

# プログラミング マニュアル

---

## 電子負荷装置

## LSG シリーズ

LSG-175 LSG-350 LSG-1050



### ■ 商標・登録商標について

本マニュアルに記載されている会社名および商品名は、それぞれの国と地域における各社および各団体の商標または登録商標です。

### ■ 取扱説明書について

本マニュアルの内容の一部または全部を転載する場合は著作権者の許諾を必要とします。また、製品の仕様および本マニュアルの内容は改善のため予告無く変更することがあります。最新版は当社ホームページを参照してください。

### ■ 本体のバージョンについて

本説明書の内容はLSGシリーズ本体のファームウェア Ver 1.27 以後に対応しております。以前の内容と一部仕様が異なりますのでご注意ください。変更内容については巻末の変更履歴を参照してください。

# 目次

<b>第 1 章 概要</b> .....	<b>1</b>
1-1. フロントパネル外観 .....	1
1-2. リアパネル外観 .....	1
<b>第 2 章 インタフェース設定</b> .....	<b>2</b>
2-1. USB インタフェース設定 .....	2
2-2. RS-232C インタフェース設定 .....	2
2-3. GP-IB インタフェース設定 .....	3
2-4. RS-232C/USB リモートコントロール機能チェック .....	4
2-5. Realterm を使用してリモート接続を確認する .....	4
2-6. GP-IB リモートコントロール機能チェック .....	7
<b>第 3 章 コマンド概要</b> .....	<b>9</b>
3-1. コマンドシンタックス .....	9
<b>第 4 章 コマンド詳細</b> .....	<b>12</b>
4-1. 共通コマンド .....	12
4-1-1. *CLS .....	12
4-1-2. *ESE .....	12
4-1-3. *ESR .....	13
4-1-4. *IDN .....	13
4-1-5. *OPC .....	13
4-1-6. *RCL .....	14
4-1-7. *RST .....	14
4-1-8. *SAV .....	14
4-1-9. *SRE .....	14
4-1-10. *STB .....	15
4-1-11. *TRG .....	15
4-1-12. *TST .....	15
4-2. トリガコマンド .....	16
4-2-1. :ABOrt .....	16
4-2-2. :INPut[:STATE]:TRIGgered .....	18
4-2-3. :INITiate[:IMMediate] .....	16
4-2-4. :INITiate:CONTInuous .....	16
4-2-5. :TRIGger[:Delay]:Time .....	16
4-2-6. :TRIGger[:PULSe]:WIDTh .....	17
4-3. 入力コマンド .....	17
4-3-1. :INPut .....	17
4-3-2. :INPut:MODE .....	18
4-3-3. [:INPut]:SHOrt .....	19
4-4. 測定コマンド .....	19
4-4-1. :MEASure:CURRent .....	19

4-4-2. :MEASure:ETIME .....	19
4-4-3. :MEASure:POWER.....	19
4-4-4. :MEASure:VOLTage .....	20
4-5. フェッチコマンド .....	20
4-5-1. :FETCh:CURRent .....	20
4-5-2. :FETCh:POWER.....	20
4-5-3. :FETCh:VOLTage .....	20
4-6. 設定サブシステム・コマンド .....	21
4-6-1. [:CONFigure]:OCP .....	21
4-6-2. [:CONFigure]:OPP .....	21
4-6-3. [:CONFigure]:UVP .....	22
4-6-4. [:CONFigure]:UVP:TIME .....	22
4-6-5. [:CONFigure]:OVP .....	23
4-6-6. [:CONFigure]:SStart.....	23
4-6-7. [:CONFigure]:VON.....	23
4-6-8. [:CONFigure]:VDElay.....	24
4-6-9. :CONFigure:RESPonse .....	24
4-6-10. [:CONFigure]:CNTime.....	25
4-6-11. [:CONFigure]:COTime .....	25
4-6-12. [:CONFigure]:CRUNit.....	26
4-6-13. :CONFigure:DYNamic.....	26
4-6-14. :CONFigure:MEMory .....	26
4-6-15. :CONFigure:SHORT .....	27
4-6-16. :CONFigure:SHORT:SAFety.....	27
4-6-17. :CONFigure:SHORT:FUNCTION.....	27
4-6-18. [:CONFigure]:GNG:SPECTest.....	28
4-6-19. [:CONFigure]:GNG:DTIME .....	28
4-6-20. [:CONFigure]:GNG:MODE .....	29
4-6-21. [:CONFigure]:GNG:[PASS] .....	29
4-6-22. [:CONFigure]:GNG:H .....	29
4-6-23. [:CONFigure]:GNG:L.....	30
4-6-24. [:CONFigure]:GNG:C .....	30
4-7. パラレルコマンド .....	31
4-7-1. [:CONFigure]:PARAllel .....	31
4-8. ステップコマンド .....	31
4-8-1. [:CONFigure]:STEP:CC .....	31
4-8-2. [:CONFigure]:STEP:CCH.....	32
4-8-3. [:CONFigure]:STEP:CCM .....	32
4-8-4. [:CONFigure]:STEP:CCL .....	33
4-8-5. [:CONFigure]:STEP:CR .....	33
4-8-6. [:CONFigure]:STEP:CRH.....	33
4-8-7. [:CONFigure]:STEP:CRM .....	34
4-8-8. [:CONFigure]:STEP:CRL .....	34
4-8-9. [:CONFigure]:STEP:CV .....	35
4-8-10. [:CONFigure]:STEP:CVH.....	35
4-8-11. [:CONFigure]:STEP:CVL .....	35

4-8-12. [:CONFigure]:STEP:CP .....	36
4-8-13. [:CONFigure]:STEP:CPH .....	36
4-8-14. [:CONFigure]:STEP:CPM .....	36
4-8-15. [:CONFigure]:STEP:CPL .....	37
4-9. 外部制御コマンド .....	37
4-9-1. [:CONFigure]:EXTErnal[:CONTRol] .....	37
4-9-2. [:CONFigure]:EXTErnal:LOADonin .....	38
4-10. モード・サブシステム・コマンド .....	38
4-10-1. :MODE .....	38
4-10-2. [:MODE]:CRANge .....	39
4-10-3. [:MODE]:VRANge .....	39
4-10-4. [:MODE]:RESPonse .....	39
4-10-5. [:MODE]:DYNAmic .....	40
4-11. 電流サブシステム・コマンド .....	40
4-11-1. :CURRent[:VA] .....	40
4-11-2. :CURRent[:VA]:TRIGgered .....	41
4-11-3. :CURRent:VB .....	41
4-11-4. :CURRent:SRATe .....	41
4-11-5. :CURRent:L1 .....	42
4-11-6. :CURRent:L2 .....	42
4-11-7. :CURRent:SET .....	43
4-11-8. :CURRent:LEVel .....	43
4-11-9. :CURRent:RISE .....	44
4-11-10. :CURRent:FALL .....	44
4-11-11. :CURRent:T1 .....	44
4-11-12. :CURRent:T2 .....	45
4-11-13. :CURRent:FREQuency .....	45
4-11-14. :CURRent:DUTY .....	46
4-11-15. :CURRent:RECall .....	46
4-12. 抵抗サブシステム・コマンド .....	47
4-12-1. :RESistance[:VA] .....	47
4-12-2. :RESistance[:VA]:TRIGgered .....	47
4-12-3. :RESistance:VB .....	48
4-12-4. :RESistance:SRATe .....	48
4-12-5. :RESistance:L1 .....	48
4-12-6. :RESistance:L2 .....	49
4-12-7. :RESistance:SET .....	49
4-12-8. :RESistance:LEVel .....	50
4-12-9. :RESistance:RISE .....	50
4-12-10. :RESistance:FALL .....	51
4-12-11. :RESistance:T1 .....	51
4-12-12. :RESistance:T2 .....	52
4-12-13. :RESistance:FREQuency .....	52
4-12-14. :RESistance:DUTY .....	53
4-12-15. :CONDuctance[:VA] .....	53
4-12-16. :CONDuctance[:VA]:TRIGgered .....	54

4-12-17. :CONDUCTance:VB .....	54
4-12-18. :CONDUCTance:L1 .....	54
4-12-19. :CONDUCTance:L2 .....	55
4-12-20. :CONDUCTance:SET .....	55
4-12-21. :CONDUCTance:RECall .....	56
4-12-22. :RESistance:RECall .....	56
4-13. 電圧サブシステム・コマンド .....	57
4-13-1. :VOLTage[:VA] .....	57
4-13-2. :VOLTage:VB .....	57
4-13-3. :VOLTage:RECall .....	58
4-14. 電力サブシステム・コマンド .....	58
4-14-1. :POWER[:VA] .....	58
4-14-2. :POWER:VB .....	59
4-14-3. :POWER:L1 .....	59
4-14-4. :POWER:L2 .....	60
4-14-5. :POWER:SET .....	60
4-14-6. :POWER:LEVel .....	61
4-14-7. :POWER:T1 .....	61
4-14-8. :POWER:T2 .....	62
4-14-9. :POWER:FREQuency .....	62
4-14-10. :POWER:DUTY .....	63
4-14-11. :POWER:RECall .....	63
4-15. プログラム・コマンド .....	64
4-15-1. :FUNCTion[:COMPLete][:RING]:TIME .....	64
4-15-2. :PROGram .....	65
4-15-3. [:PROGram]:CHAin .....	65
4-15-4. [:PROGram]:CHAin:P2P .....	66
4-15-5. [:PROGram]:CHAin[:RECall]:DEFault .....	66
4-15-6. [:PROGram]:CHAin:STARt .....	66
4-15-7. :PROGram:MEMory .....	67
4-15-8. :PROGram:OFFTime .....	67
4-15-9. :PROGram:ONTime .....	68
4-15-10. :PROGram:PFTime .....	68
4-15-11. :PROGram[:RECall]:DEFault .....	68
4-15-12. :PROGram:RUN .....	69
4-15-13. :PROGram:SAVE .....	69
4-15-14. :PROGram:STARt .....	69
4-15-15. :PROGram:STATe .....	69
4-15-16. :PROGram:STEP .....	70
4-15-17. :PROGram:STIME .....	70
4-16. ノーマルシーケンス・コマンド .....	71
4-16-1. :NSEquence .....	71
4-16-2. :NSEquence:CHAin .....	72
4-16-3. :NSEquence[:DEL]:ALL .....	73
4-16-4. :NSEquence:EDIT .....	73
4-16-1. :NSEquence:EDIT:POINT .....	74

4-16-1. :NSEquence:EDIT:END .....	74
4-16-2. :NSEquence:LAST .....	74
4-16-3. :NSEquence:LLOad .....	75
4-16-4. :NSEquence:LOOP .....	75
4-16-5. :NSEquence:MEMO .....	75
4-16-6. :NSEquence:MODE .....	76
4-16-7. :NSEquence:NUMBer .....	76
4-16-8. :NSEquence:RANGe.....	77
4-16-9. :NSEquence:SAVE .....	77
4-16-10. :NSEquence:START .....	77
4-16-11. :NSEquence:STATe.....	78
4-16-1. :NSEquence:COTime.....	78
4-17. ファストシーケンス・コマンド .....	79
4-17-1. :FSEquence .....	79
4-17-2. :FSEquence[:DEL]:ALL .....	80
4-17-3. :FSEquence:EDIT .....	80
4-17-4. :FSEquence:EDIT:POINT .....	80
4-17-5. :FSEquence:EDIT:END.....	81
4-17-6. :FSEquence[:EDIT]:FILL .....	81
4-17-7. :FSEquence:LAST .....	81
4-17-8. :FSEquence:LLOad.....	82
4-17-9. :FSEquence:LOOP .....	82
4-17-10. :FSEquence:MEMO .....	83
4-17-11. :FSEquence:MODE.....	83
4-17-12. :FSEquence:RANGe .....	83
4-17-13. :FSEquence:RPTStep.....	84
4-17-14. :FSEquence:SAVE .....	84
4-17-15. :FSEquence:TBASe .....	84
4-17-16. :FSEquence:STATe .....	85
4-18. OCP テスト コマンド .....	85
4-18-1. :OCP:STATe .....	85
4-18-2. :OCP:EDIT[:CHANnel] .....	86
4-18-3. :OCP[:CHANnel]:NUMBer .....	86
4-18-4. :OCP[:CHANnel]:RANGe.....	87
4-18-5. :OCP[:CHANnel]:START.....	87
4-18-6. :OCP[:CHANnel]:END.....	87
4-18-7. :OCP[:CHANnel]:STEP:CURRent.....	88
4-18-8. :OCP[:CHANnel]:LAST .....	88
4-18-9. :OCP[:CHANnel]:STEP:TIME .....	88
4-18-10. :OCP[:CHANnel]:DELay .....	89
4-18-11. :OCP[:CHANnel]:TRIGger .....	89
4-18-12. :OCP:CHANnel:STATus .....	89
4-18-13. :OCP:RESult.....	90
4-18-14. :OCP:SAVE .....	90
4-18-15. :OCP:RUN .....	90
4-19. ユーティリティ・コマンド .....	90

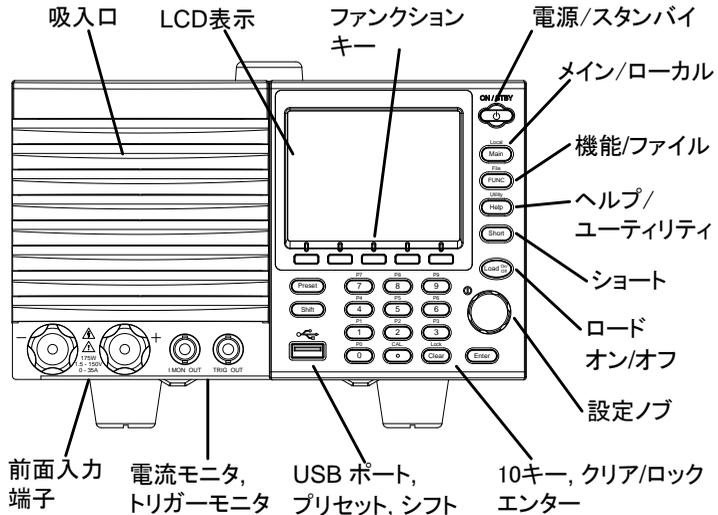
4-19-1.	:UTILity:ALARm .....	90
4-19-2.	:UTILity:BRlghtness .....	91
4-19-3.	:UTILity:CONTrast .....	91
4-19-4.	:UTILity:GNG .....	91
4-19-5.	:UTILity:KNOB .....	92
4-19-6.	:UTILity:LANGUage .....	92
4-19-7.	:UTILity:LOAD .....	92
4-19-8.	:UTILity:LOAD:MODE .....	93
4-19-9.	:UTILity:LOAD:RANGe .....	93
4-19-10.	:UTILity:REMOte .....	94
4-19-11.	:UTILity:REMOte:MODE .....	94
4-19-12.	:UTILity:SPEAker .....	94
4-19-13.	:UTILity:SYSTem .....	94
4-19-14.	:UTILity:TIME .....	95
4-19-15.	:UTILity:UNReg .....	95
4-20.	インターフェース・コマンド .....	96
4-20-1.	:UTILity:BRATe .....	96
4-20-2.	:UTILity:INTerface .....	96
4-20-3.	:UTILity:PARity .....	97
4-20-4.	:UTILity:SBIT .....	97
4-21.	ファイル・コマンド .....	98
4-21-1.	:FACTory[:RECall] .....	98
4-21-2.	:MEMory:RECall .....	98
4-21-3.	:MEMory:SAVE .....	98
4-21-4.	:PREset:RECall .....	98
4-21-5.	:PREset:SAVE .....	99
4-21-6.	:SETup:RECall .....	99
4-21-7.	:SETup:SAVE .....	99
4-21-8.	:USER[:DEFault]:RECall .....	99
4-21-9.	:USER[:DEFault]:SAVE .....	99
4-22.	SCPI レジスタ・コマンド .....	100
4-22-1.	:SYSTem:ERRor .....	100
4-22-2.	:STATus:PRESet .....	100
4-23.	Csummary ステータス・コマンド .....	101
4-23-1.	:STATus:CSUMmary:CONDition .....	101
4-23-2.	:STATus:CSUMmary:ENABLE .....	101
4-23-3.	:STATus:CSUMmary[:EVENT] .....	101
4-23-4.	:STATus:CSUMmary:NTRansition .....	102
4-23-5.	:STATus:CSUMmary:PTRansition .....	102
4-24.	Operation ステータス・コマンド .....	103
4-24-1.	:STATus:OPERation:CONDition .....	103
4-24-2.	:STATus:OPERation:ENABLE .....	103
4-24-3.	:STATus:OPERation[:EVENT] .....	103
4-24-4.	:STATus:OPERation:NTRansition .....	104
4-24-5.	:STATus:OPERation:PTRansition .....	104
4-25.	Questionable ステータス・コマンド .....	105

4-25-1. :STATus:QUEStionable:CONDition .....	105
4-25-2. :STATus:QUEStionable:ENABle.....	105
4-25-3. :STATus:QUEStionable[:EVENT].....	105
4-25-4. :STATus:QUEStionable:NTRansition.....	106
4-25-5. :STATus:QUEStionable:PTRansition.....	106
<b>第 5 章 ステータス レジスタの概要 .....</b>	<b>107</b>
5-1. ステータス レジスタの紹介 .....	107
5-2. ステータス レジスタの構成 .....	108
5-3. Csummary ステータス レジスタ グループ .....	109
5-4. Operation ステータス レジスタ グループ .....	110
5-5. Questionable ステータス レジスタ グループ .....	111
5-6. Standard イベントステータス レジスタ グループ .....	112
5-7. ステータス レジスタ グループ .....	114
<b>第 6 章 付録 .....</b>	<b>116</b>
6-1. エラーメッセージ .....	116
6-2. 変更履歴 .....	120

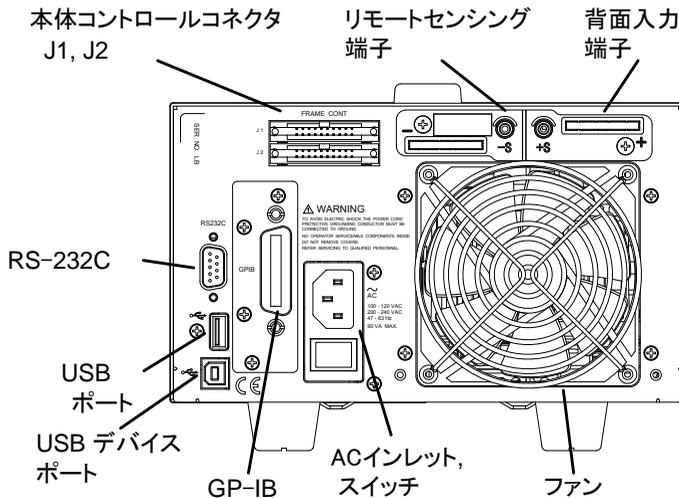
# 第1章 概要

本マニュアルは LSG シリーズのリモートコマンドについて説明したものです。

## 1-1. フロントパネル外観



## 1-2. リアパネル外観



## 第2章 インタフェース設定

### 2-1. USB インタフェース設定

USB 構成	PC 側 コネクタ	Type A, ホスト
	LSG 側コネクタ	背面 Type B, デバイスポート
	スピード	2.0 (full speed)
	USB クラス	CDC クラス



注意

USB リモートコントロールを使用する前に、添付のアクセサリ CD に収録されている USB デバイスドライバをインストールする必要があります。

- 操作
1. リアパネルの USB B ポートに USB ケーブルを接続します。
  2.  +  > *Interface[F3]* を押します。  
*Interface* を *USB* に設定します。
  3. PC からは仮想 COM ポート接続で認識されます。  
詳細は取扱説明書を参照してください。

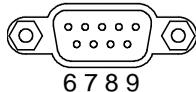
### 2-2. RS-232C インタフェース設定

RS-232C 構成	コネクタ	DB-9, オス
	ボーレート(bps)	2400, 4800, 9600, 19200, 38400
	ストップビット	1, 2
	パリティ	None, Odd, Even

- 操作
1. PC から背面パネルの RS232 ポートに RS-232C ケーブルを接続します。
  2.  +  > *Interface[F3]* を押します。  
*Interface* を *RS232* に設定します。
  4. *Baud Rate*, *Stop Bit* と *Parity* を設定します。

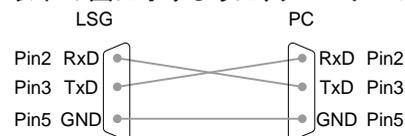
ピンアサイン

1	2	3	4	5	2: RxD (Receive data)
					3: TxD (Transmit data)
					5: GND
6	7	8	9		4,6,7,8,9: 未接続



PC 接続

以下の図に示すように、クロスケーブル接続を使用します。



## 2-3. GP-IB インタフェース設定

GP-IB を使用するには、オプションの GP-IB ポートを設定する必要があります。詳細については、ユーザーマニュアルを参照してください。

操作

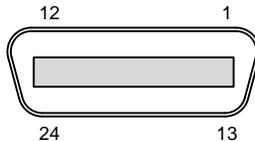
1. 本器の電源がオフになっていることを確認します。
2. GP-IB コントローラから本器の GP-IB ポートに GP-IB ケーブルを接続します。
3. 本器の電源をオンにします。
4.   +  > *Interface[F3]* を押します。  
*Interface* を *GP-IB* に設定します。
5. GP-IB アドレスを設定します。

GP-IB address 0~30

GP-IB の制限

- 最大 15 台、合計 20 メートル以下で、各機器間のケーブル長は 2m です。
- 個別なアドレスを各デバイスに割り当てます。重複設定はできません。
- 装置のうちの少なくとも 2/3 は電源がオンになっている必要があります。
- ループ接続または並列接続はできません。

ピンアサイン



ピン	信号	ピン	信号
1~4	Data I/O 1~4	13~16	Data I/O 5~8
5	EOI	17	REN
6	DAV	18	Ground (DAV)
7	NRFD	19	Ground (NRFD)
8	NDAC	20	Ground (NDAC)
9	IFC	21	Ground (IFC)
10	SRQ	22	Ground (SRQ)
11	ATN	23	Ground (ATN)
12	SHIELD Ground	24	Single GND

## 2-4. RS-232C/USB リモートコントロール機能チェック

機能チェック	<p>Realterm や Putty などの通信ソフトを起動します。 RS-232C の場合、COM ポート、ボーレート、ストップビット、データビットとパリティを設定します。 Windows の COM ポートの設定を確認するには、コントロールパネルからデバイスマネージャのポートを参照してください。</p> <hr/> <p>通信ソフトから次のクエリコマンドを実行します。 *idn?[LF] メーカー、モデル番号、シリアル番号、ファームウェアのバージョンを以下の形式で返します。 TEXIO,LSG,XXXXXXXXXXXXX, V.X.X.X.X メーカー: TEXIO モデル: LSG シリアル : XXXXXXXXXXXXXXX ファームウェアバージョン : V.X.X.X</p>
--------	--

## 2-5. Realterm を使用してリモート接続を確認する

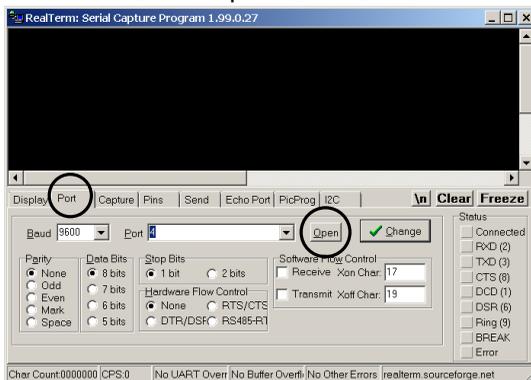
背景	<p>Realterm は、PC のシリアルポートに接続されたデバイス、または USB 経由でエミュレートされるシリアルポートを介して通信するために使用できる通信ソフトです。 次の手順は、Realterm バージョン 1.99.0.27 に適用されます。</p>
 注意	<p>Realterm は Sourceforge.net 上で無料でダウンロードすることができます。詳細については、 <a href="http://realterm.sourceforge.net/">http://realterm.sourceforge.net/</a>を参照してください。</p>
操作	<ol style="list-style-type: none"><li>1. Realterm をダウンロードし Realterm ウェブサイト上の指示に従ってインストールしてください。</li><li>2. USB または RS-232C を介して LSG を接続します。</li><li>3. RS-232C を使用している場合は、LSG に設定されたボーレート、ストップビットとパリティをメモしておきます。</li></ol>

4. Windows のデバイスマネージャを開き、接続するための COM ポート番号を確認してください。  
ポートアイコンをダブルクリックし、接続されたシリアルポートデバイスまたは USB の仮想 COM の接続された COM ポートを開きます。



ボーレート、ストップビットおよびパリティ設定は右クリックで接続されたデバイスのプロパティを開き、ポートの設定で選択することができます。COM ポートの変更は詳細設定で行います。

5. 管理者として PC 上 Realterm を実行します。  
スタートメニューの Realterm アイコンを表示させ、右クリックで表示される“管理者として実行”を選択します。
6. Realterm が起動したら、Port タブをクリックします。  
Baud, Parity, Data bits, Stop bits、および接続用の Port 番号の設定を入力します。  
ハードウェアフロー制御、ソフトウェアフロー制御オプションはデフォルト設定のままにすることができます。  
LSG に接続するには Open を押します。





## 2-6. GP-IB リモートコントロール機能チェック

### 機能チェック

GP-IB 機能を確認するために、ナショナルインスツルメンツの Measurement & Automation Controller ソフトウェアを使用してください。ナショナルインスツルメンツの Web サイト(<http://www.ni.com>)を参照してください。

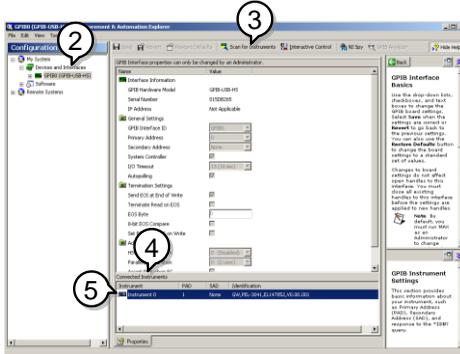
### 操作

1. NI Measurement and Automation エクスプローラ(MAX) を開始するにはデスクトップの NI Measurement and Automation Explorer (MAX)アイコンを押します。  
スタート>すべてのプログラム>National Instruments>Measurement & Automation

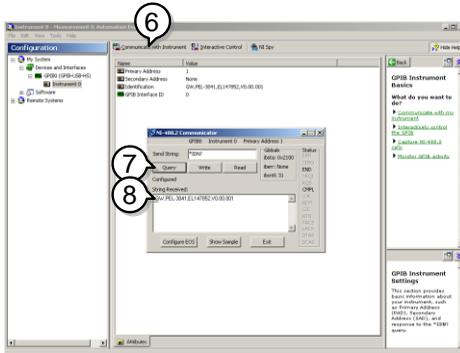


2. コンフィギュレーションパネルからアクセスします  
My System>Devices and Interfaces>GPIB0
3. Scan for Instruments ボタンを押します。
4. Connected Instruments パネルに LSG-XXXX が設定された Instrument 0 と同じアドレスで Instrument 0 として認識されています。

- Instrument 0 アイコンをダブルクリックします。



- Communicate with Instrument をクリックします。
- NI-488.2 の Communicator ウィンドウを開きます。送信テキストボックスに\*IND?が入力されていることを確認します。クエリボタンを押しくエリコマンド\*IDN?を機器へ送信します。
- 受信テキストボックスにクエリの応答が表示されます:  
TEXIO, LSG-XXXX,PXXXXXX,V1.XX  
(メーカー、モデル名、シリアル番号、バージョン)



- 機能チェックが完了しました。

## 第3章 コマンド概要

この章では、個々のコマンド説明におけるコマンドシンタックス(構文)について説明します。

### 3-1. コマンドシンタックス

対応規格	IEEE488.2 準拠 SCPI 1999 準拠
コマンド構造	SCPI(プログラマブル計測器用標準コマンド)コマンドは、ツリー状の構造、ノードに編成に従っています。コマンド・ツリーの各レベルはノードです。SCPIコマンドの各キーワードは、コマンド・ツリー内の各ノードを表します。SCPIコマンドの各キーワード(ノード)はコロンの(:)で区切られています。以下の図では、SCPI サブ構造とコマンドの例を示しています。

```
[[:MODE]:CRANGE]
```

:MODE

```
:CRANGE :VRANGE :RESPonse
```

コマンドの種類	いくつかのコマンドとクエリがあります。コマンドは、ユニットへの命令やデータを送信し、クエリがユニットからのデータまたは状況情報を受信します。
---------	--

#### Command types

単純コマンド	単一のコマンド パラメータ付き/なし
例	:CONFigure:RESPonse MAX
クエリ	クエリは、疑問符(?)が付く単純または複合コマンドです。パラメータ(データ)が返されます。
例	:CONFigure:RESPonse?
複合コマンド	同じコマンドラインで複数のコマンド。 複合コマンドはセミコロン(; )またはセミコロンとコロンの(;: ) のいずれかで区切られています。セミコロンは、2つの関連コマンドを結合するために使用され、注意として最後のコマンドは最初のコマンドの最後のノードで開始する必要があります。セミコロンとコロンは、異なるノードからの二つのコマンドを組み合わせるために使用されます。

例 CONFigure:VON  
MAX;:CONFigure:VDElay MIN

コマンド形式 コマンドとクエリは、ロングとショートとの2つの異なる形式を持っています。  
コマンド構文は、コマンドの省略形の大文字と小文字の残りの部分(長い形式)で書かれています。コマンドは大文字または小文字、ショートまたはロングのフォームで書き込むことができます。不完全なコマンドは認識されません。  
以下に正しく書き込まれたコマンドの例を示します。

ロングフォーム	ショートフォーム
:CURRENT:LEVel?	:CURR:LEV?
:current:level?	:curr:lev?

角カッコ 角カッコが含まれているコマンドは、内容がオプションであることを示します。下に示すようにコマンドの機能は角カッコで囲まれた項目の有無にかかわらず同じです。  
クエリの例を示します。  
“[:CONFigure]:GNG [:PASS]?”  
“:CONFigure:GNG:PASS?” と “:GNG?” は両方とも有効です。

コマンドの形式 :CURRENT:Set 1.00A

- |             |            |
|-------------|------------|
| 1. コマンド・ヘッダ | 3. パラメーター1 |
| 2. スペース     | 4. 単位      |

共通の	タイプ	説明	例
入力パラメータ	<Boolean>	ブール論理	0, 1
	<NR1>	整数	0, 1, 2, 3
	<NR2>	小数	0.1, 3.14, 8.5
	<NR3>	浮動小数点	4.5e-1, 8.25e+1
	<NRf>	NR1, 2, 3 のいずれか	1, 1.5, 4.5e-1
	[MIN]	コマンドの場合、最低値に設定します。このパラメータは、示された任意の数値パラメータの代わりに使用することができます。クエリの場合、それは特定の設定で許可される最小値を返します。	

[MAX] (オプションパラメータ) コマンドの場合、最大値に設定します。このパラメータは、示された任意の数値パラメータの代わりに使用することができます。クエリの場合、それは特定の設定で使用できる最大値を返します。

単位 (オプションパラメータ) 単位は、必要に応じてほとんどの NRF 型の入力パラメータに使用することができます。

[A]	アンペア	1.00A
[%]	パーセント	10%
[V]	ボルト	5.00V
[W]	ワット	3.00W
[ms]	ミリ秒	20ms
[mV]	ミリボルト	150mV
[s]	秒	5s
[Ω]	オーム	50Ω
[MHO]	ジーメンズ	0.02MHO
[mS]	ミリジーメンズ	20mS
[mA/uS]	ミリアンペア/マイクロ秒	100mA/uS
[Hz]	ヘルツ	6.0e+1Hz

---

メッセージ ターミネーター	LF	改行コード (0x0A)
------------------	----	--------------

---

## 第4章 コマンド詳細

### 4-1. 共通コマンド

#### 4-1-1. \*CLS

Set →

説明	全てのイベントレジスタとキューを初期値にします。
構文	*CLS
例	*CLS 全てのイベントレジスタとキューを初期値にします。

Set →

#### 4-1-2. \*ESE

→ Query

説明	Standard イベントステータス・イネーブルレジスタ(ESE)の設定とクエリ。 対応するビットを1にするとイベントが発生したときにはステータスバイト レジスタのイベント・サマリ・ビット(ESB)を1にすることができます。 ビットの詳細は Standard イベントステータス レジスタ グループを参照して下さい。	
構文	*ESE <NRf>	
クエリ構文	*ESE?	
パラメーター	<NR1>	Standard イベントステータス・イネーブルレジスタの設定
応答	Standard イベントステータス・イネーブルレジスタの設定値を”<NR1>”で返します。	
例	*ESE 8 Standard イベントステータス・イネーブルレジスタにビット 3 を設定します。	
クエリ 例	*ESE? >12 ビット 2 と 3 が Standard イベントステータス・イネーブルレジスタに設定されています。	

### 4-1-3. \*ESR

→ Query

説明	Standard イベントステータス・レジスタ(ESR)を読み出します。このコマンドは、読み出した後に ESR をクリアします。 ビットの詳細は Standard イベントステータス レジスタ グループを参照して下さい。
クエリ構文	*ESR?
応答	Standard イベントステータス・レジスタ値を"<NR1>"で返します。
クエリ 例	*ESR? >48 ビット 4 と 5 の実行エラーとコマンドエラーが Standard イベント・レジスタに設定されています。

### 4-1-4. \*IDN

→ Query

説明	機器メーカー、モデル番号、シリアル番号、ファームウェアバージョンを応答します。
クエリ構文	*IDN?
応答	<ASCII string> メーカー名を返します。 <ASCII string> モデル名を返します。 <NR1> シリアル番号を返します。 <ASCII string> ファームウェアのバージョンを返します。
クエリ 例	*IDN? >TEXIO,LSG-175,12345678,V1.01.001 メーカー名、モデル名、シリアル番号、ファームウェアバージョンを応答します。

Set →

### 4-1-5. \*OPC

→ Query

説明	このコマンドでは、機器はすべての保留中の操作を完了した後、Standard イベントステータス レジスタの OPC ビット (ビット 0) を 1 にします。クエリは OPC ビットのステータスを返します。
構文	*OPC
クエリ構文	*OPC?
応答	1 操作完了
例	*OPC
クエリ 例	*OPC? >1 保留中のすべての操作が完了していることを示します。

## 4-1-6. \*RCL

Set →

説明	このコマンドは、以前に保存されているメモリの設定から機器設定を呼び出します。	
構文	*RCL <NR1>	
パラメーター	<NR1>	メモリー番号 1 ~ 256
例	*RCL 20 メモリ 20 をリコールします。	
同一機能コマンド	:MEMory:RECall	

## 4-1-7. \*RST

Set →

説明	本器をリセットします。 このコマンドは :ABORt と *CLS コマンドです。	
構文	*RST	
例	*RST 本器をリセットします。	

## 4-1-8. \*SAV

Set →

説明	指定したメモリ番号に機器設定を保存します。	
構文	*SAV <NR1>	
パラメーター	<NR1>	メモリー番号 1 ~ 256
例	*SAV 20 メモリ 20 に現在の設定を保存します。	
同一機能コマンド	:MEMory:SAVE	

## 4-1-9. \*SRE

Set →

→ Query

説明	サービスリクエスト イネーブルレジスタ(SRE)を要求または設定します。 サービスリクエスト イネーブルレジスタの各ビットを1とすると、イベントが発生したときにステータスバイト レジスタ(STB)のマスター・サマリ・ビット(MSB)を1にすることができます。  ビットの詳細はステータス レジスタ グループを参照して下さい。	
構文	*SRE <NRf>	
クエリ構文	*SRE?	
パラメーター	<NR1>	サービスリクエスト イネーブルレジスタの設定

応答	サービスリクエスト イネーブルレジスタの設定値を"<NR1>"返します。
例	*SRE 8 サービスリクエスト イネーブルレジスタにビット 3 を設定します。
クエリ 例	*SRE? >12 ビット 2 と 3 のサービスリクエスト イネーブルレジスタが設定されています。

#### 4-1-10. \*STB

→ Query

説明	ステータスバイト レジスタ(STB)を読み出します。このコマンドは、ステータスバイト レジスタを読み出ししてもレジスタはクリアされません。 マスタ・サマリステータスビット(MSS)が設定されている場合は、サービス要求の理由があることを示しています。 ビットの詳細はステータス レジスタ グループを参照して下さい。
クエリ構文	*STB?
応答	ステータスバイト レジスタの値を"<NR1>"で返します。
クエリ 例	*STB? >36 ビット 2 と 5 がステータスバイト レジスタに設定されています。

#### 4-1-11. \*TRG

Set →

説明	強制トリガを発行します。
構文	*TRG
例	*TRG 強制トリガを発行します。
関連コマンド	:INITiate:CONTinuous, :INITiate[:IMMEDIATE]

#### 4-1-12. \*TST

→ Query

説明	標準の SCPI セルフテストコマンドです。本器は任意のセルフテストを実行しませんので、必ずこのコマンドに 0 を(エラーなし)を返します。
クエリ構文	*TST?
応答	0 エラーなし
クエリ 例	*TST? >0

## 4-2. トリガコマンド

### 4-2-1. :ABORt

Set →

説明	トリガ待ちをクリアしてアイドル状態になります。
構文	:ABORt
例	ABOR アイドル状態になります。

### 4-2-2. :INITiate[:IMMediate]

Set →

説明	トリガ待ち状態に遷移します。トリガがアクティブ化されたときにトリガ待ちを解除します。
構文	:INITiate[:IMMediate]
例	: INIT トリガ待ち状態に遷移します。
関連コマンド	*TRG, :INPut[:STATe]:TRIGgered, :CURRent[:VA]:TRIGgered, :RESistance[:VA]:TRIGgered

Set →

### 4-2-3. :INITiate:CONTInuous

→ Query

説明	トリガの連続待ち状態の設定とクエリ。注意:トリガ待ち状態の解除はトリガのアクティブ化が必要です。
構文	:INITiate:CONTInuous {<Boolean>   OFF   ON}
クエリ構文	:INITiate:CONTInuous?
パラメーター	OFF / 0      トリガの連続待ちを解除します。 ON / 1        トリガの連続待ちに設定します。
応答	トリガの連続待ちの設定値を"<Boolean>"で返します。
例	:INIT:CONT ON トリガの連続待ちに設定します。
クエリ 例	:INIT:CONT? >1 トリガ連続待ちの設定をしトリガを待ちます。
関連コマンド	*TRG, :INPut[:STATe]:TRIGgered, :CURRent[:VA]:TRIGgered, :RESistance[:VA]:TRIGgered

Set →

### 4-2-4. :TRIGger[:Delay]:Time

→ Query

説明	トリガの実行までの遅延時間を設定します。
構文	:TRIGger[:Delay]:Time <NR2> MINimum MAXimum

クエリ構文	:TRIGger[:Delay]:Time? [MINimum MAXimum]	
パラメーター	<NR2>	0 ~ 0.005s (0 ~ 5000 $\mu$ s)
	MINimum	最小遅延時間を設定します。
	MAXimum	最大遅延時間を設定します。
応答	遅延時間を応答します。	
例	:TRIG:T MAX トリガ遅延時間を最大に設定します。	
クエリ 例	:TRIG:T? >0.0050000 トリガ遅延時間は 5ms です。	

Set →

#### 4-2-5. :TRIGger[:PULSe]:WIDTh

→ Query

説明	トリガ出力のパルス幅を設定します。	
構文	:TRIGger[:PULSe]:WIDTh <NR2> MINimum MAXimum	
クエリ構文	:TRIGger[:PULSe]:WIDTh? [MINimum MAXimum]	
パラメーター	<NR2>	0.0000025~0.005s (2.5 $\mu$ s ~ 5000 $\mu$ s)
	MINimum	最小時間を設定します。
	MAXimum	最大時間を設定します。
応答	パルス幅時間を応答します。	
例	:TRIG:WIDTh MAX	
クエリ 例	:TRIG:WIDTh? >0.0050000 パルス幅は 5ms です。	

### 4-3. 入力コマンド

Set →

#### 4-3-1. :INPut

→ Query

説明	負荷入力のオンとオフの設定とクエリ	
構文	:INPut {<Boolean>   OFF   ON}	
クエリ構文	:INPut?	
パラメーター	OFF / 0	負荷入力のオフにします。 プログラム・シーケンス・OCP テストを中止します。
	ON / 1	負荷入力のオン設定 プログラム・シーケンス・OCP テストを開始します。
応答	負荷入力の設定値を"<Boolean>"で返します。	

例 :INP ON  
負荷入力をオンに設定します。

クエリ 例 :INP?  
>1  
負荷入力設定はオンです。

Set →

#### 4-3-1. :INPut[:STATe]:TRIGgered

→ Query

説明 トリガがアクティブ化された時に負荷入力をオンにするか設定します。

構文 :INPut[:STATe]:TRIGgered {<Boolean> | OFF | ON}

クエリ構文 :INPut[:STATe]:TRIGgered?

パラメーター OFF / 0 トリガアクティブ時に負荷入力を変更しない。  
ON / 1 トリガアクティブ時に負荷入力をオンにします。

応答 0 トリガアクティブ時に負荷入力を変更しない。  
1 トリガアクティブ時に負荷入力をオンにします。

例 :INP:TRIG ON  
トリガアクティブ時に負荷入力をオンにします。

クエリ 例 :INP:TRIG?  
>1  
トリガアクティブ時に負荷入力をオンにします。

関連コマンド \*TRG, :INITiate:CONTinuous, :INITiate[:IMMEDIATE]

Set →

#### 4-3-2. :INPut:MODE

→ Query

説明 負荷の動作モードの設定とクエリ。設定した負荷の動作モードに切り替わります。

構文 :INPut:MODE { LOAD | PROG | NSEQ | FSEQ }

クエリ構文 :INPut:MODE?

パラメーター/応答 LOAD 通常動作モードの設定  
PROG プログラム動作モードの設定  
NSEQ ノーマルシーケンス動作モードの設定  
FSEQ ファストシーケンス動作モードの設定

例 :INP:MODE LOAD  
通常動作モードに設定します。

クエリ 例 :INP:MODE?  
>LOAD  
動作モードは通常動作モードです。

Set →

→ Query

## 4-3-3. :INPut:SHORT

説明	負荷ショートのアオンとオフの設定とクエリ
構文	:INPut:SHORT {<Boolean>   OFF   ON}
クエリ構文	:INPut:SHORT?
パラメーター	OFF / 0 負荷ショートのアオフ設定 ON / 1 負荷ショートのアオン設定
応答	負荷ショートの設定を”<Boolean>”で返します。
例	:INP:SHOR ON 負荷ショートをオンに設定します。
クエリ 例	:INP:SHOR? >1 負荷ショート設定はオンです。

## 4-4. 測定コマンド

## 4-4-1. :MEASure:CURRent

→ Query

説明	電流測定のアクエリ
クエリ構文	:MEASure:CURRent?
応答	電流測定を”<NR2>”で返します。(単位 [A])
クエリ 例	:MEAS:CURR? >0.50000 電流測定は 0.5A です。

## 4-4-2. :MEASure:ETIMe

→ Query

説明	負荷入力アオンの経過時間のアクエリ
クエリ構文	:MEASure:ETIMe?
応答	負荷入力アオンの経過時間を”<NR2>”で返します。(単位 [秒])
クエリ 例	:MEAS:ETIM? >10.0 負荷入力アオンの経過時間は 10 秒です。

## 4-4-3. :MEASure:POWEr

→ Query

説明	電力測定のアクエリ
クエリ構文	:MEASure:POWEr?
応答	電力測定を”<NR2>”で返します。(単位 [W])
クエリ 例	:MEAS:POW? >15.00000 電力測定は 15W です。

#### 4-4-4. :MEASure:VOLTage

→ Query

説明	電圧測定のカエリ
クエリ構文	:MEASure:VOLTage?
応答	電圧測定を"<NR2>"で返します。(単位 [V])
クエリ 例	:MEAS:VOLT? >5.00000 電圧測定は 5V です。

#### 4-5. フェッチコマンド

##### 4-5-1. :FETCh:CURRent

→ Query

説明	電流瞬時値のカエリ
クエリ構文	FETCh:CURRent?
応答	電流を"<NR2>"で返します。(単位 [A])
クエリ 例	:FETC:CURR? >0.50000 電流は 0.5A です。

##### 4-5-2. :FETCh:POWer

→ Query

説明	電力瞬時値のカエリ
クエリ構文	FETCh:POWer?
応答	電力を"<NR2>"で返します。(単位 [W])
クエリ 例	:FETC:POW? >15.00000 電力は 15W です。

##### 4-5-3. :FETCh:VOLTage

→ Query

説明	電圧瞬時値のカエリ
クエリ構文	FETCh:VOLTage?
応答	電圧を"<NR2>"で返します。(単位 [V])
クエリ 例	:FETC:VOLT? >5.00000 電圧は 5V です。

## 4-6. 設定サブシステム・コマンド

Set →  
→ Query

### 4-6-1. [:CONFigure]:OCP

説明	OCP の設定またはクエリ。OCP は任意の値に設定でき、動作は過電流制限モードまたは負荷オフのトリップモードを設定することができます。	
構文	[:CONFigure]:OCP {<NRf>[A]   MINimum   MAXimum   LIMit   LOFF}	
クエリ構文	[:CONFigure]:OCP?	
パラメーター	<NRf>[A]	過電流制限値
	MINimum	最小過電流制限値
	MAXimum	最大過電流制限値
	LIMit	過電流制限モード設定
	LOFF	負荷オフのトリップモード設定
応答	動作モードと過電流制限値を”{Load off   LIMIT},<NR2>”で返します。	
例 1	:OCP LIM 過電流制限モードに設定します。	
例 2	:OCP 77.000 過電流制限値を 77A に設定します。	
クエリ 例	:OCP? >LIMIT, 77.000 OCP の設定は、OCP 値が 77A の電流制限モードです。	

Set →  
→ Query

### 4-6-2. [:CONFigure]:OPP

説明	OPP の設定またはクエリ。OPP は任意の値に設定でき、動作は過電力制限モードまたは負荷オフのトリップモードを設定することができます。	
構文	[:CONFigure]:OPP {<NRf>[W]   MINimum   MAXimum   LIMit   LOFF}	
クエリ構文	[:CONFigure]:OPP?	
パラメーター	<NRf>[W]	過電力制限値
	MINimum	最小過電力制限値
	MAXimum	最大過電力制限値
	LIMit	過電力制限モード設定
	LOFF	負荷オフのトリップモード設定
応答	動作モードと過電力制限値を”{Load off   LIMIT},<NR2>”で返します。	

例 1	:OPP LIM 過電力制限モードに設定します。
例 2	:OPP 10.000 過電力制限値を 10W に設定します。
クエリ 例	:OPP? >LIMIT, 10.000 OPP の設定は、OPP 値が 10W の電力制限モードです。

Set →

#### 4-6-3. [:CONFigure]:UVP

→ Query

説明	UVP の設定またはクエリ。このコマンドで UVP をクリアすることができます。	
構文	[:CONFigure]:UVP {<NRf>[V]   MINimum   MAXimum }	
クエリ構文	[:CONFigure]:UVP?	
パラメーター	<NRf>[V]	低電圧制限値
	MINimum	最小低電圧制限値
	MAXimum	最大低電圧制限値
応答	UVP の設定値を”<NR2>”で返します。	
例	:UVP 10.00 UVP を 10V に設定します。	
クエリ 例	:UVP? >10.0000 UVP の設定が 10V です。	

Set →

→ Query

#### 4-6-4. [:CONFigure]:UVP:TIME

説明	UVP のブザー時間の設定またはクエリ。	
構文	[:CONFigure]:UVP:TIME<NR1>	
クエリ構文	[:CONFigure]:UVP:TIME?	
パラメーター	<NR1>[sec]	ブザー時間
応答	UVP ブザーの設定値を”<NR1>”で返します。	
例	:UVP:TIME 5 ブザーを 5 秒に設定します。	
クエリ 例	:UVP:TIME? > 5 ブザー時間は 5 秒です。	

Set →

→ Query

## 4-6-5. [:CONFigure]:OVP

説明	OVPの設定またはクエリ。このコマンドでOVPをクリアすることができます。	
構文	[:CONFigure]:OVP {<NRf>[V]   MINimum   MAXimum }	
クエリ構文	[:CONFigure]:OVP?	
パラメーター	<NRf>[V]	過電圧制限値
	MINimum	最小過電圧制限値
	MAXimum	最大過電圧制限値 (OFF)
応答	OVP の設定値を”{<NR2>   OFF}”で返します。 ”OFF”は機能オフです。	
例	:OVP 10.00 OVP を 10V に設定します。	
クエリ 例	:OVP? >10.0000 OVP 設定は 10.0000V です。	

Set →

→ Query

## 4-6-6. [:CONFigure]:SStart

説明	ソフトスタート時間の設定またはクエリ。	
構文	[:CONFigure]:SStart {<NRf>[S]   OFF   MINimum   MAXimum }	
クエリ構文	[:CONFigure]:SStart?	
パラメーター	<NRf>[S]	ソフトスタート時間 (単位は秒)
	MINimum	最小時間=0 秒
	MAXimum	最大時間
	OFF	OFF = 0 秒
応答	ソフトスタート時間の設定値を”{<NR2>   OFF}”で返します。 ”OFF”は機能オフです。	
例	:SST OFF ソフトスタート機能をオフにします。	
クエリ 例	:SST? >OFF ソフトスタート機能がオフになっています。	

Set →

→ Query

## 4-6-7. [:CONFigure]:VON

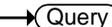
説明	Von 電圧の設定またはクエリ。	
構文	[:CONFigure]:VON {<NRf>[V]   MINimum   MAXimum   LON   LOFF }	

クエリ構文	[:CONFigure]:VON?	
パラメーター	<NRf>[V] MINimum MAXimum LON LOFF	Von 電圧レベル 最小 Von 電圧レベル 最大 von 電圧レベル ラッチオン ラッチオフ
応答	動作モードと Von 電圧値を” Latch:{OFF   ON},<NR2>”で返します。	
例	:VON 10.0V Von を 10.0V に設定します。	
クエリ 例	:VON? >Latch:OFF, 0.000 Von ラッチはオフ、Von 電圧は 0V です。	

 →  
→ 

#### 4-6-8. [:CONFigure]:VDElay

説明	Von デイレイを秒単位で設定またはクエリ。	
構文	[:CONFigure]:VDElay {<NRf>[S]   OFF   MINimum   MAXimum}	
クエリ構文	[:CONFigure]:VDElay?	
パラメーター	<NRf>[S] OFF MINimum MAXimum	遅延時間 (単位は秒) 遅延時間をオフ 最小遅延時間 最大遅延時間
応答	遅延時間の設定値を ”{<NR2>   OFF}”で返します。 ”OFF”は機能オフです。	
例 1	:VDEL 2.5mS 遅延時間を 2.5ms に設定します。	
例 2	:VDEL 0.0025S 遅延時間を 2.5ms に設定します。	
クエリ 例	:VDEL? >0.0025 遅延時間は 2.5ms です。	

 →  
→ 

#### 4-6-9. :CONFigure:RESPonse

説明	CC,CR,CP モードの応答速度の設定またはクエリ。	
構文	:CONFigure:RESPonse {<NR2>[S]   MINimum   MAXimum }	
クエリ構文	:CONFigure:RESPonse?	

パラメーター	<NR2>[S] MINimum MAXimum	0.1, 0.2, 0.5, 1.0 の応答設定 最小応答速度 最大応答速度
応答	応答速度の設定値を”<NR2>”で返します。	
例	:CONF:RESP MAX CC,CR,CP モードの応答速度を最大へ設定します。	
クエリ 例	:CONF:RESP? >1.0 CC、CP,R モードの応答速度は 1.0 です。	

Set →

→ Query

#### 4-6-10. [:CONFigure]:CNTime

説明	カウント時間タイマー機能の設定またはクエリ。	
構文	[:CONFigure]:CNTime {OFF   ON}	
クエリ構文	[:CONFigure]:CNTime?	
パラメーター/応答	OFF ON	カウント時間タイマーをオフ カウント時間タイマーをオン
例	:CNT ON カウント時間タイマーをオンにします。	
クエリ 例	:CNT? >ON カウント時間タイマーはオンになっています。	

Set →

→ Query

#### 4-6-11. [:CONFigure]:COTime

説明	負荷カットオフ時間の設定またはクエリ。 0 秒のカットオフ時間は、OFF と同等です。	
構文	[:CONFigure]:COTime {<NRf>[S]   OFF   MINimum   MAXimum}	
クエリ構文	[:CONFigure]:COTime?	
パラメーター	<NRf>[S] OFF MINimum MAXimum	秒単位のカットオフ時間(1~3599999) カットオフ時間をオフ設定 カットオフ時間を最大に設定 カットオフ時間を最小に設定
応答	カットオフ時間を”{<NR1>   OFF}”で返します。 ”OFF”は機能オフです。	
例	:COT MAX カットオフ時間を最大に設定します。	
クエリ 例	:COT? >500 カットオフ時間は 500 秒に設定されています。	

Set →

→ Query

## 4-6-12. [:CONFigure]:CRUNit

説明	CR モードの表示単位の設定またはクエリ。	
構文	[:CONFigure]:CRUNit {OHM   MHO}	
クエリ構文	[:CONFigure]:CRUNit?	
パラメーター/応答	OHM	表示単位を Ω に設定
	MHO	表示単位を mS(ミリジーメンズ)に設定
例	:CRU OHM CR モードの単位を Ω に設定します。	
クエリ 例	:CRU? >OHM CR モードの単位は Ω です。	

Set →

→ Query

## 4-6-13. :CONFigure:DYNamic

説明	ダイナミックモードの設定条件の設定とクエリ。設定条件は値またはパーセント、タイマーまたはデューティ・サイクルを選択できます。詳細は取扱説明書を参照してください。	
構文	:CONFigure:DYNamic {VALue   PERCent   TIME   FDUTy}	
クエリ構文	:CONFigure:DYNamic?	
パラメーター	VALue	値に単位を設定
	PERCent	%に単位を設定
	TIME	タイマーに設定
	FDUTy	デューティ・サイクルに設定
応答	単位とタイミングのモードを”{Value   Percent},{T1/T2   Fre./Duty}”で返します。	
例	:CONF:DYN VAL ダイナミックモードの単位を値に設定します。	
クエリ 例	:CONF:DYN? >Value,T1/T2 ダイナミックモードは値設定とタイマー設定になります。	

Set →

→ Query

## 4-6-14. :CONFigure:MEMory

説明	ローカルモードでファイルをリコールするときの確認の有無を設定します。	
構文	:CONFigure:MEMory {SAFety   DIReCt}	
クエリ構文	:CONFigure:MEMory?	
パラメーター	SAFety	確認あり

	DIRect	確認なし
応答	リコールの確認の有無を”{ Safety   Direct }”で返します。	
例	:CONF:MEM SAF 確認を有効にします。	
クエリ 例	:CONF:MEM? >Safety 確認が有効になっています。	

Set →

→ Query

#### 4-6-15. :CONFigure:SHORt

説明	ショートキー動作の設定またはクエリ。	
構文	:CONFigure:SHORt {TOGGle   HOLD}	
クエリ構文	:CONFigure:SHORt?	
パラメーター	HOLD	ホールド設定
	TOGGle	トグル設定
応答	ショートキー動作を”{Toggle   Hold}”で返します。	
例	:CONF:SHOR TOGG ショートキーをトグルに設定します。	
クエリ 例	:CONF:SHOR? >Toggle ショートキーがトグルに設定されています。	

Set →

→ Query

#### 4-6-16. :CONFigure:SHORt:SAFety

説明	ショートキー安全動作の設定またはクエリ。	
構文	:CONFigure:SHORt:SAFety {<bool> OFF ON}	
クエリ構文	:CONFigure:SHORt:SAFety?	
パラメーター	OFF	安全機能をオフします。
	ON	安全機能をオンします。Load がオンの時のみショートが ON できます。
応答	ショートキー安全動作の状態を返します。	
例	:CONF:SHOR:SAF OFF ショート安全機能をオフします。	
クエリ 例	:CONF:SHOR:SAF? >OFF ショート安全機能はオフです。	

Set →

→ Query

#### 4-6-17. :CONFigure:SHORt:FUNCTion

説明	ショート機能を有効・無効にします
----	------------------

構文	CONFigure:SHORt:FUNcTION {<bool> OFF ON}	
クエリ構文	:CONFigure:SHORt:FUNcTION?	
パラメーター	OFF	ショート機能をオフします。
	ON	ショート機能をオンします。
応答	ショート動作の状態を返します。	
例	:CONF:SHOR:FUNC OFF ショート機能をオフします。	
クエリ 例	:CONF:SHOR:FUNC? >OFF ショート機能はオフです。	

Set →

→ Query

#### 4-6-18. [:CONFigure]:GNG:SPECTest

説明	Go-NoGo テストの設定またはクエリ。	
構文	[:CONFigure]:GNG:SPECTest {OFF   ON}	
クエリ構文	[:CONFigure]:GNG:SPECTest?	
パラメーター/応答	OFF	SPEC テストをオフに設定
	ON	SPEC テストをオンに設定
例	:GNG:SPECT ON Go-NoGo テストをオンにします。	
クエリ 例	:GNG:SPECT? >OFF Go-NoGo テストはオフです。	

Set →

→ Query

#### 4-6-19. [:CONFigure]:GNG:DTIME

説明	Go-NoGo 遅延時間の設定またはクエリ。	
構文	[:CONFigure]:GNG:DTIME{<NRf>[S]   MINimum   MAXimum }	
クエリ構文	[:CONFigure]:GNG:DTIME?	
パラメーター	<NRf>[S]	Go-NoGo 遅延時間(0.0~1.0)を秒単位で設定します。0.1秒の分解能
	MINimum	最小遅延時間
	MAXimum	最大遅延時間
応答	秒単位で遅延時間を”<NR2>”で返します。	
例	:GNG:DTIM 0.5 遅延時間を 0.5 秒に設定します。	
クエリ 例	:GNG:DTIM? >0.5 遅延時間は 0.5 秒です。	

Set →

→ Query

## 4-6-20. [:CONFigure]:GNG:MODE

説明	Go-NoGo 入力モードの設定またはクエリ。 入力モードは Go-NoGo 制限値として、または中心基準値からのパーセント値として設定します。	
構文	[:CONFigure]:GNG:MODE {PERCent   VALue}	
クエリ構文	[:CONFigure]:GNG:MODE?	
パラメーター	PERCent	%のモードを設定
	VALue	値のモードを設定
応答	Go-NoGo 入力モードの設定値を”{Percent   Value}”で返します。	
例	:GNG:MODE PERC %への入力モードを設定します。	
クエリ 例	:GNG:MODE? >Percent 入力モードは%です。	
関連コマンド	[:CONFigure]:GNG:H [:CONFigure]:GNG:L [:CONFigure]:GNG:C	

## 4-6-21. [:CONFigure]:GNG[:PASS]

→ Query

説明	Go-NoGo 試験結果のクエリ。このコマンドは、すべてのテストモード(CC、CV、CR、CP)に使用することができます。	
クエリ構文	[:CONFigure]:GNG[:PASS]?	
応答	NG	No Good (失敗)
	GO	Good (成功)
クエリ 例	:GNG? >GO Go-NoGo テストの結果を返します。	

Set →

→ Query

## 4-6-22. [:CONFigure]:GNG:H

説明	電圧/電流の上限値の設定またはクエリ。入力モードが値に設定されている場合、電圧/電流上限値の単位は V/A です。入力モードがパーセントに設定されている場合、電圧/電流上限値の単位は%です。	
構文	[:CONFigure]:GNG:H <NR2>	
クエリ構文	[:CONFigure]:GNG:H?	
パラメーター	<NRf>	電圧/電流の上限値を値やパーセントで設定

応答	電圧/電流上限値の値やパーセントを”<NR2>”で返します。
例	:GNG:H 100.0 Go-NoGo テストの上限値を 100 に設定します。
クエリ 例	:GNG:H? >100.0 Go-NoGo テストの上限値を返します。
関連コマンド	[:CONFigure]:GNG:MODE [:CONFigure]:GNG:L

Set →

#### 4-6-23. [:CONFigure]:GNG:L

→ Query

説明	電圧/電流の下限値の設定またはクエリ。入力モードが値に設定されている場合、電圧/電流下限値の単位は V/A です。入力モードがパーセントに設定されている場合、電圧/電流下限値の単位は%です。
構文	[:CONFigure]:GNG:L <NRf>
クエリ構文	[:CONFigure]:GNG:L?
パラメーター	<NRf> 電圧/電流の下限値を値やパーセントで設定
応答	電圧/電流の下限値を”<NR2>”で返します。
例	:GNG:L 10.0 Go-NoGo テストの下限値を 10 に設定します。
クエリ 例	:GNG:L? >10.0 Go-NoGo テストの下限値を返します。
関連コマンド	[:CONFigure]:GNG:MODE [:CONFigure]:GNG:H

Set →

#### 4-6-24. [:CONFigure]:GNG:C

→ Query

説明	電圧/電流中心制限値の設定またはクエリ。中心制限値は入力モードをパーセントに設定されている中央の基準値として使用されます。
構文	[:CONFigure]:GNG:C <NR2>
クエリ構文	[:CONFigure]:GNG:C?
パラメーター	<NR2> 電圧または電流の中心を値で設定
応答	電圧/電流の中心値を”<NR2>”で返します。
例	:GNG:C 10.0 電圧/電流中心制限値を 10V または A に設定します。
クエリ 例	:GNG:C? >10.0 電圧/電流中心制限値を 10V または A に返します。

関連コマンド     [:CONFigure]:GNG:MODE

## 4-7. パラレルコマンド

Set →  
→ Query

### 4-7-1. [:CONFigure]:PARAllel

説明	並列運転のためのユニットの設定またはクエリ。	
構文	[:CONFigure]:PARAllel { MASTER   SLAVe   OFF   P2   P3   P4   P5   B1   B2   B3   B4}	
クエリ構文	[:CONFigure]:PARAllel?	
パラメーター	P2 / P3 / P4 / P5 B1 / B2 / B3 / B4 OFF MASTER SLAVe	スレーブ接続台数 ブースタの台数設定 単体動作に設定 マスターに設定 スレーブに設定
応答	モード(マスター/スレーブ)と接続数を返します。 マスターモードの応答は “Mode:Master,{Number:OFF   Parallel Number:{2 3 4 5}   Booster Number: {1 2 3 4}” です。 スレーブモードの応答は”Mode:Slave”です。	
例 1	:PAR MAST マスターに設定します。	
例 2	:PAR B2 ブースターユニットとの2台構成の設定です。	
クエリ 例	:PAR? >Mode:Master, Number:OFF マスターに設定され、接続されたスレーブは存在しません。	

## 4-8. ステップコマンド

### 4-8-1. [:CONFigure]:STEP:CC

→ Query

説明	各 CC モードレンジのステップ分解能のクエリ。	
クエリ構文	[:CONFigure]:STEP:CC?	
応答	各 CC モードレンジのステップ分解能を”CCH:<NR2>, CCM:<NR2>, CCL:<NR2>”で返します。	
クエリ 例	:STEP:CC? >CCH:0.002, CCM:0.0002, CCL:0.00002 各レンジの CC モード・ステップ分解能を返します。	

Set →

→ Query

#### 4-8-2. [:CONFigure]:STEP:CCH

説明	CC の H レンジのステップ分解能の設定またはクエリ。 注: ステップ分解能の設定は自動的に基本分解能の最も近い倍数に丸められます。	
構文	[:CONFigure]:STEP:CCH { <NRf>[A]   MINimum   MAXimum }	
クエリ構文	[:CONFigure]:STEP:CCH?	
パラメーター	<NRf>[A]	ステップ分解能
	MINimum	最小ステップ分解能
	MAXimum	最大ステップ分解能
応答	レンジとステップ分解能を”CCH:<NR2>”で返します。	
例	:STEP:CCH 0.002A ステップ分解能を 0.002A に設定します。	
クエリ 例	:STEP:CCH? >CCH:0.002 CCH のステップ分解能 (0.002A) を返します。	

Set →

→ Query

#### 4-8-3. [:CONFigure]:STEP:CCM

説明	CC の M レンジのステップ分解能の設定またはクエリ。 注: ステップ分解能の設定は自動的に基本分解能の最も近い倍数に丸められます。	
構文	[:CONFigure]:STEP:CCM { <NRf>[A]   MINimum   MAXimum }	
クエリ構文	[:CONFigure]:STEP:CCM?	
パラメーター	<NRf>[A]	ステップ分解能
	MINimum	最小ステップ分解能
	MAXimum	最大ステップ分解能
応答	レンジとステップ分解能を”CCM:<NR2>”で返します。	
例	:STEP:CCM 0.0002A ステップ分解能を 0.0002A に設定します。	
クエリ 例	:STEP:CCM? >CCM:0.0002 CCM のステップ分解能 (0.0002A) を返します。	

Set →

→ Query

## 4-8-4. [:CONFigure]:STEP:CCL

説明	CC の L レンジのステップ分解能の設定またはクエリ。 注: ステップ分解能の設定は自動的に基本分解能の最も近い倍数に丸められます。						
構文	[:CONFigure]:STEP:CCL {<NRf>[A]   MINimum   MAXimum }						
クエリ構文	[:CONFigure]:STEP:CCL?						
パラメーター	<table border="0"> <tr> <td>&lt;NRf&gt;[A]</td> <td>ステップ分解能</td> </tr> <tr> <td>MINimum</td> <td>最小ステップ分解能</td> </tr> <tr> <td>MAXimum</td> <td>最大ステップ分解能</td> </tr> </table>	<NRf>[A]	ステップ分解能	MINimum	最小ステップ分解能	MAXimum	最大ステップ分解能
<NRf>[A]	ステップ分解能						
MINimum	最小ステップ分解能						
MAXimum	最大ステップ分解能						
応答	レンジとステップ分解能を”CCL:<NR2>”で返します。						
例	:STEP:CCL 0.00002A ステップ分解能を 0.00002 A に設定します。						
クエリ 例	:STEP:CCL? >CCL:0.00002 CCL のステップ分解能(0.00002A)を返します。						

## 4-8-5. [:CONFigure]:STEP:CR

→ Query

説明	各 CR モードレンジのステップ分解能のクエリ。
クエリ構文	[:CONFigure]:STEP:CR?
応答	各 CR モードレンジのステップ分解能を”CRH:<NR2>, CRM:<NR2>, CRL:<NR2>”で返します。
クエリ 例	:STEP:CR? >CRH:0.8, CRM:0.08, CRL:0.008 各レンジの CR モード・ステップ分解能を返します。

Set →

→ Query

## 4-8-6. [:CONFigure]:STEP:CRH

説明	CR の H レンジのステップ分解能の設定またはクエリ。 注: ステップ分解能の設定は自動的に基本分解能の最も近い倍数に丸められます。						
構文	[:CONFigure]:STEP:CRH {<NRf>[mho]   MINimum   MAXimum }						
クエリ構文	[:CONFigure]:STEP:CRH?						
パラメーター	<table border="0"> <tr> <td>&lt;NRf&gt;[mho]</td> <td>ステップ分解能 (単位は mMHO)</td> </tr> <tr> <td>MINimum</td> <td>最小ステップ分解能</td> </tr> <tr> <td>MAXimum</td> <td>最大ステップ分解能</td> </tr> </table>	<NRf>[mho]	ステップ分解能 (単位は mMHO)	MINimum	最小ステップ分解能	MAXimum	最大ステップ分解能
<NRf>[mho]	ステップ分解能 (単位は mMHO)						
MINimum	最小ステップ分解能						
MAXimum	最大ステップ分解能						
応答	レンジとステップ分解能を”CRH:<NR2>”で返します。						

例 :STEP:CRH 0.8  
ステップ分解能を 0.8mS に設定します。

クエリ 例 :STEP:CRH?  
>CRH:0.8  
CRH のステップ分解能 (0.8mS) を返します。

Set →

→ Query

#### 4-8-7. [:CONFigure]:STEP:CRM

説明 CR の M レンジのステップ分解能の設定またはクエリ。  
注: ステップ分解能の設定は自動的に基本分解能の最も近い  
倍数に丸められます。

構文 [:CONFigure]:STEP:CRM {<NRf>[mho] | MINimum |  
MAXimum }

クエリ構文 [:CONFigure]:STEP:CRM?

パラメーター	<NRf>[mho]	ステップ分解能 (単位は mMHO)
	MINimum	最小ステップ分解能
	MAXimum	最大ステップ分解能

応答 レンジとステップ分解能を"CRM:<NR2>"で返します。

例 :STEP:CRM 0.08  
0.08mS にステップ分解能を設定します。

クエリ 例 :STEP:CRM?  
>CRM:0.08  
CRM のステップ分解能 (0.08mS) を返します。

Set →

→ Query

#### 4-8-8. [:CONFigure]:STEP:CRL

説明 CR の L レンジのステップ分解能の設定またはクエリ。  
注: ステップ分解能の設定は自動的に基本分解能の最も近い  
倍数に丸められます。

構文 [:CONFigure]:STEP:CRL {<NRf>[mho] | MINimum |  
MAXimum }

クエリ構文 [:CONFigure]:STEP:CRL?

パラメーター	<NRf>[mho]	ステップ分解能 (単位は mMHO)
	MINimum	最小ステップ分解能
	MAXimum	最大ステップ分解能

応答 レンジとステップ分解能を"CRL:<NR2>"で返します。

例 :STEP:CRL 0.008  
ステップ分解能を 0.008mS に設定します。

クエリ 例 :STEP:CRL?  
>CRL:0.008  
CRL のステップ分解能 (0.008mS) を返します。

#### 4-8-9. [:CONFigure]:STEP:CV

→ Query

説明	各 CV モードレンジのステップ分解能を返します。
クエリ構文	[:CONFigure]:STEP:CV?
応答	各 CV モードレンジのステップ分解能を”CVH:<NR2>, CVL:<NR2>”で返します。
クエリ 例	:STEP:CV? >CVH:0.01, CVL:0.001 各レンジの CV モード・ステップ分解能を返します。

Set →

#### 4-8-10. [:CONFigure]:STEP:CVH

→ Query

説明	CV の H レンジのステップ分解能の設定またはクエリ。 注: ステップ分解能の設定は自動的に基本分解能の最も近い倍数に丸められます。
構文	[:CONFigure]:STEP:CVH {<NRf>[V]   MINimum   MAXimum }
クエリ構文	[:CONFigure]:STEP:CVH?
パラメーター	<NRf>[V]      ステップ分解能 MINimum      最小ステップ分解能 MAXimum      最大ステップ分解能
応答	レンジとステップ分解能を”CVH:<NR2>”で返します。
例	:STEP:CVH 0.01V ステップ分解能を 0.01V に設定します。
クエリ 例	:STEP:CVH? >CVH:0.01 CVH のステップ分解能(0.01V)を返します。

Set →

#### 4-8-11. [:CONFigure]:STEP:CVL

→ Query

説明	CV の L レンジのステップ分解能の設定またはクエリ。 注: ステップ分解能の設定は自動的に基本分解能の最も近い倍数に丸められます。
構文	[:CONFigure]:STEP:CVL {<NRf>[V]   MINimum   MAXimum }
クエリ構文	[:CONFigure]:STEP:CVL?
パラメーター	<NRf>[V]      ステップ分解能 MINimum      最小ステップ分解能 MAXimum      最大ステップ分解能
応答	レンジとステップ分解能を”CVL:<NR2>”で返します。

例 :STEP:CVL 0.001V  
ステップ分解能を 0.001V に設定します。

クエリ 例 :STEP:CVL?  
>CVL:0.001  
CVL のステップ分解能 (0.001V) を返します。

4-8-12. [:CONFigure]:STEP:CP → [Query](#)

説明 各 CP モードレンジのステップ分解能を返します。

クエリ構文 [:CONFigure]:STEP:CP?

応答 各 CP モードレンジのステップ分解能を”CPH:<NR2>,  
CPM:<NR2>, CPL:<NR2>”で返します。

クエリ 例 :STEP:CP?  
>CPH:0.01, CPM:0.001, CPL:0.0001  
各レンジの CP モード・ステップ分解能を返します。

[Set](#) →

4-8-13. [:CONFigure]:STEP:CPH → [Query](#)

説明 CP の H レンジのステップ分解能の設定またはクエリ。  
注: ステップ分解能の設定は自動的に基本分解能の最も近い  
倍数に丸められます。

構文 [:CONFigure]:STEP:CPH {<NRf>[W] | MINimum |  
MAXimum }

クエリ構文 [:CONFigure]:STEP:CPH?

パラメーター	<NRf>[W]	ステップ分解能
	MINimum	最小ステップ分解能
	MAXimum	最大ステップ分解能

応答 レンジとステップ分解能を”CPH:<NR2>”で返します。

例 :STEP:CPH 0.01  
ステップ分解能を 0.01W に設定します。

クエリ 例 :STEP:CPH?  
>CPH:0.01  
CPH のステップ分解能 (0.01W) を返します。

[Set](#) →

4-8-14. [:CONFigure]:STEP:CPM → [Query](#)

説明 CP の M レンジのステップ分解能の設定またはクエリ。  
注: ステップ分解能の設定は自動的に基本分解能の最も近い  
倍数に丸められます。

構文 [:CONFigure]:STEP:CPM {<NRf>[W] | MINimum |  
MAXimum }

クエリ構文	[:CONFigure]:STEP:CPM?	
パラメーター	<NRf>[W] MINimum MAXimum	ステップ分解能 最小ステップ分解能 最大ステップ分解能
応答	レンジとステップ分解能を”CPM:<NR2>”で返します。	
例	:STEP:CPM 0.001 ステップ分解能を 0.001W に設定します。	
クエリ 例	:STEP:CPM? >CPM:0.001 CPM のステップ分解能 (0.001W) を返します。	

Set →

→ Query

#### 4-8-15. [:CONFigure]:STEP:CPL

説明	CP の L レンジのステップ分解能の設定またはクエリ。 注: ステップ分解能の設定は自動的に基本分解能の最も近い倍数に丸められます。	
構文	[:CONFigure]:STEP:CPL {<NRf>[W]   MINimum   MAXimum }	
クエリ構文	[:CONFigure]:STEP:CPL?	
パラメーター	<NRf>[W] MINimum MAXimum	ステップ分解能 最小ステップ分解能 最大ステップ分解能
応答	レンジとステップ分解能を”CPL:<NR2>”で返します。	
例	:STEP:CPL 0.0001 ステップ分解能を 0.0001W に設定します。	
クエリ 例	:STEP:CPL? >CPL:0.0001 CPL のステップ分解能 (0.0001W) を返します。	

### 4-9. 外部制御コマンド

Set →

→ Query

#### 4-9-1. [:CONFigure]:EXTErnal[:CONTRol]

説明	外部制御のモードの設定とクエリ。	
構文	[:CONFigure]:EXTErnal[:CONTRol] { OFF   VOLTage   RESistance   RINV }	
クエリ構文	[:CONFigure]:EXTErnal[:CONTRol]?	
パラメーター	OFF VOLTage RESistance RINV	外部制御の無効設定 外部電圧コントロール設定 外部抵抗コントロール設定 外部抵抗 (反転) コントロール設定

応答	外部制御のモードを”Control:{OFF   Volt   Res   Rinverse}”で返します。
例	:EXT OFF 外部制御を無効設定にします。
クエリ 例	:EXT? >Control:OFF, 外部制御は無効になっています。

Set →  
 → Query

#### 4-9-2. [:CONFigure]:EXTernal:LOADonin

説明	外部スイッチの設定とクエリ。 外部スイッチ(LoadOn IN)はクローズ(Low)またはオープン(High)のときに負荷がオンになっているかどうかの設定です。						
構文	[:CONFigure]:EXTernal:LOADonin {OFF   HIGH   LOW}						
クエリ構文	[:CONFigure]:EXTernal:LOADonin?						
パラメーター	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">OFF</td> <td>LoadOn IN =オフ</td> </tr> <tr> <td>HIGH</td> <td>LoadOn IN =オープン</td> </tr> <tr> <td>LOW</td> <td>LoadOn IN =クローズ</td> </tr> </table>	OFF	LoadOn IN =オフ	HIGH	LoadOn IN =オープン	LOW	LoadOn IN =クローズ
OFF	LoadOn IN =オフ						
HIGH	LoadOn IN =オープン						
LOW	LoadOn IN =クローズ						
応答	外部スイッチの設定を”LoadOn In:{OFF   High   Low}”で返します。						
例	:EXT:LOAD OFF ”LoadOn IN”の設定をオフにします。						
クエリ 例	:EXT:LOAD? >OFF ”LoadOn IN”の設定はオフになっています。						

### 4-10. モード・サブシステム・コマンド

Set →  
 → Query

#### 4-10-1. :MODE

説明	動作モードの設定またはクエリ。														
構文	:MODE {CC   CR   CV   CP   CCCV   CRCV   CPCV}														
クエリ構文	:MODE?														
パラメーター/応答	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">CC</td> <td>CC モード</td> </tr> <tr> <td>CR</td> <td>CR モード</td> </tr> <tr> <td>CV</td> <td>CV モード</td> </tr> <tr> <td>CP</td> <td>CP モード</td> </tr> <tr> <td>CCCV</td> <td>CC + CV モード</td> </tr> <tr> <td>CRCV</td> <td>CR + CV モード</td> </tr> <tr> <td>CPCV</td> <td>CP + CV モード</td> </tr> </table>	CC	CC モード	CR	CR モード	CV	CV モード	CP	CP モード	CCCV	CC + CV モード	CRCV	CR + CV モード	CPCV	CP + CV モード
CC	CC モード														
CR	CR モード														
CV	CV モード														
CP	CP モード														
CCCV	CC + CV モード														
CRCV	CR + CV モード														
CPCV	CP + CV モード														

例 :MODE CC  
CC モードへ設定します。

クエリ 例 :MODE?  
>CC  
動作モード(CC モード)を返します。

Set →

→ Query

#### 4-10-2. [:MODE]:CRANge

説明 全オペレーティング・モードの電流レンジの設定またはクエリ。

構文 [:MODE]:CRANge { HIGH | MIDDLE | LOW }

クエリ構文 [:MODE]:CRANge?

パラメーター HIGH H レンジ  
MIDDLE M レンジ  
LOW L レンジ

応答 電流レンジの設定を"{High | Mid | Low}"で返します。

例 :CRAN LOW  
電流レンジを LOW に設定します。

クエリ 例 :CRAN?  
>Low  
電流レンジは Low に設定されています。

Set →

→ Query

#### 4-10-3. [:MODE]:VRANge

説明 全オペレーティング・モードの電圧レンジの設定またはクエリ。

構文 [:MODE]:VRANge { HIGH | LOW }

クエリ構文 [:MODE]:VRANge?

パラメーター HIGH H レンジ  
LOW L レンジ

応答 電圧レンジの設定を"{High | Low}"で返します。

例 :VRAN LOW  
電圧レンジを Low に設定します。

クエリ 例 :VRAN?  
>Low  
電圧レンジが Low に設定されています。

Set →

→ Query

#### 4-10-4. [:MODE]:RESPonse

説明 CV モードの応答速度の設定とクエリ。デフォルトは高速応答です。

構文 [:MODE]:RESPonse { FAST | SLOW }

クエリ構文 [:MODE]:RESPonse?

パラメーター/応答	FAST SLOW	高速応答の設定 低速応答の設定
例	:RESP FAST CV 応答を高速に設定します。	
クエリ 例	:RESP? >FAST CV モードは高速応答に設定されています。	

Set →  
→ Query

#### 4-10-5. [:MODE]:DYNamic

説明	スイッチングモードの設定またはクエリ。	
構文	[:MODE]:DYNamic {DYNamic   STATic}	
クエリ構文	[:MODE]:DYNamic?	
パラメーター	DYNamic STATic	ダイナミックモードの設定 スタティックモードの設定
応答	スイッチングモードの設定を"{Dynamic   Static}"で返します。	
例	:DYN DYN スイッチングモードをダイナミックモードに設定します。	
クエリ 例	:DYN? >Dynamic スイッチングモードは、ダイナミックモードに設定されています。	

### 4-11. 電流サブシステム・コマンド

Set →  
→ Query

#### 4-11-1. :CURRent[:VA]

説明	CC モード "A Value" の電流の設定またはクエリ。 このコマンドは、スタティックモードに適用されます。 注: 各レンジ (H/M/L) で異なる値を設定することができます。 注: [:VA] ノードは、スタティックモードとときにのみ省略することができます。	
構文	:CURRent[:VA] {<NRf>[A]   MINimum   MAXimum }	
クエリ構文	:CURRent[:VA]?	
パラメーター	<NRf>[A] MINimum MAXimum	"A Value" の電流値 最小電流レベル 最大電流レベル
応答	"A Value" の電流値を "<NR2>" で返します。	
例	:CURR MIN 最小の電流値を設定します。	

クエリ 例 :CURR?  
>1.0  
"A Value"の電流設定値は 1A に設定されています。

#### 4-11-2. :CURRent[:VA]:TRIGgered

Set →

説明	トリガがアクティブ化された時の電流値を設定します。	
構文	:CURRent[:VA]:TRIGgered {<NR2>[A]   MINimum   MAXimum }	
パラメーター	<NRf>[A] MINimum MAXimum	"A Value"の電流値 最小電流レベル 最大電流レベル
例	:CURR:TRIG MIN トリガーがアクティブ化された時に最小電流値を設定します。	
関連コマンド	*TRG, :INITiate:CONTinuous, :INITiate[:IMMEDIATE]	

#### 4-11-3. :CURRent:VB

Set →

→ Query

説明	CC モード" B Value"の電流の設定またはクエリ。 このコマンドは、スタティックモードに適用されます。 注:各レンジ(H/M/L)で異なる値を設定することができます。	
構文	:CURRent:VB {<NRf>[A]   MINimum   MAXimum }	
クエリ構文	:CURRent:VB?	
パラメーター	<NRf>[A] MINimum MAXimum	"B Value" の電流値 最小電流レベル 最大電流レベル
応答	"B Value"の電流値を"<NR2>"で返します。	
例	:CURR:VB MIN 最小電流値を設定します。	
クエリ 例	:CURR:VB? >1.0 "B Value"の電流設定値は 1A に設定されています。	

Set →

→ Query

#### 4-11-4. :CURRent:SRATe

説明	CC スタティックモードの電流のスルーレートの設定またはクエリ。
----	----------------------------------

構文	:CURRent:SRATe {<NRf>   MINimum   MAXimum }	
クエリ構文	:CURRent:SRATe?	
パラメーター	<NRf>	スルーレートを mA/uS で設定
	MINimum	スルーレートを最小値(遅い)に設定
	MAXimum	スルーレートを最大値(速い)に設定
応答	スルーレートを"<NR2>"で返します。	
例	:CURR:SRAT MIN スルーレートを最小値(遅い)に設定します。	
クエリ 例	:CURR:SRAT? >5.0 スルーレートは 5.0mA/uS に設定されています。	

Set →

→ Query

#### 4-11-5. :CURRent:L1

説明	CC モード"Level1"の電流の設定またはクエリ。このコマンドは、ダイナミックの値モードのみ適用されます。 注:各レンジ(H/M/L)で異なる値を設定することができます。	
構文	:CURRent:L1 {<NRf>[A]   MINimum   MAXimum }	
クエリ構文	:CURRent:L1?	
パラメーター	<NRf>[A]	"Level1" の電流値
	MINimum	最小電流レベル
	MAXimum	最大電流レベル
応答	"Level1"の電流値を"<NR2>"で返します。	
例	:CURR:L1 MIN 最小の電流値を設定します。	
クエリ 例	:CURR:L1? >1.0 "Level1" の電流設定値は 1A に設定されています。	

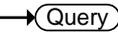
Set →

→ Query

#### 4-11-6. :CURRent:L2

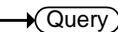
説明	CC モード"Level2"の電流の設定またはクエリ。このコマンドは、ダイナミックの値モードのみ適用されます。 注:各レンジ(H/M/L)で異なる値を設定することができます。	
構文	:CURRent:L2 {<NRf>[A]   MINimum   MAXimum }	
クエリ構文	:CURRent:L2?	
パラメーター	<NRf>[A]	"Level2" の電流値
	MINimum	最小電流レベル
	MAXimum	最大電流レベル
応答	"Level2"の電流値を"<NR2>"で返します。	

例	:CURR:L2 MIN 最小の電流値を設定します。
クエリ 例	:CURR:L2? >1.0 “Level2” の電流設定値は 1A に設定されています。

 →  
 →

#### 4-11-7. :CURRent:SET

説明	CC ダイナミックモードが%に設定されたときの電流の設定またはクエリ。このコマンドは、ダイナミックの%モードのみ適用されます。						
構文	:CURRent:SET{<NRf>[A]   MINimum   MAXimum }						
クエリ構文	:CURRent:SET?						
パラメーター	<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;"><b>&lt;NRf&gt;[A]</b></td> <td>“Level=100%”時の電流値</td> </tr> <tr> <td><b>MINimum</b></td> <td>最小電流値</td> </tr> <tr> <td><b>MAXimum</b></td> <td>最大電流値</td> </tr> </table>	<b>&lt;NRf&gt;[A]</b>	“Level=100%”時の電流値	<b>MINimum</b>	最小電流値	<b>MAXimum</b>	最大電流値
<b>&lt;NRf&gt;[A]</b>	“Level=100%”時の電流値						
<b>MINimum</b>	最小電流値						
<b>MAXimum</b>	最大電流値						
応答	“Level=100%”時の電流値を”<NR2>”で返します。						
例	:CURR:SET MIN “Level=100%”の最小電流値を設定します。						
クエリ 例	:CURR:SET? >1.0 “Level=100%”の電流値は 1A に設定されています。						
関連コマンド	:CURRent:LEVel						

 →  
 →

#### 4-11-8. :CURRent:LEVel

説明	CC ダイナミックモードが%に設定されたときの%レベル(設定電流値の割合)の設定またはクエリ。このコマンドは、ダイナミックの%モードのみ適用されます。						
構文	:CURRent:LEVel {<NRf>   MINimum   MAXimum }						
クエリ構文	:CURRent:LEVel?						
パラメーター	<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;"><b>&lt;NRf&gt;</b></td> <td>“SET”電流値の割合 (単位 [%])</td> </tr> <tr> <td><b>MINimum</b></td> <td>最小“SET”電流値の割合</td> </tr> <tr> <td><b>MAXimum</b></td> <td>最大“SET”電流値の割合</td> </tr> </table>	<b>&lt;NRf&gt;</b>	“SET”電流値の割合 (単位 [%])	<b>MINimum</b>	最小“SET”電流値の割合	<b>MAXimum</b>	最大“SET”電流値の割合
<b>&lt;NRf&gt;</b>	“SET”電流値の割合 (単位 [%])						
<b>MINimum</b>	最小“SET”電流値の割合						
<b>MAXimum</b>	最大“SET”電流値の割合						
応答	電流の“% Level”を”<NR2>”で返します。						
例	:CURR:LEV MIN 最小レベルの電流値の割合(%)を設定します。						
クエリ 例	:CURR:LEV? >50 設定電流値の割合は 50%に設定されています。						

関連コマンド :CURRent:SET

Set →

#### 4-11-9. :CURRent:RISE

→ Query

説明	CC ダイナミックモードの電流スルーレートの立上りの設定またはクエリ	
構文	:CURRent:RISE {<NRf>   MINimum   MAXimum }	
クエリ構文	:CURRent:RISE?	
パラメーター	<NRf> MINimum MAXimum	電流スルーレートの立上り(単位 [mA/uS]) 最小スルーレート 最大スルーレート
応答	電流スルーレートの立上りを”<NR2>”で返します。	
例	:CURR:RISE MIN スルーレートの立上りを最小に設定します。	
クエリ 例	:CURR:RISE? >5000 スルーレートの立上りは 5000mA/uS に設定されています。	

関連コマンド :CURRent:FALL

Set →

#### 4-11-10. :CURRent:FALL

→ Query

説明	CC ダイナミックモードの電流スルーレートの立下りを設定します。	
構文	:CURRent:FALL {<NRf>   MINimum   MAXimum }	
クエリ構文	:CURRent:FALL?	
パラメーター	<NRf> MINimum MAXimum	電流スルーレートの立下り(単位 [mA/uS]) 最小スルーレート 最大スルーレート
応答	電流スルーレートの立下りを”<NR2>”で返します。	
例	:CURR:FALL MIN スルーレートの立下りを最小に設定します。	
クエリ 例	:CURR:FALL? >5000 スルーレートの立下りは 5000mA/uS に設定されています。	

関連コマンド :CURRent:RISE

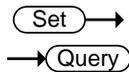
Set →

#### 4-11-11. :CURRent:T1

→ Query

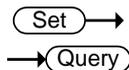
説明	CC ダイナミックモードの T1 タイマーの設定とクエリ。このコマンドは、ダイナミックのタイマーモードのみ適用されます。
----	--

構文	:CURRent:T1 {<NRf>[S]   MINimum   MAXimum }						
クエリ構文	:CURRent:T1?						
パラメーター	<table border="1"> <tr> <td>&lt;NRf&gt;[S]</td> <td>T1 タイマーの時間設定 (単位 [秒])</td> </tr> <tr> <td>MINimum</td> <td>最小時間</td> </tr> <tr> <td>MAXimum</td> <td>最大時間</td> </tr> </table>	<NRf>[S]	T1 タイマーの時間設定 (単位 [秒])	MINimum	最小時間	MAXimum	最大時間
<NRf>[S]	T1 タイマーの時間設定 (単位 [秒])						
MINimum	最小時間						
MAXimum	最大時間						
応答	T1 タイマーの設定を"<NR2>"で返します。						
例	:CURR:T1 0.2 T1 タイマーの設定値を設定します。						
クエリ 例	:CURR:T1? >0.2 T1 タイマーの設定値を返します。						
関連コマンド	:CURRent:T2						



#### 4-11-12. :CURRent:T2

説明	CC ダイナミックモードの T2 タイマーの設定とクエリ。このコマンドは、ダイナミックのタイマーモードのみ適用されます。						
構文	:CURRent:T2 {<NRf>[S]   MINimum   MAXimum }						
クエリ構文	:CURRent:T2?						
パラメーター	<table border="1"> <tr> <td>&lt;NRf&gt;[S]</td> <td>T2 タイマーの時間設定 (単位 [秒])</td> </tr> <tr> <td>MINimum</td> <td>最小時間</td> </tr> <tr> <td>MAXimum</td> <td>最大時間</td> </tr> </table>	<NRf>[S]	T2 タイマーの時間設定 (単位 [秒])	MINimum	最小時間	MAXimum	最大時間
<NRf>[S]	T2 タイマーの時間設定 (単位 [秒])						
MINimum	最小時間						
MAXimum	最大時間						
応答	秒単位で T2 タイマーの設定を"<NR2>"で返します。						
例	:CURR:T2 0.2 T2 タイマーの設定値を設定します。						
クエリ 例	:CURR:T2? >0.2 T2 タイマーの設定値を返します。						
関連コマンド	:CURRent:T1						



#### 4-11-13. :CURRent:FREQuency

説明	CC ダイナミックモードのスイッチング周波数の設定とクエリ。このコマンドは、ダイナミックのデューティ・サイクルモードのみ適用されます。						
構文	:CURRent:FREQuency {<NRf>   MINimum   MAXimum }						
クエリ構文	:CURRent:FREQuency?						
パラメーター	<table border="1"> <tr> <td>&lt;NRf&gt;</td> <td>スイッチング周波数設定 (単位 [Hz])</td> </tr> <tr> <td>MINimum</td> <td>最小周波数</td> </tr> <tr> <td>MAXimum</td> <td>最大周波数</td> </tr> </table>	<NRf>	スイッチング周波数設定 (単位 [Hz])	MINimum	最小周波数	MAXimum	最大周波数
<NRf>	スイッチング周波数設定 (単位 [Hz])						
MINimum	最小周波数						
MAXimum	最大周波数						

応答	スイッチング周波数を"<NR2>"で返します。
例	:CURR:FREQ 60 周波数を 60Hz に設定します。
クエリ 例	:CURR:FREQ? >60 スイッチング周波数は 60Hz に設定されています。
関連コマンド	:CURRent:DUTY

 →  
 → 

#### 4-11-14. :CURRent:DUTY

説明	CC ダイナミックモードのスイッチング周波数のための正のデューティ・サイクルの設定とクエリ。 このコマンドは、ダイナミックのデューティ・サイクルモードのみ適用されます。	
構文	:CURRent:DUTY {<NRf>   MINimum   MAXimum }	
クエリ構文	:CURRent:DUTY?	
パラメーター	<NRf> MINimum MAXimum	デューティ・サイクル設定 (単位 [%]) 最小デューティ・サイクル 最大デューティ・サイクル
応答	正のデューティ・サイクルを"<NR2>"で返します。	
例	:CURR:DUTY 50 デューティ・サイクルを 50% に設定します。	
クエリ 例	:CURR:DUTY? >50 デューティ・サイクルは 50% に設定されています。	
関連コマンド	:CURRent:FREQuency	

 →  
 → 

#### 4-11-15. :CURRent:RECall

説明	CC スタティックモードで A または B に設定されている電流設定値を呼び出します。	
構文	:CURRent:RECall {A B 1}	
クエリ構文	:CURRent:RECall?	
パラメーター	A   0 B   1	VA を選択 VB を選択
応答パラメータ	0 1	VA が選択されています VB が選択されています
例	:CURR:REC 1 VB を選択します。	

クエリ 例 :CURR:REC?  
>0  
VA が選択されています。

## 4-12. 抵抗サブシステム・コマンド

Set →

→ Query

### 4-12-1. :RESistance[:VA]

説明	CR モード" A Value"の抵抗値の設定またはクエリ。 このコマンドは、スタティックモードに適用されます。 注: 各レンジ(H/M/L)で異なる値を設定することができます。 注: [:VA] ノードは、スタティックモードのときにのみ省略することができます。	
構文	:RESistance[:VA] {<NRf>[OHM]   MINimum   MAXimum }	
クエリ構文	:RESistance[:VA]?	
パラメーター	<NRf>[OHM]	"A Value"の抵抗値(単位はΩ)
	MINimum	最小抵抗値レベル
	MAXimum	最大抵抗値レベル
応答	"A Value"の抵抗値を"<NR2>"で返します。	
例	:RES MIN 最小の抵抗値を設定します。	
クエリ 例	:RES? >9.840 "A Value"の抵抗設定値を返します。	

### 4-12-2. :RESistance[:VA]:TRIGgered

Set →

説明	トリガーがアクティブ化された時の抵抗値を設定します。	
構文	:RESistance[:VA]:TRIGgered {<NRf>[OHM]   MINimum   MAXimum }	
パラメーター	<NRf>[OHM]	"A Value"の抵抗値(単位はΩ)
	MINimum	最小抵抗値レベル
	MAXimum	最大抵抗値レベル
例	:RES:TRIG MIN トリガーがアクティブ化された時に最小の抵抗値を設定します。	
関連コマンド	*TRG, :INITiate:CONTInuous, :INITiate[:IMMediate]	

Set →

→ Query

## 4-12-3. :RESistance:VB

説明	CR モード" B Value"の抵抗値設定またはクエリ。 このコマンドは、スタティックモードに適用されます。 注:各レンジ(H/M/L)で異なる値を設定することができます。
構文	:RESistance:VB {<NRf>[OHM]   MINimum   MAXimum }
クエリ構文	:RESistance:VB?
パラメーター	<NRf>[OHM] "B Value"の抵抗値(単位はΩ) MINimum 最小抵抗値レベル MAXimum 最大抵抗値レベル
応答	"B Value"の抵抗値を"<NR2>"で返します。
例	:RES:VB MIN 最小の抵抗値を設定します。
クエリ 例	:RES:VB? >9.840 "B Value"の抵抗設定値を返します。

Set →

→ Query

## 4-12-4. :RESistance:SRATe

説明	CR スタティックモードのコンダクタンススルーレートの設定またはクエリ。
構文	:RESistance:SRATe {<NRf>   MINimum   MAXimum }
クエリ構文	:RESistance:SRATe?
パラメーター	<NRf> コンダクタンススルーレートを mA/uS で設定 MINimum スルーレートを最小値(遅い)設定 MAXimum スルーレートを最大値(速い)設定
応答	コンダクタンススルーレートを"<NR2>"で返します。
例	:RES:SRAT MIN スルーレートを最小値(遅い)に設定します。
クエリ 例	:RES:SRAT? >5.0000 コンダクタンススルーレートを返します。

Set →

→ Query

## 4-12-5. :RESistance:L1

説明	CR モード"Level1"の抵抗値の設定またはクエリ。 このコマンドは、ダイナミックの値モードのみ適用されます。 注:各レンジ(H/M/L)で異なる値を設定することができます。
構文	:RESistance:L1 {<NRf>[OHM]   MINimum   MAXimum }

クエリ構文	:RESistance:L1?
パラメーター	<NRf>[OHM} “Level1”の抵抗値(単位はΩ) MINimum 最小抵抗値レベル MAXimum 最大抵抗値レベル
応答	“Level1”の抵抗値を"<NR2>"で返します。
例	:RES:L1 MIN 最小の抵抗値を設定します。
クエリ 例	:RES:L1? >9.840 “Level1”の抵抗設定値を返します。

Set →  
→ Query

#### 4-12-6. :RESistance:L2

説明	CR モード"Level2"の抵抗値設定またはクエリ。 このコマンドは、ダイナミックの値モードのみ適用されます。 注: 各レンジ(H/M/L)で異なる値を設定することができます。
構文	:RESistance:L2 {<NRf>[OHM}   MINimum   MAXimum }
クエリ構文	:RESistance:L2?
パラメーター	<NRf>[OHM} “Level2”の抵抗値(単位はΩ) MINimum 最小抵抗値レベル MAXimum 最大抵抗値レベル
応答	“Level2”の抵抗値を"<NR2>"で返します。
例	:RES:L2 MIN 最小の抵抗値を設定します。
クエリ 例	:RES:L2? >9.840 “Level2”の抵抗値を返します。

Set →  
→ Query

#### 4-12-7. :RESistance:SET

説明	CR ダイナミックモードが%に設定され、Level=100%の抵抗値 の設定またはクエリ。 このコマンドは、ダイナミックの%モードのみ適用されます。
構文	:RESistance:SET {<NRf>[OHM}   MINimum   MAXimum }
クエリ構文	:RESistance:SET?
パラメーター	<NRf>[OHM} “Level=100%”時の抵抗値(単位はΩ) MINimum 最小抵抗値レベル MAXimum 最大抵抗値レベル
応答	“Level=100%”時の抵抗値を"<NR2>"で返します。

例 :RES:SET MIN  
最小の抵抗値を設定します。

クエリ 例 :RES:SET?  
>9.840  
“Level=100%”時の抵抗値を返します。

Set →

4-12-8. :RESistance:LEVel

→ Query

説明 CR ダイナミックモードが%に設定されたときの%レベル(設定ミリジェムス値の割合)設定またはクエリ。このコマンドは、ダイナミックの%モードのみ適用されます。

構文 :RESistance:LEVel {<NRf> | MINimum | MAXimum }

クエリ構文 :RESistance:LEVel?

パラメーター	<NRf>	“SET”ミリジェムス値の割合(単位 [%])
	MINimum	最小コンダクタンス値の割合
	MAXimum	最大コンダクタンス値の割合

応答 “Level”のミリジェムス値の割合(%)を“<NR2>”で返します。

例 :RES:LEV MIN  
最小レベルのミリジェムス値の割合(%)を設定します。

クエリ 例 :RES:LEV?  
>50  
設定ミリジェムス値の割合は 50%に設定されています。

関連コマンド :RESistance:SET

Set →

4-12-9. :RESistance:RISE

→ Query

説明 CR ダイナミックモードのコンダクタンススルーレートの立上りを設定します。

構文 :RESistance:RISE {<NRf> | MINimum | MAXimum }

クエリ構文 :RESistance:RISE?

パラメーター	<NRf>	コンダクタンススルーレートの立上り (単位 [mA/uS])
	MINimum	最小スルーレート
	MAXimum	最大スルーレート

応答 コンダクタンススルーレートの立上りを“<NR2>”で返します。

例 :RES:RISE MIN  
スルーレートの立上りを最小に設定します。

クエリ 例 :RES:RISE?  
>50.000  
コンダクタンススルーレートの立上りを返します。

関連コマンド :RESistance:FALL

Set →

4-12-10. :RESistance:FALL

→ Query

説明	CR ダイナミックモードのコンダクタンススルーレートの立下りを設定します。	
構文	:RESistance:FALL {<NRf>   MINimum   MAXimum }	
クエリ構文	:RESistance:FALL?	
パラメーター	<NRf>	コンダクタンススルーレートの立下り (単位 [mA/uS])
	MINimum	最小スルーレート
	MAXimum	最大スルーレート
応答	コンダクタンススルーレートの立下りを"<NR2>"で返します。	
例	:RES:FALL MIN スルーレートの立下りを最小に設定します。	
クエリ 例	:RES:FALL? >50.000 コンダクタンススルーレートの立下りを返します。	
関連コマンド	:RESistance:RISE	

Set →

4-12-11. :RESistance:T1

→ Query

説明	CR ダイナミックモードの T1 タイマーの設定とクエリ。このコマンドは、ダイナミックのタイマーモードのみ適用されます。	
構文	:RESistance:T1 {<NRf>[S]   MINimum   MAXimum }	
クエリ構文	:RESistance:T1?	
パラメーター	<NRf>[S]	T1 タイマーの時間設定(単位 [秒])
	MINimum	最小時間
	MAXimum	最大時間
応答	T1 タイマーの設定値を"<NR2>"で返します。	
例	:RES:T1 0.2 T1 タイマーの設定値を設定します。	
クエリ 例	:RES:T1? >0.2 T1 タイマーの設定値を返します。	
関連コマンド	:RESistance:T2	

Set →

→ Query

## 4-12-12. :RESistance:T2

説明	CR ダイナミックモードの T2 タイマーの設定とクエリ。このコマンドは、ダイナミックのタイマーモードのみ適用されます。	
構文	:RESistance:T2 {<NRf>[S]   MINimum   MAXimum }	
クエリ構文	:RESistance:T2?	
パラメーター	<NRf>[S] MINimum MAXimum	T2 タイマーの時間設定 (単位 [秒]) 最小時間 最大時間
応答	T2 タイマーの設定値を"<NR2>"で返します。	
例	:RES:T2 0.2 T2 タイマーの設定値を設定します。	
クエリ 例	:RES:T2? >0.2 T2 タイマーの設定値を返します。	
関連コマンド	:RESistance:T1	

Set →

→ Query

## 4-12-13. :RESistance:FREquency

説明	CR ダイナミックモードのスイッチング周波数の設定とクエリ。このコマンドは、ダイナミックのデューティ・サイクルモードのみ適用されます。	
構文	:RESistance: FREquency {<NRf>   MINimum   MAXimum }	
クエリ構文	:RESistance: FREquency?	
パラメーター	<NRf> MINimum MAXimum	スイッチング周波数設定 (単位 [Hz]) 最小周波数 最大周波数
応答	周波数を Hz の単位を"<NR2>"で返します。	
例	:RES:FREQ 60 周波数を 60Hz に設定します。	
クエリ 例	:RES:FREQ? >60 スイッチング周波数は 60Hz に設定されています。	
関連コマンド	:RESistance:DUTY	

Set →

→ Query

## 4-12-14. :RESistance:DUTY

説明	CR ダイナミックモードのスイッチング周波数のための正のデューティ・サイクルの設定とクエリ。 このコマンドは、ダイナミックのデューティ・サイクルモードのみ適用されます。
構文	:RESistance:DUTY {<NRf>   MINimum   MAXimum }
クエリ構文	:RESistance:DUTY?
パラメーター	<NRf> デューティ・サイクル設定 (単位 [%]) MINimum 最小デューティ・サイクル MAXimum 最大デューティ・サイクル
応答	デューティを%単位を"<NR2>"で返します。
例	:RES:DUTY 50 デューティ・サイクルを 50%に設定します。
クエリ 例	:RES:DUTY? >50 デューティ・サイクルは 50%に設定されています。
関連コマンド	:RESistance:FREquency

Set →

→ Query

## 4-12-15. :CONDuctance[:VA]

説明	CR モード" A Value"のコンダクタンスの設定またはクエリ。 このコマンドは、スタティックモードに適用されます。 注:各レンジ(H/M/L)で異なる値を設定することができます。 注:[:VA] ノードは、スタティックモードのときにのみ省略することができます。
構文	:CONDuctance[:VA] {<NRf>[mho]   MINimum   MAXimum }
クエリ構文	:CONDuctance[:VA]?
パラメーター	<NRf>[mho] "A Value"のミリジーメンズ値(単位は mS) MINimum 最小コンダクタンスレベル MAXimum 最大コンダクタンスレベル
応答	"A Value"のミリジーメンズ値を"<NR2>"で返します。
例	:COND MIN 最小のミリジーメンズ値を設定します。
クエリ 例	:COND? >9.840 "A Value"のミリジーメンズ設定値を返します。

#### 4-12-16. :CONDuctance[:VA]:TRIGgered

Set →

説明	トリガーがアクティブ化された時のコンダクタンスを設定します。	
構文	:CONDuctance[:VA]:TRIGgered {<NRf>[mho]   MINimum   MAXimum }	
パラメーター	<NRf>[mho] MINimum MAXimum	“A Value”のミリジーメンズ値(単位は mS) 最小コンダクタンスレベル 最大コンダクタンスレベル
例	:COND:TRIG MIN トリガーがアクティブ化された時に最小のミリジーメンズ値を設定します。	
関連コマンド	*TRG, :INITiate:CONTInuous, :INITiate[:IMMediate]	

Set →

#### 4-12-17. :CONDuctance:VB

→ Query

説明	CR モード“B Value”のコンダクタンス設定またはクエリ。 このコマンドは、スタティックモードに適用されます。 注:各レンジ(H/M/L)で異なる値を設定することができます。	
構文	:CONDuctance:VB {<NRf>[mho]   MINimum   MAXimum }	
クエリ構文	:CONDuctance:VB?	
パラメーター	<NRf>[mho] MINimum MAXimum	“B Value”のミリジーメンズ値(単位は mS) 最小コンダクタンスレベル 最大コンダクタンスレベル
応答	“B Value”のミリジーメンズ値を“<NR2>”で返します。	
例	:COND:VB MIN 最小のミリジーメンズ値を設定します。	
クエリ 例	:COND:VB? >9.840 “B Value”のミリジーメンズ設定値を返します。	

Set →

#### 4-12-18. :CONDuctance:L1

→ Query

説明	CR モード“Level1”のコンダクタンスの設定またはクエリ。 このコマンドは、ダイナミックの値モードのみ適用されます。 注:各レンジ(H/M/L)で異なる値を設定することができます。	
構文	:CONDuctance:L1 {<NRf>[mho]   MINimum   MAXimum }	
クエリ構文	:CONDuctance:L1?	
パラメーター	<NRf>[mho] MINimum	“Level1”のミリジーメンズ値(単位は mS) 最小コンダクタンスレベル

MAXimum 最大コンダクタンスレベル

応答	“Level1”のミリジーメンズ値を“<NR2>”で返します。
例	:COND:L1 MIN 最小のミリジーメンズ値を設定します。
クエリ 例	:COND:L1? >9.840 “Level1”のミリジーメンズ設定値を返します。

Set →

4-12-19. :CONDuctance:L2

→ Query

説明	CR モード“Level2”のコンダクタンス設定またはクエリ。このコマンドは、ダイナミックの値モードのみ適用されます。 注:各レンジ(H/M/L)で異なる値を設定することができます。
構文	:CONDuctance:L2 {<NRf>[mho]   MINimum   MAXimum }
クエリ構文	:CONDuctance:L2?
パラメーター	<NRf>[mho] “Level2”のミリジーメンズ値(単位は mS) MINimum 最小コンダクタンスレベル MAXimum 最大コンダクタンスレベル
応答	“Level2”のミリジーメンズ値を“<NR2>”で返します。
例	:COND:L2 MIN 最小のミリジーメンズ値を設定します。
クエリ 例	:COND:L2? >9.840 “Level2”のミリジーメンズ値を返します。

Set →

4-12-20. :CONDuctance:SET

→ Query

説明	CR ダイナミックモードが%に設定され、Level=100%のコンダクタンスの設定またはクエリ。このコマンドは、ダイナミックの%モードのみ適用されます。
構文	:CONDuctance:SET {<NRf>[mho]   MINimum   MAXimum }
クエリ構文	:CONDuctance:SET?
パラメーター	<NRf>[mho] “Level=100%”時のミリジーメンズ値(単位は mS) MINimum 最小コンダクタンスレベル MAXimum 最大コンダクタンスレベル
応答	“Level=100%”時のミリジーメンズ値を“<NR2>”で返します。
例	:COND:SET MIN 最小のミリジーメンズ値を設定します。

クエリ 例 :COND:SET?  
>9.840  
“Level=100%”時のミリジーメンスを返します。

Set →

#### 4-12-21. :CONDuctance:RECall

→ Query

説明 CR スタティックモードで A または B に設定されているコンダクタンス設定値を呼び出します。

構文 : CONDuctance:RECall {A|0|B|1}

クエリ構文 : CONDuctance:RECall?

パラメーター A | 0 A を選択  
B | 1 B を選択

応答パラメータ 0 A が選択されています  
1 B が選択されています

例 :COND:REC 1  
B を選択します。

クエリ 例 :COND:REC?  
>0  
A が選択されています。

Set →

#### 4-12-22. :RESistance:RECall

→ Query

説明 CR スタティックモードで A または B に設定されている抵抗設定値を呼び出します。

構文 : RESistance:RECall {A|0|B|1}

クエリ構文 : RESistance:RECall?

パラメーター A | 0 A を選択  
B | 1 B を選択

応答パラメータ 0 A が選択されています  
1 B が選択されています

例 :RESI:REC 1  
B を選択します。

クエリ 例 :RESI:REC?  
>0  
VA が選択されています。

## 4-13. 電圧サブシステム・コマンド

Set →

### 4-13-1. :VOLTage[:VA]

→ Query

説明	CV モード" A Value"の電圧の設定またはクエリ、CV モード以外では+CV の電圧の設定またはクエリ。 注意: 同じ値が各レンジ(H/M/L)に適用されます。 注: オプションコマンドの[:VA] ノードは、スタティックモードのときにのみ省略することができます。	
構文	:VOLTage[:VA] {<NRf>[V]   MINimum   MAXimum }	
クエリ構文	:VOLTage[:VA]?	
パラメーター	<NRf>[V] MINimum MAXimum	"A Value"の電圧値 最小電圧レベル 最大電圧レベル
応答	"A Value"の電圧値を"<NR2>"で返します。	
例	:VOLT:VA MIN 最小の電圧値を設定します。	
クエリ 例	:VOLT:VA? >1.00 "A Value"の電圧設定値は 1V に設定されています。	

Set →

### 4-13-2. :VOLTage:VB

→ Query

説明	CV モード" B Value"の電圧の設定またはクエリ。 注意: 同じ値が各レンジ(H/M/L)に適用されます。	
構文	:VOLTage:VB {<NRf>[V]   MINimum   MAXimum }	
クエリ構文	:VOLT:VB?	
パラメーター	<NRf>[V] MINimum MAXimum	"B Value"の電圧値 最小電圧レベル 最大電圧レベル
応答	"B Value"の電圧値を"<NR2>"で返します。	
例	:VOLT:VB MIN 最小の電圧値を設定します。	
クエリ 例	:VOLT:VB? >1.00 "B Value" の電圧設定値は 1V に設定されています。	

Set →

→ Query

## 4-13-3. :VOLTage:RECall

説明	CV スタティックモードで A または B に設定されている電圧設定値を呼び出します。	
構文	:VOLTage:RECall {A 0 B 1}	
クエリ構文	:VOLTage:RECall?	
パラメーター	A   0 B   1	VA を選択 VB を選択
応答パラメータ	0 1	VA が選択されています VB が選択されています
例	:VOLT:REC 1 VB を選択します。	
クエリ 例	:VOLT:REC? >0 VA が選択されています。	

## 4-14. 電力サブシステム・コマンド

Set →

→ Query

## 4-14-1. :POWer[:VA]

説明	CP モード" A Value"の電力の設定またはクエリ。 このコマンドは、スタティックモードに適用されます。 注:各レンジ(H/M/L)で異なる値を設定することができます。 注:[[:VA] ノードは、スタティックモードときにのみ省略することができます。	
構文	:POWer[:VA] {<NRf>[W]   MINimum   MAXimum }	
クエリ構文	:POWer[:VA]?	
パラメーター	<NRf>[W] MINimum MAXimum	"A Value" の電力値 最小電力レベル 最大電力レベル
応答	"A Value"の電力値を"<NR2>"で返します。	
例	:POW:VA MIN 最小電力値を設定します。	
クエリ 例	:POW:VA? >10 "A Value" の電力設定(10W)を返します。	

Set →

→ Query

## 4-14-2. :POWer:VB

説明	CP モード" B Value"の電力の設定またはクエリ。このコマンドは、スタティックモードに適用されます。 注:各レンジ(H/M/L)で異なる値を設定することができます。	
構文	:POWer:VB {<NRf>[W]   MINimum   MAXimum }	
クエリ構文	:POWer:VB?	
パラメーター	<NRf>[W] MINimum MAXimum	"B Value"の電力値 最小電力レベル 最大電力レベル
応答	"B Value"の電力値を"<NR2>"で返します。	
例	:POW:VB MIN 最小電力値を設定します。	
クエリ 例	:POW:VB? >10 "B Value" の電力設定値は 10W に設定されています。	

Set →

→ Query

## 4-14-3. :POWer:L1

説明	CP モード"Level1"の電力の設定またはクエリ。このコマンドは、ダイナミックの値モードのみ適用されます。 注:各レンジ(H/M/L)で異なる値を設定することができます。	
構文	:POWer:L1 {<NRf>[W]   MINimum   MAXimum }	
クエリ構文	:POWer:L1?	
パラメーター	<NRf>[W] MINimum MAXimum	"Level1"の電力値 最小電力レベル 最大電力レベル
応答	"Level1"の電力値を"<NR2>"で返します。	
例	:POW:L1 MIN 最小電力値を設定します。	
クエリ 例	:POW:L1? >10 "Lewel1"の電力設定値は 10W に設定されています。	

Set →

→ Query

## 4-14-4. :POWer:L2

説明	CP モード"Level2"の電力の設定またはクエリ。このコマンドは、ダイナミックの値モードのみ適用されます。 注:各レンジ(H/M/L)で異なる値を設定することができます。	
構文	:POWer:L2 {<NRf>[W]   MINimum   MAXimum }	
クエリ構文	:POWer:L2?	
パラメーター	<NRf>[W] MINimum MAXimum	"Level2"の電力値 最小電力レベル 最大電力レベル
応答	"Level2"の電力値を"<NR2>"で返します。	
例	:POW:L2 MIN 最小電力値を設定します。	
クエリ 例	:POW:L2? >10 "Level 2" の電力設定値は 10W に設定されています。	

Set →

→ Query

## 4-14-5. :POWer:SET

説明	CP ダイナミックモードが%に設定されたときの電力の設定またはクエリ。このコマンドは、ダイナミックの%モードのみ適用されます。	
構文	:POWer:SET {<NRf>[W]   MINimum   MAXimum }	
クエリ構文	:POWer:SET?	
パラメーター	<NRf>[W] MINimum MAXimum	"Level=100%"時の電力値 最小電力値 最大電力値
応答	"Level=100%"時の電力値を"<NR2>"で返します。	
例	:POW:SET MIN "Level=100%"の最小電力値を設定します。	
クエリ 例	:POW:SET? >10W "Level=100%"の電力値は 10W に設定されています。	
関連コマンド	:POWer:LEVel	

Set →

→ Query

## 4-14-6. :POWer:LEVel

説明	CP ダイナミックモードが%に設定されたときの%レベル(設定電力値の割合)設定またはクエリ。このコマンドは、ダイナミックの%モードのみ適用されます。	
構文	:POWer:LEVel {<NRf>   MINimum   MAXimum }	
クエリ構文	:POWer:LEVel?	
パラメーター	<NRf>	“SET”電力値の割合 (単位 [%])
	MINimum	最小“SET”電力値の割合
	MAXimum	最大“SET”電力値の割合
応答	電力の“% Level” を"<NR2>"で返します。	
例	:POW:LEV MIN 最小レベルの電力値の割合(%)を設定します。	
クエリ 例	:POW:LEV? >50 設定電力値の割合は 50%に設定されています。	
関連コマンド	:POWer:SET	

Set →

→ Query

## 4-14-7. :POWer:T1

説明	CP ダイナミックモードの T1 タイマーの設定とクエリ。このコマンドは、ダイナミックのタイマーモードのみ適用されます。	
構文	:POWer:T1 {<NRf>[S]   MINimum   MAXimum }	
クエリ構文	:POWer:T1?	
パラメーター	<NRf>[S]	T1 タイマーの時間設定 (単位 [秒])
	MINimum	最小時間
	MAXimum	最大時間
応答	T1 タイマーの設定を"<NR2>"で返します。	
例	:POW:T1 200 T1 タイマーの設定値を設定します。	
クエリ 例	:POW:T1? >200 T1 タイマーの設定値を返します。	
関連コマンド	:POWer:T2	

Set →

→ Query

## 4-14-8. :POWer:T2

説明	CP ダイナミックモードの T2 タイマーの設定とクエリ。このコマンドは、ダイナミックのタイマーモードのみ適用されます。	
構文	:POWer:T2 {<NRf>[S]   MINimum   MAXimum }	
クエリ構文	:POWer:T2?	
パラメーター	<NRf>[S] MINimum MAXimum	T2 タイマーの時間設定 (単位 [秒]) 最小時間 最大時間
応答	秒単位で T2 タイマーの設定を"<NR2>"で返します。	
例	:POW:T2 200 T2 タイマーの時間を 200ms に設定します。	
クエリ 例	:POW:T2? >200 T2 タイマー時間は 200ms に設定されています。	
関連コマンド	:POWer:T1	

Set →

→ Query

## 4-14-9. :POWer:FREQuency

説明	CP ダイナミックモードのスイッチング周波数の設定とクエリ。このコマンドは、ダイナミックのデューティ・サイクルモードのみ適用されます。	
構文	:POWer:FREQuency {<NRf>   MINimum   MAXimum }	
クエリ構文	:POWer:FREQuency?	
パラメーター	<NRf> MINimum MAXimum	スイッチング周波数設定 (単位 [Hz]) 最小周波数 最大周波数
応答	周波数を Hz の単位を"<NR2>"で返します。	
例	:POW:FREQ 60 周波数を 60Hz に設定します。	
クエリ 例	:POW:FREQ? >60 スイッチング周波数は 60Hz に設定されています。	
関連コマンド	:POWer:DUTY	

Set →

→ Query

## 4-14-10. :POWER:DUTY

説明	CP ダイナミックモードのスイッチング周波数のための正のデューティ・サイクルの設定とクエリ。 このコマンドは、ダイナミックのデューティ・サイクルモードのみ適用されます。	
構文	:POWER:DUTY {<NRf>   MINimum   MAXimum }	
クエリ構文	:POWER:DUTY?	
パラメーター	<NRf> MINimum MAXimum	デューティ・サイクル設定 (単位 [%]) 最小デューティ・サイクル 最大デューティ・サイクル
応答	デューティを%単位を"<NR2>"で返します。	
例	:POW:DUTY 50 デューティ・サイクルを 50%に設定します。	
クエリ 例	:POW:DUTY? >50 デューティ・サイクルは 50%に設定されています。	
関連コマンド	:POWER:FREQuency	

Set →

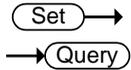
→ Query

## 4-14-11. :POWER:RECall

説明	CP スタティックモードで A または B に設定されている電力設定値を呼び出します。	
構文	: POWER:RECall {A 0 B 1}	
クエリ構文	: POWER:RECall?	
パラメーター	A   0 B   1	VA を選択 VB を選択
応答パラメータ	0 1	VA が選択されています VB が選択されています
例	:POW:REC 1 VB を選択します。	
クエリ 例	:POW:REC? >0 VA が選択されています。	

#### 4-15. プログラム・コマンド

プログラムの実行・停止は:INPUT コマンドを使用します。4-3-1. :INPutを参照してください。



##### 4-15-1. :FUNCTION[:COMPLete][:RING]:TIME

説明	プログラム動作完了後のブザー時間の設定	
構文	:FUNCTION[:COMPLete][:RING]:TIME<NR1> MINimum MAXimum INFinity	
クエリ構文	:FUNCTION[:COMPLete][:RING]:TIME? [MINimum MAXimum]	
パラメーター	<NR1> MINimum MAXimum INFinity	ブザー秒数 0 でブザーなし 最小時間を設定 最大時間を設定 連続
応答パラメータ	<NR1> INFinity OFF	秒数を応答 連続時の応答 ブザーなしの応答
例	:FUNC:TIME 5 5 秒を設定します。	
クエリ 例	:FUNC:TIME? >5 ブザーは 5 秒です。	

Set →

→ Query

#### 4-15-2. :PROGm

説明	プログラムモードの指定ステップへの全パラメーターの設定とクエリ。	
構文	:PROGm (1),(2),(3),(4),(5),(6),(7),(8)	
クエリ構文	:PROGm?	
パラメーター	(1) <NR1> (2) <NR1> (3) <NR1> (4) <ASCII string>  (5) <NRf> (6) <NRf> (7) <NRf> (8) <NRf>	プログラム番号 ステップ番号 内部メモリ番号 処理設定 AUTO 処理を実行 MANUAL 処理起動を待つ SKIP 次のステップに進む オン時間を秒で設定 オフ時間を秒で設定 P / F の遅延時間を秒で設定 ショート時間を秒で設定
応答	(1) <ASCII string> (2) <ASCII string> (3) <ASCII string> (4) <ASCII string> (5) <ASCII string> (6) <ASCII string> (7) <ASCII string> (8) <ASCII string> (9) <ASCII string>	プログラムモードを返します。{ON OFF} プログラム番号を返します。 ステップ番号を返します。 内部メモリ番号を返します。 処理を Auto/Manual/Skip で返します。 オン時間を秒で返します。 オフ時間を秒で返します。 P / F の遅延時間を秒で返します。 ショート時間を秒で返します。
例	:PROG 2,3,1,AUTO,40.1,0,0,0 指定のプログラムステップに全パラメータを設定します。	
クエリ 例	:PROG ? >Program:OFF; Start:1, Step:1, Memory:1, Run:Skip, On-Time:0.1, Off-Time:0.0, P/F-Time:0.0, Short-Time:0.0 プログラムモードの状態の後に選択されているプログラムステップのパラメータを返します。	

Set →

→ Query

#### 4-15-3. [:PROGm]:CHAn

説明                    プログラムチェーンの全チェーンの設定とクエリ

構文	[:PROGrama]:CHAIIn (1),(2),(3),(4),(5),(6),(7),(8),(9),(10),(11),(12),(13),(14),(15), (16)	
クエリ構文	[:PROGrama]:CHAIIn?	
パラメーター	(1)~(16) {<NR1>   OFF}	チェーンするプログラム番号 "OFF"はチェーンオフの設定
応答	<ASCII string>	全プログラムチェーンの設定値を返します。 "Off"はチェーンの終了です。
例	:CHA OFF,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 プログラム 1 はチェーンのオフ設定、プログラム 2~16 は チェーン設定します。	
クエリ 例	:CHA? >P1->Off;P2->P1;P3->P2;P4->P3;P5->P4;P6->P5;P7->P6 ;P8->P7;P9->P8;P10->P9;P11->P10;P12->P11;P13->P1 2;P14->P13;P15->P14;P16->P15 全プログラムチェーンの設定値を返します。	

Set →

→ Query

#### 4-15-4. [:PROGrama]:CHAIIn:P2P

説明	指定プログラムのチェーンの設定とクエリ	
構文	[:PROGrama]:CHAIIn:P2P {P1 P2 P3 P4 P5 P6 P7 P8 P9  P10 P11 P12 P13 P14 P15 P16}	
クエリ構文	[:PROGrama]:CHAIIn:P2P?	
パラメーター	Pn n:1~16	確認するプログラム番号
応答	<ASCII string>	チェーン設定を返します。
例	:CHA:P2P 4,3 プログラム 4 はプログラム 3 にチェーン設定します。	
クエリ 例	:CHA:P2P? P4 >P4->6 P4 の次のチェーンは P6 です。	

#### 4-15-5. [:PROGrama]:CHAIIn[:RECall]:DEFault

Set →

説明	プログラムチェーンを全てオフに設定	
構文	[:PROGrama]:CHAIIn[:RECall]:DEFault	
例	:CHA:DEF 全プログラムのチェーン設定をオフにします。	

Set →

#### 4-15-6. [:PROGrama]:CHAIIn:START

→ Query

説明	プログラムチェーンの開始プログラム番号の設定とクエリ	
----	----------------------------	--

構文	[:PROGram]:CHAIIn:STARt <NR1>
クエリ構文	[:PROGram]:CHAIIn:STARt?
パラメーター	<NR1> 開始プログラム番号
応答	開始プログラム番号を"P<NR1>"で返します。
例	:CHA:STAR 1 開始プログラムを 1 に設定します。
クエリ 例	:CHA:STAR? >P1 開始プログラム番号を返します。

Set →

#### 4-15-7. :PROGram:MEMory

→ Query

説明	選択されているプログラムステップのメモリー番号の設定とクエリ
構文	:PROGram:MEMory <NR1>
クエリ構文	:PROGram:MEMory?
パラメーター	<NR1> メモリー番号の設定
応答	設定中のメモリー番号を"Memory:M<NR1>"で返します。
例	:PROG:MEM 1 メモリー番号を 1 に設定します。
クエリ 例	:PROG:MEM? >Memory:M 1 設定中のメモリー番号を返します。

Set →

#### 4-15-8. :PROGram:OFFTime

→ Query

説明	選択されているプログラムステップのオフ時間の設定とクエリ
構文	:PROGram:OFFTime {<NRf>   OFF}
クエリ構文	:PROGram:OFFTime?
パラメーター	<NRf> オフ時間を秒で設定 0 秒設定は機能オフ設定 OFF 機能オフ設定
応答	オフ時間を"Off-Time:<NR2>"で返します。 "Off-Time:0.0"は機能オフです。
例	:PROG:OFFT 1 オフ時間を 1 秒に設定します。
クエリ 例	:PROG:OFFT? >Off-Time:1.0 オフ時間を返します。
関連コマンド	:PROGram:PFTIME

Set →

→ Query

## 4-15-9. :PROG:ONTime

説明	選択されているプログラムステップのオン時間の設定とクエリ
構文	:PROG:ONTime <NRf>
クエリ構文	:PROG:ONTime?
パラメーター	<NRf> オン時間を秒で設定
応答	オン時間を"On-Time:<NR2>"で返します。
例	:PROG:ONT 1 オン時間を 1 秒に設定します。
クエリ 例	:PROG:ONT? >On-Time:0.1 オン時間を返します。
関連コマンド	:PROG:PFTime :PROG:STIME

Set →

→ Query

## 4-15-10. :PROG:PFTime

説明	選択されているプログラムステップの合格/不合格判定 (Go-NoGo テスト) および判定遅延時間の設定とクエリ
構文	:PROG:PFTime {<NRf>   OFF}
クエリ構文	:PROG:PFTime?
パラメーター	<NRf> 判定遅延時間を秒で設定 0 秒設定は判定機能オフ設定 OFF 判定機能オフ設定
応答	判定遅延時間を"P/F-Time:<NR2>"で返します。 "P/F-Time:0.0"は機能オフです。
例	:PROG:PFT 1 判定遅延時間を 1 秒に設定します。
クエリ 例	:PROG:PFT >P/F-Time:1.0 判定遅延時間を返します。
関連コマンド	:PROG:OFFTime :PROG:ONTime

Set →

## 4-15-11. :PROG[:RECall]:DEFault

説明	選択されているプログラムの全ステップの初期値の設定
構文	:PROG[:RECall]:DEFault
例	:PROG:DEF 選択されているプログラムの全ステップを初期値にします。

Set →

→ Query

## 4-15-12. :PROG:RUN

説明	選択されているプログラムステップの実行処理の設定とクエリ	
構文	:PROG:RUN {AUTO   MANUAL   SKIP}	
クエリ構文	:PROG:RUN?	
パラメーター	AUTO MANUAL SKIP	自動実行処理に設定 実行指定待ちに設定 次のステップ処理に設定
応答	実行処理を"Run:{Auto   Manual   Skip}"で返します。	
例	:PROG:RUN AUTO 自動実行処理に設定します。	
クエリ 例	:PROG:RUN? >Run:Auto 実行処理の設定を返します。	

## 4-15-13. :PROG:SAVE

Set →

説明	プログラムの保存
構文	:PROG:SAVE
例	:PROG:SAVE プログラムを保存します。

Set →

## 4-15-14. :PROG:STARt

→ Query

説明	選択するプログラム番号の設定とクエリ	
構文	:PROG:STARt <NR1>	
クエリ構文	:PROG:STARt?	
パラメーター	<NR1>	選択するプログラム番号
応答	選択されているプログラム番号を"Start:<NR1>"で返します。	
例	:ROG:STAR 1 プログラム番号を 1 に設定します。	
クエリ 例	:PROG:STAR? >Start:1 選択されているプログラム番号を返します。	

Set →

## 4-15-15. :PROG:STATe

→ Query

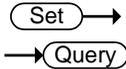
説明	プログラムモードの選択と動作中の設定とクエリ	
構文	:PROG:STATe { OFF ON PAUSE CONTINUE NEXT}	
クエリ構文	:PROG:STATe?	

パラメーター	OFF ON PAUSE CONTINUE NEXT	プログラムモードのオフ設定 プログラムモードのオン設定 プログラム実行中の中断設定 プログラム中断中の解除 プログラム手動開始待ちの解除
--------	--	--

応答 プログラムモードの状態を返します。  
プログラムモードがオンの時は“ON,{STOP | RUN | PAUSE}”  
で返します。  
 (“STOP”: 停止中、“RUN”: 実行中、“PAUSE”: 中断中)  
プログラムモードがオフの時は“OFF”を返します。

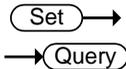
例 :PROG:STAT ON  
プログラムモードに設定します。

クエリ 例 :PROG:STAT?  
>ON,STOP  
プログラムモードの状態を返します。



#### 4-15-16. :PROG:STEP

説明	選択するプログラムのステップ番号の設定とクエリ
構文	:PROG:STEP <NR1>
クエリ構文	:PROG:STEP?
パラメーター	<NR1> 選択するステップ番号
応答	選択されているステップ番号を“Step:<NR1>”で返します。
例	:PROG:STEP 1 ステップ番号を 1 に設定します。
クエリ 例	:PROG:STEP? >Step:1 選択されているステップ番号を返します。



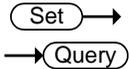
#### 4-15-17. :PROG:STIME

説明	選択されているプログラムステップの負荷ショート時間の設定とクエリ
構文	:PROG:STIME <NR1>
クエリ構文	:PROG:STIME?
パラメーター	<NRf> 負荷ショート時間を秒で設定 0 設定は負荷ショート機能オフ設定 OFF 負荷ショート機能オフ設定
応答	負荷ショート機能の時間を“Short-Time:<NR2>”で返します。 “Short-Time:0.0”は機能オフです。

例	:PROG:STIM 1 負荷ショート機能を 1 秒に設定します。
クエリ 例	:PROG:STIM? >Short-Time:0.0 負荷ショート機能の時間を返します。
関連コマンド	:PROG:ONTime

#### 4-16. ノーマルシーケンス・コマンド

シーケンスの実行・停止は:INPUT コマンドを使用します。4-3-1. :INPutを参照してください。



##### 4-16-1. :NSEquence

説明	ノーマルシーケンスの指定シーケンス番号への全パラメータの設定とクエリ。 注意)ステップデータがある場合には負荷のモード及びレンジは変更できません。	
構文	:NSEquence (1),(2),(3),(4),(5),(6),(7),(8),(9)	
クエリ構文	:NSEquence?	
パラメーター	(1) <NR1> (2) <NR1> (3) <ASCII string> (4) <ASCII string> (5) <ASCII string> (6) <ASCII string> (7) <ASCII string> (8) <NRf> (9) <ASCII string>	スタートシーケンス番号設定 シーケンス番号設定 12 文字までのメモ設定 ダブルコートで文字列を括ります。 負荷モード設定 {CC   CR   CV   CP} 負荷レンジ設定 {IHVH   IMVH   ILVH   IHVL   IMVL   ILVL} シーケンスのループ回数設定 <NR1> 0は無限回, 1~9999 の数 INFinity 無限回数 終了後の負荷条件設定{ON   OFF} 終了後の負荷設定値 (負荷条件 ON で有効になります。) 次のシーケンスのチェーン番号設定 <NR1> シーケンス番号 OFF チェーン無し
応答	(1) <ASCII string> (2) <ASCII string>	ノーマルシーケンスモードかを返します。{ON OFF} スタートシーケンス番号設定を返します。

(3) <ASCII string>	シーケンス番号設定を返します。
(4) <ASCII string>	メモの内容を返します。
(5) <ASCII string>	負荷モード設定を返します。{CC   CR   CV   CP}
(6) <ASCII string>	負荷レンジ設定を返します。{IHVH   IMVH   ILVH   IHVL   IMVL   ILVL}
(7) <ASCII string>	ループの回数設定を返します。(Infinity は無限回数)
(8) <ASCII string>	終了後の負荷条件モードを返します。{ON OFF}
(9) <ASCII string>	終了後の負荷設定を返します。
(10)<ASCII string>	次のシーケンスのチェーン設定を返します。(Off はチェーン無し)

例 :NSEQ 1,1,"ABC",CC,ILVL,5,ON,1.5000,OFF  
 スタートシーケンス番号と指定のシーケンス番号にパラメータを設定します。

クエリ 例 :NSEQ?  
 >NSeq:ON; Start:1, Seq No:1, Memo:ABC, Mode:CC, Range:ILVL, Loop:5, Last Load:ON, Last:1.5000, Chain:Off  
 ノーマルシーケンスモードの状態の後に選択されているプログラムステップのパラメータを返します。

Set →

→ Query

#### 4-16-2. :NSEquence:CHAIIn

説明	選択されているノーマルシーケンスのチェーン番号の設定とクエリ
構文	:NSEquence:CHAIIn {<NR1>   OFF}
クエリ構文	:NSEquence:CHAIIn?
パラメーター	<NR1> チェーンするノーマルシーケンス番号 OFF チェーン終了設定
応答	チェーンするノーマルシーケンス番号を"Chain:{<NR1> Off}"で返します。
例	:NSEQ:CHA 1 プログラムチェーンを 1 に設定する。
クエリ 例	:NSEQ:CHA? >Chain:1 プログラムチェーンの設定値を返します。

4-16-3. :NSEquence[:DEL]:ALL

Set →

説明	選択されているノーマルシーケンスの全ステップの削除
構文	:NSEquence[:DEL]:ALL
例	:NSEQ:ALL 選択されているノーマルシーケンスの全ステップの削除します。

4-16-4. :NSEquence:EDIT

Set →  
→ Query

説明	選択されているノーマルシーケンスのステップデータの設定とクエリ	
構文	:NSEquence:EDIT (1),(2),(3),(4),(5),(6),(7),(8),(9),(10),(11)	
クエリ構文	:NSEquence:EDIT?	
パラメーター	(1) <NR1> (2) <NR1> (3) <NRf> (4) <NR1> (5) <NR1> (6) <NR1> (7) <NR1> (8) <ASCII string> (9) <ASCII string> (10)<ASCII string> (11)<ASCII string>	編集ステップ番号 ステップ総数 動作モードの負荷設定値 時設定 分設定 秒設定 ミリ秒設定 負荷出力設定 {ON   OFF} ランプ動作設定 {ON   OFF} TRIG OUT 設定 {ON   OFF} 一時停止設定 {ON   OFF}
応答	(1) <ASCII string> (2) <ASCII string> (3) <ASCII string> (4) <ASCII string> (5) <ASCII string> (6) <ASCII string> (7) <ASCII string>	編集ステップ/ステップ総数を返します。 動作モードの負荷設定値 設定時間を返します。 負荷出力設定を返します。 ランプ動作設定を返します。 TRIG OUT 設定を返します。 一時停止設定を返します。
例	:NSEQ:EDIT 1,2,1,1,2,3,4,OFF,OFF,OFF,OFF ノーマルシーケンスプログラムのステップデータを設定する。	

クエリ 例 :NSEQ:EDIT?  
>Step:1/1, Value:0, Time:0H:0M:0S:1mS, LOAD:OFF,  
TRIG OUT:OFF, RAMP:OFF, PAUSE:OFF  
選択されているノーマルシーケンスプログラムのステップデータ  
を返します。

Set →

4-16-1. :NSEquence:EDIT:POINT

→ Query

説明 現在のノーマルシーケンスのデータ編集ステップ番号の設定とクエリ。

構文 :NSEquence:EDIT:POINT {<NR1>}

クエリ構文 :NSEquence:EDIT:POINT?

パラメーター <NR1> データ編集ステップ番号 1~1000

例 :NSEquence:EDIT:POINT 10  
データ編集ステップ番号 10 を設定します。

4-16-1. :NSEquence:EDIT:END

→ Query

説明 現在のノーマルシーケンスの一番最後のデータ編集ステップ  
番号のクエリ

クエリ構文 :NSEquence:EDIT:END?

パラメーター <NR1> 1~1000

クエリ 例 :NSEquence:EDIT:END?  
> 20  
一番最後のデータ編集ステップ番号を応答します。

Set →

4-16-2. :NSEquence:LAST

→ Query

説明 選択されているノーマルシーケンスの終了後の負荷値の設定  
とクエリ

構文 :NSEquence:LAST <NRf>

クエリ構文 :NSEquence:LAST?

パラメーター <NRf> ステップ終了後の負荷設定値

応答 ステップ終了後の負荷値を"Last:<NR2>"で返します。

例 :NSEQ:LAST 1  
ステップ終了後の負荷値を設定します。

クエリ 例 :NSEQ:LAST?  
>Last:1.00  
ステップ終了後の負荷値を返します。

関連コマンド :NSEquence:LLOad

Set →

#### 4-16-3. :NSEquence:LLOad

→ Query

説明 選択されているノーマルシーケンスの終了後の負荷設定のオンまたはオフの設定とクエリ

構文 :NSEquence:LLOad {ON | OFF| < Boolean >}

クエリ構文 :NSEquence:LLOad?

パラメーター ON ステップ終了後の負荷設定をオン  
OFF ステップ終了後の負荷設定をオフ

応答 ステップ終了後の負荷設定を"Last Load:{ON | OFF}"で返します。

例 :NSEQ:LLO ON  
ステップ終了後の負荷の出力設定をします。

クエリ 例 :NSEQ:LLO?  
>Last Load:OFF  
ステップ終了後の負荷の出力設定を返します。

関連コマンド :NSEquence:LAST

Set →

#### 4-16-4. :NSEquence:LOOP

→ Query

説明 ノーマルシーケンスのループ回数の設定とクエリ

構文 :NSEquence:LOOP {<NR1> | INFinity }

クエリ構文 :NSEquence:LOOP?

パラメーター <NR1> ループ回数設定 (1~9999)  
INFinity 無限回数

応答 ノーマルシーケンスのループ回数を"Loop:{<NR1> | Infinity}"で返します。

例 :NSEQ:LOOP 1  
ノーマルシーケンスのループ回数を1に設定します。

クエリ 例 :NSEQ:LOOP?  
>Loop:Infinity  
ノーマルシーケンスのループ回数を返します。

Set →

#### 4-16-5. :NSEquence:MEMO

→ Query

説明 選択されているノーマルシーケンスのメモの設定とクエリ

構文 :NSEquence:MEMO <ASCII string>

クエリ構文	:NSEquence:MEMO?
パラメーター	<ASCII string> 12 文字までのメモを設定 ダブルコートで文字列を括ります。
応答	ステップのメモを"Memo:<ASCII string>"で返します。
例	:NSEQ:MEMO "ABCD" ステップのメモを設定します。
クエリ 例	:NSEQ:MEMO? >Memo:ABCD ステップのメモを返します。

Set →

#### 4-16-6. :NSEquence:MODE

→ Query

説明	選択されているノーマルシーケンスプログラムの動作モードの設定とクエリ
構文	:NSEquence:MODE {CC   CR   CV   CP}
クエリ構文	:NSEquence:MODE?
パラメーター	CC 定電流 モードに設定 CR 定抵抗 モードに設定 CV 定電圧 モードに設定 CP 定電力 モードに設定
応答	ノーマルシーケンスプログラムの動作モードを"Mode:{CC   CR   CV   CP}"で返します。
例	:NSEQ:MODE CC ノーマルシーケンスの動作モードを設定します。
クエリ 例	:NSEQ:MODE? >Mode:CC ノーマルシーケンスの動作モードを返します。

Set →

#### 4-16-7. :NSEquence:NUMBer

→ Query

説明	ノーマルシーケンスのシーケンス番号の設定とクエリ
構文	:NSEquence:NUMBer <NR1>
クエリ構文	:NSEquence:NUMBer?
パラメーター	<NR1> シーケンス番号の設定
応答	選択されているシーケンス番号を"Seq No:<NR1>"で返します。
例	:NSEQ:NUMB 1 ノーマルシーケンス番号を 1 に設定します。

クエリ 例 :NSEQ:NUMB?  
 >Seq No:1  
 選択されているシーケンス番号を返します。

Set →

#### 4-16-8. :NSEquence:RANGe

→ Query

説明	選択されているノーマルシーケンスの動作レンジの設定とクエリ	
構文	:NSEquence:RANGe { IHVH   IMVH   ILVH   IHVL   IMVL   ILVL }	
クエリ構文	:NSEquence:RANGe?	
パラメーター	IHVH	電流 High, 電圧 High レンジ設定
	IMVH	電流 Middle, 電圧 High レンジ設定
	ILVH	電流 Low, 電圧 High レンジ設定
	IHVL	電流 High, 電圧 Low レンジ設定
	IMVL	電流 Middle, 電圧 Low レンジ設定
	ILVL	電流 Low, 電圧 Low レンジ設定
応答	設定されている動作レンジを"Range:{IHVH   IMVH   ILVH   IHVL   IMVL   ILVL}"で返します。	
例	:NSEQ:RANG IHVL 電流 High, 電圧 Low の動作レンジを設定します。	
クエリ 例	:NSEQ:RANG? >Range:IHVL 設定されている動作レンジを返します。	

#### 4-16-9. :NSEquence:SAVE

Set →

説明	ノーマルシーケンスのプログラム保存	
構文	:NSEquence:SAVE	
例	:NSEQ:SAVE ノーマルシーケンスのプログラム保存します。	

Set →

#### 4-16-10. :NSEquence:START

→ Query

説明	ノーマルシーケンスのスタートシーケンス番号の設定とクエリ	
構文	:NSEquence:START <NR1>	
クエリ構文	:NSEquence:START?	
パラメーター	<NR1>	スタートシーケンス番号の設定
応答	ノーマルシーケンスのスタートシーケンス番号を"Start:<NR1>"で返します。	

例 :NSEQ:STAR 1  
ノーマルシーケンスのスタートシーケンス番号を設定します。

クエリ 例 :NSEQ:STAR?  
>Start:1  
ノーマルシーケンスのスタートシーケンス番号を返します。

Set →

#### 4-16-11. :NSEquence:STATe

→ Query

説明	ノーマルシーケンスモードの選択と動作中の設定とクエリ	
構文	:NSEquence:STATe {OFF ON PAUSE CONTINUE NEXT}	
クエリ構文	:NSEquence:STATe?	
パラメーター	OFF	ノーマルシーケンスモードのオフ設定
	ON	ノーマルシーケンスモードのオン設定
	PAUSE	ノーマルシーケンス実行中の中断設定
	CONTINUE	ノーマルシーケンス中断中の解除
	NEXT	ノーマルシーケンス手動開始待ちの解除
応答	ノーマルシーケンスモードの状態を返します。 ノーマルシーケンスモードがオンの時は“ON,{ STOP   RUN   PAUSE }”で返します。 (“STOP”:停止中、“RUN”:実行中、“PAUSE”:中断中) ノーマルシーケンスモードがオフの時は“OFF”を返します。	

例 :NSEQ:STAT ON  
ノーマルシーケンスモードに設定します。

クエリ 例 :NSEQ:STAT?  
>ON,STOP  
ノーマルシーケンスモードの状態を返します。

Set →

#### 4-16-1. :NSEquence:COTime

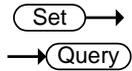
→ Query

説明	ノーマルシーケンスの時間表示を選択します。	
構文	:NSEquence:COTime {UP DOWN}	
クエリ構文	:NSEquence:COTime?	
パラメーター	UP	スタートからの経過時間を表示
	DOWN	エンドまでの残り時間を表示
応答	ノーマルシーケンスの時間表示を応答します。	
例	:NSEQ:COT UP 時間表示に経過時間を設定します。	

クエリ 例 :NSEQ:COT?  
 >UP  
 時間表示は経過時間です。

#### 4-17. ファストシーケンス・コマンド

シーケンスの実行・停止は:INPUTコマンドを使用します。4-3-1. :INPutを参照してください。



##### 4-17-1. :FSEquence

説明	ファストシーケンスの全パラメーターの設定とクエリ。	
構文	:FSEquence (1),(2),(3),(4),(5),(6),(7)	
クエリ構文	:FSEquence?	
パラメーター	(1) <ASCII string> (2) <ASCII string> (3) <ASCII string> (4) <ASCII string> (5) <NRf> (6) <ASCII string> (7) <NRf> (8) <NR1>	12 文字までのメモ設定 ダブルコートで文字列を括ります。 負荷モード設定 {CC   CR} 負荷レンジ設定 {IHVH   IMVH   ILVH   IHVL   IMVL   ILVL} シーケンスのループ回数設定 <NR1> 1~9999 の数 INFinity 無限回数 タイムベースを設定 (単位 [秒]) 終了後の負荷条件設定{ON   OFF} 終了後の負荷設定 (負荷条件 ON で有効になります。) ループ最後のステップ番号設定
応答	(1) <ASCII string> (2) <ASCII string> (3) <ASCII string> (4) <ASCII string> (5) <ASCII string> (6) <ASCII string> (7) <ASCII string> (8) <ASCII string> (9) <ASCII string>	ファストシーケンスモードのオフ設定 メモの内容を返します。 負荷モード設定を返します。{CC   CR} 負荷レンジ設定を返します。{IHVH   IMVH   ILVH   IHVL   IMVL   ILVL} シーケンスのループ回数設定を返します。(Infinity は無限回数) タイムベースを返します。(単位 [ミリ秒]) 終了後の負荷条件モードを返します。{ON OFF} 終了後の負荷設定を返します。 ループ最後のステップ番号を返します。
例	:FSEQ "ABC",CC,IHVL,1,0.025,OFF,1.0,1 ファストシーケンスモードのタイミングにパラメータを設定します。	

クエリ 例 :FSEQ?  
 >FSeq:OFF; Memo:ABC, Mode:CC, Range:IHVL, Loop:1,  
 Time Base:25.000, Last Load:OFF, Last:1.00,  
 RPTSTEP:1  
 ファストシーケンスモードの状態の後にタイミングの内容を返します。

#### 4-17-2. :FSEquence[:DEL]:ALL

Set →

説明 ファストシーケンスの全プログラム削除

構文 :FSEquence[:DEL]:ALL

例 :FSEQ:ALL

ファストシーケンスの全プログラム削除

Set →

#### 4-17-3. :FSEquence:EDIT

→ Query

説明 ファストシーケンスのデータの設定とクエリ

構文 :FSEquence:EDIT (1),(2),(3),(4)

クエリ構文 :FSEquence:EDIT?

パラメーター (1) <NR1> 編集ステップ  
 (2) <NR1> ステップ総数  
 (3) <NRf> 動作モードの負荷値の設定  
 (4) <ASCII string> TRIG OUT 設定 {ON | OFF}

応答 (1) <ASCII string> 編集ステップとステップ総数を返します。  
 (2) <ASCII string> 動作モードの負荷設定値を返します。  
 (3) <ASCII string> TRIG OUT 設定を返します。

例 :FSEQ:EDIT 2,6,1,ON

ファストシーケンスのデータを設定します。

クエリ 例 :FSEQ:EDIT?

>Step:0001/0003; Value:0.00, TRIG OUT:OFF  
 ファストシーケンスのデータを返します。

Set →

#### 4-17-4. :FSEquence:EDIT:POINT

→ Query

説明 現在のファストシーケンスのデータ編集ステップ番号の設定とクエリ。

構文 :FSEquence:EDIT:POINT {<NR1>}

クエリ構文 :FSEquence:EDIT:POINT?

パラメーター <NR1> データ編集ステップ番号 1~1000

例	:FSEquence:EDIT:POINT 10
クエリ 例	データ編集ステップ番号 10 を設定します。

#### 4-17-5. :FSEquence:EDIT:END

→ Query

説明	現在のファストシーケンスの一番最後のデータ編集ステップ番号のクエリ
クエリ構文	:FSEquence:EDIT:END?
パラメーター	<NR1> 1~1000
クエリ 例	:FSEquence:EDIT:END? > 20 一番最後のデータ編集ステップ番号を応答します。

Set →

#### 4-17-6. :FSEquence[:EDIT]:FILL

→ Query

説明	ファストシーケンスの FILL の設定とクエリ
構文	:FSEquence[:EDIT]:FILL (1),(2),(3),(4)
クエリ構文	:FSEquence[:EDIT]:FILL?
パラメーター	(1) <NRf> 動作モードの開始負荷設定値 (2) <NRf> 動作モードの終了負荷設定値 (3) <NR1> 開始ステップ番号の設定 (4) <NR1> 終了ステップ番号の設定
応答	(1)<ASCII string> 動作モードの開始負荷設定値を返します。 (2)<ASCII string> 動作モードの終了負荷設定値を返します。 (3)<ASCII string> 開始ステップ番号を返します。 (4)<ASCII string> 終了ステップ番号を返します。
例	:FSEQ:FILL 0,5,1,6 ファストシーケンスのプログラムに FILL の条件を設定します。
クエリ 例	:FSEQ:FILL? >Start Value:0.00, End Value:5.00, Start Step:1, End Step:6 ファストシーケンスの FILL の設定値を返します。

Set →

#### 4-17-7. :FSEquence:LAST

→ Query

説明	ファストシーケンスの終了後の負荷値の設定とクエリ
構文	:FSEquence:LAST <NRf>

クエリ構文	:FSEquence:LAST?	
パラメーター	<NRf>	ステップ終了後の負荷設定値 (負荷条件 ON で有効になります。)
応答	ステップ終了後の負荷値を"Last:<NR2>"で返します。	
例	:FSEQ:LAST1 ステップ終了後の負荷値を設定します。	
クエリ 例	:FSEQ:LAST? >Last:0.070000 ステップ終了後の負荷値を返します。	
関連コマンド	:FSEquence:LLOad	

Set →  
 → Query

#### 4-17-8. :FSEquence:LLOad

説明	ファストシーケンスの終了後の負荷設定の有効・無効の設定とクエリ	
構文	:FSEquence:LLOad {ON   OFF}	
クエリ構文	:FSEquence:LLOad?	
パラメーター	ON OFF	ステップ終了後の負荷設定は有効 ステップ終了後の負荷設定は無効
応答	ステップ終了後の負荷設定を"Last Load:{ON   OFF}"で返します。	
例	:FSEQ:LLO ON ステップ終了後の負荷設定を有効にします。	
クエリ 例	:FSEQ:LLO? >Last Load:OFF ステップ終了後の負荷設定の有効・無効を返します。	
関連コマンド	:FSEquence:LAST	

Set →  
 → Query

#### 4-17-9. :FSEquence:LOOP

説明	ファストシーケンスのループ回数の設定とクエリ	
構文	:FSEquence:LOOP {<NR1>   INFinity }	
クエリ構文	:FSEquence:LOOP?	
パラメーター	<NR1> INFinity	ループ回数設定 (1~9999) 無限回数
応答	ファストシーケンスのループ回数を"Loop:{<NR1>   Infinity}"で返します。	
例	:FSEQ:LOOP 1 ファストシーケンスのループ回数を1に設定します。	

クエリ 例 :FSEQ:LOOP?  
>Loop:Infinity  
ファストシーケンスのループ回数を返します。

Set →

→ Query

#### 4-17-10. :FSEquence:MEMO

説明	ファストシーケンスのメモの設定とクエリ
構文	:FSEquence:MEMO <ASCII string>
クエリ構文	:FSEquence:MEMO?
パラメーター	<ASCII string> 12文字までのメモを設定 ダブルコートで文字列を括ります。
応答	ファストシーケンスのメモを"Memo:<ASCII string>"で返します。
例	:FSEQ:MEMO "ABC" ファストシーケンスのメモを設定します。
クエリ 例	:FSEQ:MEMO? >Memo: ABCD ファストシーケンスのメモを返します。

Set →

→ Query

#### 4-17-11. :FSEquence:MODE

説明	ファストシーケンスの動作モードの設定とクエリ
構文	:FSEquence:MODE {CC   CR}
クエリ構文	:FSEquence:MODE?
パラメーター	CC 定電流 モードに設定 CR 定抵抗 モードに設定
応答	ファストシーケンスの動作モードを"Mode:{CC   CR}"で返します。
例	:FSEQ:MODE CC ファストシーケンスの動作モードを設定します。
クエリ 例	:FSEQ:MODE? >Mode:CC ファストシーケンスの動作モードを返します。

Set →

→ Query

#### 4-17-12. :FSEquence:RANGe

説明	ファストシーケンスの動作レンジの設定とクエリ
構文	:FSEquence:RANGe { IHVH   IMVH   ILVH   IHVL   IMVL   ILVL }
クエリ構文	:FSEquence:RANGe?

パラメーター	IHVH IMVH ILVH IHVL IMVL ILVL	電流 High, 電圧 High レンジ設定 電流 Middle, 電圧 High レンジ設定 電流 Low, 電圧 High レンジ設定 電流 High, 電圧 Low レンジ設定 電流 Middle, 電圧 Low レンジ設定 電流 Low, 電圧 Low レンジ設定
応答	設定されている動作レンジを"Range:{IHVH   IMVH   ILVH   IHVL   IMVL   ILVL}"で返します。	
例	:FSEQ:RANG IHVL 電流 High, 電圧 Low の動作レンジを設定します。	
クエリ 例	:FSEQ:RANG? >Range:IHVH 設定されている動作レンジを返します。	

Set →

→ Query

#### 4-17-13. :FSEquence:RPTStep

説明	ファストシーケンスのループ最後のステップ番号の設定とクエリ	
構文	:FSEquence:RPTStep <NR1>	
クエリ構文	:FSEquence:RPTStep?	
パラメーター	<NR1>	ループ最後のステップ番号設定
応答	ループ最後のステップ番号を"RPTSTEP:<NR1>"で返します。	
例	:FSEQ:RPTS 1 ループ最後のステップ番号設定します。	
クエリ 例	:FSEQ:RPTS? >RPTSTEP:1 ループ最後のステップ番号を返します。	

Set →

#### 4-17-14. :FSEquence:SAVE

説明	ファストシーケンスのプログラム保存	
構文	:FSEquence:SAVE	
例	:FSEQ:SAVE ファストシーケンスのプログラム保存します。	

Set →

#### 4-17-15. :FSEquence:TBASe

→ Query

説明	ファストシーケンスのタイムベースの設定とクエリ	
構文	:FSEquence:TBASe <NRf>	
クエリ構文	:FSEquence:TBASe?	

パラメーター	<NRf>	タイムベースを設定 (単位は秒)
応答	ファストシーケンスのタイムベースを"Time Base:<NR2>"で返します。	
例	:FSEQ:TBAS 0.6 ファストシーケンスのタイムベースを 0.6 秒に設定します。	
クエリ 例	:FSEQ:TBAS? >Time Base:0.60000 ファストシーケンスのタイムベースを返します。	

Set →  
 → Query

#### 4-17-16. :FSEquence:STATe

説明	ファストシーケンスモードの選択と動作中の設定とクエリ	
構文	:FSEquence:STATe {OFF ON}	
クエリ構文	:FSEquence:STATe?	
パラメーター	OFF ON	ファストシーケンスモードのオフ設定 ファストシーケンスモードのオン設定
応答	ファストシーケンスモードの状態を返します。 ファストシーケンスモードがオンの時は"ON,{ STOP   RUN}"で返します。 (“STOP”:停止中、“RUN”:実行中) ファストシーケンスモードがオフの時は"OFF"を返します。	
例	:FSEQ:STAT ON ファストシーケンスモードに設定します。	
クエリ 例	:FSEQ:STAT? >ON,STOP ファストシーケンスモードの状態を返します。	

#### 4-18. OCP テスト コマンド

テストの実行・停止は:INPUT コマンドを使用します。4-3-1. :INPut を参照してください。

Set →  
 → Query

#### 4-18-1. :OCP:STATe

説明	OCP テスト動作を設定します。	
構文	:OCP:STATe {<bool> OFF ON}	
クエリ構文	:OCP:STATe?	
パラメーター	ON / 1 OFF / 0	OCP テスト動作をオンします。 OCP テスト動作をオフします。
応答パラメータ	ON OFF	OCP テスト動作はオンです。 OCP テスト動作はオフです。
応答	OCP テスト動作を応答します。	

例 :OCP:STATe ON  
OCP テスト動作をオンします。

クエリ 例 :OCP:STATe?  
OFF  
OCP テスト動作はオフです。

Set →  
→ Query

#### 4-18-2. :OCP:EDIT[:CHANnel]

説明

構文 :OCP:EDIT[:CHANnel]{<NR1>,LOW|MIDDLE|HIGH,<NR2>,<NR2>,<NR2>,<NR2>,<NR2>,<NR2>}

パラメーター	<NR1>	テストパターン番号を指定します。
	LOW MIDDLE HIGH	電流レンジを指定します。
	<NR2>	開始電流値[A]を指定します。
	<NR2>	終了電流値[A]を指定します。
	<NR2>	電流ステップ[A]を指定します。
	<NR2>	1 ステップの秒数を指定します。
	<NR2>	開始遅延時間を指定します。
	<NR2>	トリガ電圧[V]を指定します。
	<NR2>	テスト完了後の電流値を指定します。

例 :OCP:EDIT 1, LOW, 0.5, 5.0, 0.1, 0.25, 0, 5.0, 1.0

クエリ構文 :OCP:EDIT?

応答パラメーター	<string>	Temp:<Seq_1~Seq_12>,Range:<High Middle Low>, Start C:<NR2>, End C:<NR2>, Step C:<NR2>, Step T:<NR2>, Delay:<NR2>, TrigV :<NR2>, Last C:<NR2>
----------	----------	--

クエリ 例 :OCP:EDIT?  
Temp:Seq\_1, Range:High, Start C:0.36749, EndC:  
0.36750, Step C:0.00001, Step T:5.00, Delay:0.00,Trig  
V:0.01, Last C:0.00000

Set →  
→ Query

#### 4-18-3. :OCP[:CHANnel]:NUMBer

説明 OCP テスト番号を指定します。

構文 :OCP[:CHANnel]:NUMBer <NR1>

クエリ構文 :OCP[:CHANnel]:NUMBer?

パラメーター <NR1> OCP テスト番号を指定します。

応答 OCP テスト番号を応答します。

例 :OCP:NUMB 1  
OCP テストを1にします。

クエリ 例 :OCP:NUMB?  
1  
OCP テスト番号は 1 です。

Set →  
→ Query

#### 4-18-4. :OCP[:CHANnel]:RANGe

説明 OCP テストの電流レンジを指定します。  
構文 :OCP[:CHANnel]:RANGe {LOW|MIDDLE|HIGH}

クエリ構文 :OCP[:CHANnel]:RANGe?

パラメーター LOW 電流レンジを LOW にします  
MIDDLE 電流レンジを MID にします  
HIGH 電流レンジを HIGH にします

応答 OCP テストの電流レンジを応答します。

例 :OCP:RANG LOW  
OCP テストの電流レンジを LOW にします。

クエリ 例 :OCP:RANG?  
Low  
OCP テストの電流レンジは LOW です。

Set →  
→ Query

#### 4-18-5. :OCP[:CHANnel]:STARt

説明 OCP テストの開始電流を指定します。

構文 :OCP[:CHANnel]:STARt {<NR2>}

クエリ構文 :OCP[:CHANnel]:STARt?

パラメーター <NR2> 開始電流[A]を指定します

応答 OCP テストの開始電流を応答します。

例 :OCP:STAR 2  
開始電流を 2A に設定します。

クエリ 例 :OCP:STAR?  
0.1000  
開始電流は 0.1A です。

Set →  
→ Query

#### 4-18-6. :OCP[:CHANnel]:END

説明 OCP テストの終了電流を指定します。

構文 :OCP[:CHANnel]:END {<NR2>}

クエリ構文 :OCP[:CHANnel]:END?

パラメーター	<NR2> 終了電流[A]を指定します
応答	OCP テストの終了電流を応答します。
例	:OCP:END 2 終了電流を 2A に設定します。
クエリ 例	:OCP:END? 0.1000 終了電流は 0.1A です。

Set →

→ Query

#### 4-18-7. :OCP[:CHANnel]:STEP:CURRent

説明	OCP テストの増分電流を指定します。
構文	:OCP[:CHANnel]: STEP:CURRent {<NR2>}
クエリ構文	:OCP[:CHANnel]: STEP:CURRent?
パラメーター	<NR2> 増分電流[A]を指定します
応答	OCP テストの増分電流を応答します。
例	:OCP:STEP:CURR 0.1 増分電流を 0.1A に設定します。
クエリ 例	:OCP:STEP:CURR? 0.1000 増分電流は 0.1A です。

Set →

→ Query

#### 4-18-8. :OCP[:CHANnel]:LAST

説明	OCP テスト終了後の電流を指定します。
構文	:OCP[:CHANnel]:LAST {<NR2>}
クエリ構文	:OCP[:CHANnel]:LAST?
パラメーター	<NR2> 終了後電流[A]を指定します
応答	OCP テストの終了後電流を応答します。
例	:OCP:LAST 2 終了後電流を 2A に設定します。
クエリ 例	:OCP:LAST? 0.1000 終了後電流は 0.1A です。

Set →

→ Query

#### 4-18-9. :OCP[:CHANnel]:STEP:TIME

説明	OCP テストの 1 ステップの時間を指定します。
構文	:OCP[:CHANnel]:STEP:TIME {<NR2>}
クエリ構文	:OCP[:CHANnel]:STEP:TIME?

パラメーター	<NR2>	1 ステップの秒数を指定します
応答	OCP テストの 1 ステップの時間を応答します。	
例	:OCP:STEP:TIME 2 1 ステップの時間を 2 秒に設定します。	
クエリ 例	:OCP:STEP:TIME? 0.10 1 ステップは 0.1 秒です。	

Set →

→ Query

#### 4-18-10. :OCP[:CHANnel]:DELay

説明	OCP テストのロードオンまたは電圧印加から開始電流の設定までの遅延時間を指定します。	
構文	:OCP[:CHANnel]:DELay {<NR2>}	
クエリ構文	:OCP[:CHANnel]:DELay:TIME?	
パラメーター	<NR2>	遅延時間の秒数を指定します
応答	OCP テストの遅延時間を応答します。	
例	:OCP:DELay 2 遅延時間を 2 秒に設定します。	
クエリ 例	:OCP:DELay? 0.10 遅延時間は 0.1 秒です。	

Set →

→ Query

#### 4-18-11. :OCP[:CHANnel]:TRIGger

説明	OCP 判定となる電圧ドロップのしきい値(トリガ電圧)を設定します。	
構文	:OCP[:CHANnel]:TRIGger {<NR2>}	
クエリ構文	:OCP[:CHANnel]:TRIGger?	
パラメーター	<NR2>	OCP 判定のトリガ電圧を設定します。
応答	OCP テストの OCP 判定電圧を応答します。	
例	:OCP:TRIG 2 OCP 判定電圧を 2V に設定します。	
クエリ 例	:OCP:TRIG? 2.0 OCP 判定電圧は 2V です。	

#### 4-18-12. :OCP:CHANnel:STATus

→ Query

説明	OCP テストの状態を応答します。	
クエリ構文	:OCP:CHANnel:STATus?	

応答パラメーター	0	テスト終了
	1	テスト中

クエリ 例 :OCP:CHAN:STAT?  
0  
OCP テスト終了

#### 4-18-13. :OCP:RESult

→(Query)

説明 OCP テストの結果を応答します。  
クエリ構文 :OCP:RESult?  
応答パラメーター <NR2>,<NR2> OCP 電流、OCP 電圧  
クエリ 例 :OCP:RES?  
3.6750,0.10  
OCP 電流:3.675A、OCP 電圧:0.10V

#### 4-18-14. :OCP:SAVE

(Set)→

説明 OCP テストの設定値を記憶します。  
構文 :OCP:SAVE

#### 4-18-15. :OCP:RUN

(Set)→

説明 OCP テストを開始します。  
構文 :OCP:RUN

### 4-19. ユーティリティ・コマンド

#### 4-19-1. :UTILity:ALARm

(Set)→

→(Query)

説明 アラーム時のスピーカ音の設定とクエリ  
構文 :UTILity:ALARm {< Boolean > | ON | OFF}  
クエリ構文 :UTILity:ALARm?  
パラメーター ON / 1 アラーム時にスピーカ音はオン  
OFF / 0 アラーム時にスピーカ音はオフ  
応答 アラーム時のスピーカ音の設定を"{ON | OFF}"で返します。  
例 :UTIL:ALAR ON  
アラーム時のスピーカ音を設定します。  
クエリ 例 :UTIL:ALAR?  
>On  
アラーム時のスピーカ音の設定を返します。

Set →

→ Query

#### 4-19-2. :UTILity:BRiGhtness

説明	LCD ディスプレイの輝度の設定とクエリ
構文	:UTILity:BRiGhtness <NR1>
クエリ構文	:UTILity:BRiGhtness?
パラメーター/応答	<NR1> 輝度の設定
例	:UTIL:BRi 70 輝度を設定します。
クエリ 例	:UTIL:BRi? >70 輝度の設定値を返します。

Set →

→ Query

#### 4-19-3. :UTILity:CONTRast

説明	LCD ディスプレイのコントラストの設定とクエリ
構文	:UTILity:CONTRast <NR1>
クエリ構文	:UTILity:CONTRast?
パラメーター/応答	<NR1> コントラストの設定
例	:UTIL:CONTR 8 コントラストを設定します。
クエリ 例	:UTIL:CONTR? >8 コントラストの設定値を返します。

Set →

→ Query

#### 4-19-4. :UTILity:GNG

説明	Go-NoGo 判定時のスピーカ音の設定とクエリ
構文	:UTILity:GNG {< Boolean >   ON   OFF}
クエリ構文	:UTILity:GNG?
パラメーター	ON / 1 Go-NoGo 判定時のスピーカ音をオンに設定 OFF / 0 Go-NoGo 判定時のスピーカ音をオフに設定
応答	Go-NoGo 判定時のスピーカ音の設定を{On   Off}で返します。
例	:UTIL:GNG ON Go-NoGo 判定時のスピーカ音を設定します。
クエリ 例	:UTIL:GNG? >On Go-NoGo 判定時のスピーカ音を返します。

Set →

→ Query

## 4-19-5. :UTILITY:KNOB

説明	つまみの操作設定の設定とクエリ	
構文	:UTILITY:KNOB { UPDated   OLD }	
クエリ構文	:UTILITY:KNOB?	
パラメーター	UPDated OLD	リアルタイムで更新設定 Enter 確定後に更新設定
応答	つまみの操作設定を{Updated   Old}で返します。	
例	:UTIL:KNOB UPD つまみの操作を設定します。	
クエリ 例	:UTIL:KNOB? >Updated つまみの操作設定を返します。	

Set →

→ Query

## 4-19-6. :UTILITY:LANGUage

説明	操作パネルの言語の設定とクエリ 補足)本器は言語は英語のみです。	
構文	:UTILITY:LANGUage ENGLish	
クエリ構文	:UTILITY:LANGUage?	
パラメーター	ENGLish	英語に設定します。
応答	言語を"English"で返します。	
例	:UTIL:LANG ENGL 言語を設定します。	
クエリ 例	:UTIL:LANG? >English 言語を返します。	

Set →

→ Query

## 4-19-7. :UTILITY:LOAD

説明	電源投入時の負荷の動作と動作モードの設定とクエリ	
構文	:UTILITY:LOAD {ON   OFF   LOAD   PROG   NSEQ   FSEQ}	
クエリ構文	:UTILITY:LOAD?	
パラメーター	ON OFF LOAD PROG NSEQ FSEQ	負荷の動作をオンに設定 負荷の動作をオフに設定 通常動作モードに設定 プログラム動作モードに設定 ノーマルシーケンス動作モードに設定 ファストシーケンス動作モードに設定

応答	電源投入時の負荷動作を”Load:{On   Off}, Load On:{Load   Prog   NSeq   FSeq}”で返します。
例 1	:UTIL:LOAD ON 電源投入時の負荷動作をオンに設定します。
例 2	:UTIL:LOAD PROG 電源投入時の動作モードをプログラムモードに設定します。
クエリ 例	:UTIL:LOAD? >Load:Off, Load On:Prog 負荷動作と動作モードを返します。

Set →

→ Query

#### 4-19-8. :UTILity:LOAD:MODE

説明	モード切り替え時に出力をオフにする
構文	:UTILity:LOAD:MODE {< Boolean >   ON   OFF}
クエリ構文	:UTILity:LOAD:MODE?
パラメーター	ON / 1      オン設定 OFF / 0      オフ設定
応答	モード切り替え時の出力設定を On, Off で返す。
例	:UTIL:LOAD:MODE ON モード切り替え時に出力をオフにする
クエリ 例	:UTIL:LOAD:MODE? >On モード切り替え時の出力設定を返します。

Set →

→ Query

#### 4-19-9. :UTILity:LOAD:RANGe

説明	レンジ切り替え時に出力をオフにする
構文	:UTILity:LOAD:RANGe {< Boolean >   ON   OFF}
クエリ構文	:UTILity:LOAD:RANGe?
パラメーター	ON / 1      オン設定 OFF / 0      オフ設定
応答	レンジ切り替え時の出力設定を On, Off で返す。
例	:UTIL:LOAD:RANG ON モード切り替え時に出力をオフにする
クエリ 例	:UTIL:LOAD:RANG? >On レンジ切り替え時の出力設定を返します。

## 4-19-10. :UTILity:REMOte

Set →

説明	リモート状態の解除
構文	:UTILity:REMOte {< Boolean >   ON   OFF}
パラメーター	ON / 1 オン設定 OFF / 0 オフ設定
例	:UTIL:REM ON リモート状態に設定します。

## 4-19-11. :UTILity:REMOte:MODE

Set →

説明	画面表示を切り替えデータ更新時間の短縮を図る。
構文	:UTILity:REMOte:MODE {< Boolean >   NORMal   FAST}
パラメーター	FAST / 1 設定画面表示を禁止し高速応答設定 NORMal / 0 通常の設定画面表示
例	:UTIL:REM:MODE FAST 高速応答画面に設定します。

Set →

## 4-19-12. :UTILity:SPEAker

→ Query

説明	キー入力とスクロール時のスピーカ音の設定とクエリ
構文	:UTILity:SPEAker {< Boolean >   ON   OFF}
クエリ構文	:UTILity: SPEAker?
パラメーター	ON / 1 キー入力とスクロール時のスピーカ音をオンに設定 OFF / 0 キー入力とスクロール時のスピーカ音をオフに設定
応答	キー入力とスクロール時のスピーカ音の設定を"{On   Off}"で返します。
例	:UTIL:SPEA ON キー入力とスクロール時のスピーカ音を設定します。
クエリ 例	:UTIL:SPEA? >On キー入力とスクロール時のスピーカ音を返します。

## 4-19-13. :UTILity:SYStem

→ Query

説明	モデル番号、シリアル番号、ファームウェアバージョンのクエリ
クエリ構文	:UTILity:SYStem?
応答	<ASCII string> モデル名を返します。

<NR1> シリアル番号を返します。  
 <ASCII string> ファームウェアのバージョンを返します。

クエリ 例 :UTIL:SYS?  
 >LSG-175,12345678,V1.01.001  
 モデル番号、シリアル番号、ファームウェアバージョンを返します。

Set →

4-19-14. :UTILity:TiMe

→ Query

説明	日付と時刻の設定とクエリ	
構文	:UTILity:TiMe (1),(2),(3),(4),(5)	
クエリ構文	:UTILity:TiMe?	
パラメーター	(1) <NR1>	月の設定
	(2) <NR1>	日の設定
	(3) <NR1>	年の設定
	(4) <NR1>	時の設定
	(5) <NR1>	分の設定
応答	日付と時刻を”{Month:<NR1>, Day:<NR1>, Year:<NR1>, Hour:<NR1>, Minute:<NR1>}”で返します。	
例	:UTIL:TIM 9,1,2013,10,11 日付と時刻を設定します。	
クエリ 例	:UTIL:TIM? >Month:9, Day:1, Year:2013, Hour:10, Minute:11 日付と時刻を返信します。	

Set →

4-19-15. :UTILity:UNReg

→ Query

説明	アンレギュレーション時のスピーカ音の設定とクエリ	
構文	:UTILity:UNReg {< Boolean >   ON   OFF}	
クエリ構文	:UTILity:UNReg?	
パラメーター	ON / 1	アンレギュレーション時のスピーカ音をオンに設定
	OFF / 0	アンレギュレーション時のスピーカ音をオフに設定
応答	アンレギュレーション時のスピーカ音の設定を”{On   Off}”で返します。	
例	:UTIL:UNR ON アンレギュレーション時のスピーカ音を設定します。	

クエリ 例 :UTIL:UNR?  
 >On  
 アンレギュレーション時のスピーカ音を返します。

#### 4-20. インターフェース・コマンド

Set →

##### 4-20-1. :UTILity:BRATe

→ Query

説明	RS-232C のボーレートの設定とクエリ 注)コマンドは RS-232C インターフェイス設定のみ有効です。	
構文	:UTILity:BRATe {2400   4800   9600   19200   38400}	
クエリ構文	:UTILity:BRATe?	
パラメーター/応答	<NR1>	ボーレートの設定
例	:UTIL:BRAT 38400 ボーレートを設定します。	
クエリ 例	:UTIL:BRAT? >38400 ボーレートを返します。	

Set →

##### 4-20-2. :UTILity:INTerface

→ Query

説明	インターフェイスの設定とクエリ	
構文	:UTILity:INTerface {USB   RS232} 注)コマンドは USB と RS232 制御時のみ有効です。	
クエリ構文	:UTILity:INTerface?	
パラメーター	送信後に設定されますが、機能を有効にするには電源再投入が必要です。	
	USB	USB の設定
	RS232	RS-232C の設定
例	:UTIL:INT RS232 RS-232C インターフェイスを設定します。	
クエリ 例	:UTIL:INT? >RS232 インターフェイスを返します。	

Set →

→ Query

### 4-20-3. :UTILity:PARity

説明	RS-232C インターフェイスのパリティビットの設定とクエリ 注)コマンドは RS-232C インターフェイス設定のみ有効です。	
構文	:UTILity:PARity { NONE   ODD   EVEN}	
クエリ構文	:UTILity:PARity?	
パラメーター	NONE	パリティ無し設定
	ODD	奇数パリティ設定
	EVEN	偶数パリティ設定
応答	パリティの設定値を{None   Odd   Even}で返します。	
例	:UTIL:PAR NONE パリティ無しに設定します。	
クエリ 例	:UTIL:PAR? >None パリティの設定を返します。	

Set →

→ Query

### 4-20-4. :UTILity:SBIT

説明	RS-232C インターフェイスのストップビットの設定とクエリ 注)コマンドは RS-232C インターフェイス設定のみ有効です。	
構文	:UTILity:SBIT {1   2}	
クエリ構文	:UTILity:SBIT?	
パラメーター/応答	<NR1>	ストップビットを設定します。
例	:UTIL:SBIT 1 ストップビットを設定します。	
クエリ 例	:UTIL:SBIT? >1 ストップビットを返します。	

## 4-21. ファイル・コマンド

### 4-21-1. :FACTory[:RECall]

Set →

説明	工場出荷時のデフォルト設定 注) インターフェースは次の様に切り替わります。インターフェース RS-232C, ボーレート 38400, ストップビット 1, パリティ 無し
構文	:FACTory[:RECall]
例	:FACT 工場出荷時のデフォルトに設定します。

### 4-21-2. :MEMory:RECall

Set →

説明	内部メモリからの読み出し設定
構文	:MEMory:RECall <NR1>
パラメーター	<NR1> 内部メモリ番号の指定 (1~256)
例	:MEM:REC 20 内部メモリ 20 の読み出して設定します。
同一機能コマンド	*RCL

### 4-21-3. :MEMory:SAVE

Set →

説明	指定の内部メモリに保存
構文	:MEMory:SAVE <NR1>
パラメーター	<NR1> 内部メモリ番号の指定 (1~256)
例	:MEM:SAVE 20 内部メモリ 20 に現在の設定を保存します。
同一機能コマンド	*SAV

### 4-21-4. :PREset:RECall

Set →

説明	プリセットメモリからの読み出し設定
構文	:PREset:RECall <NR1>
パラメーター	<NR1> プリセットメモリ番号の指定 (0~9)
例	:PRE:REC 1 プリセットメモリ(P1)の読み出します。

## 4-21-5. :PREset:SAVE

Set →

説明	指定のプリセットメモリに保存
構文	:PREset:SAVE < NR1>
パラメーター	<NR1> プリセットメモリ番号の指定 (0~9)
例	:PRE: SAVE 1 プリセットメモリ(P1)に保存します。

## 4-21-6. :SETup:RECall

Set →

説明	セットアップデータからの読み出し設定
構文	:SETup:RECall <NR1>
パラメーター	<NR1> セットアップデータ番号の指定 (1~100)
例	:SET:REC 1 セットアップデータ(1)からの読み出します。

## 4-21-7. :SETup:SAVE

Set →

説明	指定のセットアップデータに保存
構文	:SETup:SAVE < NR1>
パラメーター	<NR1> セットアップデータ番号の指定 (1~100)
例	:SET: SAVE 1 セットアップデータ(1)に保存します。

## 4-21-8. :USER[:DEFault]:RECall

Set →

説明	ユーザーのデフォルト設定からの読み出し
構文	:USER[:DEFault]:RECall
例	:USER:REC ユーザーのデフォルト設定を読み出します。

## 4-21-9. :USER[:DEFault]:SAVE

Set →

説明	ユーザーのデフォルト設定に保存
構文	:USER[:DEFault]:SAVE
例	:USER:SAVE ユーザーのデフォルト設定として現在の設定値を保存します。

## 4-22. SCPI レジスタ・コマンド

### 4-22-1. :SYSTem:ERRor

→ Query

説明	エラークエリの間合せする。最後のエラーメッセージが戻ります。最大 32 のエラーがエラークエリに保存される。
クエリ構文	:SYSTem:ERRor?
応答	エラークエリを”<NR1>,<string>”で返します。
クエリ 例	:SYST:ERR? >-113, "Undefined header"

### 4-22-2. :STATus:PRESet

Set →

説明	Operation ステータスと Questionable ステータスと Csummary ステータスの初期値の設定です。PTR (正遷移) フィルターはセットされ、NTR (負遷移) フィルターとイネーブル・レジスタはリセットされます。																														
	<table border="1"> <thead> <tr> <th>初期値</th> <th>レジスタ/フィルター</th> <th>設定値</th> </tr> </thead> <tbody> <tr> <td>Csummary ステータス</td> <td>イネーブル</td> <td>0x0000</td> </tr> <tr> <td>Csummary ステータス</td> <td>PTR (正遷移)</td> <td>0x7FFF</td> </tr> <tr> <td>Csummary ステータス</td> <td>NTR (負遷移)</td> <td>0x0000</td> </tr> <tr> <td>Operation ステータス</td> <td>イネーブル</td> <td>0x0000</td> </tr> <tr> <td>Operation ステータス</td> <td>PTR (正遷移)</td> <td>0x7FFF</td> </tr> <tr> <td>Operation ステータス</td> <td>NTR (負遷移)</td> <td>0x0000</td> </tr> <tr> <td>QUEStionable ステータス</td> <td>イネーブル</td> <td>0x0000</td> </tr> <tr> <td>QUEStionable ステータス</td> <td>PTR (正遷移)</td> <td>0x7FFF</td> </tr> <tr> <td>QUEStionable ステータス</td> <td>NTR (負遷移)</td> <td>0x0000</td> </tr> </tbody> </table>	初期値	レジスタ/フィルター	設定値	Csummary ステータス	イネーブル	0x0000	Csummary ステータス	PTR (正遷移)	0x7FFF	Csummary ステータス	NTR (負遷移)	0x0000	Operation ステータス	イネーブル	0x0000	Operation ステータス	PTR (正遷移)	0x7FFF	Operation ステータス	NTR (負遷移)	0x0000	QUEStionable ステータス	イネーブル	0x0000	QUEStionable ステータス	PTR (正遷移)	0x7FFF	QUEStionable ステータス	NTR (負遷移)	0x0000
初期値	レジスタ/フィルター	設定値																													
Csummary ステータス	イネーブル	0x0000																													
Csummary ステータス	PTR (正遷移)	0x7FFF																													
Csummary ステータス	NTR (負遷移)	0x0000																													
Operation ステータス	イネーブル	0x0000																													
Operation ステータス	PTR (正遷移)	0x7FFF																													
Operation ステータス	NTR (負遷移)	0x0000																													
QUEStionable ステータス	イネーブル	0x0000																													
QUEStionable ステータス	PTR (正遷移)	0x7FFF																													
QUEStionable ステータス	NTR (負遷移)	0x0000																													
構文	:STATus:PRESet																														
例	:STAT:PRES Operation ステータスと Questionable ステータスと Csummary ステータスの初期値に設定します。																														

## 4-23. Csummary ステータス・コマンド

### 4-23-1. :STATus:CSUMmary:CONDition

→ Query

説明	Csummary のコンディションレジスタのクエリ
クエリ構文	:STATus:CSUMmary:CONDition?
応答	Csummary のコンディションレジスタを"<NR1>"で返します。
クエリ 例	:STAT:CSUM:COND? >1 Csummary のコンディションレジスタを返します。

Set →

### 4-23-2. :STATus:CSUMmary:ENABLE

→ Query

説明	Csummary のイベントイネーブル・レジスタの設定とクエリ
構文	:STATus:CSUMmary:ENABLE <NR1>
クエリ構文	:STATus:CSUMmary:ENABLE?
パラメーター/応答	<NR1> Csummary のイベントイネーブル・レジスタの設定
例	:STAT:CSUM:ENAB 1 Csummary のイベントイネーブル・レジスタを設定します。
クエリ 例	:STAT:CSUM:ENAB? >1 Csummary のイベントイネーブル・レジスタを返します。

### 4-23-3. :STATus:CSUMmary[:EVENT]

→ Query

説明	Csummary のイベントレジスタのクエリ
クエリ構文	:STATus:CSUMmary[:EVENT]?
応答	Csummary のイベントレジスタを"<NR1>"で返します。
クエリ 例	:STAT:CSUM? >1 Csummary のイベントレジスタを返します。

Set →

→ Query

#### 4-23-4. :STATus:CSUMmary:NTRansition

説明	Csummary ステータスが正から負に変わる検出ビットの設定とクエリ
構文	:STATus:CSUMmary:NTRansition <NR1>
クエリ構文	:STATus:CSUMmary:NTRansition?
パラメーター/応答	<NR1> Csummary ステータスが正から負に変わる検出ビットの設定
例	:STAT:CSUM:NTR 1 Csummary ステータスが正から負に変わる検出ビットを設定します。
クエリ 例	:STAT:CSUM:NTR? >1 Csummary ステータスが正から負に変わる検出ビットの設定を返します。

Set →

→ Query

#### 4-23-5. :STATus:CSUMmary:PTRansition

説明	Csummary ステータスが負から正に変わる検出ビットの設定とクエリ
構文	:STATus:CSUMmary:PTRansition <NR1>
クエリ構文	:STATus:CSUMmary:PTRansition?
パラメーター/応答	<NR1> Csummary ステータスが負から正に変わる検出ビットの設定
例	:STAT:CSUM:PTR 1 Csummary ステータスが負から正に変わる検出ビットを設定します。
クエリ 例	:STAT:CSUM:PTR? >1 Csummary ステータスが負から正に変わる検出ビットの設定を返します。

## 4-24. Operation ステータス・コマンド

### 4-24-1. :STATus:OPERation:CONDition

→ Query

説明	Operation のコンディションレジスタのクエリ
クエリ構文	:STATus:OPERation:CONDition?
応答	Operation のコンディションレジスタを"<NR1>"で返します。
クエリ 例	: STAT:OPER:COND? >1 Operation のコンディションレジスタを返します。

Set →

### 4-24-2. :STATus:OPERation:ENABLE

→ Query

説明	Operation のイベントイネーブル・レジスタの設定とクエリ
構文	:STATus:OPERation:ENABLE <NR1>
クエリ構文	:STATus:OPERation:ENABLE?
パラメーター/応答	<NR1>      Operation のイベントイネーブル・レジスタの設定
例	:STAT:OPER:ENAB 1 Operation のイベントイネーブル・レジスタを設定します。
クエリ 例	:STAT:OPER:ENAB? >1 Operation のイベントイネーブル・レジスタを返します。

### 4-24-3. :STATus:OPERation[:EVENT]

→ Query

説明	Operation のイベントレジスタのクエリ
クエリ構文	:STATus:OPERation[:EVENT]?
応答	Operation のイベントレジスタを"<NR1>"で返します。
クエリ 例	:STAT:OPER? >1 Operation のイベントレジスタを返します。

Set →

→ Query

## 4-24-4. :STATus:OPERation:NTRansition

説明	Operation ステータスが正から負に変わる検出ビットの設定とクエリ
構文	:STATus:OPERation:NTRansition <NR1>
クエリ構文	:STATus:OPERation:NTRansition?
パラメーター/応答	<NR1> Operation ステータスが正から負に変わる検出ビットの設定
例	:STAT:OPER:NTR 1 Operation ステータスが正から負に変わる検出ビットを設定します。
クエリ 例	:STAT:OPER:NTR? >1 Operation ステータスが正から負に変わる検出ビットの設定を返します。

Set →

→ Query

## 4-24-5. :STATus:OPERation:PTRansition

説明	Operation ステータスが負から正に変わる検出ビットの設定とクエリ
構文	:STATus:OPERation:PTRansition <NR1>
クエリ構文	:STATus:OPERation:PTRansition?
パラメーター/応答	<NR1> Operation ステータスが負から正に変わる検出ビットの設定
例	:STAT:OPER:PTR 1 Operation ステータスが負から正に変わる検出ビットを設定します。
クエリ 例	:STAT:OPER:PTR? >1 Operation ステータスが負から正に変わる検出ビットの設定を返します。

## 4-25. Questionable ステータス・コマンド

### 4-25-1. :STATus:QUEStionable:CONDition

→ Query

説明	Questionable のコンディションレジスタのクエリ
クエリ構文	:STATus:QUEStionable:CONDition?
応答	Questionable のコンディションレジスタを"<NR1>"で返します。
クエリ 例	:STAT:QUES:COND? >1 Questionable のコンディションレジスタを返します。

Set →

### 4-25-2. :STATus:QUEStionable:ENABLE

→ Query

説明	Questionable ステータスのイベントイネーブルの設定とクエリ
構文	:STATus:QUEStionable:ENABLE <NR1>
クエリ構文	:STATus:QUEStionable:ENABLE?
パラメーター/応答	<NR1> Questionable のイベントイネーブル・レジスタの設定
例	:STAT:QUES:ENAB 1 Questionable のイベントイネーブル・レジスタを設定します。
クエリ 例	:STAT:QUES:ENAB? >1 Questionable のイベントイネーブル・レジスタを返します。

### 4-25-3. :STATus:QUEStionable[:EVENT]

→ Query

説明	Questionable のイベントレジスタのクエリ
クエリ構文	:STATus:QUEStionable [:EVENT]?
応答	Questionable のイベントレジスタを"<NR1>"で返します。
クエリ 例	:STAT:QUES? >1 Questionable のイベントレジスタを返します。

Set →

#### 4-25-4. :STATus:QUEStionable:NTRansition

→ Query

説明	Questionable ステータスが正から負に変わる検出ビットの設定とクエリ
構文	:STATus:QUEStionable:NTRansition <NR1>
クエリ構文	:STATus:QUEStionable:NTRansition?
パラメーター/応答	<NR1> Questionable ステータスが正から負に変わる検出ビットの設定
例	:STAT:QUES:NTR 1 Questionable ステータスが正から負に変わる検出ビットを設定します。
クエリ 例	:STAT:QUES:NTR? >1 Questionable ステータスが正から負に変わる検出ビットの設定を返します。

Set →

#### 4-25-5. :STATus:QUEStionable:PTRansition

→ Query

説明	Questionable ステータスが負から正に変わる検出ビットの設定とクエリ
構文	:STATus:QUEStionable:PTRansition <NR1>
クエリ構文	:STATus:QUEStionable:PTRansition?
パラメーター/応答	<NR1> Questionable ステータスが負から正に変わる検出ビットの設定
例	:STAT:QUES:PTR 1 Questionable ステータスが負から正に変わる検出ビットを設定します。
クエリ 例	:STAT:QUES:PTR? >1 Questionable ステータスが負から正に変わる検出ビットの設定を返します。

## 第5章 ステータス レジスタの概要

この章では、ステータス レジスタの構成および設定される条件に関する詳細説明です。

また、ステータス レジスタ理解することによって効果的なプログラムを作成することもできるようになります。

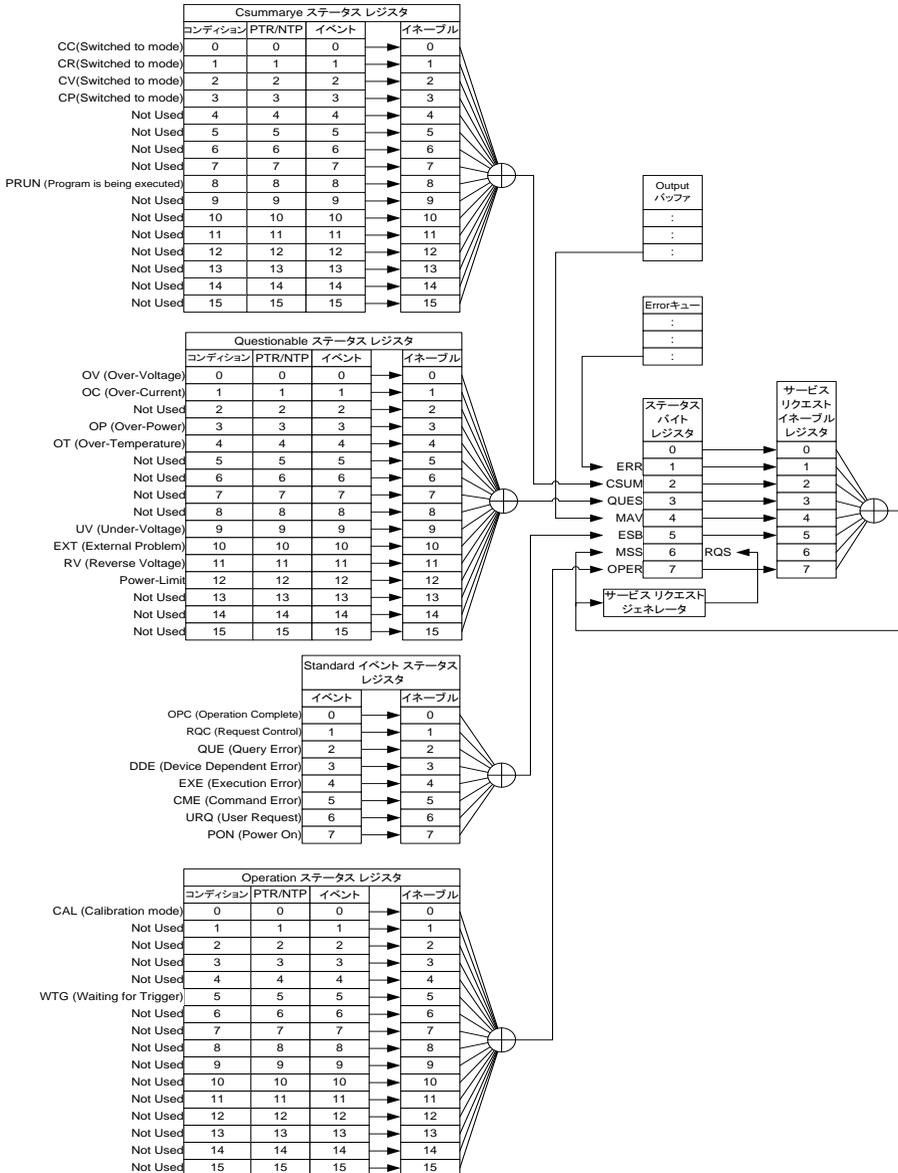
### 5-1. ステータス レジスタの紹介

---

**概要** ステータス レジスタは、本器の状態を知るのに用いられます。ステータス レジスタは、保護の状態、動作状態、セットのエラーの状態を保持します。LSG シリーズは、複数のレジスタグループを持っています。

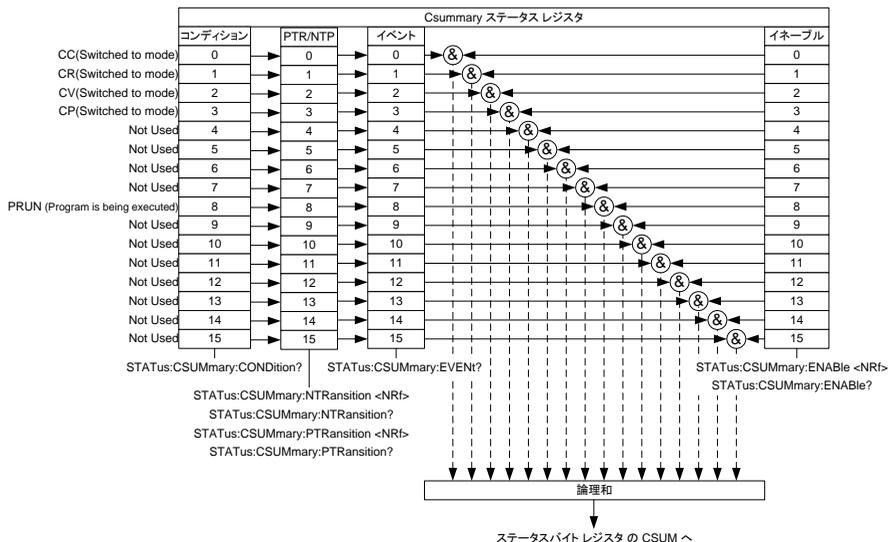
- Csummary ステータス レジスタ グループ
- Operation ステータス レジスタ グループ
- Questionable ステータス レジスタ グループ
- Standard イベント ステータス レジスタ グループ
- ステータス レジスタ グループ

## 5-2. ステータス レジスタの構成



### 5-3. Csummary ステータス レジスタ グループ

概要 Csummaryステータス レジスタ グループは、負荷モードとプログラムおよびシーケンスの動作状態を確認できます。



ビット	ビット名	説明	ビット	重み
概要	CC	定電流設定モードを示す	0	1
	CR	定抵抗設定モードを示す	1	2
	CV	定電圧設定モードを示す	2	4
	CP	定電力設定モードを示す	3	8
	PRUN	シーケンス動作モードを示す	8	256

**コンディションレジスタ** Csummary ステータスのコンディションレジスタは、負荷モードとプログラムまたはシーケンス動作モードの現在の状態を読み出せます。

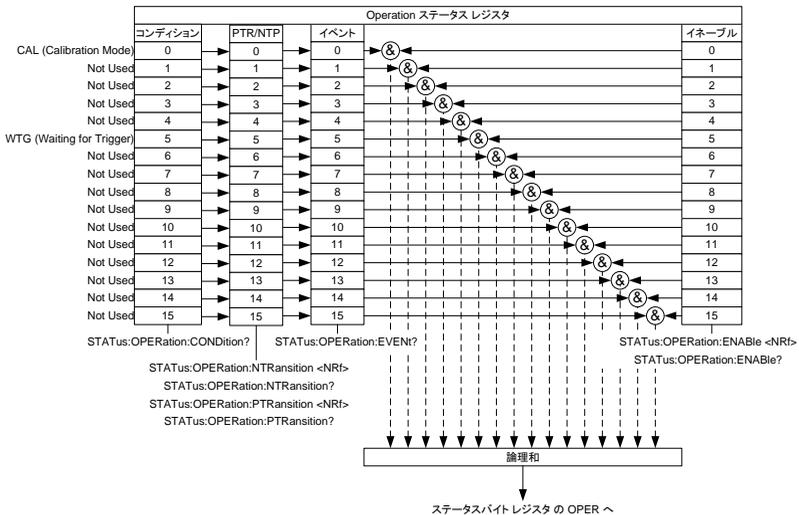
**PTR/NTR フィルター** PTR/NTR(正/負 遷移)レジスタは、コンディションレジスタのビットが変化した時にイベントレジスタに設定するビットを指定します。PTR フィルターは負から正に移行するイベントを検出する時に設定します。NTR フィルターは正から負に移行するイベントを検出する時に設定します。

PTRansition	正遷移	0→1
NTRansition	負遷移	1→0

イベントレジスタ	イベントレジスタは PTR/NTR フィルターで検出されたビットを保持します。また、イベントレジスタは内容が読み取られるかクリアされるまで検出したビットを保持します。
イネーブルレジスタ	イネーブルレジスタは、ステータスバイトレジスタの中の CSUM ビットを設定するイベントレジスタのビットを指定します。イネーブルレジスタが 0 の時には CSUM ビットは設定されません。

## 5-4. Operation ステータス レジスタ グループ

概要 Operation ステータス レジスタ グループは、校正モードとトリガー待ちの動作状態を確認できます。



ビット	ビット名	説明	ビット	重み
概要	CAL	校正モードを示す	0	1
	WTG	トリガ待ちを示す	5	32

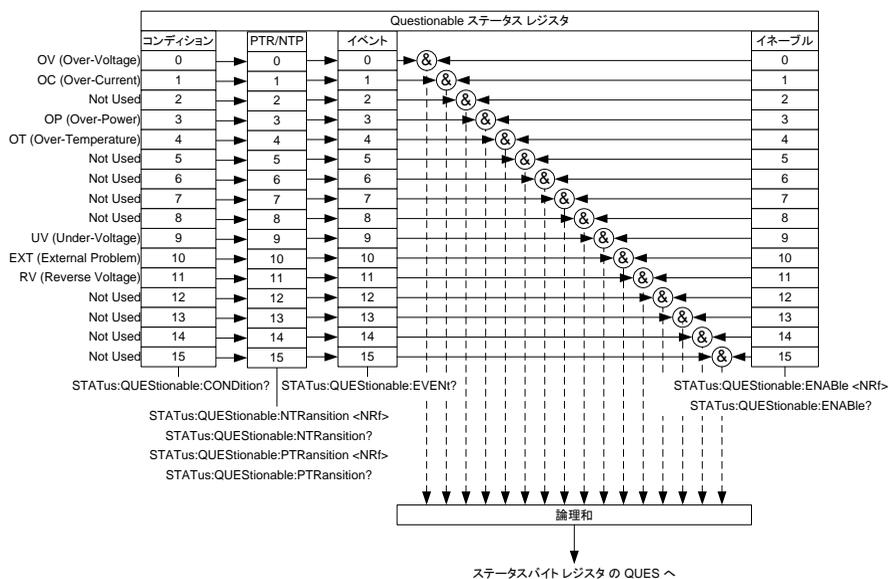
コンディションレジスタ Operation ステータスのコンディションレジスタは、校正モードとトリガー待ちの現在の状態を読み出せます。

PTR/NTR フィルター PTR/NTR (正/負 遷移) レジスタは、コンディションレジスタのビットが変化した時にイベントレジスタに設定するビットを指定します。PTR フィルターは負から正に移行するイベントを検出する時に設定します。NTR フィルターは正から負に移行するイベントを検出する時に設定します。

	PTRansition	正遷移	0→1
	NTRansition	負遷移	1→0
イベントレジスタ	イベントレジスタは PTR/NTR フィルターで検出されたビットを保持します。また、イベントレジスタは内容が読み取られるかクリアされるまで検出したビットを保持します。		
イネーブルレジスタ	イネーブルレジスタは、ステータスバイトレジスタの中の OPER ビットを設定するイベントレジスタのビットを指定します。イネーブルレジスタが 0 の時には OPER ビットは設定されません。		

## 5-5. Questionable ステータス レジスタ グループ

概要 Questionable ステータス レジスタ グループは、保護機能の動作状態を確認できます。



ビット	ビット名	説明	ビット	重み
概要	OV	過電圧状態を示す	0	1
	OC	過電流状態を示す	1	2
	OP	過電力状態を示す	3	8
	OT	過熱状態を示す	4	16
	UV	低電圧状態を示す	9	512
	EXT	外部制御の不具合状態を示す	10	1024
	RV	逆接続状態を示す	11	2048



ビット概要	ビット名	説明	ビット	重み
	OPC	すべての選ばれた未完了の動作が終了したとき、OPC ビットは設定されます。このビットは、*OPC コマンドに応じて設定されます。	0	1
	RQC	リクエストコントロール	1	2
	QUE	クエリエラー・ビットは、Output キューを読んだ時にデータがない場合に設定されます。	2	4
	DDE	デバイス規格/依存エラー	3	8
	EXE	実行エラービットは、コマンドの実行に失敗した時に設定されます。 違法なパラメータのコマンド 範囲外のパラメータ 無効なパラメータ	4	16
	CME	CME ビットは構文のエラーが発生した時に設定されます。また、<GET>コマンドがプログラムメッセージの中で受け取った場合も CME ビットは設定される。	5	32
	URQ	ユーザリクエスト	6	64
	PON	パワーオンになった時に設定されます。	7	128

---

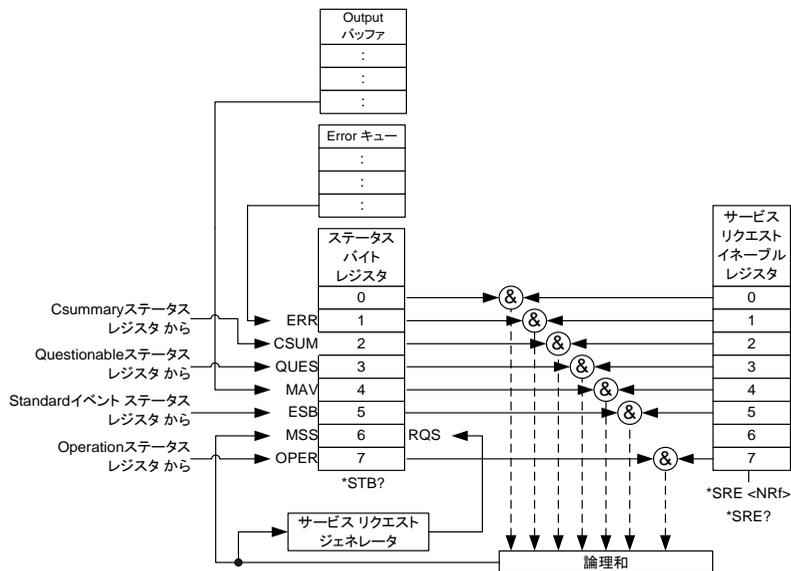
イベントレジスタ イベントレジスタは検出されたビットを保持します。また、イベントレジスタは内容が読み取られるかクリアされるまで検出したビットを保持します。

---

イネーブルレジスタ イネーブルレジスタは、ステータスバイト レジスタの中の ESB ビットを設定するイベントレジスタのビットを指定します。イネーブルレジスタが 0 の時には ESB ビットは設定されません。

## 5-7. ステータス レジスタ グループ

**概要** ステータスバイト レジスタ グループは、すべてのステータス レジスタのイベントの状況を確認できます。ステータスバイト レジスタは、"\*STB?"クエリコマンドで読むことができます。



ビット概要	ビット名	説明	ビット	重み
	ERR	Error キューにデータがある場合にビットセット	1	2
	CSUM	Csummary ステータスのサマリのビット設定	2	4
	QUES	Questionable ステータスのサマリのビット設定	3	8
	MAV	Output キューにデータがある場合にビット設定	4	16
	ESB	Standard イベント ステータス レジスタグループの大まかなビットです。	5	32
	MSS /RQS	MSS Bit は、ステータスバイト レジスタとサービスリクエストレジスタの大まかなビットです (ビット 1-5、7) MSS は 1 が設定されます。	6	64
	OPER	Operation ステータスのサマリのビット設定	7	128

ステータスバイトレジスタ ステータスバイト レジスタは 4 つのステータレジスタの他に Error キュー、Output キュー、サービス要求の状態を確認できます。

---

サービス  
リクエスト  
イネーブル  
レジスタ

サービスリクエスト イネーブルレジスタは、ステータスバイト レジスタの MSS/RQS ビットを設定する為のステータスバイト レジスタのビットを指定します。

また、MSS ビットは"\*STB?"クエリコマンドで確認できるビットです。RQS ビットは MSS ビットをサービスリクエストジェネレータで管理して GP-IB インターフェイスの RQS ビットに使用されます。RQS ビットは読み出されるとクリアされます。

## 第6章 付録

### 6-1. エラーメッセージ

Error キューを読み取るときに次のエラーメッセージが発生します。

エラーコードと文字列

説明

コマンド・エラー

0 NoError	エラーなし
-100 Command Error	コマンドエラーです。
-101 Invalid character	構文に対して無効な文字が含まれています。
-102 Syntax error	認識できないコマンドまたはデータが検出されました。
-103 Invalid separator	無効なセパレーターです。不正な文字を検出しました。
-104 Data type error	許可されたものとは異なるデータの型です。
-105 GET not allowed	許可されていないグループ実行トリガ(GET)が受信されました。
-108 parameter not allowed	許可されていないパラメータが受信されました。
-109 Missing parameter	必要な数よりも少ないパラメータが受信されました。
-110 Command header error	コマンド・ヘッダー・エラー
-111 Header separator error	ヘッダの区切りではない文字が検出されました。
-112 Program mnemonic too long	12 以上の文字が含まれています。
-113 Undefined header	ヘッダーは、文法的に正しいですが、定義されていません。
-114 Header suffix out of range	サフィックスの値が範囲外です。
-115 Unexpected number of parameters	受信されたパラメーターの数が違います。
-120 Numeric data error	数値データが違います。
-121 Invalid character in number	データに無効な文字が検出されました。
-123 Exponent too large	指数の大きさが 32000 を超えていました。
-124 Too many digits	桁数が多すぎます。
-128 Numeric data not allowed	許可されていない数値データです。
-130 Suffix error	サフィックスのエラーです。(同様のエラー-131~-139)
-131 Invalid suffix	サフィックスは、構文に従っていません。または無効なサフィックスです。
-134 Suffix too long	サフィックスは、12 以上の文字が含まれています。
-138 Suffix not allowed	サフィックスが許可されていません。
-140 Character data error	文字データのエラーです。
-141 Invalid character data	文字データに無効な文字が含まれています。
-144 Character data too long	文字データに、12 以上の文字が含まれています。
-148 Character data not allowed	許可されていない文字データが検出されました。
-150 String data error	文字列データのエラーです。
-151 Invalid string data	文字列データが無効です。
-158 String data not allowed	許可されていない文字列データが検出されました。

-160 Block data error	ブロック・データのエラーです。
-161 Invalid block data	ブロック・データが無効です。
-168 Block data not allowed	許可されていないブロック・データが検出されました。
-170 Expression error	式データのエラーです。
-171 Invalid expression	式データは無効です。
-178 Expression data not allowed	許可されていない式データが検出されました。
-180 Macro error	マクロの定義、実行のエラーです。
-181 Invalid outside macro definition	マクロの外部の定義が無効です。
-183 Invalid inside macro definition	マクロの内部の定義が無効です。
-184 Macro parameter error	マクロのパラメーターが正しくない数または型があります。
<b>実行エラー</b>	
-200 Execution error	実行エラーです。
-201 Invalid while in local	デバイスがローカル制御で、無効状態です。
-202 Settings lost due to rtl	レムから LOCS または RWLS から LWLS に変化したときのハードローカルコントロールに関連付けられた設定が失われました。
-203 Command protected	コマンドが無効になったため、パスワードで保護されたプログラムコマンドまたはクエリが実行できません。
-210 Trigger error	トリガー・エラーです。
-211 Trigger ignored	"GET、* TRG"またはトリガ信号が受信されましたが、タイミングの問題で無視されました。
-212 Arm ignored	Arm 信号が受信されましたが、無視されました。
-213 Init ignored	別の測定が既に進行中であったので、測定開始要求が無視されました。
-214 Trigger deadlock	トリガーのデッドロックが発生しました。
-215 Arm deadlock	Armのデッドロックが発生しました。
-220 parameter error	パラメーターのエラーです。
-221 Settings conflict	デバイスの状態のために実行できません。
-222 Data out of range	データが範囲外であったために実行できません。
-223 Too much data	ブロック、式、または文字列型のデータが大きすぎます。
-224 Illegal parameter value	無効なパラメーター値
-225 Out of memory.	要求された操作を実行するにはメモリが不足しています。
-226 Lists not same length.	長さの異なる個々のリストを持ってリスト構造を使用しました。
-230 Data corrupt or stale	破損または古いデータです。
-231 Data questionable	データが疑わしいことを示します。
-232 Invalid format	フォーマットが不適切であるため、実行できません。
-233 Invalid version	バージョンが間違っているため、実行できません。
-240 Hardware error	プログラムコマンドまたはクエリが、ハードウェアの問題のために実行できません。
-241 Hardware missing	プログラムコマンドまたはクエリが、ハードウェアが不足しているため実行できません。

-250 Mass storage error	マス・ストレージ・エラーが発生しました。
-251 Missing mass storage	プログラムコマンドまたはクエリが、マス・ストレージが不足しているため実行できません。
-252 Missing media	プログラムコマンドまたはクエリが、メディア不足のために実行できません。
-253 Corrupt media	プログラムコマンドまたはクエリが、破損したメディアのために実行できません。
-254 Media full	メディアが一杯になったため、プログラムコマンドまたはクエリが、実行できません。
-255 Directory full	メディアディレクトリが一杯であったため、プログラムコマンドまたはクエリが、実行できません。
-256 File name not found	メディア上のファイル名が見つからなかったため、プログラムコマンドまたはクエリが、実行できません。
-257 File name error	メディア上のファイル名に誤りがあったため、プログラムコマンドまたはクエリが、実行できません。
-258 Media protected	メディアが保護されているので、プログラムコマンドまたはクエリが、実行できません。
-260 Expression error	式プログラムのエラーが発生しました。
-261 Math error in expression	式プログラムが演算エラーによって実行できません。
-270 Macro error	マクロ関連の実行エラーが発生しました。
-271 Macro syntax error	マクロの構文エラーが発生しました。
-272 Macro execution error	マクロにエラーが発生したために実行できません。
-273 Illegal macro label	マクロ・ラベルが認識できません。
-274 Macro parameter error	マクロが不適切なパラメータを使用しました。
-275 Macro definition too long	文字列またはブロックの内容が長すぎたのでマクロは実行できません。
-276 Macro recursion error	マクロは、それが再帰的であるため実行できません。
-277 Macro redefinition not allowed	マクロはすでに定義されていたので実行できません。
-278 Macro header not found	ヘッダが先に定義されていなかったため、マクロが実行できません。
-280 Program error	プログラムの実行エラーです。
-281 Cannot create program	プログラム作成が失敗しました。 失敗の理由にはメモリ不足も含まれる場合があります。
-282 Illegal program name	プログラムの名前は無効です。
-283 Illegal variable name	プログラムに存在しない変数を参照しました。
-284 Program currently running	プログラムの実行中です。 特定の操作は違反となる場合があります。
-285 Program syntax error	プログラムの構文エラーです。
-286 Program runtime error	プログラムの実行時エラーです。
-290 Memory use error	ユーザーの要求が直接または間接的にメモリに関連するエラーを起こしていることを示し、メモリが悪いことではありません。
-291 Out of memory	メモリが不足しています。

- 292 Referenced name does not exist 参照された名前が存在しません。
- 293 Referenced name already exists 参照された名前は既に存在しています。
- 294 Incompatible type メモリ項目の種類や構造が不十分で互換性のないことを示します。

#### デバイス固有のエラー

- 300 Device-specific error デバイスに依存する一般的なエラーです。
- 310 System error システムエラーが発生しました。
- 311 Memory error メモリ内のエラーが発生しました。
- 312 PUD memory lost "\* PUD"コマンドによるユーザーデータが失われました。
- 313 Calibration memory lost "\* CAL"コマンドによる校正データが失われました。
- 314 Save/recall memory lost "\* SAV"コマンドによるデータが失われました。
- 315 Configuration memory lost コンフィギュレーション・データが失われました。
- 320 Storage fault データストレージを使用する際にファームウェアの障害が検出されました。
- 321 Out of memory メモリが不足しています。
- 330 Self-test failed セルフテストに失敗しました。
- 340 Calibration failed キャリブレーションに失敗しました。
- 350 Queue overflow キューがオーバーフローしました。
- 360 Communication error 通信エラーです。
- 361 Parity error in program message パリティビットが修正されません。
- 362 Framing error in program message ストップビットが検出されません。
- 363 Input buffer overrun 入力バッファが不適切な、または存在しないデータを使用してオーバーフローしています。
- 365 Time out error タイムアウトエラーです。

#### クエリエラー

- 400 Queries error クエリのエラーです。
- 410 Queries INTERRUPTED クエリが中断されました。
- 420 Queries UNTERMINATED クエリが閉じていません。
- 430 Queries DEADLOCKED クエリがデッドロックしました。
- 440 Queries UNTERMINATED after indefinite response クエリが終了していません。

#### イベント・コマンドパワーオン

- 500 Power on 電源の遷移オンを検出しました。

#### ユーザ要求イベント

- 600 User request ローカルコントロールのアクティブ化を検出しました。

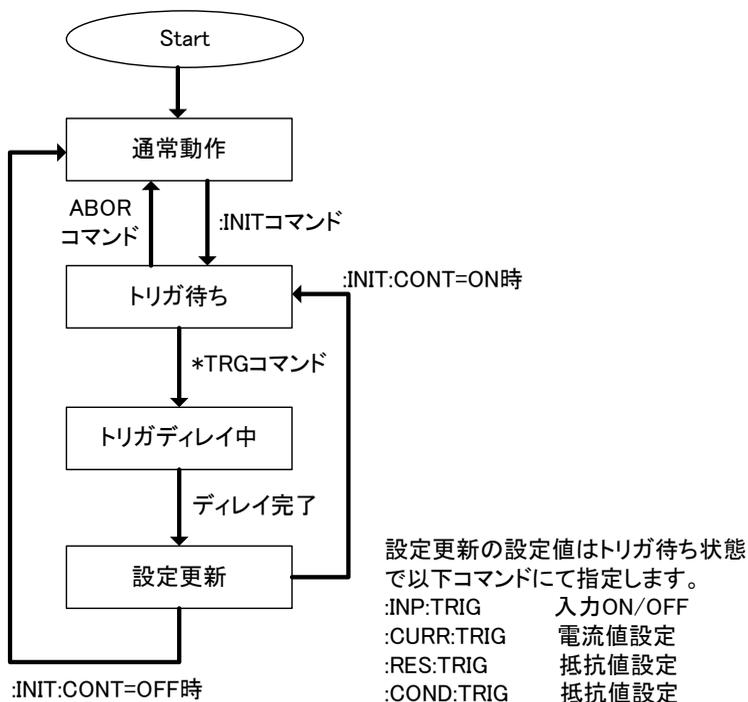
#### コントロールのイベントを要求

- 700 Request control コントローラ・イン・チャージのアクティブ化を検出しました。

#### 操作完了イベント

- 800 Operation complete 操作を完了しました。

## 6-2. トリガコマンド状態遷移



### 6-3. 変更履歴

- B71-0427-01(Ver1.01)→B71-0427-11(Ver1.02)
  - パラメータ変更(コンダクタンス値[mS]→抵抗値[Ω])
  - :RESistance[:VA]
  - :RESistance[:VA]:TRIGgered
  - :RESistance:VB
  - :RESistance:L1
  - :RESistance:L2
  - :RESistance:SET
  - コマンド追加
  - :CONDuctance[:VA]
  - :CONDuctance[:VA]:TRIGgered
  - :CONDuctance:VB
  - :CONDuctance:L1
  - :CONDuctance:L2
  - :CONDuctance:SET
  
- B71-0427-11(Ver1.02)→B71-0427-11(Ver1.03)
  - ファームウェア Ver1.26.1 以後対応
  - OCP テストコマンド群追加
  - ショート禁止関連コマンド追加



## 株式会社 テクシオ・テクノロジー

〒222-0033 神奈川県横浜市港北区新横浜 2-18-13 藤和不動産新横浜ビル 7F  
<http://www.texio.co.jp/>

---

アフターサービスに関しては下記サービスセンターへ

サービスセンター 〒222-0033 神奈川県横浜市港北区新横浜 2-18-13  
藤和不動産新横浜ビル 8F TEL.045-620-2786