

通信コマンド取扱説明書

DM7275-01 DM7275-02 DM7275-03
DM7276-01 DM7276-02 DM7276-03
直流電圧計

- ✓ 本取扱説明においてはコマンドに関する部分のみ扱っています。
- ✓ 本体の通信設定に関しては本体取扱説明書をご覧ください。
- ✓ 本取扱説明の内容につきましては万全を期していますが、ご不明な点や誤りなどお気づきのことがありましたら、コールセンターまたは最寄りの営業所までご連絡ください。
- ✓ 本取扱説明は改善のため予告なしに記載事項を変更することがあります。
- ✓ 本書には著作権によって保護される内容が含まれます。本取扱説明を弊社に無断で転載、複製、改変する事は禁止されています。

目次

1 はじめに	1
メッセージフォーマット	1
出力キューと入力バッファ	5
ステータス・バイト・レジスタ	6
イベント・レジスタ	8
測定値のフォーマット	14
初期化項目	15
コマンド実行時間	16
通信時のエラーについて	16
2 メッセージ一覧	17
3 メッセージリファレンス	24
メッセージリファレンスの見方	24
共通コマンド	25
(1) システム・データ・コマンド	25
*IDN	25
*OPT	25
(2) 内部動作コマンド	25
*RST	25
*TST	25
*SAV	26
*RCL	26
*TRG	26
(3) 同期コマンド	26
*OPC	26
*WAI	26
(4) ステータス、イベント制御コマンド	27
*CLS	27
*ESE	27
*ESR	27
*SRE	28
*STB	28
固有コマンド	29
(1) イベント・ステータス・レジスタ	29
:STATus:OPERation:CONDition	29
:STATus:OPERation:EVENT	29
:STATus:OPERation:ENABLE	30
:STATus:QUEStionable:CONDition	30
:STATus:QUEStionable:EVENT	30
:STATus:QUEStionable:ENABLE	31
(2) 測定値の読み出し	32
:FETCh	33
:READ	34
:MEASure[:VOLTage]:DC	35
:MEASure:TEMPerature	35
:DATA:LAST	35
:ABORt	36
:SYSTem:COMMunicate:FORMat	36
(3) セルフテスト	37
:TEST:ALL	37
(4) 時計	37
:SYSTem:DATE	37
:SYSTem:TIME	37
(5) 測定レンジ	38
[:SENSe:]VOLTage:DC:RANGe	38

	[:SENSe:]VOLTage:DC:RANGe:AUTO	38
(6)	測定ファンクション	38
	[:SENSe:]FUNCTioN[:ON]	38
(7)	測定スピード	39
	[:SENSe:]VOLTage[:DC]:NPLCycles	39
	[:SENSe:]VOLTage[:DC]:APERture:ENABled	39
	[:SENSe:]VOLTage[:DC]:APERture	39
(8)	トリガ	40
	:INITiate:CONTInuous	41
	:INITiate[:IMMediate]	41
	:TRIGger:SOURce	41
	:SAMPle:COUNt	42
	:TRIGger:DElay	42
	:TRIGger:DElay:AUTO	42
(9)	桁数設定	43
	[:SENSe:]VOLTage:DIGits	43
(10)	ラベル表示	43
	:SYSTem:LABel:STATe	43
	:SYSTem:LABel	43
(11)	コンパレータ	44
	:CALCulate:LIMit[:STATe]	44
	:CALCulate:LIMit:BEEPer	44
	:CALCulate:LIMit:ABSolute	44
	:CALCulate:LIMit:UPPer:ENABle	44
	:CALCulate:LIMit:UPPer[:DATA]	44
	:CALCulate:LIMit:LOWer:ENABle	45
	:CALCulate:LIMit:LOWer[:DATA]	45
	:CALCulate:LIMit:DElay	45
	:CALCulate:LIMit:DElay:COUNt	45
	:CALCulate:LIMit:CLEar[:IMMediate]	45
	:CALCulate:LIMit:RESult	45
(12)	BIN	46
	:CALCulate:BIN[:STATe]	46
	:CALCulate:BIN:ENABle	46
	:CALCulate:BIN:UPPer	46
	:CALCulate:BIN:LOWer	46
	:CALCulate:BIN:RESult	47
(13)	測定条件の保存と読み込み	47
	*SAV	47
	:SYSTem:PANel:CLEar	47
	:SYSTem:PANel:NAME	47
	:SYSTem:PANel:DATE	47
	:MMEMory:STORE:STATe	48
	:MMEMory:LOAD:STATe	48
	:MMEMory:STATe:RECall:AUTO	48
	:MMEMory:STATe:RECall:SElect	48
(14)	スムージング	49
	:CALCulate:SMOothing[:STATe]	49
	:CALCulate:SMOothing:RESPonse	49
(15)	ホールド	49
	[:SENSe:]HOLD:AUTO	49
	[:SENSe:]HOLD:BOUND	49
(16)	コンタクトチェック	50
	[:SENSe:]VOLTage:DC:CONtact:CAPacitance	50
	[:SENSe:]VOLTage:DC:CONtact:CAPacitance:STATe	50
	[:SENSe:]CONtact:CAPacitance:THReshold	50
	[:SENSe:]VOLTage:DC:CONtact:CAPacitance:TIME	50
(17)	入力抵抗の切替	51
	[:SENSe:]VOLTage:IMPedance:AUTO	51

(18) ゼロ点調節(NULL機能)	51
[:SENSe:]VOLTage:DC:NULL[:STATe]	51
[:SENSe:]VOLTage:DC:NULL:VALue	51
(19) 温度補正(TC)	52
:CALCulate:TCORrect:STATe	52
:CALCulate:TCORrect:PARAmeter	52
(20) スケーリング	52
:CALCulate:SCALE[:STATe]	52
:CALCulate:SCALE:PARAmeterA	52
:CALCulate:SCALE:PARAmeterB	52
:CALCulate:SCALE:UNIT:STATe	53
:CALCulate:SCALE:UNIT	53
(21) 統計	53
:CALCulate:AVERage[:STATe]	53
:CALCulate:AVERage:CLEar[:IMMediate]	53
:CALCulate:AVERage:CLEar:PRINt	53
:CALCulate:AVERage:COUNt?	53
:CALCulate:AVERage:ALL?	54
:CALCulate:AVERage:SDEVIation?	54
:CALCulate:AVERage:AVERage	54
:CALCulate:AVERage:MINimum	54
:CALCulate:AVERage:MAXimum	54
:CALCulate:AVERage:	54
:CALCulate:AVERage:LIMit	55
:CALCulate:AVERage:BIN	55
:CALCulate:AVERage:CP?	55
(22) キーロック	55
:SYSTem:KLOCK	55
(23) 音設定	55
:SYSTem:CLICk:STATe	55
:SYSTem:HOLD:BEEPer:STATe	55
:SYSTem:BEEPer:STATe	56
:SYSTem:BEEPer:VOLume	56
:SYSTem:BEEPer[:IMMediate]	56
(24) 表示設定	56
:DISPlay[:STATe]	56
:DISPlay:BACKlight	56
:DISPlay:TYPe	57
:DISPlay:VIEW	57
(25) 電源周波数	57
:SYSTem:LFRequency	57
(26) システムリセット	57
:SYSTem:PRESet/:STATus:PRESet	57
(27) 通信設定	58
:SYSTem:LOCAl	58
:SYSTem:COMMunicate	58
:SYSTem:COMMunicate:LOGGing	58
:SYSTem:COMMunicate:MONitor	58
:SYSTem:COMMunicate:GPIB:ADDReSS	58
:SYSTem:COMMunicate:GPIB:TERMinator	58
:SYSTem:COMMunicate:RS232C:SPEED	59
:SYSTem:COMMunicate:LAN:IPADdress	59
:SYSTem:COMMunicate:LAN:CONTRol	59
:SYSTem:COMMunicate:LAN:SMASK	59
:SYSTem:COMMunicate:LAN:GATeway	60
:SYSTem:COMMunicate:LAN:UPDate	60
:SYSTem[:COMMunicate:LAN]:MAC	60
(28) データ出力設定	61
:SYSTem:COMMunicate:DATAout	61

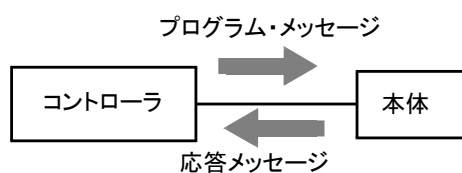
:SYSTem:COMMunicate:DATAout:FORMat	61
:SYSTem:COMMunicate:DATAout:CONDition	61
:SYSTem:LOCAle:DATE	61
:SYSTem:LOCAle:DATE:SEParator	62
:SYSTem:LOCAle:NUMeric:DOT	62
:SYSTem:LOCAle:NUMeric:SEParator	62
(29) メモリ機能	63
:R	63
:DATA:REMOve	63
:DATA:POINts:EVENT:THReshold	64
:DATA:POINts	64
:DATA:CLEar	64
:MMEMory:STORe:DATA	64
(30) EXT I/O	65
:IO:MODE	65
:IO:FILTer:STATe	65
:IO:FILTer:TIME	65
:IO:EOM:MODE	65
:IO:EOM:PULSe	65
:IO:INPut	66
:IO:OUTPut	66
(31) 画面データの保存	66
:HCOPy:SDUMp:DATA	66
(32) *IDN?の応答設定	67
:SYSTem:IDNStr	67
他社製品との互換コマンド	68
:CALCulate:SCALE:GAIN	68
:CALCulate:SCALE:OFFSet	68
:SYSTem:ERRor[:NEXT]	69
:MEASure:TEMPerature	69
*PSC	70
*CAL	70
:HCOPy:SDUMp:DATA:FORMat	70
[:SENSe:]VOLTage[:DC]:ZERO:AUTO	70
:OUTPut:TRIGger:SLOPe	70
:TRIGger:COUNt	70
:SYSTem:REMOte/:SYSTem:RWLock	70
4 データ取得方法	71
5 サンプルプログラム	73
Visual Basic 5.0/6.0 で作成する	73
RS-232C/USBでの通信 (Microsoft Visual Basic Professional MSComm使用)	73
■ シンプルな電圧測定	73
■ パソコンのキーによって電圧測定	74
■ 外部トリガによる測定 1	75
■ 外部トリガによる測定 2	76
■ 測定条件設定	77
GP-IBでの通信 (National Instruments社のGP-IBボードを使用)	78
■ シンプルな電圧測定	78
■ パソコンのキーによって電圧測定	79
■ 外部トリガによる測定 1	80
■ 外部トリガによる測定 2	81
■ 測定条件設定	82
Visual Basic 2010 で作成する	83
6 デバイスの文要書件 [GP-IB]	90

1 はじめに

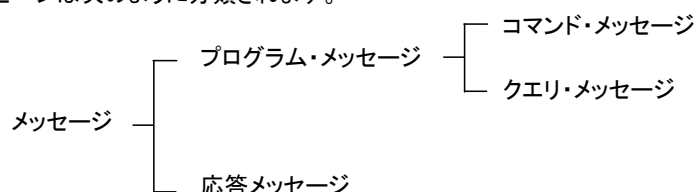
本紙では、DM7275-01/ DM7275-02/ DM7275-03/ DM7276-01/ DM7276-02/ DM7276-03を本体と表記します。

プログラム作成時には、通信モニタ機能を使用すると測定画面にコマンドや応答が表示され便利です。通信モニタ機能については本体の取扱説明書をご覧ください。

インタフェースで本体を制御するために、各種のメッセージが用意されています。
メッセージには、コンピュータなどのコントローラから本体に送信するプログラム・メッセージ と本体からコントローラに送信する応答メッセージがあります。



メッセージは次のように分類されます。



データを持つコマンドは、指定されたデータ形式で入力してください。

メッセージフォーマット

■ プログラム・メッセージ

プログラム・メッセージは、コマンド・メッセージとクエリ・メッセージに分けることができます。

(1) コマンド・メッセージ

機器の設定、リセットなどの機器を制御する命令

(例) レンジを設定する命令

:VOLTAGE:DC:RANGE 100

↑ ↑ ↑
ヘッダ部 スペース データ部

(2) クエリ・メッセージ

動作結果、測定結果、または機器の設定状態を問い合わせる命令

(例) 現在の測定レンジを問い合わせる命令

:VOLTAGE:DC:RANGE?

↑ ↑
ヘッダ部 クエスチョンマーク

参照: 「ヘッダ(p.2)」、「セパレータ(p.3)」、「データ部(p.4)」

■ 応答メッセージ

クエリ・メッセージを受信し、構文をチェックした時点で作成します。

クエリ・メッセージを受信したときに、何らかのエラーが発生した場合は、そのクエリ・メッセージに対する応答メッセージは作成されません。

■ コマンド・シンタックス

コマンド名は、実行しようとする機能にできるだけ理解しやすい命令が選ばれ、しかも短縮が可能です。コマンド名そのものを " ロングフォーム " といい短縮したものを " ショートフォーム " といいます。本書では、ショートフォームの部分を大文字で、残りの部分を小文字で記述しますが、大文字と小文字のどちらでも受け付けます。

FETCH?	OK(ロングフォーム)
FETC?	OK(ショートフォーム)
FET?	エラー

本体からの応答メッセージは、大文字のロングフォームで返されます。

■ ヘッダ

プログラム・メッセージには、必ずヘッダが必要です。

(1) コマンド・プログラム・ヘッダ

単純コマンド型、複合コマンド型、共通コマンド型の3種類があります。

- **単純コマンド型ヘッダ**

英文字から始まる1語で構成されるヘッダ

:ABORT

- **複合コマンド型ヘッダ**

コロン ":" で区切られる複数の単純コマンド型ヘッダで構成されるヘッダ

:VOLTage:DC:RANGe

- **共通コマンド型ヘッダ**

共通コマンドであることを示すアスタリスク "*" で始まるヘッダ(IEEE 488.2で規定されたもの)

***RST**

(2) クエリ・プログラム・ヘッダ

機器のコマンドに対する動作結果、測定結果、または現在の機器の設定状態を問い合わせるために使用します。

下の例のように、プログラム・ヘッダの最後にクエスチョンマーク "?" が付きます。

:FETCh?

VOLTage:DC:RANGe?

[] で囲まれた部分は省略可能です。

[:SENSe:]VOLTage:DC:RANGe



どちらも可能

:SENSe:VOLTage:DC:RANGe
VOLTage:DC:RANGe

■ メッセージ・ターミネータ

本体は、メッセージ・ターミネータ(デリミタ)として以下のものを受け付けます。

[RS-232C/USB/LAN]

- CR
- CR+LF

[GP-IB]

- LF
- CR+LF
- EOI
- EOI を伴う LF

また、応答メッセージのターミネータとして、インタフェースの設定によって以下のものが選択できます。
設定については「デリミタの設定」(p.58)をご覧ください。

[RS-232C/USB/LAN]

- CR+LF

[GP-IB]

- EOI を伴う LF(初期状態)
- CR と EOI を伴う LF

■ セパレータ

(1) メッセージ単位セパレータ

複数のメッセージは、それぞれセミコロン (;) でつなげることで、1行に記述することができます。

: VOLTAGE:DC:RANGE 10[*IDN?

- メッセージを続けて記述した場合、文中でエラーが発生すると、それ以降からメッセージ・ターミネータまでのメッセージは実行されません。

(2) ヘッダ・セパレータ

ヘッダとデータを持つメッセージは、空白 (アスキーコード 20H) を使用することで、ヘッダ部とデータ部に分離します。

:VOLTAGE:DC:RANGE[10

(3) データ・セパレータ

複数のデータを持つメッセージは、データの間にカンマ (,) で分離します。

:SYSTEM:DATE 15[1[1

■ データ部

本体では、データ部に "文字データ"、"10進数値データ" および "文字列データ" を使用し、コマンドにより使い分けます。

(1) 文字データ

必ず英文字で始まり、英文字と数字で構成されるデータです。文字データは、大文字と小文字の両方を受け付けますが、本体からの応答メッセージは必ず大文字で返します。コマンドデータ部で<1/0/ON/OFF>とあるものについては、0はOFF、1はONと同様の動作をし、クエリの応答は、OFFは0、ONは1となります。

:VOLTAGE:DC:RANGE:AUTO OFF

(2) 10進数値データ

数値データのフォーマットには、NR1、NR2、NR3形式があります。それぞれ符号付き数値、符号なし数値の両方を受け付けます。符号なし数値の場合、正の数値として扱います。また、数値の精度が本体の取扱範囲を超える場合、四捨五入します。

- NR1 整数データ (例 : +12、-23、34)
- NR2 小数データ (例 : +1.23、-23.45、3.456)
- NR3 浮動小数点指数表示データ (例 : +1.0E-2、-2.3E+4)

以上の 3 種類の形式をすべて含む形式を "NRf 形式" と呼びます。

本体では、NRf 形式で受け付けます。応答データに関しては、コマンドごとにフォーマットを指定しており、その形式で送信します。

:STATus:OPERation:ENABle 49

:FETCH?

+102.20192E-03

(3) 文字列データ

- 文字列データは、前後をクォーテーション・マークで囲みます。
- 8ビット ASCII 文字からなるデータです。
- 本体で扱えない文字は、エラーとなります。
- 下記の2文字は本体設定と通信設定で異なります。

本体設定	通信設定
Ω	@
°C	\$

※スケールリング単位、ラベル機能のみ対応しており、その他機能については、コマンドによりΩ、°Cを設定することはできません。

- クォーテーション・マークとして、本体からの送信はダブル・クォート" " ですが、受信はダブル・クォートとシングル・クォート' 'の両方を受け付けます。

:SYSTem:LABel "LABEL_01"

[GP-IB]

本体は IEEE 488.2 に完全対応ではありません。できるだけ、リファレンスに示したデータを使用してください。

■ 複合コマンド型ヘッダの省略

複合コマンドの中で、先頭の部分が共通であるもの（例：**:CALCulate:SCALe:PARAmeterA**と**:CALCulate:SCALe:PARAmeterB**など）は、これらを続けて記述する場合に限り、コマンドの共通部分（例：**:CALCulate:SCALe**）を省略することができます。

この共通部分は" カレント・パス "と呼ばれ、これがクリアされるまではそれ以降のコマンドは『カレント・パスを省略したもの』と判断して解析を行います。

カレント・パスの使用方法を以下の例に示します。

通常表記

:CALCulate:SCALe:PARAmeterA 1.0; :CALCulate:SCALe:PARAmeterB 0.0

省略表記

:CALCulate:SCALe:PARAmeterA 1.0; PARAmeterB 0.0

↑
カレント・パスとなり、次のコマンドでは省略できます。

カレント・パスは、電源投入、キー入力によるリセット、コマンドの先頭のコロンの" : "、およびメッセージ・ターミネータの検出でクリアします。

共通コマンド型のメッセージは、カレント・パスに関係なく実行可能です。また、カレント・パスに影響を与えません。

単純および複合コマンド型ヘッダの先頭にコロン" : "を付ける必要はありません。ただし、省略形との混乱と誤動作を防ぐため、弊社では、コマンドの先頭に" : "を付けることを推奨しています。

出力キューと入力バッファ

■ 出力キュー

応答メッセージは出力キューにためられ、コントローラでデータを読み出すとクリアされます。それ以外に出力キューがクリアされるのは、以下の場合です。

- 電源投入
- デバイスクリア [GP-IB]
- クエリエラー

■ 入力バッファ

入力バッファの容量は256バイトです。

256バイトを超えるデータが送信されて入力バッファがいっぱいになると、USB/LANおよびGP-IBインタフェース・バスは空きができるまで待ち状態になります。

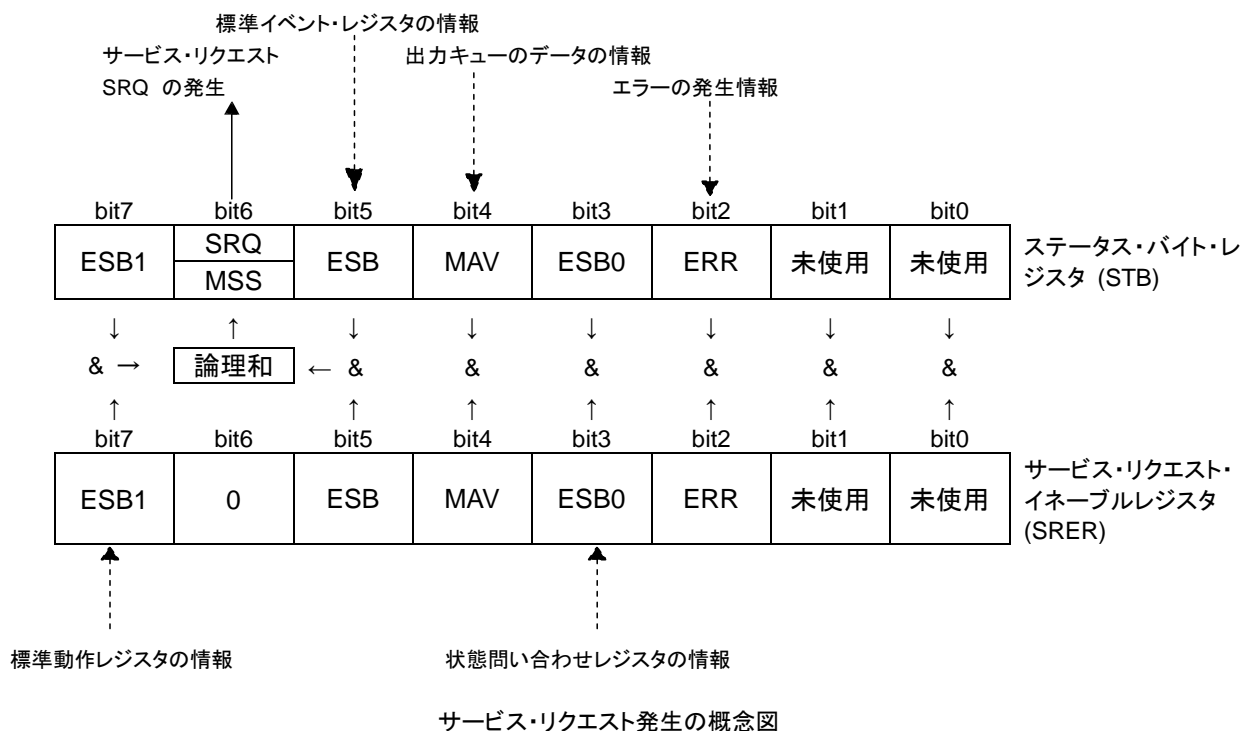
RS-232Cは256バイトを超えたデータは処理できない可能性があります。

注記: 1行のコマンドの長さは256バイト未満にしてください。

ステータス・バイト・レジスタ

[GP-IB]

本体は、サービス・リクエスト機能によるシリアル・ポールに関する部分に、IEEE 488.2で規定されているステータス・モデルを採用しています。イベントとは、サービス・リクエストを発生させるための要因となる事柄をいいます。



ステータス・バイト・レジスタには、イベント・レジスタと出力キューの情報がセットされています。これらの情報の中からサービス・リクエスト・イネーブル・レジスタによって、必要なものを選択することができます。選択された情報がセットされた場合は、ステータス・バイト・レジスタのビット6(MSS マスタ・サマリ・ステータス・ビット)がセットされ、SRQ(サービス・リクエスト)メッセージを発生し、これによりサービス・リクエストが発生します。

注記: SRQ (サービス・リクエスト) は GP-IB のみの機能です。

ただし STB (ステータス・バイト・レジスタ) の情報は *STB? コマンドを使用することで RS-232C/USB/LANでも取得することができます。

STB (ステータス・バイト・レジスタ) の情報は *STB? コマンドを使用することで取得できます。

[RS-232C/USB/LAN]

RS-232C/USB/LAN には、サービス・リクエストの発生機能はありません。ただし、SRERの設定とSTBの読み出しは可能です。

■ ステータス・バイト・レジスタ (STB)

ステータス・バイト・レジスタとは、シリアル・ポールを行うときに、本体からコントローラに出力する8ビットのレジスタです。サービス・リクエスト・イネーブル・レジスタで使用可能に設定されたビットのうち、ステータス・バイト・レジスタのビットが1つでも"0"から"1"になると、MSSビットは"1"になります。それと同時にSRQビットも"1"になり、サービス・リクエストが発生します。

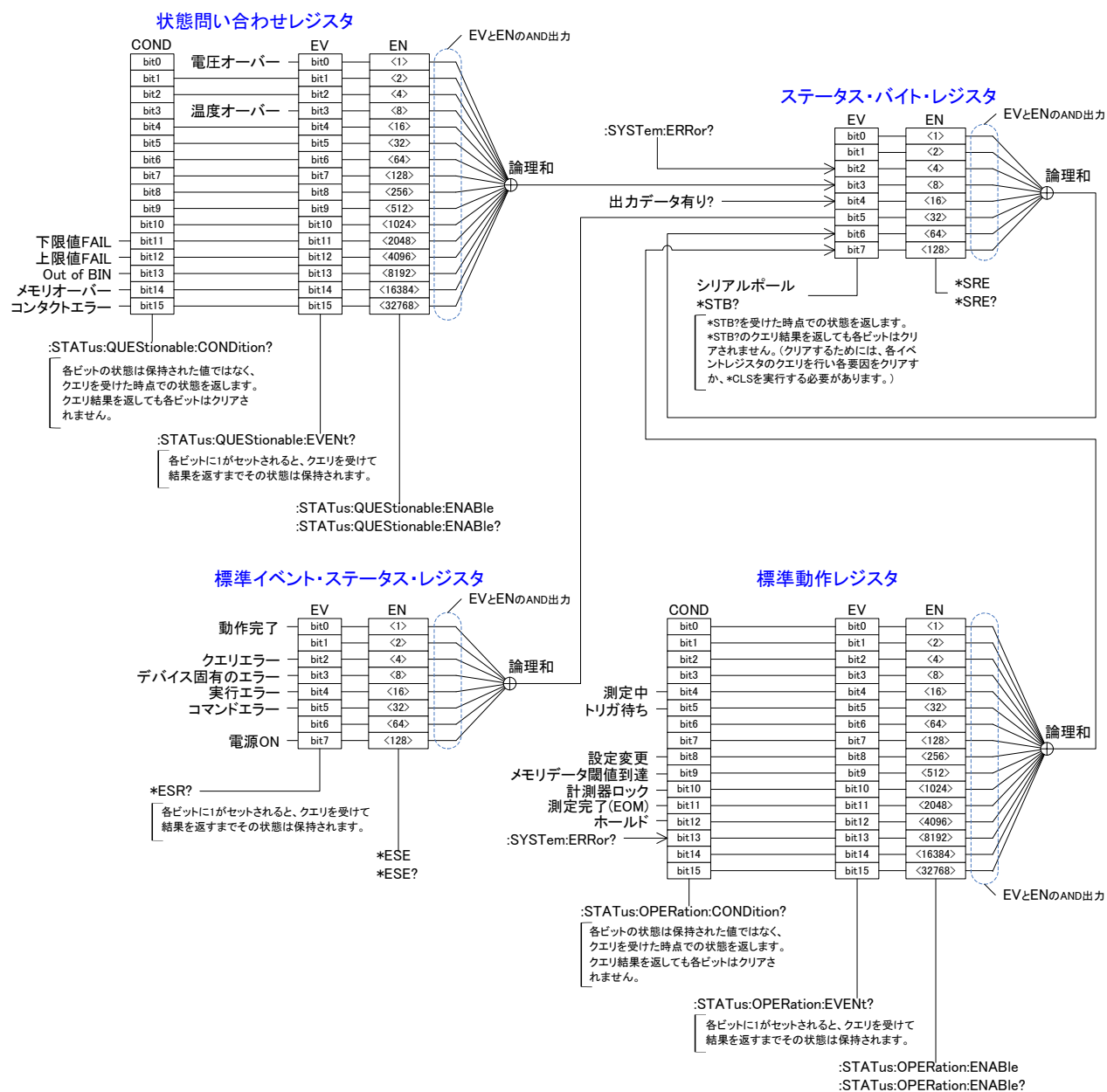
SRQ ビットは、常にサービス・リクエストに同期しておりシリアル・ポールされるときにのみ読み出され、同時にクリアされます。MSSビットは、***STB?**クエリでのみ読み出されますが ***CLS** コマンドなどでイベントをクリアするまでクリアされません。

ビット7	ESB1	イベント・サマリ(論理和)・ビット1 標準動作レジスタの論理和を示します。
ビット6	SRQ	サービス・リクエストを発信すると"1"になります。
	MSS	ステータス・バイト・レジスタの他のビットの論理和を表します。
ビット5	ESB	標準イベント・サマリ(論理和)・ビット 標準イベント・ステータス・レジスタの論理和を表します。
ビット4	MAV	メッセージ・アベイラブル 出力キューにメッセージがあることを示します。
ビット3	ESB0	イベント・サマリ(論理和)・ビット0 状態問い合わせレジスタの論理和を示します。
ビット2	ERR	エラービット エラー情報があると1になります。 :SYSTEM:ERROR?によりエラー情報を出力するとリセットされます。
ビット1		未使用
ビット0		未使用

■ サービス・リクエスト・イネーブル・レジスタ (SRER)

サービス・リクエスト・イネーブル・レジスタは、各ビットを"1"に設定すると、ステータス・バイト・レジスタ内の対応するビットが使用可能になります。

イベント・レジスタ



■ 標準イベント・ステータス・レジスタ (SESR)

標準イベント・ステータス・レジスタは、8ビットのレジスタです。標準イベント・ステータス・イネーブル・レジスタで使用可能に設定したビットのうち、標準イベント・ステータス・レジスタのビットがひとつでも"1"になると、ステータス・バイト・レジスタのビット5(ESB)が"1"になります。

参照:「標準イベント・ステータス・レジスタ(SESR)と標準イベント・ステータス・イネーブル・レジスタ(SESER)」(p.10)

標準イベント・ステータス・レジスタの内容は次のときクリアされます。

- ***CLS** コマンドを実行したとき
- イベント・レジスタの クエリを実行したとき (***ESR?**)
- 電源を再投入したとき

ビット7	PON	電源投入フラグ 電源投入時、および停電からの復帰時に"1"になります。
ビット6	URQ (未使用)	本体では使用しません。 ユーザ・リクエスト
ビット5	CME	コマンドエラー(メッセージ・ターミネータまでのコマンドを無視します。) 受信したコマンドに文法上、意味上の誤りがあるときに"1"になります。 <ul style="list-style-type: none"> • プログラム・ヘッダに誤りがある場合 • データの数が指定と違う場合 • データの形式が指定と違う場合 • 本体にないコマンドを受信した場合
ビット4	EXE	実行エラー 何らかの理由で受信したコマンドが実行できないときに"1"になります。 <ul style="list-style-type: none"> • 指定したデータが設定範囲外の場合 • 指定したデータが設定できない場合 • 別の機能が動作中で実行できない場合
ビット3	DDE (未使用)	本体では使用しません。 機器に依存したエラー コマンドエラー、クエリエラー、実行エラー以外の原因でコマンドを実行できなかったときに"1"になります。
ビット2	QYE	クエリエラー(出力キューをクリアします) 出力キュー関連の処理で異常が発生したときに"1"になります。 <ul style="list-style-type: none"> • データが出力キューをあふれた場合 • 出力キュー内のデータが失われた場合
ビット1	RQC (未使用)	本体では使用しません。 コントローラ権の要求
ビット0	OPC	動作の完了 <ul style="list-style-type: none"> • *OPCコマンドを実行した場合 • *OPCコマンドまでの全メッセージの動作が終了した場合

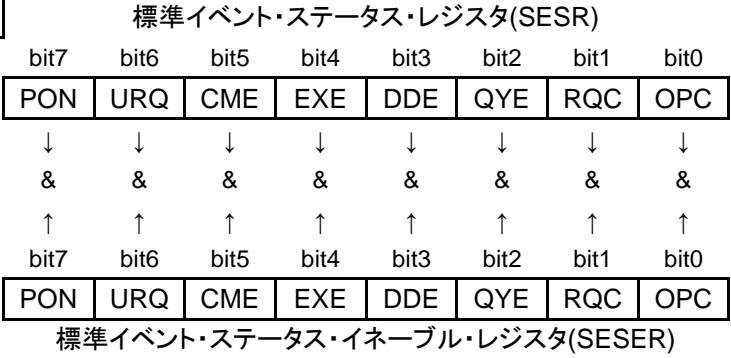
■ 標準イベント・ステータス・イネーブル・レジスタ (SESER)

標準イベント・ステータス・イネーブル・レジスタは、各ビットを "1"に設定することで、標準イベント・ステータス・レジスタ内の対応するビットを使用可能にします。

標準イベント・ステータス・レジスタ(SESER)と標準イベント・ステータス・イネーブル・レジスタ(SESER)

ステータス・バイト・レジスタ(STB)

bit6	bit5	bit4
SRQ	ESB	MAV
MSS		



■ 固有のイベント・ステータス・レジスタ

本体のイベントを管理するために、2つのイベント・ステータス・レジスタを用意しています。イベント・ステータス・レジスタは16ビットのレジスタです。

各イネーブル・レジスタで使用可能に設定したビットのうち、イベント・ステータス・レジスタのビットがひとつでも"1"になると、以下ようになります。

- 標準動作レジスタの場合：ステータス・バイト・レジスタのビット7(ESB1)が"1"
- 状態問い合わせレジスタの場合：ステータス・バイト・レジスタのビット3(ESB0)が"1"

イベント・ステータス・レジスタ 0、1の内容は次のときクリアされます。

- ***CLS** コマンドを実行したとき
- 各イベント・ステータス・レジスタのクエリを実行したとき
(**:STATus:OPERation:EVENT?**、**:STATus:QUESTIONable:EVENT?**)
- 電源を再投入したとき

標準動作レジスタ

ビット15	-	未使用
ビット14	-	未使用
ビット13	ERR	エラーが発生した場合に、"1"になります（詳細は、 :SYSTEM:ERROR? にて詳細を取得すると、クリアされます）
ビット12	HOLD	オートホールドが実行された場合に、"1"になります
ビット11	EOM	測定が完了した場合に、"1"になります
ビット10	LOCK	本体がリモート状態の場合に、"1"になります
ビット9	MEM_RDY	メモリデータの個数が、 :DATA:POINTS:EVENT:THReshold にて設定された個数に達した場合に、"1"になります
ビット8	SET	最後の :INIT 、または、最後の正常測定以降に、設定が変更された場合に、"1"になります
ビット7	-	未使用
ビット6	-	未使用
ビット5	WAIT_TRG	トリガ待ち状態の場合に、"1"になります
ビット4	MEAS	測定状態の場合に、"1"になります
ビット3	-	未使用
ビット2	-	未使用
ビット1	-	未使用
ビット0	-	未使用

標準動作レジスタ・グループのイベント・レジスタ(**:STATus:OPERation:EVENT?**)と
標準動作レジスタ・グループのイネーブル・レジスタ(**:STATus:OPERation:ENABLE**)

ステータス・バイト・レジスタ(STB)

bit7 bit6 bit5

ESB1	SRQ/MSS	MAV
------	---------	-----

標準動作レジスタ・グループのイベント・レジスタ(**:STATus:OPERation:EVENT?**)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
↑	-	MEM_OVR	ERR	HOLD	EOM	LOCK	MEM_RDY	SET	-	-	WAIT_TRG	MEAS	-	-	-	-
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
論理和 ←	&	&	&	&	&	&	&	&	&	&	&	&	&	&	&	&
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	ERR	HOLD	EOM	LOCK	MEM_RDY	SET	-	-	WAIT_TRG	MEAS	-	-	-	-

標準動作レジスタ・グループのイネーブル・レジスタ(**:STATus:OPERation:ENABLE**)

状態問い合わせレジスタ

ビット15	CNT_ERR	コンタクトエラーが発生した場合に、"1"になります
ビット14	MEM_OVR	内部の測定メモリが一杯(5000データ)になった場合に、"1"になります
ビット13	OB	BIN測定の結果が、OUT OF BIN の場合に、"1"になります
ビット12	HI	コンパレータの判定結果が、上限値FAILの場合に、"1"になります
ビット11	LO	コンパレータの判定結果が、下限値FAILの場合に、"1"になります
ビット10	-	未使用
ビット9	-	未使用
ビット8	-	未使用
ビット7	-	未使用
ビット6	-	未使用
ビット5	-	未使用
ビット4	-	未使用
ビット3	TMP_OVR	測定温度が測定範囲外の場合に、"1"になります
ビット2		未使用
ビット1	-	未使用
ビット0	VLT_OVR	測定電圧が測定範囲外の場合に、"1"になります

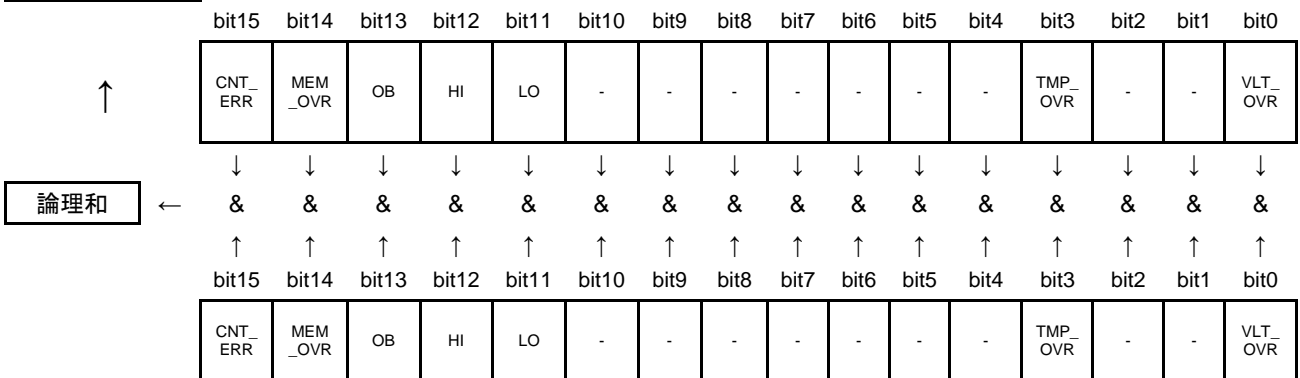
状態問い合わせレジスタ・グループのイベント・レジスタ(:**STATUS:QUESTIONABLE:EVENT?**)と
 状態問い合わせレジスタ・グループのイネーブル・レジスタ(:**STATUS:QUESTIONABLE:ENABLE**)

ステータス・バイト・レジスタ(STB)

bit4 bit3 bit2

MAV	ESB0	ERR
-----	------	-----

状態問い合わせレジスタ・グループのイベント・レジスタ(:**STATUS:QUESTIONABLE:EVENT?**)



状態問い合わせレジスタ・グループのイネーブル・レジスタ(:**STATUS:QUESTIONABLE:ENABLE**)

■ 各レジスタの読み出し・書き込み

レジスタ	読み出し	書き込み
ステータス・バイト・レジスタ	*STB?	-
サービス・リクエスト・イネーブル・レジスタ	*SRE?	*SRE
標準イベント・ステータス・レジスタ	*ESR?	-
標準イベント・ステータス・イネーブル・レジスタ	*ESE?	*ESE
標準動作レジスタ・グループのイベント・レジスタ (状態データ)	:STATus:OPERation :CONDition?	-
標準動作レジスタ・グループのイベント・レジスタ (イベントデータ)	:STATus:OPERation :EVENTt?	-
標準動作レジスタ・グループのイネーブル・レジスタ	:STATus:OPERation :ENABLE?	:STATus:OPERation :ENABLE
状態問い合わせレジスタ・グループのイベント・レジスタ 問い合わせ(状態データ)	:STATus:QUESTiona ble:CONDition?	-
状態問い合わせレジスタ・グループのイベント・レジスタ 問い合わせ(イベントデータ)	:STATus:OPERation :EVENTt?	-
状態問い合わせレジスタ・グループのイネーブル・レジスタ	:STATus:QUESTiona ble:ENABLE?	:STATus:QUESTiona ble:ENABLE

■ GP-IB コマンド

インタフェース・ファンクションにより、以下のコマンドが使用できます。

コマンド	内容
GTL	Go To Local リモート状態を解除し、ローカル状態にします。
LLO	Local Lock Out ローカル・キーを含めたすべてのキーを、操作不可能にします。
DCL	Device CLear 入力バッファ・出力キューをクリアします。
SDC	Selected Device Clear 入力バッファ・出力キューをクリアします。
GET	Group Execute Trigger 外部トリガ(トリガソース<EXTERNAL>)のとき、1回サンプリング処理を行います。

測定値のフォーマット

測定値のフォーマット設定により、**:FETCh?**、**:READ?**、**MEASure[:VOLTage]:DC?**で得られる測定値のフォーマットを以下のように変更することができます。

- 電圧値: 単位V(**:SYSTem:COMMunicate:FORMat FIX** 設定時)

測定値レンジ	測定値	±OvrRng	表示時	測定異常時
100mV	± 000.00000E-03	±990.00000E+35		+991.00000E+35
1V	± 0000.00000E-03	±9900.00000E+34		+9910.00000E+34
10V	± 00.000000E+00	±99.000000E+36		+99.100000E+36
100V	± 000.000000E+00	±990.000000E+35		+991.000000E+35
1000V	± 0000.000000E+00	±9900.000000E+34		+9910.000000E+34

注記: 小数点の位置や指数部は、スケーリングの設定により変わります。

スケーリングについては本体の取扱説明書をご覧ください。

- 整数部の桁数が足りない場合は、0が入ります。

例) 1000Vレンジで測定値が1Vの場合、+0001.0000E+00となります。

- 仮数部は桁数設定に応じて変化します。

- 電圧値: 単位V(**:SYSTem:COMMunicate:FORMat FLOAT** 設定時)

測定値	±OvrRng	表示時	測定異常時
± 0.000000000E±00	±9.900000000E+37		+9.910000000E+37

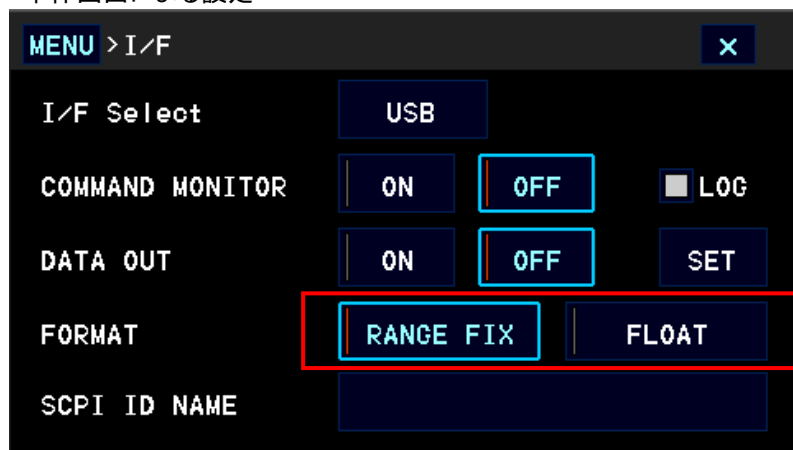
注記: 測定値は小数点以下8桁の浮動小数値で表現されます。

測定値のフォーマットは下記の方法のいずれかで変更することが可能です。

- 通信コマンドによる設定

:SYSTem:COMMunicate:FORMat FLOAT/FIX にて変更 (参照: データ出力設定 p.36)

- 本体画面による設定



初期化項目

項目	初期化方法	電源投入時	キーリセット	*RST コマンド	デバイス クリア (GP-IBのみ)	*CLS コマンド	工場出荷時
GP-IB アドレス		-	1	-	-	-	1
RS-232C 設定(通信速度)		-	9600	-	-	-	9600
LAN IPアドレス		-	0.0.0.0	-	-	-	0.0.0.0
LAN サブネットマスク		-	255.255.255.0	-	-	-	255.255.255.0
LAN デフォルトゲートウェイ		-	0.0.0.0	-	-	-	0.0.0.0
LAN ポート		-	23	-	-	-	23
デバイス固有の機能(レンジなど)		-	●	●	-	-	●
出力キュー		●	●	-	●	-	●
入力バッファ		●	●	-	●	-	●
ステータス・バイト・レジスタ		●	●	-	●*1	●*2	●
イベント・レジスタ		●*3	●	-	-	●	●
イネーブル・レジスタ		●	●	-	-	-	●
カレント・パス		●	●	-	●	-	●
応答メッセージのターミネータ(GP-IB)		-	LF+EOI	-	-	-	LF+EOI

*1. MAV ビット(ビット4)のみクリアする。

*2. MAV ビット以外をクリアする。

*3. PON ビット(ビット7)は除く。

コマンド実行時間

実行時間はロングフォームのコマンドの " 解析 + 処理時間 " を示します。

ただし、データを持つコマンドは < データ部 > で指定されるデータ形式にしたがって記述されたときの時間です。

- 通信処理の頻度、処理内容により表示処理に遅れが出る場合があります。
- ***TRG**、**:INIT**を除き、すべてのコマンドはシーケンシャル型の動作です。
- コントローラとの通信では、データの転送時間を追加する必要があります。USB、GP-IB、転送時間は、コントローラにより異なります。

RS-232C の転送時間は、スタートビット 1、データ長 8、パリティなし、ストップビット1の計10ビット、伝送速度(ボーレート)設定をN bpsとした場合は、およそ次のようになります。

転送時間T [1文字/ 秒] = ボーレートN [bps]/10[bit]

測定値は17文字なので1データの転送時間は17/Tになります。

(例)9600bpsの場合17/(9600/10) = 約17ms

- 設定用のコマンドは、変更後に測定が安定するまで、待ち時間をとってください。

コマンド	実行時間(通信時間除く)
*RST	
:SYSTem:PRESet	700ms以内
:STATus:PRESet	
[[:SENSe:]VOLTage:DC:RANGe	700ms以内
[[:SENSe:]VOLTage[:DC]:NPLCycles	2 ms以内
:FETCh?	10ms以内
:READ?	測定時間+15ms以内
*RCL	700ms以内
*TST?	20ms以内
上記以外のコマンド	10ms以内

通信時のエラーについて

以下の場合にメッセージを実行するとエラーになります。

- **コマンドエラー**
メッセージのつづりが間違っている場合
コマンドまたはクエリのデータ部の形式が間違っている場合
- **クエリエラー**
コントローラ側が受信できない状態にあり、本体より応答メッセージを送信できない場合
- **実行エラー**
指定された文字データまたは数値データ以外で設定した場合

2 メッセージ一覧

メッセージ []:省略可	データ []:省略可、0:応答データ	説明
共通コマンド		
*IDN?	<(メーカー名)><(モデル名)><(製造番号)><(ソフトウェアバージョン)>	機器の ID(識別コード) の問い合わせ
*OPT?	<(0/GPIB)><(LAN)><(0/RS232C)>	インストールされているオプションを識別
*RST		機器の初期化
*TST?	<(PASS/FAIL)>	セルフテストの実行と結果の問い合わせ
*SAV	<(パネル番号)>	測定条件の保存(パネルセーブ)
*RCL	<(パネル番号)>	測定条件の読み込み(パネルロード)
*TRG		サンプリングの要求
*OPC		実行中の全動作終了後、SESR の OPC をセット
*OPC?		実行中の全動作終了後、ASCII の 1 を応答
*WAI		コマンド処理終了後、続くコマンドを実行
*CLS		イベント・レジスタ、ステータス・バイト・レジスタのクリア
*ESE	0～255	標準イベント・ステータス・イネーブルレジスタ(SESER)の書き込み
*ESE?	(0～255)	標準イベント・ステータス・イネーブルレジスタ(SESER)の読み出し
*ESR?	0～255	標準イベント・ステータス・レジスタ (SESR) の読み出しとクリア
*SRE	0～255	標準イベント・ステータス・イネーブルレジスタ (SRER) の書き込み
*SRE?	(0～255)	標準イベント・ステータス・イネーブルレジスタ (SRER) の読み出し
*STB?	(0～255)	ステータス・バイトと MSS ビットの読み出し
イベントスレジスタ		
:STATus:OPERation:CONDition?	(0～16176)	標準動作レジスタ・グループの条件レジスタの問い合わせ
:STATus:OPERation[:EVENT]?	(0～16176)	標準動作レジスタ・グループのイベント・レジスタの合計ビット数の問い合わせ
:STATus:OPERation:ENABle	0～65535	標準動作レジスタ・グループのイネーブル・レジスタの設定
:STATus:OPERation:ENABle?	(0～16177)	標準動作レジスタ・グループのイネーブル・レジスタの問い合わせ
:STATus:QUESTionable:CONDition?	(0～63497)	状態問い合わせレジスタ・グループの条件レジスタの問い合わせ
:STATus:QUESTionable[:EVENT]?	(0～63497)	状態問い合わせレジスタ・グループのイベント・レジスタの合計ビット数の問い合わせ
:STATus:QUESTionable:ENABle	0～65535	状態問い合わせレジスタ・グループのイネーブル・レジスタの設定
:STATus:QUESTionable:ENABle?	(0～63775)	状態問い合わせレジスタ・グループのイネーブル・レジスタの問い合わせ
測定値の読み出し		
:FETCh?	・:SYST:COMM:FORM FLOAT 設定時 <(測定値 1)>,<(測定値 2)>,...,<(測定値 N)> ・:SYST:COMM:FORM FIX 設定時 <(測定値)>	最後の測定値の読み出し
:READ?	・:SYST:COMM:FORM FLOAT 設定時 <(測定値 1)>,<(測定値 2)>,...,<(測定値 N)> ・:SYST:COMM:FORM FIX 設定時 <(測定値)>	測定 (トリガ待ちと測定値の読み出し)
:MEASure[:VOLTage]:DC?	<(測定値)>	レンジを指定して電圧測定
:MEASure:TEMPerature?	<(温度測定値)>	温度測定値の読み出し
:DATA:LAST?	<(測定値)>	最後の測定値の読み出し
:ABORt		測定の中断
:SYSTem:COMMunicate:FORMat	<FIX/FLOAT>	測定値の出力形式の設定
:SYSTem:COMMunicate:FORMat?	<(FIX/FLOAT)>	測定値の出力形式の問い合わせ
セルフテスト		
:TEST:ALL?	<(PASS/FAIL)>	セルフテストの実行と結果の問い合わせ

メッセージ []:省略可	データ []:省略可、0:応答データ	説明
------------------	------------------------	----

時計

:SYSTem:DATE	< 年 >< 月 >< 日 >	日付の設定
:SYSTem:DATE?	< 年 >< 月 >< 日 >	日付の問い合わせ
:SYSTem:TIME	< 時 >< 分 >< 秒 >	時計の設定
:SYSTem:TIME?	< 時 >< 分 >< 秒 >	時計の問い合わせ

測定レンジ

[.SENSe:]VOLTage:DC:RANGe	< 測定レンジ/MAX/MIN/DEfault >	測定レンジの設定
[.SENSe:]VOLTage:DC:RANGe?	< 測定レンジ >	測定レンジの問い合わせ
[.SENSe:]VOLTage:DC:RANGe:AUTO	< 1/0/ON/OFF >	測定オートレンジの設定と問い合わせ
[.SENSe:]VOLTage:DC:RANGe:AUTO?	< 1/0 >	測定オートレンジの問い合わせ

測定ファンクション

[.SENSe:]FUNCTION[ON]	< TEMPerature/VOLTage[DC] >	測定レンジの設定
[.SENSe:]FUNCTION[ON]?	("VOLT:DC")	測定レンジの問い合わせ

測定スピード

[.SENSe:]VOLTage[DC]:NPLCycles	< 積分時間(PLC)/MAX/MIN/DEfault/SLOW/MEDium/ FAST >	積分時間(PLC)の設定
[.SENSe:]VOLTage[DC]:NPLCycles?	< 積分時間(PLC) >	積分時間(PLC)の問い合わせ
[.SENSe:]VOLTage[DC]:APERture:ENABled	< 1/0/ON/OFF >	積分時間(sec)の有効設定
[.SENSe:]VOLTage[DC]:APERture:ENABled?	< 1/0 >	積分時間(sec)の有効設定の問い合わせ
[.SENSe:]VOLTage[DC]:APERture	< 積分時間(sec)/MAX/MIN/DEfault >	積分時間(sec)の設定
[.SENSe:]VOLTage[DC]:APERture?	< 積分時間(sec) >	積分時間(sec)の問い合わせ

トリガ

:INITiate:CONTinuous	< 1/0/ON/OFF >	連続測定の設定
:INITiate:CONTinuous?	< 1/0 >	連続測定の問い合わせ
:INITiate[IMMediate]		トリガ待ちの設定
:TRIGger:SOURce	< IMMEDIATE/ EXTeRnal/BUS >	トリガソースの設定
:TRIGger:SOURce?	< IMM/ EXT >	トリガソースと問い合わせ
:SAMPle:COUNt	< 測定回数/MAX/MIN/DEfault >	測定回数の設定
:SAMPle:COUNt?	< 測定回数 >	測定回数の問い合わせ
:TRIGger:DELAy	< デレイ時間(sec)/MAX/MIN/DEfault >	トリガデレイの設定
:TRIGger:DELAy?	< デレイ時間(sec) >	トリガデレイの問い合わせ
:TRIGger:DELAy:AUTO	< 1/0/ON/OFF >	トリガデレイのプリセット設定
:TRIGger:DELAy:AUTO?	< 1/0 >	トリガデレイのプリセット設定の問い合わせ

桁数設定

[.SENSe:]VOLTage:DIGits	< 桁数/MAX/MIN/DEfault >	電圧測定値桁数の設定
[.SENSe:]VOLTage:DIGits?	< 桁数 >	電圧測定値桁数の問い合わせ

ラベル表示

:SYSTem:LABel:STATe	< 1/0/ON/OFF >	ラベル表示機能の設定
:SYSTem:LABel:STATe?	< 1/0 >	ラベル表示機能の問い合わせ
:SYSTem:LABel	< ラベル名 >	ラベル名の設定
:SYSTem:LABel?	< ラベル名 >	ラベル名の問い合わせ

コンパレータ

:CALCulate:LIMit[:STATe]	< 1/0/ON/OFF >	コンパレータの実行
:CALCulate:LIMit[:STATe]?	< 1/0 >	コンパレータの問い合わせ
:CALCulate:LIMit:BEEPPer	< 条件 >< タイプ >< 回数 >	ブザーの設定
:CALCulate:LIMit:BEEPPer?	< 条件 > < 条件 >< タイプ >< 回数 >	ブザーの問い合わせ
:CALCulate:LIMit:ABSolute	< 1/0/ON/OFF >	絶対値判定機能の設定
:CALCulate:LIMit:ABSolute?	< 1/0 >	絶対値判定機能の問い合わせ
:CALCulate:LIMit:UPPer:ENABle	< 1/0/ON/OFF >	上限値有効の設定
:CALCulate:LIMit:UPPer:ENABle?	< 1/0 >	上限値有効の問い合わせ

メッセージ []:省略可	データ []:省略可、0:応答データ	説明
:CALCulate:LIMit:UPPer[:DATA]	< 上限値 >	上限値の設定
:CALCulate:LIMit:UPPer[:DATA]?	< 上限値 >	上限値の問い合わせ
:CALCulate:LIMit:LOWer:ENABle	<1/0/ON/OFF>	下限値有効の設定
:CALCulate:LIMit:LOWer:ENABle?	< 1/0 >	下限値有効の問い合わせ
:CALCulate:LIMit:LOWer[:DATA]	< 下限値 >	下限値の設定
:CALCulate:LIMit:LOWer[:DATA]?	< 下限値 >	下限値の問い合わせ
:CALCulate:LIMit:DELay	<1/0/ON/OFF>	コンパレータの判定継続機能の設定
:CALCulate:LIMit:DELay?	< 1/0 >	コンパレータの判定継続機能の問い合わせ
:CALCulate:LIMit:DELay:COUNt	< 回数 >	コンパレータの判定継続回数の設定
:CALCulate:LIMit:DELay:COUNt?	< 回数 >	コンパレータの判定継続回数の問い合わせ
:CALCulate:LIMit:CLEar[:IMMediate]		コンパレータイベントステータスレジスタクリア
:CALCulate:LIMit:RESult?	<HL/TN/LO/ERR/OFF>	コンパレータ判定結果の問い合わせ

BIN

:CALCulate:BIN[:STATe]	<1/0/ON/OFF>	BIN 測定の実行
:CALCulate:BIN[:STATe]?	< 1/0 >	BIN 測定の問い合わせ
:CALCulate:BIN:ENABle	< 使用可能パターン >	使用可能パターンの設定
:CALCulate:BIN:ENABle?	< 使用可能パターン >	使用可能パターンの問い合わせ
:CALCulate:BIN:UPPer	<BIN No>,< 上限値 >	上限値の設定
:CALCulate:BIN:UPPer?	<BIN No> < 上限値 >	上限値の問い合わせ
:CALCulate:BIN:LOWer	<BIN No>,< 下限値 >	下限値の設定
:CALCulate:BIN:LOWer?	<BIN No> < 下限値 >	下限値の問い合わせ
:CALCulate:BIN:RESult?	(0~1024)	BIN 判定結果の問い合わせ

測定条件の保存と読み込み

*SAV	<パネル番号>	測定条件の保存(パネルセーブ)
*RCL	<パネル番号>	測定条件の読み込み(パネルロード)
:SYSTem:PANel:CLEar	<パネル番号>	パネルの削除
:SYSTem:PANel:NAME	<パネル番号>,<パネル名>	パネル名の設定
:SYSTem:PANel:NAME?	<パネル番号> <パネル番号>,<パネル名>	パネル名の問い合わせ
:SYSTem:PANel:DATE?	<パネル番号> < 西暦, 月, 日, 時, 分, 秒 >	パネルの保存日時の問い合わせ
:MMEMory:STORe:STATe	< ファイル名 >	設定ファイルの USB メモリへの保存
:MMEMory:LOAD:STATe	< ファイル名 >	設定ファイルの USB メモリからの読み込み
:MMEMory:STATe:RECall:AUTO	<1/0/ON/OFF>	起動時にパネルを読み込む機能の設定
:MMEMory:STATe:RECall:AUTO?	< 1/0 >	起動時にパネルを読み込む機能の問い合わせ
:MMEMory:STATe:RECall:SELEct	< パネル番号 >	起動時に読み込むパネル番号の設定
:MMEMory:STATe:RECall:SELEct?	< パネル番号 >	起動時に読み込むパネル番号の問い合わせ

スレージング

:CALCulate:SMOothing[:STATe]	<1/0/ON/OFF>	スレージング機能の実行
:CALCulate:SMOothing[:STATe]?	< 1/0 >	スレージング機能の問い合わせ
:CALCulate:SMOothing:RESPonse	< 回数/SLOW/MEDium/FAST >	スレージング回数の設定
:CALCulate:SMOothing:RESPonse?	< 回数 >	スレージング回数の問い合わせ

ホールド

[.SENSe:]HOLD:AUTO	<1/0/ON/OFF>	オートホールドの実行
[.SENSe:]HOLD:AUTO?	< 1/0 >	オートホールドの問い合わせ
[.SENSe:]HOLD:BOUND	< ホールド範囲/MAX/MIN/DEFault >	オートホールド範囲の設定
[.SENSe:]HOLD:BOUND?	< ホールド範囲 >	オートホールド範囲の問い合わせ

コンタクトチェック

[.SENSe:]VOLTage:DC:CONtact:CAPacitance?	< コンタクトチェックの測定値 >	コンタクトチェックの測定値の問い合わせ
[.SENSe:]VOLTage:DC:CONtact:CAPacitance:STATe	<1/0/ON/OFF>	コンタクトチェックの実行
[.SENSe:]VOLTage:DC:CONtact:CAPacitance:STATe?	< 1/0 >	コンタクトチェックの問い合わせ

メッセージ []:省略可	データ []:省略可、0:応答データ	説明
[SENSe:]VOLTage:DC:CONtact:CAPacitance:THReshold	< 閾値/MAX/MIN/Default >	コンタクトチェックの閾値の設定
[SENSe:]VOLTage:DC:CONtact:CAPacitance:THReshold?	< 閾値 >	コンタクトチェックの閾値の問い合わせ
[SENSe:]VOLTage:DC:CONtact:CAPacitance:TIME	< 積分時間(sec)/MAX/MIN/Default >	コンタクトチェックの積分時間の設定
[SENSe:]VOLTage:DC:CONtact:CAPacitance:TIME?	< 積分時間(sec) >	コンタクトチェックの積分時間の問い合わせ

入力抵抗の切替

[SENSe:]VOLTage[:DC]:iMPedance:AUTO	< 1/0/ON/OFF >	入力抵抗の設定
[SENSe:]VOLTage[:DC]:iMPedance:AUTO?	< 1/0 >	入力抵抗の問い合わせ

ゼロ点調節(NULL 機能)

[SENSe:]VOLTage:DC:NULL[:STATe]	< 1/0/ON/OFF >	ゼロ点調節の実行
[SENSe:]VOLTage:DC:NULL[:STATe]?	< 1/0 >	ゼロ点調節の問い合わせ
[SENSe:]VOLTage:DC:NULL:VALue	< 調節値(V)/MAX/MIN/Default >	ゼロ点調節値の設定
[SENSe:]VOLTage:DC:NULL:VALue?	< 調節値(V) >	ゼロ点調節値の問い合わせ

温度補正(TC)

:CALCulate:TCORrect:STATe	< 1/0/ON/OFF >	温度補正(TC)の実行
:CALCulate:TCORrect:STATe?	< 1/0 >	温度補正(TC)の問い合わせ
:CALCulate:TCORrect:PARAmeter	< 基準温度(°C)/MAX/MIN/Default >, < 温度係数(ppm/°C)/MAX/MIN/Default >	温度補正(TC)の設定
:CALCulate:TCORrect:PARAmeter?	< 基準温度(°C)>, < 温度係数(ppm/°C)>	温度補正(TC)の問い合わせ

スケーリング

:CALCulate:SCALE[:STATe]	< 1/0/ON/OFF >	スケーリング機能の実行
:CALCulate:SCALE[:STATe]?	< 1/0 >	スケーリング機能の問い合わせ
:CALCulate:SCALE:PARAmeterA	< 補正係数/MAX/MIN/Default >	スケーリング補正係数の設定
:CALCulate:SCALE:PARAmeterA?	< 補正係数 >	スケーリング補正係数の問い合わせ
:CALCulate:SCALE:PARAmeterB	< オフセット/MAX/MIN/Default >	スケーリングオフセットの設定
:CALCulate:SCALE:PARAmeterB?	< オフセット >	スケーリングオフセットの問い合わせ
:CALCulate:SCALE:UNIT:STATe	< 1/0/ON/OFF >	スケーリング単位機能の設定
:CALCulate:SCALE:UNIT:STATe?	< 1/0 >	スケーリング単位機能の問い合わせ
:CALCulate:SCALE:UNIT	< 単位 >	スケーリング単位の設定
:CALCulate:SCALE:UNIT?	< 単位 >	スケーリング単位の問い合わせ

統計

:CALCulate:AVERAge[:STATe]	< 1/0/ON/OFF >	統計演算機能の実行
:CALCulate:AVERAge[:STATe]?	(1)	統計演算機能の問い合わせ
:CALCulate:AVERAge:CLear[:IMMediate]		統計演算結果のクリア
:CALCulate:AVERAge:CLear:PRINt	< 1/0/ON/OFF >	プリントアウト時に 統計演算結果をクリアする機能の設定
:CALCulate:AVERAge:CLear:PRINt?	< 1/0 >	プリントアウト時に 統計演算結果をクリアする機能の問い合わせ
:CALCulate:AVERAge:COUNT?	< データ数 >	統計演算データ数の問い合わせ
:CALCulate:AVERAge:ALL?	< 平均値 >,< 標準偏差 >,< 最小値 >,< 最大値 >	平均値、標準偏差、最小値、最大値の問い合わせ
:CALCulate:AVERAge:SDEViation?	< 標準偏差 >	標準偏差の問い合わせ
:CALCulate:AVERAge:AVERAge?	< 平均値 >	平均値の問い合わせ
:CALCulate:AVERAge:MINimum?	< 最小値 >	最小値の問い合わせ
:CALCulate:AVERAge:MAXimum?	< 最大値 >	最大値の問い合わせ
:CALCulate:AVERAge:PTPeak?	< Peak to Peak 値 >	Peak to Peak 値の問い合わせ
:CALCulate:AVERAge:LIMit?	< Hi の数 >,< IN の数 >,< Lo の数 >,< 測定異常の数 >, < レンジオーバーの数 >	コンパレータ結果の問い合わせ
:CALCulate:AVERAge:BIN?	< BIN0 の数 >,...,< BIN9 の数 >,< OUT の数 >,< 測定異常の数 >	BIN 結果の問い合わせ
:CALCulate:AVERAge:CP?	< Cp >,< Cpk >	工程能力指数の問い合わせ

キーロック

:SYSTem:KLOCK	< 1/0/ON/OFF >	キーロック状態の設定
:SYSTem:KLOCK?	< 1/0 >	キーロック状態の問い合わせ

メッセージ []:省略可	データ []:省略可、0:応答データ	説明
------------------	------------------------	----

音設定

:SYSTem:CLICk[:STATe]	<1/0/ON/OFF>	キークリック音の設定
:SYSTem:CLICk[:STATe]?	<1/0>	キークリック音の問い合わせ
:SYSTem:HOLD:BEEPer:STATe	<1/0/ON/OFF>	ホールド音の設定
:SYSTem:HOLD:BEEPer:STATe?	<1/0>	ホールド音の問い合わせ
:SYSTem:BEEPer:STATe	<1/0/ON/OFF>	エラービープ音の設定
:SYSTem:BEEPer:STATe?	<1/0>	エラービープ音の問い合わせ
:SYSTem:BEEPer:VOLume	<音量/MAX/MIN/DEfault>	音量の設定
:SYSTem:BEEPer:VOLume?	<音量>	音量の問い合わせ
:SYSTem:BEEPer[:IMMediate]		ビープ音を一回鳴らす

表示設定

:DISPlay[:STATe]	<1/0/ON/OFF>	LCD 表示機能の設定
:DISPlay[:STATe]?	(1)	LCD 表示機能の問い合わせ
:DISPlay:BACKlight	<輝度/MAX/MIN/DEfault>	画面の輝度設定
:DISPlay:BACKlight?	<輝度>	画面の輝度設定の問い合わせ
:DISPlay:TYPe	<画面色タイプ/MAX/MIN/DEfault>	画面色の設定
:DISPlay:TYPe?	<画面色タイプ>	画面色の問い合わせ
:DISPlay:VIEW	<NUMeric /TCHart/METer/STATistics/HISTogram>	表示タイプの設定
:DISPlay:VIEW?	<NUM/TCH/MET/STAT/COMP>	表示タイプの問い合わせ

電源周波数

:SYSTem:LFRequency	<AUTO/50/60>	電源周波数の設定
:SYSTem:LFRequency?	<AUTO/50/60>	電源周波数の問い合わせ

システムリセット

:SYSTem:PRESet		システムリセットの実行
:STATus:PRESet		

通信設定

:SYSTem:LOCal		ローカル状態へ戻る
:SYSTem:COMMunicate	<USB/LAN/RS232c/KEYBD/PRINter>	通信インターフェースの設定
:SYSTem:COMMunicate?	<USB/LAN/RS232c/KEYBD/PRINter>	通信インターフェースの問い合わせ
:SYSTem:COMMunicate:LOGGing	<1/0/ON/OFF>	コマンドロギングの実行
:SYSTem:COMMunicate:LOGGing?	<1/0>	コマンドロギングの問い合わせ
:SYSTem:COMMunicate:MONitor	<1/0/ON/OFF>	コマンドモニタの実行
:SYSTem:COMMunicate:MONitor?	<1/0>	コマンドモニタの問い合わせ
:SYSTem:COMMunicate:GPIB:ADDRess	<アドレス>	GPIB のアドレスの設定
:SYSTem:COMMunicate:GPIB:ADDRess?	<アドレス>	GPIB のアドレスの問い合わせ
:SYSTem:COMMunicate:GPIB:TERMinator	<0/1>	GPIB のデリミタの設定
:SYSTem:COMMunicate:GPIB:TERMinator?	<0/1>	GPIB のデリミタの問い合わせ
:SYSTem:COMMunicate:RS232C:SPEED	<ボーレート>	ボーレートの設定
:SYSTem:COMMunicate:RS232C:SPEED?	<ボーレート>	ボーレートの問い合わせ
:SYSTem:COMMunicate:LAN:IPADdress	<IP アドレス>	IP アドレスの設定
:SYSTem:COMMunicate:LAN:IPADdress?	<IP アドレス>	IP アドレスの問い合わせ
:SYSTem:COMMunicate:LAN:CONTRol	<ポート No.>	LAN ポートの設定
:SYSTem:COMMunicate:LAN:CONTRol?	<ポート No.>	LAN ポートの問い合わせ
:SYSTem:COMMunicate:LAN:SMASK	<サブネットマスク>	サブネットマスクの設定
:SYSTem:COMMunicate:LAN:SMASK?	<サブネットマスク>	サブネットマスクの問い合わせ
:SYSTem:COMMunicate:LAN:GATeway	<アドレス>	デフォルト・ゲートウェイの設定
:SYSTem:COMMunicate:LAN:GATeway?	<アドレス>	デフォルト・ゲートウェイの問い合わせ
:SYSTem:COMMunicate:LAN:UPDate		LAN 設定の反映
:SYSTem[:COMMunicate:LAN]:MAC?	<MAC アドレス>	MAC アドレスの問い合わせ

メッセージ []: 省略可	データ []: 省略可、0: 応答データ	説明
-------------------	--------------------------	----

データ出力設定

:SYSTem:COMMunicate:DATAout	<1/0/ON/OFF>	測定同期データ出力機能の設定
:SYSTem:COMMunicate:DATAout?	<1/0>	測定同期データ出力機能の問い合わせ
:SYSTem:COMMunicate:DATAout:FORMat	< 測定値出力 >, < 日付出力 >	測定データ出力形式の設定
:SYSTem:COMMunicate:DATAout:FORMat?	< 測定値出力 >, < 日付出力 >	測定データ出力形式の問い合わせ
:SYSTem:COMMunicate:DATAout:CONDition	< 出力条件 >	測定データ出力条件の設定
:SYSTem:COMMunicate:DATAout:CONDition?	< 出力条件 >	測定データ出力条件の問い合わせ
:SYSTem:LOCALe:DATE	< 日付形式 >	日付形式の設定
:SYSTem:LOCALe:DATE?	< 日付形式 >	日付形式の問い合わせ
:SYSTem:LOCALe:DATE:SEParator	< 区切り形式 >	日付区切り形式の設定
:SYSTem:LOCALe:DATE:SEParator?	< 区切り形式 >	日付区切り形式の問い合わせ
:SYSTem:LOCALe:NUMeric:DOT	< 文字形式 >	小数点文字形式の設定
:SYSTem:LOCALe:NUMeric:DOT?	< 文字形式 >	小数点文字形式の問い合わせ
:SYSTem:LOCALe:NUMeric:SEParator	< 区切り形式 >	CSV ファイル区切り形式の設定
:SYSTem:LOCALe:NUMeric:SEParator?	< 区切り形式 >	CSV ファイル区切り形式の問い合わせ

メモリ機能

:R?	< 読み込みデータ数 > (<ヘッダ><測定値><測定値>...<測定値>)	メモリデータの問い合わせ
:DATA:REMOve?	< 読み込みデータ数 > [WAIT] <測定値><測定値>...<測定値>	メモリデータの問い合わせ(待機機能付き)
:DATA:POINts:EVENT:THReShold	< カウント数 >	イベントを発生させるメモリデータ数の設定
:DATA:POINts:EVENT:THReShold?	< カウント数 >	イベントを発生させるメモリデータ数の問い合わせ
:DATA:POINts?	< メモリデータ数 >	メモリ個数の問い合わせ
:DATA:CLEAr		メモリデータのクリア
:MMEMory:STORe:DATA	< ファイル名 >	メモリデータをファイルに保存

EXT I/O

:IO:MODE?	< NPN/PNP >	NPN/PNP スイッチ状態の問い合わせ
:IO:FILTer:STATe	<1/0/ON/OFF>	TRIG/PRINT 信号のフィルタ機能実行
:IO:FILTer:STATe?	<1/0>	TRIG/PRINT 信号のフィルタ機能の問い合わせ
:IO:FILTer:TIME	< フィルタ時間(sec) >	TRIG/PRINT 信号のフィルタ時間の設定
:IO:FILTer:TIME?	< フィルタ時間(sec) >	TRIG/PRINT 信号のフィルタ時間の問い合わせ
:IO:EOM:MODE	<1/0/ON/OFF>	EOM 出力方法の設定
:IO:EOM:MODE?	<1/0>	EOM 出力方法の問い合わせ
:IO:EOM:PULSe	< パルス幅(sec) >	EOM パルス幅の設定
:IO:EOM:PULSe?	< パルス幅(sec) >	EOM パルス幅の問い合わせ
:IO:INPut?	< 入力データ >	外部 I/O 入力の問い合わせ
:IO:OUTPut	< 出力データ >	外部 I/O 出力の実行

画面データの保存

:HCOPy:SDUMp:DATA?	(BMP バイナリデータ)	表示イメージの取得
--------------------	---------------	-----------

***IDN?の応答設定**

:SYSTem:IDNStr	< 文字列 >	*IDN?の応答設定
----------------	---------	------------

他社製品との互換コマンド

:CALCulate:SCALe:GAIN	< 補正係数/MAX/MIN/DEfult >	スケーリング補正係数の設定
:CALCulate:SCALe:GAIN?	< 補正係数 >	スケーリング補正係数の問い合わせ
:CALCulate:SCALe:OFFSet	< オフセット/MAX/MIN/DEfult >	スケーリングオフセットの設定
:CALCulate:SCALe:OFFSet?	< オフセット >	スケーリングオフセットの問い合わせ
:SYSTem:ERRor[:NEXT]?	< エラーNo. >, < エラー名 >	エラー情報の読み出し
:MEASure:TEMPerature?	< FRTD/RTD/FTH/THER/DEfult >, < タイプ 2/DEfult >, < 分解能/MAX/MIN/DEfult > < 測定値 >	温度測定値の読み出し

メッセージ []:省略可	データ []:省略可、0:応答データ	説明
*PSC	<0/1>	電源投入ステータスのクリア
*PSC?	(1)	電源投入ステータスの読み出し
*CAL?	(+0)	自動校正の実行
:HCOPy:SDUMp:DATA:FORMat	BMP	画像データ形式の設定
:HCOPy:SDUMp:DATA:FORMat?	(BMP)	画像データ形式の問い合わせ
[.:SENSe:]VOLTage[:DC]:ZERO:AUTO	< 1/0/ON/OFF/ONCE >	オートゼロ・モードの設定
[.:SENSe:]VOLTage[:DC]:ZERO:AUTO?	(1)	オートゼロ・モードの問い合わせ
:OUTPut:TRIGger:SLOPe	<POSitive/NEGative>	電圧計完了出力信号のスロープ設定
:OUTPut:TRIGger:SLOPe?	(NEG)	電圧計完了出力信号のスロープの問い合わせ
:TRIGger:COUNt	< 測定回数/MAX/MIN/DEfault >	トリガ回数の設定
:TRIGger:COUNt?	(1)	トリガ回数の問い合わせ
:SYSTem:REMOte		リモート状態に設定
:SYSTem:RWLock		

3 メッセージリファレンス

メッセージリファレンスの見方

< >: メッセージのデータ部(文字または数値パラメータ)の内容を示します。文字パラメータの場合、応答は大文字で返します。

数値パラメータ:

- NRf NR1、NR2、NR3 すべてを含む形式
- NR1 整数データ (例: +12、-23、34)
- NR2 小数データ (例: +1.23、-23.45、3.456)
- NR3 浮動小数点指数表示データ (例: +1.0E-2、-2.3E+4)

コマンドの内容を示します

メッセージの構文を記述します。
コマンドのデータ部または
応答メッセージの解説をします。
メッセージの解説をします。

実際のコマンド使用例を示します

標準イベント・ステータス・イネーブルレジスタ(SESER)の書き込みと読み出し

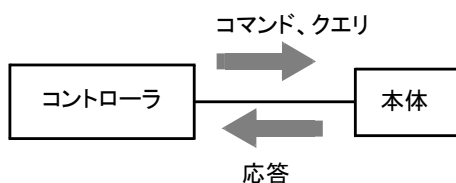
構文 コマンド ***ESE <0 ~ 255 (NR1)>**
クエリ ***ESE?**
応答 **<0 ~ 255 (NR1)>**

説明 コマンド SESER のマスクパターンを、0~255の数値で設定します。初期値(電源投入時)は0です。

クエリ *ESEコマンドで設定したSESERの内容を、0~255のNR1数値で返します。

128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

例 ***ESE 36**
(SESERのビット5とビット2をセットします)



共通コマンド

(1) システム・データ・コマンド

機器の ID(識別コード) の問い合わせ

構文	クエリ	*IDN?
応答	< メーカー名 >,< モデル名 >,< 製造番号>,< ソフトウェアバージョン >	
例	*IDN? HIOKI,DM7275-01,123456789,V1.00 機器IDは、HIOKI、DM7275-01、123456789、ソフトウェアバージョン1.00です。< モデル名 >は下記のようになります。 DM7275-01の場合: DM7275-01 DM7275-02の場合: DM7275-02 DM7275-03の場合: DM7275-03 DM7276-01の場合: DM7276-01 DM7276-02の場合: DM7276-02 DM7276-03の場合: DM7276-03	

注 記 :SYSTem:IDNStrにより返り値を任意の文字列に変更することが可能です。

インストールされているオプションを識別

構文	クエリ	*OPT?
応答	<0/GPIB>,<LAN>,<0/RS232C>	
例	・BP-IBボードを装着している場合 *OPT? GPIB,LAN,0 ・RS-232Cボードを装着している場合 *OPT? 0,LAN,RS232C ・オプションボードを装着していない場合 *OPT? 0,LAN,0	

(2) 内部動作コマンド

機器の初期化

構文	コマンド	*RST
説明	機器を既定の設定にします。	
注 記	通信条件、パネルデータは初期化されません。 イベントステータスレジスタは、クリアされません。	

セルフテストの実行と結果の問い合わせ

構文	クエリ	*TST?
応答	<PASS/FAIL>	
説明	本体のセルフテストを行い、その結果を返します。 エラーなしの場合はPASS、エラーが発生した場合はFAILを返します。	
例	*TST? FAIL エラーが発生しています。正しく測定できない可能性がありますので、使用を中止し、修理に出してください。	

測定条件の保存(パネルセーブ)

- 構文** コマンド ***SAV < パネル番号 >**
 < パネル番号 > = 1~30 (NR1)
- 説明** 指定したパネル番号に現在の測定条件を保存します。
- 例** ***SAV 1**
 現在の測定条件をパネル番号1に保存します。
- 注記** 測定条件が既に保存されているパネル番号を指定した場合は、現在の測定条件で上書きされます。

測定条件の読み込み(パネルロード)

- 構文** コマンド ***RCL < パネル番号 >**
 < パネル番号 > = 1~30 (NR1)
- 説明** 指定したパネル番号に保存された測定条件を読み出します。
- 例** ***RCL 1**
 パネル番号1に保存された測定条件を読み出します。
- 注記** 測定条件が保存されていないパネル番号を指定した場合、実行エラーになります。

サンプリングの要求

- 構文** コマンド ***TRG**
- 説明** 外部トリガ(トリガソース<EXTERNAL>)のときに 1 回測定を行います。
 レンジ切り替えやパネルロード後に測定値が安定するまでウェイトが必要です。ウェイト時間は測定対象によって異なります。
- 例** **:TRIG:SOUR EXT;*TRG**
 外部トリガ設定にし、トリガをかけます。

(3) 同期コマンド

実行中の全動作終了後、SESRのOPCをセット

- 構文** コマンド ***OPC**
- 説明** 送信されたコマンドのうち、***OPC**コマンドより前のコマンド処理が終了した時点で、SESR(標準イベント・ステータス・レジスタ)のOPC(ビット0)を1にセットします

実行中の全動作終了後、ASCIIの1を応答

- 構文** クエリ ***OPC?**
 応答 **1 (NR1)**
- 説明** 送信されたコマンドのうち、***OPC**コマンドより前のコマンド処理が終了した時点で、ASCIIの1を応答します。

コマンド処理終了後、続くコマンドを実行

- 構文** コマンド ***WAI**
- 説明** 前のコマンド動作が全て終了するまで本体を待機させます。

(4) ステータス、イベント制御コマンド

イベント・レジスタ、ステータス・バイト・レジスタのクリア (出力キューを除く)

構文	コマンド	*CLS
	説明	イベント・ステータス・レジスタをクリアします。イベント・レジスタに対応したステータス・バイト・レジスタのビットもクリアされます。(標準イベント・ステータス・レジスタ、標準動作レジスタ、状態問い合わせレジスタ)
注記	[RS-232C/USB]	出力キューは影響されません。
	[GP-IB]	出力キュー、各種イネーブル・レジスタ、ステータスバイトのMAV(ビット4)は影響されません。

標準イベント・ステータス・イネーブルレジスタ(SESER)の書込みと読み出し

構文	コマンド	*ESE <0~255 (NR1)>
	クエリ	*ESE?
説明	応答	<0~255 (NR1)>
	コマンド	SESERのマスクパターンを、0~255の数値で設定します。初期値(電源投入時)は0です。
	クエリ	*ESE コマンドで設定したSESERの内容を、0~255のNR1数値で返します。

128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

例 ***ESE 36**
SESERのビット5とビット2をセットします。

標準イベント・ステータス・レジスタ (SESR) の読み出しとクリア

構文	クエリ	*ESR?
	応答	<0~255 (NR1)>
説明	SESRの内容を0~255のNR1数値で返し、その内容をクリアします。	

[RS-232C/USB]							
128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PON	未使用	CME	EXE	DDE	QYE	未使用	OPC

[GP-IB]							
128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PON	URQ	CME	EXE	DDE	未使用	RQC	OPC

例 ***ESR?**
32
SESRのbit5が1になっています。

サービスリクエスト・イネーブルレジスタ (SRER) の書き込みと読み出し

構文 コマンド ***SRE** <0~255 (NR1)>
 クエリ ***SRE?**
 応答 <0~255 (NR1)>

説明 コマンド SRERのマスクパターンを、0 ~ 255の数値で設定します。
 数値はNRf 形式で受け付けますが、小数点以下は四捨五入して扱います。
 ビット6、未使用(-表記)ビットの値は無視されます。電源投入時は、0に初期化します。

 クエリ ***SRE**コマンドで設定したSRERの内容を0~255のNR1 数値で返します。ビット6、未使用ビット(ビット0、1)の値は常に0です。

128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
標準動作レジスタ	0	ESB	MAV	状態問合せレジスタ	システムエラー	-	-

例 ***SRE**
 12
 SRERのビット3とビット2を1にします。

***SRE?**
 12
 SRERのビット3とビット2が1になっています。

ステータス・バイトと MSS ビットの読み出し

構文 クエリ ***STB?**
 応答 <0~ 255(NR1)>

説明 STBの設定内容を0~255のNR1数値で返します。

128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ESB1	MSS	ESB	MAV	ESB0	ERR	-	-

例 ***STB?**
 16
 STBのbit4が1になっています。

固有コマンド

(1) イベント・ステータス・レジスタ

ステータス・バイト・レジスタとの関係については、下記をご確認下さい。

- ステータス・バイト・レジスタ(p.6)
- イベント・レジスタ(p.8)

標準動作レジスタ・グループの条件レジスタの問い合わせ

構文 クエリ **:STATus:OPERation:CONDition?**
 応答 <0 ~ 16176 (NR1)>

説明

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
-	-	ERR	HOLD	EOM	LOCK	MEM_RDY	SET

例

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	WAIT_TRG	MEAS	-	-	-	-

:STAT:OPER:COND?

1024

本体はリモート状態です。

注記

- 本コマンドを受け付けた時点において最新の状態を返します。
- **:STATus:OPERation:EVENT?**、**:STATus:OPERation?**と異なり、各ビット情報は保持されません。

標準動作レジスタ・グループのイベント・レジスタの合計ビット数の問い合わせ

構文 クエリ **:STATus:OPERation[:EVENT?]**
 応答 <0 ~ 16176 (NR1)>

説明

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
-	-	ERR	HOLD	EOM	LOCK	MEM_RDY	SET

例

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	WAIT_TRG	MEAS	-	-	-	-

:STAT:OPER?

1024

本体はリモート状態になったことがあります。

注記

- 各イベントが発生した際に、各イベントに対応したビットに1がセットされます。
- ***CLS**の実行、本クエリの実行、または電源の再投入を行うまでは、そのビットはクリアされません。

標準動作レジスタ・グループのイネーブル・レジスタの設定と問い合わせ

構文 コマンド **:STATus:OPERation:ENABLE <0 ~ 65535 (NR1)>**
 クエリ **:STATus:OPERation:ENABLE?**
 応答 **<0 ~ 16177 (NR1)>**

説明

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
-	-	ERR	HOLD	EOM	LOCK	MEM_ _RDY	SET

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	WAIT _TRG	MEAS	-	-	-	RSV

注記

- 電源投入の時はデータを 0 に初期化します。
- 未使用(-表記)ビットを1に設定した場合、コマンドは受け付けますが、クエリの結果には反映されません。

状態問い合わせレジスタ・グループの条件レジスタの合計ビット数の問い合わせ

構文 クエリ **:STATus:QUEStionable:CONDition?**
 応答 **<0 ~ 63497 (NR1)>**

説明

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
CNT_ ERR	MEM_ OVR	OB	HI	LO	-	-	-

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	-	-	-	-	-	-

例

:STAT:QUES:COND?
32768

コンタクトエラーが発生しています。

注記

- 本コマンドを受け付けた時点において最新の状態を返します。
- :STATus:QUEStionable:EVENT?**、**:STATus:QUEStionable?**と異なり、各ビット情報は保持されません。

状態問い合わせレジスタ・グループのイベント・レジスタの問い合わせ

構文 クエリ **:STATus:QUEStionable[:EVENT]?**
 応答 **<0 ~ 63497 (NR1)>**

説明

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
CNT_ ERR	MEM_ OVR	OB	HI	LO	-	-	-

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	-	-	TMP_ OVR	-	-	VLT_ OVR

例

:STAT:QUES?
1

本体の電圧測定において、オーバーレンジが発生したことがあります。

注記

各イベントが発生した際に、各イベントに対応したビットに1がセットされます。***CLS**の実行、本クエリの実行、または電源の再投入を行うまでは、そのビットはクリアされません。

状態問い合わせレジスタ・グループのイネーブル・レジスタの設定と問い合わせ

構 文 コマンド :STATus:QUEStionable:ENABLE <0 ~ 65535 (NR1)>
 クエリ :STATus:QUEStionable:ENABLE?
 応答 <0 ~ 63775 (NR1)>

説明

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
CNT_ ERR	MEM_ OVR	OB	HI	LO	-	-	RSV

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	-	RSV	TMP_ OVR	RSV	RSV	VLT_ OVR

- 注 記
- 電源投入の時はデータを 0 に初期化します。
 - 未使用(-表記)ビットを1に設定した場合、コマンドは受け付けますが、クエリの結果には反映されません。

(2) 測定値の読み出し

測定値のフォーマット

(a) :FETCh?, :READ?, MEASure[:VOLTage]:DC? の応答

- 電圧測定値: 単位V(:SYSTem:COMMunicate:FORMat FIX 設定時)

測定値レンジ	測定値	±OvrRng	表示時	測定異常時
100mV	± 000.00000E-03	±990.00000E+35		+991.00000E+35
1V	± 0000.00000E-03	±9900.00000E+34		+9910.00000E+34
10V	± 00.0000000E+00	±99.0000000E+36		+99.1000000E+36
100V	± 000.0000000E+00	±990.0000000E+35		+991.0000000E+35
1000V	± 0000.0000000E+00	±9900.0000000E+34		+9910.0000000E+34

注記: 小数点の位置や指数部は、スケーリングのゲインにより変わります。

スケーリングについては本体の取扱説明書をご覧ください。

・ 整数部の桁数が足りない場合は、0が入ります。

例) 1000Vレンジで測定値が1Vの場合、+0001.0000E+00となります。

・ 仮数部は桁数設定に応じて変化します。

- 電圧測定値: 単位V(:SYSTem:COMMunicate:FORMat FLOAT 設定時)

測定値	±OvrRng	表示時	測定異常時
± 0.000000000E±00	±9.900000000E+37		+9.910000000E+37

注記: 測定値は小数点以下8桁の浮動小数値で表現されます。

- 温度測定値: 単位°C(:FETCh? TEMPerature, READ? TEMPerature)

測定値	±OvrRng	表示時	測定異常時
± 00.00	±9.900E+37		+9.910E+37

:FETCh? と :READ?/MEAS:DC? で、測定値を取得するタイミングが異なります。

参照: データ取得方法(p.71)、トリガ(p.40)

(b) :DATA:LAST? の応答

- 電圧測定値: 単位V

測定値	±OvrRng	表示時	測定異常時
± 0.000000000E±00	±9.900000000E+37		+9.910000000E+37

注記: 測定値は小数点以下8桁の浮動小数値で表現されます。

(c) :MEASure:TEMPerature? の応答

- 温度測定値: 単位°C

測定値	±OvrRng	表示時	測定異常時
± 00.00	±9.900E+37		+9.910E+37

最後の測定値の読み出し

構文 クエリ
応答

:FETCh? [TEMPerature]

(1) TEMPerature 省略時

- :SYST:COMM:FORM FIX 設定時
< 電圧測定値 >

- :SYST:COMM:FORM FLOAT 設定時
< 電圧測定値1 >, < 電圧測定値2 >, ..., < 電圧測定値n >
※測定値は、取得された順番に出力されます。

(2) TEMPerature 指定時

- :SYST:COMM:FORM FIX 設定時
< 電圧測定値 >, < 温度測定値 >

- :SYST:COMM:FORM FLOAT 設定時
< 電圧測定値1 >, < 温度測定値1 >, < 電圧測定値2 >, < 温度測定値2 >, ..., < 電圧測定値n >, < 温度測定値n >
※測定値は、取得された順番に出力されます。

参照：「測定値のフォーマット」(p.32)

説明

- :SYST:COMM:FORM FIX 設定時
最後 (直近) の測定値を読み出します。トリガはしません。
- :SYST:COMM:FORM FLOAT 設定時
最後 (直近) の測定値をメモリに保存された個数分、読み出します。トリガはしません。

参照：データ取得方法(p.71)、トリガ(p.40)

例

```
:TRIG:SOUR EXT
:SYST:COMM:FORM FIX::SAMP:COUN 4
:INIT
*TRG
:FETC?
+094.31342E-03
:INIT
*TRG
:FETC?
+094.30276E-03
:FETC? TEMP
+094.30276E-03,+23.87

:SYST:COMM:FORM FLOAT
:INIT
*TRG
:FETC?
+9.46651535E-02,+9.44042868E-02,+9.42089137E-02,+9.42364743E-02
:INIT
*TRG
:FETC?
+9.46651535E-02,+9.44042868E-02,+9.42089137E-02,+9.42364743E-02,+9.46651536E-02,+9.44042867E-02,+9.42089132E-02,+9.42364741E-02
:FETC? TEMP
+9.46651535E-02,+23.80,+9.44042868E-02,+23.81,+9.42089137E-02,+23.82,+9.42364743E-02,+23.83,+9.46651536E-02,+23.84,+9.44042867E-02,+23.85,+9.42089132E-02,+23.86,+9.42364741E-02,+23.87
```

注記

- 未測定(測定値メモリが空)の場合は、実行エラーになります。
- :SYST:COMM:FORM FLOAT時に返される測定値のデータ数は、内部メモリに保存されている測定データの数(最大5000データ)となります。

測定（トリガ待ちと測定値の読み出し）

- 構文 クエリ **:READ?** [TEMPerature]
- 応答
- (1) TEMPerature省略時
 - :SYST:COMM:FORM FIX 設定時
 < 電圧測定値 >
 - :SYST:COMM:FORM FLOAT 設定時
 < 電圧測定値1 >, < 電圧測定値2 >, ..., < 電圧測定値n >
 ※測定値は、取得された順番に出力されます。
 - (2) TEMPerature指定時
 - :SYST:COMM:FORM FIX 設定時
 < 電圧測定値 >, < 温度測定値 >
 - :SYST:COMM:FORM FLOAT 設定時
 < 電圧測定値1 >, < 温度測定値1 >, < 電圧測定値2 >, < 温度測定値2 >, ..., < 電圧測定値N >, < 温度測定値N >
 ※測定値は、取得された順番に出力されます。

参照：「測定値のフォーマット」(p.32)

- 説明
- :SYST:COMM:FORM FIX 設定時
 アイドル状態からトリガ待ち状態にし、測定終了後に測定値を読み出します。
 - :SYST:COMM:FORM FLOAT 設定時
 アイドル状態からトリガ待ち状態にし、測定終了後にサンプリング回数N回分、測定値を読み出します。

オートレンジの場合は、最適レンジに移動してから測定を行います。

参照：データ取得方法(p.71)、トリガ(p.40)

例

トリガソース	動作
IMMediate	トリガし、測定値を読み出します。
EXTErnal	TRIG 信号によってトリガし、続いて測定値を読み出します。

```

:TRIG:SOUR EXT
:SYST:COMM:FORM FIX;:SAMP:COUN 4
:READ?
*TRG
+094.31342E-03
:READ? TEMP
*TRG
+094.31343E-03,+23.87

:SYST:COMM:FORM FLOAT
:READ?
*TRG
+9.46651535E-02,+9.44042868E-02,+9.42089137E-02,+9.42364743E-02
:READ? TEMP
*TRG
+9.46651534E-02,+23.84,+9.44042867E-02,+23.85,+9.42089136E-02,+23.86,+9.42364742E-02,+23.87

```

- 注記
- 本メッセージを受け付けると自動的に:INITiate:CONTinuous OFF になります。
 - 測定が終了するまで、次のコマンドは実行されません。ただし、*TRG と :ABORt は受け付けます。
 - *TRGコマンドでトリガを入れる場合、外部トリガ(トリガソース<EXTERNAL>)にしてコマンドを送信します。GP-IBの時はコマンド送信後サンプリング時間に相当する待ち時間をおいて、トリガ指定してください。
 - 外部トリガ(トリガソース<EXTERNAL>)の場合、データ出力機能がONになっていると測定値の応答が二重になってしまいます。データ出力機能はOFFにご使用ください。
 - レンジ切り替えやパネルロード後に測定値が安定するまでウェイトが必要です。ウェイト時間は測定対象によって異なります。

レンジを指定して電圧測定

構文 クエリ **:MEASure[:VOLTage]:DC? <レンジ/AUTO/MAX/MIN/DEFault>**
 応答 **<測定値>**

参照:「測定値のフォーマット」(p.32)

<レンジ> = 100mV/1V/10V/100V/1000V/測定したい電圧
 (MAX:1000V, MIN:100mV, DEFault:AUTO)
 ※Vは省略可能です。

説明 レンジを指定して測定を行い、最後の測定値を読み出します。

例 :SYST:COMM:FORM FIX
 :MEAS:DC?
 +0.00201E-03

 :SYST:COMM:FORM FLOAT
 :MEAS:DC?
 +2.01462719E-06

注記

- 本メッセージを受け付けると、自動的に測定関連の設定はリセットされ、
 :INITiate:CONTInuous OFF、温度表示がOFFになります。
- レンジは省略可能です。省略した場合は、DEFaultが設定されます。
- 100mVレンジに切り替える場合は内部キャリブレーションが働くため、応答に約700msかかります。

温度測定値の読み出し

構文 クエリ **:MEASure:TEMPerature?**
 応答 **<温度測定値>**

参照:「測定値のフォーマット」(p.32)

説明 電圧測定と温度測定を実行し、温度測定値を読み出します。

例 :MEAS:TEMP?
 +25.46

注記

- 本メッセージを受け付けると、自動的に測定関連の設定はリセットされ、
 :INITiate:CONTInuous OFF、温度表示がONになります。
- 温度は電圧値と合わせて更新されます。
- 他社製品との互換機能については、「他社製品との互換コマンド」(p.68)をご確認下さい。

最後の測定値の読み出し

構文 クエリ **:DATA:LAST? [TEMPerature]**
 応答

- TEMPerature 省略時
<電圧測定値>

- TEMPerature 指定時
<電圧測定値>,<温度測定値>

参照:「測定値のフォーマット」(p.32)

説明 最後(直近)の測定値を読み出します。トリガはしません。
 参照:データ取得方法(p.71)、トリガ(p.40)

例 :DATA:LAST?
 +5.48260994E-03

 :DATA:LAST? TEMP
 +5.48260994E-03,+23.87

注記 未測定の場合は測定異常の値を応答します。

測定の中断

構文	文	クエリ	:ABORt
	説明		:READ の中断(強制終了)をします。 なお、 ABORt により判定はクリアされます。
例			:READ?
			:ABOR
			中断します。

測定値の出力形式の設定と問い合わせ

構文	文	コマンド	:SYSTem:COMMunicate:FORMat < FIX/FLOAT >
		クエリ	:SYSTem:COMMunicate:FORMat?
		応答	< FIX/FLOAT >
説明			FETCH? , READ? , MEASure[:VOLTage]:DC? の結果が下記のように動作します。 FIX : 画面に表示される桁数とおなじ桁数で、最後の測定値のみを読み出します。 FLOAT : 小数点以下8桁の浮動小数値で、サンプリングデータ数分を読み出します。
			参照 : 「測定値のフォーマット」(p.32)
注記			ローカル状態からリモート状態に遷移した時の動作が、設定により異なります。
			FIX : 測定状態は、ローカル状態時を維持します。
			FLOAT : 測定状態はSTOP状態になります。

(3) セルフテスト

セルフテストの実行と結果の問い合わせ

構文 クエリ **:TEST:ALL?**
 応答 **< PASS/FAIL >**

例 **:TEST:ALL?**
PASS

注記 *TST?と同じ動作になります。

(4) 時計

日付の設定と問い合わせ

構文 コマンド **:SYSTem:DATE < 年 >,< 月 >,< 日 >**
 クエリ **:SYSTem:DATE?**
 応答 **< 年 >,< 月 >,< 日 >**
< 年 > = 00 ~ 99 [年]
< 月 > = 01 ~ 12 [月]
< 日 > = 01 ~ 31 [日]

説明 時計の日付設定を行います。

例 **:SYST:DATE 15,9,2**
 2015年9月2日に設定します。
:SYST:DATE?
15,9,2
 2015年9月2日です。

注記 範囲外の数値を設定すると実行エラーになります。存在しない日付(例えば 13,06,31)を設定すると実行エラーになります。

時計の設定と問い合わせ

構文 コマンド **:SYSTem:TIME < 時 >,< 分 >,< 秒 >**
 クエリ **:SYSTem:TIME?**
 応答 **< 時 >,< 分 >,< 秒 >**
< 時 > = 00 ~ 23 [時]
< 分 > = 00 ~ 59 [分]
< 秒 > = 00 ~ 59 [秒]

説明 時計の時刻設定を行います。

例 **:SYST:TIME 8,25,0**
 8時25分00秒に設定します。
:SYST:TIME?
23,9,53
 23時9分53秒です。

注記 範囲外の数値を設定すると実行エラーになります。存在しない時間(例えば 9,6,71)を設定すると実行エラーになります。

(5) 測定レンジ

測定レンジの設定と問い合わせ

構文	コマンド	[[:SENSe:]]VOLTage:DC:RANGe < 測定レンジ/MAX/MIN/DEFAult >
	クエリ	[[:SENSe:]]VOLTage:DC:RANGe?
	応答	< 測定レンジ >

コマンド

< 測定レンジ > = 100mV/1V/10V/100V/1000V/測定したい電圧
 (MAX: 1000V, MIN: 100mV, DEFAult: 1000V)
 ※Vは省略可能です。

クエリ

**< 測定レンジ > = +1.00000000E-01/ +1.00000000E+00/ +1.00000000E+01/
 +1.00000000E+02/ +1.00000000E+03**

例 VOLT:DC:RANG 100mV
 VOLT:DC:RANG?
 +1.00000000E-01

注記 • コマンドにより測定レンジを設定する際、測定したい電圧を引数に用いると、その電圧を計測するために最適なレンジが設定されます。

例)

VOLT:DC:RANG 6V
 VOLT:DC:RANG?
 +1.00000000E+01

• Autoレンジが有効な状態でのクエリの結果は、その時点で測定系が選択しているレンジとなります。

測定オートレンジの設定と問い合わせ

構文	コマンド	[[:SENSe:]]VOLTage:DC:RANGe:AUTO < 1/0/ON/OFF >
	クエリ	[[:SENSe:]]VOLTage:DC:RANGe:AUTO?
	応答	< 1/0 >

例 VOLT:DC:RANG:AUTO OFF
 VOLT:DC:RANG:AUTO?
 0

(6) 測定ファンクション

測定ファンクションの設定と問い合わせ

構文	コマンド	[[:SENSe:]]FUNCtion[:ON] < TEMPerature/VOLTage[:DC] >
	クエリ	[[:SENSe:]]FUNCtion[:ON]?
	応答	"VOLT:DC"

TEMPerature = 電圧測定値と温度測定値が表示されます。

VOLTage[:DC] = 電圧測定値のみが表示されます。

例 FUNC TEMP
 FUNC?
 "VOLT:DC"

注記 クエリの結果は、"VOLT:DC" (固定)となります。

(7) 測定スピード

積分時間(PLC)の設定と問い合わせ

構文 コマンド **[[:SENSe:]VOLTage[:DC]:NPLCycles** < 積分時間 (PLC) >
 /MAX/MIN/DEfault/SLOW/MEDium/ FAST >
 クエリ **[[:SENSe:]VOLTage[:DC]:NPLCycles?**
 応答 < 積分時間(PLC) >
 < 積分時間(PLC) > = 0.02 ~ 100 [PLC]
 (MAX: 100, MIN: 0.02, DEfault: 10)

各スピードにおける積分時間は下記になります。

スピード	積分時間[PLC]
SLOW	100
MEDium	10
FAST	1

例 :VOLT:NPLC 0.02
 :VOLT:NPLC?
 +2.00000000E-02

 :VOLT:NPLC FAST
 :VOLT:NPLC?
 +1.00000000E+00

積分時間(sec)の有効設定と問い合わせ

構文 コマンド **[[:SENSe:]VOLTage[:DC]:APERture:ENABled** < 1/0/ON/OFF >
 クエリ **[[:SENSe:]VOLTage[:DC]:APERture:ENABled?**
 応答 < 1/0 >

説明 コマンド
 積分時間(sec)を指定した測定を行う機能の設定を行います。
 クエリ
 積分時間(sec)を指定した測定を行う機能の設定を問い合わせます。

例 :VOLT:APER:ENAB ON
 :VOLT:APER:ENAB?
 1

積分時間(sec)の設定と問い合わせ

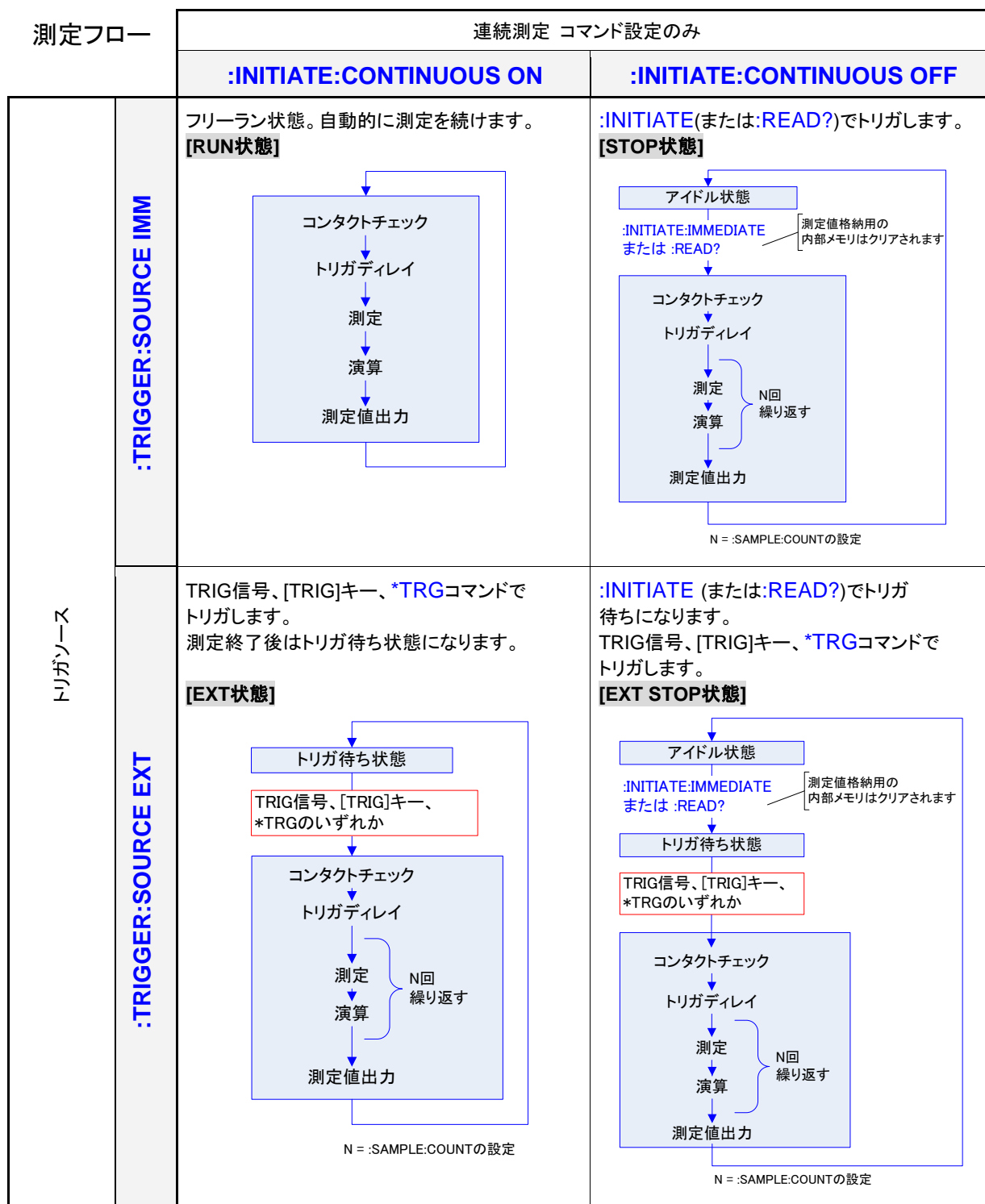
構文 コマンド **[[:SENSe:]VOLTage[:DC]:APERture** < 積分時間 (sec) >
 /MAX/MIN/DEfault >
 クエリ **[[:SENSe:]VOLTage[:DC]:APERture?**
 応答 < 積分時間(sec) >
 < 積分時間(sec) > = 0.001 ~ 9.999[sec]
 (MAX: 9.999, MIN: 0.001, DEfault: 1)

例 :VOLT:APER 2
 :VOLT:APER?
 +2.00000000E+00

(8) トリガ

トリガソースと連続測定の関係について

連続測定の設定 (:INITIATE:CONTINUOUS) (参照:p.41) とトリガソースの設定 (:TRIGGER:SOURCE) (参照:p.41) により、次のように動作します。 参照:「4 データ取得方法」(p.71)



:INITIATE:CONTINUOUS OFFに設定されている場合、ローカル状態に戻ったとき、または電源を入れ直したときは、次の電源投入時は:INITIATE:CONTINUOUS ONの状態に設定されます。

参照:「リモート状態を解除する (ローカル状態へ)」(p.58)
測定値の取得方法について:「データ取得方法」(p.71)

連続測定の設定と問い合わせ

構文	コマンド	:INITiate:CONTinuous < 1/0/ON/OFF >
	クエリ	:INITiate:CONTinuous?
説明	応答	< 1/0 >
		<ON> = 連続測定有効 <OFF> = 連続測定無効
説明		<ul style="list-style-type: none"> 連続測定有効 ([RUN状態]、[EXT状態]) : 測定終了後、トリガ待ち状態になります。内部トリガ(トリガソース<IMMEDIATE>)のときは、すぐに 次のトリガが発生するのでフリーラン状態になります。 連続測定無効 ([STOP状態]、[EXT STOP状態]) : 測定終了後、トリガ待ち状態ではなくアイドル状態になります。 アイドル状態とは、トリガを受け付けけない状態です。:INITiate[:IMMediate] によりトリガ待ち状態になります。 リモート状態が解除されると、連続測定有効になります。
例		:INIT:CONT OFF
		:INIT:CONT? 0

トリガ待ちの設定

構文	コマンド	:INITiate[:IMMediate]
	説明	トリガシステムをアイドル状態からトリガ待ち状態にします。 測定値格納用メモリはクリアされます。
例		トリガシステムを連続測定無効にし、1 回トリガして値を読み取る場合 送信 :TRIG:SOUR IMM..... トリガ待ちになったら即座にトリガ :INIT:CONT OFF..... 連続測定を無効にする :INIT..... トリガ待ちに設定 :TRIG:SOUR IMM なので即座にトリガ
注記		<ul style="list-style-type: none"> 本メッセージを受け付けると自動的に:INITIATE:CONTINUOUS OFF になります。 内部トリガ(トリガソース<IMMEDIATE>)のときは、すぐにトリガし、アイドル状態になります。 外部トリガ(トリガソース<EXTERNAL>)のときは、外部からのトリガ待ち状態になり、トリガを受け付けると1回測定してアイドル状態になります。

トリガソースの設定と問い合わせ

構文	コマンド	:TRIGger:SOURce < IMMEDIATE/ EXTeRnal/BUS >
	クエリ	:TRIGger:SOURce?
説明	応答	< IMM/ EXT >
		<IMMediate> = 内部トリガ <EXTeRnal> = 外部トリガ <BUS> = 外部トリガ
例		:TRIG:SOUR IMM :TRIG:SOUR? IMM
注記		コマンドによりBUSを設定した場合は、EXTと設定します

測定回数の設定と問い合わせ

構文 コマンド **:SAMPle:COUNT** < 測定回数/MAX/MIN/DEFault >
 クエリ **:SAMPle:COUNT?**
 応答 < 測定回数 >
 < 測定回数 > = 1 ~ 5000
 (MAX: 5000, MIN: 1, DEFault: 1)

例 :SAMP:COUN 500
 :SAMP:COUN?
 500

トリガディレイの設定と問い合わせ

構文 コマンド **:TRIGger:DELaY** < デレイ時間/MAX/MIN/DEFault >
 クエリ **:TRIGger:DELaY?**
 応答 < デレイ時間 >
 < デレイ時間 > = 0 ~ 9.999 [sec]
 (MAX: 9.999, MIN: 0, DEFault: 0)

例 :TRIG:DEL 1
 :TRIG:DEL?
 +1.00000000E+00

トリガディレイのプリセット設定と問い合わせ

構文 コマンド **:TRIGger:DELaY:AUTO** <1/0/ON/OFF >
 クエリ **:TRIGger:DELaY:AUTO?**
 応答 < 1/0 >

例 :TRIG:DEL:AUTO ON
 :TRIG:DEL:AUTO?
 1

(9) 桁数設定

電圧測定値桁数の設定と問い合わせ

構文 コマンド **[[:SENSe:]VOLTage:DIGits < 桁数/MAX/MIN/DEFAult >**
 クエリ **[[:SENSe:]VOLTage:DIGits?**
 応答 **< 桁数 >**
< 桁数 > = 4 ~ 8
 (MAX : 8, MIN : 4, DEFAult : 8)

桁数設定	画面設定
8	7.5
7	6.5
6	5.5
5	4.5
4	3.5

例 **:VOLT:DIG 8**
:VOLT:DIG?
8

(10) ラベル表示

ラベル表示機能の設定と問い合わせ

構文 コマンド **:SYSTem:LABel:STATe < 1/0/ON/OFF >**
 クエリ **:SYSTem:LABel:STATe?**
 応答 **< 1/0 >**
 例 **SYST:LAB:STAT ON**
SYST:LAB:STAT?
1

ラベル名の設定と問い合わせ

構文 コマンド **:SYSTem:LABel < ラベル名 >**
 クエリ **:SYSTem:LABel?**
 応答 **< ラベル名 >**
< ラベル名 > = “最大8文字の文字列”
 例 **SYST:LAB “LAVEL_00”**
SYST:LAB?
“LAVEL_00”

注記

- ・ラベル文字には、本体で設定可能な文字のみ指定して下さい。
- ・**:SYSTem:LABel**コマンドによりラベル名を指定すると、ラベル表示機能は強制的にONになります。
- ・ラベル表示機能がOFFの状態でクエリを実行するとOFFが返ります。

(11) コンパレータ

コマンドでコンパレータの設定をした場合、測定レンジは自動的に選択されません。

コンパレータの実行と問い合わせ

構文	コマンド	:CALCulate:LIMit[:STATe] < 1/0/ON/OFF >
	クエリ	:CALCulate:LIMit[:STATe]?
	応答	< 1/0 >
例		:CALC:LIM:STAT ON :CALC:LIM:STAT? 1

注記 コンパレータを実行すると、BIN機能は OFF になります。

ブザーの設定と問い合わせ

構文	コマンド	:CALCulate:LIMit:BEEPer < 条件 >,< タイプ >,< 回数 >
	クエリ	:CALCulate:LIMit:BEEPer? < 条件 >
	応答	< 条件 >,< タイプ >,< 回数 >
		< 条件 > = HI/ IN /LO
		< タイプ > = 0: ブザーOFF、1~3: タイプ1~タイプ3
		< 回数 > = 0: 連続、1~5: 回数[回]
例		:CALC:LIM:BEEP IN,1,0 :CALC:LIM:BEEP? IN IN,1,0

注記 引数の回数 は省略可能です。省略した場合は、回数の設定は行われません。

絶対値判定機能の設定と問い合わせ

構文	コマンド	:CALCulate:LIMit:ABSolute <1/0/ON/OFF >
	クエリ	:CALCulate:LIMit:ABSolute?
	応答	< 1/0 >
例		:CALC:LIM:ABS ON :CALC:LIM:ABS? 1

上限値有効の設定と問い合わせ

構文	コマンド	:CALCulate:LIMit:UPPer:ENABle <1/0/ON/OFF >
	クエリ	:CALCulate:LIMit:UPPer:ENABle?
	応答	< 1/0 >
例		:CALC:LIM:UPP:ENAB ON :CALC:LIM:UPP:ENAB? 1

上限値の設定と問い合わせ

構文	コマンド	:CALCulate:LIMit:UPPer[:DATA] < 上限値 >
	クエリ	:CALCulate:LIMit:UPPer[:DATA]?
	応答	< 上限値 >
		< 上限値 > = -1.0E+03 ~ +1.0E+03 (NRf) [V]
例		:CALC:LIM:UPP 1.0 :CALC:LIM:UPP? +1.00000000E+00

上限値が1.0Vになります。

下限値有効の設定と問い合わせ

構文 コマンド **:CALCulate:LIMit:LOWer:ENABLE** <1/0/ON/OFF >
クエリ **:CALCulate:LIMit:LOWer:ENABLE?**
応答 < 1/0 >
例 **:CALC:LIM:LOW:ENAB ON**
:CALC:LIM:LOW:ENAB?
1

下限値の設定と問い合わせ

構文 コマンド **:CALCulate:LIMit:LOWer[:DATA]** < 下限値 >
クエリ **:CALCulate:LIMit:LOWer[:DATA]?**
応答 < 下限値 >
< 下限値 > = -1.0E+03 ~ +1.0E+03 (NRf) [V]
例 **:CALC:LIM:LOW 0.9**
:CALC:LIM:LOW?
+9.00000000E-01

下限値が0.9Vになります。

コンパレータの判定継続機能の設定と問い合わせ

構文 コマンド **:CALCulate:LIMit:DElay** <1/0/ON/OFF >
クエリ **:CALCulate:LIMit:DElay?**
応答 < 1/0 >
例 **:CALC:LIM:DEL ON**
コンパレータの判定継続機能をONIにします。

コンパレータの判定継続回数の設定と問い合わせ

構文 コマンド **:CALCulate:LIMit:DElay:COUNt** < 回数 >
クエリ **:CALCulate:LIMit:DElay:COUNt?**
応答 < 回数 >
< 回数 > = 2 ~ 10 (NR1)
例 **:CALC:LIM:DEL:COUN 10**
コンパレータの判定継続回数を10回にします。

コンパレータイベントステータスレジスタクリア

構文 コマンド **:CALCulate:LIMit:CLEar[:IMMediate]**
説明 コンパレータ、BIN機能に関する状態問い合わせレジスタ・グループの条件レジスタのビット (bit11, bit12, bit13) をクリアします。

コンパレータ判定結果の問い合わせ

構文 クエリ **:CALCulate:LIMit:RESult?**
応答 <HI/IN/LO/ERR/OFF>
例 **:CALC:LIM:RES?**
IN
コンパレータの判定結果が IN です。

注記 未測定の場合はERRを応答します。

(12) BIN

BIN測定の実行と問い合わせ

構文 コマンド **:CALCulate:BIN[:STATe] < 1/0/ON/OFF >**
クエリ **:CALCulate:BIN[:STATe]?**
応答 **< 1/0 >**

例 **:CALC:BIN:STAT ON**
:CALC:BIN:STAT?
1

注記 BIN機能を実行すると、コンパレータ機能は、OFF になります。

使用可能パターンの設定と問い合わせ

構文 コマンド **:CALCulate:BIN:ENABLE < 使用可能パターン >**
クエリ **:CALCulate:BIN:ENABLE?**
応答 **< 使用可能パターン > = 0 ~ 1023 (10 進数) (NR1)**
BIN 測定を実行する BIN 番号の bit を 1 にします。

512	256	128	64	32	16	8	4	2	1
bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
BIN9	BIN8	BIN7	BIN6	BIN5	BIN4	BIN3	BIN2	BIN1	BIN0

例 **:CALC:BIN:ENAB 15**
BIN0 ~ BIN3 が使用可能です。

上限値の設定と問い合わせ

構文 コマンド **:CALCulate:BIN:UPPer < BIN No. >, < 上限値 >**
クエリ **:CALCulate:BIN:UPPer? < BIN No. >**
応答 **< 上限値 >**
< BIN No.> = 0 ~ 9 (NR1)
< 上限値 > = -1.0E+03 ~ +1.0E+03 (NRf) [V]

例 **:CALC:BIN:UPPer 0,1.0**
BIN0の上限値が1.0Vになります。

下限値の設定と問い合わせ

構文 コマンド **:CALCulate:BIN:LOWer < BIN No. >, < 下限値 >**
クエリ **:CALCulate:BIN:LOWer? < BIN No. >**
応答 **< 下限値 >**
< BIN No.> = 0 ~ 9 (NR1)
< 下限値 > = -1.0E+03 ~ +1.0E+03 (NRf) [V]

例 **:CALC:BIN:LOW 0,0.9**
BIN0の下限値が0.9Vになります。

BIN判定結果の問い合わせ

構文 クエリ **:CALCulate:BIN:RESult?**
 応答 **<NR1>**
<NR1> = 0 ~ 1024
 BIN測定でPASSになっている BIN 番号の bit を 1 にします。

bit10	bit9	bit8	bit7	bit6	bit5
OB	BIN9	BIN8	BIN7	BIN6	BIN5
	bit4	bit3	bit2	bit1	bit0
	BIN4	BIN3	BIN2	BIN1	BIN0

例 **:CALC:BIN:RES?**
128
 BIN7 が PASS です。

(13) 測定条件の保存と読み込み

測定条件の保存・読み込み

構文 コマンド ***SAV <パネルNo.>**
***RCL <パネル No.>**
 注記 詳細は共通コマンドの***SAV**と***RCL**を確認して下さい。

パネルの削除

構文 コマンド **:SYSTem:PANel:CLEar <パネル番号>**
< パネル番号 > = 1 ~ 30(NR1)

パネル名の設定と問い合わせ

構文 コマンド **:SYSTem:PANel:NAME <パネル番号>,<パネル名(文字列データ: 最大10文字)>**
 クエリ **:SYSTem:PANel:NAME? <パネル番号>**
 応答 **<パネル番号>,<パネル名(文字列データ: 最大10文字)>**
< パネル番号 > = 1 ~ 30(NR1)
 文字列データについてはデータ部を参照してください。(p.4)

例 **:SYST:PAN:NAME 1, "PANEL_1"**
:SYST:PAN:NAME? 1
1,"PANEL_1"

注記 パネル保存されていないパネルを指定した場合は、実行エラーになります。

パネルの保存日時の問い合わせ

構文 クエリ **:SYSTem:PANel:DATE? <パネル番号>**
 応答 **< 西暦 月 日 時 分 秒 >**
< パネル番号 > = 1 ~ 30(NR1)

例 **:SYST:PAN:DATE? 1**
15,10,31,23,1

注記 パネル保存されていないパネルを指定した場合は、実行エラーになります。

設定ファイルのUSBメモリへの保存

- 構文** コマンド **:MMEMory:STORe:STATe <ファイル名> [,ALL]**
- 例** :MMEM:STOR:STAT "SET_000.SET"
:MMEM:STOR:STAT "SET_000"
:MMEM:STOR:STAT "ALL_000.ALL", ALL
:MMEM:STOR:STAT "ALL_000", ALL
- 注記**
- :MMEM:STOR:STAT <ファイル名>, ALL と最後にALLを付けるとパネルデータを含めた全設定データがファイルに保存されます。
 - ファイル名は、拡張子を含めて指定しても、含めないで指定しても、コマンドを受け付けます。
 - 拡張子を省略した場合、下記のように動作します。
:MMEM:STOR:STAT "SET_000" … ファイル名"SET_000.SET"で保存されます。
:MMEM:STOR:STAT "ALL_000", ALL … ファイル名"ALL_000.ALL"で保存されます。
 - ファイル名の文字列長は、拡張子(.xxx)を含めて12文字以内で指定可能です。(拡張子を含めない場合は8文字以内)それ以外は実行エラーになります。
 - ファイル名に本体で設定できない文字が含まれる場合は実行エラーになります。
 - USBメモリが装着されていない場合は、実行エラーになります。
 - 通信インタフェースがUSBに設定されている場合は、実行エラーになります。

設定ファイルのUSBメモリからの読み込み

- 構文** コマンド **:MMEMory:LOAD:STATe <ファイル名>**
- 例** :MMEM:LOAD:STAT "SET_000" ※SET_000.SETをロードします
:MMEM:LOAD:STAT "ALL_000.ALL" ※ALL_000.ALLをロードします
- 注記**
- ファイル名は、拡張子を含めて指定しても、含めないで指定しても、コマンドを受け付けます。
 - 拡張子を省略した場合、拡張子は.SETのファイルを読み込みます。拡張子.ALLのファイルを読み込む場合は、必ず拡張子を指定して下さい。
 - ファイル名の文字列長は、拡張子(.xxx)を含めて12文字以内で指定可能です。(拡張子を含めない場合は8文字以内)それ以外は実行エラーになります。
 - ファイル名に本体で設定できない文字が含まれる場合は実行エラーになります。
 - USBメモリが装着されていない場合は、実行エラーになります。
 - 指定されたファイルが存在しない場合は、実行エラーになります。
 - 通信インタフェースがUSBに設定されている場合は、実行エラーになります。

起動時にパネルを読み込む機能の設定と問い合わせ

- 構文** コマンド **:MMEMory:STATe:RECall:AUTO <1/0/ON/OFF>**
クエリ **:MMEMory:STATe:RECall:AUTO?**
応答 **<1/0>**
- 例** :MMEM:STAT:REC:AUTO ON
:MMEM:STAT:REC:AUTO?
1

起動時に読み込むパネル番号の設定と問い合わせ

- 構文** コマンド **:MMEMory:STATe:RECall:SElect <パネル番号>**
クエリ **:MMEMory:STATe:RECall:SElect?**
応答 **<パネル番号>**
<パネル番号> = 0 ~ 30 (NR1)
0 : FACTORY設定をロードします。
1 ~ 30 : パネル設定をロードします。
- 例** :MMEM:STAT:REC:SEL 5
:MMEM:STAT:REC:SEL?
5

(14) スムージング

スムージング機能の実行と問い合わせ

構文	コマンド	:CALCulate:SMOothing[:STATe] <1/0/ON/OFF >
	クエリ	:CALCulate:SMOothing[:STATe]?
	応答	<1/0 >

例 CALC:SMO ON
 CALC:SMO?
 1

スムージング回数の設定と問い合わせ

構文	コマンド	:CALCulate:SMOothing:RESPonse <回数/SLOW/MEDium/FAST>
	クエリ	:CALCulate:SMOothing:RESPonse?
	応答	< 回数 >

< 回数 > = 2 ~ 100 (NR1)

SLOW、MEDium、FASTを指定した場合は下記の値が設定されます。

SLOW : 100

MEDium : 50

FAST : 10

例 CALC:SMO:RESP 50
 CALC:SMO:RESP?
 50

(15) ホールド

オートホールドの実行と問い合わせ

構文	コマンド	[:SENSe:]HOLD:AUTO < 1/0/ON/OFF >
	クエリ	[:SENSe:]HOLD:AUTO?
	応答	< 1/0 >

例 HOLD:AUTO ON
 HOLD:AUTO?
 1

注 記 オートホールドを実行すると、下記の設定になります。

- 積分時間: MED
- 入力抵抗: 10M Ω
- :INITIATE:CONTINUOUS ON
- 内部トリガ(トリガソース<IMMEDIATE>)
- コンタクトチェック: ON

オートホールド範囲の設定と問い合わせ

構文	コマンド	[:SENSe:]HOLD:BOUND < ホールド範囲/MAX/MIN/DEFAult >
	クエリ	[:SENSe:]HOLD:BOUND?
	応答	< ホールド範囲 >

< ホールド範囲 > = 0.001 ~ 1.000 (NRf) [%]

(MAX : 1.000, MIN : 0.001, DEFAult : 0.1)

例 HOLD:BOUN 0.01
 HOLD:BOUN?
 +1.00000000E-02

(16) コンタクトチェック

コンタクトチェックの測定値の問い合わせ

構文 クエリ **[[:SENSe:]VOLTage:DC:CONtact:CAPacitance?**
 応答 **< コンタクトチェックの測定値 >**

説明 最後 (直近) のコンタクトチェックの測定値 (単位 F) を読み出します。
 応答のフォーマットは下記になります。

測定値	±OvrRng	表示時	測定異常時
± □.□□□□□□□□E±0□	±9.90000000E+28		+9.91000000E+28

注記: 測定値は小数点以下8桁の浮動小数値で表現されます。

例 VOLT:DC:CONT:CAP?
 +1.00000000E-02

コンタクトチェックの実行と問い合わせ

構文 コマンド **[[:SENSe:]VOLTage:DC:CONtact:CAPacitance:STATe**
< 1/0/ON/OFF >

クエリ **[[:SENSe:] VOLTage:DC:CONtact:CAPacitance:STATe?**
 応答 **< 1/0 >**

例 VOLT:DC:CONT:CAP:STAT ON
 VOLT:DC:CONT:CAP:STAT?
 1

コンタクトチェックの閾値の設定と問い合わせ

構文 コマンド **[[:SENSe:]VOLTage:DC:CONtact:CAPacitance:THReshold**
< 閾値/MAX/MIN/DEFault >

クエリ **[[:SENSe:]VOLTage:DC:CONtact:CAPacitance:THReshold**
?

応答 **< 閾値 >**
< 閾値 > = 0.5 ~ 50.0 (NRf) [nF]
 (MAX: 50.0, MIN: 0.5, DEFault: 1.0)

例 VOLT:DC:CONT:CAP:THR 40.0
 VOLT:DC:CONT:CAP:THR?
 +4.00000000E+01

コンタクトチェックの積分時間の設定と問い合わせ

構文 コマンド **[[:SENSe:]VOLTage:DC:CONtact:CAPacitance:TIME < 積分**
時間/MAX/MIN/DEFault >

クエリ **[[:SENSe:]VOLTage:DC:CONtact:CAPacitance:TIME?**
 応答 **< 積分時間 >**
< 積分時間 > = 0.001 ~ 0.1 (NRf) [sec]
 (MAX:0.1, MIN: 0.001, DEFault: 0.01)

例 VOLT:DC:CONT:CAP:TIME 0.001
 VOLT:DC:CONT:CAP:TIME?
 +1.00000000E-03

(17) 入力抵抗の切替

入力抵抗の設定と問い合わせ

構文	コマンド	<code>[[:SENSe:]VOLTage[:DC]:IMPedance:AUTO <1/0/ON/OFF ></code>
	クエリ	<code>[[:SENSe:]VOLTage[:DC]:IMPedance:AUTO?</code>
	応答	<code>< 1/0 ></code>
例		<code>VOLT:IMP:AUTO ON</code> <code>VOLT:IMP:AUTO?</code> <code>1</code>

注記 OFF設定時は、入力抵抗は10MΩ固定になります。

(18) ゼロ点調節 (NULL 機能)

ゼロ点調節の実行と問い合わせ

構文	コマンド	<code>[[:SENSe:]VOLTage:DC:NULL[:STATe] <1/0/ON/OFF ></code>
	クエリ	<code>[[:SENSe:]VOLTage:DC:NULL[:STATe]?</code>
	応答	<code><1/0></code>
例		<code>VOLT:DC:NULL ON</code> <code>VOLT:DC:NULL?</code> <code>1</code>

ゼロ点調節値の設定と問い合わせ

構文	コマンド	<code>[[:SENSe:]VOLTage:DC:NULL:VALue < 調節値/MAX/MIN/DEFault ></code>
	クエリ	<code>[[:SENSe:]VOLTage:DC:NULL:VALue?</code>
	応答	<code>< 調節値 ></code> <code>< 調節値 > = -1.0E+3 ~ +1.0E+3 (NRf) [V]</code> <code>(MAX : 1.0E+3, MIN : -1.0E+3, DEFault : 0.0)</code>
例		<code>VOLT:DC:NULL:VAL 0.1</code> <code>VOLT:DC:NULL:VAL?</code> <code>+1.00000000E-01</code>

(19) 温度補正(TC)

温度補正(TC)の実行と問い合わせ

構文	コマンド	:CALCulate:TCORrect:STATe < 1/0/ON/OFF >
	クエリ	:CALCulate:TCORrect:STATe?
	応答	< 1/0 >
例		:CALC:TCOR:STAT ON
		:CALC:TCOR:STAT?
		0

温度補正(TC)の設定と問い合わせ

構文	コマンド	:CALCulate:TCORrect:PARAmeter < 基準温度/MAX/MIN/DEFAult >, < 温度係数/MAX/MIN/DEFAult >
	クエリ	:CALCulate:TCORrect:PARAmeter?
	応答	< 基準温度 >, < 温度係数 >
		< 基準温度 > = -10.0 ~ 60.0 (NR2) [°C] (MAX:60.0, MIN:-10.0, DEFAult:20.0)
		< 温度係数 > = -1000 ~ 1000 (NR1) [ppm/°C] (MAX:1000, MIN:-1000, DEFAult:0)
例		:CALC:TCOR:PAR 20,0
		:CALC:TCOR:PAR?
		20.0,0

注記 基準温度の単位は [°C]、温度係数の単位 [ppm/°C]です。

(20) スケーリング

スケーリング機能の実行と問い合わせ

構文	コマンド	:CALCulate:SCALE[:STATe] <1/0/ON/OFF >
	クエリ	:CALCulate:SCALE[:STATe]?
	応答	< 1/0 >
例		:CALC:SCAL:STAT ON
		:CALC:SCAL:STAT?
		1

スケーリング補正係数の設定と問い合わせ

構文	コマンド	:CALCulate:SCALE:PARAmeterA < 補正係数/MAX/MIN/DEFAult >
	クエリ	:CALCulate:SCALE:PARAmeterA?
	応答	< 補正係数 >
		< 補正係数 > = -1.0000000E+09 ~ 1.0000000E+09 (NRf) (MAX: 1.0E+09, MIN: -1.0E+09, DEF: 1.0)
例		:CALC:SCAL:PARA 2E+00
		:CALC:SCAL:PARA?
		2.0000000E+00

スケーリングオフセットの設定と問い合わせ

構文	コマンド	:CALCulate:SCALE:PARAmeterB < オフセット/MAX/MIN/DEFAult >
	クエリ	:CALCulate:SCALE:PARAmeterB?
	応答	< オフセット >
		< オフセット > = -1.0000000E+09 ~ 1.0000000E+09 (NRf) (MAX: 1.0E+09, MIN: -1.0E+09, DEF: 0.0)
例		:CALC:SCAL:PARB 1E+03
		:CALC:SCAL:PARB?
		1.0000000E+03

スケーリング単位機能の設定と問い合わせ

構文 コマンド **:CALCulate:SCALE:UNIT:STATe < 0 ~ 2 >**
クエリ **:CALCulate:SCALE:UNIT:STATe?**
応答 **< 0 ~ 2 >**
0: NONE
1: USER
2: V

例 **:CALC:SCAL:UNIT:STAT 2**
:CALC:SCAL:UNIT:STAT?
2

スケーリング単位の設定と問い合わせ

構文 コマンド **:CALCulate:SCALE:UNIT < 単位 >**
クエリ **:CALCulate:SCALE:UNIT?**
応答 **< 単位 >**
< 単位 > = 任意単位(任意単位は文字列データ: 最大3文字)
文字列データについてはデータ部を参照してください。(p.4)

例 **:CALC:SCAL:UNIT "ohm"**
:CALC:SCAL:UNIT?
"ohm"

(21) 統計

統計演算機能の実行と問い合わせ

構文 コマンド **:CALCulate:AVERage[:STATe] < 1/0/ON/OFF >**
クエリ **:CALCulate:AVERage[:STATe]?**
応答 **< 1 >**

例 **:CALC:AVER:STAT ON**
:CALC:AVER:STAT?
1

注記 統計演算機能はOFFにはできません。

統計演算結果のクリア

構文 コマンド **:CALCulate:AVERage:CLEar[:IMMediate]**

プリントアウト時に統計演算結果をクリアする機能の設定と問い合わせ

構文 コマンド **:CALCulate:AVERage:CLEar:PRINt < 1/0/ON/OFF >**
クエリ **:CALCulate:AVERage:CLEar:PRINt?**
応答 **< 1/0 >**

例 **:CALC:AVER:CLE:PRIN ON**
:CALC:AVER:CLE:PRIN?
1

統計演算データ数の問い合わせ

構文 クエリ **:CALCulate:AVERage:COUNT?**
応答 **< データ数 (NR1) >**
データ数 = 0 ~ 1000000

例 **:CALC:AVER:COUN?**
23456

平均値、標準偏差、最小値、最大値の問い合わせ

構文 クエリ **:CALCulate:AVERage:ALL?**
 応答 < 平均値 (NR3)>, < 標準偏差 (NR3)>, < 最小値 (NR3)>, < 最大値 (NR3)>
例 :CALC:AVER:ALL?
 +1.00520000E+03,+1.00512000E+03,+1.00527000E+03,+4.13500000E-01
注記 有効な統計データが無い場合は実行エラーになります。

標準偏差の問い合わせ

構文 クエリ **:CALCulate:AVERage:SDEviation?**
 応答 < 標準偏差(NR3)>
例 :CALC:AVER:SDEV?
 +1.00512000E+03
注記 有効な統計データが無い場合は実行エラーになります。

平均値の問い合わせ

構文 クエリ **:CALCulate:AVERage:AVERage?**
 応答 < 平均値 (NR3)>
例 :CALC:AVER:AVER?
 +1.00520000E+03
注記 有効な統計データが無い場合は実行エラーになります。

最小値の問い合わせ

構文 クエリ **:CALCulate:AVERage:MINimum?**
 応答 < 最小値 (NR3)>
例 :CALC:AVER:MIN?
 +1.00527000E+03
注記 有効な統計データが無い場合は実行エラーになります。

最大値の問い合わせ

構文 クエリ **:CALCulate:AVERage:MAXimum?**
 応答 < 最大値 (NR3)>
例 :CALC:AVER:MAX?
 +4.13500000E-01
注記 有効な統計データが無い場合は実行エラーになります。

Peak to Peak値の問い合わせ

構文 クエリ **:CALCulate:AVERage:PTPeak?**
 応答 < Peak to Peak値 (NR3)>
例 :CALC:AVER:PTP?
 +3.12973000E-01
注記 有効な統計データが無い場合は実行エラーになります。

コンパレータ結果の問い合わせ

構文 クエリ **:CALCulate:AVERage:LIMit?**
 応答 < Hi の数 (NR1) >,< IN の数 (NR1) >,< Lo の数 (NR1) >,
 < 測定異常の数 (NR1) >,< レンジオーバーの数 (NR1) >
例 :CALC:AVER:LIM?
 1516,9310,737,16,5

BIN結果の問い合わせ

構文 クエリ **:CALCulate:AVERage:BIN?**
 応答 < BIN0 の数 (NR1) >,...,< BIN9の数 (NR1) >,< OUTの数 (NR1) >,
 < 測定異常の数 (NR1) >
例 :CALC:AVER:BIN?
 1516,9310,10,10,10,10,10,10,10,10,100,737,16

工程能力指数の問い合わせ

構文 クエリ **:CALCulate:AVERage:CP?**
 応答 < Cp (NR2) >,< CpK (NR2) >
例 CALC:AVER:CP?
 0.86,0.14

(22) キーロック
キーロック状態の設定と問い合わせ

構文 コマンド **:SYSTem:KLOCK <1 /0/ON/OFF >**
 クエリ **:SYSTem:KLOCK?**
 応答 < 1/0 >
例 :SYST:KLOCK ON
 :SYST:KLOCK?
 1

(23) 音設定
キークリック音の設定と問い合わせ

構文 コマンド **:SYSTem:CLICk:STATe < 1/0/ON/OFF >**
 クエリ **:SYSTem:CLICk:STATe?**
 応答 < 1/0 >
例 :SYST:CLIC:STAT ON
 :SYST:CLIC:STAT?
 1

ホールド音の設定と問い合わせ

構文 コマンド **:SYSTem:HOLD:BEEPPer:STATe < 1/0/ON/OFF >**
 クエリ **:SYSTem:HOLD:BEEPPer:STATe?**
 応答 < 1/0 >
例 :SYST:HOLD:BEEP:STAT ON
 :SYST:HOLD:BEEP:STAT?
 1

エラービープ音の設定と問い合わせ

構文 コマンド **:SYSTem:BEEPer:STATe** <1/0/ON/OFF >
クエリ **:SYSTem:BEEPer:STATe?**
応答 < 1/0 >
例 :SYST:BEEP:STAT ON
:SYST:BEEP:STAT?
1

音量の設定と問い合わせ

構文 コマンド **:SYSTem:BEEPer:VOLume** < 音量/MAX/MIN/DEFAult >
クエリ **:SYSTem:BEEPer:VOLume?**
応答 < 音量 >
< 音量 > = 0/1/2/3
(MAX : 3, MIN : 0, DEFAult : 2)

音量設定	画面設定
0	OFF
1	SMALL
2	MED
3	LARGE

例 :SYST:BEEP:VOL 3
:SYST:BEEP:VOL?
3

ビープ音を一回鳴らす

構文 コマンド **:SYSTem:BEEPer[:IMMediate]**

(24) 表示設定

LCD表示機能の設定と問い合わせ

構文 コマンド **:DISPlay[:STATe]** < 1/0/ON/OFF >
クエリ **:DISPlay[:STATe]?**
応答 < 1 >
例 :DISP OFF
:DISP?
1

注記

- ・ コマンドによりOFFを設定してもLCD表示はONのままとなります。
- ・ クエリの応答は常に1となります。

画面の輝度設定と問い合わせ

構文 コマンド **:DISPlay:BACKlight** < 輝度/MAX/MIN/DEFAult >
クエリ **:DISPlay:BACKlight?**
応答 < 輝度 >
<輝度> = 0~100 [%]
(MAX: 100, MIN: 0, DEF: 80)
例 :DISP:BACK 50
:DISP:BACK?
50

注記 輝度の分解能は10%です。

画面色の設定と問い合わせ

構文 コマンド **:DISPlay:TYPe** < 画面色タイプ/MAX/MIN/DEFault >
 クエリ **:DISPlay:TYPe?**
 応答 < 画面色タイプ >
 <画面色タイプ> = 0 ~ 1
 (MAX: 1, MIN: 0, DEF: 0)
 画面色タイプは、0がBLUEタイプ、1がGRAYタイプになります。

例 :DISP:TYP 0
 :DISP:TYP?
 0

表示タイプの設定と問い合わせ

構文 コマンド **:DISPlay:VIEW** < NUMeric /TCHart/METer/STATistics/HISTogram >
 クエリ **:DISPlay:VIEW?**
 応答 < NUM/TCH/MET/STAT/COMP >
 NUMeric: 測定値のみ表示
 TCHart: トレンドチャート表示
 METer: バーグラフ表示
 STATistics: 統計表示
 HISTogram: 統計表示

例 :DISP:VIEW NUM
 :DISP:VIEW?
 NUM

注記

- HISTogramを設定した場合には、STATisticsに設定されます。
- コンパレータ測定時、またはBIN測定時にNUMericが選択されている場合、:DISPlay:VIEW?の結果はCOMPと返ります。

(25) 電源周波数

電源周波数の設定と問い合わせ

構文 コマンド **:SYSTem:LFRequency** < AUTO/50/60 >
 クエリ **:SYSTem:LFRequency?**
 応答 < AUTO/50/60 >
 例 :SYST:LFR 50
 :SYST:LFR?
 50

(26) システムリセット

システムリセットの実行

構文 コマンド **:SYSTem:PRESet**
:STATus:PRESet
 説明 *RSTの同じ動作をします。

(27) 通信設定

ローカル状態へ戻る

構文	コマンド	:SYSTem:LOCal
説明	通信によるリモート状態を解除しローカル状態に戻します。キー操作が可能となります。	
例	:SYST:LOC	

通信インタフェースの設定と問い合わせ

構文	コマンド	:SYSTem:COMMunicate < USB/LAN/RS232c/GPIB/KEYBD/PRINter/MMEMory >
クエリ	:SYSTem:COMMunicate?	
応答	< USB/LAN/RS232C/GPIB >	
注記	コマンドを受け付けた時点でインタフェースを変更します。	

コマンドロギングの実行と問い合わせ

構文	コマンド	:SYSTem:COMMunicate:LOGGing <1/0/ON/OFF >
クエリ	:SYSTem:COMMunicate:LOGGing?	
応答	<1/0 >	
例	:SYST:COMM:LOGG ON :SYST:COMM:LOGG? 1	

コマンドモニタの実行と問い合わせ

構文	コマンド	:SYSTem:COMMunicate:MONitor < 1/0/ON/OFF >
クエリ	:SYSTem:COMMunicate:MONitor?	
応答	< 1/0 >	
例	:SYST:COMM:MON ON :SYST:COMM:MON? 1	

アドレスの設定と問い合わせ [GP-IB]

構文	コマンド	:SYSTem:COMMunicate:GPIB:ADDRess < アドレス >
クエリ	:SYSTem:COMMunicate:GPIB:ADDRess?	
応答	< アドレス >	
	< アドレス > = 1 ~ 30	

デリミタの設定と問い合わせ [GP-IB]

構文	コマンド	:SYSTem:COMMunicate:GPIB:TERMinator <0/1>
クエリ	:SYSTem:COMMunicate:GPIB:TERMinator?	
応答	<0/1 >	
	<0> = LF+EOI	
	<1> = CR、LF+EOI	
例	:SYST:GPIB:TERM 1 :SYST:GPIB:TERM? 1	

注記 • RS-232C/USB/LAN のデリミタは CR+LF固定です。

ボーレートの設定と問い合わせ [RS-232C]

構文 コマンド **:SYSTem:COMMunicate:RS232C:SPEED < ボーレート >**
クエリ **:SYSTem:COMMunicate:RS232C:SPEED?**
応答 **< ボーレート >**
< ボーレート > = 9600/19200/38400

注記 コマンドを受け付けると、即座に指定されたボーレートが反映されます。

IPアドレスの設定と問い合わせ[LAN]

構文 コマンド **:SYSTem:COMMunicate:LAN:IPADdress < IP アドレス >**
クエリ **:SYSTem:COMMunicate:LAN:IPADdress?**
応答 **< IPアドレス >**
< IPアドレス > = nnn,nnn,nnn,nnn

例 **:SYST:COMM:LAN:IPAD 192,168,0,2**
:SYST:COMM:LAN:UPD
:SYST:COMM:LAN:IPAD?
192,168,0,2

注記 **:SYSTem:COMMunicate:LAN:IPADdress**によりIPアドレスを変更した場合、**:SYSTem:COMMunicate:LAN:UPDate**を実行するまでは、反映されません。

LANポートの設定と問い合わせ[LAN]

構文 コマンド **:SYSTem:COMMunicate:LAN:CONTRol < ポート No. >**
クエリ **:SYSTem:COMMunicate:LAN:CONTRol?**
応答 **< ポートNo. >**
< ポートNo. > = 1 ~ 9999

例 **:SYST:COMM:LAN:CONT 7275**
:SYST:COMM:LAN:UPD
:SYST:COMM:LAN:CONT?
7275

注記 **:SYSTem:COMMunicate:LAN:CONTRol**によりポートNo.を変更した場合、**:SYSTem:COMMunicate:LAN:UPDate**を実行するまでは、反映されません。

サブネットマスクの設定と問い合わせ[LAN]

構文 コマンド **:SYSTem:COMMunicate:LAN:SMASk < サブネットマスク >**
クエリ **:SYSTem:COMMunicate:LAN:SMASk?**
応答 **< サブネットマスク >**
< サブネットマスク > = nnn,nnn,nnn,nnn

例 **:SYST:COMM:LAN:SMAS 255,255,255,0**
:SYST:COMM:LAN:UPDate
:SYST:COMM:LAN:SMAS?
255,255,255,0

注記 **:SYSTem:COMMunicate:LAN:SMASk**によりサブネットマスクを変更した場合、**:SYSTem:COMMunicate:LAN:UPDate**を実行するまでは、反映されません。

デフォルト・ゲートウェイの設定と問い合わせ[LAN]

構文	コマンド	:SYSTem:COMMunicate:LAN:GATeway < アドレス >
	クエリ	:SYSTem:COMMunicate:LAN:GATeway?
	応答	< アドレス >
例		< アドレス > = nnn,nnn,nnn,nnn
		:SYST:COMM:LAN:GAT 192,168,0,100
		:SYST:COMM:LAN:UPD
		:SYST:COMM:LAN:GAT?
		192,168,0,100
注記		:SYSTem:COMMunicate:LAN:GATewayによりデフォルト・ゲートウェイを変更した場合、:SYSTem:COMMunicate:LAN:UPDateを実行するまでは、設定に反映されません。

LAN設定の反映[LAN]

構文	コマンド	:SYSTem:COMMunicate:LAN:UPDate
	例	:SYST:COMM:LAN:IPAD 192,168,0,2 :SYST:COMM:LAN:UPD :SYST:COMM:LAN:IPAD? 192,168,0,2
説明		LANに関連する設定を反映させます。
注記		:SYSTem:COMMunicate:LAN:IPADdress、:SYSTem:COMMunicate:LAN:CONTRol、:SYSTem:COMMunicate:LAN:GATeway、:SYSTem:COMMunicate:LAN:SMASkによりLANに関連する設定を変更した場合、:SYSTem:COMMunicate:LAN:UPDateを実行するまでは、設定に反映されません。

MACアドレスの問い合わせ[LAN]

構文	クエリ	:SYSTem[:COMMunicate:LAN]:MAC?
	応答	< MACアドレス >
例		:SYST:COMM:LAN:MAC? "00-01-67-07-03-85"

(28) データ出力設定

測定同期データ出力機能の設定と問い合わせ

構文	コマンド	:SYSTem:COMMunicate:DATAout <1/0/ON/OFF >
	クエリ	:SYSTem:COMMunicate:DATAout?
	応答	< 1/0 >
説明	<ON> = 外部トリガ(トリガソース<EXTERNAL>)にてトリガ測定が完了すると測定値が自動的に送信されます。	
	内部トリガ(トリガソース<IMMEDIATE>)のときは、[TRIG]キー、または外部IOよりTRIGを入力したときに測定値が自動的に送信されます。	
	<OFF>= 測定値の自動送信が OFF されます。	
注記	データ出力機能に対応しているインタフェースは、USB、LAN、RS-232Cとなります。	

測定データ出力形式の設定と問い合わせ

構文	コマンド	:SYSTem:COMMunicate:DATAout:FORMat < 測定値出力 > , < 日付出力 >
	クエリ	:SYSTem:COMMunicate:DATAout:FORMat?
	応答	< 測定値出力 > , < 日付出力 >
説明	< 測定値出力 > = 0: 電圧測定値、1: 電圧測定値+温度測定値	
	< 日付出力 > = 0: 日付出力無し、1: 日付出力有り	
	測定同期データ出力時のフォーマットを設定します。	
	:TRIG:SOUR EXT	
	:SYST:COMM:DATA ON	
	:SYST:COMM:DATA:FORM 0,0	
	:INIT;*TRG	
	+098.45319E-03	
	:SYST:COMM:DATA:FORM 1,0	
	:INIT;*TRG	
	+098.48965E-03,+25.3E+00	
	:SYST:COMM:DATA:FORM 0,1	
	:INIT;*TRG	
	2015/09/19,13:57:42,+098.44067E-03	
	:SYST:COMM:DATA:FORM 1,1	
	:INIT;*TRG	
	2015/09/19,13:59:15,+098.43942E-03,+25.3E+00	

測定データ出力条件の設定と問い合わせ

構文	コマンド	:SYSTem:COMMunicate:DATAout:CONDition < 出力条件 >
	クエリ	:SYSTem:COMMunicate:DATAout:CONDition?
	応答	< 出力条件 >
	<出力条件> = 0 ~ 4	
	0: 全条件、1: HI判定時、2: IN判定時、3: LO判定時、4: HI-LO判定時	
説明	測定同期データ出力時の条件を設定します。	

日付形式の設定と問い合わせ

構文	コマンド	:SYSTem:LOCALe:DATE < 日付形式 >
	クエリ	:SYSTem:LOCALe:DATE?
	応答	< 日付形式 >
	< 日付形式 > = 0 ~ 2 (0: YYYY/MM/DD, 1: DD/MM/YYYY, 2: MM/DD/YYYY)	

日付区切り形式の設定と問い合わせ

構文	コマンド	:SYSTEM:LOCAL:DATE:SEPARATOR < 区切り形式 >
	クエリ	:SYSTEM:LOCAL:DATE:SEPARATOR?
	応答	< 区切り形式 >
		< 区切り形式 > = 0 ~ 2 { 0: ' / '(スラッシュ), 1: ' - '(ハイフン), 2: ' . '(ピリオド) }

小数点文字形式の設定と問い合わせ

構文	コマンド	:SYSTEM:LOCAL:NUMERIC:DOT < 文字形式 >
	クエリ	:SYSTEM:LOCAL:NUMERIC:DOT?
	応答	< 文字形式 >
		< 文字形式 > = 0/1 { 0: ' . '(ピリオド), 1: ' , '(カンマ) }
注記	ファイル区切り形式がカンマの状態、小数点をカンマ設定にした場合、ファイル区切り形式は自動でセミコロンになります。	

CSVファイル区切り形式の設定と問い合わせ

構文	コマンド	:SYSTEM:LOCAL:NUMERIC:SEPARATOR < 区切り形式 >
	クエリ	:SYSTEM:LOCAL:NUMERIC:SEPARATOR?
	応答	< 区切り形式 >
		< 区切り形式 > = 0 ~ 3 { 0: ' , '(カンマ), 1: ' ; '(セミコロン), 2: ' ' (タブ), 3: ' ' (スペース) }

(29) メモリ機能

測定値(メモリデータ)のフォーマット

(a):R?、:DATA:REMove?の応答

電圧値:単位 V

測定値	±OvrRng	表示時	測定異常時
±□.□□□□□□□□E±□□	±9.90000000E+37	+9.91000000E+37	

注記:測定値は小数点以下 8 桁の浮動小数値で表現されます。

メモリデータの問い合わせ

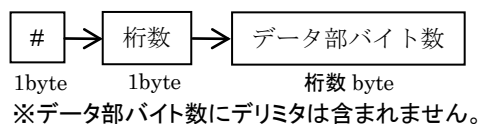
構文	クエリ	応答
	:R? < 読み込みデータ数 >	<ヘッダ><測定値1>,<測定値2>,...,<測定値N>
		読み込みデータ数

※測定値は、取得された順番に出力されます。

参照:測定値(メモリデータ)のフォーマット(p.63)

< 読み込みデータ数 > = 1 ~ 5000

説明 メモリ内のデータを読み込みデータ数を指定して読み込みます。
ヘッダ部は下記のように出力されます。



例 :R? 2
#231+1.87609454E+00,+1.87609717E+00

上記の場合のヘッダ部は、データ部バイト数の桁数は2、データ部のバイト数は31となります。

メモリデータの問い合わせ(待機機能付き)

構文	クエリ	応答
	:DATA:REMove? < 読み込みデータ数 > [, WAIT]	<測定値1>,<測定値2>,...,<測定値N>
		読み込みデータ数

※測定値は、取得された順番に出力されます。

参照:測定値(メモリデータ)のフォーマット(p.63)

< 読み込みデータ数 > = 1 ~

説明 メモリ内のデータを読み込みデータ数を指定して読み込みます。使用可能な測定値が<読み込みデータ数>で指定される数より少ない場合、WAITパラメータが指定されていない限り、エラーとなります。WAITパラメータが指定されている場合、<読み込みデータ数>の測定値が準備できるまで待機します。

例

- 測定メモリ内の測定データが2データ以上の場合
:DATA:REMove? 2
+1.87609454E+00,+1.87609717E+00
- 測定メモリ内の測定データが2データ未満の場合
:DATA:REMove? 2,WAIT
+1.87609454E+00,+1.87609717E+00

注記

- WAITパラメータは、GPIBでは対応していません。実行エラーになります。
- DM7275、DM7276シリーズの内部メモリは、5000データになります。5000以上のデータを<読み込みデータ数>で指定する場合は、必ずWAITパラメータを指定する必要があります。この際、連続測定の設定(p.41)が無効の場合には、実行エラーになります。

イベントを発生させるメモリデータ数の設定と問い合わせ

- 構文** コマンド **:DATA:POINts:EVENT:THReshold < カウント数 >**
 クエリ **:DATA:POINts:EVENT:THReshold?**
 応答 **< カウント数 > = 1 ~ 5000**
- 説明** メモリデータの個数が、設定したカウント数に達した場合に、標準動作レジスタ・グループのイベント・レジスタのビット9を1に設定します。
- 例** **:DATA:POIN:EVENT:THR 3000**
:DATA:POIN:EVENT:THR?
3000
- 注記** 起動時には1にリセットされます。

メモリデータ個数の問い合わせ

- 構文** クエリ **:DATA:POINts?**
 応答 **< メモリデータ数 > = 0 ~ 5000 (NR1)**
- 説明** 本体内部のメモリに保存されている有効測定データの個数を読み込みます。
- 例** **:DATA:POIN?**
+3000

メモリデータのクリア

- 構文** コマンド **:DATA:CLEAr**
- 説明** 本体内部のメモリに保存されている測定データをクリアします

メモリデータをファイルに保存

- 構文** コマンド **:MMEMory:STORe:DATA < ファイル名 >**
< ファイル名 > = “最大8文字の文字列 + 拡張子(3文字)”
- 説明** ファイル名は、拡張子(.csv)を含めて指定しても、含めないで指定しても、コマンドを受け付けます。拡張子を指定しない場合は、.csvにて保存されます。
 メモリデータを< ファイル名 >で指定されたファイルに保存します。
- 例** **:MMEM:STOR:DATA “MEMO_000.csv”**
- メモリデータを MEMO_000.csv に保存します。
- 注記**
- ファイル名の文字列長は、拡張子(.xxx)を含めて12文字以内で指定可能です。(拡張子を含めない場合は8文字以内)それ以外は実行エラーになります。
 - ファイル名に本体で設定できない文字が含まれる場合は実行エラーになります。
 - USBメモリが装着されていない場合は、実行エラーになります。
 - 通信インタフェースがUSBに設定されている場合は、実行エラーになります。

(30) EXT I/O

NPN/PNPスイッチ状態の問い合わせ

構文	クエリ	:IO:MODE?
	応答	<NPN/ PNP>

例 :IO:MODE?
NPN

TRIG/PRINT信号のフィルタ機能実行と問い合わせ

構文	コマンド	:IO:FILTER:STATE <1/0/ON/OFF>
	クエリ	:IO:FILTER:STATE?
	応答	<1/0>

```
例 :IO:FILT:STAT ON
:IO:FILT:STAT?
1
```

TRIG/PRINT信号のフィルタ時間の設定と問い合わせ

構文	コマンド クエリ 応答	:IO:FILTER:TIME < フィルタ時間/MAX/MIN/DEfault > :IO:FILTER:TIME? < フィルタ時間 >
----	-------------------	--

< フィルタ時間 > = 0.05 ~ 0.50 (NRf) [sec]
(MAX : 0.50 MIN : 0.05, DEFault : 0.05)

```
例 :IO:FILT:TIME 0.1
:IO:FILT:TIME?
+1.00000000E-01
```

EOM出力方法の設定と問い合わせ

構文	コマンド	:IO:EOM:MODE < 1/0/ON/OFF >
	クエリ	:IO:EOM:MODE?
	応答	< 0/1 >

0 = HOLDモード(次のトリガ信号による測定を開始するまで保持する)

1 = PULSEモード(設定されたパルス幅でEOM=OFFする)

```
例 :IO:EOM:MODE PULS
:IO:EOM:MODE?
PULSE
```

EOMパルス幅の設定と問い合わせ

構文	コマンド	:IO:EOM:PULSe < パルス幅/MAX/MIN/Default >
	クエリ	:IO:EOM:PULSe?
	応答	< パルス幅 >

< パルス幅 > = 0.001 ~ 0.100 (NRf) [sec]

(MAX : 0.100, MIN : 0.001, DEFault : 0. 005)

```
例 :IO:EOM:PULS 0.005
:IO:EOM:PULS?
+5.00000000E-03
```

外部 I/O 入力の問い合わせ

構文 クエリ **:IO:INPut?**

応答 **0 ~ 3(NR1)**

説明 EXT I/OのTRIG,PRINT端子のONエッジを読み出し、その後クリアします。
エッジを検出したときビットがセットされ、本クエリによって読み出されることで0にクリアされます。また、キーによる入力も信号と同様に検出されます。

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	-	-	-	-	PRINT	TRIG
ピン 番号	-	-	-	-	-	-	26	1

本体取扱説明書 外部制御(EXT I/O)もご覧ください。

注記 入力信号のフィルタ設定をONIにしている時は、フィルタ時間後のエッジを読み出します。

外部 I/O 出力の実行

構文 コマンド **:IO:OUTPut <出力データ0 ~ 2047>**

説明 EXT I/Oの出力モードで判定出力モードを選択している場合にEXT I/O端子から任意の11ビットデータを出力することができます。

	-	-	-	-	-	Bit10	Bit9	Bit8
	-	-	-	-	-	OUT10	OUT9	OUT8
ピン 番号	-	-	-	-	-	36	17	35

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0
ピン 番号	16	34	15	33	14	32	13	31

本体取扱説明書 外部制御(EXT I/O)もご覧ください。

(31) 画面データの保存

表示イメージの取得

構文 クエリ **:HCOPy:SDUMp:DATA?**

応答 **フロント・パネル・ディスプレイのイメージ**

説明 フロント・パネル・ディスプレイのイメージ(「スクリーン・ショット」)を BMP データ(バイナリ形式)で返します。
バイナリデータは以下の図のように、先頭に転送バイト数を出力し、続いて BMP データを出力し、最後にターミネータを出力します。

	1	2	3	4	5	6	7	8	9	10	...		
バイナリ (16進数)	23	36	31	33	31	36	34	30	42	4d	...	0d	0a
内容	#	6	BMP データバイト数						BMP データ			CR	LF

例 **:HCOP:SDUM:DATA?**
#6131640....

(32) *IDN?の応答設定

*IDN?の応答設定

構 文 コマンド **:SYSTem:IDNStr** < 文字列 >

< 文字列 > = 最大127文字の文字列

例 SYST:IDNS "Aaa,Bbb,Ccc,Ddd"

*IDN?

Aaa,Bbb,Ccc,Ddd

SYST:IDNS ""

*IDN?

HIOKI,DM7275-01,123456789,V1.00

- 注 記**
- 文字には、本体で設定可能な文字のみ指定して下さい。
 - < 文字列 >に""を指定すると、HIOKI,・・・となります。
-

他社製品との互換コマンド

スケーリング補正係数の設定と問い合わせ

構文 コマンド **:CALCulate:SCALE:GAIN < 補正係数/MAX/MIN/Default >**
クエリ **:CALCulate:SCALE:GAIN?**
応答 **< 補正係数 >**
< 補正係数 > = -1.0000000E+09~1.0000000E+09 (NRf)
(MAX: 1.0E+09, MIN: -1.0E+09, DEF: 0.0)

例 **:CALC:SCAL:GAIN 2E+00**
:CALC:SCAL:GAIN?
+2.0000000E+00

注記 **:CALCulate:SCALE:PARAMeterA**と同じ機能になります。

スケーリングオフセットの設定と問い合わせ

構文 コマンド **:CALCulate:SCALE:OFFSet < オフセット/MAX/MIN/Default >**
クエリ **:CALCulate:SCALE:OFFSet?**
応答 **< オフセット >**
< オフセット > = -1.0000000E+09~1.0000000E+09 (NRf)
(MAX: 1.0E+09, MIN: -1.0E+09, DEF: 0.0)

例 **:CALC:SCAL:OFFS 1E+03**
:CALC:SCAL:OFFS?
+1.0000000E+03
:CALC:SCAL:PARB?
-1.0000000E+03

注記 **:CALCulate:SCALE:PARAMeterB**とは符号が反転します。

エラー情報の読み出し

構文 クエリ **:SYSTem:ERRor[:NEXT]?**

応答 < エラーNo. >, <エラー名>

< エラーNo. > =

<エラー名> =

説明 エラー番号を読み出し、エラーをクリアします。

応答一覧

0,""

1,"Lower limit is higher than Upper limit."

4,"Unable to change the setting during auto-hold."

5,"Unable to set NULL due to an abnormal measurement value."

30,"Command error."

31,"Execution error. Invalid parameter."

32,"Execution error."

50,"The panel does not exist."

51,"The panel does not exist.Unable to rename."

60,"Cannot use USB memory. Set I/F function to USB-MEMORY."

61,"The drive is not ready. (No USB memory inserted)"

62,"This format is not supported."

64,"Error while reading the configuration file."

65,"File not found."

70,"No space available."

71,"Error occurred saving the file."

76,"Error occurred deleting the file."

77,"Unable to rename the file because another file with the same name already exists."

78,"Error occurred renaming the file."

80,"Unable to enter the adjustment mode."

92,"Memory access error. Turn off the power and restart after a while."

93,"Memory test error."

95,"Backup data error."

96,"Failed to detect line frequency. Select line frequency."

98,"The clock is not set. Reset? (15-01-01 00:00:00)"

99,"Failed to detect line frequency; will be set to 50 Hz."

例 :SYST:ERR?

31, "Execution error. Invalid parameter."

温度測定値の読み出し

構文 クエリ **:MEASure:TEMPerature? < タイプ1/DEFault >,< タイプ2/DEFault >,< 分解能/MAX/MIN/DEFault >**

応答 < 測定値 >

参照:「測定値のフォーマット」(p.32)

< タイプ1 > = FRTD/RTD/FTH/THER

< タイプ2 > = 数値 (NRf)

< 分解能 > = 数値 (NRf)

説明 電圧測定と温度測定を実行し、温度測定値を読み出します。

例 :MEAS:TEMP?

+25.10

注記

- 本メッセージを受け付けると、自動的に測定関連の設定はリセットされ、**:INITiate:CONTinuous OFF**、温度表示がONになります。
- 引数のタイプ1、タイプ2、分解能は、意味を持ちません。
- 引数のタイプ1、タイプ2、分解能は、省略可能です。

以下は、本製品では機能を持ちませんが、コマンドを受け付けます。

電源投入ステータスのクリアと読み出し

構文	コマンド	*PSC < 1 >
	クエリ	*PSC?
説明	応答	1
	コマンドクエリ	機能はありません 固定値1を返します。
例	*PSC 1	
	*PSC?	
	1	

自動校正の実行

構文	コマンド	*CAL?
	応答	+0
説明 自動校正の機能はありませんが、+0が固定で返ります。		
注記 クエリの結果は、+0(固定)となります。		

画像データ形式の設定と問い合わせ

構文	コマンド	:HCOPy:SDUMp:DATA:FORMat < BMP >
	クエリ	:HCOPy:SDUMp:DATA:FORMat?
	応答	BMP
注記 BMP形式のみ対応しています。		

オートゼロ・モードの設定と問い合わせ

構文	コマンド	[[:SENSe]:]VOLTage[:DC]:ZERO:AUTO < 1/0/ON/OFF/ONCE >
	クエリ	[[:SENSe]:]VOLTage[:DC]:ZERO:AUTO?
	応答	1
注記 クエリの結果は、1(固定)となります。		

電圧計完了出力信号のスロープ設定と問い合わせ

構文	コマンド	:OUTPut:TRIGger:SLOPe < POSitive/NEGative >
	クエリ	:OUTPut:TRIGger:SLOPe?
	応答	NEG
注記 スロープ設定は、NEG(固定)となります。		

トリガ回数の設定と問い合わせ

構文	コマンド	:TRIGger:COUNt < 測定回数/MAX/MIN/DEFault >
	クエリ	:TRIGger:COUNt?
	応答	1
例	:TRIG:COUN 1	
	:TRIG:COUN?	
	1	

注記 設定できる測定回数は、1回のみとなります。

リモート状態に設定

構文	コマンド	:SYSTem:REMote
		:SYSTem:RWLock

4 データ取得方法

基本的なデータ取得方法

用途に応じて、柔軟なデータの取り込みが可能です。

フリーランでのデータ取り込み

初期設定	:INITiate:CONTInuous ON (連続測定有効)
	:TRIGger:SOURce IMMEDIATE (内部トリガ)
取り込み	:FETCh?
	過去最新の測定値を取り込む

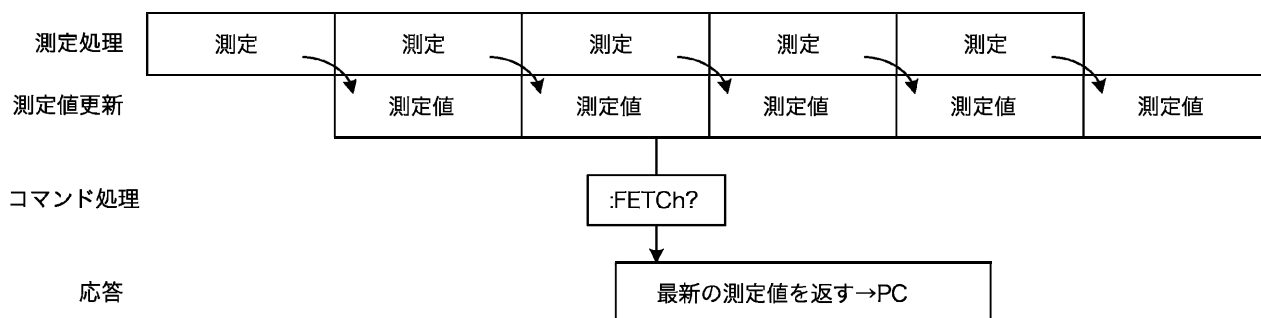
コントローラ(PC、PLC)からトリガしてデータ取り込み

初期設定	:INITiate:CONTInuous OFF (連続測定無効)
	:TRIGger:SOURce IMMEDIATE (内部トリガ)
取り込み	:READ?
	トリガをかけ、測定が終了したら測定値を転送
注記	:READ? を送信すると自動的に :INITiate:CONTInuous OFF になります。

[TRIG] キー選択 または TRIG 信号を入力してデータ取り込み

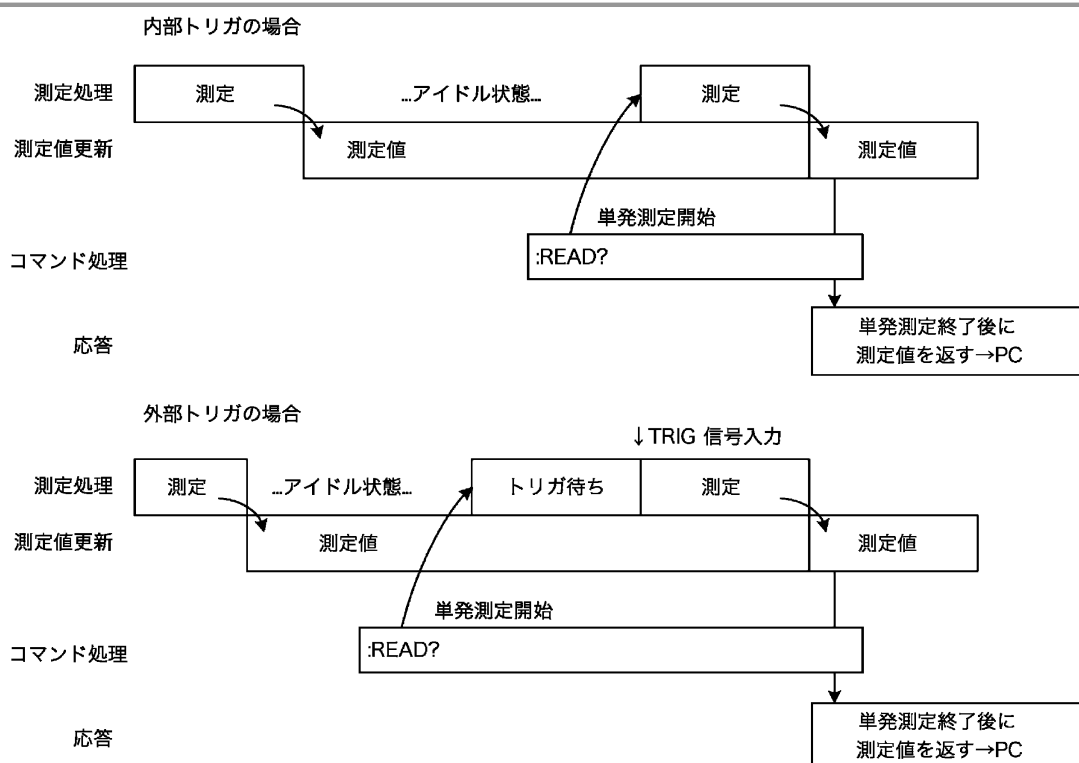
初期設定	:INITiate:CONTInuous OFF (連続測定無効)
	:TRIGger:SOURce EXT (外部トリガ)
取り込み	:READ?
	[TRIG] キーまたは TRIG 信号が入力されたら、測定して値を転送
注記	:READ? を送信すると自動的に :INITiate:CONTInuous OFF になります。

内部トリガ、連続測定ONにて :FETCh?コマンドを使う



最も簡単な測定値取得方法です。測定時間(タクト)に厳しい制約がない場合、外部との同期が不要の場合に最適です。測定対象を接続後、2回分の測定時間を待ってから測定値を取得してください。

連続測定OFFにて :READ?コマンドを使う



コントローラ(PC、PLC)や外部トリガ信号と同期をとって測定(および測定値取得)する方法です。測定時間を最短にすることができます。

5 サンプルプログラム

Visual Basic5.0/6.0とVisual Basic2010で作成する方法を紹介します。
Visual Basicは米国Microsoft社の登録商標です。

Visual Basic 5.0/6.0 で作成する

Microsoft社のVisual Basic 5.0および 6.0用のサンプルソフトです。

通信には、下記を使用しています。

RS-232C/USB通信用: Visual Basic Professional版のMSComm

通信時のターミネータは次のように設定してあるものとします。

RS-232C/USB: CR+LF

RS-232C/USB での通信 (Microsoft Visual Basic Professional MSComm 使用)

■ シンプルな電圧測定

10回測定値を取り込んで、テキストファイルとして保存します。

```
Private Sub MeasureSubRS()
    Dim recvstr As String          ' 受信文字列
    Dim i As Integer

    MSComm1.CommPort = 1          ' COM1 (通信ポートを確認してください)
    MSComm1.Settings = "9600,n,8,1" ' 通信ポートの設定(USBのときは不要)
    MSComm1.PortOpen = True       ' ポートを開きます
    Open App.Path & "¥data.csv" For Output As #1 ' 保存するテキストファイルを開きます

    MSComm1.Output = ":TRIG:SOUR IMM" & vbCrLf ' 内部トリガを選択
    MSComm1.Output = ":INIT:CONT ON" & vbCrLf ' 連続測定 ON
    For i = 1 To 10
        MSComm1.Output = ":FETCH?" & vbCrLf ' 最新の測定値取得 ":FETCH?" を送信
        recvstr = ""                        ' 以下、LF コードが来るまで受信します
        While Right(recvstr, 1) <> Chr(10)
            recvstr = recvstr + MSComm1.Input
            DoEvents
        Wend
        recvstr = Left(recvstr, Len(recvstr) - 2) ' ターミネータ(CR+LF)を削除
        Print #1, Str(i) & "," & recvstr        ' ファイルへ書き出し
    Next

    Close #1
    MSComm1.PortOpen = False
End Sub
```

■ パソコンのキーによって電圧測定

パソコンのキー入力によって、測定および取り込みをして、テキストファイルとして保存します。

```
Private Sub MeasureReadSubRS()
    Dim recvstr As String          ' 受信文字列
    Dim i As Integer

    MSComm1.CommPort = 1          ' COM1 (通信ポートを確認してください)
    MSComm1.Settings = "9600,n,8,1" ' 通信ポートの設定(USBのときは不要)
    MSComm1.PortOpen = True       ' ポートを開きます
    Open App.Path & "¥data.csv" For Output As #1 ' 保存するテキストファイルを開きます

    MSComm1.Output = ":TRIG:SOUR IMM" & vbCrLf ' 内部トリガを選択
    MSComm1.Output = ":INIT:CONT OFF" & vbCrLf ' 連続測定 OFF
    For i = 1 To 10
        ' パソコンのキー入力を待ちます
        ' キー入力チェックルーチンを作成し、キー入力時に、InputKey() = True としてください
        Do While 1
            If InputKey() = True Then Exit Do
            DoEvents
        Loop
        ' キー入力を確認後、測定を 1 回行い、測定値を読み込みます
        MSComm1.Output = ":READ?" & vbCrLf ' 測定 & 測定値取得の ":READ?" を送信
        recvstr = "" ' 以下、LF コードが来るまで受信します
        While Right(recvstr, 1) <> Chr(10)
            recvstr = recvstr + MSComm1.Input
            DoEvents
        Wend
        recvstr = Left(recvstr, Len(recvstr) - 2) ' ターミネータ(CR+LF)を削除
        Print #1, Str(i) & "," & recvstr ' ファイルへ書き出し
    Next

    Close #1
    MSComm1.PortOpen = False
End Sub
```

■ 外部トリガによる測定 1

外部トリガ ([TRIG]キー、TRIG信号入力)またはパソコンのキー入力にて、測定および取り込みをして、テキストファイルとして保存します。

```
Private Sub MeasureTrigSubRS()
    Dim recvstr As String           ' 受信文字列
    Dim i As Integer

    MSComm1.CommPort = 1           ' COM1 (通信ポートを確認してください)
    MSComm1.Settings = "9600,n,8,1" ' 通信ポートの設定(USBのときは不要)
    MSComm1.PortOpen = True        ' ポートを開きます
    Open App.Path & "¥data.csv" For Output As #1 ' 保存するテキストファイルを開きます

    MSComm1.Output = ":TRIG:SOUR EXT" & vbCrLf ' 外部トリガを選択
    MSComm1.Output = ":INIT:CONT OFF" & vbCrLf ' 連続測定 OFF
    For i = 1 To 10
        MSComm1.Output = ":READ?" & vbCrLf ' 測定&測定値取得の ":READ?" を送信
        recvstr = "" ' 以下、LF コードが来るまで受信します
        While Right(recvstr, 1) <> Chr(10)
            recvstr = recvstr + MSComm1.Input
            DoEvents
            ' パソコンのキー入力があったときにトリガ測定を実行するには、
            ' キー入力チェックルーチンを作成し、キー入力時に、InputKey() = True としてください
            If InputKey() = True Then
                MSComm1.Output = "*TRG" & vbCrLf ' キー入力があったら、測定トリガ "*TRG" を送信
            End If
        Wend
        recvstr = Left(recvstr, Len(recvstr) - 2) ' ターミネータ(CR+LF)を削除
        Print #1, Str(i) & ", " & recvstr ' ファイルへ書き出し
    Next

    Close #1
    MSComm1.PortOpen = False
End Sub
```


■ 外部トリガによる測定 2

外部トリガ([TRIG]キー、TRIG信号入力)にて、取り込みをして、テキストファイルとして保存します。(本体は連続測定状態で、トリガの入力のタイミングで最新測定値を取り込みます)

```
Private Sub MeasureTrig2SubRS()
    Dim recvstr As String      ' 受信文字列
    Dim i As Integer

    MSComm1.CommPort = 1      ' COM1 (通信ポートを確認してください)
    MSComm1.Settings = "9600,n,8,1" ' 通信ポートの設定(USBのときは不要)
    MSComm1.PortOpen = True   ' ポートを開きます
    Open App.Path & "¥data.csv" For Output As #1 ' 保存するテキストファイルを開きます

    MSComm1.Output = ":TRIG:SOUR IMM" & vbCrLf ' 内部トリガを選択
    MSComm1.Output = ":INIT:CONT ON" & vbCrLf ' 連続測定 ON

    ' 外部 I/O トリガ入力の確認をクリアする
    MSComm1.Output = ":IO:INP?" & vbCrLf
    recvstr = ""
    While Right(recvstr, 1) <> Chr(10)
        recvstr = recvstr + MSComm1.Input
        DoEvents
    Wend

    For i = 1 To 10
        ' 外部 I/O トリガ入力を待ちます
        Do While 1
            MSComm1.Output = ":IO:INP?" & vbCrLf
            recvstr = ""
            While Right(recvstr, 1) <> Chr(10)
                recvstr = recvstr + MSComm1.Input
                DoEvents
            Wend
            If Left(recvstr, 1) = "1" Then Exit Do
            DoEvents
        Loop
        MSComm1.Output = ":FETCH?" & vbCrLf ' 最新の測定値取得 ":FETCH?" を送信
        recvstr = ""                       ' 以下、LF コードが来るまで受信します
        While Right(recvstr, 1) <> Chr(10)
            recvstr = recvstr + MSComm1.Input
            DoEvents
        Wend
        recvstr = Left(recvstr, Len(recvstr) - 2) ' ターミネータ(CR+LF)を削除
        Print #1, Str(i) & ", " & recvstr        ' ファイルへ書き出し
    Next

    Close #1
    MSComm1.PortOpen = False
End Sub
```

■ 測定条件設定

測定条件をセットします。

```
' 測定条件設定
' 測定条件をセットします
' レンジ: 1V
' サンプリング: FAST
' トリガ: 外部トリガ
' コンパレータ: ON, 上限値 1V, 下限値 0.5V, High および Low でブザーをならす
Private Sub SettingsSubRS()
    MSComm1.CommPort = 1                ' COM1 (通信ポートを確認してください)
    MSComm1.Settings = "9600,n,8,1"    ' 通信ポートの設定(USBのときは不要)
    MSComm1.PortOpen = True            ' ポートを開きます

    MSComm1.Output = ":VOLT:DC:RANG 1E+0" & vbCrLf ' 測定レンジを 1V に設定
    MSComm1.Output = ":VOLT:NPLC FAST" & vbCrLf   ' サンプリングを FAST に設定
    MSComm1.Output = ":TRIG:SOUR EXT" & vbCrLf    ' 外部トリガを選択
    MSComm1.Output = ":INIT:CONT ON" & vbCrLf     ' 連続測定 ON
    MSComm1.Output = ":CALC:LIM:BEEP IN,0,0" & vbCrLf ' INのブザー OFF
    MSComm1.Output = ":CALC:LIM:BEEP HI,1,0" & vbCrLf ' Hiのブザー タイプ1 連続
    MSComm1.Output = ":CALC:LIM:BEEP LO,1,0" & vbCrLf ' Loのブザー タイプ1 連続
    MSComm1.Output = ":CALC:LIM:UPP:ENAB ON" & vbCrLf ' 上限値 有効
    MSComm1.Output = ":CALC:LIM:LOW:ENAB ON" & vbCrLf ' 下限値 有効
    MSComm1.Output = ":CALC:LIM:UPP 1E+0" & vbCrLf  ' 上限値 1V
    MSComm1.Output = ":CALC:LIM:LOW 0.5E+0" & vbCrLf ' 下限値 0.5V
    MSComm1.Output = ":CALC:LIM ON" & vbCrLf       ' コンパレータ ON

    MSComm1.PortOpen = False
End Sub
```

GP-IB での通信 (National Instruments 社の GP-IB ボードを使用)

■ シンプルな電圧測定

10回測定値を取り込んで、テキストファイルとして保存します。

```
Private Sub MeasureSub()
    Dim buffer As String * 20          ' 受信用バッファ
    Dim recvstr As String              ' 受信文字列
    Dim pad As Integer                 ' コントローラアドレス
    Dim gpibad As Integer              ' デバイスアドレス
    Dim timeout As Integer             ' タイムアウト時間
    Dim ud As Integer                  ' 状態(未使用)
    Dim i As Integer

    pad = 0                            ' ボードアドレス0
    gpibad = 1                         ' 本体アドレス1
    timeout = T10s                     ' タイムアウト10秒

    Call ibfind("gpib0", 0)            ' GP-IB 初期化
    Call ibdev(pad, gpibad, 0, timeout, 1, 0, ud)
    Call SendIFC(pad)
    Open App.Path & "¥data.csv" For Output As #1 ' 保存するテキストファイルを開きます

    Call Send(pad, gpibad, ":TRIG:SOUR IMM", NLen) ' 内部トリガを選択
    Call Send(pad, gpibad, ":INIT:CONT ON", NLen) ' 連続測定 ON
    For i = 1 To 10
        Call Send(pad, gpibad, ":FETCH?", NLen) ' 最新の測定値取得 ":FETCH?" を送信
        Call Receive(pad, gpibad, buffer, STOPend) ' 受信
        recvstr = Left(buffer, InStr(1, buffer, Chr(10)) - 1)
        Print #1, Str(i) & "," & recvstr ' ファイルへ書き出し
    Next

    Close #1
    Call ibonl(pad, 0) End Sub
```

■ パソコンのキーによって電圧測定

パソコンのキー入力によって、測定および取り込みをして、テキストファイルとして保存します。

```
Private Sub MeasureReadSub()
    Dim buffer As String * 20          ' 受信用バッファ
    Dim rcvstr As String               ' 受信文字列
    Dim pad As Integer                 ' コントローラアドレス
    Dim gpibad As Integer               ' デバイスアドレス
    Dim timeout As Integer              ' タイムアウト時間
    Dim ud As Integer                  ' 状態(未使用)
    Dim i As Integer

    pad = 0                            ' ボードアドレス 0
    gpibad = 1                          ' 本体アドレス 1
    timeout = T10s                      ' タイムアウト 10秒
    Call ibfind("gpib0", 0)             ' GP-IB 初期化
    Call ibdev(pad, gpibad, 0, timeout, 1, 0, ud)
    Call SendIFC(pad)
    Open App.Path & "¥data.csv" For Output As #1 ' 保存するテキストファイルを開きます

    Call Send(pad, gpibad, ":TRIG:SOUR IMM", NLen) ' 内部トリガを選択
    Call Send(pad, gpibad, ":INIT:CONT OFF", NLen) ' 連続測定 OFF
    For i = 1 To 10
        ' パソコンのキー入力を待ちます
        ' キー入力チェックルーチンを作成し、キー入力時に、InputKey() = True としてください
        Do While 1
            If InputKey() = True Then Exit Do
            DoEvents
        Loop
        ' キー入力を確認後、測定を 1 回行い、測定値を読み込みます
        Call Send(pad, gpibad, ":READ?", NLen) ' 測定&測定値取得の":READ?"を送信
        Call Receive(pad, gpibad, buffer, STOPend) ' 受信
        rcvstr = Left(buffer, InStr(1, buffer, Chr(10)) - 1)
        Print #1, Str(i) & ", " & rcvstr ' ファイルへ書き出し
    Next

    Close #1
    Call ibonl(pad, 0)
End Sub
```

■ 外部トリガによる測定 1

外部トリガ ([TRIG]キー、トリガ信号入力)にて、測定および取り込みをして、テキストファイルとして保存します。

```
Private Sub MeasureTrigSub()
    Dim buffer As String * 20          ' 受信用バッファ
    Dim recvstr As String              ' 受信文字列
    Dim pad As Integer                 ' コントローラアドレス
    Dim gpibad As Integer              ' デバイスアドレス
    Dim timeout As Integer             ' タイムアウト時間
    Dim ud As Integer                  ' 状態(未使用)
    Dim i As Integer

    pad = 0                            ' ボードアドレス 0
    gpibad = 1                         ' 本体アドレス 1
    timeout = T100s                    ' タイムアウト 100 秒(外部トリガ待ちのため)

    Call ibfind("gpib0", 0)            ' GP-IB 初期化
    Call ibdev(pad, gpibad, 0, timeout, 1, 0, ud)
    Call SendIFC(pad)
    Open App.Path & "¥data.csv" For Output As #1 ' 保存するテキストファイルを開きます

    Call Send(pad, gpibad, ":TRIG:SOUR EXT", NLen) ' 外部トリガを選択
    Call Send(pad, gpibad, ":INIT:CONT OFF", NLen) ' 連続測定 OFF
    For i = 1 To 10
        Call Send(pad, gpibad, ":READ?", NLen) ' 測定&測定値取得の ":READ?" を送信
        Call Receive(pad, gpibad, buffer, STOPend) ' 受信
        recvstr = Left(buffer, InStr(1, buffer, Chr(10)) - 1)
        Print #1, Str(i) & ", " & recvstr ' ファイルへ書き出し
    Next

    Close #1
    Call ibonl(pad, 0)
End Sub
```

■ 外部トリガによる測定 2

外部トリガ ([TRIG]キー、TRIG信号入力)にて、取り込みをして、テキストファイルとして保存します。(本体は連続測定状態で、トリガの入力のタイミングで最新測定値を取り込みます)

```
Private Sub MeasureTrig2Sub()
    Dim buffer As String * 20          ' 受信用バッファ
    Dim recvstr As String              ' 受信文字列
    Dim pad As Integer                 ' コントローラアドレス
    Dim gpibad As Integer              ' デバイスアドレス
    Dim timeout As Integer             ' タイムアウト時間
    Dim ud As Integer                  ' 状態(未使用)
    Dim i As Integer

    pad = 0                            ' ボードアドレス 0
    gpibad = 1                         ' 本体アドレス 1
    timeout = T100s                    ' タイムアウト 100 秒(外部トリガ待ちのため)

    Call ibfind("gpib0", 0)            ' GP-IB 初期化
    Call ibdev(pad, gpibad, 0, timeout, 1, 0, ud)
    Call SendIFC(pad)
    Open App.Path & "¥data.csv" For Output As #1 ' 保存するテキストファイルを開きます

    Call Send(pad, gpibad, ":TRIG:SOUR IMM", NLen) ' 内部トリガを選択
    Call Send(pad, gpibad, ":INIT:CONT ON", NLen) ' 連続測定 ON

    ' 外部 I/O トリガ入力の確認をクリアする
    Call Send(pad, gpibad, ":IO:INP?", NLen)
    Call Receive(pad, gpibad, buffer, STOPend)
    recvstr = Left(buffer, InStr(1, buffer, Chr(10)) - 1)
    For i = 1 To 10
        ' 外部 I/O トリガ入力を待ちます
        Do While 1
            Call Send(pad, gpibad, ":IO:INP?", NLen)
            Call Receive(pad, gpibad, buffer, STOPend)
            If Left(buffer, 1) = "1" Then Exit Do
            DoEvents
        Loop
        Call Send(pad, gpibad, ":FETCH?", NLen) ' 最新の測定値取得 ":FETCH?" を送信
        Call Receive(pad, gpibad, buffer, STOPend) ' 受信
        recvstr = Left(buffer, InStr(1, buffer, Chr(10)) - 1)
        Print #1, Str(i) & ", " & recvstr ' ファイルへ書き出し
    Next

    Close #1
    Call ibonl(pad, 0)
End Sub
```

■ 測定条件設定

測定条件をセットします。

```
' 測定条件設定
' 測定条件をセットします
' レンジ: 1V
' サンプリング: FAST
' トリガ: 外部トリガ
' コンパレータ: ON, ABSモード, 上限値 1V, 下限値 0.5V, High および Low でブザーをならす
Private Sub SettingsSub()
    Dim pad As Integer          ' コントローラアドレス
    Dim gpibad As Integer       ' デバイスアドレス
    Dim timeout As Integer      ' タイムアウト時間
    Dim ud As Integer           ' 状態(未使用)

    pad = 0                    ' ボードアドレス 0
    gpibad = 1                  ' 本体アドレス 1
    timeout = T10s              ' タイムアウト 10秒

    Call ibfind("gpib0", 0)     ' GP-IB 初期化
    Call ibdev(pad, gpibad, 0, timeout, 1, 0, ud)
    Call SendIFC(pad)

    Call Send(pad, gpibad, ":VOLT:DC:RANG 1E+0", NLEnd) ' 測定レンジを 1V に設定
    Call Send(pad, gpibad, ":VOLT:NPLC FAST", NLEnd)   ' サンプリングを FASTに設定
    Call Send(pad, gpibad, ":TRIG:SOUR EXT", NLEnd)     ' 外部トリガを選択
    Call Send(pad, gpibad, ":INIT:CONT ON", NLEnd)      ' 連続測定 ON
    Call Send(pad, gpibad, ":CALC:LIM:BEEP IN,0,0", NLEnd) ' INのブザー OFF
    Call Send(pad, gpibad, ":CALC:LIM:BEEP HI,1,0", NLEnd) ' Hiのブザー タイプ1 連続
    Call Send(pad, gpibad, ":CALC:LIM:BEEP LO,1,0", NLEnd) ' Loのブザー タイプ1 連続
    Call Send(pad, gpibad, ":CALC:LIM:UPP:ENAB ON", NLEnd) ' 上限値 有効
    Call Send(pad, gpibad, ":CALC:LIM:LOW:ENAB ON", NLEnd) ' 下限値 有効
    Call Send(pad, gpibad, ":CALC:LIM:UPP 1E+0", NLEnd)   ' 上限値 1V
    Call Send(pad, gpibad, ":CALC:LIM:LOW 0.5E+0", NLEnd) ' 下限値 0.5V
    Call Send(pad, gpibad, ":CALC:LIM ON", NLEnd)         ' コンパレータ ON

    Call ibonl(pad, 0)
End Sub
```

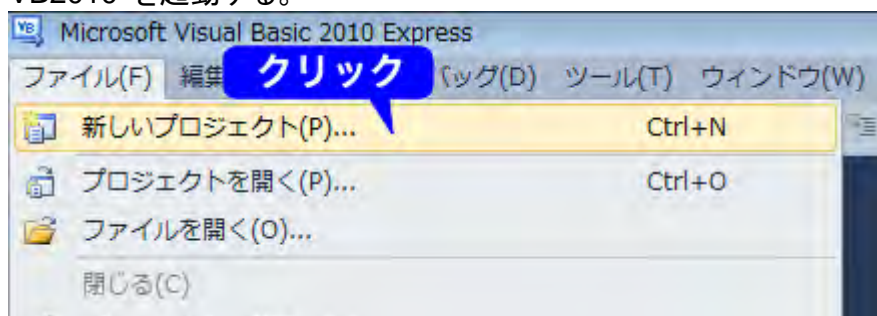
Visual Basic 2010 で作成する

Windowsの開発言語Visual Basic2010 Express Editionを利用して、RS-232C/USB経由でコンピュータから操作して、測定値を取り込んでファイルに保存する方法を例に説明します。
以下、Visual Basic2010をVB2010と記載します。

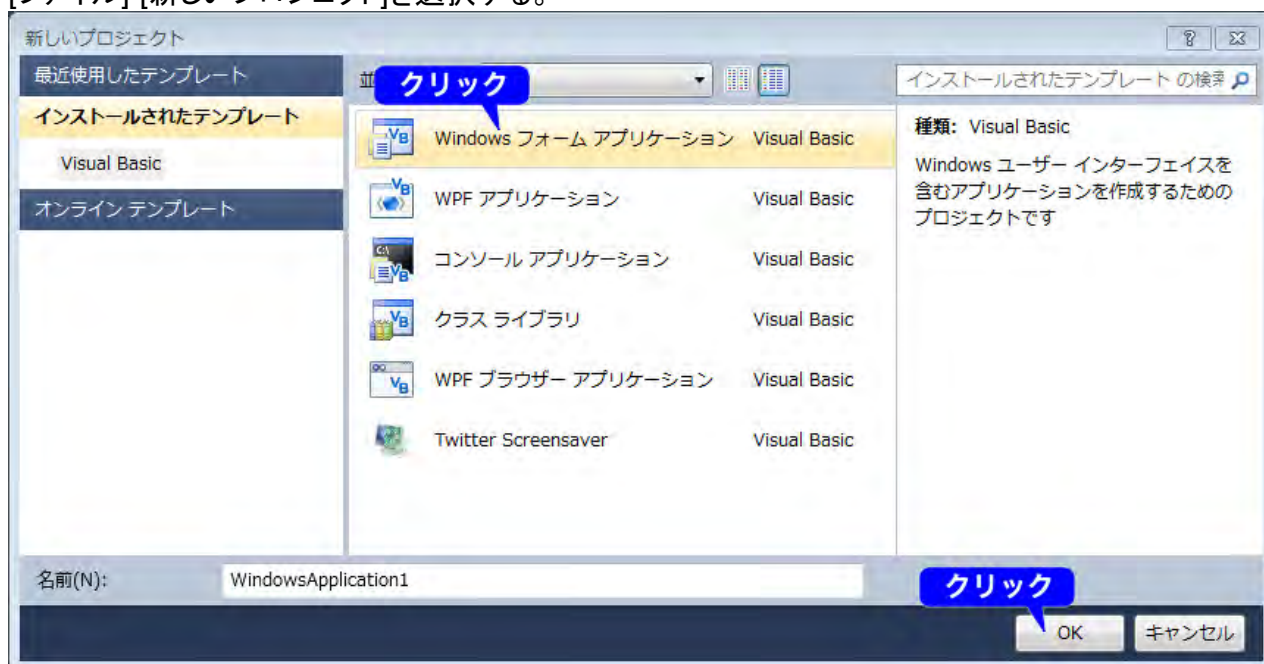
注記 コンピュータやVB2010の環境により、説明が若干異なる場合があります。VB2010の詳しい使用方法についてはVB2010の取扱説明書またはHELPをご覧ください。

1. 新規プロジェクトの作成

1. VB2010 を起動する。



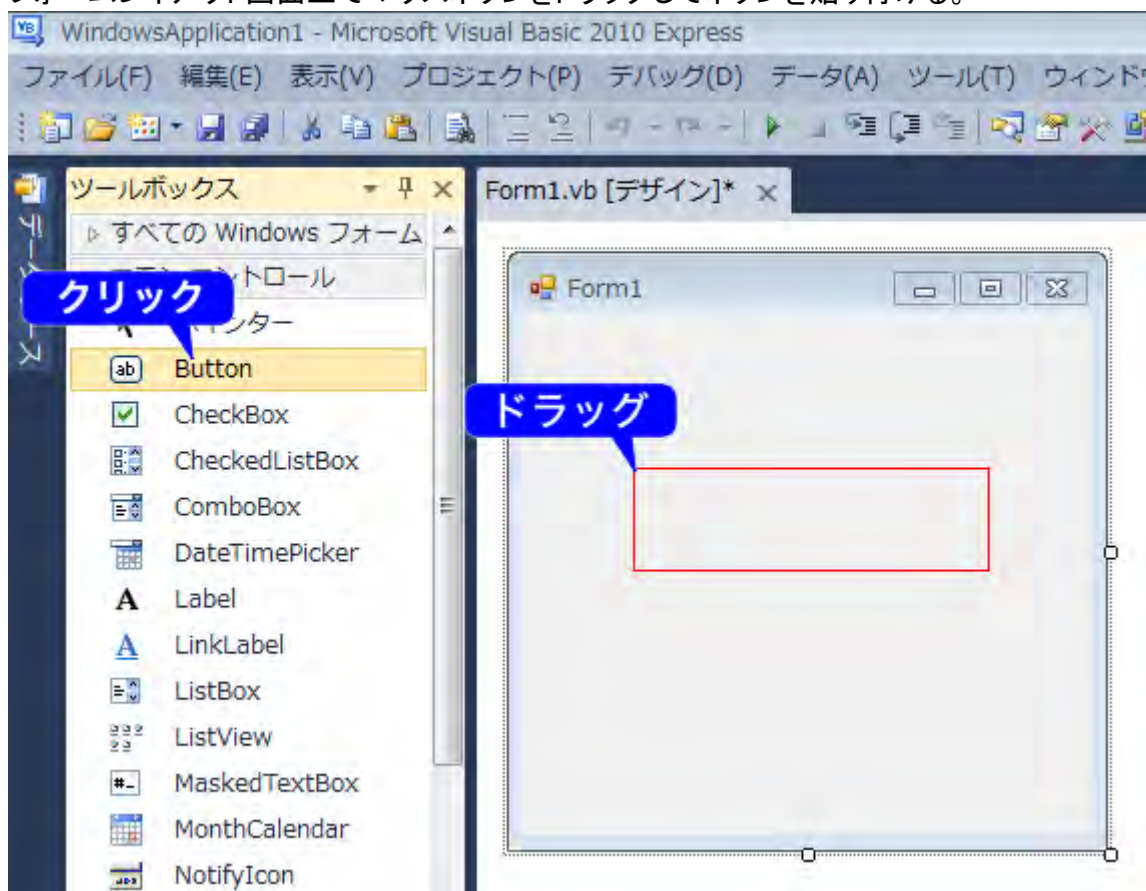
2. [ファイル]-[新しいプロジェクト]を選択する。



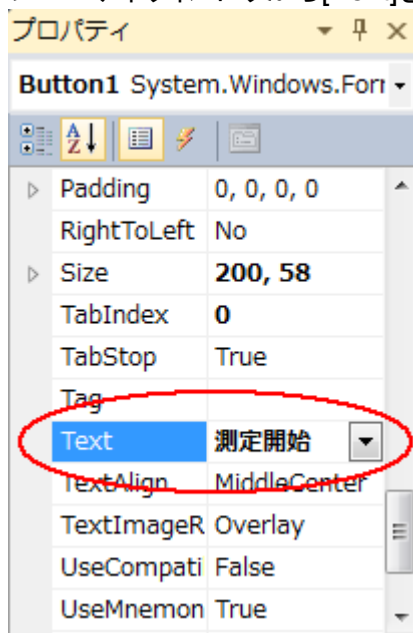
3. テンプレートから[Windows フォーム アプリケーション]を選択する。
4. [OK]ボタンをクリックする。

2. ボタンの配置

1. [ツールボックス]の[コモンコントロール]から[Button]をクリックする。
2. フォームレイアウト画面上でマウスボタンをドラッグしてボタンを貼り付ける。



3. プロパティウィンドウから[Text]を「測定開始」に変更する。

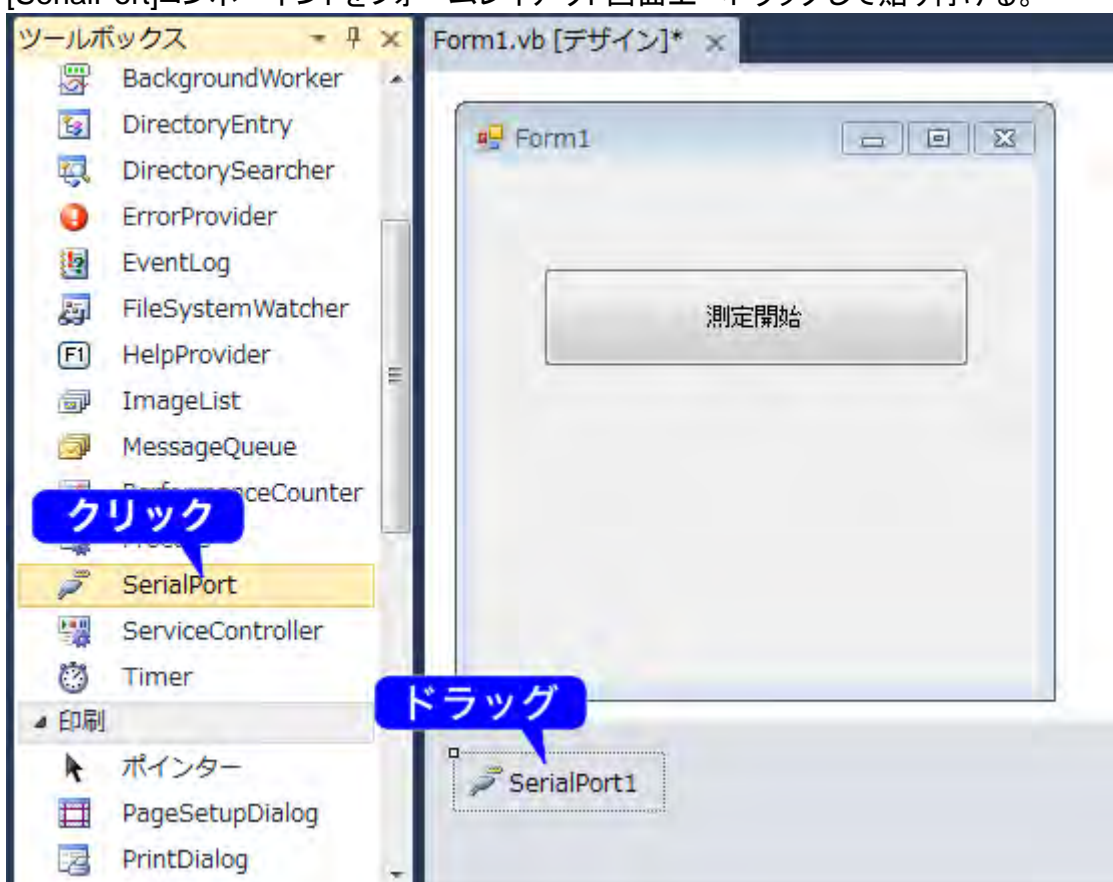


4. フォーム上に[測定開始]ボタンが配置される。

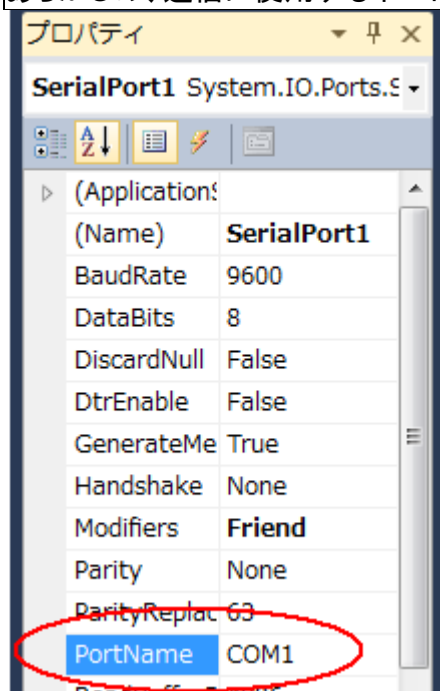


3. シリアル通信コンポーネントの配置

1. [ツールボックス]の[コンポーネント]から[SerialPort]をクリックする。
2. [SerialPort]コンポーネントをフォームレイアウト画面上へドラッグして貼り付ける。

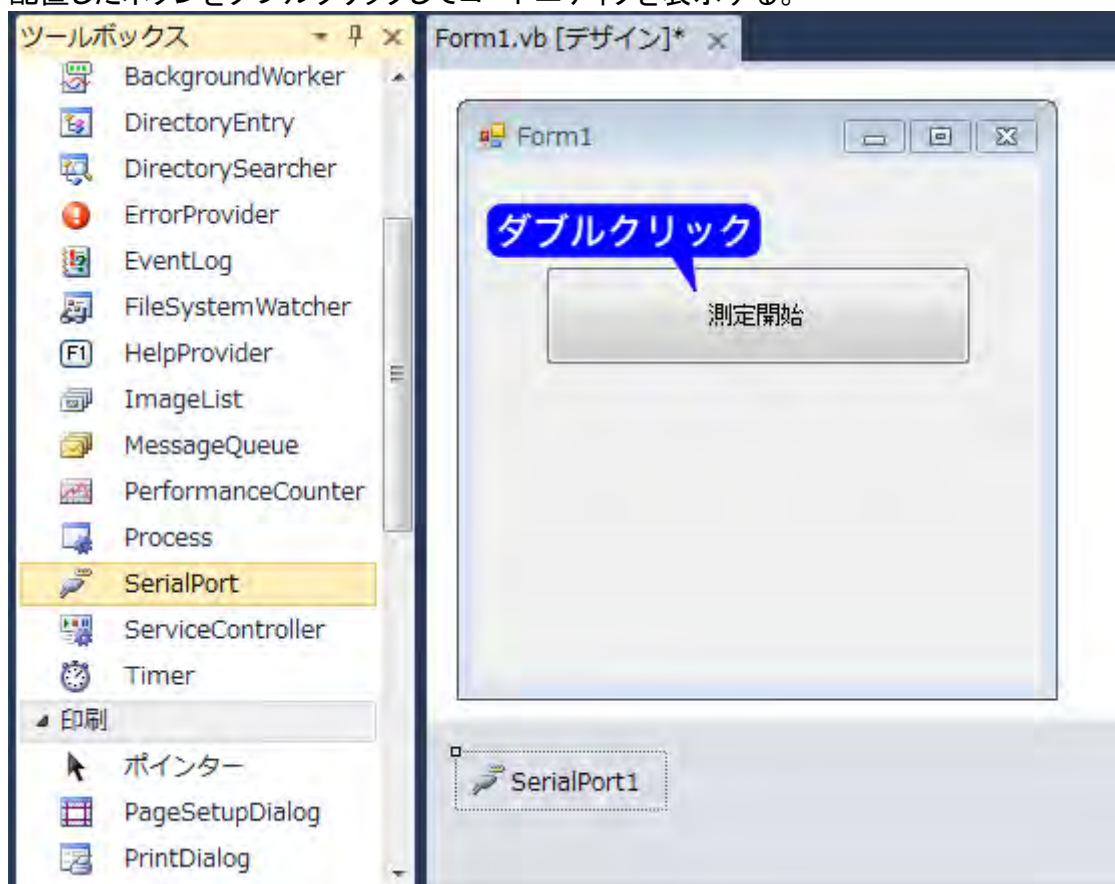


3. プロパティウィンドウから[PortName] を通信に使用するポート名に変更する。
あらかじめ、通信に使用するポートを確認してください

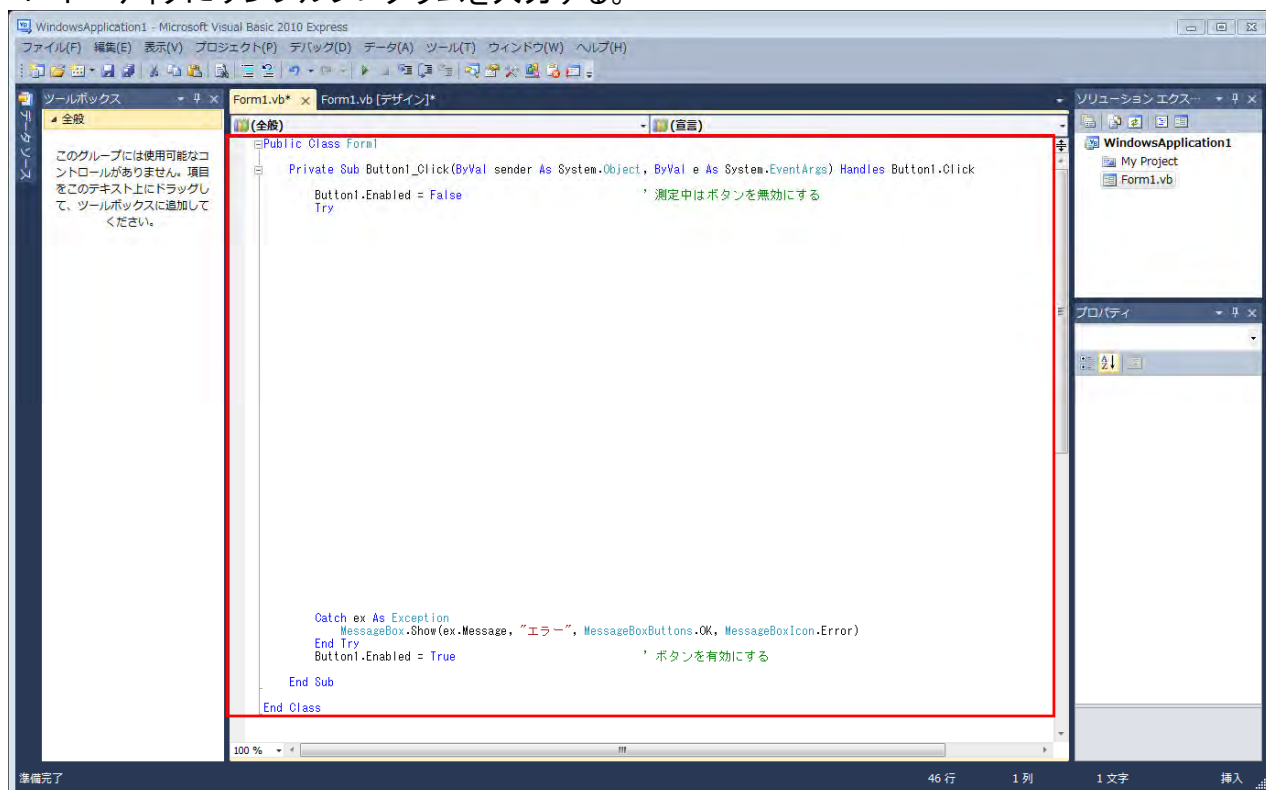


4. コードの記述

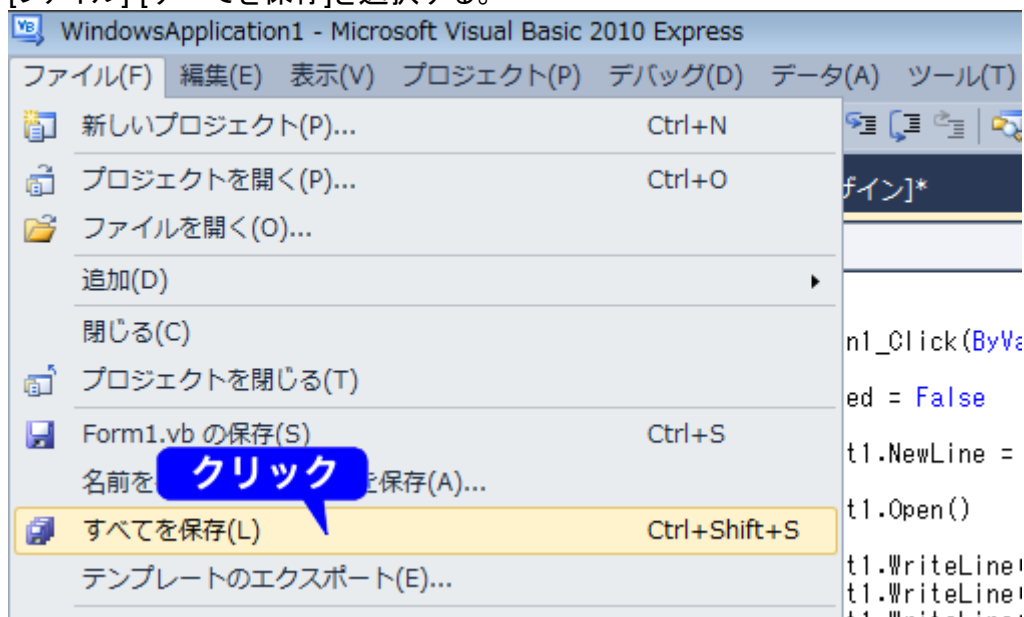
1. 配置したボタンをダブルクリックしてコードエディタを表示する。



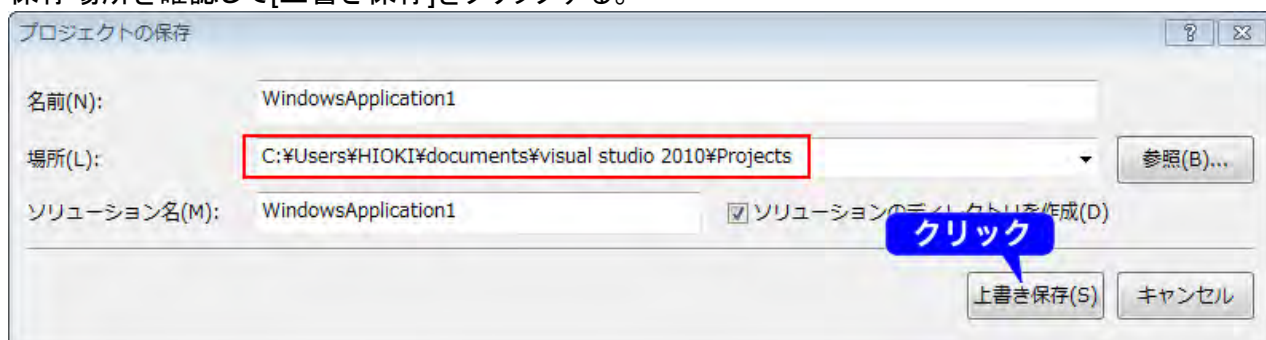
2. コードエディタにサンプルプログラムを入力する。



3. [ファイル]-[すべてを保存]を選択する。



4. 保存場所を確認して[上書き保存]をクリックする。



以下にVB2010でRS-232C/USBの通信を行い、本体の測定条件を設定して、測定した結果を取り込み、ファイルに保存するサンプルプログラムを示します。

なお、サンプルプログラムでは、下記のように記述します。

作成手順の記述	サンプルプログラムでの記述
測定開始のために作成したボタン	「Button1」
アプリケーション終了のために作成したボタン	「Button2」

「測定開始」のボタンを押すと本体で10回測定して、測定値を「data.csv」のファイルに書き出します。

「終了」ボタンを押すとプログラムを終了します。

また以降に示すプログラムは、全て「Form1」のコードとして記述します。

```
Imports System
Imports System.IO
Imports System.IO.Ports

Public Class Form1

    'Button1 を押された場合の処理を行う
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim rcvstr As String
        Dim i As Integer

        Try
            Button1.Enabled = False                                ' 通信中はボタンを押せないようにする.....(a)
            Button2.Enabled = False
            SerialPort1.NewLine = vbCrLf                          ' ターミネータの設定.....(b)
            SerialPort1.ReadTimeout = 2000                        ' タイムアウト2秒 .....(c)
            SerialPort1.Open()                                     ' ポートを開く
            SendSetting(SerialPort1)                              ' 本体の設定
            FileOpen(1, "data.csv", OpenMode.Output)              ' 保存するテキストファイルを作成 .. .....(d)
            For i = 1 To 10
                SerialPort1.WriteLine(":FETCH?")                  ' 最新の測定値取得 ":FETCH?" を送信 .....(e)
                rcvstr = SerialPort1.ReadLine()
                WriteLine(1, rcvstr)                               ' 測定結果の読み込み
                ' ファイルへ書き出し
            Next i
            FileClose(1)                                           ' ファイルを閉じる
            SerialPort1.Close()                                    ' ポートを閉じる
            Button1.Enabled = True
            Button2.Enabled = True
        Catch ex As Exception
            MessageBox.Show(ex.Message, " エラー ", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Sub

    ' 測定条件の設定を行う
    Private Sub SendSetting(ByVal sp As SerialPort)
        Try
            sp.WriteLine(":TRIG:SOUR IMM")                          ' 内部トリガを選択
            sp.WriteLine(":INIT:CONT ON")                          ' 連続測定を ON
        Catch ex As Exception
            MessageBox.Show(ex.Message, " エラー ", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Sub

    'Button2 を押された時、プログラムを終了させる
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        Me.Dispose()
    End Sub
End Class
```

-
- (a) 通信中は「測定開始」ボタンと「終了」ボタンが押せないようにします。
 - (b) 送受信文字列の最後を示すターミネータを CR + LF にします。
 - (c) 読み取り操作時間を 2 秒に設定します。
 - (d) ファイル "data.csv" を開きます。ただし、既に同名のファイルが存在する場合には、以前のファイル "data.csv" は削除され、新規にファイルを作成します。
 - (e) 本体に「1 回測定を行い、その測定結果をコンピュータに返す」コマンドを送信します。
-

6 デバイスの文要書件 [GP-IB]

IEEE488.2 規格に基づく「規格の実行方法に関する情報」

項目	内容
1. IEEE488.1 インタフェース・ファンクションの機能	参照 :「GP-IBの仕様 (インタフェース・ファンクション) 」
2. アドレスを 0 ～ 30以外に設定した場合の動作説明	0 ～ 30以外に設定できません。
3. ユーザが初期設定したアドレスの変更の認識	アドレスの変更は、変更した時点で認識されます。
4. 電源投入時における機器設定の説明	ステータス情報はクリアされます。他はバックアップされます。ただし、ヘッダ、応答メッセージ・ターミネータは初期化されます。
5. メッセージ交換オプションの記述	<ul style="list-style-type: none"> 入力バッファの容量と動作 参照 :「入力バッファ」(p.5) 複数の応答メッセージ単位を返すクエリ :FETCh? :READ? :MEASure[:VOLTage]:DC? :SYSTem:DATE? :SYSTem:TIME? :CALCulate:LIMit:BEEPPer? :SYSTem:PANel:NAME? :SYSTem:PANel:DATE? :CALCulate:TCORrect:PARAmeter? :CALCulate:AVERage:ALL? :CALCulate:AVERage:LIMit? :CALCulate:AVERage:BIN? :CALCulate:AVERage:CP? :SYSTem:COMMunicate:DATAout:FORMat? :R? :DATA:REMove? 文法チェックをしたときに応答を作成するクエリ すべてのクエリは、構文チェックをすると、応答を作成します。 読まれた場合に、応答を作成するクエリの有無 コントローラが読み込む時点で、応答を作成するクエリはありません。 カップリングされるコマンドの有無 該当するコマンドはありません。

項目	内容
6. 機器専用のコマンドを構成する場合に用いる機能的要素の一覧、複合コマンド・プログラム・ヘッダ を使用するのかについての説明	<p>以下を使用します。</p> <ul style="list-style-type: none"> ・プログラム・メッセージ ・プログラム・メッセージ・ターミネータ ・プログラム・メッセージ単位 ・プログラム・メッセージ単位セパレータ ・コマンド・メッセージ単位 ・クエリ・メッセージ単位 ・コマンド・プログラム・ヘッダ ・クエリ・プログラム・ヘッダ ・プログラム・データ ・文字プログラム・データ ・10進数値プログラム・データ ・文字列プログラム・データ ・複合コマンド・プログラム・ヘッダ
7. ブロック・データに関するバッファ容量限界の説明	ブロック・データは使用していません。
8. <表現>内に使用されるプログラム・データ要素の一覧、およびサブ表現の最大ネスティングの程度(機器が<表現>に与える構文規制も含む)	サブ表現は使用していません。使用しているプログラム・データ要素は、文字プログラム・データと10進数値プログラム・データ、文字列プログラム・データです。
9. 各クエリに対する応答構文についての説明	参照 :メッセージリファレンス(p.24)
10. 応答メッセージ要素の原則に従わない機器間メッセージ送信渋滞についての説明	機器対機器のメッセージはありません。
11. ブロック・データの応答容量の説明	ブロック・データの応答はありません。
12. 使用している共通コマンドとクエリの一覧	参照 :メッセージ一覧(p.17)
13. 校正クエリが問題なく完了した以後の機器の状態の説明	*CAL?コマンドは受け付けますが、校正は実行せずに+0を返します。
14. "*DDT"コマンドの有無	*DDTコマンドは使用していません。
15. マクロ・コマンドの有無	マクロは使用していません。
16. 識別に関するクエリ、"*IDN?"クエリに対する応答の説明	参照 :共通コマンド(p.25)
17. "*PUD"コマンド、"*PUD?"クエリが実行される場合による保護されたユーザのデータ保存領域の容量	*PUDコマンド、*PUD?クエリは使用していません。また、ユーザ・データ格納エリアはありません。
18. "*RDT"コマンド、"*RDT?"クエリを使用している場合の資源の説明	*RDTコマンド、*RDT?クエリは使用していません。また、ユーザ・データ格納エリアはありません。
19. "*RST"、"*LRN?"、"*RCL"および"*SAV"の影響を受ける状態についての説明	<p>*LRN?は使用していません。</p> <p>*RSTコマンドは、本器を初期状態に戻します。</p> <p>*SAVコマンドは、測定条件をパネルへ保存します。</p> <p>*RCLコマンドは、パネルから測定条件を読み出します。</p> <p>参照 :共通コマンド(p.25)、初期化項目(p.15)</p>
20. "*TST?"クエリにより実行される自己試験の範囲についての説明	参照 :共通コマンド(p.25)
21. 機器のステータス報告で使用する、ステータス・データの追加構造の説明	参照 :イベント・レジスタ (p.8)
22. 各コマンドが、オーバーラップまたはシーケンシャルコマンドであるかについての説明	すべてのコマンドが、シーケンシャルコマンドです。
23. 各コマンドに対する応答として、操作終了メッセージを生成する時点で要求される機能に関する基準の説明	<p>操作終了は、コマンドの解析時に生成されます。</p> <p>:READ?クエリは、測定データが作成された時点で操作終了します。</p>

- 本書の内容に関しては万全を期していますが、ご不明な点や誤りなどお気づきのことがありましたら、本社コールセンターまたは最寄りの営業所までご連絡ください。
- 本書は改善のため予告なしに記載事項を変更することがあります。
- 本書には著作権によって保護される内容が含まれます。本書の内容を弊社に無断で転載、複製、改変することは禁止されています。

HIOKI

日置電機株式会社

■ 製品のお問い合わせはコールセンターまで

 **0120-72-0560** 9:00～12:00, 13:00～17:00
土・日・祝日を除く

TEL 0268-28-0560 FAX 0268-28-0569 E-mail info@hioki.co.jp

■ 修理・校正のご依頼はお買上店（代理店）または最寄りの営業所まで
また、ご不明な点がございましたらサービスお問合せ窓口まで

TEL 0268-28-0823 FAX 0268-28-0824 E-mail cs-info@hioki.co.jp

■ お問い合わせ・販売ネットワーク

<http://www.hioki.co.jp/contact/>

本社 〒386-1192 長野県上田市小泉 81
TEL 0268-28-0555 FAX 0268-28-0559



1308