

BT4560

BT4560-50

HIOKI

通信コマンド取扱説明書

バッテリーインピーダンスメータ



取扱説明書の最新版



- ✓ 本取扱説明書は、コマンドに関する部分のみを扱っています。
- ✓ BT4560、BT4560-50 の使用前に、必ず BT4560、BT4560-50 取扱説明書をお読みください。
- ✓ BT4560、BT4560-50 の通信設定に関しては、BT4560、BT4560-50 取扱説明書をご覧ください。
- ✓ 本取扱説明書の内容につきましては万全を期していますが、ご不明な点や誤りなどお気づきになった点がございましたら、本社コールセンターまたは最寄りの営業拠点までご連絡ください。

JA

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
初期化項目	11
ローカル機能	11
コマンド実行時間	12
通信時のエラーについて	12
2 メッセージ一覧	13
3 メッセージリファレンス	16
メッセージリファレンスの見方	16
共通コマンド	17
(1) システム・データ・コマンド	17
*IDN?	
(2) 内部動作コマンド	17
*RST	
*TST	
(3) 同期コマンド	18
*OPC	
*WAI	
(4) ステータス、イベント制御コマンド	18
*CLS	
*ESE	
*ESR?	
*SRE	
*STB?	
*TRG	
固有コマンド	21
(1) 機器の形名	21
:QPID	
(2) イベント・ステータス・レジスタ	21
:ESE0	
:ESR0?	
:ESE1	
:ESR1?	
(3) I/O	22
:IO:MODE?	
(4) 測定ファンクション	23
:FUNction	
(5) 測定周波数	23
:FREQuency	
(6) 測定レンジ	24
:RANGe	
(7) サンプリング速度	24
:SAMPle:RATE	
(8) サンプルディレイ	25
:SAMPle:DELaY:MODE	
:SAMPle:DELaY:WAVE	
:SAMPle:DELaY:VOLT	
(9) 電位勾配補正	26
:ADJust:SLOPe	
(10) 電圧リミット	26
:LIMiter	

:LiMiter:VOLTagE	
(11) 測定信号ゼロクロス停止	26
:ZERO:CROSSs	
(12) アベレージ	27
:CALCulate:AVERage	
(13) ゼロアジャスト	27
:ADJust?	
:ADJust:CLEar	
:ADJust:DATA:ALL?	
:ADJust:DATA:SPOT?	
:ADJust:STATe?	
(14) セルフキャリブレーション	30
:CALibration	
:CALibration:AUTO	
(15) コンパレータ	30
:CALCulate:LiMit:STATe	
:CALCulate:LiMit:BEEPer	
:CALCulate:LiMit:ABS	
:CALCulate:LiMit:RESistance	
:CALCulate:LiMit:REACTance	
:CALCulate:LiMit:IMPedance	
:CALCulate:LiMit:PHASe	
:CALCulate:LiMit:VOLTagE	
(16) 測定条件の保存と読み込み	34
:SAVE:/:LOAD	
:SAVE:CLEar	
(17) システムリセット	34
:SYSTem:RESet	
(18) 測定値出力	34
:SYSTem:DATAout	
(19) キー操作音	35
:SYSTem:BEEPer	
(20) キーロック	35
:SYSTem:KLOCK	
(21) 通信設定	35
:SYSTem:LOCal	
(22) ヘッダ	35
:SYSTem:HEADer	
(23) 製造番号	36
:SYSTem:SERial?	
(24) LCD の設定	37
:SYSTem:DISPlay:CONTRast	
:SYSTem:DISPlay:BACKlight	
(25) トリガ	38
:INITiate:CONTInuous	
:TRIGger:SOURce	
:INITiate	
(26) 測定値の読み出し	41
:ABORt	
:MEASure:VALid	
:FETCh?	
:FETCh:TEMPerature?	
:READ?	
4 データ取得方法	45
5 サンプルプログラム	47
Visual Basic 5.0/6.0 で作成する	47
RS-232C/USBでの通信 (Microsoft Visual Basic Professional MSComm 使用)	47
■ シンプルな測定	47

■ パソコンのキーによって測定	48
■ 外部トリガによる測定	49
■ 測定条件設定	50
Visual Basic 2013 で作成する	51

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。応答メッセージは、ヘッダの有無を **:SYSTem:HEADer** コマンドで選択できます。

ヘッダ ON **:RANGE 3.0000E-3**

ヘッダ OFF **3.0000E-3**

(現在の測定レンジは 3 mΩ です。)

電源投入時は、ヘッダ OFF に設定されます。

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。

:FETCH? などヘッダが付かないものもあります。

■ 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。

CALIBRATION OK(ロングフォーム)

CAL OK(ショートフォーム)

CALIB エラー

CA 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

:ESE0

• 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

:SYSTem:RESet

• 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ションマーク "?" が付きます。

:FETCh?

:CALCulate:LIMit:BEEPer?

■ 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。

- CR
- CR+LF

また、応答メッセージのターミネータは以下のものになります。

- CR+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。

:FREQUENCY 1000;*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。

:SYSTEM:HEADER OFF

(3) データ・セパレータ

複数のデータを持つメッセージは、データの間をコンマ (,) で分離します。

:SAMPLE:RATE V MED

■ 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と同様の動作をします。

:SYSTEM:HEADER 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り扱い範囲を超える場合、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。

:ESE0 10

:FETCH?

+1.06571E-03

■ 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:LIMit:RESistance** と **:CALCulate:LIMit:VOLTage** 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:LIMit:**）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:LIMit: RESistance 1.0E-2,5.0E-3 ;:CALCulate:LIMit: VOLTage 5.0,4.0

省略表記

:CALCulate:LIMit: RESistance 1.0E-2,5.0E-3; VOLTage 5.0,4.0

↑
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

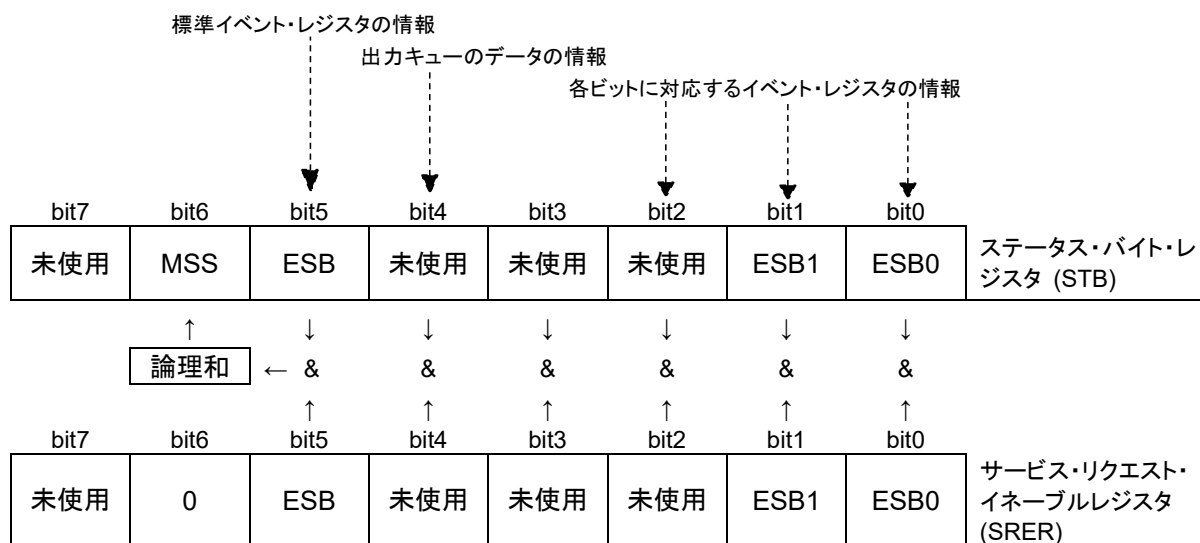
■ 入力バッファ

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、256バイトを超えたデータは受け付けません。

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



サービス・リクエスト発生 の概念図

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、更に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)がセットされます。

■ 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。

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		未使用
ビット6	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	未使用
ビット3		未使用
ビット2		未使用
ビット1	ESB1	イベント・サマリ(論理和)・ビット1 イベント・ステータス・レジスタ1の論理和を表します。
ビット0	ESB0	イベント・サマリ(論理和)・ビット0 イベント・ステータス・レジスタ0の論理和を表します。

■ 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.9)

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。

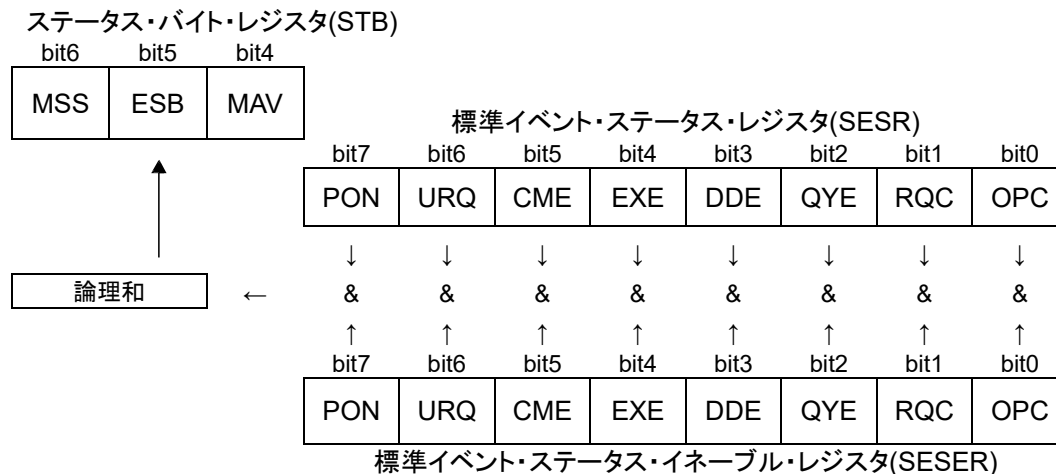
- ***CLS** コマンドを実行したとき
- イベント・レジスタの クエリを実行したとき (***ESR?**)
- 電源を再投入したとき

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	(未使用) URQ	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	(未使用) DDE	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	(未使用) 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"> • 出力キューが空のときに出力キューを読もうとした場合(GP-IBのみ) • 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 • 出力キュー内にデータが存在するときに、次のコマンドを受信した場合
ビット1	(未使用) RQC	コントローラ権の要求
ビット0	OPC	動作の完了 " *OPC "コマンドを実行すると"1"になります。 <ul style="list-style-type: none"> • "*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)



■ 固有のイベント・ステータス・レジスタ (ESR0、ESR1)

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8ビットのレジスタです。

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、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。

- イベント・ステータス・レジスタ0の場合：ステータス・バイト・レジスタ(STB)のビット0(ESB0)が"1"
- イベント・ステータス・レジスタ1の場合：ステータス・バイト・レジスタ(STB)のビット1(ESB1)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
- イベント・ステータス・レジスタのクエリを実行したとき(**:ESR0?**、**:ESR1?**)
- 電源を再投入したとき

イベント・ステータス・レジスタ0 (ESR0)

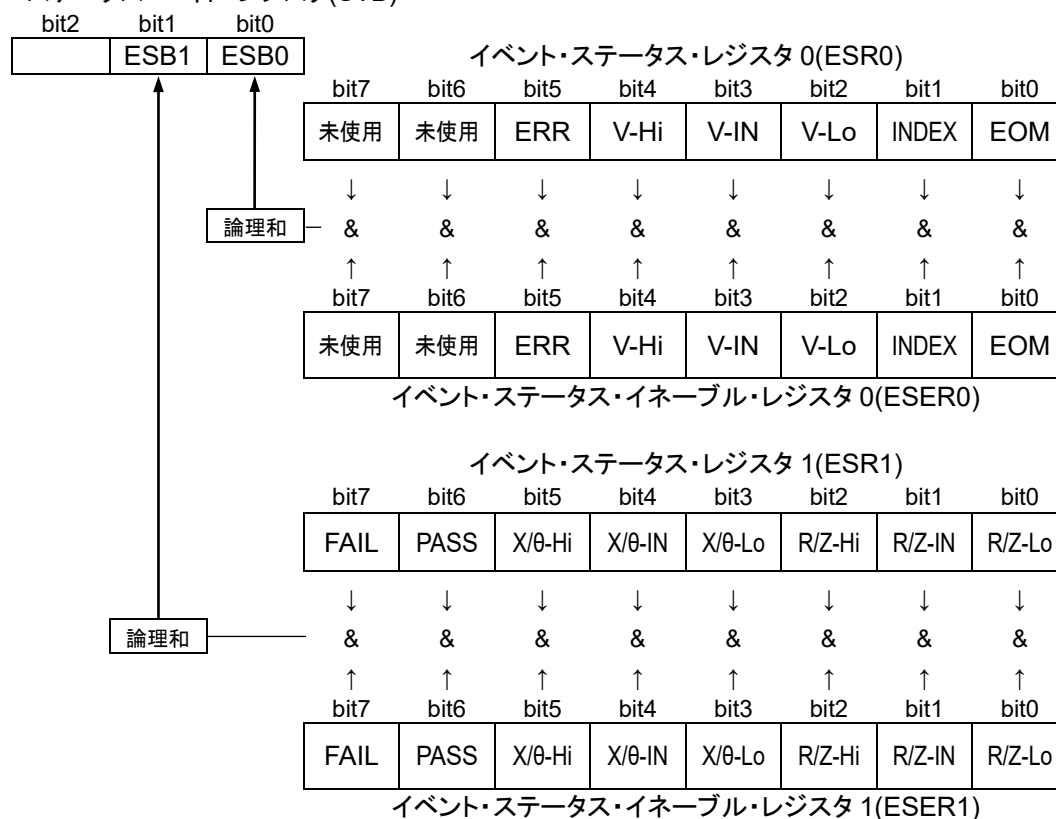
ビット7		未使用
ビット6		未使用
ビット5	ERR	測定異常
ビット4	V-Hi	V測定 コンパレータ結果Hi
ビット3	V-IN	V測定 コンパレータ結果IN
ビット2	V-Lo	V測定 コンパレータ結果Lo
ビット1	INDEX	計測終了
ビット0	EOM	測定終了

イベント・ステータス・レジスタ1 (ESR1)

ビット7	FAIL	総合判定 FAIL
ビット6	PASS	総合判定 PASS
ビット5	X/θ-Hi	X/θ測定 コンパレータ結果Hi
ビット4	X/θ-IN	X/θ測定 コンパレータ結果IN
ビット3	X/θ-Lo	X/θ測定 コンパレータ結果Lo
ビット2	R/Z-Hi	R/Z測定 コンパレータ結果Hi
ビット1	R/Z-IN	R/Z測定 コンパレータ結果IN
ビット0	R/Z-Lo	R/Z測定 コンパレータ結果Lo

イベント・ステータス・レジスタ0(ESR0)、1(ESR1)と
イベント・ステータス・イネーブル・レジスタ0(ESER0)、1(ESER1)

ステータス・バイト・レジスタ(STB)



■ 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
イベント・ステータス・レジスタ0	:ESR0?	-
イベント・ステータス・イネーブル・レジスタ0	:ESE0?	:ESE0
イベント・ステータス・レジスタ1	:ESR1?	-
イベント・ステータス・イネーブル・レジスタ1	:ESE1?	:ESE1

初期化項目

項目	初期化方法	電源投入時	*RST コマンド (ノーマルリセット)	:SYSTem:RESET コマンド (システムリセット)	*CLS コマンド	工場出荷時
RS-232C 設定(通信速度)		-	-	-	-	9600
デバイス固有の機能(レンジなど)		-	●*1	●	-	●
出力キュー		●	-	-	-	●
入力バッファ		●	-	-	-	●
ステータス・バイト・レジスタ		●	-	-	●	●
イベント・レジスタ		●*2	-	-	●	●
イネーブル・レジスタ		●	-	-	-	●
カレント・パス		●	-	-	-	●
ヘッダON/ OFF		OFF	-	-	-	OFF

*1. パネルセーブデータ、ゼロアジャスト値は除く

*2. PON ビット(ビット7)は除く。

ローカル機能

通信中はリモート状態となり、測定画面に[RMT]が表示され、操作キーは無効になります。

EXT	10mΩ	Z:MED	V:MED	RMT	
R	6.0000	mΩ			
X	-0.5000	mΩ			
V	4.00000	V	1000 Hz	25.6 °C	

■ リモート状態を解除する場合

・正面パネルの[LOCAL]キーを押すと、リモート状態は解除され、キー操作が可能となります。

LOCAL

・:SYSTem:LOCal コマンドを送信することによってもリモート状態を解除することができます。

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。

- 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**、**:CAL**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。転送時間は、コントローラにより異なります。

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]

測定値が11文字の場合は、1データの転送時間は11/Tになります。

(例)9600bpsの場合11/(9600/10) = 約11ms

コマンド	実行時間 (通信時間除く) *1
:ADJust? ALL	20 s以内
:FETCh?	4 ms以内
:READ?	測定時間 + 4 ms以内
:LOAD	90 ms以内
:CALibration	キャリブレーション時間 + 6 ms以内
*RST	75 ms以内

*1 測定を行っていない状態での参考値です。測定中に実行すると実行時間が長くなる場合があります。

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
- **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 []:省略可能	データ部 []:省略可能、():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 の問い合わせ
*RST		機器の初期化(ノーマルリセット:通信設定、ゼロアジャスト値、パネルセーブデータを除く設定の初期化)
*TST?	(0~1)	セルフテストと結果問い合わせ
*OPC		動作終了時の OPC の設定
*OPC?	(1)	動作終了の問い合わせ
*WAI		動作終了待ちウェイト
*CLS		ステータス・バイト・レジスタと関連キューのクリア(出力キューを除く)
*ESE	0~255	標準イベント・ステータス・イネーブル・レジスタの設定
*ESE?	(0~255)	標準イベント・ステータス・イネーブル・レジスタの問い合わせ
*ESR?	(0~255)	標準イベント・ステータス・レジスタの問い合わせ
*SRE	0~255	サービス・リクエスト・イネーブル・レジスタの設定
*SRE?	(0~255)	サービス・リクエスト・イネーブル・レジスタの問い合わせ
*STB?	(0~255)	ステータス・バイト・レジスタの問い合わせ
*TRG		サンプリングの要求
機器の形名		
:QPID	〈モデル名〉	機器の形名問い合わせ
イベントレジスタ		
:ESE0	0~255	イベント・ステータス・イネーブル・レジスタ 0 の設定
:ESE0?	(0~255)	イベント・ステータス・イネーブル・レジスタ 0 の問い合わせ
:ESR0?	(0~255)	イベント・ステータス・レジスタ 0 の問い合わせ
:ESE1	0~255	イベント・ステータス・イネーブル・レジスタ 1 の設定
:ESE1?	(0~255)	イベント・ステータス・イネーブル・レジスタ 1 の問い合わせ
:ESR1?	(0~255)	イベント・ステータス・レジスタ 1 の問い合わせ
I/O		
:IO:MODE?	(NPN/PNP)	NPN/PNP スイッチ状態の問い合わせ
測定ファンクション		
:FUNction	RV/ZV/R/Z/V	測定ファンクションの設定
:FUNction?	(RV/ZV/R/Z/V)	測定ファンクションの問い合わせ
測定周波数		
:FREQuency	〈周波数〉	測定周波数の設定
:FREQuency?	(〈周波数〉)	測定周波数の問い合わせ
測定レンジ		
:RANGe	〈測定レンジ〉	測定レンジの設定
:RANGe?	(〈測定レンジ〉)	測定レンジの問い合わせ
サンプリング速度		
:SAMPlE:RATE	〈V/Z〉, 〈FAST/MEDium/SLOW〉	サンプリング速度の設定
:SAMPlE:RATE?	〈V/Z〉 (〈FAST/MEDIUM/SLOW〉)	サンプリング速度の問い合わせ
サンプルディレイ		
:SAMPlE:DELAy:MODE	WAVE/VOLTage	サンプルディレイモードの設定
:SAMPlE:DELAy:MODE?	(WAVE/VOLTAGE)	サンプルディレイモードの問い合わせ
:SAMPlE:DELAy:WAVE	〈波数〉= 0.0~9.0	サンプルディレイ波数の設定
:SAMPlE:DELAy:WAVE?	(〈波数〉)	サンプルディレイ波数の問い合わせ
:SAMPlE:DELAy:VOLTage	〈電圧勾配偏差〉=0.001~10.000	サンプルディレイ電圧勾配偏差の設定
:SAMPlE:DELAy:VOLTage?	(〈電圧勾配偏差〉)	サンプルディレイ電圧勾配偏差の問い合わせ

メッセージ []:省略可能	データ部 []:省略可能、():応答データ	説明
電圧リミット		
:LIMiter	1/0/ON/OFF	電圧リミット機能の設定
:LIMiter?	(ON/OFF)	電圧リミット機能の問い合わせ
:LIMiter:VOLTage	<電圧リミット値>=0.01~5.00	電圧リミット値の設定
:LIMiter:VOLTage?	(<電圧リミット値>)	電圧リミット値の問い合わせ
測定信号ゼロクロス停止		
:ZERO:CROSSs	1/0/ON/OFF	測定信号ゼロクロス停止機能の設定
:ZERO:CROSSs?	(ON/OFF)	測定信号ゼロクロス停止機能の問い合わせ
アベレージ		
:CALCulate:AVERage	<アベレージ回数>=1~99	測定アベレージの設定
:CALCulate:AVERage?	(<アベレージ回数>)	測定アベレージの問い合わせ
ゼロアジャスト		
:ADJust?	<SPOT/ALL> (0/1)	ゼロアジャストの実行と結果の問い合わせ
:ADJust:CLEar		ゼロアジャストのクリア
:ADJust:DATA:ALL?	V/R/RV (<電圧アジャスト値>,<10Hz アジャスト値 R>,<10Hz アジャスト値 X>,<100Hz アジャスト値 R>,<100Hz アジャスト値 X>,<330Hz アジャスト値 R>,<330Hz アジャスト値 X>,<660Hz アジャスト値 R>,<660Hz アジャスト値 X>,<1kHz アジャスト値 R>,<1kHz アジャスト値 X>)	ゼロアジャスト値の問い合わせ(ALL 補正)
:ADJust:DATA:SPOT?	V/R/RV (<電圧アジャスト値>,<アジャスト値 R>,<アジャスト値 X>)	ゼロアジャスト値の問い合わせ(SPOT 補正)
:ADJust:STATe?	(ON/OFF)	ゼロアジャスト実行状態の問い合わせ
電位勾配補正		
Adjust:SLOPe	1/0/ON/OFF	交流信号応答における勾配補正の設定
Adjust:SLOPe	(ON/OFF)	交流信号応答における勾配補正の問い合わせ
セルフキャリブレーション		
:CALibration		セルフキャリブレーションの実行
:CALibration:AUTO	1/0/ON/OFF	オートセルフキャリブレーションの設定
:CALibration:AUTO?	(ON/OFF)	オートセルフキャリブレーションの問い合わせ
コンパレータ		
:CALCulate:LIMit:STATe	1/0/ON/OFF	コンパレータの設定
:CALCulate:LIMit:STATe?	(ON/OFF)	コンパレータの問い合わせ
:CALCulate:LIMit:BEEPer	OFF/HL/IN/ALL	ブザー音の設定
:CALCulate:LIMit:BEEPer?	(OFF/HL/IN/ALL)	ブザー音の問い合わせ
:CALCulate:LIMit:ABS	1/0/ON/OFF	コンパレータの電圧絶対値判定の問い合わせ
:CALCulate:LIMit:ABS?	(ON/OFF)	コンパレータの電圧絶対値判定の設定
:CALCulate:LIMit:RESistance	<上限値>,<下限値>	抵抗成分の上下限値の設定
:CALCulate:LIMit:RESistance?	(<上限値>,<下限値>)	抵抗成分の上下限値の問い合わせ
:CALCulate:LIMit:REACtance	<上限値>,<下限値>	リアクタンス成分の上下限値の設定
:CALCulate:LIMit:REACtance?	(<上限値>,<下限値>)	リアクタンス成分の上下限値の問い合わせ
:CALCulate:LIMit:IMPedance	<上限値>,<下限値>	インピーダンス成分の上下限値の設定
:CALCulate:LIMit:IMPedance?	(<上限値>,<下限値>)	インピーダンス成分の上下限値の問い合わせ
:CALCulate:LIMit:PHASe	<上限値>,<下限値>	位相成分の上下限値の設定
:CALCulate:LIMit:PHASe?	(<上限値>,<下限値>)	位相成分の上下限値の問い合わせ
:CALCulate:LIMit:VOLTage	<上限値>,<下限値>	電圧成分の上下限値の設定
:CALCulate:LIMit:VOLTage?	(<上限値>,<下限値>)	電圧成分の上下限値の問い合わせ
測定条件の保存と読み込み		
:SAVE	保存 No.	測定条件の保存
:SAVE:CLEar	保存 No.	保存されている測定条件の削除
:LOAD	保存 No.	測定条件の読み出し

メッセージ []:省略可能	データ部 []:省略可能、():応答データ	説明
システムリセット		
:SYSTem:RESet		機器の初期化(システムリセット:通信設定を除く設定の初期化)
測定完了後、測定値を出力する		
:SYSTem:DATAout	1/0/ON/OFF	測定値出力の設定
:SYSTem:DATAout?	(ON/OFF)	測定値出力の問い合わせ
キー操作音		
:SYSTem:BEEPer	1/0/ON/OFF	キー操作音の設定
:SYSTem:BEEPer?	(ON/OFF)	キー操作音の問い合わせ
キーロック		
:SYSTem:KLOCK	1/0/ON/OFF	キーロックの設定
:SYSTem:KLOCK?	(ON/OFF)	キーロックの問い合わせ
通信設定		
:SYSTem:LOCal		通信中(リモート)状態解除
ヘッダ有無		
:SYSTem:HEADer	1/0/ON/OFF	ヘッダ有無の設定
:SYSTem:HEADer?	(ON/OFF)	ヘッダ有無の問い合わせ
製造番号		
:SYSTem:SERIal?	製造番号	製造番号の問い合わせ
LCD の設定		
:SYSTem:DISPlay:CONTRast	<コントラスト>=1~100	コントラストの設定
:SYSTem:DISPlay:CONTRast?	(<コントラスト>)	コントラストの問い合わせ
:SYSTem:DISPlay:BACKlight	<輝度>=10~100	バックライト輝度の設定
:SYSTem:DISPlay:BACKlight?	(<輝度>)	バックライト輝度の問い合わせ
トリガ		
:TRIGger:SOURce	IMMediate/EXTErnal	トリガソースの設定
:TRIGger:SOURce?	(IMMediate/EXTErnal)	トリガソースの問い合わせ
:INITiate:CONTInuous	1/0/ON/OFF	連続測定の設定(アイドル状態への遷移の許可/禁止)
:INITiate:CONTInuous?	(ON/OFF)	連続測定の問い合わせ
:INITiate		トリガ待ち状態への遷移
測定値の読み出し		
:ABORt		測定の中断(強制終了)
:MEASure:Valid	<MR0>=1~7	測定値読み出しクエリの応答データの設定
:MEASure:Valid?	(<MR0>)	測定値読み出しクエリの応答データの問い合わせ
:FETCh?	(<総合判定結果>,<測定値>,<判定結果>,<測定値>,<判定結果>,...)	最後の測定値の問い合わせ
:FETCh:TEMPerature?	(<温度測定値>)	温度測定値の問い合わせ
:READ?	(<総合判定結果>,<測定値>,<判定結果>,<測定値>,<判定結果>,...)	アイドル状態の解除、測定終了後の測定値の問い合わせ

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。
通常(HEADER コマンドを除く)は
HEADER OFFのときの説明をします。

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

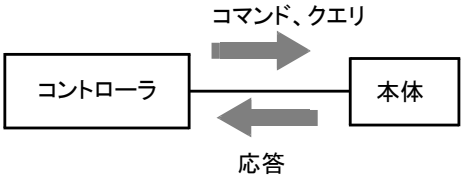
構文	コマンド	*ESE <0 ~ 255 (NR1)>
	クエリ	*ESE?
	応答	<0 ~ 255 (NR1)>

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,BT4560,123456789,V1.00
機器IDは、HIOKI、BT4560、製造番号は、123456789、ソフトウェアバージョン1.00です。

(2) 内部動作コマンド

機器の初期化(ノーマルリセット)

構 文	コマンド	*RST
-----	------	-------------

説 明 コマンド 通信設定、ゼロアジャスト値、パネルセーブデータを除き、機器を工場出荷時の設定にします。(ノーマルリセット)
初期化後は、初期画面に戻ります。

注 記 通信設定、ゼロアジャスト値、パネルセーブデータは初期化されません。
ゼロアジャスト値、パネルセーブデータも初期化したい場合は、**:SYSTem:RESet**
コマンド(p.34)を送信してください。

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	<0 ~ 1(NR1)>

<0> = エラーなし
<1> = ROMエラー

説 明 本体のセルフテストを行い、その結果を 0 ~ 1のNR1数値で返します。
エラーなしの場合、0 を返します。

例 ***TST?**
1
ROMエラーが発生しています。

(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)をセットします

例 **A;B;*OPC;C**

A,Bコマンド処理終了後、SESRのOPCをセットします。

実行中の全動作終了後、出力キューにASCIIの1を応答

構文 クエリ ***OPC?**
応答 1

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。

例 **:TRIG:SOUR EXT**
:INIT:CONT ON
***TRG;*WAI;FETC?**

トリガによる測定を終了を待つて最新の測定値を取得します。

注記 ***WAI**コマンドは、IEEE 488.2 – 1987規格の共通コマンドであるため受け付けますが、***TRG**, **:INITiate**以外の機器固有のコマンドは、シーケンシャル型のコマンドを使用しているため、***WAI**コマンドを使用してもそれによる効果はありません。

(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。(SESR、ESR0、ESR1)

注記 出力キューは影響されません。

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	未使用	未使用	未使用	OPC

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	未使用	未使用	未使用	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) の設定と問い合わせ

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、未使用ビット(ビット 2、3、7)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2、3、7)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
未使用	0	ESB	未使用	未使用	未使用	ESE1	ESE0

例 ***SRE 33**
SRERのビット0とビット5を1にします。

***SRE?**
33
SRERのビット0とビット5が1になっています。

ステータス・バイトと MSS ビットの問い合わせ

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。
応答メッセージにヘッダは付きません。

128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
未使用	MSS	ESB	未使用	未使用	未使用	ESE1	ESE0

例 ***STB?**
1
STBのbit0が1になっています。

サンプリングの要求

構文 コマンド ***TRG**

説明 外部トリガ(:TRIGger:SOURce <EXTERNAL>)のときに 1 回測定を行います。
参照: データ取得方法(p.45)、トリガ(p.38)

例 :TRIG:SOUR EXT
:INIT:CONT ON
*TRG;*WAI;;FETC?
トリガによる測定の終了を待って最新の測定値を取得します。

注記

- トリガソースが内部トリガ(:TRIGger:SOURce <IMMEDIATE>)の場合に本コマンドを実行すると実行エラーとなります。
- 連続測定がオフ(:INITiate:CONTinuous <OFF>)の場合は本コマンドによる測定は行えません。:INITIATE(または:READ?)実行後、EXT.I/O端子からTRIG信号を入力して測定を行ってください。
- *TRG;*WAI;;FETC?(*WAIは*OPC/*OPC?でもよい)、:READ?中に測定を中断するには、LOCALキーを押してください。LOCALキー受付後に測定値の応答を返すため、[RMT]表示が消えない場合があります。その場合表示が消えるまでLOCALキーを押してください。

固有コマンド

(1) 機器の形名

機器の形名の問い合わせ

構文 クエリ **:QPID**
応答 **<モデル名>**

例 :QPID
BT4560
機器の形名はBT4560です。

(2) イベント・ステータス・レジスタ

固有のイベント・ステータス・イネーブル・レジスタ ESER0 の設定と問い合わせ

構文 コマンド **:ESE0 <0 ~ 255(NR1)>**
クエリ **:ESE0?**
応答 **<0 ~ 255(NR1)>**

説明 コマンド イベント・ステータス・イネーブルレジスタ 0(ESER0) にイベント・ステータス・レジスタの使用可能パターンを設定します。

クエリ イベント・ステータス・イネーブルレジスタ 0(ESER0) にイベント・ステータス・レジスタの使用可能パターンを問い合わせます。

128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
未使用	未使用	ERR	V-Hi	V-IN	V-Lo	INDEX	EOM

注記 電源投入の時はデータを0に初期化します。

固有のイベント・ステータス・レジスタ ESR0 の問い合わせ

構 文 クエリ **:ESR0?**
 応答 <0 ~ 255(NR1)>

注 記 ESR0の各レジスタの内容は、:ESE0コマンドの説明の表を参照してください。
 ESR0?を実行するとESR0の内容はクリアされます

固有のイベント・ステータス・イネーブル・レジスタ ESE1 の設定と問い合わせ

構 文 コマンド **:ESE1** <0 ~ 255(NR1)>
 クエリ **:ESE1?**
 応答 <0 ~ 255(NR1)>

説 明 コマンド イベント・ステータス・イネーブルレジスタ 1(ESER1) にイベント・ステータス・レジスタの使用可能パターンを設定します。

クエリ イベント・ステータス・イネーブルレジスタ 1(ESER1) にイベント・ステータス・レジスタの使用可能パターンを問い合わせます。

128 bit 7	64 bit 6	32 bit 5	16 bit 4	8 bit 3	4 bit 2	2 bit 1	1 bit 0
FAIL	PASS	X/θ-Hi	X/θ-IN	X/θ-Lo	R/Z-Hi	R/Z-IN	R/Z-Lo

注 記 電源投入の時はデータを 0 に初期化します。

固有のイベント・ステータス・レジスタ ESR1 の問い合わせ

構 文 クエリ **:ESR1?**
 応答 <0 ~ 255(NR1)>

注 記 ESR1の各レジスタの内容は、:ESE1コマンドの説明の表を参照してください。
 ESR1?を実行するとESR1の内容はクリアされます

(3) 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

(4) 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	:FUNCtion <RV / ZV / R / Z / V>
	クエリ	:FUNCtion?
	応答	<RV / ZV / R / Z / V>

RV...(R,X,V,T)ファンクション(抵抗・リアクタンス・電圧・温度)
 ZV...(Z,θ,V,T)ファンクション(インピーダンス・位相角・電圧・温度)
 R.....(R,X,T)ファンクション(抵抗・リアクタンス・温度)
 Z.....(Z,θ,T)ファンクション(インピーダンス・位相角・温度)
 V.....(V,T)ファンクション(電圧・温度)

例 **:FUNC RV**
 ファンクションを(R,X,V,T)に設定します。
:FUNC?
RV
 ファンクションが(R,X,V,T)に設定されています。

(5) 測定周波数

測定周波数の設定と問い合わせ

構 文	コマンド	:FREQuency <周波数>
	クエリ	:FREQuency?
	応答	<周波数>

説 明 BT4560:<周波数(Hz)> = 0.10 ~ 1050 (NR3)
 BT4560-50:<周波数(Hz)> = 0.01 ~ 1050 (NR3)

例 **:FREQ 1000**
 測定周波数を1000(Hz)に設定します。
:FREQ?
1000
 測定周波数が1000(Hz)に設定されています。

注 記 ・ 内部トリガ、低周波数設定時に測定中に設定変更したい場合、測定時間が長い場合、設定変更が即時に変わりません。
 所望の動作をさせるためには、測定を中断しアイドル状態にする必要があります。
 アイドル状態への遷移は以下の手順で行います。

1. 連続測定無効(:INITiate:CONTinuous OFF)
2. 測定中断要求(:ABORt)
3. 測定中断待ち(*OPC?)

(6) 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	:RANGe <測定レンジ>
	クエリ	:RANGe?
	応答	<測定レンジ(Ω)>
説 明	コマンド	<測定レンジ(Ω)> = 0.0~120.0E-03(NR3) <ul style="list-style-type: none"> • 0.0 ≤ 測定レンジ(Ω) ≤ 3.0E-03 測定レンジは3mΩに設定されます。 • 3.0E-03 < 測定レンジ(Ω) ≤ 10.0E-03 測定レンジを10mΩに設定されます。 • 10.0E-03 < 測定レンジ(Ω) ≤ 120.0E-03 測定レンジを100mΩに設定されます。
	クエリ	現在設定されている測定レンジを以下のフォーマットで応答します。 <測定レンジ(Ω)> = 3.0000E-3/10.0000E-3/100.000E-3
例		:RANG 100.000E-03
		測定レンジを100m Ω に設定します。
		:RANG?
		100.000E-3
		測定レンジが100m Ω に設定されています。

(7) サンプリング速度

サンプリング速度の設定と問い合わせ

構 文	コマンド	:SAMPle:RATE <V / Z>, <測定スピード>
	クエリ	:SAMPle:RATE? <V / Z>
	応答	<測定スピード>
		<V> = V測定のサンプリング <Z> = Z測定のサンプリング <測定スピード> = FAST / MEDium / SLOW
例		:SAMP:RATE V, MED
		V測定のサンプリング速度をMEDIUMに設定します。
		:SAMP:RATE? V
		MEDIUM
		V測定のサンプリング速度がMEDIUMに設定されています。

(8) サンプルディレイ

サンプルディレイモードの設定と問い合わせ

構文	コマンド	:SAMPle:DELaY:MODE <WAVE / VOLTage>
	クエリ	:SAMPle:DELaY:MODE?
	応答	<WAVE / VOLTAGE>

<WAVE> = サンプルディレイをディレイ波数で指定するモード
 <VOLTage> = サンプルディレイを電圧勾配偏差で指定するモード

例 :SAMP:DEL:MODE WAVE
 サンプルディレイをディレイ波数で指定するモードに設定します。

:SAMP:DEL:MODE?
 WAVE

サンプルディレイがディレイ波数で指定するモードに設定されています。

サンプルディレイ波数の設定と問い合わせ

構文	コマンド	:SAMPle:DELaY:WAVE <ディレイ波数>
	クエリ	:SAMPle:DELaY:WAVE?
	応答	<波数>

<波数 (波)> = 0.0 ~ 9.0(NR2)

例 :SAMP:DEL:WAVE 6.0
 ディレイ波数を6.0波分に設定します。

:SAMP:DEL:WAVE?
 6.0

ディレイ波数が6.0波分に設定されています。

サンプルディレイ電圧勾配偏差の設定と問い合わせ

構文	コマンド	:SAMPle:DELaY:VOLTage <ディレイ電圧勾配偏差>
	クエリ	:SAMPle:DELaY:VOLTage?
	応答	<電圧勾配偏差>

<電圧勾配偏差 (mV)> = 0.001 ~ 10.000 (NR2)

例 :SAMP:DEL:VOLT 0.1
 ディレイ電圧勾配偏差を0.1mVに設定します。

:SAMP:DEL:VOLT?
 0.100

ディレイ電圧勾配偏差が0.1mVに設定されています。

(9) 電位勾配補正

電位勾配補正の設定と問い合わせ

構文	コマンド	:ADJust:SLOPe <1 / 0 / ON / OFF>
	クエリ	:ADJust:SLOPe?
	応答	<ON / OFF>

例 :ADJ:SLOP ON
:ADJ:SLOP?
ON

(10) 電圧リミット

電圧リミット機能の設定と問い合わせ

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

例 :LIM ON
:LIM?
ON

電圧リミット機能の設定と問い合わせ

構文	コマンド	:LIMiter:VOLTage <電圧リミット値>
	クエリ	:LIMiter:VOLTage?
	応答	<電圧リミット値>

<電圧リミット値(V)> = 0.01 ~ 5.00 (NR2)

例 :LIM:VOLT 5.00
:LIM:VOLT?
5.00

(11) 測定信号ゼロクロス停止

信号ゼロクロス停止機能の設定と問い合わせ

構文	コマンド	:ZERO:CROSSs <1 / 0 / ON / OFF>
	クエリ	:ZERO:CROSSs?
	応答	<ON / OFF>

例 :ZERO:CROSSs ON
:ZERO:CROSSs?
ON

(12) アベレージ

測定アベレージの設定と問い合わせ

構文	コマンド	:CALCulate:AVERage <回数>
	クエリ	:CALCulate:AVERage?
	応答	<回数>

<回数> = 1 ~ 99 (NR1)

インピーダンス測定値を指定回数だけ平均化して出力します。

例 :CALC:AVER 10
:CALC:AVER?
10

(13) ゼロアジャスト

ゼロアジャストの実行と結果の問い合わせ

構文	クエリ	:ADJust? <SPOT / ALL>
	応答	<0 / 1>

説明 <SPOT>= 現在設定されているレンジにおける周波数、電圧測定に対してゼロアジャストを行います

<ALL> = 現在設定されている周波数全域、電圧測定に対してゼロアジャストを行います。

<0> = ゼロアジャストが成功したことを示します。

<1> = ゼロアジャストが失敗したことを示します。

ゼロアジャストについては、本体取扱説明書をご覧ください。

例 :ADJ? SPOT
0

SPOT設定でゼロアジャストを実行し正常に終了しました。

ゼロアジャストのクリア

構文	コマンド	:ADJust:CLEar
----	------	---------------

説明 ゼロアジャストを解除します。

例 :ADJ:CLE

ゼロアジャスト値の問い合わせ(ALL補正)

構 文 クエリ **:ADJust:DATA:ALL? <V / R / RV>**
 応答 **<Vアジャスト値>, <100Hz Rアジャスト値>, <100Hz Xアジャスト値>, <1kHz Rアジャスト値>, <1kHz Xアジャスト値>**

説 明 クエリ
 <V> = 電圧アジャスト値を問い合わせます。
 <R> = 10Hz, 100Hz, 330Hz, 660Hz, 1kHzでの抵抗/リアクタンスのアジャスト値を問い合わせます。
 <RV> = 電圧アジャスト値と、10Hz, 100Hz, 330Hz, 660Hz, 1kHzでの抵抗/リアクタンスのアジャスト値を問い合わせます。

応答 **:ADJust:DATA:ALL? V** の応答
<Vアジャスト値>
:ADJust:DATA:ALL? R の応答
<10Hz Rアジャスト値>, <10Hz Xアジャスト値>, <100Hz Rアジャスト値>, <100Hz Xアジャスト値>, <330Hz Rアジャスト値>, <330Hz Xアジャスト値>, <660Hz Rアジャスト値>, <660Hz Xアジャスト値>, <1kHz Rアジャスト値>, <1kHz Xアジャスト値>
:ADJust:DATA:ALL? RV の応答
<Vアジャスト値>, <10Hz Rアジャスト値>, <10Hz Xアジャスト値>, <100Hz Rアジャスト値>, <100Hz Xアジャスト値>, <330Hz Rアジャスト値>, <330Hz Xアジャスト値>, <660Hz Rアジャスト値>, <660Hz Xアジャスト値>, <1kHz Rアジャスト値>, <1kHz Xアジャスト値>

例 **:ADJ:DATA:ALL? V**
1.10000E-03
 電圧アジャスト値が1.1mVに設定されています。
:ADJ:DATA:ALL? R
+1.10000E-04,+5.00000E-06,+2.50000E-04,+2.0000E-05,+4.50000E-04,+4.0000E-05,+4.00000E-04,+5.0000E-05,+1.50000E-04,+1.00000E-05
 10Hzにおける抵抗/リアクタンスのアジャスト値がそれぞれ0.11mΩ, 0.005mΩに、100Hzにおける抵抗/リアクタンスのアジャスト値がそれぞれ0.25mΩ, 0.02mΩに、330Hzにおける抵抗/リアクタンスのアジャスト値がそれぞれ0.45mΩ, 0.04mΩに、660Hzにおける抵抗/リアクタンスのアジャスト値がそれぞれ0.40mΩ, 0.05mΩに、1kHzにおける抵抗/リアクタンスのアジャスト値がそれぞれ0.15mΩ, 0.01mΩに設定されています。
:ADJ:DATA:ALL? RV
+1.10000E-03,+1.10000E-04,+5.00000E-06,+2.50000E-04,+2.0000E-05,+4.50000E-04,+4.0000E-05,+4.00000E-04,+5.0000E-05,+1.50000E-04,+1.00000E-05
 電圧アジャスト値が1.1mVに設定されています。
 また、10Hzにおける抵抗/リアクタンスのアジャスト値がそれぞれ0.11mΩ, 0.005mΩに、100Hzにおける抵抗/リアクタンスのアジャスト値がそれぞれ0.25mΩ, 0.02mΩに、330Hzにおける抵抗/リアクタンスのアジャスト値がそれぞれ0.45mΩ, 0.04mΩに、660Hzにおける抵抗/リアクタンスのアジャスト値がそれぞれ0.40mΩ, 0.05mΩに、1kHzにおける抵抗/リアクタンスのアジャスト値がそれぞれ0.15mΩ, 0.01mΩに設定されています。

ゼロアジャスト値の問い合わせ(SPOT補正)

構 文	クエリ	:ADJust:DATA:SPOT? <V / R / RV>
	応答	<Vアジャスト値>, <Rアジャスト値>, <Xアジャスト値>
説 明	データ部	<V> = 電圧アジャスト値を問い合わせます。 <R> = 現在設定されている周波数での抵抗/リアクタンスのアジャスト値を問い合わせます。 <RV> = 電圧アジャスト値と、現在設定されている周波数での抵抗/リアクタンスのアジャスト値を問い合わせます。
	応答	:ADJust:DATA:SPOT? V の応答 <Vアジャスト値> :ADJust:DATA:SPOT? R の応答 <Rアジャスト値>, <Xアジャスト値> :ADJust:DATA:SPOT? RV の応答 <Vアジャスト値>, <Rアジャスト値>, <Xアジャスト値>
例		:ADJ:DATA:SPOT? V +1.10000E-03 電圧アジャスト値が1.1mVに設定されています。 :ADJ:DATA:SPOT? R +2.50000E-04,+2.0000E-04 抵抗/リアクタンスのアジャスト値がそれぞれ0.25mΩ, 0.20mΩに設定されています。 :ADJ:DATA:SPOT? RV +1.10000E-03,+2.50000E-04,+2.0000E-04 電圧アジャスト値が1.1mVに設定されています。 また、抵抗/リアクタンスのアジャスト値がそれぞれ0.25mΩ, 0.20mΩに、 設定されています。

ゼロアジャスト実行状態の問い合わせ

構 文	クエリ	:ADJust:STATe?
	応答	<ON / OFF>
説 明		<ON> = ゼロアジャスト値の測定値への適用を有効に設定します。 <OFF> = ゼロアジャスト値の測定値への適用を無効に設定します。
例		:ADJ:STAT? OFF ゼロアジャスト値の測定値の適用が無効に設定されています。

(14) セルフキャリブレーション

セルフキャリブレーションの実行

構文	コマンド	:CALibration
----	------	--------------

注記 コマンド送信時に本体が測定中のときは、測定終了後にキャリブレーションが実行されます。

セルフキャリブレーションの実行と設定

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

<ON> = セルフキャリブレーション AUTO
電圧測定の際は必ずセルフキャリブレーションを実行する設定とします。

<OFF> = セルフキャリブレーション MANUAL
セルフキャリブレーションを手動で実行する設定とします。
:CALibrationコマンドを送信、またはEXT.IO/端子からCAL信号を入力してセルフキャリブレーションを実行します。

例 :CAL:AUTO OFF
:CAL:AUTO?
OFF

(15) コンパレータ

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	<ON / OFF >

例 :CALC:LIM:STAT ON
:CALC:LIM:STAT?
ON

ブザーの設定と問い合わせ

構文	コマンド	:CALCulate:LIMit:BEEPer <条件>
	クエリ	:CALCulate:LIMit:BEEPer?
	応答	<条件>

<条件> = OFF / HL / IN / ALL

OFF 消音
HL 上下限值外にあるときにビープ音が鳴るように設定
IN 上下限值内にあるときにビープ音が鳴るように設定
ALL 上下限値の内外にかかわらずビープ音が鳴るように設定

例 :CALC:LIM:BEEP IN
:CALC:LIM:BEEP?
IN

電圧成分のコンパレータ絶対値判定の設定と問い合わせ

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

例 :CALC:LIM:ABS ON

コンパレータの電圧判定を絶対値で行うよう設定します。

:CALC:LIM:ABS?

ON

コンパレータの電圧判定が絶対値で行うよう設定されています。

抵抗成分のコンパレータ上下限値の設定と問い合わせ

構文	コマンド	:CALCulate:LIMit:RESistance <上限値>,<下限値>
	クエリ	:CALCulate:LIMit:RESistance?
	応答	<上限値(Ω)>,<下限値(Ω)>

説明 コマンド <上限値(Ω)> = -3.00000E-03 ~ +1.20000E-01(NR3), またはOFF
 <下限値(Ω)> = -3.00000E-03 ~ +1.20000E-01(NR3), またはOFF
 上限値に範囲外の値を指定すると上限値は無効(OFF)になります。
 下限値に範囲外の値を指定すると下限値は無効(OFF)になります。

クエリ <上限値(Ω)> = -3.00000E-03 ~ +1.20000E-01(NR3), またはOFF
 <下限値(Ω)> = -3.00000E-03 ~ +1.20000E-01(NR3), またはOFF

例 :CALC:LIM:RES 0.1,0.05

上下限値が(レンジによらず)それぞれ、100mΩ, 50mΩになります。

:CALC:LIM:RES 5.0,0.05

:CALC:LIM:RES?

OFF,+5.00000E-02

上下限値がそれぞれ、OFF, 50mΩになります。

:CALC:LIM:RES 0.1,OFF

:CALC:LIM:RES?

+1.00000E-01,OFF

上下限値がそれぞれ、100mΩ, OFFになります。

注記 上限値が下限値より小さく設定された場合は上限値で設定した値を下限値として設定します。

リアクタンス成分のコンパレータ上下限値の設定と問い合わせ

構文 コマンド **:CALCulate:LIMit:REACtance <上限値>,<下限値>**
クエリ **:CALCulate:LIMit:REACtance?**
応答 **<上限値(Ω)>,<下限値(Ω)>**

説明 コマンド <上限値(Ω)> = -1.20000E-01 ~ +1.20000E-01(NR3), またはOFF
<下限値(Ω)> = -1.20000E-01 ~ +1.20000E-01(NR3), またはOFF
上限値に範囲外の値を指定すると上限値は無効(OFF)になります。
下限値に範囲外の値を指定すると下限値は無効(OFF)になります。

クエリ <上限値(Ω)> = -1.20000E-01 ~ +1.20000E-01 (NR3)またはOFF
<下限値(Ω)> = -1.20000E-01 ~ +1.20000E-01 (NR3)またはOFF

例 **:CALC:LIM:REAC 0.1,0.05**
上下限値が(レンジによらず)それぞれ、100mΩ, 50mΩになります。
:CALC:LIM:REAC 5.0,0.05
:CALC:LIM:REAC?
OFF,+5.00000E-02
上下限値がそれぞれ、OFF, 50mΩになります。
:CALC:LIM:REAC 0.1,OFF
:CALC:LIM:REAC?
+1.00000E-01,OFF
上下限値がそれぞれ、100mΩ, OFFになります。

注記 上限値が下限値より小さく設定された場合は上限値で設定した値を下限値として設定します。

インピーダンス成分のコンパレータ上下限値の設定と問い合わせ

構文 コマンド **:CALCulate:LIMit:IMPedance <上限値>,<下限値>**
クエリ **:CALCulate:LIMit:IMPedance?**
応答 **<上限値(Ω)>,<下限値(Ω)>**

説明 コマンド <上限値(Ω)> = 0 ~ +1.20000E-01(NR3), またはOFF
<下限値(Ω)> = 0 ~ +1.20000E-01(NR3), またはOFF
上限値に範囲外の値を指定すると上限値は無効(OFF)になります。
下限値に範囲外の値を指定すると下限値は無効(OFF)になります。

クエリ <上限値(Ω)> = 0 ~ +1.20000E-01 (NR3)またはOFF
<下限値(Ω)> = 0 ~ +1.20000E-01 (NR3)またはOFF

例 **:CALC:LIM:IMP 0.1,0.05**
上下限値が(レンジによらず)それぞれ、100mΩ, 50mΩになります。
:CALC:LIM:IMP 5.0,0.05
:CALC:LIM:IMP?
OFF,+5.00000E-02
上下限値がそれぞれ、OFF, 50mΩになります。
:CALC:LIM:IMP 0.1,OFF
:CALC:LIM:IMP?
+1.00000E-01,OFF
上下限値がそれぞれ、100mΩ, OFFになります。

注記 上限値が下限値より小さく設定された場合は上限値で設定した値を下限値として設定します。

位相成分のコンパレータ上下限値の設定と問い合わせ

構文	コマンド	:CALCulate:LIMit:PHASe <上限値>,<下限値>
	クエリ	:CALCulate:LIMit:PHASe?
	応答	< 上限値(°)>,< 下限値(°)>
コマンド <上限値(°)> = -1.80000E+02 ~ +1.80000E+02(NR3), またはOFF <下限値(°)> = -1.80000E+02 ~ +1.80000E+02(NR3), またはOFF 上限値に範囲外の値を指定すると上限値は無効(OFF)になります。 下限値に範囲外の値を指定すると下限値は無効(OFF)になります。		
クエリ <上限値(°)> = -1.80000E+02 ~ +1.80000E+02(NR3)またはOFF <下限値(°)> = -1.80000E+02 ~ +1.80000E+02(NR3)またはOFF		

例 :CALC:LIM:PHAS 90.0,-90.0
 上下限値がそれぞれ、90.0°, -90.0° になります。
 :CALC:LIM:PHAS 300,-90.0
 :CALC:LIM:PHAS?
 OFF,-9.00000E+01
 上下限値がそれぞれ、OFF, -90.0° になります。
 :CALC:LIM:PHAS 90.0,OFF
 :CALC:LIM:PHAS?
 +9.00000E+01,OFF
 上下限値がそれぞれ、90.0°, OFFになります。

注記 上限値が下限値より小さく設定された場合は上限値で設定した値を下限値として設定します。

電圧成分のコンパレータ上下限値の設定と問い合わせ

構文	コマンド	:CALCulate:LIMit:VOLTage <上限値>,<下限値>
	クエリ	:CALCulate:LIMit:VOLTage?
	応答	<上限値(V)>,<下限値(V)>
コマンド <上限値(V)> = -5.10000E+00 ~ +5.10000E+00(NR3), またはOFF <下限値(V)> = -5.10000E+00 ~ +5.10000E+00(NR3), またはOFF 上限値に範囲外の値を指定すると上限値は無効(OFF)になります。 下限値に範囲外の値を指定すると下限値は無効(OFF)になります。		
クエリ <上限値(V)> = -5.10000E+00 ~ +5.10000E+00(NR3)またはOFF <下限値(V)> = -5.10000E+00 ~ +5.10000E+00(NR3)またはOFF		

例 :CALC:LIM:VOLT 5.0,4.0
 上下限値がそれぞれ、5.0V, 4.0V になります。
 :CALC:LIM:VOLT 30,4.0
 :CALC:LIM:VOLT?
 OFF,+4.00000E+00
 上下限値がそれぞれ、OFF, 4.0Vになります。
 :CALC:LIM:VOLT 5.0,OFF
 :CALC:LIM:VOLT?
 +5.00000E+00,OFF
 上下限値がそれぞれ、5.0V, OFFになります。

注記 上限値が下限値より小さく設定された場合は上限値で設定した値を下限値として設定します。

(16) 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 **:SAVE** <保存 No.>
:LOAD <保存 No.>
 <保存 No.> = 1~126 (NR1)

例 **:SAVE 10**
:SAVE:CLE 10
:LOAD 5

注記 測定条件がすでに保存されている保存 No.の場合、**:SAVE** を実行すると測定条件が上書きされます。
 測定条件が保存されていない保存 No.を指定して**:LOAD** を実行すると実行エラーになります。

測定条件の削除

構文 コマンド **:SAVE:CLEar** <保存 No.>
 <保存 No.> = 1~126 (NR1)

例 **:SAVE:CLE 10**

注記 測定条件が保存されていない保存 No.を指定して**:SAVE:CLEar** を実行すると実行エラーになります。

(17) システムリセット

機器の初期化(システムリセット)

構文 コマンド **:SYSTem:RESet**

説明 通信設定を除き、機器を工場出荷時の設定にします。(システムリセット)
 ゼロアジャスト値、パネルセーブデータも初期化されます。

注記 ゼロアジャスト値、パネルセーブデータを残したい場合は***RST**コマンド(p.17)を使ってください。

(18) 測定値出力

測定完了後、測定値を出力する設定と問い合わせ

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

説明 コマンド <ON> = 測定完了後、選択されているインタフェースに測定値を出力します。
 <OFF> = 測定値の読み出し要求(:FETCh?または:READ?クエリ)があったときのみ、測定値を出力します。
 クエリ 測定終了後、測定値を出力する設定を応答します。

例 **:SYST:DATA ON**
:SYST:DATA?
 ON

(19) キー操作音

キー操作音の設定と問い合わせ

構文	コマンド	:SYSTem:BEEPer <1 / 0 / ON / OFF>
	クエリ	:SYSTem:BEEPer?
	応答	<ON / OFF>

例 :SYST:BEEP ON
:SYST:BEEP?
ON

(20) 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	<ON / OFF>

例 :SYST:KLOC ON
:SYST:KLOCK?
ON

(21) 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

(22) ヘッダ

ヘッダ有無の設定と問い合わせ

構文	コマンド	:SYSTem:HEADer <1 / 0 / ON / OFF>
	クエリ	:SYSTem:HEADer?
	応答	<ON / OFF>

説明 応答メッセージのヘッダの有無を設定します。
応答メッセージへのヘッダの設定を、ONまたはOFFで返します。

例 :SYST:HEAD ON
:SYST:HEAD?
:SYSTEM:HEADER ON

:SYST:HEAD OFF
:SYST:HEAD?
OFF

注記 電源投入時はヘッダOFFに初期化されます。

(23) 製造番号

製造番号の問い合わせ

構文	クエリ	:SYSTem:SERIal?
	応答	<製造番号>

例 :SYST:SER?
123456789

(24) LCD の設定

コントラストの設定と問い合わせ

構文	コマンド	:SYSTem:DISPlay:CONTRast <コントラスト>
	クエリ	:SYSTem:DISPlay:CONTRast?
	応答	<コントラスト>
		<コントラスト> = < 0 ~ 100 >

例 :SYST:DISP:CONT 50
ディスプレイのコントラストを50%に設定します。

:SYST:DISP:CONT?
50
ディスプレイのコントラストが50%に設定されています。

バックライトの設定と問い合わせ

構文	コマンド	:SYSTem:DISPlay:BACKlight <輝度>
	クエリ	:SYSTem:DISPlay:BACKlight?
	応答	<輝度>
		<輝度> = < 10 ~ 100 >

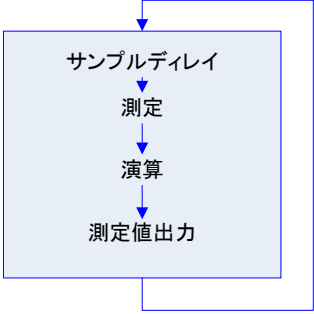
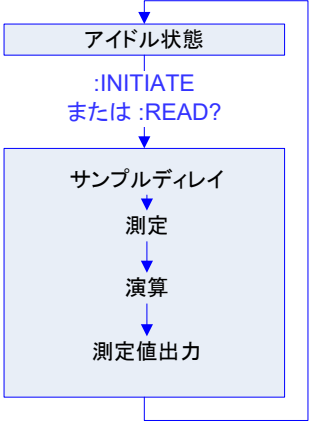
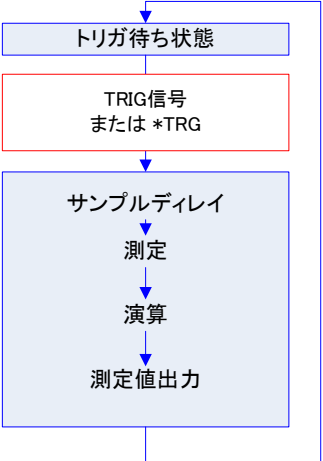

例 :SYST:DISP:BACK 50
ディスプレイのバックライトの輝度を50%に設定します。

:SYST:DISP:BACK?
50
ディスプレイのバックライトの輝度が50%に設定されています。

(25) 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.39) とトリガソースの設定 (:TRIGGER:SOURCE) (参照: p.39) により、次のように動作します。参照: 「4 データ取得方法」(p.45)

測定フロー		連続測定 コマンド設定のみ	
		:INITIATE:CONTINUOUS ON	:INITIATE:CONTINUOUS OFF
トリガソース	:TRIGGER:SOURCE IMM	<p>フリーラン状態。自動的に測定を続けます。</p> 	<p>:INITIATE(または:READ?) で測定します。</p> 
	:TRIGGER:SOURCE EXT	<p>TRIG信号または*TRGコマンドでトリガします。測定終了後はトリガ待ち状態になります。</p> 	<p>:INITIATE(または:READ?) でトリガ待ちになります。TRIG信号でトリガします。測定終了後はアイドル状態になります。</p> 

:INITIATE:CONTINUOUS OFFはリモートコマンドでのみ設定可能です。

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、次回の電源投入時は: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 (ローカル状態にする)」(p.35, p. 11)

測定値の取得方法について:「データ取得方法」(p.45)

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?**
応答 **<ON / OFF>**
<ON> = 連続測定有効
<OFF> = 連続測定無効

- 説明**
- 連続測定有効:
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:
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、トリガを受け付けられない状態です。:INITiate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**
連続測定を無効に設定します。
:INIT:CONT?
OFF
連続測定が無効に設定されています。

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>**
クエリ **:TRIGger:SOURce?**
応答 **<IMMEDIATE / EXTERNAL>**
<IMMEDIATE> = 内部トリガ
<EXTERNAL> = 外部トリガ

例 **:TRIG:SOUR IMM**
トリガソースを内部トリガに設定します。
:TRIG:SOUR?
IMMEDIATE
トリガソースが内部トリガに設定されています。

- 注記**
- 内部トリガ(トリガソース<IMMEDIATE>)を選択すると、電圧リミットがオンに設定されます(:LIMITer:VOLTage ON)。
 - 内部トリガ、低周波数設定時に測定中に設定変更したい場合、測定時間が長い場合、設定変更が即時に変わりません。
所望の動作をさせるためには、測定を中断しアイドル状態にする必要があります。
アイドル状態への遷移は以下の手順で行います。
1. 連続測定無効(:INITiate:CONTinuous OFF)
 2. 測定中断要求(:ABORT)
 3. 測定中断待ち(*OPC?)

トリガ待ち状態への遷移

構 文 コマンド **:INITiate**

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

例 **:TRIG:SOUR IMM**..... 内部トリガ
:INIT..... トリガ待ちに設定
内部トリガなので即座にトリガして1回測定

:TRIG:SOUR EXT..... 外部トリガ
:INIT..... 外部からのトリガ待ちに設定

注 記

- 本コマンド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。

(26) 測定値の読み出し

測定値のフォーマット

・インピーダンス(絶対値表示:単位Ω)

測定値	測定異常時
+ □ . □□□□ E-0□	参照: 測定値のフォーマット(測定異常時)(p.41)

・位相角(絶対値表示:単位°)

測定値	測定異常時
± □ . □□□□ E±0□	参照: 測定値のフォーマット(測定異常時)(p.41)

・抵抗、リアクタンス(絶対値表示:単位Ω)

測定値	測定異常時
± □ . □□□□ E-0□	参照: 測定値のフォーマット(測定異常時)(p.41)

・電圧(絶対値表示:単位V)

測定値	測定異常時
± □ . □□□□ E±0□	参照: 測定値のフォーマット(測定異常時)(p.41)

・温度(絶対値表示:単位℃)

測定値	測定異常時
± □ . □□□□ E±0□	参照: 測定値のフォーマット(測定異常時)(p.41)

測定値のフォーマット(測定異常時)

・インピーダンス、位相角、抵抗、リアクタンス、電圧

測定異常	エラー表示	測定値
Zレンジオーバー	OverRange	+ 1. 00000 E+08
電圧ドリフトによるインピーダンス測定エラー	DRIFT VOLTAGE	+ 2. 00000 E+08
SOURCE-L – SENSE-L間コンタクトエラー	CONTACT ERROR L	+ 3. 00000 E+08
SOURCE-H – SENSE-H間コンタクトエラー	CONTACT ERROR H	+ 4. 00000 E+08
リターンケーブル未接続エラー	RETURN CABLE ERROR	+ 5. 00000 E+08
電圧リミットエラー	OVER V LIMIT	+ 6. 00000 E+08
過電圧入力エラー	OVER VOLTAGE	+ 7. 00000 E+08
SOURCE-H – SOURCE-L間定電流エラー	-----	+ 8. 00000 E+08
A/D通信エラー	A/D ERROR	+ 9. 00000 E+08
内部電池交換エラー	VREF B ERROR	+ 1. 00000 E+09
電源投入後未測定		+ 2. 00000 E+09

・温度

測定異常	エラー表示	測定値
Tレンジオーバー	+Over °C	+ 1. 00000 E+08
Tレンジアンダー	-Under °C	+ 2. 00000 E+08
温度センサ未接続	-- °C	+ 3. 00000 E+08
電源投入後未測定		+ 4. 00000 E+08

:FETCh? /FETCh:TEMPerature? と :READ? 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.45)、トリガ(p.38)

測定の中断

構文 クエリ **:ABORt**

説明 測定の中断(強制終了)をします。
:READ? は中断できません。

例 :TRIG:SOUR EXT
:INIT:CONT ON
*TRG
:ABOR
測定を中断します。

:TRIG:SOUR EXT
:INIT:CONT ON
*TRG;*WAI
:ABOR

この場合は測定待ち状態のため、実行中の測定を中断できません。

注記 *WAIコマンドの後に送信すると前のコマンド動作が終了するまで待機するため、中断はできません。
*OPC,*OPC?コマンドの後に送信すると前のコマンド動作が終了するまで待機するため、中断はできません。

測定値の読出しクエリの応答データの設定と問い合わせ

構文 コマンド **:MEASure:VALid <MR0>**
クエリ **:MEASure:VALid?**
応答 **<MR0>**
<MR0> = 1~7(NR1)
MR: メジャメント・レジスタ(下表参照)

説明 コマンド **:FETCh? / :READ?**でクエリで応答する測定パラメータをビット(<MR0>)で指定します。

クエリ **:FETCh? / :READ?**でクエリで応答する測定パラメータをビット(<MR0>)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
未使用	未使用	未使用	未使用	未使用	総合判定結果	判定結果	測定値

メジャメント・レジスタ0 (MR0)

例 :MEAS:VAL 7

測定値, 測定結果, 総合判定結果を応答するように設定します。

:MEAS:VAL?
7

測定値, 測定結果, 総合判定結果を応答するように設定されています。

最後の測定値の読み出し

構文 クエリ
応答

:FETCh?

<総合判定結果>,<測定値>,<判定結果>
,<測定値>,<判定結果>,...

参照:「測定値のフォーマット」(p.41)

説明 最後(直近)のインピーダンスと電圧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.45)、トリガ(p.38)

(R,X,V,T)測定の場合の応答は以下のようになります。

<総合判定結果> = PASS/FAIL/OFF

<測定値> = R測定値

<判定結果> = R測定コンパレータ判定結果<HI/IN/LO/OFF>

<測定値> = X測定値

<判定結果> = X測定コンパレータ判定結果< HI/IN/LO/OFF >

<測定値> = V測定値

<判定結果> = V測定コンパレータ判定結果< HI/IN/LO/OFF >

例 **:MEAS:VAL 1**..... 測定値を応答するように設定
:FETC?

+1.02500E-01,+1.02800E-01,+3.00000E+00

:MEAS:VAL 3..... 測定値、判定結果を応答するように設定
:FETC?

+1.02500E-01,IN,+1.02800E-01,IN,+3.00000E+00,IN

:MEAS:VAL 7..... 総合判定結果、測定値、判定結果を応答するように設定
:FETC?

PASS,+1.02500E-01,IN,+1.02800E-01,IN,+3.00000E+00,IN

注記 • *TRG;*WAI::FETC?(*WAIは*OPC/*OPC?でもよい)、:READ?中に測定を中断にするには、LOCALキーを押してください。LOCALキー受付後に測定値の応答を返すため、[RMT]表示が消えない場合があります。その場合表示が消えるまでLOCALキーを押してください。

温度測定値の読み出し

構文 クエリ
応答

:FETCh:TEMPerature?

< 測定値 > 参照:「測定値のフォーマット」(p.41)

説明 最後(直近)の温度測定値を読み出します。

例 **:FETC:TEMP?**
+2.51000E+01

測定（トリガ待ちと測定値の読み出し）

構文 クエリ **:READ?**
 応答 <総合判定結果>,<測定値>,<判定結果>
 ,<測定値>,<判定結果>,,
 参照:「測定値のフォーマット」(p.41)

説明 アイドル状態から 1 回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。

トリガソース	動作
IMMediate	測定値を読み出します。
EXTernal	EXT.I/OのTRIG 信号によってトリガし、続いて測定値を読み出します。

(R,X,V,T)測定の場合の応答は以下のようになります。

<総合判定結果> = PASS/FAIL/OFF

<測定値> = R測定値

<判定結果> = R測定コンパレータ判定結果<HI/IN/LO/OFF>

<測定値> = X測定値

<判定結果> = X測定コンパレータ判定結果<HI/IN/LO/OFF>

<測定値> = V測定値

<判定結果> = V測定コンパレータ判定結果<HI/IN/LO/OFF>

例 **:TRIG:SOUR IMM**..... 内部トリガに設定
:MEAS:VAL 1..... 測定値を応答するように設定
:READ? 内部トリガなので即座にトリガして1回測定
 +1.02500E-01,+1.02800E-01,+3.00000E+00

:TRIG:SOUR EXT..... 外部トリガに設定
:MEAS:VAL 1..... 測定値を応答するように設定
:READ? 外部トリガなのでEXT.I/OのTRIG信号を受け
 +1.02500E-01,+1.02800E-01,+3.00000E+00 付けると1回測定

注記

- ・本コマンド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、1回測定して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の状態になり、EXT.I/OのTRIG信号を受け付けると1回測定してアイドル状態になります。
- ・**:READ?**実行後、通信コマンドで測定を中断できません。
キーを押して測定を中断することで、中断時点における測定値を応答します。
- ・測定が終了するまで、次のコマンドは実行されません。
- ・*TRG;*WAI;:FETC?(*WAIは*OPC/*OPC?でもよい)、**:READ?**中に測定を中断するには、LOCALキーを押してください。LOCALキー受付後に測定値の応答を返すため、[RMT]表示が消えない場合があります。その場合表示が消えるまでLOCALキーを押してください。

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

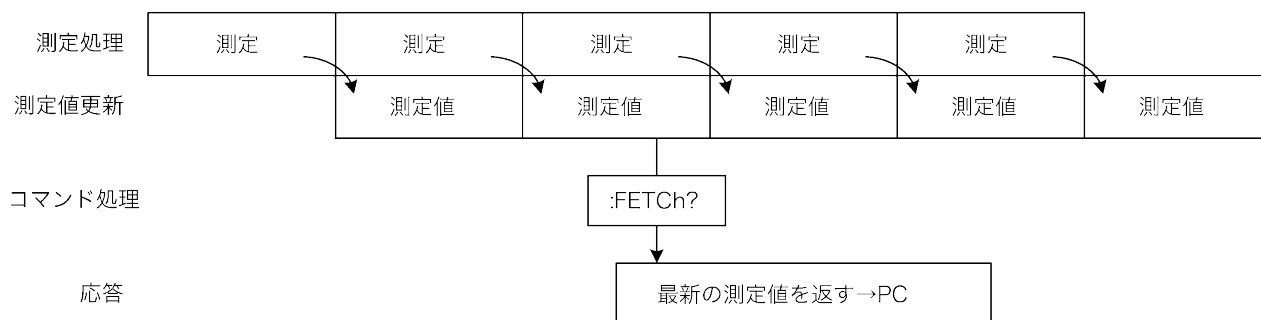
コントローラ(PC、PLC)からトリガしてデータ取り込み

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? 測定が終了したら測定値を転送

TRIG 信号を入力してデータ取り込み

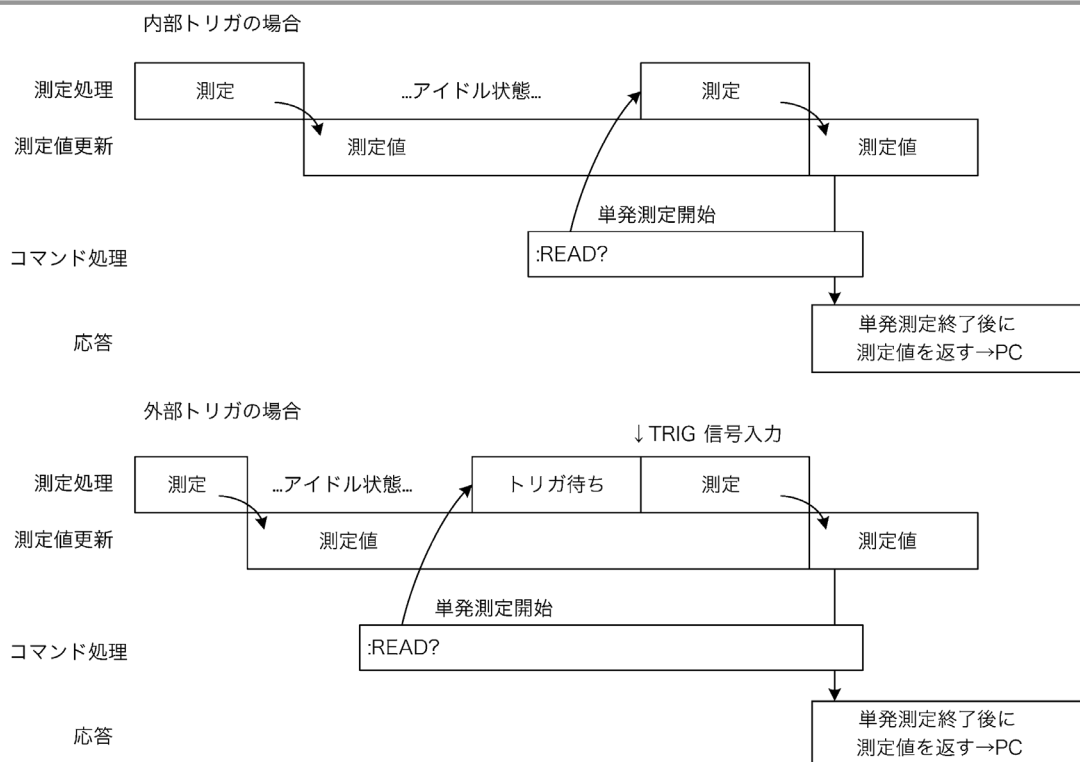
初期設定	:INITiate:CONTinuous OFF (連続測定無効) :TRIGger:SOURce EXT (外部トリガ)
取り込み	:READ? TRIG 信号が入力されたら、測定して値を転送

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)をする方法です。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 Basic2013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 = ":FUNC RV" & vbCrLf ' (R,X,V,T)測定のファンクションに設定
    MSComm1.Output = ":MEAS:VAL 1" & vbCrLf ' 測定値のみ出力
    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 = ":FUNC RV" & vbCrLf ' (R,X,V,T)測定のファンクションに設定
    MSComm1.Output = ":MEAS:VAL 1" & vbCrLf ' 測定値のみ出力
    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

外部トリガ (TRIG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 MeasureTrigSubRS()
    Dim recvstr As String                ' 受信文字列
    Dim i As Integer

    MComm1.CommPort = 1                 ' COM1 (通信ポートを確認してください)
    MComm1.Settings = "9600,n,8,1"      ' 通信ポートの設定 (USBのときは不要)
    MComm1.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

    MComm1.Output = ":FUNC RV" & vbCrLf ' [R,X,V,T]測定のファンクションに設定
    MComm1.Output = ":MEAS:VAL 1" & vbCrLf ' 測定値のみ出力
    MComm1.Output = ":TRIG:SOUR EXT" & vbCrLf ' 外部トリガを選択
    MComm1.Output = ":INIT:CONT OFF" & vbCrLf ' 連続測定 OFF
    For i = 1 To 10
        MComm1.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 + MComm1.Input
            DoEvents
            ' 外部トリガの入力があつた場合に測定と取り込みが実行されます。
        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
    MComm1.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
' ファンクション:[R,X,V,T]
' 測定周波数:1000Hz
' レンジ:100mΩ
' Zサンプリング:FAST, Vサンプリング:FAST
' トリガ:内部トリガ
' コンパレータ:ON, High および Low でブザーをならす
' 抵抗:上限値 100mΩ, 下限値 50mΩ
' リアクタンス:上限値 100mΩ 下限値 50mΩ
' インピーダンス:上限値 100mΩ 下限値 50mΩ
' 電圧:上限値 5V 下限値 2.5V

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 = ":FUNC RV" & vbCrLf               ' (R,X,V,T)ファンクションに設定
    MSComm1.Output = ":FREQ 1000"                       ' 測定周波数を1000Hzに設定
    MSComm1.Output = ":RANG 1E-1" & vbCrLf              ' 測定レンジを 100mΩ に設定
    MSComm1.Output = ":SAMP:RATE Z FAST" & vbCrLf       ' サンプリングを FAST に設定
    MSComm1.Output = ":SAMP:RATE V FAST" & vbCrLf
    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
    MSComm1.Output = ":CALC:LIM:STAT ON" & vbCrLf       ' 以下、コンパレータ設定
    MSComm1.Output = ":CALC:LIM:BEEP HL" & vbCrLf
    MSComm1.Output = ":CALC:LIM:RES 1E-1,0.5E-1" & vbCrLf
    MSComm1.Output = ":CALC:LIM:REAC 1E-1,0.5E-1" & vbCrLf
    MSComm1.Output = ":CALC:LIM:IMP 1E-1,0.5E-1" & vbCrLf
    MSComm1.Output = ":CALC:LIM:VOLT 5E+0,2.5E+0" & vbCrLf

    MSComm1.PortOpen = False
End Sub
```

Visual Basic 2013 で作成する

Windowsの開発言語Visual Basic2013 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 Basic2013をVB2013と記載します。

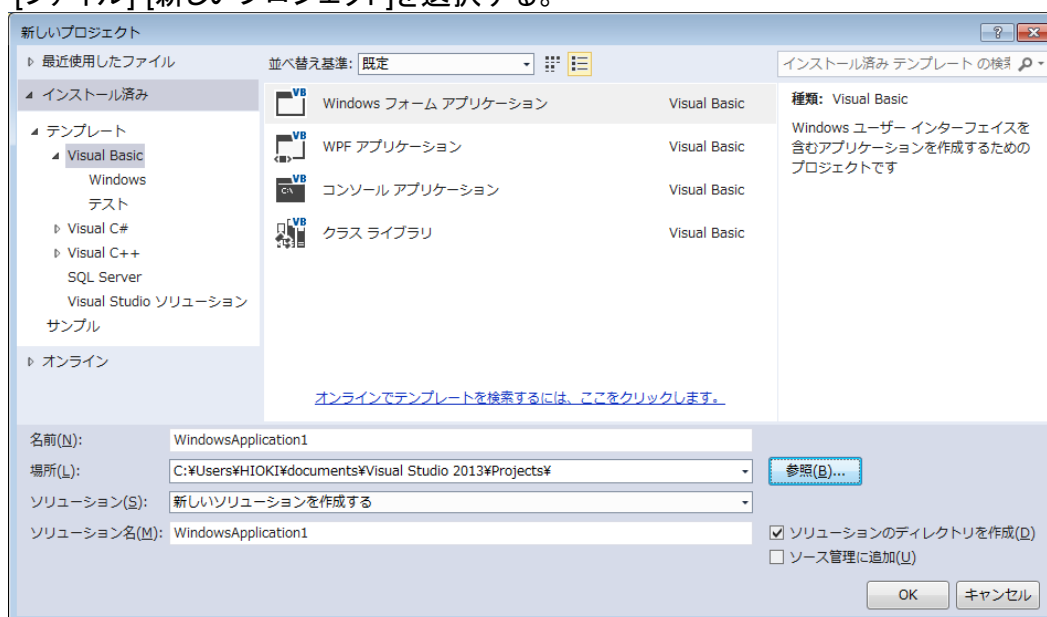
注記 コンピュータやVB2013の環境により、説明が若干異なる場合があります。VB2013の詳しい使用方法についてはVB2013の取扱説明書またはHELPをご覧ください。

1. 新規プロジェクトの作成

1. VB2013 を起動する。



2. [ファイル]-[新しいプロジェクト]を選択する。

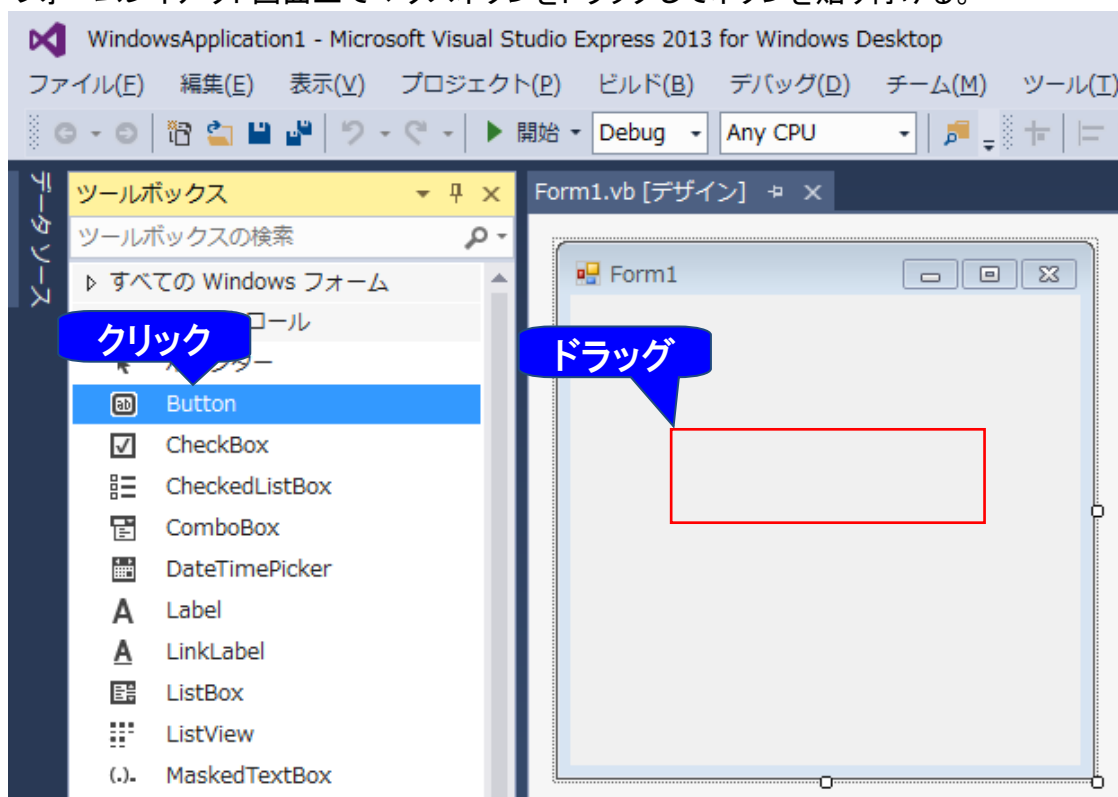


3. テンプレートから[Windows フォーム アプリケーション]を選択する。

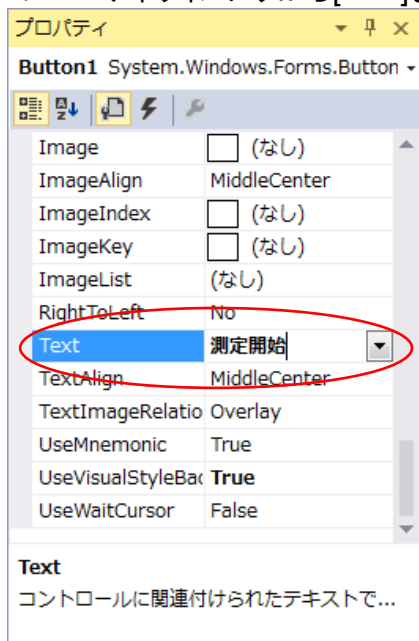
4. [OK]ボタンをクリックする。

2. ボタンの配置

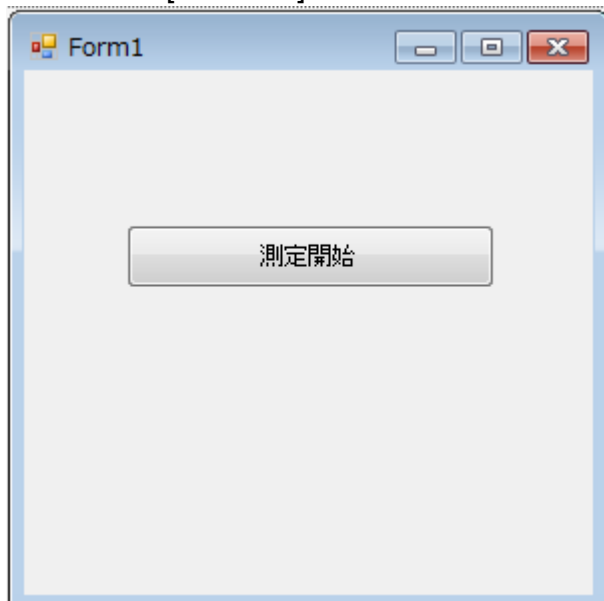
1. [ツールボックス]の[コモンコントロール]から[Button]をクリックする。
2. フォームレイアウト画面上でマウスボタンをドラッグしてボタンを貼り付ける。



3. プロパティウィンドウから[Text]を「測定開始」に変更する。

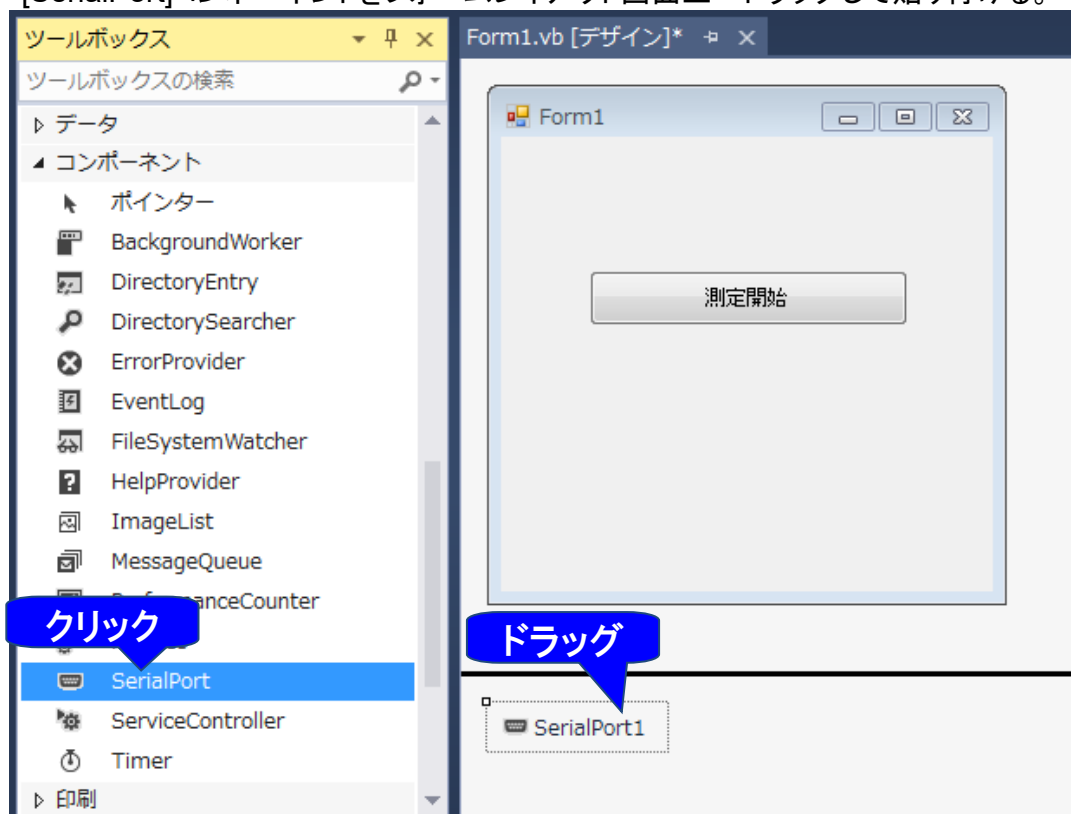


4. フォーム上に[測定開始]ボタンが配置される。



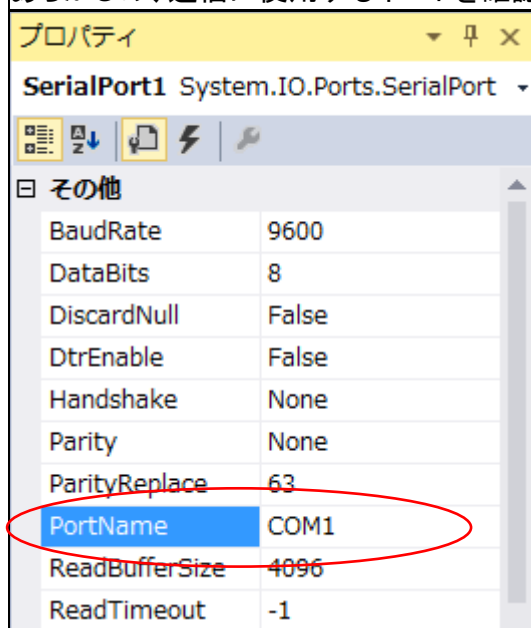
3. シリアル通信コンポーネントの配置

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。



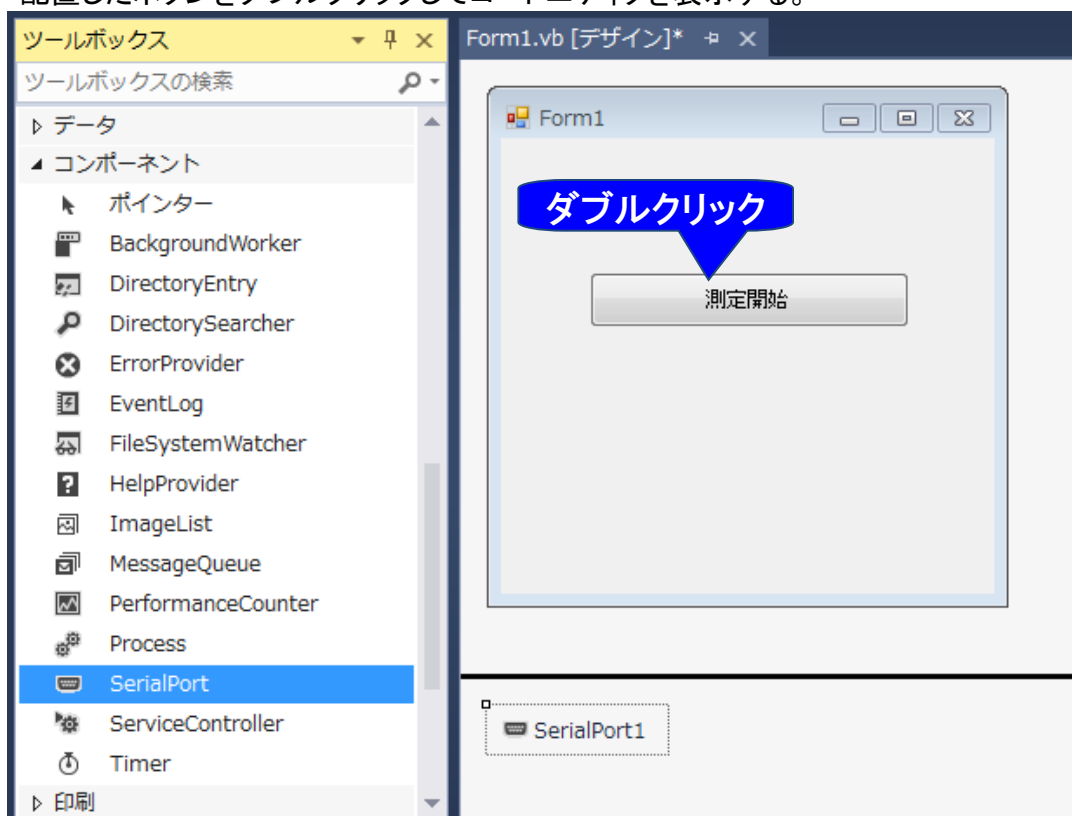
3. プロパティウィンドウから[PortName] を通信に使用するポート名に変更する。

あらかじめ、通信に使用するポートを確認してください

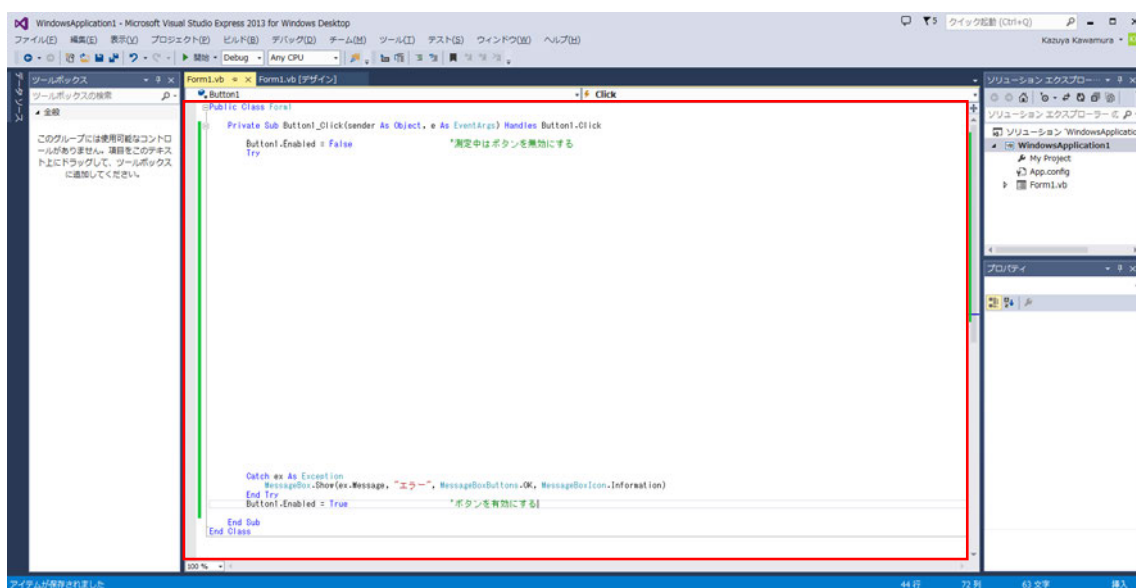


4. コードの記述

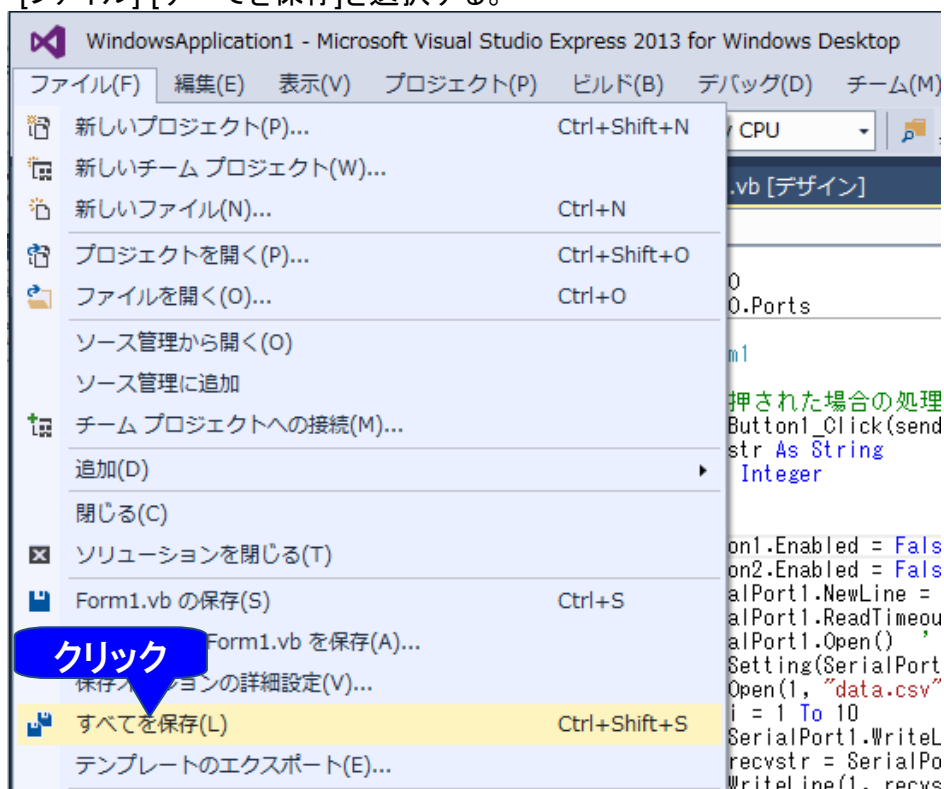
1. 配置したボタンをダブルクリックしてコードエディタを表示する。



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。



以下にVB2013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(sender As Object, e As EventArgs) Handles Button1.Click
        Dim recvstr 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)
                recvstr = SerialPort1.ReadLine() ' 測定結果の読み込み
                PrintLine(1, recvstr)           ' ファイルへ書き出し
            Next i
        Catch ex As Exception
            MessageBox.Show(ex.Message, " エラー ", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
        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

    End Sub

    ' 測定条件の設定を行う
    Private Sub SendSetting(ByVal sp As SerialPort)
        Try
            sp.WriteLine(":FUNC RV")           ' [R,X,V,T]ファンクションに設定
            sp.WriteLine(":FREQ 1000")         ' 測定周波数を1000Hzに設定
            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(sender As Object, e As 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「測定結果をコンピュータに返す」コマンドを送信します。

HIOKI

www.hioki.co.jp/

本社 〒386-1192 長野県上田市小泉 81

製品のお問い合わせ

 **0120-72-0560**

TEL 0268-28-0560 FAX 0268-28-0569

9:00 ~ 12:00, 13:00 ~ 17:00
土・日・祝日を除く
info@hioki.co.jp



国内拠点

修理・校正のお問い合わせ

ご依頼はお買上店（代理店）または最寄りの営業拠点まで
お問い合わせはサービス窓口まで

TEL 0268-28-1688 cs-info@hioki.co.jp

2103 JA

編集・発行 日置電機株式会社

Printed in Japan

- ・ CE 適合宣言は弊社ウェブサイトからダウンロードできます。
- ・ 本書の記載内容を予告なく変更することがあります。
- ・ 本書には著作権により保護される内容が含まれます。
- ・ 本書の内容を無断で転記・複製・改変することを禁止します。
- ・ 本書に記載されている会社名・商品名などは、各社の商標または登録商標です。