記

測定中は、測定開始時刻の設定はできません。

測定停止時刻より後の時刻は設定できません。

使用可能条件

測定停止の設定と問い合わせ

:CONFigure:STOP A\$

構文

コマンド

:CONFigure:STOP A\$

クエリー

:CONFigure:STOP?

応答

A\$

A\$ = MANUAL, TIME

解説

測定停止の設定をします。
現在の測定停止の設定を返します。
MANUAL: 手動
TIME: 時刻

例

:CONFigure:STOP MANUAL
:CONFigure:STOP?
(応答) :CONFIGURE:STOP MANUAL (ヘッダがONの場合)

注記

測定中は、測定停止の設定はできません。

使用可能条件

測定停止時刻の設定と問い合わせ

:CONFigure:STOPTime year,month,day,hour,minute

構文

コマンド

:CONFigure:STOPTime year,month,day,hour,minute

クエリー

:CONFigure:STOPTime?

応答

year<NR1>,month<NR1>,day<NR1>,hour<NR1>,minute<NR1>

year = 21 ~ 37 (年)

month = 1 ~ 12 (月)

day = 1 ~ 31 (日)

hour = 0 ~ 23(時)

minute = 0 ~ 59(分)

解説

測定停止時刻の設定をします。

現在の測定停止時刻の設定を返します。

例

:CONFigure:STOPTime 21,1,2,12,34

:CONFigure:STOPTime?

(応答) **:CONFIGURE:STOPTIME 21,1,2,12,34** (ヘッダがONの場合)

注記

測定中は、測定停止時刻の設定はできません。

測定開始時刻より前の時刻は設定できません。

使用可能条件

同期運転の設定と問い合わせ

:CONFigure:SYNC:SET A\$

構文

コマンド

:CONFigure:SYNC:SET A\$

クエリー

:CONFigure:SYNC:SET?

応答

A\$

A\$ = OFF, PRIMary, SECondary

解説

同期運転のプライマリ、セカンダリ、OFFを設定します。

現在の同期運転の設定を返します。

OFF: 同期運転をしない

PRIMary: プライマリ

SECondary: セカンダリ

例

:CONFigure:SYNC:SET PRIMary

:CONFigure:SYNC:SET?

(応答) :CONFIGURE:SYNC:SET PRIMARY (ヘッダがONの場合)

注記

測定中は、同期運転の設定はできません。

使用可能条件

光接続ケーブルの結線の問い合わせ

:CONFigure:SYNC:CHECK?

構文

クエリー

:CONFigure:SYNC:CHECK?

応答

A<NR1>

A = 0 ~ 255

解説

以下結線チェックの結果をNR1数値で返します。

例えば、1が返って来た場合は本器の設定がPRIMARYでは無い事を意味します。

bit0:同期運転の設定がPRIMARYで無い場合にビットが立ちます。

bit4:セカンダリーの台数が9台を超えている可能性がある場合にビットが立ちます。

bit7:光接続ケーブルが外れている可能性がある場合にビットが立ちます。

その他のbitは0に固定です。

例

:CONFigure:SYNC:CHECK?

(応答) :CONFIGURE:SYNC:CHECK 1 (ヘッダがONの場合)

注記

使用可能条件

自動保存の間引き保存データの設定と問い合わせ

`:CONFigure:THINData A$`

構文

コマンド

`:CONFigure:THINData A$`

クエリー

`:CONFigure:THINData?`

応答

A\$

A\$ = INSTANT, STATISTICS

解説

自動保存の間引き保存データを設定します。

現在の自動保存の間引き保存データの設定を返します。

INSTANT: 先頭のデータを保存します。

STATISTICS: 統計データ(最大値、最小値、平均値、先頭のデータ)を保存します。

例

`:CONFigure:THINData INSTANT`

`:CONFigure:THINData?`

(応答) `:CONFIGURE:THINDATA INSTANT` (ヘッダがONの場合)

注記

保存形式をCSVに設定し、間引き保存で2以上にしてください。

使用可能条件

自動保存の間引き保存の設定と問い合わせ

:CONFigure:THINOut A

構文

コマンド

:CONFigure:THINOut A

クエリー

:CONFigure:THINOut?

応答

A<NR1>

A = 1(OFF) ~ 100000

解説

自動保存の間引き保存を設定します。
現在の自動保存の間引き保存の設定をNR1数値で返します。
A=1は間引き保存OFFとなります。

例

:CONFigure:THINOut 1000

:CONFigure:THINOut?

(応答) :CONFIGURE:THINOUT 1000 (ヘッダがONの場合)

注記

保存形式をCSVに設定してください。

使用可能条件

3.5 入力チャネル(MODule)

モジュールのゼロ調整の実行と結果の問い合わせ

:MODule:ADJUST?

構文

クエリー

:MODule:ADJUST?

応答

A<NR1>

A = 0, 1

解説

モジュールのゼロ調整を行います。
現在のモジュールのゼロ調整結果を返します。

A = 0: 成功

A = 1: 失敗

例

:MODule:ADJUST?

(応答) :MODULE:ADJUST 1 (ヘッダがONの場合)

注記

使用可能条件

モジュールのデータ更新間隔の設定と問い合わせ

:MODule:DATARate module\$,A

構文

コマンド

:MODule:DATARate module\$,A

クエリー

:MODule:DATARate? module\$

応答

module\$,A<NR3> (小数点以下1桁)
module\$ = MODULE1 ~ MODULE10
A = 0(自動), 5.0E-3 ~ 1.0E+1 (秒)

解説

指定したモジュールのデータ更新間隔を数値(単位s)で設定します。
指定したモジュールの現在のデータ更新間隔を数値(単位s)で返します。
A = 0の場合、データ更新間隔は自動で設定されます。

例

:MODule:DATARate MODULE1,1E-1
:MODule:DATARate? MODULE1
(応答) :MODULE:DATARATE MODULE1,1.0E-01 (ヘッダがONの場合)

注記

記録間隔以上の値を設定できます。記録間隔が10s以上の場合は10sに固定されます。
設定にない値を指定した場合、設定しようとした値よりも高い更新間隔が存在するときは、最も近い更新間隔に設定されます。

使用可能条件

指定したモジュールのデジタルフィルター値の問い合わせ

`:MODule:DFILter? module$`

構文

クエリー

`:MODule:DFILter? module$`

応答

`module$,A<NR3>`

`module$ = MODULE1 ~ MODULE10`

解説

指定したモジュールのデジタルフィルター値を返します。

例

`:MODule:DFILter? MODULE1`

(応答) `:MODULE:DFILTER MODULE1 2.4E+03` (ヘッダがONの場合)

注記

使用可能条件

電源周波数フィルターの設定と問い合わせ

:MODule:FILTer A\$

構文

コマンド

:MODule:FILTer A\$

クエリー

:MODule:FILTer?

応答

A\$

A\$ = 50HZ, 60HZ

解説

電源周波数フィルターの設定をします。
現在の電源周波数フィルターの設定を返します。

例

:MODule:FILTer 50HZ

:MODule:FILTer?

(応答) :MODULE:FILTER 50HZ (ヘッダがONの場合)

注記

使用可能条件

モジュール情報の問い合わせ

`:MODUle:IDN? module$`

構文

クエリー

`:MODUle:IDN? module$`

応答

module\$,A\$,B\$,C\$,D\$
module\$ = MODULE1 ~ MODULE10
A\$ = 製品形名
B\$ = 製造番号
C\$ = モジュールバージョン
D\$ = モジュールのFPGAバージョン

解説

指定したモジュールの情報を返します。

例

`:MODUle:IDN? MODULE1`
(応答) `:MODULE:IDN MODULE1,M7100,100000000,V 100,V 100` (ヘッダがONの場合)

注記

モジュール未接続の場合は「UNKNOWN」を返します。

使用可能条件

入力チャンネルの入力種類の設定と問い合わせ

`:MODule:INMOde ch$,A$`

構文

コマンド

`:MODule:INMOde ch$,A$`

クエリー

`:MODule:INMOde? ch$`

応答

ch\$,A\$

ch\$ = CH1_1 ~ CH10_30

A\$ = VOLTAGE, TC

解説

指定したチャンネルの入力種類を設定します。

指定したチャンネルの現在の入力種類を返します。

VOLTAGE: 電圧

TC: 熱電対

例

`:MODule:INMOde CH1_1,VOLTAGE`

`:MODule:INMOde? CH1_1`

(応答) `:MODULE:INMODE CH1_1,VOLTAGE` (ヘッダがONの場合)

注記

モジュール		入力種類
M7100	電圧・温度モジュール(15Ch)	VOLTAGE(電圧) TC(熱電対)
M7102	電圧・温度モジュール(30Ch)	VOLTAGE(電圧) TC(熱電対)

使用可能条件

パルスの積算種類の設定と問い合わせ

:MODule:PCOMode pls\$,A\$

構文

コマンド

:MODule:PCOMode pls\$,A\$

クエリー

:MODule:PCOMode? pls\$

応答

pls\$,A\$

pls\$ = PLS1

A\$ = ADD, INST

解説

指定したパルスチャネルの積算種類を設定します。

指定したパルスチャネルの現在の積算種類を返します。

ADD: 加算

INST: 瞬時

例

:MODule:PCOMode PLS1,ADD

:MODule:PCOMode? PLS1

(応答) :MODULE:PCOMODE PLS1,ADD (ヘッダがONの場合)

注記

使用可能条件

パルスの積算開始タイミングの設定と問い合わせ

`:MODule:PCOSart pls$,A$`

構文

コマンド

`:MODule:PCOSart pls$,A$`

クエリー

`:MODule:PCOSart? pls$`

応答

pls\$,A\$
pls\$ = PLS1
A\$ = START, TRIGger

解説

指定したパルスチャネルの積算開始タイミングを設定します。
指定したパルスチャネルの現在の積算開始タイミングを返します。
START: スタート
TRIGger: トリガ

例

`:MODule:PCOSart PLS1,START`
`:MODule:PCOSart? PLS1`
(応答) `:MODULE:PCOSTART PLS1,START` (ヘッダがONの場合)

注記

使用可能条件

パルス回転数測定時の1回転当たりのパルス数の設定と問い合わせ

:MODule:PCOunt pls\$,A

構文

コマンド

:MODule:PCOunt pls\$,A

クエリー

:MODule:PCOunt? pls\$

応答

pls\$,A<NR1>

pls\$ = PLS1

A = 1 ~ 1000

解説

指定したパルスチャネルの1回転当たりのパルス数を設定します。
指定したパルスチャネルの現在の1回転当たりのパルス数を返します。

例

:MODule:PCOunt PLS1,1

:MODule:PCOunt? PLS1

(応答) :MODULE:PCOUNT PLS1,1 (ヘッダがONの場合)

注記

使用可能条件

パルスフィルターの設定と問い合わせ

:MODule:PFILTer pls\$,A\$

構文

コマンド

:MODule:PFILTer pls\$,A\$

クエリー

:MODule:PFILTer? pls\$

応答

pls\$,A\$

pls\$ = PLS1

A\$ = OFF, ON

解説

指定したパルスチャネルのフィルターを設定します。

指定したパルスチャネルのフィルターを返します。

例

:MODule:PFILTer PLS1,ON

:MODule:PFILTer? PLS1

(応答) :MODULE:PFILTER PLS1,ON (ヘッダがONの場合)

注記

使用可能条件

パルスの入力種類の設定と問い合わせ

:MODule:PINMOde pls\$,A\$

構文

コマンド

:MODule:PINMOde pls\$,A\$

クエリー

:MODule:PINMOde? pls\$

応答

pls\$,A\$

pls\$ = PLS1

A\$ = COUNT, REVOLVE, LOGIC

解説

指定したパルスチャネルの入力種類を設定します。

指定したパルスチャネルの現在の入力種類を返します。

COUNT: 積算

REVOLVE: 回転速度

LOGIC: ロジック

例

:MODule:PINMOde PLS1,COUNT

:MODule:PINMOde? PLS1

(応答) :MODULE:PINMODE PLS1,COUNT (ヘッダがONの場合)

注記

使用可能条件

パルス回転数のレンジの設定と問い合わせ

`:MODule:PRANGe pls$,A$`

構文

コマンド

`:MODule:PRANGe pls$,A$`

クエリー

`:MODule:PRANGe? pls$`

応答

`pls$,A$`

`pls$ = PLS1`

`A$ = RPS, RPM`

解説

指定したパルスチャネルの回転数レンジを設定します。

指定したパルスチャネルの回転数レンジを返します。

RPS: 5,000 r/s

RPM: 300,000 r/min

例

`:MODule:PRANGe PLS1,RPM`

`:MODule:PRANGe? PLS1`

(応答) `:MODULE:PRANGe PLS1,RPM` (ヘッダがONの場合)

注記

使用可能条件

パルスの入力種類が回転速度以外の場合、コマンドエラーになります。

パルスリセットの設定と問い合わせ

`:MODule:PRESet pls$,A$`

構文

コマンド

`:MODule:PRESet pls$,A$`

クエリー

`:MODule:PRESet? pls$`

応答

`pls$,A$`

`pls$ = PLS1`

`A$ = OFF, ON`

解説

指定したパルスチャネルのリセットを設定します。
指定したパルスチャネルのリセットの設定を返します。

例

`:MODule:PRESet PLS1,ON`

`:MODule:PRESet? PLS1`

(応答) `:MODULE:PRESET PLS1,ON` (ヘッダがONの場合)

注記

使用可能条件

パルスのスロープの設定と問い合わせ

:MODule:PSLOPe pls\$,A\$

構文

コマンド

:MODule:PSLOPe pls\$,A\$

クエリー

:MODule:PSLOPe? pls\$

応答

pls\$,A\$

pls\$ = PLS1

A\$ = UP, DOWN

解説

指定したパルスチャネルのスロープを設定します。
指定したパルスチャネルのスロープの設定を返します。

例

:MODule:PSLOPe PLS1,UP

:MODule:PSLOPe? PLS1

(応答) :MODULE:PSLOPE PLS1,UP (ヘッダがONの場合)

注記

使用可能条件

パルス回転速度スムージングの設定と問い合わせ

:MODule:PSMooth pls\$,A

構文

コマンド

:MODule:PSMooth pls\$,A

クエリー

:MODule:PSMooth? pls\$

応答

pls\$,A<NR1>
pls\$ = PLS1
A = 1(OFF) ~ 60

解説

指定したパルスチャネルのパルス回転速度スムージングを設定します。
指定したパルスチャネルのパルス回転速度スムージングの設定を返します。
A=1はパルス回転速度スムージングOFFとなります。

例

:MODule:PSMooth PLS1,1
:MODule:PSMooth? PLS1
(応答) :MODULE:PSMOOTH PLS1,1 (ヘッダがONの場合)

注記

使用可能条件

パルスのスレッシュホールドレベルの設定と問い合わせ

`:MODule:PTHRe pls$,A$`

構文

コマンド

`:MODule:PTHRe pls$,A$`

クエリー

`:MODule:PTHRe? pls$`

応答

pls\$,A\$
pls\$ = PLS1
A\$ = 1V, 4V

解説

指定したパルスチャネルのスレッシュホールドレベルを設定します。
指定したパルスチャネルのスレッシュホールドレベルの設定を返します。

例

`:MODule:PTHRe PLS1,1V`
`:MODule:PTHRe? PLS1`
(応答) `:MODULE:PTHRE PLS1,1V` (ヘッダがONの場合)

注記

使用可能条件

入力チャンネルの縦軸レンジの設定と問い合わせ

`:MODule:RANGe ch$,A`

構文

コマンド

`:MODule:RANGe ch$,A`

クエリー

`:MODule:RANGe? ch$`

応答

`ch$,A<NR3>` (小数点以下1桁)

`ch$ = CH1_1 ~ CH10_30`

`A = 縦軸レンジ`

解説

指定したチャンネルの縦軸レンジを設定します。

指定したチャンネルの縦軸レンジを返します。

例

`:MODule:RANGe CH1_1,1E-1`

`:MODule:RANGe? CH1_1`

(応答) `:MODULE:RANGE CH1_1,+1.0E-01` (ヘッダがONの場合)

注記

設定にない値を指定した場合、そのすぐ上のレンジがある場合は、そのレンジになります。

電圧レンジ1-5Vに設定する場合はA=15としてください。

使用可能条件

100°Cレンジ、500°Cレンジの時は、熱電対測定のセンサをBに設定できません。

温度測定 of 接点補償の設定と問い合わせ

:MODule:RJC ch\$,A\$

構文

コマンド

:MODule:RJC ch\$,A\$

クエリー

:MODule:RJC? ch\$

応答

ch\$,A\$

ch\$ = CH1_1 ~ CH10_30

A\$ = INT, EXT

解説

指定したチャンネルの温度測定 of 接点補償を設定します。

指定したチャンネルの温度測定 of 接点補償の設定を返します。

INT: 内部

EXT: 外部

例

:MODule:RJC CH1_1,INT

:MODule:RJC? CH1_1

(応答) :MODULE:RJC CH1_1,INT (ヘッダがONの場合)

注記

使用可能条件

熱電対種類の設定と問い合わせ

`:MODule:SENSor ch$,A$`

構文

コマンド

`:MODule:SENSor ch$,A$`

クエリー

`:MODule:SENSor? ch$`

応答

`ch$,A$`

`ch$ = CH1_1 ~ CH10_30`

`A$ = K, J, E, T, N, R, S, B, C`

解説

指定したチャンネルの温度測定の熱電対を設定します。

指定したチャンネルの温度測定の熱電対を返します。

例

`:MODule:SENSor CH1_1,K`

`:MODule:SENSor? CH1_1`

(応答) `:MODULE:SENSOR CH1_1,K` (ヘッダがONの場合)

注記

100°Cレンジ、500°Cレンジの時は、熱電対測定のセンサをBに設定できません。

使用可能条件

入力チャンネルの測定ON/OFFの設定と問い合わせ

`:MODule:STORe ch$,A$`

構文

コマンド

`:MODule:STORe ch$,A$`

クエリー

`:MODule:STORe? ch$`

応答

`ch$,A$`

`ch$ = CH1_1 ~ CH10_30, PLS1, LOG, ALARM, W1 ~ W30`

`A$ = OFF, ON`

解説

指定したチャンネルの測定ON/OFFを設定します。

指定したチャンネルの測定ON/OFFを返します。

例

`:MODule:STORe CH1_1,ON`

`:MODule:STORe? CH1_1`

(応答) `:MODULE:STORE CH1_1,ON` (ヘッダがONの場合)

注記

パルス入力種類がロジックに設定してあるパルスチャンネルは測定ONにできません。

また、パルス入力種類がロジックに設定されているパルスチャンネルが一つもない場合、ロジックチャンネルは測定ONにできません。

使用可能条件

熱電対の断線検出の設定と問い合わせ

`:MODule:WIRE module$,A$`

構文

コマンド

`:MODule:WIRE module$,A$`

クエリー

`:MODule:WIRE? module$`

応答

module\$,A\$
module\$ = MODULE1 ~ MODULE10
A\$ = OFF, ON

解説

指定したモジュールの熱電対の断線検出を設定します。
指定したモジュールの熱電対の断線検出の設定を返します。

例

`:MODule:WIRE MODULE1,ON`
`:MODule:WIRE? MODULE1`
(応答) `:MODULE:WIRE MODULE1,ON` (ヘッダがONの場合)

注記

使用可能条件

3.6 スケーリング(SCALing)

スケーリングの種類の設定と問い合わせ

`:SCALing:KIND ch$,A$`

構文

コマンド

`:SCALing:KIND ch$,A$`

クエリー

`:SCALing:KIND? ch$`

応答

ch\$,A\$

ch\$ = CH1_1 ~ CH10_30, PLS1

A\$ = RATIO, POINT, SENS

解説

スケーリングの種類を設定します。

現在のスケーリングの種類を返します。

RATIO: 変換比でスケーリングを行います

POINT: 2点を指定してスケーリングを行います

SENS: 感度を用いてスケーリングを行います

例

`:SCALing:KIND CH1_1,POINT`

`:SCALing:KIND? CH1_1`

(応答) `:SCALing:KIND CH1_1,POINT` (ヘッダがONの場合)

注記

モジュールの種類によって設定方法が制限されます。

使用可能条件

スケーリングのオフセットの設定と問い合わせ

:SCALing:OFFSet ch\$,A

構文

コマンド

:SCALing:OFFSet ch\$,A

クエリー

:SCALing:OFFSet? ch\$

応答

ch\$,A<NR3> (小数点以下4桁)
ch\$ = CH1_1 ~ CH10_30, PLS1
A\$ = -9.9999E+09 ~ 9.9999E+09

解説

指定したチャンネルのスケーリングのオフセットを設定します。
指定したチャンネルの現在のスケーリングのオフセットを返します。

例

:SCALing:OFFSet CH1_1,0
:SCALing:OFFSet? CH1_1
(応答) :SCALING:OFFSET CH1_1,+0.0000E+00 (ヘッダがONの場合)

注記

Aで設定できる値は本体に準じます。
本設定により以下の値が変更される可能性があります。

- スケーリングのSCALE UP、LOWの設定
- 数値演算しきい値の設定

使用可能条件

スケーリングのSCALE UP、LOWの設定と問い合わせ

`:SCALing:SCUPLOW ch$,A,B`

構文

コマンド

`:SCALing:SCUPLOW ch$,A,B`

クエリー

`:SCALing:SCUPLOW? ch$`

応答

`ch$,A<NR3>,B<NR3>` (小数点以下4桁)

`ch$ = CH1_1 ~ CH10_30, PLS1`

`A, B = -9.9999E+29 ~ +9.9999E+29`

解説

指定したチャンネルのスケーリングのSCALE UP、LOWの値を設定します。
指定したチャンネルの現在のスケーリングのSCALE UP、LOWの値を返します。

例

`:SCALing:SCUPLOW CH1_1,0.5,-0.5`

`:SCALing:SCUPLOW? CH1_1`

(応答) `:SCALING:SCUPLOW CH1_1,+5.0000E-01,-5.0000E-01` (ヘッダがONの場合)

注記

A = Bは設定できません。
本設定により以下の値が変更される可能性があります。

- スケーリングのオフセットの設定
- スケーリング定格容量の設定
- スケーリング定格出力の設定
- スケーリングの感度の設定
- スケーリングの換算値の設定

使用可能条件

スケーリングの感度の設定と問い合わせ

:SCALing:SENSE ch\$,A

構文

コマンド

:SCALing:SENSE ch\$,A

クエリー

:SCALing:SENSE? ch\$

応答

ch\$,A<NR3> (小数点以下4桁)
ch\$ = CH1_1 ~ CH10_30, PLS1
A = -1.0000E+09 ~ +1.0000E+09

解説

指定したチャンネルのスケーリングの感度を設定します。
指定したチャンネルの現在のスケーリングの感度を返します。

例

:SCALing:SENSE CH1_1,1
:SCALing:SENSE? CH1_1
(応答) :SCALING:SENSE CH1_1,+1.0000E+00 (ヘッダがONの場合)

注記

Aで設定できる値は本体に準じます。
本設定により以下の値が変更される可能性があります。

- スケーリング定格容量の設定
- スケーリング定格出力の設定
- スケーリングのSCALE UP、LOWの設定
- スケーリングの換算値の設定

使用可能条件

スケーリングのON/OFFの設定と問い合わせ

`:SCALing:SET ch$,A$`

構文

コマンド

`:SCALing:SET ch$,A$`

クエリー

`:SCALing:SET? ch$`

応答

ch\$,A

ch\$ = CH1_1 ~ CH10_30, PLS1

A\$ = OFF, ENG, SCI

解説

指定したチャンネルのスケーリングの設定をします。

指定したチャンネルの現在のスケーリングの設定を返します。

OFF: スケーリング機能を使用しません

ENG: スケーリング機能を利用し、小数形式で表示します

SCI: スケーリング機能を利用し、指数形式で表示します

例

`:SCALing:SET CH1_1,ENG`

`:SCALing:SET? CH1_1`

(応答) `:SCALING:SET CH1_1,ENG` (ヘッダがONの場合)

注記

使用可能条件

スケーリングの単位の設定と問い合わせ

:SCALing:UNIT ch\$,"A\$"

構文

コマンド

:SCALing:UNIT ch\$,"A\$"

クエリー

:SCALing:UNIT? ch\$

応答

ch\$,"A\$"

ch\$ = CH1_1 ~ CH10_30, PLS1

A\$ = 単位文字列(全角3文字、半角7文字)

解説

スケーリング単位を設定します。

現在のスケーリング単位を返します。

コマンド中のダブル・クォーテーション (") の代わりに、シングル・クォーテーション (') を使うことができます。

扱うことのできる文字は、本体で入力できる文字と同じですが、特殊文字の入力は下記ようになります。

PC	^2	^3	~u	~o	~e	~c	~+	~,	~;	^^	~~
LR8101 LR8102	2	3	μ	Ω	ε	o	±	'	"	^	~

例

:SCALing:UNIT CH1_1,"mA"

:SCALing:UNIT? CH1_1

(応答) **:SCALING:UNIT CH1_1,"mA"** (ヘッダがONの場合)

注記

最大文字数以上の文字列を入力した場合、最大文字数以降の文字は入力されません。

使用可能条件

スケーリングの換算値の設定と問い合わせ(傾き)

:SCALing:VOLT ch\$,A

構文

コマンド

:SCALing:VOLT ch\$,A

クエリー

:SCALing:VOLT? ch\$

応答

ch\$,A<NR3> (小数点以下4桁)

ch\$ = CH1_1 ~ CH10_30, PLS1

A = -9.9999E+09 ~ +9.9999E+09(パルス積算時は+1.0000E-09 ~ +9.9999E+09)

解説

指定したチャンネルのスケーリング換算値を設定します。

指定したチャンネルの現在のスケーリング換算値を返します。

例

:SCALing:VOLT CH1_1,1

:SCALing:VOLT? CH1_1

(応答) :SCALING:VOLT CH1_1,+1.0000E+00 (ヘッダがONの場合)

注記

Aで設定できる値は本体に準じます。

本設定により以下の値が変更される可能性があります。

- スケーリング定格容量の設定
- スケーリング定格出力の設定
- スケーリングのSCALE UP、LOWの設定
- スケーリングの感度の設定

使用可能条件

スケーリングのVOLT UP、LOWの設定と問い合わせ

:SCALing:VOUPLOW ch\$,A,B

構文

コマンド

:SCALing:VOUPLOW ch\$,A,B

クエリー

:SCALing:VOUPLOW? A\$

応答

ch\$,A<NR3>,B<NR3> (小数点以下4桁)

ch\$ = CH1_1 ~ CH10_30, PLS1

A, B = -9.9999E+29 ~ +9.9999E+29

解説

指定したチャンネルのスケーリングのVOLT UP、LOWの値を設定します。
指定したチャンネルの現在のスケーリングのVOLT UP、LOWの値を返します。

例

:SCALing:VOUPLOW CH1_1,0.05,-0.05

:SCALing:VOUPLOW? CH1_1

(応答) :SCALING:VOUPLOW CH1_1,+5.0000E-02,-5.0000E-02 (ヘッダがONの場合)

注記

A = Bは設定できません。
本設定により以下の値が変更される可能性があります。

- スケーリングのオフセットの設定
- スケーリング定格容量の設定
- スケーリング定格出力の設定
- スケーリングの感度の設定
- スケーリングの換算値の設定

使用可能条件

3.7 コメント(COMMent)

アラームチャネルのコメントの設定と問い合わせ

```
:COMMent:ALMCH alm$,"A$"
```

構文

コマンド

```
:COMMent:ALMCH alm$,"A$"
```

クエリー

```
:COMMent:ALMCH? alm$
```

応答

```
alm$,"A$"
```

```
alm$ = ALM1 ~ ALM4
```

```
A$ = コメント文字列 (全角20文字、半角40文字)
```

解説

指定したアラームチャネルのコメントを設定します。

指定したアラームチャネルの現在のコメントを返します。

コマンド中のダブル・クォーテーション (") の代わりに、シングル・クォーテーション (') を使うことができます。

扱うことのできる文字は、本体で入力できる文字と同じですが、特殊文字の入力は下記ようになります。

PC	^2	^3	~u	~o	~e	~c	~+	~,	~;	^^	~~
LR8101 LR8102	2	3	μ	Ω	ε	o	±	'	"	^	~

例

```
:COMMent:ALMCH ALM1,"ABCDEFGH"
```

```
:COMMent:ALMCH? ALM1
```

```
(応答) :COMMENT:ALMCH ALM1,"ABCDEFGH" (ヘッダがONの場合)
```

注記

最大文字数以上の文字列を入力した場合、最大文字数以降の文字は入力されません。
測定中はアラームチャンネルのコメントの設定は変更できません。

使用可能条件

チャンネルのコメントの設定と問い合わせ

`:COMMeNt:CH ch$,"A$"`

構文

コマンド

`:COMMeNt:CH ch$,"A$"`

クエリー

`:COMMeNt:CH? ch$`

応答

`ch$,"A$"`

`ch$ = CH1_1 ~ CH10_30, PLS1, W1 ~ W30`

`A$ = コメント文字列 (全角20文字、半角40文字)`

解説

指定したチャンネルのコメントを設定します。

指定したチャンネルの現在のコメントを返します。

コマンド中のダブル・クォーテーション (") の代わりに、シングル・クォーテーション (') を使うことができます。

扱うことのできる文字は、本体で入力できる文字と同じですが、特殊文字の入力は下記ようになります。

PC	^2	^3	~u	~o	~e	~c	~+	~,	~;	^^	~~
LR8101 LR8102	2	3	μ	Ω	ε	o	±	'	"	^	~

例

`:COMMeNt:CH CH1_1,"ABCDEFGG"`

`:COMMeNt:CH? CH1_1`

(応答) `:COMMeNt:CH CH1_1,"ABCDEFGG"` (ヘッダがONの場合)

注記

最大文字数以上の文字列を入力した場合、最大文字数以降の文字は入力されません。

測定中はチャンネルのコメントの設定は変更できません。

使用可能条件

タイトルコメントの設定と問い合わせ

:COMMeNt:TITLe "A\$"

構文

コマンド

:COMMeNt:TITLe "A\$"

クエリー

:COMMeNt:TITLe?

応答

"A\$"

A\$ = コメント文字列 (全角20文字、半角40文字)

解説

タイトルコメントを設定します。

現在のタイトルコメントを返します。

コマンド中のダブル・クォーテーション (") の代わりに、シングル・クォーテーション (') を使うことができます。

扱うことのできる文字は、本体で入力できる文字と同じですが、特殊文字の入力は下記ようになります。

PC	^2	^3	~u	~o	~e	~c	~+	~,	~;	^^	~~
LR8101 LR8102	2	3	μ	Ω	ε	ο	±	'	"	^	~

例

:COMMeNt:TITLe "HIOKI"

:COMMeNt:TITLe?

(応答) :COMMeNt:TITLe "HIOKI" (ヘッダがONの場合)

注記

最大文字数以上の文字列を入力した場合、最大文字数以降の文字は入力されません。

測定中はタイトルコメントの設定は変更できません。

使用可能条件

各モジュール識別名の設定と問い合わせ

`:COMMeNt:MODUle module$,"A$"`

構文

コマンド

`:COMMeNt:MODUle module$,"A$"`

クエリー

`:COMMeNt:MODUle? module$`

応答

`module$,"A$"`

`module$ = MODULE1 ~ MODULE10`

`A$ = コメント文字列 (全角8文字、半角16文字)`

解説

指定したモジュールのモジュール識別名を設定します。

指定したモジュールの現在のモジュール識別名を返します。

コマンド中のダブル・クォーテーション (") の代わりに、シングル・クォーテーション (') を使うことができます。

扱うことのできる文字は、本体で入力できる文字と同じですが、特殊文字の入力は下記ようになります。

PC	^2	^3	~u	~o	~e	~c	~+	~,	~;	^^	~~
LR8101 LR8102	2	3	μ	Ω	ε	ο	±	'	"	^	~

例

`:COMMeNt:MODUle MODULE1,"ABCDEFGG"`

`:COMMeNt:MODUle? MODULE1`

(応答) `:COMMENT:MODULE MODULE1,"ABCDEFGG"` (ヘッダがONの場合)

注記

最大文字数以上の文字列を入力した場合、最大文字数以降の文字は入力されません。

測定中は各モジュール識別名の設定は変更できません。

使用可能条件

3.8 トリガ(TRIGger)

開始トリガ検出日の問い合わせ

:TRIGger:DETECTDate?

構文

クエリー

:TRIGger:DETECTDate?

応答

year<NR1>,month<NR1>,day<NR1>

year = 00 ~ 99 (年)

month = 01 ~ 12 (月)

day = 01 ~ 31 (日)

解説

開始トリガ検出日を返します。

例

:TRIGger:DETECTDate?

(応答) :TRIGGER:DETECTDATE 19,12,26 (ヘッダがONの場合)

注記

ストレージデータが存在しない場合は「00,00,00」を応答します。

開始トリガがOFFの場合、測定開始日を応答します。

使用可能条件

開始トリガ検出時間の問い合わせ

:TRIGger:DETECTTime?

構文

クエリー

:TRIGger:DETECTTime?

応答

hour<NR1>,min<NR1>,sec<NR1>,ms<NR1>

hour = 00 ~ 23 (時)

min = 01 ~ 59 (分)

sec = 01 ~ 59 (秒)

ms = 000 ~ 999(ミリ秒)

解説

開始トリガ検出時間を返します。

例

:TRIGger:DETECTTime?

(応答) :TRIGGER:DETECTTIME 01,02,03,004 (ヘッドがONの場合)

注記

ストレージデータが存在しない場合は「00,00,00,000」を応答します。

開始トリガがOFFの場合、測定開始日を応答します。

使用可能条件

マニュアルトリガをかける

:TRIGger:MANUal

構文

コマンド

:TRIGger:MANUal

解説

トリガ待ち状況の時にこのコマンドを実行すると、トリガをかけることができます。

例

:TRIGger:MANUal

注記

使用可能条件

トリガ待ち、プリトリガ待ち状態であること。

繰り返し記録の設定と問い合わせ

:TRIGger:MODE A\$

構文

コマンド

:TRIGger:MODE A\$

クエリー

:TRIGger:MODE?

応答

A\$

A\$ = SINGle, REPEat

解説

繰り返し記録を設定します。

現在の繰り返し記録の設定を返します。

SINGle: 繰り返し記録OFF

REPEat: 繰り返し記録ON

例

:TRIGger:MODE REPEat

:TRIGger:MODE?

(応答) :TRIGGER:MODE REPEAT (ヘッダがONの場合)

注記

設定を変更することで、インターバルトリガの設定が変更される場合があります。

使用可能条件

プリトリガの設定と問い合わせ

:TRIGger:PRETrig day,hour,min,sec

構文

コマンド

:TRIGger:PRETrig day,hour,min,sec

クエリー

:TRIGger:PRETrig?

応答

day<NR1>,hour<NR1>,min<NR1>,sec<NR1>

day = 0 ~ 99(日)

hour = 0 ~ 23(時)

min = 0 ~ 59(分)

sec = 0 ~ 59(秒)

解説

プリトリガを設定します。
現在のプリトリガの設定を返します。

例

:TRIGger:PRETrig 0,0,0,10

:TRIGger:PRETrig?

(応答) :TRIGGER:PRETRIG 0,0,0,10 (ヘッダがONの場合)

注記

記録間隔の設定により、プリトリガの設定が制限される場合があります。

使用可能条件

トリガタイミングが「停止トリガ」以外のとき

トリガ使用の設定と問い合わせ

:TRIGger:SET A\$

構文

コマンド

:TRIGger:SET A\$

クエリー

:TRIGger:SET?

応答

A\$

A\$ = OFF, ON

解説

トリガ使用の設定をします。
現在のトリガ使用の設定を返します。
OFF: 無効
ON: 有効

例

:TRIGger:SET ON
:TRIGger:SET?
(応答) :TRIGGER:SET ON (ヘッダがONの場合)

注記

設定により、外部入力端子3の入力種類が変更される場合があります。

使用可能条件

開始トリガソース間のAND/ORの設定と問い合わせ

:TRIGger:SOURce A\$

構文

コマンド

:TRIGger:SOURce A\$

クエリー

:TRIGger:SOURce?

応答

A\$

A\$ = OR, AND

解説

開始トリガのトリガソース間AND/ORの設定をします。
現在の開始トリガのトリガソース間AND/ORを返します。
AND: 論理積
OR: 論理和

例

:TRIGger:SOURce AND
:TRIGger:SOURce?
(応答) :TRIGGER:SOURCE AND (ヘッダがONの場合)

注記

使用可能条件

停止トリガソース間のAND/ORの設定と問い合わせ

:TRIGger:SSOURce A\$

構文

コマンド

:TRIGger:SSOURce A\$

クエリー

:TRIGger:SSOURce?

応答

A\$

A\$ = OR, AND

解説

停止トリガのトリガソース間AND/ORの設定をします。
現在の停止トリガのトリガソース間AND/ORを返します。
AND: 論理積
OR: 論理和

例

:TRIGger:SSOURce AND
:TRIGger:SSOURce?
(応答) :TRIGGER:SSOURCE AND (ヘッダがONの場合)

注記

使用可能条件

インターバルトリガの設定と問い合わせ

:TRIGger:TIMEr A\$

構文

コマンド

:TRIGger:TIMEr A\$

クエリー

:TRIGger:TIMEr?

応答

A\$

A\$ = OFF, OR, AND

解説

インターバルトリガを設定します。
現在のインターバルトリガの設定を返します。
OFF: 無効
OR: 論理和
AND: 論理積

例

:TRIGger:TIMEr OR
:TRIGger:TIMEr?
(応答) :TRIGGER:TIMER OR (ヘッダがONの場合)

注記

設定を変更することで、繰り返し記録の設定が変更される場合があります。

使用可能条件

トリガタイミングの設定と問い合わせ

:TRIGger:TIMing A\$

構文

コマンド

:TRIGger:TIMing A\$

クエリー

:TRIGger:TIMing?

応答

A\$

A\$ = START, STOP, S_S

解説

トリガタイミングの設定をします。
現在のトリガタイミングを返します。
START: 開始トリガ
STOP: 停止トリガ
S_S: 開始 & 停止トリガ

例

:TRIGger:TIMing START
:TRIGger:TIMing?
(応答) :TRIGGER:TIMING START (ヘッダがONの場合)

注記

設定により、外部入力端子3の入力種類が変更される場合があります。

使用可能条件

インターバルトリガの時間間隔の設定と問い合わせ

:TRIGger:TMINTvl day,hour,min,sec

構文

コマンド

:TRIGger:TMINTvl day,hour,min,sec

クエリー

:TRIGger:TMINTvl?

応答

day<NR1>,hour<NR1>,min<NR1>,sec<NR1>

day = 0 ~ 99(日)

hour = 0 ~ 23(時)

min = 0 ~ 59(分)

sec = 0 ~ 59(秒)

解説

インターバルトリガの時間間隔の設定をします。
現在のインターバルトリガの時間間隔を返します。

例

:TRIGger:TMINTvl 1,20,30,00

:TRIGger:TMINTvl?

(応答) :TRIGGER:TMINTVL 1,20,30,00 (ヘッダがONの場合)

注記

:TRIGger:TMINTvl 0,0,0,0は設定できません。

使用可能条件

トリガの種類の設定と問い合わせ

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:KIND ch\$,A\$(開始トリガ)

:TRIGger:ANALog:STOP:KIND ch\$,A\$(停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:KIND ch\$,A\$(開始トリガ)

:TRIGger:CALCulate:STOP:KIND ch\$,A\$(停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:KIND ch\$,A\$(開始トリガ)

:TRIGger:PULSe:STOP:KIND ch\$,A\$(停止トリガ)

構文

コマンド

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:KIND ch\$,A\$(開始トリガ)

:TRIGger:ANALog:STOP:KIND ch\$,A\$(停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:KIND ch\$,A\$(開始トリガ)

:TRIGger:CALCulate:STOP:KIND ch\$,A\$(停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:KIND ch\$,A\$(開始トリガ)

:TRIGger:PULSe:STOP:KIND ch\$,A\$(停止トリガ)

クエリー

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:KIND? ch\$(開始トリガ)

:TRIGger:ANALog:STOP:KIND? ch\$(停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:KIND? ch\$(開始トリガ)

:TRIGger:CALCulate:STOP:KIND? ch\$(停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:KIND? ch\$(開始トリガ)

:TRIGger:PULSe:STOP:KIND? ch\$(停止トリガ)

応答

[アナログチャンネルトリガ]

ch\$,A\$

[波形演算チャンネルトリガ]

w\$,A\$

[パルスチャンネルトリガ]

pls\$,A\$

ch\$ = CH1_1 ~ CH10_30

w\$ = W1 ~ W30

pls\$ = PLS1

A\$ = OFF, LEVEL, WINDOW

解説

指定したチャンネルのトリガの種類を設定します。

指定したチャンネルの現在のトリガの種類を返します。

OFF

LEVEL: レベル

WINDOW: ウィンドウ

例

```
:TRIGger:ANALog:STARt:KIND CH1_1,LEVEL
```

```
:TRIGger:ANALog:STARt:KIND? CH1_1
```

(応答) :TRIGGER:ANALOG:START:KIND CH1_1,LEVEL (ヘッダがONの場合)

注記

下記の従来コマンドによる設定、問い合わせも可能。

```
:TRIGger:KIND ch$,A$ / :TRIGger:KIND? ch$
```

```
:TRIGger:SKIND ch$,A$ / :TRIGger:SKIND? ch$
```

```
:TRIGger:WKIND w$,A$ / :TRIGger:WKIND? w$
```

```
:TRIGger:SWKIND w$,A$ / :TRIGger:SWKIND? w$
```

```
:TRIGger:PKIND pls$,A$ / :TRIGger:PKIND? pls$
```

```
:TRIGger:SPKIND pls$,A$ / :TRIGger:SPKIND? pls$
```

使用可能条件

トリガレベルの設定と問い合わせ

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:LEVEL ch\$,A (開始トリガ)

:TRIGger:ANALog:STOP:LEVEL ch\$,A (停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:LEVEL w\$,A (開始トリガ)

:TRIGger:CALCulate:STOP:LEVEL w\$,A (停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:LEVEL pls\$,A (開始トリガ)

:TRIGger:PULSe:STOP:LEVEL pls\$,A (停止トリガ)

構文

コマンド

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:LEVEL ch\$,A (開始トリガ)

:TRIGger:ANALog:STOP:LEVEL ch\$,A (停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:LEVEL w\$,A (開始トリガ)

:TRIGger:CALCulate:STOP:LEVEL w\$,A (停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:LEVEL pls\$,A (開始トリガ)

:TRIGger:PULSe:STOP:LEVEL pls\$,A (停止トリガ)

クエリー

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:LEVEL? ch\$ (開始トリガ)

:TRIGger:ANALog:STOP:LEVEL? ch\$ (停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:LEVEL? w\$ (開始トリガ)

:TRIGger:CALCulate:STOP:LEVEL? w\$ (停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:LEVEL? pls\$ (開始トリガ)

:TRIGger:PULSe:STOP:LEVEL? pls\$ (停止トリガ)

応答

[アナログチャンネルトリガ]

ch\$,A<NR3> (小数点以下3桁)

[波形演算チャンネルトリガ]
w\$,A<NR3> (小数点以下4桁)

[パルスチャンネルトリガ]
pls\$,A<NR3> (小数点以下9桁)

ch\$ = CH1_1 ~ CH10_30

w\$ = W1 ~ W30

pls\$ = PLS1

[アナログチャンネルトリガ]

A = 設定可能範囲 : (測定レンジ)×(±1.5倍), 最小分解能 : (測定レンジ)×(1/1000)

[波形演算チャンネルトリガ]

A = -9.9999E+29 ~ 9.9999E+29

[パルスチャンネルトリガ]

A = 0 ~ 1000000000(積算), 0 ~ 15000(r/s), 0 ~ 900000(r/min)

解説

指定したチャンネルのトリガレベルを設定します。
指定したチャンネルの現在のトリガレベルを返します。

例

:TRIGger:ANALog:STARt:LEVEL CH1_1,0.1

:TRIGger:ANALog:STARt:LEVEL? CH1_1

(応答) :TRIGGER:ANALOG:START:LEVEL CH1_1,+1.000E-01 (ヘッダがONの場合)

注記

設定可能上限より大きい値を入力した場合、最大値が入力されます。
設定可能下限より小さい値を入力した場合、最小値が入力されます。
以下の従来コマンドによる設定、問い合わせも可能。

:TRIGger:LEVEL ch\$,A / :TRIGger:LEVEL? ch\$

:TRIGger:SLEVEL ch\$,A / :TRIGger:SLEVEL? ch\$

:TRIGger:WLEVEL w\$,A / :TRIGger:WLEVEL? w\$

:TRIGger:SWLEVEL w\$,A / :TRIGger:SWLEVEL? w\$

:TRIGger:PLEVEL pls\$,A / :TRIGger:PLEVEL? pls\$

:TRIGger:SPLEVEL pls\$,A / :TRIGger:SPLEVEL? pls\$

使用可能条件

設定可能なトリガ種類はレベルトリガです。

ウィンドウトリガ下限レベルの設定と問い合わせ

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:LOWEr ch\$,A (開始トリガ)

:TRIGger:ANALog:STOP:LOWEr ch\$,A (停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:LOWEr w\$,A (開始トリガ)

:TRIGger:CALCulate:STOP:LOWEr w\$,A (停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:LOWEr pls\$,A (開始トリガ)

:TRIGger:PULSe:STOP:LOWEr pls\$,A (停止トリガ)

構文

コマンド

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:LOWEr ch\$,A (開始トリガ)

:TRIGger:ANALog:STOP:LOWEr ch\$,A (停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:LOWEr w\$,A (開始トリガ)

:TRIGger:CALCulate:STOP:LOWEr w\$,A (停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:LOWEr pls\$,A (開始トリガ)

:TRIGger:PULSe:STOP:LOWEr pls\$,A (停止トリガ)

クエリー

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:LOWEr? ch\$ (開始トリガ)

:TRIGger:ANALog:STOP:LOWEr? ch\$ (停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:LOWEr? w\$ (開始トリガ)

:TRIGger:CALCulate:STOP:LOWEr? w\$ (停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:LOWEr? pls\$ (開始トリガ)

:TRIGger:PULSe:STOP:LOWEr? pls\$ (停止トリガ)

応答

[アナログチャンネルトリガ]

ch\$,A<NR3> (小数点以下3桁)

[波形演算チャンネルトリガ]
w\$,A<NR3> (小数点以下4桁)

[パルスチャンネルトリガ]
pls\$,A<NR3> (小数点以下9桁)

ch\$ = CH1_1 ~ CH10_30

w\$ = W1 ~ W30

pls\$ = PLS1

[アナログチャンネルトリガ]

A = 設定可能範囲 : (測定レンジ)×(±1.5倍), 最小分解能 : (測定レンジ)×(1/1000)

[波形演算チャンネルトリガ]

A = -9.9999E+29 ~ 9.9999E+29

[パルスチャンネルトリガ]

A = 0 ~ 1000000000(積算), 0 ~ 15000(r/s), 0 ~ 900000(r/min)

解説

指定したチャンネルのウィンドウトリガ下限レベルを設定します。
指定したチャンネルの現在のウィンドウトリガ下限レベルを返します。

例

:TRIGger:ANALog:STARt:LOWEr CH1_1,-0.5

:TRIGger:ANALog:STARt:LOWEr? CH1_1

(応答) :TRIGGER:ANALOG:START:LOWER CH1_1,-5.000E-01 (ヘッダがONの場合)

注記

ウィンドウトリガ上限レベル以上の値は入力できません。
設定可能上限より大きい値を入力した場合、最大値が入力されます。
設定可能下限より小さい値を入力した場合、最小値が入力されます。
以下の従来コマンドによる設定、問い合わせも可能。

:TRIGger:LOWEr ch\$,A / :TRIGger:LOWEr? ch\$

:TRIGger:SLOWEr ch\$,A / :TRIGger:SLOWEr? ch\$

:TRIGger:WLOWEr w\$,A / :TRIGger:WLOWEr? w\$

:TRIGger:SWLOWEr w\$,A / :TRIGger:SWLOWEr? w\$

:TRIGger:PLowEr pls\$,A / :TRIGger:PLowEr? pls\$

:TRIGger:SPLOWEr pls\$,A / :TRIGger:SPLOWEr? pls\$

使用可能条件

設定可能なトリガ種類はウィンドウトリガです。

トリガサイドの設定と問い合わせ

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:SIDE ch\$,A\$ (開始トリガ)

:TRIGger:ANALog:STOP:SIDE ch\$,A\$ (停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:SIDE w\$,A\$ (開始トリガ)

:TRIGger:CALCulate:STOP:SIDE w\$,A\$ (停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:SIDE pls\$,A\$ (開始トリガ)

:TRIGger:PULSe:STOP:SIDE pls\$,A\$ (停止トリガ)

構文

コマンド

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:SIDE ch\$,A\$ (開始トリガ)

:TRIGger:ANALog:STOP:SIDE ch\$,A\$ (停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:SIDE w\$,A\$ (開始トリガ)

:TRIGger:CALCulate:STOP:SIDE w\$,A\$ (停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:SIDE pls\$,A\$ (開始トリガ)

:TRIGger:PULSe:STOP:SIDE pls\$,A\$ (停止トリガ)

クエリー

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:SIDE? ch\$ (開始トリガ)

:TRIGger:ANALog:STOP:SIDE? ch\$ (停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:SIDE? w\$ (開始トリガ)

:TRIGger:CALCulate:STOP:SIDE? w\$ (停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:SIDE? pls\$ (開始トリガ)

:TRIGger:PULSe:STOP:SIDE? pls\$ (停止トリガ)

応答

[アナログチャンネルトリガ]

ch\$,A\$

[波形演算チャンネルトリガ]

w\$,A\$

[パルスチャンネルトリガ]

pls\$,A\$

ch\$ = CH1_1 ~ CH10_30

w\$ = W1 ~ W30

pls\$ = PLS1

A\$ = IN, OUT

解説

指定したチャンネルのウィンドウトリガのトリガサイドを設定します。

指定したチャンネルの現在のウィンドウトリガのトリガサイドを返します。

IN: ウィンドウイン

OUT: ウィンドウアウト

例

:TRIGger:ANALog:STARt:SIDE CH1_1,IN

:TRIGger:ANALog:STARt:SIDE? CH1_1

(応答) :TRIGGER:ANALOG:START:SIDE CH1_1,IN (ヘッダがONの場合)

注記

下記の従来コマンドによる設定、問い合わせも可能。

:TRIGger:SIDE ch\$,A\$ / :TRIGger:SIDE? ch\$

:TRIGger:SSIDE ch\$,A\$ / :TRIGger:SSIDE? ch\$

:TRIGger:WSIDE w\$,A\$ / :TRIGger:WSIDE? w\$

:TRIGger:SWSIDE w\$,A\$ / :TRIGger:SWSIDE? w\$

:TRIGger:PSIDE pls\$,A\$ / :TRIGger:PSIDE? pls\$

:TRIGger:SPSIDE pls\$,A\$ / :TRIGger:SPSIDE? pls\$

使用可能条件

設定可能なトリガ種類はウィンドウトリガです。

トリガスロープの設定と問い合わせ

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:SLOPe ch\$,A\$ (開始トリガ)

:TRIGger:ANALog:STOP:SLOPe ch\$,A\$ (停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:SLOPe w\$,A\$ (開始トリガ)

:TRIGger:CALCulate:STOP:SLOPe w\$,A\$ (停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:SLOPe pls\$,A\$ (開始トリガ)

:TRIGger:PULSe:STOP:SLOPe pls\$,A\$ (停止トリガ)

構文

コマンド

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:SLOPe ch\$,A\$ (開始トリガ)

:TRIGger:ANALog:STOP:SLOPe ch\$,A\$ (停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:SLOPe w\$,A\$ (開始トリガ)

:TRIGger:CALCulate:STOP:SLOPe w\$,A\$ (停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:SLOPe pls\$,A\$ (開始トリガ)

:TRIGger:PULSe:STOP:SLOPe pls\$,A\$ (停止トリガ)

クエリー

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:SLOPe? ch\$ (開始トリガ)

:TRIGger:ANALog:STOP:SLOPe? ch\$ (停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:SLOPe? w\$ (開始トリガ)

:TRIGger:CALCulate:STOP:SLOPe? w\$ (停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:SLOPe? pls\$ (開始トリガ)

:TRIGger:PULSe:STOP:SLOPe? pls\$ (停止トリガ)

応答

[アナログチャンネルトリガ]

ch\$,A\$

[波形演算チャネルトリガ]

w\$,A\$

[パルスチャネルトリガ]

pls\$,A\$

ch\$ = CH1_1 ~ CH10_30

w\$ = W1 ~ W30

pls\$ = PLS1

A\$ = UP, DOWN

解説

指定したチャネルのレベルトリガのトリガスロープを設定します。

指定したチャネルの現在のレベルトリガのトリガスロープを返します。

UP: 立ち上がり

DOWN: 立ち下がり

例

:TRIGger:ANALog:STARt:SLOPe CH1_1,UP

:TRIGger:ANALog:STARt:SLOPe? CH1_1

(応答) :TRIGGER:ANALOG:START:SLOPE CH1_1,UP (ヘッダがONの場合)

注記

下記の従来コマンドによる設定、問い合わせも可能。

:TRIGger:SLOPe ch\$,A\$ / :TRIGger:SLOPe? ch\$

:TRIGger:SSLOPe ch\$,A\$ / :TRIGger:SSLOPe? ch\$

:TRIGger:WSLOPe w\$,A\$ / :TRIGger:WSLOPe? w\$

:TRIGger:SWSLOPe w\$,A\$ / :TRIGger:SWSLOPe? w\$

:TRIGger:PSLOPe pls\$,A\$ / :TRIGger:PSLOPe? pls\$

:TRIGger:SPSLOPe pls\$,A\$ / :TRIGger:SPSLOPe? pls\$

使用可能条件

設定可能なトリガ種類はレベルトリガです。

ウィンドウトリガ上限レベルの設定と問い合わせ

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:UPPEr ch\$,A (開始トリガ)

:TRIGger:ANALog:STOP:UPPEr ch\$,A (停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:UPPEr w\$,A (開始トリガ)

:TRIGger:CALCulate:STOP:UPPEr w\$,A (停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:UPPEr pls\$,A (開始トリガ)

:TRIGger:PULSe:STOP:UPPEr pls\$,A (停止トリガ)

構文

コマンド

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:UPPEr ch\$,A (開始トリガ)

:TRIGger:ANALog:STOP:UPPEr ch\$,A (停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:UPPEr w\$,A (開始トリガ)

:TRIGger:CALCulate:STOP:UPPEr w\$,A (停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:UPPEr pls\$,A (開始トリガ)

:TRIGger:PULSe:STOP:UPPEr pls\$,A (停止トリガ)

クエリー

[アナログチャンネルトリガ]

:TRIGger:ANALog:STARt:UPPEr? ch\$ (開始トリガ)

:TRIGger:ANALog:STOP:UPPEr? ch\$ (停止トリガ)

[波形演算チャンネルトリガ]

:TRIGger:CALCulate:STARt:UPPEr? w\$ (開始トリガ)

:TRIGger:CALCulate:STOP:UPPEr? w\$ (停止トリガ)

[パルスチャンネルトリガ]

:TRIGger:PULSe:STARt:UPPEr? pls\$ (開始トリガ)

:TRIGger:PULSe:STOP:UPPEr? pls\$ (停止トリガ)

応答

[アナログチャンネルトリガ]

ch\$,A<NR3> (小数点以下3桁)

[波形演算チャンネルトリガ]
w\$,A<NR3> (小数点以下4桁)

[パルスチャンネルトリガ]
pls\$,A<NR3> (小数点以下9桁)

ch\$ = CH1_1 ~ CH10_30

w\$ = W1 ~ W30

pls\$ = PLS1

[アナログチャンネルトリガ]

A = 設定可能範囲 : (測定レンジ)×(±1.5倍), 最小分解能 : (測定レンジ)×(1/1000)

[波形演算チャンネルトリガ]

A = -9.9999E+29 ~ 9.9999E+29

[パルスチャンネルトリガ]

A = 0 ~ 1000000000(積算), 0 ~ 15000(r/s), 0 ~ 900000(r/min)

解説

指定したチャンネルのウィンドウトリガ上限レベルを設定します。
指定したチャンネルの現在のウィンドウトリガ上限レベルを返します。

例

:TRIGger:ANALog:STARt:UPPEr CH1_1,0.5

:TRIGger:ANALog:STARt:UPPEr? CH1_1

(応答) :TRIGGER:ANALOG:START:UPPER CH1_1,+5.000E-01 (ヘッダがONの場合)

注記

ウィンドウトリガ下限レベル以下の値は入力できません。
設定可能上限より大きい値を入力した場合、最大値が入力されます。
設定可能下限より小さい値を入力した場合、最小値が入力されます。
以下の従来コマンドによる設定、問い合わせも可能。

:TRIGger:UPPEr ch\$,A / :TRIGger:UPPEr? ch\$

:TRIGger:SUPPEr ch\$,A / :TRIGger:SUPPEr? ch\$

:TRIGger:WUPPEr w\$,A / :TRIGger:WUPPEr? w\$

:TRIGger:SWUPPEr w\$,A / :TRIGger:SWUPPEr? w\$

:TRIGger:PUPPEr pls\$,A / :TRIGger:PUPPEr? pls\$

:TRIGger:SPUPPEr pls\$,A / :TRIGger:SPUPPEr? pls\$

使用可能条件

設定可能なトリガ種類はウィンドウトリガです。

ロジックトリガのパターンの設定と問い合わせ

:TRIGger:LOGic:START:PATtern "A\$" (開始トリガ)
:TRIGger:LOGic:STOP:PATtern "A\$" (停止トリガ)

構文

コマンド

:TRIGger:LOGic:START:PATtern "A\$" (開始トリガ)
:TRIGger:LOGic:STOP:PATtern "A\$" (停止トリガ)

クエリー

:TRIGger:LOGic:START:PATtern? (開始トリガ)
:TRIGger:LOGic:STOP:PATtern? (停止トリガ)

応答

"A\$"
A\$ = X, 0, 1

解説

ロジックトリガのトリガパターンを設定します。
現在のロジックトリガのトリガパターンを返します。
X: 信号を無視します
0: LOWレベルの信号でトリガがかかります
1: HIGHレベルの信号でトリガがかかります

例

:TRIGger:LOGic:START:PATtern "1"
:TRIGger:LOGic:START:PATtern?
(応答) :TRIGGER:LOGIC:START:PATTERN "1" (ヘッダがONの場合)

注記

以下の従来コマンドによる設定、問い合わせも可能。
:TRIGger:LOGPat "A\$" / :TRIGger:LOGPat?
:TRIGger:SLOGPat "A\$" / :TRIGger:SLOGPat?

使用可能条件

外部トリガの設定と問い合わせ

:TRIGger:EXTernal:START:KIND A\$ (開始トリガ)
:TRIGger:EXTernal:STOP:KIND A\$ (停止トリガ)

構文

コマンド

:TRIGger:EXTernal:START:KIND A\$ (開始トリガ)
:TRIGger:EXTernal:STOP:KIND A\$ (停止トリガ)

クエリー

:TRIGger:EXTernal:START:KIND? (開始トリガ)
:TRIGger:EXTernal:STOP:KIND? (停止トリガ)

応答

A\$
A\$ = OFF, ON

解説

外部トリガの設定をします。
現在の外部トリガの設定を返します。
OFF: 無効
ON: 有効

例

:TRIGger:EXTernal:START:KIND ON
:TRIGger:EXTernal:START:KIND?
(応答) :TRIGGER:EXTERNAL:START:KIND ON (ヘッダがONの場合)

注記

設定により、外部入力端子3の入力種類が変更される場合があります。

使用可能条件

3.9 警報(ALARm)

警報出力端子アクティブ設定と問い合わせ

`:ALARm:ACTive alm$,A$`

構文

コマンド

`:ALARm:ACTive alm$,A$`

クエリー

`:ALARm:ACTive? alm$`

応答

`alm$,A$`

`alm$ = ALM1 ~ ALM4`

`A$ = LOW, HIGH`

`LOW: アクティブLOW`

`HIGH: アクティブHIGH`

解説

警報出力端子のアクティブ設定をします。

現在の警報出力端子のアクティブ設定を返します。

例

`:ALARm:ACTive ALM1,LOW`

`:ALARm:ACTive? ALM1`

(応答) `:ALARM:ACTIVE ALM1,LOW` (ヘッダがONの場合)

注記

使用可能条件

警報履歴の詳細の問い合わせ

:ALARm:ARCD? NO

構文

クエリー

:ALARm:ARCD? NO

応答

NO<NR1>[,ALM\$,CH\$,ERR\$,STR\$,END\$

NO = 警報履歴の番号 (1 ~ 999999)

ALM\$ = ALM1 ~ ALM4

CH\$ = CH1_1 ~ CH10_30, PLS1, LOG W1 ~ W30

ERR\$ = -, BURN_OUT (熱電対断線)

STR\$ = 発生時刻

END\$ = 解除時刻

解説

指定した番号の警報履歴の詳細を返します。

例

:ALARm:ARCD? 1

(応答) :ALARM:ARCD 1,ALM1,CH1_1,-, 20ms, 60ms (ヘッダがONの場合)

注記

発生／解除時刻の形式は時間軸表示の設定に従います。

警報が解除されていない場合、解除時刻は"-"になります。

熱電対断線時のチャンネル番号は"-"になります。

警報履歴の件数までの100件のみ有効です。

使用可能条件

警報履歴の件数が1以上の場合。

警報履歴の件数の問い合わせ

:ALARm:ARCDNum?

構文

クエリー

:ALARm:ARCDNum?

応答

A<NR1>

A = 0 ~ 999999 (0 = 警報無し)

解説

警報履歴の件数を返します。

例

:ALARm:ARCDNum?

(応答) :ALARM:ARCDNUM 10 (ヘッダがONの場合)

注記

使用可能条件

警報音の設定と問い合わせ

:ALARm:BEEP A\$

構文

コマンド

:ALARm:BEEP A\$

クエリー

:ALARm:BEEP?

応答

A\$

A\$ = OFF, ON

解説

警報音の設定をします。
現在の警報音の設定を返します。

例

:ALARm:BEEP ON

:ALARm:BEEP?

(応答) :ALARM:BEEP ON (ヘッダがONの場合)

注記

使用可能条件

断線検出時に警報の設定と問い合わせ

:ALARm:BURN alm\$,A\$

構文

コマンド

:ALARm:BURN alm\$,A\$

クエリー

:ALARm:BURN? alm\$

応答

alm\$,A\$

alm\$ = ALM1 ~ ALM4

A\$ = OFF, ON

解説

断線検出時の警報の設定をします。
現在の断線検出時の警報の設定を返します。

例

:ALARm:BURN ALM1,ON

:ALARm:BURN? ALM1

(応答) :ALARM:BURN ALM1,ON (ヘッダがONの場合)

注記

使用可能条件

警報フィルターの設定と問い合わせ

:ALARm:FILTer alm\$,A

構文

コマンド

:ALARm:FILTer alm\$,A

クエリー

:ALARm:FILTer? alm\$

応答

alm\$,A<NR1>

alm\$ = ALM1 ~ ALM4

A = 0(OFF), 2 ~ 1000

解説

警報フィルターの設定をします。
現在の警報フィルターの設定を返します。

例

:ALARm:FILTer ALM1,2

:ALARm:FILTer? ALM1

(応答) :ALARM:FILTER ALM1,2 (ヘッダがONの場合)

注記

使用可能条件

警報履歴の記録の設定と問い合わせ

:ALARm:HISTory A\$

構文

コマンド

:ALARm:HISTory A\$

クエリー

:ALARm:HISTory?

応答

A\$

A\$ = OFF, ON

解説

警報履歴の記録の設定をします。

現在の警報履歴の記録の設定を返します。

OFF: 開始から100件

ON: 最新の100件

例

:ALARm:HISTory ON

:ALARm:HISTory?

(応答) :ALARM:HISTORY ON (ヘッダがONの場合)

注記

測定中は警報履歴の記録の設定は変更できません。

使用可能条件

警報保持の設定と問い合わせ

:ALARm:HOLD A\$

構文

コマンド

:ALARm:HOLD A\$

クエリー

:ALARm:HOLD?

応答

A\$

A\$ = OFF, ON, CLEAR

解説

警報保持の設定をします。

現在の警報保持の設定を返します。

A\$にCLEARを指定すると、記録を止めずに警報を解除します。

例

:ALARm:HOLD ON

:ALARm:HOLD?

(応答) :ALARM:HOLD ON (ヘッダがONの場合)

注記

使用可能条件

A\$ = CLEARは、警報保持ONの場合のみ使用可能。

警報ソースのAND/ORの設定と問い合わせ

`:ALARm:SOURce alm$,A$`

構文

コマンド

`:ALARm:SOURce alm$,A$`

クエリー

`:ALARm:SOURce? alm$`

応答

`alm$,A$`

`alm$ = ALM1 ~ ALM4`

`A$ = OR, AND`

解説

警報ソースのAND/ORの設定をします。
現在の警報ソースのAND/ORを返します。
AND: 論理積
OR: 論理和

例

`:ALARm:SOURce ALM1,AND`

`:ALARm:SOURce? ALM1`

(応答) `ALARM:SOURCE ALM1,AND` (ヘッダがONの場合)

注記

使用可能条件

警報の種類の設定と問い合わせ

[アナログチャンネル警報]

:ALARm:ANALog:KIND alm\$,ch\$,A\$

[波形演算チャンネル警報]

:ALARm:CALCulate:KIND alm\$,w\$,A\$

[パルスチャンネル警報]

:ALARm:PULSe:KIND alm\$,pls\$,A\$

構文

コマンド

[アナログチャンネル警報]

:ALARm:ANALog:KIND alm\$,ch\$,A\$

[波形演算チャンネル警報]

:ALARm:CALCulate:KIND alm\$,w\$,A\$

[パルスチャンネル警報]

:ALARm:PULSe:KIND alm\$,pls\$,A\$

クエリー

[アナログチャンネル警報]

:ALARm:ANALog:KIND? alm\$,ch\$

[波形演算チャンネル警報]

:ALARm:CALCulate:KIND? alm\$,w\$

[パルスチャンネル警報]

:ALARm:PULSe:KIND? alm\$,pls\$

応答

[アナログチャンネル警報]

alm\$,ch\$,A\$

[波形演算チャンネル警報]

alm\$,w\$,A\$

[パルスチャンネル警報]

alm\$,pls\$,A\$

alm\$ = ALM1 ~ ALM4

ch\$ = CH1_1 ~ CH10_30

w\$ = W1 ~ W30

pls\$ = PLS1

A\$ = OFF, LEVEL, WINDow, SLOPe, SLOPE2

解説

指定したチャンネルの警報の種類を設定します。

指定したチャンネルの現在の警報の種類を返します。

OFF

LEVEL: レベル

WINDow: ウィンドウ

SLOPe: 傾き

SLOPE2: 変化量

例

:ALARm:ANALog:KIND ALM1,CH1_1,LEVEL

:ALARm:ANALog:KIND? ALM1,CH1_1

(応答) :ALARM:ANALOG:KIND ALM1,CH1_1,LEVEL (ヘッダがONの場合)

注記

下記の従来コマンドによる設定、問い合わせも可能。

:ALARm:KIND alm\$,ch\$,A\$ / :ALARm:KIND? alm\$,ch\$

:ALARm:WKIND alm\$,w\$,A\$ / :ALARm:WKIND? alm\$,w\$

:ALARm:PKIND alm\$,pls\$,A\$ / :ALARm:PKIND? alm\$,pls\$

使用可能条件

警報レベルの設定と問い合わせ

[アナログチャンネル警報]

:ALARm:ANALog:LEVEL alm\$,ch\$,A

[波形演算チャンネル警報]

:ALARm:CALCulate:LEVEL alm\$,w\$,A

[パルスチャンネル警報]

:ALARm:PULSe:LEVEL alm\$,pls\$,A

構文

コマンド

[アナログチャンネル警報]

:ALARm:ANALog:LEVEL alm\$,ch\$,A

[波形演算チャンネル警報]

:ALARm:CALCulate:LEVEL alm\$,w\$,A

[パルスチャンネル警報]

:ALARm:PULSe:LEVEL alm\$,pls\$,A

クエリー

[アナログチャンネル警報]

:ALARm:ANALog:LEVEL? alm\$,ch\$

[波形演算チャンネル警報]

:ALARm:CALCulate:LEVEL? alm\$,w\$

[パルスチャンネル警報]

:ALARm:PULSe:LEVEL? alm\$,pls\$

応答

[アナログチャンネル警報]

alm\$,ch\$,A<NR3> (小数点以下3桁)

[波形演算チャンネル警報]

alm\$,w\$,A<NR3> (小数点以下4桁)

[パルスチャンネル警報]

alm\$,pls\$,A<NR3> (小数点以下9桁)

alm\$ = ALM1 ~ ALM4

ch\$ = CH1_1 ~ CH10_30

w\$ = W1 ~ W30

pls\$ = PLS1

[アナログチャンネル警報]

A = 設定可能範囲 : (測定レンジ)×(±1.5倍), 最小分解能 : (測定レンジ)×(1/1000)

[波形演算チャンネル警報]

A = -9.9999E+29 ~ 9.9999E+29

[パルスチャンネル警報]

A = 0 ~ 1000000000(積算), 0 ~ 15000(r/s), 0 ~ 900000(r/min)

解説

指定したチャンネルの警報レベルを設定します。

指定したチャンネルの現在の警報レベルを返します。

例

:ALARm:ANALog:LEVEL ALM1,CH1_1,0.1

:ALARm:ANALog:LEVEL? ALM1,CH1_1

(応答) :ALARM:ANALOG:LEVEL ALM1,CH1_1,+1.000E-01 (ヘッダがONの場合)

注記

設定可能上限より大きい値を入力した場合、最大値が入力されます。

設定可能下限より小さい値を入力した場合、最小値が入力されます。

下記の従来コマンドによる設定、問い合わせも可能。

:ALARm:LEVEL alm\$,ch\$,A\$ / :ALARm:LEVEL? alm\$,ch\$

:ALARm:WLEVEL alm\$,w\$,A / :ALARm:WLEVEL? alm\$,w\$

:ALARm:PLEVEL alm\$,pls\$,A / :ALARm:PLEVEL? alm\$,pls\$

使用可能条件

ウィンドウ警報下限レベルの設定と問い合わせ

[アナログチャンネル警報]

:ALARm:ANALog:LOWEr alm\$,ch\$,A

[波形演算チャンネル警報]

:ALARm:CALCulate:LOWEr alm\$,w\$,A

[パルスチャンネル警報]

:ALARm:PULSe:LOWEr alm\$,pls\$,A

構文

コマンド

[アナログチャンネル警報]

:ALARm:ANALog:LOWEr alm\$,ch\$,A

[波形演算チャンネル警報]

:ALARm:CALCulate:LOWEr alm\$,w\$,A

[パルスチャンネル警報]

:ALARm:PULSe:LOWEr alm\$,pls\$,A

クエリー

[アナログチャンネル警報]

:ALARm:ANALog:LOWEr? alm\$,ch\$

[波形演算チャンネル警報]

:ALARm:CALCulate:LOWEr? alm\$,w\$

[パルスチャンネル警報]

:ALARm:PULSe:LOWEr? alm\$,pls\$

応答

[アナログチャンネル警報]

alm\$,ch\$,A<NR3> (小数点以下3桁)

[波形演算チャンネル警報]

alm\$,w\$,A<NR3> (小数点以下4桁)

[パルスチャンネル警報]

alm\$,pls\$,A<NR3> (小数点以下9桁)

alm\$ = ALM1 ~ ALM4

ch\$ = CH1_1 ~ CH10_30

w\$ = W1 ~ W30

pls\$ = PLS1

[アナログチャンネル警報]

A = 設定可能範囲 : (測定レンジ)×(±1.5倍), 最小分解能 : (測定レンジ)×(1/1000)

[波形演算チャンネル警報]

A = -9.9999E+29 ~ 9.9999E+29

[パルスチャンネル警報]

A = 0 ~ 1000000000(積算), 0 ~ 15000(r/s), 0 ~ 900000(r/min)

解説

指定したチャンネルのウインドウ警報下限レベルを設定します。
指定したチャンネルの現在のウインドウ警報下限レベルを返します。

例

:ALARm:ANALog:LOWEr ALM1,CH1_1,-0.5

:ALARm:ANALog:LOWEr? ALM1,CH1_1

(応答) :ALARM:ANALOG:LOWER ALM1,CH1_1,-5.000E-01 (ヘッダがONの場合)

注記

ウインドウ警報上限レベル以上の値は入力できません。
設定可能上限より大きい値を入力した場合、最大値が入力されます。
設定可能下限より小さい値を入力した場合、最小値が入力されます。
下記の従来コマンドによる設定、問い合わせも可能。

:ALARm:LOWEr alm\$,ch\$,A\$ / :ALARm:LOWEr? alm\$,ch\$

:ALARm:WLOWEr alm\$,w\$,A / :ALARm:WLOWEr? alm\$,w\$

:ALARm:PLOWEr alm\$,pls\$,A / :ALARm:PLOWEr? alm\$,pls\$

使用可能条件

警報サイドの設定と問い合わせ

[アナログチャンネル警報]

:ALARm:ANALog:SIDE alm\$,ch\$,A\$

[波形演算チャンネル警報]

:ALARm:CALCulate:SIDE alm\$,w\$,A\$

[パルスチャンネル警報]

:ALARm:PULSe:SIDE alm\$,pls\$,A\$

構文

コマンド

[アナログチャンネル警報]

:ALARm:ANALog:SIDE alm\$,ch\$,A\$

[波形演算チャンネル警報]

:ALARm:CALCulate:SIDE alm\$,w\$,A\$

[パルスチャンネル警報]

:ALARm:PULSe:SIDE alm\$,pls\$,A\$

クエリー

[アナログチャンネル警報]

:ALARm:ANALog:SIDE? alm\$,ch\$

[波形演算チャンネル警報]

:ALARm:CALCulate:SIDE? alm\$,w\$

[パルスチャンネル警報]

:ALARm:PULSe:SIDE? alm\$,pls\$

応答

[アナログチャンネル警報]

alm\$,ch\$,A\$

[波形演算チャンネル警報]

alm\$,w\$,A\$

[パルスチャンネル警報]

alm\$,pls\$,A\$

alm\$ = ALM1 ~ ALM4

ch\$ = CH1_1 ~ CH10_30

w\$ = W1 ~ W30

pls\$ = PLS1
A\$ = IN, OUT

解説

指定したチャンネルのウインドウ警報の警報サイドを設定します。
指定したチャンネルの現在のウインドウ警報の警報サイドを返します。
IN: ウインドウイン
OUT: ウインドウアウト

例

```
:ALARM:ANALog:SIDE ALM1,CH1_1,IN  
:ALARM:ANALog:SIDE? ALM1,CH1_1  
(応答) :ALARM:ANALOG:SIDE ALM1,CH1_1,IN (ヘッダがONの場合)
```

注記

下記の従来コマンドによる設定、問い合わせも可能。
:ALARM:SIDE alm\$,ch\$,A\$ / :ALARM:SIDE? alm\$,ch\$
:ALARM:WSIDE alm\$,w\$,A\$ / :ALARM:WSIDE? alm\$,w\$
:ALARM:PSIDE alm\$,pls\$,A\$ / :ALARM:PSIDE? alm\$,pls\$

使用可能条件

警報スロープの設定と問い合わせ

[アナログチャンネル警報]

:ALARm:ANALog:SLOPe alm\$,ch\$,A\$

[波形演算チャンネル警報]

:ALARm:CALCulate:SLOPe alm\$,w\$,A\$

[パルスチャンネル警報]

:ALARm:PULSe:SLOPe alm\$,pls\$,A\$

構文

コマンド

[アナログチャンネル警報]

:ALARm:ANALog:SLOPe alm\$,ch\$,A\$

[波形演算チャンネル警報]

:ALARm:CALCulate:SLOPe alm\$,w\$,A\$

[パルスチャンネル警報]

:ALARm:PULSe:SLOPe alm\$,pls\$,A\$

クエリー

[アナログチャンネル警報]

:ALARm:ANALog:SLOPe? alm\$,ch\$

[波形演算チャンネル警報]

:ALARm:CALCulate:SLOPe? alm\$,w\$

[パルスチャンネル警報]

:ALARm:PULSe:SLOPe? alm\$,pls\$

応答

[アナログチャンネル警報]

alm\$,ch\$,A\$

[波形演算チャンネル警報]

alm\$,w\$,A\$

[パルスチャンネル警報]

alm\$,pls\$,A\$

alm\$ = ALM1 ~ ALM4

ch\$ = CH1_1 ~ CH10_30

w\$ = W1 ~ W30

pls\$ = PLS1
A\$ = HIGH, LOW

解説

指定したチャンネルのレベル警報の警報スロープを設定します。
指定したチャンネルの現在のレベル警報の警報スロープを返します。
HIGH: レベル値より上で警報
LOW: レベル値より下で警報

例

```
:ALARM:ANALog:SLOPe ALM1,CH1_1,HIGH  
:ALARM:ANALog:SLOPe? ALM1,CH1_1  
(応答) :ALARM:ANALOG:SLOPE ALM1,CH1_1,HIGH (ヘッダがONの場合)
```

注記

下記の従来コマンドによる設定、問い合わせも可能。
:ALARM:SLOPe alm\$,ch\$,A\$ / :ALARM:SLOPe? alm\$,ch\$
:ALARM:WSLOPe alm\$,w\$,A\$ / :ALARM:WSLOPe? alm\$,w\$
:ALARM:PSLOPe alm\$,pls\$,A\$ / :ALARM:PSLOPe? alm\$,pls\$

使用可能条件

警報変化量の時間の設定と問い合わせ

[アナログチャンネル警報]

:ALARm:ANALog:SLP2:TIME alm\$,ch\$,hour,min,sec,ms

[波形演算チャンネル警報]

ALARm:CALCulate:SLP2:TIME alm\$,w\$,hour,min,sec,ms

[パルスチャンネル警報]

ALARm:PULSe:SLP2:TIME alm\$,pls\$,hour,min,sec,ms

構文

コマンド

[アナログチャンネル警報]

:ALARm:ANALog:SLP2:TIME alm\$,ch\$,hour,min,sec,ms

[波形演算チャンネル警報]

ALARm:CALCulate:SLP2:TIME alm\$,w\$,hour,min,sec,ms

[パルスチャンネル警報]

ALARm:PULSe:SLP2:TIME alm\$,pls\$,hour,min,sec,ms

クエリー

[アナログチャンネル警報]

:ALARm:ANALog:SLP2:TIME? alm\$,ch\$

[波形演算チャンネル警報]

ALARm:CALCulate:SLP2:TIME? alm\$,w\$

[パルスチャンネル警報]

ALARm:PULSe:SLP2:TIME? alm\$,pls\$

応答

[アナログチャンネル警報]

alm\$,ch\$,hour<NR1>,min<NR1>,sec<NR1>,ms<NR1>

[波形演算チャンネル警報]

alm\$,w\$,hour<NR1>,min<NR1>,sec<NR1>,ms<NR1>

[パルスチャンネル警報]

alm\$,pls\$,hour<NR1>,min<NR1>,sec<NR1>,ms<NR1>

alm\$ = ALM1 ~ ALM4

ch\$ = CH1_1 ~ CH10_30

w\$ = W1 ~ W30

pls\$ = PLS1

hour = 0 ~ 23 (時)
min = 0 ~ 59 (分)
sec = 0 ~ 59 (秒)
ms = 0 ~ 999 (ミリ秒)

解説

指定したチャンネルの警報変化量の時間を設定します。
指定したチャンネルの現在の警報変化量の時間を返します。

例

```
:ALARm:ANALog:SLP2:TIME ALM1,CH1_1,0,1,20,30  
:ALARm:ANALog:SLP2:TIME? ALM1,CH1_1  
(応答) :ALARM:ANALOG:SLP2:TIME ALM1,CH1_1,0,1,20,30 (ヘッダがONの場合)
```

注記

「10000×記録時間」以上の時間には設定できません。

使用可能条件

傾き警報時間幅の設定と問い合わせ

[アナログチャンネル警報]

:ALARm:ANALog:STIME alm\$,ch\$,hour,min,sec

[波形演算チャンネル警報]

:ALARm:CALCulate:STIME alm\$,ch\$,hour,min,sec

[パルスチャンネル警報]

:ALARm:PULSe:STIME alm\$,ch\$,hour,min,sec

構文

コマンド

[アナログチャンネル警報]

:ALARm:ANALog:STIME alm\$,ch\$,hour,min,sec

[波形演算チャンネル警報]

:ALARm:CALCulate:STIME alm\$,ch\$,hour,min,sec

[パルスチャンネル警報]

:ALARm:PULSe:STIME alm\$,ch\$,hour,min,sec

クエリー

[アナログチャンネル警報]

:ALARm:ANALog:STIME? alm\$,ch\$

[波形演算チャンネル警報]

:ALARm:CALCulate:STIME? alm\$,w\$

[パルスチャンネル警報]

:ALARm:PULSe:STIME? alm\$,pls\$

応答

[アナログチャンネル警報]

alm\$,ch\$,hour<NR1>,min<NR1>,sec<NR1>

[波形演算チャンネル警報]

alm\$,w\$,hour<NR1>,min<NR1>,sec<NR1>

[パルスチャンネル警報]

alm\$,pls\$,hour<NR1>,min<NR1>,sec<NR1>

alm\$ = ALM1 ~ ALM4

ch\$ = CH1_1 ~ CH10_30

hour = 0 ~ 6 (時)

min = 0 ~ 59 (分)

sec = 0 ~ 59 (秒)

解説

指定したチャンネルの傾き警報時間幅を設定します。
指定したチャンネルの現在の傾き警報時間幅を返します。

例

:ALARm:ANALog:STIMe ALM1,CH1_1,0,1,20

:ALARm:ANALog:STIMe? ALM1,CH1_1

(応答) :ALARM:ANALOG:STIME ALM1,CH1_1,0,1,20 (ヘッダがONの場合)

注記

0,0,0は入力できません。

使用可能条件

ウィンドウ警報上限レベルの設定と問い合わせ

[アナログチャンネル警報]

:ALARm:ANALog:UPPEr alm\$,ch\$,A

[波形演算チャンネル警報]

:ALARm:CALCulate:UPPEr alm\$,w\$,A

[パルスチャンネル警報]

:ALARm:PULSe:UPPEr alm\$,pls\$,A

構文

コマンド

[アナログチャンネル警報]

:ALARm:ANALog:UPPEr alm\$,ch\$,A

[波形演算チャンネル警報]

:ALARm:CALCulate:UPPEr alm\$,w\$,A

[パルスチャンネル警報]

:ALARm:PULSe:UPPEr alm\$,pls\$,A

クエリー

[アナログチャンネル警報]

:ALARm:ANALog:UPPEr? alm\$,ch\$

[波形演算チャンネル警報]

:ALARm:CALCulate:UPPEr? alm\$,w\$

[パルスチャンネル警報]

:ALARm:PULSe:UPPEr? alm\$,pls\$

応答

[アナログチャンネル警報]

alm\$,ch\$,A<NR3> (小数点以下3桁)

[波形演算チャンネル警報]

alm\$,w\$,A<NR3> (小数点以下4桁)

[パルスチャンネル警報]

alm\$,pls\$,A<NR3> (小数点以下9桁)

alm\$ = ALM1 ~ ALM4

ch\$ = CH1_1 ~ CH10_30

w\$ = W1 ~ W30

pls\$ = PLS1

[アナログチャンネル警報]

A = 設定可能範囲 : (測定レンジ)×(±1.5倍), 最小分解能 : (測定レンジ)×(1/1000)

[波形演算チャンネル警報]

A = -9.9999E+29 ~ 9.9999E+29

[パルスチャンネル警報]

A = 0 ~ 1000000000(積算), 0 ~ 15000(r/s), 0 ~ 900000(r/min)

解説

指定したチャンネルのウィンドウ警報上限レベルを設定します。
指定したチャンネルの現在のウィンドウ警報上限レベルを返します。

例

:ALARm:ANALog:UPPEr ALM1,CH1_1,0.5

:ALARm:ANALog:UPPEr? ALM1,CH1_1

(応答) :ALARM:ANALOG:UPPER ALM1,CH1_1,+5.000E-01 (ヘッダがONの場合)

注記

ウィンドウ警報下限レベル以下の値は入力できません。
設定可能上限より大きい値を入力した場合、最大値が入力されます。
設定可能下限より小さい値を入力した場合、最小値が入力されます。
下記の従来コマンドによる設定、問い合わせも可能。

:ALARm:UPPEr alm\$,ch\$,A\$ / :ALARm:UPPEr? alm\$,ch\$

:ALARm:WUPPEr alm\$,w\$,A / :ALARm:WUPPEr? alm\$,w\$

:ALARm:PUPPEr alm\$,pls\$,A / :ALARm:PUPPEr? alm\$,pls\$

使用可能条件

ロジック警報のパターンの設定と問い合わせ

`:ALARm:LOGic:PATtern alm$,"A$"`

構文

コマンド

`:ALARm:LOGic:PATtern alm$,"A$"`

クエリー

`:ALARm:LOGic:PATtern? alm$`

応答

alm\$ = ALM1 ~ ALM4

"A\$"

A\$ = X, 0, 1

解説

ロジック警報の警報パターンを設定します。
現在のロジック警報の警報パターンを返します。
X: 信号を無視します
0: LOWレベルの信号で警報がかかります
1: HIGHレベルの信号で警報がかかります

例

`:ALARm:LOGic:PATtern ALM1,"1"`

`:ALARm:LOGic:PATtern? ALM1`

(応答) `:ALARM:LOGIC:PATTERN ALM1,"1"` (ヘッダがONの場合)

注記

従来コマンド(`:ALARm:LOGPat alm$,"A$"` / `:ALARm:LOGPat? alm$`)による設定、問い合わせも可能。

使用可能条件

3.10 システム(SYSSTem)

手動保存のファイル名タイトルコメント付与の設定と問い合わせ

:SYSSTem:ADDComment A\$

構文

コマンド

:SYSSTem:ADDComment A\$

クエリー

:SYSSTem:ADDComment?

応答

A\$

A\$ = OFF, ON

解説

手動保存のファイル名タイトルコメント付与の設定をします。
現在の手動保存のファイル名タイトルコメント付与の設定を返します。

例

:SYSSTem:ADDComment ON

:SYSSTem:ADDComment?

(応答) :SYSTEM:ADDCOMMENT ON (ヘッダがONの場合)

注記

使用可能条件

手動保存のファイル名トリガ日付付与の設定と問い合わせ

:SYSTem:ADDDate A\$

構文

コマンド

:SYSTem:ADDDate A\$

クエリー

:SYSTem:ADDDate?

応答

A\$

A\$ = OFF, ON

解説

手動保存のファイル名トリガ日付付与の設定をします。
現在の手動保存のファイル名トリガ日付付与の設定を返します。

例

:SYSTem:ADDDate ON

:SYSTem:ADDDate?

(応答) :SYSTEM:ADDDATE ON (ヘッダがONの場合)

注記

使用可能条件

本体とモジュールの調整日の問い合わせ

:SYSTem:ADJDate? A\$

構文

クエリー

:SYSTem:ADJDate? A\$

応答

A\$ = MAIN, MODULE1 ~ MODULE10

Y<NR1>,M<NR1>,D<NR1>

Y = 年

M = 月

D = 日

解説

本体もしくはモジュールの調整日を返します。

MAIN: 本器を最後に調整した年月日を取得します。

MODULE1 ~ MODULE10: 対象モジュールを最後に調整した年月日を取得します。

例

:SYSTem:ADJDate? MAIN

(応答) :SYSTEM:ADJDATE 23,1,25 (ヘッダがONの場合)

注記

対象のモジュールがない場合のY,M,Dは、0,0,0となります。

使用可能条件

ビープ音の設定と問い合わせ

:SYSTem:BEEP A\$

構文

コマンド

:SYSTem:BEEP A\$

クエリー

:SYSTem:BEEP?

応答

A\$

A\$ = OFF, ON

解説

ビープ音の設定をします。
現在のビープ音の設定を返します。

例

:SYSTem:BEEP ON

:SYSTem:BEEP?

(応答) :SYSTEM:BEEP ON (ヘッダがONの場合)

注記

使用可能条件

数値演算結果を保存するファイルの分割の設定と問い合わせ

:SYSTem:CALCSplit A\$

構文

コマンド

:SYSTem:CALCSplit A\$

クエリー

:SYSTem:CALCSplit?

応答

A\$

A\$ = OFF, ON

解説

数値演算結果を保存するファイルの分割の設定をおこないます。

数値演算結果を保存するファイルの分割の設定を返します。

OFF: 数値演算結果を 1 つのファイルに保存します。

ON: 数値演算結果を演算ごと別ファイルに保存します。

例

:SYSTem:CALCSplit ON

:SYSTem:CALCSplit?

(応答) :SYSTEM:CALCSPLIT ON (ヘッダがONの場合)

注記

使用可能条件

ROMRAMチェックの実行と実行結果の問い合わせ

:SYSTem:CHECK

:SYSTem:CHECK:ROMRam

構文

コマンド

:SYSTem:CHECK

:SYSTem:CHECK:ROMRam

クエリー

:SYSTem:CHECK?

:SYSTem:CHECK:ROMRam?

応答

A\$

A\$ = NONE, RUN, PASS, FAIL

解説

ROMRAMチェックを実行します。

ROMRAMチェックの実行結果を返します。

NONE: 未実施

RUN: 実行中

PASS: 正常

FAIL: 異常

例

:SYSTem:CHECK:ROMRam?

(応答) :SYSTEM:CHECK:ROMRAM PASS (ヘッダがONの場合)

注記

ROMRAMチェック完了には約20分かかります。ROMRAMチェック中は、進捗に応じて各LEDが順番に点滅します。

従来コマンド:SYSTem:CHECKによる実行と問い合わせも可能です。

使用可能条件

モジュールチェックの実行と実行結果の問い合わせ

:SYSTem:CHECK:MODule

構文

コマンド

:SYSTem:CHECK:MODule

クエリー

:SYSTem:CHECK:MODule?

応答

m1m2m3m4m5m6m7m8m9m10

m1 ~ m10 = 0, 1, *, -, R

解説

モジュールチェックを実行します。
モジュールチェックの実行結果を返します。
0: 成功
1: 失敗
*: モジュールなし
-: 結果なし
R: 実行中

例

:SYSTem:CHECK:MODule
:SYSTem:CHECK:MODule?
(応答) :SYSTEM:CHECK:MODULE 010101**** (ヘッダがONの場合)

注記

使用可能条件

LAN1チェックの実行と結果の問い合わせ

`:SYSTem:CHECK:IF:LAN1 ip1,ip2,ip3,ip4`

構文

コマンド

`:SYSTem:CHECK:IF:LAN1 ip1,ip2,ip3,ip4`

クエリー

`:SYSTem:CHECK:IF:LAN1?`

応答

A\$

A\$ = NONE, RUN, PASS, FAIL

解説

PING送信先のIPを指定します。

ip1<NR1>,ip2<NR1>,ip3<NR1>,ip4<NR1>

ip1 = 0 ~ 255

ip2 = 0 ~ 255

ip3 = 0 ~ 255

ip4 = 0 ~ 255

LAN1チェックの実行結果を返します。

NONE: 未実施

RUN: 実行中

PASS: 正常

FAIL: 異常

例

`:SYSTem:CHECK:IF:LAN1 192,168,1,1`

`:SYSTem:CHECK:IF:LAN1?`

(応答) `:SYSTEM:CHECK:IF:LAN1 PASS` (ヘッダがONの場合)

注記

使用可能条件

LAN2チェックの実行と結果の問い合わせ

`:SYSTem:CHECK:IF:LAN2 ip1,ip2,ip3,ip4`

構文

コマンド

`:SYSTem:CHECK:IF:LAN2 ip1,ip2,ip3,ip4`

クエリー

`:SYSTem:CHECK:IF:LAN2?`

応答

A\$

A\$ = NONE, RUN, PASS, FAIL

解説

PING送信先のIPを指定します。

ip1<NR1>,ip2<NR1>,ip3<NR1>,ip4<NR1>

ip1 = 0 ~ 255

ip2 = 0 ~ 255

ip3 = 0 ~ 255

ip4 = 0 ~ 255

LAN2チェックの実行結果を返します。

NONE: 未実施

RUN: 実行中

PASS: 正常

FAIL: 異常

例

`:SYSTem:CHECK:IF:LAN2 192,168,1,1`

`:SYSTem:CHECK:IF:LAN2?`

(応答) `:SYSTEM:CHECK:IF:LAN2 PASS` (ヘッダがONの場合)

注記

使用可能条件

LR8102のみ使用可

メディアチェックの実行と実行結果の問い合わせ

:SYSTem:CHECK:MEDia:SD
:SYSTem:CHECK:MEDia:USB

構文

コマンド

:SYSTem:CHECK:MEDia:SD
:SYSTem:CHECK:MEDia:USB

クエリー

:SYSTem:CHECK:MEDia:SD?
:SYSTem:CHECK:MEDia:USB?

応答

A\$
A\$ = NONE, RUN, PASS, FAIL, ERR

解説

メディアチェックを実行します。
メディアチェックの実行結果を返します。
NONE: 未実施
RUN: 実行中
PASS: 正常
FAIL: 異常
ERR: エラー

例

:SYSTem:CHECK:MEDia:SD?
(応答) :SYSTEM:CHECK:MEDIA:SD PASS (ヘッダがONの場合)

注記

使用可能条件

本体とモジュールの校正日の問い合わせ

:SYSTem:CLBDate? A\$

構文

クエリー

:SYSTem:CLBDate? A\$

応答

A\$ = MAIN, MODULE1 ~ MODULE10

Y<NR1>,M<NR1>,D<NR1>

Y = 年 (Y = 0 ~ 99)

M = 月 (M = 1 ~ 12)

D = 日 (D = 1 ~ 31)

解説

本体もしくはモジュールの校正日を返します。

MAIN: 本器を最後に校正した年月日を取得します。

MODULE1 ~ MODULE10: 対象モジュールを最後に校正した年月日を取得します。

例

:SYSTem:CLBDate? MODULE1

(応答) :SYSTEM:CLBDATE 23,12,31 (ヘッダがONの場合)

注記

実装されていないモジュールの場合は、0,0,0を返します。

使用可能条件

動作クロックの設定と問い合わせ

:SYSTem:CLOCK:OUT A\$

構文

コマンド

:SYSTem:CLOCK:OUT A\$

クエリー

:SYSTem:CLOCK:OUT?

応答

A\$

A\$ = OFF, ON, PRECISION

解説

動作クロックの設定をします。

現在の動作クロックの設定を返します。

OFF: SMPL端子から確認用クロックを出力しません。

ON: SMPL端子から時計精度の確認用クロックを出力します。

PRECISION: SMPL端子から時間精度の確認用クロックを出力します。

例

:SYSTem:CLOCK:OUT ON

:SYSTem:CLOCK:OUT?

(応答) :SYSTEM:CLOCK:OUT ON (ヘッダがONの場合)

注記

SMPL端子から出力されるクロック周波数が次の範囲に入っていることを確認してください。

ONの場合: $32.768\text{kHz} \pm 0.000379\text{kHz}$

PRECISIONの場合: $10.000\text{kHz} \pm 0.0000231\text{kHz}$

使用可能条件

LAN/LAN2のDHCPの設定と問い合わせ

```
:SYSTem:COMMunicate:LAN:DHCP A$  
:SYSTem:COMMunicate:LAN2:DHCP A$
```

構文

コマンド

```
:SYSTem:COMMunicate:LAN:DHCP A$  
:SYSTem:COMMunicate:LAN2:DHCP A$
```

クエリー

```
:SYSTEM:COMMunicate:LAN:DHCP?  
:SYSTEM:COMMunicate:LAN2:DHCP?  
:SYSTem:COMMunicate:LAN:DHCP:PREParation?  
:SYSTem:COMMunicate:LAN2:DHCP:PREParation?
```

応答

A\$ = OFF, ON

解説

DHCPの設定は:SYSTem:COMMunicate:LAN:UPDate/:SYSTem:COMMunicate:LAN2:UPDateコマンド実行後に有効になります。

DHCPの設定をします。

現在のDHCPの設定を返します。

OFF: DHCPを無効にします。

ON: DHCPを有効にします。

PREParationクエリーではLAN/LAN2の設定を反映する前の設定を返します。

例

```
:SYSTem:COMMunicate:LAN:DHCP ON  
:SYSTem:COMMunicate:LAN:UPDate  
:SYSTem:COMMunicate:LAN:DHCP?  
(応答) :SYSTEM:COMMUNICATE:LAN:DHCP ON (ヘッダがONの場合)
```

注記

DHCPサーバーをONにすると、IPアドレスやサブネットマスクを自動で取得できます。

使用可能条件

測定中は設定の変更はできません。

LAN/LAN2のホスト名の設定と問い合わせ

```
:SYSTem:COMMunicate:LAN:HOSTname "A$"  
:SYSTem:COMMunicate:LAN2:HOSTname "A$"
```

構文

コマンド

```
:SYSTem:COMMunicate:LAN:HOSTname "A$"  
:SYSTem:COMMunicate:LAN2:HOSTname "A$"
```

クエリー

```
:SYSTEM:COMMunicate:LAN:HOSTname?  
:SYSTEM:COMMunicate:LAN2:HOSTname?  
:SYSTem:COMMunicate:LAN:HOSTname:PREParation?  
:SYSTem:COMMunicate:LAN2:HOSTname:PREParation?
```

応答

```
"A$"  
A$ = ホスト名文字列(半角12文字まで)
```

解説

ホスト名の設定は:SYSTem:COMMunicate:LAN:UPDate/:SYSTem:COMMunicate:LAN2:UPDateコマンド実行後に有効になります。

ホスト名の設定をします。

現在のホスト名の設定を返します。

PREParationクエリーではLAN/LAN2の設定を反映する前の設定を返します。

例

```
:SYSTem:COMMunicate:LAN:HOSTname "LR8100"  
:SYSTem:COMMunicate:LAN:UPDate  
:SYSTem:COMMunicate:LAN:HOSTname?  
(応答) :SYSTEM:COMMUNICATE:LAN:HOSTNAME "LR8100" (ヘッダがONの場合)
```

注記

使用可能条件

測定中は設定の変更はできません。

LAN/LAN2のIPアドレスの設定と問い合わせ

```
:SYSTem:COMMunicate:LAN:IPADdress ip1,ip2,ip3,ip4  
:SYSTem:COMMunicate:LAN2:IPADdress ip1,ip2,ip3,ip4
```

構文

コマンド

```
:SYSTem:COMMunicate:LAN:IPADdress ip1,ip2,ip3,ip4  
:SYSTem:COMMunicate:LAN2:IPADdress ip1,ip2,ip3,ip4
```

クエリー

```
:SYSTEM:COMMunicate:LAN:IPADdress?  
:SYSTEM:COMMunicate:LAN2:IPADdress?  
:SYSTEM:COMMunicate:LAN:IPADdress:PREParation?  
:SYSTEM:COMMunicate:LAN2:IPADdress:PREParation?
```

応答

```
ip1<NR1>,ip2<NR1>,ip3<NR1>,ip4<NR1>  
ip1 = 0 ~ 255  
ip2 = 0 ~ 255  
ip3 = 0 ~ 255  
ip4 = 0 ~ 255
```

解説

IPアドレスの設定は:SYSTem:COMMunicate:LAN:UPDate/:SYSTem:COMMunicate:LAN2:UPDate コマンド実行後に有効になります。

IPアドレスの設定をします。

現在のIPアドレスの設定を返します。

PREParationクエリーではLAN/LAN2の設定を反映する前の設定を返します。

例

```
:SYSTem:COMMunicate:LAN:IPADdress 192,168,1,10  
:SYSTem:COMMunicate:LAN:UPDate  
:SYSTem:COMMunicate:LAN:IPADdress?  
(応答) :SYSTEM:COMMUNICATE:LAN:IPADDRESS 192,168,1,10 (ヘッダがONの場合)
```

注記

他の機器と重ならないように設定してください。

DHCPサーバーをONにすると、IPアドレスを自動で設定します。

使用可能条件

測定中は設定の変更はできません。

LAN/LAN2のサブネットマスクの設定と問い合わせ

```
:SYSTem:COMMunicate:LAN:SMASk mask1,mask2,mask3,mask4
```

```
:SYSTem:COMMunicate:LAN2:SMASk mask1,mask2,mask3,mask4
```

構文

コマンド

```
:SYSTem:COMMunicate:LAN:SMASk mask1,mask2,mask3,mask4
```

```
:SYSTem:COMMunicate:LAN2:SMASk mask1,mask2,mask3,mask4
```

クエリー

```
:SYSTEM:COMMunicate:LAN:SMASk?
```

```
:SYSTEM:COMMunicate:LAN2:SMASk?
```

```
:SYSTEM:COMMunicate:LAN:SMASk:PREParation?
```

```
:SYSTEM:COMMunicate:LAN2:SMASk:PREParation?
```

応答

```
mask1<NR1>,mask2<NR1>,mask3<NR1>,mask4<NR1>
```

```
mask1 = 0 ~ 255
```

```
mask2 = 0 ~ 255
```

```
mask3 = 0 ~ 255
```

```
mask4 = 0 ~ 255
```

解説

サブネットマスクの設定

は:SYSTem:COMMunicate:LAN:UPDate/:SYSTem:COMMunicate:LAN2:UPDateコマンド実行後に有効になります。

サブネットマスクの設定をします。

現在のサブネットマスクの設定を返します。

PREParationクエリーではLAN/LAN2の設定を反映する前の設定を返します。

例

```
:SYSTem:COMMunicate:LAN:SMASk 255,255,255,0
```

```
:SYSTem:COMMunicate:LAN:UPDate
```

```
:SYSTem:COMMunicate:LAN:SMASk?
```

```
(応答) :SYSTEM:COMMUNICATE:LAN:SMASK 255,255,255,0 (ヘッダがONの場合)
```

注記

同じネットワーク内の機器のサブネットマスクと同じように設定してください。
DHCPサーバーをONにすると、サブネットマスクを自動で設定します。

使用可能条件

測定中は設定の変更はできません。

LAN/LAN2のデフォルトゲートウェイの設定と問い合わせ

```
:SYSTem:COMMunicate:LAN:GATeway ip1,ip2,ip3,ip4  
:SYSTem:COMMunicate:LAN2:GATeway ip1,ip2,ip3,ip4
```

構文

コマンド

```
:SYSTem:COMMunicate:LAN:GATeway ip1,ip2,ip3,ip4  
:SYSTem:COMMunicate:LAN2:GATeway ip1,ip2,ip3,ip4
```

クエリー

```
:SYSTEM:COMMunicate:LAN:GATeway?  
:SYSTEM:COMMunicate:LAN2:GATeway?  
:SYSTEM:COMMunicate:LAN:GATeway:PREParation?  
:SYSTEM:COMMunicate:LAN2:GATeway:PREParation?
```

応答

```
ip1<NR1>,ip2<NR1>,ip3<NR1>,ip4<NR1>  
ip1 = 0 ~ 255  
ip2 = 0 ~ 255  
ip3 = 0 ~ 255  
ip4 = 0 ~ 255
```

解説

デフォルトゲートウェイの設定

は:SYSTem:COMMunicate:LAN:UPDate/:SYSTem:COMMunicate:LAN2:UPDateコマンド実行後に有効になります。

デフォルトゲートウェイの設定をします。

現在のデフォルトゲートウェイの設定を返します。

PREParationクエリーではLAN/LAN2の設定を反映する前の設定を返します。

例

```
:SYSTem:COMMunicate:LAN:GATeway 192,168,1,12  
:SYSTem:COMMunicate:LAN:UPDate  
:SYSTem:COMMunicate:LAN:GATeway?  
(応答) :SYSTEM:COMMUNICATE:LAN:GATEWAY 192,168,1,12 (ヘッダがONの場合)
```

注記

DHCPサーバーをONにすると、デフォルトゲートウェイを自動で設定します。

使用可能条件

測定中は設定の変更はできません。

LAN/LAN2のポート番号の設定と問い合わせ

:SYSTem:COMMunicate:LAN:CONTRol no
:SYSTem:COMMunicate:LAN2:CONTRol no

構文

コマンド

:SYSTem:COMMunicate:LAN:CONTRol no
:SYSTem:COMMunicate:LAN2:CONTRol no

クエリー

:SYSTEM:COMMunicate:LAN:CONTRol?
:SYSTEM:COMMunicate:LAN2:CONTRol?
:SYSTEM:COMMunicate:LAN:CONTRol:PREPARation?
:SYSTEM:COMMunicate:LAN2:CONTRol:PREPARation?

応答

no<NR1>
no = 1020 ~ 65520

解説

ポート番号の設定は:SYSTem:COMMunicate:LAN:UPDate/:SYSTem:COMMunicate:LAN2:UPDate コマンド実行後に有効になります。

ポート番号の設定をします。

現在のポート番号の設定を返します。

PREPARationクエリーではLAN/LAN2の設定を反映する前の設定を返します。

例

:SYSTem:COMMunicate:LAN:CONTRol 8800
:SYSTem:COMMunicate:LAN:UPDate
:SYSTem:COMMunicate:LAN:CONTRol?
(応答) :SYSTEM:COMMUNICATE:LAN:CONTROL 8800 (ヘッダがONの場合)

注記

(LAN1)

例えば、8800から8809のどれを設定しても設定は8800が返されます (下1桁 0:ロガーユーティリティ, 2:通信コマンド, 5:XCP on Ethernet)。

(LAN2)

例えば、8800から8809のどれを設定しても設定は8800が返されます (下1桁 1:UDP出力, 5:XCP on Ethernet)。

使用可能条件

測定中は設定の変更はできません。

LAN/LAN2のDNSの設定と問い合わせ

`:SYSTem:COMMunicate:LAN:DNS ip1,ip2,ip3,ip4`

`:SYSTem:COMMunicate:LAN2:DNS ip1,ip2,ip3,ip4`

構文

コマンド

`:SYSTem:COMMunicate:LAN:DNS ip1,ip2,ip3,ip4`

`:SYSTem:COMMunicate:LAN2:DNS ip1,ip2,ip3,ip4`

クエリー

`:SYSTEM:COMMunicate:LAN:DNS?`

`:SYSTEM:COMMunicate:LAN2:DNS?`

`:SYSTEM:COMMunicate:LAN:DNS:PREParation?`

`:SYSTEM:COMMunicate:LAN2:DNS:PREParation?`

応答

`ip1<NR1>,ip2<NR1>,ip3<NR1>,ip4<NR1>`

`ip1 = 0 ~ 255`

`ip2 = 0 ~ 255`

`ip3 = 0 ~ 255`

`ip4 = 0 ~ 255`

解説

DNSの設定は:`:SYSTem:COMMunicate:LAN:UPDate/`:`:SYSTem:COMMunicate:LAN2:UPDate` コマンド実行後に有効になります。

DNSの設定をします。

現在のDNSの設定を返します。

PREParationクエリーではLAN/LAN2の設定を反映する前の設定を返します。

例

`:SYSTem:COMMunicate:LAN:DNS 192,168,1,13`

`:SYSTem:COMMunicate:LAN:UPDate`

`:SYSTem:COMMunicate:LAN:DNS?`

(応答) `:SYSTEM:COMMUNICATE:LAN:DNS 192,168,1,13` (ヘッダがONの場合)

注記

DNSを0,0,0,0に設定すると、DNSは無効になります。

使用可能条件

測定中は設定の変更はできません。

LAN/LAN2の設定の確定

:SYSTEM:COMMunicate:LAN:UPDate
:SYSTEM:COMMunicate:LAN2:UPDate

構文

コマンド

:SYSTEM:COMMunicate:LAN:UPDate
:SYSTEM:COMMunicate:LAN2:UPDate

解説

LAN/LAN2の設定を更新します。

例

:SYSTem:COMMunicate:LAN:IPADdress 192,168,1,10
:SYSTem:COMMunicate:LAN:SMASk 255,255,255,0
:SYSTem:COMMunicate:LAN:GATeway 192,168,1,12
:SYSTem:COMMunicate:LAN:CONTRol 8800
:SYSTEM:COMMunicate:LAN:UPDate

注記

このコマンドを実行しないと、実動作は変わりません。

使用可能条件

測定中は設定の変更はできません。
LANで通信中の接続は切断されます。

LAN2のUDP送信動作における送信先IPアドレスの設定と問い合わせ

`:SYSTem:COMMunicate:LAN2:SEND:IPADdress ip1,ip2,ip3,ip4`

構文

コマンド

`:SYSTem:COMMunicate:LAN2:SEND:IPADdress ip1,ip2,ip3,ip4`

クエリー

`:SYSTEM:COMMunicate:LAN2:SEND:IPADdress?`

応答

`ip1<NR1>,ip2<NR1>,ip3<NR1>,ip4<NR1>`

`ip1 = 0 ~ 255`

`ip2 = 0 ~ 255`

`ip3 = 0 ~ 255`

`ip4 = 0 ~ 255`

解説

LAN2のUDP送信動作の送信先IPアドレスを設定します。
現在のLAN2のUDP送信動作の送信先IPアドレスを返します。

例

`:SYSTem:COMMunicate:LAN2:SEND:IPADdress 192,168,10,3`

`:SYSTem:COMMunicate:LAN2:SEND:IPADdress?`

(応答) `:SYSTEM:COMMUNICATE:LAN2:SEND:IPADDRESS 192,168,10,3` (ヘッダがONの場合)

注記

マルチキャスト用のIP アドレス (224,0,0,0 ~ 239,255,255,255) を指定することで、マルチキャスト通信ができます。

使用可能条件

測定中は設定の変更はできません。

LAN2のUDP送信動作における送信先ポート番号の設定と問い合わせ

:SYSTem:COMMunicate:LAN2:SEND:PORT no

構文

コマンド

:SYSTem:COMMunicate:LAN2:SEND:PORT no

クエリー

:SYSTEM:COMMunicate:LAN2:SEND:PORT?

応答

no<NR1> (n = 1020 ~ 65535)

解説

LAN2のUDP送信動作の送信先ポート番号を設定します。
現在のLAN2のUDP送信動作の送信先ポート番号を返します。

例

:SYSTem:COMMunicate:LAN2:SEND:PORT 10000
:SYSTem:COMMunicate:LAN2:SEND:PORT?
(応答) :SYSTEM:COMMUNICATE:LAN2:SEND:PORT 10000 (ヘッダがONの場合)

注記

使用可能条件

測定中は設定の変更はできません。

LAN2のUDP送信動作における送信データのバイトの並び順の設定と問い合わせ

:SYSTem:COMMunicate:LAN2:SEND:ENDian A\$

構文

コマンド

:SYSTem:COMMunicate:LAN2:SEND:ENDian A\$

クエリー

:SYSTEM:COMMunicate:LAN2:SEND:ENDian?

応答

A\$ = BIG,LITTLE

解説

LAN2のUDP送信動作における送信データのバイトの並び順を設定します。
現在のLAN2のUDP送信動作における送信データのバイトの並び順を返します。

例

:SYSTem:COMMunicate:LAN2:SEND:ENDian LITTLE

:SYSTem:COMMunicate:LAN2:SEND:ENDian?

(応答) :SYSTEM:COMMUNICATE:LAN2:SEND:ENDIAN LITTLE (ヘッダがONの場合)

注記

使用可能条件

測定中は設定の変更はできません。

LAN2のUDP送信動作における送信データの出力形式の設定と問い合わせ

`:SYSTem:COMMunicate:LAN2:SEND:FORMat A$`

構文

コマンド

`:SYSTem:COMMunicate:LAN2:SEND:FORMat A$`

クエリー

`:SYSTEM:COMMunicate:LAN2:SEND:FORMat?`

応答

`A$ = INT32,FLOAT,INDex`

解説

LAN2のUDP送信動作における送信データの出力形式を設定します。
現在のLAN2のUDP送信動作における送信データの出力形式を返します。

INT32: 符号あり、32ビット、INT形式

FLOAT: 単精度浮動小数点数の形式

INDex: 指数形式(アスキー)

例

`:SYSTem:COMMunicate:LAN2:SEND:FORMat FLOAT`

`:SYSTem:COMMunicate:LAN2:SEND:FORMat?`

(応答) `:SYSTEM:COMMUNICATE:LAN2:SEND:FORMAT FLOAT` (ヘッダがONの場合)

注記

使用可能条件

測定中は設定の変更はできません。

全セカンダリー器のデータの出力設定と問い合わせ(プライマリー器の場合)

:SYSTem:COMMunicate:LAN2:SEND:SYNC A\$

構文

コマンド

:SYSTem:COMMunicate:LAN2:SEND:SYNC A\$

クエリー

:SYSTEM:COMMunicate:LAN2:SEND:SYNC?

応答

A\$ = OFF, ON

解説

全セカンダリー器のデータの出力を設定します。

現在の全セカンダリー器のデータの出力設定を返します。

OFF: 本器がプライマリー器の場合に、全セカンダリーのデータを本器のLAN2から出力しません。

ON: 本器がプライマリー器の場合に、全セカンダリーのデータを本器のLAN2から出力します。出力設定はプライマリー器に依存します。プライマリー器の測定値出力機能の設定がLAN2udpになっている場合に出力します。

例

:SYSTem:COMMunicate:LAN2:SEND:SYNC ON

:SYSTem:COMMunicate:LAN2:SEND:SYNC?

(応答) :SYSTEM:COMMUNICATE:LAN2:SEND:SYNC ON (ヘッダがONの場合)

注記

使用可能条件

測定中は設定の変更はできません。

波形データの初期化

:SYSTem:DATAClear

構文

コマンド

:SYSTem:DATAClear

解説

波形データをクリアします。

例

:SYSTem:DATAClear

注記

波形データのクリアには数秒間かかります。

使用可能条件

カレンダーの設定と問い合わせ

:SYSTem:DATE year,month,day

構文

コマンド

:SYSTem:DATE year,month,day

クエリー

:SYSTem:DATE?

応答

year<NR1>,month<NR1>,day<NR1>

year = 0 ~ 37 (年)

month = 1 ~ 12 (月)

day = 1 ~ 31 (日)

解説

カレンダーの設定をします。
現在のカレンダーを返します。

例

:SYSTem:DATE 20,1,2

:SYSTem:DATE?

(応答) **:SYSTEM:DATE 20,01,02** (ヘッダがONの場合)

注記

測定中は、カレンダーの設定はできません。
カレンダーの設定には1秒程度時間がかかります。

使用可能条件

本体日時の設定と問い合わせ

`:SYSTem:DATETime year,month,day,hour,minute,second`

構文

コマンド

`:SYSTem:DATETime year,month,day,hour,minute,second`

クエリー

`:SYSTem:DATETime?`

応答

`year<NR1>,month<NR1>,day<NR1>,hour<NR1>,minute<NR1>,second<NR1>`

year = 0 ~ 37 (年)

month = 1 ~ 12 (月)

day = 1 ~ 31 (日)

hour = 0 ~ 23 (時)

minute = 0 ~ 59 (分)

second = 0 ~ 59 (秒)

解説

本体日時の設定をします。

現在の日時を返します。

例

`:SYSTem:DATETime 20,1,2,12,34,56`

`:SYSTem:DATETime?`

(応答) `:SYSTEM:DATETIME 20,01,02,12,34,56` (ヘッダがONの場合)

注記

測定中は、本体の日時の設定はできません。

本体の日時の設定には1秒程度時間がかかります。

使用可能条件

日付フォーマットの設定と問い合わせ

:SYSTem:DFORmat A\$

構文

コマンド

:SYSTem:DFORmat A\$

クエリー

:SYSTem:DFORmat?

応答

A\$

A\$ = YYYYMMDD ,MMDDYYYY ,DDMMYYYY

解説

日付フォーマットの設定をします。
現在の日付フォーマットの設定を返します。

例

:SYSTem:DFORmat YYYYMMDD

:SYSTem:DFORmat?

(応答) :SYSTEM:DFORMAT YYYYMMDD (ヘッダがONの場合)

注記

測定中は、日付フォーマットの設定はできません。

使用可能条件

日付区切り文字の設定と問い合わせ

:SYSTem:DSEParator A\$

構文

コマンド

:SYSTem:DSEParator A\$

クエリー

:SYSTem:DSEParator?

応答

A\$

A\$ = HYPHen ,SLASh ,PERiod

解説

日付区切り文字の設定をします。
現在の日付区切り文字の設定を返します。

例

:SYSTem:DSEParator HYPHen

:SYSTem:DSEParator?

(応答) :SYSTEM:DSEPARATOR HYPHEN (ヘッダがONの場合)

注記

測定中は、日付区切り文字の設定はできません。

使用可能条件

手動保存ファイル名の設定と問い合わせ

:SYSTem:FILEName "A\$"

構文

コマンド

:SYSTem:FILEName "A\$"

クエリー

:SYSTem:FILEName?

応答

"A\$"

A\$ = ファイル名(最大半角8文字分)

解説

手動保存時のファイル名を設定します。
現在の手動保存時のファイル名を返します。

例

:SYSTem:FILENAME "MANUAL"

:SYSTem:FILENAME?

(応答) :SYSTEM:FILENAME "MANUAL" (ヘッダがONの場合)

注記

最大文字数以上の文字列を入力した場合、最大文字数以降の文字は入力されません。

使用可能条件

言語の設定と問い合わせ

:SYSTem:LANGUage A\$

構文

コマンド

:SYSTem:LANGUage A\$

クエリー

:SYSTem:LANGUage?

応答

A\$

A\$ = JAPANese, ENGLISH

解説

表示言語を設定します。
現在の表示言語の設定を返します。
JAPANese: 日本語
ENGLISH: 英語

例

:SYSTem:LANGUage JAPANese
:SYSTem:LANGUage?
(応答) :SYSTEM:LANGUGE JAPANESE (ヘッダがONの場合)

注記

使用可能条件

警報時のイベントマーク入力の設定と問い合わせ

:SYSTem:MARK A\$

構文

コマンド

:SYSTem:MARK A\$

クエリー

:SYSTem:MARK?

応答

A\$

A\$ = OFF, ON

解説

警報時のイベントマーク入力を設定します。
現在の警報時のイベントマーク入力の設定を返します。

例

:SYSTem:MARK ON

:SYSTem:MARK?

(応答) :SYSTEM:MARK ON (ヘッダがONの場合)

注記

使用可能条件

即保存の優先保存先の設定と問い合わせ

`:SYSTem:SAVEPri A$`

構文

コマンド

`:SYSTem:SAVEPri A$`

クエリー

`SYSTem:SAVEPri?`

応答

A\$

A\$ = SD, USB

解説

即保存機能の優先保存先を設定します。
現在の即保存機能の優先保存先を返します。
SD: SDメモリカード
USB: USBメモリ

例

`:SYSTem:SAVEPri SD`
`:SYSTem:SAVEPri?`
(応答) `:SYSTEM:SAVEPRI SD` (ヘッダがONの場合)

注記

使用可能条件

スタート状態保持機能の設定と問い合わせ

:SYSTem:STARt A\$

構文

コマンド

:SYSTem:STARt A\$

クエリー

:SYSTem:STARt?

応答

A\$

A\$ = OFF, ON

解説

スタート状態保持機能を設定します。
現在のスタート状態保持機能の設定を返します。

例

:SYSTem:STARt ON

:SYSTem:STARt?

(応答) :SYSTEM:START ON (ヘッダがONの場合)

注記

使用可能条件

自動保存の間引き保存データの設定と問い合わせ

`:SYSTem:THINData A$`

構文

コマンド

`:SYSTem:THINData A$`

クエリー

`:SYSTem:THINData?`

応答

A\$

A\$ = INSTANT, STATISTICS

解説

自動保存の間引き保存データを設定します。

現在の自動保存の間引き保存データの設定を返します。

INSTANT: 先頭のデータを保存します。

STATISTICS: 統計データ(最大値、最小値、平均値、先頭のデータ)を保存します。

例

`:SYSTem:THINData INSTANT`

`:SYSTem:THINData?`

(応答) `:SYSTEM:THINDATA INSTANT` (ヘッダがONの場合)

注記

保存形式をテキスト形式に設定し、間引き保存で2以上にしてください。

使用可能条件

手動保存の間引き保存の設定と問い合わせ

:SYSTem:THINOut A

構文

コマンド

:SYSTem:THINOut A

クエリー

:SYSTem:THINOut?

応答

A<NR1>

A = 1(OFF) ~ 100000

解説

手動保存時のテキスト保存間引き間隔を設定します。
現在の手動保存時のテキスト保存間引き間隔を返します。
A=1は間引き保存OFFとなります。

例

:SYSTem:THINOut 5

:SYSTem:THINOut?

(応答) :SYSTEM:THINOUT 5 (ヘッダがONの場合)

注記

使用可能条件

本体時刻の設定と問い合わせ

:SYSTem:TIME h,m,s

構文

コマンド

:SYSTem:TIME h,m,s

クエリー

:SYSTem:TIME?

応答

h<NR1>,m<NR1>,s<NR1>

h = 0 ~ 23 (時)

m = 0 ~ 59 (分)

s = 0 ~ 59 (秒)

解説

本体の時刻を設定します。
現在の時刻を返します。

例

:SYSTem:TIME 12,34,56

:SYSTem:TIME?

(応答) :SYSTEM:TIME 12,34,56 (ヘッダがONの場合)

注記

測定中は、本体の時刻の設定はできません。
本体の時刻の設定には1秒程度時間がかかります。

使用可能条件

タイムゾーンの設定と問い合わせ

:SYSTem:TIMEZone hour(,min)

構文

コマンド

:SYSTem:TIMEZone hour(,min)

クエリー

:SYSTem:TIMEZone?

応答

hour<NR1>(,min<NR1>)

hour = -12 ~ +14 (時間)

min = 30, 45 (分) (省略時は0分)

解説

タイムゾーンを設定します。
現在のタイムゾーンの設定を返します。

例

:SYSTem:TIMEZone 9

:SYSTem:TIMEZone?

(応答) :SYSTEM:TIMEZONE +9 (ヘッダがONの場合)

注記

タイムゾーンを変更すると、時計の時刻も連動して変わります。
hourとminが設定できない組み合わせの場合、エラーとなります。

使用可能条件

横軸の表示の設定と問い合わせ

:SYSTem:TMAXis A\$

構文

コマンド

:SYSTem:TMAXis A\$

クエリー

:SYSTem:TMAXis?

応答

A\$

A\$ = TIME, DATE, SCALE

解説

横軸の表示を設定します。
現在の横軸の表示の設定を返します。

TIME: 時間

DATE: 日付

SCALE: データ数

例

:SYSTem:TMAXis TIME

:SYSTem:TMAXis?

(応答) :SYSTEM:TMAXIS TIME (ヘッダがONの場合)

注記

使用可能条件

外部入力端子1の入力種類の設定と問い合わせ

:SYSTem:EXT:IO1:KIND A\$

構文

コマンド

:SYSTem:EXT:IO1:KIND A\$

クエリー

:SYSTem:EXT:IO1:KIND?

応答

A\$

A\$ = OFF, STARTIN, STOPIN, S_SIN, EVENTIN

解説

外部入力端子1(I/O 1端子)の入力種類を設定します。

現在の外部入力端子1の入力種類を返します。

OFF

STARTIN: スタート

STOPIN: ストップ

S_SIN: スタートおよびストップ

EVENTIN: イベント入力

例

:SYSTem:EXT:IO1:KIND STARTIN

:SYSTem:EXT:IO1:KIND?

(応答) :SYSTEM:EXT:IO1:KIND STARTIN (ヘッダがONの場合)

注記

使用可能条件

外部入力端子1の開始スロープの設定と問い合わせ

:SYSTem:EXT:IO1:SLOPe:STARt A\$

構文

コマンド

:SYSTem:EXT:IO1:SLOPe:STARt A\$

クエリー

:SYSTem:EXT:IO1:SLOPe:STARt?

応答

A\$

A\$ = UP, DOWN

解説

外部入力端子1(I/O 1端子)の開始スロープを設定します。

現在の外部入力端子1の開始スロープを返します。

UP: 立ち上がり

DOWN: 立ち下がり

例

:SYSTem:EXT:IO1:SLOPe:STARt UP

:SYSTem:EXT:IO1:SLOPe:STARt?

(応答) :SYSTEM:EXT:IO1:SLOPE:START UP (ヘッダがONの場合)

注記

使用可能条件

外部入力端子1の停止スロープの設定と問い合わせ

:SYSTem:EXT:IO1:SLOPe:STOP A\$

構文

コマンド

:SYSTem:EXT:IO1:SLOPe:STOP A\$

クエリー

:SYSTem:EXT:IO1:SLOPe:STOP?

応答

A\$

A\$ = UP, DOWN

解説

外部入力端子1(I/O 1端子)の停止スロープを設定します。

現在の外部入力端子1の停止スロープを返します。

UP: 立ち上がり

DOWN: 立ち下がり

例

:SYSTem:EXT:IO1:SLOPe:STOP UP

:SYSTem:EXT:IO1:SLOPe:STOP?

(応答) :SYSTEM:EXT:IO1:SLOPE:STOP UP (ヘッダがONの場合)

注記

使用可能条件

外部入力端子2の入力種類の設定と問い合わせ

:SYSTem:EXT:IO2:KIND A\$

構文

コマンド

:SYSTem:EXT:IO2:KIND A\$

クエリー

:SYSTem:EXT:IO2:KIND?

応答

A\$

A\$ = OFF, STARTIN, STOPIN, S_SIN, EVENTIN

解説

外部入力端子2(I/O 2端子)の入力種類を設定します。

現在の外部入力端子2の入力種類を返します。

OFF

STARTIN: スタート

STOPIN: ストップ

S_SIN: スタートおよびストップ

EVENTIN: イベント入力

例

:SYSTem:EXT:IO2:KIND STARTIN

:SYSTem:EXT:IO2:KIND?

(応答) :SYSTEM:EXT:IO2:KIND STARTIN (ヘッダがONの場合)

注記

使用可能条件

外部入力端子2の開始スロープの設定と問い合わせ

:SYSTem:EXT:IO2:SLOPe:STARt A\$

構文

コマンド

:SYSTem:EXT:IO2:SLOPe:STARt A\$

クエリー

:SYSTem:EXT:IO2:SLOPe:STARt?

応答

A\$

A\$ = UP, DOWN

解説

外部入力端子2(I/O 2端子)の開始スロープを設定します。

現在の外部入力端子2の開始スロープを返します。

UP: 立ち上がり

DOWN: 立ち下がり

例

:SYSTem:EXT:IO2:SLOPe:STARt UP

:SYSTem:EXT:IO2:SLOPe:STARt?

(応答) :SYSTEM:EXT:IO2:SLOPE:START UP (ヘッダがONの場合)

注記

使用可能条件

外部入力端子2の停止スロープの設定と問い合わせ

:SYSTem:EXT:IO2:SLOPe:STOP A\$

構文

コマンド

:SYSTem:EXT:IO2:SLOPe:STOP A\$

クエリー

:SYSTem:EXT:IO2:SLOPe:STOP?

応答

A\$

A\$ = UP, DOWN

解説

外部入力端子2(I/O 2端子)の停止スロープを設定します。

現在の外部入力端子2の停止スロープを返します。

UP: 立ち上がり

DOWN: 立ち下がり

例

:SYSTem:EXT:IO2:SLOPe:STOP UP

:SYSTem:EXT:IO2:SLOPe:STOP?

(応答) :SYSTEM:EXT:IO2:SLOPE:STOP UP (ヘッダがONの場合)

注記

使用可能条件

外部入力端子3の入力種類の設定と問い合わせ

`:SYSTem:EXT:IO3:KIND A$`

構文

コマンド

`:SYSTem:EXT:IO3:KIND A$`

クエリー

`:SYSTem:EXT:IO3:KIND?`

応答

A\$

A\$ = OFF, TRIGIN, EVENTIN

解説

外部入力端子3(I/O 1端子)の入力種類を設定します。

現在の外部入力端子3の入力種類を返します。

OFF

TRIGIN: トリガ入力

EVENTIN: イベント入力

例

`:SYSTem:EXT:IO3:KIND TRIGIN`

`:SYSTem:EXT:IO3:KIND?`

(応答) `:SYSTEM:EXT:IO3:KIND TRIGIN` (ヘッダがONの場合)

注記

外部トリガの設定が有効の場合、トリガ入力以外に設定を変更することはできません。

使用可能条件

外部入力端子3の開始スロープの設定と問い合わせ

:SYSTem:EXT:IO3:SLOPe:STARt A\$

構文

コマンド

:SYSTem:EXT:IO3:SLOPe:STARt A\$

クエリー

:SYSTem:EXT:IO3:SLOPe:STARt?

応答

A\$

A\$ = UP, DOWN

解説

外部入力端子3(I/O 3端子)の開始スロープを設定します。

現在の外部入力端子3の開始スロープを返します。

UP: 立ち上がり

DOWN: 立ち下がり

例

:SYSTem:EXT:IO3:SLOPe:STARt UP

:SYSTem:EXT:IO3:SLOPe:STARt?

(応答) :SYSTEM:EXT:IO3:SLOPE:START UP (ヘッダがONの場合)

注記

使用可能条件

外部入力端子3の停止スロープの設定と問い合わせ

:SYSTem:EXT:IO3:SLOPe:STOP A\$

構文

コマンド

:SYSTem:EXT:IO3:SLOPe:STOP A\$

クエリー

:SYSTem:EXT:IO3:SLOPe:STOP?

応答

A\$

A\$ = UP, DOWN

解説

外部入力端子3(I/O 3端子)の停止スロープを設定します。

現在の外部入力端子3の停止スロープを返します。

UP: 立ち上がり

DOWN: 立ち下がり

例

:SYSTem:EXT:IO3:SLOPe:STOP UP

:SYSTem:EXT:IO3:SLOPe:STOP?

(応答) :SYSTEM:EXT:IO3:SLOPE:STOP UP (ヘッダがONの場合)

注記

使用可能条件

外部出力端子の出力種類の設定と問い合わせ

:SYSTem:EXT:IO4:KIND A\$

構文

コマンド

:SYSTem:EXT:IO4:KIND A\$

クエリー

:SYSTem:EXT:IO4:KIND?

応答

A\$

A\$ = OFF, TRIGOUT

解説

外部出力端子(I/O 4端子)の出力種類を設定します。

現在の外部出力端子の出力種類を返します。

OFF

TRIGOUT: トリガ出力

例

:SYSTem:EXT:IO4:KIND TRIGOUT

:SYSTem:EXT:IO4:KIND?

(応答) :SYSTEM:EXT:IO4:KIND TRIGOUT (ヘッダがONの場合)

注記

使用可能条件

外部サンプリング・外部トリガ端子のフィルターの設定と問い合わせ

:SYSTem:EXTFILTer A\$

構文

コマンド

:SYSTem:EXTFILTer A\$

クエリー

:SYSTem:EXTFILTer?

応答

A\$

A\$ = OFF, ON

解説

外部サンプリング・外部トリガ端子のフィルターの設定をします。
現在の外部サンプリング・外部トリガ端子のフィルターの設定を返します。
OFF: フィルターを無効にします。
ON: フィルターを有効にします。

例

:SYSTem:EXTFILTer ON
:SYSTem:EXTFILTer?
(応答) :SYSTEM:EXTFILTER ON (ヘッダがONの場合)

注記

使用可能条件

外部サンプリング・外部トリガ端子のスロープの設定と問い合わせ

:SYSTem:EXTSLOPe A\$

構文

コマンド

:SYSTem:EXTSLOPe A\$

クエリー

:SYSTem:EXTSLOPe?

応答

A\$

A\$ = UP, DOWN

解説

外部サンプリング・外部トリガ端子のスロープの設定をします。
現在の外部サンプリング・外部トリガ端子のスロープの設定を返します。
UP: LowレベルからHighレベルへの立ち上がりエッジで動作します。
DOWN: HighレベルからLowレベルへの立ち下がりエッジで動作します。

例

:SYSTem:EXTSLOPe UP
:SYSTem:EXTSLOPe?
(応答) :SYSTEM:EXTSLOPE UP (ヘッダがONの場合)

注記

使用可能条件

NTPクライアント機能の接続先サーバーアドレス設定と問い合わせ

`:SYSTem:NTP:ADDRess "A$"`

構文

コマンド

`:SYSTem:NTP:ADDRess "A$"`

クエリー

`:SYSTem:NTP:ADDRess?`

応答

`"A$"`

A\$ = 送信先サーバーアドレス(半角64文字まで)

解説

NTPクライアント機能の接続先サーバアドレスを設定します。
現在のNTPクライアント機能の接続先サーバーアドレスを返します。

例

`:SYSTem:NTP:ADDRess "abcdef.com"`

`:SYSTem:NTP:ADDRess?`

(応答) `:SYSTEM:FTP:ADDRESS "abcdef.com"` (ヘッダがONの場合)

注記

最大文字数以上の文字列を入力した場合、コマンドエラーになります。

使用可能条件

NTPクライアント機能による時刻同期の実行と結果の応答

:SYSTem:NTP:CHECK?

構文

クエリー

:SYSTem:NTP:CHECK?

応答

A

A = 0(成功), 1(失敗)

解説

NTPクライアント機能による時刻同期を実行し、結果を返します。

例

:SYSTem:NTP:CHECK?

(応答) :SYSTEM:NTP:CHECK 0 (ヘッダがONの場合)

注記

使用可能条件

NTPクライアント機能の設定がONであること。

NTPクライアント機能の設定と問い合わせ

:SYSTem:NTP:KIND A\$

構文

コマンド

:SYSTem:NTP:KIND A\$

クエリー

SYSTem:NTP:KIND?

応答

A\$

A\$ = OFF, ON

解説

NTPクライアント機能の有効無効を設定します。
現在のNTPクライアント機能の有効無効を返します。
OFF: NTPクライアント機能OFF
ON: NTPクライアント機能ON

例

:SYSTem:NTP:KIND ON
:SYSTem:NTP:KIND?
(応答) :SYSTEM:NTP:KIND ON (ヘッダがONの場合)

注記

使用可能条件

NTPクライアント機能の同期タイミング設定と問い合わせ

:SYSTem:NTP:SYNC A\$

構文

コマンド

:SYSTem:NTP:SYNC A\$

クエリー

:SYSTem:NTP:SYNC?

応答

A\$

A\$ = OFF, HOUR, DAY

解説

NTPクライアント機能の同期タイミングを設定します。
現在のNTPクライアント機能の同期タイミングの設定を返します。
OFF: 同期OFF
HOUR: 1時間毎に同期
DAY: 1日毎に同期

例

:SYSTem:NTP:SYNC HOUR
:SYSTem:NTP:SYNC?
(応答) :SYSTEM:NTP:SYNC HOUR (ヘッダがONの場合)

注記

使用可能条件

測定開始前時刻同期の設定と問い合わせ

:SYSTem:NTP:STARt A\$

構文

コマンド

:SYSTem:NTP:STARt A\$

クエリー

:SYSTem:NTP:STARt?

応答

A\$

A\$ = OFF, ON

解説

NTPクライアント機能における、測定開始前時刻同期の有効無効を設定します。

現在の測定開始前時刻同期の設定を返します。

OFF: 測定前時刻同期OFF

ON: 測定前時刻同期ON

例

:SYSTem:NTP:STARt ON

:SYSTem:NTP:STARt?

(応答) :SYSTEM:NTP:START ON (ヘッダがONの場合)

注記

使用可能条件

データ自動送信先FTPサーバーアドレスの設定と問い合わせ

`:SYSTem:FTP:ADDRess "A$"`

構文

コマンド

`:SYSTem:FTP:ADDRess "A$"`

クエリー

`:SYSTem:FTP:ADDRess?`

応答

`"A$"`

A\$ = 送信先FTPサーバーアドレス(半角45文字まで)

解説

FTPデータ自動送信機能における、送信先FTPサーバーアドレスを設定します。
現在の自動送信先FTPサーバーアドレスを返します。

例

`:SYSTem:FTP:ADDRess "abcdef"`

`:SYSTem:FTP:ADDRess?`

(応答) `:SYSTEM:FTP:ADDRESS "abcdef"` (ヘッダがONの場合)

注記

最大文字数以上の文字列を入力した場合、コマンドエラーになります。

使用可能条件

送信済みファイル削除機能のON/OFFの設定と問い合わせ

:SYSTem:FTP:AUTODEl A\$

構文

コマンド

:SYSTem:FTP:AUTODEl A\$

クエリー

:SYSTem:FTP:AUTODEl?

応答

A\$

A\$ = OFF, ON

解説

FTPデータ自動送信機能における、送信済みファイル削除機能のON/OFFを設定します。

現在の送信済みファイル削除機能の設定を返します

OFF: 送信済みファイル削除OFF

ON: 送信済みファイル削除ON

例

:SYSTem:FTP:AUTODEl ON

:SYSTem:FTP:AUTODEl?

(応答) :SYSTEM:FTP:AUTODEL ON (ヘッダがONの場合)

注記

使用可能条件

サーバー証明書の確認のONOFFの設定と問い合わせ

:SYSTem:FTP:CErTificate A\$

構文

コマンド

:SYSTem:FTP:CErTificate A\$

クエリー

:SYSTem:FTP:CErTificate?

応答

A\$

A\$ = OFF, ON

解説

FTPデータ自動送信機能における、サーバー証明書のONOFFを設定します。

現在のサーバー証明書の確認機能の設定を返します。

OFF: サーバー証明書の確認OFF

ON: サーバー証明書の確認ON

例

:SYSTem:FTP:CErTificate ON

:SYSTem:FTP:CErTificate?

(応答) :SYSTEM:FTP:CErTIFICATE ON (ヘッダがONの場合)

注記

使用可能条件

FTPデータ自動送信テストの実行と結果の問い合わせ

:SYSTem:FTP:CHECK?

構文

クエリー

:SYSTem:FTP:CHECK?

応答

A

A = 0(成功), 1(失敗)

解説

FTPデータ自動送信テストを実行し、結果を返します。

例

:SYSTem:FTP:CHECK?

(応答) :SYSTEM:FTP:CHECK 0 (ヘッダがONの場合)

注記

使用可能条件

データ自動送信先ディレクトリの設定と問い合わせ

```
:SYSTem:FTP:DIR "A$"
```

構文

コマンド

```
:SYSTem:FTP:DIR "A$"
```

クエリー

```
:SYSTem:FTP:DIR?
```

応答

```
"A$"
```

A\$ = 送信先ディレクトリ(半角45文字まで)

解説

FTPデータ自動送信機能における、送信先ディレクトリを設定します。
現在の自動送信先ディレクトリを返します。

例

```
:SYSTem:FTP:DIR "/abc/def"
```

```
:SYSTem:FTP:DIR?
```

(応答) :SYSTEM:FTP:DIR "/abc/def/" (ヘッダがONの場合)

注記

最大文字数以上の文字列を入力した場合、コマンドエラーになります。

使用可能条件

送信ファイル識別子(ホスト名)の設定と問い合わせ

:SYSTem:FTP:FILE:HOST A\$

構文

コマンド

:SYSTem:FTP:FILE:HOST A\$

クエリー

:SYSTem:FTP:FILE:HOST?

応答

A\$

A\$ = OFF, ON

解説

FTPデータ自動送信機能において、送信ファイル識別子としてホスト名を付与するかを設定します。
現在の送信ファイル識別子(ホスト名)の設定を返します。

OFF: ファイル識別子としてホスト名を付加しない

ON: ファイル識別子としてホスト名を付加する

例

:SYSTem:FTP:FILE:HOST ON

:SYSTem:FTP:FILE:HOST?

(応答) :SYSTEM:FTP:FILE:HOST ON (ヘッダがONの場合)

注記

使用可能条件

送信ファイル識別子(IPアドレス)の設定と問い合わせ

:SYSTem:FTP:FILE:IP A\$

構文

コマンド

:SYSTem:FTP:FILE:IP A\$

クエリー

:SYSTem:FTP:FILE:IP?

応答

A\$

A\$ = OFF, ON

解説

FTPデータ自動送信機能において、送信ファイル識別子としてIPアドレスを付与するかを設定します。

現在の送信ファイル識別子(IPアドレス)の設定を返します。

OFF: ファイル識別子としてIPアドレスを付加しない

ON: ファイル識別子としてIPアドレスを付加する

例

:SYSTem:FTP:FILE:IP ON

:SYSTem:FTP:FILE:IP?

(応答) :SYSTEM:FTP:FILE:IP ON (ヘッダがONの場合)

注記

使用可能条件

送信ファイル識別子(時刻)の設定と問い合わせ

:SYSTem:FTP:FILE:TIME A\$

構文

コマンド

:SYSTem:FTP:FILE:TIME A\$

クエリー

:SYSTem:FTP:FILE:TIME?

応答

A\$

A\$ = OFF, ON

解説

FTPデータ自動送信機能において、送信ファイル識別子として時刻を付与するかを設定します。

現在の送信ファイル識別子(時刻)の設定を返します。

OFF: ファイル識別子として時刻を付加しない

ON: ファイル識別子として時刻を付加する

例

:SYSTem:FTP:FILE:TIME ON

:SYSTem:FTP:FILE:TIME?

(応答) :SYSTEM:FTP:FILE:TIME ON (ヘッダがONの場合)

注記

使用可能条件

FTPデータ自動送信機能パスワードの設定と照合結果の問い合わせ

`:SYSTem:FTP:PASSword "A$"`

構文

コマンド

`:SYSTem:FTP:PASSword "A$"`

クエリー

`:SYSTem:FTP:PASSword? "A$"`

応答

B\$

A\$ = パスワード(半角32文字まで)

B\$ = PASS,FAIL

PASS: 入力したパスワードが正しい

FAIL: 入力したパスワードが間違っている

解説

FTPデータ自動送信機能における、パスワードを設定します。

入力したパスワードと、現在設定されているパスワードを照合し、結果を返します。

例

`:SYSTem:FTP:PASSword "abcd"`

`:SYSTem:FTP:PASSword? "abcd"`

(応答) `:SYSTEM:FTP:PASSWORD PASS` (ヘッダがONの場合)

注記

最大文字数以上の文字列を入力した場合、コマンドエラーになります。

使用可能条件

FTPデータ自動送信機能パッシブモードの設定と問い合わせ

:SYSTem:FTP:PASV A\$

構文

コマンド

:SYSTem:FTP:PASV A\$

クエリー

:SYSTem:FTP:PASV?

応答

A\$

A\$ = OFF, ON

解説

FTPデータ自動送信機能における、パッシブモードのON/OFFを設定します。

現在のパッシブモードの設定を返します。

OFF: パッシブモードOFF

ON: パッシブモードON

例

:SYSTem:FTP:PASV ON

:SYSTem:FTP:PASV?

(応答) :SYSTEM:FTP:PASV ON (ヘッダがONの場合)

注記

使用可能条件

データ自動送信先FTPサーバーポート番号の設定と問い合わせ

:SYSTem:FTP:PORT A

構文

コマンド

:SYSTem:FTP:PORT A

クエリー

:SYSTem:FTP:PORT?

応答

A<NR1>

A = ポート番号(0 ~ 65535)

解説

FTPデータ自動送信機能における、送信先サーバーのポート番号を設定します。
現在の送信先サーバーポート番号を返します。

例

:SYSTem:FTP:PORT 1234

:SYSTem:FTP:PORT?

(応答) :SYSTEM:FTP:PORT 1234 (ヘッダがONの場合)

注記

使用可能条件

未送信ファイル有無の問い合わせ

:SYSTem:FTP:PROGress?

構文

クエリー

:SYSTem:FTP:PROGress?

応答

A\$

A\$ = YES, NO

解説

FTPデータ自動送信機能において、未送信ファイルの有無を判定し結果を返します。

YES: 未送信ファイル有り

NO: 未送信ファイル無し

例

:SYSTem:FTP:PROGress?

(応答) :SYSTEM:FTP:PROGRESS YES (ヘッダがONの場合)

注記

使用可能条件

FTP接続の保護機能の設定と問い合わせ

:SYSTem:FTP:SECURity A\$

構文

コマンド

:SYSTem:FTP:SECURity A\$

クエリー

:SYSTem:FTP:SECURity?

応答

A\$

A\$ = OFF, EXPLICIT, IMPLICIT

解説

FTP接続の保護機能を設定します。
現在のFTP接続の保護機能の設定を返します。
OFF: FTP接続の保護機能OFF
EXPLICIT: 明示的な FTP over TLS
IMPLICIT: 暗黙的な FTP over TLS

例

:SYSTem:FTP:SECURity EXPLICIT
:SYSTem:FTP:SECURity?
(応答) :SYSTEM:FTP:SECURITY EXPLICIT (ヘッダがONの場合)

注記

使用可能条件

FTPデータ自動送信機能通信状況の問い合わせ

:SYSTem:FTP:STATe?

構文

クエリー

:SYSTem:FTP:STATe?

応答

A<NR1>,B<NR1>,C<NR1>,D<NR1>

A = 総ファイル数

B = 送信済みファイル数

C = 送信失敗ファイル数

D = 未送信ファイル数

解説

FTPデータ自動送信機能の通信状況を返します。

例

:SYSTem:FTP:STATe?

(応答) :SYSTEM:FTP:STATE 10,1,5,4 (ヘッダがONの場合)

注記

使用可能条件

FTPデータ自動送信機能ONOFFの設定と問い合わせ

:SYSTem:FTP:USE A\$

構文

コマンド

:SYSTem:FTP:USE A\$

クエリー

:SYSTem:FTP:USE?

応答

A\$

A\$ = OFF, ON

解説

FTPデータ自動送信機能のONOFFを設定します。
現在のFTPデータ自動送信機能のONOFFを返します。
OFF: FTPデータ自動送信機能OFF
ON: FTPデータ自動送信機能ON

例

:SYSTem:FTP:USE ON
:SYSTem:FTP:USE?
(応答) :SYSTEM:FTP:USE ON (ヘッダがONの場合)

注記

使用可能条件

FTPデータ自動送信機能ユーザー名の設定と問い合わせ

:SYSTem:FTP:USER "A\$"

構文

コマンド

:SYSTem:FTP:USER "A\$"

クエリー

:SYSTem:FTP:USER?

応答

"A\$"

A\$ = ユーザー名(半角32文字まで)

解説

FTPデータ自動送信機能における、ユーザー名を設定します。
現在のユーザー名を返します。

例

:SYSTem:FTP:USER "LR8101"

:SYSTem:FTP:USER?

(応答) :SYSTEM:FTP:USER "LR8101" (ヘッダがONの場合)

注記

最大文字数以上の文字列を入力した場合、コマンドエラーになります。

使用可能条件

測定値出力機能の設定と問い合わせ

:SYSTem:RTOuT A\$

構文

コマンド

:SYSTem:RTOuT A\$

クエリー

:SYSTem:RTOuT?

応答

A\$

A\$ = OFF, CAN, LAN2udp

解説

測定値出力機能を設定します。

現在の測定値出力機能を返します。

OFF: CANおよびLAN2からデータを出力しません。

CAN: CAN端子からデータを出力します。

LAN2udp: LAN2からデータを出力します。

例

:SYSTem:RTOuT CAN

:SYSTem:RTOuT?

(応答) :SYSTEM:RTOuT CAN (ヘッダがONの場合)

注記

使用可能条件

3.11 ストレージデータ(MEMory)

ストレージデータの出力 (ASCII)

:MEMory:ADATa? A

構文

クエリー

:MEMory:ADATa? A

応答

B1,B2,...<NR1>(波形演算結果のみ<NR3>)

A = 1 ~ 2000 (出力個数)

Bi = -2147483648 ~ 2147483647(アナログ)

Bi = 0 ~ 2147483647(積算、回転数)

Bi = 0 ~ 1(ロジック)

Bi = 0 ~ 15(警報)

Bi = 波形演算結果(波形演算)

解説

:MEMory:APOINtで設定されたチャンネルとポイントから、ストレージデータをAで指定した個数分出力します。

出力ポイントもデータ数分増加します。

アナログのデータ値を物理量に変換する場合は、以下の計算式を用います。

物理量 = データ値 × 電圧軸レンジ / レンジあたりのデータ数

スケールが設定されている場合は、以下の計算式を用います。

変換値 = 物理量 × スケール係数 + スケールオフセット

レンジあたりのデータ数は以下の表の通りです。

モジュール		モード	レンジ当たりのデータ数 ※1Vレンジだったら、 0-1Vまでのデータ数
M7100 M7102	電圧・温度モジュール(15ch) 電圧・温度モジュール(30ch)	電圧(全レンジ) ※1-5Vは6Vと同様	100000
		熱電対(100°Cレンジ)	10000
		熱電対(500°Cレンジ)	10000
		熱電対(2000°Cレンジ)	20000

積算、波形演算のデータ値は測定値そのものを返します。

回転数の値を求めるには:MODule:PCOUntで設定した1回転あたりのパルス数でデータ値を除算する必要があります。

回転数 = データ値 / 1回転あたりのパルス数

警報チャンネルは、全チャンネル (4bit) を整数表現したものを返します。(一番下のbitから警報1、警報2...の並びで出力)

例

:MEMory:APOINT CH1_1,0

:MEMory:ADATa? 10

(応答) :MEMORY:ADATA 3176,3176,3176,3186,3186,3186,3186,3186,3198,3198 (ヘッダがONの場合)

注記

Aの値と本体表記の対応は以下の通りです。

Aの値	本体表記
2147483647	+OVER
-2147483648	-OVER
2147483646	BURNOUT
2147483645	NO DATA

[参考]

アナログチャンネルのデータ数が1,000,000ポイント記録されている場合、:MEMory:ADATa? 2000を500回繰り返した時のデータ転送時間は、約26秒です。(OS:Microsoft Windows 10 Pro(22H2), CPU:Intel(R) i7-9700F 3.00GHz, RAM:16GB)

この値はあくまでも目安であり、通信速度を保証するものではありません。ご使用の環境によって通信速度は変化します。

使用可能条件

出力ポイントが、ストレージデータ数未満であること。

:MEMory:POINTでチャンネルを指定していること。

ストレージデータがあること。

ホールドデータの出力 (ASCII)

:MEMory:AFETch? ch\$

構文

クエリー

:MEMory:AFETch? ch\$

応答

A<NR1>(波形演算結果のみ<NR3>)
ch\$ = CH1_1 ~ CH10_30, PLS1, LOG, ALARM, W1 ~ W30
A = -2147483648 ~ 2147483647(アナログ)
A = 0 ~ 2147483647(積算、回転数)
A = 0 ~ 1(ロジック)
A = 0 ~ 15(警報)
A = 波形演算結果(波形演算)

解説

:MEMory:GETRealコマンドにて取り込んだホールドデータのうち、指定したチャンネルの値を返します。
返すデータ値と測定値の関係は:MEMory:ADATa?と同様です。

例

:MEMory:AFETch? CH1_1
(応答) :MEMORY:AFETCH 3176 (ヘッダがONの場合)

注記

このコマンドの前に、ホールドデータの取り込みを実行していない場合、NO DATAを示す値として2147483645を返します。

使用可能条件

ストレージ末尾データ番号の問い合わせ(内部メモリよりも長いデータを測定する時)

:MEMory:AMAXPoint?

構文

クエリー

:MEMory:AMAXPoint?

応答

A<NR1>

A = 0(ストレージされていない), 1 ~

解説

ストレージメモリにストレージされている末尾データ番号を返します。

例

:MEMory:AMAXPoint?

(応答) :MEMORY:AMAXPOINT 800 (ヘッダがONの場合)

注記

使用可能条件

ストレージ出力ポイントの設定と問い合わせ(内部メモリよりも長いデータを測定する時)

:MEMory:APOINT ch\$,A

構文

コマンド

:MEMory:APOINT ch\$,A

クエリー

:MEMory:APOINT?

応答

ch\$,A<NR1>
ch\$ = CH1_1 ~ CH10_30, PLS1, LOG, ALARM, W1 ~ W30
A = 0 ~ (ストレージデータ数 - 1)

解説

ストレージデータの出力ポイントを設定します。
現在のストレージデータの出力ポイントを返します。

例

:MEMory:APOINT CH1_1,100
:MEMory:APOINT?
(応答) :MEMORY:APOINT CH1_1,100 (ヘッダがONの場合)

注記

ストレージデータがない場合は、出力ポイントを設定できません。
Aに設定できる値は:MEMory:AMAXPoint?で得られる値よりも小さい値です。
測定データが内部メモリよりも長い場合、指定したポイントのデータが取得できない可能性があります。

使用可能条件

リアルタイムデータの出力 (ASCII)

:MEMory:AREAL? ch\$

構文

クエリー

:MEMory:AREAL? ch\$

応答

A<NR1>(波形演算結果のみ<NR3>)
ch\$ = CH1_1 ~ CH10_30, PLS1, LOG, ALARM, W1 ~ W30
A = -2147483648 ~ 2147483647(アナログ)
A = 0 ~ 2147483647(積算、回転数)
A = 0 ~ 1(ロジック)
A = 0 ~ 15(警報)
A = 波形演算結果(波形演算)

解説

指定したチャンネルの最終ストレージデータの値を返します。

例

:MEMory:AREAL? CH1_1
(応答) :MEMORY:AREAL 3176 (ヘッダがONの場合)

注記

測定データの取り込みを行う以下の動作いずれかを実施している場合、最終ストレージデータの値を返します。

- 測定
- :MEMory:GETRealコマンド実行
上記動作を行っていない場合、NO DATAを示す値として2147483645を返します。

使用可能条件

ストレージデータの出力 (バイナリ)

:MEMory:BDATa? A

構文

クエリー

:MEMory:BDATa? A

応答

#0<以下バイナリデータ>
A = 1 ~ 5000 (出力個数)

解説

:MEMory:POINtで設定されたチャンネルとポイントから、ストレージデータをAで指定した数だけバイナリフォーマット (ビッグエンディアン) で出力します。

出力ポイントもデータ数分増加します。

出力データの先頭には、#0(バイナリフォーマットを表す)がつきます。

#0に続いて、バイナリフォーマットのストレージデータがAで指定したデータの数だけ送信されます。

チャンネル種類毎のバイナリフォーマットは以下の通りです。

- ロジック、警報チャンネル:1データ2バイト単位で出力
- アナログ、パルスチャンネル:1データ4バイト単位で出力
- 波形演算チャンネル:1データ8バイト単位で出力
(波形演算チャンネルの応答はIEEE 754における倍精度の浮動小数点形式で出力されます)
得られるデータは:MEMory:ADATa?と同様です。データの内容については、:MEMory:ADATa?を参照してください。

例

:MEMory:POINt CH1_1,0
:MEMory:BDATa? 10
(応答) :MEMORY:BDATA #0...(バイナリデータ) (ヘッダがONの場合)

注記

ストレージデータが存在しないチャンネル、出力ポイントを指定した場合、エラー値が出力されます。エラー値は指定チャンネルによって異なります。

[アナログチャンネル]
0x7fffffd(4バイト)

[パルスチャネル]
0x00000000(4バイト)

[波形演算チャネル]
0x7ff0000000000001(8バイト)

バイナリデータの中には、改行コード (文字コードの0Aや0D) が含まれることがあります。お使いのPCソフトウェアが、改行コードをデータの終了として判断している場合は、正しくデータを処理できませんので、必ずAで指定した分のデータを読み込むようにしてください。データの終わりに改行コード (LFやCR+LF) はつきません。

【参考】

アナログチャネルのデータ数が1,000,000ポイント記録されている場合、:MEMory:BDATa? 2000を500回繰り返した時のデータ転送時間は、約4秒です。(OS:Microsoft Windows 10 Pro(22H2), CPU:Intel(R) i7-9700F 3.00GHz, RAM:16GB)

この値はあくまでも目安であり、通信速度を保証するものではありません。ご使用の環境によって通信速度は変化します。

使用可能条件

出力ポイントが、ストレージデータ数未満であること。
ストレージデータがあること。

ホールドデータの出力 (バイナリ)

:MEMory:BFETch? ch\$

構文

クエリー

:MEMory:BFETch? ch\$

応答

A
ch\$ = CH1_1 ~ CH10_30, PLS1, LOG, ALARM, W1 ~ W30
A = バイナリデータ

解説

:MEMory:GETRealコマンドにて取り込んだホールドデータのうち、指定したチャンネルの値を返します。
返すデータ値と測定値の関係は:MEMory:BDATA?と同様です。

例

:MEMory:BFETch? CH1_1
(応答) :MEMORY:BFETCH (バイナリデータ) (ヘッダがONの場合)

注記

測定データの取り込みを行う以下の動作いずれかを実施している場合、最終ストレージデータの値を返します。

- 測定
- :MEMory:GETRealコマンド実行
上記動作を行っていない場合、NO DATAを示す値として2147483645を返します。

使用可能条件

リアルタイムデータの出力 (バイナリ)

:MEMory:BreAL? ch\$

構文

クエリー

:MEMory:BreAL? ch\$

応答

A
ch\$ = CH1_1 ~ CH10_30, PLS1, LOG, ALARM, W1 ~ W30
A = バイナリデータ

解説

指定したチャンネルの最終ストレージデータの値を返します。
返すデータ値と測定値の関係は:MEMory:BDATa?と同様です。

例

:MEMory:BreAL? CH1_1
(応答) :MEMory:BreAL (バイナリデータ) (ヘッダがONの場合)

注記

測定データの取り込みを行う以下の動作いずれかを実施している場合、最終ストレージデータの値を返します。

- 測定
- :MEMory:GETRealコマンド実行
上記動作を行っていない場合、指定チャンネルに応じたエラー値を返します。

使用可能条件

ストレージデータの有無の問い合わせ

:MEMory:CHSTore? ch\$

構文

クエリー

:MEMory:CHSTore? ch\$

応答

ch\$,A\$

ch\$ = CH1_1 ~ CH10_30, PLS1, LOG, ALARM, W1 ~ W30

A\$ = OFF, ON

解説

指定したチャンネルのデータがストレージメモリにストレージされているかを返します。

例

:MEMory:CHSTore? CH1_1

(応答) :MEMORY:CHSTORE CH1_1,ON (ヘッダがONの場合)

注記

使用可能条件

ホールドデータの有無の問い合わせ

:MEMory:FCHSTore? ch\$

構文

クエリー

:MEMory:FCHSTore? ch\$

応答

ch\$,A\$

ch\$ = CH1_1 ~ CH10_30, PLS1, LOG, ALARM, W1 ~ W30

A\$ = OFF, ON

解説

指定したチャンネルのホールドデータが保持されているかを返します。

例

:MEMory:FCHSTore? CH1_1

(応答) :MEMORY:FCHSTORE CH1_1,ON (ヘッダがONの場合)

注記

使用可能条件

ホールドデータの取り込み

:MEMory:GETReal

構文

コマンド

:MEMory:GETReal

解説

現在入力されているデータを、全ての測定チャンネル分取り込み、ホールドデータとして保持します。

例

:MEMory:GETReal

注記

データの保持には、数秒かかります。

:MEMory:GETRealコマンドの動作中に、:MEMory:GETRealコマンドを再び実行した場合、コマンド実行エラーとなります。

保持完了を待ってから別のコマンドを実行する場合はOPCやWAIを使用してください。

使用可能条件

ストレージデータ数の問い合わせ

:MEMory:MAXPoint?

構文

クエリー

:MEMory:MAXPoint?

応答

A<NR1>

A = ストレージデータ数 (0=ストレージされていない)

解説

ストレージメモリに記録されているデータ数を返します。

例

:MEMory:MAXPoint?

(応答) :MEMORY:MAXPOINT 2500 (ヘッダがONの場合)

注記

使用可能条件

ストレージデータの出力ポイントの設定と問い合わせ

:MEMory:POINT ch\$,A

構文

コマンド

:MEMory:POINT ch\$,A

クエリー

:MEMory:POINT?

応答

ch\$,A<NR1>

ch\$ = CH1_1 ~ CH10_30, PLS1, LOG, ALARM, W1 ~ W30

A\$ = 0 ~ 268435456(ch1_1時最大)

解説

ストレージデータの出力ポイントを設定します。
現在のストレージデータの出力ポイントを返します。

例

:MEMory:POINT CH1_1,100

:MEMory:POINT?

(応答) :MEMORY:POINT CH1_1,100 (ヘッダがONの場合)

注記

ストレージデータがない場合は、出力ポイントを設定できません。
Aに設定できる値は:MEMory:MAXPoint?で得られる値よりも小さい値です。

使用可能条件

ホールドデータの出力(モジュール単位) (ASCII)

:MEMory:TAFETch? module\$

構文

クエリー

:MEMory:TAFETch? module\$

応答

A<NR1>(波形演算結果のみ<NR3>)
module\$ = MODULE1 ~ MODULE10, PLS&ALM, CALC1, CALC2
A = -2147483648 ~ 2147483647(アナログ)
A = 0 ~ 2147483647(積算、回転数)
A = 0 ~ 1(ロジック)
A = 0 ~ 15(警報)
A = 波形演算結果(波形演算)

解説

:MEMory:GETRealコマンドにて取り込んだホールドデータのうち、指定したモジュール(データ種類)のホールドデータがあるチャンネルの値を返します。
返すデータ値と測定値の関係は:MEMory:ADATa?と同様です。

例

:MEMory:TAFETch? MODULE1
(応答) :MEMORY:TAFETCH
3176,3176,3176,3186,3186,3186,3186,3186,3198,3198,3186,3186,3186,3198,3198 (ヘッダがONの場合)

注記

このコマンドの前に、ホールドデータの取り込みを実行していない場合、コマンドエラーになります。
指定モジュール内にホールドデータを持つチャンネルが存在しない場合、コマンドエラーになります。

使用可能条件

リアルタイム測定チャネルの問い合わせ (ASCII)

:MEMory:TARCH? module\$

構文

クエリー

:MEMory:TARCH? module\$

応答

ch1\$,ch2\$,...

module\$ = MODULE1 ~ MODULE10, PLS&ALM, CALC1, CALC2

ch\$ = CH1_1 ~ CH10_30, PLS1, LOG, ALARM, W1 ~ W30

解説

指定したモジュール(データ種類)の測定ONチャネルを返します。

例

:MEMory:TARCH? MODULE1

(応答) :MEMORY:TARCH

CH1_1,CH1_2,CH1_3,CH1_4,CH1_5,CH1_6,CH1_7,CH1_8,CH1_9,CH1_10,CH1_11,CH1_12,CH1_13,CH1_14,CH1_15 (ヘッダがONの場合)

注記

使用可能条件

リアルタイムデータの出力 (モジュール単位) (ASCII)

:MEMory:TAREAL? module\$

構文

クエリー

:MEMory:TAREAL? module\$

応答

A1,A2,...<NR1>(波形演算結果のみ<NR3>)
module\$ = MODULE1 ~ MODULE10, PLS&ALM, W1 ~ W30
A = -2147483648 ~ 2147483647(アナログ)
A = 0 ~ 2147483647(積算、回転数)
A = 0 ~ 1(ロジック)
A = 0 ~ 15(警報)
A = 波形演算結果(波形演算)

解説

指定したモジュール(データ種類)の測定ONチャンネルの最終ストレージデータの値を返します。

例

:MEMory:TAREAI? MODULE1
(応答) :MEMORY:TAREAL
3176,3176,3176,3186,3186,3186,3186,3186,3198,3198,3186,3186,3186,3198,3198 (ヘッダがONの場合)

注記

測定データの取り込みを行う以下の動作いずれかを実施している場合、最終ストレージデータの値を返します。

- 測定
- :MEMory:GETRealコマンド実行
上記動作を行っていない場合、NO DATAを示す値として2147483645を返します。
指定モジュール内に測定ONのチャンネルが存在しない場合、NO_STORAGEが返ってきます。

使用可能条件

ストレージデータの有無の問い合わせ(モジュール単位)

:MEMory:TCHSTore? module\$

構文

クエリー

:MEMory:TCHSTore? module\$

応答

ch1\$,ch2\$,...

module\$ = MODULE1 ~ MODULE10, PLS&ALM, W1 ~ W30

ch\$ = CH1_1 ~ CH10_30, PLS1, LOG, ALARM, W1 ~ W30

解説

指定したモジュール(データ種類)の測定データがあるチャンネルを返します。

例

:MEMory:TCHSTore? MODULE1

(応答) :MEMORY:TCHSTORE

CH1_1,CH1_2,CH1_3,CH1_4,CH1_5,CH1_6,CH1_7,CH1_8,CH1_9,CH1_10,CH1_11,CH1_12,CH1_13,CH1_14,CH1_15 (ヘッダがONの場合)

注記

指定したモジュールが未実装の場合には「MODULE_NONE」となります。

指定したモジュール内のチャンネルの測定がすべてOFFの場合には、「NO DATA」となります。

使用可能条件

ホールドデータの有無の問い合わせ (モジュール単位)

:MEMory:TFCHSTore? module\$

構文

クエリー

:MEMory:TFCHSTore? module\$

応答

ch1\$,ch2\$,...

module\$ = MODULE1 ~ MODULE10, PLS&ALM, W1 ~ W30

ch\$ = CH1_1 ~ CH10_30, PLS1, LOG, ALARM, W1 ~ W30

解説

指定したモジュール(データ種類)のホールドデータがあるチャンネルを返します。

例

:MEMory:TFCHSTore? MODULE1

(応答) :MEMORY:TFCHSTORE

CH1_1,CH1_2,CH1_3,CH1_4,CH1_5,CH1_6,CH1_7,CH1_8,CH1_9,CH1_10,CH1_11,CH1_12,CH1_13,CH1_14,CH1_15 (ヘッダがONの場合)

注記

指定したモジュールが未実装の場合には「MODULE_NONE」となります。

指定したモジュール内のチャンネルの測定がすべてOFFの場合には、「NO DATA」となります。

使用可能条件

ストレージ先頭データ番号の問い合わせ(内部メモリよりも長いデータを測定する時)

:MEMory:TOPPoint?

構文

クエリー

:MEMory:TOPPoint?

応答

A<NR1>

A = 先頭データ番号 (0=ストレージされていない)

解説

ストレージメモリに記録されている先頭データ番号を返します。

例

:MEMory:TOPPoint?

(応答) :MEMORY:TOPPOINT 100 (ヘッダがONの場合)

注記

使用可能条件

ホールドデータの出力 (モジュール単位) (測定値)

:MEMory:TVFETch? module\$

構文

クエリー

:MEMory:TVFETch? module\$

応答

A1,A2,...<NR3>

module\$ = MODULE1 ~ MODULE10, PLS&ALM, CALC1, CALC2

Ax = 測定値

解説

:MEMory:GETRealコマンドにて取り込んだホールドデータのうち、指定したモジュール(データ種類)のホールドデータがあるチャンネルの値を測定値で返します。

得られるデータの内容については、:MEMory:VDATa?の解説を参照してください。

例

:MEMory:TVFETch? MODULE1

(応答) :MEMORY:TVFETCH

+1.00000E+03,+2.00000E+03,+3.00000E+03,+4.00000E+03,+5.00000E+03,+6.00000E+03,+7.00000E+03,+8.00000E+03,+1.00000E+03,+2.00000E+03,+3.00000E+03,+4.00000E+03,+5.00000E+03,+6.00000E+03,+7.00000E+03 (ヘッダがONの場合)

注記

このコマンドの前に、ホールドデータの取り込みを実行していない場合、コマンドエラーになります。

指定モジュール内にホールドデータを持つチャンネルが存在しない場合、コマンドエラーになります。

使用可能条件

リアルタイム測定チャネルの問い合わせ (測定値)

:MEMory:TVRCH? module\$

構文

クエリー

:MEMory:TVRCH? module\$

応答

ch1\$,ch2\$,...

module\$ = MODULE1 ~ MODULE10, PLS&ALM, CALC1, CALC2

ch\$ = CH1_1 ~ CH10_30, PLS1, LOG, ALARM, W1 ~ W30

解説

指定したモジュール(データ種類)の測定ONチャネルを返します。 (:MEMory:TARCH?と同じ)

例

:MEMory:TVRCH? MODULE1

(応答) :MEMORY:TVRCH

CH1_1,CH1_2,CH1_3,CH1_4,CH1_5,CH1_6,CH1_7,CH1_8,CH1_9,CH1_10,CH1_11,CH1_12,CH1_13,CH1_14,CH1_15 (ヘッダがONの場合)

注記

使用可能条件

リアルタイムデータの出力 (モジュール単位) (測定値)

:MEMory:TVREAL? module\$

構文

クエリー

:MEMory:TVREAL? module\$

応答

A1,A2,...<NR3>

module\$ = MODULE1 ~ MODULE10, PLS&ALM, CALC1, CALC2

Ax = 測定値

解説

指定したモジュール(データ種類)の測定ONチャンネルの最終ストレージデータの値を測定値で返します。

得られるデータの内容については、:MEMory:VDATA?の解説を参照してください。

例

:MEMory:TVREAL? MODULE1

(応答) :MEMORY:TVREAL

+1.00000E+03,+2.00000E+03,+3.00000E+03,+4.00000E+03,+5.00000E+03,+6.00000E+03,+7.00000E+03,+8.00000E+03,+1.00000E+03,+2.00000E+03,+3.00000E+03,+4.00000E+03,+5.00000E+03,+6.00000E+03,+7.00000E+03 (ヘッダがONの場合)

注記

測定データの取り込みを行う以下の動作いずれかを実施している場合、最終ストレージデータの値を返します。

- 測定
 - :MEMory:GETRealコマンド実行
上記動作を行っていない場合、いずれの入力種類においても、エラー値として9.99999E+99を返します。
- 指定モジュール内に測定ONのチャンネルが存在しない場合、NO_STORAGEが返ってきます。

使用可能条件

ストレージデータの出力 (測定値)

:MEMory:VDATa? A

構文

クエリー

:MEMory:VDATa? A

応答

B1,B2,...<NR3>

A = 1 ~ 1000

Bi = 測定値

小数点以下の桁数は指定チャンネルによって異なります。

[アナログチャンネル]

有効数字7桁

[パルスチャンネル]

有効数字10桁

[波形演算チャンネル]

有効数字12桁

[ロジック]

整数(1bit)

[警報チャンネル]

整数(4bit)

解説

:MEMory:POINtで設定されたチャンネル、およびポイントから、測定値をAで指定した個数分出力します。

出力ポイントもデータ数分増加します。

スケーリングされている場合は、スケーリングされた値が出力されます。

アナログチャンネル、パルスチャンネル、波形演算チャンネルについては、指数部が3の倍数になるように小数点位置が可変します。

[例]

+1.234567E+03,+12.34567E+03,+123.4567E+03,+1.234567E+06

例

:MEMory:POINt CH1_1,0

:MEMory:VDATa? 2

(応答) :MEMORY:VDATA +5.000000E+03,+4.000000E+03 (ヘッダがONの場合)

注記

ストレージデータが存在しないチャンネル、ポイントを指定した場合、いずれの入力種類においても、エラー値として9.99999E+99を返します。

[参考]

アナログチャンネルのデータ数が1,000,000ポイント記録されている場合、:MEMory:VDATa? 2000を500回繰り返した時のデータ転送時間は、約70秒です。(OS:Microsoft Windows 10 Pro(22H2), CPU:Intel(R) i7-9700F 3.00GHz, RAM:16GB)

この値はあくまでも目安であり、通信速度を保証するものではありません。ご使用の環境によって通信速度は変化します。

使用可能条件

出力ポイントが、ストレージデータ数未満であること。

:MEMory:POINTでチャンネルを指定していること。

データを出力する際は、ストレージデータがあること。

ホールドデータの出力 (測定値)

:MEMory:VFETch? ch\$

構文

クエリー

:MEMory:VFETch? ch\$

応答

A<NR3>
ch\$ = CH1_1 ~ CH10_30, PLS1, LOG, ALARM, W1 ~ W30
A = 測定値

解説

:MEMory:GETRealコマンドにて取り込んだホールドデータのうち、指定したチャンネルの測定値を返します。

得られるデータの内容については、:MEMory:VDATa?の解説を参照してください。

例

:MEMory:VFETch? CH1_1
(応答) :MEMORY:VFETCH +1.23000E+03 (ヘッダがONの場合)

注記

このコマンドの前に、ホールドデータの取り込みを実行していない場合、いずれの入力種類においても、エラー値として9.99999E+99を返します。

使用可能条件

リアルタイムデータの出力 (測定値)

:MEMory:VREAL? ch\$

構文

クエリー

:MEMory:VREAL? ch\$

応答

A<NR3>
ch\$ = CH1_1 ~ CH10_30, PLS1, LOG, ALARM, W1 ~ W30
A = 測定値

解説

指定したチャンネルの最終ストレージデータの測定値を返します。

例

:MEMory:VREAL? CH1_1
(応答) :MEMORY:VREAL +1.23000E+03 (ヘッダがONの場合)

注記

測定データの取り込みを行う以下の動作いずれかを実施している場合、最終ストレージデータの値を返します。

- 測定
- :MEMory:GETRealコマンド実行
上記動作を行っていない場合、NO DATAを示す値として9.99999E+99を返します。

使用可能条件

3.12 画面表示(DISPLAY)

イベントマークの挿入

:DISPlay:MARK

構文

コマンド

:DISPlay:MARK

解説

イベントマークを挿入します。

例

:DISPlay:MARK

注記

使用可能条件

スタート動作中であること。

イベントマーク数の問い合わせ

:DISPlay:MARK?

構文

クエリー

:DISPlay:MARK?

応答

A<NR1>

A = 0 ~ 1000

解説

イベントマークの総数を返します。

例

:DISPlay:MARK?

(応答) :DISPLAY:MARK 10 (ヘッダがONの場合)

注記

使用可能条件

イベントマークのデータ番号問い合わせ

:DISPlay:MARKJump? A

構文

クエリー

:DISPlay:MARKJump? A

応答

A<NR1>,B<NR1>

A = 1 ~ イベントマーク数

B = データ番号

解説

指定したイベントマークのデータ番号を返します。

例

:DISPlay:MARKJump? 10

(応答) :DISPLAY:MARKJUMP 10,500 (ヘッダがONの場合)

注記

使用可能条件

3.13 演算(CALCulate)

数値演算のON/OFF設定と問い合わせ

:CALCulate:MEASure A\$

構文

コマンド

:CALCulate:MEASure A\$

クエリー

:CALCulate:MEASure?

応答

A\$

A\$ = OFF, ON

解説

数値演算の設定します。
現在の数値演算の設定を返します。
OFF: 無効
ON: 有効

例

:CALCulate:MEASure ON
:CALCulate:MEASure?
(応答) :CALCULATE:MEASURE ON (ヘッダがONの場合)

注記

使用可能条件

数値演算結果の問い合わせ

`:CALCulate:MEAS:ANSWer? no$,ch$`

構文

クエリー

`:CALCulate:MEAS:ANSWer? no$,ch$`

応答

`no$,ch$,A<NR3>` (小数点以下11桁)
`no$ = NO1 ~ NO10`
`ch$ = CH1_1 ~ CH10_30, PLS1, W1 ~ W30`
`A = 数値演算結果`

解説

指定した演算番号、チャネルの数値演算結果を返します。

例

`:CALCulate:MEAS:ANSWer? NO1,CH1_1`
(応答) `:CALCULATE:MEAS:ANSWER NO1,CH1_1,+1.23456789012E-03` (ヘッダがONの場合)

注記

以下の条件のとき、`A=NONE`(文字列)を応答します。

- 数値演算の設定がOFFの場合
- 指定した演算番号の数値演算種類がOFFの場合
- 演算結果が存在しない、または得られない場合
従来コマンド(`:CALCulate:ANSWer? no$,ch$`)による問い合わせも可能。

使用可能条件

数値演算結果仕分けの設定と問い合わせ

:CALCulate:MEAS:FILE A\$

構文

コマンド

:CALCulate:MEAS:FILE A\$

クエリー

:CALCulate:MEAS:FILE?

応答

A\$

A\$ = OFF, ON

解説

数値演算結果仕分けの設定をします。
現在の数値演算結果仕分けの設定を返します。
OFF: 単一ファイル
ON: 演算別ファイル

例

:CALCulate:MEAS:FILE ON
:CALCulate:MEAS:FILE?
(応答) :CALCULATE:MEAS:FILE ON (ヘッダがONの場合)

注記

従来コマンド(:CALCulate:MEASFile A\$ / :CALCulate:MEASFile?)による設定、問い合わせも可能。

使用可能条件

積算、積分時の設定と問い合わせ

:CALCulate:MEAS:INTegra no\$,A\$

構文

コマンド

:CALCulate:MEAS:INTegra no\$,A\$

クエリー

:CALCulate:MEAS:INTegra? no\$

応答

no\$ = NO1 ~ NO10

A\$ = TOTAL, POSitive, NEGative, ABSolute

解説

積算、積分処理の設定を行います。
現在の積算、積分処理の設定を返します。
TOTAL: 全て加算
POSitive: 正の値のみ加算
NEGative: 負の値のみ加算
ABSolute: 絶対値のみ加算

例

:CALCulate:MEAS:INTegra NO1,TOTAL

:CALCulate:MEAS:INTegra? NO1

(応答) :CALCULATE:MEAS:INTEGRA NO1,TOTAL (ヘッダがONの場合)

注記

使用可能条件

設定が可能な演算は、下記のものに限られます。
積算 (ACC)
積分 (INT)

時間分割演算の種類の設定と問い合わせ

:CALCulate:MEAS:KIND A\$

構文

コマンド

:CALCulate:MEAS:KIND A\$

クエリー

:CALCulate:MEAS:KIND?

応答

A\$

A\$ = OFF, DIVide, ONTIME

解説

時間区切り演算の分割の設定を行います。
現在の時間区切り演算の分割の設定を返します。
OFF: 分割なし
DIVide: 分割あり
ONTIME: 定時分割

例

:CALCulate:MEAS:KIND DIVide

:CALCulate:MEAS:KIND?

(応答) :CALCULATE:MEAS:KIND DIVIDE (ヘッダがONの場合)

注記

従来コマンド(:CALCulate:MEAS:KIND A\$ / :CALCulate:MEAS:KIND?)による設定、問い合わせも可能。

使用可能条件

時間分割演算の設定と問い合わせ

`:CALCulate:MEAS:LEN day,hour,min`

構文

コマンド

`:CALCulate:MEAS:LEN day,hour,min`

クエリー

`:CALCulate:MEAS:LEN?`

応答

`day<NR1>,hour<NR1>,min<NR1>`

`day = 0 ~ 30(日)`

`hour = 0 ~ 23(時)`

`min = 0 ~ 59(分)`

解説

時間区切り演算の分割長の設定を行います。
現在の時間区切り演算の分割長の設定を返します。

例

`:CALCulate:MEAS:LEN 0,1,30`

`:CALCulate:MEAS:LEN?`

(応答) `:CALCULATE:MEAS:LEN 0,1,30` (ヘッダがONの場合)

注記

記録間隔の設定値により、分割長の設定が制限される場合があります。
従来コマンド(`:CALCulate:MEASLen day,hour,min` / `:CALCulate:MEASLen?`)による設定、問い合わせも可能。

使用可能条件

数値演算のしきい値の設定と問い合わせ

`:CALCulate:MEAS:LEVEL ch$,A`

構文

コマンド

`:CALCulate:MEAS:LEVEL ch$,A`

クエリー

`:CALCulate:MEAS:LEVEL? ch$`

応答

ch\$,A<NR3> (小数点以下4桁)
ch\$ = CH1_1 ~ CH10_30, PLS1, W1 ~ W30
A = -9.9999E+29 ~ +9.9999E+29

解説

数値演算のしきい値の設定を行います。
現在の数値演算のしきい値の設定を返します。

例

`:CALCulate:MEAS:LEVEL CH1_1,0.123`
`:CALCulate:MEAS:LEVEL? CH1_1`
(応答) `:CALCULATE:MEAS:LEVEL CH1_1,+1.2300E-01` (ヘッダがONの場合)

注記

設定が可能な演算は、下記のものに限られます。
稼働率: (OPE)
ON時間: (ONT)
OFF時間: (OFFT)
ON回数: (ONC)
OFF回数: (OFFC)
設定可能上限より大きい値を入力した場合、最大値が入力されます。
設定可能下限より小さい値を入力した場合、最小値が入力されます。

使用可能条件

時間分割演算定時分割の基準時間の設定と問い合わせ

:CALCulate:MEAS:REG hour,min

構文

コマンド

:CALCulate:MEAS:REG hour,min

クエリー

:CALCulate:MEAS:REG?

応答

hour<NR1>,min<NR1>

hour = 0 ~ 23(時)

min = 0 ~ 59(分)

解説

時間区切り演算の区切り時間の設定を行います。
現在の時間区切り演算の区切り時間の設定を返します。

例

:CALCulate:MEAS:REG 1,30

:CALCulate:MEAS:REG?

(応答) :CALCULATE:MEAS:REG 1,30 (ヘッダがONの場合)

注記

従来コマンド(:CALCulate:MEASReg hour,min / :CALCulate:MEASReg?)による設定、問い合わせも可能。

使用可能条件

演算種類の設定と問い合わせ

:CALCulate:MEAS:SET no\$,A\$

構文

コマンド

:CALCulate:MEAS:SET no\$,A\$

クエリー

:CALCulate:MEAS:SET? no\$

応答

no\$,A\$

no\$ = NO1 ~ NO10

A\$ = OFF, AVE, PP, MAX, MIN, MAXT, MINT, ACC, INT, OPE, ONT, OFFT, ONC, OFFC

解説

指定した演算番号の数値演算の種類を設定します。

指定した演算番号の現在の数値演算の種類を返します。

OFF: 演算なし

AVE: 平均値

PP: P-P値

MAX: 最大値

MIN: 最小値

MAXT: 最大値までの時間

MINT: 最小値までの時間

ACC: 積算

INT: 積分

OPE: 稼働率

ONT: ON時間

OFFT: OFF時間

ONC: ON回数

OFFC: OFF回数

例

:CALCulate:MEAS:SET NO1,AVE

:CALCulate:MEAS:SET? NO1

(応答) :CALCULATE:MEAS:SET NO1,AVE (ヘッダがONの場合)

注記

従来コマンド(:CALCulate:MEASSet no\$,A\$ / :CALCulate:MEASSet? no\$)による設定、問い合わせも可能。

使用可能条件

数値演算対象の設定と問い合わせ

:CALCulate:MEAS:TARGet no\$,ch\$

構文

コマンド

:CALCulate:MEAS:TARGet no\$,ch\$

クエリー

:CALCulate:MEAS:TARGet? no\$

応答

no\$,ch\$

no\$ = NO1 ~ NO10

ch\$ = ALL, CH1_1 ~ CH10_30, PLS1, W1 ~ W30

解説

指定した演算番号の数値演算対象の設定を行います。
指定した演算番号の現在の数値演算対象の設定を返します。

例

:CALCulate:MEAS:TARGet NO1,CH1_1

:CALCulate:MEAS:TARGet? NO1

(応答) :CALCULATE:MEAS:TARGET NO1,CH1_1 (ヘッダがONの場合)

注記

使用可能条件

時間分割演算定時分割の分割時間の設定と問い合わせ

:CALCulate:MEAS:TIME A

構文

コマンド

:CALCulate:MEAS:TIME A

クエリー

:CALCulate:MEAS:TIME?

応答

A<NR1>

A = 1, 2, 5, 10, 15, 20, 30, 60, 120, 180, 240, 360, 480, 720, 1440 (単位min)

解説

時間区切り演算の分割時間を設定します。
現在の時間区切り演算の分割時間を返します。

例

:CALCulate:MEAS:TIME 1

:CALCulate:MEAS:TIME?

(応答) :CALCULATE:MEAS:TIME 1 (ヘッダがONの場合)

注記

設定にない値を指定した場合、設定しようとした値よりも長い時間が存在するときは、最も近い時間に設定されます。

従来コマンド(:CALCulate:MEAS:TIME A / :CALCulate:MEAS:TIME? A)による設定、問い合わせも可能。

使用可能条件

波形演算の種類の設定と問い合わせ

`:CALCulate:WAVE:KIND w$,A$`

構文

コマンド

`:CALCulate:WAVE:KIND w$,A$`

クエリー

`:CALCulate:WAVE:KIND? w$`

応答

w\$,A\$

w\$ = W1 ~ W30

A\$ = OPE, SUM, AVE, MOV, INT

解説

指定したチャンネルの演算式の設定をします。

指定したチャンネルの現在の演算式の設定を返します。

OPE: 四則演算

SUM: 積算

AVE: 単純平均

MOV: 移動平均

INT: 積分

例

`:CALCulate:WAVE:KIND W1,OPE`

`:CALCulate:WAVE:KIND? W1`

(応答) `:CALCULATE:WAVE:KIND W1,OPE` (ヘッダがONの場合)

注記

従来コマンド(`:CALCulate:HTKIND w$,A$ / :CALCulate:HTKIND? w$`)による設定、問い合わせも可能。

使用可能条件

波形演算の単位の設定と問い合わせ

`:CALCulate:WAVE:STR w$,"A$"`

構文

コマンド

`:CALCulate:WAVE:STR w$,"A$"`

クエリー

`:CALCulate:WAVE:STR? w$`

応答

`w$,"A$"`

`w$ = W1 ~ W30`

`A$ = 単位 (最大7文字)`

解説

指定したチャンネルの単位の設定をします。

指定したチャンネルの現在の単位の設定を返します。

コマンド中のシングルクォーテーション(')の代わりに、ダブルクォーテーション(")を使うことができます。

扱うことのできる文字は、本体で入力できる文字と同じですが、特殊文字の入力は下記ようになります。

PC	$\wedge 2$	$\wedge 3$	$\sim u$	$\sim o$	$\sim e$	$\sim c$	$\sim +$	$\sim ,$	$\sim ;$	$\wedge \wedge$	$\sim \sim$
LR8101 LR8102	2	3	μ	Ω	ϵ	o	\pm	'	"	\wedge	\sim

例

`:CALCulate:WAVE:STR W1,"mA"`

`:CALCulate:WAVE:STR? W1`

(応答) `:CALCulate:WAVE:STR W1,"mA"` (ヘッダがONの場合)

注記

最大文字数以上の文字列を入力した場合、最大文字数以降の文字は入力されません。

従来コマンド(`:CALCulate:WVSTR w$,"A$"` / `:CALCulate:WVSTR? w$`)による設定、問い合わせも可能。

使用可能条件

波形演算:四則演算の係数の設定と問い合わせ

:CALCulate:WAVE:ARITHmetic:COEF:A w\$,A (係数A)
:CALCulate:WAVE:ARITHmetic:COEF:B w\$,B (係数B)
:CALCulate:WAVE:ARITHmetic:COEF:C w\$,C (係数C)
:CALCulate:WAVE:ARITHmetic:COEF:D w\$,D (係数D)
:CALCulate:WAVE:ARITHmetic:COEF:E w\$,E (係数E)

構文

コマンド

:CALCulate:WAVE:ARITHmetic:COEF:A w\$,A (係数A)
:CALCulate:WAVE:ARITHmetic:COEF:B w\$,B (係数B)
:CALCulate:WAVE:ARITHmetic:COEF:C w\$,C (係数C)
:CALCulate:WAVE:ARITHmetic:COEF:D w\$,D (係数D)
:CALCulate:WAVE:ARITHmetic:COEF:E w\$,E (係数E)

クエリー

:CALCulate:WAVE:ARITHmetic:COEF:A? w\$ (係数A)
:CALCulate:WAVE:ARITHmetic:COEF:B? w\$ (係数B)
:CALCulate:WAVE:ARITHmetic:COEF:C? w\$ (係数C)
:CALCulate:WAVE:ARITHmetic:COEF:D? w\$ (係数D)
:CALCulate:WAVE:ARITHmetic:COEF:E? w\$ (係数E)

応答

w\$,A<NR3> (小数点以下4桁)
w\$ = W1 ~ W30
A\$ = -9.9999E+29 ~ 9.9999E+29

解説

指定したチャンネルの係数の設定を行います。
指定したチャンネルの現在の係数を返します。

例

:CALCulate:WAVE:ARITHmetic:COEF:A W1,1
:CALCulate:WAVE:ARITHmetic:COEF:A? W1
(応答) :CALCULATE:WAVE:ARITHMETIC:COEF:A W1,+1.0000E+00 (ヘッダがONの場合)

注記

下記の従来コマンドによる設定、問い合わせも可能。

:CALCulate:WVCOE1 w\$,A / :CALCulate:WVCOE1? w\$ (係数A)

:CALCulate:WVCOE1 w\$,B / :CALCulate:WVCOE1? w\$ (係数B)

:CALCulate:WVCOE1 w\$,E / :CALCulate:WVCOE1? w\$ (係数E)

使用可能条件

波形演算:四則演算の演算子の設定と問い合わせ

:CALCulate:WAVE:ARITHmetic:OPERator:A w\$,A\$ (演算子A)

:CALCulate:WAVE:ARITHmetic:OPERator:B w\$,A\$ (演算子B)

:CALCulate:WAVE:ARITHmetic:OPERator:C w\$,A\$ (演算子C)

:CALCulate:WAVE:ARITHmetic:OPERator:D w\$,A\$ (演算子D)

構文

コマンド

:CALCulate:WAVE:ARITHmetic:OPERator:A w\$,A\$ (演算子A)

:CALCulate:WAVE:ARITHmetic:OPERator:B w\$,A\$ (演算子B)

:CALCulate:WAVE:ARITHmetic:OPERator:C w\$,A\$ (演算子C)

:CALCulate:WAVE:ARITHmetic:OPERator:D w\$,A\$ (演算子D)

クエリー

:CALCulate:WAVE:ARITHmetic:OPERator:A? w\$ (演算子A)

:CALCulate:WAVE:ARITHmetic:OPERator:B? w\$ (演算子B)

:CALCulate:WAVE:ARITHmetic:OPERator:C? w\$ (演算子C)

:CALCulate:WAVE:ARITHmetic:OPERator:D? w\$ (演算子D)

応答

w\$,A\$

w\$ = W1 ~ W30

A\$ = OFF, PLUS, MINUS, MULTI, DIV (演算子A, B, C)

A\$ = OFF, PLUS, MINUS, MULTI, DIV, EXP (演算子D)

解説

指定したチャンネルの演算子の設定を行います。

指定したチャンネルの現在の演算子を返します。

例

:CALCulate:WAVE:ARITHmetic:OPERator:A W1,PLUS

:CALCulate:WAVE:ARITHmetic:OPERator:A? W1

(応答) :CALCULATE:WAVE:ARITHMETIC:OPERATOR:A W1,PLUS (ヘッダがONの場合)

注記

演算子Aは、従来コマンド(:CALCulate:WVKINd w\$,A\$ / :CALCulate:WVKINd? w\$)による設定、問い合わせも可能。

使用可能条件

演算子Bは演算子AがOFF以外なら設定可能。

演算子Cは演算子A, BがOFF以外なら設定可能。

波形演算:移動平均ポイント数の設定と問い合わせ

`:CALCulate:WAVE:MOVE:POINT w$,A`

構文

コマンド

`:CALCulate:WAVE:MOVE:POINT w$,A`

クエリー

`:CALCulate:WAVE:MOVE:POINT? w$`

応答

`w$,A<NR1>`

`w$ = W1 ~ W30`

`A = 1 ~ 600 (ポイント数)`

解説

指定したチャンネルの移動平均ポイント数の設定を行います。
指定したチャンネルの現在の移動平均ポイント数を返します。

例

`:CALCulate:WAVE:MOVE:POINT W1,10`

`:CALCulate:WAVE:MOVE:POINT? W1`

(応答) `:CALCULATE:WAVE:MOVE:POINT W1,10` (ヘッダがONの場合)

注記

従来コマンド(`:CALCulate:HTMVPPoint w$,A / :CALCulate:HTMVPPoint? w$`)による設定、問い合わせも可能。

使用可能条件

波形演算:積算、単純平均のリセット基準時刻の設定と問い合わせ

:CALCulate:WAVE:RESet:BASE w\$,hour,min

構文

コマンド

:CALCulate:WAVE:RESet:BASE w\$,hour,min

クエリー

:CALCulate:WAVE:RESet:BASE? w\$

応答

w\$,hour<NR1>,min<NR1>

w\$ = W1 ~ W30

hour = 0 ~ 23(時)

min = 0 ~ 59(分)

解説

指定したチャンネルのリセット基準時刻の設定を行います。
指定したチャンネルの現在のリセット基準時刻を返します。

例

:CALCulate:WAVE:RESet:BASE W1,0,0

:CALCulate:WAVE:RESet:BASE? W1

(応答) **:CALCULATE:WAVE:RESET:BASE W1,0,0** (ヘッダがONの場合)

注記

従来コマンド(:CALCulate:HTBASE w\$,hour,min / :CALCulate:HTBASE? w\$)による設定、問い合わせも可能。

使用可能条件

波形演算:積算、単純平均のリセット間隔の設定と問い合わせ

`:CALCulate:WAVE:RESet:INT w$,day,hour,min`

構文

コマンド

`:CALCulate:WAVE:RESet:INT w$,day,hour,min`

クエリー

`:CALCulate:WAVE:RESet:INT? w$`

応答

`w$,day<NR1>,hour<NR1>,min<NR1>`

`w$ = W1 ~ W30`

`day = 0 ~ 30(日)`

`hour = 0 ~ 23(時)`

`min = 0 ~ 59(分)`

解説

指定したチャンネルのリセット間隔の設定を行います。

指定したチャンネルの現在のリセット間隔を返します。

例

`:CALCulate:WAVE:RESet:INT W1,0,0,1`

`:CALCulate:WAVE:RESet:INT? W1`

(応答) `:CALCULATE:WAVE:RESET:INT W1,0,0,1` (ヘッダがONの場合)

注記

従来コマンド(`:CALCulate:HTINT w$,day,hour,min` / `:CALCulate:HTINT? w$`)による設定、問い合わせも可能。

使用可能条件

波形演算:積算、単純平均の開始リセットの設定と問い合わせ

`:CALCulate:WAVE:RESet:KIND w$,A$`

構文

コマンド

`:CALCulate:WAVE:RESet:KIND w$,A$`

クエリー

`:CALCulate:WAVE:RESet:KIND? w$`

応答

w\$,A\$

w\$ = W1 ~ W30

A\$ = OFF, TRIG

解説

指定したチャンネルの開始リセットの設定を行います。
指定したチャンネルの現在の開始リセットの設定を返します。
OFF
TRIG: トリガ位置

例

`:CALCulate:WAVE:RESet:KIND W1,OFF`

`:CALCulate:WAVE:RESet:KIND? W1`

(応答) `:CALCULATE:WAVE:RESET:KIND W1,OFF` (ヘッダがONの場合)

注記

従来コマンド(`:CALCulate:HTRESet w$,A$` / `:CALCulate:HTRESet? w$`)による設定、問い合わせも可能。

使用可能条件

波形演算:積算、単純平均のリセット時間の設定と問い合わせ

`:CALCulate:WAVE:RESet:TIME w$,A$`

構文

コマンド

`:CALCulate:WAVE:RESet:TIME w$,A$`

クエリー

`:CALCulate:WAVE:RESet:TIME? w$`

応答

w\$,A\$

w\$ = W1 ~ W30

A\$ = OFF, ON, ONTIME

解説

指定したチャンネルのリセット時間の設定を行います。
指定したチャンネルの現在のリセット時間の設定を返します。
OFF: 分割なし
ON: 分割あり
ONTIME: 定時分割

例

`:CALCulate:WAVE:RESet:TIME W1,OFF`

`:CALCulate:WAVE:RESet:TIME? W1`

(応答) `:CALCulate:WAVE:RESet:TIME W1,OFF` (ヘッダがONの場合)

注記

従来コマンド(`:CALCulate:HTRSTTime w$,A$ / :CALCulate:HTRSTTime? w$`)による設定、問い合わせも可能。

使用可能条件

波形演算ソースの設定と問い合わせ

```
:CALCulate:WAVE:SOURce:SR1 w$,ch$ (ソースA)  
:CALCulate:WAVE:SOURce:SR2 w$,ch$ (ソースB)  
:CALCulate:WAVE:SOURce:SR3 w$,ch$ (ソースC)  
:CALCulate:WAVE:SOURce:SR4 w$,ch$ (ソースD)
```

構文

コマンド

```
:CALCulate:WAVE:SOURce:SR1 w$,ch$ (ソースA)  
:CALCulate:WAVE:SOURce:SR2 w$,ch$ (ソースB)  
:CALCulate:WAVE:SOURce:SR3 w$,ch$ (ソースC)  
:CALCulate:WAVE:SOURce:SR4 w$,ch$ (ソースD)
```

クエリー

```
:CALCulate:WAVE:SOURce:SR1? w$ (ソースA)  
:CALCulate:WAVE:SOURce:SR2? w$ (ソースB)  
:CALCulate:WAVE:SOURce:SR3? w$ (ソースC)  
:CALCulate:WAVE:SOURce:SR4? w$ (ソースD)
```

応答

```
w$,ch$  
w$ = W1 ~ W30  
ch$ = CH1_1 ~ CH10_30, PLS1, W1 ~ W30
```

解説

指定したチャンネルのソースチャンネルの設定を行います。
指定したチャンネルの現在のソースチャンネルを返します。

例

```
:CALCulate:WAVE:SOURce:SR1 W1,CH1_1  
:CALCulate:WAVE:SOURce:SR1? W1  
(応答) :CALCULATE:WAVE:SOURCE:SR1 W1,CH1_1 (ヘッダがONの場合)
```

注記

下記の従来コマンドによる設定、問い合わせも可能。
:CALCulate:WVSR1 w\$,ch\$ / :CALCulate:WVSR1? w\$ (ソースA)
:CALCulate:WVSR2 w\$,ch\$ / :CALCulate:WVSR2? w\$ (ソースB)

使用可能条件

3.14 エラー(ERRor)

エラービットの問い合わせ

:ERRor:BIT:ERRor?

構文

クエリー

:ERRor:BIT:ERRor?

応答

errBit

解説

起動後に発生したエラー情報のビットを16進数値で返します。

例

:ERRor:BIT:ERRor?

(応答) :ERROR:BIT:ERROR 40000 (ヘッダがONの場合)

注記

発生した直後のエラーは、本コマンドの応答に反映されないことがあります。
少し時間を置いてから再度確認してください。

使用可能条件

警告ビットの問い合わせ

`:ERRor:BIT:WARNing?`

構文

クエリー

`:ERRor:BIT:WARNing?`

応答

`warnBit`

解説

起動後、あるいは:ERRor:WARNing:CLEAr実行以降に発生した警告情報のビットを16進数値で返します。

例

`:ERRor:BIT:WARNing?`

(応答) `:ERROR:BIT:WARNING 800` (ヘッダがONの場合)

注記

発生した直後の警告は、本コマンドの応答に反映されないことがあります。
少し時間を置いてから再度確認してください。

使用可能条件

警告情報のビットの消去

`:ERRor:BIT:WARNing:CLEAr`

構文

コマンド

`:ERRor:BIT:WARNing:CLEAr`

解説

警告情報のビットを消去します。

例

`:ERRor:BIT:WARNing:CLEAr`

注記

使用可能条件

エラーログの問い合わせ

`:ERRor:LOG:ERRor? no$`

構文

クエリー

`:ERRor:LOG:ERRor? no$`

応答

`error$` :エラー情報文字列

`no$` = 1 ~ 50

解説

指定したエラーログの内容(発生時刻とエラー番号)を出力します(エラー番号の内容は本体取説参照)。
`no$ = 1`が直近で発生したエラーになります。
エラーログには最大50個のエラーが格納されています。

例

`:ERRor:LOG:ERRor? 1`

(応答) `:ERROR:LOG:ERROR 2023/02/28 16:40:47 - ERR_SY01` (ヘッダがONの場合)

注記

発生した直後のエラーは、本コマンドの応答に反映されないことがあります。
少し時間を置いてから再度確認してください。

使用可能条件

エラーログの消去

`:ERRor:LOG:ERRor:CLEAr`

構文

コマンド

`:ERRor:LOG:ERRor:CLEAr`

解説

エラーログを消去します。

例

`:ERRor:LOG:ERRor:CLEAr`

注記

使用可能条件

警告ログの問い合わせ

`:ERRor:LOG:WARning? no$`

構文

クエリー

`:ERRor:LOG:WARning? no$`

応答

`warn$` :警告情報文字列

`no$` = 1 ~ 50

解説

指定した警告ログの内容(発生時刻と警告番号)を出力します(警告番号の内容は本体取説参照)。

`no$ = 1`が直近で発生した警告になります。

警告ログには最大50個の警告が格納されています。

例

`:ERRor:LOG:WARning? 1`

(応答) `:ERROR:LOG:WARNING 2023/05/18 14:55:54 - WARN_SY01` (ヘッダがONの場合)

注記

発生した直後の警告は、本コマンドの応答に反映されないことがあります。

少し時間を置いてから再度確認してください。

使用可能条件

ワーニングログの消去

`:ERRor:LOG:WARNing:CLEAr`

構文

コマンド

`:ERRor:LOG:WARNing:CLEAr`

解説

ワーニングログを消去します。

例

`:ERRor:LOG:WARNing:CLEAr`

注記

使用可能条件

3.15 データ保存(MEDIA)

設定条件ファイルの読み込み状態の問い合わせ

```
:MEDia:SD:FINFo:SET? "fname$"  
:MEDia:USB:FINFo:SET? "fname$"
```

構文

クエリー

```
:MEDia:SD:FINFo:SET? "fname$"  
:MEDia:USB:FINFo:SET? "fname$"
```

応答

```
ans$, modules$  
ans$ = OK, NG_MODEL, NG_MODULE, BUSY  
modules$ = 0, 1, 3
```

解説

設定条件ファイルの読み込み状態を返します。

OK: 読込可能です。

NG_MODEL: 異なる機種の設定ファイルのため読込不可です。

NG_MODULE: 異なるモジュール構成の設定ファイルのため読込不可です。

BUSY: ファイル処理中のため情報取得不可です。

モジュールの種類は左からモジュール1,2,...の順で10桁の数値で返します。

0: モジュールなし

1: M7100 15ch電圧・温度モジュール

3: M7102 30ch電圧・温度モジュール

例

```
:MEDia:SD:FINFo:SET? "CONF0001.SET"  
(応答) :MEDIA:SD:FINFO:SET OK,1300000000 (ヘッダがONの場合)
```

注記

使用可能条件

メディア空き容量の問い合わせ

:MEDia:SD:FREE?

:MEDia:USB:FREE?

構文

クエリー

:MEDia:SD:FREE?

:MEDia:USB:FREE?

応答

A<NR1>

解説

メモリ空き容量を返します。

例

:MEDia:SD:FREE?

(応答) :MEDIA:SD:FREE 100 (ヘッダがONの場合)

注記

使用可能条件

メディアのフォーマット

:MEDia:SD:FORMat?
:MEDia:USB:FORMat?

構文

クエリー

:MEDia:SD:FORMat?
:MEDia:USB:FORMat?

応答

A\$

解説

メディアをフォーマットします。
FAIL: フォーマット失敗
SUCCESS: フォーマット成功

例

:MEDia:SD:FORMat?
(応答) :MEDIA:SD:FORMAT SUCCESS (ヘッダがONの場合)

注記

使用可能条件

メディア内のデータはすべて消去されます。

各メディアの/HIOKI/LR8100/CONFIG内のファイル一覧の問い合わせ

:MEDia:SD:FLISt:SET?

:MEDia:USB:FLISt:SET?

構文

クエリー

:MEDia:SD:FLISt:SET?

:MEDia:USB:FLISt:SET?

応答

A1\$,A2\$,...

A\$ = ファイル名

解説

各メディアの/HIOKI/LR8100/CONFIG内のファイル名を返します。

例

:MEDia:SD:FLISt:SET?

(応答) :MEDIA:SD:FLISt:SET CONF0002.SET,CONF0001.SET (ヘッダがONの場合)

注記

使用可能条件

取得できる一覧の最大は100までです。

ファイル名と読み込みオプションの設定と読み込み状態の問い合わせ

```
:MEDia:SD:LOAD:SET "fname$",option  
:MEDia:USB:LOAD:SET "fname$",option
```

構文

コマンド

```
:MEDia:SD:LOAD:SET "fname$",option  
:MEDia:USB:LOAD:SET "fname$",option
```

クエリー

```
:MEDia:SD:LOAD:SET?  
:MEDia:USB:LOAD:SET?
```

応答

"fname\$" = 読み込み対象の設定条件ファイル名(xxxx.SET)
option = 0 ~ 3 (読み込みオプション)
A\$ = NONE, EXECUTING_LOAD_(ファイル名), SUCCESS_LOAD_(ファイル名),
FAIL_LOAD_(ファイル名)

解説

ファイル名と読み込みオプションの設定をします。

option = 0: 測定設定

option = 1: 測定設定 + 外部端子

option = 2: 測定設定 + 通信設定

option = 3: 測定設定 + 外部端子 + 通信設定

ファイルの読み込み状態を返します。

NONE: 読み込み実行前です。

EXECUTING_LOAD_(ファイル名): 読み込み実行中です。

SUCCESS_LOAD_(ファイル名): 読み込み実行に成功しました。読み込んだファイル名が後ろに付きます。

FAIL_LOAD_(ファイル名): 読み込みに失敗しました。

例

```
:MEDia:SD:LOAD:SET "CONF0001.SET",0  
:MEDia:SD:LOAD:SET?  
(応答) :MEDIA:SD:LOAD:SET SUCCESS_LOAD_CONF0001 (ヘッダがONの場合)
```

注記

使用可能条件

データの保存と保存状態の問い合わせ

[波形データ(バイナリー形式)を保存するとき]

:MEDia:SD:SAVE:DATA:MEM

:MEDia:USB:SAVE:DATA:MEM

[波形データ(テキスト形式)を保存するとき]

:MEDia:SD:SAVE:DATA:CSV

:MEDia:USB:SAVE:DATA:CSV

[波形データ(MDF形式)を保存するとき]

:MEDia:SD:SAVE:DATA:MF4

:MEDia:USB:SAVE:DATA:MF4

[設定データを保存するとき]

:MEDia:SD:SAVE:SET

:MEDia:USB:SAVE:SET

[A2L設定データを保存するとき(LAN1)]

:MEDia:SD:SAVE:A2L:LAN1

:MEDia:USB:SAVE:A2L:LAN1

[A2L設定データを保存するとき(LAN2)]

:MEDia:SD:SAVE:A2L:LAN2

:MEDia:USB:SAVE:A2L:LAN2

[数値演算結果を保存するとき]

:MEDia:SD:SAVE:CALC:CSV

:MEDia:USB:SAVE:CALC:CSV

構文

コマンド

[波形データ(バイナリー形式)を保存するとき]

:MEDia:SD:SAVE:DATA:MEM

:MEDia:USB:SAVE:DATA:MEM

[波形データ(テキスト形式)を保存するとき]

:MEDia:SD:SAVE:DATA:CSV

:MEDia:USB:SAVE:DATA:CSV

[波形データ(MDF形式)を保存するとき]

:MEDia:SD:SAVE:DATA:MF4

:MEDia:USB:SAVE:DATA:MF4

[設定データを保存するとき]

:MEDia:SD:SAVE:SET

:MEDia:USB:SAVE:SET

[A2L設定データを保存するとき(LAN1)]

:MEDia:SD:SAVE:A2L:LAN1

:MEDia:USB:SAVE:A2L:LAN1

[A2L設定データを保存するとき(LAN2)]

:MEDia:SD:SAVE:A2L:LAN2

:MEDia:USB:SAVE:A2L:LAN2

[数値演算結果を保存するとき]

:MEDia:SD:SAVE:CALC:CSV
:MEDia:USB:SAVE:CALC:CSV

クエリー

[波形データ(バイナリー形式)を保存するとき]

:MEDia:SD:SAVE:DATA:MEM?

:MEDia:USB:SAVE:DATA:MEM?

[波形データ(テキスト形式)を保存するとき]

:MEDia:SD:SAVE:DATA:CSV?

:MEDia:USB:SAVE:DATA:CSV?

[波形データ(MDF形式)を保存するとき]

:MEDia:SD:SAVE:DATA:MF4?

:MEDia:USB:SAVE:DATA:MF4?

[設定データを保存するとき]

:MEDia:SD:SAVE:SET?

:MEDia:USB:SAVE:SET?

[A2L設定データを保存するとき(LAN1)]

:MEDia:SD:SAVE:A2L:LAN1?

:MEDia:USB:SAVE:A2L:LAN1?

[A2L設定データを保存するとき(LAN2)]

:MEDia:SD:SAVE:A2L:LAN2?

:MEDia:USB:SAVE:A2L:LAN2?

[数値演算結果を保存するとき]

:MEDia:SD:SAVE:CALC:CSV?

:MEDia:USB:SAVE:CALC:CSV?

応答

A\$ = NONE,EXECUTING,SUCCESS_(ファイル名),FAIL

解説

データを保存します。

保存状態を返します。

NONE: 保存実行前です。

EXECUTING: 保存実行中です。

SUCCESS_(ファイル名): 保存実行に成功しました。保存したファイル名が後ろに付きます。

FAIL: 保存に失敗しました。

例

:MEDia:SD:SAVE:DATA:MEM

:MEDia:SD:SAVE:DATA:MEM?

(応答) :MEDIA:SD:SAVE:DATA:MEM? SUCCESS_TEST (ヘッダがONの場合)

注記

使用可能条件

波形データのファイルサイズが1 GBを超える場合は、約1 GBごとにファイルが自動で分割保存されます。

4.1 有線LAN接続時

有線LAN接続時の動作がおかしいとき

• ケーブルは正しく接続されていますか？

→ パソコンと本機を正しく接続してください。

• 本体のIPアドレスの設定はありますか？

→ 「2.1 機器の仕様、設定」の [有線LANの設定](#)を確認してください。

• 他の機器と同じIPアドレスになっていませんか？

→ 「2.1 機器の仕様、設定」の有線 LAN の設定に従って、[IPアドレス](#)を変更してください。

• 接続されているすべての機器の電源がONになっていますか？

→ 全ての機器の電源を確認してください。

- ✓ 本取扱説明書は、コマンドに関する部分のみを扱っています。
- ✓ LR8101.LR8102 の使用前に、必ず本体の取扱説明書をお読みください。
- ✓ LR8101.LR8102 の通信設定に関しては、本取扱説明書「2.1 機器の仕様、設定」をご覧ください。
- ✓ 本取扱説明書の内容につきましては万全を期していますが、ご不明な点や誤りなどお気づきになった点がございましたら、最寄りの営業拠点または HIOKI ウェブサイトまでご連絡ください。

(<https://www.hioki.co.jp/jp/contact/product-contact/>)

HIOKI

www.hioki.co.jp/

本社 〒386-1192 長野県上田市小泉 81

製品のお問い合わせ

 **0120-72-0560**

TEL 0268-28-0560 FAX 0268-28-0569

9:00 ~ 12:00, 13:00 ~ 17:00
土・日・祝日を除く

info@hioki.co.jp

修理・校正のお問い合わせ

ご依頼はお買上店（代理店）または最寄りの営業拠点まで
お問い合わせはサービス窓口まで

TEL 0268-28-1688 cs-info@hioki.co.jp

2103 JA



国内拠点

編集・発行 日置電機株式会社

Printed in Japan

- ・CE 適合宣言は弊社ウェブサイトからダウンロードできます。
- ・本書の記載内容を予告なく変更することがあります。
- ・本書には著作権により保護される内容が含まれます。
- ・本書の内容を無断で転記・複製・改変することを禁止します。
- ・本書に記載されている会社名・商品名などは、各社の商標または登録商標です。