

**DL9500/DL9700 シリーズ**  
デジタルオシロスコープ  
通信インタフェース

**U S E R ' S M A N U A L**

---

ユーザーズマニュアル

## はじめに

このたびは、デジタルオシロスコープ DL9500/DL9700 シリーズ (DL9505L/DL9510L/DL9705L/DL9710L、以降 DL9500/DL9700 と略します) をお買い上げいただきましてありがとうございます。

この通信インタフェースユーザーズマニュアルは、下記の各インタフェースの機能やコマンドについて説明したものです。

- USB インタフェース
- イーサネットインタフェース (オプション)

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、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、DL9500/DL9700 のマニュアルとして、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
DL9500/DL9700 シリーズ デジタルオシロスコープ ユーザーズマニュアル	IM 701331-01	DL9500/DL9700 リーズの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。
DL9500/DL9700 シリーズ デジタルオシロスコープ 通信インタフェース ユーザーズ マニュアル (CD 内)	IM 701331-17	本書です。DL9500/DL9700 リーズの通信インタフェースの機能について、その操作方法を説明しています。
DL9000 シリーズ デジタルオシロスコープ シリアルバス信号解析機能 ユーザーズマニュアル	IM701310-51	オプションの I <sup>2</sup> C バス信号 /CAN バス信号 /LIN バス信号 /SPI バス信号 /UART 信号解析の各機能と操作について説明しています。
DL9000 シリーズ デジタルオシロ スコープ /SB5000 シリーズビークル シリアルバスアナライザ 電源解析機能ユーザーズマニュアル	IM 701310-61	オプションの電源解析の各機能と操作について説明しています。

## ご注意

- **このマニュアル IM 701331-17 5 版は、ファームウェアバージョン 4.40 以降のデジタルオシロスコープ DL9500/DL9700 シリーズに対応しています。**  
最新のファームウェアバージョンでない場合は、このマニュアルに記載のすべての機能をお使いいただくことができません。  
お使いの製品のファームウェアバージョンは、オーバビュー画面でご確認ください。オーバビュー画面を表示する操作方法については、本体ユーザーズマニュアル IM701331-01 の 18.4 節をご覧ください。バージョンアップの方法やファームウェアバージョンに関する情報については、下記の Web ページをご覧ください。  
<http://www.yokogawa.co.jp/tm/Bu/DL9700/>
- 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、お買い求め先か、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。

## USB インタフェースおよびイーサネットインタフェースについて

USB インタフェースによる通信機能を使用するには、PC 側に下記が必要です。

- DL シリーズ用ライブラリ (TMCTL)
- PC-DL9000 シリーズ間の USB 接続デバイスドライバ

イーサネットインタフェースによる通信機能を使用するには、PC 側に下記が必要です。

- DL シリーズ用ライブラリ (TMCTL)

上記のライブラリおよびドライバは、下記ホームページからダウンロードできます。

<http://www.yokogawa.co.jp/tm/F-SOFT/>

## サンプルプログラム

サンプルプログラムは、下記ホームページからダウンロードできます。

<http://www.yokogawa.co.jp/tm/F-SOFT>

## 商 標

- Microsoft、MS-DOS、Visual C++、Windows、および Windows NT は、米国 Microsoft Corporation の、米国およびその他の国における登録商標です。
- Adobe と Acrobat 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

2007 年 3 月	初版発行
2007 年 8 月	2 版発行
2008 年 3 月	3 版発行
2008 年 6 月	4 版発行
2009 年 4 月	5 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 5 章および付録で構成されています。

### 第 1 章 USB インタフェースについて

USB インタフェースの機能・仕様などについて説明しています。

### 第 2 章 イーサネットインタフェースについて (オプション)

イーサネットインタフェースの機能・仕様などについて説明しています。

### 第 3 章 GP-IB インタフェースについて

GP-IB インタフェースの機能・仕様などについて説明しています。

### 第 4 章 プログラムを組む前に

コマンドを送るときの書式などについて説明しています。

### 第 5 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 6 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。

### 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

50 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シンボルと表記法

### 注記

このマニュアルでは、注記を以下のようなシンボルで区別しています

---

<b>注 意</b>	取り扱いを誤った場合に、使用者が軽傷を負うか、または物的損害のみが発生する危険があるときに、それを避けるための注意事項が記載されています。
------------	-----------------------------------------------------------------------

### Note

本機器を取り扱ううえで重要な情報が記載されています。

### 操作説明のページで使用している表記法

第1～3章で操作説明をしているページでは、説明内容を区別するために、次のような表記法を使用しています。

<b>操 作</b>	数字で示す順序で、各操作をしてください。ここでは、初めて操作をすることを前提に手順を説明しています。したがって、設定内容を変更する場合はすべての操作を必要としない場合があります。
------------	-------------------------------------------------------------------------------------------

<b>解 説</b>	操作に関連する設定内容や限定事項について説明しています。
------------	------------------------------

### 文字の表記法

---

#### 太文字の操作キー名とソフトキー名

操作対象になるパネル上の操作キーの文字や、画面に表示されるソフトキー/メニューの文字を示します。

---

#### SHIFT+ 操作キー

SHIFT キーを押して、SHIFT キーのインジケータを点灯させてから、操作キーを押すという意味です。押した操作キーの上に紫色で記されている項目の設定メニューが画面に表示されます。

---

### 単位

---

k 「1000」の意味です。使用例：100kS/s( サンプルレート )

---

K 「1024」の意味です。使用例：720Kバイト( ファイルのデータサイズ )

---

## 構文の記号

主に第 4、5 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。詳細データについては、4-5 ~ 4-6 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	CHANnel<x> <x> = 1 ~ 4	CHANNEL2
{ }	{ } 内から 1 つを選択	COUPling {AC DC DC50 GND}	COUPLING AC
	排他的論理和		
[ ]	省略可能	TRIGger [:SIMPlE]:SLOPe	TRIGger:SLOPe

# 目次

このマニュアルの利用方法 .....	iii
--------------------	-----

## 第 1 章 USB インタフェースについて

1.1 各部の名称と機能.....	1-1
1.2 USB インタフェースの機能と仕様.....	1-2
1.3 USB インタフェースによる接続.....	1-4
1.4 本体の設定 (USB).....	1-5

## 第 2 章 イーサネットインタフェースについて (オプション)

2.1 各部の名称と機能.....	2-1
2.2 イーサネット /VXI-11 インタフェースの機能と仕様.....	2-2
2.3 イーサネット /VXI-11 インタフェースによる接続.....	2-4
2.4 本体の設定 (ネットワーク).....	2-5

## 第 3 章 GP-IB インタフェースについて

3.1 各部の名称と機能.....	3-1
3.2 GP-IB カードの接続方法.....	3-2
3.3 GP-IB インタフェースの機能.....	3-3
3.4 GP-IB インタフェースの仕様.....	3-4
3.5 本体の設定 (GP-IB).....	3-5
3.6 インタフェースメッセージに対する応答.....	3-6

## 第 4 章 プログラムを組む前に

4.1 メッセージ.....	4-1
4.2 命令.....	4-3
4.3 応答.....	4-5
4.4 データ.....	4-6
4.5 コントローラとの同期.....	4-8

## 第 5 章 コマンド

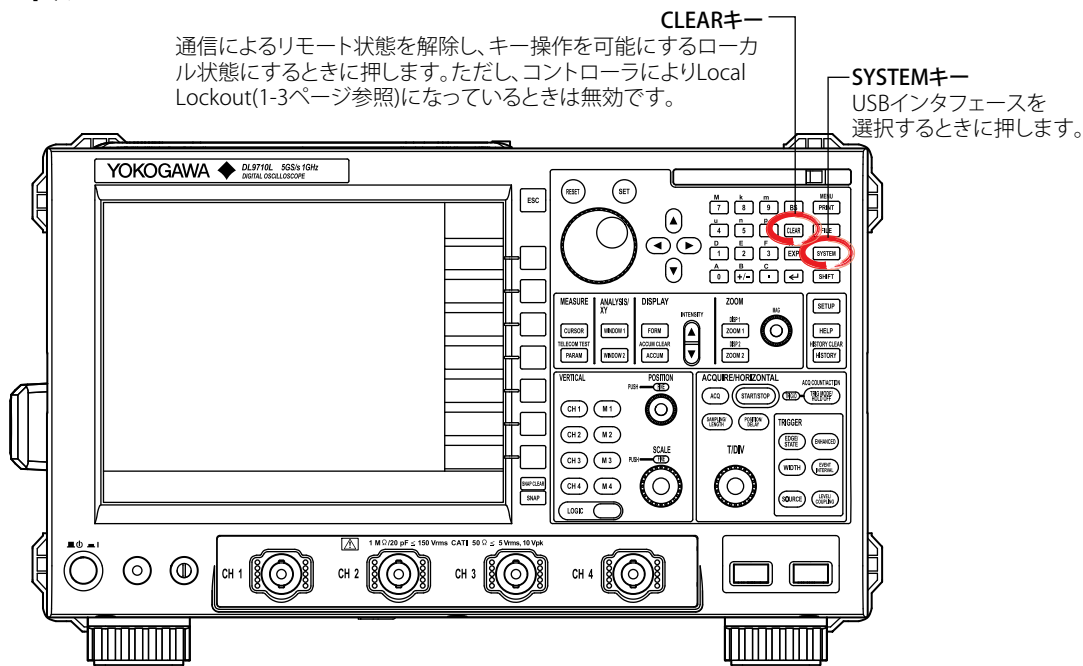
5.1 コマンド一覧表.....	5-1
5.2 ACQuire グループ.....	5-42
5.3 ANALysis グループ.....	5-43
5.4 ASETup グループ.....	5-72
5.5 CALibrate グループ.....	5-73
5.6 CHANnel グループ.....	5-74
5.7 CLEar グループ.....	5-76
5.8 COMMunicate グループ.....	5-77
5.9 CURSor グループ.....	5-79
5.10 DISPlay グループ.....	5-90
5.11 FILE グループ.....	5-93
5.12 GONogo グループ.....	5-99
5.13 Hcopy グループ.....	5-109
5.14 HISTory グループ.....	5-111
5.15 IMAGe グループ.....	5-122
5.16 INITialize グループ.....	5-123
5.17 LOGic グループ.....	5-124
5.18 MATH グループ.....	5-128
5.19 MEASure グループ.....	5-137

5.20	REFeRence グループ .....	5-150	<b>1</b>
5.21	SEARch グループ .....	5-151	
5.22	SNAP グループ .....	5-163	
5.23	SStart グループ .....	5-164	<b>2</b>
5.24	STARt グループ .....	5-165	
5.25	STATus グループ .....	5-166	
5.26	STOP グループ .....	5-167	
5.27	SYSTem グループ .....	5-168	<b>3</b>
5.28	TELeComtest グループ .....	5-169	
5.29	TIMebase グループ .....	5-174	
5.30	TRIGger グループ .....	5-175	
5.31	WAVeform グループ .....	5-257	<b>4</b>
5.32	ZOOM グループ .....	5-260	
5.33	共通コマンドグループ .....	5-263	
<b>第 6 章 ステータスレポート</b>			
6.1	ステータスレポートについて .....	6-1	<b>5</b>
6.2	ステータスバイト .....	6-3	
6.3	標準イベントレジスタ .....	6-4	<b>6</b>
6.4	拡張イベントレジスタ .....	6-5	
6.5	出力キューとエラーキュー .....	6-6	
<b>付録</b>			
付録 1	ASCII キャラクタコード .....	付 -1	<b>付</b>
付録 2	エラーメッセージ .....	付 -2	
付録 3	波形パラメータ名称対応表 .....	付 -5	
付録 4	Eye Pattern パラメータ名称対応表 .....	付 -6	<b>索</b>
<b>索引</b>			

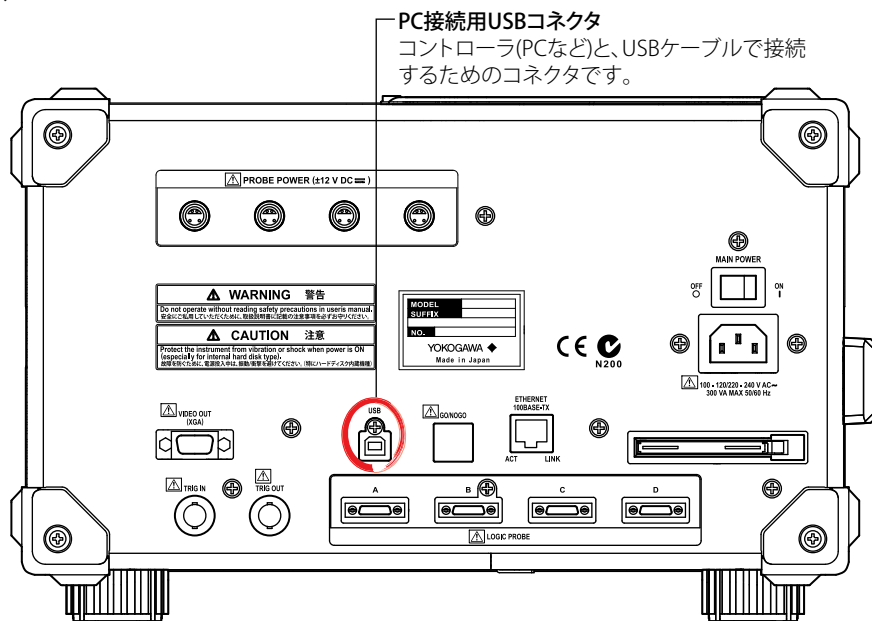


# 1.1 各部の名称と機能

## フロントパネル



## リアパネル



## 1.2 USB インタフェースの機能と仕様

### USB インタフェースの機能

#### 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。

### USB インタフェースの仕様

電氣的・機械的仕様： USB Rev.2.0 に準拠  
コネクタ： タイプ B コネクタ (レセプタクル)  
ポート数： 1  
電源： セルフパワー  
対応システム環境： Windows 2000、Windows XP で動作し、USB ポートが標準装備されている機種 (PC との接続には、別途デバイスドライバが必要)

### データ転送速度

コントローラ： PC(Pentium4 3.4GHz、USB2.0)、OS(Windows XP Professional SP1)  
使用言語： Visual C++

アナログ信号の波形データを出力するときの応答時間の目安を次に示します。

データ点数	ワードデータ	アスキーデータ
2500	約 51ms	約 0.469s
125000	約 193ms	約 22.766s
1250000	約 1606ms	約 224.890s
2500000	約 3188ms	約 451.297s
6250000	約 7841ms	約 1127.625s

ロジック信号の波形データを出力するときの応答時間の目安を次に示します。

データ点数	ダブルワードデータ	アスキーデータ
2500	約 78ms	約 0.141s
125000	約 625ms	約 3.516s
1250000	約 5547ms	約 34.531s
2500000	約 11156ms	約 69.375s
6250000	約 27812ms	約 173.266s

## 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」コマンドを受け取ると、リモート状態になります。

- 画面上部中央に REMOTE と表示されます。
- **CLEAR** 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 **CLEAR** 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ときは無効です。コンピュータから「:COMMunicate:REMOte OFF」コマンドを受信したときは、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。

- 画面上部中央の 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**

USB インタフェースは、他のインタフェース (GP-IB、イーサネットインタフェース) と同時に使用できません。

---

## 1.3 USB インタフェースによる接続

### 接続時の注意

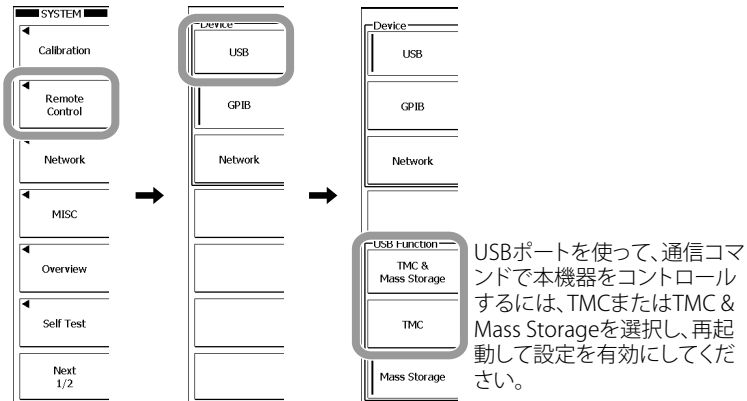
- USB ケーブルは、USB コネクタに奥までしっかりと差し込んで接続してください。
- USB ハブを使って複数の機器を接続する場合は、本機器をコントローラに最も近い USB ハブに接続してください。
- GO/NO-GO 出力端子に、誤って USB ケーブルを挿入しないでください。本機器を損傷する恐れがあります。

## 1.4 本体の設定 (USB)

### 操作

#### 通信インタフェースの選択

1. SYSTEM を押します。
2. Remote Control のソフトキーを押します。
3. USB のソフトキーを押します



#### Note

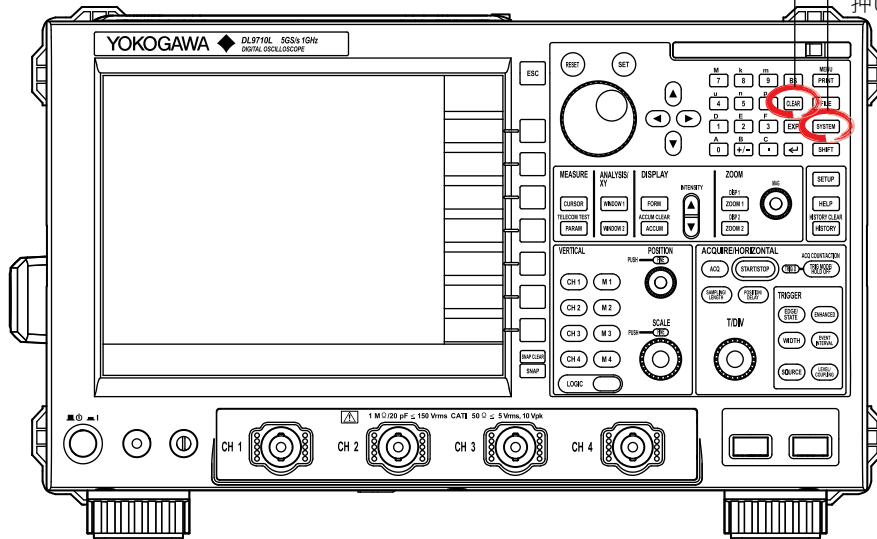
- Remote Control で選択した通信インタフェースだけが有効です。選択していない他の通信インタフェースでコマンドを送信しても、本機器は受け付けません。
- USB ポートを使って、通信コマンドで本機器をコントロール (リモートコントロール) するには、上記メニューで「TMC」を選択し、次の事項を実行してください。
  - TMC、Mass Storage、および TMC & Mass Storage の設定を有効にするには、本機器を再起動する必要があります。本機器の電源スイッチを OFF にしてから、10 秒以上待ったあとに ON にしてください。
  - 当社の USB TMC (Test and Measurement Class) 用ドライバを PC にインストールしてください。当社の USB TMC 用ドライバの入手方法については、お問い合わせ先にお問い合わせいただくか、下記の当社 Web サイトから USB ドライバ提供ページにアクセスし、USB TMC 用ドライバをダウンロードしてください。  
<http://www.yokogawa.co.jp/tm/F-SOFT/>
  - 当社以外の USB TMC 用ドライバ (またはソフトウェア) は、使用しないでください。
- TMC & Mass Storage が有効な状態で、本機器と PC を接続しているときでもリモートコントロールできます。しかし、通信コマンドでのファイル制御ができません。また、本機器でのキー操作でもファイル操作できません。PC からの接続を切り離すか、TMC の設定を有効にしてから、本機器と PC を接続してください。Mass Storage の設定が有効な状態でも本機器でのキー操作でのファイル操作ができません。

## 2.1 各部の名称と機能

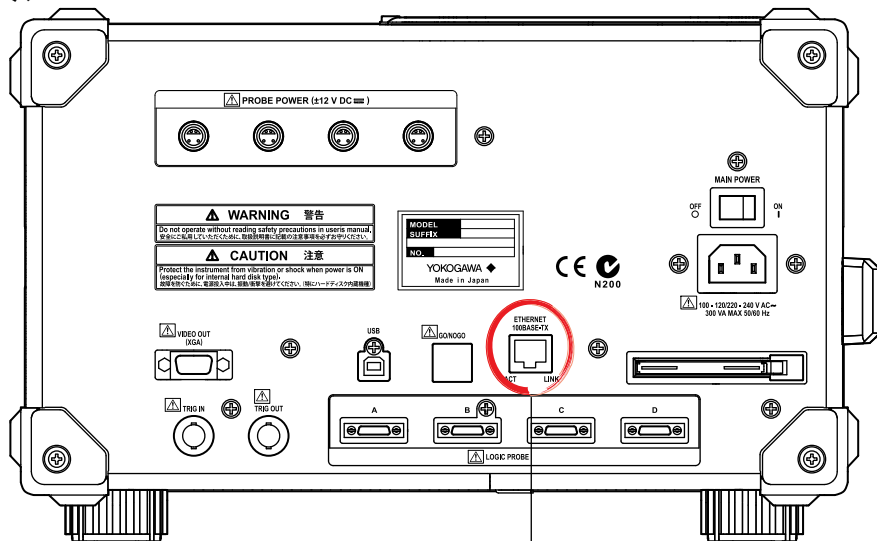
### フロントパネル

**CLEAR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。ただし、コントローラによりLocal Lockout(2-3ページ参照)になっているときは無効です。

**SYSTEMキー**  
イーサネット/VXI-11インタフェースを選択するときに押します。

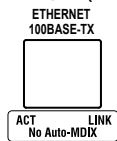


### リアパネル



**イーサネットポート**  
コントローラ(PCなど)と、イーサネットケーブルで接続するためのポートです。

LXI オプション (/C9、/C12) の場合の表記



## 2.2 イーサネット /VXI-11 インタフェースの機能と仕様

### イーサネット /VXI-11 インタフェースの機能

#### 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 /VXI-11 インタフェースの仕様

電氣的・機械的仕様	IEEE802.3 に準拠
同時接続数	1
ポート番号	イーサネット： 10001/tcp VXI-11： 10240/tcp、10250/tcp、111/tcp、111/udp

### データ転送速度

コントローラ： PC(Pentium4 3.4GHz)、OS(Windows XP Professional SP1)  
ネットワークアダプタ： Corega FEther PCI-TXL  
使用言語： Visual C++

#### イーサネットインタフェース (/C8、/C9、/C10、/C12 オプション)

アナログ信号の波形データを出力するときの応答時間の目安を次に示します。

データ点数	ワードデータ	アスキーデータ
2500	約 16ms	約 0.391s
125000	約 259ms	約 19.063s
1250000	約 2313ms	約 189.812s
2500000	約 4595ms	約 379.750s
6250000	約 10400ms	約 950.532s

ロジック信号の波形データを出力するときの応答時間の目安を次に示します。

データ点数	ダブルワードデータ	アスキーデータ
2500	約 31ms	約 0.078s
125000	約 704ms	約 3.530s
1250000	約 6768ms	約 35.327s
2500000	約 14081ms	約 70.952s
6250000	約 34523ms	約 177.313s

#### VXI-11 インタフェース (/C8、/C9、/C10、/C12 オプション)

アナログ信号の波形データを出力するときの応答時間の目安を次に示します。

データ点数	ワードデータ	アスキーデータ
2500	約 31ms	約 0.406s
125000	約 235ms	約 19.172s
1250000	約 2286ms	約 191.876s
2500000	約 4541ms	約 383.012s
6250000	約 11150ms	約 958.128s

ロジック信号の波形データを出力するときの応答時間の目安を次に示します。

データ点数	ダブルワードデータ	アスキーデータ
2500	約 31ms	約 0.094s
125000	約 828ms	約 3.906s
1250000	約 6938ms	約 37.345s
2500000	約 13344ms	約 74.143s
6250000	約 32126ms	約 184.894s

## 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」コマンドを受け取ると、リモート状態になります。

- 画面上部中央に REMOTE と表示されます。
- CLEAR 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 CLEAR 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ときは無効です。コンピュータから「:COMMunicate:REMOte OFF」コマンドを受信したときは、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。

- 画面上部中央の 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 /VXI-11 インタフェースは、他のインタフェース (USB、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。なお、DL シリーズ用ライブラリ (TMCTL) のバージョンが 1.40 以降の場合、パスワードは、MD5 アルゴリズム (RSA Data Security, Inc. MD5 Message-Digest Algorithm) を使用して暗号化後、本体に送られます。

ユーザー名とパスワードは、本機器の SYSTEM メニューの Remote Control の設定画面で設定します。設定の方法については、「2.4 本体の設定 (ネットワーク)」をご覧ください。

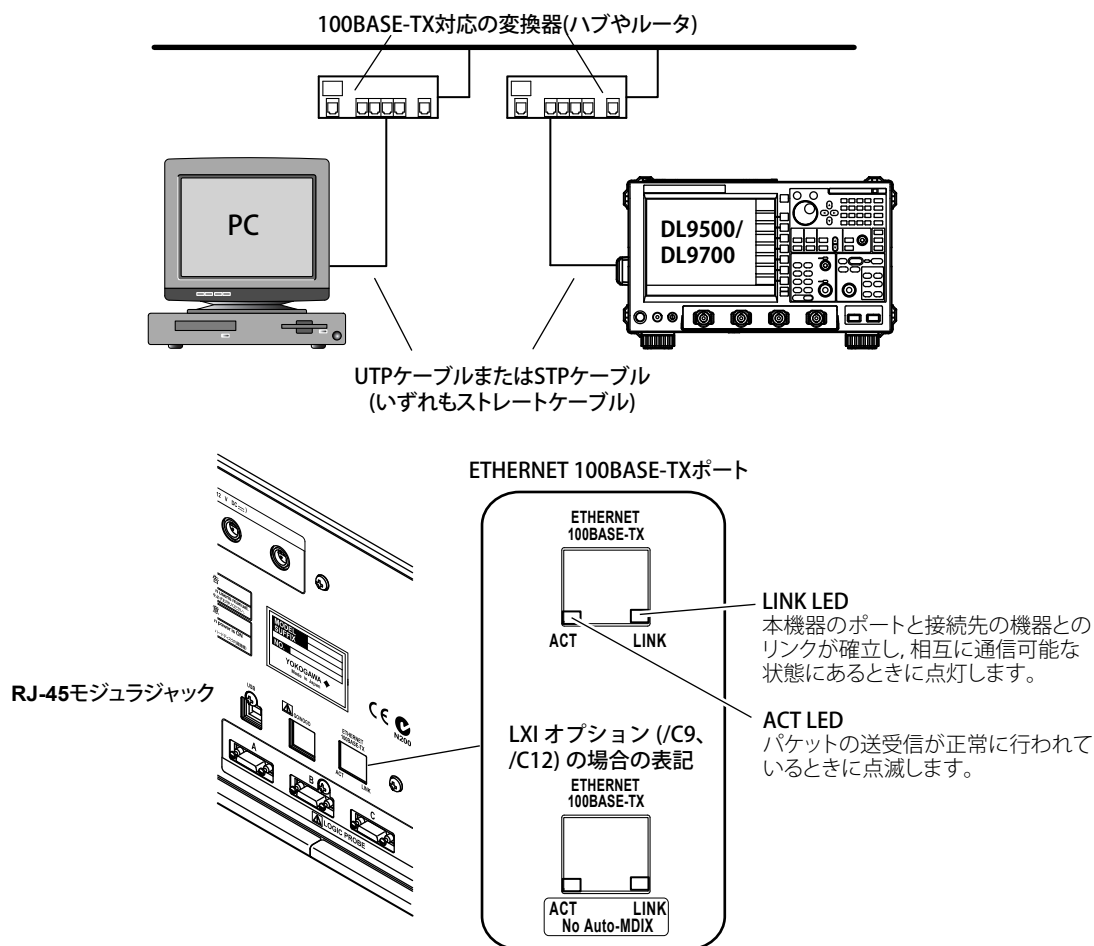
後述の Remote Control の設定画面 (Device メニュー) で、「VXI-11」を選択したときは、このユーザー認証機能の設定メニューはありません。



## 2.3 イーサネット /VXI-11 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 100BASE-TX ポートに接続します。



### 接続時の注意

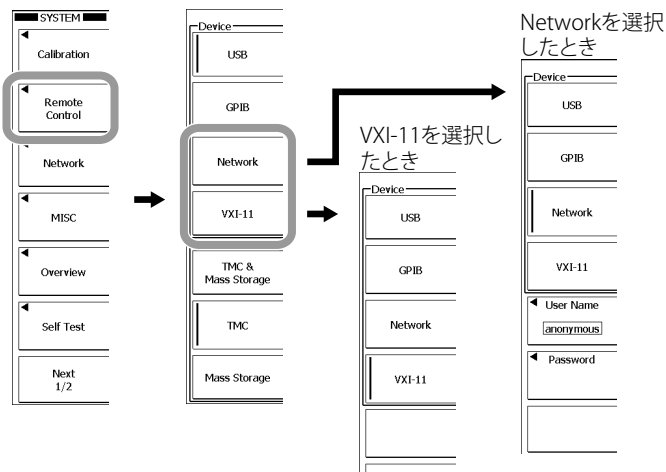
- ・ 本機器と PC との接続には、必ずハブを介してストレートケーブルを使用してください。クロスケーブルでの 1 対 1 の接続では、動作を保証することができません。
- ・ UTP ケーブル(ストレートケーブル)を使用する場合は、必ずカテゴリ 5 のものを使用してください。

## 2.4 本体の設定 ( ネットワーク )

### 操 作

#### 通信インターフェースの選択

1. SYSTEM を押します。
2. Remote Control のソフトキーを押します。Device メニューが表示されます。
3. Network または VXI-11 のソフトキーを押します。
  - ・ イーサネットインターフェースを使用するときは、Network を選択します。
  - ・ VXI-11 インターフェースを使用するときは、VXI-11 を選択します。



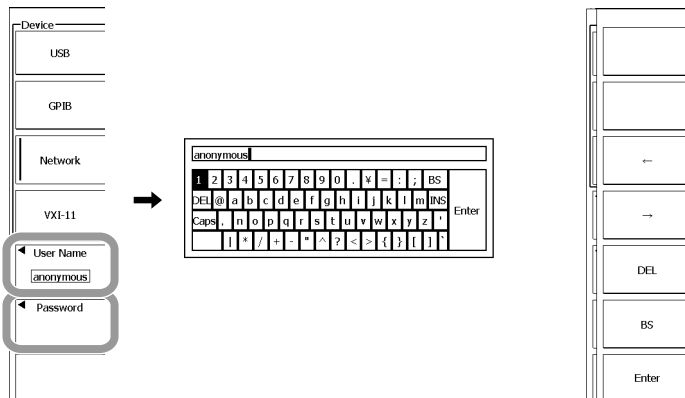
#### Note

Remote Control で選択した通信インターフェースだけが有効です。選択していない他の通信インター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

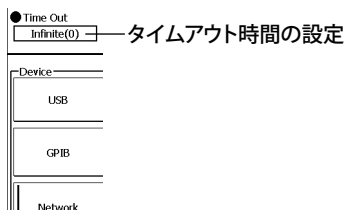
(操作3でVXI-11を選択したときは、この設定メニューはありません。)

4. UserName のソフトキーを押します。キーボードが表示されます。
5. 本体ユーザーズマニュアル (IM701331-01) の 4.2 節の操作手順にしたがって、ユーザー名を入力します。
6. 同様に、パスワードも入力します。



### タイムアウト時間の設定

7. ロータリノブで、タイムアウト時間を設定しす。



### Note

- ・ キーボード (ソフトキーボード) の操作方法については、DL9500/DL9700 ユーザーズマニュアル (IM701331-01) の「4.2 数値 / 文字列を入力する」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

イーサネット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、DL9500/DL9700 ユーザーズマニュアル (IM701331-01) の「15.2 TCP/IP の設定をする」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

イーサネット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。

Remote Control の設定画面 (Device メニュー) で、「VXI-11」を選択したときは、このユーザー認証機能の設定メニューはありません。

- ・ **ユーザー名の設定**  
30 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**  
30 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。

### TCP/IP の設定

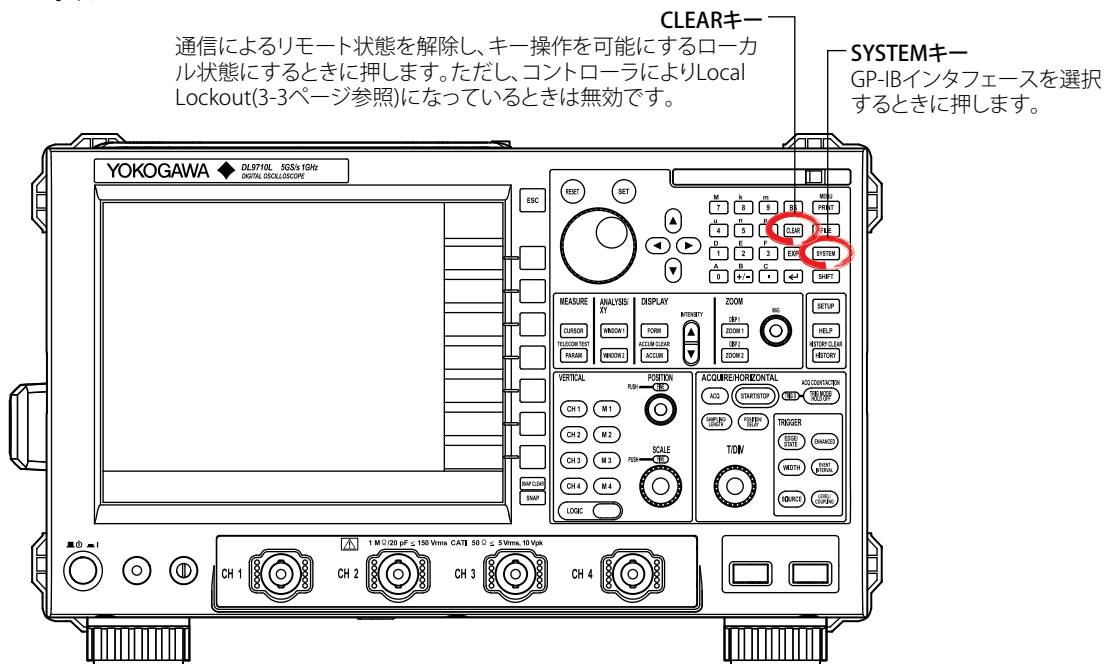
詳細は、DL9500/DL9700 ユーザーズマニュアル (IM701331-01) の「15.2 TCP/IP の設定をする」をご覧ください。

### Note

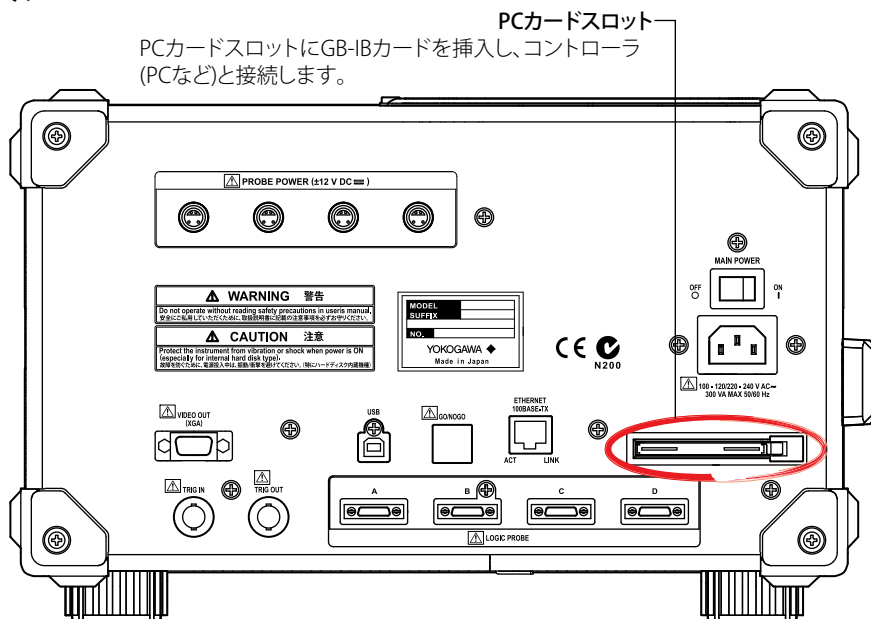
- ・ ユーザー認証エラーとなった場合、本機器との接続が切断されます。
- ・ ユーザー名が「anonymous」の場合は、パスワードは不要です。

## 3.1 各部の名称と機能

### フロントパネル



### リアパネル



## 3.2 GP-IB カードの接続方法

### GP-IB カード

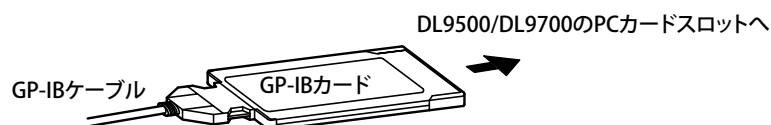
National Instruments 社製 NI PCMCIA-GPIB カードをご使用ください。

### GP-IB ケーブル

GP-IB ケーブルは、GP-IB カードに付属されているケーブルをご使用ください。

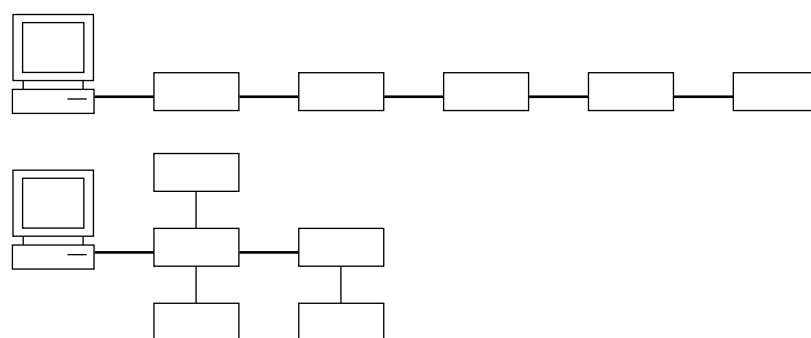
### 接続方法

DL9500/DL9700 の PC カードスロットに GP-IB カードを挿入してください。



### 接続時の注意

- GP-IB カードの取り扱いについては、GP-IB カードに付属している取扱説明書をご覧ください。
- GP-IB カードは表面を上にして、正しく装着してください。
- GP-IB カードを DL9500/DL9700 の PC カードスロットに装着してから、DL9500/DL9700 の電源を ON にしてください。
- 何本かのケーブルを接続して、複数の機器を接続することができます。ただし、1 つのバス上にコントローラを含め 15 台以上の機器を接続することはできません。
- 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。ループ形や平行形の結線はできません。



### 注 意

通信ケーブルを接続したり、取り外したりするときは、必ずパーソナルコンピュータおよび本機器の電源を OFF にしてください。OFF にしないと、誤動作を生じたり、内部回路を破損することがあります。

## 3.3 GP-IB インタフェースの機能

### GP-IB インタフェースの機能

#### 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。
- ・ 設定情報や波形データなどの、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。

#### トーカー機能

設定情報や波形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ピュータから 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。

- ・ 画面上部中央に REMOTE と表示されます。
- ・ CLEAR 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 CLEAR 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(3-6 ページ参照) になっているときは無効です。

- ・ 画面上部中央の 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**

GP-IB インタフェースは、他のインタフェース (USB、イーサネットインタフェース) と同時に使用できません。

## 3.4 GP-IB インタフェースの仕様

### 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 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
アドレス設定	: MISC メニューの GP-IB の設定画面で、0 ~ 30 のアドレスを設定可能。
リモート状態解除	: <b>CLEAR</b> 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 されているときは無効。

### 機能的仕様

機能	サブセット名	内容
ソースハンドシェーク	SH1	送信ハンドシェークの全機能あり
アクセプタハンドシェーク	AH1	受信ハンドシェークの全機能あり
トーカー	T6	基本トーカー機能、シリアルポール、MLA(My Listen Address) によるトーカー解除機能あり、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	DT0	デバイストリガ機能なし
コントローラ	CO	コントローラ機能なし
電気特性	E1	オープンコレクタ

### データ転送速度

コントローラ : PC(Pentium4 3.4GHz、USB2.0)、OS(Windows XP Professional SP1)

使用言語 : Visual C++

アナログ信号の波形データを出力するときの応答時間の目安を次に示します。

データ点数	ワードデータ	アスキーデータ
2500	約 16ms	約 0.390s
125000	約 344ms	約 19.453s
1250000	約 3172ms	約 194.516s
2500000	約 6282ms	約 389.047s
6250000	約 15641ms	約 971.985s

ロジック信号の波形データを出力するときの応答時間の目安を次に示します。

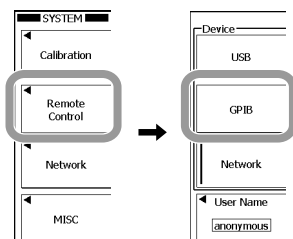
データ点数	ダブルワードデータ	アスキーデータ
2500	約 32ms	約 0.078s
125000	約 859ms	約 3.609s
1250000	約 8707ms	約 36.172s
2500000	約 17365ms	約 72.500s
6250000	約 43678ms	約 182.000s

## 3.5 本体の設定 (GP-IB)

### 操 作

#### 通信インタフェースの選択

1. SYSTEM を押します。
2. Remote Control のソフトキーを押します。Device メニューが表示されます。
3. GP-IB のソフトキーを押します。

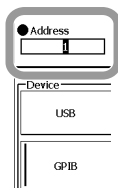


#### Note

Remote Control で選択した通信インタフェースだけが有効です。選択していない他の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。

#### アドレスの設定

4. ロータリノブを回して、アドレスを設定します。



### 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。

#### アドレスの設定

アドレスサブルモードのときの、本機器のアドレスを次の範囲で設定します。

0 ~ 30

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と重ならないように設定する必要があります。

#### Note

コントローラが GP-IB で本機器または他のデバイスと通信しているときは、アドレスを変更しないでください。



## 3.6 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に対する応答

#### ユニラインメッセージに対する応答

- **IFC(Interface Clear)**  
トーカー、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(Remote Enable)**  
リモート状態 / ローカル状態を切り替えます。  
  
IDY(Identify) は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(Go To Local)**  
ローカル状態へ移行します。
- **SDC(Selected Device Clear)**
  - 受信中のプログラムメッセージ ( コマンド ) と、出力キュー (6-5 ページ参照) をクリアします。
  - 実行中の \*OPC、\*OPC? は無効になります。
  - \*WAI、COMMunicate:WAIT は直ちに終了します。  
PPC(Parallel Poll Configure)、GET(Group Execute Trigger)、TCT(Take Control) は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(Local Lockout)**  
フロントパネルの CLEAR 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(Device Clear)**  
SDC と同じ動作をします。
- **SPE(Serial Poll Enable)**  
バス上のすべての機器のトーカー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(Serial Poll Disable)**  
バス上のすべての機器のトーカー機能のシリアルポールモードを解除します。  
  
PPU(Parallel Poll Unconfigure) はサポートしていません。

### 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

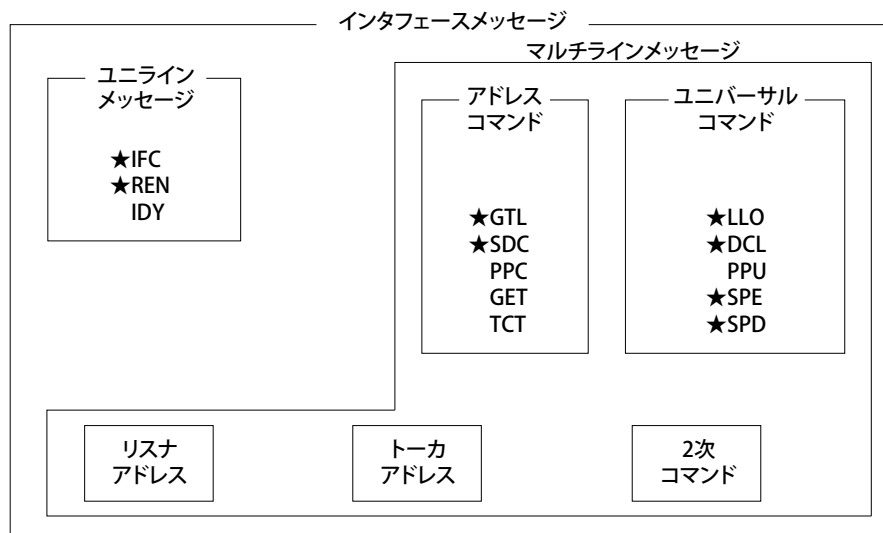
- TCT(Take Control)

#### • 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。



★印は本機器でサポートしているインタフェースメッセージです。

### Note

#### 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。

## 4.1 メッセージ

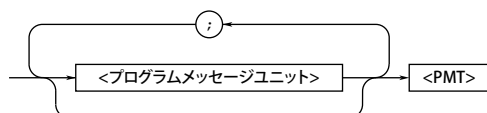
### メッセージ

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

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 `:ACQuire:MODE NORMAl;HRMode 1<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

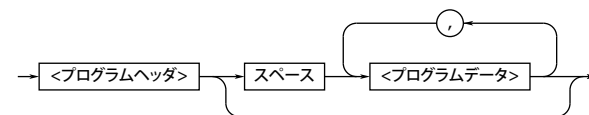
^EOM: USBTMCで定義されているENDメッセージ

(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^EOM: ENDメッセージが付加された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、4-3ページを参照してください。

#### <プログラムデータ>

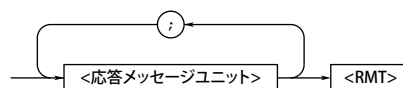
命令を実行するときに必要な条件などがあるときは、プログラムデータを付けます。プログラムデータを付けるときは、ヘッダとデータをスペース(ASCIIコード「20H」)で区切ります。複数のデータがあるときは、データとデータの間を「,」(カンマ)で区切ります。

詳しくは、4-5ページを参照してください。

例 `:ACQuire:MODE NORMAl<PMT>`  
                                         ヘッダ                                        データ

### 応答メッセージ

応答メッセージの書式は次のようになります。



#### <応答メッセージユニット>

応答メッセージは、1つ以上の応答メッセージユニットをつないだものです。応答メッセージユニットが1つの応答に相当します。

応答メッセージユニットは「;」(セミコロン)で区切られます。

応答メッセージの書式については、次ページを参照してください。

例 `:ACQUIRE:MODE NORMAL;HRMode 1<RMT>`  
                                         ユニット                                        ユニット

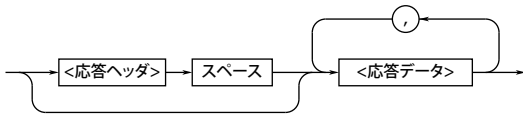
#### <RMT>

応答メッセージのターミネータで、NL^EOMです。

## 4.1 メッセージ

### 応答メッセージユニットの書式

応答メッセージユニットの書式は次のようになります。



#### < 応答ヘッダ >

応答データの前に応答ヘッダが付くことがあります。ヘッダとデータの間は、1文字のスペースで区切られます。詳しくは、4-4 ページを参照してください。

#### < 応答データ >

応答データは、応答の内容を示します。複数のデータがあるときは、データとデータの間は「,」（カンマ）で区切られます。詳しくは、4-5 ページを参照してください。

例

```
1.25E-02<RMT> :ACQUIRE:MODE NORMAL<RMT>
  データ       ヘッダ       データ
```

プログラムメッセージに複数のクエリがある場合、応答の順序はクエリの順序に従います。クエリの多くは1つの応答メッセージユニットを返しますが、複数のユニットを返すものもあります。1番目のクエリの応答は1番目のユニットですが、n番目の応答はn番目のユニットとは限りません。確実に応答を取り出したときは、プログラムメッセージを分けるようにしてください。

### メッセージの送受信時の注意

- クエリを含まないプログラムメッセージを送信したときは、いつでも次のプログラムメッセージを送信できます。
- クエリを含むプログラムメッセージを送信したときは、次のプログラムメッセージを送信する前に応答メッセージを受信しなければなりません。もし、応答メッセージを受信しないか、途中までしか受信せずに次のプログラムメッセージを送信したときは、エラーになります。受信されなかった応答メッセージは捨てられます。
- コントローラが応答メッセージがないのに受信しようとしたときは、エラーになります。もし、コントローラがプログラムメッセージを送信し終わる前に応答メッセージを受信しようとする、エラーになります。

- メッセージにユニットが複数あるプログラムメッセージを送信したときに、その中に不完全なプログラムユニットが存在すると、本機器は完全と思われるプログラムメッセージユニットを拾い上げて実行を試みますが、必ずしも成功するとは限りません。また、その中にクエリが含まれていても、必ずしも応答が返るとは限りません。

### デッドロック状態

本機器は、送受信とも最低 1024 バイトのメッセージをバッファに蓄えておくことができます（バイト数は、動作状態によって増減することがあります）。このバッファが送受信と同時にいっぱいになると、本機器は動作不能状態になります。これをデッドロック状態といいます。このときは、応答メッセージを捨てることで動作不能から回復します。

プログラムメッセージを <PMT> も含めて 1024 バイト以下にしておけば、デッドロックすることはありません。また、クエリがないプログラムメッセージは、デッドロックすることはありません。

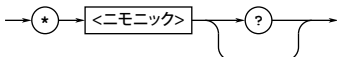
## 4.2 命令

### 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

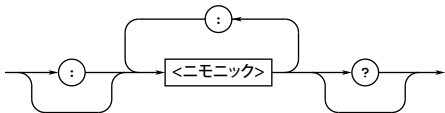
USBTMC-USB488 で規定されている命令を共通コマンドといいます。共通コマンドのヘッダの書式は次のようになります。先頭に必ず「\*」（アスタリスク）を付けます。



共通コマンドの例 \*CLS

### 複合ヘッダ

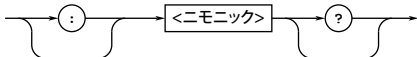
共通コマンド以外の本機器固有の命令は、機能ごとに分類されて、階層化されています。複合ヘッダの書式は次のようになります。下の階層を記述するときは、必ず「:」（コロン）を付けます。



複合ヘッダの例 :ACQUIRE:MODE

### 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 アクイジションに関するグループ

```
:ACQUIRE:AVERAGE:COUNT
:ACQUIRE:MODE
:ACQUIRE:AVERAGE:EWEIGHT
:ACQUIRE:REPETITIVE
:ACQUIRE:RLLENGTH
:ACQUIRE:INTERLEAVE
```

#### • 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 :ACQUIRE:MODE NORMAL;
INTERLEAVE 1<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（省略することはできません）。

```
例 :ACQUIRE:MODE NORMAL::DISPLAY;
FORMAT SINGLE<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（省略することはできません）。

```
例 :ACQUIRE:MODE NORMAL::START<PMT>
```

#### • 共通コマンドを続けて記述する場合

USBTMC-USB488 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 :ACQUIRE:MODE NORMAL;*CLS;
INTERLEAVE 1<PMT>
```

#### • コマンド間を <PMT> で区切った場合

ターミネータで区切ると、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 :ACQUIRE:MODE NORMAL<PMT>;ACQUIRE:
INTERLEAVE 1<PMT>
```

## 4.2 命令

---

### 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 :ACQUIRE? -> :ACQUIRE:AVERAGE:COUNT 2;
      EWEIGHT 16;:ACQUIRE:HRMODE 0;
      INTERLEAVE 0;INTERPOLATE 1;
      MODE NORMAL;REPETITIVE 0;RLENGTH 12500
```

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大文字 / 小文字は区別しません。  
例 「CURSor」 -> 「cursor」「Cursor」でも可
- 小文字の部分は省略できます。  
例 「CURSor」 -> 「CURSO」「CURS」でも可
- ヘッダの最後の「?」(クエスションマーク)は、クエリであることを示します。「?」は省略できません。  
例 「CURSor?」 -> 最小の省略形は「CURS?」
- 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 「CHANnel<x>」 -> 「CHAN」とすると「CHANnel1」の意味
- [] で囲まれた部分は省略できます。  
例 HISTory[:CURRent]:MODE -> 「HIST:MODE」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。  
例 「HISTory?」と「HISTory:CURRent?」は別のクエリになる。

## 4.3 応答

### 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 :ACQUire:MODE?<PMT> -> :ACQUire  
:MODE NORMAL<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 :MEASure:TRACe1:AREA1:PTOPeak:  
VALue?<PMT> -> 10.0E+00<RMT>

### ヘッダを付けな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。また、省略形のときは[]で囲まれた部分も省略されます。

## 4.4 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: CH1のプロープの減衰比 -> CHANnel1:PROBe 100)
<電圧><時間> <周波数><電流>	物理的な次元を持った数値 (例: 時間軸レンジ -> TIMEbase:TDiv 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(ニモニック)。{}内から選択 (例: CH1の入力カップリングの選択 -> CHANnel1: COUPling {AC DC DC50 GND})
<Boolean>	ON/OFFを表す。「ON」「OFF」または数値で設定 (例: CH1の表示をON -> CHANnel1:DISPlay ON)
<文字列データ>	任意の文字列 (例: 画面データの出力のコメント -> MATH1:UNIT:USERdefine "VOLT")
<Filename>	ファイル名を表す (例: 保存ファイル名 -> FILE:SAVE:WAVEform: NAME "CASE1")
<ブロックデータ>	任意の8ビットの値を持つデータ (例: 取り込んだ波形データの応答 -> #800000010ABCDEFHIJ)

### <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>、<電流>

<電圧>、<時間>、<周波数>、<電流>は、<10進数>のうち物理的な次元を持ったデータです。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>	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
A	アンペア	電流

- <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。ただし、メガヘルツだけは例外で、「MHZ」で表します。したがって、周波数のときは乗数に「M(ミリ)」は使用できません。
- <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。



**<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 <NRf> 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、4-4 ページの「ヘッダ解釈の規則」と同様です。

書式	例
{AC DC DC50 GND}	AC

- 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、整数に四捨五入した値が「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。

**<Filename>**

<Filename> は、ファイル名を表すデータです。次の書式のどれかで記述します。

書式	例
{<Nrf> <文字データ> <文字列データ>} 1 CASE "CASE"	

- <Nrf> は整数に丸めた 8 桁の値を ASCII コードに直したものがファイル名になります (例: 1 -> "00000001")。ただし、負の値は使えません。
- <文字データ> は、先頭の 12 文字がファイル名になります。
- <文字列データ> は、先頭から 259 文字までがファイル名になります。
- 応答メッセージは、必ず <文字列データ> で返されます。
- <文字列データ> のファイル名の文字数に関しては、本体ユーザーズマニュアルをご覧ください。

**<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
#N<N 桁の 10 進数><データバイトの並び>	#800000010ABCDEFGHIJ

- #N  
<ブロックデータ>であることを表します。「N」は次に続くデータバイト数を表わす ASCII コードの文字数 (桁) を示します。
- <N 桁の 10 進数>  
データのバイト数を表します (例: 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 (例: 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。

## 4.5 コントローラとの同期

**オーバーラップコマンドとシーケンシャルコマンド**  
コマンドには、オーバーラップコマンドとシーケンシャルコマンドの2種類があります。オーバーラップコマンドの場合は、先に送信したコマンドによる動作が完了する前に、次のコマンドによる動作が開始される場合があります。

たとえば、V/divを指定してその結果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、応答は常に最新の設定値(この場合は5V)を返します。

```
:CHANnel1:VDIV 5V;VDIV?<PMT>
```

これは、「CHANnel1:VDIV」が自身の処理を終えるまで、次の命令を待たせているためです。このような命令をシーケンシャルコマンドといいます。

これに対して、たとえばファイルロードを実行して、その結果のV/div値を問い合わせたいときに、次のプログラムメッセージを送信すると、

```
:FILE:LOAD:SETup:EXECute "CASE1";:
```

```
CHANnel1:VDIV?
```

「CHANnel1:VDIV?」はファイルロードが終了する前に実行されてしまい、応答されるV/divはファイルロードする前の値になってしまいます。

「FILE:LOAD:SETup:EXECute "CASE1"」のように、自身の処理が終わる前に次の命令を実行させることをオーバーラップ動作といい、オーバーラップ動作する命令を、オーバーラップコマンドといいます。

このようなときは、以下に示す方法でオーバーラップ動作を回避できます。

### オーバーラップコマンドとの同期をとる方法

#### • \*WAI コマンドを使う

\*WAI コマンドは、オーバーラップコマンドが終了するまで、\*WAI に続く命令を待つコマンドです。

```
例 :COMMunicate:OPSE #H0040;:FILE:LOAD:SETup:EXECute "CASE1";*WAI;:
```

```
CHANnel1:VDIV?<PMT>
```

「COMMunicate:OPSE」は「\*WAI」の対象を選ぶ命令です。ここではメディアアクセスだけを対象に指定しています。

「CHANnel1:VDIV?」の直前で「\*WAI」を実行しているので、「CHANnel1:VDIV?」は、ファイルロードが終了するまで実行されません。

#### • COMMunicate:OVERlap コマンドを使う

COMMunicate:OVERlap コマンドは、オーバーラップ動作を許可(または禁止)する命令です。

```
例 :COMMunicate:OVERlap #HFFBF;:FILE:LOAD:SETup:EXECute "CASE1";:CHANnel1:VDIV?<PMT>
```

「COMMunicate:OVERlap #HFFBF」は、メディアアクセス以外のオーバーラップ動作を許可しています。ファイルロードはオーバーラップ動作を許可されていないので、次の「FILE:LOAD:SETup:EXECute "CASE1"」は、シーケンシャルコマンドと同じ動作をします。したがって、「CHANnel1:VDIV?」は、ファイルロードが終了するまで実行されません。

#### • \*OPC コマンドを使う

\*OPC コマンドは、オーバーラップ動作が終了したときに、標準イベントレジスタ(6-4ページ参照)のビット0であるOPCビットを1にする命令です。

```
例 :COMMunicate:OPSE #H0040;*ESE 1; *ESR?;*SRE 32;:FILE:LOAD:SETup:EXECute "CASE1";*OPC<PMT>
```

```
(*ESR?の応答を読む)
```

```
(サービスリクエストの発生を待つ)
```

```
:CHANnel1:VDIV?<PMT>
```

「COMMunicate:OPSE」は「\*OPC」の対象を選ぶ命令です。ここではメディアアクセスだけを対象に指定しています。

「\*ESE 1」と「\*SRE 32」は、OPCビットが1になったときだけ、サービスリクエストを発生することを示しています。

「\*ESR?」は、標準イベントレジスタをクリアします。上の例では、「CHANnel1:VDIV?」は、サービスリクエストが発生するまで実行されません。

#### • \*OPC? クエリを使う

\*OPC? クエリは、オーバラップ動作が終了したときに応答を生成する命令です。

```
例 :COMMunicate:OPSE #H0040;:FILE:LOAD:
SETup:EXECute "CASE1";*OPC?<PMT>
(*OPC? の応答を読む)
:CHANnel1:VDIV?<PMT>
```

「COMMunicate:OPSE」は「\*OPC?」の対象を選ぶ命令です。ここではメディアアクセスだけを対象に指定しています。

「\*OPC?」はオーバラップ動作が終了するまで応答を作成しないので、「\*OPC?」の応答を読み終えたときには、ファイルロードは終了しています。

#### Note

命令のほとんどはシーケンシャルコマンドです。オーバラップコマンドは、第5章でオーバラップコマンドと明記しています。それ以外はすべてシーケンシャルコマンドです。

#### オーバラップコマンド以外の同期をとる方法

シーケンシャルコマンドの場合でも、トリガの発生などのように通信以外の要因で同期をとる必要がある場合もあります。

たとえば、トリガモードがシングルで取り込んだ波形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、波形の取り込みの終了いかにかわらず「WAVEform: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。

```
TRIGger:MODE SINGLE;:START;:WAVEform:
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 (6-5 ページ参照) の内容を問い合わせる命令です。波形の取り込み中かそうでないかは、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」なら波形の取り込み中、「0」なら取り込みストップ中を示します。

```
例 TRIGger:MODE SINGLE;:START<PMT>
:STATus:CONDition?<PMT>
(応答を読んでビット 0 が 1 なら 1 つ前に戻る)
:WAVEform:SEND?<PMT>
```

「WAVEform:SEND?」は、状態レジスタのビット 0 が「0」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 (6-5 ページ) に反映させることができます。

```
例 :STATus:FILTer1 FALL;:STATus:EESSE 1;
EESR?;*SRE 8;:TRIGger:MODE SINGLE;:
START<PMT>
(STATus:EESR? の応答を読む)
(サービスリクエストの発生を待つ)
:WAVEform:SEND?<PMT>
```

「STATus:FILTer1 FALL」は、状態レジスタのビット 0 が「1」から「0」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:EESSE 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 8」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。

「WAVEform: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 FALL;:STATus:EESR?;:
TRIGger:MODE SINGLE<PMT>
(STATus:EESR? の応答を読む)
:COMMunicate:WAIT 1;:WAVEform:SEND?
<PMT>
```

「STATus:FILTer1 FALL」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。

「WAVEform: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。

## 5.1 コマンド一覧表

コマンド	機能	ページ
<b>ACQUIRE グループ</b>		
:ACQUIRE?	波形の取り込みに関するすべての設定値を問い合わせます。	5-42
:ACQUIRE:AVERage?	アベレーシングおよび波形の取り込み回数に関するすべての設定値を問い合わせます。	5-42
:ACQUIRE:AVERage:COUNT	アベレーシングモード時の波形取り込み回数を設定 / 問い合わせします。	5-42
:ACQUIRE:AVERage:EWEight	アベレーシングモードを無限大行うときの減衰定数を設定 / 問い合わせします。	5-42
:ACQUIRE:HRMode	高分解能モードの ON/OFF を設定 / 問い合わせします。	5-42
:ACQUIRE:INTERLeave	インターリーブの ON/OFF を設定 / 問い合わせします。	5-42
:ACQUIRE:INTERPolate	データ補間の ON/OFF を設定 / 問い合わせします。	5-42
:ACQUIRE:MODE	波形取り込みモードを設定 / 問い合わせします。	5-42
:ACQUIRE:REPetitive	等価時間サンプリングの ON/OFF を設定 / 問い合わせします。	5-42
:ACQUIRE:RLENgth	レコード長を設定 / 問い合わせします。	5-42
<b>ANALYSIS グループ</b>		
:ANALYSIS?	解析機能に関するすべての設定値を問い合わせます。	5-43
:ANALYSIS:AHIStogram<x>?	アキュムレートヒストグラム機能に関するすべての設定値を問い合わせます。	5-43
:ANALYSIS:AHIStogram<x>:HORizontal	アキュムレートヒストグラムの水平軸方向範囲を設定 / 問い合わせします。	5-43
:ANALYSIS:AHIStogram<x>:MEASure?	アキュムレートヒストグラムの自動測定に関するすべての設定値を問い合わせます。	5-44
:ANALYSIS:AHIStogram<x>:MEASure:CURSor?	アキュムレートヒストグラムのカーソル測定に関するすべての設定値を問い合わせます。	5-44
:ANALYSIS:AHIStogram<x>:MEASure:CURSor:BASic?	アキュムレートヒストグラムのカーソルの Basic アイテムに関するすべての設定値を問い合わせます。	5-44
:ANALYSIS:AHIStogram<x>:MEASure:CURSor[:BASic]:ALL	アキュムレートヒストグラムのカーソルの Basic アイテムすべてを ON/OFF します。	5-44
:ANALYSIS:AHIStogram<x>:MEASure:CURSor[:BASic]:C<x>?	アキュムレートヒストグラムの各カーソルに関するすべての設定値を問い合わせます。	5-45
:ANALYSIS:AHIStogram<x>:MEASure:CURSor[:BASic]:C<x>:STATE	アキュムレートヒストグラムの各カーソルの ON/OFF を設定 / 問い合わせします。	5-45
:ANALYSIS:AHIStogram<x>:MEASure:CURSor[:BASic]:C<x>:VALue?	アキュムレートヒストグラムの各カーソルの測定値を問い合わせます。	5-45
:ANALYSIS:AHIStogram<x>:MEASure:CURSor[:BASic]:DC?	アキュムレートヒストグラムのカーソル間の測定値に関するすべての設定値を問い合わせます。	5-45
:ANALYSIS:AHIStogram<x>:MEASure:CURSor[:BASic]:DC:STATE	アキュムレートヒストグラムのカーソル間の測定値の ON/OFF を設定 / 問い合わせします。	5-45
:ANALYSIS:AHIStogram<x>:MEASure:CURSor[:BASic]:DC:VALue?	アキュムレートヒストグラムのカーソル間の測定値を問い合わせます。	5-45
:ANALYSIS:AHIStogram<x>:MEASure:CURSor:CALCulation?	アキュムレートヒストグラムのカーソルの Calc アイテムに関するすべての設定値を問い合わせます。	5-45
:ANALYSIS:AHIStogram<x>:MEASure:CURSor:CALCulation:ALL	アキュムレートヒストグラムのカーソルの Calc アイテムすべてを ON/OFF します。	5-45
:ANALYSIS:AHIStogram<x>:MEASure:CURSor:CALCulation:DEFine<x>	アキュムレートヒストグラムのカーソルの Calc アイテムの演算式を設定 / 問い合わせします。	5-46
:ANALYSIS:AHIStogram<x>:MEASure:CURSor:CALCulation:STATE<x>	アキュムレートヒストグラムのカーソルの Calc アイテムの ON/OFF を設定 / 問い合わせします。	5-46
:ANALYSIS:AHIStogram<x>:MEASure:CURSor:CALCulation:VALue<x>?	アキュムレートヒストグラムのカーソルの Calc アイテムの測定値を問い合わせます。	5-46
:ANALYSIS:AHIStogram<x>:MEASure:CURSor:HLINKage	アキュムレートヒストグラムの水平カーソルのリンクの ON/OFF を設定 / 問い合わせします。	5-46

## 5.1 コマンド一覧表

コマンド	機能	ページ
:ANALysis:AHIStoqram<x>: MEASure:CURSor:HPOsition<x>	アキュムレートヒストグラムの各水平カーソルの位置を設定 / 問い合わせし ます。	5-46
:ANALysis:AHIStoqram<x>: MEASure:CURSor:VLINkage	アキュムレートヒストグラムの垂直カーソルのリンクの ON/OFF を設定 / 問 い合わせします。	5-46
:ANALysis:AHIStoqram<x>: MEASure:CURSor:VPOsition<x>	アキュムレートヒストグラムの各垂直カーソルの位置を設定 / 問い合わせし ます。	5-47
:ANALysis:AHIStoqram<x>: MEASure:MODE	アキュムレートヒストグラムの自動測定モードを設定 / 問い合わせしま す。	5-47
:ANALysis:AHIStoqram<x>: MEASure:PARAmeter?	アキュムレートヒストグラムの波形パラメータの自動測定に関するすべての 設定値を問い合わせます。	5-47
:ANALysis:AHIStoqram<x>: MEASure:PARAmeter:AREA<x>?	アキュムレートヒストグラムの各エリアに関するすべての設定値を問 い合わせます。	5-48
:ANALysis:AHIStoqram<x>: MEASure:PARAmeter:AREA<x>:ALL	アキュムレートヒストグラムのすべての波形パラメータを一斉に ON/OFF します。	5-48
:ANALysis:AHIStoqram<x>: MEASure:PARAmeter:AREA<x>:<パラ メータ>?	アキュムレートヒストグラムの波形パラメータに関するすべての設定値を問 い合わせます。	5-48
:ANALysis:AHIStoqram<x>: MEASure:PARAmeter:AREA<x>:<パラ メータ>:STATe	アキュムレートヒストグラムの波形パラメータの ON/OFF を設定 / 問 い合わせします。	5-48
:ANALysis:AHIStoqram<x>: MEASure:PARAmeter:AREA<x>:<パラ メータ>:VALue?	アキュムレートヒストグラムの波形パラメータの自動測定値を問 い合わせます。	5-48
:ANALysis:AHIStoqram<x>: MEASure:PARAmeter:CALCulation?	アキュムレートヒストグラムの波形パラメータの Calc アイテムに関するす べての設定値を問い合わせます。	5-49
:ANALysis:AHIStoqram<x>: MEASure:PARAmeter:CALCulation: ALL	アキュムレートヒストグラムの波形パラメータのすべての Calc アイテムを 一斉に ON/OFF します。	5-49
:ANALysis:AHIStoqram<x>: MEASure:PARAmeter:CALCulation: DEFine<x>	アキュムレートヒストグラムの波形パラメータの Calc アイテムの演算式を 設定 / 問い合わせします。	5-49
:ANALysis:AHIStoqram<x>: MEASure:PARAmeter:CALCulation: STATe<x>	アキュムレートヒストグラムの波形パラメータの Calc アイテムの ON/OFF を設定 / 問い合わせします。	5-49
:ANALysis:AHIStoqram<x>: MEASure:PARAmeter:CALCulation: VALue<x>?	アキュムレートヒストグラムの波形パラメータの Calc アイテムの自動測定 値を問い合わせます。	5-49
:ANALysis:AHIStoqram<x>: MEASure:PARAmeter:HRANge<x>	アキュムレートヒストグラムの波形パラメータの水平範囲を設定 / 問 い合わせします。	5-49
:ANALysis:AHIStoqram<x>: MEASure:PARAmeter:VRANge<x>	アキュムレートヒストグラムの波形パラメータの垂直範囲を設定 / 問 い合わせします。	5-50
:ANALysis:AHIStoqram<x>:MODE	アキュムレートヒストグラムのモードを設定 / 問い合わせします。	5-50
:ANALysis:AHIStoqram<x>:TRACe	アキュムレートヒストグラムの対象トレースを設定 / 問い合わせします。	5-50
:ANALysis:AHIStoqram<x>: VERTical	アキュムレートヒストグラムの垂直範囲を設定 / 問い合わせします。	5-50
:ANALysis:AHIStoqram<x>:WINDow	アキュムレートヒストグラムの測定対象ウィンドウを設定 / 問 い合わせします。	5-50
:ANALysis:DISPlay<x>	解析機能の表示の ON/OFF を設定 / 問い合わせします。	5-50
:ANALysis:FFT<x>?	FFT 演算機能に関するすべての設定値を問い合わせます。	5-51
:ANALysis:FFT<x>:HORizontal?	FFT 演算の水平軸に関するすべての設定値を問い合わせます。	5-51
:ANALysis:FFT<x>:HORizontal: CSPan?	FFT 演算の水平軸の中心値 / スパンに関するすべての設定値を問 い合わせます。	5-51
:ANALysis:FFT<x>:HORizontal: CSPan:CENTer	FFT 演算の水平軸の中心値を設定 / 問い合わせします。	5-51
:ANALysis:FFT<x>:HORizontal: CSPan:SPAN	FFT 演算の水平軸のスパンを設定 / 問い合わせします。	5-51
:ANALysis:FFT<x>:HORizontal: LRIGHt?	FFT 演算の水平軸の左右端に関するすべての設定値を問 い合わせます。	5-51
:ANALysis:FFT<x>:HORizontal: LRIGHt:RANGe	FFT 演算の水平軸の左右端の範囲を設定 / 問い合わせします。	5-52
:ANALysis:FFT<x>:HORizontal: MODE	FFT 演算の水平軸のモードを設定 / 問い合わせします。	5-52
:ANALysis:FFT<x>:IPARt (Imag Part)	FFT 演算のイマジナリーパート対象トレースを設定 / 問 い合わせします。	5-52
:ANALysis:FFT<x>:LENGth	FFT 演算の FFT 点数を設定 / 問い合わせします。	5-52

コマンド	機能	ページ
:ANALysis:FFT<x>:MAXHold	FFT 演算の最大値保持機能の ON/OFF を設定 / 問い合わせします。	5-52
:ANALysis:FFT<x>:MEASure?	FFT 演算の自動測定に関するすべての設定値を問い合わせます。	5-52
:ANALysis:FFT<x>:MEASure:MARKer?	FFT 演算のマーカーカーソル測定に関するすべての設定値を問い合わせます。	5-53
:ANALysis:FFT<x>:MEASure:MARKer:BASic?	FFT 演算のマーカーカーソルの Basic アイテムに関するすべての設定値を問い合わせます。	5-53
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:ALL	FFT 演算のマーカーカーソルのすべての Basic アイテムを一斉に ON/OFF します。	5-53
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:DFRequency?	FFT 演算のマーカーカーソル間の周波数値に関するすべての設定値を問い合わせます。	5-53
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:DFRequency:STATe	FFT 演算のマーカーカーソル間の周波数値の ON/OFF を設定 / 問い合わせします。	5-53
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:DFRequency:VALue?	FFT 演算のマーカーカーソル間の周波数値を問い合わせます。	5-53
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:DV?	FFT 演算のマーカーカーソル間のパワー値に関するすべての設定値を問い合わせます。	5-53
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:DV:STATe	FFT 演算のマーカーカーソル間のパワー値の ON/OFF を設定 / 問い合わせします。	5-54
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:DV:VALue?	FFT 演算のマーカーカーソル間のパワー値を問い合わせます。	5-54
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:FREquency<x>?	FFT 演算の各マーカーカーソルの周波数値に関するすべての設定値を問い合わせます。	5-54
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:FREquency<x>:STATe	FFT 演算の各マーカーカーソルの周波数値の ON/OFF を設定 / 問い合わせします。	5-54
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:FREquency<x>:VALue?	FFT 演算の各マーカーカーソルの周波数値を問い合わせます。	5-54
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:POSition<x>{<Nrf>}	FFT 演算の各マーカーカーソルの位置を設定 / 問い合わせします。	5-54
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:V<x>?	FFT 演算の各マーカーカーソルのパワー値に関するすべての設定値を問い合わせます。	5-54
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:V<x>:STATe	FFT 演算の各マーカーカーソルのパワー値の ON/OFF を設定 / 問い合わせします。	5-55
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:V<x>:VALue?	FFT 演算の各マーカーカーソルのパワー値を問い合わせます。	5-55
:ANALysis:FFT<x>:MEASure:MARKer:CALCulation?	FFT 演算のマーカーカーソルの Calc アイテムに関するすべての設定値を問い合わせます。	5-55
:ANALysis:FFT<x>:MEASure:MARKer:CALCulation:ALL	FFT 演算のマーカーカーソルのすべての Calc アイテムを一斉に ON/OFF します。	5-55
:ANALysis:FFT<x>:MEASure:MARKer:CALCulation:DEFine<x>	FFT 演算のマーカーカーソルの Calc アイテムの演算式を設定 / 問い合わせします。	5-55
:ANALysis:FFT<x>:MEASure:MARKer:CALCulation:STATe<x>	FFT 演算のマーカーカーソルの Calc アイテムの ON/OFF を設定 / 問い合わせします。	5-55
:ANALysis:FFT<x>:MEASure:MARKer:CALCulation:VALue<x>?	FFT 演算のマーカーカーソルの Calc アイテムの測定値を問い合わせます。	5-55
:ANALysis:FFT<x>:MEASure:MODE	FFT 演算の自動測定モードを設定 / 問い合わせします。	5-56
:ANALysis:FFT<x>:MEASure:PEAK?	FFT 演算のピーク値測定に関するすべての設定値を問い合わせます。	5-56
:ANALysis:FFT<x>:MEASure:PEAK:BASic?	FFT 演算のピーク値の Basic アイテムに関するすべての設定値を問い合わせます。	5-56
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:ALL	FFT 演算のピーク値のすべての Basic アイテムを一斉に ON/OFF します。	5-56
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:DFRequency?	FFT 演算のピーク値間の周波数値に関するすべての設定値を問い合わせます。	5-56
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:DFRequency:STATe	FFT 演算のピーク値間の周波数値の ON/OFF を設定 / 問い合わせします。	5-56
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:DFRequency:VALue?	FFT 演算のピーク値間の周波数値を問い合わせます。	5-56
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:DV?	FFT 演算のピーク値間のパワー値に関するすべての設定値を問い合わせます。	5-57
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:DV:STATe	FFT 演算のピーク値間のパワー値の ON/OFF を設定 / 問い合わせします。	5-57

## 5.1 コマンド一覧表

コマンド	機能	ページ
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:DV:VALue?	FFT 演算のピーク値間のパワー値を問い合わせます。	5-57
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:FREQuency<x>?	FFT 演算の各ピーク周波数値に関するすべての設定値を問い合わせます。	5-57
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:FREQuency<x>:STATe	FFT 演算の各ピーク周波数値の ON/OFF を設定 / 問い合わせします。	5-57
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:FREQuency<x>:VALue?	FFT 演算の各ピーク周波数値を問い合わせます。	5-57
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:RANge<x>	FFT 演算のピーク値の各測定範囲を設定 / 問い合わせします。	5-57
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:V<x>?	FFT 演算の各ピーク値に関するすべての設定値を問い合わせます。	5-58
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:V<x>:STATe	FFT 演算の各ピーク値の ON/OFF を設定 / 問い合わせします。	5-58
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:V<x>:VALue?	FFT 演算の各ピーク値を問い合わせます。	5-58
:ANALysis:FFT<x>:MEASure:PEAK:CALCulation?	FFT 演算の Calc アイテムに関するすべての設定値を問い合わせます。	5-58
:ANALysis:FFT<x>:MEASure:PEAK:CALCulation:ALL	FFT 演算のすべての Calc アイテムを一斉に ON/OFF します。	5-58
:ANALysis:FFT<x>:MEASure:PEAK:CALCulation:DEFine<x>	FFT 演算の Calc アイテムの演算式を設定 / 問い合わせします。	5-58
:ANALysis:FFT<x>:MEASure:PEAK:CALCulation:STATe<x>	FFT 演算の Calc アイテムの ON/OFF を設定 / 問い合わせします。	5-58
:ANALysis:FFT<x>:MEASure:PEAK:CALCulation:VALue<x>?	FFT 演算の Calc アイテムの測定値を問い合わせます。	5-59
:ANALysis:FFT<x>:RANge	FFT 演算の測定対象ウィンドウを設定 / 問い合わせします。	5-59
:ANALysis:FFT<x>:RPART (Real Part)	FFT 演算のリアルパート対象トレースを設定 / 問い合わせします。	5-59
:ANALysis:FFT<x>:RPOSITion (Ref Position)	FFT 演算の垂直軸の拡大中心点を設定 / 問い合わせします。	5-59
:ANALysis:FFT<x>:VERTical?	FFT 演算の垂直軸に関するすべての設定値を問い合わせます。	5-59
:ANALysis:FFT<x>:VERTical:LEVel	FFT 演算の垂直軸の表示位置を設定 / 問い合わせします。	5-59
:ANALysis:FFT<x>:VERTical:MODE	FFT 演算の垂直軸のモードを設定 / 問い合わせします。	5-59
:ANALysis:FFT<x>:VERTical:SENSitivity	FFT 演算の垂直軸感度を設定 / 問い合わせします。	5-59
:ANALysis:FFT<x>:WINDow	FFT 演算の窓関数を設定 / 問い合わせします。	5-59
:ANALysis:TYPE<x>	解析機能のタイプを設定 / 問い合わせします。	5-59
:ANALysis:VTDisplay	VT 波形の表示の ON/OFF を設定 / 問い合わせします。	5-60
:ANALysis:WAIT<x>?	タイムアウト付きで自動測定実行の終了を待ちます。	5-60
:ANALysis:WPARameter<x>?	波形パラメータ測定機能に関するすべての設定値を問い合わせます。	5-60
:ANALysis:WPARameter<x>:BIT<x>?	ロジック波形パラメータ測定の各ロジックビットに関するすべての設定値を 5-60 問い合わせます。	5-60
:ANALysis:WPARameter<x>:BIT<x>:AREA<x>?	ロジック波形パラメータ測定の各エリアに関するすべての設定値を問い合 5-61 わせします。	5-61
:ANALysis:WPARameter<x>:BIT<x>:AREA<x>:TYPE	ロジック波形パラメータ測定のロジック波形パラメータを設定 / 問い合 5-61 わせします。	5-61
:ANALysis:WPARameter<x>:CALCulation	波形パラメータ測定の Calc アイテムを設定 / 問い合わせします。	5-61
:ANALysis:WPARameter<x>:HISTogram?	波形パラメータ測定のヒストグラム表示に関するすべての設定値を問い合 5-61 わせします。	5-61
:ANALysis:WPARameter<x>:HISTogram:MEASure?	波形パラメータ測定のヒストグラム表示の自動測定に関するすべての設定値 5-61 を問い合わせます。	5-61
:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor?	波形パラメータ測定のヒストグラムのカーソル測定に関するすべての設定値 5-62 を問い合わせます。	5-62
:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:ALL	波形パラメータ測定のすべてのヒストグラムのカーソルを一斉に ON/OFF 5-62 します。	5-62
:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:C<x>?	波形パラメータ測定のヒストグラムの各カーソル測定に関するすべての設定 5-62 値を問い合わせます。	5-62
:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:C<x>:POSITION	波形パラメータ測定のヒストグラムの各カーソルの位置を設定 / 問い合 5-62 わせします。	5-62

コマンド	機能	ページ
:ANALysis:WPARAMeter<x>: HISTogram:MEASure:CURSor:C<x>: STATE	波形パラメータ測定ヒストグラムの各カーソルの ON/OFF を設定 / 問い合わせします。	5-62
:ANALysis:WPARAMeter<x>: HISTogram:MEASure:CURSor:C<x>: VALue?	波形パラメータ測定ヒストグラムの各カーソルの測定値を問い合わせます。	5-62
:ANALysis:WPARAMeter<x>: HISTogram:MEASure:CURSor:DC?	波形パラメータ測定ヒストグラムのカーソル間の測定に関するすべての設定値を問い合わせます。	5-63
:ANALysis:WPARAMeter<x>: HISTogram:MEASure:CURSor:DC: STATE	波形パラメータ測定ヒストグラムのカーソル間の ON/OFF を設定 / 問い合わせします。	5-63
:ANALysis:WPARAMeter<x>: HISTogram:MEASure:CURSor:DC: VALue?	波形パラメータ測定ヒストグラムのカーソル間の測定値を問い合わせます。	5-63
:ANALysis:WPARAMeter<x>: HISTogram:MEASure:CURSor: LINKage	波形パラメータ測定ヒストグラムのカーソルのリンクの ON/OFF を設定 / 問い合わせします。	5-63
:ANALysis:WPARAMeter<x>: HISTogram:MEASure:MODE	波形パラメータ測定ヒストグラム表示の自動測定モードを設定 / 問い合わせします。	5-63
:ANALysis:WPARAMeter<x>: HISTogram:MEASure:PARAMeter?	波形パラメータ測定ヒストグラムパラメータの自動測定に関するすべての設定値を問い合わせます。	5-63
:ANALysis:WPARAMeter<x>: HISTogram:MEASure:PARAMeter:ALL	波形パラメータ測定ヒストグラムパラメータのすべてのヒストグラムパラメータを一斉に ON/OFF します。	5-63
:ANALysis:WPARAMeter<x>: HISTogram:MEASure:PARAMeter:<パラメータ>?	波形パラメータ測定ヒストグラムパラメータに関するすべての設定値を問い合わせます。	5-64
:ANALysis:WPARAMeter<x>: HISTogram:MEASure:PARAMeter:<パラメータ>:STATE	波形パラメータ測定ヒストグラムパラメータの ON/OFF を設定 / 問い合わせします。	5-64
:ANALysis:WPARAMeter<x>: HISTogram:MEASure:PARAMeter:<パラメータ>:VALue?	波形パラメータ測定ヒストグラムパラメータの測定値を問い合わせます。	5-64
:ANALysis:WPARAMeter<x>:LIST?	波形パラメータ測定ヒストグラム表示に関するすべての設定値を問い合わせます。	5-64
:ANALysis:WPARAMeter<x>:LIST: ITEM?	波形パラメータ測定ヒストグラム表示のアイテムを問い合わせます。	5-64
:ANALysis:WPARAMeter<x>:LIST: MODE	波形パラメータ測定ヒストグラム表示のモードを設定 / 問い合わせします。	5-64
:ANALysis:WPARAMeter<x>:LIST: SCRoll	波形パラメータ測定ヒストグラム表示のスクロール方向を設定 / 問い合わせします。	5-64
:ANALysis:WPARAMeter<x>:LIST: VALue?	波形パラメータ測定ヒストグラム表示番号の自動測定値を問い合わせます。	5-65
:ANALysis:WPARAMeter<x>:MODE	波形パラメータ測定モードを設定 / 問い合わせします。	5-65
:ANALysis:WPARAMeter<x>: TRACe<x>?	波形パラメータ測定各トレースに関するすべての設定値を問い合わせます。	5-65
:ANALysis:WPARAMeter<x>: TRACe<x>:AREA<x>?	波形パラメータ測定各エリアに関するすべての設定値を問い合わせます。	5-65
:ANALysis:WPARAMeter<x>: TRACe<x>:AREA<x>:TYPE	波形パラメータ測定波形パラメータを設定 / 問い合わせします。	5-65
:ANALysis:WPARAMeter<x>:TRENd?	波形パラメータ測定トレンド表示に関するすべての設定値を問い合わせます。	5-65
:ANALysis:WPARAMeter<x>:TRENd: ASCale[:EXECute]	波形パラメータ測定トレンド表示のオートスケールを実行します。	5-65
:ANALysis:WPARAMeter<x>:TRENd: HRANge	波形パラメータ測定トレンド表示の対象ウィンドウを設定 / 問い合わせします。	5-66
:ANALysis:WPARAMeter<x>:TRENd: HSPan	波形パラメータ測定トレンド表示の水平スパンを設定 / 問い合わせします。	5-66
:ANALysis:WPARAMeter<x>:TRENd: MEASure?	波形パラメータ測定トレンド表示の自動測定に関するすべての設定値を問い合わせます。	5-66
:ANALysis:WPARAMeter<x>:TRENd: MEASure:CURSor?	波形パラメータ測定トレンド表示のカーソル測定に関するすべての設定値を問い合わせます。	5-66
:ANALysis:WPARAMeter<x>:TRENd: MEASure:CURSor:ALL	波形パラメータ測定トレンドのすべてのカーソルを ON/OFF します。	5-66
:ANALysis:WPARAMeter<x>:TRENd: MEASure:CURSor:C<x>?	波形パラメータ測定トレンドの各カーソル測定に関するすべての設定値を問い合わせます。	5-66



## 5.1 コマンド一覧表

コマンド	機能	ページ
:ANALysis:WPARAMeter<x>:TREND:MEASure:CURSor:C<x>:POSITION	波形パラメータ測定のトレンドの各カーソルの位置を設定 / 問い合わせします。	5-66
:ANALysis:WPARAMeter<x>:TREND:MEASure:CURSor:C<x>:STATE	波形パラメータ測定のトレンドの各カーソルの ON/OFF を設定 / 問い合わせします。	5-67
:ANALysis:WPARAMeter<x>:TREND:MEASure:CURSor:C<x>:VALue?	波形パラメータ測定のトレンドの各カーソルの測定値を問い合わせます。	5-67
:ANALysis:WPARAMeter<x>:TREND:MEASure:CURSor:DC?	波形パラメータ測定のトレンドのカーソル間の測定に関するすべての設定値を問い合わせます。	5-67
:ANALysis:WPARAMeter<x>:TREND:MEASure:CURSor:DC:STATE	波形パラメータ測定のトレンドのカーソル間の ON/OFF を設定 / 問い合わせします。	5-67
:ANALysis:WPARAMeter<x>:TREND:MEASure:CURSor:DC:VALue?	波形パラメータ測定のトレンドのカーソル間の測定値を問い合わせます。	5-67
:ANALysis:WPARAMeter<x>:TREND:MEASure:CURSor:LINKage	波形パラメータ測定のトレンドのカーソルのリンクの ON/OFF を設定 / 問い合わせします。	5-67
:ANALysis:WPARAMeter<x>:TREND:MEASure:MODE	波形パラメータ測定のトレンドの自動測定モードを設定 / 問い合わせします。	5-68
:ANALysis:WPARAMeter<x>:TREND:VERTical	波形パラメータ測定のトレンドの垂直レンジを設定 / 問い合わせします。	5-68
:ANALysis:XY<x>?	XY 表示機能に関するすべての設定値を問い合わせます。	5-68
:ANALysis:XY<x>:GATE?	XY 表示のゲート機能に関するすべての設定値を問い合わせます。	5-68
:ANALysis:XY<x>:GATE:ALEVel	XY 表示のゲートのアクティブレベルを設定 / 問い合わせします。	5-69
:ANALysis:XY<x>:GATE:HYSteresis<x>	XY 表示のゲートのヒステリシスを設定 / 問い合わせします。	5-69
:ANALysis:XY<x>:GATE:LEVel<x>	XY 表示のゲートのレベルを設定 / 問い合わせします。	5-69
:ANALysis:XY<x>:GATE:TRAcE	XY 表示のゲートトレースを設定 / 問い合わせします。	5-69
:ANALysis:XY<x>:MEASure?	XY 表示の自動測定に関するすべての設定値を問い合わせます。	5-69
:ANALysis:XY<x>:MEASure:CURSor?	XY 表示のカーソル測定に関するすべての設定値を問い合わせます。	5-69
:ANALysis:XY<x>:MEASure:CURSor:XLINKage	XY 表示の水平カーソルのリンクの ON/OFF を設定 / 問い合わせします。	5-69
:ANALysis:XY<x>:MEASure:CURSor:X<x>?	XY 表示の水平カーソルに関するすべての設定値を問い合わせます。	5-69
:ANALysis:XY<x>:MEASure:CURSor:X<x>:POSITION	XY 表示の水平カーソルの位置を設定 / 問い合わせします。	5-70
:ANALysis:XY<x>:MEASure:CURSor:X<x>:VALue?	XY 表示の水平カーソルの電圧値を問い合わせます。	5-70
:ANALysis:XY<x>:MEASure:CURSor:YLINKage	XY 表示の垂直カーソルのリンクの ON/OFF を設定 / 問い合わせします。	5-70
:ANALysis:XY<x>:MEASure:CURSor:Y<x>?	XY 表示の垂直カーソルに関するすべての設定値を問い合わせます。	5-70
:ANALysis:XY<x>:MEASure:CURSor:Y<x>:POSITION	XY 表示の垂直カーソルの位置を設定 / 問い合わせします。	5-70
:ANALysis:XY<x>:MEASure:CURSor:Y<x>:VALue?	XY 表示の垂直カーソルの電圧値を問い合わせます。	5-70
:ANALysis:XY<x>:MEASure:MODE	XY 表示の自動測定モードを設定 / 問い合わせします。	5-70
:ANALysis:XY<x>:MEASure:XYINteg?	XY 表示の積分に関するすべての設定値を問い合わせます。	5-70
:ANALysis:XY<x>:MEASure:XYINteg:LOOP	XY 表示の積分方法を設定 / 問い合わせします。	5-71
:ANALysis:XY<x>:MEASure:XYINteg:POLarity	XY 表示の積分方向を設定 / 問い合わせします。	5-71
:ANALysis:XY<x>:MEASure:XYINteg:VALue?	XY 表示の積分値を問い合わせします。	5-71
:ANALysis:XY<x>:TRANge (Time Range)	XY 表示の測定範囲を設定 / 問い合わせします。	5-71
:ANALysis:XY<x>:WINdow	XY 表示の測定対象ウィンドウを設定 / 問い合わせします。	5-71
:ANALysis:XY<x>:XTRace	XY 表示の X 軸トレースを設定 / 問い合わせします。	5-71
:ANALysis:XY<x>:YTRace	XY 表示の Y 軸トレースを設定 / 問い合わせします。	5-71
<b>ASETup グループ</b>		
:ASETup:EXECute	オートセットアップを実行します。	5-72
:ASETup:UNDO	実行したオートセットアップを取り消します。	5-72

コマンド	機能	ページ
<b>CALibrate グループ</b>		
:CALibrate?	キャリブレーションに関するすべての設定値を問い合わせます。	5-73
:CALibrate:EXECute	キャリブレーションを実行します。	5-73
:CALibrate:MODE	オートキャリブレーションの ON/OFF を設定 / 問い合わせします。	5-73
<b>CHANnel グループ</b>		
:CHANnel<x>?	各チャンネルに関するすべての設定値を問い合わせます。	5-74
:CHANnel<x>:ASCAle[:EXECute]	各チャンネルのオートスケールを実行します。	5-74
:CHANnel<x>:BWIDth	各チャンネルの入力フィルタを設定 / 問い合わせします。	5-74
:CHANnel<x>:COUPling	各チャンネルの入力カップリングを設定 / 問い合わせします。	5-74
:CHANnel<x>:DESKew	各チャンネルのスキュー補正を設定 / 問い合わせします。	5-74
:CHANnel<x>:DISPlay	各チャンネルの表示の ON/OFF を設定 / 問い合わせします。	5-74
:CHANnel<x>:INVert	各チャンネルの反転表示の ON/OFF を設定 / 問い合わせします。	5-74
:CHANnel<x>:LABel?	各チャンネルの波形ラベル名に関するすべての設定値を問い合わせます。	5-74
:CHANnel<x>:LABel[:DEFine]	各チャンネルの波形ラベル名を設定 / 問い合わせします。	5-74
:CHANnel<x>:LABel:MODE	各チャンネルの波形ラベル名表示の ON/OFF を設定 / 問い合わせします。	5-74
:CHANnel<x>:OCANcel	各チャンネルのオフセットキャンセルの ON/OFF を設定 / 問い合わせします。	5-75
:CHANnel<x>:OFFSet	各チャンネルのオフセット電圧を設定 / 問い合わせします。	5-75
:CHANnel<x>:POSition	各チャンネルの垂直ポジションを設定 / 問い合わせします。	5-75
:CHANnel<x>:PROBe?	各チャンネルのプロープ減衰比に関するすべての設定値を問い合わせます。	5-75
:CHANnel<x>:PROBe[:MODE]	各チャンネルのプロープ減衰比を設定 / 問い合わせします。	5-75
:CHANnel<x>:PROBe:AUTO?	各チャンネルの AUTO 時のプロープ減衰比を問い合わせます。	5-75
:CHANnel<x>:PZCalibrate[:EXECute]	各チャンネルの電流プローブのゼロ補正を実行します。	5-75
:CHANnel<x>:SElect	各入力チャンネルに割り当てる波形 (入力 / 演算) を設定 / 問い合わせします。	5-75
:CHANnel<x>:SVALue (Scale VALUE)	各チャンネルのスケール値表示の ON/OFF を設定 / 問い合わせします。	5-75
:CHANnel<x>:VDIV	各チャンネルの垂直軸感度 (V/div) を設定 / 問い合わせします。	5-75
<b>CLEar グループ</b>		
:CLEar:ACCumulate	アキュムレート波形をクリアします。	5-76
:CLEar[:HISTory]	ヒストリ波形をクリアします。	5-76
:CLEar:SNAP	スナップショット波形をクリアします。	5-76
<b>COMMunicate グループ</b>		
:COMMunicate?	通信に関するすべての設定値を問い合わせます。	5-77
:COMMunicate:HEADer	クエリに対する応答を、ヘッダを付けて返送するか (例 CHANNEL1: VOLTAGE:PROBE 10)、付けずに返送するか (例 10) を設定 / 問い合わせします。	5-77
:COMMunicate:LOCKout	ローカルロックアウトを設定 / 解除します。	5-77
:COMMunicate:OPSE (Operation Pending Status Enable register)	*OPC、*OPC?、*WAI の対象となるオーバーラップコマンドを設定 / 問い合わせします。	5-77
:COMMunicate:OPSR? (Operation Pending Status Register)	オペレーションペンディングステータスレジスタの値を問い合わせます。	5-77
:COMMunicate:OVERlap	オーバーラップ動作にするコマンドを設定 / 問い合わせします。	5-77
:COMMunicate:REMOte	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。	5-78
:COMMunicate:STATus?	回線固有のステータスを問い合わせます。	5-78
:COMMunicate:VERBose	クエリに対する応答を、フルスペルで返送するか (例 CHANNEL1:VOLTAGE:PROBE 10)、省略形で返送するか (例 CHAN:PROB 10) を設定 / 問い合わせします。	5-78
:COMMunicate:WAIT	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。	5-78
:COMMunicate:WAIT?	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。	5-78
<b>CURSor グループ</b>		
:CURSor?	カーソル測定に関するすべての設定値を問い合わせます。	5-79
:CURSor:DISPlay	カーソルの ON/OFF を設定 / 問い合わせします。	5-79
:CURSor:HORizontal?	水平カーソルに関するすべての設定値を問い合わせます。	5-79
:CURSor:HORizontal:BASic?	水平カーソルの Basic アイテムに関するすべての設定値を問い合わせます。	5-79
:CURSor:HORizontal[:BASic]:ALL	水平カーソルの Basic アイテムすべてを ON/OFF します。	5-79
:CURSor:HORizontal[:BASic]:DV?	水平カーソルの $\Delta V$ 測定に関するすべての設定値を問い合わせます。	5-79

## 5.1 コマンド一覧表

コマンド	機能	ページ
:CURSor:HORizontal[:BASic]:DV:STATE	水平カーソルの $\Delta V$ 測定の ON/OFF を設定 / 問い合わせします。	5-79
:CURSor:HORizontal[:BASic]:DV:VALue?	水平カーソル間の電圧値を問い合わせます。	5-79
:CURSor:HORizontal[:BASic]:LINKage	水平カーソルのリンクの ON/OFF を設定 / 問い合わせします。	5-79
:CURSor:HORizontal[:BASic]:V<x>?	各水平カーソルに関するすべての設定値を問い合わせます。	5-80
:CURSor:HORizontal[:BASic]:V<x>:JUMP	各水平カーソルを指定したズーム波形の中央位置にジャンプします。	5-80
:CURSor:HORizontal[:BASic]:V<x>:POSition	各水平カーソルの位置を設定 / 問い合わせします。	5-80
:CURSor:HORizontal[:BASic]:V<x>:STATE	各水平カーソルの ON/OFF を設定 / 問い合わせします。	5-80
:CURSor:HORizontal[:BASic]:V<x>:VALue?	各水平カーソルの電圧値を問い合わせます。	5-80
:CURSor:HORizontal:CALCulation?	水平カーソルの Calc アイテムに関するすべての設定値を問い合わせます。	5-80
:CURSor:HORizontal:CALCulation:ALL	水平カーソルの Calc アイテムすべてを ON/OFF します。	5-80
:CURSor:HORizontal:CALCulation:DEFine<x>	水平カーソルの Calc アイテムの演算式を設定 / 問い合わせします。	5-80
:CURSor:HORizontal:CALCulation:STATE<x>	水平カーソルの Calc アイテムの ON/OFF を設定 / 問い合わせします。	5-80
:CURSor:HORizontal:CALCulation:VALue<x>?	水平カーソルの Calc アイテムの測定値を問い合わせます。	5-81
:CURSor:HORizontal:TRACe	水平カーソルの対象トレースを設定 / 問い合わせします。	5-81
:CURSor:MARKer?	マーカーカーソルに関するすべての設定値を問い合わせます。	5-81
:CURSor:MARKer:CALCulation?	マーカーカーソルの Calc アイテムに関するすべての設定値を問い合わせます。	5-81
:CURSor:MARKer:CALCulation:ALL	マーカーカーソルの Calc アイテムすべてを ON/OFF します。	5-81
:CURSor:MARKer:CALCulation:DEFine<x>	マーカーカーソルの Calc アイテムの演算式を設定 / 問い合わせします。	5-81
:CURSor:MARKer:CALCulation:STATE<x>	マーカーカーソルの Calc アイテムの ON/OFF を設定 / 問い合わせします。	5-81
:CURSor:MARKer:CALCulation:VALue<x>?	マーカーカーソルの Calc アイテムの測定値を問い合わせます。	5-81
:CURSor:MARKer:CM<x>?	各マーカーカーソルに関するすべての設定値を問い合わせます。	5-82
:CURSor:MARKer:CM<x>:ALL	各マーカーカーソルのすべてのアイテムを ON/OFF します。	5-82
:CURSor:MARKer:CM<x>:DT<x>?	各マーカーカーソルの $\Delta T$ 測定に関するすべての設定値を問い合わせます。	5-82
:CURSor:MARKer:CM<x>:DT<x>:STATE	各マーカーカーソルの $\Delta T$ 測定の ON/OFF を設定 / 問い合わせします。	5-82
:CURSor:MARKer:CM<x>:DT<x>:VALue?	各マーカーカーソルの $\Delta T$ 値を問い合わせます。	5-82
:CURSor:MARKer:CM<x>:DV<x>?	各マーカーカーソルの $\Delta V$ 測定に関するすべての設定値を問い合わせます。	5-82
:CURSor:MARKer:CM<x>:DV<x>:STATE	各マーカーカーソルの $\Delta V$ 測定の ON/OFF を設定 / 問い合わせします。	5-82
:CURSor:MARKer:CM<x>:DV<x>:VALue?	各マーカーカーソルの $\Delta V$ 値を問い合わせます。	5-82
:CURSor:MARKer:CM<x>:JUMP	各マーカーカーソルを指定したズーム波形の中央位置にジャンプします。	5-82
:CURSor:MARKer:CM<x>:POSition	各マーカーカーソルの位置を設定 / 問い合わせします。	5-82
:CURSor:MARKer:CM<x>:T?	各マーカーカーソルの時間測定に関するすべての設定値を問い合わせます。	5-82
:CURSor:MARKer:CM<x>:T:STATE	各マーカーカーソルの時間測定の ON/OFF を設定 / 問い合わせします。	5-83
:CURSor:MARKer:CM<x>:T:VALue?	各マーカーカーソルの時間値を問い合わせます。	5-83
:CURSor:MARKer:CM<x>:TRACe	各マーカーカーソルの対象トレースを設定 / 問い合わせします。	5-83
:CURSor:MARKer:CM<x>:V?	各マーカーカーソルの電圧測定に関するすべての設定値を問い合わせます。	5-83
:CURSor:MARKer:CM<x>:V:STATE	各マーカーカーソルの電圧測定の ON/OFF を設定 / 問い合わせします。	5-83
:CURSor:MARKer:CM<x>:V:VALue?	各マーカーカーソルの電圧値を問い合わせます。	5-83
:CURSor:MARKer:FORM	マーカーカーソルのフォームを設定 / 問い合わせします。	5-83
:CURSor:SERial?	シリアルカーソルに関するすべての設定値を問い合わせます。	5-83
:CURSor:SERial:SCURsor<x>?	各シリアルカーソルに関するすべての設定値を問い合わせます。	5-83
:CURSor:SERial:SCURsor<x>:ACTIVE	各シリアルカーソルのアクティブレベルを設定 / 問い合わせします。	5-83
:CURSor:SERial:SCURsor<x>:BCOunt	各シリアルカーソルのビット長を設定 / 問い合わせします。	5-84

コマンド	機能	ページ
:CURSor:SERIal:SCURSor<x>: BITRate	各シリアルカーソルのビットレートを設定 / 問い合わせします。	5-84
:CURSor:SERIal:SCURSor<x>: BITOrder	各シリアルカーソルのビットオーダーを設定 / 問い合わせします。	5-84
:CURSor:SERIal:SCURSor<x>: FORMat	各シリアルカーソルの表示形式を設定 / 問い合わせします。	5-84
:CURSor:SERIal:SCURSor<x>: HYSTeresis	各シリアルカーソルのヒステリシスを設定 / 問い合わせします。	5-84
:CURSor:SERIal:SCURSor<x>:JUMP	各シリアルカーソルを指定方向に移動します。	5-84
:CURSor:SERIal:SCURSor<x>:LEVel	シリアルカーソルのしきい値 (Threshold) レベルを設定 / 問い合わせします。	5-84
:CURSor:SERIal:SCURSor<x>:MODE	シリアルカーソルの ON/OFF を設定 / 問い合わせします。	5-84
:CURSor:SERIal:SCURSor<x>: POSition	シリアルカーソルの位置を設定 / 問い合わせします。	5-84
:CURSor:SERIal:SCURSor<x>:TRACE	各シリアルカーソルのトレースを設定 / 問い合わせします。	5-85
:CURSor:SERIal:SCURSor<x>:TRACK	各シリアルカーソルを指定したズーム波形上にジャンプします。	5-85
:CURSor:SERIal:SCURSor<x>: VALue?	各シリアルカーソルの測定値を問い合わせます。	5-85
:CURSor:TYPE	カーソルの種類を設定 / 問い合わせします。	5-85
:CURSor:VERTical?	垂直カーソルに関するすべての設定値を問い合わせます。	5-85
:CURSor:VERTical:BASic?	垂直カーソルの Basic アイテムに関するすべての設定値を問い合わせます。	5-85
:CURSor:VERTical[:BASic]:ALL	垂直カーソルの Basic アイテムすべてを ON/OFF します。	5-85
:CURSor:VERTical[:BASic]:DT?	垂直カーソルの $\Delta T$ 測定に関するすべての設定値を問い合わせます。	5-85
:CURSor:VERTical[:BASic]:DT: STATe	垂直カーソルの $\Delta T$ 測定の ON/OFF を設定 / 問い合わせします。	5-85
:CURSor:VERTical[:BASic]:DT: VALue?	垂直カーソルの $\Delta T$ 値を問い合わせます。	5-85
:CURSor:VERTical[:BASic]: LINKage	垂直カーソルのリンクの ON/OFF を設定 / 問い合わせします。	5-85
:CURSor:VERTical[:BASic]:PERDt?	垂直カーソルの $1/\Delta T$ 測定に関するすべての設定値を問い合わせます。	5-86
:CURSor:VERTical[:BASic]:PERDt: STATe	垂直カーソルの $1/\Delta T$ 測定の ON/OFF を設定 / 問い合わせします。	5-86
:CURSor:VERTical[:BASic]:PERDt: VALue?	垂直カーソルの $1/\Delta T$ 値を問い合わせます。	5-86
:CURSor:VERTical[:BASic]:T<x>?	各垂直カーソルに関するすべての設定値を問い合わせます。	5-86
:CURSor:VERTical[:BASic]:T<x>: JUMP	各垂直カーソルを指定したズーム波形の中央位置にジャンプします。	5-86
:CURSor:VERTical[:BASic]:T<x>: POSition	各垂直カーソルの位置を設定 / 問い合わせします。	5-86
:CURSor:VERTical[:BASic]:T<x>: STATe	各垂直カーソルの ON/OFF を設定 / 問い合わせします。	5-86
:CURSor:VERTical[:BASic]:T<x>: VALue?	各垂直カーソルの時間値を問い合わせます。	5-86
:CURSor:VERTical:CALCulation?	垂直カーソルの Calc アイテムに関するすべての設定値を問い合わせます。	5-86
:CURSor:VERTical:CALCulation: ALL	垂直カーソルの Calc アイテムすべてを ON/OFF します。	5-86
:CURSor:VERTical:CALCulation: DEFine<x>	垂直カーソルの Calc アイテムの演算式を設定 / 問い合わせします。	5-86
:CURSor:VERTical:CALCulation: STATe<x>	垂直カーソルの Calc アイテムの ON/OFF を設定 / 問い合わせします。	5-87
:CURSor:VERTical:CALCulation: VALue<x>?	垂直カーソルの Calc アイテムの測定値を問い合わせます。	5-87
:CURSor:VT?	VT カーソルに関するすべての設定値を問い合わせます。	5-87
:CURSor:VT:BASic?	VT カーソルの Basic アイテムに関するすべての設定値を問い合わせます。	5-87
:CURSor:VT[:BASic]:ALL	VT カーソルの Basic アイテムすべてを ON/OFF します。	5-87
:CURSor:VT[:BASic]:GROup<x>?	VT カーソルの各ロジックグループに関するすべての設定値を問い合わせます。	5-87
:CURSor:VT[:BASic]:GROup<x>: BITOrder	VT カーソルの各ロジックグループの測定ビットオーダを設定 / 問い合わせします。	5-87
:CURSor:VT[:BASic]:GROup<x>: FORMat	VT カーソルの各ロジックグループの測定値の表示形式を設定 / 問い合わせします。	5-88
:CURSor:VT[:BASic]:GROup<x>: STATe	VT カーソルの各ロジックグループの測定値の ON/OFF を設定 / 問い合わせします。	5-88
:CURSor:VT[:BASic]:GROup<x>: VALue?	VT カーソルの各ロジックグループの測定値を問い合わせます。	5-88

## 5.1 コマンド一覧表

コマンド	機能	ページ
:CURSor:VT[:BASic]:POSition	VT カーソルの位置を設定 / 問い合わせします。	5-88
:CURSor:VT[:BASic]:T?	VT カーソルの時間測定に関するすべての設定値を問い合わせします。	5-88
:CURSor:VT[:BASic]:T:STATe	VT カーソルの時間測定の ON/OFF を設定 / 問い合わせします。	5-88
:CURSor:VT[:BASic]:T:VALue?	VT カーソルの時間値を問い合わせします。	5-88
:CURSor:VT[:BASic]:V<x>?	VT カーソルの電圧測定に関するすべての設定値を問い合わせします。	5-88
:CURSor:VT[:BASic]:V<x>:STATe	VT カーソルの電圧値測定の ON/OFF を設定 / 問い合わせします。	5-88
:CURSor:VT[:BASic]:V<x>:VALue?	VT カーソルの電圧値を問い合わせします。	5-88
:CURSor:VT:CALCulation?	VT カーソルの Calc アイテムに関するすべての設定値を問い合わせます。	5-88
:CURSor:VT:CALCulation:ALL	VT カーソルの Calc アイテムすべてを ON/OFF します。	5-88
:CURSor:VT:CALCulation: DEFine<x>	VT カーソルの Calc アイテムの演算式を設定 / 問い合わせします。	5-89
:CURSor:VT:CALCulation:STATe<x>	VT カーソルの Calc アイテムの ON/OFF を設定 / 問い合わせします。	5-89
:CURSor:VT:CALCulation: VALue<x>?	VT カーソルの Calc アイテムの測定値を問い合わせします。	5-89
:CURSor:VT:JUMP	VT カーソルを指定したズーム波形の中央位置にジャンプします。	5-89

### DISPlay グループ

:DISPlay?	表示に関するすべての設定値を問い合わせます。	5-90
:DISPlay:ACCumulate?	アキュムレート (波形の重ね書き表示) に関するすべての設定値を問い合わせます。	5-90
:DISPlay:ACCumulate:GRADe	アキュムレートモードを設定 / 問い合わせします。	5-90
:DISPlay:ACCumulate:MODE	アキュムレートモードの ON/OFF を設定 / 問い合わせします。	5-90
:DISPlay:ACCumulate: PERSistence?	パーシスタンスに関するすべての設定値を問い合わせます。	5-90
:DISPlay:ACCumulate: PERSistence:COUNT	パーシスタンスの回数を設定 / 問い合わせします。	5-90
:DISPlay:ACCumulate: PERSistence:MODE	パーシスタンスのモードを設定 / 問い合わせします。	5-90
:DISPlay:ACCumulate: PERSistence:TIME	パーシスタンスの時間を設定 / 問い合わせします。	5-90
:DISPlay:BLIGHt?	バックライトに関するすべての設定値を問い合わせます。	5-90
:DISPlay:BLIGHt:AUTOoff	バックライトを自動的に消すかどうかを設定 / 問い合わせします。	5-91
:DISPlay:BLIGHt:BRIGHtness	バックライトの輝度を設定 / 問い合わせします。	5-91
:DISPlay:BLIGHt:LCD	バックライトの ON/OFF を設定 / 問い合わせします。	5-91
:DISPlay:BLIGHt:TIMEout	バックライトのタイムアウトを設定 / 問い合わせします。	5-91
:DISPlay:COLor?	波形の表示色に関するすべての設定値を問い合わせます。	5-91
:DISPlay:COLor:{CHANnel<x> LOGic LState MATH<x> REFerence<x>}	各々の波形の色を設定 / 問い合わせします。	5-91
:DISPlay:FORMat	表示フォーマットを設定 / 問い合わせします。	5-91
:DISPlay:GRATICule	グラティクル (目盛り) を設定 / 問い合わせします。	5-91
:DISPlay:INTENSity?	表示アイテムの輝度に関するすべての設定値を問い合わせます。	5-91
:DISPlay:INTENSity[:WAVEform]	波形の輝度を設定 / 問い合わせします。	5-91
:DISPlay:INTENSity:{CURSor GRID MARKer ZBOX}	各表示アイテムの輝度を設定 / 問い合わせします。	5-92
:DISPlay:INTERpolate	表示補間方式を設定 / 問い合わせします。	5-92
:DISPlay:MAPPing?	分割画面への波形の割り当てに関するすべての設定値を問い合わせます。	5-92
:DISPlay:MAPPing[:MODE]	分割フォーマットへの波形の割り当てのモードを設定 / 問い合わせします。	5-92
:DISPlay:MAPPing:TRACe<x>	分割フォーマットへの各波形の割り当てを設定 / 問い合わせします。	5-92
:DISPlay:TRANslucent	半透明モードの ON/OFF を設定 / 問い合わせします。	5-92

### FILE グループ

:FILE?	指定したストレージメディア、内部メモリに関するすべての設定値を問い合わせます。	5-93
:FILE:DIRectory?	指定したストレージメディアに関するすべての設定値を問い合わせます。	5-93
:FILE[:DIRectory]:CDIRectory (Change Directory)	カレントディレクトリを設定 / 問い合わせします。	5-93
:FILE[:DIRectory]:FREE?	指定したストレージメディアの空き容量をバイト数で問い合わせます。	5-93
:FILE[:DIRectory]:MDIRectory (Make Directory)	指定したディレクトリの下にディレクトリを作成します。オーバーラップコマンドです。	5-94
:FILE:INTernal?	内部メモリに関するすべての設定値を問い合わせます。	5-94
:FILE:INTernal:RECall:DMEMory[: EXECute]	内部メモリからのアキュムレートデータの読み込みを実行します。オーバーラップコマンドです。	5-94

コマンド	機能	ページ
:FILE:INTernal:RECall:SETup[:EXECute]	内部メモリからの設定データの読み込みを実行します。オーバーラップコマンドです。	5-94
:FILE:INTernal:STORe?	内部メモリへの保存に関するすべての設定値を問い合わせます。	5-94
:FILE:INTernal:STORe:BINary?	内部メモリへのバイナリデータの保存に関するすべての設定値を問い合わせます。	5-94
:FILE:INTernal:STORe:BINary[:EXECute]	内部メモリへのバイナリデータの保存を実行します。オーバーラップコマンドです。	5-94
:FILE:INTernal:STORe:BINary:TRACe	内部メモリへのバイナリデータで保存するトレースを設定/問い合わせします。	5-94
:FILE:INTernal:STORe:DMEMory?	内部メモリへのアキュムレートデータの保存に関するすべての設定値を問い合わせます。	5-94
:FILE:INTernal:STORe:DMEMory[:EXECute]	内部メモリへのアキュムレートデータの保存を実行します。オーバーラップコマンドです。	5-95
:FILE:INTernal:STORe:DMEMory:TRACe	内部メモリへのアキュムレートデータで保存するトレースを設定/問い合わせします。	5-95
:FILE:INTernal:STORe:SETup?	内部メモリへの設定データの保存に関するすべての設定値を問い合わせます。	5-95
:FILE:INTernal:STORe:SETup:COMMeNt<x>	内部メモリへの設定データ保存時のコメントを設定/問い合わせします。	5-95
:FILE:INTernal:STORe:SETup:DATE<x>?	内部メモリへの設定データ保存時の日付/時刻を問い合わせます。	5-95
:FILE:INTernal:STORe:SETup[:EXECute]	内部メモリへの設定データの保存を実行します。オーバーラップコマンドです。	5-95
:FILE:INTernal:STORe:SETup:LOCK<x>	内部メモリへの設定データ書き禁止のON/OFFを設定/問い合わせします。	5-95
:FILE:INTernal:UNLoad:DMEMory[:EXECute]	読み込んだアキュムレートデータを取り消します。	5-95
:FILE:LOAD?	指定したストレージメディアのファイルの読み込みに関するすべての設定値を問い合わせます。	5-95
:FILE:LOAD:{BINary ZPOLYgon ZWAve}?	各種データの読み込みに関するすべての設定値を問い合わせます。	5-95
:FILE:LOAD:{BINary DMEMory MASK SETup SYMBol ZPOLYgon ZWAve}:ABORt	各種データの読み込みを中止します。	5-96
:FILE:LOAD:{BINary DMEMory MASK SETup SYMBol ZPOLYgon ZWAve}[:EXECute]	各種データの読み込みを実行します。オーバーラップコマンドです。	5-96
:FILE:LOAD:BINary:REFeRence	バイナリデータの読み込み先を設定/問い合わせします。	5-96
:FILE:LOAD:{ZPOLYgon ZWAve}:ZONE	各種データのゾーンデータの読み込み先を設定/問い合わせします。	5-96
:FILE:SAVE?	データの保存に関するすべての設定値を問い合わせます。	5-96
:FILE:SAVE:{AHIStoGram ASCIi BINary DMEMory FFT FLOat SBUS ZWAve}?	各種データの保存に関するすべての設定値を問い合わせます。	5-96
:FILE:SAVE:{AHIStoGram ASCIi BINary DMEMory FFT FLOat SETup SBUS WPARaMeter ZWAve}:ABORt	各種データの保存を中止します。	5-96
:FILE:SAVE:{AHIStoGram ASCIi BINary DMEMory FFT FLOat SETup SBUS WPARaMeter ZWAve}[:EXECute]	各種データをファイル名を指定して保存を実行します。オーバーラップコマンドです。	5-97
:FILE:SAVE:{AHIStoGram FFT SBUS}:ANALySis	各種データで保存するトレースを設定/問い合わせします。	5-97
:FILE:SAVE:ANAMing	保存するデータのファイル名の自動作成の種類を設定/問い合わせします。	5-97
:FILE:SAVE:{ASCIi BINary FLOat}:COMPRession	各種データの圧縮保存方法を設定/問い合わせします。	5-97
:FILE:SAVE:{ASCIi BINary FLOat}:HISTory	各種データの保存対象が全データか選択されたデータかを設定/問い合わせします。	5-97
:FILE:SAVE:{ASCIi BINary FLOat}:LENGth	各種データで保存する波形データのサイズを設定/問い合わせします。	5-97
:FILE:SAVE:ASCIi:RANGe	アスキーデータの保存する範囲を設定/問い合わせします。	5-97
:FILE:SAVE:{ASCIi BINary FLOat}:TRACe	各種データで保存するトレースを設定/問い合わせします。	5-98
:FILE:SAVE:COMMeNt	各種データの保存するコメントを設定/問い合わせします。	5-98
:FILE:SAVE:DMEMory:TRACe	アキュムレートデータで保存するトレースを設定/問い合わせします。	5-98
:FILE:SAVE:NAME	各種データの保存するデータのファイル名を設定/問い合わせします。	5-98

## 5.1 コマンド一覧表

コマンド	機能	ページ
:FILE:SAVE:ZWAVE:ZONE	ゾーンデータで保存するゾーンを設定 / 問い合わせします。	5-98
<b>GONogo グループ</b>		
:GONogo?	GO/NO-GO 判定に関するすべての設定値を問い合わせます。	5-99
:GONogo:ABORt	GO/NO-GO 判定を中止します。	5-99
:GONogo:ACTion?	判定が NO-GO 時の動作およびその基準に関するすべての設定値を問い合わせます。	5-99
:GONogo:ACTion:BUZZer	判定が NO-GO 時に警告音を鳴らす (ON)/ 鳴らさない (OFF) を設定 / 問い合わせします。	5-99
:GONogo:ACTion:HCOpy	判定が NO-GO 時にプリンタに画面イメージを出力する (ON)/ しない (OFF) を設定 / 問い合わせします。	5-99
:GONogo:ACTion:MAIL?	判定が NO-GO 時のメール送信に関するすべての設定値を問い合わせます。	5-99
:GONogo:ACTion:MAIL:INTerval	判定が NO-GO 時にメール送信する間隔を設定 / 問い合わせします。	5-99
:GONogo:ACTion:MAIL:MODE	判定が NO-GO 時にメール転送する (ON)/ しない (OFF) を設定 / 問い合わせします。	5-99
:GONogo:ACTion:SAVE	判定が NO-GO 時に波形データをメディアに保存する (ON)/ しない (OFF) を設定 / 問い合わせします。	5-100
:GONogo:CONDition<x>	GO/NO-GO 判定の各判定基準を設定 / 問い合わせします。	5-100
:GONogo:COUNT?	GO/NO-GO の実判定回数を問い合わせます。	5-100
:GONogo:EXECute	GO/NO-GO 判定を実行します。オーバーラップコマンドです。	5-100
:GONogo:LOGic	GO/NO-GO 判定のロジックを設定 / 問い合わせします。	5-100
:GONogo:MODE	GO/NO-GO 判定の種類を設定 / 問い合わせします。	5-100
:GONogo:NGCount?	GO/NO-GO 判定の実 NO-GO 回数を問い合わせます。	5-100
:GONogo:SCONdition? (Stop Condition)	判定終了条件に関するすべての設定値を問い合わせます。	5-100
:GONogo[:SCONdition]:NGCount	判定終了 NO-GO 回数を設定 / 問い合わせします。	5-100
:GONogo[:SCONdition]:STOPcount	判定終了取り込み回数を設定 / 問い合わせします。	5-100
:GONogo:TELEcomtest?	テレコムテスト判定に関するすべての設定値を問い合わせます。	5-100
:GONogo:TELEcomtest:SElect<x>?	テレコムテスト判定の各条件に関するすべての設定値を問い合わせます。	5-100
:GONogo:TELEcomtest:SElect<x>:EYEPattern?	各条件のアイパターン判定に関するすべての設定値を問い合わせます。	5-101
:GONogo:TELEcomtest:SElect<x>:EYEPattern:<パラメータ>	アイパターン判定の波形パラメータの上下限値を設定 / 問い合わせします。	5-101
:GONogo:TELEcomtest:SElect<x>:MASK?	各条件のマスク判定に関するすべての設定値を問い合わせます。	5-101
:GONogo:TELEcomtest:SElect<x>:MASK:ELEMent<x>?	マスク判定に使用する各エレメントに関するすべての設定値を問い合わせます。	5-101
:GONogo:TELEcomtest:SElect<x>:MASK:ELEMent<x>:PSPCount (Sample Point Count %)	各エレメントのサンプルデータ数に対するエラー率の上下限値を設定 / 問い合わせします。	5-101
:GONogo:TELEcomtest:SElect<x>:MASK:ELEMent<x>:PWCount (Wave Count %)	各エレメントのアクイジション回数に対するエラー率の上下限値を設定 / 問い合わせします。	5-101
:GONogo:TELEcomtest:SElect<x>:MASK:ELEMent<x>:SPCount (Sample Point Count)	各エレメントのエラーになったサンプルデータ数の上下限値を設定 / 問い合わせします。	5-102
:GONogo:TELEcomtest:SElect<x>:MASK:ELEMent<x>:WCount (Wave Count)	各エレメントのエラーになったアクイジション回数の上下限値を設定 / 問い合わせします。	5-102
:GONogo:ZPARAMeter?	ゾーン / パラメータ判定に関するすべての設定値を問い合わせます。	5-102
:GONogo:ZPARAMeter:SElect<x>?	ゾーン / パラメータ判定の各条件に関するすべての設定値を問い合わせます。	5-102
:GONogo:ZPARAMeter:SElect<x>:MODE	各条件のモードを設定 / 問い合わせします。	5-102
:GONogo:ZPARAMeter:SElect<x>:PARAMeter?	各条件のパラメータに関するすべての設定値を問い合わせます。	5-103
:GONogo:ZPARAMeter:SElect<x>:PARAMeter:CATegory	パラメータのカテゴリを設定 / 問い合わせします。	5-103
:GONogo:ZPARAMeter:SElect<x>:PARAMeter:FFT<x>?	各 FFT 判定に関するすべての設定値を問い合わせます。	5-103
:GONogo:ZPARAMeter:SElect<x>:PARAMeter:FFT<x>:CALCulation<x>	各 FFT 判定の各 Calc アイテムの上下限値を設定 / 問い合わせします。	5-103
:GONogo:ZPARAMeter:SElect<x>:PARAMeter:FFT<x>:PEAK?	各 FFT 判定のピーク値に関するすべての設定値を問い合わせます。	5-103

コマンド	機能	ページ
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:FFT<x>:PEAK: DFREquency	各 FFT 判定のピーク周波数間の上下限値を設定 / 問い合わせします。	5-103
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:FFT<x>:PEAK:DV	各 FFT 判定のピーク電圧値間の上下限値を設定 / 問い合わせします。	5-104
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:FFT<x>:PEAK: FREquency<x>	各 FFT 判定の各ピーク周波数の上下限値を設定 / 問い合わせします。	5-104
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:FFT<x>:PEAK:V<x>	各 FFT 判定の各ピーク電圧値の上下限値を設定 / 問い合わせします。	5-104
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:MEASure?	波形パラメータの自動測定での判定 (メジャー判定) に関するすべての設定値を問い合わせます。	5-104
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:MEASure:BIT<x>?	メジャー判定の各ロジックビットに関するすべての設定値を問い合わせます。	5-104
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:MEASure:BIT<x>: AREA<x>?	メジャー判定の各エリアに関するすべての設定値を問い合わせます。	5-105
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:MEASure:BIT<x>: AREA<x>:TYPE?	メジャー判定のロジック波形パラメータに関するすべての設定値を問い合わせます。	5-105
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:MEASure:BIT<x>: AREA<x>:TYPE:<パラメータ>	メジャー判定のロジック波形パラメータの上下限値を設定 / 問い合わせします。	5-105
:GONogo:ZPARAMeter: SElect<x>:PARAmeter:MEASure: CALCulation<x>	メジャー判定の各 Calc アイテムの上下限値を設定 / 問い合わせします。	5-105
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:MEASure:STATistics	メジャー判定の統計値を設定 / 問い合わせします。	5-106
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:MEASure:TRACe<x>?	メジャー判定の各トレースに関するすべての設定値を問い合わせます。	5-106
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:MEASure:TRACe<x>: AREA<x>?	メジャー判定の各エリアに関するすべての設定値を問い合わせます。	5-106
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:MEASure:TRACe<x>: AREA<x>:TYPE?	メジャー判定の波形パラメータに関するすべての設定値を問い合わせます。	5-106
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:MEASure:TRACe<x>: AREA<x>:TYPE:<パラメータ>	メジャー判定の波形パラメータの上下限値を設定 / 問い合わせします。	5-106
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:XY<x>?	各 XY 判定に関するすべての設定値を問い合わせます。	5-107
:GONogo:ZPARAMeter:SElect<x>: PARAmeter:XY<x>:XYINteg	各 XY 判定の積分値の上下限値を設定 / 問い合わせします。	5-107
:GONogo:ZPARAMeter:SElect<x>: RECTangle?	Rectangle 判定に関するすべての設定値を問い合わせます。	5-107
:GONogo:ZPARAMeter:SElect<x>: RECTangle:HORizontal	Rectangle 判定で用いる四角形の水平位置を設定 / 問い合わせします。	5-107
:GONogo:ZPARAMeter:SElect<x>: RECTangle:VERTical	Rectangle 判定で用いる四角形の垂直位置を設定 / 問い合わせします。	5-107
:GONogo:ZPARAMeter:SElect<x>: TRACe	ゾーン / パラメータ判定の対象トレースを設定 / 問い合わせします。	5-107
:GONogo:ZPARAMeter:SElect<x>: WAVE?	WAVE 判定に関するすべての設定値を問い合わせます。	5-107
:GONogo:ZPARAMeter:SElect<x>: WAVE:EDIT<x>:EXIT	WAVE 判定のゾーンの編集メニューから抜けます。	5-108
:GONogo:ZPARAMeter:SElect<x>: WAVE:EDIT<x>:PART	WAVE 判定のゾーン部分編集をします。	5-108
:GONogo:ZPARAMeter:SElect<x>: WAVE:EDIT<x>:WHOLE	波形ゾーンの全体ゾーン編集を設定します。	5-108
:GONogo:ZPARAMeter:SElect<x>: WAVE:TRANge	各ゾーン判定の判定区間を設定 / 問い合わせします。	5-108
:GONogo:ZPARAMeter:SElect<x>: WINDow	各ゾーン判定の対象ウィンドウを設定 / 問い合わせします。	5-108



## 5.1 コマンド一覧表

コマンド	機能	ページ
<b>Hcopy グループ</b>		
:HCOpy? (Hard COPY)	画面データの出力に関するすべての設定値を問い合わせます。	5-109
:HCOpy:ABORt	データ出力、紙送りを中止します。	5-109
:HCOpy:DIRection	データ出力先を設定 / 問い合わせします。	5-109
:HCOpy:EXECute	データ出力を実行します。オーバーラップコマンドです。	5-109
:HCOpy:EXTPrinter?	外部プリンタ出力に関するすべての設定値を問い合わせます。	5-109
:HCOpy:EXTPrinter:TONE	外部プリンタ出力のハーフトーンを設定 / 問い合わせします。	5-109
:HCOpy:EXTPrinter:TYPE	外部プリンタ出力コマンドの種類を設定 / 問い合わせします。	5-109
:HCOpy:FILE?	ファイル出力に関するすべての設定値を問い合わせます。	5-109
:HCOpy:FILE:FORMat	ファイル出力画像フォーマットを設定 / 問い合わせします。	5-109
:HCOpy:FILE:SAVE?	ファイル出力の保存に関するすべての設定値を問い合わせます。	5-109
:HCOpy:FILE:SAVE:ANAMing	保存ファイル名の自動作成の種類を設定 / 問い合わせします。	5-110
:HCOpy:FILE:SAVE:CDIRectory (Change Directory)	保存ディレクトリ名を設定 / 問い合わせします。	5-110
:HCOpy:FILE:SAVE:NAME	保存ファイル名を設定 / 問い合わせします。	5-110
:HCOpy:NETPrint?	ネットワークプリンタ出力に関するすべての設定値を問い合わせします。	5-110
:HCOpy:NETPrint:TONE	ネットワークプリンタ出力のハーフトーンを設定 / 問い合わせします。	5-110
:HCOpy:NETPrint:TYPE	ネットワークプリンタ出力コマンドの種類を設定 / 問い合わせします。	5-110
:HCOpy:PRINter?	内蔵プリンタ出力に関するすべての設定値を問い合わせます。	5-110
:HCOpy:PRINter:HRMode	内蔵プリンタ出力の高分解モードの ON/OFF を設定 / 問い合わせします。	5-110
<b>HISTory グループ</b>		
:HISTory?	ヒストリ機能に関するすべての設定を問い合わせます。	5-111
:HISTory:CURRent?	カレント波形 (CH1 ~ 4, M1 ~ 8) のヒストリ機能に関するすべての設定を 問い合わせます。	5-111
:HISTory[:CURRent]:DISPlay	ヒストリ波形の表示レコードの開始番号と終了番号を設定 / 問い合わせしま す。	5-111
:HISTory[:CURRent]:DMODE (Display Mode)	ヒストリ波形の表示モードを設定 / 問い合わせします。	5-112
:HISTory[:CURRent]:MODE	ヒストリ波形のハイライト表示モードを設定 / 問い合わせします。	5-112
:HISTory[:CURRent]:RECORD	ヒストリ波形の対象レコードを設定 / 問い合わせします。	5-112
:HISTory[:CURRent]:RECORD? MINimum	ヒストリ波形の最小のレコード番号を問い合わせます。	5-112
:HISTory[:CURRent]:REPLay?	ヒストリ波形のリプレイ機能に関するすべての設定を問い合わせます。	5-112
:HISTory[:CURRent]:REPLay:JUMP	ヒストリ波形を指定のレコード番号へジャンプさせます。	5-112
:HISTory[:CURRent]:REPLay:SPEEd	ヒストリ波形のリプレイ速度を設定 / 問い合わせします。	5-112
:HISTory[:CURRent]:REPLay:STARt	ヒストリ波形のリプレイを指定方向へ開始します。	5-112
:HISTory[:CURRent]:REPLay:STOP	ヒストリ波形のリプレイを停止します。	5-112
:HISTory[:CURRent]:SEARCh?	ヒストリサーチ機能に関するすべての設定を問い合わせます。	5-113
:HISTory[:CURRent][:SEARCh]: ABORt	ヒストリサーチを中止します。	5-113
:HISTory[:CURRent][:SEARCh]: EXECute	ヒストリサーチを実行します。オーバーラップコマンドです。	5-113
:HISTory[:CURRent][:SEARCh]: LOGic	ヒストリサーチのロジックを設定 / 問い合わせします。	5-113
:HISTory[:CURRent][:SEARCh]: RESet	ヒストリサーチのサーチ条件をリセットします。	5-113
:HISTory[:CURRent][:SEARCh]: SELEct<x>?	各ヒストリサーチ条件に関するすべての設定値を問い合わせます。	5-113
:HISTory[:CURRent][:SEARCh]: SELEct<x>:CONDition	各ヒストリサーチ条件の判定基準を設定 / 問い合わせします。	5-113
:HISTory[:CURRent][:SEARCh]: SELEct<x>:MODE	各ヒストリサーチ条件のモードを設定 / 問い合わせします。	5-113
:HISTory[:CURRent][:SEARCh]: SELEct<x>:PARAMeter?	各ヒストリサーチ条件のパラメータに関する設定値をすべて問い合わせま す。	5-114
:HISTory[:CURRent][:SEARCh]: SELEct<x>:PARAMeter:CATegory	パラメータのカテゴリを設定 / 問い合わせします。	5-114
:HISTory[:CURRent][:SEARCh]: SELEct<x>:PARAMeter:FFT<x>?	各 FFT サーチに関するすべての設定値を問い合わせます。	5-114
:HISTory[:CURRent][:SEARCh]: SELEct<x>:PARAMeter:FFT<x>: CALCulation<x>	各 FFT サーチの各 Calc アイテムの上下限值を設定 / 問い合わせします。	5-114

コマンド	機能	ページ
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:FFT<x>: PEAK?	各 FFT サーチのピーク値に関するすべての設定値を問い合わせます。	5-114
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:FFT<x>: PEAK:DFREquency	各 FFT サーチのピーク周波数間の上下限値を設定 / 問い合わせします。	5-115
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:FFT<x>: PEAK:DV	各 FFT サーチのピーク電圧値間の上下限値を設定 / 問い合わせします。	5-115
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:FFT<x>: PEAK:FREquency<x>	各 FFT サーチの各ピーク周波数の上下限値を設定 / 問い合わせします。	5-115
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:FFT<x>: PEAK:V<x>	各 FFT サーチの各ピーク電圧値の上下限値を設定 / 問い合わせします。	5-115
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:MEASure?	波形パラメータの自動測定での検索 (メジャーサーチ) に関するすべての設定値を問い合わせます。	5-115
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:MEASure: BIT<x>?	メジャーサーチの各ロジックビットに関するすべての設定値を問い合わせます。	5-116
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:MEASure: BIT<x>:AREA<x>?	メジャーサーチの各エリアに関するすべての設定値を問い合わせます。	5-116
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:MEASure: BIT<x>:AREA<x>:TYPE?	メジャーサーチのロジック波形パラメータに関するすべての設定値を問い合わせます。	5-116
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:MEASure: BIT<x>:AREA<x>:TYPE:<パラメータ>	メジャーサーチのロジック波形パラメータの上下限値を設定 / 問い合わせします。	5-116
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:MEASure: CALCulation<x>	メジャーサーチの各 Calc アイテムの上下限値を設定 / 問い合わせします。	5-116
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:MEASure: TRACe<x>?	メジャーサーチの各トレースに関するすべての設定値を問い合わせます。	5-117
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:MEASure: TRACe<x>:AREA<x>?	メジャーサーチの各エリアに関するすべての設定値を問い合わせます。	5-117
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:MEASure: TRACe<x>:AREA<x>:TYPE?	メジャーサーチの波形パラメータに関するすべての設定値を問い合わせます。	5-117
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:MEASure: TRACe<x>:AREA<x>:TYPE:<パラメータ>	メジャーサーチの波形パラメータの上下限値を設定 / 問い合わせします。	5-117
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:XY<x>?	各 XY サーチに関するすべての設定値を問い合わせます。	5-118
:HISTory[:CURRent][:SEARCh]: SElect<x>:PARAmeter:XY<x>: XYINteg	各 XY サーチの積分値の上下限値を設定 / 問い合わせします。	5-118
:HISTory[:CURRent][:SEARCh]: SElect<x>:RECTangle?	Rectangle サーチに関するすべての設定値を問い合わせます。	5-118
:HISTory[:CURRent][:SEARCh]: SElect<x>:RECTangle:HORizontal	Rectangle サーチで用いる四角形の水平位置を設定 / 問い合わせします。	5-118
:HISTory[:CURRent][:SEARCh]: SElect<x>:RECTangle:VERTical	Rectangle サーチで用いる四角形の垂直位置を設定 / 問い合わせします。	5-118
:HISTory[:CURRent][:SEARCh]: SElect<x>:TRACe	ヒストリサーチの対象トレースを設定 / 問い合わせします。	5-118
:HISTory[:CURRent][:SEARCh]: SElect<x>:WAVE?	波形ゾーンでの検索 (WAVE サーチ) に関するすべての設定値を問い合わせます。	5-119
:HISTory[:CURRent][:SEARCh]: SElect<x>:WAVE:EDIT<x>:EXIT	WAVE サーチのゾーン編集メニューから抜けます。	5-119
:HISTory[:CURRent][:SEARCh]: SElect<x>:WAVE:EDIT<x>:PART	WAVE サーチのゾーン部分編集を設定します。	5-119

## 5.1 コマンド一覧表

コマンド	機能	ページ
:HISTory[:CURRent][:SEARCh]: SELEct<x>:WAVE:EDIT<x>:WHOLE	WAVE サーチのゾーン全体編集を設定します。	5-119
:HISTory[:CURRent][:SEARCh]: SELEct<x>:WAVE:TRANge	WAVE サーチの判定区間を設定 / 問い合わせします。	5-119
:HISTory[:CURRent][:SEARCh]: SELEct<x>:WINDow	各ヒストリサーチの対象ウィンドウを設定 / 問い合わせします。	5-119
:HISTory[:CURRent]:TIME?	ヒストリ波形の対象レコード番号の時間を問い合わせします。	5-120
:HISTory:REFeRence<x>?	各リファレンスのヒストリ機能に関するすべての設定値を問い合わせます。	5-120
:HISTory:REFeRence<x>: DMODE (Display Mode)	各リファレンスのヒストリ波形の表示モードを設定 / 問い合わせします。	5-120
:HISTory:REFeRence<x>:MODE	各リファレンスのヒストリ波形のハイライト表示モードを設定 / 問い合わせ	5-120
:HISTory:REFeRence<x>:RECORD	各リファレンスのヒストリ波形の対象レコードを設定 / 問い合わせします。	5-120
:HISTory:REFeRence<x>:RECORD? MINimum	各リファレンスのヒストリ波形の最小のレコード番号を問い合わせます。	5-120
:HISTory:REFeRence<x>:REPLay?	各リファレンスのヒストリ波形のリプレイ機能に関するすべての設定値を問	5-120
:HISTory:REFeRence<x>:REPLay: JUMP	各リファレンスのヒストリ波形を指定レコード番号へジャンプさせます。	5-120
:HISTory:REFeRence<x>:REPLay: SPEEd	各リファレンスのヒストリ波形のリプレイ速度を設定 / 問い合わせします。	5-120
:HISTory:REFeRence<x>:REPLay: START	各リファレンスのヒストリ波形のリプレイを指定方向へ開始します。	5-120
:HISTory:REFeRence<x>:REPLay: STOP	各リファレンスのヒストリ波形のリプレイを停止します。	5-121
:HISTory:REFeRence<x>:TIME?	リファレンス波形の対象レコード番号の時間を問い合わせします。	5-121

### IMAGe グループ

:IMAGe?	画面イメージデータの出力に関するすべての設定値を問い合わせます。	5-122
:IMAGe:FORMAt	画面イメージデータの出力形式を設定 / 問い合わせします。	5-122
:IMAGe:SEND?	画面イメージデータを問い合わせます。	5-122
:IMAGe:TONE	画面イメージデータの色調を設定 / 問い合わせします。	5-122

### INITialize グループ

:INITialize:EXECute	イニシャライズを実行します。	5-123
:INITialize:UNDO	実行したイニシャライズを取り消します。	5-123

### LOGic グループ

:LOGic?	ロジックに関するすべての設定値を問い合わせます。	5-124
:LOGic:AWINDow?	アナログ波形表示に関するすべての設定値を問い合わせます。	5-124
:LOGic:AWINDow[:DISPlay]	アナログ波形表示の ON/OFF を設定 / 問い合わせします。	5-124
:LOGic:AWINDow:RATio	アナログ波形の表示比率を設定 / 問い合わせします。	5-124
:LOGic:DESKew	ロジック信号のスキュー補正を設定 / 問い合わせします。	5-124
:LOGic:GROup<x>?	各ロジックグループのすべての設定値を問い合わせます。	5-124
:LOGic:GROup<x>:BUNDle?	各ロジックグループのバンドルに関するすべての設定値を問い合わせます。	5-124
:LOGic:GROup<x>:BUNDle:FORMAt	各ロジックグループのバンドル値の表示形式 (バス表示) を設定 / 問	5-125
:LOGic:GROup<x>:BUNDle:MODE	各ロジックグループのバンドルのモードを設定 / 問い合わせします。	5-125
:LOGic:GROup<x>:BUNDle:SYMBol	各ロジックグループのバンドル値のシンボルアイテムを設定します。	5-125
:LOGic:GROup<x>:DISPlay	各ロジックグループの表示の ON/OFF を設定 / 問い合わせします。	5-125
:LOGic:GROup<x>:MAPPing	各ロジックグループのビット配置を設定 / 問い合わせします。	5-125
:LOGic:GROup<x>:ORDer	各ロジックグループの表示順を設定 / 問い合わせします。	5-125
:LOGic:GROup<x>:STATe	各ロジックグループのステート表示の ON/OFF を設定 / 問い合わせします。	5-125
:LOGic:LAbel?	ロジック信号のラベル名に関するすべての設定値を問い合わせます。	5-125
:LOGic:LAbel:BNAMe?	ロジック信号のビット名に関するすべての設定値を問い合わせます。	5-125
:LOGic:LAbel:BNAMe:{A<x> B<x> C<x> D<x>}	ロジック信号の各ビット名を設定 / 問い合わせします。	5-126
:LOGic:LAbel:LNAMe?	ロジック信号のグループ名に関するすべての設定値を問い合わせます。	5-126
:LOGic:LAbel:LNAMe:GROup<x>	ロジック信号の各グループ名を設定 / 問い合わせします。	5-126
:LOGic:LAbel:MODE	ロジック信号のラベル名の ON/OFF を設定 / 問い合わせします。	5-126
:LOGic:MODE	ロジック信号の ON/OFF を設定 / 問い合わせします。	5-126
:LOGic:POSition	ロジック信号の垂直ポジションを設定 / 問い合わせします。	5-126

コマンド	機能	ページ
:LOGic:SCLock?	ロジック信号のステータクロックに関するすべての設定値を問い合わせます。	5-126
:LOGic:SCLock:POLarity	ロジック信号のステータクロックの極性を設定 / 問い合わせします。	5-126
:LOGic:SCLock:SOURce	ロジック信号のステータクロックソースを設定 / 問い合わせします。	5-126
:LOGic:SIZE	ロジック信号の表示サイズを設定 / 問い合わせします。	5-126
:LOGic:THReshold?	ロジック信号のスレシヨルドレベルに関するすべての設定値を問い合わせます。	5-127
:LOGic:THReshold:{PODA PODB PODC PODD}?	指定ポッド (ポート) のスレシヨルドレベルに関するすべての設定値を問い合わせます。	5-127
:LOGic:THReshold:{PODA PODB PODC PODD}:TYPE	指定ポッド (ポート) のスレシヨルドレベルのタイプを設定 / 問い合わせします。	5-127
:LOGic:THReshold:{PODA PODB PODC PODD}:USERlevel	指定ポッド (ポート) のスレシヨルドレベルを設定 / 問い合わせします。	5-127

## MATH グループ

:MATH<x>?	演算に関するすべての設定値を問い合わせます。	5-128
:MATH<x>:DA?	D/A 変換に関するすべての設定値を問い合わせます。	5-128
:MATH<x>:DA:ARANging	D/A 変換のオートレンジを実行します。	5-128
:MATH<x>:DA:BFORmat	D/A 変換のバイナリフォーマットを設定 / 問い合わせます。	5-128
:MATH<x>:DA:HISTory:ABORt	D/A 変換のヒストリ演算を中止します。	5-128
:MATH<x>:DA:HISTory:EXECute	D/A 変換のヒストリ演算を実行します。	5-128
:MATH<x>:DA:RESCaling?	D/A 変換のリスケージングに関するすべての設定値を問い合わせます。	5-128
:MATH<x>:DA:RESCaling:AVALue	D/A 変換のリスケージング係数 A を設定 / 問い合わせします。	5-128
:MATH<x>:DA:RESCaling:BVALue	D/A 変換のリスケージングオフセット値 B を設定 / 問い合わせします。	5-128
:MATH<x>:DISPlay	演算波形の ON/OFF を設定 / 問い合わせします。	5-128
:MATH<x>:ECOUNT?	エッジカウント演算に関するすべての設定値を問い合わせます。	5-129
:MATH<x>:ECOUNT:HYSteresis	エッジカウント演算のエッジ検出レベルのヒステリシスを設定 / 問い合わせします。	5-129
:MATH<x>:ECOUNT:POLarity	エッジカウント演算のエッジ検出極性を設定 / 問い合わせします。	5-129
:MATH<x>:FILTer?	フィルタに関するすべての設定値を問い合わせます。	5-129
:MATH<x>:FILTer:DELAy?	遅延演算に関するすべての設定値を問い合わせます。	5-129
:MATH<x>:FILTer:DELAy:TIME	遅延演算の遅延時間を設定 / 問い合わせします。	5-129
:MATH<x>:FILTer:IIR?	IIR フィルタ演算に関するすべての設定値を問い合わせます。	5-129
:MATH<x>:FILTer:IIR:FORDer (Filter Order)	IIR フィルタ演算のフィルタ次数を設定 / 問い合わせします。	5-129
:MATH<x>:FILTer:IIR:HIPass?	IIR ハイパスフィルタ演算に関するすべての設定値を問い合わせます。	5-129
:MATH<x>:FILTer:IIR:HIPass:COFF	IIR ハイパスフィルタ演算のカットオフ周波数を設定 / 問い合わせします。	5-129
:MATH<x>:FILTer:IIR:LOWPass?	IIR ローパスフィルタ演算に関するすべての設定値を問い合わせます。	5-129
:MATH<x>:FILTer:IIR:LOWPass:COFF	IIR ローパスフィルタ演算のカットオフ周波数を設定 / 問い合わせします。	5-130
:MATH<x>:FILTer:MAVG?	移動平均演算に関するすべての設定値を問い合わせます。	5-130
:MATH<x>:FILTer:MAVG:WEIGHt	移動平均演算の重みを設定 / 問い合わせします。	5-130
:MATH<x>:FILTer:RESCaling?	フィルタのリスケージングに関するすべての設定値を問い合わせます。	5-130
:MATH<x>:FILTer:RESCaling:AVALue	フィルタのリスケージング係数値 A を設定 / 問い合わせします。	5-130
:MATH<x>:FILTer:RESCaling:BVALue	フィルタのリスケージングオフセット値 B を設定 / 問い合わせします。	5-130
:MATH<x>:FILTer:TYPE	フィルタのタイプを設定 / 問い合わせします。	5-130
:MATH<x>:INTegral?	積分演算に関するすべての設定値を問い合わせます。	5-130
:MATH<x>:INTegral:PSCaling?	積分演算のプリスケージングに関するすべての設定値を問い合わせます。	5-130
:MATH<x>:INTegral:PSCaling:AVALue	積分演算のプリスケージング係数値 A を設定 / 問い合わせします。	5-130
:MATH<x>:INTegral:PSCaling:BVALue	積分演算のプリスケージングオフセット値 B を設定 / 問い合わせします。	5-131
:MATH<x>:INTegral:RESCaling?	積分演算のリスケージングに関するすべての設定値を問い合わせます。	5-131
:MATH<x>:INTegral:RESCaling:AVALue	積分演算のリスケージング係数値 A を設定 / 問い合わせします。	5-131
:MATH<x>:INTegral:RESCaling:BVALue	積分演算のリスケージングオフセット値 B を設定 / 問い合わせします。	5-131
:MATH<x>:INVert	演算波形の反転表示の ON/OFF を設定 / 問い合わせします。	5-131
:MATH<x>:IPOINT? (Initial Point)	演算基準点に関するすべての設定値を問い合わせます。	5-131
:MATH<x>:IPOINT:JUMP	演算基準点を指定の場所にジャンプさせます。	5-131
:MATH<x>:IPOINT:POSition	演算基準点を設定 / 問い合わせします。	5-131

## 5.1 コマンド一覧表

コマンド	機能	ページ
:MATH<x>:LAbel?	演算波形のラベル名に関するすべての設定値を問い合わせます。	5-131
:MATH<x>:LAbel[:DEFine]	演算波形のラベル名を設定 / 問い合わせします。	5-131
:MATH<x>:LAbel:MODE	演算波形のラベル名表示の ON/OFF を設定 / 問い合わせします。	5-131
:MATH<x>:MINus?	減算に関するすべての設定値を問い合わせます。	5-132
:MATH<x>:MINus:PSCaling<x>?	減算のプリスケールに関するすべての設定値を問い合わせます。	5-132
:MATH<x>:MINus:PSCaling<x>:AVALue	減算のプリスケール係数値 A を設定 / 問い合わせします。	5-132
:MATH<x>:MINus:PSCaling<x>:BVALue	減算のプリスケールオフセット値 B を設定 / 問い合わせします。	5-132
:MATH<x>:MINus:RESCaling?	減算のリスケールに関するすべての設定値を問い合わせます。	5-132
:MATH<x>:MINus:RESCaling:AVALue	減算のリスケール係数値 A を設定 / 問い合わせします。	5-132
:MATH<x>:MINus:RESCaling:BVALue	減算のリスケールオフセット値 B を設定 / 問い合わせします。	5-132
:MATH<x>:MULTiple?	乗算に関するすべての設定値を問い合わせます。	5-132
:MATH<x>:MULTiple:PSCaling<x>?	乗算のプリスケールに関するすべての設定値を問い合わせます。	5-132
:MATH<x>:MULTiple:PSCaling<x>:AVALue	乗算のプリスケール係数値 A を設定 / 問い合わせします。	5-133
:MATH<x>:MULTiple:PSCaling<x>:BVALue	乗算のプリスケールオフセット値 B を設定 / 問い合わせします。	5-133
:MATH<x>:MULTiple:RESCaling?	乗算のリスケールに関するすべての設定値を問い合わせます。	5-133
:MATH<x>:MULTiple:RESCaling:AVALue	乗算のリスケール係数値 A を設定 / 問い合わせします。	5-133
:MATH<x>:MULTiple:RESCaling:BVALue	乗算のリスケールオフセット値 B を設定 / 問い合わせします。	5-133
:MATH<x>:OPERation	演算子を設定 / 問い合わせします。	5-133
:MATH<x>:PLUS?	加算に関するすべての設定値を問い合わせます。	5-133
:MATH<x>:PLUS:PSCaling<x>?	加算のプリスケールに関するすべての設定値を問い合わせます。	5-133
:MATH<x>:PLUS:PSCaling<x>:AVALue	加算のプリスケール係数値 A を設定 / 問い合わせします。	5-134
:MATH<x>:PLUS:PSCaling<x>:BVALue	加算のプリスケールオフセット値 B を設定 / 問い合わせします。	5-134
:MATH<x>:PLUS:RESCaling?	加算のリスケールに関するすべての設定値を問い合わせます。	5-134
:MATH<x>:PLUS:RESCaling:AVALue	加算のリスケール係数値 A を設定 / 問い合わせします。	5-134
:MATH<x>:PLUS:RESCaling:BVALue	加算のリスケールオフセット値 B を設定 / 問い合わせします。	5-134
:MATH<x>:POSITION	演算波形の垂直ポジションを設定 / 問い合わせします。	5-134
:MATH<x>:SCALE?	スケールに関するすべての設定値を問い合わせます。	5-134
:MATH<x>:SCALE:CENTer	演算波形のオフセットを設定 / 問い合わせします。	5-134
:MATH<x>:SCALE:MODE	スケール方法を設定 / 問い合わせします。	5-134
:MATH<x>:SCALE:SENSitivity	演算波形の垂直軸感度を設定 / 問い合わせします。	5-134
:MATH<x>:SElect	表示オプションを設定 / 問い合わせします。	5-135
:MATH<x>:SVALue (Scale VALUE)	スケール値表示の ON/OFF を設定 / 問い合わせします。	5-135
:MATH<x>:THReshold<x>	カウント演算のエッジ検出レベルを設定 / 問い合わせします。	5-135
:MATH<x>:UNIT?	演算単位に関するすべての設定値を問い合わせます。	5-135
:MATH<x>:UNIT[:DEFine]	演算単位を設定 / 問い合わせします。	5-135
:MATH<x>:UNIT:MODE	演算単位の自動 / 手動付加を設定 / 問い合わせします。	5-135
:MATH<x>:USERdefine?	ユーザー定義演算に関するすべての設定値を問い合わせします。	5-135
:MATH<x>:USERdefine:ARANging	ユーザー定義演算のオートレンジを実行します。	5-135
:MATH<x>:USERdefine:CONSTant<x>	ユーザー定義演算の定数を設定 / 問い合わせします。	5-135
:MATH<x>:USERdefine:DEFine	ユーザー定義演算の演算式を設定 / 問い合わせします。	5-135
:MATH<x>:USERdefine:HISTory:	ユーザー定義演算のヒストリ演算を中止します。	5-136
ABORt		
:MATH<x>:USERdefine:HISTory:	ユーザー定義演算のヒストリ演算を実行します。	5-136
EXECute		
<b>MEASure グループ</b>		
:MEASure?	波形パラメータの自動測定に関するすべての設定値を問い合わせます。	5-137
:MEASure:BIT<x>?	各ロジックビットに関するすべての設定値を問い合わせます。	5-137
:MEASure:BIT<x>:AREA<x>?	各エリアに関するすべての設定値を問い合わせます。	5-137
:MEASure:BIT<x>:AREA<x>:ALL	ロジック波形パラメータのすべてを ON/OFF します。	5-138
:MEASure:BIT<x>:AREA<x>:<パラメータ>?	ロジック波形パラメータに関するすべての設定値を問い合わせます。	5-138
:MEASure:BIT<x>:AREA<x>:<パラメータ>:COUNT?	ロジック波形パラメータの継続統計処理の回数を問い合わせます。	5-138

コマンド	機能	ページ
:MEASure:BIT<x>:AREA<x>:<パラメータ>:{MAXimum MEAN MINimum SDEVIation}?	ロジック波形パラメータの各統計値を問い合わせます。	5-138
:MEASure:BIT<x>:AREA<x>:<パラメータ>:STATe	ロジック波形パラメータの ON/OFF を設定 / 問い合わせします。	5-138
:MEASure:BIT<x>:AREA<x>:<パラメータ>:VALue?	ロジック波形パラメータの自動測定値を問い合わせます。	5-139
:MEASure:BIT<x>:AREA<x>:DElay:MEASure?	チャンネル間ディレイ測定の測定対象波形の測定条件に関するすべての設定値を問い合わせます。	5-139
:MEASure:BIT<x>:AREA<x>:DElay:MEASure:COUNT	チャンネル間ディレイ測定の測定対象波形のエッジ検出回数を設定 / 問い合わせします。	5-139
:MEASure:BIT<x>:AREA<x>:DElay:MEASure:POLarity	チャンネル間ディレイ測定の測定対象波形の極性を設定 / 問い合わせします。	5-139
:MEASure:BIT<x>:AREA<x>:DElay:REference?	チャンネル間ディレイ測定の基準波形に関するすべての設定値を問い合わせます。	5-139
:MEASure:BIT<x>:AREA<x>:DElay:REference:COUNT	チャンネル間ディレイ測定の基準波形のエッジ検出回数を設定 / 問い合わせします。	5-140
:MEASure:BIT<x>:AREA<x>:DElay:REference:POLarity	チャンネル間ディレイ測定の基準波形の極性を設定 / 問い合わせします。	5-140
:MEASure:BIT<x>:AREA<x>:DElay:REference:TRACe	チャンネル間ディレイ測定の基準波形のトレースを設定 / 問い合わせします。	5-140
:MEASure:BIT<x>:AREA<x>:DElay:SOURce	チャンネル間ディレイ測定の基準を設定 / 問い合わせします。	5-140
:MEASure:CALCulation?	Calc アイテムに関するすべての設定値を問い合わせます。	5-140
:MEASure:CALCulation:ALL	Calc アイテムのすべてを ON/OFF します。	5-140
:MEASure:CALCulation:COUNT<x>?	Calc アイテムの統計処理の回数を問い合わせます。	5-140
:MEASure:CALCulation:DEFine<x>	Calc アイテムの演算式を設定 / 問い合わせします。	5-141
:MEASure:CALCulation:{MAXimum<x> MEAN<x> MINimum<x> SDEVIation<x>}?	Calc アイテムの各統計値を問い合わせます。	5-141
:MEASure:CALCulation:STATe<x>	Calc アイテムの ON/OFF を設定 / 問い合わせします。	5-141
:MEASure:CALCulation:VALue<x>?	Calc アイテムの自動測定値を問い合わせます。	5-141
:MEASure:CONTInuous?	継続統計処理 (CONTInuous Statistics) に関するすべての設定値を問い合わせます。	5-141
:MEASure:CONTInuous:COUNT	継続統計処理 (CONTInuous Statistics) の回数を設定 / 問い合わせします。	5-141
:MEASure:CONTInuous:REStArt	継続統計処理 (CONTInuous Statistics) を再スタートします。	5-141
:MEASure:CYCLe?	周期統計処理 (CYCLe Statistics) に関するすべての設定を問い合わせます。	5-141
:MEASure:CYCLe:ABORT	周期統計処理 (CYCLe Statistics) の実行を中止します。	5-141
:MEASure:CYCLe:EXECute	周期統計処理 (CYCLe Statistics) を実行します。オーバーラップコマンドです。	5-141
:MEASure:CYCLe:TRACe	周期統計処理 (CYCLe Statistics) の周期対象トレースを設定 / 問い合わせします。	5-142
:MEASure:DISPlay	波形パラメータの自動測定結果の表示の ON/OFF を設定 / 問い合わせします。	5-142
:MEASure:HISTory:ABORT	ヒストリデータの統計処理 (HISTory Statistics) の実行を中止します。	5-142
:MEASure:HISTory:EXECute	ヒストリデータの統計処理 (HISTory Statistics) を実行します。オーバーラップコマンドです。	5-142
:MEASure:MODE	波形パラメータの自動測定モードを設定 / 問い合わせします。	5-142
:MEASure:THReshold?	波形パラメータの自動測定のしきい値 (THReshold) に関するすべての設定値を問い合わせます。	5-142
:MEASure:THReshold:TRACe<x>?	各トレースのしきい値 (THReshold) に関するすべての設定値を問い合わせます。	5-142
:MEASure:THReshold:TRACe<x>:AUTO	しきい値 (THReshold) の自動設定時の検出方法を設定 / 問い合わせします。	5-143
:MEASure:THReshold:TRACe<x>:LHYSteresis?	しきい値 (THReshold) のレベル、ヒステリシス設定におけるすべての設定値を問い合わせます。	5-143
:MEASure:THReshold:TRACe<x>:LHYSteresis:HYSTeresis	しきい値 (THReshold) のヒステリシスを設定 / 問い合わせします。	5-143
:MEASure:THReshold:TRACe<x>:LHYSteresis:LEVel	しきい値 (THReshold) のレベルを設定 / 問い合わせします。	5-143
:MEASure:THReshold:TRACe<x>:MODE	しきい値 (THReshold) の設定方法を問い合わせます。	5-143
:MEASure:THReshold:TRACe<x>:ULOWer?	しきい値 (THReshold) の上下限值設定におけるすべての設定値を問い合わせます。	5-143
:MEASure:THReshold:TRACe<x>:ULOWer:RANGE	しきい値 (THReshold) の上下限值を設定 / 問い合わせします。	5-144

## 5.1 コマンド一覧表

コマンド	機能	ページ
:MEASure:TRACe<x>?	各トレースに関するすべての設定値を問い合わせます。	5-144
:MEASure:TRACe<x>:AREA<x>?	各エリアに関するすべての設定値を問い合わせます。	5-145
:MEASure:TRACe<x>:AREA<x>:ALL	波形パラメータのすべてを ON/OFF します。	5-145
:MEASure:TRACe<x>:AREA<x>:<パラメータ>?	波形パラメータに関するすべての設定値を問い合わせます。	5-145
:MEASure:TRACe<x>:AREA<x>:<パラメータ>:COUNT?	波形パラメータの継続統計処理の回数を問い合わせます。	5-145
:MEASure:TRACe<x>:AREA<x>:<パラメータ>:{MAXimum MEAN MINimum SD EVIation}?	波形パラメータの各統計値を問い合わせます。	5-146
:MEASure:TRACe<x>:AREA<x>:<パラメータ>:STATE	波形パラメータの ON/OFF を設定 / 問い合わせします。	5-146
:MEASure:TRACe<x>:AREA<x>:<パラメータ>:VALue?	波形パラメータの自動測定値を問い合わせます。	5-146
:MEASure:TRACe<x>:AREA<x>:DElay:MEASure?	チャンネル間ディレイ測定の測定対象波形の測定条件に関するすべての設定値を問い合わせます。	5-146
:MEASure:TRACe<x>:AREA<x>:DElay:MEASure:COUNT	チャンネル間ディレイ測定の測定対象波形のエッジ検出回数を設定 / 問い合わせします。	5-147
:MEASure:TRACe<x>:AREA<x>:DElay:MEASure:POLarity	チャンネル間ディレイ測定の測定対象波形の極性を設定 / 問い合わせします。	5-147
:MEASure:TRACe<x>:AREA<x>:DElay:REFerence?	チャンネル間ディレイ測定の基準波形に関するすべての設定値を問い合わせます。	5-147
:MEASure:TRACe<x>:AREA<x>:DElay:REFerence:COUNT	チャンネル間ディレイ測定の基準波形のエッジ検出回数を設定 / 問い合わせします。	5-147
:MEASure:TRACe<x>:AREA<x>:DElay:REFerence:POLarity	チャンネル間ディレイ測定の基準波形の極性を設定 / 問い合わせします。	5-147
:MEASure:TRACe<x>:AREA<x>:DElay:REFerence:TRACe	チャンネル間ディレイ測定の基準波形のトレースを設定 / 問い合わせします。	5-147
:MEASure:TRACe<x>:AREA<x>:DElay:SOURce	チャンネル間ディレイ測定の基準を設定 / 問い合わせします。	5-148
:MEASure:TRACe<x>:AREA<x>:DPRoximal?	ディスタル / プロキシマル値に関するすべての設定値を問い合わせます。	5-148
:MEASure:TRACe<x>:AREA<x>:DPRoximal:MODE	ディスタル / プロキシマル値の単位を設定 / 問い合わせします。	5-148
:MEASure:TRACe<x>:AREA<x>:DPRoximal:PERCent	ディスタル / プロキシマル値を % で設定 / 問い合わせします。	5-148
:MEASure:TRACe<x>:AREA<x>:DPRoximal:UNIT	ディスタル / プロキシマル値を UNIT で設定 / 問い合わせします。	5-148
:MEASure:TRACe<x>:AREA<x>:METHod	波形パラメータの自動測定の High/Low レベルの検出方法を設定 / 問い合わせします。	5-148
:MEASure:TRANge<x> (Time Range)	測定範囲を設定 / 問い合わせします。	5-149
:MEASure:WAIT?	タイムアウト付きで自動測定実行の終了を待ちます。	5-149
:MEASure:WINDow<x>	各エリアの測定対象ウィンドウを設定 / 問い合わせします。	5-149

### REFerence グループ

:REFerence<x>?	各リファレンスに関するすべての設定値を問い合わせます。	5-150
:REFerence<x>:DISPlay	各リファレンスの表示の ON/OFF を設定 / 問い合わせします。	5-150
:REFerence<x>:INVert	各リファレンスの反転表示を設定 / 問い合わせします。	5-150
:REFerence<x>:LABel?	各リファレンスの波形ラベルに関するすべての設定値を問い合わせます。	5-150
:REFerence<x>:LABel[:DEFine]	各リファレンスの波形ラベル名を設定 / 問い合わせします。	5-150
:REFerence<x>:LABel:MODE	各リファレンスの波形ラベル名表示の ON/OFF を設定 / 問い合わせします。	5-150
:REFerence<x>:LOAD	各リファレンスに波形をロードします。	5-150
:REFerence<x>:POSition	各リファレンスの垂直ポジションを設定 / 問い合わせします。	5-150
:REFerence<x>:SELect	各演算チャンネルに割り当てる波形 (演算 / リファレンス) を設定 / 問い合わせします。	5-150
:REFerence<x>:SVALue (Scale VALUE)	各リファレンスのスケール値表示の ON/OFF を設定 / 問い合わせします。	5-150

### SEARch グループ

:SEARch<x>?	サーチ機能に関するすべての設定値を問い合わせます。	5-151
:SEARch<x>:ABORt	サーチを中止します。	5-151
:SEARch<x>:CLOCK?	クロックチャンネルに関するすべての設定値を問い合わせます。	5-151
:SEARch<x>:CLOCK:POLarity	クロックチャンネルの極性を設定 / 問い合わせします。	5-151

コマンド	機能	ページ
:SEARCh<x>:CLOCK:SOURCE	サーチのクロックトレースを設定 / 問い合わせします。	5-151
:SEARCh<x>:DECimation	スキップモードの間引き検出を設定 / 問い合わせします。	5-151
:SEARCh<x>:EXECute	サーチを実行します。オーバーラップコマンドです。	5-151
:SEARCh<x>:HOLDoff	ホールドオフ検出を設定 / 問い合わせします。	5-152
:SEARCh<x>:LOGic	サーチロジックを設定 / 問い合わせします。	5-152
:SEARCh<x>:MARK	サーチマークの ON/OFF を設定 / 問い合わせします。	5-152
:SEARCh<x>:POLarity	サーチ極性を設定 / 問い合わせします。	5-152
:SEARCh<x>:SElect	サーチ機能での検出波形番号を設定し、検出波形番号に相当する位置を問い合わせます。	5-152
:SEARCh<x>:SElect? MAXimum	サーチ機能での検出回数を問い合わせます。	5-152
:SEARCh<x>:SLOGic?	ロジックサーチに関するすべての設定値を問い合わせます。	5-153
:SEARCh<x>:SLOGic:CLOCK?	ロジックサーチのクロックに関するすべての設定値を問い合わせます。	5-153
:SEARCh<x>:SLOGic:CLOCK:POLarity	ロジックサーチのクロックの極性を設定 / 問い合わせします。	5-153
:SEARCh<x>:SLOGic:CLOCK:SOURCE	ロジックサーチのクロックのソースを設定 / 問い合わせします。	5-153
:SEARCh<x>:SLOGic:POLarity	ロジックサーチの極性を設定 / 問い合わせします。	5-153
:SEARCh<x>:SLOGic:SOURCE	ロジックサーチのソースを設定 / 問い合わせします。	5-154
:SEARCh<x>:SLOGic:SPATtern? (Serial Pattern)	ロジックシリアルパターンサーチに関するすべての設定値を問い合わせます。	5-154
:SEARCh<x>:SLOGic:SPATtern:CLOCK?	ロジックシリアルパターンサーチのクロックに関するすべての設定値を問い合わせます。	5-154
:SEARCh<x>:SLOGic:SPATtern:CLOCK:MODE	ロジックシリアルパターンサーチのクロックの有効 / 無効を設定 / 問い合わせします。	5-154
:SEARCh<x>:SLOGic:SPATtern:CLOCK:POLarity	ロジックシリアルパターンサーチのクロックトレースの極性を設定 / 問い合わせします。	5-154
:SEARCh<x>:SLOGic:SPATtern:CLOCK:SOURCE	ロジックシリアルパターンサーチのクロックトレースを設定 / 問い合わせします。	5-154
:SEARCh<x>:SLOGic:SPATtern:CS	ロジックシリアルパターンサーチのチップセレクトの有効 / 無効を設定 / 問い合わせします。	5-154
:SEARCh<x>:SLOGic:SPATtern:LATCh?	ロジックシリアルパターンサーチのラッチに関するすべての設定値を問い合わせます。	5-155
:SEARCh<x>:SLOGic:SPATtern:LATCh:POLarity	ロジックシリアルパターンサーチのラッチトレースの極性を設定 / 問い合わせします。	5-155
:SEARCh<x>:SLOGic:SPATtern:LATCh:TRACe	ロジックシリアルパターンサーチのラッチトレースを設定 / 問い合わせします。	5-155
:SEARCh<x>:SLOGic:SPATtern[SETup]?	ロジックシリアルパターンサーチのセットアップに関するすべての設定値を問い合わせます。	5-155
:SEARCh<x>:SLOGic:SPATtern[SETup]:BITRate	ロジックシリアルパターンサーチのビットレートを設定 / 問い合わせします。	5-155
:SEARCh<x>:SLOGic:SPATtern[SETup]:CLEar	ロジックシリアルパターンサーチのパターンをすべてクリア (Don't care) します。	5-155
:SEARCh<x>:SLOGic:SPATtern[SETup]:DATA?	ロジックシリアルパターンサーチのデータに関するすべての設定値を問い合わせます。	5-155
:SEARCh<x>:SLOGic:SPATtern[SETup]:DATA:ACTive	ロジックシリアルパターンサーチのデータのトレースのアクティブレベルを設定 / 問い合わせします。	5-156
:SEARCh<x>:SLOGic:SPATtern[SETup]:DATA:TRACe	ロジックシリアルパターンサーチのデータのトレースを設定 / 問い合わせします。	5-156
:SEARCh<x>:SLOGic:SPATtern[SETup]:HEXA	ロジックシリアルパターンサーチのパターンを HEXA で設定します。	5-156
:SEARCh<x>:SLOGic:SPATtern[SETup]:PATtern	ロジックシリアルパターンサーチのパターンを BINARY で設定 / 問い合わせします。	5-156
:SEARCh<x>:SLOGic:STATe?	ロジックステートサーチに関するすべての設定値を問い合わせます。	5-156
:SEARCh<x>:SLOGic:STATe:BIT?	ロジックステートサーチのビットに関するすべての設定値を問い合わせます。	5-157
:SEARCh<x>:SLOGic:STATe:BIT:{A<x> B<x> C<x> D<x>}	ロジックステートサーチの各ビットの成立条件を設定 / 問い合わせします。	5-157
:SEARCh<x>:SLOGic:STATe:BIT:CLEar	ロジックステートサーチの各ビットの成立条件をすべてクリア (Don't care) します。	5-157
:SEARCh<x>:SLOGic:STATe:BIT:LOGic	ロジックステートサーチのロジックを設定 / 問い合わせします。	5-157
:SEARCh<x>:SLOGic:STATe:GROup<x>?	ロジックステートサーチの各グループに関するすべての設定値を問い合わせます。	5-157
:SEARCh<x>:SLOGic:STATe:GROup<x>:CLEar	ロジックステートサーチの各グループの成立条件をすべてクリア (Don't care) します。	5-157



## 5.1 コマンド一覧表

コマンド	機能	ページ
:SEARCH<x>:SLOGic:STATe: GROUp<x>:CONDition	ロジックステートサーチの各グループの判定条件を設定 / 問い合わせします。	5-157
:SEARCH<x>:SLOGic:STATe: GROUp<x>:DATA<x>	ロジックステートサーチの各グループのデータの比較データを設定 / 問い合わせします。	5-158
:SEARCH<x>:SLOGic:STATe: GROUp<x>:HEXA	ロジックステートサーチの各グループの成立条件を HEXA で設定します。	5-158
:SEARCH<x>:SLOGic:STATe: GROUp<x>:PATTern	ロジックステートサーチの各グループの成立条件を BINARY で設定 / 問い合わせします。	5-158
:SEARCH<x>:SLOGic:STATe: GROUp<x>:SYMBOL	ロジックステートサーチの各グループのシンボルアイテムを設定します。	5-158
:SEARCH<x>:SLOGic:STATe:TYPE	ロジックステートサーチの設定方法を設定 / 問い合わせします。	5-158
:SEARCH<x>:SLOGic:WIDTh?	ロジックパルス幅サーチのすべての設定値を問い合わせます。	5-158
:SEARCH<x>:SLOGic:WIDTh:MODE	ロジックパルス幅サーチの判定モードを設定 / 問い合わせします。	5-159
:SEARCH<x>:SLOGic:WIDTh:TIME<x>	ロジックパルス幅サーチのパルス幅を設定 / 問い合わせします。	5-159
:SEARCH<x>:SLOGic:WIDTh:TYPE	ロジックパルス幅サーチタイプを設定 / 問い合わせします。	5-159
:SEARCH<x>:SMODE	スキップモードを設定 / 問い合わせします。	5-159
:SEARCH<x>:SPATtern? (Serial Pattern)	シリアルパターンサーチに関するすべての設定値を問い合わせます。	5-159
:SEARCH<x>:SPATtern:CLOCK?	シリアルパターンサーチのクロックに関するすべての設定値を問い合わせます。	5-159
:SEARCH<x>:SPATtern:CLOCK:MODE	シリアルパターンサーチのクロックの有効 / 無効を設定 / 問い合わせします。	5-159
:SEARCH<x>:SPATtern:CLOCK: POLarity	シリアルパターンサーチのクロックトレースの極性を設定 / 問い合わせします。	5-159
:SEARCH<x>:SPATtern:CLOCK: SOURCE	シリアルパターンサーチのクロックトレースを設定 / 問い合わせします。	5-159
:SEARCH<x>:SPATtern:CS	シリアルパターンサーチのチップセレクトの有効 / 無効を設定 / 問い合わせします。	5-160
:SEARCH<x>:SPATtern:LATCH?	シリアルパターンサーチのラッチに関するすべての設定値を問い合わせます。	5-160
:SEARCH<x>:SPATtern:LATCH: POLarity	シリアルパターンサーチのラッチトレースの極性を設定 / 問い合わせします。	5-160
:SEARCH<x>:SPATtern:LATCH:TRACe	シリアルパターンサーチのラッチトレースを設定 / 問い合わせします。	5-160
:SEARCH<x>:SPATtern:SETUp?	シリアルパターンサーチのセットアップに関するすべての設定値を問い合わせます。	5-160
:SEARCH<x>:SPATtern[:SETUp]: BITRate	シリアルパターンサーチのビットレートを設定 / 問い合わせします。	5-160
:SEARCH<x>:SPATtern[:SETUp]: CLEAR	シリアルパターンサーチのパターンをすべてクリア (Don't care) します。	5-160
:SEARCH<x>:SPATtern[:SETUp]: DATA?	シリアルパターンサーチのデータに関するすべての設定値を問い合わせます。	5-160
:SEARCH<x>:SPATtern[:SETUp]: DATA:ACTive	シリアルパターンサーチのデータのトレースのアクティブレベルを設定 / 問い合わせします。	5-160
:SEARCH<x>:SPATtern[:SETUp]: DATA:TRACe	シリアルパターンサーチのデータのトレースを設定 / 問い合わせします。	5-161
:SEARCH<x>:SPATtern[:SETUp]: HEXA	シリアルパターンサーチのパターンを HEXA で設定します。	5-161
:SEARCH<x>:SPATtern[:SETUp]: PATTern	シリアルパターンサーチのパターンを BINARY で設定 / 問い合わせします。	5-161
:SEARCH<x>:SPOINT	サーチ開始位置を設定 / 問い合わせします。	5-161
:SEARCH<x>:STRace	サーチ対象トレースを設定 / 問い合わせします。	5-161
:SEARCH<x>:TRACe<x>?	各トレースのサーチ条件に関するすべての設定値を問い合わせます。	5-161
:SEARCH<x>:TRACe<x>:CONDition	各トレースの成立条件を設定 / 問い合わせします。	5-161
:SEARCH<x>:TRACe<x>:HYSTeresis	各トレースのヒステリシスを設定 / 問い合わせします。	5-161
:SEARCH<x>:TRACe<x>:LEVel	各トレースのしきい値 (Threshold) レベルを設定 / 問い合わせします。	5-162
:SEARCH<x>:TYPE	サーチタイプを設定 / 問い合わせします。	5-162
:SEARCH<x>:WIDTh?	パルス幅サーチに関するすべての設定値を問い合わせます。	5-162
:SEARCH<x>:WIDTh:MODE	パルス幅判定モードを設定 / 問い合わせします。	5-162
:SEARCH<x>:WIDTh:TIME<x>	パルス幅サーチのパルス幅を設定 / 問い合わせします。	5-162
:SEARCH<x>:WIDTh:TYPE	パルス幅サーチタイプを設定 / 問い合わせします。	5-162
<b>SNAP グループ</b>		
:SNAP	スナップショットを実行します。	5-163

コマンド	機能	ページ
<b>SStart グループ</b>		
:SStart?	トリガモードをシングルにして START し、指定時間内に STOP した場合にその時点で 0 を返します。	5-164
<b>START グループ</b>		
:START	波形の取り込みをスタートします。	5-165
<b>STATus グループ</b>		
:STATus?	通信のステータス機能に関連する設定をすべて問い合わせます。	5-166
:STATus:CONDition?	状態レジスタの内容の問い合わせます。	5-166
:STATus:EESSE	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。	5-166
:STATus:EESR?	拡張イベントレジスタの内容の問い合わせ、レジスタをクリアします。	5-166
:STATus:ERRor?	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) を問い合わせます。	5-166
:STATus:FILTer<x>	遷移フィルタを設定 / 問い合わせします。	5-166
:STATus:QENable	エラー以外のメッセージをエラーキューに格納するかしないかを設定 / 問い合わせします。	5-166
:STATus:QMEssage	「:STATus:ERRor?」の応答にメッセージ内容を付けるか付けないかを設定 / 問い合わせします。	5-166
:STATus:SPOLL? (Serial Poll)	シリアルポールを実行します。	5-166
<b>STOP グループ</b>		
:STOP	波形の取り込みをストップします。	5-167
<b>SYSTEM グループ</b>		
:SYStem?	システムに関するすべての設定値を問い合わせます。	5-168
:SYStem:CLICk	クリック音の ON/OFF を設定 / 問い合わせします。	5-168
:SYStem:CLOCk?	日付・時刻・GMT からの時差に関するすべての設定値を問い合わせます。	5-168
:SYStem:CLOCk:DTIME	日付・時刻・GMT からの時差を設定 / 問い合わせします。	5-168
:SYStem:CLOCk:MODE	日付・時刻・GMT からの時差表示の ON/OFF を設定 / 問い合わせします。	5-168
:SYStem:FORMat:IMEMory[:EXECute]	内部メモリをフォーマットします。	5-168
:SYStem:FORMat:IHDD[:EXECute]	内蔵ハードディスクをフォーマットします。	5-168
:SYStem:FORMat:SDElete[:EXECute] (Sure Delete)	内部メモリを消去し、フォーマットします。	5-168
:SYStem:LANGuage	メッセージの言語を設定 / 問い合わせします。	5-168
:SYStem:MFSize	メニューのフォントサイズを設定 / 問い合わせします。	5-168
:SYStem:MLANGuage	メニューの言語を設定 / 問い合わせします。	5-168
:SYStem:OVERview	システム情報を表示します。	5-168
:SYStem:USBKeyboard	USB キーボードの種類を設定 / 問い合わせします。	5-168
<b>TELecomtest グループ</b>		
:TELecomtest?	テレコムテストに関するすべての設定値を問い合わせます。	5-169
:TELecomtest:CATegory	テレコムテストの種類を設定 / 問い合わせします。	5-169
:TELecomtest:DISPlay	テレコムテストの表示の ON/OFF を設定 / 問い合わせします。	5-169
:TELecomtest:EYEPattern?	アイパターンに関するすべての設定値を問い合わせます。	5-169
:TELecomtest:EYEPattern:ALL	アイパターンのパラメータのすべてを ON/OFF します。	5-169
:TELecomtest:EYEPattern:<パラメータ>?	アイパターンの波形パラメータに関するすべての設定値を問い合わせます。	5-170
:TELecomtest:EYEPattern:<パラメータ>:STATE	アイパターンの波形パラメータの ON/OFF を設定 / 問い合わせします。	5-170
:TELecomtest:EYEPattern:<パラメータ>:VALue?	アイパターンの波形パラメータの値を問い合わせます。	5-170
:TELecomtest:EYEPattern:TLEVels?	アイパターンのしきい値 (Threshold) レベルに関するすべての設定値を問い合わせます。	5-170
:TELecomtest:EYEPattern:TLEVels:MODE	アイパターンのしきい値 (Threshold) レベルの単位を設定 / 問い合わせします。	5-170
:TELecomtest:EYEPattern:TLEVels:PERCent	アイパターンのしきい値 (Threshold) レベルを % で設定 / 問い合わせします。	5-170

## 5.1 コマンド一覧表

コマンド	機能	ページ
:TELeComtest:EYEPattern: TLEVels:UNIT	アイパターンのしきい値 (Threshold) レベルを UNIT で設定 / 問い合わせし ます。	5-170
:TELeComtest:EYEPattern:VDARK	アイパターンのダークレベル (ゼロライトレベル) を設定 / 問い合わせしま す。	5-171
:TELeComtest:MASK?	マスクテストに関するすべての設定値を問い合わせます。	5-171
:TELeComtest:MASK:ELEMeNt<x>?	マスクテストに使用する各エレメントに関するすべての設定値を問い合わ せします。	5-171
:TELeComtest:MASK:ELEMeNt<x>: ALL	各エレメントのすべてのアイテムを一斉に ON/OFF します。	5-171
:TELeComtest:MASK:ELEMeNt<x>: PSPCount? (Sample Point Count %)	各エレメントのサンプルデータ数に対するエラー率に関する設定値を問い合 わせします。	5-171
:TELeComtest:MASK:ELEMeNt<x>: PSPCount:STATe	各エレメントのサンプルデータ数に対するエラー率の測定の ON/OFF を設 定 / 問い合わせします。	5-171
:TELeComtest:MASK:ELEMeNt<x>: PSPCount:VALue?	各エレメントのサンプルデータ数に対するエラー率を問い合わせます。	5-171
:TELeComtest:MASK:ELEMeNt<x>: PWCount? (Wave Count %)	各エレメントのアクイジション回数に対するエラー率に関する設定値を問い 合わせます。	5-171
:TELeComtest:MASK:ELEMeNt<x>: PWCount:STATe	各エレメントのアクイジション回数に対するエラー率の測定の ON/OFF を 設定 / 問い合わせします。	5-172
:TELeComtest:MASK:ELEMeNt<x>: PWCount:VALue?	各エレメントのアクイジション回数に対するエラー率を問い合わせます。	5-172
:TELeComtest:MASK:ELEMeNt<x>: SPCount? (Sample Point Count)	各エレメントのエラーになったサンプルデータ数に関する設定値を問い合わ せします。	5-172
:TELeComtest:MASK:ELEMeNt<x>: SPCount:STATe	各エレメントのエラーになったサンプルデータ数の測定の ON/OFF を設定 / 問い合わせします。	5-172
:TELeComtest:MASK:ELEMeNt<x>: SPCount:VALue?	各エレメントのエラーになったサンプルデータ数を問い合わせします。	5-172
:TELeComtest:MASK:ELEMeNt<x>: WCount? (Wave Count)	各エレメントのエラーになったアクイジション回数に関する設定値を問い合 わせします。	5-172
:TELeComtest:MASK:ELEMeNt<x>: WCount:STATe	各エレメントのエラーになったアクイジション回数の測定の ON/OFF を設 定 / 問い合わせします。	5-172
:TELeComtest:MASK:ELEMeNt<x>: WCount:VALue?	各エレメントのエラーになったアクイジション回数を問い合わせします。	5-172
:TELeComtest:MMODE	マスク表示の ON/OFF を設定 / 問い合わせします。	5-172
:TELeComtest:TRACe	テレコムテストの対象トレースを設定 / 問い合わせします。	5-173
:TELeComtest:TRANge (Time Range)	テレコムテストの測定範囲を設定 / 問い合わせします。	5-173
:TELeComtest:WINDow	テレコムテストの測定対象ウィンドウを設定 / 問い合わせします。	5-173

### TIMebase グループ

:TIMebase?	タイムベースに関するすべての設定値を問い合わせます。	5-174
:TIMebase:SRATE? (Sample RATE)	サンプルレートを問い合わせます。	5-174
:TIMebase:TDIV	T/div 値を設定 / 問い合わせします。	5-174

### TRIGger グループ

:TRIGger?	トリガに関するすべての設定値を問い合わせます。	5-175
:TRIGger:ACTIon?	アクションオントリガに関するすべての設定値を問い合わせます。	5-175
:TRIGger:ACTIon:ACQCount	アクションオントリガのアクション回数を設定 / 問い合わせします。	5-175
:TRIGger:ACTIon:BUZZer	アクション時に警告音を鳴らす (ON) / 鳴らさない (OFF) を設定 / 問い合わ せします。	5-175
:TRIGger:ACTIon:HCOpy	アクション時に画面イメージデータを出力する (ON) / しない (OFF) を設定 / 問い合わせします。	5-175
:TRIGger:ACTIon:MAIL?	アクション時のメール送信に関するすべての設定値を問い合わせます。	5-175
:TRIGger:ACTIon:MAIL:INTerval	アクション時にメール送信する間隔を設定 / 問い合わせします。	5-176
:TRIGger:ACTIon:MAIL:MODE	アクション時にメール送信する (ON) / しない (OFF) を設定 / 問い合わ せします。	5-176
:TRIGger:ACTIon:MODE	アクションオントリガのモードを設定 / 問い合わせします。	5-176
:TRIGger:ACTIon:SAVE	アクション時に波形データをメディアに保存する (ON) / しない (OFF) を設定 / 問い合わせします。	5-176
:TRIGger:ACTIon:STARt	アクションオントリガを開始します。	5-176
:TRIGger:ACTIon:STOP	アクションオントリガを中止します。	5-176
:TRIGger:CLOCK?	クロックチャンネルに関するすべての設定値を問い合わせます。	5-176
:TRIGger:CLOCK:POLarity	クロックチャンネルの極性を設定 / 問い合わせします。	5-176
:TRIGger:CLOCK:SOURce	クロックチャンネルの対象波形を設定 / 問い合わせします。	5-176

コマンド	機能	ページ
:TRIGger:DElay?	トリガディレイに関するすべての設定値を問い合わせます。	5-176
:TRIGger:DElay:EDGecount?	トリガディレイのエッジカウントに関するすべての設定値を問い合わせます。	5-176
:TRIGger:DElay:EDGecount:COUNT	トリガディレイのエッジカウントの回数を設定 / 問い合わせします。	5-176
:TRIGger:DElay:MODE	トリガディレイの ON/OFF を設定 / 問い合わせします。	5-177
:TRIGger:DElay:POLarity	トリガディレイのエッジ極性を設定 / 問い合わせします。	5-177
:TRIGger:DElay:SOURce	トリガディレイのエッジソースを設定 / 問い合わせします。	5-177
:TRIGger:DElay:TIME	トリガディレイの遅延時間を設定 / 問い合わせします。	5-177
:TRIGger:DElay:TYPE	トリガディレイの種類を設定 / 問い合わせします。	5-177
:TRIGger:EINterval?	イベントインターバルに関するすべての設定値を問い合わせます。	5-177
:TRIGger:EINterval:EVENT<x>?	各イベントに関するすべての設定値を問い合わせます。	5-178
:TRIGger:EINterval:EVENT<x>:CANBus?	各イベントの CAN バス信号トリガに関するすべての設定値を問い合わせます。	5-178
:TRIGger:EINterval:EVENT<x>:CANBus:ACK	CAN バス信号トリガの ACK 条件を設定 / 問い合わせします。	5-179
:TRIGger:EINterval:EVENT<x>:CANBus:BRATe	CAN バス信号トリガのビットレート (データ転送速度) を設定 / 問い合わせします。	5-179
:TRIGger:EINterval:EVENT<x>:CANBus:DATA?	CAN バス信号トリガのデータに関するすべての設定値を問い合わせます。	5-179
:TRIGger:EINterval:EVENT<x>:CANBus:DATA:BORDer	CAN バス信号トリガのデータのバイトオーダを設定 / 問い合わせします。	5-179
:TRIGger:EINterval:EVENT<x>:CANBus:DATA:CONDition	CAN バス信号トリガのデータ条件を設定 / 問い合わせします。	5-179
:TRIGger:EINterval:EVENT<x>:CANBus:DATA:DATA<x>	CAN バス信号トリガのデータの比較データを設定 / 問い合わせします。	5-180
:TRIGger:EINterval:EVENT<x>:CANBus:DATA:DLC	CAN バス信号トリガのデータの有効バイト数 (DLC) を設定 / 問い合わせします。	5-180
:TRIGger:EINterval:EVENT<x>:CANBus:DATA:HEXA	CAN バス信号トリガのデータを HEXA で設定します。	5-180
:TRIGger:EINterval:EVENT<x>:CANBus:DATA:MSBLsb	CAN バス信号トリガのデータの MSB/LSB のビットを設定 / 問い合わせします。	5-180
:TRIGger:EINterval:EVENT<x>:CANBus:DATA:PATtern	CAN バス信号トリガのデータを BINARY で設定 / 問い合わせします。	5-180
:TRIGger:EINterval:EVENT<x>:CANBus:DATA:SIGN	CAN バス信号トリガのデータの符号を設定 / 問い合わせします。	5-181
:TRIGger:EINterval:EVENT<x>:CANBus:IDEXt?	CAN バス信号トリガの拡張フォーマットの ID に関するすべての設定値を問い合わせます。	5-181
:TRIGger:EINterval:EVENT<x>:CANBus:IDEXt:HEXA	CAN バス信号トリガの拡張フォーマットの ID を HEXA で設定します。	5-181
:TRIGger:EINterval:EVENT<x>:CANBus:IDEXt:PATtern	CAN バス信号トリガの拡張フォーマットの ID を BINARY で設定 / 問い合わせします。	5-181
:TRIGger:EINterval:EVENT<x>:CANBus:IDOR?	CAN バス信号トリガの OR 条件に関するすべての設定値を問い合わせます。	5-181
:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>?	CAN バス信号トリガの OR 条件の各 ID に関するすべての設定値を問い合わせます。	5-182
:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:ACK	CAN バス信号トリガの OR 条件の各 ACK 条件を設定 / 問い合わせします。	5-182
:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA?	CAN バス信号トリガの OR 条件の各データに関するすべての設定値を問い合わせます。	5-182
:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:BORDer	CAN バス信号トリガの OR 条件の各データのバイトオーダを設定 / 問い合わせします。	5-182
:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:CONDition	CAN バス信号トリガの OR 条件の各データ条件を設定 / 問い合わせします。	5-182
:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:DATA<x>	CAN バス信号トリガの OR 条件の各データの比較データを設定 / 問い合わせします。	5-183
:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:DLC	CAN バス信号トリガの OR 条件の各データの有効バイト数 (DLC) を設定 / 問い合わせします。	5-183
:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:HEXA	CAN バス信号トリガの OR 条件の各データを HEXA で設定します。	5-183
:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:MSBLsb	CAN バス信号トリガの OR 条件の各データの MSB/LSB のビットを設定 / 問い合わせします。	5-183
:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:PATtern	CAN バス信号トリガの OR 条件の各データを BINARY で設定 / 問い合わせします。	5-183

## 5.1 コマンド一覧表

コマンド	機能	ページ
:TRIGger:EINterval:EVENT<x>: CANBus:IDOR:ID<x>:DATA:SIGN	CANバス信号トリガのOR条件の各データの符号を設定/問い合わせします。	5-184
:TRIGger:EINterval:EVENT<x>: CANBus:IDOR:ID<x>:FORMat	CANバス信号トリガのOR条件の各メッセージフォーマット(標準/拡張)を設定/問い合わせします。	5-184
:TRIGger:EINterval:EVENT<x>: CANBus:IDOR:ID<x>:IDEXt?	CANバス信号トリガのOR条件の各拡張フォーマットのIDに関するすべての設定値を問い合わせます。	5-184
:TRIGger:EINterval:EVENT<x>: CANBus:IDOR:ID<x>:IDEXt:HEXA	CANバス信号トリガのOR条件の各拡張フォーマットのIDをHEXAで設定します。	5-184
:TRIGger:EINterval:EVENT<x>: CANBus:IDOR:ID<x>:IDEXt:PATtern	CANバス信号トリガのOR条件の各拡張フォーマットのIDをBINARYで設定/問い合わせします。	5-184
:TRIGger:EINterval:EVENT<x>: CANBus:IDOR:ID<x>:IDSTd?	CANバス信号トリガのOR条件の各標準フォーマットのIDに関するすべての設定値を問い合わせます。	5-184
:TRIGger:EINterval:EVENT<x>: CANBus:IDOR:ID<x>:IDSTd:HEXA	CANバス信号トリガのOR条件の各標準フォーマットのIDをHEXAで設定します。	5-185
:TRIGger:EINterval:EVENT<x>: CANBus:IDOR:ID<x>:IDSTd:PATtern	CANバス信号トリガのOR条件の各標準フォーマットのIDをBINARYで設定/問い合わせします。	5-185
:TRIGger:EINterval:EVENT<x>: CANBus:IDOR:ID<x>:MODE	CANバス信号トリガのOR条件の各条件の有効(1)/無効(0)を設定/問い合わせします。	5-185
:TRIGger:EINterval:EVENT<x>: CANBus:IDOR:ID<x>:RTR	CANバス信号トリガのOR条件の各RTRを設定/問い合わせします。	5-185
:TRIGger:EINterval:EVENT<x>: CANBus:IDSTd?	CANバス信号トリガの標準フォーマットのIDに関するすべての設定値を問い合わせます。	5-185
:TRIGger:EINterval:EVENT<x>: CANBus:IDSTd:HEXA	CANバス信号トリガの標準フォーマットのIDをHEXAで設定します。	5-185
:TRIGger:EINterval:EVENT<x>: CANBus:IDSTd:PATtern	CANバス信号トリガの標準フォーマットのIDをBINARYで設定/問い合わせします。	5-185
:TRIGger:EINterval:EVENT<x>: CANBus:MODE	CANバス信号トリガのモードを設定/問い合わせします。	5-186
:TRIGger:EINterval:EVENT<x>: CANBus:REcessive	CANバス信号トリガのリセッスレベル(バスレベル)を設定/問い合わせします。	5-186
:TRIGger:EINterval:EVENT<x>: CANBus:RTR	CANバス信号トリガのRTRを設定/問い合わせします。	5-186
:TRIGger:EINterval:EVENT<x>: CANBus:SOURce	CANバス信号トリガのトリガソースを設定/問い合わせします。	5-186
:TRIGger:EINterval:EVENT<x>: CANBus:SPOint	CANバス信号トリガのサンプルポイントを設定/問い合わせします。	5-186
:TRIGger:EINterval:EVENT<x>: CLOCK?	各イベントのクロックチャンネルに関するすべての設定値を問い合わせます。	5-186
:TRIGger:EINterval:EVENT<x>: CLOCK:POLarity	各イベントのクロックチャンネルの極性を設定/問い合わせします。	5-187
:TRIGger:EINterval:EVENT<x>: CLOCK:SOURce	各イベントのクロックチャンネルの対象波形を設定/問い合わせします。	5-187
:TRIGger:EINterval:EVENT<x>: ESTate?	Edge/Stateトリガに関するすべての設定値を問い合わせます。	5-187
:TRIGger:EINterval:EVENT<x>: ESTate:POLarity	Edge/Stateトリガの極性を設定/問い合わせします。	5-187
:TRIGger:EINterval:EVENT<x>: ESTate:SOURce	Edge/Stateトリガのトリガソースを設定/問い合わせします。	5-187
:TRIGger:EINterval:EVENT<x>: I2CBus?	各イベントのI <sup>2</sup> Cバstriガに関するすべての設定値を問い合わせます。	5-188
:TRIGger:EINterval:EVENT<x>: I2CBus:ADATa?	I <sup>2</sup> Cバstriガのアドレスに関するすべての設定値を問い合わせます。	5-188
:TRIGger:EINterval:EVENT<x>: I2CBus:ADATa:BIT10address?	I <sup>2</sup> Cバstriガの10bitアドレスに関するすべての設定値を問い合わせます。	5-188
:TRIGger:EINterval:EVENT<x>: I2CBus:ADATa:BIT10address:HEXA	I <sup>2</sup> Cバstriガの10bitアドレスをHEXAで設定します。	5-188
:TRIGger:EINterval:EVENT<x>: I2CBus:ADATa:BIT10address: PATtern	I <sup>2</sup> Cバstriガの10bitアドレスをBINARYで設定/問い合わせします。	5-188
:TRIGger:EINterval:EVENT<x>: I2CBus:ADATa:BIT7Address?	I <sup>2</sup> Cバstriガの7bitアドレスに関するすべての設定値を問い合わせます。	5-188
:TRIGger:EINterval:EVENT<x>: I2CBus:ADATa:BIT7Address:HEXA	I <sup>2</sup> Cバstriガの7bitアドレスをHEXAで設定します。	5-189

コマンド	機能	ページ
:TRIGger:EINterval:EVENT<x>: I2CBus:ADATa:BIT7Address: PATtern	I <sup>2</sup> C バストリガの 7bit アドレスを BINARY で設定 / 問い合わせします。	5-189
:TRIGger:EINterval:EVENT<x>: I2CBus:ADATa:BIT7APsub?	I <sup>2</sup> C バストリガの 7bit + Sub アドレスに関するすべての設定値を問い合わせ	5-189
:TRIGger:EINterval:EVENT<x>: I2CBus:ADATa:BIT7APsub:ADDRESS?	I <sup>2</sup> C バストリガの 7bit + Sub アドレスの 7bit アドレスに関するすべての設	5-189
:TRIGger:EINterval:EVENT<x>: I2CBus:ADATa:BIT7APsub:ADDRESS: HEXA	定値を問い合わせます。 I <sup>2</sup> C バストリガの 7bit + Sub アドレスの 7bit アドレスを HEXA で設定します。	5-189
:TRIGger:EINterval:EVENT<x>: I2CBus:ADATa:BIT7APsub:ADDRESS: PATtern	I <sup>2</sup> C バストリガの 7bit + Sub アドレスの 7bit アドレスを BINARY で設定 / 問	5-189
:TRIGger:EINterval:EVENT<x>: I2CBus:ADATa:BIT7APsub: SADDRESS?	い合わせします。 I <sup>2</sup> C バストリガの 7bit + Sub アドレスの Sub アドレスに関するすべての設	5-189
:TRIGger:EINterval:EVENT<x>: I2CBus:ADATa:BIT7APsub: SADDRESS:HEXA	定値を問い合わせます。 I <sup>2</sup> C バストリガの 7bit + Sub アドレスの Sub アドレスを HEXA で設定します。	5-189
:TRIGger:EINterval:EVENT<x>: I2CBus:ADATa:BIT7APsub: SADDRESS:PATtern	I <sup>2</sup> C バストリガの 7bit + Sub アドレスの Sub アドレスを BINARY で設定 / 問	5-190
:TRIGger:EINterval:EVENT<x>: I2CBus:ADATa:TYPE	い合わせします。 I <sup>2</sup> C バストリガのアドレスの種類を設定 / 問い合わせします。	5-190
:TRIGger:EINterval:EVENT<x>: I2CBus:CLOCK?	I <sup>2</sup> C バストリガのクロックに関するすべての設定値を問い合わせます。	5-190
:TRIGger:EINterval:EVENT<x>: I2CBus:CLOCK:SOURce	I <sup>2</sup> C バストリガのクロックトレースを設定 / 問い合わせします。	5-190
:TRIGger:EINterval:EVENT<x>: I2CBus:DATA?	I <sup>2</sup> C バストリガのデータに関するすべての設定値を問い合わせます。	5-190
:TRIGger:EINterval:EVENT<x>: I2CBus:DATA:BYTE	I <sup>2</sup> C バストリガの設定データ数を設定 / 問い合わせします。	5-190
:TRIGger:EINterval:EVENT<x>: I2CBus:DATA:CONDition	I <sup>2</sup> C バストリガのデータの判定方法 (一致 / 不一致) を設定 / 問い合わせしま	5-191
:TRIGger:EINterval:EVENT<x>: I2CBus:DATA:DPOSITION	す。 I <sup>2</sup> C バストリガのデータのパターン比較する位置を設定 / 問い合わせします。	5-191
:TRIGger:EINterval:EVENT<x>: I2CBus:DATA:HEXA<x>	I <sup>2</sup> C バストリガのデータを HEXA で設定します。	5-191
:TRIGger:EINterval:EVENT<x>: I2CBus:DATA:MODE	I <sup>2</sup> C バストリガのデータ条件の有効 / 無効を設定 / 問い合わせします。	5-191
:TRIGger:EINterval:EVENT<x>: I2CBus:DATA:PATtern<x>	I <sup>2</sup> C バストリガのデータを BINARY で設定 / 問い合わせします。	5-191
:TRIGger:EINterval:EVENT<x>: I2CBus:DATA:PMODE	I <sup>2</sup> C バストリガのデータのパターン比較先頭位置モードを設定 / 問い合わせ	5-191
:TRIGger:EINterval:EVENT<x>: I2CBus:DATA:SOURce	します。 I <sup>2</sup> C バストリガのデータトレースを設定 / 問い合わせします。	5-192
:TRIGger:EINterval:EVENT<x>: I2CBus:GCALL?	I <sup>2</sup> C バストリガのジェネラルコールに関するすべての設定値を問い合わせま	5-192
:TRIGger:EINterval:EVENT<x>: I2CBus:GCALL:BIT7maddress?	す。 I <sup>2</sup> C バストリガのジェネラルコールの 7bit マスタアドレスに関するすべての	5-192
:TRIGger:EINterval:EVENT<x>: I2CBus:GCALL:BIT7maddress:HEXA	設定値を問い合わせます。 I <sup>2</sup> C バストリガのジェネラルコールの 7bit マスタアドレスを HEXA で設定し	5-192
:TRIGger:EINterval:EVENT<x>: I2CBus:GCALL:BIT7maddress: PATtern	ます。 I <sup>2</sup> C バストリガのジェネラルコールの 7bit マスタアドレスを BINARY で設定	5-192
:TRIGger:EINterval:EVENT<x>: I2CBus:GCALL:SBYTE (Second Byte)	/ 問い合わせします。 I <sup>2</sup> C バストリガのジェネラルコールのセカンドバイトのタイプを設定 / 問い	5-192
:TRIGger:EINterval:EVENT<x>: I2CBus:MODE	合わせします。 I <sup>2</sup> C バストリガのトリガモードを設定 / 問い合わせします。	5-192
:TRIGger:EINterval:EVENT<x>: I2CBus:NAIGNore?	I <sup>2</sup> C バストリガの NON ACK 無視モードに関するすべての設定値を問い合わ	5-193
:TRIGger:EINterval:EVENT<x>: I2CBus:NAIGNore:HSMODE	せます。 I <sup>2</sup> C バストリガのハイスピードモードで NON ACK を無視する / しないを設	5-193
:TRIGger:EINterval:EVENT<x>: I2CBus:NAIGNore:RACcess	定 / 問い合わせします。 I <sup>2</sup> C バストリガのリードアクセスモードで NON ACK を無視する / しないを	5-193

## 5.1 コマンド一覧表

コマンド	機能	ページ
:TRIGger:EINterval:EVENT<x>: I2Cbus:NAIGnore:SBYTE (Start Byte)	I <sup>2</sup> CバストリガのスタートバイトでNON ACKを無視する/しないを設定/問 い合わせします。	5-193
:TRIGger:EINterval:EVENT<x>: I2Cbus:SBHSmode?	I <sup>2</sup> Cバストリガのスタートバイト/ハイスピードモードに関するすべての設 定値を問い合わせます。	5-193
:TRIGger:EINterval:EVENT<x>: I2Cbus:SBHSmode:TYPE	I <sup>2</sup> Cバストリガのスタートバイト/ハイスピードモードのタイプを設定/問 い合わせします。	5-193
:TRIGger:EINterval:EVENT<x>: LINbus?	各イベントのLINバス信号トリガに関するすべての設定値を問い合わせしま す。	5-193
:TRIGger:EINterval:EVENT<x>: LINbus:BRATE	LINバス信号トリガのビットレート(データ転送速度)を設定/問い合わせ します。	5-194
:TRIGger:EINterval:EVENT<x>: LINbus:SOURce	LINバス信号トリガのトリガソースを設定/問い合わせします。	5-194
:TRIGger:EINterval:EVENT<x>: LOGic?	各イベントのロジックトリガに関するすべての設定値を問い合わせます。	5-194
:TRIGger:EINterval:EVENT<x>: LOGic:CLOCK?	ロジックトリガのクロックに関するすべての設定値を問い合わせます。	5-194
:TRIGger:EINterval:EVENT<x>: LOGic:CLOCK:POLarity	ロジックトリガのクロックの極性を設定/問い合わせします。	5-195
:TRIGger:EINterval:EVENT<x>: LOGic:CLOCK:SOURce	ロジックトリガのクロックのソースを設定/問い合わせします。	5-195
:TRIGger:EINterval:EVENT<x>: LOGic:ESTate?	ロジックのEdge/State トリガに関するすべての設定値を問い合わせます。	5-195
:TRIGger:EINterval:EVENT<x>: LOGic:ESTate:POLarity	ロジックのEdge/State トリガの極性を設定/問い合わせします。	5-195
:TRIGger:EINterval:EVENT<x>: LOGic:ESTate:SOURce	ロジックのEdge/State トリガのトリガソースを設定/問い合わせします。	5-195
:TRIGger:EINterval:EVENT<x>: LOGic:I2Cbus?	各イベントのロジック I <sup>2</sup> Cバストリガに関するすべての設定値を問い合わせ ます。	5-196
:TRIGger:EINterval:EVENT<x>: LOGic:I2Cbus:ADATa?	ロジック I <sup>2</sup> Cバストリガのアドレスに関するすべての設定値を問い合わせま す。	5-196
:TRIGger:EINterval:EVENT<x>: LOGic:I2Cbus:ADATa: BIT10address?	ロジック I <sup>2</sup> Cバストリガの 10bit アドレスに関するすべての設定値を問 い合わせます。	5-196
:TRIGger:EINterval:EVENT<x>: LOGic:I2Cbus:ADATa: BIT10address:HEXA	ロジック I <sup>2</sup> Cバストリガの 10bit アドレスを HEXA で設定します。	5-196
:TRIGger:EINterval:EVENT<x>: LOGic:I2Cbus:ADATa: BIT10address:PATtern	ロジック I <sup>2</sup> Cバストリガの 10bit アドレスを BINARY で設定/問 い合わせします。	5-197
:TRIGger:EINterval:EVENT<x>: LOGic:I2Cbus:ADATa:BIT7Address?	ロジック I <sup>2</sup> Cバストリガの 7bit アドレスに関するすべての設定値を問 い合わせます。	5-197
:TRIGger:EINterval:EVENT<x>: LOGic:I2Cbus:ADATa:BIT7Address: HEXA	ロジック I <sup>2</sup> Cバストリガの 7bit アドレスを HEXA で設定します。	5-197
:TRIGger:EINterval:EVENT<x>: LOGic:I2Cbus:ADATa:BIT7Address: PATtern	ロジック I <sup>2</sup> Cバストリガの 7bit アドレスを BINARY で設定/問 い合わせします。	5-197
:TRIGger:EINterval:EVENT<x>: LOGic:I2Cbus:ADATa:BIT7APsub?	ロジック I <sup>2</sup> Cバストリガの 7bit + Sub アドレスに関するすべての設定値を 問い合わせます。	5-197
:TRIGger:EINterval:EVENT<x>: LOGic:I2Cbus:ADATa:BIT7APsub: ADDRESS?	ロジック I <sup>2</sup> Cバストリガの 7bit + Sub アドレスの 7bit アドレスに関するす べての設定値を問い合わせます。	5-197
:TRIGger:EINterval:EVENT<x>: LOGic:I2Cbus:ADATa:BIT7APsub: ADDRESS:HEXA	ロジック I <sup>2</sup> Cバストリガの 7bit + Sub アドレスの 7bit アドレスを HEXA で 設定します。	5-198
:TRIGger:EINterval:EVENT<x>: LOGic:I2Cbus:ADATa:BIT7APsub: ADDRESS:PATtern	ロジック I <sup>2</sup> Cバストリガの 7bit + Sub アドレスの 7bit アドレスを BINARY で設定/問い合わせします。	5-198
:TRIGger:EINterval:EVENT<x>: LOGic:I2Cbus:ADATa:BIT7APsub: SADDRESS?	ロジック I <sup>2</sup> Cバストリガの 7bit + Sub アドレスの Sub アドレスに関するす べての設定値を問い合わせます。	5-198
:TRIGger:EINterval:EVENT<x>: LOGic:I2Cbus:ADATa:BIT7APsub: SADDRESS:HEXA	ロジック I <sup>2</sup> Cバストリガの 7bit + Sub アドレスの Sub アドレスを HEXA で 設定します。	5-198

コマンド	機能	ページ
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:ADATa:BIT7APsub: SADdress:PATtern	ロジック I <sup>2</sup> C バストリガの 7bit + Sub アドレスの Sub アドレスを BINARY で設定 / 問い合わせします。	5-198
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:ADATa:TYPE	ロジック I <sup>2</sup> C バストリガのアドレスの種類を設定 / 問い合わせします。	5-198
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:CLOCK?	ロジック I <sup>2</sup> C バストリガのクロックに関するすべての設定値を問い合わせます。	5-199
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:CLOCK:SOURce	ロジック I <sup>2</sup> C バストリガのクロックトレースを設定 / 問い合わせします。	5-199
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:DATA?	ロジック I <sup>2</sup> C バストリガのデータに関するすべての設定値を問い合わせます。	5-199
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:DATA:BYTE	ロジック I <sup>2</sup> C バストリガの設定データ数を設定 / 問い合わせします。	5-199
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:DATA:CONDition	ロジック I <sup>2</sup> C バストリガのデータの判定方法 (一致 / 不一致) を設定 / 問い合わせします。	5-199
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:DATA:DPOSITION	ロジック I <sup>2</sup> C バストリガのデータのパターン比較する位置を設定 / 問い合わせします。	5-199
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:DATA:HEXA<x>	ロジック I <sup>2</sup> C バストリガのデータを HEXA で設定します。	5-200
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:DATA:MODE	ロジック I <sup>2</sup> C バストリガのデータ条件の有効 / 無効を設定 / 問い合わせします。	5-200
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:DATA:PATtern<x>	ロジック I <sup>2</sup> C バストリガのデータを BINARY で設定 / 問い合わせします。	5-200
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:DATA:PMODE	ロジック I <sup>2</sup> C バストリガのデータのパターン比較先頭位置モードを設定 / 問い合わせします。	5-200
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:DATA:SOURce	ロジック I <sup>2</sup> C バストリガのデータトレースを設定 / 問い合わせします。	5-200
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:GCALl?	ロジック I <sup>2</sup> C バストリガのジェネラルコールに関するすべての設定値を問い合わせます。	5-200
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:GCALl: BIT7maddress?	ロジック I <sup>2</sup> C バストリガのジェネラルコールの 7bit マスタアドレスに関するすべての設定値を問い合わせます。	5-201
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:GCALl: BIT7maddress:HEXA	ロジック I <sup>2</sup> C バストリガのジェネラルコールの 7bit マスタアドレスを HEXA で設定します。	5-201
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:GCALl: BIT7maddress:PATtern	ロジック I <sup>2</sup> C バストリガのジェネラルコールの 7bit マスタアドレスを BINARY で設定 / 問い合わせします。	5-201
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:GCALl:SBYTE (Second Byte)	ロジック I <sup>2</sup> C バストリガのジェネラルコールのセカンドバイトのタイプを設定 / 問い合わせします。	5-201
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:MODE	ロジック I <sup>2</sup> C バストリガのトリガモードを設定 / 問い合わせします。	5-201
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:NAIgnore?	ロジック I <sup>2</sup> C バストリガの NON ACK 無視モードに関するすべての設定値を問い合わせます。	5-201
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:NAIgnore:HSMODE	ロジック I <sup>2</sup> C バストリガのハイスピードモードで NON ACK を無視する / しないを設定 / 問い合わせします。	5-202
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:NAIgnore:RACcESS	ロジック I <sup>2</sup> C バストリガのリードアクセスモードで NON ACK を無視する / しないを設定 / 問い合わせします。	5-202
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:NAIgnore:SBYTE (Start Byte)	ロジック I <sup>2</sup> C バストリガのスタートバイトで NON ACK を無視する / しないを設定 / 問い合わせします。	5-202
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:SBHSmode?	ロジック I <sup>2</sup> C バストリガのスタートバイト / ハイスピードモードに関するすべての設定値を問い合わせます。	5-202
:TRIGger:EIInterval:EVENT<x>: LOGic:I2CBus:SBHSmode:TYPE	ロジック I <sup>2</sup> C バストリガのスタートバイト / ハイスピードモードのタイプを設定 / 問い合わせします。	5-202
:TRIGger:EIInterval:EVENT<x>: LOGic:LINBus?	各イベントのロジック LIN バス信号トリガに関するすべての設定値を問い合わせします。	5-202
:TRIGger:EIInterval:EVENT<x>: LOGic:LINBus:BRATe	ロジック LIN バス信号トリガのビットレート (データ転送速度) を設定 / 問い合わせします。	5-202
:TRIGger:EIInterval:EVENT<x>: LOGic:LINBus:SOURce	ロジック LIN バス信号トリガのトリガソースを設定 / 問い合わせします。	5-203
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern? (Serial Pattern)	各イベントのロジックシリアルパターントリガに関するすべての設定値を問い合わせます。	5-203



## 5.1 コマンド一覧表

コマンド	機能	ページ
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern:BITRate	ロジックシリアルパターントリガのビットレートを設定 / 問い合わせし ます。	5-203
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern:CLEAr	ロジックシリアルパターントリガのパターンをすべてクリア (Don't care) し ます。	5-203
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern:CLOCK?	ロジックシリアルパターントリガのクロックに関するすべての設定値を問い 合わせます。	5-203
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern:CLOCK:MODE	ロジックシリアルパターントリガのクロックの有効 / 無効を設定 / 問い合わ せします。	5-204
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern:CLOCK:POLarity	ロジックシリアルパターントリガのクロックトレースの極性を設定 / 問い合 わせします。	5-204
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern:CLOCK:SOURce	ロジックシリアルパターントリガのクロックトレースを設定 / 問い合わせし ます。	5-204
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern:CS	ロジックシリアルパターントリガのチップセレクトの有効 / 無効を設定 / 問 い合わせします。	5-204
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern:DATA?	ロジックシリアルパターントリガのデータに関するすべての設定値を問い合 わせします。	5-204
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern:DATA:ACTive	ロジックシリアルパターントリガのデータのアクティブレベルを設定 / 問い 合わせします。	5-205
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern:DATA:SOURce	ロジックシリアルパターントリガのデータトレースを設定 / 問い合わせし ます。	5-205
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern:HEXA	ロジックシリアルパターントリガのパターンを HEXA で設定します。	5-205
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern:LATCh?	ロジックシリアルパターントリガのラッチに関するすべての設定値を問い合 わせします。	5-205
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern:LATCh:POLarity	ロジックシリアルパターントリガのラッチトレースの極性を設定 / 問い合わ せします。	5-205
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern:LATCh:SOURce	ロジックシリアルパターントリガのラッチトレースを設定 / 問い合わせし ます。	5-206
:TRIGger:EIInterval:EVENT<x>: LOGic:SPATtern:PATtern	ロジックシリアルパターントリガのパターンを BINARY で設定 / 問い合わせ します。	5-206
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus?	各イベントのロジック SPI バストリガに関するすべての設定値を問い合せ ます。	5-206
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus:BITorder	ロジック SPI バストリガのビットオーダーを設定 / 問い合わせします。	5-206
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus:CLOCK?	ロジック SPI バストリガのクロックに関するすべての設定値を問い合わせま す。	5-206
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus:CLOCK:POLarity	ロジック SPI バストリガのクロックトレースの極性を設定 / 問い合わせし ます。	5-207
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus:CLOCK:SOURce	ロジック SPI バストリガのクロックトレースを設定 / 問い合わせします。	5-207
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus:CS?	ロジック SPI バストリガのチップセレクトに関するすべての設定値を問い合 わせします。	5-207
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus:CS:ACTive	ロジック SPI バストリガのチップセレクトのアクティブレベルを設定 / 問い 合わせします。	5-207
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus:CS:SOURce	ロジック SPI バストリガのチップセレクトトレースを設定 / 問い合わせし ます。	5-207
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus:DATA<x>?	ロジック SPI バストリガの各データに関するすべての設定値を問い合わせま す。	5-207
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus:DATA<x>:BYTE	ロジック SPI バストリガの各データの設定データ数を設定 / 問い合わせし ます。	5-208
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus:DATA<x>:CONDition	ロジック SPI バストリガの各データの判定方法 (一致 / 不一致) を設定 / 問 い合わせします。	5-208
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus:DATA<x>:DPOSition	ロジック SPI バストリガの各データのパターン比較先頭位置を設定 / 問い合 わせします。	5-208
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus:DATA<x>:HEXA<x>	ロジック SPI バストリガの各データを HEXA で設定します。	5-208
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus:DATA<x>:PATtern<x>	ロジック SPI バストリガの各データを BINARY で設定 / 問い合わせします。	5-208
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus:DATA<x>:SOURce	ロジック SPI バストリガの各データのトレースを設定 / 問い合わせします。	5-208
:TRIGger:EIInterval:EVENT<x>: LOGic:SPIBus:MODE	ロジック SPI バストリガの結線方式 (3 線式 / 4 線式) を設定 / 問い合わせし ます。	5-209
:TRIGger:EIInterval:EVENT<x>: LOGic:STATe?	ロジックステートトリガに関するすべての設定値を問い合わせます。	5-209

コマンド	機能	ページ
:TRIGger:EINterval:EVENT<x>: LOGic:StAtE:BIT?	ロジックステートトリガのビットに関するすべての設定値を問い合わせます。	5-209
:TRIGger:EINterval:EVENT<x>: LOGic:StAtE:BIT:{A<y> B<y> C<y>  D<y>}	ロジックステートトリガの各ビットの成立条件を設定 / 問い合わせします。	5-209
:TRIGger:EINterval:EVENT<x>: LOGic:StAtE:BIT:CLear	ロジックステートトリガの各ビットの成立条件をすべてクリア (Don't care) します。	5-209
:TRIGger:EINterval:EVENT<x>: LOGic:StAtE:BIT:LOGic	ロジックステートトリガのロジックを設定 / 問い合わせします。	5-210
:TRIGger:EINterval:EVENT<x>: LOGic:StAtE:GROup<x>?	ロジックステートトリガの各グループに関するすべての設定値を問い合わせます。	5-210
:TRIGger:EINterval:EVENT<x>: LOGic:StAtE:GROup<x>:CLear	ロジックステートトリガの各グループの成立条件をすべてクリア (Don't care) します。	5-210
:TRIGger:EINterval:EVENT<x>: LOGic:StAtE:GROup<x>:CONDition	ロジックステートトリガの各グループの判定条件を設定 / 問い合わせします。	5-210
:TRIGger:EINterval:EVENT<x>: LOGic:StAtE:GROup<x>:HEXA	ロジックステートトリガの各グループの成立条件を HEXA で設定します。	5-210
:TRIGger:EINterval:EVENT<x>: LOGic:StAtE:GROup<x>:PATtern	ロジックステートトリガの各グループの成立条件を BINARY で設定 / 問い合わせします。	5-210
:TRIGger:EINterval:EVENT<x>: LOGic:StAtE:GROup<x>:SYMBOL	ロジックステートトリガの各グループのシンボルアイテムを設定します。	5-211
:TRIGger:EINterval:EVENT<x>: LOGic:StAtE:TYPE	ロジックステートトリガの設定方法を設定 / 問い合わせします。	5-211
:TRIGger:EINterval:EVENT<x>: LOGic:WIDTh?	ロジックパルス幅トリガのすべての設定値を問い合わせます。	5-211
:TRIGger:EINterval:EVENT<x>: LOGic:WIDTh:MODE	ロジックパルス幅トリガの判定モードを設定 / 問い合わせします。	5-211
:TRIGger:EINterval:EVENT<x>: LOGic:WIDTh:POLarity	ロジックパルス幅トリガの極性を設定 / 問い合わせします。	5-211
:TRIGger:EINterval:EVENT<x>: LOGic:WIDTh:SOURce	ロジックパルス幅トリガのトリガソースを設定 / 問い合わせします。	5-211
:TRIGger:EINterval:EVENT<x>: LOGic:WIDTh:TIME<x>	ロジックパルス幅トリガのパルス幅を設定 / 問い合わせします。	5-212
:TRIGger:EINterval:EVENT<x>: SPATtern? (Serial Pattern)	各イベントのシリアルパターントリガに関するすべての設定値を問い合わせます。	5-212
:TRIGger:EINterval:EVENT<x>: SPATtern:BITRate	シリアルパターントリガのビットレートを設定 / 問い合わせします。	5-212
:TRIGger:EINterval:EVENT<x>: SPATtern:CLear	シリアルパターントリガのパターンをすべてクリア (Don't care) します。	5-212
:TRIGger:EINterval:EVENT<x>: SPATtern:CLOCK?	シリアルパターントリガのクロックに関するすべての設定値を問い合わせます。	5-212
:TRIGger:EINterval:EVENT<x>: SPATtern:CLOCK:MODE	シリアルパターントリガのクロックの有効 / 無効を設定 / 問い合わせします。	5-212
:TRIGger:EINterval:EVENT<x>: SPATtern:CLOCK:POLarity	シリアルパターントリガのクロックトレースの極性を設定 / 問い合わせします。	5-212
:TRIGger:EINterval:EVENT<x>: SPATtern:CLOCK:SOURce	シリアルパターントリガのクロックトレースを設定 / 問い合わせします。	5-213
:TRIGger:EINterval:EVENT<x>: SPATtern:CS	シリアルパターントリガのチップセレクトの有効 / 無効を設定 / 問い合わせします。	5-213
:TRIGger:EINterval:EVENT<x>: SPATtern:DATA?	シリアルパターントリガのデータに関するすべての設定値を問い合わせます。	5-213
:TRIGger:EINterval:EVENT<x>: SPATtern:DATA:ACTive	シリアルパターントリガのデータのアクティブレベルを設定 / 問い合わせします。	5-213
:TRIGger:EINterval:EVENT<x>: SPATtern:DATA:SOURce	シリアルパターントリガのデータトレースを設定 / 問い合わせします。	5-213
:TRIGger:EINterval:EVENT<x>: SPATtern:HEXA	シリアルパターントリガのパターンを HEXA で設定します。	5-213
:TRIGger:EINterval:EVENT<x>: SPATtern:LATCh?	シリアルパターントリガのラッチに関するすべての設定値を問い合わせます。	5-213
:TRIGger:EINterval:EVENT<x>: SPATtern:LATCh:POLarity	シリアルパターントリガのラッチトレースの極性を設定 / 問い合わせします。	5-214
:TRIGger:EINterval:EVENT<x>: SPATtern:LATCh:SOURce	シリアルパターントリガのラッチトレースを設定 / 問い合わせします。	5-214
:TRIGger:EINterval:EVENT<x>: SPATtern:PATtern	シリアルパターントリガのパターンを BINARY で設定 / 問い合わせします。	5-214

## 5.1 コマンド一覧表

コマンド	機能	ページ
:TRIGger:EINterval:EVENT<x>:SPIBus?	各イベントの SPI バストリガに関するすべての設定値を問い合わせます。	5-214
:TRIGger:EINterval:EVENT<x>:SPIBus:BITorder	SPI バストリガのビットオーダーを設定 / 問い合わせします。	5-214
:TRIGger:EINterval:EVENT<x>:SPIBus:CLOCK?	SPI バストリガのクロックに関するすべての設定値を問い合わせます。	5-214
:TRIGger:EINterval:EVENT<x>:SPIBus:CLOCK:POLarity	SPI バストリガのクロックトレースの極性を設定 / 問い合わせします。	5-215
:TRIGger:EINterval:EVENT<x>:SPIBus:CLOCK:SOURce	SPI バストリガのクロックトレースを設定 / 問い合わせします。	5-215
:TRIGger:EINterval:EVENT<x>:SPIBus:CS?	SPI バストリガのチップセレクトに関するすべての設定値を問い合わせます。	5-215
:TRIGger:EINterval:EVENT<x>:SPIBus:CS:ACTive	SPI バストリガのチップセレクトのアクティブレベルを設定 / 問い合わせします。	5-215
:TRIGger:EINterval:EVENT<x>:SPIBus:CS:SOURce	SPI バストリガのチップセレクトトレースを設定 / 問い合わせします。	5-215
:TRIGger:EINterval:EVENT<x>:SPIBus:DATA<x>?	SPI バストリガの各データに関するすべての設定値を問い合わせます。	5-215
:TRIGger:EINterval:EVENT<x>:SPIBus:DATA<x>:BYTE	SPI バストリガの各データの設定データ数を設定 / 問い合わせします。	5-215
:TRIGger:EINterval:EVENT<x>:SPIBus:DATA<x>:CONDition	SPI バストリガの各データの判定方法 (一致 / 不一致) を設定 / 問い合わせします。	5-216
:TRIGger:EINterval:EVENT<x>:SPIBus:DATA<x>:DPOSition	SPI バストリガの各データのパターン比較先頭位置を設定 / 問い合わせします。	5-216
:TRIGger:EINterval:EVENT<x>:SPIBus:DATA<x>:HEXA<x>	SPI バストリガの各データを HEXA で設定します。	5-216
:TRIGger:EINterval:EVENT<x>:SPIBus:DATA<x>:PATtern<x>	SPI バストリガの各データを BINARY で設定 / 問い合わせします。	5-216
:TRIGger:EINterval:EVENT<x>:SPIBus:DATA<x>:SOURce	SPI バストリガの各データのトレースを設定 / 問い合わせします。	5-216
:TRIGger:EINterval:EVENT<x>:SPIBus:MODE	SPI バストリガの結線方式 (3 線式 / 4 線式) を設定 / 問い合わせします。	5-216
:TRIGger:EINterval:EVENT<x>:STATe?	各イベントの成立条件に関するすべての設定値を問い合わせます。	5-217
:TRIGger:EINterval:EVENT<x>:STATe:CHANnel<x>	各チャンネルの成立条件を設定 / 問い合わせします。	5-217
:TRIGger:EINterval:EVENT<x>:STATe:LOGic	成立条件のロジックを設定 / 問い合わせします。	5-217
:TRIGger:EINterval:EVENT<x>:TYPE	各イベントのトリガの種類を設定 / 問い合わせします。	5-217
:TRIGger:EINterval:EVENT<x>:WIDTH?	各イベントのパルス幅トリガのすべての設定値を問い合わせます。	5-217
:TRIGger:EINterval:EVENT<x>:WIDTH:MODE	パルス幅トリガの判定モードを設定 / 問い合わせします。	5-218
:TRIGger:EINterval:EVENT<x>:WIDTH:POLarity	パルス幅トリガの極性を設定 / 問い合わせします。	5-218
:TRIGger:EINterval:EVENT<x>:WIDTH:SOURce	パルス幅トリガのトリガソースを設定 / 問い合わせします。	5-218
:TRIGger:EINterval:EVENT<x>:WIDTH:TIME<x>	パルス幅トリガのパルス幅を設定 / 問い合わせします。	5-218
:TRIGger:EINterval:MODE	イベントインターバルの判定モードを設定 / 問い合わせします。	5-218
:TRIGger:EINterval:TIME<x>	イベントインターバルのインターバル時間を設定 / 問い合わせします。	5-218
:TRIGger:EINterval:TRY?	イベントインターバルのテストに関するすべての設定値を問い合わせします。	5-218
:TRIGger:EINterval:TRY:MODE	テストモードを設定 / 問い合わせします。	5-219
:TRIGger:EINterval:TRY:SELect	テストモードの対象イベントを設定 / 問い合わせします。	5-219
:TRIGger:ENHanced?	Enhanced トリガに関するすべての設定値を問い合わせます。	5-219
:TRIGger:ENHanced:CANBus?	CAN バス信号トリガに関するすべての設定値を問い合わせます。	5-220
:TRIGger:ENHanced:CANBus:ACK	CAN バス信号トリガの ACK 条件を設定 / 問い合わせします。	5-220
:TRIGger:ENHanced:CANBus:BRATe	CAN バス信号トリガのビットレート (データ転送速度) を設定 / 問い合わせします。	5-220
:TRIGger:ENHanced:CANBus:DATA?	CAN バス信号トリガのデータに関するすべての設定値を問い合わせます。	5-220
:TRIGger:ENHanced:CANBus:DATA:BORDER	CAN バス信号トリガのデータのバイトオーダーを設定 / 問い合わせします。	5-220

コマンド	機能	ページ
:TRIGger:ENHanced:CANBus:DATA:CONDition	CAN バス信号トリガのデータ条件を設定 / 問い合わせします。	5-220
:TRIGger:ENHanced:CANBus:DATA:DATA<x>	CAN バス信号トリガのデータの比較データを設定 / 問い合わせします。	5-221
:TRIGger:ENHanced:CANBus:DATA:DLC	CAN バス信号トリガのデータの有効バイト数 (DLC) を設定 / 問い合わせします。	5-221
:TRIGger:ENHanced:CANBus:DATA:HEXA	CAN バス信号トリガのデータを HEXA で設定します。	5-221
:TRIGger:ENHanced:CANBus:DATA:MSBLSb	CAN バス信号トリガのデータの MSB/LSB のビットを設定 / 問い合わせします。	5-221
:TRIGger:ENHanced:CANBus:DATA:PATtern	CAN バス信号トリガのデータを BINARY で設定 / 問い合わせします。	5-221
:TRIGger:ENHanced:CANBus:DATA:SIGN	CAN バス信号トリガのデータの符号を設定 / 問い合わせします。	5-221
:TRIGger:ENHanced:CANBus:IDExt?	CAN バス信号トリガの拡張フォーマットの ID に関するすべての設定値を問い合わせます。	5-221
:TRIGger:ENHanced:CANBus:IDExt:HEXA	CAN バス信号トリガの拡張フォーマットの ID を HEXA で設定します。	5-222
:TRIGger:ENHanced:CANBus:IDExt:PATtern	CAN バス信号トリガの拡張フォーマットの ID を BINARY で設定 / 問い合わせします。	5-222
:TRIGger:ENHanced:CANBus:IDOR?	CAN バス信号トリガの OR 条件に関するすべての設定値を問い合わせます。	5-222
:TRIGger:ENHanced:CANBus:IDOR:ID<x>?	CAN バス信号トリガの OR 条件の各 ID に関するすべての設定値を問い合わせます。	5-222
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:ACK	CAN バス信号トリガの OR 条件の各 ACK 条件を設定 / 問い合わせします。	5-223
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA?	CAN バス信号トリガの OR 条件の各データに関するすべての設定値を問い合わせます。	5-223
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:BORder	CAN バス信号トリガの OR 条件の各データのバイトオーダを設定 / 問い合わせします。	5-223
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:CONDition	CAN バス信号トリガの OR 条件の各データ条件を設定 / 問い合わせします。	5-223
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:DATA<x>	CAN バス信号トリガの OR 条件の各データの比較データを設定 / 問い合わせします。	5-223
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:DLC	CAN バス信号トリガの OR 条件の各データの有効バイト数 (DLC) を設定 / 問い合わせします。	5-224
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:HEXA	CAN バス信号トリガの OR 条件の各データを HEXA で設定します。	5-224
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:MSBLSb	CAN バス信号トリガの OR 条件の各データの MSB/LSB のビットを設定 / 問い合わせします。	5-224
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:PATtern	CAN バス信号トリガの OR 条件の各データを BINARY で設定 / 問い合わせします。	5-224
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:SIGN	CAN バス信号トリガの OR 条件の各データの符号を設定 / 問い合わせします。	5-224
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:FORMat	CAN バス信号トリガの OR 条件の各メッセージフォーマット (標準 / 拡張) を設定 / 問い合わせします。	5-224
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDExt?	CAN バス信号トリガの OR 条件の各拡張フォーマットの ID に関するすべての設定値を問い合わせます。	5-225
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDExt:HEXA	CAN バス信号トリガの OR 条件の各拡張フォーマットの ID を HEXA で設定します。	5-225
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDExt:PATtern	CAN バス信号トリガの OR 条件の各拡張フォーマットの ID を BINARY で設定 / 問い合わせします。	5-225
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDStd?	CAN バス信号トリガの OR 条件の各標準フォーマットの ID に関するすべての設定値を問い合わせます。	5-225
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDStd:HEXA	CAN バス信号トリガの OR 条件の各標準フォーマットの ID を HEXA で設定します。	5-225
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDStd:PATtern	CAN バス信号トリガの OR 条件の各標準フォーマットの ID を BINARY で設定 / 問い合わせします。	5-225
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:MODE	CAN バス信号トリガの OR 条件の各条件の有効 (1) / 無効 (0) を設定 / 問い合わせします。	5-225
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:RTR	CAN バス信号トリガの OR 条件の各 RTR を設定 / 問い合わせします。	5-226
:TRIGger:ENHanced:CANBus:IDStd?	CAN バス信号トリガの標準フォーマットの ID に関するすべての設定値を問い合わせます。	5-226
:TRIGger:ENHanced:CANBus:IDStd:HEXA	CAN バス信号トリガの標準フォーマットの ID を HEXA で設定します。	5-226

## 5.1 コマンド一覧表

コマンド	機能	ページ
:TRIGger:ENHanced:CANBus:IDSTd: PATtern	CAN バス信号トリガの標準フォーマットの ID を BINARY で設定 / 問い合わせします。	5-226
:TRIGger:ENHanced:CANBus:MODE	CAN バス信号トリガのモードを設定 / 問い合わせします。	5-226
:TRIGger:ENHanced:CANBus: RECESSive	CAN バス信号トリガのリセッシブレベル (バスレベル) を設定 / 問い合わせします。	5-226
:TRIGger:ENHanced:CANBus:RTR	CAN バス信号トリガの RTR を設定 / 問い合わせします。	5-226
:TRIGger:ENHanced:CANBus:SOURce	CAN バス信号トリガのトリガソースを設定 / 問い合わせします。	5-226
:TRIGger:ENHanced:CANBus:SPOint	CAN バス信号トリガのサンプルポイントを設定 / 問い合わせします。	5-226
:TRIGger:ENHanced:I2CBus?	I <sup>2</sup> C バストリガに関するすべての設定値を問い合わせます。	5-227
:TRIGger:ENHanced:I2CBus:ADATa?	I <sup>2</sup> C バストリガのアドレスに関するすべての設定値を問い合わせます。	5-227
:TRIGger:ENHanced:I2CBus:ADATa: BIT10address?	I <sup>2</sup> C バストリガの 10bit アドレスに関するすべての設定値を問い合わせます。	5-227
:TRIGger:ENHanced:I2CBus:ADATa: BIT10address:HEXA	I <sup>2</sup> C バストリガの 10bit アドレスを HEXA で設定します。	5-227
:TRIGger:ENHanced:I2CBus:ADATa: BIT10address:PATtern	I <sup>2</sup> C バストリガの 10bit アドレスを BINARY で設定 / 問い合わせします。	5-227
:TRIGger:ENHanced:I2CBus:ADATa: BIT7Address?	I <sup>2</sup> C バストリガの 7bit アドレスに関するすべての設定値を問い合わせます。	5-227
:TRIGger:ENHanced:I2CBus:ADATa: BIT7Address:HEXA	I <sup>2</sup> C バストリガの 7bit アドレスを HEXA で設定します。	5-227
:TRIGger:ENHanced:I2CBus:ADATa: BIT7Address:PATtern	I <sup>2</sup> C バストリガの 7bit アドレスを BINARY で設定 / 問い合わせします。	5-228
:TRIGger:ENHanced:I2CBus:ADATa: BIT7APsub?	I <sup>2</sup> C バストリガの 7bit + Sub アドレスに関するすべての設定値を問い合わせます。	5-228
:TRIGger:ENHanced:I2CBus:ADATa: BIT7APsub:ADDRESS?	I <sup>2</sup> C バストリガの 7bit + Sub アドレスの 7bit アドレスに関するすべての設定値を問い合わせます。	5-228
:TRIGger:ENHanced:I2CBus:ADATa: BIT7APsub:ADDRESS:HEXA	I <sup>2</sup> C バストリガの 7bit + Sub アドレスの 7bit アドレスを HEXA で設定します。	5-228
:TRIGger:ENHanced:I2CBus:ADATa: BIT7APsub:ADDRESS:PATtern	I <sup>2</sup> C バストリガの 7bit + Sub アドレスの 7bit アドレスを BINARY で設定 / 問い合わせします。	5-228
:TRIGger:ENHanced:I2CBus:ADATa: BIT7APsub:SADDRESS?	I <sup>2</sup> C バストリガの 7bit + Sub アドレスの Sub アドレスに関するすべての設定値を問い合わせます。	5-228
:TRIGger:ENHanced:I2CBus:ADATa: BIT7APsub:SADDRESS:HEXA	I <sup>2</sup> C バストリガの 7bit + Sub アドレスの Sub アドレスを HEXA で設定します。	5-228
:TRIGger:ENHanced:I2CBus:ADATa: BIT7APsub:SADDRESS:PATtern	I <sup>2</sup> C バストリガの 7bit + Sub アドレスの Sub アドレスを BINARY で設定 / 問い合わせします。	5-228
:TRIGger:ENHanced:I2CBus:ADATa: TYPE	I <sup>2</sup> C バストリガのアドレスの種類を設定 / 問い合わせします。	5-229
:TRIGger:ENHanced:I2CBus:CLOCK?	I <sup>2</sup> C バストリガのクロックに関するすべての設定値を問い合わせます。	5-229
:TRIGger:ENHanced:I2CBus:CLOCK: SOURce	I <sup>2</sup> C バストリガのクロックトレースを設定 / 問い合わせします。	5-229
:TRIGger:ENHanced:I2CBus:DATA?	I <sup>2</sup> C バストリガのデータに関するすべての設定値を問い合わせます。	5-229
:TRIGger:ENHanced:I2CBus:DATA: BYTE	I <sup>2</sup> C バストリガの設定データ数を設定 / 問い合わせします。	5-229
:TRIGger:ENHanced:I2CBus:DATA: CONDition	I <sup>2</sup> C バストリガのデータの判定方法 (一致 / 不一致) を設定 / 問い合わせします。	5-229
:TRIGger:ENHanced:I2CBus:DATA: DPOSITION	I <sup>2</sup> C バストリガのデータのパターン比較する位置を設定 / 問い合わせします。	5-229
:TRIGger:ENHanced:I2CBus:DATA: HEXA<x>	I <sup>2</sup> C バストリガのデータを HEXA で設定します。	5-229
:TRIGger:ENHanced:I2CBus:DATA: MODE	I <sup>2</sup> C バストリガのデータ条件の有効 / 無効を設定 / 問い合わせします。	5-230
:TRIGger:ENHanced:I2CBus:DATA: PATtern<x>	I <sup>2</sup> C バストリガのデータを BINARY で設定 / 問い合わせします。	5-230
:TRIGger:ENHanced:I2CBus:DATA: PMODE	I <sup>2</sup> C バストリガのデータのパターン比較先頭位置モードを設定 / 問い合わせします。	5-230
:TRIGger:ENHanced:I2CBus:DATA: SOURce	I <sup>2</sup> C バストリガのデータトレースを設定 / 問い合わせします。	5-230
:TRIGger:ENHanced:I2CBus:GCAL1?	I <sup>2</sup> C バストリガのジェネラルコールに関するすべての設定値を問い合わせます。	5-230
:TRIGger:ENHanced:I2CBus:GCAL1: BIT7maddress?	I <sup>2</sup> C バストリガのジェネラルコールの 7bit マスタアドレスに関するすべての設定値を問い合わせます。	5-230
:TRIGger:ENHanced:I2CBus:GCAL1: BIT7maddress:HEXA	I <sup>2</sup> C バストリガのジェネラルコールの 7bit マスタアドレスを HEXA で設定します。	5-230

コマンド	機能	ページ
:TRIGger:ENHanced:I2CBus:GCAL1:BIT7maddress:PATtern	I <sup>2</sup> C バストリガのジェネラルコールの 7bit マスタアドレスを BINARY で設定 / 問い合わせします。	5-230
:TRIGger:ENHanced:I2CBus:GCAL1:SBYte (Second Byte)	I <sup>2</sup> C バストリガのジェネラルコールのセカンドバイトのタイプを設定 / 問い合わせします。	5-231
:TRIGger:ENHanced:I2CBus:MODE	I <sup>2</sup> C バストリガのトリガモードを設定 / 問い合わせします。	5-231
:TRIGger:ENHanced:I2CBus:NAIGnore?	I <sup>2</sup> C バストリガの NON ACK 無視モードに関するすべての設定値を問い合わせます。	5-231
:TRIGger:ENHanced:I2CBus:NAIGnore:HSMODE	I <sup>2</sup> C バストリガのハイスピードモードで NON ACK を無視する / しないを設定 / 問い合わせします。	5-231
:TRIGger:ENHanced:I2CBus:NAIGnore:RACcess	I <sup>2</sup> C バストリガのリードアクセスモードで NON ACK を無視する / しないを設定 / 問い合わせします。	5-231
:TRIGger:ENHanced:I2CBus:NAIGnore:SBYte (Start Byte)	I <sup>2</sup> C バストリガのスタートバイトで NON ACK を無視する / しないを設定 / 問い合わせします。	5-231
:TRIGger:ENHanced:I2CBus:SBHSMODE?	I <sup>2</sup> C バストリガのスタートバイト / ハイスピードモードに関するすべての設定値を問い合わせます。	5-231
:TRIGger:ENHanced:I2CBus:SBHSMODE:TYPE	I <sup>2</sup> C バストリガのスタートバイト / ハイスピードモードのタイプを設定 / 問い合わせします。	5-231
:TRIGger:ENHanced:LINBus?	LIN バストリガに関するすべての設定値を問い合わせします。	5-232
:TRIGger:ENHanced:LINBus:BRATe	LIN バス信号トリガのビットレート (データ転送速度) を設定 / 問い合わせします。	5-232
:TRIGger:ENHanced:LINBus:SOURce	LIN バス信号トリガのトリガソースを設定 / 問い合わせします。	5-232
:TRIGger:ENHanced:SPATtern?(Serial Pattern)	シリアルパターントリガに関するすべての設定値を問い合わせます。	5-232
:TRIGger:ENHanced:SPATtern:BITRate	シリアルパターントリガのビットレートを設定 / 問い合わせします。	5-232
:TRIGger:ENHanced:SPATtern:CLEar	シリアルパターントリガのパターンをすべてクリア (Don't care) します。	5-232
:TRIGger:ENHanced:SPATtern:CLOCK?	シリアルパターントリガのクロックに関するすべての設定値を問い合わせます。	5-232
:TRIGger:ENHanced:SPATtern:CLOCK:MODE	シリアルパターントリガのクロックの有効 / 無効を設定 / 問い合わせします。	5-232
:TRIGger:ENHanced:SPATtern:CLOCK:POLarity	シリアルパターントリガのクロックトレースの極性を設定 / 問い合わせします。	5-233
:TRIGger:ENHanced:SPATtern:CLOCK:SOURce	シリアルパターントリガのクロックトレースを設定 / 問い合わせします。	5-233
:TRIGger:ENHanced:SPATtern:CS	シリアルパターントリガのチップセレクトの有効 / 無効を設定 / 問い合わせします。	5-233
:TRIGger:ENHanced:SPATtern:DATA?	シリアルパターントリガのデータに関するすべての設定値を問い合わせます。	5-233
:TRIGger:ENHanced:SPATtern:DATA:ACTive	シリアルパターントリガのデータのアクティブレベルを設定 / 問い合わせします。	5-233
:TRIGger:ENHanced:SPATtern:DATA:SOURce	シリアルパターントリガのデータトレースを設定 / 問い合わせします。	5-233
:TRIGger:ENHanced:SPATtern:HEXA	シリアルパターントリガのパターンを HEXA で設定します。	5-233
:TRIGger:ENHanced:SPATtern:LATCh?	シリアルパターントリガのラッチに関するすべての設定値を問い合わせます。	5-233
:TRIGger:ENHanced:SPATtern:LATCh:POLarity	シリアルパターントリガのラッチトレースの極性を設定 / 問い合わせします。	5-234
:TRIGger:ENHanced:SPATtern:LATCh:SOURce	シリアルパターントリガのラッチトレースを設定 / 問い合わせします。	5-234
:TRIGger:ENHanced:SPATtern:PATtern	シリアルパターントリガのパターンを BINARY で設定 / 問い合わせします。	5-234
:TRIGger:ENHanced:SPIBus?	SPI バストリガに関するすべての設定値を問い合わせます。	5-234
:TRIGger:ENHanced:SPIBus:BITOrder	SPI バストリガのビットオーダーを設定 / 問い合わせします。	5-234
:TRIGger:ENHanced:SPIBus:CLOCK?	SPI バストリガのクロックに関するすべての設定値を問い合わせます。	5-234
:TRIGger:ENHanced:SPIBus:CLOCK:POLarity	SPI バストリガのクロックトレースの極性を設定 / 問い合わせします。	5-234
:TRIGger:ENHanced:SPIBus:CLOCK:SOURce	SPI バストリガのクロックトレースを設定 / 問い合わせします。	5-235
:TRIGger:ENHanced:SPIBus:CS?	SPI バストリガのチップセレクトに関するすべての設定値を問い合わせます。	5-235
:TRIGger:ENHanced:SPIBus:CS:ACTive	SPI バストリガのチップセレクトのアクティブレベルを設定 / 問い合わせします。	5-235

## 5.1 コマンド一覧表

コマンド	機能	ページ
:TRIGger:ENHanced:SPIBus:CS:SOURce	SPI バストリガのチップセレクトトレースを設定 / 問い合わせします。	5-235
:TRIGger:ENHanced:SPIBus:DATA<x>?	SPI バストリガの各データに関するすべての設定値を問い合わせます。	5-235
:TRIGger:ENHanced:SPIBus:DATA<x>:BYTE	SPI バストリガの各データの設定データ数を設定 / 問い合わせします。	5-235
:TRIGger:ENHanced:SPIBus:DATA<x>:CONDition	SPI バストリガの各データの判定方法 (一致 / 不一致) を設定 / 問い合わせします。	5-235
:TRIGger:ENHanced:SPIBus:DATA<x>:DPOSITion	SPI バストリガの各データのパターン比較先頭位置を設定 / 問い合わせします。	5-236
:TRIGger:ENHanced:SPIBus:DATA<x>:HEXA<x>	SPI バストリガの各データを HEXA で設定します。	5-236
:TRIGger:ENHanced:SPIBus:DATA<x>:PATTERn<x>	SPI バストリガの各データを BINARY で設定 / 問い合わせします。	5-236
:TRIGger:ENHanced:SPIBus:DATA<x>:SOURce	SPI バストリガの各データのトレースを設定 / 問い合わせします。	5-236
:TRIGger:ENHanced:SPIBus::MODE	SPI バストリガの結線方式 (3 線式 / 4 線式) を設定 / 問い合わせします。	5-236
:TRIGger:ENHanced:TV?	TV トリガに関するすべての設定値を問い合わせます。	5-236
:TRIGger:ENHanced:TV:COUPLing?	TV トリガのトリガカップリングを問い合わせます。	5-236
:TRIGger:ENHanced:TV:CUSTomize	TV トリガの Sync Guard 機能の ON/OFF を設定 / 問い合わせします。	5-237
:TRIGger:ENHanced:TV:FIELD	TV トリガのフィールドを設定 / 問い合わせします。	5-237
:TRIGger:ENHanced:TV:FRAME	TV トリガのフレームスキップ機能を設定 / 問い合わせします。	5-237
:TRIGger:ENHanced:TV:{HDTV NTSC PAL SDTV USERdefine}?	TV トリガのモードに関するすべての設定値を問い合わせます。	5-237
:TRIGger:ENHanced:TV:{HDTV NTSC PAL SDTV}:HFRejection?(HighFrequencyREJECTION)	TV トリガのローパスフィルタ (HF リジェクション) を問い合わせます。	5-237
:TRIGger:ENHanced:TV:{HDTV NTSC PAL SDTV USERdefine}:LINE	TV トリガをかけるラインを設定 / 問い合わせします。	5-237
:TRIGger:ENHanced:TV:{HDTV NTSC PAL SDTV USERdefine}:POLarity	TV トリガの入力の極性を設定 / 問い合わせします。	5-237
:TRIGger:ENHanced:TV:LEVel	TV トリガのトリガレベルを設定 / 問い合わせします。	5-237
:TRIGger:ENHanced:TV:SGUard	TV トリガの Sync Guard を設定 / 問い合わせします。	5-237
:TRIGger:ENHanced:TV:SOURce	TV トリガのトリガソースを設定 / 問い合わせします。	5-238
:TRIGger:ENHanced:TV:TYPE	TV トリガの入力の種類を設定 / 問い合わせします。	5-238
:TRIGger:ENHanced:TV:USERdefine:DEFinition	ユーザー定義の解像度を設定 / 問い合わせします。	5-238
:TRIGger:ENHanced:TV:USERdefine:HFRejection(HighFrequencyREJECTION)	ユーザー定義のローパスフィルタ (HF リジェクション) を設定 / 問い合わせします。	5-238
:TRIGger:ENHanced:TV:USERdefine:HSYnc(Hsync Freq)	ユーザー定義の水平同期信号の周波数を設定 / 問い合わせします。	5-238
:TRIGger:ENHanced:UART?	UART 信号トリガに関するすべて設定値を問い合わせします。	5-238
:TRIGger:ENHanced:UART:BRATe	UART 信号トリガのビットレート (データ転送速度) を設定 / 問い合わせします。	5-238
:TRIGger:ENHanced:UART:FORMat	UART 信号トリガのフォーマットを設定 / 問い合わせします。	5-238
:TRIGger:ENHanced:UART:POLarity	UART 信号トリガの極性を設定 / 問い合わせします。	5-239
:TRIGger:ENHanced:UART:SOURce	UART 信号トリガのトリガソースを設定 / 問い合わせします。	5-239
:TRIGger:ENHanced:UART:SPOINT	UART 信号トリガのサンプルポイントを設定 / 問い合わせします。	5-239
:TRIGger:ESTate?	Edge/State トリガに関するすべての設定値を問い合わせます。	5-239
:TRIGger:ESTate:EOR?	OR トリガに関するすべての設定値を問い合わせます。	5-239
:TRIGger:ESTate:EOR:CHANnel<x>	OR トリガの各チャンネルの極性を設定 / 問い合わせします。	5-239
:TRIGger:ESTate:POLarity	Edge/State トリガの極性を設定 / 問い合わせします。	5-239
:TRIGger:ESTate:SOURce	Edge/State トリガのトリガソースを設定 / 問い合わせします。	5-239
:TRIGger:HOLDoff	ホールドオフ時間を設定 / 問い合わせします。	5-240
:TRIGger:LOGic?	ロジックトリガに関するすべての設定値を問い合わせます。	5-240
:TRIGger:LOGic:CLOCK?	ロジックトリガのクロックに関するすべての設定値を問い合わせます。	5-240
:TRIGger:LOGic:CLOCK:POLarity	ロジックトリガのクロックの極性を設定 / 問い合わせします。	5-240
:TRIGger:LOGic:CLOCK:SOURce	ロジックトリガのクロックのソースを設定 / 問い合わせします。	5-240
:TRIGger:LOGic:ESTate?	ロジックの Edge/State トリガに関するすべての設定値を問い合わせます。	5-240
:TRIGger:LOGic:ESTate:POLarity	ロジックの Edge/State トリガの極性を設定 / 問い合わせします。	5-240
:TRIGger:LOGic:ESTate:SOURce	ロジックの Edge/State トリガのトリガソースを設定 / 問い合わせします。	5-241
:TRIGger:LOGic:I2CBus?	ロジック I <sup>2</sup> C バストリガに関するすべての設定値を問い合わせます。	5-241

コマンド	機能	ページ
:TRIGger:LOGic:I2Cbus:ADATa?	ロジック I2C バストリガのアドレスに関するすべての設定値を問い合わせます。	5-241
:TRIGger:LOGic:I2Cbus:ADATa:BIT10address?	ロジック I2C バストリガの 10bit アドレスに関するすべての設定値を問い合わせます。	5-241
:TRIGger:LOGic:I2Cbus:ADATa:BIT10address:HEXA	ロジック I2C バストリガの 10bit アドレスを HEXA で設定します。	5-241
:TRIGger:LOGic:I2Cbus:ADATa:BIT10address:PATtern	ロジック I2C バストリガの 10bit アドレスを BINARY で設定 / 問い合わせします。	5-241
:TRIGger:LOGic:I2Cbus:ADATa:BIT7Address?	ロジック I2C バストリガの 7bit アドレスに関するすべての設定値を問い合わせます。	5-241
:TRIGger:LOGic:I2Cbus:ADATa:BIT7Address:HEXA	ロジック I2C バストリガの 7bit アドレスを HEXA で設定します。	5-242
:TRIGger:LOGic:I2Cbus:ADATa:BIT7Address:PATtern	ロジック I2C バストリガの 7bit アドレスを BINARY で設定 / 問い合わせします。	5-242
:TRIGger:LOGic:I2Cbus:ADATa:BIT7APsub?	ロジック I2C バストリガの 7bit + Sub アドレスに関するすべての設定値を問い合わせます。	5-242
:TRIGger:LOGic:I2Cbus:ADATa:BIT7APsub:ADdRes?	ロジック I2C バストリガの 7bit + Sub アドレスの 7bit アドレスに関するすべての設定値を問い合わせます。	5-242
:TRIGger:LOGic:I2Cbus:ADATa:BIT7APsub:ADdRes:HEXA	ロジック I2C バストリガの 7bit + Sub アドレスの 7bit アドレスを HEXA で設定します。	5-242
:TRIGger:LOGic:I2Cbus:ADATa:BIT7APsub:ADdRes:PATtern	ロジック I2C バストリガの 7bit + Sub アドレスの 7bit アドレスを BINARY で設定 / 問い合わせします。	5-242
:TRIGger:LOGic:I2Cbus:ADATa:BIT7APsub:SADdRes?	ロジック I2C バストリガの 7bit + Sub アドレスの Sub アドレスに関するすべての設定値を問い合わせます。	5-242
:TRIGger:LOGic:I2Cbus:ADATa:BIT7APsub:SADdRes:HEXA	ロジック I2C バストリガの 7bit + Sub アドレスの Sub アドレスを HEXA で設定します。	5-242
:TRIGger:LOGic:I2Cbus:ADATa:BIT7APsub:SADdRes:PATtern	ロジック I2C バストリガの 7bit + Sub アドレスの Sub アドレスを BINARY で設定 / 問い合わせします。	5-243
:TRIGger:LOGic:I2Cbus:ADATa:TYPE	ロジック I2C バストリガのアドレスの種類を設定 / 問い合わせします。	5-243
:TRIGger:LOGic:I2Cbus:CLOCK?	ロジック I2C バストリガのクロックに関するすべての設定値を問い合わせます。	5-243
:TRIGger:LOGic:I2Cbus:CLOCK:SOURce	ロジック I2C バストリガのクロックトレースを設定 / 問い合わせします。	5-243
:TRIGger:LOGic:I2Cbus:DATA?	ロジック I2C バストリガのデータに関するすべての設定値を問い合わせます。	5-243
:TRIGger:LOGic:I2Cbus:DATA:BYTE	ロジック I2C バストリガの設定データ数を設定 / 問い合わせします。	5-243
:TRIGger:LOGic:I2Cbus:DATA:CONDition	ロジック I2C バストリガのデータの判定方法 (一致 / 不一致) を設定 / 問い合わせします。	5-243
:TRIGger:LOGic:I2Cbus:DATA:DPOSition	ロジック I2C バストリガのデータのパターン比較する位置を設定 / 問い合わせします。	5-243
:TRIGger:LOGic:I2Cbus:DATA:HEXA<x>	ロジック I2C バストリガのデータを HEXA で設定します。	5-244
:TRIGger:LOGic:I2Cbus:DATA:MODE	ロジック I2C バストリガのデータ条件の有効 / 無効を設定 / 問い合わせします。	5-244
:TRIGger:LOGic:I2Cbus:DATA:PATtern<x>	ロジック I2C バストリガのデータを BINARY で設定 / 問い合わせします。	5-244
:TRIGger:LOGic:I2Cbus:DATA:PMODE	ロジック I2C バストリガのデータのパターン比較先頭位置モードを設定 / 問い合わせします。	5-244
:TRIGger:LOGic:I2Cbus:DATA:SOURce	ロジック I2C バストリガのデータトレースを設定 / 問い合わせします。	5-244
:TRIGger:LOGic:I2Cbus:GCALl?	ロジック I2C バストリガのジェネラルコールに関するすべての設定値を問い合わせます。	5-244
:TRIGger:LOGic:I2Cbus:GCALl:BIT7maddress?	ロジック I2C バストリガのジェネラルコールの 7bit マスタアドレスに関するすべての設定値を問い合わせます。	5-244
:TRIGger:LOGic:I2Cbus:GCALl:BIT7maddress:HEXA	ロジック I2C バストリガのジェネラルコールの 7bit マスタアドレスを HEXA で設定します。	5-244
:TRIGger:LOGic:I2Cbus:GCALl:BIT7maddress:PATtern	ロジック I2C バストリガのジェネラルコールの 7bit マスタアドレスを BINARY で設定 / 問い合わせします。	5-245
:TRIGger:LOGic:I2Cbus:GCALl:SBYTE (Second Byte)	ロジック I2C バストリガのジェネラルコールのセカンドバイトのタイプを設定 / 問い合わせします。	5-245
:TRIGger:LOGic:I2Cbus:MODE	ロジック I2C バストリガのトリガモードを設定 / 問い合わせします。	5-245
:TRIGger:LOGic:I2Cbus:NAIGnore?	ロジック I2C バストリガの NON ACK 無視モードに関するすべての設定値を問い合わせます。	5-245
:TRIGger:LOGic:I2Cbus:NAIGnore:HSMODE	ロジック I2C バストリガのハイスピードモードで NON ACK を無視する / しないを設定 / 問い合わせします。	5-245



## 5.1 コマンド一覧表

コマンド	機能	ページ
:TRIGger:LOGic:I2Cbus:NAIGnore:RACcEss	ロジック I <sup>2</sup> C バストリガのリードアクセスモードで NON ACK を無視する / しないを設定 / 問い合わせします。	5-245
:TRIGger:LOGic:I2Cbus:NAIGnore:SBYTe (Start Byte)	ロジック I <sup>2</sup> C バストリガのスタートバイトで NON ACK を無視する / しないを設定 / 問い合わせします。	5-245
:TRIGger:LOGic:I2Cbus:SBHSmode?	ロジック I <sup>2</sup> C バストリガのスタートバイト / ハイスピードモードに関するすべての設定値を問い合わせます。	5-245
:TRIGger:LOGic:I2Cbus:SBHSmode:TYPE	ロジック I <sup>2</sup> C バストリガのスタートバイト / ハイスピードモードのタイプを設定 / 問い合わせします。	5-246
:TRIGger:LOGic:LINBus?	ロジック LIN バス信号トリガに関するすべての設定値を問い合わせします。	5-246
:TRIGger:LOGic:LINBus:BRATe	ロジック LIN バス信号トリガのビットレート (データ転送速度) を設定 / 問い合わせします。	5-246
:TRIGger:LOGic:LINBus:SOURce	ロジック LIN バス信号トリガのトリガソースを設定 / 問い合わせします。	5-246
:TRIGger:LOGic:SPATtern? (Serial Pattern)	ロジックシリアルパターントリガに関するすべての設定値を問い合わせます。	5-246
:TRIGger:LOGic:SPATtern:BITRate	ロジックシリアルパターントリガのビットレートを設定 / 問い合わせします。	5-246
:TRIGger:LOGic:SPATtern:CLear	ロジックシリアルパターントリガのパターンをすべてクリア (Don't care) します。	5-246
:TRIGger:LOGic:SPATtern:CLOCK?	ロジックシリアルパターントリガのクロックに関するすべての設定値を問い合わせます。	5-246
:TRIGger:LOGic:SPATtern:CLOCK:MODE	ロジックシリアルパターントリガのクロックの有効 / 無効を設定 / 問い合わせします。	5-246
:TRIGger:LOGic:SPATtern:CLOCK:POLarity	ロジックシリアルパターントリガのクロックトレースの極性を設定 / 問い合わせします。	5-247
:TRIGger:LOGic:SPATtern:CLOCK:SOURce	ロジックシリアルパターントリガのクロックトレースを設定 / 問い合わせします。	5-247
:TRIGger:LOGic:SPATtern:CS	ロジックシリアルパターントリガのチップセレクトの有効 / 無効を設定 / 問い合わせします。	5-247
:TRIGger:LOGic:SPATtern:DATA?	ロジックシリアルパターントリガのデータに関するすべての設定値を問い合わせます。	5-247
:TRIGger:LOGic:SPATtern:DATA:ACTive	ロジックシリアルパターントリガのデータのアクティブレベルを設定 / 問い合わせします。	5-247
:TRIGger:LOGic:SPATtern:DATA:SOURce	ロジックシリアルパターントリガのデータトレースを設定 / 問い合わせします。	5-247
:TRIGger:LOGic:SPATtern:HEXA	ロジックシリアルパターントリガのパターンを HEXA で設定します。	5-247
:TRIGger:LOGic:SPATtern:LATCh?	ロジックシリアルパターントリガのラッチに関するすべての設定値を問い合わせます。	5-247
:TRIGger:LOGic:SPATtern:LATCh:POLarity	ロジックシリアルパターントリガのラッチトレースの極性を設定 / 問い合わせします。	5-248
:TRIGger:LOGic:SPATtern:LATCh:SOURce	ロジックシリアルパターントリガのラッチトレースを設定 / 問い合わせします。	5-248
:TRIGger:LOGic:SPATtern:PATtern	ロジックシリアルパターントリガのパターンを BINARY で設定 / 問い合わせします。	5-248
:TRIGger:LOGic:SPIBus?	ロジック SPI バストリガに関するすべての設定値を問い合わせます。	5-248
:TRIGger:LOGic:SPIBus:BITOrder	ロジック SPI バストリガのビットオーダーを設定 / 問い合わせします。	5-248
:TRIGger:LOGic:SPIBus:CLOCK?	ロジック SPI バストリガのクロックに関するすべての設定値を問い合わせます。	5-248
:TRIGger:LOGic:SPIBus:CLOCK:POLarity	ロジック SPI バストリガのクロックトレースの極性を設定 / 問い合わせします。	5-248
:TRIGger:LOGic:SPIBus:CLOCK:SOURce	ロジック SPI バストリガのクロックトレースを設定 / 問い合わせします。	5-249
:TRIGger:LOGic:SPIBus:CS?	ロジック SPI バストリガのチップセレクトに関するすべての設定値を問い合わせます。	5-249
:TRIGger:LOGic:SPIBus:CS:ACTive	ロジック SPI バストリガのチップセレクトのアクティブレベルを設定 / 問い合わせします。	5-249
:TRIGger:LOGic:SPIBus:CS:SOURce	ロジック SPI バストリガのチップセレクトトレースを設定 / 問い合わせします。	5-249
:TRIGger:LOGic:SPIBus:DATA<x>?	ロジック SPI バストリガの各データに関するすべての設定値を問い合わせます。	5-249
:TRIGger:LOGic:SPIBus:DATA<x>:BYTE	ロジック SPI バストリガの各データの設定データ数を設定 / 問い合わせします。	5-249
:TRIGger:LOGic:SPIBus:DATA<x>:CONDition	ロジック SPI バストリガの各データの判定方法 (一致 / 不一致) を設定 / 問い合わせします。	5-249
:TRIGger:LOGic:SPIBus:DATA<x>:DPOSition	ロジック SPI バストリガの各データのパターン比較先頭位置を設定 / 問い合わせします。	5-249

コマンド	機能	ページ
:TRIGger:LOGic:SPIBus:DATA<x>: HEXA<x>	ロジック SPI バストリガの各データを HEXA で設定します。	5-250
:TRIGger:LOGic:SPIBus:DATA<x>: PATtern<x>	ロジック SPI バストリガの各データを BINARY で設定 / 問い合わせします。	5-250
:TRIGger:LOGic:SPIBus:DATA<x>: SOURce	ロジック SPI バストリガの各データのトレースを設定 / 問い合わせします。	5-250
:TRIGger:LOGic:SPIBus:MODE	ロジック SPI バストリガの結線方式 (3 線式 / 4 線式) を設定 / 問い合わせします。	5-250
:TRIGger:LOGic:STATe?	ロジックステートトリガに関するすべての設定値を問い合わせます。	5-250
:TRIGger:LOGic:STATe:BIT?	ロジックステートトリガのビットに関するすべての設定値を問い合わせます。	5-250
:TRIGger:LOGic:STATe:BIT:{A<x>  B<x> C<x> D<x>}	ロジックステートトリガの各ビットの成立条件を設定 / 問い合わせします。	5-251
:TRIGger:LOGic:STATe:BIT:CLear	ロジックステートトリガの各ビットの成立条件をすべてクリア (Don't care) します。	5-251
:TRIGger:LOGic:STATe:BIT:LOGic	ロジックステートトリガのロジックを設定 / 問い合わせします。	5-251
:TRIGger:LOGic:STATe:GROup<x>?	ロジックステートトリガの各グループに関するすべての設定値を問い合わせます。	5-251
:TRIGger:LOGic:STATe:GROup<x>: CLear	ロジックステートトリガの各グループの成立条件をすべてクリア (Don't care) します。	5-251
:TRIGger:LOGic:STATe:GROup<x>: CONDition	ロジックステートトリガの各グループの判定条件を設定 / 問い合わせします。	5-251
:TRIGger:LOGic:STATe:GROup<x>: HEXA	ロジックステートトリガの各グループの成立条件を HEXA で設定します。	5-251
:TRIGger:LOGic:STATe:GROup<x>: PATtern	ロジックステートトリガの各グループの成立条件を BINARY で設定 / 問い合わせします。	5-251
:TRIGger:LOGic:STATe:GROup<x>: SYMBOL	ロジックステートトリガの各グループのシンボルアイテムを設定します。	5-251
:TRIGger:LOGic:STATe:TYPE	ロジックステートトリガの設定方法を設定 / 問い合わせします。	5-252
:TRIGger:LOGic:UART?	ロジック UART 信号トリガに関するすべて設定値を問い合わせします。	5-252
:TRIGger:LOGic:UART:BRATe	ロジック UART 信号トリガのビットレート (データ転送速度) を設定 / 問い合わせします。	5-252
:TRIGger:LOGic:UART:FORMat	ロジック UART 信号トリガのフォーマットを設定 / 問い合わせします。	5-252
:TRIGger:LOGic:UART:POLarity	ロジック UART 信号トリガの極性を設定 / 問い合わせします。	5-252
:TRIGger:LOGic:UART:SOURce	ロジック UART 信号トリガのトリガソースを設定 / 問い合わせします。	5-252
:TRIGger:LOGic:UART:SPOint	ロジック UART 信号トリガのサンプルポイントを設定 / 問い合わせします。	5-252
:TRIGger:LOGic:WIDTh?	ロジックパルス幅トリガのすべての設定値を問い合わせます。	5-252
:TRIGger:LOGic:WIDTh:MODE	ロジックパルス幅トリガの判定モードを設定 / 問い合わせします。	5-252
:TRIGger:LOGic:WIDTh:POLarity	ロジックパルス幅トリガの極性を設定 / 問い合わせします。	5-253
:TRIGger:LOGic:WIDTh:SOURce	ロジックパルス幅トリガのトリガソースを設定 / 問い合わせします。	5-253
:TRIGger:LOGic:WIDTh:TIME<x>	ロジックパルス幅トリガのパルス幅を設定 / 問い合わせします。	5-253
:TRIGger:MODE	トリガモードを設定 / 問い合わせします。	5-253
:TRIGger:POSition	トリガポジションを設定 / 問い合わせします。	5-253
:TRIGger:SCOUNT (Single (N) Count)	トリガモードが Single (N) 時のトリガ成立回数を設定 / 問い合わせします。	5-253
:TRIGger:SOURce?	トリガソースに関するすべての設定値を問い合わせます。	5-253
:TRIGger:SOURce:CHANnel<x>?	トリガソースの各チャンネルに関するすべての設定値を問い合わせます。	5-253
:TRIGger:SOURce:CHANnel<x>: COUpling	各チャンネルのトリガカップリングを設定 / 問い合わせします。	5-254
:TRIGger:SOURce: CHANnel<x>:HFRejection (HighFrequencyREJECTION)	各チャンネルのローパスフィルタ (HF リジェクション) を設定 / 問い合わせします。	5-254
:TRIGger:SOURce:CHANnel<x>: HYSTeresis	各チャンネルのヒステリシスを設定 / 問い合わせします。	5-254
:TRIGger:SOURce:CHANnel<x>: LEVel	各チャンネルのトリガレベルを設定 / 問い合わせします。	5-254
:TRIGger:SOURce:CHANnel<x>: STATe	各チャンネルの成立条件を設定 / 問い合わせします。	5-254
:TRIGger:SOURce:CHANnel<x>: WIDTh	各チャンネルのウィンドウトリガの幅を設定 / 問い合わせします。	5-254
:TRIGger:SOURce:CHANnel<x>: WINDow	各チャンネルのウィンドウの ON/OFF を設定 / 問い合わせします。	5-255
:TRIGger:SOURce:EXTernal?	外部トリガに関するすべての設定値を問い合わせます。	5-255
:TRIGger:SOURce:EXTernal:LEVel	外部トリガのトリガレベルを設定 / 問い合わせします。	5-255

## 5.1 コマンド一覧表

コマンド	機能	ページ
:TRIGger:SOURce:EXtERnal:PROBe	外部トリガのプロープ減衰比を設定 / 問い合わせします。	5-255
:TRIGger:SOURce:LOGic	トリガソースのロジックを設定 / 問い合わせします。	5-255
:TRIGger:TYPE	トリガの種類を設定 / 問い合わせします。	5-255
:TRIGger:WIDTh?	パルス幅トリガのすべての設定値を問い合わせます。	5-255
:TRIGger:WIDTh:MODE	パルス幅トリガの判定モードを設定 / 問い合わせします。	5-255
:TRIGger:WIDTh:POLarity	パルス幅トリガの極性を設定 / 問い合わせします。	5-256
:TRIGger:WIDTh:SOURce	パルス幅トリガのトリガソースを設定 / 問い合わせします。	5-256
:TRIGger:WIDTh:TIME<x>	パルス幅トリガのパルス幅を設定 / 問い合わせします。	5-256

### WAVeform グループ

:WAVeform?	波形データのすべての情報を問い合わせます。	5-257
:WAVeform:BITS?	「:WAVeform:TRACe」で指定した波形データのビット長を問い合わせます。	5-257
:WAVeform:BYTeorder	2バイト以上のワードフォーマットのときの送信順序を設定 / 問い合わせします。	5-257
:WAVeform:END	「:WAVeform:TRACe」で指定した波形の、どの点を最後のデータとするかを設定 / 問い合わせします。	5-257
:WAVeform:FORMat	送信するデータのフォーマットを設定 / 問い合わせします。	5-257
:WAVeform:LENGth?	「:WAVeform:TRACe」で指定した波形の全データ点数を問い合わせます。	5-257
:WAVeform:OFFSet?	「:WAVeform:TRACe」で指定した波形データを物理値に変換するときのオフセット値を問い合わせます。	5-257
:WAVeform:POSition?	「:WAVeform:FORMat」で「RBYTe」を指定した場合の、電圧に換算するとき使用する垂直軸ポジションを問い合わせます。	5-257
:WAVeform:RANGe?	「:WAVeform:TRACe」で指定した波形データを物理値に換算するときのレンジ値を問い合わせます。	5-257
:WAVeform:RECOrd	WAVeform グループで対象となるレコード番号を設定 / 問い合わせします。	5-258
:WAVeform:RECOrd? MINimum	対象チャンネルのヒストリの最小レコード番号を問い合わせします。	5-258
:WAVeform:SEND?	「:WAVeform:TRACe」で指定した波形データを問い合わせます。	5-258
:WAVeform:SIGN?	「:WAVeform:TRACe」で指定した対象波形をバイナリデータで問い合わせる場合の、符号の有無を問い合わせます。	5-258
:WAVeform:SRATE? (Sample RATE)	「:WAVeform:RECOrd」で指定したレコードのサンプルレートを問い合わせます。	5-259
:WAVeform:START	「:WAVeform:TRACe」で指定した波形の、どの点を最初のデータとするかを設定 / 問い合わせします。	5-259
:WAVeform:TRACe	WAVeform グループで対象となる波形を設定 / 問い合わせします。	5-259
:WAVeform:TRIGger?	「:WAVeform:RECOrd」で指定したレコードのトリガポジションを問い合わせます。	5-259
:WAVeform:TYPE?	「:WAVeform:TRACe」で指定した波形のアクイジションモードを問い合わせます。	5-259

### ZOOM グループ

:ZOOM?	波形のズームに関するすべての設定値を問い合わせます。	5-260
:ZOOM:ALLocation<x>?	ズーム対象波形に関するすべての設定値を問い合わせます。	5-260
:ZOOM:ALLocation:ALLon	全波形をズーム対象にします。	5-260
:ZOOM:ALLocation:TRACe<x>	ズーム対象にしたいトレースの ON/OFF を設定 / 問い合わせします。	5-260
:ZOOM:FORMat<x>	ズーム波形の表示フォーマットを設定 / 問い合わせします。	5-260
:ZOOM:HLINKage	水平軸方向リンクの ON/OFF を設定 / 問い合わせします。	5-260
:ZOOM:HORizontal<x>?	水平軸方向のズームに関するすべての設定値を問い合わせます。	5-260
:ZOOM:HORizontal<x>:ASCROLL?	オートスクロール機能に関するすべての設定値を問い合わせます。	5-260
:ZOOM:HORizontal<x>:ASCROLL: JUMP	ズームの中心位置をメイン画面の左端、右端に移動します。	5-260
:ZOOM:HORizontal<x>:ASCROLL: SPEEd	オートスクロールのスピードを設定 / 問い合わせします。	5-261
:ZOOM:HORizontal<x>:ASCROLL: START	オートスクロールを開始します。	5-261
:ZOOM:HORizontal<x>:ASCROLL: STOP	オートスクロールを停止します。	5-261
:ZOOM:HORizontal<x>:MAG	水平軸方向のズーム倍率を設定 / 問い合わせします。	5-261
:ZOOM:HORizontal<x>:POSition	水平軸方向のズームの中心位置を設定 / 問い合わせします。	5-261
:ZOOM:MODE	ズーム波形の表示方法を設定 / 問い合わせします。	5-261
:ZOOM:TYPE<x>	ズームタイプを設定 / 問い合わせします。	5-261
:ZOOM:VERTical<x>?	垂直軸方向のズームに関するすべての設定値を問い合わせます。	5-261
:ZOOM:VERTical<x>:INITialize	垂直軸方向のズームの初期化をします。	5-261
:ZOOM:VERTical<x>:MAG	垂直軸方向のズーム倍率を設定 / 問い合わせします。	5-261

コマンド	機能	ページ
:ZOOM:VERTical<x>:POSition	垂直軸方向のズームポジションを設定 / 問い合わせします。	5-261
:ZOOM:VERTical<x>:TRACe	垂直軸方向のズーム画面に表示したいトレースを設定 / 問い合わせします。	5-261
:ZOOM:VLINKage	垂直軸方向リンクの ON/OFF を設定 / 問い合わせします。	5-262

### 共通コマンドグループ

*CAL? (CALibrate)	キャリブレーションを実行し、結果を問い合わせます。	5-263
*CLS (CLear Status)	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。	5-263
*ESE (standard Event Status Enable register)	標準イベントイネーブルレジスタの値を設定 / 問い合わせします。	5-263
*ESR? (standard Event Status Register)	標準イベントレジスタの値を問い合わせ、同時にクリアします。	5-263
*IDN? (IDeNtify)	機種を問い合わせます。	5-263
*LRN? (LeaRN)	以下のグループの現在の設定を一度に問い合わせます。	5-264
*OPC (OPeration Complete)	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 0(OPC ビット) を 1 にセットします。	5-265
*OPC? (OPeration Complete)	*OPC? を送信すると、指定したオーバーラップコマンドが終了していれば、ASCII コードの「1」を返します。	5-265
*OPT? (OPTion)	装備しているオプションを問い合わせます。	5-265
*PSC (Power-on Status Clear)	電源 ON 時に以下のレジスタをクリアするかしないかを設定 / 問い合わせします。整数に丸めた値が「0 以外」のときにクリアされます。	5-265
*RST (ReSeT)	設定の初期化 (イニシャライズ) をします。	5-265
*SRE (Service Request Enable register)	サービスリクエストイネーブルレジスタの値を設定 / 問い合わせします。	5-265
*STB? (STatus Byte)	ステータスバイトレジスタの値を問い合わせます。	5-265
*TST?	セルフテストを実行し、結果を問い合わせます。セルフテストの内容は、内 5-266 部のメインメモリテストです。	5-266
*WAI (WAIit)	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。	5-266

## 5.2 ACQUIRE グループ

### :ACQUIRE?

機能 波形の取り込みに関するすべての設定値を問い合わせます。

構文 :ACQUIRE?

例 :ACQUIRE? -> :ACQUIRE:AVERAGE:  
COUNT 2;EWEIGHT 16;:ACQUIRE:HRMODE 0;  
INTERLEAVE 0;INTERPOLATE 1;  
MODE NORMAL;REPETITIVE 0;  
RLENGTH 12500

### :ACQUIRE:AVERAGE?

機能 アベレーシングおよび波形の取り込み回数に関するすべての設定値を問い合わせます。

構文 :ACQUIRE:AVERAGE?

例 :ACQUIRE:AVERAGE? -> :ACQUIRE:  
AVERAGE:COUNT 2;EWEIGHT 16

### :ACQUIRE:AVERAGE:COUNT

機能 アベレーシングモード時の波形取り込み回数を設定 / 問い合わせします。

構文 :ACQUIRE:AVERAGE:COUNT {<NRF>}  
:ACQUIRE:AVERAGE:COUNT?  
<NRF> = 2 ~ 65536(2<sup>n</sup> ステップ)

例 :ACQUIRE:AVERAGE:COUNT 2  
:ACQUIRE:AVERAGE:COUNT? -> :ACQUIRE:  
AVERAGE:COUNT 2

### :ACQUIRE:AVERAGE:EWEIGHT

機能 アベレーシングモードを無限大行うときの減衰定数を設定 / 問い合わせします。

構文 :ACQUIRE:AVERAGE:EWEIGHT {<NRF>}  
:ACQUIRE:AVERAGE:EWEIGHT?  
<NRF> = 2 ~ 1024(2<sup>n</sup> ステップ)

例 :ACQUIRE:AVERAGE:EWEIGHT 16  
:ACQUIRE:AVERAGE:EWEIGHT?  
-> :ACQUIRE:AVERAGE:EWEIGHT 16

### :ACQUIRE:HRMODE

機能 高分解能モードの ON/OFF を設定 / 問い合わせします。

構文 :ACQUIRE:HRMODE {<Boolean>}  
:ACQUIRE:HRMODE?

例 :ACQUIRE:HRMODE ON  
:ACQUIRE:HRMODE? -> :ACQUIRE:HRMODE 1

### :ACQUIRE:INTERLEAVE

機能 インターリーブの ON/OFF を設定 / 問い合わせします。

構文 :ACQUIRE:INTERLEAVE {<Boolean>}  
:ACQUIRE:INTERLEAVE?

例 :ACQUIRE:INTERLEAVE ON  
:ACQUIRE:INTERLEAVE? -> :ACQUIRE:  
INTERLEAVE 1

### :ACQUIRE:INTERPOLATE

機能 データ補間の ON/OFF を設定 / 問い合わせします。

構文 :ACQUIRE:INTERPOLATE {<Boolean>}  
:ACQUIRE:INTERPOLATE?

例 :ACQUIRE:INTERPOLATE ON  
:ACQUIRE:INTERPOLATE? -> :ACQUIRE:  
INTERPOLATE 1

### :ACQUIRE:MODE

機能 波形取り込みモードを設定 / 問い合わせします。

構文 :ACQUIRE:MODE {AVERAGE|ENVELOPE|  
NORMAL}

例 :ACQUIRE:MODE? -> :ACQUIRE:  
MODE NORMAL

### :ACQUIRE:REPETITIVE

機能 等価時間サンプリングの ON/OFF を設定 / 問い合わせします。

構文 :ACQUIRE:REPETITIVE {<Boolean>}  
:ACQUIRE:REPETITIVE?

例 :ACQUIRE:REPETITIVE ON  
:ACQUIRE:REPETITIVE? -> :ACQUIRE:  
REPETITIVE 1

### :ACQUIRE:RLENGTH

機能 レコード長を設定 / 問い合わせします。

構文 :ACQUIRE:RLENGTH {<NRF>}  
:ACQUIRE:RLENGTH?  
<NRF> = 本体ユーザーズマニュアル参照。

例 :ACQUIRE:RLENGTH 6250000  
:ACQUIRE:RLENGTH?  
-> :ACQUIRE:RLENGTH 6250000

## 5.3 ANALysis グループ

### :ANALysis?

機能 解析機能に関するすべての設定値を問い合わせます。

構文 :ANALysis?

例

```
:ANALYSIS? -> :ANALYSIS:AHISTOGRAM1:
HORIZONTAL 0.000E+00,0.000E+00;
MEASURE:CURSOR:BASIC:C1:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
CURSOR:BASIC:C2:STATE 1;;ANALYSIS:
AHISTOGRAM1:MEASURE:CURSOR:BASIC:DC:
STATE 1;;ANALYSIS:AHISTOGRAM1:
MEASURE:CURSOR:CALCULATION:
DEFINE1 "C1";DEFINE2 "C2";
DEFINE3 "C1";DEFINE4 "C2";
STATE1 1;STATE2 1;STATE3 1;STATE4 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
HLINKAGE 1;HPOSITION1 0.000E+00;
HPOSITION2 1.000E+00;VLINKAGE 1;
VPOSITION1 1.000E+00;
VPOSITION2 1.000E+00;;ANALYSIS:
AHISTOGRAM1:MEASURE:MODE CURSOR;
PARAMETER:AREAL:MAXIMUM:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:MEAN:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:MEDIAN:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:MINIMUM:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:PEAK:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:RMS:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:SD2INTEG:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:SD3INTEG:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:SDEVIATION:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:SDINTEG:STATE 1.....
```

### :ANALysis:AHISTogram<x>?

機能 アキュムレートヒストグラム機能に関するすべての設定値を問い合わせます。

構文 :ANALysis:AHISTogram<x>?

<x> = 1、2

例

```
:ANALYSIS:AHISTOGRAM1? -> :ANALYSIS:
AHISTOGRAM1:HORIZONTAL 0.000E+00,
0.000E+00;MEASURE:CURSOR:BASIC:C1:
STATE 1;;ANALYSIS:AHISTOGRAM1:
MEASURE:CURSOR:BASIC:C2:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
CURSOR:BASIC:DC:STATE 1;;ANALYSIS:
AHISTOGRAM1:MEASURE:CURSOR:
CALCULATION:DEFINE1 "C1";
DEFINE2 "C2";DEFINE3 "C1";
DEFINE4 "C2";STATE1 1;STATE2 1;
STATE3 1;STATE4 1;;ANALYSIS:
AHISTOGRAM1:MEASURE:CURSOR:
HLINKAGE 1;HPOSITION1 0.000E+00;
HPOSITION2 1.000E+00;VLINKAGE 1;
VPOSITION1 1.000E+00;
VPOSITION2 1.000E+00;;ANALYSIS:
AHISTOGRAM1:MEASURE:MODE CURSOR;
PARAMETER:AREAL:MAXIMUM:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:MEAN:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:MEDIAN:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:MINIMUM:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:PEAK:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:RMS:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:SD2INTEG:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:SD3INTEG:STATE 1;;
ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:SDEVIATION:STATE
1;;ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREAL:SDINTEG:STATE 1.....
```

### :ANALysis:AHISTogram<x>:HORizontal

機能 アキュムレートヒストグラムの水平軸方向範囲を設定 / 問い合わせします。

構文 :ANALysis:AHISTogram<x>:HORizontal  
{<NRf>,<NRf>}

:ANALysis:AHISTogram<x>:HORizontal?

<x> = 1、2

<NRf> = -4 ~ 4(div)

例

```
:ANALYSIS:AHISTOGRAM1:HORIZONTAL 0,1
:ANALYSIS:AHISTOGRAM1:HORIZONTAL?
-> :ANALYSIS:AHISTOGRAM1:
HORIZONTAL 0.000E+00,0.000E+00
```

### 5.3 ANALysis グループ

#### **:ANALysis:AHISTogram<x>:MEASure?**

機能 アキュムレートヒストグラムの自動測定に関するすべての設定値を問い合わせます。

構文 :ANALysis:AHISTogram<x>:MEASure?  
<x> = 1, 2

例 :ANALYSIS:AHISTOGRAM1:MEASURE?  
-> :ANALYSIS:AHISTOGRAM1:MEASURE:  
CURSOR:BASIC:C1:STATE 1;:ANALYSIS:  
AHISTOGRAM1:MEASURE:CURSOR:BASIC:  
C2:STATE 1;:ANALYSIS:AHISTOGRAM1:  
MEASURE:CURSOR:BASIC:DC:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
CALCULATION:DEFINE1 "C1";  
DEFINE2 "C2";DEFINE3 "C1";  
DEFINE4 "C2";STATE1 1;STATE2 1;  
STATE3 1;STATE4 1;:ANALYSIS:  
AHISTOGRAM1:MEASURE:CURSOR:  
HLINKAGE 1;HPOSITION1 0.000E+00;  
HPOSITION2 1.000E+00;VLINKAGE 1;  
VPOSITION1 0.000E+00;  
VPOSITION2 1.000E+00;:ANALYSIS:  
AHISTOGRAM1:MEASURE:MODE CURSOR;  
PARAMETER:AREA1:MAXIMUM:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:MEAN:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:MEDIAN:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:MINIMUM:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:PEAK:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:RMS:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:SD2INTEG:  
STATE 1;:ANALYSIS:AHISTOGRAM1:  
MEASURE:PARAMETER:AREA1:SD3INTEG:  
STATE 1;:ANALYSIS:AHISTOGRAM1:  
MEASURE:PARAMETER:AREA1:SDEVIATION:  
STATE 1;:ANALYSIS:AHISTOGRAM1:  
MEASURE:PARAMETER:AREA1:SDINTEG:  
STATE 1.....

#### **:ANALysis:AHISTogram<x>:MEASure: CURSor?**

機能 アキュムレートヒストグラムのカーソル測定に関するすべての設定値を問い合わせます。

構文 :ANALysis:AHISTogram<x>:MEASure:  
CURSor?  
<x> = 1, 2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR?  
-> :ANALYSIS:AHISTOGRAM1:MEASURE:  
CURSOR:BASIC:C1:STATE 1;:ANALYSIS:  
AHISTOGRAM1:MEASURE:CURSOR:BASIC:C2:  
STATE 1;:ANALYSIS:AHISTOGRAM1:  
MEASURE:CURSOR:BASIC:DC:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
CALCULATION:DEFINE1 "C1";  
DEFINE2 "C2";DEFINE3 "C1";  
DEFINE4 "C2";STATE1 1;STATE2 1;  
STATE3 1;STATE4 1;:ANALYSIS:  
AHISTOGRAM1:MEASURE:CURSOR:  
HLINKAGE 1;HPOSITION1 0.000E+00;  
HPOSITION2 1.000E+00;VLINKAGE 1;  
VPOSITION1 0.000E+00;  
VPOSITION2 1.000E+00

#### **:ANALysis:AHISTogram<x>:MEASure: CURSor:BASic?**

機能 アキュムレートヒストグラムのカーソルの Basic アイテムに関するすべての設定値を問い合わせます。

構文 :ANALysis:AHISTogram<x>:MEASure:  
CURSor:BASic?  
<x> = 1, 2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
BASIC? -> :ANALYSIS:AHISTOGRAM1:  
MEASURE:CURSOR:BASIC:C1:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
BASIC:C2:STATE 1;:ANALYSIS:  
AHISTOGRAM1:MEASURE:CURSOR:BASIC:DC:  
STATE 1

#### **:ANALysis:AHISTogram<x>:MEASure: CURSor[:BASic]:ALL**

機能 アキュムレートヒストグラムのカーソルの Basic アイテムすべてを ON/OFF します。

構文 :ANALysis:AHISTogram<x>:MEASure:  
CURSor[:BASic]:ALL {<Boolean>}  
<x> = 1, 2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
BASIC:ALL ON

**:ANALysis:AHISTogram<x>:MEASure:  
CURSor[:BASic]:C<x>?**

機能 アキュムレートヒストグラムの各カーソルに関するすべての設定値を問い合わせます。

構文 :ANALysis:AHISTogram<x>:MEASure:  
CURSor[:BASic]:C<x>?

AHISTogram<x> の <x> = 1、2

C<x> の <x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
BASIC:C1? -> :ANALYSIS:AHISTOGRAM1:  
MEASURE:CURSOR:BASIC:C1:STATE 1

**:ANALysis:AHISTogram<x>:MEASure:  
CURSor[:BASic]:C<x>:STATe**

機能 アキュムレートヒストグラムの各カーソルの ON/OFF を設定 / 問い合わせします。

構文 :ANALysis:AHISTogram<x>:MEASure:  
CURSor[:BASic]:C<x>:STATe {<Boolean>}

:ANALysis:AHISTogram<x>:MEASure:

CURSor[:BASic]:C<x>:STATe?

AHISTogram<x> の <x> = 1、2

C<x> の <x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
BASIC:C1:STATE ON  
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
BASIC:C1:STATE? -> :ANALYSIS:  
AHISTOGRAM1:MEASURE:CURSOR:  
BASIC:C1:STATE 1

**:ANALysis:AHISTogram<x>:MEASure:  
CURSor[:BASic]:C<x>:VALue?**

機能 アキュムレートヒストグラムの各カーソルの測定値を問い合わせます。

構文 :ANALysis:AHISTogram<x>:MEASure:  
CURSor[:BASic]:C<x>:VALue?

AHISTogram<x> の <x> = 1、2

C<x> の <x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
BASIC:C1:VALUE? -> :ANALYSIS:  
AHISTOGRAM1:MEASURE:CURSOR:  
BASIC:C1:VALUE 1.000E+00

**:ANALysis:AHISTogram<x>:MEASure:  
CURSor[:BASic]:DC?**

機能 アキュムレートヒストグラムのカーソル間の測定値に関するすべての設定値を問い合わせます。

構文 :ANALysis:AHISTogram<x>:MEASure:  
CURSor[:BASic]:DC?

<x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
BASIC:DC? -> :ANALYSIS:AHISTOGRAM1:  
MEASURE:CURSOR:BASIC:DC:STATE 1

**:ANALysis:AHISTogram<x>:MEASure:  
CURSor[:BASic]:DC:STATe**

機能 アキュムレートヒストグラムのカーソル間の測定値の ON/OFF を設定 / 問い合わせします。

構文 :ANALysis:AHISTogram<x>:MEASure:  
CURSor[:BASic]:DC:STATe {<Boolean>}

:ANALysis:AHISTogram<x>:MEASure:

CURSor[:BASic]:DC:STATe?

<x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
BASIC:DC:STATE ON  
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
BASIC:DC:STATE? -> :ANALYSIS:  
AHISTOGRAM1:MEASURE:CURSOR:BASIC:DC:  
STATE 1

**:ANALysis:AHISTogram<x>:MEASure:  
CURSor[:BASic]:DC:VALue?**

機能 アキュムレートヒストグラムのカーソル間の測定値を問い合わせます。

構文 :ANALysis:AHISTogram<x>:MEASure:  
CURSor[:BASic]:DC:VALue?

<x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
BASIC:DC:VALUE? -> :ANALYSIS:  
AHISTOGRAM1:MEASURE:CURSOR:  
BASIC:DC:VALUE 1.000E+00

**:ANALysis:AHISTogram<x>:MEASure:  
CURSor:CALCulation?**

機能 アキュムレートヒストグラムのカーソルの Calc アイテムに関するすべての設定値を問い合わせます。

構文 :ANALysis:AHISTogram<x>:MEASure:  
CURSor:CALCulation?

<x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
CALCULATION? -> :ANALYSIS:  
AHISTOGRAM1:MEASURE:CURSOR:  
CALCULATION:DEFINE1 "C1";  
DEFINE2 "C2";DEFINE3 "C1";  
DEFINE4 "C2";STATE1 1;STATE2 1;  
STATE3 1;STATE4 1

**:ANALysis:AHISTogram<x>:MEASure:  
CURSor:CALCulation:ALL**

機能 アキュムレートヒストグラムのカーソルの Calc アイテムすべてを ON/OFF します。

構文 :ANALysis:AHISTogram<x>:MEASure:  
CURSor:CALCulation:ALL {<Boolean>}

<x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
CALCULATION:ALL ON



### 5.3 ANALYSIS グループ

#### **:ANALYSIS:AHISTogram<x>:MEASURE:**

##### **CURSOr:CALCulation:DEFine<x>**

機能 アキュムレートヒストグラムのカーソルの Calc アイテムの演算式を設定 / 問い合わせします。

構文 :ANALYSIS:AHISTogram<x>:MEASURE:  
CURSOR:CALCulation:  
DEFine<x> {<文字列>}  
:ANALYSIS:AHISTogram<x>:MEASURE:  
CURSOR:CALCulation:DEFine<x>?  
AHISTogram<x> の <x> = 1、2  
DEFine<x> の <x> = 1 ~ 4  
<文字列> = 128 文字以内

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
CALCULATION:DEFINE1 "C1"  
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
CALCULATION:DEFINE1? -> :ANALYSIS:  
AHISTOGRAM1:MEASURE:CURSOR:  
CALCULATION:DEFINE1 "C1"

#### **:ANALYSIS:AHISTogram<x>:MEASURE:**

##### **CURSOr:CALCulation:STATE<x>**

機能 アキュムレートヒストグラムのカーソルの Calc アイテムの ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:AHISTogram<x>:MEASURE:  
CURSOR:CALCulation:  
STATE<x> {<Boolean>}  
:ANALYSIS:AHISTogram<x>:MEASURE:  
CURSOR:CALCulation:STATE<x>?  
AHISTogram<x> の <x> = 1、2  
STATE<x> の <x> = 1 ~ 4

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
CALCULATION:STATE1 ON  
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
CALCULATION:STATE1? -> :ANALYSIS:  
AHISTOGRAM1:MEASURE:CURSOR:  
CALCULATION:STATE1 1

#### **:ANALYSIS:AHISTogram<x>:MEASURE:**

##### **CURSOr:CALCulation:VALUE<x>?**

機能 アキュムレートヒストグラムのカーソルの Calc アイテムの測定値を問い合わせします。

構文 :ANALYSIS:AHISTogram<x>:MEASURE:  
CURSOR:  
CALCulation:VALUE<x>?  
AHISTogram<x> の <x> = 1、2  
VALUE<x> の <x> = 1 ~ 4

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
CALCULATION:VALUE1? -> :ANALYSIS:  
AHISTOGRAM1:MEASURE:CURSOR:  
CALCULATION:VALUE1 1.000E+00

#### **:ANALYSIS:AHISTogram<x>:MEASURE:**

##### **CURSOr:HLINKage**

機能 アキュムレートヒストグラムの水平カーソルの リンクの ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:AHISTogram<x>:MEASURE:  
CURSOR:HLINKage {<Boolean>}  
:ANALYSIS:AHISTogram<x>:MEASURE:  
CURSOR:HLINKage?  
<x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
HLINKAGE ON  
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
HLINKAGE? -> :ANALYSIS:AHISTOGRAM1:  
MEASURE:CURSOR:HLINKAGE 1

#### **:ANALYSIS:AHISTogram<x>:MEASURE:**

##### **CURSOr:HPOSITION<x>**

機能 アキュムレートヒストグラムの各水平カーソルの位置を設定 / 問い合わせします。

構文 :ANALYSIS:AHISTogram<x>:MEASURE:  
CURSOR:  
HPOSITION<x> {<NRF>}  
:ANALYSIS:AHISTogram<x>:MEASURE:  
CURSOR:HPOSITION<x>?  
AHISTogram<x> の <x> = 1、2  
HPOSITION<x> の <x> = 1、2  
<NRF> = - 5 ~ 5(div)

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
HPOSITION1 1  
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
HPOSITION1? -> :ANALYSIS:AHISTOGRAM1:  
MEASURE:CURSOR:HPOSITION1 1.000E+00

#### **:ANALYSIS:AHISTogram<x>:MEASURE:**

##### **CURSOr:VLINKage**

機能 アキュムレートヒストグラムの垂直カーソルの リンクの ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:AHISTogram<x>:MEASURE:  
CURSOR:  
VLINKage {<Boolean>}  
:ANALYSIS:AHISTogram<x>:MEASURE:  
CURSOR:VLINKage?  
<x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
VLINKAGE ON  
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
VLINKAGE? -> :ANALYSIS:AHISTOGRAM1:  
MEASURE:CURSOR:VLINKAGE 1

**:ANALYSIS:AHISTogram<x>:MEASURE:  
CURSOR:VPOSITION<x>**

機能 アキュムレートヒストグラムの各垂直カーソルの位置を設定 / 問い合わせします。

構文 :ANALYSIS:AHISTogram<x>:MEASURE:  
CURSOR:VPOSITION<x> {<Nrf>}  
:ANALYSIS:AHISTogram<x>:MEASURE:  
CURSOR:VPOSITION<x>?

AHISTogram<x> の <x> = 1、2

VPOSITION<x> の <x> = 1、2

<Nrf> = - 4 ~ 4(div)

例 :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
VPOSITION1 1  
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:  
VPOSITION1? -> :ANALYSIS:AHISTOGRAM1:  
MEASURE:CURSOR:VPOSITION1 1.000E+00

**:ANALYSIS:AHISTogram<x>:MEASURE:MODE**

機能 アキュムレートヒストグラムの自動測定の方法を設定 / 問い合わせします。

構文 :ANALYSIS:AHISTogram<x>:MEASURE:MODE  
{CURSOR|OFF|PARAMETER}  
:ANALYSIS:AHISTogram<x>:MEASURE:MODE?  
<x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:  
MODE CURSOR  
:ANALYSIS:AHISTOGRAM1:MEASURE:MODE?  
-> :ANALYSIS:AHISTOGRAM1:MEASURE:  
MODE CURSOR

**:ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER?**

機能 アキュムレートヒストグラムの波形パラメータの自動測定に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER?  
<x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER? -> :ANALYSIS:AHISTOGRAM1:  
MEASURE:PARAMETER:AREA1:MAXIMUM:  
STATE 1;:ANALYSIS:AHISTOGRAM1:  
MEASURE:PARAMETER:AREA1:MEAN:  
STATE 1;:ANALYSIS:AHISTOGRAM1:  
MEASURE:PARAMETER:AREA1:MEDIAN:  
STATE 1;:ANALYSIS:AHISTOGRAM1:  
MEASURE:PARAMETER:AREA1:MINIMUM:  
STATE 1;:ANALYSIS:AHISTOGRAM1:  
MEASURE:PARAMETER:AREA1:PEAK:  
STATE 1;:ANALYSIS:AHISTOGRAM1:  
MEASURE:PARAMETER:AREA1:RMS:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:SD2INTEG:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:SD3INTEG:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:SDEVIATION:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:SDINTEG:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA2:MAXIMUM:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA2:MEAN:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA2:MEDIAN:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA2:MINIMUM:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA2:PEAK:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA2:RMS:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA2:SD2INTEG:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA2:SD3INTEG:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA2:SDEVIATION:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA2:SDINTEG:STATE 1.....

### 5.3 ANALysis グループ

#### **:ANALysis:AHISTogram<x>:MEASure:**

##### **PARAMeter:AREA<x>?**

機能 アキュムレートヒストグラムの各エリアに関するすべての設定値を問い合わせます。

構文 :ANALysis:AHISTogram<x>:MEASure:  
PARAMeter:AREA<x>?

AHISTogram<x> の <x> = 1、2

AREA<x> の <x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1? -> :ANALYSIS:  
AHISTOGRAM1:MEASURE:PARAMETER:  
AREA1:MAXIMUM:STATE 1;:ANALYSIS:  
AHISTOGRAM1:MEASURE:PARAMETER:  
AREA1:MEAN:STATE 1;:ANALYSIS:  
AHISTOGRAM1:MEASURE:PARAMETER:AREA1:  
MEDIAN:STATE 1;:ANALYSIS:AHISTOGRAM1:  
MEASURE:PARAMETER:AREA1:MINIMUM:  
STATE 1;:ANALYSIS:AHISTOGRAM1:  
MEASURE:PARAMETER:AREA1:PEAK:  
STATE 1;:ANALYSIS:AHISTOGRAM1:  
MEASURE:PARAMETER:AREA1:RMS:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:SD2INTEG:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:SD3INTEG:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:SDEVIATION:STATE 1;:  
ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:SDINTEG:STATE 1

#### **:ANALysis:AHISTogram<x>:MEASure:**

##### **PARAMeter:AREA<x>:ALL**

機能 アキュムレートヒストグラムのすべての波形パラメータを一斉に ON/OFF します。

構文 :ANALysis:AHISTogram<x>:MEASure:  
PARAMeter:AREA<x>:ALL {<Boolean>}

AHISTogram<x> の <x> = 1、2

AREA<x> の <x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:ALL ON

#### **:ANALysis:AHISTogram<x>:MEASure:**

##### **PARAMeter:AREA<x>:<パラメータ>?**

機能 アキュムレートヒストグラムの波形パラメータに関するすべての設定値を問い合わせます。

構文 :ANALysis:AHISTogram<x>:MEASure:  
PARAMeter:AREA<x>:<パラメータ>?

AHISTogram<x> の <x> = 1、2

AREA<x> の <x> = 1、2

<パラメータ> = {MAXimum|MEAN|MEDian|  
MINimum|PEAK|RMS|SD2integ|SD3integ|  
SDEVIation|SDINteG}

例 (以下は、エリア 1 の最大値についての例です。)

:ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:MAXIMUM?  
-> :ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:MAXIMUM:STATE 1

#### **:ANALysis:AHISTogram<x>:MEASure:**

##### **PARAMeter:AREA<x>:<パラメータ>:STATE**

機能 アキュムレートヒストグラムの波形パラメータの ON/OFF を設定 / 問い合わせします。

構文 :ANALysis:AHISTogram<x>:MEASure:  
PARAMeter:AREA<x>:<パラメータ>:  
STATE {<Boolean>}

:ANALysis:AHISTogram<x>:MEASure:

PARAMeter:AREA<x>:<パラメータ>:STATE?

AHISTogram<x> の <x> = 1、2

AREA<x> の <x> = 1、2

<パラメータ> = {MAXimum|MEAN|MEDian|  
MINimum|PEAK|RMS|SD2integ|SD3integ|  
SDEVIation|SDINteG}

例 (以下は、エリア 1 の最大値についての例です。)

:ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:MAXIMUM:STATE ON  
:ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:MAXIMUM:STATE?  
-> :ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:MAXIMUM:STATE 1

#### **:ANALysis:AHISTogram<x>:MEASure:**

##### **PARAMeter:AREA<x>:<パラメータ>:VALUE?**

機能 アキュムレートヒストグラムの波形パラメータの自動測定値を問い合わせます。

構文 :ANALysis:AHISTogram<x>:MEASure:  
PARAMeter:AREA<x>:<パラメータ>:VALUE?

AHISTogram<x> の <x> = 1、2

AREA<x> の <x> = 1、2

<パラメータ> = {MAXimum|MEAN|MEDian|  
MINimum|PEAK|RMS|SD2integ|SD3integ|  
SDEVIation|SDINteG}

例 (以下は、エリア 1 の最大値についての例です。)

:ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:MAXIMUM:VALUE?  
-> :ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:AREA1:MAXIMUM:  
VALUE 1.000E+00

**:ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER:CALCulation?**

機能 アキュムレートヒストグラムの波形パラメータの Calc アイテムに関するすべての設定値を問い合わせます。

構文 :ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER:CALCulation?  
<x> = 1, 2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:CALCULATION? -> :ANALYSIS:  
AHISTOGRAM1:MEASURE:PARAMETER:  
CALCULATION:DEFINE1 "MEAN";  
DEFINE2 "MAX";DEFINE3 "MIN";  
DEFINE4 "PEAK";STATE1 1;STATE2 1;  
STATE3 1;STATE4 1

**:ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER:CALCulation:ALL**

機能 アキュムレートヒストグラムの波形パラメータのすべての Calc アイテムを一斉に ON/OFF します。

構文 :ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER:CALCulation:ALL {<Boolean>}  
<x> = 1, 2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:CALCULATION:ALL ON

**:ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER:CALCulation:DEFine<x>**

機能 アキュムレートヒストグラムの波形パラメータの Calc アイテムの演算式を設定 / 問い合わせします。

構文 :ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER:CALCulation:  
DEFine<x> {<文字列>}  
:ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER:CALCulation:DEFine<x>?  
AHISTogram<x> の <x> = 1, 2  
DEFine<x> の <x> = 1 ~ 4  
<文字列> = 128 文字以内

例 :ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:CALCULATION:DEFINE1 "MEAN"  
:ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:CALCULATION:DEFINE1?  
-> :ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:CALCULATION:DEFINE1 "MEAN"

**:ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER:CALCulation:STATE<x>**

機能 アキュムレートヒストグラムの波形パラメータの Calc アイテムの ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER:CALCulation:  
STATE<x> {<Boolean>}

例 :ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER:CALCulation:STATE<x>?  
AHISTogram<x> の <x> = 1, 2  
STATE<x> の <x> = 1 ~ 4

例 :ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:CALCULATION:STATE1 ON  
:ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:CALCULATION:STATE1?  
-> :ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:CALCULATION:STATE1 1

**:ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER:CALCulation:VALue<x>?**

機能 アキュムレートヒストグラムの波形パラメータの Calc アイテムの自動測定値を問い合わせます。

構文 :ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER:CALCulation:VALue<x>?  
AHISTogram<x> の <x> = 1, 2  
VALue<x> の <x> = 1 ~ 4

例 :ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:CALCULATION:VALUE1?  
-> :ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:CALCULATION:VALUE1  
1.000E+00

**:ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER:HRANge<x>**

機能 アキュムレートヒストグラムの波形パラメータの水平範囲を設定 / 問い合わせします。

構文 :ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER:HRANge<x> {<NRF>,<NRF>}  
:ANALYSIS:AHISTogram<x>:MEASURE:  
PARAMETER:HRANge<x>?  
AHISTogram<x> の <x> = 1, 2  
HRANge<x> の <x> = 1, 2  
<NRF> = 5 ~ 5(div)

例 :ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:HRANGE1 1,2  
:ANALYSIS:AHISTOGRAM1:MEASURE:  
PARAMETER:HRANGE1? -> :ANALYSIS:  
AHISTOGRAM1:MEASURE:PARAMETER:  
HRANGE1 2.000E+00,1.000E+00

### 5.3 ANALYSIS グループ

#### **:ANALYSIS:AHISTOGRAM<x>:MEASURE:PARAMETER:VRANGE<x>**

機能 アキュムレートヒストグラムの波形パラメータの垂直範囲を設定 / 問い合わせします。

構文 :ANALYSIS:AHISTOGRAM<x>:MEASURE:PARAMETER:VRANGE<x> {<NRf>,<NRf>}  
:ANALYSIS:AHISTOGRAM<x>:MEASURE:PARAMETER:VRANGE<x>?  
AHISTOGRAM<x> の <x> = 1、2  
VRANGE<x> の <x> = 1、2  
<NRf> = -4 ~ 4(div)

例 :ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:VRANGE1 1,2  
:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:VRANGE1? -> :ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:VRANGE1 2.000E+00,1.000E+00

#### **:ANALYSIS:AHISTOGRAM<x>:MODE**

機能 アキュムレートヒストグラムのモードを設定 / 問い合わせします。

構文 :ANALYSIS:AHISTOGRAM<x>:MODE {HORIZONTAL|VERTICAL}  
:ANALYSIS:AHISTOGRAM<x>:MODE?  
<x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:MODE HORIZONTAL  
:ANALYSIS:AHISTOGRAM1:MODE?  
-> :ANALYSIS:AHISTOGRAM1:MODE HORIZONTAL

#### **:ANALYSIS:AHISTOGRAM<x>:TRACE**

機能 アキュムレートヒストグラムの対象トレースを設定 / 問い合わせします。

構文 :ANALYSIS:AHISTOGRAM<x>:TRACE {<NRf>}  
:ANALYSIS:AHISTOGRAM<x>:TRACE?  
<x> = 1、2  
<NRf> = 1 ~ 8

例 :ANALYSIS:AHISTOGRAM1:TRACE 1  
:ANALYSIS:AHISTOGRAM1:TRACE?  
-> :ANALYSIS:AHISTOGRAM1:TRACE 1

#### **:ANALYSIS:AHISTOGRAM<x>:VERTICAL**

機能 アキュムレートヒストグラムの垂直範囲を設定 / 問い合わせします。

構文 :ANALYSIS:AHISTOGRAM<x>:VERTICAL {<NRf>,<NRf>}  
:ANALYSIS:AHISTOGRAM<x>:VERTICAL?  
<x> = 1、2  
<NRf> = -4 ~ 4(div)

例 :ANALYSIS:AHISTOGRAM1:VERTICAL 1,2  
:ANALYSIS:AHISTOGRAM1:VERTICAL?  
-> :ANALYSIS:AHISTOGRAM1:VERTICAL 2.000E+00,1.000E+00

#### **:ANALYSIS:AHISTOGRAM<x>:WINDOW**

機能 アキュムレートヒストグラムの測定対象ウィンドウを設定 / 問い合わせします。

構文 :ANALYSIS:AHISTOGRAM<x>:WINDOW {MAIN|Z1|Z2}  
:ANALYSIS:AHISTOGRAM<x>:WINDOW?  
<x> = 1、2

例 :ANALYSIS:AHISTOGRAM1:WINDOW MAIN  
:ANALYSIS:AHISTOGRAM1:WINDOW?  
-> :ANALYSIS:AHISTOGRAM1:WINDOW MAIN

#### **:ANALYSIS:DISPLAY<x>**

機能 解析機能の表示の ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:DISPLAY<x> {<Boolean>}  
:ANALYSIS:DISPLAY<x>?  
<x> = 1、2

例 :ANALYSIS:DISPLAY1 ON  
:ANALYSIS:DISPLAY1? -> :ANALYSIS:DISPLAY1 1

**:ANALYSIS:FFT<x>?**

機能 FFT 演算機能に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>?

<x> = 1, 2

例 :ANALYSIS:FFT1? -> :ANALYSIS:FFT1:  
HORIZONTAL:CSPAN:CENTER 1.000E+00;  
SPAN 1.000E+00;;ANALYSIS:FFT1:  
HORIZONTAL:LRIGHT:RANGE 2.000E+00,  
1.000E+00;;ANALYSIS:FFT1:HORIZONTAL:  
MODE AUTO;;ANALYSIS:FFT1:IPART 1;  
LENGTH 2500;MAXHOLD 1;MEASURE:MARKER:  
BASIC:DFREQUENCY:STATE 1;;ANALYSIS:  
FFT1:MEASURE:MARKER:BASIC:DV:  
STATE 1;;ANALYSIS:FFT1:MEASURE:  
MARKER:BASIC:FREQUENCY1:STATE 1;;  
ANALYSIS:FFT1:MEASURE:MARKER:BASIC:  
FREQUENCY2:STATE 1;;ANALYSIS:FFT1:  
MEASURE:MARKER:BASIC:  
POSITION1 1.000E+00;;ANALYSIS:FFT1:  
MEASURE:MARKER:BASIC:  
POSITION2 2.000E+00;V1:STATE 1;;  
ANALYSIS:FFT1:MEASURE:MARKER:BASIC:  
V2:STATE 1;;ANALYSIS:FFT1:MEASURE:  
MARKER:CALCULATION:DEFINE1 "V(F1)";  
DEFINE2 "V(F2)";DEFINE3 "V(F1)";  
DEFINE4 "V(F2)";STATE1 1;STATE2 1;  
STATE3 1;STATE4 1;;ANALYSIS:FFT1:  
MEASURE:MODE MARKER;PEAK:BASIC:  
DFREQUENCY:STATE 1;;ANALYSIS:FFT1:  
MEASURE:PEAK:BASIC:DV:STATE 1;;  
ANALYSIS:FFT1:MEASURE:PEAK:BASIC:  
FREQUENCY1:STATE 1;;ANALYSIS:FFT1:  
MEASURE:PEAK:BASIC:FREQUENCY2:  
STATE 1;;ANALYSIS:FFT1:MEASURE:PEAK:  
BASIC:RANGE1 1.000E+00,0.000E+00;  
RANGE2 1.000E+00,0.000E+00;V1:  
STATE 1;;ANALYSIS:FFT1:MEASURE:PEAK:  
BASIC:V2:STATE 1.....

**:ANALYSIS:FFT<x>:HORIZONTAL?**

機能 FFT 演算の水平軸に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:HORIZONTAL?

<x> = 1, 2

例 :ANALYSIS:FFT1:HORIZONTAL?  
-> :ANALYSIS:FFT1:HORIZONTAL:CSPAN:  
CENTER 1.000E+00;SPAN 1.000E+00;;  
ANALYSIS:FFT1:HORIZONTAL:LRIGHT:  
RANGE 2.000E+00,1.000E+00;;ANALYSIS:  
FFT1:HORIZONTAL:MODE AUTO

**:ANALYSIS:FFT<x>:HORIZONTAL:CSPAN?**

機能 FFT 演算の水平軸の中心値 / スパンに関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:HORIZONTAL:CSPAN?

<x> = 1, 2

例 :ANALYSIS:FFT1:HORIZONTAL:CSPAN?  
-> :ANALYSIS:FFT1:HORIZONTAL:CSPAN:  
CENTER 1.000E+00;SPAN 1.000E+00

**:ANALYSIS:FFT<x>:HORIZONTAL:CSPAN: CENTER**

機能 FFT 演算の水平軸の中心値を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:HORIZONTAL:CSPAN:

CENTER {<周波数>}

:ANALYSIS:FFT<x>:HORIZONTAL:CSPAN:  
CENTER?

<x> = 1, 2

<周波数> = 0 ~ 250G(Hz)

例 :ANALYSIS:FFT1:HORIZONTAL:CSPAN:  
CENTER 1HZ  
:ANALYSIS:FFT1:HORIZONTAL:CSPAN:  
CENTER? -> :ANALYSIS:FFT1:HORIZONTAL:  
CSPAN:CENTER 1.000E+00

**:ANALYSIS:FFT<x>:HORIZONTAL:CSPAN: SPAN**

機能 FFT 演算の水平軸のスパンを設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:HORIZONTAL:CSPAN:

SPAN {<周波数>}

:ANALYSIS:FFT<x>:HORIZONTAL:CSPAN:  
SPAN?

<x> = 1, 2

<周波数> = 0 ~ 250G(Hz)

例 :ANALYSIS:FFT1:HORIZONTAL:CSPAN:  
SPAN 1HZ  
:ANALYSIS:FFT1:HORIZONTAL:CSPAN:SPAN?  
-> :ANALYSIS:FFT1:HORIZONTAL:CSPAN:  
SPAN 1.000E+00

**:ANALYSIS:FFT<x>:HORIZONTAL:LRIGHT?**

機能 FFT 演算の水平軸の左右端に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:HORIZONTAL:LRIGHT?

<x> = 1, 2

例 :ANALYSIS:FFT1:HORIZONTAL:LRIGHT?  
-> :ANALYSIS:FFT1:HORIZONTAL:LRIGHT:  
RANGE 2.000E+00,1.000E+00

### 5.3 ANALYSIS グループ

#### :ANALYSIS:FFT<x>:HORIZONTAL:LRIGHT:

##### RANGE

機能 FFT 演算の水平軸の左右端の範囲を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:HORIZONTAL:LRIGHT:

RANGE {<周波数>,<周波数>}

:ANALYSIS:FFT<x>:HORIZONTAL:LRIGHT:

RANGE?

<x> = 1, 2

<周波数> = 0 ~ 250G(Hz)

例 :ANALYSIS:FFT1:HORIZONTAL:LRIGHT:

RANGE 1HZ,2HZ

:ANALYSIS:FFT1:HORIZONTAL:LRIGHT:

RANGE? -> :ANALYSIS:FFT1:HORIZONTAL:

LRIGHT:RANGE 2.000E+00,1.000E+00

#### :ANALYSIS:FFT<x>:HORIZONTAL:MODE

機能 FFT 演算の水平軸のモードを設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:HORIZONTAL:

MODE {AUTO|CSPan|LRIGHT}

:ANALYSIS:FFT<x>:HORIZONTAL:MODE?

<x> = 1, 2

例 :ANALYSIS:FFT1:HORIZONTAL:MODE AUTO

:ANALYSIS:FFT1:HORIZONTAL:MODE?

-> :ANALYSIS:FFT1:HORIZONTAL:

MODE AUTO

#### :ANALYSIS:FFT<x>:IPART (Imag Part)

機能 FFT 演算のイマジナリーパート対象トレースを設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:IPART

{<NRf>|DONTcare}

:ANALYSIS:FFT<x>:IPART?

<x> = 1, 2

<NRf> = 1 ~ 8

例 :ANALYSIS:FFT1:IPART 1

:ANALYSIS:FFT1:IPART? -> :ANALYSIS:

FFT1:IPART 1

#### :ANALYSIS:FFT<x>:LENGTH

機能 FFT 演算の FFT 点数を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:LENGTH {<NRf>}

:ANALYSIS:FFT<x>:LENGTH?

<x> = 1, 2

<NRf> = 2500、6250、12500、25000、62500、125000、250000

例 :ANALYSIS:FFT1:LENGTH 2500

:ANALYSIS:FFT1:LENGTH? -> :ANALYSIS:

FFT1:LENGTH 2500

#### :ANALYSIS:FFT<x>:MAXHold

機能 FFT 演算の最大値保持機能の ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MAXHold {<Boolean>}

:ANALYSIS:FFT<x>:MAXHold?

<x> = 1, 2

例 :ANALYSIS:FFT1:MAXHOLD ON

:ANALYSIS:FFT1:MAXHOLD? -> :ANALYSIS:

FFT1:MAXHOLD 1

#### :ANALYSIS:FFT<x>:MEASURE?

機能 FFT 演算の自動測定に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE?

<x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE? -> :ANALYSIS:

FFT1:MEASURE:MARKER:BASIC:DFREQUENCY:

STATE 1;:ANALYSIS:FFT1:MEASURE:

MARKER:BASIC:DV:STATE 1;:ANALYSIS:

FFT1:MEASURE:MARKER:BASIC:FREQUENCY1:

STATE 1;:ANALYSIS:FFT1:MEASURE:

MARKER:BASIC:FREQUENCY2:STATE 1;:

ANALYSIS:FFT1:MEASURE:MARKER:BASIC:

POSITION1 1.000E+00;:ANALYSIS:FFT1:

MEASURE:MARKER:BASIC:

POSITION2 2.000E+00;V1:STATE 1;:

ANALYSIS:FFT1:MEASURE:MARKER:BASIC:

V2:STATE 1;:ANALYSIS:FFT1:MEASURE:

MARKER:CALCULATION:DEFINE1 "V(F1)";

DEFINE2 "V(F2)";DEFINE3 "V(F1)";

DEFINE4 "V(F2)";STATE1 1;STATE2 1;

STATE3 1;STATE4 1;:ANALYSIS:FFT1:

MEASURE:MODE MARKER;PEAK:BASIC:

DFREQUENCY:STATE 1;:ANALYSIS:FFT1:

MEASURE:PEAK:BASIC:DV:STATE 1;:

ANALYSIS:FFT1:MEASURE:PEAK:BASIC:

FREQUENCY1:STATE 1;:ANALYSIS:FFT1:

MEASURE:PEAK:BASIC:FREQUENCY2:

STATE 1;:ANALYSIS:FFT1:MEASURE:PEAK:

BASIC:RANGE1 1.000E+00,0.000E+00;

RANGE2 1.000E+00,0.000E+00;V1:

STATE 1;:ANALYSIS:FFT1:MEASURE:PEAK:

BASIC:V2:STATE 1;:ANALYSIS:FFT1:

MEASURE:PEAK:CALCULATION:

DEFINE1 "V(P1)";DEFINE2 "V(P2)";

DEFINE3 "V(P1)";DEFINE4 "V(P2)";

STATE1 1;STATE2 1;STATE3 1;STATE4 1

**:ANALYSIS:FFT<x>:MEASURE:MARKER?**

機能 FFT 演算のマーカーカーソル測定に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:MARKER?

<x> = 1, 2

例

```
:ANALYSIS:FFT1:MEASURE:MARKER?
-> :ANALYSIS:FFT1:MEASURE:MARKER:
BASIC:DFREQUENCY:STATE 1;:ANALYSIS:
FFT1:MEASURE:MARKER:BASIC:DV:
STATE 1;:ANALYSIS:FFT1:MEASURE:
MARKER:BASIC:FREQUENCY1:STATE 1;:
ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY2:STATE 1;:ANALYSIS:FFT1:
MEASURE:MARKER:BASIC:
POSITION1 1.000E+00;:ANALYSIS:FFT1:
MEASURE:MARKER:BASIC:
POSITION2 2.000E+00;V1:STATE 1;:
ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
V2:STATE 1;:ANALYSIS:FFT1:MEASURE:
MARKER:CALCULATION:DEFINE1 "V(F1)";
DEFINE2 "V(F2)";DEFINE3 "V(F1)";
DEFINE4 "V(F2)";STATE1 1;STATE2 1;
STATE3 1;STATE4 1
```

**:ANALYSIS:FFT<x>:MEASURE:MARKER: BASIC?**

機能 FFT 演算のマーカーカーソルの Basic アイテムに関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:MARKER:

BASIC?

<x> = 1, 2

例

```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC?
-> :ANALYSIS:FFT1:MEASURE:MARKER:
BASIC:DFREQUENCY:STATE 1;:ANALYSIS:
FFT1:MEASURE:MARKER:BASIC:DV:
STATE 1;:ANALYSIS:FFT1:MEASURE:
MARKER:BASIC:FREQUENCY1:STATE 1;:
ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY2:STATE 1;:ANALYSIS:FFT1:
MEASURE:MARKER:BASIC:
POSITION1 1.000E+00;:ANALYSIS:FFT1:
MEASURE:MARKER:BASIC:
POSITION2 2.000E+00;V1:STATE 1;:
ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
V2:STATE 1
```

**:ANALYSIS:FFT<x>:MEASURE:****MARKER[:BASIC]:ALL**

機能 FFT 演算のマーカーカーソルのすべての Basic アイテムを一斉に ON/OFF します。

構文 :ANALYSIS:FFT<x>:MEASURE:

MARKER[:BASIC]:ALL {<Boolean>}

<x> = 1, 2

例

```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
ALL ON
```

**:ANALYSIS:FFT<x>:MEASURE:****MARKER[:BASIC]:DFREQUENCY?**

機能 FFT 演算のマーカーカーソル間の周波数値に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:

MARKER[:BASIC]:DFREQUENCY?

<x> = 1, 2

例

```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
DFREQUENCY? -> :ANALYSIS:FFT1:
MEASURE:MARKER:BASIC:DFREQUENCY:
STATE 1
```

**:ANALYSIS:FFT<x>:MEASURE:****MARKER[:BASIC]:DFREQUENCY:STATE**

機能 FFT 演算のマーカーカーソル間の周波数値の ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MEASURE:

MARKER[:BASIC]:DFREQUENCY:STATE

{<Boolean>}

:ANALYSIS:FFT<x>:MEASURE:

MARKER[:BASIC]:DFREQUENCY:STATE?

<x> = 1, 2

例

```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
DFREQUENCY:STATE ON
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
DFREQUENCY:STATE? -> :ANALYSIS:FFT1:
MEASURE:MARKER:BASIC:DFREQUENCY:
STATE 1
```

**:ANALYSIS:FFT<x>:MEASURE:****MARKER[:BASIC]:DFREQUENCY:VALUE?**

機能 FFT 演算のマーカーカーソル間の周波数値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:

MARKER[:BASIC]:DFREQUENCY:VALUE?

<x> = 1, 2

例

```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
DFREQUENCY:VALUE? -> :ANALYSIS:FFT1:
MEASURE:MARKER:BASIC:DFREQUENCY:
VALUE 1.000E+00
```

**:ANALYSIS:FFT<x>:MEASURE:****MARKER[:BASIC]:DV?**

機能 FFT 演算のマーカーカーソル間のパワー値に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:

MARKER[:BASIC]:DV?

<x> = 1, 2

例

```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
DV? -> :ANALYSIS:FFT1:MEASURE:MARKER:
BASIC:DV:STATE 1
```



### 5.3 ANALYSIS グループ

#### **:ANALYSIS:FFT<x>:MEASURE:**

##### **MARKER[:BASiC]:DV:STATE**

機能 FFT 演算のマーカーカーソル間のパワー値の ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MEASURE:  
MARKER[:BASiC]:DV:STATE {<Boolean>}  
:ANALYSIS:FFT<x>:MEASURE:  
MARKER[:BASiC]:DV:STATE?  
<x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:MARKER:BASiC:  
DV:STATE ON  
:ANALYSIS:FFT1:MEASURE:MARKER:BASiC:  
DV:STATE? -> :ANALYSIS:FFT1:MEASURE:  
MARKER:BASiC:DV:STATE 1

#### **:ANALYSIS:FFT<x>:MEASURE:**

##### **MARKER[:BASiC]:DV:VALUE?**

機能 FFT 演算のマーカーカーソル間のパワー値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:  
MARKER[:BASiC]:DV:VALUE?  
<x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:MARKER:BASiC:  
DV:VALUE? -> :ANALYSIS:FFT1:MEASURE:  
MARKER:BASiC:DV:VALUE 1.000E+00

#### **:ANALYSIS:FFT<x>:MEASURE:**

##### **MARKER[:BASiC]:FREQUENCY<x>?**

機能 FFT 演算の各マーカーカーソルの周波数値に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:  
MARKER[:BASiC]:FREQUENCY<x>?  
FFT<x> の <x> = 1, 2  
FREQUENCY<x> の <x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:MARKER:BASiC:  
FREQUENCY1? -> :ANALYSIS:FFT1:  
MEASURE:MARKER:BASiC:FREQUENCY1:  
STATE 1

#### **:ANALYSIS:FFT<x>:MEASURE:**

##### **MARKER[:BASiC]:FREQUENCY<x>:STATE**

機能 FFT 演算の各マーカーカーソルの周波数値の ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MEASURE:  
MARKER[:BASiC]:FREQUENCY<x>:STATE  
{<Boolean>}  
:ANALYSIS:FFT<x>:MEASURE:  
MARKER[:BASiC]:FREQUENCY<x>:STATE?  
FFT<x> の <x> = 1, 2  
FREQUENCY<x> の <x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:MARKER:BASiC:  
FREQUENCY1:STATE ON  
:ANALYSIS:FFT1:MEASURE:MARKER:BASiC:  
FREQUENCY1:STATE? -> :ANALYSIS:FFT1:  
MEASURE:MARKER:BASiC:FREQUENCY1:  
STATE 1

#### **:ANALYSIS:FFT<x>:MEASURE:**

##### **MARKER[:BASiC]:FREQUENCY<x>:VALUE?**

機能 FFT 演算の各マーカーカーソルの周波数値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:  
MARKER[:BASiC]:FREQUENCY<x>:VALUE?  
FFT<x> の <x> = 1, 2  
FREQUENCY<x> の <x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:MARKER:BASiC:  
FREQUENCY1:VALUE? -> :ANALYSIS:FFT1:  
MEASURE:MARKER:BASiC:FREQUENCY1:  
VALUE 1.000E+00

#### **:ANALYSIS:FFT<x>:MEASURE:**

##### **MARKER[:BASiC]:POSITION<x> {<NRf>}**

機能 FFT 演算の各マーカーカーソルの位置を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MEASURE:  
MARKER[:BASiC]:POSITION<x> {<NRf>}  
:ANALYSIS:FFT<x>:MEASURE:  
MARKER[:BASiC]:POSITION<x>?  
FFT<x> の <x> = 1, 2  
POSITION<x> の <x> = 1, 2  
<NRf> = -5 ~ 5(div)

例 :ANALYSIS:FFT1:MEASURE:MARKER:BASiC:  
POSITION1 1  
:ANALYSIS:FFT1:MEASURE:MARKER:BASiC:  
POSITION1? -> :ANALYSIS:FFT1:MEASURE:  
MARKER:BASiC:POSITION1 1.000E+00

#### **:ANALYSIS:FFT<x>:MEASURE:**

##### **MARKER[:BASiC]:V<x>?**

機能 FFT 演算の各マーカーカーソルのパワー値に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:  
MARKER[:BASiC]:V<x>?  
FFT<x> の <x> = 1, 2  
V<x> の <x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:MARKER:BASiC:  
V1? -> :ANALYSIS:FFT1:MEASURE:MARKER:  
BASiC:V1:STATE 1

**:ANALYSIS:FFT<x>:MEASURE:****MARKER[:BASIC]:V<x>:STATE**

機能 FFT 演算の各マーカーカーソルのパワー値の ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MEASURE:  
MARKER[:BASIC]:V<x>:STATE {<Boolean>}  
:ANALYSIS:FFT<x>:MEASURE:  
MARKER[:BASIC]:V<x>:STATE?  
FFT<x> の <x> = 1、2  
V<x> の <x> = 1、2

例 :ANALYSIS:FFT1:MEASURE:MARKER: BASIC:  
V1:STATE ON  
:ANALYSIS:FFT1:MEASURE:MARKER: BASIC:  
V1:STATE? -> :ANALYSIS:FFT1:MEASURE:  
MARKER: BASIC:V1:STATE 1

**:ANALYSIS:FFT<x>:MEASURE:****MARKER[:BASIC]:V<x>:VALUE?**

機能 FFT 演算の各マーカーカーソルのパワー値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:  
MARKER[:BASIC]:V<x>:VALUE?  
FFT<x> の <x> = 1、2  
V<x> の <x> = 1、2

例 :ANALYSIS:FFT1:MEASURE:MARKER: BASIC:  
V1:VALUE? -> :ANALYSIS:FFT1:MEASURE:  
MARKER: BASIC:V1:VALUE 1.000E+00

**:ANALYSIS:FFT<x>:MEASURE:MARKER:****CALCULATION?**

機能 FFT 演算のマーカーカーソルの Calc アイテムに関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:MARKER:  
CALCULATION?  
<x> = 1、2

例 :ANALYSIS:FFT1:MEASURE:MARKER:  
CALCULATION? -> :ANALYSIS:FFT1:  
MEASURE:MARKER:CALCULATION:  
DEFINE1 "V(F1)";DEFINE2  
"V(F2)";DEFINE3 "V(F1)";  
DEFINE4 "V(F2)";STATE1 1;STATE2 1;  
STATE3 1;STATE4 1

**:ANALYSIS:FFT<x>:MEASURE:MARKER:****CALCULATION:ALL**

機能 FFT 演算のマーカーカーソルのすべての Calc アイテムを一斉に ON/OFF します。

構文 :ANALYSIS:FFT<x>:MEASURE:MARKER:  
CALCULATION:ALL {<Boolean>}  
<x> = 1、2

例 :ANALYSIS:FFT1:MEASURE:MARKER:  
CALCULATION:ALL ON

**:ANALYSIS:FFT<x>:MEASURE:MARKER:****CALCULATION:DEFINE<x>**

機能 FFT 演算のマーカーカーソルの Calc アイテムの演算式を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MEASURE:MARKER:  
CALCULATION:DEFINE<x> {<文字列>}  
:ANALYSIS:FFT<x>:MEASURE:MARKER:  
CALCULATION:DEFINE<x>?  
FFT<x> の <x> = 1、2  
DEFINE<x> の <x> = 1 ~ 4  
<文字列> = 128 文字以内

例 :ANALYSIS:FFT1:MEASURE:MARKER:  
CALCULATION:DEFINE1 "V(F1)"  
:ANALYSIS:FFT1:MEASURE:MARKER:  
CALCULATION:DEFINE1? -> :ANALYSIS:  
FFT1:MEASURE:MARKER:CALCULATION:  
DEFINE1 "V(F1)"

**:ANALYSIS:FFT<x>:MEASURE:MARKER:****CALCULATION:STATE<x>**

機能 FFT 演算のマーカーカーソルの Calc アイテムの ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MEASURE:MARKER:C  
ALCULATION:STATE<x> {<Boolean>}  
:ANALYSIS:FFT<x>:MEASURE:MARKER:  
CALCULATION:STATE<x>?  
FFT<x> の <x> = 1、2  
STATE<x> の <x> = 1 ~ 4

例 :ANALYSIS:FFT1:MEASURE:MARKER:  
CALCULATION:STATE1 ON  
:ANALYSIS:FFT1:MEASURE:MARKER:  
CALCULATION:STATE1? -> :ANALYSIS:  
FFT1:MEASURE:MARKER:CALCULATION:  
STATE1 1

**:ANALYSIS:FFT<x>:MEASURE:MARKER:****CALCULATION:VALUE<x>?**

機能 FFT 演算のマーカーカーソルの Calc アイテムの測定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:MARKER:  
CALCULATION:VALUE<x>?  
FFT<x> の <x> = 1、2  
VALUE<x> の <x> = 1 ~ 4

例 :ANALYSIS:FFT1:MEASURE:MARKER:  
CALCULATION:VALUE1? -> :ANALYSIS:  
FFT1:MEASURE:MARKER:CALCULATION:  
VALUE1 1.000E+00

### 5.3 ANALYSIS グループ

#### **:ANALYSIS:FFT<x>:MEASURE:MODE**

機能 FFT 演算の自動測定モードを設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MEASURE:  
MODE {MARKER|OFF|PEAK}  
:ANALYSIS:FFT<x>:MEASURE:MODE?  
<x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:MODE MARKER  
:ANALYSIS:FFT1:MEASURE:MODE?  
-> :ANALYSIS:FFT1:MEASURE:MODE MARKER

#### **:ANALYSIS:FFT<x>:MEASURE:PEAK?**

機能 FFT 演算のピーク値測定に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK?  
<x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:PEAK?  
-> :ANALYSIS:FFT1:MEASURE:PEAK: BASIC:  
DFREQUENCY:STATE 1;:ANALYSIS:FFT1:  
MEASURE:PEAK: BASIC:DV:STATE 1;:  
ANALYSIS:FFT1:MEASURE:PEAK: BASIC:  
FREQUENCY1:STATE 1;:ANALYSIS:FFT1:  
MEASURE:PEAK: BASIC:FREQUENCY2:  
STATE 1;:ANALYSIS:FFT1:MEASURE:PEAK:  
BASIC:RANGE1 1.000E+00,0.000E+00;  
RANGE2 1.000E+00,0.000E+00;V1:  
STATE 1;:ANALYSIS:FFT1:MEASURE:PEAK:  
BASIC:V2:STATE 1;:ANALYSIS:FFT1:  
MEASURE:PEAK:CALCULATION:  
DEFINE1 "V(P1)";DEFINE2 "V(P2)";  
DEFINE3 "V(P1)";DEFINE4 "V(P2)";  
STATE1 1;STATE2 1;STATE3 1;STATE4 1

#### **:ANALYSIS:FFT<x>:MEASURE:PEAK: BASIC?**

機能 FFT 演算のピーク値の Basic アイテムに関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK: BASIC?  
<x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:PEAK: BASIC?  
-> :ANALYSIS:FFT1:MEASURE:PEAK: BASIC:  
DFREQUENCY:STATE 1;:ANALYSIS:FFT1:  
MEASURE:PEAK: BASIC:DV:STATE 1;:  
ANALYSIS:FFT1:MEASURE:PEAK: BASIC:  
FREQUENCY1:STATE 1;:ANALYSIS:FFT1:  
MEASURE:PEAK: BASIC:FREQUENCY2:  
STATE 1;:ANALYSIS:FFT1:MEASURE:PEAK:  
BASIC:RANGE1 1.000E+00,0.000E+00;  
RANGE2 1.000E+00,0.000E+00;V1:  
STATE 1;:ANALYSIS:FFT1:MEASURE:PEAK:  
BASIC:V2:STATE 1

#### **:ANALYSIS:FFT<x>:MEASURE:**

##### **PEAK[:BASIC]:ALL**

機能 FFT 演算のピーク値のすべての Basic アイテムを一斉に ON/OFF します。

構文 :ANALYSIS:FFT<x>:MEASURE:  
PEAK[:BASIC]:ALL {<Boolean>}  
<x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:PEAK: BASIC:  
ALL ON

#### **:ANALYSIS:FFT<x>:MEASURE:**

##### **PEAK[:BASIC]:DFREQUENCY?**

機能 FFT 演算のピーク値間の周波数値に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK[:  
BASIC]:DFREQUENCY?  
<x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:PEAK: BASIC:  
DFREQUENCY? -> :ANALYSIS:FFT1:  
MEASURE:PEAK: BASIC:DFREQUENCY:STATE 1

#### **:ANALYSIS:FFT<x>:MEASURE:**

##### **PEAK[:BASIC]:DFREQUENCY: STATE**

機能 FFT 演算のピーク値間の周波数値の ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MEASURE:  
PEAK[:BASIC]:DFREQUENCY:  
STATE {<Boolean>}  
:ANALYSIS:FFT<x>:MEASURE:  
PEAK[:BASIC]:DFREQUENCY: STATE?  
<x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:PEAK: BASIC:  
DFREQUENCY:STATE ON  
:ANALYSIS:FFT1:MEASURE:PEAK: BASIC:  
DFREQUENCY:STATE? -> :ANALYSIS:FFT1:  
MEASURE:PEAK: BASIC:DFREQUENCY:STATE 1

#### **:ANALYSIS:FFT<x>:MEASURE:**

##### **PEAK[:BASIC]:DFREQUENCY: VALUE?**

機能 FFT 演算のピーク値間の周波数値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:  
PEAK[:BASIC]:DFREQUENCY: VALUE?  
<x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:PEAK: BASIC:  
DFREQUENCY:VALUE? -> :ANALYSIS:FFT1:  
MEASURE:PEAK: BASIC:DFREQUENCY:  
VALUE 1.000E+00

**:ANALYSIS:FFT<x>:MEASURE:****PEAK[:BASIC]:DV?**

機能 FFT 演算のピーク値間のパワー値に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK[:BASIC]:DV?  
<x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:DV?  
-> :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:DV:STATE 1

**:ANALYSIS:FFT<x>:MEASURE:****PEAK[:BASIC]:DV:STATE**

機能 FFT 演算のピーク値間のパワー値の ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK[:BASIC]:DV:STATE {<Boolean>}  
:ANALYSIS:FFT<x>:MEASURE:PEAK[:BASIC]:DV:STATE?  
<x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:DV:STATE ON  
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:DV:STATE? -> :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:DV:STATE 1

**:ANALYSIS:FFT<x>:MEASURE:****PEAK[:BASIC]:DV:VALUE?**

機能 FFT 演算のピーク値間のパワー値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK[:BASIC]:DV:VALUE?  
<x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:DV:VALUE? -> :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:DV:VALUE 1.000E+00

**:ANALYSIS:FFT<x>:MEASURE:****PEAK[:BASIC]:FREQUENCY<x>?**

機能 FFT 演算の各ピーク周波数値に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK[:BASIC]:FREQUENCY<x>?  
FFT<x> の <x> = 1, 2  
FREQUENCY<x> の <x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:FREQUENCY1? -> :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:FREQUENCY1:STATE 1

**:ANALYSIS:FFT<x>:MEASURE:****PEAK[:BASIC]:FREQUENCY<x>:STATE**

機能 FFT 演算の各ピーク周波数値の ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK[:BASIC]:FREQUENCY<x>:STATE {<Boolean>}  
:ANALYSIS:FFT<x>:MEASURE:PEAK[:BASIC]:FREQUENCY<x>:STATE?  
FFT<x> の <x> = 1, 2  
FREQUENCY<x> の <x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:FREQUENCY1:STATE ON  
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:FREQUENCY1:STATE? -> :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:FREQUENCY1:STATE 1

**:ANALYSIS:FFT<x>:MEASURE:****PEAK[:BASIC]:FREQUENCY<x>:VALUE?**

機能 FFT 演算の各ピーク周波数値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK[:BASIC]:FREQUENCY<x>:VALUE?  
FFT<x> の <x> = 1, 2  
FREQUENCY<x> の <x> = 1, 2

例 :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:FREQUENCY1:VALUE? -> :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:FREQUENCY1:VALUE 1.000E+00

**:ANALYSIS:FFT<x>:MEASURE:****PEAK[:BASIC]:RANGE<x>**

機能 FFT 演算のピーク値の各測定範囲を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK[:BASIC]:RANGE<x> {<NRf>, <NRf>}  
:ANALYSIS:FFT<x>:MEASURE:PEAK[:BASIC]:RANGE<x>?  
FFT<x> の <x> = 1, 2  
RANGE<x> の <x> = 1, 2  
<NRf> = -5 ~ 5(div)

例 :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:RANGE1 0,1  
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:RANGE1? -> :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:RANGE1 1.000E+00,0.000E+00

### 5.3 ANALYSIS グループ

#### **:ANALYSIS:FFT<x>:MEASURE:**

##### **PEAK[:BASIC]:V<x>?**

機能 FFT 演算の各ピーク値に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK[:  
BASIC]:V<x>?

FFT<x> の <x> = 1、2

V<x> の <x> = 1、2

例 :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:V1?  
-> :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:  
V1:STATE 1

#### **:ANALYSIS:FFT<x>:MEASURE:**

##### **PEAK[:BASIC]:V<x>:STATE**

機能 FFT 演算の各ピーク値の ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK[:  
BASIC]:V<x>:STATE {<Boolean>}

:ANALYSIS:FFT<x>:MEASURE:

PEAK[:BASIC]:V<x>:STATE?

FFT<x> の <x> = 1、2

V<x> の <x> = 1、2

例 :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:V1:  
STATE ON  
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:V1:  
STATE? -> :ANALYSIS:FFT1:MEASURE:  
PEAK:BASIC:V1:STATE 1

#### **:ANALYSIS:FFT<x>:MEASURE:**

##### **PEAK[:BASIC]:V<x>:VALUE?**

機能 FFT 演算の各ピーク値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK[:  
BASIC]:

V<x>:VALUE?

FFT<x> の <x> = 1、2

V<x> の <x> = 1、2

例 :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:V1:  
VALUE? -> :ANALYSIS:FFT1:MEASURE:  
PEAK:BASIC:V1:VALUE 1.000E+00

#### **:ANALYSIS:FFT<x>:MEASURE:PEAK:**

##### **CALCULATION?**

機能 FFT 演算の Calc アイテムに関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK:  
CALCULATION?

<x> = 1、2

例 :ANALYSIS:FFT1:MEASURE:PEAK:  
CALCULATION? -> :ANALYSIS:FFT1:  
MEASURE:  
PEAK:CALCULATION:DEFINE1 "V(P1)";  
DEFINE2 "V(P2)";DEFINE3 "V(P1)";  
DEFINE4 "V(P2)";STATE1 1;STATE2 1;  
STATE3 1;STATE4 1

#### **:ANALYSIS:FFT<x>:MEASURE:PEAK:**

##### **CALCULATION:ALL**

機能 FFT 演算のすべての Calc アイテムを一斉に ON/OFF します。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK:  
CALCULATION:ALL {<Boolean>}

<x> = 1、2

例 :ANALYSIS:FFT1:MEASURE:PEAK:  
CALCULATION:ALL ON

#### **:ANALYSIS:FFT<x>:MEASURE:PEAK:**

##### **CALCULATION:DEFINE<x>**

機能 FFT 演算の Calc アイテムの演算式を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK:  
CALCULATION:DEFINE<x> {<文字列>}

:ANALYSIS:FFT<x>:MEASURE:PEAK:

CALCULATION:DEFINE<x>?

FFT<x> の <x> = 1、2

DEFINE<x> の <x> = 1 ~ 4

<文字列> = 128 文字以内

例 :ANALYSIS:FFT1:MEASURE:PEAK:  
CALCULATION:DEFINE1 "V(P1)"  
:ANALYSIS:FFT1:MEASURE:PEAK:  
CALCULATION:DEFINE1? -> :ANALYSIS:  
FFT1:MEASURE:PEAK:CALCULATION:  
DEFINE1 "V(P1)"

#### **:ANALYSIS:FFT<x>:MEASURE:PEAK:**

##### **CALCULATION:STATE<x>**

機能 FFT 演算の Calc アイテムの ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK:  
CALCULATION:STATE<x> {<Boolean>}

:ANALYSIS:FFT<x>:MEASURE:PEAK:

CALCULATION:STATE<x>?

FFT<x> の <x> = 1、2

STATE<x> の <x> = 1 ~ 4

例 :ANALYSIS:FFT1:MEASURE:PEAK:  
CALCULATION:STATE1 ON  
:ANALYSIS:FFT1:MEASURE:PEAK:  
CALCULATION:STATE1? -> :ANALYSIS:  
FFT1:MEASURE:PEAK:CALCULATION:  
STATE1 1

**:ANALYSIS:FFT<x>:MEASURE:PEAK:****CALCULATION:VALUE<x>?**

機能 FFT 演算の Calc アイテムの測定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:MEASURE:PEAK:

CALCULATION:VALUE<x>?

FFT<x> の <x> = 1、2

VALUE<x> の <x> = 1 ~ 4

例 :ANALYSIS:FFT1:MEASURE:PEAK:  
CALCULATION:VALUE1? -> :ANALYSIS:  
FFT1:MEASURE:PEAK:CALCULATION:  
VALUE1 1.000E+00

**:ANALYSIS:FFT<x>:RANGE**

機能 FFT 演算の測定対象ウィンドウを設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:RANGE {MAIN|Z1|Z2}

:ANALYSIS:FFT<x>:RANGE?

<x> = 1、2

例 :ANALYSIS:FFT1:RANGE MAIN  
:ANALYSIS:FFT1:RANGE? -> :ANALYSIS:  
FFT1:RANGE MAIN

**:ANALYSIS:FFT<x>:RPART (Real Part)**

機能 FFT 演算のリアルパート対象トレースを設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:RPART {<NRF>}

:ANALYSIS:FFT<x>:RPART?

<x> = 1、2

<NRF> = 1 ~ 8

例 :ANALYSIS:FFT1:RPART 1  
:ANALYSIS:FFT1:RPART? -> :ANALYSIS:  
FFT1:RPART 1

**:ANALYSIS:FFT<x>:RPOSITION  
(Ref Position)**

機能 FFT 演算の垂直軸の拡大中心点を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:RPOSITION {<NRF>}

:ANALYSIS:FFT<x>:RPOSITION?

<x> = 1、2

<NRF> = -4 ~ 4(div)

例 :ANALYSIS:FFT1:RPOSITION 1  
:ANALYSIS:FFT1:RPOSITION?  
-> :ANALYSIS:FFT1:RPOSITION 1.000E+00

**:ANALYSIS:FFT<x>:VERTICAL?**

機能 FFT 演算の垂直軸に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:FFT<x>:VERTICAL?

<x> = 1、2

例 :ANALYSIS:FFT1:VERTICAL?  
-> :ANALYSIS:FFT1:VERTICAL:  
LEVEL 1.000E+00;MODE AUTO;  
SENSITIVITY 1.000E+00

**:ANALYSIS:FFT<x>:VERTICAL:LEVEL**

機能 FFT 演算の垂直軸の表示位置を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:VERTICAL:LEVEL

{<NRF>}

:ANALYSIS:FFT<x>:VERTICAL:LEVEL?

<x> = 1、2

<NRF> = -1.000E+31 ~ 1.000E+31(dBV)

例 :ANALYSIS:FFT1:VERTICAL:LEVEL 1  
:ANALYSIS:FFT1:VERTICAL:LEVEL?  
-> :ANALYSIS:FFT1:VERTICAL:  
LEVEL 1.000E+00

**:ANALYSIS:FFT<x>:VERTICAL:MODE**

機能 FFT 演算の垂直軸のモードを設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:VERTICAL:MODE {AUTO|

MANUAL}

:ANALYSIS:FFT<x>:VERTICAL:MODE?

<x> = 1、2

例 :ANALYSIS:FFT1:VERTICAL:MODE AUTO  
:ANALYSIS:FFT1:VERTICAL:MODE?  
-> :ANALYSIS:FFT1:VERTICAL:MODE AUTO

**:ANALYSIS:FFT<x>:VERTICAL:****SENSITIVITY**

機能 FFT 演算の垂直軸感度を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:VERTICAL:SENSITIVITY

{<NRF>}

:ANALYSIS:FFT<x>:VERTICAL:

SENSITIVITY?

<x> = 1、2

<NRF> = 0 ~ 1.000E+31(dBV)

例 :ANALYSIS:FFT1:VERTICAL:SENSITIVITY 1  
:ANALYSIS:FFT1:VERTICAL:SENSITIVITY?  
-> :ANALYSIS:FFT1:VERTICAL:  
SENSITIVITY 1.000E+00

**:ANALYSIS:FFT<x>:WINDOW**

機能 FFT 演算の窓関数を設定 / 問い合わせします。

構文 :ANALYSIS:FFT<x>:WINDOW {FLATTOP|

HANNING|RECTANGLE}

:ANALYSIS:FFT<x>:WINDOW?

<x> = 1、2

例 :ANALYSIS:FFT1:WINDOW FLATTOP  
:ANALYSIS:FFT1:WINDOW? -> :ANALYSIS:  
FFT1:WINDOW FLATTOP

**:ANALYSIS:TYPE<x>**

機能 解析機能のタイプを設定 / 問い合わせします。

構文 :ANALYSIS:TYPE<x> {AHISTOGRAM|FFT|

WPARAMETER|XY}

:ANALYSIS:TYPE<x>?

<x> = 1、2

例 :ANALYSIS:TYPE1 AHISTOGRAM  
:ANALYSIS:TYPE1? -> :ANALYSIS:TYPE1  
AHISTOGRAM

### 5.3 ANALysis グループ

#### :ANALysis:VTDisplay

機能 VT 波形の表示の ON/OFF を設定 / 問い合わせします。

構文 :ANALysis:VTDisplay {<Boolean>}  
:ANALysis:VTDisplay?

例 :ANALYSIS:VTDISPLAY ON  
:ANALYSIS:VTDISPLAY? -> :ANALYSIS:  
VTDISPLAY 1

#### :ANALysis:WAIT<x>?

機能 タイムアウト付きで自動測定実行の終了を待ちます。

構文 ANALysis:WAIT<x>? {<NRf>}  
<x> = 1、2  
<NRf> = 1 ~ 360000(タイムアウト時間、10ms)

例 ANALYSIS:WAIT1? 100 ->  
:ANALYSIS:WAIT1 1

解説 タイムアウト時間内に自動測定の実行が終了すると「0」、終了してないか自動測定が行われていない場合は「1」を返します。  
タイムアウト時間を長く設定しても、自動測定実行が終了した時点で「0」を返します。

#### :ANALysis:WPARAMeter<x>?

機能 波形パラメータ測定機能に関するすべての設定値を問い合わせます。

構文 :ANALysis:WPARAMeter<x>?  
<x> = 1、2

例 :ANALYSIS:WPARAMETER1? -> :ANALYSIS:  
WPARAMETER1:CALCULATION 1;  
HISTOGRAM:MEASURE:MODE CURSOR;CURSOR:  
C1:POSITION 1.000E+00;STATE 1;;  
ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:CURSOR:C2:POSITION 2.000E+00;  
STATE 1;;ANALYSIS:WPARAMETER1:  
HISTOGRAM:MEASURE:CURSOR:DC:STATE 1;;  
ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:CURSOR:LINKAGE 1;;ANALYSIS:  
WPARAMETER1:HISTOGRAM:MEASURE:  
PARAMETER:MEAN:STATE 1;;ANALYSIS:  
WPARAMETER1:HISTOGRAM:MEASURE:  
PARAMETER:PEAK:STATE 1;;ANALYSIS:  
WPARAMETER1:HISTOGRAM:MEASURE:  
PARAMETER:SD3INTEG:STATE 1;;ANALYSIS:  
WPARAMETER1:HISTOGRAM:MEASURE:  
PARAMETER:SDEVIATION:STATE 1;;  
ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:PARAMETER:SDINTEG:STATE 1;;  
ANALYSIS:WPARAMETER1:LIST:  
MODE STATISTIC;SCROLL HORIZONTAL;;  
ANALYSIS:WPARAMETER1:  
MODE HISTOGRAM;TREND:HRANGE MAIN;  
HSPAN 1;MEASURE:CURSOR:C1:  
POSITION 1.000E+00;STATE 1;;ANALYSIS:  
WPARAMETER1:TREND:MEASURE:CURSOR:C2:  
POSITION 2.000E+00;STATE 1;;ANALYSIS:  
WPARAMETER1:TREND:MEASURE:CURSOR:DC:  
STATE 1;;ANALYSIS:WPARAMETER1:TREND:  
MEASURE:MODE CURSOR;;ANALYSIS:  
WPARAMETER1:TREND:  
VERTICAL 2.000E+00,1.000E+00

#### :ANALysis:WPARAMeter<x>:BIT<x>?

機能 ロジック波形パラメータ測定の各ロジックビットに関するすべての設定値を問い合わせます。

構文 :ANALysis:WPARAMeter<x>:BIT<x>?  
WPARAMeter<x>の<x> = 1、2

BIT<x>の<x> = 1 ~ 32(DL9510L、DL9505Lは  
BIT<x>の<x> = 1 ~ 8、17 ~ 24が有効です。)

例 :ANALYSIS:WPARAMETER1:BIT1? -> :  
ANALYSIS:WPARAMETER1:BIT1:AREA1:  
TYPE COUNT

**:ANALYSIS:WPARAMETER<x>:BIT<x>:AREA<x>?**

機能 ロジック波形パラメータ測定の各エリアに関するすべての設定値を問い合わせます。

構文 :ANALYSIS:WPARAMETER<x>:BIT<x>:AREA<x>?

WPARAMETER<x> の <x> = 1, 2  
BIT<x> の <x> = 1 ~ 32(DL9510L、DL9505L は BIT<x> の <x> = 1 ~ 8、17 ~ 24 が有効です。)  
AREA<x> の <x> = 1, 2

例 :ANALYSIS:WPARAMETER1:BIT1:AREA1?  
-> :ANALYSIS:WPARAMETER1:BIT1:AREA1:TYPE COUNT

**:ANALYSIS:WPARAMETER<x>:BIT<x>:AREA<x>:TYPE**

機能 ロジック波形パラメータ測定のロジック波形パラメータを設定 / 問い合わせします。

構文 :ANALYSIS:WPARAMETER<x>:BIT<x>:AREA<x>:TYPE {COUNT|DElay|DT|DUTYcycle|FREquency|NWIDTH|PERFrequency|PERiod|PWIDTH}

:ANALYSIS:WPARAMETER<x>:BIT<x>:AREA<x>:TYPE?

WPARAMETER<x> の <x> = 1, 2  
BIT<x> の <x> = 1 ~ 32(DL9510L、DL9505L は BIT<x> の <x> = 1 ~ 8、17 ~ 24 が有効です。)  
AREA<x> の <x> = 1, 2

例 :ANALYSIS:WPARAMETER1:BIT1:AREA1:TYPE COUNT  
:ANALYSIS:WPARAMETER1:BIT1:AREA1:TYPE? -> :ANALYSIS:WPARAMETER1:BIT1:AREA1:TYPE COUNT

**:ANALYSIS:WPARAMETER<x>:CALCulation**

機能 波形パラメータ測定の Calc アイテムを設定 / 問い合わせします。

構文 :ANALYSIS:WPARAMETER<x>:CALCulation {<NRF>}

:ANALYSIS:WPARAMETER<x>:CALCulation? <x> = 1, 2  
<NRF> = 1 ~ 4

例 :ANALYSIS:WPARAMETER1:CALCULATION 1  
:ANALYSIS:WPARAMETER1:CALCULATION?  
-> :ANALYSIS:WPARAMETER1:CALCULATION 1

**:ANALYSIS:WPARAMETER<x>:HISTogram?**

機能 波形パラメータ測定のヒストグラム表示に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:WPARAMETER<x>:HISTogram? <x> = 1, 2

例 :ANALYSIS:WPARAMETER1:HISTOGRAM?  
-> :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:MODE CURSOR;CURSOR:C1:POSITION 1.000E+00;STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C2:POSITION 2.000E+00;STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:DC:STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:LINKAGE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:PARAMETER:MEAN:STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:PARAMETER:PEAK:STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:PARAMETER:SD3INTEG:STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:PARAMETER:SDEVIATION:STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:PARAMETER:SDINTEG:STATE 1

**:ANALYSIS:WPARAMETER<x>:HISTogram:MEASURE?**

機能 波形パラメータ測定のヒストグラム表示の自動測定に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:WPARAMETER<x>:HISTogram:MEASURE? <x> = 1, 2

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE? -> :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:MODE CURSOR;CURSOR:C1:POSITION 1.000E+00;STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C2:POSITION 2.000E+00;STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:DC:STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:LINKAGE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:PARAMETER:MEAN:STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:PARAMETER:PEAK:STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:PARAMETER:SD3INTEG:STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:PARAMETER:SDEVIATION:STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:PARAMETER:SDINTEG:STATE 1



### 5.3 ANALYSIS グループ

#### **:ANALYSIS:WPARAMETER<x>:HISTOGRAM:MEASURE:CURSOR?**

機能 波形パラメータ測定の実ログラムのカーソル測定に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:WPARAMETER<x>:HISTOGRAM:MEASURE:CURSOR?  
<x> = 1、2

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR? -> :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C1:POSITION 1.000E+00;STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C2:POSITION 2.000E+00;STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:DC:STATE 1;:ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:LINKAGE 1

#### **:ANALYSIS:WPARAMETER<x>:HISTOGRAM:MEASURE:CURSOR:ALL**

機能 波形パラメータ測定の実ログラムのカーソルを一斉に ON/OFF します。

構文 :ANALYSIS:WPARAMETER<x>:HISTOGRAM:MEASURE:CURSOR:ALL {<Boolean>}  
<x> = 1、2

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:ALL ON

#### **:ANALYSIS:WPARAMETER<x>:HISTOGRAM:MEASURE:CURSOR:C<x>?**

機能 波形パラメータ測定の実ログラムの各カーソル測定に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:WPARAMETER<x>:HISTOGRAM:MEASURE:CURSOR:C<x>?  
WPARAMETER<x> の <x> = 1、2  
C<x> の <x> = 1、2

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C1? -> :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C1:POSITION 1.000E+00;STATE 1

#### **:ANALYSIS:WPARAMETER<x>:HISTOGRAM:MEASURE:CURSOR:C<x>:POSITION**

機能 波形パラメータ測定の実ログラムの各カーソルの位置を設定 / 問い合わせします。

構文 :ANALYSIS:WPARAMETER<x>:HISTOGRAM:MEASURE:CURSOR:C<x>:POSITION {<Nrf>}

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C<x>:POSITION? WPARAMETER<x> の <x> = 1、2  
C<x> の <x> = 1、2  
<Nrf> = -5 ~ 5(div)

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C1:POSITION ON :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C1:POSITION? -> :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C1:POSITION 1.000E+00

#### **:ANALYSIS:WPARAMETER<x>:HISTOGRAM:MEASURE:CURSOR:C<x>:STATE**

機能 波形パラメータ測定の実ログラムの各カーソルの ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:WPARAMETER<x>:HISTOGRAM:MEASURE:CURSOR:C<x>:STATE {<Boolean>}

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C<x>:STATE? WPARAMETER<x> の <x> = 1、2  
C<x> の <x> = 1、2

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C1:STATE ON :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C1:STATE? -> :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C1:STATE 1

#### **:ANALYSIS:WPARAMETER<x>:HISTOGRAM:MEASURE:CURSOR:C<x>:VALUE?**

機能 波形パラメータ測定の実ログラムの各カーソルの測定値を問い合わせます。

構文 :ANALYSIS:WPARAMETER<x>:HISTOGRAM:MEASURE:CURSOR:C<x>:VALUE?  
WPARAMETER<x> の <x> = 1、2  
C<x> の <x> = 1、2

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C1:VALUE? -> :ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:CURSOR:C1:VALUE 1.000E+00

**:ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:Cursors:DC?**

機能 波形パラメータ測定の手istogramのカーソル間の測定に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:Cursors:DC?  
<x> = 1, 2

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:Cursors:DC? -> :ANALYSIS:  
WPARAMETER1:HISTOGRAM:MEASURE:Cursors:  
DC:STATE 1

**:ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:Cursors:DC:STATE**

機能 波形パラメータ測定の手istogramのカーソル間のON/OFFを設定/問い合わせします。

構文 :ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:PARAMETER:DC:STATE  
{<Boolean>}  
:ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:PARAMETER:DC:STATE?  
<x> = 1, 2

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:Cursors:DC:STATE ON  
:ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:Cursors:DC:STATE?  
-> :ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:Cursors:DC:STATE 1

**:ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:Cursors:DC:VALUE?**

機能 波形パラメータ測定の手istogramのカーソル間の測定値を問い合わせます。

構文 :ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:Cursors:DC:VALUE?  
<x> = 1, 2

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:Cursors:DC:VALUE?  
-> :ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:Cursors:DC:VALUE 1.000E+00

**:ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:Cursors:LINKAGE**

機能 波形パラメータ測定の手istogramのカーソルのリンクのON/OFFを設定/問い合わせします。

構文 :ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:Cursors:LINKAGE {<Boolean>}  
:ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:Cursors:LINKAGE?  
<x> = 1, 2

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:Cursors:LINKAGE ON  
:ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:Cursors:LINKAGE? -> :ANALYSIS:  
WPARAMETER1:HISTOGRAM:MEASURE:Cursors:  
LINKAGE 1

**:ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:MODE**

機能 波形パラメータ測定の手istogram表示の自動測定モードを設定/問い合わせします。

構文 :ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:MODE {Cursors|OFF|PARAMETER}  
:ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:MODE?  
<x> = 1, 2

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:MODE CURSOR  
:ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:MODE? -> :ANALYSIS:  
WPARAMETER1:HISTOGRAM:  
MEASURE:MODE CURSOR

**:ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:PARAMETER?**

機能 波形パラメータ測定の手istogramパラメータの自動測定に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:PARAMETER?  
<x> = 1, 2

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:PARAMETER? -> :ANALYSIS:  
WPARAMETER1:HISTOGRAM:  
MEASURE:PARAMETER:MEAN:STATE 1;:  
ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:PARAMETER:PEAK:STATE 1;:  
ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:PARAMETER:SD3INTEG:STATE 1;:  
ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:PARAMETER:SDEVIATION:  
STATE 1;:ANALYSIS:WPARAMETER1:  
HISTOGRAM:MEASURE:PARAMETER:SDINTEG:  
STATE 1

**:ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:PARAMETER:ALL**

機能 波形パラメータ測定の手istogramすべてのパラメータを一斉にON/OFFします。

構文 :ANALYSIS:WPARAMETER<x>:HISTOGRAM:  
MEASURE:PARAMETER:ALL {<Boolean>}  
<x> = 1, 2

例 :ANALYSIS:WPARAMETER1:HISTOGRAM:  
MEASURE:PARAMETER:ALL ON

## 5.3 ANALysis グループ

### **:ANALysis:WPARAMeter<x>:HISTogram:**

#### **MEASure:PARAmeter:<パラメータ>?**

機能 波形パラメータ測定のカスタムパラメータに関するすべての設定値を問い合わせます。

構文 :ANALysis:WPARAMeter<x>:HISTogram:

MEASure:PARAmeter:<パラメータ>?

<x> = 1, 2

<パラメータ> = {MEAN|PEAK|SD3integ|SDEVIation|SDINteG}

例 (以下は、平均値についての例です。)

:ANALYSIS:WPARAMETER1:HISTOGRAM:

MEASURE:PARAMETER:MEAN? -> :ANALYSIS:

WPARAMETER1:HISTOGRAM:MEASURE:

PARAMETER:MEAN:STATE 1

### **:ANALysis:WPARAMeter<x>:HISTogram:**

#### **MEASure:PARAmeter:<パラメータ>:STATe**

機能 波形パラメータ測定のカスタムパラメータのON/OFFを設定/問い合わせします。

構文 :ANALysis:WPARAMeter<x>:HISTogram:

MEASure:PARAmeter:<パラメータ>:STATe

{<Boolean>}

:ANALysis:WPARAMeter<x>:HISTogram:

MEASure:PARAmeter:<パラメータ>:STATe?

<x> = 1, 2

<パラメータ> = {MEAN|PEAK|SD3integ|SDEVIation|SDINteG}

例 (以下は、平均値についての例です。)

:ANALYSIS:WPARAMETER1:HISTOGRAM:

MEASURE:PARAMETER:MEAN:STATE ON

:ANALYSIS:WPARAMETER1:HISTOGRAM:

MEASURE:PARAMETER:MEAN:STATE?

-> :ANALYSIS:WPARAMETER1:HISTOGRAM:

MEASURE:PARAMETER:MEAN:STATE 1

### **:ANALysis:WPARAMeter<x>:HISTogram:**

#### **MEASure:PARAmeter:<パラメータ>:VALue?**

機能 波形パラメータ測定のカスタムパラメータの測定値を問い合わせます。

構文 :ANALysis:WPARAMeter<x>:HISTogram:

MEASure:PARAmeter:<パラメータ>:VALue?

<x> = 1, 2

<パラメータ> = {MEAN|PEAK|SD3integ|SDEVIation|SDINteG}

例 (以下は、平均値についての例です。)

:ANALYSIS:WPARAMETER1:HISTOGRAM:

MEASURE:PARAMETER:MEAN:VALUE?

-> :ANALYSIS:WPARAMETER1:HISTOGRAM:

MEASURE:PARAMETER:MEAN:VALUE

1.000E+00

### **:ANALysis:WPARAMeter<x>:LIST?**

機能 波形パラメータ測定のカスタム表示に関するすべての設定値を問い合わせます。

構文 :ANALysis:WPARAMeter<x>:LIST?

<x> = 1, 2

例 :ANALYSIS:WPARAMETER1:LIST?

-> :ANALYSIS:WPARAMETER1:LIST:

MODE STATISTIC;SCROLL HORIZONTAL

### **:ANALysis:WPARAMeter<x>:LIST:ITEM?**

機能 波形パラメータ測定のカスタム表示のアイテムを問い合わせます。

構文 :ANALysis:WPARAMeter<x>:LIST:ITEM?

<x> = 1, 2

例 :ANALYSIS:WPARAMETER1:LIST:ITEM?

-> :ANALYSIS:WPARAMETER1:LIST:

ITEM "LOW(C1)"

### **:ANALysis:WPARAMeter<x>:LIST:MODE**

機能 波形パラメータ測定のカスタム表示のモードを設定/問い合わせします。

構文 :ANALysis:WPARAMeter<x>:LIST:

MODE {STATistic|TREND}

:ANALysis:WPARAMeter<x>:LIST:MODE?

<x> = 1, 2

例 :ANALYSIS:WPARAMETER1:LIST:

MODE STATISTIC

:ANALYSIS:WPARAMETER1:LIST:MODE?

-> :ANALYSIS:WPARAMETER1:LIST:

MODE STATISTIC

### **:ANALysis:WPARAMeter<x>:LIST:SCROLL**

機能 波形パラメータ測定のカスタム表示のスクロール方向を設定/問い合わせします。

構文 :ANALysis:WPARAMeter<x>:LIST:SCROLL

{HORizontal|VERTical}

:ANALysis:WPARAMeter<x>:LIST:SCROLL?

<x> = 1, 2

例 :ANALYSIS:WPARAMETER1:LIST:

SCROLL HORIZONTAL

:ANALYSIS:WPARAMETER1:LIST:SCROLL?

-> :ANALYSIS:WPARAMETER1:LIST:

SCROLL HORIZONTAL

**:ANALysis:WPARAMeter<x>:LIST:VALue?**

機能 波形パラメータ測定のリスト表示番号の自動測定値を問い合わせます。

構文 :ANALysis:WPARAMeter<x>:LIST:VALue?  
{<NRF>|MAXimum}  
<x> = 1, 2  
<NRF> = 1 ~ 100000

例 :ANALYSIS:WPARAMETER1:LIST:VALUE? 1  
-> :ANALYSIS:WPARAMETER1:LIST:  
VALUE 1,1.000E+00

解説

- 測定不可能な場合は、「NAN(Not A Number)」が返されます。
- リスト表示番号の値が存在しない場合には「NAN(Not A Number)」が返されます。
- MAXimumを指定すると、最大リスト表示番号になります。

**:ANALysis:WPARAMeter<x>:MODE**

機能 波形パラメータ測定モードを設定 / 問い合わせします。

構文 :ANALysis:WPARAMeter<x>:MODE  
{HISTogram|LIST|TRENd}  
:ANALysis:WPARAMeter<x>:MODE?  
<x> = 1, 2

例 :ANALYSIS:WPARAMETER1:MODE HISTOGRAM  
:ANALYSIS:WPARAMETER1:MODE?  
-> :ANALYSIS:WPARAMETER1:  
MODE HISTOGRAM

**:ANALysis:WPARAMeter<x>:TRACe<x>?**

機能 波形パラメータ測定各トレースに関するすべての設定値を問い合わせます。

構文 :ANALysis:WPARAMeter<x>:TRACe<x>?  
WPARAMeter<x>の<x> = 1, 2  
TRACe<x>の<x> = 1 ~ 8

例 :ANALYSIS:WPARAMETER1:TRACE1?  
-> :ANALYSIS:WPARAMETER1:TRACE1:  
AREA1:TYPE BURST

**:ANALysis:WPARAMeter<x>:TRACe<x>:  
AREA<x>?**

機能 波形パラメータ測定各エリアに関するすべての設定値を問い合わせます。

構文 :ANALysis:WPARAMeter<x>:TRACe<x>:  
AREA<x>?  
WPARAMeter<x>の<x> = 1, 2  
TRACe<x>の<x> = 1 ~ 8  
AREA<x>の<x> = 1, 2

例 :ANALYSIS:WPARAMETER1:TRACE1:AREA1?  
-> :ANALYSIS:WPARAMETER1:TRACE1:  
AREA1:TYPE BURST

**:ANALysis:WPARAMeter<x>:TRACe<x>:  
AREA<x>:TYPE**

機能 波形パラメータ測定波形パラメータを設定 / 問い合わせします。

構文 :ANALysis:WPARAMeter<x>:TRACe<x>:  
AREA<x>:TYPE {<パラメータ>}  
:ANALysis:WPARAMeter<x>:TRACe<x>:  
AREA<x>:TYPE?

WPARAMeter<x>の<x> = 1, 2  
TRACe<x>の<x> = 1 ~ 8

AREA<x>の<x> = 1, 2

<パラメータ> = {BURSt|CMEan|COUNT|CRMS|  
CSDeviation|DElay|DT|DUTYcycle|FALL|  
FREQuency|HIGH|HILow|LOW|MAXimum|MEAN|  
MINimum|NOvershoot|NWIDTH|PERFrequency|  
PERiod|POvershoot|PTOPeak|PWIDth|RISE|  
RMS|SDEviation|TYCInteg|TYINteg|V1|V2}

例 :ANALYSIS:WPARAMETER1:TRACE1:AREA1:  
TYPE BURST  
:ANALYSIS:WPARAMETER1:TRACE1:AREA1:  
TYPE? -> :ANALYSIS:WPARAMETER1:  
TRACE1:AREA1:TYPE BURST

**:ANALysis:WPARAMeter<x>:TRENd?**

機能 波形パラメータ測定トレンド表示に関するすべての設定値を問い合わせます。

構文 :ANALysis:WPARAMeter<x>:TRENd?  
<x> = 1, 2

例 :ANALYSIS:WPARAMETER1:TREND?  
-> :ANALYSIS:WPARAMETER1:TREND:  
HRANGE MAIN;HSPAN 1;MEASURE:CURSOR:  
C1:POSITION 1.000E+00;STATE 1;;  
ANALYSIS:WPARAMETER1:TREND:MEASURE:  
CURSOR:C2:POSITION 2.000E+00;  
STATE 1;;ANALYSIS:WPARAMETER1:TREND:  
MEASURE:CURSOR:DC:STATE 1;;ANALYSIS:  
WPARAMETER1:TREND:MEASURE:  
MODE CURSOR;;ANALYSIS:WPARAMETER1:  
TREND:VERTICAL 2.000E+00,1.000E+00

**:ANALysis:WPARAMeter<x>:TRENd:  
AScale[:EXECute]**

機能 波形パラメータ測定トレンド表示のオートスケールを実行します。

構文 :ANALysis:WPARAMeter<x>:TRENd:  
AScale[:EXECute]  
<x> = 1, 2

例 :ANALYSIS:WPARAMETER1:TREND:ASCALE:  
EXECUTE

### 5.3 ANALysis グループ

**:ANALysis:WPARAMeter<x>:TRENd:HRANge**  
 機能 波形パラメータ測定の特ンド表示の対象ウィンドウを設定 / 問い合わせします。  
 構文 :ANALysis:WPARAMeter<x>:TRENd:HRANge{MAIN|Z1|Z2}  
 :ANALysis:WPARAMeter<x>:TRENd:HRANge?<x> = 1、2  
 例 :ANALYSIS:WPARAMETER1:TREND:HRANGE  
 MAIN  
 :ANALYSIS:WPARAMETER1:TREND:HRANGE?  
 -> :ANALYSIS:WPARAMETER1:TREND:HRANGE MAIN

**:ANALysis:WPARAMeter<x>:TRENd:HSPan**  
 機能 波形パラメータ測定の特ンド表示の水平スパンを設定 / 問い合わせします。  
 構文 :ANALysis:WPARAMeter<x>:TRENd:HSPan{<NRF>}  
 :ANALysis:WPARAMeter<x>:TRENd:HSPan?<x> = 1、2  
 <NRF> = 1 ~ 100000  
 例 :ANALYSIS:WPARAMETER1:TREND:HSPAN 1  
 :ANALYSIS:WPARAMETER1:TREND:HSPAN?  
 -> :ANALYSIS:WPARAMETER1:TREND:HSPAN 1

**:ANALysis:WPARAMeter<x>:TRENd:MEASure?**  
 機能 波形パラメータ測定の特ンド表示の自動測定に関するすべての設定値を問い合わせます。  
 構文 :ANALysis:WPARAMeter<x>:TRENd:MEASure?  
 <x> = 1、2  
 例 :ANALYSIS:WPARAMETER1:TREND:MEASURE?  
 -> :ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:C1:POSITION 1.000E+00;STATE 1;:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:C2:POSITION 2.000E+00;STATE 1;:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:DC:STATE 1;:ANALYSIS:WPARAMETER1:TREND:MEASURE:MODE CURSOR

**:ANALysis:WPARAMeter<x>:TRENd:MEASure:CURSor?**  
 機能 波形パラメータ測定の特ンド表示のカーソル測定に関するすべての設定値を問い合わせます。  
 構文 :ANALysis:WPARAMeter<x>:TRENd:MEASure:CURSor?  
 <x> = 1、2  
 例 :ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR? -> :ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:C1:POSITION 1.000E+00;STATE 1;:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:C2:POSITION 2.000E+00;STATE 1;:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:DC:STATE 1

**:ANALysis:WPARAMeter<x>:TRENd:MEASure:CURSor:ALL**  
 機能 波形パラメータ測定の特ンドのすべてのカーソルを ON/OFF します。  
 構文 :ANALysis:WPARAMeter<x>:TRENd:MEASure:CURSor:ALL {<Boolean>}<x> = 1、2  
 例 :ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:ALL ON

**:ANALysis:WPARAMeter<x>:TRENd:MEASure:CURSor:C<x>?**  
 機能 波形パラメータ測定の特ンドの各カーソル測定に関するすべての設定値を問い合わせます。  
 構文 :ANALysis:WPARAMeter<x>:TRENd:MEASure:CURSor:C<x>?<x> = 1、2  
 C<x> の <x> = 1、2  
 例 :ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:C1? -> :ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:C1:POSITION 1.000E+00;STATE 1

**:ANALysis:WPARAMeter<x>:TRENd:MEASure:CURSor:C<x>:POSition**  
 機能 波形パラメータ測定の特ンドの各カーソルの位置を設定 / 問い合わせします。  
 構文 :ANALysis:WPARAMeter<x>:TRENd:MEASure:CURSor:C<x>:POSition {<NRF>}<x> = 1、2  
 C<x> の <x> = 1、2  
 <NRF> = - 5 ~ 5(div)  
 例 :ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:C1:POSITION 1  
 :ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:C1:POSITION? -> :ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:C1:POSITION 1.000E+00

**:ANALYSIS:WPARAMETER<x>:TRENd:**  
**MEASure:CURSor:C<x>:STATe**  
 機能 波形パラメータ測定の特レンドの各カーソルの ON/OFF を設定 / 問い合わせします。  
 構文 :ANALYSIS:WPARAMETER<x>:TRENd:  
 MEASure:CURSor:C<x>:STATe {<Boolean>}  
 :ANALYSIS:WPARAMETER<x>:TRENd:  
 MEASure:CURSor:C<x>:STATe?  
 WPARAMETER<x> の <x> = 1、2  
 C<x> の <x> = 1、2  
 例 :ANALYSIS:WPARAMETER1:TRENd:MEASURE:  
 CURSOR:C1:STATe ON  
 :ANALYSIS:WPARAMETER1:TRENd:MEASURE:  
 CURSOR:C1:STATe? -> :ANALYSIS:  
 WPARAMETER1:TRENd:MEASURE:CURSOR:C1:  
 STATe 1

**:ANALYSIS:WPARAMETER<x>:TRENd:**  
**MEASure:CURSor:C<x>:VALue?**  
 機能 波形パラメータ測定の特レンドの各カーソルの 測定値を問い合わせします。  
 構文 :ANALYSIS:WPARAMETER<x>:TRENd:  
 MEASure:CURSor:C<x>:VALue?  
 WPARAMETER<x> の <x> = 1、2  
 C<x> の <x> = 1、2  
 例 :ANALYSIS:WPARAMETER1:TRENd:MEASURE:  
 CURSOR:C1:VALue? -> :ANALYSIS:  
 WPARAMETER1:TRENd:MEASURE:CURSOR:C1:  
 VALue 1.000E+00

**:ANALYSIS:WPARAMETER<x>:TRENd:**  
**MEASure:CURSor:DC?**  
 機能 波形パラメータ測定の特レンドのカーソル間の 測定に関するすべての設定値を問い合わせします。  
 構文 :ANALYSIS:WPARAMETER<x>:TRENd:  
 MEASure:CURSor:DC?  
 WPARAMETER<x> の <x> = 1、2  
 例 :ANALYSIS:WPARAMETER1:TRENd:MEASURE:  
 CURSOR:DC? -> :ANALYSIS:WPARAMETER1:  
 TRENd:MEASURE:CURSOR:DC:STATe 1

**:ANALYSIS:WPARAMETER<x>:TRENd:**  
**MEASure:CURSor:DC:STATe**  
 機能 波形パラメータ測定の特レンドのカーソル間の ON/OFF を設定 / 問い合わせします。  
 構文 :ANALYSIS:WPARAMETER<x>:TRENd:  
 MEASure:CURSor:DC:STATe {<Boolean>}  
 :ANALYSIS:WPARAMETER<x>:TRENd:  
 MEASure:CURSor:DC:STATe?  
 WPARAMETER<x> の <x> = 1、2  
 例 :ANALYSIS:WPARAMETER1:TRENd:MEASURE:  
 CURSOR:DC:STATe ON  
 :ANALYSIS:WPARAMETER1:TRENd:MEASURE:  
 CURSOR:DC:STATe? -> :ANALYSIS:  
 WPARAMETER1:TRENd:MEASURE:CURSOR:DC:  
 STATe 1

**:ANALYSIS:WPARAMETER<x>:TRENd:**  
**MEASure:CURSor:DC:VALue?**  
 機能 波形パラメータ測定の特レンドのカーソル間の 測定値を問い合わせします。  
 構文 :ANALYSIS:WPARAMETER<x>:TRENd:  
 MEASure:CURSor:DC:VALue?  
 WPARAMETER<x> の <x> = 1、2  
 例 :ANALYSIS:WPARAMETER1:TRENd:MEASURE:  
 CURSOR:DC:VALue? -> :ANALYSIS:  
 WPARAMETER1:TRENd:MEASURE:CURSOR:DC:  
 VALue 1.000E+00

**:ANALYSIS:WPARAMETER<x>:TRENd:**  
**MEASure:CURSor:LINKage**  
 機能 波形パラメータ測定の特レンドのカーソルのリ ンクの ON/OFF を設定 / 問い合わせします。  
 構文 :ANALYSIS:WPARAMETER<x>:TRENd:  
 MEASure:CURSor:LINKage {<Boolean>}  
 :ANALYSIS:WPARAMETER<x>:TRENd:  
 MEASure:CURSor:LINKage?  
 <x> = 1、2  
 例 :ANALYSIS:WPARAMETER1:TRENd:MEASURE:  
 CURSOR:LINKage ON  
 :ANALYSIS:WPARAMETER1:TRENd:MEASURE:  
 CURSOR:LINKage? -> :ANALYSIS:  
 WPARAMETER1:TRENd:MEASURE:CURSOR:  
 LINKage 1

### 5.3 ANALysis グループ

**:ANALysis:WPARAMeter<x>:TREND:  
MEASure:MODE**

機能 波形パラメータ測定の特レンドの自動測定  
のモードを設定 / 問い合わせします。

構文 :ANALysis:WPARAMeter<x>:TREND:  
MEASure:MODE {CURSor|OFF}  
:ANALysis:WPARAMeter<x>:TREND:  
MEASure:MODE?  
<x> = 1、2

例 :ANALYSIS:WPARAMETER1:TREND:MEASURE:  
MODE CURSOR  
:ANALYSIS:WPARAMETER1:TREND:MEASURE:  
MODE? -> :ANALYSIS:WPARAMETER1:TREND:  
MEASURE:MODE CURSOR

**:ANALysis:WPARAMeter<x>:TREND:  
VERTical**

機能 波形パラメータ測定の特レンドの垂直レンジを  
設定 / 問い合わせします。

構文 :ANALysis:WPARAMeter<x>:TREND:  
VERTical {<Nrf>,<Nrf>}  
:ANALysis:WPARAMeter<x>:TREND:  
VERTical?  
<x> = 1、2  
<Nrf> = -1.000E+31 ~ 1.000E+31(div)

例 :ANALYSIS:WPARAMETER1:TREND:  
VERTICAL 1,2  
:ANALYSIS:WPARAMETER1:TREND:VERTICAL?  
-> :ANALYSIS:WPARAMETER1:TREND:  
VERTICAL 2.000E+00,1.000E+00

**:ANALysis:XY<x>?**

機能 XY表示機能に関するすべての設定値を問  
い合わせます。

構文 :ANALysis:XY<x>?  
<x> = 1、2

例 :ANALYSIS:XY1? -> :ANALYSIS:XY1:GATE:  
ALEVEL HIGH;HYSTERESIS1 1.000E+00;  
HYSTERESIS2 1.000E+00;  
HYSTERESIS3 1.000E+00;  
HYSTERESIS4 1.000E+00;  
HYSTERESIS5 1.000E+00;  
HYSTERESIS6 1.000E+00;  
HYSTERESIS7 1.000E+00;  
HYSTERESIS8 1.000E+00;LEVEL1  
1.000E+00;LEVEL2 1.000E+00;  
LEVEL3 1.000E+00;LEVEL4 1.000E+00;  
LEVEL5 1.000E+00;LEVEL6 1.000E+00;  
LEVEL7 1.000E+00;LEVEL8 1.000E+00;  
TRACE 1;:ANALYSIS:XY1:MEASURE:CURSOR:  
X1:POSITION 1.000E+00;:ANALYSIS:XY1:  
MEASURE:CURSOR:X2:  
POSITION 2.000E+00;:ANALYSIS:XY1:  
MEASURE:CURSOR:XLINKAGE 1;Y1:  
POSITION 1.000E+00;:ANALYSIS:XY1:  
MEASURE:CURSOR:Y2:  
POSITION 2.000E+00;:ANALYSIS:XY1:  
MEASURE:CURSOR:YLINKAGE 1;:ANALYSIS:  
XY1:MEASURE:MODE CURSOR;XYINTEG:  
LOOP CLOSE;POLARITY CCW;:ANALYSIS:  
XY1:TRANGE 1.000E+00,0.000E+00;  
WINDOW MAIN;XTRACE 1;YTRACE 1

**:ANALysis:XY<x>:GATE?**

機能 XY表示のゲート機能に関するすべての設定値を  
問い合わせます。

構文 :ANALysis:XY<x>:GATE?  
<x> = 1、2

例 :ANALYSIS:XY1:GATE? -> :ANALYSIS:XY1:  
GATE:ALEVEL HIGH;  
HYSTERESIS1 1.000E+00;  
HYSTERESIS2 1.000E+00;  
HYSTERESIS3 1.000E+00;  
HYSTERESIS4 1.000E+00;  
HYSTERESIS5 1.000E+00;  
HYSTERESIS6 1.000E+00;  
HYSTERESIS7 1.000E+00;  
HYSTERESIS8 1.000E+00;  
LEVEL1 1.000E+00;LEVEL2 1.000E+00;  
LEVEL3 1.000E+00;LEVEL4 1.000E+00;  
LEVEL5 1.000E+00;LEVEL6 1.000E+00;  
LEVEL7 1.000E+00;LEVEL8 1.000E+00;  
TRACE 1

**:ANALYSIS:XY<x>:GATE:ALEVEL**

機能 XY表示のゲートのアクティブレベルを設定/問い合わせします。

構文 :ANALYSIS:XY<x>:GATE:  
ALEVEL {HIGH|LOW}  
:ANALYSIS:XY<x>:GATE:ALEVEL?  
<x> = 1, 2

例 :ANALYSIS:XY1:GATE:ALEVEL HIGH  
:ANALYSIS:XY1:GATE:ALEVEL?  
-> :ANALYSIS:XY1:GATE:ALEVEL HIGH

**:ANALYSIS:XY<x>:GATE:HYSTERESIS<x>**

機能 XY表示のゲートのヒステリシスを設定/問い合わせします。

構文 :ANALYSIS:XY<x>:GATE:  
HYSTERESIS<x> {<NRF>}  
:ANALYSIS:XY<x>:GATE:HYSTERESIS<x>?  
XY<x>の<x> = 1, 2  
HYSTERESIS<x>の<x> = 1~8  
<NRF> = 0~4(div)

例 :ANALYSIS:XY1:GATE:HYSTERESIS1 1  
:ANALYSIS:XY1:GATE:HYSTERESIS1?  
-> :ANALYSIS:XY1:GATE:  
HYSTERESIS1 1.000E+00

**:ANALYSIS:XY<x>:GATE:LEVEL<x>**

機能 XY表示のゲートのレベルを設定/問い合わせします。

構文 :ANALYSIS:XY<x>:GATE:LEVEL<x>  
{<NRF>,<電圧>,<電流>}  
:ANALYSIS:XY<x>:GATE:LEVEL<x>?  
XY<x>の<x> = 1, 2  
LEVEL<x>の<x> = 1~8

<NRF>,<電圧>,<電流> = 本体ユーザーズマニュアル参照。  
例 :ANALYSIS:XY1:GATE:LEVEL1 1  
:ANALYSIS:XY1:GATE:LEVEL1?  
-> :ANALYSIS:XY1:GATE:  
LEVEL1 1.000E+00

**:ANALYSIS:XY<x>:GATE:TRACE**

機能 XY表示のゲートトレースを設定/問い合わせします。

構文 :ANALYSIS:XY<x>:GATE:  
TRACE {<NRF>|OFF}  
:ANALYSIS:XY<x>:GATE:TRACE?  
<x> = 1, 2  
<NRF> = 1~8

例 :ANALYSIS:XY1:GATE:TRACE 1  
:ANALYSIS:XY1:GATE:TRACE?  
-> :ANALYSIS:XY1:GATE:TRACE 1

**:ANALYSIS:XY<x>:MEASURE?**

機能 XY表示の自動測定に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:XY<x>:MEASURE?  
<x> = 1, 2  
例 :ANALYSIS:XY1:MEASURE? -> :ANALYSIS:  
XY1:MEASURE:Cursors:X1:  
POSITION 1.000E+00;:ANALYSIS:XY1:  
MEASURE:Cursors:X2:  
POSITION 2.000E+00;:ANALYSIS:XY1:  
MEASURE:Cursors:XLINKAGE 1;Y1:  
POSITION 1.000E+00;:ANALYSIS:XY1:  
MEASURE:Cursors:Y2:  
POSITION 2.000E+00;:ANALYSIS:XY1:  
MEASURE:Cursors:YLINKAGE 1;:ANALYSIS:  
XY1:MEASURE:MODE CURSOR;XYINTEG:  
LOOP CLOSE;POLARITY CCW

**:ANALYSIS:XY<x>:MEASURE:Cursors?**

機能 XY表示のカーソル測定に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:XY<x>:MEASURE:Cursors?  
<x> = 1, 2

例 :ANALYSIS:XY1:MEASURE:Cursors?  
-> :ANALYSIS:XY1:MEASURE:Cursors:X1:  
POSITION 1.000E+00;:ANALYSIS:XY1:  
MEASURE:Cursors:X2:  
POSITION 2.000E+00;:ANALYSIS:XY1:  
MEASURE:Cursors:XLINKAGE 1;Y1:  
POSITION 1.000E+00;:ANALYSIS:XY1:  
MEASURE:Cursors:Y2:  
POSITION 2.000E+00;:ANALYSIS:XY1:  
MEASURE:Cursors:YLINKAGE 1

**:ANALYSIS:XY<x>:MEASURE:Cursors:XLINKAGE**

機能 XY表示の水平カーソルのリンクのON/OFFを設定/問い合わせします。

構文 :ANALYSIS:XY<x>:MEASURE:Cursors:  
XLINKAGE {<Boolean>}  
:ANALYSIS:XY<x>:MEASURE:Cursors:  
XLINKAGE?  
<x> = 1, 2

例 :ANALYSIS:XY1:MEASURE:Cursors:  
XLINKAGE ON  
:ANALYSIS:XY1:MEASURE:Cursors:  
XLINKAGE? -> :ANALYSIS:XY1:MEASURE:  
Cursors:XLINKAGE 1

**:ANALYSIS:XY<x>:MEASURE:Cursors:X<x>?**

機能 XY表示の水平カーソルに関するすべての設定値を問い合わせます。

構文 :ANALYSIS:XY<x>:MEASURE:Cursors:X<x>?  
XY<x>の<x> = 1, 2  
X<x>の<x> = 1, 2

例 :ANALYSIS:XY1:MEASURE:Cursors:X1?  
-> :ANALYSIS:XY1:MEASURE:Cursors:X1:  
POSITION 1.000E+00



### 5.3 ANALYSIS グループ

#### **:ANALYSIS:XY<x>:MEASURE:Cursors:X<x>:POSITION**

機能 XY 表示の水平カーソルの位置を設定 / 問い合わせします。

構文 :ANALYSIS:XY<x>:MEASURE:Cursors:X<x>:POSITION {<NRF>}  
:ANALYSIS:XY<x>:MEASURE:Cursors:X<x>:POSITION?  
XY<x> の <x> = 1、2  
X<x> の <x> = 1、2  
<NRF> = -4 ~ 4(div)

例 :ANALYSIS:XY1:MEASURE:Cursors:X1:POSITION 1  
:ANALYSIS:XY1:MEASURE:Cursors:X1:POSITION? -> :ANALYSIS:XY1:MEASURE:Cursors:X1:POSITION 1.000E+00

#### **:ANALYSIS:XY<x>:MEASURE:Cursors:X<x>:VALUE?**

機能 XY 表示の水平カーソルの電圧値を問い合わせます。

構文 :ANALYSIS:XY<x>:MEASURE:Cursors:X<x>:VALUE?  
XY<x> の <x> = 1、2  
X<x> の <x> = 1、2

例 :ANALYSIS:XY1:MEASURE:Cursors:X1:VALUE? -> :ANALYSIS:XY1:MEASURE:Cursors:X1:VALUE 1.000E+00

#### **:ANALYSIS:XY<x>:MEASURE:Cursors:YLINKage**

機能 XY 表示の垂直カーソルのリンクの ON/OFF を設定 / 問い合わせします。

構文 :ANALYSIS:XY<x>:MEASURE:Cursors:YLINKage {<Boolean>}  
:ANALYSIS:XY<x>:MEASURE:Cursors:YLINKage?  
<x> = 1、2

例 :ANALYSIS:XY1:MEASURE:Cursors:YLINKage ON  
:ANALYSIS:XY1:MEASURE:Cursors:YLINKage? -> :ANALYSIS:XY1:MEASURE:Cursors:YLINKage 1

#### **:ANALYSIS:XY<x>:MEASURE:Cursors:Y<x>?**

機能 XY 表示の垂直カーソルに関するすべての設定値を問い合わせます。

構文 :ANALYSIS:XY<x>:MEASURE:Cursors:Y<x>?  
XY<x> の <x> = 1、2  
Y<x> の <x> = 1、2

例 :ANALYSIS:XY1:MEASURE:Cursors:Y1?  
-> :ANALYSIS:XY1:MEASURE:Cursors:Y1:POSITION 1.000E+00

#### **:ANALYSIS:XY<x>:MEASURE:Cursors:Y<x>:POSITION**

機能 XY 表示の垂直カーソルの位置を設定 / 問い合わせします。

構文 :ANALYSIS:XY<x>:MEASURE:Cursors:Y<x>:POSITION {<NRF>}  
:ANALYSIS:XY<x>:MEASURE:Cursors:Y<x>:POSITION?  
XY<x> の <x> = 1、2  
Y<x> の <x> = 1、2  
<NRF> = -4 ~ 4(div)

例 :ANALYSIS:XY1:MEASURE:Cursors:Y1:POSITION 1  
:ANALYSIS:XY1:MEASURE:Cursors:Y1:POSITION? -> :ANALYSIS:XY1:MEASURE:Cursors:Y1:POSITION 1.000E+00

#### **:ANALYSIS:XY<x>:MEASURE:Cursors:Y<x>:VALUE?**

機能 XY 表示の垂直カーソルの電圧値を問い合わせます。

構文 :ANALYSIS:XY<x>:MEASURE:Cursors:Y<x>:VALUE?  
XY<x> の <x> = 1、2  
Y<x> の <x> = 1、2

例 :ANALYSIS:XY1:MEASURE:Cursors:Y1:VALUE? -> :ANALYSIS:XY1:MEASURE:Cursors:Y1:VALUE 1.000E+00

#### **:ANALYSIS:XY<x>:MEASURE:MODE**

機能 XY 表示の自動測定モードを設定 / 問い合わせします。

構文 :ANALYSIS:XY<x>:MEASURE:MODE {Cursors|OFF|XYINtegr}  
:ANALYSIS:XY<x>:MEASURE:MODE?  
<x> = 1、2

例 :ANALYSIS:XY1:MEASURE:MODE Cursors  
:ANALYSIS:XY1:MEASURE:MODE?  
-> :ANALYSIS:XY1:MEASURE:MODE Cursors

#### **:ANALYSIS:XY<x>:MEASURE:XYINtegr?**

機能 XY 表示の積分に関するすべての設定値を問い合わせます。

構文 :ANALYSIS:XY<x>:MEASURE:XYINtegr?  
<x> = 1、2

例 :ANALYSIS:XY1:MEASURE:XYINtegr?  
-> :ANALYSIS:XY1:MEASURE:XYINtegr:LOOP CLOSE;POLARITY CCW

**:ANALYSIS:XY<x>:MEASURE:XYINTEG:LOOP**

機能 XY表示の積分方法を設定 / 問い合わせします。  
 構文 :ANALYSIS:XY<x>:MEASURE:XYINTEG:LOOP  
 {CLOSE|OPEN}  
 :ANALYSIS:XY<x>:MEASURE:XYINTEG:LOOP?  
 <x> = 1, 2  
 例 :ANALYSIS:XY1:MEASURE:XYINTEG:  
 LOOP CLOSE  
 :ANALYSIS:XY1:MEASURE:XYINTEG:LOOP?  
 -> :ANALYSIS:XY1:MEASURE:XYINTEG:  
 LOOP CLOSE

**:ANALYSIS:XY<x>:MEASURE:XYINTEG:POLARITY**

機能 XY表示の積分方向を設定 / 問い合わせします。  
 構文 :ANALYSIS:XY<x>:MEASURE:XYINTEG:  
 POLARITY {CCW|CW}  
 :ANALYSIS:XY<x>:MEASURE:XYINTEG:  
 POLARITY?  
 <x> = 1, 2  
 例 :ANALYSIS:XY1:MEASURE:XYINTEG:  
 POLARITY CCW  
 :ANALYSIS:XY1:MEASURE:XYINTEG:  
 POLARITY? -> :ANALYSIS:XY1:MEASURE:  
 XYINTEG:POLARITY CCW

**:ANALYSIS:XY<x>:MEASURE:XYINTEG:VALUE?**

機能 XY表示の積分値を問い合わせします。  
 構文 :ANALYSIS:XY<x>:MEASURE:XYINTEG:  
 VALUE?  
 <x> = 1, 2  
 例 :ANALYSIS:XY1:MEASURE:XYINTEG:VALUE?  
 -> :ANALYSIS:XY1:MEASURE:XYINTEG:  
 VALUE 1.000E+00

**:ANALYSIS:XY<x>:TRANGE (Time Range)**

機能 XY表示の測定範囲を設定 / 問い合わせします。  
 構文 :ANALYSIS:XY<x>:TRANGE {<NRF>,<NRF>}  
 :ANALYSIS:XY<x>:TRANGE?  
 <x> = 1, 2  
 <NRF> = - 5 ~ 5(div)  
 例 :ANALYSIS:XY1:TRANGE 0,1  
 :ANALYSIS:XY1:TRANGE? -> :ANALYSIS:  
 XY1:TRANGE 1.000E+00,  
 0.000E+00

**:ANALYSIS:XY<x>:WINDOW**

機能 XY表示の測定対象ウィンドウを設定 / 問い合わせします。  
 構文 :ANALYSIS:XY<x>:WINDOW {MAIN|Z1|Z2}  
 :ANALYSIS:XY<x>:WINDOW?  
 <x> = 1, 2  
 例 :ANALYSIS:XY1:WINDOW MAIN  
 :ANALYSIS:XY1:WINDOW? -> :ANALYSIS:  
 XY1:WINDOW MAIN

**:ANALYSIS:XY<x>:XTRACE**

機能 XY表示のX軸トレースを設定 / 問い合わせします。  
 構文 :ANALYSIS:XY<x>:XTRACE {<NRF>}  
 :ANALYSIS:XY<x>:XTRACE?  
 <x> = 1, 2  
 <NRF> = 1 ~ 8  
 例 :ANALYSIS:XY1:XTRACE 1  
 :ANALYSIS:XY1:XTRACE? -> :ANALYSIS:  
 XY1:XTRACE 1

**:ANALYSIS:XY<x>:YTRACE**

機能 XY表示のY軸トレースを設定 / 問い合わせします。  
 構文 :ANALYSIS:XY<x>:YTRACE {<NRF>}  
 :ANALYSIS:XY<x>:YTRACE?  
 <x> = 1, 2  
 <NRF> = 1 ~ 8  
 例 :ANALYSIS:XY1:YTRACE 1  
 :ANALYSIS:XY1:YTRACE? -> :ANALYSIS:  
 XY1:YTRACE 1

## 5.4 ASETup グループ

### **:ASETup:EXECute**

機能 オートセットアップを実行します。

構文 :ASETup:EXECute

例 :ASETUP:EXECUTE

### **:ASETup:UNDO**

機能 実行したオートセットアップを取り消します。

構文 :ASETup:UNDO

例 :ASETUP:UNDO

## 5.5 CALibrate グループ

### **:CALibrate?**

機能 キャリブレーションに関するすべての設定値を  
問い合わせます。

構文 :CALibrate?

例 :CALIBRATE? -> :CALIBRATE:MODE AUTO

### **:CALibrate:EXECute**

機能 キャリブレーションを実行します。

構文 :CALibrate:EXECute

例 :CALIBRATE:EXECUTE

### **:CALibrate:MODE**

機能 オートキャリブレーションの ON/OFF を設定 / 問  
い合わせします。

構文 :CALibrate:MODE {AUTO|OFF}

:CALibrate:MODE?

例 :CALIBRATE:MODE AUTO

:CALIBRATE:MODE?

-> :CALIBRATE:MODE AUTO

## 5.6 CHANnel グループ

### **:CHANnel<x>?**

機能 各チャンネルに関するすべての設定値を問い合わせます。

構文 :CHANnel<x>?  
<x> = 1 ~ 4

例 :CHANNEL1? -> :CHANNEL1:SELECT INPUT;  
DISPLAY 1;BWIDTh FULL;COUPLing DC;  
DESKEW 0.000E+00;INVERT 0;LABel:  
DEFINE "CH1";MODE 1;:CHANNEL1:  
OCANCEL 0;OFFSEt 0.000E+00;  
POSITION 0.000E+00;PROBE:MODE 1;:  
CHANNEL1:SVALUE 0;VDIV 1.000E+00

### **:CHANnel<x>:AScale[:EXECute]**

機能 各チャンネルのオートスケールを実行します。

構文 CHANnel<x>:AScale[:EXECute]  
<x> = 1 ~ 4

例 CHANNEL1:ASCALE:EXECUTE

### **:CHANnel<x>:BWIDTh**

機能 各チャンネルの入力フィルタを設定 / 問い合わせします。

構文 :CHANnel<x>:BWIDTh {<周波数>|FULL}  
:CHANnel<x>:BWIDTh?  
<x> = 1 ~ 4

<周波数> = 本体ユーザーズマニュアル参照。  
例 :CHANNEL1:BWIDTh FULL  
:CHANNEL1:BWIDTh? -> :CHANNEL1:  
BWIDTh FULL

### **:CHANnel<x>:COUPLing**

機能 各チャンネルの入力カップリングを設定 / 問い合わせします。

構文 :CHANnel<x>:COUPLing {AC|DC|DC50|GND}  
:CHANnel<x>:COUPLing?  
<x> = 1 ~ 4

例 :CHANNEL1:COUPLing GND  
:CHANNEL1:COUPLing? -> :CHANNEL1:  
COUPLing GND

### **:CHANnel<x>:DESKew**

機能 各チャンネルのスキュー補正を設定 / 問い合わせします。

構文 :CHANnel<x>:DESKew {<時間>}  
:CHANnel<x>:DESKew?  
<x> = 1 ~ 4

<時間> = - 80ns ~ 80ns(10ps ステップ)  
例 :CHANNEL1:DESKEW 1NS  
:CHANNEL1:DESKEW? -> :CHANNEL1:  
DESKEW 1.000E-09

### **:CHANnel<x>:DISPlay**

機能 各チャンネルの表示の ON/OFF を設定 / 問い合わせします。

構文 :CHANnel<x>:DISPlay {<Boolean>}  
:CHANnel<x>:DISPlay?  
<x> = 1 ~ 4

例 :CHANNEL1:DISPlay ON  
:CHANNEL1:DISPlay? -> :CHANNEL1:  
DISPlay 1

### **:CHANnel<x>:INVert**

機能 各チャンネルの反転表示の ON/OFF を設定 / 問い合わせします。

構文 :CHANnel<x>:INVert {<Boolean>}  
:CHANnel<x>:INVert?  
<x> = 1 ~ 4

例 :CHANNEL1:INVERT ON  
:CHANNEL1:INVERT -> :CHANNEL1:  
INVERT 1

### **:CHANnel<x>:LABel?**

機能 各チャンネルの波形ラベル名に関するすべての設定値を問い合わせます。

構文 :CHANnel<x>:LABel?  
<x> = 1 ~ 4

例 :CHANNEL1:LABel? -> :CHANNEL1:LABel:  
DEFINE "CH1";MODE 0

### **:CHANnel<x>:LABel[:DEFine]**

機能 各チャンネルの波形ラベル名を設定 / 問い合わせします。

構文 :CHANnel<x>:LABel[:DEFine] {<文字列>}  
:CHANnel<x>:LABel[:DEFine]?  
<x> = 1 ~ 4  
<文字列> = 8 文字以内

例 :CHANNEL1:LABel:DEFINE "CH1"  
:CHANNEL1:LABel:DEFINE? -> :CHANNEL1:  
LABel:DEFINE "CH1"

### **:CHANnel<x>:LABel:MODE**

機能 各チャンネルの波形ラベル名表示の ON/OFF を設定 / 問い合わせします。

構文 :CHANnel<x>:LABel:MODE {<Boolean>}  
:CHANnel<x>:LABel:MODE?  
<x> = 1 ~ 4

例 :CHANNEL1:LABel:MODE ON  
:CHANNEL1:LABel:MODE? -> :CHANNEL1:  
LABel:MODE 1

**:CHANnel<x>:OCANcel**

機能 各チャンネルのオフセットキャンセルの ON/OFF を設定 / 問い合わせします。

構文 :CHANnel<x>:OCANcel {<Boolean>}  
:CHANnel<x>:OCANcel?  
<x> = 1 ~ 4

例 :CHANNEL1:OCANCEL ON  
:CHANNEL1:OCANCEL? -> :CHANNEL1:  
OCANCEL 1

**:CHANnel<x>:OFFSet**

機能 各チャンネルのオフセット電圧を設定 / 問い合わせします。

構文 :CHANnel<x>:OFFSet {<電圧>|<電流>}  
:CHANnel<x>:OFFSet?  
<x> = 1 ~ 4

<電圧>、<電流> = 本体ユーザーズマニュアル参照。

例 :CHANNEL1:OFFSET 1V  
:CHANNEL1:OFFSET? -> :CHANNEL1:  
OFFSET 1.000E+00

**:CHANnel<x>:POSition**

機能 各チャンネルの垂直ポジションを設定 / 問い合わせします。

構文 :CHANnel<x>:POSition {<Nrf>}  
:CHANnel<x>:POSition?  
<x> = 1 ~ 4

<Nrf> = -4 ~ 4(div)

例 :CHANNEL1:POSITION 1  
:CHANNEL1:POSITION? -> :CHANNEL1:  
POSITION 1.000E+00

**:CHANnel<x>:PROBe?**

機能 各チャンネルのプローブ減衰比に関するすべての設定値を問い合わせます。

構文 :CHANnel<x>:PROBe?  
<x> = 1 ~ 4

例 :CHANNEL1:PROBE? -> :CHANNEL1:PROBE:  
MODE 1

**:CHANnel<x>:PROBe[:MODE]**

機能 各チャンネルのプローブ減衰比を設定 / 問い合わせします。

構文 :CHANnel<x>:PROBe[:MODE] {<Nrf>|AUTO|  
C1|C10|C100}  
:CHANnel<x>:PROBe[:MODE]?  
<x> = 1 ~ 4

<Nrf> = 1、2、5、10、20、50、100、200、500、1000

例 :CHANNEL1:PROBE:MODE 1  
:CHANNEL1:PROBE:MODE? -> :CHANNEL1:  
PROBE:MODE 1

**:ChanneL<x>:PROBe:AUTO?**

機能 各チャンネルの AUTO 時のプローブ減衰比を問い合わせます。

構文 :ChanneL<x>:PROBe:AUTO?  
<x> = 1 ~ 4

例 :CHANNEL1:PROBE:AUTO? -> :CHANNEL1:  
PROBE:AUTO 1

**:CHANnel<x>:PZCalibrate[:EXECute]**

機能 各チャンネルの電流プローブのゼロ補正を実行します。

構文 :CHANnel<x>:PZCalibrate[:EXECute]  
<x> = 1 ~ 4

例 :CHANNEL1:PZCALIBRATE:EXECUTE

**:CHANnel<x>:SElect**

機能 各入力チャンネルに割り当てる波形 (入力 / 演算) を設定 / 問い合わせします。

構文 :CHANnel<x>:SElect {INPut|MATH}  
:CHANnel<x>:SElect?  
<x> = 1 ~ 4

例 :CHANNEL1:SELECT INPUT  
:CHANNEL1:SELECT? -> :CHANNEL1:  
SELECT INPUT

**:CHANnel<x>:SVALue (Scale VALUE)**

機能 各チャンネルのスケール値表示の ON/OFF を設定 / 問い合わせします。

構文 :CHANnel<x>:SVALue {<Boolean>}  
:CHANnel<x>:SVALue?  
<x> = 1 ~ 4

例 :CHANNEL1:SVALUE ON  
:CHANNEL1:SVALUE? -> :CHANNEL1:  
SVALUE 1

**:CHANnel<x>:VDIV**

機能 各チャンネルの垂直軸感度 (V/div) を設定 / 問い合わせします。

構文 :CHANnel<x>:VDIV {<電圧>|<電流>}  
:CHANnel<x>:VDIV?  
<x> = 1 ~ 4

<電圧>、<電流> = 本体ユーザーズマニュアル参照。

例 :CHANNEL1:VDIV 5V  
:CHANNEL1:VDIV? -> :CHANNEL1:  
VDIV 5.000E+00

## 5.7 CLEar グループ

### **:CLEar:ACCumulate**

機能 アキュムレート波形をクリアします。

構文 :CLEar:ACCumulate

例 :CLEAR:ACCUMULATE

### **:CLEar[:HISTory]**

機能 ヒストリ波形をクリアします。

構文 :CLEar[:HISTory]

例 :CLEAR:HISTORY

### **:CLEar:SNAP**

機能 スナップショット波形をクリアします。

構文 :CLEar:SNAP

例 :CLEAR:SNAP

## 5.8 COMMunicate グループ

COMMunicate グループは、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 通信に関するすべての設定値を問い合わせます。  
構文 :COMMunicate?  
例 :COMMUNICATE? -> :COMMUNICATE:  
HEADER 1;OPSE 352;OVERLAP 352;  
VERBOSE 1

### :COMMunicate:HEADer

機能 クエリに対する応答を、ヘッダを付けて返送するか(例 CHANNEL1:VOLTAGE:PROBE 10)、付けな  
いで返送するか(例 10)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 USB、イーサネットインタフェース(オプション)  
用のコマンドです。

### :COMMunicate:OPSE (Operation Pending Status Enable register)

機能 \*OPC、\*OPC?、\*WAI の対象となるオーバラッ  
プコマンドを設定/問い合わせします。  
構文 :COMMunicate:OPSE <Register>  
:COMMunicate:OPSE?  
<Register> = 0 ~ 65535、:COMMunicate:  
OVERlap コマンドの図参照  
例 :COMMUNICATE:OPSE 65535  
:COMMUNICATE:OPSE?  
-> :COMMUNICATE:OPSE 2400  
解説 上の例では、全ビットを1にして、すべてのオー  
バラップコマンドを対象にしています。ただし、  
0 固定のビットは1にならないので、問い合わせ  
に対してはビット 5、6、8、11 が1になっています。

### :COMMunicate:OPSR? (Operation Pending Status Register)

機能 オペレーションペンディングステータスレジス  
タの値を問い合わせます。  
構文 :COMMunicate:OPSR?  
例 :COMMUNICATE:OPSR? -> 0  
解説 オペレーションペンディングステータスレジスタ  
については、:COMMunicate:OVERlap コマンドの  
図を参照してください。

### :COMMunicate:OVERlap

機能 オーバラップ動作にするコマンドを設定/問い合  
わせします。  
構文 :COMMunicate:OVERlap <Register>  
:COMMunicate:OVERlap?  
<Register> = 0 ~ 65535  
例 :COMMUNICATE:OVERLAP 65535  
:COMMUNICATE:OVERLAP?  
-> :COMMUNICATE:OVERLAP 2400

解説  
・上の例では、全ビットを1にして、すべてのオー  
バラップコマンドを対象にしています。ただし、  
0 固定のビットは1にならないので、問い合わ  
せに対してはビット 5、6、8、11 が1になって  
います。  
・:COMMunicate:OVERlap を使った同期のとり  
方については、4-7 ページを参照してください。  
・上の例では、ビット 5、6、8、11 を1にして、  
すべてのオーバラップコマンドを対象にしてい  
ます。

オペレーションペンディングステータスレジスタ  
/オーバラップイネーブルレジスタ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	SCH	0	0	HST	0	ACS	PRN	0	0	0	0	0

ビット 5(PRN) = 1 のとき:

内蔵プリンタ動作未完了

ビット 6(ACS) = 1 のとき:

メディアへのアクセス未完了

ビット 8(HST) = 1 のとき:

ヒストリ検索実行未完了

ビット 11(SCH) = 1 のとき:

サーチ検索実行未完了



## 5.8 COMMunicate グループ

### **: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 :COMMunicate:REMOte {<Boolean>}  
:COMMunicate:REMOte?

例 :COMMunicate:REMOte ON  
:COMMunicate:REMOte? -> :COMMunicate:REMOte 1

解説 USB、イーサネットインタフェース (オプション) 用のコマンドです。

### **:COMMunicate:STATus?**

機能 回線固有のステータスを問い合わせます。

構文 :COMMunicate:STATus?

例 :COMMunicate:STATus? -> :COMMunicate:STATus 0

解説 ステータスの各ビットの意味は次のとおりです。

ビット	GP-IB
0	回復不能な送信エラー
1	常に 0
2	常に 0
3 ~	常に 0

USB、ネットワークでは、常に 0 が返ります。ステータスは要因が発生したときに該当するビットがセットされ、読むとクリアされます。

### **: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 (例 CHANNEL1:VOLTAGE:PROBE 10)、省略形で返送するか (例 CHAN:PROB 10) を設定 / 問い合わせします。

構文 :COMMunicate:VERBose {<Boolean>}  
:COMMunicate:VERBose?

例 :COMMunicate:VERBose ON  
:COMMunicate:VERBose?  
-> :COMMunicate:VERBose 1

### **: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 :COMMunicate:WAIT <Register>  
<Register> = 0 ~ 65535 (拡張イベントレジスタ、6-5 ページ参照)

例 :COMMunicate:WAIT 65535

解説 :COMMunicate:WAIT を使った同期のとり方については、4-8 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 :COMMunicate:WAIT? <Register>  
<Register> = 0 ~ 65535 (拡張イベントレジスタ、6-5 ページ参照)

例 :COMMunicate:WAIT? 65535 -> 1

## 5.9 CURSor グループ

### :CURSor?

機能 カーソル測定に関するすべての設定値を問い合わせます。

構文 :CURSor?

例 :CURSOR? -> :CURSOR:DISPLAY 1;  
HORIZONTAL:BASIC:DV:STATE 1;:CURSOR:  
HORIZONTAL:BASIC:LINKAGE 1;V1:  
POSITION 1.000E+00;STATE 0;:CURSOR:  
HORIZONTAL:BASIC:V2:  
POSITION 1.000E+00;STATE 0;:CURSOR:  
HORIZONTAL:CALCULATION:DEFINE1 "V1";  
DEFINE2 "V2";DEFINE3 "V1";  
DEFINE4 "V2";STATE1 0;STATE2 0;  
STATE3 0;STATE4 0;:CURSOR:HORIZONTAL:  
TRACE 1;:CURSOR:MARKER:CALCULATION:  
DEFINE1 "T2";DEFINE2 "V2";  
DEFINE3 "V3";DEFINE4 "V4";STATE1 0;  
STATE2 0;STATE3 0;STATE4 0;:CURSOR:  
MARKER:CM1:DT2:STATE 1;:CURSOR:  
MARKER:CM1:DT3:STATE 1;:CURSOR:  
MARKER:CM1:DT4:STATE 1;:CURSOR:  
MARKER:CM1:DV2:STATE 1;:CURSOR:  
MARKER:CM1:DV3:STATE 1;:CURSOR:  
MARKER:CM1:DV4:STATE 1;:CURSOR:  
MARKER:CM1:POSITION 1.000E+00;T:  
STATE 1;:CURSOR:MARKER:CM1:TRACE 1;V:  
STATE 1;:CURSOR:MARKER:CM2:DT1:  
STATE 1;:CURSOR:MARKER:CM2:DT3:  
STATE 1;:CURSOR:MARKER:CM2:DT4:  
STATE 1;:CURSOR:MARKER:CM2:DV1:  
STATE 1;:CURSOR:MARKER:CM2:DV3:  
STATE .....

### :CURSor:DISPlay

機能 カーソルの ON/OFF を設定 / 問い合わせします。

構文 :CURSor:DISPlay {<Boolean>}

例 :CURSOR:DISPlay?  
:CURSOR:DISPLAY ON  
:CURSOR:DISPLAY? -> :CURSOR:DISPLAY 1

### :CURSor:HORizontal?

機能 水平カーソルに関するすべての設定値を問い合わせます。

構文 :CURSor:HORizontal?

例 :CURSOR:HORIZONTAL? -> :CURSOR:  
HORIZONTAL:BASIC:DV:STATE 1;:  
CURSOR:HORIZONTAL:BASIC:LINKAGE 1;V1:  
POSITION 1.000E+00;STATE 0;:CURSOR:  
HORIZONTAL:BASIC:V2:  
POSITION 1.000E+00;STATE 0;:CURSOR:  
HORIZONTAL:CALCULATION:DEFINE1 "V1";  
DEFINE2 "V2";DEFINE3 "V1";  
DEFINE4 "V2";STATE1 0;STATE2 0;  
STATE3 0;STATE4 0;:CURSOR:HORIZONTAL:  
TRACE 1

### :CURSor:HORizontal:BASic?

機能 水平カーソルの Basic アイテムに関するすべての設定値を問い合わせます。

構文 :CURSor:HORizontal:BASic?

例 :CURSOR:HORIZONTAL:BASIC? -> :CURSOR:  
HORIZONTAL:BASIC:DV:STATE 1;:  
CURSOR:HORIZONTAL:BASIC:LINKAGE 1;V1:  
POSITION 1.000E+00;STATE 0;:CURSOR:  
HORIZONTAL:BASIC:V2:  
POSITION 1.000E+00;STATE 0

### :CURSor:HORizontal[:BASic]:ALL

機能 水平カーソルの Basic アイテムすべてを ON/OFF します。

構文 :CURSor:HORizontal[:BASic]:

ALL {<Boolean>}

例 :CURSOR:HORIZONTAL:BASIC:ALL ON

### :CURSor:HORizontal[:BASic]:DV?

機能 水平カーソルの  $\Delta V$  測定に関するすべての設定値を問い合わせます。

構文 :CURSor:HORizontal[:BASic]:DV?

例 :CURSOR:HORIZONTAL:BASIC:DV?  
-> :CURSOR:HORIZONTAL:BASIC:DV:  
STATE 1

### :CURSor:HORizontal[:BASic]:DV:STATE

機能 水平カーソルの  $\Delta V$  測定の ON/OFF を設定 / 問い合わせします。

構文 :CURSor:HORizontal[:BASic]:DV:STATE

{<Boolean>}

:CURSor:HORizontal[:BASic]:DV:STATE?

例 :CURSOR:HORIZONTAL:BASIC:DV:STATE ON  
:CURSOR:HORIZONTAL:BASIC:DV:STATE?  
-> :CURSOR:HORIZONTAL:BASIC:DV:  
STATE 1

### :CURSor:HORizontal[:BASic]:DV:VALue?

機能 水平カーソル間の電圧値を問い合わせます。

構文 :CURSor:HORizontal[:BASic]:DV:VALue?

例 :CURSOR:HORIZONTAL:BASIC:DV:VALUE?  
-> :CURSOR:HORIZONTAL:BASIC:DV:  
VALUE 1.000E+00

### :CURSor:HORizontal[:BASic]:LINKage

機能 水平カーソルのリンクの ON/OFF を設定 / 問い合わせします。

構文 :CURSor:HORizontal[:BASic]:LINKage

{<Boolean>}

:CURSor:HORizontal[:BASic]:LINKage?

例 :CURSOR:HORIZONTAL:BASIC:LINKAGE ON  
:CURSOR:HORIZONTAL:BASIC:LINKAGE?  
-> :CURSOR:HORIZONTAL:BASIC:LINKAGE 1

## 5.9 CURSor グループ

### **:CURSor:HORizontal[:BASic]:V<x>?**

機能 各水平カーソルに関するすべての設定値を問い合わせます。

構文 :CURSor:HORizontal[:BASic]:V<x>?  
<x> = 1、2

例 :CURSOR:HORIZONTAL:BASIC:V1?  
-> :CURSOR:HORIZONTAL:BASIC:V1:  
POSITION 1.000E+00;STATE 1

### **:CURSor:HORizontal[:BASic]:V<x>:JUMP**

機能 各水平カーソルを指定したズーム波形の中央位置にジャンプします。

構文 :CURSor:HORizontal[:BASic]:V<x>:  
JUMP {Z1|Z2}  
<x> = 1、2

例 :CURSOR:HORIZONTAL:BASIC:V1:JUMP Z1

### **:CURSor:HORizontal[:BASic]:V<x>:POSition**

機能 各水平カーソルの位置を設定 / 問い合わせします。

構文 :CURSor:HORizontal[:BASic]:V<x>:  
POSITION {<NRf>}  
:CURSor:HORizontal[:BASic]:V<x>:  
POSITION?  
<x> = 1、2  
<NRf> = -4 ~ 4(div)

例 :CURSOR:HORIZONTAL:BASIC:V1:  
POSITION 1  
:CURSOR:HORIZONTAL:BASIC:V1:POSITION?  
-> :CURSOR:HORIZONTAL:BASIC:V1:  
POSITION 1.000E+00

### **:CURSor:HORizontal[:BASic]:V<x>:STATE**

機能 各水平カーソルの ON/OFF を設定 / 問い合わせします。

構文 :CURSor:HORizontal[:BASic]:V<x>:  
STATE {<Boolean>}  
:CURSor:HORizontal[:BASic]:V<x>:  
STATE?  
<x> = 1、2

例 :CURSOR:HORIZONTAL:BASIC:V1:STATE ON  
:CURSOR:HORIZONTAL:BASIC:V1:STATE?  
-> :CURSOR:HORIZONTAL:BASIC:V1:  
STATE 1

### **:CURSor:HORizontal[:BASic]:V<x>:VALue?**

機能 各水平カーソルの電圧値を問い合わせます。

構文 :CURSor:HORizontal[:BASic]:V<x>:  
VALue?  
<x> = 1、2

例 :CURSOR:HORIZONTAL:BASIC:V1:VALUE?  
-> :CURSOR:HORIZONTAL:BASIC:V1:  
VALUE 1.000E+00

### **:CURSor:HORizontal:CALCulation?**

機能 水平カーソルの Calc アイテムに関するすべての設定値を問い合わせます。

構文 :CURSor:HORizontal:CALCulation?  
:CURSOR:HORIZONTAL:CALCULATION?  
例 -> :CURSOR:HORIZONTAL:CALCULATION:  
DEFINE1 "V1";DEFINE2 "V2";  
DEFINE3 "V1";DEFINE4 "V2";STATE1 0;  
STATE2 0;STATE3 0;STATE4 0

### **:CURSor:HORizontal:CALCulation:ALL**

機能 水平カーソルの Calc アイテムすべてを ON/OFF します。

構文 :CURSor:HORizontal:CALCulation:  
ALL {<Boolean>}

例 :CURSOR:HORIZONTAL:CALCULATION:ALL ON

### **:CURSor:HORizontal:CALCulation:DEFine<x>**

機能 水平カーソルの Calc アイテムの演算式を設定 / 問い合わせします。

構文 :CURSor:HORizontal:CALCulation:  
DEFine<x> {<文字列>}  
:CURSor:HORizontal:CALCulation:  
DEFine<x>?  
<x> = 1 ~ 4  
<文字列> = 128 文字以内

例 :CURSOR:HORIZONTAL:CALCULATION:  
DEFINE1 "V1"  
:CURSOR:HORIZONTAL:CALCULATION:  
DEFINE1? -> :CURSOR:HORIZONTAL:  
CALCULATION:DEFINE1 "V1"

### **:CURSor:HORizontal:CALCulation:STATE<x>**

機能 水平カーソルの Calc アイテムの ON/OFF を設定 / 問い合わせします。

構文 :CURSor:HORizontal:CALCulation:  
STATE<x> {<Boolean>}  
:CURSor:HORizontal:CALCulation:  
STATE<x>?  
<x> = 1 ~ 4

例 :CURSOR:HORIZONTAL:CALCULATION:  
STATE1 ON  
:CURSOR:HORIZONTAL:CALCULATION:  
STATE1? -> :CURSOR:HORIZONTAL:  
CALCULATION:STATE1 1



## 5.9 CURSor グループ

### :CURSor:MARKer:CM<x>?

機能 各マーカーカーソルに関するすべての設定値を問い合わせます。

構文 :CURSor:MARKer:CM<x>?

<x> = 1 ~ 4

例 :CURSOR:MARKER:CM1? -> :CURSOR:  
MARKER:CM1:DT2:STATE 1;:CURSOR:  
MARKER:CM1:DT3:STATE 1;:CURSOR:  
MARKER:CM1:DT4:STATE 1;:CURSOR:  
MARKER:CM1:DV2:STATE 1;:CURSOR:  
MARKER:CM1:DV3:STATE 1;:CURSOR:  
MARKER:CM1:DV4:STATE 1;:CURSOR:  
MARKER:CM1:POSITION 1.000E+00;T:  
STATE 1;:CURSOR:MARKER:CM1:TRACE 1;V:  
STATE 1

### :CURSor:MARKer:CM<x>:ALL

機能 各マーカーカーソルのすべてのアイテムを ON/OFF します。

構文 :CURSor:MARKer:CM<x>ALL {<Boolean>}

<x> = 1 ~ 4

例 :CURSOR:MARKER:CM1:ALL ON

### :CURSor:MARKer:CM<x>:DT<x>?

機能 各マーカーカーソルの ΔT 測定に関するすべての設定値を問い合わせます。

構文 :CURSor:MARKer:CM<x>:DT<x>?

CM<x> の <x> = 1 ~ 4

DT<x> の <x> = 1 ~ 4

例 :CURSOR:MARKER:CM1:DT1? -> :CURSOR:  
MARKER:CM1:DT1:STATE 1

### :CURSor:MARKer:CM<x>:DT<x>:STATE

機能 各マーカーカーソルの ΔT 測定の ON/OFF を設定 / 問い合わせします。

構文 :CURSor:MARKer:CM<x>:DT<x>:STATE

{<Boolean>}

:CURSor:MARKer:CM<x>:DT<x>:STATE?

CM<x> の <x> = 1 ~ 4

DT<x> の <x> = 1 ~ 4

例 :CURSOR:MARKER:CM1:DT2:STATE ON  
:CURSOR:MARKER:CM1:DT2:STATE?  
-> :CURSOR:MARKER:CM1:DT2:STATE 1

### :CURSor:MARKer:CM<x>:DT<x>:VALue?

機能 各マーカーカーソルの ΔT 値を問い合わせます。

構文 :CURSor:MARKer:CM<x>:DT<x>:VALue?

CM<x> の <x> = 1 ~ 4

DT<x> の <x> = 1 ~ 4

例 :CURSOR:MARKER:CM1:DT2:VALUE?  
-> :CURSOR:MARKER:CM1:DT2:  
VALUE 0.000E+00

### :CURSor:MARKer:CM<x>:DV<x>?

機能 各マーカーカーソルの ΔV 測定に関するすべての設定値を問い合わせます。

構文 :CURSor:MARKer:CM<x>:DV<x>?

CM<x> の <x> = 1 ~ 4

DV<x> の <x> = 1 ~ 4

例 :CURSOR:MARKER:CM1:DV2? -> :CURSOR:  
MARKER:CM1:DV2:STATE 1

### :CURSor:MARKer:CM<x>:DV<x>:STATE

機能 各マーカーカーソルの ΔV 測定の ON/OFF を設定 / 問い合わせします。

構文 :CURSor:MARKer:CM<x>:DV<x>:STATE

{<Boolean>}

:CURSor:MARKer:CM<x>:DV<x>:STATE?

CM<x> の <x> = 1 ~ 4

DV<x> の <x> = 1 ~ 4

例 :CURSOR:MARKER:CM1:DV2:STATE ON  
:CURSOR:MARKER:CM1:DV2:STATE?  
-> :CURSOR:MARKER:CM1:DV2:STATE 1

### :CURSor:MARKer:CM<x>:DV<x>:VALue?

機能 各マーカーカーソルの ΔV 値を問い合わせます。

構文 :CURSor:MARKer:CM<x>:DV<x>:VALue?

CM<x> の <x> = 1 ~ 4

DV<x> の <x> = 1 ~ 4

例 :CURSOR:MARKER:CM1:DV2:VALUE?  
-> :CURSOR:MARKER:CM1:DV2:VALUE 0

### :CURSor:MARKer:CM<x>:JUMP

機能 各マーカーカーソルを指定したズーム波形の中央位置にジャンプします。

構文 :CURSor:MARKer:CM<x>:JUMP {Z1|Z2}

<x> = 1 ~ 4

例 :CURSOR:MARKER:CM1:JUMP Z1

### :CURSor:MARKer:CM<x>:POSition

機能 各マーカーカーソルの位置を設定 / 問い合わせします。

構文 :CURSor:MARKer:CM<x>:POSition {<NRF>}

:CURSor:MARKer:CM<x>:POSition?

<x> = 1 ~ 4

<NRF> = - 5 ~ 5(div)

例 :CURSOR:MARKER:CM1:POSITION 1  
:CURSOR:MARKER:CM1:POSITION?  
-> :CURSOR:MARKER:CM1:  
POSITION 1.000E+00

### :CURSor:MARKer:CM<x>:T?

機能 各マーカーカーソルの時間測定に関するすべての設定値を問い合わせます。

構文 :CURSor:MARKer:CM<x>:T?

<x> = 1 ~ 4

例 :CURSOR:MARKER:CM1:T? -> :CURSOR:  
MARKER:CM1:T:STATE 1

**:CURSor:MARKer:CM<x>:T:STATE**

機能 各マーカーカーソルの時間測定の ON/OFF を設定 / 問い合わせします。

構文 :CURSor:MARKer:CM<x>:T:STATE  
{<Boolean>}

:CURSor:MARKer:CM<x>:T:STATE?  
<x> = 1 ~ 4

例 :CURSOR:MARKER:CM1:T:STATE ON  
:CURSOR:MARKER:CM1:T:STATE?  
-> :CURSOR:MARKER:CM1:T:STATE 1

**:CURSor:MARKer:CM<x>:T:VALue?**

機能 各マーカーカーソルの時間値を問い合わせます。

構文 :CURSor:MARKer:CM<x>:T:VALue?  
<x> = 1 ~ 4

例 :CURSOR:MARKER:CM1:T:VALUE?  
-> :CURSOR:MARKER:CM1:T:  
VALUE 0.000E+00

**:CURSor:MARKer:CM<x>:TRACe**

機能 各マーカーカーソルの対象トレースを設定 / 問い合わせします。

構文 :CURSor:MARKer:CM<x>:TRACe  
{<Nrf>|OFF}

:CURSor:MARKer:CM<x>:TRACe?  
<x> = 1 ~ 4

<Nrf> = 1 ~ 8

例 :CURSOR:MARKER:CM1:TRACE 1  
:CURSOR:MARKER:CM1:TRACE? -> :CURSOR:  
MARKER:CM1:TRACE 1

**:CURSor:MARKer:CM<x>:V?**

機能 各マーカーカーソルの電圧測定に関するすべての設定値を問い合わせます。

構文 :CURSor:MARKer:CM<x>:V?  
<x> = 1 ~ 4

例 :CURSOR:MARKER:CM1:V? -> :CURSOR:  
MARKER:CM1:V:STATE 1

**:CURSor:MARKer:CM<x>:V:STATE**

機能 各マーカーカーソルの電圧測定の ON/OFF を設定 / 問い合わせします。

構文 :CURSor:MARKer:CM<x>:V:STATE  
{<Boolean>}

:CURSor:MARKer:CM<x>:V:STATE?  
<x> = 1 ~ 4

例 :CURSOR:MARKER:CM1:V:STATE ON  
:CURSOR:MARKER:CM1:V:STATE?  
-> :CURSor:MARKER:CM1:V:STATE 1

**:CURSor:MARKer:CM<x>:V:VALue?**

機能 各マーカーカーソルの電圧値を問い合わせします。

構文 :CURSor:MARKer:CM<x>:V:VALue?  
<x> = 1 ~ 4

例 :CURSOR:MARKER:CM1:V:VALUE?  
-> :CURSOR:MARKER:CM1:V:  
VALUE 0.000E+00

**:CURSor:MARKer:FORM**

機能 マーカーカーソルのフォームを設定 / 問い合わせします。

構文 :CURSor:MARKer:FORM {LINE|MARK}  
:CURSor:MARKer:FORM?

例 :CURSOR:MARKER:FORM LINE  
:CURSOR:MARKER:FORM? -> :CURSOR:  
MARKER:FORM LINE

**:CURSor:SERial?**

機能 シリアルカーソルに関するすべての設定値を問い合わせます。

構文 :CURSor:SERial?

例 :CURSOR:SERIAL? -> :CURSOR:SERIAL:  
SCURSOR1:ACTIVE HIGH;BCOUNT 8;  
BITRATE 10.00E+00;BITORDER MSBFIRST;  
FORMAT BINARY;HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;MODE 1;  
POSITION -4.000E+00;TRACE 1;  
TRACK OFF;:CURSOR:SERIAL:SCURSOR2:  
ACTIVE HIGH;BCOUNT 8;  
BITRATE 1.000E+00;BITORDER MSBFIRST;  
FORMAT BINARY;HYSTERESIS 3.000E+00;  
LEVEL 0.000E+00;MODE 0;  
POSITION -4.000E+00;TRACE 2;TRACK OFF

**:CURSor:SERial:SCURsor<x>?**

機能 各シリアルカーソルに関するすべての設定値を問い合わせます。

構文 :CURSor:SERial:SCURsor<x>?  
<x> = 1, 2

例 :CURSOR:SERIAL:SCURSOR1? -> :CURSOR:  
SERIAL:SCURSOR1:ACTIVE HIGH;  
BCOUNT 8;BITRATE 1.25E+06;  
BITORDER MSBFIRST;FORMAT BINARY;  
HYSTERESIS 300.00E-03;  
LEVEL 0.000E+00;MODE 1;  
POSITION -4.00E+00;TRACE 1;TRACK OFF

**:CURSor:SERial:SCURsor<x>:ACTive**

機能 各シリアルカーソルのアクティブレベルを設定 / 問い合わせします。

構文 :CURSor:SERial:SCURsor<x>:  
ACTive {HIGH|LOW}

:CURSor:SERial:SCURsor<x>:ACTive?  
<x> = 1, 2

例 :CURSOR:SERIAL:SCURSOR1:ACTIVE HIGH  
:CURSOR:SERIAL:SCURSOR1:ACTIVE?  
-> :CURSOR:SERIAL:SCURSOR1:  
ACTIVE HIGH

## 5.9 CURSor グループ

**:CURSor:SErIal:SCURsor<x>:BCOunt**  
 機能 各シリアルカーソルのビット長を設定 / 問い合わせします。

構文 :CURSor:SErIal:SCURsor<x>:  
 BCOunt {<NRf>}  
 :CURSor:SErIal:SCURsor<x>:BCOunt?  
 <x> = 1, 2  
 <NRf> = 1 ~ 128(bit)

例 :CURSOR:SERIAL:SCURSOR1:BCOUNT 8  
 :CURSOR:SERIAL:SCURSOR1:BCOUNT?  
 -> :CURSOR:SERIAL:SCURSOR1:BCOUNT 8

**:CURSor:SErIal:SCURsor<x>:BITRate**  
 機能 各シリアルカーソルのビットレートを設定 / 問い合わせします。

構文 :CURSor:SErIal:SCURsor<x>:BITRate  
 {<NRf>}  
 :CURSor:SErIal:SCURsor<x>:BITRate?  
 <x> = 1, 2  
 <NRf> = 1 ~ 1G(bps)

例 :CURSOR:SERIAL:SCURSOR1:BITRATE 10  
 :CURSOR:SERIAL:SCURSOR1:BITRATE?  
 -> :CURSOR:SERIAL:SCURSOR1:  
 BITRATE 10.00E+00

**:CURSor:SErIal:SCURsor<x>:BITorder**  
 機能 各シリアルカーソルのビットオーダーを設定 / 問い合わせします。

構文 :CURSor:SErIal:SCURsor<x>:BITorder  
 {LSBFirst|MSBFirst}  
 :CURSor:SErIal:SCURsor<x>:BITorder?  
 <x> = 1, 2

例 :CURSOR:SERIAL:SCURSOR1:  
 BITORDER LSBFIRST  
 :CURSOR:SERIAL:SCURSOR1:BITORDER?  
 -> :CURSOR:SERIAL:SCURSOR1:  
 BITORDER LSBFIRST

**:CURSor:SErIal:SCURsor<x>:FORMat**  
 機能 各シリアルカーソルの表示形式を設定 / 問い合わせします。

構文 :CURSor:SErIal:SCURsor<x>:FORMat  
 {BINary|HEXa}  
 :CURSor:SErIal:SCURsor<x>:FORMat?  
 <x> = 1, 2

例 :CURSOR:SERIAL:SCURSOR1:FORMAT HEXA  
 :CURSOR:SERIAL:SCURSOR1:FORMAT?  
 -> :CURSOR:SERIAL:SCURSOR1:  
 FORMAT HEXA

**:CURSor:SErIal:SCURsor<x>:HYSTeresis**  
 機能 各シリアルカーソルのヒステリシスを設定 / 問い合わせします。

構文 :CURSor:SErIal:SCURsor<x>:HYSTeresis  
 {<NRf>}  
 :CURSor:SErIal:SCURsor<x>:HYSTeresis?  
 <x> = 1, 2  
 <NRf> = 0 ~ 4(div)

例 :CURSOR:SERIAL:SCURSOR1:HYSTERESIS 1  
 :CURSOR:SERIAL:SCURSOR1:HYSTERESIS?  
 -> :CURSOR:SERIAL:SCURSOR1:  
 HYSTERESIS 1.000E+00

**:CURSor:SErIal:SCURsor<x>:JUMP**  
 機能 各シリアルカーソルを指定方向に移動します。

構文 :CURSor:SErIal:SCURsor<x>:JUMP  
 {BACK|FRONT}  
 <x> = 1, 2

例 :CURSOR:SERIAL:SCURSOR1:JUMP BACK

**:CURSor:SErIal:SCURsor<x>:LEVel**  
 機能 シリアルカーソルのしきい値 (Threshold) レベルを設定 / 問い合わせします。

構文 :CURSor:SErIal:SCURsor<x>:LEVel {<電  
 圧>|<電流>}  
 :CURSor:SErIal:SCURsor<x>:LEVel?  
 <x> = 1, 2  
 <電圧>、<電流> = 本体ユーザーズマニュアル  
 参照。

例 :CURSOR:SERIAL:SCURSOR1:LEVEL 0V  
 :CURSOR:SERIAL:SCURSOR1:LEVEL?  
 -> :CURSOR:SERIAL:SCURSOR1:  
 LEVEL 0.000E+00

**:CURSor:SErIal:SCURsor<x>:MODE**  
 機能 シリアルカーソルの ON/OFF を設定 / 問い合わせします。

構文 :CURSor:SErIal:SCURsor<x>:MODE  
 {<Boolean>}  
 :CURSor:SErIal:SCURsor<x>:MODE?  
 <x> = 1, 2

例 :CURSOR:SERIAL:SCURSOR1:MODE ON  
 :CURSOR:SERIAL:SCURSOR1:MODE?  
 -> :CURSOR:SERIAL:SCURSOR1:MODE 1

**:CURSor:SErIal:SCURsor<x>:POSition**  
 機能 シリアルカーソルの位置を設定 / 問い合わせします。

構文 :CURSor:SErIal:SCURsor<x>:POSition  
 {<NRf>}  
 :CURSor:SErIal:SCURsor<x>:POSition?  
 <x> = 1, 2  
 <NRf> = - 5 ~ 5(div)

例 :CURSOR:SERIAL:SCURSOR1:POSITION 1  
 :CURSOR:SERIAL:SCURSOR1:POSITION?  
 -> :CURSOR:SERIAL:SCURSOR1:  
 POSITION 1.000E+00

**:CURSor:SERIAL:SCURsor<x>:TRACE**

機能 各シリアルカーソルのトレースを設定 / 問い合わせします。

構文 :CURSor:SERIAL:SCURsor<x>:  
TRACE {<NRf>|A<y>|B<y>|C<y>|D<y>}  
:CURSor:SERIAL:SCURsor<x>:TRACE?  
<x> = 1, 2  
<NRf> = 1 ~ 8  
<y> = 0 ~ 7

例 :CURSOR:SERIAL:SCURSOR1:TRACE 1  
:CURSOR:SERIAL:SCURSOR1:TRACE?  
-> :CURSOR:SERIAL:SCURSOR1:TRACE 1

解説 DL9510L、DL9505L は {<NRf>|A<y>|C<y>} が有効です。

**:CURSor:SERIAL:SCURsor<x>:TRACK**

機能 各シリアルカーソルを指定したズーム波形上にジャンプします。

構文 :CURSor:SERIAL:SCURsor<x>:TRACK {OFF|Z1|Z2}  
:CURSor:SERIAL:SCURsor<x>:TRACK?  
<x> = 1, 2

例 :CURSOR:SERIAL:SCURSOR1:TRACK OFF  
:CURSOR:SERIAL:SCURSOR1:TRACK?  
-> :CURSOR:SERIAL:SCURSOR1:TRACK OFF

**:CURSor:SERIAL:SCURsor<x>:VALue?**

機能 各シリアルカーソルの測定値を問い合わせます。

構文 :CURSor:SERIAL:SCURsor<x>:VALue?  
<x> = 1, 2

例 :CURSOR:SERIAL:SCURSOR1:VALUE?  
-> :CURSOR:SERIAL:SCURSOR1:  
VALUE "11000111"

**:CURSor:TYPE**

機能 カーソルの種類を設定 / 問い合わせします。

構文 :CURSor:TYPE {HORizontal|HAVertical|MARKer|SERial|VERTical|VT}  
:CURSor:TYPE?

例 :CURSOR:TYPE HORIZONTAL  
:CURSOR:TYPE? -> :CURSOR:  
TYPE HORIZONTAL

**:CURSor:VERTical?**

機能 垂直カーソルに関するすべての設定値を問い合わせます。

構文 :CURSor:VERTical?  
例 :CURSOR:VERTICAL? -> :CURSOR:  
VERTICAL:BASIC:DT:STATE 1;:CURSOR:  
VERTICAL:BASIC:LINKAGE 1;PERDT:  
STATE 1;:CURSOR:VERTICAL:BASIC:T1:  
POSITION -4.000E+00;STATE 1;:CURSOR:  
VERTICAL:BASIC:T2:  
POSITION -4.000E+00;STATE 1;:CURSOR:  
VERTICAL:CALCULATION:DEFINE1 "T1";  
DEFINE2 "T2";DEFINE3 "T1";  
DEFINE4 "T2";STATE1 0;STATE2 0;  
STATE3 0;STATE4 0

**:CURSor:VERTical:BASic?**

機能 垂直カーソルの Basic アイテムに関するすべての設定値を問い合わせます。

構文 :CURSor:VERTical:BASic?  
例 :CURSOR:VERTICAL:BASIC? -> :CURSOR:  
VERTICAL:BASIC:DT:STATE 1;:  
CURSOR:VERTICAL:BASIC:LINKAGE 1;  
PERDT:STATE 1;:CURSOR:VERTICAL:BASIC:  
T1:POSITION -4.000E+00;STATE 1;:  
CURSOR:VERTICAL:BASIC:T2:  
POSITION -4.000E+00;STATE 1

**:CURSor:VERTical[:BASic]:ALL**

機能 垂直カーソルの Basic アイテムすべてを ON/OFF します。

構文 :CURSor:VERTical[:BASic]:  
ALL {<Boolean>}  
例 :CURSOR:VERTICAL:BASIC:ALL ON

**:CURSor:VERTical[:BASic]:DT?**

機能 垂直カーソルの  $\Delta T$  測定に関するすべての設定値を問い合わせます。

構文 :CURSor:VERTical[:BASic]:DT?  
例 :CURSOR:VERTICAL:BASIC:DT?  
-> :CURSOR:VERTICAL:BASIC:DT:STATE 1

**:CURSor:VERTical[:BASic]:DT:STATE**

機能 垂直カーソルの  $\Delta T$  測定の ON/OFF を設定 / 問い合わせします。

構文 :CURSor:VERTical[:BASic]:DT:STATE  
{<Boolean>}  
:CURSor:VERTical[:BASic]:DT:STATE?  
例 :CURSOR:VERTICAL:BASIC:DT:STATE ON  
:CURSOR:VERTICAL:BASIC:DT:STATE?  
-> :CURSOR:VERTICAL:BASIC:DT:STATE 1

**:CURSor:VERTical[:BASic]:DT:VALue?**

機能 垂直カーソルの  $\Delta T$  値を問い合わせます。

構文 :CURSor:VERTical[:BASic]:DT:VALue?  
例 :CURSOR:VERTICAL:BASIC:DT:VALUE?  
-> :CURSOR:VERTICAL:BASIC:DT:  
VALUE 1.000E+00

**:CURSor:VERTical[:BASic]:LINKage**

機能 垂直カーソルのリンクの ON/OFF を設定 / 問い合わせします。

構文 :CURSor:VERTical[:BASic]:LINKage  
{<Boolean>}  
:CURSor:VERTical[:BASic]:LINKage?  
例 :CURSOR:VERTICAL:BASIC:LINKAGE ON  
:CURSOR:VERTICAL:BASIC:LINKAGE?  
-> :CURSOR:VERTICAL:BASIC:LINKAGE 1



## 5.9 CURSOR グループ

### **:CURSOR:VERTICAL[:BASIC]:PERDt?**

機能 垂直カーソルの 1/ΔT 測定に関するすべての設定値を問い合わせます。

構文 :CURSOR:VERTICAL[:BASIC]:PERDt?

例 :CURSOR:VERTICAL:BASIC:PERDt?  
-> :CURSOR:VERTICAL:BASIC:PERDt:  
STATE 1

### **:CURSOR:VERTICAL[:BASIC]:PERDt:STATE**

機能 垂直カーソルの 1/ΔT 測定の ON/OFF を設定 / 問い合わせします。

構文 :CURSOR:VERTICAL[:BASIC]:PERDt:STATE  
{<Boolean>}

例 :CURSOR:VERTICAL[:BASIC]:PERDt:STATE?  
:CURSOR:VERTICAL:BASIC:PERDt:STATE?  
-> :CURSOR:VERTICAL:BASIC:PERDt:  
STATE 1

### **:CURSOR:VERTICAL[:BASIC]:PERDt:VALUE?**

機能 垂直カーソルの 1/ΔT 値を問い合わせます。

構文 :CURSOR:VERTICAL[:BASIC]:PERDt:VALUE?

例 :CURSOR:VERTICAL:BASIC:PERDt:VALUE?  
-> :CURSOR:VERTICAL:BASIC:PERDt:  
VALUE 0.000E+00

### **:CURSOR:VERTICAL[:BASIC]:T<x>?**

機能 各垂直カーソルに関するすべての設定値を問い合わせます。

構文 :CURSOR:VERTICAL[:BASIC]:T<x>?

例 <x> = 1, 2  
:CURSOR:VERTICAL:BASIC:T1?  
-> :CURSOR:VERTICAL:BASIC:T1:  
POSITION -4.000E+00;STATE 1

### **:CURSOR:VERTICAL[:BASIC]:T<x>:JUMP**

機能 各垂直カーソルを指定したズーム波形の中央位置にジャンプします。

構文 :CURSOR:VERTICAL[:BASIC]:T<x>:

JUMP {Z1|Z2}  
<x> = 1, 2  
例 :CURSOR:VERTICAL:BASIC:T1:JUMP Z1

### **:CURSOR:VERTICAL[:BASIC]:T<x>:POSITION**

機能 各垂直カーソルの位置を設定 / 問い合わせします。

構文 :CURSOR:VERTICAL[:BASIC]:T<x>:

POSITION {<NRf>}  
:CURSOR:VERTICAL[:BASIC]:T<x>:  
POSITION?  
<x> = 1, 2  
<NRf> = -5 ~ 5(div)  
例 :CURSOR:VERTICAL:BASIC:T1:POSITION 1  
:CURSOR:VERTICAL:BASIC:T1:POSITION?  
-> :CURSOR:VERTICAL:BASIC:T1:  
POSITION 1.000E+00

### **:CURSOR:VERTICAL[:BASIC]:T<x>:STATE**

機能 各垂直カーソルの ON/OFF を設定 / 問い合わせします。

構文 :CURSOR:VERTICAL[:BASIC]:T<x>:STATE

{<Boolean>}  
:CURSOR:VERTICAL[:BASIC]:T<x>:STATE?  
<x> = 1, 2

例 :CURSOR:VERTICAL:BASIC:T1:STATE ON  
:CURSOR:VERTICAL:BASIC:T1:STATE?  
-> :CURSOR:VERTICAL:BASIC:T1:STATE 1

### **:CURSOR:VERTICAL[:BASIC]:T<x>:VALUE?**

機能 各垂直カーソルの時間値を問い合わせます。

構文 :CURSOR:VERTICAL[:BASIC]:T<x>:VALUE?

例 <x> = 1, 2  
:CURSOR:VERTICAL:BASIC:T1:VALUE?  
-> :CURSOR:VERTICAL:BASIC:T1:  
VALUE 0.000E+00

### **:CURSOR:VERTICAL:CALCulation?**

機能 垂直カーソルの Calc アイテムに関するすべての設定値を問い合わせます。

構文 :CURSOR:VERTICAL:CALCulation?

例 :CURSOR:VERTICAL:CALCulation?  
-> :CURSOR:VERTICAL:CALCulation:  
DEFINE1 "T1";DEFINE2 "T2";  
DEFINE3 "T1";DEFINE4 "T2";STATE1 0;  
STATE2 0;STATE3 0;STATE4 0

### **:CURSOR:VERTICAL:CALCulation:ALL**

機能 垂直カーソルの Calc アイテムすべてを ON/OFF します。

構文 :CURSOR:VERTICAL:CALCulation:ALL  
{<Boolean>}

例 :CURSOR:VERTICAL:CALCulation:ALL ON

### **:CURSOR:VERTICAL:CALCulation:DEFine<x>**

機能 垂直カーソルの Calc アイテムの演算式を設定 / 問い合わせします。

構文 :CURSOR:VERTICAL:CALCulation:

DEFine<x> {<文字列>}  
:CURSOR:VERTICAL:CALCulation:  
DEFine<x>?  
<x> = 1 ~ 4  
<文字列> = 128 文字以内  
例 :CURSOR:VERTICAL:CALCulation:  
DEFINE1 "T1"  
:CURSOR:VERTICAL:CALCulation:DEFine1?  
-> :CURSOR:VERTICAL:CALCulation:  
DEFINE1 "T1"

**:CURSor:VERTical:CALCulation:****STATE<x>**

機能 垂直カーソルの Calc アイテムの ON/OFF を設定 / 問い合わせします。

構文 :CURSor:VERTical:CALCulation:STATE<x>  
{<Boolean>}  
:CURSor:VERTical:CALCulation:  
STATE<x>?  
<x> = 1 ~ 4

例 :CURSOR:VERTICAL:CALCULATION:  
STATE1 ON  
:CURSOR:VERTICAL:CALCULATION:STATE1?  
-> :CURSOR:VERTICAL:CALCULATION:  
STATE1 1

**:CURSor:VERTical:CALCulation:****VALue<x>?**

機能 垂直カーソルの Calc アイテムの測定値を問い合わせます。

構文 :CURSor:VERTical:CALCulation:  
VALue<x>?  
<x> = 1 ~ 4

例 :CURSOR:VERTICAL:CALCULATION:VALUE1?  
-> :CURSOR:VERTICAL:CALCULATION:  
VALUE1 0.000E+00

**:CURSor:VT?**

機能 VT カーソルに関するすべての設定値を問い合わせます。

構文 :CURSOR:VT?

例 :CURSOR:VT? -> :CURSOR:VT:BaSiC:  
GRoUp1:BITORDER MSBFIRST;FORMAT HEXA;  
STATE 1;:CURSOR:VT:BaSiC:GRoUp2:  
BITORDER MSBFIRST;FORMAT HEXA;  
STATE 1;:CURSOR:VT:BaSiC:GRoUp3:  
BITORDER MSBFIRST;FORMAT HEXA;  
STATE 1;:CURSOR:VT:BaSiC:GRoUp4:  
BITORDER MSBFIRST;FORMAT HEXA;  
STATE 1;:CURSOR:VT:BaSiC:GRoUp5:  
BITORDER MSBFIRST;FORMAT HEXA;  
STATE 1;:CURSOR:VT:BaSiC:  
POSITION 0.000E+00;T:STATE 1;:CURSOR:  
VT:BaSiC:V1:STATE 1;:CURSOR:VT:BaSiC:  
V2:STATE 1;:CURSOR:VT:BaSiC:V3:  
STATE 1;:CURSOR:VT:BaSiC:V4:  
STATE 1;:CURSOR:VT:BaSiC:V5:STATE 1;:  
CURSOR:VT:BaSiC:V6:STATE 1;:CURSOR:  
VT:BaSiC:V7:STATE 1;:CURSOR:VT:BaSiC:  
V8:STATE 1;:CURSOR:VT:CaLcULation:  
DEFINe1 "T1";DEFINe2 "V(C1)";  
DEFINe3 "V(C2)";DEFINe4 "V(C3)";  
STATE1 0;STATE2 0;STATE3 0;STATE4 0

**:CURSor:VT:BaSiC?**

機能 VT カーソルの Basic アイテムに関するすべての設定値を問い合わせます。

構文 :CURSOR:VT:BaSiC?

例 :CURSOR:VT:BaSiC? -> :CURSOR:VT:  
BaSiC:GRoUp1:BITORDER MSBFIRST;  
FORMAT HEXA;STATE 1;:CURSOR:VT:BaSiC:  
GRoUp2:BITORDER MSBFIRST;FORMAT HEXA;  
STATE 1;:CURSOR:VT:BaSiC:GRoUp3:  
BITORDER MSBFIRST;FORMAT HEXA;  
STATE 1;:CURSOR:VT:BaSiC:GRoUp4:  
BITORDER MSBFIRST;FORMAT HEXA;  
STATE 1;:CURSOR:VT:BaSiC:GRoUp5:  
BITORDER MSBFIRST;FORMAT HEXA;  
STATE 1;:CURSOR:VT:BaSiC:  
POSITION 0.000E+00;T:STATE 1;:CURSOR:  
VT:BaSiC:V1:STATE 1;:CURSOR:VT:BaSiC:  
V2:STATE 1;:CURSOR:VT:BaSiC:V3:  
STATE 1;:CURSOR:VT:BaSiC:V4:STATE 1;:  
CURSOR:VT:BaSiC:V5:STATE 1;:CURSOR:  
VT:BaSiC:V6:STATE 1;:CURSOR:VT:  
BaSiC:V7:STATE 1;:CURSOR:VT:BaSiC:V8:  
STATE 1

**:CURSor:VT[:BaSiC]:ALL**

機能 VT カーソルの Basic アイテムすべてを ON/OFF します。

構文 :CURSOR:VT[:BaSiC]:ALL {<Boolean>}

例 :CURSOR:VT:BaSiC:ALL ON

**:CURSor:VT[:BaSiC]:GRoUp<x>?**

機能 VT カーソルの各ロジックグループに関するすべての設定値を問い合わせます。

構文 :CURSOR:VT[:BaSiC]:GRoUp<x>?

<x> = 1 ~ 5

例 :CURSOR:VT:BaSiC:GRoUp1? -> :CURSOR:  
VT:BaSiC:GRoUp1:BITORDER MSBFIRST;  
FORMAT HEXA;STATE 1

**:CURSor:VT[:BaSiC]:GRoUp<x>:BITorder**

機能 VT カーソルの各ロジックグループの測定ビットオーダを設定 / 問い合わせします。

構文 :CURSOR:VT[:BaSiC]:GRoUp<x>:  
BITorder {LSBFIRST|MSBFIRST}

:CURSOR:VT[:BaSiC]:GRoUp<x>:BITorder?  
<x> = 1 ~ 5

例 :CURSOR:VT:BaSiC:GRoUp1:  
BITORDER LSBFIRST  
:CURSOR:VT:BaSiC:GRoUp1:BITORDER?  
-> :CURSOR:VT:BaSiC:GRoUp1:  
BITORDER LSBFIRST

## 5.9 CURSOR グループ

**:CURSOR:VT[:BASic]:GROup<x>:FORMat**  
機能 VT カーソルの各ロジックグループの測定値の表示形式を設定 / 問い合わせします。  
構文 :CURSOR:VT[:BASic]:GROup<x>:  
FORMat {BINary|HEXa}  
:CURSOR:VT[:BASic]:GROup<x>:FORMat?  
<x> = 1 ~ 5  
例 :CURSOR:VT:BASIC:GROUp1:FORMat HEXA  
:CURSOR:VT:BASIC:GROUp1:FORMat?  
-> :CURSOR:VT:BASIC:GROUp1:  
FORMat HEXA

**:CURSOR:VT[:BASic]:GROup<x>:STATe**  
機能 VT カーソルの各ロジックグループの測定値の ON/OFF を設定 / 問い合わせします。  
構文 :CURSOR:VT[:BASic]:GROup<x>:  
STATe {<Boolean>}  
:CURSOR:VT[:BASic]:GROup<x>:STATe?  
<x> = 1 ~ 5  
例 :CURSOR:VT:BASIC:GROUp1:STATe ON  
:CURSOR:VT:BASIC:GROUp1:STATe?  
-> :CURSOR:VT:BASIC:GROUp1:STATe 1

**:CURSOR:VT[:BASic]:GROup<x>:VALue?**  
機能 VT カーソルの各ロジックグループの測定値を問い合わせます。  
構文 :CURSOR:VT[:BASic]:GROup<x>:VALue?  
<x> = 1 ~ 5  
例 :CURSOR:VT:BASIC:GROUp1:VALue?  
-> :CURSOR:VT:BASIC:GROUp1:  
VALue 4294967295

**:CURSOR:VT[:BASic]:POSITion**  
機能 VT カーソルの位置を設定 / 問い合わせします。  
構文 :CURSOR:VT[:BASic]:POSITion {<NRf>}  
:CURSOR:VT[:BASic]:POSITion?  
<NRf> = - 5 ~ 5(div)  
例 :CURSOR:VT:BASIC:POSITion 1  
:CURSOR:VT:BASIC:POSITion?  
-> :CURSOR:VT:BASIC:  
POSITion 1.000E+00

**:CURSOR:VT[:BASic]:T?**  
機能 VT カーソルの時間測定に関するすべての設定値を問い合わせします。  
構文 :CURSOR:VT[:BASic]:T?  
例 :CURSOR:VT:BASIC:T? -> :CURSOR:VT:  
BASIC:T:STATe 1

**:CURSOR:VT[:BASic]:T:STATe**  
機能 VT カーソルの時間測定 ON/OFF を設定 / 問い合わせします。  
構文 :CURSOR:VT[:BASic]:T:STATe  
{<Boolean>}  
:CURSOR:VT[:BASic]:T:STATe?  
例 :CURSOR:VT:BASIC:T:STATe ON  
:CURSOR:VT:BASIC:T:STATe? -> :CURSOR:  
VT:BASIC:T:STATe 1

**:CURSOR:VT[:BASic]:T:VALue?**  
機能 VT カーソルの時間値を問い合わせします。  
構文 :CURSOR:VT[:BASic]:T:VALue?  
例 :CURSOR:VT:BASIC:T:VALue? -> :CURSOR:  
VT:BASIC:T:VALue 0.000E+00

**:CURSOR:VT[:BASic]:V<x>?**  
機能 VT カーソルの電圧測定に関するすべての設定値を問い合わせします。  
構文 :CURSOR:VT[:BASic]:V<x>?  
<x> = 1 ~ 8  
例 :CURSOR:VT:BASIC:V1? -> :CURSOR:VT:  
BASIC:V1:STATe 1

**:CURSOR:VT[:BASic]:V<x>:STATe**  
機能 VT カーソルの電圧値測定の ON/OFF を設定 / 問い合わせします。  
構文 :CURSOR:VT[:BASic]:V<x>:STATe  
{<Boolean>}  
:CURSOR:VT[:BASic]:V<x>:STATe?  
<x> = 1 ~ 8  
例 :CURSOR:VT:BASIC:V1:STATe ON  
:CURSOR:VT:BASIC:V1:STATe?  
-> :CURSOR:VT:BASIC:V1:STATe 1

**:CURSOR:VT[:BASic]:V<x>:VALue?**  
機能 VT カーソルの電圧値を問い合わせします。  
構文 :CURSOR:VT[:BASic]:V<x>:VALue?  
<x> = 1 ~ 8  
例 :CURSOR:VT:BASIC:V1:VALue?  
-> :CURSOR:VT:BASIC:V1:  
VALue 0.000E+00

**:CURSOR:VT:CALCulation?**  
機能 VT カーソルの Calc アイテムに関するすべての設定値を問い合わせます。  
構文 :CURSOR:VT:CALCulation?  
例 :CURSOR:VT:CALCulation? -> :CURSOR:  
VT:CALCulation:DEFIne1 "T1";  
DEFIne2 "V(C1)";DEFIne3 "V(C2)";  
DEFIne4 "V(C3)";STATe1 0;STATe2 0;  
STATe3 0;STATe4 0

**:CURSOR:VT:CALCulation:ALL**  
機能 VT カーソルの Calc アイテムすべてを ON/OFF します。  
構文 :CURSOR:VT:CALCulation:  
ALL {<Boolean>}  
例 :CURSOR:VT:CALCulation:ALL ON

**:CURSOR:VT:CALCULATION:DEFINE<x>**

機能 VT カーソルの Calc アイテムの演算式を設定 / 問い合わせします。

構文 :CURSOR:VT:CALCULATION:DEFINE<x> {<文字列>}

:CURSOR:VT:CALCULATION:DEFINE<x>?

<x> = 1 ~ 4

<文字列> = 128 文字以内

例 :CURSOR:VT:CALCULATION:DEFINE1 "T1"

:CURSOR:VT:CALCULATION:DEFINE1?

-> :CURSOR:VT:CALCULATION:

DEFINE1 "T1"

**:CURSOR:VT:CALCULATION:STATE<x>**

機能 VT カーソルの Calc アイテムの ON/OFF を設定 / 問い合わせします。

構文 :CURSOR:VT:CALCULATION:STATE<x>

{<Boolean>}

:CURSOR:VT:CALCULATION:STATE<x>?

<x> = 1 ~ 4

例 :CURSOR:VT:CALCULATION:STATE1 ON

:CURSOR:VT:CALCULATION:STATE1?

-> :CURSOR:VT:CALCULATION:STATE1 0

**:CURSOR:VT:CALCULATION:VALUE<x>?**

機能 VT カーソルの Calc アイテムの測定値を問い合わせします。

構文 :CURSOR:VT:CALCULATION:VALUE<x>?

<x> = 1 ~ 4

例 :CURSOR:VT:CALCULATION:VALUE1?

-> :CURSOR:VT:CALCULATION:VALUE1

0.000E+00

**:CURSOR:VT:JUMP**

機能 VT カーソルを指定したズーム波形の中央位置にジャンプします。

構文 :CURSOR:VT:JUMP {Z1|Z2}

例 :CURSOR:VT:JUMP Z1

## 5.10 DISPlay グループ

### :DISPlay?

機能 表示に関するすべての設定値を問い合わせます。

構文 :DISPlay?

例 :DISPLAY? -> :DISPLAY:ACCUMULATE:  
GRADE INTENSITY;MODE 0;PERSISTENCE:  
COUNT INFINITY;MODE COUNT;  
TIME 1.000E+00;:DISPLAY:BLIGHT:  
AUTOOFF 0;BRIGHTNESS 1;LCD 1;  
TIMEOUT 1;:DISPLAY:COLOR:  
CHANNEL1 BLUE;CHANNEL2 BGREEN;  
CHANNEL3 CYAN;CHANNEL4 DBLUE;  
LOGIC YELLOW;LSTATE GREEN;MATH1 GRAY;  
MATH2 GREEN;MATH3 LBLUE;MATH4 LGREEN;  
REFERENCE1 MAGENTA;REFERENCE2 MGREEN;  
REFERENCE3 ORANGE;REFERENCE4 PINK;:  
DISPLAY:FORMAT SINGLE;GRATICULE GRID;  
INTENSITY:WAVEFORM 10;CURSOR 10;  
GRID 10;MARKER 10;ZBOX 10;:DISPLAY:  
INTERPOLATE SINE;MAPPING:MODE AUTO;  
TRACE1 1;TRACE2 1;TRACE3 1;TRACE4 1;  
TRACE5 1;TRACE6 1;TRACE7 1;TRACE8 1;:  
DISPLAY:TRANSLUCENT 1

### :DISPlay:ACCumulate?

機能 アキュムレート (波形の重ね書き表示)に関するすべての設定値を問い合わせます。

構文 :DISPlay:ACCumulate?

例 :DISPLAY:ACCUMULATE? -> :DISPLAY:  
ACCUMULATE:GRADE INTENSITY;  
MODE 0;PERSISTENCE:COUNT INFINITY;  
MODE COUNT;TIME 1.000E+00

### :DISPlay:ACCumulate:GRADE

機能 アキュムレートモードを設定 / 問い合わせします。

構文 :DISPlay:ACCumulate:GRADE {COLor|  
INTensity}

例 :DISPlay:ACCumulate:GRADE?  
:DISPLAY:ACCUMULATE:GRADE?  
-> :DISPLAY:ACCUMULATE:  
GRADE INTENSITY

### :DISPlay:ACCumulate:MODE

機能 アキュムレートモードの ON/OFF を設定 / 問い合わせします。

構文 :DISPlay:ACCumulate:MODE {<Boolean>}  
:DISPlay:ACCumulate:MODE?

例 :DISPlay:ACCumulate:MODE ON  
:DISPlay:ACCumulate:MODE?  
-> :DISPlay:ACCumulate:MODE 1

### :DISPlay:ACCumulate:PERSistence?

機能 パーシスタンスに関するすべての設定値を問い合わせます。

構文 :DISPlay:ACCumulate:PERSistence?

例 :DISPlay:ACCumulate:PERSistence?  
-> :DISPlay:ACCumulate:PERSistence:  
COUNT INFINITY;MODE COUNT;  
TIME 1.000E+00

### :DISPlay:ACCumulate:PERSistence: COUNT

機能 パーシスタンスの回数を設定 / 問い合わせします。

構文 :DISPlay:ACCumulate:PERSistence:COUNT  
{<Nrf>|INfinity}

例 :DISPlay:ACCumulate:PERSistence:  
COUNT?  
<Nrf> = 1 ~ (現在のレコード長での最大ヒスト  
リー枚数)

例 :DISPlay:ACCumulate:PERSistence:  
COUNT INFINITY  
:DISPlay:ACCumulate:PERSistence:  
COUNT? -> :DISPlay:ACCumulate:  
PERSistence:COUNT INFINITY

### :DISPlay:ACCumulate:PERSistence:MODE

機能 パーシスタンスのモードを設定 / 問い合わせします。

構文 :DISPlay:ACCumulate:PERSistence:MODE  
{COUNT|TIME}

例 :DISPlay:ACCumulate:PERSistence:MODE?  
:DISPlay:ACCumulate:PERSistence:  
MODE COUNT  
:DISPlay:ACCumulate:PERSistence:MODE?  
-> :DISPlay:ACCumulate:PERSistence:  
MODE COUNT

### :DISPlay:ACCumulate:PERSistence:TIME

機能 パーシスタンスの時間を設定 / 問い合わせします。

構文 :DISPlay:ACCumulate:PERSistence:  
TIME {<時間>|INfinity}

例 :DISPlay:ACCumulate:PERSistence:TIME?  
<時間> = 100ms ~ 1s(100ms ステップ),1s ~  
10s(200ms ステップ),10s ~ 100s(2s ステップ)

例 :DISPlay:ACCumulate:PERSistence:  
TIME 1S  
:DISPlay:ACCumulate:PERSistence:TIME?  
-> :DISPlay:ACCumulate:PERSistence:  
TIME 1.000E+00

### :DISPlay:BLIGHT?

機能 バックライトに関するすべての設定値を問い合わせます。

構文 :DISPlay:BLIGHT?

例 :DISPlay:BLIGHT? -> :DISPlay:BLIGHT:  
AUTOOFF 0;BRIGHTNESS 1;  
LCD 1;TIMEOUT 1

**:DISPLAY:BLIGHT:AUTOOFF**

機能 バックライトを自動的に消すかどうかを設定 / 問い合わせします。

構文 :DISPLAY:BLIGHT:AUTOOFF {<Boolean>}  
:DISPLAY:BLIGHT:AUTOOFF?

例 :DISPLAY:BLIGHT:AUTOOFF ON  
:DISPLAY:BLIGHT:AUTOOFF? -> :DISPLAY:  
BLIGHT:AUTOOFF 1

**:DISPLAY:BLIGHT:BRIGHTNESS**

機能 バックライトの輝度を設定 / 問い合わせします。

構文 :DISPLAY:BLIGHT:BRIGHTNESS {<Nrf>}  
:DISPLAY:BLIGHT:BRIGHTNESS?  
<Nrf> = 1 ~ 8

例 :DISPLAY:BLIGHT:BRIGHTNESS 1  
:DISPLAY:BLIGHT:BRIGHTNESS?  
-> :DISPLAY:BLIGHT:BRIGHTNESS 1

**:DISPLAY:BLIGHT:LCD**

機能 バックライトの ON/OFF を設定 / 問い合わせします。

構文 :DISPLAY:BLIGHT:LCD {<Boolean>}  
:DISPLAY:BLIGHT:LCD?

例 :DISPLAY:BLIGHT:LCD ON  
:DISPLAY:BLIGHT:LCD? -> :DISPLAY:  
BLIGHT:LCD 1

**:DISPLAY:BLIGHT:TIMEOUT**

機能 バックライトのタイムアウトを設定 / 問い合わせします。

構文 :DISPLAY:BLIGHT:TIMEOUT {<Nrf>}  
:DISPLAY:BLIGHT:TIMEOUT?  
<Nrf> = 1 ~ 60(分)

例 :DISPLAY:BLIGHT:TIMEOUT 60  
:DISPLAY:BLIGHT:TIMEOUT? -> :DISPLAY:  
BLIGHT:TIMEOUT 60

**:DISPLAY:COLOR?**

機能 波形の表示色に関するすべての設定値を問い合わせます。

構文 :DISPLAY:COLOR?

例 :DISPLAY:COLOR? -> :DISPLAY:COLOR:  
CHANNEL1 BLUE;CHANNEL2 BGREEN;  
CHANNEL3 CYAN;CHANNEL4 DBLUE;  
LOGIC YELLOW;LSTATE GREEN;MATH1 GRAY;  
MATH2 GREEN;MATH3 LBLUE;MATH4 LGREEN;  
REFERENCE1 MAGENTA;REFERENCE2 MGREEN;  
REFERENCE3 ORANGE;REFERENCE4 PINK

**:DISPLAY:COLOR:{CHANNEL<x>|LOGIC|LSTATE|MATH<x>|REFERENCE<x>}**

機能 各々の波形の色を設定 / 問い合わせします。

構文 :DISPLAY:COLOR:{CHANNEL<x>|LOGIC|  
LSTATE|MATH<x>|REFERENCE<x>} {BLUE|  
BGREEN|CYAN|DBLUE|GRAY|GREEN|LBLUE|  
LGREEN|MAGENTA|MGREEN|ORANGE|PINK|  
PURPLE|RED|SPINK|YELLOW}  
:DISPLAY:COLOR:{CHANNEL<x>|LOGIC|  
LSTATE|MATH<x>|REFERENCE<x>}?

CHANNEL<x> の <x> = 1 ~ 4

MATH<x> の <x> = 1 ~ 4

REFERENCE<x> の <x> = 1 ~ 4

例 :DISPLAY:COLOR:CHANNEL1 BLUE  
:DISPLAY:COLOR:CHANNEL1? -> :DISPLAY:  
COLOR:CHANNEL1 BLUE

**:DISPLAY:FORMAT**

機能 表示フォーマットを設定 / 問い合わせします。

構文 :DISPLAY:FORMAT {DUAL|QUAD|SINGLE|  
TRIAD}  
:DISPLAY:FORMAT?

例 :DISPLAY:FORMAT SINGLE  
:DISPLAY:FORMAT? -> :DISPLAY:  
FORMAT SINGLE

**:DISPLAY:GRATICULE**

機能 グラティクル(目盛り)を設定 / 問い合わせします。

構文 :DISPLAY:GRATICULE {CROSSHAIR|FRAME|  
GRID|LINE}  
:DISPLAY:GRATICULE?

例 :DISPLAY:GRATICULE CROSSHAIR  
:DISPLAY:GRATICULE? -> :DISPLAY:  
GRATICULE CROSSHAIR

**:DISPLAY:INTENSITY?**

機能 表示アイテムの輝度に関するすべての設定値を問い合わせます。

構文 :DISPLAY:INTENSITY?  
:DISPLAY:INTENSITY? ->

例 :DISPLAY:INTENSITY? -> :DISPLAY:  
INTENSITY:WAVEFORM 10;  
CURSOR 10;GRID 10;MARKER 10

**:DISPLAY:INTENSITY[:WAVEFORM]**

機能 波形の輝度を設定 / 問い合わせします。

構文 :DISPLAY:INTENSITY[:WAVEFORM] {<Nrf>}  
:DISPLAY:INTENSITY[:WAVEFORM]?

<Nrf> = 1 ~ 20

例 :DISPLAY:INTENSITY:WAVEFORM 10  
:DISPLAY:INTENSITY:WAVEFORM?  
-> :DISPLAY:INTENSITY:WAVEFORM 10

## 5.10 DISPlay グループ

### **:DISPlay:INTENsity:{CURSor|GRID|MARKer|ZBOX}**

機能 各表示アイテムの輝度を設定 / 問い合わせします。

構文 :DISPlay:INTENsity:{CURSor|GRID|MARKer|ZBOX} {<NRf>}  
:DISPlay:INTENsity:{CURSor|GRID|MARKer|ZBOX}?  
<NRf> = 0 ~ 31

例 :DISPLAY:INTENSITY:CURSOR 10  
:DISPLAY:INTENSITY:CURSOR?  
-> :DISPLAY:INTENSITY:CURSOR 10

### **:DISPlay:INTERpolate**

機能 表示補間方式を設定 / 問い合わせします。

構文 :DISPlay:INTERpolate {LINE|OFF|PULSe|SINE}  
:DISPlay:INTERpolate?

例 :DISPLAY:INTERPOLATE SINE  
:DISPLAY:INTERPOLATE? -> :DISPLAY:  
INTERPOLATE SINE

### **:DISPlay:MAPPING?**

機能 分割画面への波形の割り当てに関するすべての設定値を問い合わせます。

構文 :DISPlay:MAPPING?

例 :DISPLAY:MAPPING? -> :DISPLAY:  
MAPPING:MODE AUTO;TRACE1 1;  
TRACE2 1;TRACE3 1;TRACE4 1;TRACE5 1;  
TRACE6 1;TRACE7 1;TRACE8 1

### **:DISPlay:MAPPING[:MODE]**

機能 分割フォーマットへの波形の割り当てのモードを設定 / 問い合わせします。

構文 :DISPlay:MAPPING[:MODE] {AUTO|MANual}  
:DISPlay:MAPPING[:MODE]?

例 :DISPLAY:MAPPING:MODE AUTO  
:DISPLAY:MAPPING:MODE? -> :DISPLAY:  
MAPPING:MODE AUTO

### **:DISPlay:MAPPING:TRACe<x>**

機能 分割フォーマットへの各波形の割り当てを設定 / 問い合わせします。

構文 :DISPlay:MAPPING:TRACe<x> {<NRf>}  
:DISPlay:MAPPING:TRACe<x>?  
<x> = 1 ~ 8  
<NRf> = 1 ~ 4

例 :DISPLAY:MAPPING:TRACE1 3  
:DISPLAY:MAPPING:TRACE1? -> :DISPLAY:  
MAPPING:TRACE1 3

### **:DISPlay:TRANslucent**

機能 半透明モードの ON/OFF を設定 / 問い合わせします。

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

例 :DISPLAY:TRANSLUCENT ON  
:DISPLAY:TRANSLUCENT? -> :DISPLAY:  
TRANSLUCENT 1

## 5.11 FILE グループ

### :FILE?

機能 指定したストレージメディア、内部メモリに関するすべての設定値を問い合わせます。

構文 :FILE?

例 :FILE? -> :FILE:DIRECTORY:  
CDIRECTORY "\FlashMem\HOME";:FILE:  
INTERNAL:STORE:BINARy:TRACE 1;:FILE:  
INTERNAL:STORE:DMEMORY:TRACE 1;:FILE:  
INTERNAL:STORE:SETUP:  
COMMENT1 "WAVE1";COMMENT2 "";  
COMMENT3 "";COMMENT4 "";COMMENT5 "";  
COMMENT6 "";COMMENT7 "";COMMENT8 "";  
COMMENT9 "";COMMENT10 "";  
COMMENT11 "";COMMENT12 "";LOCK1 1;  
LOCK2 0;LOCK3 0;LOCK4 0;LOCK5 0;  
LOCK6 0;LOCK7 0;LOCK8 0;LOCK9 0;  
LOCK10 0;LOCK11 0;LOCK12 0;:FILE:  
LOAD:BINARy:REFERENCE 1;:FILE:LOAD:  
ZPOLYGON:  
ZONE 1;:FILE:LOAD:ZWAVE:ZONE1;:FILE:  
SAVE:AHISTOGRAM:ANALYSIS 1;:FILE:  
SAVE:ANAMING DATE;ASCIi:  
COMPRESSION DECIMATION;HISTORY ONE;  
LENGTH 2500;RANGE MAIN;TRACE 1;:FILE:  
SAVE:BINARy:COMPRESSION DECIMATION;  
HISTORY ONE;LENGTH 2500;TRACE 1;:  
FILE:SAVE:COMMENT "THIS IS TEST";  
DMEMORY:TRACE 1;:FILE:SAVE:FFT:  
ANALYSIS 1;:FILE:SAVE:FLOAT:  
COMPRESSION DECIMATION;HISTORY ONE;  
LENGTH 2500;TRACE 1;:FILE:SAVE:  
NAME "DL9000";SBUS:ANALYSIS 1;:  
FILE:SAVE:ZWAVE:ZONE 1

### :FILE:DIReCtory?

機能 指定したストレージメディアに関するすべての設定値を問い合わせます。

構文 :FILE:DIReCtory?

例 :FILE:DIReCtory? -> :FILE:DIReCtory:  
CDIRECTORY "\Flash Mem\HOME"

### :FILE[:DIReCtory]:CDIReCtory (Change Directory)

機能 カレントディレクトリを設定/問い合わせします。

構文 :FILE[:DIReCtory]:

CDIReCtory {<文字列>}

:FILE[:DIReCtory]:CDIReCtory?

<文字列> = 259文字以内

例 :FILE:DIReCtory:

CDIRECTORY "\Flash Mem\DIR1\DIR2"

(絶対パス指定)

:FILE:DIReCtory:CDIRECTORY "DIR2"

(相対パス指定)

:FILE:DIReCtory:CDIRECTORY "\"

(ルートディレクトリ指定)

:FILE:DIReCtory:CDIRECTORY? -> :FILE:  
DIReCtory:

CDIRECTORY "\Flash Mem\DIR1\DIR2"

解説 ・ルートディレクトリ直下は、ドライブ指定ディレクトリです。

選択できるドライブは、以下の5つです。

- ・内蔵ハードディスク: "HD"
- ・内部メモリ: "Flash Mem"
- ・ネットワークドライブ: "Network"
- ・PCカード: "Storage Card<x>"
- ・USBストレージ: "USB Storage<x>"  
<x> = 1~4(但し、サポート範囲は "Storage Card" と "USB Storage" を合計して4つまでです(パーティション含む))
- ・指定したディレクトリをセーブ、ロードのカレントディレクトリに設定します。
- ・絶対パス、相対パス指定が可能です。
- ・絶対パスで指定する場合は先頭に "\" を付けます。
- ・上位の階層への相対パス指定はできません。

### :FILE[:DIReCtory]:FREE?

機能 指定したストレージメディアの空き容量をバイト数で問い合わせます。

構文 :FILE[:DIReCtory]:FREE?

例 :FILE:DIReCtory:FReE? -> :FILE:  
DIReCtory:FReE 65536

解説 ・カレントディレクトリの容量を返します。

- ・ルートディレクトリが、カレントディレクトリのときは "0" を返します。



## 5.11 FILE グループ

### **:FILE[:DIReCTory]:MDIRectory (Make Directory)**

機能 指定したディレクトリの下にディレクトリを作成します。オーバーラップコマンドです。

構文 :FILE[:DIReCTory]:  
MDIRectory {<文字列>}  
<文字列> = 259 文字以内

例 :FILE:DIReCTory:MDIRectory "\\Flash  
Mem\DIR1\DIR2"  
(絶対パス指定)  
:FILE:DIReCTory:MDIRectory "DIR2"  
(相対パス指定)

解説 <文字列> に関するの詳細は、「FILE[:DIReCTory]:CDIRectory」の解説を参照してください。

### **:FILE:INTErnal?**

機能 内部メモリに関するすべての設定値を問い合わせます。

構文 :FILE:INTErnal?

例 :FILE:INTErnal? -> :FILE:INTErnal:  
STORE:BINARy:TRAcE 1;:FILE:INTErnal:  
STORE:DMEMORy:TRAcE 1;:FILE:INTErnal:  
STORE:SETUP:COMMENT1 "WAVE1";  
COMMENT2 "";COMMENT3 "";COMMENT4 "";  
COMMENT5 "";COMMENT6 "";COMMENT7 "";  
COMMENT8 "";COMMENT9 "";COMMENT10 "";  
COMMENT11 "";COMMENT12 "";LOCK1 1;  
LOCK2 0;LOCK3 0;LOCK4 0;LOCK5 0;  
LOCK6 0;LOCK7 0;LOCK8 0;LOCK9 0;  
LOCK10 0;LOCK11 0;LOCK12 0

### **:FILE:INTErnal:RECall:DMEMORy [:EXECute]**

機能 内部メモリからのアキュムレートデータの読み込みを実行します。オーバーラップコマンドです。

構文 :FILE:INTErnal:RECall:DMEMORy  
[:EXECute] {<NRf>}  
<NRf> = 1 ~ 4

例 :FILE:INTErnal:RECall:DMEMORy:  
EXECUTE 1

### **:FILE:INTErnal:RECall:SETup [:EXECute]**

機能 内部メモリからの設定データの読み込みを実行します。オーバーラップコマンドです。

構文 :FILE:INTErnal:RECall:SETup  
[:EXECute] {<NRf>}  
<NRf> = 1 ~ 12

例 :FILE:INTErnal:RECall:SETup:EXECUTE 1

### **:FILE:INTErnal:STORE?**

機能 内部メモリへの保存に関するすべての設定値を問い合わせます。

構文 :FILE:INTErnal:STORE?

例 :FILE:INTErnal:STORE? -> :FILE:  
INTErnal:STORE:BINARy:TRAcE 1;:  
FILE:INTErnal:STORE:DMEMORy:TRAcE 1;:  
FILE:INTErnal:STORE:SETUP:  
COMMENT1 "WAVE1";COMMENT2 "";  
COMMENT3 "";COMMENT4 "";COMMENT5 "";  
COMMENT6 "";COMMENT7 "";COMMENT8 "";  
COMMENT9 "";COMMENT10 "";  
COMMENT11 "";COMMENT12 "";LOCK1 1;  
LOCK2 0;LOCK3 0;LOCK4 0;LOCK5 0;  
LOCK6 0;LOCK7 0;LOCK8 0;LOCK9 0;  
LOCK10 0;LOCK11 0;LOCK12 0

### **:FILE:INTErnal:STORE:BINARy?**

機能 内部メモリへのバイナリデータの保存に関するすべての設定値を問い合わせます。

構文 :FILE:INTErnal:STORE:BINARy?

例 :FILE:INTErnal:STORE:BINARy?  
-> :FILE:INTErnal:STORE:BINARy:  
TRAcE 1

### **:FILE:INTErnal:STORE:BINARy [:EXECute]**

機能 内部メモリへのバイナリデータの保存を実行します。オーバーラップコマンドです。

構文 :FILE:INTErnal:STORE:BINARy  
[:EXECute] {<NRf>}  
<NRf> = 1 ~ 4

例 :FILE:INTErnal:STORE:BINARy:EXECUTE 1

### **:FILE:INTErnal:STORE:BINARy:TRAcE**

機能 内部メモリへのバイナリデータで保存するトレースを設定 / 問い合わせします。

構文 :FILE:INTErnal:STORE:BINARy:  
TRAcE {<NRf>}  
:FILE:INTErnal:STORE:BINARy:TRAcE?  
<NRf> = 1 ~ 8

例 :FILE:INTErnal:STORE:BINARy:TRAcE 1  
:FILE:INTErnal:STORE:BINARy:TRAcE?  
-> :FILE:INTErnal:STORE:BINARy:  
TRAcE 1

### **:FILE:INTErnal:STORE:DMEMORy?**

機能 内部メモリへのアキュムレートデータの保存に関するすべての設定値を問い合わせます。

構文 :FILE:INTErnal:STORE:DMEMORy?  
例 :FILE:INTErnal:STORE:DMEMORy?

-> :FILE:INTErnal:STORE:DMEMORy:  
TRAcE 1

**:FILE:INTERNAL:STORE:DMEemory  
[:EXECute]**

機能 内部メモリへのアキュムレートデータの保存を実行します。オーバラップコマンドです。

構文 :FILE:INTERNAL:STORE:DMEemory  
[:EXECute] {<Nrf>}  
<Nrf> = 1 ~ 4

例 :FILE:INTERNAL:STORE:DMEemory:  
EXECUTE 1

**:FILE:INTERNAL:STORE:DMEemory:TRACe**

機能 内部メモリへのアキュムレートデータで保存するトレースを設定 / 問い合わせします。

構文 :FILE:INTERNAL:STORE:DMEemory:  
TRACe {<Nrf>|ALL|XY1|XY2}  
:FILE:INTERNAL:STORE:DMEemory:TRACe?  
<Nrf> = 1 ~ 8

例 :FILE:INTERNAL:STORE:DMEemory:TRACe 1  
:FILE:INTERNAL:STORE:DMEemory:TRACe?  
-> :FILE:INTERNAL:STORE:DMEemory:  
TRACe 1

**:FILE:INTERNAL:STORE:SETup?**

機能 内部メモリへの設定データの保存に関するすべての設定値を問い合わせます。

構文 :FILE:INTERNAL:STORE:SETup?

例 :FILE:INTERNAL:STORE:SETup? ->  
:FILE:INTERNAL:STORE:SETup:  
COMMENT1 "WAVE1";COMMENT2 "";  
COMMENT3 "";COMMENT4 "";COMMENT5 "";  
COMMENT6 "";COMMENT7 "";COMMENT8 "";  
COMMENT9 "";COMMENT10 "";  
COMMENT11 "";COMMENT12 "";LOCK1 1;  
LOCK2 0;LOCK3 0;LOCK4 0;LOCK5 0;  
LOCK6 0;LOCK7 0;LOCK8 0;LOCK9 0;  
LOCK10 0;LOCK11 0;LOCK12 0

**:FILE:INTERNAL:STORE:SETup:  
COMMeNT<x>**

機能 内部メモリへの設定データ保存時のコメントを設定 / 問い合わせします。

構文 :FILE:INTERNAL:STORE:SETup:  
COMMeNT<x> {<文字列>}  
:FILE:INTERNAL:STORE:SETup:  
COMMeNT<x>?  
<x> = 1 ~ 12  
<文字列> = 16 文字以内

例 :FILE:INTERNAL:STORE:SETup:  
COMMENT1 "WAVE1"  
:FILE:INTERNAL:STORE:SETup:COMMENT1?  
-> :FILE:INTERNAL:STORE:SETup:  
COMMENT1 "WAVE1"

**:FILE:INTERNAL:STORE:SETup:DATE<x>?**

機能 内部メモリへの設定データ保存時の日付 / 時刻を問い合わせます。

構文 :FILE:INTERNAL:STORE:SETup:DATE<x>?  
<x> = 1 ~ 12

例 :FILE:INTERNAL:STORE:SETup:DATE1?  
-> :FILE:INTERNAL:STORE:SETup:  
DATE1 "2007/01/12 14:58:02"

**:FILE:INTERNAL:STORE:SETup[:EXECute]**

機能 内部メモリへの設定データの保存を実行します。オーバラップコマンドです。

構文 :FILE:INTERNAL:STORE:SETup  
[:EXECute] {<Nrf>}  
<Nrf>=1 ~ 12

例 :FILE:INTERNAL:STORE:SETup:EXECUTE 1

**:FILE:INTERNAL:STORE:SETup:LOCK<x>**

機能 内部メモリへの設定データ上書き禁止の ON/OFF を設定 / 問い合わせします。

構文 :FILE:INTERNAL:STORE:SETup:  
LOCK<x> {<Boolean>}  
:FILE:INTERNAL:STORE:SETup:LOCK<x>?  
<x> = 1 ~ 12

例 :FILE:INTERNAL:STORE:SETup:LOCK1 ON  
:FILE:INTERNAL:STORE:SETup:LOCK1?  
-> :FILE:INTERNAL:STORE:SETup:LOCK1 1

**:FILE:INTERNAL:UNLoad:DMEemory  
[:EXECute]**

機能 読み込んだアキュムレートデータを取り消します。

構文 :FILE:INTERNAL:UNLoad:DMEemory[:  
EXECute]

例 :FILE:INTERNAL:UNLoad:DMEemory:EXECUTE

**:FILE:LOAD?**

機能 指定したストレージメディアのファイルの読み込みに関するすべての設定値を問い合わせます。

構文 :FILE:LOAD?

例 :FILE:LOAD? -> :FILE:LOAD:BINARy:  
REFERENCE 1;:FILE:LOAD:ZPOLYGoN:  
ZONE 1;:FILE:LOAD:ZWAVE:ZONE 1

**:FILE:LOAD:{BINARy|ZPOLYGoN|ZWAVE}?**

機能 各種データの読み込みに関するすべての設定値を問い合わせます。

構文 :FILE:LOAD:{BINARy|ZPOLYGoN|ZWAVE}?

例 (以下は、バイナリデータについての例です。)  
:FILE:LOAD:BINARy? -> :FILE:LOAD:  
BINARy:REFERENCE 1

## 5.11 FILE グループ

### **: FILE : LOAD : { BINARY | DMEMORY | MASK | SETUP | SYMBOL | ZPOLYGON | ZWAVE } : ABORT**

機能 各種データの読み込みを中止します。  
 構文 : FILE:LOAD:{BINARY|DMEMORY|MASK|SETUP|SYMBOL|ZPOLYGON|ZWAVE}:ABORT  
 例 (以下は、バイナリデータについての例です。)  
 : FILE:LOAD: BINARY: ABORT

### **: FILE : LOAD : { BINARY | DMEMORY | MASK | SETUP | SYMBOL | ZPOLYGON | ZWAVE } [ : EXECUTE ]**

機能 各種データの読み込みを実行します。オーバーラップコマンドです。  
 構文 : FILE:LOAD:{BINARY|DMEMORY|MASK|SETUP|SYMBOL|ZPOLYGON|ZWAVE}[ : EXECUTE ] { < 文字列 > }  
 < 文字列 > = 259 文字以内  
 例 (以下は、バイナリデータについての例です。)  
 : FILE:LOAD: BINARY: EXECUTE "\Flash Mem\DIR\DATA"  
 (絶対パス指定)  
 : FILE:LOAD: BINARY: EXECUTE "DATA"  
 (相対パス指定)  
 解説 < 文字列 > に関する詳細は、「FILE[:DIRectory]:CDIRectory」の解説を参照してください。

### **: FILE : LOAD : BINARY : REFERENCE**

機能 バイナリデータの読み込み先を設定 / 問い合わせします。  
 構文 : FILE:LOAD: BINARY: REFERENCE { < Nrf > | ACQMemory }  
 : FILE:LOAD: BINARY: REFERENCE ?  
 < Nrf > = 1 ~ 4  
 例 : FILE:LOAD: BINARY: REFERENCE 1  
 : FILE:LOAD: BINARY: REFERENCE ?  
 -> : FILE:LOAD: BINARY: REFERENCE 1

### **: FILE : LOAD : { ZPOLYGON | ZWAVE } : ZONE**

機能 各種データのゾーンデータの読み込み先を設定 / 問い合わせします。  
 構文 : FILE:LOAD:{ZPOLYGON|ZWAVE}: ZONE { < Nrf > }  
 : FILE:LOAD:{ZPOLYGON|ZWAVE}: ZONE ?  
 < Nrf > = 1 ~ 4  
 例 (以下は、ZONE WAVE データについての例です。)  
 : FILE:LOAD: ZWAVE: ZONE 1  
 : FILE:LOAD: ZWAVE: ZONE ? -> : FILE:LOAD: ZWAVE: ZONE 1

### **: FILE : SAVE ?**

機能 データの保存に関するすべての設定値を問い合わせます。  
 構文 : FILE:SAVE?  
 例 : FILE:SAVE? -> : FILE:SAVE: AHISTOGRAM: ANALYSIS 1; : FILE:SAVE: ANAMING DATE; ASCII: COMPRESSION DECIMATION; HISTORY ONE; LENGTH 2500; RANGE MAIN; TRACE 1; : FILE:SAVE: BINARY: COMPRESSION DECIMATION; HISTORY ONE; LENGTH 2500; TRACE 1; : FILE:SAVE: COMMENT "THIS IS TEST"; DMEMORY: TRACE 1; : FILE:SAVE: FFT: ANALYSIS 1; : FILE:SAVE: FLOAT: COMPRESSION DECIMATION; HISTORY ONE; LENGTH 2500; TRACE 1; : FILE:SAVE: NAME "DL9000"; SBUS: ANALYSIS 1; : FILE:SAVE: ZWAVE: ZONE 1

### **: FILE : SAVE : { AHISTOGRAM | ASCII | BINARY | DMEMORY | FFT | FLOAT | SBUS | ZWAVE } ?**

機能 各種データの保存に関するすべての設定値を問い合わせます。  
 構文 : FILE:SAVE:{AHISTOGRAM|ASCII|BINARY|DMEMORY|FFT|FLOAT|SBUS|ZWAVE} ?  
 例 (以下は、バイナリデータについての例です。)  
 : FILE:SAVE: BINARY ? -> : FILE:SAVE: BINARY: COMPRESSION DECIMATION; HISTORY ONE; TRACE 1

### **: FILE : SAVE : { AHISTOGRAM | ASCII | BINARY | DMEMORY | FFT | FLOAT | SETUP | SBUS | WPARAMETER | ZWAVE } : ABORT**

機能 各種データの保存を中止します。  
 構文 : FILE:SAVE:{AHISTOGRAM|ASCII|BINARY|DMEMORY|FFT|FLOAT|SETUP|SBUS|WPARAMETER|ZWAVE}: ABORT  
 例 (以下は、バイナリデータについての例です。)  
 : FILE:SAVE: BINARY: ABORT

**:FILE:SAVE:{AHISTogram|ASCii|BINary|DMEemory|FFT|FLOat|SETup|SBUS|WPARAMeter|ZWAVE}{:EXECute}**

**機能** 各種データをファイル名を指定して保存を実行します。オーバーラップコマンドです。

**構文** :FILE:SAVE:{AHISTogram|ASCii|BINary|DMEemory|FFT|FLOat|SETup|SBUS|WPARAMeter|ZWAVE}{:EXECute} {<文字列>}  
<文字列> = 259 文字以内

**例** (以下は、バイナリデータについての例です。)  
:FILE:SAVE:BINary:EXECUTE  
:FILE:SAVE:BINary:  
EXECUTE "\Flash Mem\DIR\DATA"  
(絶対パス指定)  
:FILE:SAVE:BINary:EXECUTE "DATA"  
(相対パス指定)

**解説**

- ・パスを指定しないときは、「:FILE:SAVE:NAME」で指定したファイル名で保存します。
- ・パスを指定したときは、ファイル名の自動作成機能は動作しません。
- ・<文字列>に関するの詳細は、「FILE[:DIRectory]:CDIRectory」の解説を参照してください。

**:FILE:SAVE:{AHISTogram|FFT|SBUS}:ANALysis**

**機能** 各種データで保存するトレースを設定 / 問い合わせします。

**構文** FILE:SAVE:{AHISTogram|FFT|SBUS}:ANALysis {<Nrf>}  
FILE:SAVE:{AHISTogram|FFT|SBUS}:ANALysis?  
<Nrf> = 1 ~ 8

**例** (以下は、FFT データについての例です。)  
:FILE:SAVE:FFT:ANALYSIS 1  
:FILE:SAVE:FFT:ANALYSIS? -> :FILE:SAVE:FFT:ANALYSIS 1

**:FILE:SAVE:ANAMing**

**機能** 保存するデータのファイル名の自動作成の種類を設定 / 問い合わせします。

**構文** :FILE:SAVE:ANAMing {DATE|NUMBer|OFF}

**例** :FILE:SAVE:ANAMing DATE  
:FILE:SAVE:ANAMing? -> :FILE:SAVE:ANAMing DATE

**:FILE:SAVE:{ASCii|BINary|FLOat}:COMPRession**

**機能** 各種データの圧縮保存方法を設定 / 問い合わせします。

**構文** :FILE:SAVE:{ASCii|BINary|FLOat}:COMPRession {DECimation|OFF|PTOPeak|STATE}  
:FILE:SAVE:{ASCii|BINary|FLOat}:COMPRession?

**例** (以下は、バイナリデータについての例です。)  
:FILE:SAVE:BINary:  
COMPRession DECIMATION  
:FILE:SAVE:BINary:COMPRession?  
-> :FILE:SAVE:BINary:  
COMPRession DECIMATION

**解説**

- ・「:FILE:SAVE:{ASCii|BINary}:TRACe LOGic」のときは、{DECimation}は無効です。
- ・「:FILE:SAVE:ASCii:TRACe LOGic」以外のときは、{STATE}は無効です。

**:FILE:SAVE:{ASCii|BINary|FLOat}:HISTory**

**機能** 各種データの保存対象が全データが選択されたデータかを設定 / 問い合わせします。

**構文** :FILE:SAVE:{ASCii|BINary|FLOat}:HISTory {ALL|ONE}  
:FILE:SAVE:{ASCii|BINary|FLOat}:HISTory?

**例** (以下は、バイナリデータについての例です。)  
:FILE:SAVE:BINary:HISTory ALL  
:FILE:SAVE:BINary:HISTory? -> :FILE:SAVE:BINary:HISTory ALL

**:FILE:SAVE:{ASCii|BINary|FLOat}:LENGth**

**機能** 各種データで保存する波形データのサイズを設定 / 問い合わせします。

**構文** :FILE:SAVE:{ASCii|BINary|FLOat}:LENGth {<Nrf>}  
:FILE:SAVE:{ASCii|BINary|FLOat}:LENGth?  
<Nrf> = 本体ユーザーズマニュアル参照。

**例** (以下は、バイナリデータについての例です。)  
:FILE:SAVE:BINary:LENGth 2500  
:FILE:SAVE:BINary:LENGth? -> :FILE:SAVE:BINary:LENGth 2500

**:FILE:SAVE:ASCii:RANGe**

**機能** アスキーデータの保存する範囲を設定 / 問い合わせします。

**構文** :FILE:SAVE:ASCii:RANGe {MAIN|Z1|Z2}  
:FILE:SAVE:ASCii:RANGe?

**例** :FILE:SAVE:ASCii:RANGe MAIN  
:FILE:SAVE:ASCii:RANGe? -> :FILE:SAVE:ASCii:RANGe MAIN

## 5.11 FILE グループ

### **:FILE:SAVE:{ASCIi|BINArY|FLOat}:**

#### **TRACe**

機能 各種データで保存するトレースを設定 / 問い合わせします。

構文 :FILE:SAVE:{ASCIi|BINArY|FLOat}:  
TRACe {<NRf>|ALL|LOGic}  
:FILE:SAVE:{ASCIi|BINArY|FLOat}:  
TRACe?

例 (以下は、バイナリデータについての例です。)  
<NRf> = 1 ~ 8

```
:FILE:SAVE:BINArY:TRACe 1  
:FILE:SAVE:BINArY:TRACe? -> :FILE:  
SAVE:BINArY:TRACe 1
```

解説 「:FILE:SAVE:{FLOat}:TRACe」のときは、対象波形 {LOGic} は無効です。

### **:FILE:SAVE:COMMeNt**

機能 各種データの保存するコメントを設定 / 問い合わせします。

構文 :FILE:SAVE:COMMeNt {<文字列>}  
:FILE:SAVE:COMMeNt?

例 <文字列> = 160 文字以内  
:FILE:SAVE:COMMeNt "THIS IS TEST"  
:FILE:SAVE:COMMeNt? -> :FILE:SAVE:  
COMMeNt "THIS IS TEST"

### **:FILE:SAVE:DMEMory:TRACe**

機能 アキュムレートデータで保存するトレースを設定 / 問い合わせします。

構文 :FILE:SAVE:DMEMory:TRACe {<NRf>|ALL|  
XY1|XY2}  
:FILE:SAVE:DMEMory:TRACe?

例 <NRf> = 1 ~ 8  
:FILE:SAVE:DMEMory:TRACe 1  
:FILE:SAVE:DMEMory:TRACe? -> :FILE:  
SAVE:DMEMory:TRACe 1

### **:FILE:SAVE:NAME**

機能 各種データの保存するデータのファイル名を設定 / 問い合わせします。

構文 :FILE:SAVE:NAME {<Filename>}  
:FILE:SAVE:NAME?

例 :FILE:SAVE:NAME "DL9000"  
:FILE:SAVE:NAME? -> :FILE:SAVE:  
NAME "DL9000"

### **:FILE:SAVE:ZWAVe:ZONE**

機能 ゾーンデータで保存するゾーンを設定 / 問い合わせします。

構文 :FILE:SAVE:ZWAVe:ZONE {<NRf>}  
:FILE:SAVE:ZWAVe:ZONE?

例 <NRf> = 1 ~ 4  
:FILE:SAVE:ZWAVe:ZONE 1  
:FILE:SAVE:ZWAVe:ZONE? -> :FILE:SAVE:  
ZWAVe:ZONE 1

## 5.12 GONogo グループ

### :GONogo?

機能 GO/NO-GO 判定に関するすべての設定値を問い合わせます。

構文 :GONogo?

例 :GONOGO? -> :GONOGO:ACTION:BUZZER 0;  
HCOPIY 0;MAIL:INTERVAL OFF;MODE 0;;  
GONOGO:ACTION:SAVE 0;;GONOGO:  
CONDITION1 DONTCARE;  
CONDITION2 DONTCARE;  
CONDITION3 DONTCARE;  
CONDITION4 DONTCARE;  
LOGIC AND;MODE OFF;SCONDITION:  
NGCOUNT 1;STOPCOUNT 1;;GONOGO:  
TELECOMTEST:SELECT1:MASK:ELEMENT1:  
WCOUNT 1.000E+00,2.000E+00;;GONOGO:  
TELECOMTEST:SELECT2:MASK:ELEMENT1:  
WCOUNT 1.000E+00,2.000E+00;;GONOGO:  
TELECOMTEST:SELECT3:MASK:ELEMENT1:  
WCOUNT 1.000E+00,2.000E+00;;GONOGO:  
TELECOMTEST:SELECT4:MASK:ELEMENT1:  
WCOUNT 2.000E+00,3.000E+00;;GONOGO:  
ZPARAMETER:SELECT1:MODE PARAMETER;  
PARAMETER:CATEGORY FFT;FFT1:  
CALCULATION10.000E+00,1.000E+00;;  
GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
MEASURE:STATISTICS MAXIMUM;;GONOGO:  
ZPARAMETER:SELECT1:RECTANGLE:  
HORIZONTAL 0.000E+00,1.000E+00;  
VERTICAL 0.000E+00,1.000E+00;;GONOGO:  
ZPARAMETER:SELECT1:TRACE 1;WAVE:  
TRANGE 1.000E+00,2.000E+00;;GONOGO:  
ZPARAMETER:SELECT1:WINDOW MAIN;;  
GONOGO:ZPARAMETER:SELECT2:MODE  
PARAMETER;PARAMETER:  
CATEGORY FFT;FFT1:  
CALCULATION1 0.000E+00,1.000E+00...

### :GONogo:ABORt

機能 GO/NO-GO 判定を中止します。

構文 :GONogo:ABORt

例 :GONOGO:ABORT

### :GONogo:ACTion?

機能 判定が NO-GO 時の動作およびその基準に関するすべての設定値を問い合わせます。

構文 :GONogo:ACTion?

例 :GONOGO:ACTION? -> :GONOGO:ACTION:  
BUZZER 0;HCOPIY 0;MAIL:INTERVAL OFF;  
MODE 0;;GONOGO:ACTION:SAVE 0

### :GONogo:ACTion:BUZZer

機能 判定が NO-GO 時に警告音を鳴らす (ON)/ 鳴らさない (OFF) を設定 / 問い合わせします。

構文 :GONogo:ACTion:BUZZer {<Boolean>}  
:GONogo:ACTion:BUZZer?

例 :GONOGO:ACTION:BUZZER ON  
:GONOGO:ACTION:BUZZER? -> :GONOGO:  
ACTION:BUZZER 1

### :GONogo:ACTion:HCOPIy

機能 判定が NO-GO 時にプリンタに画面イメージを出力する (ON)/ しない (OFF) を設定 / 問い合わせします。

構文 :GONogo:ACTion:HCOPIy {<Boolean>}  
:GONogo:ACTion:HCOPIy?

例 :GONOGO:ACTION:HCOPIY ON  
:GONOGO:ACTION:HCOPIY? -> :GONOGO:  
ACTION:HCOPIY 1

### :GONogo:ACTion:MAIL?

機能 判定が NO-GO 時のメール送信に関するすべての設定値を問い合わせます。

構文 :GONogo:ACTion:MAIL?

例 :GONOGO:ACTION:MAIL? -> :GONOGO:  
ACTION:MAIL:INTERVAL 10;MODE 1

### :GONogo:ACTion:MAIL:INTerval

機能 判定が NO-GO 時にメール送信する間隔を設定 / 問い合わせします。

構文 :GONogo:ACTion:MAIL:INTerval  
{OFF|<Nrf>}  
:GONogo:ACTion:MAIL:INTerval?  
<Nrf> = 1 ~ 1440(min)

例 :GONOGO:ACTION:MAIL:INTERVAL 10  
:GONOGO:ACTION:MAIL:INTERVAL?  
-> :GONOGO:ACTION:MAIL:INTERVAL 10

### :GONogo:ACTion:MAIL:MODE

機能 判定が NO-GO 時にメール転送する (ON)/ しない (OFF) を設定 / 問い合わせします。

構文 :GONogo:ACTion:MAIL:MODE {<Boolean>}  
:GONogo:ACTion:MAIL:MODE?

例 :GONOGO:ACTION:MAIL:MODE ON  
:GONOGO:ACTION:MAIL:MODE? -> :GONOGO:  
ACTION:MAIL:MODE 1

## 5.12 GONogo グループ

### :GONogo:ACTion:SAVE

機能 判定が NO-GO 時に波形データをメディアに保存する (ON)/ しない (OFF) を設定 / 問い合わせします。

構文 :GONogo:ACTion:SAVE {<Boolean>}  
:GONogo:ACTion:SAVE?

例 :GONOGO:ACTion:SAVE ON  
:GONOGO:ACTion:SAVE? -> :GONOGO:  
ACTion:SAVE 1

### :GONogo:CONDition<x>

機能 GO/NO-GO 判定の各判定基準を設定 / 問い合わせします。

構文 :GONogo:CONDition<x>  
{DONTcare|IN|OUT}  
:GONogo:CONDition<x>?  
<x> = 1 ~ 4

例 :GONOGO:CONDition1 DONTCARE  
:GONOGO:CONDition1? -> :GONOGO:  
CONDition1 DONTCARE

### :GONogo:COUNT?

機能 GO/NO-GO の実判定回数を問い合わせます。

構文 :GONogo:COUNT?  
例 :GONOGO:COUNT? -> :GONOGO:COUNT 1

### :GONogo:EXECute

機能 GO/NO-GO 判定を実行します。オーバーラップコマンドです。

構文 :GONogo:EXECute  
例 :GONOGO:EXECUTE

### :GONogo:LOGic

機能 GO/NO-GO 判定のロジックを設定 / 問い合わせします。

構文 :GONogo:LOGic {AND|OR}  
:GONogo:LOGic?

例 :GONOGO:LOGic AND  
:GONOGO:LOGic? -> :GONOGO:LOGic AND

### :GONogo:MODE

機能 GO/NO-GO 判定の種類を設定 / 問い合わせします。

構文 GONogo:MODE {OFF|TELEcomtest|  
ZPARAMeter}  
:GONogo:MODE?

例 GONOGO:MODE OFF  
:GONOGO:MODE? -> :GONOGO:MODE OFF

### :GONogo:NGCount?

機能 GO/NO-GO 判定の実 NO-GO 回数を問い合わせます。

構文 :GONogo:NGCount?  
例 :GONOGO:NGCOUNT? -> :GONOGO:NGCOUNT 1

### :GONogo:SCONdition?

#### (Stop Condition)

機能 判定終了条件に関するすべての設定値を問い合わせます。

構文 :GONogo:SCONdition?  
例 :GONOGO:SCONdition? -> :GONOGO:  
SCONdition:NGCOUNT 1;STOPCOUNT 1

### :GONogo[:SCONdition]:NGCount

機能 判定終了 NO-GO 回数を設定 / 問い合わせします。

構文 :GONogo[:SCONdition]:NGCount {<Nrf>|  
INFinite}  
:GONogo[:SCONdition]:NGCount?  
<Nrf> = 1 ~ 1000

例 :GONOGO:SCONdition:NGCOUNT 1  
:GONOGO:SCONdition:NGCOUNT?  
-> :GONOGO:SCONdition:NGCOUNT 1

### :GONogo[:SCONdition]:STOPcount

機能 判定終了取り込み回数を設定 / 問い合わせします。

構文 :GONogo[:SCONdition]:STOPcount  
{<Nrf>|INFinite}  
:GONogo[:SCONdition]:STOPcount?  
<Nrf> = 1 ~ 1000000

例 :GONOGO:SCONdition:STOPCOUNT 1  
:GONOGO:SCONdition:STOPCOUNT?  
-> :GONOGO:SCONdition:STOPCOUNT 1

### :GONogo:TELEcomtest?

機能 テレコムテスト判定に関するすべての設定値を問い合わせます。

構文 :GONogo:TELEcomtest?

例 :GONOGO:TELECOMTEST? -> :GONOGO:  
TELECOMTEST:SELECT1:MASK:ELEMENT1:  
WCOUNT 2.000E+00,1.000E+00;:GONOGO:  
TELECOMTEST:SELECT2:MASK:ELEMENT1:  
WCOUNT 2.000E+00,1.000E+00;:GONOGO:  
TELECOMTEST:SELECT3:MASK:ELEMENT1:  
WCOUNT 2.000E+00,1.000E+00;:GONOGO:  
TELECOMTEST:SELECT4:MASK:ELEMENT1:  
WCOUNT 2.000E+00,1.000E+00

### :GONogo:TELEcomtest:SElect<x>?

機能 テレコムテスト判定の各条件に関するすべての設定値を問い合わせます。

構文 :GONogo:TELEcomtest:SElect<x>?  
<x> = 1 ~ 4

例 :GONOGO:TELECOMTEST:SELECT1?  
-> :GONOGO:TELECOMTEST:SELECT1:MASK:  
ELEMENT1:PSPCOUNT 2.000E+00,1.000E+00

**:GONogo:TELEcomtest:SElect<x>:****EYEPattern?**

機能 各条件のアイパターン判定に関するすべての設定値を問い合わせます。

構文 :GONogo:TELEcomtest:SElect<x>:  
EYEPattern?

<x> = 1 ~ 4

例 :GONOGO:TELECOMTEST:SELECT1:  
EYEPATTERN? -> :GONOGO:TELECOMTEST:  
SELECT1:EYEPATTERN:  
DBERATE 2.000E+00,1.000E+00

**:GONogo:TELEcomtest:SElect<x>:****EYEPattern:<パラメータ>**

機能 アイパターン判定の波形パラメータの上下限値を設定 / 問い合わせします。

構文 :GONogo:TELEcomtest:SElect<x>:  
EYEPattern:<パラメータ>{<Nrf>,<Nrf>|<  
電圧>,<電圧>|<電流>,<電流>|<時間>,<  
時間>}

:GONogo:TELEcomtest:SElect<x>:

EYEPattern:<パラメータ>?

SElect<x>の<x> = 1 ~ 4

<パラメータ> = {DBERate|EHEight|EWIDTh|  
FALL|JITTer<y>|PCROSSing|PDUtycycle|  
QFActor|RISE|SDBase|SDTop|T1CRossing|  
T2CRossing|VBASe|VCRossing|VTOP}  
<Nrf>,<電圧>,<電流>,<時間> = 本体ユーザー  
ズマニュアル (IM701331-01) を参照してください。  
<y>=1、2

例 以下は DBERate についての例です。  
:GONOGO:TELECOMTEST:SELECT1:  
EYEPATTERN:DBERATE 1,2  
:GONOGO:TELECOMTEST:SELECT1:  
EYEPATTERN:DBERATE? -> :GONOGO:  
TELECOMTEST:SELECT1:EYEPATTERN:  
DBERATE 2.000E+00,1.000E+00

解説

- 通信コマンドと本体で 사용되는パラメータの対応は付録 4 を参照してください。
- パラメータの詳細は本体ユーザーズマニュアル (IM701331-01) を参照してください。

**:GONogo:TELEcomtest:SElect<x>:MASK?**

機能 各条件のマスク判定に関するすべての設定値を問い合わせます。

構文 :GONogo:TELEcomtest:SElect<x>:MASK?  
<x> = 1 ~ 4

例 :GONOGO:TELECOMTEST:SELECT1:MASK?  
-> :GONOGO:TELECOMTEST:SELECT1:MASK:  
ELEMENT1:PSPCOUNT 2.000E+00,1.000E+00

**:GONogo:TELEcomtest:SElect<x>:MASK:****ELEment<x>?**

機能 マスク判定に使用する各エレメントに関するすべての設定値を問い合わせます。

構文 :GONogo:TELEcomtest:SElect<x>:MASK:  
ELEment<x>?

SElect<x>の<x> = 1 ~ 4

ELEment<x>の<x> = 1 ~ 4

例 :GONOGO:TELECOMTEST:SELECT1:MASK:  
ELEMENT1? -> :GONOGO:TELECOMTEST:  
SELECT1:MASK:ELEMENT1:  
PSPCOUNT 2.000E+00,1.000E+00

**:GONogo:TELEcomtest:SElect<x>:MASK:****ELEment<x>:PSPCount****(Sample Point Count %)**

機能 各エレメントのサンプルデータ数に対するエラー率の上下限値を設定 / 問い合わせします。

構文 :GONogo:TELEcomtest:SElect<x>:MASK:  
ELEment<x>:PSPCount {<Nrf>,<Nrf>}

:GONogo:TELEcomtest:SElect<x>:MASK:

ELEment<x>:PSPCount?

SElect<x>の<x> = 1 ~ 4

ELEment<x>の<x> = 1 ~ 4

<Nrf> = 0 ~ 100(%)

例 :GONOGO:TELECOMTEST:SELECT1:MASK:  
ELEMENT1:PSPCOUNT 1,2  
:GONOGO:TELECOMTEST:SELECT1:MASK:  
ELEMENT1:PSPCOUNT? -> :GONOGO:  
TELECOMTEST:SELECT1:MASK:ELEMENT1:  
PSPCOUNT 2.000E+00,1.000E+00

**:GONogo:TELEcomtest:SElect<x>:MASK:****ELEment<x>:PWCount****(Wave Count %)**

機能 各エレメントのアクイジション回数に対するエラー率の上下限値を設定 / 問い合わせします。

構文 :GONogo:TELEcomtest:SElect<x>:MASK:  
ELEment<x>:PWCount {<Nrf>,<Nrf>}

:GONogo:TELEcomtest:SElect<x>:MASK:

ELEment<x>:PWCount?

SElect<x>の<x> = 1 ~ 4

ELEment<x>の<x> = 1 ~ 4

<Nrf> = 0 ~ 100(%)

例 :GONOGO:TELECOMTEST:SELECT1:MASK:  
ELEMENT1:PWCOUNT 1,2  
:GONOGO:TELECOMTEST:SELECT1:MASK:  
ELEMENT1:PWCOUNT? -> :GONOGO:  
TELECOMTEST:SELECT1:MASK:  
ELEMENT1:PWCOUNT 2.000E+00,1.000E+00



## 5.12 GONogo グループ

### :GONogo:TELecomtest:SElect<x>:MASK: ELEMeNt<x>:SPCount (Sample Point Count)

機能 各エレメントのエラーになったサンプルデータ数の上下限値を設定 / 問い合わせします。

構文 :GONogo:TELecomtest:SElect<x>:MASK:ELEMeNt<x>:SPCount {<NRf>,<NRf>}  
:GONogo:TELecomtest:SElect<x>:MASK:  
ELEMeNt<x>:SPCount?  
SElect<x> の <x> = 1 ~ 4  
ELEMeNt<x> の <x> = 1 ~ 4  
<NRf> = 本体ユーザーズマニュアル参照。

例 :GONOGO:TELECOMTEST:SELECT1:MASK:  
ELEMENT1:SPCOUNT 1,2  
:GONOGO:TELECOMTEST:SELECT1:MASK:  
ELEMENT1:SPCOUNT? -> :GONOGO:  
TELECOMTEST:SELECT1:MASK:ELEMENT1:  
SPCOUNT 2.000E+00,1.000E+00

### :GONogo:TELecomtest:SElect<x>:MASK: ELEMeNt<x>:WCOunt (Wave Count)

機能 各エレメントのエラーになったアクイジション回数の上下限値を設定 / 問い合わせします。

構文 :GONogo:TELecomtest:SElect<x>:MASK:  
ELEMeNt<x>:WCOunt {<NRf>,<NRf>}  
:GONogo:TELecomtest:SElect<x>:MASK:  
ELEMeNt<x>:WCOunt?  
SElect<x> の <x> = 1 ~ 4  
ELEMeNt<x> の <x> = 1 ~ 4  
<NRf> = 本体ユーザーズマニュアル参照。

例 :GONOGO:TELECOMTEST:SELECT1:MASK:  
ELEMENT1:WCOUNT 1,2  
:GONOGO:TELECOMTEST:SELECT1:MASK:  
ELEMENT1:WCOUNT? -> :GONOGO:  
TELECOMTEST:SELECT1:MASK:ELEMENT1:  
WCOUNT 2.000E+00,1.000E+00

### :GONogo:ZPARAmeter?

機能 ゾーン / パラメータ判定に関するすべての設定値を問い合わせます。

構文 :GONogo:ZPARAmeter?  
例 :GONOGO:ZPARAMETER? -> :GONOGO:  
ZPARAMETER:SELECT1:MODE PARAMETER;  
PARAMETER:CATEGORY FFT;FFT1:  
CALCULATION1 0.000E+00,1.000E+00;;  
GONOGO:ZPARAMETER:SELECT1:RECTANGLE:  
HORIZONTAL 0.000E+00,1.000E+00;  
VERTICAL 0.000E+00,1.000E+00;;GONOGO:  
ZPARAMETER:SELECT1:TRACE 1;WAVE:  
TRANGE 1.000E+00,2.000E+00;;GONOGO:  
ZPARAMETER:SELECT1:WINDOW MAIN;;  
GONOGO:ZPARAMETER:SELECT2:  
MODE PARAMETER;PARAMETER:  
CATEGORY FFT;;FFT1:  
CALCULATION1 0.000E+00,1.000E+00;;  
GONOGO:ZPARAMETER:SELECT2:RECTANGLE:  
HORIZONTAL 0.000E+00,1.000E+00;  
VERTICAL 0.000E+00,1.000E+00;;GONOGO:  
ZPARAMETER:SELECT2:TRACE 1;WAVE:  
TRANGE 1.000E+00,2.000E+00;;GONOGO:  
ZPARAMETER:SELECT2:WINDOW MAIN;;  
GONOGO:ZPARAMETER:SELECT3:  
MODE PARAMETER;PARAMETER:  
CATEGORY FFT;FFT1:  
CALCULATION1 0.000E+00,1.000E+00;...  
..

### :GONogo:ZPARAmeter:SElect<x>?

機能 ゾーン / パラメータ判定の各条件に関するすべての設定値を問い合わせます。

構文 :GONogo:ZPARAmeter:SElect<x>?  
<x> = 1 ~ 4

例 :GONOGO:ZPARAMETER:SELECT1?  
-> :GONOGO:ZPARAMETER:SELECT1:  
MODE PARAMETER;PARAMETER:  
CATEGORY FFT;FFT1:  
CALCULATION1 0.000E+00,1.000E+00;;  
GONOGO:ZPARAMETER:SELECT1:RECTANGLE:  
HORIZONTAL 0.000E+00,1.000E+00;  
VERTICAL 0.000E+00,1.000E+00;;GONOGO:  
ZPARAMETER:SELECT1:TRACE 1;WAVE:  
TRANGE 1.000E+00,2.000E+00;;GONOGO:  
ZPARAMETER:SELECT1:WINDOW MAIN

### :GONogo:ZPARAmeter:SElect<x>:MODE

機能 各条件のモードを設定 / 問い合わせします。

構文 :GONogo:ZPARAmeter:SElect<x>:MODE  
{PARAmeter|POLYgon|RECTangle|WAVE}  
:GONogo:ZPARAmeter:SElect<x>:MODE?  
<x> = 1 ~ 4

例 :GONOGO:ZPARAMETER:SELECT1:MODE WAVE  
:GONOGO:ZPARAMETER:SELECT1:MODE?  
-> :GONOGO:ZPARAMETER:SELECT1:  
MODE WAVE

**:GONogo:ZPARAMeter:SElect<x>:  
PARAMeter?**

機能 各条件のパラメータに関するすべての設定値を  
問い合わせます。

構文 :GONogo:ZPARAMeter:SElect<x>:  
PARAMeter?  
<x> = 1 ~ 4

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER?  
-> :GONOGO:ZPARAMETER:SELECT1:  
PARAMETER:CATEGORY FFT;PARAMETER:  
CATEGORY FFT;FFT1:  
CALCULATION1 1.000E+00,0.000E+00

**:GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:CATegory**

機能 パラメータのカテゴリを設定 / 問い合わせしま  
す。

構文 :GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:  
CATegory {FFT|MEASure|XY}  
:GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:CATegory?  
<x> = 1 ~ 4

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
CATEGORY FFT  
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
CATEGORY? -> :GONOGO:ZPARAMETER:  
SELECT1:PARAMETER:CATEGORY FFT

解説 「:MEASURE:MODE CYCLE」のときに有効です。

**:GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:FFT<x>?**

機能 各 FFT 判定に関するすべての設定値を問い合  
わせます。

構文 :GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:FFT<x>?  
SElect<x> の <x> = 1 ~ 4  
FFT<x> の <x> = 1, 2

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
FFT1? -> :GONOGO:ZPARAMETER:SELECT1:  
PARAMETER:FFT1:PEAK:  
FREQUENCY1 0.000E+00,0.000E+00

**:GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:FFT<x>:CALCulation<x>**

機能 各 FFT 判定の各 Calc アイテムの上下限值を設定  
 / 問い合わせします。

構文 :GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:FFT<x>:  
CALCulation<x> {<NRF>,<NRF>}  
:GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:FFT<x>:CALCulation<x>?  
SElect<x> の <x> = 1 ~ 4  
FFT<x> の <x> = 1, 2  
CALCulation<x> の <x> = 1 ~ 4  
<NRF> = -4 ~ 4(div)

例 GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
FFT1:CALCULATION1 0,1  
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
FFT1:CALCULATION1? -> :GONOGO:  
ZPARAMETER:SELECT1:PARAMETER:  
FFT1:CALCULATION1 1.000E+00,0.000E+00

**:GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:FFT<x>:PEAK?**

機能 各 FFT 判定のピーク値に関するすべての設定値  
を問い合わせます。

構文 :GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:  
FFT<x>:PEAK?  
SElect<x> の <x> = 1 ~ 4  
FFT<x> の <x> = 1, 2

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
FFT1:PEAK? -> :GONOGO:ZPARAMETER:  
SELECT1:PARAMETER:FFT1:PEAK:  
FREQUENCY1 1.000E+00,0.000E+00

**:GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:FFT<x>:PEAK:DFREquency**

機能 各 FFT 判定のピーク周波数間の上下限值を設定  
 / 問い合わせします。

構文 :GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:FFT<x>:PEAK:  
DFREquency {<周波数>,<周波数>}  
:GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:FFT<x>:PEAK:DFREquency?  
SElect<x> の <x> = 1 ~ 4  
FFT<x> の <x> = 1, 2  
<周波数> = 本体ユーザーズマニュアル参照。

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
FFT1:PEAK:DFREQUENCY 0,1  
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
FFT1:PEAK:DFREQUENCY? -> :GONOGO:  
ZPARAMETER:SELECT1:PARAMETER:FFT1:  
PEAK:DFREQUENCY 1.000E+00,0.000E+00

## 5.12 GONogo グループ

### **:GONogo:ZPARAMeter:SElect<x>: PARAMeter:FFT<x>:PEAK:DV**

機能 各 FFT 判定のピーク電圧値間の上下限値を設定 / 問い合わせします。

構文 :GONogo:ZPARAMeter:SElect<x>:

PARAMeter:FFT<x>:PEAK:

DV {<NRf>, <NRf>}

:GONogo:ZPARAMeter:SElect<x>:

PARAMeter:FFT<x>:PEAK:DV?

SElect<x> の <x> = 1 ~ 4

FFT<x> の <x> = 1, 2

<NRf> = -4 ~ 4(div)

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:

FFT1:PEAK:DV 0,1

:GONOGO:ZPARAMETER:SELECT1:PARAMETER:

FFT1:PEAK:DV? -> :GONOGO:ZPARAMETER:

SELECT1:PARAMETER:FFT1:PEAK:

DV 1.000E+00,0.000E+00

### **:GONogo:ZPARAMeter:SElect<x>:**

#### **PARAMeter:FFT<x>:PEAK:FREQuency<x>**

機能 各 FFT 判定の各ピーク周波数の上下限値を設定 / 問い合わせします。

構文 :GONogo:ZPARAMeter:SElect<x>:

PARAMeter:FFT<x>:PEAK:

FREQuency<x> {<周波数>, <周波数>}

:GONogo:ZPARAMeter:SElect<x>:

PARAMeter:FFT<x>:PEAK:FREQuency<x>?

SElect<x> の <x> = 1 ~ 4

FFT<x> の <x> = 1, 2

FREQuency<x> の <x> = 1, 2

<周波数> = 本体ユーザーズマニュアル参照。

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:

FFT1:PEAK:FREQUENCY1 0,1

:GONOGO:ZPARAMETER:SELECT1:PARAMETER:

FFT1:PEAK:FREQUENCY1? -> :GONOGO:

ZPARAMETER:SELECT1:PARAMETER:FFT1:

PEAK:FREQUENCY1 1.000E+00,0.000E+00

### **:GONogo:ZPARAMeter:SElect<x>:**

#### **PARAMeter:FFT<x>:PEAK:V<x>**

機能 各 FFT 判定の各ピーク電圧値の上下限値を設定 / 問い合わせします。

構文 :GONogo:ZPARAMeter:SElect<x>:

PARAMeter:FFT<x>:PEAK:

V<x> {<NRf>, <NRf>}

:GONogo:ZPARAMeter:SElect<x>:

PARAMeter:FFT<x>:PEAK:V<x>?

SElect<x> の <x> = 1 ~ 4

FFT<x> の <x> = 1, 2

V<x> の <x> = 1, 2

<NRf> = -4 ~ 4(div)

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:

FFT1:PEAK:V1 1,2

:GONOGO:ZPARAMETER:SELECT1:PARAMETER:

FFT1:PEAK:V1? -> :GONOGO:ZPARAMETER:

SELECT1:PARAMETER:FFT1:PEAK:

V1 2.000E+00,1.000E+00

### **:GONogo:ZPARAMeter:SElect<x>:**

#### **PARAMeter:MEASure?**

機能 波形パラメータの自動測定での判定 (メジャー判定) に関するすべての設定値を問い合わせます。

構文 :GONogo:ZPARAMeter:SElect<x>:

PARAMeter:MEASure?

<x> = 1 ~ 4

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:

MEASURE? -> :GONOGO:ZPARAMETER:

SELECT1:PARAMETER:MEASURE:

CALCULATION1 2.000E+00,1.000E+00

### **:GONogo:ZPARAMeter:SElect<x>:**

#### **PARAMeter:MEASure:BIT<x>?**

機能 メジャー判定の各ロジックビットに関するすべての設定値を問い合わせます。

構文 :GONogo:ZPARAMeter:SElect<x>:

PARAMeter:MEASure:BIT<x>?

SElect<x> の <x> = 1 ~ 4

BIT<x> の <x> = 1 ~ 32(DL9510L、DL9505L は

BIT<x> の <x> = 1 ~ 8、17 ~ 24 が有効です。)

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:

MEASURE:BIT1? -> :GONOGO:ZPARAMETER:

SELECT1:PARAMETER:MEASURE:BIT1:AREA1:

TYPE:COUNT 2.000E+00,1.000E+00

**:GONogo:ZPARAMeter:SElect<x>:****PARAMeter:MEASure:BIT<x>:AREA<x>?**

機能 メジャー判定の各エリアに関するすべての設定値を問い合わせます。

構文 :GONogo:ZPARAMeter:SElect<x>:

PARAMeter:MEASure:BIT<x>:AREA<x>?

SElect<x> の <x> = 1 ~ 4

BIT<x> の <x> = 1 ~ 32(DL9510L、DL9505L は BIT<x> の <x> = 1 ~ 8、17 ~ 24 が有効です。)

AREA<x> の <x> = 1、2

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
MEASURE:BIT1:AREA1? -> :GONOGO:  
ZPARAMETER:SELECT1:PARAMETER:  
MEASURE:BIT1:AREA1:TYPE:  
COUNT 2.000E+00,1.000E+00

**:GONogo:ZPARAMeter:SElect<x>:****PARAMeter:MEASure:BIT<x>:AREA<x>:****TYPE?**

機能 メジャー判定のロジック波形パラメータに関するすべての設定値を問い合わせます。

構文 :GONogo:ZPARAMeter:SElect<x>:

PARAMeter:MEASure:BIT<x>:AREA<x>:

TYPE?

SElect<x> の <x> = 1 ~ 4

BIT<x> の <x> = 1 ~ 32(DL9510L、DL9505L は BIT<x> の <x> = 1 ~ 8、17 ~ 24 が有効です。)

AREA<x> の <x> = 1、2

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
MEASURE:BIT1:AREA1:TYPE? -> :GONOGO:  
ZPARAMETER:SELECT1:PARAMETER:  
MEASURE:BIT1:AREA1:TYPE:  
COUNT 2.000E+00,1.000E+00

**:GONogo:ZPARAMeter:SElect<x>:****PARAMeter:MEASure:BIT<x>:AREA<x>:****TYPE:<パラメータ>**

機能 メジャー判定のロジック波形パラメータの上下限値を設定 / 問い合わせします。

構文 :GONogo:ZPARAMeter:SElect<x>:

PARAMeter:MEASure:BIT<x>:AREA<x>:

TYPE:<パラメータ> {<NRf>,<NRf>|<時間>,<時間>|<周波数>,<周波数>}

:GONogo:ZPARAMeter:SElect<x>:

PARAMeter:MEASure:BIT<x>:AREA<x>:

TYPE:<パラメータ>?

SElect<x> の <x> = 1 ~ 4

BIT<x> の <x> = 1 ~ 32(DL9510L、DL9505L は BIT<x> の <x> = 1 ~ 8、17 ~ 24 が有効です。)

AREA<x> の <x> = 1、2

<パラメータ> = {COUNT|DElay|DT|DUTYcycle|

FREQuency|NWIDth|PERFfrequency|PERiod|

PWIDth}

<NRf>、<時間>、<周波数> = 本体ユーザーズマニュアル参照。

例 (以下は、トレース 1、エリア 1 のカウントについての例です。)

```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
MEASURE:BIT1:AREA1:TYPE:COUNT 1,2
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
MEASURE:BIT1:AREA1:TYPE:COUNT?
-> :GONOGO:ZPARAMETER:SELECT1:
PARAMETER:MEASURE:BIT1:AREA1:TYPE:
COUNT 2.000E+00,1.000E+00
```

解説 「:MEASURE:MODE CYCLE」のときは、周期統計処理 (Cycle Statistics) の対象です。

**:GONogo:ZPARAMeter:SElect<x>:****PARAMeter:MEASure:CALCulation<x>**

機能 メジャー判定の各 Calc アイテムの上下限値を設定 / 問い合わせします。

構文 :GONogo:ZPARAMeter:SElect<x>:

PARAMeter:MEASure:

CALCulation<x> {<NRf>,<NRf>}

:GONogo:ZPARAMeter:SElect<x>:

PARAMeter:MEASure:CALCulation<x>?

SElect<x> の <x> = 1 ~ 4

CALCulation<x> の <x> = 1 ~ 4

<NRf> = - 4 ~ 4(div)

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
MEASURE:CALCULATION1 1,2  
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
MEASURE:CALCULATION1? -> :GONOGO:  
ZPARAMETER:SELECT1:PARAMETER:MEASURE:  
CALCULATION1 2.000E+00,1.000E+00

## 5.12 GONogo グループ

### **:GONogo:ZPARAMeter:SElect<x>: PARAMeter:MEASure:STATistics**

機能 メジャー判定の統計値を設定 / 問い合わせします。

構文 :GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:MEASure:  
STATistics {MAXimum|MEAN|MINimum|  
SIGMa}  
:GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:MEASure:STATistics?  
<x> = 1 ~ 4

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
MEASURE:STATISTICS MAXIMUM  
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
MEASURE:STATISTICS? -> :GONOGO:  
ZPARAMETER:SELECT1:PARAMETER:  
MEASURE:STATISTICS MAXIMUM

解説 「:MEASURE:MODE CYCLE」のときに有効です。

### **:GONogo:ZPARAMeter:SElect<x>: PARAMeter:MEASure:TRACe<x>?**

機能 メジャー判定の各トレースに関するすべての設定値を問い合わせます。

構文 :GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:MEASure:TRACe<x>?  
SElect<x>の<x> = 1 ~ 4  
TRACe<x>の<x> = 1 ~ 8

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
MEASURE:TRACE1? -> :GONOGO:  
ZPARAMETER:SELECT1:PARAMETER:  
MEASURE:TRACE1:AREA1:TYPE:  
MAXIMUM 2.000E+00,1.000E+00

### **:GONogo:ZPARAMeter:SElect<x>: PARAMeter:MEASure:TRACe<x>:AREA<x>?**

機能 メジャー判定の各エリアに関するすべての設定値を問い合わせます。

構文 :GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:MEASure:TRACe<x>:AREA<x>?  
SElect<x>の<x> = 1 ~ 4  
TRACe<x>の<x> = 1 ~ 8  
AREA<x>の<x> = 1, 2

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
MEASURE:TRACE1:AREA1? -> :GONOGO:  
ZPARAMETER:SELECT1:PARAMETER:  
MEASURE:TRACE1:AREA1:TYPE:  
MAXIMUM 2.000E+00,1.000E+00

### **:GONogo:ZPARAMeter:SElect<x>: PARAMeter:MEASure:TRACe<x>:AREA<x>: TYPE?**

機能 メジャー判定の波形/パラメータに関するすべての設定値を問い合わせます。

構文 :GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:MEASure:TRACe<x>:AREA<x>:  
TYPE?  
SElect<x>の<x> = 1 ~ 4  
TRACe<x>の<x> = 1 ~ 8  
AREA<x>の<x> = 1, 2

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
MEASURE:TRACE1:AREA1:TYPE?  
-> :GONOGO:ZPARAMETER:SELECT1:  
PARAMETER:MEASURE:TRACE1:AREA1:TYPE:  
MAXIMUM 2.000E+00,1.000E+00

### **:GONogo:ZPARAMeter:SElect<x>: PARAMeter:MEASure:TRACe<x>:AREA<x>: TYPE:<パラメータ>**

機能 メジャー判定の波形/パラメータの上下限值を設定 / 問い合わせします。

構文 :GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:MEASure:TRACe<x>:AREA<x>:  
TYPE:<パラメータ> {<NRF>,<NRF>|<電圧>,<電圧>|<電流>,<電流>|<時間>,<時間>|<周波数>,<周波数>}  
:GONogo:ZPARAMeter:SElect<x>:  
PARAMeter:MEASure:TRACe<x>:AREA<x>:  
TYPE:<パラメータ>?

SElect<x>の<x> = 1 ~ 4  
TRACe<x>の<x> = 1 ~ 8  
AREA<x>の<x> = 1, 2  
<パラメータ> = {BURSt|CMean|COUNT|CRMS|  
CSDeviation|DElay|DT|DUTYcycle|FALL|  
FREQuency|HIGH|HILow|LOW|MAXimum|MEAN|  
MINimum|NOVershoot|NWDth|PERFrequency|  
PERiod|POVershoot|PTOPeak|PWIDth|RISE|  
RMS|SDEviation|TYCInteg|TYINteg|V1|V2}  
<NRF>、<電圧>、<電流>、<時間>、<周波数> = 本体ユーザーズマニュアル参照。

例 (以下は、トレース 1、エリア 1 の最大値についての例です。)

```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
MEASURE:TRACE1:AREA1:TYPE:MAXIMUM 1,2
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
MEASURE:TRACE1:AREA1:TYPE:MAXIMUM? ->
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
MEASURE:TRACE1:AREA1:TYPE:
MAXIMUM 2.000E+00,1.000E+00
```

解説 「:MEASURE:MODE CYCLE」のときは、周期統計処理 (Cycle Statistics) の対象です。

**:GONogo:ZPARAMeter:SElect<x>:****PARAMeter:XY<x>?**

機能 各 XY 判定に関するすべての設定値を問い合わせます。

構文 :GONogo:ZPARAMeter:SElect<x>:

PARAMeter:XY<x>?

SElect<x> の <x> = 1 ~ 4

XY<x> の <x> = 1, 2

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
XY1? -> :GONOGO:ZPARAMETER:SELECT1:  
PARAMETER:XY1:  
XYINTEG 2.000E+00,1.000E+00

**:GONogo:ZPARAMeter:SElect<x>:****PARAMeter:XY<x>:XYINteG**

機能 各 XY 判定の積分値の上下限值を設定 / 問い合わせします。

構文 :GONogo:ZPARAMeter:SElect<x>:

PARAMeter:XY<x>:XYINteG {<NRf>,<NRf>}

:GONogo:ZPARAMeter:SElect<x>:

PARAMeter:XY<x>:XYINteG?

SElect<x> の <x> = 1 ~ 4

XY<x> の <x> = 1, 2

<NRf> = - 4 ~ 4(div)

例 :GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
XY1:XYINTEG 1,2  
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:  
XY1:XYINTEG? -> :GONOGO:ZPARAMETER:  
SELECT1:PARAMETER:XY1:  
XYINTEG 2.000E+00,1.000E+00

**:GONogo:ZPARAMeter:SElect<x>:****RECTangle?**

機能 Rectangle 判定に関するすべての設定値を問い合わせます。

構文 :GONogo:ZPARAMeter:SElect<x>:

RECTangle?

<x> = 1 ~ 4

例 :GONOGO:ZPARAMETER:SELECT1:RECTANGLE?  
-> :GONOGO:ZPARAMETER:SELECT1:  
RECTANGLE:HORIZONTAL 1.000E+00,  
0.000E+00;VERTICAL  
1.000E+00,0.000E+00

**:GONogo:ZPARAMeter:SElect<x>:****RECTangle:HORizontal**

機能 Rectangle 判定で用いる四角形の水平位置を設定 / 問い合わせします。

構文 :GONogo:ZPARAMeter:SElect<x>:

RECTangle:HORizontal {<NRf>,<NRf>}

:GONogo:ZPARAMeter:SElect<x>:

RECTangle:HORizontal?

<x> = 1 ~ 4

<NRf> = - 5 ~ 5(div)

例 :GONOGO:ZPARAMETER:SELECT1:RECTANGLE:  
HORIZONTAL 0,1  
:GONOGO:ZPARAMETER:SELECT1:RECTANGLE:  
HORIZONTAL? -> :GONOGO:ZPARAMETER:  
SELECT1:RECTANGLE:  
HORIZONTAL 1.000E+00,0.000E+00

**:GONogo:ZPARAMeter:SElect<x>:****RECTangle:VERTical**

機能 Rectangle 判定で用いる四角形の垂直位置を設定 / 問い合わせします。

構文 :GONogo:ZPARAMeter:SElect<x>:

RECTangle:VERTical {<NRf>,<NRf>}

:GONogo:ZPARAMeter:SElect<x>:

RECTangle:VERTical?

<x> = 1 ~ 4

<NRf> = - 4 ~ 4(div)

例 :GONOGO:ZPARAMETER:SELECT1:RECTANGLE:  
VERTICAL 0,1  
:GONOGO:ZPARAMETER:SELECT1:RECTANGLE:  
VERTICAL? -> :GONOGO:ZPARAMETER:  
SELECT1:RECTANGLE:  
VERTICAL 1.000E+00,0.000E+00

**:GONogo:ZPARAMeter:SElect<x>:TRACe**

機能 ゾーン / パラメータ判定の対象トレースを設定 / 問い合わせします。

構文 :GONogo:ZPARAMeter:SElect<x>:TRACe

{<NRf>}

:GONogo:ZPARAMeter:SElect<x>:TRACe?

<x> = 1 ~ 4

<NRf> = 1 ~ 8

例 :GONOGO:ZPARAMETER:SELECT1:TRACE 1  
:GONOGO:ZPARAMETER:SELECT1:TRACE?  
-> :GONOGO:ZPARAMETER:SELECT1:TRACE 1

**:GONogo:ZPARAMeter:SElect<x>:WAVE?**

機能 WAVE 判定に関するすべての設定値を問い合わせます。

構文 :GONogo:ZPARAMeter:SElect<x>:WAVE?

<x> = 1 ~ 4

例 :GONOGO:ZPARAMETER:SELECT1:WAVE?

-> :GONOGO:ZPARAMETER:SELECT1:WAVE:

TRANGE 2.000E+00,1.000E+00

## 5.12 GONogo グループ

### :GONogo:ZPARAMeter:SElect<x>:WAVE: EDIT<x>:EXIT

機能 WAVE 判定のゾーンの編集メニューから抜けま  
す。

構文 :GONogo:ZPARAMeter:SElect<x>:WAVE:  
EDIT<x>:EXIT {<Nrf>|QUIT}  
SElect<x> の <x> = 1 ~ 4  
EDIT<x> の <x> = 1 ~ 13(1 ~ 8 は各トレース、  
9 ~ 13 は各内部メモリ)  
<Nrf> = 1 ~ 4(各内部メモリ)

例 :GONOGO:ZPARAMETER:SELECT1:WAVE:  
EDIT1:EXIT 1

解説

- ・ゾーン編集中外は、このコマンドはエラーに  
なります。
- ・<Nrf> で指定した内部メモリにゾーン波形を保  
存します。

### :GONogo:ZPARAMeter:SElect<x>:WAVE: EDIT<x>:PART

機能 WAVE 判定のゾーン部分編集をします。

構文 :GONogo:ZPARAMeter:SElect<x>:WAVE:  
EDIT<x>:PART {<Nrf>,<Nrf>,<Nrf>,  
<Nrf>}  
SElect<x> の <x> = 1 ~ 4  
EDIT<x> の <x> = 1 ~ 13(1 ~ 8 は各トレース、  
9 ~ 12 は各内部メモリ、13 は現在表示されてい  
るゾーン波形が編集対象)  
<Nrf> = -5 ~ 5(div:部分編集時間軸カーソル 1、  
2)  
-8 ~ 8(div:上下)(部分編集時間軸カーソル 1、  
部分編集時間軸カーソル 2、↑、↓の順)

例 :GONOGO:ZPARAMETER:SELECT1:WAVE:  
EDIT1:PART 1,2,3,4

解説 上下移動量は、現在表示されているゾーン波形を  
基準にした相対値です。

### :GONogo:ZPARAMeter:SElect<x>:WAVE: EDIT<x>:WHOLE

機能 波形ゾーンの全体ゾーン編集を設定します。

構文 :GONogo:ZPARAMeter:SElect<x>:WAVE:  
EDIT<x>:WHOLE {<Nrf>,<Nrf>,<Nrf>,  
<Nrf>}  
SElect<x> の <x> = 1 ~ 4  
EDIT<x> の <x> = 1 ~ 13(1 ~ 8 は各トレース、  
9 ~ 12 は各内部メモリ、13 は現在表示されてい  
るゾーン波形が編集対象)  
<Nrf> = 0 ~ 5(div:左右)  
0 ~ 8(div:上下)  
(←、→、↑、↓の順)

例 :GONOGO:ZPARAMETER:SELECT1:WAVE:  
EDIT1:WHOLE 1,2,3,4

解説 各移動量は、基準波形を基準にした相対値です。

### :GONogo:ZPARAMeter:SElect<x>:WAVE: TRANge

機能 各ゾーン判定の判定区間を設定 / 問い合わせしま  
す。

構文 :GONogo:ZPARAMeter:SElect<x>:WAVE:  
TRANge {<Nrf>,<Nrf>}  
:GONogo:ZPARAMeter:SElect<x>:WAVE:  
TRANge?  
<x> = 1 ~ 4  
<Nrf> = -5 ~ 5(div)

例 :GONOGO:ZPARAMETER:SELECT1:WAVE:  
TRANGE 1,2  
:GONOGO:ZPARAMETER:SELECT1:WAVE:  
TRANGE? -> :GONOGO:ZPARAMETER:  
SELECT1:WAVE:  
TRANGE 2.000E+00,1.000E+00

### :GONogo:ZPARAMeter:SElect<x>:WINDow

機能 各ゾーン判定の対象ウインドウを設定 / 問い合わ  
せします。

構文 :GONogo:ZPARAMeter:SElect<x>:WINDow  
{MAIN|XY1|XY2|Z1|Z2}  
:GONogo:ZPARAMeter:SElect<x>:WINDow?  
<x> = 1 ~ 4

例 :GONOGO:ZPARAMETER:SELECT1:WINDOW  
MAIN  
:GONOGO:ZPARAMETER:SELECT1:WINDOW?  
-> :GONOGO:ZPARAMETER:SELECT1:WINDOW  
MAIN

## 5.13 Hcopy グループ

### :HCOpy? (Hard COPY)

機能 画面データの出力に関するすべての設定値を問い合わせます。

構文 :HCOpy?

例 :HCOpy? -> :HCOpy:  
DIRECTION EXTPRINTER;EXTPRINTER:  
TONE 1;TYPE EINKJET;;HCOpy:FILE:  
FORMAT BMP;SAVE:ANAMING DATE;  
CDIRECTORY "\Flash Mem\DIR1\DIR2";  
NAME "SAMPLE";:HCOpy:FILE:  
TONE COLOR;;HCOpy:NETPRINT:TONE 1;  
TYPE HINKJET;;HCOpy:PRINTER:HRMODE 1

### :HCOpy:ABORT

機能 データ出力、紙送りを中止します。

構文 :HCOpy:ABORT

例 :HCOpy:ABORT

解説 「:HCOpy:DIRECTION EXTPrinter|PRINter」のときに有効です。

### :HCOpy:DIRection

機能 データ出力先を設定 / 問い合わせします。

構文 :HCOpy:DIRection {EXTPrinter|FILE|  
NETPrint|PRINter}  
:HCOpy:DIRection?

例 :HCOpy:DIRECTION EXTPRINTER  
:HCOpy:DIRECTION? -> :HCOpy:  
DIRECTION EXTPRINTER

### :HCOpy:EXECute

機能 データ出力を実行します。オーバーラップコマンドです。

構文 :HCOpy:EXECute {<文字列>}  
<文字列> = 259文字以内

例 :HCOpy:EXECUTE  
:HCOpy:EXECUTE "\Flash Mem\DIR\DATA"  
(絶対パス指定)  
:HCOpy:EXECUTE "DATA"  
(相対パス指定)

解説

- ・パスを指定しないときは、「:HCOpy:FILE:SAVE:NAME」で指定したファイル名で保存します。
- ・パスを指定したときは、ファイル名の自動作成機能は動作しません。
- ・<文字列> に関するの詳細は、「:HCOpy:FILE:SAVE:CDIRECTory」の解説を参照してください。

### :HCOpy:EXTPrinter?

機能 外部プリンタ出力に関するすべての設定値を問い合わせます。

構文 :HCOpy:EXTPrinter?

例 :HCOpy:EXTPRINTER? -> :HCOpy:  
EXTPRINTER:TONE 1;TYPE EINKJET

### :HCOpy:EXTPrinter:TONE

機能 外部プリンタ出力のハーフトーンを設定 / 問い合わせします。

構文 :HCOpy:EXTPrinter:TONE {<Boolean>}  
:HCOpy:EXTPrinter:TONE?

例 :HCOpy:EXTPRINTER:TONE ON  
:HCOpy:EXTPRINTER:TONE? -> :HCOpy:  
EXTPRINTER:TONE 1

### :HCOpy:EXTPrinter:TYPE

機能 外部プリンタ出力コマンドの種類を設定 / 問い合わせします。

構文 :HCOpy:EXTPrinter:TYPE {EINKjet|  
HINKjet}  
:HCOpy:EXTPrinter:TYPE?

例 :HCOpy:EXTPRINTER:TYPE EINKJET  
:HCOpy:EXTPRINTER:TYPE? -> :HCOpy:  
EXTPRINTER:TYPE EINKJET

### :HCOpy:FILE?

機能 ファイル出力に関するすべての設定値を問い合わせます。

構文 :HCOpy:FILE?

例 :HCOpy:FILE? -> :HCOpy:FILE:  
FORMAT BMP;SAVE:ANAMING DATE;  
CDIRECTORY "\Flash Mem\DIR1\DIR2";  
NAME "SAMPLE";:HCOpy:FILE:TONE COLOR

### :HCOpy:FILE:FORMat

機能 ファイル出力画像フォーマットを設定 / 問い合わせします。

構文 :HCOpy:FILE:FORMat {BMP|JPEG|PNG}  
:HCOpy:FILE:FORMat?

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

### :HCOpy:FILE:SAVE?

機能 ファイル出力の保存に関するすべての設定値を問い合わせます。

構文 :HCOpy:FILE:SAVE?

例 :HCOpy:FILE:SAVE? -> :HCOpy:FILE:  
SAVE:ANAMING DATE;  
CDIRECTORY "\Flash Mem\DIR1\DIR2";  
NAME "SAMPLE"



## 5.13 Hcopy グループ

### **:HCOpy:FILE:SAVE:ANAMing**

機能 保存ファイル名の自動作成の種類を設定 / 問い合わせします。

構文 :HCOpy:FILE:SAVE:ANAMing {DATE|NUMBERing|OFF}  
:HCOpy:FILE:SAVE:ANAMing?

例 :HCOpy:FILE:SAVE:ANAMING DATE  
:HCOpy:FILE:SAVE:ANAMING? -> :HCOpy:FILE:SAVE:ANAMING DATE

### **:HCOpy:FILE:SAVE:CDIRECTory (Change Directory)**

機能 保存ディレクトリ名を設定 / 問い合わせします。

構文 :HCOpy:FILE:SAVE:CDIRECTory {<文字列>}  
:HCOpy:FILE:SAVE:CDIRECTory?

<文字列> = 259文字以内  
例 :HCOpy:FILE:SAVE:CDIRECTORY "\Flash Mem\DIR1\DIR2"

(絶対パス指定)

:HCOpy:FILE:SAVE:CDIRECTORY "DIR2"

(相対パス指定)

:HCOpy:FILE:SAVE:CDIRECTORY "\"

(ルートディレクトリ指定)

:HCOpy:FILE:SAVE:CDIRECTORY? -> :

HCOpy:FILE:SAVE:CDIRECTORY "\Flash Mem\DIR1\DIR2"

解説

- ・ ルートディレクトリ直下は、ドライブ指定ディレクトリです。  
選択できるドライブは、以下の5つです。
- ・ 内蔵ハードディスク: "HD"
- ・ 内部メモリ: "Flash Mem"
- ・ ネットワークドライブ: "Network"
- ・ PCカード: "Storage Card<x>"
- ・ USBストレージ: "USB Storage<x>"  
<x> = 1~4(但し、サポート範囲は "Storage Card" と "USB Storage" を合計して4つまでです(パーティション含む))
- ・ 指定したディレクトリをセーブ、ロードのレントディレクトリに設定します。
- ・ 絶対パス、相対パス指定が可能です。
- ・ 絶対パスで指定する場合は先頭に "\" を付けます。
- ・ 上位の階層への相対パス指定はできません。

### **:HCOpy:FILE:SAVE:NAME**

機能 保存ファイル名を設定 / 問い合わせします。

構文 :HCOpy:FILE:SAVE:NAME <Filename>  
:HCOpy:FILE:SAVE:NAME?

例 :HCOpy:FILE:SAVE:NAME "SAMPLE"  
:HCOpy:FILE:SAVE:NAME? -> :HCOpy:FILE:SAVE:NAME "SAMPLE"

### **:HCOpy:NETPrint?**

機能 ネットワークプリンタ出力に関するすべての設定値を問い合わせします。

構文 :HCOpy:NETPrint?  
例 :HCOpy:NETPRINT? -> :HCOpy:NETPRINT:TONE 1;TYPE HINKJET

### **:HCOpy:NETPrint:TONE**

機能 ネットワークプリンタ出力のハーフトーンを設定 / 問い合わせします。

構文 :HCOpy:NETPrint:TONE {<Boolean>}  
:HCOpy:NETPrint:TONE?

例 :HCOpy:NETPRINT:TONE ON  
:HCOpy:NETPRINT:TONE? -> :HCOpy:NETPRINT:TONE 1

解説 「:HCOpy:NETPrint:TYPE HLASer」のときは {ON} が無効です。

### **:HCOpy:NETPrint:TYPE**

機能 ネットワークプリンタ出力コマンドの種類を設定 / 問い合わせします。

構文 :HCOpy:NETPrint:TYPE {HINKjet|HLASer}  
:HCOpy:NETPrint:TYPE?

例 :HCOpy:NETPRINT:TYPE HINKJET  
:HCOpy:NETPRINT:TYPE? -> :HCOpy:NETPRINT:TYPE HINKJET

### **:HCOpy:PRINter?**

機能 内蔵プリンタ出力に関するすべての設定値を問い合わせます。

構文 :HCOpy:PRINter?

例 :HCOpy:PRINTER? -> :HCOpy:PRINTER:HRMODE 1

### **:HCOpy:PRINter:HRMode**

機能 内蔵プリンタ出力の高分解能モードのON/OFFを設定 / 問い合わせします。

構文 :HCOpy:PRINter:HRMode {<Boolean>}  
:HCOpy:PRINter:HRMode?

例 :HCOpy:PRINTER:HRMODE ON  
:HCOpy:PRINTER:HRMODE? -> :HCOpy:PRINTER:HRMODE 1

## 5.14 HISTory グループ

### :HISTory?

機能 ヒストリ機能に関するすべての設定を問い合わせます。

構文 :HISTory?

例 :HISTORY? -> :HISTORY:CURRENT:  
DISPLAY 0,-10;DMODE ONE;MODE RECORD;  
RECORD 0;REPLAY:SPEED 1;:HISTORY:  
CURRENT:SEARCH:LOGIC AND;SELECT1:  
CONDITION IN;MODE PARAMETER;  
PARAMETER:CATEGORY MEASURE;MEASURE:  
CALCULATION1 1.000E+00,2.000E+00;;  
HISTORY:CURRENT:SEARCH:SELECT1:  
RECTANGLE:HORIZONTAL 1.000E+00,  
2.000E+00;VERTICAL 1.000E+00,  
2.000E+00;;:HISTORY:CURRENT:SEARCH:  
SELECT1:TRACE 1;:HISTORY:CURRENT:  
SEARCH:SELECT1:WAVE:TRANGE 1.000E+00,  
2.000E+00;;:HISTORY:CURRENT:SEARCH:  
SELECT1:WINDOW MAIN;;:HISTORY:CURRENT:  
SEARCH:SELECT2:CONDITION IN;  
MODE PARAMETER;PARAMETER:  
CATEGORY MEASURE;MEASURE:  
CALCULATION1 1.000E+00,2.000E+00;;  
HISTORY:CURRENT:SEARCH:SELECT2:  
RECTANGLE:HORIZONTAL 1.000E+00,  
2.000E+00;VERTICAL 1.000E+00,  
2.000E+00;;:HISTORY:CURRENT:SEARCH:  
SELECT2:TRACE 1;:HISTORY:CURRENT:  
SEARCH:SELECT2:WAVE:TRANGE 1.000E+00,  
2.000E+00;;:HISTORY:CURRENT:SEARCH:  
SELECT2:WINDOW MAIN;;:HISTORY:CURRENT:  
SEARCH;SELECT3:CONDITION IN;  
MODE PARAMETER;PARAMETER:  
CATEGORY MEASURE;MEASURE:  
CALCULATION1 1.000E+00,2.000E+00.....  
.

### :HISTory:CURRent?

機能 カレント波形 (CH1 ~ 4、M1 ~ 8) のヒストリ機能に関するすべての設定を問い合わせます。

構文 :HISTory:CURRent?

例 :HISTORY:CURRENT? -> :HISTORY:  
CURRENT:DISPLAY 0,-10;DMODE ONE;  
MODE RECORD;RECORD 0;REPLAY:SPEED 1;;  
HISTORY:CURRENT:SEARCH:LOGIC AND;  
SELECT1:CONDITION IN;MODE PARAMETER;  
PARAMETER:CATEGORY MEASURE;MEASURE:  
CALCULATION1 1.000E+00,2.000E+00;;  
HISTORY:CURRENT:SEARCH:SELECT1:  
RECTANGLE:HORIZONTAL 1.000E+00,  
2.000E+00;VERTICAL 1.000E+00,  
2.000E+00;;:HISTORY:CURRENT:SEARCH:  
SELECT1:TRACE 1;:HISTORY:CURRENT:  
SEARCH:SELECT1:WAVE:TRANGE 1.000E+00,  
2.000E+00;;:HISTORY:CURRENT:SEARCH:  
SELECT1:WINDOW MAIN;;:HISTORY:CURRENT:  
SEARCH:SELECT2:CONDITION IN;  
MODE PARAMETER;PARAMETER:  
CATEGORY MEASURE;MEASURE:  
CALCULATION1 1.000E+00,2.000E+00;;  
HISTORY:CURRENT:SEARCH:SELECT2:  
RECTANGLE:HORIZONTAL 1.000E+00,  
2.000E+00;VERTICAL 1.000E+00,  
2.000E+00;;:HISTORY:CURRENT:SEARCH:  
SELECT2:TRACE 1;:HISTORY:CURRENT:  
SEARCH:SELECT2:WAVE:TRANGE 1.000E+00,  
2.000E+00;;:HISTORY:CURRENT:SEARCH:  
SELECT2:WINDOW MAIN;;:HISTORY:CURRENT:  
SEARCH;SELECT3:CONDITION IN;  
MODE PARAMETER;PARAMETER:  
CATEGORY MEASURE;MEASURE:  
CALCULATION1 1.000E+00,2.000E+00.....  
.

### :HISTory[:CURRent]:DISPlay

機能 ヒストリ波形の表示レコードの開始番号と終了番号を設定 / 問い合わせします。

構文 :HISTory[:CURRent]:DISPlay

{<NRf>,<NRf>}

:HISTory[:CURRent]:DISPlay?

<NRf> = 本体ユーザズマニュアル参照。

例 :HISTORY:CURRENT:DISPLAY 0,-10

:HISTORY:CURRENT:DISPLAY?

-> :HISTORY:CURRENT:DISPLAY 0,-10

## 5.14 HISTory グループ

### **:HISTory[:CURRENT]:DMODE (Display Mode)**

機能 ヒストリ波形の表示モードを設定 / 問い合わせします。

構文 :HISTory[:CURRENT]:DMODE {ACOLor|AHTone|AINTensity|ONE}  
:HISTory[:CURRENT]:DMODE?

例 :HISTORY:CURRENT:DMODE ONE  
:HISTORY:CURRENT:DMODE? -> :HISTORY:CURRENT:DMODE ONE

### **:HISTory[:CURRENT]:MODE**

機能 ヒストリ波形のハイライト表示モードを設定 / 問い合わせします。

構文 :HISTory[:CURRENT]:MODE {AVERage|RECORD}  
:HISTory[:CURRENT]:MODE?

例 :HISTORY:CURRENT:MODE RECORD  
:HISTORY:CURRENT:MODE? -> :HISTORY:CURRENT:MODE RECORD

### **:HISTory[:CURRENT]:RECORD**

機能 ヒストリ波形の対象レコードを設定 / 問い合わせします。

構文 :HISTory[:CURRENT]:RECORD {<NRf>|MINimum}  
:HISTory[:CURRENT]:RECORD?  
<NRf> = 本体ユーザーズマニュアル参照。

例 :HISTORY:CURRENT:RECORD 0  
:HISTORY:CURRENT:RECORD? -> :HISTORY:CURRENT:RECORD 0

解説 「MINimum」を指定すると、最小のレコード番号になります。

### **:HISTory[:CURRENT]:RECORD? MINimum**

機能 ヒストリ波形の最小のレコード番号を問い合わせます。

構文 :HISTory[:CURRENT]:RECORD? MINimum

例 :HISTORY:CURRENT:RECORD? MINIMUM  
-> :HISTORY:CURRENT:RECORD -1

解説 「MINimum」を指定すると、最小のレコード番号になります。

### **:HISTory[:CURRENT]:REPLAY?**

機能 ヒストリ波形のリプレイ機能に関するすべての設定を問い合わせます。

構文 :HISTory[:CURRENT]:REPLAY?

例 :HISTORY:CURRENT:REPLAY? -> :HISTORY:CURRENT:REPLAY:SPEED 1

### **:HISTory[:CURRENT]:REPLAY:JUMP**

機能 ヒストリ波形を指定のレコード番号へジャンプさせます。

構文 :HISTory[:CURRENT]:REPLAY:JUMP {MAXimum|MINimum}

例 :HISTORY:CURRENT:REPLAY:JUMP MAXIMUM

### **:HISTory[:CURRENT]:REPLAY:SPEED**

機能 ヒストリ波形のリプレイ速度を設定 / 問い合わせします。

構文 :HISTory[:CURRENT]:REPLAY:SPEED {<NRf>|PER3|PER10|PER30|PER60}  
:HISTory[:CURRENT]:REPLAY:SPEED?  
<NRf> = 1、3、10

例 :HISTORY:CURRENT:REPLAY:SPEED 1  
:HISTORY:CURRENT:REPLAY:SPEED?  
-> :HISTORY:CURRENT:REPLAY:SPEED 1

### **:HISTory[:CURRENT]:REPLAY:START**

機能 ヒストリ波形のリプレイを指定方向へ開始します。

構文 :HISTory[:CURRENT]:REPLAY:START {MAXimum|MINimum}

例 :HISTORY:CURRENT:REPLAY:START MAXIMUM

### **:HISTory[:CURRENT]:REPLAY:STOP**

機能 ヒストリ波形のリプレイを停止します。

構文 :HISTory[:CURRENT]:REPLAY:STOP

例 :HISTORY:CURRENT:REPLAY:STOP

**:HISTORY[:CURRENT]:SEARCH?**

機能 ヒストリサーチ機能に関するすべての設定を問い合わせます。

構文 :HISTORY[:CURRENT]:SEARCH?  
 例 :HISTORY:CURRENT:SEARCH? -> :HISTORY:CURRENT:SEARCH:LOGIC AND;  
 SELECT1:CONDITION IN;MODE PARAMETER;  
 PARAMETER:CATEGORY MEASURE;MEASURE:  
 CALCULATION1 1.000E+00,2.000E+00;;  
 HISTORY:CURRENT:SEARCH:SELECT1:  
 RECTANGLE:HORIZONTAL 1.000E+00,  
 2.000E+00;VERTICAL 1.000E+00,  
 2.000E+00;;HISTORY:CURRENT:SEARCH:  
 SELECT1:TRACE 1;;HISTORY:CURRENT:  
 SEARCH:SELECT1:WAVE:TRANGE 1.000E+00,  
 2.000E+00;;HISTORY:CURRENT:SEARCH:  
 SELECT1:WINDOW MAIN;;HISTORY:CURRENT:  
 SEARCH:SELECT2:CONDITION IN;  
 MODE PARAMETER;PARAMETER:  
 CATEGORY MEASURE;MEASURE:  
 CALCULATION1 1.000E+00,2.000E+00;;  
 HISTORY:CURRENT:SEARCH:SELECT2:  
 RECTANGLE:HORIZONTAL 1.000E+00,  
 2.000E+00;VERTICAL 1.000E+00,  
 2.000E+00;;HISTORY:CURRENT:SEARCH:  
 SELECT2:TRACE 1;;HISTORY:CURRENT:  
 SEARCH:SELECT2:WAVE:TRANGE 1.000E+00,  
 2.000E+00;;HISTORY:CURRENT:SEARCH:  
 SELECT2:WINDOW MAIN;;HISTORY:CURRENT:  
 SEARCH;SELECT3:CONDITION IN;  
 MODE PARAMETER;PARAMETER:  
 CATEGORY MEASURE;MEASURE:  
 CALCULATION1 1.000E+00,2.000E+00.....  
 .

**:HISTORY[:CURRENT]:SEARCH:ABORT**

機能 ヒストリサーチを中止します。

構文 :HISTORY[:CURRENT]:SEARCH:ABORT  
 例 :HISTORY:CURRENT:SEARCH:ABORT

**:HISTORY[:CURRENT]:SEARCH:EXECUTE**

機能 ヒストリサーチを実行します。オーバーラップコマンドです。

構文 :HISTORY[:CURRENT]:SEARCH:EXECUTE  
 例 :HISTORY:CURRENT:SEARCH:EXECUTE

**:HISTORY[:CURRENT]:SEARCH:LOGIC**

機能 ヒストリサーチのロジックを設定 / 問い合わせします。

構文 :HISTORY[:CURRENT]:SEARCH:  
 LOGIC {AND|OR}  
 例 :HISTORY[:CURRENT]:SEARCH:LOGIC?  
 :HISTORY:CURRENT:SEARCH:LOGIC AND  
 :HISTORY:CURRENT:SEARCH:LOGIC?  
 -> :HISTORY:CURRENT:SEARCH:LOGIC AND

**:HISTORY[:CURRENT]:SEARCH:RESET**

機能 ヒストリサーチのサーチ条件をリセットします。

構文 :HISTORY[:CURRENT]:SEARCH:RESET  
 例 :HISTORY:CURRENT:SEARCH:RESET

**:HISTORY[:CURRENT]:SEARCH:SELECT<x>?**

機能 各ヒストリサーチ条件に関するすべての設定値を問い合わせます。

構文 :HISTORY[:CURRENT]:SEARCH:  
 SELECT<x>?  
 <x> = 1 ~ 4  
 例 :HISTORY:CURRENT:SEARCH:SELECT1?  
 -> :HISTORY:CURRENT:SEARCH:SELECT1:  
 CONDITION IN;MODE PARAMETER;  
 PARAMETER:CATEGORY MEASURE;MEASURE:  
 CALCULATION1 1.000E+00,2.000E+00;;  
 HISTORY:CURRENT:SEARCH:SELECT1:  
 RECTANGLE:HORIZONTAL 1.000E+00,  
 2.000E+00;VERTICAL 1.000E+00,  
 2.000E+00;;HISTORY:CURRENT:SEARCH:  
 SELECT1:TRACE 1;;HISTORY:CURRENT:  
 SEARCH:SELECT1:WAVE:TRANGE 1.000E+00,  
 2.000E+00;;HISTORY:CURRENT:SEARCH:  
 SELECT1:WINDOW MAIN

**:HISTORY[:CURRENT]:SEARCH:SELECT<x>:CONDITION**

機能 各ヒストリサーチ条件の判定基準を設定 / 問い合わせします。

構文 :HISTORY[:CURRENT]:SEARCH:  
 SELECT<x>:CONDITION {DONTcare|IN|OUT}  
 :HISTORY[:CURRENT]:SEARCH:  
 SELECT<x>:CONDITION?  
 <x> = 1 ~ 4  
 例 :HISTORY:CURRENT:SEARCH:SELECT1:  
 CONDITION IN  
 :HISTORY:CURRENT:SEARCH:SELECT1:  
 CONDITION? -> :HISTORY:CURRENT:  
 SEARCH:SELECT1:CONDITION IN

**:HISTORY[:CURRENT]:SEARCH:SELECT<x>:MODE**

機能 各ヒストリサーチ条件のモードを設定 / 問い合わせします。

構文 :HISTORY[:CURRENT]:SEARCH:  
 SELECT<x>:MODE {PARAMETER|POLYGON|  
 RECTANGLE|WAVE}  
 :HISTORY[:CURRENT]:SEARCH:  
 SELECT<x>:MODE?  
 <x> = 1 ~ 4  
 例 :HISTORY:CURRENT:SEARCH:SELECT1:  
 MODE WAVE  
 :HISTORY:CURRENT:SEARCH:SELECT1:MODE?  
 -> :HISTORY:CURRENT:SEARCH:SELECT1:  
 MODE WAVE

## 5.14 HISTory グループ

### **:HISTory[:CURRENT][:SEARCH]:**

#### **SElect<x>:PARAMeter?**

機能 各ヒストリサーチ条件のパラメータに関する設定値をすべて問い合わせます。

構文 :HISTory[:CURRENT][:SEARCH]:

SElect<x>:

PARAMeter?

<x> = 1 ~ 4

例

```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER? -> :HISTORY:CURRENT:
SEARCH:SELECT1:PARAMETER:
CATEGORY MEASURE;MEASURE:
CALCULATION1 2.000E+00,1.000E+00
```

### **:HISTory[:CURRENT][:SEARCH]:**

#### **SElect<x>:PARAMeter:CATegory**

機能 パラメータのカテゴリを設定 / 問い合わせします。

構文 :HISTory[:CURRENT][:SEARCH]:

SElect<x>:PARAMeter:

CATegory {FFT|MEASure|XY}

:HISTory[:CURRENT][:SEARCH]:

SElect<x>:PARAMeter:CATegory?

<x> = 1 ~ 4

例

```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:CATEGORY MEASURE
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:CATEGORY? -> :HISTORY:
CURRENT:SEARCH:SELECT1:PARAMETER:
CATEGORY MEASURE
```

### **:HISTory[:CURRENT][:SEARCH]:S**

#### **Elect<x>:PARAMeter:FFT<x>?**

機能 各 FFT サーチに関するすべての設定値を問い合わせます。

構文 :HISTory[:CURRENT][:SEARCH]:

SElect<x>:PARAMeter:FFT<x>?

SElect<x> の <x> = 1 ~ 4

FFT<x> の <x> = 1、2

例

```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1? -> :HISTORY:CURRENT:
SEARCH:SELECT1:PARAMETER:FFT1:
CALCULATION1 2.000E+00,1.000E+00
```

### **:HISTory[:CURRENT][:SEARCH]:**

#### **SElect<x>:PARAMeter:FFT<x>:**

#### **CALCulation<x>**

機能 各 FFT サーチの各 Calc アイテムの上下限値を設定 / 問い合わせします。

構文 :HISTory[:CURRENT][:SEARCH]:

SElect<x>:PARAMeter:FFT<x>:

CALCulation<x> {<Nrf>,<Nrf>}

:HISTory[:CURRENT][:SEARCH]:

SElect<x>:PARAMeter:FFT<x>:

CALCulation<x>?

SElect<x> の <x> = 1 ~ 4

FFT<x> の <x> = 1、2

CALCulation<x> の <x> = 1 ~ 4

<Nrf> = 本体ユーザーズマニュアル参照。

例

```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:CALCULATION1 1,2
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:CALCULATION1?
-> :HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:
CALCULATION1 2.000E+00,1.000E+00
```

### **:HISTory[:CURRENT][:SEARCH]:**

#### **SElect<x>:PARAMeter:FFT<x>:PEAK?**

機能 各 FFT サーチのピーク値に関するすべての設定値を問い合わせます。

構文 :HISTory[:CURRENT][:SEARCH]:

SElect<x>:

PARAMeter:FFT<x>:PEAK?

SElect<x> の <x> = 1 ~ 4

FFT<x> の <x> = 1、2

例

```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:PEAK? -> :HISTORY:
CURRENT:SEARCH:SELECT1:PARAMETER:
FFT1:PEAK:DV 2.000E+00,1.000E+00
```

**:HISTORY[:CURRENT][:SEARCH]:**  
**SElect<x>:PARAMeter:FFT<x>:PEAK:**  
**DFREquency**  
 機能 各 FFT サーチのピーク周波数間の上下限値を設定 / 問い合わせします。  
 構文 :HISTORY[:CURRENT][:SEARCH]:  
 SElect<x>:PARAMeter:FFT<x>:PEAK:  
 DFREquency {<周波数>, <周波数>}  
 :HISTORY[:CURRENT][:SEARCH]:  
 SElect<x>:PARAMeter:FFT<x>:PEAK:  
 DFREquency?  
 SElect<x> の <x> = 1 ~ 4  
 FFT<x> の <x> = 1, 2  
 <周波数> = 本体ユーザーズマニュアル参照。  
 例 :HISTORY:CURRENT:SEARCH:SELECT1:  
 PARAMETER:FFT1:PEAK:DFREQUENCY 1,10  
 :HISTORY:CURRENT:SEARCH:SELECT1:  
 PARAMETER:FFT1:PEAK:DFREQUENCY?  
 -> :HISTORY:CURRENT:SEARCH:SELECT1:  
 PARAMETER:FFT1:PEAK:  
 DFREQUENCY 10.00E+00,1.000E+00

**:HISTORY[:CURRENT][:SEARCH]:**  
**SElect<x>:PARAMeter:FFT<x>:PEAK:DV**  
 機能 各 FFT サーチのピーク電圧値間の上下限値を設定 / 問い合わせします。  
 構文 :HISTORY[:CURRENT][:SEARCH]:  
 SElect<x>:  
 PARAMeter:FFT<x>:PEAK:DV  
 {<Nrf>, <Nrf>}  
 :HISTORY[:CURRENT][:SEARCH]:  
 SElect<x>:PARAMeter:FFT<x>:PEAK:DV?  
 SElect<x> の <x> = 1 ~ 4  
 FFT<x> の <x> = 1, 2  
 <Nrf> = - 4 ~ 4(div)  
 例 :HISTORY:CURRENT:SEARCH:SELECT1:  
 PARAMETER:FFT1:PEAK:DV 1,2  
 :HISTORY:CURRENT:SEARCH:SELECT1:  
 PARAMETER:FFT1:PEAK:DV? -> :HISTORY:  
 CURRENT:SEARCH:SELECT1:PARAMETER:  
 FFT1:PEAK:DV 2.000E+00,1.000E+00

**:HISTORY[:CURRENT][:SEARCH]:**  
**SElect<x>:PARAMeter:FFT<x>:PEAK:**  
**FREquency<x>**  
 機能 各 FFT サーチの各ピーク周波数の上下限値を設定 / 問い合わせします。  
 構文 :HISTORY[:CURRENT][:SEARCH]:  
 SElect<x>:PARAMeter:FFT<x>:PEAK:  
 FREquency<x> {<周波数>, <周波数>}  
 :HISTORY[:CURRENT][:SEARCH]:  
 SElect<x>:PARAMeter:FFT<x>:PEAK:  
 FREquency<x>?  
 SElect<x> の <x> = 1 ~ 4  
 FFT<x> の <x> = 1, 2  
 FREquency<x> の <x> = 1, 2  
 <周波数> = 本体ユーザーズマニュアル参照。  
 例 :HISTORY:CURRENT:SEARCH:SELECT1:  
 PARAMETER:FFT1:PEAK:FREQUENCY1 1,2  
 :HISTORY:CURRENT:SEARCH:SELECT1:  
 PARAMETER:FFT1:PEAK:FREQUENCY1?  
 -> :HISTORY:CURRENT:SEARCH:SELECT1:  
 PARAMETER:FFT1:PEAK:  
 FREQUENCY1 2.000E+00,1.000E+00

**:HISTORY[:CURRENT][:SEARCH]:**  
**SElect<x>:PARAMeter:FFT<x>:PEAK:V<x>**  
 機能 各 FFT サーチの各ピーク電圧値の上下限値を設定 / 問い合わせします。  
 構文 :HISTORY[:CURRENT][:SEARCH]:  
 SElect<x>:PARAMeter:FFT<x>:PEAK:V<x>  
 {<Nrf>, <Nrf>}  
 :HISTORY[:CURRENT][:SEARCH]:  
 SElect<x>:PARAMeter:FFT<x>:PEAK:V<x>?  
 SElect<x> の <x> = 1 ~ 4  
 FFT<x> の <x> = 1, 2  
 V<x> の <x> = 1, 2  
 <Nrf> = - 4 ~ 4(div)  
 例 :HISTORY:CURRENT:SEARCH:SELECT1:  
 PARAMETER:FFT1:PEAK:V1 1,2  
 :HISTORY:CURRENT:SEARCH:SELECT1:  
 PARAMETER:FFT1:PEAK:V1? -> :HISTORY:  
 CURRENT:SEARCH:SELECT1:PARAMETER:  
 FFT1:PEAK:V1 2.000E+00,1.000E+00

**:HISTORY[:CURRENT][:SEARCH]:**  
**SElect<x>:PARAMeter:MEASure?**  
 機能 波形パラメータの自動測定での検索 (メジャーサーチ) に関するすべての設定値を問い合わせます。  
 構文 :HISTORY[:CURRENT][:SEARCH]:  
 SElect<x>:PARAMeter:MEASure?  
 <x> = 1 ~ 4  
 例 :HISTORY:CURRENT:SEARCH:SELECT1:  
 PARAMETER:MEASURE? -> :HISTORY:  
 CURRENT:SEARCH:SELECT1:PARAMETER:  
 MEASURE:  
 CALCULATION1 2.000E+00,1.000E+00

## 5.14 HISTory グループ

**:HISTory[:CURRENT][:SEARCH]:  
SElect<x>:PARAMeter:MEASure:BIT<x>?**

機能 メジャーサーチの各ロジックビットに関するすべての設定値を問い合わせます。

構文 :HISTory[:CURRENT][:SEARCH]:  
SElect<x>:PARAMeter:MEASure:BIT<x>?  
SElect<x>の<x> = 1 ~ 4  
BIT<x>の<x> = 1 ~ 32(DL9510L、DL9505Lは  
BIT<x>の<x> = 1 ~ 8、17 ~ 24が有効です。)

例 :HISTory:CURRENT:SEARCH:SELECT1:  
PARAMETER:MEASURE:BIT1? -> :HISTory:  
CURRENT:SEARCH:SELECT1:PARAMETER:  
MEASURE:BIT1:AREA1:TYPE:  
COUNT 1.000E+00,0.000E+00

**:HISTory[:CURRENT][:SEARCH]:  
SElect<x>:PARAMeter:MEASure:BIT<x>:  
AREA<x>?**

機能 メジャーサーチの各エリアに関するすべての設定値を問い合わせます。

構文 :HISTory[:CURRENT][:SEARCH]:  
SElect<x>:PARAMeter:MEASure:BIT<x>:  
AREA<x>?  
SElect<x>の<x> = 1 ~ 4  
BIT<x>の<x> = 1 ~ 32(DL9510L、DL9505Lは  
BIT<x>の<x> = 1 ~ 8、17 ~ 24が有効です。)  
AREA<x>の<x> = 1、2

例 :HISTory:CURRENT:SEARCH:SELECT1:  
PARAMETER:MEASURE:BIT1:AREA1?  
-> :HISTory:CURRENT:SEARCH:SELECT1:  
PARAMETER:MEASURE:BIT1:AREA1:TYPE:  
COUNT 1.000E+00,0.000E+00

**:HISTory[:CURRENT][:SEARCH]:  
SElect<x>:PARAMeter:MEASure:BIT<x>:  
AREA<x>:TYPE?**

機能 メジャーサーチのロジック波形パラメータに関するすべての設定値を問い合わせます。

構文 :HISTory[:CURRENT][:SEARCH]:  
SElect<x>:  
PARAMeter:MEASure:BIT<x>:AREA<x>:  
TYPE?  
SElect<x>の<x> = 1 ~ 4  
BIT<x>の<x> = 1 ~ 32(DL9510L、DL9505Lは  
BIT<x>の<x> = 1 ~ 8、17 ~ 24が有効です。)  
AREA<x>の<x> = 1、2

例 :HISTory:CURRENT:SEARCH:SELECT1:  
PARAMETER:MEASURE:BIT1:AREA1:TYPE?  
-> :HISTory:CURRENT:SEARCH:SELECT1:  
PARAMETER:MEASURE:BIT1:AREA1:TYPE:  
COUNT 1.000E+00,0.000E+00

**:HISTory[:CURRENT][:SEARCH]:  
SElect<x>:PARAMeter:MEASure:BIT<x>:  
AREA<x>:TYPE:<パラメータ>**

機能 メジャーサーチのロジック波形パラメータの上下限値を設定 / 問い合わせします。

構文 :HISTory[:CURRENT][:SEARCH]:  
SElect<x>:PARAMeter:MEASure:  
BIT<x>:AREA<x>:TYPE:<パラメータ>  
{(<NRF>,<NRF>) | (<時間>,<時間>) | (<周波数>,<周波数>)}  
:HISTory[:CURRENT][:SEARCH]:  
SElect<x>:PARAMeter:MEASure:BIT<x>:  
AREA<x>:TYPE:<パラメータ>?  
SElect<x>の<x> = 1 ~ 4  
BIT<x>の<x> = 1 ~ 32(DL9510L、DL9505Lは  
BIT<x>の<x> = 1 ~ 8、17 ~ 24が有効です。)  
AREA<x>の<x> = 1、2  
<パラメータ> = {COUNT|DElay|DT|DUTYcycle|  
FREQuency|NWIDth|PERFrequency|  
PERiod|PWIDth}  
<NRF>、<時間>、<周波数> = 本体ユーザーズ  
マニュアル参照。

例 (以下は、トレース 1、エリア 1 のカウントにつ  
いての例です。)  
:HISTory:CURRENT:SEARCH:SELECT1:  
PARAMETER:MEASURE:BIT1:AREA1:TYPE:  
COUNT 0,1  
:HISTory:CURRENT:SEARCH:SELECT1:  
PARAMETER:MEASURE:BIT1:AREA1:TYPE:  
COUNT? -> :HISTory:CURRENT:SEARCH:  
SELECT1:PARAMETER:MEASURE:BIT1:AREA1:  
TYPE:COUNT 1.000E+00,0.000E+00

**:HISTory[:CURRENT][:SEARCH]:  
SElect<x>:PARAMeter:MEASure:  
CALCulation<x>**

機能 メジャーサーチの各 Calc アイテムの上下限値を設定 / 問い合わせします。

構文 :HISTory[:CURRENT][:SEARCH]:  
SElect<x>:PARAMeter:MEASure:  
CALCulation<x> {<NRF>,<NRF>}  
:HISTory[:CURRENT][:SEARCH]:  
SElect<x>:PARAMeter:MEASure:  
CALCulation<x>?  
SElect<x>の<x> = 1 ~ 4  
CALCulation<x>の<x> = 1 ~ 4  
<NRF> = - 4 ~ 4(div)

例 :HISTory:CURRENT:SEARCH:SELECT1:  
PARAMETER:MEASURE:CALCULATION1 1,2  
:HISTory:CURRENT:SEARCH:SELECT1:  
PARAMETER:MEASURE:CALCULATION1?  
-> :HISTory:CURRENT:SEARCH:SELECT1:  
PARAMETER:MEASURE:  
CALCULATION1 2.000E+00,1.000E+00

**:HISTory[:CURRENT][:SEARCH]:**  
**SElect<x>:PARAMeter:MEASure:**  
**TRACe<x>?**  
 機能 メジャーサーチの各トレースに関するすべての設定値を問い合わせます。  
 構文 :HISTory[:CURRENT][:SEARCH]:  
 SElect<x>:PARAMeter:MEASure:TRACe<x>?  
 SElect<x>の<x> = 1 ~ 4  
 TRACe<x>の<x> = 1 ~ 8  
 例 :HISTory:CURRENT:SEARCH:SELECT1:  
 PARAMETER:MEASURE:TRACE1?  
 -> :HISTory:CURRENT:SEARCH:SELECT1:  
 PARAMETER:MEASURE:TRACE1:AREA1:TYPE:  
 BURST 2.000E+00,1.000E+00

**:HISTory[:CURRENT][:SEARCH]:**  
**SElect<x>:PARAMeter:MEASure:**  
**TRACe<x>:AREA<x>?**  
 機能 メジャーサーチの各エリアに関するすべての設定値を問い合わせます。  
 構文 :HISTory[:CURRENT][:SEARCH]:  
 SElect<x>:PARAMeter:MEASure:TRACe<x>:  
 AREA<x>?  
 SElect<x>の<x> = 1 ~ 4  
 TRACe<x>の<x> = 1 ~ 8  
 AREA<x>の<x> = 1, 2  
 例 :HISTory:CURRENT:SEARCH:SELECT1:  
 PARAMETER:MEASURE:TRACE1:AREA1?  
 -> :HISTory:CURRENT:SEARCH:SELECT1:  
 PARAMETER:MEASURE:TRACE1:AREA1:TYPE:  
 BURST 2.000E+00,1.000E+00

**:HISTory[:CURRENT][:SEARCH]:**  
**SElect<x>:PARAMeter:MEASure:**  
**TRACe<x>:AREA<x>:TYPE?**  
 機能 メジャーサーチの波形パラメータに関するすべての設定値を問い合わせます。  
 構文 :HISTory[:CURRENT][:SEARCH]:  
 SElect<x>:PARAMeter:MEASure:TRACe<x>:  
 AREA<x>:TYPE?  
 SElect<x>の<x> = 1 ~ 4  
 TRACe<x>の<x> = 1 ~ 8  
 AREA<x>の<x> = 1, 2  
 例 :HISTory:CURRENT:SEARCH:SELECT1:  
 PARAMETER:MEASURE:TRACE1:AREA1:TYPE?  
 -> :HISTory:CURRENT:SEARCH:SELECT1:  
 PARAMETER:MEASURE:TRACE1:AREA1:TYPE:  
 BURST 2.000E+00,1.000E+00

**:HISTory[:CURRENT][:SEARCH]:**  
**SElect<x>:PARAMeter:MEASure:**  
**TRACe<x>:AREA<x>:TYPE:<パラメータ>**  
 機能 メジャーサーチの波形パラメータの上下限值を設定 / 問い合わせします。  
 構文 :HISTory[:CURRENT][:SEARCH]:  
 SElect<x>:PARAMeter:MEASure:  
 TRACe<x>:AREA<x>:TYPE:<パラメータ>  
 {(<NRf>,<NRf>) | (<電圧>,<電圧>) | (<電流>,<電流>) | (<時間>,<時間>) | (<周波数>,<周波数>)}  
 :HISTory[:CURRENT][:SEARCH]:  
 SElect<x>:PARAMeter:MEASure:TRACe<x>:  
 AREA<x>:TYPE:<パラメータ>?  
 SElect<x>の<x> = 1 ~ 4  
 TRACe<x>の<x> = 1 ~ 8  
 AREA<x>の<x> = 1, 2  
 <パラメータ> = {BURSt|CMEan|COUNT|CRMS|CSDeviation|DElay|DT|DUTYcycle|FALL|FREQuency|HIGH|HILow|LOW|MAXimum|MEAN|MINimum|NOVershoot|NWIDTH|PERFrequency|PERiod|POVershoot|PTOPeak|PWIDTH|RISE|RMS|SDEVIation|TYCInteg|TYINteg|V1|V2|<NRf>、<電圧>、<電流>、<時間>、<周波数> = 本体ユーザーズマニュアル参照。  
 例 (以下は、トレース 1、エリア 1 の最大値についての例です。)  
 :HISTory:CURRENT:SEARCH:SELECT1:  
 PARAMETER:MEASURE:TRACE1:AREA1:TYPE:  
 MAXIMUM 0,1  
 :HISTory:CURRENT:SEARCH:SELECT1:  
 PARAMETER:MEASURE:TRACE1:AREA1:TYPE:  
 MAXIMUM? -> :HISTory:CURRENT:SEARCH:  
 SELECT1:PARAMETER:MEASURE:TRACE1:  
 AREA1:  
 TYPE:MAXIMUM 1.000E+00,0.000E+00



## 5.14 HISTory グループ

### **:HISTory[:CURRENT][:SEARCH]:**

#### **SElect<x>:PARAMeter:XY<x>?**

機能 各 XY サーチに関するすべての設定値を問い合わせます。

構文 :HISTory[:CURRENT][:SEARCH]:

SElect<x>:PARAMeter:XY<x>?

SElect<x> の <x> = 1 ~ 4

XY<x> の <x> = 1、2

例

```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:XY1? -> :HISTORY:CURRENT:
SEARCH:SELECT1:PARAMETER:XY1:
XYINTEG 2.000E+00,1.000E+00
```

### **:HISTory[:CURRENT][:SEARCH]:**

#### **SElect<x>:PARAMeter:XY<x>:XYINteG**

機能 各 XY サーチの積分値の上下限值を設定 / 問い合わせします。

構文 :HISTory[:CURRENT][:SEARCH]:

SElect<x>:PARAMeter:XY<x>:

XYINteG {<NRf>,<NRf>}

:HISTory[:CURRENT][:SEARCH]:

SElect<x>:PARAMeter:XY<x>:XYINteG?

SElect<x> の <x> = 1 ~ 4

XY<x> の <x> = 1、2

<NRf> = - 4 ~ 4(div)

例

```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:XY1:XYINTEG 1,2
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:XY1:XYINTEG? -> :HISTORY:
CURRENT:SEARCH:SELECT1:PARAMETER:XY1:
XYINTEG 2.000E+00,1.000E+00
```

### **:HISTory[:CURRENT][:SEARCH]:**

#### **SElect<x>:RECTangle?**

機能 Rectangle サーチに関するすべての設定値を問い合わせます。

構文 :HISTory[:CURRENT][:SEARCH]:

SElect<x>:

RECTangle?

<x> = 1 ~ 4

例

```
:HISTORY:CURRENT:SEARCH:SELECT1:
RECTANGLE? -> :HISTORY:CURRENT:
SEARCH:SELECT1:RECTANGLE:
HORIZONTAL 2.000E+00,
1.000E+00;VERTICAL
2.000E+00,1.000E+00
```

### **:HISTory[:CURRENT][:SEARCH]:**

#### **SElect<x>:RECTangle:HORizontal**

機能 Rectangle サーチで用いる四角形の水平位置を設定 / 問い合わせします。

構文 :HISTory[:CURRENT][:SEARCH]:

SElect<x>:RECTangle:

HORizontal {<NRf>,<NRf>}

:HISTory[:CURRENT][:SEARCH]:

SElect<x>:RECTangle:HORizontal?

<x> = 1 ~ 4

<NRf> = - 5 ~ 5(div)

例

```
:HISTORY:CURRENT:SEARCH:SELECT1:
RECTANGLE:HORIZONTAL 1,2
:HISTORY:CURRENT:SEARCH:SELECT1:
RECTANGLE:HORIZONTAL? -> :HISTORY:
CURRENT:SEARCH:SELECT1:RECTANGLE:
HORIZONTAL 2.000E+00,1.000E+00
```

### **:HISTory[:CURRENT][:SEARCH]:**

#### **SElect<x>:RECTangle:VERTical**

機能 Rectangle サーチで用いる四角形の垂直位置を設定 / 問い合わせします。

構文 :HISTory[:CURRENT][:SEARCH]:

SElect<x>:RECTangle:

VERTical {<NRf>,<NRf>}

:HISTory[:CURRENT][:SEARCH]:

SElect<x>:RECTangle:VERTical?

<x> = 1 ~ 4

<NRf> = - 4 ~ 4(div)

例

```
:HISTORY:CURRENT:SEARCH:SELECT1:
RECTANGLE:VERTICAL 1,2
:HISTORY:CURRENT:SEARCH:SELECT1:
RECTANGLE:VERTICAL? -> :HISTORY:
CURRENT:SEARCH:SELECT1:RECTANGLE:
VERTICAL 2.000E+00,1.000E+00
```

### **:HISTory[:CURRENT][:SEARCH]:**

#### **SElect<x>:TRACe**

機能 ヒストリサーチの対象トレースを設定 / 問い合わせします。

構文 :HISTory[:CURRENT][:SEARCH]:

SElect<x>:TRACe {<NRf>}

:HISTory[:CURRENT][:SEARCH]:

SElect<x>:TRACe?

<x> = 1 ~ 4

<NRf> = 1 ~ 8

例

```
:HISTORY:CURRENT:SEARCH:SELECT1:
TRACE 1
:HISTORY:CURRENT:SEARCH:SELECT1:
TRACE? -> :HISTORY:CURRENT:SEARCH:
SELECT1:TRACE 1
```

解説

Window が XY のときは、無効になります。

**:HISTory[:CURRENT][:SEARCH]:****SElect<x>:WAVE?**

機能 波形ゾーンでの検索(WAVE サーチ)に関するすべての設定値を問い合わせます。

構文 :HISTory[:CURRENT][:SEARCH]:  
SElect<x>:WAVE?

<x> = 1 ~ 4

例 :HISTORY:CURRENT:SEARCH:SELECT1:WAVE?  
-> :HISTORY:CURRENT:SEARCH:SELECT1:  
WAVE:TRANGE 2.000E+00,1.000E+00

**:HISTory[:CURRENT][:SEARCH]:****SElect<x>:WAVE:EDIT<x>:EXIT**

機能 WAVE サーチのゾーン編集メニューから抜けます。

構文 :HISTory[:CURRENT][:SEARCH]:  
SElect<x>:

WAVE:EDIT<x>:EXIT {<NRF>|QUIT}

SElect<x> の <x> = 1 ~ 4

EDIT<x> の <x> = 1 ~ 13

<NRF> = 1 ~ 4(各内部メモリ)

例 :HISTORY:CURRENT:SEARCH:SELECT1:WAVE:  
EDIT1:EXIT 1

解説

- ・ゾーン編集中外は、このコマンドはエラーになります。
- ・<NRF> で指定した内部メモリにゾーン波形を保存します。

**:HISTory[:CURRENT][:SEARCH]:****SElect<x>:WAVE:EDIT<x>:PART**

機能 WAVE サーチのゾーン部分編集を設定します。

構文 :HISTory[:CURRENT][:SEARCH]:  
SElect<x>:

WAVE:EDIT<x>:PART {<NRF>,<NRF>,<NRF>,<NRF>}

SElect<x> の <x> = 1 ~ 4

EDIT<x> の <x> = 1 ~ 13(1 ~ 8 は各トレース、9 ~ 12 は各内部メモリ、13 は現在表示されているゾーン波形が編集対象)

<NRF> = - 5 ~ 5(div: 部分編集時間軸カーソル 1、部分編集時間軸カーソル 2)

- 8 ~ 8(div: 上下)(部分編集時間軸カーソル 1、部分編集時間軸カーソル 2、↑、↓の順)

例 :HISTORY:CURRENT:SEARCH:SELECT1:WAVE:  
EDIT1:PART 1,2,3,4

解説 上下移動量は、現在表示されているゾーン波形を基準にした相対値です。

**:HISTory[:CURRENT][:SEARCH]:****SElect<x>:WAVE:EDIT<x>:WHOLE**

機能 WAVE サーチのゾーン全体編集を設定します。

構文 :HISTory[:CURRENT][:SEARCH]:  
SElect<x>:WAVE:EDIT<x>:WHOLE {<NRF>,<NRF>,<NRF>,<NRF>}

SElect<x> の <x> = 1 ~ 4

EDIT<x> の <x> = 1 ~ 13(1 ~ 8 は各トレース、9 ~ 12 は各内部メモリ、13 は現在表示されているゾーン波形が編集対象)

<NRF> = 0 ~ 5(div: 左右)

0 ~ 8(div: 上下)

(←、→、↑、↓の順)

例 :HISTORY:CURRENT:SEARCH:SELECT1:WAVE:  
EDIT1:WHOLE 1,2,3,4

解説 各移動量は、基準波形を基準にした相対値です。

**:HISTory[:CURRENT][:SEARCH]:****SElect<x>:WAVE:TRANGE**

機能 WAVE サーチの判定区間を設定 / 問い合わせます。

構文 :HISTory[:CURRENT][:SEARCH]:  
SElect<x>:WAVE:TRANGE {<NRF>,<NRF>}

:HISTory[:CURRENT][:SEARCH]:

SElect<x>:WAVE:TRANGE?

<x> = 1 ~ 4

<NRF> = - 5 ~ 5(div)

例 :HISTORY:CURRENT:SEARCH:SELECT1:WAVE:  
TRANGE 1,2

:HISTORY:CURRENT:SEARCH:SELECT1:WAVE:

TRANGE? -> :HISTORY:CURRENT:SEARCH:

SELECT1:WAVE:TRANGE

2.000E+00,1.000E+00

**:HISTory[:CURRENT][:SEARCH]:****SElect<x>:WINDOW**

機能 各ヒストリサーチの対象ウィンドウを設定 / 問い合わせます。

構文 :HISTory[:CURRENT][:SEARCH]:  
SElect<x>:WINDOW {MAIN|XY1|XY2|Z1|Z2}

:HISTory[:CURRENT][:SEARCH]:

SElect<x>:WINDOW?

<x> = 1 ~ 4

例 :HISTORY:CURRENT:SEARCH:SELECT1:  
WINDOW MAIN

:HISTORY:CURRENT:SEARCH:SELECT1:

WINDOW? -> :HISTORY:CURRENT:SEARCH:

SELECT1:WINDOW MAIN

解説 XY1、XY2 はレクタングルポリゴンのときに有効です。

## 5.14 HISTory グループ

### **:HISTory[:CURRent]:TIME?**

機能 ヒストリ波形の対象レコード番号の時間を問い合わせします。

構文 :HISTory[:CURRent]:TIME? {<NRf>|MINimum}

例 :HISTORY:CURRENT:TIME? -1

-> :HISTORY:CURRENT:  
TIME "-1 10:20:30.10"

解説 「MINimum」を指定すると、最小のレコード番号になります。

### **:HISTory:REFeRence<x>?**

機能 各リファレンスのヒストリ機能に関するすべての設定値を問い合わせます。

構文 :HISTory:REFeRence<x>?  
<x> = 1 ~ 4

例 :HISTORY:REFERENCE1? -> HISTORY:  
REFERENCE1:DMODE ACOLOR;  
MODE AVERAGE;RECORD 1;REPLAY:SPEED 1

### **:HISTory:REFeRence<x>:DMODE (Display Mode)**

機能 各リファレンスのヒストリ波形の表示モードを設定 / 問い合わせします。

構文 :HISTory:REFeRence<x>:DMODE {ACOLOR|  
AHTone|AINTensity|ONE}  
:HISTory:REFeRence<x>:DMODE?  
<x> = 1 ~ 4

例 :HISTORY:REFERENCE1:DMODE ACOLOR  
:HISTORY:REFERENCE1:DMODE?  
-> :HISTORY:REFERENCE1:DMODE ACOLOR

### **:HISTory:REFeRence<x>:MODE**

機能 各リファレンスのヒストリ波形のハイライト表示モードを設定 / 問い合わせします。

構文 :HISTory:REFeRence<x>:MODE {AVERAge|  
RECOrd}  
:HISTory:REFeRence<x>:MODE?  
<x> = 1 ~ 4

例 :HISTORY:REFERENCE1:MODE AVERAGE  
:HISTORY:REFERENCE1:MODE?  
-> :HISTORY:REFERENCE1:MODE AVERAGE

### **:HISTory:REFeRence<x>:RECOrd**

機能 各リファレンスのヒストリ波形の対象レコードを設定 / 問い合わせします。

構文 :HISTory:REFeRence<x>:RECOrd {<NRf>|  
MINimum}  
:HISTory:REFeRence<x>:RECOrd?  
<x> = 1 ~ 4

例 :HISTORY:REFERENCE1:RECORD 1  
:HISTORY:REFERENCE1:RECORD?  
-> :HISTORY:REFERENCE1:RECORD 1

### **:HISTory:REFeRence<x>:RECOrd?**

機能 各リファレンスのヒストリ波形の最小のレコード番号を問い合わせます。

構文 :HISTory:REFeRence<x>:RECOrd?  
{MINimum}  
<x> = 1 ~ 4

例 :HISTORY:REFERENCE1:RECORD? MINIMUM  
-> :HISTORY:REFERENCE1:RECORD -1

解説 「MINimum」を指定すると、最小のレコード番号になります。

### **:HISTory:REFeRence<x>:REPLAy?**

機能 各リファレンスのヒストリ波形のリプレイ機能に関するすべての設定値を問い合わせます。

構文 :HISTory:REFeRence<x>:REPLAy?  
<x> = 1 ~ 4

例 :HISTORY:REFERENCE1:REPLAY?  
-> :HISTORY:REFERENCE1:REPLAY:SPEED 1

### **:HISTory:REFeRence<x>:REPLAy:JUMP**

機能 各リファレンスのヒストリ波形を指定レコード番号へジャンプさせます。

構文 :HISTory:REFeRence<x>:REPLAy:  
JUMP {MAXimum|MINimum}  
<x> = 1 ~ 4

例 :HISTORY:REFERENCE1:REPLAY:JUMP  
MAXIMUM

### **:HISTory:REFeRence<x>:REPLAy:SPEEd**

機能 各リファレンスのヒストリ波形のリプレイ速度を設定 / 問い合わせします。

構文 :HISTory:REFeRence<x>:REPLAy:SPEEd  
{<NRf>|PER3|PER10|PER30|PER60}  
:HISTory:REFeRence<x>:REPLAy:SPEEd?  
<x> = 1 ~ 4  
<NRf> = 1,3,10

例 :HISTORY:REFERENCE1:REPLAY:SPEED 1  
:HISTORY:REFERENCE1:REPLAY:SPEED?  
-> :HISTORY:REFERENCE1:REPLAY:SPEED 1

### **:HISTory:REFeRence<x>:REPLAy:STARt**

機能 各リファレンスのヒストリ波形のリプレイを指定方向へ開始します。

構文 :HISTory:REFeRence<x>:REPLAy:STARt  
{MAXimum|MINimum}  
<x> = 1 ~ 4

例 :HISTORY:REFERENCE1:REPLAY:  
START MAXIMUM

**:HISTory:REFeRence<x>:REPLay:STOP**

機能 各リファレンスの履歴波形のリプレイを停止します。

構文 :HISTory:REFeRence<x>:REPLay:STOP  
<x> = 1 ~ 4

例 :HISTORY:REFERENCE1:REPLAY:STOP

**:HISTory:REFeRence<x>:TIME?**

機能 リファレンス波形の対象レコード番号の時間を問い合わせします。

構文 :HISTory:REFeRence<x>:TIME? {<NRf>|  
MINimum}  
<x> = 1 ~ 4

例 :HISTORY:REFERENCE1:TIME? -1  
-> :HISTORY:REFERENCE1:TIME "-1  
10:20:30.10"

解説 「MINimum」を指定すると、最小のレコード番号になります。

## 5.15 IMAGE グループ

### **:IMAGE?**

機能 画面イメージデータの出力に関するすべての設定値を問い合わせます。

構文 :IMAGE?

例 :IMAGE? -> :IMAGE:FORMAT BMP;TONE  
COLOR

### **:IMAGE:FORMAt**

機能 画面イメージデータの出力形式を設定 / 問い合わせします。

構文 :IMAGE:FORMAt {BMP|JPEG|PNG}

:IMAGE:FORMAt?

例 :IMAGE:FORMAT BMP

:IMAGE:FORMAT? -> :IMAGE:FORMAT BMP

### **:IMAGE:SEND?**

機能 画面イメージデータを問い合わせます。

構文 :IMAGE:SEND?

例 :IMAGE:SEND? -> #6(6桁のバイト数)(データバイトの並び)(ブロックデータ)

解説 <ブロックデータ>については、4-6ページを参照してください。

### **:IMAGE:TONE**

機能 画面イメージデータの色調を設定 / 問い合わせします。

構文 :IMAGE:TONE {COLor|GRAY|OFF|REVerse}

:IMAGE:TONE?

例 :IMAGE:TONE COLOR

:IMAGE:TONE? -> :IMAGE:TONE COLOR

解説 「:IMAGE:FORMAt JPEG」のときは、「OFF」は選択できません。

## 5.16 INITIALize グループ

### **:INITialize:EXECute**

機能       イニシャライズを実行します。

構文       :INITialize:EXECute

例         :INITIALIZE:EXECUTE

### **:INITialize:UNDO**

機能       実行したイニシャライズを取り消します。

構文       :INITialize:UNDO

例         :INITIALIZE:UNDO

## 5.17 LOGic グループ

### :LOGic?

機能 ロジックに関するすべての設定値を問い合わせます。

構文 :LOGic?

例 :LOGIC? -> :LOGIC:AWINDOW:DISPLAY 1;  
RATIO 50;:LOGIC:DESKEW 0.00000E+00;  
GROUP1:BUNDLE:FORMAT HEXA;MODE 0;:  
LOGIC:GROUP1:DISPLAY 0;  
MAPPING "A7A6A5A4A3A2A1A0";ORDER 1;  
STATE 0;:LOGIC:GROUP2:BUNDLE:  
FORMAT HEXA;MODE 0;:LOGIC:GROUP2:  
DISPLAY 0;MAPPING "B7B6B5B4B3B2B1B0";  
ORDER 2;STATE 0;:LOGIC:GROUP3:BUNDLE:  
FORMAT HEXA;MODE 0;:LOGIC:GROUP3:  
DISPLAY 0;MAPPING "C7C6C5C4C3C2C1C0";  
ORDER 3;STATE 0;:LOGIC:GROUP4:BUNDLE:  
FORMAT HEXA;MODE 0;:LOGIC:GROUP4:  
DISPLAY 0;MAPPING "D7D6D5D4D3D2D1D0";  
ORDER 4;STATE 0;:LOGIC:GROUP5:BUNDLE:  
FORMAT HEXA;MODE 0;:LOGIC:GROUP5:  
DISPLAY 0;MAPPING "";ORDER 5;STATE 0;:  
LOGIC:LABEL:BNAME:A0 "A0";A1 "A1";  
A2 "A2";A3 "A3";A4 "A4";A5 "A5";  
A6 "A6";A7 "A7";B0 "B0";B1 "B1";  
B2 "B2";B3 "B3";B4 "B4";B5 "B5";  
B6 "B6";B7 "B7";C0 "C0";C1 "C1";  
C2 "C2";C3 "C3";C4 "C4";C5 "C5";  
C6 "C6";C7 "C7";D0 "D0";D1 "D1";  
D2 "D2";D3 "D3";D4 "D4";D5 "D5";  
D6 "D6";D7 "D7";:LOGIC:LABEL:LNAME:  
GROUP1 "Group1";GROUP2 "Group2";  
GROUP3 "Group3";GROUP4 "Group4";  
GROUP5 "Group5";:LOGIC:LABEL:MODE 1;:  
LOGIC:MODE 0;POSITION 0;SCLOCK:  
POLARITY RISE;SOURCE A0;:LOGIC:  
SIZE MEDIUM;THRESHOLD:PODA:  
TYPE CMOS5;USERLEVEL 0.000E+00;:  
LOGIC:THRESHOLD:PODB:TYPE CMOS5;  
USERLEVEL 0.000E+00;:LOGIC:THRESHOLD:  
PODC:TYPE CMOS5;USERLEVEL 0.000E+00;:  
LOGIC:THRESHOLD:PODD:TYPE CMOS5;  
USERLEVEL 0.000E+00;:LOGIC:WINDOW:  
ANALOG 1;RATIO A1\_L1

### :LOGic:AWINDOW?

機能 アナログ波形表示に関するすべての設定値を問い合わせます。

構文 :LOGic:AWINDOW?

例 :LOGIC:AWINDOW?  
-> :LOGIC:AWINDOW:DISPLAY 1;RATIO 50

### :LOGic:AWINDOW[:DISPLAY]

機能 アナログ波形表示の ON/OFF を設定 / 問い合わせます。

構文 :LOGic:AWINDOW[:DISPLAY] {<Boolean>}  
:LOGic:AWINDOW[:DISPLAY]?

例 :LOGIC:AWINDOW:DISPLAY ON  
:LOGIC:AWINDOW:DISPLAY? -> :LOGIC:  
AWINDOW:DISPLAY 1

### :LOGic:AWINDOW:RATIO

機能 アナログ波形の表示比率を設定 / 問い合わせます。

構文 :LOGic:AWINDOW:RATIO {<NRf>}  
:LOGic:AWINDOW:RATIO?  
<NRf> = 25(%)、50(%)、75(%)

例 :LOGIC:AWINDOW:RATIO 25  
:LOGIC:AWINDOW:RATIO? -> :LOGIC:  
AWINDOW:RATIO 25

### :LOGic:DESKew

機能 ロジック信号のスキュー補正を設定 / 問い合わせます。

構文 :LOGic:DESKew {<時間>}  
:LOGic:DESKew?  
<時間> = - 80ns ~ 80ns(10ps ステップ)

例 :LOGIC:DESKew 1NS  
:LOGIC:DESKew? -> :LOGIC:  
DESKew 1.000E-09

### :LOGic:GROUP<x>?

機能 各ロジックグループのすべての設定値を問い合わせます。

構文 :LOGic:GROUP<x>?  
<x> = 1 ~ 5

例 :LOGIC:GROUP1? -> :LOGIC:GROUP1:  
BUNDLE:  
FORMAT HEXA;MODE 0;:LOGIC:GROUP1:  
DISPLAY 0;MAPPING "A7A6A5A4A3A2A1A0";  
ORDER 1;STATE 0

### :LOGic:GROUP<x>:BUNDLE?

機能 各ロジックグループのバンドルに関するすべての設定値を問い合わせます。

構文 :LOGic:GROUP<x>:BUNDLE?  
<x> = 1 ~ 5

例 :LOGIC:GROUP1:BUNDLE? -> :LOGIC:  
GROUP1:BUNDLE:FORMAT HEXA;MODE 0

**:LOGic:GRoup<x>:BUNDle:FORMat**

機能 各ロジックグループのバンドル値の表示形式 (パス表示) を設定 / 問い合わせします。

構文 :LOGic:GRoup<x>:BUNDle:  
FORMat {BINary|HEXa|SYMBol}  
:LOGic:GRoup<x>:BUNDle:FORMat?  
<x> = 1 ~ 5

例 :LOGic:GRoup1:BUNDle:FORMat HEXA  
:LOGic:GRoup1:BUNDle:FORMat?  
-> :LOGic:GRoup1:BUNDle:FORMat HEXA

**:LOGic:GRoup<x>:BUNDle:MODE**

機能 各ロジックグループのバンドルのモードを設定 / 問い合わせします。

構文 :LOGic:GRoup<x>:BUNDle:MODE  
{<Boolean>}  
:LOGic:GRoup<x>:BUNDle:MODE?  
<x> = 1 ~ 5

例 :LOGic:GRoup1:BUNDle:MODE ON  
:LOGic:GRoup1:BUNDle:MODE? -> :LOGic:  
GRoup1:BUNDle:MODE 1

**:LOGic:GRoup<x>:BUNDle:SYMBol**

機能 各ロジックグループのバンドル値のシンボルアイテムを設定します。

構文 :LOGic:GRoup<x>:BUNDle:SYMBol {<文字  
列>,<Boolean>}  
<x> = 1 ~ 5

<文字列> = 16 文字以内  
例 :LOGic:GRoup1:BUNDle:SYMBol "TEST",ON

**:LOGic:GRoup<x>:DISPlay**

機能 各ロジックグループの表示の ON/OFF を設定 / 問い合わせします。

構文 :LOGic:GRoup<x>:DISPlay {<Boolean>}  
:LOGic:GRoup<x>:DISPlay?  
<x> = 1 ~ 5

例 :LOGic:GRoup1:DISPlay ON  
:LOGic:GRoup1:DISPlay? -> :LOGic:  
GRoup1:DISPlay 1

**:LOGic:GRoup<x>:MAPPing**

機能 各ロジックグループのビット配置を設定 / 問い合わせします。

構文 :LOGic:GRoup<x>:MAPPing {<文字列>}  
:LOGic:GRoup<x>:MAPPing?  
<x> = 1 ~ 5

<文字列> = "A0 ~ A7"、"B0 ~ B7"、"C0 ~ C7"、"D0 ~ D7" までの組み合わせ 64 文字以内 (DL9510L、DL9505L は <文字列> = "A0 ~ A7"、"C0 ~ C7" までの組み合わせ 32 文字以内)  
例 :LOGic:GRoup1:MAPPing "A7A6A5A4B3B2B1  
B0C7C6C5C4D3D2D1D0"  
:LOGic:GRoup1:MAPPing? -> :LOGic:  
GRoup1:MAPPing "A7A6A5A4B3B2B1B0C7C6C  
5C4D3D2D1D0"

**:LOGic:GRoup<x>:ORDER**

機能 各ロジックグループの表示順を設定 / 問い合わせします。

構文 :LOGic:GRoup<x>:ORDER {<Nrf>}  
:LOGic:GRoup<x>:ORDER?  
<x> = 1 ~ 5  
<Nrf> = 1 ~ 5

例 :LOGic:GRoup1:ORDER 1  
:LOGic:GRoup1:ORDER? -> :LOGic:  
GRoup1:ORDER 1

**:LOGic:GRoup<x>:STATE**

機能 各ロジックグループのステート表示の ON/OFF を設定 / 問い合わせします。

構文 :LOGic:GRoup<x>:STATE {<Boolean>}  
:LOGic:GRoup<x>:STATE?  
<x> = 1 ~ 5

例 :LOGic:GRoup1:STATE ON  
:LOGic:GRoup1:STATE? -> :LOGic:  
GRoup1:STATE 1

**:LOGic:LAbel?**

機能 ロジック信号のラベル名に関するすべての設定値を問い合わせます。

構文 :LOGic:LAbel?

例 :LOGic:LAbel? -> :LOGic:LAbel:BNAME:  
A0 "A0";A1 "A1";A2 "A2";A3 "A3";  
A4 "A4";A5 "A5";A6 "A6";A7 "A7";  
B0 "B0";B1 "B1";B2 "B2";B3 "B3";  
B4 "B4";B5 "B5";B6 "B6";B7 "B7";  
C0 "C0";C1 "C1";C2 "C2";C3 "C3";  
C4 "C4";C5 "C5";C6 "C6";C7 "C7";  
D0 "D0";D1 "D1";D2 "D2";D3 "D3";  
D4 "D4";D5 "D5";D6 "D6";D7 "D7";:  
LOGic:LAbel:LNAME:GRoup1 "Group1";  
GRoup2 "Group2";GRoup3 "Group3";  
GRoup4 "Group4";GRoup5 "Group5";:  
LOGic:LAbel:MODE 1

**:LOGic:LAbel:BNAME?**

機能 ロジック信号のビット名に関するすべての設定値を問い合わせます。

構文 :LOGic:LAbel:BNAME?

例 :LOGic:LAbel:BNAME? -> :LOGic:LAbel:  
BNAME:A0 "A0";A1 "A1";A2 "A2";A3  
"A3";A4 "A4";A5 "A5";A6 "A6";A7 "A7";  
B0 "B0";B1 "B1";B2 "B2";B3 "B3";  
B4 "B4";B5 "B5";B6 "B6";B7 "B7";  
C0 "C0";C1 "C1";C2 "C2";C3 "C3";  
C4 "C4";C5 "C5";C6 "C6";C7 "C7";  
D0 "D0";D1 "D1";D2 "D2";D3 "D3";  
D4 "D4";D5 "D5";D6 "D6";D7 "D7"



## 5.17 LOGic グループ

### **:LOGic:LABel:BNAME:{A<x>|B<x>|C<x>|D<x>}**

機能 ロジック信号の各ビット名を設定 / 問い合わせします。

構文 :LOGic:LABel:BNAME:{A<x>|B<x>|C<x>|D<x>} {<文字列>}  
:LOGic:LABel:BNAME:{A<x>|B<x>|C<x>|D<x>}?  
<x> = 0 ~ 7

<文字列> = 8文字以内

例 :LOGic:LABel:BNAME:A0 "NO\_1"  
:LOGic:LABel:BNAME:A0? -> :LOGic:LABel:BNAME:A0 "NO\_1"

解説 DL9510L、DL9505L は {A<x>|C<x>} が有効です。

### **:LOGic:LABel:LNAME?**

機能 ロジック信号のグループ名に関するすべての設定値を問い合わせます。

構文 :LOGic:LABel:LNAME?

例 :LOGic:LABel:LNAME? -> :LOGic:LABel:LNAME:GROUP1 "Group1";  
GROUP2 "Group2";GROUP3 "Group3";  
GROUP4 "Group4";GROUP5 "Group5"

### **:LOGic:LABel:LNAME:GRouP<x>**

機能 ロジック信号の各グループ名を設定 / 問い合わせします。

構文 :LOGic:LABel:LNAME:GRouP<x> {<文字列>}  
:LOGic:LABel:LNAME:GRouP<x>?

<x> = 1 ~ 5

<文字列> = 8文字以内

例 :LOGic:LABel:LNAME:GROUP1 "NO\_1"  
:LOGic:LABel:LNAME:GROUP1? -> :LOGic:LABel:LNAME:GROUP1 "NO\_1"

### **:LOGic:LABel:MODE**

機能 ロジック信号のラベル名の ON/OFF を設定 / 問い合わせします。

構文 :LOGic:LABel:MODE {<Boolean>}

:LOGic:LABel:MODE?

例 :LOGic:LABel:MODE ON  
:LOGic:LABel:MODE? -> :LOGic:LABel:MODE 1

### **:LOGic:MODE**

機能 ロジック信号の ON/OFF を設定 / 問い合わせします。

構文 :LOGic:MODE {<Boolean>}

:LOGic:MODE?

例 :LOGic:MODE ON  
:LOGic:MODE? -> :LOGic:MODE 1

### **:LOGic:POStion**

機能 ロジック信号の垂直ポジションを設定 / 問い合わせします。

構文 :LOGic:POStion {<NRF>}

:LOGic:POStion?

<NRF> = - 25 ~ 31

例 :LOGic:POStion 0

:LOGic:POStion? -> :LOGic:POStion 0

### **:LOGic:SCLock?**

機能 ロジック信号のステータクロックに関するすべての設定値を問い合わせます。

構文 :LOGic:SCLock?

例 :LOGic:SCLock? -> :LOGic:SCLock:POLARITY RISE;SOURCE A0

### **:LOGic:SCLock:POLarity**

機能 ロジック信号のステータクロックの極性を設定 / 問い合わせします。

構文 :LOGic:SCLock:POLarity

{BOTH|FALL|RISE}

:LOGic:SCLock:POLarity?

例 :LOGic:SCLock:POLARITY BOTH

:LOGic:SCLock:POLARITY? -> :LOGic:SCLock:POLARITY BOTH

### **:LOGic:SCLock:SOURce**

機能 ロジック信号のステータクロックソースを設定 / 問い合わせします。

構文 :LOGic:SCLock:SOURce {A<x>|B<x>|C<x>|D<x>}

:LOGic:SCLock:SOURce?

<x> = 0 ~ 7

例 :LOGic:SCLock:SOURCE A0

:LOGic:SCLock:SOURCE? -> :LOGic:SCLock:SOURCE A0

解説 DL9510L、DL9505L は {A<x>|C<x>} が有効です。

### **:LOGic:SIZE**

機能 ロジック信号の表示サイズを設定 / 問い合わせします。

構文 :LOGic:SIZE {LARGE|MIDium|SMALL|

XLARge|XSMall}

:LOGic:SIZE?

例 :LOGic:SIZE LARGE

:LOGic:SIZE? -> :LOGic:SIZE LARGE

**:LOGic:THReshold?**

機能 ロジック信号のスレシヨルドレベルに関するすべての設定値を問い合わせます。

構文 :LOGic:THReshold?

例 :LOGic:THRESHOLD? -> :LOGIC:  
THRESHOLD:PODA:TYPE CMOS5;  
USERLEVEL 0.000E+00;:LOGIC:THRESHOLD:  
PODB:TYPE CMOS5;USERLEVEL 0.000E+00;:  
LOGIC:THRESHOLD:PODC:TYPE CMOS5;  
USERLEVEL 0.000E+00;:LOGIC:THRESHOLD:  
PODD:TYPE CMOS5;USERLEVEL 0.000E+00

**:LOGic:THReshold:{PODA|PODB|PODC|PODD}?**

機能 指定ポッド (ポート) のスレシヨルドレベルに関するすべての設定値を問い合わせます。

構文 :LOGic:THReshold:{PODA|PODB|PODC|PODD}?

例 :LOGic:THRESHOLD:PODA? -> :LOGIC:  
THRESHOLD:PODA:TYPE CMOS5;  
USERLEVEL 0.000E+00

解説 DL9510L、DL9505L は {PODA|PODC} が有効です。

**:LOGic:THReshold:{PODA|PODB|PODC|PODD}:TYPE**

機能 指定ポッド (ポート) のスレシヨルドレベルのタイプを設定 / 問い合わせします。

構文 :LOGic:THReshold:{PODA|PODB|PODC|PODD}:TYPE {CMOS1|CMOS2|CMOS3|CMOS5|ECL|USER}

例 :LOGic:THRESHOLD:PODA:TYPE CMOS1  
:LOGic:THRESHOLD:PODA:TYPE? -> :  
LOGIC:THRESHOLD:PODA:TYPE CMOS1

解説 DL9510L、DL9505L は {PODA|PODC} が有効です。

**:LOGic:THReshold:{PODA|PODB|PODC|PODD}:USERlevel**

機能 指定ポッド (ポート) のスレシヨルドレベルを設定 / 問い合わせします。

構文 :LOGic:THReshold:{PODA|PODB|PODC|PODD}:USERlevel {<電圧>}  
:LOGic:THReshold:{PODA|PODB|PODC|PODD}:USERlevel?  
<電圧> = - 10 ~ 10V(0.1V ステップ)

例 :LOGic:THRESHOLD:PODA:USERLEVEL 1V  
:LOGic:THRESHOLD:PODA:USERLEVEL?  
-> :LOGic:THRESHOLD:PODA:USERLEVEL  
1.0E+00

解説 DL9510L、DL9505L は {PODA|PODC} が有効です。

## 5.18 MATH グループ

### :MATH<x>?

機能 演算に関するすべての設定値を問い合わせます。

構文 :MATH<x>?  
<x> = 1 ~ 8

例 :MATH1? -> :MATH1:SELECT MATH;DA:  
BFORMAT SBINARY;RESCALING:  
AVALUE 1.0000000E+00;  
BVALUE 0.0000000E+00;;MATH1:  
DISPLAY 1;ECOUNT:  
HYSTERESIS 100.0E-03;POLARITY RISE;;  
MATH1:FILTER:DELAY:  
TIME 0.00000000000E+00;;MATH1:FILTER:  
IIR:FORDER 2;HIPASS:  
COFF 10.000000E+06;;MATH1:FILTER:IIR:  
LOWPASS:COFF 10.000000E+06;;MATH1:  
FILTER:MAVG:WEIGHT 2;;MATH1:FILTER:  
RESCALING:AVALUE 1.0000000E+00;  
BVALUE 0.0000000E+00;;MATH1:FILTER:  
TYPE THROUGH;;MATH1:I2T:UNIT:  
DEFINE "EU";:MATH1:INTEGRAL:PSCALING:  
AVALUE 1.0000000E+00;  
BVALUE 0.0000000E+00;;MATH1:INTEGRAL:  
RESCALING:AVALUE 1.0000000E+00;  
BVALUE 0.0000000E+00;;MATH1:INVERT 0;  
IPOINT:POSITION -5.000E+00;;MATH1:  
LABEL:DEFINE "Math1";MODE 0;;MATH1:  
MINUS:PSCALING1:AVALUE 1.0000000E+00;  
BVALUE 0.0000000E+00;;MATH1:MINUS:  
PSCALING2:AVALUE 1.0000000E+00;  
BVALUE 0.0000000E+00;;MATH1:MINUS:  
RESCALING:AVALUE 1.0000000E+00;  
BVALUE 0.0000000E+00 .....

### :MATH<x>:DA?

機能 D/A 変換に関するすべての設定値を問い合わせます。

構文 :MATH<x>:DA?  
<x> = 1 ~ 4

例 :MATH1:DA? -> :MATH1:DA:  
BFORMAT SBINARY;RESCALING:  
AVALUE 1.0000000E+00;  
BVALUE 0.0000000E+00

### :MATH<x>:DA:ARANging

機能 D/A 変換のオートレンジを実行します。

構文 :MATH<x>:DA:ARANging  
<x> = 1 ~ 4

例 :MATH1:DA:ARANGING

### :MATH<x>:DA:BFOrmat

機能 D/A 変換のバイナリフォーマットを設定 / 問い合わせます。

構文 :MATH<x>:DA:BFOrmat {SBINary|  
TCOMplement}  
:MATH<x>:DA:BFOrmat?  
<x> = 1 ~ 4

例 :MATH1:DA:BFORMAT SBINARY  
:MATH1:DA:BFORMAT? -> :MATH1:DA:  
BFORMAT SBINARY

### :MATH<x>:DA:HISTory:ABORt

機能 D/A 変換のヒストリ演算を中止します。

構文 :MATH<x>:DA:HISTory:ABORt  
<x> = 1 ~ 4

例 :MATH1:DA:HISTORY:ABORT

### :MATH<x>:DA:HISTory:EXECute

機能 D/A 変換のヒストリ演算を実行します。

構文 :MATH<x>:DA:HISTory:EXECute  
<x> = 1 ~ 4

例 :MATH1:DA:HISTORY:EXECUTE

### :MATH<x>:DA:RESCaling?

機能 D/A 変換のリスキューリングに関するすべての設定値を問い合わせます。

構文 :MATH<x>:DA:RESCaling?  
<x> = 1 ~ 4

例 :MATH1:DA:RESCALING? -> :MATH1:DA:  
RESCALING:AVALUE 1.000E+00;  
BVALUE 1.000E+00

### :MATH<x>:DA:RESCaling:AVALue

機能 D/A 変換のリスキューリング係数 A を設定 / 問い合わせします。

構文 :MATH<x>:DA:RESCaling:AVALue {<Nrf>}  
:MATH<x>:DA:RESCaling:AVALue?  
<x> = 1 ~ 4

例 :MATH1:DA:RESCALING:AVALUE 1  
:MATH1:DA:RESCALING:AVALUE?  
-> :MATH1:DA:RESCALING:  
AVALUE 1.000E+00

### :MATH<x>:DA:RESCaling:BVALue

機能 D/A 変換のリスキューリングオフセット値 B を設定 / 問い合わせします。

構文 :MATH<x>:DA:RESCaling:BVALue {<Nrf>}  
:MATH<x>:DA:RESCaling:BVALue?  
<x> = 1 ~ 4  
<Nrf> = -1.0000E+31 ~ 1.0000E+31

例 :MATH1:DA:RESCALING:BVALUE 1  
:MATH1:DA:RESCALING:BVALUE?  
-> :MATH1:DA:RESCALING:  
BVALUE 1.000E+00

### :MATH<x>:DISPlay

機能 演算波形の ON/OFF を設定 / 問い合わせします。

構文 :MATH<x>:DISPlay {<Boolean>}  
:MATH<x>:DISPlay?  
<x> = 1 ~ 8

例 :MATH1:DISPLAY ON  
:MATH1:DISPLAY? -> :MATH1:DISPLAY 1

**:MATH<x>:ECOUNT?**

機能 エッジカウント演算に関するすべての設定値を問い合わせます。

構文 :MATH<x>:ECOUNT?

<x> = 1 ~ 8

例 :MATH1:ECOUNT? -> :MATH1:ECOUNT:  
HYSTERESIS 1.000E+00;POLARITY RISE

**:MATH<x>:ECOUNT:HYSTERESIS**

機能 エッジカウント演算のエッジ検出レベルのヒステリシスを設定 / 問い合わせします。

構文 :MATH<x>:ECOUNT:HYSTERESIS {<NRf>}

:MATH<x>:ECOUNT:HYSTERESIS?

<x> = 1 ~ 8

<NRf> = 0 ~ 4(div)

例 :MATH1:ECOUNT:HYSTERESIS 1  
:MATH1:ECOUNT:HYSTERESIS? -> :MATH1:  
ECOUNT:HYSTERESIS 1.000E+00

**:MATH<x>:ECOUNT:POLARITY**

機能 エッジカウント演算のエッジ検出極性を設定 / 問い合わせします。

構文 :MATH<x>:ECOUNT:POLARITY {FALL|RISE}

:MATH<x>:ECOUNT:POLARITY?

<x> = 1 ~ 8

例 :MATH1:ECOUNT:POLARITY RISE  
:MATH1:ECOUNT:POLARITY? -> :MATH1:  
ECOUNT:POLARITY RISE

**:MATH<x>:FILTER?**

機能 フィルタに関するすべての設定値を問い合わせます。

構文 :MATH<x>:FILTER?

<x> = 1 ~ 8

例 :MATH1:FILTER? -> :MATH1:FILTER:  
DELAY:  
TIME 0.000E+00;:MATH1:FILTER:IIR:  
FORDER 2;HIPASS:COFF 10.00E+06;:  
MATH1:FILTER:IIR:LOWPASS:  
COFF 10.00E+06;:MATH1:FILTER:MAVG:  
WEIGHT 2;:MATH1:FILTER:RESCALING:  
AVALUE 1.000E+00;BVALUE 0.000E+00;:  
MATH1:FILTER:TYPE THROUGH

**:MATH<x>:FILTER:DElay?**

機能 遅延演算に関するすべての設定値を問い合わせます。

構文 :MATH<x>:FILTER:DElay?

<x> = 1 ~ 8

例 :MATH1:FILTER:DElay? -> :MATH1:  
FILTER:DElay:TIME 0.000E+00

**:MATH<x>:FILTER:DElay:TIME**

機能 遅延演算の遅延時間を設定 / 問い合わせします。

構文 :MATH<x>:FILTER:DElay:TIME {<時間>}

:MATH<x>:FILTER:DElay:TIME?

<x> = 1 ~ 8

<時間> = 本体ユーザーズマニュアル参照。

例 :MATH1:FILTER:DElay:TIME 1S  
:MATH1:FILTER:DElay:TIME? -> :MATH1:  
FILTER:DElay:TIME 1.000E+00

**:MATH<x>:FILTER:IIR?**

機能 IIR フィルタ演算に関するすべての設定値を問い合わせます。

構文 :MATH<x>:FILTER:IIR?

<x> = 1 ~ 8

例 :MATH1:FILTER:IIR? -> :MATH1:FILTER:  
IIR:FORDER 2;HIPASS:  
COFF 10.00E+06;:MATH1:FILTER:IIR:  
LOWPASS:COFF 10.00E+00

**:MATH<x>:FILTER:IIR:FORDER****(Filter Order)**

機能 IIR フィルタ演算のフィルタ次数を設定 / 問い合わせします。

構文 :MATH<x>:FILTER:IIR:FORDER {<NRf>}

:MATH<x>:FILTER:IIR:FORDER?

<x> = 1 ~ 8

<NRf> = 1、2

例 :MATH1:FILTER:IIR:FORDER 2  
:MATH1:FILTER:IIR:FORDER? -> :MATH1:  
FILTER:IIR:FORDER 2

**:MATH<x>:FILTER:IIR:HIPASS?**

機能 IIR ハイパスフィルタ演算に関するすべての設定値を問い合わせます。

構文 :MATH<x>:FILTER:IIR:HIPASS?

<x> = 1 ~ 8

例 :MATH1:FILTER:IIR:HIPASS? -> :MATH1:  
FILTER:IIR:HIPASS:COFF 10.00E+06

**:MATH<x>:FILTER:IIR:HIPASS:COFF**

機能 IIR ハイパスフィルタ演算のカットオフ周波数を設定 / 問い合わせします。

構文 :MATH<x>:FILTER:IIR:HIPASS:COFF {<周波数>}

:MATH<x>:FILTER:IIR:HIPASS:COFF?

<x> = 1 ~ 8

<周波数> = 0.01 ~ 1G(Hz)

例 :MATH1:FILTER:IIR:HIPASS:COFF 10MHZ  
:MATH1:FILTER:IIR:HIPASS:COFF?  
-> :MATH1:FILTER:IIR:HIPASS:COFF  
10.00E+06

**:MATH<x>:FILTER:IIR:LOWPass?**

機能 IIR ローパスフィルタ演算に関するすべての設定値を問い合わせます。

構文 :MATH<x>:FILTER:IIR:LOWPass?

<x> = 1 ~ 8

例 :MATH1:FILTER:IIR:LOWPASS? -> :MATH1:  
FILTER:IIR:LOWPASS:COFF 10.00E+06

## 5.18 MATH グループ

### **:MATH<x>:FILTer:IIR:LOWPass:COFF**

機能 IIR ローパスフィルタ演算のカットオフ周波数を設定 / 問い合わせします。

構文 :MATH<x>:FILTer:IIR:LOWPass:COFF {<周波数>}  
:MATH<x>:FILTer:IIR:LOWPass:COFF?  
<x> = 1 ~ 8  
<周波数> = 0.01 ~ 1G(Hz)

例 :MATH1:FILTer:IIR:LOWPass:COFF 10HZ  
:MATH1:FILTer:IIR:LOWPass:COFF?  
-> :MATH1:FILTer:IIR:LOWPass:COFF 10.00E+00

### **:MATH<x>:FILTer:MAVG?**

機能 移動平均演算に関するすべての設定値を問い合わせます。

構文 :MATH<x>:FILTer:MAVG?  
<x> = 1 ~ 8

例 :MATH1:FILTer:MAVG? -> :MATH1:FILTer:MAVG:WEIGHT 2

### **:MATH<x>:FILTer:MAVG:WEIGHT**

機能 移動平均演算の重みを設定 / 問い合わせします。

構文 :MATH<x>:FILTer:MAVG:WEIGHT {<Nrf>}  
:MATH<x>:FILTer:MAVG:WEIGHT?  
<x> = 1 ~ 8  
<Nrf> = 2 ~ 128(2<sup>n</sup> ステップ)

例 :MATH1:FILTer:MAVG:WEIGHT 2  
:MATH1:FILTer:MAVG:WEIGHT? -> :MATH1:FILTer:MAVG:WEIGHT 2

### **:MATH<x>:FILTer:RESCaling?**

機能 フィルタのリスケージングに関するすべての設定値を問い合わせます。

構文 :MATH<x>:FILTer:RESCaling?  
<x> = 1 ~ 8

例 :MATH1:FILTer:RESCaling? -> :MATH1:FILTer:RESCaling:AVAlue 1.000E+00;  
BVALue 0.000E+00

### **:MATH<x>:FILTer:RESCaling:AVAlue**

機能 フィルタのリスケージング係数値 A を設定 / 問い合わせします。

構文 :MATH<x>:FILTer:RESCaling:AVAlue {<Nrf>}  
:MATH<x>:FILTer:RESCaling:AVAlue?  
<x> = 1 ~ 8  
<Nrf> = - 1.0000E+31 ~ 1.0000E+31

例 :MATH1:FILTer:RESCaling:AVAlue 1  
:MATH1:FILTer:RESCaling:AVAlue?  
-> :MATH1:FILTer:RESCaling:AVAlue 1.000E+00

### **:MATH<x>:FILTer:RESCaling:BVALue**

機能 フィルタのリスケージングオフセット値 B を設定 / 問い合わせします。

構文 :MATH<x>:FILTer:RESCaling:BVALue {<Nrf>}  
:MATH<x>:FILTer:RESCaling:BVALue?  
<x> = 1 ~ 8  
<Nrf> = - 1.0000E+31 ~ 1.0000E+31

例 :MATH1:FILTer:RESCaling:BVALue 0  
:MATH1:FILTer:RESCaling:BVALue?  
-> :MATH1:FILTer:RESCaling:BVALue 0.000E+00

### **:MATH<x>:FILTer:TYPE**

機能 フィルタのタイプを設定 / 問い合わせします。

構文 :MATH<x>:FILTer:TYPE {DElay|IHPass|ILPass|MAVG|ThRough}  
:MATH<x>:FILTer:TYPE?  
<x> = 1 ~ 8

例 :MATH1:FILTer:TYPE DELAY  
:MATH1:FILTer:TYPE? -> :MATH1:FILTer:TYPE DELAY

### **:MATH<x>:INTEgral?**

機能 積分演算に関するすべての設定値を問い合わせます。

構文 :MATH<x>:INTEgral?  
<x> = 1 ~ 8

例 :MATH1:INTEgral? -> :MATH1:INTEgral:PSCaling:AVAlue 1.000E+00;  
BVALue 0.000E+00;:MATH1:INTEgral:RESCaling:AVAlue 1.000E+00;  
BVALue 0.000E+00

### **:MATH<x>:INTEgral:PSCaling?**

機能 積分演算のプリスケージングに関するすべての設定値を問い合わせます。

構文 :MATH<x>:INTEgral:PSCaling?  
<x> = 1 ~ 8

例 :MATH1:INTEgral:PSCaling? -> :MATH1:INTEgral:PSCaling:AVAlue 1.000E+00;  
BVALue 0.000E+00

### **:MATH<x>:INTEgral:PSCaling:AVAlue**

機能 積分演算のプリスケージング係数値 A を設定 / 問い合わせします。

構文 :MATH<x>:INTEgral:PSCaling:AVAlue {<Nrf>}  
:MATH<x>:INTEgral:PSCaling:AVAlue?  
<x> = 1 ~ 8  
<Nrf> = - 1.0000E+31 ~ 1.0000E+31

例 :MATH1:INTEgral:PSCaling:AVAlue 1  
:MATH1:INTEgral:PSCaling:AVAlue?  
-> :MATH1:INTEgral:PSCaling:AVAlue 1.000E+00

**:MATH<x>:INTEgral:PSCaling:BVALue**

機能 積分演算のプリスケールリングオフセット値 B を設定 / 問い合わせします。

構文 :MATH<x>:INTEgral:PSCaling:  
BVALue {<Nrf>}  
:MATH<x>:INTEgral:PSCaling:BVALue?  
<x> = 1 ~ 8

例 :MATH1:INTEGRAL:PSCALING:BVALUE 0  
:MATH1:INTEGRAL:PSCALING:BVALUE?  
-> :MATH1:INTEGRAL:PSCALING:  
BVALUE 0.000E+00

**:MATH<x>:INTEgral:RESCaling?**

機能 積分演算のリスケールリングに関するすべての設定値を問い合わせします。

構文 :MATH<x>:INTEgral:RESCaling?  
<x> = 1 ~ 8

例 :MATH1:INTEGRAL:RESCALING? -> :MATH1:  
INTEGRAL:RESCALING:  
AVALUE 1.000E+00;BVALUE 0.000E+00

**:MATH<x>:INTEgral:RESCaling:AVALue**

機能 積分演算のリスケールリング係数値 A を設定 / 問い合わせします。

構文 :MATH<x>:INTEgral:RESCaling:  
AVALue {<Nrf>}  
:MATH<x>:INTEgral:RESCaling:AVALue?  
<x> = 1 ~ 8

例 :MATH1:INTEGRAL:RESCALING:AVALUE 1  
:MATH1:INTEGRAL:RESCALING:AVALUE?  
-> :MATH1:INTEGRAL:RESCALING:  
AVALUE 1.000E+00

**:MATH<x>:INTEgral:RESCaling:BVALue**

機能 積分演算のリスケールリングオフセット値 B を設定 / 問い合わせします。

構文 :MATH<x>:INTEgral:RESCaling:  
BVALue {<Nrf>}  
:MATH<x>:INTEgral:RESCaling:BVALue?  
<x> = 1 ~ 8

例 :MATH1:INTEGRAL:RESCALING:BVALUE 0  
:MATH1:INTEGRAL:RESCALING:BVALUE?  
-> :MATH1:INTEGRAL:RESCALING:  
BVALUE 0.000E+00

**:MATH<x>:INVert**

機能 演算波形の反転表示の ON/OFF を設定 / 問い合わせします。

構文 :MATH<x>:INVert {<Boolean>}  
:MATH<x>:INVert?  
<x> = 1 ~ 8

例 :MATH1:INVERT ON  
:MATH1:INVERT? -> :MATH1:INVERT 1

**:MATH<x>:IPoint? (Initial Point)**

機能 演算基準点に関するすべての設定値を問い合わせします。

構文 :MATH<x>:IPoint?  
<x> = 1 ~ 8

例 :MATH1:IPOINT? -> :MATH1:IPOINT:  
POSITION 0.000E+00

**:MATH<x>:IPoint:JUMP**

機能 演算基準点を指定の場所にジャンプさせます。

構文 :MATH<x>:IPoint:JUMP {<Nrf>|TRIGger|  
Z1|Z2}  
<x> = 1 ~ 8  
<Nrf> = -5, 0(div)

例 :MATH1:IPOINT:JUMP TRIGGER

**:MATH<x>:IPoint:POSITION**

機能 演算基準点を設定 / 問い合わせします。

構文 :MATH<x>:IPoint:POSITION {<Nrf>}  
:MATH<x>:IPoint:POSITION?  
<x> = 1 ~ 8  
<Nrf> = -5 ~ 5(div)

例 :MATH1:IPOINT:POSITION 0  
:MATH1:IPOINT:POSITION? -> :MATH1:  
IPOINT:POSITION 0.000E+00

解説 「:MATH<x>:OPERation INTEgral」のとき有効です。

**:MATH<x>:LAbel?**

機能 演算波形のラベル名に関するすべての設定値を問い合わせします。

構文 :MATH<x>:LAbel?  
<x> = 1 ~ 8

例 :MATH1:LABEL? -> :MATH1:LABEL:  
DEFINE "MATH1";MODE 0

**:MATH<x>:LAbel[:DEFine]**

機能 演算波形のラベル名を設定 / 問い合わせします。

構文 :MATH<x>:LAbel[:DEFine] {<文字列>}  
:MATH<x>:LAbel[:DEFine]?  
<x> = 1 ~ 8  
<文字列> = 8 文字以内

例 :MATH1:LABEL:DEFINE "MATH1"  
:MATH1:LABEL:DEFINE? -> :MATH1:LABEL:  
DEFINE "MATH1"

**:MATH<x>:LAbel:MODE**

機能 演算波形のラベル名表示の ON/OFF を設定 / 問い合わせします。

構文 :MATH<x>:LAbel:MODE {<Boolean>}  
:MATH<x>:LAbel:MODE?  
<x> = 1 ~ 8

例 :MATH1:LABEL:MODE ON  
:MATH1:LABEL:MODE? -> :MATH1:LABEL:  
MODE 1

## 5.18 MATH グループ

### **:MATH<x>:MINus?**

機能 減算に関するすべての設定値を問い合わせます。

構文 :MATH<x>:MINus?

<x> = 1 ~ 8

例 :MATH1:MINUS? -> :MATH1:MINUS:

PSCALING1:AVAlUE 1.000E+00;

BVALUE 0.000E+00;:MATH1:MINUS:

PSCALING2:AVAlUE 1.000E+00;

BVALUE 0.000E+00;:MATH1:MINUS:

RESCALING:AVAlUE 1.000E+00;

BVALUE 0.000E+00+00

### **:MATH<x>:MINus:PSCaling<x>?**

機能 減算のプリスケールリングに関するすべての設定値を問い合わせます。

構文 :MATH<x>:MINus:PSCaling<x>?

MATH<x> の <x> = 1 ~ 8

PSCaling<x> の <x> = 1、2

例 :MATH1:MINUS:PSCALING1? -> :MATH1:

MINUS:PSCALING1:AVAlUE 1.000E+00;

BVALUE 0.000E+00

### **:MATH<x>:MINus:PSCaling<x>:AVAlue**

機能 減算のプリスケールリング係数値 A を設定 / 問い合わせします。

構文 :MATH<x>:MINus:PSCaling<x>:

AVAlue {<Nrf>}

:MATH<x>:MINus:PSCaling<x>:AVAlue?

MATH<x> の <x> = 1 ~ 8

PSCaling<x> の <x> = 1、2

<Nrf> = -1.0000E+31 ~ 1.0000E+31

例 :MATH1:MINUS:PSCALING1:AVAlUE 1

:MATH1:MINUS:PSCALING1:AVAlUE?

-> :MATH1:MINUS:PSCALING1:

AVAlUE 1.000E+00

### **:MATH<x>:MINus:PSCaling<x>:BVALue**

機能 減算のプリスケールリングオフセット値 B を設定 / 問い合わせします。

構文 :MATH<x>:MINus:PSCaling<x>:

BVALue {<Nrf>}

:MATH<x>:MINus:PSCaling<x>:BVALue?

MATH<x> の <x> = 1 ~ 8

PSCaling<x> の <x> = 1、2

<Nrf> = -1.0000E+31 ~ 1.0000E+31

例 :MATH1:MINUS:PSCALING1:BVALUE 0

:MATH1:MINUS:PSCALING1:BVALUE?

-> :MATH1:MINUS:PSCALING1:

BVALUE 0.000E+00

### **:MATH<x>:MINus:RESCaling?**

機能 減算のリスケーリングに関するすべての設定値を問い合わせます。

構文 :MATH<x>:MINus:RESCaling?

<x> = 1 ~ 8

例 :MATH1:MINUS:RESCALING? -> :MATH1:

MINUS:RESCALING:

AVAlUE 1.000E+00;BVALUE 0.000E+00

### **:MATH<x>:MINus:RESCaling:AVAlue**

機能 減算のリスケーリング係数値 A を設定 / 問い合わせします。

構文 :MATH<x>:MINus:RESCaling:

AVAlue {<Nrf>}

:MATH<x>:MINus:RESCaling:AVAlue?

<x> = 1 ~ 8

<Nrf> = -1.0000E+31 ~ 1.0000E+31

例 :MATH1:MINUS:RESCALING:AVAlUE 1

:MATH1:MINUS:RESCALING:AVAlue?

-> :MATH1:MINUS:RESCALING:

AVAlUE 1.000E+00

### **:MATH<x>:MINus:RESCaling:BVALue**

機能 減算のリスケーリングオフセット値 B を設定 / 問い合わせします。

構文 :MATH<x>:MINus:RESCaling:

BVALue {<Nrf>}

:MATH<x>:MINus:RESCaling:BVALue?

<x> = 1 ~ 8

<Nrf> = -1.0000E+31 ~ 1.0000E+31

例 :MATH1:MINUS:RESCALING:BVALUE 0

:MATH1:MINUS:RESCALING:BVALue?

-> :MATH1:MINUS:RESCALING:

BVALUE 0.000E+00

### **:MATH<x>:MULTiple?**

機能 乗算に関するすべての設定値を問い合わせます。

構文 :MATH<x>:MULTiple?

<x> = 1 ~ 8

例 :MATH1:MULTIPLE? -> :MATH1:MULTIPLE:

PSCALING1:AVAlUE 1.000E+00;

BVALUE 0.000E+00;:MATH1:MULTIPLE:

PSCALING2:AVAlUE 1.000E+00;

BVALUE 0.000E+00;:MATH1:MULTIPLE:

RESCALING:AVAlUE 1.000E+00;

BVALUE 0.000E+00

### **:MATH<x>:MULTiple:PSCaling<x>?**

機能 乗算のプリスケールリングに関するすべての設定値を問い合わせます。

構文 :MATH<x>:MULTiple:PSCaling<x>?

MATH<x> の <x> = 1 ~ 8

PSCaling<x> の <x> = 1、2

例 :MATH1:MULTIPLE:PSCALING1? -> :MATH1:

MULTIPLE:PSCALING1:

AVAlUE 1.000E+00;BVALUE 0.000E+00

**:MATH<x>:MULTiple:PSCaling<x>:AVALue**

機能 乗算のプリスケール係数値 A を設定 / 問い合わせします。

構文 :MATH<x>:MULTiple:PSCaling<x>:  
AVALue {<NRf>}  
:MATH<x>:MULTiple:PSCaling<x>:AVALue?  
MATH<x> の <x> = 1 ~ 8  
PSCaling<x> の <x> = 1, 2  
<NRf> = -1.0000E+31 ~ 1.0000E+31

例 :MATH1:MULTIPLE:PSCALING1:AVALUE 1  
:MATH1:MULTIPLE:PSCALING1:AVALUE?  
-> :MATH1:MULTIPLE:PSCALING1:  
AVALUE 1.000E+00

**:MATH<x>:MULTiple:PSCaling<x>:BVALue**

機能 乗算のプリスケールオフセット値 B を設定 / 問い合わせします。

構文 :MATH<x>:MULTiple:PSCaling<x>:  
BVALue {<NRf>}  
:MATH<x>:MULTiple:PSCaling<x>:BVALue?  
MATH<x> の <x> = 1 ~ 8  
PSCaling<x> の <x> = 1, 2  
<NRf> = -1.0000E+31 ~ 1.0000E+31

例 :MATH1:MULTIPLE:PSCALING1:BVALUE 0  
:MATH1:MULTIPLE:PSCALING1:BVALUE?  
-> :MATH1:MULTIPLE:PSCALING1:  
BVALUE 0.000E+00

**:MATH<x>:MULTiple:RESCaling?**

機能 乗算のリスケールに関するすべての設定値を問い合わせます。

構文 :MATH<x>:MULTiple:RESCaling?  
<x> = 1 ~ 8

例 :MATH1:MULTIPLE:RESCALING? -> :MATH1:  
MULTIPLE:RESCALING:  
AVALUE 1.000E+00;BVALUE 0.000E+00

**:MATH<x>:MULTiple:RESCaling:AVALue**

機能 乗算のリスケール係数値 A を設定 / 問い合わせします。

構文 :MATH<x>:MULTiple:RESCaling:  
AVALue {<NRf>}  
:MATH<x>:MULTiple:RESCaling:AVALue?  
<x> = 1 ~ 8  
<NRf> = -1.0000E+31 ~ 1.0000E+31

例 :MATH1:MULTIPLE:RESCALING:AVALUE 1  
:MATH1:MULTIPLE:RESCALING:AVALUE?  
-> :MATH1:MULTIPLE:RESCALING:  
AVALUE 1.000E+00

**:MATH<x>:MULTiple:RESCaling:BVALue**

機能 乗算のリスケールオフセット値 B を設定 / 問い合わせします。

構文 :MATH<x>:MULTiple:RESCaling:  
BVALue {<NRf>}  
:MATH<x>:MULTiple:RESCaling:BVALue?  
<x> = 1 ~ 8  
<NRf> = -1.0000E+31 ~ 1.0000E+31

例 :MATH1:MULTIPLE:RESCALING:BVALUE 0  
:MATH1:MULTIPLE:RESCALING:BVALUE?  
-> :MATH1:MULTIPLE:RESCALING:  
BVALUE 0.000E+00

**:MATH<x>:OPERation**

機能 演算子を設定 / 問い合わせします。

構文 :MATH<x>:OPERation  
{(DA|ECount|FILTer|INTEgral|MINus|MULTiple|PLUS|RCOUNT|USERdefine), (<NRf>|GROup<x>), <NRf>}  
:MATH<x>:OPERation?  
MATH<x> = 1 ~ 8((DA|USERdefine) 選択の場合は、<x> = 1 ~ 4 だけ)  
GROup<x> = 1 ~ 5({GROup<x>}) は、演算子が DA のときだけ有効)  
<NRf> = 1 ~ 4

例 :MATH1:OPERATION FILTER,1  
:MATH1:OPERATION?

-> :MATH1:OPERATION FILTER,1

解説 単項演算子 (ECount|FILTer|INTEgral) の場合は、最初の <NRf> に対象波形を選択します。2 項演算子 (MINus|MULTiple|PLUS|RCOUNT) の場合は、最初の <NRf> に 1 項目の対象波形を選択し、次の <NRf> に 2 項目の対象波形を選択します。DA 演算子の場合は、GROup<x> を選択します。USERdefine 演算子の場合、<NRf> は必要ありません。

**:MATH<x>:PLUS?**

機能 加算に関するすべての設定値を問い合わせます。

構文 :MATH<x>:PLUS?  
<x> = 1 ~ 8

例 :MATH1:PLUS? -> :MATH1:PLUS:  
PSCALING1:AVALUE 1.000E+00;  
BVALUE 0.000E+00;:MATH1:PLUS:  
PSCALING2:AVALUE 1.000E+00;  
BVALUE 0.000E+00;:MATH1:PLUS:  
RESCALING:AVALUE 1.000E+00;  
BVALUE 0.000E+00

**:MATH<x>:PLUS:PSCaling<x>?**

機能 加算のプリスケールに関するすべての設定値を問い合わせます。

構文 :MATH<x>:PLUS:PSCaling<x>?  
MATH<x> の <x> = 1 ~ 8  
PSCaling<x> の <x> = 1, 2

例 :MATH1:PLUS:PSCALING1? -> :MATH1:  
PLUS:PSCALING1:AVALUE 1.000E+00;  
BVALUE 0.000E+00



## 5.18 MATH グループ

### **:MATH<x>:PLUS:PSCaling<x>:AVALue**

機能 加算のプリスケール係数値 A を設定 / 問い合わせします。

構文 :MATH<x>:PLUS:PSCaling<x>:  
AVALue {<Nrf>}  
:MATH<x>:PLUS:PSCaling<x>:AVALue?  
MATH<x> の <x> = 1 ~ 8  
PSCaling<x> の <x> = 1、2  
<Nrf> = -1.0000E+31 ~ 1.0000E+31  
例 :MATH1:PLUS:PSCALING1:AVALUE 1  
:MATH1:PLUS:PSCALING1:AVALUE?  
-> :MATH1:PLUS:PSCALING1:  
AVALUE 1.0000E+00

### **:MATH<x>:PLUS:PSCaling<x>:BVALue**

機能 加算のプリスケールオフセット値 B を設定 / 問い合わせします。

構文 :MATH<x>:PLUS:PSCaling<x>:  
BVALue {<Nrf>}  
:MATH<x>:PLUS:PSCaling<x>:BVALue?  
MATH<x> の <x> = 1 ~ 8  
PSCaling<x> の <x> = 1、2  
<Nrf> = -1.0000E+31 ~ 1.0000E+31  
例 :MATH1:PLUS:PSCALING1:BVALUE 0  
:MATH1:PLUS:PSCALING1:BVALUE?  
-> :MATH1:PLUS:PSCALING1:  
BVALUE 0.0000E+00

### **:MATH<x>:PLUS:RESCaling?**

機能 加算のリスケールに関するすべての設定値を問い合わせます。

構文 :MATH<x>:PLUS:RESCaling?  
<x> = 1 ~ 8  
例 :MATH1:PLUS:RESCALING? -> :MATH1:  
PLUS:RESCALING:AVALUE 1.0000E+00;  
BVALUE 0.0000E+00

### **:MATH<x>:PLUS:RESCaling:AVALue**

機能 加算のリスケール係数値 A を設定 / 問い合わせします。

構文 :MATH<x>:PLUS:RESCaling:  
AVALue {<Nrf>}  
:MATH<x>:PLUS:RESCaling:AVALue?  
<x> = 1 ~ 8  
<Nrf> = -1.0000E+31 ~ 1.0000E+31  
例 :MATH1:PLUS:RESCALING:AVALUE 1  
:MATH1:PLUS:RESCALING:AVALUE?  
-> :MATH1:PLUS:RESCALING:  
AVALUE 1.0000E+00

### **:MATH<x>:PLUS:RESCaling:BVALue**

機能 加算のリスケールオフセット値 B を設定 / 問い合わせします。

構文 :MATH<x>:PLUS:RESCaling:  
BVALue {<Nrf>}  
:MATH<x>:PLUS:RESCaling:BVALue?  
<x> = 1 ~ 8  
<Nrf> = -1.0000E+31 ~ 1.0000E+31  
例 :MATH1:PLUS:RESCALING:BVALUE 0  
:MATH1:PLUS:RESCALING:BVALUE?  
-> :MATH1:PLUS:RESCALING:  
BVALUE 0.0000E+00

### **:MATH<x>:POSition**

機能 演算波形の垂直ポジションを設定 / 問い合わせします。

構文 :MATH<x>:POSition {<Nrf>}  
:MATH<x>:POSition?  
<x> = 1 ~ 8  
<Nrf> = -4 ~ 4(div)

例 :MATH1:POSITION 0  
:MATH1:POSITION? -> :MATH1:  
POSITION 0.0000E+00

### **:MATH<x>:SCALE?**

機能 スケールに関するすべての設定値を問い合わせます。

構文 :MATH<x>:SCALE?  
<x> = 1 ~ 8

例 :MATH1:SCALE? -> :MATH1:SCALE:  
CENTER 1.0000E+00;MODE AUTO;  
SENSITIVITY 1.0000E+00

### **:MATH<x>:SCALE:CENTer**

機能 演算波形のオフセットを設定 / 問い合わせします。

構文 :MATH<x>:SCALE:CENTer {<Nrf>|<電圧>|  
<電流>}  
:MATH<x>:SCALE:CENTer?  
<x> = 1 ~ 8

<Nrf>、<電圧>、<電流> = 本体ユーザーズマニュアル参照。

例 :MATH1:SCALE:CENTER 1  
:MATH1:SCALE:CENTER? -> :MATH1:SCALE:  
CENTER 1.0000E+00

### **:MATH<x>:SCALE:MODE**

機能 スケール方法を設定 / 問い合わせします。

構文 :MATH<x>:SCALE:MODE {AUTO|MANual}  
:MATH<x>:SCALE:MODE?  
<x> = 1 ~ 8

例 :MATH1:SCALE:MODE AUTO  
:MATH1:SCALE:MODE? -> :MATH1:SCALE:  
MODE AUTO

### **:MATH<x>:SCALE:SENSitivity**

機能 演算波形の垂直軸感度を設定 / 問い合わせします。

構文 :MATH<x>:SCALE:SENSitivity {<Nrf>|<電  
圧>|<電流>}  
:MATH<x>:SCALE:SENSitivity?  
<x> = 1 ~ 8

<Nrf>、<電圧>、<電流> = 本体ユーザーズマニュアル参照。

例 :MATH1:SCALE:SENSITIVITY 1  
:MATH1:SCALE:SENSITIVITY? -> :MATH1:  
SCALE:SENSITIVITY 1.0000E+00

**:MATH<x>:SElect**

機能 表示オプションを設定 / 問い合わせします。

構文 :MATH<x>:SElect  
{ INPUT|MATH|REFERENCE}  
:MATH<x>:SElect?  
<x> = 1 ~ 8

例 :MATH1:SELECT INPUT  
:MATH1:SELECT? -> :MATH1:SELECT INPUT

**:MATH<x>:SVALue (Scale VALUE)**

機能 スケール値表示の ON/OFF を設定 / 問い合わせします。

構文 :MATH<x>:SVALue {<Boolean>}  
:MATH<x>:SVALue?  
<x> = 1 ~ 8

例 :MATH1:SVALUE ON  
:MATH1:SVALUE? -> MATH1:SVALUE 1

**:MATH<x>:THReshold<x>**

機能 カウント演算のエッジ検出レベルを設定 / 問い合わせします。

構文 :MATH<x>:THReshold<x> {<Nrf>|<電圧>|  
<電流>}  
:MATH<x>:THReshold<x>?  
MATH<x>の<x> = 1 ~ 8  
THReshold<x>の<x> = 1、2  
<Nrf>、<電圧>、<電流> = 本体ユーザーズマニュアル参照。

例 :MATH1:THRESHOLD1 1  
:MATH1:THRESHOLD1? -> :MATH1:  
THRESHOLD1 1.000E+00

解説 THReshold2 は、演算子が {RCOut} のときに有効です。

**:MATH<x>:UNIT?**

機能 演算単位に関するすべての設定値を問い合わせします。

構文 :MATH<x>:UNIT?  
<x> = 1 ~ 8

例 :MATH1:UNIT? -> :MATH1:UNIT:  
DEFINE "EU";MODE AUTO

**:MATH<x>:UNIT[:DEFine]**

機能 演算単位を設定 / 問い合わせします。

構文 :MATH<x>:UNIT[:DEFine] {<文字列>}  
:MATH<x>:UNIT[:DEFine]?  
<x> = 1 ~ 8  
<文字列> = 4 文字以内

例 :MATH1:UNIT:DEFINE "EU"  
:MATH1:UNIT:DEFINE? -> :MATH1:UNIT:  
DEFINE "EU"

**:MATH<x>:UNIT:MODE**

機能 演算単位の自動 / 手動付加を設定 / 問い合わせします。

構文 :MATH<x>:UNIT:MODE {AUTO|USERdefine}  
:MATH<x>:UNIT:MODE?  
<x> = 1 ~ 8

例 :MATH1:UNIT:MODE AUTO  
:MATH1:UNIT:MODE? -> :MATH1:UNIT:  
MODE AUTO

**:MATH<x>:USERdefine?**

機能 ユーザー定義演算に関するすべての設定値を問い合わせします。

構文 :MATH<x>:USERdefine?  
<x> = 1 ~ 4

例 :MATH1:USERDEFINE? -> :MATH1:  
USERDEFINE:CONSTANT1 1.000E+00;  
CONSTANT2 1.000E+00;  
CONSTANT3 1.000E+00;  
CONSTANT4 1.000E+00;DEFINE "C1-C2"

**:MATH<x>:USERdefine:ARANging**

機能 ユーザー定義演算のオートレンジを実行します。

構文 :MATH<x>:USERdefine:ARANging  
<x> = 1 ~ 4

例 :MATH1:USERDEFINE:ARANGING

**:MATH<x>:USERdefine:CONSTant<x>**

機能 ユーザー定義演算の定数を設定 / 問い合わせします。

構文 :MATH<x>:USERdefine:  
CONSTant<x> {<Nrf>}  
:MATH<x>:USERdefine:CONSTant<x>?  
MATH<x>の<x> = 1 ~ 4  
CONSTant<x>の<x> = 1 ~ 4

<Nrf> = -1.0000E+31 ~ 1.0000E+31  
例 :MATH1:USERDEFINE:CONSTANT1 1  
:MATH1:USERDEFINE:CONSTANT1?  
-> :MATH1:USERDEFINE:  
CONSTANT1 1.000E+00

**:MATH<x>:USERdefine:DEFine**

機能 ユーザー定義演算の演算式を設定 / 問い合わせします。

構文 :MATH<x>:USERdefine:DEFine {<文字列>}  
:MATH<x>:USERdefine:DEFine?  
<x> = 1 ~ 4

<文字列> = 128 文字以内

例 :MATH1:USERDEFINE:DEFINE "C1-C2"  
:MATH1:USERDEFINE:DEFINE? -> :MATH1:  
USERDEFINE:DEFINE "C1-C2"

解説 本体画面に表示されるキーボード以外の文字や記号は使用できません。  
演算式の詳細は本体ユーザーズマニュアル参照。

## 5.18 MATH グループ

---

### **:MATH<x>:USERdefine:HISTory:ABORt**

機能 ユーザー定義演算のヒストリ演算を中止します。

構文 :MATH<x>:USERdefine:HISTory:ABORt

<x> = 1 ~ 4

例 :MATH1:USERDEFINE:HISTORY:ABORT

### **:MATH<x>:USERdefine:HISTory:EXECute**

機能 ユーザー定義演算のヒストリ演算を実行します。

構文 :MATH<x>:USERdefine:HISTory:EXECute

<x> = 1 ~ 4

例 :MATH1:USERDEFINE:HISTORY:EXECUTE

## 5.19 MEASure グループ

### :MEASure?

機能 波形パラメータの自動測定に関するすべての設定値を問い合わせます。

構文 :MEASure?

例 :MEASURE? -> :MEASURE:BIT1:AREA1:COUNT:STATE 0;:MEASURE:BIT1:AREA1:DELAY:MEASURE:COUNT 1;POLARITY RISE;:MEASURE:BIT1:AREA1:DELAY:REFERENCE:COUNT 1;POLARITY RISE;TRACE A0;:MEASURE:BIT1:AREA1:DELAY:SOURCE TRACE;STATE 0;:MEASURE:BIT1:AREA1:DT:STATE 0;:MEASURE:BIT1:AREA1:DUTYCYCLE:STATE 0;:MEASURE:BIT1:AREA1:FREQUENCY:STATE 0;:MEASURE:BIT1:AREA1:NWIDTH:STATE 0;:MEASURE:BIT1:AREA1:PERFREQUENCY:STATE 0;:MEASURE:BIT1:AREA1:PERIOD:STATE 0;:MEASURE:BIT1:AREA1:PWIDITH:STATE 0;:MEASURE:BIT1:AREA2:COUNT:STATE 0;:MEASURE:BIT1:AREA2:DELAY:MEASURE:COUNT 1;POLARITY RISE;:MEASURE:BIT1:AREA2:DELAY:REFERENCE:COUNT 1;POLARITY RISE;TRACE A0;:MEASURE:BIT1:AREA2:DELAY:SOURCE TRACE;STATE 0;:MEASURE:BIT1:AREA2:DUTYCYCLE:STATE 0;:MEASURE:BIT1:AREA2:FREQUENCY:STATE 0;:MEASURE:BIT1:AREA2:NWIDTH:STATE 0;:MEASURE:BIT1:AREA2:PERFREQUENCY:STATE 0;:MEASURE:BIT1:AREA2:PERIOD:STATE 0;:MEASURE:BIT1:AREA2:PWIDITH:STATE 0;:MEASURE:BIT2:AREA1:COUNT:STATE 0;:MEASURE:BIT2:AREA1:DELAY:MEASURE:COUNT 1;POLARITY RISE;:MEASURE:BIT2:AREA1:DELAY:REFERENCE:COUNT 1;POLARITY RISE;TRACE A0;:MEASURE:BIT2:AREA1:DELAY:SOURCE TRACE;STATE 0;:MEASURE:BIT2:AREA1:DUTYCYCLE:STATE 0;:MEASURE:BIT2:AREA1:FREQUENCY:STATE 0;:MEASURE:BIT2:AREA1:NWIDTH:STATE 0;:MEASURE:BIT2:AREA1:PERFREQUENCY:STATE 0;:MEASURE:BIT2:AREA1:PERIOD:STATE 0;:MEASURE:BIT2:AREA1:PWIDITH:STATE 0 . . . . .

### :MEASure:BIT<x>?

機能 各ロジックビットに関するすべての設定値を問い合わせます。

構文 :MEASure:BIT<x>?

例 <x> = 1 ~ 32(DL9510L、DL9505LはBIT<x>の<x> = 1 ~ 8、17 ~ 24が有効です。)  
:MEASURE:BIT1? -> :MEASURE:BIT1:AREA1:COUNT:STATE 0;:MEASURE:BIT1:AREA1:DELAY:MEASURE:COUNT 1;POLARITY RISE;:MEASURE:BIT1:AREA1:DELAY:REFERENCE:COUNT 1;POLARITY RISE;TRACE A0;:MEASURE:BIT1:AREA1:DT:STATE 0;:MEASURE:BIT1:AREA1:DUTYCYCLE:STATE 0;:MEASURE:BIT1:AREA1:FREQUENCY:STATE 0;:MEASURE:BIT1:AREA1:NWIDTH:STATE 0;:MEASURE:BIT1:AREA1:PERFREQUENCY:STATE 0;:MEASURE:BIT1:AREA1:PERIOD:STATE 0;:MEASURE:BIT1:AREA1:PWIDITH:STATE 0;:MEASURE:BIT1:AREA2:COUNT:STATE 0;:MEASURE:BIT1:AREA2:DELAY:MEASURE:COUNT 1;POLARITY RISE;:MEASURE:BIT1:AREA2:DELAY:REFERENCE:COUNT 1;POLARITY RISE;TRACE A0;:MEASURE:BIT1:AREA2:DELAY:SOURCE TRACE;STATE 0;:MEASURE:BIT1:AREA2:DUTYCYCLE:STATE 0;:MEASURE:BIT1:AREA2:FREQUENCY:STATE 0;:MEASURE:BIT1:AREA2:NWIDTH:STATE 0;:MEASURE:BIT1:AREA2:PERFREQUENCY:STATE 0;:MEASURE:BIT1:AREA2:PERIOD:STATE 0;:MEASURE:BIT1:AREA2:PWIDITH:STATE 0

### :MEASure:BIT<x>:AREA<x>?

機能 各エリアに関するすべての設定値を問い合わせます。

構文 :MEASure:BIT<x>:AREA<x>?

例 BIT<x>の<x> = 1 ~ 32(DL9510L、DL9505LはBIT<x>の<x> = 1 ~ 8、17 ~ 24が有効です。)  
AREA<x>の<x> = 1、2  
:MEASURE:BIT1:AREA1? -> :MEASURE:BIT1:AREA1:COUNT:STATE 0;:MEASURE:BIT1:AREA1:DELAY:MEASURE:COUNT 1;POLARITY RISE;:MEASURE:BIT1:AREA1:DELAY:REFERENCE:COUNT 1;POLARITY RISE;TRACE A0;:MEASURE:BIT1:AREA1:DELAY:SOURCE TRACE;STATE 0;:MEASURE:BIT1:AREA1:DT:STATE 0;:MEASURE:BIT1:AREA1:DUTYCYCLE:STATE 0;:MEASURE:BIT1:AREA1:FREQUENCY:STATE 0;:MEASURE:BIT1:AREA1:NWIDTH:STATE 0;:MEASURE:BIT1:AREA1:PERFREQUENCY:STATE 0;:MEASURE:BIT1:AREA1:PERIOD:STATE 0;:MEASURE:BIT1:AREA1:PWIDITH:STATE 0

## 5.19 MEASure グループ

### **:MEASure:BIT<x>:AREA<x>:ALL**

機能 ロジック波形パラメータのすべてを ON/OFF します。

構文 :MEASure:BIT<x>:AREA<x>:ALL  
{<Boolean>}  
BIT<x> の <x> = 1 ~ 32(DL9510L、DL9505L は BIT<x> の <x> = 1 ~ 8、17 ~ 24 が有効です。) AREA<x> の <x> = 1、2

例 :MEASURE:BIT1:AREA1:ALL ON

### **:MEASure:BIT<x>:AREA<x>:<パラメータ>?**

機能 ロジック波形パラメータに関するすべての設定値を問い合わせます。

構文 :MEASure:BIT<x>:AREA<x>:<パラメータ>?  
BIT<x> の <x> = 1 ~ 32(DL9510L、DL9505L は BIT<x> の <x> = 1 ~ 8、17 ~ 24 が有効です。) AREA<x> の <x> = 1、2  
<パラメータ> = {COUNT|DElay|DT|DUTYcycle|FREQuency|NWIDth|PERFrequency|PERiod|PWIDth}

例 (以下は、ビット 1、エリア 1 のカウントについての例です。)  
:MEASURE:BIT1:AREA1:COUNT?  
-> :MEASURE:BIT1:AREA1:COUNT:STATE 0

### **:MEASure:BIT<x>:AREA<x>:<パラメータ>:COUNT?**

機能 ロジック波形パラメータの継続統計処理の回数を問い合わせます。

構文 :MEASure:BIT<x>:AREA<x>:<パラメータ>:COUNT?  
BIT<x> の <x> = 1 ~ 32(DL9510L、DL9505L は BIT<x> の <x> = 1 ~ 8、17 ~ 24 が有効です。) AREA<x> の <x> = 1、2  
<パラメータ> = {COUNT|DElay|DT|DUTYcycle|FREQuency|NWIDth|PERFrequency|PERiod|PWIDth}

例 (以下は、ビット 1、エリア 1 のカウントについての例です。)  
:MEASURE:BIT1:AREA1:COUNT:COUNT?  
-> :MEASURE:BIT1:AREA1:COUNT:COUNT 0

### **:MEASure:BIT<x>:AREA<x>:<パラメータ>:{MAXimum|MEAN|MINimum|SDEVIation}?**

機能 ロジック波形パラメータの各統計値を問い合わせます。

構文 :MEASure:BIT<x>:AREA<x>:<パラメータ>:MAXimum?  
BIT<x> の <x> = 1 ~ 32(DL9510L、DL9505L は BIT<x> の <x> = 1 ~ 8、17 ~ 24 が有効です。) AREA<x> の <x> = 1、2  
<パラメータ> = {COUNT|DElay|DT|DUTYcycle|FREQuency|NWIDth|PERFrequency|PERiod|PWIDth}

例 (以下は、ビット 1、エリア 1 のカウントについての例です。)  
:MEASURE:BIT1:AREA1:COUNT:MAXIMUM?  
-> :MEASURE:BIT1:AREA1:COUNT:MAXIMUM 0

### **:MEASure:BIT<x>:AREA<x>:<パラメータ>:STATE**

機能 ロジック波形パラメータの ON/OFF を設定 / 問い合わせします。

構文 :MEASure:BIT<x>:AREA<x>:<パラメータ>:STATE {<Boolean>}  
:MEASure:BIT<x>:AREA<x>:<パラメータ>:STATE?  
BIT<x> の <x> = 1 ~ 32(DL9510L、DL9505L は BIT<x> の <x> = 1 ~ 8、17 ~ 24 が有効です。) AREA<x> の <x> = 1、2  
<パラメータ> = {COUNT|DElay|DT|DUTYcycle|FREQuency|NWIDth|PERFrequency|PERiod|PWIDth}

例 (以下は、ビット 1、エリア 1 のカウントについての例です。)  
:MEASURE:BIT1:AREA1:COUNT:STATE ON  
:MEASURE:BIT1:AREA1:COUNT:STATE?  
-> :MEASURE:BIT1:AREA1:COUNT:STATE 1

**:MEASure:BIT<x>:AREA<x>:<パラメータ>:VALUE?**

機能	ロジック波形パラメータの自動測定値を問い合わせます。
構文	:MEASure:BIT<x>:AREA<x>:<パラメータ>:VALUE? {<Nrf>} BIT<x>の<x> = 1 ~ 32(DL9510L、DL9505LはBIT<x>の<x> = 1 ~ 8、17 ~ 24が有効です。) AREA<x>の<x> = 1、2 <パラメータ> = {COUNT DElay DT DUTYcycle FREQuency NWIDth PERFrequency PERiod PWIDth} <Nrf> = 1 ~ 100000
例	(以下は、ビット1、エリア1のカウントについての例です。) :MEASURE:BIT1:AREA1:COUNT:VALUE? -> :MEASURE:BIT1:AREA1:COUNT:VALUE 0
解説	<ul style="list-style-type: none"> <li>測定不可能な場合は、「NAN(Not A Number)」が返されます。</li> <li>「&lt;Nrf&gt;」は、過去に測定した自動測定値の「&lt;Nrf&gt;」番目を指定します。周期統計処理の場合は、画面左から「&lt;Nrf&gt;」周期目を指定します。最古の自動測定値を指定する場合は、「1」を設定してください。「&lt;Nrf&gt;」を省略した場合は、最新の自動測定値の指定になります。該当回数目の値が存在しない場合には「NAN(Not A Number)」が返されます。</li> </ul>

**:MEASure:BIT<x>:AREA<x>:DElay:MEASure?**

機能	チャンネル間ディレイ測定の測定対象波形の測定条件に関するすべての設定値を問い合わせます。
構文	:MEASure:BIT<x>:AREA<x>:DElay:MEASure? BIT<x>の<x> = 1 ~ 32(DL9510L、DL9505LはBIT<x>の<x> = 1 ~ 8、17 ~ 24が有効です。) AREA<x>の<x> = 1、2
例	:MEASURE:BIT1:AREA1:DELAY:MEASURE? -> :MEASURE:BIT1:AREA1:DELAY:MEASURE:COUNT 1;POLARITY RISE

**:MEASure:BIT<x>:AREA<x>:DElay:MEASure:COUNT**

機能	チャンネル間ディレイ測定の測定対象波形のエッジ検出回数を設定 / 問い合わせします。
構文	:MEASure:BIT<x>:AREA<x>:DElay:MEASure:COUNT {<Nrf>} :MEASure:BIT<x>:AREA<x>:DElay:MEASure:COUNT? BIT<x>の<x> = 1 ~ 32(DL9510L、DL9505LはBIT<x>の<x> = 1 ~ 8、17 ~ 24が有効です。) AREA<x>の<x> = 1、2 <Nrf> = 1 ~ 10
例	:MEASURE:BIT1:AREA1:DELAY:MEASURE:COUNT 1 :MEASURE:BIT1:AREA1:DELAY:MEASURE:COUNT? -> :MEASURE:BIT1:AREA1:DELAY:MEASURE:COUNT 1

**:MEASure:BIT<x>:AREA<x>:DElay:MEASure:POLarity**

機能	チャンネル間ディレイ測定の測定対象波形の極性を設定 / 問い合わせします。
構文	:MEASure:BIT<x>:AREA<x>:DElay:MEASure:POLarity {FALL RISE} :MEASure:BIT<x>:AREA<x>:DElay:MEASure:POLarity? BIT<x>の<x> = 1 ~ 32(DL9510L、DL9505LはBIT<x>の<x> = 1 ~ 8、17 ~ 24が有効です。) AREA<x>の<x> = 1、2
例	:MEASURE:BIT1:AREA1:DELAY:MEASURE:POLARITY FALL :MEASURE:BIT1:AREA1:DELAY:MEASURE:POLARITY? -> :MEASURE:BIT1:AREA1:DELAY:MEASURE:POLARITY FALL

**:MEASure:BIT<x>:AREA<x>:DElay:REFERENCE?**

機能	チャンネル間ディレイ測定の基準波形に関するすべての設定値を問い合わせます。
構文	:MEASure:BIT<x>:AREA<x>:DElay:REFERENCE? BIT<x>の<x> = 1 ~ 32(DL9510L、DL9505LはBIT<x>の<x> = 1 ~ 8、17 ~ 24が有効です。) AREA<x>の<x> = 1、2
例	:MEASURE:BIT1:AREA1:DELAY:REFERENCE? -> :MEASURE:BIT1:AREA1:DELAY:REFERENCE:COUNT 1;POLARITY RISE;TRACE A0

## 5.19 MEASure グループ

### **:MEASure:BIT<x>:AREA<x>:DElay:**

#### **REfereNce:COUNt**

機能 チャンネル間ディレイ測定の基準波形のエッジ検出回数を設定 / 問い合わせします。

構文 :MEASure:BIT<x>:AREA<x>:DElay:REfereNce:COUNt {<NRf>}  
:MEASure:BIT<x>:AREA<x>:DElay:REfereNce:COUNt?  
BIT<x> の <x> = 1 ~ 32(DL9510L、DL9505L は BIT<x> の <x> = 1 ~ 8、17 ~ 24 が有効です。) AREA<x> の <x> = 1、2 <NRf> = 1 ~ 10

例 :MEASURE:BIT1:AREA1:DELAY:REFERENCE:COUNT 1  
:MEASURE:BIT1:AREA1:DELAY:REFERENCE:COUNT? -> :MEASURE:BIT1:AREA1:DELAY:REFERENCE:COUNT 1

### **:MEASure:BIT<x>:AREA<x>:DElay:**

#### **REfereNce:POLarity**

機能 チャンネル間ディレイ測定の基準波形の極性を設定 / 問い合わせします。

構文 :MEASure:BIT<x>:AREA<x>:DElay:REfereNce:POLarity {FALL|RISE}  
:MEASure:BIT<x>:AREA<x>:DElay:REfereNce:POLarity?  
BIT<x> の <x> = 1 ~ 32(DL9510L、DL9505L は BIT<x> の <x> = 1 ~ 8、17 ~ 24 が有効です。) AREA<x> の <x> = 1、2

例 :MEASURE:BIT1:AREA1:DELAY:REFERENCE:POLARITY FALL  
:MEASURE:BIT1:AREA1:DELAY:REFERENCE:POLARITY? -> :MEASURE:BIT1:AREA1:DELAY:REFERENCE:POLARITY FALL

### **:MEASure:BIT<x>:AREA<x>:DElay:**

#### **REfereNce:TRACe**

機能 チャンネル間ディレイ測定の基準波形のトレースを設定 / 問い合わせします。

構文 :MEASure:BIT<x>:AREA<x>:DElay:REfereNce:TRACe {<NRf>|A<y>|B<y>|C<y>|D<y>}  
:MEASure:BIT<x>:AREA<x>:DElay:REfereNce:TRACe?  
BIT<x> の <x> = 1 ~ 32(DL9510L、DL9505L は BIT<x> の <x> = 1 ~ 8、17 ~ 24 が有効です。) AREA<x> の <x> = 1、2 <NRf> = 1 ~ 8 <y> = 0 ~ 7

例 :MEASURE:BIT1:AREA1:DELAY:REFERENCE:TRACE 1  
:MEASURE:BIT1:AREA1:DELAY:REFERENCE:TRACE? -> :MEASURE:BIT1:AREA1:DELAY:REFERENCE:TRACE 1

解説 DL9510L、DL9505L は {<NRf>|A<y>|C<y>} が有効です。

### **:MEASure:BIT<x>:AREA<x>:DElay:SOURce**

機能 チャンネル間ディレイ測定の基準を設定 / 問い合わせします。

構文 :MEASure:BIT<x>:AREA<x>:DElay:SOURce {TRACe|TRIGGer}  
:MEASure:BIT<x>:AREA<x>:DElay:SOURce?  
BIT<x> の <x> = 1 ~ 32(DL9510L、DL9505L は BIT<x> の <x> = 1 ~ 8、17 ~ 24 が有効です。) AREA<x> の <x> = 1、2

例 :MEASURE:BIT1:AREA1:DELAY:SOURCE TRACE  
:MEASURE:BIT1:AREA1:DELAY:SOURCE? -> :MEASURE:BIT1:AREA1:DELAY:SOURCE TRACE

### **:MEASure:CALCulation?**

機能 Calc アイテムに関するすべての設定値を問い合わせます。

構文 :MEASure:CALCulation?

例 :MEASURE:CALCULATION? -> :MEASURE:CALCULATION:DEFINE1 "MAX (C1)";  
DEFINE2 "MIN (C2)";DEFINE3 "HIGH (C3)";  
DEFINE4 "LOW (C4)";STATE1 0;STATE2 0;  
STATE3 0;STATE4 0

### **:MEASure:CALCulation:ALL**

機能 Calc アイテムのすべてを ON/OFF します。

構文 :MEASure:CALCulation:ALL {<Boolean>}

例 :MEASURE:CALCULATION:ALL ON

### **:MEASure:CALCulation:COUNt<x>?**

機能 Calc アイテムの統計処理の回数を問い合わせます。

構文 :MEASure:CALCulation:COUNt<x>?  
<x> = 1 ~ 4

例 :MEASURE:CALCULATION:COUNT1?  
-> :MEASURE:CALCULATION:COUNT1 1

**:MEASure:CALCulation:DEFine<x>**

機能 Calc アイテムの演算式を設定 / 問い合わせします。

構文 :MEASure:CALCulation:DEFine<x> {<文字列>}  
:MEASure:CALCulation:DEFine<x>?  
<x> = 1 ~ 4  
<文字列> = 128 文字以内

例 :MEASURE:CALCULATION:DEFINE1  
"MAX(C1)"  
:MEASURE:CALCULATION:DEFINE1?  
-> :MEASURE:CALCULATION:  
DEFINE1 "MAX(C1)"

**:MEASure:CALCulation:{MAXimum<x>|MEAN<x>|MINimum<x>|SDEVIation<x>}?**

機能 Calc アイテムの各統計値を問い合わせます。

構文 :MEASure:CALCulation:{MAXimum<x>|MEAN<x>|MINimum<x>|SDEVIation<x>}?  
MAXimum<x> の <x> = 1 ~ 4  
MEAN<x> の <x> = 1 ~ 4  
MINimum<x> の <x> = 1 ~ 4  
SDEVIation<x> の <x> = 1 ~ 4

例 (以下は、最大値についての例です。)  
:MEASURE:CALCULATION:MAXIMUM1?  
-> :MEASURE:CALCULATION:  
MAXIMUM1 1.000E+00

**:MEASure:CALCulation:STATe<x>**

機能 Calc アイテムの ON/OFF を設定 / 問い合わせします。

構文 :MEASure:CALCulation:STATe<x>  
{<Boolean>}  
:MEASure:CALCulation:STATe<x>?  
<x> = 1 ~ 4

例 :MEASURE:CALCULATION:STATE1 ON  
:MEASURE:CALCULATION:STATE1?  
-> :MEASURE:CALCULATION:STATE1 1

**:MEASure:CALCulation:VALue<x>?**

機能 Calc アイテムの自動測定値を問い合わせます。

構文 :MEASure:CALCulation:VALue<x>?  
{<NRF>}  
<x> = 1 ~ 4  
<NRF> = 1 ~ 100000

例 :MEASURE:CALCULATION:VALUE1?  
-> :MEASURE:CALCULATION:  
VALUE1 1.000E+00

解説

- 測定不可能な場合は、「NAN(Not A Number)」が返されます。
- 「<NRF>」は、過去に測定した自動測定値の「<NRF>」番目を指定します。周期統計処理の場合は、画面左から「<NRF>」周期目を指定します。最古の自動測定値を指定する場合は、「1」を設定してください。「<NRF>」を省略した場合は、最新の自動測定値の指定になります。該当回数目の値が存在しない場合には「NAN(Not A Number)」が返されます。

**:MEASure:CONTinuous?**

機能 継続統計処理 (CONTinuous Statistics) に関するすべての設定値を問い合わせます。

構文 :MEASure:CONTinuous?  
例 :MEASURE:CONTINUOUS? -> :MEASURE:  
CONTINUOUS:COUNT 0

**:MEASure:CONTinuous:COUNT**

機能 継続統計処理 (CONTinuous Statistics) の回数を設定 / 問い合わせします。

構文 :MEASure:CONTinuous:COUNT {<NRF>}  
:MEASure:CONTinuous:COUNT?  
<NRF> = 0 ~ 100000

例 :MEASURE:CONTINUOUS:COUNT 10  
:MEASURE:CONTINUOUS:COUNT?  
-> :MEASURE:CONTINUOUS:COUNT 10

解説 <NRF> = 0 のときは、現設定で採りうる最大の回数を自動的に設定します。

**:MEASure:CONTinuous:REStart**

機能 継続統計処理 (CONTinuous Statistics) を再スタートします。

構文 :MEASure:CONTinuous:REStart  
例 :MEASURE:CONTINUOUS:RESTART  
解説 以前の統計データをクリアします。

**:MEASure:CYCLE?**

機能 周期統計処理 (Cycle Statistics) に関するすべての設定を問い合わせます。

構文 :MEASure:CYCLE?  
例 :MEASURE:CYCLE? -> :MEASURE:CYCLE:  
TRACE 1

**:MEASure:CYCLE:ABORT**

機能 周期統計処理 (Cycle Statistics) の実行を中止します。

構文 :MEASure:CYCLE:ABORT  
例 :MEASURE:CYCLE:ABORT

**:MEASure:CYCLE:EXECute**

機能 周期統計処理 (Cycle Statistics) を実行します。オーバーラップコマンドです。

構文 :MEASure:CYCLE:EXECute  
例 :MEASURE:CYCLE:EXECUTE  
解説 以前の統計データをクリアせず、継続して実行します。



## 5.19 MEASure グループ

### :MEASure:CYCLE:TRACe

機能 周期統計処理 (Cycle Statistics) の周期対象トレースを設定 / 問い合わせます。

構文 :MEASure:CYCLE:TRACe  
{<NRf>|A<x>|B<x>|  
C<x>|D<x>}

:MEASure:CYCLE:TRACe?

<NRf> = 1 ~ 8

<x> = 0 ~ 7

例 :MEASURE:CYCLE:TRACE 1  
:MEASURE:CYCLE:TRACE? -> :MEASURE:  
CYCLE:TRACE 1

解説 DL9510L、DL9505L は {<NRf>|A<x>|C<x>} が有効です。

### :MEASure:DISPlay

機能 波形パラメータの自動測定結果の表示の ON/OFF を設定 / 問い合わせします。

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

例 :MEASURE:DISPLAY ON  
:MEASURE:DISPLAY? -> :MEASURE:  
DISPLAY 1

### :MEASure:HISTory:ABORt

機能 ヒストリデータの統計処理 (History Statistics) の実行を中止します。

構文 :MEASure:HISTory:ABORt  
例 :MEASURE:HISTORY:ABORT

### :MEASure:HISTory:EXECute

機能 ヒストリデータの統計処理 (History Statistics) を実行します。オーバーラップコマンドです。

構文 :MEASure:HISTory:EXECute  
例 :MEASURE:HISTORY:EXECUTE

### :MEASure:MODE

機能 波形パラメータの自動測定モードを設定 / 問い合わせします。

構文 :MEASure:MODE  
{BASic|CONTinuous|CYCLE|  
HISTory}

:MEASure:MODE?

例 :MEASURE:MODE BASIC  
:MEASURE:MODE? -> :MEASURE:MODE BASIC

### :MEASure:THReshold?

機能 波形パラメータの自動測定しきい値 (Threshold) に関するすべての設定値を問い合わせます。

構文 :MEASure:THReshold?  
例 :MEASURE:THRESHOLD? -> :MEASURE:  
THRESHOLD:TRACE1:AUTO PTOPEAK;  
LHYSTERESIS:HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;;MEASURE:THRESHOLD:  
TRACE1:MODE AUTO;ULOWER:  
RANGE 2.000E+00,1.000E+00;;MEASURE:  
THRESHOLD:TRACE2:AUTO PTOPEAK;  
LHYSTERESIS:HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;;MEASURE:THRESHOLD:  
TRACE2:MODE AUTO;ULOWER:  
RANGE 2.000E+00,1.000E+00;;MEASURE:  
THRESHOLD:TRACE3:AUTO PTOPEAK;  
LHYSTERESIS:HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;;MEASURE:THRESHOLD:  
TRACE3:MODE AUTO;ULOWER:  
RANGE 2.000E+00,1.000E+00;;MEASURE:  
THRESHOLD:TRACE4:AUTO PTOPEAK;  
LHYSTERESIS:HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;;MEASURE:THRESHOLD:  
TRACE4:MODE AUTO;ULOWER:  
RANGE 2.000E+00,1.000E+00;;MEASURE:  
THRESHOLD:TRACE5:AUTO PTOPEAK;  
LHYSTERESIS:HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;;MEASURE:THRESHOLD:  
TRACE5:MODE AUTO;ULOWER:  
RANGE 2.000E+00,1.000E+00;;MEASURE:  
THRESHOLD:TRACE6:AUTO PTOPEAK;  
LHYSTERESIS:HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;;MEASURE:THRESHOLD:  
TRACE6:MODE AUTO;ULOWER:RANGE . . . . .

解説 「:MEASure:TRACe<x>:AREA<x>:<パラメータ>」の<パラメータ>が「RISE」/「FALL」以外のときに有効です。

### :MEASure:THReshold:TRACe<x>?

機能 各トレースのしきい値 (Threshold) に関するすべての設定値を問い合わせます。

構文 :MEASure:THReshold:TRACe<x>?  
<x> = 1 ~ 8

例 :MEASURE:THRESHOLD:TRACE1?  
-> :MEASURE:THRESHOLD:TRACE1:  
AUTO PTOPEAK;LHYSTERESIS:  
HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;;MEASURE:THRESHOLD:  
TRACE1:MODE AUTO;ULOWER:  
RANGE 1.000E+00,2.000E+00

解説 「:MEASure:TRACe<x>:AREA<x>:<パラメータ>」の<パラメータ>が「RISE」/「FALL」以外のときに有効です。

**:MEASure:THReshold:TRACe<x>:AUTO**

機能	しきい値 (Threshold) の自動設定時の検出方法を設定 / 問い合わせします。
構文	:MEASure:THReshold:TRACe<x>: AUTO {HLOW PTOPeak} :MEASure:THReshold:TRACe<x>:AUTO? <x> = 1 ~ 8
例	:MEASURE:THRESHOLD:TRACE1:AUTO PTOPEAK :MEASURE:THRESHOLD:TRACE1:AUTO? -> :MEASURE:THRESHOLD:TRACE1: AUTO PTOPEAK
解説	「:MEASure:TRACe<x>:AREA<x>:<パラメータ>」の<パラメータ>が「RISE」 / 「FALL」以外のときに有効です。

**:MEASure:THReshold:TRACe<x>:  
LHYSteresis?**

機能	しきい値 (Threshold) のレベル、ヒステリシス設定におけるすべての設定値を問い合わせます。
構文	:MEASure:THReshold:TRACe<x>: LHYSteresis? <x> = 1 ~ 8
例	:MEASURE:THRESHOLD:TRACE1: LHYSTERESIS? -> :MEASURE:THRESHOLD: TRACE1:LHYSTERESIS: HYSTERESIS 1.000E+00; LEVEL 0.000E+00
解説	「:MEASure:TRACe<x>:AREA<x>:<パラメータ>」の<パラメータ>が「RISE」 / 「FALL」以外のときに有効です。

**:MEASure:THReshold:TRACe<x>:  
LHYSteresis:HYSteresis**

機能	しきい値 (Threshold) のヒステリシスを設定 / 問い合わせします。
構文	:MEASure:THReshold:TRACe<x>: LHYSteresis:HYSteresis {<NRf>} :MEASure:THReshold:TRACe<x>: LHYSteresis:HYSteresis? <x> = 1 ~ 8 <NRf> = 0 ~ 4(div)
例	:MEASURE:THRESHOLD:TRACE1: LHYSTERESIS: HYSTERESIS 1 :MEASURE:THRESHOLD:TRACE1: LHYSTERESIS:HYSTERESIS? -> :MEASURE: THRESHOLD:TRACE1:LHYSTERESIS: HYSTERESIS 1.000E+00
解説	「:MEASure:TRACe<x>:AREA<x>:<パラメータ>」の<パラメータ>が「RISE」 / 「FALL」以外のときに有効です。

**:MEASure:THReshold:TRACe<x>:  
LHYSteresis:LEVel**

機能	しきい値 (Threshold) のレベルを設定 / 問い合わせします。
構文	:MEASure:THReshold:TRACe<x>: LHYSteresis:LEVel {<NRf> <電圧>  <電流>} :MEASure:THReshold:TRACe<x>: LHYSteresis:LEVel? <x> = 1 ~ 8 <NRf>、<電圧>、<電流> = 本体ユーザーズマニュアル参照。
例	:MEASURE:THRESHOLD:TRACE1: LHYSTERESIS:LEVEL 1 :MEASURE:THRESHOLD:TRACE1: LHYSTERESIS:LEVEL? -> :MEASURE: THRESHOLD:TRACE1:LHYSTERESIS: LEVEL 1.000E+00
解説	「:MEASure:TRACe<x>:AREA<x>:<パラメータ>」の<パラメータ>が「RISE」 / 「FALL」以外のときに有効です。

**:MEASure:THReshold:TRACe<x>:MODE**

機能	しきい値 (Threshold) の設定方法を問い合わせます。
構文	:MEASure:THReshold:TRACe<x>: MODE {AUTO LHYSteresis ULOWer} :MEASure:THReshold:TRACe<x>:MODE? <x> = 1 ~ 8
例	:MEASURE:THRESHOLD:TRACE1: MODE LHYSTERESIS :MEASURE:THRESHOLD:TRACE1:MODE? -> :MEASURE:THRESHOLD:TRACE1: MODE LHYSTERESIS
解説	「:MEASure:TRACe<x>:AREA<x>:<パラメータ>」の<パラメータ>が「RISE」 / 「FALL」以外のときに有効です。

**:MEASure:THReshold:TRACe<x>:ULOWer?**

機能	しきい値 (Threshold) の上下限值設定におけるすべての設定値を問い合わせます。
構文	:MEASure:THReshold:TRACe<x>:ULOWer? <x> = 1 ~ 8
例	:MEASURE:THRESHOLD:TRACE1:ULOWER? -> :MEASURE:THRESHOLD:TRACE1:ULOWER: RANGE 2.000E+00,1.000E+00
解説	「:MEASure:TRACe<x>:AREA<x>:<パラメータ>」の<パラメータ>が「RISE」 / 「FALL」以外のときに有効です。

## 5.19 MEASure グループ

### :MEASure:THReshold:TRACe<x>:ULOWer:

#### RANGE

機能 しきい値 (Threshold) の上下限值を設定 / 問い合わせします。

構文 :MEASure:THReshold:TRACe<x>:ULOWer:

RANGE { (<NRf>, <NRf>) | (<電圧>, <電圧>) | (<電流>, <電流>) }

:MEASure:THReshold:TRACe<x>:ULOWer: RANGE?

<x> = 1 ~ 8

<NRf>, <電圧>, <電流> = 本体ユーザズマニュアル参照。

例 :MEASURE:THRESHOLD:TRACE1:ULOWER:

RANGE 1,2

:MEASURE:THRESHOLD:TRACE1:ULOWER:

RANGE? -> :MEASURE:THRESHOLD:TRACE1:

ULOWER:RANGE 2.000E+00,1.000E+00

解説 「:MEASure:TRACe<x>:AREA<x>:<パラメータ>」の<パラメータ>が「RISE」/「FALL」以外のときに有効です。

### :MEASure:TRACe<x>?

機能 各トレースに関するすべての設定値を問い合わせます。

構文 :MEASure:TRACe<x>?

<x> = 1 ~ 8

例 :MEASURE:TRACE1? -> :MEASURE:TRACE1: AREA1:BURST:STATE 0;:MEASURE:TRACE1: AREA1:CMEAN:STATE 0;:MEASURE:TRACE1: AREA1:CMODE 0;COUNT:STATE 0;:MEASURE: TRACE1:AREA1:CRMS:STATE 0;:MEASURE: TRACE1:AREA1:CSDEVIATION:STATE 0;: MEASURE:TRACE1:AREA1:DELAY:MEASURE: COUNT 1;POLARITY RISE;:MEASURE: TRACE1:AREA1:DELAY:REFERENCE:COUNT 1; POLARITY RISE;TRACE 1;:MEASURE: TRACE1:AREA1:DELAY:SOURCE TRIGGER; STATE 0;:MEASURE:TRACE1:AREA1: DPROXIMAL:MODE PERCENT;PERCENT 10,90; UNIT -3.0000000E+00,3.0000000E+00;: MEASURE:TRACE1:AREA1:DT:STATE 0;: MEASURE:TRACE1:AREA1:DUTYCYCLE: STATE 0;:MEASURE:TRACE1:AREA1:FALL: STATE 0;:MEASURE:TRACE1:AREA1: FREQUENCY:STATE 0;:MEASURE:TRACE1: AREA1:HIGH:STATE 0;:MEASURE:TRACE1: AREA1:HILOW:STATE 0;:MEASURE:TRACE1: AREA1:LOW:STATE 0;:MEASURE:TRACE1: AREA1:MAXIMUM:STATE 0;:MEASURE: TRACE1:AREA1:MEAN:STATE 0;:MEASURE: TRACE1:AREA1:METHOD AUTO;MINIMUM: STATE 0;:MEASURE:TRACE1:AREA1: NOVERSHOOT:STATE 0;:MEASURE:TRACE1: AREA1:NWIDTH:STATE 0;:MEASURE:TRACE1: AREA1:PERFREQUENCY:STATE 0;:MEASURE: TRACE1:AREA1:PERIOD:STATE 0;:MEASURE: TRACE1:AREA1:POVERSHOOT:STATE 0;: MEASURE:TRACE1:AREA1:PTOPEAK: STATE 0;:MEASURE:TRACE1:AREA1:PWIDTH: STATE 0;:MEASURE:TRACE1:AREA1:RISE: STATE 0;:MEASURE:TRACE1:AREA1:RMS: STATE 0 . . . . .

**:MEASure:TRACe<x>:AREA<x>?**

機能 各エリアに関するすべての設定値を問い合わせます。

構文 :MEASure:TRACe<x>:AREA<x>?

TRACe<x> の <x> = 1 ~ 8

AREA<x> の <x> = 1、2

例 :MEASURE:TRACE1:AREA1? -> :MEASURE:TRACE1:AREA1:BURST:STATE 0;;  
 MEASURE:TRACE1:AREA1:CMEAN:STATE 0;;  
 MEASURE:TRACE1:AREA1:CMODE 0;COUNT:STATE 0;;MEASURE:TRACE1:AREA1:CRMS:STATE 0;;MEASURE:TRACE1:AREA1:CSDEVIATION:STATE 0;;MEASURE:TRACE1:AREA1:DELAY:MEASURE:COUNT 1;  
 POLARITY RISE;;MEASURE:TRACE1:AREA1:DELAY:REFERENCE:COUNT 1;  
 POLARITY RISE;TRACE 1;;MEASURE:TRACE1:AREA1:DELAY:SOURCE TRIGGER;STATE 0;;MEASURE:TRACE1:AREA1:DPROXIMAL:MODE PERCENT;PERCENT 10,90;UNIT -3.0000000E+00,3.0000000E+00;;MEASURE:TRACE1:AREA1:DT:STATE 0;;MEASURE:TRACE1:AREA1:DUTYCYCLE:STATE 0;;MEASURE:TRACE1:AREA1:FALL:STATE 0;;MEASURE:TRACE1:AREA1:FREQUENCY:STATE 0;;MEASURE:TRACE1:AREA1:HIG:STATE 0;;MEASURE:TRACE1:AREA1:HILOW:STATE 0;;MEASURE:TRACE1:AREA1:LOW:STATE 0;;MEASURE:TRACE1:AREA1:MAXIMUM:STATE 0;;MEASURE:TRACE1:AREA1:MEAN:STATE 0;;MEASURE:TRACE1:AREA1:METH:METHOD AUTO;MINIMUM:STATE 0;;MEASURE:TRACE1:AREA1:NOVERSHOOT:STATE 0;;MEASURE:TRACE1:AREA1:NWIDTH:STATE 0;;MEASURE:TRACE1:AREA1:PERF:PERFREQUENCY:STATE 0;;MEASURE:TRACE1:AREA1:PERIOD:STATE 0;;MEASURE:TRACE1:AREA1:POVERSHOOT:STATE 0 .....

**:MEASure:TRACe<x>:AREA<x>:ALL**

機能 波形パラメータのすべてを ON/OFF します。

構文 :MEASure:TRACe<x>:AREA<x>:ALL

{<Boolean>}

TRACe<x> の <x> = 1 ~ 8

AREA<x> の <x> = 1、2

例 :MEASURE:TRACE1:AREA1:ALL ON

**:MEASure:TRACe<x>:AREA<x>:****<パラメータ>?**

機能 波形パラメータに関するすべての設定値を問い合わせます。

構文 :MEASure:TRACe<x>:AREA<x>:

<パラメータ>?

TRACe<x> の <x> = 1 ~ 8

AREA<x> の <x> = 1、2

<パラメータ> = {BURSt|CMEan|COUNT|CRMS|CSDeviation|DElay|DT|DUTYcycle|FALL|FREQuency|HIGH|HILOW|LOW|MAXimum|MEAN|MINimum|NOVershoot|NWIDTH|PERFrequency|PERiod|POVershoot|PTOPeak|PWIDTH|RISE|RMS|SDEviation|TYCInteg|TYINteg|V1|V2}

例 (以下は、トレース 1、エリア 1 の最大値についての例です。)

```
:MEASURE:TRACE1:AREA1:MAXIMUM?
-> :MEASURE:TRACE1:AREA1:MAXIMUM:STATE 0
```

**:MEASure:TRACe<x>:AREA<x>:****<パラメータ>:COUNT?**

機能 波形パラメータの継続統計処理の回数を問い合わせます。

構文 :MEASure:TRACe<x>:AREA<x>:<パラメータ>:COUNT?

TRACe<x> の <x> = 1 ~ 8

AREA<x> の <x> = 1、2

<パラメータ> = {BURSt|CMEan|COUNT|CRMS|CSDeviation|DElay|DT|DUTYcycle|FALL|FREQuency|HIGH|HILOW|LOW|MAXimum|MEAN|MINimum|NOVershoot|NWIDTH|PERFrequency|PERiod|POVershoot|PTOPeak|PWIDTH|RISE|RMS|SDEviation|TYCInteg|TYINteg|V1|V2}

例 (以下は、トレース 1、エリア 1 の最大値についての例です。)

```
:MEASURE:TRACE1:AREA1:MAXIMUM:COUNT?
-> :MEASURE:TRACE1:AREA1:MAXIMUM:COUNT 0
```

## 5.19 MEASure グループ

### **:MEASure:TRACe<x>:AREA<x>: <パラメータ>:{MAXimum|MEAN|MINimum| SDEVIation}?**

機能 波形パラメータの各統計値を問い合わせます。  
構文 :MEASure:TRACe<x>:AREA<x>:<パラメータ>:{MAXimum|MEAN|MINimum|SDEVIation}?

TRACe<x>の<x> = 1 ~ 8  
AREA<x>の<x> = 1, 2  
<パラメータ> = {BURSt|CMEan|COUNT|CRMS|  
CSDeviation|DElay|DT|DUTYcycle|FALL|  
FREQuency|HIGH|HILow|LOW|MAXimum|MEAN|  
MINimum|NOVershoot|NWIDTH|PERFrequency|  
PERiod|POVershoot|PTOPeak|PWIDth|RISE|  
RMS|SDEVIation|TYCInteg|TYINteg|V1|V2}

例 (以下は、最大値についての例です。)  
:MEASure:TRACe1:AREA1:MAXIMUM:  
MAXIMUM? -> :MEASure:TRACe1:AREA1:  
MAXIMUM:MAXIMUM 1.000E+00

### **:MEASure:TRACe<x>:AREA<x>: <パラメータ>:STATe**

機能 波形パラメータの ON/OFF を設定 / 問い合わせします。  
構文 :MEASure:TRACe<x>:AREA<x>:<パラメータ>:STATe {<Boolean>}

:MEASure:TRACe<x>:AREA<x>:<パラメータ>:STATe?  
TRACe<x>の<x> = 1 ~ 8  
AREA<x>の<x> = 1, 2  
<パラメータ> = {BURSt|CMEan|COUNT|CRMS|  
CSDeviation|DElay|DT|DUTYcycle|FALL|  
FREQuency|HIGH|HILow|LOW|MAXimum|MEAN|  
MINimum|NOVershoot|NWIDTH|PERFrequency|  
PERiod|POVershoot|PTOPeak|PWIDth|RISE|  
RMS|SDEVIation|TYCInteg|TYINteg|V1|V2}

例 (以下は、トレース 1、エリア 1 の最大値についての例です。)  
:MEASure:TRACe1:AREA1:MAXIMUM:STATe  
ON  
:MEASure:TRACe1:AREA1:MAXIMUM:STATe?  
-> :MEASure:TRACe1:AREA1:MAXIMUM:  
STATe 1

### **:MEASure:TRACe<x>:AREA<x>: <パラメータ>:VALue?**

機能 波形パラメータの自動測定値を問い合わせます。  
構文 :MEASure:TRACe<x>:AREA<x>:<パラメータ>:VALue? {<NRF>}

TRACe<x>の<x> = 1 ~ 8  
AREA<x>の<x> = 1, 2  
<パラメータ> = {BURSt|CMEan|COUNT|CRMS|  
CSDeviation|DElay|DT|DUTYcycle|FALL|  
FREQuency|HIGH|HILow|LOW|MAXimum|MEAN|  
MINimum|NOVershoot|NWIDTH|PERFrequency|  
PERiod|POVershoot|PTOPeak|PWIDth|RISE|  
RMS|SDEVIation|TYCInteg|TYINteg|V1|V2}

例 (以下は、トレース 1、エリア 1 の最大値についての例です。)  
:MEASure:TRACe1:AREA1:MAXIMUM:VALue?  
-> :MEASure:TRACe1:AREA1:MAXIMUM:  
VALue 1.000E+00

解説

- 測定不可能な場合は、「NAN(Not A Number)」が返されます。
- 「<NRF>」は、過去に測定した自動測定値の「<NRF>」番目を指定します。周期統計処理の場合は、画面左から「<NRF>」周期目を指定します。最古の自動測定値を指定する場合は、「1」を設定してください。「<NRF>」を省略した場合は、最新の自動測定値の指定になります。該当回数目の値が存在しない場合には「NAN(Not A Number)」が返されます。

### **:MEASure:TRACe<x>:AREA<x>:DElay: MEASure?**

機能 チャネル間ディレイ測定の測定対象波形の測定条件に関するすべての設定値を問い合わせます。  
構文 :MEASure:TRACe<x>:AREA<x>:DElay:MEASure?

TRACe<x>の<x> = 1 ~ 8  
AREA<x>の<x> = 1, 2

例 :MEASure:TRACe1:AREA1:DElay:MEASure?  
-> :MEASure:TRACe1:AREA1:DElay:  
MEASure:COUNT 1;POLARITY RISE

**:MEASure:TRACe<x>:AREA<x>:DELay:  
MEASure:COUNT**

機能 チャンネル間ディレイ測定の測定対象波形のエッジ検出回数を設定 / 問い合わせします。

構文 :MEASure:TRACe<x>:AREA<x>:DELay:  
MEASure:COUNT {<NRf>}  
:MEASure:TRACe<x>:AREA<x>:DELay:  
MEASure:COUNT?

TRACe<x> の <x> = 1 ~ 8

AREA<x> の <x> = 1, 2

<NRf> = 1 ~ 10

例 :MEASure:TRACe1:AREA1:DELay:MEASure:  
COUNT 1

:MEASure:TRACe1:AREA1:DELay:MEASure:  
COUNT? -> :MEASure:TRACe1:AREA1:  
DELay:MEASure:COUNT 1

**:MEASure:TRACe<x>:AREA<x>:DELay:  
MEASure:POLarity**

機能 チャンネル間ディレイ測定の測定対象波形の極性を設定 / 問い合わせします。

構文 :MEASure:TRACe<x>:AREA<x>:DELay:  
MEASure:POLarity {FALL|RISE}  
:MEASure:TRACe<x>:AREA<x>:DELay:  
MEASure:POLarity?

TRACe<x> の <x> = 1 ~ 8

AREA<x> の <x> = 1, 2

例 :MEASure:TRACe1:AREA1:DELay:MEASure:  
POLarity RISE

:MEASure:TRACe1:AREA1:DELay:MEASure:  
POLarity? -> :MEASure:TRACe1:AREA1:  
DELay:MEASure:POLarity RISE

**:MEASure:TRACe<x>:AREA<x>:DELay:  
REFerence?**

機能 チャンネル間ディレイ測定の基準波形に関するすべての設定値を問い合わせます。

構文 :MEASure:TRACe<x>:AREA<x>:DELay:  
REFerence?

TRACe<x> の <x> = 1 ~ 8

AREA<x> の <x> = 1, 2

例 :MEASure:TRACe1:AREA1:DELay:

REFerence? -> :MEASure:TRACe1:AREA1:  
DELay:REFerence:COUNT 1;  
POLarity FALL;TRACE 1

**:MEASure:TRACe<x>:AREA<x>:DELay:  
REFerence:COUNT**

機能 チャンネル間ディレイ測定の基準波形のエッジ検出回数を設定 / 問い合わせします。

構文 :MEASure:TRACe<x>:AREA<x>:DELay:  
REFerence:COUNT {<NRf>}  
:MEASure:TRACe<x>:AREA<x>:DELay:  
REFerence:COUNT?

TRACe<x> の <x> = 1 ~ 8

AREA<x> の <x> = 1, 2

<NRf> = 1 ~ 10

例 :MEASure:TRACe1:AREA1:DELay:

REFerence:COUNT 1  
:MEASure:TRACe1:AREA1:DELay:  
REFerence:COUNT? -> :MEASure:TRACe1:  
AREA1:DELay:REFerence:COUNT 1

**:MEASure:TRACe<x>:AREA<x>:DELay:  
REFerence:POLarity**

機能 チャンネル間ディレイ測定の基準波形の極性を設定 / 問い合わせします。

構文 :MEASure:TRACe<x>:AREA<x>:DELay:  
REFerence:POLarity {FALL|RISE}  
:MEASure:TRACe<x>:AREA<x>:DELay:  
REFerence:POLarity?

TRACe<x> の <x> = 1 ~ 8

AREA<x> の <x> = 1, 2

例 :MEASure:TRACe1:AREA1:DELay:

REFerence:POLarity FALL  
:MEASure:TRACe1:AREA1:DELay:  
REFerence:POLarity? -> :MEASure:  
TRACe1:AREA1:DELay:REFerence:  
POLarity FALL

**:MEASure:TRACe<x>:AREA<x>:DELay:  
REFerence:TRACe**

機能 チャンネル間ディレイ測定の基準波形のトレースを設定 / 問い合わせします。

構文 :MEASure:TRACe<x>:AREA<x>:DELay:  
REFerence:TRACe {<NRf>|A<y>|B<y>|  
C<y>|D<y>}

:MEASure:TRACe<x>:AREA<x>:DELay:

REFerence:TRACe?

TRACe<x> の <x> = 1 ~ 8

AREA<x> の <x> = 1, 2

<NRf> = 1 ~ 8

<y> = 0 ~ 7

例 :MEASure:TRACe1:AREA1:DELay:

REFerence:  
TRACE 1  
:MEASure:TRACe1:AREA1:DELay:  
REFerence:TRACe? -> :MEASure:TRACe1:  
AREA1:DELay:REFerence:TRACE 1

解説 DL9510L、DL9505L は {<NRf>|A<y>|C<y>} が有効です。

## 5.19 MEASure グループ

### **:MEASure:TRACe<x>:AREA<x>:DElay:SOURCE**

機能 チャンネル間ディレイ測定の基準を設定 / 問い合わせします。

構文 :MEASure:TRACe<x>:AREA<x>:DElay:SOURCE {TRACe|TRIGger}  
:MEASure:TRACe<x>:AREA<x>:DElay:SOURCE?

TRACe<x> の <x> = 1 ~ 8  
AREA<x> の <x> = 1, 2

例 :MEASURE:TRACE1:AREA1:DELAY:  
SOURCE TRACE  
:MEASURE:TRACE1:AREA1:DELAY:SOURCE?  
-> :MEASURE:TRACE1:AREA1:DELAY:  
SOURCE TRACE

### **:MEASure:TRACe<x>:AREA<x>:DPRoximal?**

機能 ディスタル / プロキシマル値に関するすべての設定値を問い合わせます。

構文 :MEASure:TRACe<x>:AREA<x>:DPRoximal?  
TRACe<x> の <x> = 1 ~ 8  
AREA<x> の <x> = 1, 2

例 :MEASURE:TRACE1:AREA1:DPROXIMAL?  
-> ;MEASURE:TRACE1:AREA1:DPROXIMAL:  
MODE PERCENT;PERCENT 10,20;  
UNIT 1.000E+00,1.000E+00

解説 「:MEASure:TRACe<x>:AREA<x>:<パラメータ>」の<パラメータ>が「RISE」 / 「FALL」のときに有効です。

### **:MEASure:TRACe<x>:AREA<x>:DPRoximal:MODE**

機能 ディスタル / プロキシマル値の単位を設定 / 問い合わせします。

構文 :MEASure:TRACe<x>:AREA<x>:DPRoximal:MODE {PERCent|UNIT}  
:MEASure:TRACe<x>:AREA<x>:DPRoximal:MODE?

TRACe<x> の <x> = 1 ~ 8  
AREA<x> の <x> = 1, 2

例 :MEASURE:TRACE1:AREA1:DPROXIMAL:  
MODE PERCENT  
:MEASURE:TRACE1:AREA1:DPROXIMAL:MODE?  
-> :MEASURE:TRACE1:AREA1:DPROXIMAL:  
MODE PERCENT

解説 「:MEASure:TRACe<x>:AREA<x>:<パラメータ>」の<パラメータ>が「RISE」 / 「FALL」のときに有効です。

### **:MEASure:TRACe<x>:AREA<x>:DPRoximal:PERCent**

機能 ディスタル / プロキシマル値を % で設定 / 問い合わせします。

構文 :MEASure:TRACe<x>:AREA<x>:DPRoximal:PERCent {<NRf>,<NRf>}  
:MEASure:TRACe<x>:AREA<x>:DPRoximal:PERCent?

TRACe<x> の <x> = 1 ~ 8  
AREA<x> の <x> = 1, 2  
<NRf> = 0 ~ 100(%)

例 :MEASURE:TRACE1:AREA1:DPROXIMAL:  
PERCENT 10,90  
:MEASURE:TRACE1:AREA1:DPROXIMAL:  
PERCENT? -> :MEASURE:TRACE1:AREA1:  
DPROXIMAL:PERCENT 10,90

解説 「:MEASure:TRACe<x>:AREA<x>:<パラメータ>」の<パラメータ>が「RISE」 / 「FALL」のときに有効です。

### **:MEASure:TRACe<x>:AREA<x>:DPRoximal:UNIT**

機能 ディスタル / プロキシマル値を UNIT で設定 / 問い合わせします。

構文 :MEASure:TRACe<x>:AREA<x>:DPRoximal:UNIT {(<NRf>,<NRf>)|(<電圧>,<電圧>)|(<電流>,<電流>)}  
:MEASure:TRACe<x>:AREA<x>:DPRoximal:UNIT?

TRACe<x> の <x> = 1 ~ 8  
AREA<x> の <x> = 1, 2  
<NRf>,<電圧>,<電流> = 本体ユーザズマニュアル参照。

例 :MEASURE:TRACE1:AREA1:DPROXIMAL:  
UNIT 1,-1  
:MEASURE:TRACE1:AREA1:DPROXIMAL:UNIT?  
-> :MEASURE:TRACE1:AREA1:DPROXIMAL:  
UNIT -1.000E+00,1.000E+00

解説 「:MEASure:TRACe<x>:AREA<x>:<パラメータ>」の<パラメータ>が「RISE」 / 「FALL」のときに有効です。

### **:MEASure:TRACe<x>:AREA<x>:METHod**

機能 波形パラメータの自動測定の High/Low レベルの検出方法を設定 / 問い合わせします。

構文 :MEASure:TRACe<x>:AREA<x>:METHod {AUTO|HISTogram|MAXMin}  
:MEASure:TRACe<x>:AREA<x>:METHod?

TRACe<x> の <x> = 1 ~ 8  
AREA<x> の <x> = 1,2

例 :MEASURE:TRACE1:AREA1:METHOD AUTO  
:MEASURE:TRACE1:AREA1:METHOD?  
-> :MEASURE:TRACE1:AREA1:METHOD AUTO

**:MEASure:TRANge<x> (Time Range)**

機能 測定範囲を設定 / 問い合わせします。

構文 :MEASure:TRANge<x> {<NRf>, <NRf>}  
:MEASure:TRANge<x>?  
<x> = 1, 2  
<NRf> = -5 ~ 5(div)

例 :MEASURE:TRANGE1 -5,0  
:MEASURE:TRANGE1? -> :MEASURE:  
TRANGE1 0.000E+00, -5.000E+00

**:MEASure:WAIT?**

機能 タイムアウト付きで自動測定実行の終了を待ちます。

構文 MEASure:WAIT? {<NRf>}  
<NRf> = 1 ~ 360000(タイムアウト時間、10ms)

例 MEASURE:WAIT 100? -> :MEASURE:WAIT 1

解説 タイムアウト時間内に自動測定の実行が終了すると「0」、終了してないか自動測定が行われていない場合は「1」を返します。  
タイムアウト時間を長く設定しても自動測定実行が終了した時点で「0」を返します。

**:MEASure:WINDow<x>**

機能 各エリアの測定対象ウインドウを設定 / 問い合わせします。

構文 :MEASure:WINDow<x> {MAIN|Z1|Z2}  
:MEASure:WINDow<x>?  
<x> = 1, 2

例 :MEASURE:WINDOW1 MAIN  
:MEASURE:WINDOW1? -> :MEASURE:  
WINDOW1 MAIN



## 5.20 REference グループ

### **:REference<x>?**

機能 各リファレンスに関するすべての設定値を問い合わせます。

構文 :REference<x>?  
<x> = 1 ~ 4

例 :REFERENCE1? -> :REFERENCE1:  
SELECT REFERENCE; DISPLAY 1; INVERT 0;  
LABEL: DEFINE "REF1"; MODE 1;  
REFERENCE1: POSITION 1.000E+00;  
SVALUE 1

### **:REference<x>:DISPlay**

機能 各リファレンスの表示の ON/OFF を設定 / 問い合わせします。

構文 :REference<x>:DISPlay {<Boolean>}  
:REference<x>:DISPlay?  
<x> = 1 ~ 4

例 :REFERENCE1: DISPLAY ON  
:REFERENCE1: DISPLAY? -> :REFERENCE1:  
DISPLAY 1

### **:REference<x>:INVert**

機能 各リファレンスの反転表示を設定 / 問い合わせします。

構文 :REference<x>:INVert {<Boolean>}  
:REference<x>:INVert?  
<x> = 1 ~ 4

例 :REFERENCE1: INVERT ON  
:REFERENCE1: INVERT? -> :REFERENCE1:  
INVERT 1

### **:REference<x>:LAbel?**

機能 各リファレンスの波形ラベルに関するすべての設定値を問い合わせます。

構文 :REference<x>:LAbel?  
<x> = 1 ~ 4

例 :REFERENCE1: LABEL? -> :REFERENCE1:  
LABEL: DEFINE "REF1"; MODE 1

### **:REference<x>:LAbel[:DEFine]**

機能 各リファレンスの波形ラベル名を設定 / 問い合わせします。

構文 :REference<x>:LAbel[:DEFine] {<文字列>}  
:REference<x>:LAbel[:DEFine]?  
<x> = 1 ~ 4  
<文字列> = 8 文字以内

例 :REFERENCE1: LABEL: DEFINE "REF1"  
:REFERENCE1: LABEL: DEFINE?  
-> :REFERENCE1: LABEL: DEFINE "REF1"

### **:REference<x>:LAbel:MODE**

機能 各リファレンスの波形ラベル名表示の ON/OFF を設定 / 問い合わせします。

構文 :REference<x>:LAbel:MODE {<Boolean>}  
:REference<x>:LAbel:MODE?  
<x> = 1 ~ 4

例 :REFERENCE1: LABEL: MODE ON  
:REFERENCE1: LABEL: MODE?  
-> :REFERENCE1: LABEL: MODE 1

### **:REference<x>:LOAD**

機能 各リファレンスに波形をロードします。

構文 :REference<x>:LOAD {<Nrf>}  
<x> = 1 ~ 4  
<Nrf> = 1 ~ 12 (1 ~ 8 は各トレース、9 ~ 12 は各内部メモリ)

例 :REFERENCE1: LOAD 1

### **:REference<x>:POSition**

機能 各リファレンスの垂直ポジションを設定 / 問い合わせします。

構文 :REference<x>:POSition {<Nrf>}  
:REference<x>:POSition?  
<x> = 1 ~ 4  
<Nrf> = -4 ~ 4 (div)

例 :REFERENCE1: POSITION 1  
:REFERENCE1: POSITION? -> :REFERENCE1:  
POSITION 1.000E+00

### **:REference<x>:SElect**

機能 各演算チャンネルに割り当てる波形 (演算 / リファレンス) を設定 / 問い合わせします。

構文 :REference<x>:SElect {MATH|REference}  
:REference<x>:SElect?  
<x> = 1 ~ 4

例 :REFERENCE1: SELECT MATH  
:REFERENCE1: SELECT? -> :REFERENCE1:  
SELECT MATH

### **:REference<x>:SVALue (Scale VALUE)**

機能 各リファレンスのスケール値表示の ON/OFF を設定 / 問い合わせします。

構文 :REference<x>:SVALue {<Boolean>}  
:REference<x>:SVALue?  
<x> = 1 ~ 4

例 :REFERENCE1: SVALUE ON  
:REFERENCE1: SVALUE? -> :REFERENCE1:  
SVALUE 1

## 5.21 SEARch グループ

### : SEARch<x>?

機能 サーチ機能に関するすべての設定値を問い合わせます。

構文 :SEARch<x>?

<x> = 1、2

例 :SEARCH1? -> :SEARCH1:CLOCK:SOURCE 1;  
POLARITY FALL;;SEARCH1:DECIMATION 1;  
HOLDOFF 0.000E+00;LOGIC AND;  
POLARITY RISE;SLOGIC:POLARITY RISE;  
SOURCE A0;;SEARCH1:SMODE OFF;  
SPATTERN:CLOCK:MODE 1;POLARITY FALL;  
SOURCE 1;;SEARCH1:SPATTERN:CS 1;  
LATCH:TRACE 1;POLARITY FALL;;SEARCH1:  
SPATTERN:SETUP:BITRATE 1.000E+00;  
PATTERN "1100110111101111";:SEARCH1:  
SPOINT -5.000E+00;STRACE 1;TRACE1:  
CONDITION DONTCARE;  
HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;;SEARCH1:TRACE2:  
CONDITION DONTCARE;  
HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;;SEARCH1:TRACE3:  
CONDITION DONTCARE;  
HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;;SEARCH1:TRACE4:  
CONDITION DONTCARE;  
HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;;SEARCH1:TRACE5:  
CONDITION DONTCARE;  
HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;;SEARCH1:TRACE6:  
CONDITION DONTCARE;  
HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;;SEARCH1:TRACE7:  
CONDITION DONTCARE;  
HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;;SEARCH1:TRACE8:  
CONDITION DONTCARE;  
HYSTERESIS 1.000E+00;  
LEVEL 0.000E+00;;SEARCH1:TYPE EDGE;  
WIDTH:MODE OUT;TIME1 1.000E+00;  
TIME2 2.000E+00;TYPE PULSE

### : SEARch<x>:ABORt

機能 サーチを中止します。

構文 :SEARch<x>:ABORt

<x> = 1、2

例 :SEARCH1:ABORt

### : SEARch<x>:CLOCK?

機能 クロックチャンネルに関するすべての設定値を問い合わせます。

構文 :SEARch<x>:CLOCK?

<x> = 1、2

例 :SEARCH1:CLOCK? -> :SEARCH1:CLOCK:  
SOURCE 1;POLARITY RISE

### : SEARch<x>:CLOCK:POLarity

機能 クロックチャンネルの極性を設定 / 問い合わせします。

構文 :SEARch<x>:CLOCK:POLarity {FALL|RISE}

:SEARch<x>:CLOCK:POLarity?

<x> = 1、2

例 :SEARCH1:CLOCK:POLARITY RISE  
:SEARCH1:CLOCK:POLARITY? -> :SEARCH1:  
CLOCK:POLARITY RISE

解説

- ・「:SEARch<x>:CLOCK:SOURce NONE」のときは無効です。
- ・「:SEARch<x>:TYPE STATE」のときに有効です。
- ・「:SEARch<x>:TYPE WIDTH」且つ「:SEARch<x>:WIDTH:TYPE PSTATE」のときに有効です。

### : SEARch<x>:CLOCK:SOURce

機能 サーチのクロックトレースを設定 / 問い合わせします。

構文 :SEARch<x>:CLOCK:SOURce {<Nrf>|NONE}

:SEARch<x>:CLOCK:SOURce?

<x> = 1、2

<Nrf> = 1 ~ 8

例 :SEARCH1:CLOCK:SOURCE NONE  
:SEARCH1:CLOCK:SOURCE? -> :SEARCH1:  
CLOCK:SOURCE NONE

解説

- ・「:SEARch<x>:TYPE STATE」のときに有効です。
- ・「:SEARch<x>:TYPE WIDTH」且つ「:SEARch<x>:WIDTH:TYPE PSTATE」のときに有効です。

### : SEARch<x>:DECimation

機能 スキップモードの間引き検出を設定 / 問い合わせします。

構文 :SEARch<x>:DECimation {<Nrf>}

:SEARch<x>:DECimation?

<x> = 1、2

<Nrf> = 1 ~ 9999

例 :SEARCH1:DECIMATION 1  
:SEARCH1:DECIMATION? -> :SEARCH1:  
DECIMATION 1

### : SEARch<x>:EXECute

機能 サーチを実行します。オーバーラップコマンドです。

構文 :SEARch<x>:EXECute

<x> = 1、2

例 :SEARCH1:EXECUTE

## 5.21 SEARCh グループ

### :SEARCh<x>:HOLDoff

機能 ホールドオフ検出を設定 / 問い合わせします。

構文 :SEARCh<x>:HOLDoff {<時間>}

:SEARCh<x>:HOLDoff?

<x> = 1、2

<時間> = 0s ~ 1s(100ps ステップ)

例 :SEARCH1:HOLDOFF 0S

:SEARCH1:HOLDOFF? -> :SEARCH1:

HOLDOFF 0.000E+00

### :SEARCh<x>:LOGic

機能 サーチロジックを設定 / 問い合わせします。

構文 :SEARCh<x>:LOGic {AND|OR}

:SEARCh<x>:LOGic?

<x> = 1、2

例 :SEARCH1:LOGIC OR

:SEARCH1:LOGIC? -> :SEARCH1:LOGIC OR

解説 ・「:SEARCh<x>:TYPE EQUalify|SPATtern|  
STATE」のときに有効です。

・「:SEARCh<x>:TYPE WIDTH」且つ  
「:SEARCh<x>:WIDTH:TYPE PQUalify|  
PSTATE」のときに有効です。

### :SEARCh<x>:MARK

機能 サーチマークの ON/OFF を設定 / 問い合わせします。

構文 :SEARCh<x>:MARK {<Boolean>}

:SEARCh<x>:MARK?

<x> = 1、2

例 :SEARCH1:MARK ON

:SEARCH1:MARK? -> :SEARCH1:MARK 1

### :SEARCh<x>:POLarity

機能 サーチ極性を設定 / 問い合わせします。

構文 :SEARCh<x>:POLarity {ENTER|EXIT|FALL|

FALSe|NEGative|POSitive|RISE|TRUE}

:SEARCh<x>:POLarity?

<x> = 1、2

例 :SEARCH1:POLARITY ENTER

:SEARCH1:POLARITY? ->

:SEARCH1:POLARITY ENTER

解説 ・「:SEARCh<x>:TYPE EDGE|EQUalify」のとき  
は {FALL|RISE} が有効です。

・「:SEARCh<x>:TYPE STATE」のときは  
{ENTER|EXIT} が有効です。

・「:SEARCh<x>:TYPE WIDTH」且つ  
「:SEARCh<x>:WIDTH:TYPE PQUalify|  
PULSe」のときは {NEGative|POSitive} が  
有効です。

・「:SEARCh<x>:TYPE WIDTH」且つ  
「:SEARCh<x>:WIDTH:TYPE PSTATE」のとき  
は {FALSe|TRUE} が有効です。

### :SEARCh<x>:SElect

機能 サーチ機能での検出波形番号を設定し、検出波形  
番号に相当する位置を問い合わせます。

構文 :SEARCh<x>:SElect {<NRF>|MAXimum}

:SEARCh<x>:SElect?

<x> = 1、2

<NRF> = 0 ~ 4999

例 :SEARCH1:SELECT 1

:SEARCH1:SELECT? -> :SEARCH1:

SELECT 1.500E+00

解説 検索された位置が無い場合は、"NAN" がクエリと  
して返ってきます。

### :SEARCh<x>:SElect? MAXimum

機能 サーチ機能での検出回数を問い合わせます。

構文 :SEARCh<x>:SElect? {MAXimum}

<x> = 1、2

例 :SEARCH1:SECT? MAXIMUM -> :SEARCH1:

SELECT 100

解説 検索された位置が無い場合は、"NAN" がクエリと  
して返ってきます。

**:SEARCh<x>:SLOGic?**

機能 ロジックサーチに関するすべての設定値を問い合わせます。

構文 :SEARCh<x>:SLOGic?

<x> = 1、2

例 :SEARCH1:SLOGIC? -> :SEARCH1:SLOGIC:  
CLOCK:POLARITY FALL;SOURCE A0;;  
SEARCH1:SLOGIC:I2CBUS:CLOCK:  
SOURCE A0;;SEARCH1:SLOGIC:  
POLARITY RISE;SOURCE A0;SPATTERN:  
CLOCK:MODE 1;POLARITY FALL;  
SOURCE A0;;SEARCH1:SLOGIC:SPATTERN:  
CS 1;LATCH:TRACE DONTCARE;  
POLARITY RISE;;SEARCH1:SLOGIC:  
SPATTERNSETUP:BITRATE 1.0000000E+06;  
DATA:ACTIVE HIGH;TRACE A0;;SEARCH1:  
SLOGIC:SPATTERN:SETUP:  
PATTERN "XXXXXXXX";:SEARCH1:SLOGIC:  
STATE:BIT:A0 DONTCARE;A1 DONTCARE;  
A2 DONTCARE;A3 DONTCARE;A4 DONTCARE;  
A5 DONTCARE;A6 DONTCARE;A7 DONTCARE;  
B0 DONTCARE;B1 DONTCARE;B2 DONTCARE;  
B3 DONTCARE;B4 DONTCARE;B5 DONTCARE;  
B6 DONTCARE;B7 DONTCARE;C0 DONTCARE;  
C1 DONTCARE;C2 DONTCARE;C3 DONTCARE;  
C4 DONTCARE;C5 DONTCARE;C6 DONTCARE;  
C7 DONTCARE;D0 DONTCARE;D1 DONTCARE;  
D2 DONTCARE;D3 DONTCARE;D4 DONTCARE;  
D5 DONTCARE;D6 DONTCARE;D7 DONTCARE;  
LOGIC AND;;SEARCH1:SLOGIC:STATE:  
GROUP1:CONDITION DONTCARE;DATA1 0;  
DATA2 255;PATTERN "XXXXXXXX";:  
SEARCH1:SLOGIC:STATE:GROUP2:  
CONDITION TRUE;DATA1 0;DATA2 255;  
PATTERN "XXXXXXXX";:SEARCH1:SLOGIC:  
STATE:GROUP3:CONDITION TRUE;DATA1 0;  
DATA2 255;PATTERN "XXXXXXXX";:  
SEARCH1:SLOGIC:STATE:GROUP4:  
CONDITION TRUE;DATA1 0;DATA2 255;  
PATTERN "XXXXXXXX";:SEARCH1:SLOGIC:  
STATE:GROUP5:CONDITION DONTCARE;  
DATA1 0;DATA2 0;PATTERN ""::SEARCH1:  
SLOGIC:STATE:TYPE BIT;;SEARCH1:  
SLOGIC:WIDTH:MODE OUT;  
TIME1 1.0000000E-09;  
TIME2 1.0000000E-09;TYPE PULSE

**:SEARCh<x>:SLOGic:CLOCK?**

機能 ロジックサーチのクロックに関するすべての設定値を問い合わせます。

構文 :SEARCh<x>:SLOGic:CLOCK?

<x> = 1、2

例 :SEARCH1:SLOGIC:CLOCK? -> :SEARCH1:  
SLOGIC:CLOCK:POLARITY FALL;SOURCE A0

**:SEARCh<x>:SLOGic:CLOCK:POLarity**

機能 ロジックサーチのクロックの極性を設定 / 問い合わせします。

構文 :SEARCh<x>:SLOGic:CLOCK:

POLarity {FALL|RISE}

:SEARCh<x>:SLOGic:CLOCK:POLarity?

<x> = 1、2

例 :SEARCH1:SLOGIC:CLOCK:POLARITY FALL

:SEARCH1:SLOGIC:CLOCK:POLARITY?

-> :SEARCH1:SLOGIC:CLOCK:

POLARITY FALL

**:SEARCh<x>:SLOGic:CLOCK:SOURCE**

機能 ロジックサーチのクロックのソースを設定 / 問い合わせします。

構文 :SEARCh<x>:SLOGic:CLOCK:SOURCE {A<y>|

B<y>|C<y>|D<y>|DONTcare}

:SEARCh<x>:SLOGic:CLOCK:SOURCE?

<x> = 1、2

<y> = 0 ~ 7

例 :SEARCH1:SLOGIC:CLOCK:SOURCE A0

:SEARCH1:SLOGIC:CLOCK:SOURCE?

-> :SEARCH1:SLOGIC:CLOCK:SOURCE A0

解説 DL9510L、DL9505L は {A<y>|C<y>|DONTcare} が有効です。

**:SEARCh<x>:SLOGic:POLarity**

機能 ロジックサーチの極性を設定 / 問い合わせします。

構文 :SEARCh<x>:SLOGic:

POLarity {ENTER|EXIT|FALL|FALSE|

NEGative|POSitive|RISE|TRUE}

:SEARCh<x>:SLOGic:POLarity?

<x> = 1、2

例 :SEARCH1:SLOGIC:POLARITY FALL

:SEARCH1:SLOGIC:POLARITY?

-> :SEARCH1:SLOGIC:POLARITY FALL

解説

- ・「:SEARCh<x>:TYPE LEDGe|LQQualify」のときは {FALL|RISE} が有効です。
- ・「:SEARCh<x>:TYPE LState」のときは {ENTER|EXIT} が有効です。
- ・「:SEARCh<x>:TYPE LWidth」且つ「:SEARCh<x>:SLOGic:WIDTH:TYPE PQQualify|PULSe」のときは {NEGative|POSitive} が有効です。
- ・「:SEARCh<x>:TYPE LWidth」且つ「:SEARCh<x>:SLOGic:WIDTH:TYPE PState」のときは {FALSE|TRUE} が有効です。

## 5.21 SEARCH グループ

### :SEARCH<x>:SLOGic:SOURCE

機能 ロジックサーチのソースを設定 / 問い合わせします。

構文 :SEARCH<x>:SLOGic:SOURCE {A<y>|B<y>|C<y>|D<y>}  
:SEARCH<x>:SLOGic:SOURCE?  
<x> = 1、2  
<y> = 0 ~ 7

例 :SEARCH1:SLOGIC:SOURCE A0  
:SEARCH1:SLOGIC:SOURCE? -> :SEARCH1:  
SLOGIC:SOURCE A0

解説 DL9510L、DL9505L は {A<y>|C<y>} が有効です。

### :SEARCH<x>:SLOGic:SPATtern? (Serial Pattern)

機能 ロジックシリアルパターンサーチに関するすべての設定値を問い合わせます。

構文 :SEARCH<x>:SLOGic:SPATtern?  
<x> = 1、2

例 :SEARCH1:SLOGIC:SPATTERN?  
-> :SEARCH1:SLOGIC:SPATTERN:CLOCK:  
MODE 1;POLARITY FALL;SOURCE A0;;  
SEARCH1:SLOGIC:SPATTERN:CS 1;LATCH:  
TRACE A0;POLARITY FALL;;SEARCH1:  
SLOGIC:SPATTERNSETUP:  
BITRATE 1.0000000E+00;DATA:  
ACTIVE HIGH;TRACE A0;;SEARCH1:  
SLOGIC:SPATTERN:SETUP:  
PATTERN "1100110111101111"

### :SEARCH<x>:SLOGic:SPATtern:CLOCK?

機能 ロジックシリアルパターンサーチのクロックに関するすべての設定値を問い合わせます。

構文 :SEARCH<x>:SLOGic:SPATtern:CLOCK?  
<x> = 1、2

例 :SEARCH1:SLOGIC:SPATTERN:CLOCK?  
-> :SEARCH1:SLOGIC:SPATTERN:CLOCK:  
MODE 1;POLARITY FALL;SOURCE A0

### :SEARCH<x>:SLOGic:SPATtern:CLOCK:MODE

機能 ロジックシリアルパターンサーチのクロックの有効 / 無効を設定 / 問い合わせします。

構文 :SEARCH<x>:SLOGic:SPATtern:CLOCK:  
MODE {<Boolean>}  
:SEARCH<x>:SLOGic:SPATtern:CLOCK:  
MODE?  
<x> = 1、2

例 :SEARCH1:SLOGIC:SPATTERN:CLOCK:  
MODE ON  
:SEARCH1:SLOGIC:SPATTERN:CLOCK:MODE?  
-> :SEARCH1:SLOGIC:SPATTERN:CLOCK:  
MODE 1

### :SEARCH<x>:SLOGic:SPATtern:CLOCK:POLarity

機能 ロジックシリアルパターンサーチのクロックトレースの極性を設定 / 問い合わせします。

構文 :SEARCH<x>:SLOGic:SPATtern:CLOCK:  
POLarity {FALL|RISE}  
:SEARCH<x>:SLOGic:SPATtern:CLOCK:  
POLarity?  
<x> = 1、2

例 :SEARCH1:SLOGIC:SPATTERN:CLOCK:  
POLARITY FALL  
:SEARCH1:SLOGIC:SPATTERN:CLOCK:  
POLARITY? -> :SEARCH1:SLOGIC:  
SPATTERN:CLOCK:POLARITY FALL

解説 「:SEARCH<x>:SLOGic:SPATtern:CLOCK:  
MODE ON」のときに有効です。

### :SEARCH<x>:SLOGic:SPATtern:CLOCK:SOURCE

機能 ロジックシリアルパターンサーチのクロックトレースを設定 / 問い合わせします。

構文 :SEARCH<x>:SLOGic:SPATtern:CLOCK:  
SOURCE {A<y>|B<y>|C<y>|D<y>}  
:SEARCH<x>:SLOGic:SPATtern:CLOCK:  
SOURCE?  
<x> = 1、2  
<y> = 0 ~ 7

例 :SEARCH1:SLOGIC:SPATTERN:CLOCK:  
SOURCE A0  
:SEARCH1:SLOGIC:SPATTERN:CLOCK:  
SOURCE? -> :SEARCH1:SLOGIC:SPATTERN:  
CLOCK:  
SOURCE A0

解説 ・「:SEARCH<x>:SLOGic:SPATtern:CLOCK:  
MODE ON」のときに有効です。  
・DL9510L、DL9505L は {A<y>|C<y>} が有効です。

### :SEARCH<x>:SLOGic:SPATtern:CS

機能 ロジックシリアルパターンサーチのチップセレクトの有効 / 無効を設定 / 問い合わせします。

構文 :SEARCH<x>:SLOGic:SPATtern:  
CS {<Boolean>}  
:SEARCH<x>:SLOGic:SPATtern:CS?  
<x> = 1、2

例 :SEARCH1:SLOGIC:SPATTERN:CS ON  
:SEARCH1:SLOGIC:SPATTERN:CS?  
-> :SEARCH1:SLOGIC:SPATTERN:CS 1

解説 「:SEARCH<x>:SLOGic:SPATtern:CLOCK:  
MODE ON」のときに有効です。

**:SEARCh<x>:SLOGic:SPATtern:LATCh?**

機能 ロジックシリアルパターンサーチのラッチに関するすべての設定値を問い合わせます。

構文 :SEARCh<x>:SLOGic:SPATtern:LATCh?  
<x> = 1、2

例 :SEARCH1:SLOGIC:SPATTERN:LATCH?  
-> :SEARCH1:SLOGIC:SPATTERN:LATCH:  
TRACE A0;POLARITY FALL

**:SEARCh<x>:SLOGic:SPATtern:LATCh:POLarity**

機能 ロジックシリアルパターンサーチのラッチトレースの極性を設定 / 問い合わせします。

構文 :SEARCh<x>:SLOGic:SPATtern:LATCh:  
POLarity {FALL|RISE}  
:SEARCh<x>:SLOGic:SPATtern:LATCh:  
POLarity?  
<x> = 1、2

例 :SEARCH1:SLOGIC:SPATTERN:LATCH:  
POLARITY FALL  
:SEARCH1:SLOGIC:SPATTERN:LATCH:  
POLARITY? -> :SEARCH1:SLOGIC:  
SPATTERN:LATCH:POLARITY FALL

解説 ・「:SEARCh<x>:SLOGic:SPATtern:CLOCK:  
MODE ON」のときに有効です。  
・「:SEARCh<x>:SLOGic:SPATtern:LATCh:  
TRACE DONTcare」のときは無効です。

**:SEARCh<x>:SLOGic:SPATtern:LATCh:TRACe**

機能 ロジックシリアルパターンサーチのラッチトレースを設定 / 問い合わせします。

構文 :SEARCh<x>:SLOGic:SPATtern:LATCh:  
TRACe {A<y>|B<y>|C<y>|D<y>|DONTcare}  
:SEARCh<x>:SLOGic:SPATtern:LATCh:  
TRACe?  
<x> = 1、2  
<y> = 0 ~ 7

例 :SEARCH1:SLOGIC:SPATTERN:LATCH:TRACE  
A0  
:SEARCH1:SLOGIC:SPATTERN:LATCH:TRACE?  
-> :SEARCH1:SLOGIC:SPATTERN:LATCH:  
TRACE A0

解説 ・「:SEARCh<x>:SLOGic:SPATtern:CLOCK:  
MODE ON」のときに有効です。  
・DL9510L、DL9505Lは {A<y>|C<y>|  
DONTcare} が有効です。

**:SEARCh<x>:SLOGic:SPATtern[:SETup]?**

機能 ロジックシリアルパターンサーチのセットアップに関するすべての設定値を問い合わせます。

構文 :SEARCh<x>:SLOGic:SPATtern[:SETup]?  
<x> = 1、2

例 :SEARCH1:SLOGIC:SPATTERN:SETUP?  
-> :SEARCH1:SLOGIC:SPATTERN:SETUP:  
BITRATE 1.0000000E+00;DATA:  
ACTIVE HIGH;TRACE A0;;SEARCH1:SLOGIC:  
SPATTERN:SETUP:  
PATTERN "1100110111101111"

**:SEARCh<x>:SLOGic:SPATtern[:SETup]:BITRate**

機能 ロジックシリアルパターンサーチのビットレートを設定 / 問い合わせします。

構文 :SEARCh<x>:SLOGic:SPATtern[:SETup]:  
BITRate {<NRf>}  
:SEARCh<x>:SLOGic:SPATtern[:SETup]:  
BITRate?  
<x> = 1、2  
<NRf> = 1 ~ 1G(bps)

例 :SEARCH1:SLOGIC:SPATTERN:SETUP:  
BITRATE 1  
:SEARCH1:SLOGIC:SPATTERN:SETUP:  
BITRATE? -> :SEARCH1:SLOGIC:SPATTERN:  
SETUP:BITRATE 1.000E+00

解説 「:SEARCh<x>:SLOGic:SPATtern:CLOCK:  
MODE OFF」のときに有効です。

**:SEARCh<x>:SLOGic:SPATtern[:SETup]:CLEAr**

機能 ロジックシリアルパターンサーチのパターンをすべてクリア (Don't care) します。

構文 :SEARCh<x>:SLOGic:SPATtern[:SETup]:  
CLEAr  
<x> = 1、2

例 :SEARCH1:SLOGIC:SPATTERN:SETUP:CLEAR

**:SEARCh<x>:SLOGic:SPATtern[:SETup]:DATA?**

機能 ロジックシリアルパターンサーチのデータに関するすべての設定値を問い合わせます。

構文 :SEARCh<x>:SLOGic:SPATtern[:SETup]:  
DATA?  
<x> = 1、2

例 :SEARCH1:SLOGIC:SPATTERN:SETUP:DATA?  
-> :SEARCH1:SLOGIC:SPATTERN:SETUP:  
DATA:ACTIVE HIGH;TRACE A0

## 5.21 SEARCh グループ

### : SEARCh<x>: SLOGic: SPATtern [: SETUp] :

#### DATA: ACTIve

機能 ロジックシリアルパターンサーチのデータのトレースのアクティブレベルを設定 / 問い合わせします。

構文 : SEARCh<x>: SLOGic: SPATtern[: SETUp] :  
DATA: ACTIve {HIGH|LOW}  
: SEARCh<x>: SLOGic: SPATtern[: SETUp] :  
DATA: ACTIve?  
<x> = 1、2

例 : SEARCH1: SLOGIC: SPATTERN: SETUP: DATA:  
ACTIVE HIGH  
: SEARCH1: SLOGIC: SPATTERN: SETUP: DATA:  
ACTIVE? -> : SEARCH1: SLOGIC: SPATTERN:  
SETUP: DATA: ACTIVE HIGH

### : SEARCh<x>: SLOGic: SPATtern [: SETUp] :

#### DATA: TRACe

機能 ロジックシリアルパターンサーチのデータのトレースを設定 / 問い合わせします。

構文 : SEARCh<x>: SLOGic: SPATtern[: SETUp] :  
DATA: TRACe {A<y>|B<y>|C<y>|D<y>}  
: SEARCh<x>: SLOGic: SPATtern[: SETUp] :  
DATA: TRACe?  
<x> = 1、2  
<y> = 0 ~ 7

例 : SEARCH1: SLOGIC: SPATTERN: SETUP: DATA:  
TRACE A0  
: SEARCH1: SLOGIC: SPATTERN: SETUP: DATA:  
TRACE? -> : SEARCH1: SLOGIC: SPATTERN:  
SETUP: DATA: TRACE A0

解説 DL9510L、DL9505L は {A<y>|C<y>} が有効です。

### : SEARCh<x>: SLOGic: SPATtern [: SETUp] :

#### HEXA

機能 ロジックシリアルパターンサーチのパターンをHEXAで設定します。

構文 : SEARCh<x>: SLOGic: SPATtern[: SETUp] :  
HEXA {<文字列>}  
<x> = 1、2  
<文字列> = '0' ~ 'F'、'X'の組み合わせ 32文字以内

例 : SEARCH1: SLOGIC: SPATTERN: SETUP:  
HEXA "ABCD"

### : SEARCh<x>: SLOGic: SPATtern [: SETUp] :

#### PATtern

機能 ロジックシリアルパターンサーチのパターンをBINARYで設定 / 問い合わせします。

構文 : SEARCh<x>: SLOGic: SPATtern[: SETUp] :  
PATtern {<文字列>}  
: SEARCh<x>: SLOGic: SPATtern[: SETUp] :  
PATtern?  
<x> = 1、2  
<文字列> = '0'、'1'、'X'の組み合わせ 128文字以内

例 : SEARCH1: SLOGIC: SPATTERN: SETUP:  
PATTERN "1100110111101111"  
: SEARCH1: SLOGIC: SPATTERN: SETUP:  
PATTERN? -> : SEARCH1: SLOGIC: SPATTERN:  
SETUP: PATTERN "1100110111101111"

### : SEARCh<x>: SLOGic: STATe?

機能 ロジックステートサーチに関するすべての設定値を問い合わせます。

構文 : SEARCh<x>: SLOGic: STATe?  
<x> = 1、2

例 : SEARCH1: SLOGIC: STATE? -> : SEARCH1:  
SLOGIC: STATE: BIT: A0 LOW; A1 LOW;  
A2 LOW; A3 LOW; A4 HIGH; A5 HIGH;  
A6 HIGH; A7 HIGH; B0 LOW; B1 LOW; B2 LOW;  
B3 LOW; B4 HIGH; B5 HIGH; B6 HIGH;  
B7 HIGH; C0 LOW; C1 LOW; C2 LOW; C3 LOW;  
C4 HIGH; C5 HIGH; C6 HIGH; C7 HIGH;  
D0 LOW; D1 LOW; D2 LOW; D3 LOW; D4 HIGH;  
D5 HIGH; D6 HIGH; D7 HIGH; LOGIC AND; :  
SEARCH1: SLOGIC: STATE: GROUP1:  
CONDITION DONTCARE;  
DATA1 0.0000000E+00;  
DATA2 255.000000E+00;  
PATTERN "1111000011110000111100001111  
0000"; : SEARCH1: SLOGIC: STATE: GROUP2:  
CONDITION DONTCARE;  
DATA1 0.0000000E+00;  
DATA2 1.0000000E+00; PATTERN "": :  
SEARCH1: SLOGIC: STATE: GROUP3:  
CONDITION DONTCARE;  
DATA1 0.0000000E+00;  
DATA2 1.0000000E+00; PATTERN "": :  
SEARCH1: SLOGIC: STATE: GROUP4:  
CONDITION DONTCARE;  
DATA1 0.0000000E+00;  
DATA2 1.0000000E+00; PATTERN "": :  
SEARCH1: SLOGIC: STATE: GROUP5:  
CONDITION DONTCARE;  
DATA1 0.0000000E+00;  
DATA2 0.0000000E+00; PATTERN "": :  
SEARCH1: SLOGIC: STATE: TYPE BIT

**:SEARCh<x>:SLOGic:STAtE:BIT?**

機能 ロジックステートサーチのビットに関するすべての設定値を問い合わせます。

構文 :SEARCh<x>:SLOGic:STAtE:BIT?  
<x> = 1、2

例 :SEARCH1:SLOGIC:STATE:BIT?  
-> :SEARCH1:SLOGIC:STATE:BIT:  
A0 DONTCARE;A1 DONTCARE;A2 DONTCARE;  
A3 DONTCARE;A4 DONTCARE;A5 DONTCARE;  
A6 DONTCARE;A7 DONTCARE;B0 DONTCARE;  
B1 DONTCARE;B2 DONTCARE;B3 DONTCARE;  
B4 DONTCARE;B5 DONTCARE;B6 DONTCARE;  
B7 DONTCARE;C0 DONTCARE;C1 DONTCARE;  
C2 DONTCARE;C3 DONTCARE;C4 DONTCARE;  
C5 DONTCARE;C6 DONTCARE;C7 DONTCARE;  
D0 DONTCARE;D1 DONTCARE;D2 DONTCARE;  
D3 DONTCARE;D4 DONTCARE;D5 DONTCARE;  
D6 DONTCARE;D7 DONTCARE;LOGIC AND

**:SEARCh<x>:SLOGic:STAtE:BIT:{A<x>|B<x>|C<x>|D<x>}**

機能 ロジックステートサーチの各ビットの成立条件を設定 / 問い合わせします。

構文 :SEARCh<x>:SLOGic:STAtE:BIT:{A<x>|B<x>|C<x>|D<x>} {DONTcare|HIGH|LOW}  
:SEARCh<x>:SLOGic:STAtE:BIT:{A<x>|B<x>|C<x>|D<x>}?

<x> = 1、2  
<y> = 0 ~ 7  
例 :SEARCH1:SLOGIC:STATE:BIT:A0 DONTCARE  
:SEARCH1:SLOGIC:STATE:BIT:A0?  
-> :SEARCH1:SLOGIC:STATE:BIT:  
A0 DONTCARE

解説 DL9510L、DL9505L は {A<x>|C<x>} が有効です。

**:SEARCh<x>:SLOGic:STAtE:BIT:CLEAr**

機能 ロジックステートサーチの各ビットの成立条件をすべてクリア (Don't care) します。

構文 :SEARCh<x>:SLOGic:STAtE:BIT:CLEAr  
<x> = 1、2

例 :SEARCH1:SLOGIC:STATE:BIT:CLEAr

**:SEARCh<x>:SLOGic:STAtE:BIT:LOGic**

機能 ロジックステートサーチのロジックを設定 / 問い合わせします。

構文 :SEARCh<x>:SLOGic:STAtE:BIT:  
LOGic {AND|OR}  
:SEARCh<x>:SLOGic:STAtE:BIT:LOGic?  
<x> = 1、2

例 :SEARCH1:SLOGIC:STATE:BIT:LOGIC AND  
:SEARCH1:SLOGIC:STATE:BIT:LOGIC?  
-> :SEARCH1:SLOGIC:STATE:BIT:  
LOGIC AND

**:SEARCh<x>:SLOGic:STAtE:GRouP<x>?**

機能 ロジックステートサーチの各グループに関するすべての設定値を問い合わせます。

構文 :SEARCh<x>:SLOGic:STAtE:GRouP<x>?  
SEARCh<x> の <x> = 1、2  
GRouP<x> の <x> = 1 ~ 5

例 :SEARCH1:SLOGIC:STATE:GROUP1?  
-> :SEARCH1:SLOGIC:STATE:GROUP1:  
CONDITION DONTCARE;  
DATA1 0.0000000E+00;  
DATA2 255.00000E+00;PATTERN "11110000  
111100001111000011110000"

**:SEARCh<x>:SLOGic:STAtE:GRouP<x>:CLEAr**

機能 ロジックステートサーチの各グループの成立条件をすべてクリア (Don't care) します。

構文 :SEARCh<x>:SLOGic:STAtE:GRouP<x>:  
CLEAr  
SEARCh<x> の <x> = 1、2  
GRouP<x> の <x> = 1 ~ 5

例 :SEARCH1:SLOGIC:STATE:GROUP1:CLEAr

**:SEARCh<x>:SLOGic:STAtE:GRouP<x>:CONDition**

機能 ロジックステートサーチの各グループの判定条件を設定 / 問い合わせします。

構文 :SEARCh<x>:SLOGic:STAtE:GRouP<x>:  
CONDition {BETween|DONTcare|FALSe|GTHan|LTHan|ORANge|TRUE}  
:SEARCh<x>:SLOGic:STAtE:GRouP<x>:  
CONDition?  
SEARCh<x> の <x> = 1、2  
GRouP<x> の <x> = 1 ~ 5

例 :SEARCH1:SLOGIC:STATE:GROUP1:  
CONDITION BETWEEN  
:SEARCH1:SLOGIC:STATE:GROUP1:  
CONDITION? -> :SEARCH1:SLOGIC:STATE:  
GROUP1:CONDITION BETWEEN



## 5.21 SEARCH グループ

### :SEARCH<x>:SLOGic:STAtE:GRouP<x>:DATA<x>

機能 ロジックステートサーチの各グループのデータの比較データを設定 / 問い合わせします。

構文 :SEARCH<x>:SLOGic:STAtE:GRouP<x>:

DATA<x> {<NRf>}

:SEARCH<x>:SLOGic:STAtE:GRouP<x>:

DATA<x>?

SEARCH<x> の <x> = 1, 2

GRouP<x> の <x> = 1 ~ 5

DATA<x> の <x> = 1, 2

<NRf> = 本体ユーザーズマニュアル参照。

例 :SEARCH1:SLOGic:STAtE:GRouP1:DATA1 1

:SEARCH1:SLOGic:STAtE:GRouP1:DATA1?

-> :SEARCH1:SLOGic:STAtE:GRouP1:

DATA1 1.0000000E+00

解説

- 「:SEARCH<x>:SLOGic:STAtE:GRouP<x>:CONDition GTHan」のときは

「:SEARCH<x>:SLOGic:STAtE:GRouP<x>:DATA1」で設定します。

- 「:SEARCH<x>:SLOGic:STAtE:GRouP<x>:CONDition LTHan」のときは

「:SEARCH<x>:SLOGic:STAtE:GRouP<x>:DATA2」で設定します。

- 「:SEARCH<x>:SLOGic:STAtE:GRouP<x>:CONDition BETWen|ORANge」のときは、小さい値を「:SEARCH<x>:SLOGic:STAtE:GRouP<x>:DATA1」、大きい値を

「:SEARCH<x>:SLOGic:STAtE:GRouP<x>:DATA2」で設定します。

### :SEARCH<x>:SLOGic:STAtE:GRouP<x>:HEXA

機能 ロジックステートサーチの各グループの成立条件を HEXA で設定します。

構文 :SEARCH<x>:SLOGic:STAtE:GRouP<x>:HEXA {<文字列>}

SEARCH<x> の <x> = 1, 2

GRouP<x> の <x> = 1 ~ 5

<文字列> = '0' ~ 'F'、'X' の組み合わせ 8 文字以内

例 :SEARCH1:SLOGic:STAtE:GRouP1

:HEXA "1A3F24CD"

解説 「:LOGic:GRouP<x>:MAPPING」で設定したビット配置数が多い場合は下位に "X" が設定されます。少ない場合は上位が設定されます。

### :SEARCH<x>:SLOGic:STAtE:GRouP<x>:PATTERN

機能 ロジックステートサーチの各グループの成立条件を BINARY で設定 / 問い合わせします。

構文 :SEARCH<x>:SLOGic:STAtE:GRouP<x>:

PATtern {<文字列>}

:SEARCH<x>:SLOGic:STAtE:GRouP<x>:

PATtern?

SEARCH<x> の <x> = 1, 2

GRouP<x> の <x> = 1 ~ 5

<文字列> = '0'、'1'、'X' の組み合わせ 32 文字以内

例 :SEARCH1:SLOGic:STAtE:GRouP1:

PATTERN "11110000111100001111000011110000"

:SEARCH1:SLOGic:STAtE:GRouP1:PATTERN?

-> :SEARCH1:SLOGic:STAtE:GRouP1:

PATTERN "11110000111100001111000011110000"

### :SEARCH<x>:SLOGic:STAtE:GRouP<x>:SYMBOL

機能 ロジックステートサーチの各グループのシンボルアイテムを設定します。

構文 :SEARCH<x>:SLOGic:STAtE:GRouP<x>:

SYMBOL {<文字列>}

SEARCH<x> の <x> = 1, 2

GRouP<x> の <x> = 1 ~ 5

<文字列> = 16 文字以内

例 :SEARCH1:SLOGic:STAtE:GRouP1:

SYMBOL "TEST"

### :SEARCH<x>:SLOGic:STAtE:TYPE

機能 ロジックステートサーチの設定方法を設定 / 問い合わせします。

構文 :SEARCH<x>:SLOGic:STAtE:TYPE {BIT|GRouP}

:SEARCH<x>:SLOGic:STAtE:TYPE?

<x> = 1, 2

例 :SEARCH1:SLOGic:STAtE:TYPE BIT

:SEARCH1:SLOGic:STAtE:TYPE?

-> :SEARCH1:SLOGic:STAtE:TYPE BIT

### :SEARCH<x>:SLOGic:WIDTh?

機能 ロジックパルス幅サーチのすべての設定値を問い合わせます。

構文 :SEARCH<x>:SLOGic:WIDTh?

<x> = 1, 2

例 :SEARCH1:SLOGic:WIDTh? -> :SEARCH1:

SLOGic:WIDTh:MODE BETWEEN;

TIME1 1.0000000E+00;

TIME2 1.0000000E+00;TYPE PQUALIFY

**:SEARCh<x>:SLOGic:WIDTh:MODE**

機能 ロジックパルス幅サーチの判定モードを設定 / 問い合わせします。

構文 :SEARCh<x>:SLOGic:WIDTh:MODE  
{BETWEEIn|  
IN|NOTBetween|OUT|TIMEout}  
:SEARCh<x>:SLOGic:WIDTh:MODE?  
<x> = 1, 2

例 :SEARCH1:SLOGIC:WIDTH:MODE BETWEEN  
:SEARCH1:SLOGIC:WIDTH:MODE?  
-> :SEARCH1:SLOGIC:WIDTH:MODE BETWEEN

**:SEARCh<x>:SLOGic:WIDTh:TIME<x>**

機能 ロジックパルス幅サーチのパルス幅を設定 / 問い合わせします。

構文 :SEARCh<x>:SLOGic:WIDTh:  
TIME<x> {<時間>}  
:SEARCh<x>:SLOGic:WIDTh:TIME<x>?  
SEARCh<x> の <x> = 1, 2  
TIME<x> の <x> = 1, 2  
<時間> = 1ns ~ 10s(500ps ステップ)

例 :SEARCH1:SLOGIC:WIDTH:TIME1 1S  
:SEARCH1:SLOGIC:WIDTH:TIME1?  
-> :SEARCH1:SLOGIC:WIDTH:  
TIME1 1.000E+00

解説 TIME2 は、「:SEARCh<x>:SLOGic:WIDTh:MODE  
BETWEEIn|NOTBetween」のときに有効です。

**:SEARCh<x>:SLOGic:WIDTh:TYPE**

機能 ロジックパルス幅サーチタイプを設定 / 問い合わせします。

構文 :SEARCh<x>:SLOGic:WIDTh:  
TYPE {PQUALify|PSTate|PULSe}  
:SEARCh<x>:SLOGic:WIDTh:TYPE?  
<x> = 1, 2

例 :SEARCH1:SLOGIC:WIDTH:TYPE PQUALIFY  
:SEARCH1:SLOGIC:WIDTH:TYPE?  
-> :SEARCH1:SLOGIC:WIDTH:  
TYPE PQUALIFY

**:SEARCh<x>:SMODE**

機能 スキップモードを設定 / 問い合わせします。

構文 :SEARCh<x>:SMODE  
{DECimation|HOLDoff|OFF}  
:SEARCh<x>:SMODE?  
<x> = 1, 2

例 :SEARCH1:SMODE HOLDOFF  
:SEARCH1:SMODE? -> :SEARCH1:  
SMODE HOLDOFF

**:SEARCh<x>:SPATtern? (Serial Pattern)**

機能 シリアルパターンサーチに関するすべての設定値を問い合わせます。

構文 :SEARCh<x>:SPATtern?  
<x> = 1, 2

例 :SEARCH1:SPATTERN? ->:SEARCH1:  
SPATTERN:CLOCK:MODE 1;POLARITY FALL;  
SOURCE 1;:SEARCH1:SPATTERN:CS 1;  
LATCH:TRACE 1;POLARITY FALL;:SEARCH1:  
SPATTERN:SETUP:BITRATE 1.000E+00;  
PATTERN "1100110111101111"

**:SEARCh<x>:SPATtern:CLOCK?**

機能 シリアルパターンサーチのクロックに関するすべての設定値を問い合わせます。

構文 :SEARCh<x>:SPATtern:CLOCK?  
<x> = 1, 2

例 :SEARCH1:SPATTERN:CLOCK? -> :SEARCH1:  
SPATTERN:CLOCK:MODE 1;  
POLARITY FALL;SOURCE 1

**:SEARCh<x>:SPATtern:CLOCK:MODE**

機能 シリアルパターンサーチのクロックの有効 / 無効を設定 / 問い合わせします。

構文 :SEARCh<x>:SPATtern:CLOCK:  
MODE {<Boolean>}  
:SEARCh<x>:SPATtern:CLOCK:MODE?  
<x> = 1, 2

例 :SEARCH1:SPATTERN:CLOCK:MODE ON  
:SEARCH1:SPATTERN:CLOCK:MODE?  
-> :SEARCH1:SPATTERN:CLOCK:MODE 1

**:SEARCh<x>:SPATtern:CLOCK:POLarity**

機能 シリアルパターンサーチのクロックトレースの極性を設定 / 問い合わせします。

構文 :SEARCh<x>:SPATtern:CLOCK:  
POLarity {FALL|RISE}  
:SEARCh<x>:SPATtern:CLOCK:POLarity?  
<x> = 1, 2

例 :SEARCH1:SPATTERN:CLOCK:POLARITY FALL  
:SEARCH1:SPATTERN:CLOCK:POLARITY? ->  
:SEARCH1:SPATTERN:CLOCK:POLARITY FALL

解説 「:SEARCh<x>:SPATtern:CLOCK:MODE ON」のときに有効です。

**:SEARCh<x>:SPATtern:CLOCK:SOURCE**

機能 シリアルパターンサーチのクロックトレースを設定 / 問い合わせします。

構文 :SEARCh<x>:SPATtern:CLOCK:  
SOURCE {<Nrf>}  
:SEARCh<x>:SPATtern:CLOCK:SOURCE?  
<x> = 1, 2  
<Nrf> = 1 ~ 8

例 :SEARCH1:SPATTERN:CLOCK:SOURCE 1  
:SEARCH1:SPATTERN:CLOCK:SOURCE?  
-> :SEARCH1:SPATTERN:CLOCK:SOURCE 1

解説 「:SEARCh<x>:SPATtern:CLOCK:MODE ON」のときに有効です。

## 5.21 SEARCh グループ

### :SEARCh<x>:SPATtern:CS

機能 シリアルパターンサーチのチップセレクトの有効/無効を設定/問い合わせします。

構文 :SEARCh<x>:SPATtern:CS {<Boolean>}  
:SEARCh<x>:SPATtern:CS?  
<x> = 1、2

例 :SEARCH1:SPATTERN:CS ON  
:SEARCH1:SPATTERN:CS? -> :SEARCH1:  
SPATTERN:CS 1

解説 「:SEARCh<x>:SPATtern:CLOCK:MODE ON」のときに有効です。

### :SEARCh<x>:SPATtern:LATCh?

機能 シリアルパターンサーチのラッチに関するすべての設定値を問い合わせます。

構文 :SEARCh<x>:SPATtern:LATCh?  
<x> = 1、2

例 :SEARCH1:SPATTERN:LATCH? -> :SEARCH1:  
SPATTERN:LATCH:TRACE 1;  
POLARITY FALL

### :SEARCh<x>:SPATtern:LATCh:POLarity

機能 シリアルパターンサーチのラッチトレースの極性を設定/問い合わせします。

構文 :SEARCh<x>:SPATtern:LATCh:  
POLarity {FALL|RISE}  
:SEARCh<x>:SPATtern:LATCh:POLarity?  
<x> = 1、2

例 :SEARCH1:SPATTERN:LATCH:POLARITY FALL  
:SEARCH1:SPATTERN:LATCH:POLARITY? ->  
:SEARCH1:SPATTERN:LATCH:POLARITY FALL

解説 ・「:SEARCh<x>:SPATtern:CLOCK:MODE ON」のときに有効です。  
・「:SEARCh<x>:SPATtern:LATCh:TRACE NONE」のときは無効です。

### :SEARCh<x>:SPATtern:LATCh:TRACe

機能 シリアルパターンサーチのラッチトレースを設定/問い合わせします。

構文 :SEARCh<x>:SPATtern:LATCh:TRACe  
{<NRf>}  
NONE}  
:SEARCh<x>:SPATtern:LATCh:TRACe?  
<x> = 1、2  
<NRf> = 1 ~ 8

例 :SEARCH1:SPATTERN:LATCH:TRACE 1  
:SEARCH1:SPATTERN:LATCH:TRACE?  
-> :SEARCH1:SPATTERN:LATCH:TRACE 1

解説 「:SEARCh<x>:SPATtern:CLOCK:MODE ON」のときに有効です。

### :SEARCh<x>:SPATtern:SETup?

機能 シリアルパターンサーチのセットアップに関するすべての設定値を問い合わせます。

構文 :SEARCh<x>:SPATtern:SETup?  
<x> = 1、2

例 :SEARCH1:SPATTERN:SETUP? -> :SEARCH1:  
SPATTERN:SETUP:BITRATE 1.000E+03;  
DATA:ACTIVE HIGH;TRACE 1;:SEARCH1:  
SPATTERN:SETUP:PATTERN "1100"

### :SEARCh<x>:SPATtern[:SETup]:BITRate

機能 シリアルパターンサーチのビットレートを設定/問い合わせします。

構文 :SEARCh<x>:SPATtern[:SETup]:  
BITRate {<NRf>}  
:SEARCh<x>:SPATtern[:SETup]:BITRate?  
<x> = 1、2  
<NRf> = 1 ~ 1G(bps)

例 :SEARCH1:SPATTERN:SETUP:BITRATE 1  
:SEARCH1:SPATTERN:SETUP:BITRATE?  
-> :SEARCH1:SPATTERN:SETUP:  
BITRATE 1.000E+00

解説 「:SEARCh<x>:SPATtern:CLOCK:MODE OFF」のときに有効です。

### :SEARCh<x>:SPATtern[:SETup]:CLEar

機能 シリアルパターンサーチのパターンをすべてクリア (Don't care) します。

構文 :SEARCh<x>:SPATtern[:SETup]:CLEar  
<x> = 1、2

例 :SEARCH1:SPATTERN:SETUP:CLEAR

### :SEARCh<x>:SPATtern[:SETup]:DATA?

機能 シリアルパターンサーチのデータに関するすべての設定値を問い合わせます。

構文 :SEARCh<x>:SPATtern[:SETup]:DATA?  
<x> = 1、2

例 :SEARCH1:SPATTERN:SETUP:DATA?  
-> :SEARCH1:SPATTERN:SETUP:DATA:  
ACTIVE HIGH;TRACE 1

### :SEARCh<x>:SPATtern[:SETup]:DATA:ACTivE

機能 シリアルパターンサーチのデータのトレースのアクティブレベルを設定/問い合わせします。

構文 :SEARCh<x>:SPATtern[:SETup]:DATA:  
ACTivE {HIGH|LOW}  
:SEARCh<x>:SPATtern[:SETup]:DATA:  
ACTivE?  
<x> = 1、2

例 :SEARCH1:SPATTERN:SETUP:DATA:  
ACTIVE HIGH  
:SEARCH1:SPATTERN:SETUP:DATA:ACTIVE?  
-> :SEARCH1:SPATTERN:SETUP:DATA:  
ACTIVE HIGH

**:SEARCH<x>:SPATtern[:SETup]:DATA:****TRACe**

機能 シリアルパターンサーチのデータのトレースを設定 / 問い合わせします。

構文 :SEARCH<x>:SPATtern[:SETup]:DATA:  
TRACe {<Nrf>}  
:SEARCH<x>:SPATtern[:SETup]:DATA:  
TRACe?

<x> = 1, 2  
<Nrf> = 1 ~ 8

例 :SEARCH1:SPATtern:SETUP:DATA:TRACE 1  
:SEARCH1:SPATtern:SETUP:DATA:TRACE?  
-> :SEARCH1:SPATtern:SETUP:DATA:  
TRACE 1

**:SEARCH<x>:SPATtern[:SETup]:HEXA**

機能 シリアルパターンサーチのパターンを HEXA で設定します。

構文 :SEARCH<x>:SPATtern[:SETup]:  
HEXA {<文字列>}

<x> = 1, 2  
<文字列> = '0' ~ 'F', 'X' の組み合わせ 32 文字以内

例 :SEARCH1:SPATtern:SETUP:HEXA "ABCD"

**:SEARCH<x>:SPATtern[:SETup]:PATTern**

機能 シリアルパターンサーチのパターンを BINARY で設定 / 問い合わせします。

構文 :SEARCH<x>:SPATtern[:SETup]:  
PATTern {<文字列>}  
:SEARCH<x>:SPATtern[:SETup]:PATTern?  
<x> = 1, 2

<文字列> = '0', '1', 'X' の組み合わせ 128 文字以内

例 :SEARCH1:SPATtern:SETUP:  
PATTern "1100110111101111"  
:SEARCH1:SPATtern:SETUP:PATTern?  
-> :SEARCH1:SPATtern:SETUP:  
PATTern "1100110111101111"

**:SEARCH<x>:SPOint**

機能 サーチ開始位置を設定 / 問い合わせします。

構文 :SEARCH<x>:SPOint {<Nrf>}  
:SEARCH<x>:SPOint?

<x> = 1, 2  
<Nrf> = -5 ~ 5(div)

例 :SEARCH1:SPOint 1  
:SEARCH1:SPOint? -> :SEARCH1:  
SPOint 1.000E+00

**:SEARCH<x>:STRace**

機能 サーチ対象トレースを設定 / 問い合わせします。

構文 :SEARCH<x>:STRace {<Nrf>}  
:SEARCH<x>:STRace?

<x> = 1, 2  
<Nrf> = 1 ~ 8

例 :SEARCH1:STRace 1  
:SEARCH1:STRace? -> :SEARCH1:STRace 1

解説 ・「:SEARCH<x>:TYPE EDGE|EQUalify」のときに有効です。  
・「:SEARCH<x>:TYPE WIDTH」且つ「:SEARCH<x>:WIDTH:TYPE PQUalify|PULSe」のときに有効です。

**:SEARCH<x>:TRACe<x>?**

機能 各トレースのサーチ条件に関するすべての設定値を問い合わせます。

構文 :SEARCH<x>:TRACe<x>?  
SEARCH<x>の<x> = 1, 2  
TRACe<x>の<x> = 1 ~ 8

例 :SEARCH1:TRACE1? ->  
:SEARCH1:TRACE1:CONDITION DONTCARE;  
HYSTERESIS 1.000E+00;LEVEL 0.000E+00

**:SEARCH<x>:TRACe<x>:CONDition**

機能 各トレースの成立条件を設定 / 問い合わせします。

構文 :SEARCH<x>:TRACe<x>:  
CONDition {DONTcare|HIGH|LOW}  
:SEARCH<x>:TRACe<x>:CONDition?  
SEARCH<x>の<x> = 1, 2  
TRACe<x>の<x> = 1 ~ 8

例 :SEARCH1:TRACE1:CONDITION HIGH  
:SEARCH1:TRACE1:CONDITION?

-> :SEARCH1:TRACE1:CONDITION HIGH  
解説 ・「:SEARCH<x>:TYPE EQUalify|SPATtern|STAtE」のときに有効です。  
・「:SEARCH<x>:TYPE WIDTH」且つ「:SEARCH<x>:WIDTH:TYPE PQUalify|PSTAtE」のときに有効です。

**:SEARCH<x>:TRACe<x>:HYSTeresis**

機能 各トレースのヒステリシスを設定 / 問い合わせします。

構文 :SEARCH<x>:TRACe<x>:HYSTeresis  
{<Nrf>}

:SEARCH<x>:TRACe<x>:HYSTeresis?  
SEARCH<x>の<x> = 1, 2

TRACe<x>の<x> = 1 ~ 8  
<Nrf> = 0 ~ 4(div, 0.1div ステップ)

例 :SEARCH1:TRACE1:HYSTERESIS 1  
:SEARCH1:TRACE1:HYSTERESIS?

-> :SEARCH1:TRACE1:  
HYSTERESIS 1.000E+00

## 5.21 SEARCh グループ

### :SEARCh<x>:TRACe<x>:LEVel

機能 各トレースのしきい値 (Threshold) レベルを設定 / 問い合わせします。

構文 :SEARCh<x>:TRACe<x>:LEVel {<Nrf>|<電圧>|<電流>}  
:SEARCh<x>:TRACe<x>:LEVel?  
SEARCh<x> の <x> = 1、2  
TRACe<x> の <x> = 1 ~ 8  
<Nrf>、<電圧>、<電流> = 本体ユーザズマニュアル参照。

例 :SEARCH1:TRACE1:LEVEL 0  
:SEARCH1:TRACE1:LEVEL? -> :SEARCH1:  
TRACE1:LEVEL 0.000E+00

### :SEARCh<x>:TYPE

機能 サーチタイプを設定 / 問い合わせします。

構文 :SEARCh<x>:TYPE {EDGE|EQualify|LEDge|LINBus|LQualify|LSPAttern|LState|LWIDth|SPAttern|STATE|WIDTh}  
:SEARCh<x>:TYPE?  
<x> = 1、2

例 :SEARCH1:TYPE EDGE  
:SEARCH1:TYPE? -> :SEARCH1:TYPE EDGE

### :SEARCh<x>:WIDTh?

機能 パルス幅サーチに関するすべての設定値を問い合わせます。

構文 :SEARCh<x>:WIDTh?  
<x> = 1、2

例 :SEARCH1:WIDTH? -> :SEARCH1:WIDTH:  
MODE OUT;TIME1 1.000E-09;  
TIME2 2.000E-09;TYPE PULSE

### :SEARCh<x>:WIDTh:MODE

機能 パルス幅判定モードを設定 / 問い合わせします。

構文 :SEARCh<x>:WIDTh:MODE {BETWeen|IN|NOTBetween|OUT|TIMEout}  
:SEARCh<x>:WIDTh:MODE?  
<x> = 1、2

例 :SEARCH1:WIDTH:MODE TIMEOUT  
:SEARCH1:WIDTH:MODE? -> :SEARCH1:  
WIDTH:MODE TIMEOUT

### :SEARCh<x>:WIDTh:TIME<x>

機能 パルス幅サーチのパルス幅を設定 / 問い合わせします。

構文 :SEARCh<x>:WIDTh:TIME<x> {<時間>}  
:SEARCh<x>:WIDTh:TIME<x>?  
SEARCh<x> の <x> = 1、2  
TIME<x> の <x> = 1、2  
<時間> = 1ns ~ 10s(500ps ステップ)

例 :SEARCH1:WIDTH:TIME1 1S  
:SEARCH1:WIDTH:TIME1? -> :SEARCH1:  
WIDTH:TIME1 1.000E+00

解説 TIME2 は、「:SEARCh<x>:WIDTh:MODE BETWEEN|NOTBetween」のときに有効です。

### :SEARCh<x>:WIDTh:TYPE

機能 パルス幅サーチタイプを設定 / 問い合わせします。

構文 :SEARCh<x>:WIDTh:  
TYPE {PQualify|PState|PULSe}  
:SEARCh<x>:WIDTh:TYPE?  
<x> = 1、2

例 :SEARCH1:WIDTH:TYPE PQUALIFY  
:SEARCH1:WIDTH:TYPE? -> :SEARCH1:  
WIDTH:TYPE PQUALIFY

## 5.22 SNAP グループ

**: SNAP**

機能 スナップショットを実行します。

構文 :SNAP

例 :SNAP

## 5.23 SStart グループ

### :SStart?

**機能** トリガモードをシングルにして START し、指定時間内に STOP した場合にその時点で 0 を返します。指定時間内に STOP しなかった場合は、1 を返します。

**構文** :SStart? {<Nrf>}  
<Nrf> = 1 ~ 360000(10ms 単位: 待ち時間、START して待つ)  
0(START するだけ。待ちなし)  
- 360000 ~ -1(10ms 単位: 待ち時間、START しないで待つ)

**例** :SSTART? 100 -> :SSTART 0

**解説**

- ・ 指定時間が + 値の場合は、指定時間内において、SINGLE TRIGGER で START して、STOP するのを待ちます。
- ・ 指定時間が 0 の場合は、START して、STOP を待たずに 0 が返って来ます。
- ・ 指定時間が - 値の場合は、START せずにただ、指定時間内で STOP するのを待ちます。

## 5.24 STARt グループ

### : STARt

機能	波形の取り込みをスタートします。
構文	:STARt
例	:START
解説	取り込みのストップは、「STOP」で行います。



## 5.25 STATus グループ

STATus グループは、通信のステータス機能に関する設定と問い合わせを行うグループです。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。ステータスレポートについては、第 6 章をご覧ください。

### : STATus?

機能 通信のステータス機能に関連する設定をすべて問い合わせます。

構文 : STATus?

例 : STATUS? -> : STATUS:EESE 0;  
FILTER1 NEVER; FILTER2 NEVER;  
FILTER3 NEVER; FILTER4 NEVER;  
FILTER5 NEVER; FILTER6 NEVER;  
FILTER7 NEVER; FILTER8 NEVER;  
FILTER9 NEVER; FILTER10 NEVER;  
FILTER11 NEVER; FILTER12 NEVER;  
FILTER13 NEVER; FILTER14 NEVER;  
FILTER15 NEVER; FILTER16 NEVER;  
QENABLE 1; QMESSAGE 1

### : STATus: CONDItion?

機能 状態レジスタの内容の問い合わせます。

構文 : STATus: CONDItion?

例 : STATUS: CONDITION -> 16

解説 状態レジスタについては、「第 6 章 ステータスレポート」をご覧ください。

### : STATus: EESE

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: EESE <Register>  
: STATus: EESE?

<Register> = 0 ~ 65535

例 : STATUS: EESE 257

: STATUS: EESE? -> : STATUS: EESE 257

解説 拡張イベントイネーブルレジスタについては、「第 6 章 ステータスレポート」をご覧ください。

### : STATus: EESR?

機能 拡張イベントレジスタの内容の問い合わせ、レジスタをクリアします。

構文 : STATus: EESR?

例 : STATUS: EESR? -> 1

解説 拡張イベントレジスタについては、「第 6 章 ステータスレポート」をご覧ください。

### : 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

解説 遷移フィルタについては、「第 6 章 ステータスレポート」をご覧ください。

### : STATus: QENable

機能 エラー以外のメッセージをエラーキューに格納するかしないかを設定 / 問い合わせします。

構文 : STATus: QENable {<Boolean>}

: STATus: QENable?

例 : STATUS: QENABLE ON

: STATUS: QENABLE? -> : STATUS: QENABLE 1

### : STATus: QMESSage

機能 「: STATus: ERRor?」の応答にメッセージ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

### : STATus: SPOLL? (Serial Poll)

機能 シリアルポーラを実行します。

構文 : STATus: SPOLL?

例 : STATUS: SPOLL? -> STATUS: SPOLL 0

解説 イーサネット (オプション) インタフェース専用のコマンドです。

## 5.26 STOP グループ

### **: STOP**

機能	波形の取り込みをストップします。
構文	:STOP
例	:STOP
解説	取り込みのスタートは、「START」で行います。

## 5.27 SYSTem グループ

### :SYSTem?

機能 システムに関するすべての設定値を問い合わせます。

構文 :SYSTem?

例 :SYSTEM? -> :SYSTEM:CLICK 1;CLOCK:DTIME "2007/01/06","11:37:32","09:00";MODE 1;:SYSTEM:LANGUAGE JAPANESE;MFSIZE SMALL;MLanguage ENGLISH;USBKEYBOARD ENGLISH

### :SYSTem:CLICk

機能 クリック音の ON/OFF を設定 / 問い合わせします。

構文 :SYSTem:CLICk {<Boolean>}

例 :SYSTEM:CLICK ON

:SYSTEM:CLICK? -> :SYSTEM:CLICK 1

### :SYSTem:CLOCk?

機能 日付・時刻・GMT からの時差に関するすべての設定値を問い合わせます。

構文 :SYSTem:CLOCk?

例 :SYSTEM:CLOCK? -> :SYSTEM:CLOCK:DTIME "2007/01/06","11:37:32","09:00";MODE 1

### :SYSTem:CLOCk:DTIME

機能 日付・時刻・GMT からの時差を設定 / 問い合わせします。

構文 :SYSTem:CLOCk:DTIME {<文字列>,<文字列>,<文字列>}

:SYSTem:CLOCk:DTIME?

左側の <文字列> = YYYY/MM/DD、本体ユーザーマニュアル参照。

中央の <文字列> = HH:MM:SS、本体ユーザーズマニュアル参照。

右側の <文字列> = HH:MM、本体ユーザーズマニュアル参照。

例 :SYSTEM:CLOCK:DTIME "2005/05/06","11:37:32","09:00"

:SYSTEM:CLOCK:DTIME? -> :SYSTEM:CLOCK:DTIME "2005/05/06","11:37:32","09:00"

### :SYSTem:CLOCk:MODE

機能 日付・時刻・GMT からの時差表示の ON/OFF を設定 / 問い合わせします。

構文 :SYSTem:CLOCk:MODE {<Boolean>}

:SYSTem:CLOCk:MODE?

例 :SYSTEM:CLOCK:MODE ON

:SYSTEM:CLOCK:MODE? -> :SYSTEM:CLOCK:MODE 1

### :SYSTem:FORMat:IMEMory[:EXECute]

機能 内部メモリをフォーマットします。

構文 :SYSTem:FORMat:IMEMory[:EXECute]

例 :SYSTEM:FORMAT:IMEMORY:EXECUTE

### :SYSTem:FORMat:IHDD[:EXECute]

機能 内蔵ハードディスクをフォーマットします。

構文 :SYSTem:FORMat:IHDD[:EXECute]

例 :SYSTEM:FORMAT:IHDD:EXECUTE

### :SYSTem:FORMat:SDElete[:EXECute] (Sure Delete)

機能 内部メモリを消去し、フォーマットします。

構文 :SYSTem:FORMat:SDElete[:EXECute]

例 :SYSTEM:FORMAT:SDELETE:EXECUTE

### :SYSTem:LANGuage

機能 メッセージの言語を設定 / 問い合わせします。

構文 :SYSTem:LANGuage {CHINese|ENGLish|JAPANese|KOREan}

:SYSTem:LANGuage?

例 :SYSTEM:LANGUAGE JAPANESE

:SYSTEM:LANGUAGE? -> :SYSTEM:LANGUAGE JAPANESE

### :SYSTem:MFSIZE

機能 メニューのフォントサイズを設定 / 問い合わせします。

構文 :SYSTem:MFSIZE {LARGE|SMALL}

例 :SYSTEM:MFSIZE LARGE

:SYSTEM:MFSIZE? -> :SYSTEM:MFSIZE LARGE

### :SYSTem:MLanguage

機能 メニューの言語を設定 / 問い合わせします。

構文 :SYSTem:MLanguage {CHINese|ENGLish|JAPANese|KOREan}

:SYSTem:MLanguage?

例 :SYSTEM:MLANGUAGE ENGLISH

:SYSTEM:MLANGUAGE? -> :SYSTEM:MLANGUAGE ENGLISH

### :SYSTem:OVERview

機能 システム情報を表示します。

構文 :SYSTem:OVERview

例 :SYSTEM:OVERVIEW

### :SYSTem:USBKeyboard

機能 USB キーボードの種類を設定 / 問い合わせします。

構文 :SYSTem:USBKeyboard {ENGLish|JAPANese}

:SYSTem:USBKeyboard?

例 :SYSTEM:USBKEYBOARD ENGLISH

:SYSTEM:USBKEYBOARD? -> :SYSTEM:USBKEYBOARD ENGLISH

## 5.28 TELecomtest グループ

### :TELecomtest?

機能 テレコムテストに関するすべての設定値を問い合わせます。

構文 :TELecomtest? -> :TELECOMTEST:  
CATEGORY MASK;DISPLAY 0;EYEPATTERN:  
DBERATE:STATE 0;:TELECOMTEST:  
EYEPATTERN:EHEIGHT:STATE 0;:  
TELECOMTEST:EYEPATTERN:EWIDTH:  
STATE 0;:TELECOMTEST:EYEPATTERN:FALL:  
STATE 0;:TELECOMTEST:EYEPATTERN:  
JITTER1:STATE 0;:TELECOMTEST:  
EYEPATTERN:JITTER2:STATE 1;:  
TELECOMTEST:EYEPATTERN:PCROSSING:  
STATE 0;:TELECOMTEST:EYEPATTERN:  
PDUTYCYCLE:STATE 0;:TELECOMTEST:  
EYEPATTERN:QFACTOR:STATE 0;:  
TELECOMTEST:EYEPATTERN:RISE:STATE 0;:  
TELECOMTEST:EYEPATTERN:SDBASE:  
STATE 0;:TELECOMTEST:EYEPATTERN:  
SDTOP:STATE 0;:TELECOMTEST:  
EYEPATTERN:T1CROSSING:STATE 0;:  
TELECOMTEST:EYEPATTERN:T2CROSSING:  
STATE 0;:TELECOMTEST:EYEPATTERN:  
TLEVELS:MODE PERCENT;PERCENT 90,10;  
UNIT 1.000E+00,0.000E+00;:  
TELECOMTEST:EYEPATTERN:VBASE:  
STATE 0.....

### :TELecomtest:CATegory

機能 テレコムテストの種類を設定 / 問い合わせします。

構文 :TELecomtest:CATegory  
{EYEPattern|MASK}  
:TELecomtest:CATegory?  
例 :TELECOMTEST:CATEGORY EYEPATTERN  
:TELECOMTEST:CATEGORY?  
-> :TELECOMTEST:CATEGORY EYEPATTERN

### :TELecomtest:DISPlay

機能 テレコムテストの表示の ON/OFF を設定 / 問い合わせします。

構文 :TELecomtest:DISPlay {<Boolean>}  
:TELecomtest:DISPlay?  
例 :TELECOMTEST:DISPLAY ON  
:TELECOMTEST:DISPLAY?  
-> :TELECOMTEST:DISPLAY 1

### :TELecomtest:EYEPattern?

機能 アイパターンに関するすべての設定値を問い合わせます。

構文 :TELecomtest:EYEPattern?  
例 :TELECOMTEST:EYEPATTERN?  
-> :TELECOMTEST:EYEPATTERN:DBERATE:  
STATE 1;:TELECOMTEST:EYEPATTERN:  
EHEIGHT:STATE 1;:TELECOMTEST:  
EYEPATTERN:EWIDTH:STATE 1;:  
TELECOMTEST:EYEPATTERN:FALL:STATE 1;:  
TELECOMTEST:EYEPATTERN:JITTER1:  
STATE 1;:TELECOMTEST:EYEPATTERN:  
JITTER2:STATE 1;:TELECOMTEST:  
EYEPATTERN:PCROSSING:STATE 1;:  
TELECOMTEST:EYEPATTERN:PDUTYCYCLE:  
STATE 1;:TELECOMTEST:EYEPATTERN:  
QFACTOR:STATE 1;:TELECOMTEST:  
EYEPATTERN:RISE:STATE 1;:TELECOMTEST:  
EYEPATTERN:SDBASE:STATE 1;:  
TELECOMTEST:EYEPATTERN:SDTOP:  
STATE 1;:TELECOMTEST:EYEPATTERN:  
T1CROSSING:STATE 1;:TELECOMTEST:  
EYEPATTERN:T2CROSSING:STATE 1;:  
TELECOMTEST:EYEPATTERN:TLEVELS:MODE  
PERCENT;PERCENT 90,10;UNIT 1.000E+00,  
0.000E+00;:TELECOMTEST:EYEPATTERN:  
VBASE:STATE 1;:TELECOMTEST:  
EYEPATTERN:VCROSSING:STATE 1;:  
TELECOMTEST:EYEPATTERN:VDARK 1;VTOP:  
STATE 1

### :TELecomtest:EYEPattern:ALL

機能 アイパターンのパラメータのすべてを ON/OFF します。

構文 :TELecomtest:EYEPattern:ALL  
{<Boolean>}  
例 :TELECOMTEST:EYEPATTERN:ALL ON

## 5.28 TELecomtest グループ

### **:TELecomtest:EYEPattern:<パラメータ>?**

**機能** アイパターンの波形パラメータに関するすべての設定値を問い合わせます。

**構文** :TELecomtest:EYEPattern:<パラメータ>  
<パラメータ> = {DBERate|EHEight|EWIDth|FALL|JITter<x>|PCROssing|PDUtYcycle|QFActor|RISE|SDBase|SDTop|T1CRossing|T2CRossing|VBASe|VCRossing|VTOP}  
<x>=1、2

**例** (以下は、DBERate についての例です。)  
:TELECOMTEST:EYEPATTERN:DBERATE?  
-> :TELECOMTEST:EYEPATTERN:DBERATE:  
STATE 1

**解説**

- ・通信コマンドと本体で使用されるパラメータの対応は、付録 4 を参照。
- ・パラメータの詳細は、本体ユーザーズマニュアル参照。

### **:TELecomtest:EYEPattern:<パラメータ>:STATE**

**機能** アイパターンの波形パラメータの ON/OFF を設定 / 問い合わせします。

**構文** :TELecomtest:EYEPattern:<パラメータ>:  
STATE {<Boolean>}  
:TELecomtest:EYEPattern:<パラメータ>:  
STATE?  
<パラメータ> = {DBERate|EHEight|EWIDth|FALL|JITter<x>|PCROssing|PDUtYcycle|QFActor|RISE|SDBase|SDTop|T1CRossing|T2CRossing|VBASe|VCRossing|VTOP}  
<x>=1、2

**例** (以下は、DBERate についての例です。)  
:TELECOMTEST:EYEPATTERN:DBERATE:  
STATE ON  
:TELECOMTEST:EYEPATTERN:DBERATE:  
STATE? -> :TELECOMTEST:EYEPATTERN:  
DBERATE:STATE 1

### **:TELecomtest:EYEPattern:<パラメータ>:VALue?**

**機能** アイパターンの波形パラメータの値を問い合わせます。

**構文** :TELecomtest:EYEPattern:<パラメータ>:  
VALue?  
<パラメータ> = {DBERate|EHEight|EWIDth|FALL|JITter<x>|PCROssing|PDUtYcycle|QFActor|RISE|SDBase|SDTop|T1CRossing|T2CRossing|VBASe|VCRossing|VTOP}  
<x>=1、2

**例** (以下は、DBERate についての例です。)  
:TELECOMTEST:EYEPATTERN:DBERATE:  
VALUE? -> :TELECOMTEST:EYEPATTERN:  
DBERATE:VALUE 1.000E+00

### **:TELecomtest:EYEPattern:TLEVels?**

**機能** アイパターンのしきい値 (Threshold) レベルに関するすべての設定値を問い合わせます。

**構文** :TELecomtest:EYEPattern:TLEVels?  
**例** :TELECOMTEST:EYEPATTERN:TLEVELS?  
-> :TELECOMTEST:EYEPATTERN:TLEVELS:  
MODE PERCENT;PERCENT 90,10;  
UNIT 1.000E+00,0.000E+00

### **:TELecomtest:EYEPattern:TLEVels:MODE**

**機能** アイパターンのしきい値 (Threshold) レベルの単位を設定 / 問い合わせします。

**構文** :TELecomtest:EYEPattern:TLEVels:  
MODE {PERCent|UNIT}  
:TELecomtest:EYEPattern:TLEVels:MODE?  
**例** :TELECOMTEST:EYEPATTERN:TLEVELS:  
MODE PERCENT  
:TELECOMTEST:EYEPATTERN:TLEVELS:MODE?  
-> :TELECOMTEST:EYEPATTERN:TLEVELS:  
MODE PERCENT

### **:TELecomtest:EYEPattern:TLEVels:PERCent**

**機能** アイパターンのしきい値 (Threshold) レベルを % で設定 / 問い合わせします。

**構文** :TELecomtest:EYEPattern:TLEVels:  
PERCent {<NRf>,<NRf>}  
:TELecomtest:EYEPattern:TLEVels:  
PERCent?  
<NRf> = 0 ~ 100(%)  
**例** :TELECOMTEST:EYEPATTERN:TLEVELS:  
PERCENT 90,10  
:TELECOMTEST:EYEPATTERN:TLEVELS:  
PERCENT? -> :TELECOMTEST:EYEPATTERN:  
TLEVELS:PERCENT 90,10

### **:TELecomtest:EYEPattern:TLEVels:UNIT**

**機能** アイパターンのしきい値 (Threshold) レベルを UNIT で設定 / 問い合わせします。

**構文** :TELecomtest:EYEPattern:TLEVels:  
UNIT {<NRf>,<NRf>|<電圧>,<電圧>|<電流>,<電流>}  
:TELecomtest:EYEPattern:TLEVels:UNIT?  
<NRf>,<電圧>,<電流> = 本体ユーザーズマニュアル参照。  
**例** :TELECOMTEST:EYEPATTERN:TLEVELS:  
UNIT 1,0  
:TELECOMTEST:EYEPATTERN:TLEVELS:UNIT?  
-> :TELECOMTEST:EYEPATTERN:TLEVELS:  
UNIT 1.000E+00,0.000E+00

**:TELecomtest:EYEPattern:VDARK**

機能 アイパターンのダークレベル(ゼロライトレベル)を設定/問い合わせします。

構文 :TELecomtest:EYEPattern:  
VDARK {<Nrf>|<電圧>|<電流>}  
:TELecomtest:EYEPattern:VDARK?  
<Nrf>、<電圧>、<電流>=本体ユーザーズマニュアル参照。

例 :TELECOMTEST:EYEPATTERN:  
VDARK 1.000E+00  
:TELECOMTEST:EYEPATTERN:VDARK?  
-> :TELECOMTEST:EYEPATTERN:  
VDARK 1.000E+00

**:TELecomtest:MASK?**

機能 マスクテストに関するすべての設定値を問い合わせます。

構文 :TELecomtest:MASK?

例 :TELECOMTEST:MASK? -> :TELECOMTEST:  
MASK:ELEMENT1:PSPCOUNT:STATE 1;;  
TELECOMTEST:MASK:ELEMENT1:PWCOUNT:  
STATE 1;;TELECOMTEST:MASK:ELEMENT1:  
SPCOUNT:STATE 1;;TELECOMTEST:MASK:  
ELEMENT1:WCOUNT:STATE 1;;TELECOMTEST:  
MASK:ELEMENT2:PSPCOUNT:STATE 1;;  
TELECOMTEST:MASK:ELEMENT2:PWCOUNT:  
STATE 1;;TELECOMTEST:MASK:ELEMENT2:  
SPCOUNT:STATE 1;;TELECOMTEST:MASK:  
ELEMENT2:WCOUNT:STATE 1;;TELECOMTEST:  
MASK:ELEMENT3:PSPCOUNT:STATE 1;;  
TELECOMTEST:MASK:ELEMENT3:PWCOUNT:  
STATE 1;;TELECOMTEST:MASK:ELEMENT3:  
SPCOUNT:STATE 1;;TELECOMTEST:MASK:  
ELEMENT3:WCOUNT:STATE 1;;TELECOMTEST:  
MASK:ELEMENT4:PSPCOUNT:STATE 1;;  
TELECOMTEST:MASK:ELEMENT4:PWCOUNT:  
STATE 1;;TELECOMTEST:MASK:ELEMENT4:  
SPCOUNT:STATE 1;;TELECOMTEST:MASK:  
ELEMENT4:WCOUNT:STATE 1

**:TELecomtest:MASK:ELEMENT<x>?**

機能 マスクテストに使用する各エレメントに関するすべての設定値を問い合わせます。

構文 :TELecomtest:MASK:ELEMENT<x>?  
<x> = 1 ~ 4

例 :TELECOMTEST:MASK:ELEMENT1?  
-> :TELECOMTEST:MASK:ELEMENT1:  
PSPCOUNT:STATE 1;;TELECOMTEST:  
MASK:ELEMENT1:PWCOUNT:STATE 1;;  
TELECOMTEST:MASK:ELEMENT1:SPCOUNT:  
STATE 1;;TELECOMTEST:MASK:ELEMENT1:  
WCOUNT:STATE 1

**:TELecomtest:MASK:ELEMENT<x>:ALL**

機能 各エレメントのすべてのアイテムを一斉に ON/OFF します。

構文 :TELecomtest:MASK:ELEMENT<x>:  
ALL {<Boolean>}  
<x> = 1 ~ 4

例 :TELECOMTEST:MASK:ELEMENT1:ALL ON

**:TELecomtest:MASK:ELEMENT<x>:****PSPCount? (Sample Point Count %)**

機能 各エレメントのサンプルデータ数に対するエラー率に関する設定値を問い合わせます。

構文 :TELecomtest:MASK:ELEMENT<x>:  
PSPCount?  
<x> = 1 ~ 4

例 :TELECOMTEST:MASK:ELEMENT1:PSPCOUNT?  
-> :TELECOMTEST:MASK:ELEMENT1:  
PSPCOUNT:STATE 1

**:TELecomtest:MASK:ELEMENT<x>:****PSPCount:STATE**

機能 各エレメントのサンプルデータ数に対するエラー率の測定の ON/OFF を設定/問い合わせします。

構文 :TELecomtest:MASK:ELEMENT<x>:  
PSPCount:STATE {<Boolean>}  
:TELecomtest:MASK:ELEMENT<x>:  
PSPCount:STATE?  
<x> = 1 ~ 4

例 :TELECOMTEST:MASK:ELEMENT1:PSPCOUNT:  
STATE ON  
:TELECOMTEST:MASK:ELEMENT1:PSPCOUNT:  
STATE? -> :TELECOMTEST:MASK:ELEMENT1:  
PSPCOUNT:STATE 1

**:TELecomtest:MASK:ELEMENT<x>:****PSPCount:VALUE?**

機能 各エレメントのサンプルデータ数に対するエラー率を問い合わせます。

構文 :TELecomtest:MASK:ELEMENT<x>:  
PSPCount:VALUE?  
<x> = 1 ~ 4

例 :TELECOMTEST:MASK:ELEMENT1:PSPCOUNT:  
VALUE? -> :TELECOMTEST:MASK:ELEMENT1:  
PSPCOUNT:VALUE 1.000E+00

**:TELecomtest:MASK:ELEMENT<x>:****PWCount? (Wave Count %)**

機能 各エレメントのアクイジション回数に対するエラー率に関する設定値を問い合わせます。

構文 :TELecomtest:MASK:ELEMENT<x>:PWCount?  
<x> = 1 ~ 4

例 :TELECOMTEST:MASK:ELEMENT1:PWCOUNT?  
-> :TELECOMTEST:MASK:ELEMENT1:  
PWCOUNT:STATE 1

## 5.28 TELecomtest グループ

### **:TELecomtest:MASK:ELEMent<x>:**

#### **PWCount:STATE**

機能 各エレメントのアクイジション回数に対するエラー率の測定の ON/OFF を設定 / 問い合わせしません。

構文 :TELecomtest:MASK:ELEMent<x>:PWCount:STATE {<Boolean>}  
:TELecomtest:MASK:ELEMent<x>:PWCount:STATE?  
<x> = 1 ~ 4

例 :TELECOMTEST:MASK:ELEMENT1:PWCOUNT:STATE ON  
:TELECOMTEST:MASK:ELEMENT1:PWCOUNT:STATE? -> :TELECOMTEST:MASK:ELEMENT1:PWCOUNT:STATE 1

### **:TELecomtest:MASK:ELEMent<x>:**

#### **PWCount:VALue?**

機能 各エレメントのアクイジション回数に対するエラー率を問い合わせます。

構文 :TELecomtest:MASK:ELEMent<x>:PWCount:VALue?  
<x> = 1 ~ 4

例 :TELECOMTEST:MASK:ELEMENT1:PWCOUNT:VALUE? -> :TELECOMTEST:MASK:ELEMENT1:PWCOUNT:VALUE 1.000E+00

### **:TELecomtest:MASK:ELEMent<x>:**

#### **SPCount? (Sample Point Count)**

機能 各エレメントのエラーになったサンプルデータ数に関する設定値を問い合わせます。

構文 :TELecomtest:MASK:ELEMent<x>:SPCount?  
<x> = 1 ~ 4

例 :TELECOMTEST:MASK:ELEMENT1:SPCOUNT?  
-> :TELECOMTEST:MASK:ELEMENT1:SPCOUNT:STATE 1

### **:TELecomtest:MASK:ELEMent<x>:**

#### **SPCount:STATE**

機能 各エレメントのエラーになったサンプルデータ数の測定の ON/OFF を設定 / 問い合わせします。

構文 :TELecomtest:MASK:ELEMent<x>:SPCount:STATE {<Boolean>}  
:TELecomtest:MASK:ELEMent<x>:SPCount:STATE?  
<x> = 1 ~ 4

例 :TELECOMTEST:MASK:ELEMENT1:SPCOUNT:STATE ON  
:TELECOMTEST:MASK:ELEMENT1:SPCOUNT:STATE? -> :TELECOMTEST:MASK:ELEMENT1:SPCOUNT:STATE 1

### **:TELecomtest:MASK:ELEMent<x>:**

#### **SPCount:VALue?**

機能 各エレメントのエラーになったサンプルデータ数を問い合わせします。

構文 :TELecomtest:MASK:ELEMent<x>:SPCount:VALue?  
<x> = 1 ~ 4

例 :TELECOMTEST:MASK:ELEMENT1:SPCOUNT:VALUE? -> :TELECOMTEST:MASK:ELEMENT1:SPCOUNT:VALUE 1

### **:TELecomtest:MASK:ELEMent<x>:WCOunt?**

#### **(Wave Count)**

機能 各エレメントのエラーになったアクイジション回数に関する設定値を問い合わせます。

構文 :TELecomtest:MASK:ELEMent<x>:WCOunt?  
<x> = 1 ~ 4

例 :TELECOMTEST:MASK:ELEMENT1:WCOUNT?  
-> :TELECOMTEST:MASK:ELEMENT1:WCOUNT:STATE 1

### **:TELecomtest:MASK:ELEMent<x>:WCOunt:**

#### **STATE**

機能 各エレメントのエラーになったアクイジション回数の測定の ON/OFF を設定 / 問い合わせしません。

構文 :TELecomtest:MASK:ELEMent<x>:WCOunt:STATE {<Boolean>}  
:TELecomtest:MASK:ELEMent<x>:WCOunt:STATE?  
<x> = 1 ~ 4

例 :TELECOMTEST:MASK:ELEMENT1:WCOUNT:STATE ON  
:TELECOMTEST:MASK:ELEMENT1:WCOUNT:STATE? -> :TELECOMTEST:MASK:ELEMENT1:WCOUNT:STATE 1

### **:TELecomtest:MASK:ELEMent<x>:**

#### **WCOunt:VALue?**

機能 各エレメントのエラーになったアクイジション回数を問い合わせします。

構文 :TELecomtest:MASK:ELEMent<x>:WCOunt:VALue?  
<x> = 1 ~ 4

例 :TELECOMTEST:MASK:ELEMENT1:WCOUNT:VALUE? -> :TELECOMTEST:MASK:ELEMENT1:WCOUNT:VALUE 1

### **:TELecomtest:MMODE**

機能 マスク表示の ON/OFF を設定 / 問い合わせしません。

構文 :TELecomtest:MMODE {<Boolean>}  
:TELecomtest:MMODE?

例 :TELECOMTEST:MMODE ON  
:TELECOMTEST:MMODE? -> :TELECOMTEST:MMODE 1

**:TELecomtest:TRACe**

機能 テレコムテストの対象トレースを設定 / 問い合わせします。

構文 :TELecomtest:TRACe {<NRf>}  
:TELecomtest:TRACe?  
<NRf> = 1 ~ 8

例 :TELECOMTEST:TRACE 1  
:TELECOMTEST:TRACE? -> :TELECOMTEST:  
TRACE 1

**:TELecomtest:TRANge (Time Range)**

機能 テレコムテストの測定範囲を設定 / 問い合わせします。

構文 :TELecomtest:TRANge {<NRf>,<NRf>}  
:TELecomtest:TRANge?  
<NRf> = - 5 ~ 5(div)

例 :TELECOMTEST:TRANGE -5,0  
:TELECOMTEST:TRANGE? -> :TELECOMTEST:  
TRANGE 0.000E+00,-5.000E+00

**:TELecomtest:WINDow**

機能 テレコムテストの測定対象ウインドウを設定 / 問い合わせします。

構文 :TELecomtest:WINDow {MAIN|Z1|Z2}  
:TELecomtest:WINDow?

例 :TELECOMTEST:WINDOW MAIN  
:TELECOMTEST:WINDOW? -> :TELECOMTEST:  
WINDOW MAIN



## 5.29 TIMEbase グループ

### **:TIMEbase?**

機能 タイムベースに関するすべての設定値を問い合わせます。

構文 :TIMEbase?

例 :TIMEBASE? -> :TIMEBASE:TDIV

1.000E-06

### **:TIMEbase:SRATE? (Sample RATE)**

機能 サンプルレートを問い合わせます。

構文 :TIMEbase:SRATE?

例 :TIMEBASE:SRATE? -> :TIMEBASE:

SRATE 125.0E+06

### **:TIMEbase:TDIV**

機能 T/div 値を設定 / 問い合わせします。

構文 :TIMEbase:TDIV {<時間>}

:TIMEbase:TDIV?

<時間> = 500ps ~ 50s

例 :TIMEBASE:TDIV 1NS

:TIMEBASE:TDIV? -> :TIMEBASE:

TDIV 1.000E-06

## 5.30 TRIGger グループ

### :TRIGger?

機能 トリガに関するすべての設定値を問い合わせます。

構文 :TRIGger?

例 :TRIGGER? -> :TRIGGER:ACTION:ACQCOUNT 1;BUZZER 0;HCOPY 1;MAIL:INTERVAL OFF;MODE 0;:TRIGGER:ACTION:MODE ACONDITION;SAVE 1;:TRIGGER:TYPE EICYCLE;CLOCK:SOURCE 1;POLARITY RISE;:TRIGGER:DELAY:EDGECOUNT:COUNT 1;:TRIGGER:DELAY:MODE 1;POLARITY FALL;SOURCE 4;TIME 1.000E+00;TYPE EDGECOUNT;:TRIGGER:EINTERVAL:EVENT1:TYPE EDGE;CLOCK:SOURCE 1;POLARITY FALL;:TRIGGER:EINTERVAL:EVENT1:ESTATE:SOURCE 1;POLARITY FALL;:TRIGGER:EINTERVAL:EVENT1:CANBUS:ACK DONTCARE;BRATE 1000000;DATA:BORDER BIG;CONDITION DONTCARE;DATA1 0.0000000E+00;DATA2 255.000000E+00;DLC 8;MSBLSB 7,0;PATTERN "111001010110010001111000100110010101000100001000111111111010";SIGN UNSIGN;:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDEXT:PATTERN "XXXXXXXXXXXXXXXXXXXXXXXXX";:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:ACK DONTCARE;DATA:BORDER BIG;CONDITION DONTCARE;DATA1 0.0000000E+00;DATA2 255.000000E+00;DLC 8;MSBLSB 7,0;PATTERN "0000000100100011010001010110011110001001101011110011011110111";SIGN UNSIGN;:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:FORMAT STD;IDEXT:PATTERN "110101011110011011101110000";:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:IDSTD:PATTERN "00100100011";:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:MODE 0;RTR DATA;:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID2:ACK DONTCARE;DATA:BORDER BIG;CONDITION DONTCARE;DATA1 0.0000000E+00;DATA2 255.000000E+00;DLC 8;MSBLSB 7,0;PATTERN "1111111011011100101110101001100001110110010101000011001000010000";SIGN UNSIGN;:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID2:FORMAT STD;IDEXT:PATTERN "10010001101000101011001111000";:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID2:IDSTD:PATTERN "10001010110";:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID2:MODE 0

### :TRIGger:ACTion?

機能 アクションオントリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:ACTion?

例 :TRIGGER:ACTION? -> :TRIGGER:ACTION:ACQCOUNT 1;BUZZER 0;HCOPY 1;MAIL:INTERVAL OFF;MODE 0;:TRIGGER:ACTION:MODE ACONDITION;SAVE 1

### :TRIGger:ACTion:ACQCount

機能 アクションオントリガのアクション回数を設定 / 問い合わせします。

構文 :TRIGger:ACTion:ACQCount {<NRf>|INFinite}

:TRIGger:ACTion:ACQCount?<NRf> = 1 ~ 1000000

例 :TRIGGER:ACTION:ACQCOUNT 10  
:TRIGGER:ACTION:ACQCOUNT?  
-> :TRIGGER:ACTION:ACQCOUNT 10

### :TRIGger:ACTion:BUZZer

機能 アクション時に警告音を鳴らす (ON)/ 鳴らさない (OFF) を設定 / 問い合わせします。

構文 :TRIGger:ACTion:BUZZer {<Boolean>}

:TRIGger:ACTion:BUZZer?

例 :TRIGGER:ACTION:BUZZER ON  
:TRIGGER:ACTION:BUZZER? -> :TRIGGER:ACTION:BUZZER 1

### :TRIGger:ACTion:HCOPY

機能 アクション時に画面イメージデータを出力する (ON)/ しない (OFF) を設定 / 問い合わせします。

構文 :TRIGger:ACTion:HCOPY {<Boolean>}

:TRIGger:ACTion:HCOPY?

例 :TRIGGER:ACTION:HCOPY ON  
:TRIGGER:ACTION:HCOPY? -> :TRIGGER:ACTION:HCOPY 1

### :TRIGger:ACTion:MAIL?

機能 アクション時のメール送信に関するすべての設定値を問い合わせます。

構文 :TRIGger:ACTion:MAIL?

例 :TRIGGER:ACTION:MAIL? -> :TRIGGER:ACTION:MAIL:INTERVAL 10;MODE 1

## 5.30 TRIGger グループ

### **:TRIGger:ACTIon:MAIL:INTERval**

機能 アクション時にメール送信する間隔を設定 / 問い合わせします。

構文 :TRIGger:ACTIon:MAIL:INTERval  
{OFF|<Nrf>}  
:TRIGger:ACTIon:MAIL:INTERval?  
<Nrf>=1 ~ 1440(min)

例 :TRIGGER:ACTION:MAIL:INTERVAL 10  
:TRIGGER:ACTION:MAIL:INTERVAL?  
-> :TRIGGER:ACTION:MAIL:INTERVAL 10

### **:TRIGger:ACTIon:MAIL:MODE**

機能 アクション時にメール送信する (ON)/ しない (OFF) を設定 / 問い合わせします。

構文 :TRIGger:ACTIon:MAIL:MODE {<Boolean>}  
:TRIGger:ACTIon:MAIL:MODE?

例 :TRIGGER:ACTION:MAIL:MODE ON  
:TRIGGER:ACTION:MAIL:MODE?  
-> :TRIGGER:ACTION:MAIL:MODE 1

### **:TRIGger:ACTIon:MODE**

機能 アクションオントリガのモードを設定 / 問い合わせします。

構文 :TRIGger:ACTIon:MODE {ACONdition|OFF}  
:TRIGger:ACTIon:MODE?

例 :TRIGGER:ACTION:MODE ACONDITION  
:TRIGGER:ACTION:MODE? -> :TRIGGER:  
ACTION:MODE ACONDITION

### **:TRIGger:ACTIon:SAVE**

機能 アクション時に波形データをメディアに保存する (ON)/ しない (OFF) を設定 / 問い合わせします。

構文 :TRIGger:ACTIon:SAVE {<Boolean>}  
:TRIGger:ACTIon:SAVE?

例 :TRIGGER:ACTION:SAVE ON  
:TRIGGER:ACTION:SAVE? -> :TRIGGER:  
ACTION:SAVE 1

### **:TRIGger:ACTIon:START**

機能 アクションオントリガを開始します。

構文 :TRIGger:ACTIon:START

例 :TRIGGER:ACTION:START

### **:TRIGger:ACTIon:STOP**

機能 アクションオントリガを中止します。

構文 :TRIGger:ACTIon:STOP

例 :TRIGGER:ACTION:STOP

### **:TRIGger:CLOCK?**

機能 クロックチャンネルに関するすべての設定値を問い合わせします。

構文 :TRIGger:CLOCK?

例 :TRIGGER:CLOCK? -> :TRIGGER:CLOCK:  
SOURCE 1;POLARITY RISE

### **:TRIGger:CLOCK:POLarity**

機能 クロックチャンネルの極性を設定 / 問い合わせします。

構文 :TRIGger:CLOCK:POLarity {ENTER|EXIT|  
FALL|RISE}  
:TRIGger:CLOCK:POLarity?

例 :TRIGGER:CLOCK:POLARITY RISE  
:TRIGGER:CLOCK:POLARITY? -> :TRIGGER:  
CLOCK:POLARITY RISE

解説

- ・「:TRIGger:CLOCK:SOURCE NONE」のときは無効です。
- ・「:TRIGger:SOURCE:CHANNEL<x>:WINDOW ON」のときは {ENTER|EXIT}、それ以外のときは {FALL|RISE} が有効です。
- ・「:TRIGger:TYPE PSTATE|STATE」のときに有効です。

### **:TRIGger:CLOCK:SOURCE**

機能 クロックチャンネルの対象波形を設定 / 問い合わせします。

構文 :TRIGger:CLOCK:SOURCE {<Nrf>|NONE}  
:TRIGger:CLOCK:SOURCE?  
<NRF> = 1 ~ 4

例 :TRIGGER:CLOCK:SOURCE NONE  
:TRIGGER:CLOCK:SOURCE? -> :TRIGGER:  
CLOCK:SOURCE NONE

解説 「:TRIGger:TYPE PSTATE|STATE」のときに有効です。

### **:TRIGger:DElay?**

機能 トリガディレイに関するすべての設定値を問い合わせします。

構文 :TRIGger:DElay?

例 :TRIGGER:DELAY? -> :TRIGGER:DELAY:  
EDGEcount:COUNT 1;:TRIGGER:DELAY:  
MODE 1;POLARITY FALL;SOURCE 4;  
TIME 1.000E+00;TYPE EDGEcount

### **:TRIGger:DElay:EDGEcount?**

機能 トリガディレイのエッジカウントに関するすべての設定値を問い合わせます。

構文 :TRIGger:DElay:EDGEcount?

例 :TRIGGER:DELAY:EDGEcount?  
-> :TRIGGER:DELAY:EDGEcount:COUNT 1

### **:TRIGger:DElay:EDGEcount:COUNT**

機能 トリガディレイのエッジカウントの回数を設定 / 問い合わせします。

構文 :TRIGger:DElay:EDGEcount:COUNT  
{<Nrf>}  
:TRIGger:DElay:EDGEcount:COUNT?  
<NRF> = 1 ~ 1000000000

例 :TRIGGER:DELAY:EDGEcount:COUNT 1  
:TRIGGER:DELAY:EDGEcount:COUNT?  
-> :TRIGGER:DELAY:EDGEcount:COUNT 1

**:TRIGger:DElay:MODE**

機能 トリガディレイの ON/OFF を設定 / 問い合わせします。

構文 :TRIGger:DElay:MODE {<Boolean>}  
:TRIGger:DElay:MODE?

例 :TRIGGER:DELAY:MODE ON  
:TRIGGER:DELAY:MODE? -> :TRIGGER:  
DELAY:MODE 1

**:TRIGger:DElay:POLarity**

機能 トリガディレイのエッジ極性を設定 / 問い合わせします。

構文 :TRIGger:DElay:POLarity {FALL|RISE}  
:TRIGger:DElay:POLarity?

例 :TRIGGER:DELAY:POLARITY RISE  
:TRIGGER:DELAY:POLARITY? -> :TRIGGER:  
DELAY:POLARITY RISE

解説 「:TRIGger:DElay:TYPE EDGecount|  
FEADelay」のときに有効です。

**:TRIGger:DElay:SOURce**

機能 トリガディレイのエッジソースを設定 / 問い合わせします。

構文 :TRIGger:DElay:SOURce  
{<NRf>|EXternal}  
:TRIGger:DElay:SOURce?

例 :TRIGGER:DELAY:SOURCE 1  
:TRIGGER:DELAY:SOURCE? -> :TRIGGER:  
DELAY:SOURCE 1

解説 「:TRIGger:DElay:TYPE EDGecount|  
FEADelay」のときに有効です。

**:TRIGger:DElay:TIME**

機能 トリガディレイの遅延時間を設定 / 問い合わせします。

構文 :TRIGger:DElay:TIME {<時間>}  
:TRIGger:DElay:TIME?  
<時間> = 0s ~ 10s(5ps ステップ)

例 :TRIGGER:DELAY:TIME 1S  
:TRIGGER:DELAY:TIME? -> :TRIGGER:  
DELAY:TIME 1.000E+00

解説 「:TRIGger:DElay:TYPE BYTime|FEADelay」  
のときに有効です。

**:TRIGger:DElay:TYPE**

機能 トリガディレイの種類を設定 / 問い合わせします。

構文 :TRIGger:DElay:TYPE  
{BYTime|EDGecount|  
FEADelay}  
:TRIGger:DElay:TYPE?

例 :TRIGGER:DELAY:TYPE BYTIME  
:TRIGGER:DELAY:TYPE? -> :TRIGGER:  
DELAY:TYPE BYTIME

**:TRIGger:EINterval?**

機能 イベントインターバルに関するすべての設定値  
を問い合わせます。

構文 :TRIGger:EINterval?

例 :TRIGGER:EINTERVAL? -> :TRIGGER:  
EINTERVAL:EVENT1:TYPE EDGE;  
CLOCK:SOURCE 1;POLARITY FALL;;  
TRIGGER:EINTERVAL:EVENT1:ESTATE:  
SOURCE 1;POLARITY FALL;;TRIGGER:  
EINTERVAL:EVENT1:I2CBUS:ADATA:  
BIT10ADDRESS:PATTERN "10111011111";;  
TRIGGER:EINTERVAL:EVENT1:I2CBUS:  
ADATA:BIT7ADDRESS:  
PATTERN "11011110";:TRIGGER:  
EINTERVAL:EVENT1:I2CBUS:ADATA:  
BIT7APSUB:ADDRESS:P  
ATTERN "10101011";:TRIGGER:EINTERVAL:  
EVENT1:I2CBUS:ADATA:BIT7APSUB:  
SADDRESS:PATTERN "11001101";;  
TRIGGER:EINTERVAL:EVENT1:I2CBUS:  
ADATA:TYPE BIT10ADDRESS;;TRIGGER:  
EINTERVAL:EVENT1:I2CBUS:CLOCK:  
SOURCE 1;;TRIGGER:EINTERVAL:EVENT1:  
I2CBUS:DATA:BYTE 1;CONDITION TRUE;  
DPOSITION 1;MODE 1;  
PATTERN1 "10101011";  
PATTERN2 "10101010";  
PATTERN3 "10101111";  
PATTERN4 "10101011";PMODE DONTCARE;  
SOURCE 1;;TRIGGER:EINTERVAL:EVENT1:  
I2CBUS:GCALL:BIT7MADDRESS:  
PATTERN "1010101";:TRIGGER:EINTERVAL:  
EVENT1:I2CBUS:GCALL:  
SBYTE BIT7MADDRESS;;TRIGGER:  
EINTERVAL:EVENT1:I2CBUS:MODE ADATA;  
NAIGNORE:HSMODE 1;RACCESS 1;SBYTE  
1;;TRIGGER:EINTERVAL:EVENT1:I2CBUS:  
SBHSMODE:TYPE HSMODE;;TRIGGER:  
EINTERVAL:EVENT1:SPATTERN:  
BITRATE 1.000E+00;CLOCK:MODE 1;  
POLARITY FALL;SOURCE 1;;TRIGGER:  
EINTERVAL:EVENT1:SPATTERN:CS 1;DATA:  
ACTIVE HIGH;SOURCE 1;;TRIGGER:  
EINTERVAL:EVENT1:SPATTERN:LATCH:  
SOURCE 1;POLARITY FALL;;TRIGGER:  
EINTERVAL:EVENT1:SPATTERN:  
PATTERN "1100110111101111".....

### 5.30 TRIGger グループ

**:TRIGger:EIInterval:EVENT<x>?**

機能 各イベントに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1?  
-> :TRIGGER:EIINTERVAL:EVENT1:  
TYPE EDGE;CLOCK:SOURCE 1;  
POLARITY FALL;:TRIGGER:EIINTERVAL:  
EVENT1:ESTATE:SOURCE 1;  
POLARITY FALL;:TRIGGER:EIINTERVAL:  
EVENT1:I2CBUS:ADATA:BIT10ADDRESS:  
PATTERN "10111011111";:TRIGGER:  
EIINTERVAL:EVENT1:I2CBUS:ADATA:  
BIT7ADDRESS:PATTERN "11011110";:  
TRIGGER:EIINTERVAL:EVENT1:I2CBUS:  
ADATA:BIT7APSUB:ADDRESS:  
PATTERN "10101011";:TRIGGER:  
EIINTERVAL:EVENT1:I2CBUS:ADATA:  
BIT7APSUB:SADDRESS:  
PATTERN "11001101";:TRIGGER:  
EIINTERVAL:EVENT1:I2CBUS:ADATA:  
TYPE BIT10ADDRESS;:TRIGGER:EIINTERVAL:  
EVENT1:I2CBUS:CLOCK:SOURCE 1;:  
TRIGGER:EIINTERVAL:EVENT1:I2CBUS:DATA:  
BYTE 1;CONDITION TRUE;DPOSITION 1;  
MODE 1;PATTERN1 "10101011";  
PATTERN2 "10101010";  
PATTERN3 "10101111";  
PATTERN4 "10101011";PMODE DONTCARE;  
SOURCE 1;:TRIGGER:EIINTERVAL:EVENT1:  
I2CBUS:GCALL:BIT7MADDRESS:  
PATTERN "1010101";:TRIGGER:EIINTERVAL:  
EVENT1:I2CBUS:GCALL:  
SBYTE BIT7MADDRESS;:TRIGGER:  
EIINTERVAL:EVENT1:I2CBUS:MODE ADATA;  
NAIGNORE:HSMODE 1;RACCESS 1;SBYTE 1;:  
TRIGGER:EIINTERVAL:EVENT1:I2CBUS:  
SBHSMODE:TYPE HSMODE;:TRIGGER:  
EIINTERVAL:EVENT1:SPATTERN:  
BITRATE 1.000E+00;CLOCK:MODE 1;  
POLARITY FALL;SOURCE 1;:TRIGGER:  
EIINTERVAL:EVENT1:SPATTERN:CS 1;  
DATA:ACTIVE HIGH;SOURCE 1;:TRIGGER:  
EIINTERVAL:EVENT1:SPATTERN:LATCH:  
SOURCE 1;POLARITY FALL;:TRIGGER:  
EIINTERVAL:EVENT1:SPATTERN:  
PATTERN "1100110111101111".....  
EVENT2 は、「:TRIGger:TYPE EIDelay|  
EISequence」のときに有効です。

解説

**:TRIGger:EIInterval:EVENT<x>:CANBus?**

機能 各イベントの CAN バス信号トリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:CANBus?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:CANBUS?  
-> :TRIGGER:EIINTERVAL:EVENT1:CANBUS:  
ACK DONTCARE;BRATE 1000000;DATA:  
BORDER BIG;CONDITION DONTCARE;  
DATA1 0.0000000E+00;  
DATA2 255.000000E+00;  
DLC 8;MSBLSB 7,0;PATTERN "111001010  
11001000111100010010011001010100010  
001000111111111010";SIGN UNSIGN;:  
TRIGGER:EIINTERVAL:EVENT1:CANBUS:  
IDEXT:PATTERN "XXXXXXXXXXXXXXXXXXXX  
XXXXXXXX";:TRIGGER:EIINTERVAL:EVENT1:  
CANBUS:IDOR:ID1:ACK DONTCARE;DATA:  
BORDER BIG;CONDITION DONTCARE;  
DATA1 0.0000000E+00;  
DATA2 255.000000E+00;DLC 8;MSBLSB 7,0;  
PATTERN "000000010010001101000101011  
0011110001001101010111100110111011  
11";SIGN UNSIGN;:TRIGGER:EIINTERVAL:  
EVENT1:CANBUS:IDOR:ID1:FORMAT STD;  
IDEXT:PATTERN "110101011110011011110  
11110000";:TRIGGER:EIINTERVAL:EVENT1:  
CANBUS:IDOR:ID1:IDSTD:  
PATTERN "00100100011";:TRIGGER:  
EIINTERVAL:EVENT1:CANBUS:IDOR:ID1:  
MODE 0;RTR DATA;:TRIGGER:EIINTERVAL:  
EVENT1:CANBUS:IDOR:ID2:ACK DONTCARE;  
DATA:BORDER BIG;CONDITION DONTCARE;  
DATA1 0.0000000E+00;  
DATA2 255.000000E+00;DLC 8;MSBLSB 7,0;  
PATTERN "111111101101110010111010100  
11000011101100101010000110010000100  
00";SIGN UNSIGN;:TRIGGER:EIINTERVAL:  
EVENT1:CANBUS:IDOR:ID2:FORMAT STD;  
IDEXT:PATTERN "100100011010001010110  
01111000";:TRIGGER:EIINTERVAL:EVENT1:  
CANBUS:IDOR:ID2:IDSTD:  
PATTERN "10001010110";:TRIGGER:  
EIINTERVAL:EVENT1:CANBUS:IDOR:ID2:  
MODE 0.....

**:TRIGger:EINterval:EVENT<x>:CANBus:****ACK**

機能 CANバス信号トリガのACK条件を設定/問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
ACK {ACK|ACKBoth|DONTcare|NONack}  
:TRIGger:EINterval:EVENT<x>:CANBus:  
ACK?

<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:  
ACK ACK  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
ACK? -> :TRIGGER:EINTERVAL:EVENT1:  
CANBUS:ACK ACK

**:TRIGger:EINterval:EVENT<x>:CANBus:  
BRATe**

機能 CANバス信号トリガのビットレート(データ転送速度)を設定/問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
BRATe {<Nrf>|USER,<Nrf>}  
:TRIGger:EINterval:EVENT<x>:CANBus:  
BRATe?

<x> = 1, 2

<Nrf> = 33300, 83300, 125000, 250000,  
500000,  
1000000

USERの<Nrf> = 本体ユーザーズマニュアル参照。

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:  
BRATE 83300  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
BRATE? -> :TRIGGER:EINTERVAL:EVENT1:  
CANBUS:BRATE 83300

**:TRIGger:EINterval:EVENT<x>:CANBus:  
DATA?**

機能 CANバス信号トリガのデータに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
DATA?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:  
DATA? -> :TRIGGER:EINTERVAL:EVENT1:  
CANBUS:DATA:BORDER BIG;  
CONDITION DONTCARE;  
DATA1 0.0000000E+00;  
DATA2 255.00000E+00;DLC 8;MSBSLB 7,0;  
PATTERN "1110010101100100011110001001  
00110010101000100001000111111111010"  
;SIGN UNSIGN

**:TRIGger:EINterval:EVENT<x>:CANBus:****DATA:BORDER**

機能 CANバス信号トリガのデータのバイトオーダを設定/問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
DATA:BORDER {BIG|LITTLE}  
:TRIGger:EINterval:EVENT<x>:CANBus:  
DATA:BORDER?

<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:  
DATA:BORDER BIG;TRIGGER:EINTERVAL:  
EVENT1:CANBUS:DATA:BORDER?  
-> :TRIGGER:EINTERVAL:EVENT1:CANBUS:  
DATA:BORDER BIG

**:TRIGger:EINterval:EVENT<x>:CANBus:  
DATA:CONDition**

機能 CANバス信号トリガのデータ条件を設定/問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
DATA:CONDition {BETween|DONTcare|  
FALSE|GTHan|LTHan|ORANGE|TRUE}  
:TRIGger:EINterval:EVENT<x>:CANBus:  
DATA:CONDition?

<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:  
DATA:CONDITION BETWEEN  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
DATA:CONDITION? -> :TRIGGER:  
EINTERVAL:EVENT1:CANBUS:DATA:  
CONDITION BETWEEN

### 5.30 TRIGger グループ

#### **:TRIGger:EINterval:EVENT<x>:CANBus:DATA:DATA<x>**

機能 CAN バス信号トリガのデータの比較データを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:DATA:DATA<x> {<Nrf>}  
:TRIGger:EINterval:EVENT<x>:CANBus:DATA:DATA<x>?

EVENT<x> の <x> = 1, 2  
DATA<x> の <x> = 1, 2  
<Nrf> = 本体ユーザーズマニュアル参照。

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:DATA1  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:DATA1? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:DATA1 1.0000000E+00

解説

- 「:TRIGger:EINterval:EVENT<x>:CANBus:DATA:CONDition GTHan」のときは「:TRIGger:EINterval:EVENT<x>:CANBus:DATA:DATA1」で設定します。
- 「:TRIGger:EINterval:EVENT<x>:CANBus:DATA:CONDition LTHan」のときは「:TRIGger:EINterval:EVENT<x>:CANBus:DATA:DATA2」で設定します。
- 「:TRIGger:EINterval:EVENT<x>:CANBus:DATA:CONDition BETWEEN|ORANge」のときは、小さい値を「:TRIGger:EINterval:EVENT<x>:CANBus:DATA:DATA1」、大きい値を「:TRIGger:EINterval:EVENT<x>:CANBus:DATA:DATA2」で設定します。

#### **:TRIGger:EINterval:EVENT<x>:CANBus:DATA:DLC**

機能 CAN バス信号トリガのデータの有効バイト数 (DLC) を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:DATA:DLC {<Nrf>}  
:TRIGger:EINterval:EVENT<x>:CANBus:DATA:DLC?

<x> = 1, 2  
<Nrf> = 0 ~ 8

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:DLC 0 :TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:DLC? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:DLC 0

#### **:TRIGger:EINterval:EVENT<x>:CANBus:DATA:HEXA**

機能 CAN バス信号トリガのデータを HEXA で設定します。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:DATA:HEXA {<文字列>}

<x> = 1, 2  
<文字列> = '0' ~ 'F', 'X' の組み合わせ 16 文字以内 (1 バイト単位)

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:HEXA "A9"

#### **:TRIGger:EINterval:EVENT<x>:CANBus:DATA:MSBLsb**

機能 CAN バス信号トリガのデータの MSB/LSB のビットを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:DATA:MSBLsb {<Nrf>,<Nrf>}  
:TRIGger:EINterval:EVENT<x>:CANBus:DATA:MSBLsb?

<x> = 1, 2  
<Nrf> = 本体ユーザーズマニュアル参照。

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:MSBLSB 1,0  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:MSBLSB? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:MSBLSB 1,0

#### **:TRIGger:EINterval:EVENT<x>:CANBus:DATA:PATtern**

機能 CAN バス信号トリガのデータを BINARY で設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:DATA:PATtern {<文字列>}  
:TRIGger:EINterval:EVENT<x>:CANBus:DATA:PATtern?

<x> = 1, 2  
<文字列> = '0', '1', 'X' の組み合わせ 64 文字以内 (1 バイト単ハ)

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:PATTERN "11011111"  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:PATTERN? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:PATTERN "11011111"

**:TRIGger:EINterval:EVENT<x>:CANBus:DATA:SIGN**

機能 CAN バス信号トリガのデータの符号を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:DATA:SIGN {SIGN|UNSign}  
:TRIGger:EINterval:EVENT<x>:CANBus:DATA:SIGN?  
<x> = 1、2

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:SIGN SIGN  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:SIGN? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:SIGN SIGN

**:TRIGger:EINterval:EVENT<x>:CANBus:IDEXT?**

機能 CAN バス信号トリガの拡張フォーマットの ID に関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:IDEXT?  
<x> = 1、2

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDEXT? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDEXT:PATTERN "110010110111000011101110111111"

**:TRIGger:EINterval:EVENT<x>:CANBus:IDEXT:HEXA**

機能 CAN バス信号トリガの拡張フォーマットの ID を HEXA で設定します。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:IDEXT:HEXA {<文字列>}  
<x> = 1、2  
<文字列> = '0' ~ 'F'、'X' の組み合わせ 8 文字

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDEXT:HEXA "1AEF5906"

**:TRIGger:EINterval:EVENT<x>:CANBus:IDEXT:PATtern**

機能 CAN バス信号トリガの拡張フォーマットの ID を BINARY で設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:IDEXT:PATtern {<文字列>}  
:TRIGger:EINterval:EVENT<x>:CANBus:IDEXT:PATtern?  
<x> = 1、2  
<文字列> = '0'、'1'、'X' の組み合わせ 29 文字

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDEXT:PATTERN "1100101101110000111011011111"  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDEXT:PATTERN? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDEXT:PATTERN "11001011011100001110111011111"

**:TRIGger:EINterval:EVENT<x>:CANBus:IDOR?**

機能 CAN バス信号トリガの OR 条件に関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:IDOR?  
<x> = 1、2

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:ACK DONTCARE;DATA:BORDER BIG;CONDITION DONTCARE;DATA1 0.000000E+00;DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;PATTERN "0000001001000110100010101100111000100110101011110011011101111";SIGN UNSIGN;;TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:FORMAT STD;IDEXT:PATTERN "1101010111100110111101110000";:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:IDSTD:PATTERN "00100100011";:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:MODE 0;RTR DATA;;TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID2:ACK DONTCARE;DATA:BORDER BIG;CONDITION DONTCARE;DATA1 0.000000E+00;DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;PATTERN "11111101101100101110101001100001101100101010000110010000100000";SIGN UNSIGN;;TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID2:FORMAT STD;IDEXT:PATTERN"100100011010001010110011110000";:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID2:IDSTD:PATTERN "10001010110";:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID2:MODE 0;RTR DATA;;TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID3:ACK DONTCARE;DATA:BORDER BIG;CONDITION DONTCARE;DATA1 0.000000E+00;DATA2 255.00000E+00;DLC 8.....



### 5.30 TRIGger グループ

#### **:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>?**

機能 CAN バス信号トリガの OR 条件の各 ID に関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>?

EVENT<x> の <x> = 1、2

ID<x> の <x> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:ACK DONTCARE; DATA:BORDER BIG;CONDITION DONTCARE; DATA1 0.0000000E+00; DATA2 255.00000E+00;DLC 8;MSBLSB 7,0; PATTERN "00000001001000110100010101100111000100110110111001101110011011101111";SIGN UNSIGN;;TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:FORMAT STD; IDEXT:PATTERN "110101011110011011101110000";:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:IDSTD: PATTERN "00100100011";:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:MODE 0;RTR DATA

#### **:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:ACK**

機能 CAN バス信号トリガの OR 条件の各 ACK 条件を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:ACK {ACK|ACKBoth|DONTcare|NONack}

:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:ACK?

EVENT<x> の <x> = 1、2

ID<x> の <x> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:ACK ACK :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:ACK? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:ACK ACK

#### **:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA?**

機能 CAN バス信号トリガの OR 条件の各データに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA?

EVENT<x> の <x> = 1、2

ID<x> の <x> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:DATA? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:DATA:BORDER BIG;CONDITION DONTCARE; DATA1 0.0000000E+00; DATA2 255.00000E+00;DLC 8;MSBLSB 7,0; PATTERN "0000000100100011010001010110011100010011010101110011011101111";SIGN UNSIGN

#### **:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:BORDER**

機能 CAN バス信号トリガの OR 条件の各データのバイトオーダを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:BORDER {BIG|LITTLE}

:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:BORDER?

EVENT<x> の <x> = 1、2

ID<x> の <x> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:DATA:BORDER BIG :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:DATA:BORDER? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:DATA:BORDER BIG

#### **:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:CONDition**

機能 CAN バス信号トリガの OR 条件の各データ条件を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:CONDition {BETween|DONTcare|FALSe|GTHan|LTHan|ORANge|TRUE}

:TRIGger:EINterval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:CONDition?

EVENT<x> の <x> = 1、2

ID<x> の <x> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:DATA:CONDITION BETWEEN :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:DATA:CONDITION? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:DATA:CONDITION BETWEEN

**:TRIGGER:EVENT<x>:CANBUS:  
IDOR:ID<x>:DATA:DATA<x>**

機能 CANバス信号トリガのOR条件の各データの比較データを設定/問い合わせします。

構文 :TRIGGER:EVENT<x>:CANBUS:  
IDOR:ID<x>:DATA:DATA<x> {<Nrf>}  
:TRIGGER:EVENT<x>:CANBUS:  
IDOR:ID<x>:DATA:DATA<x>?  
EVENT<x>の<x>=1、2  
ID<x>の<x>=1~4  
DATA<x>の<x>=1、2  
<Nrf>=本体ユーザーズマニュアル参照。

例 :TRIGGER:EVENT:EVENT1:CANBUS:  
IDOR:ID1:DATA:DATA1 1  
:TRIGGER:EVENT:EVENT1:CANBUS:  
IDOR:ID1:DATA:DATA1? -> :TRIGGER:  
EVENT:EVENT1:CANBUS:IDOR:ID1:  
DATA:DATA1 1.000000E+00

解説

- 「:TRIGGER:EVENT<x>:  
CANBUS:IDOR:ID<x>:DATA:CONDITION  
GTHan」のときは「:TRIGGER:EVENT<x>:  
CANBUS:IDOR:ID<x>:DATA:  
DATA1」で設定します。
- 「:TRIGGER:EVENT<x>:  
CANBUS:IDOR:ID<x>:DATA:CONDITION  
LTHan」のときは「:TRIGGER:EVENT<x>:  
CANBUS:IDOR:ID<x>:DATA:  
DATA2」で設定します。
- 「:TRIGGER:EVENT<x>:  
CANBUS:IDOR:ID<x>:DATA:CONDITION  
BETWEEN|ORANGE」のときは、小さい値を  
「:TRIGGER:EVENT<x>:  
CANBUS:IDOR:ID<x>:DATA:DATA1」、大きい  
値を「:TRIGGER:EVENT<x>:  
CANBUS:IDOR:ID<x>:DATA:DATA2」で設定  
します。

**:TRIGGER:EVENT<x>:CANBUS:  
IDOR:ID<x>:DATA:DLC**

機能 CANバス信号トリガのOR条件の各データの有効バイト数(DLC)を設定/問い合わせします。

構文 :TRIGGER:EVENT<x>:CANBUS:  
IDOR:ID<x>:DATA:DLC {<Nrf>}  
:TRIGGER:EVENT<x>:CANBUS:  
IDOR:ID<x>:DATA:DLC?  
EVENT<x>の<x>=1、2  
ID<x>の<x>=1~4  
<Nrf>=0~8

例 :TRIGGER:EVENT:EVENT1:CANBUS:  
IDOR:ID1:DATA:DLC 0  
:TRIGGER:EVENT:EVENT1:CANBUS:  
IDOR:ID1:DATA:DLC? -> :TRIGGER:  
EVENT:EVENT1:CANBUS:IDOR:ID1:  
DATA:DLC 0

**:TRIGGER:EVENT<x>:CANBUS:  
IDOR:ID<x>:DATA:HEXA**

機能 CANバス信号トリガのOR条件の各データをHEXAで設定します。

構文 :TRIGGER:EVENT<x>:CANBUS:  
IDOR:ID<x>:DATA:HEXA {<文字列>}  
EVENT<x>の<x>=1、2  
ID<x>の<x>=1~4  
<文字列>='0'~'F','X'の組み合わせ16文字  
以内(1バイト単位)

例 :TRIGGER:EVENT:EVENT1:CANBUS:  
IDOR:ID1:DATA:HEXA "A9"

**:TRIGGER:EVENT<x>:CANBUS:  
IDOR:ID<x>:DATA:MSBLSb**

機能 CANバス信号トリガのOR条件の各データのMSB/LSBのビットを設定/問い合わせします。

構文 :TRIGGER:EVENT<x>:CANBUS:  
IDOR:ID<x>:DATA:MSBLSb {<Nrf>,<Nrf>}  
:TRIGGER:EVENT<x>:CANBUS:  
IDOR:ID<x>:DATA:MSBLSb?  
EVENT<x>の<x>=1、2  
ID<x>の<x>=1~4  
<Nrf>=本体ユーザーズマニュアル参照。

例 :TRIGGER:EVENT:EVENT1:CANBUS:  
IDOR:ID1:DATA:MSBLSB 1,0  
:TRIGGER:EVENT:EVENT1:CANBUS:  
IDOR:ID1:DATA:MSBLSB? -> :TRIGGER:  
EVENT:EVENT1:CANBUS:IDOR:ID1:  
DATA:MSBLSB 1,0

**:TRIGGER:EVENT<x>:CANBUS:  
IDOR:ID<x>:DATA:PATTERN**

機能 CANバス信号トリガのOR条件の各データをBINARYで設定/問い合わせします。

構文 :TRIGGER:EVENT<x>:CANBUS:  
IDOR:ID<x>:DATA:PATTERN {<文字列>}  
:TRIGGER:EVENT<x>:CANBUS:  
IDOR:ID<x>:DATA:PATTERN?  
EVENT<x>の<x>=1、2  
ID<x>の<x>=1~4  
<文字列>='0','1','X'の組み合わせ64文字以  
内(1バイト単位)

例 :TRIGGER:EVENT:EVENT1:CANBUS:  
IDOR:ID1:DATA:PATTERN "11011111"  
:TRIGGER:EVENT:EVENT1:CANBUS:  
IDOR:ID1:DATA:PATTERN? -> :TRIGGER:  
EVENT:EVENT1:CANBUS:IDOR:ID1:  
DATA:PATTERN "11011111"

### 5.30 TRIGger グループ

#### **:TRIGger:EINTerval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:SIGN**

機能 CAN バス信号トリガの OR 条件の各データの符号を設定 / 問い合わせします。

構文 :TRIGger:EINTerval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:SIGN {SIGN|UNSign}  
:TRIGger:EINTerval:EVENT<x>:CANBus:IDOR:ID<x>:DATA:SIGN?  
EVENT<x> の <x> = 1, 2  
ID<x> の <x> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:DATA:SIGN SIGN  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:DATA:SIGN? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:DATA:SIGN SIGN

#### **:TRIGger:EINTerval:EVENT<x>:CANBus:IDOR:ID<x>:FORMat**

機能 CAN バス信号トリガの OR 条件の各メッセージフォーマット (標準 / 拡張) を設定 / 問い合わせします。

構文 :TRIGger:EINTerval:EVENT<x>:CANBus:IDOR:ID<x>:FORMat {STD|EXT}  
:TRIGger:EINTerval:EVENT<x>:CANBus:IDOR:ID<x>:FORMat?  
EVENT<x> の <x> = 1, 2  
ID<x> の <x> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:FORMAT STD  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:FORMAT? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:FORMAT STD

#### **:TRIGger:EINTerval:EVENT<x>:CANBus:IDOR:ID<x>:IDExt?**

機能 CAN バス信号トリガの OR 条件の各拡張フォーマットの ID に関するすべての設定値を問い合わせます。

構文 :TRIGger:EINTerval:EVENT<x>:CANBus:IDOR:ID<x>:IDExt?  
EVENT<x> の <x> = 1, 2  
ID<x> の <x> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:IDEXT? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:IDEXT:PATTERN "11001011011100001110111011111"

#### **:TRIGger:EINTerval:EVENT<x>:CANBus:IDOR:ID<x>:IDExt:HEXA**

機能 CAN バス信号トリガの OR 条件の各拡張フォーマットの ID を HEXA で設定します。

構文 :TRIGger:EINTerval:EVENT<x>:CANBus:IDOR:ID<x>:IDExt:HEXA {<文字列>}  
EVENT<x> の <x> = 1, 2  
ID<x> の <x> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:IDEXT:HEXA "1AEF5906"

#### **:TRIGger:EINTerval:EVENT<x>:CANBus:IDOR:ID<x>:IDExt:PATtern**

機能 CAN バス信号トリガの OR 条件の各拡張フォーマットの ID を BINARY で設定 / 問い合わせします。

構文 :TRIGger:EINTerval:EVENT<x>:CANBus:IDOR:ID<x>:IDExt:PATtern {<文字列>}  
:TRIGger:EINTerval:EVENT<x>:CANBus:IDOR:ID<x>:IDExt:PATtern?  
EVENT<x> の <x> = 1, 2  
ID<x> の <x> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:IDEXT:PATTERN "110010110111000011101110111111"  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:IDEXT:PATTERN? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:IDEXT:PATTERN "11001011011100001110111011111"

#### **:TRIGger:EINTerval:EVENT<x>:CANBus:IDOR:ID<x>:IDStd?**

機能 CAN バス信号トリガの OR 条件の各標準フォーマットの ID に関するすべての設定値を問い合わせます。

構文 :TRIGger:EINTerval:EVENT<x>:CANBus:IDOR:ID<x>:IDStd?  
EVENT<x> の <x> = 1, 2  
ID<x> の <x> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:IDSTD? -> :TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:ID1:IDSTD:PATTERN "00011111101"

**:TRIGger:EINterval:EVENT<x>:CANBus:  
IDOR:ID<x>:IDSTd:HEXA**

機能 CAN バス信号トリガの OR 条件の各標準フォーマットの ID を HEXA で設定します。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
IDOR:ID<x>:IDSTd:HEXA {<文字列>}  
EVENT<x> の <x> = 1, 2  
ID<x> の <x> = 1 ~ 4

例 <文字列> = '0' ~ 'F', 'X' の組み合わせ 3 文字  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
IDOR:ID1:IDSTD:HEXA "5DF"

**:TRIGger:EINterval:EVENT<x>:CANBus:  
IDOR:ID<x>:IDSTd:PATtern**

機能 CAN バス信号トリガの OR 条件の各標準フォーマットの ID を BINARY で設定/問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
IDOR:ID<x>:IDSTd:PATtern {<文字列>}  
:TRIGger:EINterval:EVENT<x>:CANBus:  
IDOR:ID<x>:IDSTd:PATtern?  
EVENT<x> の <x> = 1, 2  
ID<x> の <x> = 1 ~ 4

例 <文字列> = '0', '1', 'X' の組み合わせ 11 文字  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
IDOR:ID1:IDSTD:PATTERN "10111011111"  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
IDOR:ID1:IDSTD:PATTERN? -> :TRIGGER:  
EINTERVAL:EVENT1:CANBUS:IDOR:ID1:  
IDSTD:PATTERN "10111011111"

**:TRIGger:EINterval:EVENT<x>:CANBus:  
IDOR:ID<x>:MODE**

機能 CAN バス信号トリガの OR 条件の各条件の有効 (1)/無効 (0) を設定/問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
IDOR:ID<x>:MODE {<Boolean>}  
:TRIGger:EINterval:EVENT<x>:CANBus:  
IDOR:ID<x>:MODE?  
EVENT<x> の <x> = 1, 2  
ID<x> の <x> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:  
IDOR:ID1:MODE ON  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
IDOR:ID1:MODE? -> :TRIGGER:EINTERVAL:  
EVENT1:CANBUS:IDOR:ID1:MODE 1

**:TRIGger:EINterval:EVENT<x>:CANBus:  
IDOR:ID<x>:RTR**

機能 CAN バス信号トリガの OR 条件の各 RTR を設定/問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
IDOR:ID<x>:RTR {DATA|DONTcare|REMOte}  
:TRIGger:EINterval:EVENT<x>:CANBus:  
IDOR:ID<x>:RTR?

例 EVENT<x> の <x> = 1, 2  
ID<x> の <x> = 1 ~ 4  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
IDOR:ID1:RTR DATA  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
IDOR:ID1:RTR? -> :TRIGGER:EINTERVAL:  
EVENT1:CANBUS:IDOR:ID1:RTR DATA

**:TRIGger:EINterval:EVENT<x>:CANBus:  
IDSTd?**

機能 CAN バス信号トリガの標準フォーマットの ID に関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
IDSTd?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:  
IDSTD? -> :TRIGGER:EINTERVAL:EVENT1:  
CANBUS:IDSTD:PATTERN "00011111101"

**:TRIGger:EINterval:EVENT<x>:CANBus:  
IDSTd:HEXA**

機能 CAN バス信号トリガの標準フォーマットの ID を HEXA で設定します。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
IDSTd:HEXA {<文字列>}  
<x> = 1, 2  
<文字列> = '0' ~ 'F', 'X' の組み合わせ 3 文字

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:  
IDSTD:HEXA "5DF"

**:TRIGger:EINterval:EVENT<x>:CANBus:  
IDSTd:PATtern**

機能 CAN バス信号トリガの標準フォーマットの ID を BINARY で設定/問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
IDSTd:PATtern {<文字列>}  
:TRIGger:EINterval:EVENT<x>:CANBus:  
IDSTd:PATtern?  
<x> = 1, 2

例 <文字列> = '0', '1', 'X' の組み合わせ 11 文字  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
IDSTD:PATTERN "10111011111"  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
IDSTD:PATTERN? -> :TRIGGER:EINTERVAL:  
EVENT1:CANBUS:IDSTD:PATTERN  
"10111011111"

### 5.30 TRIGger グループ

#### **:TRIGger:EINterval:EVENT<x>:CANBus:**

##### **MODE**

機能 CAN バス信号トリガのモードを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
MODE {EFrame|IDExt|IDOR|IDStd|SOF}  
:TRIGger:EINterval:EVENT<x>:CANBus:  
MODE?

<x> = 1、2

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:  
MODE EFRAME  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
MODE? -> :TRIGGER:EINTERVAL:EVENT1:  
CANBUS:MODE EFRAME

#### **:TRIGger:EINterval:EVENT<x>:CANBus:**

##### **REcessive**

機能 CAN バス信号トリガのリセッスレベル(バスレベル)を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
REcessive {HIGH|LOW}  
:TRIGger:EINterval:EVENT<x>:CANBus:  
REcessive?

<x> = 1、2

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:  
RECESSIVE HIGH  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
RECESSIVE? -> :TRIGGER:EINTERVAL:  
EVENT1:CANBUS:RECESSIVE HIGH

#### **:TRIGger:EINterval:EVENT<x>:CANBus:**

##### **RTR**

機能 CAN バス信号トリガの RTR を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
RTR {DATA|DONTcare|REmote}  
:TRIGger:EINterval:EVENT<x>:CANBus:  
RTR?

<x> = 1、2

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:  
RTR DATA  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
RTR? -> :TRIGGER:EINTERVAL:EVENT1:  
CANBUS:RTR DATA

#### **:TRIGger:EINterval:EVENT<x>:CANBus:**

##### **SOURCE**

機能 CAN バス信号トリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
SOURCE {<Nrf>}  
:TRIGger:EINterval:EVENT<x>:CANBus:  
SOURCE?

<x> = 1、2

<NRF> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:  
SOURCE 1  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:  
CANBUS:SOURCE 1

#### **:TRIGger:EINterval:EVENT<x>:CANBus:**

##### **SPOint**

機能 CAN バス信号トリガのサンプルポイントを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CANBus:  
SPOint {<Nrf>}  
:TRIGger:EINterval:EVENT<x>:CANBus:  
SPOint?

<x> = 1、2

<NRF> = 18.8 ~ 90.6(%)

例 :TRIGGER:EINTERVAL:EVENT1:CANBUS:  
SPOINT 18.8  
:TRIGGER:EINTERVAL:EVENT1:CANBUS:  
SPOINT? -> :TRIGGER:EINTERVAL:EVENT1:  
CANBUS:SPOINT 18.8E+00

#### **:TRIGger:EINterval:EVENT<x>:CLOCK?**

機能 各イベントのクロックチャンネルに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:CLOCK?  
<x> = 1、2

例 :TRIGGER:EINTERVAL:EVENT1:CLOCK?  
-> :TRIGGER:EINTERVAL:EVENT1:CLOCK:  
SOURCE 1;POLARITY FALL

**:TRIGger:EINterval:EVENT<x>:CLOCK:POLarity**

機能 各イベントのクロックチャンネルの極性を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CLOCK:POLarity {ENTER|EXIT|FALL|RISE}  
:TRIGger:EINterval:EVENT<x>:CLOCK:POLarity?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:CLOCK:POLARITY FALL  
:TRIGGER:EINTERVAL:EVENT1:CLOCK:POLARITY? -> :TRIGGER:EINTERVAL:EVENT1:CLOCK:POLARITY FALL

解説

- ・「:TRIGger:EINterval:EVENT<x>:CLOCK:SOURCE NONE」のときは無効です。
- ・「:TRIGger:SOURCE:CHANNEL<x>:WINDOW ON」のときは {ENTER|EXIT}、それ以外のときは {FALL|RISE} が有効です。
- ・「:TRIGger:EINterval:EVENT<x>:TYPE PSTATE|STATE」のときに有効です。

**:TRIGger:EINterval:EVENT<x>:CLOCK:SOURCE**

機能 各イベントのクロックチャンネルの対象波形を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:CLOCK:SOURCE {<Nrf>|NONE}  
:TRIGger:EINterval:EVENT<x>:CLOCK:SOURCE?  
<x> = 1, 2  
<Nrf> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:CLOCK:SOURCE 1  
:TRIGGER:EINTERVAL:EVENT1:CLOCK:SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:CLOCK:SOURCE 1

解説 「:TRIGger:EINterval:EVENT<x>:TYPE PSTATE|STATE」のときに有効です。

**:TRIGger:EINterval:EVENT<x>:ESTate?**

機能 Edge/State トリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:ESTate?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:ESTATE?  
-> :TRIGGER:EINTERVAL:EVENT1:ESTATE:SOURCE 1;POLARITY FALL

**:TRIGger:EINterval:EVENT<x>:ESTate:POLarity**

機能 Edge/State トリガの極性を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:ESTate:POLarity {ENTER|EXIT|FALL|RISE}  
:TRIGger:EINterval:EVENT<x>:ESTate:POLarity?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:ESTATE:POLARITY ENTER  
:TRIGGER:EINTERVAL:EVENT1:ESTATE:POLARITY? -> :TRIGGER:EINTERVAL:EVENT1:ESTATE:POLARITY ENTER

解説

- ・「:TRIGger:EINterval:EVENT<x>:TYPE EDGE」且つ「:TRIGger:EINterval:EVENT<x>:ESTate:SOURCE LINE」のときは無効です。
- ・「:TRIGger:EINterval:EVENT<x>:TYPE EDGE|EQualify」且つ「:TRIGger:SOURCE:CHANNEL<x>:WINDOW ON」のときは {ENTER|EXIT}、「:TRIGger:SOURCE:CHANNEL<x>:WINDOW OFF」のときは {FALL|RISE} が有効です。
- ・「:TRIGger:EINterval:EVENT<x>:TYPE STATE」のときは {ENTER|EXIT} が有効です。

**:TRIGger:EINterval:EVENT<x>:ESTate:SOURCE**

機能 Edge/State トリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:ESTate:SOURCE {<Nrf>|EXTERNAL|LINE}  
:TRIGger:EINterval:EVENT<x>:ESTate:SOURCE?  
<x> = 1, 2  
<Nrf> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:ESTATE:SOURCE 1  
:TRIGGER:EINTERVAL:EVENT1:ESTATE:SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:ESTATE:SOURCE 1

解説

- ・「:TRIGger:EINterval:EVENT<x>:TYPE EDGE|EQualify」のときに有効です。
- ・「:TRIGger:EINterval:EVENT<x>:TYPE EDGE」のときは {<Nrf>|EXTERNAL|LINE} が有効です。
- ・「:TRIGger:EINterval:EVENT<x>:TYPE EQualify」のときは {<Nrf>|EXTERNAL} が有効です。

### 5.30 TRIGger グループ

#### **:TRIGger:EIInterval:EVENT<x>:I2Cbus?**

機能 各イベントのI<sup>2</sup>Cバストリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:I2Cbus?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS?  
-> :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:  
ADATA:BIT10ADDRESS:  
PATTERN "10111011111";:TRIGGER:  
EIINTERVAL:EVENT1:I2CBUS:ADATA:  
BIT7ADDRESS:PATTERN "11011110";:  
TRIGGER:EIINTERVAL:EVENT1:I2CBUS:  
ADATA:BIT7APSUB:ADDRESS:  
PATTERN "10101011";:TRIGGER:  
EIINTERVAL:EVENT1:I2CBUS:ADATA:  
BIT7APSUB:SADDRESS:  
PATTERN "10101011";:TRIGGER:  
EIINTERVAL:EVENT1:I2CBUS:ADATA:  
TYPE BIT10ADDRESS;:TRIGGER:EIINTERVAL:  
EVENT1:I2CBUS:CLOCK:SOURCE 1;:  
TRIGGER:EIINTERVAL:EVENT1:I2CBUS:DATA:  
BYTE 1;CONDITION TRUE;DPOSITION 1;  
MODE 1;PATTERN1 "10101011";  
PATTERN2 "10101010";  
PATTERN3 "10101111";  
PATTERN4 "10101011";PMODE DONTCARE;  
SOURCE 1;:TRIGGER:EIINTERVAL:EVENT1:  
I2CBUS:GCALL:BIT7MADDRESS:  
PATTERN "1010101";:TRIGGER:EIINTERVAL:  
EVENT1:I2CBUS:GCALL:  
SBYTE BIT7MADDRESS;:TRIGGER:  
EIINTERVAL:EVENT1:I2CBUS:MODE ADATA;  
NAIGNORE:HSMODE 1;RACCESS 1;SBYTE 1;:  
TRIGGER:EIINTERVAL:EVENT1:I2CBUS:  
SBHSMODE:TYPE HSMODE

#### **:TRIGger:EIInterval:EVENT<x>:I2Cbus:ADATA?**

機能 I<sup>2</sup>Cバストリガのアドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:I2Cbus:  
ADATA?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:  
ADATA? -> :TRIGGER:EIINTERVAL:EVENT1:  
I2CBUS:ADATA:BIT10ADDRESS:  
PATTERN "10111011111";:TRIGGER:  
EIINTERVAL:EVENT1:I2CBUS:ADATA:  
BIT7ADDRESS:PATTERN "11011110";:  
TRIGGER:EIINTERVAL:EVENT1:I2CBUS:  
ADATA:BIT7APSUB:ADDRESS:  
PATTERN "10101011";:TRIGGER:  
EIINTERVAL:EVENT1:I2CBUS:ADATA:  
BIT7APSUB:SADDRESS:  
PATTERN "10101011";:TRIGGER:  
EIINTERVAL:EVENT1:I2CBUS:ADATA:  
TYPE BIT10ADDRESS

#### **:TRIGger:EIInterval:EVENT<x>:I2Cbus:ADATA:BIT10address?**

機能 I<sup>2</sup>Cバストリガの10bitアドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:I2Cbus:  
ADATA:BIT10address?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:  
ADATA:BIT10ADDRESS? -> :TRIGGER:  
EIINTERVAL:EVENT1:I2CBUS:ADATA:  
BIT10ADDRESS:PATTERN "10111011111"

#### **:TRIGger:EIInterval:EVENT<x>:I2Cbus:ADATA:BIT10address:HEXA**

機能 I<sup>2</sup>Cバストリガの10bitアドレスをHEXAで設定します。

構文 :TRIGger:EIInterval:EVENT<x>:I2Cbus:  
ADATA:BIT10address:HEXA {<文字列>}  
<x> = 1, 2

<文字列> = '0'~'F','X'の組み合わせ3文字(ビット8は、R/Wビット)  
例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:  
ADATA:BIT10ADDRESS:HEXA "7AB"

#### **:TRIGger:EIInterval:EVENT<x>:I2Cbus:ADATA:BIT10address:PATtern**

機能 I<sup>2</sup>Cバストリガの10bitアドレスをBINARYで設定/問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:I2Cbus:  
ADATA:BIT10address:PATtern {<文字列>}  
:TRIGger:EIInterval:EVENT<x>:I2Cbus:  
ADATA:BIT10address:PATtern?  
<x> = 1, 2

<文字列> = '0','1','X'の組み合わせ11文字(ビット8は、R/Wビット)  
例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:  
ADATA:BIT10ADDRESS:PATTERN  
"10111011111"  
:TRIGGER:EIINTERVAL:EVENT1:I2CBUS:  
ADATA:BIT10ADDRESS:PATTERN?  
-> :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:  
ADATA:BIT10ADDRESS:  
PATTERN "10111011111"

#### **:TRIGger:EIInterval:EVENT<x>:I2Cbus:ADATA:BIT7Address?**

機能 I<sup>2</sup>Cバストリガの7bitアドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:I2Cbus:  
ADATA:BIT7Address?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:  
ADATA:BIT7ADDRESS? -> :TRIGGER:  
EIINTERVAL:EVENT1:I2CBUS:ADATA:  
BIT7ADDRESS:PATTERN "11011110"

**:TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7ADress:HEXA**  
 機能 I<sup>2</sup>C バストリガの 7bit アドレスを HEXA で設定します。  
 構文 :TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7ADress:HEXA {<文字列>}  
 <x> = 1, 2  
 <文字列> = '0' ~ 'F', 'X' の組み合わせ 2 文字 (ビット 0 は、R/W ビット)  
 例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7ADDRESS:HEXA "DE"

**:TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7ADress:PATtern**  
 機能 I<sup>2</sup>C バストリガの 7bit アドレスを BINARY で設定 / 問い合わせします。  
 構文 :TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7ADress:PATtern {<文字列>}  
 :TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7ADress:PATtern?  
 <x> = 1, 2  
 <文字列> = '0', '1', 'X' の組み合わせ 8 文字 (ビット 0 は、R/W ビット)  
 例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7ADDRESS:PATTERN "11011110"  
 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7ADDRESS:PATTERN?  
 -> :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7ADDRESS:PATTERN "11011110"

**:TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7APsub?**  
 機能 I<sup>2</sup>C バストリガの 7bit + Sub アドレスに関するすべての設定値を問い合わせます。  
 構文 :TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7APsub?  
 <x> = 1, 2  
 例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB? -> :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS:PATTERN "10101011";:TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:SADDRESS:PATTERN "10101011"

**:TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7APsub:ADDRESS?**  
 機能 I<sup>2</sup>C バストリガの 7bit + Sub アドレスの 7bit アドレスに関するすべての設定値を問い合わせます。  
 構文 :TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7APsub:ADDRESS?  
 <x> = 1, 2  
 例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS? -> :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS:PATTERN "10101011"

**:TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7APsub:ADDRESS:HEXA**  
 機能 I<sup>2</sup>C バストリガの 7bit + Sub アドレスの 7bit アドレスを HEXA で設定します。  
 構文 :TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7APsub:ADDRESS:HEXA {<文字列>}  
 <x> = 1, 2  
 <文字列> = '0' ~ 'F', 'X' の組み合わせ 2 文字 (ビット 0 は、R/W ビット)  
 例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS:HEXA "AB"

**:TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7APsub:ADDRESS:PATtern**  
 機能 I<sup>2</sup>C バストリガの 7bit + Sub アドレスの 7bit アドレスを BINARY で設定 / 問い合わせします。  
 構文 :TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7APsub:ADDRESS:PATtern {<文字列>}  
 :TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7APsub:ADDRESS:PATtern?  
 <x> = 1, 2  
 <文字列> = '0', '1', 'X' の組み合わせ 8 文字 (ビット 0 は、R/W ビット)  
 例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS:PATTERN "10101011"  
 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS:PATTERN?  
 -> :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS:PATTERN "10101011"

**:TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7APsub:SADDRESS?**  
 機能 I<sup>2</sup>C バストリガの 7bit + Sub アドレスの Sub アドレスに関するすべての設定値を問い合わせます。  
 構文 :TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7APsub:SADDRESS?  
 <x> = 1, 2  
 例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:SADDRESS? -> :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:SADDRESS:PATTERN "10101011"

**:TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7APsub:SADDRESS:HEXA**  
 機能 I<sup>2</sup>C バストリガの 7bit + Sub アドレスの Sub アドレスを HEXA で設定します。  
 構文 :TRIGger:EIInterval:EVENT<x>:I2CBUS:ADATa:BIT7APsub:SADDRESS:HEXA {<文字列>}  
 <x> = 1, 2  
 <文字列> = '0' ~ 'F', 'X' の組み合わせ 2 文字  
 例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:SADDRESS:HEXA "EF"



### 5.30 TRIGger グループ

#### **:TRIGger:EINterval:EVENT<x>:I2Cbus:ADATa:BIT7APsub:SADdress:PATtern**

機能 I<sup>2</sup>C バストリガの 7bit + Sub アドレスの Sub アドレスを BINARY で設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:I2Cbus:ADATa:BIT7APsub:SADdress:  
PATtern {<文字列>}  
:TRIGger:EINterval:EVENT<x>:I2Cbus:  
ADATa:BIT7APsub:SADdress:PATtern?  
<x> = 1, 2

例 <文字列> = '0', '1', 'X' の組み合わせ 8 文字  
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:  
ADATA:BIT7APSUB:SADDRESS:  
PATTERN "10101011"  
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:  
ADATA:BIT7APSUB:SADDRESS:PATTERN?  
-> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:  
ADATA:BIT7APSUB:SADDRESS:  
PATTERN "10101011"

#### **:TRIGger:EINterval:EVENT<x>:I2Cbus:ADATa:TYPE**

機能 I<sup>2</sup>C バストリガのアドレスの種類を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:I2Cbus:  
ADATa:TYPE {BIT10address|BIT7Address|  
BIT7APsub}  
:TRIGger:EINterval:EVENT<x>:I2Cbus:  
ADATa:TYPE?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:  
ADATA:TYPE BIT10ADDRESS  
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:  
ADATA:TYPE? -> :TRIGGER:EINTERVAL:  
EVENT1:I2CBUS:ADATA:TYPE BIT10ADDRESS

#### **:TRIGger:EINterval:EVENT<x>:I2Cbus:CLOCK?**

機能 I<sup>2</sup>C バストリガのクロックに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:I2Cbus:  
CLOCK?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:  
CLOCK? -> :TRIGGER:EINTERVAL:EVENT1:  
I2CBUS:CLOCK:SOURCE 1

#### **:TRIGger:EINterval:EVENT<x>:I2Cbus:CLOCK:SOURCE**

機能 I<sup>2</sup>C バストリガのクロックトレースを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:I2Cbus:  
CLOCK:SOURCE {<Nrf>}  
:TRIGger:EINterval:EVENT<x>:I2Cbus:  
CLOCK:SOURCE?  
<x> = 1, 2  
<Nrf> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:  
CLOCK:SOURCE 1  
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:  
CLOCK:SOURCE? -> :TRIGGER:EINTERVAL:  
EVENT1:I2CBUS:CLOCK:SOURCE 1

#### **:TRIGger:EINterval:EVENT<x>:I2Cbus:DATA?**

機能 I<sup>2</sup>C バストリガのデータに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:I2Cbus:  
DATA?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:  
DATA? -> :TRIGGER:EINTERVAL:EVENT1:  
I2CBUS:DATA:BYTE 1;CONDITION TRUE;  
DPOSITION 1;MODE 1;PATTERN1  
"10101011";PATTERN2 "10101010";  
PATTERN3 "10101111";  
PATTERN4 "10101011";PMODE DONTCARE;  
SOURCE 1

#### **:TRIGger:EINterval:EVENT<x>:I2Cbus:DATA:BYTE**

機能 I<sup>2</sup>C バストリガの設定データ数を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:I2Cbus:  
DATA:BYTE {<Nrf>}  
:TRIGger:EINterval:EVENT<x>:I2Cbus:  
DATA:BYTE?  
<x> = 1, 2  
<Nrf> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:  
DATA:BYTE 1  
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:  
DATA:BYTE? -> :TRIGGER:EINTERVAL:  
EVENT1:I2CBUS:DATA:BYTE 1

**:TRIGger:EINterval:EVENT<x>:I2CBus:DATA:CONDition**

機能 I<sup>2</sup>C バストリガのデータの判定方法 (一致 / 不一致) を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:I2CBus:DATA:CONDition {FALSE|TRUE}  
:TRIGger:EINterval:EVENT<x>:I2CBus:DATA:CONDition?

<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:CONDITION TRUE  
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:CONDITION? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:CONDITION TRUE

**:TRIGger:EINterval:EVENT<x>:I2CBus:DATA:DPOsition**

機能 I<sup>2</sup>C バストリガのデータのパターン比較する位置を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:I2CBus:DATA:DPOsition {<NRF>}  
:TRIGger:EINterval:EVENT<x>:I2CBus:DATA:DPOsition?

<x> = 1, 2

<NRF> = 0 ~ 9999

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:DPOSITION 1  
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:DPOSITION? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:DPOSITION 1

**:TRIGger:EINterval:EVENT<x>:I2CBus:DATA:HEXA<x>**

機能 I<sup>2</sup>C バストリガのデータを HEXA で設定します。

構文 :TRIGger:EINterval:EVENT<x>:I2CBus:DATA:HEXA<x> {<文字列>}

EVENT<x> の <x> = 1, 2

HEXA<x> の <x> = 1 ~ 4

<文字列> = '0' ~ 'F', 'X' の組み合わせ 2 文字

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:HEXA1 "AB"

**:TRIGger:EINterval:EVENT<x>:I2CBus:DATA:MODE**

機能 I<sup>2</sup>C バストリガのデータ条件の有効 / 無効を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:I2CBus:DATA:MODE {<Boolean>}  
:TRIGger:EINterval:EVENT<x>:I2CBus:DATA:MODE?

<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:MODE ON  
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:MODE? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:MODE 1

**:TRIGger:EINterval:EVENT<x>:I2CBus:DATA:PATtern<x>**

機能 I<sup>2</sup>C バストリガのデータを BINARY で設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:I2CBus:DATA:PATtern<x> {<文字列>}  
:TRIGger:EINterval:EVENT<x>:I2CBus:DATA:PATtern<x>?

EVENT<x> の <x> = 1, 2

PATtern<x> の <x> = 1 ~ 4

<文字列> = '0', '1', 'X' の組み合わせ 8 文字

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:PATTERN1 "10101011"  
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:PATTERN1? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:PATTERN1 "10101011"

**:TRIGger:EINterval:EVENT<x>:I2CBus:DATA:PMODE**

機能 I<sup>2</sup>C バストリガのデータのパターン比較先頭位置モードを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:I2CBus:DATA:PMODE {DONTcare|SElect}  
:TRIGger:EINterval:EVENT<x>:I2CBus:DATA:PMODE?

<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:PMODE SELECT  
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:PMODE? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:PMODE SELECT

### 5.30 TRIGger グループ

#### **:TRIGger:EIInterval:EVENT<x>:I2Cbus:DATA:SOURce**

機能 I<sup>2</sup>C バストリガのデータトレースを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:I2Cbus:DATA:SOURce {<NRf>}  
:TRIGger:EIInterval:EVENT<x>:I2Cbus:DATA:SOURce?  
<x> = 1, 2  
<NRf> = 1 ~ 4

例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:DATA:SOURCE 1  
:TRIGGER:EIINTERVAL:EVENT1:I2CBUS:DATA:SOURCE? -> :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:DATA:SOURCE 1

#### **:TRIGger:EIInterval:EVENT<x>:I2Cbus:GCAL1?**

機能 I<sup>2</sup>C バストリガのジェネラルコールに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:I2Cbus:GCAL1?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:GCAL1? -> :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:GCAL1:BIT7MADDRESS:PATTERN "1010101";:TRIGGER:EIINTERVAL:EVENT1:I2CBUS:GCAL1:SBYTE BIT7MADDRESS

#### **:TRIGger:EIInterval:EVENT<x>:I2Cbus:GCAL1:BIT7maddress?**

機能 I<sup>2</sup>C バストリガのジェネラルコールの 7bit マスタアドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:I2Cbus:GCAL1:BIT7maddress?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:GCAL1:BIT7MADDRESS? -> :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:GCAL1:BIT7MADDRESS:PATTERN "1010101"

#### **:TRIGger:EIInterval:EVENT<x>:I2Cbus:GCAL1:BIT7maddress:HEXA**

機能 I<sup>2</sup>C バストリガのジェネラルコールの 7bit マスタアドレスを HEXA で設定します。

構文 :TRIGger:EIInterval:EVENT<x>:I2Cbus:GCAL1:BIT7maddress:HEXA {<文字列>}  
<x> = 1, 2  
<文字列> = '0' ~ 'F', 'X' の組み合わせ 2 文字 (ビット 0 は、'1' に固定)

例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:GCAL1:BIT7MADDRESS:HEXA "AB"

#### **:TRIGger:EIInterval:EVENT<x>:I2Cbus:GCAL1:BIT7maddress:PATtern**

機能 I<sup>2</sup>C バストリガのジェネラルコールの 7bit マスタアドレスを BINARY で設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:I2Cbus:GCAL1:BIT7maddress:PATtern {<文字列>}  
:TRIGger:EIInterval:EVENT<x>:I2Cbus:GCAL1:BIT7maddress:PATtern?  
<x> = 1, 2  
<文字列> = '0', '1', 'X' の組み合わせ 7 文字

例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:GCAL1:BIT7MADDRESS:PATTERN "1010101"  
:TRIGGER:EIINTERVAL:EVENT1:I2CBUS:GCAL1:BIT7MADDRESS:PATTERN?  
-> :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:GCAL1:BIT7MADDRESS:PATTERN "1010101"

#### **:TRIGger:EIInterval:EVENT<x>:I2Cbus:GCAL1:SBYTE (Second Byte)**

機能 I<sup>2</sup>C バストリガのジェネラルコールのセカンドバイトのタイプを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:I2Cbus:GCAL1:SBYTE {BIT7maddress|DONTcare|H04|H06}  
:TRIGger:EIInterval:EVENT<x>:I2Cbus:GCAL1:SBYTE?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:GCAL1:SBYTE BIT7MADDRESS  
:TRIGGER:EIINTERVAL:EVENT1:I2CBUS:GCAL1:SBYTE? -> :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:GCAL1:SBYTE BIT7MADDRESS

#### **:TRIGger:EIInterval:EVENT<x>:I2Cbus:MODE**

機能 I<sup>2</sup>C バストリガのトリガモードを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:I2Cbus:MODE {ADATa|ESTart|GCAL1|NAIgnore|SBHSmode}  
:TRIGger:EIInterval:EVENT<x>:I2Cbus:MODE?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:MODE ADATA  
:TRIGGER:EIINTERVAL:EVENT1:I2CBUS:MODE? -> :TRIGGER:EIINTERVAL:EVENT1:I2CBUS:MODE ADATA

**:TRIGger:EINterval:EVENT<x>:I2CBus:NAIGnore?**

機能 I<sup>2</sup>C バストリガの NON ACK 無視モードに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:I2CBus:NAIGnore?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:NAIGNORE? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:NAIGNORE:HSMODE 1;RACCESS 1;SBYTE 1

**:TRIGger:EINterval:EVENT<x>:I2CBus:NAIGnore:HSMode**

機能 I<sup>2</sup>C バストリガのハイスピードモードで NON ACK を無視する / しないを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:I2CBus:NAIGnore:HSMode {<Boolean>}  
:TRIGger:EINterval:EVENT<x>:I2CBus:NAIGnore:HSMode?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:NAIGNORE:HSMODE ON  
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:NAIGNORE:HSMODE? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:NAIGNORE:HSMODE 1

**:TRIGger:EINterval:EVENT<x>:I2CBus:NAIGnore:RACcess**

機能 I<sup>2</sup>C バストリガのリードアクセスモードで NON ACK を無視する / しないを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:I2CBus:NAIGnore:RACcess {<Boolean>}  
:TRIGger:EINterval:EVENT<x>:I2CBus:NAIGnore:RACcess?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:NAIGNORE:RACCESS ON  
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:NAIGNORE:RACCESS? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:NAIGNORE:RACCESS 1

**:TRIGger:EINterval:EVENT<x>:I2CBus:NAIGnore:SBYTE (Start Byte)**

機能 I<sup>2</sup>C バストリガのスタートバイトで NON ACK を無視する / しないを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:I2CBus:NAIGnore:SBYTE {<Boolean>}  
:TRIGger:EINterval:EVENT<x>:I2CBus:NAIGnore:SBYTE?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:NAIGNORE:SBYTE ON  
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:NAIGNORE:SBYTE? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:NAIGNORE:SBYTE 1

**:TRIGger:EINterval:EVENT<x>:I2CBus:SBHSmode?**

機能 I<sup>2</sup>C バストリガのスタートバイト / ハイスピードモードに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:I2CBus:SBHSmode?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:SBHSMODE? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:SBHSMODE:TYPE HSMODE

**:TRIGger:EINterval:EVENT<x>:I2CBus:SBHSmode:TYPE**

機能 I<sup>2</sup>C バストリガのスタートバイト / ハイスピードモードのタイプを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:I2CBus:SBHSmode:TYPE {HSMODE|SBYTE}  
:TRIGger:EINterval:EVENT<x>:I2CBus:SBHSmode:TYPE?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:I2CBUS:SBHSMODE:TYPE HSMODE  
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:SBHSMODE:TYPE? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:SBHSMODE:TYPE HSMODE

**:TRIGger:EINterval:EVENT<x>:LINBus?**

機能 各イベントの LIN バス信号トリガに関するすべての設定値を問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LINBus?  
<x> = 1,2

例 TRIGGER:EINTERVAL:EVENT1:LINBUS?  
-> :TRIGGER:EINTERVAL:EVENT1:LINBUS:BRATE 19200;SOURCE 1

### 5.30 TRIGger グループ

#### :TRIGger:EIInterval:EVENT<x>:LINBus:

##### BRATe

機能 LIN バス信号トリガのビットレート (データ転送速度) を設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LINBus:

BRATe {<Nrf>|USER,<Nrf>}

:TRIGger:EIInterval:EVENT<x>:LINBus:

BRATe?

<x> = 1,2

<Nrf> = 1200、2400、4800、9600、19200

USER の <Nrf> = 本体ユーザーズマニュアル参照。

例 :TRIGGER:EIInterval:EVENT1:LINBUS:

BRATE 19200

:TRIGGER:EIInterval:EVENT1:LINBUS:

BRATE? -> :TRIGGER:EIInterval:EVENT1:

LINBUS:BRATE 19200

#### :TRIGger:EIInterval:EVENT<x>:LINBus:

##### SOURCE

機能 LIN バス信号トリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LINBus:

SOURCE {<Nrf>}

:TRIGger:EIInterval:EVENT<x>:LINBus:

SOURCE?

<x> = 1,2

<Nrf> = 1 ~ 4

例 :TRIGGER:EIInterval:EVENT1:LINBUS:

SOURCE 1

:TRIGGER:EIInterval:EVENT1:LINBUS:

SOURCE? -> :TRIGGER:EIInterval:EVENT1:

LINBUS:SOURCE 1

#### :TRIGger:EIInterval:EVENT<x>:LOGic?

機能 各イベントのロジックトリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic?

<x> = 1、2

例 :TRIGGER:EIInterval:EVENT1:LOGIC?

-> :TRIGGER:EIInterval:EVENT1:LOGIC:

CLOCK:POLARITY RISE;SOURCE A0;:

TRIGGER:EIInterval:EVENT1:LOGIC:

ESTATE:POLARITY RISE;SOURCE A0;:

TRIGGER:EIInterval:EVENT1:LOGIC:

I2CBUS:ADATA:BIT10ADDRESS:

PATTERN "1011101111";:TRIGGER:

EIInterval:EVENT1:LOGIC:I2CBUS:

ADATA:BIT7ADDRESS:

PATTERN "11011110";:TRIGGER:

EIInterval:EVENT1:LOGIC:I2CBUS:

ADATA:BIT7APSUB:ADDRESS:

PATTERN "10101011";:TRIGGER:

EIInterval:EVENT1:LOGIC:I2CBUS:ADATA:

BIT7APSUB:SADDRESS:

PATTERN "10101011";:TRIGGER:

EIInterval:EVENT1:LOGIC:I2CBUS:ADATA:

TYPE BIT7ADDRESS;:TRIGGER:EIInterval:

EVENT1:LOGIC:I2CBUS:CLOCK:SOURCE A0;:

TRIGGER:EIInterval:EVENT1:LOGIC:

I2CBUS:DATA:BYTE 1;CONDITION FALSE;

DPOSITION 1;MODE 1;

PATTERN1 "10101011";

PATTERN2 "XXXXXXXX";

PATTERN3 "XXXXXXXX";

PATTERN4 "XXXXXXXX";PMode

DONTCARE;SOURCE A0;:TRIGGER:

EIInterval:EVENT1:LOGIC:I2CBUS:GCALL:

BIT7MADDRESS:PATTERN "1010101";:

TRIGGER:EIInterval:EVENT1:LOGIC:

I2CBUS:GCALL:SBYTE BIT7MADDRESS;:

TRIGGER:EIInterval:EVENT1:LOGIC:

I2CBUS:MODE ADATA;NAIGNORE:HSMODE 1;

RACCESS 1;SBYTE 1;:TRIGGER:EIInterval:

EVENT1:LOGIC:I2CBUS:SBHSMODE:

TYPE HSMODE;:TRIGGER:EIInterval:

EVENT1:LOGIC:LINBUS:BRATE 19200;

SOURCE A0;:TRIGGER:EIInterval:EVENT1:

LOGIC:SPATTERN:BITRATE 1.0000000E+03;

CLOCK:MODE 1;POLARITY FALL .....

#### :TRIGger:EIInterval:EVENT<x>:LOGic:

##### CLOCK?

機能 ロジックトリガのクロックに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:

CLOCK?

<x> = 1、2

例 :TRIGGER:EIInterval:EVENT1:LOGIC:

CLOCK? -> :TRIGGER:EIInterval:EVENT1:

LOGIC:CLOCK:POLARITY RISE;SOURCE A0

**:TRIGGER:EVENT<x>:LOGIC:  
CLOCK:POLARITY**

機能 ロジックトリガのクロックの極性を設定 / 問い合わせします。

構文 :TRIGGER:EVENT<x>:LOGIC:  
CLOCK:POLARITY {FALL|RISE}  
:TRIGGER:EVENT<x>:LOGIC:  
CLOCK:POLARITY?  
<x> = 1, 2

例 :TRIGGER:EVENT1:LOGIC:  
CLOCK:POLARITY FALL  
:TRIGGER:EVENT1:LOGIC:  
CLOCK:POLARITY? -> :TRIGGER:  
EVENT1:LOGIC:CLOCK:  
POLARITY FALL

**:TRIGGER:EVENT<x>:LOGIC:  
CLOCK:SOURCE**

機能 ロジックトリガのクロックのソースを設定 / 問い合わせします。

構文 :TRIGGER:EVENT<x>:LOGIC:  
CLOCK:SOURCE {A<y>|B<y>|C<y>|D<y>|  
DONTcare}  
:TRIGGER:EVENT<x>:LOGIC:  
CLOCK:SOURCE?

例 :TRIGGER:EVENT1:LOGIC:  
CLOCK:  
SOURCE A0  
:TRIGGER:EVENT1:LOGIC:  
CLOCK:SOURCE? -> :TRIGGER:EVENT1:  
EVENT1:LOGIC:CLOCK:SOURCE A0

解説 DL9510L、DL9505L は {A<y>|C<y>|DONTcare} が有効です。

**:TRIGGER:EVENT<x>:LOGIC:  
ESTATE?**

機能 ロジックの Edge/State トリガに関するすべての設定値を問い合わせます。

構文 :TRIGGER:EVENT<x>:LOGIC:  
ESTATE?  
<x> = 1, 2

例 :TRIGGER:EVENT1:LOGIC:  
ESTATE? -> :TRIGGER:EVENT1:  
LOGIC:ESTATE:POLARITY RISE;SOURCE A0

**:TRIGGER:EVENT<x>:LOGIC:  
ESTATE:POLARITY**

機能 ロジックの Edge/State トリガの極性を設定 / 問い合わせします。

構文 :TRIGGER:EVENT<x>:LOGIC:  
ESTATE:POLARITY {ENTER|EXIT|FALL|  
RISE}  
:TRIGGER:EVENT<x>:LOGIC:  
ESTATE:POLARITY?

例 :TRIGGER:EVENT1:LOGIC:  
ESTATE:  
POLARITY ENTER  
:TRIGGER:EVENT1:LOGIC:  
ESTATE:POLARITY? -> :TRIGGER:  
EVENT1:LOGIC:ESTATE:  
POLARITY ENTER

解説  
・「:TRIGGER:EVENT<x>:TYPE  
LState」のときは、{ENTER|EXIT} が有効です。  
・「:TRIGGER:EVENT<x>:TYPE  
LState」以外のときは、{FALL|RISE} が有効です。

**:TRIGGER:EVENT<x>:LOGIC:  
ESTATE:SOURCE**

機能 ロジックの Edge/State トリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGGER:EVENT<x>:LOGIC:  
ESTATE:SOURCE {A<y>|B<y>|C<y>|D<y>}  
:TRIGGER:EVENT<x>:LOGIC:  
ESTATE:SOURCE?

例 :TRIGGER:EVENT1:LOGIC:  
ESTATE:  
SOURCE A0  
:TRIGGER:EVENT1:LOGIC:  
ESTATE:  
SOURCE? -> :TRIGGER:EVENT1:  
LOGIC:ESTATE:SOURCE A0

解説 DL9510L、DL9505L は {A<y>|C<y>} が有効です。

### 5.30 TRIGger グループ

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:I2CBus?**

機能 各イベントのロジック I<sup>2</sup>C バストリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:I2CBus?

<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA:BIT10ADDRESS:PATTERN "10111011111";:TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA:BIT7ADDRESS:PATTERN "11011110";:TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA:BIT7APSUB:ADDRESS:PATTERN "10101011";:TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA:BIT7APSUB:SADDRESS:PATTERN "10101011";:TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA:TYPE BIT7ADDRESS;:TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:CLOCK:SOURCE A0;:TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:DATA:BYTE 1;CONDITION FALSE;DPOSITION 1;MODE 1;PATTERN1 "10101011";PATTERN2 "XXXXXXXX";PATTERN3 "XXXXXXXX";PATTERN4 "XXXXXXXX";PMODE DONTCARE;SOURCE A0;:TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:GCALL:BIT7MADDRESS:PATTERN "1010101";:TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:GCALL:SBYTE BIT7MADDRESS;:TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:MODE ADATA;NAIGNORE:HSMODE 1;RACCESS 1;SBYTE 1;:TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:SBHSMODE:TYPE HSMODE

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:I2CBus:ADATa?**

機能 ロジック I<sup>2</sup>C バストリガのアドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:I2CBus:ADATa?

<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA? ->:TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA:BIT10ADDRESS:PATTERN "10111011111";:TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA:BIT7ADDRESS:PATTERN "11011110";:TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA:BIT7APSUB:ADDRESS:PATTERN "10101011";:TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA:BIT7APSUB:SADDRESS:PATTERN "10101011";:TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA:TYPE BIT7ADDRESS

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:I2CBus:ADATa:BIT10address?**

機能 ロジック I<sup>2</sup>C バストリガの 10bit アドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:I2CBus:ADATa:BIT10address?

<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA:BIT10ADDRESS?-> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA:BIT10ADDRESS:PATTERN "10111011111"

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:I2CBus:ADATa:BIT10address:HEXA**

機能 ロジック I<sup>2</sup>C バストリガの 10bit アドレスを HEXA で設定します。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:I2CBus:ADATa:BIT10address:HEXA {<文字列>}

<x> = 1, 2

<文字列> = '0' ~ 'F', 'X' の組み合わせ 3 文字 (ビット 8 は、R/W ビット)

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA:BIT10ADDRESS:HEXA "7AB"

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:ADATa:BIT10address:PATtern**  
機能 ロジック I<sup>2</sup>C バストリガの 10bit アドレスを  
BINARY で設定 / 問い合わせします。  
構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:ADATa:BIT10address:  
PATtern {<文字列>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:ADATa:BIT10address:PATtern?  
<x> = 1, 2  
<文字列> = '0','1','X'の組み合わせ 11 文字(ビッ  
ト 8 は、R/W ビット)  
例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT10ADDRESS:  
PATTERN "10111011111"  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT10ADDRESS:PATTERN?  
-> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT10ADDRESS:  
PATTERN "10111011111"

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:ADATa:BIT7Address?**  
機能 ロジック I<sup>2</sup>C バストリガの 7bit アドレスに関する  
すべての設定値を問い合わせます。  
構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:ADATa:BIT7Address?  
<x> = 1, 2  
例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7ADDRESS?  
-> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7ADDRESS:  
PATTERN "11011110"

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:ADATa:BIT7Address:HEXA**  
機能 ロジック I<sup>2</sup>C バストリガの 7bit アドレスを HEXA  
で設定します。  
構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:ADATa:BIT7Address:  
HEXA {<文字列>}  
<x> = 1, 2  
<文字列> = '0' ~ 'F','X'の組み合わせ 2 文字(ビッ  
ト 0 は、R/W ビット)  
例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7ADDRESS:HEXA "DE"

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:ADATa:BIT7Address:PATtern**  
機能 ロジック I<sup>2</sup>C バストリガの 7bit アドレスを  
BINARY で設定 / 問い合わせします。  
構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:ADATa:BIT7Address:  
PATtern {<文字列>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:ADATa:BIT7Address:PATtern?  
<x> = 1, 2  
<文字列> = '0','1','X'の組み合わせ 8 文字(ビッ  
ト 0 は、R/W ビット)  
例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7ADDRESS:  
PATTERN "11011110"  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7ADDRESS:PATTERN?  
-> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7ADDRESS:  
PATTERN "11011110"

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:ADATa:BIT7APsub?**  
機能 ロジック I<sup>2</sup>C バストリガの 7bit + Sub アドレス  
に関するすべての設定値を問い合わせます。  
構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:ADATa:BIT7APsub?  
<x> = 1, 2  
例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7APSUB? -> :TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:I2CBUS:  
ADATA:BIT7APSUB:ADDRESS:  
PATTERN "10101011";:TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA:  
BIT7APSUB:SADDRESS:PATTERN "10101011"

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:ADATa:BIT7APsub:ADDRESS?**  
機能 ロジック I<sup>2</sup>C バストリガの 7bit + Sub アドレス  
の 7bit アドレスに関するすべての設定値を問  
合わせます。  
構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:ADATa:BIT7APsub:ADDRESS?  
<x> = 1, 2  
例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7APSUB:ADDRESS?  
-> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7APSUB:ADDRESS:  
PATTERN "10101011"



### 5.30 TRIGger グループ

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2Cbus:ADATa:BIT7APsub:ADDRESS:HEXA**  
機能 ロジック I<sup>2</sup>C バストリガの 7bit + Sub アドレス  
の 7bit アドレスを HEXA で設定します。  
構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2Cbus:ADATa:BIT7APsub:ADDRESS:  
HEXA {<文字列>}  
<x> = 1、2  
<文字列> = '0' ~ 'F'、'X' の組み合わせ 2 文字 (ビット  
0 は、R/W ビット)  
例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7APSUB:ADDRESS:  
HEXA "AB"

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2Cbus:ADATa:BIT7APsub:ADDRESS:  
PATTERN**  
機能 ロジック I<sup>2</sup>C バストリガの 7bit + Sub アドレス  
の 7bit アドレスを BINARY で設定 / 問い合わせし  
ます。  
構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2Cbus:ADATa:BIT7APsub:ADDRESS:  
PATTERN {<文字列>}  
<x> = 1、2  
<文字列> = '0'、'1'、'X' の組み合わせ 8 文字 (ビット  
0 は、R/W ビット)  
例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7APSUB:ADDRESS:  
PATTERN "10101011"  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7APSUB:ADDRESS:  
PATTERN? -> :TRIGGER:EIINTERVAL:  
EVENT1:LOGIC:I2CBUS:ADATA:BIT7APSUB:  
ADDRESS:PATTERN "10101011"

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2Cbus:ADATa:BIT7APsub:SADDRESS?**  
機能 ロジック I<sup>2</sup>C バストリガの 7bit + Sub アドレス  
の Sub アドレスに関するすべての設定値を問  
い合わせます。  
構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2Cbus:ADATa:BIT7APsub:SADDRESS?  
<x> = 1、2  
例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7APSUB:SADDRESS? ->  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7APSUB:SADDRESS:  
PATTERN "10101011"

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2Cbus:ADATa:BIT7APsub:SADDRESS:HEXA**  
機能 ロジック I<sup>2</sup>C バストリガの 7bit + Sub アドレス  
の Sub アドレスを HEXA で設定します。  
構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2Cbus:ADATa:BIT7APsub:SADDRESS:  
HEXA {<文字列>}  
<x> = 1、2  
<文字列> = '0' ~ 'F'、'X' の組み合わせ 2 文字  
例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7APSUB:SADDRESS:  
HEXA "EF"

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2Cbus:ADATa:BIT7APsub:SADDRESS:  
PATTERN**  
機能 ロジック I<sup>2</sup>C バストリガの 7bit + Sub アドレス  
の Sub アドレスを BINARY で設定 / 問い合わせし  
ます。  
構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2Cbus:ADATa:BIT7APsub:SADDRESS:  
PATTERN {<文字列>}  
<x> = 1、2  
<文字列> = '0'、'1'、'X' の組み合わせ 8 文字  
例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7APSUB:SADDRESS:  
PATTERN "10101011"  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:BIT7APSUB:SADDRESS:  
PATTERN? -> :TRIGGER:EIINTERVAL:  
EVENT1:LOGIC:I2CBUS:ADATA:BIT7APSUB:  
SADDRESS:PATTERN "10101011"

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2Cbus:ADATa:TYPE**  
機能 ロジック I<sup>2</sup>C バストリガのアドレスの種類を設定  
 / 問い合わせします。  
構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2Cbus:ADATa:TYPE {BIT10address|  
BIT7Address|BIT7APsub}  
<x> = 1、2  
例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:TYPE BIT10ADDRESS  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:ADATA:TYPE? -> :TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:I2CBUS:ADATA:  
TYPE BIT10ADDRESS

**:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:CLOCK?**

機能 ロジック I<sup>2</sup>C バストリガのクロックに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:CLOCK?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:CLOCK? -> :TRIGGER:EINTERVAL:  
EVENT1:LOGIC:I2CBUS:CLOCK:SOURCE A0

**:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:CLOCK:SOURCE**

機能 ロジック I<sup>2</sup>C バストリガのクロックトレースを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:CLOCK:SOURCE {A<y>|B<y>|C<y>|  
D<y>}

:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:CLOCK:SOURCE?  
<x> = 1, 2  
<y> = 0 ~ 7

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:CLOCK:SOURCE A0  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:CLOCK:SOURCE? -> :TRIGGER:  
EINTERVAL:EVENT1:LOGIC:I2CBUS:CLOCK:  
SOURCE A0

解説 DL9510L、DL9505L は {A<y>|C<y>} が有効です。

**:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:DATA?**

機能 ロジック I<sup>2</sup>C バストリガのデータに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:DATA?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA? -> :TRIGGER:EINTERVAL:  
EVENT1:LOGIC:I2CBUS:DATA:BYTE 1;  
CONDITION FALSE;DPOSITION 1;MODE 1;  
PATTERN1 "10101011";  
PATTERN2 "XXXXXXXX";  
PATTERN3 "XXXXXXXX";  
PATTERN4 "XXXXXXXX";PMODE DONTCARE;  
SOURCE A0

**:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:DATA:BYTE**

機能 ロジック I<sup>2</sup>C バストリガの設定データ数を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:DATA:BYTE {<NRf>}  
:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:DATA:BYTE?

<x> = 1, 2  
<NRf> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA:BYTE 1  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA:BYTE? -> :TRIGGER:  
EINTERVAL:EVENT1:LOGIC:I2CBUS:DATA:  
BYTE 1

**:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:DATA:CONDITION**

機能 ロジック I<sup>2</sup>C バストリガのデータの判定方法 (一致 / 不一致) を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:DATA:CONDITION {FALSe|TRUE}  
:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:DATA:CONDITION?

<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA:CONDITION FALSE  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA:CONDITION? -> :TRIGGER:  
EINTERVAL:EVENT1:LOGIC:I2CBUS:DATA:  
CONDITION FALSE

**:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:DATA:DPOSITION**

機能 ロジック I<sup>2</sup>C バストリガのデータのパターン比較する位置を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:DATA:DPOSITION {<NRf>}  
:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:DATA:DPOSITION?

<x> = 1, 2  
<NRf> = 0 ~ 9999

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA:DPOSITION 1  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA:DPOSITION? -> :TRIGGER:  
EINTERVAL:EVENT1:LOGIC:I2CBUS:DATA:  
DPOSITION 1

### 5.30 TRIGger グループ

#### **:TRIGger:EIInterval:EVENT<x>:LOGic: I2CBus:DATA:HEXA<x>**

機能 ロジック I<sup>2</sup>C バストリガのデータを HEXA で設定  
します。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:DATA:HEXA<x> {<文字列>}  
EVENT<x> の <x> = 1、2  
HEXA<x> の <x> = 1 ~ 4  
<文字列> = '0' ~ 'F'、'X' の組み合わせ 2 文字

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA:HEXA1 "AB"

#### **:TRIGger:EIInterval:EVENT<x>:LOGic: I2CBus:DATA:MODE**

機能 ロジック I<sup>2</sup>C バストリガのデータ条件の有効 / 無  
効を設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:DATA:MODE {<Boolean>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:DATA:MODE?  
<x> = 1、2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA:MODE ON  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA:MODE? -> :TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:I2CBUS:DATA:  
MODE 1

#### **:TRIGger:EIInterval:EVENT<x>:LOGic: I2CBus:DATA:PATtern<x>**

機能 ロジック I<sup>2</sup>C バストリガのデータを BINARY で設  
定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:DATA:PATtern<x> {<文字列>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:DATA:PATtern<x>?  
EVENT<x> の <x> = 1、2  
PATtern<x> の <x> = 1 ~ 4  
<文字列> = '0'、'1'、'X' の組み合わせ 8 文字

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA:PATTERN1 "10101011"  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA:PATTERN1? -> :TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:I2CBUS:DATA:  
PATTERN1 "10101011"

#### **:TRIGger:EIInterval:EVENT<x>:LOGic: I2CBus:DATA:PMODE**

機能 ロジック I<sup>2</sup>C バストリガのデータのパターン比較  
先頭位置モードを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:DATA:PMODE {DONTcare|SElect}  
:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:DATA:PMODE?  
<x> = 1、2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA:PMODE DONTCARE  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA:PMODE? -> :TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:I2CBUS:DATA:  
PMODE DONTCARE

#### **:TRIGger:EIInterval:EVENT<x>:LOGic: I2CBus:DATA:SOURce**

機能 ロジック I<sup>2</sup>C バストリガのデータトレースを設定  
 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:DATA:SOURce {A<y>|B<y>|C<y>|  
D<y>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:DATA:SOURce?  
<x> = 1、2  
<y> = 0 ~ 7

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA:SOURCE A0  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:DATA:SOURCE? -> :TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:I2CBUS:DATA:  
SOURCE A0

解説 DL9510L、DL9505L は {A<y>|C<y>} が有効です。

#### **:TRIGger:EIInterval:EVENT<x>:LOGic: I2CBus:GCALL?**

機能 ロジック I<sup>2</sup>C バストリガのジェネラルコールに関  
するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:GCALL?  
<x> = 1、2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:GCALL? -> :TRIGGER:EIINTERVAL:  
EVENT1:LOGIC:I2CBUS:GCALL:  
BIT7MADDRESS:PATTERN "1010101";:  
TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:GCALL:SBYTE BIT7MADDRESS

**:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:GCALl:BIT7maddress?**

機能 ロジック I<sup>2</sup>C バストリガのジェネラルコールの 7bit マスタアドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:GCALl:BIT7maddress?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:GCALL:BIT7MADDRESS?  
-> :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:GCALL:BIT7MADDRESS:  
PATTERN "1010101"

**:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:GCALl:BIT7maddress:HEXA**

機能 ロジック I<sup>2</sup>C バストリガのジェネラルコールの 7bit マスタアドレスを HEXA で設定します。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:GCALl:BIT7maddress:HEXA {<文字列>}  
<x> = 1, 2  
<文字列> = '0' ~ 'F', 'X' の組み合わせ 2 文字 (ビット 0 は、'1' に固定)

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:GCALL:BIT7MADDRESS:HEXA "AB"

**:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:GCALl:BIT7maddress:PATtern**

機能 ロジック I<sup>2</sup>C バストリガのジェネラルコールの 7bit マスタアドレスを BINARY で設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:GCALl:BIT7maddress:  
PATtern {<文字列>}  
:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:GCALl:BIT7maddress:PATtern?  
<x> = 1, 2  
<文字列> = '0', '1', 'X' の組み合わせ 7 文字

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:GCALL:BIT7MADDRESS:  
PATTERN "1010101"  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:GCALL:BIT7MADDRESS:PATTERN?  
-> :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:GCALL:BIT7MADDRESS:  
PATTERN "1010101"

**:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:GCALl:SBYTE (Second Byte)**

機能 ロジック I<sup>2</sup>C バストリガのジェネラルコールのセカンダリバイトのタイプを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:GCALl:SBYTE {BIT7maddress|  
DONTcare|H04|H06}  
:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:GCALl:SBYTE?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:GCALL:SBYTE BIT7MADDRESS  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:GCALL:SBYTE? -> :TRIGGER:  
EINTERVAL:EVENT1:LOGIC:I2CBUS:GCALL:  
SBYTE BIT7MADDRESS

**:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:MODE**

機能 ロジック I<sup>2</sup>C バストリガのトリガモードを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:MODE {ADATa|ESTart|GCALl|  
NAIGNore|SBHSmode}  
:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:MODE?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:MODE ADATA  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:MODE? -> :TRIGGER:EINTERVAL:  
EVENT1:LOGIC:I2CBUS:MODE ADATA

**:TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:NAIGNore?**

機能 ロジック I<sup>2</sup>C バストリガの NON ACK 無視モードに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
I2CBus:NAIGNore?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
I2CBUS:NAIGNORE? -> :TRIGGER:  
EINTERVAL:EVENT1:LOGIC:I2CBUS:  
NAIGNORE:HSMODE 1;RACCESS 1;SBYTE 1

### 5.30 TRIGger グループ

#### **:TRIGger:EIInterval:EVENT<x>:LOGic: I2CBus:NAIIgnore:HSMoDe**

機能 ロジック I<sup>2</sup>C バストリガのハイスピードモードで NON ACK を無視する / しないを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:NAIIgnore:HSMoDe {<Boolean>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:NAIIgnore:HSMoDe?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:NAIGNORE:HSMODE ON  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:NAIGNORE:HSMODE? -> :TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:I2CBUS:  
NAIGNORE:HSMODE 1

#### **:TRIGger:EIInterval:EVENT<x>:LOGic: I2CBus:NAIIgnore:RACcEss**

機能 ロジック I<sup>2</sup>C バストリガのリードアクセスモードで NON ACK を無視する / しないを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:NAIIgnore:RACcEss {<Boolean>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:NAIIgnore:RACcEss?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:NAIGNORE:RACCESS ON  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:NAIGNORE:RACCESS? -> :TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:I2CBUS:  
NAIGNORE:RACCESS 1

#### **:TRIGger:EIInterval:EVENT<x>:LOGic: I2CBus:NAIIgnore:SBYTe (Start Byte)**

機能 ロジック I<sup>2</sup>C バストリガのスタートバイトで NON ACK を無視する / しないを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:NAIIgnore:SBYTe {<Boolean>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:NAIIgnore:SBYTe?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:NAIGNORE:SBYTE ON  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:NAIGNORE:SBYTE? -> :TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:I2CBUS:  
NAIGNORE:SBYTE 1

#### **:TRIGger:EIInterval:EVENT<x>:LOGic: I2CBus:SBHSMoDe?**

機能 ロジック I<sup>2</sup>C バストリガのスタートバイト / ハイスピードモードに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:SBHSMoDe?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:SBHSMODE? -> :TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:I2CBUS:  
SBHSMODE:TYPE HSMODE

#### **:TRIGger:EIInterval:EVENT<x>:LOGic: I2CBus:SBHSMoDe:TYPE**

機能 ロジック I<sup>2</sup>C バストリガのスタートバイト / ハイスピードモードのタイプを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:SBHSMoDe:TYPE {HSMoDe|SBYTe}  
:TRIGger:EIInterval:EVENT<x>:LOGic:  
I2CBus:SBHSMoDe:TYPE?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:SBHSMODE:TYPE HSMODE  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
I2CBUS:SBHSMODE:TYPE? -> :TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:I2CBUS:  
SBHSMODE:TYPE HSMODE

#### **:TRIGger:EIInterval:EVENT<x>:LOGic: LINBus?**

機能 各イベントのロジック LIN バス信号トリガに関するすべての設定値を問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
LINBus?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
LINBUS? -> :TRIGGER:EIINTERVAL:EVENT1:  
LOGIC:LINBUS:BRATE 19200;SOURCE A0

#### **:TRIGger:EIInterval:EVENT<x>:LOGic: LINBus:BRATe**

機能 ロジック LIN バス信号トリガのビットレート (データ転送速度) を設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
LINBus:BRATe {<NRf>|USER,<NRf>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:  
LINBus:BRATe?  
<x> = 1, 2

<NRf> = 1200, 2400, 4800, 9600, 19200  
USER の <NRf> = 本体ユーザーズマニュアル参照。  
例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
LINBUS:BRATE 19200  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
LINBUS:BRATE? -> :TRIGGER:EIINTERVAL:  
EVENT1:LOGIC:LINBUS:BRATE 19200

**:TRIGger:EINterval:EVENT<x>:LOGic:  
LINBus:SOURce**

機能 ロジック LIN バス信号トリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
LINBus:SOURce {A<y>|B<y>|C<y>|D<y>}  
:TRIGger:EINterval:EVENT<x>:LOGic:  
LINBus:SOURce?  
<x> = 1, 2  
<y> = 0 ~ 7

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
LINBUS:SOURCE A0  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:  
LINBUS:SOURCE? -> :TRIGGER:EINTERVAL:  
EVENT1:LOGIC:LINBUS:SOURCE A0

解説 DL9510L、DL9505L は {A<y>|C<y>} が有効です。

**:TRIGger:EINterval:EVENT<x>:LOGic:  
SPATtern? (Serial Pattern)**

機能 各イベントのロジックシリアルパターントリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
SPATtern?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
SPATTERN? -> :TRIGGER:EINTERVAL:  
EVENT1:LOGIC:SPATTERN:  
BITRATE 1.0000000E+03;CLOCK:MODE 1;  
POLARITY FALL;SOURCE A0;;TRIGGER:  
EINTERVAL:EVENT1:LOGIC:SPATTERN:CS 1;  
DATA:ACTIVE HIGH;SOURCE A0;;TRIGGER:  
EINTERVAL:EVENT1:LOGIC:SPATTERN:  
LATCH:SOURCE A0;POLARITY FALL;;  
TRIGGER:EINTERVAL:EVENT1:LOGIC:  
SPATTERN:PATTERN "1100110111101111"

**:TRIGger:EINterval:EVENT<x>:LOGic:  
SPATtern:BITRate**

機能 ロジックシリアルパターントリガのビットレートを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
SPATtern:BITRate {<NRf>}  
:TRIGger:EINterval:EVENT<x>:LOGic:  
SPATtern:BITRate?  
<x> = 1, 2  
<NRf> = 1 ~ 50M(bps)

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
SPATTERN:BITRATE 1  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:  
SPATTERN:BITRATE? -> :TRIGGER:  
EINTERVAL:EVENT1:LOGIC:  
SPATTERN:BITRATE 1.000E+00

解説 「:TRIGger:EINterval:EVENT<x>:LOGic:  
SPATtern:CLOCK:MODE OFF」のときに有効で  
す。

**:TRIGger:EINterval:EVENT<x>:LOGic:  
SPATtern:CLEAr**

機能 ロジックシリアルパターントリガのパターンをすべてクリア (Don't care) します。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
SPATtern:CLEAr  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
SPATTERN:CLEAR

**:TRIGger:EINterval:EVENT<x>:LOGic:  
SPATtern:CLOCK?**

機能 ロジックシリアルパターントリガのクロックに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:  
SPATtern:CLOCK?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
SPATTERN:CLOCK? -> :TRIGGER:  
EINTERVAL:EVENT1:LOGIC:SPATTERN:  
CLOCK:MODE 1;POLARITY FALL;SOURCE A0

### 5.30 TRIGger グループ

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CLOCK:MODE**

機能 ロジックシリアルパターントリガのクロックの有効 / 無効を設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CLOCK:MODE {<Boolean>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CLOCK:MODE?

例 <x> = 1, 2  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:CLOCK:MODE ON  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:CLOCK:MODE? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:CLOCK:MODE 1

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CLOCK:POLarity**

機能 ロジックシリアルパターントリガのクロックレーズの極性を設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CLOCK:POLarity {FALL|RISE}  
:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CLOCK:POLarity?

例 <x> = 1, 2  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:CLOCK:POLARITY FALL  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:CLOCK:POLARITY? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:CLOCK:POLARITY FALL

解説 「:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CLOCK:MODE ON」のときに有効です。

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CLOCK:SOURce**

機能 ロジックシリアルパターントリガのクロックトレースを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CLOCK:SOURce {A<y>|B<y>|C<y>|D<y>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CLOCK:SOURce?

例 <x> = 1, 2  
<y> = 0 ~ 7  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:CLOCK:SOURCE A0  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:CLOCK:SOURCE? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:CLOCK:SOURCE A0

解説  
・「:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CLOCK:MODE ON」のときに有効です。  
・DL9510L、DL9505L は {A<y>|C<y>} が有効です。

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CS**

機能 ロジックシリアルパターントリガのチップセレクトの有効 / 無効を設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CS {<Boolean>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CS?

例 <x> = 1, 2  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:CS ON  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:CS? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:CS 1

解説 「:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CLOCK:MODE ON」のときに有効です。

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:DATA?**

機能 ロジックシリアルパターントリガのデータに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:DATA?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:DATA? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:DATA:ACTIVE HIGH;SOURCE A0

**:TRIGger:EINterval:EVENT<x>:LOGic:SPATtern:DATA:ACTive**

機能 ロジックシリアルパターントリガのデータのアクティブレベルを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:SPATtern:DATA:ACTive {HIGH|LOW}  
:TRIGger:EINterval:EVENT<x>:LOGic:SPATtern:DATA:ACTive?

例 <x> = 1, 2  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:SPATTERN:DATA:ACTIVE HIGH  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:SPATTERN:DATA:ACTIVE? -> :TRIGGER:EINTERVAL:EVENT1:LOGIC:SPATTERN:DATA:ACTIVE HIGH

**:TRIGger:EINterval:EVENT<x>:LOGic:SPATtern:DATA:SOURce**

機能 ロジックシリアルパターントリガのデータソースを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:SPATtern:DATA:SOURce {A<y>|B<y>|C<y>|D<y>}  
:TRIGger:EINterval:EVENT<x>:LOGic:SPATtern:DATA:SOURce?

例 <x> = 1, 2  
<y> = 0 ~ 7  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:SPATTERN:DATA:SOURCE A0  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:SPATTERN:DATA:SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:LOGIC:SPATTERN:DATA:SOURCE A0  
解説 DL9510L、DL9505L は {A<y>|C<y>} が有効です。

**:TRIGger:EINterval:EVENT<x>:LOGic:SPATtern:HEXA**

機能 ロジックシリアルパターントリガのパターンをHEXA で設定します。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:SPATtern:HEXA {<文字列>}  
<x> = 1, 2

<文字列> = '0' ~ 'F'、'X' の組み合わせ 32 文字以内。  
例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:SPATTERN:HEXA "ABCD"

**:TRIGger:EINterval:EVENT<x>:LOGic:SPATtern:LATCh?**

機能 ロジックシリアルパターントリガのラッチに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:SPATtern:LATCh?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:SPATTERN:LATCH? -> :TRIGGER:EINTERVAL:EVENT1:LOGIC:SPATTERN:LATCH:SOURCE A0;POLARITY FALL

**:TRIGger:EINterval:EVENT<x>:LOGic:SPATtern:LATCh:POLarity**

機能 ロジックシリアルパターントリガのラッチレースの極性を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:SPATtern:LATCh:POLarity {FALL|RISE}  
:TRIGger:EINterval:EVENT<x>:LOGic:SPATtern:LATCh:POLarity?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:SPATTERN:LATCH:POLARITY FALL  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:SPATTERN:LATCH:POLARITY? -> :TRIGGER:EINTERVAL:EVENT1:LOGIC:SPATTERN:LATCH:POLARITY FALL

解説  
・「:TRIGger:EINterval:EVENT<x>:LOGic:SPATtern:CLOCK:MODE ON」のときに有効です。  
・「:TRIGger:EINterval:EVENT<x>:LOGic:SPATtern:LATCh:SOURce DONTcare」のときは無効です。



### 5.30 TRIGger グループ

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:LATCh:SOURce**

機能 ロジックシリアルパターントリガのラッチトレースを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:LATCh:SOURce {A<y>|B<y>|C<y>|D<y>|DONTcare}

:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:LATCh:SOURce?

<x> = 1、2

<y> = 0 ~ 7

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:LATCH:SOURCE A0

:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:LATCH:SOURCE? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:LATCH:SOURCE A0

解説

- ・「:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:CLOCK:MODE ON」のときに有効です。
- ・DL9510L、DL9505L は {A<y>|C<y>|DONTcare} が有効です。

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:PATtern**

機能 ロジックシリアルパターントリガのパターンをBINARY で設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:PATtern {<文字列>}

:TRIGger:EIInterval:EVENT<x>:LOGic:SPATtern:PATtern?

<x> = 1、2

<文字列> = '0','1','X'の組み合わせ 128文字以内。

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:PATTERN "1100110111101111"

:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:PATTERN? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPATTERN:PATTERN "1100110111101111"

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPIBUS?**

機能 各イベントのロジック SPI バストリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPIBUS?

<x> = 1、2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:BITORDER LSBFIRST;CLOCK:POLARITY FALL;SOURCE A0;:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:CS:ACTIVE HIGH;SOURCE A0;:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:BYTE 1;CONDITION FALSE;DPOSITION 1;PATTERN1 "10101011";PATTERN2 "XXXXXXXX";PATTERN3 "XXXXXXXX";PATTERN4 "XXXXXXXX";SOURCE A0;:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA2:BYTE 1;CONDITION TRUE;DPOSITION 0;PATTERN1 "XXXXXXXX";PATTERN2 "XXXXXXXX";PATTERN3 "XXXXXXXX";PATTERN4 "XXXXXXXX";SOURCE A2;:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:MODE WIRE3

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPIBUS:BITorder**

機能 ロジック SPI バストリガのビットオーダーを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPIBUS:BITorder {LSBFirst|MSBFirst}

:TRIGger:EIInterval:EVENT<x>:LOGic:SPIBUS:BITorder?

<x> = 1、2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:BITORDER LSBFIRST

:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:BITORDER? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:BITORDER LSBFIRST

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPIBUS:CLOCK?**

機能 ロジック SPI バストリガのクロックに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPIBUS:CLOCK?

<x> = 1、2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:CLOCK? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:CLOCK:POLARITY FALL;SOURCE A0

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:CLOCK:POLarity**

機能 ロジック SPI バストリガのクロックトレースの極性を設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:CLOCK:POLarity {FALL|RISE}  
:TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:CLOCK:POLarity?

<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
SPIBUS:CLOCK:POLARITY FALL  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
SPIBUS:CLOCK:POLARITY? -> :TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:SPIBUS:CLOCK:  
POLARITY FALL

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:CLOCK:SOURce**

機能 ロジック SPI バストリガのクロックトレースを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:CLOCK:SOURce {A<y>|B<y>|C<y>|  
D<y>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:CLOCK:SOURce?

<x> = 1, 2

<y> = 0 ~ 7

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
SPIBUS:  
CLOCK:SOURCE A0  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
SPIBUS:CLOCK:SOURCE? -> :TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:SPIBUS:CLOCK:  
SOURCE A0

解説 DL9510L、DL9505L は {A<y>|C<y>} が有効です。

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:CS?**

機能 ロジック SPI バストリガのチップセレクトに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:CS?  
<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
SPIBUS:CS? -> :TRIGGER:EIINTERVAL:  
EVENT1:LOGIC:SPIBUS:CS:ACTIVE HIGH;  
SOURCE A0

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:CS:ACTive**

機能 ロジック SPI バストリガのチップセレクトのアクティブレベルを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:CS:ACTive {HIGH|LOW}  
:TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:CS:ACTive?

<x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
SPIBUS:CS:ACTIVE HIGH  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
SPIBUS:CS:ACTIVE? -> :TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:SPIBUS:CS:  
ACTIVE HIGH

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:CS:SOURce**

機能 ロジック SPI バストリガのチップセレクトトレースを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:CS:SOURce  
{A<y>|B<y>|C<y>|D<y>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:CS:SOURce?

<x> = 1, 2

<y> = 0 ~ 7

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
SPIBUS:CS:SOURCE A0  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
SPIBUS:CS:SOURCE? -> :TRIGGER:  
EIINTERVAL:EVENT1:LOGIC:SPIBUS:CS:  
SOURCE A0

解説 DL9510L、DL9505L は {A<y>|C<y>} が有効です。

**:TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:DATA<x>?**

機能 ロジック SPI バストリガの各データに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:  
SPIBus:DATA<x>?  
EVENT<x> の <x> = 1, 2  
DATA<x> の <x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:  
SPIBUS:DATA1? -> :TRIGGER:EIINTERVAL:  
EVENT1:LOGIC:SPIBUS:DATA1:BYTE 1;  
CONDITION FALSE;DPOSITION 1;  
PATTERN1 "10101011";  
PATTERN2 "XXXXXXXX";  
PATTERN3 "XXXXXXXX";  
PATTERN4 "XXXXXXXX";SOURCE A0

解説 DATA2 は、「:TRIGger:EIInterval:  
EVENT<x>:LOGic:SPIBus:MODE WIRE4」の  
ときに有効です。

### 5.30 TRIGger グループ

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:BYTE**

機能 ロジック SPI バストリガの各データの設定データを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:BYTE {<NRf>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:BYTE?  
EVENT<x> の <x> = 1, 2  
DATA<x> の <x> = 1, 2  
<NRf> = 1 ~ 4

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:BYTE 1  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:BYTE? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:BYTE 1

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:CONDition**

機能 ロジック SPI バストリガの各データの判定方法 (一致 / 不一致) を設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:CONDition {FALSE|TRUE}  
:TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:CONDition?  
EVENT<x> の <x> = 1, 2  
DATA<x> の <x> = 1, 2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:CONDITION FALSE  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:CONDITION? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:CONDITION FALSE

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:DPOSITION**

機能 ロジック SPI バストリガの各データのパターン比較先頭位置を設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:DPOSITION {<NRf>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:DPOSITION?  
EVENT<x> の <x> = 1, 2  
DATA<x> の <x> = 1, 2  
<NRf> = 0 ~ 9999

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:DPOSITION 1  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:DPOSITION? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:DPOSITION 1

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:HEXA<x>**

機能 ロジック SPI バストリガの各データを HEXA で設定します。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:HEXA<x> {<文字列>}  
EVENT<x> の <x> = 1, 2  
DATA<x> の <x> = 1, 2  
HEXA<x> の <x> = 1 ~ 4  
<文字列> = '0' ~ 'F', 'X' の組み合わせ 2 文字。

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:HEXA1 "AB"

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:PATTern<x>**

機能 ロジック SPI バストリガの各データを BINARY で設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:PATTern<x> {<文字列>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:PATTern<x>?  
EVENT<x> の <x> = 1, 2  
DATA<x> の <x> = 1, 2  
PATTern<x> の <x> = 1 ~ 4  
<文字列> = '0', '1', 'X' の組み合わせ 8 文字。

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:PATTERN1 "10101011"  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:PATTERN1? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:PATTERN1 "10101011"

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:SOURCE**

機能 ロジック SPI バストリガの各データのトレースを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:SOURCE {A<y>|B<y>|C<y>|D<y>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:SPIBus:DATA<x>:SOURCE?  
EVENT<x> の <x> = 1, 2  
DATA<x> の <x> = 1, 2  
<y> = 0 ~ 7

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:SOURCE A0  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:SOURCE? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:SPIBUS:DATA1:SOURCE A0

解説 DL9510L、DL9505L は {A<y>|C<y>} が有効です。

**:TRIGger:EINterval:EVENT<x>:LOGic:SPIBus:MODE**

機能 ロジック SPI バストリガの結線方式 (3 線式 / 4 線式) を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:SPIBus:MODE {WIRE3|WIRE4}  
:TRIGger:EINterval:EVENT<x>:LOGic:SPIBus:MODE?  
<x> = 1、2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:SPIBUS:MODE WIRE3  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:SPIBUS:MODE? -> :TRIGGER:EINTERVAL:EVENT1:LOGIC:SPIBUS:MODE WIRE3

**:TRIGger:EINterval:EVENT<x>:LOGic:STATE?**

機能 ロジックステートトリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:STATE?  
<x> = 1、2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:STATE? -> :TRIGGER:EINTERVAL:EVENT1:LOGIC:STATE:BIT:A0 DONTCARE;  
A1 DONTCARE;A2 DONTCARE;A3 DONTCARE;  
A4 DONTCARE;A5 DONTCARE;A6 DONTCARE;  
A7 DONTCARE;B0 DONTCARE;B1 DONTCARE;  
B2 DONTCARE;B3 DONTCARE;B4 DONTCARE;  
B5 DONTCARE;B6 DONTCARE;B7 DONTCARE;  
C0 DONTCARE;C1 DONTCARE;C2 DONTCARE;  
C3 DONTCARE;C4 DONTCARE;C5 DONTCARE;  
C6 DONTCARE;C7 DONTCARE;D0 DONTCARE;  
D1 DONTCARE;D2 DONTCARE;D3 DONTCARE;  
D4 DONTCARE;D5 DONTCARE;D6 DONTCARE;  
D7 DONTCARE;LOGIC AND;;:TRIGGER:EINTERVAL:EVENT1:LOGIC:STATE:GROUP1:CONDITION DONTCARE;  
PATTERN "XXXXXXXX";:TRIGGER:EINTERVAL:EVENT1:LOGIC:STATE:GROUP2:CONDITION TRUE;PATTERN "XXXXXXXX";:TRIGGER:EINTERVAL:EVENT1:LOGIC:STATE:GROUP3:CONDITION TRUE;  
PATTERN "XXXXXXXX";:TRIGGER:EINTERVAL:EVENT1:LOGIC:STATE:GROUP4:CONDITION TRUE;PATTERN "XXXXXXXX";:TRIGGER:EINTERVAL:EVENT1:LOGIC:STATE:GROUP5:CONDITION DONTCARE;  
PATTERN ""::TRIGGER:EINTERVAL:EVENT1:LOGIC:STATE:TYPE BIT

**:TRIGger:EINterval:EVENT<x>:LOGic:STATE:BIT?**

機能 ロジックステートトリガのビットに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:STATE:BIT?  
<x> = 1、2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:STATE:BIT? -> :TRIGGER:EINTERVAL:EVENT1:LOGIC:STATE:BIT:A0 DONTCARE;  
A1 DONTCARE;A2 DONTCARE;A3 DONTCARE;  
A4 DONTCARE;A5 DONTCARE;A6 DONTCARE;  
A7 DONTCARE;B0 DONTCARE;B1 DONTCARE;  
B2 DONTCARE;B3 DONTCARE;B4 DONTCARE;  
B5 DONTCARE;B6 DONTCARE;B7 DONTCARE;  
C0 DONTCARE;C1 DONTCARE;C2 DONTCARE;  
C3 DONTCARE;C4 DONTCARE;C5 DONTCARE;  
C6 DONTCARE;C7 DONTCARE;D0 DONTCARE;  
D1 DONTCARE;D2 DONTCARE;D3 DONTCARE;  
D4 DONTCARE;D5 DONTCARE;D6 DONTCARE;  
D7 DONTCARE;LOGIC AND

**:TRIGger:EINterval:EVENT<x>:LOGic:STATE:BIT:{A<y>|B<y>|C<y>|D<y>}**

機能 ロジックステートトリガの各ビットの成立条件を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:STATE:BIT:{A<y>|B<y>|C<y>|D<y>}  
{DONTcare|HIGH|LOW}  
:TRIGger:EINterval:EVENT<x>:LOGic:STATE:BIT:{A<y>|B<y>|C<y>|D<y>}?  
<x> = 1、2  
<y> = 0 ~ 7

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:STATE:BIT:A0 DONTCARE  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:STATE:BIT:A0? -> :TRIGGER:EINTERVAL:EVENT1:LOGIC:STATE:BIT:A0 DONTCARE

解説 DL9510L、DL9505L は {A<y>|C<y>} が有効です。

**:TRIGger:EINterval:EVENT<x>:LOGic:STATE:BIT:CLEAr**

機能 ロジックステートトリガの各ビットの成立条件をすべてクリア (Don't care) します。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:STATE:BIT:CLEAr  
<x> = 1、2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:STATE:BIT:CLEAR

### 5.30 TRIGger グループ

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:STATE:BIT:LOGic**

機能 ロジックステートトリガのロジックを設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:STATE:BIT:LOGic {AND|OR}  
:TRIGger:EIInterval:EVENT<x>:LOGic:STATE:BIT:LOGic?  
<x> = 1、2

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:STATE:BIT:LOGIC AND  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:STATE:BIT:LOGIC? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:STATE:BIT:LOGIC AND

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:STATE:GROup<x>?**

機能 ロジックステートトリガの各グループに関するすべての設定値を問い合わせます。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:STATE:GROup<x>?  
EVENT<x> の <x> = 1、2  
GROup<x> の <x> = 1 ~ 5

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:STATE:GROUP1? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:STATE:GROUP1:CONDITION DONTCARE;PATTERN "11110000111100001111000011110000"

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:STATE:GROup<x>:CLEAr**

機能 ロジックステートトリガの各グループの成立条件をすべてクリア (Don't care) します。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:STATE:GROup<x>:CLEAr  
EVENT<x> の <x> = 1、2  
GROup<x> の <x> = 1 ~ 5

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:STATE:GROUP1:CLEAr

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:STATE:GROup<x>:CONDition**

機能 ロジックステートトリガの各グループの判定条件を設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:STATE:GROup<x>:CONDition {DONTcare|TRUE}  
:TRIGger:EIInterval:EVENT<x>:LOGic:STATE:GROup<x>:CONDition?

EVENT<x> の <x> = 1、2  
GROup<x> の <x> = 1 ~ 5

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:STATE:GROUP1:CONDITION DONTCARE  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:STATE:GROUP1:CONDITION? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:STATE:GROUP1:CONDITION DONTCARE

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:STATE:GROup<x>:HEXA**

機能 ロジックステートトリガの各グループの成立条件を HEXA で設定します。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:STATE:GROup<x>:HEXA {<文字列>}  
EVENT<x> の <x> = 1、2  
GROup<x> の <x> = 1 ~ 5

<文字列> = '0' ~ 'F'、'X' の組み合わせ 8 文字以内

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:STATE:GROUP1:HEXA "1A3F24CD"

解説 「:LOGic:GROup<x>:MAPPING」で設定したビット配置数が多い場合は下位に "X" が設定されます。少ない場合は上位が設定されます。

#### **:TRIGger:EIInterval:EVENT<x>:LOGic:STATE:GROup<x>:PATTern**

機能 ロジックステートトリガの各グループの成立条件を BINARY で設定 / 問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:LOGic:STATE:GROup<x>:PATTern {<文字列>}  
:TRIGger:EIInterval:EVENT<x>:LOGic:STATE:GROup<x>:PATTern?

EVENT<x> の <x> = 1、2  
GROup<x> の <x> = 1 ~ 5  
<文字列> = '0'、'1'、'X' の組み合わせ 32 文字以内

例 :TRIGGER:EIINTERVAL:EVENT1:LOGIC:STATE:GROUP1:PATTERN "111100001111000001111000011110000"  
:TRIGGER:EIINTERVAL:EVENT1:LOGIC:STATE:GROUP1:PATTERN? -> :TRIGGER:EIINTERVAL:EVENT1:LOGIC:STATE:GROUP1:PATTERN "11110000111100001111000011110000"

**:TRIGger:EINterval:EVENT<x>:LOGic:STATE:GROup<x>:SYMBOL**

機能 ロジックステートトリガの各グループのシンボルアイテムを設定します。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:STATE:GROup<x>:SYMBOL {<文字列>}  
EVENT<x> の <x> = 1, 2  
GROup<x> の <x> = 1 ~ 5  
<文字列> = 16 文字以内

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
STATE:GROUP1:SYMBOL "TEST"

**:TRIGger:EINterval:EVENT<x>:LOGic:STATE:TYPE**

機能 ロジックステートトリガの設定方法を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:STATE:TYPE {BIT|GROup}  
:TRIGger:EINterval:EVENT<x>:LOGic:STATE:TYPE?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
STATE:TYPE BIT  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:  
STATE:TYPE? -> :TRIGGER:EINTERVAL:  
EVENT1:LOGIC:STATE:TYPE BIT

**:TRIGger:EINterval:EVENT<x>:LOGic:WIDTH?**

機能 ロジックパルス幅トリガのすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:WIDTH?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
WIDTH? -> :TRIGGER:EINTERVAL:EVENT1:  
LOGIC:WIDTH:MODE OUT;  
POLARITY POSITIVE;SOURCE A0;  
TIME1 1.0000000E-09;  
TIME2 1.0000000E-09

**:TRIGger:EINterval:EVENT<x>:LOGic:WIDTH:MODE**

機能 ロジックパルス幅トリガの判定モードを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:WIDTH:MODE {BETween|IN|NOTBetween|OUT|TIMEout}  
:TRIGger:EINterval:EVENT<x>:LOGic:WIDTH:MODE?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
WIDTH:MODE BETWEEN  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:  
WIDTH:MODE? -> :TRIGGER:EINTERVAL:  
EVENT1:LOGIC:WIDTH:MODE BETWEEN

**:TRIGger:EINterval:EVENT<x>:LOGic:WIDTH:POLarity**

機能 ロジックパルス幅トリガの極性を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:WIDTH:POLarity {FALSE|NEGative|POSitive|TRUE}  
:TRIGger:EINterval:EVENT<x>:LOGic:WIDTH:POLarity?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
WIDTH:POLARITY FALSE  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:  
WIDTH:POLARITY? -> :TRIGGER:  
EINTERVAL:EVENT1:LOGIC:WIDTH:  
POLARITY FALSE

解説  
・「:TRIGger:EINterval:EVENT<x>:TYPE LPState」のときは {FALSE|TRUE} が有効です。  
・「:TRIGger:EINterval:EVENT<x>:TYPE LPULse」のときは {NEGative|POSitive} が有効です。

**:TRIGger:EINterval:EVENT<x>:LOGic:WIDTH:SOURce**

機能 ロジックパルス幅トリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:LOGic:WIDTH:SOURce {A<y>|B<y>|C<y>|D<y>}  
:TRIGger:EINterval:EVENT<x>:LOGic:WIDTH:SOURce?  
<x> = 1, 2  
<y> = 0 ~ 7

例 :TRIGGER:EINTERVAL:EVENT1:LOGIC:  
WIDTH:SOURCE A0  
:TRIGGER:EINTERVAL:EVENT1:LOGIC:  
WIDTH:SOURCE? -> :TRIGGER:EINTERVAL:  
EVENT1:LOGIC:WIDTH:SOURCE A0

解説 DL9510L、DL9505L は {A<y>|C<y>} が有効です。

### 5.30 TRIGGER グループ

#### **:TRIGGER:EVENT<x>:LOGIC:WIDTH:TIME<x>**

機能 ロジックパルス幅トリガのパルス幅を設定 / 問い合わせします。

構文 :TRIGGER:EVENT<x>:LOGIC:WIDTH:TIME<x> {<時間>}  
:TRIGGER:EVENT<x>:LOGIC:WIDTH:TIME<x>?  
EVENT<x> の <x> = 1, 2  
TIME<x> の <x> = 1, 2

例 <時間> = 1ns ~ 10s(500ps ステップ)  
:TRIGGER:EVENT1:LOGIC:WIDTH:TIME1 1S

解説 :TRIGGER:EVENT1:LOGIC:WIDTH:TIME1? -> :TRIGGER:EVENT1:EVENT1:LOGIC:WIDTH:TIME1 1.000E+00  
TIME2 は、「:TRIGGER:EVENT<x>:WIDTH:MODE BETWEEN|NOTBetween」のときに有効です。

#### **:TRIGGER:EVENT<x>:SPATTERN? (Serial Pattern)**

機能 各イベントのシリアルパターントリガに関するすべての設定値を問い合わせます。

構文 :TRIGGER:EVENT<x>:SPATTERN?  
<x> = 1, 2

例 :TRIGGER:EVENT1:SPATTERN?  
-> :TRIGGER:EVENT1:SPATTERN:BITRATE 1.000E+00;CLOCK:MODE 1;POLARITY FALL;SOURCE 1;;TRIGGER:EVENT1:SPATTERN:CS 1;DATA:ACTIVE HIGH;SOURCE 1;;TRIGGER:EVENT1:SPATTERN:LATCH:SOURCE 1;POLARITY FALL;;TRIGGER:EVENT1:SPATTERN:PATTERN "1100110111101111"

#### **:TRIGGER:EVENT<x>:SPATTERN:BITRate**

機能 シリアルパターントリガのビットレートを設定 / 問い合わせします。

構文 :TRIGGER:EVENT<x>:SPATTERN:BITRate {<NRf>}  
:TRIGGER:EVENT<x>:SPATTERN:BITRate?  
<x> = 1, 2  
<NRf> = 1 ~ 50M(bps)

例 :TRIGGER:EVENT1:SPATTERN:BITRATE 1  
:TRIGGER:EVENT1:SPATTERN:BITRATE? -> :TRIGGER:EVENT1:EVENT1:SPATTERN:BITRATE 1.000E+00

解説 「:TRIGGER:EVENT<x>:SPATTERN:CLOCK:MODE OFF」のときに有効です。

#### **:TRIGGER:EVENT<x>:SPATTERN:CLEAr**

機能 シリアルパターントリガのパターンをすべてクリア (Don't care) します。

構文 :TRIGGER:EVENT<x>:SPATTERN:CLEAr  
<x> = 1, 2

例 :TRIGGER:EVENT1:SPATTERN:CLEAr

#### **:TRIGGER:EVENT<x>:SPATTERN:CLOCK?**

機能 シリアルパターントリガのクロックに関するすべての設定値を問い合わせます。

構文 :TRIGGER:EVENT<x>:SPATTERN:CLOCK?  
<x> = 1, 2

例 :TRIGGER:EVENT1:SPATTERN:CLOCK? -> :TRIGGER:EVENT1:SPATTERN:CLOCK:MODE 1;POLARITY FALL;SOURCE 1

#### **:TRIGGER:EVENT<x>:SPATTERN:CLOCK:MODE**

機能 シリアルパターントリガのクロックの有効 / 無効を設定 / 問い合わせします。

構文 :TRIGGER:EVENT<x>:SPATTERN:CLOCK:MODE {<Boolean>}  
:TRIGGER:EVENT<x>:SPATTERN:CLOCK:MODE?  
<x> = 1, 2

例 :TRIGGER:EVENT1:SPATTERN:CLOCK:MODE ON  
:TRIGGER:EVENT1:SPATTERN:CLOCK:MODE? -> :TRIGGER:EVENT1:SPATTERN:CLOCK:MODE 1

#### **:TRIGGER:EVENT<x>:SPATTERN:CLOCK:POLarity**

機能 シリアルパターントリガのクロックトレースの極性を設定 / 問い合わせします。

構文 :TRIGGER:EVENT<x>:SPATTERN:CLOCK:POLarity {FALL|RISE}  
:TRIGGER:EVENT<x>:SPATTERN:CLOCK:POLarity?  
<x> = 1, 2

例 :TRIGGER:EVENT1:SPATTERN:CLOCK:POLARITY FALL  
:TRIGGER:EVENT1:SPATTERN:CLOCK:POLARITY? -> :TRIGGER:EVENT1:SPATTERN:CLOCK:POLARITY FALL

解説 「:TRIGGER:EVENT<x>:SPATTERN:CLOCK:MODE ON」のときに有効です。

**:TRIGger:EINterval:EVENT<x>:****SPATtern:CLOCK:SOURCE**

機能 シリアルパターントリガのクロックトレースを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:SPATtern:CLOCK:SOURCE {<NRF>}  
:TRIGger:EINterval:EVENT<x>:SPATtern:CLOCK:SOURCE?  
<x> = 1, 2  
<NRF> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:SPATTERN:CLOCK:SOURCE 1  
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:CLOCK:SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:SPATTERN:CLOCK:SOURCE 1

解説 「:TRIGger:EINterval:EVENT<x>:SPATtern:CLOCK:MODE ON」のときに有効です。

**:TRIGger:EINterval:EVENT<x>:****SPATtern:CS**

機能 シリアルパターントリガのチップセレクトの有効 / 無効を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:SPATtern:CS {<Boolean>}  
:TRIGger:EINterval:EVENT<x>:SPATtern:CS?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:SPATTERN:CS ON  
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:CS? -> :TRIGGER:EINTERVAL:EVENT1:SPATTERN:CS 1

解説 「:TRIGger:EINterval:EVENT<x>:SPATtern:CLOCK:MODE ON」のときに有効です。

**:TRIGger:EINterval:EVENT<x>:****SPATtern:DATA?**

機能 シリアルパターントリガのデータに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:SPATtern:DATA?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:SPATTERN:DATA? -> :TRIGGER:EINTERVAL:EVENT1:SPATTERN:DATA:ACTIVE HIGH;SOURCE 1

**:TRIGger:EINterval:EVENT<x>:****SPATtern:DATA:ACTIVE**

機能 シリアルパターントリガのデータのアクティブレベルを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:SPATtern:DATA:ACTIVE {HIGH|LOW}  
:TRIGger:EINterval:EVENT<x>:SPATtern:DATA:ACTIVE?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:SPATTERN:DATA:ACTIVE HIGH  
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:DATA:ACTIVE? -> :TRIGGER:EINTERVAL:EVENT1:SPATTERN:DATA:ACTIVE HIGH

**:TRIGger:EINterval:EVENT<x>:****SPATtern:DATA:SOURCE**

機能 シリアルパターントリガのデータトレースを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:SPATtern:DATA:SOURCE {<NRF>}  
:TRIGger:EINterval:EVENT<x>:SPATtern:DATA:SOURCE?  
<x> = 1, 2  
<NRF> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:SPATTERN:DATA:SOURCE 1  
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:DATA:SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:SPATTERN:DATA:SOURCE 1

**:TRIGger:EINterval:EVENT<x>:****SPATtern:HEXA**

機能 シリアルパターントリガのパターンを HEXA で設定します。

構文 :TRIGger:EINterval:EVENT<x>:SPATtern:HEXA {<文字列>}  
<x> = 1, 2  
<文字列> = '0' ~ 'F', 'X' の組み合わせ 32 文字以内。

例 :TRIGGER:EINTERVAL:EVENT1:SPATTERN:HEXA "ABCD"

**:TRIGger:EINterval:EVENT<x>:****SPATtern:LATCH?**

機能 シリアルパターントリガのラッチに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:SPATtern:LATCH?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:SPATTERN:LATCH? -> :TRIGGER:EINTERVAL:EVENT1:SPATTERN:LATCH:SOURCE 1;POLARITY FALL



### 5.30 TRIGger グループ

#### **:TRIGger:EIInterval:EVENT<x>:**

##### **SPATtern:LATCh:POLarity**

機能 シリアルパターントリガのラッチトレースの極性を設定 / 問い合わせします。

構文 `:TRIGger:EIInterval:EVENT<x>:SPATtern:LATCh:POLarity {FALL|RISE}`  
`:TRIGger:EIInterval:EVENT<x>:SPATtern:LATCh:POLarity?`  
`<x> = 1, 2`

例 `:TRIGGER:EIINTERVAL:EVENT1:SPATTERN:LATCH:POLARITY FALL`  
`:TRIGGER:EIINTERVAL:EVENT1:SPATTERN:LATCH:POLARITY? -> :TRIGGER:EIINTERVAL:EVENT1:SPATTERN:LATCH:POLARITY FALL`

解説

- 「`:TRIGger:EIInterval:EVENT<x>:SPATtern:CLOCK:MODE ON`」のときに有効です。
- 「`:TRIGger:EIInterval:EVENT<x>:SPATtern:LATCh:SOURce NONE`」のときは無効です。

#### **:TRIGger:EIInterval:EVENT<x>:**

##### **SPATtern:LATCh:SOURce**

機能 シリアルパターントリガのラッチトレースを設定 / 問い合わせします。

構文 `:TRIGger:EIInterval:EVENT<x>:SPATtern:LATCh:SOURce {<NRF>|NONE}`  
`:TRIGger:EIInterval:EVENT<x>:SPATtern:LATCh:SOURce?`  
`<x> = 1, 2`  
`<NRF> = 1 ~ 4`

例 `:TRIGGER:EIINTERVAL:EVENT1:SPATTERN:LATCH:SOURCE 1`  
`:TRIGGER:EIINTERVAL:EVENT1:SPATTERN:LATCH:SOURCE? -> :TRIGGER:EIINTERVAL:EVENT1:SPATTERN:LATCH:SOURCE 1`

解説 「`:TRIGger:EIInterval:EVENT<x>:SPATtern:CLOCK:MODE ON`」のときに有効です。

#### **:TRIGger:EIInterval:EVENT<x>:**

##### **SPATtern:PATtern**

機能 シリアルパターントリガのパターンを BINARY で設定 / 問い合わせします。

構文 `:TRIGger:EIInterval:EVENT<x>:SPATtern:PATtern {<文字列>}`  
`:TRIGger:EIInterval:EVENT<x>:SPATtern:PATtern?`  
`<x> = 1, 2`  
`<文字列> = '0','1','X'の組み合わせ 128文字以内。`

例 `:TRIGGER:EIINTERVAL:EVENT1:SPATTERN:PATTERN "1100110111101111"`  
`:TRIGGER:EIINTERVAL:EVENT1:SPATTERN:PATTERN? -> :TRIGGER:EIINTERVAL:EVENT1:SPATTERN:PATTERN "1100110111101111"`

#### **:TRIGger:EIInterval:EVENT<x>:SPIBus?**

機能 各イベントの SPI バストリガに関するすべての設定値を問い合わせます。

構文 `:TRIGger:EIInterval:EVENT<x>:SPIBus?<x> = 1, 2`

例 `:TRIGGER:EIINTERVAL:EVENT1:SPIBUS? -> :TRIGGER:EIINTERVAL:EVENT1:SPIBUS:BITORDER LSBFIRST;CLOCK:POLARITY FALL;SOURCE 1;:TRIGGER:EIINTERVAL:EVENT1:SPIBUS:CS:ACTIVE HIGH;SOURCE 1;:TRIGGER:EIINTERVAL:EVENT1:SPIBUS:DATA1:BYTE 1;CONDITION TRUE;DPOSITION 1;PATTERN1 "00010010";PATTERN2 "00110100";PATTERN3 "01010110";PATTERN4 "00010010";SOURCE 3;:TRIGGER:EIINTERVAL:EVENT1:SPIBUS:DATA2:BYTE 4;CONDITION TRUE;DPOSITION 1;PATTERN1 "00010010";PATTERN2 "00110100";PATTERN3 "01010110";PATTERN4 "00010010";SOURCE 3;:TRIGGER:EIINTERVAL:EVENT1:SPIBUS:MODE WIRE3`

#### **:TRIGger:EIInterval:EVENT<x>:SPIBus:BITOrder**

機能 SPI バストリガのビットオーダーを設定 / 問い合わせします。

構文 `:TRIGger:EIInterval:EVENT<x>:SPIBus:BITOrder {LSBFirst|MSBFirst}`  
`:TRIGger:EIInterval:EVENT<x>:SPIBus:BITOrder?`  
`<x> = 1, 2`

例 `:TRIGGER:EIINTERVAL:EVENT1:SPIBUS:BITORDER LSBFIRST`  
`:TRIGGER:EIINTERVAL:EVENT1:SPIBUS:BITORDER? -> :TRIGGER:EIINTERVAL:EVENT1:SPIBUS:BITORDER LSBFIRST`

#### **:TRIGger:EIInterval:EVENT<x>:SPIBus:CLOCK?**

機能 SPI バストリガのクロックに関するすべての設定値を問い合わせます。

構文 `:TRIGger:EIInterval:EVENT<x>:SPIBus:CLOCK?`  
`<x> = 1, 2`

例 `:TRIGGER:EIINTERVAL:EVENT1:SPIBUS:CLOCK? -> :TRIGGER:EIINTERVAL:EVENT1:SPIBUS:CLOCK:POLARITY FALL;SOURCE 1`

**:TRIGger:EINterval:EVENT<x>:SPIBus:  
CLOCK:POLarity**

機能 SPIバストリガのクロックトレースの極性を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:SPIBus:  
CLOCK:POLarity {FALL|RISE}  
:TRIGger:EINterval:EVENT<x>:SPIBus:  
CLOCK:POLarity?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:SPIBUS:  
CLOCK:POLARITY FALL  
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:  
CLOCK:POLARITY? -> :TRIGGER:  
EINTERVAL:EVENT1:SPIBUS:CLOCK:  
POLARITY FALL

**:TRIGger:EINterval:EVENT<x>:SPIBus:  
CLOCK:SOURce**

機能 SPIバストリガのクロックトレースを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:SPIBus:  
CLOCK:SOURce {<Nrf>}  
:TRIGger:EINterval:EVENT<x>:SPIBus:  
CLOCK:SOURce?  
<x> = 1, 2  
<Nrf> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:SPIBUS:  
CLOCK:SOURCE 1  
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:  
CLOCK:SOURCE? -> :TRIGGER:EINTERVAL:  
EVENT1:SPIBUS:CLOCK:SOURCE 1

**:TRIGger:EINterval:EVENT<x>:SPIBus:  
CS?**

機能 SPIバストリガのチップセレクトに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:SPIBus:  
CS?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS?  
-> :TRIGGER:EINTERVAL:EVENT1:SPIBUS:  
CS:ACTIVE HIGH;SOURCE 1

**:TRIGger:EINterval:EVENT<x>:SPIBus:  
CS:ACTive**

機能 SPIバストリガのチップセレクトのアクティブレベルを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:SPIBus:  
CS:ACTive {HIGH|LOW}  
:TRIGger:EINterval:EVENT<x>:SPIBus:  
CS:ACTive?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS:  
ACTIVE HIGH  
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS:  
ACTIVE? -> :TRIGGER:EINTERVAL:EVENT1:  
SPIBUS:CS:ACTIVE HIGH

**:TRIGger:EINterval:EVENT<x>:SPIBus:  
CS:SOURce**

機能 SPIバストリガのチップセレクトトレースを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:SPIBus:  
CS:SOURce {<Nrf>}  
:TRIGger:EINterval:EVENT<x>:SPIBus:  
CS:SOURce?  
<x> = 1, 2  
<Nrf> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS:  
SOURCE 1  
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS:  
SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:  
SPIBUS:CS:SOURCE 1

**:TRIGger:EINterval:EVENT<x>:SPIBus:  
DATA<x>?**

機能 SPIバストリガの各データに関するすべての設定値を問い合わせます。

構文 :TRIGger:EINterval:EVENT<x>:SPIBus:  
DATA<x>?  
EVENT<x> の <x> = 1, 2  
DATA<x> の <x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:SPIBUS:  
DATA1? -> :TRIGGER:EINTERVAL:EVENT1:  
SPIBUS:DATA1:BYTE 1;CONDITION TRUE;  
DPOSITION 1;PATTERN1 "00010010";  
PATTERN2 "00110100";  
PATTERN3 "01010110";  
PATTERN4 "00010010";SOURCE 3

解説 DATA2 は、「:TRIGger:EINterval:  
EVENT<x>:SPIBus:MODE WIRE4」のときに有効です。

**:TRIGger:EINterval:EVENT<x>:SPIBus:  
DATA<x>:BYTE**

機能 SPIバストリガの各データの設定データ数を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:SPIBus:  
DATA<x>:BYTE {<Nrf>}  
:TRIGger:EINterval:EVENT<x>:SPIBus:  
DATA<x>:BYTE?  
EVENT<x> の <x> = 1, 2  
DATA<x> の <x> = 1, 2  
<Nrf> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:SPIBUS:  
DATA1:BYTE 1  
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:  
DATA1:BYTE? -> :TRIGGER:EINTERVAL:  
EVENT1:SPIBUS:DATA1:BYTE 1

### 5.30 TRIGger グループ

#### **:TRIGger:EIInterval:EVENT<x>:SPIBUS:DATA<x>:CONDition**

機能 SPIバストリガの各データの判定方法(一致/不一致)を設定/問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:SPIBUS:DATA<x>:CONDition {FALSE|TRUE}  
:TRIGger:EIInterval:EVENT<x>:SPIBUS:DATA<x>:CONDition?  
EVENT<x>の<x> = 1、2  
DATA<x>の<x> = 1、2

例 :TRIGGER:EIINTERVAL:EVENT1:SPIBUS:DATA1:CONDITION TRUE  
:TRIGGER:EIINTERVAL:EVENT1:SPIBUS:DATA1:CONDITION? -> :TRIGGER:EIINTERVAL:EVENT1:SPIBUS:DATA1:CONDITION TRUE

#### **:TRIGger:EIInterval:EVENT<x>:SPIBUS:DATA<x>:DPOSITION**

機能 SPIバストリガの各データのパターン比較先頭位置を設定/問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:SPIBUS:DATA<x>:DPOSITION {<NRf>}  
:TRIGger:EIInterval:EVENT<x>:SPIBUS:DATA<x>:DPOSITION?  
EVENT<x>の<x> = 1、2  
DATA<x>の<x> = 1、2  
<NRf> = 0 ~ 9999

例 :TRIGGER:EIINTERVAL:EVENT1:SPIBUS:DATA1:DPOSITION 1  
:TRIGGER:EIINTERVAL:EVENT1:SPIBUS:DATA1:DPOSITION? -> :TRIGGER:EIINTERVAL:EVENT1:SPIBUS:DATA1:DPOSITION 1

#### **:TRIGger:EIInterval:EVENT<x>:SPIBUS:DATA<x>:HEXA<x>**

機能 SPIバストリガの各データをHEXAで設定します。

構文 :TRIGger:EIInterval:EVENT<x>:SPIBUS:DATA<x>:HEXA<x> {<文字列>}  
EVENT<x>の<x> = 1、2  
DATA<x>の<x> = 1、2  
HEXA<x>の<x> = 1 ~ 4  
<文字列> = '0' ~ 'F'、'X'の組み合わせ2文字。

例 :TRIGGER:EIINTERVAL:EVENT1:SPIBUS:DATA1:HEXA1 "AB"

#### **:TRIGger:EIInterval:EVENT<x>:SPIBUS:DATA<x>:PATTERN<x>**

機能 SPIバストリガの各データをBINARYで設定/問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:SPIBUS:DATA<x>:PATTERN<x> {<文字列>}  
:TRIGger:EIInterval:EVENT<x>:SPIBUS:DATA<x>:PATTERN<x>?  
EVENT<x>の<x> = 1、2  
DATA<x>の<x> = 1、2  
PATTERN<x>の<x> = 1 ~ 4  
<文字列> = '0'、'1'、'X'の組み合わせ8文字。

例 :TRIGGER:EIINTERVAL:EVENT1:SPIBUS:DATA1:PATTERN1 "10101011"  
:TRIGGER:EIINTERVAL:EVENT1:SPIBUS:DATA1:PATTERN1? -> :TRIGGER:EIINTERVAL:EVENT1:SPIBUS:DATA1:PATTERN1 "10101011"

#### **:TRIGger:EIInterval:EVENT<x>:SPIBUS:DATA<x>:SOURCE**

機能 SPIバストリガの各データのトレースを設定/問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:SPIBUS:DATA<x>:SOURCE {<NRf>}  
:TRIGger:EIInterval:EVENT<x>:SPIBUS:DATA<x>:SOURCE?  
EVENT<x>の<x> = 1、2  
DATA<x>の<x> = 1、2  
<NRf> = 1 ~ 4

例 :TRIGGER:EIINTERVAL:EVENT1:SPIBUS:DATA1:SOURCE 1  
:TRIGGER:EIINTERVAL:EVENT1:SPIBUS:DATA1:SOURCE? -> :TRIGGER:EIINTERVAL:EVENT1:SPIBUS:DATA1:SOURCE 1

#### **:TRIGger:EIInterval:EVENT<x>:SPIBUS:MODE**

機能 SPIバストリガの結線方式(3線式/4線式)を設定/問い合わせします。

構文 :TRIGger:EIInterval:EVENT<x>:SPIBUS:MODE {WIRE3|WIRE4}  
:TRIGger:EIInterval:EVENT<x>:SPIBUS:MODE?  
<x> = 1、2

例 :TRIGGER:EIINTERVAL:EVENT1:SPIBUS:MODE WIRE3  
:TRIGGER:EIINTERVAL:EVENT1:SPIBUS:MODE? -> :TRIGGER:EIINTERVAL:EVENT1:SPIBUS:MODE WIRE3

**:TRIGger:EINterval:EVENT<x>:STATE?**  
 機能 各イベントの成立条件に関するすべての設定値を問い合わせます。  
 構文 :TRIGger:EINterval:EVENT<x>:STATE?  
 <x> = 1, 2  
 例 :TRIGGER:EINTERVAL:EVENT1:STATE?  
 -> :TRIGGER:EINTERVAL:EVENT1:STATE:  
 CHANNEL1 DONTCARE;CHANNEL2 DONTCARE;  
 CHANNEL3 DONTCARE;CHANNEL4 DONTCARE;  
 LOGIC AND

**:TRIGger:EINterval:EVENT<x>:STATE:CHANnel<x>**  
 機能 各チャンネルの成立条件を設定 / 問い合わせします。  
 構文 :TRIGger:EINterval:EVENT<x>:STATE:  
 CHANnel<x> {DONTcare|HIGH|IN|LOW|OUT}  
 :TRIGger:EINterval:EVENT<x>:STATE:  
 CHANnel<x>?  
 EVENT<x> の <x> = 1, 2  
 CHANnel<x> の <x> = 1 ~ 4  
 例 :TRIGGER:EINTERVAL:EVENT1:STATE:  
 CHANNEL1 HIGH  
 :TRIGGER:EINTERVAL:EVENT1:STATE:  
 CHANNEL1? -> :TRIGGER:EINTERVAL:  
 EVENT1:STATE:CHANNEL1 HIGH  
 解説  
 ・「:TRIGger:EINterval:EVENT<x>:TYPE  
 EQUalify|I2Cbus|PQUalify|PSTATE|  
 SPATtern|STATE」のときに有効です。  
 ・「:TRIGger:EINterval:EVENT<x>:TYPE  
 I2Cbus|SPATtern」のときに {HIGH|LOW}  
 が有効です。  
 ・「:TRIGger:EINterval:EVENT<x>:TYPE  
 EQUalify|PQUalify|PSTATE|STATE」  
 且つ「TRIGger:SOURce:CHANnel<x>:  
 WINDow ON」のときは {IN|OUT}、「TRIGger:  
 SOURce:CHANnel<x>:WINDow OFF」のとき  
 は {HIGH|LOW} が有効です。

**:TRIGger:EINterval:EVENT<x>:STATE:LOGic**  
 機能 成立条件のロジックを設定 / 問い合わせします。  
 構文 :TRIGger:EINterval:EVENT<x>:STATE:  
 LOGic {AND|OR}  
 :TRIGger:EINterval:EVENT<x>:STATE:  
 LOGic?  
 <x> = 1, 2  
 例 :TRIGGER:EINTERVAL:EVENT1:STATE:  
 LOGIC AND  
 :TRIGGER:EINTERVAL:EVENT1:STATE:  
 LOGIC? -> :TRIGGER:EINTERVAL:EVENT1:  
 STATE:LOGIC AND  
 解説 「:TRIGger:EINterval:EVENT<x>:TYPE  
 EQUalify|I2Cbus|PQUalify|PSTATE|  
 SPATtern|STATE」のときに有効です。

**:TRIGger:EINterval:EVENT<x>:TYPE**  
 機能 各イベントのトリガの種類を設定 / 問い合わせします。  
 構文 :TRIGger:EINterval:EVENT<x>:TYPE  
 {CANbus|EDGE|EQUalify|I2Cbus|LEDge|  
 LINbus|LI2Cbus|LLINbus|LSPATtern|  
 LSPiBus|LPSTATE|LPULse|LQUalify|  
 LSTATE|PQUalify|PSTATE|PULse|  
 SPATtern|SPIbus|STATE}  
 :TRIGger:EINterval:EVENT<x>:TYPE?  
 <x> = 1, 2  
 例 :TRIGGER:EINTERVAL:EVENT1:TYPE CANBUS  
 :TRIGGER:EINTERVAL:EVENT1:TYPE?  
 -> :TRIGGER:EINTERVAL:EVENT1:  
 TYPE CANBUS

**:TRIGger:EINterval:EVENT<x>:WIDTh?**  
 機能 各イベントのパルス幅トリガのすべての設定値を問い合わせます。  
 構文 :TRIGger:EINterval:EVENT<x>:WIDTh?  
 <x> = 1, 2  
 例 :TRIGGER:EINTERVAL:EVENT1:WIDTh?  
 -> :TRIGGER:EINTERVAL:EVENT1:WIDTh:  
 MODE TIMEOUT;POLARITY POSITIVE;  
 SOURCE EXTERNAL;TIME1 1.000E+00;  
 TIME2 1.000E+00

### 5.30 TRIGger グループ

#### **:TRIGger:EINterval:EVENT<x>:WIDTh:**

##### **MODE**

機能 パルス幅トリガの判定モードを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:WIDTh:MODE {BETween|IN|NOTBetween|OUT|TIMEout}

:TRIGger:EINterval:EVENT<x>:WIDTh:MODE?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:WIDTH:MODE TIMEOUT  
:TRIGGER:EINTERVAL:EVENT1:WIDTH:MODE?  
-> :TRIGGER:EINTERVAL:EVENT1:WIDTH:MODE TIMEOUT

#### **:TRIGger:EINterval:EVENT<x>:WIDTh:**

##### **POLarity**

機能 パルス幅トリガの極性を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:WIDTh:POLarity {FALSe|IN|NEGative|OUT|POSitive|TRUE}

:TRIGger:EINterval:EVENT<x>:WIDTh:POLarity?  
<x> = 1, 2

例 :TRIGGER:EINTERVAL:EVENT1:WIDTH:POLARITY POSITIVE  
:TRIGGER:EINTERVAL:EVENT1:WIDTH:POLARITY? -> :TRIGGER:EINTERVAL:EVENT1:WIDTH:POLARITY POSITIVE

解説  
・「:TRIGger:EINterval:EVENT<x>:TYPE PQQualify|PULSe」且つ「:TRIGger:SOURce:CHANnel<x>:WINDow ON」のときは {IN|OUT}、「:TRIGger:SOURce:CHANnel<x>:WINDow OFF」のときは {Positive|NEGative} が有効です。  
・「:TRIGger:EINterval:EVENT<x>:TYPE PSTate」のときは {FALSe|TRUE} が有効です。

#### **:TRIGger:EINterval:EVENT<x>:WIDTh:**

##### **SOURce**

機能 パルス幅トリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:WIDTh:SOURce {<Nrf>|EXtErnal}

:TRIGger:EINterval:EVENT<x>:WIDTh:SOURce?  
<x> = 1, 2  
<Nrf> = 1 ~ 4

例 :TRIGGER:EINTERVAL:EVENT1:WIDTH:SOURCE EXTERNAL  
:TRIGGER:EINTERVAL:EVENT1:WIDTH:SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:WIDTH:SOURCE EXTERNAL

解説 「:TRIGger:EINterval:EVENT<x>:TYPE PQQualify|PULSe」のときに有効です。

#### **:TRIGger:EINterval:EVENT<x>:WIDTh:**

##### **TIME<x>**

機能 パルス幅トリガのパルス幅を設定 / 問い合わせします。

構文 :TRIGger:EINterval:EVENT<x>:WIDTh:TIME<x> {<時間>}

:TRIGger:EINterval:EVENT<x>:WIDTh:TIME<x>?  
EVENT<x> の <x> = 1, 2  
TIME<x> の <x> = 1, 2  
<時間> = 1ns ~ 10s(500ps ステップ)

例 :TRIGGER:EINTERVAL:EVENT1:WIDTH:TIME1 1S  
:TRIGGER:EINTERVAL:EVENT1:WIDTH:TIME1? -> :TRIGGER:EINTERVAL:EVENT1:WIDTH:TIME1 1.000E+00

解説 TIME2 は、「:TRIGger:EINterval:EVENT<x>:WIDTh:MODE BETWEEen|NOTBetween」のときに有効です。

#### **:TRIGger:EINterval:MODE**

機能 イベントインターバルの判定モードを設定 / 問い合わせします。

構文 :TRIGger:EINterval:MODE {BETWEEen|IN|NOTBetween|OUT|TIMEout}

:TRIGger:EINterval:MODE?  
例 :TRIGGER:EINTERVAL:MODE BETWEEN  
:TRIGGER:EINTERVAL:MODE? -> :TRIGGER:EINTERVAL:MODE BETWEEN

#### **:TRIGger:EINterval:TIME<x>**

機能 イベントインターバルのインターバル時間を設定 / 問い合わせします。

構文 :TRIGger:EINterval:TIME<x> {<時間>}

:TRIGger:EINterval:TIME<x>?  
<x> = 1, 2  
<時間> = 1.5ns ~ 10s(500ps ステップ)

例 :TRIGGER:EINTERVAL:TIME1 1S  
:TRIGGER:EINTERVAL:TIME1?  
-> :TRIGGER:EINTERVAL:TIME1 1.000E+00

解説  
・TIME2 は、「:TRIGger:EINterval:MODE BETWEEen|NOTBetween」のときに有効です。  
・:TRIGger:TYPE コマンドで EIDelay または EISequence を設定し、:TRIGger:EINterval:EVENT<x>:TYPE コマンドで EVENT1(イベント1)と EVENT2(イベント2)に、アナログ信号トリガとロジック信号トリガを混在して設定した場合、インターバル時間の最小設定値は 1.5ns から 20ns になります。

#### **:TRIGger:EINterval:TRY?**

機能 イベントインターバルのテストに関するすべての設定値を問い合わせします。

構文 :TRIGger:EINterval:TRY?

例 :TRIGGER:EINTERVAL:TRY? -> :TRIGGER:EINTERVAL:TRY:MODE 0;SELECT 1

**:TRIGger:EINterval:TRY:MODE**

機能 テストモードを設定 / 問い合わせします。

構文 :TRIGger:EINterval:TRY:MODE

{<Boolean>}

:TRIGger:EINterval:TRY:MODE?

例 :TRIGGER:EINTERVAL:TRY:MODE ON

:TRIGGER:EINTERVAL:TRY:MODE?

-> :TRIGGER:EINTERVAL:TRY:MODE 1

解説 「:TRIGger:EINterval:MODE BETWEEN|NOTBetween」 のときに有効です。

**:TRIGger:EINterval:TRY:SElect**

機能 テストモードの対象イベントを設定 / 問い合わせします。

構文 :TRIGger:EINterval:TRY:SElect {<Nrf>}

:TRIGger:EINterval:TRY:SElect?

<Nrf> = 1、2

例 :TRIGGER:EINTERVAL:TRY:SELECT 1

:TRIGGER:EINTERVAL:TRY:SELECT?

-> :TRIGGER:EINTERVAL:TRY:SELECT 1

解説 「:TRIGger:EINterval:MODE BETWEEN|NOTBetween」 のときに有効です。

**:TRIGger:ENHanced?**

機能 Enhanced トリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced?

例 :TRIGGER:ENHANCED? -> :TRIGGER:

ENHANCED:I2CBUS:ADATA:

BIT10ADDRESS:PATTERN "00011111101";:

TRIGGER:ENHANCED:I2CBUS:ADATA:

BIT7ADDRESS:PATTERN "11011110";:

TRIGGER:ENHANCED:I2CBUS:ADATA:

BIT7APSUB:ADDRESS:

PATTERN "11001101";:TRIGGER:ENHANCED:

I2CBUS:ADATA:BIT7APSUB:SADDRESS:

PATTERN "11101111";:TRIGGER:

ENHANCED:I2CBUS:ADATA:

TYPE BIT10ADDRESS;:TRIGGER:ENHANCED:

I2CBUS:CLOCK:SOURCE 1;:TRIGGER:

ENHANCED:I2CBUS:DATA:BYTE 1;

CONDITION TRUE;DPOSITION 1;

MODE 1;PATTERN1 "10101011";

PATTERN2 "10001011";

PATTERN3 "00101011";

PATTERN4 "10101011";PMODE DONTCARE;

SOURCE 1;:TRIGGER:ENHANCED:I2CBUS:

GCALL:BIT7MADDRESS:

PATTERN "1010101";:

TRIGGER:ENHANCED:I2CBUS:GCALL:

SBYTE BIT7MADDRESS;:TRIGGER:ENHANCED:

I2CBUS:MODE ADATA;NAIGNORE:HSMODE 1;

RACCESS 1;SBYTE 1;:TRIGGER:ENHANCED:

I2CBUS:SBHSMODE:TYPE HSMODE;:TRIGGER:

ENHANCED:SPATTERN:BITRATE 1.000E+00;

CLOCK:MODE 1;POLARITY FALL;SOURCE 1;:

TRIGGER:ENHANCED:SPATTERN:CS 1;DATA:

ACTIVE HIGH;SOURCE 1;:TRIGGER:

ENHANCED:SPATTERN:LATCH:SOURCE 1;

POLARITY FALL;:TRIGGER:ENHANCED:

SPATTERN:PATTERN "1100110111101111";:

TRIGGER:ENHANCED:SPIBUS:

BITORDER MSBFIRST;CLOCK:

POLARITY FALL;SOURCE 1;:TRIGGER:

ENHANCED:SPIBUS:CS:ACTIVE HIGH;

SOURCE 1;:TRIGGER:ENHANCED:SPIBUS:

DATA1:BYTE 1;CONDITION TRUE;

DPOSITION 0;PATTERN1 "00010010";

PATTERN2 "00110100";

PATTERN3 "01010110";

PATTERN4 "00010010";SOURCE 3;:

TRIGGER:ENHANCED:SPIBUS:DATA2:BYTE 1;

CONDITION TRUE;DPOSITION 0;

PATTERN1 "00010010";

PATTERN2 "00110100";

PATTERN3 "01010110";

PATTERN4 "00010010";SOURCE DONTCARE;:

TRIGGER:ENHANCED:SPIBUS:

MODE WIRE3.....

### 5.30 TRIGger グループ

#### :TRIGger:ENHanced:CANBus?

機能 CAN バス信号トリガに関するすべての設定値を  
問い合わせます。

構文 :TRIGger:ENHanced:CANBus?

例 :TRIGGER:ENHANCED:CANBUS?  
-> :TRIGGER:ENHANCED:CANBUS:  
ACK DONTCARE;BRATE 1000000;DATA:  
BORDER BIG;CONDITION DONTCARE;  
DATA1 0.000000E+00;  
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;  
PATTERN "11100101011001000111100010  
01001100101010001000010001111111111  
010";SIGN UNSIGN;:TRIGGER:ENHANCED:  
CANBUS:IDEXT:PATTERN "XXXXXXXXXXXXXXXX  
XXXXXXXXXXXXXXXX";:TRIGGER:ENHANCED:  
CANBUS:IDOR:ID1:ACK DONTCARE;DATA:  
BORDER BIG;CONDITION DONTCARE;  
DATA1 0.000000E+00;  
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;  
PATTERN "00000001001000110100010101  
100111000100110101011110011011101  
111";SIGN UNSIGN;:TRIGGER:ENHANCED:  
CANBUS:IDOR:ID1:FORMAT STD;IDEXT:  
PATTERN "11010101111001101110111000  
0";:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:  
IDSTD:PATTERN "00100100011";:TRIGGER:  
ENHANCED:CANBUS:IDOR:ID1:MODE 0;  
RTR DATA;:TRIGGER:ENHANCED:CANBUS:  
IDOR:ID2:ACK DONTCARE;DATA:  
BORDER BIG;CONDITION DONTCARE;  
DATA1 0.000000E+00;  
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;  
PATTERN "11111110110111001011101010  
01100001110110010101000011001000010  
000";SIGN UNSIGN;:TRIGGER:ENHANCED:  
CANBUS:IDOR:ID2:FORMAT STD;IDEXT:  
PATTERN "1001000110100010101100111100  
0";:TRIGGER:ENHANCED:CANBUS:IDOR:ID2:  
IDSTD:PATTERN "10001010110";:TRIGGER:  
ENHANCED:CANBUS:IDOR:ID2:MODE 0.....

#### :TRIGger:ENHanced:CANBus:ACK

機能 CAN バス信号トリガの ACK 条件を設定 / 問い合  
合わせします。

構文 :TRIGger:ENHanced:CANBus:ACK {ACK|  
ACKBoth|DONTcare|NONack}

例 :TRIGger:ENHanced:CANBus:ACK?  
:TRIGGER:ENHANCED:CANBUS:ACK ACK  
:TRIGGER:ENHANCED:CANBUS:ACK?  
-> :TRIGGER:ENHANCED:CANBUS:ACK ACK

#### :TRIGger:ENHanced:CANBus:BRATe

機能 CAN バス信号トリガのビットレート (データ転  
送速度) を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:BRATe  
{<NRf>|USER,<NRf>}  
:TRIGger:ENHanced:CANBus:BRATe?  
<NRf> = 33300、83300、125000、250000、  
500000、1000000  
USERの<NRf> = 本体ユーザーズマニュアル参照。

例 :TRIGGER:ENHANCED:CANBUS:BRATE 83300  
:TRIGGER:ENHANCED:CANBUS:BRATE?  
-> :TRIGGER:ENHANCED:CANBUS:  
BRATE 83300

#### :TRIGger:ENHanced:CANBus:DATA?

機能 CAN バス信号トリガのデータに関するすべての  
設定値を問い合わせます。

構文 :TRIGger:ENHanced:CANBus:DATA?

例 :TRIGGER:ENHANCED:CANBUS:DATA?  
-> :TRIGGER:ENHANCED:CANBUS:DATA:  
BORDER BIG;CONDITION DONTCARE;  
DATA1 0.000000E+00;  
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;  
PATTERN "1110010101100100011110001001  
00110010101000100001000111111111010"  
;SIGN UNSIGN

#### :TRIGger:ENHanced:CANBus:DATA:Border

機能 CAN バス信号トリガのデータのバイトオーダを  
設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:DATA:Border  
{BIG|LITtle}  
:TRIGger:ENHanced:CANBus:DATA:Border?  
例 :TRIGGER:ENHANCED:CANBUS:DATA:  
BORDER BIG:TRIGGER:ENHANCED:CANBUS:  
DATA:Border? -> :TRIGGER:ENHANCED:  
CANBUS:DATA:Border BIG

#### :TRIGger:ENHanced:CANBus:DATA: CONDition

機能 CAN バス信号トリガのデータ条件を設定 / 問い  
合わせします。

構文 :TRIGger:ENHanced:CANBus:DATA:  
CONDition {BETWeen|DONTcare|FALSe|  
GTHan|LTHan|ORANge|TRUE}  
:TRIGger:ENHanced:CANBus:DATA:  
CONDition?  
例 :TRIGGER:ENHANCED:CANBUS:DATA:  
CONDITION BETWEEN  
:TRIGGER:ENHANCED:CANBUS:DATA:  
CONDITION? -> :TRIGGER:ENHANCED:  
CANBUS:DATA:CONDITION BETWEEN

**:TRIGger:ENHanced:CANBus:DATA:DATA<x>**

機能 CAN バス信号トリガのデータの比較データを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:DATA:DATA<x> {<NRf>}  
:TRIGger:ENHanced:CANBus:DATA:DATA<x>?

例 <x> = 1, 2  
<NRf> = 本体ユーザーズマニュアル参照。  
:TRIGGER:ENHANCED:CANBUS:DATA:DATA1:TRIGGER:ENHANCED:CANBUS:DATA:DATA1? -> :TRIGGER:ENHANCED:CANBUS:DATA:DATA1 1.0000000E+00

解説

- 「:TRIGger:ENHanced:CANBus:DATA:Condition GTHan」のときは「:TRIGger:ENHanced:CANBus:DATA:DATA1」で設定します。
- 「:TRIGger:ENHanced:CANBus:DATA:Condition LTHan」のときは「:TRIGger:ENHanced:CANBus:DATA:DATA2」で設定します。
- 「:TRIGger:ENHanced:CANBus:DATA:Condition BETWEEN|ORANGE」のときは、小さい値を「:TRIGger:ENHanced:CANBus:DATA:DATA1」、大きい値を「:TRIGger:ENHanced:CANBus:DATA:DATA2」で設定します。

**:TRIGger:ENHanced:CANBus:DATA:DLC**

機能 CAN バス信号トリガのデータの有効バイト数 (DLC) を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:DATA:DLC {<NRf>}  
:TRIGger:ENHanced:CANBus:DATA:DLC?<NRf> = 0 ~ 8

例 :TRIGGER:ENHANCED:CANBUS:DATA:DLC 0  
:TRIGGER:ENHANCED:CANBUS:DATA:DLC? -> :TRIGGER:ENHANCED:CANBUS:DATA:DLC 0

**:TRIGger:ENHanced:CANBus:DATA:HEXA**

機能 CAN バス信号トリガのデータを HEXA で設定します。

構文 :TRIGger:ENHanced:CANBus:DATA:HEXA {<文字列>}  
<文字列> = '0' ~ 'F'、'X' の組み合わせ 16 文字以内 (1 バイト単位)

例 :TRIGGER:ENHANCED:CANBUS:DATA:HEXA "A9"

**:TRIGger:ENHanced:CANBus:DATA:MSBLSb**

機能 CAN バス信号トリガのデータの MSB/LSB のビットを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:DATA:MSBLSb {<NRf>,<NRf>}  
:TRIGger:ENHanced:CANBus:DATA:MSBLSb?<NRf> = 本体ユーザーズマニュアル参照。

例 :TRIGGER:ENHANCED:CANBUS:DATA:MSBLSB 1,0  
:TRIGGER:ENHANCED:CANBUS:DATA:MSBLSB? -> :TRIGGER:ENHANCED:CANBUS:DATA:MSBLSB 1,0

**:TRIGger:ENHanced:CANBus:DATA:PATtern**

機能 CAN バス信号トリガのデータを BINARY で設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:DATA:PATtern {<文字列>}  
:TRIGger:ENHanced:CANBus:DATA:PATtern?

例 <文字列> = '0'、'1'、'X' の組み合わせ 64 文字以内 (1 バイト単位)  
:TRIGGER:ENHANCED:CANBUS:DATA:PATTERN "11011111"  
:TRIGGER:ENHANCED:CANBUS:DATA:PATTERN? -> :TRIGGER:ENHANCED:CANBUS:DATA:PATTERN "11011111"

**:TRIGger:ENHanced:CANBus:DATA:SIGN**

機能 CAN バス信号トリガのデータの符号を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:DATA:SIGN {SIGN|UNSign}

例 :TRIGGER:ENHANCED:CANBUS:DATA:SIGN SIGN  
:TRIGGER:ENHANCED:CANBUS:DATA:SIGN? -> :TRIGGER:ENHANCED:CANBUS:DATA:SIGN SIGN

**:TRIGger:ENHanced:CANBus:IDEXT?**

機能 CAN バス信号トリガの拡張フォーマットの ID に関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:CANBus:IDEXT?  
例 :TRIGGER:ENHANCED:CANBUS:IDEXT? -> :TRIGGER:ENHANCED:CANBUS:IDEXT:PATTERN "11001011011100001110111011111"



## 5.30 TRIGger グループ

### **:TRIGger:ENHanced:CANBus:IDEXt:HEXA**

機能 CAN バス信号トリガの拡張フォーマットの ID を HEXA で設定します。

構文 :TRIGger:ENHanced:CANBus:IDEXt:  
HEXA {<文字列>}  
<文字列> = '0' ~ 'F'、'X' の組み合わせ 8 文字

例 :TRIGGER:ENHANCED:CANBUS:IDEXT:  
HEXA "1AEF5906"

### **:TRIGger:ENHanced:CANBus:IDEXt: PATTERn**

機能 CAN バス信号トリガの拡張フォーマットの ID を BINARY で設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:IDEXt:  
PATTERn {<文字列>}  
:TRIGger:ENHanced:CANBus:IDEXt:  
PATTERn?  
<文字列> = '0'、'1'、'X' の組み合わせ 29 文字  
例 :TRIGGER:ENHANCED:CANBUS:IDEXT:  
PATTERN "1100101101110000111011101111  
1"  
:TRIGGER:ENHANCED:CANBUS:IDEXT:  
PATTERN? -> :TRIGGER:ENHANCED:CANBUS:  
IDEXT:PATTERN "1100101101110000111011  
1011111"

### **:TRIGger:ENHanced:CANBus:IDOR?**

機能 CAN バス信号トリガの OR 条件に関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:CANBus:IDOR?  
例 :TRIGGER:ENHANCED:CANBUS:IDOR:  
-> :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:  
ACK DONTCARE;DATA:BORDER BIG;  
CONDITION DONTCARE;  
DATA1 0.0000000E+00;  
DATA2 255.000000E+00;DLC 8;MSBLSB 7,0;  
PATTERN "00000001001000110100010101  
10011110001001101010111100110111101  
111";SIGN UNSIGN;:TRIGGER:ENHANCED:  
CANBUS:IDOR:ID1:FORMAT STD;IDEXT:  
PATTERN "1101010111100110111101111000  
0";:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:  
IDSTD:PATTERN "00100100011";:TRIGGER:  
ENHANCED:CANBUS:IDOR:ID1:MODE 0;  
RTR DATA;:TRIGGER:ENHANCED:CANBUS:  
IDOR:ID2:ACK DONTCARE;DATA:  
BORDER BIG;CONDITION DONTCARE;  
DATA1 0.0000000E+00;  
DATA2 255.000000E+00;DLC 8;MSBLSB 7,0;  
PATTERN "11111110110111001011101010  
01100001110110010101000011001000010  
000";SIGN UNSIGN;:TRIGGER:ENHANCED:  
CANBUS:IDOR:ID2:FORMAT STD;IDEXT:  
PATTERN "1001000110100010101100111100  
0";:TRIGGER:ENHANCED:CANBUS:IDOR:ID2:  
IDSTD:PATTERN "10001010110";:TRIGGER:  
ENHANCED:CANBUS:IDOR:ID2:MODE 0;  
RTR DATA;:TRIGGER:ENHANCED:CANBUS:  
IDOR:ID3:ACK DONTCARE;DATA:  
BORDER BIG;CONDITION DONTCARE;  
DATA1 0.0000000E+00;  
DATA2 255.000000E+00;DLC 8.....

### **:TRIGger:ENHanced:CANBus:IDOR:ID<x>?**

機能 CAN バス信号トリガの OR 条件の各 ID に関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:CANBus:IDOR:ID<x>?  
<x> = 1 ~ 4

例 :TRIGGER:ENHANCED:CANBUS:IDOR:ID1? ->  
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:  
ACK DONTCARE;DATA:BORDER BIG;  
CONDITION DONTCARE;  
DATA1 0.0000000E+00;  
DATA2 255.000000E+00;DLC 8;MSBLSB 7,0;  
PATTERN "00000001001000110100010101  
10011110001001101010111100110111101  
111";SIGN UNSIGN;:TRIGGER:ENHANCED:  
CANBUS:IDOR:ID1:FORMAT STD;IDEXT:  
PATTERN "1101010111100110111101111000  
0";:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:  
IDSTD:PATTERN "00100100011";:TRIGGER:  
ENHANCED:CANBUS:IDOR:ID1:MODE 0;  
RTR DATA

**:TRIGger:ENHanced:CANBus:IDOR:ID<x>:****ACK**

機能 CAN バス信号トリガの OR 条件の各 ACK 条件を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:IDOR:ID<x>:  
ACK {ACK|ACKBoth|DONTcare|NONack}  
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:  
ACK?  
<x> = 1 ~ 4

例 :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:  
ACK ACK  
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:  
ACK? -> :TRIGGER:ENHANCED:CANBUS:  
IDOR:ID1:ACK ACK

**:TRIGger:ENHanced:CANBus:IDOR:ID<x>:  
DATA?**

機能 CAN バス信号トリガの OR 条件の各データに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:CANBus:IDOR:ID<x>:  
DATA?  
<x> = 1 ~ 4  
<Nrf> = 0 ~ 8

例 :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:  
DATA? -> :TRIGGER:ENHANCED:CANBUS:  
IDOR:ID1:DATA:BORDER BIG;  
CONDITION DONTCARE;  
DATA1 0.0000000E+00;  
DATA2 255.000000E+00;DLC 8;MSBLSB 7,0;  
PATTERN "0000000100100011010001010110  
011110001001101010111100110111101111"  
;SIGN UNSIGN

**:TRIGger:ENHanced:CANBus:IDOR:ID<x>:  
DATA:BORDER**

機能 CAN バス信号トリガの OR 条件の各データのバイトオーダを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:IDOR:ID<x>:  
DATA:BORDER {BIG|LITTLE}  
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:  
DATA:BORDER?  
<x> = 1 ~ 4

例 :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:  
DATA:BORDER BIG  
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:  
DATA:BORDER? -> :TRIGGER:ENHANCED:  
CANBUS:IDOR:ID1:DATA:BORDER BIG

**:TRIGger:ENHanced:CANBus:IDOR:ID<x>:****DATA:CONDition**

機能 CAN バス信号トリガの OR 条件の各データ条件を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:IDOR:ID<x>:  
DATA:CONDition {BETWen|DONTcare|  
FALSe|GTHan|LTHan|ORANge|TRUE}  
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:  
DATA:CONDition?  
<x> = 1 ~ 4

例 :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:  
DATA:CONDITION BETWEEN  
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:  
DATA:CONDITION? -> :TRIGGER:ENHANCED:  
CANBUS:IDOR:ID1:DATA:  
CONDITION BETWEEN

**:TRIGger:ENHanced:CANBus:IDOR:ID<x>:  
DATA:DATA<x>**

機能 CAN バス信号トリガの OR 条件の各データの比較データを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:IDOR:ID<x>:  
DATA:DATA<x> {<Nrf>}  
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:  
DATA:DATA<x>?  
ID<x> の <x> = 1 ~ 4  
DATA<x> の <x> = 1, 2  
<Nrf> = 本体ユーザーズマニュアル参照。

例 :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:  
DATA:DATA1 1  
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:  
DATA:DATA1? -> :TRIGGER:ENHANCED:  
CANBUS:IDOR:ID1:DATA:  
DATA1 1.0000000E+00

解説

- 「:TRIGger:ENHANCED:CANBus:IDOR:ID<x>:DATA:CONDition GTHan」のときは「:TRIGger:ENHANCED:CANBus:IDOR:ID<x>:DATA:DATA1」で設定します。
- 「:TRIGger:ENHANCED:CANBus:IDOR:ID<x>:DATA:CONDition LTHan」のときは「:TRIGger:ENHANCED:CANBus:IDOR:ID<x>:DATA:DATA2」で設定します。
- 「:TRIGger:ENHANCED:CANBus:IDOR:ID<x>:DATA:CONDition BETWen|ORANge」のときは、小さい値を「:TRIGger:ENHANCED:CANBus:IDOR:ID<x>:DATA:DATA1」、大きい値を「:TRIGger:ENHANCED:CANBus:IDOR:ID<x>:DATA:DATA2」で設定します。

### 5.30 TRIGger グループ

#### : TRIGger: ENHanced: CANBus: IDOR: ID<x>:

##### DATA: DLC

機能 CAN バス信号トリガの OR 条件の各データの有効バイト数 (DLC) を設定 / 問い合わせします。

構文 : TRIGger: ENHanced: CANBus: IDOR: ID<x>:

DATA: DLC {<Nrf>}

: TRIGger: ENHanced: CANBus: IDOR: ID<x>:

DATA: DLC?

<x> = 1 ~ 4

<Nrf> = 0 ~ 8

例 : TRIGGER: ENHANCED: CANBUS: IDOR: ID1:

DATA: DLC 0

: TRIGGER: ENHANCED: CANBUS: IDOR: ID1:

DATA: DLC? -> : TRIGGER: ENHANCED:

CANBUS: IDOR: ID1: DATA: DLC 0

#### : TRIGger: ENHanced: CANBus: IDOR: ID<x>:

##### DATA: HEXA

機能 CAN バス信号トリガの OR 条件の各データを HEXA で設定します。

構文 : TRIGger: ENHanced: CANBus: IDOR: ID<x>:

DATA: HEXA {<文字列>}

<x> = 1 ~ 4

<文字列> = '0' ~ 'F', 'X' の組み合わせ 16 文字以内 (1 バイト単位)

例 : TRIGGER: ENHANCED: CANBUS: IDOR: ID1:

DATA: HEXA "A9"

#### : TRIGger: ENHanced: CANBus: IDOR: ID<x>:

##### DATA: MSBLSb

機能 CAN バス信号トリガの OR 条件の各データの MSB/LSB のビットを設定 / 問い合わせします。

構文 : TRIGger: ENHanced: CANBus: IDOR: ID<x>:

DATA: MSBLSb {<Nrf>, <Nrf>}

: TRIGger: ENHanced: CANBus: IDOR: ID<x>:

DATA: MSBLSb?

<x> = 1 ~ 4

<Nrf> = 本体ユーザーズマニュアル参照。

例 : TRIGGER: ENHANCED: CANBUS: IDOR: ID1:

DATA: MSBLSB 1, 0

: TRIGGER: ENHANCED: CANBUS: IDOR: ID1:

DATA: MSBLSB? -> : TRIGGER: ENHANCED:

CANBUS: IDOR: ID1: DATA: MSBLSB 1, 0

#### : TRIGger: ENHanced: CANBus: IDOR: ID<x>:

##### DATA: PATtern

機能 CAN バス信号トリガの OR 条件の各データを BINARY で設定 / 問い合わせします。

構文 : TRIGger: ENHanced: CANBus: IDOR: ID<x>:

DATA: PATtern {<文字列>}

: TRIGger: ENHanced: CANBus: IDOR: ID<x>:

DATA: PATtern?

<x> = 1 ~ 4

<文字列> = '0', '1', 'X' の組み合わせ 64 文字以内 (1 バイト単位)

例 : TRIGGER: ENHANCED: CANBUS: IDOR: ID1:

DATA: PATtern "11011111"

: TRIGGER: ENHANCED: CANBUS: IDOR: ID1:

DATA: PATtern? -> : TRIGGER: ENHANCED:

CANBUS: IDOR: ID1: DATA: PATtern

"11011111"

#### : TRIGger: ENHanced: CANBus: IDOR: ID<x>:

##### DATA: SIGN

機能 CAN バス信号トリガの OR 条件の各データの符号を設定 / 問い合わせします。

構文 : TRIGger: ENHanced: CANBus: IDOR: ID<x>:

DATA: SIGN {SIGN|UNSign}

: TRIGger: ENHanced: CANBus: IDOR: ID<x>:

DATA: SIGN?

<x> = 1 ~ 4

例 : TRIGGER: ENHANCED: CANBUS: IDOR: ID1:

DATA: SIGN SIGN

: TRIGGER: ENHANCED: CANBUS: IDOR: ID1:

DATA: SIGN? -> : TRIGGER: ENHANCED:

CANBUS: IDOR: ID1: DATA: SIGN SIGN

#### : TRIGger: ENHanced: CANBus: IDOR: ID<x>:

##### FORMAt

機能 CAN バス信号トリガの OR 条件の各メッセージフォーマット (標準 / 拡張) を設定 / 問い合わせします。

構文 : TRIGger: ENHanced: CANBus: IDOR: ID<x>:

FORMAt {STD|EXT}

: TRIGger: ENHanced: CANBus: IDOR: ID<x>:

FORMAt?

<x> = 1 ~ 4

例 : TRIGGER: ENHANCED: CANBUS: IDOR: ID1:

FORMAT STD

: TRIGGER: ENHANCED: CANBUS: IDOR: ID1:

FORMAT? -> : TRIGGER: ENHANCED: CANBUS:

IDOR: ID1: FORMAT STD

**:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDEXt?**

機能 CAN バス信号トリガの OR 条件の各拡張フォーマットの ID に関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDEXt?  
<x> = 1 ~ 4

例 :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDEXT? -> :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDEXT:PATTERN "1100101101110001110110111111"

**:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDEXt:HEXA**

機能 CAN バス信号トリガの OR 条件の各拡張フォーマットの ID を HEXA で設定します。

構文 :TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDEXt:HEXA {<文字列>}  
<x> = 1 ~ 4

例 :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDEXT:HEXA "1AEF5906"

**:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDEXt:PATtern**

機能 CAN バス信号トリガの OR 条件の各拡張フォーマットの ID を BINARY で設定/問い合わせします。

構文 :TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDEXt:PATtern {<文字列>}  
<x> = 1 ~ 4

例 :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDEXT:PATTERN "11001011011100001110110111111":TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDEXT:PATTERN? -> :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDEXT:PATTERN "11001011011100001110110111111"

**:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd?**

機能 CAN バス信号トリガの OR 条件の各標準フォーマットの ID に関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd?  
<x> = 1 ~ 4

例 :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDSTD? -> :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDSTD:PATTERN "00011111101"

**:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd:HEXA**

機能 CAN バス信号トリガの OR 条件の各標準フォーマットの ID を HEXA で設定します。

構文 :TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd:HEXA {<文字列>}  
<x> = 1 ~ 4  
<文字列> = '0' ~ 'F', 'X' の組み合わせ 3 文字

例 :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDSTD:HEXA "5DF"

**:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd:PATtern**

機能 CAN バス信号トリガの OR 条件の各標準フォーマットの ID を BINARY で設定/問い合わせします。

構文 :TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd:PATtern {<文字列>}  
<x> = 1 ~ 4

例 :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDSTD:PATTERN "101110111111":TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDSTD:PATTERN? -> :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDSTD:PATTERN "101110111111"

**:TRIGger:ENHanced:CANBus:IDOR:ID<x>:MODE**

機能 CAN バス信号トリガの OR 条件の各条件の有効 (1)/無効 (0) を設定/問い合わせします。

構文 :TRIGger:ENHanced:CANBus:IDOR:ID<x>:MODE {<Boolean>}  
<x> = 1 ~ 4

例 :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:MODE ON  
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:MODE? -> :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:MODE 1

## 5.30 TRIGger グループ

### **:TRIGger:ENHanced:CANBus:IDOR:ID<x>:**

#### **RTR**

機能 CAN バス信号トリガの OR 条件の各 RTR を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:IDOR:ID<x>:

RTR {DATA|DONTcare|REMOte}

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:

RTR?

<x> = 1 ~ 4

例 :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:

RTR DATA

:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:

RTR? -> :TRIGGER:ENHANCED:CANBUS:

IDOR:ID1:RTR DATA

### **:TRIGger:ENHanced:CANBus:IDSTd?**

機能 CAN バス信号トリガの標準フォーマットの ID に 関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:CANBus:IDSTd?

:TRIGGER:ENHANCED:CANBUS:IDSTD?

-> :TRIGGER:ENHANCED:CANBUS:IDSTD:

PATTERN "00011111101"

### **:TRIGger:ENHanced:CANBus:IDSTd:HEXA**

機能 CAN バス信号トリガの標準フォーマットの ID を HEXA で設定します。

構文 :TRIGger:ENHanced:CANBus:IDSTd:HEXA

{<文字列>}

<文字列> = '0' ~ 'F'、'X' の組み合わせ 3 文字

例 :TRIGGER:ENHANCED:CANBUS:IDSTD:

HEXA "5DF"

### **:TRIGger:ENHanced:CANBus:IDSTd:**

#### **PATtern**

機能 CAN バス信号トリガの標準フォーマットの ID を BINARY で設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:IDSTd:

PATtern {<文字列>}

:TRIGger:ENHanced:CANBus:IDSTd:

PATtern?

<文字列> = '0'、'1'、'X' の組み合わせ 11 文字

例 :TRIGGER:ENHANCED:CANBUS:IDSTD:

PATTERN "10111011111"

:TRIGGER:ENHANCED:CANBUS:IDSTD:

PATtern? -> :TRIGGER:ENHANCED:CANBUS:

IDSTD:PATTERN "10111011111"

### **:TRIGger:ENHanced:CANBus:MODE**

機能 CAN バス信号トリガのモードを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:

MODE {EFrame|IDEXt|IDOR|IDSTd|SOF}

:TRIGger:ENHanced:CANBus:MODE?

例 :TRIGGER:ENHANCED:CANBUS:MODE EFRAME

:TRIGGER:ENHANCED:CANBUS:MODE?

-> :TRIGGER:ENHANCED:CANBUS:

MODE EFRAME

### **:TRIGger:ENHanced:CANBus:REcessive**

機能 CAN バス信号トリガのリセッシブレベル (バス レベル) を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:

REcessive {HIGH|LOW}

:TRIGger:ENHanced:CANBus:REcessive?

例 :TRIGGER:ENHANCED:CANBUS:

RECESSIVE HIGH

:TRIGGER:ENHANCED:CANBUS:

RECESSIVE? -> :TRIGGER:ENHANCED:

CANBUS:RECESSIVE HIGH

### **:TRIGger:ENHanced:CANBus:RTR**

機能 CAN バス信号トリガの RTR を設定 / 問い合わせ します。

構文 :TRIGger:ENHanced:CANBus:

RTR {DATA|DONTcare|REMOte}

:TRIGger:ENHanced:CANBus:RTR?

例 :TRIGGER:ENHANCED:CANBUS:RTR DATA

:TRIGGER:ENHANCED:CANBUS:RTR?

-> :TRIGGER:ENHANCED:CANBUS:RTR DATA

### **:TRIGger:ENHanced:CANBus:SOURce**

機能 CAN バス信号トリガのトリガソースを設定 / 問 合わせします。

構文 :TRIGger:ENHanced:CANBus:SOURce

{<Nrf>}

:TRIGger:ENHanced:CANBus:SOURce?

<Nrf> = 1 ~ 4

例 :TRIGGER:ENHANCED:CANBUS:SOURCE 1

:TRIGGER:ENHANCED:CANBUS:SOURCE?

-> :TRIGGER:ENHANCED:CANBUS:SOURCE 1

### **:TRIGger:ENHanced:CANBus:SPOint**

機能 CAN バス信号トリガのサンプルポイントを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:CANBus:

SPOint {<Nrf>}

:TRIGger:ENHanced:CANBus:SPOint?

<Nrf> = 18.8 ~ 90.6(%)

例 :TRIGGER:ENHANCED:CANBUS:SPOINT 18.8

:TRIGGER:ENHANCED:CANBUS:

SPOINT? -> :TRIGGER:ENHANCED:CANBUS:

SPOINT 18.8E+00

**:TRIGger:ENHanced:I2CBus?**

機能 I<sup>2</sup>C バストリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:I2CBus?

例 :TRIGGER:ENHANCED:I2CBUS?  
-> :TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT10ADDRESS:PATTERN "10111011111";:  
TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT7ADDRESS:PATTERN "11011110";:  
TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT7APSUB:ADDRESS:  
PATTERN "10101011";:TRIGGER:ENHANCED:  
I2CBUS:ADATA:BIT7APSUB:SADDRESS:  
PATTERN "10101011";:TRIGGER:ENHANCED:  
I2CBUS:ADATA:TYPE BIT10ADDRESS;:  
TRIGGER:ENHANCED:I2CBUS:CLOCK:  
SOURCE 1;:TRIGGER:ENHANCED:I2CBUS:  
DATA:BYTE 1;CONDITION TRUE;  
DPOSITION 1;MODE 1;  
PATTERN1 "10101011";  
PATTERN2 "10101010";  
PATTERN3 "10101111";  
PATTERN4 "10101011";PMODE DONTCARE;  
SOURCE 1;:TRIGGER:ENHANCED:I2CBUS:  
GCALL:BIT7MADDRESS:  
PATTERN "1010101";:TRIGGER:ENHANCED:  
I2CBUS:GCALL:SBYTE BIT7MADDRESS;:  
TRIGGER:ENHANCED:I2CBUS:MODE ADATA;  
NAIGNORE:HSMODE 1;RACCESS 1;SBYTE 1;:  
TRIGGER:ENHANCED:I2CBUS:SBHSMODE:  
TYPE HSMODE

**:TRIGger:ENHanced:I2CBus:ADATa?**

機能 I<sup>2</sup>C バストリガのアドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:I2CBus:ADATa?

例 :TRIGGER:ENHANCED:I2CBUS:ADATa?  
-> :TRIGGER:ENHANCED:I2CBUS:ADATa:  
BIT10ADDRESS:PATTERN "10111011111";:  
TRIGGER:ENHANCED:I2CBUS:ADATa:  
BIT7ADDRESS:PATTERN "11011110";:  
TRIGGER:ENHANCED:I2CBUS:ADATa:  
BIT7APSUB:ADDRESS:  
PATTERN "10101011";:TRIGGER:ENHANCED:  
I2CBUS:ADATA:BIT7APSUB:SADDRESS:  
PATTERN "10101011";:TRIGGER:ENHANCED:  
I2CBUS:ADATA:TYPE BIT10ADDRESS

**:TRIGger:ENHanced:I2CBus:ADATa:BIT10address?**

機能 I<sup>2</sup>C バストリガの 10bit アドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:I2CBus:ADATa:  
BIT10address?

例 :TRIGGER:ENHANCED:I2CBUS:ADATa:  
BIT10ADDRESS? -> :TRIGGER:ENHANCED:  
I2CBUS:ADATA:BIT10ADDRESS:  
PATTERN "10111011111"

**:TRIGger:ENHanced:I2CBus:ADATa:BIT10address:HEXA**

機能 I<sup>2</sup>C バストリガの 10bit アドレスを HEXA で設定します。

構文 :TRIGger:ENHanced:I2CBus:ADATa:

BIT10address:HEXA {<文字列>}  
<文字列> = '0'~'F','X' の組み合わせ 3 文字 (ビット 8 は、R/W ビット)

例 :TRIGGER:ENHANCED:I2CBUS:ADATa:  
BIT10ADDRESS:HEXA "7AB"

**:TRIGger:ENHanced:I2CBus:ADATa:BIT10address:PATtern**

機能 I<sup>2</sup>C バストリガの 10bit アドレスを BINARY で設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2CBus:ADATa:

BIT10address:PATtern {<文字列>}  
:TRIGger:ENHanced:I2CBus:ADATa:  
BIT10address:PATtern?  
<文字列> = '0','1','X' の組み合わせ 11 文字 (ビット 8 は、R/W ビット)

例 :TRIGGER:ENHANCED:I2CBUS:ADATa:  
BIT10ADDRESS:PATTERN "10111011111"  
:TRIGGER:ENHANCED:I2CBUS:ADATa:  
BIT10ADDRESS:PATTERN? -> :TRIGGER:  
ENHANCED:I2CBUS:ADATA:BIT10ADDRESS:  
PATTERN "10111011111"

**:TRIGger:ENHanced:I2CBus:ADATa:BIT7Address?**

機能 I<sup>2</sup>C バストリガの 7bit アドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:I2CBus:ADATa:  
BIT7Address?

例 :TRIGGER:ENHANCED:I2CBUS:ADATa:  
BIT7ADDRESS? -> :TRIGGER:ENHANCED:  
I2CBUS:ADATA:BIT7ADDRESS:  
PATTERN "11011110"

**:TRIGger:ENHanced:I2CBus:ADATa:BIT7Address:HEXA**

機能 I<sup>2</sup>C バストリガの 7bit アドレスを HEXA で設定します。

構文 :TRIGger:ENHanced:I2CBus:ADATa:

BIT7Address:HEXA {<文字列>}  
<文字列> = '0'~'F','X' の組み合わせ 2 文字 (ビット 0 は、R/W ビット)

例 :TRIGGER:ENHANCED:I2CBUS:ADATa:  
BIT7ADDRESS:HEXA "DE"

## 5.30 TRIGger グループ

### **:TRIGger:ENHanced:I2Cbus:ADATa:**

#### **BIT7Address: PATtern**

機能 I<sup>2</sup>C バストリガの 7bit アドレスを BINARY で設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2Cbus:ADATa:  
BIT7Address:PATtern {<文字列>}  
:TRIGger:ENHanced:I2Cbus:ADATa:  
BIT7Address:PATtern?  
<文字列> = '0','1','X' の組み合わせ 8 文字 (ビット 0 は、R/W ビット)

例 :TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT7ADDRESS:PATTERN "11011110"  
:TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT7ADDRESS:PATTERN? -> :TRIGGER:  
ENHANCED:I2CBUS:ADATA:BIT7ADDRESS:  
PATTERN "11011110"

### **:TRIGger:ENHanced:I2Cbus:ADATa:**

#### **BIT7APsub?**

機能 I<sup>2</sup>C バストリガの 7bit + Sub アドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:I2Cbus:ADATa:  
BIT7APsub?

例 :TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT7APSUB? -> :TRIGGER:ENHANCED:  
I2CBUS:ADATA:BIT7APSUB:ADDRESS:  
PATTERN "10101011";:TRIGGER:ENHANCED:  
I2CBUS:ADATA:BIT7APSUB:SADDRESS:  
PATTERN "10101011"

### **:TRIGger:ENHanced:I2Cbus:ADATa:**

#### **BIT7APsub: ADDRESS?**

機能 I<sup>2</sup>C バストリガの 7bit + Sub アドレスの 7bit アドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:I2Cbus:ADATa:  
BIT7APsub:ADDRESS?

例 :TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT7APSUB:ADDRESS? -> :TRIGGER:  
ENHANCED:I2CBUS:ADATA:BIT7APSUB:  
ADDRESS:PATTERN "10101011"

### **:TRIGger:ENHanced:I2Cbus:ADATa:**

#### **BIT7APsub: ADDRESS: HEXA**

機能 I<sup>2</sup>C バストリガの 7bit + Sub アドレスの 7bit アドレスを HEXA で設定します。

構文 :TRIGger:ENHanced:I2Cbus:ADATa:  
BIT7APsub:ADDRESS:HEXA {<文字列>}  
<文字列> = '0'~'F','X' の組み合わせ 2 文字 (ビット 0 は、R/W ビット)

例 :TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT7APSUB:ADDRESS:HEXA "AB"

### **:TRIGger:ENHanced:I2Cbus:ADATa:**

#### **BIT7APsub: ADDRESS: PATtern**

機能 I<sup>2</sup>C バストリガの 7bit + Sub アドレスの 7bit アドレスを BINARY で設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2Cbus:ADATa:  
BIT7APsub:ADDRESS:PATtern {<文字列>}  
:TRIGger:ENHanced:I2Cbus:ADATa:  
BIT7APsub:ADDRESS:PATtern?  
<文字列> = '0','1','X' の組み合わせ 8 文字 (ビット 0 は、R/W ビット)

例 :TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT7APSUB:ADDRESS:PATTERN "10101011"  
:TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT7APSUB:ADDRESS:PATTERN?  
-> :TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT7APSUB:ADDRESS:PATTERN "10101011"

### **:TRIGger:ENHanced:I2Cbus:ADATa:**

#### **BIT7APsub: SADDRESS?**

機能 I<sup>2</sup>C バストリガの 7bit + Sub アドレスの Sub アドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:I2Cbus:ADATa:  
BIT7APsub:SADDRESS?

例 :TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT7APSUB:SADDRESS? -> :TRIGGER:  
ENHANCED:I2CBUS:ADATA:BIT7APSUB:  
SADDRESS:PATTERN "10101011"

### **:TRIGger:ENHanced:I2Cbus:ADATa:**

#### **BIT7APsub: SADDRESS: HEXA**

機能 I<sup>2</sup>C バストリガの 7bit + Sub アドレスの Sub アドレスを HEXA で設定します。

構文 :TRIGger:ENHanced:I2Cbus:ADATa:  
BIT7APsub:SADDRESS:HEXA {<文字列>}  
<文字列> = '0'~'F','X' の組み合わせ 2 文字

例 :TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT7APSUB:SADDRESS:HEXA "EF"

### **:TRIGger:ENHanced:I2Cbus:ADATa:**

#### **BIT7APsub: SADDRESS: PATtern**

機能 I<sup>2</sup>C バストリガの 7bit + Sub アドレスの Sub アドレスを BINARY で設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2Cbus:ADATa:  
BIT7APsub:SADDRESS:PATTERN {<文字列>}  
:TRIGger:ENHanced:I2Cbus:ADATa:  
BIT7APsub:SADDRESS:PATTERN?  
<文字列> = '0','1','X' の組み合わせ 8 文字

例 :TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT7APSUB:SADDRESS:PATTERN "10101011"  
:TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT7APSUB:SADDRESS:PATTERN?  
-> :TRIGGER:ENHANCED:I2CBUS:ADATA:  
BIT7APSUB:SADDRESS:PATTERN "10101011"

**:TRIGger:ENHanced:I2CBus:ADATa:TYPE**

機能 I<sup>2</sup>C バストリガのアドレスの種類を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2CBus:ADATa:  
TYPE {BIT10address|BIT7Address|  
BIT7APsub}  
:TRIGger:ENHanced:I2CBus:ADATa:TYPE?

例 :TRIGGER:ENHANCED:I2CBUS:ADATa:  
TYPE BIT10ADDRESS  
:TRIGGER:ENHANCED:I2CBUS:ADATa:TYPE?  
-> :TRIGGER:ENHANCED:I2CBUS:ADATa:  
TYPE BIT10ADDRESS

**:TRIGger:ENHanced:I2CBus:CLOCK?**

機能 I<sup>2</sup>C バストリガのクロックに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:I2CBus:CLOCK?  
例 :TRIGGER:ENHANCED:I2CBUS:CLOCK?

-> :TRIGGER:ENHANCED:I2CBUS:CLOCK:  
SOURCE 1

**:TRIGger:ENHanced:I2CBus:CLOCK:****SOURCE**

機能 I<sup>2</sup>C バストリガのクロックトレースを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2CBus:CLOCK:  
SOURCE {<Nrf>}  
:TRIGger:ENHanced:I2CBus:CLOCK:  
SOURCE?  
<Nrf> = 1 ~ 4

例 :TRIGGER:ENHANCED:I2CBUS:CLOCK:  
SOURCE 1  
:TRIGGER:ENHANCED:I2CBUS:CLOCK:  
SOURCE? -> :TRIGGER:ENHANCED:I2CBUS:  
CLOCK:SOURCE 1

**:TRIGger:ENHanced:I2CBus:DATA?**

機能 I<sup>2</sup>C バストリガのデータに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:I2CBus:DATA?  
<x> = 1, 2

例 :TRIGGER:ENHANCED:I2CBUS:DATA? -> :  
TRIGGER:ENHANCED:I2CBUS:DATA:BYTE 1;  
CONDITION TRUE;DPOSITION 1;MODE 1;  
PATTERN1 "10101011";  
PATTERN2 "10101010";  
PATTERN3 "10101111";  
PATTERN4 "10101011";PMODE DONTCARE;  
SOURCE 1

**:TRIGger:ENHanced:I2CBus:DATA:BYTE**

機能 I<sup>2</sup>C バストリガの設定データ数を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2CBus:DATA:  
BYTE {<Nrf>}  
:TRIGger:ENHanced:I2CBus:DATA:BYTE?  
<Nrf> = 1 ~ 4

例 :TRIGGER:ENHANCED:I2CBUS:DATA:BYTE 1  
:TRIGGER:ENHANCED:I2CBUS:DATA:BYTE?  
-> :TRIGGER:ENHANCED:I2CBUS:DATA:  
BYTE 1

**:TRIGger:ENHanced:I2CBus:DATA:****CONDition**

機能 I<sup>2</sup>C バストリガのデータの判定方法 (一致 / 不一致) を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2CBus:DATA:  
CONDition {FALSE|TRUE}  
:TRIGger:ENHanced:I2CBus:DATA:  
CONDition?

例 :TRIGGER:ENHANCED:I2CBUS:DATA:  
CONDITION TRUE  
:TRIGGER:ENHANCED:I2CBUS:DATA:  
CONDITION? -> :TRIGGER:ENHANCED:  
I2CBUS:DATA:CONDITION TRUE

**:TRIGger:ENHanced:I2CBus:DATA:****DPOSITION**

機能 I<sup>2</sup>C バストリガのデータのパターン比較する位置を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2CBus:DATA:  
DPOSITION {<Nrf>}  
:TRIGger:ENHanced:I2CBus:DATA:  
DPOSITION?  
<Nrf> = 0 ~ 9999

例 :TRIGGER:ENHANCED:I2CBUS:DATA:  
DPOSITION 1  
:TRIGGER:ENHANCED:I2CBUS:DATA:  
DPOSITION? -> :TRIGGER:ENHANCED:  
I2CBUS:DATA:DPOSITION 1

**:TRIGger:ENHanced:I2CBus:DATA:****HEXA<x>**

機能 I<sup>2</sup>C バストリガのデータを HEXA で設定します。

構文 :TRIGger:ENHanced:I2CBus:DATA:  
HEXA<x> {<文字列>}  
<x> = 1 ~ 4

<文字列> = '0' ~ 'F', 'X' の組み合わせ 2 文字

例 :TRIGGER:ENHANCED:I2CBUS:DATA:  
HEXA1 "AB"



## 5.30 TRIGger グループ

### **:TRIGger:ENHanced:I2CBus:DATA:MODE**

機能 I<sup>2</sup>C バストリガのデータ条件の有効/無効を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2CBus:DATA:  
MODE {<Boolean>}

例 :TRIGGER:ENHANCED:I2CBUS:DATA:MODE?  
:TRIGGER:ENHANCED:I2CBUS:DATA:MODE ON  
-> :TRIGGER:ENHANCED:I2CBUS:DATA:  
MODE 1

### **:TRIGger:ENHanced:I2CBus:DATA: PATTERN<x>**

機能 I<sup>2</sup>C バストリガのデータを BINARY で設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2CBus:DATA:  
PATTERN<x> {<文字列>}  
:TRIGger:ENHanced:I2CBus:DATA:  
PATTERN<x>?  
<x> = 1 ~ 4

例 <文字列> = '0', '1', 'X' の組み合わせ 8 文字  
:TRIGGER:ENHANCED:I2CBUS:DATA:  
PATTERN1 "10101011"  
:TRIGGER:ENHANCED:I2CBUS:DATA:  
PATTERN1? -> :TRIGGER:ENHANCED:  
I2CBUS:DATA:PATTERN1 "10101011"

### **:TRIGger:ENHanced:I2CBus:DATA:PMODE**

機能 I<sup>2</sup>C バストリガのデータのパターン比較先頭位置  
モードを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2CBus:DATA:  
PMODE {DONTcare|SElect}  
:TRIGger:ENHanced:I2CBus:DATA:PMODE?  
例 :TRIGGER:ENHANCED:I2CBUS:DATA:  
PMODE SELECT  
:TRIGGER:ENHANCED:I2CBUS:DATA:PMODE?  
-> :TRIGGER:ENHANCED:I2CBUS:DATA:  
PMODE SELECT

### **:TRIGger:ENHanced:I2CBus:DATA:SOURce**

機能 I<sup>2</sup>C バストリガのデータトレースを設定 / 問い合  
わせします。

構文 :TRIGger:ENHanced:I2CBus:DATA:  
SOURce {<Nrf>}  
:TRIGger:ENHanced:I2CBus:DATA:SOURce?  
<Nrf> = 1 ~ 4

例 :TRIGGER:ENHANCED:I2CBUS:DATA:  
SOURCE 1  
:TRIGGER:ENHANCED:I2CBUS:DATA:SOURCE?  
-> :TRIGGER:ENHANCED:I2CBUS:DATA:  
SOURCE 1

### **:TRIGger:ENHanced:I2CBus:GCALl?**

機能 I<sup>2</sup>C バストリガのジェネラルコールに関するすべ  
ての設定値を問い合わせます。

構文 :TRIGger:ENHanced:I2CBus:GCALl?  
<x> = 1, 2

例 :TRIGGER:ENHANCED:I2CBUS:GCALL?  
-> :TRIGGER:ENHANCED:I2CBUS:GCALL:  
BIT7MADDRESS:PATTERN "1010101";;  
TRIGGER:ENHANCED:I2CBUS:GCALL:  
SBYTE BIT7MADDRESS

### **:TRIGger:ENHanced:I2CBus:GCALl: BIT7maddress?**

機能 I<sup>2</sup>C バストリガのジェネラルコールの 7bit マスタ  
アドレスに関するすべての設定値を問い合わせ  
ます。

構文 :TRIGger:ENHanced:I2CBus:GCALl:  
BIT7maddress?  
<x> = 1, 2

例 :TRIGGER:ENHANCED:I2CBUS:GCALL:  
BIT7MADDRESS? -> :TRIGGER:ENHANCED:  
I2CBUS:GCALL:BIT7MADDRESS:  
PATTERN "1010101"

### **:TRIGger:ENHanced:I2CBus:GCALl: BIT7maddress:HEXA**

機能 I<sup>2</sup>C バストリガのジェネラルコールの 7bit マスタ  
アドレスを HEXA で設定します。

構文 :TRIGger:ENHanced:I2CBus:GCALl:  
BIT7maddress:HEXA {<文字列>}  
<x> = 1, 2

<文字列> = '0' ~ 'F', 'X' の組み合わせ 2 文字 (ビッ  
ト 0 は, '1' に固定)  
例 :TRIGGER:ENHANCED:I2CBUS:GCALL:  
BIT7MADDRESS:HEXA "AB"

### **:TRIGger:ENHanced:I2CBus:GCALl: BIT7maddress:PATtern**

機能 I<sup>2</sup>C バストリガのジェネラルコールの 7bit マスタ  
アドレスを BINARY で設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2CBus:GCALl:  
BIT7maddress:PATtern {<文字列>}  
:TRIGger:ENHanced:I2CBus:GCALl:  
BIT7maddress:PATtern?  
<x> = 1, 2

<文字列> = '0', '1', 'X' の組み合わせ 7 文字  
例 :TRIGGER:ENHANCED:I2CBUS:GCALL:  
BIT7MADDRESS:PATTERN "1010101"  
:TRIGGER:ENHANCED:I2CBUS:GCALL:  
BIT7MADDRESS:PATTERN? -> :TRIGGER:  
ENHANCED:I2CBUS:GCALL:BIT7MADDRESS:  
PATTERN "1010101"

**:TRIGger:ENHanced:I2Cbus:GCALl:SBYTE (Second Byte)**

機能 I<sup>2</sup>C バストリガのジェネラルコールのセカンドバイトのタイプを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2Cbus:GCALl:SBYTE {BIT7maddress|DONTcare|H04|H06}  
:TRIGger:ENHanced:I2Cbus:GCALl:SBYTE?

例 :TRIGGER:ENHANCED:I2CBUS:GCALL:SBYTE BIT7MADDRESS  
:TRIGGER:ENHANCED:I2CBUS:GCALL:SBYTE?  
-> :TRIGGER:ENHANCED:I2CBUS:GCALL:SBYTE BIT7MADDRESS

**:TRIGger:ENHanced:I2Cbus:MODE**

機能 I<sup>2</sup>C バストリガのトリガモードを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2Cbus:MODE {ADATa|ESTart|GCALl|NAIGnore|SBHSmode}  
:TRIGger:ENHanced:I2Cbus:MODE?

例 :TRIGGER:ENHANCED:I2CBUS:MODE ADATA  
:TRIGGER:ENHANCED:I2CBUS:MODE? -> :TRIGGER:ENHANCED:I2CBUS:MODE ADATA

**:TRIGger:ENHanced:I2Cbus:NAIGnore?**

機能 I<sup>2</sup>C バストリガの NON ACK 無視モードに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:I2Cbus:NAIGnore?  
例 :TRIGGER:ENHANCED:I2CBUS:NAIGNORE?

-> :TRIGGER:ENHANCED:I2CBUS:NAIGNORE:HSMODE 1;RACCESS 1;SBYTE 1

**:TRIGger:ENHanced:I2Cbus:NAIGnore:HSMode**

機能 I<sup>2</sup>C バストリガのハイスピードモードで NON ACK を無視する / しないを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2Cbus:NAIGnore:HSMode {<Boolean>}  
:TRIGger:ENHanced:I2Cbus:NAIGnore:HSMode?

例 :TRIGGER:ENHANCED:I2CBUS:NAIGNORE:HSMODE ON  
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:HSMODE? -> :TRIGGER:ENHANCED:I2CBUS:NAIGNORE:HSMODE 1

**:TRIGger:ENHanced:I2Cbus:NAIGnore:RACcess**

機能 I<sup>2</sup>C バストリガのリードアクセスモードで NON ACK を無視する / しないを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2Cbus:NAIGnore:RACcess {<Boolean>}  
:TRIGger:ENHanced:I2Cbus:NAIGnore:RACcess?

例 :TRIGGER:ENHANCED:I2CBUS:NAIGNORE:RACCESS ON  
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:RACCESS? -> :TRIGGER:ENHANCED:I2CBUS:NAIGNORE:RACCESS 1

**:TRIGger:ENHanced:I2Cbus:NAIGnore:SBYTE (Start Byte)**

機能 I<sup>2</sup>C バストリガのスタートバイトで NON ACK を無視する / しないを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2Cbus:NAIGnore:SBYTE {<Boolean>}  
:TRIGger:ENHanced:I2Cbus:NAIGnore:SBYTE?

例 :TRIGGER:ENHANCED:I2CBUS:NAIGNORE:SBYTE ON  
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:SBYTE? -> :TRIGGER:ENHANCED:I2CBUS:NAIGNORE:SBYTE 1

**:TRIGger:ENHanced:I2Cbus:SBHSmode?**

機能 I<sup>2</sup>C バストリガのスタートバイト / ハイスピードモードに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:I2Cbus:SBHSmode?  
例 :TRIGGER:ENHANCED:I2CBUS:SBHSMODE?

-> :TRIGGER:ENHANCED:I2CBUS:SBHSMODE:TYPE HSMODE

**:TRIGger:ENHanced:I2Cbus:SBHSmode:TYPE**

機能 I<sup>2</sup>C バストリガのスタートバイト / ハイスピードモードのタイプを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:I2Cbus:SBHSmode:TYPE {HSMode|SBYTE}  
:TRIGger:ENHanced:I2Cbus:SBHSmode:TYPE?

例 :TRIGGER:ENHANCED:I2CBUS:SBHSMODE:TYPE HSMODE  
:TRIGGER:ENHANCED:I2CBUS:SBHSMODE:TYPE? -> :TRIGGER:ENHANCED:I2CBUS:SBHSMODE:TYPE HSMODE

## 5.30 TRIGger グループ

### **:TRIGger:ENHanced:LINBus?**

機能 LIN バストリガに関するすべての設定値を問い合わせします。

構文 :TRIGger:ENHanced:LINBus?

例 :TRIGGER:ENHANCED:LINBUS?

-> :TRIGGER:ENHANCED:LINBUS:  
BRATE 19200;SOURCE 1

### **:TRIGger:ENHanced:LINBus:BRATe**

機能 LIN バス信号トリガのビットレート (データ転送速度) を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:LINBus:BRATe  
{<NRf>|USER,<NRf>}

:TRIGger:ENHanced:LINBus:BRATe?

<NRf> = 1200、2400、4800、9600、19200

USER の <NRf> = 本体ユーザーズマニュアル参照。

例 :TRIGGER:ENHANCED:LINBUS:BRATE 19200

:TRIGGER:ENHANCED:LINBUS:BRATE?

-> :TRIGGER:ENHANCED:LINBUS:

BRATE 19200

### **:TRIGger:ENHanced:LINBus:SOURce**

機能 LIN バス信号トリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:LINBus:

SOURce {<NRf>}

:TRIGger:ENHanced:LINBus:SOURce?

<NRf> = 1 ~ 4

例 :TRIGGER:ENHANCED:LINBUS:SOURCE 1

:TRIGGER:ENHANCED:LINBUS:SOURCE?

-> :TRIGGER:ENHANCED:LINBUS:SOURCE 1

### **:TRIGger:ENHanced:SPATtern? (Serial Pattern)**

機能 シリアルパターントリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:SPATtern?

例 :TRIGGER:ENHANCED:SPATTERN?

-> :TRIGGER:ENHANCED:SPATTERN:

BITRATE 1.000E+00;CLOCK:MODE 1;

POLARITY FALL;SOURCE 1;:TRIGGER:

ENHANCED:SPATTERN:CS 1;DATA:

ACTIVE HIGH;SOURCE 1;:TRIGGER:

ENHANCED:SPATTERN:LATCH:SOURCE 1;

POLARITY FALL;:TRIGGER:ENHANCED:

SPATTERN:PATTERN "1100110111101111"

### **:TRIGger:ENHanced:SPATtern:BITRate**

機能 シリアルパターントリガのビットレートを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:SPATtern:

BITRate {<NRf>}

:TRIGger:ENHanced:SPATtern:BITRate?

<NRf> = 1 ~ 50M(bps)

例 :TRIGGER:ENHANCED:SPATTERN:BITRATE 1

:TRIGGER:ENHANCED:SPATTERN:BITRATE?

-> :TRIGGER:ENHANCED:SPATTERN:

BITRATE 1.000E+00

解説 「:TRIGger:ENHanced:SPATtern:CLOCK:  
MODE OFF」のときに有効です。

### **:TRIGger:ENHanced:SPATtern:CLEar**

機能 シリアルパターントリガのパターンをすべてクリア (Don't care) します。

構文 :TRIGger:ENHanced:SPATtern:CLEar

例 :TRIGGER:ENHANCED:SPATTERN:CLEAR

### **:TRIGger:ENHanced:SPATtern:CLOCK?**

機能 シリアルパターントリガのクロックに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:SPATtern:CLOCK?

例 :TRIGGER:ENHANCED:SPATTERN:CLOCK?

-> :TRIGGER:ENHANCED:SPATTERN:CLOCK:

MODE 1;POLARITY FALL;SOURCE 1

### **:TRIGger:ENHanced:SPATtern:CLOCK:MODE**

機能 シリアルパターントリガのクロックの有効 / 無効を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:SPATtern:CLOCK:

MODE {<Boolean>}

:TRIGger:ENHanced:SPATtern:CLOCK:

MODE?

例 :TRIGGER:ENHANCED:SPATTERN:CLOCK:

MODE ON

:TRIGGER:ENHANCED:SPATTERN:CLOCK:

MODE? -> :TRIGGER:ENHANCED:SPATTERN:

CLOCK:MODE 1

**:TRIGger:ENHanced:SPATtern:CLOCK:POLarity**

機能 シリアルパターントリガのクロックトレースの極性を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:SPATtern:CLOCK:POLarity {FALL|RISE}  
:TRIGger:ENHanced:SPATtern:CLOCK:POLarity?

例 :TRIGGER:ENHANCED:SPATTERN:CLOCK:POLARITY FALL  
:TRIGGER:ENHANCED:SPATTERN:CLOCK:POLARITY? -> :TRIGGER:ENHANCED:SPATTERN:CLOCK:POLARITY FALL

解説 「:TRIGger:ENHanced:SPATtern:CLOCK:MODE ON」のときに有効です。

**:TRIGger:ENHanced:SPATtern:CLOCK:SOURce**

機能 シリアルパターントリガのクロックトレースを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:SPATtern:CLOCK:SOURce {<Nrf>}  
:TRIGger:ENHanced:SPATtern:CLOCK:SOURce?

例 <Nrf> = 1 ~ 4  
:TRIGGER:ENHANCED:SPATTERN:CLOCK:SOURCE 1  
:TRIGGER:ENHANCED:SPATTERN:CLOCK:SOURCE? -> :TRIGGER:ENHANCED:SPATTERN:CLOCK:SOURCE 1

解説 「:TRIGger:ENHanced:SPATtern:CLOCK:MODE ON」のときに有効です。

**:TRIGger:ENHanced:SPATtern:CS**

機能 シリアルパターントリガのチップセレクトの有効 / 無効を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:SPATtern:CS {<Boolean>}  
:TRIGger:ENHanced:SPATtern:CS?

例 :TRIGGER:ENHANCED:SPATTERN:CS ON  
:TRIGGER:ENHANCED:SPATTERN:CS?  
-> :TRIGGER:ENHANCED:SPATTERN:CS 1

解説 「:TRIGger:ENHanced:SPATtern:CLOCK:MODE ON」のときに有効です。

**:TRIGger:ENHanced:SPATtern:DATA?**

機能 シリアルパターントリガのデータに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:SPATtern:DATA?

例 :TRIGGER:ENHANCED:SPATTERN:DATA?  
-> :TRIGGER:ENHANCED:SPATTERN:DATA:ACTIVE HIGH;SOURCE 1

**:TRIGger:ENHanced:SPATtern:DATA:ACTiVe**

機能 シリアルパターントリガのデータのアクティブレベルを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:SPATtern:DATA:ACTiVe {HIGH|LOW}  
:TRIGger:ENHanced:SPATtern:DATA:ACTiVe?

例 :TRIGGER:ENHANCED:SPATTERN:DATA:ACTIVE HIGH  
:TRIGGER:ENHANCED:SPATTERN:DATA:ACTIVE? -> :TRIGGER:ENHANCED:SPATTERN:DATA:ACTIVE HIGH

**:TRIGger:ENHanced:SPATtern:DATA:SOURce**

機能 シリアルパターントリガのデータトレースを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:SPATtern:DATA:SOURce {<Nrf>}  
:TRIGger:ENHanced:SPATtern:DATA:SOURce?  
<Nrf> = 1 ~ 4

例 :TRIGGER:ENHANCED:SPATTERN:DATA:SOURCE 1  
:TRIGGER:ENHANCED:SPATTERN:DATA:SOURCE? -> :TRIGGER:ENHANCED:SPATTERN:DATA:SOURCE 1

**:TRIGger:ENHanced:SPATtern:HEXA**

機能 シリアルパターントリガのパターンを HEXA で設定します。

構文 :TRIGger:ENHanced:SPATtern:HEXA {<文字列>}  
<文字列> = '0' ~ 'F'、'X' の組み合わせ 32 文字以内。

例 :TRIGGER:ENHANCED:SPATTERN:HEXA "ABCD"

**:TRIGger:ENHanced:SPATtern:LATCh?**

機能 シリアルパターントリガのラッチに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:SPATtern:LATCh?

例 :TRIGGER:ENHANCED:SPATTERN:LATCH?  
-> :TRIGGER:ENHANCED:SPATTERN:LATCH:SOURCE 1;POLARITY FALL

### 5.30 TRIGger グループ

#### **:TRIGger:ENHanced:SPATtern:LATCh:POLarity**

機能 シリアルパターントリガのラッチトレースの極性を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:SPATtern:LATCh:  
POLarity {FALL|RISE}  
:TRIGger:ENHanced:SPATtern:LATCh:  
POLarity?

例 :TRIGGER:ENHANCED:SPATTERN:LATCH:  
POLARITY FALL  
:TRIGGER:ENHANCED:SPATTERN:LATCH:  
POLARITY? -> :TRIGGER:ENHANCED:  
SPATTERN:LATCH:POLARITY FALL

解説

- ・「:TRIGger:ENHanced:SPATtern:CLOCK:MODE ON」のときに有効です。
- ・「:TRIGger:ENHanced:SPATtern:LATCh:SOURCE NONE」のときは無効です。

#### **:TRIGger:ENHanced:SPATtern:LATCh:SOURCE**

機能 シリアルパターントリガのラッチトレースを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:SPATtern:LATCh:  
SOURCE {<NRF>|NONE}  
:TRIGger:ENHanced:SPATtern:LATCh:  
SOURCE?  
<NRF> = 1 ~ 4

例 :TRIGGER:ENHANCED:SPATTERN:LATCH:  
SOURCE 1  
:TRIGGER:ENHANCED:SPATTERN:LATCH:  
SOURCE? -> :TRIGGER:ENHANCED:  
SPATTERN:  
LATCH:SOURCE 1

解説 「:TRIGger:ENHanced:SPATtern:CLOCK:MODE ON」のときに有効です。

#### **:TRIGger:ENHanced:SPATtern:PATtern**

機能 シリアルパターントリガのパターンを BINARY で設定 / 問い合わせします。

構文 :TRIGger:ENHanced:SPATtern:  
PATTERN {<文字列>}  
:TRIGger:ENHanced:SPATtern:PATtern?  
<文字列> = '0','1','X'の組み合わせ 128文字以内。

例 :TRIGGER:ENHANCED:SPATTERN:  
PATTERN "1100110111101111"  
:TRIGGER:ENHANCED:SPATTERN:  
PATTERN? -> :TRIGGER:ENHANCED:SPATTERN:  
PATTERN "1100110111101111"

#### **:TRIGger:ENHanced:SPIBus?**

機能 SPI バストリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:SPIBus?  
例 :TRIGGER:ENHANCED:SPIBUS?

-> :TRIGGER:ENHANCED:SPIBUS:  
BITORDER LSBFIRST;CLOCK:  
POLARITY FALL;SOURCE 1;:TRIGGER:  
ENHANCED:SPIBUS:CS:ACTIVE HIGH;  
SOURCE 1;:TRIGGER:ENHANCED:SPIBUS:  
DATA1:BYTE 1;CONDITION TRUE;  
DPOSITION 1;PATTERN1 "00010010";  
PATTERN2 "00110100";  
PATTERN3 "01010110";  
PATTERN4 "00010010";SOURCE 3;:  
TRIGGER:ENHANCED:SPIBUS:DATA2:BYTE 4;  
CONDITION TRUE;DPOSITION 1;  
PATTERN1 "00010010";  
PATTERN2 "00110100";  
PATTERN3 "01010110";  
PATTERN4 "00010010";SOURCE 3;:  
TRIGGER:ENHANCED:SPIBUS:MODE WIRE3

#### **:TRIGger:ENHanced:SPIBus:BITorder**

機能 SPI バストリガのビットオーダーを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:SPIBus:  
BITorder {LSBFirst|MSBFirst}  
:TRIGger:ENHanced:SPIBus:BITorder?

例 :TRIGGER:ENHANCED:SPIBUS:  
BITORDER LSBFIRST  
:TRIGGER:ENHANCED:SPIBUS:BITORDER?  
-> :TRIGGER:ENHANCED:SPIBUS:  
BITORDER LSBFIRST

#### **:TRIGger:ENHanced:SPIBus:CLOCK?**

機能 SPI バストリガのクロックに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:SPIBus:CLOCK?

例 :TRIGGER:ENHANCED:SPIBUS:CLOCK?  
-> :TRIGGER:ENHANCED:SPIBUS:CLOCK:  
POLARITY FALL;SOURCE 1

#### **:TRIGger:ENHanced:SPIBus:CLOCK:POLarity**

機能 SPI バストリガのクロックトレースの極性を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:SPIBus:CLOCK:  
POLarity {FALL|RISE}  
:TRIGger:ENHanced:SPIBus:CLOCK:  
POLarity?

例 :TRIGGER:ENHANCED:SPIBUS:CLOCK:  
POLARITY FALL  
:TRIGGER:ENHANCED:SPIBUS:CLOCK:  
POLARITY? -> :TRIGGER:ENHANCED:  
SPIBUS:CLOCK:POLARITY FALL

**:TRIGger:ENHanced:SPIBus:CLOCK:****SOURCE**

機能 SPIバストリガのクロックトレースを設定/問い合わせします。

構文 :TRIGger:ENHanced:SPIBus:CLOCK:  
SOURCE {<Nrf>}  
:TRIGger:ENHanced:SPIBus:CLOCK:  
SOURCE?

<Nrf> = 1 ~ 4

例 :TRIGGER:ENHANCED:SPIBUS:CLOCK:  
SOURCE 1  
:TRIGGER:ENHANCED:SPIBUS:CLOCK:  
SOURCE? -> :TRIGGER:ENHANCED:SPIBUS:  
CLOCK:SOURCE 1

**:TRIGger:ENHanced:SPIBus:CS?**

機能 SPIバストリガのチップセレクトに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:SPIBus:CS?  
例 :TRIGGER:ENHANCED:SPIBUS:CS?

-> :TRIGGER:ENHANCED:SPIBUS:CS:  
ACTIVE HIGH;SOURCE 1

**:TRIGger:ENHanced:SPIBus:CS:Active**

機能 SPIバストリガのチップセレクトのアクティブレベルを設定/問い合わせします。

構文 :TRIGger:ENHanced:SPIBus:CS:  
Active {HIGH|LOW}  
:TRIGger:ENHanced:SPIBus:CS:Active?

例 :TRIGGER:ENHANCED:SPIBUS:CS:ACTIVE  
HIGH  
:TRIGGER:ENHANCED:SPIBUS:CS:ACTIVE?  
-> :TRIGGER:ENHANCED:SPIBUS:CS:  
ACTIVE HIGH

**:TRIGger:ENHanced:SPIBus:CS:SOURCE**

機能 SPIバストリガのチップセレクトトレースを設定/問い合わせします。

構文 :TRIGger:ENHanced:SPIBus:CS:  
SOURCE {<Nrf>}  
:TRIGger:ENHanced:SPIBus:CS:SOURCE?  
<Nrf> = 1 ~ 4

例 :TRIGGER:ENHANCED:SPIBUS:CS:SOURCE 1  
:TRIGGER:ENHANCED:SPIBUS:CS:SOURCE?  
-> :TRIGGER:ENHANCED:SPIBUS:CS:  
SOURCE 1

**:TRIGger:ENHanced:SPIBus:DATA<x>?**

機能 SPIバストリガの各データに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:SPIBus:DATA<x>?  
<x> = 1, 2

例 :TRIGGER:ENHANCED:SPIBUS:DATA1? -> :  
TRIGGER:ENHANCED:SPIBUS:DATA1:BYTE 1;  
CONDITION TRUE;DPOSITION 1;  
PATTERN1 "00010010";  
PATTERN2 "00110100";  
PATTERN3 "01010110";  
PATTERN4 "00010010";SOURCE 3

解説 DATA2は、「:TRIGger:ENHanced:SPIBus:  
MODE WIRE4」のときに有効です。

**:TRIGger:ENHanced:SPIBus:DATA<x>:****BYTE**

機能 SPIバストリガの各データの設定データ数を設定/問い合わせします。

構文 :TRIGger:ENHanced:SPIBus:DATA<x>:  
BYTE {<Nrf>}  
:TRIGger:ENHanced:SPIBus:DATA<x>:  
BYTE?

<x> = 1, 2

<Nrf> = 1 ~ 4

例 :TRIGGER:ENHANCED:SPIBUS:DATA1:BYTE 1  
:TRIGGER:ENHANCED:SPIBUS:DATA1:BYTE?  
-> :TRIGGER:ENHANCED:SPIBUS:DATA1:  
BYTE 1

**:TRIGger:ENHanced:SPIBus:DATA<x>:****CONDition**

機能 SPIバストリガの各データの判定方法(一致/不一致)を設定/問い合わせします。

構文 :TRIGger:ENHanced:SPIBus:DATA<x>:  
CONDition {FALSE|TRUE}  
:TRIGger:ENHanced:SPIBus:DATA<x>:  
CONDition?

<x> = 1, 2

例 :TRIGGER:ENHANCED:SPIBUS:DATA1:  
CONDITION TRUE  
:TRIGGER:ENHANCED:SPIBUS:DATA1:  
CONDITION? -> :TRIGGER:ENHANCED:  
SPIBUS:DATA1:CONDITION TRUE

### 5.30 TRIGger グループ

#### **:TRIGger:ENHanced:SPIBus:DATA<x>:DPOsition**

機能 SPIバストリガの各データのパターン比較先頭位置を設定/問い合わせします。

構文 :TRIGger:ENHanced:SPIBus:DATA<x>:  
DPOsition {<NRf>}  
:TRIGger:ENHanced:SPIBus:DATA<x>:  
DPOsition?  
<x> = 1、2  
<NRf> = 0 ~ 9999

例 :TRIGGER:ENHANCED:SPIBUS:DATA1:  
DPOSITION 1  
:TRIGGER:ENHANCED:SPIBUS:DATA1:  
DPOSITION? -> :TRIGGER:ENHANCED:  
SPIBUS:DATA1:DPOSITION 1

#### **:TRIGger:ENHanced:SPIBus:DATA<x>:HEXA<x>**

機能 SPIバストリガの各データを HEXA で設定します。

構文 :TRIGger:ENHanced:SPIBus:DATA<x>:  
HEXA<x> {<文字列>}  
DATA<x> の <x> = 1、2  
HEXA<x> の <x> = 1 ~ 4  
<文字列> = '0' ~ 'F'、'X' の組み合わせ 2 文字。

例 :TRIGGER:ENHANCED:SPIBUS:DATA1:  
HEXA1 "AB"

#### **:TRIGger:ENHanced:SPIBus:DATA<x>:PATtern<x>**

機能 SPIバストリガの各データを BINARY で設定/問い合わせします。

構文 :TRIGger:ENHanced:SPIBus:DATA<x>:  
PATtern<x> {<文字列>}  
:TRIGger:ENHanced:SPIBus:DATA<x>:  
PATtern<x>?  
DATA<x> の <x> = 1、2  
PATtern<x> の <x> = 1 ~ 4  
<文字列> = '0'、'1'、'X' の組み合わせ 8 文字

例 :TRIGGER:ENHANCED:SPIBUS:DATA1:  
PATTERN1 "10101011"  
:TRIGGER:ENHANCED:SPIBUS:DATA1:  
PATTERN1? -> :TRIGGER:ENHANCED:  
SPIBUS:DATA1:PATTERN1 "10101011"

#### **:TRIGger:ENHanced:SPIBus:DATA<x>:SOURCE**

機能 SPIバストリガの各データのトレースを設定/問い合わせします。

構文 :TRIGger:ENHanced:SPIBus:DATA<x>:  
SOURCE {<NRf>}  
:TRIGger:ENHanced:SPIBus:DATA<x>:  
SOURCE?  
<x> = 1、2  
<NRf> = 1 ~ 4

例 :TRIGGER:ENHANCED:SPIBUS:DATA1:  
SOURCE 1  
:TRIGGER:ENHANCED:SPIBUS:DATA1:  
SOURCE? -> :TRIGGER:ENHANCED:SPIBUS:  
DATA1:SOURCE 1

#### **:TRIGger:ENHanced:SPIBus::MODE**

機能 SPIバストリガの結線方式 (3 線式 / 4 線式) を設定/問い合わせします。

構文 :TRIGger:ENHanced:SPIBus:MODE {WIRE3|  
WIRE4}  
:TRIGger:ENHanced:SPIBus:MODE?

例 :TRIGGER:ENHANCED:SPIBUS:MODE WIRE3  
:TRIGGER:ENHANCED:SPIBUS:MODE?  
-> :TRIGGER:ENHANCED:SPIBUS:  
MODE WIRE3

#### **:TRIGger:ENHanced:TV?**

機能 TVトリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:TV?  
例 :TRIGGER:ENHANCED:TV? -> :TRIGGER:  
ENHANCED:TV:CUSTOMIZE 1;  
FIELD DONTCARE;FRAME 2;HDTV:LINE 2;  
POLARITY NEGATIVE;;TRIGGER:ENHANCED:  
TV:LEVEL 1.000E+00;NTSC:LINE 5;  
POLARITY NEGATIVE;;TRIGGER:ENHANCED:  
TV:PAL:LINE 2;POLARITY NEGATIVE;;  
TRIGGER:ENHANCED:TV:SDTV:LINE 8;  
POLARITY NEGATIVE;;TRIGGER:  
ENHANCED:TV:SGUARD 60;SOURCE 1;TYPE  
HDTV;USERDEFINE:DEFINITION HD;  
HFREJECTION OFF;HSYNC 50.00E+06;  
LINE 2;POLARITY NEGATIVE

#### **:TRIGger:ENHanced:TV:COUPLing?**

機能 TVトリガのトリガカップリングを問い合わせます。

構文 :TRIGger:ENHanced:TV:COUPLing?

例 :TRIGGER:ENHANCED:TV:COUPLING?  
-> :TRIGGER:ENHANCED:TV:COUPLING TV

**:TRIGger:ENHanced:TV:CUSTomize**

機能 TVトリガの Sync Guard 機能の ON/OFF を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:TV:  
CUSTomize {<Boolean>}

例 :TRIGger:ENHanced:TV:CUSTomize?  
:TRIGGER:ENHANCED:TV:CUSTOMIZE ON  
:TRIGGER:ENHANCED:TV:CUSTOMIZE?  
-> :TRIGGER:ENHANCED:TV:CUSTOMIZE 1

**:TRIGger:ENHanced:TV:FIELD**

機能 TVトリガのフィールドを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:TV:FIELD {DONTcare|  
<Nrf>}  
:TRIGger:ENHanced:TV:FIELD?  
<Nrf> = 1、2

例 :TRIGGER:ENHANCED:TV:FIELD DONTCARE  
:TRIGGER:ENHANCED:TV:FIELD?  
-> :TRIGGER:ENHANCED:TV:  
FIELD DONTCARE

**:TRIGger:ENHanced:TV:FRAME**

機能 TVトリガのフレームスキップ機能を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:TV:FRAME {<Nrf>}  
:TRIGger:ENHanced:TV:FRAME?  
<Nrf> = 1、2、4、8

例 :TRIGGER:ENHANCED:TV:FRAME 2  
:TRIGGER:ENHANCED:TV:FRAME?  
-> :TRIGGER:ENHANCED:TV:FRAME 2

**:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|SDTV|USERdefine}?**

機能 TVトリガのモードに関するすべての設定値を問い合わせます。

構文 :TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|  
SDTV|USERdefine}?

例 (以下は、HDTV についての例です。)  
:TRIGGER:ENHANCED:TV:HDTV?  
-> :TRIGGER:ENHANCED:TV:HDTV:LINE 2;  
POLARITY NEGATIVE

**:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|SDTV}:HFRejection?  
(HighFrequencyREJECTION)**

機能 TVトリガのローパスフィルタ (HF リジェクション) を問い合わせます。

構文 TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|  
SDTV}:HFRejection?

例 (以下は、HDTV についての例です。)  
:TRIGGER:ENHANCED:TV:HDTV:  
HFREJECTION? -> :TRIGGER:ENHANCED:TV:  
HDTV:HFREJECTION OFF

**:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|SDTV|USERdefine}:LINE**

機能 TVトリガをかけるラインを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|  
SDTV|USERdefine}:LINE

例 :TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|  
SDTV|USERdefine}:LINE {<Nrf>}  
<Nrf> = 2 ~ 2251(HDTV の場合)  
5 ~ 1054(NTSC の場合)  
2 ~ 1251(PAL の場合)  
8 ~ 2251(SDTV の場合)  
2 ~ 2251(USERdefine の場合)

例 (以下は、HDTV についての例です。)  
:TRIGGER:ENHANCED:TV:HDTV:LINE 10  
:TRIGGER:ENHANCED:TV:HDTV:LINE?  
-> :TRIGGER:ENHANCED:TV:HDTV:LINE 10

**:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|SDTV|USERdefine}:POLarity**

機能 TVトリガの入力の極性を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|  
SDTV|USERdefine}:POLarity {NEGative|  
POSitive}  
:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|  
SDTV|USERdefine}:POLarity?

例 (以下は、HDTV についての例です。)  
:TRIGGER:ENHANCED:TV:HDTV:  
POLARITY NEGATIVE  
:TRIGGER:ENHANCED:TV:HDTV:POLARITY?  
-> :TRIGGER:ENHANCED:TV:HDTV:  
POLARITY NEGATIVE

**:TRIGger:ENHanced:TV:LEVel**

機能 TVトリガのトリガレベルを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:TV:LEVel {<Nrf>}  
:TRIGger:ENHanced:TV:LEVel?  
<Nrf> = 0.1 ~ 2.0(div)

例 :TRIGGER:ENHANCED:TV:LEVEL 1  
:TRIGGER:ENHANCED:TV:LEVEL?  
-> :TRIGGER:ENHANCED:TV:  
LEVEL 1.000E+00

**:TRIGger:ENHanced:TV:SGUard**

機能 TVトリガの Sync Guard を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:TV:SGUard {<Nrf>}  
:TRIGger:ENHanced:TV:SGUard?  
<Nrf> = 60 ~ 90(%)

例 :TRIGGER:ENHANCED:TV:SGUARD 60  
:TRIGGER:ENHANCED:TV:SGUARD?  
-> :TRIGGER:ENHANCED:TV:SGUARD 60

解説 「:TRIGGER:ENHANCED:TV:TYPE HDTV|NTSC|  
PAL」のときは有効です。



## 5.30 TRIGger グループ

### **:TRIGger:ENHanced:TV:SOURce**

機能 TV トリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:TV:SOURce {<Nrf>}  
:TRIGger:ENHanced:TV:SOURce?  
<Nrf> = 1 ~ 4

例 :TRIGGER:ENHANCED:TV:SOURCE 1  
:TRIGGER:ENHANCED:TV:SOURCE?  
-> :TRIGGER:ENHANCED:TV:SOURCE 1

### **:TRIGger:ENHanced:TV:TYPE**

機能 TV トリガの入力の種類を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:TV:TYPE {HDTV|NTSC|PAL|SDTV|USERdefine,I1080\_50|I1080\_60|P1080\_24|P1080\_25|P1080\_60|P720\_60|SF1080\_24}  
:TRIGger:ENHanced:TV:TYPE?

例 :TRIGGER:ENHANCED:TV:TYPE NTSC  
:TRIGGER:ENHANCED:TV:TYPE?  
-> TRIGGER:ENHANCED:TV:TYPE NTSC

解説 {HDTV} の場合は、次の {I1080\_50|I1080\_60|P1080\_24|P1080\_25|P1080\_60|P720\_60|SF1080\_24} を選択します。選択しない場合は自動的に {I1080\_60} を選択します。

### **:TRIGger:ENHanced:TV:USERdefine:DEFinition**

機能 ユーザー定義の解像度を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:TV:USERdefine:DEFinition {HD|SD}  
:TRIGger:ENHanced:TV:USERdefine:DEFinition?

例 :TRIGGER:ENHANCED:TV:USERDEFINE:DEFINITION HD  
:TRIGGER:ENHANCED:TV:USERDEFINE:DEFINITION? -> :TRIGGER:ENHANCED:TV:USERDEFINE:DEFINITION HD

### **:TRIGger:ENHanced:TV:USERdefine:HFRejection (HighFrequencyREJECTION)**

機能 ユーザー定義のローパスフィルタ (HF リジエクション) を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:TV:USERdefine:HFRejection {<周波数>|OFF}  
:TRIGger:ENHanced:TV:USERdefine:HFRejection?

例 <周波数> = 300kHz  
:TRIGGER:ENHANCED:TV:USERDEFINE:HFREJECTION OFF  
:TRIGGER:ENHANCED:TV:USERDEFINE:HFREJECTION? -> :TRIGGER:ENHANCED:TV:USERDEFINE:HFREJECTION OFF

### **:TRIGger:ENHanced:TV:USERdefine:HSYNc (Hsync Freq)**

機能 ユーザー定義の水平同期信号の周波数を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:TV:USERdefine:HSYNc {<周波数>}  
:TRIGger:ENHanced:TV:USERdefine:HSYNc?

例 <周波数> = 10k ~ 200k(Hz)  
:TRIGGER:ENHANCED:TV:USERDEFINE:HSYNC 10KHZ  
:TRIGGER:ENHANCED:TV:USERDEFINE:HSYNC? -> :TRIGGER:ENHANCED:TV:USERDEFINE:HSYNC 10.00E+03

### **:TRIGger:ENHanced:UART?**

機能 UART 信号トリガに関するすべて設定値を問い合わせします。

構文 :TRIGger:ENHanced:UART?

例 :TRIGGER:ENHANCED:UART? -> :TRIGGER:ENHANCED:UART:BRATE 19200;  
FORMAT BIT7PARITY;POLARITY NEGATIVE;  
SOURCE 1;SPOINT 18.8E+00

### **:TRIGger:ENHanced:UART:BRATE**

機能 UART 信号トリガのビットレート (データ転送速度) を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:UART:BRATE {<Nrf>|USER,<Nrf>}  
:TRIGger:ENHanced:UART:BRATE?  
<Nrf> = 1200、2400、4800、9600、19200、38400、57600、115200  
USER の <Nrf> = 本体ユーザーズマニュアル参照。

例 :TRIGGER:ENHANCED:UART:BRATE 19200  
:TRIGGER:ENHANCED:UART:BRATE?  
-> :TRIGGER:ENHANCED:UART:BRATE 19200

### **:TRIGger:ENHanced:UART:FORMat**

機能 UART 信号トリガのフォーマットを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:UART:FORMat {BIT7parity|BIT8Noparity|BIT8Parity}  
:TRIGger:ENHanced:UART:FORMat?

例 :TRIGGER:ENHANCED:UART:FORMAT BIT7PARITY  
:TRIGGER:ENHANCED:UART:FORMAT? -> :TRIGGER:ENHANCED:UART:FORMAT BIT7PARITY

**:TRIGger:ENHanced:UART:POLarity**

機能 UART 信号トリガの極性を設定 / 問い合わせします。

構文 :TRIGger:ENHanced:UART:  
POLarity {NEGative|POSitive}  
:TRIGger:ENHanced:UART:POLarity?

例 :TRIGGER:ENHANCED:UART:  
POLARITY NEGATIVE  
:TRIGGER:ENHANCED:UART:POLARITY?  
-> :TRIGGER:ENHANCED:UART:  
POLARITY NEGATIVE

**:TRIGger:ENHanced:UART:SOURce**

機能 UART 信号トリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:UART:SOURce {<Nrf>}  
:TRIGger:ENHanced:UART:SOURce?  
<Nrf> = 1 ~ 4

例 :TRIGGER:ENHANCED:UART:SOURCE 1  
:TRIGGER:ENHANCED:UART:SOURCE?  
-> :TRIGGER:ENHANCED:UART:SOURCE 1

**:TRIGger:ENHanced:UART:SPOint**

機能 UART 信号トリガのサンプルポイントを設定 / 問い合わせします。

構文 :TRIGger:ENHanced:UART:SPOint {<Nrf>}  
:TRIGger:ENHanced:UART:SPOint?  
<Nrf> = 18.8 ~ 90.6(%)

例 :TRIGGER:ENHANCED:UART:SPOINT 18.8  
:TRIGGER:ENHANCED:UART:SPOINT?  
-> :TRIGGER:ENHANCED:UART:  
SPOINT 18.8E+00

**:TRIGger:ESTate?**

機能 Edge/State トリガに関するすべての設定値を問い合わせします。

構文 :TRIGger:ESTate?  
例 :TRIGGER:ESTATE? -> :TRIGGER:ESTATE:  
EOR:CHANNEL1 DONTCARE;  
CHANNEL2 DONTCARE;CHANNEL3 DONTCARE;  
CHANNEL4 DONTCARE;;TRIGGER:ESTATE:  
SOURCE 1;POLARITY ENTER

**:TRIGger:ESTate:EOR?**

機能 OR トリガに関するすべての設定値を問い合わせします。

構文 :TRIGger:ESTate:EOR?  
例 :TRIGGER:ESTATE:EOR? -> :TRIGGER:  
ESTATE:EOR:CHANNEL1 DONTCARE;  
CHANNEL2 DONTCARE;CHANNEL3 DONTCARE;  
CHANNEL4 DONTCARE

**:TRIGger:ESTate:EOR:CHANnel<x>**

機能 OR トリガの各チャンネルの極性を設定 / 問い合わせします。

構文 :TRIGger:ESTate:EOR:  
CHANnel<x> {DONTcare|ENTER|EXIT|FALL|  
RISE}

:TRIGger:ESTate:EOR:CHANnel<x>?  
<x> = 1 ~ 4

例 :TRIGGER:ESTATE:EOR:CHANNEL1 DONTCARE  
:TRIGGER:ESTATE:EOR:CHANNEL1?  
-> :TRIGGER:ESTATE:EOR:  
CHANNEL1 DONTCARE

解説  
・「:TRIGger:TYPE EOR」のときに有効です。  
・「:TRIGger:SOURce:CHANnel<x>:WINDOW ON」のときは {ENTER|EXIT}、それ以外のときは {FALL|RISE} が有効です。

**:TRIGger:ESTate:POLarity**

機能 Edge/State トリガの極性を設定 / 問い合わせします。

構文 :TRIGger:ESTate:POLarity {ENTER|EXIT|  
FALL|RISE}

:TRIGger:ESTate:POLarity?  
例 :TRIGGER:ESTATE:POLARITY ENTER  
:TRIGGER:ESTATE:POLARITY?

-> :TRIGGER:ESTATE:POLARITY ENTER  
解説  
・「:TRIGger:TYPE EDGE|EQUALify|STATE」のときに有効です。  
・「:TRIGger:TYPE EDGE」且つ「:TRIGger:ESTate:SOURce LINE」のときは無効です。  
・「:TRIGger:TYPE EDGE|EQUALify」且つ「:TRIGger:SOURce:CHANnel<x>:WINDOW ON」のときは {ENTER|EXIT}、「:TRIGger:SOURce:CHANnel<x>:WINDOW OFF」のときは {FALL|RISE} が有効です。  
・「:TRIGger:TYPE STATE」のときは {ENTER|EXIT} が有効です。

**:TRIGger:ESTate:SOURce**

機能 Edge/State トリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger:ESTate:SOURce {<Nrf>|  
EXTERNAL|LINE}

:TRIGger:ESTate:SOURce?  
<Nrf> = 1 ~ 4

例 :TRIGGER:ESTATE:SOURCE EXTERNAL  
:TRIGGER:ESTATE:SOURCE? -> :TRIGGER:  
ESTATE:SOURCE EXTERNAL

解説  
・「:TRIGger:TYPE EDGE|EQUALify」のときに有効です。  
・「:TRIGger:TYPE EDGE」のときは {<Nrf>|EXTERNAL|LINE} が有効です。  
・「:TRIGger:TYPE EQUALify」のときは {<Nrf>|EXTERNAL} が有効です。

## 5.30 TRIGger グループ

### :TRIGger:HOLDoff

機能 ホールドオフ時間を設定 / 問い合わせします。  
構文 :TRIGger:HOLDoff {<時間>}  
:TRIGger:HOLDoff?  
<時間> = 20ns ~ 10s(5ns ステップ)  
例 :TRIGGER:HOLDOFF 1S  
:TRIGGER:HOLDOFF? -> :TRIGGER:  
HOLDOFF 1.000E+00

### :TRIGger:LOGic?

機能 ロジックトリガに関するすべての設定値を問い合わせます。  
構文 :TRIGger:LOGic?  
例 :TRIGGER:LOGIC? -> :TRIGGER:LOGIC:  
CLOCK:POLARITY RISE;SOURCE A0;;  
TRIGGER:LOGIC:ESTATE:POLARITY RISE;  
SOURCE A0;;TRIGGER:LOGIC:I2CBUS:  
ADATA:BIT10ADDRESS:  
PATTERN "10111011111";;  
TRIGGER:LOGIC:I2CBUS:ADATA:  
BIT7ADDRESS:PATTERN "11011110";;  
TRIGGER:LOGIC:I2CBUS:ADATA:BIT7APSUB:  
ADDRESS:PATTERN "10101011";:TRIGGER:  
LOGIC:I2CBUS:ADATA:BIT7APSUB:  
SADDRESS:PATTERN "10101011";:TRIGGER:  
LOGIC:I2CBUS:ADATA:  
TYPE BIT10ADDRESS;:TRIGGER:LOGIC:  
I2CBUS:CLOCK:SOURCE A0;:TRIGGER:  
LOGIC:I2CBUS:DATA:BYTE 1;  
CONDITION FALSE;DPOSITION 1;MODE 1;  
PATTERN1 "10101011";  
PATTERN2 "XXXXXXXX";  
PATTERN3 "XXXXXXXX";  
PATTERN4 "XXXXXXXX";PMODE DONTCARE;  
SOURCE A0;:TRIGGER:LOGIC:I2CBUS:  
GCALL:BIT7MADDRESS:  
PATTERN "1010101";:TRIGGER:LOGIC:  
I2CBUS:GCALL:SBYTE BIT7MADDRESS;:  
TRIGGER:LOGIC:I2CBUS:MODE ADATA;  
NAIGNORE:HSMODE 1;RACCESS 1;SBYTE 1;:  
TRIGGER:LOGIC:I2CBUS:SBHSMODE:  
TYPE HSMODE;:TRIGGER:LOGIC:LINBUS:  
BRATE 19200;SOURCE A0;:TRIGGER:LOGIC:  
SPATTERN:BITRATE 1.000000E+03;CLOCK:  
MODE 1;POLARITY FALL;SOURCE A0;:  
TRIGGER:LOGIC:SPATTERN:CS 1;DATA:  
ACTIVE HIGH;SOURCE A0;:TRIGGER:LOGIC:  
SPATTERN:LATCH:SOURCE A0;  
POLARITY FALL;:TRIGGER:LOGIC:  
SPATTERN:PATTERN "1100110111101111";:  
TRIGGER:LOGIC:SPIBUS:  
BITORDER LSBFIRST;CLOCK:  
POLARITY FALL;SOURCE A0;:TRIGGER:  
LOGIC:SPIBUS:CS:ACTIVE HIGH .....

### :TRIGger:LOGic:CLOCK?

機能 ロジックトリガのクロックに関するすべての設定値を問い合わせます。  
構文 :TRIGger:LOGic:CLOCK?  
例 :TRIGGER:LOGIC:CLOCK? -> :TRIGGER:  
LOGIC:CLOCK:POLARITY RISE;  
SOURCE A0

### :TRIGger:LOGic:CLOCK:POLarity

機能 ロジックトリガのクロックの極性を設定 / 問い合わせします。  
構文 :TRIGger:LOGic:CLOCK:POLarity {FALL|  
RISE}  
:TRIGger:LOGic:CLOCK:POLarity?  
例 :TRIGGER:LOGIC:CLOCK:POLARITY FALL  
:TRIGGER:LOGIC:CLOCK:POLARITY?  
-> :TRIGGER:LOGIC:CLOCK:POLARITY FALL

### :TRIGger:LOGic:CLOCK:SOURce

機能 ロジックトリガのクロックのソースを設定 / 問い合わせします。  
構文 :TRIGger:LOGic:CLOCK:SOURce  
{A<x>|B<x>|  
C<x>|D<x>|DONTcare}  
:TRIGger:LOGic:CLOCK:SOURce?  
<x> = 0 ~ 7  
例 :TRIGGER:LOGIC:CLOCK:SOURCE A0  
:TRIGGER:LOGIC:CLOCK:SOURCE?  
-> :TRIGGER:LOGIC:CLOCK:SOURCE A0  
解説 DL9510L、DL9505L は {A<x>|C<x>|DONTcare} が有効です。

### :TRIGger:LOGic:ESTate?

機能 ロジックの Edge/State トリガに関するすべての設定値を問い合わせます。  
構文 :TRIGger:LOGic:ESTate?  
例 :TRIGGER:LOGIC:ESTATE? -> :TRIGGER:  
LOGIC:ESTATE:POLARITY RISE;  
SOURCE A0

### :TRIGger:LOGic:ESTate:POLarity

機能 ロジックの Edge/State トリガの極性を設定 / 問い合わせします。  
構文 :TRIGger:LOGic:ESTate:POLarity  
{ENTer|  
EXIT|FALL|RISE}  
:TRIGger:LOGic:ESTate:POLarity?  
例 :TRIGGER:LOGIC:ESTATE:POLARITY ENTER  
:TRIGGER:LOGIC:ESTATE:POLARITY?  
-> :TRIGGER:LOGIC:ESTATE:  
POLARITY ENTER  
解説  
・「:TRIGger:TYPE LState」のときは、  
{ENTer|EXIT} が有効です。  
・「:TRIGger:TYPE LState」以外のときは、  
{FALL|RISE} が有効です。

**:TRIGger:LOGic:ESTate:SOURce**

機能 ロジックの Edge/State トリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger:LOGic:ESTate:SOURce {A<x>|B<x>|C<x>|D<x>}  
:TRIGger:LOGic:ESTate:SOURce?  
<x> = 0 ~ 7

例 :TRIGGER:LOGIC:ESTATE:SOURCE A0  
:TRIGGER:LOGIC:ESTATE:SOURCE?  
-> :TRIGGER:LOGIC:ESTATE:SOURCE A0

解説 DL9510L、DL9505L は {A<x>|C<x>} が有効です。

**:TRIGger:LOGic:I2Cbus?**

機能 ロジック I<sup>2</sup>C バストリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:LOGic:I2Cbus?

例 :TRIGGER:LOGIC:I2CBUS? -> :TRIGGER:  
LOGIC:I2CBUS:ADATA:BIT10ADDRESS:  
PATTERN "10111011111";:TRIGGER:LOGIC:  
I2CBUS:ADATA:BIT7ADDRESS:  
PATTERN "11011110";:TRIGGER:LOGIC:  
I2CBUS:ADATA:BIT7APSUB:ADDRESS:  
PATTERN "10101011";:TRIGGER:LOGIC:  
I2CBUS:ADATA:BIT7APSUB:SADDRESS:  
PATTERN "10101011";:TRIGGER:LOGIC:  
I2CBUS:ADATA:TYPE BIT10ADDRESS;:  
TRIGGER:LOGIC:I2CBUS:CLOCK:  
SOURCE A0;:TRIGGER:LOGIC:I2CBUS:DATA:  
BYTE 1;CONDITION FALSE;DPOSITION 1;  
MODE 1;PATTERN1 "10101011";  
PATTERN2 "XXXXXXXX";  
PATTERN3 "XXXXXXXX";  
PATTERN4 "XXXXXXXX";PMODE DONTCARE;  
SOURCE A0;:TRIGGER:LOGIC:I2CBUS:  
GCALL:BIT7MADDRESS:  
PATTERN "1010101";:TRIGGER:LOGIC:  
I2CBUS:GCALL:SBYTE BIT7MADDRESS;:  
TRIGGER:LOGIC:I2CBUS:MODE ADATA;  
NAIGNORE:HSMODE 1;RACCESS 1;SBYTE 1;:  
TRIGGER:LOGIC:I2CBUS:SBHSMODE:  
TYPE HSMODE

**:TRIGger:LOGic:I2Cbus:ADATa?**

機能 ロジック I<sup>2</sup>C バストリガのアドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:LOGic:I2Cbus:ADATa?

例 :TRIGGER:LOGIC:I2CBUS:ADATa?  
-> :TRIGGER:LOGIC:I2CBUS:ADATa:  
BIT10ADDRESS:PATTERN "10111011111";:  
TRIGGER:LOGIC:I2CBUS:ADATa:  
BIT7ADDRESS:PATTERN "11011110";:  
TRIGGER:LOGIC:I2CBUS:ADATa:BIT7APSUB:  
ADDRESS:PATTERN "10101011";:TRIGGER:  
LOGIC:I2CBUS:ADATa:BIT7APSUB:  
SADDRESS:PATTERN "10101011";:TRIGGER:  
LOGIC:I2CBUS:ADATa:TYPE BIT10ADDRESS

**:TRIGger:LOGic:I2Cbus:ADATa:  
BIT10address?**

機能 ロジック I<sup>2</sup>C バストリガの 10bit アドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:LOGic:I2Cbus:ADATa:  
BIT10address?

例 :TRIGGER:LOGIC:I2CBUS:ADATa:  
BIT10ADDRESS? -> :TRIGGER:LOGIC:  
I2CBUS:ADATa:BIT10ADDRESS:  
PATTERN "10111011111"

**:TRIGger:LOGic:I2Cbus:ADATa:  
BIT10address:HEXA**

機能 ロジック I<sup>2</sup>C バストリガの 10bit アドレスを HEXA で設定します。

構文 :TRIGger:LOGic:I2Cbus:ADATa:  
BIT10address:HEXA {<文字列>}  
<文字列> = '0' ~ 'F', 'X' の組み合わせ 3 文字 (ビット 8 は、R/W ビット)

例 :TRIGGER:LOGIC:I2CBUS:ADATa:  
BIT10ADDRESS:HEXA "7AB"

**:TRIGger:LOGic:I2Cbus:ADATa:  
BIT10address:PATtern**

機能 ロジック I<sup>2</sup>C バストリガの 10bit アドレスを BINARY で設定 / 問い合わせします。

構文 :TRIGger:LOGic:I2Cbus:ADATa:  
BIT10address:PATtern {<文字列>}  
:TRIGger:LOGic:I2Cbus:ADATa:  
BIT10address:PATtern?

<文字列> = '0', '1', 'X' の組み合わせ 11 文字 (ビット 8 は、R/W ビット)

例 :TRIGGER:LOGIC:I2CBUS:ADATa:  
BIT10ADDRESS:PATTERN "10111011111"  
:TRIGGER:LOGIC:I2CBUS:ADATa:  
BIT10ADDRESS:PATTERN? -> :TRIGGER:  
LOGIC:I2CBUS:ADATa:BIT10ADDRESS:  
PATTERN "10111011111"

**:TRIGger:LOGic:I2Cbus:ADATa:  
BIT7Address?**

機能 ロジック I<sup>2</sup>C バストリガの 7bit アドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:LOGic:I2Cbus:ADATa:  
BIT7Address?

例 :TRIGGER:LOGIC:I2CBUS:ADATa:  
BIT7ADDRESS? -> :TRIGGER:LOGIC:  
I2CBUS:ADATa:BIT7ADDRESS:  
PATTERN "11011110"

## 5.30 TRIGger グループ

### **:TRIGger:LOGic:I2CBus:ADATa:**

#### **BIT7Address:HEXA**

機能 ロジック I<sup>2</sup>C バストリガの 7bit アドレスを HEXA で設定します。

構文 :TRIGger:LOGic:I2CBus:ADATa:  
BIT7Address:HEXA {<文字列>}  
<文字列> = '0' ~ 'F', 'X' の組み合わせ 2 文字 (ビット 0 は、R/W ビット)

例 :TRIGGER:LOGIC:I2CBUS:ADATA:  
BIT7ADDRESS:HEXA "DE"

### **:TRIGger:LOGic:I2CBus:ADATa:**

#### **BIT7Address:PATtern**

機能 ロジック I<sup>2</sup>C バストリガの 7bit アドレスを BINARY で設定 / 問い合わせします。

構文 :TRIGger:LOGic:I2CBus:ADATa:  
BIT7Address:PATtern {<文字列>}  
:TRIGger:LOGic:I2CBus:ADATa:  
BIT7Address:PATtern?  
<文字列> = '0', '1', 'X' の組み合わせ 8 文字 (ビット 0 は、R/W ビット)

例 :TRIGGER:LOGIC:I2CBUS:ADATA:  
BIT7ADDRESS:PATTERN "11011110"  
:TRIGGER:LOGIC:I2CBUS:ADATA:  
BIT7ADDRESS:PATTERN? -> :TRIGGER:  
LOGIC:I2CBUS:ADATA:BIT7ADDRESS:  
PATTERN "11011110"

### **:TRIGger:LOGic:I2CBus:ADATa:**

#### **BIT7APsub?**

機能 ロジック I<sup>2</sup>C バストリガの 7bit + Sub アドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:LOGic:I2CBus:ADATa:  
BIT7APsub?

例 :TRIGGER:LOGIC:I2CBUS:ADATA:  
BIT7APSUB? -> :TRIGGER:LOGIC:I2CBUS:  
ADATA:BIT7APSUB:ADDRESS:  
PATTERN "10101011"; :TRIGGER:LOGIC:  
I2CBUS:ADATA:BIT7APSUB:SADDRESS:  
PATTERN "10101011"

### **:TRIGger:LOGic:I2CBus:ADATa:**

#### **BIT7APsub:ADDRESS?**

機能 ロジック I<sup>2</sup>C バストリガの 7bit + Sub アドレスの 7bit アドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:LOGic:I2CBus:ADATa:  
BIT7APsub:ADDRESS?

例 :TRIGGER:LOGIC:I2CBUS:ADATA:  
BIT7APSUB:ADDRESS? -> :TRIGGER:LOGIC:  
I2CBUS:ADATA:BIT7APSUB:ADDRESS:  
PATTERN "10101011"

### **:TRIGger:LOGic:I2CBus:ADATa:**

#### **BIT7APsub:ADDRESS:HEXA**

機能 ロジック I<sup>2</sup>C バストリガの 7bit + Sub アドレスの 7bit アドレスを HEXA で設定します。

構文 :TRIGger:LOGic:I2CBus:ADATa:  
BIT7APsub:ADDRESS:HEXA {<文字列>}  
<文字列> = '0' ~ 'F', 'X' の組み合わせ 2 文字 (ビット 0 は、R/W ビット)

例 :TRIGGER:LOGIC:I2CBUS:ADATA:  
BIT7APSUB:ADDRESS:HEXA "AB"

### **:TRIGger:LOGic:I2CBus:ADATa:**

#### **BIT7APsub:ADDRESS:PATtern**

機能 ロジック I<sup>2</sup>C バストリガの 7bit + Sub アドレスの 7bit アドレスを BINARY で設定 / 問い合わせします。

構文 :TRIGger:LOGic:I2CBus:ADATa:  
BIT7APsub:ADDRESS:PATtern {<文字列>}  
:TRIGger:LOGic:I2CBus:ADATa:  
BIT7APsub:ADDRESS:PATtern?  
<文字列> = '0', '1', 'X' の組み合わせ 8 文字 (ビット 0 は、R/W ビット)

例 :TRIGGER:LOGIC:I2CBUS:ADATA:  
BIT7APSUB:ADDRESS:PATTERN "10101011"  
:TRIGGER:LOGIC:I2CBUS:ADATA:  
BIT7APSUB:ADDRESS:PATTERN?  
-> :TRIGGER:LOGIC:I2CBUS:ADATA:  
BIT7APSUB:ADDRESS:PATTERN "10101011"

### **:TRIGger:LOGic:I2CBus:ADATa:**

#### **BIT7APsub:SADDRESS?**

機能 ロジック I<sup>2</sup>C バストリガの 7bit + Sub アドレスの Sub アドレスに関するすべての設定値を問い合わせます。

構文 :TRIGger:LOGic:I2CBus:ADATa:  
BIT7APsub:SADDRESS?

例 :TRIGGER:LOGIC:I2CBUS:ADATA:  
BIT7APSUB:SADDRESS? -> :TRIGGER:  
LOGIC:I2CBUS:ADATA:BIT7APSUB:  
SADDRESS:PATTERN "10101011"

### **:TRIGger:LOGic:I2CBus:ADATa:**

#### **BIT7APsub:SADDRESS:HEXA**

機能 ロジック I<sup>2</sup>C バストリガの 7bit + Sub アドレスの Sub アドレスを HEXA で設定します。

構文 :TRIGger:LOGic:I2CBus:ADATa:  
BIT7APsub:SADDRESS:HEXA {<文字列>}  
<文字列> = '0' ~ 'F', 'X' の組み合わせ 2 文字

例 :TRIGGER:LOGIC:I2CBUS:ADATA:  
BIT7APSUB:SADDRESS:HEXA "EF"

**:TRIGger:LOGic:I2CBUS:ADATa:****BIT7APsub:SADdResS:PATtern**

機能 ロジック I<sup>2</sup>C バストリガの 7bit + Sub アドレスの Sub アドレスを BINARY で設定 / 問い合わせします。

構文 :TRIGger:LOGic:I2CBUS:ADATa:  
BIT7APsub:SADdResS:PATtern {<文字列>}  
:TRIGger:LOGic:I2CBUS:ADATa:

例 <文字列> = '0'、'1'、'X' の組み合わせ 8 文字  
:TRIGGER:LOGIC:I2CBUS:ADATa:  
BIT7APSUB:SADDRESS:PATTERN "10101011"  
:TRIGGER:LOGIC:I2CBUS:ADATa:  
BIT7APSUB:SADDRESS:PATTERN?  
-> :TRIGGER:LOGIC:I2CBUS:ADATa:  
BIT7APSUB:SADDRESS:PATTERN "10101011"

**:TRIGger:LOGic:I2CBUS:ADATa:TYPE**

機能 ロジック I<sup>2</sup>C バストリガのアドレスの種類を設定 / 問い合わせします。

構文 :TRIGger:LOGic:I2CBUS:ADATa:TYPE  
{BIT10address|BIT7Address|BIT7APsub}  
:TRIGger:LOGic:I2CBUS:ADATa:TYPE?

例 :TRIGGER:LOGIC:I2CBUS:ADATa:  
TYPE BIT10ADDRESS  
:TRIGGER:LOGIC:I2CBUS:ADATa:TYPE?  
-> :TRIGGER:LOGIC:I2CBUS:ADATa:  
TYPE BIT10ADDRESS

**:TRIGger:LOGic:I2CBUS:CLOCK?**

機能 ロジック I<sup>2</sup>C バストリガのクロックに関するすべての設定値を問い合わせます。

構文 :TRIGger:LOGic:I2CBUS:CLOCK?

例 :TRIGGER:LOGIC:I2CBUS:CLOCK?  
-> :TRIGGER:LOGIC:I2CBUS:CLOCK:  
SOURCE A0

**:TRIGger:LOGic:I2CBUS:CLOCK:SOURce**

機能 ロジック I<sup>2</sup>C バストリガのクロックトレースを設定 / 問い合わせします。

構文 :TRIGger:LOGic:I2CBUS:CLOCK:SOURce  
{A<x>|B<x>|C<x>|D<x>}  
:TRIGger:LOGic:I2CBUS:CLOCK:SOURce?  
<x> = 0 ~ 7

例 :TRIGGER:LOGIC:I2CBUS:CLOCK:SOURCE A0  
:TRIGGER:LOGIC:I2CBUS:CLOCK:SOURCE?  
-> :TRIGGER:LOGIC:I2CBUS:CLOCK:  
SOURCE A0

解説 DL9510L、DL9505L は {A<x>|C<x>} が有効です。

**:TRIGger:LOGic:I2CBUS:DATA?**

機能 ロジック I<sup>2</sup>C バストリガのデータに関するすべての設定値を問い合わせます。

構文 :TRIGger:LOGic:I2CBUS:DATA?

例 :TRIGGER:LOGIC:I2CBUS:DATA?  
-> :TRIGGER:LOGIC:I2CBUS:DATA:BYTE 1;  
CONDITION FALSE;DPOSITION 1;MODE 1;  
PATTERN1 "10101011";  
PATTERN2 "XXXXXXXX";  
PATTERN3 "XXXXXXXX";  
PATTERN4 "XXXXXXXX";PMODE DONTCARE;  
SOURCE A0

**:TRIGger:LOGic:I2CBUS:DATA:BYTE**

機能 ロジック I<sup>2</sup>C バストリガの設定データ数を設定 / 問い合わせします。

構文 :TRIGger:LOGic:I2CBUS:DATA:BYTE  
{<Nrf>}

:TRIGGER:LOGIC:I2CBUS:DATA:BYTE?  
<Nrf> = 1 ~ 4

例 :TRIGGER:LOGIC:I2CBUS:DATA:BYTE 1  
:TRIGGER:LOGIC:I2CBUS:DATA:BYTE?  
-> :TRIGGER:LOGIC:I2CBUS:DATA:BYTE 1

**:TRIGger:LOGic:I2CBUS:DATA:CONDition**

機能 ロジック I<sup>2</sup>C バストリガのデータの判定方法 (一致 / 不一致) を設定 / 問い合わせします。

構文 :TRIGger:LOGic:I2CBUS:DATA:CONDition  
{FALSE|TRUE}  
:TRIGger:LOGic:I2CBUS:DATA:CONDition?

例 :TRIGGER:LOGIC:I2CBUS:DATA:  
CONDITION FALSE  
:TRIGGER:LOGIC:I2CBUS:DATA:CONDITION?  
-> :TRIGGER:LOGIC:I2CBUS:DATA:  
CONDITION FALSE

**:TRIGger:LOGic:I2CBUS:DATA:DPOSITION**

機能 ロジック I<sup>2</sup>C バストリガのデータのパターン比較する位置を設定 / 問い合わせします。

構文 :TRIGger:LOGic:I2CBUS:DATA:DPOSITION  
{<Nrf>}  
:TRIGger:LOGic:I2CBUS:DATA:DPOSITION?  
<Nrf> = 0 ~ 9999

例 :TRIGGER:LOGIC:I2CBUS:DATA:  
DPOSITION 1  
:TRIGGER:LOGIC:I2CBUS:DATA:DPOSITION?  
-> :TRIGGER:LOGIC:I2CBUS:DATA:  
DPOSITION 1

### 5.30 TRIGger グループ

**:TRIGger:LOGic:I2CBus:DATA:HEXA<x>**  
機能 ロジック I<sup>2</sup>C バストリガのデータを HEXA で設定  
します。  
構文 :TRIGger:LOGic:I2CBus:DATA:HEXA<x> {<  
文字列>}  
<x> = 1 ~ 4  
<文字列> = '0' ~ 'F', 'X' の組み合わせ 2 文字  
例 :TRIGGER:LOGIC:I2CBUS:DATA:HEXA1 "AB"

**:TRIGger:LOGic:I2CBus:DATA:MODE**  
機能 ロジック I<sup>2</sup>C バストリガのデータ条件の有効 / 無  
効を設定 / 問い合わせします。  
構文 :TRIGger:LOGic:I2CBus:DATA:MODE  
{<Boolean>}  
<Boolean> = :TRIGger:LOGic:I2CBus:DATA:MODE?  
例 :TRIGGER:LOGIC:I2CBUS:DATA:MODE ON  
:TRIGGER:LOGIC:I2CBUS:DATA:MODE?  
-> :TRIGGER:LOGIC:I2CBUS:DATA:MODE 1

**:TRIGger:LOGic:I2CBus:DATA:  
PATtern<x>**  
機能 ロジック I<sup>2</sup>C バストリガのデータを BINARY で設  
定 / 問い合わせします。  
構文 :TRIGger:LOGic:I2CBus:DATA:PATtern<x>  
{<文字列>}  
<文字列> = '0', '1', 'X' の組み合わせ 8 文字  
例 :TRIGGER:LOGIC:I2CBUS:DATA:  
PATTERN1 "10101011"  
:TRIGGER:LOGIC:I2CBUS:DATA:PATTERN1?  
-> :TRIGGER:LOGIC:I2CBUS:DATA:  
PATTERN1 "10101011"

**:TRIGger:LOGic:I2CBus:DATA:PMODE**  
機能 ロジック I<sup>2</sup>C バストリガのデータのパターン比較  
先頭位置モードを設定 / 問い合わせします。  
構文 :TRIGger:LOGic:I2CBus:DATA:PMODE  
{DONTcare|SElect}  
例 :TRIGGER:LOGIC:I2CBUS:DATA:  
PMODE DONTCARE  
:TRIGGER:LOGIC:I2CBUS:DATA:PMODE?  
-> :TRIGGER:LOGIC:I2CBUS:DATA:  
PMODE DONTCARE

**:TRIGger:LOGic:I2CBus:DATA:SOURce**  
機能 ロジック I<sup>2</sup>C バストリガのデータトレースを設定  
 / 問い合わせします。  
構文 :TRIGger:LOGic:I2CBus:DATA:SOURce  
{A<x>|B<x>|C<x>|D<x>}  
<x> = 0 ~ 7  
例 :TRIGGER:LOGIC:I2CBUS:DATA:SOURCE A0  
:TRIGGER:LOGIC:I2CBUS:DATA:SOURCE?  
-> :TRIGGER:LOGIC:I2CBUS:DATA:  
SOURCE A0  
解説 DL9510L、DL9505L は {A<x>|C<x>} が有効です。

**:TRIGger:LOGic:I2CBus:GCALl?**  
機能 ロジック I<sup>2</sup>C バストリガのジェネラルコールに関  
するすべての設定値を問い合わせます。  
構文 :TRIGger:LOGic:I2CBus:GCALl?  
例 :TRIGGER:LOGIC:I2CBUS:GCALL?  
-> :TRIGGER:LOGIC:I2CBUS:GCALL:  
BIT7MADDRESS:PATTERN "1010101";;  
TRIGGER:LOGIC:I2CBUS:GCALL:  
SBYTE BIT7MADDRESS

**:TRIGger:LOGic:I2CBus:GCALl:  
BIT7maddress?**  
機能 ロジック I<sup>2</sup>C バストリガのジェネラルコールの  
7bit マスタアドレスに関するすべての設定値を問  
い合わせます。  
構文 :TRIGger:LOGic:I2CBus:GCALl:  
BIT7maddress?  
例 :TRIGGER:LOGIC:I2CBUS:GCALL:  
BIT7MADDRESS? -> :TRIGGER:LOGIC:  
I2CBUS:GCALL:BIT7MADDRESS:  
PATTERN "1010101"

**:TRIGger:LOGic:I2CBus:GCALl:  
BIT7maddress:HEXA**  
機能 ロジック I<sup>2</sup>C バストリガのジェネラルコールの  
7bit マスタアドレスを HEXA で設定します。  
構文 :TRIGger:LOGic:I2CBus:GCALl:  
BIT7maddress:HEXA {<文字列>}  
<文字列> = '0' ~ 'F', 'X' の組み合わせ 2 文字 (ビッ  
ト 0 は、'1' に固定)  
例 :TRIGGER:LOGIC:I2CBUS:GCALL:  
BIT7MADDRESS:HEXA "AB"

**:TRIGger:LOGic:I2Cbus:GCALl:****BIT7maddress:PATtern**

機能 ロジック I<sup>2</sup>C バストリガのジェネラルコールの 7bit マスタアドレスを BINARY で設定 / 問い合わせします。

構文 :TRIGger:LOGic:I2Cbus:GCALl:  
BIT7maddress:PATtern {<文字列>}  
:TRIGger:LOGic:I2Cbus:GCALl:  
BIT7maddress:PATtern?

例 <文字列> = '0'、'1'、'X' の組み合わせ 7 文字  
:TRIGGER:LOGIC:I2CBUS:GCALL:  
BIT7MADDRESS:PATTERN "1010101"  
:TRIGGER:LOGIC:I2CBUS:GCALL:  
BIT7MADDRESS:PATTERN? -> :TRIGGER:  
LOGIC:I2CBUS:GCALL:BIT7MADDRESS:  
PATTERN "1010101"

**:TRIGger:LOGic:I2Cbus:GCALl:SBYTE  
(Second Byte)**

機能 ロジック I<sup>2</sup>C バストリガのジェネラルコールのセカンドバイトのタイプを設定 / 問い合わせします。

構文 :TRIGger:LOGic:I2Cbus:GCALl:SBYTE  
{BIT7maddress|DONTCare|H04|H06}  
:TRIGger:LOGic:I2Cbus:GCALl:SBYTE?

例 :TRIGGER:LOGIC:I2CBUS:GCALL:  
SBYTE BIT7MADDRESS  
:TRIGGER:LOGIC:I2CBUS:GCALL:SBYTE?  
-> :TRIGGER:LOGIC:I2CBUS:GCALL:  
SBYTE BIT7MADDRESS

**:TRIGger:LOGic:I2Cbus:MODE**

機能 ロジック I<sup>2</sup>C バストリガのトリガモードを設定 / 問い合わせします。

構文 :TRIGger:LOGic:I2Cbus:MODE {ADATa|  
ESTart|GCALl|NAIGNore|SBHSmode}  
:TRIGger:LOGic:I2Cbus:MODE?

例 :TRIGGER:LOGIC:I2CBUS:MODE ADATA  
:TRIGGER:LOGIC:I2CBUS:MODE?  
-> :TRIGGER:LOGIC:I2CBUS:MODE ADATA

**:TRIGger:LOGic:I2Cbus:NAIGNore?**

機能 ロジック I<sup>2</sup>C バストリガの NON ACK 無視モードに関するすべての設定値を問い合わせます。

構文 :TRIGger:LOGic:I2Cbus:NAIGNore?

例 :TRIGGER:LOGIC:I2CBUS:NAIGNORE?  
-> :TRIGGER:LOGIC:I2CBUS:NAIGNORE:  
HSMODE 1;RACCESS 1;SBYTE 1

**:TRIGger:LOGic:I2Cbus:NAIGNore:****HSMode**

機能 ロジック I<sup>2</sup>C バストリガのハイスピードモードで NON ACK を無視する / しないを設定 / 問い合わせします。

構文 :TRIGger:LOGic:I2Cbus:NAIGNore:HSMode  
{<Boolean>}  
:TRIGger:LOGic:I2Cbus:NAIGNore:  
HSMode?

例 :TRIGGER:LOGIC:I2CBUS:NAIGNORE:  
HSMODE ON  
:TRIGGER:LOGIC:I2CBUS:NAIGNORE:  
HSMODE? -> :TRIGGER:LOGIC:I2CBUS:  
NAIGNORE:HSMODE 1

**:TRIGger:LOGic:I2Cbus:NAIGNore:****RACcess**

機能 ロジック I<sup>2</sup>C バストリガのリードアクセスモードで NON ACK を無視する / しないを設定 / 問い合わせします。

構文 :TRIGger:LOGic:I2Cbus:NAIGNore:  
RACcess {<Boolean>}  
:TRIGger:LOGic:I2Cbus:NAIGNore:  
RACcess?

例 :TRIGGER:LOGIC:I2CBUS:NAIGNORE:  
RACCESS ON  
:TRIGGER:LOGIC:I2CBUS:NAIGNORE:  
RACCESS? -> :TRIGGER:LOGIC:I2CBUS:  
NAIGNORE:RACCESS 1

**:TRIGger:LOGic:I2Cbus:NAIGNore:SBYTE  
(Start Byte)**

機能 ロジック I<sup>2</sup>C バストリガのスタートバイトで NON ACK を無視する / しないを設定 / 問い合わせします。

構文 :TRIGger:LOGic:I2Cbus:NAIGNore:SBYTE  
{<Boolean>}  
:TRIGger:LOGic:I2Cbus:NAIGNore:SBYTE?

例 :TRIGGER:LOGIC:I2CBUS:NAIGNORE:  
SBYTE ON  
:TRIGGER:LOGIC:I2CBUS:NAIGNORE:SBYTE?  
-> :TRIGGER:LOGIC:I2CBUS:NAIGNORE:  
SBYTE 1

**:TRIGger:LOGic:I2Cbus:SBHSmode?**

機能 ロジック I<sup>2</sup>C バストリガのスタートバイト / ハイスピードモードに関するすべての設定値を問い合わせます。

構文 :TRIGger:LOGic:I2Cbus:SBHSmode?

例 :TRIGGER:LOGIC:I2CBUS:SBHSMODE?  
-> :TRIGGER:LOGIC:I2CBUS:SBHSMODE:  
TYPE HSMODE



### 5.30 TRIGger グループ

**:TRIGger:LOGic:I2CBus:SBHSmode:TYPE**  
機能 ロジック I<sup>2</sup>C バストリガのスタートバイト / ハイ  
スピードモードのタイプを設定 / 問い合わせしま  
す。  
構文 :TRIGger:LOGic:I2CBus:SBHSmode:TYPE  
{HSMMode|SBYTE}  
例 :TRIGger:LOGic:I2CBus:SBHSmode:TYPE?  
:TRIGGER:LOGIC:I2CBUS:SBHSMODE:  
TYPE HSMODE  
:TRIGGER:LOGIC:I2CBUS:SBHSMODE:TYPE?  
-> :TRIGGER:LOGIC:I2CBUS:SBHSMODE:  
TYPE HSMODE

**:TRIGger:LOGic:LINBus?**  
機能 ロジック LIN バス信号トリガに関するすべての設  
定値を問い合わせします。  
構文 :TRIGger:LOGic:LINBus?  
例 :TRIGGER:LOGIC:LINBUS? -> :TRIGGER:  
LOGIC:LINBUS:BRATE 19200;  
SOURCE A0

**:TRIGger:LOGic:LINBus:BRATE**  
機能 ロジック LIN バス信号トリガのビットレート  
(データ転送速度)を設定 / 問い合わせします。  
構文 :TRIGger:LOGic:LINBus:BRATE {<Nrf>|  
USER,<Nrf>}  
:TRIGger:LOGic:LINBus:BRATE?  
<Nrf> = 1200、2400、4800、9600、19200  
USER の <Nrf> = 本体ユーザーズマニュアル参照。  
例 :TRIGGER:LOGIC:LINBUS:BRATE 19200  
:TRIGGER:LOGIC:LINBUS:BRATE?  
-> :TRIGGER:LOGIC:LINBUS:BRATE 19200

**:TRIGger:LOGic:LINBus:SOURce**  
機能 ロジック LIN バス信号トリガのトリガソースを設  
定 / 問い合わせします。  
構文 :TRIGger:LOGic:LINBus:SOURce {A<x>|  
B<x>|C<x>|D<x>}  
:TRIGger:LOGic:LINBus:SOURce?  
<x> = 0 ~ 7  
例 :TRIGGER:LOGIC:LINBUS:SOURCE A0  
:TRIGGER:LOGIC:LINBUS:SOURCE?  
-> :TRIGGER:LOGIC:LINBUS:SOURCE A0  
解説 DL9510L、DL9505L は {A<x>|C<x>} が有効です。

**:TRIGger:LOGic:SPATtern? (Serial  
Pattern)**  
機能 ロジックシリアルパターントリガに関するすべ  
ての設定値を問い合わせます。  
構文 :TRIGger:LOGic:SPATtern?  
例 :TRIGGER:LOGIC:SPATTERN?  
-> :TRIGGER:LOGIC:SPATTERN:  
BITRATE 1.0000000E+03;CLOCK:MODE 1;  
POLARITY FALL;SOURCE A0;:TRIGGER:  
LOGIC:SPATTERN:CS 1;DATA:ACTIVE HIGH;  
SOURCE A0;:TRIGGER:LOGIC:SPATTERN:  
LATCH:SOURCE A0;POLARITY FALL;:  
TRIGGER:LOGIC:SPATTERN:  
PATTERN "1100110111101111"

**:TRIGger:LOGic:SPATtern:BITRate**  
機能 ロジックシリアルパターントリガのビットレ  
ートを設定 / 問い合わせします。  
構文 :TRIGger:LOGic:SPATtern:  
BITRate {<Nrf>}  
:TRIGger:LOGic:SPATtern:BITRate?  
<Nrf> = 1 ~ 50M(bps)  
例 :TRIGGER:LOGIC:SPATTERN:BITRATE 1  
:TRIGGER:LOGIC:SPATTERN:BITRATE?  
-> :TRIGGER:LOGIC:SPATTERN:  
BITRATE 1.000E+00  
解説 「:TRIGger:LOGic:SPATtern:CLOCK:  
MODEOFF」のときに有効です。

**:TRIGger:LOGic:SPATtern:CLEar**  
機能 ロジックシリアルパターントリガのパターンを  
すべてクリア (Don't care) します。  
構文 :TRIGger:LOGic:SPATtern:CLEar  
例 :TRIGGER:LOGIC:SPATTERN:CLEAR

**:TRIGger:LOGic:SPATtern:CLOCK?**  
機能 ロジックシリアルパターントリガのクロックに  
関するすべての設定値を問い合わせます。  
構文 :TRIGger:LOGic:SPATtern:CLOCK?  
例 :TRIGGER:LOGIC:SPATTERN:CLOCK?  
-> :TRIGGER:LOGIC:SPATTERN:CLOCK:  
MODE 1;POLARITY FALL;SOURCE A0

**:TRIGger:LOGic:SPATtern:CLOCK:MODE**  
機能 ロジックシリアルパターントリガのクロックの  
有効 / 無効を設定 / 問い合わせします。  
構文 :TRIGger:LOGic:SPATtern:CLOCK:MODE  
{<Boolean>}  
:TRIGger:LOGic:SPATtern:CLOCK:MODE?  
例 :TRIGGER:LOGIC:SPATTERN:CLOCK:MODE ON  
:TRIGGER:LOGIC:SPATTERN:CLOCK:MODE?  
-> :TRIGGER:LOGIC:SPATTERN:CLOCK:  
MODE 1

**:TRIGger:LOGic:SPATtern:CLOCK:POLarity**

機能 ロジックシリアルパターントリガのクロックレースの極性を設定 / 問い合わせします。

構文 :TRIGger:LOGic:SPATtern:CLOCK:POLarity {FALL|RISE}  
:TRIGger:LOGic:SPATtern:CLOCK:POLarity?

例 :TRIGGER:LOGIC:SPATTERN:CLOCK:POLARITY FALL  
:TRIGGER:LOGIC:SPATTERN:CLOCK:POLARITY? -> :TRIGGER:LOGIC:SPATTERN:CLOCK:POLARITY FALL

解説 「:TRIGger:LOGic:SPATtern:CLOCK:MODE ON」のときに有効です。

**:TRIGger:LOGic:SPATtern:CLOCK:SOURce**

機能 ロジックシリアルパターントリガのクロックレースを設定 / 問い合わせします。

構文 :TRIGger:LOGic:SPATtern:CLOCK:SOURce {A<x>|B<x>|C<x>|D<x>}  
:TRIGger:LOGic:SPATtern:CLOCK:SOURce? <x> = 0 ~ 7

例 :TRIGGER:LOGIC:SPATTERN:CLOCK:SOURCE A0  
:TRIGGER:LOGIC:SPATTERN:CLOCK:SOURCE? -> :TRIGGER:LOGIC:SPATTERN:CLOCK:SOURCE A0

解説 ・「:TRIGger:LOGic:SPATtern:CLOCK:MODE ON」のときに有効です。  
・DL9510L、DL9505L は {A<x>|C<x>} が有効です。

**:TRIGger:LOGic:SPATtern:CS**

機能 ロジックシリアルパターントリガのチップセレクトの有効 / 無効を設定 / 問い合わせします。

構文 :TRIGger:LOGic:SPATtern:CS {<Boolean>}  
:TRIGger:LOGic:SPATtern:CS?

例 :TRIGGER:LOGIC:SPATTERN:CS ON  
:TRIGGER:LOGIC:SPATTERN:CS? -> :TRIGGER:LOGIC:SPATTERN:CS 1

解説 「:TRIGger:LOGic:SPATtern:CLOCK:MODE ON」のときに有効です。

**:TRIGger:LOGic:SPATtern:DATA?**

機能 ロジックシリアルパターントリガのデータに関するすべての設定値を問い合わせます。

構文 :TRIGger:LOGic:SPATtern:DATA?  
例 :TRIGGER:LOGIC:SPATTERN:DATA? -> :TRIGGER:LOGIC:SPATTERN:DATA:ACTIVE HIGH;SOURCE A0

**:TRIGger:LOGic:SPATtern:DATA:ACTIVE**

機能 ロジックシリアルパターントリガのデータのアクティブレベルを設定 / 問い合わせします。

構文 :TRIGger:LOGic:SPATtern:DATA:ACTIVE {HIGH|LOW}  
:TRIGger:LOGic:SPATtern:DATA:ACTIVE?

例 :TRIGGER:LOGIC:SPATTERN:DATA:ACTIVE HIGH  
:TRIGGER:LOGIC:SPATTERN:DATA:ACTIVE? -> :TRIGGER:LOGIC:SPATTERN:DATA:ACTIVE HIGH

**:TRIGger:LOGic:SPATtern:DATA:SOURce**

機能 ロジックシリアルパターントリガのデータレースを設定 / 問い合わせします。

構文 :TRIGger:LOGic:SPATtern:DATA:SOURce {A<x>|B<x>|C<x>|D<x>}  
:TRIGger:LOGic:SPATtern:DATA:SOURce? <x> = 0 ~ 7

例 :TRIGGER:LOGIC:SPATTERN:DATA:SOURCE A0  
:TRIGGER:LOGIC:SPATTERN:DATA:SOURCE? -> :TRIGGER:LOGIC:SPATTERN:DATA:SOURCE A0

解説 DL9510L、DL9505L は {A<x>|C<x>} が有効です。

**:TRIGger:LOGic:SPATtern:HEXA**

機能 ロジックシリアルパターントリガのパターンをHEXA で設定します。

構文 :TRIGger:LOGic:SPATtern:HEXA {<文字列>}  
<文字列> = '0' ~ 'F'、'X' の組み合わせ 32 文字以内。

例 :TRIGGER:LOGIC:SPATTERN:HEXA "ABCD"

**:TRIGger:LOGic:SPATtern:LATCh?**

機能 ロジックシリアルパターントリガのラッチに関するすべての設定値を問い合わせます。

構文 :TRIGger:LOGic:SPATtern:LATCh?  
例 :TRIGGER:LOGIC:SPATTERN:LATCH? -> :TRIGGER:LOGIC:SPATTERN:LATCH:SOURCE A0;POLARITY FALL

## 5.30 TRIGger グループ

**:TRIGger:LOGic:SPATtern:LATCh:POLarity**  
機能 ロジックシリアルパターントリガのラッチトレースの極性を設定 / 問い合わせします。  
構文 :TRIGger:LOGic:SPATtern:LATCh:POLarity {FALL|RISE}  
:TRIGger:LOGic:SPATtern:LATCh:POLarity?  
例 :TRIGGER:LOGIC:SPATTERN:LATCH:POLARITY FALL  
:TRIGGER:LOGIC:SPATTERN:LATCH:POLARITY? -> :TRIGGER:LOGIC:SPATTERN:LATCH:POLARITY FALL  
解説  
・「:TRIGger:LOGic:SPATtern:CLOCK:MODE ON」のときに有効です。  
・「:TRIGger:LOGic:SPATtern:LATCh:SOURce DONTcare」のときは無効です。

**:TRIGger:LOGic:SPATtern:LATCh:SOURce**  
機能 ロジックシリアルパターントリガのラッチトレースを設定 / 問い合わせします。  
構文 :TRIGger:LOGic:SPATtern:LATCh:SOURce {A<x>|B<x>|C<x>|D<x>|DONTcare}  
:TRIGger:LOGic:SPATtern:LATCh:SOURce?<x> = 0 ~ 7  
例 :TRIGGER:LOGIC:SPATTERN:LATCH:SOURCE A0  
:TRIGGER:LOGIC:SPATTERN:LATCH:SOURCE? -> :TRIGGER:LOGIC:SPATTERN:LATCH:SOURCE A0  
解説  
・「:TRIGger:LOGic:SPATtern:CLOCK:MODE ON」のときに有効です。  
・DL9510L、DL9505L は {A<x>|C<x>|DONTcare} が有効です。

**:TRIGger:LOGic:SPATtern:PATtern**  
機能 ロジックシリアルパターントリガのパターンをBINARY で設定 / 問い合わせします。  
構文 :TRIGger:LOGic:SPATtern:PATtern {<文字列>}  
:TRIGger:LOGic:SPATtern:PATtern?<文字列> = '0','1','X'の組み合わせ 128文字以内。  
例 :TRIGGER:LOGIC:SPATTERN:PATTERN "1100110111101111"  
:TRIGGER:LOGIC:SPATTERN:PATTERN? -> :TRIGGER:LOGIC:SPATTERN:PATTERN "1100110111101111"

**:TRIGger:LOGic:SPIBus?**  
機能 ロジック SPI バストリガに関するすべての設定値を問い合わせます。  
構文 :TRIGger:LOGic:SPIBus?  
例 :TRIGGER:LOGIC:SPIBUS? -> :TRIGGER:LOGIC:SPIBUS:BITORDER LSBFIRST;CLOCK:POLARITY FALL;SOURCE A0;;:TRIGGER:LOGIC:SPIBUS:CS:ACTIVE HIGH;SOURCE A0;;:TRIGGER:LOGIC:SPIBUS:DATA1:BYTE 1;CONDITION FALSE;DPOSITION 1;PATTERN1 "10101011";PATTERN2 "XXXXXXXX";PATTERN3 "XXXXXXXX";PATTERN4 "XXXXXXXX";SOURCE A0;;:TRIGGER:LOGIC:SPIBUS:DATA2:BYTE 1;CONDITION TRUE;DPOSITION 0;PATTERN1 "XXXXXXXX";PATTERN2 "XXXXXXXX";PATTERN3 "XXXXXXXX";PATTERN4 "XXXXXXXX";SOURCE A2;;:TRIGGER:LOGIC:SPIBUS:MODE WIRE3

**:TRIGger:LOGic:SPIBus:BITorder**  
機能 ロジック SPI バストリガのビットオーダーを設定 / 問い合わせします。  
構文 :TRIGger:LOGic:SPIBus:BITorder {LSBFirst|MSBFirst}  
:TRIGger:LOGic:SPIBus:BITorder?  
例 :TRIGGER:LOGIC:SPIBUS:BITORDER LSBFIRST  
:TRIGGER:LOGIC:SPIBUS:BITORDER? -> :TRIGGER:LOGIC:SPIBUS:BITORDER LSBFIRST

**:TRIGger:LOGic:SPIBus:CLOCK?**  
機能 ロジック SPI バストリガのクロックに関するすべての設定値を問い合わせます。  
構文 :TRIGger:LOGic:SPIBus:CLOCK?  
例 :TRIGGER:LOGIC:SPIBUS:CLOCK? -> :TRIGGER:LOGIC:SPIBUS:CLOCK:POLARITY FALL;SOURCE A0

**:TRIGger:LOGic:SPIBus:CLOCK:POLarity**  
機能 ロジック SPI バストリガのクロックトレースの極性を設定 / 問い合わせします。  
構文 :TRIGger:LOGic:SPIBus:CLOCK:POLarity {FALL|RISE}  
:TRIGger:LOGic:SPIBus:CLOCK:POLarity?  
例 :TRIGGER:LOGIC:SPIBUS:CLOCK:POLARITY FALL  
:TRIGGER:LOGIC:SPIBUS:CLOCK:POLARITY? -> :TRIGGER:LOGIC:SPIBUS:CLOCK:POLARITY FALL

**:TRIGger:LOGic:SPIBus:CLOCK:SOURce**  
 機能 ロジック SPI バストリガのクロックトレースを設定 / 問い合わせします。  
 構文 :TRIGger:LOGic:SPIBus:CLOCK:SOURce  
 {A<x>|B<x>|C<x>|D<x>}  
 :TRIGger:LOGic:SPIBus:CLOCK:SOURce?  
 <x> = 0 ~ 7  
 例 :TRIGGER:LOGIC:SPIBUS:CLOCK:SOURCE A0  
 :TRIGGER:LOGIC:SPIBUS:CLOCK:SOURCE?  
 -> :TRIGGER:LOGIC:SPIBUS:CLOCK:  
 SOURCE A0  
 解説 DL9510L、DL9505L は {A<x>|C<x>} が有効です。

**:TRIGger:LOGic:SPIBus:CS?**  
 機能 ロジック SPI バストリガのチップセレクトに関するすべての設定値を問い合わせます。  
 構文 :TRIGger:LOGic:SPIBus:CS?  
 例 :TRIGGER:LOGIC:SPIBUS:CS?  
 -> :TRIGGER:LOGIC:SPIBUS:CS:  
 ACTIVE HIGH;SOURCE A0

**:TRIGger:LOGic:SPIBus:CS:ACTive**  
 機能 ロジック SPI バストリガのチップセレクトのアクティブレベルを設定 / 問い合わせします。  
 構文 :TRIGger:LOGic:SPIBus:CS:  
 ACTive {HIGH|LOW}  
 :TRIGger:LOGic:SPIBus:CS:ACTive?  
 例 :TRIGGER:LOGIC:SPIBUS:CS:ACTIVE HIGH  
 :TRIGGER:LOGIC:SPIBUS:CS:ACTIVE?  
 -> :TRIGGER:LOGIC:SPIBUS:CS:  
 ACTIVE HIGH

**:TRIGger:LOGic:SPIBus:CS:SOURce**  
 機能 ロジック SPI バストリガのチップセレクトトレースを設定 / 問い合わせします。  
 構文 :TRIGger:LOGic:SPIBus:CS:SOURce  
 {A<x>|  
 B<x>|C<x>|D<x>}  
 :TRIGger:LOGic:SPIBus:CS:SOURce?  
 <x> = 0 ~ 7  
 例 :TRIGGER:LOGIC:SPIBUS:CS:SOURCE A0  
 :TRIGGER:LOGIC:SPIBUS:CS:SOURCE?  
 -> :TRIGGER:LOGIC:SPIBUS:CS:SOURCE A0  
 解説 DL9510L、DL9505L は {A<x>|C<x>} が有効です。

**:TRIGger:LOGic:SPIBus:DATA<x>?**  
 機能 ロジック SPI バストリガの各データに関するすべての設定値を問い合わせます。  
 構文 :TRIGger:LOGic:SPIBus:DATA<x>?  
 <x> = 1、2  
 例 :TRIGGER:LOGIC:SPIBUS:DATA1?  
 -> :TRIGGER:LOGIC:SPIBUS:DATA1:  
 BYTE 1;CONDITION FALSE;DPOSITION 1;  
 PATTERN1 "10101011";  
 PATTERN2 "XXXXXXXX";  
 PATTERN3 "XXXXXXXX";  
 PATTERN4 "XXXXXXXX";SOURCE A0

**:TRIGger:LOGic:SPIBus:DATA<x>:BYTE**  
 機能 ロジック SPI バストリガの各データの設定データ数を設定 / 問い合わせします。  
 構文 :TRIGger:LOGic:SPIBus:DATA<x>:  
 BYTE {<Nrf>}  
 :TRIGger:LOGic:SPIBus:DATA<x>:BYTE?  
 <x> = 1、2  
 <Nrf> = 1 ~ 4  
 例 :TRIGGER:LOGIC:SPIBUS:DATA1:BYTE 1  
 :TRIGGER:LOGIC:SPIBUS:DATA1:BYTE?  
 -> :TRIGGER:LOGIC:SPIBUS:DATA1:BYTE 1

**:TRIGger:LOGic:SPIBus:DATA<x>:CONDITION**  
 機能 ロジック SPI バストリガの各データの判定方法 (一致 / 不一致) を設定 / 問い合わせします。  
 構文 :TRIGger:LOGic:SPIBus:DATA<x>:  
 CONDition {FALSE|TRUE}  
 :TRIGger:LOGic:SPIBus:DATA<x>:  
 CONDition?  
 <x> = 1、2  
 例 :TRIGGER:LOGIC:SPIBUS:DATA1:  
 CONDITION FALSE  
 :TRIGGER:LOGIC:SPIBUS:DATA1:  
 CONDITION? -> :TRIGGER:LOGIC:SPIBUS:  
 DATA1:CONDITION FALSE

**:TRIGger:LOGic:SPIBus:DATA<x>:DPOSITION**  
 機能 ロジック SPI バストリガの各データのパターン比較先頭位置を設定 / 問い合わせします。  
 構文 :TRIGger:LOGic:SPIBus:DATA<x>:  
 DPOSITION {<Nrf>}  
 :TRIGger:LOGic:SPIBus:DATA<x>:  
 DPOSITION?  
 <x> = 1、2  
 <Nrf> = 0 ~ 9999  
 例 :TRIGGER:LOGIC:SPIBUS:DATA1:  
 DPOSITION 1  
 :TRIGGER:LOGIC:SPIBUS:DATA1:  
 DPOSITION? -> :TRIGGER:LOGIC:SPIBUS:  
 DATA1:DPOSITION 1

## 5.30 TRIGger グループ

**:TRIGger:LOGic:SPIBus:DATA<x>:HEXA<x>**  
機能 ロジック SPI バストリガの各データを HEXA で設定します。  
構文 :TRIGger:LOGic:SPIBus:DATA<x>:  
HEXA<x> {<文字列>}  
DATA<x> の <x> = 1、2  
HEXA<x> の <x> = 1 ~ 4  
<文字列> = '0' ~ 'F'、'X' の組み合わせ 2 文字。  
例 :TRIGGER:LOGIC:SPIBUS:DATA1:  
HEXA1 "AB"

**:TRIGger:LOGic:SPIBus:DATA<x>:PATtern<x>**  
機能 ロジック SPI バストリガの各データを BINARY で設定 / 問い合わせします。  
構文 :TRIGger:LOGic:SPIBus:DATA<x>:  
PATtern<x> {<文字列>}  
:TRIGger:LOGic:SPIBus:DATA<x>:  
PATtern<x>?  
DATA<x> の <x> = 1、2  
PATtern<x> の <x> = 1 ~ 4  
<文字列> = '0'、'1'、'X' の組み合わせ 8 文字。  
例 :TRIGGER:LOGIC:SPIBUS:DATA1:  
PATTERN1 "10101011"  
:TRIGGER:LOGIC:SPIBUS:DATA1:PATTERN1?  
-> :TRIGGER:LOGIC:SPIBUS:DATA1:  
PATTERN1 "10101011"

**:TRIGger:LOGic:SPIBus:DATA<x>:SOURce**  
機能 ロジック SPI バストリガの各データのトレースを設定 / 問い合わせします。  
構文 :TRIGger:LOGic:SPIBus:DATA<x>:SOURce  
{A<y>|B<y>|C<y>|D<y>}  
:TRIGger:LOGic:SPIBus:DATA<x>:SOURce?  
<x> = 1、2  
<y> = 0 ~ 7  
例 :TRIGGER:LOGIC:SPIBUS:DATA1:SOURCE A0  
:TRIGGER:LOGIC:SPIBUS:DATA1:SOURCE?  
-> :TRIGGER:LOGIC:SPIBUS:DATA1:  
SOURCE A0  
解説 DL9510L、DL9505L は {A<y>|C<y>} が有効です。

**:TRIGger:LOGic:SPIBus:MODE**  
機能 ロジック SPI バストリガの結線方式 (3 線式 / 4 線式) を設定 / 問い合わせします。  
構文 :TRIGger:LOGic:SPIBus:MODE {WIRE3|WIRE4}  
:TRIGger:LOGic:SPIBus:MODE?  
例 :TRIGGER:LOGIC:SPIBUS:MODE WIRE3  
:TRIGGER:LOGIC:SPIBUS:MODE?  
-> :TRIGGER:LOGIC:SPIBUS:MODE WIRE3

**:TRIGger:LOGic:STATE?**  
機能 ロジックステートトリガに関するすべての設定値を問い合わせます。  
構文 :TRIGger:LOGic:STATE?  
例 :TRIGGER:LOGIC:STATE? -> :TRIGGER:  
LOGIC:STATE:BIT:A0 DONTCARE;  
A1 DONTCARE;A2 DONTCARE;A3 DONTCARE;  
A4 DONTCARE;A5 DONTCARE;A6 DONTCARE;  
A7 DONTCARE;B0 DONTCARE;B1 DONTCARE;  
B2 DONTCARE;B3 DONTCARE;B4 DONTCARE;  
B5 DONTCARE;B6 DONTCARE;B7 DONTCARE;  
C0 DONTCARE;C1 DONTCARE;C2 DONTCARE;  
C3 DONTCARE;C4 DONTCARE;C5 DONTCARE;  
C6 DONTCARE;C7 DONTCARE;D0 DONTCARE;  
D1 DONTCARE;D2 DONTCARE;D3 DONTCARE;  
D4 DONTCARE;D5 DONTCARE;D6 DONTCARE;  
D7 DONTCARE;LOGIC AND;:TRIGGER:LOGIC:  
STATE:GROUP1:CONDITION DONTCARE;  
PATTERN "XXXXXXXX";:TRIGGER:LOGIC:  
STATE:GROUP2:CONDITION TRUE;  
PATTERN "XXXXXXXX";:TRIGGER:LOGIC:  
STATE:GROUP3:CONDITION TRUE;  
PATTERN "XXXXXXXX";:TRIGGER:LOGIC:  
STATE:GROUP4:CONDITION TRUE;  
PATTERN "XXXXXXXX";:TRIGGER:LOGIC:  
STATE:GROUP5:CONDITION DONTCARE;  
PATTERN "";:TRIGGER:LOGIC:STATE:  
TYPE BIT

**:TRIGger:LOGic:STATE:BIT?**  
機能 ロジックステートトリガのビットに関するすべての設定値を問い合わせます。  
構文 :TRIGger:LOGic:STATE:BIT?  
例 :TRIGGER:LOGIC:STATE:BIT?  
-> :TRIGGER:LOGIC:STATE:BIT:  
A0 DONTCARE;A1 DONTCARE;A2 DONTCARE;  
A3 DONTCARE;A4 DONTCARE;A5 DONTCARE;  
A6 DONTCARE;A7 DONTCARE;B0 DONTCARE;  
B1 DONTCARE;B2 DONTCARE;B3 DONTCARE;  
B4 DONTCARE;B5 DONTCARE;B6 DONTCARE;  
B7 DONTCARE;C0 DONTCARE;C1 DONTCARE;  
C2 DONTCARE;C3 DONTCARE;C4 DONTCARE;  
C5 DONTCARE;C6 DONTCARE;C7 DONTCARE;  
D0 DONTCARE;D1 DONTCARE;D2 DONTCARE;  
D3 DONTCARE;D4 DONTCARE;D5 DONTCARE;  
D6 DONTCARE;D7 DONTCARE;LOGIC AND

**:TRIGger:LOGic:STATe:BIT:{A<x>|B<x>|C<x>|D<x>}**

機能 ロジックステートトリガの各ビットの成立条件を設定 / 問い合わせします。

構文 :TRIGger:LOGic:STATe:BIT:{A<x>|B<x>|C<x>|D<x>} {DONTcare|HIGH|LOW}  
:TRIGger:LOGic:STATe:BIT:{A<x>|B<x>|C<x>|D<x>}?  
<x> = 0 ~ 7

例 :TRIGGER:LOGIC:STATE:BIT:A0 DONTCARE  
:TRIGGER:LOGIC:STATE:BIT:A0?  
-> :TRIGGER:LOGIC:STATE:BIT:  
A0 DONTCARE

解説 DL9510L、DL9505L は {A<x>|C<x>} が有効です。

**:TRIGger:LOGic:STATe:BIT:CLEar**

機能 ロジックステートトリガの各ビットの成立条件をすべてクリア (Don't care) します。

構文 :TRIGger:LOGic:STATe:BIT:CLEar  
例 :TRIGGER:LOGIC:STATE:BIT:CLEAR

**:TRIGger:LOGic:STATe:BIT:LOGic**

機能 ロジックステートトリガのロジックを設定 / 問い合わせします。

構文 :TRIGger:LOGic:STATe:BIT:LOGic  
{AND|OR}  
:TRIGger:LOGic:STATe:BIT:LOGic?

例 :TRIGGER:LOGIC:STATE:BIT:LOGIC AND  
:TRIGGER:LOGIC:STATE:BIT:LOGIC?  
-> :TRIGGER:LOGIC:STATE:BIT:LOGIC AND

**:TRIGger:LOGic:STATe:GROup<x>?**

機能 ロジックステートトリガの各グループに関するすべての設定値を問い合わせます。

構文 :TRIGger:LOGic:STATe:GROup<x>?  
<x> = 1 ~ 5

例 :TRIGGER:LOGIC:STATE:GROUP1?  
-> :TRIGGER:LOGIC:STATE:GROUP1:  
CONDITION DONTCARE;PATTERN "111100001  
11100001111000011110000"

**:TRIGger:LOGic:STATe:GROup<x>:CLEar**

機能 ロジックステートトリガの各グループの成立条件をすべてクリア (Don't care) します。

構文 :TRIGger:LOGic:STATe:GROup<x>:CLEar  
<x> = 1 ~ 5

例 :TRIGGER:LOGIC:STATE:GROUP1:CLEAR

**:TRIGger:LOGic:STATe:GROup<x>:CONDition**

機能 ロジックステートトリガの各グループの判定条件を設定 / 問い合わせします。

構文 :TRIGger:LOGic:STATe:GROup<x>:  
CONDition {DONTcare|TRUE}  
:TRIGger:LOGic:STATe:GROup<x>:  
CONDition?  
<x> = 1 ~ 5

例 :TRIGGER:LOGIC:STATE:GROUP1:  
CONDITION DONTCARE  
:TRIGGER:LOGIC:STATE:GROUP1:  
CONDITION? -> :TRIGGER:LOGIC:STATE:  
GROUP1:CONDITION DONTCARE

**:TRIGger:LOGic:STATe:GROup<x>:HEXA**

機能 ロジックステートトリガの各グループの成立条件を HEXA で設定します。

構文 :TRIGger:LOGic:STATe:GROup<x>:  
HEXA {<文字列>}  
<x> = 1 ~ 5  
<文字列> = '0' ~ 'F'、'X' の組み合わせ 8 文字以内

例 :TRIGGER:LOGIC:STATE:GROUP1:  
HEXA "1A3F24CD"

解説 「:LOGic:GROup<x>:MAPPing」で設定したビット配置数が多い場合は下位に "X" が設定されます。少ない場合は上位が設定されます。

**:TRIGger:LOGic:STATe:GROup<x>:PATTern**

機能 ロジックステートトリガの各グループの成立条件を BINARY で設定 / 問い合わせします。

構文 :TRIGger:LOGic:STATe:GROup<x>:  
PATTern {<文字列>}  
:TRIGger:LOGic:STATe:GROup<x>:  
PATTern?  
<x> = 1 ~ 5

例 :TRIGGER:LOGIC:STATE:GROUP1:  
PATTERN "1111000011110000111100001111  
0000"  
:TRIGGER:LOGIC:STATE:GROUP1:PATTERN?  
-> :TRIGGER:LOGIC:STATE:GROUP1:  
PATTERN "1111000011110000111100001111  
0000"

**:TRIGger:LOGic:STATe:GROup<x>:SYMBOL**

機能 ロジックステートトリガの各グループのシンボルアイテムを設定します。

構文 :TRIGger:LOGic:STATe:GROup<x>:  
SYMBOL {<文字列>}  
<x> = 1 ~ 5  
<文字列> = 16 文字以内

例 :TRIGGER:LOGIC:STATE:GROUP1:  
SYMBOL "TEST"

## 5.30 TRIGger グループ

### **:TRIGger:LOGic:STATe:TYPE**

機能 ロジックステートトリガの設定方法を設定 / 問い合わせします。

構文 :TRIGger:LOGic:STATe:TYPE {BIT|GRoup}  
:TRIGger:LOGic:STATe:TYPE?

例 :TRIGGER:LOGIC:STATE:TYPE BIT  
:TRIGGER:LOGIC:STATE:TYPE?  
-> :TRIGGER:LOGIC:STATE:TYPE BIT

### **:TRIGger:LOGic:UART?**

機能 ロジック UART 信号トリガに関するすべて設定値を問い合わせします。

構文 :TRIGger:LOGic:UART?  
例 :TRIGGER:LOGIC:UART? -> :TRIGGER:  
LOGIC:UART:BRATE 19200;  
FORMAT BIT7PARITY;POLARITY NEGATIVE;  
SOURCE A0;SPOINT 18.8E+00

### **:TRIGger:LOGic:UART:BRATE**

機能 ロジック UART 信号トリガのビットレート (データ転送速度) を設定 / 問い合わせします。

構文 :TRIGger:LOGic:UART:  
BRATE {<Nrf>|USER,<Nrf>}  
:TRIGger:LOGic:UART:BRATE?  
<Nrf> = 1200、2400、4800、9600、19200、  
38400、57600、115200  
USER の <Nrf> = 本体ユーザーズマニュアル参照。  
例 :TRIGGER:LOGIC:UART:BRATE 19200  
:TRIGGER:LOGIC:UART:BRATE?  
-> :TRIGGER:LOGIC:UART:BRATE 19200

### **:TRIGger:LOGic:UART:FORMat**

機能 ロジック UART 信号トリガのフォーマットを設定 / 問い合わせします。

構文 :TRIGger:LOGic:UART:  
FORMat {BIT7parity|BIT8Noparity|  
BIT8Parity}  
:TRIGger:LOGic:UART:FORMat?  
例 :TRIGGER:LOGIC:UART:FORMAT BIT7PARITY  
:TRIGGER:LOGIC:UART:FORMAT?  
-> :TRIGGER:LOGIC:UART:  
FORMAT BIT7PARITY

### **:TRIGger:LOGic:UART:POLarity**

機能 ロジック UART 信号トリガの極性を設定 / 問い合わせします。

構文 :TRIGger:LOGic:UART:  
POLarity {NEGative|POSitive}  
:TRIGger:LOGic:UART:POLarity?  
例 :TRIGGER:LOGIC:UART:POLARITY NEGATIVE  
:TRIGGER:LOGIC:UART:POLARITY?  
-> :TRIGGER:LOGIC:UART:  
POLARITY NEGATIVE

### **:TRIGger:LOGic:UART:SOURce**

機能 ロジック UART 信号トリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger:LOGic:UART:SOURce  
{A<x>|B<x>|C<x>|D<x>}  
:TRIGger:LOGic:UART:SOURce?  
<x> = 0 ~ 7

例 :TRIGGER:LOGIC:UART:SOURCE A0  
:TRIGGER:LOGIC:UART:SOURCE?  
-> :TRIGGER:LOGIC:UART:SOURCE A0  
解説 DL9510L、DL9505L は {A<x>|C<x>} が有効です。

### **:TRIGger:LOGic:UART:SPOint**

機能 ロジック UART 信号トリガのサンプルポイントを設定 / 問い合わせします。

構文 :TRIGger:LOGic:UART:SPOint {<Nrf>}  
:TRIGger:LOGic:UART:SPOint?  
<Nrf> = 18.8 ~ 90.6(%)  
例 :TRIGGER:LOGIC:UART:SPOINT 18.8  
:TRIGGER:LOGIC:UART:SPOINT?  
-> :TRIGGER:LOGIC:UART:  
SPOINT 18.8E+00

### **:TRIGger:LOGic:WIDTh?**

機能 ロジックパルス幅トリガのすべての設定値を問い合わせします。

構文 :TRIGger:LOGic:WIDTh?  
例 :TRIGGER:LOGIC:WIDTH? -> :TRIGGER:  
LOGIC:WIDTH:MODE OUT;  
POLARITY POSITIVE;SOURCE A0;  
TIME1 1.0000000E-09;TIME2  
1.0000000E-09

### **:TRIGger:LOGic:WIDTh:MODE**

機能 ロジックパルス幅トリガの判定モードを設定 / 問い合わせします。

構文 :TRIGger:LOGic:WIDTh:MODE  
{BETween|IN|  
NOTBetween|OUT|TIMEout}  
:TRIGger:LOGic:WIDTh:MODE?  
例 :TRIGGER:LOGIC:WIDTH:MODE BETWEEN  
:TRIGGER:LOGIC:WIDTH:MODE?  
-> :TRIGGER:LOGIC:WIDTH:MODE BETWEEN

**:TRIGger:LOGic:WIDTh:POLarity**

機能 ロジックパルス幅トリガの極性を設定 / 問い合わせします。

構文 :TRIGger:LOGic:WIDTh:POLarity {FALSE|NEGative|POSitive|TRUE}  
:TRIGger:LOGic:WIDTh:POLarity?

例 :TRIGGER:LOGIC:WIDTH:POLARITY FALSE  
:TRIGGER:LOGIC:WIDTH:POLARITY?  
-> :TRIGGER:LOGIC:WIDTH:  
POLARITY FALSE

解説 ・「:TRIGger:TYPE LPState」のときは、  
{FALSE|TRUE} が有効です。  
・「:TRIGger:TYPE LPULse」のときは、  
{NEGative|POSitive} が有効です。

**:TRIGger:LOGic:WIDTh:SOURce**

機能 ロジックパルス幅トリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger:LOGic:WIDTh:SOURce  
{A<x>|B<x>|  
C<x>|D<x>}  
:TRIGger:LOGic:WIDTh:SOURce?  
<x> = 0 ~ 7

例 :TRIGGER:LOGIC:WIDTH:SOURCE A0  
:TRIGGER:LOGIC:WIDTH:SOURCE?  
-> :TRIGGER:LOGIC:WIDTH:SOURCE A0

解説 DL9510L、DL9505L は {A<x>|C<x>} が有効です。

**:TRIGger:LOGic:WIDTh:TIME<x>**

機能 ロジックパルス幅トリガのパルス幅を設定 / 問い合わせします。

構文 :TRIGger:LOGic:WIDTh:TIME<x> {<時間>}  
:TRIGger:LOGic:WIDTh:TIME<x>?  
<x> = 1、2

例 <時間> = 1ns ~ 10s(500ps ステップ)  
:TRIGGER:LOGIC:WIDTH:TIME1 1S  
:TRIGGER:LOGIC:WIDTH:TIME1?  
-> :TRIGGER:LOGIC:WIDTH:  
TIME1 1.000E+00

解説 TIME2 は、「:TRIGger:WIDTh:MODE  
BETween|NOTBetween」のときに有効です。

**:TRIGger:MODE**

機能 トリガモードを設定 / 問い合わせします。

構文 :TRIGger:MODE {ALEVel|AUTO|NORMal|  
NSingle|SINGLE}  
:TRIGger:MODE?

例 :TRIGGER:MODE ALEVEL  
:TRIGGER:MODE? -> :TRIGGER:  
MODE ALEVEL

**:TRIGger:POSition**

機能 トリガポジションを設定 / 問い合わせします。

構文 :TRIGger:POSition {<NRf>}  
:TRIGger:POSition?  
<NRf> = 0 ~ 100(%)

例 :TRIGGER:POSITION 10  
:TRIGGER:POSITION? -> :TRIGGER:  
POSITION 10

**:TRIGger:SCount (Single(N) Count)**

機能 トリガモードが Single(N) 時のトリガ成立回数を設定 / 問い合わせします。

構文 :TRIGger:SCount {<NRf>}  
:TRIGger:SCount?

<NRf> = 本体ユーザーズマニュアル参照。

例 :TRIGGER:SCOUNT 1  
:TRIGGER:SCOUNT? -> :TRIGGER:SCOUNT 1

**:TRIGger:SOURce?**

機能 トリガソースに関するすべての設定値を問い合わせます。

構文 :TRIGger:SOURce?

例 :TRIGGER:SOURCE? -> :TRIGGER:SOURCE:  
CHANNEL1:COUPLING DC;HFREJECTION OFF;  
HYSTERESIS HIGH;LEVEL 1.000E+00;  
STATE HIGH;WIDTH 1.000E+00;WINDOW 0;;  
TRIGGER:SOURCE:CHANNEL2:COUPLING DC;  
HFREJECTION OFF;HYSTERESIS HIGH;  
LEVEL 1.000E+00;STATE HIGH;  
WIDTH 1.000E+00;WINDOW 0;;TRIGGER:  
SOURCE:CHANNEL3:COUPLING DC;  
HFREJECTION OFF;HYSTERESIS HIGH;  
LEVEL 1.000E+00;STATE HIGH;  
WIDTH 1.000E+00;WINDOW 0;;TRIGGER:  
SOURCE:CHANNEL4:COUPLING DC;  
HFREJECTION OFF;HYSTERESIS HIGH;  
LEVEL 1.000E+00;STATE HIGH;  
WIDTH 1.000E+00;WINDOW 0;;TRIGGER:  
SOURCE:EXTERNAL:LEVEL 0.000E+00;  
PROBE 1;;TRIGGER:SOURCE:LOGIC AND

**:TRIGger:SOURce:CHANnel<x>?**

機能 トリガソースの各チャンネルに関するすべての設定値を問い合わせます。

構文 :TRIGger:SOURce:CHANnel<x>?  
<x> = 1 ~ 4

例 :TRIGGER:SOURCE:CHANNEL1?  
-> :TRIGGER:SOURCE:CHANNEL1:  
COUPLING DC;HFREJECTION OFF;  
HYSTERESIS HIGH;LEVEL 1.000E+00;  
STATE HIGH;WIDTH 1.000E+00;WINDOW 0



### 5.30 TRIGger グループ

**:TRIGger:SOURCE:CHANNEL<x>:COUPLing**  
 機能 各チャンネルのトリガカップリングを設定 / 問い合わせします。  
 構文 :TRIGger:SOURCE:CHANNEL<x>:  
 COUPLing {AC|DC}  
 :TRIGger:SOURCE:CHANNEL<x>:COUPLing?  
 <x> = 1 ~ 4  
 例 :TRIGGER:SOURCE:CHANNEL1:COUPLING AC  
 :TRIGGER:SOURCE:CHANNEL1:COUPLING?  
 -> :TRIGGER:SOURCE:CHANNEL1:  
 COUPLING DC

**:TRIGger:SOURCE:CHANNEL<x>:  
 HFRejection (HighFrequencyREJECTION)**  
 機能 各チャンネルのローパスフィルタ (HF リジエクシオン) を設定 / 問い合わせします。  
 構文 :TRIGger:SOURCE:CHANNEL<x>:  
 HFRejection {<周波数>|OFF}  
 :TRIGger:SOURCE:CHANNEL<x>:  
 HFRejection?  
 <x> = 1 ~ 4  
 <周波数> = 20MHz、15kHz  
 例 :TRIGGER:SOURCE:CHANNEL1:  
 HFREJECTION OFF  
 :TRIGGER:SOURCE:CHANNEL1:HFREJECTION?  
 -> :TRIGGER:SOURCE:CHANNEL1:  
 HFREJECTION OFF  
 解説 トリガソースが、{EXTERNAL|LINE} のときは無効です。

**:TRIGger:SOURCE:CHANNEL<x>:  
 HYSTeresis**  
 機能 各チャンネルのヒステリシスを設定 / 問い合わせします。  
 構文 :TRIGger:SOURCE:CHANNEL<x>:  
 HYSTeresis {HIGH|LOW}  
 :TRIGger:SOURCE:CHANNEL<x>:  
 HYSTeresis?  
 <x> = 1 ~ 4  
 例 :TRIGGER:SOURCE:CHANNEL1:  
 HYSTERESIS HIGH  
 :TRIGGER:SOURCE:CHANNEL1:HYSTERESIS?  
 -> :TRIGGER:SOURCE:CHANNEL1:  
 HYSTERESIS HIGH

**:TRIGger:SOURCE:CHANNEL<x>: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 :TRIGger:SOURCE:CHANNEL<x>:LEVel {<電圧>|<電流>}  
 :TRIGger:SOURCE:CHANNEL<x>:LEVel?  
 <x> = 1 ~ 4  
 <電圧>、<電流> = 本体ユーザーズマニュアル参照。  
 例 :TRIGGER:SOURCE:CHANNEL1:LEVEL 1V  
 :TRIGGER:SOURCE:CHANNEL1:LEVEL?  
 -> :TRIGGER:SOURCE:CHANNEL1:  
 LEVEL 1.000E+00

**:TRIGger:SOURCE:CHANNEL<x>:STATE**  
 機能 各チャンネルの成立条件を設定 / 問い合わせします。  
 構文 :TRIGger:SOURCE:CHANNEL<x>:  
 STATE {DONTcare|HIGH|IN|LOW|OUT}  
 :TRIGger:SOURCE:CHANNEL<x>:STATE?  
 <x> = 1 ~ 4  
 例 :TRIGGER:SOURCE:CHANNEL1:STATE HIGH  
 :TRIGGER:SOURCE:CHANNEL1:STATE?  
 -> :TRIGGER:SOURCE:CHANNEL1:  
 STATE HIGH  
 解説
 

- ・「:TRIGger:TYPE EQUalify|I2Cbus|PQUalify|PState|SPATtern|STATE」のときに有効です。
- ・「:TRIGger:TYPE I2Cbus|SPATtern」のときは {HIGH|LOW} が有効です。
- ・「:TRIGger:TYPE EQUalify|PQUalify|PState|STATE」且つ「:TRIGger:SOURCE:CHANNEL<x>:WINDOW ON」のときは {IN|OUT}、 「TRIGger:SOURCE:CHANNEL<x>:WINDOW OFF」のときは {HIGH|LOW} が有効です。

**:TRIGger:SOURCE:CHANNEL<x>:WIDTh**  
 機能 各チャンネルのウィンドウトリガの幅を設定 / 問い合わせします。  
 構文 :TRIGger:SOURCE:CHANNEL<x>:  
 WIDTh {<電圧>|<電流>}  
 :TRIGger:SOURCE:CHANNEL<x>:WIDTh?  
 <x> = 1 ~ 4  
 <電圧>、<電流> = 本体ユーザーズマニュアル参照。  
 例 :TRIGGER:SOURCE:CHANNEL1:WIDTH 1V  
 :TRIGGER:SOURCE:CHANNEL1:WIDTH?  
 -> :TRIGGER:SOURCE:CHANNEL1:  
 WIDTH 1.000E+00  
 解説 「:TRIGger:SOURCE:CHANNEL<x>:WINDOW ON」のときに有効です。

**:TRIGger:SOURCE:CHANnel<x>:WINDow**

機能 各チャンネルのウィンドウの ON/OFF を設定 / 問い合わせします。

構文 :TRIGger:SOURCE:CHANnel<x>:  
WINDow {<Boolean>}  
:TRIGger:SOURCE:CHANnel<x>:WINDow?  
<x> = 1 ~ 4

例 :TRIGGER:SOURCE:CHANNEL1:WINDOW ON  
:TRIGGER:SOURCE:CHANNEL1:WINDOW?  
-> :TRIGGER:SOURCE:CHANNEL1:WINDOW 1

**:TRIGger:SOURCE:EXTErnal?**

機能 外部トリガに関するすべての設定値を問い合わせます。

構文 :TRIGger:SOURCE:EXTErnal?  
例 :TRIGGER:SOURCE:EXTERNAL?  
-> :TRIGGER:SOURCE:EXTERNAL:  
LEVEL 0.000E+00;PROBE 1

**:TRIGger:SOURCE:EXTErnal:LEVEl**

機能 外部トリガのトリガレベルを設定 / 問い合わせします。

構文 :TRIGger:SOURCE:EXTErnal:  
LEVEl {<電圧>|<電流>}  
:TRIGger:SOURCE:EXTErnal:LEVEl?  
<x> = 1 ~ 4  
<電圧>、<電流> = 本体ユーザーズマニュアル参照。

例 :TRIGGER:SOURCE:EXTERNAL:LEVEL 1V  
:TRIGGER:SOURCE:EXTERNAL:LEVEL?  
-> :TRIGGER:SOURCE:EXTERNAL:  
LEVEL 1.000E+00

解説 「:TRIGger:TYPE EDGE|EQUalify|  
PQUalify|PULSe」のときに有効です。

**:TRIGger:SOURCE:EXTErnal:PROBe**

機能 外部トリガのプロブ減衰比を設定 / 問い合わせします。

構文 :TRIGger:SOURCE:EXTErnal:PROBe  
{<Nrf>}  
:TRIGger:SOURCE:EXTErnal:PROBe?  
<Nrf> = 1、10

例 :TRIGGER:SOURCE:EXTERNAL:PROBE 1  
:TRIGGER:SOURCE:EXTERNAL:PROBE?  
-> :TRIGGER:SOURCE:EXTERNAL:PROBE 1

解説 「:TRIGger:TYPE EDGE|EQUalify|  
PQUalify|PULSe」のときに有効です。

**:TRIGger:SOURCE:LOGic**

機能 トリガソースのロジックを設定 / 問い合わせします。

構文 :TRIGger:SOURCE:LOGic {AND|OR}  
:TRIGger:SOURCE:LOGic?

例 :TRIGGER:SOURCE:LOGIC AND  
:TRIGGER:SOURCE:LOGIC? -> :TRIGGER:  
SOURCE:LOGIC AND

解説 「:TRIGger:TYPE EQUalify|I2Cbus|  
PQUalify|PState|SPATtern|STATE」のときに有効です。

**:TRIGger:TYPE**

機能 トリガの種類を設定 / 問い合わせします。

構文 :TRIGger:TYPE {CANbus|EDGE|EICycle|  
EIDelay|EISequence|EOR|EQUalify|  
I2Cbus|LEDge|LINbus|LI2Cbus|  
LLINbus|LSPattern|LSPiBus|LPState|  
LPULse|LQUalify|LState|LUART|  
PQUalify|PState|PULse|SPATtern|  
SPIbus|STATE|TV|UART}  
:TRIGger:TYPE?

例 :TRIGGER:TYPE CANBUS  
:TRIGGER:TYPE? -> :TRIGGER:  
TYPE CANBUS

**:TRIGger:WIDTh?**

機能 パルス幅トリガのすべての設定値を問い合わせます。

構文 :TRIGger:WIDTh?  
例 :TRIGGER:WIDTH? -> :TRIGGER:WIDTH:  
MODE OUT;POLARITY POSITIVE;SOURCE 1;  
TIME1 1.000E-09;TIME2 1.000E-09

**:TRIGger:WIDTh:MODE**

機能 パルス幅トリガの判定モードを設定 / 問い合わせします。

構文 :TRIGger:WIDTh:MODE {BETween|IN|  
NOTBetween|OUT|TIMEout}  
:TRIGger:WIDTh:MODE?

例 :TRIGGER:WIDTH:MODE BETWEEN  
:TRIGGER:WIDTH:MODE? -> :TRIGGER:  
WIDTH:MODE BETWEEN

### 5.30 TRIGger グループ

#### **:TRIGger:WIDTh:POLarity**

- 機能 パルス幅トリガの極性を設定 / 問い合わせします。
- 構文 :TRIGger:WIDTh:POLarity {FALSE|IN|NEGative|OUT|POSitive|TRUE}  
:TRIGger:WIDTh:POLarity?
- 例 :TRIGGER:WIDTH:POLARITY POSITIVE  
:TRIGGER:WIDTH:POLARITY? -> :TRIGGER:WIDTH:POLARITY POSITIVE
- 解説
- ・「:TRIGger:TYPE PQUALify|PULSE」且つ「:TRIGger:SOURce:CHANnel<x>:WINDOW ON」のときは {IN|OUT}、 「:TRIGger:SOURce:CHANnel<x>:WINDOW OFF」のときは {POSitive|NEGative} が有効です
  - ・「:TRIGger:TYPE PSTATE」のときは {FALSE|TRUE} が有効です。

#### **:TRIGger:WIDTh:SOURce**

- 機能 パルス幅トリガのトリガソースを設定 / 問い合わせします。
- 構文 :TRIGger:WIDTh:SOURce {<NRf>|EXTernal}  
:TRIGger:WIDTh:SOURce?  
<NRf> = 1 ~ 4
- 例 :TRIGGER:WIDTH:SOURCE EXTERNAL  
:TRIGGER:WIDTH:SOURCE? -> :TRIGGER:WIDTH:SOURCE EXTERNAL
- 解説 「:TRIGger:TYPE PQUALify|PULSE」のときに有効です。

#### **:TRIGger:WIDTh:TIME<x>**

- 機能 パルス幅トリガのパルス幅を設定 / 問い合わせします。
- 構文 :TRIGger:WIDTh:TIME<x> {<時間>}  
:TRIGger:WIDTh:TIME<x>?  
<x> = 1、2  
<時間> = 1ns ~ 10s(500ps ステップ)
- 例 :TRIGGER:WIDTH:TIME1 1S  
:TRIGGER:WIDTH:TIME1? -> :TRIGGER:WIDTH:TIME1 1.000E+00
- 解説 TIME2 は、「:TRIGger:WIDTh:MODE BETWEEN|NOTBetween」のときに有効です。

## 5.31 WAVEform グループ

WAVEform グループは、取り込んだ波形データ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。

### :WAVEform?

機能 波形データのすべての情報を問い合わせます。  
構文 :WAVEform?  
例 :WAVEFORM? -> :WAVEFORM:TRACE 1;  
RECORD 0;START 0;END 6249999;  
FORMAT WORD;BYTEORDER LSBFIRST

### :WAVEform:BITS?

機能 「:WAVEform:TRACe」で指定した波形データのビット長を問い合わせます。  
構文 :WAVEform:BITS?  
例 :WAVEFORM:BITS? -> :WAVEFORM:BITS 16

### :WAVEform:BYTeorder

機能 2バイト以上のワードフォーマットのときの送信順序を設定/問い合わせします。  
構文 :WAVEform:BYTeorder  
{LSBFirst|MSBFirst}  
:WAVEform:BYTeorder?  
例 :WAVEFORM:BYTEORDER LSBFIRST  
:WAVEFORM:BYTEORDER? -> :WAVEFORM:  
BYTEORDER LSBFIRST

### :WAVEform:END

機能 「:WAVEform:TRACe」で指定した波形の、どの点を最後のデータとするかを設定/問い合わせします。  
構文 :WAVEform:END {<NRf>}  
:WAVEform:END?  
<NRf> = 0 ~ 6,249,999  
例 :WAVEFORM:END 12499  
:WAVEFORM:END? -> :WAVEFORM:END 12499  
解説 全データ点数は「:WAVEform:LENGth?」で問い合わせできます。

### :WAVEform:FORMAt

機能 送信するデータのフォーマットを設定/問い合わせします。  
構文 :WAVEform:FORMAt {ASCIi|BYTe|DWORD|  
RBYTe|WORD}  
:WAVEform:FORMAt?  
例 :WAVEFORM:FORMAT ASCII  
:WAVEFORM:FORMAT? ->  
:WAVEFORM:FORMAT ASCII  
解説  
・このフォーマットの設定による違いは、「:WAVEform:SEND?」の解説を参照してください。  
・「:WAVEform:TRACe LGROUP<x>」以外ときは、{DWORD}は無効です。  
・「:WAVEform:TRACe LGROUP<x>」のときは、{RBYTe}は無効です。

### :WAVEform:LENGth?

機能 「:WAVEform:TRACe」で指定した波形の全データ点数を問い合わせます。  
構文 :WAVEform:LENGth?  
例 :WAVEFORM:LENGth? -> :WAVEFORM:  
LENGth 12500

### :WAVEform:OFFSet?

機能 「:WAVEform:TRACe」で指定した波形データを物理値に変換するときのオフセット値を問い合わせます。  
構文 :WAVEform:OFFSet?  
例 :WAVEFORM:OFFSet? -> 0.000E+00  
解説  
・このオフセット値は、「:WAVEform:SEND?」で出力される<ブロックデータ>を物理値に換算するときに使用します。  
・「:CHANnel<x>:OCANcel」が「ON」のときは0が返されます。  
・「:WAVEform:TRACe LGROUP<x>」のときは0が返されます。

### :WAVEform:POSition?

機能 「:WAVEform:FORMAt」で「RBYTe」を指定した場合の、電圧に換算するとき使用する垂直軸ポジションを問い合わせます。  
構文 :WAVEform:POSition?  
例 :WAVEFORM:POSITION? -> :WAVEFORM:  
POSITION 128

### :WAVEform:RANGe?

機能 「:WAVEform:TRACe」で指定した波形データを物理値に換算するときのレンジ値を問い合わせます。  
構文 :WAVEform:RANGe?  
例 :WAVEFORM:RANGe? -> 5.000E+00  
解説 このレンジ値は、「:WAVEform:SEND?」で出力される<ブロックデータ>を物理値に換算するときに使用します。

## 5.31 WAVEform グループ

### :WAVEform:RECORD

機能	WAVEform グループで対象となるレコード番号を設定 / 問い合わせします。
構文	:WAVEform:RECORD {AVERAge MINimum <NRf>} :WAVEform:RECORD? <NRf> = 0 ~ 1999
例	:WAVEFORM:RECORD 0 :WAVEFORM:RECORD? -> :WAVEFORM:RECORD 0
解説	「AVERAge」を指定すると、ヒストリ - アベレージ波形が対象になります。 「MINimum」を指定すると、最小のレコード番号になります。設定できるレコード番号は、モデルやアクイジションの設定によって異なります。詳しくは本体ユーザーズマニュアルをご覧ください。

### :WAVEform:RECORD? MINimum

機能	対象チャンネルのヒストリの最小レコード番号を問い合わせします。
構文	:WAVEform:RECORD? MINimum
例	:WAVEFORM:RECORD? MINimum -> :WAVEFORM:RECORD -1999

### :WAVEform:SEND?

機能	「:WAVEform:TRACe」で指定した波形データを問い合わせます。
構文	:WAVEform:SEND? [{<NRf>}] <NRf> = 1 ~ 2000
例	レコード長の設定により異なります。 :WAVEFORM:SEND? -> #8(8 桁のバイト数)(データ列) または、<NRf>,<NRf>,...
解説	• 「:WAVEform:FORMat」の設定によって、「:WAVEform:SEND?」の出力形式が変わります。 (1) 「AScii」にしたとき • 「:WAVEform:TRACe」が、ロジックグループ以外の場合は、<電圧>、<電圧>、...<電圧>の形式で返されます。 • 「:WAVEform:TRACe」がロジックグループの場合は、<NR1>、<NR1>、...、<NR1>の形式で返されます。<NR1>は、ロジックビットパターンを10進数にしたものです。 (2) 「BYTe」、「WORD」、「DWORD」にしたとき <ブロックデータ>の形式で返されます。 次の式で、換算できます。 電圧(演算値) = (レンジ×データ÷ Division*) + オフセット * 「BYTe」: Division = 12.5(ロジックグループのときは1) 「WORD」: Division = 3200(ロジックグループのときは1) 「DWORD」: Division = 1(ロジックグループのときだけ) 「:LOGic:GRoup<x>:MAPPING」で設定されたビット配置数が多い場合は、各FORMatに応じて下位ビットが出力されます。 (3) 「RBYTe」にしたとき <ブロックデータ>の形式で返されます。 次の式で、換算できます。 電圧(演算値) = (レンジ×(データ - Position) ÷ Division*) + オフセット * Division = 25 Position = 「:WAVEform:POSition?」の返値 • <NRf> は省略可能です。<NRf> を付けると、「:WAVEform:RECORD」で設定したレコード番号 - <NRf> + 1 から <NRf> 回、波形データを順に問い合わせます。

### :WAVEform:SIGN?

機能	「:WAVEform:TRACe」で指定した対象波形をバイナリデータで問い合わせる場合の、符号の有無を問い合わせます。
構文	:WAVEform:SIGN?
例	:WAVEFORM:SIGN? -> :WAVEFORM:SIGN 1
解説	「:WAVEform:TRACe LGROUP<x>」のときは、0が返されます。

**:WAVEform:SRATE? (Sample RATE)**

機能 「:WAVEform:RECOrd」で指定したレコードのサンプルレートを問い合わせます。

構文 :WAVEform:SRATE?

例 :WAVEFORM:SRATE? -> :WAVEFORM:  
SRATE 1.25E+09

**:WAVEform:START**

機能 「:WAVEform:TRACe」で指定した波形の、どの点を最初のデータとするかを設定 / 問い合わせします。

構文 :WAVEform:START {<NRf>}

:WAVEform:START?

<NRf> = 0 ~ 6,249,999(2.5MW モデルでは 0 ~ 2,499,999)

例 :WAVEFORM:START 0  
:WAVEFORM:START? -> :WAVEFORM:START 0

**:WAVEform:TRACe**

機能 WAVEform グループで対象となる波形を設定 / 問い合わせします。

構文 :WAVEform:TRACe {<NRf>|LGRoup<x>|

MATH<x>|REFeRence<x>}

:WAVEform:TRACe?

<NRf> = 1 ~ 4

LGRoup<x> の <x> = 1 ~ 5

MATH<x> の <x> = 1 ~ 8

REFeRence<x> の <x> = 1 ~ 4

例 :WAVEFORM:TRACE 1  
:WAVEFORM:TRACE? -> :WAVEFORM:TRACE 1

**:WAVEform:TRIGger?**

機能 「:WAVEform:RECOrd」で指定したレコードのトリガポジションを問い合わせます。

構文 :WAVEform:TRIGger?

例 :WAVEFORM:TRIGGER? -> :WAVEFORM:  
TRIGGER 6250

解説 レコード長の先頭からトリガポジションまでの点数を問い合わせます。

**:WAVEform:TYPE?**

機能 「:WAVEform:TRACe」で指定した波形のアクイジションモードを問い合わせます。

構文 :WAVEform:TYPE?

例 :WAVEFORM:TYPE? -> :WAVEFORM:  
TYPE NORMAL

## 5.32 ZOOM グループ

### : ZOOM?

機能 波形のズームに関するすべての設定値を問い合わせます。

構文 : ZOOM?

例 : ZOOM? -> : ZOOM: ALLOCATION1: TRACE1 1; TRACE2 1; TRACE3 1; TRACE4 1; TRACE5 1; TRACE6 1; TRACE7 1; TRACE8 1; : ZOOM: ALLOCATION2: TRACE1 1; TRACE2 1; TRACE3 1; TRACE4 1; TRACE5 1; TRACE6 1; TRACE7 1; TRACE8 1; : ZOOM: FORMAT1 MAIN; FORMAT2 MAIN; HLINKAGE 0; HORIZONTAL1: ASCROLL: SPEED 5; : ZOOM: HORIZONTAL1: MAG 2.000E+00; POSITION 0.000E+00; : ZOOM: HORIZONTAL2: ASCROLL: SPEED 5; : ZOOM: HORIZONTAL2: MAG 2.000E+00; POSITION 0.000E+00; : ZOOM: MODE MAIN; TYPE1 HORIZONTAL; TYPE2 HORIZONTAL; VERTICAL1: MAG 1.000E+00; POSITION 0.000E+00; TRACE 1; : ZOOM: VERTICAL2: MAG 1.000E+00; POSITION 0.000E+00; TRACE 1; : ZOOM: VLINKAGE 0

### : ZOOM: ALLOCATION<x>?

機能 ズーム対象波形に関するすべての設定値を問い合わせます。

構文 : ZOOM: ALLOCATION<x>?

<x> = 1, 2

例 : ZOOM: ALLOCATION1? -> : ZOOM: ALLOCATION1: TRACE1 1; TRACE2 1; TRACE3 1; TRACE4 1; TRACE5 1; TRACE6 1; TRACE7 1; TRACE8 1

### : ZOOM: ALLOCATION: ALLON

機能 全波形をズーム対象にします。

構文 : ZOOM: ALLOCATION<x>: ALLON

<x> = 1, 2

例 : ZOOM: ALLOCATION1: ALLON

### : ZOOM: ALLOCATION: TRACe<x>

機能 ズーム対象にしたいトレースの ON/OFF を設定 / 問い合わせします。

構文 : ZOOM: ALLOCATION<x>:

TRACe<x> {<Boolean>}

: ZOOM: ALLOCATION<x>: TRACe<x>?

ALLOCATION<x> の <x> = 1, 2

TRACe<x> の <x> = 1 ~ 8

例 : ZOOM: ALLOCATION1: TRACE1 ON  
: ZOOM: ALLOCATION1: TRACE1? -> : ZOOM: ALLOCATION1: TRACE1 1

### : ZOOM: FORMat<x>

機能 ズーム波形の表示フォーマットを設定 / 問い合わせします。

構文 : ZOOM: FORMat<x>

{DUAL|MAIN|QUAD|SINGLE|

TRIad}

: ZOOM: FORMat<x>?

<x> = 1, 2

例 : ZOOM: FORMAT1 SINGLE  
: ZOOM: FORMAT1? -> : ZOOM: FORMAT1 SINGLE

### : ZOOM: HLINKage

機能 水平軸方向リンクの ON/OFF を設定 / 問い合わせします。

構文 : ZOOM: HLINKage {<Boolean>}

: ZOOM: HLINKage?

例 : ZOOM: HLINKAGE ON  
: ZOOM: HLINKAGE? -> : ZOOM: HLINKAGE 1

### : ZOOM: HORizontal<x>?

機能 水平軸方向のズームに関するすべての設定値を問い合わせます。

構文 : ZOOM: HORizontal<x>?

<x> = 1, 2

例 : ZOOM: HORIZONTAL1? -> : ZOOM: HORIZONTAL1: ASCROLL: SPEED 5; : ZOOM: HORIZONTAL1: MAG 2.000E+00; POSITION 4.000E+00

### : ZOOM: HORizontal<x>: ASCRoll?

機能 オートスクロール機能に関するすべての設定値を問い合わせます。

構文 : ZOOM: HORizontal<x>: ASCRoll?

<x> = 1, 2

例 : ZOOM: HORIZONTAL1: ASCROLL? -> : ZOOM: HORIZONTAL1: ASCROLL: SPEED 5

### : ZOOM: HORizontal<x>: ASCRoll: JUMP

機能 ズームの中心位置をメイン画面の左端、右端に移動します。

構文 : ZOOM: HORizontal<x>: ASCRoll:

JUMP {LEFT|RIGHT}

<x> = 1, 2

例 : ZOOM: HORIZONTAL1: ASCROLL: JUMP RIGHT

**:ZOOM:HORizontal<x>:ASCROLL:SPEEd**  
 機能 オートスクロールのスピードを設定 / 問い合わせ  
 します。  
 構文 :ZOOM:HORizontal<x>:ASCROLL:SPEEd  
 {<NRf>}  
 :ZOOM:HORizontal<x>:ASCROLL:SPEEd?  
 <x> = 1, 2  
 <NRf> = 1, 2, 5, 10, 20, 50  
 例 :ZOOM:HORIZONTAL1:ASCROLL:SPEED 1  
 :ZOOM:HORIZONTAL1:ASCROLL:SPEED?  
 -> :ZOOM:HORIZONTAL1:ASCROLL:SPEED 1

**:ZOOM:HORizontal<x>:ASCROLL:START**  
 機能 オートスクロールを開始します。  
 構文 :ZOOM:HORizontal<x>:ASCROLL:START  
 {LEFT|RIGHT}  
 <x> = 1, 2  
 例 :ZOOM:HORIZONTAL1:ASCROLL:START LEFT

**:ZOOM:HORizontal<x>:ASCROLL:STOP**  
 機能 オートスクロールを停止します。  
 構文 :ZOOM:HORizontal<x>:ASCROLL:STOP  
 <x> = 1, 2  
 例 :ZOOM:HORIZONTAL1:ASCROLL:STOP

**:ZOOM:HORizontal<x>:MAG**  
 機能 水平軸方向のズーム倍率を設定 / 問い合わせしま  
 します。  
 構文 :ZOOM:HORizontal<x>:MAG {<NRf>}  
 :ZOOM:HORizontal<x>:MAG?  
 <x> = 1, 2  
 <NRf> = 本体ユーザーズマニュアル参照。  
 例 :ZOOM:HORIZONTAL1:MAG 2  
 :ZOOM:HORIZONTAL1:MAG? -> :ZOOM:  
 HORIZONTAL1:MAG 2.000E+00

**:ZOOM:HORizontal<x>:POSITION**  
 機能 水平軸方向のズームの中心位置を設定 / 問い合わ  
 せします。  
 構文 :ZOOM:HORizontal<x>:POSITION {<NRf>}  
 :ZOOM:HORizontal<x>:POSITION?  
 <x> = 1, 2  
 <NRf> = - 5 ~ 5(div)  
 例 :ZOOM:HORIZONTAL1:POSITION 1  
 :ZOOM:HORIZONTAL1:POSITION? -> :ZOOM:  
 HORIZONTAL1:POSITION 1.000E+00

**:ZOOM:MODE**  
 機能 ズーム波形の表示方法を設定 / 問い合わせしま  
 します。  
 構文 :ZOOM:MODE {MAIN|MAIN\_Z1|MAIN\_Z1\_Z2|  
 MAIN\_Z2|Z1|Z1\_Z2|Z2}  
 :ZOOM:MODE?  
 例 :ZOOM:MODE MAIN\_Z1\_Z2  
 :ZOOM:MODE? -> :ZOOM:MODE MAIN\_Z1\_Z2

**:ZOOM:TYPE<x>**  
 機能 ズームタイプを設定 / 問い合わせします。  
 構文 :ZOOM:TYPE<x> {HORizontal|VERTical}  
 :ZOOM:TYPE<x>?  
 <x> = 1, 2  
 例 :ZOOM:TYPE1 VERTICAL  
 :ZOOM:TYPE1? -> :ZOOM:TYPE1 VERTICAL

**:ZOOM:VERTical<x>?**  
 機能 垂直軸方向のズームに関するすべての設定値を  
 問い合わせます。  
 構文 :ZOOM:VERTical<x>?  
 <x> = 1, 2  
 例 :ZOOM:VERTICAL1? -> :ZOOM:VERTICAL1:  
 MAG 1.000E+00;POSITION 0.000E+00;  
 TRACE 1

**:ZOOM:VERTical<x>:INITialize**  
 機能 垂直軸方向のズームの初期化をします。  
 構文 :ZOOM:VERTical<x>:INITialize  
 <x> = 1, 2  
 例 :ZOOM:VERTICAL1:INITIALIZE

**:ZOOM:VERTical<x>:MAG**  
 機能 垂直軸方向のズーム倍率を設定 / 問い合わせしま  
 します。  
 構文 :ZOOM:VERTical<x>:MAG {<NRf>}  
 :ZOOM:VERTical<x>:MAG?  
 <x> = 1, 2  
 <NRf> = 本体ユーザーズマニュアル参照。  
 例 :ZOOM:VERTICAL1:MAG 1  
 :ZOOM:VERTICAL1:MAG? -> :ZOOM:  
 VERTICAL1:MAG 1.000E+00

**:ZOOM:VERTical<x>:POSITION**  
 機能 垂直軸方向のズームポジションを設定 / 問い合わ  
 せします。  
 構文 :ZOOM:VERTical<x>:POSITION {<NRf>}  
 :ZOOM:VERTical<x>:POSITION?  
 <x> = 1, 2  
 <NRf> = - 4 ~ 4(div)  
 例 :ZOOM:VERTICAL1:POSITION 1  
 :ZOOM:VERTICAL1:POSITION? -> :ZOOM:  
 VERTICAL1:POSITION 1.000E+00

**:ZOOM:VERTical<x>:TRACE**  
 機能 垂直軸方向のズーム画面に表示したいトレース  
 を設定 / 問い合わせします。  
 構文 :ZOOM:VERTical<x>:TRACE {<NRf>}  
 :ZOOM:VERTical<x>:TRACE?  
 <x> = 1, 2  
 <NRf> = 1 ~ 8  
 例 :ZOOM:VERTICAL1:TRACE 1  
 :ZOOM:VERTICAL1:TRACE? -> :ZOOM:  
 VERTICAL1:TRACE 1



## 5.32 ZOOM グループ

---

### **:ZOOM:VLINKage**

機能 垂直軸方向リンクの ON/OFF を設定 / 問い合わせ  
します。

構文 :ZOOM:VLINKage {<Boolean>}  
:ZOOM:VLINKage?

例 :ZOOM:VLINKAGE ON  
:ZOOM:VLINKAGE? -> :ZOOM:VLINKAGE 1

## 5.33 共通コマンドグループ

共通コマンドグループは、USBTMC-USB488 で規定されている、機器固有の機能に依存しないコマンドのグループです。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。

### \*CAL? (CALibrate)

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「1」が返されます。

### \*CLS (CLear Status)

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

- ・\*CLS コマンドがプログラムメッセージターミナーターのすぐ後ろにあるときは、出力キューもクリアされます。
- ・各レジスタ、キューについては、第6章を参照してください。

### \*ESE (standard Event Status Enable register)

機能 標準イベントイネーブルレジスタの値を設定 / 問い合わせします。

構文 \*ESE {<Nrf>}

\*ESE?

<Nrf> = 0 ~ 255

例 \*ESE 251

\*ESE? -> 251

解説

- ・各ビットの10進数の和で設定します。
- ・たとえば、「\*ESE 251」とすると、標準イベントイネーブルレジスタを「11111011」にセットします。つまり、標準イベントレジスタのビット2を無効にし、「問い合わせエラー」が起これどもステータスバイトレジスタのビット5(ESB)を「1」にしません。
- ・初期値は「\*ESE 0」(全ビット無効)です。
- ・\*ESE? で問い合わせても、標準イベントイネーブルレジスタの内容はクリアされません。
- ・標準イベントイネーブルレジスタについては、6-4 ページを参照してください。

### \*ESR? (standard Event Status Register)

機能 標準イベントレジスタの値を問い合わせ、同時にクリアします。

構文 \*ESR?

例 \*ESR? -> 32

解説

- ・各ビットの10進数の和が返されます。
- ・SRQ が発生しているときに、どんな種類のイベントが起こったかを調べることができます。
- ・たとえば、「32」が返されると、標準イベントレジスタが「00100000」にセットされていることを示します。つまり、「コマンド文法エラー」が起こったためにSRQが発生したことがわかります。
- ・\*ESR? で問い合わせると、標準イベントレジスタの内容がクリアされます。
- ・標準イベントレジスタについては、6-4 ページを参照してください。

### \*IDN? (IDeNtify)

機能 機種を問い合わせます。

構文 \*IDN?

例 \*IDN? -> YOKOGAWA, 701331, 27E100000, F1.10

解説 <製造者>、<機種>、<シリアル No.>、<ファームウェアバージョン>の形式で返されます。<機種>は、DL9505L のとき 701320、DL9510L のとき 701321、DL9705L のとき 701330、DL9710L のとき 701331 が返されます。

### 5.33 共通コマンドグループ

#### \*LRN? (Learn)

機能 以下のグループの現在の設定を一度に問い合わせます。

ACQUIRE、CHANNEL<x>、TIMEBASE、TRIGGER

構文 \*LRN?

例 \*LRN? -> :ACQUIRE:AVERAGE:COUNT 2;  
 EWEIGHT 16;:ACQUIRE:HRMODE 0;  
 INTERLEAVE 0;INTERPOLATE 1;  
 MODE NORMAL;REPETITIVE 0;  
 RLENGTH 12500;:CHANNEL1:SELECT INPUT;  
 DISPLAY 1;BWIDTH FULL;COUPLING DC;  
 DESKEW 0.000E+00;INVERT 0;LABEL:  
 DEFINE "CH1";MODE 1;:CHANNEL1:  
 OCANCEL 0;OFFSET 0.000E+00;  
 POSITION 0.000E+00;PROBE:MODE 1;:  
 CHANNEL1:SVALUE 0;VDIV 1.000E+00;:  
 CHANNEL2:SELECT INPUT;DISPLAY 1;  
 BWIDTH FULL;COUPLING DC;  
 DESKEW 0.000E+00;INVERT 0;LABEL:  
 DEFINE "CH2";MODE 1;:CHANNEL2:  
 OCANCEL 0;OFFSET 0.000E+00;  
 POSITION 0.000E+00;PROBE:MODE 1;:  
 CHANNEL2:SVALUE 0;VDIV 1.000E+00;:  
 CHANNEL3:SELECT INPUT;DISPLAY 1;  
 BWIDTH FULL;COUPLING DC;  
 DESKEW 0.000E+00;INVERT 0;LABEL:  
 DEFINE "CH3";MODE 1;:CHANNEL3:  
 OCANCEL 0;OFFSET 0.000E+00;  
 POSITION 0.000E+00;PROBE:MODE 1;:  
 CHANNEL3:SVALUE 0;VDIV 1.000E+00;:  
 CHANNEL4:SELECT INPUT;DISPLAY 1;  
 BWIDTH FULL;COUPLING DC;  
 DESKEW 0.000E+00;INVERT 0;LABEL:  
 DEFINE "CH4";MODE 1;:CHANNEL4:  
 OCANCEL 0;OFFSET 0.000E+00;  
 POSITION 0.000E+00;PROBE:MODE 1;:  
 CHANNEL4:SVALUE 0;VDIV 1.000E+00;:  
 TIMEBASE:TDIV 1.000E-06;:TRIGGER:  
 ACTION:ACQCOUNT 1;BUZZER 0;HCOPY 0;  
 MODE OFF;SAVE 0;:TRIGGER:TYPE EDGE;  
 CLOCK:SOURCE 1;POLARITY RISE;:  
 TRIGGER:DELAY:EDGE:COUNT 1;:  
 TRIGGER:DELAY:MODE 0;POLARITY RISE;  
 SOURCE 1;TIME 0.000E+00;TYPE BYTIME;:  
 TRIGGER:EINTERVAL:EVENT1:TYPE EDGE;  
 CLOCK:SOURCE 1;POLARITY RISE;:  
 TRIGGER:EINTERVAL:EVENT1:ESTATE:  
 SOURCE 1;POLARITY RISE;:TRIGGER:  
 EINTERVAL:EVENT1:STATE:  
 CHANNEL1 DONTCARE;CHANNEL2 DONTCARE;  
 CHANNEL3 DONTCARE;CHANNEL4 DONTCARE;  
 LOGIC AND;:TRIGGER:EINTERVAL:EVENT1:  
 WIDTH:MODE OUT;POLARITY POSITIVE;  
 SOURCE 1;TIME1 1.000E-09;  
 TIME2 2.000E-09;:TRIGGER:EINTERVAL:  
 EVENT2:TYPE EDGE;CLOCK:SOURCE 1;  
 POLARITY RISE;:TRIGGER:EINTERVAL:

```

EVENT2:ESTATE:SOURCE 1;
POLARITY RISE;:TRIGGER:EINTERVAL:
EVENT2:STATE:CHANNEL1 DONTCARE;
CHANNEL2 DONTCARE;CHANNEL3 DONTCARE;
CHANNEL4 DONTCARE;LOGIC AND;:TRIGGER:
EINTERVAL:EVENT2:WIDTH:MODE OUT;
POLARITY POSITIVE;SOURCE 1;
TIME1 1.000E-09;TIME2 2.000E-09;:
TRIGGER:EINTERVAL:MODE OUT;
TIME1 1.500E-09;TIME2 2.000E-09;TRY:
MODE 0;SELECT 1;:TRIGGER:ENHANCED:TV:
CUSTOMIZE 0;FIELD 1;FRAME 1;HDTV:
LINE 2;POLARITY POSITIVE;:TRIGGER:
ENHANCED:TV:LEVEL 500.0E-03;NTSC:
LINE 5;POLARITY NEGATIVE;:TRIGGER:
ENHANCED:TV:PAL:LINE 2;
POLARITY NEGATIVE;:TRIGGER:ENHANCED:
TV:SGUARD 75;SOURCE 1;TYPE NTSC;
USERDEFINE:DEFINITION HD;
HFREJECTION OFF;HSYNC 31.500E+03;
LINE 2;POLARITY POSITIVE;:TRIGGER:
ESTATE:EOR:CHANNEL1 RISE;
CHANNEL2 RISE;CHANNEL3 RISE;
CHANNEL4 RISE;:TRIGGER:ESTATE:
SOURCE 1;POLARITY RISE;:TRIGGER:
HOLDOFF 20.00E-09;MODE AUTO;
POSITION 50;SCOUNT 1;SOURCE:CHANNEL1:
COUPLING DC;HFREJECTION OFF;
HYSTERESIS LOW;LEVEL 0.000E+00;
STATE DONTCARE;WIDTH 1.000E+00;
WINDOW 0;:TRIGGER:SOURCE:CHANNEL2:
COUPLING DC;HFREJECTION OFF;
HYSTERESIS LOW;LEVEL 0.000E+00;
STATE DONTCARE;WIDTH 1.000E+00;
WINDOW 0;:TRIGGER:SOURCE:CHANNEL3:
COUPLING DC;HFREJECTION OFF;
HYSTERESIS LOW;LEVEL 0.000E+00;
STATE DONTCARE;WIDTH 1.000E+00;
WINDOW 0;:TRIGGER:SOURCE:CHANNEL4:
COUPLING DC;HFREJECTION OFF;
HYSTERESIS LOW;LEVEL 0.000E+00;
STATE DONTCARE;WIDTH 1.000E+00;
WINDOW 0;:TRIGGER:SOURCE:EXTERNAL:
LEVEL 0.000E+00;PROBE 1;:TRIGGER:
SOURCE:LOGIC AND;:TRIGGER:WIDTH:
MODE OUT;POLARITY POSITIVE;SOURCE 1;
TIME1 1.000E-09;TIME2 2.000E-09

```

**\*OPC (Operation Complete)**

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0(OPCビット)を1にセットします。

構文 例 \*OPC  
\*OPC

解説

- ・\*OPCを使った同期のとり方については、4-7ページを参照してください。
- ・オーバーラップコマンドの指定は、「:COMMunicate:OPSE」で行います。
- ・メッセージの最後でない\*OPCの動作は保証されません。

**\*OPC? (Operation Complete)**

機能 \*OPC?を送信すると、指定したオーバーラップコマンドが終了していれば、ASCIIコードの「1」を返します。

構文 例 \*OPC?  
\*OPC? -> 1

解説

- ・\*OPC?を使った同期のとり方については、4-8ページを参照してください。
- ・オーバーラップコマンドの指定は「:COMMunicate:OPSE」で行います。
- ・メッセージの最後でない\*OPC?の動作は保証されません。

**\*OPT? (OPTion)**

機能 装備しているオプションを問い合わせます。

構文 例 \*OPT? -> CH6.25MW, PRINTER, ETHER, HDD, USERDEFINE, I2C, CAN, LIN, SPI, UART, PANALYZE, PROBEPOWER4, LXIHDDETHER, LXIETHER

解説

- ・<メモリモデル>、<内蔵プリンタ>、<イーサネット>、<内蔵ハードディスク>、<ユーザー定義演算>、<I2C解析機能>、<CAN解析機能>、<LIN解析機能>、<SPI解析機能>、<UART解析機能>、<電源解析機能>、<背面パネルプローブパワー>、<内蔵ハードディスク+LXI対応イーサネット>、<LXI対応イーサネット>の有無が返されます。
- ・「\*OPT?」はプログラムメッセージの最後のクエリ(問い合わせ)でなければなりません。後ろにほかのクエリがあるときは、エラーになります。

**\*PSC (Power-on Status Clear)**

機能 電源ON時に以下のレジスタをクリアするかしないかを設定/問い合わせします。整数に丸めた値が「0以外」のときにクリアされます。

- ・標準イベントイネーブルレジスタ
- ・拡張イベントイネーブルレジスタ
- ・遷移フィルタ

構文 \*PSC {<Nrf>}  
\*PSC?

<Nrf> = 0(クリアしない)、0以外(クリアする)

例 \*PSC 1  
\*PSC? -> 1

解説 各レジスタについては第6章を参照してください。

**\*RST (ReSet)**

機能 設定の初期化(イニシャライズ)をします。

構文 例 \*RST  
\*RST

解説 以前に送った\*OPCおよび\*OPC?も取り消します。

**\*SRE (Service Request Enable register)**

機能 サービスリクエストイネーブルレジスタの値を設定/問い合わせします。

構文 \*SRE <Nrf>  
\*SRE?  
<Nrf> = 0 ~ 255

例 \*SRE 239  
\*SRE? -> 239

解説

- ・各ビットの10進数の和で設定します。
- ・たとえば、「\*SRE 239」とすると、サービスリクエストイネーブルレジスタを「11101111」にセットします。つまり、サービスリクエストイネーブルレジスタのビット4を無効にし、「出力キューが空でない」ときでもステータスバイトレジスタのビット4(MAV)を「1」にしません。
- ・ただし、ステータスバイトレジスタのビット6(MSS)はMSSビット自身なので、無視されません。
- ・初期値は「\*SRE 0」(全ビット無効)です。
- ・\*SRE?で問い合わせても、サービスリクエストイネーブルレジスタの内容はクリアされません。
- ・サービスリクエストイネーブルレジスタについては、6-2ページを参照してください。

**\*STB? (STatus Byte)**

機能 ステータスバイトレジスタの値を問い合わせます。

構文 例 \*STB?  
\*STB? -> 4

解説

- ・各ビットの和が10進数で返されます。
- ・シリアルポールを実行せずにレジスタを読むので、ビット6はRQSではなくMSSビットになります。
- ・たとえば、「4」が返されると、ステータスバイトレジスタが「00000100」にセットされていることを示します。つまり、「エラーキューが空でない」(エラーが発生した)ことがわかります。
- ・\*STB?で問い合わせても、ステータスバイトレジスタの内容はクリアされません。
- ・ステータスバイトレジスタについては、6-3ページを参照してください。

### 5.33 共通コマンドグループ

---

#### **\*TST?**

機能 セルフテストを実行し、結果を問い合わせます。  
セルフテストの内容は、内部のメインメモリテストです。

構文 \*TST?

例 \*TST? -> 0

解説 セルフテストの結果が正常なときは「0」、異常があるときは「1」が返されます。

#### **\*WAI (WAIT)**

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。

構文 \*WAI

例 \*WAI

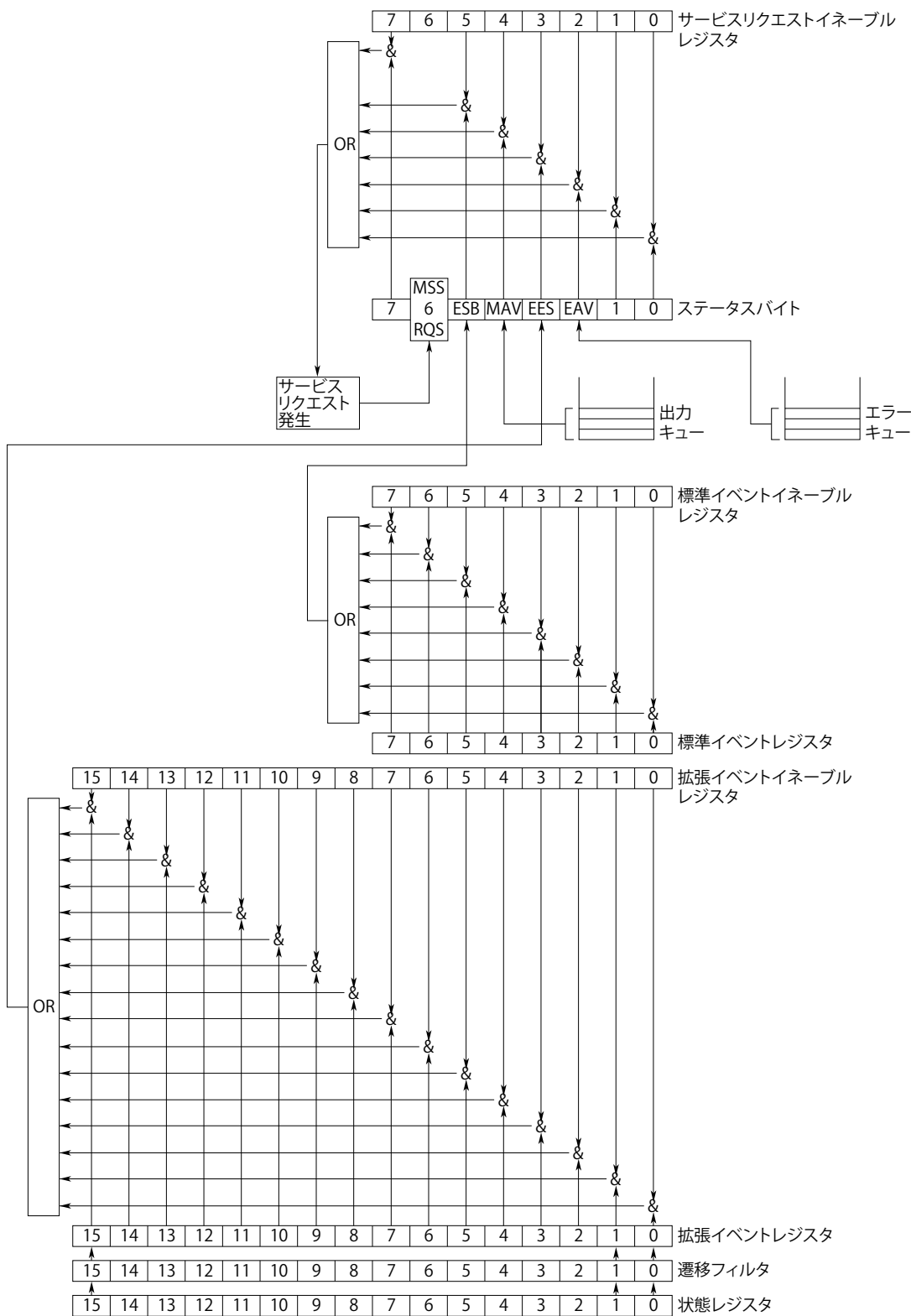
解説

- \*WAI を使った同期のとり方については、4-7 ページを参照してください。
- オーバーラップコマンドの指定は、「:COMMunicate:OPSE」で行います。

# 6.1 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。



## 6.1 ステータスレポートについて

### 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:EEER?
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 5 章で詳しく説明しています。

## 6.2 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。6-6 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、機器の内部であるイベントが起こったときに 1 になります。6-5 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べきデータがあるときに 1 になります。6-6 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、機器の内部であるイベントが起こったときに 1 になります。6-4 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、シリアルポールか、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 5 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積が 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。



## 6.3 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、スタート中にハードコピーを取ろうとした
- **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 5 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)の要因にしたくないときには、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 5 章をお読みください。

### 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。

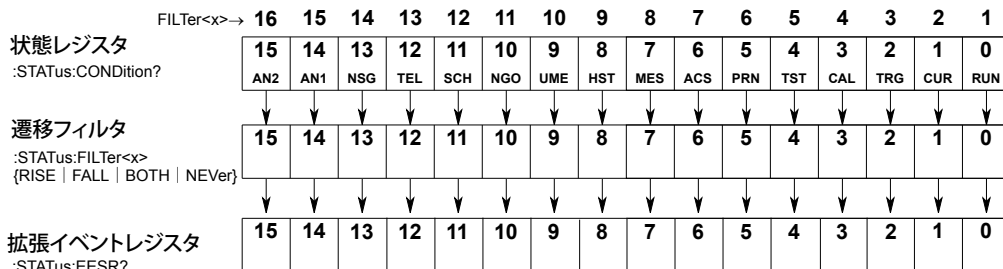
### 標準イベントレジスタのクリア

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

## 6.4 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、次の通りです。

ビット 0	RUN(Running)	波形の取り込み中は 1 になります。
ビット 1	CUR(Cursor)	カーソル測定中に 1 になります。
ビット 2	TRG(Awaiting trigger)	トリガ待ちのときに 1 になります。
ビット 3	CAL(Calibration)	キャリブレーション実行中に 1 になります。
ビット 4	TST(Testing)	セルフテスト中に 1 になります。
ビット 5	PRN(Printing)	内蔵プリンタの動作中、外部 (USB/ ネットワーク) プリンタにデータ出力中、および画面イメージデータを保存中に 1 になります。
ビット 6	ACS(Accessing)	各ドライブへのアクセス中に 1 になります。
ビット 7	MES(Measuring)	波形パラメータの自動測定中に 1 になります。
ビット 8	HST(History Search)	ヒストリ検索実行中に 1 になります。
ビット 9	UME(User Math Executing)	ユーザー定義演算中に 1 になります。
ビット 10	NGO(Go/No-go)	GO/NO-GO 検索実行中に 1 になります。
ビット 11	SCH(Search)	サーチ検索実行中に 1 になります。
ビット 12	TEL(Telecom Test)	Telecom Test 実行中に 1 になります。
ビット 13	NSG(N-Single)	トリガモードがシングル (N) のときの連続アキュイジション中に 1 になります。
ビット 14	AN1(Analysis1)	Analysis1 実行中に 1 になります。
ビット 15	AN2(Analysis2)	Analysis2 実行中に 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。

## 6.5 出力キューとエラーキュー

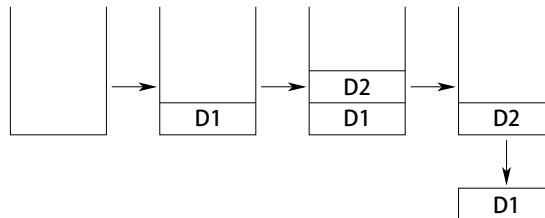
### 出力キュー

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、取り込んだ波形データの出力を要求する:WAVEform: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。

下図のように、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(4-2 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) で確認します。



### 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。

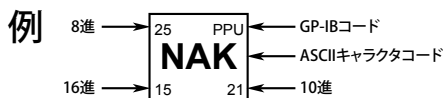
- ・ \*CLS コマンドを受信したとき
- ・ 電源の再投入

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。

# 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		トーカー 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。

- 本機器本体には日本語 / 英語のどちらでも表示することができますが、パーソナルコンピュータなどで読み出したときは英語で表示されます。
  - サービスが必要なときは、お買い求め先まで修理をお申しつけください。
  - ここに記載しているエラーメッセージは、通信に関するものに限定しています。通信以外のエラーメッセージについては、ユーザーズマニュアル IM 701331-01 を参照してください。
    - 通信文法エラー 100～199
    - 通信実行エラー 200～299
    - 機種固有(その他) 300～398
    - 通信クエリエラー 400～499
    - システムエラー(通信) 399
- } 以下に記載

### 通信文法エラー (100 ~ 199) Error in communication command

コード	メッセージ	対処方法	ページ
102	構文が間違っています。 Syntax error.	以下のコード以外で構文が間違っています。	4章、5章
103	<DATA SEPARATOR>がありません。 Invalid separator.	データとデータは「、」(カンマ)で区切ってください。	4-1
104	<DATA>の種類が間違っています。 Data type error.	正しいデータ形式で記述してください。	4-6 ~ 4-7
105	デバイストリガ機能は使えません。 GET not allowed.	インタフェースメッセージに対する応答で GET はサポートしていません。	3-6
108	<DATA>が多すぎます。 Parameter not allowed.	データの数を確認してください。	4-6、5章
109	必要な <DATA>がありません。 Missing parameter.	必要なデータを記述してください。	4-6、5章
111	<HEADER SEPARATOR>がありません。 Header separator error.	ヘッダとデータはスペースで区切ってください。	4-1
112	<mnemonic>が長すぎます。 Program mnemonic too long.	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。	5章
113	そのような命令はありません。 Undefined header.	ヘッダを確認してください。	4-4、5章
114	<HEADER>の数値が間違っています。 Header suffix out of range.	ヘッダを確認してください。	4-4、5章
120	数値の仮数部分がありません。 Numeric data error.	<NRf>形式のときは数字が必要です。	4-6
123	指数が大きすぎます。 Exponent too large.	<NR3>形式のときの「E」のあとの指数を小さくしてください。	4-6、5章
124	有効桁数が多すぎます。 Too many digits.	数字は255桁以内にしてください。	4-6、5章
128	数値データは使えません。 Numeric data not allowed.	<NRf>形式以外のデータ形式で記述してください。	4-6、5章
131	単位が間違っています。 Invalid suffix.	<電圧>、<時間>、<周波数>、<電流>の単位を確認してください。	4-6
134	単位のつづりが長すぎます。 Suffix too long.	<電圧>、<時間>、<周波数>、<電流>の単位を確認してください。	4-6
138	単位は使えません。 Suffix not allowed.	<電圧>、<時間>、<周波数>、<電流>以外では単位は使えません。	4-6
141	そのような選択肢はありません。 Invalid character data.	{... ... ...}の中にある文字列を記述してください。	4-7、5章
144	<CHARACTER DATA>のつづりが長すぎます。 Character data too long.	{... ... ...}の文字列のつづりを確認してください。	4-7、5章

コード	メッセージ	対処方法	ページ
148	<CHARACTER DATA> は使えません。 Character data not allowed.	{... ... ...} 以外のデータ形式で記述してください。	4-5、5章
150	<STRING DATA> の右の区切りがありません。 String data error.	<文字列> の場合は「"」または「'」で囲ってください。4-7	
151	<STRING DATA> の内容が不適当です。 Invalid string data.	<文字列> が長すぎるか、使用不可能な文字があります。4-7、5章	
158	<STRING DATA> は使えません。 String data not allowed.	<文字列> 以外のデータ形式で記述してください。	4-6、5章
161	<BLOCK DATA> のデータ長が合っていない。 Invalid block data.	<ブロックデータ> は使用できません。	4-7、5章
168	<BLOCK DATA> は使えません。 Block data not allowed.	<ブロックデータ> は使用できません。	4-7、5章
171	<EXPRESSION DATA> の中に許されない文字があります。 Invalid expression.	演算式は使用できません。	5章
178	<EXPRESSION DATA> は使えません。 Expression data not allowed.	演算式は使用できません。	5章

### 通信実行エラー (200 ~ 299)

#### Error in communication execution

コード	メッセージ	対処方法	ページ
221	設定内容に矛盾があります。 Setting conflict.	関連のある設定値を確認してください。	5章
222	データの値が範囲外です。 Data out of range.	設定範囲を確認してください。	5章
223	データのバイト長が長すぎます。 Too much data.	データのバイト長を確認してください。	5章
224	データの値が不適当です。 Illegal parameter value.	設定範囲を確認してください。	5章
241	ハードウェアが実装されていません。 Hardware missing.	オプションの有無を確認してください。	—
260	<EXPRESSION DATA> が間違っています。 Expression error.	演算式は使用できません。	—

### 通信クエリエラー (400 ~ 499)

#### Error in communication Query

コード	メッセージ	対処方法	ページ
410	応答の送信が中断されました。 Query INTERRUPTED.	送受信の順序を確認してください。	4-2
420	送信できる応答がありません。 Query UNTERMINATED.	送受信の順序を確認してください。	4-2
430	送受信がデッドロックしました。送信を中止します。 Query DEADLOCKED.	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。	4-2
440	応答を要求する順番が間違っています。 Query UNTERMINATED after indefinite response.	*IDN?, *OPT? の後ろにはクエリを記述しないでください。—	—

## システムエラー (通信) (399) Error in System Operation

コード	メッセージ	対処方法	ページ
399	通信ドライバーエラー Fatal error in the communication driver.	サービスが必要です。	—

## 警告 (通信) (50) Warning

コード	メッセージ	対処方法	ページ
50	*OPC/? がメッセージの途中にあります。 *OPC/? exists in message.	*OPC または *OPC? は、プログラムメッセージの最後においてください。	—

## 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。	6-6

### **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? クエリにのみ出力されるエラーで、画面には表示されません。

## 付録3 波形パラメータ名称対応表

機器本体画面の設定メニューに表示される名称	通信コマンドで使用する名称	測定結果を機器本体画面に表示するときの名称
Max	MAXimum	Max
Min	MINimum	Min
High	HIGH	High
Low	LOW	Low
P-P	PTOPeak	P-P
Hi-Low	HiLow	Hi-Low
+Over	POVershoot	+Over
-Over	NOVershoot	-Over
Rms	RMS	Rms
Mean	MEAN	Mean
Sdev	SDEViation	Sdev
IntegTY	TYINteg	ITY
C.Rms	CRMS	CRms
C.Mean	CMEan	CMean
C.Sdev	CSDeviation	CSdev
C.IntegTY	TYCInteg	CITY
Freq	FREQuency	Freq
1/Freq	PERFrequency	1/FR
Count	COUNT	Count
Burst	BURSt	Burst
+Width	PWIDth	+Width
-Width	NWIDth	-Width
Period	PERiod	Period
Duty	DUTYcycle	Duty
Rise	RISE	Rise
Fall	FALL	Fall
Delay	DELay	Dly



## 付録 4 Eye Pattern パラメータ名称対応表

機器本体画面の設定メニューに表示される名称	通信コマンドで使用する名称	測定結果を機器本体画面に表示するときの名称
Crossing %	PCROSSing	Cross%
Eye Height	EHEight	EyeHi
Eye Width	EWIDth	EyeWid
Q Factor	QFACtor	QFact
Jitter	JITTer	Jitter
Duty Cycle Distraction %	PDUTYcycle	DCDTime%
Vtop	VTOP	Vtop
Vbase	VBASE	Vbase
$\sigma$ top	SDTop	$\sigma$ top
$\sigma$ base	SDBase	$\sigma$ base
Tcrossing1	T1CROSSing	Tcros1
Tcrossing2	T2CROSSing	Tcros2
Vcrossing	VCROSSing	Vcros
Ext Rate dB	DBERate	ERdB
Rise	RISE	Rise
Fall	FALL	Fall

# 索引

## 記号

記号	ページ
<Boolean>.....	4-7
<NRf>.....	4-7
<時間>.....	4-6
<周波数>.....	4-6
<電圧>.....	4-6
<電流>.....	4-6

## A

A	ページ
ACQuire グループ.....	5-42
ANALySis グループ.....	5-43
ASeTup グループ.....	5-72

## C

C	ページ
Calc アイテム.....	5-140
CALibrate グループ.....	5-73
CHANnel グループ.....	5-74
CLEar グループ.....	5-76
COMMunicate グループ.....	5-77
CONTInuous Statistics.....	5-141
CURSor グループ.....	5-79
Cycle Statistics.....	5-141

## D

D	ページ
DISPlay グループ.....	5-90
DL シリズ用ライブラリ.....	ii

## E

E	ページ
Edge/State トリガ.....	5-239
Enhanced トリガ.....	5-219

## F

F	ページ
FFT 演算	
演算式.....	5-58
自動測定.....	5-52
水平軸のモード.....	5-52
測定対象ウィンドウ.....	5-59
中心値 / スパン.....	5-51
ピーク周波数値.....	5-57
ピーク値測定.....	5-56
マーカーカーソル測定.....	5-53
FFT 演算機能.....	5-51
FFT サーチ.....	5-114
FFT 点数.....	5-52
FFT 判定.....	5-103

## G

G	ページ
GO/NO-GO 判定.....	5-99
GONogo グループ.....	5-99

## H

H	ページ
Hcopy グループ.....	5-109
HF リジェクション.....	5-237
History Statistics.....	5-142
HISTory グループ.....	5-111

## I

I	ページ
IIR フィルタ演算.....	5-129
IMAGe グループ.....	5-122
INITialize グループ.....	5-123

## L

L	ページ
LOGic グループ.....	5-124

## M

M	ページ
MATH グループ.....	5-128
MEASure グループ.....	5-137

## N

N	ページ
NO-GO 時の動作.....	5-99

## R

R	ページ
Rectangle サーチ.....	5-118
Rectangle 判定.....	5-107
REference グループ.....	5-150
Remote Control.....	1-5, 2-5, 3-5

## S

S	ページ
SEARch グループ.....	5-151
SNAP グループ.....	5-163
SStart グループ.....	5-164
StARt グループ.....	5-165
StATus グループ.....	5-166
StOP グループ.....	5-167
SYSTem グループ.....	5-168

## T

T	ページ
T/div.....	5-174
TCP/IP の設定.....	2-6
TELecomtest グループ.....	5-169
Threshold.....	5-142
TIMebase グループ.....	5-174
TMCTL.....	ii
TRIGger グループ.....	5-175
TV トリガ.....	5-236

## U

U	ページ
USB インタフェース.....	1-2
USB ケーブル.....	1-4
USB ハブ.....	1-4

## V

V	ページ
V/div.....	5-75
VT カーソル.....	5-87
VT 波形の表示の ON/OFF.....	5-60

## W

W	ページ
WAVeform グループ.....	5-257
WAVE サーチ.....	5-119
WAVE 判定.....	5-107

## X

X	ページ
XY サーチ.....	5-118
XY 判定.....	5-107

## 索引

XY 表示	5-69
カーソル測定	5-69
自動測定	5-69
自動測定モード	5-70
積分	5-70
測定対象ウインドウ	5-71
XY 表示機能	5-68

## Z ページ

ZOOM グループ	5-260
-----------	-------

## A ページ

アキュムレート	5-90
アキュムレートヒストグラム	
Basic アイテム	5-44
Calc アイテム	5-45
演算式	5-46, 5-49
カーソル測定	5-44
カーソルの ON/OFF	5-45
自動測定	5-44, 5-47
垂直カーソル	5-47
垂直範囲	5-50
水平カーソル	5-46
水平軸方向範囲	5-43
測定対象ウインドウ	5-50
測定値	5-45
対象トレース	5-50
波形パラメータの自動測定	5-47
波形パラメータの垂直範囲	5-50
モード	5-50
アキュムレートヒストグラム機能	5-43
アクションオントリガ	5-175
アクティブレベル (XY 表示)	5-69
アベレージング	5-42

## I ページ

イーサネットインタフェース	2-2
移動平均演算	5-130
イニシャライズ	5-123
イベントインターバル	5-177
インターバル時間	5-218
インターリーブ	5-42

## U ページ

上書き禁止	5-95
-------	------

## E ページ

エラーキュー	6-6
エラーメッセージ	付-2
エラー率	5-101
エラー率 (テレコムテスト)	5-171
演算	5-128
演算基準点	5-131
演算子	5-133

## O ページ

応答	4-5
応答メッセージ	4-1
オートスクロール機能	5-260
オートセットアップ	5-72
オーバーラップ	5-77
オーバーラップコマンド	4-8
オフセット (演算波形)	5-134
オフセットキャンセル	5-75
オフセット電圧	5-75

## カ ページ

カーソル測定	5-79
カーソルの ON/OFF	5-79
カーソルの種類	5-85
解析機能	5-43
解析機能のタイプ	5-59
解像度	5-238
外部トリガ	5-255
拡張イベントレジスタ	5-166, 6-5
加算	5-133
画面イメージデータ	5-122
画面データの出力	5-109

## キ ページ

機種	5-263
キャリブレーション	5-73
共通コマンド	4-3
共通コマンドグループ	5-263

## ク ページ

クエリ	4-1, 4-4
グラティクル	5-91
クリック音	5-168
グループ	4-3
グループ名	5-126
クロックソース	5-126
クロックチャネル	5-152, 5-176
クックトレース	5-151

## ケ ページ

継続統計処理	5-141
継続統計処理の回数	5-145
ゲート機能 (XY 表示)	5-68
ゲートトレース	5-69
減算	5-132
減衰定数	5-42

## コ ページ

高分解能モード	5-42
---------	------

## サ ページ

サーチ条件 (ヒストリサーチ)	5-113
サーチロジック	5-152
サービスリクエストイネーブルレジスタ	5-265
最小レコード番号	5-258
最大値保持機能	5-52
サンプルレート	5-174

## シ ページ

しきい値	5-142
システム	5-168
周期統計処理	5-141
出力キュー	6-6
上位クエリ	4-4
乗算	5-132
状態レジスタ	5-166, 6-5
省略形	4-5
シリアルカーソル	5-83
シリアルポール	5-166

## ス ページ

垂直カーソル	5-85
垂直軸感度	5-75
垂直ポジション	5-75, 5-126
水平カーソル	5-79

ズーム対象波形	5-260	<b>二</b>	ページ
ズームポジション	5-261	入力カップリング	5-74
スキップモード	5-151	入力フィルタ	5-74
スキュー補正	5-74, 5-124	<b>ハ</b>	ページ
スケーリング	5-134	パーシスタンス	5-90
スケール値表示の ON/OFF	5-75	波形ゾーンでの検索	5-119
ステータスバイト	6-3	波形データのビット長	5-257
ステータスバイトレジスタ	5-265	波形取り込みモード	5-42
ステータスレポート	6-1	波形のアクイジションモード	5-259
ステートクロック	5-126	波形の重ね書き表示	5-90
ステート表示	5-125	波形のクリア	5-76
スナップショット	5-163	波形のズーム	5-260
スレシヨルドレベル	5-127	波形の全データ点数	5-257
<b>セ</b>	ページ	波形の取り込み	5-42
積分値 (XY 判定)	5-107	波形取り込み回数	5-42
積分演算	5-130	波形の取り込みスタート	5-165
セルフテスト	5-266	波形の取り込みストップ	5-167
遷移フィルタ	5-166, 6-5	波形の割り当て	5-92
<b>ソ</b>	ページ	波形パラメータ測定	5-60
ゾーン/パラメータ判定	5-102	エリア	5-65
測定対象ウィンドウ (テレコムテスト)	5-173	トレース	5-65
測定範囲 (テレコムテスト)	5-173	波形パラメータの自動測定	5-137
測定範囲 (波形パラメータの自動測定)	5-149	波形パラメータ名称対応表	付-5
<b>タ</b>	ページ	波形ラベル名	5-74
対象ウィンドウ (ヒストリサーチ)	5-119	バス表示	5-125
対象トレース (ヒストリサーチ)	5-118	パスワード	2-5
タイムアウト時間	2-6	バックライト	5-90
タイムベース	5-174	パルス幅サーチ	5-162
<b>チ</b>	ページ	パルス幅トリガ	5-218, 5-255
遅延演算	5-129	判定基準 (GO/NO-GO)	5-100
チャンネル間ディレイ測定	5-146	判定の種類 (GO/NO-GO)	5-100
チャンネルの ON/OFF	5-74	反転表示	5-74
<b>ツ</b>	ページ	反転表示の ON/OFF	5-131
通信のステータス機能	5-166	バンドル	5-125
<b>テ</b>	ページ	<b>ヒ</b>	ページ
ディスタル/プロキシマル値	5-148	ヒストグラム表示 (波形パラメータ)	5-61
データ	4-6	ヒストリ機能	5-111
データ転送速度	1-2, 2-2	ヒストリサーチ機能	5-113
データ補間	5-42	ヒストリデータの統計処理	5-142
デッドロック状態	4-2	ヒストリ波形 (リファレンス)	5-120
テレコムテスト	5-169	ヒストリ波形の表示	5-112
テレコムテスト判定	5-100	ビット配置	5-125
<b>ト</b>	ページ	ビット名	5-125
等価時間サンプリング	5-42	表示サイズ	5-126
統計処理の回数	5-140	表示フォーマット	5-91
トリガ	5-175	表示補間方式	5-92
トリガソース	5-253	標準イベントレジスタ	6-4
トリガディレイ	5-176	<b>フ</b>	ページ
トリガの種類	5-255	フィルタ	5-129
トリガポジション	5-253	フィルタのタイプ	5-130
トリガモード	5-253	プローブ減衰比	5-75
トレンド表示 (波形パラメータ)	5-65	プログラムメッセージ	4-1
<b>ナ</b>	ページ	ブロックデータ	4-7
内蔵ハードディスクのフォーマット	5-168	フロントパネル	1-1, 2-1, 3-1
内蔵プリンタ出力	5-110	<b>ヘ</b>	ページ
内部メモリのフォーマット	5-168	ヘッダ	4-5, 5-77
		<b>ホ</b>	ページ
		ホールドオフ検出	5-152
		ホールドオフ時間	5-240

## 索引

---

### マ ページ

マーカーカーソル .....	5-81
マスクテスト .....	5-171
マスク判定 .....	5-101
窓関数 .....	5-59

### メ ページ

命令 .....	4-3
メジャーサーチ .....	5-116
メジャー判定 .....	5-106
メッセージ .....	4-1
目盛り .....	5-91

### モ ページ

モード (波形パラメータ) .....	5-65
文字データ .....	4-7
文字の表記法 .....	iv
文字列データ .....	4-7

### ユ ページ

ユーザー定義 (TV トリガ) .....	5-238
ユーザー認証 .....	2-3
ユーザー名 .....	2-5

### ラ ページ

ライン (TV トリガ) .....	5-237
ラベル名 .....	5-125

### リ ページ

リアパネル .....	1-1, 2-1, 3-1
リスケーリング (フィルタ) .....	5-130
リスト表示 (波形パラメータ) .....	5-64
リファレンスのヒストリ機能 .....	5-120
リファレンスの表示 .....	5-150
リモート/ローカル .....	5-78
リモート/ローカル切り替え .....	1-3, 2-3

### レ ページ

レコード長 .....	5-42
-------------	------

### ロ ページ

ローカルロックアウト .....	5-77
ローパスフィルタ .....	5-237
ロジックグループ .....	5-125
ロジックサーチ .....	5-153
ロジックトリガ .....	5-240

### ワ ページ

ワードフォーマット .....	5-257
-----------------	-------