

**TA720**

**タイムインターバルアナライザ  
通信インタフェース**

# USER'S MANUAL

---

ユーザーズマニュアル

## はじめに

このたびは、タイムインターバルアナライザTA720をお買い上げいただきましてありがとうございました。

この通信インタフェースユーザズマニュアルは、通信インタフェースの機能やコマンド通信について説明したものです。ご使用前にこのマニュアルをよくお読みいただき、正しくお使いください。

お読みになったあとは、ご使用時にすぐにご覧になれるところに大切に保存してください。ご使用中に操作がわからなくなったときなどにきつとお役に立ちます。

なお、TA720のマニュアルとして、このマニュアルのほかに次のマニュアルがあります。あわせてお読みください。

マニュアル名	マニュアルNo.	内容
TA720ユーザズマニュアル	IM 704510-01	コマンド通信機能を除く全機能とその操作方法について説明しています。

## ご注意

- 本書の内容は、性能・機能の向上などにより、将来予告なしに変更することがあります。
- 本書の内容に関しては万全を期していますが、万一ご不審の点や誤りなどお気づきのことがありましたら、お手数ですが、裏表紙に記載の当社支社・支店・営業所までご連絡ください。
- 本書の内容の全部または一部を無断で転載、複製することは禁止されています。
- 本製品のTCP/IPソフトウェア、およびTCP/IPソフトウェアに関するドキュメントは、カリフォルニア大学からライセンスされたBSD Networking Software, Release 1をもとに当社で開発/作成したものです。

## 商 標

- Microsoft, MS-DOS, およびWindowsは、米国Microsoft Corporationの、米国およびその他の国における登録商標または商標です。
- Adobe, AcrobatおよびPostScriptは、アドビシステムズ社の商標または登録商標です。
- 本文中の各社の登録商標または商標には、TM, ®マークは表示していません。
- その他、本文中に使われている会社名、商品名は、各社の登録商標または商標です。

## 履 歴

2002年8月 初版発行

2002年11月 2版発行

# このマニュアルの利用方法

## このマニュアルの構成

このユーザズマニュアルは、以下に示す第1章～第6章および付録・索引で構成されています。

### 第1章 GP-IB通信インタフェースについて

GP-IB通信インタフェースを使用するときの設定方法などについて説明しています。

### 第2章 イーサネット通信インタフェース(オプション)について

イーサネット通信インタフェースを使用するときの設定方法などについて説明しています。

### 第3章 プログラム書式とプログラム作成時の注意事項

プログラムの書き方と、プログラム作成時の注意について説明しています。

### 第4章 コマンド

使用できる全コマンドについて1つずつ説明しています。

### 第5章 ステータスレポート

ステータスバイトや各種レジスタ・キューなどについて説明しています。

### 第6章 サンプルプログラム

Visual Basicで組んだプログラム例を紹介しています。

### 付 録

ASCIIキャラクタコード表などの参考資料を紹介しています。

### 索 引

五十音順、アルファベット順の2種類の索引があります。

## このマニュアルで使用している記号

### ● 注記やキーを表す記号

種類	記号	意 味
単位	k	1000 例：100kHz
	K	1024 例：128KB(メモリ容量)
注記	<b>Note</b>	知っておいたほうがよい情報が記載されています。

### ● 構文の記号

主に第3章の構文で使用している記号を下表に示します。なお、これはBNF(Backus-Naur Form)記号と呼ばれるものです。なお、データ(プログラムヘッダの後ろにスペースを空けて記述する条件や数値)に使用する記号については、3-5～3-6ページを参照してください。

記号	意 味	例	入力例
<>	定義された値	WINDOW<x> <x>=1～16	WINDOW2
{ }	{ }内から1つを選択	MODE {AUTO MANual}	MODE AUTO
	排他的論理和	MODE {AUTO MANual}	MODE AUTO
[ ]	省略可能		:MEASure[ :MODE]
...	繰り返し可能		

# 目次

はじめに .....	i
このマニュアルの利用方法 .....	ii
<b>第1章 GP-IB通信インタフェースについて</b>	
1.1 GP-IB通信に関連する各部の名称と機能 .....	1-1
1.2 GP-IBケーブルの接続方法 .....	1-2
1.3 GP-IB通信の機能と仕様 .....	1-3
1.4 本体の設定 .....	1-4
1.5 インタフェースメッセージに対する応答 .....	1-5
<b>第2章 イーサネット通信インタフェース(オプション)について</b>	
2.1 イーサネット通信に関連する各部の名称と機能 .....	2-1
2.2 イーサネット通信の機能と仕様 .....	2-2
2.3 イーサネット通信インタフェースによる接続 .....	2-3
2.4 本体の設定 .....	2-4
<b>第3章 プログラム書式とプログラム作成時の注意事項</b>	
3.1 メッセージ .....	3-1
3.2 命令 .....	3-3
3.3 応答 .....	3-4
3.4 データ .....	3-5
3.5 コントローラとの同期 .....	3-7
<b>第4章 コマンド</b>	
4.1 コマンド一覧 .....	4-1
4.2 ASCaleグループ .....	4-8
4.3 CALCulationグループ .....	4-9
4.4 COMMunicateグループ .....	4-24
4.5 DISPlayグループ .....	4-26
4.6 FILEグループ .....	4-30
4.7 HCOPyグループ .....	4-33
4.8 HHIStogroup<x>, THIStogroup<x>グループ .....	4-35
4.9 IHIStogroupグループ .....	4-43
4.10 INPutグループ .....	4-50
4.11 MEASureグループ .....	4-53
4.12 MEMoryグループ .....	4-55
4.13 RECallグループ .....	4-57
4.14 SAMPlEグループ .....	4-58
4.15 SStartグループ .....	4-62
4.16 StARtグループ .....	4-62
4.17 StATusグループ .....	4-63
4.18 StOPグループ .....	4-64
4.19 StOReグループ .....	4-64
4.20 SYStemグループ .....	4-65
4.21 TVARiation<x>グループ .....	4-67
4.22 UNITグループ .....	4-69
4.23 共通コマンドグループ .....	4-70

**第5章 ステータスレポート**

5.1	ステータスレポートについて .....	5-1
5.2	ステータスバイト .....	5-2
5.3	標準イベントレジスタ .....	5-3
5.4	拡張イベントレジスタ .....	5-4
5.5	出力キューとエラーキュー .....	5-5

**第6章 サンプルプログラム**

6.1	プログラムを組む前に .....	6-1
6.2	サンプルプログラムイメージ .....	6-1
6.3	初期化・エラー・実行関数 .....	6-2
6.4	測定パラメータの設定/問い合わせ(GP-IB用) .....	6-6
6.5	周期測定の測定統計値の問い合わせ(GP-IB用) .....	6-8
6.6	周期A&周期B測定の測定統計値の問い合わせ(GP-IB用) .....	6-11
6.7	AtoBタイムインターバル測定の測定データの問い合わせ(GP-IB用) .....	6-14
6.8	測定パラメータの設定/問い合わせ(Ethernet用) .....	6-17
6.9	パルス幅測定の測定統計値の問い合わせ(Ethernet用) .....	6-19

**付録**

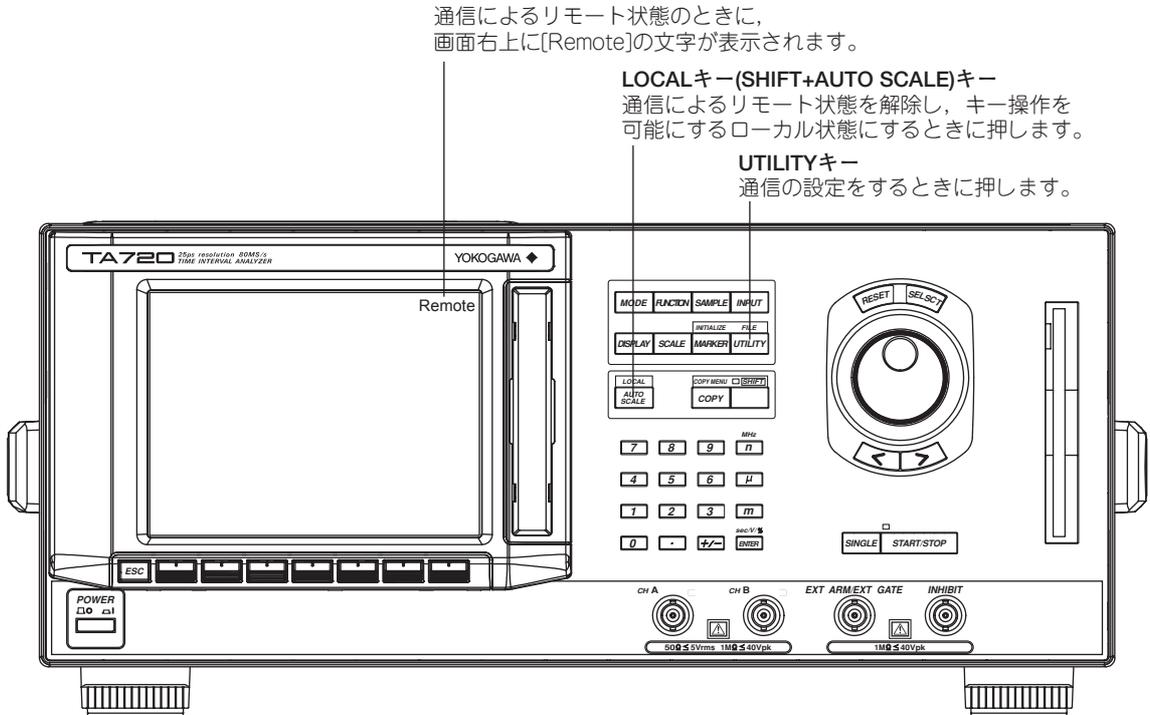
付録1	ASCIIキャラクタコード .....	付-1
付録2	エラーメッセージ .....	付-2
付録3	IEEE 488.3-1992について .....	付-5

**索引**

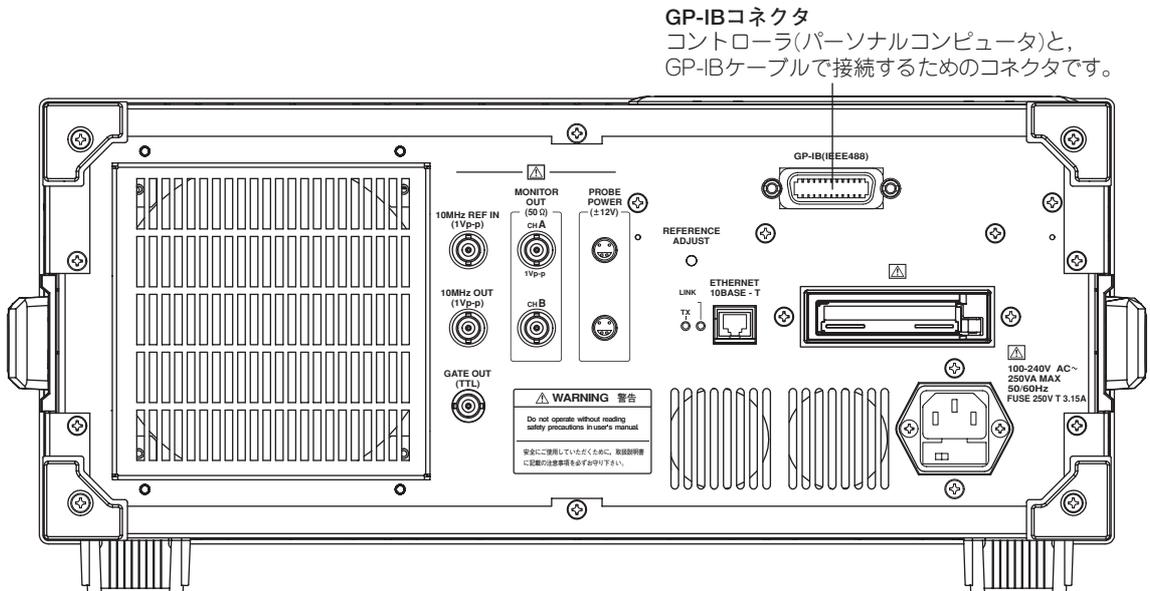
# 第1章 GP-IB通信インタフェースについて

## 1.1 GP-IB通信に関連する各部の名称と機能

### フロントパネル



### リアパネル



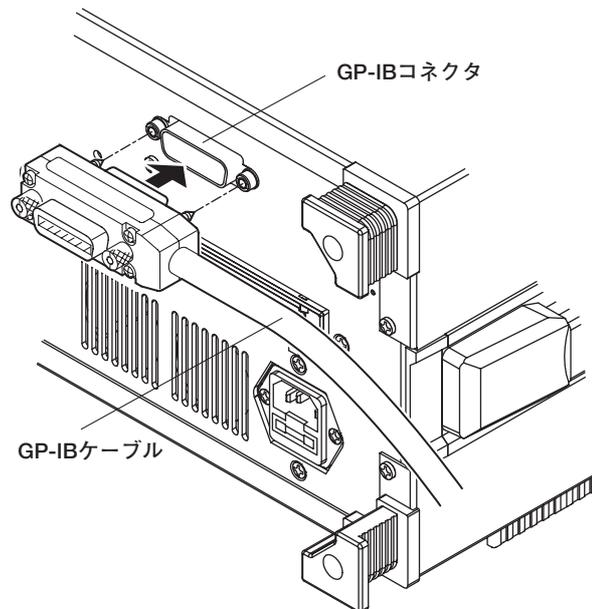
## 1.2 GP-IBケーブルの接続方法

### GP-IBケーブル

本機器のGP-IBコネクタは、IEEE St'd 488-1978規格の24ピンコネクタです。GP-IBケーブルは、IEEE St'd 488-1978に合ったものを使用してください。

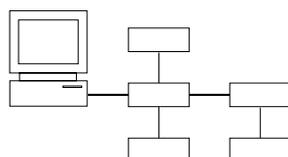
### 接続方法

下図のように接続してください。



### 接続時の注意

- GP-IBケーブルのコネクタに付いているねじは、しっかりと固定してください。
- 1つのGP-IBシステムには、複数の機器を接続することができます。ただし、1つのシステムには、コントローラを含め、15台までです。
- 複数の機器を接続するときは、それぞれのアドレスを同じに設定することはできません。
- 機器間をつなぐケーブルは2m以下のものを使用してください。
- ケーブルの長さは合計で20mを超えないようにしてください。
- 通信を行っているときは、少なくとも全体の2/3以上の機器の電源をONにしておいてください。
- 複数の機器を接続するときは、下図に示すようなスター形またはリニア形の結線にしてください。ループ形やパラレル形の結線はできません。



## 1.3 GP-IB通信の機能と仕様

### GP-IB通信の機能

#### ● リスナ機能

- ・ 電源のON/OFFと通信の設定を除き、本機器のキー操作で設定できる同じ内容の設定ができます。
- ・ 設定情報や測定データなどの、コントローラからの出力指令を受けることができます。
- ・ その他、ステータスレポートに関するコマンドなどを受けることができます。

#### ● トーカ機能

設定情報や測定データなどを出力することができます。

#### Note

リスンオンリ、トークオンリおよびコントローラ機能はありません。

### リモート/ローカル切り替え時の動作

#### ● ローカル→リモート切り替え時

ローカル状態のときにコントローラからREN(Remote Enable)のメッセージを受け取ると、リモート状態になります。

- ・ 画面右上(1-1ページ参照)に[Remote]の文字が表示されます。
- ・ **LOCAL(SHIFT+AUTO SCALE)**キー以外のキー操作はできなくなります。
- ・ ローカル状態での設定は、リモート状態になっても保持されます。

#### ● リモート→ローカル切り替え時

リモート状態のときに、**LOCAL(SHIFT+AUTO SCALE)**キーを押すと、ローカル状態になります。ただし、コントローラによりLocal Lockout(1-5ページ参照)になっているときは無効です。

- ・ 画面右上の[Remote]の文字が消えます。
- ・ キー操作が可能になります。
- ・ リモート状態での設定は、ローカル状態になっても保持されます。

### GP-IBインタフェースの仕様

電氣的・機械的仕様：IEEE St'd 488-1978(JIS C 1901-1987)に準拠

機能的仕様：下表

プロトコル：IEEE St'd 488.2-1992に準拠

使用コード：ISO(ASCII)コード

モード：アドレスサブルモード

アドレス選択：UTILITYキーのGP-IB設定メニューで0~30のアドレスを選択可能

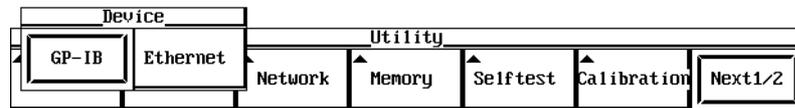
リモート状態解除：LOCAL(SHIFT+AUTO SCALE)キーを押すことで、リモート状態の解除可能。ただし、コントローラによりLocal Lockoutされているときのキー操作は、無効。

機能	サブセット名	内容
ソースハンドシェイク	SH1	送信ハンドシェイクの全機能あり
アクセプタハンドシェイク	AH1	受信ハンドシェイクの全機能あり
トーカ	T6	基本トーカ機能、シリアルポール、MLA(My Listen Address)によるトーカ解除機能あり、トークオンリ機能なし
リスナ	L4	基本リスナ機能、MTA(My Talk Address)によるリスナ解除機能あり、リスンオンリ機能なし
サービスリクエスト	SR1	サービスリクエストの全機能あり
リモートローカル	RL1	リモート/ローカルの全機能あり
パラレルポール	PP0	パラレルポール機能なし
デバイスクリア	DC1	デバイスクリアの全機能あり
デバイストリガ	DT1	デバイストリガ機能あり
コントローラ	C0	コントローラ機能なし
電気特性	E1	オープンコレクタ

## 1.4 本体の設定

## 操 作

1. UTILITYキーを押して，[Utility]メニューを表示します。
2. [Device]のソフトキーを押して，[Device]選択メニューを表示します。
3. ソフトキーで[GP-IB]を選択します。



4. ロータリノブまたはテンキーで，アドレスを設定します。



## 解 説

## アドレスの設定

GP-IBで接続できる各装置は，GP-IBシステム内で固有のアドレスを持ちます。このアドレスによって他の装置と識別されます。したがって本機器をパーソナルコンピュータなどに接続するときは，本機器のアドレスを他の機器のアドレスと重ならないように設定する必要があります。

アドレスは次の中から選択します。初期設定は「1」です。

選択範囲：0～30

## 1.5 インタフェースメッセージに対する応答

### インタフェースメッセージとは

インタフェースメッセージは、インタフェースコマンドまたはバスコマンドとも呼ばれ、コントローラから発せられるコマンドのことです。次のような分類になっています。

#### ● ユニラインメッセージ

1本の管理ラインを経由してメッセージを送ります。次の3種類があります。  
IFC(Interface Clear), REN(Remote Enable), IDY(Identify)

#### ● マルチラインメッセージ

8本のデータラインを経由してメッセージを送ります。次のように分類されます。

##### ・ アドレスコマンド

機器がリスナあるいはトーカに指定されているときに有効なコマンドです。次の5種類があります。

##### ・ リスナに指定している機器に有効なコマンド

GTL(Go To Local), SDC(Selected Device Clear), PPC(Parallel Poll Configure), GET(Group Execute Trigger)

##### ・ トーカに指定している機器に有効なコマンド

TCT(Take Control)

##### ・ ユニバーサルコマンド

リスナ・トーカの指定の有無に関わらず、すべての機器に有効です。次の5種類があります。

LLO(Local Lockout), DCL(Device Clear), PPU(Parallel Poll Unconfigure), SPE(Serial Poll Enable), SPD(Serial Poll Disable)

##### ・ その他、インタフェースメッセージとして、リスナアドレス、トーカアドレス、2次コマンドがあります。

##### ・ SDCとDCLの違い

マルチラインメッセージのうち、SDCはトーカ・リスナの指定が必要なアドレスコマンド、DCLはトーカ・リスナの指定が不要なユニバーサルコマンドです。したがって、SDCはある特定の機器を対象にしますが、DCLはバス上のすべての機器を対象にします。

### インタフェースメッセージに対する応答

#### ● ユニラインメッセージに対する応答

- ・ IFC : トーカ、リスナを解除します。データ出力中のときは出力を中止します。
- ・ REN : リモート状態/ローカル状態を切り替えます。
- ・ IDY : サポートしていません。

#### ● マルチラインメッセージ(アドレスコマンド)に対する応答

- ・ GTL : ローカル状態へ移行します。
- ・ SDC : 受信中のプログラムメッセージ(コマンド)と、出力キュー(5-5ページ参照)をクリアします。
- ・ GET : \*TRGと同じ動作をします。
- ・ COMMunicate:WAITは直ちに終了します。
- ・ PPC, TCT : サポートしていません。

#### ● マルチラインメッセージ(ユニバーサルコマンド)に対する応答

- ・ LLO : フロントパネルのLOCALキーの操作を無効にし、ローカル状態への移行を禁止します。
- ・ DCL : SDCと同じ動作をします。
- ・ SPE : バス上のすべての機器のトーカ機能をシリアルポールモードにします。コントローラは各機器を順番にポーリングします。
- ・ SPD : バス上のすべての機器のトーカ機能のシリアルポールモードを解除します。
- ・ PPU : サポートしていません。

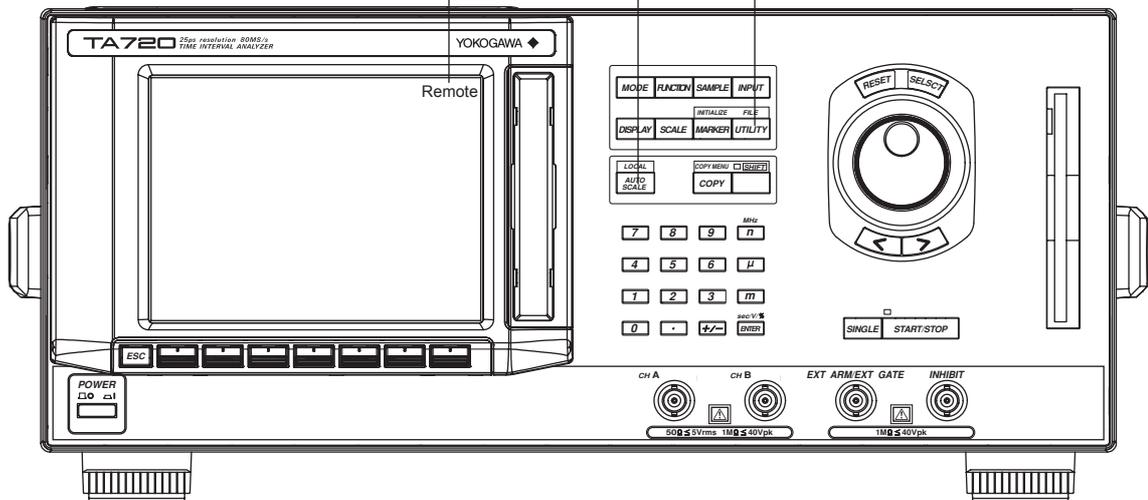
## 第2章 イーサネット通信インタフェース(オプション)について

### 2.1 イーサネット通信に関連する各部の名称と機能 フロントパネル

通信によるリモート状態のときに、画面右上に[Remote]の文字が表示されます。

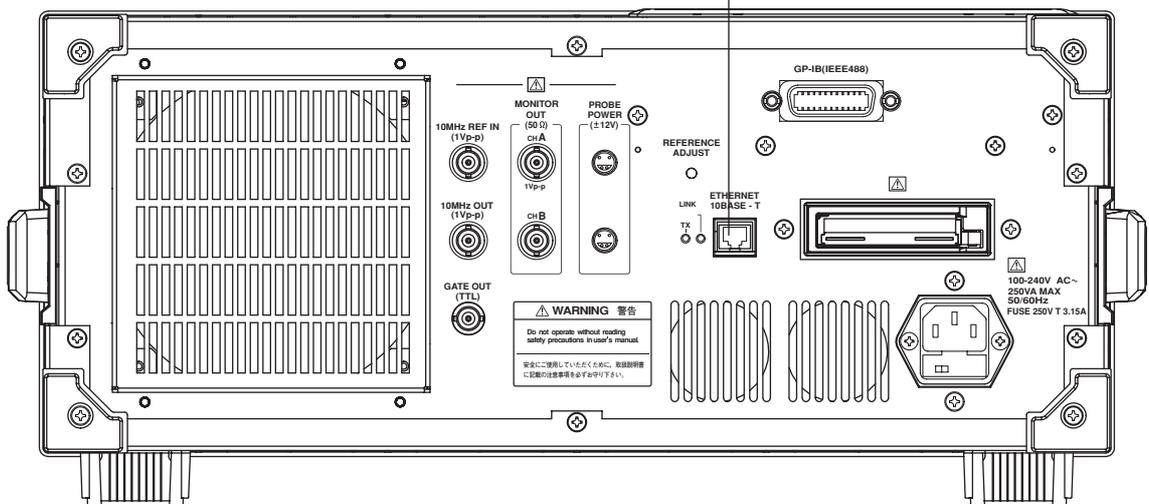
**LOCALキー-(SHIFT+AUTO SCALE)キー**  
通信によるリモート状態を解除し、キー操作を可能にするローカル状態にするときに押します。

**UTILITYキー**  
ユーザ認証用のユーザ名/パスワードの設定やTCP/IPの設定をするときに押します。



### リアパネル

**イーサネットインタフェースコネクタ**  
コントローラ(パーソナルコンピュータなど)と、イーサネットケーブルで接続するためのコネクタです。接続の方法は、2-3ページをお読みください。



### 2.2 イーサネット通信の機能と仕様

#### 受信機能

フロントパネルのキー操作による設定と同じ設定ができます。  
測定/演算データ、パネルの設定情報、エラーコードの出力要求を受けられます。

#### 送信機能

測定/演算データを出力できます。  
パネルの設定情報、ステータスバイトを出力できます。  
発生したエラーコードを出力できます。

#### イーサネット通信インタフェースの仕様

通信ポート数 : 1  
電氣的・機械的仕様 : IEEE802.3に準拠  
伝送方式 : 10BASE-T  
最大伝送速度 : 10Mbps  
コネクタ形状 : RJ-45

#### リモート/ローカル切り替え時の動作

##### ● ローカル→リモート切り替え時

ローカル状態のときにコンピュータから「:COMMunicate:REMOte ON」コマンドを受信すると、リモート状態になります。

- ・ 画面右上にREMOTEと表示されます。
- ・ **LOCAL(SHIFT+AUTO SCALE)**キー以外はキーが効かなくなります。
- ・ ローカル状態での設定は、リモート状態になっても保持されます。

##### ● リモート→ローカル切り替え時

リモート状態のときに**LOCAL(SHIFT+AUTO SCALE)**キーを押すと、ローカル状態になります。ただし、コンピュータから「:COMMunicate:LOCKout ON」コマンドを受信している(ローカルロックアウト状態)ときは無効です。

また、コンピュータから「:COMMunicate:REMOte OFF」コマンドを受信すると、ローカルロック状態に関係なく、ローカル状態になります。

- ・ 画面右上のREMOTE表示が消えます。
- ・ キー操作が可能になります。
- ・ リモート状態での設定は、ローカル状態になっても保持されます。

#### **Note**

---

イーサネット通信インタフェースは、GP-IBインタフェースと同時に使用できません。

---

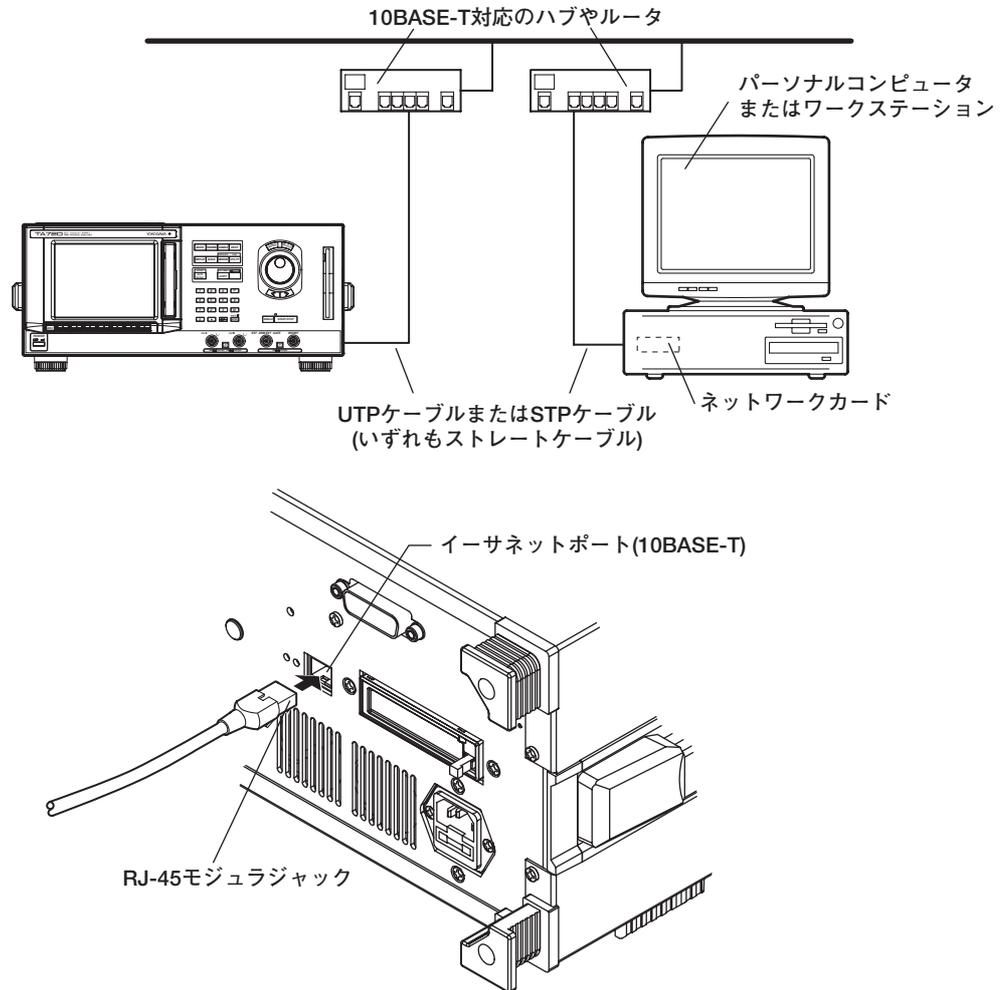
#### ユーザー認証機能

イーサネット通信インタフェースでは、ネットワーク接続時にユーザー名とパスワードを入力する必要があります。ユーザー名とパスワードは、本機器のUtilityメニューの[User Account]ポップアップウィンドウで設定します。詳細は、「2.4 本体の設定」をご覧ください。

## 2.3 イーサネット通信インタフェースによる接続

### 接続方法

ハブなどに接続されたUTP(Unshielded Twisted-Pair)ケーブルまたはSTP(Shielded Twisted-Pair)ケーブルを本機器のリアパネルにある10BASE-Tポートに接続します。



### 接続時の注意

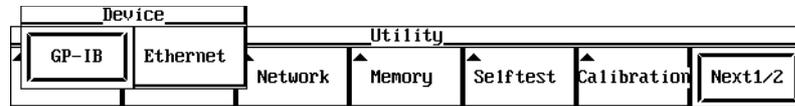
本機器とパーソナルコンピュータとの接続には、必ずハブを介してストレートケーブルを使用してください。クロスケーブルでの1対1の接続では、動作を保證することができません。

## 2.4 本体の設定

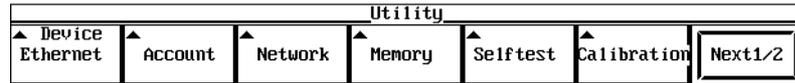
## 操作手順

## ● 通信インタフェースの選択

- UTILITYキーを押して、[UTILITY]メニューを表示します。
- [Device]のソフトキーを押して、[Device]選択メニューを表示します。



- [Ethernet]のソフトキーを押します。

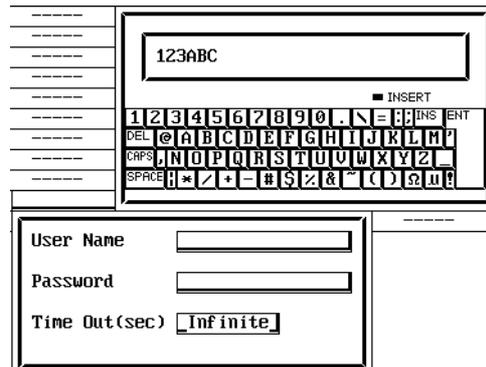
**Note**

Deviceで選択した通信インタフェースだけが有効です。選択していないGP-IB通信インタフェースでコマンドを送信しても、本機器は受け付けません。

## ● ユーザー名/パスワード/タイムアウト時間の設定

- [Account]のソフトキーを押します。  
ユーザー認証ダイアログボックスが表示されます。

- ロータリノブを回して、[User Name]にカーソルを移動します。
- SELECTキーを押して、キーボードを表示します。
- ユーザー名を入力します。アクセス制限をしない場合は、[anonymous]に設定します。アクセス制限する場合は、ユーザー名を15文字以内で入力します。



- ロータリノブを回して、[Password]にカーソルを移動します。
- SELECTキーを押して、キーボードを表示します。
- パスワードを15文字以内で入力します。
- ロータリノブを回して、[Time Out]にカーソルを移動します。
- SELECTキーを押します。
- ロータリノブまたはテンキーで、タイムアウト時間を設定します。

- SELECTキーを押して、設定を確定します。

**Note**

- ・ キーボードの操作方法については、本体ユーザズマニュアル IM704510-01の「4.4 数値および文字列の入力方法」をご覧ください。
- ・ ユーザー名とパスワードは、大文字、小文字の区別があります。

**● TCP/IPの設定**

ネットワークインタフェース機能を利用するには、TCP/IPの以下の設定が必要です。

- ・ IPアドレス
- ・ サブネットマスク
- ・ デフォルトゲートウェイ

これらの設定方法の詳細については、本体ユーザズマニュアル IM704510-01の「12.2 TCP/IPの設定」をご覧ください。

**解 説**

本機器のキー操作で設定できる内容をコントローラで設定するときや、コントローラに設定情報や波形データを出力するときは、下記の設定をします。

**● ユーザー名/パスワードの設定**

ネットワークインタフェースには、ユーザー認証機能があります。あらかじめ本機器のユーザー名とパスワードを設定しておきます。

- ・ ユーザー名の設定  
15文字以内で設定します。初期値は「anonymous」です。
- ・ パスワードの設定  
15文字以内で設定します。

**● タイムアウト時間の設定**

ここで設定した時間、本機器にアクセスがないと、自動的に本機器との接続が切断されます。初期設定は、[Infinite](タイムアウトなし)です。

**● TCP/IPの設定**

詳細は、本体ユーザズマニュアル (IM704510-01)の「12.2 TCP/IPの設定」をご覧ください。

**Note**

- ・ ユーザー認証エラーとなった場合、本機器との接続が切断されます。
- ・ ユーザー名が「anonymous」の場合は、パスワードは不要です。
- ・ ユーザー名、パスワード、タイムアウト時間、TCP/IPの設定を変更した場合、設定内容を反映させるには、電源を入れ直してください。

## 第3章 プログラム書式とプログラム作成時の注意事項

### 3.1 メッセージ

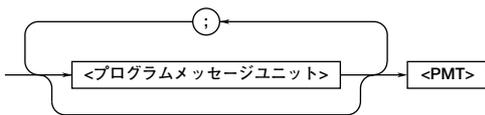
#### メッセージ

コントローラと本機器の間の送受信は、メッセージという単位で行います。コントローラから本機器に送信するメッセージをプログラムメッセージといい、コントローラが本機器から受信するメッセージを応答メッセージといいます。

プログラムメッセージの中に応答を要求する命令(クエリといいます)があるときは、本機器はプログラムメッセージを受信したあとに、応答メッセージを送信します。1つのプログラムメッセージに対する応答は、必ず1つの応答メッセージになります。

#### プログラムメッセージ

コントローラから本機器に送信するデータをプログラムメッセージといいます。プログラムメッセージの書式は次のようになります。



#### <プログラムメッセージユニット>

プログラムメッセージは、1つ以上のプログラムメッセージユニットをつないだものです。プログラムメッセージユニットが1つの命令に相当します。本機器は受信した順序で命令を実行していきます。

プログラムメッセージユニットは「;」(セミコロン)で区切ります。

プログラムメッセージの書式については、次項を参照してください。

例

```
:MEASURE:MODE HHISTOGRAM;FUNCTION PERIOD,A<PMT>
```

ユニット
ユニット

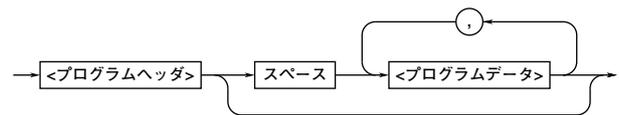
#### <PMT>

プログラムメッセージのターミネータです。次の3種類があります。

- NL(ニューライン) : LF(ラインフィード)と同じ、ASCIIコード「0AH」の一文字
- ^END : IEEE488.1で定義されているENDメッセージ(EOI信号)  
(ENDメッセージと同時に送信されたデータバイトは、プログラムメッセージの最後のデータになります)
- NL^END : ENDメッセージが追加されたNL  
(NLはプログラムメッセージには含まれません)

#### ● プログラムメッセージユニットの書式

プログラムメッセージユニットの書式は次のようになります。



#### <プログラムヘッダ>

プログラムヘッダは命令の種類を表わします。詳しくは、3-3ページを参照してください。

#### <プログラムデータ>

命令を実行するときに必要な条件などがあるときは、プログラムデータを付けます。プログラムデータを付けるときは、ヘッダとデータをスペース(ASCIIコード「20H」)で区切ります。複数のデータがあるときは、データとデータの間を「,」(カンマ)で区切ります。詳しくは、3-5ページを参照してください。

例 :SAMPLE:GATE:MODE EVENT<PMT>

ヘッダ
データ



## 3.2 命令

### 命令

コントローラから本機器に送信される命令(プログラムヘッダ)には、以下に示す3種類があります。それぞれプログラムヘッダの書式が異なります。

#### 共通コマンドヘッダ

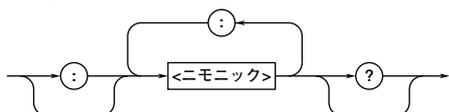
IEEE 488.2-1992で規定されている命令を共通コマンドといいます。共通コマンドのヘッダの書式は次のようになります。先頭に必ず「\*」(アスタリスク)を付けます。



共通コマンドの例：\*CLS

#### 複合ヘッダ

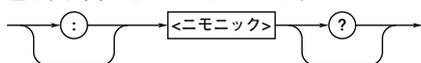
共通コマンド以外の本機器固有の命令は、機能ごとに分類されて、階層化されています。複合ヘッダの書式は次のようになります。下の階層を記述するときは、必ず「:」(コロン)を付けます。



複合ヘッダの例：:MEASURE:FUNCTION

#### 単純ヘッダ

機能的に独立した、階層を持たない命令です。ヘッダの書式は次のようになります。



単純ヘッダの例：:START

#### Note

<ニモニック>とは、アルファベットと数字からなる文字列です。

#### 命令を続けて記述する場合

##### ● グループについて

ヘッダが階層化された共通の複合ヘッダを持つコマンド群をグループといいます。グループの中にさらに小さいグループが存在することもあります。

例 サンプリングに関するグループ

```
:SAMPLE?
:SAMPLE:GATE?
:SAMPLE:GATE:MODE
:SAMPLE:GATE:EVENTSIZE
:SAMPLE:GATE:POLARITY
:SAMPLE:GATE:TIME
:SAMPLE:INTERVAL
:SAMPLE:ARMING:SOURCE
:SAMPLE:ARMING:DELAY:MODE
:SAMPLE:ARMING:DELAY:TIME
```

##### ● 同じグループの命令を続けて記述する場合

本機器は、実行している命令がどの階層の命令であるかを記憶し、次に送信した命令も同じ階層に属しているものと仮定して解析を行っています。したがって、同じグループの命令は、共通のヘッダの部分を省略することができます。

例 :INPUT:ACHANNEL:COUPLING AC;  
IMPEDANCE I50<PMT>

##### ● 違うグループの命令を続けて記述する場合

グループが違う命令を後ろに記述するときは、ヘッダの先頭に「:」(コロン)を付けます。

例 :MEASURE:MODE TSTAMP;:DISPLAY:  
ITEM LIST<PMT>

##### ● 単純ヘッダを続けて記述する場合

他の命令に単純ヘッダを続けるときは、単純ヘッダの先頭に「:」(コロン)を付けます。

例 :MEASURE:MODE TSTAMP;:START<PMT>

##### ● 共通コマンドを続けて記述する場合

IEEE 488.2-1992で定義された共通コマンドは、階層には無関係です。「:」(コロン)はつける必要はありません。

例 :MEASURE:MODE TSTAMP;\*CLS;  
FUNCTION PERIOD,A<PMT>

### ● コマンド間を<PMT>で区切った場合

ターミナーで区切ると、2つのプログラムメッセージを送信することになります。したがって、同じグループでのコマンドを続ける場合でも、共通のヘッダを省略することはできません。

```
例 :MEASURE:MODE TSTAMP<PMT>:MEASURE:
    FUNCTION PERIOD,A<PMT>
```

### 上位クエリ

初めて出てくるグループの最上位のコマンドに「?」を付けたクエリを上位クエリといいます。この上位クエリを実行すると、そのグループで設定できるすべての設定をまとめて受信することができます。階層が3階層以上あるグループで、下の階層をすべて出力するものもあります。

```
例 :MEASURE?<PMT> -> :MEASURE:
    MODE HHISTOGRAM;FUNCTION PERIOD,A;
    SLOPE RISE
```

上位クエリの応答は、そのまま本機器にプログラムメッセージとして送信することができます。送信すると、上位クエリを行ったときの設定を再現できます。ただし、上位クエリでは現在使われていない設定情報を返さないものもあります。必ずしもそのグループのすべての情報が応答として出力されるわけではないので、注意してください。

### ヘッダの解釈の規則

本機器は、受信したヘッダを次の規則に従って解釈します。

- ・ ニモニックのアルファベットの太文字/小文字は区別しません。  
例 「MEASure」->「measure」「Measure」でも可
  - ・ 小文字の部分は省略できます。  
例 「MEASure」->「MEASU」「MEAS」でも可
  - ・ ヘッダの最後の「?」(クエスチョンマーク)は、クエリであることを示します。「?」は省略できません。  
例 「MEASure?」->最小の省略形は「MEAS?」
  - ・ ニモニックの最後に付いている<x>(数値)を省略すると、x=1と解釈します。  
例 「WINDow<x>」->「WIND」とすると「WINDow1」の意味
  - ・ []で囲まれた部分は省略できます。  
例 :CALCulation[:WINDow1]:AVERage? -> :CALCulation:AVERage?でも可
- ただし上位クエリの場合、最後の部分は省略できません。

## 3.3 応答

コントローラが「?」の付いた命令であるクエリを送信すると、本機器はそのクエリに対する応答メッセージを返します。返される形式は、次の2つに分けられます。

- ・ ヘッダ+データの応答  
応答をそのままプログラムメッセージとして利用できるものは、命令のヘッダを付けて返されます。  
例 :SAMPLE:GATE:MODE?<PMT> -> :SAMPLE:GATE:MODE EVENT<RMT>
- ・ データだけの応答  
そのままプログラムメッセージとして利用できないもの(クエリ専用の命令)は、ヘッダを付けずにデータだけで返されます。ただし、ヘッダを付けて返すクエリ専用の命令もあります。  
例 :STATUS:ERROR?<PMT> -> 0,"NO ERROR"<RMT>

### ● ヘッダを付けたい応答を返したい場合

「ヘッダ+データ」で返されるものでも、ヘッダを強制的に付けないようにすることができます。これには、「:COMMunicate:HEADer」命令を使用します。

### ● 省略形について

応答のヘッダは、通常は小文字の部分を省略した形で返されます。これを省略しないフルスペルにすることもできます。これには、「:COMMunicate:VERBoSe」命令を使用します。

## 3.4 データ

### データ

データとは、ヘッダの後ろにスペースを空けて記述する条件や数値です。データは次のように分類されます。

データ	意味
<10進数>	10進数で表された数値 (例: 測定サンプル数 -> :SAMPLE:GATE:EVENTsize 100)
<電圧><時間>	物理的な次元を持った数値
<周波数>	(例: ゲート時間)
<パーセント>	-> :SAMPLE:GATE:TIME 1US)
<Register>	2, 8, 10, 16進数のどれかで表されたレジスタ値 (例: 拡張イベントレジスタ値 -> :STATUS:EESE #HFE)
<文字データ>	規定された文字列(二モニック)。{}内から選択 (例: ゲートモードの選択 -> :SAMPLE:GATE:MODE {EVENT TIME EXTERNAL})
<Boolean>	ON/OFFを表す。「ON」「OFF」または数値で設定 (例: パノラマ表示をON -> :DISPLAY:PANorama:STATE ON)
<文字列データ>	任意の文字列 (例: 削除ファイル名 -> :FILE:DELETE:SETup "SETUP_1")
<ブロックデータ>	任意の8ビットの値を持つデータ (例: 取り込んだ測定データの応答 -> #6000010ABCDEFGHJ)

### <10進数>

<10進数>は下表のように10進数で表現された数値です。なお、これはANSI X3.42-1975で規定されているNR形式で記述します。

記号	意味	例
<NR1>	整数	125 -1 +1000
<NR2>	固定小数点数	125.0 -.90 +001.
<NR3>	浮動小数点数	125.0E+0 -9E-1 +.1E4
<NRf>	<NR1>~<NR3>のどれでも可能	

- 本機器がコントローラから送られた10進数を受け取る時は、<NR1>~<NR3>のどの形式でも受け付けます。これを<NRf>で表します。
- 本機器からコントローラに返される応答メッセージは、<NR1>~<NR3>のどれを使用するかはクエリごとに決められています。値の大きさによって使用する形式が変わることはありません。
- <NR3>形式の場合、「E」のあとの「+」は省略できます。「-」は省略できません。
- 設定範囲外の値を記述したときは、設定できる値でいちばん近い値になります。
- 精度以上の値を記述したときは、四捨五入します。

### <電圧>, <時間>, <周波数>, <パーセント>

<電圧>, <時間>, <周波数>, <パーセント>は、<10進数>のうち物理的な次元を持ったデータです。<パーセント>を除き、前述の<NRf>形式に<乗数>および<単位>を付けることができます。次の書式のどれかで記述します。

書式	例
<NRf><乗数><単位>	5MV
<NRf><単位>	5E-3V
<NRf><乗数>	5M
<NRf>	5E-3

### ・<乗数>

使用できる<乗数>は下表のとおりです。

記号	読み	乗数
EX	エクサ	10 <sup>18</sup>
PE	ペタ	10 <sup>15</sup>
T	テラ	10 <sup>12</sup>
G	ギガ	10 <sup>9</sup>
MA	メガ	10 <sup>6</sup>
K	キロ	10 <sup>3</sup>
M	ミリ	10 <sup>-3</sup>
U	マイクロ	10 <sup>-6</sup>
N	ナノ	10 <sup>-9</sup>
P	ピコ	10 <sup>-12</sup>
F	フェムト	10 <sup>-15</sup>
A	アト	10 <sup>-18</sup>

### ・<単位>

使用できる<単位>は下表のとおりです。

記号	読み	意味
V	ボルト	電圧
S	セカンド	時間
HZ	ヘルツ	周波数
MHZ	メガヘルツ	周波数
PCT	パーセント	パーセント

- <乗数>と<単位>は、大文字/小文字の区別がありません。
- マイクロの「μ」は「U」で表します。
- メガの「M」はミリと区別するため、「MA」で表します。
- <乗数>も<単位>も省略したときは、デフォルトの単位になります。
- 応答メッセージは必ず<NR3>形式になります。また、<乗数>および<単位>をつけずにデフォルトの単位で返します。

### 3.4 データ

#### <Register>

<Register>は整数ですが、<10進数>のほかに<16進数><8進数><2進数>でも表現できるデータです。数値がビットごとに意味を持つときに使用します。次の書式のどれかで記述します。

書式	例
<NRf>	1
#H<0~9, A~Fからなる16進数>	#H0F
#Q<0~7からなる8進数>	#q777
#B<0または1からなる2進数>	#B001100

- ・<Register>は、大文字/小文字の区別はありません。
- ・応答メッセージは必ず<NR 1>で返されます。

#### <文字データ>

<文字データ>は、規定された文字(ニモニック)のデータです。主に選択肢を表現するときに使用され、{}内の文字列からどれか1つを選んで記述します。データの解釈のしかたは、3-4ページの「ヘッダ解釈の規則」と同様です。

書式	例
{EVENT TIME EXtErnal}	EVENT

- ・応答メッセージでは、ヘッダと同様に「:COMMunicate:VERBose」を使って、フルスペルで返すか、省略形で返すかを選ぶことができます。
- ・「:COMMunicate:HEADer」の設定は<文字データ>には影響しません。

#### <Boolean>

<Boolean>は、ONまたはOFFを示すデータです。次の書式のどれかで記述します。

書式	例
{ON OFF <NRf>}	ON OFF 1 0

- ・<NRf>で表す場合は、整数に丸めた値が「0」のときがOFF、「0以外」のときがONになります。
- ・応答メッセージは必ず、ONのときは「1」、OFFのときは「0」で返されます。

#### <文字列データ>

<文字列データ>は、<文字データ>のように規定された文字列ではなく、任意の綴りの文字列です。次のように、「'」(シングルクォーテーション)または「"」(ダブルクォーテーション)で囲った書式で記述します。

書式	例
<文字列データ>	'ABC' "IEEE488.2-1987"

- ・「" "」内に文字列として「"」があるときは、「""」で表します。「'」のときも同様です。
- ・応答メッセージは、必ず「"」(ダブルクォーテーション)で囲って返されます。
- ・<文字列データ>は任意の綴りなので、最後の「'」(シングルクォーテーション)または「"」(ダブルクォーテーション)がないと、本機器は残りのプログラムメッセージユニットを<文字列データ>の一部と解釈してしまい、エラーが正しく検出できない場合があります。

#### <ブロックデータ>

<ブロックデータ>は、任意の8ビットの値を持つデータです。本機器では、応答メッセージだけに使用されます。書式は次のとおりです。

書式	例
#8<8桁の10進数> <データバイトの並び>	#800000010ABCDEFGHIJ

- ・#8  
<ブロックデータ>であることを表します。
- ・<8桁の10進数>  
データのバイト数を表します(00000010=10バイト)。
- ・<データバイトの並び>  
実際のデータを表します(ABCDEFGHIJ)。
- ・データは8ビットでとり得る値(0~255)です。したがって、「NL」を示すASCIIコード「0AH」もデータになることがありますので、コントローラ側では注意が必要です。

### 3.5 コントローラとの同期

TA720では、先の動作が完了する前に次のコマンドによる動作が開始することが許されるオーバーラップコマンドはありません。連続してコマンドを送信したときは、先の動作が終了するまで、次のコマンドによる動作の実行を待ちます。

#### 同期をとる方法

トリガの発生など、通信以外の要因と同期をとる必要がある場合もあります。

たとえば、シングル測定した測定データを問い合わせるときに、次のプログラムメッセージを送信すると、データの取り込みの終了いかんにかかわらず「:MEMORY:SEND?」が実行され、コマンド実行エラーになる可能性があります。

```
:SStart;:MEMory:SEND?<PMT>
```

このときは、次に示す方法で取り込みが終了したときのタイミングをとる必要があります。

#### ● STATUS:CONDition?クエリを使う

「:STATUS:CONDition?」は状態レジスタ(5-4ページ参照)の内容を問い合わせる命令です。測定データが有効かそうでないかは、状態レジスタのビット0を読むことで判断できます。状態レジスタのビット0が「1」なら測定データは有効、「0」なら測定中または演算実行中で、測定データは無効であることを示します。

```
例 :SStart<PMT>
```

```
:STATus:CONDition?<PMT>
```

(応答を読んでビット0が0なら1つ前に戻る)

```
:MEMory:SEND?<PMT>
```

「:MEMORY:SEND?」は、状態レジスタのビット0が「1」になるまで実行されません。

#### ● 拡張イベントレジスタを使う

状態レジスタの変化は、拡張イベントレジスタ(5-4ページ)に反映させることができます。

```
例 :STATus:FILTer1 RISE;:STATus:
```

```
EESE 1;EESR?;*SRE 8;SStart<PMT>
```

(サービスリクエストの発生を待つ)

```
:MEMory:SEND?<PMT>
```

「:STATUS:FILTer1 RISE」は、状態レジスタのビット0が「0」から「1」に変化したときに、拡張イベントレジスタのビット0(FILTer1)を「1」にセットするように、遷移フィルタを設定することを示しています。

「:STATUS:EESE 1」は、拡張イベントレジスタのビット0だけをステータスバイトに反映するようにする命令です。

「:STATUS:EESR?」は、拡張イベントレジスタをクリアするために行っています。

「\*SRE」は、拡張イベントレジスタの原因だけでサービスリクエストが発生するようにする命令です。

「:MEMORY:SEND?」は、サービスリクエストが発生するまで実行されません。

#### ● COMMunicate:WAITコマンドを使う

「:COMMunicate:WAIT」は、特定のイベントが発生するのを待つ命令です。

```
例 :STATus:FILTer1 RISE;:STATus:EESR?;
```

```
:SStart<PMT>
```

(STATUS:EESR?の応答を読む)

```
:COMMunicate:WAIT 1;:MEMory:
```

```
SEND?<PMT>
```

「:STATUS:FILTer1 RISE」および「:STATUS:EESR?」の意味は、前述の拡張イベントレジスタの場合と同一です。

「:COMMunicate:WAIT 1」は、拡張イベントレジスタのビット0が「1」にセットされるのを待つことを示しています。

「:MEMORY:SEND?」は、拡張イベントレジスタのビット0が「1」になるまで実行されません。

## 第4章 コマンド

### 4.1 コマンド一覧

コマンド	機能	ページ
<b>ASCaLeグループ</b>		
:ASCaLe	オートスケーリングの実行	4-8
<b>CALCulationグループ</b>		
:CALCulation?	統計値に関する全設定の問い合わせ	4-10
:CALCulation:AREA	演算範囲の設定/問い合わせ	4-10
:CALCulation[:MEAS<x>]:AUTot?	オートウインドウトの問い合わせ	4-11
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}]:TAVerage?	タイムバリエーション時の平均値の問い合わせ	4-11
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}]:TFLutter?	タイムバリエーション時の $\sigma$ /AVE値(フラッタ)の問い合わせ	4-11
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}]:TJITter?	タイムバリエーション時のP-P/AVE値(ジッタ)の問い合わせ	4-12
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}]:TMAXimum?	タイムバリエーション時の最大値の問い合わせ	4-12
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}]:TMINimum?	タイムバリエーション時の最小値の問い合わせ	4-13
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}]:TPTopeak?	タイムバリエーション時のP-P値の問い合わせ	4-13
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}]:TRF?	タイムバリエーション時のRF値の問い合わせ	4-14
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}]:TSDeviation?	タイムバリエーション時の標準偏差( $\sigma$ )の問い合わせ	4-14
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}]:TSNumber?	タイムバリエーション時の統計演算対象のサンプル数の問い合わせ	4-15
:CALCulation[:MEAS<x>]:CONSTt	定数Tの設定/問い合わせ	4-15
:CALCulation[:MEAS<x>][:WINDow<x>]:AVERage?	ヒストグラム時の平均値の問い合わせ	4-16
:CALCulation[:MEAS<x>][:WINDow<x>]:DEViation?	ヒストグラム時のDeviationの問い合わせ	4-16
:CALCulation[:MEAS<x>][:WINDow<x>]:DEVT?	ヒストグラム時のDeviation/T値の問い合わせ	4-17
:CALCulation[:MEAS<x>][:WINDow<x>]:FLUTter?	ヒストグラム時の $\sigma$ /AVE値(フラッタ)の問い合わせ	4-17
:CALCulation[:MEAS<x>][:{WINDow<x> SUMMation}]:JITTer?	ヒストグラム時の $\sigma$ /T値(ジッタ)の問い合わせ	4-18
:CALCulation[:MEAS<x>][:WINDow<x>]:MAXimum?	ヒストグラム時の最大値の問い合わせ	4-18
:CALCulation[:MEAS<x>][:WINDow<x>]:MEDian?	ヒストグラム時の中央値の問い合わせ	4-19
:CALCulation[:MEAS<x>][:WINDow<x>]:MINimum?	ヒストグラム時の最小値の問い合わせ	4-19
:CALCulation[:MEAS<x>][:WINDow<x>]:MODE?	ヒストグラム時の最頻値の問い合わせ	4-20
:CALCulation[:MEAS<x>][:{WINDow<x> SUMMation}]:PTOPeak?	ヒストグラム時のP-P値の問い合わせ	4-20
:CALCulation[:MEAS<x>][:{WINDow<x> SUMMation}]:SDEViation?	ヒストグラム時の標準偏差( $\sigma$ )の問い合わせ	4-21
:CALCulation[:MEAS<x>][:{WINDow<x> SUMMation}]:SNUMBER?	ヒストグラム時の統計演算対象のサンプル数の問い合わせ	4-21
:CALCulation:PARameter?	統計演算全体のON/OFFの問い合わせ	4-22

## 4.1 コマンド

コマンド	機能	ページ
:CALCulation:PARAmeter:CLEar	全統計演算値のオフの実行	4-22
:CALCulation:PARAmeter:AVErAge	ヒストグラム時の平均値の演算のON/OFFの設定/問い合わせ	4-22
:CALCulation:PARAmeter:DEVIation	ヒストグラム時のDeviationの演算のON/OFFの設定/問い合わせ	4-22
:CALCulation:PARAmeter:DEVT	ヒストグラム時のDeviation/T値の演算のON/OFFの設定/問い合わせ	4-22
:CALCulation:PARAmeter:FLUTter	ヒストグラム時の $\sigma$ /AVE値(フラッタ)の演算のON/OFFの設定/問い合わせ	4-22
:CALCulation:PARAmeter:JITter	ヒストグラム時の $\sigma$ /T値(ジッタ)の演算のON/OFFの設定/問い合わせ	4-22
:CALCulation:PARAmeter:MAXimum	ヒストグラム時の最大値の演算のON/OFFの設定/問い合わせ	4-22
:CALCulation:PARAmeter:MEDIan	ヒストグラム時の中央値の演算のON/OFFの設定/問い合わせ	4-22
:CALCulation:PARAmeter:MINimum	ヒストグラム時の最小値の演算のON/OFFの設定/問い合わせ	4-22
:CALCulation:PARAmeter:MODE	ヒストグラム時の最頻値の演算のON/OFFの設定/問い合わせ	4-22
:CALCulation:PARAmeter:PTOPeak	ヒストグラム時のP-P値の演算のON/OFFの設定/問い合わせ	4-23
:CALCulation:PARAmeter:SDEVIation	ヒストグラム時の標準偏差( $\sigma$ )の演算のON/OFFの設定/問い合わせ	4-23
:CALCulation:PARAmeter:TAVerage	タイムバリエーション時の平均値の演算のON/OFFの設定/問い合わせ	4-23
:CALCulation:PARAmeter:TFLutter	タイムバリエーション時の $\sigma$ /AVE値(フラッタ)の演算のON/OFFの設定/問い合わせ	4-23
:CALCulation:PARAmeter:TJITter	タイムバリエーション時のP-P/AVE値(ジッタ)の演算のON/OFFの設定/問い合わせ	4-23
:CALCulation:PARAmeter:TMAXimum	タイムバリエーション時の最大値の演算のON/OFFの設定/問い合わせ	4-23
:CALCulation:PARAmeter:TMINimum	タイムバリエーション時の最小値の演算のON/OFFの設定/問い合わせ	4-23
:CALCulation:PARAmeter:TPTopeak	タイムバリエーション時のP-P値の演算のON/OFFの設定/問い合わせ	4-23
:CALCulation:PARAmeter:TRF	タイムバリエーション時のRF値の演算のON/OFFの設定/問い合わせ	4-23
:CALCulation:PARAmeter:TSDeviation	タイムバリエーション時の標準偏差( $\sigma$ )の演算のON/OFFの設定/問い合わせ	4-23
:CALCulation:POLarity	両パルス幅/両エッジ測定時の解析ポラリティの設定/問い合わせ	4-24
<b>COMMunicateグループ</b>		
:COMMunicate?	通信に関する全設定の問い合わせ	4-24
:COMMunicate:HEADer	応答データにヘッダを付ける/付けないの設定/問い合わせ	4-24
:COMMunicate:LOCKout	ローカルロックアウトの設定/解除	4-24
:COMMunicate:REMOte	リモート/ローカルの切り替え	4-25
:COMMunicate:VERBose	応答データの省略形/非省略形の設定/問い合わせ	4-25
:COMMunicate:WAIT	指定拡張イベント発生の待機	4-25
:COMMunicate:WAIT?	指定拡張イベント発生時の応答作成	4-25
<b>DISPlayグループ</b>		
:DISPlay?	表示に関する全設定の問い合わせ	4-27
:DISPlay:BGRaph	両パルス幅/両エッジ測定時でBOTH Graph表示のON/OFFの設定/問い合わせ	4-27
:DISPlay:BLOCK	表示するブロックの設定/問い合わせ	4-27
:DISPlay:DOTConnect	タイムバリエーション表示におけるドットコネクットのON/OFFの設定/問い合わせ	4-27
:DISPlay:DOTType	タイムバリエーション表示における測定点の表示方法の設定/問い合わせ	4-27
:DISPlay:GRAPhsize	グラフの表示の大きさの設定/問い合わせ	4-27
:DISPlay:GRID	タイムバリエーション表示におけるグリッド表示のON/OFFの設定/問い合わせ	4-27
:DISPlay:ITEM	表示形式の設定/問い合わせ	4-28
:DISPlay:OVERlap	重ね描きするか(ON), 否か(OFF)の設定/問い合わせ	4-28
:DISPlay:PANorama[:STATE]	パノラマ表示のON/OFFの設定/問い合わせ	4-28
:DISPlay:SGRaph	マルチウインドウ、オートウインドウのALL表示時の画面下部の表示の設定/問い合わせ	4-28
:DISPlay:SITem	統計値表示の表示する統計値のタイプの設定/問い合わせ	4-28
:DISPlay:SSTYle	統計値表示の表示形式に設定/問い合わせ	4-28
:DISPlay:STATistic	ヒストグラムまたはタイムバリエーション表示時の統計値表示のON/OFFの設定/問い合わせ	4-29
:DISPlay:WINDow	表示するウインドウの設定/問い合わせ	4-29

コマンド	機能	ページ
<b>FILEグループ</b>		
:FILE?	ファイルに関する全設定の問い合わせ	4-30
:FILE:CDIRectroy	カレントディレクトリの変更	4-30
:FILE:DELeTe: {BINary BMP POSTscript SETup STATistic TIFF TEXT}	各種ファイルの消去	4-31
:FILE:DRIVE	対象ドライブの設定	4-31
:FILE:FORMat	フロッピーディスクの初期化の実行	4-31
:FILE:FREE?	ドライブの空き容量をバイト数で問い合わせ	4-31
:FILE:LOAD: {BINary SETup}	各種ファイルの呼び出し	4-31
:FILE:MDIRectory	ディレクトリの作成	4-31
:FILE:PATH?	カレントディレクトリの問い合わせ	4-31
:FILE:SAVE?	ファイル保存に関する全設定の問い合わせ	4-31
:FILE:SAVE:ANAMing	保存するファイル名の自動生成のON/OFFの設定/問い合わせ	4-31
:FILE:SAVE: {BINary SETup STATistic TEXT}	各種ファイルの保存	4-32
:FILE:SAVE:COMMeNt	画面の左上部のコメントの設定/問い合わせ	4-32
:FILE:SAVE:SITem	保存する統計データファイルのタイプの設定/問い合わせ	4-32
<b>HCOPYグループ</b>		
:HCOPY?	画面イメージデータの出力に関する全設定の問い合わせ	4-33
:HCOPY:ABORt	画面イメージのプリントアウトの中止	4-33
:HCOPY:ANAMing	ファイルへの画面イメージの保存で、ファイル名の自動生成のON/OFFの設定/問い合わせ	4-33
:HCOPY:CDIRectory	カレントディレクトリの変更	4-34
:HCOPY:COMMeNt	画面左上部のコメントの設定/問い合わせ	4-34
:HCOPY:COMPreSSion	BMP形式で画面イメージを保存するときの圧縮のON/OFFの設定/問い合わせ	4-34
:HCOPY:DEVIce	画面イメージの出力先の設定/問い合わせ	4-34
:HCOPY:DRIVE	対象ドライブの設定	4-34
:HCOPY[:EXECute]	画面イメージのプリントアウトの実行	4-34
:HCOPY:FILEName	画面イメージの保存ファイル名の設定/問い合わせ	4-34
:HCOPY:FORMat	画面イメージの保存をするときのフォーマットの設定/問い合わせ	4-34
:HCOPY:TONE	画面イメージを保存するときの色/階調の設定/問い合わせ	4-34
<b>HHistogram, THISTogramグループ</b>		
{HHISTogram<x> THISTogram<x>}?	ヒストグラム表示に関する全設定の問い合わせ	4-36
{HHISTogram<x> THISTogram<x>}:AUTO?	オートウインドウに関する全設定の問い合わせ	4-36
{HHISTogram<x> THISTogram<x>}:AUTO:MODulation	オートウインドウでの変調方式の設定/問い合わせ	4-36
{HHISTogram<x> THISTogram<x>}:AUTO:TTYPE	オートウインドウでの定数Tの決め方の設定/問い合わせ	4-36
{HHISTogram<x> THISTogram<x>}:AUTO:WINDow1:STATe	オートウインドウでのウインドウ1のON/OFFの設定/問い合わせ	4-37
{HHISTogram<x> THISTogram<x>}:MARKer?	マーカーに関する全設定の問い合わせ	4-37
{HHISTogram<x> THISTogram<x>}:MARKer:LOW	ローマーカー値の設定/問い合わせ	4-37
{HHISTogram<x> THISTogram<x>}:MARKer[:STATe]	マーカーのON/OFFの設定/問い合わせ	4-37
{HHISTogram<x> THISTogram<x>}[:MODE]	ウインドウモードの設定/問い合わせ	4-37
{HHISTogram<x> THISTogram<x>}:MULTi?	マルチウインドウに関する全設定の問い合わせ	4-37
{HHISTogram<x> THISTogram<x>}:MULTi:FREQuency	周波数形式による定数Tの値の設定/問い合わせ	4-37

## 4.1 コマンド

コマンド	機能	ページ
:{HHIStogram<x>   THIStogram<x>}:MULTi:OFFSet	定数Tへのオフセット値の設定/問い合わせ	4-38
:{HHIStogram<x>   THIStogram<x>}:MULTi:SIZE	ウインドウ数の設定/問い合わせ	4-38
:{HHIStogram<x>   THIStogram<x>}:MULTi:TVALue	定数Tの値の設定/問い合わせ	4-38
:{HHIStogram<x>   THIStogram<x>}:MULTi:UPDate	定数T, オフセット値をもとにウインドウの設定を変更	4-38
:{HHIStogram<x>   THIStogram<x>}:MULTi:WINDow<x>?	各ウインドウに関する全設定の問い合わせ	4-38
:{HHIStogram<x>   THIStogram<x>}:MULTi:WINDow<x>:HORizontal?	各ウインドウの水平軸(X軸)に関する全設定の問い合わせ	4-38
:{HHIStogram<x>   THIStogram<x>}:MULTi:WINDow<x>:HORizontal:CENTer	各ウインドウの水平軸(X軸)の中央位置の設定/問い合わせ	4-39
:{HHIStogram<x>   THIStogram<x>}:MULTi:WINDow<x>:HORizontal:SPAN	各ウインドウの水平軸(X軸)の幅の設定/問い合わせ	4-39
:{HHIStogram<x>   THIStogram<x>}:MULTi:WINDow<x>:LABel	各ウインドウのラベルの設定/問い合わせ	4-39
:{HHIStogram<x>   THIStogram<x>}:MULTi:WINDow<x>:MARKer?	各ウインドウの水平軸(X軸)マーカーに関する全設定の問い合わせ	4-39
:{HHIStogram<x>   THIStogram<x>}:MULTi:WINDow<x>:MARKer:{LEFT RIGHT}	各ウインドウの各マーカーの位置の設定/問い合わせ	4-40
:{HHIStogram<x>   THIStogram<x>}:MULTi:WINDow<x>:MARKer:{LVALue? RVALue?}	各ウインドウの各マーカーの位置の度数の設定/問い合わせ	4-40
:{HHIStogram<x>   THIStogram<x>}:SINGLE?	シングルウインドウに関する全設定の問い合わせ	4-40
:{HHIStogram<x>   THIStogram<x>}:SINGLE:FREQuency	周波数形式による定数Tの値の設定/問い合わせ	4-40
:{HHIStogram<x>   THIStogram<x>}:SINGLE:HORizontal?	シングルウインドウの水平軸(X軸)に関する全設定の問い合わせ	4-40
:{HHIStogram<x>   THIStogram<x>}:SINGLE:HORizontal:CENTer	シングルウインドウの水平軸(X軸)の中央位置の設定/問い合わせ	4-41
:{HHIStogram<x>   THIStogram<x>}:SINGLE:HORizontal:SPAN	シングルウインドウの水平軸(X軸)の幅の設定/問い合わせ	4-41
:{HHIStogram<x>   THIStogram<x>}:SINGLE:MARKer?	シングルウインドウのマーカーに関する全設定の問い合わせ	4-41
:{HHIStogram<x>   THIStogram<x>}:SINGLE:MARKer:{LEFT RIGHT}	シングルウインドウの各マーカーの位置の設定/問い合わせ	4-41
:{HHIStogram<x>   THIStogram<x>}:SINGLE:MARKer:{LVALue? RVALue?}	シングルウインドウの各マーカーの位置の度数の問い合わせ	4-41
:{HHIStogram<x>   THIStogram<x>}:SINGLE:TVALue	シングルウインドウの定数Tの値の設定/問い合わせ	4-42
:{HHIStogram<x>   THIStogram<x>}:VERTical?	垂直軸(Y軸)に関する全設定の問い合わせ	4-42
:{HHIStogram<x>   THIStogram<x>}:VERTical:AXIS	垂直軸(Y軸)スケールの種類の設定の問い合わせ	4-42
:{HHIStogram<x>   THIStogram<x>}:VERTical:HIGH	垂直軸(Y軸)スケールの上限の設定の問い合わせ	4-42
<b>IHIStogramグループ</b>		
:IHIStogram?	ISIモードのヒストグラム表示に関する全設定の問い合わせ	4-45
:IHIStogram:AUTO?	オートウインドウに関する全設定の問い合わせ	4-45
:IHIStogram:AUTO:MODulation	オートウインドウでの変調方式の設定/問い合わせ	4-45
:IHIStogram:AUTO:TTYPe	オートウインドウでの定数Tの決め方の設定/問い合わせ	4-45
:IHIStogram:AUTO:WINDow1:STATe	オートウインドウでのウインドウ 1 のON/OFFの設定/問い合わせ	4-45
:IHIStogram:{MARK<x>   SPACe<x>}	マーク/スペースの設定/問い合わせ	4-46

コマンド	機能	ページ
:IHIStogram[:MODE]	ウインドウモードの設定/問い合わせ	4-46
:IHIStogram:MULTi?	マルチウインドウに関する全設定の問い合わせ	4-46
:IHIStogram:MULTi:FREQuency	周波数形式による定数Tの値の設定/問い合わせ	4-46
:IHIStogram:MULTi:OFFSet	定数Tへのオフセット値の設定/問い合わせ	4-47
:IHIStogram:MULTi:SIZE	ウインドウ数の設定/問い合わせ	4-47
:IHIStogram:MULTi:TVALue	定数Tの値の設定/問い合わせ	4-47
:IHIStogram:MULTi:UPDate	定数T, オフセット値をもとにウインドウ設定を変更	4-47
:IHIStogram:MULTi:WINDow<x>?	各ウインドウに関する全設定の問い合わせ	4-47
:IHIStogram:MULTi:WINDow<x>:HORizontal?	各ウインドウの水平軸(X軸)に関する全設定の問い合わせ	4-47
:IHIStogram:MULTi:WINDow<x>:HORizontal:CENTer	各ウインドウの水平軸(X軸)の中央位置の設定/問い合わせ	4-47
:IHIStogram:MULTi:WINDow<x>:HORizontal:SPAN	ウインドウの水平軸(X軸)の幅の設定/問い合わせ	4-47
:IHIStogram:MULTi:WINDow<x>:LABel	ウインドウラベルの設定/問い合わせ	4-48
:IHIStogram:MULTi:WINDow<x>:MARKer?	各ウインドウの水平軸(X軸)マーカーに関する全設定の問い合わせ	4-48
:IHIStogram:MULTi:WINDow<x>:MARKer:{LEFT RIGHT}	各ウインドウの各マーカーの位置の設定/問い合わせ	4-48
:IHIStogram:POLarity	ポラリティの設定/問い合わせ	4-48
:IHIStogram:SYnc	Sync機能のON/OFFの設定/問い合わせ	4-48
:IHIStogram:TARGet	解析対象の設定/問い合わせ	4-48
:IHIStogram:TMOde	トリガモードの設定/問い合わせ	4-49
:IHIStogram:TRIGger	トリガ条件の設定/問い合わせ	4-49
<b>INPutグループ</b>		
:INPut?	入力に関する全設定の問い合わせ	4-50
:INPut:{ACHannel BCHannel}?	各チャンネルに関する全設定の問い合わせ	4-50
:INPut:{ACHannel BCHannel}:COUPling	各チャンネルのカップリングの設定/問い合わせ	4-51
:INPut:{ACHannel BCHannel}:IMPedance	各チャンネルの入力インピーダンスの設定/問い合わせ	4-51
:INPut:{ACHannel BCHannel}:TRIGger?	各チャンネルのトリガに関する全設定の問い合わせ	4-51
:INPut:{ACHannel BCHannel}:TRIGger:LEVel	各チャンネルのトリガレベルの設定/問い合わせ	4-51
:INPut:{ACHannel BCHannel}:TRIGger:MOde	各チャンネルのトリガモードの設定/問い合わせ	4-51
:INPut:AGATe?	アーミング/外部ゲートの全設定の問い合わせ	4-51
:INPut:AGATe:LEVel	アーミング/ゲートのレベルの設定/問い合わせ	4-51
:INPut:BCHannel:PHASe?	CH Bの位相調整に関する全設定の問い合わせ	4-51
:INPut:BCHannel:PHASe:ADJust	CH Bの位相調整時間の設定/問い合わせ	4-51
:INPut:INHibit?	インヒビットに関する全設定の問い合わせ	4-52
:INPut:INHibit:LEVel	インヒビットレベルの設定/問い合わせ	4-52
<b>MEASureグループ</b>		
:MEASure?	測定条件の全設定の問い合わせ	4-53
:MEASure:FUNcTion	測定ファクションの設定/問い合わせ	4-53
:MEASure:MOde	測定モードの設定/問い合わせ	4-54
:MEASure:POLarity	パルス幅のポラリティの設定/問い合わせ	4-54
:MEASure:SLOPe	周期/AtoBタイムインターバルのスロープの設定/問い合わせ	4-54

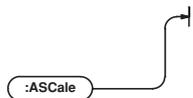
## 4.1 コマンド

コマンド	機能	ページ
<b>MEMoryグループ</b>		
:MEMory?	測定データの外部への送信に関する全設定の問い合わせ	4-55
:MEMory:BLOCK	ブロックサンプリング時の対象ブロックの設定/問い合わせ	4-55
:MEMory:BSIZE?	測定が有効なブロック数の問い合わせ	4-55
:MEMory:BYTeorder	バイナリデータの送信順序の設定/問い合わせ	4-55
:MEMory:CLear	測定データのクリア	4-55
:MEMory:DATaselect	送信対象となるデータの設定/問い合わせ	4-56
:MEMory:END	送信終了のデータ位置の設定/問い合わせ	4-56
:MEMory:FORMat	送信するデータの形式の設定/問い合わせ	4-56
:MEMory:SEND<x>?	「:MEMory:DATaselect」で設定した測定データの送信の実行	4-56
:MEMory:SIZE<x>?	測定したデータ数の問い合わせ	4-56
:MEMory:STArt	送信開始のデータ位置の設定/問い合わせ	4-57
<b>RECallグループ</b>		
:RECall	設定情報のリコール	4-57
<b>SAMpleグループ</b>		
:SAMple?	サンプリングに関する全設定の問い合わせ	4-59
:SAMple:ARMinG?	アーミングに関する全設定の問い合わせ	4-59
:SAMple:ARMinG:DELAy?	アーミングディレイに関する全設定の問い合わせ	4-59
:SAMple:ARMinG:DELAy:{AEVentsize BEVentsize EVENTsize}	各チャンネルのアーミングディレイイベントの設定/問い合わせ	4-60
:SAMple:ARMinG:DELAy:{ATIME BTIME TIME}	各チャンネルのアーミングディレイ時間の設定/問い合わせ	4-60
:SAMple:ARMinG:DELAy[:MODE]	アーミングディレイモードの設定/問い合わせ	4-60
:SAMple:ARMinG:SLOPe	アーミングスロープの設定/問い合わせ	4-60
:SAMple:ARMinG:SOURCe	アーミングソースの設定/問い合わせ	4-60
:SAMple:BLOCK?	ブロックサンプリングに関する全設定の問い合わせ	4-60
:SAMple:BLOCK:REST?	ブロックサンプリングの休止に関する全設定の問い合わせ	4-60
:SAMple:BLOCK:REST:EVENT	ブロックサンプリングの休止時間をイベント数で設定/問い合わせ	4-61
:SAMple:BLOCK:REST[:MODE]	ブロックサンプリングの休止モードの設定/問い合わせ	4-61
:SAMple:BLOCK:REST:TIME	ブロックサンプリングの休止時間の設定/問い合わせ	4-61
:SAMple:BLOCK:SIZE	ブロックサンプリングのブロック数の設定/問い合わせ	4-61
:SAMple:BLOCK[:STATe]	ブロックサンプリングのON/OFFの設定/問い合わせ	4-61
:SAMple:GATE?	ゲートに関する全設定の問い合わせ	4-61
:SAMple:GATE:EVENTsize	ゲートをイベント数で設定/問い合わせ	4-61
:SAMple:GATE[:MODE]	ゲートの種類の設定/問い合わせ	4-61
:SAMple:GATE:POLarity	外部ゲートの極性の設定/問い合わせ	4-62
:SAMple:GATE:TIME	ゲート時間を時間で設定/問い合わせ	4-62
:SAMple:INHibit	インヒビット入力の極性の設定/問い合わせ	4-62
:SAMple:INTerval	サンプリングインターバルの設定/問い合わせ	4-62
:SAMple:RCLock	サンプリングの基準クロックの設定/問い合わせ	4-62
<b>SStArtグループ</b>		
:SStArt	シングル測定の実行	4-62
<b>StArtグループ</b>		
:StArt	測定のスタート	4-62
<b>STATusグループ</b>		
:STATus?	通信のステータスに関する全設定の問い合わせ	4-63
:STATus:CONDition?	状態レジスタの内容の問い合わせ	4-63
:STATus:EESSE	拡張イベントイネーブルレジスタの設定/問い合わせ	4-63
:STATus:EESR?	拡張イベントレジスタの内容の問い合わせとレジスタクリア	4-63
:STATus:ERRor?	発生エラーコードとメッセージの内容の問い合わせ	4-63
:STATus:FILTer<x>	遷移フィルタの設定/問い合わせ	4-63
:STATus:QMESSage	「:STATus:ERRor?」の応答にメッセージの内容を付ける付けないの設定/問い合わせ	4-63

コマンド	機能	ページ
<b>STOPグループ</b>		
:STOP	測定のストップ	4-64
<b>STOReグループ</b>		
:STORe	設定情報のストア	4-64
<b>SYSTemグループ</b>		
:SYSTem?	SYSTemグループの全設定の問い合わせ	4-65
:SYSTem:BEEP	ピープ音のON/OFFの設定/問い合わせ	4-65
:SYSTem:CLICkSound	クリック音のON/OFFの設定/問い合わせ	4-66
:SYSTem:DATE	日付の設定/問い合わせ	4-66
:SYSTem:LCD?	液晶画面に関する全設定の問い合わせ	4-66
:SYSTem:LCD:AOFF	液晶画面のオートオフのON/OFFの設定/問い合わせ	4-66
:SYSTem:LCD:ATIME	液晶画面のオートオフの時間の設定/問い合わせ	4-66
:SYSTem:LCD:BRIGhtness	液晶画面の明るさの設定/問い合わせ	4-66
:SYSTem:LCD[:STATE]	液晶画面のバックライトのON/OFFの設定/問い合わせ	4-66
:SYSTem:TIME	時刻の設定/問い合わせ	4-66
:SYSTem:WARning	ワーニング表示のON/OFFの設定/問い合わせ	4-66
<b>TVARiationグループ</b>		
:TVARiation<x>?	タイムバリエーションに関する全設定の問い合わせ	4-67
:TVARiation<x>:HORizonatal?	タイムバリエーション表示の水平軸(X軸)に関する全設定の問い合わせ	4-67
:TVARiation<x>:HORizonatal:MINimum	水平軸(X軸)スケールの左端の設定/問い合わせ	4-68
:TVARiation<x>:HORizonatal:SPAN	水平軸(X軸)スケールの幅の設定/問い合わせ	4-68
:TVARiation<x>:MARKer?	マーカーに関する全設定の問い合わせ	4-68
:TVARiation<x>:MARKer:{HIGH LOW}	ハイ/ローマーカーの位置の設定/問い合わせ	4-68
:TVARiation<x>:MARKer:{LEFT RIGHT}	レフト/ライトマーカーの位置の設定/問い合わせ	4-68
:TVARiation<x>:MARKer:{LVALue? RVALue?}	レフト/ライトマーカー位置の測定値の問い合わせ	4-68
:TVARiation<x>:MARKer[:STATE]	マーカーカーソルのON/OFFの設定/問い合わせ	4-68
:TVARiation<x>:VERTical?	垂直軸(Y軸)に関する全設定の問い合わせ	4-68
:TVARiation<x>:VERTical:CENTer	垂直軸(Y軸)の中央値の設定/問い合わせ	4-68
:TVARiation<x>:VERTical:SPAN	垂直軸(Y軸)の幅の設定/問い合わせ	4-69
<b>UNITグループ</b>		
:UNIT?	電圧/時間/周波数のデフォルトの単位の問い合わせ	4-69
:UNIT:VOLTage	電圧のデフォルトの単位の設定/問い合わせ	4-69
:UNIT:TIME	時間のデフォルトの単位の設定/問い合わせ	4-69
:UNIT:FREQuency	周波数のデフォルトの単位の設定/問い合わせ	4-69
<b>共通コマンドグループ</b>		
*CAL?	キャリブレーションを実行し、結果を問い合わせ	4-70
*CLS	標準イベントレジスタ、拡張イベントレジスタ、エラーキューのクリア	4-70
*ESE	標準イベントイネーブルレジスタ値の設定/問い合わせ	4-70
*ESR?	標準イベントレジスタ値の問い合わせとレジスタのクリア	4-70
*IDN?	機種名の問い合わせ	4-70
*OPC	指定オーバーラップ終了後、標準イベントレジスタのビット0(OPCビット)を1にセット	4-70
*OPC?	指定オーバーラップ終了後、応答作成	4-70
*RST	設定の初期化	4-71
*SRE	サービスリクエストイネーブルレジスタ値の設定/問い合わせ	4-71
*STB?	ステータスバイトレジスタの問い合わせ	4-71
*TRG?	シングル測定の実行	4-71
*TST?	セルフテストの実行と結果の問い合わせ	4-71
*WAI	指定オーバーラップ終了まで*WAIに続く命令を待機	4-71

## 4.2 ASCaleグループ

オートスケールに関するグループです。フロントパネルのAUTO SCALEキーと同じ内容を実行できます。

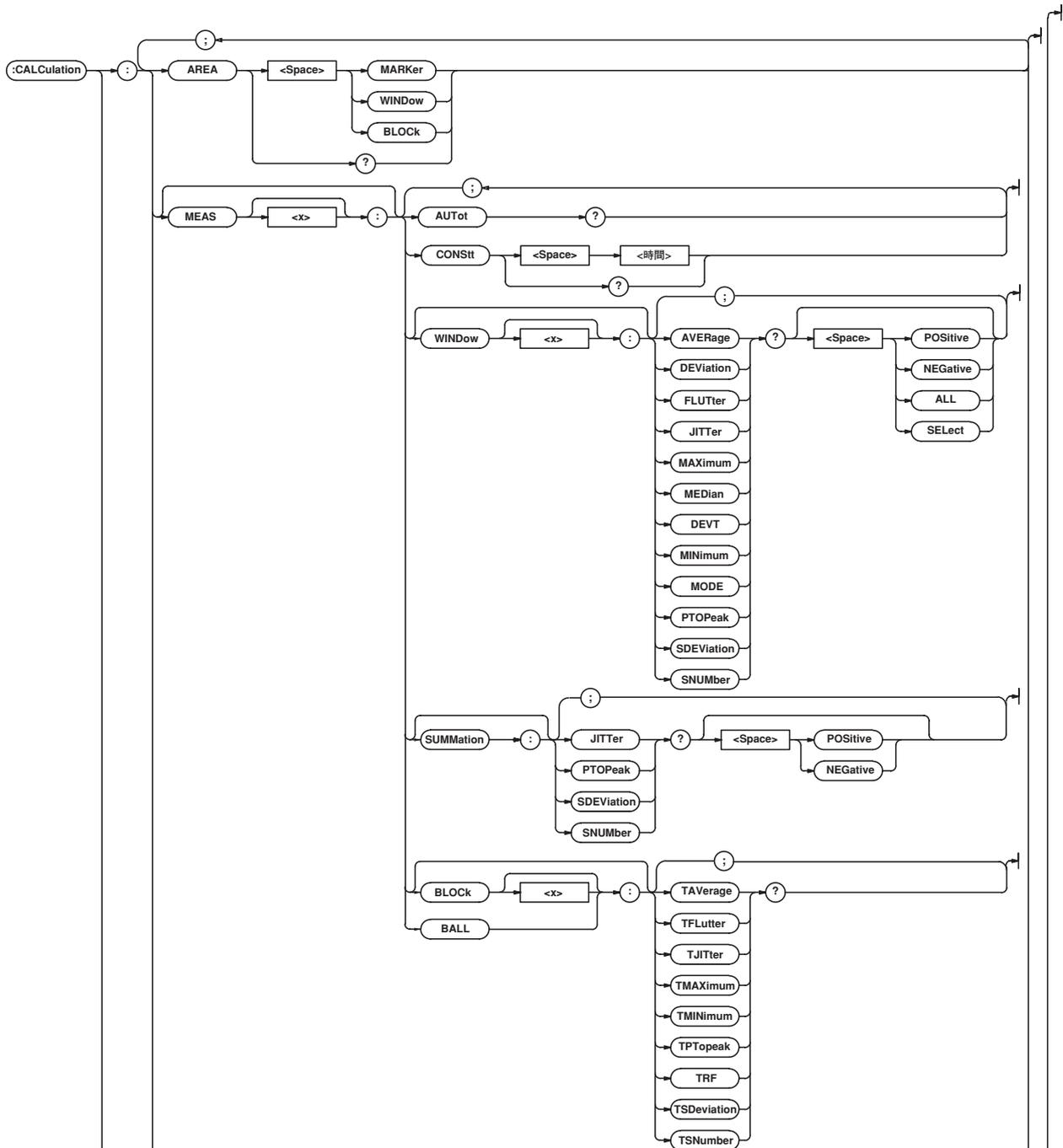


### **:ASCale**

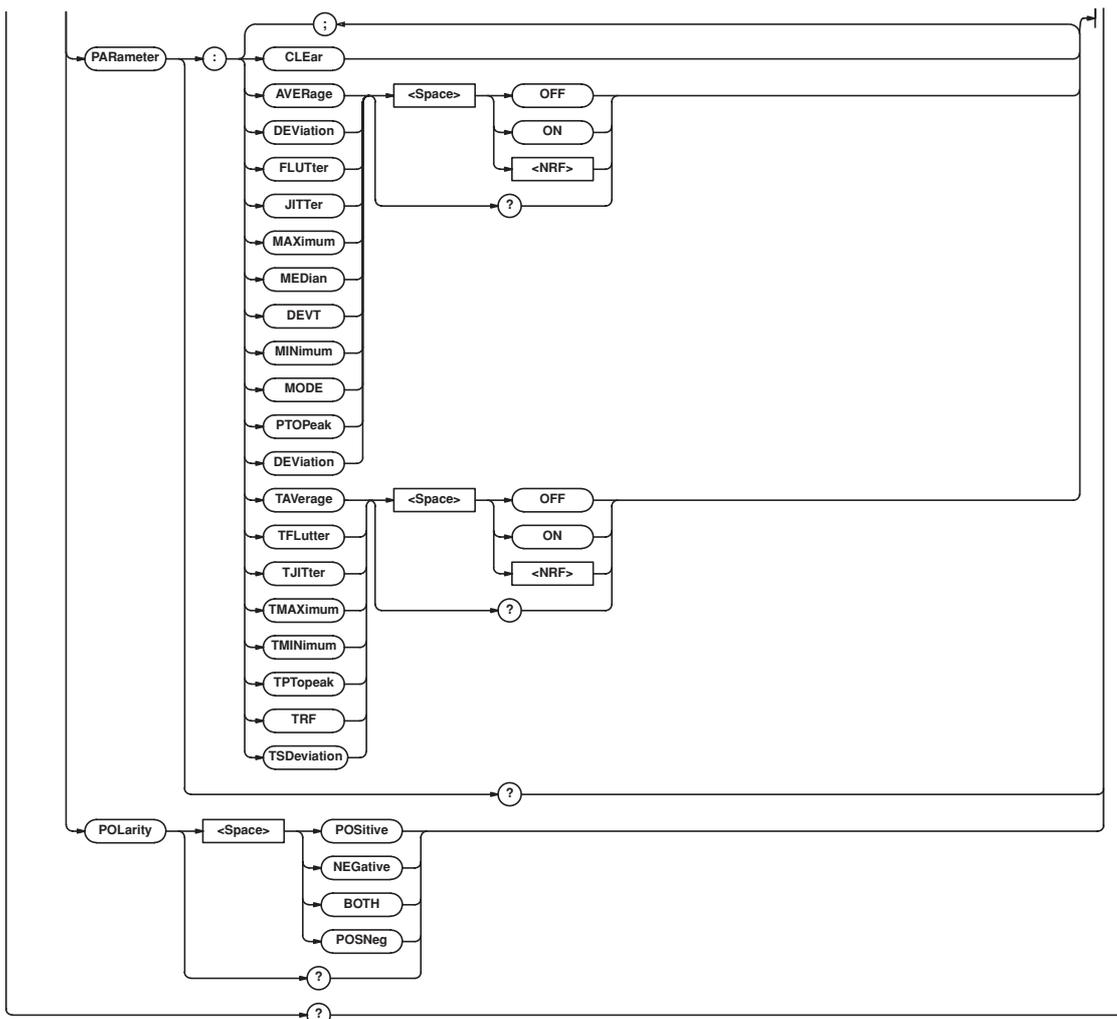
機能	オートスケールを実行します。
構文	<b>:ASCale</b>
例	<b>:ASCALe</b>
解説	マルチウインドウ/オートウインドウのとき、 オートスケールは実行できません。

### 4.3 CALCulationグループ

統計演算に関するグループです。フロントパネルのDISPLAYキーのCalc/Statメニューと同じ設定、および設定内容の問い合わせと統計値の問い合わせができます。



### 4.3 CALCulationグループ



#### :CALCulation

機能 統計値に関する全設定を問い合わせます。  
 構文 :CALCulation?  
 例 :CALCULATION? -> :CALCULATION:  
 AREA BLOCK;PARAMETER:AVERAGE 1;  
 DEVIATION 0;FLUTTER 0;JITTER 0;  
 MAXIMUM 0;MEDIAN 0;DEVT 0;  
 MINIMUM 0;MODE 0;PTOPEAK 1;  
 SDEVIATION 1;TAVERAGE 1;TFLUTTER 0;  
 TJITTER 0;TMAXIMUM 0;TMINIMUM 0;  
 TPTOPEAK 1;TRF 1;TSDEVIATION 1

#### :CALCulation:AREA

機能 統計値の演算範囲を設定/問い合わせします。  
 構文 :CALCulation:AREA  
 {MARKer|WINDow|BLOCK}  
 :CALCulation:AREA?  
 例 :CALCULATION:AREA MARKER  
 :CALCULATION:AREA? -> :CALCULATION:  
 AREA MARKER  
 解説
 

- ・マルチウインドウ/オートウインドウのときは、設定できません。
- ・ヒストグラムの場合は、BLOCKは設定できません。
- ・周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定の場合、BLOCKは設定できません。

**:CALCulation[:MEAS<x>]:AUTot?**

機能 オートウインドウの測定T値(Measured T)/推定T値(Estimated T)を問い合わせます。

構文 **:CALCulation[:MEAS<x>]:AUTot?**  
<x>=1~2

例 ・ 周期/AtoBタイムインターバル/パルス幅測定  
のとき

**:CALCULATION:AUTOT?**  
-> **:CALCULATION:AUTOT 3.6750E-8**

・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定  
のとき

**:CALCULATION:MEAS1:AUTOT?**  
-> **:CALCULATION:MEAS1:**  
**AUTOT 3.6750E-8**

解説 ・ 周期/AtoBタイムインターバル/パルス幅測定  
のときは、「**:MEAS<x>**」を省略してください。  
・ オートウインドウのとき、問い合わせができません。

**:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}]:TAverage?**

機能 タイムバリエーション時の平均値を問い合わせます。

構文 **:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}]:TAverage?**  
MEAS<x>の<x>=1~2  
BLOCK<x>の<x>=ブロック番号(1~ブロック数)  
BALL:ブロック全体の統計値を問い合わせます。

例 ・ 周期/AtoBタイムインターバル/パルス幅測定、  
ブロックサンプリングではないとき

**:CALCULATION:TAVERAGE?**  
-> **:CALCULATION:**  
**TAVERAGE 2.4035E-9**

・ 周期/AtoBタイムインターバル/パルス幅測定、  
ブロックサンプリングのとき

**:CALCULATION:BLOCK1:TAVERAGE?**  
-> **:CALCULATION:BLOCK1:**  
**TAVERAGE 2.4035E-9**

・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定  
のとき

**:CALCULATION:MEAS1:TAVERAGE?**  
-> **:CALCULATION:MEAS1:**  
**TAVERAGE 2.4035E-9**

解説 ・ 統計値が有効でない場合は、問い合わせに対して「**NAN**」を返します。  
・ ブロックサンプリングで、演算範囲がブロックのとき「**:{BLOCK<x>|BALL}**」を指定してください。  
・ 周期/AtoBタイムインターバル/パルス幅測定  
のときは、「**:MEAS<x>**」を省略してください。

**:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}]:TFLutter?**

機能 タイムバリエーション時のσ/AVE値(フラッタ)を問い合わせます。

構文 **:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}]:TFLutter?**  
MEAS<x>の<x>=1~2  
BLOCK<x>の<x>=ブロック番号(1~ブロック数)  
BALL:ブロック全体の統計値を問い合わせます。

例 ・ 周期/AtoBタイムインターバル/パルス幅測定、  
ブロックサンプリングではないとき

**:CALCULATION:TFLUTTER?**  
-> **:CALCULATION:**  
**TFLUTTER 1.2887E+1**

・ 周期/AtoBタイムインターバル/パルス幅測定、  
ブロックサンプリングのとき

**:CALCULATION:BLOCK1:TFLUTTER?**  
-> **:CALCULATION:BLOCK1:**  
**TFLUTTER 1.2887E+1**

・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定  
のとき

**:CALCULATION:MEAS1:TFLUTTER?**  
-> **:CALCULATION:MEAS1:**  
**TFLUTTER 1.2887E+1**

解説 ・ 統計値が有効でない場合は、問い合わせに対して「**NAN**」を返します。

・ ブロックサンプリングで、演算範囲がブロックのとき「**:{BLOCK<x>|BALL}**」を指定してください。

・ 周期/AtoBタイムインターバル/パルス幅測定  
のときは、「**:MEAS<x>**」を省略してください。

### 4.3 CALCulationグループ

#### **:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}] :TJITter?**

機能 タイムバリエーション時のP-P/AVE値(ジッタ)を問い合わせます。

構文 :CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}] :TJITter?  
MEAS<x>の<x>=1~2  
BLOCK<x>の<x>=ブロック番号(1~ブロック数)  
BALL : ブロック全体の統計値を問い合わせます。

例  
・ 周期/AtoBタイムインターバル/パルス幅測定, ブロックサンプリングではないとき  
:CALCULATION:TJITTER?  
-> :CALCULATION:TJITTER 1.382E+00  
・ 周期/AtoBタイムインターバル/パルス幅測定, ブロックサンプリングのとき  
:CALCULATION:BLOCK1:TJITTER?  
-> :CALCULATION:BLOCK1:TJITTER 1.382E+00  
・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定のと  
:CALCULATION:MEAS1:TJITTER?  
-> :CALCULATION:MEAS1:TJITTER 1.382E+00

解説  
・ 統計値が有効でない場合は, 問い合わせに対して「NAN」を返します。  
・ ブロックサンプリングで, 演算範囲がブロックのとき「:{BLOCK<x>|BALL}」を指定してください。  
・ 周期/AtoBタイムインターバル/パルス幅測定のと  
ときは, 「:MEAS<x>」を省略してください。

#### **:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}] :TMAXimum?**

機能 タイムバリエーション時の最大値を問い合わせます。

構文 :CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}] :TMAXimum?  
MEAS<x>の<x>=1~2  
BLOCK<x>の<x>=ブロック番号(1~ブロック数)  
BALL : ブロック全体の統計値を問い合わせます。

例  
・ 周期/AtoBタイムインターバル/パルス幅測定, ブロックサンプリングではないとき  
:CALCULATION:TMAXIMUM?  
-> :CALCULATION:TMAXIMUM 1.1287E-7  
・ 周期/AtoBタイムインターバル/パルス幅測定, ブロックサンプリングのとき  
:CALCULATION:BLOCK1:TMAXIMUM?  
-> :CALCULATION:BLOCK1:TMAXIMUM 1.1287E-7  
・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定のと  
:CALCULATION:MEAS1:TMAXIMUM?  
-> :CALCULATION:MEAS1:TMAXIMUM 1.1287E-7

解説  
・ 統計値が有効でない場合は, 問い合わせに対して「NAN」を返します。  
・ ブロックサンプリングで, 演算範囲がブロックのとき「:{BLOCK<x>|BALL}」を指定してください。  
・ 周期/AtoBタイムインターバル/パルス幅測定のと  
ときは, 「:MEAS<x>」を省略してください。

**:CALCulation[:MEAS<x>][[:{BLOCK<x>|BALL}]:TMINimum?**

機能 タイムバリエーション時の最小値を問い合わせます。

構文 **:CALCulation[:MEAS<x>][[:{BLOCK<x>|BALL}]:TMINimum?**  
 MEAS<x>の<x>=1~2  
 BLOCK<x>の<x>=ブロック番号(1~ブロック数)  
 BALL:ブロック全体の統計値を問い合わせます。

例 ・ 周期/AtoBタイムインターバル/パルス幅測定、ブロックサンプリングではないとき

**:CALCULATION:TMINIMUM?**

-> **:CALCULATION:**

**TMINIMUM 9.99437E-8**

・ 周期/AtoBタイムインターバル/パルス幅測定、ブロックサンプリングのとき

**:CALCULATION:BLOCK1:TMINIMUM?**

-> **:CALCULATION:BLOCK1:**

**TMINIMUM 9.99437E-8**

・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定するとき

**:CALCULATION:MEAS1:TMINIMUM?**

-> **:CALCULATION:MEAS1:**

**TMINIMUM 9.99437E-8**

解説 ・ 統計値が有効でない場合は、問い合わせに対して「NAN」を返します。

・ ブロックサンプリングで、演算範囲がブロックのとき「:{BLOCK<x>|BALL}」を指定してください。

・ 周期/AtoBタイムインターバル/パルス幅測定の場合は、「:MEAS<x>」を省略してください。

**:CALCulation[:MEAS<x>][[:{BLOCK<x>|BALL}]:TPTopeak?**

機能 タイムバリエーション時のP-P値を問い合わせます。

構文 **:CALCulation[:MEAS<x>][[:{BLOCK<x>|BALL}]:TPTopeak?**  
 MEAS<x>の<x>=1~2  
 BLOCK<x>の<x>=ブロック番号(1~ブロック数)  
 BALL:ブロック全体の統計値を問い合わせます。

例 ・ 周期/AtoBタイムインターバル/パルス幅測定、ブロックサンプリングではないとき

**:CALCULATION:TPTOPEAK?**

-> **:CALCULATION:**

**TPTOPEAK 3.0245E-9**

・ 周期/AtoBタイムインターバル/パルス幅測定、ブロックサンプリングのとき

**:CALCULATION:BLOCK1:TPTOPEAK?**

-> **:CALCULATION:BLOCK1:**

**TPTOPEAK 3.0245E-9**

・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定するとき

**:CALCULATION:MEAS1:TPTOPEAK?**

-> **:CALCULATION:MEAS1:**

**TPTOPEAK 3.0245E-9**

解説 ・ 統計値が有効でない場合は、問い合わせに対して「NAN」を返します。

・ ブロックサンプリングで、演算範囲がブロックのとき「:{BLOCK<x>|BALL}」を指定してください。

・ 周期/AtoBタイムインターバル/パルス幅測定の場合は、「:MEAS<x>」を省略してください。

### 4.3 CALCulationグループ

#### **:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}]:TRF?**

- 機能** タイムバリエーション時のRF値を問い合わせます。
- 構文** :CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}]:TRF?  
MEAS<x>の<x>=1~2  
BLOCK<x>の<x>=ブロック番号(1~ブロック数)  
BALL:ブロック全体の統計値を問い合わせます。
- 例**
- ・ 周期/AtoBタイムインターバル/パルス幅測定, ブロックサンプリングではないとき  
:CALCULATION:TRF?  
-> :CALCULATION:TRF 1.4775E-1
  - ・ 周期/AtoBタイムインターバル/パルス幅測定, ブロックサンプリングのとき  
:CALCULATION:BLOCK1:TRF?  
-> :CALCULATION:BLOCK1:TRF 1.4775E-1
  - ・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定のとき  
:CALCULATION:MEAS1:TRF?  
-> :CALCULATION:MEAS1:TRF 1.4775E-1
- 解説**
- ・ 統計値が有効でない場合は, 問い合わせに対して「NAN」を返します。
  - ・ ブロックサンプリングで, 演算範囲がブロックのとき「:{BLOCK<x>|BALL}」を指定してください。
  - ・ 周期/AtoBタイムインターバル/パルス幅測定の場合は, 「:MEAS<x>」を省略してください。

#### **:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}]:TSDeviation?**

- 機能** タイムバリエーション時の標準偏差( $\sigma$ )を問い合わせます。
- 構文** :CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}]:TSDeviation?  
MEAS<x>の<x>=1~2  
BLOCK<x>の<x>=ブロック番号(1~ブロック数)  
BALL:ブロック全体の統計値を問い合わせます。
- 例**
- ・ 周期/AtoBタイムインターバル/パルス幅測定, ブロックサンプリングではないとき  
:CALCULATION:TSDEVIATION?  
-> :CALCULATION:TSDEVIATION 2.4035E-9
  - ・ 周期/AtoBタイムインターバル/パルス幅測定, ブロックサンプリングのとき  
:CALCULATION:BLOCK1:TSDEVIATION?  
-> :CALCULATION:BLOCK1:TSDEVIATION 2.4035E-9
  - ・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定のとき  
:CALCULATION:MEAS1:TSDEVIATION?  
-> :CALCULATION:MEAS1:TSDEVIATION 2.4035E-9
- 解説**
- ・ 統計値が有効でない場合は, 問い合わせに対して「NAN」を返します。
  - ・ ブロックサンプリングで, 演算範囲がブロックのとき「:{BLOCK<x>|BALL}」を指定してください。
  - ・ 周期/AtoBタイムインターバル/パルス幅測定の場合は, 「:MEAS<x>」を省略してください。

**:CALCulation[:MEAS<x>][[:{BLOCK<x>|BALL}]:TSNumber?**

機能 タイムバリエーション時の統計演算対象のサンプル数を問い合わせます。

構文 **:CALCulation[:MEAS<x>][[:{BLOCK<x>|BALL}]:TSNumber?**  
 MEAS<x>の<x>=1~2  
 BLOCK<x>の<x>=ブロック番号(1~ブロック数)  
 BALL:ブロック全体の統計値を問い合わせます。

例

- ・ 周期/AtoBタイムインターバル/パルス幅測定、ブロックサンプリングではないとき  
**:CALCULATION:TSNUMBER?**  
 -> **:CALCULATION:TSNUMBER 1.000E+3**
- ・ 周期/AtoBタイムインターバル/パルス幅測定、ブロックサンプリングのとき  
**:CALCULATION:BLOCK1:TSNUMBER?**  
 -> **:CALCULATION:BLOCK1:TSNUMBER 1.000E+3**
- ・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定のとき  
**:CALCULATION:MEAS1:TSNUMBER?**  
 -> **:CALCULATION:MEAS1:TSNUMBER 1.000E+3**

解説

- ・ 統計値が有効でない場合は、問い合わせに対して「NAN」を返します。
- ・ ブロックサンプリングで、演算範囲がブロックのとき「[:{BLOCK<x>|BALL}]」を指定してください。
- ・ 周期/AtoBタイムインターバル/パルス幅測定の場合は、「:MEAS<x>」を省略してください。

**:CALCulation[:MEAS<x>]:CONSTt**

機能 定数Tを設定/問い合わせします。

構文 **:CALCulation[:MEAS<x>]:CONSTt {<時間>}**

**:CALCulation[:MEAS<x>]:CONSTt?**

<x>=1~2

<時間>=1ns~250ns(25psステップ)

例

- ・ 周期/AtoBタイムインターバル/パルス幅測定  
 のとき

**:CALCULATION:CONSTT 100ns**

**:CALCULATION:CONSTT?**

-> **:CALCULATION:**

**CONSTT 100.000E-9**

- ・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定  
 のとき

**:CALCULATION:MEAS1:CONSTT 100ns**

**:CALCULATION:MEAS1:CONSTT?**

-> **:CALCULATION:MEAS1:**

**CONSTT 100.000E-9**

解説

- ・ シングルウィンドウ、マルチウィンドウのとき、設定/問い合わせができます。

- ・ 周期/AtoBタイムインターバル/パルス幅測定  
 のときは、「:MEAS<x>」を省略してください。

### 4.3 CALCulationグループ

**:CALCulation[:MEAS<x>][:WINDOW<x>]:**

**AVERage?**

機能 ヒストグラム時の平均値を問い合わせます。  
 構文 :CALCulation[:MEAS<x>][:WINDOW<x>]:  
 AVERage? [{POSitive|NEGative|ALL|  
 SElect}]  
 MEAS<x>の<x>=1~2  
 WINDOW<x>の<x>=ウインドウ番号(1~ウイン  
 ドウ数)

例 ●タイムスタンプ/ハードウェアヒストグラム  
 モードの場合  
 ・周期/AtoBタイムインターバル/パルス幅測  
 定, シングルウインドウのとき  
 :CALCULATION:AVERAGE?  
 -> :CALCULATION:  
 AVERAGE 1.00035E-7  
 ・周期/AtoBタイムインターバル/パルス幅測  
 定, マルチウインドウ/オートウインドウの  
 とき  
 :CALCULATION:WINDOW1:AVERAGE?  
 -> :CALCULATION:WINDOW1:  
 AVERAGE 1.00035E-7  
 ・周期A&周期B/周期A&AtoBタイムインター  
 バル/パルス幅A&AtoBタイムインターバル/パ  
 ルス幅A&パルス幅B測定, シングルウイン  
 ドウのとき  
 :CALCULATION:MEAS1:AVERAGE?  
 -> :CALCULATION:MEAS1:  
 AVERAGE 1.00035E-7  
 ・周期A&周期B/周期A&AtoBタイムインター  
 バル/パルス幅A&AtoBタイムインターバル/パ  
 ルス幅A&パルス幅B測定, マルチウインドウ/  
 オートウインドウのとき  
 :CALCULATION:MEAS1:WINDOW1:  
 AVERAGE?  
 -> :CALCULATION:MEAS1:WINDOW1:  
 AVERAGE 1.00035E-7  
 ●符号間干渉解析モードの場合

:CALCULATION:WINDOW1:AVERAGE? ALL  
 -> :CALCULATION:WINDOW1:  
 AVERAGE 1.00035E-7  
 解説  
 ・統計値が有効でない場合は, 問い合わせに対し  
 て「NAN」を返します。  
 ・シングルウインドウのときは「:WINDOW<x>」  
 を省略してください  
 ・周期/AtoBタイムインターバル/パルス幅測定  
 のときは, 「:MEAS<x>」を省略してくださ  
 い。  
 ・「:CALCulation:POLarity」が  
 「POSNeg」のとき, 「POSitive|  
 NEGative」を指定してください。  
 ・符号間干渉解析モードのとき, 「ALL|  
 SElect」を指定してください。

**:CALCulation[:MEAS<x>][:WINDOW<x>]:**

**DEVIation?**

機能 ヒストグラム時のDeviationを問い合わせます。  
 構文 :CALCulation[:MEAS<x>][:WINDOW<x>]:  
 DEVIation? [{POSitive|NEGative|ALL|  
 SElect}]  
 MEAS<x>の<x>=1~2  
 WINDOW<x>の<x>=ウインドウ番号(1~ウイン  
 ドウ数)

例 ●タイムスタンプ/ハードウェアヒストグラム  
 モードの場合  
 ・周期/AtoBタイムインターバル/パルス幅測  
 定, シングルウインドウのとき  
 :CALCULATION:DEVIATION?  
 -> :CALCULATION:  
 DEVIATION 1.4425E+1  
 ・周期/AtoBタイムインターバル/パルス幅測  
 定, マルチウインドウ/オートウインドウの  
 とき  
 :CALCULATION:WINDOW1:DEVIATION?  
 -> :CALCULATION:WINDOW1:  
 DEVIATION 1.4425E+1  
 ・周期A&周期B/周期A&AtoBタイムインター  
 バル/パルス幅A&AtoBタイムインターバル/パ  
 ルス幅A&パルス幅B測定, シングルウイン  
 ドウのとき  
 :CALCULATION:MEAS1:DEVIATION?  
 -> :CALCULATION:MEAS1:  
 DEVIATION 1.4425E+1  
 ・周期A&周期B/周期A&AtoBタイムインター  
 バル/パルス幅A&AtoBタイムインターバル/パ  
 ルス幅A&パルス幅B測定, マルチウインドウ/  
 オートウインドウのとき  
 :CALCULATION:MEAS1:WINDOW1:  
 DEVIATION?  
 -> :CALCULATION:MEAS1:WINDOW1:  
 DEVIATION 1.4425E+1  
 ●符号間干渉解析モードの場合

:CALCULATION:WINDOW1:DEVIATION?  
 ALL  
 -> :CALCULATION:WINDOW1:  
 DEVIATION 1.4425E+1  
 解説  
 ・統計値が有効でない場合は, 問い合わせに対し  
 て「NAN」を返します。  
 ・シングルウインドウのときは「:WINDOW<x>」  
 を省略してください  
 ・周期/AtoBタイムインターバル/パルス幅測定  
 のときは, 「:MEAS<x>」を省略してくださ  
 い。  
 ・「:CALCulation:POLarity」が  
 「POSNeg」のとき, 「POSitive|  
 NEGative」を指定してください。  
 ・符号間干渉解析モードのとき, 「ALL|  
 SElect」を指定してください。

**:CALCulation[:MEAS<x>][:WINDow<x>]:DEVT?**

機能 ヒストグラム時のDeviation/T値を問い合わせます。

構文 **:CALCulation[:MEAS<x>][:WINDow<x>]:DEVT? [{POSitive|NEGative|ALL|SElect}]**  
 MEAS<x>の<x>=1~2  
 WINDow<x>の<x>=ウインドウ番号(1~ウインドウ数)

例 ●タイムスタンプ/ハードウェアヒストグラムモードの場合

・周期/AtoBタイムインターバル/パルス幅測定, シングルウインドウのとき

**:CALCULATION:DEVT?**  
 -> **:CALCULATION:DEVT 9.99437E-8**

・周期/AtoBタイムインターバル/パルス幅測定, マルチウインドウ/オートウインドウのとき

**:CALCULATION:WINDOW1:DEVT?**  
 -> **:CALCULATION:WINDOW1:DEVT 9.99437E-8**

・周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定, シングルウインドウのとき

**:CALCULATION:MEAS1:DEVT?**  
 -> **:CALCULATION:MEAS1:DEVT 9.99437E-8**

・周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定, マルチウインドウ/オートウインドウのとき

**:CALCULATION:MEAS1:WINDOW1:DEVT?**  
 -> **:CALCULATION:MEAS1:WINDOW1:DEVT 9.99437E-8**

●符号間干渉解析モードの場合

**:CALCULATION:WINDOW1:DEVT? ALL**  
 -> **:CALCULATION:WINDOW1:DEVT 9.99437E-8**

解説 ・統計値が有効でない場合は, 問い合わせに対して「NAN」を返します。

・シングルウインドウのときは「:WINDow<x>」を省略してください

・周期/AtoBタイムインターバル/パルス幅測定のときは, 「:MEAS<x>」を省略してください。

・「:CALCulation:POLarity」が「POSneg」のとき, 「POSitive|NEGative」を指定してください。

・符号間干渉解析モードのとき, 「ALL|SElect」を指定してください。

**:CALCulation[:MEAS<x>][:WINDow<x>]:****FLUTter?**

機能 ヒストグラム時の $\sigma$ /AVE値(フラッタ)を問い合わせます。

構文 **:CALCulation[:MEAS<x>][:WINDow<x>]:FLUTter? [{POSitive|NEGative|ALL|SElect}]**  
 MEAS<x>の<x>=1~2  
 WINDow<x>の<x>=ウインドウ番号(1~ウインドウ数)

例 ●タイムスタンプ/ハードウェアヒストグラムモードの場合

・周期/AtoBタイムインターバル/パルス幅測定, シングルウインドウのとき

**:CALCULATION:FLUTTER?**  
 -> **:CALCULATION:FLUTTER 1.2887E+1**

・周期/AtoBタイムインターバル/パルス幅測定, マルチウインドウ/オートウインドウのとき

**:CALCULATION:WINDOW1:FLUTTER?**  
 -> **:CALCULATION:WINDOW1:FLUTTER 1.2887E+1**

・周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定, シングルウインドウのとき

**:CALCULATION:MEAS1:FLUTTER?**  
 -> **:CALCULATION:MEAS1:FLUTTER 1.2887E+1**

・周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定, マルチウインドウ/オートウインドウのとき

**:CALCULATION:MEAS1:WINDOW1:FLUTTER?**  
 -> **:CALCULATION:MEAS1:WINDOW1:FLUTTER 1.2887E+1**

●符号間干渉解析モードの場合

**:CALCULATION:WINDOW1:FLUTTER? ALL**  
 -> **:CALCULATION:WINDOW1:FLUTTER 1.2887E+1**

解説 ・統計値が有効でない場合は, 問い合わせに対して「NAN」を返します。

・シングルウインドウのときは「:WINDow<x>」を省略してください

・周期/AtoBタイムインターバル/パルス幅測定のときは, 「:MEAS<x>」を省略してください。

・「:CALCulation:POLarity」が「POSneg」のとき, 「POSitive|NEGative」を指定してください。

・符号間干渉解析モードのとき, 「ALL|SElect」を指定してください。

### 4.3 CALCulationグループ

**:CALCulation[:MEAS<x>][:WINDow<x>|SUMMation]:JITTer?**

機能 ヒストグラム時の $\sigma/T$ 値(ジッタ)を問い合わせます。

構文 :CALCulation[:MEAS<x>][:WINDow<x>|SUMMation]:JITTer? [{POSitive|NEGative|ALL|SElect}]  
MEAS<x>の<x>=1~2  
WINDow<x>の<x>=ウインドウ番号(1~ウインドウ数)

SUMMation: 全ウインドウを重ね合わせたときの統計値を問い合わせます。

例 ●タイムスタンプ/ハードウェアヒストグラムモードの場合

・周期/AtoBタイムインターバル/パルス幅測定, シングルウインドウのとき

```
:CALCULATION:JITTER?
-> :CALCULATION:JITTER 1.2366E+1
```

・周期/AtoBタイムインターバル/パルス幅測定, マルチウインドウ/オートウインドウのとき

```
:CALCULATION:WINDOW1:JITTER?
-> :CALCULATION:WINDOW1:
JITTER 1.2366E+1
```

・周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定, シングルウインドウのとき

```
:CALCULATION:MEAS1:JITTER?
-> :CALCULATION:MEAS1:
JITTER 1.2366E+1
```

・周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定, マルチウインドウ/オートウインドウのとき

```
:CALCULATION:MEAS1:WINDOW1:
JITTER?
-> :CALCULATION:MEAS1:WINDOW1:
JITTER 1.2366E+1
```

●符号間干渉解析モードの場合

```
:CALCULATION:WINDOW1:JITTER? ALL
-> :CALCULATION:WINDOW1:
JITTER 1.2366E+1
```

解説 ・統計値が有効でない場合は, 問い合わせに対して「NAN」を返します。

・シングルウインドウのときは「{:WINDow<x>|SUMMation}」を省略してください

・周期/AtoBタイムインターバル/パルス幅測定のときは, 「:MEAS<x>」を省略してください。

・「:CALCulation:POLarity」が「POSNeg」のとき, 「POSitive|NEGative」を指定してください。

・符号間干渉解析モードのとき, 「ALL|SElect」を指定してください。

**:CALCulation[:MEAS<x>][:WINDow<x>]:**

**MAXimum?**

機能 ヒストグラム時の最大値を問い合わせます。

構文 :CALCulation[:MEAS<x>][:WINDow<x>]:MAXimum? [{POSitive|NEGative|ALL|SElect}]  
MEAS<x>の<x>=1~2

WINDow<x>の<x>=ウインドウ番号(1~ウインドウ数)

例 ●タイムスタンプ/ハードウェアヒストグラムモードの場合

・周期/AtoBタイムインターバル/パルス幅測定, シングルウインドウのとき

```
:CALCULATION:MAXIMUM?
-> :CALCULATION:MAXIMUM 1.1287E-7
```

・周期/AtoBタイムインターバル/パルス幅測定, マルチウインドウ/オートウインドウのとき

```
:CALCULATION:WINDOW1:MAXIMUM?
-> :CALCULATION:WINDOW1:
MAXIMUM 1.1287E-7
```

・周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定, シングルウインドウのとき

```
:CALCULATION:MEAS1:MAXIMUM?
-> :CALCULATION:MEAS1:
MAXIMUM 1.1287E-7
```

・周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定, マルチウインドウ/オートウインドウのとき

```
:CALCULATION:MEAS1:WINDOW1:
MAXIMUM?
-> :CALCULATION:MEAS1:WINDOW1:
MAXIMUM 1.1287E-7
```

●符号間干渉解析モードの場合

```
:CALCULATION:WINDOW1:MAXIMUM? ALL
-> :CALCULATION:WINDOW1:
MAXIMUM 1.1287E-7
```

解説 ・統計値が有効でない場合は, 問い合わせに対して「NAN」を返します。

・シングルウインドウのときは「:WINDow<x>」を省略してください

・周期/AtoBタイムインターバル/パルス幅測定のときは, 「:MEAS<x>」を省略してください。

・「:CALCulation:POLarity」が「POSNeg」のとき, 「POSitive|NEGative」を指定してください。

・符号間干渉解析モードのとき, 「ALL|SElect」を指定してください。

**:CALCulation[:MEAS<x>][:WINDOW<x>]:****MEDian?**

機能 ヒストグラム時の中央値を問い合わせます。  
 構文 **:CALCulation[:MEAS<x>][:WINDOW<x>]:**  
**MEDian? [{POSitive|NEGative|ALL|**  
**SElect}]**  
 MEAS<x>の<x>=1~2  
 WINDOW<x>の<x>=ウインドウ番号(1~ウイン  
 ドウ数)

例 ●タイムスタンプ/ハードウェアヒストグラム  
 モードの場合

- ・周期/AtoBタイムインターバル/パルス幅測  
 定, シングルウインドウのとき

**:CALCULATION:MEDIAN?**  
 -> **:CALCULATION:MEDIAN 1.00145E-7**

- ・周期/AtoBタイムインターバル/パルス幅測  
 定, マルチウインドウ/オートウインドウの  
 とき

**:CALCULATION:WINDOW1:MEDIAN?**  
 -> **:CALCULATION:WINDOW1:**  
**MEDIAN 1.00145E-7**

- ・周期A&周期B/周期A&AtoBタイムインター  
 バル/パルス幅A&AtoBタイムインターバル/パ  
 ルス幅A&パルス幅B測定, シングルウイン  
 ドウのとき

**:CALCULATION:MEAS1:MEDIAN?**  
 -> **:CALCULATION:MEAS1:**  
**MEDIAN 1.00145E-7**

- ・周期A&周期B/周期A&AtoBタイムインター  
 バル/パルス幅A&AtoBタイムインターバル/パ  
 ルス幅A&パルス幅B測定, マルチウインドウ/  
 オートウインドウのとき

**:CALCULATION:MEAS1:WINDOW1:**  
**MEDIAN?**  
 -> **:CALCULATION:MEAS1:WINDOW1:**  
**MEDIAN 1.00145E-7**

●符号間干渉解析モードの場合

**:CALCULATION:WINDOW1:MEDIAN? ALL**

-> **:CALCULATION:WINDOW1:**

**MEDIAN 1.00145E-7**

解説 ・統計値が有効でない場合は、問い合わせに対し

て「NAN」を返します。

- ・シングルウインドウのときは「:WINDOW<x>」  
 を省略してください

- ・周期/AtoBタイムインターバル/パルス幅測定  
 のときは、「:MEAS<x>」を省略してくださ  
 い。

- ・「:CALCulation:POLarity」が  
 「POSneg」のとき、「POSitive|  
 NEGative」を指定してください。

- ・符号間干渉解析モードのとき、「ALL|  
 SElect」を指定してください。

**:CALCulation[:MEAS<x>][:WINDOW<x>]:****MINimum?**

機能 ヒストグラム時の最小値を問い合わせます。  
 構文 **:CALCulation[:MEAS<x>][:WINDOW<x>]:**  
**MINimum? [{POSitive|NEGative|ALL|**  
**SElect}]**  
 MEAS<x>の<x>=1~2  
 WINDOW<x>の<x>=ウインドウ番号(1~ウイン  
 ドウ数)

例 ●タイムスタンプ/ハードウェアヒストグラム  
 モードの場合

- ・周期/AtoBタイムインターバル/パルス幅測  
 定, シングルウインドウのとき

**:CALCULATION:MINIMUM?**  
 -> **:CALCULATION:**  
**MINIMUM 9.99437E-8**

- ・周期/AtoBタイムインターバル/パルス幅測  
 定, マルチウインドウ/オートウインドウの  
 とき

**:CALCULATION:WINDOW1:MINIMUM?**  
 -> **:CALCULATION:WINDOW1:**  
**MINIMUM 9.99437E-8**

- ・周期A&周期B/周期A&AtoBタイムインター  
 バル/パルス幅A&AtoBタイムインターバル/パ  
 ルス幅A&パルス幅B測定, シングルウイン  
 ドウのとき

**:CALCULATION:MEAS1:MINIMUM?**  
 -> **:CALCULATION:MEAS1:**  
**MINIMUM 9.99437E-8**

- ・周期A&周期B/周期A&AtoBタイムインター  
 バル/パルス幅A&AtoBタイムインターバル/パ  
 ルス幅A&パルス幅B測定, マルチウインドウ/  
 オートウインドウのとき

**:CALCULATION:MEAS1:WINDOW1:**  
**MINIMUM?**  
 -> **:CALCULATION:MEAS1:WINDOW1:**  
**MINIMUM 9.99437E-8**

●符号間干渉解析モードの場合

**:CALCULATION:WINDOW1:MINIMUM? ALL**

-> **:CALCULATION:WINDOW1:**

**MINIMUM 9.99437E-8**

解説 ・統計値が有効でない場合は、問い合わせに対し

- て「NAN」を返します。
- ・シングルウインドウのときは「:WINDOW<x>」  
 を省略してください

- ・周期/AtoBタイムインターバル/パルス幅測定  
 のときは、「:MEAS<x>」を省略してくださ  
 い。

- ・「:CALCulation:POLarity」が  
 「POSneg」のとき、「POSitive|  
 NEGative」を指定してください。

- ・符号間干渉解析モードのとき、「ALL|  
 SElect」を指定してください。

### 4.3 CALCulationグループ

#### :CALCulation[:MEAS<x>][:WINDOW<x>]:MODE?

機能 ヒストグラム時の最頻値を問い合わせます。  
 構文 :CALCulation[:MEAS<x>][:WINDOW<x>]:  
 MODE? [{POSitive|NEGative|ALL|  
 SElect}]  
 MEAS<x>の<x>=1~2  
 WINDOW<x>の<x>=ウインドウ番号(1~ウイン  
 ドウ数)

#### 例 ●タイムスタンプ/ハードウェアヒストグラム モードの場合

・周期/AtoBタイムインターバル/パルス幅測  
 定, シングルウインドウのとき

:CALCULATION:MODE?  
 -> :CALCULATION:MODE 1.00025E-7

・周期/AtoBタイムインターバル/パルス幅測  
 定, マルチウインドウ/オートウインドウの  
 とき

:CALCULATION:WINDOW1:MODE?  
 -> :CALCULATION:WINDOW1:  
 MODE 1.00025E-7

・周期A&周期B/周期A&AtoBタイムインター  
 バル/パルス幅A&AtoBタイムインターバル/パ  
 ルス幅A&パルス幅B測定, シングルウイン  
 ドウのとき

:CALCULATION:MEAS1:MODE?  
 -> :CALCULATION:MEAS1:  
 MODE 1.00025E-7

・周期A&周期B/周期A&AtoBタイムインター  
 バル/パルス幅A&AtoBタイムインターバル/パ  
 ルス幅A&パルス幅B測定, マルチウインドウ/  
 オートウインドウのとき

:CALCULATION:MEAS1:WINDOW1:MODE?  
 -> :CALCULATION:MEAS1:WINDOW1:  
 MODE 1.00025E-7

#### ●符号間干渉解析モードの場合

:CALCULATION:WINDOW1:MODE? ALL  
 -> :CALCULATION:WINDOW1:  
 MODE 1.00025E-7

解説  
 ・統計値が有効でない場合は, 問い合わせに対し  
 て「NAN」を返します。  
 ・シングルウインドウのときは「:WINDOW<x>」  
 を省略してください  
 ・周期/AtoBタイムインターバル/パルス幅測  
 定のときは, 「:MEAS<x>」を省略してくだ  
 さい。  
 ・「:CALCulation:POLarity」が  
 「POSNeg」のとき, 「POSitive|  
 NEGative」を指定してください。  
 ・符号間干渉解析モードのとき, 「ALL|  
 SElect」を指定してください。

#### :CALCulation[:MEAS<x>][:{WINDOW<x>| SUMMation}]:PTOPeak?

機能 ヒストグラム時のP-P値を問い合わせます。  
 構文 :CALCulation[:MEAS<x>][:{WINDOW<x>|  
 SUMMation}]:PTOPeak? [{POSitive|  
 NEGative|ALL|SElect}]  
 MEAS<x>の<x>=1~2  
 WINDOW<x>の<x>=ウインドウ番号(1~ウイン  
 ドウ数)

SUMMation: 全ウインドウを重ね合わせたとき  
 の統計値を問い合わせます。

#### 例 ●タイムスタンプ/ハードウェアヒストグラム モードの場合

・周期/AtoBタイムインターバル/パルス幅測  
 定, シングルウインドウのとき

:CALCULATION:PTOPEAK?  
 -> :CALCULATION:PTOPEAK 3.0245E-9

・周期/AtoBタイムインターバル/パルス幅測  
 定, マルチウインドウ/オートウインドウの  
 とき

:CALCULATION:WINDOW1:PTOPEAK?  
 -> :CALCULATION:WINDOW1:  
 PTOPEAK 3.0245E-9

・周期A&周期B/周期A&AtoBタイムインター  
 バル/パルス幅A&AtoBタイムインターバル/パ  
 ルス幅A&パルス幅B測定, シングルウイン  
 ドウのとき

:CALCULATION:MEAS1:PTOPEAK?  
 -> :CALCULATION:MEAS1:  
 PTOPEAK 3.0245E-9

・周期A&周期B/周期A&AtoBタイムインター  
 バル/パルス幅A&AtoBタイムインターバル/パ  
 ルス幅A&パルス幅B測定, マルチウインドウ/  
 オートウインドウのとき

:CALCULATION:MEAS1:WINDOW1:  
 PTOPEAK?  
 -> :CALCULATION:MEAS1:WINDOW1:  
 PTOPEAK 3.0245E-9

#### ●符号間干渉解析モードの場合

:CALCULATION:WINDOW1:PTOPeak? ALL  
 -> :CALCULATION:WINDOW1:  
 PTOPeak 3.0245E-9

解説  
 ・統計値が有効でない場合は, 問い合わせに対し  
 て「NAN」を返します。  
 ・シングルウインドウのときは  
 「:{WINDOW<x>|SUMMation}」を省略して  
 ください  
 ・周期/AtoBタイムインターバル/パルス幅測  
 定のときは, 「:MEAS<x>」を省略してくだ  
 さい。  
 ・「:CALCulation:POLarity」が  
 「POSNeg」のとき, 「POSitive|  
 NEGative」を指定してください。  
 ・符号間干渉解析モードのとき, 「ALL|  
 SElect」を指定してください。

**:CALCulation[:MEAS<x>][[:{WINDOW<x>|SUMMation}]:SDEVIation?**

機能 ヒストグラム時の標準偏差( $\sigma$ )を問い合わせます。  
 構文 :CALCulation[:MEAS<x>][[:{WINDOW<x>|SUMMation}]:SDEVIation? [{POSitive|NEGative|ALL|SElect}]]  
 MEAS<x>の<x>=1~2  
 WINDOW<x>の<x>=ウインドウ番号(1~ウインドウ数)  
 SUMMation: 全ウインドウを重ね合わせたときの統計値を問い合わせます。

**●タイムスタンプ/ハードウェアヒストグラムモードの場合**

・ 周期/AtoBタイムインターバル/パルス幅測定, シングルウインドウのとき

:CALCULATION:SDEVIATION?

-> :CALCULATION:

SDEVIATION 2.4035E-9

・ 周期/AtoBタイムインターバル/パルス幅測定, マルチウインドウ/オートウインドウのとき

:CALCULATION:WINDOW1:SDEVIATION?

-> :CALCULATION:WINDOW1:

SDEVIATION 2.4035E-9

・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定, シングルウインドウのとき

:CALCULATION:MEAS1:SDEVIATION?

-> :CALCULATION:MEAS1:

SDEVIATION 2.4035E-9

・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定, マルチウインドウ/オートウインドウのとき

:CALCULATION:MEAS1:WINDOW1:

SDEVIATION?

-> :CALCULATION:MEAS1:WINDOW1:

SDEVIATION 2.4035E-9

**●符号間干渉解析モードの場合**

:CALCULATION:WINDOW1:SDEVIATION?

ALL

-> :CALCULATION:WINDOW1:

SDEVIATION 2.4035E-9

解説 ・ 統計値が有効でない場合は, 問い合わせに対して「NAN」を返します。

・ シングルウインドウのときは  
「:{WINDOW<x>|SUMMation}」を省略してください

・ 周期/AtoBタイムインターバル/パルス幅測定の場合は, 「:MEAS<x>」を省略してください。

・ 「:CALCulation:POLarity」が「POSNeg」のとき, 「POSitive|NEGative」を指定してください。

・ 符号間干渉解析モードのとき, 「ALL|SElect」を指定してください。

**:CALCulation[:MEAS<x>][[:{WINDOW<x>|SUMMation}]:SNUMBER?**

機能 ヒストグラム時の統計演算対象のサンプル数を問い合わせます。

構文 :CALCulation[:MEAS<x>][[:{WINDOW<x>|SUMMation}]:SNUMBER? [{POSitive|NEGative|ALL|SElect}]]  
 MEAS<x>の<x>=1~2  
 WINDOW<x>の<x>=ウインドウ番号(1~ウインドウ数)  
 SUMMation: 全ウインドウを重ね合わせたときの統計値を問い合わせます。

**●タイムスタンプ/ハードウェアヒストグラムモードの場合**

・ 周期/AtoBタイムインターバル/パルス幅測定, シングルウインドウのとき

:CALCULATION:SNUMBER?

-> :CALCULATION:SNUMBER 6.000E+3

・ 周期/AtoBタイムインターバル/パルス幅測定, マルチウインドウ/オートウインドウのとき

:CALCULATION:WINDOW1:SNUMBER?

-> :CALCULATION:WINDOW1:

SNUMBER 6.000E+3

・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定, シングルウインドウのとき

:CALCULATION:MEAS1:SNUMBER?

-> :CALCULATION:MEAS1:

SNUMBER 6.000E+3

・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定, マルチウインドウ/オートウインドウのとき

:CALCULATION:MEAS1:WINDOW1:

SNUMBER?

-> :CALCULATION:MEAS1:WINDOW1:

SNUMBER 6.000E+3

**●符号間干渉解析モードの場合**

:CALCULATION:WINDOW1:SNUMBER? ALL

-> :CALCULATION:WINDOW1:

SNUMBER 6.000E+3

解説 ・ 統計値が有効でない場合は, 問い合わせに対して「NAN」を返します。

・ シングルウインドウのときは  
「:{WINDOW<x>|SUMMation}」を省略してください

・ 周期/AtoBタイムインターバル/パルス幅測定の場合は, 「:MEAS<x>」を省略してください。

・ 「:CALCulation:POLarity」が「POSNeg」のとき, 「POSitive|NEGative」を指定してください。

・ 符号間干渉解析モードのとき, 「ALL|SElect」を指定してください。

### 4.3 CALCulationグループ

#### **:CALCulation:PARAmeter?**

機能 統計演算全体のON/OFFを問い合わせます。  
構文 :CALCulation:PARAmeter?  
例 :CALCULATION:PARAMETER?  
-> :CALCULATION:PARAMETER:AVERAGE  
1;  
DEVIATION 1;DEVT 1;FLUTTER 1;  
JITTER 1;MAXIMUM 1;MEDIAN 1;  
MINIMUM 1;MODE 1;PTOPEAK 1;  
SDEVIATION 1

#### **:CALCulation:PARAmeter:CLEAr**

機能 全統計演算値のオフを実行します。  
構文 :CALCulation:PARAmeter:CLEAr  
例 :CALCULATION:PARAMETER:CLEAR

#### **:CALCulation:PARAmeter:AVERAge**

機能 ヒストグラム時の平均値の演算のON/OFFを設定/問い合わせします。  
構文 :CALCulation:PARAmeter:AVERAge  
{<Boolean>}  
:CALCulation:PARAmeter:AVERAge?  
例 :CALCULATION:PARAMETER:AVERAGE ON  
:CALCULATION:PARAMETER:AVERAGE?  
-> :CALCULATION:PARAMETER:AVERAGE 1

#### **:CALCulation:PARAmeter:DEVIation**

機能 ヒストグラム時のDeviationの演算のON/OFFを設定/問い合わせします。  
構文 :CALCulation:PARAmeter:DEVIation  
{<Boolean>}  
:CALCulation:PARAmeter:DEVIation?  
例 :CALCULATION:PARAMETER:  
DEVIATION ON  
:CALCULATION:PARAMETER:DEVIATION?  
-> :CALCULATION:PARAMETER:  
DEVIATION 1

#### **:CALCulation:PARAmeter:DEVT**

機能 ヒストグラム時のDeviation/Tの演算のON/OFFを設定/問い合わせします。  
構文 :CALCulation:PARAmeter:DEVT  
{<Boolean>}  
:CALCulation:PARAmeter:DEVT?  
例 :CALCULATION:PARAMETER:DEVT ON  
:CALCULATION:PARAMETER:DEVT?  
-> :CALCULATION:PARAMETER:DEVT 1

#### **:CALCulation:PARAmeter:FLUTter**

機能 ヒストグラム時の $\sigma$ /AVE値(フラッタ)の演算のON/OFFを設定/問い合わせします。  
構文 :CALCulation:PARAmeter:FLUTter  
{<Boolean>}  
:CALCulation:PARAmeter:FLUTter?  
例 :CALCULATION:PARAMETER:FLUTTER ON  
:CALCULATION:PARAMETER:FLUTTER?  
-> :CALCULATION:PARAMETER:FLUTTER 1

#### **:CALCulation:PARAmeter:JITTer**

機能 ヒストグラム時の $\sigma$ /T値(ジッタ)の演算のON/OFFを設定/問い合わせします。  
構文 :CALCulation:PARAmeter:JITTer  
{<Boolean>}  
:CALCulation:PARAmeter:JITTer?  
例 :CALCULATION:PARAMETER:JITTER ON  
:CALCULATION:PARAMETER:JITTER?  
-> :CALCULATION:PARAMETER:JITTER 1

#### **:CALCulation:PARAmeter:MAXimum**

機能 ヒストグラム時の最大値の演算のON/OFFを設定/問い合わせします。  
構文 :CALCulation:PARAmeter:MAXimum  
{<Boolean>}  
:CALCulation:PARAmeter:MAXimum?  
例 :CALCULATION:PARAMETER:MAXIMUM ON  
:CALCULATION:PARAMETER:MAXIMUM?  
-> :CALCULATION:PARAMETER:MAXIMUM 1

#### **:CALCulation:PARAmeter:MEDIan**

機能 ヒストグラム時の中央値の演算のON/OFFを設定/問い合わせします。  
構文 :CALCulation:PARAmeter:MEDIan  
{<Boolean>}  
:CALCulation:PARAmeter:MEDIan?  
例 :CALCULATION:PARAMETER:MEDIAN ON  
:CALCULATION:PARAMETER:MEDIAN?  
-> :CALCULATION:PARAMETER:MEDIAN 1

#### **:CALCulation:PARAmeter:MINimum**

機能 ヒストグラム時の最小値の演算のON/OFFを設定/問い合わせします。  
構文 :CALCulation:PARAmeter:MINimum  
{<Boolean>}  
:CALCulation:PARAmeter:MINimum?  
例 :CALCULATION:PARAMETER:MINIMUM ON  
:CALCULATION:PARAMETER:MINIMUM?  
-> :CALCULATION:PARAMETER:MINIMUM 1

#### **:CALCulation:PARAmeter:MODE**

機能 ヒストグラム時の最頻値の演算のON/OFFを設定/問い合わせします。  
構文 :CALCulation:PARAmeter:MODE  
{<Boolean>}  
:CALCulation:PARAmeter:MODE?  
例 :CALCULATION:PARAMETER:MODE ON  
:CALCULATION:PARAMETER:MODE?  
-> :CALCULATION:PARAMETER:MODE 1

**:CALCulation:PARAmeter:PTOPeak**

機能 ヒストグラム時のP-P値の演算のON/OFFを設定/問い合わせします。

構文 :CALCulation:PARAmeter:PTOPeak  
{<Boolean>}

例 :CALCulation:PARAmeter:PTOPeak?  
:CALCULATION:PARAMETER:PTOPEAK ON  
:CALCULATION:PARAMETER:PTOPEAK?  
-> :CALCULATION:PARAMETER:PTOPEAK 1

**:CALCulation:PARAmeter:SDEVIation**

機能 ヒストグラム時の標準偏差( $\sigma$ )の演算のON/OFFを設定/問い合わせします。

構文 :CALCulation:PARAmeter:SDEVIation  
{<Boolean>}

例 :CALCulation:PARAmeter:SDEVIation?  
:CALCULATION:PARAMETER:SDEVIATION ON  
:CALCULATION:PARAMETER:SDEVIATION?  
-> :CALCULATION:PARAMETER:SDEVIATION 1

**:CALCulation:PARAmeter:TAVerage**

機能 タイムバリエーション時の平均値の演算のON/OFFを設定/問い合わせします。

構文 :CALCulation:PARAmeter:TAVerage  
{<Boolean>}

例 :CALCulation:PARAmeter:TAVerage?  
:CALCULATION:PARAMETER:TAVERAGE ON  
:CALCULATION:PARAMETER:TAVERAGE?  
-> :CALCULATION:PARAMETER:TAVERAGE 1

**:CALCulation:PARAmeter:TFLutter**

機能 タイムバリエーション時の $\sigma$ /AVE値(フラッタ)の演算のON/OFFを設定/問い合わせします。

構文 :CALCulation:PARAmeter:TFLutter  
{<Boolean>}

例 :CALCulation:PARAmeter:TFLutter?  
:CALCULATION:PARAMETER:TFLUTTER ON  
:CALCULATION:PARAMETER:TFLUTTER?  
-> :CALCULATION:PARAMETER:TFLUTTER 1

**:CALCulation:PARAmeter:TJITter**

機能 タイムバリエーション時のP-P/AVE値(ジッタ)の演算のON/OFFを設定/問い合わせします。

構文 :CALCulation:PARAmeter:TJITter  
{<Boolean>}

例 :CALCulation:PARAmeter:TJITter?  
:CALCULATION:PARAMETER:TJITTER ON  
:CALCULATION:PARAMETER:TJITTER?  
-> :CALCULATION:PARAMETER:TJITTER 1

**:CALCulation:PARAmeter:TMAXimum**

機能 タイムバリエーション時の最大値の演算のON/OFFを設定/問い合わせします。

構文 :CALCulation:PARAmeter:TMAXimum  
{<Boolean>}

例 :CALCulation:PARAmeter:TMAXimum?  
:CALCULATION:PARAMETER:TMAXIMUM ON  
:CALCULATION:PARAMETER:TMAXIMUM?  
-> :CALCULATION:PARAMETER:TMAXIMUM 1

**:CALCulation:PARAmeter:TMINimum**

機能 タイムバリエーション時の最小値の演算のON/OFFを設定/問い合わせします。

構文 :CALCulation:PARAmeter:TMINimum  
{<Boolean>}

例 :CALCulation:PARAmeter:TMINimum?  
:CALCULATION:PARAMETER:TMINIMUM ON  
:CALCULATION:PARAMETER:TMINIMUM?  
-> :CALCULATION:PARAMETER:TMINIMUM 1

**:CALCulation:PARAmeter:TPTOpeak**

機能 タイムバリエーション時のP-P値の演算のON/OFFを設定/問い合わせします。

構文 :CALCulation:PARAmeter:TPTOpeak  
{<Boolean>}

例 :CALCulation:PARAmeter:TPTOpeak?  
:CALCULATION:PARAMETER:TPTOPEAK ON  
:CALCULATION:PARAMETER:TPTOPEAK?  
-> :CALCULATION:PARAMETER:TPTOPEAK 1

**:CALCulation:PARAmeter:TRF**

機能 タイムバリエーション時のRF値の演算のON/OFFを設定/問い合わせします。

構文 :CALCulation:PARAmeter:TRF  
{<Boolean>}

例 :CALCulation:PARAmeter:TRF?  
:CALCULATION:PARAMETER:TRF ON  
:CALCULATION:PARAMETER:TRF?  
-> :CALCULATION:PARAMETER:TRF 1

**:CALCulation:PARAmeter:TSDEVIation**

機能 タイムバリエーション時の標準偏差( $\sigma$ )の演算のON/OFFを設定/問い合わせします。

構文 :CALCulation:PARAmeter:TSDEVIation  
{<Boolean>}

例 :CALCulation:PARAmeter:TSDEVIation?  
:CALCULATION:PARAMETER:TSDEVIATION ON  
:CALCULATION:PARAMETER:TSDEVIATION?  
-> :CALCULATION:PARAMETER:TSDEVIATION 1

### 4.3 CALCulationグループ/4.4 COMMunicateグループ

#### :CALCulation:POLarity

機能 両パルス幅/両エッジ測定時の解析ポラリティを設定/問い合わせします。

構文 `:CALCulation:POLarity {POSitive|NEGative|BOTH|POSNeg}`  
`:CALCulation:POLarity?`

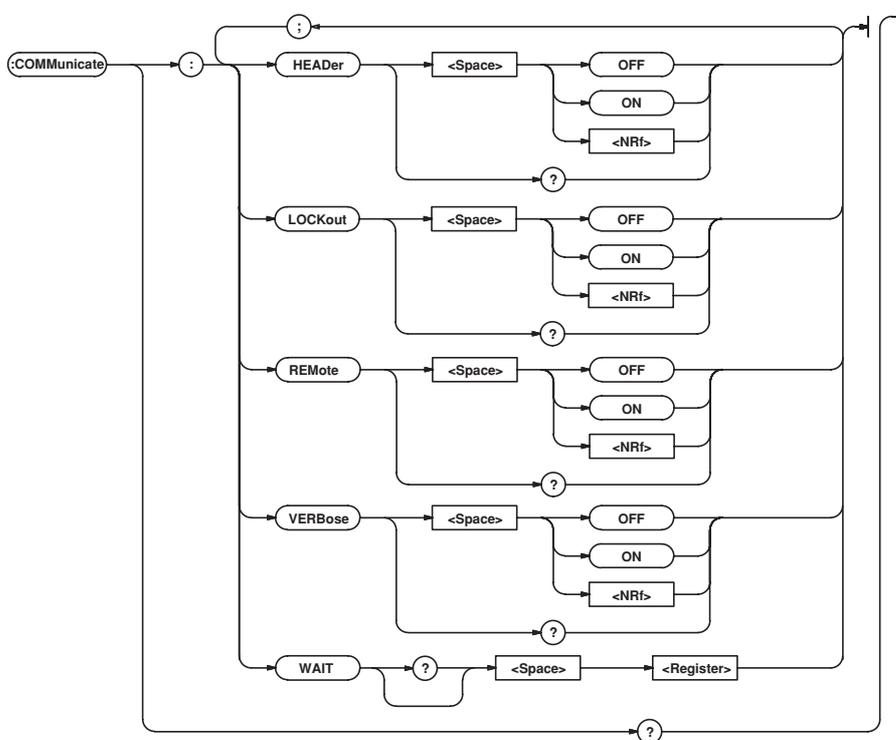
例 `:CALCULATION:POLARITY POSITIVE`  
`:CALCULATION:POLARITY?`  
`-> :CALCULATION:POLARITY POSITIVE`

解説

- 両パルス幅/両エッジ測定するとき、このコマンドで指定したポラリティの統計値を応答します。
- 「POSNeg」を設定したときは、統計値を問い合わせるクエリに「POSitive|NEGative」を指定してください。

### 4.4 COMMunicateグループ

通信に関するグループです。このグループに相当するフロントパネルのキーはありません。



#### :COMMunicate?

機能 通信に関する全設定を問い合わせます。

構文 `:COMMunicate?`

例 `:COMMUNICATE?`  
`-> :COMMUNICATE:HEADER 1;VERBOSE 1`

#### :COMMunicate:HEADer

機能 クエリに対する応答にヘッダを付けるか付けないかを設定/問い合わせします。

構文 `:COMMunicate:HEADer {<Boolean>}`  
`:COMMunicate:HEADer?`

例 `:COMMUNICATE:HEADER ON`  
`:COMMUNICATE:HEADER?`  
`-> :COMMUNICATE:HEADER 1`

#### :COMMunicate:LOCKout

機能 ローカルロックアウトを設定/解除します。

構文 `:COMMunicate:LOCKout {<Boolean>}`  
`:COMMunicate:LOCKout?`

例 `:COMMUNICATE:LOCKOUT ON`  
`:COMMUNICATE:LOCKOUT?`  
`-> :COMMUNICATE:LOCKOUT 1`

解説 イーサネット通信インタフェース専用のコマンドです。

**:COMMunicate:REMOte**

機能	リモート/ローカルを設定します。ONのときにリモートになります。
構文	<code>:COMMunicate:REMOte {&lt;Boolean&gt;}</code> <code>:COMMunicate:REMOte?</code>
例	<code>:COMMUNICATE:REMOTE ON</code> <code>:COMMUNICATE:REMOTE?</code> -> <code>:COMMUNICATE:REMOTE 1</code>
解説	イーサネット通信インタフェース専用のコマンドです。

**:COMMunicate:VERBose**

機能	クエリに対する応答をフルスペルで返送するか省略形で返送するかを設定/問い合わせします。
構文	<code>:COMMunicate:VERBose {&lt;Boolean&gt;}</code> <code>:COMMunicate:VERBose?</code>
例	<code>:COMMUNICATE:VERBOSE ON</code> <code>:COMMUNICATE:VERBOSE?</code> -> <code>:COMMUNICATE:HEADER ON</code>

**:COMMunicate:WAIT**

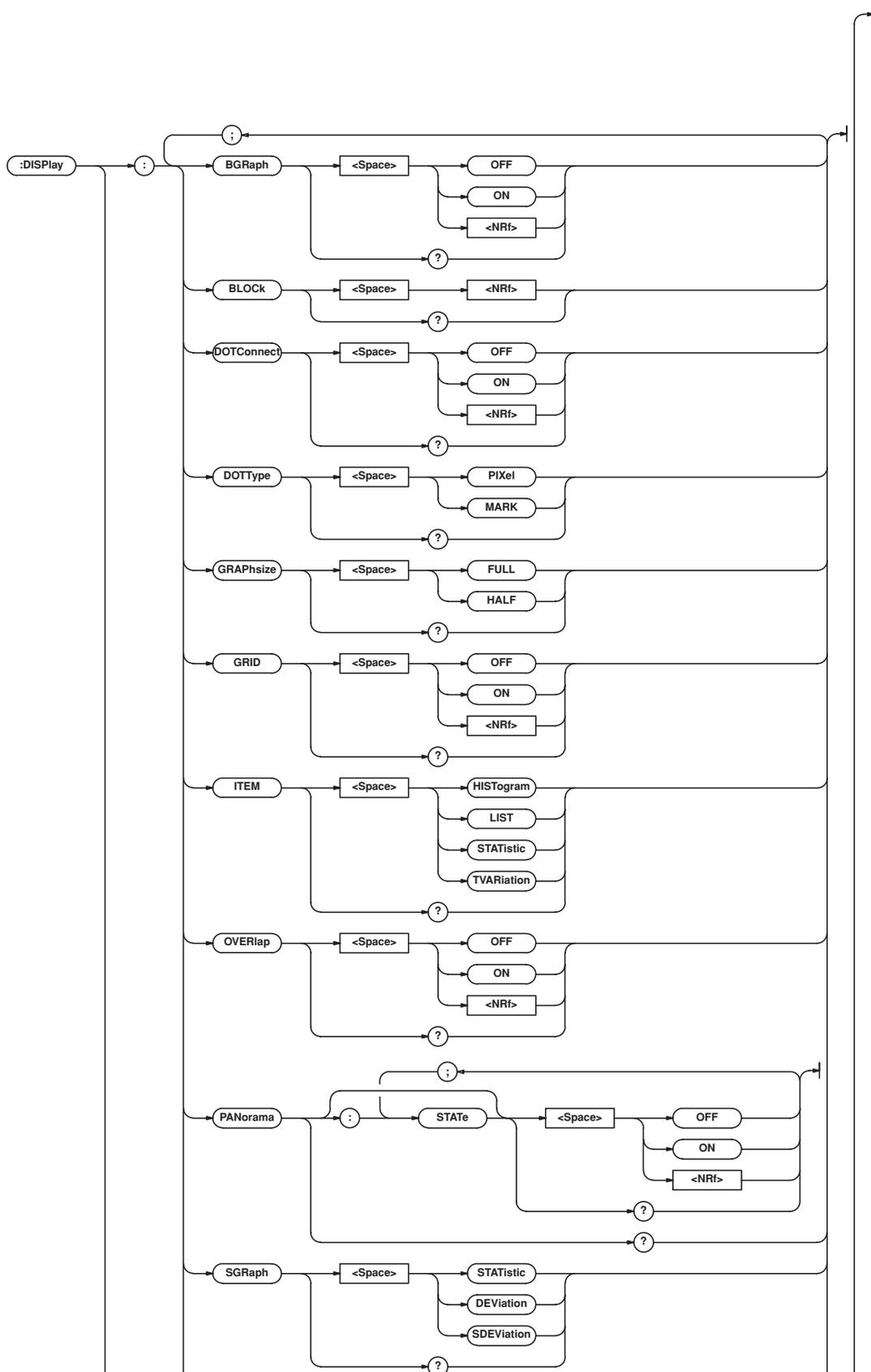
機能	指定された拡張イベントのどれかが発生するのを待ちます。
構文	<code>:COMMunicate:WAIT {&lt;Register&gt;}</code> <Register>=0~65535(拡張イベントレジスタ, 5-4ページ参照)
例	<code>:COMMUNICATE:WAIT 65535</code>
解説	「:COMMunicate:WAIT」を使った同期のとり方については、3-7ページを参照してください。

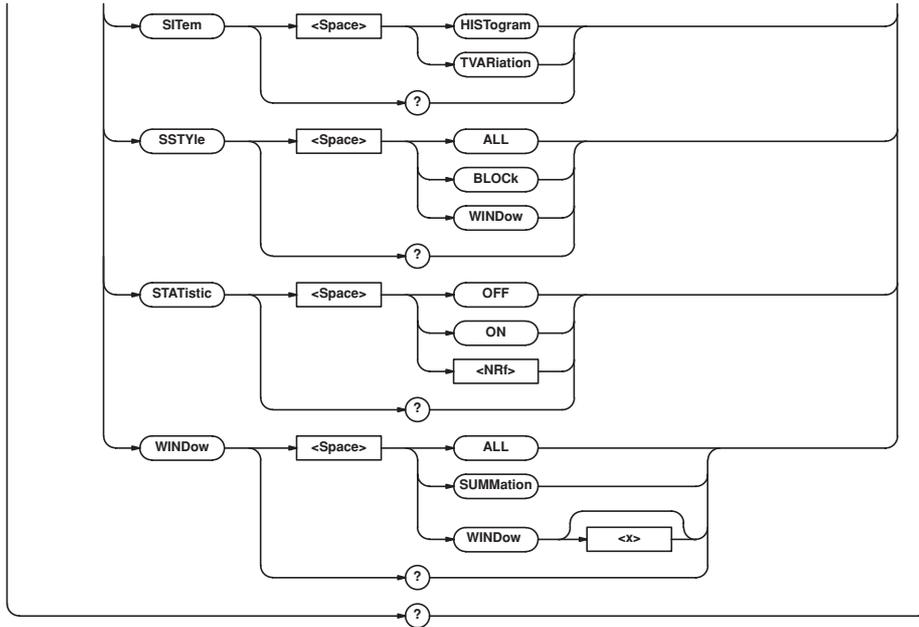
**:COMMunicate:WAIT?**

機能	指定された拡張イベントのどれかが発生したときに応答を作成します。
構文	<code>:COMMunicate:WAIT? {&lt;Register&gt;}</code> <Register>=0~65535(拡張イベントレジスタ, 5-4ページ参照)
例	<code>:COMMUNICATE:WAIT? 65535 -&gt; 1</code>

### 4.5 DISPlayグループ

表示の設定に関するグループです。フロントパネルのDISPLAYキーと同じ設定，および設定内容の問い合わせができます。



**:DISPlay?**

機能 表示に関する全設定を問い合わせます。

構文 :DISPlay?

例 :DISPLAY?

```
-> :DISPLAY:GRAPHSize FULL;
ITEM HISTOGRAM;PANORAMA:STATE 1;:
DISPLAY:STATISTIC 1
```

**:DISPlay:BGRaph**

機能 両パルス幅/両エッジ測定時でBOTH Graph表示のON/OFFを設定/問い合わせします。

構文 :DISPlay:BGRaph {<Boolean>}

例 :DISPLAY:BGRAPH OFF

:DISPLAY:BGRAPH?

```
-> :DISPLAY:BGRAPH 0
```

解説

- ・ヒストグラム表示，両パルス幅/両エッジ測定で，「:CALCulation:POLarity」が「POSitive|NEGative|POSNeg」のとき，設定ができます。
- ・詳細は，本体ユーザーズマニュアルの7章を参照してください。

**:DISPlay:BLOCK**

機能 表示するブロックを設定/問い合わせします。

構文 :DISPlay:BLOCK {<NRf>}

:DISPLAY:BLOCK?

<NRf>=ブロック番号(0~ブロック数)

<NRf>=0は，ブロック全体を表示します。

例 :DISPLAY:BLOCK 1

:DISPLAY:BLOCK?

```
-> :DISPLAY:BLOCK 1
```

解説

- ・測定モードがタイムスタンプモードで，「:CALCulation:AREA」が「BLOCK」のとき，設定ができます。
- ・「:SAMPle:BLOCK[:STATe]」が「ON」のとき，設定ができます。

**:DISPlay:DOTConnect**

機能 タイムバリエーション表示におけるドット接続のON/OFFを設定/問い合わせします。

構文 :DISPlay:DOTConnect {<Boolean>}

例 :DISPLAY:DOTCONNECT ON

:DISPLAY:DOTCONNECT?

```
-> :DISPLAY:DOTCONNECT 1
```

**:DISPlay:DOTType**

機能 タイムバリエーション表示における測定点の表示方法を設定/問い合わせします。

構文 :DISPlay:DOTType {PIXel|MARK}

例 :DISPLAY:DOTTYPE PIXEL

:DISPLAY:DOTTYPE?

```
-> :DISPLAY:DOTTYPE PIXEL
```

**:DISPlay:GRAPHsize**

機能 グラフの表示の大きさを設定/問い合わせします。

構文 :DISPlay:GRAPHsize {FULL|HALF}

例 :DISPLAY:GRAPHSize FULL

:DISPLAY:GRAPHSize?

```
-> :DISPLAY:GRAPHSize FULL
```

**:DISPlay:GRID**

機能 タイムバリエーション表示におけるグリッド表示のON/OFFを設定/問い合わせします。

構文 :DISPlay:GRID {<Boolean>}

例 :DISPLAY:GRID ON

:DISPLAY:GRID?

```
-> :DISPLAY:GRID 1
```

## 4.5 DISPLAYグループ

### **:DISPLAY:ITEM**

機能 表示形式を設定/問い合わせします。

構文 **:DISPLAY:ITEM** {HISTogram|LIST|STATistic|TVARiation}

例 **:DISPLAY:ITEM?**  
**:DISPLAY:ITEM HISTOGRAM**  
**:DISPLAY:ITEM?**  
-> **:DISPLAY:ITEM HISTOGRAM**

解説

- 測定モードがハードウェアヒストグラムモードでは、「TVARiation」は設定できません。
- 測定モードが符号間干渉解析モードでは「HISTogram|LIST」だけ設定できます。

### **:DISPLAY:OVERlap**

機能 ヒストグラム表示における両パルス幅/両エッジ測定で、両ポラリティのヒストグラムを重ね描きするか(ON)否か(OFF)を設定/問い合わせします。

タイムバリエーション表示における測定1、測定2を波形を重ね描きするか(ON)否か(OFF)を設定/問い合わせします。

構文 **:DISPLAY:OVERlap** {<Boolean>}

例 **:DISPLAY:OVERlap ON**  
**:DISPLAY:OVERlap?**  
-> **:DISPLAY:OVERlap 1**

解説

- ヒストグラム表示、両パルス幅/両エッジ測定で、「:CALCulation:POLarity」が「POSitive|NEGative|POSNeg」のとき、設定ができます。
- タイムバリエーション表示、周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定するとき、設定ができます。
- 詳細は、本体ユーザーズマニュアルの7章を参照してください。

### **:DISPLAY:PANorama[:STATE]**

機能 パノラマ表示のON/OFFを設定/問い合わせします。

構文 **:DISPLAY:PANorama[:STATE]** {<Boolean>}

例 **:DISPLAY:PANorama:STATE?**  
**:DISPLAY:PANorama:STATE ON**  
**:DISPLAY:PANorama:STATE?**  
-> **:DISPLAY:PANorama:STATE 1**

解説 ヒストグラム/タイムバリエーション表示のとき、設定ができます。

### **:DISPLAY:SGRaph**

機能 マルチウインドウ/オートウインドウのAll表示のとき、画面下部に統計値(STATistic)、Deviation(DEViation)、標準偏差(SDEViation)のどれを表示するかを設定/問い合わせします。

構文 **:DISPLAY:SGRaph** {STATistic|DEViation|SDEViation}

例 **:DISPLAY:SGRaph?**  
**:DISPLAY:SGRAPH DEVIATION**  
**:DISPLAY:SGRAPH?**  
-> **:DISPLAY:SGRAPH DEVIATION**

解説 ヒストグラム表示、マルチウインドウ/オートウインドウで「:DISPLAY:WINDow」が「ALL」のとき、設定ができます。

### **:DISPLAY:SITem**

機能 統計値表示をする統計値のタイプを設定/問い合わせします。

構文 **:DISPLAY:SITem** {HISTogram|TVARiation}

例 **:DISPLAY:SITem HISTOGRAM**  
**:DISPLAY:SITem?**  
-> **:DISPLAY:SITem HISTOGRAM**

解説 測定モードがタイムスタンプモードで、統計値表示のとき、設定ができます。

### **:DISPLAY:SSTYLE**

機能 統計値表示の表示形式を設定/問い合わせします。

構文 **:DISPLAY:SSTYLE** {ALL|BLOCK|WINDow}

例 **:DISPLAY:SSTYLE ALL**  
**:DISPLAY:SSTYLE?**  
-> **:DISPLAY:SSTYLE ALL**

解説

- 測定モードがハードウェアヒストグラムモードで、マルチウインドウ/オートウインドウのとき、「ALL|WINDow」が設定できます。
- 測定モードがタイムスタンプモードで、マルチウインドウ/オートウインドウで、「:DISPLAY:SITem」が「HISTogram」のとき、「ALL|WINDow」が設定できます。
- 測定モードがタイムスタンプモードで、「:SAMPLE:BLOCK:STATE」が「ON」、 「:DISPLAY:SITem」が「TVARiation」、 「:CALCulation:AREA」が「BLOCK」のとき、「ALL|BLOCK」が設定できます。

**:DISPlay:STATistic**

機能 ヒストグラム/タイムバリエーション表示時の統計値表示のON/OFFを設定/問い合わせします。

構文 :DISPlay:STATistic {<Boolean>}  
:DISPlay:STATistic?

例 :DISPlay:STATistic ON  
:DISPlay:STATistic?  
-> :DISPlay:STATistic 1

**:DISPlay:WINDow**

機能 表示するウインドウを設定/問い合わせします。

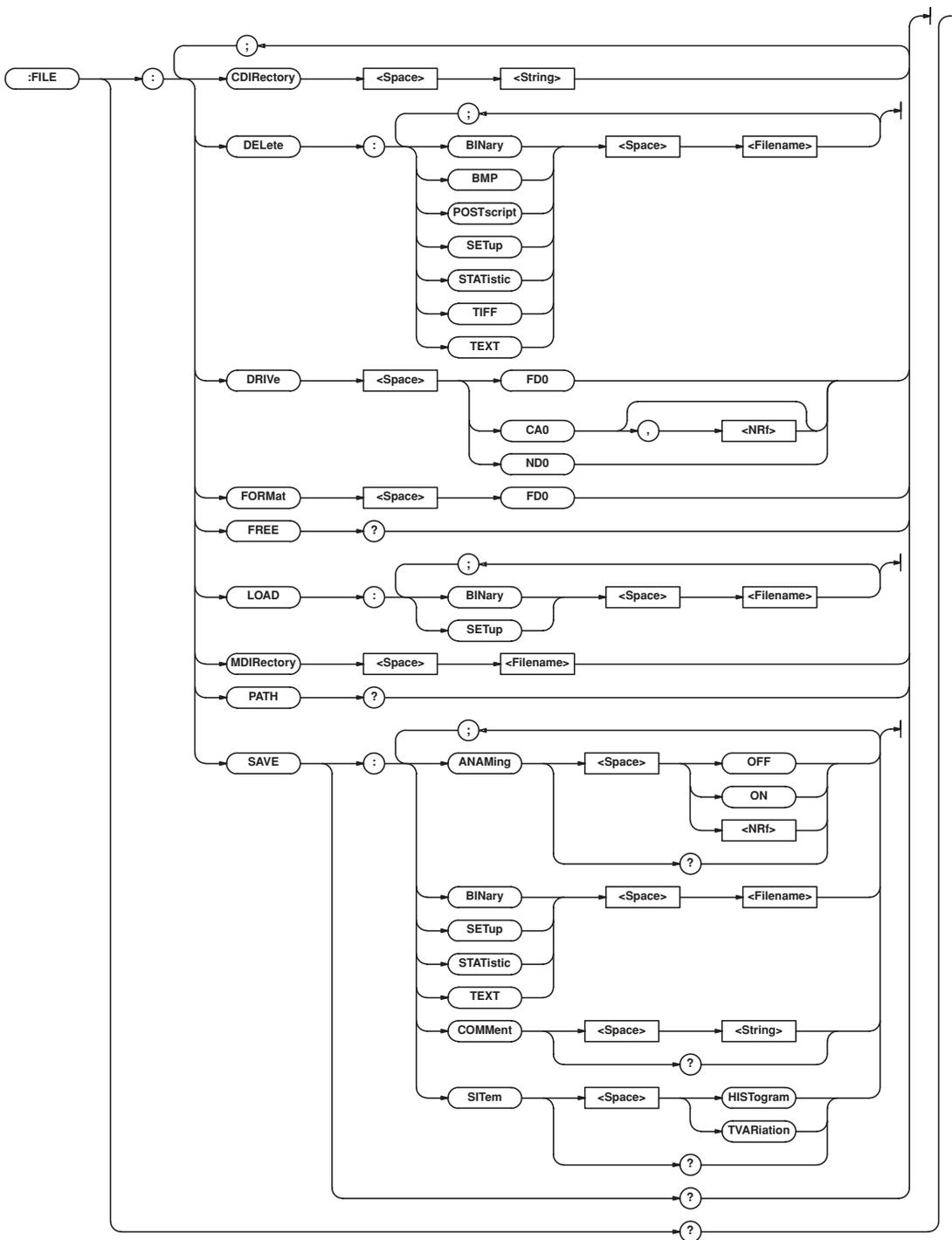
構文 :DISPlay:WINDow {ALL|SUMMation|  
WINDow<x>}  
:DISPlay:WINDow?  
<x>=ウインドウ番号(1~ウインドウ数)

例 :DISPlay:WINDow ALL  
:DISPlay:WINDow?  
-> :DISPlay:WINDow ALL

解説 ヒストグラム表示で、マルチウインドウ/オートウインドウのとき、設定ができます。

### 4.6 FILEグループ

ファイル操作に関するグループです。SHIFT+UTILITYキーと同じ設定・実行、および設定内容の問い合わせができます。



**:FILE?**

機能 ファイルに関する全設定を問い合わせます。  
 構文 **:FILE?**  
 例 **:FILE?**  
 -> **:FILE:SAVE:ANAMING 1;**  
**COMMENT "" ;SITEM TVARIATION**

**:FILE:CDIRECTORY**

機能 カレントディレクトリを変更します。  
 構文 **:FILE:CDIRECTORY {<文字列>}**  
 例 **:FILE:CDIRECTORY "NO\_1"**

**:FILE:DELeTe: {BINArY | BMP | POSTscript | SETUp | STATistic | TIFF | TEXT}**

機能 各種データのファイルを消去します。

構文 **:FILE:DELeTe: {BINArY | BMP | POSTscript | SETUp | STATistic | TIFF | TEXT} {<文字列>}**  
 <文字列>="ファイル名"(8文字以内、本体ユーザーマニュアル参照)  
 BINArY: バイナリ測定データファイル  
 BMP: BMP画面イメージファイル  
 POSTscript: ポストスクリプト画面イメージファイル  
 SETUp: 設定データファイル  
 STATistic: 統計データファイル  
 TIFF: TIFF画面イメージファイル  
 TEXT: テキスト測定データファイル  
 例 (以下は、設定データについての例です。)

解説 **:FILE:DELeTe:SETUP "TRASH"**  
 消去対象の記憶メディアは、「:FILE:DRIVE」で指定します。

**:FILE:DRIVE**

機能 対象ドライブを設定します。

構文 **:FILE:DRIVE {FD0 | CA0 [, <NRf>] | ND0}**  
 FD0: フロッピーディスクドライブ  
 CA0: PCカードドライブ  
 <NRf>=パーティション(0~3)  
 ND0: ネットワークドライブ

例 **:FILE:DRIVE FD0**  
 解説  
 ・PCカードドライブ/ネットワークドライブは、これらのオプションがあるモデルだけで設定可能です。  
 ・パーティションを設定していないPCカードの場合、<NRf>は省略してください。

**:FILE:FORMat**

機能 フロッピーディスクの初期化を実行します。

構文 **:FILE:FORMat {FD0}**

例 **:FILE:FORMat FD0**

**:FILE:FRee?**

機能 ドライブの空き容量をバイト数で問い合わせます。

構文 **:FILE:FRee?**

例 **:FILE:FRee?**  
 -> **:FILE:FRee 163840**

解説  
 ・対象記憶メディアは、「:FILE:DRIVE」で指定します。  
 ・対象記憶メディアがネットワークドライブのときは、問い合わせできません。

**:FILE:LOAD: {BINArY | SETUp}**

機能 各種データ呼び出します。

構文 **:FILE:LOAD: {BINArY | SETUp} {<文字列>}**  
 <文字列>="ファイル名"(8文字以内、本体ユーザーマニュアル参照)

BINArY: バイナリ測定データファイル

SETUp: 設定データファイル

例 (以下は、バイナリ測定データファイルの例です。)

**:FILE:LOAD:BINArY "TARGET"**

**:FILE:MDIRectory**

機能 ディレクトリを作成します。

構文 **:FILE:MDIRectory <文字列>**  
 <文字列>="ファイル名"(8文字以内、本体ユーザーマニュアル参照)

例 **:FILE:MDIRectory "NEW"**

**:FILE:PATH?**

機能 カレントディレクトリを問い合わせます。

構文 **:FILE:PATH?**

例 **:FILE:PATH?**  
 -> **:FILE:PATH "Path=FD0"**

**:FILE:SAVe?**

機能 ファイルの保存に関する全設定を問い合わせます。

構文 **:FILE:SAVe?**

例 **:FILE:SAVe?**  
 -> **:FILE:SAVe:ANAMing 0;**  
**COMMENT "SAMPLE"**

**:FILE:SAVe:ANAMing**

機能 保存時のファイル名の自動生成のON/OFFを設定/問い合わせします。

構文 **:FILE:SAVe:ANAMing {<Boolean>}**

**:FILE:SAVe:ANAMing?**

例 **:FILE:SAVe:ANAMing ON**

**:FILE:SAVe:ANAMing?**

-> **:FILE:SAVe:ANAMing 1**

## 4.6 FILEグループ

### **:FILE:SAVE:{BINary|SETup|STATistic|TEXT}**

機能 各種データを保存します。

構文 **:FILE:SAVE:{BINary|SETup|STATistic|TEXT} {<文字列>}**  
<文字列>="ファイル名"(8文字以内、本体ユーザーズマニュアル参照)  
BINary：バイナリ測定データファイル  
SETup：設定データファイル  
STATistic：統計データファイル  
TEXT：テキスト測定データファイル

例 (以下は、統計データファイルの例です。)  
**:FILE:SAVE:STATISTIC "TARGET"**

解説 測定モードが、タイムスタンプモードのとき、統計データファイルの保存は、「**:FILE:SAVE:SITem**」で指定した統計データを保存します。

### **:FILE:SAVE:COMment**

機能 画面左上部のコメントを設定/問い合わせします。

構文 **:FILE:SAVE:COMment {<文字列>}**  
**:FILE:SAVE:COMment?**  
<文字列>="コメント"(25文字以内、本体ユーザーズマニュアル参照)

例 **:FILE:SAVE:COMMENT "SAMPLE"**  
**:FILE:SAVE:COMMENT?**  
-> **:FILE:SAVE:COMMENT "SAMPLE"**

### **:FILE:SAVE:SITem**

機能 保存する統計データファイルのタイプを設定/問い合わせします。

構文 **:FILE:SAVE:SITem {HISTogram|TVARiation}**  
**:FILE:SAVE:SITem?**

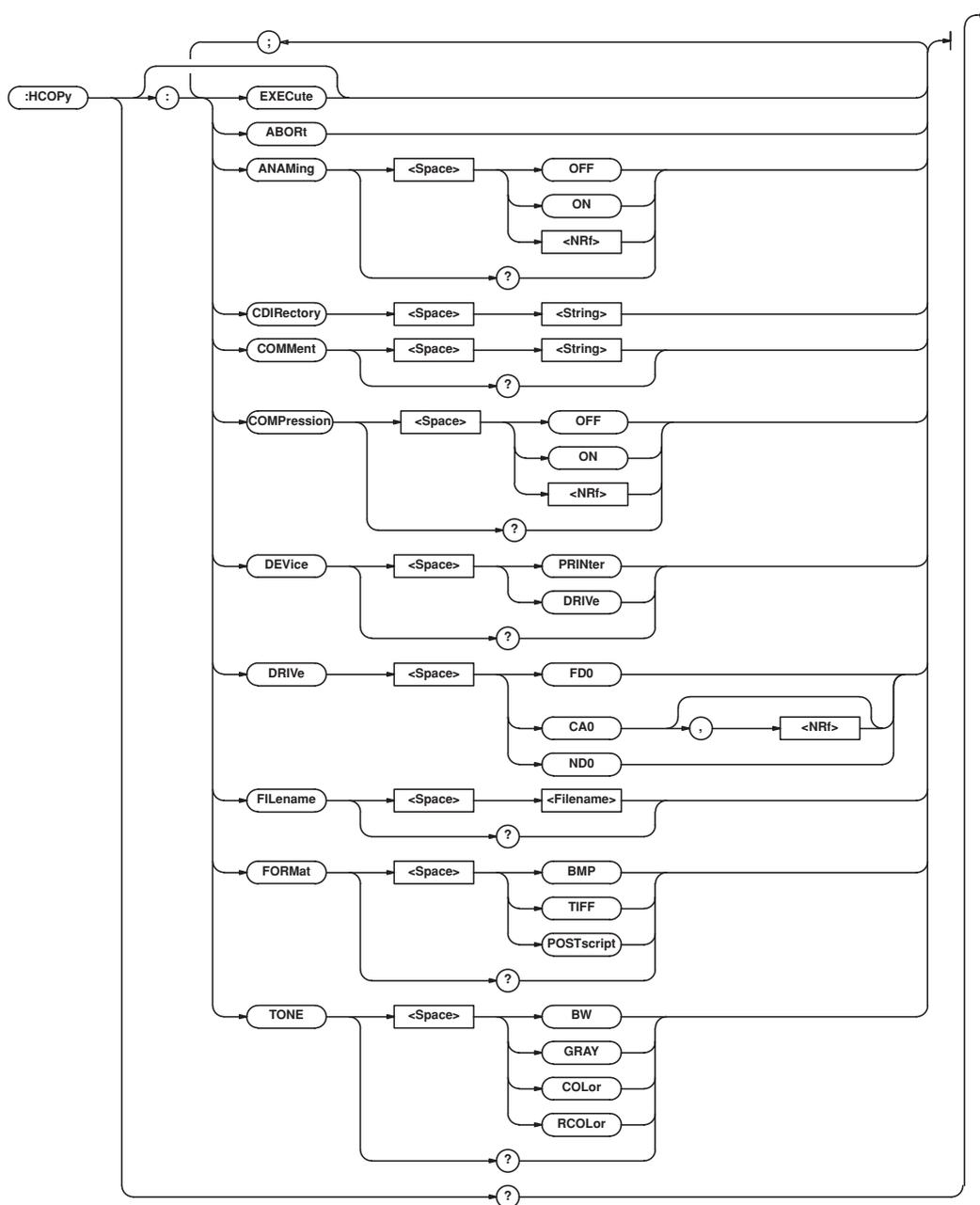
例 **:FILE:SAVE:SITEM TVARIATION**  
**:FILE:SAVE:SITEM?**  
-> **:FILE:SAVE:SITEM TVARIATION**

解説 測定モードが、タイムスタンプモードのとき、設定ができます。

## 4.7 HCOpyグループ

画面イメージデータの出力に関するグループです。

フロントパネルの(SHIFT+)COPYキーと同じ設定・実行、および設定内容の問い合わせができます。



### :HCOpy?

機能 画面イメージデータの出力に関する全設定を問い合わせます。

構文 :HCOpy?

例 :HCOpy?

```
-> :HCOpy:DEVICE PRINTER;
COMMENT "SAMPLE"
```

### :HCOpy:ABORt

機能 画面イメージのプリントアウトを中止します。

構文 :HCOpy:ABORt

例 :HCOpy:ABORt

### :HCOpy:ANAMing

機能 画面イメージ保存時のファイル名の自動生成のON/OFFを設定/問い合わせします。

構文 :HCOpy:ANAMing {<Boolean>}

:HCOpy:ANAMing?

例 :HCOpy:ANAMING ON

:HCOpy:ANAMING?

```
-> :HCOpy:ANAMING 1
```

## 4.7 HCOpyグループ

### **:HCOpy:COMment**

機能 画面左上部のコメントを設定/問い合わせします。

構文 **:HCOpy:COMment** {<文字列>}  
**:HCOpy:COMment?**  
<文字列>="コメント"(25文字以内, 本体ユーザーズマニュアル参照)

例 **:HCOpy:COMMENT "SAMPLE"**  
**:HCOpy:COMMENT?**  
-> **:HCOpy:COMMENT "SAMPLE"**

### **:HCOpy:COMPression**

機能 BMP形式で画面イメージを保存するときの圧縮のON/OFFを設定/問い合わせします。

構文 **:HCOpy:COMPression** {<Boolean>}  
**:HCOpy:COMPression?**

例 **:HCOpy:COMPRESSION ON**  
**:HCOpy:COMPRESSION?**  
-> **:HCOpy:COMPRESSION 1**

### **:HCOpy:DEvice**

機能 画面イメージの出力先を設定/問い合わせします。

構文 **:HCOpy:DEvice** {PRINter|DRive}  
**:HCOpy:DEvice?**

例 **:HCOpy:DEVICE PRINTER**  
**:HCOpy:DEVICE?**  
-> **:HCOpy:DEVICE PRINTER**

解説 「DRive」を設定したとき,  
「:HCOpy:DRive」で指定したドライブに出力されます。

### **:HCOpy:DRive**

機能 対象ドライブを設定します。

構文 **:HCOpy:DRive** {FD0|CA0[,<NRf>]|ND0}  
FD0: フロッピディスクドライブ  
CA0: PCカードドライブ  
<NRf>=パーティション(0~3)  
ND0: ネットワークドライブ

例 **:HCOpy:DRIVE FD0**

解説

- ・ 「:HCOpy:DEvice」が「DRive」のときに、出力先となるドライブを設定します。
- ・ PCカードドライブ、ネットワークドライブは、対応モデルのみ設定可能です。
- ・ パーティションを設定していないPCカードの場合、<NRf>は省略してください。

### **:HCOpy:CDIRectory**

機能 カレントディレクトリを変更します。

構文 **:HCOpy:CDIRectory** {<文字列>}  
例 **:HCOpy:CDIRECTORY "NO\_1"**

### **:HCOpy[:EXECute]**

機能 画面イメージのプリントアウトを実行します。

構文 **:HCOpy[:EXECute]**  
例 **:HCOpy:EXECUTE**

### **:HCOpy:FILEname**

機能 画面イメージの保存ファイル名を設定/問い合わせします。

構文 **:HCOpy:FILEname** {<文字列>}  
**:HCOpy:FILEname?**  
<文字列>="ファイル名"(8文字以内, 本体ユーザーズマニュアル参照)

例 **:HCOpy:FILENAME "KEEP"**  
**:HCOpy:FILENAME?**  
-> **:HCOpy:FILENAME "KEEP"**

### **:HCOpy:FORMat**

機能 画面イメージを保存するときのデータ形式を設定/問い合わせします。

構文 **:HCOpy:FORMat** {BMP|TIFF|POSTscript}  
**:HCOpy:FORMat?**

例 **:HCOpy:FORMAT BMP**  
**:HCOpy:FORMAT?**  
-> **:HCOpy:FORMAT BMP**

### **:HCOpy:TONE**

機能 画面イメージを保存するときの色/階調を設定/問い合わせします。

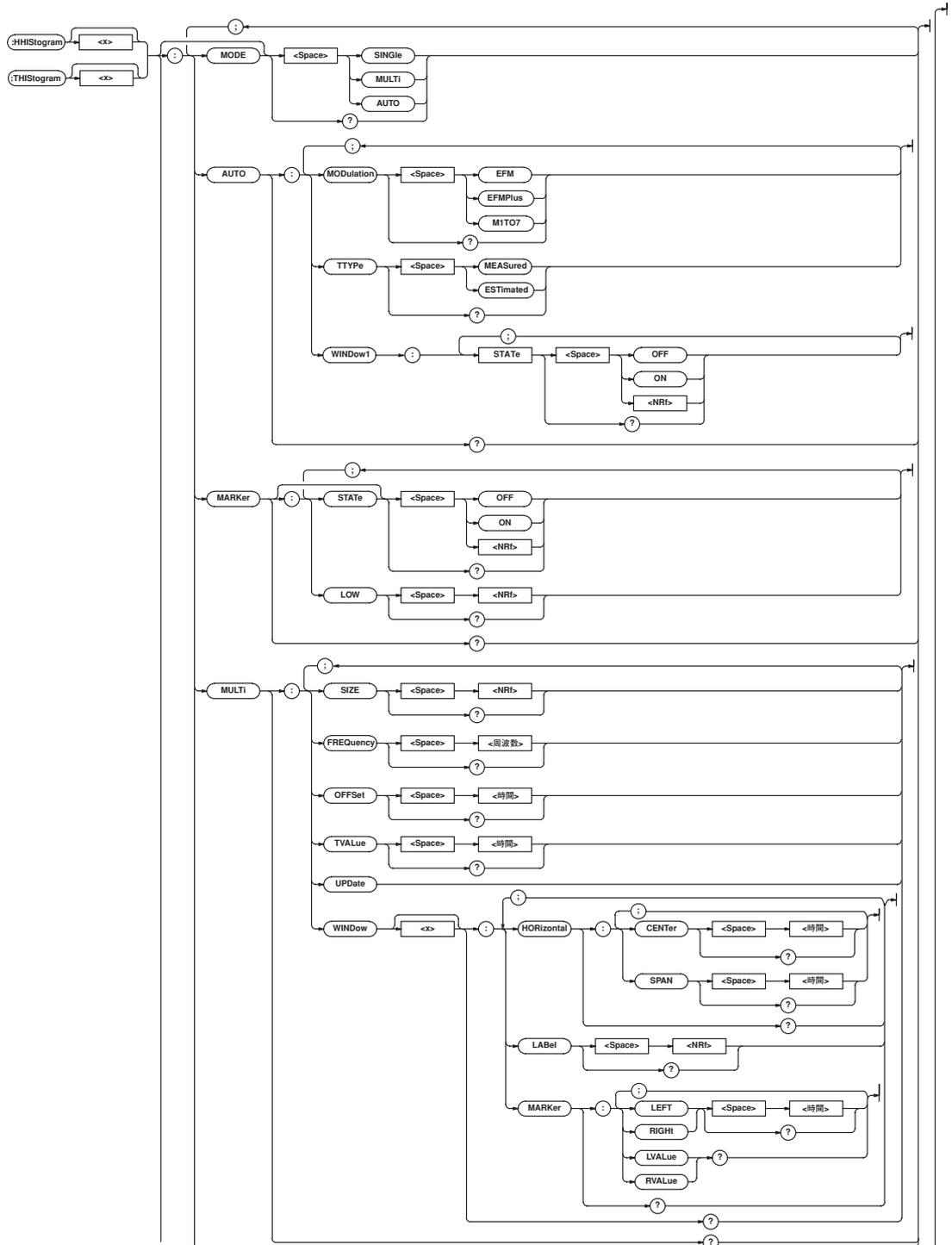
構文 **:HCOpy:TONE** {BW|GRAY|COLor|RCOLor}

例 **:HCOpy:TONE BW**  
**:HCOpy:TONE?**  
-> **:HCOpy:TONE BW**

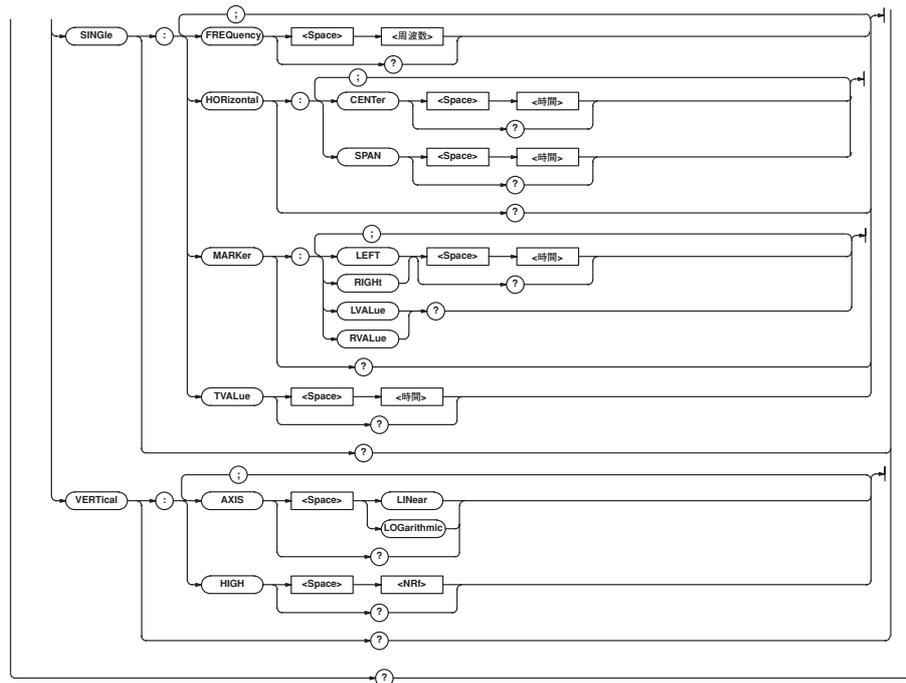
## 4.8 HHISistogram&lt;x&gt;, THISistogram&lt;x&gt;グループ

ハードウェアヒストグラム/タイムスタンプモードのヒストグラム表示に関するグループです。フロントパネルのDISPLAY, SCALE, MARKキーと同じ設定、および設定内容の問い合わせができます。

- ・ サンプルモードがハードウェアヒストグラムモードのときは、「:HHISistogram<x>」を使用してください。
- ・ サンプルモードがタイムスタンプモードのときは、「:THISistogram<x>」を使用してください。
- ・ 周期/AtoBタイムインターバル/パルス幅測定、または周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定で測定1に対するコマンドは、「HHISistogram1|THISistogram1」または、「HHISistogram|THISistogram」を使用してください。ただし、クエリは、「HHISistogram1|THISistogram1」となります。
- ・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定で、測定2に対するコマンドは、「HHISistogram2|THISistogram2」を使用してください。



#### 4.8 HHistogram<x>, THISTogram<x>グループ



##### : {HHistogram<x> | THISTogram<x>}?

機能 ハードウェアヒストグラム/タイムスタンプモードのヒストグラム表示に関する全設定を問い合わせます。

構文 : {HHistogram<x> | THISTogram<x>}?  
<x>=1~2

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM?
-> :HHISTOGRAM1:MARKER:LOW 0;
STATE 1;:HHISTOGRAM1:MODE SINGLE;
SINGLE:HORIZONTAL:
CENTER 150.000E-09;SPAN 0.30E-06;;
HHISTOGRAM1:SINGLE:MARKER:
LEFT 0.0E+00;RIGHT 300.000E-09;;
HHISTOGRAM1:SINGLE:
FREQUENCY 26.315E+06;;
TVALUE 38.000E-09;:HHISTOGRAM1:
VERTICAL:AXIS LOGARITHMIC;
HIGH 0.1E+03
```

##### : {HHistogram<x> | THISTogram<x>} :AUTO?

機能 オートウィンドウに関する全設定を問い合わせます。

構文 : {HHistogram<x> | THISTogram<x>} :  
AUTO?  
<x>=1~2

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:AUTO?
-> :HHISTOGRAM1:AUTO:
MODULATION EFM;TTYPE ESTIMATED;
WINDOW1:STATE 1
```

##### : {HHistogram<x> | THISTogram<x>} :AUTO:

###### MODulation

機能 オートウィンドウでの変調方式を設定/問い合わせします。

構文 : {HHistogram<x> | THISTogram<x>} :  
AUTO:MODulation {EFM|EFMPlus|M1T0?}  
: {HHistogram<x> | THISTogram<x>} :  
AUTO:MODulation?  
<x>=1~2

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:AUTO:MODULATION EFM
:HHISTOGRAM1:AUTO:MODULATION?
-> :HHISTOGRAM1:AUTO:MODULATION EFM
```

##### : {HHistogram<x> | THISTogram<x>} :AUTO:

###### TTYPE

機能 オートウィンドウでの定数Tの決め方を設定/問い合わせします。

構文 : {HHistogram<x> | THISTogram<x>} :  
AUTO:TTYPE {MEASURED|ESTIMATED}  
: {HHistogram<x> | THISTogram<x>} :  
AUTO:TTYPE?  
<x>=1~2

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:AUTO:TTYPE MEASURED
:HHISTOGRAM1:AUTO:TTYPE?
-> :HHISTOGRAM1:AUTO:TTYPE MEASURED
```

**:{HHIStogram<x>|THIStogram<x>}:AUTO:****WINDOW1:STATE**

機能 オートウインドウでのウインドウ1のON/OFFを設定/問い合わせします。

構文 :{HHIStogram<x>|THIStogram<x>}:  
AUTO:WINDOW1:STATE {<Boolean>}  
:{HHIStogram<x>|THIStogram<x>}:  
AUTO:WINDOW1:STATE?  
<x>=1~2

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:AUTO:WINDOW1:STATE ON
:HHISTOGRAM1:AUTO:WINDOW1:STATE?
-> :HHISTOGRAM1:AUTO:WINDOW1:
STATE ON
```

**:{HHIStogram<x>|THIStogram<x>}:MARKER?**

機能 マーカーに関する全設定を問い合わせます。

構文 :{HHIStogram<x>|THIStogram<x>}:  
MARKER?  
<x>=1~2

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:MARKER?
-> :HHISTOGRAM1:MARKER:LOW 0;
STATE 1
```

**:{HHIStogram<x>|THIStogram<x>}:MARKER:****LOW**

機能 ローマーカーの位置を設定/問い合わせします。

構文 :{HHIStogram<x>|THIStogram<x>}:  
MARKER:LOW {<NRf>}  
:{HHIStogram<x>|THIStogram<x>}:  
MARKER:LOW?  
<x>=1~2  
<NRf>=度数

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:MARKER:LOW 100
:HHISTOGRAM1:MARKER:LOW?
-> :HHISTOGRAM1:MARKER:LOW 100
```

**:{HHIStogram<x>|THIStogram<x>}:MARKER[:STATE]**

機能 マーカーのON/OFFを設定/問い合わせします。

構文 :{HHIStogram<x>|THIStogram<x>}:  
MARKER[:STATE] {<Boolean>}  
:{HHIStogram<x>|THIStogram<x>}:  
MARKER:STATE?  
<x>=1~2

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:MARKER:STATE ON
:HHISTOGRAM1:MARKER:STATE?
-> :HHISTOGRAM1:MARKER:STATE 1
```

**:{HHIStogram<x>|THIStogram<x>}[:MODE]**

機能 モードを設定/問い合わせします。

構文 :{HHIStogram<x>|THIStogram<x>}  
[:MODE] {SINGLE|MULTI|AUTO}  
:{HHIStogram<x>|THIStogram<x>}:  
MODE?  
<x>=1~2

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:MODE SINGLE
:HHISTOGRAM1:MODE?
-> :HHISTOGRAM1:MODE SINGLE
```

**:{HHIStogram<x>|THIStogram<x>}:MULTI?**

機能 マルチウインドウに関する全設定を問い合わせます。

構文 :{HHIStogram<x>|THIStogram<x>}:  
MULTI?  
<x>=1~2

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:MULTI?
-> :HHISTOGRAM1:MULTI:SIZE 1;
FREQUENCY 26.315E+06;
TVALUE 38.000E-09;
OFFSET 10.000-09E;:HHISTOGRAM1:
MULTI:WINDOW1:HORIZONTAL:
CENTER 114.000E-09;SPAN 0.060E-06;:
HHISTOGRAM1:MULTI:WINDOW1:LABEL 3;
MARKER:LEFT 95.000E-09;
RIGHT 133.000E-09
```

**:{HHIStogram<x>|THIStogram<x>}:MULTI:****FREQUENCY**

機能 周波数形式で定数Tの値を設定/問い合わせします。

構文 :{HHIStogram<x>|THIStogram<x>}:  
MULTI:FREQUENCY {<周波数>}  
:{HHIStogram<x>|THIStogram<x>}:  
MULTI:FREQUENCY?  
<x>=1~2

<周波数>=4MHz~1000MHz

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:MULTI:FREQUENCY 50MHZ
:HHISTOGRAM1:MULTI:FREQUENCY?
-> :HHISTOGRAM1:MULTI:
FREQUENCY 50.000E+06
```

解説 「:{HHIStogram<x>|THIStogram<x>}:  
MULTI:TVALUE」の周波数形式での設定/問い合わせです。

## 4.8 HHISTogram<x>, THISTogram<x>グループ

**:{HHISTogram<x>|THISTogram<x>}:MULTI:OFFSet**

機能 定数Tへのオフセット値を設定/問い合わせします。

構文 **:{HHISTogram<x>|THISTogram<x>}:MULTI:OFFSet {<時間>}**  
**:{HHISTogram<x>|THISTogram<x>}:MULTI:OFFSet?**  
<x>=1~2  
<時間>=-100ns~300ns(25psステップ)

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:MULTI:OFFSET 10ns
:HHISTOGRAM1:MULTI:OFFSET?
-> :HHISTOGRAM1:MULTI:OFFSET 10.000E-09
```

**:{HHISTogram<x>|THISTogram<x>}:MULTI:SIZE**

機能 ウィンドウ数を設定/問い合わせします。

構文 **:{HHISTogram<x>|THISTogram<x>}:MULTI:SIZE {<NRf>}**  
**:{HHISTogram<x>|THISTogram<x>}:MULTI:SIZE?**  
<x>=1~2  
<NRf>=ウィンドウ数(1~14)

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:MULTI:SIZE 10
:HHISTOGRAM1:MULTI:SIZE?
-> :HHISTOGRAM1:MULTI:SIZE 10
```

**:{HHISTogram<x>|THISTogram<x>}:MULTI:TVALue**

機能 定数Tの値を設定/問い合わせします。

構文 **:{HHISTogram<x>|THISTogram<x>}:MULTI:TVALue {<時間>}**  
**:{HHISTogram<x>|THISTogram<x>}:MULTI:TVALue?**  
<x>=1~2  
<時間>=1ns~250ns(25psステップ)

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:MULTI:TVALUE 100ns
:HHISTOGRAM1:MULTI:TVALUE?
-> :HHISTOGRAM1:MULTI:TVALUE 100.000E-09
```

**:{HHISTogram<x>|THISTogram<x>}:MULTI:UPDATE**

機能 定数Tの値、オフセット値をもとにウィンドウの設定を変更します。

構文 **:{HHISTogram<x>|THISTogram<x>}:MULTI:UPDATE**  
<x>=1~2

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM:MULTI:UPDATE
```

**:{HHISTogram<x>|THISTogram<x>}:MULTI:WINDOW<x>?**

機能 各ウィンドウに関する全設定を問い合わせます。

構文 **:{HHISTogram<x>|THISTogram<x>}:MULTI:WINDOW<x>?**  
HHISTogram<x>|THISTogram<x>の<x>=1~2  
WINDOW<x>の<x>=ウィンドウ番号(1~ウィンドウ数)

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:MULTI:WINDOW1?
-> :HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:CENTER 114.000E-09;
SPAN 0.060E-06;:HHISTOGRAM1:MULTI:
WINDOW1:LABEL 3;MARKER:
LEFT 95.000E-09;RIGHT 133.000E-09
```

**:{HHISTogram<x>|THISTogram<x>}:MULTI:WINDOW<x>:HORizontal?**

機能 各ウィンドウの水平軸(X軸)に関する全設定を問い合わせます。

構文 **:{HHISTogram<x>|THISTogram<x>}:MULTI:WINDOW<x>:HORizontal?**  
HHISTogram<x>|THISTogram<x>の<x>=1~2  
WINDOW<x>の<x>=ウィンドウ番号(1~ウィンドウ数)

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL?
-> :HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:CENTER 150.000E-09;
SPAN 0.3E-06
```

**:{HHISistogram<x>|THISistogram<x>}:MULTI:****WINDOW<x>:HORIZONTAL:CENTER**

機能 各ウインドウの水平軸(X軸)の中央位置を設定/問い合わせします。

構文 **:{HHISistogram<x>|THISistogram<x>}:  
WINDOW<x>:HORIZONTAL:CENTER <時間>  
:{HHISistogram<x>|THISistogram<x>}:  
WINDOW<x>:HORIZONTAL:CENTER?  
HHISistogram<x>|THISistogram<x>の<x>=1~  
2  
WINDOW<x>の<x>=ウインドウ番号(1~ウイン  
ドウ数)**

<時間>=-50ns~3.2μs(25psステップ)  
例 (以下は、ハードウェアヒストグラムモードにお  
ける例です。)

```
:HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:CENTER 150ns
:HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:CENTER?
-> :HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:CENTER 150.000E-09
```

**:{HHISistogram<x>|THISistogram<x>}:MULTI:****WINDOW<x>:HORIZONTAL:SPAN**

機能 各ウインドウの水平軸(X軸)の幅を設定/問い合わせします。

構文 **:{HHISistogram<x>|THISistogram<x>}:  
MULTI:WINDOW<x>:HORIZONTAL:SPAN {<時  
間>}  
:{HHISistogram<x>|THISistogram<x>}:  
MULTI:WINDOW<x>:HORIZONTAL:SPAN?  
HHISistogram<x>|THISistogram<x>の<x>=1~  
2  
WINDOW<x>の<x>=ウインドウ番号(1~ウイン  
ドウ数)**

<時間>=1.5ns, 3ns, 7.5ns, 15ns, 30ns,  
60ns, 150ns, 300ns, 600ns, 1.5μs,  
3μs, 6μs

例 (以下は、ハードウェアヒストグラムモードにお  
ける例です。)

```
:HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:SPAN 300ns
:HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:SPAN?
-> :HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:SPAN 0.3E-06
```

**:{HHISistogram<x>|THISistogram<x>}:MULTI:****WINDOW<x>:LABEL**

機能 各ウインドウのラベルを設定/問い合わせしま  
す。

構文 **:{HHISistogram<x>|THISistogram<x>}:  
MULTI:WINDOW<x>:LABEL {<NRF>}  
:{HHISistogram<x>|THISistogram<x>}:  
MULTI:WINDOW<x>:LABEL?  
HHISistogram<x>|THISistogram<x>の<x>=1~  
2  
WINDOW<x>の<x>=ウインドウ番号(1~ウイン  
ドウ数)**

<NRF>=1~16  
例 (以下は、ハードウェアヒストグラムモードにお  
ける例です。)

```
:HHISTOGRAM1:MULTI:WINDOW1:LABEL 3
:HHISTOGRAM1:MULTI:WINDOW1:LABEL?
-> :HHISTOGRAM1:MULTI:WINDOW1:
LABEL 3
```

**:{HHISistogram<x>|THISistogram<x>}:MULTI:****WINDOW<x>:MARKER?**

機能 各ウインドウの水平軸(X軸)マーカーに関する全  
設定を問い合わせます。

構文 **:{HHISistogram<x>|THISistogram<x>}:  
MULTI:WINDOW<x>:MARKER?  
HHISistogram<x>|THISistogram<x>の<x>=1~  
2  
WINDOW<x>の<x>=ウインドウ番号(1~ウイン  
ドウ数)**

例 (以下は、ハードウェアヒストグラムモードにお  
ける例です。)

```
:HHISTOGRAM1:MULTI:WINDOW1:MARKER?
-> :HHISTOGRAM1:MULTI:WINDOW1:
MARKER:LEFT 100.000E-09;
RIGHT 200.000E-09
```

#### 4.8 HHistogram<x>, THISTogram<x>グループ

**:{HHistogram<x>|THISTogram<x>}:MULTI:WINDOW<x>:MARKER:{LEFT|RIGHT}**

機能 各ウインドウのレフト/ライトマーカーの位置を設定/問い合わせします。

構文 **:{HHistogram<x>|THISTogram<x>}:MULTI:WINDOW<x>:MARKER:{LEFT|RIGHT}<時間>**  
**:{HHistogram<x>|THISTogram<x>}:MULTI:WINDOW<x>:MARKER:{LEFT|RIGHT}?**  
**HHistogram<x>|THISTogram<x>の<x>=1~2**  
**WINDOW<x>の<x>=ウインドウ番号(1~ウインドウ数)**  
**<時間>=マーカー位置(25psステップ)**

例 (以下は、ハードウェアヒストグラムモード、レフトマーカーにおける例です。)

```

:HHISTOGRAM1:MULTI:WINDOW1:MARKER:
LEFT 100ns
:HHISTOGRAM1:MULTI:WINDOW1:MARKER:
LEFT?
-> :HHISTOGRAM1:MULTI:WINDOW1:
MARKER:LEFT 100.000E-09

```

**:{HHistogram<x>|THISTogram<x>}:MULTI:WINDOW<x>:MARKER:{LVALUE|RVALUE}?**

機能 各ウインドウの各マーカーの位置の度数を問い合わせます。

構文 **:{HHistogram<x>|THISTogram<x>}:MULTI:WINDOW<x>:MARKER:{LVALUE|RVALUE}? [{POSITIVE|NEGATIVE}]**  
**HHistogram<x>|THISTogram<x>の<x>=1~2**  
**WINDOW<x>の<x>=ウインドウ番号(1~ウインドウ数)**

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```

:HHISTOGRAM1:MULTI:WINDOW1:MARKER:
LVALUE?
-> :HHISTOGRAM1:MULTI:WINDOW1:
MARKER:LVALUE 1234

```

解説 両パルス幅/両エッジ測定で、「CALCulation:POLarity」が「POSneg」のとき、「POSitive|NEGative」を指定してください。

**:{HHistogram<x>|THISTogram<x>}:SINGLE?**

機能 シングルウインドウに関する全設定を問い合わせます。

構文 **:{HHistogram<x>|THISTogram<x>}:SINGLE?**  
**<x>=1~2**

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```

:HHISTOGRAM1:SINGLE?
-> :HHISTOGRAM1:SINGLE:HORIZONTAL:
CENTER 150.000E-09;SPAN 0.30E-06;;
HHISTOGRAM1:SINGLE:MARKER:
LEFT 0.0E+00;RIGHT 300.000E-09;
:HHISTOGRAM1:SINGLE:
FREQUENCY 26.315E+06;
TVALUE 38.000E-09

```

**:{HHistogram<x>|THISTogram<x>}:SINGLE:FREQUENCY**

機能 周波数形式で定数Tの値を設定/問い合わせします。

構文 **:{HHistogram<x>|THISTogram<x>}:SINGLE:FREQUENCY {<周波数>}**  
**:{HHistogram<x>|THISTogram<x>}:SINGLE:FREQUENCY?**  
**<x>=1~2**  
**<周波数>=4MHz~1000MHz**

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```

:HHISTOGRAM1:SINGLE:FREQUENCY 50MHz
:HHISTOGRAM1:SINGLE:FREQUENCY?
-> :HHISTOGRAM1:SINGLE:
FREQUENCY 50.000E+06

```

解説 「:{HHistogram<x>|THISTogram<x>}:SINGLE:TVALUE」の周波数形式での設定/問い合わせです。

**:{HHistogram<x>|THISTogram<x>}:SINGLE:HORIZONTAL?**

機能 シングルウインドウの水平軸(X軸)に関する全設定を問い合わせます。

構文 **:{HHistogram<x>|THISTogram<x>}:SINGLE:HORIZONTAL?**  
**<x>=1~2**

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```

:HHISTOGRAM1:SINGLE:HORIZONTAL?
-> :HHISTOGRAM1:SINGLE:HORIZONTAL:
CENTER 150.000E-06;SPAN 0.3E-6

```

**:{HHISogram<x>|THISogram<x>}:SINGLE:****HORizontal:CENTer**

機能 シングルウインドウの水平軸(X軸)の中央位置を設定/問い合わせします。

構文 **:{HHISogram<x>|THISogram<x>}:SINGLE:HORizontal:CENTer {<時間>}**  
**:{HHISogram<x>|THISogram<x>}:SINGLE:HORizontal:CENTer?**  
 <x>=1~2

●ハードウェアヒストグラムモードの場合  
 <時間>=-50ns~3.2μs(25psステップ)

●タイムスタンプモードの場合  
 <時間>=-50ns~20ms(25psステップ)

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:SINGLE:HORIZONTAL:
CENTER 150ns
:HHISTOGRAM1:SINGLE:HORIZONTAL:
CENTER?
-> :HHISTOGRAM1:SINGLE:HORIZONTAL:
CENTER 150.000E-09
```

**:{HHISogram<x>|THISogram<x>}:SINGLE:****HORizontal:SPAN**

機能 シングルウインドウの水平軸(X軸)の幅を設定/問い合わせします。

構文 **:{HHISogram<x>|THISogram<x>}:SINGLE:HORizontal:SPAN {<時間>}**  
**:{HHISogram<x>|THISogram<x>}:SINGLE:HORizontal:SPAN?**  
 <x>=1~2

●ハードウェアヒストグラムモードの場合  
 <時間>=1.5ns, 3ns, 7.5ns, 15ns, 30ns, 60ns, 150ns, 300ns, 600ns, 1.5μs, 3μs, 6μs

●タイムスタンプモードの場合  
 <時間>=1.5ns, 3ns, 7.5ns, 15ns, 30ns, 60ns, 150ns, 300ns, 600ns, 1.5μs, 3μs, 6μs, 15μs, 30μs, 60μs, 150μs, 300μs, 600μs, 1.5ms, 3ms, 6ms, 15ms, 30ms

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:SINGLE:HORIZONTAL:
SPAN 300ns
:HHISTOGRAM1:SINGLE:HORIZONTAL:
SPAN?
-> :HHISTOGRAM1:SINGLE:HORIZONTAL:
SPAN 0.3E-06
```

**:{HHISogram<x>|THISogram<x>}:SINGLE:****MARKer?**

機能 水平軸(X軸)マーカーに関する全設定を問い合わせします。

構文 **:{HHISogram<x>|THISogram<x>}:SINGLE:MARKer?**  
 <x>=1~2

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:SINGLE:MARKER?
-> :HHISTOGRAM1:SINGLE:MARKER:
LEFT 100.000E-09;RIGHT 200.000E-09
```

**:{HHISogram<x>|THISogram<x>}:SINGLE:****MARKer:{LEFT|RIGHT}**

機能 レフト/ライトマーカーの位置を設定/問い合わせします。

構文 **:{HHISogram<x>|THISogram<x>}:SINGLE:MARKer:{LEFT|RIGHT} {<時間>}**  
**:{HHISogram<x>|THISogram<x>}:SINGLE:MARKer:{LEFT|RIGHT}?**  
 <x>=1~2

<時間>=マーカー位置(25psステップ)

例 (以下は、ハードウェアヒストグラムモード、レフトマーカーにおける例です。)

```
:HHISTOGRAM1:SINGLE:MARKER:
LEFT 100ns
:HHISTOGRAM1:SINGLE:MARKER:LEFT?
-> :HHISTOGRAM1:SINGLE:MARKER:
LEFT 100.000E-09
```

**:{HHISogram<x>|THISogram<x>}:SINGLE:****MARKer:{LVALue|RVALue}?**

機能 レフト/ライトマーカーの位置の度数を問い合わせします。

構文 **:{HHISogram<x>|THISogram<x>}:SINGLE:MARKer:{LVALue|RVALue}?**  
**[{POSitive|NEGative}]**  
 <x>=1~2

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

```
:HHISTOGRAM1:SINGLE:MARKER:LVALE?
-> :HHISTOGRAM1:SINGLE:MARKER:
LVALE 1234
```

解説 両パルス幅/両エッジ測定で、「:CALCulation:POLarity」が「POSNeg」のとき、「POSitive|NEGative」を指定してください。

#### 4.8 HHistogram<x>, THISTogram<x>グループ

##### **:{HHistogram<x>|THISTogram<x>}:SINGLE:TVALue**

機能 定数T値を設定/問い合わせします。

構文 **:{HHistogram<x>|THISTogram<x>}:SINGLE:TVALue {<時間>}**  
**:{HHistogram<x>|THISTogram<x>}:SINGLE:TVALue?**

<x>=1~2

<時間>=1ns~250ns(25psステップ)

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

**:HHISTOGRAM1:SINGLE:TVALUE 100ns**  
**:HHISTOGRAM1:SINGLE:TVALUE?**

##### **:{HHistogram<x>|THISTogram<x>}:VERTICAL?**

機能 ハードウェアヒストグラムの垂直軸(Y軸)に関する全設定を問い合わせます。

構文 **:{HHistogram<x>|THISTogram<x>}:VERTICAL?**

<x>=1~2

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

**:HHISTOGRAM1:VERTICAL?**  
**-> :HHISTOGRAM1:VERTICAL:**  
**AXIS LINEAR;HIGH 1.0E+03**

##### **:{HHistogram<x>|THISTogram<x>}:VERTICAL:**

###### **AXIS**

機能 垂直軸(Y軸)スケールの種類を設定/問い合わせします。

構文 **:{HHistogram<x>|THISTogram<x>}:VERTICAL:AXIS {LINEAR|LOGarithmic}**  
**:{HHistogram<x>|THISTogram<x>}:VERTICAL:AXIS?**

<x>=1~2

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

**:HHISTOGRAM1:VERTICAL:AXIS LINEAR**  
**:HHISTOGRAM1:VERTICAL:AXIS?**  
**-> :HHISTOGRAM1:VERTICAL:**  
**AXIS LINEAR**

##### **:{HHistogram<x>|THISTogram<x>}:VERTICAL:**

###### **HIGH**

機能 垂直軸(Y軸)スケールの上限を設定/問い合わせします。

構文 **:{HHistogram<x>|THISTogram<x>}:VERTICAL:HIGH {<NRf>}**  
**:{HHistogram<x>|THISTogram<x>}:VERTICAL:HIGH?**

<x>=1~2

<NRf>=上限の度数

リニアスケール時

10, 20, 40, 100, 200, 400, 1000, 400000, 1E+6, 1E+7, 1E+8, 1E+9

対数スケール時

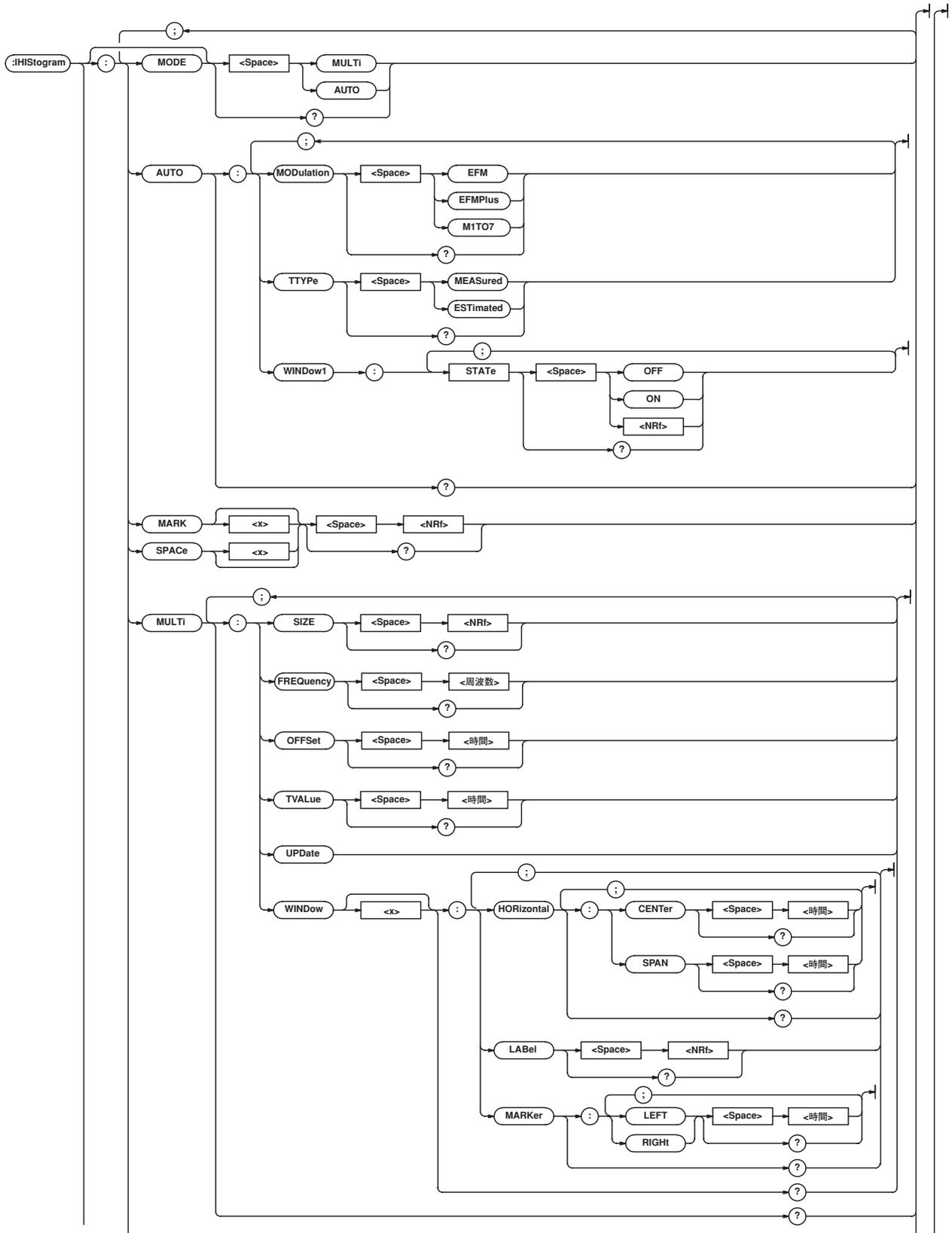
1E+1, 1E+2, ……., 1E+9

例 (以下は、ハードウェアヒストグラムモードにおける例です。)

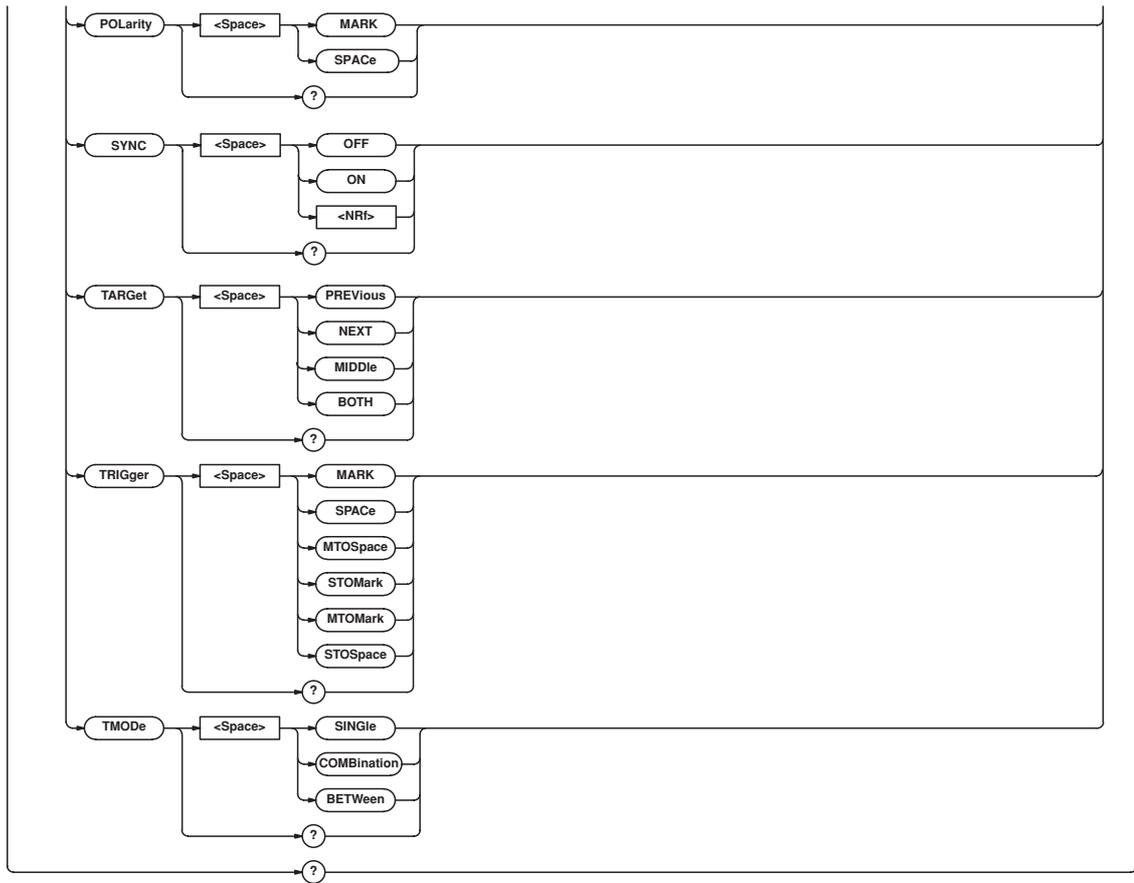
**:HHISTOGRAM1:VERTICAL:HIGH 1000**  
**:HHISTOGRAM1:VERTICAL:HIGH?**  
**-> :HHISTOGRAM1:VERTICAL:**  
**HIGH 1.0E+03**

### 4.9 IHistogramグループ

符号間干渉解析モードのヒストグラム表示の設定に関するグループです。フロントパネルのDISPLAY, SCALE, MARKキーと同じ設定、および設定内容の問い合わせができます。



#### 4.9 IHistogramグループ



**: IHistogram?**

機能 符号間干渉解析モードのヒストグラム表示に関する全設定を問い合わせます。

構文 : IHistogram?

例 : IHISTGRAM?

```
-> : IHISTOGRAM: MARK1 4; MODE MULTI;
MULTI: SIZE 9; FREQUENCY 26.315E+06;
TVALUE 38.000E-09;
OFFSET 10.000E-09; WINDOW1:
HORIZONTAL: CENTER 86.000E-09;
SPAN 0.060E-06; : IHISTOGRAM: MULTI:
WINDOW1: LABEL 2; MARKER:
LEFT 67.000E-09; RIGHT 105.000E-09; :
IHISTOGRAM: MULTI: WINDOW2: HORIZONTAL:
CENTER 124.000E-09; SPAN 0.060E-06; :
IHISTOGRAM: MULTI: WINDOW2: LABEL 3;
MARKER: LEFT 105.000E-09;
RIGHT 143.000E-09; : IHISTOGRAM:
MULTI: WINDOW3: HORIZONTAL:
CENTER 162.000E-09; SPAN 0.060E-06; :
IHISTOGRAM: MULTI: WINDOW3: LABEL 4;
MARKER: LEFT 143.000E-09;
RIGHT 181.000E-09; : IHISTOGRAM:
MULTI: WINDOW4: HORIZONTAL:
CENTER 200.000E-09; SPAN 0.060E-06; :
IHISTOGRAM: MULTI: WINDOW4: LABEL 5;
MARKER: LEFT 181.000E-09;
RIGHT 219.000E-09; : IHISTOGRAM:
MULTI: WINDOW5: HORIZONTAL:
CENTER 238.000E-09; SPAN 0.060E-06; :
IHISTOGRAM: MULTI: WINDOW5: LABEL 6;
MARKER: LEFT 219.000E-09;
RIGHT 257.000E-09; : IHISTOGRAM:
MULTI: WINDOW6: HORIZONTAL:
CENTER 276.000E-09; SPAN 0.060E-06; :
IHISTOGRAM: MULTI: WINDOW6: LABEL 7;
MARKER: LEFT 257.000E-09;
RIGHT 295.000E-09; : IHISTOGRAM:
MULTI: WINDOW7: HORIZONTAL:
CENTER 314.000E-09; SPAN 0.060E-06; :
IHISTOGRAM: MULTI: WINDOW7: LABEL 8;
MARKER: LEFT 295.000E-09;
RIGHT 333.000E-09; : IHISTOGRAM:
MULTI: WINDOW8: HORIZONTAL:
CENTER 352.000E-09; SPAN 0.060E-06; :
IHISTOGRAM: MULTI: WINDOW8: LABEL 9;
MARKER: LEFT 333.000E-09;
RIGHT 371.000E-09; : IHISTOGRAM:
MULTI: WINDOW9: HORIZONTAL:
CENTER 390.000E-09; SPAN 0.060E-06; :
IHISTOGRAM: MULTI: WINDOW9: LABEL 10;
MARKER: LEFT 371.000E-09;
RIGHT 409.000E-09; : IHISTOGRAM:
POLARITY SPACE; TADJUST: STATE 0; :
IHISTOGRAM: TARGET BOTH;
TRIGGER MARK; TMODE SINGLE
```

**: IHistogram: AUTO?**

機能 オートウインドウに関する全設定を問い合わせます。

構文 : IHistogram: AUTO?

例 : IHISTOGRAM: AUTO?

```
-> : IHISTOGRAM: AUTO: MODULATION
EFMPLUS; TTYPE ESTIMATED; WINDOW1:
STATE 1
```

**: IHistogram: AUTO: MODulation**

機能 オートウインドウでの変調方式を設定/問い合わせします。

構文 : IHistogram: AUTO: MODulation

```
{EFM|EFMplus|M1T07}
```

```
: IHistogram: AUTO: MODulation?
```

例 : IHISTOGRAM: AUTO: MODULATION EFMPLUS

```
: IHISTOGRAM: AUTO: MODULATION?
```

```
-> : IHISTOGRAM: AUTO:
```

```
MODULATION EFMPLUS
```

**: IHistogram: AUTO: TTYPE**

機能 オートウインドウでの定数Tの決め方を設定/問い合わせします。

構文 : IHistogram: AUTO: TTYPE

```
{MEASured|ESTimated}
```

```
: IHistogram: AUTO: TTYPE?
```

例 : IHISTOGRAM: AUTO: TTYPE MEASURED

```
: IHISTOGRAM: AUTO: TTYPE?
```

```
-> : IHISTOGRAM: AUTO: TTYPE MEASURED
```

**: IHistogram: AUTO: WINDOW1: STATE**

機能 オートウインドウでのウインドウ1のON/OFFを設定/問い合わせします。

構文 : IHistogram: AUTO: WINDOW1: STATE

```
{<Boolean>}
```

```
: IHistogram: AUTO: WINDOW1: STATE?
```

例 : IHISTOGRAM: AUTO: WINDOW1: STATE ON

```
: IHISTOGRAM: AUTO: WINDOW1: STATE?
```

```
-> : IHISTOGRAM: AUTO: WINDOW1: STATE
```

```
ON
```

## 4.9 IHistogramグループ

### **:IHistogram: {MARK<x> | SPACe<x>}**

機能 マークを設定/問い合わせします。  
 構文 **:IHistogram: {MARK<x> | SPACe<x>}**  
**{<Nrf>[ , {LESS | MORE}]}**  
**:IHistogram: {MARK<x> | SPACe<x>}?**  
 MARK<x>の<x>=1~2  
 SPACe<x>の<x>=1~2  
 <Nrf>=ウインドウのラベル  
 例 **:IHISTOGRAM:MARK1 3**  
**:IHISTOGRAM:MARK1?**  
**-> :IHISTOGRAM:MARK1 3**

解説  
 ・「:IHistogram:TRIGger」が「MARK」のとき、MARK1の設定ができます。  
 ・「:IHistogram:TRIGger」が「SPACe」のとき、SPACe1の設定ができます。  
 ・「:IHistogram:TRIGger」が「MTOSpace」のとき、MARK1、SPACe2の設定ができます。  
 ・「:IHistogram:TRIGger」が「STOMark」のとき、SPACe1、MARK2の設定ができます。  
 ・「:IHistogram:TRIGger」が「MTOMark」のとき、MARK1、MARK2の設定ができます。  
 ・「:IHistogram:TRIGger」が「STOSpace」のとき、SPACe1、SPACe2の設定ができます。

### **:IHistogram[ :MODE]**

機能 ウインドウモードを設定/問い合わせします。  
 構文 **:IHistogram[ :MODE] {MULTi | AUTO}**  
**:IHistogram:MODE?**  
 例 **:IHISTOGRAM:MODE MULTI**  
**:IHISTOGRAM:MODE?**  
**-> :IHISTOGRAM:MODE MULTI**

### **:IHistogram:MULTi?**

機能 マルチウインドウに関する全設定を問い合わせます。  
 構文 **:IHistogram:MULTi?**  
 例 **:IHISTOGRAM:MULTi?**  
**-> :IHISTOGRAM:MULTi:SIZE 10;**  
**FREQUENCY 26.315E+06;**  
**TVALUE 38.000E-09;**  
**OFFSET 10.000E-**  
**09;WINDOW1:HORIZONTAL:**  
**CENTER 114.000E-09;**  
**SPAN 0.060E-06;:IHISTOGRAM:MULTi:**  
**WINDOW1:LABEL 1;MARKER:**  
**LEFT 95.000E-09;RIGHT 133.000E-09;:**  
**IHISTOGRAM:MULTi:WINDOW2:**  
**HORIZONTAL:CENTER 152.000E-09;**  
**SPAN 0.060E-06;:IHISTOGRAM:MULTi:**  
**WINDOW2:LABEL 2;MARKER:**  
**LEFT 133.000E-09;**  
**RIGHT 171.000E-09;:IHISTOGRAM:**

**MULTi:WINDOW3:HORIZONTAL:**  
**CENTER 190.000E-09;SPAN 0.060E-06;:**  
**IHISTOGRAM:MULTi:WINDOW3:LABEL 3;**  
**MARKER:LEFT 171.000E-09;**  
**RIGHT 209.000E-09;:IHISTOGRAM:**  
**MULTi:WINDOW4:HORIZONTAL:**  
**CENTER 228.000E-09;SPAN 0.060E-06;:**  
**IHISTOGRAM:MULTi:WINDOW4:LABEL 4;**  
**MARKER:LEFT 209.000E-09;**  
**RIGHT 247.000E-09;:IHISTOGRAM:**  
**MULTi:WINDOW5:HORIZONTAL:**  
**CENTER 266.000E-09;SPAN 0.060E-06;:**  
**IHISTOGRAM:MULTi:WINDOW5:LABEL 5;**  
**MARKER:LEFT 247.000E-09;**  
**RIGHT 285.000E-09;:IHISTOGRAM:**  
**MULTi:WINDOW6:HORIZONTAL:**  
**CENTER 304.000E-09;SPAN 0.060E-06;:**  
**IHISTOGRAM:MULTi:WINDOW6:LABEL 6;**  
**MARKER:LEFT 285.000E-09;**  
**RIGHT 323.000E-09;:IHISTOGRAM:**  
**MULTi:WINDOW7:HORIZONTAL:**  
**CENTER 342.000E-09;SPAN 0.060E-06;:**  
**IHISTOGRAM:MULTi:WINDOW7:LABEL 7;**  
**MARKER:LEFT 323.000E-09;**  
**RIGHT 361.000E-09;:IHISTOGRAM:**  
**MULTi:WINDOW8:HORIZONTAL:**  
**CENTER 380.000E-09;SPAN 0.060E-06;:**  
**IHISTOGRAM:MULTi:WINDOW8:LABEL 8;**  
**MARKER:LEFT 361.000E-09;**  
**RIGHT 399.000E-09;:IHISTOGRAM:**  
**MULTi:WINDOW9:HORIZONTAL:**  
**CENTER 418.000E-09;SPAN 0.060E-06;:**  
**IHISTOGRAM:MULTi:WINDOW9:LABEL 9;**  
**MARKER:LEFT 399.000E-09;**  
**RIGHT 437.000E-09;:IHISTOGRAM:**  
**MULTi:WINDOW10:HORIZONTAL:**  
**CENTER 532.000E-09;SPAN 0.060E-06;:**  
**IHISTOGRAM:MULTi:WINDOW10:LABEL 10;**  
**MARKER:LEFT 513.000E-09;**  
**RIGHT 551.000E-09**

### **:IHistogram:MULTi:FREQUENCY**

機能 周波数形式による定数Tの値を設定/問い合わせします。  
 構文 **:IHistogram:MULTi:FREQUENCY {<周波数>}**  
**:IHistogram:MULTi:FREQUENCY?**  
 <周波数>=4MHz~1000MHz  
 例 **:IHISTOGRAM:MULTi:FREQUENCY 50MHZ**  
**:IHISTOGRAM:MULTi:FREQUENCY?**  
**-> :IHISTOGRAM:MULTi:**  
**FREQUENCY 50.000E+06**  
 解説 「:IHistogram:MULTi:TVALUE」の周波数形式での設定/問い合わせです。

**: IHistogram: MULTi: OFFSet**

機能 定数Tへのオフセット値を設定/問い合わせします。

構文 : IHistogram: MULTi: OFFSet {<時間>}  
: IHistogram: MULTi: OFFSet?  
<時間>= -100ns~300ns(25psステップ)

例 : IHISTOGRAM: MULTi: OFFSET 10ns  
: IHISTOGRAM: MULTi: OFFSET?  
-> : IHISTOGRAM: MULTi:  
OFFSET 10.000E-09

**: IHistogram: MULTi: SIZE**

機能 ウィンドウ数を設定/問い合わせします。

構文 : IHistogram: MULTi: SIZE {<NRf>}  
: IHistogram: MULTi: SIZE?  
<NRf>= ウィンドウ数(1~14)

例 : IHISTOGRAM: MULTi: SIZE 10  
: IHISTOGRAM: MULTi: SIZE?  
-> : IHISTOGRAM: MULTi: SIZE 10

**: IHistogram: MULTi: TVALue**

機能 定数Tの値を設定/問い合わせします。

構文 : IHistogram: MULTi: TVALue {<時間>}  
: IHistogram: MULTi: TVALue?  
<時間>= 1ns~250ns(25psステップ)

例 : IHISTOGRAM: MULTi: TVALUE 38ns  
: IHISTOGRAM: MULTi: TVALUE?  
-> : IHISTOGRAM: MULTi:  
TVALUE 38.000E-09

**: IHistogram: MULTi: UDate**

機能 定数Tの値、オフセット値をもとにウィンドウの設定を変更します。

構文 : IHistogram: MULTi: UDate

例 : IHISTOGRAM: MULTi: UPDATE

**: IHistogram: MULTi: WINDow<x>?**

機能 各ウィンドウに関する全設定を問い合わせます。

構文 : IHistogram: MULTi: WINDow<x>?  
<x>= ウィンドウ番号(1~ウィンドウ数)

例 : IHISTOGRAM: MULTi: WINDOW1?  
-> : IHISTOGRAM: MULTi: WINDOW1:  
HORIZONTAL: CENTER 114.000E-09;  
SPAN 0.060E-06; : IHISTOGRAM: MULTi:  
WINDOW1: LABEL 1; MARKER:  
LEFT 95.000E-09; RIGHT 133.000E-09

**: IHistogram: MULTi: WINDow<x>: HORizontal?**

機能 各ウィンドウの水平軸(X軸)に関する全設定を問い合わせます。

構文 : IHistogram: MULTi: WINDow<x>:  
HORizontal?  
<x>= ウィンドウ番号(1~ウィンドウ数)

例 : IHISTOGRAM: MULTi: WINDOW1:  
HORIZONTAL?  
-> : IHISTOGRAM: MULTi: WINDOW1:  
HORIZONTAL: CENTER 114.000E-09;  
SPAN 0.060E-06;

**: IHistogram: MULTi: WINDow<x>: HORizontal: CENTER**

機能 各ウィンドウの水平軸(X軸)の中央位置を設定/問い合わせします。

構文 : IHistogram: MULTi: WINDow<x>:  
HORizontal: CENTER {<時間>}  
: IHistogram: MULTi: WINDow<x>:  
HORizontal: CENTER?

<x>= ウィンドウ番号(1~ウィンドウ数)  
<時間>= -50ns~3.2μs(25psステップ)

例 : IHISTOGRAM: MULTi: WINDOW1:  
HORIZONTAL: CENTER 114ns  
: IHISTOGRAM: MULTi: WINDOW1:  
HORIZONTAL: CENTER?  
-> : IHISTOGRAM: MULTi: WINDOW1:  
HORIZONTAL: CENTER 114.000E-09

**: IHistogram: MULTi: WINDow<x>: HORizontal: SPAN**

機能 各ウィンドウの水平軸(X軸)の幅を設定/問い合わせします。

構文 : IHistogram: MULTi: WINDow<x>:  
HORizontal: SPAN {<時間>}  
: IHistogram: MULTi: WINDow<x>:  
HORizontal: SPAN?

<x>= ウィンドウ番号(1~ウィンドウ数)  
<時間>= 1.5ns, 3ns, 7.5ns, 15ns, 30ns,  
60ns, 150ns, 300ns, 600ns, 1.5μs, 6μs

例 : IHISTOGRAM: MULTi: WINDOW1:  
HORIZONTAL: SPAN 60ns  
: IHISTOGRAM: MULTi: WINDOW1:  
HORIZONTAL: SPAN?  
-> : IHISTOGRAM: MULTi: WINDOW1:  
HORIZONTAL: SPAN 0.060E-06

## 4.9 IHistogramグループ

### **: IHistogram: MULTi: WINDOW<x>: LABEL**

機能 ウィンドウラベルを設定/問い合わせします。

構文 : IHistogram: MULTi: WINDOW<x>: LABEL  
{<NRf>}  
: IHistogram: MULTi: WINDOW<x>: LABEL?  
<x>=ウィンドウ番号(1~ウィンドウ数)  
<NRf>=1~16

例 : IHISTOGRAM: MULTi: WINDOW1: LABEL 3  
: IHISTOGRAM: MULTi: WINDOW1: LABEL?  
-> : IHISTOGRAM: MULTi: WINDOW1:  
LABEL 3

### **: IHistogram: MULTi: WINDOW<x>: MARKer?**

機能 各ウィンドウの水平軸(X軸)マーカーに関する全設定を問い合わせます。

構文 : IHistogram: MULTi: WINDOW<x>: MARKer?  
<x>=ウィンドウ番号(1~ウィンドウ数)

例 : IHISTOGRAM: MULTi: WINDOW1: MARKer?  
-> : IHISTOGRAM: MULTi: WINDOW1:  
MARKer: LEFT 95.000E-09;  
RIGHT 133.000E-09

### **: IHistogram: MULTi: WINDOW<x>: MARKer:**

#### **{LEFT | RIGHT}**

機能 各ウィンドウの各マーカーの位置の設定/問い合わせします。

構文 : IHistogram: MULTi: WINDOW<x>: MARKer:  
{LEFT | RIGHT} {<時間>}  
: IHistogram: MULTi: WINDOW<x>: MARKer:  
{LEFT | RIGHT}?  
<x>=ウィンドウ番号(1~ウィンドウ数)  
<時間>=マーカー位置(25psステップ)

例 : IHISTOGRAM: MULTi: WINDOW1: MARKer:  
LEFT 100ns  
: IHISTOGRAM: MULTi: WINDOW1: MARKer:  
LEFT?  
-> : IHISTOGRAM: MULTi: WINDOW1:  
MARKer: LEFT 100.000E-09

### **: IHistogram: POLarity**

機能 ポラリティを設定/問い合わせします。

構文 : IHistogram: POLarity {MARK | SPACE}  
: IHistogram: POLarity?

例 : IHISTOGRAM: POLARITY SPACE  
: IHISTOGRAM: POLARITY?  
-> : IHISTOGRAM: POLARITY SPACE

### **: IHistogram: SYNC**

機能 Sync機能のON/OFFを設定/問い合わせします。

構文 : IHistogram: SYNC {<Boolean>}  
: IHistogram: SYNC?

例 : IHISTOGRAM: SYNC ON  
: IHISTOGRAM: SYNC?  
-> : IHISTOGRAM: SYNC 1

### **: IHistogram: TARGet**

機能 解析対象を設定/問い合わせします。

構文 : IHistogram: TARGet  
{PREVIOUS | MIDDLE | NEXT | BOTH}  
: IHistogram: TARGet?

例 : IHISTOGRAM: TARGET NEXT

: IHISTOGRAM: TARGET?

-> : IHISTOGRAM: TARGET NEXT

解説

- ・パルス幅/パルス幅A→パルス幅B測定で、「: IHistogram: TMODE」が「SINGLE」のとき、「PREVIOUS | NEXT | BOTH」が設定できます。
- ・パルス幅/パルス幅A→パルス幅B測定で、「: IHistogram: TMODE」が「COMBINATION」のとき、「PREVIOUS | NEXT」が設定できます。
- ・パルス幅A→AtoBタイムインターバル測定で、片エッジ測定、「: IHistogram: TMODE」が「COMBINATION」のとき、「PREVIOUS | NEXT」が設定できます。
- ・パルス幅A→AtoBタイムインターバル測定で、両エッジ測定、「: IHistogram: TMODE」が「SINGLE」のとき、「PREVIOUS | NEXT」が設定できません。
- ・パルス幅A→AtoBタイムインターバル測定で、両エッジ測定、「: IHistogram: TMODE」が「COMBINATION」のとき「PREVIOUS | MIDDLE | NEXT」が設定できます。
- ・「: IHistogram: TMODE」が「BETWEEN」のとき、設定できません。

**: IHistogram: TMODe**

機能 トリガモードを設定/問い合わせします。

構文 : IHistogram: TMODe  
{SINGle|COMBination|BETWeen}  
: IHistogram: TMODe?

例 : IHISTOGRAM: TMODE COMBINATION  
: IHISTOGRAM: TMODE?

-> : IHISTOGRAM: TMODE COMBINATION  
パルス幅A→AtoBタイムインターバル測定のと  
きは, 「BETWeen」は設定できません。

**: IHistogram: TRIGger**

機能 トリガ条件を設定/問い合わせします。

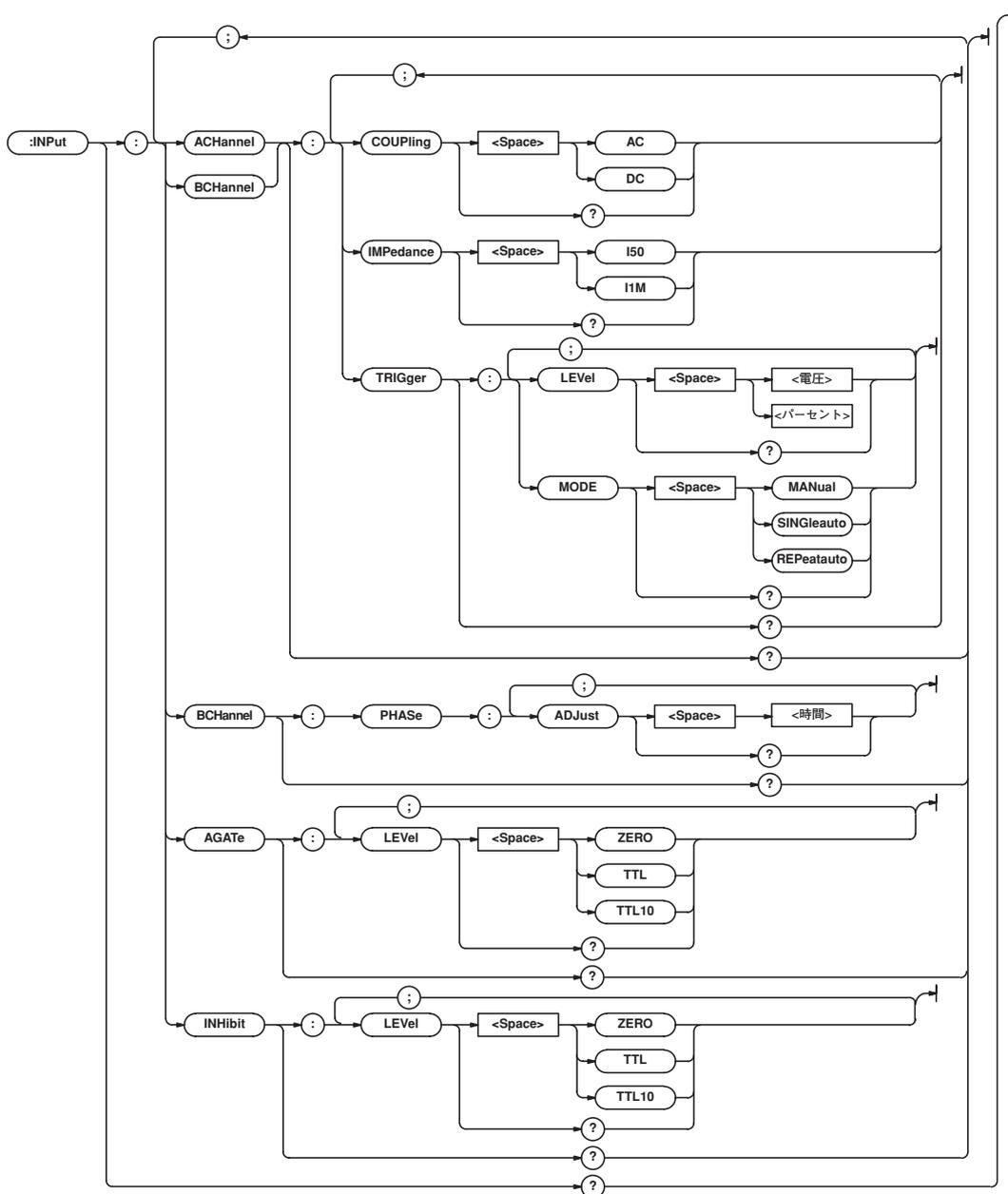
構文 : IHistogram: TRIGger  
{MARK|SPACE|MTOSpace|STOMark|  
MTOMark|STOSpace}  
: IHistogram: TRIGger?

例 : IHISTOGRAM: TRIGGER MARK  
: IHISTOGRAM: TRIGGER?

-> : IHISTOGRAM: TRIGGER MARK  
・ 「: IHistogram: TMODe」が「SINGle」の  
とき, 「MARK|SPACE」が設定できます。  
・ 「: IHistogram: TMODe」が  
「COMBination」のとき,  
「MTOSpace|STOMark」が設定できます。  
・ 「: IHistogram: TMODe」が「BETWeen」の  
とき, 「MTOMark|STOSpace」が設定できま  
す。

### 4.10 INPutグループ

入力部に関するグループです。フロントパネルのINPUTキーと同じ設定、および設定内容の問い合わせができます。



**:INPut?**

機能 入力条件に関する全設定を問い合わせます。

構文

:INPut?

例

:INPUT?

```
-> :INPUT:ACHANNEL:COUPLING DC;
IMPEDANCE I1M;TRIGGER:
LEVEL 2.500E+00;MODE MANUAL;:INPUT:
AGATE:LEVEL TTL;:INPUT:BCHANNEL:
COUPLING DC;IMPEDANCE I1M;TRIGGER:
LEVEL 0.000E+00;MODEMANUAL;:INPUT:
INHIBIT:LEVEL TTL
```

**:INPut:{ACHannel|BChannel}?**

機能 各チャンネルに関する全設定を問い合わせます。

構文

:INPut:{ACHannel|BChannel}?

例

(以下は、CH Aにおける例です。)

```
:INPUT:ACHANNEL?
-> :INPUT:ACHANNEL:COUPLING DC;
IMPEDANCE I1M;TRIGGER:
LEVEL 2.500E+00;TRIGGER:MODE MANUAL
```

**:INPut:{ACHannel|BCHannel}:COUPLing**

機能 各チャンネルのカップリングを設定/問い合わせします。

構文 :INPut:{ACHannel|BCHannel}:COUPLing {AC|DC}  
:INPut:{ACHannel|BCHannel}:COUPLing?

例 (以下は, CH Aにおける例です。)

```
:INPUT:ACHANNEL:COUPLING DC
:INPUT:ACHANNEL:COUPLING?
-> :INPUT:ACHANNEL:COUPLING DC
```

**:INPut:{ACHannel|BCHannel}:IMPedance**

機能 各チャンネルの入カインピーダンスを設定/問い合わせします。

構文 :INPut:{ACHannel|BCHannel}:IMPedance {I50|I1M}  
:INPut:{ACHannel|BCHannel}:IMPedance?

例 (以下は, CH Aにおける例です。)

```
:INPUT:ACHANNEL:IMPEDANCE I1M
:INPUT:ACHANNEL:IMPEDANCE?
-> :INPUT:ACHANNEL:IMPEDANCE I1M
```

**:INPut:{ACHannel|BCHannel}:TRIGger?**

機能 各チャンネルのトリガに関する全設定を問い合わせします。

構文 :INPut:{ACHannel|BCHannel}:TRIGger?  
例 (以下は, CH Aにおける例です。)  
:INPUT:ACHANNEL:TRIGGER?  
-> :INPUT:ACHANNEL:TRIGGER:LEVEL 2.500E+00;MODE MANUAL

**:INPut:{ACHannel|BCHannel}:TRIGger:LEVel**

機能 各チャンネルのトリガレベルを設定/問い合わせします。

構文 :INPut:{ACHannel|BCHannel}:TRIGger:LEVel {<電圧>|<パーセント>}  
:INPut:{ACHannel|BCHannel}:TRIGger:LEVel?  
トリガマニュアルモード <電圧>=-5V~+5V(1mVステップ)  
トリガシングル/リピートオート時 <パーセント>=>0PCT~100PCT

例 (以下は, CH A, トリガマニュアルモードにおける例です。)

```
:INPUT:ACHANNEL:TRIGGER:LEVEL 2.5V
:INPUT:ACHANNEL:TRIGGER:LEVEL?
-> :INPUT:ACHANNEL:TRIGGER:LEVEL 2.500E+00
```

**:INPut:{ACHannel|BCHannel}:TRIGger:MODE**

機能 各チャンネルのトリガモードを設定/問い合わせします。

構文 :INPut:{ACHannel|BCHannel}:TRIGger:MODE {MANual|SINGleauto|REPeatauto}  
:INPut:{ACHannel|BCHannel}:TRIGger:MODE?

例 (以下は, CH Aにおける例です。)

```
:INPUT:ACHANNEL:TRIGGER:MODE MANUAL
:INPUT:ACHANNEL:TRIGGER:MODE?
-> :INPUT:ACHANNEL:TRIGGER:MODE MANUAL
```

**:INPut:AGATe?**

機能 アーミング/外部ゲートの全設定を問い合わせます。

構文 :INPut:AGATe?

例 :INPUT:AGATE?

-> :INPUT:AGATE:LEVEL ZERO

解説 外部アーミング入力端子と外部ゲート入力端子は兼用なので, 設定は共通です。

**:INPut:AGATe:LEVel**

機能 アーミング/ゲートレベルを設定/問い合わせします。

構文 :INPut:AGATe:LEVel {ZERO|TTL|TTL10}  
:INPut:AGATe:LEVel?

例 :INPUT:AGATE:LEVEL ZERO

:INPUT:AGATE:LEVEL?

-> :INPUT:AGATE:LEVEL ZERO

**:INPut:BCHannel:PHASe?**

機能 CH Bの位相調整に関する全設定を問い合わせます。

構文 :INPut:BCHannel:PHASe?

例 :INPUT:BCHANNEL:PHASE?

-> :INPUT:BCHANNEL:PHASE:ADJUST 6.0E-09

解説 AtoBタイムインターバル/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル測定のと看, 問い合わせができます。

**:INPut:BCHannel:PHASe:ADJust**

機能 CH Bの位相調整時間を設定/問い合わせします。

構文 :INPut:BCHannel:PHASe:ADJust {<時間>}  
:INPut:BCHannel:PHASe:ADJust?

<時間>=0s~10ns(100psステップ)

例 :INPUT:BCHANNEL:PHASE:ADJUST 100ps

ADJUST 100ps

:INPUT:BCHANNEL:PHASE:ADJUST?

-> :INPUT:BCHANNEL:PHASE:ADJUST 100.0E-12

## 4.10 INPutグループ

---

### **:INPut:INHibit?**

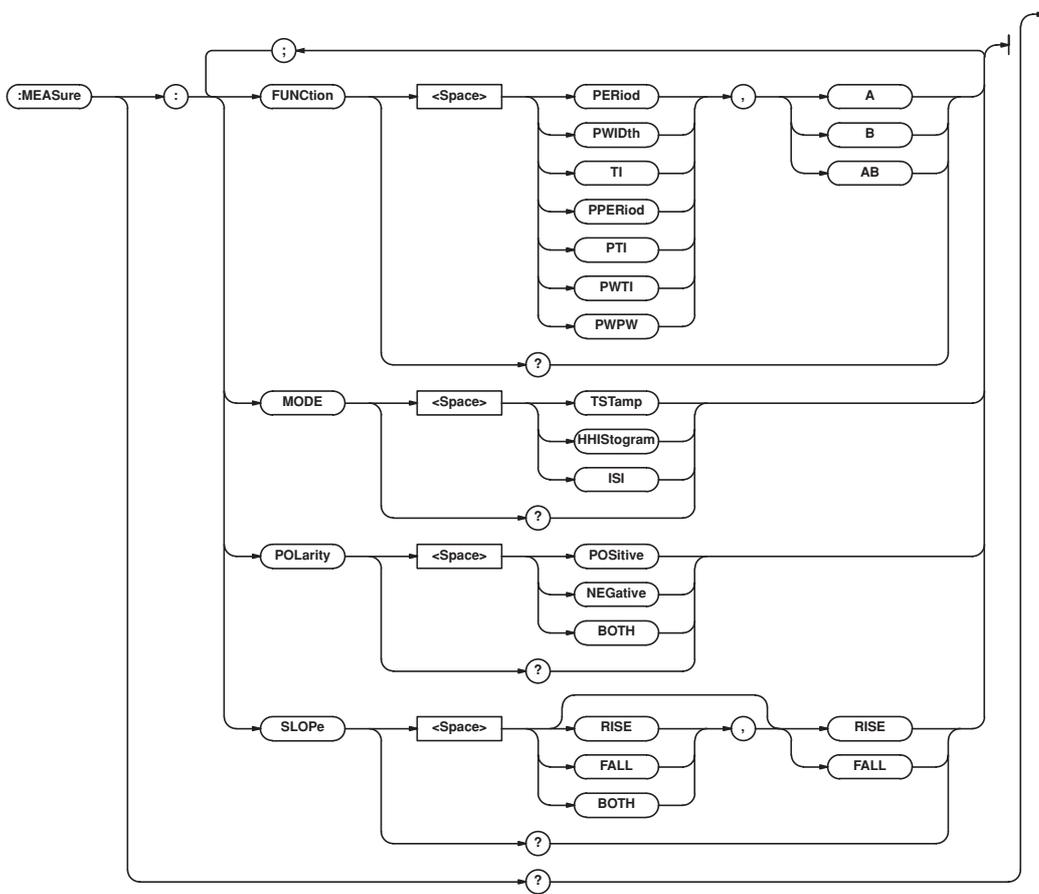
機能      インヒビットに関する全設定を問い合わせます。  
構文      :INPut:INHibit?  
例        :INPUT:INHIBIT?  
          -> :INPUT:INHIBIT:LEVEL ZERO

### **:INPut:INHibit:LEVel**

機能      インヒビットレベルを設定/問い合わせします。  
構文      :INPut:INHibit:LEVel  
          {ZERO|TTL|TTL10}  
          :INPut:INHibit:LEVel?  
例        :INPUT:INHIBIT:LEVEL TTL  
          :INPUT:INHIBIT:LEVEL?  
          -> :INPUT:INHIBIT:LEVEL TTL

## 4.11 MEASureグループ

測定条件に関するグループです。フロントパネルのMODEキーと同じ設定、および設定内容の問い合わせができます。



### :MEASure?

機能 測定条件の全設定を問い合わせます。

構文 :MEASure?

例 :MEASURE?

-> :MEASURE:MODE HHISTOGRAM;  
FUNCTION PERIOD,A;SLOPE RISE

### :MEASure:FUNCTION

機能 測定ファンクションを設定/問い合わせします。

構文 :MEASure:FUNCTION

{PERiod|PWIDTH|TI|PPERiod|PTI|PWTI|PWPW},{A|B|AB}

:MEASure:FUNCTION?

例 :MEASURE:FUNCTION PERIOD,A

:MEASURE:FUNCTION?

-> :MEASURE:FUNCTION PERIOD,A

解説

- ・ 周期(PERIOD)/パルス幅(PWIDTH)測定のと  
き、第2パラメータは、「A|B」の設定が  
できます。
- ・ AtoBタイムインターバル(TI)/周期A&周期B  
(PPERiod)/周期A&AtoBタイムインター  
バル(PTI)/パルス幅A&AtoBタイムインター  
バル(PWTI)/パルス幅A&パルス幅B(PWPW)測定  
のとき、第2パラメータは、ABのみが指定可  
能です。
- ・ サンプリングモードが符号間干渉解析モードの  
ときは、「PWIDTH|PWTI|PWPW」の設定がで  
きます。

#### 4.11 MEASureグループ

##### **:MEASure:MODE**

機能 サンプルングモードを設定/問い合わせします。

構文 **:MEASure:MODE**  
{TSTamp|HHIStogram|ISI}

例 **:MEASure:MODE HHISTOGRAM**  
**:MEASure:MODE?**  
-> **:MEASure:MODE HHISTOGRAM**

##### **:MEASure:POLarity**

機能 パルス幅/パルス幅A&AtoBタイムインターバル/  
パルス幅A&パルス幅B測定時のポラリティを設定/  
問い合わせします。

構文 **:MEASure:POLarity**  
{POSitive|NEGative|BOTH}

例 **:MEASure:POLarity POSITIVE**  
**:MEASure:POLarity?**  
-> **:MEASure:POLarity POSITIVE**

解説 パルス幅A&AtoBタイムインターバル/パルス幅  
A&パルス幅B測定の場合は、問い合わせができません。

##### **:MEASure:SLOPe**

機能 周期/AtoBタイムインターバル/周期A&周期B/  
周期A&AtoBタイムインターバル/パルス幅  
A&AtoBタイムインターバル測定時のスロープを設定/  
問い合わせします。

構文 **:MEASure:SLOPe**  
[ {RISE|FALL|BOTH}, ] {RISE|FALL}

例 **周期測定の場合**  
**:MEASure:SLOPe RISE**  
**:MEASure:SLOPe?**  
-> **:MEASure:SLOPe RISE**

**AtoBタイムインターバル測定の場合**  
**:MEASure:SLOPe BOTH,RISE**  
**:MEASure:SLOPe?**

-> **:MEASure:SLOPe BOTH,RISE**  
**周期A&周期B測定の場合**  
**:MEASure:SLOPe FALL,FALL**  
**:MEASure:SLOPe?**

-> **:MEASure:SLOPe FALL,FALL**  
**周期A&AtoBタイムインターバル測定の場合**  
**:MEASure:SLOPe FALL,RISE**  
**:MEASure:SLOPe?**

-> **:MEASure:SLOPe FALL,RISE**  
**パルス幅A&AtoBタイムインターバル**  
**:MEASure:SLOPe?**

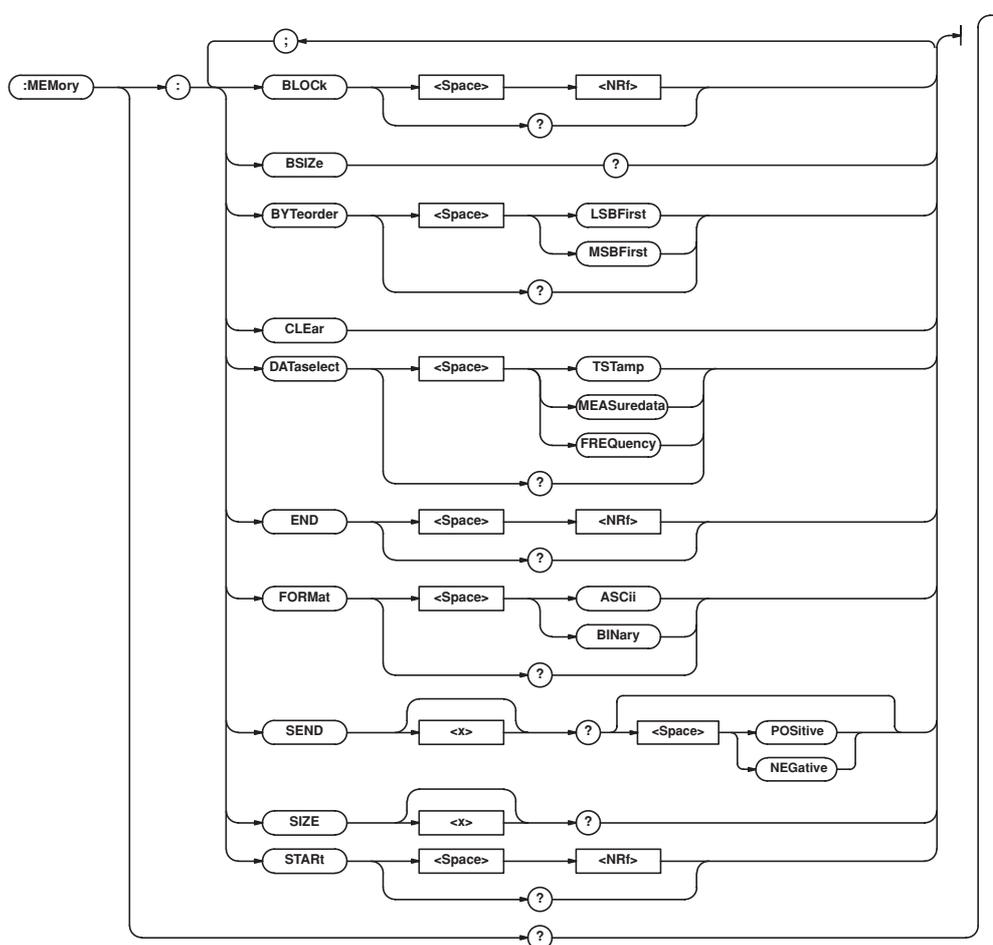
-> **:MEASure:SLOPe BOTH,RISE**

解説

- ・周期測定の場合は、第2パラメーターだけを設定してください。
- ・周期A&周期B測定の場合は、「RISE,RISE」または「FALL,FALL」の設定ができます。
- ・周期A&AtoBタイムインターバル測定の場合は、「RISE,RISE」または「FALL,RISE」の設定ができます。
- ・サンプルングモードが、タイムスタンプ/ハードウェアヒストグラム、パルス幅A&AtoBタイムインターバル測定の場合は「BOTH,RISE」または「Both,FALL」の設定ができます。

## 4.12 MEMoryグループ

測定データの外部への送信に関するグループです。このグループに相当するフロントパネルのキーはありません。



### :MEMory?

機能 測定データの外部への送信に関する全設定を問い合わせます。

構文 :MEMory?

例 :MEMory?

```
-> :MEMory:BLOCK 0;
BYTeORDER LSBFIRST;
DATASELECT TSTAMP;END 51000;
FORMAt ASCII;START 1
```

### :MEMory:BLOCK

機能 ブロックサンプリング時の対象ブロックを設定/問い合わせします。

構文 :MEMory:BLOCK {<NRf>}

例 :MEMory:BLOCK 1  
<NRf>=ブロック番号(0~1000)

例 :MEMory:BLOCK 1

例 :MEMory:BLOCK?

例 -> :MEMory:BLOCK 1

解説 <NRf>=0は、ブロック全体を表します。

### :MEMory:BSIZE?

機能 ブロックサンプリングで、実際に測定したブロック数を問い合わせます。

構文 :MEMory:BSIZE?

例 :MEMory:BSIZE?

-> :MEMory:BSIZE 100

### :MEMory:BYTeorder

機能 バイナリデータの送信順序を設定/問い合わせします。

構文 :MEMory:BYTeorder  
{LSBFirst|MSBFirst}

例 :MEMory:BYTeorder?

例 :MEMory:BYTeorder LSBFIRST

例 :MEMory:BYTeorder?

-> :MEMory:BYTeorder LSBFIRST

### :MEMory:CLear

機能 測定データをクリアします。

構文 :MEMory:CLear

例 :MEMory:CLEAR

## 4.12 MEMoryグループ

### :MEMory:DATaselect

機能	送信対象となるデータを設定/問い合わせします。
構文	:MEMory:DATaselect {TSTamp MEASuredata FREQuency} :MEMory:DATaselect?
例	:MEMORY:DATASELECT TSTAMP :MEMORY:DATASELECT?
解説	-> :MEMORY:DATASELECT TSTAMP <ul style="list-style-type: none"> <li>測定モードがハードウェアヒストグラムモードのとき、タイムスタンプ(TSTamp)を指定し、「:MEMory:SEND&lt;x&gt;?」コマンドを実行するとエラーとなります。</li> <li>測定モードがタイムスタンプ/符号間干渉解析モードのとき、度数(FREQuency)を指定し、「:MEMory:SEND&lt;x&gt;?」コマンドを実行するとエラーとなります。</li> </ul>

### :MEMory:END

機能	送信終了のデータ位置を設定/問い合わせします。
構文	:MEMory:END {<NRf> <時間>} :MEMory:END? <ul style="list-style-type: none"> <li>●ハードウェアヒストグラムモードの場合 &lt;時間&gt;=-2ns~3.2μs</li> <li>●タイムスタンプ/符号間干渉解析モードの場合  <ul style="list-style-type: none"> <li>周期/AtoBタイムインターバル/パルス幅 &lt;NRf&gt;=点数(1~1,024,000)</li> <li>周期A&amp;周期B/周期A&amp;AtoBタイムインターバル/パルス幅A&amp;AtoBタイムインターバル/パルス幅A&amp;パルス幅B &lt;NRf&gt;=点数(1~512,000)</li> </ul> </li> </ul>
例	:MEMORY:END 1ns :MEORY:END? -> :MEMORY:END 1.000E-09

### :MEMory:FORMat

機能	送信するデータの形式を設定/問い合わせします。
構文	:MEMory:FORMat {ASCIi BINary} :MEMory:FORMat?
例	:MEMORY:FORMAT ASCII :MEMORY:FORMAT? -> :MEMORY:FORMAT ASCII

### :MEMory:SEND<x>?

機能	「MEMory:DATaselect」で設定した測定データの送信を実行します。
構文	:MEMory:SEND<x>? [{POSitive NEGative}] <x>=1~2
例	:MEMORY:SEND1? -> #800000016abcdabceabcfabcb
解説	<ul style="list-style-type: none"> <li>送信フォーマットがASCIIの場合、&lt;NR3&gt;形式の応答データがデータ点数分だけ「,」で区切って出力されます。</li> <li>BINARYの場合、単一のブロックデータが出力されます。</li> <li>バイナリデータの実数への変換式は以下のとおりです。 <ul style="list-style-type: none"> <li>●ハードウェアヒストグラムモードの場合 <ul style="list-style-type: none"> <li>両パルス幅/両エッジ測定で、「:CALCulation:POLarity」が「POSneg」のとき、「POSitive NEGative」により出力するデータを指定してください。</li> <li>度数(FREQuency) 符号なしの4バイト整数で、度数が格納されています。</li> <li>測定値(MEASuredata) : 符号ありの4バイト整数とみなし、これに25psを掛けた値が測定値になります。</li> <li>タイムスタンプ値(TSTamp) : ハードウェアヒストグラムモードではタイムスタンプを直接出力することはできません。</li> </ul> </li> <li>●タイムスタンプ/符号間干渉解析モードの場合 <ul style="list-style-type: none"> <li>度数(FREQuency) タイムスタンプ/符号間干渉解析モードでは度数を直接出力することはできません。</li> <li>測定値(MEASuredata) : 符号なしの4バイト整数とみなし、これに25psを掛けた値が測定値になります。</li> <li>タイムスタンプ値(TSTamp) : 符号なしの4バイト整数とみなし、これに100nsの時間を掛けた値がタイムスタンプ値になります。</li> <li>Binary時のデータフォーマットは、3-6ページの&lt;ブロックデータ&gt;を参照してください。</li> </ul> </li> </ul> </li> </ul>

### :MEMory:SIZE<x>?

機能	測定したデータ数を問い合わせます。
構文	:MEMory:SIZE<x>? <x>=1~2
例	:MEMORY:SIZE1? -> 10000
解説	測定モードがハードウェアヒストグラムモードのときは、「NAN」を返します。

**:MEMory:STARt**

機能 送信開始のデータ位置を設定/問い合わせします。

構文 :MEMory:STARt {<NRf>}  
:MEMory:STARt?

●ハードウェアヒストグラムモードの場合

<時間>=-2ns~3.2μs

●タイムスタンプ/符号間干渉解析モードの場合

・周期/AtoBタイムインターバル/パルス幅

<NRf>=点数(1~1,024,000)

・周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B

<NRf>=点数(1~512,000)

例 :MEMORY:START 1ns

:MEMORY:START?

-> :MEMORY:START 1.000E-09

**4.13 RECallグループ****:RECall**

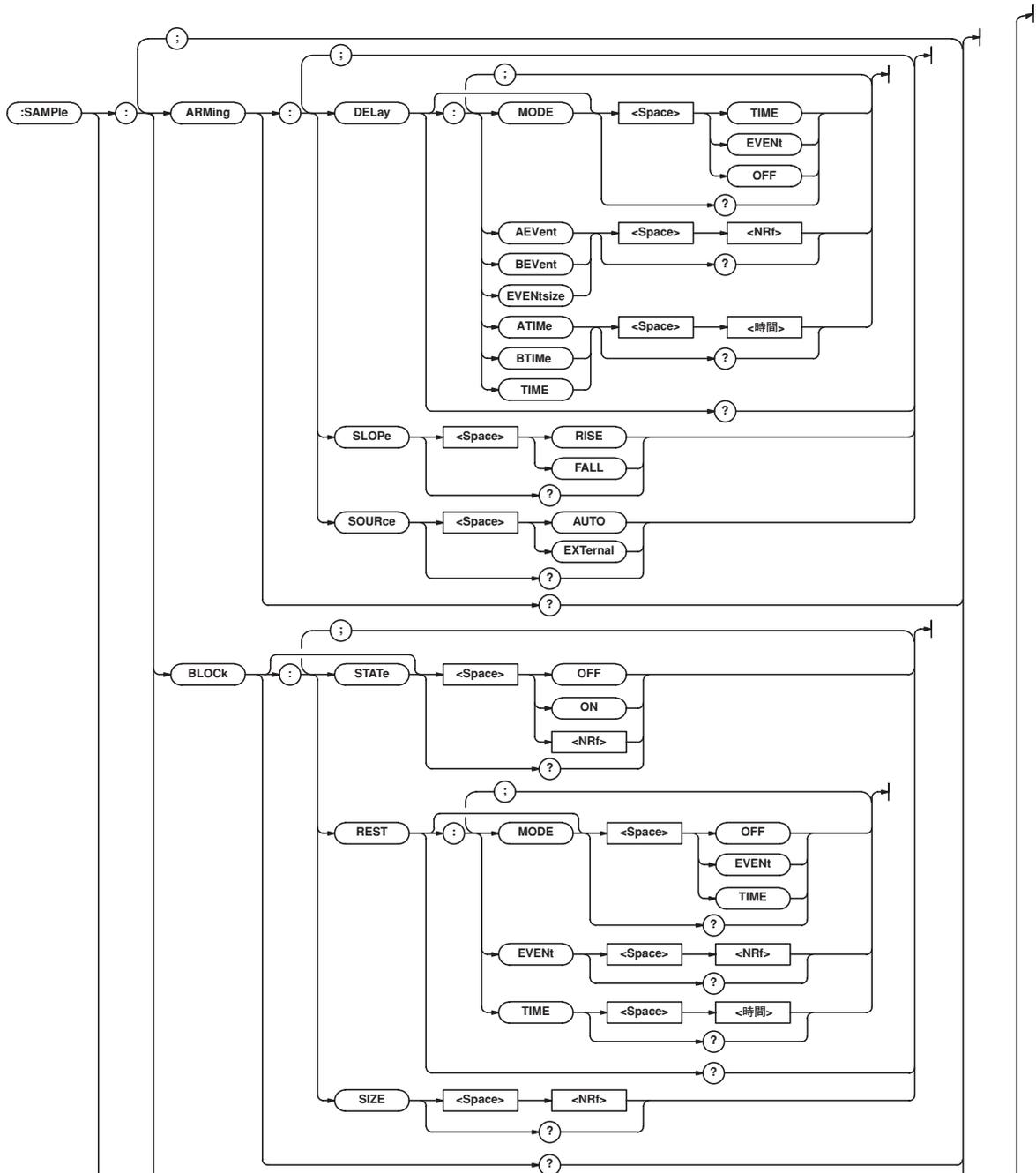
機能 設定情報をリコールします。

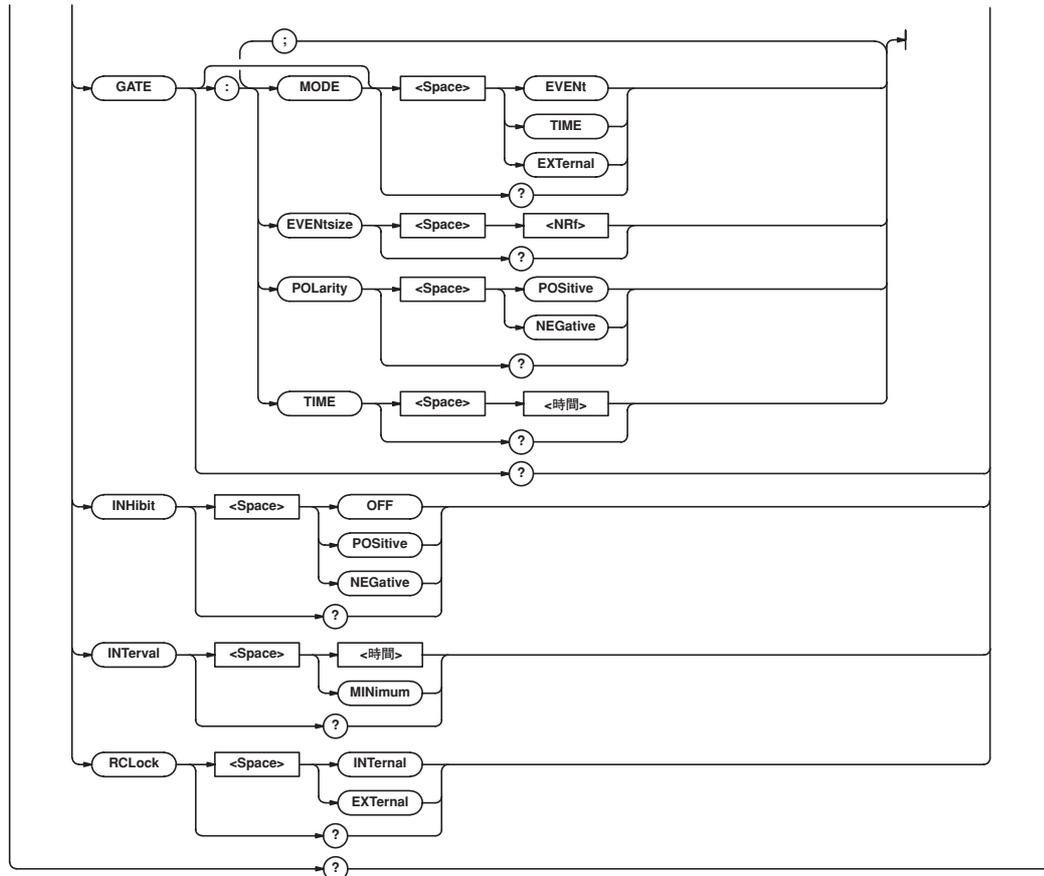
構文 :RECall <NRf>  
<NRf>=0~31

例 :RECALL 0

### 4.14 SAMPlEグループ

サンプリング条件の設定に関するグループです。フロントパネルのSAMPLEキーと同じ設定、および設定内容の問い合わせができます。



**:SAMPlE?**

機能 サンプリングに関する全設定を問い合わせます。

構文 :SAMPlE?

例 :SAMPLE?

```
-> :SAMPLE:ARMING:DELAY:
ATIME 1.0E-06;BTIME 1.0E-06;
MODE TIME;;SAMPLE:ARMING:
SLOPE FALL;SOURCE EXTERNAL;;SAMPLE:
GATE:EVENTSIZE 100;MODE EVENT;;
SAMPLE:INHIBIT OFF;RCLOCK INTERNAL
```

**:SAMPlE:ARMIing?**

機能 アーミングに関する全設定を問い合わせます。

構文 :SAMPlE:ARMIing?

例 :SAMPLE:ARMING?

```
-> :SAMPLE:ARMING:DELAY:
ATIME 1.0E-06;BTIME 1.0E-06;
MODE TIME;;SAMPLE:ARMING:
SLOPE FALL;SOURCE EXTERNAL
```

解説 「:SAMPlE:GATE:MODE」が「EXTernal」のとき、問い合わせができません。

**:SAMPlE:ARMIing:DELAy?**

機能 アーミングディレイに関する全設定を問い合わせます。

構文 :SAMPlE:ARMIing:DELAy?

例 :SAMPLE:ARMING:DELAY?

```
-> :SAMPLE:ARMING:DELAY:
ATIME 1.0E-06;BTIME 1.0E-06;
MODE TIME
```

#### 4.14 SAMPLeグループ

**:SAMPLe:ARMinG:DELAy:{AEVentsize|BEVentsize|EVENTsize}**

機能 各チャンネルのアーミングディレイイベントを設定/問い合わせします。

構文 **:SAMPLe:ARMinG:DELAy:{AEVentsize|BEVentsize|EVENTsize} {<NRf>}**  
**:SAMPLe:ARMinG:DELAy:{AEVentsize|BEVentsize|EVENTsize}?**  
<NRf>=イベント数(1~1000000)

例 **:SAMPLe:ARMinG:DELAy:**  
**AEVENTSIZE 1000**  
**:SAMPLe:ARMinG:DELAy:AEVENTSIZE?**  
-> **:SAMPLe:ARMinG:DELAy:**  
**AEVENTSIZE 1000**

解説  
・「**:SAMPLe:ARMinG:DELAy:MODE**」が「**EVENT**」のとき、設定ができます。  
・周期A&周期B/パルス幅A&パルス幅B測定するとき、各チャンネル{**AEVENTsize|BEVENTsize**}ごとに設定ができます。  
・上記以外の場合、{**EVENTsize**}で設定してください。

**:SAMPLe:ARMinG:DELAy:{ATIME|BTIME|TIME}**

機能 各チャンネルのアーミングディレイ時間を設定/問い合わせします。

構文 **:SAMPLe:ARMinG:DELAy:{ATIME|BTIME|TIME} {<時間>}**  
**:SAMPLe:ARMinG:DELAy:{ATIME|BTIME|TIME}?**  
<時間>=1 $\mu$ s~1s(100nsステップ)

例 **:SAMPLe:ARMinG:DELAy:BTIME 1us**  
**:SAMPLe:ARMinG:DELAy:BTIME?**  
-> **:SAMPLe:ARMinG:DELAy:**  
**BTIME 1.0E-06**

解説  
・「**:SAMPLe:ARMinG:DELAy:MODE**」が「**TIME**」のとき、設定ができます。  
・周期A&周期B/パルス幅A&パルス幅B測定するとき、各チャンネル{**ATIME|BTIME**}ごとに設定ができます。  
・上記以外の場合、{**TIME**}で設定してください。

**:SAMPLe:ARMinG:DELAy[:MODE]**

機能 アーミングディレイモードを設定/問い合わせします。

構文 **:SAMPLe:ARMinG:DELAy[:MODE]**  
**{TIME|EVENT|OFF}**  
**:SAMPLe:ARMinG:DELAy:MODE?**

例 **:SAMPLe:ARMinG:DELAy:MODE OFF**  
**:SAMPLe:ARMinG:DELAy:MODE?**  
-> **:SAMPLe:ARMinG:DELAy:MODE OFF**

**:SAMPLe:ARMinG:SLOPe**

機能 アーミングスロープを設定/問い合わせします。

構文 **:SAMPLe:ARMinG:SLOPe {RISE|FALL}**  
**:SAMPLe:ARMinG:SLOPe?**

例 **:SAMPLe:ARMinG:SLOPe RISE**  
**:SAMPLe:ARMinG:SLOPe?**  
-> **:SAMPLe:ARMinG:SLOPe RISE**

**:SAMPLe:ARMinG:SOURce**

機能 アーミングソースを設定/問い合わせします。

構文 **:SAMPLe:ARMinG:SOURce**  
**{AUTO|EXTErnal}**  
**:SAMPLe:ARMinG:SOURce?**

例 **:SAMPLe:ARMinG:SOURce AUTO**  
**:SAMPLe:ARMinG:SOURce?**  
-> **:SAMPLe:ARMinG:SOURce AUTO**

**:SAMPLe:BLOCK?**

機能 ブロックサンプリングに関する全設定を問い合わせます。

構文 **:SAMPLe:BLOCK?**

例 **:SAMPLe:BLOCK?**  
-> **:SAMPLe:BLOCK:REST:EVENT 1000;**  
**MODE EVENT;;SAMPLe:BLOCK:SIZE 10;**  
**STATE 1**

解説  
・周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定するとき、問い合わせできません。  
・「**:SAMPLe:GATE:MODE**」が「**EXTErnal**」のとき、問い合わせできません。  
・符号間干渉解析モードのとき、問い合わせできません。

**:SAMPLe:BLOCK:REST?**

機能 ブロックサンプリングの休止に関する全設定を問い合わせます。

構文 **:SAMPLe:BLOCK:REST?**

例 **:SAMPLe:BLOCK:REST?**  
-> **:SAMPLe:BLOCK:EVENT 1000;REST:**  
**MODE EVENT**

**:SAMPlE:BLOCK:REST:EVENT**

機能 ブロックサンプリングの休止時間をイベント数で設定/問い合わせします。

構文 :SAMPlE:BLOCK:REST:EVENT {<NRf>}  
:SAMPlE:BLOCK:REST:EVENT?  
<NRf>=イベント数(1~1,000,000)

例 :SAMPLE:BLOCK:REST:EVENT 1000  
:SAMPLE:BLOCK:REST:EVENT?  
-> :SAMPLE:BLOCK:REST:EVENT 1000

**:SAMPlE:BLOCK:REST[:MODE]**

機能 ブロックサンプリングの休止モードを設定/問い合わせします。

構文 :SAMPlE:BLOCK:REST[:MODE]  
{OFF|EVENT|TIME}  
:SAMPlE:BLOCK:REST:MODE?

例 :SAMPLE:BLOCK:REST:MODE OFF  
:SAMPLE:BLOCK:REST:MODE?  
-> :SAMPLE:BLOCK:REST:MODE OFF

**:SAMPlE:BLOCK:REST:TIME**

機能 ブロックサンプリングの休止時間を設定/問い合わせします。

構文 :SAMPlE:BLOCK:REST:TIME {<時間>}  
:SAMPlE:BLOCK:REST:TIME?  
<時間>=1 $\mu$ s~1s(100nsステップ)

例 :SAMPLE:BLOCK:REST:TIME 1us  
:SAMPLE:BLOCK:REST:TIME?  
-> :SAMPLE:BLOCK:REST:TIME 1.0E-6

**:SAMPlE:BLOCK:SIZE**

機能 ブロックサンプリングのブロック数を設定/問い合わせします。

構文 :SAMPlE:BLOCK:SIZE {<NRf>}  
:SAMPlE:BLOCK:SIZE?

●タイムスタンプ/符号間干渉解析モードのとき  
「:SAMPlE:BLOCK:REST:MODE」が「OFF」  
<NRf>=ブロック数(2~1000)  
「:SAMPlE:BLOCK:REST:MODE」が「TIME|EVENT」または「:SAMPlE:ARMIing:SOURce」が「EXTERNAL」  
<NRf>=ブロック数(2~250)

●ハードウェアヒストグラムモードのとき  
<NRf>=ブロック数(2~1000)

例 :SAMPLE:BLOCK:SIZE 10  
:SAMPLE:BLOCK:SIZE?  
-> :SAMPLE:BLOCK:SIZE 10

**:SAMPlE:BLOCK[:STATe]**

機能 ブロックサンプリングのON/OFFを設定/問い合わせします。

構文 :SAMPlE:BLOCK[:STATe] {<Boolean>}  
:SAMPlE:BLOCK:STATe?

例 :SAMPLE:BLOCK:STATE ON  
:SAMPLE:BLOCK:STATE?  
-> :SAMPLE:BLOCK:STATE 1

**:SAMPlE:GATE?**

機能 ゲートに関する全設定を問い合わせます。

構文 :SAMPlE:GATE?

例 :SAMPLE:GATE?  
-> :SAMPLE:GATE:EVENTSIZE 1000;  
MODE EVENT

**:SAMPlE:GATE:EVENTsize**

機能 ゲートをイベント数で設定/問い合わせします。

構文 :SAMPlE:GATE:EVENTsize {<NRf>}  
:SAMPlE:GATE:EVENTsize?

●タイムスタンプ/符号間干渉解析モードのとき  
周期/AtoBタイムインターバル/パルス幅測定  
<NRf>=イベント数(2~1,024,000)  
周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定  
<NRf>=イベント数(1~512,000)

●ハードウェアヒストグラムモードのとき  
周期/AtoBタイムインターバル/パルス幅測定  
<NRf>=イベント数(2~1,000,000,000)  
周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定  
<NRf>=イベント数(1~1,000,000,000)

例 :SAMPLE:GATE:EVENTSIZE 1000  
:SAMPLE:GATE:EVENTSIZE?  
-> :SAMPLE:GATE:EVENTSIZE 1000

解説 「:SAMPlE:GATE:MODE」が「EVENT」のとき、設定ができます。

**:SAMPlE:GATE[:MODE]**

機能 ゲートの種類を設定/問い合わせします。

構文 :SAMPlE:GATE[:MODE]  
{EVENT|TIME|EXTERNAL}  
:SAMPlE:GATE:MODE?

例 :SAMPLE:GATE:MODE EVENT  
:SAMPLE:GATE:MODE?  
-> :SAMPLE:GATE:MODE EVENT

#### :SAMPlE:GATE:POLarity

機能 外部ゲートの極性を設定/問い合わせします。  
 構文 :SAMPlE:GATE:POLarity  
 {POSitive|NEGative}  
 :SAMPlE:GATE:POLarity?  
 例 :SAMPlE:GATE:POLarity POSITIVE  
 :SAMPlE:GATE:POLarity?  
 -> :SAMPlE:GATE:POLarity POSITIVE  
 解説 「SAMPlE:GATE:MODE」が「EXternal」のとき、設定ができます。

#### :SAMPlE:GATE:TIME

機能 ゲートを時間で設定/問い合わせします。  
 構文 :SAMPlE:GATE:TIME {<時間>}  
 :SAMPlE:GATE:TIME?  
 <時間>=1 $\mu$ s~10s(100nsステップ)  
 例 :SAMPlE:GATE:TIME 1us  
 :SAMPlE:GATE:TIME?  
 -> :SAMPlE:GATE:TIME 1.0E-06  
 解説 「SAMPlE:GATE:MODE」が「TIME」のとき、設定ができます。

#### :SAMPlE:INHibit

機能 インヒビット入力の極性を設定/問い合わせします。  
 構文 :SAMPlE:INHibit  
 {OFF|POSitive|NEGative}  
 :SAMPlE:INHibit?  
 例 :SAMPlE:INHIBIT OFF  
 :SAMPlE:INHIBIT?  
 -> :SAMPlE:INHIBIT OFF  
 解説 "OFF"にすると、インヒビット入力は無効になります。

#### :SAMPlE:INTerval

機能 サンプリングインターバルを設定/問い合わせします。  
 構文 :SAMPlE:INTerval {<時間>|MINimum}  
 :SAMPlE:INTerval?  
 <時間>=0~1s(1 $\mu$ sステップ)  
 例 :SAMPlE:INTERVAL MINIMUM  
 :SAMPlE:INTERVAL?  
 -> :SAMPlE:INTERVAL MINIMUM  
 解説 ・データに「MINimum」を送信するか、<時間>で0を送信すると、最小のインターバルでサンプリングを行います。  
 ・測定モードがタイムスタンプのとき、設定ができます。

#### :SAMPlE:RClock

機能 サンプリングの基準クロックを設定/問い合わせします。  
 構文 :SAMPlE:RClock {INTernal|EXTernal}  
 :SAMPlE:RClock?  
 例 :SAMPlE:RCLOCK INTERNAL  
 :SAMPlE:RCLOCK?  
 -> :SAMPlE:RCLOCK INTERNAL

### 4.15 SStArtグループ



#### :SStArt

機能 シングル測定を実行します。  
 構文 :SStArt  
 例 :SSTART

### 4.16 StARtグループ

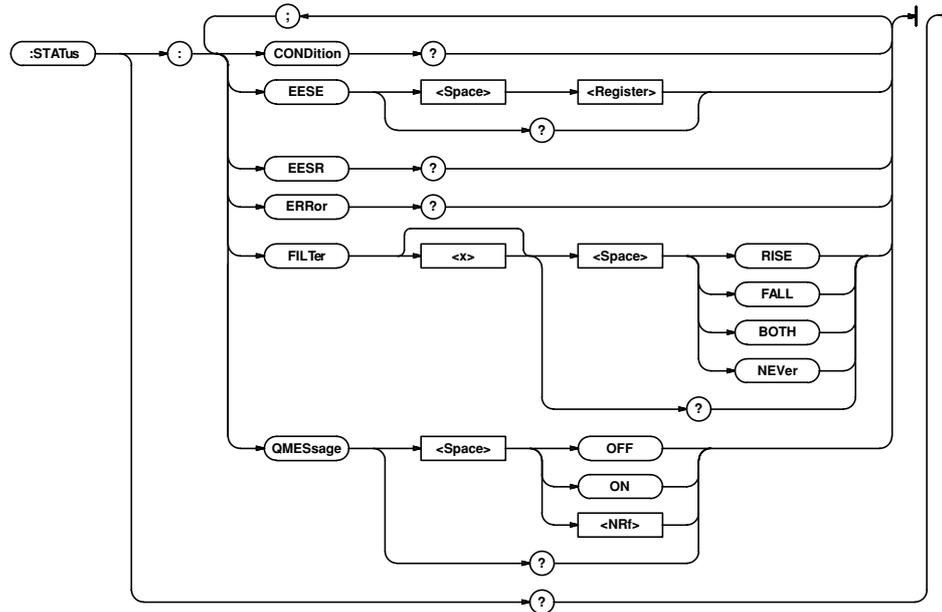


#### :StARt

機能 測定をスタートします。  
 構文 :StARt  
 例 :START

## 4.17 STATUSグループ

通信のステータスに関するグループです。ステータスレポートについては第5章を参照してください。



### :STATUS?

機能 通信のステータスに関する全設定を問い合わせします。

構文 :STATUS?

例 :STATUS?

```
-> :STATUS:EESR 0;FILTER1 RISE;
FILTER2 NEVER;FILTER3 NEVER;
FILTER4 NEVER;FILTER5 NEVER;
FILTER6 NEVER;FILTER7 NEVER;
FILTER8 NEVER;FILTER9 RISE;
FILTER10 RISE;FILTER11 RISE;
FILTER12 RISE;FILTER13 RISE;
FILTER14 NEVER;FILTER15 NEVER;
FILTER16 NEVER;QMESsAGE 1
```

### :STATUS:CONDition?

機能 状態レジスタの内容を問い合わせします。

構文 :STATUS:CONDition?

例 :STATUS:CONDITON?

```
-> 16
```

### :STATUS:EESR

機能 拡張イベントイネーブルレジスタを設定/問い合わせします。

構文 :STATUS:EESR {<Register>}

```
:STATUS:EESR?
<Register>=0~65535
```

例 :STATUS:EESR 257

```
:STATUS:EESR?
-> :STATUS:EESR 257
```

### :STATUS:EESR?

機能 拡張イベントレジスタの内容を問い合わせ、レジスタをクリアします。

構文 :STATUS:EESR?

例 :STATUS:EESR?

```
-> 1
```

### :STATUS:ERRor?

機能 発生したエラーのコードとメッセージの内容(エラーキューの先頭)を問い合わせします。

構文 :STATUS:ERRor?

例 :STATUS:ERROR?

```
-> 113, "Undefined header"
```

### :STATUS:FILTer<x>

機能 遷移フィルタを設定/問い合わせします。

構文 :STATUS:FILTer<x>

```
{RISE | FALL | BOTH | NEVer}
```

```
:STATUS:FILTer<x>?
```

```
<x>=1~16
```

例 :STATUS:FILTER2 RISE

```
:STATUS:FILTER2?
```

```
-> :STATUS:FILTER2 RISE
```

### :STATUS:QMESsage

機能 「STATUS:ERRor?」の応答にメッセージの内容を付けるか付けないかを設定/問い合わせします。

構文 :STATUS:QMESsage {<Boolean>}

```
:STATUS:QMESsage?
```

例 :STATUS:QMESsAGE OFF

```
:STATUS:QMESsAGE?
```

```
-> :STATUS:QMESsAGE 0
```

## 4.18 STOPグループ



### **:STOP**

機能 測定をストップします。

構文 **:STOP**

例 **:STOP**

## 4.19 STOReグループ



### **:STORe**

機能 現在の設定をストアします。

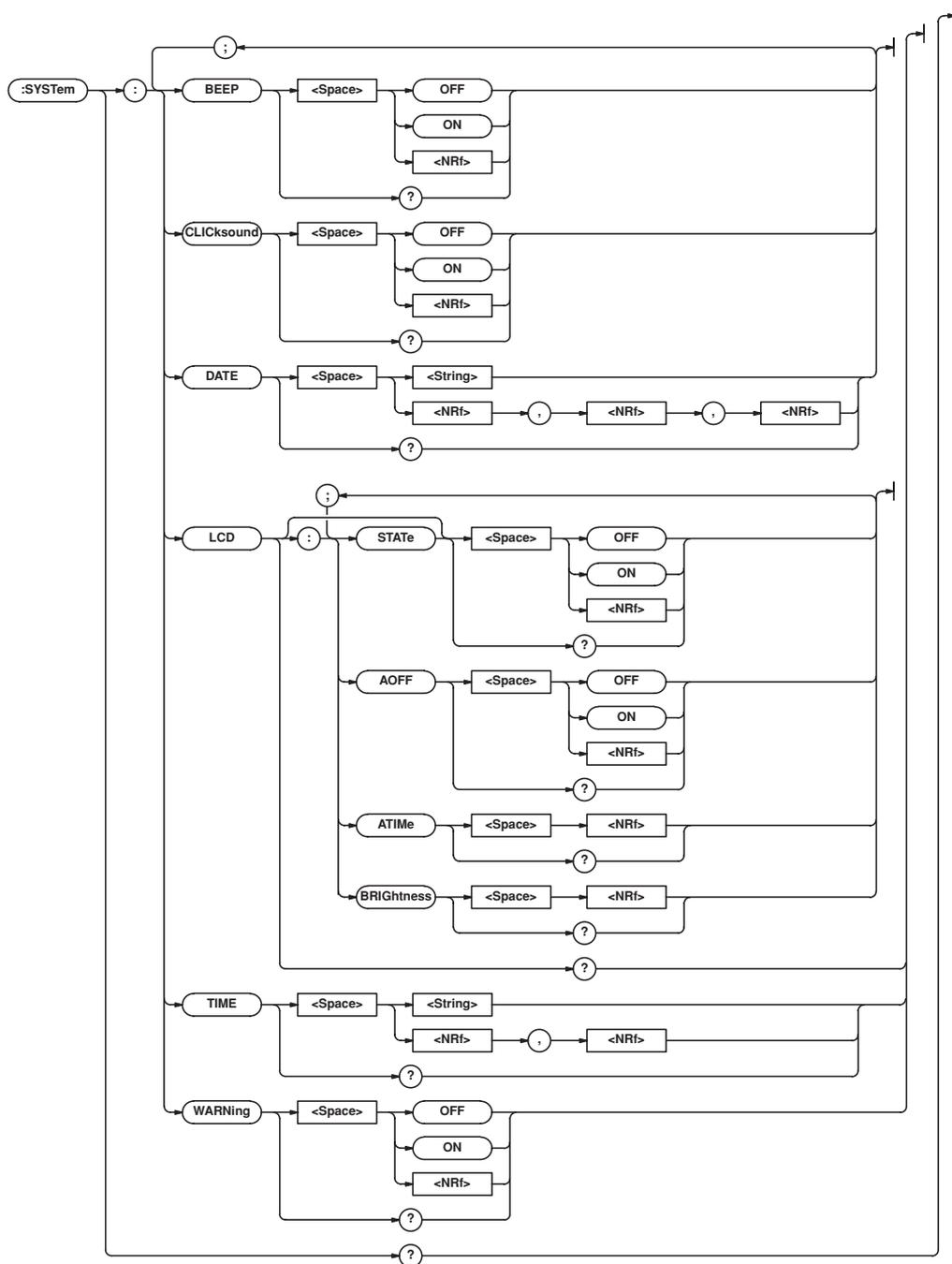
構文 **:STORe {<NRf>}**

<NRf>=0~31

例 **:STORe 0**

## 4.20 SYSTemグループ

日付/時刻設定などの基本的なシステム設定に関するグループです。フロントパネルのUTILITYキーのConfigメニュー、LCDメニューと同じ設定、および設定内容を問い合わせることができます。



### :SYSTem?

機能 SYSTemグループの全設定を問い合わせます。

構文 :SYSTem?

例 :SYSTem?

```
-> :SYSTem:BEEP 1:CLICKSOUND 1;LCD:
AOFF 0;STATE 1;BRIGHTNESS 4;;
SYSTem:WARNIng 0
```

### :SYSTem:BEEP

機能 ビープ音のON/OFFを設定/問い合わせします。

構文 :SYSTem:BEEP {<Boolean>}

:SYSTem:BEEP?

例 :SYSTem:BEEP ON

:SYSTem:BEEP?

```
-> :SYSTem:BEEP 1
```

## 4.20 SYSTemグループ

### **:SYSTem:CLICksound**

機能 クリック音のON/OFFを設定/問い合わせします。

構文 **:SYSTem:CLICksound** {<Boolean>}  
**:SYSTem:CLICksound?**

例 **:SYSTem:CLICksound ON**  
**:SYSTem:CLICksound?**  
-> **:SYSTem:CLICksound 1**

### **:SYSTem:DATE**

機能 日付を設定/問い合わせします。

構文 **:SYSTem:DATE** {<文字列>|,<NRf>,<NRf>,<NRf>}  
**:SYSTem:DATE?**

例 **:SYSTem:DATE "2002/01/01"**  
**:SYSTem:DATE?**  
-> "2002/01/01"

解説 <文字列>は上記のように, "年(4桁)/月(2桁)/日(2桁)"という書式でなければいけません。<NRf>形式の場合は, 年, 月, 日の順に記述します。

### **:SYSTem:LCD?**

機能 液晶画面に関する全設定を問い合わせます。

構文 **:SYSTem:LCD?**

例 **:SYSTem:LCD?**  
-> **:SYSTem:LCD:AOff 1;ATime 0**

### **:SYSTem:LCD:AOff**

機能 液晶画面のオートオフのON/OFFを設定/問い合わせします。

構文 **:SYSTem:LCD:AOff** {<Boolean>}  
**:SYSTem:LCD:AOff?**

例 **:SYSTem:LCD:AOff ON**  
**:SYSTem:LCD:AOff?**  
-> **:SYSTem:LCD:AOff 1**

### **:SYSTem:LCD:ATime**

機能 液晶画面のオートオフの時間を設定/問い合わせします。

構文 **:SYSTem:LCD:ATime** {<NRf>}  
**:SYSTem:LCD:ATime?**  
<NRf>=1~60(分)

例 **:SYSTem:LCD:ATime 15**  
**:SYSTem:LCD:ATime?**  
-> **:SYSTem:LCD:ATime 15**

解説 「**:SYSTem:LCD:AOff**」が「ON」のとき, 設定ができます。

### **:SYSTem:LCD:BRIGhtness**

機能 液晶画面の明るさを設定/問い合わせします。

構文 **:SYSTem:LCD:BRIGhtness** {<NRf>}  
**:SYSTem:LCD:BRIGhtness?**  
<NRf>=0~7

例 **:SYSTem:LCD:BRIGhtness**  
**:SYSTem:LCD:BRIGhtness?**  
-> **:SYSTem:LCD:BRIGhtness**

### **:SYSTem:LCD[:STATe]**

機能 液晶画面のバックライトのON/OFFを設定/問い合わせします。

構文 **:SYSTem:LCD[:STATe]** {<Boolean>}  
**:SYSTem:LCD:STATe?**

例 **:SYSTem:LCD:STATe ON**  
**:SYSTem:LCD:STATe?**  
-> **:SYSTem:LCD:STATe 1**

### **:SYSTem:TIME**

機能 時刻を設定/問い合わせします。

構文 **:SYSTem:TIME** {<文字列>|,<NRf>,<NRf>}  
**:SYSTem:TIME?**

例 **:SYSTem:TIME "12:00"**  
**:SYSTem:TIME?**  
-> "12:00:00"

解説 <文字列>は上記のように, "時(2桁):分(2桁)"という書式でなければいけません。<NRf>形式の場合は, 時, 分の順に記述します。クエリの応答はこれに秒が加わります。

### **:SYSTem:WARning**

機能 ワーニングを表示する/しないを設定/問い合わせします。

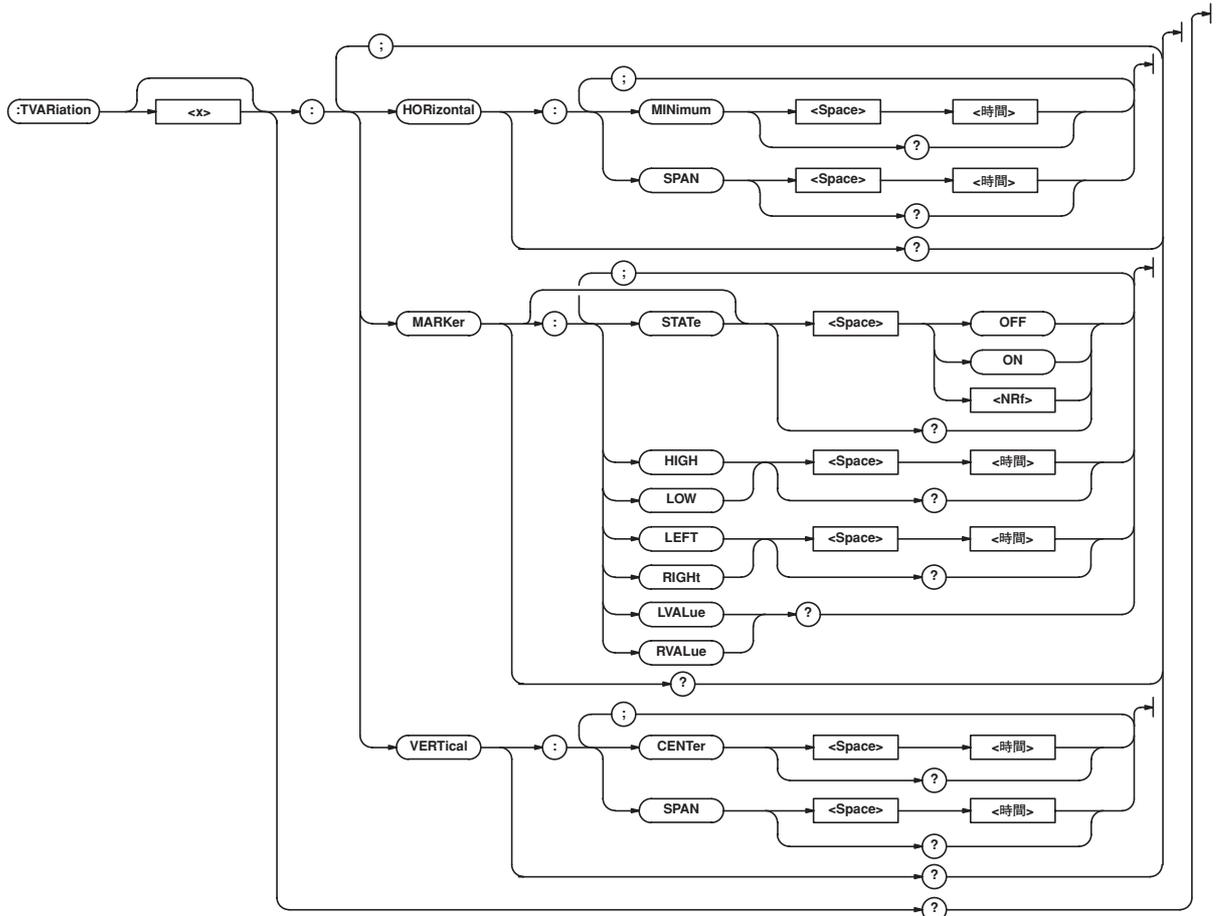
構文 **:SYSTem:WARning** <Boolean>  
**:SYSTem:WARning?**

例 **:SYSTem:WARning ON**  
**:SYSTem:WARning?**  
-> **:SYSTem:WARning 1**

## 4.21 TVARiation<x>グループ

タイムスタンプモードのタイムバリエーション表示に関するグループです。フロントパネルのDISPLAY, SCALE, MARKキーと同じ設定、および設定内容の問い合わせができます。

- ・ 周期/AtoBタイムインターバル/パルス幅測定、または周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定で測定1に対するコマンドは、「TVARiation1」または、「TVARiation」を使用してください。ただし、クエリは、「TVARIATION1」となります。
- ・ 周期A&周期B/周期A&AtoBタイムインターバル/パルス幅A&AtoBタイムインターバル/パルス幅A&パルス幅B測定で、測定2に対するコマンドは、「TVARiation2」を使用してください。



### :TVARiation<x>?

機能 タイムバリエーションに関する全設定を問い合わせます。

構文 :TVARiation<x>?  
<x>= 1~2

例 :TVARIATION?  
-> :TVARIATION1:HORIZONTAL:  
MINIMUM 100.0E-09;SPAN 0.06E-03;;  
TVARIATION1:MARKER:  
HIGH 150.000E-09;LEFT 1.0E-06;  
LOW 100.000E-09;RIGHT 2.0E-06;  
STATE 1:TVARIATION1:VERTICAL:  
CENTER 100.000E-09;SPAN 0.2E-06

### :TVARiation<x>:HORizontal?

機能 タイムバリエーション表示の水平軸(X軸)に関する全設定を問い合わせます。

構文 :TVARiation<x>:HORizontal?  
<x>= 1~2

例 :TVARIATION:HORIZONTAL?  
-> :TVARIATION1:HORIZONTAL:  
MINIMUM 100.0E-09;SPAN 0.06E-03

## 4.21 TVARiation<x>グループ

### :TVARiation<x>:HORizontal:MINimum

機能 水平軸(X軸)スケールの左端を設定/問い合わせします。

構文 :TVARiation<x>:HORizontal:MINimum {<時間>}  
:TVARiation<x>:HORizontal:MINimum? <x>=1~2

<時間>=0~320s(100nsステップ)

例 :TVARIATION1:HORIZONTAL:MINIMUM 100ns  
:TVARIATION1:HORIZONTAL:MINIMUM?  
-> :TVARIATION1:HORIZONTAL:MINIMUM 100.0E-09

### :TVARiation<x>:HORizontal:SPAN

機能 水平軸(X軸)スケールの幅を設定/問い合わせします。

構文 :TVARiation<x>:HORizontal:SPAN {<時間>}  
:TVARiation<x>:HORizontal:SPAN? <x>=1~2

<時間>=6μs, 12μs, 30μs, 60μs, 120μs, 300μs, 600μs, 1.2ms, 3ms, 6ms, 12ms, 30ms, 60ms, 120ms, 300ms, 600ms, 1.2s, 3s, 6s, 12s, 30s, 60s, 120s, 300s, 600s

例 :TVARIATION1:HORIZONTAL:SPAN 60us  
:TVARIATION1:HORIZONTAL:SPAN?  
-> :TVARIATION1:HORIZONTAL:SPAN 0.06E-03

### :TVARiation<x>:MARKer?

機能 マーカーに関する全設定を問い合わせます。

構文 :TVARiation<x>:MARKer? <x>=1~2

例 :TVARIATION1:MARKer?  
-> :TVARIATION1:MARKer:  
HIGH 150.000E-09;LEFT 1.0E-06;  
LOW 100.000E-09;RIGHT 2.0E-06;  
STATE 1

### :TVARiation<x>:MARKer:{HIGH|LOW}

機能 ハイ/ローマーカーの位置を設定/問い合わせします。

構文 :TVARiation<x>:MARKer:{HIGH|LOW} {<時間>}  
:TVARiation<x>:MARKer:{HIGH|LOW}? <x>=1~2

<時間>=マーカー位置(25psステップ)

(以下は、ハイマーカーにおける例です。)

例 :TVARIATION1:MARKer:HIGH 150ns  
:TVARIATION1:MARKer:HIGH?  
-> :TVARIATION1:MARKer:  
HIGH 150.000E-09

### :TVARiation<x>:MARKer:{LEFT|RIGHT}

機能 レフト/ライトマーカーの位置を設定/問い合わせします。

構文 :TVARiation:MARKer:{LEFT|RIGHT} {<時間>}  
:TVARiation:MARKer:{LEFT|RIGHT}? <x>=1~2

<時間>=0~920s(100nsステップ)

(以下は、レフトマーカーにおける例です。)

例 :TVARIATION1:MARKer:LEFT 150us  
:TVARIATION1:MARKer:LEFT?  
-> :TVARIATION1:MARKer:  
LEFT 150.0E-03

### :TVARiation<x>:MARKer:{LVALue|RVALue}?

機能 レフト/ライトマーカー位置の測定値を問い合わせます。

構文 :TVARiation<x>:MARKer:  
{LVALue|RVALue}? <x>=1~2

(以下は、レフトマーカー位置における例です。)

例 :TVARIATION1:MARKer:LVALUE?  
-> 1.2000E-06

### :TVARiation<x>:MARKer[:STATE]

機能 マーカーカーソルのON/OFFを設定/問い合わせします。

構文 :TVARiation<x>:MARKer[:STATE] {<Boolean>}  
:TVARiation<x>:MARKer:STATE? <x>=1~2

例 :TVARIATION1:MARKer:STATE ON  
:TVARIATION1:MARKer:STATE?  
-> :TVARIATION1:MARKer:STATE 1

### :TVARiation<x>:VERTical?

機能 垂直軸(Y軸)に関する全設定を問い合わせます。

構文 :TVARiation<x>:VERTical? <x>=1~2

例 :TVARIATION1:VERTical:CENTer?  
-> :TVARIATION1:VERTical:CENTer  
100.000E-09;SPAN 0.2E-06

### :TVARiation<x>:VERTical:CENTer

機能 垂直軸(Y軸)の中央値を設定/問い合わせします。

構文 :TVARiation<x>:VERTical:CENTer {<時間>}  
:TVARiation<x>:VERTical:CENTer? <x>=1~2

<時間>=-50ns~20ms(25psステップ)

例 :TVARIATION1:VERTical:CENTer 100ns  
:TVARIATION1:VERTical:CENTer?  
-> :TVARIATION1:VERTical:  
CENTer 100.000E-09

**:TVARiation<x>:VERTical:SPAN**

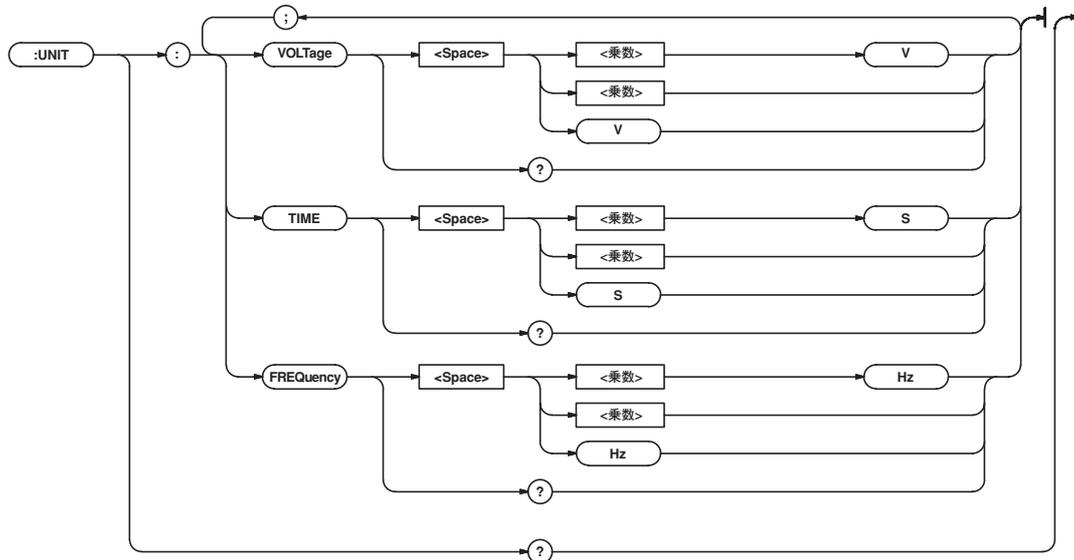
機能 垂直軸(Y軸)の幅を設定/問い合わせします。

構文 **:TVARiation<x>:VERTical:SPAN {<時間>}**  
**:TVARiation<x>:VERTical:SPAN?**  
 <x>= 1~2  
 <時間>=500ps, 1ns, 2.5ns, 5ns, 10ns, 20ns, 50ns, 100ns, 200ns, 500ns, 1μs, 2μs, 5μs, 10μs, 20μs, 50μs, 100μs, 200μs, 500μs, 1ms, 2ms, 5ms, 10ms, 20ms

例 **:TVARIATION1:VERTICAL:SPAN 200ns**  
**:TVARIATION1:VERTICAL:SPAN?**  
 -> **:TVARIATION1:VERTICAL:SPAN 0.2E-06**

**4.22 UNITグループ**

電圧、時間、周波数の各データのデフォルトの単位に関するグループです。

**:UNIT?**

機能 電圧/時間/周波数のデフォルトの単位を問い合わせます。

構文 **:UNIT?**

例 **:UNIT?**  
 -> **:UNIT:FREQUENCY HZ;VOLTAGE V; TIME S**

**:UNIT:VOLTage**

機能 電圧のデフォルトの単位を設定/問い合わせします。

構文 **:UNIT:VOLTage {<乗数>}V**  
**:UNIT:VOLTage?**  
 <乗数> 3.4節を参照してください。

例 **:UNIT:VOLTAGE V**  
**:UNIT:VOLTAGE?**  
 -> **:UNIT:VOLTAGE V**

**:UNIT:TIME**

機能 時間のデフォルトの単位を設定/問い合わせします。

構文 **:UNIT:TIME {<乗数>}S**  
**:UNIT:TIME?**  
 <乗数> 3.4節を参照してください。

例 **:UNIT:TIME s**  
**:UNIT:TIME?**  
 -> **:UNIT:TIME S**

**:UNIT:FREQUENCY**

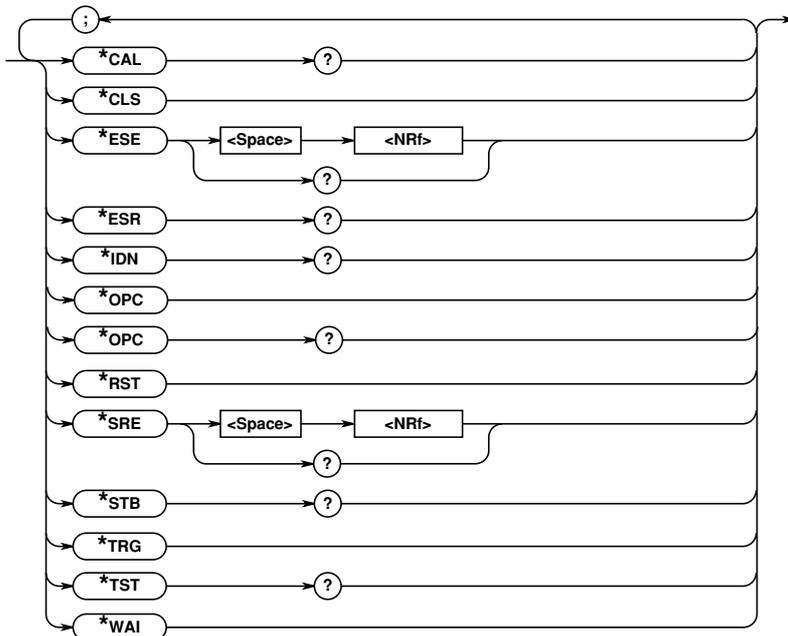
機能 周波数のデフォルトの単位を設定/問い合わせします。

構文 **:UNIT:FREQUENCY {<乗数>}Hz**  
**:UNIT:FREQUENCY?**  
 <乗数>3.4節を参照してください。

例 **:UNIT:FREQUENCY HZ**  
**:UNIT:FREQUENCY?**  
 -> **:UNIT:FREQUENCY HZ**

### 4.23 共通コマンドグループ

IEEE 488.2-1992で規定されているコマンドで、機器固有の機能に依存しないコマンドのグループです。



**\*CAL?**

機能 キャリブレーションを実行し、結果を問い合わせます。  
 構文 \*CAL?  
 例 \*CAL?  
 -> 0  
 解説 キャリブレーションが正常に終了したときは「0」、異常があるときは0以外の値が返されます。

**\*CLS**

機能 標準イベントレジスタ、拡張イベントレジスタ、エラーキューをクリアします。  
 構文 \*CLS  
 例 \*CLS  
 解説 標準イベントレジスタ、拡張イベントレジスタ、エラーキューについては、「第5章 ステータスレポート」をご覧ください。

**\*ESE**

機能 標準イベントイネーブルレジスタ値を設定/問い合わせします。  
 構文 \*ESE {<NRf>}  
 \*ESE?  
 <NRf>=0~255  
 例 \*ESE 253  
 \*ESE?  
 -> 253  
 解説 標準イベントイネーブルレジスタについては、「第5章 ステータスレポート」をご覧ください。

**\*ESR?**

機能 標準イベントレジスタ値を問い合わせ、レジスタをクリアをします。  
 構文 \*ESR?  
 例 \*ESR?  
 -> 253  
 解説 標準イベントレジスタについては、「第5章 ステータスレポート」をご覧ください。

**\*IDN?**

機能 機種を問い合わせます。  
 構文 \*IDN?  
 例 \*IDN?  
 -> YOKOGAWA,704510,0,F1.01  
 解説 メーカー名、機種名、シリアル番号(未使用:常に0)、ファームウェアバージョンが返されます。

**\*OPC**

機能 指定したオーバーラップコマンドが終了したときに、標準イベントレジスタのビットに「1」をセットします。TA720ではオーバーラップコマンドをサポートしていないため、このコマンドを送っても無視します。  
 構文 \*OPC

**\*OPC?**

機能 指定したオーバーラップコマンドが終了していれば、「1」を返します。TA720ではオーバーラップコマンドをサポートしていないため、常に「1」を返します。  
 構文 \*OPC?

**\*RST**

機能 設定情報の初期化(イニシャライズ)をします。  
 構文 \*RST  
 例 \*RST  
 解説 INIT(SHIFT+MARKER)キーの[Execute]ソフトキーを押したときと同じ動作をします。初期化の内容は本体ユーザーズマニュアルをご覧ください。

**\*SRE**

機能 サービスリクエストイネーブルレジスタ値を設定/問い合わせします。  
 構文 \*SRE {<NRf>}  
 \*SRE?  
 <NRf>=0~255  
 例 \*SRE 239  
 \*SRE?  
 -> 239

**\*STB?**

機能 ステータスバイトレジスタ値を問い合わせます。  
 構文 \*STB?  
 例 \*STB?  
 -> 4  
 解説 ステータスバイトレジスタについては、「第5章 ステータスレポート」をご覧ください。

**\*TRG?**

機能 SINGLEキーを押したときと同じ動作をします。  
 構文 \*TRG  
 解説 マルチラインメッセージGET(Group Execute Trigger)もこのコマンドと同じ動作をします。

**\*TST?**

機能 セルフテストを実行し結果を問い合わせます。  
 構文 \*TST?  
 例 \*TST?  
 -> 0  
 解説 ・ [Selftest]ソフトキーの[Board]テストと同じ動作をします。  
 ・ すべてのテストが正常終了したときは「0」, 異常があるときは0以外の値が返されます。

**\*WAI**

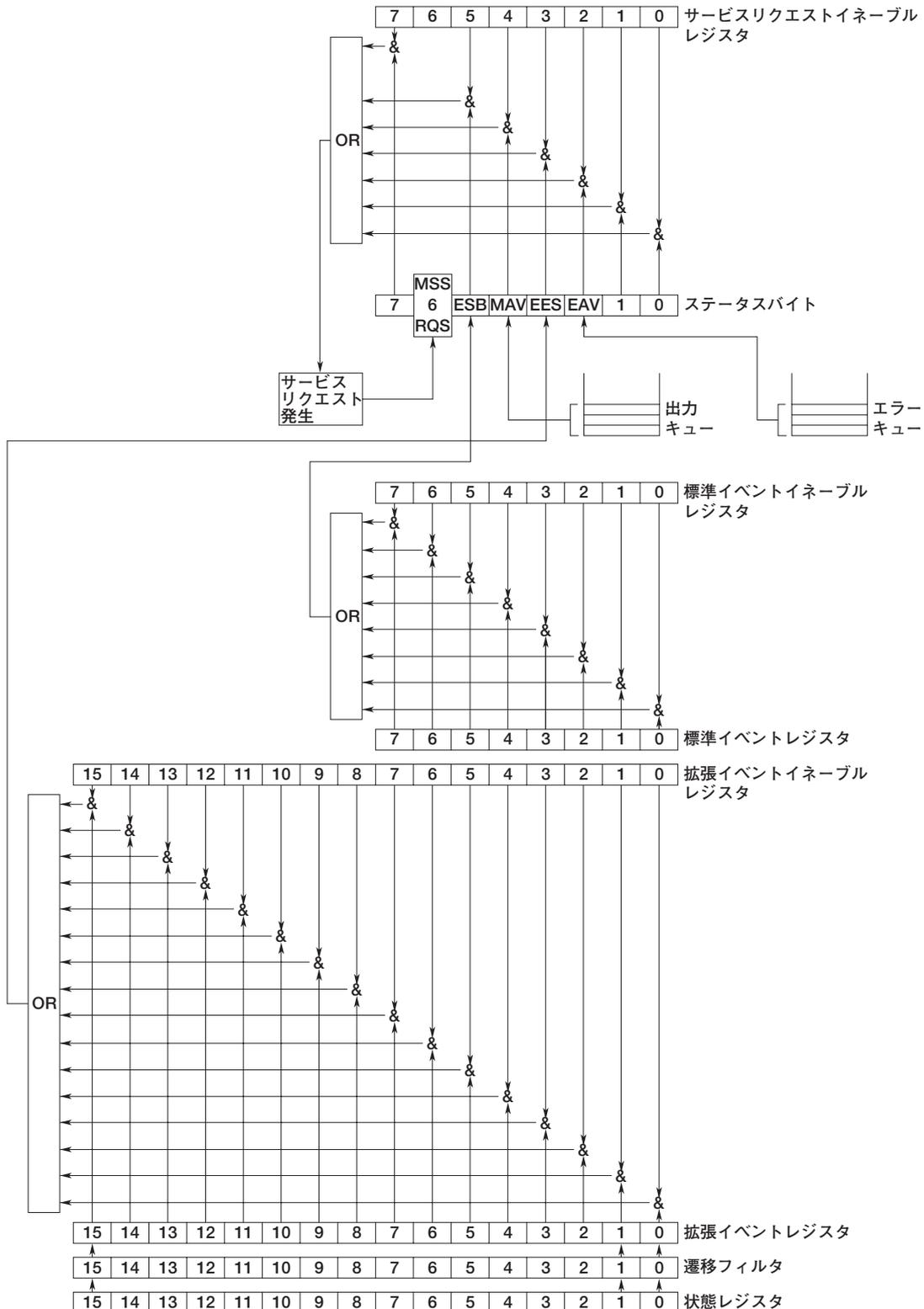
機能 指定したオーバーラップコマンドが終了するまで \*WAIに続く命令を待ちます。TA720ではオーバーラップコマンドをサポートしていないため、このコマンドを送っても無視します。  
 構文 \*WAI

# 第5章 ステータスレポート

## 5.1 ステータスレポートについて

### ステータスレポート

シリアルポールで読まれるステータスレポートは下図のとおりです。これは、IEEE 488.2-1992で規定されたものを拡張したものです。



## 各レジスタ・キューの概要

名称	機能	書き込み	読み出し
ステータスバイト		—	シリアルポール (RQS), *STB?(MSS)
サービスリクエストイネーブルレジスタ	ステータスバイトのマスク	*SRE	*SRE?
標準イベントレジスタ	機器の状態の変化	—	*ESR?
標準イベントイネーブルレジスタ	標準イベントレジスタのマスク	*ESE	*ESE?
拡張イベントレジスタ	機器の状態の変化	—	STATUS:ESR?
拡張イベントイネーブルレジスタ	拡張イベントレジスタのマスク	STATUS:EESE	STATUS:EESE?
状態レジスタ	現在の機器の状態	—	STATUS:CONDition?
遷移フィルタ	拡張イベントレジスタの変化の条件	STATUS:FILTer <x>	STATUS:FILTer <x>?
出力キュー	問い合わせに対する応答メッセージを格納	各問い合わせコマンド	
エラーキュー	エラーNo.とメッセージを格納	—	STATUS:ERRor?

## ステータスバイトに影響を与えるレジスタとキュー

ステータスバイトの各ビットに影響を与えるレジスタを整理すると、次のようになります。

標準イベントレジスタ	：ステータスバイトのビット5(ESB)を1/0にセット
出力キュー	：ステータスバイトのビット4(MAV)を1/0にセット
拡張イベントレジスタ	：ステータスバイトのビット3(EES)を1/0にセット
エラーキュー	：ステータスバイトのビット2(EAV)を1/0にセット

## 各イネーブルレジスタ

各ビットをマスクして、そのビットが1であってもステータスバイトの要因にしないようにできるレジスタを整理すると、次のようになります。

ステータスバイト	：サービスリクエストイネーブルレジスタにより、各ビットをマスク
標準イベントレジスタ	：標準イベントイネーブルレジスタにより、各ビットをマスク
拡張イベントレジスタ	：拡張イベントイネーブルレジスタにより、各ビットをマスク

## 各レジスタの書き込み/読み出し

たとえば、標準イベントイネーブルレジスタの各ビットを1または0にするには、\*ESEコマンドを使います。また、標準イベントイネーブルレジスタの各ビットが1であるか0であるかを確認するには、\*ESE?コマンドを使います。これらの各コマンドについては、第4章で詳しく説明しています。

## 5.2 ステータスバイト

## ステータスバイト



- **ビット0, 1, 7**  
未使用(常に0)
- **ビット2 EAV(Error Available)**  
エラーキューが空でないときに1にセットされます。つまり、エラーが発生すると1になります。5-5ページを参照してください。
- **ビット3 EES(Extend Event Summary Bit)**  
拡張イベントレジスタと、そのイネーブルレジスタの各ビットの論理和が1のときに、1にセットされます。つまり、機器の内部であるイベントが起こったときに1になります。5-4ページを参照してください。
- **ビット4 MAV(Message Available)**  
出力キューが空でないときに1にセットされます。つまり、問い合わせを行って出力するべきデータがあるときに1になります。5-5ページを参照してください。
- **ビット5 ESB(Event Summary Bit)**  
標準イベントレジスタと、そのイネーブルレジスタの各ビットの論理和が1のときに、1にセットされます。つまり、機器の内部であるイベントが起こったときに1になります。5-3ページを参照してください。
- **ビット6 RQS(Request Service)/MSS(Master Status Summary)**  
ビット6以外のステータスバイトと、サービスリクエストイネーブルレジスタの論理積が0でないときに、1にセットされます。つまり、機器がコントローラにサービス要求をしているときに1になります。RQSは、MSSが0から1になったときに1にセットされ、シリアルポールが、MSSが0になったときにクリアされます。

## 各ビットのマスク

ステータスバイトのあるビットをマスクしてSRQの要因にしたくないときには、サービスリクエストイネーブルレジスタの対応するビットを0にします。たとえば、ビット2(EAV)をマスクして、エラーが発生してもサービスを要求しないようにするには、サービスリクエストイネーブルレジスタのビット2を0にします。これは\*SREコマンドで行います。また、サービスリクエストイネーブルレジスタの各ビットが1であるか0であるかは、\*SRE?で問い合わせられます。\*SREコマンドについては、第4章をお読みください。

### ステータスバイトの動作

ステータスバイトのビット6が1になると、サービスリクエストが発生します。ビット6以外のどれかのビットが1になると、ビット6が1になります(サービスリクエストイネーブルレジスタの対応するビットも1のとき)。たとえば、何かのイベントが起こって、標準イベントレジスタとそのイネーブルレジスタの各ビットの論理和が1になったときは、ビット5(ESB)が1にセットされます。このとき、サービスリクエストイネーブルレジスタのビット5が1であれば、ビット6(MSS)が1にセットされ、コントローラにサービスを要求します。

また、ステータスバイトの内容を読むことにより、どんな種類のイベントが起こったのかを確認することができます。

### ステータスバイトの読み出し

ステータスバイトの内容を読み出すには、次の2つの方法があります。

#### ● \*STB?による問い合わせ

\*STB?で問い合わせると、ビット6はMSSになります。したがって、MSSを読み出すことになります。読み出したあとは、ステータスバイトのどのビットもクリアしません。

#### ● シリアルポール

シリアルポールを実行すると、ビット6はRQSになります。したがって、RQSを読み出すことになります。読み出したあと、RQSだけをクリアします。シリアルポールではMSSを読み出すことはできません。

### ステータスバイトのクリア

ステータスバイトの全ビットを強制的にクリアする方法はありません。各動作に対してクリアされるビットを以下に示します。

#### ● \*STB?で問い合わせたとき

どのビットもクリアされません。

#### ● シリアルポールを実行したとき

RQSビットだけがクリアされます。

#### ● \*CLSコマンドを受信したとき

\*CLSコマンドを受信すると、ステータスバイト自体はクリアされませんが、各ビットに影響する標準イベントレジスタなどの内容がクリアされます。その結果、それに対応したステータスバイトのビットがクリアされます。ただし、出力キューは\*CLSコマンドではクリアできないので、ステータスバイトのビット4(MAV)は影響を受けません。ただし、\*CLSコマンドをプログラムメッセージターミネータのすぐあとに受信したときは、出力キューもクリアされます。

## 5.3 標準イベントレジスタ

### 標準イベントレジスタ

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

- **ビット7 PON(Power ON) 電源ON**  
本機器の電源がONになったときに、1になります。
- **ビット6 URQ(User Request) ユーザーリクエスト**  
未使用(常に0)
- **ビット5 CME(Command Error) コマンド文法エラー**  
コマンドの文法に誤りがあるときに、1になります。  
例 コマンド名のつづりの誤り、8進データ中に「9」がある
- **ビット4 EXE(Execution Error) コマンド実行エラー**  
コマンドの文法は正しいが、現在の状態では実行不可能なときに、1になります。  
例 パラメータが設定範囲外
- **ビット3 DDE(Device Error) 機器特有のエラー**  
コマンド文法エラー、コマンド実行エラー以外の機器の内部的原因で、コマンドが実行できなかったときに、1になります。
- **ビット2 QYE(Query Error) 問い合わせエラー**  
問い合わせコマンドを送信したが、出力キューが空かデータが失われていたときに、1になります。  
例 応答データがない、出力キューがあふれてデータが失われた
- **ビット1 RQC(Request Control) リクエストコントロール**  
未使用(常に0)
- **ビット0 OPC(Operation Complete) 操作終了**  
\*OPCコマンド(第4章参照)によって指定された動作が終了したときに、1になります。

### 各ビットのマスク

標準イベントレジスタのあるビットをマスクして、ステータスバイトのビット5(ESB)の要因にしたいときには、標準イベントイネーブルレジスタの対応するビットを0にします。

たとえば、ビット2(QYE)をマスクして問い合わせエラーが発生してもESBを1にしないようにするには、標準イベントイネーブルレジスタのビット2を0にします。これは\*ESEコマンドで行います。また、標準イベントイネーブルレジスタの各ビットが1であるか0であるかは、\*ESE?で問い合わせられます。\*ESEコマンドについては、第4章をお読みください。

### 5.3 標準イベントレジスタ/5.4 拡張イベントレジスタ

#### 標準イベントレジスタの動作

標準イベントレジスタは、機器の内部に起こった8種類のイベントに対するレジスタです。どれかのビットが1になると、ステータスバイトのビット5(ESB)を1にセットします(標準イベントイネーブルレジスタの対応するビットも1のとき)。

例

1. 問い合わせエラー発生
2. ビット2(QYE)が1にセットされる
3. 標準イベントイネーブルレジスタのビット2が1ならば、ステータスバイトのビット5(ESB)が1にセットされる

また、標準イベントレジスタの内容を読むことにより、機器の内部に起こったイベントの種類を確認することができます。

#### 標準イベントレジスタの読み出し

標準イベントレジスタの内容は、\*ESR?で読み出すことができます。読み出されたあとは、レジスタはクリアされます。

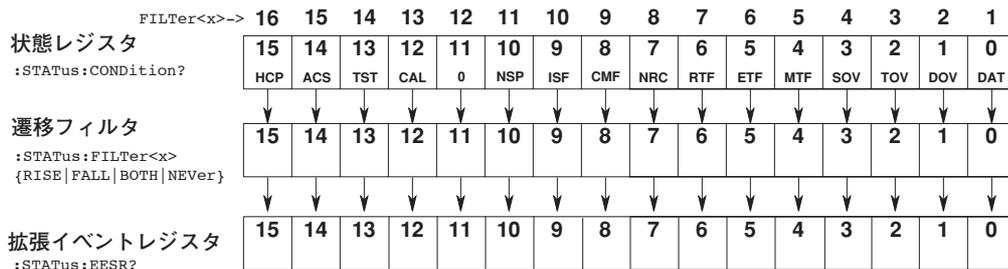
#### 標準イベントレジスタのクリア

標準イベントレジスタがクリアされるのは、次の3つの場合です。

- ・ \*ESR?で標準イベントレジスタの内容が読み出されたとき
- ・ \*CLSコマンドを受信したとき
- ・ 電源再投入時

## 5.4 拡張イベントレジスタ

拡張イベントレジスタは、機器の内部状態を表す状態レジスタの状態変化が、遷移フィルタでエッジ検出された結果が入ります。



状態レジスタの各ビットの意味は、次のとおりです。

ビット0	DAT(DATa available)	測定データや演算結果が有効なときに1になります。
ビット1	DOV(Data OVerflow)	測定データがオーバーフローを起こしたときに1になります。
ビット2	TOV(Time stamp OVerflow)	タイムスタンプデータがオーバーフローを起こしたときに1になります。
ビット3	SOV(Sample OVerflow)	測定サンプル数がオーバーフローを起こしたときに1になります。
ビット4	MTF(Measured T Failure)	Measured T測定に失敗したときに1になります。
ビット5	ETF(Estimated T Failure)	Estimated T測定に失敗したときに1になります。
ビット6	RTF(Rest Time Failure)	ブロックサンプリングの休止時間が500nsに満たないときに1になります。
ビット7	NRC(No Reference Clock)	リファレンス信号がないときに1になります。
ビット8	CMF(Continuation Measurement Failure)	連続測定に失敗したときに1になります。
ビット9	ISF(ISI analysis Failure)	符号間干渉解析に失敗したときに1になります。
ビット10	NSP(No Sync Pattern)	シンボルサーチに失敗したときに1になります。
ビット11	Reserved	未使用
ビット12	CAL(CALibration)	キャリブレーション実行中に1になります。
ビット13	TST(TeSTing)	セルフテスト中に1になります。
ビット14	ACS(file ACceSsing)	メディアへのアクセス中に1になります。
ビット15	HCP(Hard-CoPying)	ハードコピー中に1になります。

遷移フィルタのパラメータは、状態レジスタの指定されたビット(数値サフィックス1~16)の変化を次のように抽出し、拡張イベントレジスタを書き換えます。

RISE	0→1の変化で、拡張イベントレジスタの指定ビットを、1にします。
FALL	1→0の変化で、拡張イベントレジスタの指定ビットを、1にします。
BOTH	0→1または1→0の変化で、拡張イベントレジスタの指定ビットを、1にします。
NEVer	常に0。

## 5.5 出力キューとエラーキュー

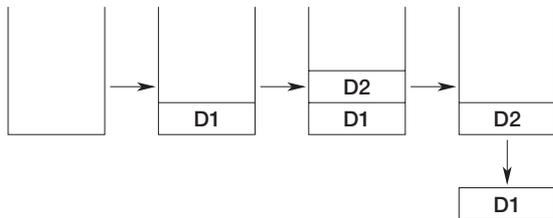
### 出力キュー

出力キューは、問い合わせ(クエリ)に対する応答メッセージを格納します。たとえば、測定データの出力を要求するMEMory:SEND?を送信すると、そのデータはそれが読み出されるまで出力キューに蓄えられます。

以下の例のように、データは順番に蓄えられ、古いものから読み出されます。読み出す以外にも、次のときに出力キューは空になります。

- ・新しいメッセージをコントローラから受信したとき
- ・デッドロック状態になったとき(3-2ページ参照)
- ・デバイスクリア(DCLまたはSDC)を受信したとき
- ・電源の再投入

なお、\*CLSコマンドでは出力キューを空にすることはできません。出力キューが空であるかどうかは、ステータスバイトのビット4(MAV)で確認できます。



### エラーキュー

エラーキューは、エラーが発生したときにその番号とメッセージを格納します。たとえば、コントローラが間違ったプログラムメッセージを送信したら、エラーが表示されたときに「113, "Undefined header"」という番号とエラーメッセージがエラーキューに格納されます。

エラーキューの内容は、STATus:ERRor?クエリで読み出すことができます。エラーキューは出力キューと同様に古いものから読み出されます。

エラーキューがあふれたときは、最後のメッセージを「350, "Queue overflow"」というメッセージに置き換えます。

読み出す以外にも次のときにエラーキューは空になります。

- ・\*CLSコマンドを受信したとき
- ・電源の再投入

なお、エラーキューが空であるかどうかは、ステータスバイトのビット2(EAV)で確認できます。

## 第6章 サンプルプログラム

### 6.1 プログラムを組む前に

#### 対象モデルと言語

対象モデル： MS-DOS/Vマシン  
 対象言語： Visual Basic Ver6.0 Professional Edition以上  
 GPIBボード： NI社製 AT-GPIB/TNT IEEE-488.2

#### Visual Basic上での設定

使用標準モジュール： Niglobal.bas  
 Vbib-32.bas  
 tmctl.bas  
 tmval.bas

#### Note

「tmctl.bas」および「tmval.bas」は、弊社Webサイトの下記のURLから、サンプルプログラムとともにダウンロードすることができます。

<http://www.yokogawa.co.jp/Measurement/F-SOFT/sample.htm>

#### 本機器の設定

##### GP-IB

この章のサンプルプログラムは、すべて本機器のGP-IBアドレスを「1」として作成しています。

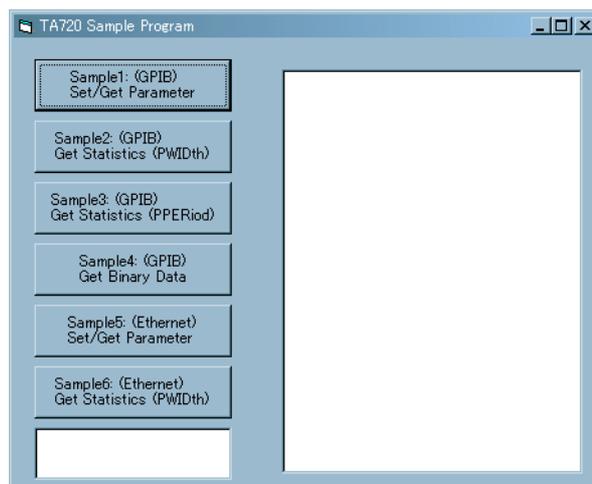
1-4ページの操作説明に従って、GP-IBアドレスを「1」に設定してください。

##### Ethernet

この章のサンプルプログラムは、すべて本機器の設定を「IPアドレス：11.22.33.44」「ユーザー名：anonymous」，「パスワード：なし」として作成しています。

2-4ページの操作説明に従って、ユーザー名/パスワードの設定、Time Outの設定、TCP/IPの設定を行ってください。

### 6.2 サンプルプログラムイメージ



## 6.3 初期化・エラー・実行関数

```

Option Explicit
Dim StartFlag As Integer           'Start Flag
Dim addr As Integer                'GPIOB Address
Dim Timeout As Integer             'Timeout
Dim Dev As Integer                 'Device ID(GPIOB)
Dim eDev As Long                  'Device ID(Ethernet)
Dim CtsFlag As Integer             'CTS Flag
Dim term As String                 'Terminator
Dim Query(100) As String           'Query String
Dim Dummy As Integer

-----
Private Function InitGpib() As Integer
    Dim eos As Integer              'EOS
    Dim eot As Integer              'EOI
    Dim brd As Integer              'GPIOB Board ID

    eos = &HCOA                     'Terminator = LF
    eot = 1                           'EOI = Enable
    term = Chr(10)                    'Terminator = LF
    Timeout = T10s                     'Timeout = 10s

    brd = ifind("GPIOB")
    If (brd < 0) Then
        Call DisplayGPIBError(brd, "ifind")
        InitGpib = 1
        Exit Function
    End If
    Dev = ildev(0, addr, 0, Timeout, eot, eos)
    If (Dev < 0) Then
        Call DisplayGPIBError(Dev, "ildev")
        InitGpib = 1
        Exit Function
    End If
    sts = ilsic(brd)                  'Set IFC
    If (sts < 0) Then
        Call DisplayGPIBError(sts, "ilsic")
        InitGpib = 1
        Exit Function
    End If
    InitGpib = 0
End Function

-----
Private Function InitEthernet() As Integer
    Dim ret As Long

    ret = TmInitialize(4, "11.22.33.44,anonymous,", eDev)
    If (ret <> 0) Then
        Call DisplayEthernetError(eDev, "TmInitialize")
        InitEthernet = 1
        Exit Function
    End If

    ret = TmSetTerm(eDev, 2, 1)
    If (ret <> 0) Then
        Call DisplayEthernetError(eDev, "TmSetTerm")
        InitEthernet = 1
        Exit Function
    End If

    ret = TmSetTimeout(eDev, 300)      'Timeout = 30s
    If (ret <> 0) Then
        Call DisplayEthernetError(eDev, "TmSetTimeout")
        InitEthernet = 1
        Exit Function
    End If

    ret = TmSetRen(eDev, 1)
    If (ret <> 0) Then
        Call DisplayEthernetError(eDev, "TmSetRen")
        InitEthernet = 1
        Exit Function
    End If

    InitEthernet = 0
End Function

-----
Private Sub DisplayGPIBError(ByVal sts As Integer, ByVal msg As String)
    Dim wrn As String
    Dim ers As String
    Dim ern As Integer

    If (sts And TIMO) Then
        wrn = "Time out" + Chr(13)
    Else
        wrn = ""
    End If

```

```

If (sts And EERR) Then
    ern = iberr
    If (ern = EDVR) Then
        ers = "EDVR:System error"
    ElseIf (ern = ECIC) Then
        ers = "ECIC:Function requires GPIB board to be CIC"
    ElseIf (ern = ENOL) Then
        ers = "ENOL:No Listeners on the GPIB"
    ElseIf (ern = EADR) Then
        ers = "EADR:GPIB board not addressed correctly"
    ElseIf (ern = EARG) Then
        ers = "EARG:Invalid argument to function call"
    ElseIf (ern = ESAC) Then
        ers = "ESAC:GPIB board not System Controller as required"
    ElseIf (ern = EABO) Then
        ers = "EABO:I/O operation aborted(timeout)"
    ElseIf (ern = ENEB) Then
        ers = "ENEB:Nonexistent GPIB board"
    ElseIf (ern = EDMA) Then
        ers = "EDMA:DMA error"
    ElseIf (ern = EOIP) Then
        ers = "EOIP:I/O operation started before previous operation completed"
    ElseIf (ern = ECAP) Then
        ers = "ECAP:No capability for intended operation"
    ElseIf (ern = EFSO) Then
        ers = "EFSO:File system operation error"
    ElseIf (ern = EBUS) Then
        ers = "EBUS:GPIB bus error"
    ElseIf (ern = ESTB) Then
        ers = "ESTB:Serial poll status byte queue overflow"
    ElseIf (ern = ESRQ) Then
        ers = "ESRQ:SRQ remains asserted"
    ElseIf (ern = ETAB) Then
        ers = "ETAB:The return buffer is full"
    ElseIf (ern = ELCK) Then
        ers = "ELCK:Address or board is locked"
    Else
        ers = ""
    End If
Else
    ers = ""
End If

MsgBox ("Status No. " + Str(sts) + Chr(13) + wrn + "Error No. " + Str(ern) + Chr(13) + ers + Chr(13) + msg),
vbExclamation, "Error!"
Call ibclr(Dev)
Call ibonl(Dev, 0)
Dev = -1
End Sub

-----
Private Sub DisplayEthernetError(ByVal sts As Integer, ByVal msg As String)
    Dim wrn As String
    Dim ers As String
    Dim ern As Integer

    ern = TmGetLastError(eDev)

    If (ern = 2) Then
        ers = "Device not found"
    ElseIf (ern = 4) Then
        ers = "Connection to device failed"
    ElseIf (ern = 8) Then
        ers = "Device not connected"
    ElseIf (ern = 16) Then
        ers = "Device already connected"
    ElseIf (ern = 32) Then
        ers = "Incompatible PC"
    ElseIf (ern = 64) Then
        ers = "Illegal parameter"
    ElseIf (ern = 256) Then
        ers = "Send error"
    ElseIf (ern = 512) Then
        ers = "Receive error"
    ElseIf (ern = 1024) Then
        ers = "Received data not block data"
    ElseIf (ern = 4096) Then
        ers = "System error"
    ElseIf (ern = 8192) Then
        ers = "Illegal device ID"
    End If

    MsgBox ("Status No. " + Str(sts) + Chr(13) + wrn + "Error No. " + Str(ern) + Chr(13) + ers + Chr(13) + msg),
vbExclamation, "Error!"
    Call TmFinish(eDev)
    Dev = -1
End Sub

-----
Private Sub Command1_Click()
    Dim sts As Integer

    If (StartFlag = 1) Then

```

### 6.3 初期化・エラー・実行関数

---

```
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"
    List1.Clear
    Dummy = DoEvents()
    sts = SetParameter                                     'Run Sample1 Set/Get Measure Parameter (GP-IB)
    If (sts = 0) Then
        List1.AddItem Query(0)
        List1.AddItem Query(1)
    End If
    Text1.Text = "END"
    StartFlag = 0
End Sub
-----
Private Sub Command2_Click()
    Dim sts As Integer

    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"
    List1.Clear
    Dummy = DoEvents()
    sts = GetStatistics 1                                 'Run Sample2 GetStatistics 1 (GP-IB)
    If (sts = 0) Then
        List1.AddItem Query(0)
        List1.AddItem Query(1)
        List1.AddItem Query(2)
        List1.AddItem Query(3)
        List1.AddItem Query(4)
        List1.AddItem Query(5)
        List1.AddItem Query(6)
        List1.AddItem Query(7)
        List1.AddItem Query(8)
        List1.AddItem Query(9)
        List1.AddItem Query(10)
    End If
    Text1.Text = "END"
    StartFlag = 0
End Sub
-----
Private Sub Command3_Click()
    Dim sts As Integer

    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"
    List1.Clear
    Dummy = DoEvents()
    sts = GetStatistics2                                 'Run Sample3 Get Statistics 2 (GP-IB)
    If (sts = 0) Then
        List1.AddItem Query(0)
        List1.AddItem Query(1)
        List1.AddItem Query(2)
        List1.AddItem Query(3)
        List1.AddItem Query(4)
        List1.AddItem Query(5)
        List1.AddItem Query(6)
        List1.AddItem Query(7)
        List1.AddItem Query(8)
        List1.AddItem Query(9)
        List1.AddItem Query(10)
    End If
    Text1.Text = "END"
    StartFlag = 0
End Sub
-----
Private Sub Command4_Click()
    Dim sts As Integer
    Dim i As Integer

    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"
    List1.Clear
    Dummy = DoEvents()
    sts = GetBinaryData                                 'Run Sample4 Get Binary Data (GP-IB)
    If (sts = 0) Then
        For i = 0 To 99
            List1.AddItem Query(i)
        Next i
    End If
    Text1.Text = "END"
    StartFlag = 0
End Sub
```

```

Private Sub Command5_Click()
    Dim sts As Integer

    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"
    List1.Clear
    Dummy = DoEvents()
    sts = SetParameterEthernet           'Run Sample5 Set/Get Measure Parameter (Ethernet)
    If (sts = 0) Then
        List1.AddItem Query(0)
        List1.AddItem Query(1)
    End If
    Text1.Text = "END"
    StartFlag = 0
End Sub
-----
Private Sub Command6_Click()
    Dim sts As Integer

    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"
    List1.Clear
    Dummy = DoEvents()
    sts = GetStatisticsEthernet         'Run Sample6 GetStatistics (Ethernet)
    If (sts = 0) Then
        List1.AddItem Query(0)
        List1.AddItem Query(1)
        List1.AddItem Query(2)
        List1.AddItem Query(3)
        List1.AddItem Query(4)
        List1.AddItem Query(5)
        List1.AddItem Query(6)
        List1.AddItem Query(7)
        List1.AddItem Query(8)
        List1.AddItem Query(9)
        List1.AddItem Query(10)
    End If
    Text1.Text = "END"
    StartFlag = 0
End Sub
-----
Private Sub Form_Load()

    StartFlag = 0                       'Clear Start Flag
    Dev = -1                             'Clear device id
    addr = 1                             'GPIO Address = 1
    Command1.Caption = "Sample1: (GPIO)" + Chr(13) + "Set/Get Parameter"
    Command2.Caption = "Sample2: (GPIO)" + Chr(13) + "Get Statistics (PWIDth)"
    Command3.Caption = "Sample3: (GPIO)" + Chr(13) + "Get Statistics (PPERiod)"
    Command4.Caption = "Sample4: (GPIO)" + Chr(13) + "Get Binary Data"
    Command5.Caption = "Sample5: (Ethernet)" + Chr(13) + "Set/Get Parameter"
    Command6.Caption = "Sample6: (Ethernet)" + Chr(13) + "Get Statistics (PWIDth)"
    Text1.Text = ""
End Sub
-----

```

## 6.4 測定パラメータの設定/問い合わせ(GP-IB用)

```

Sample1(GPIB)
-----
Private Function SetParameter() As Integer
Dim msg As String           'Command buffer
Dim qry As String          'Query buffer
Dim sts As Integer

msg = Space$(100)
qry = Space$(100)

sts = InitGpib              'Initialize GPIB
If (sts <> 0) Then
  SetParameter = 1
  Exit Function
End If
msg = "*RST" + term        'Initialize Setting Parameter
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  SetParameter = 1
  Exit Function
End If

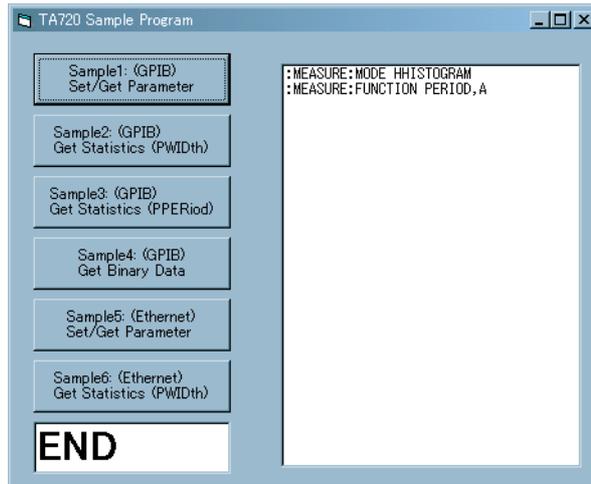
msg = ":COMMUNICATE:HEADER ON" + term  'Header = ON
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  SetParameter = 1
  Exit Function
End If
msg = ":COMM:VERBOSE ON" + term        'Verbose = ON
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  SetParameter = 1
  Exit Function
End If

msg = ":MEASURE:MODE HHISTOGRAM" + term  'Mode = HardHistogram
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  SetParameter = 1
  Exit Function
End If
msg = ":MEASURE:MODE?" + term           'Get Mode
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  SetParameter = 1
  Exit Function
End If
sts = ilrd(Dev, qry, Len(qry))          'Receive Query
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  SetParameter = 1
  Exit Function
End If
Query(0) = Left$(qry, ibcntl - 1)

msg = ":MEASURE:FUNCTION PERIOD,A" + term  'Function = Period, Ch = A
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  SetParameter = 1
  Exit Function
End If
msg = ":MEASURE:FUNCTION?" + term        'Get Function
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  SetParameter = 1
  Exit Function
End If
sts = ilrd(Dev, qry, Len(qry))          'Receive Query
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  SetParameter = 1
  Exit Function
End If
Query(1) = Left$(qry, ibcntl - 1)

Call ibonl(Dev, 0)
SetParameter = 0
End Function
-----

```



## 6.5 周期測定の測定統計値の問い合わせ(GP-IB用)

```

Sample2(GPIB) Get Statistics
-----
Private Function GetStatistics1() As Integer
Dim msg As String           'Command buffer
Dim qry As String           'Query buffer
Dim sts As Integer
Dim winlabel(10) As String  'Widow Label
Dim i As Integer
Dim temp As String

msg = Space$(100)
qry = Space$(100)

sts = InitGpib               'Initialize GPIB
If (sts <> 0) Then
  GetStatistics1 = 1
  Exit Function
End If
msg = "*RST" + term         'Initialize Setting Parameter
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics1 = 1
  Exit Function
End If

msg = ":MEAS:MODE TST" + term   'Mode = TimeStamp
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics1 = 1
  Exit Function
End If
msg = ":MEAS:FUNC PWID,A" + term 'Function = PulseWidth Ch = A
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics1 = 1
  Exit Function
End If
msg = ":SAMP:GATE:MODE EVEN;EVEN 100000" + term 'Gate = Event; EventSize 100000
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics1 = 1
  Exit Function
End If
msg = ":THIS:MODE MULT" + term  'WindowMode = MULTI WINDOW
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics1 = 1
  Exit Function
End If
msg = ":THIS:MULT:SIZE 10;TVAL 37ns" + term  'WINDOW SIZE 10; T=37ns
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics1 = 1
  Exit Function
End If
                             'Set Window Label
winlabel(0) = "3": winlabel(1) = "4": winlabel(2) = "5"
winlabel(3) = "6": winlabel(4) = "7": winlabel(5) = "8"
winlabel(6) = "9": winlabel(7) = "10": winlabel(8) = "11"
winlabel(9) = "14"
For i = 1 To 10
  msg = ":THIS:MULT:WIND" + LTrim(Str(i)) + ":LAB " + winlabel(i - 1) + term
  sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
  If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics1 = 1
    Exit Function
  End If
Next i
msg = ":THIS:MULT:UPD" + term    'UPDATE Window Scale
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics1 = 1
  Exit Function
End If
msg = "CALC:PAR:AVER ON;SDEV ON" + term  'Calc/Stat On
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics1 = 1
  Exit Function
End If

```

```

msg = "STAT:FILT1 RISE" + term           'Filter 1 Rise(Data Available)
sts = ilwrt(Dev, msg, Len(msg))         'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics1 = 1
Exit Function
End If
msg = "STAT:EESR?" + term               'Clear Extended Event Register
sts = ilwrt(Dev, msg, Len(msg))         'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics1 = 1
Exit Function
End If
sts = ilrd(Dev, qry, Len(qry))           'Receive Query
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics1 = 1
Exit Function
End If
msg = "SST" + term                      'Single Measure Start
sts = ilwrt(Dev, msg, Len(msg))         'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics1 = 1
Exit Function
End If
msg = "COMM:WAIT 1" + term              'Wait until data available
sts = ilwrt(Dev, msg, Len(msg))         'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics1 = 1
Exit Function
End If
msg = ":COMM:HEAD OFF" + term           'Header = OFF
sts = ilwrt(Dev, msg, Len(msg))         'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics1 = 1
Exit Function
End If

Query(0) = Space$(14) + "Average" + Space$(8) + "Sigma"
For i = 1 To 10
  temp = LTrim(Str(i))
  Query(i) = "WINDOW" + temp + Space$(2 - Len(temp))
  'Get Average value
  msg = ":CALC:WIND" + LTrim(Str(i)) + ":AVER?" + term
  sts = ilwrt(Dev, msg, Len(msg))       'Send Command
  If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics1 = 1
    Exit Function
  End If
  sts = ilrd(Dev, qry, Len(qry))         'Receive Query
  If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics1 = 1
    Exit Function
  End If
  temp = Left$(qry, ibcctl - 1)
  Query(i) = Query(i) + Space$(13 - Len(temp)) + temp
  'Get Standard Deviation value
  msg = ":CALC:WIND" + LTrim(Str(i)) + ":SDEV?" + term
  sts = ilwrt(Dev, msg, Len(msg))       'Send Command
  If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics1 = 1
    Exit Function
  End If
  sts = ilrd(Dev, qry, Len(qry))         'Receive Query
  If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics1 = 1
    Exit Function
  End If
  temp = Left$(qry, ibcctl - 1)
  Query(i) = Query(i) + Space$(13 - Len(temp)) + temp
Next i

msg = ":COMM:HEAD ON" + term            'Header = ON
sts = ilwrt(Dev, msg, Len(msg))         'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics1 = 1
Exit Function
End If

Call ibonl(Dev, 0)
GetStatistics1 = 0
End Function

```

## 6.5 周期測定 of 測定統計値の問い合わせ(GP-IB用)

The screenshot shows a software window titled "TA720 Sample Program". On the left side, there is a vertical stack of buttons for different samples:

- Sample1: (GPIB) Set/Get Parameter
- Sample2: (GPIB) Get Statistics (PWIDth) - This button is highlighted with a dashed border.
- Sample3: (GPIB) Get Statistics (PPERiod)
- Sample4: (GPIB) Get Binary Data
- Sample5: (Ethernet) Set/Get Parameter
- Sample6: (Ethernet) Get Statistics (PWIDth)
- END

On the right side, a table displays statistical data for ten windows:

	Average	Sigma
WINDOW1	1.1092125E-7	4.96325E-9
WINDOW2	1.5063700E-7	4.27625E-9
WINDOW3	1.8397875E-7	3.84675E-9
WINDOW4	2.1922675E-7	4.17275E-9
WINDOW5	2.567400E-7	4.0275E-9
WINDOW6	2.957025E-7	4.0575E-9
WINDOW7	3.326300E-7	3.7550E-9
WINDOW8	3.693775E-7	3.8375E-9
WINDOW9	4.062350E-7	4.0300E-9
WINDOW10	5.172775E-7	3.3000E-9

## 6.6 周期A&amp;周期B測定の測定統計値の問い合わせ(GP-IB用)

```

Sample3(GPIB) Get Statistics
-----
Private Function GetStatistics2() As Integer
Dim msg As String           'Command buffer
Dim qry As String          'Query buffer
Dim sts As Integer
Dim winlabel(10) As String 'Widow Label
Dim i As Integer
Dim temp As String

msg = Space$(100)
qry = Space$(100)

sts = InitGpib              'Initialize GPIB
If (sts <> 0) Then
  GetStatistics2 = 1
  Exit Function
End If
msg = "*RST" + term        'Initialize Setting Parameter
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics2 = 1
  Exit Function
End If

msg = ":MEASURE:MODE HHISTOGRAM" + term 'Mode = HardHistogram
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics2 = 1
  Exit Function
End If
msg = ":MEASURE:FUNCTION PPERIOD,AB" + term 'Function = Period & Period Ch = A,B
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics2 = 1
  Exit Function
End If

msg = ":SAMPLE:GATE:MODE EVENT;EVENTSIZE 10000" + term 'Gate = Event; EventSize 10000
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics2 = 1
  Exit Function
End If
msg = ":HHISTOGRAM:MODE MULTI" + term 'WindowMode = MULTI
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics2 = 1
  Exit Function
End If
msg = ":HHISTOGRAM:MULTI:SIZE 10" + term 'WINDOW SIZE 10
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics2 = 1
  Exit Function
End If
msg = ":HHISTOGRAM1:MULTI:TVALUE 37ns" + term 'Meas1: T=37ns
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics2 = 1
  Exit Function
End If
msg = ":HHISTOGRAM2:MULTI:TVALUE 37ns" + term 'Meas2: T=37ns
sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics2 = 1
  Exit Function
End If

winlabel(0) = "3": winlabel(1) = "4": winlabel(2) = "5"
winlabel(3) = "6": winlabel(4) = "7": winlabel(5) = "8"
winlabel(6) = "9": winlabel(7) = "10": winlabel(8) = "11"
winlabel(9) = "14"
For i = 1 To 10
  msg = ":HHISTOGRAM:MULTI:WINDOW" + LTrim(Str(i)) _
    + ":LABEL " + winlabel(i - 1) + term
  sts = ilwrt(Dev, msg, Len(msg))
                             'Send Command
  If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics2 = 1
  End If

```

## 6.6 周期A&周期B測定の測定統計値の問い合わせ(GP-IB用)

```

Exit Function
End If
Next i
msg = ".HHISTOGRAM:MULTI:UPDATE" + term           'UPDATE Window Scale
sts = ilwrt(Dev, msg, Len(msg))                   'Send Command
If (sts < 0) Then
Call DisplayGPIBError(sts, msg)
GetStatistics2 = 1
Exit Function
End If
msg = "CALCULATION:PARAMETER:AVERAGE ON" + term 'Calc/Stat On
sts = ilwrt(Dev, msg, Len(msg))                   'Send Command
If (sts < 0) Then
Call DisplayGPIBError(sts, msg)
GetStatistics2 = 1
Exit Function
End If

msg = "STATUS:FILTER1 RISE" + term                 'Filter1 Rise(Data Available)
sts = ilwrt(Dev, msg, Len(msg))                   'Send Command
If (sts < 0) Then
Call DisplayGPIBError(sts, msg)
GetStatistics2 = 1
Exit Function
End If
msg = "STATUS:EESR?" + term                       'Clear Extended Event Register
sts = ilwrt(Dev, msg, Len(msg))                   'Send Command
If (sts < 0) Then
Call DisplayGPIBError(sts, msg)
GetStatistics2 = 1
Exit Function
End If
sts = ilrd(Dev, qry, Len(qry))                     'Receive Query
If (sts < 0) Then
Call DisplayGPIBError(sts, msg)
GetStatistics2 = 1
Exit Function
End If
msg = "SSTART" + term                             'Single Measure Start
sts = ilwrt(Dev, msg, Len(msg))                   'Send Command
If (sts < 0) Then
Call DisplayGPIBError(sts, msg)
GetStatistics2 = 1
Exit Function
End If
msg = "COMMUNICATE:WAIT 1" + term                 'Wait until data available
sts = ilwrt(Dev, msg, Len(msg))                   'Send Command
If (sts < 0) Then
Call DisplayGPIBError(sts, msg)
GetStatistics2 = 1
Exit Function
End If
msg = ".COMMUNICATE:HEADER OFF" + term           'Header = OFF
sts = ilwrt(Dev, msg, Len(msg))                   'Send Command
If (sts < 0) Then
Call DisplayGPIBError(sts, msg)
GetStatistics2 = 1
Exit Function
End If

Query(0) = Space$(16) + "Meas1" + Space$(8) + "Meas2"
For i = 1 To 10
temp = LTrim(Str(i))
Query(i) = "WINDOW" + temp + Space$(2 - Len(temp))
'Get Average value
msg = ".CALCULATION:MEAS1:WINDOW" + LTrim(Str(i)) _
+ ".AVERAGE?" + term
sts = ilwrt(Dev, msg, Len(msg))                   'Send Command
If (sts < 0) Then
Call DisplayGPIBError(sts, msg)
GetStatistics2 = 1
Exit Function
End If
sts = ilrd(Dev, qry, Len(qry))                     'Receive Query
If (sts < 0) Then
Call DisplayGPIBError(sts, msg)
GetStatistics2 = 1
Exit Function
End If
temp = Left$(qry, ibcctl - 1)
Query(i) = Query(i) + Space$(13 - Len(temp)) + temp

'Get Standard Deviation value
msg = ".CALCULATION:MEAS2:WINDOW" + LTrim(Str(i)) _
+ ".AVERAGE?" + term
sts = ilwrt(Dev, msg, Len(msg))                   'Send Command
If (sts < 0) Then
Call DisplayGPIBError(sts, msg)
GetStatistics2 = 1
Exit Function
End If

```

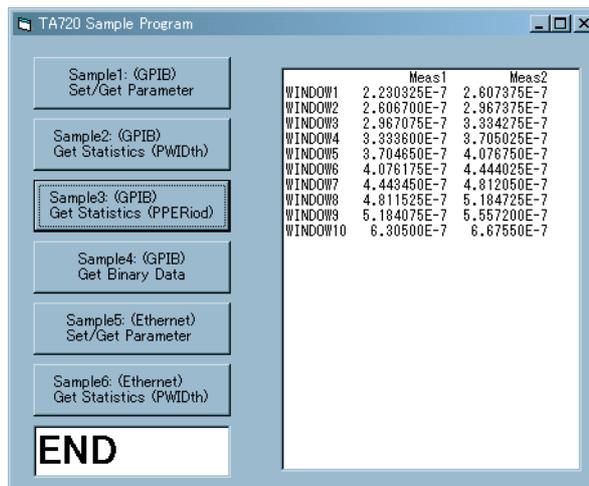
```

sts = ird(Dev, qry, Len(qry))           'Receive Query
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics2 = 1
  Exit Function
End If
temp = Left$(qry, ibcctl - 1)
Query(i) = Query(i) + Space$(13 - Len(temp)) + temp
Next i

msg = ":COMMUNICATE:HEADER ON" + term   'Header = ON
sts = ilwrt(Dev, msg, Len(msg))        'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetStatistics2 = 1
  Exit Function
End If

Call ibonl(Dev, 0)
GetStatistics2 = 0
End Function

```



## 6.7 AtoBタイムインターバル測定の測定データの問い合わせ(GP-IB用)

```

Sample4(GPIB)
-----
Private Function GetBinaryData() As Integer
Dim msg As String           'Command buffer
Dim qry As String           'Query buffer
Dim sts As Integer
Dim idata(1000) As Integer   'Data buffer(Binary)
Dim num As Integer
Dim data_h As Long
Dim data_l As Long
Dim data As Variant
Dim eos As Integer          'EOS
Dim i As Integer

msg = Space$(100)
qry = Space$(1000)

sts = InitGpib               'Initialize GPIB
If (sts <> 0) Then
  GetBinaryData = 1
  Exit Function
End If
msg = "*RST" + term          'Initialize Setting Parameter
sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If

msg = ":MEASURE:MODE TSTAMP" + term 'Mode = Time Stamp
sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If
msg = ":MEASURE:FUNCTION TI,AB" + term 'Function = Time Interval, Ch = A,B
sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If

msg = ":SAMPLE:GATE:MODE EVENT;EVENTSIZE 100" + term 'Gate = Event; EventSize 100
sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If

msg = "STATUS:FILTER1 RISE" + term 'Filter 1 Rise(Data Available)
sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If
msg = "STATUS:EESR?" + term 'Clear Extended Event Register
sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If
sts = iird(Dev, qry, Len(qry)) 'Receive Query
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If
msg = "SSTART" + term 'Single Measure Start
sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If
msg = "COMMUNICATE:WAIT 1" + term 'Wait until data available
sts = ilwrt(Dev, msg, Len(msg))
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If

```

```

msg = "MEMORY:DATASELECT MEASUREDATA" + term
sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If

msg = "MEMORY:FORMAT BINARY" + term
sts = ilwrt(Dev, msg, Len(msg)) 'Data Format = Binary
                                'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If

msg = "MEMORY:BYTEORDER LSBFIRST" + term
sts = ilwrt(Dev, msg, Len(msg)) 'Data Byte order = LSB First(for Little Endian)
                                'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If

msg = "MEMORY:START 1;END 100" + term
sts = ilwrt(Dev, msg, Len(msg)) 'START 1, END 100
                                'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If

eos = 0
sts = ileos(Dev, eos) 'Terminator = None(for Binary Data)
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If

msg = "MEMORY:SEND?" + term
sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If

sts = ilrd(Dev, qry, 10) 'Receive Header(#8******)
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If

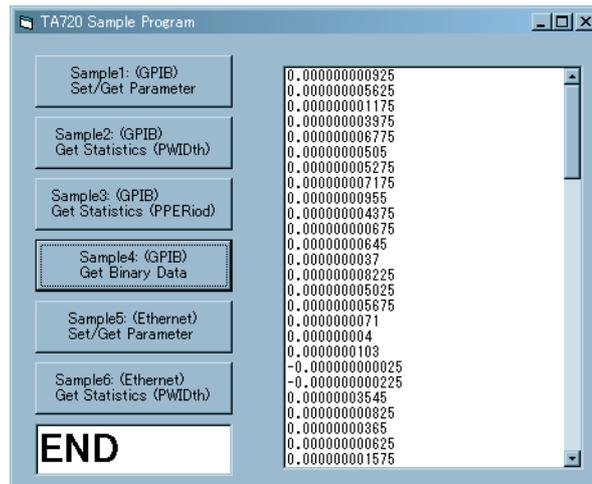
num = Val(Mid$(qry, 3, 8)) / 4
sts = ilrdi(Dev, idata(), 1000) 'Receive Binary Data + LF
For i = 0 To num - 1
  data_h = idata(i * 2 + 1)
  data_l = idata(i * 2)
  If data_h >= 0 And data_l >= 0 Then
    data = (data_h * 65536#) + data_l
  ElseIf data_h >= 0 And data_l < 0 Then
    data = (data_h * 65536#) + (data_l + 65536#)
  ElseIf data_h < 0 And data_l >= 0 Then
    data = (data_h + 65536#) * 65536# + data_l
    data = data - (65536# * 65536#)
  Else ' data_h < 0 And data_l < 0
    data = (data_h + 65536#) * 65536# + (data_l + 65536#)
    data = data - (65536# * 65536#)
  End If
  Query(i) = Format(data * 0.000000000025)
Next i

eos = &HCOA
sts = ileos(Dev, eos) 'Terminator = LF
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GetBinaryData = 1
  Exit Function
End If

Call ibonl(Dev, 0)
GetBinaryData = 0
End Function

```

## 6.7 AtoBタイムインターバル測定の測定データの問い合わせ(GP-IB用)



## 6.8 測定パラメータの設定/問い合わせ(Ethernet用)

```

Sample5(Ethernet)
-----
Private Function SetParameterEthernet() As Integer
Dim msg As String           'Command buffer
Dim qry As String           'Query buffer
Dim sts As Integer
Dim length As Long         'Receive query length

msg = Space$(100)
qry = Space$(100)

sts = InitEthernet          'Initialize Ethernet
If (sts <> 0) Then
  SetParameterEthernet = 1
  Exit Function
End If
msg = "*RST" + term        'Initialize Setting Parameter
sts = TmSend(eDev, msg)    'Send Command
If (sts <> 0) Then
  Call DisplayEthernetError(sts, msg)
  SetParameterEthernet = 1
  Exit Function
End If

msg = ":COMMUNICATE:HEADER ON" + term    'Header = ON
sts = TmSend(eDev, msg)                  'Send Command
If (sts <> 0) Then
  Call DisplayEthernetError(sts, msg)
  SetParameterEthernet = 1
  Exit Function
End If
msg = ":COMMUNICATE:VERBOSE ON" + term   'Verbose = ON
sts = TmSend(eDev, msg)                  'Send Command
If (sts <> 0) Then
  Call DisplayEthernetError(sts, msg)
  SetParameterEthernet = 1
  Exit Function
End If

msg = ":MEASURE:MODE ISI" + term          'Mode = ISI
sts = TmSend(eDev, msg)                  'Send Command
If (sts <> 0) Then
  Call DisplayEthernetError(sts, msg)
  SetParameterEthernet = 1
  Exit Function
End If
msg = ":MEASURE:MODE?" + term            'Get Mode
sts = TmSend(eDev, msg)                  'Send Command
If (sts <> 0) Then
  Call DisplayEthernetError(sts, msg)
  SetParameterEthernet = 1
  Exit Function
End If
sts = TmReceive(eDev, qry, 1000, length)  'Receive Query
If (sts <> 0) Then
  Call DisplayEthernetError(sts, msg)
  SetParameterEthernet = 1
  Exit Function
End If
Query(0) = Left$(qry, length - 1)

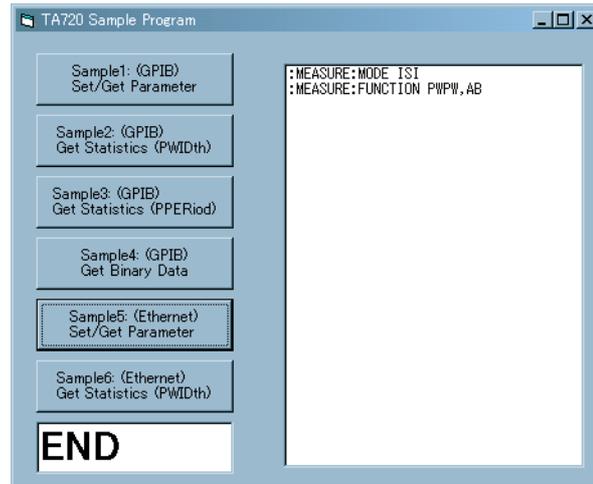
msg = ":MEASURE:FUNCTION PWPW,AB" + term  'Function = PW & PW Ch = A B
sts = TmSend(eDev, msg)                  'Send Command
If (sts <> 0) Then
  Call DisplayEthernetError(sts, msg)
  SetParameterEthernet = 1
  Exit Function
End If
msg = ":MEASURE:FUNCTION?" + term        'Get Function
sts = TmSend(eDev, msg)                  'Send Command
If (sts <> 0) Then
  Call DisplayEthernetError(sts, msg)
  SetParameterEthernet = 1
  Exit Function
End If
sts = TmReceive(eDev, qry, 1000, length)  'Receive Query
If (sts <> 0) Then
  Call DisplayEthernetError(sts, msg)
  SetParameterEthernet = 1
  Exit Function
End If
Query(1) = Left$(qry, length - 1)

Call TmFinish(eDev)
SetParameterEthernet = 0
End Function
-----

```

## 6.8 測定パラメータの設定/問い合わせ(Ethernet用)

---



## 6.9 パルス幅測定の測定統計値の問い合わせ(Ethernet用)

```

Sample6(Ethernet)
-----
Private Function GetStatisticsEthernet() As Integer
Dim msg As String           'Command buffer
Dim qry As String           'Query buffer
Dim sts As Integer
Dim length As Long          'Receive query length
Dim winlabel(10) As String  'Widow Label
Dim i As Integer
Dim temp As String

msg = Space$(100)
qry = Space$(100)

sts = InitEthernet          'Initialize Ethernet
If (sts <> 0) Then
    GetStatisticsEthernet = 1
    Exit Function
End If
msg = "*RST" + term         'Initialize Setting Parameter
sts = TmSend(eDev, msg)     'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If

msg = ":MEAS:MODE HHIS" + term 'Mode = HardHistogram
sts = TmSend(eDev, msg)     'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If
msg = ":MEAS:FUNC PWID,A" + term 'Function = PulseWidth Ch = A
sts = TmSend(eDev, msg)     'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If
msg = ":SAMP:GATE:MODE EVEN;EVEN 10000" + term 'Gate = Event; EventSize 10000
sts = TmSend(eDev, msg)     'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If
msg = ":HHIS:MODE MULT" + term 'WindowMode = MULTI
sts = TmSend(eDev, msg)     'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If
msg = ":HHIS:MULT:SIZE 10;TVAL 37ns" + term 'WINDOW SIZE 10; T=37ns
sts = TmSend(eDev, msg)     'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If

' Set Window Label
winlabel(0) = "3": winlabel(1) = "4": winlabel(2) = "5"
winlabel(3) = "6": winlabel(4) = "7": winlabel(5) = "8"
winlabel(6) = "9": winlabel(7) = "10": winlabel(8) = "11"
winlabel(9) = "14"
For i = 1 To 10
    msg = ":HHIS:MULT:WIND" + LTrim(Str(i)) + ":LAB " + winlabel(i - 1) + term
    sts = TmSend(eDev, msg) 'Send Command
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If
Next i
msg = ":HHIS:MULT:UPD" + term 'UPDATE Window Scale
sts = TmSend(eDev, msg)     'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If
msg = ":CALC:PAR:AVER ON;SDEV ON" + term 'Calc/Stat On
sts = TmSend(eDev, msg)     'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1

```

## 6.9 パルス幅測定の測定統計値の問い合わせ(Ethernet用)

```

Exit Function
End If

msg = "STAT:FILT 1 RISE" + term
sts = TmSend(eDev, msg) 'Filter 1 Rise(Data Available)
                          'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If

msg = "STAT:EESR?" + term
sts = TmSend(eDev, msg) 'Clear Extended Event Register
                          'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If

sts = TmReceive(eDev, qry, 1000, length) 'Receive Query
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If

msg = "SST" + term
sts = TmSend(eDev, msg) 'Single Measure Start
                          'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If

msg = "COMM:WAIT 1" + term
sts = TmSend(eDev, msg) 'Wait until data available
                          'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If

msg = "COMM:HEAD OFF" + term
sts = TmSend(eDev, msg) 'Header = OFF
                          'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If

Query(0) = Space$(14) + "Average" + Space$(8) + "Sigma"
For i = 1 To 10
    temp = LTrim(Str(i))
    Query(i) = "WINDOW" + temp + Space$(2 - Len(temp))
    'Get Average value

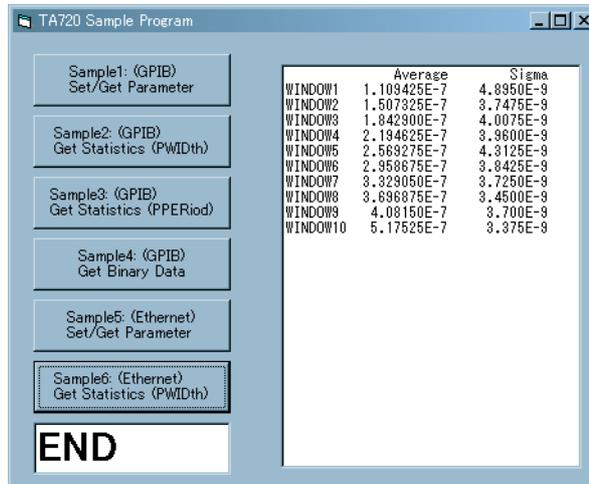
    msg = "CALC:WIND" + LTrim(Str(i)) + ":AVER?" + term
    sts = TmSend(eDev, msg) 'Send Command
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If
    sts = TmReceive(eDev, qry, 1000, length) 'Receive Query
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If
    temp = Left$(qry, length - 1)
    Query(i) = Query(i) + Space$(13 - Len(temp)) + temp
    'Get Standard Deviation value

    msg = "CALC:WIND" + LTrim(Str(i)) + ":SDEV?" + term
    sts = TmSend(eDev, msg) 'Send Command
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If
    sts = TmReceive(eDev, qry, 1000, length) 'Receive Query
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If
    temp = Left$(qry, length - 1)
    Query(i) = Query(i) + Space$(13 - Len(temp)) + temp
Next i

msg = "COMM:HEAD ON" + term
sts = TmSend(eDev, msg) 'Header = ON
                          'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If

```

```
Call TmFinish(eDev)  
GetStatisticsEthernet = 0  
End Function
```



The screenshot shows a window titled "TA720 Sample Program" with a list of statistics for ten windows. The statistics are presented in a table with columns for window name, average, and sigma. The interface also includes several control buttons for different samples and an "END" button.

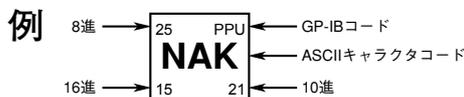
	Average	Sigma
WINDOW1	1.109425E-7	4.8950E-9
WINDOW2	1.507325E-7	3.7475E-9
WINDOW3	1.842900E-7	4.0075E-9
WINDOW4	2.194825E-7	3.9600E-9
WINDOW5	2.568275E-7	4.3125E-9
WINDOW6	2.958675E-7	3.8425E-9
WINDOW7	3.328050E-7	3.7250E-9
WINDOW8	3.698875E-7	3.4500E-9
WINDOW9	4.08160E-7	3.700E-9
WINDOW10	5.17525E-7	3.375E-9

# 付録

## 付録1 ASCIIキャラクターコード

ここでは、ASCII のキャラクターコード表を紹介しています。

	0	1	2	3	4	5	6	7
0	0 NUL	20 DEL	40 SP	60 0	80 @	100 P	120 '	140 p
1	1 SOH	21 DC1	41 !	61 1	81 A	101 Q	121 a	141 q
2	2 STX	22 DC2	42 "	62 2	82 B	102 R	122 b	142 r
3	3 ETX	23 DC3	43 #	63 3	83 C	103 S	123 c	143 s
4	4 EOT	24 DC4	44 \$	64 4	84 D	104 T	124 d	144 t
5	5 ENQ	25 NAK	45 %	65 5	85 E	105 U	125 e	145 u
6	6 ACK	26 SYN	46 &	66 6	86 F	106 V	126 f	146 v
7	7 BEL	27 ETB	47 ,	67 7	87 G	107 W	127 g	147 w
8	10 BS	30 CAN	50 (	70 8	90 H	110 X	130 h	150 x
9	11 HT	31 EM	51 )	71 9	91 I	111 Y	131 i	151 y
A	12 LF	32 SUB	52 *	72 :	92 J	112 Z	132 j	152 z
B	13 VT	33 ESC	53 +	73 ;	93 K	113 [	133 k	153 {
C	14 FF	34 FS	54 ,	74 <	94 L	114 \	134 l	154 
D	15 CR	35 GS	55 -	75 =	95 M	115 ]	135 m	155 }
E	16 SO	36 RS	56 .	76 >	96 N	116 ^	136 n	156 ~
F	17 SI	37 US	57 /	77 ?	97 O	117 _	137 o	157 DEL (RUBOUT)
	アドレス コマンド	ユニバーサル コマンド	リスナ アドレス		トーカ アドレス		2次 コマンド	



## 付録2 エラーメッセージ

ここでは、通信に関するエラーメッセージについて説明しています。

- サービスが必要なときは、裏表紙に記載の横河エンジニアリングサービス(株)まで修理をお申しつけください。
- 通信以外のエラーメッセージについては、本体ユーザーズマニュアル(IM704510-01)を参照してください。

## 通信文法エラー(100~199)

## Error in communication command

コード	メッセージ	エラー内容	ページ
102	Syntax error	以下のコード以外で構文や文法が間違っています。	3章, 4章
103	Invalid separator	<DATA SEPARATOR>がありません。データとデータは「,」(カンマ)で区切ってください。	3-1
104	Data type error	<DATA>の種類が間違っています。3-5~3-6ページを参照して、正しいデータ形式で記述してください。	3-5~3-6
108	Parameter not allowed	<DATA>が多すぎます。データの数を確認してください。	3-5, 4章
109	Missing parameter	必要な<DATA>がありません。必要なデータを記述してください。	3-5, 4章
111	Header separator error	<HEADER SEPARATOR>がありません。ヘッダとデータはスペースで区切ってください。	3-2
112	Program mnemonic too long	<mnemonic>が長すぎます。二モニック(アルファベットと数字からなる文字列)を確認してください。	4章
113	Undefined header	そのような命令はありません。ヘッダを確認してください。	4章
114	Header suffix out of range	<HEADER>の数値が間違っています。ヘッダを確認してください。	4章
120	Numeric data error	数値の仮数部分がありません。<NRf>形式のときは指数の前に仮数が必要です。	3-5
123	Exponent too large	指数が大きすぎます。<NR3>形式のときの「E」のあとの指数を小さくしてください。	3-5, 4章
124	Too many digits	有効桁数が多すぎます。数字は255桁以内にしてください。	3-5, 4章
128	Numeric data not allowed	数値データは使えません。<NRf>形式以外のデータ形式で記述してください。4章	3-5, 4章
131	Invalid suffix	単位が間違っています。<電圧>, <時間>の単位を確認してください。	3-5
134	Suffix too long	単位のつづりが長すぎます。<電圧>, <時間>の単位を確認してください。	3-5
138	Suffix not allowed	単位は使えません。<電圧>, <時間>以外では単位は使えません。	3-5
141	Invalid character data	そのような選択肢はありません。{... ... ...}の中にある文字列を記述してください。	4章
144	Character data too long	<CHARACTER DATA>のつづりが長すぎます。{... ... ...}の文字列のつづりを確認してください。	4章
148	Character data not allowed	<CHARACTER DATA>は使えません。{... ... ...}以外のデータ形式で記述してください。	4章
150	String data error	<STRING DATA>の右の区切りがありません。<文字列>の場合は「"」または「'」で囲ってください。	3-6

コード	メッセージ	エラー内容	ページ
151	Invalid string data	<STRING DATA>の内容が不適当です。 <文字列>が長すぎるか、使用不可能な文字があります。	4章
158	String data not allowed	<STRING DATA>は使えません。 <文字列>以外のデータ形式で記述してください。	4章
161	Invalid block data	<BLOCK DATA>のデータ長が合っていません。 <ブロックデータ>は使用できません。	3-6, 4章
168	Block data not allowed	<BLOCK DATA>は使えません。 <ブロックデータ>は使用できません。	3-6, 4章
171	Invalid expression	<EXPRESSION DATA>の中に許されない文字があります。 演算式は使用できません。	4章
178	Expression data not allowed	<EXPRESSION DATA>は使えません。 演算式は使用できません。	4章
181	Invalid outside macro definition	ブレースホルダがマクロの外にあります。 IEEE488.2のマクロ機能には対応していません。	—

### 通信実行エラー(200~299)

#### Error in communication execution

コード	メッセージ	エラー内容	ページ
221	Setting conflict	設定内容に矛盾があります。 関連のある設定値を確認してください。	4章
222	Data out of range	データの値が範囲外です。 設定範囲を確認してください。	4章
223	Too much data	データのバイト長が長すぎます。 データのバイト長を確認してください。	4章
224	Illegal parameter value	データの値が不適当です。 設定範囲を確認してください。	4章
241	Hardware missing	ハードウェアが実装されていません。 オプションの有無を確認してください。	—
260	Expression error	<EXPRESSION DATA>が間違っています。 演算式は使用できません。	—
270	Macro error	マクロのネストが深すぎます。 IEEE488.2のマクロ機能には対応していません。	—
272	Macro execution error	マクロでは使用できません。 IEEE488.2のマクロ機能には対応していません。	—
273	Illegal macro label	マクロラベルが不適当です。 IEEE488.2のマクロ機能には対応していません。	—
275	Macro definition too long	マクロが長すぎます。 IEEE488.2のマクロ機能には対応していません。	—
276	Macro recursion error	マクロが再帰呼び出しされました。 IEEE488.2のマクロ機能には対応していません。	—
277	Macro redefinition not allowed	マクロの二重定義はできません。 IEEE488.2のマクロ機能には対応していません。	—
278	Macro header not found	そのようなマクロは定義されていません。 IEEE488.2のマクロ機能には対応していません。	—

**通信クエリエラー(400~499)**  
**Error in communication Query**

コード	メッセージ	エラー内容	ページ
410	Query INTERRUPTED	応答の送信が中断されました。 送受信の順序を確認してください。	3-2
420	Query UNTERMINATED	送信できる応答がありません。 送受信の順序を確認してください。	3-2
430	Query DEADLOCKED	送受信がデッドロックしました。 送信を中止します。 プログラムメッセージは<PMT>も含めて 1024バイト以下にしてください。	3-2
440	Query UNTERMINATED after indefinite response	応答を要求する順番が間違っています。 *IDN?, *OPT?の後ろにはクエリを記述 しないでください。	—

**実行エラー(600~899)**  
**Error in Execution(本体のマニュアルを参照してください)**

**システムエラー(通信) (912~915)**  
**Error in System Operation**

コード	メッセージ	エラー内容	ページ
912	Fatal error in Communication-driver	通信ドライバーエラー サービスが必要です。	—
914	A communication time-out has occurred.	通信タイムアウトエラー タイムアウト時間内にデータを受け取る ようにしてください。または、通信回線 に異常がある場合が考えられます。	—

**その他(350)**

コード	メッセージ	対処方法	ページ
350	Queue overflow	エラーキューを読み出してください。	5-5

**Note**

コード「350」は、エラーキューがあふれたときに発生します。STATus:ERRor?クエリのと  
きだけに出力されるエラーで、画面には表示されません。

### 付録3 IEEE 488.3-1992について

本機器の通信インタフェースは、IEEE 488.3-1992規格に準じています。この規格では、以下の23の項目について「ドキュメントに記載しなければならない」としてあります。ここでは、これらについて説明しています。

**(1)IEEE 488.1インタフェース機能のうち、サポートしているサブセット**

1-3ページの「GP-IBインタフェースの仕様」を参照してください。

**(2)アドレスが0~30以外に設定されたときのデバイスの動作**

アドレスを0~30以外に設定することはできません。

**(3)ユーザーがアドレス変更を初期化したときの動作**

アドレスの変更は、UTILITYキーの[GP-IB]メニューでアドレスを選択した時点で認識されます。選択したアドレスは、次に変更するまで有効です。

**(4)電源ON時のデバイスのセッティング。電源ON時に使用可能なコマンド**

基本的には、以前の設定(その前に電源をOFFにしたときの設定)になります。

電源ON時に設定を限定するコマンドはありません。

**(5)メッセージ交換のオプション**

**(a)入力バッファのサイズ**

1024バイトです。

**(b)複数の返送データを返すクエリ**

4章の各コマンドの例を参照してください。

**(c)構文解析時に応答データを作成するクエリ**

すべてのクエリは、構文を解析すると返送データを作成します。

**(d)受信時に応答データを作成するクエリ**

コントローラが受信する時点で応答データを生成するクエリはありません。

**(e)制限し合うパラメータを有するコマンド**

4章の各コマンドの例を参照してください。

**(6)コマンドを構成する機能エレメントおよび複合ヘッダのエレメントに含まれるもの**

3章および4章を参照してください。

**(7)ブロックデータの転送に影響するバッファのサイズ**

ブロックデータのバッファサイズは、64Kバイトです。

**(8)演算式で使えるプログラムデータのエレメントの一覧と、そのネストの制限**

演算式は使えません。

**(9)各問い合わせに対する応答の構文**

4章の各コマンドの例を参照してください。

**(10)応答の文法に従わないデバイス間の通信について**

ありません。

**(11)返送データのブロックデータのサイズ**

0~4096000バイト

**(12)サポートしている共通コマンドの一覧**

「4.23 共通コマンドグループ」を参照してください。

**(13)キャリブレーション正常終了時のデバイスの状態**

設定はキャリブレーション実行前の状態になり、測定は終了状態となり前測定データは無効になります。

**(14)\*DDTのトリガマクロの定義で使用できるブロックデータの最大長**

サポートしていません。

**(15)マクロ定義のマクロラベルの最大長、マクロ定義で使用できるブロックデータの最大長、マクロ定義で再帰を使ったときの処理**

マクロ機能は対応していません。

**(16)\*IDN?に対する返送**

「4.23 共通コマンドグループ」を参照してください。

**(17)\*PUD, \*PUD?のプロテクトユーザーデータの保存エリアのサイズ**

\*PUD, \*PUD?の共通コマンドはサポートしていません。

**(18)\*RDT, \*RDT?のリソース名の長さ**

\*RDT, \*RDT?はサポートしていません。

**(19)\*RST, \*LRN?, \*RCL, \*SAVによる状態の変化**

\*RST

「4.23 共通コマンドグループ」を参照してください。

\*LRN?, \*RCL, \*SAV

これらの共通コマンドはサポートしていません。

**(20)\*TST?によるセルフテストの実行範囲**

UTILITYキーのSelftestメニューのBoardテストと同じ内容を実行します。

**(21)拡張されたリターンステータスの構造**

4章を参照してください。

**(22)各コマンドの処理がオーバーラップするか、シーケンシャルに行われるか**

「3.5 コントローラとの同期」および4章を参照してください。

**(23)各コマンドの実行内容**

4章の各コマンドの機能と本体ユーザズマニュアルを参照してください。

## 索引

## 記号

<Nrf> .....	3-5
<Register> .....	3-6
<時間> .....	3-5
<周波数> .....	3-5
<乗数> .....	3-5
<単位> .....	3-5
<電圧> .....	3-5
<パーセント> .....	3-5
<文字データ> .....	3-6

## A

Account .....	2-4
---------------	-----

## B

Boolean .....	3-6
BOTH Graph .....	4-27

## E

Estimated T .....	4-11
-------------------	------

## G

GP-IB インタフェースの仕様 .....	1-3
GP-IB ケーブル .....	1-2
GP-IB コネクタ .....	1-1
Graph Parmeter .....	4-27

## I

Infinite .....	2-5
ISI .....	4-45
ISI のトリガモード .....	4-49

## L

LCD .....	4-66
LOCAL キー .....	1-1

## M

Measured T .....	4-11
------------------	------

## O

OVERlap .....	4-28
---------------	------

## P

Password .....	2-4
----------------	-----

## S

Stat Item .....	4-28
Sync 機能の ON/OFF .....	4-48

## T

Time Out .....	2-4
----------------	-----

## U

User Name .....	2-4
-----------------	-----

## X

X Center, スケール値 .....	4-41
X Span .....	4-41

## ア

アーミング .....	4-51, 4-59
アーミングスロープ .....	4-60
アーミングソース .....	4-60
アーミングディレイ .....	4-59, 4-60
アーミングディレイモード .....	4-60
アーミングレベル .....	4-51
明るさ .....	4-66
空き容量 .....	4-31
圧縮の ON/OFF .....	4-34
アドレス .....	1-4

## イ

位相調整 .....	4-51
イベント数 .....	4-61
インタフェースメッセージ .....	1-5
インヒビット .....	4-52, 4-62
インヒビットレベル .....	4-52

## ウ

ウインドウ 1 の ON/OFF .....	4-37, 4-45
ウインドウ数 .....	4-38, 4-47
ウインドウのオフセット値 .....	4-38
ウインドウの水平軸(X 軸) .....	4-47
ウインドウモード .....	4-37, 4-46
ウインドウラベル .....	4-48

## エ

エラーキュー .....	4-63, 5-5
演算範囲 .....	4-10

## 索引

### オ

オートウインドウ .....	4-36, 4-45
オートウインドウでの定数 T .....	4-45
オートウインドウでの変調方式 .....	4-45
オートスケール .....	4-8
オートネーミング .....	4-31
オーバラップ .....	4-28
応答メッセージ .....	3-2

### カ

外部ゲート .....	4-51
拡張イベント .....	4-25
拡張イベントイネーブルレジスタ .....	4-63
拡張イベントレジスタ .....	4-63, 5-4
画面イメージデータ .....	4-33
画面イメージの出力先 .....	4-34
画面イメージの保存ファイル名 .....	4-34
画面イメージ保存時の色 / 階調 .....	4-34
画面イメージ保存時のデータ形式 .....	4-34

### キ

基準クロック .....	4-62
休止時間 .....	4-61
休止モード .....	4-61
極性 .....	4-54, 4-62

### ク

グラフの設定 .....	4-27
クリック音 .....	4-66
グループ .....	3-3

### ケ

ゲート .....	4-61
ゲート時間 .....	4-62
ゲートレベル .....	4-51

### コ

コメントの設定 / 問い合わせ .....	4-32, 4-34
-----------------------	------------

### サ

サンプリングインターバル .....	4-62
サンプリングモード .....	4-53, 4-54

### シ

時刻 .....	4-66
周波数形式による定数 T の値 .....	4-46
出力キュー .....	5-5
状態レジスタ .....	4-63, 5-4
省略形 .....	3-4
シングルウインドウ .....	4-40
シングル測定 .....	4-62

### ス

スケール値 .....	4-42
スケールの上限 .....	4-42
ステータスバイト .....	5-2
ステータスレポート .....	5-1
スペース .....	4-46
スロープ .....	4-54

### セ

設定情報のストア .....	4-64
設定情報のリコール .....	4-57
設定データ .....	4-31
遷移フィルタ .....	4-63, 5-4

### ソ

測定条件 .....	4-53
測定スタート .....	4-62
測定ストップ .....	4-64
測定データの送信 .....	4-56
測定点の表示方法 .....	4-27
測定ファンクション .....	4-53

### タ

タイムスタンプモード .....	4-36
タイムバリエーション .....	4-67
タイムバリエーションの統計演算 .....	4-11, 4-23

### ツ

通信に関する設定 .....	4-24
通信のステータス .....	4-63

### テ

データ .....	3-5
データ数 .....	4-56
データの保存 .....	4-31
データの呼び出し .....	4-31
定数 T .....	4-15, 4-36, 4-37, 4-38, 4-40
定数 T の値 .....	4-42, 4-47
定数 T へのオフセット値 .....	4-38, 4-47
ディレクトリの作成 .....	4-31
デッドロック .....	3-2

### ト

統計対象 .....	4-28
統計値演算項目 .....	4-10
統計値表示の ON/OFF .....	4-29
ドット .....	4-27
ドットコネクタ .....	4-27
ドライブ .....	4-31, 4-34
トリガ条件 .....	4-49
トリガモード .....	4-51
トリガレベル .....	4-51

**ニ**

入力インピーダンス .....	4-51
入力カップリング .....	4-51
入力条件 .....	4-50

**ハ**

ハードウェアヒストグラム .....	4-36
バイナリ測定データ .....	4-31
バイナリデータ .....	4-55
バックライトの ON/OFF .....	4-66
パノラマ表示 .....	4-28

**ヒ**

ピープ音 .....	4-65
ヒストグラム .....	4-36
ヒストグラムの統計演算 .....	4-16, 4-22
日付 .....	4-66
表示 .....	4-27
表示形式 .....	4-28
表示サイズ .....	4-27

**フ**

ファイル .....	4-30
ファイルの削除 .....	4-31
ファイル名の自動生成 .....	4-31
フォーマット .....	4-31
符号間干渉解析モード .....	4-45
プリントアウト .....	4-33, 4-34
プログラムメッセージ .....	3-1
ブロック .....	4-27, 4-55
ブロックサンプリング .....	4-60
ブロックサンプリングの休止 .....	4-60
ブロックサンプリングの休止時間 .....	4-61
ブロック数 .....	4-55, 4-61
ブロックデータ .....	3-6

**ヘ**

ヘッダ .....	3-3, 3-4, 4-24
変調方式 .....	4-36, 4-45

**ホ**

保存 .....	4-31
ポラリティ .....	4-54

**マ**

マーカー .....	4-37, 4-39, 4-40, 4-41, 4-68
マーク .....	4-46
マニュアルスケリング .....	4-67
マルチウインドウ .....	4-37, 4-46
マルチラインメッセージ .....	1-5

**メ**

メッセージ .....	3-1
目盛り .....	4-27

**モ**

文字列データ .....	3-6
--------------	-----

**ユ**

ユニラインメッセージ .....	1-5
------------------	-----

**ヨ**

呼び出し .....	4-31
------------	------

**リ**

リモート / ローカル切り替え .....	1-3
-----------------------	-----

**コマンドリスト**

{HHIStogram<x>   THIStogram<x>}グループ	
:{HHIStogram<x>   THIStogram<x>}? .....	4-36
:AUTO:MODulation .....	4-36
:AUTO:TTYPe .....	4-36
:AUTO:WINDow1:STATe .....	4-37
:AUTO? .....	4-36
:MARKer:LOW .....	4-37
:MARKer? .....	4-37
:MARKer[:STATe] .....	4-37
:MULTi:FREQuency .....	4-37
:MULTi:OFFSet .....	4-38
:MULTi:SIZE .....	4-38
:MULTi:TVALue .....	4-38
:MULTi:UPDate .....	4-38
:MULTi:WINDow<x>:HORizontal:CENTer .....	4-39
:MULTi:WINDow<x>:HORizontal:SPAN .....	4-39
:MULTi:WINDow<x>:HORizontal? .....	4-38
:MULTi:WINDow<x>:LABel .....	4-39
:MULTi:WINDow<x>:MARKer:{LEFT   RIGHT} .....	4-40
:MULTi:WINDow<x>:MARKer:{LVALue   RVALue}? .....	4-40
:MULTi:WINDow<x>:MARKer? .....	4-39
:MULTi:WINDow<x>? .....	4-38
:MULTi? .....	4-37
:SINGle:FREQuency .....	4-40
:SINGle:HORizontal:CENTer .....	4-41
:SINGle:HORizontal:SPAN .....	4-41
:SINGle:HORizontal? .....	4-40
:SINGle:MARKer:{LEFT   RIGHT} .....	4-41
:SINGle:MARKer:{LVALue   RVALue}? .....	4-41
:SINGle:MARKer? .....	4-41
:SINGle:TVALue .....	4-42
:SINGle? .....	4-40
:VERTical:AXIS .....	4-42
:VERTical:HIGH .....	4-42
:VERTical? .....	4-42
[:MODE] .....	4-37

## 索引

AScale グループ		DISPlay グループ	
:AScale .....	4-8	:BGRaph .....	4-27
CALCulation グループ		:BLOCK .....	4-27
:AREA .....	4-10	:DISPlay? .....	4-27
:CALCulation .....	4-10	:DOTConnect .....	4-27
:PARAmeter:AVERage .....	4-22	:DOTType .....	4-27
:PARAmeter:CLEar .....	4-22	:GRAPhsize .....	4-27
:PARAmeter:DEViation .....	4-22	:GRID .....	4-27
:PARAmeter:DEVT .....	4-22	:ITEM .....	4-28
:PARAmeter:FLUTter .....	4-22	:OVERlap .....	4-28
:PARAmeter:JITter .....	4-22	:PANorama[:STATe]? .....	4-28
:PARAmeter:MAXimum .....	4-22	:SGRaph .....	4-28
:PARAmeter:MEDian .....	4-22	:SITem .....	4-28
:PARAmeter:MINimum .....	4-22	:SSTYLE .....	4-28
:PARAmeter:MODE .....	4-22	:STATistic .....	4-29
:PARAmeter:PTOPeak .....	4-23	:WINDow .....	4-29
:PARAmeter:SDEViation .....	4-23	FILE グループ	
:PARAmeter:TAVerage .....	4-23	:CDIRectory .....	4-30
:PARAmeter:TFLutter .....	4-23	:DELete:{BINary BMP POSTscript SETup	
:PARAmeter:TJITter .....	4-23	STATistic TIFF TEXT} .....	4-31
:PARAmeter:TMAXimum .....	4-23	:DRIVE .....	4-31
:PARAmeter:TMINimum .....	4-23	:FILE? .....	4-30
:PARAmeter:TPTOpeak .....	4-23	:FORMat .....	4-31
:PARAmeter:TRF .....	4-23	:FREE? .....	4-31
:PARAmeter:TSDeviation .....	4-23	:LOAD:{BINary SETup} .....	4-31
:PARAmeter? .....	4-22	:MDIRectory .....	4-31
:POLarity .....	4-24	:PATH? .....	4-31
[:MEAS<x>]:AUTot? .....	4-11	:SAVE:{BINary SETup STATistic TEXT} .....	4-32
[:MEAS<x>]:CONSTt .....	4-15	:SAVE:ANAMing .....	4-31
[:MEAS<x>][:{BLOCK<x> BALL}]:TAVerage? .....	4-11	:SAVE:COMMeNt .....	4-32
[:MEAS<x>][:{BLOCK<x> BALL}]:TFLutter? .....	4-11	:SAVE:SITem .....	4-32
[:MEAS<x>][:{BLOCK<x> BALL}]:TJITter? .....	4-12	:SAVE? .....	4-31
[:MEAS<x>][:{BLOCK<x> BALL}]:TMAXimum? .....	4-12	HCOPY グループ	
[:MEAS<x>][:{BLOCK<x> BALL}]:TMINimum? .....	4-13	:ABORT .....	4-33
[:MEAS<x>][:{BLOCK<x> BALL}]:TPTOpeak? .....	4-13	:ANAMing .....	4-33
[:MEAS<x>][:{BLOCK<x> BALL}]:TRF? .....	4-14	:CDIRectory .....	4-34
[:MEAS<x>][:{BLOCK<x> BALL}]:TSDeviation? .....	4-14	:COMMeNt .....	4-34
[:MEAS<x>][:{BLOCK<x> BALL}]:TSNumber? .....	4-15	:COMPreSSion .....	4-34
[:MEAS<x>][:{WINDow<x> SUMMation}]:JITter? .....	4-18	:DEVice .....	4-34
[:MEAS<x>][:{WINDow<x> SUMMation}]:PTOPeak? .....	4-20	:DRIVE .....	4-34
[:MEAS<x>][:{WINDow<x> SUMMation}]:SDEViation? .....	4-21	:FILename .....	4-34
[:MEAS<x>][:{WINDow<x> SUMMation}]:SNUMBER? .....	4-21	:FORMat .....	4-34
[:MEAS<x>][:WINDow<x>]:AVERage? .....	4-16	:HCOPY? .....	4-33
[:MEAS<x>][:WINDow<x>]:DEViation? .....	4-16	:TONE .....	4-34
[:MEAS<x>][:WINDow<x>]:DEVT? .....	4-17	[:EXECute] .....	4-34
[:MEAS<x>][:WINDow<x>]:FLUTter? .....	4-17	IHIStogram グループ	
[:MEAS<x>][:WINDow<x>]:MAXimum? .....	4-18	[:{MARK<x> SPACE<x>} .....	4-46
[:MEAS<x>][:WINDow<x>]:MEDian? .....	4-19	:AUTO:MODulation .....	4-45
[:MEAS<x>][:WINDow<x>]:MINimum? .....	4-19	:AUTO:TTYPe .....	4-45
[:MEAS<x>][:WINDow<x>]:MODE? .....	4-20	:AUTO:WINDow1:STATe .....	4-45
COMMUnicate グループ		:AUTO? .....	4-45
:COMMUnicate? .....	4-24	:IHIStogram? .....	4-45
:HEADer .....	4-24	:MULTi:FREQuency .....	4-46
:VERBoSe .....	4-25	:MULTi:OFFSet .....	4-47
:WAIT .....	4-25	:MULTi:SIZE .....	4-47
:WAIT? .....	4-25	:MULTi:TVALue .....	4-47
		:MULTi:UPDate .....	4-47

:MULTi:WINDow<x>:HORizontal:CENTer .....	4-47	:BLOCK:REST:TIME .....	4-61
:MULTi:WINDow<x>:HORizontal:SPAN .....	4-47	:BLOCK:REST? .....	4-60
:MULTi:WINDow<x>:HORizontal? .....	4-47	:BLOCK:REST[:MODE] .....	4-61
:MULTi:WINDow<x>:LABel .....	4-48	:BLOCK:SIZE .....	4-61
:MULTi:WINDow<x>:MARKer:{LEFT   RIGHT} .....	4-48	:BLOCK? .....	4-60
:MULTi:WINDow<x>:MARKer? .....	4-48	:BLOCK[:STATe] .....	4-61
:MULTi:WINDow<x>? .....	4-47	:GATE:EVENTsize .....	4-61
:MULTi? .....	4-46	:GATE:POLarity .....	4-62
:POLarity .....	4-48	:GATE:TIME .....	4-62
:SYNC:STATe .....	4-48	:GATE? .....	4-61
:TARGet .....	4-48	:GATE[:MODE] .....	4-61
:TMODE .....	4-49	:INHibit .....	4-62
:TRIGger .....	4-49	:INTerval .....	4-62
[:MODE] .....	4-46	:RCLock .....	4-62
INPut グループ		:SAMPle? .....	4-59
:{ACHannel   BCHannel}:COUPling .....	4-51	SSStart グループ	
:{ACHannel   BCHannel}:IMPedance .....	4-51	:SSStart .....	4-62
:{ACHannel   BCHannel}:TRIGger:LEVel .....	4-51	STARt グループ	
:{ACHannel   BCHannel}:TRIGger:MODE .....	4-51	:STARt .....	4-62
:{ACHannel   BCHannel}:TRIGger? .....	4-51	STATus グループ	
:{ACHannel   BCHannel}? .....	4-50	:CONDition? .....	4-63
:AGATe:LEVel .....	4-51	:EESE .....	4-63
:AGATe? .....	4-51	:EESR? .....	4-63
:BCHannel:PHASe:ADJust .....	4-51	:ERRor? .....	4-63
:BCHannel:PHASe? .....	4-51	:FiLTer<x> .....	4-63
:INHibit:LEVel .....	4-52	:QMESsage .....	4-63
:INHibit? .....	4-52	:STATus? .....	4-63
:INPut? .....	4-50	STOP グループ	
MEASure グループ		:STOP .....	4-64
:FUNction .....	4-53	STORe グループ	
:MEASure? .....	4-53	:STORe .....	4-64
:MODE .....	4-54	SYSTem グループ	
:POLarity .....	4-54	:BEEP .....	4-65
:SLOPe .....	4-54	:CLiCKsound .....	4-66
MEMory グループ		:DATE .....	4-66
:BLOCK .....	4-55	:LCD:AOff .....	4-66
:BSIZe? .....	4-55	:LCD:ATime .....	4-66
:BYTeorder .....	4-55	:LCD:BRiGhtness .....	4-66
:DATaselect .....	4-56	:LCD? .....	4-66
:END .....	4-56	:LCD[:STATe] .....	4-66
:FORMat .....	4-56	:SYSTem? .....	4-65
:MEMory? .....	4-55	:TIME .....	4-66
:SEND<x>?[{POSitive   NEGative}] .....	4-56	:WARning .....	4-66
:SIZE<x>? .....	4-56	TVARiation<x> グループ	
:STARt .....	4-57	:HORizontal:MINimum .....	4-68
RECall グループ		:HORizontal:SPAN .....	4-68
:RECall .....	4-57	:HORizontal? .....	4-67
SAMPle グループ		:MARKer:{HIGH   LOW} .....	4-68
:ARMinG:DELay:{AEVentsize   BEVentsize   EVENTsize} 4-60		:MARKer:{LEFT   RIGHT} .....	4-68
:ARMinG:DELay:{ATime   BTime   TIME} .....	4-60	:MARKer:{LVALue   RVALue}? .....	4-68
:ARMinG:DELay? .....	4-59	:MARKer? .....	4-68
:ARMinG:DELay[:MODE] .....	4-60	:MARKer[:STATe] .....	4-68
:ARMinG:SLOPe .....	4-60	:TVARiation<x>? .....	4-67
:ARMinG:SOURce .....	4-60	:VERTical:CENTer .....	4-68
:ARMinG? .....	4-59	:VERTical:SPAN .....	4-69
:BLOCK:REST:EVENT .....	4-61		

## 索引

---

:VERTical? .....	4-68
UNIT グループ	
:FREQuency .....	4-69
:TIME .....	4-69
:UNIT? .....	4-69
:VOLTage .....	4-69
共通コマンドグループ	
*CAL? .....	4-70
*CLS .....	4-70
*ESE .....	4-70
*ESR? .....	4-70
*IDN? .....	4-70
*OPC .....	4-70
*OPC? .....	4-70
*RST .....	4-71
*SRE .....	4-71
*STB? .....	4-71
*TRG .....	4-71
*TST? .....	4-71
*WAI .....	4-71