

DLM5034HD、DLM5038HD、 DLM5054HD、DLM5058HD

**高分解能オシロスコープ
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、高分解能オシロスコープ DLM5034HD、DLM5038HD、DLM5054HD、DLM5058HD 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ズマニュアルは、次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 (ソケットインタフェースを含む)
- GP-IB 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、本機器のマニュアルとして、次節の「マニュアルの構成」に示すマニュアルがあります。あわせてお読みください。

各国や地域の当社営業拠点の連絡先は、次のシートに記載されています。

ドキュメント No.	内容
PIM 113-01Z2	国内海外の連絡先一覧

ご注意

- 性能・機能の向上などにより、本書の内容を予告なしに変更することがあります。最新のマニュアルは、当社 Web サイトでご確認ください。
- 本書に記載の画面表示内容は実際のものとは多少異なることがあります。
- 本書の内容に関しては万全を期していますが、万一の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。
- 本製品の TCP/IP ソフトウェア、および TCP/IP ソフトウェアに関するドキュメントは、カリフォルニア大学からライセンスされた BSD Networking Software, Release 1 をもとに当社で開発 / 作成したものです。

商標

- Microsoft、MS-DOS、Windows、Windows 10、および Windows 11 は、米国 Microsoft Corporation の、米国およびその他の国における登録商標または商標です。
- Adobe、Acrobat は、アドビシステムズ社の登録商標または商標です。
- DLM は横河電機株式会社の登録商標です。
- 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。

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 側に次のソフトウェアが必要です。

- DLM シリーズ用通信ライブラリ (TMCTL)
- PC-DLM5000HD シリーズ間の USB 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 側に次のソフトウェアが必要です。

- DLM シリーズ用通信ライブラリ (TMCTL)

上記のライブラリおよびドライバは、次の Web サイトの提供ページからダウンロードできます。

<https://tmi.yokogawa.com/jp/library/>

サンプルプログラム

サンプルプログラムは、次の Web サイトの提供ページからダウンロードできます。

<https://tmi.yokogawa.com/jp/library/>

履歴

- 2023 年 9 月 初版発行
- 2024 年 7 月 2 版発行

マニュアルの構成

本機器のマニュアルとして、このマニュアルを含め、次のものがあります。あわせてお読みください。

製品に添付されているマニュアル

マニュアル名	マニュアル No.	内容
DLM5034HD、DLM5038HD、 DLM5054HD、DLM5058HD 高分解能オシロスコープ スタートガイド	IM DLM5058HD-03JA	本機器の取り扱い上の注意、共通操作、 困ったときの対処方法、仕様について記 述しています。
DLM5034HD、DLM5038HD、 DLM5054HD、DLM5058HD 高分解能オシロスコープ オペレーションガイド	IM DLM5058HD-04JA	本機器の基本的な操作について説明して います。「本機器の準備」から「波形の表 示」「波形の測定」「画面イメージの保存」 までを段階を追って説明しています。
DLM5034HD、DLM5038HD、 DLM5054HD、DLM5058HD 高分解能オシロスコープ マニュアルのダウンロードのお願い	IM DLM5058HD-73JA	Web サイトで提供しているマニュアルに ついて説明しています。
DLM5034HD、DLM5038HD、 DLM5054HD、DLM5058HD High Definition Oscilloscope	IM DLM5058HD-92Z1	中国向け文書
Safety Instruction Manual	IM 00C01C01-01Z1	安全マニュアル (欧州の言語)

Web サイトで提供しているマニュアル

次のマニュアルは当社の Web サイトからダウンロードしてご使用ください。

マニュアル名	マニュアル No.	内容
DLM5034HD、DLM5038HD、 DLM5054HD、DLM5058HD 高分解能オシロスコープ ユーザーズマニュアル [機能編]	IM DLM5058HD-01JA	通信インタフェースの機能を除く、本機 器の全機能について説明しています。
DLM5034HD、DLM5038HD、 DLM5054HD、DLM5058HD 高分解能オシロスコープ ユーザーズマニュアル [操作編]	IM DLM5058HD-02JA	本機器の各設定操作について説明してい ます。
DLM5034HD、DLM5038HD、 DLM5054HD、DLM5058HD 高分解能オシロスコープ 通信インタフェースユーザーズマニュアル	IM DLM5058HD-17JA	本書です。本機器の通信インタフェース の機能について、設定方法や、インタ フェースを使って PC から本機器をコン トロールするコマンドについて説明して います。

マニュアルのダウンロードについては、マニュアルのダウンロードのお願い (IM DLM5058HD-73JA) をご覧ください。

マニュアル No. の「JA」、「Z1」は言語コードです。

オンラインヘルプ

ユーザーズマニュアル [機能編] (IM DLM5058HD-01JA) と同様の内容が、ヘルプとして本機器に組み込まれています (内容を一部省略している場合があります)。ヘルプの操作方法については、スタートガイド (IM DLM5058HD-03JA) の 3.10 節をご覧ください。

このマニュアルの利用方法

このマニュアルの章構成

このユーザーズマニュアルは、以下に示す第 1 章～第 6 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

イーサネットインタフェースの機能・仕様や、ソケットインタフェースの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 (オプション)

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、通信に関するエラーメッセージ、IEEE 488.2-1992 などについて説明しています。

索引

アルファベット順、50 音順の索引があります。

このマニュアルで使用している記号と表記法

接頭語の k と K について

単位の前に使用される接頭語の k と K を、次のように区別して使用しています。

k……1000 の意味です。 使用例：100 kS/s(サンプルレート)

K……1024 の意味です。 使用例：720 K バイト (ファイルの容量)

表示文字

操作説明文にある太字の文字は、操作対象のパネル上の文字や画面上のメニューの文字を示します。

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。

警告

取り扱いを誤った場合に、使用者が死亡または重傷を負う危険があるときに、その危険を避けるための注意事項が記載されています。

注意

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。

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、4-6 ～ 4-7 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}	COUPLING AC
	排他的論理和		
[]	省略可能	CURSor[:TY]:TYPE	CURSor:TYPE

目次

はじめに	i
マニュアルの構成	iii
このマニュアルの利用方法	iv
このマニュアルで使用している記号と表記法	v

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

2.1 各部の名称と機能	2-1
2.2 イーサネットインタフェースの機能と仕様と、 ソケットインタフェースの仕様	2-2
2.3 イーサネットインタフェースによる接続	2-5
2.4 本体の設定 (ネットワーク)	2-6

第 3 章 GP-IB インタフェースについて (オプション)

3.1 各部の名称と機能	3-1
3.2 GP-IB インタフェースの機能と仕様	3-2
3.3 GP-IB インタフェースによる接続	3-4
3.4 本体の設定 (GP-IB)	3-5
3.5 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	5-60
5.3 ANALysis グループ	5-61
5.4 ASETup グループ	5-81
5.5 CALibrate グループ	5-82
5.6 CHANnel グループ	5-83
5.7 CHUTil グループ	5-86
5.8 CLEar グループ	5-87
5.9 COMMunicate グループ	5-88
5.10 CURSor グループ	5-90
5.11 DISPlay グループ	5-98
5.12 FFT グループ	5-101
5.13 FILE グループ	5-109
5.14 GONogo グループ	5-113
5.15 HCOPy グループ	5-119
5.16 HISTory グループ	5-121

5.17	IMAGe グループ	5-127
5.18	INITialize グループ	5-129
5.19	LOGic グループ	5-130
5.20	MATH グループ	5-137
5.21	MEASure グループ	5-143
5.22	RECall グループ	5-154
5.23	REFeRence グループ	5-155
5.24	SEARch グループ	5-156
5.25	SERialbus グループ	5-162
5.26	SNAP グループ	5-249
5.27	SSTarT グループ	5-250
5.28	STARt グループ	5-251
5.29	STATus グループ	5-252
5.30	STOP グループ	5-253
5.31	STORe グループ	5-254
5.32	SYSTem グループ	5-255
5.33	TIMEbase グループ	5-260
5.34	TRIGger グループ	5-261
5.35	WAVEform グループ	5-338
5.36	WPARameter グループ	5-341
5.37	XY グループ	5-345
5.38	ZOOM グループ	5-348
5.39	共通コマンドグループ	5-350

第 6 章

ステータスレポート

6.1	ステータスレポートについて	6-1
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
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

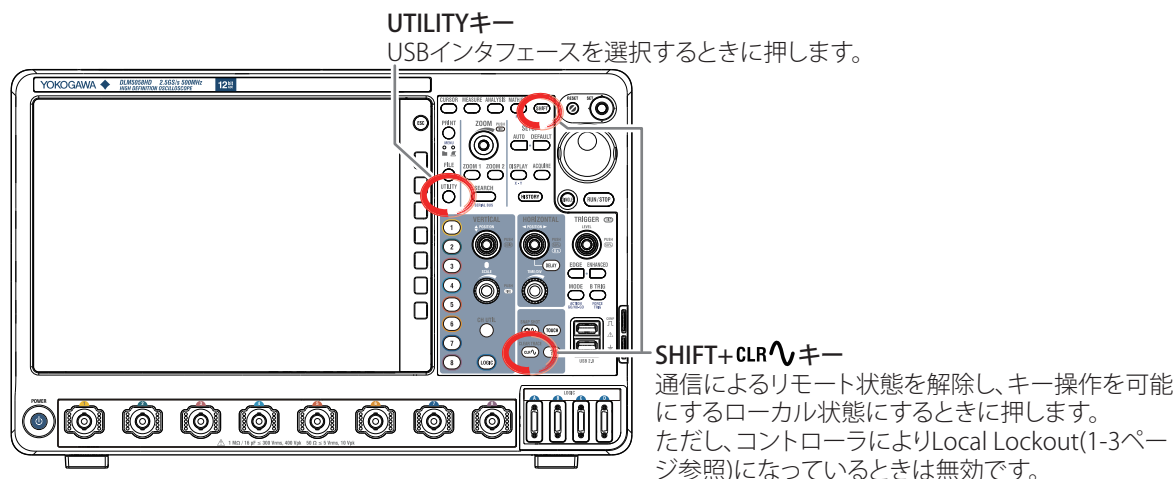
付録

付録 1	ASCII キャラクタコード	付 -1
付録 2	エラーメッセージ	付 -2
付録 3	IEEE 488.2-1992 について	付 -5

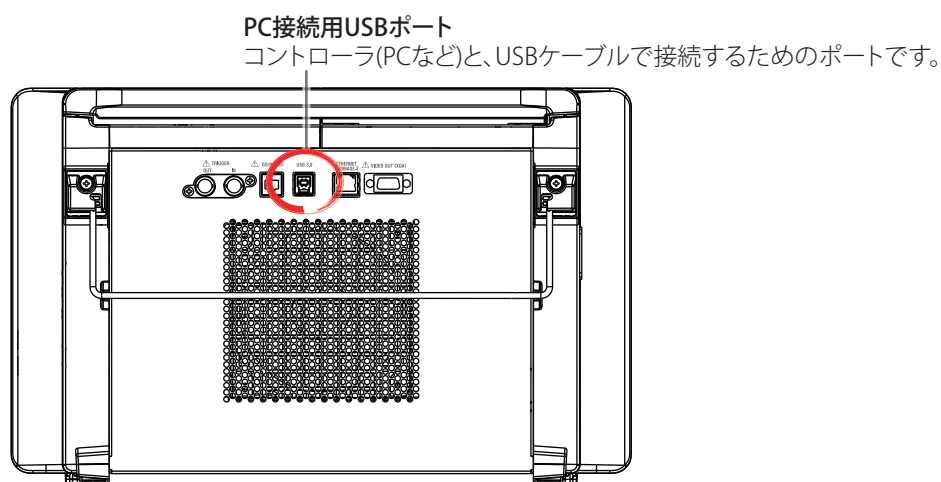
索引

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. 3.0 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 10(32 bit、64 bit) または Windows 11 の日本語版または英語版が動作する PC(別途デバイスドライバが必要)

データ転送速度

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。

対象モデル：DLM5058HD

コントローラ：PC(Intel(R) Xeon(R) E-2124 CPU @ 3.30GHz 3.31GHz)、USB(Intel(R) USB 3.1 eXtensible Host Controller - 1.10(Microsoft))、OS(Windows 10 Enterprise 64 bit)

使用言語：Visual C++

データ点数	ワードデータの場合
125000	9 ms
250000000	7160 ms

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

リモート→ローカル切り替え時

リモート状態のときに **SHIFT+CLR** キーを押すと、ローカル状態になります。ただし、コンピュータから「:COMMunicate:LOCKout ON」コマンドを受信しているときは無効です。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**RMT**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。
- PC 接続用 USB ポートに PC を接続するときは、接続する前に PC を本機器の接地と同じ電位に接地してください。
- PC 接続用 USB ポートには、コントローラ (PC など) 以外の機器を接続しないでください。

1.4 本体の設定 (USB)

ここでは、USB インタフェースでリモートコントロールをするときの通信インタフェースの設定について説明しています。

UTILITY Remote Control メニュー

UTILITY キー > Remote Control のソフトキーを押します。次のメニューが表示されます。

画面左上の MENU (E) をタップして、表示されるトップメニューの UTILITY から UTILITY メニューを選択することもできます。

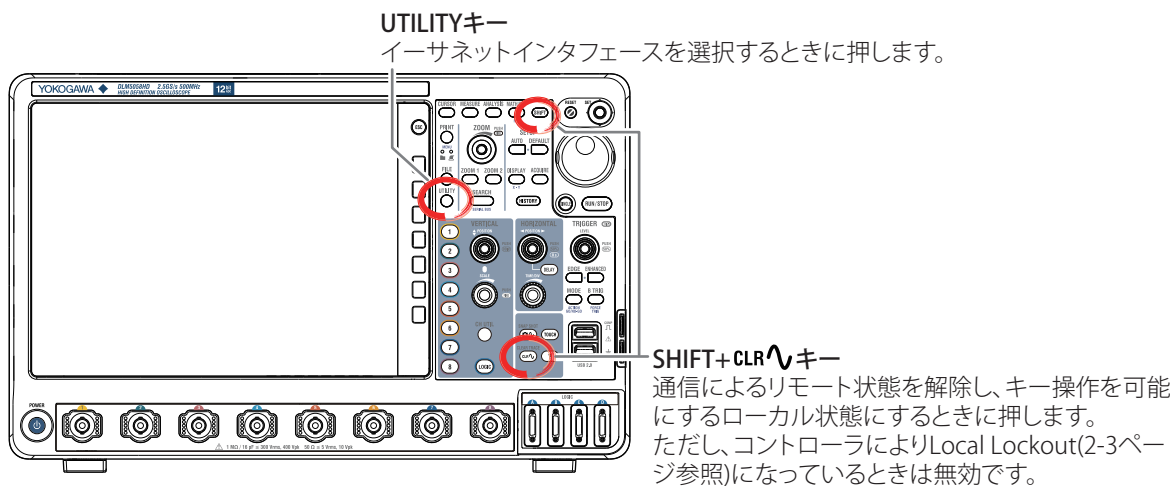


Note

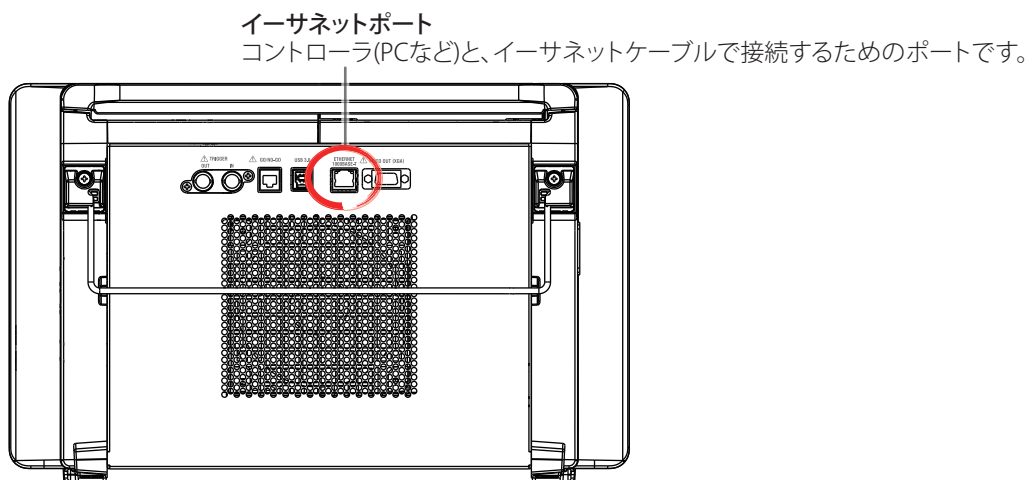
- 選択していない他の通信インタフェースにも同時にコマンドを送信すると、コマンドが正常に実行されません。
- USB ポートを使って、通信コマンドで本機器をコントロール (リモートコントロール) するには、上記メニューの USB Function を「USB TMC」に設定し、次の事項を実行してください。
 - USB Function の設定を有効にするには、本機器を再起動する必要があります。本機器の電源スイッチをオフにしてから、10 秒以上待ったあとにオンにしてください。
 - 当社の USB ドライバを PC にインストールしてください。当社の USB ドライバの入手方法については、お買い求め先にお問い合わせいただくか、次の当社 Web サイトから USB ドライバ提供ページにアクセスし、USB ドライバをダウンロードしてください。
<https://tmi.yokogawa.com/jp/library/>
- 当社以外の USB ドライバ (またはソフトウェア) は、使用しないでください。

2.1 各部の名称と機能

フロントパネル



リアパネル



2.2 イーサネットインタフェースの機能と仕様と、ソケットインタフェースの仕様

イーサネット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

プロトコル：VXI-11

対応システム環境：Windows 10(32 bit、64 bit) または Windows 11 の日本語版または英語版が動作する PC

データ転送速度

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。

対象モデル：DLM5058HD

コントローラ：PC(Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz 2.71GHz)、Ethernet(Intel(R) Ethernet Connection(4)I219-LM)、OS(Windows 10 Enterprise 64 bit)

使用言語：Visual C++

データ点数	ワードデータの場合
125000	14 ms
250000000	18900 ms

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

リモート→ローカル切り替え時

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

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

タイムアウト時間

ある一定時間 (タイムアウト時間) を過ぎても本機器へのアクセスがない場合、本機器がネットワークとの接続を閉じます。0 ~ 3600s と Infinite の範囲で設定できます。初期定設は Infinite です。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。

ソケットインタフェースの仕様

電氣的・機械的仕様：IEEE 802.3 に準拠

同時接続数： 1

プロトコル： なし

ポート番号： 10002/tcp

ターミネータ：LF(0Ah)

対応システム環境：Windows 10(32 bit、64 bit) または Windows 11

制限事項：一度に送信可能なコマンドのバイト数は、最大 4K バイト

ソケット通信サンプルプログラム

```
#include "stdafx.h"
#include <stdio.h>
#include <winsock2.h>
#include <ws2tcpip.h>

#pragma comment (lib, "Ws2_32.lib")

int _tmain(int argc, _TCHAR* argv[])
{
    /* ソケット */
    int dstSocket;

    /* sockaddr_in 構造体 */
    struct sockaddr_in dstAddr;

    /* 受信バッファ */
    char buffer[200];

    /* *****/
    /* Windows Socket API の初期化处理 */
    WSADATA data;
    WSStartup(MAKEWORD(2,0), &data);

    /* sockaddr_in 構造体のセット */
    memset(&dstAddr, 0, sizeof(dstAddr));
    dstAddr.sin_port = htons(10002);
    dstAddr.sin_family = AF_INET;
    dstAddr.sin_addr.s_addr = inet_addr("192.168.0.1");

    /* ソケット生成 */
    dstSocket = socket(AF_INET, SOCK_STREAM, 0);

    /* 接続 */
    connect(dstSocket, (struct sockaddr *) &dstAddr, sizeof(dstAddr));

    /* パケット送出 */
    send(dstSocket, "*IDN?¥n", 6, 0);

    /* パケット受信 */
    recv(dstSocket, buffer, 200, 0);

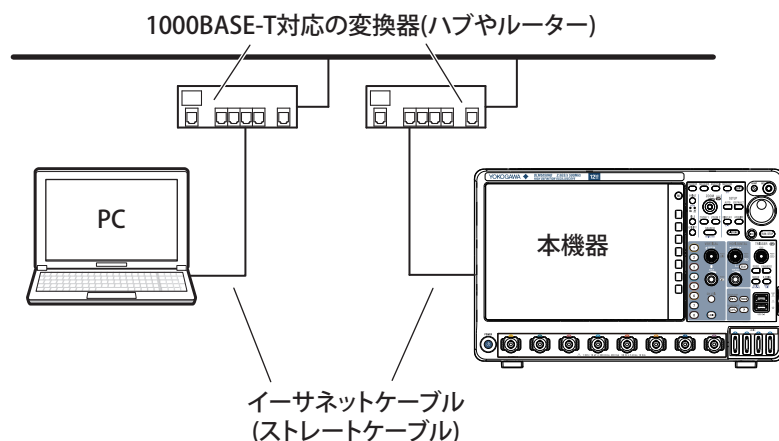
    /* Windows Socket API の終了処理 */
    closesocket(dstSocket);
    WSACleanup();

    return 0;
}
```

2.3 イーサネット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イーサネットケーブルを本機器のリアパネルにあるイーサネット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、必ずハブやルーター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、動作を保証できません。
- ・ ご使用のネットワーク環境(伝送速度)に対応したケーブルを使用してください。

Note

接続方法の詳細については、ユーザーズマニュアル[操作編](IM DLM5058HD-02JA)の「18.1 本機器をネットワークに接続する」をご覧ください。

2.4 本体の設定 (ネットワーク)

ここでは、イーサネットインターフェースでリモートコントロールをするときの次の設定について説明しています。

- ・ 通信インターフェース
- ・ VXI-11 とソケット (Socket) の ON/OFF
- ・ ネットワーク接続のタイムアウト時間

UTILITY _ Remote Control メニュー

UTILITY キー > Remote Control のソフトキーを押します。次のメニューが表示されます。

画面左上の MENU() をタップして、表示されるトップメニューの UTILITY から UTILITY メニューを選択することもできます。



Note

- ・ 選択していない他の通信インターフェースにも同時にコマンドを送信すると、コマンドが正常に実行されません。
- ・ VXI-11 または Socket が ON のときは、イーサネットインターフェースまたはソケットインターフェースが使用可能です。これらの機能を使用しないときは、OFF にすることをお勧めします。

TCP/IP

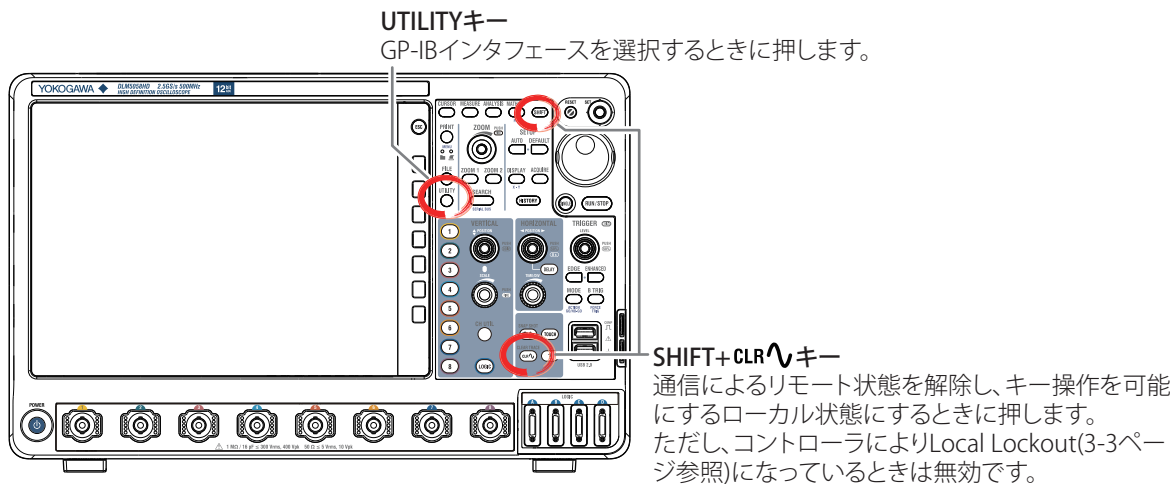
イーサネットインターフェース機能を利用するには、TCP/IP の以下の設定が必要です。

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

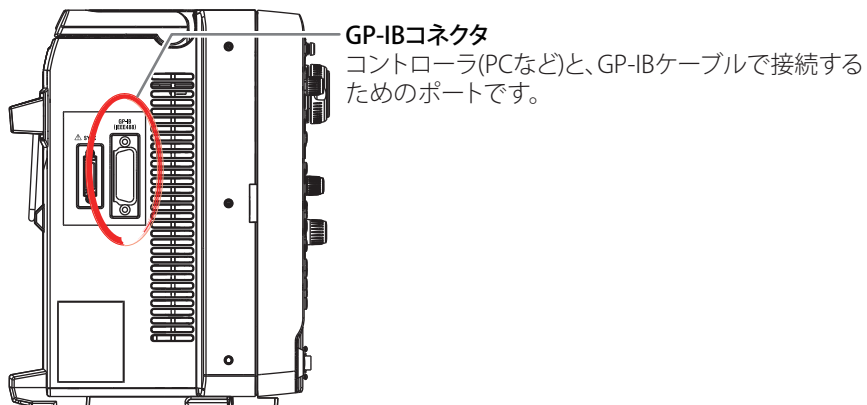
これらの設定方法の詳細については、ユーザズマニュアル [操作編] (IM DLM5058HD-02JA) の「18.2 TCP/IP の設定をする」をご覧ください。

3.1 各部の名称と機能

フロントパネル



サイドパネル



3.2 GP-IB インタフェースの機能と仕様

GP-IB インタフェースの機能

リスナー機能

- ・ 電源のオン / オフ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。
- ・ 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。
- ・ 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

- ・ 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。

Note

リスンオンリー、トークオンリー、およびコントローラ機能はありません。

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：UTILITY 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：**SHIFT+CLR** キーを押すことで、リモート状態の解除可能。ただし、コントローラにより Local Lockout されているときは無効。

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、トークオンリー機能なし
リスナー	L4	基本リスナー機能、MTA(My Talk Address) によるリスナー解除機能あり、リスンオンリー機能なし
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	C0	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

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。

対象モデル：DLM5058HD

コントローラ：PC(Intel(R) Xeon(R) E-2124 CPU @ 3.30GHz 3.31GHz)、GP-IB(NI PCI-GPIB)、
OS(Windows 10 Enterprise 64 bit)

使用言語：Visual C++

データ点数	ワードデータの場合
125000	370 ms
250000000	734085 ms

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

リモート→ローカル切り替え時

リモート状態のときに **SHIFT+CLR** キーを押すと、ローカル状態になります。ただし、コントローラにより Local Lockout(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 **RMT** 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。

3.3 GP-IB インタフェースによる接続

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 PC および本機器の電源をオフにしてください。オフにしないと、誤動作を生じたり、内部回路を破損することがあります。

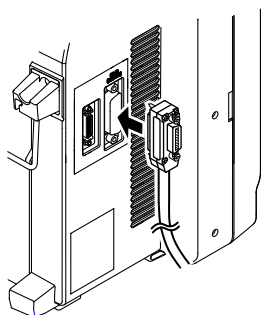
GP-IB ケーブル

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

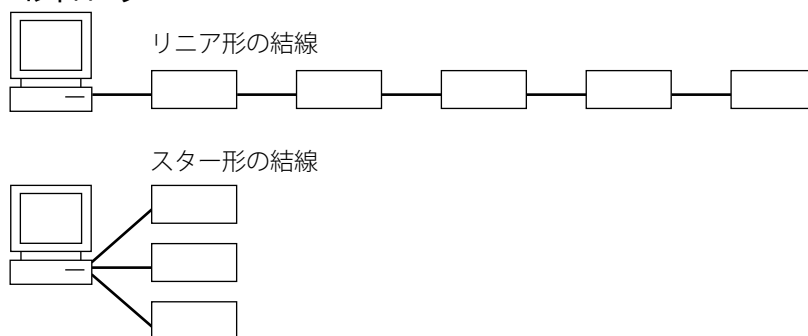
接続方法

下図のように接続してください。複数の機器を接続するときは、下図に示すようリニア形またはスター形の結線にしてください。その組み合わせも可能です。ループ形の結線はできません。

本機器



コントローラ



接続時の注意

- ・ GP-IB ケーブルのコネクタに付いているねじは、しっかりと固定してください。
- ・ PC 側の GP-IB ボード (またはカード) には、N.I(ナショナルインスツルメンツ) 社製をご使用ください。詳細については、3.2 節をご覧ください。
- ・ 本機器と PC 間を接続する通信ケーブルの途中に変換器を介した場合 (たとえば、GP-IB と USB 変換のように)、正常に動作しないときがあります。詳細については、お買い求め先にお問い合わせください。
- ・ 複数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 2 m 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 20 m 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オンにしてください。

3.4 本体の設定 (GP-IB)

ここでは、GP-IB インタフェースでリモートコントロールをするときの次の設定について説明しています。

- ・ 通信インタフェース
- ・ GP-IB アドレス

UTILITY Remote Control メニュー

UTILITY キー > Remote Control のソフトキーを押します。次のメニューが表示されます。

画面左上の MENU() をタップして、表示されるトップメニューの UTILITY から UTILITY メニューを選択することもできます。



Note

- ・ 選択していない他の通信インタフェースにも同時にコマンドを送信すると、コマンドが正常に実行されません。
- ・ GP-IB を介してコントローラが 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。
- ・ 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 PC 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。

3.5 インタフェースメッセージに対する応答

インタフェースメッセージに対する応答

ユニラインメッセージに対する応答

IFC(Interface Clear)

トーカー、リスナーを解除します。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 (6-6 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 SHIFT+CLR キーの操作を無効にし、ローカル状態への移行を禁止します。

DCL(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を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

機器がリスナーあるいはトーカー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。

リスナー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)

トーカーに指定している機器に有効なコマンド

- TCT(Take Control)

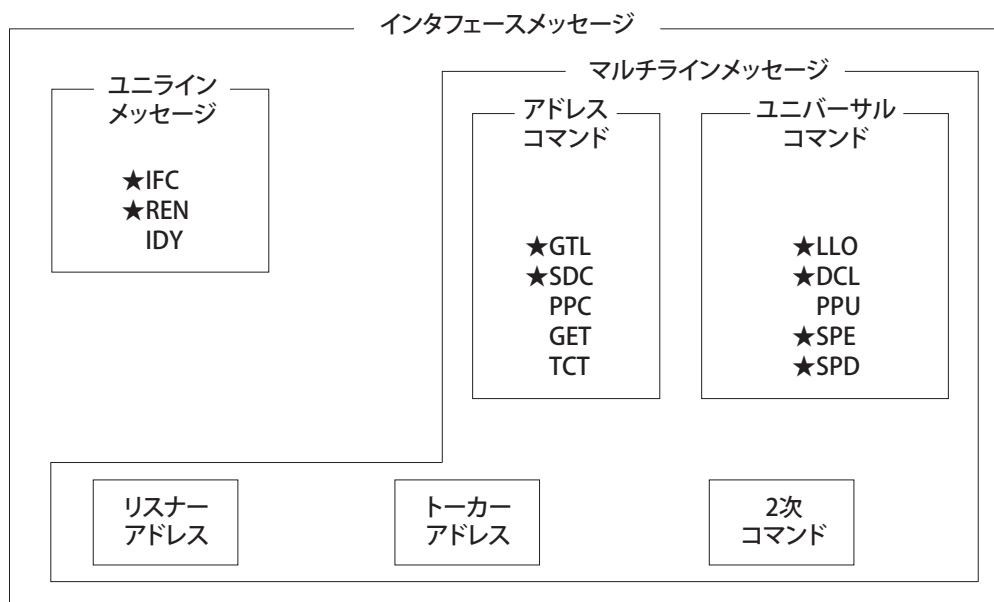
ユニバーサルコマンド

リスナー・トーカー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)

3.5 インタフェースメッセージに対する応答

その他、インタフェースメッセージとして、リスナーアドレス、トーカーアドレス、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 はトーカー・リスナーの指定が必要なアドレスコマンド、DCL はトーカー・リスナー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

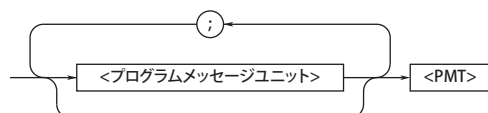
メッセージ

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;RESolution 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:

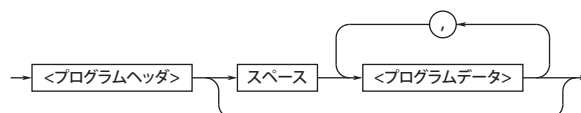
USBTMCで定義されているENDメッセージ(ENDメッセージと同時に送信されたデータバイトは、プログラムメッセージの最後のデータになります。)

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。

例:

```
:ACQuire:MODE NORMAL<PMT>
```

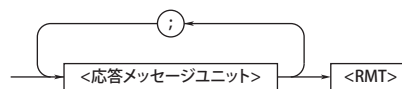
ヘッダ データ

```
:MATH1:OPERation PLUS,1,2<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;RESolution 1<RMT>
```

ユニット ユニット

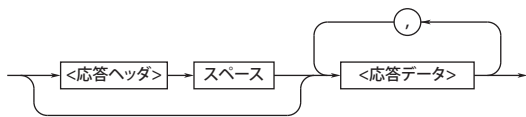
<RMT>

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、4-5 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すると、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。

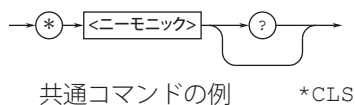
4.2 命令

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。

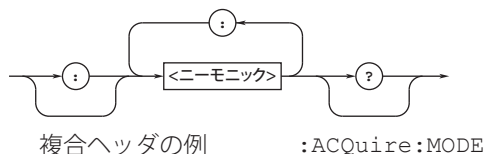
共通コマンドヘッダ

USB TMC-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) 付けます。



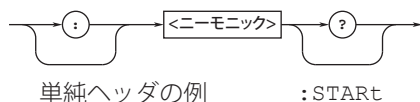
複合ヘッダ

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) 付けます。



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。



Note

<ニーマモニック> 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:RLENgth

• 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;SAMPling REAL

<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

USB TMC-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;

SAMPling REAL<PMT>

4.2 命令

• 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:
SAmPling REAL<PMT>
```

上位クエリ

グループの上位のコマンドに「?」を付けたクエリを上位クエリといいます。最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;ACQUIRE:
COUNT INFINITY;MODE NORMAL;
RESOLUTION 0;RLENGTH 125000;
SAMPLING INTERPOLATE
```

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。

- ニー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?」

- ニー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：

TRIGger[:ATRigger]:SIMPlE:LEVel は、

「TRIG:SIMP:LEV」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：

「TRIGger?」と「TRIGger:ATRigger?」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:CHANnel1: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:EES #HFE)
<文字データ>	規定された文字列(ニーモニック)。{}内から選択 (例: CH1の入力カップリングの選択 -> CHANnel1:COUPling {AC DC DC50})
<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 -> FILE:SAVE:COMMeNt)
<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 -> #800000010ABCDEFGHIJ)

<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 進数を受け取るときは、<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 ¹⁸
PE	ペタ	10 ¹⁵
T	テラ	10 ¹²
G	ギガ	10 ⁹
MA	メガ	10 ⁶
K	キロ	10 ³
M	ミリ	10 ⁻³
U	マイクロ	10 ⁻⁶
N	ナノ	10 ⁻⁹
P	ピコ	10 ⁻¹²
F	フェムト	10 ⁻¹⁵
A	アト	10 ⁻¹⁸

<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 <NR1> で返されます。

<文字データ>

<文字データ> は、規定された文字 (ニーモニック) 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}	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。
- ・ ファイル名の文字数に関しては、ユーザーズマニュアル機能編をご覧ください。

<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>	#800000010ABCDEFGHIJ イトの並び>

- ・ #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。命令のほとんどはシーケンシャルコマンドです。

Note

オーバーラップコマンドは、第5章でオーバー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。

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

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。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。

```
:FILE:LOAD:SETup:EXECute "CASE1";:  
CHANnel1:VDIV?<PMT>
```

一方、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、上のプログラムメッセージを送信すると、「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をする命令をオーバーラップコマンドといいます。このように、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ことがあります。このようなときは、次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同期をとる方法

以下に示すどれかの方法で、オーバーラップコマンドと同期を取り、オーバーラップ動作を回避できます。

• *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。

```
:COMMunicate:OPSE #H0040;:FILE:LOAD:  
SETup:EXECute "CASE1";*WAI;;  
CHANnel1:VDIV?<PMT>
```

上の例で、「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」を実行しているのに、「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。

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

上の例で、「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。

```
: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>
```

上の例で、「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が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。したがって、「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、オーバーラップ動作が終了したときに応答を生成する命令です。

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

上の例で、「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?」はオーバー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。

シーケンシャルコマンド (通常のコマンド) と同期をとる方法

シーケンシャルコマンドの場合でも、トリガの発生など通信以外の要因で同期をとる必要が生じることがあります。

```
:ACQuire:COUNT 1;:TRIGger:
MODE NORMAl<PMT>
:WAVeform:SEND?<PMT>
```

たとえば、波形の取り込み回数を設定し、トリガモードをノーマルにして取り込んだ波形データを問い合わせるときに、上のプログラムメッセージを送信すると、波形の取り込みが終了している / していないにかかわらず「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。

このようなときは、以下に示すどれかの方法で取り込みが終了するタイミングを取る必要があります。

• 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。

```
:ACQuire:COUNT 1;:TRIGger:
MODE NORMAl<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>
```

上の例で、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。「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

(サービスリクエスト (SRQ) が使用できる場合)

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。

```
:STATus:FILTer1 FALL;:STATus:EESR 1;
EESR?;*SRE 8<PMT>
:ACQuire:COUNT 1;:TRIGger:
MODE NORMAl<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>
```

上の例で、「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、遷移フィルターを設定することを示しています。

「STATus:EESR 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命令です。

「STATus:EESR?」は、拡張イベントレジスタをクリアします。

「*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を発生する命令です。したがって、「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。

```
:STATus:FILTer1 FALL;:STATus:EESR?
<PMT>
:ACQuire:COUNT 1;:TRIGger:
MODE NORMAl<PMT>
(STATus:EESR? の応答を読む)
:COMMunicate:WAIT 1;:WAVeform:SEND?
<PMT>
```

上の例で、「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じです。

「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
ACQuire グループ		
: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。	5-60
: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。	5-60
:AVERage		
:COUNt	アベレーシングモード時の、指数化平均の減衰定数、または単純平均のアベレーシング回数を設定 / 問い合わせします。	5-60
:COUNt	ノーマルモード・エンベロープモード・アベレーシングモード時の波形の取り込み回数を設定 / 問い合わせします。	5-60
:MODE	波形の取り込みモード (アキュイジションモード) を設定 / 問い合わせします。	5-60
:RESolution	高分解能モードの ON/OFF を設定 / 問い合わせします。	5-60
:RLENgth	レコード長を設定 / 問い合わせします。	5-60
:SAMPLing	サンプリングモードを設定 / 問い合わせします。	5-60
ANALysis グループ		
:ANALysis?	解析機能に関するすべての設定値を問い合わせます。	5-61
:ANALysis		
:AHIStogram<x>?	解析機能に関するすべての設定値を問い合わせます。	5-61
:AHIStogram<x>		
:DISPlay	波形のヒストグラムの表示の ON/OFF を設定 / 問い合わせします。	5-61
:HORizontal	波形の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。	5-61
:MEASure?	波形のヒストグラムの測定 (モードの ON/OFF を含む) に関するすべての設定値を問い合わせます。	5-61
:MEASure		
:MODE	波形のヒストグラムの測定モードを設定 / 問い合わせします。	5-61
:PARAmeter?	モードが Param のときの波形のヒストグラムの測定に関するすべての設定値を問い合わせます。	5-61
:PARAmeter		
:ALL	波形のヒストグラムのすべての測定項目を一斉に ON/OFF します。	5-61
:<パラメータ>?	波形のヒストグラムの指定した測定項目に関するすべての設定値を問い合わせます。	5-61
:<パラメータ>		
:STATe	波形のヒストグラムの指定した測定項目の ON/OFF を設定 / 問い合わせします。	5-62
:VALue?	波形のヒストグラムの指定した測定項目の自動測定値を問い合わせます。	5-62
:HPOSition<y>	波形のヒストグラムの波形 Cursor1 または Cursor2 水平位置を設定 / 問い合わせします。	5-62
:VPOSition<y>	波形のヒストグラムの Cursor1 または Cursor2 の垂直位置を設定 / 問い合わせします。	5-62
:MODE	波形のヒストグラムの対象軸を設定 / 問い合わせします。	5-62
:RANGe	波形の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。	5-62
:TRACe	波形のヒストグラムの対象波形を設定 / 問い合わせします。	5-62
:VERTical	波形のヒストグラムの垂直範囲を設定 / 問い合わせします。	5-63
:PANalyze?	電源解析に関するすべての設定値を問い合わせます。	5-63
:PANalyze		
:HARMonics?	高調波解析機能に関するすべての設定値を問い合わせます。	5-63
:HARMonics		
:CCLass?	高調波解析のクラス C に関するすべての設定値を問い合わせます。	5-63
:CCLass		
:GETLambda	高調波解析のクラス C の現在の力率を設定します。	5-63
:LAMBda	高調波解析のクラス C の力率を設定 / 問い合わせします。	5-63
:MAXCurrent	高調波解析のクラス C の基本波の電流値を設定 / 問い合わせします。	5-63
:OPower	高調波解析のクラス C の有効電力 25W を超える / 超えないを設定 / 問い合わせします。	5-63
:CLASs	高調波解析の対象機器の適用クラスを設定 / 問い合わせします。	5-63
:DCLass?	高調波解析のクラス D に関するすべての設定値を問い合わせます。	5-63

5.1 コマンド一覧表

コマンド	機能	ページ
:DCLass		
:POWer	高調波解析のクラス D の電力値を設定 / 問い合わせします。	5-64
:DMODE	高調波解析の表示モードを設定 / 問い合わせします。	5-64
:DETail?	高調波解析の解析結果リストに関するすべての設定値を問い合わせます。	5-64
:DETail		
:DISPlay	高調波解析の解析結果リストの表示モードを設定 / 問い合わせします。	5-64
:LIST		
:ITEM?	高調波解析の解析結果リストに表示される項目を問い合わせます。	5-64
:VALue?	高調波解析の解析結果リストの指定した解析番号の全データを問い合わせます。	5-64
:GR0uping	高調波解析のグルーピングを設定 / 問い合わせします。	5-64
:IECedition	高調波解析の測定モードの測定規格 (IEC61000-4-7) を設定 / 問い合わせします。	5-64
:RMS?	高調波演算の実効値を問い合わせます。	5-65
:SPOint	高調波解析の演算開始点を設定 / 問い合わせします。	5-65
:SVOLtage	高調波解析の電源電圧を設定 / 問い合わせします。	5-65
:THD?	高調波演算の高調波ひずみ率を問い合わせます。	5-65
:I2T?	ジュール積分に関するすべての設定値を問い合わせます。	5-65
:I2T		
:MATH	ジュール積分波形の表示の ON/OFF を設定 / 問い合わせします。	5-65
:MEASure?	ジュール積分の自動測定に関するすべての設定値を問い合わせます。	5-65
:MEASure		
:I2T?	ジュール積分に関する設定を問い合わせます。	5-65
:I2T		
:COUNT?	ジュール積分の継続統計処理の回数を問い合わせます。	5-65
:{MAXimum MEAN MINimum SDEVIation}?	ジュール積分の各統計値を問い合わせます。	5-65
:STATe	ジュール積分の測定を行う (ON)/ 行わない (OFF) を設定 / 問い合わせします。	5-65
:VALue?	ジュール積分の自動測定値を問い合わせます。	5-66
:RANGE	測定対象ウィンドウを設定 / 問い合わせします。	5-66
:SCALE?	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。	5-66
:SCALE		
:CENTer	マニュアルスケーリング時の中心値を設定 / 問い合わせします。	5-66
:MODE	スケーリングの方法を設定 / 問い合わせします。	5-66
:SENSitivity	マニュアルスケーリング時の中心からのスパンを設定 / 問い合わせします。	5-66
:TRANge	測定範囲を設定 / 問い合わせします。	5-66
:SETup?	電源解析の入力に関するすべての設定値を問い合わせます。	5-66
:SETup		
:ADESkew	電源解析のオートスキュー補正を実行します。	5-66
:I?	電源解析の電流入力チャンネルに関するすべての設定値を問い合わせます。	5-67
:I		
:DESKew	電源解析の電流入力チャンネルのスキュー補正を設定 / 問い合わせします。	5-67
:INPut	電源解析の電流入力チャンネルを設定 / 問い合わせします。	5-67
:PROBe	電源解析の電流入力チャンネルのプロープの電流 - 電圧換算比を設定 / 問い合わせします。	5-67
:U?	電源解析の電圧入力チャンネルに関するすべての設定値を問い合わせます。	5-67
:U		
:DESKew	電源解析の電圧入力チャンネルのスキュー補正を設定 / 問い合わせします。	5-67
:INPut	電源解析の電圧入力チャンネルを設定 / 問い合わせします。	5-67
:PROBe	電源解析の電圧入力チャンネルのプロープの減衰比を設定 / 問い合わせします。	5-67
:RTRace	電源解析のスキュー補正の対象トレースを設定 / 問い合わせします。	5-67
:SOA?	XY 表示 (安全動作領域) に関するすべての設定値を問い合わせます。	5-67
:SOA		
:CURSor?	XY 表示 (安全動作領域) 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。	5-68
:CURSor		
:X<x>?	XY 表示 (安全動作領域) 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。	5-68
:X<x>		
:POSition	XY 表示 (安全動作領域) の水平カーソルの位置を設定 / 問い合わせします。	5-68
:VALue?	XY 表示 (安全動作領域) の水平カーソルの電圧値を問い合わせます。	5-68

5.1 コマンド一覧表

コマンド	機能	ページ
:Y<x>?	XY 表示 (安全動作領域) 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。	5-68
:Y<x>		
:POSition	XY 表示 (安全動作領域) の垂直カーソルの位置を設定 / 問い合わせします。	5-68
:VALue?	XY 表示 (安全動作領域) の垂直カーソルの電圧値を問い合わせます。	5-68
:MODE	XY 表示 (安全動作領域) の自動測定モードを設定 / 問い合わせします。	5-68
:TRANge	XY 表示 (安全動作領域) する T-Y 波形の範囲を設定 / 問い合わせします。	5-68
:VTDisplay	XY 表示 (安全動作領域) の VT 波形の表示の ON/OFF を設定 / 問い合わせします。	5-69
:SWLoss?	スイッチング損失に関するすべての設定値を問い合わせます。	5-69
:SWLoss		
:CCALc	トータル損失の計算式を設定 / 問い合わせします。	5-69
:DPROximal?	ディスタル・メシアル・プロキシマルに関するすべての設定値を問い合わせます。	5-69
:DPROximal		
:MODE	ディスタル・メシアル・プロキシマル点のモードを設定 / 問い合わせします。	5-69
:PERCent	ディスタル・メシアル・プロキシマル点を % で設定 / 問い合わせします。	5-69
:UNIT	ディスタル・メシアル・プロキシマル点を電圧値で設定 / 問い合わせします。	5-69
:ILEVl	トータル損失の損失ゼロ区間と判定する電流レベルを設定 / 問い合わせします。	5-70
:INDicator	トータル損失の測定箇所の表示を設定 / 問い合わせします。	5-70
:MATH	電力波形の表示の ON/OFF を設定 / 問い合わせします。	5-70
:MEASure?	電源解析項目 (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。	5-70
:MEASure		
:<パラメータ>?	電源解析項目 (パラメータ) に関するすべての設定値を問い合わせます。	5-70
:<パラメータ>		
:COUNT?	電源解析項目 (パラメータ) の継続統計処理の回数を問い合わせます。	5-70
:{MAXimum MEAN MINimum SDEVIation}?	電源解析項目 (パラメータ) の各統計値を問い合わせます。	5-70
:STATe	電源解析項目 (パラメータ) の ON/OFF を設定 / 問い合わせします。	5-70
:VALue?	電源解析項目 (パラメータ) の自動測定値を問い合わせます。	5-71
:CONTinuous?	トータル損失の自動測定の通常の統計処理に関するすべての設定値を問い合わせます。	5-71
:CONTinuous		
:REStart	トータル損失の自動測定の通常の統計処理を再スタートします。	5-71
:TLCHange	トータル損失の自動測定の通常の統計処理再スタートするか否かを設定 / 問い合わせします。	5-71
:CYCLe		
:ABOrt	トータル損失の自動測定のサイクル統計処理の実行を中止します。	5-71
:EXECute	トータル損失の自動測定のサイクル統計処理の実行します。	5-71
:HISTory		
:ABOrt	トータル損失の自動測定のヒストリ波形の統計処理の実行を中止します。	5-71
:EXECute	トータル損失の自動測定のヒストリ波形の統計処理の実行します。	5-71
:MODE	トータル損失の自動測定の ON/OFF / 統計処理を設定 / 問い合わせします。	5-72
:ZLINKage	トータル損失の自動測定の測定結果の結果番号とズーム位置との連動 (ON/OFF) を設定 / 問い合わせします。	5-72
:METHod	High・Low 点の算出方法を設定 / 問い合わせします。	5-72
:RANGe	測定対象ウィンドウを設定 / 問い合わせします。	5-72
:RDS	トータル損失のオン抵抗値を設定 / 問い合わせします。	5-72
:SCALE?	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。	5-72
:SCALE		
:CENTer	マニュアルスケーリング時の中心値を設定 / 問い合わせします。	5-72
:MODE	スケーリングの方法を設定 / 問い合わせします。	5-72
:SENSitivity	マニュアルスケーリング時の中心からのスパンを設定 / 問い合わせします。	5-73
:TRANge	測定範囲を設定 / 問い合わせします。	5-73
:ULEVl	トータル損失の損失計算区間と判定する電圧レベルを設定 / 問い合わせします。	5-73
:UNIT	電力量の単位を設定 / 問い合わせします。	5-73
:VCE	トータル損失のコレクタ・エミッタ間飽和電圧値を設定 / 問い合わせします。	5-73
:TYPE	電源解析の種類を設定 / 問い合わせします。	5-73
:PMEASURE<x>?	電力測定に関するすべての設定値を問い合わせます。	5-73
:PMEASURE<x>		

5.1 コマンド一覧表

コマンド	機能	ページ
:CYCMode	電力測定のカイクルモードを設定 / 問い合わせします。	5-73
:CYCTrace	電力測定のカイクルモードの周期対象波形を設定 / 問い合わせします。	5-73
:IDPRoximal?	ディスタル・メシアル・プロキシマルに関するすべての設定値を問い合わせます。	5-74
:IDPRoximal		
:MODE	ディスタル・メシアル・プロキシマル点のモードを設定 / 問い合わせします。	5-74
:PERCent	ディスタル・メシアル・プロキシマル点を%で設定 / 問い合わせします。	5-74
:UNIT	ディスタル・メシアル・プロキシマル点を電圧値で設定 / 問い合わせします。	5-74
:IMEThod	High・Low 点を設定 / 問い合わせします。	5-74
:INDicator	測定箇所の表示を設定 / 問い合わせします。	5-74
:MEASure?	電力測定項目 (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。	5-74
:MEASure		
:<パラメータ>?	電力測定項目 (パラメータ) に関するすべての設定値を問い合わせます。	5-75
:<パラメータ>		
:COUNT?	電力測定項目 (パラメータ) の通常の統計処理の回数を問い合わせます。	5-75
:{MAXimum MEAN MINimum SDEVIation}?	電力測定項目 (パラメータ) の各統計値を問い合わせます。	5-75
:STATe	電力測定項目 (パラメータ) の ON/OFF を設定 / 問い合わせします。	5-75
:VALue?	電力測定項目 (パラメータ) の自動測定値を問い合わせます。	5-75
:ALL	電力測定項目 (パラメータ) を一括 ON/OFF します。	5-75
:CONTinuous?	電力測定の自動測定の通常の統計処理に関するすべての設定値を問い合わせます。	5-75
:CONTinuous		
:REStart	電力測定の自動測定の通常の統計処理を再スタートします。	5-76
:TLCHange	電力測定の自動測定の通常の統計処理再スタートするか否かを設定 / 問い合わせします。	5-76
:CYCLe?	電力測定の自動測定のサイクル統計処理に関するすべての設定値を問い合わせます。	5-76
:CYCLe		
:ABORt	電力測定の自動測定のサイクル統計処理の実行を中止します。	5-76
:EXECute	電力測定の自動測定のサイクル統計処理を実行します。	5-76
:TRACe	電力測定の自動測定のサイクル統計処理の周期対象波形を設定 / 問い合わせします。	5-76
:HISTory		
:ABORt	電力測定の自動測定のヒストリ波形の統計処理を中止します。	5-76
:EXECute	電力測定の自動測定のヒストリ波形の統計処理を実行します。	5-76
:MODE	電力測定の自動測定の ON/OFF / 統計処理を設定 / 問い合わせします。	5-76
:ZLINKage	電力測定の自動測定の測定結果の結果番号とズーム位置との連動 (ON/OFF) を設定 / 問い合わせします。	5-77
:MODE	電力測定の ON/OFF を設定 / 問い合わせします。	5-77
:RANGe	測定対象ウィンドウを設定 / 問い合わせします。	5-77
:SETUp?	電力測定の入力に関するすべての設定値を問い合わせます。	5-77
:SETUp		
:ADESkew	電力測定のオートスキュー補正を実行します。	5-77
:I?	電力測定の電流入力チャンネルに関するすべての設定値を問い合わせます。	5-77
:I		
:DESKew	電力測定の電流入力チャンネルのスキュー補正を設定 / 問い合わせします。	5-77
:PROBe	電力測定の電流入力チャンネルのプロープの電流 - 電圧換算比を設定 / 問い合わせします。	5-77
:RTRace	電力測定のスキュー補正の対象トレースを設定 / 問い合わせします。	5-77
:U?	電力測定の電圧入力チャンネルに関するすべての設定値を問い合わせます。	5-78
:U		
:DESKew	電力測定の電圧入力チャンネルのスキュー補正を設定 / 問い合わせします。	5-78
:PROBe	電力測定の電圧入力チャンネルのプロープの減衰比を設定 / 問い合わせします。	5-78
:TRANge	測定範囲を設定 / 問い合わせします。	5-78
:UDPRoximal?	ディスタル・メシアル・プロキシマルに関するすべての設定値を問い合わせます。	5-78
:UDPRoximal		
:MODE	ディスタル・メシアル・プロキシマル点のモードを設定 / 問い合わせします。	5-78
:PERCent	ディスタル・メシアル・プロキシマル点を%で設定 / 問い合わせします。	5-78
:UNIT	ディスタル・メシアル・プロキシマル点を電圧値で設定 / 問い合わせします。	5-78

5.1 コマンド一覧表

コマンド	機能	ページ
:UMETHod	High・Low 点を設定 / 問い合わせします。	5-79
:UNIT	電力量の単位を設定 / 問い合わせします。	5-79
:USER<y>?	Calc アイテ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。	5-79
:USER<y>		
:COUNT?	Calc アイテムの自動測定値の統計処理の回数を問い合わせます。	5-79
:DEFine	Calc アイテムの自動測定値の演算式を設定 / 問い合わせします。	5-79
:{MAXimum MEAN MINimum S DEVIation}?	Calc アイテムの自動測定値の各統計値を問い合わせます。	5-79
:NAME	Calc アイテムの名称を設定 / 問い合わせします。	5-79
:STATe	Calc アイテムの自動測定の ON/OFF を設定 / 問い合わせします。	5-79
:UNIT	Calc アイテムの単位を設定 / 問い合わせします。	5-79
:VALue?	Calc アイテムの自動測定値の測定値を問い合わせます。	5-80
:WAIT?	タイムアウト付きで自動測定実行の終了を待ちます。	5-80

ASETup グループ

:ASETup		
:EXECute	オートセットアップを実行します。	5-81
:UNDO	実行したオートセットアップを取り消します。	5-81

CALibrate グループ

: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。	5-82
:CALibrate		
[:EXECute]	キャリブレーションを実行します。	5-82
: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。	5-82

CHANnel グループ

:CHANnel<x>?	各チャンネルの垂直軸に関するすべての設定値を問い合わせます。	5-83
:CHANnel<x>		
:AScale		
[:EXECute]	各チャンネルのオートスケールを実行します。	5-83
:BWIDth	各チャンネルの入力フィルターを設定 / 問い合わせします。	5-83
: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。	5-83
: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。	5-83
:DISPlay	各チャンネルの表示の ON/OFF を設定 / 問い合わせします。	5-83
:INVert	インバートモード (波形の反転表示) の ON/OFF を設定 / 問い合わせします。	5-83
: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。	5-83
: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。	5-83
:DISPlay	各チャンネルの波形ラベル名の表示の ON/OFF を設定 / 問い合わせします。	5-84
:LSCale?	各チャンネルのリニアスケールに関するすべての設定値を問い合わせます。	5-84
:LSCale		
:AVALue	スケール係数 A を設定 / 問い合わせします。	5-84
:BVALue	オフセット値 B を設定 / 問い合わせします。	5-84
:MODE	リニアスケールリングの ON/OFF を設定 / 問い合わせします。	5-84
:UNIT	リニアスケールリング結果に付加する単位を設定 / 問い合わせします。	5-84
: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。	5-84
: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。	5-84
:PROBe?	各チャンネルのプロープの減衰比に関するすべての設定値を問い合わせます。	5-84
:PROBe		
:DZCalibrate	各チャンネルの消磁とゼロ補正を実行します。	5-84
[:MODE]	各チャンネルのプロープの減衰比を設定 / 問い合わせします。	5-85
:PZCalibrate	各チャンネルの電圧プロープのゼロ補正を実行します。	5-85
:VARiable	各チャンネルの電圧軸感度を V/div の 0.01 ステップで設定 / 問い合わせします。	5-85
:VDIV	各チャンネルの電圧軸感度 (V/div) を設定 / 問い合わせします。	5-85

5.1 コマンド一覧表

コマンド	機能	ページ
CHUTil グループ		
:CHUTil?	チャンネル間の設定コピーに関するすべての設定値を問い合わせます。	5-86
:CHUTil		
:ALL		
:DISPlay	全チャンネルの表示の ON/OFF を設定します。	5-86
:COPYch?	チャンネル間の設定コピーに関するすべての設定値を問い合わせます。	5-86
:COPYch		
:EXECute	チャンネル間の設定コピーを実行します。	5-86
:SOURce	コピー対象のチャンネルを設定 / 問い合わせします。	5-86
:DESTination?	コピー先のチャンネルに関するすべての設定値を問い合わせます。	5-86
:DESTination		
:ALL	コピー先のチャンネルを全チャンネルとする (ON)、しない (OFF) を設定します。	5-86
:CHANnel<x>	チャンネル間の設定コピーのコピー先に設定する (ON)、しない (OFF) を設定 / 問い合わせします。	5-86
:UNDO	チャンネル間の設定コピーの実行を取り消します。	5-86
CLEar グループ		
:CLEar	クリアトレースを実行します。	5-87
COMMunicate グループ		
:COMMunicate?	通信に関するすべての設定値を問い合わせます。	5-88
: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:PROBE:MODE 10)、付けないで返送するか (例 10) を設定 / 問い合わせします。	5-88
:LOCKout	ローカルロックアウトを設定 / 解除します。	5-88
:OPSE	*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。	5-88
:OPSR?	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。	5-88
: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。	5-88
: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。	5-88
:VERBose	クエリに対する応答を、フルスベルで返送するか (例 CHANNEL1:PROBE:MODE 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。	5-89
: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。	5-89
: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。	5-89
CURSor グループ		
: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。	5-90
:CURSor		
[:TY]?	カーソルに関するすべての設定値を問い合わせます。	5-90
[:TY]		
:DEGRee?	角度カーソルに関するすべての設定値を問い合わせます。	5-90
:DEGRee		
:ALL	T-Y 表示の角度カーソルの測定値を一括 ON/OFF します	5-90
:D<x>?	角度カーソルの角度に関するすべての設定値を問い合わせます。	5-90
:D<x>		
:STATe	角度カーソルの角度の ON/OFF を設定 / 問い合わせします。	5-90
:VALue?	角度カーソルの角度を問い合わせます。	5-90
:DD?	角度カーソル間の角度差に関するすべての設定値を問い合わせます。	5-90
:DD		
:STATe	角度カーソル間の角度差 ΔD 値の ON/OFF を設定 / 問い合わせします。	5-90
:VALue?	角度カーソル間の角度差 ΔD 値を問い合わせます。	5-90
:DV?	角度カーソル間の ΔV に関するすべての設定値を問い合わせます。	5-90
:DV		
:STATe	角度カーソル間の ΔV 値の ON/OFF を設定 / 問い合わせします。	5-90
:VALue?	角度カーソル間の ΔV 値を問い合わせます。	5-91
:JUMP	T-Y 表示の角度カーソルを指定したズーム波形上にジャンプさせます。	5-91
:POSition<x>	角度カーソルの位置を設定 / 問い合わせします。	5-91

5.1 コマンド一覧表

コマンド	機能	ページ
:REfErEnce<x>	基準角度の始点 (Reference1) または終点 (Reference2) の位置を設定 / 問い合わせします。	5-91
:RVALue	基準角度を設定 / 問い合わせします。	5-91
:TRACe	角度カーソルの対象波形を設定 / 問い合わせします。	5-91
:UNIT	角度カーソルの単位を設定 / 問い合わせします。	5-91
:V<x>?	角度カーソルの電圧に関するすべての設定値を問い合わせます。	5-91
:V<x>		
:STATe	角度カーソルの電圧の ON/OFF を設定 / 問い合わせします。	5-91
:VALue?	角度カーソルの電圧値を問い合わせます。	5-92
:HORizontal?	ΔV カーソルに関するすべての設定値を問い合わせます。	5-92
:HORizontal		
:ALL	T-Y 表示の ΔV カーソルの測定値を一括 ON/OFF します。	5-92
:DV?	ΔV カーソル間の垂直軸値に関するすべての設定値を問い合わせます。	5-92
:DV		
:STATe	ΔV カーソル間の垂直軸値の ON/OFF を設定 / 問い合わせします。	5-92
:VALue?	ΔV カーソル間の垂直軸値を問い合わせます。	5-92
:DVDT?	ΔV カーソルの ΔV / ΔT に関するすべての設定値を問い合わせます。	5-92
:DVDT		
:STATe	ΔV カーソルの ΔV / ΔT 値の ON/OFF を設定 / 問い合わせします。	5-92
:VALue?	ΔV カーソルの ΔV / ΔT 値を問い合わせます。	5-92
:POSition<x>	ΔV カーソルの位置を設定 / 問い合わせします。	5-92
:TRACe	ΔV カーソルの対象波形を設定 / 問い合わせします。	5-92
:V<x>?	ΔV カーソルの垂直軸に関するすべての設定値を問い合わせます。	5-92
:V<x>		
:STATe	ΔV カーソルの垂直軸値の ON/OFF を設定 / 問い合わせします。	5-93
:VALue?	ΔV カーソルの垂直軸値を問い合わせます。	5-93
: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。	5-93
:MARKer		
:FORM	マーカーカーソルの表示形式を設定 / 問い合わせします。	5-93
:M<x>?	指定した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。	5-93
:M<x>		
:ALL	T-Y 表示のマーカーカーソルの測定値を一括 ON/OFF します。	5-93
:DT<y>?	マーカーカーソル間の時間軸に関するすべての設定値を問い合わせます。	5-93
:DT<y>		
:STATe	マーカーカーソル間の時間軸値の ON/OFF を設定 / 問い合わせします。	5-93
:VALue?	マーカーカーソル間の時間軸値を問い合わせます。	5-93
:DV<y>?	マーカーカーソル間の垂直軸に関するすべての設定値を問い合わせます。	5-93
:DV<y>		
:STATe	マーカーカーソル間の垂直軸値の ON/OFF を設定 / 問い合わせします。	5-93
:VALue?	マーカーカーソル間の垂直軸値を問い合わせます。	5-94
:JUMP	T-Y 表示のマーカーカーソルを指定したズーム波形上にジャンプさせます。	5-94
:POSition	マーカーカーソルの時間軸値を設定 / 問い合わせします。	5-94
:T?	マーカーカーソルの時間軸に関するすべての設定値を問い合わせます。	5-94
:T		
:STATe	マーカーカーソルの時間軸値の ON/OFF を設定 / 問い合わせします。	5-94
:VALue?	マーカーカーソルの時間軸値を問い合わせます。	5-94
:TRACe	マーカーカーソルの対象波形を設定 / 問い合わせします。	5-94
:V?	マーカーカーソルの垂直軸に関するすべての設定値を問い合わせます。	5-94
:V		
:STATe	マーカーカーソルの垂直軸値の ON/OFF を設定 / 問い合わせします。	5-94
:VALue?	マーカーカーソルの垂直軸値を問い合わせます。	5-94
:TYPE	カーソルの種類を設定 / 問い合わせします。	5-95
:VERTical?	ΔT カーソルに関するすべての設定値を問い合わせます。	5-95
:VERTical		
:ALL	T-Y 表示の ΔT カーソルの測定値を一括 ON/OFF します。	5-95
:DT?	ΔT カーソル間の時間軸に関するすべての設定値を問い合わせます。	5-95
:DT		
:STATe	ΔT カーソル間の時間軸値の ON/OFF を設定 / 問い合わせします。	5-95
:VALue?	ΔT カーソル間の時間軸値を問い合わせます。	5-95
:DV?	ΔT カーソル間の垂直軸に関するすべての設定値を問い合わせます。	5-95
:DV		
:STATe	ΔT カーソル間の垂直軸値の ON/OFF を設定 / 問い合わせします。	5-95

5.1 コマンド一覧表

コマンド	機能	ページ
:VALue?	ΔT カーソル間の垂直軸値を問い合わせます。	5-95
:DVDT?	ΔT カーソルの $\Delta V / \Delta T$ に関するすべての設定値を問い合わせます。	5-95
:DVDT		
:STATe	ΔT カーソルの $\Delta V / \Delta T$ 値の ON/OFF を設定 / 問い合わせします。	5-95
:VALue?	ΔT カーソルの $\Delta V / \Delta T$ 値を問い合わせます。	5-95
:JUMP	T-Y 表示の垂直カーソルを指定したズーム波形上にジャンプさせます。	5-96
:PERDt?	ΔT カーソル間の $1 / \Delta T$ に関するすべての設定値を問い合わせます。	5-96
:PERDt		
:STATe	ΔT カーソル間の $1 / \Delta T$ 値の ON/OFF を設定 / 問い合わせします。	5-96
:VALue?	ΔT カーソル間の $1 / \Delta T$ 値を問い合わせます。	5-96
:POSition<x>	ΔT カーソルの位置を設定 / 問い合わせします。	5-96
:T<x>?	ΔT カーソルの時間軸に関するすべての設定値を問い合わせます。	5-96
:T<x>		
:STATe	ΔT カーソルの時間軸値の ON/OFF を設定 / 問い合わせします。	5-96
:VALue?	ΔT カーソルの時間軸値を問い合わせます。	5-96
:TRACe	ΔT カーソルの対象波形を設定 / 問い合わせします。	5-96
:V<x>?	ΔT カーソルの垂直軸に関するすべての設定値を問い合わせます。	5-96
:V<x>		
:STATe	ΔT カーソルの垂直軸値の ON/OFF を設定 / 問い合わせします。	5-96
:VALue?	ΔT カーソルの垂直軸値を問い合わせます。	5-97

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。	5-98
:DISPlay		
:ACCumulate?	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。	5-98
:ACCumulate		
:MODE	アキュムレートモードを設定 / 問い合わせします。	5-98
:PERsistence	アキュムレート時間を設定 / 問い合わせします。	5-98
: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。	5-98
:COLor		
:{CHANnel<x> MATH<x>}	各々の波形の色を設定 / 問い合わせします。	5-98
:{LOGic LSTate}	ロジック波形の色とステート表示の色を設定 / 問い合わせします。	5-98
:SERialbus<x>?	シリアルバスごとの全トレンドの色設定を問い合わせます。	5-98
:SERialbus<x>		
:TREND<y>	シリアルバストrendの色を設定 / 問い合わせします。	5-99
:FORMat	表示フォーマットを設定 / 問い合わせします。	5-99
:FGRIId	ファイングリッド表示の ON/OFF を設定 / 問い合わせします。	5-99
:GRATicule	グラティクル (目盛り) を設定 / 問い合わせします。	5-99
: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。	5-99
:INTENSity		
[:WAVeform]	波形の輝度を設定 / 問い合わせします。	5-99
:{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。	5-99
:INTERpolate	補間方式を設定 / 問い合わせします。	5-99
:MAPPING	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。	5-99
:SMAPPING?	分割フォーマットへの全波形の割り当てを問い合わせます。	5-99
:SMAPPING		
:{CHANnel<x> MATH<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。	5-100
:SVALue	スケール値表示の ON/OFF を設定 / 問い合わせします。	5-100

FFT グループ

:FFT<x>?	FFT 解析機能に関するすべての設定値を問い合わせます。	5-101
:FFT<x>		
:AVERage?	FFT 解析のアベレージングに関するすべての設定値を問い合わせます。	5-101
:AVERage		
:EWEight	FFT 解析の指数化平均の減衰定数を設定 / 問い合わせします。	5-101
:DATA?	送信する FFT 波形データのすべての情報を問い合わせます。	5-101
:DATA		
:BYTeorder	FFT 波形データのフォーマットがバイナリのときの送信順序を設定 / 問い合わせします。	5-101
:END	送信する FFT 波形データの終了点を設定 / 問い合わせします。	5-101
:FORMat	送信する FFT 波形データのフォーマットを設定 / 問い合わせします。	5-101

5.1 コマンド一覧表

コマンド	機能	ページ
:LENGth?	送信する FFT 波形の全データ点数を問い合わせます。	5-101
:SEND?	FFT 波形データを問い合わせます。	5-101
:START	送信する FFT 波形データの開始点を設定 / 問い合わせします。	5-102
:DISPlay	FFT 解析をする (ON)/ しない (OFF) を設定 / 問い合わせします。	5-102
:HORizontal?	FFT 解析の横軸に関するすべての設定値を問い合わせます。	5-102
:HORizontal		
:CSPan?	FFT 解析の横軸の中心点 / スパンに関するすべての設定値を問い合わせます。	5-102
:CSPan		
:CENTer	FFT 解析の横軸の中心値を設定 / 問い合わせします。	5-102
:SPAN	FFT 解析の横軸のスパンを設定 / 問い合わせします。	5-102
:LRIGht?	FFT 解析の横軸の左右端に関するすべての設定値を問い合わせます。	5-102
:LRIGht		
:RANGe	FFT 解析の横軸の左右端の範囲を設定 / 問い合わせします。	5-102
:MODE	FFT 解析の横軸のモードを設定 / 問い合わせします。	5-102
:LENGth	FFT 解析の FFT 点数を設定 / 問い合わせします。	5-102
:MEASure?	FFT 解析の自動測定に関するすべての設定値を問い合わせます。	5-103
:MEASure		
:MARKer?	FFT 解析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。	5-103
:MARKer		
[:BASic]?	FFT 解析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。	5-103
[:BASic]		
:ALL	FFT 解析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。	5-103
:DFRequency?	FFT 解析のマーカーカーソル間の周波数に関するすべての設定値を問い合わせます。	5-103
:DFRequency		
:STATe	FFT 解析のマーカーカーソル間の周波数の ON/OFF を設定 / 問い合わせします。	5-103
:VALue?	FFT 解析のマーカーカーソル間の周波数を問い合わせます。	5-103
:DV?	FFT 解析のマーカーカーソル間のレベルに関するすべての設定値を問い合わせます。	5-103
:DV		
:STATe	FFT 解析のマーカーカーソル間のレベルの ON/OFF を設定 / 問い合わせします。	5-103
:VALue?	FFT 解析のマーカーカーソル間のレベルを問い合わせます。	5-103
:FREQuency<y>?	FFT 解析の各マーカーカーソルの周波数に関するすべての設定値を問い合わせます。	5-103
:FREQuency<y>		
:STATe	FFT 解析の各マーカーカーソルの周波数の ON/OFF を設定 / 問い合わせします。	5-104
:VALue?	FFT 解析の各マーカーカーソルの周波数を問い合わせます。	5-104
:POSition<y>	FFT 解析の各マーカーカーソルの位置を設定 / 問い合わせします。	5-104
:V<y>?	FFT 解析の各マーカーカーソルのレベルに関するすべての設定値を問い合わせます。	5-104
:V<y>		
:STATe	FFT 解析の各マーカーカーソルのレベルの ON/OFF を設定 / 問い合わせします。	5-104
:VALue?	FFT 解析の各マーカーカーソルのレベルを問い合わせます。	5-104
:MODE	FFT 解析の自動測定モードを設定 / 問い合わせします。	5-104
:PARAmeter?	FFT 解析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。	5-104
:PARAmeter		
:OVERall?	FFT 解析のパラメータのオーバーオールに関するすべての設定値を問い合わせます。	5-104
:OVERall		
:STATe	FFT 解析のパラメータのオーバーオール値の ON/OFF を設定 / 問い合わせします。	5-105
:VALue?	FFT 解析のパラメータのオーバーオール値を問い合わせます。	5-105
:PEAK?	FFT 解析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。	5-105
:PEAK		
:DETail?	FFT 解析結果リストに関するすべての設定値を問い合わせます。	5-105
:DETail		

5.1 コマンド一覧表

コマンド	機能	ページ
:LIST		
:ITEM?	FFT 解析結果リストに表示される項目を問い合わせます。	5-105
:MAXPeak		
:EXECute	FFT 解析結果の最大ピーク値に移動します。	5-105
:NUMBer	FFT 解析結果リストの解析番号を設定 / 問い合わせします。	5-105
:VALue?	FFT 解析結果リストの指定した解析番号の全データを問い合わせます。	5-105
:EXCursion	FFT 解析のピーク値の山谷差を設定 / 問い合わせします。	5-105
:FVALue?	FFT 解析のピーク周波数を問い合わせます。	5-106
:RANGe	FFT 解析のピーク値の測定範囲を設定 / 問い合わせします。	5-106
:THReshold	FFT 解析のピーク値のしきい値を設定 / 問い合わせします。	5-106
:VVALue?	FFT 解析のピーク値を問い合わせます。	5-106
:MODE	FFT 解析の波形の表示方式を設定 / 問い合わせします。	5-106
:RANGe	FFT 解析の測定対象ウィンドウを設定 / 問い合わせします。	5-106
:RPOSition	FFT 解析の縦軸の拡大中心点を設定 / 問い合わせします。	5-106
:TRACe	FFT 解析の対象波形を設定 / 問い合わせします。	5-106
:TYPE	FFT 解析のスペクトラムを設定 / 問い合わせします。	5-107
:UNIT?	FFT 演算単位に関するすべての設定値を問い合わせます。	5-107
:UNIT		
:DBM?	FFT 演算単位が dBm のときに関するすべての設定値を問い合わせます。	5-107
:DBM		
:IMPedance	FFT 演算単位が dBm のときのインピーダンスを設定 / 問い合わせします。	5-107
:MODE	FFT 演算単位の自動 / 手動付加を設定 / 問い合わせします。	5-107
:VERTical?	FFT 解析の縦軸に関するすべての設定値を問い合わせます。	5-107
:VERTical		
:LEVel	FFT 解析の縦軸の表示位置を設定 / 問い合わせします。	5-107
:MODE	FFT 解析の縦軸のモードを設定 / 問い合わせします。	5-107
:SENSitivity	FFT 解析の縦軸感度を設定 / 問い合わせします。	5-107
:VTDisplay	FFT 解析の VT 波形の表示の ON/OFF を設定 / 問い合わせします。	5-108
:WINDow	FFT 解析の窓関数を設定 / 問い合わせします。	5-108

FILE グループ

:FILE?	データストレージに関するすべての設定値を問い合わせます。	5-109
:FILE		
:COPY		
:ABORt	ファイルのコピーを中止します。	5-109
:CDIRectory	ファイルのコピー先のディレクトリを変更します。	5-109
:DRIVe	ファイルのコピー先のメディアを設定します。	5-109
[:EXECute]	ファイルのコピーを実行します。オーバーラップコマンドです。	5-109
:PATH?	ファイルのコピー先のディレクトリを問い合わせます。	5-109
:DELeTe		
{:AHIStoGram ASCIi BINary BMP FFT HLISt JPEG MEASure PNG SBUS SETup SNAP ZPOLygon ZWAVe}		
[:EXECute]	各種データのファイルの削除を実行します。オーバーラップコマンドです。	5-109
[:DIRectory]		
:CDIRectory	カレントのディレクトリを変更します。	5-109
:DRIVe	カレントのメディアを設定します。	5-109
:FREE?	カレントのメディアの空き容量をバイト数で問い合わせます。	5-109
:MDIRectory	カレント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。	5-109
:PATH?	カレントのディレクトリを問い合わせます。	5-110
:LOAD		
:BINary		
:ABORt	波形データの呼び出しを中止します。	5-110
[:EXECute]	波形データの呼び出しを実行します。オーバーラップコマンドです。	5-110
{:SETup SNAP ZPOLygon<x> ZWAVE<x>}		
:ABORt	各種データの呼び出しを中止します。	5-110
[:EXECute]	各種データの呼び出しを実行します。オーバーラップコマンドです。	5-110
:MOVE		
:ABORt	ファイルの移動を中止します。	5-110
:CDIRectory	ファイルの移動先のディレクトリを変更します。	5-110

5.1 コマンド一覧表

コマンド	機能	ページ
:DRive	ファイルの移動先のメディアを設定します。	5-110
[:EXECute]	ファイルの移動を実行します。オーバーラップコマンドです。	5-110
:PATH?	ファイルの移動先のディレクトリを問い合わせます。	5-110
:PROTeCt		
[:EXECute]	ファイルのプロテクトの ON/OFF を設定します。	5-110
:REName		
[:EXECute]	ファイルのリネームを設定します。	5-110
:SAVE?	ファイルの保存に関するすべての設定値を問い合わせます。	5-110
:SAVE		
: {AHIStoGram ASCIi BINary FFT HLISt HARMonics 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。	5-110
: {AHIStoGram ASCIi BINary FFT HLISt HARMonics MEASure SBUS SETup SNAP ZWAVe}		
:ABORt	各種データの保存を中止します。	5-111
[:EXECute]	各種データのファイルへの保存を実行します。オーバーラップコマンドです。	5-111
: {AHIStoGram FFT HARMonics SBUS ZWAVe}		
:SElect	各種データの保存するエリアを設定 / 問い合わせます。	5-111
:ANAMing	保存するファイル名の自動作成の ON/OFF を設定 / 問い合わせします。	5-111
: {ASCIi BINary}		
:COMPreSSion	各種データの圧縮保存方法を設定 / 問い合わせます。	5-111
:HISTory	各種ヒストリ波形の保存方法を設定 / 問い合わせします。	5-111
:LENGth	各種のデータを圧縮保存または間引き保存するときのデータ点数を設定 / 問い合わせします。	5-111
:RANGe	各種データで保存するウィンドウを設定 / 問い合わせします。	5-111
:TRACe	各種データで保存する波形を設定 / 問い合わせします。	5-112
:ASCIi		
:TINformation	保存する波形データの時刻情報の有り (ON)/ 無し (OFF) を設定 / 問い合わせをします。	5-112
:COMMeNt	保存するデータのコメントを設定 / 問い合わせします。	5-112
:FFT		
:FINformation	保存する FFT データの周波数情報の有り (ON)/ 無し (OFF) を設定 / 問い合わせをします。	5-112
:NAME	保存するデータのファイル名を設定 / 問い合わせします。	5-112

GONogo グループ

: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。	5-113
:GONogo		
:ABORt	GO/NO-GO 判定を中止します。	5-113
: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。	5-113
:ACTioN		
:BUZZer	判定が NO-GO 時に警告音を鳴らすか鳴らさないかを設定 / 問い合わせします。	5-113
:HCOPy	判定が NO-GO 時に内蔵プリンタ (オプション) または外部プリンタへハードコピーをとるかとらないかを設定 / 問い合わせします。	5-113
:MAIL?	判定が NO-GO 時のメール通知に関するすべての設定値を問い合わせます。	5-113
:MAIL		
:COUNT	判定が NO-GO 時のメールで通知時にメール件数の上限を設定 / 問い合わせします。	5-113
:MODE	判定が NO-GO 時にメールで通知するかしないかを設定 / 問い合わせします。	5-113
:SAVE	判定が NO-GO 時にメディアに保存するかしないかを設定 / 問い合わせします。	5-113
:COUNT?	実際に行った GO/NO-GO の判定回数を問い合わせます。	5-113
: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。	5-113
:LOGic	GO/NO-GO 判定の種類を設定 / 問い合わせします。	5-113
:NGCount?	GO/NO-GO 判定の NO-GO 回数を問い合わせます。	5-113
:NGStopcount	判定終了 NO-GO 回数を設定 / 問い合わせします。	5-114
:STOPcount	判定終了取り込み回数を設定 / 問い合わせします。	5-114
:WAIT?	GO/NO-GO 判定の終了をタイムアウト付きで待ちます。	5-114
[: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。	5-114

5.1 コマンド一覧表

コマンド	機能	ページ
[:ZPARameter]		
:NUMBer<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。	5-114
:NUMBer<x>		
:CAUSE?	ゾーン / パラメータ判定の各波形パラメータが NO-GO の原因かそうでないかを問い合わせます。	5-114
:CONDition	ゾーン / パラメータ判定の各波形パラメータの判定基準を設定 / 問い合わせします。	5-114
:MODE	各条件のモードを設定 / 問い合わせします。	5-114
: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。	5-114
:PARameter		
:ITEM	パラメータ判定の各波形パラメータのアイテムを設定 / 問い合わせします。	5-115
:LIMit	パラメータ判定の各波形パラメータの上下限值を設定 / 問い合わせします。	5-115
:TRACe	パラメータ判定の各波形パラメータの対象波形を設定 / 問い合わせします。	5-115
:VALue?	パラメータ判定の各波形パラメータの測定値を問い合わせます。	5-115
:POLYgon?	ポリゴンゾーン判定に関するすべての設定値を問い合わせます。	5-115
:POLYgon		
:HPOSition	ポリゴンゾーン判定で用いる水平位置を設定 / 問い合わせします。	5-116
:RANGe	ポリゴンゾーン判定で用いる対象ウィンドウを設定 / 問い合わせします。	5-116
:TRACe	ポリゴンゾーン判定で用いる対象波形を設定 / 問い合わせします。	5-116
:VPOSition	ポリゴンゾーン判定で用いる垂直位置を設定 / 問い合わせします。	5-116
:ZNUMber	ポリゴンゾーン判定で用いるゾーン番号を設定 / 問い合わせします。	5-116
:RECTangle?	方形ゾーン判定に関するすべての設定値を問い合わせます。	5-116
:RECTangle		
:HORizontal	方形ゾーン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。	5-117
:RANGe	方形ゾーン判定で用いる四角形の対象ウィンドウを設定 / 問い合わせします。	5-117
:TRACe	方形ゾーン判定で用いる四角形の対象波形を設定 / 問い合わせします。	5-117
:VERTical	方形ゾーン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。	5-117
:WAVE?	波形ゾーン判定に関するすべての設定値を問い合わせます。	5-117
:WAVE		
:EDIT		
:EXIT	波形ゾーンの編集メニューから抜けます。	5-117
:NEW	波形ゾーンの編集で、基本波形を設定します。	5-117
:PART	波形ゾーンの編集で、部分編集を実行します。	5-118
:WHOLe	波形ゾーンの編集で、波形全体のゾーンを設定します。	5-118
:RANGe	波形ゾーン判定で用いる対象ウィンドウを設定 / 問い合わせします。	5-118
:TRACe	波形ゾーン判定で用いる対象波形を設定 / 問い合わせします。	5-118
:TRANge	波形ゾーン判定で用いる判定区間を設定 / 問い合わせします。	5-118
:ZNUMber	波形ゾーン判定で用いるゾーン番号を設定 / 問い合わせします。	5-118

HCOPy グループ

:HCOPy?	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。	5-119
:HCOPy		
:ABORt	データ出力、紙送りを中止します。	5-119
:DIRection	データ出力先を設定 / 問い合わせします。	5-119
:EXECute	データ出力を実行します。	5-119
:EXTPrinter?	外部プリンタへの出力に関するすべての設定値を問い合わせます。	5-119
:EXTPrinter		
:MODE	外部プリンタへ出力するときのノーマルコピー / ハードコピーを設定 / 問い合わせします。	5-119
:TONE	外部プリンタ出力のカラーを設定 / 問い合わせします。	5-119
:MULTitarget?	マルチターゲットに関するすべての設定値を問い合わせます。	5-119
:MULTitarget		
:EXTPrinter	マルチターゲットの外部プリンタへの出力を設定 / 問い合わせします。	5-119
:PRINter	マルチターゲットの内部プリンタへの出力を設定 / 問い合わせします。	5-119
:NETPrinter	マルチターゲットのネットワークプリンタへの出力を設定 / 問い合わせします。	5-119
:FILE	マルチターゲットのファイルへの出力を設定 / 問い合わせします。	5-119
:WAVEform	マルチターゲットの波形ファイルへの出力を設定 / 問い合わせします。	5-120
:NETPrint?	ネットワークプリンタへの出力に関するすべての設定値を問い合わせます。	5-120
:NETPrint		

コマンド	機能	ページ
:MODE	ネットワークプリンタへ出力するときのノーマルコピー/ハードコピーを設定/問い合わせします。	5-120
:TONE	ネットワークプリンタ出力のカラーを設定/問い合わせします。	5-120
:TYPE	ネットワークプリンタへの出力コマンドの種類を設定/問い合わせします。	5-120
:PRINter?	内蔵プリンタへの出力に関するすべての設定値を問い合わせます。	5-120
:MODE	内蔵プリンタへ出力するときのショートコピー/ハードコピーを設定/問い合わせします。	5-120
:REPort	内蔵プリンタへ付加情報を出力する/しないを設定/問い合わせします。	5-120

HISTory グループ

:HISTory?	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。	5-121
:HISTory		
:AVERAge	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。	5-121
:DISPlay	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。	5-121
:DMODE	ヒストリ波形の表示モードを設定/問い合わせします。	5-121
: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。	5-121
:REPLay?	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。	5-121
: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。	5-121
: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。	5-121
:START	ヒストリ波形のリプレイを指定方向へ開始します。	5-121
:STOP	ヒストリ波形のリプレイを停止します。	5-121
:SEARCh?	ヒストリ波形の検索に関するすべての設定値を問い合わせます。	5-121
[:SEARCh]		
:ABORt	検索を中止します。	5-122
:EXECute	検索を実行します。オーバーラップコマンドです。	5-122
:LOGic	ヒストリ波形の検索の論理を設定/問い合わせします。	5-122
:NUMBer<x>?	各検索条件に関するすべての設定値を問い合わせます。	5-122
:NUMBer<x>		
:CONDition	各検索条件の判定基準を設定/問い合わせします。	5-122
:MODE	各検索条件のモードを設定/問い合わせします。	5-122
:PARAmeter?	パラメータ検索に関するすべての設定値を問い合わせます。	5-122
:PARAmeter		
:ITEM	パラメータ検索の各波形パラメータのアイテムを設定/問い合わせします。	5-122
:LIMit	パラメータ検索の各波形パラメータの上下限值を設定/問い合わせします。	5-122
:TRACe	パラメータ検索の各波形パラメータの対象波形を設定/問い合わせします。	5-123
:VALue?	パラメータ検索の各波形パラメータの測定値を問い合わせます。	5-123
:POLYgon?	ポリゴンゾーン検索に関するすべての設定値を問い合わせます。	5-123
:POLYgon		
:HPOSition	ポリゴンゾーン検索で用いる水平位置を設定/問い合わせします。	5-123
:RANGe	ポリゴンゾーン検索で用いる対象ウィンドウを設定/問い合わせします。	5-123
:TRACe	ポリゴンゾーン検索で用いる対象波形を設定/問い合わせします。	5-123
:VPOSition	ポリゴンゾーン検索で用いる垂直位置を設定/問い合わせします。	5-124
:ZNUMber	ポリゴンゾーン検索で用いるゾーン番号を設定/問い合わせします。	5-124
:RECTangle?	方形ゾーン検索に関するすべての設定値を問い合わせます。	5-124
:RECTangle		
:HORizontal	方形ゾーン検索で用いる四角形の水平位置を設定/問い合わせします。	5-124
:RANGe	方形ゾーン検索で用いる四角形の対象ウィンドウを設定/問い合わせします。	5-124
:TRACe	方形ゾーン検索で用いる四角形の対象波形を設定/問い合わせします。	5-124
:VERTical	方形ゾーン検索で用いる四角形の垂直位置を設定/問い合わせします。	5-125
:WAVE?	波形ゾーン検索に関するすべての設定値を問い合わせます。	5-125
:WAVE		
:RANGe	波形ゾーン検索で用いる対象ウィンドウを設定/問い合わせします。	5-125
:TRACe	波形ゾーン検索で用いる対象波形を設定/問い合わせします。	5-125
:TRANge	波形ゾーン検索で用いる判定区間を設定/問い合わせします。	5-125
:ZNUMber	波形ゾーン検索で用いるゾーン番号を設定/問い合わせします。	5-125
:RESet	ヒストリ波形の検索条件をリセットします。	5-125
:SIMPlE?	簡易検索に関するすべての設定値を問い合わせます。	5-125
:SIMPlE		
:HORizontal	簡易検索で用いる四角形の水平位置を設定/問い合わせします。	5-125
:RANGe	簡易検索で用いる四角形の対象ウィンドウを設定/問い合わせします。	5-126

5.1 コマンド一覧表

コマンド	機能	ページ
:TRACe	簡易検索で用いる四角形の対象トレースを設定 / 問い合わせします。	5-126
:VERTical	簡易検索で用いる四角形の垂直位置を設定 / 問い合わせします。	5-126
:TIME?	対象レコード番号の時間を問い合わせます。	5-126

IMAGe グループ

:IMAGe?	画面イメージ出力に関するすべての設定値を問い合わせます。	5-127
:IMAGe		
:ABORt	画面イメージをメディアに保存するのを中止します。	5-127
:BACKground	画面イメージのバックグラウンドを設定 / 問い合わせします。	5-127
:EXECute	画面イメージをメディアに保存します。	5-127
:FORMat	画面イメージの出力形式を設定 / 問い合わせします。	5-127
:INFormation	画面イメージに設定情報を付加する / しないを設定 / 問い合わせします。	5-127
:MODE	画面イメージの出力方式を設定 / 問い合わせします。	5-127
:SAVE?	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。	5-127
:SAVE		
:ANAMing	出力ファイル名の自動作成の ON/OFF を設定 / 問い合わせします。	5-127
:CDIRectory	ファイルのディレクトリを変更します。	5-127
:DRIVe	作成するファイルのメディアを設定します。	5-127
:NAME	作成するファイルのファイル名を設定 / 問い合わせします。	5-127
:SEND?	画面イメージの値を問い合わせます。	5-128
:TONE	出力する画面イメージの色調を設定 / 問い合わせします。	5-128
:WGRadation	出力する画面イメージの波形部分の階調の有無を設定 / 問い合わせします。	5-128

INITialize グループ

:INITialize		
:EXECute	イニシャライズを実行します。	5-129
:UNDO	実行したイニシャライズを取り消します。	5-129

LOGic グループ

:LOGic?	ロジック入力波形に関するすべての設定値を問い合わせます。	5-130
:LOGic		
:BITOrder	ロジック入力のビットオーダーを設定 / 問い合わせします。	5-130
:BUS<x>?	ロジック入力のポートの各バスのすべての設定値を問い合わせます。	5-130
:BUS<x>		
:ASSignment	ロジック入力のポートの各バスのアサインを設定 / 問い合わせします。	5-130
:DISPlay	ロジック入力のポート各バスの表示の ON/OFF を設定 / 問い合わせします。	5-130
:FORMat	ロジック入力のポートの各バスの表示形式 (バス表示) を設定 / 問い合わせします。	5-130
:LABel	ロジック入力のポートのラベル名を設定 / 問い合わせします。	5-131
:MODE	ロジック入力の ON/OFF を設定 / 問い合わせします。	5-131
:PODA?	ロジック入力のポート A のすべての設定値を問い合わせます。	5-131
:PODA		
:ALL?	ロジック入力のポート A の全ビットのすべての設定値を問い合わせます。	5-131
:ALL		
:DISPlay	ロジック入力のポート A の全ビットの表示の ON/OFF を設定 / 問い合わせします。	5-131
:LEVel	ロジック入力のポート A のユーザー定義でのスレシヨルドレベルを設定 / 問い合わせします。	5-131
:TYPE	ロジック入力のポート A のスレシヨルドレベルの選択を実行します。	5-131
:BIT<x>?	ロジック入力のポート A の各ビットのすべての設定値を問い合わせます。	5-131
:BIT<x>		
:DISPlay	ロジック入力のポート A の各ビットの表示の ON/OFF を設定 / 問い合わせします。	5-131
:LABel	ロジック入力のポート A の各ビットのラベル名を設定 / 問い合わせします。	5-131
:LEVel	ロジック入力のポート A の各ビットのユーザー定義でのスレシヨルドレベルを設定 / 問い合わせします。	5-131
:TYPE	ロジック入力のポート A の各ビットのスレシヨルドレベルを設定 / 問い合わせします。	5-132
:DESKew	ロジック入力のポート A のスキュー補正を設定 / 問い合わせします。	5-132
:HYSTeresis	ロジック入力のポート A のヒステリシスを設定 / 問い合わせします。	5-132
:PODB?	ロジック入力のポート B のすべての設定値を問い合わせます。	5-132

コマンド	機能	ページ
:PODB		
:ALL?	ロジック入力ポート B の全ビットのすべての設定値を問い合わせます。	5-132
:ALL		
:DISPlay	ロジック入力ポート B の全ビットの表示の ON/OFF を設定 / 問い合わせします。	5-132
:LEVel	ロジック入力ポート B のユーザー定義でのスレシヨルドレベルを設定 / 問い合わせします。	5-132
:TYPE	ロジック入力ポート B のスレシヨルドレベルの選択を実行します。	5-132
:BIT<x>?	ロジック入力ポート B の各ビットのすべての設定値を問い合わせます。	5-132
:BIT<x>		
:DISPlay	ロジック入力ポート B の各ビットの表示の ON/OFF を設定 / 問い合わせします。	5-132
:LABel	ロジック入力ポート B の各ビットのラベル名を設定 / 問い合わせします。	5-132
:LEVel	ロジック入力ポート B の各ビットのユーザー定義でのスレシヨルドレベルを設定 / 問い合わせします。	5-133
:TYPE	ロジック入力ポート B の各ビットのスレシヨルドレベルを設定 / 問い合わせします。	5-133
:DESKew	ロジック入力ポート B のスキュー補正を設定 / 問い合わせします。	5-133
:HYSTeresis	ロジック入力ポート B のヒステリシスを設定 / 問い合わせします。	5-133
:PODC?	ロジック入力ポート C のすべての設定値を問い合わせます。	5-133
:PODC		
:ALL?	ロジック入力ポート C の全ビットのすべての設定値を問い合わせます。	5-133
:ALL		
:DISPlay	ロジック入力ポート C の全ビットの表示の ON/OFF を設定 / 問い合わせします。	5-133
:LEVel	ロジック入力ポート C のユーザー定義でのスレシヨルドレベルを設定 / 問い合わせします。	5-133
:TYPE	ロジック入力ポート C のスレシヨルドレベルの選択を実行します。	5-133
:BIT<x>?	ロジック入力ポート C の各ビットのすべての設定値を問い合わせます。	5-133
:BIT<x>		
:DISPlay	ロジック入力ポート C の各ビットの表示の ON/OFF を設定 / 問い合わせします。	5-133
:LABel	ロジック入力ポート C の各ビットのラベル名を設定 / 問い合わせします。	5-134
:LEVel	ロジック入力ポート C の各ビットのユーザー定義でのスレシヨルドレベルを設定 / 問い合わせします。	5-134
:TYPE	ロジック入力ポート C の各ビットのスレシヨルドレベルを設定 / 問い合わせします。	5-134
:DESKew	ロジック入力ポート C のスキュー補正を設定 / 問い合わせします。	5-134
:HYSTeresis	ロジック入力ポート C のヒステリシスを設定 / 問い合わせします。	5-134
:PODD?	ロジック入力ポート D のすべての設定値を問い合わせます。	5-134
:PODD		
:ALL?	ロジック入力ポート D の全ビットのすべての設定値を問い合わせます。	5-134
:ALL		
:DISPlay	ロジック入力ポート D の全ビットの表示の ON/OFF を設定 / 問い合わせします。	5-134
:LEVel	ロジック入力ポート D のユーザー定義でのスレシヨルドレベルを設定 / 問い合わせします。	5-134
:TYPE	ロジック入力ポート D のスレシヨルドレベルの選択を実行します。	5-135
:BIT<x>?	ロジック入力ポート D の各ビットのすべての設定値を問い合わせます。	5-135
:BIT<x>		
:DISPlay	ロジック入力ポート D の各ビットの表示の ON/OFF を設定 / 問い合わせします。	5-135
:LABel	ロジック入力ポート D の各ビットのラベル名を設定 / 問い合わせします。	5-135
:LEVel	ロジック入力ポート D の各ビットのユーザー定義でのスレシヨルドレベルを設定 / 問い合わせします。	5-135
:TYPE	ロジック入力ポート D の各ビットのスレシヨルドレベルを設定 / 問い合わせします。	5-135
:DESKew	ロジック入力ポート D のスキュー補正を設定 / 問い合わせします。	5-135
:HYSTeresis	ロジック入力ポート D のヒステリシスを設定 / 問い合わせします。	5-135
: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。	5-136
:SIZE	ロジック信号の表示サイズを設定 / 問い合わせします。	5-136
:STATe?	ロジック入力の状態表示に関するすべての設定値を問い合わせます。	5-136
:STATe		

5.1 コマンド一覧表

コマンド	機能	ページ
:ASSignment?	ロジック入力の状態表示のアサインに関するすべての設定値を問い合わせます。	5-136
:ASSignment		
:ALL	ロジック入力の状態表示のアサインのすべての設定値を設定 / 問い合わせします。	5-136
:BUS<x>	ロジック入力の状態表示のアサインのバスの設定値を設定 / 問い合わせします。	5-136
: { PODA<x> PODB<x> PODC<x> PODD<x> }	ロジック入力の状態表示のアサインの各ビットのすべての設定値を設定 / 問い合わせします。	5-136
:CLOCK	ロジック入力の状態表示の基準クロック波形を設定 / 問い合わせします。	5-136
:MODE	ロジック入力の状態表示の ON/OFF を設定 / 問い合わせします。	5-136
:POLarity	ロジック入力の状態表示の基準クロック波形の極性を設定 / 問い合わせします。	5-136

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。	5-137
:MATH<x>		
:DISPlay	演算波形を表示する (ON)/ 表示しない (OFF) を設定 / 問い合わせします。	5-137
:ECOUNT?	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。	5-137
:ECOUNT		
:HYSTeresis	エッジカウント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。	5-137
:POLarity	エッジカウントのエッジ検出極性を設定 / 問い合わせします。	5-137
:THReshold	エッジ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。	5-137
:FILTer?	フィルターに関するすべての設定値を問い合わせます。	5-137
:FILTer		
:FORDer	IIR フィルターのフィルター次数を設定 / 問い合わせします。	5-137
:HCUToff	IIR ハイパスフィルターのカットオフ周波数を設定 / 問い合わせします。	5-137
:LCUToff	IIR ローパスフィルターのカットオフ周波数を設定 / 問い合わせします。	5-138
:TIME	位相シフトの遅延時間を設定 / 問い合わせします。	5-138
:TYPE	フィルターのタイプを設定 / 問い合わせします。	5-138
:WEIGHt	移動平均の重みを設定 / 問い合わせします。	5-138
:INTegral?	積分に関するすべての設定値を問い合わせます。	5-138
:INTegral		
:SPOint	積分の開始位置を設定 / 問い合わせします。	5-138
: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。	5-138
:LABel		
[:DEFine]	演算波形のラベル名を設定 / 問い合わせします。	5-138
: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。	5-138
:OPERation	演算子を設定 / 問い合わせします。	5-139
:RCOUNT?	ロータリ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。	5-139
:RCOUNT		
:COUNT?	ロータリカウント演算のカウント条件に関するすべての設定値を問い合わせます。	5-139
:COUNT		
:CWTiming<y>	ロータリカウント演算の CW(clockwise) のカウントタイミングを設定 / 問い合わせします。	5-139
:MULTi	ロータリカウント演算の通倍を設定 / 問い合わせします。	5-139
:HYSTeresis<y>	ロータリカウント演算のヒステリシスを設定 / 問い合わせします。	5-139
:RESet?	ロータリカウント演算のリセット条件に関するすべての設定値を問い合わせます。	5-139
:RESet		
:CWTiming	ロータリカウント演算の CW(clockwise) のリセットタイミングを設定 / 問い合わせします。	5-139
:RTIMing	ロータリカウント演算のリセットタイミングを設定 / 問い合わせします。	5-139
:THReshold<y>	ロータリカウント演算の判定レベルを設定 / 問い合わせします。	5-140
:ZPOLarity	ロータリカウント演算の極性を設定 / 問い合わせします。	5-140
:SCALE?	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。	5-140
:SCALE		
:CENTer	マニュアルスケーリング時の中心値を設定 / 問い合わせします。	5-140
:MODE	スケーリングの方法を設定 / 問い合わせします。	5-140

5.1 コマンド一覧表

コマンド	機能	ページ
:SENSitivity	マニュアルスケーリング時の中心からのスパンを設定 / 問い合わせします。	5-140
:UNIT?	演算単位に関するすべての設定値を問い合わせます。	5-140
:UNIT		
[[:DEFine]	演算単位を設定 / 問い合わせします。	5-140
:MODE	演算単位の自動 / 手動付加を設定 / 問い合わせします。	5-140
: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ます。	5-140
:USERdefine		
:AVERage?	ユーザー定義演算のアベレージングに関するすべての設定値を問い合わせます。	5-140
:AVERage		
:EWEight	ユーザー定義演算の指数化平均の減衰定数を設定 / 問い合わせします。	5-141
:MODE	ユーザー定義演算のアベレージングモードを設定 / 問い合わせします。	5-141
:CONSitant<y>	ユーザー定義演算の定数を設定 / 問い合わせします。	5-141
:DEFine	ユーザー定義演算の演算式を設定 / 問い合わせします。	5-141
:FILTer<y>?	ユーザー定義演算のフィルターに関するすべての設定値を問い合わせます。	5-141
:FILTer<y>		
:BAND	ユーザー定義演算のフィルターのバンド (周波数帯) を設定 / 問い合わせします。	5-141
:CUTOff<z>	ユーザー定義演算のカットオフ周波数を設定 / 問い合わせします。	5-141
:TYPE	ユーザー定義演算のフィルターのタイプを設定 / 問い合わせします。	5-142
:HISTory		
:ABORt	ヒストリ波形のユーザー定義演算 (Math on History) を取り消します。	5-142
:EXECute	ヒストリ波形のユーザー定義演算 (Math on History) を実行します。	5-142
:SCALe?	ユーザー定義演算のスケール変換に関するすべての設定値を問い合わせます。	5-142
:SCALe		
:ARANGing	ユーザー定義演算のオートレンジを実行します。	5-142
:CENTer	ユーザー定義演算のスケール変換の中心値を設定 / 問い合わせします。	5-142
:SENSitivity	ユーザー定義演算のスケール変換の中心からのスパンを設定 / 問い合わせします。	5-142

MEASure グループ

: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。	5-143
:MEASure		
: {CHANnel<x> MATH<x>} ?	各波形の各パラメータの ON/OFF をすべて問い合わせます。	5-143
: {CHANnel<x> MATH<x>}		
: {AREA1 AREA2} ?	各 Area の各波形の各パラメータの ON/OFF をすべて問い合わせます。	5-143
: {[:AREA1] :AREA2}		
:ALL	各 Area の各波形のすべての測定アイテムの ON/OFF を一度に設定します。	5-143
:<パラメータ>?	各 Area の各波形の各波形パラメータに関する設定を問い合わせます。	5-143
:<パラメータ>		
:COUNT?	各 Area の各波形パラメータの統計処理の回数を問い合わせます。	5-143
: {MAXimum MEAN MINimum SDEVIation} ?	各 Area の各波形パラメータの各統計値を問い合わせます。	5-143
:STATe	各 Area の各波形パラメータの ON/OFF を 1 つずつ設定 / 問い合わせします。	5-144
:VALue?	各 Area の各波形パラメータの自動測定値を問い合わせます。	5-144
:COPY	各 Area のある波形の全測定アイテムの ON/OFF 設定を他のすべての波形にコピーします。	5-145
:CYCLe	各 Area の周期モードを設定 / 問い合わせします。	5-145
:DELay?	各 Area の各波形のチャンネル間ディレイに関するすべての設定値を問い合わせます。	5-145
:DELay		
:MEASure?	各 Area の各波形のチャンネル間ディレイの対象波形に関するすべての設定値を問い合わせます。	5-145
:MEASure		
:COUNT	各 Area の各波形のチャンネル間ディレイの対象波形のカウント数を設定 / 問い合わせします。	5-145
:SLOPe	各 Area の各波形のチャンネル間ディレイの対象波形のスロープを設定 / 問い合わせします。	5-145
:REFERence?	各 Area の各波形のチャンネル間ディレイの基準波形に関するすべての設定値を問い合わせます。	5-145
:REFERence		

5.1 コマンド一覧表

コマンド	機能	ページ
:COUNT	各 Area の各波形のチャンネル間ディレイの基準波形のカウント数を設定 / 問い合わせします。	5-146
:SLOPe	各 Area の各波形のチャンネル間ディレイの基準波形のスロープを設定 / 問い合わせします。	5-146
:SOURce	各 Area の各波形のチャンネル間ディレイの基準対象をトリガ点にするか、波形にするかを設定 / 問い合わせをします。	5-146
:TRACe	各 Area の各波形のチャンネル間ディレイの基準波形のエッジを設定 / 問い合わせします。	5-146
:DPRoximal?	ディスタル・メシアル・プロキシマルに関するすべての設定値を問い合わせます。	5-146
:DPRoximal		
:MODE	ディスタル・メシアル・プロキシマル点のモードを設定 / 問い合わせします。	5-147
:PERCent	ディスタル・メシアル・プロキシマル点を % で設定 / 問い合わせします。	5-147
:UNIT	ディスタル・メシアル・プロキシマル点を電圧値で設定 / 問い合わせします。	5-147
:METHod	High・Low 点を設定 / 問い合わせします。	5-147
:CONTInuous?	波形パラメータの自動測定の通常の統計処理に関するすべての設定を問い合わせます。	5-147
:CONTInuous		
:REStArt	波形パラメータの自動測定の通常の統計処理を再スタートします。	5-147
:TLCHange	トリガレベル変更時に、波形パラメータの自動測定の通常の統計処理を再スタートするか否かを設定 / 問い合わせします。	5-148
:CYCLe?	自動測定のサイクル統計処理に関するすべての設定値を問い合わせます。	5-148
:CYCLe		
:ABORt	自動測定のサイクル統計処理の実行を中止します。	5-148
:EXECute	自動測定のサイクル統計処理を実行します。	5-148
:TRACe	自動測定のサイクル統計処理の周期対象波形を設定 / 問い合わせします。	5-148
:HISTory		
:ABORt	自動測定のヒストリ波形の統計処理の実行を中止します。	5-148
:EXECute	自動測定のヒストリ波形の統計処理を実行します。	5-148
:INDicator	測定箇所の表示を設定 / 問い合わせします。	5-148
:MODE	自動測定の ON/OFF / 統計処理を設定 / 問い合わせします。	5-148
: {PODA<x> POdB<x> PODC<x> PODD<x>} ?	ロジック波形の各パラメータの ON/OFF をすべて問い合わせます。	5-148
: {PODA<x> POdB<x> PODC<x> PODD<x>}		
: {AREA1 AREA2} ? { [:AREA1] :AREA2 }	各 Area のロジック波形の各パラメータの ON/OFF をすべて問い合わせます。	5-148
:ALL	各 Area のロジック波形のすべての測定アイテムの ON/OFF を一度に設定します。	5-149
:<パラメータ>?	各 Area のロジック波形パラメータに関する設定を問い合わせます。	5-149
:<パラメータ>		
:COUNT?	各 Area のロジック波形パラメータの統計処理の回数を問い合わせます。	5-149
: {MAXimum MEAN MINimum SDEVIation} ?	各 Area のロジック波形パラメータの各統計値を問い合わせます。	5-149
:STATe	各 Area のロジック波形パラメータの ON/OFF を 1 つずつ設定 / 問い合わせします。	5-149
:VALue?	各 Area の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。	5-149
:COPY	各 Area のあるロジック波形の全測定アイテムの ON/OFF 設定を他のすべての波形にコピーします。	5-150
:DELay?	各 Area のロジック波形のチャンネル間ディレイに関するすべての設定値を問い合わせます。	5-150
:DELay		
:MEASure?	各 Area のロジック波形のチャンネル間ディレイに関するすべての設定値を問い合わせます。	5-150
:MEASure		
:COUNT	各 Area のロジック波形のチャンネル間ディレイの対象波形のカウント数を設定 / 問い合わせします。	5-150
:SLOPe	各 Area のロジック波形のチャンネル間ディレイの対象波形のスロープを設定 / 問い合わせします。	5-150
:REFerence?	各 Area のロジック波形のチャンネル間ディレイの基準波形に関するすべての設定値を問い合わせます。	5-150
:REFerence		

5.1 コマンド一覧表

コマンド	機能	ページ
:COUNT	各 Area のロジック波形のチャンネル間ディレイの基準波形のカウント数を設定 / 問い合わせします。	5-150
:SLOPe	各 Area のロジック波形のチャンネル間ディレイの基準波形のスロープを設定 / 問い合わせします。	5-151
:SOURce	各 Area のロジック波形のチャンネル間ディレイの基準対象をトリガ点にするか、波形にするかを設定 / 問い合わせします。	5-151
:TRACe	各 Area のロジック波形のチャンネル間ディレイの基準波形のエッジを設定 / 問い合わせします。	5-151
:State	各 Area のロジック波形のチャンネル間ディレイの ON/OFF を設定 / 問い合わせします。	5-151
:RANGe<x>	測定対象ウィンドウを設定 / 問い合わせします。	5-151
:TRANge<x>	測定範囲を設定 / 問い合わせします。	5-152
:USER<x>?	Calc アイテ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。	5-152
:USER<x>		
:COUNT?	Calc アイテムの自動測定値の統計処理の回数を問い合わせます。	5-152
:DEFine	Calc アイテムの自動測定値の演算式を設定 / 問い合わせします。	5-152
:{MAXimum MEAN MINimum SDEViation}?	Calc アイテムの自動測定値の各統計値を問い合わせます。	5-152
:NAME	Calc アイテムの名称を設定 / 問い合わせします。	5-152
:STATe	Calc アイテムの自動測定の ON/OFF を設定 / 問い合わせします。	5-152
:UNIT	Calc アイテムの単位を設定 / 問い合わせします。	5-152
:VALue?	Calc アイテムの自動測定値の測定値を問い合わせます。	5-152
:WAIT?	タイムアウト付きで、自動測定実行の終了を待ちます。	5-153
:ZLINKage	波形パラメータの自動測定の測定結果の測定番号とズーム位置との連動 (ON/OFF) を設定 / 問い合わせします。	5-153

RECall グループ

:RECall		
:SETup<x>		
:EXECute	内部メモリに保存されている設定データを呼び出します。	5-154

REference グループ

:REference<x>?	リファレンス波形に関するすべての設定値を問い合わせます。	5-155
:REference<x>		
:DISPlay	リファレンス波形の表示の ON/OFF を設定 / 問い合わせします。	5-155
:LABel?	各リファレンスのラベル名に関するすべての設定値を問い合わせます。	5-155
:LABel		
[:DEFine]	各リファレンスのラベル名を設定 / 問い合わせします。	5-155
:MODE	各リファレンスのラベル名表示の ON/OFF を設定 / 問い合わせします。	5-155
:LOAD	リファレンス波形をロードします。	5-155
:POSition	リファレンス波形の垂直ポジションを設定 / 問い合わせします。	5-155

SEARch グループ

:SEARch?	波形の検索に関するすべての設定値を問い合わせます。	5-156
:SEARch		
:ABORt	検索を中止します。	5-156
:ASCRoll<x>?	オートスクロールに関するすべての設定値を問い合わせます。	5-156
:ASCRoll<x>		
: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。	5-156
:SPEed	オートスクロールのズームボックスの移動速度を設定 / 問い合わせします。	5-156
:STARt	オートスクロールを開始します。	5-156
:STOP	オートスクロールを停止します。	5-156
:EDGE?	エッジ検索に関するすべての設定値を問い合わせます。	5-156
:EDGE		
:HYSTeresis	エッジ検索レベルのヒステリシスを設定 / 問い合わせします。	5-156
:LEVeL	エッジ検索レベルを設定 / 問い合わせします。	5-156
:SLOPe	エッジ検索のスロープを設定 / 問い合わせします。	5-156
:SOURce	エッジ検索の対象波形を設定 / 問い合わせします。	5-157
:EPOint	検索終了位置を設定 / 問い合わせします。	5-157
:EXECute	検索を実行します。オーバーラップコマンドです。	5-157
:MARK	検索点マークの ON/OFF を設定 / 問い合わせします。	5-157

5.1 コマンド一覧表

コマンド	機能	ページ
:MAG<x>	ズーム画面のズーム率を設定 / 問い合わせします。	5-157
:PATtern?	複数入力のリ組み合わせ (パターン) 検索に関するすべての設定値を問い合わせます。	5-157
:PATtern		
:{CHANnel<x> MATH<x>}?	複数入力のリ組み合わせ (パターン) 検索の各波形に関するすべての設定値を問い合わせます。	5-157
:{CHANnel<x> MATH<x>}		
:HYSTeresis	複数入力のリ組み合わせ (パターン) 検索の対象波形のヒステリシスを設定 / 問い合わせします。	5-157
:LEVel	複数入力のリ組み合わせ (パターン) 検索の対象波形の検索レベルを設定 / 問い合わせします。	5-157
:PATtern	複数入力のリ組み合わせ (パターン) 検索の対象波形の検索パターンを設定 / 問い合わせします。	5-158
:CLOCK?	複数入力のリ組み合わせ (パターン) 検索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。	5-158
:CLOCK		
:HYSTeresis	複数入力のリ組み合わせ (パターン) 検索のクロックチャンネルのヒステリシスを設定 / 問い合わせします。	5-158
:LEVel	複数入力のリ組み合わせ (パターン) 検索のクロックチャンネルの検索レベルを設定 / 問い合わせします。	5-158
:SLOPe	複数入力のリ組み合わせ (パターン) 検索のクロックチャンネルのスロープを設定 / 問い合わせします。	5-158
:SOURce	複数入力のリ組み合わせ (パターン) 検索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。	5-158
:CONDition	複数入力のリ組み合わせ (パターン) 検索の成立条件を設定 / 問い合わせします。	5-158
:LOGic	複数入力のリ組み合わせ (パターン) 検索のロジックを設定 / 問い合わせします。	5-158
:TIME<x>	複数入力のリ組み合わせ (パターン) 検索の判定時間を設定 / 問い合わせします。	5-159
:TQQualify	複数入力のリ組み合わせ (パターン) の時間条件を設定 / 問い合わせします。	5-159
:{PODA PODB PODC PODD}?	複数入力のリ組み合わせ (パターン) のロジックに関するすべての設定値を問い合わせます。	5-159
:{PODA PODB PODC PODD}		
:HEXa	複数入力のリ組み合わせ (パターン) のロジックの成立条件を HEXA で設定します。	5-159
:PATtern	複数入力のリ組み合わせ (パターン) のロジックの成立条件を BINARY で設定 / 問い合わせします。	5-159
:POSition<x>	ズームボックスの位置を設定 / 問い合わせします。	5-159
:PULSe?	パルス幅検索に関するすべての設定値を問い合わせます。	5-159
:PULSe		
:HYSTeresis	パルス幅検索レベルのヒステリシスを設定 / 問い合わせします。	5-159
:LEVel	パルス幅検索レベルを設定 / 問い合わせします。	5-160
:POLarity	パルス幅検索レベルの極性を設定 / 問い合わせします。	5-160
:SOURce	パルス幅検索の対象波形を設定 / 問い合わせします。	5-160
:TIME<x>	パルス幅検索のパルス幅を設定 / 問い合わせします。	5-160
:TQQualify	パルス幅検索の検索種類を設定 / 問い合わせします。	5-160
:SElect?	ズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。	5-160
:SKIP?	スキップモードに関するすべての設定値を問い合わせます。	5-160
:SKIP		
:DECimation	スキップモードの間引き検出の設定 / 問い合わせます。	5-160
:HOLDoff	ホールドオフ検出の設定 / 問い合わせます。	5-160
:MODE	スキップモードのモードの設定 / 問い合わせます。	5-161
:SPoint	検索開始位置を設定 / 問い合わせします。	5-161
:TIMEout?	タイムアウト検索のすべての設定値を問い合わせます。	5-161
:TIMEout		
:POLarity	タイムアウト検索の極性を設定 / 問い合わせします。	5-161
:SOURce	タイムアウト検索の対象波形を設定 / 問い合わせします。	5-161
:TIME	タイムアウト検索のタイムアウト時間を設定 / 問い合わせします。	5-161
:TWINdow	検索された部分を表示するズーム画面を設定 / 問い合わせします。	5-161
:TYPE	検索タイプを設定 / 問い合わせします。	5-161

コマンド	機能	ページ
SERialbus グループ		
:SERialbus<x>?	シリアルバス信号の解析 / 検索に関するすべての設定値を問い合わせます。	5-162
:SERialbus<x>		
:ASETup?	シリアルバス信号のオート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。	5-162
:ASETup		
:ABORt	シリアルバス信号のオートセットアップを中止します。	5-162
:EXECute	シリアルバス信号のオートセットアップを実行します。	5-162
:SWAVeform	シリアルバス信号のオートセットアップの対象波形を設定 / 問い合わせします。	5-162
:TSETting	シリアルバス信号のオートセットアップのトリガ設定の有無を設定 / 問い合わせします。	5-162
:CAN?	CAN バス信号に関するすべての設定値を問い合わせます。	5-162
:CAN		
:ANALyze?	CAN バス信号解析に関するすべての設定値を問い合わせます。	5-163
[:ANALyze]		
:SETup?	CAN バス信号解析のバス設定に関するすべての設定値を問い合わせます。	5-163
:SETup		
:BRATe	CAN バス信号解析のビットレート (データ転送速度) を設定 / 問い合わせします。	5-163
:RECCessive	CAN バス信号解析のリセッショ電位を設定 / 問い合わせします。	5-163
:SOURce	CAN バス信号解析のソースを設定 / 問い合わせします。	5-163
:SPOint	CAN バス信号解析のサンプルポイントを設定 / 問い合わせします。	5-163
:DETail?	CAN バス信号の解析結果リストに関するすべての設定値を問い合わせます。	5-163
:DETail		
:DISPlay	CAN バス信号の解析結果リストの表示モードを設定 / 問い合わせします。	5-163
:FILTer?	CAN バス信号の解析結果リストのフィルタ機能に関するすべての設定値を問い合わせます。	5-164
:FILTer		
:IDENtifier?	CAN バス信号の解析結果リストのフィルタの ID に関するすべての設定値を問い合わせます。	5-164
:IDENtifier		
:HEXa	CAN バス信号の解析結果リストのフィルタ条件の ID を 16 進数で設定します。	5-164
:MFORmat	CAN バス信号の解析結果リストのフィルタの ID のフレームフォーマット (標準 / 拡張) を設定 / 問い合わせします。	5-164
:MODE	CAN バス信号の解析結果リストのフィルタの ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-164
:LIST		
:ITEM?	CAN バス信号の解析結果リストに表示される項目を問い合わせます。	5-164
:VALue?	CAN バス信号の解析結果リストの指定した解析番号の全データを問い合わせます。	5-164
:TREFeRence	CAN バス信号の解析結果リストの時間基準を設定 / 問い合わせします。	5-164
:SEARch?	CAN バス信号の検索に関するすべての設定値を問い合わせます。	5-165
:SEARch		
:ABORt	CAN バス信号の検索を中止します。	5-165
:EXECute	CAN バス信号の検索を実行します。	5-165
:FJUMp		
:ACK	CAN バス信号検索の結果の ACK Field へのフィールドジャンプを実行します。	5-165
:CONTRol	CAN バス信号検索の結果の Control Field へのフィールドジャンプを実行します。	5-165
:CRC	CAN バス信号検索の結果の CRC Field へのフィールドジャンプを実行します。	5-165
:DATA	CAN バス信号検索の結果の Data Field へのフィールドジャンプを実行します。	5-165
:IDENtifier	CAN バス信号検索の結果の Identifier へのフィールドジャンプを実行します。	5-165
:SOF	CAN バス信号検索の結果の SOF へのフィールドジャンプを実行します。	5-165
:SELect	CAN バス信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。	5-165
:SELect? MAXimum	CAN バス信号検索の検索点の総数を問い合わせます。	5-165
:SETup?	CAN バス信号検索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。	5-165
:SETup		

5.1 コマンド一覧表

コマンド	機能	ページ
:EFrame?	CAN バス信号検索の Error に関するすべての設定値を問い合わせます。	5-166
:EFrame		
[:MODE]	CAN バス信号検索の Error Frame を設定 / 問い合わせします。	5-166
:CRC	CAN バス信号検索の CRC Error を設定 / 問い合わせします。	5-166
:STUFF	CAN バス信号検索の STUFF Error を設定 / 問い合わせします。	5-166
:IDData?	CAN バス信号検索の ID とデータ条件に関するすべての設定値を問い合わせます。	5-166
[:IDData]		
:ACK?	CAN バス信号検索の ID とデータ条件の ACK に関するすべての設定値を問い合わせます。	5-166
:ACK		
:MODE	CAN バス信号検索の ID とデータ条件の ACK モードの設定 / 問い合わせします。	5-166
:TYPE	CAN バス信号検索の ID とデータ条件の ACK 条件の設定 / 問い合わせします。	5-166
:DATA?	CAN バス信号検索の ID とデータ条件のデータに関するすべての設定値を問い合わせます。	5-167
:DATA		
:CONditiOn	CAN バス信号検索の ID とデータ条件のデータ判定条件の設定 / 問い合わせします。	5-167
:DECimal<y>	CAN バス信号検索の ID とデータ条件の判定データを 10 進数で設定 / 問い合わせします。	5-167
:DLC	CAN バス信号検索の ID とデータ条件の有効バイト数 (DLC) の設定 / 問い合わせします。	5-167
:DPATtern?	CAN バス信号検索の ID とデータ条件の判定データに関するすべての設定値を問い合わせます。	5-167
:DPATtern		
:HEXa	CAN バス信号検索の ID とデータ条件の判定データを 16 進数で設定します。	5-167
:PATtern	CAN バス信号検索の ID とデータ条件のデータを 2 進数で設定 / 問い合わせします。	5-168
:ENDian	CAN バス信号検索の ID とデータ条件の判定データのエンディアンを設定 / 問い合わせします。	5-168
:MODE	CAN バス信号検索の ID とデータ条件のデータ条件 (有効 / 無効) の設定 / 問い合わせします。	5-168
:MSBLsb	CAN バス信号検索の ID とデータ条件のデータの MSB/LSB のビットを設定 / 問い合わせします。	5-168
:SIGN	CAN バス信号検索の ID とデータ条件のデータの符号を設定 / 問い合わせします。	5-168
:IDENtifier?	CAN バス信号検索の ID とデータ条件の識別子に関するすべての設定値を問い合わせます。	5-168
:IDENtifier		
:HEXa	CAN バス信号検索の ID とデータ条件の ID を 16 進数で設定します。	5-168
:MFORmat	CAN バス信号検索の ID とデータ条件の ID のフレームフォーマット (標準 / 拡張) を設定 / 問い合わせします。	5-169
:MODE	CAN バス信号検索の ID とデータ条件の識別子条件 (有効 / 無効) の設定 / 問い合わせします。	5-169
:PATtern	CAN バス信号検索の ID とデータ条件の ID 条件を 2 進数で設定 / 問い合わせします。	5-169
:PFORmat	CAN バス信号検索の ID とデータ条件の ID の入力形式を設定 / 問い合わせします。	5-169
:MSIGNal?	CAN バス信号検索の ID とデータ条件のメッセージシグナルに関するすべての設定値を問い合わせます。	5-169
:MSIGNal		
:MESSage		
:ITEM	CAN バス信号検索の ID とデータ条件のメッセージのアイテムを設定します。	5-169
:SElect	CAN バス信号検索の ID とデータ条件のメッセージシグナルの条件を設定 / 問い合わせします。	5-169
:SIGNal?	CAN バス信号検索の ID とデータ条件のシグナルに関するすべての設定値を問い合わせます。	5-170
:SIGNal		
:CONditiOn	CAN バス信号検索の ID とデータ条件のシグナルのデータ条件を設定 / 問い合わせします。	5-170
:DECimal<y>	CAN バス信号検索の ID とデータ条件のシグナルの判定データを 10 進数で設定 / 問い合わせします。	5-170
:ITEM	CAN バス信号検索の ID とデータ条件のシグナルのアイテムを設定します。	5-170

5.1 コマンド一覧表

コマンド	機能	ページ
:RTR	CAN バス信号検索の ID とデータ条件の RTR を設定 / 問い合わせします。	5-170
:MODE	CAN バス信号検索の検索の種類を設定 / 問い合わせします。	5-170
:CANFD?	CAN FD バス信号に関するすべての設定値を問い合わせます。	5-170
:CANFD		
:ANALyze?	CAN FD バス信号解析に関するすべての設定値を問い合わせます。	5-171
[:ANALyze]		
:SETup?	CAN FD バス信号解析のバス設定に関するすべての設定値を問い合わせます。	5-171
:SETup		
:BRATe	CAN FD バス信号解析のビットレート (データ転送速度) を設定 / 問い合わせします。	5-171
:DBRate	CAN FD バス信号解析のデータビットレート (データフェーズのデータ転送速度) を設定 / 問い合わせします。	5-171
:DSPoint	CAN FD バス信号解析のデータフェーズのサンプルポイントを設定 / 問い合わせします。	5-171
:FDStandard	解析する CAN FD バス信号が、ISO 標準か否かを設定 / 問い合わせします。	5-171
:RECESSive	CAN FD バス信号解析のリセッシブ電位を設定 / 問い合わせします。	5-171
:SOURce	CAN FD バス信号解析のソースを設定 / 問い合わせします。	5-172
:SPoint	CAN FD バス信号解析のサンプルポイントを設定 / 問い合わせします。	5-172
:DETail?	CAN FD バス信号の解析結果リストに関するすべての設定値を問い合わせます。	5-172
:DETail		
:DISPlay	CAN FD バス信号の解析結果リストの表示モードを設定 / 問い合わせします。	5-172
:FILTer?	CAN FD バス信号の解析結果リストのフィルタ機能に関するすべての設定値を問い合わせます。	5-172
:FILTer		
:IDENtifier?	CAN FD バス信号の解析結果リストのフィルタの ID に関するすべての設定値を問い合わせます。	5-172
:IDENtifier		
:HEXa	CAN FD バス信号の解析結果リストのフィルタ条件の ID を 16 進数で設定します。	5-172
:MFORmat	CAN FD バス信号の解析結果リストのフィルタの ID のフレームフォーマット (標準 / 拡張) を設定 / 問い合わせします。	5-172
:MODE	CAN FD バス信号の解析結果リストのフィルタの ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-173
:LIST		
:ITEM?	CAN FD バス信号の解析結果リストに表示される項目を問い合わせます。	5-173
:VALue?	CAN FD バス信号の解析結果リストの指定した解析番号の全データを問い合わせます。	5-173
:TREference	CAN FD バス信号の解析結果リストの時間基準を設定 / 問い合わせします。	5-173
:SEARch?	CAN FD バス信号の検索に関するすべての設定値を問い合わせます。	5-173
:SEARch		
:ABORt	CAN FD バス信号の検索を中止します。	5-173
:EXECute	CAN FD バス信号の検索を実行します。	5-173
:FJUMp		
:ACK	CAN FD バス信号検索の結果の ACKField へのフィールドジャンプを実行します。	5-173
:CONTRol	CAN FD バス信号検索の結果の Control Field へのフィールドジャンプを実行します。	5-173
:CRC	CAN FD バス信号検索の結果の CRC Field へのフィールドジャンプを実行します。	5-174
:DATA	CAN FD バス信号検索の結果の Data Field へのフィールドジャンプを実行します。	5-174
:IDENtifier	CAN FD バス信号検索の結果の Identifier へのフィールドジャンプを実行します。	5-174
:SOF	CAN FD バス信号検索の結果の SOF へのフィールドジャンプを実行します。	5-174
:SELEct	CAN FD バス信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。	5-174
:SELEct? MAXimum	CAN FD バス信号検索の検索点の総数を問い合わせます。	5-174
:SETup?	CAN FD バス信号検索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。	5-174
:SETup		
:EFRAme?	CAN FD バス信号検索の Error に関するすべての設定値を問い合わせます。	5-174

5.1 コマンド一覧表

コマンド	機能	ページ
:EFrame		
:CRC	CAN FD バス信号検索の CRC Error を設定 / 問い合わせします。	5-174
:CRCEFactor?	CAN FD バス信号検索の CRC エラー要因に関するすべての設定値を問い合わせます。	5-174
:CRCEFactor		
:CRCSequence	CAN FD バス信号検索の CRC エラー要因である CRC Sequence を設定 / 問い合わせします。	5-175
:SCount	CAN FD バス信号検索の CRC エラー要因である Stuff Count を設定 / 問い合わせします。	5-175
:FSTuff	CAN FD バス信号検索の固定 STUFF Error を設定 / 問い合わせします。	5-175
[:MODE]	CAN FD バス信号検索の Error Frame を設定 / 問い合わせします。	5-175
:STUFF	CAN FD バス信号検索の STUFF Error を設定 / 問い合わせします。	5-175
:FDF		
:CONDition	CAN FD バス信号検索の FDF の判定条件を設定 / 問い合わせします。	5-175
:IDData?	CAN FD バス信号検索の ID/Data 条件に関するすべての設定値を問い合わせます。	5-176
[:IDData]		
:ACK?	CAN FD バス信号検索の ID/Data 条件の ACK に関するすべての設定値を問い合わせます。	5-176
:ACK		
:MODE	CAN FD バス信号検索の ID/Data 条件の ACK モードの設定 / 問い合わせします。	5-176
:TYPE	CAN FD バス信号検索の ID/Data 条件の ACK 条件の設定 / 問い合わせします。	5-176
:DATA?	CAN FD バス信号検索の ID/Data 条件のデータに関するすべての設定値を問い合わせます。	5-176
:DATA		
:BCount	CAN FD バス信号検索の ID/Data 条件のパターン比較する位置を設定 / 問い合わせします。	5-176
:CONDition	CAN FD バス信号検索の ID/Data 条件のデータ判定条件の設定 / 問い合わせします。	5-176
:DBYTe	CAN FD バス信号検索の ID/Data 条件のデータのバイト数を設定 / 問い合わせします。	5-177
:DECimal<y>	CAN FD バス信号検索の ID/Data 条件の判定データを 10 進数で設定 / 問い合わせします。	5-177
:DPATtern?	CAN FD バス信号検索の ID/Data 条件の判定データに関するすべての設定値を問い合わせます。	5-177
:DPATtern		
:HEXa	CAN FD バス信号検索の ID/Data 条件の判定データを 16 進数で設定します。	5-177
:PATtern	CAN FD バス信号検索の ID/Data 条件のデータを 2 進数で設定 / 問い合わせします。	5-177
:ENDian	CAN FD バス信号検索の ID/Data 条件の判定データのエンディアンを設定 / 問い合わせします。	5-177
:MODE	CAN FD バス信号検索の ID/Data 条件のデータ条件 (有効 / 無効) を設定 / 問い合わせします。	5-177
:MSBLsb	CAN FD バス信号検索の ID/Data 条件のデータの MSB/LSB のビットを設定 / 問い合わせします。	5-178
:SIGN	CAN FD バス信号検索の ID/Data 条件のデータの符号を設定 / 問い合わせします。	5-178
:IDENtifier?	CAN FD バス信号検索の ID/Data 条件の識別子に関するすべての設定値を問い合わせます。	5-178
:IDENtifier		
:HEXa	CAN FD バス信号検索の ID/Data 条件の ID を 16 進数で設定します。	5-178
:MFORmat	CAN FD バス信号検索の ID/Data 条件の ID のフレームフォーマット (標準 / 拡張) を設定 / 問い合わせします。	5-178
:MODE	CAN FD バス信号検索の ID/Data 条件の識別子条件 (有効 / 無効) を設定 / 問い合わせします。	5-178
:PATtern	CAN FD バス信号検索の ID/Data 条件の ID 条件を 2 進数で設定 / 問い合わせします。	5-178
:PFORmat	CAN FD バス信号検索の ID/Data 条件の ID の入力形式を設定 / 問い合わせします。	5-179
:MSIGnal?	CAN FD バス信号検索の ID/Data 条件のメッセージシグナルに関するすべての設定値を問い合わせます。	5-179
:MSIGnal		
:MESSage		

5.1 コマンド一覧表

コマンド	機能	ページ
:ITEM	CAN FD バス信号検索の ID/Data 条件のメッセージのアイテムを設定します。	5-179
:SElect	CAN FD バス信号検索の ID/Data 条件のメッセージシグナルの条件を設定 / 問い合わせします。	5-179
:SIGNal?	CAN FD バス信号検索の ID/Data 条件のシグナルに関するすべての設定値を問い合わせます。	5-179
:SIGNal		
:CONDition	CAN FD バス信号検索の ID/Data 条件のシグナルのデータ条件を設定 / 問い合わせします。	5-179
:DECimal<y>	CAN FD バス信号検索の ID/Data 条件のシグナルの判定データを 10 進数で設定 / 問い合わせします。	5-179
:ITEM	CAN FD バス信号検索の ID とデータ条件のシグナルのアイテムを設定します。	5-180
:RTR	CAN FD バス信号検索の ID とデータ条件の RTR を設定 / 問い合わせします。	5-180
:MODE	CAN FD バス信号検索の検索の種類を設定 / 問い合わせします。	5-180
:CXPI?	CXPI バス信号の解析 / 検索に関するすべての設定値を問い合わせます。	5-180
:CXPI		
:ANALyze?	CXPI バス信号解析に関するすべての設定値を問い合わせます。	5-180
[:ANALyze]		
:SETup?	CXPI バス信号解析のバス設定に関するすべての設定値を問い合わせます。	5-180
:SETup		
:BRATe	CXPI バス信号解析のビットレート (データ転送速度) を設定 / 問い合わせします。	5-180
:CEDetection	CXPI バス信号解析の Counter Error 検出の有効 / 無効を設定 / 問い合わせします。	5-180
:CTOLerance	CXPI バス信号解析のクロック許容差を設定 / 問い合わせします。	5-181
:SOURce	CXPI バス信号解析のソースを設定 / 問い合わせします。	5-181
:TSAMple	CXPI バス信号解析の論理値 1/0 判定の閾値を設定 / 問い合わせします。	5-181
:DETail?	CXPI バス信号の解析結果リストに関するすべての設定値を問い合わせます。	5-181
:DETail		
:DISPlay	CXPI バス信号の解析結果リストの表示モードを設定 / 問い合わせします。	5-181
:LIST		
:ALL?	CXPI バス信号の解析結果リストのすべての解析番号の全データを問い合わせます。	5-181
:ITEM?	CXPI バス信号の解析結果リストに表示される項目を問い合わせます。	5-181
:VALue?	CXPI バス信号の解析結果リストの指定した解析番号の全データを問い合わせます。	5-181
:SEARch?	CXPI バス信号の検索に関するすべての設定値を問い合わせます。	5-182
:SEARch		
:ABORt	CXPI バス信号の検索を中止します。	5-182
:EXECute	CXPI バス信号の検索を実行します。	5-182
:SElect	CXPI バス信号検索のズームウインドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。	5-182
:SETup?	CXPI バス信号検索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。	5-182
:SETup		
:ERRor?	CXPI バス信号検索の Error に関するすべての設定値を問い合わせます。	5-182
:ERRor		
:CLOCK	CXPI バス信号検索の Clock Error を設定 / 問い合わせします。	5-182
:COUNter	CXPI バス信号検索の Counter Error を設定 / 問い合わせします。	5-182
:CRC	CXPI バス信号検索の CRC Error を設定 / 問い合わせします。	5-182
:DLENgth	CXPI バス信号検索の Data Length Error を設定 / 問い合わせします。	5-183
:FRAMing	CXPI バス信号検索の Framing Error を設定 / 問い合わせします。	5-183
:IBS	CXPI バス信号検索の IBS Error を設定 / 問い合わせします。	5-183
:PARity	CXPI バス信号検索の Parity Error を設定 / 問い合わせします。	5-183
:IDData?	CXPI バス信号検索の ID とデータ条件に関するすべての設定値を問い合わせます。	5-183
[:IDData]		
:DATA?	CXPI バス信号検索の ID とデータ条件のデータに関するすべての設定値を問い合わせます。	5-183
:DATA		
:BCount	CXPI バス信号検索の ID とデータ条件のパターン比較する位置を設定 / 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
:CONDition	CXPI バス信号検索の ID とデータ条件のデータ判定条件の設定 / 問い合わせ	5-184
:DBYTe	CXPI バス信号検索の ID とデータ条件のデータのバイト数を設定 / 問い合わせ	5-184
:DECimal<y>	CXPI バス信号検索の ID とデータ条件の判定データを 10 進数で設定 / 問い合わせ	5-184
:DPATtern?	CXPI バス信号検索の ID とデータ条件の判定データに関するすべての設定値を問い合わせ	5-184
:DPATtern		
:HEXa	CXPI バス信号検索の ID とデータ条件の判定データを 16 進数で設定	5-184
:PATtern	CXPI バス信号検索の ID とデータ条件の判定データを 2 進数で設定 / 問い合わせ	5-184
:ENDian	CXPI バス信号検索の ID とデータ条件の判定データのエンディアンを設定 / 問い合わせ	5-185
:MODE	CXPI バス信号検索の ID とデータ条件 (有効 / 無効) を設定 / 問い合わせ	5-185
:MSBLsb	CXPI バス信号検索の ID とデータ条件のデータの MSB/LSB のビットを設定 / 問い合わせ	5-185
:SIGN	CXPI バス信号検索の ID とデータ条件のデータの符号を設定 / 問い合わせ	5-185
:FINformation?	CXPI バス信号検索の ID とデータ条件のフレーム情報に関するすべての設定値を問い合わせ	5-185
:FINformation		
:CT	CXPI バス信号検索の ID とデータ条件のフレーム情報のカウンタ値を設定 / 問い合わせ	5-185
:MODE	CXPI バス信号検索の ID とデータ条件のフレーム情報 (有効 / 無効) を設定 / 問い合わせ	5-186
:SLEEP	CXPI バス信号検索の ID とデータ条件のフレーム情報の Sleep ビットを設定 / 問い合わせ	5-186
:WAKEup	CXPI バス信号検索の ID とデータ条件のフレーム情報の Wakeup ビットを設定 / 問い合わせ	5-186
:ID?	CXPI バス信号検索の ID とデータ条件の ID に関するすべての設定値を問い合わせ	5-186
:ID		
:HEXa	CXPI バス信号検索の ID とデータ条件の ID を 16 進数で設定	5-186
:MODE	CXPI バス信号検索の ID とデータ条件の ID 条件 (有効 / 無効) を設定 / 問い合わせ	5-186
:PATtern	CXPI バス信号検索の ID とデータ条件の ID を 2 進数で設定 / 問い合わせ	5-186
:PTYPE	CXPI バス信号検索の ID とデータ条件の PTYPE の判定条件を設定 / 問い合わせ	5-187
:MODE	CXPI バス信号検索の種類を設定 / 問い合わせ	5-187
:PTYPE?	CXPI バス信号検索の PTYPE に関するすべての設定値を問い合わせ	5-187
:PTYPE		
:CONDition	CXPI バス信号検索の PTYPE の判定条件を設定 / 問い合わせ	5-187
:WAKEupsleep?	CXPI バス信号検索の Wakeup と Sleep に関するすべての設定値を問い合わせ	5-187
:WAKEupsleep		
:SFRame	CXPI バス信号検索の Sleep フレームを設定 / 問い合わせ	5-187
:SLEEP	CXPI バス信号検索の Sleep (クロックが無い状態) を設定 / 問い合わせ	5-187
:WAKEup	CXPI バス信号検索の Wakeup (クロック有り状態) を設定 / 問い合わせ	5-188
:WPULse	CXPI バス信号検索の Wakeup パルスを設定 / 問い合わせ	5-188
:DECode?	シリアルバス信号のデコードに関するすべての設定値を問い合わせ	5-188
:DECode		
[:FORMat]	シリアルバス信号のデコード表示形式を設定 / 問い合わせ	5-188
:LABel	シリアルバス信号のデコードのラベル名を設定 / 問い合わせ	5-188
:POSition	シリアルバス信号のデコードの垂直ポジションを設定 / 問い合わせ	5-188
:SENT?	SENT 信号のデコードに関するすべての設定値を問い合わせ	5-188
:SENT		
:FAST?	SENT 信号の Fast CH のデコードに関するすべての設定値を問い合わせ	5-188
:FAST		
:LABel	SENT 信号の Fast CH のデコードのラベル名を設定 / 問い合わせ	5-189

5.1 コマンド一覧表

コマンド	機能	ページ
:POSition	SENT 信号の Fast CH のデコードの垂直ポジションを設定 / 問い合わせします。	5-189
:SLOW?	SENT 信号の Slow CH のデコードに関するすべての設定値を問い合わせます。	5-189
:SLOW		
:LABel	SENT 信号の Slow CH のデコードのラベル名を設定 / 問い合わせします。	5-189
:POSition	SENT 信号の Slow CH のデコードの垂直ポジションを設定 / 問い合わせします。	5-189
:SPI?	SPI バス信号のデコードに関するすべての設定値を問い合わせます。	5-189
:SPI		
:DATA<y>?	SPI バス信号のデコードの各データに関するすべての設定値を問い合わせます。	5-189
:DATA<y>		
:LABel	SPI バス信号のデコードの各ラベル名を設定 / 問い合わせします。	5-189
:POSition	SPI バス信号のデコードの各垂直ポジションを設定 / 問い合わせします。	5-190
:DISPlay	シリアルバス信号の解析をする (ON) / しない (OFF) を設定 / 問い合わせします。	5-190
:FLEXray?	FlexRay バス信号の解析 / 検索に関するすべての設定値を問い合わせます。	5-190
:FLEXray		
:ANALyze?	FlexRay バス信号解析に関するすべての設定値を問い合わせます。	5-190
[:ANALyze]		
:SETup?	FlexRay バス信号解析のバス設定に関するすべての設定値を問い合わせます。	5-190
:SETup		
:BCHannel	FlexRay バス信号解析のチャンネルのバスタイプを設定 / 問い合わせします。	5-190
:BRATe	FlexRay バス信号解析のビットレート (データ転送速度) を設定 / 問い合わせします。	5-190
:SOURce	FlexRay バス信号解析のソースを設定 / 問い合わせします。	5-191
:DETail?	FlexRay バス信号の解析結果リストに関するすべての設定値を問い合わせます。	5-191
:DETail		
:DISPlay	FlexRay バス信号の解析結果リストの表示モードを設定 / 問い合わせします。	5-191
:LIST		
:ITEM?	FlexRay バス信号の解析結果リストに表示される項目を問い合わせます。	5-191
:VALue?	FlexRay バス信号の解析結果リストの指定した解析番号の全データを問い合わせます。	5-191
:SEARch?	FlexRay バス信号検索に関するすべての設定値を問い合わせます。	5-191
:SEARch		
:ABORt	FlexRay バス信号検索を中止します。	5-191
:EXECute	FlexRay バス信号検索を実行します。	5-191
:FJUMp		
:CCOunt	FlexRay バス信号解析の結果を対象に Cycle Count Field へのフィールドジャンプを実行します。	5-191
:CRC	FlexRay バス信号解析の結果を対象に CRC Field へのフィールドジャンプを実行します。	5-191
:FRAMeid	FlexRay バス信号解析の結果を対象に Frame ID Field へのフィールドジャンプを実行します。	5-192
:HCRC	FlexRay バス信号解析の結果を対象に Header CRC Field へのフィールドジャンプを実行します。	5-192
:PLENght	FlexRay バス信号解析の結果を対象に Payload Length Field へのフィールドジャンプを実行します。	5-192
:SElect	FlexRay バス信号検索による検出波形番号を設定し、検出波形番号に相当するズーム位置を問い合わせます。	5-192
:SETup?	FlexRay バス信号検索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。	5-192
:SETup		
:ERRor?	FlexRay バス信号検索の Error に関するすべての設定値を問い合わせます。	5-192
:ERRor		
:BSS	FlexRay バス信号検索の BSS Error を設定 / 問い合わせします。	5-192
:CRC	FlexRay バス信号検索の CRC Error を設定 / 問い合わせします。	5-192
:FES	FlexRay バス信号検索の FES Error を設定 / 問い合わせします。	5-193
:HCRC	FlexRay バス信号検索の Header CRC Error を設定 / 問い合わせします。	5-193
:IDData?	FlexRay バス信号検索の ID とデータ条件に関するすべての設定値を問い合わせます。	5-193
[:IDData]		

5.1 コマンド一覧表

コマンド	機能	ページ
:CCOut?	FlexRay バス信号検索の Cycle Count に関するすべての設定値を問い合わせます。	5-193
:CCOut		
:CONDition	FlexRay バス信号検索の Cycle Count のデータ条件を設定 / 問い合わせします。	5-193
:COUNT<y>	FlexRay バス信号検索の Cycle Count を設定 / 問い合わせします。	5-193
:MODE	FlexRay バス信号検索の ID とデータ条件の Cycle Count モードを設定 / 問い合わせします。	5-194
:DATA?	FlexRay バス信号検索の ID とデータ条件のデータに関するすべての設定値を問い合わせます。	5-194
:DATA		
:BCount	FlexRay バス信号検索の ID とデータ条件のパターン比較する位置を設定 / 問い合わせします。	5-194
:CONDition	FlexRay バス信号検索の ID とデータ条件のデータ判定条件を設定 / 問い合わせします。	5-194
:DBYTe	FlexRay バス信号検索の ID とデータ条件のデータのバイト数を設定 / 問い合わせします。	5-194
:DECimal<y>	FlexRay バス信号検索の ID とデータ条件の判定データを 10 進数で設定 / 問い合わせします。	5-194
:DPATtern?	FlexRay バス信号検索の ID とデータ条件の判定データに関わるすべての設定値を問い合わせます。	5-195
:DPATtern		
:HEXa	FlexRay バス信号検索の ID とデータ条件の判定データを 16 進数で設定します。	5-195
:PATtern	FlexRay バス信号検索の ID とデータ条件のデータを 2 進数で設定 / 問い合わせします。	5-195
:ENDian	FlexRay バス信号検索の ID とデータ条件の判定データのエンディアンを設定 / 問い合わせします。	5-195
:MODE	FlexRay バス信号検索の ID とデータ条件のデータ条件 (有効 / 無効) の設定 / 問い合わせします。	5-195
:MSBLsb	FlexRay バス信号検索の ID とデータ条件のデータの MSB/LSB のビットを設定 / 問い合わせします。	5-195
:SIGN	FlexRay バス信号検索の ID とデータ条件のデータの符号を設定 / 問い合わせします。	5-196
:FID?	FlexRay バス信号検索の ID とデータ条件の Frame ID に関するすべての設定値を問い合わせます。	5-196
:FID		
:CONDition	FlexRay バス信号検索の ID とデータ条件の Frame ID のデータ条件を設定 / 問い合わせします。	5-196
:ID<y>	FlexRay バス信号検索の ID とデータ条件の Frame ID の値を設定 / 問い合わせします。	5-196
:MODE	FlexRay バス信号検索の ID とデータ条件の Frame ID 条件 (有効 / 無効) の設定 / 問い合わせします。	5-196
:INDicator?	FlexRay バス信号検索の ID とデータ条件の Indicator に関するすべての設定値を問い合わせます。	5-196
:INDicator		
:MODE	FlexRay バス信号検索の ID とデータ条件の Indicator 条件 (有効 / 無効) の設定 / 問い合わせします。	5-197
:NFrame	FlexRay バス信号検索の ID とデータ条件の Indicator の Null frame を設定 / 問い合わせします。	5-197
:PPreamble	FlexRay バス信号検索の ID とデータ条件の Indicator の Payload preamble を設定 / 問い合わせします。	5-197
:STFrame	FlexRay バス信号検索の ID とデータ条件の Indicator の Start frame を設定 / 問い合わせします。	5-197
:SYFrame	FlexRay バス信号検索の ID とデータ条件の Indicator の Sync frame を設定 / 問い合わせします。	5-197
:MODE	FlexRay バス信号検索のサーチモードを設定 / 問い合わせします。	5-197
:I2C?	I2C バス信号の解析 / 検索に関するすべての設定値を問い合わせます。	5-198
:I2C		
:ANALyze?	I2C バス信号解析に関するすべての設定値を問い合わせます。	5-198
[:ANALyze]		
:SETup?	I2C バス信号解析のバス設定に関するすべての設定値を問い合わせます。	5-198
:SETup		
:CLOCK?	I2C バス信号解析のクロックに関するすべての設定値を問い合わせます。	5-198

5.1 コマンド一覧表

コマンド	機能	ページ
:CLOCK		
:SOURCE	I2C バス信号解析のクロックを設定 / 問い合わせします。	5-198
:DATA?	I2C バス信号解析のデータに関するすべての設定値を問い合わせます。	5-198
:DATA		
:SOURCE	I2C バス信号解析のデータを設定 / 問い合わせします。	5-198
:INCLuderw	I2C バス信号解析のアドレス R/W の有効 (ON)/ 無効 (OFF) を設定 / 問い合わせします。	5-198
:DETail?	I2C バス信号の解析結果リストに関するすべての設定値を問い合わせます。	5-198
:DETail		
:DISPlay	I2C バス信号の解析結果リストの表示モードを設定 / 問い合わせします。	5-199
:LIST		
:ITEM?	I2C バス信号の解析結果リストに表示される項目を問い合わせます。	5-199
:VALue?	I2C バス信号の解析結果リストの指定した解析番号の全データを問い合わせます。	5-199
:SEARCh?	I2C バス信号検索に関するすべての設定値を問い合わせます。	5-199
:SEARCh		
:ABORt	I2C バス信号検索を中止します。	5-199
:EXECute	I2C バス信号検索を実行します。	5-199
:SElect	I2C バス信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。	5-199
:SETup?	I2C バス信号検索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。	5-199
[:SETup]		
:ADDRess?	I2C バス信号検索のアドレスパターンに関するすべての設定値を問い合わせます。	5-199
:ADDRess		
:ADDRess?	I2C バス信号検索のアドレスパターンのアドレスに関するすべての設定値を問い合わせます。	5-199
:ADDRess		
:BIT10Address?	I2C バス信号検索の 10bit アドレスに関するすべての設定値を問い合わせます。	5-199
:BIT10Address		
:DIRectioN	I2C バス信号検索の 10bit アドレスのデータ方向を設定 / 問い合わせします。	5-200
:HEXa	I2C バス信号検索の 10bit アドレスを 16 進数で設定します。	5-200
:PATtern	I2C バス信号検索の 10bit アドレスを 2 進数で設定 / 問い合わせします。	5-200
:BIT7Address?	I2C バス信号検索の 7bit アドレスに関するすべての設定値を問い合わせます。	5-200
[:BIT7Address]		
:DIRectioN	I2C バス信号検索の 7bit アドレスのデータ方向を設定 / 問い合わせします。	5-200
:HEXa	I2C バス信号検索の 7bit アドレスを 16 進数で設定します。	5-200
:PATtern	I2C バス信号検索の 7bit アドレスを 2 進数で設定 / 問い合わせします。	5-201
:BIT7APsub?	I2C バス信号検索の 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。	5-201
:BIT7APsub		
:ADDRess?	I2C バス信号検索の 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。	5-201
:ADDRess		
:DIRectioN	I2C バス信号検索の 7bit+Sub アドレスのデータ方向を設定 / 問い合わせします。	5-201
:HEXa	I2C バス信号検索の 7bit+Sub アドレスの 7bit アドレスを 16 進数で設定します。	5-201
:PATtern	I2C バス信号検索の 7bit+Sub アドレスの 7bit アドレスを 2 進数で設定 / 問い合わせします。	5-201
:SADDRess?	I2C バス信号検索の 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。	5-202
:SADDRess		
:HEXa	I2C バス信号検索の 7bit+Sub アドレスの Sub アドレスを 16 進数で設定します。	5-202
:PATtern	I2C バス信号検索の 7bit+Sub アドレスの Sub アドレスを 2 進数で設定 / 問い合わせします。	5-202
:MODE	I2C バス信号検索のアドレス条件の有効 / 無効を設定 / 問い合わせします。	5-202
:TYPE	I2C バス信号検索のアドレス条件のアドレス形式を設定 / 問い合わせします。	5-202

5.1 コマンド一覧表

コマンド	機能	ページ
:DATA?	I2C バス信号検索のアドレスパターンのデータに関するすべての設定値を問い合わせます。	5-202
:DATA		
:BCount	I2C バス信号検索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。	5-203
:BMODe	I2C バス信号検索のデータのパターンを比較する位置の ON/OFF を設定 / 問い合わせします。	5-203
:CONDition	I2C バス信号検索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。	5-203
:DBYTe	I2C バス信号検索の判定データ数を設定 / 問い合わせします。	5-203
:DPATtern?	I2C バス信号検索のデータ条件のデータに関するすべての設定値を問い合わせます。	5-203
:DPATtern		
:HEXa	I2C バス信号検索のデータ条件のデータを 16 進数で設定します。	5-203
:PATtern	I2C バス信号検索のデータ条件のデータを 2 進数で設定 / 問い合わせします。	5-204
:MODE	I2C バス信号検索のデータ条件の有効 / 無効を設定 / 問い合わせします。	5-204
:GENeralcall?	I2C バス信号検索のゼ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。	5-204
:GENeralcall		
:BIT7Maddress?	I2C バス信号検索のゼネラルコールの 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。	5-204
:BIT7Maddress		
:HEXa	I2C バス信号検索のゼネラルコールの 7bit マスタアドレスを 16 進数で設定します。	5-204
:PATtern	I2C バス信号検索のゼネラルコールの 7bit マスタアドレスを 2 進数で設定 / 問い合わせします。	5-204
:SBYTe	I2C バス信号検索のゼ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。	5-204
:MODE	I2C バス信号検索の検索の種類を設定 / 問い合わせします。	5-205
:NONack?	I2C バス信号検索の 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。	5-205
:NONack		
:HSMoDe	I2C バス信号検索のハイスピードモードで NONACK を無視する / しないを設定 / 問い合わせします。	5-205
:READaccess	I2C バス信号検索のリードアクセスモードで NONACK を無視する / しないを設定 / 問い合わせします。	5-205
:STARtbyte	I2C バス信号検索のスタートバイトで NONACK を無視する / しないを設定 / 問い合わせします。	5-205
:LIN?	LIN バス信号の解析 / 検索に関するすべての設定値を問い合わせます。	5-205
:LIN		
:ANALyze?	LIN バス信号解析に関するすべての設定値を問い合わせます。	5-205
[:ANALyze]		
:SETup?	LIN バス信号解析のバス設定に関するすべての設定値を問い合わせます。	5-205
:SETup		
:BRATe	LIN バス信号解析のビットレート (データ転送速度) を設定 / 問い合わせします。	5-206
:REVision	LIN バス信号解析のレビジョン番号を設定 / 問い合わせします。	5-206
:SOURce	LIN バス信号解析のソースを設定 / 問い合わせします。	5-206
:SPoiNt	LIN バス信号解析のサンプルポイントを設定 / 問い合わせします。	5-206
:DETail?	LIN バス信号の解析結果リストに関するすべての設定値を問い合わせます。	5-206
:DETail		
:DISPlay	LIN バス信号の解析結果リストの表示モードを設定 / 問い合わせします。	5-206
:LIST		
:ITEM?	LIN バス信号の解析結果リストに表示される項目を問い合わせます。	5-206
:VALue?	LIN バス信号の解析結果リストの指定した解析番号の全データを問い合わせます。	5-207
:SEARch?	LIN バス信号検索に関するすべての設定値を問い合わせます。	5-207
:SEARch		
:ABORt	LIN バス信号検索を中止します。	5-207
:EXECute	LIN バス信号検索を実行します。	5-207
:FJUMp		
:BREAk	LIN バス信号検索の結果の Break Field へのフィールドジャンプを実行します。	5-207

5.1 コマンド一覧表

コマンド	機能	ページ
:CSUM	LIN バス信号検索の結果の Checksum Field へのフィールドジャンプを実行します。	5-207
:DATA	LIN バス信号検索の結果の Data Field へのフィールドジャンプを実行します。	5-207
:IDENtifier	LIN バス信号検索の結果の Identifier Field へのフィールドジャンプを実行します。	5-207
:SYNCh	LIN バス信号検索の結果の Sync Field へのフィールドジャンプを実行します。	5-207
:SElect	LIN バス信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。	5-207
:SElect? MAXimum	LIN バス信号検索の検索点の総数を問い合わせます。	5-207
:SETup?	LIN バス信号検索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。	5-207
:SETup		
:ERRor?	LIN バス信号検索の Error に関するすべての設定値を問い合わせます。	5-208
:ERRor		
:CHECksum	LIN バス信号検索の Checksum Error を設定 / 問い合わせします。	5-208
:FRAMing	LIN バス信号検索の Framing Error を設定 / 問い合わせします。	5-208
:PARity	LIN バス信号検索の Parity Error を設定 / 問い合わせします。	5-208
:SYNCh	LIN バス信号検索の Synch Error を設定 / 問い合わせします。	5-208
:TIMEout	LIN バス信号検索の Timeout Error を設定 / 問い合わせします。	5-208
:IDData?	LIN バス信号検索の ID とデータ条件に関するすべての設定値を問い合わせます。	5-209
:IDData		
:DATA?	LIN バス信号検索のデータに関するすべての設定値を問い合わせます。	5-209
:DATA		
:CONDition	LIN バス信号検索のデータ条件を設定 / 問い合わせします。	5-209
:DBYTe	LIN バス信号検索のデータのバイト数を設定 / 問い合わせします。	5-209
:DECimal<y>	LIN バス信号検索のデータを 10 進数で設定 / 問い合わせします。	5-209
:DPATtern?	LIN バス信号検索のデータに関わるすべての設定値を問い合わせます。	5-209
:DPATtern		
:HEXa	LIN バス信号検索のデータを 16 進数で設定します。	5-209
:PATtern	LIN バス信号検索のデータを 2 進数で設定 / 問い合わせします。	5-209
:ENDian	LIN バス信号検索のデータのエンディアンを設定 / 問い合わせします。	5-210
:MODE	LIN バス信号検索のデータ条件 (有効 / 無効) の設定 / 問い合わせします。	5-210
:MSBLsb	LIN バス信号検索のデータの MSB/LSB のビットを設定 / 問い合わせします。	5-210
:SIGN	LIN バス信号検索のデータの符号を設定 / 問い合わせします。	5-210
:IDENtifier?	LIN バス信号検索の識別子に関するすべての設定値を問い合わせます。	5-210
:IDENtifier		
:ID?	LIN バス信号検索の ID に関するすべての設定値を問い合わせます。	5-210
:ID		
:HEXa	LIN バス信号検索の ID を 16 進数で設定します。	5-210
:MODE	LIN バス信号検索の ID 条件 (有効 / 無効) の設定 / 問い合わせします。	5-211
:PATtern	LIN バス信号検索の ID を 2 進数で設定 / 問い合わせします。	5-211
:MODE	LIN バス信号検索の検索の種類を設定 / 問い合わせします。	5-211
:PSI5?	PSI5 信号の解析 / 検索に関するすべての設定値を問い合わせます。	5-211
:PSI5		
:ANALyze?	PSI5 信号解析に関するすべての設定値を問い合わせます。	5-211
[:ANALyze]		
:SETup?	PSI5 信号解析のバス設定に関するすべての設定値を問い合わせます。	5-211
:SETup		
:DATA?	PSI5 信号解析のデータ信号に関するすべての設定値を問い合わせます。	5-211
:DATA		
:BRATe	PSI5 信号解析のビットレート (データ転送速度) を設定 / 問い合わせします。	5-211
:CTOLerance	PSI5 信号解析のクロック許容範囲を設定 / 問い合わせします。	5-211
:DBITs	PSI5 信号解析のデータ長を設定 / 問い合わせします。	5-212
:EDETection	PSI5 信号解析のエラー検出方式を設定 / 問い合わせします。	5-212
:SNRejection?	PSI5 信号解析の同期ノイズ除去のすべての設定値を問い合わせます。	5-212
:SNRejection		
:ETIME	PSI5 信号解析の同期ノイズ除去の終了時間を設定 / 問い合わせします。	5-212
:MODE	PSI5 信号解析の同期ノイズ除去の ON/OFF を設定 / 問い合わせします。	5-212
:SOURce	PSI5 信号解析のデータソースを設定 / 問い合わせします。	5-212
:NUMBerofslot	PSI5 信号解析のスロット数を設定 / 問い合わせします。	5-213
:SLOT<y>	PSI5 信号解析のスロットの開始 / 終了時間を設定 / 問い合わせします。	5-213

5.1 コマンド一覧表

コマンド	機能	ページ
:SYNC	PSI5 信号解析の同期信号を設定 / 問い合わせします。	5-213
:DETail?	PSI5 信号解析の解析リストに関するすべての設定値を問い合わせます。	5-213
:DETail		
:DISPlay	PSI5 信号解析の解析リストの表示モードを設定 / 問い合わせします。	5-213
:LIST		
:ALL?	PSI5 信号の解析結果リストのすべての解析番号の全データを問い合わせます。	5-213
:ITEM?	PSI5 信号の解析結果リストに表示される項目を問い合わせます。	5-214
:VALue?	PSI5 信号の解析結果リストの指定した解析番号の全データを問い合わせます。	5-214
:SEARCh?	PSI5 信号検索に関するすべての設定値を問い合わせます。	5-214
:SEARCh		
:ABORt	PSI5 信号検索を中止します。	5-214
:EXECute	PSI5 信号検索を実行します。	5-214
:SELect	PSI5 信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。	5-214
:SETup?	PSI5 信号検索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。	5-214
:SETup		
:DATA?	PSI5 信号のデータ検索に関するすべての設定値を問い合わせます。	5-214
:DATA		
:DATA?	PSI5 信号のデータ検索のデータに関するすべての設定値を問い合わせます。	5-214
:DATA		
:CONDition	PSI5 信号のデータ検索の判定条件を設定 / 問い合わせします。	5-215
:DECimal<y>	PSI5 信号のデータ検索のデータを 10 進数で設定 / 問い合わせします。	5-215
:HEXa	PSI5 信号のデータ検索のデータを 16 進数で設定します。	5-215
:PATtern	PSI5 信号のデータ検索のデータを 2 進数で設定 / 問い合わせします。	5-215
:FRAMeinslot?	PSI5 信号のデータ検索のスロット指定に関するすべての設定値を問い合わせます。	5-215
:FRAMeinslot		
:MODE	PSI5 信号のデータ検索のスロット指定条件 (有効 / 無効) を設定 / 問い合わせします。	5-215
:SNUMber	PSI5 信号のデータ検索のスロット番号を設定 / 問い合わせします。	5-216
:ERRor?	PSI5 信号の Error 検索に関するすべての設定値を問い合わせます。	5-216
:ERRor		
:FNUMber	PSI5 信号の Frame Number Error 検索を設定 / 問い合わせします。	5-216
:FRAMe	PSI5 信号の Frame Error 検索を設定 / 問い合わせします。	5-216
:MANChester	PSI5 信号の Manchester Error 検索を設定 / 問い合わせします。	5-216
:PCRC	PSI5 信号の Parity/CRC Error 検索を設定 / 問い合わせします。	5-216
:SBIT	PSI5 信号の Start Bit Error 検索を設定 / 問い合わせします。	5-217
:SBOundary	PSI5 信号の Slot Boundary Error 検索を設定 / 問い合わせします。	5-217
:FRAMeinslot?	PSI5 信号のスロット指定検索のすべての設定値を問い合わせます。	5-217
:FRAMeinslot		
:SNUMber	PSI5 信号のスロット指定検索のスロット番号を設定 / 問い合わせします。	5-217
:MODE	PSI5 信号検索の種類を設定 / 問い合わせします。	5-217
:TREND<y>?	PSI5 信号解析の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。	5-217
:TREND<y>		
:AScale	PSI5 信号解析のトレンド表示のオートスケールを実行します。	5-217
:CURSor?	PSI5 信号解析の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。	5-217
:CURSor		
:DT		
:VALue?	PSI5 信号解析のトレンド表示のカーソル間の時間軸値を問い合わせます。	5-218
:DV		
:VALue?	PSI5 信号解析のトレンド表示のカーソル間の垂直軸値を問い合わせます。	5-218
:MODE	PSI5 信号解析のトレンド表示のカーソルのモードを設定 / 問い合わせします。	5-218
:POSition<z>	PSI5 信号解析のトレンド表示の各カーソルの位置を設定 / 問い合わせします。	5-218
:T<z>		
:VALue?	PSI5 信号解析のトレンド表示のカーソルの時間軸値を問い合わせます。	5-218
:V<z>		
:VALue?	PSI5 信号解析のトレンド表示のカーソルの垂直軸値を問い合わせます。	5-218
:DISPlay	PSI5 信号解析のトレンドの表示の ON/OFF を設定 / 問い合わせします。	5-218

5.1 コマンド一覧表

コマンド	機能	ページ
:HRANge	PSI5 信号解析の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。	5-218
:SOURce	PSI5 信号解析のトレンド表示の対象スロット番号を設定 / 問い合わせします。	5-219
:VERTical	PSI5 信号解析のトレンド表示の垂直レンジを設定 / 問い合わせします。	5-219
:VTDisplay	PSI5 信号解析のトレンド表示の VT 波形の表示の ON/OFF を設定 / 問い合わせします。	5-219
:RWINdow	検索点を拡大表示するウィンドウで検索された部分を表示するズーム画面を設定 / 問い合わせします。	5-219
:SENT?	SENT 信号の解析 / 検索に関するすべての設定値を問い合わせます。	5-219
:SENT		
:ANALyze?	SENT 信号解析の関するすべての設定値を問い合わせます。	5-219
[:ANALyze]		
:SETup?	SENT 信号解析のパス設定に関するすべての設定値を問い合わせます。	5-219
:SETup		
:DISPlay	SENT 信号解析の解析結果の表示方法を設定 / 問い合わせします。	5-219
:FAST?	SENT 信号解析の Fast CH に関するすべての設定値を問い合わせます。	5-219
:FAST		
:DType	SENT 信号解析の Fast CH のデータ形式を設定 / 問い合わせします。	5-220
:USEUp?	SENT 信号解析の Fast CH のユーザー定義データに関するすべての設定値を問い合わせます。	5-220
:USEUp		
:DATA<y>?	SENT 信号解析の Fast CH の各ユーザー定義データに関するすべての設定値を問い合わせます。	5-220
:DATA<y>		
:MODE	SENT 信号解析の Fast CH の各ユーザー定義データの有効 / 無効を設定 / 問い合わせします。	5-220
:ORDer	SENT 信号解析の Fast CH の各ユーザー定義データのエンディアンを設定 / 問い合わせします。	5-220
:SIZE	SENT 信号解析の Fast CH の各ユーザー定義データのデータサイズを設定 / 問い合わせします。	5-220
:MULTiplexing	SENT 信号解析の FastCH のユーザー定義データで、Multiplexing の有効 / 無効を設定 / 問い合わせします。	5-220
:FORMat?	SENT 信号解析の解析フォーマットに関するすべての設定値を問い合わせます。	5-221
:FORMat		
:CEFactor?	SENT 信号解析のエラー要因に関するすべての設定値を問い合わせます。	5-221
:CEFactor		
:SAComm?	SENT 信号解析の Status and Communication のエラー要因に関するすべての設定値を問い合わせます。	5-221
:SAComm		
:BIT<y>?	SENT 信号解析の Status and Communication のエラー要因のビットを設定 / 問い合わせします。	5-221
:SCPulses	SENT 信号解析の連続キャリブレーションパルスのエラー要因を設定 / 問い合わせします。	5-221
:CRCType	SENT 信号解析の CRC 演算方式を設定 / 問い合わせします。	5-221
:CTICk	SENT 信号解析のクロックティック値を設定 / 問い合わせします。	5-221
:CTOLerance	SENT 信号解析のクロック許容差を設定 / 問い合わせします。	5-222
:DNIBbles	SENT 信号解析のデータニブル数を設定 / 問い合わせします。	5-222
:PPULse	SENT 信号解析のポーズパルスの有無を設定 / 問い合わせします。	5-222
:VERSion	SENT 信号解析の仕様バージョンを設定 / 問い合わせします。	5-222
:STYPe	SENT 信号解析の Slow CH 形式を設定 / 問い合わせします。	5-222
:SOURce	SENT 信号解析のソースを設定 / 問い合わせします。	5-222
:DETail?	SENT 信号解析の解析リストに関するすべての設定値を問い合わせます。	5-223
:DETail		
:DISPlay	SENT 信号解析の解析リストの表示モードを設定 / 問い合わせします。	5-223
:LIST		
:ALL?	SENT 信号の解析結果リストのすべての解析番号の全データを問い合わせます。	5-223
:ITEM?	SENT 信号の解析結果リストに表示される項目を問い合わせます。	5-223
:VALue?	SENT 信号の解析結果リストの指定した解析番号の全データを問い合わせます。	5-223
:SEARch?	SENT 信号検索に関するすべての設定値を問い合わせます。	5-223
:SEARch		

5.1 コマンド一覧表

コマンド	機能	ページ
:ABORt	SENT 信号検索を中止します。	5-223
:EXECute	SENT 信号検索を実行します。	5-223
:SELEct	SENT 信号検索のズームウインドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。	5-223
:SELEct? MAXimum	SENT 信号検索の検索点の総数を問い合わせます。	5-224
:SETup?	SENT 信号検索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。	5-224
[:SETup]		
:ERRor?	SENT 信号検索の Error に関するすべての設定値を問い合わせます。	5-224
:ERRor		
:FCRC	SENT 信号検索の Fast CH の CRC エラーを設定 / 問い合わせします。	5-224
:NDValue	SENT 信号検索のニブルデータバリュエラーを設定 / 問い合わせします。	5-224
:NNUmber	SENT 信号検索のニブルデータ数エラーを設定 / 問い合わせします。	5-224
:SAComm	SENT 信号検索の Status and Communication エラーを設定 / 問い合わせします。	5-225
:SCPulses	SENT 信号検索の連続キャリブレーションパルスエラーを設定 / 問い合わせします。	5-225
:SCRC	SENT 信号検索の Slow CH の CRC エラーを設定 / 問い合わせします。	5-225
:FDATa?	SENT 信号検索の Fast CH のデータに関するすべての設定値を問い合わせます。	5-225
:FDATa		
:DATA<y>?	SENT 信号検索の Fast CH の各ユーザーデータに関するすべての設定値を問い合わせます。	5-225
:DATA<y>		
:CONDition	SENT 信号検索の Fast CH の各ユーザーデータの判定条件を設定 / 問い合わせします。	5-225
:DECimal<z>	SENT 信号検索の Fast CH の各ユーザーデータのデータを 10 進数で設定 / 問い合わせします。	5-225
:MODE	SENT 信号検索の Fast CH の各ユーザーデータのデータ条件 (有効 / 無効) を設定 / 問い合わせします。	5-226
:DNIBbles?	SENT 信号検索の Fast CH のニブルデータに関するすべての設定値を問い合わせます。	5-226
:DNIBbles		
:CONDition	SENT 信号検索の Fast CH のニブルデータの判定条件を設定 / 問い合わせします。	5-226
:HEXa	SENT 信号検索の Fast CH のニブルデータのデータを 16 進数で設定します。	5-226
:PATtern	SENT 信号検索の Fast CH のニブルデータのデータを 2 進数で設定 / 問い合わせします。	5-226
:FSAComm?	SENT 信号検索の Fast CH の Status and Communication ニブルに関するすべての設定値を問い合わせます。	5-226
:FSAComm		
:HEXa	SENT 信号検索の Fast CH の Status and Communication ニブルのデータを 16 進数で設定します。	5-226
:PATtern	SENT 信号検索の Fast CH Status and Communication ニブルのデータを 2 進数で設定 / 問い合わせします。	5-226
:MODE	SENT 信号検索のモードを設定 / 問い合わせします。	5-227
:SDATa?	SENT 信号検索の Slow CH のデータに関するすべての設定値を問い合わせます。	5-227
:SDATa		
:ENHanced?	SENT 信号検索の Slow CH の Enhanced タイプに関するすべての設定値を問い合わせます。	5-227
:ENHanced		
:CBIT	SENT 信号検索の Slow CH の Enhanced タイプの Configuration ビットを設定 / 問い合わせします。	5-227
:D12Bit?	SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID に関するすべての設定値を問い合わせます。	5-227
:D12Bit		
:DATA?	SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータに関するすべての設定値を問い合わせます。	5-227
:DATA		
:CONDition	SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータの判定条件を設定 / 問い合わせします。	5-227
:DECimal<y>	SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータを 10 進数で設定 / 問い合わせします。	5-228

5.1 コマンド一覧表

コマンド	機能	ページ
:HEXa	SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータを 16 進数で設定します。	5-228
:MODE	SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータ条件 (有効 / 無効) を設定 / 問い合わせします。	5-228
:PATtern	SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータを 2 進数で設定 / 問い合わせします。	5-228
:ID?	SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID の ID に関するすべての設定値を問い合わせます。	5-228
:ID		
:CONDition	SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID の ID 判定条件を設定 / 問い合わせします。	5-228
:DECimal<y>	SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID の ID を 10 進数で設定 / 問い合わせします。	5-229
:MODE	SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID の ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-229
:Dl6Bit?	SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID に関するすべての設定値を問い合わせます。	5-229
:Dl6Bit		
:DATA?	SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID 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
:DATA		
:CONDition	SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータの判定条件を設定 / 問い合わせします。	5-229
:DECimal<y>	SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータを 10 進数で設定 / 問い合わせします。	5-229
:HEXa	SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータを 16 進数で設定します。	5-229
:MODE	SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータ条件 (有効 / 無効) を設定 / 問い合わせします。	5-230
:PATtern	SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータを 2 進数で設定 / 問い合わせします。	5-230
:ID?	SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID に関するすべての設定値を問い合わせます。	5-230
:ID		
:CONDition	SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID 判定条件を設定 / 問い合わせします。	5-230
:DECimal<y>	SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID を 10 進数で設定 / 問い合わせします。	5-230
:MODE	SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-231
:SHORT?	SENT 信号検索の Slow CH の Short タイプに関するすべての設定値を問い合わせます。	5-231
:SHORT		
:DATA?	SENT 信号検索の Slow CH の Short タイプのデータに関するすべての設定値を問い合わせます。	5-231
:DATA		
:CONDition	SENT 信号検索の Slow CH の Short タイプのデータの判定条件を設定 / 問い合わせします。	5-231
:DECimal<y>	SENT 信号検索の Slow CH の Short タイプのデータを 10 進数で設定 / 問い合わせします。	5-231
:HEXa	SENT 信号検索の Slow CH の Short タイプのデータを 16 進数で設定します。	5-231
:MODE	SENT 信号検索の Slow CH の Short タイプのデータ条件 (有効 / 無効) を設定 / 問い合わせします。	5-231
:PATtern	SENT 信号検索の Slow CH の Short タイプのデータを 2 進数で設定 / 問い合わせします。	5-232
:ID?	SENT 信号検索の Slow CH の Short タイプの ID に関するすべての設定値を問い合わせます。	5-232
:ID		
:CONDition	SENT 信号検索の Slow CH の Short タイプの ID 判定条件を設定 / 問い合わせします。	5-232
:DECimal<y>	SENT 信号検索の Slow CH の Short タイプの ID を 10 進数で設定 / 問い合わせします。	5-232
:MODE	SENT 信号検索の Slow CH の Short タイプの ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-232

5.1 コマンド一覧表

コマンド	機能	ページ
:TRENd<y>?	SENT 信号解析の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。	5-232
:TRENd<y>		
:ASCaLe	SENT 信号解析のトレンド表示のオートスケールを実行します。	5-232
:CURSor?	SENT 信号解析の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。	5-232
:CURSor		
:DT		
:VALue?	SENT 信号解析のトレンド表示のカーソル間の時間軸値を問い合わせます。	5-233
:DV		
:VALue?	SENT 信号解析のトレンド表示のカーソル間の垂直軸値を問い合わせます。	5-233
:MODE	SENT 信号解析のトレンド表示の自動測定モードを設定 / 問い合わせします。	5-233
:POSition<z>	SENT 信号解析のトレンド表示の各カーソルの位置を設定 / 問い合わせします。	5-233
:T<z>		
:VALue?	SENT 信号解析のトレンド表示のカーソルの時間軸値を問い合わせます。	5-233
:V<z>		
:VALue?	SENT 信号解析のトレンド表示のカーソルの垂直軸値を問い合わせます。	5-233
:DISPlay	SENT 信号解析のトレンドを表示する (ON)/ 表示しない (OFF) を設定 / 問い合わせします。	5-233
:FCONtrol	SENT 信号解析のトレンド表示 FrameControl 値を設定 / 問い合わせします。	5-233
:HRAnge	SENT 信号解析の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。	5-234
:SID	SENT 信号解析のトレンド表示の Slow CH の ID を 16 進数で設定 / 問い合わせします。	5-234
:SOURce	SENT 信号解析のトレンド表示の対象チャンネルを設定 / 問い合わせします。	5-234
:UDATa	SENT 信号解析のトレンド表示の Fast CH のユーザー定義データを設定 / 問い合わせします。	5-234
:VERTical	SENT 信号解析のトレンド表示の垂直レンジを設定 / 問い合わせします。	5-234
:VTDisplay	SENT 信号解析のトレンド表示の VT 波形の表示の ON/OFF を設定 / 問い合わせします。	5-234
:SOURce?	解析 / 検索に関する全設定を問い合わせます。	5-234
:SOURce		
:{CHANnel<y> MATH<y>}?	対象波形に関する全設定を問い合わせます。	5-234
:{CHANnel<y> MATH<y>}		
:HYSteresis	対象波形のヒステリシスに関する設定 / 問い合わせをします。	5-235
:LEVel	対象波形のレベルに関する設定 / 問い合わせをします。	5-235
:SPATtern?	ユーザー定義バス信号解析 / 検索機能に関するすべての設定値を問い合わせます。	5-235
:SPATtern		
:ANALyze?	ユーザー定義バス信号解析実行に関するすべての設定値を問い合わせます。	5-235
[:ANALyze]		
:SETup?	ユーザー定義バス信号解析に関するすべての設定値を問い合わせます。	5-235
:SETup		
:BRATe	ユーザー定義バス信号解析のビットレートを設定 / 問い合わせします。	5-235
:CLOCK?	ユーザー定義バス信号解析のクロック信号のすべての設定値を問い合わせます。	5-235
:CLOCK		
:MODE	ユーザー定義バス信号解析のクロック信号の有効 / 無効を設定 / 問い合わせします。	5-235
:POLarity	ユーザー定義バス信号解析のクロック信号のスロープを設定 / 問い合わせします。	5-236
:SOURce	ユーザー定義バス信号解析のクロック信号を設定 / 問い合わせします。	5-236
:CS?	ユーザー定義バス信号解析のチップセレクト信号のすべての設定値を問い合わせます。	5-236
:CS		
:ACTive	ユーザー定義バス信号解析のチップセレクト信号のアクティブ状態を設定 / 問い合わせします。	5-236
:SOURce	ユーザー定義バス信号解析のチップセレクト信号を設定 / 問い合わせします。	5-236
:DATA?	ユーザー定義バス信号解析のデータ信号のすべての設定値を問い合わせます。	5-236
:DATA		

5.1 コマンド一覧表

コマンド	機能	ページ
:ACTive	ユーザー定義バス信号解析のデータ信号のアクティブ状態を設定 / 問い合わせします。	5-237
:SOURce	ユーザー定義バス信号解析のデータ信号を設定 / 問い合わせします。	5-237
:LATCh?	ユーザー定義バス信号解析のラッチ信号のすべての設定値を問い合わせます。	5-237
:LATCh		
:POLarity	ユーザー定義バス信号解析のラッチ信号のスロープを設定 / 問い合わせします。	5-237
:SOURce	ユーザー定義バス信号解析のラッチ信号を設定 / 問い合わせします。	5-237
:SPOint	ユーザー定義バス信号解析の解析開始点を設定 / 問い合わせします。	5-237
:SEARch?	ユーザー定義バス信号検索に関するすべての設定値を問い合わせます。	5-238
:SEARch		
:ABOrt	ユーザー定義バス信号検索を中止します。	5-238
:EXECute	ユーザー定義バス信号検索を実行します。	5-238
:SElect	ユーザー定義バス信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。	5-238
:SElect? MAXimum	ユーザー定義バス信号検索の検索点の総数を問い合わせます。	5-238
:SETup?	ユーザー定義バス信号検索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。	5-238
:SETup		
:BITSize	ユーザー定義バス信号検索のビット長を設定 / 問い合わせします。	5-238
:HEXa	ユーザー定義バス信号検索のデータ条件を 16 進数で設定します。	5-238
:PATtern	ユーザー定義バス信号検索のデータ条件を 2 進数で設定 / 問い合わせします。	5-238
:SPI?	SPI バス信号解析 / 検索機能に関するすべての設定値を問い合わせます。	5-238
:SPI		
:ANALyze?	SPI バス信号解析に関するすべての設定値を問い合わせます。	5-239
[:ANALyze]		
:SETup?	SPI バス信号解析のバス設定に関するすべての設定値を問い合わせます。	5-239
:SETup		
:BITOrder	SPI バス信号解析データのビットオーダーを設定 / 問い合わせします。	5-239
:CLOCK?	SPI バス信号解析のクロック信号のすべての設定値を問い合わせます。	5-239
:CLOCK		
:POLarity	SPI バス信号解析のクロック信号のスロープを設定 / 問い合わせします。	5-239
:SOURce	SPI バス信号解析のクロック信号を設定 / 問い合わせします。	5-239
:CS?	SPI バス信号解析のチップセレクト信号のすべての設定値を問い合わせます。	5-239
:CS		
:ACTive	SPI バス信号解析のチップセレクト信号のアクティブ状態を設定 / 問い合わせします。	5-239
:SOURce	SPI バス信号解析のチップセレクト信号を設定 / 問い合わせします。	5-240
:DATA<y>?	SPI バス信号解析のデータ信号のすべての設定値を問い合わせます。	5-240
:DATA<y>		
:SOURce	SPI バス信号解析のデータ信号を設定 / 問い合わせします。	5-240
:FIELD	SPI バス信号解析のデータの Field Size を設定 / 問い合わせします。	5-240
:GROuping	SPI バス信号解析のチップセレクト信号がない場合のグルーピングの ON/OFF を設定 / 問い合わせします。	5-240
:ITIME	SPI バス信号解析のチップセレクト信号がない場合のアイドルタイムを設定 / 問い合わせします。	5-240
:MODE	SPI バス信号解析のデータ信号の結線方式 (3 線式 / 4 線式) を設定 / 問い合わせします。	5-241
:MSBLsb	SPI バス信号解析のデータの MSB/LSB のビットを設定 / 問い合わせします。	5-241
:TYPE	SPI バス信号解析のデータのタイプを設定 / 問い合わせします。	5-241
:DETail?	SPI バス信号の解析結果リストに関するすべての設定値を問い合わせます。	5-241
:DETail		
:DISPlay	SPI バス信号の解析結果リストの表示モードを設定 / 問い合わせします。	5-241
:LIST		
:ITEM?	SPI バス信号の解析結果リストに表示される項目を問い合わせます。	5-241
:VALue?	SPI バス信号の解析結果リストの指定した解析番号の全データを問い合わせます。	5-241
:SEARch?	SPI バス信号の検索に関するすべての設定値を問い合わせます。	5-241
:SEARch		
:ABOrt	SPI バス信号検索を中止します。	5-241
:EXECute	SPI バス信号検索を実行します。	5-241

5.1 コマンド一覧表

コマンド	機能	ページ
:SElect	SPI バス信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。	5-242
:SElect? MAXimum	SPI バス信号検索の検索点の総数を問い合わせます。	5-242
:SETup?	SPI バス信号検索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。	5-242
[:SETup]		
:DATA<y>?	SPI バス信号検索の各データに関するすべての設定値を問い合わせます。	5-242
:DATA<y>		
:BCOunt	SPI バス信号検索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。	5-242
:CONDition	SPI バス信号検索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。	5-242
:DBYTe	SPI バス信号検索の各データのデータサイズ (バイト数) を設定 / 問い合わせします。	5-242
:DPATtern?	SPI バス信号検索の各データに関するすべての設定値を問い合わせます。	5-243
:DPATtern		
:HEXa	SPI バス信号検索の各データを 16 進数で設定します。	5-243
:DPATtern		
:PATtern	SPI バス信号検索の各データを 2 進数で設定 / 問い合わせします。	5-243
:MODE	SPI バス信号検索のデータ条件 (有効 / 無効) の設定 / 問い合わせします。	5-243
:TYPE	検索タイプを設定 / 問い合わせします。	5-243
:UART?	UART 信号の解析 / 検索に関するすべての設定値を問い合わせます。	5-243
:UART		
:ANALyze?	UART 信号解析に関するすべての設定値を問い合わせます。	5-243
[:ANALyze]		
:SETup?	UART 信号解析のバス設定に関するすべての設定値を問い合わせます。	5-243
:SETup		
:BITorder	UART 信号解析のビットオーダーを設定 / 問い合わせします。	5-244
:BRATe	UART 信号解析のビットレート (データ転送速度) を設定 / 問い合わせします。	5-244
:BSPace	UART 信号解析のバイトスペースを設定 / 問い合わせします。	5-244
:GROuping	UART 信号解析のグルーピングの ON/OFF を設定 / 問い合わせします。	5-244
:POLarity	UART 信号解析の極性を設定 / 問い合わせします。	5-244
:SOURce	UART 信号解析の信号を設定 / 問い合わせします。	5-244
:SPOint	UART 信号解析のサンプルポイントを設定 / 問い合わせします。	5-245
:DETail?	UART 信号の解析結果リストに関するすべての設定値を問い合わせます。	5-245
:DETail		
:DISPlay	UART 信号の解析結果リストの表示モードを設定 / 問い合わせします。	5-245
:DTIME	UART 信号の解析結果リストのグルーピングを設定 / 問い合わせします。	5-245
:LIST		
:ITEM?	UART 信号の解析結果リストに表示される項目を問い合わせます。	5-245
:VALue?	UART 信号の解析結果リストの指定した解析番号の全データを問い合わせます。	5-245
:SEARch?	UART 信号の検索に関するすべての設定値を問い合わせます。	5-245
:SEARch		
:ABORt	UART 信号検索を中止します。	5-245
:EXECute	UART 信号検索を実行します。	5-245
:SElect	UART 信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。	5-246
:SElect? MAXimum	UART 信号検索の検索点の総数を問い合わせます。	5-246
:SETup?	UART 信号検索の検索条件に関するすべての設定値を問い合わせます。	5-246
:SETup		
:DATA?	UART 信号検索のデータに関するすべての設定値を問い合わせます。	5-246
:DATA		
:ASCIi	UART 信号検索のデータを ASCII で設定します。	5-246
:CONDition	UART 信号検索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。	5-246
:CSENSitive	UART 信号検索の ASCII データの大文字、小文字を区別する / しないの設定 / 問い合わせします。	5-246
:DBYTe	UART 信号検索のデータのバイト数を設定 / 問い合わせします。	5-246
:DPATtern?	UART バス信号検索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
:DPATtern		
:HEXa	UART バス信号検索のデータを 16 進数で設定します。	5-247
:DPATtern		
:PATtern	UART 信号検索のデータを 2 進数で設定 / 問い合わせします。	5-247

5.1 コマンド一覧表

コマンド	機能	ページ
:PFormat	UART 信号検索の ID とデータ条件のデータの入力形式を設定 / 問い合わせします。	5-247
:ERRor?	UART 信号検索の Error に関するすべての設定値を問い合わせます。	5-247
:ERRor		
:FRAMing	UART 信号検索の Framing Error を設定 / 問い合わせします。	5-247
:PARity	UART 信号検索の Parity Error を設定 / 問い合わせします。	5-247
:PMODE	UART 信号検索の Parity モードを設定 / 問い合わせします。	5-248
:FORMat	UART 信号検索のフォーマットを設定 / 問い合わせします。	5-248
:MODE	UART 信号検索のモードを設定 / 問い合わせします。	5-248
:ZLINKage	シリアルバス信号解析の解析結果の解析番号とズーム位置との連動 (ON/OFF) を設定 / 問い合わせします。	5-248

SNAP グループ

:SNAP	スナップショットを実行します。	5-249
-------	-----------------	-------

SStart グループ

: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。指定時間内に STOP しなかった場合は、1 を返します。	5-250
----------	---	-------

STARt グループ

:STARt	波形の取り込みをスタートします。	5-251
--------	------------------	-------

STATus グループ

: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。	5-252
:STATus		
:CONDition?	状態レジスタの内容を問い合わせます。	5-252
:EESR?	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。	5-252
: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。	5-252
:FILTer<x>	遷移フィルターを設定 / 問い合わせします。	5-252
: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どうかを設定 / 問い合わせします。	5-252
: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。	5-252
:SPOll?	シリアルポールを実行します。	5-252

STOP グループ

:STOP	波形の取り込みをストップします。	5-253
-------	------------------	-------

STORe グループ

:STORe?	内部メモリへ保存された設定データに関するすべての情報を問い合わせます。	5-254
:STORe		
:SETup<x>?	指定された番号の内部メモリに保存された設定データに関する情報を問い合わせます。	5-254
:SETup<x>		
:COMMeNt	指定された番号の内部メモリに保存する設定データのコメントを設定 / 問い合わせします。	5-254
:DATE?	指定された番号の内部メモリに保存した設定データの日付 / 時間を問い合わせます。	5-254
:EXECute	指定された番号の内部メモリに設定データを保存します。	5-254
:LOCK	指定された番号の内部メモリに保存した設定データの保護の ON/OFF を設定 / 問い合わせします。	5-254

SYSTem グループ

:SYSTem?	システムに関するすべての設定値を問い合わせます。	5-255
:SYSTem		
:BEEP	ビーブ音を鳴らします。	5-255

5.1 コマンド一覧表

コマンド	機能	ページ
:CLICk	クリック音の ON/OFF を設定 / 問い合わせします。	5-255
:CLOCk?	日付・時刻に関するすべての設定値を問い合わせます。	5-255
:CLOCk		
:DATE	日付を設定 / 問い合わせします。	5-255
:FORMat	日付の表示順を設定 / 問い合わせします。	5-255
:MODE	日付・時刻表示の ON/OFF を設定 / 問い合わせします。	5-255
:SNTP?	SNTP による日付時刻設定を問い合わせます。	5-255
:SNTP		
:EXECute	SNTP による日付時刻を設定します。	5-255
:GMTime	グリニッジ標準時との時差を設定 / 問い合わせします。	5-255
:TIME	時刻を設定 / 問い合わせします。	5-255
:DCANcel	設定した遅延時間を時間測定値に反映する (ON)/ しない (OFF) を設定 / 問い合わせします。	5-255
:DLMSync?	同期運転に関するすべての設定値を問い合わせます。	5-256
:DLMSync		
:ADESkew	同期運転時のオートスキュー補正を実行します。	5-256
:CONNect?	同期運転を開始し、結果を問い合わせます。	5-256
:DESKew	同期運転時のスキュー補正を設定 / 問い合わせします。	5-256
:DISConnect?	同期運転を終了し、結果を問い合わせます。	5-256
:LANGuage	メッセージの言語を設定 / 問い合わせします。	5-256
:LCD?	液晶画面に関するすべての設定値を問い合わせます。	5-256
:LCD		
:AUTO?	液晶画面のバックライトのオートオフに関するすべての設定値を問い合わせます。	5-256
:AUTO		
:MODE	液晶画面のバックライトのオートオフの ON/OFF を設定 / 問い合わせします。	5-256
:TIME	液晶画面のバックライトのオートオフまでの時間を設定 / 問い合わせします。	5-256
:BRIGhtness	液晶画面の明るさを設定 / 問い合わせします。	5-256
:MODE	液晶画面のバックライトの消灯 (OFF)/ 点灯 (ON) を設定 / 問い合わせします。	5-256
:MLANguage	メニューの言語を設定 / 問い合わせします。	5-256
:MPARmeter?	波形パラメータの自動測定値・カーソル測定値に関するすべての設定値を問い合わせます。	5-257
:MPARmeter		
:FSIZE	波形パラメータの自動測定値・カーソル測定値の表示文字の大きさを設定 / 問い合わせします。	5-257
:MROWs	波形パラメータの自動測定値・カーソル測定値をメニュー表示エリアから何行表示するかを設定 / 問い合わせします。	5-257
:OCANcel	設定したオフセット電圧を測定結果や演算結果に反映する (ON)/ しない (OFF) を設定 / 問い合わせします。	5-257
:POWERoff		
:EXECute	本機器の電源を切ります。また、次に主電源スイッチを ON したときの動作を設定します。	5-257
:PPOWER		
:STATus?	プローブ電源の状態を問い合わせます。	5-257
:TOUCh	タッチパネル機能の ON/OFF を設定 / 問い合わせします。	5-257
:TOUT?	トリガ出力に関するすべての設定値を問い合わせます。	5-258
:TOUT		
:POLarity	トリガ出力の極性を設定 / 問い合わせします。	5-258
:RTIMing	トリガ出力をリセットするタイミングを設定 / 問い合わせします。	5-258
:TRANsparency?	メニューの透過モードに関するすべての設定値を問い合わせます。	5-258
:TRANsparency		
:LEVel	メニューの透過モードの透過率を設定 / 問い合わせします。	5-258
:MODE	透過モードの有効 / 無効を設定 / 問い合わせします。	5-258
:TSYNchro?	時刻同期に関するすべての設定値を問い合わせます。	5-258
:TSYNchro		
:IEEE1588?	IEEE 1588 時刻同期に関するすべての設定値を問い合わせます。	5-258
:IEEE1588		
:DNUMBER	IEEE 1588 時刻同期のドメイン番号を設定 / 問い合わせします。	5-258
:LSECond	IEEE 1588 時刻同期のうるう秒を設定 / 問い合わせします。	5-258
:NLAYer	IEEE 1588 時刻同期のネットワークレイヤーを設定 / 問い合わせします。	5-258
:PRIority<x>	IEEE 1588 時刻同期の Priority を設定 / 問い合わせします。	5-259
:PSTATe	IEEE 1588 時刻同期の状態を問い合わせます。	5-259

5.1 コマンド一覧表

コマンド	機能	ページ
:SONLy	IEEE 1588 時刻同期の Slave Only を設定 / 問い合わせします。	5-259
:STATe	IEEE 1588 時刻同期の ON/OFF を設定 / 問い合わせします。	5-259
:STATe?	本機器の時刻同期の状態を問い合わせます。	5-259
:USBKeyboard	USB キーボードの種類を設定 / 問い合わせします。	5-259

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。	5-260
:TImebase		
:SRATe?	サンプルレートを問い合わせます。	5-260
:TDIV	Time/div 値を設定 / 問い合わせします。	5-260

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。	5-261
:TRIGger		
:ABN?	A->B(N) トリガに関するすべての設定値を問い合わせます。	5-261
:ABN		
:COUNT	A->B(N) トリガの条件 B の成立回数を設定 / 問い合わせします。	5-261
: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。	5-261
: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。	5-261
: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。	5-262
:HCOPy	アクション時に、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。	5-262
:MAIL?	アクション時のメール通知に関するすべての設定値を問い合わせます。	5-262
:MAIL		
:COUNT	アクション時のメール通知のメール件数の上限を設定 / 問い合わせします。	5-262
:MODE	アクション時にメールで通知するかしないかを設定 / 問い合わせします。	5-262
: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。	5-262
:START	アクションオントリガを開始します。	5-262
:STOP	アクションオントリガを中止します。	5-262
:ADB?	A Delay B トリガに関するすべての設定値を問い合わせます。	5-262
:ADB		
:DElay	A Delay B トリガの条件 B のディレイ時間を設定 / 問い合わせします。	5-262
:{ATRigger BTRigger}?	トリガ条件に関するすべての設定値を問い合わせます。	5-263
{[:ATRigger] :BTRigger}		
:CAN?	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。	5-263
:CAN		
: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。	5-263
:EFrame?	CAN バス信号トリガの Error に関するすべての設定値を問い合わせます。	5-263
:EFrame		
:CRC	CAN バス信号トリガの CRC Error を設定 / 問い合わせします。	5-263
[:MODE]	CAN バス信号トリガの Error Frame を設定 / 問い合わせします。	5-263
:STUFF	CAN バス信号トリガの STUFF Error を設定 / 問い合わせします。	5-263
:IDData?	CAN バス信号トリガの ID とデータ条件に関するすべての設定値を問い合わせます。	5-263
[:IDData]		
:ACK?	CAN バス信号トリガの ID とデータ条件の ACK に関するすべての設定値を問い合わせます。	5-263
:ACK		
:MODE	CAN バス信号トリガの ID とデータ条件の ACK モードを設定 / 問い合わせします。	5-264
:TYPE	CAN バス信号トリガの ID とデータ条件の ACK 条件を設定 / 問い合わせします。	5-264
:DATA?	CAN バス信号トリガの ID とデータ条件のデータに関するすべての設定値を問い合わせます。	5-264
:DATA		
:CONDition	CAN バス信号トリガの ID とデータ条件のデータ判定条件を設定 / 問い合わせします。	5-264

5.1 コマンド一覧表

コマンド	機能	ページ
:DECimal<x>	CAN バス信号トリガの ID とデータ条件の判定データを 10 進数で設定します。	5-264
:DLC	CAN バス信号トリガの ID とデータ条件の有効バイト数 (DLC) を設定 / 問い合わせします。	5-264
:DPATtern?	CAN バス信号トリガの ID とデータ条件の判定データに関するすべての設定値を問い合わせます。	5-265
:DPATtern		
:HEXa	CAN バス信号トリガの ID とデータ条件の判定データを 16 進数で設定します。	5-265
:PATtern	CAN バス信号トリガの ID とデータ条件のデータを 2 進数で設定 / 問い合わせします。	5-265
:ENDian	CAN バス信号トリガの ID とデータ条件の判定データのエンディアンを設定 / 問い合わせします。	5-265
:MODE	CAN バス信号トリガの ID とデータ条件のデータ条件 (有効 / 無効) を設定 / 問い合わせします。	5-265
:MSBLsb	CAN バス信号トリガの ID とデータ条件のデータの MSB/LSB のビットを設定 / 問い合わせします。	5-265
:SIGN	CAN バス信号トリガの ID とデータ条件のデータの符号を設定 / 問い合わせします。	5-266
:IDENtifier?	CAN バス信号トリガの ID とデータ条件の識別子に関するすべての設定値を問い合わせます。	5-266
:IDENtifier		
:ID?	CAN バス信号トリガの ID とデータ条件の ID に関するすべての設定値を問い合わせます。	5-266
:ID		
:HEXa	CAN バス信号トリガの ID とデータ条件の ID を 16 進数で設定します。	5-266
:MODE	CAN バス信号トリガの ID とデータ条件の ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-266
:PATtern	CAN バス信号トリガの ID とデータ条件の ID 条件を 2 進数で設定 / 問い合わせします。	5-266
:MFORmat	CAN バス信号トリガの ID とデータ条件の ID のフレームフォーマット (標準 / 拡張) を設定 / 問い合わせします。	5-266
:PFORmat	CAN バス信号トリガの ID とデータ条件の ID の入力形式を設定 / 問い合わせします。	5-267
:MSIGnal?	CAN バス信号トリガの ID とデータ条件のメッセージシグナルに関するすべての設定値を問い合わせます。	5-267
:MSIGnal		
:MESSage		
:ITEM	CAN バス信号トリガの ID とデータ条件のメッセージのアイテムを設定します。	5-267
:SElect	CAN バス信号トリガのメッセージシグナルの条件を設定 / 問い合わせします。	5-267
:SIGNal?	CAN バス信号トリガの ID とデータ条件のシグナルに関するすべての設定値を問い合わせます。	5-267
:SIGNal		
:CONDition	CAN バス信号トリガの ID とデータ条件のシグナルのデータ条件を設定 / 問い合わせします。	5-267
:DECimal<x>	CAN バス信号トリガの ID とデータ条件のシグナルの判定データを 10 進数で設定します。	5-267
:ITEM	CAN バス信号トリガの ID とデータ条件のシグナルのアイテムを設定します。	5-268
:RTR	CAN バス信号トリガの ID とデータ条件の RTR を設定 / 問い合わせします。	5-268
:IDOR?	CAN バス信号トリガの ID OR 条件に関するすべての設定値を問い合わせます。	5-268
:IDOR		
:ACK?	CAN バス信号トリガの ID OR 条件の ACK に関するすべての設定値を問い合わせます。	5-268
:ACK		
:MODE	CAN バス信号トリガの ID OR 条件の ACK モードを設定 / 問い合わせします。	5-268
:TYPE	CAN バス信号トリガの ID OR 条件の ACK 条件を設定 / 問い合わせします。	5-268
:DATA?	CAN バス信号トリガの ID OR 条件のデータに関するすべての設定値を問い合わせます。	5-268
:DATA		
[:MODE]	CAN バス信号トリガの ID OR 条件のデータ条件 (有効 / 無効) を設定 / 問い合わせします。	5-268

5.1 コマンド一覧表

コマンド	機能	ページ
:IDENTifier?	CAN バス信号トリガの ID OR 条件の識別子に関するすべての設定値を問い合わせます。	5-268
:IDENTifier		
:ID<x>?	CAN バス信号トリガの ID OR 条件の ID に関するすべての設定値を問い合わせます。	5-268
:ID<x>		
:HEXa	CAN バス信号トリガの ID OR 条件の ID を 16 進数で設定します。	5-269
:MODE	CAN バス信号トリガの ID OR 条件の ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-269
:PATtern	CAN バス信号トリガの ID OR 条件の ID 条件を 2 進数で設定 / 問い合わせします。	5-269
:MFormat	CAN バス信号トリガの ID OR 条件の ID のフレームフォーマット (標準 / 拡張) を設定 / 問い合わせします。	5-269
:MODE	CAN バス信号トリガの ID OR 条件の識別子条件 (有効 / 無効) を設定 / 問い合わせします。	5-269
:PFormat	CAN バス信号トリガの ID OR 条件の ID の入力形式を設定 / 問い合わせします。	5-269
:MSIGnal<x>		
:MESSage		
:ITEM	CAN バス信号トリガの ID OR 条件のメッセージのアイテムを設定します。	5-270
:RTR	CAN バス信号トリガの ID OR 条件の RTR を設定 / 問い合わせします。	5-270
:MODE	CAN バス信号トリガのモードを設定 / 問い合わせします。	5-270
:REcessive	CAN バス信号トリガのリセッショングレートを設定 / 問い合わせします。	5-270
:SOURce	CAN バス信号トリガの信号を設定 / 問い合わせします。	5-270
: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。	5-270
:CANFD?	CAN FD 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。	5-270
:CANFD		
:BRate	CAN FD 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。	5-270
:DBRate	CAN FD バス信号トリガのデータフェーズのビットレート (データ転送速度) を設定 / 問い合わせします。	5-271
:DSPoint	CAN FD バス信号トリガのデータフェーズのサンプルポイントを設定 / 問い合わせします。	5-271
:EFrame?	CAN FD バス信号トリガの Error に関するすべての設定値を問い合わせます。	5-271
:EFrame		
:CRC	CAN FD バス信号トリガの CRC Error を設定 / 問い合わせします。	5-271
:CRCEFactor?	CAN FD バス信号トリガの CRC エラー要因に関するすべての設定値を問い合わせます。	5-271
:CRCEFactor		
:CRCSequence	CAN FD バス信号トリガの CRC エラー要因の CRC Sequence を設定 / 問い合わせします。	5-271
:SCount	CAN FD バス信号トリガの CRC エラー要因の Stuff Count を設定 / 問い合わせします。	5-271
:FSTuff	CAN FD バス信号トリガの固定 STUFF Error を設定 / 問い合わせします。	5-271
[:MODE]	CAN FD バス信号トリガの Error Frame を設定 / 問い合わせします。	5-272
:STUFF	CAN FD バス信号トリガの STUFF Error を設定 / 問い合わせします。	5-272
:FDF?	CAN FD バス信号トリガの FDF に関するすべての設定値を問い合わせます。	5-272
:FDF		
:CONDition	CAN FD バス信号トリガの FDF を設定 / 問い合わせします。	5-272
:FDStandard	CAN FD バス信号トリガが、ISO 標準か否かを設定 / 問い合わせします。	5-272
:IDData?	CAN FD バス信号トリガの ID/Data 条件に関するすべての設定値を問い合わせます。	5-272
[:IDData]		
:ACK?	CAN FD バス信号トリガの ID/Data 条件の ACK に関するすべての設定値を問い合わせます。	5-272
:ACK		
:MODE	CAN FD バス信号トリガの ID/Data 条件の ACK モードを設定 / 問い合わせします。	5-272
:TYPE	CAN FD バス信号トリガの ID/Data 条件の ACK 条件を設定 / 問い合わせします。	5-273
:DATA?	CAN FD バス信号トリガの ID/Data 条件のデータに関するすべての設定値を問い合わせます。	5-273
:DATA		

5.1 コマンド一覧表

コマンド	機能	ページ
:BCount	CAN FD バス信号トリガの ID/Data 条件のデータ判定条件のパターン比較する位置を設定 / 問い合わせします。	5-273
:CONDition	CAN FD バス信号トリガの ID/Data 条件のデータ判定条件を設定 / 問い合わせします。	5-273
:DBYTe	CAN FD バス信号トリガの ID/Data 条件のデータのバイト数を設定 / 問い合わせします。	5-273
:DECimal<x>	CAN FD バス信号トリガの ID/Data 条件の判定データ (a,b) を 10 進数で設定します。	5-273
:DPATtern?	CAN FD バス信号トリガの ID/Data 条件の判定データに関するすべての設定値を問い合わせます。	5-274
:DPATtern		
:HEXa	CAN FD バス信号トリガの ID/Data 条件の判定データを 16 進数で設定します。	5-274
:PATtern	CAN FD バス信号トリガの ID/Data 条件のデータを 2 進数で設定 / 問い合わせします。	5-274
:ENDian	CAN FD バス信号トリガの ID/Data 条件の判定データのエンディアンを設定 / 問い合わせします。	5-274
:MODE	CAN FD バス信号トリガの ID/Data 条件のデータ条件 (有効 / 無効) を設定 / 問い合わせします。	5-274
:MSBLsb	CAN FD バス信号トリガの ID/Data 条件のデータの MSB/LSB のビットを設定 / 問い合わせします。	5-274
:SIGN	CAN FD バス信号トリガの ID/Data 条件のデータの符号を設定 / 問い合わせします。	5-275
:IDENTifier?	CAN FD バス信号トリガの ID/Data 条件の識別子に関するすべての設定値を問い合わせます。	5-275
:IDENTifier		
:ID?	CAN FD バス信号トリガの ID/Data 条件の ID に関するすべての設定値を問い合わせます。	5-275
:ID		
:HEXa	CAN FD バス信号トリガの ID/Data 条件の ID を 16 進数で設定します。	5-275
:MODE	CAN FD バス信号トリガの ID/Data 条件の ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-275
:PATtern	CAN FD バス信号トリガの ID/Data 条件の ID 条件を 2 進数で設定 / 問い合わせします。	5-275
:MFORmat	CAN FD バス信号トリガの ID/Data 条件の ID のフレームフォーマット (標準 / 拡張) を設定 / 問い合わせします。	5-275
:PFORmat	CAN FD バス信号トリガの ID/Data 条件の ID の入力形式を設定 / 問い合わせします。	5-276
:MSIGnal?	CAN FD バス信号トリガの ID/Data 条件のメッセージシグナルに関するすべての設定値を問い合わせます。	5-276
:MSIGnal		
:MESSage		
:ITEM	CAN FD バス信号トリガの ID/Data 条件のメッセージのアイテムを設定します。	5-276
:SElect	CAN FD バス信号トリガの ID/Data 条件のメッセージシグナルの条件を設定 / 問い合わせします。	5-276
:SIGNal?	CAN FD バス信号トリガの ID/Data 条件のシグナルに関するすべての設定値を問い合わせます。	5-276
:SIGNal		
:CONDition	CAN FD バス信号トリガの ID/Data 条件のシグナルのデータ条件を設定 / 問い合わせします。	5-276
:DECimal<x>	CAN FD バス信号トリガの ID/Data 条件のシグナルの判定データを 10 進数で設定します。	5-276
:ITEM	CAN FD バス信号トリガの ID/Data 条件のシグナルのアイテムを設定します。	5-277
:RTR	CAN FD バス信号トリガの ID/Data 条件の RTR を設定 / 問い合わせします。	5-277
:IDOR?	CAN FD バス信号トリガの ID OR 条件に関するすべての設定値を問い合わせます。	5-277
:IDOR		
:ACK?	CAN FD バス信号トリガの ID OR 条件の ACK に関するすべての設定値を問い合わせます。	5-277
:ACK		
:MODE	CAN FD バス信号トリガの ID OR 条件の ACK モードを設定 / 問い合わせします。	5-277
:TYPE	CAN FD バス信号トリガの ID OR 条件の ACK 条件を設定 / 問い合わせします。	5-277

5.1 コマンド一覧表

コマンド	機能	ページ
:DATA?	CAN FD バス信号トリガの ID OR 条件のデータに関するすべての設定値を問い合わせます。	5-277
:DATA [:MODE]	CAN FD バス信号トリガの ID OR 条件のデータ条件 (有効 / 無効) を設定 / 問い合わせします。	5-277
:IDENtifier?	CAN FD バス信号トリガの ID OR 条件の識別子に関するすべての設定値を問い合わせます。	5-278
:IDENtifier :ID<x>?	CAN FD バス信号トリガの ID OR 条件の ID に関するすべての設定値を問い合わせます。	5-278
:ID<x> :HEXa	CAN FD バス信号トリガの ID OR 条件の ID を 16 進数で設定します。	5-278
:MODE	CAN FD バス信号トリガの ID OR 条件の ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-278
:PATtern	CAN FD バス信号トリガの ID OR 条件の ID 条件を 2 進数で設定 / 問い合わせします。	5-278
:MFOrmat	CAN FD バス信号トリガの ID OR 条件の ID のフレームフォーマット (標準 / 拡張) を設定 / 問い合わせします。	5-278
:MODE	CAN FD バス信号トリガの ID OR 条件の識別子条件 (有効 / 無効) を設定 / 問い合わせします。	5-278
:PFOrmat	CAN FD バス信号トリガの ID OR 条件の ID の入力形式を設定 / 問い合わせします。	5-279
:MSIGnal<x> :MESSage :ITEM	CAN FD バス信号トリガの ID OR 条件のメッセージのアイテムを設定します。	5-279
:RTR	CAN FD バス信号トリガの ID OR 条件の RTR を設定 / 問い合わせします。	5-279
:MODE	CAN FD バス信号トリガのモードを設定 / 問い合わせします。	5-279
:RECEssive	CAN FD バス信号トリガのリセツシブ電位を設定 / 問い合わせします。	5-279
:SOURce	CAN FD バス信号トリガの信号を設定 / 問い合わせします。	5-279
:SPOint	CAN FD 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。	5-279
:CXPI?	CXPI 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。	5-279
:CXPI :BRAtE	CXPI 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。	5-280
:CTOLerance	CXPI バス信号トリガのクロック許容差を設定 / 問い合わせします。	5-280
:ERROR?	CXPI バス信号トリガの Error に関するすべての設定値を問い合わせます。	5-280
:ERROR :CLOCK	CXPI バス信号トリガの Clock Error を設定 / 問い合わせします。	5-280
:CRC	CXPI バス信号トリガの CRC Error を設定 / 問い合わせします。	5-280
:DLENgth	CXPI バス信号トリガの Data Length Error を設定 / 問い合わせします。	5-280
:FRAMing	CXPI バス信号トリガの Framing Error を設定 / 問い合わせします。	5-280
:IBS	CXPI バス信号トリガの IBS Error を設定 / 問い合わせします。	5-281
:PARity	CXPI バス信号トリガの Parity Error を設定 / 問い合わせします。	5-281
:IDData?	CXPI バス信号トリガの ID とデータ条件に関するすべての設定値を問い合わせます。	5-281
[:IDData] :DATA?	CXPI バス信号トリガの ID とデータ条件のデータに関するすべての設定値を問い合わせます。	5-281
:DATA :BCOunt	CXPI バス信号トリガの ID とデータ条件のパターン比較する位置を設定 / 問い合わせします。	5-281
:CONDition	CXPI バス信号トリガの ID とデータ条件のデータ判定条件の設定 / 問い合わせします。	5-281
:DBYTe	CXPI バス信号トリガの ID とデータ条件のデータのバイト数を設定 / 問い合わせします。	5-281
:DECimal<x>	CXPI バス信号トリガの ID とデータ条件の判定データを 10 進数で設定します。	5-282
:DPATtern?	CXPI バス信号トリガの ID とデータ条件の ID に関わるすべての設定値を問い合わせます。	5-282
:DPATtern :HEXa	CXPI バス信号トリガの ID とデータ条件の ID を 16 進数で設定します。	5-282
:PATtern	CXPI バス信号トリガの ID とデータ条件の ID 条件を 2 進数で設定 / 問い合わせします。	5-282

5.1 コマンド一覧表

コマンド	機能	ページ
:ENDian	CXPI バス信号トリガの ID とデータ条件の判定データのエンディアンを設定 / 問い合わせします。	5-282
:MODE	CXPI バス信号トリガの ID とデータ条件 (有効 / 無効) を設定 / 問い合わせします。	5-282
:MSBLsb	CXPI バス信号トリガの ID とデータ条件のデータの MSB/LSB のビットを設定 / 問い合わせします。	5-283
:SIGN	CXPI バス信号トリガの ID とデータ条件のデータの符号を設定 / 問い合わせします。	5-283
:FINformation?	CXPI バス信号トリガの ID とデータ条件のフレーム情報に関するすべての設定値を問い合わせます。	5-283
:FINformation		
:CT	CXPI バス信号トリガの ID とデータ条件のフレーム情報のカウンタ値を設定 / 問い合わせします。	5-283
:MODE	CXPI バス信号トリガの ID とデータ条件のフレーム情報 (有効 / 無効) を設定 / 問い合わせします。	5-283
:SLEEP	CXPI バス信号トリガの ID とデータ条件のフレーム情報の Sleep ビットを設定 / 問い合わせします。	5-283
:WAKEup	CXPI バス信号トリガの ID とデータ条件のフレーム情報の Wakeup ビットを設定 / 問い合わせします。	5-283
:ID?	CXPI バス信号トリガの ID とデータ条件の ID に関するすべての設定値を問い合わせます。	5-284
:ID		
:HEXa	CXPI バス信号トリガの ID とデータ条件の ID を 16 進数で設定します。	5-284
:MODE	CXPI バス信号トリガの ID とデータ条件の ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-284
:PATtern	CXPI バス信号トリガの ID とデータ条件の ID を 2 進数で設定 / 問い合わせします。	5-284
:PTYPE	CXPI バス信号トリガの ID とデータ条件の PTYPE の判定条件を設定 / 問い合わせします。	5-284
:IDOR?	CXPI バス信号トリガの IDOR 条件に関するすべての設定値を問い合わせます。	5-284
:IDOR		
:IDENtifier?	CXPI バス信号トリガの IDOR 条件の識別子に関するすべての設定値を問い合わせます。	5-284
:IDENtifier		
:ID<x>?	CXPI バス信号トリガの IDOR 条件の ID に関するすべての設定値を問い合わせます。	5-284
:ID<x>		
:HEXa	CXPI バス信号トリガの IDOR 条件の ID を 16 進数で設定します。	5-284
:MODE	CXPI バス信号トリガの IDOR 条件の ID 条件 (有効 / 無効) の設定 / 問い合わせします。	5-285
:PATtern	CXPI バス信号トリガの IDOR 条件の ID 条件を 2 進数で設定 / 問い合わせします。	5-285
:MODE	CXPI バス信号トリガの IDOR 条件の識別子条件 (有効 / 無効) の設定 / 問い合わせします。	5-285
:PTYPE	CXPI バス信号トリガの IDOR 条件の PTYPE の判定条件を設定 / 問い合わせします。	5-285
:MODE	CXPI バス信号トリガの種類を設定 / 問い合わせします。	5-285
:PTYPE?	CXPI バス信号トリガの PTYPE に関するすべての設定値を問い合わせます。	5-285
:PTYPE		
:CONDition	CXPI バス信号トリガの PTYPE の判定条件を設定 / 問い合わせします。	5-285
:SOURce	CXPI バス信号トリガの信号を設定 / 問い合わせします。	5-286
:TSAMple	CXPI バス信号トリガの論理値 1/0 判定の閾値を設定 / 問い合わせします。	5-286
:WAKEupsleep?	CXPI バス信号トリガの Wakeup と Sleep に関するすべての設定値を問い合わせます。	5-286
:WAKEupsleep		
:TYPE	CXPI バス信号トリガの Wakeup と Sleep の条件を設定 / 問い合わせます。	5-286
:FLEXray?	FlexRay 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。	5-286
:FLEXray		
:BCHannel	FlexRay バス信号トリガのチャネルのバスタイプを設定 / 問い合わせします。	5-286
:BRATe	FlexRay 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。	5-286
:ERRor?	FlexRay バス信号トリガの Error に関するすべての設定値を問い合わせます。	5-286
:ERRor		

5.1 コマンド一覧表

コマンド	機能	ページ
:BSS	FlexRay バス信号トリガの BSS Error を設定 / 問い合わせします。	5-286
:CRC	FlexRay バス信号トリガの CRC Error を設定 / 問い合わせします。	5-287
:FES	FlexRay バス信号トリガの FES Error を設定 / 問い合わせします。	5-287
:IDData?	FlexRay バス信号トリガの ID とデータ条件に関するすべての設定値を問い合わせます。	5-287
[:IDData]		
:CCount?	FlexRay バス信号トリガの ID とデータ条件の Cycle Count に関するすべての設定値を問い合わせます。	5-287
:CCount		
:CONditiOn	FlexRay バス信号トリガの ID とデータ条件の Cycle Count のデータ条件を設定 / 問い合わせします。	5-287
:COUnT<x>	FlexRay バス信号トリガの ID とデータ条件の Cycle Count を設定 / 問い合わせします。	5-287
:MODE	FlexRay バス信号トリガの ID とデータ条件の Cycle Count 条件 (有効 / 無効) を設定 / 問い合わせします。	5-288
:DATA<x>?	FlexRay バス信号トリガの ID とデータ条件の Data Field に関するすべての設定値を問い合わせます。	5-288
:DATA<x>		
:BCount	FlexRay バス信号トリガの ID とデータ条件の Data Field 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。	5-288
:CONditiOn	FlexRay バス信号トリガの ID とデータ条件の Data Field のデータ条件を設定 / 問い合わせします。	5-288
:DBYTe	FlexRay バス信号トリガの ID とデータ条件の Data Field のデータのバイト数を設定 / 問い合わせします。	5-288
:DECimal<y>	FlexRay バス信号トリガの ID とデータ条件の判定データを 16 進数で設定します。	5-289
:DPATtern?	FlexRay バス信号トリガの ID とデータ条件の Data Field の判定データに関するすべての設定値を問い合わせます。	5-289
:DPATtern		
:HEXa	FlexRay バス信号トリガの ID とデータ条件の Data Field の判定データを 16 進数で設定します。	5-289
:PATtern	FlexRay バス信号トリガの ID とデータ条件の Data Field のデータを BINARY で設定 / 問い合わせします。	5-289
:ENDian	FlexRay バス信号トリガの ID とデータ条件の Data Field のエンディアンを設定 / 問い合わせします。	5-289
:MODE	FlexRay バス信号トリガの ID とデータ条件の Data Field 条件 (有効 / 無効) を設定 / 問い合わせします。	5-290
:MSBLsb	FlexRay バス信号トリガの ID とデータ条件の Data Field のデータの MSB/LSB のビットを設定 / 問い合わせします。	5-290
:SIGN	FlexRay バス信号トリガの ID とデータ条件の Data Field のデータの符号を設定 / 問い合わせします。	5-290
:FID?	FlexRay バス信号トリガの ID とデータ条件の Frame ID に関するすべての設定値を問い合わせます。	5-290
:FID		
:CONditiOn	FlexRay バス信号トリガの ID とデータ条件の Frame ID のデータ条件を設定 / 問い合わせします。	5-290
:ID<x>	FlexRay バス信号トリガの ID とデータ条件の Frame ID の値を設定 / 問い合わせします。	5-290
:MODE	FlexRay バス信号トリガの ID とデータ条件の Frame ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-291
:INDicator?	FlexRay バス信号トリガの ID とデータ条件の Indicator に関するすべての設定値を問い合わせます。	5-291
:INDicator		
:MODE	FlexRay バス信号トリガの ID とデータ条件の Indicator 条件 (有効 / 無効) を設定 / 問い合わせします。	5-291
:NFRame	FlexRay バス信号トリガの ID とデータ条件の Indicator の Null frame を設定 / 問い合わせします。	5-291
:PPReamble	FlexRay バス信号トリガの ID とデータ条件の Indicator の Payload preamble を設定 / 問い合わせします。	5-291
:STFRame	FlexRay バス信号トリガの ID とデータ条件の Indicator の Start frame を設定 / 問い合わせします。	5-291
:SYFRame	FlexRay バス信号トリガの ID とデータ条件の Indicator の Sync frame を設定 / 問い合わせします。	5-292

5.1 コマンド一覧表

コマンド	機能	ページ
:IDOR?	FlexRay 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。	5-292
:IDOR		
:ID<x>?	FlexRay バス信号トリガの OR 条件の各 Frame ID に関するすべての設定値を問い合わせます。	5-292
:ID<x>		
:CCount?	FlexRay バス信号トリガの OR 条件の各 Frame ID の Cycle Count に関するすべての設定値を問い合わせます。	5-292
:CCount		
:CONDition	FlexRay バス信号トリガの OR 条件の各 Frame ID の Cycle Count の判定条件を設定 / 問い合わせします。	5-292
:COUNT<y>	FlexRay バス信号トリガの OR 条件の各 Frame ID の Cycle Count の判定データを設定 / 問い合わせします。	5-292
:FID?	FlexRay バス信号トリガの OR 条件の各 Frame ID の ID 条件に関するすべての設定値を問い合わせます。	5-292
:FID		
:CONDition	FlexRay バス信号トリガの OR 条件の各 Frame ID の ID の判定条件を設定 / 問い合わせします。	5-293
:ID<y>	FlexRay バス信号トリガの OR 条件の各 Frame ID の ID の判定データを設定 / 問い合わせします。	5-293
:MODE	FlexRay バス信号トリガの OR 条件の各 Frame ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-293
:MODE	FlexRay バス信号トリガのモードを設定 / 問い合わせします。	5-293
:SOURce	FlexRay 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。	5-293
:I2C?	I2C 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。	5-293
:I2C		
:ADDRESS?	I2C バス信号トリガのアドレスパターンに関するすべての設定値を問い合わせます。	5-293
:ADDRESS		
:ADDRESS?	I2C バス信号トリガのアドレスパターンのアドレスに関するすべての設定値を問い合わせます。	5-293
:ADDRESS		
:BIT10Address?	I2C バス信号トリガの 10bit アドレスに関するすべての設定値を問い合わせます。	5-294
:BIT10Address		
:DIRection	I2C バス信号トリガの 10bit アドレスのデータ方向を設定 / 問い合わせします。	5-294
:HEXa	I2C バス信号トリガの 10bit アドレスを 16 進数で設定します。	5-294
:PATtern	I2C バス信号トリガの 10bit アドレスを 2 進数で設定 / 問い合わせします。	5-294
:BIT7Address?	I2C バス信号トリガの 7bit アドレスに関するすべての設定値を問い合わせます。	5-294
[:BIT7Address]		
:DIRection	I2C バス信号トリガの 7bit アドレスのデータ方向を設定 / 問い合わせします。	5-294
:HEXa	I2C バス信号トリガの 7bit アドレスを 16 進数で設定します。	5-294
:PATtern	I2C バス信号トリガの 7bit アドレスを 2 進数で設定 / 問い合わせします。	5-294
:BIT7APsub?	I2C バス信号トリガの 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。	5-295
:BIT7APsub		
:ADDRESS?	I2C バス信号トリガの 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。	5-295
:ADDRESS		
:DIRection	I2C バス信号トリガの 7bit+Sub アドレスのデータ方向を設定 / 問い合わせします。	5-295
:HEXa	I2C バス信号トリガの 7bit+Sub アドレスの 7bit アドレスを 16 進数で設定します。	5-295
:PATtern	I2C バス信号トリガの 7bit+Sub アドレスの 7bit アドレスを 2 進数で設定 / 問い合わせします。	5-295
:SADDRESS?	I2C バス信号トリガの 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。	5-295
:SADDRESS		
:HEXa	I2C バス信号トリガの 7bit+Sub アドレスの Sub アドレスを 16 進数で設定します。	5-295
:PATtern	I2C バス信号トリガの 7bit+Sub アドレスの Sub アドレスを 2 進数で設定 / 問い合わせします。	5-296

5.1 コマンド一覧表

コマンド	機能	ページ
:MODE	I2C バス信号トリガのアドレス条件の有効 / 無効を設定 / 問い合わせします。	5-296
:TYPE	I2C バス信号トリガのアドレス条件のアドレス形式を設定 / 問い合わせします。	5-296
:DATA?	I2C バス信号トリガのアドレスパターンのデータに関するすべての設定値を問い合わせます。	5-296
:DATA		
:BCount	I2C バス信号トリガのデータのパターンを比較する位置を設定 / 問い合わせします。	5-296
:BMODe	I2C バス信号トリガのデータのパターン比較する位置の ON/OFF を設定 / 問い合わせします。	5-296
:CONDition	I2C バス信号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。	5-296
:DBYTe	I2C バス信号トリガの設定データ数を設定 / 問い合わせします。	5-297
:DPATtern?	I2C バス信号トリガのデータに関する設定値をすべて問い合わせます。	5-297
:DPATtern		
:HEXa	I2C バス信号トリガのデータを 16 進数で設定します。	5-297
:PATtern	I2C バス信号トリガのデータ条件のデータを 2 進数で設定 / 問い合わせします。	5-297
:MODE	I2C バス信号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。	5-297
:GENeralcall?	I2C バス信号トリガのゼ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。	5-297
:GENeralcall		
:BIT7Maddress?	I2C バス信号トリガのゼネラルコールの 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。	5-297
:BIT7Maddress		
:HEXa	I2C バス信号トリガのゼネラルコールの 7bit マスタアドレスを 16 進数で設定します。	5-297
:PATtern	I2C バス信号トリガのゼネラルコールの 7bit マスタアドレスを 2 進数で設定 / 問い合わせします。	5-298
:SBYTe	I2C バス信号トリガのゼ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。	5-298
:INCLuderw	I2C バス信号トリガのアドレス R/W の有効 (ON) / 無効 (OFF) を設定 / 問い合わせします。	5-298
:MODE	I2C バス信号トリガのモードを設定 / 問い合わせします。	5-298
:NONAck?	I2C バス信号トリガの 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。	5-298
:NONAck		
:HSMode	I2C バス信号トリガのハイスピードモードで NONACK を無視する / しないを設定 / 問い合わせします。	5-298
:READaccess	I2C バス信号トリガのリードアクセスバイトで NON ACK を無視する / しないを設定 / 問い合わせします。	5-298
:STARTbyte	I2C バス信号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。	5-299
:SCL	I2C バス信号トリガのクロック信号を設定 / 問い合わせします。	5-299
:SDA	I2C バス信号トリガのデータ信号を設定 / 問い合わせします。	5-299
:LIN?	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ます。	5-299
:LIN		
:BLENght	LIN バス信号トリガの Break length を設定 / 問い合わせします。	5-299
: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。	5-299
:ERRor?	LIN バス信号トリガの Error に関するすべての設定値を問い合わせます。	5-299
:ERRor		
:PARity	LIN バス信号トリガの Parity Error を設定 / 問い合わせします。	5-299
:SYNCh	LIN バス信号トリガの Synch Error を設定 / 問い合わせします。	5-300
:IDData?	LIN バス信号トリガの ID とデータに関するすべての設定値を問い合わせます。	5-300
:IDData		
:DATA?	LIN 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。	5-300
:DATA		
:CONDition	LIN バス信号トリガのデータ条件を設定 / 問い合わせします。	5-300
:DBYTe	LIN バス信号トリガのデータのバイト数を設定 / 問い合わせします。	5-300
:DECimal<x>	LIN バス信号トリガのデータを 10 進数で設定します。	5-300
:DPATtern?	LIN 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。	5-300

5.1 コマンド一覧表

コマンド	機能	ページ
:DPATtern		
:HEXa	LIN バス信号トリガのデータを 16 進数で設定します。	5-300
:PATtern	LIN バス信号トリガのデータを 2 進数で設定 / 問い合わせします。	5-301
:ENDian	LIN バス信号トリガのデータのエンディアンを設定 / 問い合わせします。	5-301
:MODE	LIN バス信号トリガのデータ条件 (有効 / 無効) を設定 / 問い合わせします。	5-301
:MSBLsb	LIN 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。	5-301
:SIGN	LIN 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。	5-301
:IDENtifier?	LIN バス信号トリガの識別子に関するすべての設定値を問い合わせます。	5-301
:IDENtifier		
:ID?	LIN バス信号トリガの ID に関するすべての設定値を問い合わせます。	5-301
:ID		
:HEXa	LIN バス信号トリガの ID を 16 進数で設定します。	5-301
:MODE	LIN バス信号トリガの ID とデータ条件の ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-302
:PATtern	LIN バス信号トリガの ID を 2 進数で設定 / 問い合わせします。	5-302
:IDOR?	LIN バス信号トリガの ID OR 条件に関するすべての設定値を問い合わせます。	5-302
:IDENtifier?	LIN バス信号トリガの ID OR 条件の ID に関するすべての設定値を問い合わせます。	5-302
:IDENtifier		
:ID<x>?	LIN バス信号トリガの ID OR 条件のそれぞれの ID に関するすべての設定値を問い合わせます。	5-302
:ID<x>		
:HEXa	LIN バス信号トリガの ID OR 条件の ID を 16 進数で設定します。	5-302
:MODE	LIN バス信号トリガの ID OR 条件の ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-302
:PATtern	LIN バス信号トリガの ID OR 条件の ID 条件を 2 進数で設定 / 問い合わせします。	5-302
:MODE	LIN バス信号トリガの ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-303
:MODE	LIN バス信号トリガのモードを設定 / 問い合わせします。	5-303
:SOURce	LIN バス信号トリガの信号を設定 / 問い合わせします。	5-303
:SPoint	LIN 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。	5-303
:OR?	エッジ OR トリガに関するすべての設定値を問い合わせます。	5-303
:OR		
:ALL	すべてのチャンネルを一括して、エッジ OR トリガのトリガソースの対象にする / しないを設定 / 問い合わせします。	5-303
:CHANnel<x>	エッジ OR トリガの各チャンネルのスロープを設定 / 問い合わせします。	5-303
:PATtern?	複数入力のリ組み合わせ (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。	5-303
:PATtern		
:CHANnel<x>	複数入力のリ組み合わせ (パターン) トリガのときの各チャンネルのステートを設定 / 問い合わせします。	5-304
:CLOCK?	複数入力のリ組み合わせ (パターン) トリガの時のクロック信号に関するすべての設定値を問い合わせます。	5-304
:CLOCK		
:SLOPe	複数入力のリ組み合わせ (パターン) トリガのときのクロック信号のスロープを設定 / 問い合わせします。	5-304
:SOURce	複数入力のリ組み合わせ (パターン) トリガのときのクロック信号を設定 / 問い合わせします。	5-304
:CONDition	複数入力のリ組み合わせ (パターン) トリガのときのトリガ条件を設定 / 問い合わせします。	5-304
:LOGic	複数入力のリ組み合わせ (パターン) トリガのときの組み合わせ条件を設定 / 問い合わせします。	5-304
:{PODA PODB PODC PODD}?	複数入力のリ組み合わせ (パターン) トリガのロジック入力に関するすべての設定値を問い合わせます。	5-304
:{PODA PODB PODC PODD}		
:HEXa	複数入力のリ組み合わせ (パターン) トリガのロジック入力のステートを 16 進数で設定します。	5-305
:PATtern	複数入力のリ組み合わせ (パターン) トリガのロジック入力のパターンで設定 / 問い合わせします。	5-305
:TQQualify	複数入力のリ組み合わせ (パターン) トリガの時間条件を設定 / 問い合わせします。	5-305
:PSI5?	PSI5 信号トリガに関するすべての設定値を問い合わせます。	5-305
:PSI5		

5.1 コマンド一覧表

コマンド	機能	ページ
:DATA?	PSI5 信号のデータトリガに関するすべての設定値を問い合わせます。	5-305
:DATA		
:CONdITION	PSI5 信号のデータトリガの判定条件を設定 / 問い合わせします。	5-305
:DECimal<x>	PSI5 信号のデータトリガのデータを 10 進数で設定 / 問い合わせします。	5-305
:DPATtern?	PSI5 信号のデータ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。	5-305
:DPATtern		
:HEXa	PSI5 信号のデータトリガのデータを 16 進数で設定します。	5-306
:PATtern	PSI5 信号のデータトリガのデータを 2 進数で設定 / 問い合わせします。	5-306
:FRAMeinslot?	PSI5 信号のデータトリガのスロット指定に関するすべての設定値を問い合わせます。	5-306
:FRAMeinslot		
:MODE	PSI5 信号のデータトリガのスロット指定条件 (有効 / 無効) を設定 / 問い合わせします。	5-306
:SNUMber	PSI5 信号のデータトリガのスロット番号を設定 / 問い合わせします。	5-306
:ERRor?	PSI5 信号の Error トリガに関するすべての設定値を問い合わせます。	5-306
:ERRor		
:FNUMber	PSI5 信号の Frame Number Error トリガを設定 / 問い合わせします。	5-306
:FRAMe	PSI5 信号の Frame Error トリガを設定 / 問い合わせします。	5-306
:MANChester	PSI5 信号の Manchester Error トリガを設定 / 問い合わせします。	5-307
:PCRC	PSI5 信号の Parity/CRC Error トリガを設定 / 問い合わせします。	5-307
:SBIT	PSI5 信号の Start Bit Error トリガを設定 / 問い合わせします。	5-307
:SBoundary	PSI5 信号の Slot Boundary Error トリガを設定 / 問い合わせします。	5-307
:FRAMeinslot?	PSI5 信号のスロット指定トリガのすべての設定値を問い合わせます。	5-307
:FRAMeinslot		
:SNUMber	PSI5 信号のスロット指定トリガのスロット番号を設定 / 問い合わせします。	5-307
:MODE	PSI5 信号トリガの種類を設定 / 問い合わせします。	5-307
:SETup?	PSI5 信号トリガのバス設定に関するすべての設定値を問い合わせます。	5-308
:SETup		
:DATA?	PSI5 信号トリガのデータ信号に関するすべての設定値を問い合わせます。	5-308
:DATA		
:BRATe	PSI5 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。	5-308
:CTOLerance	PSI5 信号トリガのクロック許容範囲を設定 / 問い合わせします。	5-308
:DBITs	PSI5 信号トリガのデータ長を設定 / 問い合わせします。	5-308
:EDETection	PSI5 信号トリガのエラー検出方式を設定 / 問い合わせします。	5-308
:SNRejection?	PSI5 信号トリガの同期ノイズ除去に関するすべての設定値を問い合わせます。	5-308
:SNRejection		
:ETIMe	PSI5 信号トリガの同期ノイズ除去の終了時間を設定 / 問い合わせします。	5-309
:MODE	PSI5 信号トリガの同期ノイズ除去の ON/OFF を設定 / 問い合わせします。	5-309
:SOURce	PSI5 信号トリガのデータソースを設定 / 問い合わせします。	5-309
:NUMBerofslot	PSI5 信号トリガのスロット数を設定 / 問い合わせします。	5-309
:SLOT<x>	PSI5 信号トリガのスロットの開始 / 終了時間を設定 / 問い合わせします。	5-309
:SYNC	PSI5 信号トリガの同期信号を設定 / 問い合わせします。	5-309
:SENT?	SENT 信号トリガに関するすべての設定値を問い合わせます。	5-310
:SENT		
:CEFactor?	SENT 信号トリガのエラー要因に関するすべての設定値を問い合わせます。	5-310
:CEFactor		
:SAComm?	SENT 信号トリガの Status and Communication のエラー要因に関するすべての設定値を問い合わせます。	5-310
:SAComm		
:BIT<x>	SENT 信号トリガの Status and Communication のエラー要因のビットを設定 / 問い合わせします。	5-310
:SCPulses	SENT 信号トリガの連続キャリブレーションパルスのエラー要因を設定 / 問い合わせします。	5-310
:CRCType	SENT 信号トリガの CRC 演算方式を設定 / 問い合わせします。	5-310
:CTICk	SENT 信号トリガのクロックティック値を設定 / 問い合わせします。	5-310
:CTOLerance	SENT 信号トリガのクロック許容差を設定 / 問い合わせします。	5-311
:DNIBbles	SENT 信号トリガのデータニブル数を設定 / 問い合わせします。	5-311
:DTYPe	SENT 信号トリガの Fast CH のデータ形式を設定 / 問い合わせします。	5-311
:ERRor?	SENT 信号トリガの Error に関するすべての設定値を問い合わせます。	5-311
:ERRor		

5.1 コマンド一覧表

コマンド	機能	ページ
:FCRC	SENT 信号トリガの Fast CH の CRC エラーを設定 / 問い合わせします。	5-311
:NDValue	SENT 信号トリガのニブルデータバリュエラーを設定 / 問い合わせします。	5-311
:NNUmber	SENT 信号トリガのニブルデータ数エラーを設定 / 問い合わせします。	5-311
:SAComm	SENT 信号トリガの Status and Communication エラーを設定 / 問い合わせします。	5-312
:SCPulses	SENT 信号トリガの連続キャリブレーションパルスエラーを設定 / 問い合わせします。	5-312
:SCRC	SENT 信号トリガの Slow CH の CRC エラーを設定 / 問い合わせします。	5-312
:FData?	SENT 信号トリガの Fast CH のデータに関するすべての設定値を問い合わせします。	5-312
:FData		
:DATA<x>?	SENT 信号トリガの Fast CH の各ユーザーデータに関するすべての設定値を問い合わせします。	5-312
:DATA<x>		
:CONDition	SENT 信号トリガの Fast CH の各ユーザーデータの判定条件を設定 / 問い合わせします。	5-312
:DECimal<y>	SENT 信号トリガの Fast CH の各ユーザーデータのデータを 10 進数で設定 / 問い合わせします。	5-312
:MODE	SENT 信号トリガの Fast CH の各ユーザーデータのデータ条件 (有効 / 無効) を設定 / 問い合わせします。	5-313
:DNIBbles?	SENT 信号トリガの Fast CH のニブルデータに関するすべての設定値を問い合わせします。	5-313
:DNIBbles		
:CONDition	SENT 信号トリガの Fast CH のニブルデータの判定条件を設定 / 問い合わせします。	5-313
:HEXa	SENT 信号トリガの Fast CH のニブルデータのデータを 16 進数で設定します。	5-313
:PATtern	SENT 信号トリガの Fast CH のニブルデータのデータを 2 進数で設定 / 問い合わせします。	5-313
:FSAComm?	SENT 信号トリガの Fast CH の Status and Communication ニブルに関するすべての設定値を問い合わせします。	5-313
:FSAComm		
:HEXa	SENT 信号トリガの Fast CH の Status and Communication ニブルのデータを 16 進数で設定します。	5-313
:PATtern	SENT 信号トリガの Fast CH の Status and Communication ニブルのデータを 2 進数で設定します。	5-313
:MODE	SENT 信号トリガのモードを設定 / 問い合わせします。	5-314
:PPULse	SENT 信号トリガのポーズパルスの有無を設定 / 問い合わせします。	5-314
:SDa?	SENT 信号トリガの Slow CH のデータに関するすべての設定値を問い合わせします。	5-314
:SDa		
:ENHanced?	SENT 信号トリガの Slow CH の Enhanced タイプに関するすべての設定値を問い合わせします。	5-314
:ENHanced		
:CBIT	SENT 信号トリガの Slow CH の Enhanced タイプの Configuration ビットを設定 / 問い合わせします。	5-314
:D12Bit?	SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID に関するすべての設定値を問い合わせします。	5-314
:D12Bit		
:DATA?	SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータに関するすべての設定値を問い合わせします。	5-314
:DATA		
:CONDition	SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータの判定条件を設定 / 問い合わせします。	5-314
:DECimal<x>	SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータを 10 進数で設定 / 問い合わせします。	5-315
:HEXa	SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータを 16 進数で設定します。	5-315
:MODE	SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータ条件 (有効 / 無効) を設定 / 問い合わせします。	5-315
:PATtern	SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータを 2 進数で設定 / 問い合わせします。	5-315
:ID?	SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID の ID に関するすべての設定値を問い合わせします。	5-315

5.1 コマンド一覧表

コマンド	機能	ページ
:ID		
:CONDition	SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID の ID 判定条件を設定 / 問い合わせします。	5-315
:DECimal<x>	SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID の ID を 10 進数で設定します。	5-316
:MODE	SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID の ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-316
:PFORmat	SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータの入力形式を設定します。	5-316
:Dl6Bit?	SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID に関するすべての設定値を問い合わせます。	5-316
:Dl6Bit		
:DATA?	SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータに関するすべての設定値を問い合わせます。	5-316
:DATA		
:CONDition	SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータの判定条件を設定 / 問い合わせします。	5-316
:DECimal<x>	SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータを 10 進数で設定 / 問い合わせします。	5-317
:HEXa	SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータを 16 進数で設定します。	5-317
:MODE	SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータ条件 (有効 / 無効) を設定 / 問い合わせします。	5-317
:PATtern	SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータを 2 進数で設定 / 問い合わせします。	5-317
:ID?	SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID に関するすべての設定値を問い合わせます。	5-317
:ID		
:CONDition	SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID 判定条件を設定 / 問い合わせします。	5-317
:DECimal<x>	SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID を 10 進数で設定します。	5-318
:MODE	SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-318
:PFORmat	SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータの入力形式を設定します。	5-318
:SHORT?	SENT 信号トリガの Slow CH の Short タイプに関するすべての設定値を問い合わせます。	5-318
:SHORT		
:DATA?	SENT 信号トリガの Slow CH の Short タイプのデータに関するすべての設定値を問い合わせます。	5-318
:DATA		
:CONDition	SENT 信号トリガの Slow CH の Short タイプのデータの判定条件を設定 / 問い合わせします。	5-318
:DECimal<x>	SENT 信号トリガの Slow CH の Short タイプのデータを 10 進数で設定 / 問い合わせします。	5-318
:HEXa	SENT 信号トリガの Slow CH の Short タイプのデータを 16 進数で設定します。	5-319
:MODE	SENT 信号トリガの Slow CH の Short タイプのデータ条件 (有効 / 無効) を設定 / 問い合わせします。	5-319
:PATtern	SENT 信号トリガの Slow CH の Short タイプのデータを 2 進数で設定 / 問い合わせします。	5-319
:ID?	SENT 信号トリガの Slow CH の Short タイプの ID に関するすべての設定値を問い合わせます。	5-319
:ID		
:CONDition	SENT 信号トリガの Slow CH の Short タイプの ID 判定条件を設定 / 問い合わせします。	5-319
:DECimal<x>	SENT 信号トリガの Slow CH の Short タイプの ID を 10 進数で設定します。	5-319
:MODE	SENT 信号トリガの Slow CH の Short タイプの ID 条件 (有効 / 無効) を設定 / 問い合わせします。	5-319
:PFORmat	SENT 信号トリガの Slow CH の Short タイプのデータの入力形式を設定します。	5-320
:SOURce	SENT 信号トリガのソースを設定 / 問い合わせします。	5-320
:STYPe	SENT 信号トリガの Slow CH 形式を設定 / 問い合わせします。	5-320

5.1 コマンド一覧表

コマンド	機能	ページ
:USEUp?	SENT 信号トリガの Fast CH のユーザー定義データに関するすべての設定値を問い合わせます。	5-320
:USEUp		
:DATA<x>?	SENT 信号トリガの Fast CH の各ユーザー定義データに関するすべての設定値を問い合わせます。	5-320
:DATA<x>		
:MODE	SENT 信号トリガの Fast CH の各ユーザー定義データの有効 / 無効を設定 / 問い合わせします。	5-320
:ORDer	SENT 信号トリガの Fast CH の各ユーザー定義データのエンディアンを設定 / 問い合わせします。	5-320
:SIZE	SENT 信号トリガの Fast CH の各ユーザー定義データのデータサイズを設定 / 問い合わせします。	5-321
:MULTiplexing	SENT 信号トリガの FastCH のユーザー定義データで、Multiplexing の有効 / 無効を設定 / 問い合わせします。	5-321
:VERsion	SENT 信号トリガの仕様バージョンを設定 / 問い合わせします。	5-321
:SIMPlE?	エッジトリガに関するすべての設定値を問い合わせます。	5-321
:SIMPlE		
:HFRejection	エッジトリガのトリガソースのローパスフィルター (HF リジェクション) の ON/OFF を設定 / 問い合わせします。	5-321
:LEVeL	エッジトリガのトリガソースのトリガレベルを設定 / 問い合わせします。	5-321
:NREJection	エッジトリガのトリガソースのレベルのノイズリジェクションを設定 / 問い合わせします。	5-322
:PROBe	エッジトリガの外部トリガソースの、プローブを設定 / 問い合わせします。	5-322
:RANge	エッジトリガの外部トリガソースの入力レンジを設定 / 問い合わせします。	5-322
:SLOPe	エッジトリガのトリガソースのトリガスロープを設定 / 問い合わせします。	5-322
:SOURce	エッジトリガのトリガソースを設定 / 問い合わせします。	5-322
:SPATtern?	ユーザー定義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。	5-322
:SPATtern		
:BITSize	ユーザー定義バス信号トリガのビット長を設定 / 問い合わせします。	5-323
:BRATe	ユーザー定義バス信号トリガのビットレートを設定 / 問い合わせします。	5-323
:CLOCK?	ユーザー定義バス信号トリガのクロック信号に関するすべての設定値を問い合わせます。	5-323
:CLOCK		
:MODE	ユーザー定義バス信号トリガのクロック信号を使用する / しないを設定 / 問い合わせします。	5-323
:POLarity	ユーザー定義バス信号トリガのクロック信号の極性を設定 / 問い合わせします。	5-323
:SOURce	ユーザー定義バス信号トリガのクロック信号を設定 / 問い合わせします。	5-323
:CS?	ユーザー定義バス信号トリガのチップセレクト信号のすべての設定値を問い合わせます。	5-323
:CS		
:ACTive	ユーザー定義バス信号トリガのチップセレクト信号のアクティブ状態を設定 / 問い合わせします。	5-324
:SOURce	ユーザー定義バス信号トリガのチップセレクト信号を設定 / 問い合わせします。	5-324
:DATA?	ユーザー定義バス信号トリガのデータ信号のすべての設定値を問い合わせます。	5-324
:DATA		
:ACTive	ユーザー定義バス信号トリガのデータ信号のアクティブ状態を設定 / 問い合わせします。	5-324
:SOURce	ユーザー定義バス信号トリガのデータ信号を設定 / 問い合わせします。	5-324
:HEXa	ユーザー定義バス信号トリガのデータ条件を 16 進数で設定します。	5-324
:LATCh?	ユーザー定義バス信号トリガのラッチ信号のすべての設定値を問い合わせます。	5-324
:LATCh		
:POLarity	ユーザー定義バス信号トリガのラッチ信号の極性を設定 / 問い合わせします。	5-325
:SOURce	ユーザー定義バス信号トリガのラッチ信号を設定 / 問い合わせします。	5-325
:PATtern	ユーザー定義バス信号トリガのデータ条件を 2 進数で設定 / 問い合わせします。	5-325
:SPI?	SPI 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。	5-325
:SPI		
:BITOrder	SPI バス信号トリガデータのビットオーダーを設定 / 問い合わせします。	5-325

5.1 コマンド一覧表

コマンド	機能	ページ
:CLOCK?	SPI バス信号トリガのクロック信号のすべての設定値を問い合わせます。	5-325
:CLOCK		
:POLarity	SPI バス信号トリガのクロック信号の極性を設定 / 問い合わせします。	5-325
:SOURce	SPI バス信号トリガのクロック信号を設定 / 問い合わせします。	5-325
:CS?	SPI バス信号トリガのチップセレクト信号のすべての設定値を問い合わせます。	5-326
:CS		
:ACTive	SPI バス信号トリガのチップセレクト信号のアクティブ状態を設定 / 問い合わせします。	5-326
:SOURce	SPI バス信号トリガのチップセレクト信号を設定 / 問い合わせします。	5-326
:DATA<x>?	SPI バス信号トリガのデータ信号のすべての設定値を問い合わせます。	5-326
:DATA<x>		
:BCount	SPI バス信号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。	5-326
:CONDition	SPI バス信号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。	5-326
:DBYTe	SPI バス信号トリガの各データのデータサイズ (バイト数) を設定 / 問い合わせします。	5-326
:DPATtern?	SPI バス信号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。	5-326
:DPATtern		
:HEXa	SPI バス信号トリガの各データを 16 進数で設定します。	5-327
:PATtern	SPI バス信号トリガの各データを 2 進数で設定 / 問い合わせします。	5-327
:MODE	SPI バス信号トリガのデータ信号の有効 (ON)/ 無効 (OFF) を設定 / 問い合わせします。	5-327
:SOURce	SPI バス信号トリガのデータ信号を設定 / 問い合わせします。	5-327
:MODE	SPI バス信号トリガの結線方式 (3 線式 / 4 線式) を設定 / 問い合わせします。	5-327
:TYPE	トリガの種類を設定 / 問い合わせします。	5-327
:UART?	UART 信号トリガに関するすべての設定値を問い合わせます。	5-328
:UART		
:BITorder	UART 信号トリガのビットオーダーを設定 / 問い合わせします。	5-328
: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。	5-328
:DATA?	UART 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。	5-328
:DATA		
:ASCIi	UART 信号トリガのデータを ASCII で設定します。	5-328
:CONDition	UART 信号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。	5-328
:CSensitive	UART 信号トリガの ASCII データの大文字、小文字を区別する / しないの設定 / 問い合わせします。	5-328
:DBYTe	UART 信号トリガのデータのバイト数を設定 / 問い合わせします。	5-328
:DPATtern?	UART 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。	5-328
:DPATtern		
:HEXa	UART 信号トリガのデータを 16 進数で設定します。	5-329
:PATtern	UART 信号トリガのデータを 2 進数で設定 / 問い合わせします。	5-329
:PFORMAT	UART 信号トリガの ID とデータ条件のデータの入力形式を設定 / 問い合わせします。	5-329
:ERRor?	UART 信号トリガの Error に関するすべての設定値を問い合わせます。	5-329
:ERRor		
:FRAMing	UART 信号トリガの Framing Error を設定 / 問い合わせします。	5-329
:PARity	UART 信号トリガの Parity Error を設定 / 問い合わせします。	5-329
:PMODE	UART 信号トリガの Parity モードを設定 / 問い合わせします。	5-329
: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。	5-330
:MODE	UART 信号トリガのトリガの種類を設定 / 問い合わせします。	5-330
:POLarity	UART 信号解析の極性を設定 / 問い合わせします。	5-330
:SOURce	UART 信号解析の信号を設定 / 問い合わせします。	5-330
:SPOint	UART 信号解析のサンプルポイントを設定 / 問い合わせします。	5-330
:WINDow?	ウィンドウトリガのすべての設定値を問い合わせます。	5-330
:WINDow		
:POLarity	ウィンドウトリガのトリガソースの極性を設定 / 問い合わせします。	5-330
:SOURce	ウィンドウトリガのトリガソースを設定 / 問い合わせします。	5-330
:TQQualify	ウィンドウトリガの時間条件を設定 / 問い合わせします。	5-331
:WOR?	ウィンドウ OR トリガに関するすべての設定値を問い合わせます。	5-331
:WOR		

5.1 コマンド一覧表

コマンド	機能	ページ
:ALL	すべてのチャンネルを一括して、ウィンドウ OR トリガのトリガソースの対象にする / しないを設定 / 問い合わせします	5-331
:CHANnel	ウィンドウ OR トリガの各チャンネルのスロープを設定 / 問い合わせします。	5-331
[:ATRigger]		
:INTerval?	インターバルトリガに関するすべての設定値を問い合わせます。	5-331
:INTerval		
:SLOPe	インターバルトリガのトリガスロープを設定 / 問い合わせします。	5-331
:SOURce	インターバルトリガのトリガソースを設定 / 問い合わせします。	5-331
:TQQualify	インターバルトリガの時間条件を設定 / 問い合わせします。	5-331
:PULSe?	パルス幅トリガに関するすべての設定値を問い合わせます。	5-331
:PULSe		
:POLarity	パルス幅トリガのトリガソースの極性を設定 / 問い合わせします。	5-332
: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。	5-332
:TQQualify	パルス幅トリガの時間条件を設定 / 問い合わせします。	5-332
:RFTime?	立ち上がり / 立ち下がり時間トリガのすべての設定値を問い合わせます。	5-332
:RFTime		
:SLOPe	立ち上がり / 立ち下がり時間トリガのトリガスロープを設定 / 問い合わせします。	5-332
:SOURce	立ち上がり / 立ち下がり時間トリガのトリガソースを設定 / 問い合わせします。	5-332
:TQQualify	立ち上がり / 立ち下がり時間トリガの時間条件を設定 / 問い合わせします。	5-332
:RUNT?	ラントトリガに関するすべての設定値を問い合わせます。	5-332
:RUNT		
:POLarity	ラントトリガのトリガソースの極性を設定 / 問い合わせします。	5-332
:SOURce	ラントトリガのトリガソースを設定 / 問い合わせします。	5-332
:TQQualify	ラントトリガの時間条件を設定 / 問い合わせします。	5-333
:TImeout?	タイムアウトトリガのすべての設定値を問い合わせます。	5-333
:TImeout		
:POLarity	タイムアウトトリガのトリガソースの極性を設定 / 問い合わせします。	5-333
:SOURce	タイムアウトトリガのトリガソースを設定 / 問い合わせします。	5-333
:TV?	TV トリガに関するすべての設定値を問い合わせます。	5-333
:TV		
:FIEld	TV トリガをかけるフィールドを設定 / 問い合わせします。	5-333
: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。	5-333
: {HDTV NTSC PAL SDTV USE Rdefine}?	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。	5-333
: {HDTV NTSC PAL SDTV USE Rdefine}		
:LINE	TV トリガをかけるラインを設定 / 問い合わせします。	5-333
:POLarity	TV トリガの入力の極性を設定 / 問い合わせします。	5-334
:LEVel	TV トリガのトリガレベルを設定 / 問い合わせします。	5-334
:LFORmat	TV トリガをかけるラインの指定方式を設定 / 問い合わせします。	5-334
:SOURce	TV トリガのトリガソースを設定 / 問い合わせします。	5-334
:TYPE	TV トリガの入力の種類を設定 / 問い合わせします。	5-334
:USERdefine?	ユーザー定義 TV トリガに関するすべての設定値を問い合わせます。	5-334
:USERdefine		
:DEFinition	ユーザー定義 TV トリガの解像度を設定 / 問い合わせします。	5-334
:HFRejection	ユーザー定義 TV トリガ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。	5-334
:HSYNc	ユーザー定義 TV トリガの水平同期信号の周波数を設定 / 問い合わせします。	5-335
:SGUard	ユーザー定義 TV トリガの Sync Guard を設定 / 問い合わせします。	5-335
:COMBination	トリガの組み合わせを設定 / 問い合わせします。	5-335
:DELaY?	ディレイに関するすべての設定値を問い合わせます。	5-335
:DELaY		
:TIME	ディレイ (トリガ点からトリガポジションまでの時間) を設定 / 問い合わせします。	5-335
:DETect?	トリガがかかったかどうかを問い合わせます。	5-335
:FORCe	強制的にトリガをかけます。	5-335
:HOLDoff?	ホールドオフに関するすべての設定値を問い合わせます。	5-336
:HOLDoff		
:TIME	ホールドオフ時間を設定 / 問い合わせします。	5-336
:MODE	トリガモードを設定 / 問い合わせします。	5-336
:POSition	トリガポジションを設定 / 問い合わせします。	5-336

コマンド	機能	ページ
:SCount	トリガモードがN Singleのときのトリガ成立回数を設定 / 問い合わせします。	5-336
:SOURce?	ENHANCED トリガの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。	5-336
:SOURce		
:CHANnel<x>?	ENHANCED トリガの指定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。	5-336
:CHANnel<x>		
:HFRejection	ENHANCED トリガのときの指定トリガソースのローパスフィルタを設定 / 問い合わせします。	5-336
:LEVel	ENHANCED トリガのときの指定トリガソースのトリガレベルを設定 / 問い合わせします。	5-336
:NREJection	ENHANCED トリガのときの指定トリガソースのノイズリジェクションを設定 / 問い合わせします。	5-337
:ULLevel	ENHANCED トリガのトリガソースのトリガレベルの上下限值を設定 / 問い合わせします。	5-337
:TIME<x>	ENHANCED トリガのときの時間を設定 / 問い合わせします。	5-337

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。	5-338
:WAVeform		
:ALL?	「:WAVeform:ALL:SEND?」に関するすべての設定値を問い合わせます。	5-338
:ALL		
:SEND?	「:WAVeform:ALL:TRACe」で指定した波形データを問い合わせます。	5-338
:TRACe	「:WAVeform:ALL:SEND?」で対象となる波形を設定 / 問い合わせます。	5-339
: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。	5-339
:BYTeorder	波形データが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。	5-339
: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。	5-339
:FORMat	送信する波形データのフォーマットを設定 / 問い合わせします。	5-339
: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。	5-339
: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。	5-339
: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。	5-339
: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。	5-339
: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。	5-339
: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ます。	5-340
:SEND?	「:WAVeform:TRACe」で指定した波形データを問い合わせます。	5-340
: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。	5-340
:SRATe?	「: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。	5-340
: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。	5-340
: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。	5-340
: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。	5-340
: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。	5-340

WPARameter グループ

: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。	5-341
:WPARameter<x>		
:DISPlay	波形パラメータ測定の表示の ON/OFF を設定 / 問い合わせします。	5-341
: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わせます。	5-341
: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を問い合わせます。	5-341
:MEASure		

5.1 コマンド一覧表

コマンド	機能	ページ
:MODE	波形パラメータ測定のはistogram表示の自動測定モードを設定 / 問い合わせします。	5-341
:PARAmeter?	波形パラメータ測定のは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。	5-341
:PARAmeter		
:ALL	波形パラメータ測定のはistogramパラメータを一斉に ON/OFF します。	5-341
:<パラメータ>?	波形パラメータ測定のはistogramパラメータに関するすべての設定値を問い合わせます。	5-341
:<パラメータ>		
:STATe	波形パラメータ測定のはistogramパラメータの ON/OFF を設定 / 問い合わせします。	5-341
:VALue?	波形パラメータ測定のはistogramパラメータの測定値を問い合わせます。	5-342
:POSition<y>	波形パラメータ測定のはistogramの各パラメータの位置を設定 / 問い合わせします。	5-342
:ITEM	波形パラメータ測定のはistogramパラメータを設定 / 問い合わせします。	5-342
:MODE	波形パラメータ測定のはistogramモードを設定 / 問い合わせします。	5-342
:TREND?	波形パラメータ測定のはistogram表示に関するすべての設定値を問い合わせます。	5-343
:TREND		
:AScale	波形パラメータ測定のはistogram表示のオートスケールを実行します。	5-343
:CURSor?	波形パラメータ測定のはistogram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。	5-343
:CURSor		
:C<y>?	波形パラメータ測定のはistogramの各カーソルの測定値を問い合わせます。	5-343
:DC?	波形パラメータ測定のはistogramのカーソル間の測定値を問い合わせます。	5-343
:MODE	波形パラメータ測定のはistogramの自動測定のはistogramモードを設定 / 問い合わせします。	5-343
:POSition<y>	波形パラメータ測定のはistogramの各カーソルの位置を設定 / 問い合わせします。	5-343
:HRANge	波形パラメータ測定のはistogram表示の対象ウィンドウを設定 / 問い合わせします。	5-343
:HSPan	波形パラメータ測定のはistogram表示の水平スパンを設定 / 問い合わせします。	5-343
:VERTical	波形パラメータ測定のはistogramの垂直レンジを設定 / 問い合わせします。	5-344
:VTDisplay	VT 波形の表示の ON/OFF を設定 / 問い合わせします。	5-344
:WAIT?	タイムアウト付きで自動測定実行の終了を待ちます。	5-344

XY グループ

:XY<x>?	XY 表示に関するすべての設定値を問い合わせます。	5-345
:XY<x>		
:DISPlay	XY 表示を画面に表示する (ON)/ しない (OFF) を設定 / 問い合わせします。	5-345
: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。	5-345
: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。	5-345
:CURSor		
:DX?	X-Y 表示の水平カーソル間の電圧差に関するすべての設定値を問い合わせます。	5-345
:DX		
:STATe	X-Y 表示の水平カーソル間の電圧差の ON/OFF を設定 / 問い合わせします。	5-345
:VALue?	X-Y 表示の水平カーソル間の電圧差を問い合わせます。	5-345
:DY?	X-Y 表示の垂直カーソル間の電圧差に関するすべての設定値を問い合わせます。	5-345
:DY		
:STATe	X-Y 表示の垂直カーソル間の電圧差の ON/OFF を設定 / 問い合わせします。	5-345
:VALue?	X-Y 表示の垂直カーソル間の電圧差を問い合わせます。	5-345
:X<y>?	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。	5-345
:X<y>		
:POSition	XY 表示の水平カーソルの位置を設定 / 問い合わせます。	5-346
:STATe	X-Y 表示の水平カーソルの電圧値の ON/OFF を設定 / 問い合わせします。	5-346
:VALue?	XY 表示の水平カーソルの電圧値を問い合わせます。	5-346
:Y<y>?	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。	5-346
:Y<y>		

コマンド	機能	ページ
:POSition	XY 表示の垂直カーソルの位置を設定 / 問い合わせます。	5-346
:STATe	X-Y 表示の垂直カーソルの電圧値の ON/OFF を設定 / 問い合わせします。	5-346
:VALue?	XY 表示の垂直カーソルの電圧値を問い合わせます。	5-346
:INTeg?	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。	5-346
:INTeg		
:LOOP	XY 表示の積分方法を設定 / 問い合わせます。	5-346
:POLarity	XY 表示の積分方向を設定 / 問い合わせます。	5-347
:VALue?	XY 表示の積分値を問い合わせます。	5-347
:MODE	XY 表示の自動測定モードを設定 / 問い合わせます。	5-347
:RANGe	X-Y 表示する T-Y 波形の対象ウインドウを設定 / 問い合わせします。	5-347
:SPLit	XY 表示で XY1/XY2 の表示画面を分割する / しないを、設定 / 問い合わせます。	5-347
:TRANge	XY 表示する T-Y 波形の範囲を設定 / 問い合わせします。	5-347
:VTDisplay	XY 表示の VT 波形の表示の ON/OFF を設定 / 問い合わせします。	5-347
:XTRace	XY 表示の X 軸に割り当てるチャンネルを設定 / 問い合わせします。	5-347
:YTRace	XY 表示の Y 軸に割り当てるチャンネルを設定 / 問い合わせします。	5-347

ZOOM グループ

:ZOOM<x>?	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。	5-348
:ZOOM<x>		
:ALLocation?	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。	5-348
:ALLocation		
: {CHANnel<y> LOGic MATH<y>}	ズーム対象波形を設定 / 問い合わせします。	5-348
:DISPlay	ズーム波形表示の ON/OFF を設定 / 問い合わせします。	5-348
:FORMat	ズーム波形表示フォーマットを設定 / 問い合わせします。	5-348
:MAG	対象ズーム波形表示画面の倍率を設定 / 問い合わせします。	5-348
:MAGFine	対象ズーム波形表示画面の倍率 (Fine) を設定 / 問い合わせします。	5-348
:MAIN	ズーム波形表示の表示比率を設定 / 問い合わせします。	5-348
:POSition	対象ズームボックスの位置を設定 / 問い合わせします。	5-348
:VERTical?	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。	5-348
:VERTical		
: {CHANnel<y> MATH<y>}	垂直軸方向のズーム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。	5-349
: {CHANnel<y> MATH<y>}		
:MAG	垂直軸方向のズーム倍率を設定 / 問い合わせします。	5-349
: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。	5-349
:RESet	垂直軸方向のズームに関するすべての設定値をリセットします。	5-349
:TRACe	垂直軸方向のズーム画面に表示するトレースを設定 / 問い合わせします。	5-349

共通コマンドグループ

*CAL	キャリブレーションを実行し、結果を問い合わせます。	5-350
*CLS	標準イベントレジスタ、拡張イベントレジスタ、エラーキューをクリアします。	5-350
*ESE	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。	5-350
*ESR?	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。	5-350
*IDN?	機種を問い合わせます。	5-350
*OPC	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。	5-350
*OPC?	*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。	5-351
*OPT?	装備しているオプションを問い合わせます。	5-351
*RST	設定の初期化 (イニシャライズ) をします。	5-351
*SRE	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。	5-351
*STB?	ステータスバイトレジスタの値を問い合わせます。	5-351
*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 5-352 部の各メモリテストです。	5-352
*WAI	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。	5-352

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?

: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?

:ACQuire:AVERage:COUNT

機能 アベレーシングモード時の、指数化平均の減衰定数、または単純平均のアベレージ回数を設定 / 問い合わせします。

構文 :ACQuire:AVERage:COUNT {<NRf>}

<NRf> = 2 ~ 1024、2n ステップ

例 :ACQUIRE:AVERAGE:COUNT 2

:ACQUIRE:AVERAGE:COUNT?

-> :ACQUIRE:AVERAGE:COUNT 2

:ACQuire:COUNT

機能 ノーマルモード・エンベロープモード・アベレーシングモード時の波形の取り込み回数を設定 / 問い合わせします。

構文 :ACQuire:COUNT {<NRf>|INfInity}

<NRf> = 1 ~ 65536

例 :ACQUIRE:COUNT 1

:ACQUIRE:COUNT?

-> :ACQUIRE:COUNT 1

:ACQuire:MODE

機能 波形の取り込みモード (アキュイジションモード) を設定 / 問い合わせします。

構文 :ACQuire:MODE {AVERage|ENVELOpe|NORMal}

例 :ACQUIRE:MODE AVERAGE

:ACQUIRE:MODE?

-> :ACQUIRE:MODE AVERAGE

:ACQuire:RESolution

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:RESolution {<Boolean>}

例 :ACQUIRE:RESOLUTION ON

:ACQUIRE:RESOLUTION?

-> :ACQUIRE:RESOLUTION 1

解説 ハイレゾリューションモードを ON にすると、最大レコード長が半分になります。

:ACQuire:RELENgth

機能 レコード長を設定 / 問い合わせします。

構文 :ACQuire:RELENgth {<NRf>}

<NRf> = 本体ユーザーズマニュアル機能編参照。

例

:ACQUIRE:RELENGTH 1250

:ACQUIRE:RELENGTH?

-> :ACQUIRE:RELENGTH 1250

:ACQuire:SAMPLing

機能 サンプリングモードを設定 / 問い合わせします。

構文 :ACQuire:SAMPLing {REAL|INTERporate|REPetitive}

例

:ACQUIRE:SAMPLING REAL

:ACQUIRE:SAMPLING?

-> :ACQUIRE:SAMPLING REAL

5.3 ANALysis グループ

電源解析機能 (:ANALysis:PANalyze や :ANALysis:PMEASure 系コマンド) はオプションです。電源解析機能のオプション付でないモデルでは、電源解析に関するコマンドは使用できません。

: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?

:ANALysis:AHISTogram<x>?

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:AHISTogram<x>?

<x> = 1 ~ 2

:ANALysis:AHISTogram<x>:DISPlay

機能 波形のヒストグラムの表示の ON/OFF を設定 / 問い合わせします。

構文 :ANALysis:AHISTogram<x>:

DISPlay {<Boolean>}

:ANALysis:AHISTogram<x>:DISPlay?

<x> = 1 ~ 2

例 :ANALYSIS:AHISTOGRAM1:DISPLAY ON

:ANALYSIS:AHISTOGRAM1:DISPLAY?

-> :ANALYSIS:AHISTOGRAM1:DISPLAY 1

:ANALysis:AHISTogram<x>:HORizontal

機能 波形の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 1.000E+00,0.000E+00

:ANALysis:AHISTogram<x>:MEASure?

機能 波形のヒストグラムの測定 (モードの ON/OFF を含む) 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

:ANALysis:AHISTogram<x>:MEASure:MODE

機能 波形のヒストグラムの測定モードを設定 / 問い合わせします。

構文 :ANALysis:AHISTogram<x>:MEASure:

MODE {OFF|PARAMeter}

:ANALysis:AHISTogram<x>:MEASure:

MODE?

<x> = 1 ~ 2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:

MODE OFF

:ANALYSIS:AHISTOGRAM1:MEASURE:MODE?

-> :ANALYSIS:AHISTOGRAM1:MEASURE:

MODE OFF

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

機能 モードが Param のときの波形のヒストグラム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:AHISTogram<x>:MEASure:

PARAMeter?

<x> = 1 ~ 2

:ANALysis:AHISTogram<x>:MEASure:PARAMeter:ALL

機能 波形のヒストグラムのすべての測定項目を一斉に ON/OFF します。

構文 :ANALysis:AHISTogram<x>:MEASure:

PARAMeter:ALL {<Boolean>}

<x> = 1 ~ 2

例 :ANALYSIS:AHISTOGRAM1:MEASURE:

PARAMETER:ALL ON

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

機能 波形のヒストグラムの指定した測定項目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:<パラメータ>?

<x> = 1 ~ 2

<パラメータ> = {C1|C2|DC|MAXimum|MEAN|MEDian|MINimum|PEAK|SD2integ|SD3integ|SDEViation|SDINteg}

5.3 ANALysis グループ

:ANALysis:AHIStogram<x>:MEASure:PARa meter:<パラメータ>:STATe

機能 波形のヒストグラムの指定した測定項目の ON/
OFF を設定 / 問い合わせします。

構文 :ANALysis:AHIStogram<x>:MEASure:
PARaMeter:<パラメータ>:STATe
{<Boolean>}
:ANALysis:AHIStogram<x>:MEASure:
PARaMeter:<パラメータ>:STATe?
<x> = 1 ~ 2
<パラメータ> = {C1|C2|DC|MAXimum|MEAN|M
EDian|MINimum|PEAK|SD2integ|SD3integ|SDEVia
tion|SDInteg}

例 以下は、最大値についての例です。
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:MAXIMUM:STATE ON
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:MAXIMUM:STATE?
-> :ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:MAXIMUM:STATE 1

:ANALysis:AHIStogram<x>:MEASure:PARa meter:<パラメータ>:VALue?

機能 波形のヒストグラムの指定した測定項目の自動
測定値を問い合わせます。

構文 :ANALysis:AHIStogram<x>:MEASure:
PARaMeter:<パラメータ>:VALue?
<x> = 1 ~ 2
<パラメータ> = {C1|C2|DC|MAXimum|MEAN|M
EDian|MINimum|PEAK|SD2integ|SD3integ|SDEVia
tion|SDInteg}

例 以下は、最大値についての例です。
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:MAXIMUM:VALUE?
-> :ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:MAXIMUM:VALUE 1.000E+00

:ANALysis:AHIStogram<x>:MEASure:PARa meter:HPOSition<y>

機能 波形のヒストグラムの波形 Cursor1 または
Cursor2 水平位置を設定 / 問い合わせします。

構文 :ANALysis:AHIStogram<x>:MEASure:
PARaMeter:HPOSition<y> {<NRf>}
:ANALysis:AHIStogram<x>:MEASure:
PARaMeter:HPOSition<y>?
<x> = 1 ~ 2
<y> = 1, 2
<NRf> = -5 ~ 5(div)

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

:ANALysis:AHIStogram<x>:MEASure:PARa meter:VPOSition<y>

機能 波形のヒストグラムの Cursor1 または Cursor2 の
垂直位置を設定 / 問い合わせします。

構文 :ANALysis:AHIStogram<x>:MEASure:
PARaMeter:VPOSition<y> {<NRf>}
:ANALysis:AHIStogram<x>:MEASure:
PARaMeter:VPOSition<y>?
<x> = 1 ~ 2
<y> = 1, 2
<NRf> = -4 ~ 4(div)

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

:ANALysis:AHIStogram<x>:MODE

機能 波形のヒストグラムの対象軸を設定 / 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>:RANGe

機能 波形の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>:RANGe {MAIN|
Z1|Z2}
:ANALysis:AHIStogram<x>:RANGe?
<x> = 1 ~ 2

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

:ANALysis:AHIStogram<x>:TRACe

機能 波形のヒストグラムの対象波形を設定 / 問い
合わせします。

構文 :ANALysis:AHIStogram<x>:
TRACe {<NRf>|MATH<y>}
:ANALysis:AHIStogram<x>:TRACe?
<x> = 1 ~ 2
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)

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

:ANALysis:AHISTogram<x>:VERTical

機能 波形の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 0,1
:ANALYSIS:AHISTOGRAM1:VERTICAL?
-> :ANALYSIS:AHISTOGRAM1:
VERTICAL 1.000E+00,0.000E+00

:ANALysis:PANalyze?

機能 電源解析に関するすべての設定値を問い合わせします。

構文 :ANALysis:PANalyze?

:ANALysis:PANalyze:HARMonics?

機能 高調波解析機能に関するすべての設定値を問い合わせします。

構文 :ANALysis:PANalyze:HARMonics?

:ANALysis:PANalyze:HARMonics:CCLass?

機能 高調波解析のクラス C に関するすべての設定値を問い合わせします。

構文 :ANALysis:PANalyze:HARMonics:CCLass?

:ANALysis:PANalyze:HARMonics:CCLass:GETLambda

機能 高調波解析のクラス C の現在の力率を設定します。

構文 :ANALysis:PANalyze:HARMonics:CCLass:
GETLambda

例 :ANALYSIS:PANALYZE:HARMONICS:CCLASS:
GETLAMBDA

:ANALysis:PANalyze:HARMonics:CCLass:LAMBda

機能 高調波解析のクラス C の力率を設定 / 問い合わせします。

構文 :ANALysis:PANalyze:HARMonics:CCLass:
LAMBda {<Nrf>}
:ANALysis:PANalyze:HARMonics:CCLass:
LAMBda?
<Nrf> = 0.01 ~ 1

例 :ANALYSIS:PANALYZE:HARMONICS:CCLASS:
LAMBDA 0.10
:ANALYSIS:PANALYZE:HARMONICS:CCLASS:
LAMBDA?
-> :ANALYSIS:PANALYZE:HARMONICS:
CCLASS:LAMBDA 100.0E-03

:ANALysis:PANalyze:HARMonics:CCLass:MAXCurrent

機能 高調波解析のクラス C の基本波の電流値を設定 / 問い合わせします。

構文 :ANALysis:PANalyze:HARMonics:CCLass:
MAXCurrent {<Nrf>|<電流>}
:ANALysis:PANalyze:HARMonics:CCLass:
MAXCurrent?

例 :ANALYSIS:PANALYZE:HARMONICS:CCLASS:
MAXCURRENT 50A
:ANALYSIS:PANALYZE:HARMONICS:CCLASS:
MAXCURRENT?
-> :ANALYSIS:PANALYZE:HARMONICS:
CCLASS:MAXCURRENT 50.000E+00

:ANALysis:PANalyze:HARMonics:CCLass:OPower

機能 高調波解析のクラス C の有効電力 25W を超える / 超えないを設定 / 問い合わせします。

構文 :ANALysis:PANalyze:HARMonics:CCLass:
OPower {FALSE|TRUE}
:ANALysis:PANalyze:HARMonics:CCLass:
OPower?

例 :ANALYSIS:PANALYZE:HARMONICS:CCLASS:
OPower FALSE
:ANALYSIS:PANALYZE:HARMONICS:CCLASS:
OPower?
-> :ANALYSIS:PANALYZE:HARMONICS:
CCLASS:OPower FALSE

:ANALysis:PANalyze:HARMonics:CLASs

機能 高調波解析の対象機器の適用クラスを設定 / 問い合わせします。

構文 :ANALysis:PANalyze:HARMonics:
CLASs {A|B|C|D}
:ANALysis:PANalyze:HARMonics:CLASs?
例 :ANALYSIS:PANALYZE:HARMONICS:CLASS A
:ANALYSIS:PANALYZE:HARMONICS:CLASS?
-> :ANALYSIS:PANALYZE:HARMONICS:
CLASS A

:ANALysis:PANalyze:HARMonics:DCLass?

機能 高調波解析のクラス D に関するすべての設定値を問い合わせします。

構文 :ANALysis:PANalyze:HARMonics:DCLass?

5.3 ANALysis グループ

:ANALysis: PANalyze: HARMonics: DCLass: POWER

機能 高調波解析のクラス D の電力値を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: HARMonics: DCLass: POWER {<NRf>}

:ANALysis: PANalyze: HARMonics: DCLass: POWER?

例 <NRf> = -1.0000E+31 ~ 1.0000E+31

:ANALYSIS: PANALYZE: HARMONICS: DCLASS: POWER 1V

:ANALYSIS: PANALYZE: HARMONICS: DCLASS: POWER?

-> :ANALYSIS: PANALYZE: HARMONICS: DCLASS: POWER 1.000E+00

:ANALysis: PANalyze: HARMonics: DMODe

機能 高調波解析の表示モードを設定 / 問い合わせします。

構文 :ANALysis: PANalyze: HARMonics: DMODe {LINear|LOG}

:ANALysis: PANalyze: HARMonics: DMODe?

例 :ANALYSIS: PANALYZE: HARMONICS: DMODE LINEAR

:ANALYSIS: PANALYZE: HARMONICS: DMODE?

-> :ANALYSIS: PANALYZE: HARMONICS: DMODE LINEAR

:ANALysis: PANalyze: HARMonics: DETail?

機能 高調波解析の解析結果リストに関するすべての設定値を問い合わせます。

構文 :ANALysis: PANalyze: HARMonics: DETail?

:ANALysis: PANalyze: HARMonics: DETail: DISPlay

機能 高調波解析の解析結果リストの表示モードを設定 / 問い合わせします。

構文 :ANALysis: PANalyze: HARMonics: DETail: DISPlay {FULL|LOWer|UPPer}

:ANALysis: PANalyze: HARMonics: DETail: DISPlay?

例 :ANALYSIS: PANALYZE: HARMONICS: DETAIL: DISPLAY FULL

:ANALYSIS: PANALYZE: HARMONICS: DETAIL: DISPLAY?

-> :ANALYSIS: PANALYZE: HARMONICS: DETAIL: DISPLAY FULL

:ANALysis: PANalyze: HARMonics: DETail: LIST: ITEM?

機能 高調波解析の解析結果リストに表示される項目を問い合わせます。

構文 :ANALysis: PANalyze: HARMonics: DETail: LIST: ITEM?

例 :ANALYSIS: PANALYZE: HARMONICS: DETAIL: LIST: ITEM?

-> :ANALYSIS: PANALYZE: HARMONICS: DETAIL: LIST: ITEM "Order., Measure(A), Limit(A), Measure(%), Limit(%), Info,"

:ANALysis: PANalyze: HARMonics: DETail: LIST: VALue?

機能 高調波解析の解析結果リストの指定した解析番号の全データを問い合わせます。

構文 :ANALysis: PANalyze: HARMonics: DETail: LIST: VALue? {<NRf>}

<NRf> = 1 ~ 40(高調波の次数)

例 :ANALYSIS: PANALYZE: HARMONICS: DETAIL: LIST: VALUE? 2

-> :ANALYSIS: PANALYZE: HARMONICS: DETAIL: LIST: VALUE " 2, 0.031, 0.020, 3.149, 2.000, NG,"

:ANALysis: PANalyze: HARMonics: GROUping

機能 高調波解析のグルーピングを設定 / 問い合わせします。

構文 :ANALysis: PANalyze: HARMonics: GROUping {OFF|TYPE1|TYPE2}

:ANALysis: PANalyze: HARMonics: GROUping?

例 :ANALYSIS: PANALYZE: HARMONICS: GROUPING OFF

:ANALYSIS: PANALYZE: HARMONICS: GROUPING?

-> :ANALYSIS: PANALYZE: HARMONICS: GROUPING OFF

:ANALysis: PANalyze: HARMonics: IECedit ion

機能 高調波解析の測定モードの測定規格 (IEC61000-4-7) を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: HARMonics: IECedit ion {ED2 _ 0|ED2 _ 1}

:ANALysis: PANalyze: HARMonics: IECedit ion?

例 :ANALYSIS: PANALYZE: HARMONICS: IECEDITION ED2 _ 0

:ANALYSIS: PANALYZE: HARMONICS: IECEDITION?

-> :ANALYSIS: PANALYZE: HARMONICS: IECEDITION ED2 _ 0

:ANALysis: PANalyze: HARMonics: RMS?

機能 高調波演算の実効値を問い合わせます。
 構文 :ANALysis: PANalyze: HARMonics: RMS?

:ANALysis: PANalyze: HARMonics: SPOint

機能 高調波解析の演算開始点を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: HARMonics: SPOint {<NRf>}
 :ANALysis: PANalyze: HARMonics: SPOint? <NRf> = -5 ~ 5(div)

例 :ANALYSIS: PANALYZE: HARMONICS: SPOINT 1
 :ANALYSIS: PANALYZE: HARMONICS: SPOINT?
 -> :ANALYSIS: PANALYZE: HARMONICS: SPOINT 1.000E+00

:ANALysis: PANalyze: HARMonics: SVOLtage

機能 高調波解析の電源電圧を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: HARMonics: SVOLtage {<NRf>|<電圧>}
 :ANALysis: PANalyze: HARMonics: SVOLtage?

例 :ANALYSIS: PANALYZE: HARMONICS: SVOLTAGE 230
 :ANALYSIS: PANALYZE: HARMONICS: SVOLTAGE?
 -> :ANALYSIS: PANALYZE: HARMONICS: SVOLTAGE 230.00000E+00

:ANALysis: PANalyze: HARMonics: THD?

機能 高調波演算の高調波ひずみ率を問い合わせます。

構文 :ANALysis: PANalyze: HARMonics: THD?
 例 :ANALYSIS: PANALYZE: HARMONICS: THD?

:ANALysis: PANalyze: I2T?

機能 ジュール積分に関するすべての設定値を問い合わせます。

構文 :ANALysis: PANalyze: I2T?

:ANALysis: PANalyze: I2T: MATH

機能 ジュール積分波形の表示の ON/OFF を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: I2T: MATH {I2T| OFF}
 :ANALysis: PANalyze: I2T: MATH?

例 :ANALYSIS: PANALYZE: I2T: MATH I2T
 :ANALYSIS: PANALYZE: I2T: MATH?
 -> :ANALYSIS: PANALYZE: I2T: MATH I2T

:ANALysis: PANalyze: I2T: MEASure?

機能 ジュール積分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: PANalyze: I2T: MEASure?

:ANALysis: PANalyze: I2T: MEASure: I2T?

機能 ジュール積分に関する設定を問い合わせます。

構文 :ANALysis: PANalyze: I2T: MEASure: I2T?

:ANALysis: PANalyze: I2T: MEASure: I2T: COUNT?

機能 ジュール積分の継続統計処理の回数を問い合わせます。

構文 :ANALysis: PANalyze: I2T: MEASure: I2T: COUNT?

例 :ANALYSIS: PANALYZE: I2T: MEASURE: I2T: COUNT?
 -> :ANALYSIS: PANALYZE: I2T: MEASURE: I2T: COUNT 100

:ANALysis: PANalyze: I2T: MEASure: I2T: { MAXimum | MEAN | MINimum | SDEVIation }?

機能 ジュール積分の各統計値を問い合わせます。

構文 :ANALysis: PANalyze: I2T: MEASure: I2T: {MAXimum|MEAN|MINimum|SDEVIation}?

例 :ANALYSIS: PANALYZE: I2T: MEASURE: I2T: MAXIMUM?
 -> :ANALYSIS: PANALYZE: I2T: MEASURE: I2T: MAXIMUM 10.0000E+03

解説 統計値を取得できない場合は、「NAN(Not A Number)」が返されます。

:ANALysis: PANalyze: I2T: MEASure: I2T: STATE

機能 ジュール積分の測定を行う (ON)/ 行わない (OFF) を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: I2T: MEASure: I2T: STATE {<Boolean>}

:ANALysis: PANalyze: I2T: MEASure: I2T: STATE?

例 :ANALYSIS: PANALYZE: I2T: MEASURE: I2T: STATE ON
 :ANALYSIS: PANALYZE: I2T: MEASURE: I2T: STATE?
 -> :ANALYSIS: PANALYZE: I2T: MEASURE: I2T: STATE 1

5.3 ANALysis グループ

:ANALysis: PANalyze: I2T: MEASure: I2T: VALue?

機能	ジュール積分の自動測定値を問い合わせます。
構文	:ANALysis: PANalyze: I2T: MEASure: I2T: VALue? [{<NRf>}]
例	<NRf> = 本体ユーザーズマニュアル機能編参照。 :ANALYSIS: PANALYZE: I2T: MEASURE: I2T: VALUE? -> :ANALYSIS: PANALYZE: I2T: MEASURE: I2T: VALUE 10.0000E+03
解説	- 測定不可能な場合は、「NaN(Not A Number)」が返されます。 <NRf> は、自動測定を実行してから <NRf> 回目の測定値を個々に問い合わせるときに使用します。 <NRf> に「1」を設定すると、自動測定値のメモリの中の最も古い測定値を問い合わせます。 - 設定した回数に測定値が存在しないときは、「NaN(Not A Number)」が返されます。 <NRf> を省略すると、最新の測定値を問い合わせます。 - 自動測定のサイクル統計処理をしている場合、<NRf> を設定すると、表示されている波形で画面の左から <NRf> 回目の 1 周期の測定値を問い合わせます。<NRf> を省略すると、表示されている波形で最後に測定された 1 周期の測定値を問い合わせます。

:ANALysis: PANalyze: I2T: RANGE

機能	測定対象ウィンドウを設定 / 問い合わせします。
構文	:ANALysis: PANalyze: I2T: RANGE {MAIN Z1 Z2}
例	:ANALYSIS: PANALYZE: I2T: RANGE? :ANALYSIS: PANALYZE: I2T: RANGE MAIN :ANALYSIS: PANALYZE: I2T: RANGE? -> :ANALYSIS: PANALYZE: I2T: RANGE MAIN

:ANALysis: PANalyze: I2T: SCALE?

機能	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	:ANALysis: PANalyze: I2T: SCALE?

:ANALysis: PANalyze: I2T: SCALE: CENTER

機能	マニュアルスケーリング時の中心値を設定 / 問い合わせします。
構文	:ANALysis: PANalyze: I2T: SCALE: CENTER {<NRf>}
例	:ANALYSIS: PANALYZE: I2T: SCALE: CENTER? <NRf> = -1.0000E+31 ~ 1.0000E+31 :ANALYSIS: PANALYZE: I2T: SCALE: CENTER 1 :ANALYSIS: PANALYZE: I2T: SCALE: CENTER? -> :ANALYSIS: PANALYZE: I2T: SCALE: CENTER 1.00000E+00

:ANALysis: PANalyze: I2T: SCALE: MODE

機能	スケーリングの方法を設定 / 問い合わせします。
構文	:ANALysis: PANalyze: I2T: SCALE: MODE {AUTO MANual}
例	:ANALYSIS: PANALYZE: I2T: SCALE: MODE? :ANALYSIS: PANALYZE: I2T: SCALE: MODE AUTO :ANALYSIS: PANALYZE: I2T: SCALE: MODE? -> :ANALYSIS: PANALYZE: I2T: SCALE: MODE AUTO

:ANALysis: PANalyze: I2T: SCALE: SENSitivity

機能	マニュアルスケーリング時の中心からのスパンを設定 / 問い合わせします。
構文	:ANALysis: PANalyze: I2T: SCALE: SENSitivity {<NRf>}
例	:ANALYSIS: PANALYZE: I2T: SCALE: SENSitivity? <NRf> = 1.0000E-31 ~ 1.0000E+31 :ANALYSIS: PANALYZE: I2T: SCALE: SENSITIVITY 10 :ANALYSIS: PANALYZE: I2T: SCALE: SENSITIVITY? -> :ANALYSIS: PANALYZE: I2T: SCALE: SENSITIVITY 10.0000E+00

:ANALysis: PANalyze: I2T: TRANGE (TimeRange)

機能	測定範囲を設定 / 問い合わせします。
構文	:ANALysis: PANalyze: I2T: TRANGE {<NRf>, <NRf>}
例	:ANALYSIS: PANALYZE: I2T: TRANGE? <NRf> = -5 ~ 5(div)、(10div/ 表示レコード長) ステップ :ANALYSIS: PANALYZE: I2T: TRANGE -4, 4 :ANALYSIS: PANALYZE: I2T: TRANGE? -> :ANALYSIS: PANALYZE: I2T: TRANGE -4.00E+00, 4.00E+00

:ANALysis: PANalyze: SETup?

機能	電源解析の入力に関するすべての設定値を問い合わせます。
構文	:ANALysis: PANalyze: SETup?

:ANALysis: PANalyze: SETup: ADESkew

機能	電源解析のオートスキュー補正を実行します。
構文	:ANALysis: PANalyze: SETup: ADESkew
例	:ANALYSIS: PANALYZE: SETUP: ADESKW

:ANALysis: PANalyze: SETUp: I?

機能 電源解析の電流入力チャネルに関するすべての設定値を問い合わせます。

構文 :ANALysis: PANalyze: SETUp: I?

:ANALysis: PANalyze: SETUp: I: DESKew

機能 電源解析の電流入力チャネルのスキュー補正を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SETUp: I:
DESKew {< 時間 >}
:ANALysis: PANalyze: SETUp: I: DESKew?
< 時間 > = -1000.00ns ~ 1000.00ns, 10ps ステップ

例 :ANALYSIS: PANALYZE: SETUP: I:
DESKEW 1NS
:ANALYSIS: PANALYZE: SETUP: I: DESKEW?
-> :ANALYSIS: PANALYZE: SETUP: I:
DESKEW 1.000E-09

:ANALysis: PANalyze: SETUp: I: INPut

機能 電源解析の電流入力チャネルを設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SETUp: I: INPut {2|4}
:ANALysis: PANalyze: SETUp: I: INPut?

例 :ANALYSIS: PANALYZE: SETUP: I: INPUT 2
:ANALYSIS: PANALYZE: SETUP: I: INPUT?
-> :ANALYSIS: PANALYZE: SETUP: I:
INPUT 2

:ANALysis: PANalyze: SETUp: I: PROBe

機能 電源解析の電流入力チャネルのプロープの電流 - 電圧換算比を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SETUp: I:
PROBe {C0 _ 001|C0 _ 002|C0 _ 005|C0 _ 01|C0 _ 02|C0 _ 05|C0 _ 1|C0 _ 2|C0 _ 5|C1|C2|C5|C10|C20|C50|C100|C200|C500|C1000|C2000}}

例 :ANALYSIS: PANALYZE: SETUP: I: PROBe?
:ANALYSIS: PANALYZE: SETUP: I:
PROBE C0 _ 001
:ANALYSIS: PANALYZE: SETUP: I: PROBE?
-> :ANALYSIS: PANALYZE: SETUP: I:
PROBE C0 _ 001

:ANALysis: PANalyze: SETUp: U?

機能 電源解析の電圧入力チャネルに関するすべての設定値を問い合わせます。

構文 :ANALysis: PANalyze: SETUp: U?

:ANALysis: PANalyze: SETUp: U: DESKew

機能 電源解析の電圧入力チャネルのスキュー補正を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SETUp: U:
DESKew {< 時間 >}
:ANALysis: PANalyze: SETUp: U: DESKew?
< 時間 > = -1000.00ns ~ 1000.00ns, 10ps ステップ
例 :ANALYSIS: PANALYZE: SETUP: U:
DESKEW 1NS
:ANALYSIS: PANALYZE: SETUP: U: DESKEW?
-> :ANALYSIS: PANALYZE: SETUP: U:
DESKEW 1.000E-09

:ANALysis: PANalyze: SETUp: U: INPut

機能 電源解析の電圧入力チャネルを設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SETUp: U: INPut {1|3}
:ANALysis: PANalyze: SETUp: U: INPut?
例 :ANALYSIS: PANALYZE: SETUP: U: INPUT 1
:ANALYSIS: PANALYZE: SETUP: U: INPUT?
-> :ANALYSIS: PANALYZE: SETUP: U:
INPUT 1

:ANALysis: PANalyze: SETUp: U: PROBe

機能 電源解析の電圧入力チャネルのプロープの減衰比を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SETUp: U:
PROBe {<NRF>}
:ANALysis: PANalyze: SETUp: U: PROBe?
<NRF> = 本体ユーザーズマニュアル機能編参照。
例 :ANALYSIS: PANALYZE: SETUP: U: PROBE 1
:ANALYSIS: PANALYZE: SETUP: U: PROBE?
-> :ANALYSIS: PANALYZE: SETUP: U:
PROBE 1.000

:ANALysis: PANalyze: SETUp: RTRace

機能 電源解析のスキュー補正の対象トレースを設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SETUp: RTRace {I|U}
:ANALysis: PANalyze: SETUp: RTRace?
例 :ANALYSIS: PANALYZE: SETUP: RTRACE I
:ANALYSIS: PANALYZE: SETUP: RTRACE?
-> :ANALYSIS: PANALYZE: SETUP: RTRACE I

:ANALysis: PANalyze: SOA?

機能 XY 表示 (安全動作領域) に関するすべての設定値を問い合わせます。

構文 :ANALysis: PANalyze: SOA?

5.3 ANALysis グループ

:ANALysis: PANalyze: SOA: CURSor?

機能 XY 表示 (安全動作領域) 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: PANalyze: SOA: CURSor?

:ANALysis: PANalyze: SOA: CURSor: X<x>?

機能 XY 表示 (安全動作領域) 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: PANalyze: SOA: CURSor: X<x>?
<x> = 1 ~ 2

:ANALysis: PANalyze: SOA: CURSor: X<x>: POSITION

機能 XY 表示 (安全動作領域) の水平カーソルの位置を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SOA: CURSor: X<x>:
POSITION {<NRf>}
:ANALysis: PANalyze: SOA: CURSor: X<x>:
POSITION?
<x> = 1 ~ 2
<NRf> = -4 ~ 4(div)

例 :ANALYSIS: PANALYZE: SOA: CURSOR: X1:
POSITION 1
:ANALYSIS: PANALYZE: SOA: CURSOR: X1:
POSITION?
-> :ANALYSIS: PANALYZE: SOA: CURSOR: X:
POSITION 1.000E+00

:ANALysis: PANalyze: SOA: CURSor: X<x>: VALue?

機能 XY 表示 (安全動作領域) の水平カーソルの電圧値を問い合わせます。

構文 :ANALysis: PANalyze: SOA: CURSor: X<x>:
VALue?
<x> = 1 ~ 2

例 :ANALYSIS: PANALYZE: SOA: CURSOR: X1:
VALUE?
-> :ANALYSIS: PANALYZE: SOA: CURSOR: X1:
VALUE 1.000E+00

:ANALysis: PANalyze: SOA: CURSor: Y<y>?

機能 XY 表示 (安全動作領域) 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: PANalyze: SOA: CURSor: Y<y>?
<y> = 1 ~ 2

:ANALysis: PANalyze: SOA: CURSor: Y<y>: POSITION

機能 XY 表示 (安全動作領域) の垂直カーソルの位置を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SOA: CURSor: Y<y>:
POSITION {<NRf>}
:ANALysis: PANalyze: SOA: CURSor: Y<y>:
POSITION?
<y> = 1 ~ 2
<NRf> = -4 ~ 4(div)

例 :ANALYSIS: PANALYZE: SOA: CURSOR: Y1:
POSITION 1
:ANALYSIS: PANALYZE: SOA: CURSOR: Y1:
POSITION?
-> :ANALYSIS: PANALYZE: SOA: CURSOR: Y:
POSITION 1.000E+00

:ANALysis: PANalyze: SOA: CURSor: Y<y>: VALue?

機能 XY 表示 (安全動作領域) の垂直カーソルの電圧値を問い合わせます。

構文 :ANALysis: PANalyze: SOA: CURSor: Y<y>:
VALue?
<y> = 1 ~ 2

例 :ANALYSIS: PANALYZE: SOA: CURSOR: Y1:
VALUE?
-> :ANALYSIS: PANALYZE: SOA: CURSOR: Y1:
VALUE 1.000E+00

:ANALysis: PANalyze: SOA: MODE

機能 XY 表示 (安全動作領域) の自動測定モードを設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SOA: MODE {CURSor|
OFF}
:ANALysis: PANalyze: SOA: MODE?

例 :ANALYSIS: PANALYZE: SOA: MODE CURSOR
:ANALYSIS: PANALYZE: SOA: MODE?
-> :ANALYSIS: PANALYZE: SOA:
MODE CURSOR

:ANALysis: PANalyze: SOA: TRANge (Time Range)

機能 XY 表示 (安全動作領域) する T-Y 波形の範囲を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SOA:
TRANge {<NRf>, <NRf>}
:ANALysis: PANalyze: SOA: TRANge?
<NRf> = -5 ~ 5(div)、(10div/ 表示レコード長)
ステップ

例 :ANALYSIS: PANALYZE: SOA: TRANGE -4, 4
:ANALYSIS: PANALYZE: SOA: TRANGE?
-> :ANALYSIS: PANALYZE: SOA:
TRANGE -4.00, 4.00

:ANALysis: PANalyze: SOA: VTDisplay

機能 XY 表示 (安全動作領域) の VT 波形の表示の ON/OFF を設定 / 問い合わせします。

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

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

:ANALysis: PANalyze: SWLoss?

機能 スイッチング損失に関するすべての設定値を問い合わせます。

構文 :ANALysis: PANalyze: SWLoss?
解説 トータル損失を求めるための基準レベル (ディスタル / メシアル / プロキシマルなど) は、次のコマンドで設定します。
:MEASure: {CHANnel<x>|MATH<x>}: DPROximal?
:MEASure: {CHANnel<x>|MATH<x>}: DPROximal: M
ODE
:MEASure: {CHANnel<x>|MATH<x>}: DPROximal: PE
RCent
:MEASure: {CHANnel<x>|MATH<x>}: DPROximal: U
NIT
:MEASure: {CHANnel<x>|MATH<x>}: METHod

:ANALysis: PANalyze: SWLoss: CCALc

機能 トータル損失の計算式を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SWLoss: CCALc {RDS|
VCE|WAVEform}

例 :ANALysis: PANalyze: SWLoss: CCALc?
:ANALYSIS: PANALYZE: SWLOSS: CCALc RDS
:ANALYSIS: PANALYZE: SWLOSS: CCALc?
-> :ANALYSIS: PANALYZE: SWLOSS: CCALc RDS

:ANALysis: PANalyze: SWLoss: DPROximal?

機能 ディスタル・メシアル・プロキシマルに関するすべての設定値を問い合わせます。

構文 :ANALysis: PANalyze: SWLoss: DPROximal?
例 :ANALYSIS: PANALYZE: SWLOSS: DPROXIMAL?

:ANALysis: PANalyze: SWLoss: DPROximal:**MODE**

機能 ディスタル・メシアル・プロキシマル点のモードを設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SWLoss: DPROximal: MODE {PERCent|UNIT}
:ANALysis: PANalyze: SWLoss: DPROximal: MODE?

例 :ANALYSIS: PANALYZE: SWLOSS: DPROXIMAL: MODE PERCENT
:ANALYSIS: PANALYZE: SWLOSS: DPROXIMAL: MODE?
-> :ANALYSIS: PANALYZE: SWLOSS: DPROXIMAL: MODE PERCENT

:ANALysis: PANalyze: SWLoss: DPROximal: PERCent

機能 ディスタル・メシアル・プロキシマル点を % で設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SWLoss: DPROximal: PERCent {<NRf>, <NRf>, <NRf>}
:ANALysis: PANalyze: SWLoss: DPROximal: PERCent?

<NRf>, <NRf>, <NRf> = 0(%) ~ 100(%)、1% ステップ

例 :ANALYSIS: PANALYZE: SWLOSS: DPROXIMAL: PERCENT 10, 50, 90
:ANALYSIS: PANALYZE: SWLOSS: DPROXIMAL: PERCENT?
-> :ANALYSIS: PANALYZE: SWLOSS: DPROXIMAL: PERCENT 10, 50, 90

:ANALysis: PANalyze: SWLoss: DPROximal: UNIT

機能 ディスタル・メシアル・プロキシマル点を電圧値で設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SWLoss: DPROximal: UNIT {<NRf>, <NRf>, <NRf>}
:ANALysis: PANalyze: SWLoss: DPROximal: UNIT?

<NRf>, <NRf>, <NRf> = 本体ユーザズマニュアル機能編参照。

例 :ANALYSIS: PANALYZE: SWLOSS: DPROXIMAL: UNIT -1, 0, 1
:ANALYSIS: PANALYZE: SWLOSS: DPROXIMAL: UNIT?
-> :ANALYSIS: PANALYZE: SWLOSS: DPROXIMAL: UNIT -1.0000000E+00, 0.0000000E+00, 1.0000000E+00

5.3 ANALysis グループ

:ANALysis: PANalyze: SWLoss: ILEVel
機能 トータル損失の損失ゼロ区間と判定する電流レベルを設定 / 問い合わせします。
構文 :ANALysis: PANalyze: SWLoss: ILEVel {<NRf>|<電流>}
:ANALysis: PANalyze: SWLoss: ILEVel? <NRf>、<電流> = 本体ユーザーズマニュアル機能編参照。
例 :ANALYSIS: PANALYZE: SWLOSS: ILEVEL 1
:ANALYSIS: PANALYZE: SWLOSS: ILEVEL?
-> :ANALYSIS: PANALYZE: SWLOSS: ILEVEL 1.0000000E+00

:ANALysis: PANalyze: SWLoss: INDicator
機能 トータル損失の測定箇所の表示を設定 / 問い合わせします。
構文 :ANALysis: PANalyze: SWLoss: INDicator {<Boolean>}
:ANALysis: PANalyze: SWLoss: INDicator? <Boolean>
例 :ANALYSIS: PANALYZE: SWLOSS: INDICATOR OFF
:ANALYSIS: PANALYZE: SWLOSS: INDICATOR?
-> :ANALYSIS: PANALYZE: SWLOSS: INDICATOR OFF

:ANALysis: PANalyze: SWLoss: MATH
機能 電力波形の表示の ON/OFF を設定 / 問い合わせします。
構文 :ANALysis: PANalyze: SWLoss: MATH {OFF|Power}
:ANALysis: PANalyze: SWLoss: MATH? <Boolean>
例 :ANALYSIS: PANALYZE: SWLOSS: MATH OFF
:ANALYSIS: PANALYZE: SWLOSS: MATH?
-> :ANALYSIS: PANALYZE: SWLOSS: MATH OFF

:ANALysis: PANalyze: SWLoss: MEASure?
機能 電源解析項目 (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: PANalyze: SWLoss: MEASure?

:ANALysis: PANalyze: SWLoss: MEASure: <パラメータ>?
機能 電源解析項目 (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: PANalyze: SWLoss: MEASure: <パラメータ>?
<パラメータ> = {CCOunt|PON|PTOff|PTON|PTOTa||WHON|WHTOff|WHTON|WHTOTa|}

:ANALysis: PANalyze: SWLoss: MEASure: <パラメータ>: COUNT?
機能 電源解析項目 (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 :ANALysis: PANalyze: SWLoss: MEASure: <パラメータ>: COUNT?
<パラメータ> = {CCOunt|PON|PTOff|PTON|PTOTa||WHON|WHTOff|WHTON|WHTOTa|}
例 :ANALYSIS: PANALYZE: SWLOSS: MEASURE: PON: COUNT?
-> :ANALYSIS: PANALYZE: SWLOSS: MEASURE: PON: COUNT 100

:ANALysis: PANalyze: SWLoss: MEASure: <パラメータ>: {MAXimum|MEAN|MINimum|SDEVIation}?
機能 電源解析項目 (パラメータ) の各統計値を問い合わせます。
構文 :ANALysis: PANalyze: SWLoss: MEASure: <パラメータ>: {MAXimum|MEAN|MINimum|SDEVIation}?
<パラメータ> = {CCOunt|PON|PTOff|PTON|PTOTa||WHON|WHTOff|WHTON|WHTOTa|}
例 :ANALYSIS: PANALYZE: SWLOSS: MEASURE: PON: MAXIMUM?
-> :ANALYSIS: PANALYZE: SWLOSS: MEASURE: PON: MAXIMUM 1.000E+00
解説 統計値を取得できない場合は、「NAN(Not A Number)」が返されます。

:ANALysis: PANalyze: SWLoss: MEASure: <パラメータ>: STATE
機能 電源解析項目 (パラメータ) の ON/OFF を設定 / 問い合わせします。
構文 :ANALysis: PANalyze: SWLoss: MEASure: <パラメータ>: STATE {<Boolean>}
:ANALysis: PANalyze: SWLoss: MEASure: <パラメータ>: STATE? <Boolean>
<パラメータ> = {CCOunt|PON|PTOff|PTON|PTOTa||WHON|WHTOff|WHTON|WHTOTa|}
例 :ANALYSIS: PANALYZE: SWLOSS: MEASURE: PON: STATE ON
:ANALYSIS: PANALYZE: SWLOSS: MEASURE: PON: STATE?
-> :ANALYSIS: PANALYZE: SWLOSS: MEASURE: PON: STATE 1

:ANALysis: PANalyze: SWLoss: MEASure: <パラメータ>: VALue?

機能 電源解析項目 (パラメータ) の自動測定値を問い合わせます。

構文 :ANALysis: PANalyze: SWLoss: MEASure: <パラメータ>: VALue? [{<NRf>}]
 <NRf> = 本体ユーザーズマニュアル機能編参照。
 <パラメータ> = {CCOunt|PON|PTOff|PTON|PTOTal|WHON|WHTOff|WHTON|WHTOTal}

例 :ANALYSIS: PANALYZE: SWLOSS: MEASURE: PON: VALUE?
 -> :ANALYSIS: PANALYZE: SWLOSS: MEASURE: PON: VALUE 10.0000E+03

解説

- 測定不可能な場合は、「NaN(Not A Number)」が返されます。
- <NRf> は、自動測定を実行してから <NRf> 回目の測定値を個々に問い合わせるときに使用します。
- <NRf> に「1」を設定すると、自動測定値のメモリの中の最も古い測定値を問い合わせます。
- 設定した回数に測定値が存在しないときは、「NaN(Not A Number)」が返されます。
- <NRf> を省略すると、最新の測定値を問い合わせます。
- 自動測定のサイクル統計処理をしている場合、<NRf> を設定すると、表示されている波形で画面の左から <NRf> 回目の 1 周期の測定値を問い合わせます。<NRf> を省略すると、表示されている波形で最後に測定された 1 周期の測定値を問い合わせます。

:ANALysis: PANalyze: SWLoss: MEASure: CONTinuous?

機能 トータル損失の自動測定の通常の統計処理に関するすべての設定値を問い合わせます。

:ANALysis: PANalyze: SWLoss: MEASure: CONTinuous: REStart

機能 トータル損失の自動測定の通常の統計処理を再スタートします。

構文 :ANALysis: PANalyze: SWLoss: MEASure: CONTinuous: REStart

例 :ANALYSIS: PANALYZE: SWLOSS: MEASURE: CONTINUOUS: RESTART

:ANALysis: PANalyze: SWLoss: MEASure: CONTinuous: TLCHange

機能 トータル損失の自動測定の通常の統計処理再スタートするか否かを設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SWLoss: MEASure: CONTinuous: TLCHange {REStart|IGNore}
 :ANALysis: PANalyze: SWLoss: MEASure: CONTinuous: TLCHange?

例 :ANALYSIS: PANALYZE: SWLOSS: MEASURE: CONTINUOUS: TLCHANGE RESTART
 :ANALYSIS: PANALYZE: SWLOSS: MEASURE: CONTINUOUS: TLCHANGE?
 -> :ANALYSIS: PANALYZE: SWLOSS: MEASURE: CONTINUOUS: TLCHANGE RESTART

:ANALysis: PANalyze: SWLoss: MEASure: CYCLE: ABORT

機能 トータル損失の自動測定のサイクル統計処理の実行を中止します。

構文 :ANALysis: PANalyze: SWLoss: MEASure: CYCLE: ABORT

例 :ANALYSIS: PANALYZE: SWLOSS: MEASURE: CYCLE: ABORT

:ANALysis: PANalyze: SWLoss: MEASure: CYCLE: EXECute

機能 トータル損失の自動測定のサイクル統計処理の実行します。

構文 :ANALysis: PANalyze: SWLoss: MEASure: CYCLE: EXECute

例 :ANALYSIS: PANALYZE: SWLOSS: MEASURE: CYCLE: EXECUTE

:ANALysis: PANalyze: SWLoss: MEASure: HISTory: ABORT

機能 トータル損失の自動測定のヒストリ波形の統計処理の実行を中止します。

構文 :ANALysis: PANalyze: SWLoss: MEASure: HISTory: ABORT

例 :ANALYSIS: PANALYZE: SWLOSS: MEASURE: HISTORY: ABORT

:ANALysis: PANalyze: SWLoss: MEASure: HISTory: EXECute

機能 トータル損失の自動測定のヒストリ波形の統計処理の実行します。

構文 :ANALysis: PANalyze: SWLoss: MEASure: HISTory: EXECute

例 :ANALYSIS: PANALYZE: SWLOSS: MEASURE: HISTORY: EXECUTE

5.3 ANALysis グループ

:ANALysis: PANalyze: SWLoss: MEASure: MO DE

機能 トータル損失の自動測定の ON/OFF/ 統計処理を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SWLoss: MEASure: MODE {OFF|CONTInuous|CYCLe|HISTory}

例 :ANALYSIS: PANALYZE: SWLOSS: MEASure: MODE OFF

:ANALYSIS: PANALYZE: SWLOSS: MEASure: MODE?

-> :ANALYSIS: PANALYZE: SWLOSS: MEASure: MODE OFF

:ANALysis: PANalyze: SWLoss: MEASure: ZL INKage

機能 トータル損失の自動測定の測定結果の結果番号とズーム位置との連動 (ON/OFF) を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SWLoss: MEASure: ZLINKage {<Boolean>}

例 :ANALYSIS: PANALYZE: SWLOSS: MEASure: ZLINKAGE ON

:ANALYSIS: PANALYZE: SWLOSS: MEASure: ZLINKAGE?

-> :ANALYSIS: PANALYZE: SWLOSS: MEASure: ZLINKAGE ON

:ANALysis: PANalyze: SWLoss: METHod

機能 High・Low 点の算出方法を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SWLoss: METHod {AUTO|MAXimum|HISTogram}

例 :ANALYSIS: PANALYZE: SWLOSS: METHod AUTO

:ANALYSIS: PANALYZE: SWLOSS: METHod?

-> :ANALYSIS: PANALYZE: SWLOSS: METHod AUTO

:ANALysis: PANalyze: SWLoss: RANGE

機能 測定対象ウィンドウを設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SWLoss: RANGE {MAIN|Z1|Z2}

例 :ANALYSIS: PANALYZE: SWLOSS: RANGE MAIN

:ANALYSIS: PANALYZE: SWLOSS: RANGE?

-> :ANALYSIS: PANALYZE: SWLOSS: RANGE MAIN

:ANALysis: PANalyze: SWLoss: RDS

機能 トータル損失のオン抵抗値を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SWLoss: RDS{<NRf>}

例 :ANALYSIS: PANALYZE: SWLOSS: RDS 1

:ANALYSIS: PANALYZE: SWLOSS: RDS?

-> :ANALYSIS: PANALYZE: SWLOSS: RDS 1.000E+00

:ANALysis: PANalyze: SWLoss: SCALE?

機能 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 :ANALysis: PANalyze: SWLoss: SCALE?

:ANALysis: PANalyze: SWLoss: SCALE: CENT er

機能 マニュアルスケーリング時の中心値を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SWLoss: SCALE: CENTER {<NRf>}

例 :ANALYSIS: PANALYZE: SWLOSS: SCALE: CENTER 1

:ANALYSIS: PANALYZE: SWLOSS: SCALE: CENTER?

<NRf> = -1.0000E+31 ~ 1.0000E+31

-> :ANALYSIS: PANALYZE: SWLOSS: SCALE: CENTER 1.00000E+00

:ANALysis: PANalyze: SWLoss: SCALE: MODE

機能 スケーリングの方法を設定 / 問い合わせします。

構文 :ANALysis: PANalyze: SWLoss: SCALE: MODE {AUTO|MANual}

例 :ANALYSIS: PANALYZE: SWLOSS: SCALE: MODE AUTO

:ANALYSIS: PANALYZE: SWLOSS: SCALE: MODE?

-> :ANALYSIS: PANALYZE: SWLOSS: SCALE: MODE AUTO

:ANALYSIS: PANalyze: SWLoss: SCALE: SENSitivity

機能 マニュアルスケリング時の中心からのスパンを設定 / 問い合わせします。

構文 :ANALYSIS: PANalyze: SWLoss: SCALE: SENSitivity {<NRf>}

:ANALYSIS: PANalyze: SWLoss: SCALE: SENSitivity?

例 <NRf> = 1.0000E-31 ~ 1.0000E+31

:ANALYSIS: PANALYZE: SWLOSS: SCALE: SENSITIVITY 10

:ANALYSIS: PANALYZE: SWLOSS: SCALE: SENSITIVITY?

-> :ANALYSIS: PANALYZE: SWLOSS: SCALE: SENSITIVITY 10.0000E+00

:ANALYSIS: PANalyze: SWLoss: TRAnge (Time Range)

機能 測定範囲を設定 / 問い合わせします。

構文 :ANALYSIS: PANalyze: SWLoss: TRAnge {<NRf>, <NRf>}

:ANALYSIS: PANalyze: SWLoss: TRAnge?

<NRf> = -5 ~ 5(div), (10div/ 表示レコード長) ステップ

例 :ANALYSIS: PANALYZE: SWLOSS: TRANGE -4, 4

:ANALYSIS: PANALYZE: SWLOSS: TRANGE?

-> :ANALYSIS: PANALYZE: SWLOSS: TRANGE -4.00E+00, 4.00E+00

:ANALYSIS: PANalyze: SWLoss: ULEVel

機能 トータル損失の損失計算区間と判定する電圧レベルを設定 / 問い合わせします。

構文 :ANALYSIS: PANalyze: SWLoss: ULEVel {<NRf>|<電圧>}

:ANALYSIS: PANalyze: SWLoss: ULEVel?

<NRf>, <電圧> = 本体ユーザーズマニュアル機能編参照。

例 :ANALYSIS: PANALYZE: SWLOSS: ULEVEL 1V

:ANALYSIS: PANALYZE: SWLOSS: ULEVEL?

-> :ANALYSIS: PANALYZE: SWLOSS: ULEVEL 1.0000000E+00

:ANALYSIS: PANalyze: SWLoss: UNIT

機能 電力量の単位を設定 / 問い合わせします。

構文 :ANALYSIS: PANalyze: SWLoss: UNIT {J|WH}

:ANALYSIS: PANalyze: SWLoss: UNIT?

例 :ANALYSIS: PANALYZE: SWLOSS: UNIT WH

:ANALYSIS: PANALYZE: SWLOSS: UNIT?

-> :ANALYSIS: PANALYZE: SWLOSS: UNIT WH

:ANALYSIS: PANalyze: SWLoss: VCE

機能 トータル損失のコレクタ - エミッタ間飽和電圧値を設定 / 問い合わせします。

構文 :ANALYSIS: PANalyze: SWLoss: VCE {<NRf>|<電圧>}

:ANALYSIS: PANalyze: SWLoss: VCE?

<NRf>, <電圧> = 0V ~ 50V, 100mV ステップ

例 :ANALYSIS: PANALYZE: SWLOSS: VCE 1V

:ANALYSIS: PANALYZE: SWLOSS: VCE?

-> :ANALYSIS: PANALYZE: SWLOSS:

VCE 1.0000000E+00

:ANALYSIS: PANalyze: TYPE

機能 電源解析の種類を設定 / 問い合わせします。

構文 :ANALYSIS: PANalyze: TYPE {HARMONICS|I2T|OFF|SOA|SWLoss}

:ANALYSIS: PANalyze: TYPE?

例 :ANALYSIS: PANALYZE: TYPE HARMONICS

:ANALYSIS: PANALYZE: TYPE?

-> :ANALYSIS: PANALYZE: TYPE HARMONICS

:ANALYSIS: PMEAsure<x>?

機能 電力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: PMEAsure<x>?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS: PMAASURE1?

-> :ANALYSIS: PMAASURE1

:ANALYSIS: PMEAsure<x>: CYCMode

機能 電力測定のサイクルモードを設定 / 問い合わせします。

構文 :ANALYSIS: PMEAsure<x>: CYCMode {OFF|NCYCLe}

:ANALYSIS: PMEAsure<x>: CYCMode?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS: PMAASURE1: CYCMODE OFF

:ANALYSIS: PMAASURE1: CYCMODE?

-> :ANALYSIS: PMAASURE1: CYCMODE OFF

:ANALYSIS: PMEAsure<x>: CYCTrace

機能 電力測定のサイクルモードの周期対象波形を設定 / 問い合わせします。

構文 :ANALYSIS: PMEAsure<x>:

CYCTrace {<NRf>|MATH<x>}

:ANALYSIS: PMEAsure<x>: CYCTrace?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS: PMAASURE1: CYCTRACE 1

:ANALYSIS: PMAASURE1: CYCTRACE?

-> :ANALYSIS: PMAASURE1: CYCTRACE 1

5.3 ANALysis グループ

:ANALysis:PMEASURE<x>:IDPROximal?

機能 ディスタル・メシアル・プロキシマルに関するすべての設定値を問い合わせます。

構文 :ANALysis:PMEASURE<x>:IDPROximal?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:IDPROXIMAL?

:ANALysis:PMEASURE<x>:IDPROximal:MODE

機能 ディスタル・メシアル・プロキシマル点のモードを設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:IDPROximal:
MODE {PERCent|UNIT}
:ANALysis:PMEASURE<x>:IDPROximal:
MODE?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)
例 :ANALYSIS:PMEASURE1:IDPROXIMAL:
MODE PERCENT
:ANALYSIS:PMEASURE1:IDPROXIMAL:MODE?
-> :ANALYSIS:PMEASURE1:IDPROXIMAL:
MODE PERCENT

:ANALysis:PMEASURE<x>:IDPROximal:PERCent

機能 ディスタル・メシアル・プロキシマル点を%で設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:IDPROximal:
PERCent {<NRf>,<NRf>,<NRf>}
:ANALysis:PMEASURE<x>:IDPROximal:
PERCent?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)
<NRf>、<NRf>、<NRf> = 0(%) ~ 100(%)、1%
ステップ
例 :ANALYSIS:PMEASURE1:IDPROXIMAL:
PERCENT 10,50,90
:ANALYSIS:PMEASURE1:IDPROXIMAL:
PERCENT?
-> :ANALYSIS:PMEASURE1:IDPROXIMAL:
PERCENT 10,50,90

:ANALysis:PMEASURE<x>:IDPROximal:UNIT

機能 ディスタル・メシアル・プロキシマル点を電圧値で設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:IDPROximal:
UNIT {<NRf>,<NRf>,<NRf>}
:ANALysis:PMEASURE<x>:IDPROximal:
UNIT?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)
<NRf>、<NRf>、<NRf> = 本体ユーザーズマニュアル機能編参照

例 :ANALYSIS:PMEASURE1:IDPROXIMAL:
UNIT -1,0,1
:ANALYSIS:PMEASURE1:IDPROXIMAL:UNIT?
-> :ANALYSIS:PMEASURE1:IDPROXIMAL:
UNIT -1.0000000E+00,0.0000000E+00,
1.0000000E+00

:ANALysis:PMEASURE<x>:IMETHod

機能 High・Low 点を設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:IMETHod {AUTO|
MAXimum|HISTogram}

:ANALysis:PMEASURE<x>:IMETHod?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:IMETHOD AUTO
:ANALYSIS:PMEASURE1:IMETHOD?
-> :ANALYSIS:PMEASURE1:IMETHOD AUTO

:ANALysis:PMEASURE<x>:INDicator

機能 測定箇所の表示を設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:
INDicator </パラメータ>
:ANALysis:PMEASURE<x>:INDicator?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)
</パラメータ> = {AH|AHABs|AHN|AHP|IAC|IA
VGfreq|IDC|IMN|INPeak|IPPeak|IPToPeak|IRMN|
IRMS|LAMBda|P|Q|S|UAC|UAVGfreq|UDC|UMN|
UNPeak|UPPeak|UPToPeak|URMN|URMS|WH|W
HABs|WHN|WHP|Z}

例 :ANALYSIS:PMEASURE1:INDICATOR AH
:ANALYSIS:PMEASURE1:INDICATOR?
-> :ANALYSIS:PMEASURE1:INDICATOR AH

:ANALysis:PMEASURE<x>:MEASURE?

機能 電力測定項目 (/ 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:PMEASURE<x>:MEASURE?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:MEASURE?

:ANALysis:PMEASURE<x>:MEASURE:<パラメータ>?

機能 電力測定項目 (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:PMEASURE<x>:MEASURE:
<パラメータ>?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)
<パラメータ> = {AH|AHABs|AHN|AHP|IAC|IAVGfreq|IDC|IMN|INPeak|IPPeak|IPTOpeak|IRMN|IRMS|LAMBda|P|Q|S|UAC|UAVGfreq|UDC|UMN|UNPeak|UPPeak|UPTOpeak|URMN|URMS|WH|WHABs|WHN|WHP|Z}

例 :ANALYSIS:PMEASURE1:MEASURE:AH?

:ANALysis:PMEASURE<x>:MEASURE:<パラメータ>:COUNT?

機能 電力測定項目 (パラメータ) の通常の統計処理の回数を問い合わせます。

構文 :ANALysis:PMEASURE<x>:MEASURE:
<パラメータ>:COUNT?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)
<パラメータ> = {AH|AHABs|AHN|AHP|IAC|IAVGfreq|IDC|IMN|INPeak|IPPeak|IPTOpeak|IRMN|IRMS|LAMBda|P|Q|S|UAC|UAVGfreq|UDC|UMN|UNPeak|UPPeak|UPTOpeak|URMN|URMS|WH|WHABs|WHN|WHP|Z}

例 :ANALYSIS:PMEASURE1:MEASURE:AH:
COUNT?
-> :ANALYSIS:PMEASURE1:MEASURE:AH:
COUNT 1

:ANALysis:PMEASURE<x>:MEASURE:<パラメータ>:{MAXimum|MEAN|MINimum|SDEVIation}?

機能 電力測定項目 (パラメータ) の各統計値を問い合わせます。

構文 :ANALysis:PMEASURE<x>:MEASURE:
<パラメータ>:{MAXimum|MEAN|MINimum|SDEViation}?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)
<パラメータ> = {AH|AHABs|AHN|AHP|IAC|IAVGfreq|IDC|IMN|INPeak|IPPeak|IPTOpeak|IRMN|IRMS|LAMBda|P|Q|S|UAC|UAVGfreq|UDC|UMN|UNPeak|UPPeak|UPTOpeak|URMN|URMS|WH|WHABs|WHN|WHP|Z}

例 :ANALYSIS:PMEASURE1:MEASURE:AH:
MAXIMUM?
-> :ANALYSIS:PMEASURE1:MEASURE:AH:
MAXIMUM 10.0000E-6

:ANALysis:PMEASURE<x>:MEASURE:<パラメータ>:STATE

機能 電力測定項目 (パラメータ) の ON/OFF を設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:MEASURE:
<パラメータ>:STATE {<Boolean>}
:ANALysis:PMEASURE<x>:MEASURE:
<パラメータ>:STATE?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)
<パラメータ> = {AH|AHABs|AHN|AHP|IAC|IAVGfreq|IDC|IMN|INPeak|IPPeak|IPTOpeak|IRMN|IRMS|LAMBda|P|Q|S|UAC|UAVGfreq|UDC|UMN|UNPeak|UPPeak|UPTOpeak|URMN|URMS|WH|WHABs|WHN|WHP|Z}

例 :ANALYSIS:PMEASURE1:MEASURE:AH:
STATE ON
:ANALYSIS:PMEASURE1:MEASURE:AH:
STATE?
-> :ANALYSIS:PMEASURE1:MEASURE:AH:
STATE 1

:ANALysis:PMEASURE<x>:MEASURE:<パラメータ>:VALUE?

機能 電力測定項目 (パラメータ) の自動測定値を問い合わせます。

構文 :ANALysis:PMEASURE<x>:MEASURE:
<パラメータ>:VALUE?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)
<パラメータ> = {AH|AHABs|AHN|AHP|IAC|IAVGfreq|IDC|IMN|INPeak|IPPeak|IPTOpeak|IRMN|IRMS|LAMBda|P|Q|S|UAC|UAVGfreq|UDC|UMN|UNPeak|UPPeak|UPTOpeak|URMN|URMS|WH|WHABs|WHN|WHP|Z}

例 :ANALYSIS:PMEASURE1:MEASURE:AH:
VALUE?
-> :ANALYSIS:PMEASURE1:MEASURE:AH:
VALUE -1.5000E-03

:ANALysis:PMEASURE<x>:MEASURE:ALL

機能 電力測定項目 (パラメータ) を一括 ON/OFF します。

構文 :ANALysis:PMEASURE<x>:MEASURE:
ALL {<Boolean>}
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:MEASURE:ALL ON

:ANALysis:PMEASURE<x>:MEASURE:CONTInuous?

機能 電力測定の自動測定の通常の統計処理に関するすべての設定値を問い合わせます。

構文 :ANALysis:PMEASURE<x>:MEASURE:CONTIn
uous?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

5.3 ANALysis グループ

:ANALysis:PMEASURE<x>:MEASURE:CONTInuous:REStArt

機能 電力測定の自動測定の通常の統計処理を再スタートします。

構文 :ANALysis:PMEASURE<x>:MEASURE:
CONTInuous:REStArt
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:MEASURE:
CONTINUOUS:RESTART

:ANALysis:PMEASURE<x>:MEASURE:CONTInuous:TLCHange

機能 電力測定の自動測定の通常の統計処理再スタートするか否かを設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:MEASURE:
CONTInuous:TLCHange {REStArt|IGNore}
:ANALysis:PMEASURE<x>:MEASURE:
CONTInuous:TLCHange?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:MEASURE:
CONTINUOUS:TLCHANGE RESTART
:ANALYSIS:PMEASURE1:MEASURE:
CONTINUOUS:TLCHANGE
-> :ANALYSIS:PMEASURE1:MEASURE:
CONTINUOUS:TLCHANGE RESTART

:ANALysis:PMEASURE<x>:MEASURE:CYCLE?

機能 電力測定の自動測定のサイクル統計処理に関するすべての設定値を問い合わせます。
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

:ANALysis:PMEASURE<x>:MEASURE:CYCLE:ABORt

機能 電力測定の自動測定のサイクル統計処理の実行を中止します。

構文 :ANALysis:PMEASURE<x>:MEASURE:CYCLE:
ABORt
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:MEASURE:CYCLE:
ABORT

:ANALysis:PMEASURE<x>:MEASURE:CYCLE:EXECute

機能 電力測定の自動測定のサイクル統計処理を実行します。

構文 :ANALysis:PMEASURE<x>:MEASURE:CYCLE:
EXECute
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:MEASURE:CYCLE:
EXECUTE

:ANALysis:PMEASURE<x>:MEASURE:CYCLE:TRACe

機能 電力測定の自動測定のサイクル統計処理の周期対象波形を設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:MEASURE:CYCLE:
TRACe {<NRf>|MATH<y>}
:ANALysis:PMEASURE<x>:MEASURE:CYCLE:
TRACe?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)
<NRf> = 1 ~ 8 (4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8 (4ch モデルでは 1 ~ 4)

例 :ANALYSIS:PMEASURE1:MEASURE:CYCLE:
TRACE 1
:ANALYSIS:PMEASURE1:MEASURE:CYCLE:
TRACE?
-> :ANALYSIS:PMEASURE1:MEASURE:
CYCLE:TRACE 1

:ANALysis:PMEASURE<x>:MEASURE:HISTorY:ABORt

機能 電力測定の自動測定のヒストリ波形の統計処理を中止します。

構文 :ANALysis:PMEASURE<x>:MEASURE:
HISTorY:ABORt
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:MEASURE:HISTORY:
ABORT

:ANALysis:PMEASURE<x>:MEASURE:HISTorY:EXECute

機能 電力測定の自動測定のヒストリ波形の統計処理を実行します。

構文 :ANALysis:PMEASURE<x>:MEASURE:
HISTorY:EXECute
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:MEASURE:HISTORY:
EXECUTE

:ANALysis:PMEASURE<x>:MEASURE:MODE

機能 電力測定の自動測定の ON/OFF/ 統計処理を設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:MEASURE:
MODE {OFF|CONTInuous|CYCLE|HISTorY}
:ANALysis:PMEASURE<x>:MEASURE:MODE?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:MEASURE:MODE OFF
:ANALYSIS:PMEASURE1:MEASURE:MODE?
-> :ANALYSIS:PMEASURE1:MEASURE:
MODE OFF

:ANALysis:PMEASURE<x>:MEASURE:ZLINKage

機能 電力測定の実動測定の測定結果の結果番号とズーム位置との連動 (ON/OFF) を設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:MEASURE:
ZLINKage {<Boolean>}
:
:
:
ZLINKage?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:MEASURE:
ZLINKAGE ON
:
:
:
ZLINKAGE?
-> :ANALYSIS:PMEASURE1:MEASURE:
ZLINKAGE ON

:ANALysis:PMEASURE<x>:MODE

機能 電力測定の実動測定の ON/OFF を設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:
MODE {<Boolean>}
:
:
:
MODE?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:MODE ON
:
:
:
MODE?
-> :ANALYSIS:PMEASURE1:MODE 1

:ANALysis:PMEASURE<x>:RANGE

機能 測定対象ウィンドウを設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:RANGE {MAIN|
Z1|Z2}
:
:
:
RANGE?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:RANGE MAIN
:
:
:
RANGE?
-> :ANALYSIS:PMEASURE1:RANGE MAIN

:ANALysis:PMEASURE<x>:SETup?

機能 電力測定の実動測定の入力に関するすべての設定値を問い合わせします。

構文 :ANALysis:PMEASURE<x>:SETup?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:SETUP?

:ANALysis:PMEASURE<x>:SETup:ADESkew

機能 電力測定の実動測定のオートスキュー補正を実行します。

構文 :ANALysis:PMEASURE<x>:SETup:ADESkew
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:SETUP:ADESKEW

:ANALysis:PMEASURE<x>:SETup:I?

機能 電力測定の実動測定の電流入力チャネルに関するすべての設定値を問い合わせします。

構文 :ANALysis:PMEASURE<x>:SETup:I?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:SETUP:I?

:ANALysis:PMEASURE<x>:SETup:I:DESKew

機能 電力測定の実動測定の電流入力チャネルのスキュー補正を設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:SETup:I:
DESKew {<時間>}
:
:
:
DESKew?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

<時間> = -100.0ns ~ 100.0ns、10ps ステップ
例 :ANALYSIS:PMEASURE1:SETUP:I:
DESKew 1NS
:
:
:
DESKew?
-> :ANALYSIS:PMEASURE1:SETUP:I:
DESKew 1.000E-09

:ANALysis:PMEASURE<x>:SETup:I:PROBe

機能 電力測定の実動測定の電流入力チャネルのプロブの電流 - 電圧換算比を設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:SETup:I:
PROBe {C0_001|C0_002|C0_005|C0_01|
C0_02|C0_05|C0_1|C0_2|C0_5|C1|C2
|C5|
C10|C20|C50|C100|C200|C500|C1000|
C2000}
:
:
:
PROBe?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:SETUP:I:
PROBE C0_001
:
:
:
PROBE?
-> :ANALYSIS:PMEASURE1:SETUP:I:
PROBE C0_001

:ANALysis:PMEASURE<x>:SETup:RTRace

機能 電力測定の実動測定のスキュー補正の対象トレースを設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:SETup:
RTRace {I|U}
:
:
:
RTRace?
<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:SETUP:RTRACE I
:
:
:
RTRACE?
-> :ANALYSIS:PMEASURE1:SETUP:
RTRACE I

5.3 ANALysis グループ

:ANALysis:PMEASURE<x>:SETup:U?	
機能	電力測定の入力チャネルに関するすべての設定値を問い合わせます。
構文	:ANALysis:PMEASURE<x>:SETup:U? <x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)
例	:ANALYSIS:PMEASURE1:SETUP:U?
:ANALysis:PMEASURE<x>:SETup:U:DESKew	
機能	電力測定の入力チャネルのスキュー補正を設定 / 問い合わせします。
構文	:ANALysis:PMEASURE<x>:SETup:U: DESKew {< 時間 >} :ANALysis:PMEASURE<x>:SETup:U: DESKew? <x> = 1 ~ 4 (4ch モデルでは 1 ~ 2) < 時間 > = -100.0ns ~ 100.0ns、10ps ステップ
例	:ANALYSIS:PMEASURE1:SETUP:U: DESKew 1ns :ANALYSIS:PMEASURE1:SETUP:U:DESKew? -> :ANALYSIS:PMEASURE1:SETUP:U: DESKew 1.000E-09
:ANALysis:PMEASURE<x>:SETup:U:PROBE	
機能	電力測定の入力チャネルのプロープの減衰比を設定 / 問い合わせします。
構文	:ANALysis:PMEASURE<x>:SETup:U: PROBE {<Nrf>} :ANALysis:PMEASURE<x>:SETup:U:PROBE? <x> = 1 ~ 4 (4ch モデルでは 1 ~ 2) <Nrf> = 本体ユーザーズマニュアル機能編参照。
例	:ANALYSIS:PMEASURE1:SETUP:U:PROBE 1 :ANALYSIS:PMEASURE1:SETUP:U:PROBE? -> :ANALYSIS:PMEASURE1:SETUP:U: PROBE 1.000
:ANALysis:PMEASURE<x>:TRANGE	
機能	測定範囲を設定 / 問い合わせします。
構文	:ANALysis:PMEASURE<x>:TRANGE {<Nrf>, <Nrf>} :ANALysis:PMEASURE<x>:TRANGE? <x> = 1 ~ 4 (4ch モデルでは 1 ~ 2) <Nrf>、<Nrf> = -5 ~ 5(div)、(10div/ 表示レコード長) ステップ
例	:ANALYSIS:PMEASURE1:TRANGE 5,-5 :ANALYSIS:PMEASURE1:TRANGE? -> :ANALYSIS:PMEASURE1: TRANGE 5.0000000E+00,-5.0000000E+00
:ANALysis:PMEASURE<x>:UDPROximal?	
機能	ディスタル・メシアル・プロキシマルに関するすべての設定値を問い合わせます。
構文	:ANALysis:PMEASURE<x>:UDPROximal? <x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)
例	:ANALYSIS:PMEASURE1:UDPROXIMAL?

:ANALysis:PMEASURE<x>:UDPROximal:MODE	
機能	ディスタル・メシアル・プロキシマル点のモードを設定 / 問い合わせします。
構文	:ANALysis:PMEASURE<x>:UDPROximal: MODE {PERCent UNIT} :ANALysis:PMEASURE<x>:UDPROximal: MODE? <x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)
例	:ANALYSIS:PMEASURE1:UDPROXIMAL: MODE PERCENT :ANALYSIS:PMEASURE1:UDPROXIMAL:MODE? -> :ANALYSIS:PMEASURE1:UDPROXIMAL: MODE PERCENT
:ANALysis:PMEASURE<x>:UDPROximal:PERCent	
機能	ディスタル・メシアル・プロキシマル点を % で設定 / 問い合わせします。
構文	:ANALysis:PMEASURE<x>:UDPROximal: PERCent {<Nrf>,<Nrf>,<Nrf>} :ANALysis:PMEASURE<x>:UDPROximal: PERCent? <x> = 1 ~ 4 (4ch モデルでは 1 ~ 2) <Nrf>、<Nrf>、<Nrf> = 0(%) ~ 100(%)、1% ステップ
例	:ANALYSIS:PMEASURE1:UDPROXIMAL: PERCENT 10,50,90 :ANALYSIS:PMEASURE1:UDPROXIMAL: PERCENT? -> :ANALYSIS:PMEASURE1:UDPROXIMAL: PERCENT 10,50,90
:ANALysis:PMEASURE<x>:UDPROximal:UNIT	
機能	ディスタル・メシアル・プロキシマル点を電圧値で設定 / 問い合わせします。
構文	:ANALysis:PMEASURE<x>:UDPROximal: UNIT {<Nrf>,<Nrf>,<Nrf>} :ANALysis:PMEASURE<x>:UDPROximal: UNIT? <x> = 1 ~ 4 (4ch モデルでは 1 ~ 2) <Nrf>、<Nrf>、<Nrf> = 本体ユーザーズマニュアル機能編参照
例	:ANALYSIS:PMEASURE1:UDPROXIMAL: UNIT -1,0,1 :ANALYSIS:PMEASURE1:UDPROXIMAL:UNIT? -> :ANALYSIS:PMEASURE1:UDPROXIMAL: UNIT -1.0000000E+00,0.0000000E+00, 1.0000000E+00

:ANALysis:PMEASURE<x>:UMETHOD

機能 High・Low 点を設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:UMETHOD {AUTO|MAXimum|HISTogram}

:ANALysis:PMEASURE<x>:UMETHOD?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:UMETHOD AUTO

:ANALYSIS:PMEASURE1:UMETHOD?

-> :ANALYSIS:PMEASURE1:UMETHOD AUTO

:ANALysis:PMEASURE<x>:UNIT

機能 電力量の単位を設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:UNIT {J|WH}

:ANALysis:PMEASURE<x>:UNIT?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

例 :ANALYSIS:PMEASURE1:UNIT J

:ANALYSIS:PMEASURE1:UNIT?

-> :ANALYSIS:PMEASURE1:UNIT J

:ANALysis:PMEASURE<x>:USER<y>?

機能 Calc アイテ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:PMEASURE<x>:USER<y>?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

<y> = 1 ~ 4

例 :ANALYSIS:PMEASURE:USER?

:ANALysis:PMEASURE<x>:USER<y>:COUNT?

機能 Calc アイテムの自動測定値の統計処理の回数を問い合わせます。

構文 :ANALysis:PMEASURE<x>:USER<y>:COUNT?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

<y> = 1 ~ 4

例 :ANALYSIS:PMEASURE:USER:COUNT?

-> :ANALYSIS:PMEASURE:USER:COUNT 1

:ANALysis:PMEASURE<x>:USER<y>:DEFINE

機能 Calc アイテムの自動測定値の演算式を設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:USER<y>:DEFINE {<文字列>}

:ANALysis:PMEASURE<x>:USER<y>:DEFINE?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

<y> = 1 ~ 4

<文字列> = 128 文字以内

例 :ANALYSIS:PMEASURE:USER:DEFINE "ABC"

:ANALYSIS:PMEASURE:USER:DEFINE?

-> :ANALYSIS:PMEASURE:USER:DEFINE "ABC"

:ANALysis:PMEASURE<x>:USER<y>:{MAXimum|MEAN|MINimum|SDEVIation}?

機能 Calc アイテムの自動測定値の各統計値を問い合わせます。

構文 :ANALysis:PMEASURE<x>:USER<y>:{MAXimum|MEAN|MINimum|SDEVIation}?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

<y> = 1 ~ 4

例 :ANALYSIS:PMEASURE:USER:MAXIMUM?

-> :ANALYSIS:PMEASURE:USER:MAXIMUM 0.0

統計値がとれない場合は、「NAN(Not A Number)」が返されます。

:ANALysis:PMEASURE<x>:USER<y>:NAME

機能 Calc アイテムの名称を設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:USER<y>:NAME {<文字列>}

:ANALysis:PMEASURE<x>:USER<y>:NAME?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

<y> = 1 ~ 4

<文字列> = 8 文字以内

例 :ANALYSIS:PMEASURE:USER:NAME "ABC"

:ANALYSIS:PMEASURE:USER:NAME?

-> :ANALYSIS:PMEASURE:USER:NAME "ABC"

:ANALysis:PMEASURE<x>:USER<y>:STATE

機能 Calc アイテムの自動測定の ON/OFF を設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:USER<y>:STATE {<Boolean>}

:ANALysis:PMEASURE<x>:USER<y>:STATE?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

<y> = 1 ~ 4

例 :ANALYSIS:PMEASURE:USER:STATE ON

:ANALYSIS:PMEASURE:USER:STATE?

-> :ANALYSIS:PMEASURE:USER:STATE 1

:ANALysis:PMEASURE<x>:USER<y>:UNIT

機能 Calc アイテムの単位を設定 / 問い合わせします。

構文 :ANALysis:PMEASURE<x>:USER<y>:UNIT {<文字列>}

:ANALysis:PMEASURE<x>:USER<y>:UNIT?

<x> = 1 ~ 4 (4ch モデルでは 1 ~ 2)

<y> = 1 ~ 4

<文字列> = 4 文字以内

例 :ANALYSIS:PMEASURE:USER:UNIT "ABC"

:ANALYSIS:PMEASURE:USER:UNIT?

-> :ANALYSIS:PMEASURE:USER:UNIT "ABC"

5.3 ANALysis グループ

:ANALysis:PMEasure<x>:USER<y>:VALue?

機能	Calc アイテムの自動測定値の測定値を問い合わせます。
構文	<pre>:ANALysis:PMEasure<x>:USER<y>:VALue? {<NRf>} <x> = 1 ~ 4 (4ch モデルでは 1 ~ 2) <y> = 1 ~ 4 <NRf> = 1 ~ 100000</pre>
例	<pre>:ANALYSIS:PMEASURE:USER:VALUE? -> :ANALYSIS:PMEASURE:USER:VALUE 0.0</pre>
解説	- 測定不可能な場合は、「NAN(Not A Number)」が返されます。 <NRf> は、自動測定を実行してから <NRf> 回の測定値を個々に問い合わせるときに使用します。 <NRf> に「1」を設定すると、自動測定値のメモリの中の最も古い測定値を問い合わせます。 - 設定した回数に測定値が存在しないときは、「NAN(Not A Number)」が返されます。 <NRf> を省略すると、最新の測定値を問い合わせます。

:ANALysis: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	<pre>:ANALysis:WAIT? {<NRf>} <NRf> = 1 ~ 36000(タイムアウト時間、100ms)</pre>
例	<pre>:ANALYSIS:WAIT? -> :ANALYSIS:WAIT 1</pre>
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。

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?

: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}

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 :CHANnel<x>?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

: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 ~ 8(4ch モデルでは 1 ~ 4)

例 CHANNEL1:ASCALE:EXECUTE
解説 チャンネルごとにオートスケーリングが設定できます。

- V/div
垂直ポジションの位置を変えずに、波形の全振幅が見えるように表示されます。
- Offset
入力カップリングが AC のとき
0V
入力カップリングが DC のとき
Center = (Max+Min)/2
- Trig Level
DC オフセット位置

:CHANnel<x>:BWIDth

機能 各チャンネルの入力フィルターを設定 / 問い合わせします。

構文 :CHANnel<x>:BWIDth {FULL|<周波数>}
:CHANnel<x>:BWIDth?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
{<周波数>} = 本体ユーザーズマニュアル機能編参照。

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}
:CHANnel<x>:COUPling?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :CHANNEL1:COUPLING AC
:CHANNEL1:COUPLING?
-> :CHANNEL1:COUPLING AC

: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 ~ 8(4ch モデルでは 1 ~ 4)
<時間> = -1000.00ns ~ 1000.00ns, 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 ~ 8(4ch モデルでは 1 ~ 4)

例 :CHANNEL1:DISPLAY ON
:CHANNEL1:DISPLAY?
-> :CHANNEL1:DISPLAY 1

:CHANnel<x>:INVert

機能 インバートモード (波形の反転表示) 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 ~ 8(4ch モデルでは 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 ~ 8(4ch モデルでは 1 ~ 4)

: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 ~ 8(4ch モデルでは 1 ~ 4)
<文字列> = 8 文字以内

例 :CHANNEL1:LABEL:DEFINE "CH1"
:CHANNEL1:LABEL:DEFINE?
-> :CHANNEL1:LABEL:DEFINE "CH1"

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。

5.6 CHANnel グループ

:CHANnel<x>:LABel:DISPlay

機能 各チャンネルの波形ラベル名の表示の ON/OFF を設定 / 問い合わせします。

構文 :CHANnel<x>:LABel:
DISPlay {<Boolean>}
:CHANnel<x>:LABel:DISPlay?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :CHANNEL1:LABEL:DISPLAY ON
:CHANNEL1:LABEL:DISPLAY?
-> :CHANNEL1:LABEL:DISPLAY 1

:CHANnel<x>:LSCale?

機能 各チャンネルのリニア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 :CHANnel<x>:LSCale?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:CHANnel<x>:LSCale:AVALue

機能 スケーリング係数 A を設定 / 問い合わせします。

構文 :CHANnel<x>:LSCale:AVALue {<NRf>}
:CHANnel<x>:LSCale:AVALue?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = -1.0000E+31 ~ 1.0000E+31

例 :CHANNEL1:LSCALE:AVALUE 10
:CHANNEL1:LSCALE:AVALUE?
-> :CHANNEL1:LSCALE:
AVALUE 10.0000E+00

:CHANnel<x>:LSCale:BVALue

機能 オフセット値 B を設定 / 問い合わせします。

構文 :CHANnel<x>:LSCale:BVALue {<NRf>}
:CHANnel<x>:LSCale:BVALue?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = -1.0000E+31 ~ 1.0000E+31

例 :CHANNEL1:LSCALE:BVALUE 10
:CHANNEL1:LSCALE:BVALUE?
-> :CHANNEL1:LSCALE:
BVALUE 10.0000E+00

:CHANnel<x>:LSCale:MODE

機能 リニアスケーリングの ON/OFF を設定 / 問い合わせします。

構文 :CHANnel<x>:LSCale:MODE {<Boolean>}
:CHANnel<x>:LSCale:MODE?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :CHANNEL1:LSCALE:MODE ON
:CHANNEL1:LSCALE:MODE?
-> :CHANNEL1:LSCALE:MODE 1

:CHANnel<x>:LSCale:UNIT

機能 リニアスケーリング結果に付加する単位を設定 / 問い合わせします。

構文 :CHANnel<x>:LSCale:UNIT {<文字列>}
:CHANnel<x>:LSCale:UNIT?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<文字列> = 4 文字以内

例 :CHANNEL1:LSCALE:UNIT "EU"
:CHANNEL1:LSCALE:UNIT?
-> :CHANNEL1:LSCALE:UNIT "EU"

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。単位によってスケール値に影響をおよぼすことはありません。

: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 ~ 8(4ch モデルでは 1 ~ 4)
<電圧>、<電流> = 本体ユーザズマニュアル機能編参照。

例 :CHANNEL1:OFFSET 0V
:CHANNEL1:OFFSET?
-> :CHANNEL1:OFFSET 0.000E+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 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = -4 ~ 4(div)

例 :CHANNEL1:POSITION 1
:CHANNEL1:POSITION?
-> :CHANNEL1:POSITION 1.00E+00

:CHANnel<x>:PROBe?

機能 各チャンネルのプロープの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 ~ 8(4ch モデルでは 1 ~ 4)

:CHANnel<x>:PROBe:DZCalibrate

機能 各チャンネルの消磁とゼロ補正を実行します。

構文 :CHANnel<x>:PROBe:DZCalibrate
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :CHANNEL1:PROBE:DZCALIBRATE

:CHANnel<x>:PROBe[:MODE]

機能 各チャンネルのプローブの減衰比を設定 / 問い合わせします。

構文 :CHANnel<x>:PROBe[:MODE] {<Nrf>|
C0 _ 001|C0 _ 002|C0 _ 005|C0 _ 01|C0 _ 02|
C0 _ 05|C0 _ 1|C0 _ 2|C0 _ 5|C1|C2|C5|C10|
C20|C50|C100|C200|C500|C1000|C2000}
:CHANnel<x>:PROBe[:MODE]?

例 <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
:CHANNEL1:PROBE:MODE C0 _ 001
:CHANNEL1:PROBE:MODE?
-> :CHANNEL1:PROBE:MODE C0 _ 001

:CHANnel<x>:PROBe:PZCalibrate

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>:PROBe:PZCalibrate
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :CHANNEL1:PROBE:PZCALIBRATE

:CHANnel<x>:VARiable

機能 各チャンネルの電圧軸感度を V/div の 0.01 ステップで設定 / 問い合わせします。

構文 :CHANnel<x>:VARiable {< 電圧 >|< 電流 >}
:CHANnel<x>:VARiable?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
< 電圧 >、< 電流 > = 本体ユーザーズマニュアル
操作編参照。

例 :CHANNEL1:VARIABLE 1V
:CHANNEL1:VARIABLE?
-> :CHANNEL1:VARIABLE 1.000E+00

:CHANnel<x>:VDIV

機能 各チャンネルの電圧軸感度 (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 ~ 8(4ch モデルでは 1 ~ 4)
< 電圧 >、< 電流 > = 本体ユーザーズマニュアル
操作編参照。

例 :CHANNEL1:VDIV 2V
:CHANNEL1:VDIV?
-> :CHANNEL1:VDIV 2.000E+00

5.7 CHUTil グループ

:CHUTil?

機能 チャンネル間の設定コピーに関するすべての設定値を問い合わせます。

構文 :CHUTil?

:CHUTil:ALL:DISPlay

機能 全チャンネルの表示の ON/OFF を設定します。

構文 :CHUTil:ALL:DISPlay {<Boolean>}

:CHUTil:COPIch?

機能 チャンネル間の設定コピーに関するすべての設定値を問い合わせます。

構文 :CHUTil:COPIch?

:CHUTil:COPIch:EXECute

機能 チャンネル間の設定コピーを実行します。

構文 :CHUTil:COPIch:EXECute

例 :CHUTIL:COPICH:EXECUTE

:CHUTil:COPIch:SOURce

機能 コピー対象のチャンネルを設定 / 問い合わせします。

構文 :CHUTil:COPIch:SOURce {<NRf>}

:CHUTil:COPIch:SOURce?

<NRf> = 1 ~ 8

例 :CHUTIL:COPICH:SOURCE 1

:CHUTIL:COPICH:SOURCE?

-> :CHUTIL:COPICH:SOURCE 1

:CHUTil:COPIch:DESTination?

機能 コピー先の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 :CHUTil:COPIch:DESTination?

:CHUTil:COPIch:DESTination:ALL

機能 コピー先のチャンネルを全チャンネルとする (ON)、しない (OFF) を設定します。

構文 :CHUTil:COPIch:DESTination:

ALL {<Boolean>}

例 :CHUTIL:COPICH:DESTINATION:ALL ON

:CHUTIL:COPICH:DESTINATION:ALL?

-> :CHUTIL:COPICH:DESTINATION:ALL 1

:CHUTil:COPIch:DESTination:CHANnel<x>

機能 チャンネル間の設定コピーのコピー先に設定する (ON)、しない (OFF) を設定 / 問い合わせします。

構文 :CHUTil:COPIch:DESTination:

CHANnel <x> {<Boolean>}

:CHUTil:COPIch:DESTination:

CHANnel <x>?

<x> = 1 ~ 8

例

:CHUTIL:COPICH:DESTINATION:

CHANNEL1 ON

:CHUTIL:COPICH:DESTINATION:CHANNEL1?

-> :CHUTIL:COPICH:DESTINATION:

CHANNEL1 1

:CHUTil:COPIch:UNDO

機能 チャンネル間の設定コピーの実行を取り消します。

構文 :CHUTil:COPIch:UNDO

例 :CHUTIL:COPICH:UNDO

5.8 CLEar グループ

:CLEar

機能 クリアトレースを実行します。

構文 :CLEar

例 :CLEAR

5.9 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?

: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:PROBE:MODE 10)、付けずに返送するか (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

:COMMunicate:OPSE (Operation Pending Status Enable register)

機能 *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。

構文 :COMMunicate:OPSE <Register>
:COMMunicate:OPSE?
<Register> = 0 ~ 65535 (:COMMunicate:WAIT? 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、すべてのオーバー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 だけが 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:WAIT? コマンドの図を参照してください。

:COMMunicate:OVERlap

機能 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。

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、すべてのオーバー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 だけが 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、4-8 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、すべてのオーバーラップコマンドを対象にしています (:COMMunicate:WAIT? コマンドの図参照)。

: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

: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:PROBE:MODE 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、4-9 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

解説 オペレーションペンディングステータスレジス
タ / オーバー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:

エッジ / パターン検索実行未完了

5.10 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?

:CURSor[:TY]?

機能 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[:TY]?

:CURSor[:TY]:DEGREE?

機能 角度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[:TY]:DEGREE?

:CURSor[:TY]:DEGREE:ALL

機能 T-Y 表示の角度カーソルの測定値を一括 ON/OFF します

構文 :CURSor[:TY]:DEGREE:ALL {<Boolean>}

例 :CURSOR:TY:DEGREE:ALL ON

:CURSor[:TY]:DEGREE:D<x>?

機能 角度カーソルの角度に関するすべての設定値を問い合わせます。

構文 :CURSor[:TY]:DEGREE:D<x>?

<x> = 1 ~ 2

:CURSor[:TY]:DEGREE:D<x>:STATE

機能 角度カーソルの角度の ON/OFF を設定 / 問い合わせします。

構文 :CURSor[:TY]:DEGREE:D<x>:

STATE {<Boolean>}

:CURSor[:TY]:DEGREE:D<x>:STATE?

<x> = 1 ~ 2

例 :CURSOR:TY:DEGREE:D1:STATE ON

:CURSOR:TY:DEGREE:D1:STATE?

-> :CURSOR:TY:DEGREE:D1:STATE 1

:CURSor[:TY]:DEGREE:D<x>:VALUE?

機能 角度カーソルの角度を問い合わせます。

構文 :CURSor[:TY]:DEGREE:D<x>:VALUE?

<x> = 1 ~ 2

例 :CURSOR:TY:DEGREE:D1:VALUE?

-> :CURSOR:TY:DEGREE:D1:

VALUE -120.00000E+00

:CURSor[:TY]:DEGREE:DD?

機能 角度カーソル間の角度差に関するすべての設定値を問い合わせます。

構文 :CURSor[:TY]:DEGREE:DD?

:CURSor[:TY]:DEGREE:DD:STATE

機能 角度カーソル間の角度差 ΔD 値の ON/OFF を設定 / 問い合わせします。

構文 :CURSor[:TY]:DEGREE:DD:

STATE {<Boolean>}

:CURSor[:TY]:DEGREE:DD:STATE?

例 :CURSOR:TY:DEGREE:DD:STATE ON

:CURSOR:TY:DEGREE:DD:STATE?

-> :CURSOR:TY:DEGREE:DD:STATE 1

:CURSor[:TY]:DEGREE:DD:VALUE?

機能 角度カーソル間の角度差 ΔD 値を問い合わせます。

構文 :CURSor[:TY]:DEGREE:DD:VALUE?

例 :CURSOR:TY:DEGREE:DD:VALUE?

-> :CURSOR:TY:DEGREE:DD:

VALUE 180.00000E+00

:CURSor[:TY]:DEGREE:DV?

機能 角度カーソル間の ΔV に関するすべての設定値を問い合わせます。

構文 :CURSor[:TY]:DEGREE:DV?

:CURSor[:TY]:DEGREE:DV:STATE

機能 角度カーソル間の ΔV 値の ON/OFF を設定 / 問い合わせします。

構文 :CURSor[:TY]:DEGREE:DV:

STATE {<Boolean>}

:CURSor[:TY]:DEGREE:DV:STATE?

例 :CURSOR:TY:DEGREE:DV:STATE ON

:CURSOR:TY:DEGREE:DV:STATE?

-> :CURSOR:TY:DEGREE:DV:STATE 1

:CURSor[:TY]:DEGREE:DV:VALue?

機能 角度カーソル間の ΔV 値を問い合わせます。

構文 :CURSor[:TY]:DEGREE:DV:
VALue? {[V12]|V21}

例 :CURSOR:TY:DEGREE:DV:VALue?
-> :CURSOR:TY:DEGREE:DV:
VALUE -6.2500000E-03
:CURSOR:TY:DEGREE:DV:VALue? V21
-> :CURSOR:TY:DEGREE:DV:
VALUE 6.2500000E-03

解説

- 選択肢「V12」「V21」の両方を省略可能です。省略したときは V1-V2 の計算結果を問い合わせます。
- 「V12」選択時は V1-V2 の計算結果を問い合わせます。
- 「V21」選択時は V2-V1 の計算結果を問い合わせます。
- リニアスケーリングが ON のときは、スケーリング値の問い合わせになります。

:CURSor[:TY]:DEGREE:JUMP

機能 T-Y 表示の角度カーソルを指定したズーム波形上にジャンプさせます。

構文 :CURSor[:TY]:DEGREE:JUMP {C1 _ Z1|
C1 _ Z2|C2 _ Z1|C2 _ Z2}

例 :CURSOR:TY:DEGREE:JUMP C1 _ Z1

解説 ズーム波形の中央位置にジャンプします。C1、C2 は、Cursor1、Cursor2 を示します。

:CURSor[:TY]:DEGREE:POSition<x>

機能 角度カーソルの位置を設定 / 問い合わせします。

構文 :CURSor[:TY]:DEGREE:
POSition<x> {<Nrf>}
:CURSor[:TY]:DEGREE:POSition<x>?
<x> = 1 ~ 2
<Nrf> = -5 ~ 5(div)、(10div/ 表示レコード長)
ステップ

例 :CURSOR:TY:DEGREE:POSITION1 2
:CURSOR:TY:DEGREE:POSITION1?
-> :CURSOR:TY:DEGREE:
POSITION1 2.0000000

:CURSor[:TY]:DEGREE:REFeRence<x>

機能 基準角度の始点 (Reference1) または終点 (Reference2) の位置を設定 / 問い合わせします。

構文 :CURSor[:TY]:DEGREE:
REFeRence<x> {<Nrf>}
:CURSor[:TY]:DEGREE:REFeRence<x>?
<x> = 1 ~ 2
<Nrf> = -5 ~ 5(div)、(10div/ 表示レコード長)
ステップ

例 :CURSOR:TY:DEGREE:REFERENCE1 -1
:CURSOR:TY:DEGREE:REFERENCE1?
-> :CURSOR:TY:DEGREE:
REFERENCE1 -1.0000000

:CURSor[:TY]:DEGREE:RVALue

機能 基準角度を設定 / 問い合わせします。

構文 :CURSor[:TY]:DEGREE:RVALue {<Nrf>}
:CURSor[:TY]:DEGREE:RVALue?
<Nrf> = 1 ~ 720

例 :CURSOR:TY:DEGREE:RVALUE 180
:CURSOR:TY:DEGREE:RVALUE?
-> :CURSOR:TY:DEGREE:RVALUE 180

:CURSor[:TY]:DEGREE:TRACe

機能 角度カーソルの対象波形を設定 / 問い合わせします。

構文 :CURSor[:TY]:DEGREE:TRACe {<Nrf>|
ALL|MATH<x>}
:CURSor[:TY]:DEGREE:TRACe?
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :CURSOR:TY:DEGREE:TRACE 1
:CURSOR:TY:DEGREE:TRACE?
-> :CURSOR:TY:DEGREE:TRACE 1

:CURSor[:TY]:DEGREE:UNIT

機能 角度カーソルの単位を設定 / 問い合わせします。

構文 :CURSor[:TY]:DEGREE:UNIT {<文字列>}
:CURSor[:TY]:DEGREE:UNIT?

<文字列> = 4 文字以内
例 :CURSOR:TY:DEGREE:UNIT "DEG"
:CURSOR:TY:DEGREE:UNIT?
-> :CURSOR:TY:DEGREE:UNIT "DEG"

:CURSor[:TY]:DEGREE:V<x>?

機能 角度カーソルの電圧に関するすべての設定値を問い合わせます。

構文 :CURSor[:TY]:DEGREE:V<x>?
<x> = 1 ~ 2

:CURSor[:TY]:DEGREE:V<x>:STATe

機能 角度カーソルの電圧の ON/OFF を設定 / 問い合わせします。

構文 :CURSor[:TY]:DEGREE:V<x>:
STATe {<Boolean>}
:CURSor[:TY]:DEGREE:V<x>:STATe?
<x> = 1 ~ 2

例 :CURSOR:TY:DEGREE:V1:STATE ON
:CURSOR:TY:DEGREE:V1:STATE?
-> :CURSOR:TY:DEGREE:V1:STATE 1

5.10 CURSor グループ

:CURSor[:TY]:DEGRee:V<x>:VALue?

機能 角度カーソルの電圧値を問い合わせます。

構文 :CURSor[:TY]:DEGRee:V<x>:VALue?
<x> = 1 ~ 2

例 :CURSOR:TY:DEGREE:V1:VALUE?

-> :CURSOR:TY:DEGREE:V1:

VALUE 10.000000E-03

:CURSor[:TY]:HORizontal?

機能 ΔV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[:TY]:HORizontal?

:CURSor[:TY]:HORizontal:ALL

機能 T-Y表示のΔVカーソルの測定値を一括 ON/OFF します。

構文 :CURSor[:TY]:HORizontal:

ALL {<Boolean>}

例 :CURSOR:TY:HORIZONTAL:ALL ON

:CURSor[:TY]:HORizontal:DV?

機能 ΔVカーソル間の垂直軸値に関するすべての設定値を問い合わせます。

構文 :CURSor[:TY]:HORizontal:DV?

:CURSor[:TY]:HORizontal:DV:STATE

機能 ΔVカーソル間の垂直軸値の ON/OFF を設定 / 問い合わせします。

構文 :CURSor[:TY]:HORizontal:DV:

STATE {<Boolean>}

:CURSor[:TY]:HORizontal:DV:STATE?

例 :CURSOR:TY:HORIZONTAL:DV:STATE ON

:CURSOR:TY:HORIZONTAL:DV:STATE?

-> :CURSOR:TY:HORIZONTAL:DV:STATE 1

:CURSor[:TY]:HORizontal:DV:VALue?

機能 ΔVカーソル間の垂直軸値を問い合わせます。

構文 :CURSor[:TY]:HORizontal:DV:VALue?

例 :CURSOR:TY:HORIZONTAL:DV:VALUE?

-> :CURSOR:TY:HORIZONTAL:DV:

VALUE 3.0000000E+00

解説 リニアスケールリングが ON のときは、スケールリング値の問い合わせになります。

:CURSor[:TY]:HORizontal:DVDT?

機能 ΔVカーソルのΔV/ΔTに関するすべての設定値を問い合わせます。

例 :CURSOR:TY:HORIZONTAL:DVDT?

:CURSor[:TY]:HORizontal:DVDT:STATE

機能 ΔVカーソルのΔV/ΔT値の ON/OFF を設定 / 問い合わせします。

構文 :CURSor[:TY]:HORizontal:DVDT:

STATE {Boolean}

:CURSor[:TY]:HORizontal:DVDT:STATE?

例 :CURSOR:TY:HORIZONTAL:DVDT:STATE ON

:CURSOR:TY:HORIZONTAL:DVDT:STATE?

-> :CURSOR:TY:HORIZONTAL:DVDT:

STATE 1

:CURSor[:TY]:HORizontal:DVDT:VALue?

機能 ΔVカーソルのΔV/ΔT値を問い合わせます。

構文 :CURSor[:TY]:HORizontal:DVDT:VALue?

例 :CURSOR:TY:HORIZONTAL:DVDT:VALUE?

-> :CURSOR:TY:HORIZONTAL:DVDT:

VALUE 1.5E+03

解説 リニアスケールリングが ON のときは、スケールリング値の問い合わせになります。

:CURSor[:TY]:HORizontal:POSition<x>

機能 ΔVカーソルの位置を設定 / 問い合わせします。

構文 :CURSor[:TY]:HORizontal:

POSition<x> {<NRf>}

:CURSor[:TY]:HORizontal:POSition<x>?

<x> = 1 ~ 2

<NRf> = -4 ~ 4(div)、1/100 ステップ

例 :CURSOR:TY:HORIZONTAL:POSITION1 -4

:CURSOR:TY:HORIZONTAL:POSITION1?

-> :CURSOR:TY:HORIZONTAL:

POSITION1 -4

解説 <NRf> は、小数点以下 2 桁まで有効です。

:CURSor[:TY]:HORizontal:TRACe

機能 ΔVカーソルの対象波形を設定 / 問い合わせします。

構文 :CURSor[:TY]:HORizontal:

TRACe {<NRf>|MATH<x>}

:CURSor[:TY]:HORizontal:TRACe?

<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)

<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :CURSOR:TY:HORIZONTAL:TRACE 1

:CURSOR:TY:HORIZONTAL:TRACE?

-> :CURSOR:TY:HORIZONTAL:TRACE 1

:CURSor[:TY]:HORizontal:V<x>?

機能 ΔVカーソル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 :CURSor[:TY]:HORizontal:V<x>?

<x> = 1 ~ 2

:CURSor[:TY]:HORizontal:V<x>:STATe
 機能 Δ V カーソルの垂直軸値の ON/OFF を設定 / 問い合わせします。
 構文 :CURSor[:TY]:HORizontal:V<x>:STATe {<Boolean>}
 :CURSor[:TY]:HORizontal:V<x>:STATe?
 <x> = 1 ~ 2
 例 :CURSOR:TY:HORIZONTAL:V1:STATE ON
 :CURSOR:TY:HORIZONTAL:V1:STATE?
 -> :CURSOR:TY:HORIZONTAL:V1:STATE 1

:CURSor[:TY]:HORizontal:V<x>:VALue?
 機能 Δ V カーソルの垂直軸値を問い合わせます。
 構文 :CURSor[:TY]:HORizontal:V<x>:VALue?
 <x> = 1 ~ 2
 例 :CURSOR:TY:HORIZONTAL:V1:VALUE?
 -> :CURSOR:TY:HORIZONTAL:V1:VALUE -1.5000000E+00
 解説 リニアスケールリングが ON のときはスケールリング値の問い合わせになります。

:CURSor[:TY]:MARKer?
 機能 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[:TY]:MARKer?

:CURSor[:TY]:MARKer:FORM
 機能 マーカーカーソルの表示形式を設定 / 問い合わせします。
 構文 :CURSor[:TY]:MARKer:FORM {LINE|MARK}
 :CURSor[:TY]:MARKer:FORM?
 例 :CURSOR:TY:MARKER:FORM LINE
 :CURSOR:TY:MARKER:FORM?
 -> :CURSOR:TY:MARKER:FORM LINE

:CURSor[:TY]:MARKer:M<x>?
 機能 指定した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[:TY]:MARKer:M<x>?
 <x> = 1 ~ 4

:CURSor[:TY]:MARKer:M<x>:ALL
 機能 T-Y 表示のマーカーカーソルの測定値を一括 ON/OFF します。
 構文 :CURSor[:TY]:MARKer:M<x>:ALL {<Boolean>}
 <x> = 1 ~ 4
 例 :CURSOR:TY:MARKER:M1:ALL ON

:CURSor[:TY]:MARKer:M<x>:DT<y>?
 機能 マーカーカーソル間の時間軸に関するすべての設定値を問い合わせます。
 構文 :CURSor[:TY]:MARKer:M<x>:DT<y>?
 <x> = 1 ~ 4
 <y> = 1 ~ 4

:CURSor[:TY]:MARKer:M<x>:DT<y>:STATe
 機能 マーカーカーソル間の時間軸値の ON/OFF を設定 / 問い合わせします。
 構文 :CURSor[:TY]:MARKer:M<x>:DT<y>:STATe {<Boolean>}
 :CURSor[:TY]:MARKer:M<x>:DT<y>:STATe?
 <x> = 1 ~ 4
 <y> = 1 ~ 4
 例 :CURSOR:TY:MARKER:M:DT1:STATE ON
 :CURSOR:TY:MARKER:M:DT1:STATE?
 -> :CURSOR:TY:MARKER:M:DT1:STATE 1

:CURSor[:TY]:MARKer:M<x>:DT<y>:VALue?
 機能 マーカーカーソル間の時間軸値を問い合わせます。
 構文 :CURSor[:TY]:MARKer:M<x>:DT<y>:VALue?
 <x> = 1 ~ 4
 <y> = 1 ~ 4
 例 :CURSOR:TY:MARKER:M:DT1:VALUE?
 -> :CURSOR:TY:MARKER:M:DT1:VALUE 0.0000000E+00

:CURSor[:TY]:MARKer:M<x>:DV<y>?
 機能 マーカーカーソル間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 :CURSor[:TY]:MARKer:M<x>:DV<y>?
 <x> = 1 ~ 4
 <y> = 1 ~ 4

:CURSor[:TY]:MARKer:M<x>:DV<y>:STATe
 機能 マーカーカーソル間の垂直軸値の ON/OFF を設定 / 問い合わせします。
 構文 :CURSor[:TY]:MARKer:M<x>:DV<y>:STATe {<Boolean>}
 :CURSor[:TY]:MARKer:M<x>:DV<y>:STATe?
 <x> = 1 ~ 4
 <y> = 1 ~ 4
 例 :CURSOR:TY:MARKER:M:DV1:STATE ON
 :CURSOR:TY:MARKER:M:DV1:STATE?
 -> :CURSOR:TY:MARKER:M:DV1:STATE 1

5.10 CURSor グループ

:CURSor[:TY]:MARKer:M<x>:DV<y>:VALue?

機能	マーカーカーソル間の垂直軸値を問い合わせます。
構文	:CURSor[:TY]:MARKer:M<x>:DV<y>:VALue? <x> = 1 ~ 4 <y> = 1 ~ 4
例	:CURSOR:TY:MARKER:M:DV1:VALUE? -> :CURSOR:TY:MARKER:M:DV1:VALUE 500.00000E-03
解説	リニアスケールリングが ON のときはスケールリング値の問い合わせになります。

:CURSor[:TY]:MARKer:M<x>:JUMP

機能	T-Y 表示のマーカーカーソルを指定したズーム波形上にジャンプさせます。
構文	:CURSor[:TY]:MARKer:M<x>:JUMP {Z1 Z2} <x> = 1 ~ 4
例	:CURSOR:TY:MARKER:M1:JUMP Z1
解説	ズーム波形の中央位置にジャンプします。

:CURSor[:TY]:MARKer:M<x>:POSition

機能	マーカーカーソルの時間軸値を設定 / 問い合わせします。
構文	:CURSor[:TY]:MARKer:M<x>:Position {<Nrf>} :CURSor[:TY]:MARKer:M<x>:POSition? <x> = 1 ~ 4 <Nrf> = -5 ~ 5(div)、(10div/ 表示レコード長) ステップ
例	:CURSOR:TY:MARKER:M1:POSITION -1 :CURSOR:TY:MARKER:M1:POSITION? -> :CURSOR:TY:MARKER:M1:POSITION -1.0000000

:CURSor[:TY]:MARKer:M<x>:T?

機能	マーカーカーソルの時間軸に関するすべての設定値を問い合わせます。
構文	:CURSor[:TY]:MARKer:M<x>:T? <x> = 1 ~ 4

:CURSor[:TY]:MARKer:M<x>:T:STATE

機能	マーカーカーソルの時間軸値の ON/OFF を設定 / 問い合わせします。
構文	:CURSor[:TY]:MARKer:M<x>:T:STATE {<Boolean>} :CURSor[:TY]:MARKer:M<x>:T:STATE? <x> = 1 ~ 4
例	:CURSOR:TY:MARKER:M1:T:STATE ON :CURSOR:TY:MARKER:M1:T:STATE? -> :CURSOR:TY:MARKER:M1:T:STATE 1

:CURSor[:TY]:MARKer:M<x>:T:VALue?

機能	マーカーカーソルの時間軸値を問い合わせます。
構文	:CURSor[:TY]:MARKer:M<x>:T:VALue? <x> = 1 ~ 4
例	:CURSOR:TY:MARKER:M1:T:VALUE? -> :CURSOR:TY:MARKER:M1:T:VALUE -4.5000E-03

:CURSor[:TY]:MARKer:M<x>:TRACE

機能	マーカーカーソルの対象波形を設定 / 問い合わせします。
構文	:CURSor[:TY]:MARKer:M<x>:TRACE {<Nrf> MATH<y> OFF} :CURSor[:TY]:MARKer:M<x>:TRACE? <x> = 1 ~ 4 <Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4) <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例	:CURSOR:TY:MARKER:M1:TRACE 1 :CURSOR:TY:MARKER:M1:TRACE? -> :CURSOR:TY:MARKER:M1:TRACE 1

:CURSor[:TY]:MARKer:M<x>:V?

機能	マーカーカーソル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	:CURSor[:TY]:MARKer:M<x>:V? <x> = 1 ~ 4

:CURSor[:TY]:MARKer:M<x>:V:STATE

機能	マーカーカーソルの垂直軸値の ON/OFF を設定 / 問い合わせします。
構文	:CURSor[:TY]:MARKer:M<x>:V:STATE {<Boolean>} :CURSor[:TY]:MARKer:M<x>:V:STATE? <x> = 1 ~ 4
例	:CURSOR:TY:MARKER:M1:V:STATE ON :CURSOR:TY:MARKER:M1:V:STATE? -> :CURSOR:TY:MARKER:M1:V:STATE 1

:CURSor[:TY]:MARKer:M<x>:V:VALue?

機能	マーカーカーソルの垂直軸値を問い合わせます。
構文	:CURSor[:TY]:MARKer:M<x>:V:VALue? <x> = 1 ~ 4
例	:CURSOR:TY:MARKER:M1:V:VALUE? -> :CURSOR:TY:MARKER:M1:V:VALUE 1.5000E-03
解説	リニアスケールリングが ON のときはスケールリング値の問い合わせになります。

:CURSor[:TY]:TYPE

機能 カーソルの種類を設定 / 問い合わせします。

構文 :CURSor[:TY]:TYPE {OFF|HORIZONTAL|HAvERTICAL|MARKer|VERTical|DEGRee}

例 :CURSor[:TY]:TYPE?

:CURSOR:TY:TYPE HORIZONTAL

-> :CURSOR:TY:TYPE HORIZONTAL

:CURSor[:TY]:VERTical?

機能 Δ T 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[:TY]:VERTical?

:CURSor[:TY]:VERTical:ALL

機能 T-Y 表示の Δ T カーソルの測定値を一括 ON/OFF します。

構文 :CURSor[:TY]:VERTical:ALL {<Boolean>}

例 :CURSOR:TY:VERTICAL:ALL ON

:CURSor[:TY]:VERTical:DT?

機能 Δ T カーソル間の時間軸に関するすべての設定値を問い合わせます。

構文 :CURSor[:TY]:VERTical:DT?

:CURSor[:TY]:VERTical:DT:STATE

機能 Δ T カーソル間の時間軸値の ON/OFF を設定 / 問い合わせします。

構文 :CURSor[:TY]:VERTical:DT:STATE {<Boolean>}

例 :CURSOR:TY:VERTICAL:DT:STATE ON

:CURSOR:TY:VERTICAL:DT:STATE?

-> :CURSOR:TY:VERTICAL:DT:STATE 1

:CURSor[:TY]:VERTical:DT:VALue?

機能 Δ T カーソル間の時間軸値を問い合わせます。

構文 :CURSor[:TY]:VERTical:DT:VALue?

例 :CURSOR:TY:VERTICAL:DT:VALue?

-> :CURSOR:TY:VERTICAL:DT:VALue 2.50E-06

:CURSor[:TY]:VERTical:DV?

機能 Δ T カーソル間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 :CURSor[:TY]:VERTical:DV?

:CURSor[:TY]:VERTical:DV:STATE

機能 Δ T カーソル間の垂直軸値の ON/OFF を設定 / 問い合わせします。

構文 :CURSor[:TY]:VERTical:DV:STATE {<Boolean>}

例 :CURSOR:TY:VERTICAL:DV:STATE ON

:CURSOR:TY:VERTICAL:DV:STATE?

-> :CURSOR:TY:VERTICAL:DV:STATE 1

:CURSor[:TY]:VERTical:DV:VALue?

機能 Δ T カーソル間の垂直軸値を問い合わせます。

構文 :CURSor[:TY]:VERTical:DV:VALue? {[V12]|V21}

例 :CURSOR:TY:VERTICAL:DV:VALue?

-> :CURSOR:TY:VERTICAL:DV:VALue -1.50E+03

:CURSOR:TY:VERTICAL:DV:VALue? V21

-> :CURSOR:TY:VERTICAL:DV:VALue 1.50E+03

解説

- 選択肢「V12」「V21」の両方を省略可能です。省略したときは V1-V2 の計算結果を問い合わせます。
- 「V12」選択時は V1-V2 の計算結果を問い合わせます。
- 「V21」選択時は V2-V1 の計算結果を問い合わせます。
- リニアスケールリングが ON のときは、スケールリング値の問い合わせになります。

:CURSor[:TY]:VERTical:DVDT?

機能 Δ T カーソルの Δ V / Δ T に関するすべての設定値を問い合わせます。

例 :CURSOR:TY:VERTICAL:DVDT?

:CURSor[:TY]:VERTical:DVDT:STATE

機能 Δ T カーソルの Δ V / Δ T 値の ON/OFF を設定 / 問い合わせします。

構文 :CURSor[:TY]:VERTical:DVDT:STATE {<Boolean>}

例 :CURSOR:TY:VERTICAL:DVDT:STATE ON

:CURSOR:TY:VERTICAL:DVDT:STATE?

-> :CURSOR:TY:VERTICAL:DVDT:STATE 1

:CURSor[:TY]:VERTical:DVDT:VALue?

機能 Δ T カーソルの Δ V / Δ T 値を問い合わせます。

構文 :CURSor[:TY]:VERTical:DVDT:VALue?

例 :CURSOR:TY:VERTICAL:DVDT:VALue?

-> :CURSOR:TY:VERTICAL:DVDT:VALue 1.5E+03

解説

リニアスケールリングが ON のときは、スケールリング値の問い合わせになります。

5.10 CURSor グループ

:CURSor[:TY]:VERTical:JUMP

機能 T-Y 表示の垂直カーソルを指定したズーム波形上にジャンプさせます。

構文 :CURSor[:TY]:VERTical:JUMP {C1 _ Z1|C1 _ Z2|C2 _ Z1|C2 _ Z2}

例 :CURSOR:TY:VERTICAL:JUMP C1 _ Z1

解説 ズーム波形の中央位置にジャンプします。C1、C2 は、V カーソルの Cursor1、Cursor2 を示します。

:CURSor[:TY]:VERTical:PERDt?

機能 Δ T カーソル間の 1/ Δ T に関するすべての設定値を問い合わせます。

構文 :CURSor[:TY]:VERTical:PERDt?

:CURSor[:TY]:VERTical:PERDt:STATE

機能 Δ T カーソル間の 1/ Δ T 値の ON/OFF を設定 / 問い合わせします。

構文 :CURSor[:TY]:VERTical:PERDt:STATE {<Boolean>}

例 :CURSOR:TY:VERTICAL:PERDT:STATE?

:CURSOR:TY:VERTICAL:PERDT:STATE?
-> :CURSOR:TY:VERTICAL:PERDT:STATE 1

:CURSor[:TY]:VERTical:PERDt:VALue?

機能 Δ T カーソル間の 1/ Δ T 値を問い合わせます。

構文 :CURSor[:TY]:VERTical:PERDt:VALue?

例 :CURSOR:TY:VERTICAL:PERDT:VALUE?

-> :CURSOR:TY:VERTICAL:PERDT:
VALUE 2.50E+06

:CURSor[:TY]:VERTical:POSition<x>

機能 Δ T カーソルの位置を設定 / 問い合わせします。

構文 :CURSor[:TY]:VERTical:

POSition<x> {<NRf>}

:CURSor[:TY]:VERTical:POSition<x>?

<x> = 1 ~ 2

<NRf> = -5 ~ 5(div)、(10div/ 表示レコード長) ステップ

例 :CURSOR:TY:VERTICAL:POSITION1 2

:CURSOR:TY:VERTICAL:POSITION1?

-> :CURSOR:TY:VERTICAL:

POSITION1 2.00E+00

:CURSor[:TY]:VERTical:T<x>?

機能 Δ T カーソルの時間軸に関するすべての設定値を問い合わせます。

構文 :CURSor[:TY]:VERTical:T<x>?

<x> = 1 ~ 2

:CURSor[:TY]:VERTical:T<x>:STATE

機能 Δ T カーソルの時間軸値の ON/OFF を設定 / 問い合わせします。

構文 :CURSor[:TY]:VERTical:T<x>:

STATE {<Boolean>}

:CURSor[:TY]:VERTical:T<x>:STATE?

<x> = 1 ~ 2

例 :CURSOR:TY:VERTICAL:T1:STATE ON

:CURSOR:TY:VERTICAL:T1:STATE?

-> :CURSOR:TY:VERTICAL:T1:STATE 1

:CURSor[:TY]:VERTical:T<x>:VALue?

機能 Δ T カーソルの時間軸値を問い合わせます。

構文 :CURSor[:TY]:VERTical:T<x>:VALue?

<x> = 1 ~ 2

例 :CURSOR:TY:VERTICAL:T1:VALUE?

-> :CURSOR:TY:VERTICAL:T1:

VALUE -2.50E-06

:CURSor[:TY]:VERTical:TRACe

機能 Δ T カーソルの対象波形を設定 / 問い合わせします。

構文 :CURSor[:TY]:VERTical:TRACe {<NRf>|

ALL|MATH<x>}

:CURSor[:TY]:VERTical:TRACe?

<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)

<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :CURSOR:TY:VERTICAL:TRACE 1

:CURSOR:TY:VERTICAL:TRACE?

-> :CURSOR:TY:VERTICAL:TRACE 1

:CURSor[:TY]:VERTical:V<x>?

機能 Δ T カーソル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 :CURSor[:TY]:VERTical:V<x>?

<x> = 1 ~ 2

:CURSor[:TY]:VERTical:V<x>:STATE

機能 Δ T カーソルの垂直軸値の ON/OFF を設定 / 問い合わせします。

構文 :CURSor[:TY]:VERTical:V<x>:

STATE {<Boolean>}

:CURSor[:TY]:VERTical:V<x>:STATE?

<x> = 1 ~ 2

例 :CURSOR:TY:VERTICAL:V1:STATE ON

:CURSOR:TY:VERTICAL:V1:STATE?

-> :CURSOR:TY:VERTICAL:V1:STATE 1

:CURSor[:TY]:VERTical:V<x>:VALue?

機能 Δ T カーソルの垂直軸値を問い合わせます。

構文 :CURSor[:TY]:VERTical:V<x>:VALue?

<x> = 1 ~ 2

例 :CURSOR:TY:VERTICAL:V1:VALUE?

-> :CURSOR:TY:VERTICAL:V1:

VALUE 2.50E+03

5.11 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?

:DISPlay:ACCumulate?

機能 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?

:DISPlay:ACCumulate:MODE

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:MODE {COLor|OFF|PERSistence}
:DISPlay:ACCumulate:MODE?
例 :DISPlay:ACCumulate:MODE COLor
:DISPlay:ACCumulate:MODE?
-> :DISPlay:ACCumulate:MODE COLor

:DISPlay:ACCumulate:PERSistence

機能 アキュムレート時間を設定 / 問い合わせします。
構文 :DISPlay:ACCumulate:
PERSistence {<時間>|INFINITY}
:DISPlay:ACCumulate:PERSistence?
<時間> = 100ms ~ 100s
例 :DISPlay:ACCumulate:
PERSistence 100ms
:DISPlay:ACCumulate:PERSistence?
-> :DISPlay:ACCumulate:
PERSistence 0.1000

: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?

:DISPlay:COLor: {CHANnel<x>|MATH<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>|
MATH<x>} {BLUE|BGReen|CYAN|DBLue|
GRAY|GReen|LBLue|LGReen|MAGenta|
MGReen|ORANge|PINK|PURPle|RED|SPINK|
YELLow}
:DISPlay:COLor:{CHANnel<x>|MATH<x>}?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例 :DISPlay:COLor:CHANnel1 BLUE
:DISPlay:COLor:CHANnel1?
-> :DISPlay:COLor:CHANnel1 BLUE

:DISPlay:COLor: {LOGic|LState}

機能 ロジック波形の色とステート表示の色を設定 / 問い合わせします。
構文 :DISPlay:COLor:{LOGic|LState} {BLUE|
BGReen|CYAN|DBLue|GRAY|GReen|LBLue|
LGReen|MAGenta|MGReen|ORANge|PINK|
PURPle|RED|SPINK|YELLow}
:DISPlay:COLor:{LOGic|LState}?
例 :DISPlay:COLor:LState BLUE
:DISPlay:COLor:LState?
-> :DISPlay:COLor:LState BLUE

:DISPlay:COLor:SERialbus<x>?

機能 シリアルバスごとの全トレンドの色設定を問い合わせます。
構文 :DISPlay:COLor:SERialbus<x>?
<x> = 1 ~ 4
解説トレンド表示をサポートしたシリアルバスオプション付きのモデルで使用できます。

:DISPlay:COLor:SErIalbus<x>:TREND<y>
機能 シリアルバストレンドの色を設定 / 問い合わせします。

構文 :DISPlay:COLor:SErIalbus<x>:
TREND<y> {BLUE|BGReen|CYAN|DBLue|
GRAY|GReeN|LBLue|LGReeN|MAGenta|
MGReen|ORANge|PINK|PURPle|RED|SPINK|
YELLow}
:DISPlay:COLor:SErIalbus<x>:
TREND<y>?
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :DISPLAY:COLOR:SERIALBUS1:
TREND1 BLUE
:DISPLAY:COLOR:SERIALBUS1:TREND1?
-> :DISPLAY:COLOR:SERIALBUS1:
TREND1 BLUE

解説 トレンド表示をサポートしたシリアルバスオブ
ション付きのモデルで使用できます。

: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 {AUTO|SINGLE|DUAL|
TRIad|QUAD|HEXa|OCTal}
:DISPlay:FORMat?

例 :DISPLAY:FORMAT AUTO
:DISPLAY:FORMAT?
-> :DISPLAY:FORMAT AUTO

:DISPlay:FGRId

機能 ファイングリッド表示の ON/OFF を設定 / 問い合
わせします。

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

例 :DISPLAY:FGRID ON
:DISPLAY:FGRID?
-> :DISPLAY:FGRID 1

: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[: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 ~ 64

例 :DISPLAY:INTENSITY:WAVEFORM 10
:DISPLAY:INTENSITY:WAVEFORM?
-> :DISPLAY:INTENSITY:WAVEFORM 10

: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 以下は、カーソルについての例です。
:DISPLAY:INTENSITY:Cursors 10
:DISPLAY:INTENSITY:Cursors?
-> :DISPLAY:INTENSITY:Cursors 10

:DISPlay:INTERpolate

機能 補間方式を設定 / 問い合わせします。

構文 :DISPlay:INTERpolate {OFF|LINE|
PULSe|SINE}
:DISPlay:INTERpolate?

例 :DISPLAY:INTERPOLATE OFF
:DISPLAY:INTERPOLATE?
-> :DISPLAY:INTERPOLATE OFF

:DISPlay:MAPPING

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 {AUTO|MANual}
:DISPlay:MAPPING?

例 :DISPLAY:MAPPING AUTO
:DISPLAY:MAPPING?
-> :DISPLAY:MAPPING AUTO

:DISPlay:SMAPPING?

機能 分割フォーマットへの全波形の割り当てを問い
合わせます。

構文 :DISPlay:SMAPPING?

5.11 DISPlay グループ

:DISPlay:SMAPping:{CHANnel<x>|MATH<x>} (Set Mapping)

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:SMAPping:CHANnel<x> {<NRf>}
:DISPlay:SMAPping:CHANnel<x>?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = 1 ~ 8

例 :DISPLAY:SMAPPING:CHANNEL1 1
:DISPLAY:SMAPPING:CHANNEL1?
-> :DISPLAY:SMAPPING:CHANNEL1 1

:DISPlay: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 :DISPlay:SVALue {<Boolean>}
:DISPlay:SVALue?

例 :DISPLAY:SVALUE ON
:DISPLAY:SVALUE?
-> :DISPLAY:SVALUE 1

5.12 FFT グループ

:FFT<x>?

機能 FFT 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 :FFT<x>?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:FFT<x>:AVERAge?

機能 FFT 解析のアベレ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 :FFT<x>:AVERAge?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:FFT<x>:AVERAge:EWEight

機能 FFT 解析の指数化平均の減衰定数を設定 / 問い合わせします。

構文 :FFT<x>:AVERAge:EWEight {<NRf>}
:FFT<x>:AVERAge:EWEight?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<NRf> = 2 ~ 1024、2n ステップ

例 :FFT1:AVERAGE:EWEIGHT 2
:FFT1:AVERAGE:EWEIGHT?
-> :FFT1:AVERAGE:EWEIGHT 2

:FFT<x>:DATA?

機能 送信する FFT 波形データのすべての情報を問い合わせます。

構文 :FFT<x>:DATA?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:FFT<x>:DATA:BYTeorder

機能 FFT 波形データのフォーマットがバイナリの際の送信順序を設定 / 問い合わせします。

構文 :FFT<x>:DATA:BYTeorder {LSBFirst|MSBFirst}
:FFT<x>:DATA:BYTeorder?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:DATA:BYTEORDER LSBFIRST
:FFT1:DATA:BYTEORDER?
-> :FFT1:DATA:BYTEORDER LSBFIRST

:FFT<x>:DATA:END

機能 送信する FFT 波形データの終了点を設定 / 問い合わせします。

構文 :FFT<x>:DATA:END {<NRf>}
:FFT<x>:DATA:END?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<NRf> = 0 ~ 1250000

例 :FFT1:DATA:END 125000
:FFT1:DATA:END?
-> :FFT1:DATA:END 125000

:FFT<x>:DATA:FORMat

機能 送信する FFT 波形データのフォーマットを設定 / 問い合わせします。

構文 :FFT<x>:DATA:FORMat {ASCIi|BINary}
:FFT<x>:DATA:FORMat?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:DATA:FORMAT ASCII
:FFT1:DATA:FORMAT?
-> :FFT:DATA:FORMAT ASCII

:FFT<x>:DATA:LENGth?

機能 送信する FFT 波形の全データ点数を問い合わせます。

構文 :FFT<x>:DATA:LENGth?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:DATA:LENGTH?
-> :FFT1:DATA:LENGTH 6251

:FFT<x>:DATA:SEND?

機能 FFT 波形データを問い合わせます。
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

構文 :FFT<x>:DATA:SEND?
例 :FFT1:DATA:SEND?
-> :FFT1:DATA:SEND #6

(6 桁のバイト数) (データ列) または、
<NRf>,<NRf>,...

解説 「:FFT<x>:DATA:FORMat」の設定によって、「:FFT<x>:DATA:SEND?」の出力形式が変わります。

- (1) 「ASCIi」にしたとき <NRf>、<NRf>、...<NRf> の形式で返されます。
- (2) 「BINary」にしたとき <ブロックデータ> の形式で返されます。1 点のデータは 4 バイトであり、<ブロックデータ> は FLOAT 型変換で換算できます。

5.12 FFT グループ

:FFT<x>:DATA:START

機能 送信する FFT 波形データの開始点を設定 / 問い合わせします。

構文 :FFT<x>:DATA:START {<NRf>}
:FFT<x>:DATA:START?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<NRf> = 0 ~ 125000

例 :FFT1:DATA:START 0
:FFT1:DATA:START?
-> :FFT1:DATA:START 0

:FFT<x>:DISPLAY

機能 FFT 解析をする (ON)/ しない (OFF) を設定 / 問い合わせします。

構文 :FFT<x>:DISPLAY {<Boolean>}
:FFT<x>:DISPLAY?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:DISPLAY ON
:FFT1:DISPLAY?
-> :FFT1:DISPLAY 1

:FFT<x>:HORIZONTAL?

機能 FFT 解析の横軸に関するすべての設定値を問い合わせます。

構文 :FFT<x>:HORIZONTAL?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:FFT<x>:HORIZONTAL:CSpan?

機能 FFT 解析の横軸の中心点 / スパンに関するすべての設定値を問い合わせます。

構文 :FFT<x>:HORIZONTAL:CSpan?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:FFT<x>:HORIZONTAL:CSpan:Center

機能 FFT 解析の横軸の中心値を設定 / 問い合わせします。

構文 :FFT<x>:HORIZONTAL:CSpan:
CENTER {<周波数>}
:FFT<x>:HORIZONTAL:CSpan:Center?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<周波数> = 0 ~ 125G(Hz)

例 :FFT1:HORIZONTAL:CSpan:Center 1HZ
:FFT1:HORIZONTAL:CSpan:Center?
-> :FFT1:HORIZONTAL:CSpan:
CENTER 1.000E+00

:FFT<x>:HORIZONTAL:CSpan:SPAN

機能 FFT 解析の横軸のスパンを設定 / 問い合わせします。

構文 :FFT<x>:HORIZONTAL:CSpan:
SPAN {<周波数>}
:FFT<x>:HORIZONTAL:CSpan:SPAN?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<周波数> = 0 ~ 125G(Hz)

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

:FFT<x>:HORIZONTAL:LRIGHT?

機能 FFT 解析の横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 :FFT<x>:HORIZONTAL:LRIGHT?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:FFT<x>:HORIZONTAL:LRIGHT:Range

機能 FFT 解析の横軸の左右端の範囲を設定 / 問い合わせします。

構文 :FFT<x>:HORIZONTAL:LRIGHT:
Range {<周波数>,<周波数>}
:FFT<x>:HORIZONTAL:LRIGHT:Range?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<周波数> = 0 ~ 125G(Hz)

例 :FFT1:HORIZONTAL:LRIGHT:Range 1HZ,
2HZ
:FFT1:HORIZONTAL:LRIGHT:Range?
-> :FFT1:HORIZONTAL:LRIGHT:
Range 2.000E+00,1.000E+00

:FFT<x>:HORIZONTAL:MODE

機能 FFT 解析の横軸のモードを設定 / 問い合わせします。

構文 :FFT<x>:HORIZONTAL:MODE {AUTO|CSpan|
LRIGHT}
:FFT<x>:HORIZONTAL:MODE?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:HORIZONTAL:MODE AUTO
:FFT1:HORIZONTAL:MODE?
-> :FFT1:HORIZONTAL:MODE AUTO

:FFT<x>:LENGTH

機能 FFT 解析の FFT 点数を設定 / 問い合わせします。

構文 :FFT<x>:LENGTH {<NRf>}
:FFT<x>:LENGTH?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<NRf> = 1250、2500、12500、25000、125000、
250000、1250000

例 :FFT1:LENGTH 1250
:FFT1:LENGTH?
-> :FFT1:LENGTH 1250

:FFT<x>:MEASure?

機能 FFT 解析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 :FFT<x>:MEASure?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:FFT<x>:MEASure:MARKer?

機能 FFT 解析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 :FFT<x>:MEASure:MARKer?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:FFT<x>:MEASure:MARKer[:BASic]?

機能 FFT 解析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 :FFT<x>:MEASure:MARKer:BASic?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:FFT<x>:MEASure:MARKer[:BASic]:ALL

機能 FFT 解析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 :FFT<x>:MEASure:MARKer[:BASic]:
ALL {<Boolean>}
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:MEASURE:MARKER:BASIC:ALL ON

:FFT<x>:MEASure:MARKer[:BASic]:DFRequency?

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

構文 :FFT<x>:MEASure:MARKer[:BASic]:
DFRequency?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:FFT<x>:MEASure:MARKer[:BASic]:DFRequency:STATE

機能 FFT 解析のマーカーカーソル間の周波数の ON/OFF を設定 / 問い合わせします。

構文 :FFT<x>:MEASure:MARKer[:BASic]:
DFRequency:STATE {<Boolean>}
:FFT<x>:MEASure:MARKer[:BASic]:
DFRequency:STATE?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

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

:FFT<x>:MEASure:MARKer[:BASic]:DFRequency:VALue?

機能 FFT 解析のマーカーカーソル間の周波数を問い合わせます。

構文 :FFT<x>:MEASure:MARKer[:BASic]:
DFRequency:VALue?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

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

:FFT<x>:MEASure:MARKer[:BASic]:DV?

機能 FFT 解析のマーカーカーソル間のレベルに関するすべての設定値を問い合わせます。

構文 :FFT<x>:MEASure:MARKer[:BASic]:DV?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:FFT<x>:MEASure:MARKer[:BASic]:DV:STAtE

機能 FFT 解析のマーカーカーソル間のレベルの ON/OFF を設定 / 問い合わせします。

構文 :FFT<x>:MEASure:MARKer[:BASic]:DV:
STAtE {<Boolean>}
:FFT<x>:MEASure:MARKer[:BASic]:DV:
STAtE?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
例 :FFT1:MEASURE:MARKER:BASIC:DV:
STATE ON
:FFT1:MEASURE:MARKER:BASIC:DV:STATE?
-> :FFT1:MEASURE:MARKER:BASIC:DV:
STATE 1

:FFT<x>:MEASure:MARKer[:BASic]:DV:VALue?

機能 FFT 解析のマーカーカーソル間のレベルを問い合わせます。

構文 :FFT<x>:MEASure:MARKer[:BASic]:DV:
VALue?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

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

:FFT<x>:MEASure:MARKer[:BASic]:FREQuency<y>?

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

構文 :FFT<x>:MEASure:MARKer[:BASic]:
FREQuency<y>?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<y> = 1 ~ 2

5.12 FFT グループ

:FFT<x>:MEASure:MARKer[:BASic]:FREQuency<y>:STATE

機能 FFT 解析の各マーカーカーソルの周波数の ON/OFF を設定 / 問い合わせします。

構文 :FFT<x>:MEASure:MARKer[:BASic]:FREQuency<y>:STATE {<Boolean>}
:FFT<x>:MEASure:MARKer[:BASic]:FREQuency<y>:STATE?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<y> = 1 ~ 2

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

:FFT<x>:MEASure:MARKer[:BASic]:FREQuency<y>:VALue?

機能 FFT 解析の各マーカーカーソルの周波数を問い合わせます。

構文 :FFT<x>:MEASure:MARKer[:BASic]:FREQuency<y>:VALue?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<y> = 1 ~ 2

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

:FFT<x>:MEASure:MARKer[:BASic]:POSition<y>

機能 FFT 解析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 :FFT<x>:MEASure:MARKer[:BASic]:POSITION<y> {<NRf>}
:FFT<x>:MEASure:MARKer[:BASic]:POSITION<y>?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<y> = 1 ~ 2
<NRf> = -5 ~ 5(div)

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

:FFT<x>:MEASure:MARKer[:BASic]:V<y>?

機能 FFT 解析の各マーカーカーソルのレベルに関するすべての設定値を問い合わせます。

構文 :FFT<x>:MEASure:MARKer[:BASic]:V<y>?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<y> = 1 ~ 2

:FFT<x>:MEASure:MARKer[:BASic]:V<y>:STATE

機能 FFT 解析の各マーカーカーソルのレベルの ON/OFF を設定 / 問い合わせします。

構文 :FFT<x>:MEASure:MARKer[:BASic]:V<y>:STATE {<Boolean>}
:FFT<x>:MEASure:MARKer[:BASic]:V<y>:STATE?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<y> = 1 ~ 2

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

:FFT<x>:MEASure:MARKer[:BASic]:V<y>:VALue?

機能 FFT 解析の各マーカーカーソルのレベルを問い合わせます。

構文 :FFT<x>:MEASure:MARKer[:BASic]:V<y>:VALue?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<y> = 1 ~ 2

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

:FFT<x>:MEASure:MODE

機能 FFT 解析の自動測定モードを設定 / 問い合わせします。

構文 :FFT<x>:MEASure:MODE {MARKer|OFF|PARAMeter|PEAK}
:FFT<x>:MEASure:MODE?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

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

:FFT<x>:MEASure:PARAmeter?

機能 FFT 解析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 :FFT<x>:MEASure:PARAmeter?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:MEASURE:PARAMETER?

:FFT<x>:MEASure:PARAmeter:OVERall?

機能 FFT 解析のパラメータのオーバーオールに関するすべての設定値を問い合わせます。

構文 :FFT<x>:MEASure:PARAmeter:OVERall?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:MEASURE:PARAMETER:OVERALL?

:FFT<x>:MEASure:PARAMeter:OVERall:STAtE

機能 FFT 解析のパラメータのオーバーオール値の ON/OFF を設定 / 問い合わせします。

構文 :FFT<x>:MEASure:PARAMeter:OVERall:STATE {<Boolean>}

:FFT<x>:MEASure:PARAMeter:OVERall:STATE?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:MEASURE:PARAMETER:OVERALL:STATE ON

:FFT1:MEASURE:PARAMETER:OVERALL:STATE?

->:FFT1:MEASURE:PARAMETER:OVERALL:STATE 1

:FFT<x>:MEASure:PARAMeter:OVERall:VALue?

機能 FFT 解析のパラメータのオーバーオール値を問い合わせます。

構文 :FFT<x>:MEASure:PARAMeter:OVERall:VALue?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:MEASURE:PARAMETER:OVERALL:VALUE?

-> :FFT1:MEASURE:PARAMETER:OVERALL:VALUE 2.27806E+00

:FFT<x>:MEASure:PEAK?

機能 FFT 解析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 :FFT<x>:MEASure:PEAK?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:FFT<x>:MEASure:PEAK:DETail?

機能 FFT 解析結果リストに関するすべての設定値を問い合わせます。

構文 :FFT<x>:MEASure:PEAK:DETail?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:FFT<x>:MEASure:PEAK:DETail:LIST:ITEM?

機能 FFT 解析結果リストに表示される項目を問い合わせます。

構文 :FFT<x>:MEASure:PEAK:DETail:LIST:ITEM?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:MEASURE:PEAK:DETAIL:LIST:ITEM?

-> :FFT1:MEASURE:PEAK:DETAIL:LIST:ITEM? "No, Frequency, Peak"

:FFT<x>:MEASure:PEAK:DETail:LIST:MAXPeak:EXECute

機能 FFT 解析結果の最大ピーク値に移動します。

構文 :FFT<x>:MEASure:PEAK:DETail:LIST:MAXPeak:EXECute

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:MEASURE:PEAK:DETAIL:LIST:MAXPEAK:EXECUTE

:FFT<x>:MEASure:PEAK:DETail:LIST:NUMBER

機能 FFT 解析結果リストの解析番号を設定 / 問い合わせします。

構文 :FFT<x>:MEASure:PEAK:DETail:LIST:NUMBER {<Nrf>}

:FFT<x>:MEASure:PEAK:DETail:LIST:NUMBER?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

<Nrf> = 1 ~ 10

例 :FFT1:MEASURE:PEAK:DETAIL:LIST:NUMBER 1

:FFT1:MEASURE:PEAK:DETAIL:LIST:NUMBER?

-> :FFT1:MEASURE:PEAK:DETAIL:LIST:NUMBER 1

:FFT<x>:MEASure:PEAK:DETail:LIST:VALue?

機能 FFT 解析結果リストの指定した解析番号の全データを問い合わせます。

構文 :FFT<x>:MEASure:PEAK:DETail:LIST:VALue? {<Nrf>}

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

<Nrf> = 1 ~ 10

例 :FFT1:MEASURE:PEAK:DETAIL:LIST:VALUE? 1

-> :FFT1:MEASURE:PEAK:DETAIL:LIST:VALUE "1, 0.000000 Hz, -10.00000 dBV"

:FFT<x>:MEASure:PEAK:EXCursion

機能 FFT 解析のピーク値の山谷差を設定 / 問い合わせします。

構文 :FFT<x>:MEASure:PEAK:EXCursion {<Nrf>}

:FFT<x>:MEASure:PEAK:EXCursion?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

<Nrf> = 0 ~ 200

例 :FFT1:MEASURE:PEAK:EXCURSION 1

:FFT1:MEASURE:PEAK:EXCURSION?

-> :FFT1:MEASURE:PEAK:EXCURSION 1.000E+00

5.12 FFT グループ

:FFT<x>:MEASure:PEAK:FVALue?

機能 FFT 解析のピーク周波数を問い合わせます。
構文 :FFT<x>:MEASure:PEAK:FVALue?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
例 :FFT1:MEASure:PEAK:FVALue?
-> :FFT1:MEASure:PEAK:
FVALUE 1.000E+00

:FFT<x>:MEASure:PEAK:RANGe

機能 FFT 解析のピーク値の測定範囲を設定 / 問い合わせします。
構文 :FFT<x>:MEASure:PEAK:RANGe {<NRf>,<NRf>}
:FFT<x>:MEASure:PEAK:RANGe?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<NRf> = -5 ~ 5(div), (10 div/ 表示レコード長) ステップ
例 :FFT1:MEASure:PEAK:RANGe 0,1
:FFT1:MEASure:PEAK:RANGe?
-> :FFT1:MEASure:PEAK:
RANGe 0.000E+00,1.000E+00

:FFT<x>:MEASure:PEAK:THReshold

機能 FFT 解析のピーク値のしきい値を設定 / 問い合わせします。
構文 :FFT<x>:MEASure:PEAK:THReshold {<NRf>}
:FFT<x>:MEASure:PEAK:THReshold?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<NRf> = -1.0000E+31 ~ 1.0000E+31
例 :FFT1:MEASure:PEAK:THRESHOLD 1
:FFT1:MEASure:PEAK:THRESHOLD?
-> :FFT1:MEASure:PEAK:
THRESHOLD 1.000E+00

:FFT<x>:MEASure:PEAK:VVALue?

機能 FFT 解析のピーク値を問い合わせます。
構文 :FFT<x>:MEASure:PEAK:VVALue?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
例 :FFT1:MEASure:PEAK:VVALue?
-> :FFT1:MEASure:PEAK:
VVALUE 1.000E+00

:FFT<x>:MODE

機能 FFT 解析の波形の表示方式を設定 / 問い合わせします。
構文 :FFT<x>:MODE {AVERAge|MAXHold|NORMal}
:FFT<x>:MODE?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
例 :FFT1:MODE AVERAGE
:FFT1:MODE?
-> :FFT1:MODE AVERAGE

:FFT<x>:RANGe

機能 FFT 解析の測定対象ウィンドウを設定 / 問い合わせします。
構文 :FFT<x>:RANGe {MAIN|Z1|Z2}
:FFT<x>:RANGe?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
例 :FFT1:RANGe MAIN
:FFT1:RANGe?
-> :FFT1:RANGe MAIN

:FFT<x>:RPOSition

機能 FFT 解析の縦軸の拡大中心点を設定 / 問い合わせします。
構文 :FFT<x>:RPOSition {<NRf>}
:FFT<x>:RPOSition?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<NRf> = -4 ~ 4(div)
例 :FFT1:RPOSITION 1
:FFT1:RPOSITION?
-> :FFT1:RPOSITION 1.000E+00

:FFT<x>:TRACe

機能 FFT 解析の対象波形を設定 / 問い合わせします。
構文 :FFT<x>:TRACe {<NRf>|MATH<y>}
:FFT<x>:TRACe?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例 :FFT1:TRACE 1
:FFT1:TRACE?
-> :FFT1:TRACE 1

:FFT<x>:TYPE

機能 FFT 解析のスペクトラムを設定 / 問い合わせします。

構文 :FFT<x>:TYPE {CH _MAG|CS _IMAG|CS _LOGMAG|CS _MAG|CS _PHASE|CS _REAL|LS _IMAG|LS _LOGMAG|LS _MAG|LS _PHASE|LS _REAL|PS _LOGMAG|PS _MAG|PSD _LOGMAG|PSD _MAG|RS _LOGMAG|RS _MAG|TF _IMAG|TF _LOGMAG|TF _MAG|TF _PHASE|TF _REAL[,<NRf>|MATH<y>]}

:FFT<x>:TYPE?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)

MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :FFT1:TYPE CH _MAG,1

:FFT1:TYPE?

-> :FFT1:TYPE CH _MAG,1

解説

- ・ ユーザー定義演算 (オプション) に対応してないと、エラーになります。
- ・ {CH _MAG|CS _IMAG|CS _LOGMAG|CS _MAG|CS _PHASE|CS _REAL|TF _IMAG|TF _LOGMAG|TF _MAG|TF _PHASE|TF _REAL} の場合は、<NRf> に 2 項目の対象波形を選択します。

:FFT<x>:UNIT?

機能 FFT 演算単位に関するすべての設定値を問い合わせます。

構文 :FFT<x>:UNIT?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:FFT<x>:UNIT:DBM?

機能 FFT 演算単位が dBm のときに関するすべての設定値を問い合わせます。

構文 :FFT<x>:UNIT:DBM ?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:UNIT:DBM?

:FFT<x>:UNIT:DBM:IMPedance

機能 FFT 演算単位が dBm のときのインピーダンスを設定 / 問い合わせします。

構文 :FFT<x>:UNIT:DBM:IMPedance {<NRf>}

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

<NRf> = 40 ~ 20000

例 :FFT1:UNIT:DBM:IMPEDANCE 50

:FFT1:UNIT:DBM:IMPEDANCE?

-> :FFT1:UNIT:DBM:IMPEDANCE 50

:FFT<x>:UNIT:MODE

機能 FFT 演算単位の自動 / 手動付加を設定 / 問い合わせします。

構文 :FFT<x>:UNIT:MODE {AUTO|DBM}

:FFT<x>:UNIT:MODE?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:UNIT:MODE AUTO

:FFT1:UNIT:MODE?

-> :FFT1:UNIT:MODE AUTO

:FFT<x>:VERTical?

機能 FFT 解析の縦軸に関するすべての設定値を問い合わせます。

構文 :FFT<x>:VERTical?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:FFT<x>:VERTical:LEVel

機能 FFT 解析の縦軸の表示位置を設定 / 問い合わせします。

構文 :FFT<x>:VERTical:LEVel {<NRf>}

:FFT<x>:VERTical:LEVel?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

<NRf> = -1.0000E+31 ~ 1.0000E+31

例 :FFT1:VERTICAL:LEVEL 1

:FFT1:VERTICAL:LEVEL?

-> :FFT1:VERTICAL:LEVEL 1.000E+00

:FFT<x>:VERTical:MODE

機能 FFT 解析の縦軸のモードを設定 / 問い合わせします。

構文 :FFT<x>:VERTical:MODE {AUTO|MANual}

:FFT<x>:VERTical:MODE?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:VERTICAL:MODE AUTO

:FFT1:VERTICAL:MODE?

-> :FFT1:VERTICAL:MODE AUTO

:FFT<x>:VERTical:SENSitivity

機能 FFT 解析の縦軸感度を設定 / 問い合わせします。

構文 :FFT<x>:VERTical:SENSitivity {<NRf>}

:FFT<x>:VERTical:SENSitivity?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

<NRf> = -1.0000E+31 ~ 1.0000E+31

例 :FFT1:VERTICAL:SENSITIVITY 1

:FFT1:VERTICAL:SENSITIVITY?

-> :FFT1:VERTICAL:SENSITIVITY 1.000E+00

5.12 FFT グループ

:FFT<x>:VTDisplay

機能 FFT 解析の VT 波形の表示の ON/OFF を設定 / 問い合わせします。

構文 :FFT<x>:VTDisplay {<Boolean>}
:FFT<x>:VTDisplay?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :FFT1:VTDISPLAY ON
:FFT1:VTDISPLAY?
-> :FFT1:VTDISPLAY 1

:FFT<x>:WINDOW

機能 FFT 解析の窓関数を設定 / 問い合わせします。

構文 :FFT<x>:WINDOW {FLATtop|HANNing|
RECTangle}
:FFT<x>:WINDOW?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

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

5.13 FILE グループ

:FILE?

機能 データストレージに関するすべての設定値を問い合わせます。

構文 :FILE?

:FILE:COPY:ABORT

機能 ファイルのコピーを中止します。

構文 :FILE:COPY:ABORT

例 :FILE:COPY:ABORT

:FILE:COPY:CDIRECTORY

機能 ファイルのコピー先のディレクトリを変更します。

構文 :FILE:COPY:CDIRECTORY {<文字列>}
<文字列> = 本体ユーザーズマニュアル機能編参照。

例 :FILE:COPY:CDIRECTORY "UTIL"
(相対パス指定)

:FILE:COPY:DRIVE

機能 ファイルのコピー先のメディアを設定します。

構文 :FILE:COPY:DRIVE {FLASHmem|NETWork|
USB,<Nrf>}
<Nrf> = 0 ~ 1

例 :FILE:COPY:DRIVE FLASHMEM

解説 「USB」の<Nrf>は、パーティションまたはLUNで区切られていない場合は省略できます。

:FILE:COPY[:EXECute]

機能 ファイルのコピーを実行します。オーバーラップコマンドです。

構文 :FILE:COPY[:EXECute] {<文字列>}
<文字列> = 本体ユーザーズマニュアル機能編参照。

例 :FILE:COPY:EXECUTE "DATA.PNG"

:FILE:COPY:PATH?

機能 ファイルのコピー先のディレクトリを問い合わせます。

構文 :FILE:COPY:PATH?

例 :FILE:COPY:PATH?

-> :FILE:COPY:
PATH "PATH = FLASHMEM/UTIL"

:FILE:DELeTe: {AHIStoGram|ASCii|BINary|BMP|FFT|HLISt|JPEG|MEASure|PNG|SBUS|SETup|SNAP|ZPOLygon|ZWAVe}[:EXECute]

機能 各種データのファイルの削除を実行します。オーバーラップコマンドです。

構文 :FILE:DELeTe:{AHIStoGram|ASCii|
BINary|BMP|FFT|HLISt|JPEG|MEASure|
PNG|SBUS|SETup|SNAP|ZPOLygon|
ZWAVe}[:EXECute] {<Filename>}

例 以下は、波形データについての例です。

:FILE:DELeTe:AHISTOGRAM:
EXECUTE "DATA"

解説 削除対象メディアは、「:FILE:DIRectory:DRive」で選択します。

:FILE[:DIRectory]:CDIRECTORY

機能 カレントのディレクトリを変更します。

構文 :FILE[:DIRectory]:CDIRECTORY {<文字列>}
<文字列> = 本体ユーザーズマニュアル機能編参照。

例 :FILE:DIRECTORY:CDIRECTORY "UTIL"

:FILE[:DIRectory]:DRIVE

機能 カレントのメディアを設定します。

構文 :FILE[:DIRectory]:DRIVE {FLASHmem|
NETWork|USB,<Nrf>}
<Nrf> = 0 ~ 1

例 :FILE:DIRECTORY:DRIVE FLASHMEM

解説 「USB」の<Nrf>は、パーティションまたはLUNで区切られていない場合は省略できます。

:FILE[:DIRectory]:FREE?

機能 カレントの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

:FILE[:DIRectory]:MDIRECTORY

機能 カレント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> = 本体ユーザーズマニュアル機能編参照。

例 :FILE:DIRECTORY:MDIRECTORY "DIR2"

5.13 FILE グループ

:FILE[:DIRectory]:PATH?

機能 カレントのディレクトリを問い合わせます。
構文 :FILE[:DIRectory]:PATH?
例 :FILE:DIRECTORY:PATH?
-> :FILE:DIRECTORY:
PATH "PATH = FLASHMEM/UTIL"

:FILE:LOAD:BINary:ABORt

機能 波形データの呼び出しを中止します。
構文 :FILE:LOAD:BINary:ABORt
例 :FILE:LOAD:BINARY:ABORT

:FILE:LOAD:BINary[:EXECute]

機能 波形データの呼び出しを実行します。オーバーラップコマンドです。
構文 :FILE:LOAD:BINary[:EXECute] {<Filename>{,ACQMemory|REfERENCE<x>}}
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例 :FILE:LOAD:BINARY:EXECUTE "DATA"

:FILE:LOAD:{SETup|SNAP|ZPOLygon<x>|ZWAVE<x>}:ABORt

機能 各種データの呼び出しを中止します。
構文 :FILE:LOAD:{SETup|SNAP|ZPOLygon<x>|ZWAVE<x>}:ABORt
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例 以下は、設定データについての例です。
:FILE:LOAD:SETUP:ABORT

:FILE:LOAD:{SETup|SNAP|ZPOLygon<x>|ZWAVE<x>}[:EXECute]

機能 各種データの呼び出しを実行します。オーバーラップコマンドです。
構文 :FILE:LOAD:{SETup|SNAP|ZPOLygon<x>|ZWAVE<x>}[:EXECute] {<Filename>}
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例 以下は、設定データについての例です。
:FILE:LOAD:SETUP:EXECUTE "DATA"

:FILE:MOVE:ABORt

機能 ファイルの移動を中止します。
構文 :FILE:MOVE:ABORt
例 :FILE:MOVE:ABORT

:FILE:MOVE:CDIRectory

機能 ファイルの移動先のディレクトリを変更します。
構文 :FILE:MOVE:CDIRectory {<文字列>}
<文字列> = 本体ユーザーズマニュアル機能編参照。
例 :FILE:MOVE:CDIRECTORY "UTIL"

:FILE:MOVE:DRive

機能 ファイルの移動先のメディアを設定します。
構文 :FILE:MOVE:DRive {FLASHmem|USB,<Nrf>}
<Nrf> = 0 ~ 1
例 :FILE:MOVE:DRIVE FLASHMEM
解説 「USB」の<Nrf>は、パーティションまたはLUNで区切られていない場合は省略できます。

:FILE:MOVE[:EXECute]

機能 ファイルの移動を実行します。オーバーラップコマンドです。
構文 :FILE:MOVE[:EXECute] {<文字列>}
<文字列> = 本体ユーザーズマニュアル機能編参照。
例 :FILE:MOVE:EXECUTE "DATA.PNG"

:FILE:MOVE:PATH?

機能 ファイルの移動先のディレクトリを問い合わせます。
構文 :FILE:MOVE:PATH?
例 :FILE:MOVE:PATH?
-> :FILE:MOVE:
PATH "PATH = FLASHMEM/UTIL"

:FILE:PROTeCt[:EXECute]

機能 ファイルのプロテクトのON/OFFを設定します。
構文 :FILE:PROTeCt[:EXECute] {<文字列>,<Boolean>}
<文字列> = 本体ユーザーズマニュアル機能編参照。
例 :FILE:PROTECT:EXECUTE "DATA.PNG",ON

:FILE:REName[:EXECute]

機能 ファイルのリネームを設定します。
構文 :FILE:REName[:EXECute] {<文字列>,<文字列>}
<文字列> = 本体ユーザーズマニュアル機能編参照。
例 :FILE:RENAME:EXECUTE "DATA.PNG",
"000.PNG"

:FILE:SAVE?

機能 ファイル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?

:FILE:SAVE:{AHIStogram|ASCIi|BINary|FFT|HLISt|HARMonics|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|FFT|HLISt|HARMonics|SBUS|ZWAVE}?

:FILE:SAVE:{AHISTogram|ASCIi|BINary|FFT|HLIST|HARMonics|MEASure|SBUS|SETup|SNAP|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|FFT|HLIST|HARMonics|MEASure|SBUS|SETup|SNAP|ZWAVE}:ABORT

例 以下は、波形データについての例です。
:FILE:SAVE:BINARY:ABORT

:FILE:SAVE:{AHISTogram|ASCIi|BINary|FFT|HLIST|HARMonics|MEASure|SBUS|SETup|SNAP|ZWAVE}[:EXECute]

機能 各種データのファイルへ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:SAVE:{AHISTogram|ASCIi|BINary|FFT|HARMonics|MEASure|SBUS|SETup|SNAP|ZWAVE}[:EXECute] {<Filename>}

例 以下は、波形データについての例です。
:FILE:SAVE:BINARY:EXECUTE "DATA"

:FILE:SAVE:{AHISTogram|FFT|HARMonics|SBUS|ZWAVE}:SELEct

機能 各種データの保存するエリアを設定 / 問い合わせます。

構文 :FILE:SAVE:AHISTogram:SELEct {<NRF>}
:FILE:SAVE:AHISTogram:SELEct?

HARMonics の <NRF> = 1

AHISTogram の <NRF> = 1 ~ 2

FFT の <NRF> = 1 ~ 4(4ch モデルでは 1 ~ 2)

SBUS の <NRF> = 1 ~ 4

ZWAVE の <NRF> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 以下は、FFT についての例です。

:FILE:SAVE:FFT:SELECT 1

:FILE:SAVE:FFT:SELECT?

-> :FILE:SAVE:FFT:SELECT 1

:FILE:SAVE:ANAMing

機能 保存するファイル名の自動作成の ON/OFF を設定 / 問い合わせします。

構文 :FILE:SAVE:ANAMing {DATE|DATE2|NUMBERing|OFF|ON}

:FILE:SAVE:ANAMing?

例 :FILE:SAVE:ANAMING DATE

:FILE:SAVE:ANAMING?

-> :FILE:SAVE:ANAMING DATE

:FILE:SAVE:{ASCIi|BINary}:COMPRESSi on

機能 各種データの圧縮保存方法を設定 / 問い合わせます。

構文 :FILE:SAVE:{ASCIi|BINary}:COMPRESSion {DECimation|OFF|PTOPeak}
:FILE:SAVE:{ASCIi|BINary}:COMPRESSion?

例 以下は、波形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

:FILE:SAVE:{ASCIi|BINary|SBUS}:HISTo ry

機能 各種ヒストリ波形の保存方法を設定 / 問い合わせします。

構文 :FILE:SAVE:{ASCIi|BINary|SBUS}:HISTORY {ALL|AVERAGE|ONE}
:FILE:SAVE:{ASCIi|BINary|SBUS}:HISTORY?

例 以下は、波形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}:LENGTH

機能 各種のデータを圧縮保存または間引き保存するときのデータ点数を設定 / 問い合わせします。

構文 :FILE:SAVE:{ASCIi|BINary}:LENGTH {<NRF>}

:FILE:SAVE:{ASCIi|BINary}:LENGTH?

<NRF> = 本体ユーザーズマニュアル機能編参照。

例 以下は、波形データについての例です。

:FILE:SAVE:BINARY:LENGTH 1250

:FILE:SAVE:BINARY:LENGTH?

-> :FILE:SAVE:BINARY:LENGTH 1250

:FILE:SAVE:{ASCIi|BINary}:RANGE

機能 各種データで保存するウィンドウを設定 / 問い合わせします。

構文 :FILE:SAVE:{ASCIi|BINary}:RANGE {MAIN|Z1|Z2}

:FILE:SAVE:{ASCIi|BINary}:RANGE?

例 以下は、波形データについての例です。

:FILE:SAVE:BINARY:RANGE MAIN

:FILE:SAVE:BINARY:RANGE?

-> :FILE:SAVE:BINARY:RANGE MAIN

5.13 FILE グループ

:FILE:SAVE:{ASCIi|BINary}:TRACe

機能 各種データで保存する波形を設定 / 問い合わせします。

構文 :FILE:SAVE:{ASCIi|BINary}:
TRACE {<NRf>|ALL|MATH<x>}
:FILE:SAVE:{ASCIi|BINary}:TRACE?
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 以下は、波形データについての例です。

```
:FILE:SAVE:BINary:TRACE 1  
:FILE:SAVE:BINary:TRACE?  
-> :FILE:SAVE:ASCIi:TRACE 1
```

:FILE:SAVE:ASCIi:TINformation (Time Information)

機能 保存する波形データの時刻情報の有り (ON)/ 無し (OFF) を設定 / 問い合わせをします。

構文 :FILE:SAVE:ASCIi:
TINformation {<Boolean>}
:FILE:SAVE:ASCIi:TINformation?

例 :FILE:SAVE:ASCIi:TINFORMATION ON
:FILE:SAVE:ASCIi:TINFORMATION?
-> :FILE:SAVE:ASCIi:TINFORMATION 1

:FILE:SAVE:COMMeNt

機能 保存するデータのコメントを設定 / 問い合わせします。

構文 :FILE:SAVE:COMMeNt {<文字列>}
:FILE:SAVE:COMMeNt?
<文字列> = 128 文字以内

例 :FILE:SAVE:COMMENT "THIS IS TEST."
:FILE:SAVE:COMMENT?

-> :FILE:SAVE:COMMENT "THIS IS TEST."

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。

:FILE:SAVE:FFT:FINformation (FrequencyInformation)

機能 保存する FFT データの周波数情報の有り (ON)/ 無し (OFF) を設定 / 問い合わせをします。

構文 :FILE:SAVE:FFT:
FINformation {<Boolean>}
:FILE:SAVE:FFT:FINformation?

例 :FILE:SAVE:FFT:FINFORMATION ON
:FILE:SAVE:FFT:FINFORMATION?
-> :FILE:SAVE:FFT:FINFORMATION 1

:FILE:SAVE:NAME

機能 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 "CASE1"
:FILE:SAVE:NAME?
-> :FILE:SAVE:NAME "CASE1"

5.14 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?

: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?

:GONogo:ACTion:BUZZer

機能 判定が NO-GO 時に警告音を鳴らすか鳴らさないかを設定 / 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:HCOpy

機能 判定が NO-GO 時に内蔵プリンタ (オプション) または外部プリンタへハードコピーをとるかをとらないかを設定 / 問い合わせします。

構文 :GONogo:ACTion:HCOpy {<Boolean>}

:GONogo:ACTion:HCOpy?

例 :GONOGO:ACTion:HCOpy ON

:GONOGO:ACTion:HCOpy?

-> :GONOGO:ACTion:HCOpy 1

:GONogo:ACTion:MAIL?

機能 判定が NO-GO 時のメール通知に関するすべての設定値を問い合わせます。

構文 :GONogo:ACTion:MAIL?

:GONogo:ACTion:MAIL:COUNT

機能 判定が NO-GO 時のメールで通知時にメール件数の上限を設定 / 問い合わせします。

構文 :GONogo:ACTion:MAIL:COUNT {<NrF>}

:GONogo:ACTion:MAIL:COUNT?

<NrF> = 1 ~ 1000

例 :GONOGO:ACTion:MAIL:COUNT 100

:GONOGO:ACTion:MAIL:COUNT?

-> :GONOGO:ACTion:MAIL:COUNT 100

:GONogo:ACTion:MAIL:MODE

機能 判定が NO-GO 時にメールで通知するかしないかを設定 / 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

:GONogo:ACTion:SAVE

機能 判定が NO-GO 時にメディアに保存するかしないかを設定 / 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

解説 メディアの種類の設定 / 問い合わせは「:FILE:DIRECTORY:DRIVE」で行います。

:GONogo:COUNT?

機能 実際に行った GO/NO-GO の判定回数を問い合わせます。

構文 :GONogo:COUNT?

例 :GONOGO:COUNT?

-> :GONOGO:COUNT 100

: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 判定の種類を設定 / 問い合わせします。

構文 :GONogo:LOGic {AND|OFF|OR}

:GONogo:LOGic?

例 :GONOGO:LOGIC AND

:GONOGO:LOGIC?

-> :GONOGO:LOGIC AND

:GONogo:NGCount?

機能 GO/NO-GO 判定の NO-GO 回数を問い合わせます。

構文 :GONogo:NGCount?

例 :GONOGO:NGCOUNT?

-> :GONOGO:NGCOUNT 5

5.14 GONogo グループ

:GONogo:NGStopcount

機能 判定終了 NO-GO 回数を設定 / 問い合わせします。

構文 :GONogo:NGStopcount {<Nrf>|INfInity}

:GONogo:NGStopcount?

<Nrf> = 1 ~ 1000

例 :GONOGO:NGSTOPCOUNT 100

:GONOGO:NGSTOPCOUNT?

-> :GONOGO:NGSTOPCOUNT 100

:GONogo: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:STOPcount {<Nrf>|INfInity}

:GONogo:STOPcount?

<Nrf> = 1 ~ 1000000

例 :GONOGO:STOPCOUNT 1000

:GONOGO:STOPCOUNT?

-> :GONOGO:STOPCOUNT 1000

:GONogo:WAIT?

機能 GO/NO-GO 判定の終了をタイムアウト付きで待ちます。

構文 :GONogo:WAIT? {<Nrf>}

<Nrf> = 0 ~ 864000(10ms 単位)

例 (タイムアウトを 5 秒に設定)

:GONOGO:WAIT? 500

-> :GONOGO:WAIT 0

解説 タイムアウトしたときは「1」、時間内に終了したときは「0」が返ります。

: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]?

:GONogo[:ZPARAMeter]:NUMBER<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]:NUMBER<x>?

<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:GONogo[:ZPARAMeter]:NUMBER<x>:CAUSE?

機能 ゾーン / パラメータ判定の各波形パラメータが NO-GO の原因かそうでないかを問い合わせます。

構文 :GONogo[:ZPARAMeter]:NUMBER<x>:

CAUSE?

<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :GONOGO:ZPARAMETER:NUMBER1:CAUSE?

-> :GONOGO:ZPARAMETER:NUMBER1:

CAUSE 1

解説 そのパラメータが NO-GO の原因の場合は「1」、そうでない場合は「0」が返されます。

:GONogo[:ZPARAMeter]:NUMBER<x>:CONDITION

機能 ゾーン / パラメータ判定の各波形パラメータの判定基準を設定 / 問い合わせします。

構文 :GONogo[:ZPARAMeter]:NUMBER<x>:

CONDition {IN|OUT}

:GONogo[:ZPARAMeter]:NUMBER<x>:

CONDition?

<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :GONOGO:ZPARAMETER:NUMBER1:

CONDition IN

:GONOGO:ZPARAMETER:NUMBER1:

CONDition?

-> :GONOGO:ZPARAMETER:NUMBER1:

CONDition IN

:GONogo[:ZPARAMeter]:NUMBER<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]:NUMBER<x>:

MODE {OFF|PARAMeter|POLYgon|

RECTangle|WAVE}

:GONogo[:ZPARAMeter]:NUMBER<x>:MODE?

<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :GONOGO:ZPARAMETER:NUMBER1:

MODE PARAMETER

:GONOGO:ZPARAMETER:NUMBER1:MODE?

-> :GONOGO:ZPARAMETER:NUMBER1:

MODE PARAMETER

:GONogo[:ZPARAMeter]:NUMBER<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]:NUMBER<x>:

PARAMeter?

<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:GONogo[:ZPARAMeter]:NUMBER<x>:PARAMeter:ITEM

機能 パラメータ判定の各波形パラメータのアイテムを設定 / 問い合わせします。

構文 :GONogo[:ZPARAMeter]:NUMBER<x>:
PARAMeter:ITEM {<パラメータ>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
PARAMeter:ITEM?

<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<パラメータ> = {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth|DElay|DT|DUTYcycle|ENUMber|FALL|FREQuency|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWIDth|PERiod|PNUMBER|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEVIation|TY1Integ|TY2Integ|V1|V2}

例 :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:ITEM AVERAGE
:GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:ITEM?
-> :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:ITEM AVERAGE

:GONogo[:ZPARAMeter]:NUMBER<x>:PARAMeter:LIMit

機能 パラメータ判定の各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]:NUMBER<x>:
PARAMeter:LIMit {<NRf>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
PARAMeter:LIMit?

<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = 本体ユーザーズマニュアル機能編参照。

例 :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:LIMIT 0,1
:GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:LIMIT?
-> :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:LIMIT 1.000E+00,0.000E+00

:GONogo[:ZPARAMeter]:NUMBER<x>:PARAMeter:TRACe

機能 パラメータ判定の各波形パラメータの対象波形を設定 / 問い合わせします。

構文 :GONogo[:ZPARAMeter]:NUMBER<x>:
PARAMeter:TRACe {<NRf>|BIT<y>|
FFT<y>|MATH<y>|XY<y>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
PARAMeter:TRACe?

<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<y>, PODB, PODC, PODD の <y> = 0 ~ 7
FFT<y> の <y> = 1 ~ 4(4ch モデルでは 1 ~ 2)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
XY<y> の <y> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:TRACE 1
:GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:TRACE?
-> :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:TRACE 1

:GONogo[:ZPARAMeter]:NUMBER<x>:PARAMeter:VALue?

機能 パラメータ判定の各波形パラメータの測定値を問い合わせます。

構文 :GONogo[:ZPARAMeter]:NUMBER<x>:
PARAMeter:VALue?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:VALUE?
-> :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:VALUE 1.98E-03

解説 モードが OFF など測定不可能な場合は、「NAN」(非数)が返されます。

:GONogo[:ZPARAMeter]:NUMBER<x>:POLYgon?

機能 ポリゴンゾ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[:ZPARAMeter]:NUMBER<x>:
POLYgon?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

5.14 GONogo グループ

:GONogo[:ZPARAMeter]:NUMBER<x>:POLYgon:HPOSITION

機能 ポリゴンゾーン判定で用いる水平位置を設定 / 問い合わせします。

構文 :GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:HPOSITION {<NRf>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:HPOSITION?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = -5 ~ 5(div)

例 :GONOGO:ZPARAMETER:NUMBER1:POLYGON:
HPOSITION 1
:GONOGO:ZPARAMETER:NUMBER1:POLYGON:
HPOSITION?
-> :GONOGO:ZPARAMETER:NUMBER1:
POLYGON:HPOSITION 1.000E+00

解説 TRAcE が XY<x> のときは <NRf> = -4 ~ 4(div) となります。

:GONogo[:ZPARAMeter]:NUMBER<x>:POLYgon:RANGE

機能 ポリゴンゾーン判定で用いる対象ウィンドウを設定 / 問い合わせします。

構文 :GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:RANGE {MAIN|Z1|Z2}
:GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:RANGE?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :GONOGO:ZPARAMETER:NUMBER1:POLYGON:
RANGE MAIN
:GONOGO:ZPARAMETER:NUMBER1:POLYGON:
RANGE?
-> :GONOGO:ZPARAMETER:NUMBER1:
POLYGON:RANGE MAIN

:GONogo[:ZPARAMeter]:NUMBER<x>:POLYgon:TRACE

機能 ポリゴンゾーン判定で用いる対象波形を設定 / 問い合わせします。

構文 :GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:TRACE {<NRf>|MATH<y>|XY<y>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:TRACE?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
XY<y> の <y> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :GONOGO:ZPARAMETER:NUMBER1:POLYGON:
TRACE 1
:GONOGO:ZPARAMETER:NUMBER1:POLYGON:
TRACE?
-> :GONOGO:ZPARAMETER:NUMBER1:
POLYGON:TRACE 1

:GONogo[:ZPARAMeter]:NUMBER<x>:POLYgon:VPOSITION

機能 ポリゴンゾーン判定で用いる垂直位置を設定 / 問い合わせします。

構文 :GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:VPOSITION {<NRf>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:VPOSITION?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = -4 ~ 4(div)

例 :GONOGO:ZPARAMETER:NUMBER1:POLYGON:
VPOSITION 1
:GONOGO:ZPARAMETER:NUMBER1:POLYGON:
VPOSITION?
-> :GONOGO:ZPARAMETER:NUMBER1:
POLYGON:VPOSITION 1.000E+00

:GONogo[:ZPARAMeter]:NUMBER<x>:POLYgon:ZNUMBER

機能 ポリゴンゾーン判定で用いるゾーン番号を設定 / 問い合わせします。

構文 :GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:ZNUMBER {<NRf>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:ZNUMBER?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = 1 ~ 4

例 :GONOGO:ZPARAMETER:NUMBER1:POLYGON:
ZNUMBER 1
:GONOGO:ZPARAMETER:NUMBER1:POLYGON:
ZNUMBER?
-> :GONOGO:ZPARAMETER:NUMBER1:
POLYGON:ZNUMBER 1

:GONogo[:ZPARAMeter]:NUMBER<x>:RECTangle?

機能 方形ゾーン判定に関するすべての設定値を問い合わせします。

構文 :GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:GONogo[:ZPARAMeter]:NUMBER<x>:RECTangle:HORizontal

機能 方形ゾーン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]:NUMBER<x>:
RECTangle:HORizontal {<Nrf>,<Nrf>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle:HORizontal?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<Nrf> = -5 ~ 5(div)

例 :GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:HORIZONTAL 1,2
:GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:HORIZONTAL?
-> :GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:HORIZONTAL 2.000E+00,
1.000E+00

解説 TRaCe が XY<x> のときは <Nrf> = -4 ~ 4(div) となります。

:GONogo[:ZPARAMeter]:NUMBER<x>:RECTangle:RANGE

機能 方形ゾーン判定で用いる四角形の対象ウィンドウを設定 / 問い合わせします。

構文 :GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle:RANGE {MAIN|Z1|Z2}
:GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle:RANGE?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:RANGE MAIN
:GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:RANGE?
-> :GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:RANGE MAIN

:GONogo[:ZPARAMeter]:NUMBER<x>:RECTangle:TRAcE

機能 方形ゾーン判定で用いる四角形の対象波形を設定 / 問い合わせします。

構文 :GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle:TRAcE {<Nrf>|MATH<y>|
XY<y>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle:TRAcE?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 2)
XY<y> の <y> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:TRACE 1
:GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:TRACE?
-> :GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:TRACE 1

:GONogo[:ZPARAMeter]:NUMBER<x>:RECTangle:VERTical

機能 方形ゾーン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]:NUMBER<x>:
RECTangle:VERTical {<Nrf>,<Nrf>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle:VERTical?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<Nrf> = -4 ~ 4(div)

例 :GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:VERTICAL 1,2
:GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:VERTICAL?
-> :GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:VERTICAL 2.000E+00,
1.000E+00

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE?

機能 波形ゾ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[:ZPARAMeter]:NUMBER<x>:WAVE?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:EDIT:EXIT

機能 波形ゾーンの編集メニューから抜けます。

構文 :GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:
EDIT:EXIT {QUIT|STORE}
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :GONOGO:ZPARAMETER:NUMBER1:WAVE:
EDIT:EXIT STORE

解説

- ・ STORE 指定時は、編集内容を保存して編集メニューから抜けます。
- ・ QUIT 指定時は、編集内容を保存せずに編集メニューから抜けます。

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:EDIT:NEW

機能 波形ゾーンの編集で、基本波形を設定します。

構文 :GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:
EDIT:NEW {<Nrf>|MATH<y>}
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 2)
:GONOGO:ZPARAMETER:NUMBER1:WAVE:
EDIT:NEW 1

解説 編集が終了したら「:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:EDIT:EXIT」を利用して、編集メニューから抜ける必要があります。

5.14 GONogo グループ

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:EDIT:PART

機能	波形ゾーンの編集で、部分編集を実行します。
構文	:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE: EDIT:PART {<NRF>,<NRF>,<NRF>,<NRF>} <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <NRF> = -5.00 ~ 5.00(div、T_Range1/T_Range2)、-8.00 ~ 8.00(div(差分)、上下)
例	:GONOGO:ZPARAMETER:NUMBER1:WAVE: EDIT:PART -2.00,2.00,1.00,1.00
解説	編集が終了したら「:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:EDIT:EXIT」を利用して、編集メニューから抜ける必要があります。

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:EDIT:WHOLE

機能	波形ゾーンの編集で、波形全体のゾーンを設定します。
構文	:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE: EDIT:WHOLE {<NRF>,<NRF>,<NRF>,<NRF>} <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <NRF> = -5.00 ~ 5.00(div、左右)、0 ~ 8.00(div、上下)
例	:GONOGO:ZPARAMETER:NUMBER1:WAVE: EDIT:WHOLE 0.50,0.50,1.00,1.00
解説	編集が終了したら「:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:EDIT:EXIT」を利用して、編集メニューから抜ける必要があります。

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:RANGe

機能	波形ゾーン判定で用いる対象ウィンドウを設定 / 問い合わせします。
構文	:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE: RANGe {MAIN Z1 Z2} :GONogo[:ZPARAMeter]:NUMBER<x>:WAVE: RANGe? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例	:GONOGO:ZPARAMETER:NUMBER1:WAVE: RANGe MAIN :GONOGO:ZPARAMETER:NUMBER1:WAVE: RANGe? -> :GONOGO:ZPARAMETER:NUMBER1:WAVE: RANGe MAIN

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:TRACe

機能	波形ゾーン判定で用いる対象波形を設定 / 問い合わせします。
構文	:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE: TRACe {<NRf> MATH<y>} :GONogo[:ZPARAMeter]:NUMBER<x>:WAVE: TRACe? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4) MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 2)
例	:GONOGO:ZPARAMETER:NUMBER1:WAVE: TRACe 1 :GONOGO:ZPARAMETER:NUMBER1:WAVE: TRACe? -> :GONOGO:ZPARAMETER:NUMBER1:WAVE: TRACe 1

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:TRANGe

機能	波形ゾーン判定で用いる判定区間を設定 / 問い合わせします。
構文	:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE: TRANGe {<NRf>,<NRf>} :GONogo[:ZPARAMeter]:NUMBER<x>:WAVE: TRANGe? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <NRf> = -5 ~ 5(div)
例	:GONOGO:ZPARAMETER:NUMBER1:WAVE: TRANGe 1,2 :GONOGO:ZPARAMETER:NUMBER1:WAVE: TRANGe? -> :GONOGO:ZPARAMETER:NUMBER1:WAVE: TRANGe 2.000E+00,1.000E+00

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:ZNUMber

機能	波形ゾーン判定で用いるゾーン番号を設定 / 問い合わせします。
構文	:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE: ZNUMber {<NRf>} :GONogo[:ZPARAMeter]:NUMBER<x>:WAVE: ZNUMber? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <NRf> = 1 ~ 8
例	:GONOGO:ZPARAMETER:NUMBER1:WAVE: ZNUMber 1 :GONOGO:ZPARAMETER:NUMBER1:WAVE: ZNUMber? -> :GONOGO:ZPARAMETER:NUMBER1:WAVE: ZNUMber 1

5.15 HCOpy グループ

:HCOpy?

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?

: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

: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|
PRINter|NETPrinter|FILE|MULTitarget}
: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。

構文 :HCOpy:EXECute

例 :HCOpy:EXECUTE

:HCOpy:EXTPrinter?

機能 外部プリンタへ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:EXTPrinter?

:HCOpy:EXTPrinter:MODE

機能 外部プリンタへ出力するときのノーマルコピー / ハードコピーを設定 / 問い合わせします。

構文 :HCOpy:EXTPrinter:MODE {HARD1|HARD2}
:HCOpy:EXTPrinter:MODE?

例 :HCOpy:EXTPRINTER:MODE HARD1
:HCOpy:EXTPRINTER:MODE?
-> :HCOpy:EXTPRINTER:MODE HARD1

:HCOpy:EXTPrinter:TONE

機能 外部プリンタ出力のカラーを設定 / 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:MULTitarget?

機能 マルチターゲットに関するすべての設定値を問い合わせます。

構文 :HCOpy:MULTitarget?

:HCOpy:MULTitarget:EXTPrinter

機能 マルチターゲットの外部プリンタへの出力を設定 / 問い合わせします。

構文 :HCOpy:MULTitarget:
EXTPrinter {<Boolean>}
:HCOpy:MULTitarget:EXTPrinter?

例 :HCOpy:MULTITARGET:EXTPRINTER ON
:HCOpy:MULTITARGET:EXTPRINTER?
-> :HCOpy:MULTITARGET:EXTPRINTER 1

:HCOpy:MULTitarget:PRINter

機能 マルチターゲットの内部プリンタへの出力を設定 / 問い合わせします。

構文 :HCOpy:MULTitarget:
PRINter {<Boolean>}
:HCOpy:MULTitarget:PRINter?
例 :HCOpy:MULTITARGET:PRINTER ON
:HCOpy:MULTITARGET:PRINTER?
-> :HCOpy:MULTITARGET:PRINTER 1

:HCOpy:MULTitarget:NETPrinter

機能 マルチターゲットのネットワークプリンタへの出力を設定 / 問い合わせします。

構文 :HCOpy:MULTitarget:
NETPrinter {<Boolean>}
:HCOpy:MULTitarget:NETPrinter?
例 :HCOpy:MULTITARGET:NETPRINTER ON
:HCOpy:MULTITARGET:NETPRINTER?
-> :HCOpy:MULTITARGET:NETPRINTER 1

:HCOpy:MULTitarget:FILE

機能 マルチターゲットのファイルへの出力を設定 / 問い合わせします。

構文 :HCOpy:MULTitarget:FILE {<Boolean>}
:HCOpy:MULTitarget:FILE?
例 :HCOpy:MULTITARGET:FILE ON
:HCOpy:MULTITARGET:FILE?
-> :HCOpy:MULTITARGET:FILE 1

5.15 HCOpy グループ

:HCOpy:MULTitarget:WAVEform

機能 マルチターゲットの波形ファイルへの出力を設定 / 問い合わせします。

構文 :HCOpy:MULTitarget:
WAVEform{<Boolean>}
:HCOpy:MULTitarget:WAVEform?

例 :HCOpy:MULTITARGET:WAVEFORM ON
:HCOpy:MULTITARGET:WAVEFORM?
-> :HCOpy:MULTITARGET:WAVEFORM 1

:HCOpy:NETPrint?

機能 ネットワークプリンタへ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:NETPrint?

:HCOpy:NETPrint:MODE

機能 ネットワークプリンタへ出力するときのノーマルコピー / ハードコピーを設定 / 問い合わせします。

構文 :HCOpy:NETPrint:MODE {HARD1|HARD2}
:HCOpy:NETPrint:MODE?

例 :HCOpy:NETPRINT:MODE HARD1
:HCOpy:NETPRINT:MODE?
-> :HCOpy:NETPRINT:MODE HARD1

:HCOpy:NETPrint:TONE

機能 ネットワークプリンタ出力のカラーを設定 / 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

:HCOpy:NETPrint:TYPE

機能 ネットワークプリンタへの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 EINKJET
:HCOpy:NETPRINT:TYPE?
-> :HCOpy:NETPRINT:TYPE EINKJET

:HCOpy:PRINter?

機能 内蔵プリンタへ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:PRINter?

:HCOpy:PRINter:MODE

機能 内蔵プリンタへ出力するときのショートコピー / ハードコピーを設定 / 問い合わせします。

構文 :HCOpy:PRINter:MODE {HARD1|HARD2}
:HCOpy:PRINter:MODE?

例 :HCOpy:PRINTER:MODE HARD1
:HCOpy:PRINTER:MODE?
-> :HCOpy:PRINTER:MODE HARD1

:HCOpy:PRINter:REPort

機能 内蔵プリンタへ付加情報を出力する / しないを設定 / 問い合わせします。

構文 :HCOpy:PRINter:REPort {<Boolean>}
:HCOpy:PRINter:REPort?

例 :HCOpy:PRINTER:REPORT ON
:HCOpy:PRINTER:REPORT?
-> :HCOpy:PRINTER:REPORT 1

5.16 HISTory グループ

:HISTory?

機能 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?

:HISTory:AVERage

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:AVERage {<Boolean>}

:HISTory:AVERage?

例 :HISTORY: AVERAGE ON

:HISTORY: AVERAGE?

-> :HISTORY: AVERAGE 1

:HISTory:DISPlay

機能 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:DISPlay {<NRf>,<NRf>}

:HISTory:DISPlay?

<NRf> = 本体ユーザーズマニュアル機能編参照。

例 :HISTORY: DISPLAY 0,-100

:HISTORY: DISPLAY?

-> :HISTORY: DISPLAY 0,-100

:HISTory:DMODE

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:DMODE {ALL|COLor|INTensity|ONE}

:HISTory:DMODE?

例 :HISTORY: DMODE ALL

:HISTORY: DMODE?

-> :HISTORY: DMODE ALL

:HISTory: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:RECORD {<NRf>|MINimum}

:HISTory:RECORD?

<NRf> = 本体ユーザーズマニュアル機能編参照。

例 :HISTORY: RECORD 0

:HISTORY: RECORD?

-> :HISTORY: 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: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値を問い合わせます。

構文 :HISTory:REPLay?

:HISTory: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:REPLay:JUMP {MAXimum|

MINimum}

例 :HISTORY: REPLAY: JUMP MAXIMUM

:HISTory: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:REPLay:SPEEd {<NRf>|PER3|

PER10|PER30|PER60}

:HISTory:REPLay:SPEEd?

<NRf> = 1、3、10

例 :HISTORY: REPLAY: SPEED 1

:HISTORY: REPLAY: SPEED?

-> :HISTORY: REPLAY: SPEED 1

:HISTory: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:REPLay:START {MAXimum|

MINimum}

例 :HISTORY: REPLAY: START MAXIMUM

:HISTory:REPLay: STOP

機能 ヒストリ波形のリプレイを停止します。

構文 :HISTory:REPLay:STOP

例 :HISTORY: REPLAY: STOP

:HISTory:SEARCh?

機能 ヒストリ波形の検索に関するすべての設定値を問い合わせます。

構文 :HISTory:SEARCh?

5.16 HISTory グループ

:HISTory[:SEARCh]:ABORt

機能 検索を中止します。
 構文 :HISTory[:SEARCh]:ABORt
 例 :HISTORY:SEARCH:ABORT

:HISTory[:SEARCh]:EXECute

機能 検索を実行します。オーバーラップコマンドです。
 構文 :HISTory[:SEARCh]:EXECute
 例 :HISTORY:SEARCH:EXECUTE

:HISTory[:SEARCh]:LOGic

機能 ヒストリ波形の検索の論理を設定 / 問い合わせします。
 構文 :HISTory[:SEARCh]:LOGic {AND|OR|SIMPlE}
 :HISTory[:SEARCh]:LOGic?
 例 :HISTORY:SEARCH:LOGIC AND
 :HISTORY:SEARCH:LOGIC?
 -> :HISTORY:SEARCH:LOGIC AND

:HISTory[:SEARCh]:NUMBer<x>?

機能 各検索条件に関するすべての設定値を問い合わせます。
 構文 :HISTory[:SEARCh]:NUMBer<x>?
 <x> = 1 ~ 8

:HISTory[:SEARCh]:NUMBer<x>:CONDitiOn

機能 各検索条件の判定基準を設定 / 問い合わせします。
 構文 :HISTory[:SEARCh]:NUMBer<x>:
 CONDitiOn {IN|OUT}
 :HISTory[:SEARCh]:NUMBer<x>:
 CONDitiOn?
 <x> = 1 ~ 8
 例 :HISTORY:SEARCH:NUMBER1:CONDITION IN
 :HISTORY:SEARCH:NUMBER1:CONDITION?
 -> :HISTORY:SEARCH:NUMBER1:
 CONDITION IN

:HISTory[:SEARCh]:NUMBer<x>:MODE

機能 各検索条件のモードを設定 / 問い合わせします。
 構文 :HISTory[:SEARCh]:NUMBer<x>:
 MODE {OFF|PARAMeter|POLYgon|RECTangle|WAVE}
 :HISTory[:SEARCh]:NUMBer<x>:MODE?
 <x> = 1 ~ 8
 例 :HISTORY:SEARCH:NUMBER1:
 MODE PARAMETER
 :HISTORY:SEARCH:NUMBER1:MODE?
 -> :HISTORY:SEARCH:NUMBER1:
 MODE PARAMETER

:HISTory[:SEARCh]:NUMBer<x>:PARAmeter?

機能 パラメータ検索に関するすべての設定値を問い合わせます。
 構文 :HISTory[:SEARCh]:NUMBer<x>:
 PARAmeter?
 <x> = 1 ~ 8

:HISTory[:SEARCh]:NUMBer<x>:PARAmeter:ITEM

機能 パラメータ検索の各波形パラメータのアイテムを設定 / 問い合わせします。
 構文 :HISTory[:SEARCh]:NUMBer<x>:
 PARAmeter:ITEM { /パラメータ }
 :HISTory[:SEARCh]:NUMBer<x>:
 PARAmeter:ITEM?
 <x> = 1 ~ 8
 <パラメータ> = {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth|DElay|DT|DUTYcycle|ENUMber|FALL|FREQuency|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWIDth|PERiod|PNUmber|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEViation|TY1Integ|TY2Integ|V1|V2}
 例 :HISTORY:SEARCH:NUMBER1:PARAMETER:
 ITEM AVERAGE
 :HISTORY:SEARCH:NUMBER1:PARAMETER:
 ITEM?
 -> :HISTORY:SEARCH:NUMBER1:
 PARAMETER:ITEM AVERAGE

:HISTory[:SEARCh]:NUMBer<x>:PARAmeter:LIMit

機能 パラメータ検索の各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[:SEARCh]:NUMBer<x>:
 PARAmeter:LIMit {<NRf>}
 :HISTory[:SEARCh]:NUMBer<x>:
 PARAmeter:LIMit?
 <x> = 1 ~ 8
 <NRf> = 本体ユーザーズマニュアル機能編参照。
 例 :HISTORY:SEARCH:NUMBER1:PARAMETER:
 LIMIT 0,1
 :HISTORY:SEARCH:NUMBER1:PARAMETER:
 LIMIT?
 -> :HISTORY:SEARCH:NUMBER1:
 PARAMETER:LIMIT 1.000E+00,0.000E+00

:HISTory[:SEARCH]:NUMBER<x>:PARAMETER:TRACE

機能 パラメータ検索の各波形パラメータの対象波形を設定 / 問い合わせします。

構文 :HISTory[:SEARCH]:NUMBER<x>:
PARAMETER:TRACE {<Nrf>|BIT<y>|
FFT<y>|MATH<y>|XY<y>}
:HISTory[:SEARCH]:NUMBER<x>:
PARAMETER:TRACE?
<x> = 1 ~ 8
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 2)
PODA<y>、PODB、PODC、PODD の <y> = 0
~ 7
FFT<y> の <y> = 1 ~ 4(4ch モデルでは 1 ~ 2)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
XY<y> の <y> = 1 ~ 4(4ch モデルでは 1 ~ 2)
例 :HISTory:SEARCH:NUMBER1:PARAMETER:
TRACE 1
:HISTory:SEARCH:NUMBER1:PARAMETER:
TRACE?
-> :HISTory:SEARCH:NUMBER1:
PARAMETER:TRACE 1

:HISTory[:SEARCH]:NUMBER<x>:PARAMETER:VALUE?

機能 パラメータ検索の各波形パラメータの測定値を問い合わせます。

構文 :HISTory[:SEARCH]:NUMBER<x>:
PARAMETER:VALUE?
<x> = 1 ~ 8

例 :HISTory:SEARCH:NUMBER1:PARAMETER:
VALUE?
-> :HISTory:SEARCH:NUMBER1:
PARAMETER:VALUE 1.98E-03

解説 モードが OFF など測定不可能な場合は、「NAN」(非数) が返されます。

:HISTory[:SEARCH]:NUMBER<x>:POLYgon?

機能 ポリゴンゾーン検索に関するすべての設定値を問い合わせます。

構文 :HISTory[:SEARCH]:NUMBER<x>:POLYgon?
<x> = 1 ~ 8

:HISTory[:SEARCH]:NUMBER<x>:POLYgon:HPOSITION

機能 ポリゴンゾーン検索で用いる水平位置を設定 / 問い合わせします。

構文 :HISTory[:SEARCH]:NUMBER<x>:POLYgon:
HPOSITION {<Nrf>}
:HISTory[:SEARCH]:NUMBER<x>:POLYgon:
HPOSITION?
<x> = 1 ~ 8
<Nrf> = -5 ~ 5(div)

例 :HISTory:SEARCH:NUMBER1:POLYgon:
HPOSITION 1
:HISTory:SEARCH:NUMBER1:POLYgon:
HPOSITION?
-> :HISTory:SEARCH:NUMBER1:POLYgon:
HPOSITION 1.000E+00

解説 TRACE が XY<x> のときは <Nrf> = -4 ~ 4(div) となります。

:HISTory[:SEARCH]:NUMBER<x>:POLYgon:RANGE

機能 ポリゴンゾーン検索で用いる対象ウィンドウを設定 / 問い合わせします。

構文 :HISTory[:SEARCH]:NUMBER<x>:POLYgon:
RANGE {MAIN|Z1|Z2}
:HISTory[:SEARCH]:NUMBER<x>:POLYgon:
RANGE?
<x> = 1 ~ 8

例 :HISTory:SEARCH:NUMBER1:POLYgon:
RANGE MAIN
:HISTory:SEARCH:NUMBER1:POLYgon:
RANGE?
-> :HISTory:SEARCH:NUMBER1:POLYgon:
RANGE MAIN

:HISTory[:SEARCH]:NUMBER<x>:POLYgon:TRACE

機能 ポリゴンゾーン検索で用いる対象波形を設定 / 問い合わせします。

構文 :HISTory[:SEARCH]:NUMBER<x>:POLYgon:
TRACE {<Nrf>|MATH<y>|XY<y>}
:HISTory[:SEARCH]:NUMBER<x>:POLYgon:
TRACE?
<x> = 1 ~ 8
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 2)

MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
XY<y> の <y> = 1 ~ 4(4ch モデルでは 1 ~ 2)
例 :HISTory:SEARCH:NUMBER1:POLYgon:
TRACE 1
:HISTory:SEARCH:NUMBER1:POLYgon:
TRACE?
-> :HISTory:SEARCH:NUMBER1:POLYgon:
TRACE 1

5.16 HISTory グループ

:HISTory[:SEARCh]:NUMBer<x>:POLYgon:VPOSition

機能 ポリゴンゾーン検索で用いる垂直位置を設定 / 問い合わせします。

構文 :HISTory[:SEARCh]:NUMBer<x>:POLYgon:
VPOSition {<NRf>}
:HISTory[:SEARCh]:NUMBer<x>:POLYgon:
VPOSition?
<x> = 1 ~ 8
<NRf> = -4 ~ 4(div)

例 :HISTORY:SEARCH:NUMBER1:POLYGON:
VPOSITION 1
:HISTORY:SEARCH:NUMBER1:POLYGON:
VPOSITION?
-> :HISTORY:SEARCH:NUMBER1:POLYGON:
VPOSITION 1.000E+00

:HISTory[:SEARCh]:NUMBer<x>:POLYgon:ZNUMBer

機能 ポリゴンゾーン検索で用いるゾーン番号を設定 / 問い合わせします。

構文 :HISTory[:SEARCh]:NUMBer<x>:POLYgon:
ZNUMBer {<NRf>}
:HISTory[:SEARCh]:NUMBer<x>:POLYgon:
ZNUMBer?
<x> = 1 ~ 8
<NRf> = 1 ~ 4

例 :HISTORY:SEARCH:NUMBER1:POLYGON:
ZNUMBER 1
:HISTORY:SEARCH:NUMBER1:POLYGON:
ZNUMBER?
-> :HISTORY:SEARCH:NUMBER1:POLYGON:
ZNUMBER 1

:HISTory[:SEARCh]:NUMBer<x>:RECTangle?

機能 方形ゾーン検索に関するすべての設定値を問い合わせます。

構文 :HISTory[:SEARCh]:NUMBer<x>:
RECTangle?
<x> = 1 ~ 8

:HISTory[:SEARCh]:NUMBer<x>:RECTangle:HORizontal

機能 方形ゾーン検索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[:SEARCh]:NUMBer<x>:
RECTangle:HORizontal {<NRf>,<NRf>}
:HISTory[:SEARCh]:NUMBer<x>:
RECTangle:HORizontal?
<x> = 1 ~ 8
<NRf> = -5 ~ 5(div)

例 :HISTORY:SEARCH:NUMBER1:RECTANGLE:
HORIZONTAL 1,2
:HISTORY:SEARCH:NUMBER1:RECTANGLE:
HORIZONTAL?
-> :HISTORY:SEARCH:NUMBER1:
RECTANGLE:HORIZONTAL 2.000E+00,
1.000E+00

解説 TRACe が XY<x> のときは <NRf> = -4 ~ 4(div) となります。

:HISTory[:SEARCh]:NUMBer<x>:RECTangle:RANGe

機能 方形ゾーン検索で用いる四角形の対象ウィンドウを設定 / 問い合わせします。

構文 :HISTory[:SEARCh]:NUMBer<x>:
RECTangle:RANGe {MAIN|Z1|Z2}
:HISTory[:SEARCh]:NUMBer<x>:
RECTangle:RANGe?
<x> = 1 ~ 8

例 :HISTORY:SEARCH:NUMBER1:RECTANGLE:
RANGE MAIN
:HISTORY:SEARCH:NUMBER1:RECTANGLE:
RANGE?
-> :HISTORY:SEARCH:NUMBER1:
RECTANGLE:RANGE MAIN

:HISTory[:SEARCh]:NUMBer<x>:RECTangle:TRACe

機能 方形ゾーン検索で用いる四角形の対象波形を設定 / 問い合わせします。

構文 :HISTory[:SEARCh]:NUMBer<x>:
RECTangle:TRACe {<NRf>|MATH<y>|
XY<y>}
:HISTory[:SEARCh]:NUMBer<x>:
RECTangle:TRACe?
<x> = 1 ~ 8
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 2)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
XY<y> の <y> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :HISTORY:SEARCH:NUMBER1:RECTANGLE:
TRACE 1
:HISTORY:SEARCH:NUMBER1:RECTANGLE:
TRACE?
-> :HISTORY:SEARCH:NUMBER1:
RECTANGLE:TRACE 1

:HISTory[:SEARCH]:NUMBER<x>:RECTangle:VERTical

機能 方形ゾーン検索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[:SEARCH]:NUMBER<x>:
RECTangle:VERTical {<Nrf>,<Nrf>}
:HISTory[:SEARCH]:NUMBER<x>:
RECTangle:VERTical?
<x> = 1 ~ 8
<Nrf> = -4 ~ 4(div)

例 :HISTORY:SEARCH:NUMBER1:RECTANGLE:
VERTICAL 1,2
:HISTORY:SEARCH:NUMBER1:RECTANGLE:
VERTICAL?
-> :HISTORY:SEARCH:NUMBER1:
RECTANGLE:VERTICAL 2.000E+00,
1.000E+00

:HISTory[:SEARCH]:NUMBER<x>:WAVE?

機能 波形ゾーン検索に関するすべての設定値を問い合わせします。

構文 :HISTory[:SEARCH]:NUMBER<x>:WAVE?
<x> = 1 ~ 8

:HISTory[:SEARCH]:NUMBER<x>:WAVE:RANGe

機能 波形ゾーン検索で用いる対象ウィンドウを設定 / 問い合わせします。

構文 :HISTory[:SEARCH]:NUMBER<x>:WAVE:
RANGe {MAIN|Z1|Z2}
:HISTory[:SEARCH]:NUMBER<x>:WAVE:
RANGe?
<x> = 1 ~ 8

例 :HISTORY:SEARCH:NUMBER1:WAVE:
RANGE MAIN
:HISTORY:SEARCH:NUMBER1:WAVE:RANGE?
-> :HISTORY:SEARCH:NUMBER1:WAVE:
RANGE MAIN

:HISTory[:SEARCH]:NUMBER<x>:WAVE:TRACe

機能 波形ゾーン検索で用いる対象波形を設定 / 問い合わせします。

構文 :HISTory[:SEARCH]:NUMBER<x>:WAVE:
TRACe {<Nrf>|MATH<y>}
:HISTory[:SEARCH]:NUMBER<x>:WAVE:
TRACe?
<x> = 1 ~ 8

例 :HISTORY:SEARCH:NUMBER1:WAVE:TRACE 1
:HISTORY:SEARCH:NUMBER1:WAVE:TRACE?
-> :HISTORY:SEARCH:NUMBER1:WAVE:
TRACE 1

:HISTory[:SEARCH]:NUMBER<x>:WAVE:TRANge

機能 波形ゾーン検索で用いる判定区間を設定 / 問い合わせします。

構文 :HISTory[:SEARCH]:NUMBER<x>:WAVE:
TRANge {<Nrf>,<Nrf>}
:HISTory[:SEARCH]:NUMBER<x>:WAVE:
TRANge?
<Nrf> = -5 ~ 5(div)

例 :HISTORY:SEARCH:NUMBER1:WAVE:
TRANGE 1,2
:HISTORY:SEARCH:NUMBER1:WAVE:TRANGE?
-> :HISTORY:SEARCH:NUMBER1:WAVE:
TRANGE 2.000E+00,1.000E+00

:HISTory[:SEARCH]:NUMBER<x>:WAVE:ZNUMber

機能 波形ゾーン検索で用いるゾーン番号を設定 / 問い合わせします。

構文 :HISTory[:SEARCH]:NUMBER<x>:WAVE:
ZNUMber {<Nrf>}
:HISTory[:SEARCH]:NUMBER<x>:WAVE:
ZNUMber?
<x> = 1 ~ 8
<Nrf> = 1 ~ 4

例 :HISTORY:SEARCH:NUMBER1:WAVE:
ZNUMBER 1
:HISTORY:SEARCH:NUMBER1:WAVE:
ZNUMBER?
-> :HISTORY:SEARCH:NUMBER1:WAVE:
ZNUMBER 1

:HISTory[:SEARCH]:RESet

機能 ヒストリ波形の検索条件をリセットします。

構文 :HISTory[:SEARCH]:RESet

例 :HISTORY:SEARCH:RESET

:HISTory[:SEARCH]:SIMPlE?

機能 簡易検索に関するすべての設定値を問い合わせします。

構文 :HISTory[:SEARCH]:SIMPlE?

:HISTory[:SEARCH]:SIMPlE:HORizontal

機能 簡易検索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[:SEARCH]:SIMPlE:
HORizontal {<Nrf>,<Nrf>}
:HISTory[:SEARCH]:SIMPlE:HORizontal?
<Nrf> = -5 ~ 5(div)

例 :HISTORY:SEARCH:SIMPLE:HORIZONTAL 1,2
:HISTORY:SEARCH:SIMPLE:HORIZONTAL?
-> :HISTORY:SEARCH:SIMPLE:
HORIZONTAL 2.000E+00,1.000E+00

5.16 HISTory グループ

:HISTory[:SEARCh]:SIMPlE:RANGe

機能 簡易検索で用いる四角形の対象ウインドウを設定 / 問い合わせします。

構文 :HISTory[:SEARCh]:SIMPlE:

RANGe {MAIN|Z1|Z2}

:HISTory[:SEARCh]:SIMPlE:RANGe?

例 :HISTORY:SEARCH:SIMPLE:RANGE MAIN

:HISTORY:SEARCH:SIMPLE:RANGE?

-> :HISTORY:SEARCH:SIMPLE:RANGE MAIN

:HISTory[:SEARCh]:SIMPlE:TRACe

機能 簡易検索で用いる四角形の対象トレースを設定 / 問い合わせします。

構文 :HISTory[:SEARCh]:SIMPlE:

TRACe {<NRf>|MATH<x>|XY<x>}

:HISTory[:SEARCh]:SIMPlE:TRACe?

<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 2)

MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)

XY<y> の <y> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:HISTory[:SEARCh]:SIMPlE:VERTical

機能 簡易検索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[:SEARCh]:SIMPlE:

VERTical {<NRf>,<NRf>}

:HISTory[:SEARCh]:SIMPlE:VERTical?

<NRf> = -4 ~ 4(div)

:HISTory:TIME?

機能 対象レコード番号の時間を問い合わせます。

構文 :HISTory:TIME? {<NRf>|MINimum}

例 :HISTORY:TIME? -100

-> :HISTORY:TIME " -100 10:20:30.400"

解説 「MINimum」を指定すると、最小のレコード番号になります。

5.17 IMAGE グループ

: IMAGE?

機能 画面イメージ出力に関するすべての設定値を問い合わせます。

構文 :IMAGE?

: IMAGE:ABORt

機能 画面イメージをメディアに保存するのを中止します。

構文 :IMAGE:ABORt

例 :IMAGE:ABORt

: IMAGE:BACKground

機能 画面イメージのバックグラウンドを設定 / 問い合わせします。

構文 :IMAGE:BACKground {NORMal|
TRANSPARENT}
:IMAGE:BACKground?

例 :IMAGE:BACKGROUND NORMAL
:IMAGE:BACKGROUND?
-> :IMAGE:BACKGROUND NORMAL

: IMAGE:EXECute

機能 画面イメージをメディアに保存します。

構文 :IMAGE:EXECute
:IMAGE:EXECUTE

: IMAGE:FORMat

機能 画面イメージの出力形式を設定 / 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:INformation

機能 画面イメージに設定情報を付加する / しないを設定 / 問い合わせします。

構文 :IMAGE:INformation {<Boolean>}
:IMAGE:INformation?

例 :IMAGE:INFORMATION ON
:IMAGE:INFORMATION?
-> :IMAGE:INFORMATION 1

: IMAGE:MODE

機能 画面イメージの出力方式を設定 / 問い合わせします。

構文 :IMAGE:MODE {HARD1|HARD2|WIDE}
:IMAGE:MODE?

例 :IMAGE:MODE HARD1
:IMAGE:MODE?
-> :IMAGE:MODE HARD1

: IMAGE:SAVE?

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 :IMAGE:SAVE?

: IMAGE:SAVE:ANAMing

機能 出力ファイル名の自動作成の ON/OFF を設定 / 問い合わせします。

構文 :IMAGE:SAVE:ANAMing {DATE|DATE2|
NUMBERing|OFF}
:IMAGE:SAVE:ANAMing?

例 :IMAGE:SAVE:ANAMING DATE
:IMAGE:SAVE:ANAMING?
-> :IMAGE:SAVE:ANAMING DATE

: IMAGE:SAVE:CDIRectory

機能 ファイルのディレクトリを変更します。

構文 :IMAGE:SAVE:CDIRectory {<文字列>}
<文字列> = 本体ユーザーズマニュアル機能編
参照。

例 :IMAGE:SAVE:CDIRECTORY "ABC"

: IMAGE:SAVE:DRIVE

機能 作成するファイルのメディアを設定します。

構文 :IMAGE:SAVE:DRIVE {FLASHmem|NETWork|
USB,<Nrf>}
<Nrf> = 0 ~ 3

例 :IMAGE:SAVE:DRIVE FLASHMEM
解説 「USB」の <Nrf> は、パーティションまたは LUN
で区切られていない場合は省略できます。

: IMAGE:SAVE:NAME

機能 作成するファイルのファイル名を設定 / 問い合わせします。

構文 :IMAGE:SAVE:NAME {<Filename>}
:IMAGE:SAVE:NAME?

例 :IMAGE:SAVE:NAME "DISP_1"
:IMAGE:SAVE:NAME?
-> :IMAGE:SAVE:NAME "DISP_1"

5.17 IMAGE グループ

: IMAGE: SEND?

機能	画面イメージの値を問い合わせます。
構文	:IMAGE:SEND?
例	:IMAGE:SEND? -> :IMAGE: SEND #8(8 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、4-7 ページを参照してください。

: IMAGE: TONE

機能	出力する画面イメージの色調を設定 / 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

: IMAGE: WGRadation

機能	出力する画面イメージの波形部分の階調の有無を設定 / 問い合わせします。
構文	:IMAGE:WGRadation {<boolean>} :IMAGE:WGRadation?
例	:IMAGE:WGRADATION ON :IMAGE:WGRADATION? -> :IMAGE:WGRADATION 1
解説	「:IMAGE:TONE REVerse」のときのみ有効になります。

5.18 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

5.19 LOGic グループ

PODC と PODD を使用した通信コマンドは、32 ビット入力 (L4 オプション) 対応のモデルで使用できます。

:LOGic?

機能 ロジック入力波形に関するすべての設定値を問い合わせます。

構文 :LOGic?

:LOGic:BITOrder

機能 ロジック入力のビットオーダーを設定 / 問い合わせします。

構文 :LOGic:BITOrder {<文字列>}

:LOGic:BITOrder?

<文字列> = "A0" ~ "A7"、"B0" ~ "B7"、"C0" ~ "C7"、"D0" ~ "D7"、"BUS1" ~ "BUS4" の組み合わせ (80 文字)

例 :LOGIC:BITORDER "BUS1BUS2BUS3
BUS4 A0A1A2A3A4A5A6A7 B0B1B2B3B4B5B6
B7 C0C1C2C3C4C5C6C7 D0D1D2D3D4D5D6D7"
:LOGIC:BITORDER?
-> :LOGIC:BITORDER "BUS1BUS2BUS3
BUS4 A0A1A2A3A4A5A6A7 B0B1B2B3B4B5B6
B7 C0C1C2C3C4C5C6C7 D0D1D2D3D4D5D6D7"

解説

- 各文字列の間にスペースを合計 8 個まで入力できます。
- 32 ビット入力 (L4 オプション) に対応していない場合、<文字列> = "A0" ~ "A7"、"B0" ~ "B7"、"BUS1" ~ "BUS2" の組み合わせ (40 文字) と、各文字列の間にスペースを合計 4 個まで入力できます。

:LOGic:BUS<x>?

機能 ロジック入力のポートの各バスのすべての設定値を問い合わせます。

構文 :LOGic:BUS<x>?

<x> = 1 ~ 4

:LOGic:BUS<x>:ASSignment

機能 ロジック入力のポートの各バスのアサインを設定 / 問い合わせします。

構文 :LOGic:BUS<x>:ASSignment {<文字列>}

:LOGic:BUS<x>:ASSignment?

<x> = 1 ~ 4

<文字列> = "A0" ~ "A7"、"B0" ~ "B7"、"C0" ~ "C7"、"D0" ~ "D7" の組み合わせ (64 文字以内)

例 :LOGIC:BUS1:
ASSIGNMENT "A0A1A2A3A4A5A6A7B0B1B2B3
B4B5B6B7C0C1C2C3C4C5C6C7D0D1D2D3D4D5D
6D7"
:LOGIC:BUS1:ASSIGNMENT?
-> :LOGIC:BUS1:ASSIGNMENT "A0A1A2A3A4
A5A6A7B0B1B2B3B4B5B6B7C0C1C2C3C4C5C6C
7D0D1D2D3D4D5D6D7"

解説 32 ビット入力 (L4 オプション) に対応していない場合、<文字列> = "A0" ~ "A7"、"B0" ~ "B7" の組み合わせ (32 文字以内) が入力できます。

:LOGic:BUS<x>:DISPlay

機能 ロジック入力のポート各バスの表示の ON/OFF を設定 / 問い合わせします。

構文 :LOGic:BUS<x>:DISPlay {<Boolean>}

:LOGic:BUS<x>:DISPlay?

<x> = 1 ~ 4

例 :LOGIC:BUS1:DISPLAY ON
:LOGIC:BUS1:DISPLAY?
-> :LOGIC:BUS1:DISPLAY ON

:LOGic:BUS<x>:FORmat

機能 ロジック入力のポートの各バスの表示形式 (バス表示) を設定 / 問い合わせします。

構文 :LOGic:BUS<x>:FORmat {HEXa|BINary}

:LOGic:BUS<x>:FORmat?

<x> = 1 ~ 4

例 :LOGIC:BUS1:FORMAT HEXA
:LOGIC:BUS1:FORMAT?
-> :LOGIC:BUS1:FORMAT HEXA

:LOGic:BUS<x>:LABel

機能 ロジック入力ポートのラベル名を設定 / 問い合わせします。

構文 :LOGic:BUS<x>:LABel {<文字列>}
:LOGic:BUS<x>:LABel?
<x> = 1 ~ 4
<文字列> = 8 文字以内

例 :LOGic:BUS1:LABel "ABC"
:LOGic:BUS1:LABel?
-> :LOGic:BUS1:LABel "ABC"

:LOGic:MODE

機能 ロジック入力ポートの 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:PODA?

機能 ロジック入力ポート A のすべての設定値を問い合わせします。

構文 :LOGic:PODA?

:LOGic:PODA:ALL?

機能 ロジック入力ポート A の全ビットのすべての設定値を問い合わせします。

構文 :LOGic:PODA:ALL?

:LOGic:PODA:ALL:DISPlay

機能 ロジック入力ポート A の全ビットの表示の ON/OFF を設定 / 問い合わせします。

構文 :LOGic:PODA:ALL:DISPlay {<Boolean>}
:LOGic:PODA:ALL:DISPlay?
例 :LOGic:PODA:ALL:DISPlay ON
:LOGic:PODA:ALL:DISPlay?
-> :LOGic:PODA:ALL:DISPlay 1

:LOGic:PODA:ALL:LEVel

機能 ロジック入力ポート A のユーザー定義でのスレシヨルドレベルを設定 / 問い合わせします。

構文 :LOGic:PODA:ALL:LEVel {<電圧>}
:LOGic:PODA:ALL:LEVel?
<電圧> = -10V ~ 10V、0.1V ステップ

例 :LOGic:PODA:ALL:LEVel 1V
:LOGic:PODA:ALL:LEVel?
-> :LOGic:PODA:ALL:LEVel 1.0E+00

:LOGic:PODA:ALL:TYPE

機能 ロジック入力ポート A のスレシヨルドレベルの選択を実行します。

構文 :LOGic:PODA:TYPE {CMOS1|CMOS2|CMOS3|CMOS5|ECL|USER}

例 :LOGic:PODA:ALL:TYPE CMOS1

:LOGic:PODA:BIT<x>?

機能 ロジック入力ポート A の各ビットのすべての設定値を問い合わせします。

構文 :LOGic:PODA:BIT<x>?
<x> = 0 ~ 7

:LOGic:PODA:BIT<x>:DISPlay

機能 ロジック入力ポート A の各ビットの表示の ON/OFF を設定 / 問い合わせします。

構文 :LOGic:PODA:BIT<x>:
DISPlay {<Boolean>}
:LOGic:PODA:BIT<x>:DISPlay?
<x> = 0 ~ 7

例 :LOGic:PODA:BIT1:DISPlay ON
:LOGic:PODA:BIT1:DISPlay?
-> :LOGic:PODA:BIT1:DISPlay 1

:LOGic:PODA:BIT<x>:LABel

機能 ロジック入力ポート A の各ビットのラベル名を設定 / 問い合わせします。

構文 :LOGic:PODA:BIT<x>:LABel {<文字列>}
:LOGic:PODA:BIT<x>:LABel?
<x> = 0 ~ 7
<文字列> = 8 文字以内

例 :LOGic:PODA:BIT1:LABel "NO _ 1"
:LOGic:PODA:BIT1:LABel?
-> :LOGic:PODA:BIT1:LABel "NO _ 1"

:LOGic:PODA:BIT<x>:LEVel

機能 ロジック入力ポート A の各ビットのユーザー定義でのスレシヨルドレベルを設定 / 問い合わせします。

構文 :LOGic:PODA:BIT<x>:LEVel {<電圧>}
:LOGic:PODA:BIT<x>:LEVel?
<x> = 0 ~ 7
<電圧> = -10V ~ 10V、0.1V ステップ

例 :LOGic:PODA:BIT1:LEVel 1V
:LOGic:PODA:BIT1:LEVel?
-> :LOGic:PODA:BIT1:LEVel 1.0E+00

5.19 LOGic グループ

:LOGic:PODA:BIT<x>:TYPE

機能 ロジック入力ポート A の各ビットのスレッシュ
ールドレベルを設定 / 問い合わせします。

構文 :LOGic:PODA:BIT<x>:TYPE {CMOS1|
CMOS2|CMOS3|CMOS5|ECL|USER}
<x> = 0 ~ 7

例 :LOGic:PODA:BIT1:TYPE CMOS1

:LOGic:PODA:DESKew

機能 ロジック入力ポート A のスキュー補正を設定 /
問い合わせします。

構文 :LOGic:PODA:DESKew {<時間>}
:LOGic:PODA:DESKew?
<時間> = -100ns ~ 100ns、10ps ステップ

例 :LOGic:PODA:DESKew 1NS
:LOGic:PODA:DESKew?
-> :LOGic:PODA:DESKew 1.000E-09

:LOGic:PODA:HYSTeresis

機能 ロジック入力ポート A のヒステリシスを設定 /
問い合わせします。

構文 :LOGic:PODA:HYSTeresis {HIGH|LOW}
:LOGic:PODA:HYSTeresis?

例 :LOGic:PODA:HYSTERESIS HIGH
:LOGic:PODA:HYSTERESIS?
-> :LOGic:PODA:HYSTERESIS HIGH

:LOGic:PODB?

機能 ロジック入力ポート B のすべての設定値を問
い合わせます。

構文 :LOGic:PODB?

:LOGic:PODB:ALL?

機能 ロジック入力ポート B の全ビットのすべての
設定値を問い合わせます。

構文 :LOGic:PODB:ALL?

:LOGic:PODB:ALL:DISPlay

機能 ロジック入力ポート B の全ビットの表示の
ON/OFF を設定 / 問い合わせします。

構文 :LOGic:PODB:ALL:DISPlay {<Boolean>}
:LOGic:PODB:ALL:DISPlay?

例 :LOGic:PODB:ALL:DISPlay ON
:LOGic:PODB:ALL:DISPlay?
-> :LOGic:PODB:ALL:DISPlay 1

:LOGic:PODB:ALL:LEVel

機能 ロジック入力ポート B のユーザー定義でのス
レッシュレベルを設定 / 問い合わせします。

構文 :LOGic:PODB:ALL:LEVel {<電圧>}
:LOGic:PODB:ALL:LEVel?
<電圧> = -10V ~ 10V、0.1V ステップ

例 :LOGic:PODB:ALL:LEVel 1V
:LOGic:PODB:ALL:LEVel?
-> :LOGic:PODB:ALL:LEVel 1.0E+00

:LOGic:PODB:ALL:TYPE

機能 ロジック入力ポート B のスレッシュレベル
の選択を実行します。

構文 :LOGic:PODB:TYPE {CMOS1|CMOS2|CMOS3|
CMOS5|ECL|USER}

例 :LOGic:PODB:ALL:TYPE CMOS1

:LOGic:PODB:BIT<x>?

機能 ロジック入力ポート B の各ビットのすべての
設定値を問い合わせます。

構文 :LOGic:PODB:BIT<x>?
<x> = 0 ~ 7

:LOGic:PODB:BIT<x>:DISPlay

機能 ロジック入力ポート B の各ビットの表示の
ON/OFF を設定 / 問い合わせします。

構文 :LOGic:PODB:BIT<x>:
DISPlay {<Boolean>}
:LOGic:PODB:BIT<x>:DISPlay?
<x> = 0 ~ 7

例 :LOGic:PODB:BIT1:DISPlay ON
:LOGic:PODB:BIT1:DISPlay?
-> :LOGic:PODB:BIT1:DISPlay 1

:LOGic:PODB:BIT<x>:LABel

機能 ロジック入力ポート B の各ビットのラベル名
を設定 / 問い合わせします。

構文 :LOGic:PODB:BIT<x>:LABel {<文字列>}
:LOGic:PODB:BIT<x>:LABel?
<x> = 0 ~ 7
<文字列> = 8 文字以内

例 :LOGic:PODB:BIT1:LABel "NO _ 1"
:LOGic:PODB:BIT1:LABel?
-> :LOGic:PODB:BIT1:LABel "NO _ 1"

:LOGic:PODB:BIT<x>:LEVel

機能 ロジック入力ポート B の各ビットのユーザー定義でのスレシヨルドレベルを設定 / 問い合わせします。

構文 :LOGic:PODB:BIT<x>:LEVel {< 電圧 >}
:LOGic:PODB:BIT<x>:LEVel?
<x> = 0 ~ 7

例 < 電圧 > = -10V ~ 10V、0.1V ステップ
:LOGic:PODB:BIT1:LEVel 1V
:LOGic:PODB:BIT1:LEVel?
-> :LOGic:PODB:BIT1:LEVel 1.0E+00

:LOGic:PODB:BIT<x>:TYPE

機能 ロジック入力ポート B の各ビッ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:PODB:BIT<x>:TYPE {CMOS1|
CMOS2|CMOS3|CMOS5|ECL|USER}
<x> = 0 ~ 7

例 :LOGic:PODB:BIT1:TYPE CMOS1

:LOGic:PODB:DESKew

機能 ロジック入力ポート B のスキュー補正を設定 / 問い合わせします。

構文 :LOGic:PODB:DESKew {< 時間 >}
:LOGic:PODB:DESKew?
< 時間 > = -100ns ~ 100ns、10ps ステップ

例 :LOGic:PODB:DESKew 1NS
:LOGic:PODB:DESKew?
-> :LOGic:PODB:DESKew 1.000E-09

:LOGic:PODB:HYSTeresis

機能 ロジック入力ポート B のヒステリシスを設定 / 問い合わせします。

構文 :LOGic:PODB:HYSTeresis {HIGH|LOW}
:LOGic:PODB:HYSTeresis?

例 :LOGic:PODB:HYSTERESIS HIGH
:LOGic:PODB:HYSTERESIS?
-> :LOGic:PODB:HYSTERESIS HIGH

:LOGic:PODC?

機能 ロジック入力ポート C のすべての設定値を問い合わせします。

構文 :LOGic:PODC?

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:PODC:ALL?

機能 ロジック入力ポート C の全ビットのすべての設定値を問い合わせします。

構文 :LOGic:PODC:ALL?

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:PODC:ALL:DISPlay

機能 ロジック入力ポート C の全ビットの表示の ON/OFF を設定 / 問い合わせします。

構文 :LOGic:PODC:ALL:DISPlay {<Boolean>}
:LOGic:PODC:ALL:DISPlay?

例 :LOGic:PODC:ALL:DISPlay ON
:LOGic:PODC:ALL:DISPlay?
-> :LOGic:PODC:ALL:DISPlay 1

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:PODC:ALL:LEVel

機能 ロジック入力ポート C のユーザー定義でのスレシヨルドレベルを設定 / 問い合わせします。

構文 :LOGic:PODC:ALL:LEVel {< 電圧 >}
:LOGic:PODC:ALL:LEVel?
< 電圧 > = -10V ~ 10V、0.1V ステップ

例 :LOGic:PODC:ALL:LEVel 1V
:LOGic:PODC:ALL:LEVel?
-> :LOGic:PODC:ALL:LEVel 1.0E+00

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:PODC:ALL:TYPE

機能 ロジック入力ポート C のスレシヨルドレベルの選択を実行します。

構文 :LOGic:PODC:TYPE {CMOS1|CMOS2|CMOS3|
CMOS5|ECL|USER}

例 :LOGic:PODC:ALL:TYPE CMOS1

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:PODC:BIT<x>?

機能 ロジック入力ポート C の各ビットのすべての設定値を問い合わせします。

構文 :LOGic:PODC:BIT<x>?

<x> = 0 ~ 7

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:PODC:BIT<x>:DISPlay

機能 ロジック入力ポート C の各ビットの表示の ON/OFF を設定 / 問い合わせします。

構文 :LOGic:PODC:BIT<x>:
DISPlay {<Boolean>}
:LOGic:PODC:BIT<x>:DISPlay?

<x> = 0 ~ 7

例 :LOGic:PODC:BIT1:DISPlay ON
:LOGic:PODC:BIT1:DISPlay?
-> :LOGic:PODC:BIT1:DISPlay 1

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

5.19 LOGic グループ

:LOGic:PODC:BIT<x>:LABel

機能 ロジック入力ポート C の各ビットのラベル名を設定 / 問い合わせします。

構文 :LOGic:PODC:BIT<x>:LABel {<文字列>}
:LOGic:PODC:BIT<x>:LABel?

<x> = 0 ~ 7

<文字列> = 8 文字以内

例 :LOGic:PODC:BIT1:LABel "NO _ 1"
:LOGic:PODC:BIT1:LABel?

-> :LOGic:PODC:BIT1:LABel "NO _ 1"

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:PODC:BIT<x>:LEVel

機能 ロジック入力ポート C の各ビットのユーザー定義でのスレシヨルドレベルを設定 / 問い合わせします。

構文 :LOGic:PODC:BIT<x>:LEVel {<電圧>}
:LOGic:PODC:BIT<x>:LEVel?

<x> = 0 ~ 7

<電圧> = -10V ~ 10V、0.1V ステップ

例 :LOGic:PODC:BIT1:LEVel 1V
:LOGic:PODC:BIT1:LEVel?

-> :LOGic:PODC:BIT1:LEVel 1.0E+00

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:PODC:BIT<x>:TYPE

機能 ロジック入力ポート C の各ビッ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:PODC:BIT<x>:TYPE {CMOS1|
CMOS2|CMOS3|CMOS5|ECL|USER}

<x> = 0 ~ 7

例 :LOGic:PODC:BIT1:TYPE CMOS1

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:PODC:DESKew

機能 ロジック入力ポート C のスキュー補正を設定 / 問い合わせします。

構文 :LOGic:PODC:DESKew {<時間>}
:LOGic:PODC:DESKew?

<時間> = -100ns ~ 100ns、10ps ステップ

例 :LOGic:PODC:DESKew 1NS
:LOGic:PODC:DESKew?

-> :LOGic:PODC:DESKew 1.000E-09

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:PODC:HYSTeresis

機能 ロジック入力ポート C のヒステリシスを設定 / 問い合わせします。

構文 :LOGic:PODC:HYSTeresis {HIGH|LOW|USER}
:LOGic:PODC:HYSTeresis?

例 :LOGic:PODC:HYSTeresis HIGH

:LOGic:PODC:HYSTeresis?

-> :LOGic:PODC:HYSTeresis HIGH

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:PODD?

機能 ロジック入力ポート D のすべての設定値を問い合わせします。

構文 :LOGic:PODD?

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:PODD:ALL?

機能 ロジック入力ポート D の全ビットのすべての設定値を問い合わせします。

構文 :LOGic:PODD:ALL?

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:PODD:ALL:DISPlay

機能 ロジック入力ポート D の全ビットの表示の ON/OFF を設定 / 問い合わせします。

構文 :LOGic:PODD:ALL:DISPlay {<Boolean>}
:LOGic:PODD:ALL:DISPlay?

例 :LOGic:PODD:ALL:DISPlay ON

:LOGic:PODD:ALL:DISPlay?

-> :LOGic:PODD:ALL:DISPlay 1

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:PODD:ALL:LEVel

機能 ロジック入力ポート D のユーザー定義でのスレシヨルドレベルを設定 / 問い合わせします。

構文 :LOGic:PODD:ALL:LEVel {<電圧>}
:LOGic:PODD:ALL:LEVel?

<電圧> = -10V ~ 10V、0.1V ステップ

例 :LOGic:PODD:ALL:LEVel 1V

:LOGic:PODD:ALL:LEVel?

-> :LOGic:PODD:ALL:LEVel 1.0E+00

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:PODD:ALL:TYPE

機能 ロジック入力ポート D のスレシヨルドレベルの選択を実行します。

構文 :LOGic:PODD:TYPE {CMOS1|CMOS2|CMOS3|CMOS5|ECL|USER}

例 :LOGic:PODD:ALL:TYPE CMOS1

解説 32 ビット入力 (L4 オプション) に対応していないと、エラーになります。

:LOGic:PODD:BIT<x>?

機能 ロジック入力ポート D の各ビットのすべての設定値を問い合わせます。

構文 :LOGic:PODD:BIT<x>?

<x> = 0 ~ 7

解説 32 ビット入力 (L4 オプション) に対応していないと、エラーになります。

:LOGic:PODD:BIT<x>:DISPlay

機能 ロジック入力ポート D の各ビットの表示の ON/OFF を設定 / 問い合わせします。

構文 :LOGic:PODD:BIT<x>:DISPlay {<Boolean>}

:LOGic:PODD:BIT<x>:DISPlay?

<x> = 0 ~ 7

例 :LOGic:PODD:BIT1:DISPlay ON

:LOGic:PODD:BIT1:DISPlay?

-> :LOGic:PODD:BIT1:DISPlay 1

解説 32 ビット入力 (L4 オプション) に対応していないと、エラーになります。

:LOGic:PODD:BIT<x>:LABel

機能 ロジック入力ポート D の各ビットのラベル名を設定 / 問い合わせします。

構文 :LOGic:PODD:BIT<x>:LABel {<文字列>}

:LOGic:PODD:BIT<x>:LABel?

<x> = 0 ~ 7

<文字列> = 8 文字以内

例 :LOGic:PODD:BIT1:LABel "NO _ 1"

:LOGic:PODD:BIT1:LABel?

-> :LOGic:PODD:BIT1:LABel "NO _ 1"

解説 32 ビット入力 (L4 オプション) に対応していないと、エラーになります。

:LOGic:PODD:BIT<x>:LEVel

機能 ロジック入力ポート D の各ビットのユーザー定義でのスレシヨルドレベルを設定 / 問い合わせします。

構文 :LOGic:PODD:BIT<x>:LEVel {<電圧>}

:LOGic:PODD:BIT<x>:LEVel?

<x> = 0 ~ 7

<電圧> = -10V ~ 10V、0.1V ステップ

例 :LOGic:PODD:BIT1:LEVel 1V

:LOGic:PODD:BIT1:LEVel?

-> :LOGic:PODD:BIT1:LEVel 1.0E+00

解説 32 ビット入力 (L4 オプション) に対応していないと、エラーになります。

:LOGic:PODD:BIT<x>:TYPE

機能 ロジック入力ポート D の各ビッ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:PODD:BIT<x>:TYPE {CMOS1|CMOS2|CMOS3|CMOS5|ECL|USER}

<x> = 0 ~ 7

例 :LOGic:PODD:BIT1:TYPE CMOS1

解説 32 ビット入力 (L4 オプション) に対応していないと、エラーになります。

:LOGic:PODD:DESKew

機能 ロジック入力ポート D のスキュー補正を設定 / 問い合わせします。

構文 :LOGic:PODD:DESKew {<時間>}

:LOGic:PODD:DESKew?

<時間> = -100ns ~ 100ns、10ps ステップ

例 :LOGic:PODD:DESKew 1NS

:LOGic:PODD:DESKew?

-> :LOGic:PODD:DESKew 1.000E-09

解説 32 ビット入力 (L4 オプション) に対応していないと、エラーになります。

:LOGic:PODD:HYSTeresis

機能 ロジック入力ポート D のヒステリシスを設定 / 問い合わせします。

構文 :LOGic:PODD:HYSTeresis {HIGH|LOW}

:LOGic:PODD:HYSTeresis?

例 :LOGic:PODD:HYSTERESIS HIGH

:LOGic:PODD:HYSTERESIS?

-> :LOGic:PODD:HYSTERESIS HIGH

解説 32 ビット入力 (L4 オプション) に対応していないと、エラーになります。

5.19 LOGic グループ

: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> = -7 ~ 39

例 :LOGic:POStion 0
:LOGic:POStion?
-> :LOGic:POStion 0

: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|MEDium|SMALL}
:LOGic:SIZE?

例 :LOGic:SIZE LARGE
:LOGic:SIZE?
-> :LOGic:SIZE LARGE

:LOGic:StAte?

機能 ロジック入力の状態表示に関するすべての設定値を問い合わせます。

構文 :LOGic:StAte?

:LOGic:StAte:ASSignment?

機能 ロジック入力の状態表示のアサインに関するすべての設定値を問い合わせます。

構文 :LOGic:StAte:ASSignment?
例 :LOGic:PODA:StAte:ASSignment?

解説 32 ビット入力 (I4 オプション) に対応していないと、エラーになります。

:LOGic:StAte:ASSignment:ALL

機能 ロジック入力の状態表示のアサインのすべての設定値を設定 / 問い合わせします。

構文 :LOGic:StAte:ASSignment:
ALL {<Boolean>}
:LOGic:StAte:ASSignment:ALL?
例 :LOGic:PODA:StAte:ASSignment:ALL ON
:LOGic:PODA:StAte:ASSignment:ALL?
-> :LOGic:PODA:StAte:ASSignment:
ALL 1

:LOGic:StAte:ASSignment:BUS<x>

機能 ロジック入力の状態表示のアサインのバスの設定値を設定 / 問い合わせします。

構文 :LOGic:StAte:ASSignment:
BUS<x> {<Boolean>}
:LOGic:StAte:ASSignment:BUS<x>?
<x> = 1 ~ 4

例 :LOGic:PODA:StAte:ASSignment:BUS1 ON
:LOGic:PODA:StAte:ASSignment:BUS1?
-> :LOGic:PODA:StAte:ASSignment:
BUS1 1

:LOGic:StAte:ASSignment: {PODA<x>|PODB<x>|PODC<x>|PODD<x>}

機能 ロジック入力の状態表示のアサインの各ビットのすべての設定値を設定 / 問い合わせします。

構文 :LOGic:StAte:ASSignment:{PODA<x>|
PODB<x>|PODC<x>|PODD<x>} {<Boolean>}
:LOGic:StAte:ASSignment:{PODA<x>|
PODB<x>|PODC<x>|PODD<x>}?
<x> = 0 ~ 7

例 :LOGic:PODA:StAte:ASSignment:BIT1 ON
:LOGic:PODA:StAte:ASSignment:BIT1?
-> :LOGic:PODA:StAte:ASSignment:
BIT1 1

解説 32 ビット入力 (I4 オプション) に対応していないと、PODC<x>、PODD<x> はエラーになります。

:LOGic:StAte:CLOCK

機能 ロジック入力の状態表示の基準クロック波形を設定 / 問い合わせします。

構文 :LOGic:StAte:CLOCK {PODA<x>|PODB<x>|
PODC<x>|PODD<x>}
:LOGic:StAte:CLOCK?
<x> = 0 ~ 7

例 :LOGic:PODA:StAte:CLOCK 1
:LOGic:PODA:StAte:CLOCK?
-> :LOGic:PODA:StAte:CLOCK 1

解説 32 ビット入力 (I4 オプション) に対応していないと、PODC<x>、PODD<x> はエラーになります。

:LOGic:StAte:MODE

機能 ロジック入力の状態表示の ON/OFF を設定 / 問い合わせします。

構文 :LOGic:StAte:MODE {<Boolean>}
:LOGic:StAte:MODE?
例 :LOGic:PODA:StAte:MODE ON
:LOGic:PODA:StAte:MODE?
-> :LOGic:PODA:StAte:MODE 1

:LOGic:StAte:POLarity

機能 ロジック入力の状態表示の基準クロック波形の極性を設定 / 問い合わせします。

構文 :LOGic:StAte:POLarity {RISE|FALL|
BOTH}
:LOGic:StAte:POLarity?

例 :LOGic:PODA:StAte:POLARITY RISE
:LOGic:PODA:StAte:POLARITY?
-> :LOGic:PODA:StAte:POLARITY RISE

5.20 MATH グループ

ユーザー定義演算はオプションです。ユーザー定義演算のオプション付でないモデルでは、ユーザー定義演算に関するコマンドは使用できません。

: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(4ch モデルでは 1 ~ 4)

:MATH<x>:DISPlay

機能 演算波形を表示する (ON)/ 表示しない (OFF) を設定 / 問い合わせします。

構文 :MATH<x>:DISPlay {<Boolean>}
:MATH<x>:DISPlay?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :MATH1:DISPlay ON
:MATH1:DISPlay?
-> :MATH1:DISPlay 1

:MATH<x>:ECount?

機能 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 :MATH<x>:ECount?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:MATH<x>:ECount:HYSteresis

機能 エッジカウント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(4ch モデルでは 1 ~ 4)
<NRf> = 0(div) ~ 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(4ch モデルでは 1 ~ 4)

例 :MATH1:ECount:POLarity FALL
:MATH1:ECount:POLarity?
-> :MATH1:ECount:POLarity FALL

:MATH<x>:ECount:THReshold

機能 エッジ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>:ECount:THReshold {<NRf>|
<電圧>|<電流>}
:MATH<x>:ECount:THReshold?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<電圧>、<電流> = 本体ユーザーズマニュアル
機能編参照。

例 :MATH1:ECount:THRESHOLD 1
:MATH1:ECount:THRESHOLD?
-> :MATH1:ECount:THRESHOLD 1.000E+00

:MATH<x>:FILTer?

機能 フィルター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(4ch モデルでは 1 ~ 4)

:MATH<x>:FILTer:FORDER

機能 IIR フィルターのフィルター次数を設定 / 問い合わせします。

構文 :MATH<x>:FILTer:FORDER {<NRf>}
:MATH<x>:FILTer:FORDER?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = 1、2

例 :MATH1:FILTer:FORDER 1
:MATH1:FILTer:FORDER?
-> :MATH1:FILTer:FORDER 1

:MATH<x>:FILTer:HCUToff

機能 IIR ハイパスフィルター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:HCUToff {<周波数>}
:MATH<x>:FILTer:HCUToff?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<周波数> = 0.01(Hz) ~ 500M(Hz)

例 :MATH1:FILTer:HCUTOFF 10MHZ
:MATH1:FILTer:HCUTOFF?
-> :MATH1:FILTer:HCUTOFF 10.00E+06

5.20 MATH グループ

:MATH<x>:FILTer:LCUToff

機能 IIR ローパスフィルター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:LCUToff {< 周波数 >}
:MATH<x>:FILTer:LCUToff?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
< 周波数 > = 0.01(Hz) ~ 500M(Hz)

例 :MATH1:FILTER:LCUTOFF 10HZ
:MATH1:FILTER:LCUTOFF?
-> :MATH1:FILTER:LCUTOFF 10.00E+00

:MATH<x>:FILTer:TIME

機能 位相シフトの遅延時間を設定 / 問い合わせします。

構文 :MATH<x>:FILTer:TIME {< 時間 >}
:MATH<x>:FILTer:TIME?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
< 時間 > = 本体ユーザーズマニュアル機能編参照。

例 :MATH1:FILTER:TIME 1S
:MATH1:FILTER:TIME?
-> :MATH1:FILTER:TIME 1.000E+00

:MATH<x>:FILTer:TYPE

機能 フィルターのタイプを設定 / 問い合わせします。

構文 :MATH<x>:FILTer:TYPE {DElay|MAVG|
IHPass|ILPass}
:MATH<x>:FILTer:TYPE?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :MATH1:FILTER:TYPE DELAY
:MATH1:FILTER:TYPE?
-> :MATH1:FILTER:TYPE DELAY

:MATH<x>:FILTer:WEIGHT

機能 移動平均の重みを設定 / 問い合わせします。

構文 :MATH<x>:FILTer:WEIGHT {<NRf>}
:MATH<x>:FILTer:WEIGHT?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = 2 ~ 128、2n ステップ

例 :MATH1:FILTER:WEIGHT 2
:MATH1:FILTER:WEIGHT?
-> :MATH1:FILTER:WEIGHT 2

:MATH<x>:INTEgral?

機能 積分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(4ch モデルでは 1 ~ 4)

:MATH<x>:INTEgral:SPOint

機能 積分の開始位置を設定 / 問い合わせします。

構文 :MATH<x>:INTEgral:SPOint {<NRf>}
:MATH<x>:INTEgral:SPOint?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = -5 ~ 5(div)、(10div/ 表示レコード長)
ステップ

例 :MATH1:INTEGRAL:SPOINT 1.5
:MATH1:INTEGRAL:SPOINT?
-> :MATH1:INTEGRAL:SPOINT 1.5000000

: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ます。

構文 :MATH<x>:LABel?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

: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(4ch モデルでは 1 ~ 4)
< 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(4ch モデルでは 1 ~ 4)

例 :MATH1:LABEL:MODE ON
:MATH1:LABEL:MODE?
-> :MATH1:LABEL:MODE 1

:MATH<x>:OPERation

機能	演算子を設定 / 問い合わせします。
構文	:MATH<x>:OPERation {ECount FILTer INTEgral MINus MULTiple PLUS RCount USERdefine},{<NRf> MATH<y>}[, {<NRf> MATH<y>}], [{<NRf> MATH<y>} NONE] :MATH<x>:OPERation? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4) <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例	:MATH1:OPERATION PLUS,1,2 :MATH1:OPERATION? -> :MATH1:OPERATION PLUS,1,2
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) の場合は、最初の <NRf> に 1 項目の対象波形を選択し、2 つ目の <NRf> に 2 項目の対象波形を選択します。 ・3項演算子 (RCount) の場合は、最初の <NRf> に 1 項目の対象波形を選択し、2 つ目の <NRf> に 2 項目の対象波形を選択し、3 つ目の <NRf> に 3 項目の対象波形を選択します。 ・USERdefine 演算子の場合、<NRf> は必要ありません。

:MATH<x>:RCount?

機能	ロータリ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:RCOUNT? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:MATH<x>:RCount:COUNT?

機能	ロータリカウント演算のカウント条件に関するすべての設定値を問い合わせます。
構文	:MATH<x>:RCount:COUNT? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例	:MATH1:RCOUNT:COUNT?

:MATH<x>:RCount:COUNT:CWTiming<y>

機能	ロータリカウント演算の CW(clockwise) のカウントタイミングを設定 / 問い合わせします。
構文	:MATH<x>:RCount:COUNT:CWTiming<y> {ARISe AFALl BRISe BFALl} :MATH<x>:RCount:COUNT:CWTiming<y>? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <y> = 1, 2
例	:MATH1:RCOUNT:COUNT:CWTIMING1 ARISE :MATH1:RCOUNT:COUNT:CWTIMING1? -> :MATH1:RCOUNT:COUNT:CWTIMING1 ARISE

:MATH<x>:RCount:COUNT:MULTi

機能	ロータリカウント演算の通倍を設定 / 問い合わせします。
構文	:MATH<x>:RCount:COUNT:MULTi <NRf> :MATH<x>:RCount:COUNT:MULTi? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <NRf> = 1, 2, 4
例	:MATH1:RCOUNT:COUNT:MULTI 4 :MATH1:RCOUNT:COUNT:MULTI? -> :MATH1:RCOUNT:COUNT:MULTI 4

:MATH<x>:RCount:HYSTeresis<y>

機能	ロータリカウント演算のヒステリシスを設定 / 問い合わせします。
構文	:MATH<x>:RCount:HYSTeresis<y> {<NRf>} :MATH<x>:RCount:HYSTeresis<y>? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <y> = 1 ~ 3 <NRf> = 0.3 ~ 8.0(div)
例	:MATH1:RCOUNT:HYSTERESIS1 0.3 :MATH1:RCOUNT:HYSTERESIS1? -> :MATH1:RCOUNT:HYSTERESIS1 0.3000

:MATH<x>:RCount:RESet?

機能	ロータリカウント演算のリセット条件に関するすべての設定値を問い合わせます。
構文	:MATH<x>:RCount:RESet? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例	:MATH1:RCOUNT:RESET?

:MATH<x>:RCount:RESet:CWTiming

機能	ロータリカウント演算の CW(clockwise) のリセットタイミングを設定 / 問い合わせします。
構文	:MATH<x>:RCount:RESet:CWTiming {ARISe AFALl BRISe BFALl} :MATH<x>:RCount:RESet:CWTiming? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例	:MATH1:RCOUNT:RESET:CWTIMING ARISE :MATH1:RCOUNT:RESET:CWTIMING? -> :MATH1:RCOUNT:RESET:CWTIMING ARISE

:MATH<x>:RCount:RESet:RTIMing

機能	ロータリカウント演算のリセットタイミングを設定 / 問い合わせします。
構文	:MATH<x>:RCount:RESet:RTIMing {ZONLy EONZ} :MATH<x>:RCount:RESet:RTIMing? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例	:MATH1:RCOUNT:RESET:RTIMING ZONLY :MATH1:RCOUNT:RESET:RTIMING? -> :MATH1:RCOUNT:RESET:RTIMING ZONLY

5.20 MATH グループ

:MATH<x>:RCOunt:THReshold<y>

機能 ロータリカウント演算の判定レベルを設定 / 問い合わせします。

構文 :MATH<x>:RCOunt:THReshold<y> {<NRf>|<電圧>|<電流>}
:MATH<x>:RCOunt:THReshold<y>?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<y> = 1 ~ 3
<電圧>、<電流> = 本体ユーザーズマニュアル機能編参照。

例 :MATH1:RCOUNT:THRESHOLD1 1
:MATH1:RCOUNT:THRESHOLD1?
-> :MATH1:RCOUNT:
THRESHOLD1 1.000E+00

:MATH<x>:RCOunt:ZPOLarity

機能 ロータリカウント演算の極性を設定 / 問い合わせします。

構文 :MATH<x>:RCOunt:ZPOLarity {NEGative|POSitive}
:MATH<x>:RCOunt:ZPOLarity?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :MATH1:RCOUNT:ZPOLARITY NEGATIVE
:MATH1:RCOUNT:ZPOLARITY?
-> :MATH1:RCOUNT:ZPOLARITY NEGATIVE

:MATH<x>:SCALE?

機能 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>:SCALE?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:MATH<x>:SCALE:CENTer

機能 マニュアルスケーリング時の中心値を設定 / 問い合わせします。

構文 :MATH<x>:SCALE:CENTer {<NRf>}
:MATH<x>:SCALE:CENTer?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = -1.0000E+31 ~ 1.0000E+31

例 :MATH1:SCALE:CENTER 1
:MATH1:SCALE:CENTER?
-> :MATH1:SCALE:CENTER 1.00000E+00

:MATH<x>:SCALE:MODE

機能 スケーリングの方法を設定 / 問い合わせします。

構文 :MATH<x>:SCALE:MODE {AUTO|MANual}
:MATH<x>:SCALE:MODE?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :MATH1:SCALE:MODE AUTO
:MATH1:SCALE:MODE?
-> :MATH1:SCALE:MODE AUTO

:MATH<x>:SCALE:SENSitivity

機能 マニュアルスケーリング時の中心からのスパンを設定 / 問い合わせします。

構文 :MATH<x>:SCALE:SENSitivity {<NRf>}
:MATH<x>:SCALE:SENSitivity?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = -1.0000E-31 ~ 1.0000E+31

例 :MATH1:SCALE:SENSITIVITY 10
:MATH1:SCALE:SENSITIVITY?
-> :MATH1:SCALE:
SENSITIVITY 10.0000E+00

: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ます。

構文 :MATH<x>:UNIT?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

: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(4ch モデルでは 1 ~ 4)
<文字列> = 4 文字以内

例 :MATH1:UNIT:DEFINE "EU"
:MATH1:UNIT:DEFINE?
-> :MATH1:UNIT:DEFINE "EU"

解説 単位は、スケール値に反映されます。演算結果に影響を及ぼすことはありません。

: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(4ch モデルでは 1 ~ 4)

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ます。

構文 :MATH<x>:USERdefine?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:MATH<x>:USERdefine:AVERage?

機能 ユーザー定義演算のアベレ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>:USERdefine:AVERage?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

解説 MATH1 ~ MATH8 で共通のコマンドです。

:MATH<x>:USERdefine:AVERage:EWEight

機能	ユーザー定義演算の指数化平均の減衰定数を設定 / 問い合わせします。
構文	:MATH<x>:USERdefine:AVERage: EWEight {<Nrf>}
	:MATH<x>:USERdefine:AVERage:EWEight? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <Nrf> = 2 ~ 1024、2n ステップ
例	:MATH1:USERDEFINE:AVERAGE:EWEIGHT 2 :MATH1:USERDEFINE:AVERAGE:EWEIGHT? -> :MATH1:USERDEFINE:AVERAGE: EWEIGHT 2
解説	MATH1 ~ MATH8 で共通のコマンドです。

:MATH<x>:USERdefine:AVERage:MODE

機能	ユーザー定義演算のアベレージングモードを設定 / 問い合わせします。
構文	:MATH<x>:USERdefine:AVERage: MODE {EXponent OFF}
	:MATH<x>:USERdefine:AVERage:MODE? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例	:MATH1:USERDEFINE:AVERAGE: MODE EXPONENT :MATH1:USERDEFINE:AVERAGE:MODE? -> :MATH1:USERDEFINE:AVERAGE: MODE EXPONENT
解説	MATH1 ~ MATH8 で共通のコマンドです。

:MATH<x>:USERdefine:CONSItant<y>

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: CONSItant<y> {<Nrf>}
	:MATH<x>:USERdefine:CONSItant<y>? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <y> = 1 ~ 4 <Nrf> = -1.0000E+31 ~ 1.0000E+31
例	:MATH1:USERDEFINE:CONSITANT1 1 :MATH1:USERDEFINE:CONSITANT1? -> :MATH1:USERDEFINE: CONSITANT1 1.0000E+00
解説	MATH1 ~ MATH8 で共通のコマンドです。

: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 ~ 8(4ch モデルでは 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。

:MATH<x>:USERdefine:FILTer<y>?

機能	ユーザー定義演算のフィルターに関するすべての設定値を問い合わせます。
構文	:MATH<x>:USERdefine:FILTer<y>?
	<x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <y> = 1、2
解説	MATH1 ~ MATH8 で共通のコマンドです。

:MATH<x>:USERdefine:FILTer<y>:BAND

機能	ユーザー定義演算のフィルターのバンド (周波数帯) を設定 / 問い合わせします。
構文	:MATH<x>:USERdefine:FILTer<y>: BAND {BPASS HPASS LPASS}
	:MATH<x>:USERdefine:FILTer<y>:BAND? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <y> = 1、2
例	:MATH1:USERDEFINE:FILTER1:BAND BPASS :MATH1:USERDEFINE:FILTER1:BAND? -> :MATH1:USERDEFINE:FILTER1: BAND BPASS
解説	MATH1 ~ MATH8 で共通のコマンドです。

:MATH<x>:USERdefine:FILTer<y>:CUTOff<z>

機能	ユーザー定義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>:USERdefine:FILTer<y>: CUTOff<z> {<Nrf>}
	:MATH<x>:USERdefine:FILTer<y>: CUTOff<z>? <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <y> = 1、2 <z> = 1、2 <Nrf> = 2 ~ 30(%), (0.2% ステップ)
例	:MATH1:USERDEFINE:FILTER1:CUTOFF1 10 :MATH1:USERDEFINE:FILTER1:CUTOFF1? -> :MATH1:USERDEFINE:FILTER1: CUTOFF1 10.0
解説	「:MATH<x>:USERdefine:FILTer<x>:BAND BPASS」以外は CUTOff2 は設定できません。 MATH1 ~ MATH8 で共通のコマンドです。

5.20 MATH グループ

:MATH<x>:USERdefine:FILTer<y>:TYPE

機能 ユーザー定義演算のフィルターのタイプを設定 / 問い合わせします。

構文 :MATH<x>:USERdefine:FILTer<y>:
TYPE {IIR|FIR}
:MATH<x>:USERdefine:FILTer<y>:TYPE?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<y> = 1, 2

例 :MATH1:USERDEFINE:FILTER1:TYPE IIR
:MATH1:USERDEFINE:FILTER1:TYPE?
-> :MATH1:USERDEFINE:FILTER1:
TYPE IIR

解説 MATH1 ~ MATH8 で共通のコマンドです。

:MATH<x>:USERdefine:HISTory:ABORt

機能 ヒストリ波形のユーザー定義演算 (Math on History) を取り消します。

構文 :MATH<x>:USERdefine:HISTory:ABORt
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :MATH1:USERDEFINE:HISTORY:ABORT

:MATH<x>:USERdefine:HISTory:EXECute

機能 ヒストリ波形のユーザー定義演算 (Math on History) を実行します。

構文 :MATH<x>:USERdefine:HISTory:EXECute
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :MATH1:USERDEFINE:HISTORY:EXECUTE

:MATH<x>:USERdefine:SCALE?

機能 ユーザー定義演算のスケール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>:USERdefine:SCALE?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:MATH<x>:USERdefine:SCALE: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:SCALE:ARANging
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :MATH1:USERDEFINE:SCALE:ARANGING

:MATH<x>:USERdefine:SCALE:CENTer

機能 ユーザー定義演算のスケール変換の中心値を設定 / 問い合わせします。

構文 :MATH<x>:USERdefine:SCALE:
CENTer {<NRf>}
:MATH<x>:USERdefine:SCALE:CENTer?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = -1.0000E+31 ~ 1.0000E+31

例 :MATH1:USERDEFINE:SCALE:CENTER 1
:MATH1:USERDEFINE:SCALE:CENTER?
-> :MATH1:USERDEFINE:SCALE:
CENTER 1.00000E+00

:MATH<x>:USERdefine:SCALE:SENSitivi ty

機能 ユーザー定義演算のスケール変換の中心からのスパンを設定 / 問い合わせします。

構文 :MATH<x>:USERdefine:SCALE:
SENSitivity {<NRf>}
:MATH<x>:USERdefine:SCALE:
SENSitivity?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = -1.0000E+31 ~ 1.0000E+31

例 :MATH1:USERDEFINE:SCALE:
SENSITIVITY 10
:MATH1:USERDEFINE:SCALE:SENSITIVITY?
-> :MATH1:USERDEFINE:SCALE:
SENSITIVITY 10.0000E+00

5.21 MEASure グループ

ロジック入力が装着されていないモデルでは、ロジック波形に関するコマンドは使用できません。

: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?

:MEASure: {CHANnel<x>|MATH<x>}?

機能 各波形の各パラメータの ON/OFF をすべて問い合わせます。

構文 :MEASure:{CHANnel<x>|MATH<x>}?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:MEASure: {CHANnel<x>|MATH<x>} : {AREA1|AREA2}?

機能 各 Area の各波形の各パラメータの ON/OFF をすべて問い合わせます。

構文 :MEASure:{CHANnel<x>|MATH<x>}:
{AREA1|AREA2}?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:MEASure: {CHANnel<x>|MATH<x>} {[:AREA1]|:AREA2}:ALL

機能 各 Area の各波形のすべての測定アイテムの ON/OFF を一度に設定します。

構文 :MEASure:{CHANnel<x>|MATH<x>}{[:
AREA1]|:AREA2}:ALL <Boolean>
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :MEASURE:CHANNEL1:AREA1:ALL ON

:MEASure: {CHANnel<x>|MATH<x>} {[:AREA1]|:AREA2}:<パラメータ>?

機能 各 Area の各波形の各波形パラメータに関する設定を問い合わせます。

構文 :MEASure:{CHANnel<x>|MATH<x>}{[:
AREA1]|:AREA2}:<パラメータ>?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<パラメータ> = {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth|DElay|DT|DUTYcycle|ENUMber|FALL|FREQuency|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWIDth|PERiod|PNUMber|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEViation|TY1Integ|TY2Integ|V1|V2}

:MEASure: {CHANnel<x>|MATH<x>} {[:AREA1]|:AREA2}:<パラメータ>:COUNT?

機能 各 Area の各波形パラメータの統計処理の回数を問い合わせます。

構文 :MEASure:{CHANnel<x>|MATH<x>}{[:
AREA1]|:AREA2}:<パラメータ>:COUNT?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<パラメータ> = {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth|DElay|DT|DUTYcycle|ENUMber|FALL|FREQuency|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWIDth|PERiod|PNUMber|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEViation|TY1Integ|TY2Integ|V1|V2}

例 以下は、CH1 の平均値についての例です。

```
:MEASURE:CHANNEL1:AREA1:AVERAGE:  
COUNT?  
-> :MEASURE:CHANNEL1:AREA1:AVERAGE:  
COUNT 100
```

:MEASure: {CHANnel<x>|MATH<x>} : AREA2:<パラメータ>: {MAXimum|MEAN|MINimum|SDEViation}?

機能 各 Area の各波形パラメータの各統計値を問い合わせます。

構文 :MEASure:{CHANnel<x>|MATH<x>}{[:
AREA1]|:AREA2}:<パラメータ>:{MAXimum|
MEAN|MINimum|SDEViation}?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<パラメータ> = {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth|DElay|DT|DUTYcycle|ENUMber|FALL|FREQuency|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWIDth|PERiod|PNUMber|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEViation|TY1Integ|TY2Integ|V1|V2}

例 以下は、CH1 の平均値についての例です。

```
:MEASURE:CHANNEL1:AREA1:AVERAGE:  
MAXIMUM?  
-> :MEASURE:CHANNEL1:AREA1:AVERAGE:  
MAXIMUM 10.0000E+03
```

解説 統計値を取得できない場合は、「NAN(Not A Number)」が返されます。

5.21 MEASure グループ

:MEASure: {CHANnel<x>|MATH<x>}

{ [:AREA1] | :AREA2 } :< パラメータ > :STATe

機能 各 Area の各波形パラメータの ON/OFF を 1 つずつ設定 / 問い合わせします。

構文 :MEASure: {CHANnel<x>|MATH<x>} :AREA2: < パラメータ > :STATe {< Boolean >}
:MEASure: {CHANnel<x>|MATH<x>} { [:AREA1] | :AREA2 } :< パラメータ > :STATe?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
< パラメータ > = {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth|DElay|DT|DUTYcycle|ENUMber|FALL|FREQuency|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWIDth|PERiod|PNUMber|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEVIation|TY1Integ|TY2Integ|V1|V2}

例 以下は、CH1 の平均値についての例です。

```
:MEASURE:CHANNEL1:AREA1:AVERAGE:
STATE ON
:MEASURE:CHANNEL1:AREA1:AVERAGE:
STATE?
-> :MEASURE:CHANNEL1:AREA1:AVERAGE:
STATE 1
```

:MEASure: {CHANnel<x>|MATH<x>}

{ [:AREA1] | :AREA2 } :< パラメータ > :VALue?

機能 各 Area の各波形パラメータの自動測定値を問い合わせます。

構文 :MEASure: {CHANnel<x>|MATH<x>} { [:AREA1] | :AREA2 } :< パラメータ > :VALue? [< NRf > [, STATus]]
:MEASure: {CHANnel<x>|MATH<x>} { [:AREA1] | :AREA2 } :< パラメータ > :VALue? [STATus]
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
< パラメータ > = {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth|DElay|DT|DUTYcycle|ENUMber|FALL|FREQuency|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWIDth|PERiod|PNUMber|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEVIation|TY1Integ|TY2Integ|V1|V2}
< NRf > = 1 ~ 100000

例 以下は、CH1 の平均値についての例です。

```
:MEASURE:CHANNEL1:AREA1:AVERAGE:
VALUE?
-> :MEASURE:CHANNEL1:AREA1:AVERAGE:
VALUE 10.0000E+03
```

- 解説
- 測定不可能な場合は、「NAN(Not A Number)」が返されます。
 - < NRf > は、自動測定を実行してから < NRf > 回目の測定値を個々に問い合わせるときに使用します。
 - < NRf > に「1」を設定すると、自動測定値のメモリの中の最も古い測定値を問い合わせます。
 - 設定した回数に測定値が存在しないときは、「NAN(Not A Number)」が返されます。
 - < NRf > を省略すると、最新の測定値を問い合わせます。
 - 自動測定のサイクル統計処理をしている場合、< NRf > を設定すると、表示されている波形で画面の左から < NRf > 回目の 1 周期の測定値を問い合わせず、< NRf > を省略すると、表示されている波形で最後に測定された 1 周期の測定値を問い合わせます。
 - 最後の、省略可能な「STATus」は、パラメータが、「FALL」「NWIDth」「PERiod」「PWIDth」「RISE」の場合に有効です。

これらのパラメータは、測定値が測定分解能以下の場合、「STATus」を付けないうち、測定値にマイナスを付けて返します。「STATus」を付けたとき、「LOW_RESOL」を返します。

**:MEASure: {CHANnel<x>|MATH<x>}
{ [:AREA1] | :AREA2 } :COPY**

機能 各 Area のある波形の全測定アイテムの ON/OFF 設定を他のすべての波形にコピーします。

構文 :MEASure: {CHANnel<x>|MATH<x>} { [:AREA1] | :AREA2 } :COPY
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例 :MEASURE:CHANNEL1:AREA1:COPY

**:MEASure: {CHANnel<x>|MATH<x>}
{ [:AREA1] | :AREA2 } :CYCLE**

機能 各 Area の周期モードを設定 / 問い合わせします。

構文 :MEASure: {CHANnel<x>|MATH<x>} { [:AREA1] | :AREA2 } :CYCLE {ONE|N|OFF}
:MEASure: {CHANnel<x>|MATH<x>} { [:AREA1] | :AREA2 } :CYCLE?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例 :MEASURE:CHANNEL1:AREA1:CYCLE ONE
:MEASURE:CHANNEL1:AREA2:CYCLE?
-> :MEASURE:CHANNEL1:AREA1:CYCLE ONE
解説
・「:ANALysis:PANalyze:TYPE SWLoss」のときは「SWLoss」が問い合わせの応答として返ってきます。
・「:ANALysis:PMEASure<x>:MODE ON」かつ「:ANALysis:PMEASure:CYCMode NCYCLE」のときは「NCChannel」が問い合わせの応答として返ってきます。

**:MEASure: {CHANnel<x>|MATH<x>}
{ [:AREA1] | :AREA2 } :DElay?**

機能 各 Area の各波形のチャンネル間ディレイに関するすべての設定値を問い合わせます。

構文 :MEASure: {CHANnel<x>|MATH<x>} { [:AREA1] | :AREA2 } :DElay?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

**:MEASure: {CHANnel<x>|MATH<x>} :AREA2 :
DElay:MEASure?**

機能 各 Area の各波形のチャンネル間ディレイの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 :MEASure: {CHANnel<x>|MATH<x>} :AREA2 :
DElay:MEASure?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:MEASure: {CHANnel<x>|MATH<x>} { [:AREA1] | :AREA2 } :DElay:MEASure:COUNT

機能 各 Area の各波形のチャンネル間ディレイの対象波形のカウント数を設定 / 問い合わせします。

構文 :MEASure: {CHANnel<x>|MATH<x>} { [:AREA1] | :AREA2 } :DElay:MEASure:COUNT {<Nrf>}
:MEASure: {CHANnel<x>|MATH<x>} { [:AREA1] | :AREA2 } :DElay:MEASure:COUNT?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<Nrf> = 1 ~ 99
例 :MEASURE:CHANNEL1:AREA1:DElay:MEASURE:COUNT 2
:MEASURE:CHANNEL1:AREA1:DElay:MEASURE:COUNT?
-> :MEASURE:CHANNEL1:AREA1:DElay:MEASURE:COUNT 2

:MEASure: {CHANnel<x>|MATH<x>} { [:AREA1] | :AREA2 } :DElay:MEASure:SLOPe

機能 各 Area の各波形のチャンネル間ディレイの対象波形のスロープを設定 / 問い合わせします。

構文 :MEASure: {CHANnel<x>|MATH<x>} { [:AREA1] | :AREA2 } :DElay:MEASure:SLOPe {FALL|RISE}
:MEASure: {CHANnel<x>|MATH<x>} { [:AREA1] | :AREA2 } :DElay:MEASure:SLOPe?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例 :MEASURE:CHANNEL1:AREA1:DElay:MEASURE:SLOPe FALL
:MEASURE:CHANNEL1:AREA1:DElay:MEASURE:SLOPe?
-> :MEASURE:CHANNEL1:AREA1:DElay:MEASURE:SLOPe FALL

:MEASure: {CHANnel<x>|MATH<x>} { [:AREA1] | :AREA2 } :DElay:REfERENCE?

機能 各 Area の各波形のチャンネル間ディレイ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: {CHANnel<x>|MATH<x>} { [:AREA1] | :AREA2 } :DElay:REfERENCE?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

5.21 MEASure グループ

:MEASure: {CHANnel<x>|MATH<x>} {[:AREA1]|:AREA2}:DElay:REFeRence:COUNT

機能 各 Area の各波形のチャネル間ディレイの基準波形のカウント数を設定 / 問い合わせします。

構文 :MEASure:{CHANnel<x>|MATH<x>}{[:AREA1]|:AREA2}:DElay:REFeRence:COUNT {<Nrf>}

:MEASure:{CHANnel<x>|MATH<x>}{[:AREA1]|:AREA2}:DElay:REFeRence:COUNT?

CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

<Nrf> = 1 ~ 99

例 :MEASURE:CHANNEL1:AREA1:DELAY:REFERENCE:COUNT 2

:MEASURE:CHANNEL1:AREA1:DELAY:REFERENCE:COUNT?

-> :MEASURE:CHANNEL1:AREA1:DELAY:REFERENCE:COUNT 2

:MEASure: {CHANnel<x>|MATH<x>} {[:AREA1]|:AREA2}:DElay:REFeRence:SLOPe

機能 各 Area の各波形のチャネル間ディレイの基準波形のスロープを設定 / 問い合わせします。

構文 :MEASure:{CHANnel<x>|MATH<x>}{[:AREA1]|:AREA2}:DElay:REFeRence:SLOPe {FALL|RISE}

:MEASure:{CHANnel<x>|MATH<x>}{[:AREA1]|:AREA2}:DElay:REFeRence:SLOPe?

CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :MEASURE:CHANNEL1:AREA1:DELAY:REFERENCE:SLOPE FALL

:MEASURE:CHANNEL1:AREA1:DELAY:REFERENCE:SLOPE?

-> :MEASURE:CHANNEL1:AREA1:DELAY:REFERENCE:SLOPE FALL

:MEASure: {CHANnel<x>|MATH<x>} {[:AREA1]|:AREA2}:DElay:REFeRence:SOURce

機能 各 Area の各波形のチャネル間ディレイの基準対象をトリガ点にするか、波形にするかを設定 / 問い合わせをします。

構文 :MEASure:{CHANnel<x>|MATH<x>}{[:AREA1]|:AREA2}:DElay:REFeRence:SOURce {TRACe|TRIGger}

:MEASure:{CHANnel<x>|MATH<x>}{[:AREA1]|:AREA2}:DElay:REFeRence:SOURce?

CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :MEASURE:CHANNEL1:AREA1:DELAY:REFERENCE:SOURCE TRACE

:MEASURE:CHANNEL1:AREA1:DELAY:REFERENCE:SOURCE?

-> :MEASURE:CHANNEL1:AREA1:DELAY:REFERENCE:SOURCE TRACE

:MEASure: {CHANnel<x>|MATH<x>} {[:AREA1]|:AREA2}:DElay:REFeRence:TRACe

機能 各 Area の各波形のチャネル間ディレイの基準波形のエッジを設定 / 問い合わせします。

構文 :MEASure:{CHANnel<x>|MATH<x>}{[:AREA1]|:AREA2}:DElay:REFeRence:TRACe {<Nrf>|MATH<y>}

:MEASure:{CHANnel<x>|MATH<x>}{[:AREA1]|:AREA2}:DElay:REFeRence:TRACe?

CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)

MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :MEASURE:CHANNEL1:AREA1:DELAY:REFERENCE:TRACE 2

:MEASURE:CHANNEL1:AREA1:DELAY:REFERENCE:TRACE?

-> :MEASURE:CHANNEL1:AREA1:DELAY:REFERENCE:TRACE 2

:MEASure: {CHANnel<x>|MATH<x>}:DPRoximal?

機能 ディスタル・メシアル・プロキシマルに関するすべての設定値を問い合わせます。

構文 :MEASure:{CHANnel<x>|MATH<x>}:DPRoximal?

CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:MEASure:{CHANnel<x>|MATH<x>}:DPRoximal:MODE

機能 ディスタル・メシアル・プロキシマル点のモードを設定 / 問い合わせします。

構文 :MEASure:{CHANnel<x>|MATH<x>}:
DPRoximal:MODE {PERCent|UNIT}
:MEASure:{CHANnel<x>|MATH<x>}:
DPRoximal:MODE?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :MEASURE:CHANNEL1:DPROXIMAL:
MODE PERCENT
:MEASURE:CHANNEL1:DPROXIMAL:MODE?
-> :MEASURE:CHANNEL1:DPROXIMAL:
MODE PERCENT

:MEASure:{CHANnel<x>|MATH<x>}:DPRoximal:PERCent

機能 ディスタル・メシアル・プロキシマル点を%で設定 / 問い合わせします。

構文 :MEASure:{CHANnel<x>|MATH<x>}:
DPRoximal:PERCent {<NRf>,<NRf>,<NRf>}
:MEASure:{CHANnel<x>|MATH<x>}:
DPRoximal:PERCent?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = 0 ~ 100(%)、(1 ステップ)

例 :MEASURE:CHANNEL1:DPROXIMAL:
PERCENT 80,60,40
:MEASURE:CHANNEL1:DPROXIMAL:PERCENT?
-> :MEASURE:CHANNEL1:DPROXIMAL:
PERCENT 40,60,80

:MEASure:{CHANnel<x>|MATH<x>}:DPRoximal:UNIT

機能 ディスタル・メシアル・プロキシマル点を電圧値で設定 / 問い合わせします。

構文 :MEASure:{CHANnel<x>|MATH<x>}:
DPRoximal:UNIT {<電圧>,<電圧>,<電圧>|<NRf>,<NRf>,<NRf>}
:MEASure:{CHANnel<x>|MATH<x>}:
DPRoximal:UNIT?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

<電圧>、<NRf> = 本体ユーザーズマニュアル機能編参照。

例 :MEASURE:CHANNEL1:DPROXIMAL:
UNIT 50V,0V,-50V
:MEASURE:CHANNEL1:DPROXIMAL:UNIT?
-> :MEASURE:CHANNEL1:DPROXIMAL:
UNIT -50.0E+00,0.0E+00,50.0E+00

解説 電流プローブ設定時は、<電流>値の設定 / 問い合わせとなります。

:MEASure:{CHANnel<x>|MATH<x>}:METHod

機能 High・Low 点を設定 / 問い合わせします。

構文 :MEASure:{CHANnel<x>|MATH<x>}:
METHod {AUTO|MAXMin|HISTogram}
:MEASure:{CHANnel<x>|MATH<x>}:
METHod?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :MEASURE:CHANNEL1:METHOd AUTO
:MEASURE:CHANNEL1:METHOd?
-> :MEASURE:CHANNEL1:METHOd AUTO

:MEASure:CONTinuous?

機能 波形パラメータの自動測定の通常の統計処理に関するすべての設定を問い合わせます。

構文 :MEASure:CONTinuous?

:MEASure:CONTinuous:REStart

機能 波形パラメータの自動測定 of 通常の統計処理を再スタートします。

構文 :MEASure:CONTinuous:REStart

例 :MEASURE:CONTINUOUS:RESTART

5.21 MEASure グループ

:MEASure:CONTinuous:TLCHange (TriggerLevel Change)

機能 トリガレベル変更時に、波形パラメータの自動測定の通常の統計処理を再スタートするか否かを設定 / 問い合わせします。

構文 :MEASure:CONTinuous:TLCHange{REStArt|IGNore}
:MEASure:CONTinuous:TLCHange?
例 :MEASURE:CONTINUOUS:TLCHANGE RESTART
:MEASURE:CONTINUOUS:TLCHANGE?
-> :MEASURE:CONTINUOUS:TLCHANGE RESTART

:MEASure:CYCLE?

機能 自動測定のサイクル統計処理に関するすべての設定値を問い合わせます。

構文 :MEASure:CYCLE?

:MEASure:CYCLE:ABORT

機能 自動測定のサイクル統計処理の実行を中止します。

構文 :MEASure:CYCLE:ABORT

例 :MEASURE:CYCLE:ABORT

:MEASure:CYCLE:EXECute

機能 自動測定のサイクル統計処理を実行します。

構文 :MEASure:CYCLE:EXECute

例 :MEASURE:CYCLE:EXECUTE

:MEASure:CYCLE:TRACe

機能 自動測定のサイクル統計処理の周期対象波形を設定 / 問い合わせします。

構文 :MEASure:CYCLE:TRACe {OWN|<NRf>|
MATH<x>|PODA<y>|PODB<y>|PODC<y>|
PODD<y>}
:MEASure:CYCLE:TRACe?
<NRf> = 1 ~ 8
MATH<x> の <x> = 1 ~ 4
PODA<y>、PODB<y>、PODC<y>、PODD<y>
の <y> = 0 ~ 7
例 :MEASURE:CYCLE:TRACE 1
:MEASURE:CYCLE:TRACE?
-> :MEASURE:CYCLE:TRACE 1

:MEASure:HISTory:ABORT

機能 自動測定のヒストリ波形の統計処理の実行を中止します。

構文 :MEASure:HISTory:ABORT

:MEASURE:HISTORY:ABORT

:MEASure:HISTory:EXECute

機能 自動測定のヒストリ波形の統計処理を実行します。

構文 :MEASure:HISTory:EXECute

例 :MEASURE:HISTORY:EXECUTE

:MEASure:INDicator

機能 測定箇所の表示を設定 / 問い合わせします。

構文 :MEASure:INDicator {<NRf>|MATH<x>|
OFF|PODA<x>|PODB<x>|PODC<x>|
PODD<x>{,<パラメータ>,2}}
:MEASure:INDicator?
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<x>、PODB<x>、PODC<x>、PODD<x>
の <x> = 0 ~ 7
<パラメータ> = {AMPLitude|AVERage|AVGFreq
|AVGPeriod|BWIDth|DElay|DUTYcycle|FALL|FREQ
uency|HIGH|LOW|MAXimum|MINimum|NOVersh
oot|NWIDTH|P|PABS|PERiod|PN|POVershoot|PP|PT
OPeak|PWIDth|RISE|RMS|V1|V2|WH|WHABs|WHN|
WHP|Z}
例 :MEASURE:INDICATOR 1,AVERAGE
:MEASURE:INDICATOR?
-> :MEASURE:INDICATOR 1,AVERAGE

:MEASure:MODE

機能 自動測定の ON/OFF/ 統計処理を設定 / 問い合わせします。

構文 :MEASure:MODE {OFF|ON|CONTinuous|
CYCLE|HISTory}
:MEASure:MODE?

例 :MEASURE:MODE OFF
:MEASURE:MODE?
-> :MEASURE:MODE OFF

:MEASure: { PODA<x> | PODB<x> | PODC<x> | PO DD<x> } ?

機能 ロジック波形の各パラメータの ON/OFF をすべて問い合わせます。

構文 :MEASure:{PODA<x>|PODB<x>|PODC<x>|
PODD<x>}?
<x> = 0 ~ 7

:MEASure: { PODA<x> | PODB<x> | PODC<x> | PO DD<x> } : { AREA1 | AREA2 } ?

機能 各 Area のロジック波形の各パラメータの ON/OFF をすべて問い合わせます。

構文 :MEASure:{PODA<x>|PODB<x>|PODC<x>|
PODD<x>}:{AREA1|AREA2}?
<x> = 0 ~ 7

:MEASure: {PODA<x>|PODB<x>|PODC<x>|PODD<x>} {[:AREA1]|:AREA2}:ALL

機能 各 Area のロジック波形のすべての測定アイテムの ON/OFF を一度に設定します。

構文 :MEASure:{PODA<x>|PODB<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:
ALL {<Boolean>}
<x> = 0 ~ 7

例 :MEASURE:PODA1:AREA1:ALL ON

:MEASure: {PODA<x>|PODB<x>|PODC<x>|PODD<x>} {[:AREA1]|:AREA2}:<パラメータ>?

機能 各 Area のロジック波形パラメータに関する設定を問い合わせます。

構文 :MEASure:{PODA<x>|PODB<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:<パラメータ>?
<x> = 0 ~ 7
<パラメータ> = {AVGFreq|DElay|DUTYcycle|FREQuency|PERiod|PNUMber}

:MEASure: {PODA<x>|PODB<x>|PODC<x>|PODD<x>} {[:AREA1]|:AREA2}:<パラメータ>:COUNT?

機能 各 Area のロジック波形パラメータの統計処理の回数を問い合わせます。

構文 :MEASure:{PODA<x>|PODB<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:<パラメータ>:
COUNT?
<x> = 0 ~ 7
<パラメータ> = {AVGFreq|DElay|DUTYcycle|FREQuency|PERiod|PNUMber}

例 以下は、ビット 1 の平均周波数についての例です。
:MEASURE:PODA1:AREA1:AVGFREQ:COUNT?
-> :MEASURE:PODA1:AREA1:AVGFREQ:
COUNT 100

:MEASure: {PODA<x>|PODB<x>|PODC<x>|PODD<x>} {[:AREA1]|:AREA2}:<パラメータ>:{MAXimum|MEAN|MINimum|SDEviation}?

機能 各 Area の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:{PODA<x>|PODB<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:<パラメータ>:
{MAXimum|MEAN|MINimum|SDEviation}?
<x> = 0 ~ 7
<パラメータ> = {AVGFreq|DElay|DUTYcycle|FREQuency|PERiod|PNUMber}

例 以下は、ビット 1 の平均周波数についての例です。
:MEASURE:PODA1:AREA1:AVGFREQ:
MAXIMUM?
-> :MEASURE:PODA1:AREA1:AVGFREQ:
MAXIMUM 10.0000E+03

解説 統計値がとれない場合は、「NaN(Not A Number)」が返されます。

:MEASure: {PODA<x>|PODB<x>|PODC<x>|PODD<x>} {[:AREA1]|:AREA2}:<パラメータ>:STATE

機能 各 Area のロジック波形パラメータの ON/OFF を 1 つずつ設定 / 問い合わせします。

構文 :MEASure:{PODA<x>|PODB<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:<パラメータ>:
STATE {<Boolean>}
:MEASure:{PODA<x>|PODB<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:<パラメータ>:
STATE?
<x> = 0 ~ 7
<パラメータ> = {AVGFreq|DUTYcycle|FREQuency|PERiod|PNUMber}

例 以下は、ビット 1 の平均周波数についての例です。
:MEASURE:PODA1:AREA1:AVGFREQ:
STATE ON
:MEASURE:PODA1:AREA1:AVGFREQ:STATE?
-> :MEASURE:PODA1:AREA1:AVGFREQ:
STATE 1

:MEASure: {PODA<x>|PODB<x>|PODC<x>|PODD<x>} {[:AREA1]|:AREA2}:<パラメータ>:VALUE?

機能 各 Area の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:{PODA<x>|PODB<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:<パラメータ>:
VALUE? [<Nrf>[,STATUS]]
<x> = 0 ~ 7
<パラメータ> = {AVGFreq|DElay|DUTYcycle|FREQuency|PERiod|PNUMber}
<Nrf> = 1 ~ 100000

例 以下は、ビット 1 の平均周波数についての例です。
:MEASURE:PODA1:AREA1:AVGFREQ:VALUE?
-> :MEASURE:PODA1:AREA1:AVGFREQ:
VALUE 10.0000E+03

解説

- 測定不可能な場合は、「NaN(Not A Number)」が返されます。
- <Nrf> は、自動測定を実行してから <Nrf> 回目の測定値を個々に問い合わせるときに使用します。
- <Nrf> に「1」を設定すると、自動測定値のメモリの中の最も古い測定値を問い合わせます。
- 設定した回数に測定値が存在しないときは、「NaN(Not A Number)」が返されます。
- <Nrf> を省略すると、最新の測定値を問い合わせます。
- 自動測定のサイクル統計処理をしている場合、<Nrf> を設定すると、表示されている波形で画面の左から <Nrf> 回目の 1 周期の測定値を問い合わせます。<Nrf> を省略すると、表示されている波形で最後に測定された 1 周期の測定値を問い合わせます。

5.21 MEASure グループ

:MEASure: { PODA<x> | POdB<x> | PODC<x> | PODD<x> } { [: AREA1] | : AREA2 } : COPY

機能 各 Area のあるロジック波形の全測定アイテムの ON/OFF 設定を他のすべての波形にコピーします。

構文 :MEASure:{PODA<x>|PODb<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:COPY
<x> = 0 ~ 7

例 :MEASURE:PODA1:AREA1:COPY

:MEASure: { PODA<x> | POdB<x> | PODC<x> | PODD<x> } { [: AREA1] | : AREA2 } : DELay?

機能 各 Area のロジック波形のチャネル間ディレイに関するすべての設定値を問い合わせます。

構文 :MEASure:{PODA<x>|PODb<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay?
<x> = 0 ~ 7

:MEASure: { PODA<x> | POdB<x> | PODC<x> | PODD<x> } { [: AREA1] | : AREA2 } : DELay:MEASure?

機能 各 Area のロジック波形のチャネル間ディレイに関するすべての設定値を問い合わせます。

構文 :MEASure:{PODA<x>|PODb<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay:MEASure?
<x> = 0 ~ 7

:MEASure: { PODA<x> | POdB<x> | PODC<x> | PODD<x> } { [: AREA1] | : AREA2 } : DELay:MEASure: COUNT

機能 各 Area のロジック波形のチャネル間ディレイの対象波形のカウント数を設定 / 問い合わせします。

構文 :MEASure:{PODA<x>|PODb<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay:MEASure:COUNt {<NRf>}
:MEASure:{PODA<x>|PODb<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay:MEASure:COUNt?
<x> = 0 ~ 7
<NRf> = 1 ~ 99

例 :MEASURE:PODA1:AREA1:DELAY:MEASURE:COUNT 2
:MEASURE:PODA1:AREA1:DELAY:MEASURE:COUNT?
-> :MEASURE:PODA1:AREA1:DELAY:MEASURE:COUNT 2

:MEASure: { PODA<x> | POdB<x> | PODC<x> | PODD<x> } { [: AREA1] | : AREA2 } : DELay:MEASure: SLOPe

機能 各 Area のロジック波形のチャネル間ディレイの対象波形のスロープを設定 / 問い合わせします。

構文 :MEASure:{PODA<x>|PODb<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay:MEASure:SLOPe {FALL|RISE}
:MEASure:{PODA<x>|PODb<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay:MEASure:SLOPe?
<x> = 0 ~ 7

例 :MEASURE:PODA1:AREA1:DELAY:MEASURE:SLOPE FALL
:MEASURE:PODA1:AREA1:DELAY:MEASURE:SLOPE?
-> :MEASURE:PODA1:AREA1:DELAY:MEASURE:SLOPE FALL

:MEASure: { PODA<x> | POdB<x> | PODC<x> | PODD<x> } { [: AREA1] | : AREA2 } : DELay:REFERENCE?

機能 各 Area のロジック波形のチャネル間ディレイ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:{PODA<x>|PODb<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay:REFERENCE?
<x> = 0 ~ 7

:MEASure: { PODA<x> | POdB<x> | PODC<x> | PODD<x> } { [: AREA1] | : AREA2 } : DELay:REFERENCE: COUNT

機能 各 Area のロジック波形のチャネル間ディレイの基準波形のカウント数を設定 / 問い合わせします。

構文 :MEASure:{PODA<x>|PODb<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay:REFERENCE:COUNt {<NRf>}
:MEASure:{PODA<x>|PODb<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay:REFERENCE:COUNt?
<x> = 0 ~ 7
<NRf> = 1 ~ 99

例 :MEASURE:PODA1:AREA1:DELAY:REFERENCE:COUNT 2
:MEASURE:PODA1:AREA1:DELAY:REFERENCE:COUNT?
-> :MEASURE:PODA1:AREA1:DELAY:REFERENCE:COUNT 2

:MEASure: { PODA<x> | POdB<x> | PODC<x> | PODD<x> } { [:AREA1] | :AREA2 } :DElay: REFere nce: SLOPe

機能 各 Area のロジック波形のチャンネル間ディレイの基準波形のスロープを設定 / 問い合わせします。

構文 :MEASure:{PODA<x>|PODB<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay: REFere nce:SLOPe {FALL|RISE}
:MEASure:{PODA<x>|PODB<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay: REFere nce:SLOPe?

例 :MEASURE:PODA1:AREA1:DELAY:
REFERENCE:SLOPE FALL
:MEASURE:PODA1:AREA1:DELAY:
REFERENCE:SLOPE?
-> :MEASURE:PODA1:AREA1:DELAY:
REFERENCE:SLOPE FALL

:MEASure: { PODA<x> | POdB<x> | PODC<x> | PODD<x> } { [:AREA1] | :AREA2 } :DElay: REFere nce: SOURce

機能 各 Area のロジック波形のチャンネル間ディレイの基準対象をトリガ点にするか、波形にするかを設定 / 問い合わせします。

構文 :MEASure:{PODA<x>|PODB<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay: REFere nce:SOURce {TRACe|TRIGger}
:MEASure:{PODA<x>|PODB<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay: REFere nce:SOURce?

例 :MEASURE:PODA1:AREA1:DELAY:
REFERENCE:SOURCE TRACE
:MEASURE:PODA1:AREA1:DELAY:
REFERENCE:SOURCE?
-> :MEASURE:PODA1:AREA1:DELAY:
REFERENCE:SOURCE TRACE

:MEASure: { PODA<x> | POdB<x> | PODC<x> | PODD<x> } { [:AREA1] | :AREA2 } :DElay: REFere nce: TRACe

機能 各 Area のロジック波形のチャンネル間ディレイの基準波形のエッジを設定 / 問い合わせします。

構文 :MEASure:{PODA<x>|PODB<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay: REFere nce:TRACe {<NRf>|MATH<x>|PODA<x>|PODB<x>|PODC<x>|PODD<x>}
:MEASure:{PODA<x>|PODB<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay: REFere nce:TRACe?

例 :MEASURE:PODA1:AREA1:DELAY:
REFERENCE:TRACE 2
:MEASURE:PODA1:AREA1:DELAY:
REFERENCE:TRACE?
-> :MEASURE:PODA1:AREA1:DELAY:
REFERENCE:TRACE 2

:MEASure: { PODA<x> | POdB<x> | PODC<x> | PODD<x> } { [:AREA1] | :AREA2 } :DElay: STATe

機能 各 Area のロジック波形のチャンネル間ディレイの ON/OFF を設定 / 問い合わせします。

構文 :MEASure:{PODA<x>|PODB<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay:STATe {<Boolean>}
:MEASure:{PODA<x>|PODB<x>|PODC<x>|PODD<x>}{[:AREA1]|:AREA2}:DElay:STATe?

例 :MEASURE:PODA1:AREA1:DELAY:STATE ON
:MEASURE:PODA1:AREA1:DELAY:STATE?
-> :MEASURE:PODA1:AREA1:DELAY:STATE 1

:MEASure: RANGe<x>

機能 測定対象ウィンドウを設定 / 問い合わせします。

構文 :MEASure:RANGe<x> {MAIN|Z1|Z2}
:MEASure:RANGe<x>?
<x> = 1 ~ 2

例 :MEASURE:RANGE1 MAIN
:MEASURE:RANGE1?
-> :MEASURE:RANGE1 MAIN

解説 「:Measure:RANGE1」は通常測定時および 2 領域測定時の Area1 の測定対象ウィンドウ、
「:Measure:RANGE1」は 2 領域測定時の Area2 の測定対象ウィンドウを設定 / 問い合わせします。

5.21 MEASure グループ

: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)、(10div/ 表示レコード長) ステップ
例	:MEASURE:TRANGE1 -4,4 :MEASURE:TRANGE1? -> :MEASURE:TRANGE1 -4.00E+00, 4.00E+00
解説	「:Measure:TRANGE1」は通常測定時および 2 領域測定時の Area1 の測定範囲、「:Measure:TRANGE2」は 2 領域測定時の Area2 の測定範囲を設定 / 問い合わせします。

:MEASure:USER<x>?

機能	Calc アイテ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	:MEASure:USER<x>? <x> = 1 ~ 4

:MEASure:USER<x>:COUNT?

機能	Calc アイテムの自動測定値の統計処理の回数を問い合わせます。
構文	:MEASure:USER<x>:COUNT? <x> = 1 ~ 4
例	:MEASURE:USER1:COUNT? -> :MEASURE:USER1:COUNT 100

:MEASure:USER<x>:DEFINE

機能	Calc アイテムの自動測定値の演算式を設定 / 問い合わせします。
構文	:MEASure:USER<x>:DEFINE {<文字列>} :MEASure:USER<x>:DEFINE? <x> = 1 ~ 4 <文字列> = 128 文字以内
例	:MEASURE:USER1:DEFINE "MAX(C1)" :MEASURE:USER1:DEFINE? -> :MEASURE:USER1:DEFINE "MAX(C1)"

:MEASure:USER<x>:{MAXimum|MEAN|MINimum|SDEviation}?

機能	Calc アイテムの自動測定値の各統計値を問い合わせます。
構文	:MEASure:USER<x>:MAXimum? <x> = 1 ~ 4
例	以下は、最大値についての例です。 :MEASURE:USER1:MAXIMUM? -> :MEASURE:USER1:MAXIMUM 1.000E+00
解説	統計値を取得できない場合は、「NAN(Not A Number)」が返されます。

:MEASure:USER<x>:NAME

機能	Calc アイテムの名称を設定 / 問い合わせします。
構文	:MEASure:USER<x>:NAME {<文字列>} :MEASure:USER<x>:NAME? <x> = 1 ~ 4 <文字列> = 8 文字以内
例	:MEASURE:USER1:NAME "MAX" :MEASURE:USER1:NAME? -> :MEASURE:USER1:NAME "MAX"

:MEASure:USER<x>:STATE

機能	Calc アイテムの自動測定の ON/OFF を設定 / 問い合わせします。
構文	:MEASure:USER<x>:STATE {<Boolean>} :MEASure:USER<x>:STATE? <x> = 1 ~ 4
例	:MEASURE:USER1:STATE ON :MEASURE:USER1:STATE? -> :MEASURE:USER1:STATE 1

:MEASure:USER<x>:UNIT

機能	Calc アイテムの単位を設定 / 問い合わせします。
構文	:MEASure:USER<x>:UNIT {<文字列>} :MEASure:USER<x>:UNIT? <x> = 1 ~ 4 <文字列> = 4 文字以内
例	:MEASURE:USER1:UNIT "V" :MEASURE:USER1:UNIT? -> :MEASURE:USER1:UNIT "V"
解説	単位によって測定値に影響をおよぼすことはありません。

:MEASure:USER<x>:VALue?

機能	Calc アイテムの自動測定値の測定値を問い合わせます。
構文	:MEASure:USER<x>:VALue? [<NRf>] <x> = 1 ~ 4 <NRf> = 1 ~ 100000
例	:MEASURE:USER1:VALUE? -> :MEASURE:USER1: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、自動測定を実行してから <NRf> 回目の測定値を個々に問い合わせるときに使用します。 ・ <NRf> に「1」を設定すると、自動測定値のメモリの最古の測定値を問い合わせます。 ・ 設定した回数に測定値が存在しないときは、「NAN(Not A Number)」が返されます。 ・ <NRf> を省略すると、最新の測定値を問い合わせます。

: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 ~ 36000(タイムアウト時間、100ms 単位)

例 :MEASURE:WAIT? 1
-> :MEASURE:WAIT 0

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:ZLINKage

機能 波形パラメータの自動測定の測定結果の測定番号とズーム位置との連動 (ON/OFF) を設定 / 問い合わせします。

構文 :MEASure:ZLINKage {<Boolean>}
:MEASure:ZLINKage?

例 :MEASURE:ZLINKAGE ON
:MEASURE:ZLINKAGE?
-> :MEASURE:ZLINKAGE 1

5.22 RECall グループ

:RECall:SETup<x>:EXECute

機能 内部メモリに保存されている設定データを呼び出します。

構文 :RECall:SETup<x>:EXECute
<x> = 1 ~ 3

例 :RECALL:SETUP1:EXECUTE

解説 保存されていない番号を指定して実行するとエラーになります。

5.23 REference グループ

:REference<x>?

機能 リファレンス波形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 ~ 8(4ch モデルでは 1 ~ 4)

:REference<x>:DISPlay

機能 リファレンス波形の表示の ON/OFF を設定 / 問い合わせします。

構文 :REference<x>:DISPlay {<Boolean>}
:REference<x>:DISPlay?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :REFERENCE1:DISPLAY ON
:REFERENCE1:DISPLAY?
-> :REFERENCE1:DISPLAY 1

:REference<x>:LABel?

機能 各リファレンス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 :REference<x>:LABel
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:REference<x>:LABel[:DEFine]

機能 各リファレンスのラベル名を設定 / 問い合わせします。

構文 :REference<x>:LABel[:DEFine] {<文字列>}
:REference<x>:LABel[:DEFine]?
<x> = 1 ~ 8(4ch モデルでは 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 ~ 8(4ch モデルでは 1 ~ 4)

例 :REFERENCE1:LABEL:MODE ON
:REFERENCE1:LABEL:MODE?
-> :REFERENCE1:LABEL:MODE 1

:REference<x>:LOAD

機能 リファレンス波形をロードします。

構文 :REference<x>:LOAD {<Nrf>|MATH<y>}
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<y> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :REFERENCE1:LOAD 1
解説 <x> と <y> の設定範囲は次のとおりです。

<x>	<y> の設定範囲
1	2 ~ 4
2	1、3、4
3	1、2、4
4	1 ~ 3
5	6 ~ 8
6	5、7、8
7	5、6、8
8	5 ~ 7

:REference<x>:POSition

機能 リファレンス波形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 ~ 8(4ch モデルでは 1 ~ 4)
<Nrf> = -4 ~ 4(div)

例 :REFERENCE1:POSITION 1
:REFERENCE1:POSITION?
-> :REFERENCE1:POSITION 1.000E+00

5.24 SEARch グループ

検索のしかたに関する通信コマンドと、操作キーを押して表示される検索タイプの対応を以下に示します。

検索の種類	通信コマンド	設定メニュー	
		操作キー	検索タイプ
エッジ	EDGE	SEARCH キー	Edge
複数入力 of の組み合わせ (パターン)	PATtern		Pattern
パルス幅	PULSe		Pulse Width
タイムアウト	TI梅out		Timeout

: SEARch?

機能 波形の検索に関するすべての設定値を問い合わせます。

構文 :SEARch?

: SEARch: ABORt

機能 検索を中止します。

構文 :SEARch:ABORt

例 :SEARCH:ABORT

: SEARch: ASCRoll<x>?

機能 オートスクロールに関するすべての設定値を問い合わせます。

構文 :SEARch:ASCROLL<x>?

<x> = 1 ~ 2

: SEARch: ASCRoll<x>: 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 :SEARch:ASCROLL<x>:JUMP {LEFT|RIGHT}

<x> = 1 ~ 2

例 :SEARCH:ASCROLL1:JUMP LEFT

: SEARch: ASCRoll<x>: SPEEd

機能 オートスクロールのズームボックスの移動速度を設定 / 問い合わせします。

構文 :SEARch:ASCROLL<x>:SPEEd {<NRf>}

:SEARch:ASCROLL<x>:SPEEd?

<x> = 1 ~ 2

<NRf> = 1 ~ 50

例 :SEARCH:ASCROLL1:SPEED 1

:SEARCH:ASCROLL1:SPEED?

-> :SEARCH:ASCROLL1:SPEED 1

: SEARch: ASCRoll<x>: START

機能 オートスクロールを開始します。

構文 :SEARch:ASCROLL<x>:START {LEFT|RIGHT}

<x> = 1 ~ 2

例 :SEARCH:ASCROLL1:START LEFT

: SEARch: ASCRoll<x>: STOP

機能 オートスクロールを停止します。

構文 :SEARch:ASCROLL<x>:STOP

<x> = 1 ~ 2

例 :SEARCH:ASCROLL1:STOP

: SEARch: EDGE?

機能 エッジ検索に関するすべての設定値を問い合わせます。

構文 :SEARch:EDGE?

: SEARch: EDGE: HYSTeresis

機能 エッジ検索レベルのヒステリシスを設定 / 問い合わせします。

構文 :SEARch:EDGE:HYSTeresis {<NRf>}

:SEARch:EDGE:HYSTeresis?

<NRf> = 0.3(div) ~ 8(div)、0.1div ステップ

例 :SEARCH:EDGE:HYSTERESIS 1

:SEARCH:EDGE:HYSTERESIS?

-> :SEARCH:EDGE:HYSTERESIS 1.0

: SEARch: EDGE: LEVel

機能 エッジ検索レベルを設定 / 問い合わせします。

構文 :SEARch:EDGE:LEVel {<NRf>|<電圧>}

:SEARch:EDGE:LEVel?

<NRf> = 本体ユーザーズマニュアル機能編参照。

例 :SEARCH:EDGE:LEVEL -10mV

:SEARCH:EDGE:LEVEL?

-> :SEARCH:EDGE:LEVEL -10.00E-03

解説 電流プローブ設定時は、<電流> 値の設定 / 問い合わせとなります。

: SEARch: EDGE: SLOPe

機能 エッジ検索のスロープを設定 / 問い合わせします。

構文 :SEARch:EDGE:SLOPe {BOTH|FALL|RISE}

:SEARch:EDGE:SLOPe?

例 :SEARCH:EDGE:SLOPE RISE

:SEARCH:EDGE:SLOPE?

-> :SEARCH:EDGE:SLOPE RISE

:SEARCh:EDGE:SOURce

機能	エッジ検索の対象波形を設定 / 問い合わせします。
構文	:SEARCh:EDGE:SOURce {<NRf> MATH<x> PODA<x> PODB<x> PODC<x> PODD<x>} :SEARCh:EDGE:SOURce? <NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4) MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) PODA<x> PODB<x> PODC<x> PODD<x> の <x> = 0 ~ 7
例	:SEARCH:EDGE:SOURCE 1 :SEARCH:EDGE:SOURCE? -> :SEARCH:EDGE:SOURCE 1

:SEARCh:EPOint

機能	検索終了位置を設定 / 問い合わせします。
構文	:SEARCh:EPOint {<NRf>} :SEARCh:EPOint? <NRf> = -5 ~ 5(div)、(10div/ 表示レコード長) ステップ
例	:SEARCH:EPOINT 0 :SEARCH:EPOINT? -> :SEARCH:EPOINT 0.000E+00

:SEARCh:EXECute

機能	検索を実行します。オーバーラップコマンドです。
構文	:SEARCh:EXECute
例	:SEARCH:EXECUTE

:SEARCh:MARK

機能	検索点マークの ON/OFF を設定 / 問い合わせします。
構文	:SEARCh:MARK {<Boolean>} :SEARCh:MARK?
例	:SEARCH:MARK ON :SEARCH:MARK? -> :SEARCH:MARK 1

:SEARCh:MAG<x>

機能	ズーム画面のズーム率を設定 / 問い合わせします。
構文	:SEARCh:MAG<x> {<NRf>} :SEARCh:MAG<x>? <x> = 1 ~ 2 <NRf> = 本体ユーザーズマニュアル機能編参照。
例	:SEARCH:MAG1 2 :SEARCH:MAG1? -> :SEARCH:MAG1 2

:SEARCh:PATtern?

機能	複数入力の組み合わせ (パターン) 検索に関するすべての設定値を問い合わせます。
構文	:SEARCh:PATtern?

:SEARCh:PATtern:{CHANnel<x>|MATH<x>}?

機能	複数入力の組み合わせ (パターン) 検索の各波形に関するすべての設定値を問い合わせます。
構文	:SEARCh:PATtern:{CHANnel<x> MATH<x>}? CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:SEARCh:PATTtern:{CHANnel<x>|MATH<x>} :HYSTeresis

機能	複数入力の組み合わせ (パターン) 検索の対象波形のヒステリシスを設定 / 問い合わせします。
構文	:SEARCh:PATTtern:{CHANnel<x> MATH<x>} :HYSTeresis {<NRf>} :SEARCh:PATTtern:{CHANnel<x> MATH<x>} :HYSTeresis? CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <NRf> = 0.3(div) ~ 8.0(div)、0.1div ステップ
例	:SEARCH:PATTERN:CHANNEL1: HYSTERESIS 0.3 :SEARCH:PATTERN:CHANNEL1:HYSTERESIS? -> :SEARCH:PATTERN:CHANNEL1: HYSTERESIS 0.30

:SEARCh:PATtern:{CHANnel<x>|MATH<x>} :LEVel

機能	複数入力の組み合わせ (パターン) 検索の対象波形の検索レベルを設定 / 問い合わせします。
構文	:SEARCh:PATtern:{CHANnel<x> MATH<x>} :LEVel {<NRf> <電圧>} :SEARCh:PATtern:{CHANnel<x> MATH<x>} :LEVel? CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) <NRf> = 本体ユーザーズマニュアル機能編参照。
例	:SEARCH:PATTERN:CHANNEL1:LEVEL 1V :SEARCH:PATTERN:CHANNEL1:LEVEL? -> :SEARCH:PATTERN:CHANNEL1: LEVEL 1.000E+00
解説	電流プローブ設定時は、<電流> 値の設定 / 問い合わせとなります。

5.24 SEARCh グループ

:SEARCh:PATtern:{CHANnel<x>|MATH<x>} :PATtern

機能	複数入力の組み合わせ (パターン) 検索の対象波形の検索パターンを設定 / 問い合わせします
構文	:SEARCh:PATtern:{CHANnel<x> MATH<x>}:PATtern {DONTcare HIGH LOW} :SEARCh:PATtern:{CHANnel<x> MATH<x>}:PATtern? CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例	:SEARCH:PATTERN:CHANNEL1: PATTERN HIGH :SEARCH:PATTERN:CHANNEL1:PATTERN? -> :SEARCH:PATTERN:CHANNEL1: PATTERN HIGH

:SEARCh:PATtern:CLOCK?

機能	複数入力の組み合わせ (パターン) 検索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	:SEARCh:PATtern:CLOCK?

:SEARCh:PATtern:CLOCK:HYSTeresis

機能	複数入力の組み合わせ (パターン) 検索のクロック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	:SEARCh:PATtern:CLOCK: HYSTeresis {<Nrf>} :SEARCh:PATtern:CLOCK:HYSTeresis? <Nrf> = 0.3(div) ~ 8.0(div)、0.1div ステップ
例	:SEARCH:PATTERN:CLOCK:HYSTERESIS 4.0 :SEARCH:PATTERN:CLOCK:HYSTERESIS? -> :SEARCH:PATTERN:CLOCK: HYSTERESIS 4.00
解説	「:SEARCh:PATtern:CLOCK:SOURce」が「NONE」のときは、エラーになります。

:SEARCh:PATtern:CLOCK:LEVel

機能	複数入力の組み合わせ (パターン) 検索のクロックチャンネルの検索レベルを設定 / 問い合わせします。
構文	:SEARCh:PATtern:CLOCK:LEVel {<Nrf> <電圧>} :SEARCh:PATtern:CLOCK:LEVel? <Nrf> = 本体ユーザーマニュアル機能編参照。
例	:SEARCH:PATTERN:CLOCK:LEVEL :SEARCH:PATTERN:CLOCK:LEVEL? -> :SEARCH:PATTERN:CLOCK:LEVEL
解説	・「:SEARCh:PATtern:CLOCK:SOURce」が「NONE」のときは、エラーになります。 ・電流プローブ設定時は、<電流> 値の設定 / 問い合わせとなります。

:SEARCh:PATtern:CLOCK:SLOPe

機能	複数入力の組み合わせ (パターン) 検索のクロックチャンネルのスロープを設定 / 問い合わせします。
構文	:SEARCh:PATtern:CLOCK:SLOPe {FALL RISE} :SEARCh:PATtern:CLOCK:SLOPe? 例 :SEARCH:PATTERN:CLOCK:SLOPE RISE :SEARCH:PATTERN:CLOCK:SLOPE? -> :SEARCH:PATTERN:CLOCK:SLOPE RISE
解説	「:SEARCh:PATtern:CLOCK:SOURce」が「NONE」のときは、エラーになります。

:SEARCh:PATtern:CLOCK:SOURce

機能	複数入力の組み合わせ (パターン) 検索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	:SEARCh:PATtern:CLOCK:SOURce {<Nrf> NONE MATH<x> PODA<x> PODB<x> PODC<x> PODD<x>} :SEARCh:PATtern:CLOCK:SOURce? CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4) PODA<x>、PODB<x>、PODC<x>、PODD<x> の <x> = 0 ~ 7 例 :SEARCH:PATTERN:CLOCK:SOURCE 1 :SEARCH:PATTERN:CLOCK:SOURCE? -> :SEARCH:PATTERN:CLOCK:SOURCE 1

:SEARCh:PATtern:CONDition

機能	複数入力の組み合わせ (パターン) 検索の成立条件を設定 / 問い合わせします。
構文	:SEARCh:PATtern:CONDition {ENTER EXIT FALSe TRUE} :SEARCh:PATtern:CONDition? 例 :SEARCH:PATTERN:CONDITION ENTER :SEARCH:PATTERN:CONDITION? -> :SEARCH:PATTERN:CONDITION ENTER

:SEARCh:PATtern:LOGic

機能	複数入力の組み合わせ (パターン) 検索のロジックを設定 / 問い合わせします。
構文	:SEARCh:PATtern:LOGic {AND OR} :SEARCh:PATtern:LOGic? 例 :SEARCH:PATTERN:LOGIC AND :SEARCH:PATTERN:LOGIC? -> :SEARCH:PATTERN:LOGIC AND

:SEARCh:PATtern:TIME<x>

機能 複数入力の組み合わせ (パターン) 検索の判定時間を設定 / 問い合わせします。

構文 :SEARCh:PATtern:TIME<x> {<時間>}
:SEARCh:PATtern:TIME<x>?
<x> = 1 ~ 2
<時間> = 1ns ~ 10s

例 :SEARCH:PATTERN:TIME1 1S
:SEARCH:PATTERN:TIME1?
-> :SEARCH:PATTERN:TIME1 1.000E+00

:SEARCh:PATtern:TQQualify

機能 複数入力の組み合わせ (パターン) の時間条件を設定 / 問い合わせします。

構文 :SEARCh:PATtern:TQQualify {INSide|LTHan|MTHan|OUTSide|TImeout}
:SEARCh:PATtern:TQQualify?

例 :SEARCH:PATTERN:TQUALIFY INSIDE
:SEARCH:PATTERN:TQUALIFY?
-> :SEARCH:PATTERN:TQUALIFY INSIDE

:SEARCh:PATtern:{PODA|PODB|PODC|PODD}? :HEXa

機能 複数入力の組み合わせ (パターン) の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 :SEARCh:PATtern:{PODA|PODB|PODC|PODD}?

:SEARCh:PATtern:{PODA|PODB|PODC|PODD}:HEXa

機能 複数入力の組み合わせ (パターン) のロジックの成立条件を HEXA で設定します。

構文 :SEARCh:PATtern:{PODA|PODB|PODC|PODD}:HEXa {<文字列>}

例 :SEARCH:PATTERN:PODA:HEXa "AB"

:SEARCh:PATtern:{PODA|PODB|PODC|PODD}:PATtern

機能 複数入力の組み合わせ (パターン) のロジックの成立条件を BINARY で設定 / 問い合わせします。

構文 :SEARCh:PATtern:{PODA|PODB|PODC|PODD}:PATtern {<文字列>}
:SEARCh:PATtern:{PODA|PODB|PODC|PODD}:PATtern?

例 :SEARCH:PATTERN:PODA:
PATTERN "1100xx01"
:SEARCH:PATTERN:PODA:PATTERN?
-> :SEARCH:PATTERN:PODA:
PATTERN "1100xx01"

:SEARCh:POSition<x>

機能 ズームボックスの位置を設定 / 問い合わせします。

構文 :SEARCh:POSition<x> {<Nrf>}
:SEARCh:POSition<x>?
<x> = 1 ~ 2
<Nrf> = -5 ~ 5(div)、(10div/ 表示レコード長) ステップ

例 :SEARCH:POSITION1 1
:SEARCH:POSITION1?
-> :SEARCH:POSITION1 1.000E+00

:SEARCh:PULSe?

機能 パルス幅検索に関するすべての設定値を問い合わせます。

構文 :SEARCh:PULSe?

:SEARCh:PULSe:HYSTeresis

機能 パルス幅検索レベルのヒステリシスを設定 / 問い合わせします。

構文 :SEARCh:PULSe:HYSTeresis {<Nrf>}
:SEARCh:PULSe:HYSTeresis?
<Nrf> = 0.3(div) ~ 8.0(div)、0.1div ステップ

例 :SEARCH:PULSE:HYSTERESIS 1
:SEARCH:PULSE:HYSTERESIS?
-> :SEARCH:PULSE:HYSTERESIS 1.0

5.24 SEARCH グループ

:SEARCH:PULSe:LEVel

機能	パルス幅検索レベルを設定 / 問い合わせします。
構文	:SEARCH:PULSe:LEVel {<NRf> <電圧>} :SEARCH:PULSe:LEVel? <NRf> = 本体ユーザーズマニュアル機能編参照。
例	:SEARCH:PULSE:LEVEL 1V :SEARCH:PULSE:LEVEL? -> :SEARCH:PULSE:LEVEL 1.000E+00
解説	電流プローブ設定時は、<電流>値の設定 / 問い合わせとなります。

:SEARCH:PULSe:POLarity

機能	パルス幅検索レベルの極性を設定 / 問い合わせします。
構文	:SEARCH:PULSe:POLarity {HIGH LOW} :SEARCH:PULSe:POLarity?
例	:SEARCH:PULSE:POLARITY HIGH :SEARCH:PULSE:POLARITY? -> :SEARCH:PULSE:POLARITY HIGH

:SEARCH:PULSe:SOURce

機能	パルス幅検索の対象波形を設定 / 問い合わせします。
構文	:SEARCH:PULSe:SOURce {<NRf> MATH<x> NONE PODA<x> PODB<x> PODC<x> PODD<x>} :SEARCH:PULSe:SOURce? CHANnel<x>の<x> = 1 ~ 8(4ch モデルでは 1 ~ 4) MATH<x>の<x> = 1 ~ 8(4ch モデルでは 1 ~ 4) PODA<x>、PODB<x>、PODC<x>、PODD<x> の<x> = 0 ~ 7
例	:SEARCH:PULSE:SOURCE 1 :SEARCH:PULSE:SOURCE? -> :SEARCH:PULSE:SOURCE 1

:SEARCH:PULSe:TIME<x>

機能	パルス幅検索のパルス幅を設定 / 問い合わせします。
構文	:SEARCH:PULSe:TIME<x> {<時間>} :SEARCH:PULSe:TIME<x>? <x> = 1 ~ 2 <時間> = 1ns ~ 10s
例	:SEARCH:PULSE:TIME1 1S :SEARCH:PULSE:TIME1? -> :SEARCH:PULSE:TIME1 1.000E+00

:SEARCH:PULSe:TQQualify

機能	パルス幅検索の検索種類を設定 / 問い合わせします。
構文	:SEARCH:PULSe:TQQualify {INSide LTHan MTHan OUTSide TIMEout} :SEARCH:PULSe:TQQualify?
例	:SEARCH:PULSE:TQUALIFY INSIDE :SEARCH:PULSE:TQUALIFY? -> :SEARCH:PULSE:TQUALIFY INSIDE

:SEARCH:SElect?

機能	ズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。
構文	:SEARCH:SElect {<NRf> MAXimum} :SEARCH:SElect? <NRf> = 0 ~ 49999
例	:SEARCH:SELECT 1 :SEARCH:SELECT? -> :SEARCH:SELECT 1.50000000
解説	検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:SEARCH:SKIP?

機能	スキップモードに関するすべての設定値を問い合わせます。
構文	:SEARCH:SKIP?

:SEARCH:SKIP:DECimation

機能	スキップモードの間引き検出の設定 / 問い合わせます。
構文	:SEARCH:SKIP:DECimation {<NRf>} :SEARCH:SKIP:DECimation? <NRf> = 1 ~ 9999
例	:SEARCH:SKIP:DECIMATION 1 :SEARCH:SKIP:DECIMATION? -> :SEARCH:SKIP:DECIMATION 1

:SEARCH:SKIP:HOLDoff

機能	ホールドオフ検出の設定 / 問い合わせます。
構文	:SEARCH:SKIP:HOLDoff {<時間>} :SEARCH:SKIP:HOLDoff? <時間> = 0.1ns ~ 1s
例	:SEARCH:SKIP:HOLDOFF 0S :SEARCH:SKIP:HOLDOFF? -> :SEARCH:SKIP:HOLDOFF 0.000E+00

:SEARCh:SKIP:MODE

機能 スキップモードのモードの設定 / 問い合わせま
す。
構文 :SEARCh:SKIP:MODE {DECimation|
HOLDoff|OFF}
:SEARCh:SKIP:MODE?
例 :SEARCH:SKIP:MODE DECIMATION
:SEARCH:SKIP:MODE?
-> :SEARCH:SKIP:MODE DECIMATION

:SEARCh:SPOint

機能 検索開始位置を設定 / 問い合わせします。
構文 :SEARCh:SPOint {<NRf>}
:SEARCh:SPOint?
<NRf> = -5 ~ 5(div)、(10div/ 表示レコード長)
ステップ
例 :SEARCH:SPOINT -1
:SEARCH:SPOINT?
-> :SEARCH:SPOINT -1.000E+00

:SEARCh:TIMEout?

機能 タイムアウト検索のすべての設定値を問い合
わせます。
構文 :SEARCh:TIMEout?

:SEARCh:TIMEout:POLarity

機能 タイムアウト検索の極性を設定 / 問い合わせし
ます。
構文 :SEARCh:TIMEout:POLarity {NEGative|
POSitive}
:SEARCh:TIMEout:POLarity?
例 :SEARCH:TIMEOUT:POLARITY NEGATIVE
:SEARCH:TIMEOUT:POLARITY?
-> :SEARCH:TIMEOUT:POLARITY NEGATIVE

:SEARCh:TIMEout:SOURce

機能 タイムアウト検索の対象波形を設定 / 問い合わ
せします。
構文 :SEARCh:TIMEout:SOURce {<NRf>|
MATH<x>|PODA<x>|PODB<x>|PODC<x>|
PODD<x>}
:SEARCh:TIMEout:SOURce?
CHANnel<x> の <x> = 1 ~ 8(4ch モデルでは 1
~ 4)
MATH<x> の <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<x>、PODB<x>、PODC<x>、PODD<x>
の <x> = 0 ~ 7
例 :SEARCH:TIMEOUT:SOURCE 1
:SEARCH:TIMEOUT:SOURCE?
-> :SEARCH:TIMEOUT:SOURCE 1

:SEARCh:TIMEout:TIME

機能 タイムアウト検索のタイムアウト時間を設定 / 問
い合わせします。
構文 :SEARCh:TIMEout:TIME <時間>
:SEARCh:TIMEout:TIME?
<時間> = 1ns ~ 10s
例 :SEARCH:TIMEOUT:TIME 1
:SEARCH:TIMEOUT:TIME?
-> :SEARCH:TIMEOUT:TIME 1

:SEARCh:TWINDOW

機能 検索された部分を表示するズーム画面を設定 / 問
い合わせします。
構文 :SEARCh:TWINDOW {Z1|Z2}
:SEARCh:TWINDOW?
例 :SEARCH:TWINDOW Z1
:SEARCH:TWINDOW?
-> :SEARCH:TWINDOW Z1

:SEARCh:TYPE

機能 検索タイプを設定 / 問い合わせします。
構文 :SEARCh:TYPE {EDGE|PATtern|PULSe|
TIMEout}
:SEARCh:TYPE?
例 :SEARCH:TYPE EDGE
:SEARCH:TYPE?
-> :SEARCH:TYPE EDGE

5.25 SERIALbus グループ

シリアルバスの種類に関する通信コマンドと、操作キーを押して表示されるシリアルバスタイプの対応を以下に示します。

シリアルバスの種類	通信コマンド	設定メニュー 操作キー	シリアルバスタイプ
CAN	CAN*	SHIFT+SEARCH(SERIAL BUS) キー	CAN
CAN FD	CANFD*		CAN FD
CXPI	CXPI*		CXPI
FlexRay	FLEXray*		FlexRay
I2C	I2C*		I2C
LIN	LIN*		LIN
PSI5	PSI5*		PSI5 Airbag
SENT	SENT*		SENT
ユーザー定義	SPATtern		User Define
SPI	SPI*		SPI
UART	UART*		UART

* CAN、CANFD、CXPI、FLEXray、I2C、LIN、PSI5、SENT、SPI、UART はシリアルバスオプションです。シリアルバスオプション付でないモデルでは、これらのシリアルバスに関するコマンドは使用できません。

:SERIALbus<x>?

機能 シリアルバス信号の解析 / 検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>?
<x> = 1 ~ 4

:SERIALbus<x>:ASETup?

機能 シリアルバス信号のオート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 :SERIALbus<x>:ASETup?
<x> = 1 ~ 4

:SERIALbus<x>:ASETup:ABOrt

機能 シリアルバス信号のオートセットアップを中止します。

構文 :SERIALbus<x>:ASETup:ABOrt
<x> = 1 ~ 4

例 :SERIALBUS1:ASETUP:ABORT

:SERIALbus<x>:ASETup:EXECute

機能 シリアルバス信号のオートセットアップを実行します。

構文 :SERIALbus<x>:ASETup:EXECute
<x> = 1 ~ 4

例 :SERIALBUS1:ASETUP:EXECUTE

:SERIALbus<x>:ASETup:SWAVEform

機能 シリアルバス信号のオートセットアップの対象波形を設定 / 問い合わせします。

構文 :SERIALbus<x>:ASETup:SWAVEform {KEEP|UPDate}
:SERIALbus<x>:ASETup:SWAVEform?
<x> = 1 ~ 4

例 :SERIALBUS1:ASETUP:SWAVEFORM KEEP
:SERIALBUS1:ASETUP:SWAVEFORM?

-> :SERIALBUS1:ASETUP:SWAVEFORM KEEP

- 解説
- SERIALbus1 ~ SERIALbus4 で共通のコマンドです。
 - 「KEEP」の場合、画面に表示している波形が対象です。
 - 「UPDate」の場合、新たに取得する波形が対象です。

:SERIALbus<x>:ASETup:TSETting

機能 シリアルバス信号のオートセットアップのトリガ設定の有無を設定 / 問い合わせします。

構文 :SERIALbus<x>:ASETup:
TSETting {<boolean>}
:SERIALbus<x>:ASETup:TSETting?
<x> = 1 ~ 4

例 :SERIALBUS1:ASETUP:TSETTING ON
:SERIALBUS1:ASETUP:TSETTING?

-> :SERIALBUS1:ASETUP:TSETTING 1

解説 SERIALbus1 ~ SERIALbus4 で共通のコマンドです。

:SERIALbus<x>:CAN?

機能 CAN バス信号に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CAN?
<x> = 1 ~ 4

:SERIALbus<x>:CAN:ANALyze?

機能 CAN バス信号解析に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CAN:ANALyze?
<x> = 1 ~ 4

:SERIALbus<x>:CAN[:ANALyze]:SETup?

機能 CAN バス信号解析のバス設定に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CAN[:ANALyze]:SETup?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:ANALYZE:SETUP?
-> :SERIALBUS1:CAN:ANALYZE:SETUP

:SERIALbus<x>:CAN[:ANALyze]:SETup:BRATe

機能 CAN バス信号解析のビットレート (データ転送速度) を設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN[:ANALyze]:SETup:
BRATe {<Nrf>|USER,<Nrf>}
:SERIALbus<x>:CAN[:ANALyze]:SETup:
BRATe?
<x> = 1 ~ 4
<Nrf> = 33300、83300、125000、250000、
500000、1000000
USER の <Nrf> = 本体ユーザーズマニュアル機能
編参照。

例 :SERIALBUS1:CAN:ANALYZE:SETUP:
BRATE 83300
:SERIALBUS1:CAN:ANALYZE:SETUP:BRATE?
-> :SERIALBUS1:CAN:ANALYZE:SETUP:
BRATE 83300

:SERIALbus<x>:CAN[:ANALyze]:SETup:RECESSive

機能 CAN バス信号解析のリセツシブ電位を設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN[:ANALyze]:SETup:
RECESSive {HIGH|LOW}
:SERIALbus<x>:CAN[:ANALyze]:SETup:
RECESSive?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:ANALYZE:SETUP:
RECESSIVE HIGH
:SERIALBUS1:CAN:ANALYZE:SETUP:
RECESSIVE?
-> :SERIALBUS1:CAN:ANALYZE:SETUP:
RECESSIVE HIGH

:SERIALbus<x>:CAN[:ANALyze]:SETup:SOURce

機能 CAN バス信号解析のソースを設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN[:ANALyze]:SETup:
SOURce {<Nrf>|MATH<y>}
:SERIALbus<x>:CAN[:ANALyze]:SETup:
SOURce?
<x> = 1 ~ 4
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :SERIALBUS1:CAN:ANALYZE:SETUP:
SOURCE 1
:SERIALBUS1:CAN:ANALYZE:SETUP:
SOURCE?
-> :SERIALBUS1:CAN:ANALYZE:SETUP:
SOURCE 1

:SERIALbus<x>:CAN[:ANALyze]:SETup:SPOint

機能 CAN バス信号解析のサンプルポイントを設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN[:ANALyze]:SETup:
SPOint {<Nrf>}
:SERIALbus<x>:CAN[:ANALyze]:SETup:
SPOint?
<x> = 1 ~ 4
<Nrf> = 18.8 ~ 90.6

例 :SERIALBUS1:CAN:ANALYZE:SETUP:
SPOINT 18.8
:SERIALBUS1:CAN:ANALYZE:SETUP:
SPOINT?
-> :SERIALBUS1:CAN:ANALYZE:SETUP:
SPOINT 18.8

:SERIALbus<x>:CAN:DETail?

機能 CAN バス信号の解析結果リストに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CAN:DETail?
<x> = 1 ~ 4

:SERIALbus<x>:CAN:DETail:DISPlay

機能 CAN バス信号の解析結果リストの表示モードを設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:DETail:
DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:CAN:DETail:DISPlay?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:DETAIL:DISPLAY FULL
:SERIALBUS1:CAN:DETAIL:DISPLAY?
-> :SERIALBUS1:CAN:DETAIL:
DISPLAY FULL

5.25 SERIALbus グループ

:SERIALbus<x>:CAN:DETail:FILTer?

機能 CAN バス信号の解析結果リストのフィルタ機能に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CAN:DETail:FILTer?
<x> = 1 ~ 4

:SERIALbus<x>:CAN:DETail:FILTer:IDENtifier?

機能 CAN バス信号の解析結果リストのフィルタの ID に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CAN:DETail:FILTer:
IDENtifier?
<x> = 1 ~ 4

:SERIALbus<x>:CAN:DETail:FILTer:IDENtifier:HEXa

機能 CAN バス信号の解析結果リストのフィルタ条件の ID を 16 進数で設定します。

構文 :SERIALbus<x>:CAN:DETail:FILTer:
IDENtifier:HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:DETAIL:FILTER:
IDENTIFIER:HEXA "1AB"

:SERIALbus<x>:CAN:DETail:FILTer:IDENtifier:MFORmat

機能 CAN バス信号の解析結果リストのフィルタの ID のフレームフォーマット (標準 / 拡張) を設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:DETail:FILTer:
IDENtifier:MFORmat {EXTENDED|
STANDARD}
:SERIALbus<x>:CAN:DETail:FILTer:
IDENtifier:MFORmat?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:DETAIL:FILTER:
IDENTIFIER:MFORMAT EXTENDED
:SERIALBUS1:CAN:DETAIL:FILTER:
IDENTIFIER:MFORMAT?
-> :SERIALBUS1:CAN:DETAIL:FILTER:
IDENTIFIER:MFORMAT EXTENDED

:SERIALbus<x>:CAN:DETail:FILTer:IDENtifier:MODE

機能 CAN バス信号の解析結果リストのフィルタの ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:DETail:FILTer:
IDENtifier:MODE {<Boolean>}
:SERIALbus<x>:CAN:DETail:FILTer:
IDENtifier:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:DETAIL:FILTER:
IDENTIFIER:MODE ON
:SERIALBUS1:CAN:DETAIL:FILTER:
IDENTIFIER:MODE?
-> :SERIALBUS1:CAN:DETAIL:FILTER:
IDENTIFIER:MODE 1

:SERIALbus<x>:CAN:DETail:LIST:ITEM?

機能 CAN バス信号の解析結果リストに表示される項目を問い合わせます。

構文 :SERIALbus<x>:CAN:DETail:LIST:ITEM?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:DETAIL:LIST:ITEM?
-> :SERIALBUS1:CAN:DETAIL:LIST:
ITEM "No.,Time(ms),Frame,ID,DLC,
Data,CRC,Ack,Information"

:SERIALbus<x>:CAN:DETail:LIST:VALue?

機能 CAN バス信号の解析結果リストの指定した解析番号の全データを問い合わせます。

構文 :SERIALbus<x>:CAN:DETail:LIST:
VALue? {<Nrf>|MAXimum|MINimum}
<x> = 1 ~ 4
<Nrf> = -99999 ~ 99999

例 :SERIALBUS1:CAN:DETAIL:LIST:VALUE? 0
-> :SERIALBUS1:CAN:DETAIL:LIST:
VALUE "0,0.0750,Data,012,1,FE,2263,
Y,,,"

:SERIALbus<x>:CAN:DETail:TREference

機能 CAN バス信号の解析結果リストの時間基準を設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:DETail:
TREference {PROW|TPOStion}
:SERIALbus<x>:CAN:DETail:TREference?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:DETAIL:
TREFERENCE PROW
:SERIALBUS1:CAN:DETAIL:TREFERENCE?
-> :SERIALBUS1:CAN:DETAIL:
TREFERENCE PROW

:SERIALbus<x>:CAN:SEARCH?

機能 CAN バス信号の検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CAN:SEARCH?
<x> = 1 ~ 4

:SERIALbus<x>:CAN:SEARCH:ABORT

機能 CAN バス信号の検索を中止します。

構文 :SERIALbus<x>:CAN:SEARCH:ABORT
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:ABORT

:SERIALbus<x>:CAN:SEARCH:EXECute

機能 CAN バス信号の検索を実行します。

構文 :SERIALbus<x>:CAN:SEARCH:EXECute
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:EXECUTE

:SERIALbus<x>:CAN:SEARCH:FJUMP:ACK

機能 CAN バス信号検索の結果の ACK Field へのフィールドジャンプを実行します。

構文 :SERIALbus<x>:CAN:SEARCH:FJUMP:ACK
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:FJUMP:ACK

:SERIALbus<x>:CAN:SEARCH:FJUMP:CONTROL

機能 CAN バス信号検索の結果の Control Field へのフィールドジャンプを実行します。

構文 :SERIALbus<x>:CAN:SEARCH:FJUMP:
CONTROL
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:FJUMP:CONTROL

:SERIALbus<x>:CAN:SEARCH:FJUMP:CRC

機能 CAN バス信号検索の結果の CRC Field へのフィールドジャンプを実行します。

構文 :SERIALbus<x>:CAN:SEARCH:FJUMP:CRC
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:FJUMP:CRC

:SERIALbus<x>:CAN:SEARCH:FJUMP:DATA

機能 CAN バス信号検索の結果の Data Field へのフィールドジャンプを実行します。

構文 :SERIALbus<x>:CAN:SEARCH:FJUMP:DATA
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:FJUMP:DATA

:SERIALbus<x>:CAN:SEARCH:FJUMP:IDENTifier

機能 CAN バス信号検索の結果の Identifier へのフィールドジャンプを実行します。

構文 :SERIALbus<x>:CAN:SEARCH:FJUMP:
IDENTifier
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:FJUMP:
IDENTIFIER

:SERIALbus<x>:CAN:SEARCH:FJUMP:SOF

機能 CAN バス信号検索の結果の SOF へのフィールドジャンプを実行します。

構文 :SERIALbus<x>:CAN:SEARCH:FJUMP:SOF
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:FJUMP:SOF

:SERIALbus<x>:CAN:SEARCH:SELEct

機能 CAN バス信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。

構文 :SERIALbus<x>:CAN:SEARCH:
SELEct {<NRf>|MAXimum}
:SERIALbus<x>:CAN:SEARCH:SELEct?
<x> = 1 ~ 4
<NRf> = 0 ~ 49999

例 :SERIALBUS1:CAN:SEARCH:SELEct 1
:SERIALBUS1:CAN:SEARCH:SELEct?
-> :SERIALBUS1:CAN:SEARCH:
SELEct 1.50000000

解説 検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:SERIALbus<x>:CAN:SEARCH:SELEct? MAXimum

機能 CAN バス信号検索の検索点の総数を問い合わせます。

構文 :SERIALbus<x>:CAN:SEARCH:
SELEct? MAXimum
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:
SELEct? MAXIMUM
-> :SERIALBUS1:CAN:SEARCH:SELEct 100

解説 検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:SERIALbus<x>:CAN:SEARCH:SETup?

機能 CAN バス信号検索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 :SERIALbus<x>:CAN:SEARCH:SETup?
<x> = 1 ~ 4

5.25 SERIALbus グループ

:SERIALbus<x>:CAN:SEARCH:SETup:EFram me?

機能 CAN バス信号検索の Error に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CAN:SEARCH:SETup:
EFram?
<x> = 1 ~ 4

:SERIALbus<x>:CAN:SEARCH:SETup:EFram e[:MODE]

機能 CAN バス信号検索の Error Frame を設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCH:SETup:
EFram[:MODE] {<Boolean>}
:SERIALbus<x>:CAN:SEARCH:SETup:
EFram[:MODE]?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:EFram:
MODE ON
:SERIALBUS1:CAN:SEARCH:SETUP:EFram:
MODE?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
EFram:MODE 1

:SERIALbus<x>:CAN:SEARCH:SETup:EFram e:CRC

機能 CAN バス信号検索の CRC Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCH:SETup:
EFram:CRC {<Boolean>}
:SERIALbus<x>:CAN:SEARCH:SETup:
EFram:CRC?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:EFram:
CRC ON
:SERIALBUS1:CAN:SEARCH:SETUP:EFram:
CRC?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
EFram:CRC 1

:SERIALbus<x>:CAN:SEARCH:SETup:EFram e:STUFF

機能 CAN バス信号検索の STUFF Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCH:SETup:
EFram:STUFF {<Boolean>}
:SERIALbus<x>:CAN:SEARCH:SETup:
EFram:STUFF?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:EFram:
STUFF ON
:SERIALBUS1:CAN:SEARCH:SETUP:EFram:
STUFF?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
EFram:STUFF 1

:SERIALbus<x>:CAN:SEARCH:SETup:IDDa ta?

機能 CAN バス信号検索の ID とデータ条件に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CAN:SEARCH:SETup:
IDData?
<x> = 1 ~ 4

:SERIALbus<x>:CAN:SEARCH:SETup[:IDDa ta]:ACK?

機能 CAN バス信号検索の ID とデータ条件の ACK 関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CAN:SEARCH:SETup[:
IDData]:ACK?
<x> = 1 ~ 4

:SERIALbus<x>:CAN:SEARCH:SETup[:IDDa ta]:ACK:MODE

機能 CAN バス信号検索の ID とデータ条件の ACK モードの設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCH:SETup[:
IDData]:ACK:MODE {<Boolean>}
:SERIALbus<x>:CAN:SEARCH:SETup[:
IDData]:ACK:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
ACK:MODE ON
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
ACK:MODE?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:ACK:MODE 1

:SERIALbus<x>:CAN:SEARCH:SETup[:IDDa ta]:ACK:TYPE

機能 CAN バス信号検索の ID とデータ条件の ACK 条件の設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCH:SETup[:
IDData]:ACK:TYPE {ACK|ACKBoth|
NONack}
:SERIALbus<x>:CAN:SEARCH:SETup[:
IDData]:ACK:TYPE?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
ACK:TYPE ACK
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
ACK:TYPE?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:ACK:TYPE ACK

:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA?

機能 CAN バス信号検索の ID とデータ条件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 :SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA?
<x> = 1 ~ 4

:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA:CONDition

機能 CAN バス信号検索の ID とデータ条件のデータ判定条件の設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA:CONDition {BETween|EQUAL|FALSE|GREATER|LESS|NOTBetween|NOTEQUAL|TRUE}
:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA:CONDition?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:CONDITION BETWEEN
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:CONDITION?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:CONDITION BETWEEN

:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA:DECimal<y>

機能 CAN バス信号検索の ID とデータ条件の判定データを 10 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA:DECimal<y> {<Nrf>,<Nrf>}
:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA:DECimal<y>?
<x> = 1 ~ 4
<y> = 1, 2
<Nrf> = -9E+18 ~ 9E+18

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:DECIMAL1 1
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:DECIMAL1?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:DECIMAL1 1

:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA:DLC

機能 CAN バス信号検索の ID とデータ条件の有効バイト数 (DLC) の設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA:DLC {<Nrf>}
:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA:DLC?
<x> = 1 ~ 4
<Nrf> = 0 ~ 8

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:DLC 0
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:DLC?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:DLC 0

:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA:DPATtern?

機能 CAN バス信号検索の ID とデータ条件の判定データに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA:DPATtern?
<x> = 1 ~ 4

:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA:DPATtern:HEXa

機能 CAN バス信号検索の ID とデータ条件の判定データを 16 進数で設定します。

構文 :SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA:DPATtern:HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:DPATTERN:HEXA "12"

解説 「:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:DATA:DLC」の設定値によって設定できるバイト数が変動します。

5.25 SERIALbus グループ

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:DPATtern:PATtern

機能 CAN バス信号検索の ID とデータ条件のデータを 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:DPATtern:PATtern {<文字列>}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:DPATtern:PATtern?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:DPATTERN:PATTERN "00110101"
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:DPATTERN:PATTERN?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:DPATTERN:PATTERN "00110101"

解説 「:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:DLC」 の設定値によって設定できるバイト数が変動します。

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:ENDian

機能 CAN バス信号検索の ID とデータ条件の判定データのエンディアンの設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:ENDian {BIG|LITtle}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:ENDian?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:ENDIAN BIG
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:ENDIAN?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:ENDIAN BIG

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:MODE

機能 CAN バス信号検索の ID とデータ条件のデータ条件 (有効 / 無効) の設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:MODE {<Boolean>}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:MODE ON
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:MODE?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:MODE 1

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:MSBLsb

機能 CAN バス信号検索の ID とデータ条件のデータの MSB/LSB のビットを設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:MSBLsb {<Nrf>,<Nrf>}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:MSBLsb?
<x> = 1 ~ 4
<Nrf> = 0 ~ 63

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:MSBLSB 1,0
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:MSBLSB?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:MSBLSB 1,0

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:SIGN

機能 CAN バス信号検索の ID とデータ条件のデータの符号を設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:SIGN {SIGN|UNSign}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:SIGN?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:SIGN SIGN
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:SIGN?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:DATA:SIGN SIGN

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier?

機能 CAN バス信号検索の ID とデータ条件の識別子に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier?
<x> = 1 ~ 4

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:HEXa

機能 CAN バス信号検索の ID とデータ条件の ID を 16 進数で設定します。

構文 :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:IDENTIFIER:HEXA "1AB"

解説 「:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:MFormat」 の設定値によって設定できるバイト数が変動します。

:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:IDENTifier:MFormat

機能 CAN バス信号検索の ID とデータ条件の ID のフレームフォーマット (標準 / 拡張) を設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:IDENTifier:MFormat {EXTended|STANdard}
:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:IDENTifier:MFormat?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:IDENTIFIER:MFORMAT EXTENDED
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:IDENTIFIER:MFORMAT?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:IDENTIFIER:MFORMAT EXTENDED

:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:IDENTifier:MODE

機能 CAN バス信号検索の ID とデータ条件の識別子条件 (有効 / 無効) の設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:IDENTifier:MODE {<Boolean>}
:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:IDENTifier:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:IDENTIFIER:MODE ON
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:IDENTIFIER:MODE?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:IDENTIFIER:MODE 1

:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:IDENTifier:PATtern

機能 CAN バス信号検索の ID とデータ条件の ID 条件を 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:IDENTifier:PATtern{<文字列>}
:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:IDENTifier:PATtern?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:IDENTIFIER:PATTERN "11100001111"
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:IDENTIFIER:PATTERN?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:IDENTIFIER:PATTERN "11100001111"

解説 「:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:IDENTifier:MFormat」の設定値によって設定できるバイト数が変動します。

:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:IDENTifier:PFormat

機能 CAN バス信号検索の ID とデータ条件の ID の入力形式を設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:IDENTifier:PFormat {PATtern|MESSAge}
:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:IDENTifier:PFormat?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:IDENTIFIER:PFORMAT PATTERN
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:IDENTIFIER:PFORMAT?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:IDENTIFIER:PFORMAT PATTERN

:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:MSIGNAL?

機能 CAN バス信号検索の ID とデータ条件のメッセージングシグナルに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:MSIGNAL?
<x> = 1 ~ 4

:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:MSIGNAL:MESSAge:ITEM

機能 CAN バス信号検索の ID とデータ条件のメッセージのアイテムを設定します。

構文 :SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:MSIGNAL:MESSAge:ITEM{<文字列>}
<x> = 1 ~ 4
<文字列> = 32 文字以内

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:MSIGNAL:MESSAGE:ITEM "TEST"

:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:MSIGNAL:SELEct

機能 CAN バス信号検索の ID とデータ条件のメッセージングシグナルの条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:MSIGNAL:SELEct {MESSAge|SIGNAl}
:SERIALbus<x>:CAN:SEARCH:SETup[:IDData]:MSIGNAL:SELEct?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:MSIGNAL:SELECT MESSAGE
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:MSIGNAL:SELECT?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:MSIGNAL:SELECT MESSAGE

5.25 SERIALbus グループ

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGNAL:SIGNAL?

機能 CAN バス信号検索の ID とデータ条件のシグナルに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGNAL:SIGNAL?
<x> = 1 ~ 4

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGNAL:SIGNAL:CONDition

機能 CAN バス信号検索の ID とデータ条件のシグナルのデータ条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGNAL:SIGNAL:
CONDition {BETween|EQual|GREater|
LESS|NOTBetween|NOTEqual}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGNAL:SIGNAL:CONDition?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
MSIGNAL:SIGNAL:CONDITION BETWEEN
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
MSIGNAL:SIGNAL:CONDITION?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:MSIGNAL:SIGNAL:
CONDITION BETWEEN

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGNAL:SIGNAL:DECimal<y>

機能 CAN バス信号検索の ID とデータ条件のシグナルの判定データを 10 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGNAL:SIGNAL:
DECimal<y> {<Nrf>}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGNAL:SIGNAL:DECimal<y>?
<x> = 1 ~ 4
<y> = 1, 2
<Nrf> = -9E+18 ~ 9E+18

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
MSIGNAL:SIGNAL:DECIMAL1 1
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
MSIGNAL:SIGNAL:DECIMAL1?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:MSIGNAL:SIGNAL:
DECIMAL1 1.000E+00

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGNAL:SIGNAL:ITEM

機能 CAN バス信号検索の ID とデータ条件のシグナルのアイテムを設定します。

構文 :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGNAL:SIGNAL:ITEM {<文字列>,
<文字列>}
<x> = 1 ~ 4
<文字列> = 32 文字以内

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
MSIGNAL:SIGNAL:ITEM "Sig _ Test",
"Mess _ Test"

解説 設定する順番は、シグナルアイテム、メッセージアイテムの順です。

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:RTR

機能 CAN バス信号検索の ID とデータ条件の RTR を設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:RTR {<Boolean>}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:RTR?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
RTR ON
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
RTR?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:RTR 1

:SERIALbus<x>:CAN:SEARCh:SETup:MODE

機能 CAN バス信号検索の検索の種類を設定 / 問い合わせします。

構文 :SERIALbus<x>:CAN:SEARCh:SETup:
MODE {EFrame|IDData|SOF}
:SERIALbus<x>:CAN:SEARCh:SETup:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:CAN:SEARCH:SETUP:
MODE EFRAME
:SERIALBUS1:CAN:SEARCH:SETUP:MODE?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
MODE EFRAME

:SERIALbus<x>:CANFD?

機能 CAN FD バス信号に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CANFD?
<x> = 1 ~ 4

:SERIALbus<x>:CANFD:ANALyze?

機能 CAN FD バス信号解析に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CANFD:ANALyze?
<x> = 1 ~ 4

:SERIALbus<x>:CANFD[:ANALyze]:SETup?

機能 CAN FD バス信号解析のバス設定に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CANFD[:ANALyze]:SETup?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:ANALYZE:SETUP?
-> :SERIALBUS1:CANFD:ANALYZE:SETUP

:SERIALbus<x>:CANFD[:ANALyze]:SETup:BRATe

機能 CAN FD バス信号解析のビットレート (データ転送速度) を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD[:ANALyze]:SETup:
BRATe {<Nrf>|USER,<Nrf>}
:SERIALbus<x>:CANFD[:ANALyze]:SETup:
BRATe?
<x> = 1 ~ 4
<Nrf> = 250000、500000、1000000
USER の <Nrf> = 20000 ~ 1000000(設定分解能 100)

例 :SERIALBUS1:CANFD:ANALYZE:SETUP:
BRATE 1000000
:SERIALBUS1:CANFD:ANALYZE:SETUP:
BRATE?
-> :SERIALBUS1:CANFD:ANALYZE:SETUP:
BRATE 1000000

:SERIALbus<x>:CANFD[:ANALyze]:SETup:DBRATe

機能 CAN FD バス信号解析のデータビットレート (データフェーズのデータ転送速度) を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD[:ANALyze]:SETup:
DBRATe {<Nrf>|USER,<Nrf>}
:SERIALbus<x>:CANFD[:ANALyze]:SETup:
DBRATe?
<x> = 1 ~ 4
<Nrf> = 500000、1000000、2000000、
4000000、5000000、8000000
USER の <Nrf> = 250000 ~ 10000000(設定分解能 100)

例 :SERIALBUS1:CANFD:ANALYZE:SETUP:
DBRATE 5000000
:SERIALBUS1:CANFD:ANALYZE:SETUP:
DBRATE?
-> :SERIALBUS1:CANFD:ANALYZE:SETUP:
DBRATE 5000000

:SERIALbus<x>:CANFD[:ANALyze]:SETup:DSPoint

機能 CAN FD バス信号解析のデータフェーズのサンプルポイントを設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD[:ANALyze]:SETup:
DSPoint {<Nrf>}
:SERIALbus<x>:CANFD[:ANALyze]:SETup:
DSPoint?
<x> = 1 ~ 4
<Nrf> = 18.8 ~ 90.6(設定分解能 0.1)

例 :SERIALBUS1:CANFD:ANALYZE:SETUP:
DSPOINT 75
:SERIALBUS1:CANFD:ANALYZE:SETUP:
DSPOINT?
-> :SERIALBUS1:CANFD:ANALYZE:SETUP:
DSPOINT 75.00000000

:SERIALbus<x>:CANFD[:ANALyze]:SETup:FDSTandard

機能 解析する CAN FD バス信号が、ISO 標準か否かを設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD[:ANALyze]:SETup:
FDSTandard {ISO|NISO}
:SERIALbus<x>:CANFD[:ANALyze]:SETup:
FDSTandard?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:ANALYZE:SETUP:
FDSTANDARD ISO
:SERIALBUS1:CANFD:ANALYZE:SETUP:
FDSTANDARD?
-> :SERIALBUS1:CANFD:ANALYZE:SETUP:
FDSTANDARD ISO

:SERIALbus<x>:CANFD[:ANALyze]:SETup:RECCessive

機能 CAN FD バス信号解析のリセッシブ電位を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD[:ANALyze]:SETup:
RECCessive {HIGH|LOW}
:SERIALbus<x>:CANFD[:ANALyze]:SETup:
RECCessive?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:ANALYZE:SETUP:
RECESSIVE HIGH
:SERIALBUS1:CANFD:ANALYZE:SETUP:
RECESSIVE?
-> :SERIALBUS1:CANFD:ANALYZE:SETUP:
RECESSIVE HIGH

5.25 Serialbus グループ

:Serialbus<x>:CANFD[:ANALyze]:SETup:SOURce

機能	CAN FD バス信号解析のソースを設定 / 問い合わせします。
構文	:Serialbus<x>:CANFD[:ANALyze]:SETup: SOURce {<NRf> MATH<y>} :Serialbus<x>:CANFD[:ANALyze]:SETup: SOURce? <x> = 1 ~ 4 <NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4) MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例	:SERIALBUS1:CANFD:ANALYZE:SETUP: SOURCE 1 :SERIALBUS1:CANFD:ANALYZE:SETUP: SOURCE? -> :SERIALBUS1:CANFD:ANALYZE:SETUP: SOURCE 1

:Serialbus<x>:CANFD[:ANALyze]:SETup:SPOint

機能	CAN FD バス信号解析のサンプルポイントを設定 / 問い合わせします。
構文	:Serialbus<x>:CANFD[:ANALyze]:SETup: SPOint {<NRf>} :Serialbus<x>:CANFD[:ANALyze]:SETup: SPOint? <x> = 1 ~ 4 <NRf> = 18.8 ~ 90.6(設定分解能 0.1)
例	:SERIALBUS1:CANFD:ANALYZE:SETUP: SPOINT 75 :SERIALBUS1:CANFD:ANALYZE:SETUP: SPOINT? -> :SERIALBUS1:CANFD:ANALYZE:SETUP: SPOINT 75

:Serialbus<x>:CANFD:DETail?

機能	CAN FD バス信号の解析結果リストに関するすべての設定値を問い合わせます。
構文	:Serialbus<x>:CANFD:DETail? <x> = 1 ~ 4

:Serialbus<x>:CANFD:DETail:DISPlay

機能	CAN FD バス信号の解析結果リストの表示モードを設定 / 問い合わせします。
構文	:Serialbus<x>:CANFD:DETail: DISPlay {FULL LOWer UPPer} :Serialbus<x>:CANFD:DETail:DISPlay <x> = 1 ~ 4
例	:SERIALBUS1:CANFD:DETAIL: DISPLAY FULL :SERIALBUS1:CANFD:DETAIL:DISPLAY? -> :SERIALBUS1:CANFD:DETAIL: DISPLAY FULL

:Serialbus<x>:CANFD:DETail:FILTer?

機能	CAN FD バス信号の解析結果リストのフィルタ機能に関するすべての設定値を問い合わせます。
構文	:Serialbus<x>:CANFD:DETail:FILTer? <x> = 1 ~ 4

:Serialbus<x>:CANFD:DETail:FILTer:IDEntifier?

機能	CAN FD バス信号の解析結果リストのフィルタの ID に関するすべての設定値を問い合わせます。
構文	:Serialbus<x>:CANFD:DETail:FILTer: IDEntifier? <x> = 1 ~ 4

:Serialbus<x>:CANFD:DETail:FILTer:IDEntifier:HEXa

機能	CAN FD バス信号の解析結果リストのフィルタ条件の ID を 16 進数で設定します。
構文	:Serialbus<x>:CANFD:DETail:FILTer: IDEntifier:HEXa {<文字列>} <x> = 1 ~ 4
例	:SERIALBUS1:CANFD:DETAIL:FILTER: IDENTIFIER:HEXA "1AB"

:Serialbus<x>:CANFD:DETail:FILTer:IDEntifier:MFOFormat

機能	CAN FD バス信号の解析結果リストのフィルタの ID のフレームフォーマット (標準 / 拡張) を設定 / 問い合わせします。
構文	:Serialbus<x>:CANFD:DETail:FILTer: IDEntifier:MFOFormat {EXTended STANdard} :Serialbus<x>:CANFD:DETail:FILTer: IDEntifier:MFOFormat? <x> = 1 ~ 4
例	:SERIALBUS1:CANFD:DETAIL:FILTER: IDENTIFIER:MFORMAT EXTENDED :SERIALBUS1:CANFD:DETAIL:FILTER: IDENTIFIER:MFORMAT? -> :SERIALBUS1:CANFD:DETAIL:FILTER: IDENTIFIER:MFORMAT EXTENDED

:SERIALbus<x>:CANFD:DETAIL:FILTER:IDENTIFIER:MODE

機能 CAN FD バス信号の解析結果リストのフィルタの ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:DETAIL:FILTER:
IDENTIFIER:MODE {<Boolean>
:SERIALbus<x>:CANFD:DETAIL:FILTER:
IDENTIFIER:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:DETAIL:FILTER:
IDENTIFIER:MODE ON
:SERIALBUS1:CANFD:DETAIL:FILTER:
IDENTIFIER:MODE?
-> :SERIALBUS1:CANFD:DETAIL:FILTER:
IDENTIFIER:MODE 1

:SERIALbus<x>:CANFD:DETAIL:LIST:ITEM?

機能 CAN FD バス信号の解析結果リストに表示される項目を問い合わせます。

構文 :SERIALbus<x>:CANFD:DETAIL:LIST:
ITEM?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:DETAIL:LIST:ITEM?
-> :SERIALBUS1:CANFD:DETAIL:LIST:
ITEM "No.,Time(ms),Frame,ID,DLC,
Data,CRC,Ack,Information"

:SERIALbus<x>:CANFD:DETAIL:LIST:VALUE?

機能 CAN FD バス信号の解析結果リストの指定した解析番号の全データを問い合わせます。

構文 :SERIALbus<x>:CANFD:DETAIL:LIST:
VALUE? {<Nrf>|MAXimum|MINimum}
<x> = 1 ~ 4
<Nrf> = -49999 ~ 49999

例 :SERIALBUS1:CANFD:DETAIL:LIST:
VALUE? 0
-> :SERIALBUS1:CANFD:DETAIL:LIST:
VALUE "0,0.0750,Data,012,1,FE,2263,
Y,, "

:SERIALbus<x>:CANFD:DETAIL:TREFERENCE

機能 CAN FD バス信号の解析結果リストの時間基準を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:DETAIL:
TREFERENCE {PROW|TPOSITION}
:SERIALbus<x>:CANFD:DETAIL:
TREFERENCE?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:DETAIL:
TREFERENCE PROW
:SERIALBUS1:CANFD:DETAIL:TREFERENCE?
-> :SERIALBUS1:CANFD:DETAIL:
TREFERENCE PROW

:SERIALbus<x>:CANFD:SEARCH?

機能 CAN FD バス信号の検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CANFD:SEARCH?
<x> = 1 ~ 4

:SERIALbus<x>:CANFD:SEARCH:ABORT

機能 CAN FD バス信号の検索を中止します。

構文 :SERIALbus<x>:CANFD:SEARCH:ABORT
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:ABORT

:SERIALbus<x>:CANFD:SEARCH:EXECUTE

機能 CAN FD バス信号の検索を実行します。

構文 :SERIALbus<x>:CANFD:SEARCH:EXECUTE
<x> = 1 ~ 4
例 :SERIALBUS1:CANFD:SEARCH:EXECUTE

:SERIALbus<x>:CANFD:SEARCH:FJUMP:ACK

機能 CAN FD バス信号検索の結果の ACKField へのフィールドジャンプを実行します。
<x> = 1 ~ 4

構文 :SERIALbus<x>:CANFD:SEARCH:FJUMP:ACK
例 :SERIALBUS1:CANFD:SEARCH:FJUMP:ACK

:SERIALbus<x>:CANFD:SEARCH:FJUMP:CONTROL

機能 CAN FD バス信号検索の結果の Control Field へのフィールドジャンプを実行します。

構文 :SERIALbus<x>:CANFD:SEARCH:FJUMP:
CONTROL
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:FJUMP:
CONTROL

5.25 Serialbus グループ

:Serialbus<x>:CANFD:SEARCh:FJUMp:CRC

機能	CAN FD バス信号検索の結果の CRC Field へのフィールドジャンプを実行します。
構文	:Serialbus<x>:CANFD:SEARCh:FJUMp:CRC <x> = 1 ~ 4
例	:SERIALBUS1:CANFD:SEARCH:FJUMP:CRC

:Serialbus<x>:CANFD:SEARCh:FJUMp:DA TA

機能	CAN FD バス信号検索の結果の Data Field へのフィールドジャンプを実行します。
構文	:Serialbus<x>:CANFD:SEARCh:FJUMp: DATA <x> = 1 ~ 4
例	:SERIALBUS1:CANFD:SEARCH:FJUMP:DATA

:Serialbus<x>:CANFD:SEARCh:FJUMp:IDE Ntifier

機能	CAN FD バス信号検索の結果の Identifier へのフィールドジャンプを実行します。
構文	:Serialbus<x>:CANFD:SEARCh:FJUMp: IDENTifier <x> = 1 ~ 4
例	:SERIALBUS1:CANFD:SEARCH:FJUMP: IDENTIFIER

:Serialbus<x>:CANFD:SEARCh:FJUMp:SOF

機能	CAN FD バス信号検索の結果の SOF へのフィールドジャンプを実行します。
構文	:Serialbus<x>:CANFD:SEARCh:FJUMp:SOF <x> = 1 ~ 4
例	:SERIALBUS1:CANFD:SEARCH:FJUMP:SOF

:Serialbus<x>:CANFD:SEARCh:SElect

機能	CAN FD バス信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。
構文	:Serialbus<x>:CANFD:SEARCh: SElect {<Nrf> MAXimum} :Serialbus<x>:CANFD:SEARCh:SElect? <x> = 1 ~ 4 <Nrf> = 0 ~ 49999
例	:SERIALBUS1:CANFD:SEARCH:SELECT 1 :SERIALBUS1:CANFD:SEARCH:SELECT? -> :SERIALBUS1:CANFD:SEARCH: SELECT 1.50000000
解説	検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:Serialbus<x>:CANFD:SEARCh:SElect? MAXimum

機能	CAN FD バス信号検索の検索点の総数を問い合わせます。
構文	:Serialbus<x>:CANFD:SEARCh: SElect? MAXimum <x> = 1 ~ 4
例	:SERIALBUS1:CANFD:SEARCH: SELECT? MAXIMUM -> :SERIALBUS1:CANFD:SEARCH: SELECT 100
解説	検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:Serialbus<x>:CANFD:SEARCh:SETup?

機能	CAN FD バス信号検索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	:Serialbus<x>:CANFD:SEARCh:SETup? <x> = 1 ~ 4

:Serialbus<x>:CANFD:SEARCh:SETup:EFR ame?

機能	CAN FD バス信号検索の Error に関するすべての設定値を問い合わせます。
構文	:Serialbus<x>:CANFD:SEARCh:SETup: EFrame? <x> = 1 ~ 4

:Serialbus<x>:CANFD:SEARCh:SETup:EFR ame:CRC

機能	CAN FD バス信号検索の CRC Error を設定 / 問い合わせします。
構文	:Serialbus<x>:CANFD:SEARCh:SETup: EFrame:CRC {<Boolean>} :Serialbus<x>:CANFD:SEARCh:SETup: EFrame:CRC? <x> = 1 ~ 4
例	:SERIALBUS1:CANFD:SEARCH:SETUP: EFRAME:CRC ON :SERIALBUS1:CANFD:SEARCH:SETUP: EFRAME:CRC? -> :SERIALBUS1:CANFD:SEARCH:SETUP: EFRAME:CRC 1

:Serialbus<x>:CANFD:SEARCh:SETup:EFR ame:CRCEFactor?

機能	CAN FD バス信号検索の CRC エラー要因に関するすべての設定値を問い合わせます。
構文	:Serialbus<x>:CANFD:SEARCh:SETup: EFrame:CRCEFactor? <x> = 1 ~ 4

:SERIALbus<x>:CANFD:SEARCH:SETup:EFRame:CRCEFactor:CRCSequence

機能 CAN FD バス信号検索の CRC エラー要因である CRC Sequence を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup:
EFRame:CRCEFactor:
CRCSequence {<Boolean>}
:SERIALbus<x>:CANFD:SEARCH:SETup:
EFRame:CRCEFactor:CRCSequence?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:CRCEFACTOR:CRCSEQUENCE ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:CRCEFACTOR:CRCSEQUENCE?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:CRCEFACTOR:CRCSEQUENCE 1

:SERIALbus<x>:CANFD:SEARCH:SETup:EFRame:CRCEFactor:SCount

機能 CAN FD バス信号検索の CRC エラー要因である Stuff Count を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup:
EFRame:CRCEFactor:SCount {<Boolean>}
:SERIALbus<x>:CANFD:SEARCH:SETup:
EFRame:CRCEFactor:SCount?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:CRCEFACTOR:SCOUNT ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:CRCEFACTOR:SCOUNT?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:CRCEFACTOR:SCOUNT 1

:SERIALbus<x>:CANFD:SEARCH:SETup:EFRame:FSTuff

機能 CAN FD バス信号検索の固定 STUFF Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup:
EFRame:FSTuff {<Boolean>}
:SERIALbus<x>:CANFD:SEARCH:SETup:
EFRame:FSTuff?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:FSTUFF ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:FSTUFF?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:FSTUFF 1

:SERIALbus<x>:CANFD:SEARCH:SETup:EFRame[:MODE]

機能 CAN FD バス信号検索の Error Frame を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup:
EFRame[:MODE] {<Boolean>}
:SERIALbus<x>:CANFD:SEARCH:SETup:
EFRame[:MODE]?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:MODE ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:MODE?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:MODE 1

:SERIALbus<x>:CANFD:SEARCH:SETup:EFRame:STUFF

機能 CAN FD バス信号検索の STUFF Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup:
EFRame:STUFF {<Boolean>}
:SERIALbus<x>:CANFD:SEARCH:SETup:
EFRame:STUFF?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:STUFF ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:STUFF?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:STUFF 1

:SERIALbus<x>:CANFD:SEARCH:SETup:FDF:CONdition

機能 CAN FD バス信号検索の FDF の判定条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup:
FDF:CONdition {<Boolean>}
:SERIALbus<x>:CANFD:SEARCH:SETup:
FDF:CONdition?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:FDF:
CONDITION ON
:SERIALBUS1:CANFD:SEARCH:SETUP:FDF:
CONDITION?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
FDF:CONDITION 1

解説 ON のときは CAN FD、OFF のときは CAN のフレームを検索します。

5.25 SERIALbus グループ

:SERIALbus<x>:CANFD:SEARCH:SETup:IDData?

機能 CAN FD バス信号検索の ID/Data 条件に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup:IDData?
<x> = 1 ~ 4

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:ACK?

機能 CAN FD バス信号検索の ID/Data 条件の ACK に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:ACK?
<x> = 1 ~ 4

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:ACK:MODE

機能 CAN FD バス信号検索の ID/Data 条件の ACK モードの設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:ACK:MODE {<Boolean>}
:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:ACK:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:ACK:MODE ON
:SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:ACK:MODE?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:ACK:MODE 1

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:ACK:TYPE

機能 CAN FD バス信号検索の ID/Data 条件の ACK 条件の設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:ACK:TYPE {ACK|ACKBoth|NONack}
:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:ACK:TYPE?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:ACK:TYPE ACK
:SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:ACK:TYPE?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:ACK:TYPE ACK

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:DATA?

機能 CAN FD バス信号検索の ID/Data 条件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 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:DATA?
<x> = 1 ~ 4

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:DATA:BCount

機能 CAN FD バス信号検索の ID/Data 条件のパターン比較する位置を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:DATA:BCOUNT {<Nrf>}
<x> = 1 ~ 4
<Nrf> = 0 ~ 63

例 :SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:DATA:BCOUNT 4
:SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:DATA:BCOUNT?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:DATA:BCOUNT 4

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:DATA:CONDition

機能 CAN FD バス信号検索の ID/Data 条件のデータ判定条件の設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:DATA:CONDition {BETween|EQUal|FALSe|GREater|LESS|NOTBetween|NOTEqual|TRUE}
:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:DATA:CONDition?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:DATA:CONDITION BETWEEN
:SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:DATA:CONDITION?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:DATA:CONDITION BETWEEN

:SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:DBYTE

機能 CAN FD バス信号検索の ID/Data 条件のデータのバイト数を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:DBYTE {<Nrf>}
<x> = 1 ~ 4
<Nrf> = 0 ~ 8

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DBYTE 2
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DBYTE?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DBYTE 2

:SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:DECIMAL<y>

機能 CAN FD バス信号検索の ID/Data 条件の判定データを 10 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:DECIMAL<y> {<Nrf>}
:SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:DECIMAL<y>?
<x> = 1 ~ 4
<y> = 1, 2
<Nrf> = -9E+18 ~ 9E+18

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1 1000
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1 1.0000000E+03

:SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:DPATTERN?

機能 CAN FD バス信号検索の ID/Data 条件の判定データに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:DPATTERN?
<x> = 1 ~ 4

:SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:DPATTERN:HEXA

機能 CAN FD バス信号検索の ID/Data 条件の判定データを 16 進数で設定します。

構文 :SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:DPATTERN:HEXA {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:HEXA "12"

解説 「:SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:DBYTE」の設定値によって設定できるバイト数が変動します。

:SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:DPATTERN:PATTERN

機能 CAN FD バス信号検索の ID/Data 条件のデータを 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:DPATTERN:
PATTERN {<文字列>}
:SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:DPATTERN:PATTERN?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:
PATTERN "00110101"
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:PATTERN?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:
PATTERN "00110101"

解説 「:SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:DBYTE」の設定値によって設定できるバイト数が変動します。

:SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:ENDIAN

機能 CAN FD バス信号検索の ID/Data 条件の判定データのエンディアンの設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:ENDIAN {BIG|LITTLE}
:SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:ENDIAN?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:ENDIAN BIG
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:ENDIAN?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:ENDIAN BIG

:SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:MODE

機能 CAN FD バス信号検索の ID/Data 条件のデータ条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:MODE {<Boolean>}
:SERIALbus<x>:CANFD:SEARCH:SETUP[:IDData]:DATA:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:MODE ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:MODE?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:MODE 1

5.25 SERIALbus グループ

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:DATA:MSBLSb

機能 CAN FD バス信号検索の ID/Data 条件のデータの MSB/LSB のビットを設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:DATA:MSBLSb {<NRf>,<NRf>}
:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:DATA:MSBLSb?

<x> = 1 ~ 4

<NRf> = 0 ~ 63

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:MSBLSB 15,0
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:MSBLSB?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:MSBLSB 15,0

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:DATA:SIGN

機能 CAN FD バス信号検索の ID/Data 条件のデータの符号を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:DATA:SIGN {SIGN|UNSign}
:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:DATA:SIGN?

<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:SIGN SIGN
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:SIGN?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:SIGN SIGN

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTifier?

機能 CAN FD バス信号検索の ID/Data 条件の識別子に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTifier?
<x> = 1 ~ 4

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTifier:HEXa

機能 CAN FD バス信号検索の ID/Data 条件の ID を 16 進数で設定します。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTifier:HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:HEXa "1AB"

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTifier:MFORMAT

機能 CAN FD バス信号検索の ID/Data 条件の ID のフレームフォーマット (標準 / 拡張) を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTifier:
MFORMAT {EXTended|STANDARD}
:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTifier:MFORMAT?

<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:MFORMAT EXTENDED
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:MFORMAT?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:MFORMAT EXTENDED

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTifier:MODE

機能 CAN FD バス信号検索の ID/Data 条件の識別子条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTifier:MODE {<Boolean>}
:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTifier:MODE?

<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:MODE ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:MODE?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:MODE 1

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTifier:PATtern

機能 CAN FD バス信号検索の ID/Data 条件の ID 条件を 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTifier:PATtern {<文字列>}
:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTifier:PATtern?

<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:
PATTERN "11100001111"
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:PATTERN?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:
PATTERN "11100001111"

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTIFIER:PFORMAT

機能 CAN FD バス信号検索の ID/Data 条件の ID の入力形式を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTIFIER:PFORMAT {PATTERN|MESSAGE}
:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:IDENTIFIER:PFORMAT?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:PFORMAT PATTERN
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:PFORMAT?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:PFORMAT PATTERN

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL?

機能 CAN FD バス信号検索の ID/Data 条件のメッセージングナルに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL?
<x> = 1 ~ 4

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL:MESSAGE:ITEM

機能 CAN FD バス信号検索の ID/Data 条件のメッセージのアイテムを設定します。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL:MESSAGE:ITEM {<文字列>}
<x> = 1 ~ 4
<文字列> = 32 文字以内

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:MESSAGE:ITEM "TEST"

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL:SElect

機能 CAN FD バス信号検索の ID/Data 条件のメッセージングナルの条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL:SElect {MESSAGE|SIGNAL}
:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL:SElect?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SELECT MESSAGE
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SELECT?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SELECT MESSAGE

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL:SIGNAL?

機能 CAN FD バス信号検索の ID/Data 条件のシグナルに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL:SIGNAL?
<x> = 1 ~ 4

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL:SIGNAL:CONDition

機能 CAN FD バス信号検索の ID/Data 条件のシグナルのデータ条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL:SIGNAL:
CONDition {BETween|EQual|GREater|LESS|NOTBetween|NOTEqual}
:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL:SIGNAL:CONDition?
<x> = 1 ~ 4

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SIGNAL:
CONDITION BETWEEN
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SIGNAL:CONDITION?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SIGNAL:
CONDITION BETWEEN

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL:SIGNAL:DECimal<y>

機能 CAN FD バス信号検索の ID/Data 条件のシグナルの判定データを 10 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL:SIGNAL:
DECimal<y> {<NRf>}
:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL:SIGNAL:DECimal<y>?
<x> = 1 ~ 4
<y> = 1, 2
<NRf> = -9E+18 ~ 9E+18

例 :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SIGNAL:DECIMAL1 1000
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SIGNAL:DECIMAL1?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SIGNAL:
DECIMAL1 1.0000000E+03

5.25 SERIALbus グループ

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL:SIGNAL:ITEM

機能	CAN FD バス信号検索の ID とデータ条件のシグナルのアイテムを設定します。
構文	:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:MSIGNAL:SIGNAL:ITEM {<文字列>,<文字列>} <x> = 1 ~ 4 <文字列> = 32 文字以内
例	:SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:MSIGNAL:SIGNAL:ITEM "Sig_Test","Mess_Test"
解説	設定する順番は、シグナルアイテム、メッセージアイテムの順です。

:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:RTR

機能	CAN FD バス信号検索の ID とデータ条件の RTR を設定 / 問い合わせします。
構文	:SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:RTR {<Boolean>} :SERIALbus<x>:CANFD:SEARCH:SETup[:IDData]:RTR? <x> = 1 ~ 4
例	:SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:RTR ON :SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:RTR? -> :SERIALBUS1:CANFD:SEARCH:SETUP:IDDATA:RTR 1

:SERIALbus<x>:CANFD:SEARCH:SETup:MODE

機能	CAN FD バス信号検索の検索の種類を設定 / 問い合わせします。
構文	:SERIALbus<x>:CANFD:SEARCH:SETup:MODE {EFrame ESI FDF IDData SOF} :SERIALbus<x>:CANFD:SEARCH:SETup:MODE? <x> = 1 ~ 4
例	:SERIALBUS1:CANFD:SEARCH:SETUP:MODE EFRAME :SERIALBUS1:CANFD:SEARCH:SETUP:MODE? -> :SERIALBUS1:CANFD:SEARCH:SETUP:MODE EFRAME

:SERIALbus<x>:CXPI?

機能	CXPI バス信号の解析 / 検索に関するすべての設定値を問い合わせます。
構文	:SERIALbus<x>:CXPI? <x> = 1 ~ 4

:SERIALbus<x>:CXPI:ANALyze?

機能	CXPI バス信号解析に関するすべての設定値を問い合わせます。
構文	:SERIALbus<x>:CXPI:ANALyze? <x> = 1 ~ 4

:SERIALbus<x>:CXPI[:ANALyze]:SETup?

機能	CXPI バス信号解析のバス設定に関するすべての設定値を問い合わせます。
構文	:SERIALbus<x>:CXPI[:ANALyze]:SETup? <x> = 1 ~ 4

:SERIALbus<x>:CXPI[:ANALyze]:SETup:BRATE

機能	CXPI バス信号解析のビットレート (データ転送速度) を設定 / 問い合わせします。
構文	:SERIALbus<x>:CXPI[:ANALyze]:SETup:BRATE {<Nrf> USER,<Nrf>} :SERIALbus<x>:CXPI[:ANALyze]:SETup:BRATE? <x> = 1 ~ 4 <Nrf> = 4800、9600、19200 USER の <Nrf> = 4000 ~ 50000
例	:SERIALBUS1:CXPI:ANALYZE:SETUP:BRATE 19200 :SERIALBUS1:CXPI:ANALYZE:SETUP:BRATE? -> :SERIALBUS1:CXPI:ANALYZE:SETUP:BRATE 19200

:SERIALbus<x>:CXPI[:ANALyze]:SETup:CEDetection

機能	CXPI バス信号解析の Counter Error 検出の有効 / 無効を設定 / 問い合わせします。
構文	:SERIALbus<x>:CXPI[:ANALyze]:SETup:CEDetection {<Boolean>} :SERIALbus<x>:CXPI[:ANALyze]:SETup:CEDetection? <x> = 1 ~ 4
例	:SERIALBUS1:CXPI:ANALYZE:SETUP:CEDETECTION ON :SERIALBUS1:CXPI:ANALYZE:SETUP:CEDETECTION? -> :SERIALBUS1:CXPI:ANALYZE:SETUP:CEDETECTION 1

:SERIALbus<x>:CXPI[:ANALyze]:SETup:CTOLerance

機能 CXPI バス信号解析のクロック許容差を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI[:ANALyze]:SETup:CTOLerance {<NRf>}
:SERIALbus<x>:CXPI[:ANALyze]:SETup:CTOLerance?
<x> = 1 ~ 4
<NRf> = ± 0.5% ~ ± 5.0%(設定分解能 0.1%)

例 :SERIALBUS1:CXPI:ANALYZE:SETUP:CTOLERANCE 5
:SERIALBUS1:CXPI:ANALYZE:SETUP:CTOLERANCE?
-> :SERIALBUS1:CXPI:ANALYZE:SETUP:CTOLERANCE 5.00E+00

:SERIALbus<x>:CXPI[:ANALyze]:SETup:SOURce

機能 CXPI バス信号解析のソースを設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI[:ANALyze]:SETup:SOURce {<NRf>|MATH<y>}
:SERIALbus<x>:CXPI[:ANALyze]:SETup:SOURce?
<x> = 1 ~ 4
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :SERIALBUS1:CXPI:ANALYZE:SETUP:SOURCE 1
:SERIALBUS1:CXPI:ANALYZE:SETUP:SOURCE?
-> :SERIALBUS1:CXPI:ANALYZE:SETUP:SOURCE 1

:SERIALbus<x>:CXPI[:ANALyze]:SETup:TSAMple

機能 CXPI バス信号解析の論理値 1/0 判定の閾値を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI[:ANALyze]:SETup:TSAMple {<NRf>}
<x> = 1 ~ 4
<NRf> = 0.01Tbit ~ 0.3Tbit(設定分解能 0.001Tbit)

例 :SERIALBUS1:CXPI:ANALYZE:SETUP:TSAMPLE 0.04
:SERIALBUS1:CXPI:ANALYZE:SETUP:TSAMPLE?
-> :SERIALBUS1:CXPI:ANALYZE:SETUP:TSAMPLE 0.04

解説 論理値 1/0 判定の閾値については、本体ユーザーズマニュアル機能編の CXPI 解析の T Sample をご覧ください。

:SERIALbus<x>:CXPI:DETail?

機能 CXPI バス信号の解析結果リストに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CXPI:DETail?
<x> = 1 ~ 4

:SERIALbus<x>:CXPI:DETail:DISPlay

機能 CXPI バス信号の解析結果リストの表示モードを設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:DETail:DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:CXPI:DETail:DISPlay?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:DETAIL:DISPLAY FULL
:SERIALBUS1:CXPI:DETAIL:DISPLAY?
-> :SERIALBUS1:CXPI:DETAIL:DISPLAY FULL

:SERIALbus<x>:CXPI:DETail:LIST:ALL?

機能 CXPI バス信号の解析結果リストのすべての解析番号の全データを問い合わせます。

構文 :SERIALbus<x>:CXPI:DETail:LIST:ALL?
<x> = 1 ~ 4

例 :SERIALBUS<x>:CXPI:DETAIL:LIST:ALL?
-> #8(8 桁のバイト数)(データ列)

解説 すべての解析番号の全データがブロックデータ形式で返されます。解析番号ごとのデータ間は、ASCII コードの「0AH」で区切られます。

:SERIALbus<x>:CXPI:DETail:LIST:ITEM?

機能 CXPI バス信号の解析結果リストに表示される項目を問い合わせます。

構文 :SERIALbus<x>:CXPI:DETail:LIST:ITEM?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:DETAIL:LIST:ITEM?
-> :SERIALBUS1:CXPI:DETAIL:LIST:ITEM "No.,Time(ms),ID,DLC,W/S,CT,Data,CRC,Information"

:SERIALbus<x>:CXPI:DETail:LIST:VALue?

機能 CXPI バス信号の解析結果リストの指定した解析番号の全データを問い合わせます。

構文 :SERIALbus<x>:CXPI:DETail:LIST:VALue? {<NRf>|MAXimum|MINimum}
<x> = 1 ~ 4
<NRf> = -9999 ~ 9999

例 :SERIALBUS1:CXPI:DETAIL:LIST:VALUE? 1
-> "1,4.7228,P3,8,01,0,00 00 00 00 01 03 05 06,FE,,,"

5.25 Serialbus グループ

:Serialbus<x>:CXPI:SEARCh?

機能 CXPI バス信号の検索に関するすべての設定値を問い合わせます。

構文 :Serialbus<x>:CXPI:SEARCh?
<x> = 1 ~ 4

:Serialbus<x>:CXPI:SEARCh:ABORt

機能 CXPI バス信号の検索を中止します。

構文 :Serialbus<x>:CXPI:SEARCh:ABORt
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:ABORT

:Serialbus<x>:CXPI:SEARCh:EXECute

機能 CXPI バス信号の検索を実行します。

構文 :Serialbus<x>:CXPI:SEARCh:EXECute
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:EXECUTE

:Serialbus<x>:CXPI:SEARCh:SELEct

機能 CXPI バス信号検索のズームウインドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。

構文 :Serialbus<x>:CXPI:SEARCh:
SELEct {<NRf>|MAXimum}
:Serialbus<x>:CXPI:SEARCh:SELEct?
<x> = 1 ~ 4
<NRf> = 0 ~ 49999

例 :SERIALBUS1:CXPI:SEARCH:SELECT 1
:SERIALBUS1:CXPI:SEARCH:SELECT?
-> :SERIALBUS1:CXPI:SEARCH:
SELECT 1.50000000

解説 検索された位置がない場合は、"NAN" が問い合わせの応答として返ってきます。

:Serialbus<x>:CXPI:SEARCh:SETUp?

機能 CXPI バス信号検索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 :Serialbus<x>:CXPI:SEARCh:SETUp?
<x> = 1 ~ 4

:Serialbus<x>:CXPI:SEARCh:SETUp:ERRor?

機能 CXPI バス信号検索の Error に関するすべての設定値を問い合わせます。

構文 :Serialbus<x>:CXPI:SEARCh:SETUp:
ERRor?
<x> = 1 ~ 4

:Serialbus<x>:CXPI:SEARCh:SETUp:ERRor:CLOCK

機能 CXPI バス信号検索の Clock Error を設定 / 問い合わせします。

構文 :Serialbus<x>:CXPI:SEARCh:SETUp:
ERRor:CLOCK {<Boolean>}
:Serialbus<x>:CXPI:SEARCh:SETUp:
ERRor:CLOCK?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
CLOCK ON
:SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
CLOCK?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
ERROR:CLOCK 1

:Serialbus<x>:CXPI:SEARCh:SETUp:ERRor:COUNter

機能 CXPI バス信号検索の Counter Error を設定 / 問い合わせします。

構文 :Serialbus<x>:CXPI:SEARCh:SETUp:
ERRor:COUNter {<Boolean>}
:Serialbus<x>:CXPI:SEARCh:SETUp:
ERRor:COUNter?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
COUNTER ON
:SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
COUNTER?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
ERROR:COUNTER 1

:Serialbus<x>:CXPI:SEARCh:SETUp:ERRor:CRc

機能 CXPI バス信号検索の CRC Error を設定 / 問い合わせします。

構文 :Serialbus<x>:CXPI:SEARCh:SETUp:
ERRor:CRc {<Boolean>}
:Serialbus<x>:CXPI:SEARCh:SETUp:
ERRor:CRc?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
CRC ON
:SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
CRC?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
ERROR:CRc 1

:SERIALbus<x>:CXPI:SEARCh:SETup:ERRor:DLENGth

機能 CXPI バス信号検索の Data Length Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup:
ERRor:DLENGth {<Boolean>}
:SERIALbus<x>:CXPI:SEARCh:SETup:
ERRor:DLENGth?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
DLENGTH ON
:SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
DLENGTH?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
ERROR:DLENGTH 1

:SERIALbus<x>:CXPI:SEARCh:SETup:ERRor:FRAMing

機能 CXPI バス信号検索の Framing Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup:
ERRor:FRAMing {<Boolean>}
:SERIALbus<x>:CXPI:SEARCh:SETup:
ERRor:FRAMing?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
FRAMING ON
:SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
FRAMING?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
ERROR:FRAMING 1

:SERIALbus<x>:CXPI:SEARCh:SETup:ERRor:IBS

機能 CXPI バス信号検索の IBS Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup:
ERRor:IBS {<Boolean>}
:SERIALbus<x>:CXPI:SEARCh:SETup:
ERRor:IBS?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
IBS ON
:SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
IBS?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
ERROR:IBS 1

:SERIALbus<x>:CXPI:SEARCh:SETup:ERRor:PARity

機能 CXPI バス信号検索の Parity Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup:
ERRor:PARity {<Boolean>}
:SERIALbus<x>:CXPI:SEARCh:SETup:
ERRor:PARity?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
PARITY ON
:SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
PARITY?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
ERROR:PARITY 1

:SERIALbus<x>:CXPI:SEARCh:SETup:IDDa ta?

機能 CXPI バス信号検索の ID とデータ条件に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup:
IDData?
<x> = 1 ~ 4

:SERIALbus<x>:CXPI:SEARCh:SETup[:IDD ata]:DATA?

機能 CXPI バス信号検索の ID とデータ条件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 :SERIALbus<x>:CXPI:SEARCh:SETup[:
IDData]:DATA?
<x> = 1 ~ 4

:SERIALbus<x>:CXPI:SEARCh:SETup[:IDD ata]:DATA:BCount

機能 CXPI バス信号検索の ID とデータ条件のパターン比較する位置を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup[:
IDData]:DATA:BCount {<NRF>}
:SERIALbus<x>:CXPI:SEARCh:SETup[:
IDData]:DATA:BCount?
<x> = 1 ~ 4
<NRF> = 0 ~ 254

例 :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:BCOUNT 1
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:BCOUNT?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:BCOUNT 1

5.25 SERIALbus グループ

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:CONDition

機能 CXPI バス信号検索の ID とデータ条件のデータ判定条件の設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:CONDition {BETween|EQUAL|FALSE|GREATER|LESS|NOTBetween|NOTEQUAL|TRUE}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:CONDition?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:CONDITION BETWEEN
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:CONDITION?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:CONDITION BETWEEN

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DBYTe

機能 CXPI バス信号検索の ID とデータ条件のデータのバイト数を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DBYTe {<NRF>}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DBYTe?
<x> = 1 ~ 4
<NRF> = 0 ~ 8

例 :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DBYTE 1
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DBYTE?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DBYTE 1

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DECimal<y>

機能 CXPI バス信号検索の ID とデータ条件の判定データを 10 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DECimal<y> {<NRF>}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DECimal<y>?
<x> = 1 ~ 4
<y> = 1, 2
<NRF> = -9E+18 ~ 9E+18

例 :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1 1
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1 1.00000000E+00

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DPATtern?

機能 CXPI バス信号検索の ID とデータ条件の判定データに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DPATtern?
<x> = 1 ~ 4

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DPATtern:HEXa

機能 CXPI バス信号検索の ID とデータ条件の判定データを 16 進数で設定します。

構文 :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DPATtern:HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:HEXA "12"

解説 「:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DBYTE」の設定値によって設定できるバイト数が変動します。

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DPATtern:PATtern

機能 CXPI バス信号検索の ID とデータ条件の判定データを 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DPATtern:
PATtern {<文字列>}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DPATtern:PATtern?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:
PATTERN "00110101"
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:PATTERN?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:
PATTERN "00110101"

解説 「:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DBYTE」の設定値によって設定できるバイト数が変動します。

**:SERIALbus<x>:CXPI:SEARCh:SETup[:IDD
ata]:DATA:ENDian**

機能 CXPI バス信号検索の ID とデータ条件の判定データのエンディアンの設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup[:
IDData]:DATA:ENDian {BIG|LITtle}
:SERIALbus<x>:CXPI:SEARCh:SETup[:
IDData]:DATA:ENDian?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:ENDIAN BIG
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:ENDIAN?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:ENDIAN BIG

**:SERIALbus<x>:CXPI:SEARCh:SETup[:IDD
ata]:DATA:MODE**

機能 CXPI バス信号検索の ID とデータ条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup[:
IDData]:DATA:MODE {<Boolean>}
:SERIALbus<x>:CXPI:SEARCh:SETup[:
IDData]:DATA:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:MODE ON
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:MODE?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:MODE 1

**:SERIALbus<x>:CXPI:SEARCh:SETup[:IDD
ata]:DATA:MSBLsb**

機能 CXPI バス信号検索の ID とデータ条件のデータの MSB/LSB のビットを設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup[:
IDData]:DATA:MSBLsb {<Nrf>,<Nrf>}
:SERIALbus<x>:CXPI:SEARCh:SETup[:
IDData]:DATA:MSBLsb?
<x> = 1 ~ 4
<Nrf> = 0 ~ 63

例 :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:MSBLSB 7,0
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:MSBLSB?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:MSBLSB 7,0

**:SERIALbus<x>:CXPI:SEARCh:SETup[:IDD
ata]:DATA:SIGN**

機能 CXPI バス信号検索の ID とデータ条件のデータの符号を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup[:
IDData]:DATA:SIGN {SIGN|UNSign}
:SERIALbus<x>:CXPI:SEARCh:SETup[:
IDData]:DATA:SIGN?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:SIGN SIGN
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:SIGN?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:SIGN SIGN

**:SERIALbus<x>:CXPI:SEARCh:SETup[:IDD
ata]:FINformation?**

機能 CXPI バス信号検索の ID とデータ条件のフレーム情報に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup[:
IDData]:FINformation?
<x> = 1 ~ 4

**:SERIALbus<x>:CXPI:SEARCh:SETup[:IDD
ata]:FINformation:CT**

機能 CXPI バス信号検索の ID とデータ条件のフレーム情報のカウンタ値を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup[:
IDData]:FINformation:CT {<Nrf>|
DONTcare}
:SERIALbus<x>:CXPI:SEARCh:SETup[:
IDData]:FINformation:CT?
<x> = 1 ~ 4
<Nrf> = 0 ~ 3

例 :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:FINFORMATION:CT 3
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:FINFORMATION:CT?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:FINFORMATION:CT 3

5.25 SERIALbus グループ

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:MODE

機能 CXPI バス信号検索の ID とデータ条件のフレーム情報の (有効 / 無効) を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:MODE {<Boolean>}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:MODE ON
:SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:MODE?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:MODE 1

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:SLEEP

機能 CXPI バス信号検索の ID とデータ条件のフレーム情報の Sleep ビットを設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:SLEEP {0|1|X}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:SLEEP?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:SLEEP 1
:SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:SLEEP?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:SLEEP 1

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:WAKEup

機能 CXPI バス信号検索の ID とデータ条件のフレーム情報の Wakeup ビットを設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:WAKEup {0|1|X}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:WAKEup?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:WAKEUP 1
:SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:WAKEUP?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:WAKEUP 1

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID?

機能 CXPI バス信号検索の ID とデータ条件の ID に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID?
<x> = 1 ~ 4

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:HEXa

機能 CXPI バス信号検索の ID とデータ条件の ID を 16 進数で設定します。

構文 :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:ID:HEXa "1E"

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:MODE

機能 CXPI バス信号検索の ID とデータ条件の ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:MODE {<Boolean>}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:ID:MODE ON
:SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:ID:MODE?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:ID:MODE 1

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:PATtern

機能 CXPI バス信号検索の ID とデータ条件の ID を 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:PATtern {<文字列>}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:PATtern?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:ID:PATTERN "0010000"
:SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:ID:PATTERN?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:ID:PATTERN "0010000"

:SERIALbus<x>:CXPI:SEARCh:SETup[:IDData]:ID:PTYPE

機能 CXPI バス信号検索の ID とデータ条件の PTYPE の判定条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup[:IDData]:ID:PTYPE {DONTcare|NO|YES}
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:ID:PTYPE DONTCARE
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:ID:PTYPE?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:ID:PTYPE DONTCARE

:SERIALbus<x>:CXPI:SEARCh:SETup:MODE

機能 CXPI バス信号検索の種類を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup:
MODE {ERRor|IDData|PTYPE|SOF|
WAKEupsleep}
:SERIALbus<x>:CXPI:SEARCh:SETup:
MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:
MODE SOF
:SERIALBUS1:CXPI:SEARCH:SETUP:MODE?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
MODE SOF

:SERIALbus<x>:CXPI:SEARCh:SETup:PTYPE

機能 CXPI バス信号検索の PTYPE に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup:
PTYPE?
<x> = 1 ~ 4

:SERIALbus<x>:CXPI:SEARCh:SETup:PTYPE:CONDition

機能 CXPI バス信号検索の PTYPE の判定条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup:
PTYPE:CONDition {EPTYPE|NOResponse}
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:PTYPE:
CONDITION EPTYPE
:SERIALBUS1:CXPI:SEARCH:SETUP:PTYPE:
CONDITION?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
PTYPE:CONDITION EPTYPE

解説 EPTYPE のときはすべての、NOResponse のときは応答の無い PTYPE を検索します。

:SERIALbus<x>:CXPI:SEARCh:SETup:WAKEupsleep?

機能 CXPI バス信号検索の Wakeup と Sleep に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup:
WAKEupsleep?
<x> = 1 ~ 4

:SERIALbus<x>:CXPI:SEARCh:SETup:WAKEupsleep:SFRame

機能 CXPI バス信号検索の Sleep フレームを設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup:
WAKEupsleep:SFRame {<Boolean>}
:SERIALbus<x>:CXPI:SEARCh:SETup:
WAKEupsleep:SFRame?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:
WAKEUPSLEEP:SFRAME ON
:SERIALBUS1:CXPI:SEARCH:SETUP:
WAKEUPSLEEP:SFRAME?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
WAKEUPSLEEP:SFRAME 1

:SERIALbus<x>:CXPI:SEARCh:SETup:WAKEupsleep:SLEEP

機能 CXPI バス信号検索の Sleep(クロックが無い状態)を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SETup:
WAKEupsleep:SLEEP {<Boolean>}
:SERIALbus<x>:CXPI:SEARCh:SETup:
WAKEupsleep:SLEEP?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:
WAKEUPSLEEP:SLEEP ON
:SERIALBUS1:CXPI:SEARCH:SETUP:
WAKEUPSLEEP:SLEEP?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
WAKEUPSLEEP:SLEEP 1

5.25 SERIALbus グループ

:SERIALbus<x>:CXPI:SEARCh:SEtUp:WAKEupsleep:WAKEup

機能 CXPIバス信号検索のWakeup(クロック有り状態)を設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SEtUp:WAKEupsleep:WAKEup {<Boolean>}
:SERIALbus<x>:CXPI:SEARCh:SEtUp:WAKEupsleep:WAKEup?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:WAKEUPSLEEP:WAKEUP ON
:SERIALBUS1:CXPI:SEARCH:SETUP:WAKEUPSLEEP:WAKEUP?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:WAKEUPSLEEP:WAKEUP 1

:SERIALbus<x>:CXPI:SEARCh:SEtUp:WAKEupsleep:WPULse

機能 CXPIバス信号検索のWakeupパルスを設定 / 問い合わせします。

構文 :SERIALbus<x>:CXPI:SEARCh:SEtUp:WAKEupsleep:WPULse {<Boolean>}
:SERIALbus<x>:CXPI:SEARCh:SEtUp:WAKEupsleep:WPULse?
<x> = 1 ~ 4

例 :SERIALBUS1:CXPI:SEARCH:SETUP:WAKEUPSLEEP:WPULSE ON
:SERIALBUS1:CXPI:SEARCH:SETUP:WAKEUPSLEEP:WPULSE?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:WAKEUPSLEEP:WPULSE 1

:SERIALbus<x>:DECode?

機能 シリアルバス信号のデコードに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:DECode?
<x> = 1 ~ 4

:SERIALbus<x>:DECode[:FORMAT]

機能 シリアルバス信号のデコード表示形式を設定 / 問い合わせします。

構文 :SERIALbus<x>:DECode[:FORMAT] {BINARY|DECIMAL|HEXa|ASCII|SYMBOL}
:SERIALbus<x>:DECode[:FORMAT]?
<x> = 1 ~ 4

例 :SERIALBUS1:DECODE:FORMAT BINARY
:SERIALBUS1:DECODE:FORMAT?
-> :SERIALBUS1:DECODE:FORMAT BINARY

:SERIALbus<x>:DECode:LABel

機能 シリアルバス信号のデコードのラベル名を設定 / 問い合わせします。

構文 :SERIALbus<x>:DECode:LABel {<文字列>}
:SERIALbus<x>:DECode:LABel?
<x> = 1 ~ 4
<文字列> = 8文字以内

例 :SERIALBUS1:DECODE:LABEL "S1"
:SERIALBUS1:DECODE:LABEL?
-> :SERIALBUS1:DECODE:LABEL "S1"

解説 「:SERIALbus<x>:TYPE」が「SPI」、「SENT」のときは、設定 / 問い合わせできません。

:SERIALbus<x>:DECode:POStion

機能 シリアルバス信号のデコードの垂直ポジションを設定 / 問い合わせします。

構文 :SERIALbus<x>:DECode:POStion {AUTO|MANual,<Nrf>}
:SERIALbus<x>:DECode:POStion?
<x> = 1 ~ 4
MANualの<Nrf> = -3.8 ~ 3.8(div)、(10 div/表示レコード長)ステップ

例 :SERIALBUS1:DECODE:POSITION AUTO
:SERIALBUS1:DECODE:POSITION?

-> :SERIALBUS1:DECODE:POSITION AUTO
解説 「:SERIALbus<x>:TYPE」が「SPI」、「SENT」のときは、設定 / 問い合わせできません。

:SERIALbus<x>:DECode:SENT?

機能 SENT信号のデコードに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:DECode:SENT?
<x> = 1 ~ 4

例 :SERIALBUS1:DECODE:SENT?

:SERIALbus<x>:DECode:SENT:FAST?

機能 SENT信号のFast CHのデコードに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:DECode:SENT:FAST?
<x> = 1 ~ 4

例 :SERIALBUS1:DECODE:SENT:FAST?

:SERIALbus<x>:DECODE:SENT:FAST:LABEL

機能 SENT 信号の Fast CH のデコードのラベル名を設定 / 問い合わせします。

構文 :SERIALbus<x>:DECODE:SENT:FAST:
LABEL {<文字列>}

:SERIALbus<x>:DECODE:SENT:FAST:
LABEL?

<x> = 1 ~ 4

<文字列> = 8 文字以内

例 :SERIALBUS1:DECODE:SENT:FAST:

LABEL "S1-F"

:SERIALBUS1:DECODE:SENT:FAST:LABEL?

-> :SERIALBUS1:DECODE:SENT:FAST:

LABEL "S1-F"

:SERIALbus<x>:DECODE:SENT:FAST:POSITION

機能 SENT 信号の Fast CH のデコードの垂直ポジションを設定 / 問い合わせします。

構文 :SERIALbus<x>:DECODE:SENT:FAST:
POSITION {AUTO|MANual,<Nrf>}

:SERIALbus<x>:DECODE:SENT:FAST:
POSITION?

<x> = 1 ~ 4

MANual の <Nrf> = -3.8 ~ 3.8(div)、(10 div/ 表示レコード長) ステップ

例 :SERIALBUS1:DECODE:SENT:FAST:

POSITION AUTO

:SERIALBUS1:DECODE:SENT:FAST:

POSITION?

-> :SERIALBUS1:DECODE:SENT:FAST:

POSITION AUTO

:SERIALbus<x>:DECODE:SENT:SLOW?

機能 SENT 信号の Slow CH のデコードに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:DECODE:SENT:SLOW?

<x> = 1 ~ 4

例 :SERIALbus<x>:DECODE:SENT:SLOW?

:SERIALbus<x>:DECODE:SENT:SLOW:LABEL

機能 SENT 信号の Slow CH のデコードのラベル名を設定 / 問い合わせします。

構文 :SERIALbus<x>:DECODE:SENT:SLOW:
LABEL {<文字列>}

:SERIALbus<x>:DECODE:SENT:SLOW:
LABEL?

<x> = 1 ~ 4

<文字列> = 8 文字以内

例 :SERIALBUS1:DECODE:SENT:SLOW:

LABEL "S1-S"

:SERIALBUS1:DECODE:SENT:SLOW:LABEL?

-> :SERIALBUS1:DECODE:SENT:SLOW:

LABEL "S1-S"

:SERIALbus<x>:DECODE:SENT:SLOW:POSITION

機能 SENT 信号の Slow CH のデコードの垂直ポジションを設定 / 問い合わせします。

構文 :SERIALbus<x>:DECODE:SENT:SLOW:

POSITION {AUTO|MANual,<Nrf>}

:SERIALbus<x>:DECODE:SENT:SLOW:
POSITION?

<x> = 1 ~ 4

MANual の <Nrf> = -3.8 ~ 3.8(div)、(10 div/ 表示レコード長) ステップ

例 :SERIALBUS1:DECODE:SENT:SLOW:

POSITION AUTO

:SERIALBUS1:DECODE:SENT:SLOW:

POSITION?

-> :SERIALBUS1:DECODE:SENT:SLOW:

POSITION AUTO

:SERIALbus<x>:DECODE:SPI?

機能 SPI バス信号のデコードに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:DECODE:SPI?

<x> = 1 ~ 4

例 :SERIALBUS1:DECODE:SPI?

:SERIALbus<x>:DECODE:SPI:DATA<y>?

機能 SPI バス信号のデコード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 :SERIALbus<x>:DECODE:SPI:DATA<y>?

<x> = 1 ~ 4

<y> = 1, 2

例 :SERIALBUS1:DECODE:SPI:DATA1?

:SERIALbus<x>:DECODE:SPI:DATA<y>:LABEL

機能 SPI バス信号のデコードの各ラベル名を設定 / 問い合わせします。

構文 :SERIALbus<x>:DECODE:SPI:DATA<y>:

LABEL {<文字列>}

:SERIALbus<x>:DECODE:SPI:DATA<y>:

LABEL?

<x> = 1 ~ 4

<y> = 1, 2

<文字列> = 8 文字以内

例 :SERIALBUS1:DECODE:SPI:DATA1:

LABEL "S1-1"

:SERIALBUS1:DECODE:SPI:DATA1:LABEL?

-> :SERIALBUS1:DECODE:SPI:DATA1:

LABEL "S1-1"

5.25 SERIALbus グループ

:SERIALbus<x>:DECODE:SPI:DATA<y>:POSITION

機能 SPI バス信号のデコードの各垂直ポジションを設定 / 問い合わせします。

構文 :SERIALbus<x>:DECODE:SPI:DATA<y>:
POSITION {AUTO|MANual,<NRf>}
:SERIALbus<x>:DECODE:SPI:DATA<y>:
POSITION?
<x> = 1 ~ 4
<y> = 1, 2

MANual の <NRf> = -3.8 ~ 3.8(div)、(10 div/ 表示レコード長) ステップ

例 :SERIALBUS1:DECODE:SPI:DATA1:
POSITION AUTO
:SERIALBUS1:DECODE:SPI:DATA1:
POSITION?
-> :SERIALBUS1:DECODE:SPI:DATA1:
POSITION AUTO

:SERIALbus<x>:DISPLAY

機能 シリアルバス信号の解析をする (ON)/ しない (OFF) を設定 / 問い合わせします。

構文 :SERIALbus<x>:DISPLAY {<Boolean>}
:SERIALbus<x>:DISPLAY?
<x> = 1 ~ 4

例 :SERIALBUS1:DISPLAY ON
:SERIALBUS1:DISPLAY?
-> :SERIALBUS1:DISPLAY 1

:SERIALbus<x>:FLEXray?

機能 FlexRay バス信号の解析 / 検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:FLEXray?
<x> = 1 ~ 4

:SERIALbus<x>:FLEXray:ANALyze?

機能 FlexRay バス信号解析に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:FLEXray:ANALyze?
<x> = 1 ~ 4

:SERIALbus<x>:FLEXray[:ANALyze]:SETup?

機能 FlexRay バス信号解析のバス設定に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:FLEXray[:ANALyze]:
SETup?
<x> = 1 ~ 4

:SERIALbus<x>:FLEXray[:ANALyze]:SETup:BCHannel

機能 FlexRay バス信号解析のチャンネルのバスタイプを設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray[:ANALyze]:
SETup:BCHannel {A|B}
:SERIALbus<x>:FLEXray[:ANALyze]:
SETup:BCHannel?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:ANALYZE:SETUP:
BCHANNEL A
:SERIALBUS1:FLEXRAY:ANALYZE:SETUP:
BCHANNEL?
-> :SERIALBUS1:FLEXRAY:ANALYZE:
SETUP:BCHANNEL A

:SERIALbus<x>:FLEXray[:ANALyze]:SETup:BRATe

機能 FlexRay バス信号解析のビットレート (データ転送速度) を設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray[:ANALyze]:
SETup:BRATe {<NRf>}
:SERIALbus<x>:FLEXray[:ANALyze]:
SETup:BRATe?
<x> = 1 ~ 4

<NRf> = 2500000、5000000、10000000
例 :SERIALBUS1:FLEXRAY:ANALYZE:SETUP:
BRATE 5000000
:SERIALBUS1:FLEXRAY:ANALYZE:SETUP:
BRATE?
-> :SERIALBUS1:FLEXRAY:ANALYZE:
SETUP:BRATE 5000000

**:SERIALbus<x>:FLEXray[:ANALyze]:SETu
p:SOURce**

機能	FlexRay バス信号解析のソースを設定 / 問い合わせします。
構文	:SERIALbus<x>:FLEXray[:ANALyze]: SETup:SOURce {<Nrf> MATH<y>} :SERIALbus<x>:FLEXray[:ANALyze]: SETup:SOURce? <x> = 1 ~ 4 <Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4) MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例	:SERIALBUS1:FLEXRAY:ANALYZE:SETUP: SOURCE 1 :SERIALBUS1:FLEXRAY:ANALYZE:SETUP: SOURCE? -> :SERIALBUS1:FLEXRAY:ANALYZE: SETUP:SOURCE 1

:SERIALbus<x>:FLEXray:DETAil?

機能	FlexRay バス信号の解析結果リストに関するすべての設定値を問い合わせます。
構文	:SERIALbus<x>:FLEXray:DETAil? <x> = 1 ~ 4

:SERIALbus<x>:FLEXray:DETAil:DISPlay

機能	FlexRay バス信号の解析結果リストの表示モードを設定 / 問い合わせします。
構文	:SERIALbus<x>:FLEXray:DETAil: DISPlay {FULL LOWer UPPer} :SERIALbus<x>:FLEXray:DETAil: DISPlay? <x> = 1 ~ 4
例	:SERIALBUS1:FLEXRAY:DETAIL: DISPLAY FULL :SERIALBUS1:FLEXRAY:DETAIL:DISPLAY? -> :SERIALBUS1:FLEXRAY:DETAIL: DISPLAY FULL

**:SERIALbus<x>:FLEXray:DETAil:LIST:IT
EM?**

機能	FlexRay バス信号の解析結果リストに表示される項目を問い合わせます。
構文	:SERIALbus<x>:FLEXray:DETAil:LIST: ITEM? <x> = 1 ~ 4
例	:SERIALBUS1:FLEXRAY:DETAIL:LIST: ITEM? -> :SERIALBUS1:FLEXRAY:DETAIL:LIST: ITEM "No.,Time(ms),S/D,IND,ID,Len, CC,Data,Information,"

**:SERIALbus<x>:FLEXray:DETAil:LIST:VA
Lue?**

機能	FlexRay バス信号の解析結果リストの指定した解析番号の全データを問い合わせます。
構文	:SERIALbus<x>:FLEXray:DETAil:LIST: VALue? {<Nrf> MAXimum MINimum} <x> = 1 ~ 4 <Nrf> = -5000 ~ 5000
例	:SERIALBUS1:FLEXRAY:DETAIL:LIST: VALUE? 0 -> :SERIALBUS1:FLEXRAY:DETAIL:LIST: VALUE" 0,-0.0001968,S,1111, 4, 4,2, 01 02 03 04 05 06 07 08,, "

:SERIALbus<x>:FLEXray:SEARCh?

機能	FlexRay バス信号検索に関するすべての設定値を問い合わせます。
構文	:SERIALbus<x>:FLEXray:SEARCh? <x> = 1 ~ 4

:SERIALbus<x>:FLEXray:SEARCh:ABORt

機能	FlexRay バス信号検索を中止します。
構文	:SERIALbus<x>:FLEXray:SEARCh:ABORt <x> = 1 ~ 4
例	:SERIALBUS1:FLEXRAY:SEARCH:ABORT

:SERIALbus<x>:FLEXray:SEARCh:EXECute

機能	FlexRay バス信号検索を実行します。
構文	:SERIALbus<x>:FLEXray:SEARCh:EXECute <x> = 1 ~ 4
例	:SERIALBUS1:FLEXRAY:SEARCH:EXECUTE

**:SERIALbus<x>:FLEXray:SEARCh:FJUMp:C
Count**

機能	FlexRay バス信号解析の結果を対象に Cycle Count Field へのフィールドジャンプを実行します。
構文	:SERIALbus<x>:FLEXray:SEARCh:FJUMp: CCount <x> = 1 ~ 4
例	:SERIALBUS1:FLEXRAY:SEARCH:FJUMP: CCOUNT

**:SERIALbus<x>:FLEXray:SEARCh:FJUMp:C
RC**

機能	FlexRay バス信号解析の結果を対象に CRC Field へのフィールドジャンプを実行します。
構文	:SERIALbus<x>:FLEXray:SEARCh:FJUMp: CRC <x> = 1 ~ 4
例	:SERIALBUS1:FLEXRAY:SEARCH:FJUMP:CRC

5.25 Serialbus グループ

:Serialbus<x>:FLEXray:SEARCH:FJUMP:FRAMEid

機能 FlexRay バス信号解析の結果を対象に Frame ID Field へのフィールドジャンプを実行します。

構文 :Serialbus<x>:FLEXray:SEARCH:FJUMP:FRAMEid
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:FJUMP:FRAMEID

:Serialbus<x>:FLEXray:SEARCH:FJUMP:HCRC

機能 FlexRay バス信号解析の結果を対象に Header CRC Field へのフィールドジャンプを実行します。

構文 :Serialbus<x>:FLEXray:SEARCH:FJUMP:HCRC
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:FJUMP:HCRC

:Serialbus<x>:FLEXray:SEARCH:FJUMP:PLENgt

機能 FlexRay バス信号解析の結果を対象に Payload Length Field へのフィールドジャンプを実行します。

構文 :Serialbus<x>:FLEXray:SEARCH:FJUMP:PLENgt
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:FJUMP:PLENGTH

:Serialbus<x>:FLEXray:SEARCH:SELEct

機能 FlexRay バス信号検索による検出波形番号を設定し、検出波形番号に相当するズーム位置を問い合わせます。

構文 :Serialbus<x>:FLEXray:SEARCH:SELEct {<Nrf>|MAXimum}
:Serialbus<x>:FLEXray:SEARCH:SELEct?
<x> = 1 ~ 4
<Nrf> = 0 ~ 49999

例 :SERIALBUS1:FLEXRAY:SEARCH:SELECT 1
:SERIALBUS1:FLEXRAY:SEARCH:SELECT?
-> :SERIALBUS1:FLEXRAY:SEARCH:SELECT 1.50000000

:Serialbus<x>:FLEXray:SEARCH:SETup?

機能 FlexRay バス信号検索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 :Serialbus<x>:FLEXray:SEARCH:SETup?
<x> = 1 ~ 4

:Serialbus<x>:FLEXray:SEARCH:SETup:ERror?

機能 FlexRay バス信号検索の Error に関するすべての設定値を問い合わせます。

構文 :Serialbus<x>:FLEXray:SEARCH:SETup:ERror?
<x> = 1 ~ 4

:Serialbus<x>:FLEXray:SEARCH:SETup:ERror:BSS

機能 FlexRay バス信号検索の BSS Error を設定 / 問い合わせします。

構文 :Serialbus<x>:FLEXray:SEARCH:SETup:ERror:BSS {<Boolean>}
:Serialbus<x>:FLEXray:SEARCH:SETup:ERror:BSS?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:BSS ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:BSS?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:BSS 1

:Serialbus<x>:FLEXray:SEARCH:SETup:ERror:CRc

機能 FlexRay バス信号検索の CRC Error を設定 / 問い合わせします。

構文 :Serialbus<x>:FLEXray:SEARCH:SETup:ERror:CRc {<Boolean>}
:Serialbus<x>:FLEXray:SEARCH:SETup:ERror:CRc?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:CRc ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:CRc?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:CRc 1

:SERIALbus<x>:FLEXray:SEARCH:SETup:ERROR:FES

機能 FlexRay バス信号検索の FES Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup:
ERROR:FES {<Boolean>}
:SERIALbus<x>:FLEXray:SEARCH:SETup:
ERROR:FES?

<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
ERROR:FES ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
ERROR:FES?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
ERROR:FES 1

:SERIALbus<x>:FLEXray:SEARCH:SETup:ERROR:HCRC

機能 FlexRay バス信号検索の Header CRC Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup:
ERROR:HCRC {<Boolean>}
:SERIALbus<x>:FLEXray:SEARCH:SETup:
ERROR:HCRC?

<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
ERROR:HCRC ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
ERROR:HCRC?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
ERROR:HCRC 1

:SERIALbus<x>:FLEXray:SEARCH:SETup:IDData?

機能 FlexRay バス信号検索の ID とデータ条件に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup:
IDData?
<x> = 1 ~ 4

:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:CCOUNT?

機能 FlexRay バス信号検索の Cycle Count に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:CCOUNT?
<x> = 1 ~ 4

:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:CCOUNT:CONDITION

機能 FlexRay バス信号検索の Cycle Count のデータ条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:CCOUNT:CONDITION {BETWEEN|
EQUAL|GREATER|LESS|NOTBetween|
NOTEQUAL}

:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:CCOUNT:CONDITION?

<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:CONDITION BETWEEN
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:CONDITION?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:CONDITION BETWEEN

:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:CCOUNT:COUNT<y>

機能 FlexRay バス信号検索の Cycle Count を設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:CCOUNT:COUNT<y> {<NRF>}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:CCOUNT:COUNT<y>?

<x> = 1 ~ 4

<y> = 1, 2

<NRF> = 0 ~ 63

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:COUNT1 10
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:COUNT1?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:COUNT1 10

解説

- 「:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:CCOUNT:CONDITIONEQUAL|GREATER|NOTEQUAL」のときは「:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:CCOUNT:COUNT1」で設定します。
- 「:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:CCOUNT:CONDITION LESS」のときは「:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:CCOUNT:COUNT2」で設定します。
- 「:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:CCOUNT:CONDITIONBETWEEN|NOTBetween」のときは、小さい値を「:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:CCOUNT:COUNT1」、大きい値を「:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:CCOUNT:COUNT2」で設定します。

5.25 SERIALbus グループ

:SERIALbus<x>:FLEXray:SEARCH:SETup[: IDData]:CCOUNT:MODE

機能 FlexRay バス信号検索の ID とデータ条件の Cycle Count モードを設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:CCOUNT:MODE {<Boolean>}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:CCOUNT:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:MODE ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:MODE?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:MODE 1

:SERIALbus<x>:FLEXray:SEARCH:SETup[: IDData]:DATA?

機能 FlexRay バス信号検索の ID とデータ条件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 :SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:DATA?
<x> = 1 ~ 4

:SERIALbus<x>:FLEXray:SEARCH:SETup[: IDData]:DATA:BCOUNT

機能 FlexRay バス信号検索の ID とデータ条件のパターン比較する位置を設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:DATA:BCOUNT {<Nrf>}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:DATA:BCOUNT?
<x> = 1 ~ 4
<Nrf> = 0 ~ 253

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:BCOUNT 1
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:BCOUNT?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:BCOUNT 1

:SERIALbus<x>:FLEXray:SEARCH:SETup[: IDData]:DATA:CONDition

機能 FlexRay バス信号検索の ID とデータ条件のデータ判定条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:DATA:CONDition {BETween|
EQUAL|FALSE|GREATER|LESS|NOTBetween|
NOTEQUAL|TRUE}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:DATA:CONDition?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:CONDITION BETWEEN
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:CONDITION?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:CONDITION BETWEEN

:SERIALbus<x>:FLEXray:SEARCH:SETup[: IDData]:DATA:DBYTE

機能 FlexRay バス信号検索の ID とデータ条件のデータのバイト数を設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:DATA:DBYTE {<Nrf>}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:DATA:DBYTE?
<x> = 1 ~ 4
<Nrf> = 1 ~ 8

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DBYTE 1
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DBYTE?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DBYTE 1

:SERIALbus<x>:FLEXray:SEARCH:SETup[: IDData]:DATA:DECimal<y>

機能 FlexRay バス信号検索の ID とデータ条件の判定データを 10 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:DATA:DECimal<y> {<Nrf>}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:DATA:DECimal<y>?
<x> = 1 ~ 4
<y> = 1, 2
<Nrf> = -9E+18 ~ 9E+18

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1 1
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1

:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:DPATtern?

機能 FlexRay バス信号検索の ID とデータ条件の判定データに関わるすべての設定値を問い合わせます。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:DPATtern?
<x> = 1 ~ 4

:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:DPATtern:HEXa

機能 FlexRay バス信号検索の ID とデータ条件の判定データを 16 進数で設定します。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:DPATtern:HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:HEXA "12"

解説 「:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:DBYTe」の設定値によって設定できるバイト数が変動します。

:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:DPATtern:PATtern

機能 FlexRay バス信号検索の ID とデータ条件のデータを 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:DPATtern:
PATtern {<文字列>}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:DPATtern:PATtern?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:
PATTERN "00110101"
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:PATTERN?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:
PATTERN "00110101"

解説 「:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:DBYTe」の設定値によって設定できるバイト数が変動します。

:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:ENDian

機能 FlexRay バス信号検索の ID とデータ条件の判定データのエンディアンを設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:ENDian {BIG|LITtle}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:ENDian?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:ENDIAN BIG
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:ENDIAN?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:ENDIAN BIG

:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:MODE

機能 FlexRay バス信号検索の ID とデータ条件のデータ条件 (有効 / 無効) の設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:MODE {<Boolean>}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:MODE ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:MODE?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:MODE 1

:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:MSBLsb

機能 FlexRay バス信号検索の ID とデータ条件のデータの MSB/LSB のビットを設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:MSBLsb {<NRF>,<NRF>}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:MSBLsb?
<x> = 1 ~ 4
<NRF> = 0 ~ 63

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:MSBLSB 7,0
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:MSBLSB?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:MSBLSB 7,0

5.25 SERIALbus グループ

:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:SIGN

機能 FlexRay バス信号検索の ID とデータ条件のデータの符号を設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:SIGN {SIGN|UNSign}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:DATA:SIGN?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:SIGN SIGN
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:SIGN?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:SIGN SIGN

:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:FID?

機能 FlexRay バス信号検索の ID とデータ条件の Frame ID に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:FID?
<x> = 1 ~ 4

:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:FID:CONDition

機能 FlexRay バス信号検索の ID とデータ条件の Frame ID のデータ条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:FID:CONDition {BETween|EQUAL|GREATER|LESS|NOTBetween|NOTEQUAL}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:FID:CONDition?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:CONDITION BETWEEN
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:CONDITION?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:CONDITION BETWEEN

:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:FID:ID<y>

機能 FlexRay バス信号検索の ID とデータ条件の Frame ID の値を設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:FID:ID<y> {<Nrf>}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:FID:ID<y>?
<x> = 1 ~ 4
<y> = 1, 2
<Nrf> = 1 ~ 2047

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:ID1 100
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:ID1?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:ID1 100

:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:FID:MODE

機能 FlexRay バス信号検索の ID とデータ条件の Frame ID 条件 (有効 / 無効) の設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:FID:MODE {<Boolean>}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:FID:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:MODE ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:MODE?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:MODE 1

:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:INDicator?

機能 FlexRay バス信号検索の ID とデータ条件の Indicator に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:INDicator?
<x> = 1 ~ 4

**:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:INDicator:MODE**

機能 FlexRay バス信号検索の ID とデータ条件の Indicator 条件 (有効 / 無効) の設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:INDicator:MODE {<Boolean>}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:INDicator:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:MODE ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:MODE?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:MODE 1

**:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:INDicator:NFRame**

機能 FlexRay バス信号検索の ID とデータ条件の Indicator の Null frame を設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:INDicator:NFRame {0|1|X}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:INDicator:NFRame?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:NFRAME 1
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:NFRAME?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:NFRAME 1

**:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:INDicator:PPreamble**

機能 FlexRay バス信号検索の ID とデータ条件の Indicator の Payload preamble を設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:INDicator:PPreamble {0|1|X}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:INDicator:PPreamble?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:PPREAMBLE 1
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:PPREAMBLE?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:PPREAMBLE 1

**:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:INDicator:STFRame**

機能 FlexRay バス信号検索の ID とデータ条件の Indicator の Start frame を設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:INDicator:STFRame {0|1|X}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:INDicator:STFRame?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:STFRAME 1
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:STFRAME?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:STFRAME 1

**:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:INDicator:SYFRame**

機能 FlexRay バス信号検索の ID とデータ条件の Indicator の Sync frame を設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:INDicator:SYFRame {0|1|X}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:
IDData]:INDicator:SYFRame?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:SYFRAME 1
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:SYFRAME?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:SYFRAME 1

**:SERIALbus<x>:FLEXray:SEARCH:SETup:M
ODE**

機能 FlexRay バス信号検索のサーチモードを設定 / 問い合わせします。

構文 :SERIALbus<x>:FLEXray:SEARCH:SETup:
MODE {ERROR|FStart|IDData}
:SERIALbus<x>:FLEXray:SEARCH:SETup:
MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
MODE FSTART
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
MODE?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
MODE FSTART

5.25 SERIALbus グループ

:SERIALbus<x>:I2C?

機能 I2C バス信号の解析 / 検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:I2C?
<x> = 1 ~ 4

:SERIALbus<x>:I2C:ANALyze?

機能 I2C バス信号解析に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:I2C:ANALyze?
<x> = 1 ~ 4

:SERIALbus<x>:I2C[:ANALyze]:SETup?

機能 I2C バス信号解析のバス設定に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:I2C[:ANALyze]:SETup?
<x> = 1 ~ 4

:SERIALbus<x>:I2C[:ANALyze]:SETup:CLOCK?

機能 I2C バス信号解析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 :SERIALbus<x>:I2C[:ANALyze]:SETup:CLOCK?
<x> = 1 ~ 4

:SERIALbus<x>:I2C[:ANALyze]:SETup:CLOCK:SOURCE

機能 I2C バス信号解析のクロックを設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C[:ANALyze]:SETup:CLOCK:SOURCE {<NRf>|MATH<y>|PODA<y>|PODB<y>|PODC<y>|PODD<y>}
:SERIALbus<x>:I2C[:ANALyze]:SETup:CLOCK:SOURCE?
<x> = 1 ~ 4
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<y>、PODB<y>、PODC<y>、PODD<y> の <y> = 0 ~ 7

例 :SERIALBUS1:I2C:ANALYZE:SETUP:CLOCK:SOURCE 1
:SERIALBUS1:I2C:ANALYZE:SETUP:CLOCK:SOURCE?
-> :SERIALBUS1:I2C:ANALYZE:SETUP:CLOCK:SOURCE 1

:SERIALbus<x>:I2C[:ANALyze]:SETup:DATA?

機能 I2C バス信号解析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 :SERIALbus<x>:I2C[:ANALyze]:SETup:DATA?
<x> = 1 ~ 4

:SERIALbus<x>:I2C[:ANALyze]:SETup:DATA:SOURCE

機能 I2C バス信号解析のデータを設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C[:ANALyze]:SETup:DATA:SOURCE {<NRf>|MATH<y>|PODA<y>|PODB<y>|PODC<y>|PODD<y>}
:SERIALbus<x>:I2C[:ANALyze]:SETup:DATA:SOURCE?
<x> = 1 ~ 4
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<y>、PODB<y>、PODC<y>、PODD<y> の <y> = 0 ~ 7

例 :SERIALBUS1:I2C:ANALYZE:SETUP:DATA:SOURCE 1
:SERIALBUS1:I2C:ANALYZE:SETUP:DATA:SOURCE?
-> :SERIALBUS1:I2C:ANALYZE:SETUP:DATA:SOURCE 1

:SERIALbus<x>:I2C[:ANALyze]:SETup:INCLUDerw

機能 I2C バス信号解析のアドレス R/W の有効 (ON)/ 無効 (OFF) を設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C[:ANALyze]:SETup:INCLUDerw {<Boolean>}
:SERIALbus<x>:I2C[:ANALyze]:SETup:INCLUDerw?
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:ANALYZE:SETUP:INCLUDerw ON
:SERIALBUS1:I2C:ANALYZE:SETUP:INCLUDerw?
-> :SERIALBUS1:I2C:ANALYZE:SETUP:INCLUDerw 1

:SERIALbus<x>:I2C:DETail?

機能 I2C バス信号の解析結果リストに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:I2C:DETail?
<x> = 1 ~ 4

:SERIALbus<x>:I2C:DETAil:DISPlay

機能 I2C バス信号の解析結果リストの表示モードを設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:DETAil:
DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:I2C:DETAil:DISPlay?
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:DETAIL:DISPLAY FULL
:SERIALBUS1:I2C:DETAIL:DISPLAY?
-> :SERIALBUS1:I2C:DETAIL:
DISPLAY FULL

:SERIALbus<x>:I2C:DETAil:LIST:ITEM?

機能 I2C バス信号の解析結果リストに表示される項目を問い合わせます。

構文 :SERIALbus<x>:I2C:DETAil:LIST:ITEM?
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:DETAIL:LIST:ITEM?
-> :SERIALBUS1:I2C:DETAIL:LIST:
ITEM "No.,Time(ms),1st,2nd,R/W,Data,
Information,"

:SERIALbus<x>:I2C:DETAil:LIST:VALue?

機能 I2C バス信号の解析結果リストの指定した解析番号の全データを問い合わせます。

構文 :SERIALbus<x>:I2C:DETAil:LIST:
VALue? {<NRf>|MAXimum|MINimum}
<x> = 1 ~ 4
<NRf> = -299999 ~ 299999

例 :SERIALBUS1:I2C:DETAIL:LIST:VALUE? 0
-> :SERIALBUS1:I2C:DETAIL:LIST:
VALUE "0,-0.07000,AA*,W,AE*88*,
7-bit,"

:SERIALbus<x>:I2C:SEARCh?

機能 I2C バス信号検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:I2C:SEARCh?
<x> = 1 ~ 4

:SERIALbus<x>:I2C:SEARCh:ABORt

機能 I2C バス信号検索を中止します。

構文 :SERIALbus<x>:I2C:SEARCh:ABORt
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:ABORT

:SERIALbus<x>:I2C:SEARCh:EXECute

機能 I2C バス信号検索を実行します。

構文 :SERIALbus<x>:I2C:SEARCh:EXECute
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:EXECUTE

:SERIALbus<x>:I2C:SEARCh:SELEct

機能 I2C バス信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。

構文 :SERIALbus<x>:I2C:SEARCh:
SELEct {<NRf>|MAXimum}
:SERIALbus<x>:I2C:SEARCh:SELEct?
<x> = 1 ~ 4
<NRf> = 0 ~ 49999

例 :SERIALBUS1:I2C:SEARCH:SELECT 1
:SERIALBUS1:I2C:SEARCH:SELECT?
-> :SERIALBUS1:I2C:SEARCH:
SELECT 1.50000000

解説 検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:SERIALbus<x>:I2C:SEARCh:SETup?

機能 I2C バス信号検索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 :SERIALbus<x>:I2C:SEARCh:SETup?
<x> = 1 ~ 4

:SERIALbus<x>:I2C:SEARCh[:SETup]:ADD Ress?

機能 I2C バス信号検索のアドレス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 :SERIALbus<x>:I2C:SEARCh[:SETup]:
ADDRess?
<x> = 1 ~ 4

:SERIALbus<x>:I2C:SEARCh[:SETup]:ADD Ress:ADDRess?

機能 I2C バス信号検索のアドレスパターン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 :SERIALbus<x>:I2C:SEARCh[:SETup]:
ADDRess:ADDRess?
<x> = 1 ~ 4

:SERIALbus<x>:I2C:SEARCh[:SETup]:ADD Ress:ADDRess:BIT10Address?

機能 I2C バス信号検索の 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 :SERIALbus<x>:I2C:SEARCh[:SETup]:
ADDRess:ADDRess:BIT10Address?
<x> = 1 ~ 4

5.25 SERIALbus グループ

:SERIALbus<x>:I2C:SEARCH[:SETup]:ADDRESS:BIT10ADDRESS:DIRection

機能 I2C バス信号検索の 10bit アドレスのデータ方向を設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETup]:
ADDRESS:ADDRESS:BIT10ADDRESS:
DIRection {READ|WRITE|X}
:SERIALbus<x>:I2C:SEARCH[:SETup]:
ADDRESS:ADDRESS:BIT10ADDRESS:
DIRection?
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT10ADDRESS:
DIRECTION READ
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT10ADDRESS:
DIRECTION?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT10ADDRESS:
DIRECTION READ

:SERIALbus<x>:I2C:SEARCH[:SETup]:ADDRESS:BIT10ADDRESS:HEXa

機能 I2C バス信号検索の 10bit アドレスを 16 進数で設定します。

構文 :SERIALbus<x>:I2C:SEARCH[:SETup]:
ADDRESS:ADDRESS:BIT10ADDRESS:
HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT10ADDRESS:
HEXa "1AB"

:SERIALbus<x>:I2C:SEARCH[:SETup]:ADDRESS:BIT10ADDRESS:PATtern

機能 I2C バス信号検索の 10bit アドレスを 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETup]:
ADDRESS:ADDRESS:BIT10ADDRESS:
PATtern {<文字列>}
:SERIALbus<x>:I2C:SEARCH[:SETup]:
ADDRESS:ADDRESS:BIT10ADDRESS:
PATtern?
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT10ADDRESS:
PATTERN "11010111001"
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT10ADDRESS:
PATTERN?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT10ADDRESS:
PATTERN "11010111001"

:SERIALbus<x>:I2C:SEARCH[:SETup]:ADDRESS:BIT7ADDRESS?

機能 I2C バス信号検索の 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 :SERIALbus<x>:I2C:SEARCH[:SETup]:
ADDRESS:ADDRESS:BIT7ADDRESS?
<x> = 1 ~ 4

:SERIALbus<x>:I2C:SEARCH[:SETup]:ADDRESS:BIT7ADDRESS:DIRection

機能 I2C バス信号検索の 7bit アドレスのデータ方向を設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETup]:
ADDRESS:ADDRESS[:BIT7ADDRESS]:
DIRection {READ|WRITE|X}
:SERIALbus<x>:I2C:SEARCH[:SETup]:
ADDRESS:ADDRESS[:BIT7ADDRESS]:
DIRection?
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7ADDRESS:
DIRECTION READ
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7ADDRESS:
DIRECTION?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7ADDRESS:
DIRECTION READ

:SERIALbus<x>:I2C:SEARCH[:SETup]:ADDRESS:BIT7ADDRESS:HEXa

機能 I2C バス信号検索の 7bit アドレスを 16 進数で設定します。

構文 :SERIALbus<x>:I2C:SEARCH[:SETup]:
ADDRESS:ADDRESS[:BIT7ADDRESS]:
HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7ADDRESS:
HEXa "5C"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS[:BIT7ADDRESS]:PATTERN

機能 I2C バス信号検索の 7bit アドレスを 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS[:BIT7ADDRESS]:
PATTERN {<文字列>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS[:BIT7ADDRESS]:
PATTERN?

例 <x> = 1 ~ 4
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7ADDRESS:
PATTERN "11100110"
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7ADDRESS:PATTERN?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7ADDRESS:
PATTERN "11100110"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:BIT7APsub?

機能 I2C バス信号検索の 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 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub?
<x> = 1 ~ 4

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:BIT7APsub:ADDRESS?

機能 I2C バス信号検索の 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 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS?
<x> = 1 ~ 4

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:BIT7APsub:ADDRESS:DIRECTION

機能 I2C バス信号検索の 7bit+Sub アドレスのデータ方向を設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
DIRECTION {READ|WRITE|X}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
DIRECTION?

例 <x> = 1 ~ 4
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:ADDRESS:
DIRECTION READ
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:ADDRESS:
DIRECTION?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:ADDRESS:
DIRECTION READ

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:BIT7APsub:ADDRESS:HEXa

機能 I2C バス信号検索の 7bit+Sub アドレスの 7bit アドレスを 16 進数で設定します。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:ADDRESS:
HEXa "AB"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:BIT7APsub:ADDRESS:PATTERN

機能 I2C バス信号検索の 7bit+Sub アドレスの 7bit アドレスを 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
PATTERN {<文字列>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
PATTERN?

例 <x> = 1 ~ 4
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:ADDRESS:
PATTERN "00111010"
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:ADDRESS:
PATTERN?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:ADDRESS:
PATTERN "00111010"

5.25 SERIALbus グループ

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:BIT7APsub:SADDRESS?

機能 I2C バス信号検索の 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 :SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:BIT7APsub:SADDRESS?<x> = 1 ~ 4

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:BIT7APsub:SADDRESS:HEXA

機能 I2C バス信号検索の 7bit+Sub アドレスの Sub アドレスを 16 進数で設定します。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:BIT7APsub:SADDRESS:HEXA {<文字列>}<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:ADDRESS:ADDRESS:BIT7APSUB:SADDRESS:HEXA "EF"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:BIT7APsub:SADDRESS:PATTERN

機能 I2C バス信号検索の 7bit+Sub アドレスの Sub アドレスを 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:BIT7APsub:SADDRESS:PATTERN {<文字列>}:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:BIT7APsub:SADDRESS:PATTERN?<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:ADDRESS:ADDRESS:BIT7APSUB:SADDRESS:PATTERN "00111010":SERIALBUS1:I2C:SEARCH:SETUP:ADDRESS:ADDRESS:BIT7APSUB:SADDRESS:PATTERN?-> :SERIALBUS1:I2C:SEARCH:SETUP:ADDRESS:ADDRESS:BIT7APSUB:SADDRESS:PATTERN "00111010"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:MODE

機能 I2C バス信号検索のアドレス条件の有効 / 無効を設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:MODE {<Boolean>}:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:MODE?<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:ADDRESS:ADDRESS:MODE ON:SERIALBUS1:I2C:SEARCH:SETUP:ADDRESS:ADDRESS:MODE?-> :SERIALBUS1:I2C:SEARCH:SETUP:ADDRESS:ADDRESS:MODE 1

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:TYPE

機能 I2C バス信号検索のアドレス条件のアドレス形式を設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:TYPE {BIT10Address|BIT7Address|BIT7APsub}:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:TYPE?<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:ADDRESS:ADDRESS:TYPE BIT10ADDRESS:SERIALBUS1:I2C:SEARCH:SETUP:ADDRESS:ADDRESS:TYPE?-> :SERIALBUS1:I2C:SEARCH:SETUP:ADDRESS:ADDRESS:TYPE BIT10ADDRESS

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:DATA?

機能 I2C バス信号検索のアドレスパターン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 :SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:DATA?<x> = 1 ~ 4

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRess:DATA:BCOUNT

機能 I2C バス信号検索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
 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
 ADDRess:DATA:BCOUNT {<Nrf>}
 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
 ADDRess:DATA:BCOUNT?
 <x> = 1 ~ 4
 <Nrf> = 0 ~ 9999

例
 :SERIALBUS1:I2C:SEARCH:SETUP:
 ADDRESS:DATA:BCOUNT 0
 :SERIALBUS1:I2C:SEARCH:SETUP:
 ADDRESS:DATA:BCOUNT?
 -> :SERIALBUS1:I2C:SEARCH:SETUP:
 ADDRESS:DATA:BCOUNT 0

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRess:DATA:BMODE

機能 I2C バス信号検索のデータのパターンを比較する位置の ON/OFF を設定 / 問い合わせします。

構文
 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
 ADDRess:DATA:BMODE {<Boolean>}
 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
 ADDRess:DATA:BMODE?
 <x> = 1 ~ 4

例
 :SERIALBUS1:I2C:SEARCH:SETUP:
 ADDRESS:DATA:BMODE ON
 :SERIALBUS1:I2C:SEARCH:SETUP:
 ADDRESS:DATA:BMODE?
 -> :SERIALBUS1:I2C:SEARCH:SETUP:
 ADDRESS:DATA:BMODE 1

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRess:DATA:CONDition

機能 I2C バス信号検索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
 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
 ADDRess:DATA:CONDition {FALSe|TRUE}
 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
 ADDRess:DATA:CONDition?
 <x> = 1 ~ 4

例
 :SERIALBUS1:I2C:SEARCH:SETUP:
 ADDRESS:DATA:CONDITION FALSE
 :SERIALBUS1:I2C:SEARCH:SETUP:
 ADDRESS:DATA:CONDITION?
 -> :SERIALBUS1:I2C:SEARCH:SETUP:
 ADDRESS:DATA:CONDITION FALSE

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRess:DATA:DBYTE

機能 I2C バス信号検索の判定データ数を設定 / 問い合わせします。

構文
 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
 ADDRess:DATA:DBYTE {<Nrf>}
 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
 ADDRess:DATA:DBYTE?
 <x> = 1 ~ 4
 <Nrf> = 1 ~ 4

例
 :SERIALBUS1:I2C:SEARCH:SETUP:
 ADDRESS:DATA:DBYTE 1
 :SERIALBUS1:I2C:SEARCH:SETUP:
 ADDRESS:DATA:DBYTE?
 -> :SERIALBUS1:I2C:SEARCH:SETUP:
 ADDRESS:DATA:DBYTE 1

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRess:DATA:DPATtern?

機能 I2C バス信号検索のデータ条件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
 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
 ADDRess:DATA:DPATtern?
 <x> = 1 ~ 4

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRess:DATA:DPATtern:HEXa

機能 I2C バス信号検索のデータ条件のデータを 16 進数で設定します。

構文
 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
 ADDRess:DATA:DPATtern:HEXa {<文字列>}
 <x> = 1 ~ 4

例
 :SERIALBUS1:I2C:SEARCH:SETUP:
 ADDRESS:DATA:DPATTERN:HEXA "AB"

解説
 「:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRess:DATA:DBYTE」の設定値によって設定できるバイト数変動します。

5.25 SERIALbus グループ

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:DATA:DPATTERN:PATTERN

機能 I2C バス信号検索のデータ条件のデータを 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:DATA:DPATTERN:
PATTERN {<文字列>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:DATA:DPATTERN:PATTERN?
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:DPATTERN:
PATTERN "10001101"
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:DPATTERN:PATTERN?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:DPATTERN:
PATTERN "10001101"

解説 「:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:DATA:DBYTE」の設定値によって設定できるバイト数が変動します。

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:DATA:MODE

機能 I2C バス信号検索のデータ条件の有効 / 無効を設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:DATA:MODE {<Boolean>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:DATA:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:MODE ON
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:MODE?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:MODE 1

:SERIALbus<x>:I2C:SEARCH[:SETUP]:GENERALCALL?

機能 I2C バス信号検索のゼ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 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
GENERALCALL?
<x> = 1 ~ 4

:SERIALbus<x>:I2C:SEARCH[:SETUP]:GENERALCALL:BIT7MADDRESS?

機能 I2C バス信号検索のゼネラルコールの 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 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
GENERALCALL:BIT7MADDRESS?
<x> = 1 ~ 4

:SERIALbus<x>:I2C:SEARCH[:SETUP]:GENERALCALL:BIT7MADDRESS:HEXA

機能 I2C バス信号検索のゼネラルコールの 7bit マスタアドレスを 16 進数で設定します。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
GENERALCALL:BIT7MADDRESS:HEXA{<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:
GENERALCALL:BIT7MADDRESS:HEXA "AB"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:GENERALCALL:BIT7MADDRESS:PATTERN

機能 I2C バス信号検索のゼネラルコールの 7bit マスタアドレスを 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
GENERALCALL:BIT7MADDRESS:
PATTERN{<文字列>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
GENERALCALL:BIT7MADDRESS:PATTERN?
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:
GENERALCALL:BIT7MADDRESS:
PATTERN "0010110"
:SERIALBUS1:I2C:SEARCH:SETUP:
GENERALCALL:BIT7MADDRESS:PATTERN?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
GENERALCALL:BIT7MADDRESS:
PATTERN "0010110"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:GENERALCALL:SBYTE

機能 I2C バス信号検索のゼ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 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
GENERALCALL:SBYTE {BIT7MADDRESS|
DONTcare|H04|H06}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
GENERALCALL:SBYTE?
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:
GENERALCALL:SBYTE BIT7MADDRESS
:SERIALBUS1:I2C:SEARCH:SETUP:
GENERALCALL:SBYTE?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
GENERALCALL:SBYTE BIT7MADDRESS

:SERIALbus<x>:I2C:SEARCH[:SETUP]:MODE

機能 I2C バス信号検索の検索の種類を設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
MODE {ADDRData|EVERystart|
GENeralcall|HSMode|NONack|STARTbyte}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:
MODE ADDRDATA
:SERIALBUS1:I2C:SEARCH:SETUP:MODE?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
MODE ADDRDATA

:SERIALbus<x>:I2C:SEARCH[:SETUP]:NONack?

機能 I2C バス信号検索の 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 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
NONack?
<x> = 1 ~ 4

:SERIALbus<x>:I2C:SEARCH[:SETUP]:NONack:HSMODE

機能 I2C バス信号検索のハイスピードモードで NONACK を無視する / しないを設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
NONack:HSMODE {<Boolean>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
NONack:HSMODE?
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:NONACK:
HSMODE ON
:SERIALBUS1:I2C:SEARCH:SETUP:NONACK:
HSMODE?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
NONACK:HSMODE 1

:SERIALbus<x>:I2C:SEARCH[:SETUP]:NONack:READAccess

機能 I2C バス信号検索のリードアクセスモードで NONACK を無視する / しないを設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
NONack:READAccess {<Boolean>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
NONack:READAccess?
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:NONACK:
READACCESS ON
:SERIALBUS1:I2C:SEARCH:SETUP:NONACK:
READACCESS?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
NONACK:READACCESS 1

:SERIALbus<x>:I2C:SEARCH[:SETUP]:NONack:STARTbyte

機能 I2C バス信号検索のスタートバイトで NONACK を無視する / しないを設定 / 問い合わせします。

構文 :SERIALbus<x>:I2C:SEARCH[:SETUP]:
NONack:STARTbyte {<Boolean>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
NONack:STARTbyte?
<x> = 1 ~ 4

例 :SERIALBUS1:I2C:SEARCH:SETUP:NONACK:
STARTBYTE ON
:SERIALBUS1:I2C:SEARCH:SETUP:NONACK:
STARTBYTE?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
NONACK:STARTBYTE 1

:SERIALbus<x>:LIN?

機能 LIN バス信号の解析 / 検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:LIN?
<x> = 1 ~ 4

:SERIALbus<x>:LIN:ANALyze?

機能 LIN バス信号解析に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:LIN:ANALyze?
<x> = 1 ~ 4

:SERIALbus<x>:LIN[:ANALyze]:SETup?

機能 LIN バス信号解析のバス設定に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:LIN[:ANALyze]:SETup?
<x> = 1 ~ 4

5.25 SERIALbus グループ

:SERIALbus<x>:LIN[:ANALyze]:SETup:BRATe

機能 LIN バス信号解析のビットレート (データ転送速度) を設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN[:ANALyze]:SETup:
BRATe {<Nrf>|USER,<Nrf>}
:SERIALbus<x>:LIN[:ANALyze]:SETup:
BRATe?

例 <x> = 1 ~ 4
<Nrf> = 1200、2400、4800、9600、19200
USER の <Nrf> = 1000bps ~ 20000bps
:SERIALBUS1:LIN:ANALYZE:SETUP:
BRATE 2400
:SERIALBUS1:LIN:ANALYZE:SETUP:BRATE?
-> :SERIALBUS1:LIN:ANALYZE:SETUP:
BRATE 2400

:SERIALbus<x>:LIN[:ANALyze]:SETup:REVision

機能 LIN バス信号解析のレビジョン番号を設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN[:ANALyze]:SETup:
REvision {BOTH|LIN1 _ 3|LIN2 _ 0}
:SERIALbus<x>:LIN[:ANALyze]:SETup:
REvision?

例 <x> = 1 ~ 4
:SERIALBUS1:LIN:ANALYZE:SETUP:
REVISION LIN1 _ 3
:SERIALBUS1:LIN:ANALYZE:SETUP:
REVISION?
-> :SERIALBUS1:LIN:ANALYZE:SETUP:
REVISION LIN1 _ 3

:SERIALbus<x>:LIN[:ANALyze]:SETup:SOURce

機能 LIN バス信号解析のソースを設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN[:ANALyze]:SETup:
SOURce {<Nrf>|MATH<y>}
:SERIALbus<x>:LIN[:ANALyze]:SETup:
SOURce?

例 <x> = 1 ~ 4
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
:SERIALBUS1:LIN:ANALYZE:SETUP:
SOURCE 1
:SERIALBUS1:LIN:ANALYZE:SETUP:
SOURCE?
-> :SERIALBUS1:LIN:ANALYZE:SETUP:
SOURCE 1

:SERIALbus<x>:LIN[:ANALyze]:SETup:SPOint

機能 LIN バス信号解析のサンプルポイントを設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN[:ANALyze]:SETup:
SPOint {<Nrf>}
:SERIALbus<x>:LIN[:ANALyze]:SETup:
SPOint?

例 <x> = 1 ~ 4
<Nrf> = 18.8 ~ 90.6
:SERIALBUS1:LIN:ANALYZE:SETUP:
SPOINT 18.8
:SERIALBUS1:LIN:ANALYZE:SETUP:
SPOINT?
-> :SERIALBUS1:LIN:ANALYZE:SETUP:
SPOINT 18.8

:SERIALbus<x>:LIN:DETAil?

機能 LIN バス信号の解析結果リストに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:LIN:DETAil?
<x> = 1 ~ 4

:SERIALbus<x>:LIN:DETAil:DISPlay

機能 LIN バス信号の解析結果リストの表示モードを設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:DETAil:
DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:LIN:DETAil:DISPlay?
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:DETAIL:DISPLAY FULL
:SERIALBUS1:LIN:DETAIL:DISPLAY?
-> :SERIALBUS1:LIN:DETAIL:
DISPLAY FULL

:SERIALbus<x>:LIN:DETAil:LIST:ITEM?

機能 LIN バス信号の解析結果リストに表示される項目を問い合わせます。

構文 :SERIALbus<x>:LIN:DETAil:LIST:ITEM?
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:DETAIL:LIST:ITEM?
-> :SERIALBUS1:LIN:DETAIL:LIST:
ITEM "No.,Time(ms),ID,ID-Field,Data,
Checksum,Information"

:SERIALbus<x>:LIN:DETAil:LIST:VALue?

機能 LIN バス信号の解析結果リストの指定した解析番号の全データを問い合わせます。

構文 :SERIALbus<x>:LIN:DETAil:LIST:
VALue? {<NRf>|MAXimum|MINimum}
<x> = 1 ~ 4
<NRf> = -99999 ~ 99999

例 :SERIALBUS1:LIN:DETAil:LIST:VALUE? 0
-> :SERIALBUS1:LIN:DETAil:LIST:
VALUE "0,-1,0016,30,F0,2D04,CE,, "

:SERIALbus<x>:LIN:SEARCh?

機能 LIN バス信号検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:LIN:SEARCh?
<x> = 1 ~ 4

:SERIALbus<x>:LIN:SEARCh:ABORt

機能 LIN バス信号検索を中止します。

構文 :SERIALbus<x>:LIN:SEARCh:ABORt
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCh:ABORt

:SERIALbus<x>:LIN:SEARCh:EXECute

機能 LIN バス信号検索を実行します。

構文 :SERIALbus<x>:LIN:SEARCh:EXECute
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCh:EXECUTE

:SERIALbus<x>:LIN:SEARCh:FJUMp:BREAk

機能 LIN バス信号検索の結果の Break Field へのフィールドジャンプを実行します。

構文 :SERIALbus<x>:LIN:SEARCh:FJUMp:BREAk
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCh:FJUMp:BREAk

:SERIALbus<x>:LIN:SEARCh:FJUMp:CSUM

機能 LIN バス信号検索の結果の Checksum Field へのフィールドジャンプを実行します。

構文 :SERIALbus<x>:LIN:SEARCh:FJUMp:CSUM
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCh:FJUMp:CSUM

:SERIALbus<x>:LIN:SEARCh:FJUMp:DATA

機能 LIN バス信号検索の結果の Data Field へのフィールドジャンプを実行します。

構文 :SERIALbus<x>:LIN:SEARCh:FJUMp:DATA
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCh:FJUMp:DATA

:SERIALbus<x>:LIN:SEARCh:FJUMp:IDENTifier

機能 LIN バス信号検索の結果の Identifier Field へのフィールドジャンプを実行します。

構文 :SERIALbus<x>:LIN:SEARCh:FJUMp:
IDENTifier
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCh:FJUMp:
IDENTIFIER

:SERIALbus<x>:LIN:SEARCh:FJUMp:SYNCh

機能 LIN バス信号検索の結果の Sync Field へのフィールドジャンプを実行します。

構文 :SERIALbus<x>:LIN:SEARCh:FJUMp:SYNCh
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCh:FJUMp:SYNCh

:SERIALbus<x>:LIN:SEARCh:SELEct

機能 LIN バス信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。

構文 :SERIALbus<x>:LIN:SEARCh:
SELEct {<NRf>|MAXimum}
:SERIALbus<x>:LIN:SEARCh:SELEct?
<x> = 1 ~ 4
<NRf> = 0 ~ 49999

例 :SERIALBUS1:LIN:SEARCh:SELEct 1
:SERIALBUS1:LIN:SEARCh:SELEct?
-> :SERIALBUS1:LIN:SEARCh:
SELEct 1.50000000

解説 検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:SERIALbus<x>:LIN:SEARCh:SELEct?**MAXimum**

機能 LIN バス信号検索の検索点の総数を問い合わせます。

構文 :SERIALbus<x>:LIN:SEARCh:
SELEct? MAXimum
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCh:
SELEct? MAXIMUM
-> :SERIALBUS1:LIN:SEARCh:SELEct 100

解説 検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:SERIALbus<x>:LIN:SEARCh:SETUp?

機能 LIN バス信号検索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 :SERIALbus<x>:LIN:SEARCh:SETUp?
<x> = 1 ~ 4

5.25 SERIALbus グループ

:SERIALbus<x>:LIN:SEARCH:SETup:ERROR?

機能 LIN バス信号検索の Error に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
ERROR?
<x> = 1 ~ 4

:SERIALbus<x>:LIN:SEARCH:SETup:ERROR:CHECKsum

機能 LIN バス信号検索の Checksum Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:CHECKsum {<Boolean>}
:SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:CHECKsum?
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
CHECKSUM ON
:SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
CHECKSUM?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
ERROR:CHECKSUM 1

:SERIALbus<x>:LIN:SEARCH:SETup:ERROR:FRAMing

機能 LIN バス信号検索の Framing Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:FRAMing {<Boolean>}
:SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:FRAMing?
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
FRAMING ON
:SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
FRAMING?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
ERROR:FRAMING 1

:SERIALbus<x>:LIN:SEARCH:SETup:ERROR:PARity

機能 LIN バス信号検索の Parity Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:PARity {<Boolean>}
:SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:PARity?
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
PARITY ON
:SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
PARITY?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
ERROR:PARITY 1

:SERIALbus<x>:LIN:SEARCH:SETup:ERROR:SYNCh

機能 LIN バス信号検索の Synch Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:SYNCh {<Boolean>}
:SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:SYNCh?
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
SYNCH ON
:SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
SYNCH?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
ERROR:SYNCH 1

:SERIALbus<x>:LIN:SEARCH:SETup:ERROR:TIMEout

機能 LIN バス信号検索の Timeout Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:TIMEout {<Boolean>}
:SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:TIMEout?
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
TIMEOUT ON
:SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
TIMEOUT?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
ERROR:TIMEOUT 1

:SERIALbus<x>:LIN:SEARCH:SETup:IDData?

機能 LIN バス信号検索の ID とデータ条件に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
IDData?
<x> = 1 ~ 4

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:DATA?

機能 LIN バス信号検索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 :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA?
<x> = 1 ~ 4

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:DATA:CONDition

機能 LIN バス信号検索のデータ条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:CONDition {BETween|
EQUAL|FALSE|GREATER|LESS|NOTBetween|
NOTEQUAL|TRUE}
:SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:CONDition?
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:CONDITION BETWEEN
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:CONDITION?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:CONDITION BETWEEN

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:DATA:DBYTE

機能 LIN バス信号検索のデータのバイト数を設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:DBYTE {<NRF>}
:SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:DBYTE?
<x> = 1 ~ 4
<NRF> = 1 ~ 8

例 :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:DBYTE 1
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:DBYTE?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:DBYTE 1

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:DATA:DECimal<y>

機能 LIN バス信号検索のデータを 10 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:DECimal<y> {<NRF>}
:SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:DECimal<y>?
<x> = 1 ~ 4
<y> = 1, 2
<NRF> = -9E+18 ~ 9E+18

例 :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:DECIMAL1 1
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:DECIMAL1?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1 1

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:DATA:DPATtern?

機能 LIN バス信号検索のデータに関わるすべての設定値を問い合わせます。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:DPATtern?
<x> = 1 ~ 4

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:DATA:DPATtern:HEXa

機能 LIN バス信号検索のデータを 16 進数で設定します。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:DPATtern:HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:DPATTERN:HEXA "12"

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:DATA:DPATtern:PATtern

機能 LIN バス信号検索のデータを 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:DPATtern:
PATtern {<文字列>}
:SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:DPATtern:PATtern?
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:DPATTERN:PATTERN "00110101"
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:DPATTERN:PATTERN?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:
PATTERN "00110101"

5.25 SERIALbus グループ

:SERIALbus<x>:LIN:SEARCh:SETup:IDData:DATA:ENDian

機能 LIN バス信号検索のデータのエンディアンを設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCh:SETup:
IDData:DATA:ENDian {BIG|LITTLe}
:SERIALbus<x>:LIN:SEARCh:SETup:
IDData:DATA:ENDian?
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:ENDIAN BIG
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:ENDIAN?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:ENDIAN BIG

:SERIALbus<x>:LIN:SEARCh:SETup:IDData:DATA:MODE

機能 LIN バス信号検索のデータ条件 (有効 / 無効) の設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCh:SETup:
IDData:DATA:MODE {<Boolean>}
:SERIALbus<x>:LIN:SEARCh:SETup:
IDData:DATA:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:MODE ON
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:MODE?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:MODE 1

:SERIALbus<x>:LIN:SEARCh:SETup:IDData:DATA:MSBLsb

機能 LIN バス信号検索のデータの MSB/LSB のビットを設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCh:SETup:
IDData:DATA:MSBLsb {<NRf>,<NRf>}
:SERIALbus<x>:LIN:SEARCh:SETup:
IDData:DATA:MSBLsb?
<x> = 1 ~ 4
<NRf> = 0 ~ 63

例 :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:MSBLSB 1,0
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:MSBLSB?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:MSBLSB 1,0

:SERIALbus<x>:LIN:SEARCh:SETup:IDData:DATA:SIGN

機能 LIN バス信号検索のデータの符号を設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCh:SETup:
IDData:DATA:SIGN {SIGN|UNSign}
:SERIALbus<x>:LIN:SEARCh:SETup:
IDData:DATA:SIGN?
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:SIGN SIGN
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:SIGN?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:SIGN SIGN

:SERIALbus<x>:LIN:SEARCh:SETup:IDData:IDENTifier?

機能 LIN バス信号検索の識別子に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:LIN:SEARCh:SETup:
IDData:IDENTifier?
<x> = 1 ~ 4

:SERIALbus<x>:LIN:SEARCh:SETup:IDData:IDENTifier:ID?

機能 LIN バス信号検索の ID に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:LIN:SEARCh:SETup:
IDData:IDENTifier:ID?
<x> = 1 ~ 4

:SERIALbus<x>:LIN:SEARCh:SETup:IDData:IDENTifier:ID:HEXa

機能 LIN バス信号検索の ID を 16 進数で設定します。

構文 :SERIALbus<x>:LIN:SEARCh:SETup:
IDData:IDENTifier:ID:HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
IDENTIFIER:ID:HEXa "1E"

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:IDENTIFIER:ID:MODE

機能 LIN バス信号検索の ID 条件 (有効 / 無効) の設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:IDENTIFIER:ID:
MODE {<Boolean>}

:SERIALbus<x>:LIN:SEARCH:SETup:
IDData:IDENTIFIER:ID:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
IDENTIFIER:ID:MODE ON
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
IDENTIFIER:ID:MODE?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:IDENTIFIER:ID:MODE 1

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:IDENTIFIER:ID:PATTERN

機能 LIN バス信号検索の ID を 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:IDENTIFIER:ID:PATTERN{<文字列>}
:SERIALbus<x>:LIN:SEARCH:SETup:
IDData:IDENTIFIER:ID:PATTERN?
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
IDENTIFIER:ID:PATTERN "101100"
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
IDENTIFIER:ID:PATTERN?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:IDENTIFIER:ID:
PATTERN "101100"

:SERIALbus<x>:LIN:SEARCH:SETup:MODE

機能 LIN バス信号検索の検索の種類を設定 / 問い合わせします。

構文 :SERIALbus<x>:LIN:SEARCH:SETup:
MODE {BSYNch|ERRor|IDData}
:SERIALbus<x>:LIN:SEARCH:SETup:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:LIN:SEARCH:SETUP:
MODE BSYNCH
:SERIALBUS1:LIN:SEARCH:SETUP:MODE?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
MODE BSYNCH

:SERIALbus<x>:PSI5?

機能 PSI5 信号の解析 / 検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:PSI5?
<x> = 1 ~ 4

:SERIALbus<x>:PSI5:ANALyze?

機能 PSI5 信号解析に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:PSI5:ANALyze?
<x> = 1 ~ 4

:SERIALbus<x>:PSI5[:ANALyze]:SETup?

機能 PSI5 信号解析のバス設定に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:PSI5[:ANALyze]:SETup?
<x> = 1 ~ 4

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA?

機能 PSI5 信号解析のデータ信号に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA?
<x> = 1 ~ 4

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:BRATE

機能 PSI5 信号解析のビットレート (データ転送速度) を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA:BRATE {<Nrf>|USER,<Nrf>}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA:BRATE?
<x> = 1 ~ 4

<Nrf> = 125000、189000
USER の <Nrf> = 10000 ~ 1000000
例 :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:
BRATE 125000
:SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:
BRATE?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:
DATA:BRATE 125000

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:CTOLerance

機能 PSI5 信号解析のクロック許容範囲を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA:CTOLerance {<Nrf>}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA:CTOLerance?
<x> = 1 ~ 4

<Nrf> = ± 0.5% ~ ± 33.3% (設定分解能 0.1%)
例 :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:
CTOLERANCE 5
:SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:
CTOLERANCE?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:
DATA:CTOLERANCE 5.00E+00

5.25 SERIALbus グループ

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:DBITs

機能 PSI5 信号解析のデータ長を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA:DBITs {<NRf>}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA:DBITs?
<x> = 1 ~ 4
<NRf> = 10, 16

例 :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:
DBITS 10
:SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:
DBITS?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:
DATA:DBITS 10

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:EDETection

機能 PSI5 信号解析のエラー検出方式を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA:EDETection {CRC|PARity}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA:EDETection?
<x> = 1 ~ 4

例 :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:
EDETECTION CRC
:SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:
EDETECTION?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:
DATA:EDETECTION CRC

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:SNRejection?

機能 PSI5 信号解析の同期ノイズ除去のすべての設定値を問い合わせます。

構文 :SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA:SNRejection?
<x> = 1 ~ 4

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:SNRejection:ETIme

機能 PSI5 信号解析の同期ノイズ除去の終端時間を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA:SNRejection:ETIme {<NRf>}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA:SNRejection:ETIme?
<x> = 1 ~ 4
<NRf> = 0 ~ 0.02s(0.1us ステップ)

例 :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:
SNREJECTION:ETIME 44.0000E-06
:SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:
SNREJECTION:ETIME?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:
DATA:SNREJECTION:ETIME 44.0000E-06

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:SNRejection:MODE

機能 PSI5 信号解析の同期ノイズ除去の ON/OFF を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA:SNRejection:MODE {<Boolean>}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA:SNRejection:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:
SNREJECTION:MODE ON
:SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:
SNREJECTION:MODE?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:
DATA:SNREJECTION:MODE 1

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:SOURce

機能 PSI5 信号解析のデータソースを設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA:SOURce {<NRf>|MATH<y>}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:
DATA:SOURce?
<x> = 1 ~ 4
<NRf> = 1 ~ 8 (4ch モデルでは 1 ~ 4)
<y> = 1 ~ 8 (4ch モデルでは 1 ~ 4)

例 :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:
SOURCE 1
:SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:
SOURCE?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:
DATA:SOURCE 1

:SERIALbus<x>:PSI5[:ANALyze]:SETup:NUMBERofslot

機能 PSI5 信号解析のスロット数を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5[:ANALyze]:SETup:NUMBERofslot {AUTO|<Nrf>}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:NUMBERofslot?
<x> = 1 ~ 4
<Nrf> = 1 ~ 6

例 :SERIALBUS1:PSI5:ANALYZE:SETUP:NUMBEROFSLLOT AUTO
:SERIALBUS1:PSI5:ANALYZE:SETUP:NUMBEROFSLLOT?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:NUMBEROFSLLOT AUTO

:SERIALbus<x>:PSI5[:ANALyze]:SETup:SLOT<y>

機能 PSI5 信号解析のスロットの開始 / 終了時間を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5[:ANALyze]:SETup:SLOT<y> {<Nrf>[,<Nrf>]}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:SLOT<y>?
<x> = 1 ~ 4
<y> = 1 ~ 6 (スロット番号)
<Nrf>[,<Nrf>] = 0 ~ 0.02s(0.1us ステップ)

例 :SERIALBUS1:PSI5:ANALYZE:SETUP:SLOT6 870.000E-06,1.08830E-03
:SERIALBUS1:PSI5:ANALYZE:SETUP:SLOT6?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:SLOT6 870.000E-06,1.08830E-03

解説 最初の <Nrf> にスロットの開始時間、2 つ目の <Nrf> が終了時間です。
「:SERIALbus<x>:PSI5[:ANALyze]:SETup:NUMBERofslot」が 1 ~ 6 の場合は、最後のスロット番号が <y> で指定されたときに、2 つ目の <Nrf> が有効になります。
「:SERIALbus<x>:PSI5[:ANALyze]:SETup:NUMBERofslot」が「AUTO」の場合は、問い合わせできません。

:SERIALbus<x>:PSI5[:ANALyze]:SETup:SYNC

機能 PSI5 信号解析の同期信号を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5[:ANALyze]:SETup:SYNC {<Nrf>|MATH<y>|NONE}
:SERIALbus<x>:PSI5:ANALyze:SETup:SYNC?
<x> = 1 ~ 4
<y> = 1 ~ 8 (4ch モデルでは 1 ~ 4)
<Nrf> = 1 ~ 8 (4ch モデルでは 1 ~ 4)

例 :SERIALBUS1:PSI5:ANALYZE:SETUP:SYNC 1
:SERIALBUS1:PSI5:ANALYZE:SETUP:SYNC?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:SYNC 1

:SERIALbus<x>:PSI5:DETail?

機能 PSI5 信号解析の解析リストに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:PSI5:DETail?
<x> = 1 ~ 4

:SERIALbus<x>:PSI5:DETail:DISPlay

機能 PSI5 信号解析の解析リストの表示モードを設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5:DETail:DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:PSI5:DETail:DISPlay?
<x> = 1 ~ 4

例 :SERIALBUS1:PSI5:DETAIL:DISPLAY FULL
:SERIALBUS1:PSI5:DETAIL:DISPLAY?
-> :SERIALBUS1:PSI5:DETAIL:DISPLAY FULL

:SERIALbus<x>:PSI5:DETail:LIST:ALL?

機能 PSI5 信号の解析結果リストのすべての解析番号の全データを問い合わせます。

構文 :SERIALbus<x>:PSI5:DETail:LIST:ALL?
<x> = 1 ~ 4

例 :SERIALBUS1:PSI5:DETAIL:LIST:ALL?

-> #8(8 桁のバイト数)(データ列)

解説 すべての解析番号の全データがブロックデータの形式で返されます。
解析番号ごとのデータ間は、ASCII コードの「0AH」で区切られます。

5.25 SERIALbus グループ

:SERIALbus<x>:PSI5:DETAil:LIST:ITEM?

機能 PSI5 信号の解析結果リストに表示される項目を問い合わせます。

構文 :SERIALbus<x>:PSI5:DETAil:LIST:ITEM?
<x> = 1 ~ 4

例 :SERIALBUS1:PSI5:DETAil:LIST:ITEM?
-> :SERIALBUS1:PSI5:DETAil:LIST:
ITEM "No.Time(ms)Slot No.DataParity/
CRCInformation"

:SERIALbus<x>:PSI5:DETAil:LIST:VALUE?

機能 PSI5 信号の解析結果リストの指定した解析番号の全データを問い合わせます。

構文 :SERIALbus<x>:PSI5:DETAil:LIST:
VALUE? {<NRf>|MAXimum|MINimum}
<x> = 1 ~ 4

例 :SERIALBUS1:PSI5:DETAil:LIST:
VALUE? 1
-> :SERIALBUS1:PSI5:DETAil:LIST:
VALUE "1, 0.062104,1,AB67,2,,,"

:SERIALbus<x>:PSI5:SEARch?

機能 PSI5 信号検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:PSI5:SEARch?
<x> = 1 ~ 4

:SERIALbus<x>:PSI5:SEARch:ABORT

機能 PSI5 信号検索を中止します。

構文 :SERIALbus<x>:PSI5:SEARch:ABORT
<x> = 1 ~ 4

例 :SERIALBUS1:PSI5:SEARch:ABORT

:SERIALbus<x>:PSI5:SEARch:EXECute

機能 PSI5 信号検索を実行します。

構文 :SERIALbus<x>:PSI5:SEARch:EXECute
<x> = 1 ~ 4

例 :SERIALBUS1:PSI5:SEARch:EXECUTE

:SERIALbus<x>:PSI5:SEARch:SElect

機能 PSI5 信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。

構文 :SERIALbus<x>:PSI5:SEARch:
SElect {<NRf>|MAXimum}
:SERIALbus<x>:PSI5:SEARch:SElect?
<x> = 1 ~ 4
<NRf> = 0 ~ 49999

例 :SERIALBUS1:PSI5:SEARch:SElect 1
:SERIALBUS1:PSI5:SEARch:SElect?
-> :SERIALBUS1:PSI5:SEARch:
SElect 1.50000000

解説 検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:SERIALbus<x>:PSI5:SEARch:SETup?

機能 PSI5 信号検索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 :SERIALbus<x>:PSI5:SEARch:SETup?
<x> = 1 ~ 4

:SERIALbus<x>:PSI5:SEARch[:SETup]:DATA?

機能 PSI5 信号のデータ検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:PSI5:SEARch[:SETup]:
DATA?
<x> = 1 ~ 4

:SERIALbus<x>:PSI5:SEARch[:SETup]:DATA:DATA?

機能 PSI5 信号のデータ検索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 :SERIALbus<x>:PSI5:SEARch[:SETup]:
DATA:DATA?
<x> = 1 ~ 4

:SERIALbus<x>:PSI5:SEARCh[:SETup]:DATA:DATA:CONDition

機能 PSI5 信号のデータ検索の判定条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5:SEARCh[:SETup]:
DATA:DATA:CONDition {EQUAL|FALSE|
GREATER|INSide|LESS|NOTEQUAL|
OUTSide|TRUE}
:SERIALbus<x>:PSI5:SEARCh[:SETup]:
DATA:DATA:CONDition?
<x> = 1 ~ 4

例 :SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
DATA:CONDITION EQUAL
:SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
DATA:CONDITION?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
DATA:DATA:CONDITION EQUAL

:SERIALbus<x>:PSI5:SEARCh[:SETup]:DATA:DATA:DECimal<y>

機能 PSI5 信号のデータ検索のデータを 10 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5:SEARCh[:SETup]:
DATA:DATA:DECimal<y> {<NRF>}
:SERIALbus<x>:PSI5:SEARCh[:SETup]:
DATA:DATA:DECimal<y>?
<x> = 1 ~ 4
<y> = 1, 2
<NRF> = 10 ビットデータのとき、-512 ~ 511
16 ビットデータのとき、-32768 ~ 32767

例 :SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
DATA:DECIMAL1 -10
:SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
DATA:DECIMAL1?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
DATA:DECIMAL1 -10

解説 「:SERIALbus<x>:PSI5:SEARCh[:SETup]:
DATA:DATA:CONDition」が「EQUAL」、
「GREATER」、「INSide」、「LESS」、「NOTEQUAL」、
「OUTSide」のときに有効です。

:SERIALbus<x>:PSI5:SEARCh[:SETup]:DATA:DATA:HEXa

機能 PSI5 信号のデータ検索のデータを 16 進数で設定します。

構文 :SERIALbus<x>:PSI5:SEARCh[:SETup]:
DATA:DATA:HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
DATA:HEXa "3A0F"

解説 「:SERIALbus<x>:PSI5:SEARCh[:SETup]:D
ATA:DATA:CONDition」が「TRUE」、「FALSE」
のときに有効です。

:SERIALbus<x>:PSI5:SEARCh[:SETup]:DATA:DATA:PATtern

機能 PSI5 信号のデータ検索のデータを 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5:SEARCh[:SETup]:
DATA:DATA:PATtern {<文字列>}
:SERIALbus<x>:PSI5:SEARCh[:SETup]:
DATA:DATA:PATtern?
<x> = 1 ~ 4

例 :SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
DATA:PATTERN "0011010100001111"
:SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
DATA:PATTERN?

-> :SERIALBUS1:PSI5:SEARCH:SETUP:
DATA:DATA:PATTERN "0011010100001111"

解説 「:SERIALbus<x>:PSI5:SEARCh[:SETup]:D
ATA:DATA:CONDition」が「TRUE」、「FALSE」
のときに有効です。

:SERIALbus<x>:PSI5:SEARCh[:SETup]:DATA:FRAMEinslot?

機能 PSI5 信号のデータ検索のスロット指定に関する
すべての設定値を問い合わせます。

構文 :SERIALbus<x>:PSI5:SEARCh[:SETup]:
DATA:FRAMEinslot?
<x> = 1 ~ 4

:SERIALbus<x>:PSI5:SEARCh[:SETup]:DATA:FRAMEinslot:MODE

機能 PSI5 信号のデータ検索のスロット指定条件 (有効
/ 無効) を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5:SEARCh[:SETup]:
DATA:FRAMEinslot:MODE {<Boolean>}
:SERIALbus<x>:PSI5:SEARCh[:SETup]:
DATA:FRAMEinslot:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
FRAMEINSLOT:MODE ON
:SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
FRAMEINSLOT:MODE?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
DATA:FRAMEINSLOT:MODE 1

5.25 SERIALbus グループ

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:DATA:FRAMEinslot:SNUMBER

機能 PSI5 信号のデータ検索のスロット番号を設定 / 問い合わせします。

構文
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
DATA:FRAMEinslot:SNUMBER {<NRf>}
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
DATA:FRAMEinslot:SNUMBER?
<x> = 1 ~ 4
<NRf> = 1 ~ 6

例
:SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
FRAMEINSLOT:SNUMBER 1
:SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
FRAMEINSLOT:SNUMBER?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
DATA:FRAMEINSLOT:SNUMBER 1

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:ERROR?

機能 PSI5 信号の Error 検索に関するすべての設定値を問い合わせます。

構文
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR?
<x> = 1 ~ 4

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:ERROR:FNUMBER

機能 PSI5 信号の Frame Number Error 検索を設定 / 問い合わせします。

構文
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:FNUMBER {<Boolean>}
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:FNUMBER?
<x> = 1 ~ 4

例
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
FNUMBER ON
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
FNUMBER?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
ERROR:FNUMBER 1

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:ERROR:FRAME

機能 PSI5 信号の Frame Error 検索を設定 / 問い合わせします。

構文
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:FRAME {<Boolean>}
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:FRAME?
<x> = 1 ~ 4

例
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
FRAME ON
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
FRAME?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
ERROR:FRAME 1

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:ERROR:MANCHESTER

機能 PSI5 信号の Manchester Error 検索を設定 / 問い合わせします。

構文
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:MANCHESTER {<Boolean>}
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:MANCHESTER?
<x> = 1 ~ 4

例
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
MANCHESTER ON
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
MANCHESTER?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
ERROR:MANCHESTER 1

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:ERROR:PCRC

機能 PSI5 信号の Parity/CRC Error 検索を設定 / 問い合わせします。

構文
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:PCRC {<Boolean>}
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:PCRC?
<x> = 1 ~ 4

例
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
PCRC ON
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
PCRC?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
ERROR:PCRC 1

**:SERIALbus<x>:PSI5:SEARCh[:SETup]:ER
Ror:SBIT**

機能 PSI5 信号の Start Bit Error 検索を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5:SEARCh[:SETup]:
ERROR:SBIT {<Boolean>}
:SERIALbus<x>:PSI5:SEARCh[:SETup]:
ERROR:SBIT?
<x> = 1 ~ 4

例 :SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
SBIT ON
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
SBIT?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
ERROR:SBIT 1

**:SERIALbus<x>:PSI5:SEARCh[:SETup]:ER
Ror:SBOundary**

機能 PSI5 信号の Slot Boundary Error 検索を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5:SEARCh[:SETup]:
ERROR:SBOundary {<Boolean>}
:SERIALbus<x>:PSI5:SEARCh[:SETup]:
ERROR:SBOundary?
<x> = 1 ~ 4

例 :SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
SBOUNDARY ON
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
SBOUNDARY?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
ERROR:SBOUNDARY 1

**:SERIALbus<x>:PSI5:SEARCh[:SETup]:FR
AMeinslot?**

機能 PSI5 信号のスロット指定検索のすべての設定値を問い合わせます。

構文 :SERIALbus<x>:PSI5:SEARCh[:SETup]:
FRAMeinslot?
<x> = 1 ~ 4

**:SERIALbus<x>:PSI5:SEARCh[:SETup]:FR
AMeinslot:SNUMber**

機能 PSI5 信号のスロット指定検索のスロット番号を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5:SEARCh[:SETup]:
FRAMeinslot:SNUMber {<NRf>}
:SERIALbus<x>:PSI5:SEARCh[:SETup]:
FRAMeinslot:SNUMber?
<x> = 1 ~ 4
<NRf> = 1 ~ 6

例 :SERIALBUS1:PSI5:SEARCH:SETUP:
FRAMEINSLOT:SNUMBER 1
:SERIALBUS1:PSI5:SEARCH:SETUP:
FRAMEINSLOT:SNUMBER?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
FRAMEINSLOT:SNUMBER 1

**:SERIALbus<x>:PSI5:SEARCh[:SETup]:MO
DE**

機能 PSI5 信号検索の種類を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5:SEARCh[:SETup]:
MODE {DATA|ERROR|FRAMeinslot|SBIT|
SYNC}
:SERIALbus<x>:PSI5:SEARCh[:SETup]:
MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:PSI5:SEARCH:SETUP:
MODE DATA
:SERIALBUS1:PSI5:SEARCH:SETUP:MODE?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
MODE DATA

:SERIALbus<x>:PSI5:TREND<y>?

機能 PSI5 信号解析の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 :SERIALbus<x>:PSI5:TREND<y>?
<x> = 1 ~ 4
<y> = 1 ~ 4

:SERIALbus<x>:PSI5:TREND<y>:AScale

機能 PSI5 信号解析のトレンド表示のオートスケールを実行します。

構文 :SERIALbus<x>:PSI5:TREND<y>:AScale
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:PSI5:TREND1:ASCALE

:SERIALbus<x>:PSI5:TREND<y>:CURSor?

機能 PSI5 信号解析の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。

構文 :SERIALbus<x>:PSI5:TREND<y>:CURSor?
<x> = 1 ~ 4
<y> = 1 ~ 4

5.25 SERIALbus グループ

:SERIALbus<x>:PSI5:TREND<y>:CURSOR:DT:VALUE?

機能 PSI5 信号解析のトレンド表示のカーソル間の時間軸値を問い合わせます。

構文 :SERIALbus<x>:PSI5:TREND<y>:CURSOR:
DT:VALUE?
<x> = 1 ~ 4
<y> = 1 ~ 4

:SERIALbus<x>:PSI5:TREND<y>:CURSOR:DV:VALUE?

機能 PSI5 信号解析のトレンド表示のカーソル間の垂直軸値を問い合わせます。

構文 :SERIALbus<x>:PSI5:TREND<y>:CURSOR:
DV:VALUE?
<x> = 1 ~ 4
<y> = 1 ~ 4

:SERIALbus<x>:PSI5:TREND<y>:CURSOR:MODE

機能 PSI5 信号解析のトレンド表示のカーソルのモードを設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5:TREND<y>:CURSOR:
MODE {<Boolean>}
:SERIALbus<x>:PSI5:TREND<y>:CURSOR:
MODE?
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:PSI5:TREND1:CURSOR:
MODE ON
:SERIALBUS1:PSI5:TREND1:CURSOR:MODE?
-> :SERIALBUS1:PSI5:TREND1:CURSOR:
MODE 1

:SERIALbus<x>:PSI5:TREND<y>:CURSOR:POSITION<z>

機能 PSI5 信号解析のトレンド表示の各カーソルの位置を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5:TREND<y>:CURSOR:
POSITION<z> {<NRF>}
:SERIALbus<x>:PSI5:TREND<y>:CURSOR:
POSITION<z>?
<x> = 1 ~ 4
<y> = 1 ~ 4
<z> = 1、2
<NRF> = -5 ~ 5(div)、(10div/ 表示レコード長) ステップ

例 :SERIALBUS1:PSI5:TREND1:CURSOR:
POSITION1 -5
:SERIALBUS1:PSI5:TREND1:CURSOR:
POSITION1?
-> :SERIALBUS1:PSI5:TREND1:CURSOR:
POSITION1 -5

:SERIALbus<x>:PSI5:TREND<y>:CURSOR:T<z>:VALUE?

機能 PSI5 信号解析のトレンド表示のカーソルの時間軸値を問い合わせます。

構文 :SERIALbus<x>:PSI5:TREND<y>:CURSOR:
T<z>:VALUE?
<x> = 1 ~ 4
<y> = 1 ~ 4
<z> = 1、2

:SERIALbus<x>:PSI5:TREND<y>:CURSOR:V<z>:VALUE?

機能 PSI5 信号解析のトレンド表示のカーソルの垂直軸値を問い合わせます。

構文 :SERIALbus<x>:PSI5:TREND<y>:CURSOR:
V<z>:VALUE?
<x> = 1 ~ 4
<y> = 1 ~ 4
<z> = 1、2

:SERIALbus<x>:PSI5:TREND<y>:DISPLAY

機能 PSI5 信号解析のトレンドの表示の ON/OFF を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5:TREND<y>:
DISPLAY {<Boolean>}
:SERIALbus<x>:PSI5:TREND<y>:DISPLAY?
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:PSI5:TREND1:DISPLAY ON
:SERIALBUS1:PSI5:TREND1:DISPLAY?
-> :SERIALBUS1:PSI5:TREND1:DISPLAY 1

:SERIALbus<x>:PSI5:TREND<y>:HRANGE

機能 PSI5 信号解析の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。

構文 :SERIALbus<x>:PSI5:TREND<y>:
HRANGE {MAIN|Z1|Z2}
:SERIALbus<x>:PSI5:TREND<y>:HRANGE?
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:PSI5:TREND1:HRANGE MAIN
:SERIALBUS1:PSI5:TREND1:HRANGE?
-> :SERIALBUS1:PSI5:TREND1:
HRANGE MAIN

:SERIALbus<x>:PSI5:TREND<y>:SOURCE

機能 PSI5 信号解析のトレンド表示の対象スロット番号を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5:TREND<y>:
SOURCE {<Nrf>}
:SERIALbus<x>:PSI5:TREND<y>:SOURCE?
<x> = 1 ~ 4
<y> = 1 ~ 4
<Nrf> = 1 ~ 6

例 :SERIALBUS1:PSI5:TREND1:SOURCE 1
:SERIALBUS1:PSI5:TREND1:SOURCE?
-> :SERIALBUS1:PSI5:TREND1:SOURCE 1

:SERIALbus<x>:PSI5:TREND<y>:VERTICAL

機能 PSI5 信号解析のトレンド表示の垂直レンジを設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5:TREND<y>:
VERTICAL {<Nrf>,<Nrf>}
:SERIALbus<x>:PSI5:TREND<y>:
VERTICAL?
<x> = 1 ~ 4
<y> = 1 ~ 4
<Nrf> = -1.0000E+31 ~ 1.0000E+31

例 :SERIALBUS1:PSI5:TREND1:VERTICAL 1,
-1
:SERIALBUS1:PSI5:TREND1:VERTICAL?
-> :SERIALBUS1:PSI5:TREND1:
VERTICAL 1.0000000E+00,
-1.0000000E+00

:SERIALbus<x>:PSI5:TREND<y>:VTDISPLAY

機能 PSI5 信号解析のトレンド表示の VT 波形の表示の ON/OFF を設定 / 問い合わせします。

構文 :SERIALbus<x>:PSI5:TREND<y>:
VTDISPLAY {<Boolean>}
:SERIALbus<x>:PSI5:TREND<y>:
VTDISPLAY?
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:PSI5:TREND1:VTDISPLAY ON
:SERIALBUS1:PSI5:TREND1:VTDISPLAY?
-> :SERIALBUS1:PSI5:TREND1:
VTDISPLAY 1

:SERIALbus<x>:RWINDOW

機能 検索点を拡大表示するウィンドウで検索された部分を表示するズーム画面を設定 / 問い合わせします。

構文 :SERIALbus<x>:RWINDOW {Z1|Z2}
:SERIALbus<x>:RWINDOW?
<x> = 1 ~ 4

例 :SERIALBUS1:RWINDOW Z1
:SERIALBUS1:RWINDOW?
-> :SERIALBUS1:RWINDOW Z1

:SERIALbus<x>:SENT?

機能 SENT 信号の解析 / 検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:ANALYZE?

機能 SENT 信号解析の関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT:ANALYZE?
<x> = 1 ~ 4

:SERIALbus<x>:SENT[:ANALYZE]:SETUP?

機能 SENT 信号解析のバス設定に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT[:ANALYZE]:SETUP?
<x> = 1 ~ 4

:SERIALbus<x>:SENT[:ANALYZE]:SETUP:DISPLAY

機能 SENT 信号解析の解析結果の表示方法を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALYZE]:SETUP:
DISPLAY {BOTH|FAST|SLOW}
:SERIALbus<x>:SENT[:ANALYZE]:SETUP:
DISPLAY?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:ANALYZE:SETUP:
DISPLAY BOTH
:SERIALBUS1:SENT:ANALYZE:SETUP:
DISPLAY?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
DISPLAY BOTH

:SERIALbus<x>:SENT[:ANALYZE]:SETUP:FAST?

機能 SENT 信号解析の Fast CH に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT[:ANALYZE]:SETUP:
FAST?
<x> = 1 ~ 4

5.25 SERIALbus グループ

:SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:DTYPE

機能 SENT 信号解析の Fast CH のデータ形式を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:DTYPE {NIBble|USER}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:DTYPE?

<x> = 1 ~ 4

例 :SERIALBUS1:SENT:ANALYZE:SETUP:FAST:
DTYPE NIBBLE
:SERIALBUS1:SENT:ANALYZE:SETUP:FAST:
DTYPE?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FAST:DTYPE NIBBLE

:SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:USERDEF

機能 SENT 信号解析の Fast CH のユーザー定義データに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:USERDEF?
<x> = 1 ~ 4

:SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:ORDER

機能 SENT 信号解析の Fast CH の各ユーザー定義データに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:ORDER?
<x> = 1 ~ 4
<y> = 1 ~ 4

:SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:MODE

機能 SENT 信号解析の Fast CH の各ユーザー定義データの有効 / 無効を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:MODE {<Boolean>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:MODE?
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:SENT:ANALYZE:SETUP:FAST:
MODE ON
:SERIALBUS1:SENT:ANALYZE:SETUP:FAST:
MODE?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FAST:MODE 1

:SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:ORDER

機能 SENT 信号解析の Fast CH の各ユーザー定義データのエンディアンを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:ORDER {BIG|
LITTLE}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:ORDER?

<x> = 1 ~ 4

<y> = 1 ~ 4

例 :SERIALBUS1:SENT:ANALYZE:SETUP:FAST:
ORDER BIG
:SERIALBUS1:SENT:ANALYZE:SETUP:FAST:
ORDER?
->:SERIALBUS1:SENT:ANALYZE:SETUP:
FAST:ORDER BIG

:SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:SIZE

機能 SENT 信号解析の Fast CH の各ユーザー定義データのデータサイズを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:SIZE {<NRf>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:SIZE?

<x> = 1 ~ 4

<y> = 1 ~ 4

<NRf> = 0 ~ 24

例 :SERIALBUS1:SENT:ANALYZE:SETUP:FAST:
SIZE 24
:SERIALBUS1:SENT:ANALYZE:SETUP:FAST:
SIZE?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FAST:SIZE 24

:SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:MULTIPLEXING

機能 SENT 信号解析の FastCH のユーザー定義データで、Multiplexing の有効 / 無効を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:MULTIPLEXING {<Boolean>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:MULTIPLEXING?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:ANALYZE:SETUP:FAST:
MULTIPLEXING ON
:SERIALBUS1:SENT:ANALYZE:SETUP:FAST:
MULTIPLEXING?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FAST:MULTIPLEXING ON

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat?

機能 SENT 信号解析の解析フォーマットに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat?
<x> = 1 ~ 4

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CEFactor?

機能 SENT 信号解析のエラー要因に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CEFactor?
<x> = 1 ~ 4

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CEFactor:SAComm?

機能 SENT 信号解析の Status and Communication のエラー要因に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:SAComm?
<x> = 1 ~ 4

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CEFactor:SAComm:BIT<y>?

機能 SENT 信号解析の Status and Communication のエラー要因のビットを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CEFactor:SAComm:BIT<y> {<Boolean>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CEFactor:SAComm:BIT<y>?
<x> = 1 ~ 4
<y> = 0 ~ 1

例 :SERIALBUS1:SENT:ANALYZE:SETUP:FORMAT:CEFACTOR:SAComm:BIT0 ON
:SERIALBUS1:SENT:ANALYZE:SETUP:FORMAT:CEFACTOR:SAComm:BIT0?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:FORMAT:CEFACTOR:SAComm:BIT0 1

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CEFactor:SCPulses

機能 SENT 信号解析の連続キャリブレーションパルスのエラー要因を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CEFactor:SCPulses {OFF|OPT2|POPTION}
:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CEFactor:SCPulses?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:ANALYZE:SETUP:FORMAT:CEFACTOR:SCPULSES OFF
:SERIALBUS1:SENT:ANALYZE:SETUP:FORMAT:CEFACTOR:SCPULSES?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:FORMAT:CEFACTOR:SCPULSES OFF

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CRCType

機能 SENT 信号解析の CRC 演算方式を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CRCType {LEGacy|RECommended}
:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CRCType?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:ANALYZE:SETUP:FORMAT:CRCTYPE LEGACY
:SERIALBUS1:SENT:ANALYZE:SETUP:FORMAT:CRCTYPE?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:FORMAT:CRCTYPE LEGACY

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CTICK

機能 SENT 信号解析のクロックティック値を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CTICK {<Nrf>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CTICK?
<x> = 1 ~ 4

<Nrf> = 1.00μs ~ 100.00μs(設定分解能 0.01μs)
例 :SERIALBUS1:SENT:ANALYZE:SETUP:FORMAT:CTICK 0.000001
:SERIALBUS1:SENT:ANALYZE:SETUP:FORMAT:CTICK?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:FORMAT:CTICK 1.000000E-06

5.25 SERIALbus グループ

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:CTOLerance

機能 SENT 信号解析のクロック許容差を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:CTOLerance {<NRf>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:CTOLerance?
<x> = 1 ~ 4
<NRf> = ± 1.0% ~ ± 30.0%(設定分解能 0.1%)

例 :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:CTOLerance 25.0
:SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:CTOLERANCE?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:CTOLERANCE 25.0E+00

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:DNIBbles

機能 SENT 信号解析のデータニブル数を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:DNIBbles {<NRf>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:DNIBbles?
<x> = 1 ~ 4
<NRf> = 1 ~ 6

例 :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:DNIBBLES 6
:SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:DNIBBLES?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:DNIBBLES 6

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:PPULse

機能 SENT 信号解析のポーズパルスの有無を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:PPULse {<Boolean>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:PPULse?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:PPULSE ON
:SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:PPULSE?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:PPULSE 1

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMat:VERsion

機能 SENT 信号解析の仕様バージョンを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:VERsion {APR2016|FEB2008|
JAN2010}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:VERsion?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:VERSION JAN2010
:SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:VERSION?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:VERSION JAN2010

:SERIALbus<x>:SENT[:ANALyze]:SETup:SType

機能 SENT 信号解析の Slow CH 形式を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:
SType {ENHanced|SHORT}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
SType?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:ANALYZE:SETUP:
SType ENHANCED
:SERIALBUS1:SENT:ANALYZE:SETUP:
SType?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
SType ENHANCED

:SERIALbus<x>:SENT[:ANALyze]:SETup:SOURce

機能 SENT 信号解析のソースを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT[:ANALyze]:SETup:
SOURce {<NRf>|MATH<y>|PODA<y>|
PODB<y>|PODC<u>|PODD<y>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
SOURce?
<x> = 1 ~ 4
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<y>、PODB<y>、PODC<y>、PODD<y>
の <y> = 0 ~ 7

例 :SERIALBUS1:SENT:ANALYZE:SETUP:
SOURCE 1
:SERIALBUS1:SENT:ANALYZE:SETUP:
SOURCE?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
SOURCE 1

:SERIALbus<x>:SENT:DETAil?

機能 SENT 信号解析の解析リストに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT:DETAil?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:DETAil:DISPlay

機能 SENT 信号解析の解析リストの表示モードを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:DETAil:DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:SENT:DETAil:DISPlay?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:DETAil:DISPlay FULL
:SERIALBUS1:SENT:DETAil:DISPlay?
-> :SERIALBUS1:SENT:DETAil:DISPlay FULL

:SERIALbus<x>:SENT:DETAil:LIST:ALL?

機能 SENT 信号の解析結果リストのすべての解析番号の全データを問い合わせます。

構文 :SERIALbus<x>:SENT:DETAil:LIST:ALL?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:DETAil:LIST:ALL?

-> #8(8 桁のバイト数)(データ列)

解説

- すべての解析番号の全データがブロックデータの形式で返されます。
- 解析番号ごとのデータ間は、ASCII コードの「0AH」で区切られます。

:SERIALbus<x>:SENT:DETAil:LIST:ITEM?

機能 SENT 信号の解析結果リストに表示される項目を問い合わせます。

構文 :SERIALbus<x>:SENT:DETAil:LIST:ITEM?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:DETAil:LIST:ITEM?

-> :SERIALBUS1:SENT:DETAil:LIST:ITEM "No.,Time(ms),Sync(us),Tick(us),S&C,Data,CRC,Length(tick),Information,SlowCH,"

解説 「:SERIALbus<x>:SENT[:ANALyze]:SETup:DISPlay {BOTH|FAST|SLOW}」の設定により、解析結果リスト表示項目は切り替わります。

:SERIALbus<x>:SENT:DETAil:LIST:VALUE?

機能 SENT 信号の解析結果リストの指定した解析番号の全データを問い合わせます。

構文 :SERIALbus<x>:SENT:DETAil:LIST:VALUE? {<NRf>|MAXimum|MINimum}
<x> = 1 ~ 4

<NRf> = -100000 ~ 100000

例 :SERIALBUS1:SENT:DETAil:LIST:VALUE? 0

-> :SERIALBUS1:SENT:DETAil:LIST:VALUE "0,-0.1672,168.00,3.00,1100,6,3,5,E,B,9,3,283.73,,,"

解説 「:SERIALbus<x>:SENT[:ANALyze]:SETup:DISPlay {BOTH|FAST|SLOW}」の設定により、解析結果リスト表示項目は切り替わります。

:SERIALbus<x>:SENT:SEARCh?

機能 SENT 信号検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT:SEARCh?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:SEARCh:ABORT

機能 SENT 信号検索を中止します。

構文 :SERIALbus<x>:SENT:SEARCh:ABORT
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCh:ABORT

:SERIALbus<x>:SENT:SEARCh:EXECute

機能 SENT 信号検索を実行します。

構文 :SERIALbus<x>:SENT:SEARCh:EXECute
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCh:EXECUTE

:SERIALbus<x>:SENT:SEARCh:SElect

機能 SENT 信号検索のズームウインドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。

構文 :SERIALbus<x>:SENT:SEARCh:SElect {<NRf>|MAXimum}
:SERIALbus<x>:SENT:SEARCh:SElect?
<x> = 1 ~ 4

<NRf> = 0 ~ 49999

例 :SERIALBUS1:SENT:SEARCh:SElect 1

:SERIALBUS1:SENT:SEARCh:SElect?

-> :SERIALBUS1:SENT:SEARCh:SElect 1.50000000

解説 検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

5.25 SERIALbus グループ

:SERIALbus<x>:SENT:SEARCH:SElect?

MAXimum

機能 SENT 信号検索の検索点の総数を問い合わせます。

構文 :SERIALbus<x>:SENT:SEARCH:
SElect? MAXimum
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:
SELECT? MAXimum
-> :SERIALBUS1:SENT:SEARCH:
SELECT 100

解説 検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:SERIALbus<x>:SENT:SEARCH:SETup?

機能 SENT 信号検索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 :SERIALbus<x>:SENT:SEARCH:SETup?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:SEARCH[:SETup]:ER Ror?

機能 SENT 信号検索の Error に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT:SEARCH[:SETup]:
ERRor?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:SEARCH[:SETup]:ER Ror:FCRC

機能 SENT 信号検索の Fast CH の CRC エラーを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETup]:
ERRor:FCRC {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
ERRor:FCRC?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
FCRC ON
:SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
FCRC?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
ERROR:FCRC 1

:SERIALbus<x>:SENT:SEARCH[:SETup]:ER Ror:NDValue

機能 SENT 信号検索のニブルデータバリュエラーを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETup]:
ERRor:NDValue {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
ERRor:NDValue?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
NDVALUE ON
:SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
NDVALUE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
ERROR:NDVALUE 1

:SERIALbus<x>:SENT:SEARCH[:SETup]:ER Ror:NNUmber

機能 SENT 信号検索のニブルデータ数エラーを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETup]:
ERRor:NNUmber {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
ERRor:NNUmber?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
NNUmber ON
:SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
NNUmber?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
ERROR:NNUmber 1

:SERIALbus<x>:SENT:SEARCH[:SETUP]:ERROR:SACOMM

機能 SENT 信号検索の Status and Communication エラーを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
ERROR:SACOMM {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
ERROR:SACOMM?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
SACOMM ON
:SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
SACOMM?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
ERROR:SACOMM 1

:SERIALbus<x>:SENT:SEARCH[:SETUP]:ERROR:SCPULSES

機能 SENT 信号検索の連続キャリブレーションパルスエラーを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
ERROR:SCPULSES{<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
ERROR:SCPULSES?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
SCPULSES ON
:SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
SCPULSES?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
ERROR:SCPULSES 1

解説 「:SERIALbus<x>:SENT[:ANALyze]:SETUP:
FORMAT:CEFactor:SCPULSES」が「OFF」の場
合の設定は、「OFF」固定です。

:SERIALbus<x>:SENT:SEARCH[:SETUP]:ERROR:SCRC

機能 SENT 信号検索の Slow CH の CRC エラーを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
ERROR:SCRC {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
ERROR:SCRC?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
SCRC ON
:SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
SCRC?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
ERROR:SCRC 1

:SERIALbus<x>:SENT:SEARCH[:SETUP]:FDATA:ATA?

機能 SENT 信号検索の Fast CH のデータに関するすべての設定値を問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
FDATA?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:FDATA:DATA<y>?

機能 SENT 信号検索の Fast CH の各ユーザーデータに関するすべての設定値を問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
FDATA:DATA<y>?
<x> = 1 ~ 4
<y> = 1 ~ 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:FDATA:DATA<y>:CONDITION

機能 SENT 信号検索の Fast CH の各ユーザーデータの判定条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
FDATA:DATA<y>:CONDITION {BETWEEN|
EQUAL|GREATER|LESS|NOTBetween|
NOTEQUAL}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
FDATA:DATA<y>:CONDITION?
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DATA1:CONDITION BETWEEN
:SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DATA1:CONDITION?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
FDATA:DATA1:CONDITION BETWEEN

:SERIALbus<x>:SENT:SEARCH[:SETUP]:FDATA:DATA<y>:DECIMAL<z>

機能 SENT 信号検索の Fast CH の各ユーザーデータのデータを 10 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
FDATA:DATA<y>:DECIMAL<z> {<NRF>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
FDATA:DATA<y>:DECIMAL<z>?
<x> = 1 ~ 4
<y> = 1 ~ 4
<z> = 1、2
<NRF> = 本体ユーザーズマニュアル機能編参照。

例 :SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DATA1:DECIMAL1 1
:SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DATA1:DECIMAL1?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
FDATA:DATA1:DECIMAL1 1

5.25 Serialbus グループ

:Serialbus<x>:SENT:SEARCH[:SETUP]:FDATA:DATA<y>:MODE

機能 SENT 信号検索の Fast CH の各ユーザーデータのデータ条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :Serialbus<x>:SENT:SEARCH[:SETUP]:
FDATA:DATA<y>:MODE {<Boolean>}
:Serialbus<x>:SENT:SEARCH[:SETUP]:
FDATA:DATA<y>:MODE?
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DATA1:MODE ON
:SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DATA1:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
FDATA:DATA1:MODE 1

:Serialbus<x>:SENT:SEARCH[:SETUP]:FDATA:DNIBBles?

機能 SENT 信号検索の Fast CH のニブルデータに関するすべての設定値を問い合わせます。

構文 :Serialbus<x>:SENT:SEARCH[:SETUP]:
FDATA:DNIBBles?
<x> = 1 ~ 4

:Serialbus<x>:SENT:SEARCH[:SETUP]:FDATA:DNIBBles:CONDition

機能 SENT 信号検索の Fast CH のニブルデータの判定条件を設定 / 問い合わせします。

構文 :Serialbus<x>:SENT:SEARCH[:SETUP]:
FDATA:DNIBBles:CONDition {FALSE|
TRUE}
:Serialbus<x>:SENT:SEARCH[:SETUP]:
FDATA:DNIBBles:CONDition?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DNIBBLES:CONDITION TRUE
:SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DNIBBLES:CONDITION?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
FDATA:DNIBBLES:CONDITION TRUE

:Serialbus<x>:SENT:SEARCH[:SETUP]:FDATA:DNIBBles:HEXa

機能 SENT 信号検索の Fast CH のニブルデータのデータを 16 進数で設定します。

構文 :Serialbus<x>:SENT:SEARCH[:SETUP]:
FDATA:DNIBBles:HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DNIBBLES:HEXa "112233"

:Serialbus<x>:SENT:SEARCH[:SETUP]:FDATA:DNIBBles:PATtern

機能 SENT 信号検索の Fast CH のニブルデータのデータを 2 進数で設定 / 問い合わせします。

構文 :Serialbus<x>:SENT:SEARCH[:SETUP]:
FDATA:DNIBBles:PATtern {<文字列>}
:Serialbus<x>:SENT:SEARCH[:SETUP]:
FDATA:DNIBBles:PATtern?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DNIBBLES:
PATTERN "111100001111000011110000"
:SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DNIBBLES:PATTERN?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
FDATA:DNIBBLES:
PATTERN "111100001111000011110000"

:Serialbus<x>:SENT:SEARCH[:SETUP]:FSAComm?

機能 SENT 信号検索の Fast CH の Status and Communication ニブルに関するすべての設定値を問い合わせます。

構文 :Serialbus<x>:SENT:SEARCH[:SETUP]:
FSAComm?
<x> = 1 ~ 4

:Serialbus<x>:SENT:SEARCH[:SETUP]:FSAComm:HEXa

機能 SENT 信号検索の Fast CH の Status and Communication ニブルのデータを 16 進数で設定します。

構文 :Serialbus<x>:SENT:SEARCH[:SETUP]:
FSAComm:HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:
FSACOMM:HEXa "F"

:Serialbus<x>:SENT:SEARCH[:SETUP]:FSAComm:PATtern

機能 SENT 信号検索の Fast CH Status and Communication ニブルのデータを 2 進数で設定 / 問い合わせします。

構文 :Serialbus<x>:SENT:SEARCH[:SETUP]:
FSAComm:PATtern {<文字列>}
:Serialbus<x>:SENT:SEARCH[:SETUP]:
FSAComm:PATtern?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:
FSACOMM:PATTERN "1111"
:SERIALBUS1:SENT:SEARCH:SETUP:
FSACOMM:PATTERN?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
FSACOMM:PATTERN "1111"

:SERIALbus<x>:SENT:SEARCH[:SETUP]:MODE

機能 SENT 信号検索のモードを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
MODE {EFAST|ERROR|ESLOW|FDATA|
FSAComm|SDATA}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:
MODE ERROR
:SERIALBUS1:SENT:SEARCH:SETUP:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
MODE ERROR

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA?

機能 SENT 信号検索の Slow CH 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 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED?

機能 SENT 信号検索の Slow CH の Enhanced タイプに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:CBIT

機能 SENT 信号検索の Slow CH の Enhanced タイプの Configuration ビットを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:CBIT {D12Bit|D16Bit}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:CBIT?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:CBIT D12Bit
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:CBIT?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:CBIT D12Bit

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D12BIT?

機能 SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D12Bit?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D12BIT:DATA?

機能 SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID 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 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D12BIT:DATA?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D12BIT:DATA:CONDITION

機能 SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータの判定条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D12BIT:DATA:
CONDITION {BETWEEN|EQUAL|FALSE|
GREATER|LESS|NOTBETWEEN|NOTEQUAL|
TRUE}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D12BIT:DATA:
CONDITION?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:CONDITION FALSE
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:CONDITION?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D12BIT:DATA:
CONDITION FALSE

5.25 SERIALbus グループ

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D12Bit:DATA:DECimal<y>

機能 SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータを 10 進数で設定 / 問い合わせします。

構文
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATA:ENHanced:D12Bit:DATA:
DECimal<y> {<Nrf>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATA:ENHanced:D12Bit:DATA:
DECimal<y>?
<x> = 1 ~ 4
<y> = 1、2
<Nrf> = 本体ユーザーズマニュアル機能編参照。

例
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:DECIMAL1 123
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:DECIMAL1?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D12BIT:DATA:
DECIMAL1 123

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D12Bit:DATA:HEXa

機能 SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータを 16 進数で設定します。

構文
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATA:ENHanced:D12Bit:DATA:
HEXa {<文字列>}
<x> = 1 ~ 4

例
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:HEXA "123"

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D12Bit:DATA:MODE

機能 SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータ条件 (有効 / 無効) を設定 / 問い合わせします。

構文
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATA:ENHanced:D12Bit:DATA:
MODE {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATA:ENHanced:D12Bit:DATA:MODE?
<x> = 1 ~ 4

例
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:MODE ON
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D12BIT:DATA:MODE 1

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D12Bit:DATA:PATtern

機能 SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータを 2 進数で設定 / 問い合わせします。

構文
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATA:ENHanced:D12Bit:DATA:
PATtern {<文字列>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATA:ENHanced:D12Bit:DATA:PATtern?
<x> = 1 ~ 4

例
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:
PATTERN "111101010101"
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:PATTERN?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D12BIT:DATA:
PATTERN "111101010101"

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D12Bit:ID?

機能 SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID の ID に関するすべての設定値を問い合わせます。

構文
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATA:ENHanced:D12Bit:ID?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D12Bit:ID:CONDition

機能 SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID の ID 判定条件を設定 / 問い合わせします。

構文
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATA:ENHanced:D12Bit:ID:
CONDition {BETween|EQual|GREater|
LESS|NOTBetween|NOTEqual}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATA:ENHanced:D12Bit:ID:CONDition?
<x> = 1 ~ 4

例
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:ID:CONDITION BETWEEN
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:ID:CONDITION?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D12BIT:ID:
CONDITION BETWEEN

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHanced:D12Bit:ID:DECimal<y>

機能 SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID の ID を 10 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHanced:D12Bit:ID:
DECimal<y> {<Nrf>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHanced:D12Bit:ID:HEXa<y>?
<x> = 1 ~ 4
<y> = 1, 2

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:ID:DECIMAL1 0
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:ID:DECIMAL1?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D12BIT:ID:DECIMAL1 0

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHanced:D12Bit:ID:MODE

機能 SENT 信号検索の Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID の ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHanced:D12Bit:ID:
MODE {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHanced:D12Bit:ID:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:ID:MODE ON
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:ID:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D12BIT:ID:MODE 1

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHanced:D16Bit?

機能 SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHanced:D16Bit?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHanced:D16Bit:DATA?

機能 SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID 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 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHanced:D16Bit:DATA?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHanced:D16Bit:DATA:CONDition

機能 SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータの判定条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHanced:D16Bit:DATA:
CONDition {BETween|EQual|FALSe|
GREater|LESS|NOTBetween|NOTEqual|
TRUE}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHanced:D16Bit:DATA:
CONDition?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:DATA:CONDITION FALSE
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:DATA:CONDITION?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D16BIT:DATA:
CONDITION FALSE

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHanced:D16Bit:DATA:DECimal<y>

機能 SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータを 10 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHanced:D16Bit:DATA:
DECimal<y> {<Nrf>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHanced:D16Bit:DATA:
DECimal<y>?
<x> = 1 ~ 4
<y> = 1, 2
<Nrf> = 本体ユーザーズマニュアル機能編参照。

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:DATA:DECIMAL1 123
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:DATA:DECIMAL1?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D16BIT:DATA:
DECIMAL1 123

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHanced:D16Bit:DATA:HEXa

機能 SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータを 16 進数で設定します。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHanced:D16Bit:DATA:
HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:DATA:HEXA "0000"

5.25 Serialbus グループ

:Serialbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D16Bit:DATA:MODE

機能 SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータ条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :Serialbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D16Bit:DATA:MODE {<Boolean>}
:Serialbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D16Bit:DATA:MODE?<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:ENHANCED:D16BIT:DATA:MODE ON
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:ENHANCED:D16BIT:DATA:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:ENHANCED:D16BIT:DATA:MODE 1

:Serialbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D16Bit:DATA:PATtern

機能 SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータを 2 進数で設定 / 問い合わせします。

構文 :Serialbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D16Bit:DATA:PATtern {<文字列>}
:Serialbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D16Bit:DATA:PATtern?<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:ENHANCED:D16BIT:DATA:PATTERN "1111000011110000"
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:ENHANCED:D16BIT:DATA:PATTERN?
-> :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:ENHANCED:D16BIT:DATA:PATTERN "1111000011110000"

:Serialbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D16Bit:ID?

機能 SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID に関するすべての設定値を問い合わせます。

構文 :Serialbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D16Bit:ID?<x> = 1 ~ 4

:Serialbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D16Bit:ID:CONDition

機能 SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID 判定条件を設定 / 問い合わせします。

構文 :Serialbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D16Bit:ID:CONDition {BETween|EQual|GREater|LESS|NOTBetween|NOTEqual}
:Serialbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D16Bit:ID:CONDition?<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:ENHANCED:D16BIT:ID:CONDITION BETWEEN
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:ENHANCED:D16BIT:ID:CONDITION?
-> :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:ENHANCED:D16BIT:ID:CONDITION BETWEEN

:Serialbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D16Bit:ID:DECimal<y>

機能 SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID を 10 進数で設定 / 問い合わせします。

構文 :Serialbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D16Bit:ID:DECimal<y> {<NRF>}
:Serialbus<x>:SENT:SEARCH[:SETup]:SDATA:ENHanced:D16Bit:ID:DECimal<y>?<x> = 1 ~ 4
<y> = 1, 2

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:ENHANCED:D16BIT:ID:DECIMAL1 0
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:ENHANCED:D16BIT:ID:DECIMAL1?
-> :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:ENHANCED:D16BIT:ID:DECIMAL1 0

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHanced:D16Bit:ID:MODE

機能 SENT 信号検索の Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHanced:D16Bit:ID:
MODE {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHanced:D16Bit:ID:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:ID:MODE ON
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:ID:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D16BIT:ID:MODE 1

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT?

機能 SENT 信号検索の Slow CH の Short タイプに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:DATA?

機能 SENT 信号検索の Slow CH の Short タイプ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 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:DATA?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:DATA:CONDition

機能 SENT 信号検索の Slow CH の Short タイプのデータの判定条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:DATA:CONDition {BETween|
EQUAL|FALSE|GREATER|LESS|NOTBetween|
NOTEQUAL|TRUE}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:DATA:CONDition?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
SHORT:DATA:CONDITION FALSE
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
SHORT:DATA:CONDITION?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:SHORT:DATA:CONDITION FALSE

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:DATA:DECimal<y>

機能 SENT 信号検索の Slow CH の Short タイプのデータを 10 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:DATA:DECimal<y> {<Nrf>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:DATA:DECimal<y>?
<x> = 1 ~ 4
<y> = 1, 2
<Nrf> = 本体ユーザーズマニュアル機能編参照。

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
SHORT:DATA:DECIMAL1 123
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
SHORT:DATA:DECIMAL1?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:SHORT:DATA:DECIMAL1 123

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:DATA:HEXa

機能 SENT 信号検索の Slow CH の Short タイプのデータを 16 進数で設定します。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:DATA:HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
SHORT:DATA:HEXA "00"

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:DATA:MODE

機能 SENT 信号検索の Slow CH の Short タイプのデータ条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:DATA:MODE {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:DATA:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
SHORT:DATA:MODE ON
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
SHORT:DATA:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:SHORT:DATA:MODE 1

5.25 SERIALbus グループ

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:DATA:PATTERN

機能 SENT 信号検索の Slow CH の Short タイプのデータを 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:DATA:PATTERN {<文字列>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:DATA:PATTERN?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:SHORT:DATA:PATTERN "11110000"
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:SHORT:DATA:PATTERN?
-> :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:SHORT:DATA:PATTERN "11110000"

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:ID?

機能 SENT 信号検索の Slow CH の Short タイプの ID に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:ID?
<x> = 1 ~ 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:ID:CONDition

機能 SENT 信号検索の Slow CH の Short タイプの ID 判定条件を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:ID:CONDition {BETween|EQUal|GREater|LESS|NOTBetween|NOTEqual}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:ID:CONDition?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:SHORT:ID:CONDition BETWEEN
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:SHORT:ID:CONDition?
-> :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:SHORT:ID:CONDition BETWEEN

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:ID:DECimal<y>

機能 SENT 信号検索の Slow CH の Short タイプの ID を 10 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:ID:DECimal<y> {<NRF>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:ID:DECimal<y>?
<x> = 1 ~ 4
<y> = 1, 2

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:SHORT:ID:DECimal 0
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:SHORT:ID:DECIMAL1?
-> :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:SHORT:ID:DECIMAL1 0

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:ID:MODE

機能 SENT 信号検索の Slow CH の Short タイプの ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:ID:MODE {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:ID:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:SHORT:ID:MODE ON
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:SHORT:ID:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:SHORT:ID:MODE 1

:SERIALbus<x>:SENT:TREND<y>?

機能 SENT 信号解析の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 :SERIALbus<x>:SENT:TREND<y>?
<x> = 1 ~ 4
<y> = 1 ~ 4

:SERIALbus<x>:SENT:TREND<y>:AScale

機能 SENT 信号解析のトレンド表示のオートスケールを実行します。

構文 :SERIALbus<x>:SENT:TREND<y>:AScale
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:SENT:TREND1:ASCALE

:SERIALbus<x>:SENT:TREND<y>:CURSor?

機能 SENT 信号解析の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。

構文 :SERIALbus<x>:SENT:TREND<y>:CURSor?
<x> = 1 ~ 4
<y> = 1 ~ 4

:SERIALbus<x>:SENT:TREND<y>:CURSOR:DT:VALUE?

機能 SENT 信号解析のトレンド表示のカーソル間の時間軸値を問い合わせます。

構文 :SERIALbus<x>:SENT:TREND<y>:CURSOR:
DT:VALUE?
<x> = 1 ~ 4
<y> = 1 ~ 4

:SERIALbus<x>:SENT:TREND<y>:CURSOR:DV:VALUE?

機能 SENT 信号解析のトレンド表示のカーソル間の垂直軸値を問い合わせます。

構文 :SERIALbus<x>:SENT:TREND<y>:CURSOR:
DV:VALUE?
<x> = 1 ~ 4
<y> = 1 ~ 4

:SERIALbus<x>:SENT:TREND<y>:CURSOR:MODE

機能 SENT 信号解析のトレンド表示の自動測定モードを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:TREND<y>:CURSOR:
MODE {<Boolean>}
:SERIALbus<x>:SENT:TREND<y>:CURSOR:
MODE?
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:SENT:TREND1:CURSOR:
MODE ON
:SERIALBUS1:SENT:TREND1:CURSOR:MODE?
-> :SERIALBUS1:SENT:TREND1:CURSOR:
MODE 1

:SERIALbus<x>:SENT:TREND<y>:CURSOR:POSITION<z>

機能 SENT 信号解析のトレンド表示の各カーソルの位置を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:TREND<y>:CURSOR:
POSITION<z> {<NRf>}
:SERIALbus<x>:SENT:TREND<y>:CURSOR:
POSITION<z>?
<x> = 1 ~ 4
<y> = 1 ~ 4
<z> = 1, 2
<NRf> = -5 ~ 5(div)、(10div/ 表示レコード長) ステップ

例 :SERIALBUS1:SENT:TREND1:CURSOR:
POSITION1 2
:SERIALBUS1:SENT:TREND1:CURSOR:
POSITION1?
-> :SERIALBUS1:SENT:TREND1:CURSOR:
POSITION1 2.00E+00

:SERIALbus<x>:SENT:TREND<y>:CURSOR:T<z>:VALUE?

機能 SENT 信号解析のトレンド表示のカーソルの時間軸値を問い合わせます。

構文 :SERIALbus<x>:SENT:TREND<y>:CURSOR:
T<z>:VALUE?
<x> = 1 ~ 4
<y> = 1 ~ 4
<z> = 1, 2

:SERIALbus<x>:SENT:TREND<y>:CURSOR:V<z>:VALUE?

機能 SENT 信号解析のトレンド表示のカーソルの垂直軸値を問い合わせます。

構文 :SERIALbus<x>:SENT:TREND<y>:CURSOR:
V<z>:VALUE?
<x> = 1 ~ 4
<y> = 1 ~ 4
<z> = 1, 2

:SERIALbus<x>:SENT:TREND<y>:DISPLAY

機能 SENT 信号解析のトレンドを表示する (ON)/ 表示しない (OFF) を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:TREND<y>:
DISPLAY {<Boolean>}
:SERIALbus<x>:SENT:TREND<y>:DISPLAY?
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:SENT:TREND1:DISPLAY ON
:SERIALBUS1:SENT:TREND1:DISPLAY?
-> :SERIALBUS1:SENT:TREND1:DISPLAY 1

:SERIALbus<x>:SENT:TREND<y>:FCONTROL

機能 SENT 信号解析のトレンド表示 FrameControl 値を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:TREND<y>:
FCONTROL {<文字列>}
:SERIALbus<x>:SENT:TREND<y>:
FCONTROL?
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:SENT:TREND1:FCONTROL "1"
:SERIALBUS1:SENT:TREND1:FCONTROL?
-> :SERIALBUS1:SENT:TREND1:
FCONTROL "1"

5.25 SERIALbus グループ

:SERIALbus<x>:SENT:TREND<y>:HRANGE

機能 SENT 信号解析の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。

構文 :SERIALbus<x>:SENT:TREND<y>:
HRANGE {MAIN|Z1|Z2}
:SERIALbus<x>:SENT:TREND<y>:HRANGE?
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:SENT:TREND1:HRANGE MAIN
:SERIALBUS1:SENT:TREND1:HRANGE?
-> :SERIALBUS1:SENT:TREND1:
HRANGE MAIN

:SERIALbus<x>:SENT:TREND<y>:SID

機能 SENT 信号解析のトレンド表示の Slow CH の ID を 16 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:TREND<y>:
SID {<文字列>}
:SERIALbus<x>:SENT:TREND<y>:SID?
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:SENT:TREND1:SID "01"
:SERIALBUS1:SENT:TREND1:SID?
-> :SERIALBUS1:SENT:TREND1:SID "01"

解説

- Slow CH Type 設定が Short のときは、本コマンドパラメータの 1 桁目の値が設定されます。
- Slow CH Type 設定に関わらず、本コマンドパラメータは 2 桁の 16 進数で指定します。

:SERIALbus<x>:SENT:TREND<y>:SOURCE

機能 SENT 信号解析のトレンド表示の対象チャンネルを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:TREND<y>:
SOURCE {FAST|SLOW}
:SERIALbus<x>:SENT:TREND<y>:SOURCE?
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:SENT:TREND1:SOURCE FAST
:SERIALBUS1:SENT:TREND1:SOURCE?
-> :SERIALBUS1:SENT:TREND1:
SOURCE FAST

:SERIALbus<x>:SENT:TREND<y>:UDATA

機能 SENT 信号解析のトレンド表示の Fast CH のユーザー定義データを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:TREND<y>:
UDATA {<NRf>}
:SERIALbus<x>:SENT:TREND<y>:UDATA?
<x> = 1 ~ 4
<y> = 1 ~ 4
<NRf> = 1 ~ 4

例 :SERIALBUS1:SENT:TREND1:UDATA 1
:SERIALBUS1:SENT:TREND1:UDATA?
-> :SERIALBUS1:SENT:TREND1:UDATA 1

:SERIALbus<x>:SENT:TREND<y>:VERTICAL

機能 SENT 信号解析のトレンド表示の垂直レンジを設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:TREND<y>:
VERTICAL {<NRf>,<NRf>}
:SERIALbus<x>:SENT:TREND<y>:
VERTICAL?
<x> = 1 ~ 4
<y> = 1 ~ 4
<NRf> = -1.0000E+31 ~ 1.0000E+31

例 :SERIALBUS1:SENT:TREND1:VERTICAL 1,
-1
:SERIALBUS1:SENT:TREND1:VERTICAL?
-> :SERIALBUS1:SENT:TREND1:
VERTICAL 1.0000000E+00,
-1.0000000E+00

:SERIALbus<x>:SENT:TREND<y>:VTDISPLAY

機能 SENT 信号解析のトレンド表示の VT 波形の表示の ON/OFF を設定 / 問い合わせします。

構文 :SERIALbus<x>:SENT:TREND<y>:
VTDISPLAY {<Boolean>}
:SERIALbus<x>:SENT:TREND<y>:
VTDISPLAY?
<x> = 1 ~ 4
<y> = 1 ~ 4

例 :SERIALBUS1:SENT:TREND1:VTDISPLAY ON
:SERIALBUS1:SENT:TREND1:VTDISPLAY?
-> :SERIALBUS1:SENT:TREND1:
VTDISPLAY 1

:SERIALbus<x>:SOURCE?

機能 解析 / 検索に関する全設定を問い合わせます。

構文 :SERIALbus<x>:SOURCE?
<x> = 1 ~ 4

:SERIALbus<x>:SOURCE:{CHANNEL<y>|MATH<y>}?

機能 対象波形に関する全設定を問い合わせます。

構文 :SERIALbus<x>:SOURCE:{CHANNEL<y>|
MATH<y>}?
<x> = 1 ~ 4
CHANNEL<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:SERIALbus<x>:SOURCE:{CHANNEL<y>|MATH<y>}:HYSTERESIS

機能 対象波形のヒステリシスに関する設定 / 問い合わせをします。

構文 :SERIALbus<x>:SOURCE:{CHANNEL<y>|
MATH<y>}:HYSTERESIS {<NRf>}
:SERIALbus<x>:SOURCE:{CHANNEL<y>|
MATH<y>}:HYSTERESIS?
<x> = 1 ~ 4
CHANNEL<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = 本体ユーザーズマニュアル機能編参照。

:SERIALbus<x>:SOURCE:{CHANNEL<y>|MATH<y>}:LEVEL

機能 対象波形のレベルに関する設定 / 問い合わせをします。

構文 :SERIALbus<x>:SOURCE:{CHANNEL<y>|
MATH<y>}:LEVEL {< 電圧 >, < 電圧 >}
:SERIALbus<x>:SOURCE:{CHANNEL<y>|
MATH<y>}:LEVEL?
<x> = 1 ~ 4
CHANNEL<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
< 電圧 > = 本体ユーザーズマニュアル機能編参照。

例 :SERIALBUS1:SOURCE:CHANNEL1:LEVEL 0V
:SERIALBUS1:SOURCE:CHANNEL1:LEVEL?
-> :SERIALBUS1:SOURCE:CHANNEL1:
LEVEL 0.000E+00

:SERIALbus<x>:SPATTERN?

機能 ユーザー定義バス信号解析 / 検索機能に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPATTERN?
<x> = 1 ~ 4

:SERIALbus<x>:SPATTERN:ANALYZE?

機能 ユーザー定義バス信号解析実行に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPATTERN:ANALYZE?
<x> = 1 ~ 4

:SERIALbus<x>:SPATTERN[:ANALYZE]:SETUP?

機能 ユーザー定義バス信号解析に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPATTERN[:ANALYZE]:
SETUP?
<x> = 1 ~ 4

:SERIALbus<x>:SPATTERN[:ANALYZE]:SETUP:BRATE

機能 ユーザー定義バス信号解析のビットレートを設定 / 問い合わせします。

構文 :SERIALbus<x>:SPATTERN[:ANALYZE]:
SETUP:BRATE {<NRf>}
:SERIALbus<x>:SPATTERN[:ANALYZE]:
SETUP:BRATE?
<x> = 1 ~ 4
<NRf> = 1000 ~ 50000000(bps)

例 :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
BRATE 1000
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
BRATE?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:BRATE 1000

:SERIALbus<x>:SPATTERN[:ANALYZE]:SETUP:CLOCK?

機能 ユーザー定義バス信号解析のクロック信号のすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPATTERN[:ANALYZE]:
SETUP:CLOCK?
<x> = 1 ~ 4

:SERIALbus<x>:SPATTERN[:ANALYZE]:SETUP:CLOCK:MODE

機能 ユーザー定義バス信号解析のクロック信号の有効 / 無効を設定 / 問い合わせします。

構文 :SERIALbus<x>:SPATTERN[:ANALYZE]:
SETUP:CLOCK:MODE {<Boolean>}
:SERIALbus<x>:SPATTERN[:ANALYZE]:
SETUP:CLOCK:MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CLOCK:MODE ON
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CLOCK:MODE?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:CLOCK:MODE 1

5.25 SERIALbus グループ

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:CLOCK:POLarity

機能 ユーザー定義バス信号解析のクロック信号のスロープを設定 / 問い合わせします。

構文 :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CLOCK:POLarity {FALL|RISE}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CLOCK:POLarity?
<x> = 1 ~ 4

例 :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CLOCK:POLARITY FALL
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CLOCK:POLARITY?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:CLOCK:POLARITY FALL

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:CLOCK:SOURce

機能 ユーザー定義バス信号解析のクロック信号を設定 / 問い合わせします。

構文 :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CLOCK:SOURce {<NRf>}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CLOCK:SOURce?
<x> = 1 ~ 4
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CLOCK:SOURCE 1
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CLOCK:SOURCE?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:CLOCK:SOURCE 1

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:CS?

機能 ユーザー定義バス信号解析のチップセレクト信号のすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CS?
<x> = 1 ~ 4

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:CS:ACTive

機能 ユーザー定義バス信号解析のチップセレクト信号のアクティブ状態を設定 / 問い合わせします。

構文 :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CS:ACTive {HIGH|LOW}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CS:ACTive?
<x> = 1 ~ 4

例 :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CS:ACTIVE HIGH
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CS:ACTIVE?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:CS:ACTIVE HIGH

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:CS:SOURce

機能 ユーザー定義バス信号解析のチップセレクト信号を設定 / 問い合わせします。

構文 :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CS:SOURce {<NRf>|NONE}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CS:SOURce?
<x> = 1 ~ 4
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CS:SOURCE 1
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CS:SOURCE?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:CS:SOURCE 1

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:DATA?

機能 ユーザー定義バス信号解析のデータ信号のすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:DATA?
<x> = 1 ~ 4

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:DATA:ACTive

機能 ユーザー定義バス信号解析のデータ信号のアクティブ状態を設定 / 問い合わせします。

構文 :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:DATA:ACTive {HIGH|LOW}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:DATA:ACTive?
<x> = 1 ~ 4

例 :SERIALBUS1:SPATtern:ANALyze:SETup:
DATA:ACTive HIGH
:SERIALBUS1:SPATtern:ANALyze:SETup:
DATA:ACTive?
-> :SERIALBUS1:SPATtern:ANALyze:
SETup:DATA:ACTive HIGH

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:DATA:SOURce

機能 ユーザー定義バス信号解析のデータ信号を設定 / 問い合わせします。

構文 :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:DATA:SOURce {<Nrf>}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:DATA:SOURce?
<x> = 1 ~ 4

例 :SERIALBUS1:SPATtern:ANALyze:SETup:
DATA:SOURce 1
:SERIALBUS1:SPATtern:ANALyze:SETup:
DATA:SOURce?
-> :SERIALBUS1:SPATtern:ANALyze:
SETup:DATA:SOURce 1

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:LATCh?

機能 ユーザー定義バス信号解析のラッチ信号のすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:LATCh?
<x> = 1 ~ 4

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:LATCh:POLarity

機能 ユーザー定義バス信号解析のラッチ信号のスロープを設定 / 問い合わせします。

構文 :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:LATCh:POLarity {FALL|RISE}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:LATCh:POLarity?
<x> = 1 ~ 4

例 :SERIALBUS1:SPATtern:ANALyze:SETup:
LATCh:POLarity FALL
:SERIALBUS1:SPATtern:ANALyze:SETup:
LATCh:POLarity?
-> :SERIALBUS1:SPATtern:ANALyze:
SETup:LATCh:POLarity FALL

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:LATCh:SOURce

機能 ユーザー定義バス信号解析のラッチ信号を設定 / 問い合わせします。

構文 :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:LATCh:SOURce {<Nrf>|NONE}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:LATCh:SOURce?
<x> = 1 ~ 4

例 :SERIALBUS1:SPATtern:ANALyze:SETup:
LATCh:SOURce 1
:SERIALBUS1:SPATtern:ANALyze:SETup:
LATCh:SOURce?
-> :SERIALBUS1:SPATtern:ANALyze:
SETup:LATCh:SOURce 1

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:SPOint

機能 ユーザー定義バス信号解析の解析開始点を設定 / 問い合わせします。

構文 :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:SPOint {<Nrf>}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:SPOint?
<x> = 1 ~ 4

例 :SERIALBUS1:SPATtern:ANALyze:SETup:
SPOint -5
:SERIALBUS1:SPATtern:ANALyze:SETup:
SPOint?
-> :SERIALBUS1:SPATtern:ANALyze:
SETup:SPOint -5.0000000E+00

5.25 SERIALbus グループ

:SERIALbus<x>:SPATtern:SEARCh?

機能 ユーザー定義バス信号検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPATtern:SEARCh?
<x> = 1 ~ 4

:SERIALbus<x>:SPATtern:SEARCh:ABORt

機能 ユーザー定義バス信号検索を中止します。

構文 :SERIALbus<x>:SPATtern:SEARCh:ABORt
<x> = 1 ~ 4

例 :SERIALBUS1:SPATtern:SEARCH:ABORT

:SERIALbus<x>:SPATtern:SEARCh:EXECute

機能 ユーザー定義バス信号検索を実行します。

構文 :SERIALbus<x>:SPATtern:SEARCh:
EXECute
<x> = 1 ~ 4

例 :SERIALBUS1:SPATtern:SEARCH:EXECUTE

:SERIALbus<x>:SPATtern:SEARCh:SELeCt

機能 ユーザー定義バス信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。

構文 :SERIALbus<x>:SPATtern:SEARCh:
SELeCt {<NRf>|MAXimum}
:SERIALbus<x>:SPATtern:SEARCh:
SELeCt?
<x> = 1 ~ 4
<NRf> = 0 ~ 49999

例 :SERIALBUS1:SPATtern:SEARCH:SELECT 1
:SERIALBUS1:SPATtern:SEARCH:SELECT?
-> :SERIALBUS1:SPATtern:SEARCH:
SELECT 1.50000000

解説 検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:SERIALbus<x>:SPATtern:SEARCh:SELeCt? MAXimum

機能 ユーザー定義バス信号検索の検索点の総数を問い合わせます。

構文 :SERIALbus<x>:SPATtern:SEARCh:
SELeCt? MAXimum
<x> = 1 ~ 4

例 :SERIALBUS1:SPATtern:SEARCH:
SELECT? MAXIMUM
-> :SERIALBUS1:SPATtern:SEARCH:
SELECT 100

解説 検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:SERIALbus<x>:SPATtern:SEARCh:SEtUp?

機能 ユーザー定義バス信号検索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 :SERIALbus<x>:SPATtern:SEARCh:SEtUp?
<x> = 1 ~ 4

:SERIALbus<x>:SPATtern:SEARCh:SEtUp:BITSize

機能 ユーザー定義バス信号検索のビット長を設定 / 問い合わせします。

構文 :SERIALbus<x>:SPATtern:SEARCh:SEtUp:
BITSize {<NRf>}
:SERIALbus<x>:SPATtern:SEARCh:SEtUp:
BITSize?
<x> = 1 ~ 4
<NRf> = 1 ~ 128

例 :SERIALBUS1:SPATtern:SEARCH:SETUP:
BITSIZE 1
:SERIALBUS1:SPATtern:SEARCH:SETUP:
BITSIZE?
-> :SERIALBUS1:SPATtern:SEARCH:
SETUP:BITSIZE 1

:SERIALbus<x>:SPATtern:SEARCh:SEtUp:HEXa

機能 ユーザー定義バス信号検索のデータ条件を 16 進数で設定します。

構文 :SERIALbus<x>:SPATtern:SEARCh:SEtUp:
HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:SPATtern:SEARCH:SETUP:
HEXA "12"

:SERIALbus<x>:SPATtern:SEARCh:SEtUp:PATtern

機能 ユーザー定義バス信号検索のデータ条件を 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:SPATtern:SEARCh:SEtUp:
PATtern {<文字列>}
:SERIALbus<x>:SPATtern:SEARCh:SEtUp:
PATtern?
<x> = 1 ~ 4

例 :SERIALBUS1:SPATtern:SEARCH:SETUP:
PATTERN "00110101"
:SERIALBUS1:SPATtern:SEARCH:SETUP:
PATTERN?
-> :SERIALBUS1:SPATtern:SEARCH:
SETUP:PATTERN "00110101"

:SERIALbus<x>:SPI?

機能 SPI / バス信号解析 / 検索機能に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPI?
<x> = 1 ~ 4

:SERIALbus<x>:SPI:ANALyze?

機能 SPIバス信号解析に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPI:ANALyze?
<x> = 1 ~ 4

:SERIALbus<x>:SPI[:ANALyze]:SETup?

機能 SPIバス信号解析のバス設定に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPI[:ANALyze]:SETup?
<x> = 1 ~ 4

:SERIALbus<x>:SPI[:ANALyze]:SETup:BITOrder

機能 SPIバス信号解析データのビットオーダーを設定 / 問い合わせします。

構文 :SERIALbus<x>:SPI[:ANALyze]:SETup:
BITOrder {MSBFirst|LSBFirst}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
BITOrder?
<x> = 1 ~ 4

例 :SERIALBUS1:SPI:ANALYZE:SETUP:
BITORDER MSBFIRST
:SERIALBUS1:SPI:ANALYZE:SETUP:
BITORDER?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:
BITORDER MSBFIRST

:SERIALbus<x>:SPI[:ANALyze]:SETup:CLOCK?

機能 SPIバス信号解析のクロック信号のすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPI[:ANALyze]:SETup:
CLOCK?
<x> = 1 ~ 4

:SERIALbus<x>:SPI[:ANALyze]:SETup:CLOCK:POLarity

機能 SPIバス信号解析のクロック信号のスロープを設定 / 問い合わせします。

構文 :SERIALbus<x>:SPI[:ANALyze]:SETup:
CLOCK:POLarity {FALL|RISE}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
CLOCK:POLarity?
<x> = 1 ~ 4

例 :SERIALBUS1:SPI:ANALYZE:SETUP:CLOCK:
POLARITY FALL
:SERIALBUS1:SPI:ANALYZE:SETUP:CLOCK:
POLARITY?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:
CLOCK:POLARITY FALL

:SERIALbus<x>:SPI[:ANALyze]:SETup:CLOCK:SOURce

機能 SPIバス信号解析のクロック信号を設定 / 問い合わせします。

構文 :SERIALbus<x>:SPI[:ANALyze]:SETup:
CLOCK:SOURce {<NRF>|MATH<y>|PODA<y>|
PODB<y>|PODC<y>|PODD<y>}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
CLOCK:SOURce?
<x> = 1 ~ 4
<NRF> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<y>、PODB<y>、PODC<y>、PODD<y>
の <y> = 0 ~ 7

例 :SERIALBUS1:SPI:ANALYZE:SETUP:CLOCK:
SOURCE 1
:SERIALBUS1:SPI:ANALYZE:SETUP:CLOCK:
SOURCE?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:
CLOCK:SOURCE 1

:SERIALbus<x>:SPI[:ANALyze]:SETup:CS?

機能 SPIバス信号解析のチップセレクト信号のすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPI[:ANALyze]:SETup:
CS?
<x> = 1 ~ 4

:SERIALbus<x>:SPI[:ANALyze]:SETup:CS:ACTIVE

機能 SPIバス信号解析のチップセレクト信号のアクティブ状態を設定 / 問い合わせします。

構文 :SERIALbus<x>:SPI[:ANALyze]:SETup:
CS:ACTIVE {HIGH|LOW}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
CS:ACTIVE?
<x> = 1 ~ 4

例 :SERIALBUS1:SPI:ANALYZE:SETUP:CS:
ACTIVE HIGH
:SERIALBUS1:SPI:ANALYZE:SETUP:CS:
ACTIVE?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:CS:
ACTIVE HIGH

5.25 SERIALbus グループ

:SERIALbus<x>:SPI[:ANALyze]:SETup:CS:SOURce

機能	SPI バス信号解析のチップセレクト信号を設定 / 問い合わせします。
構文	:SERIALbus<x>:SPI[:ANALyze]:SETup: CS:SOURce {<NRf> MATH<y> PODA<y> PODB<y> PODC<y> PODD<y>} :SERIALbus<x>:SPI[:ANALyze]:SETup: CS:SOURce? <x> = 1 ~ 4 <NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4) MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4) PODA<y>、PODB<y>、PODC<y>、PODD<y> の <y> = 0 ~ 7
例	:SERIALBUS1:SPI:ANALYZE:SETUP:CS: SOURCE 1 :SERIALBUS1:SPI:ANALYZE:SETUP:CS: SOURCE? -> :SERIALBUS1:SPI:ANALYZE:SETUP:CS: SOURCE 1

:SERIALbus<x>:SPI[:ANALyze]:SETup:DATA<y>?

機能	SPI バス信号解析のデータ信号のすべての設定値を問い合わせます。
構文	:SERIALbus<x>:SPI[:ANALyze]:SETup: DATA<y>? <x> = 1 ~ 4 <y> = 1、2

:SERIALbus<x>:SPI[:ANALyze]:SETup:DATA<y>:SOURce

機能	SPI バス信号解析のデータ信号を設定 / 問い合わせします。
構文	:SERIALbus<x>:SPI[:ANALyze]:SETup: DATA<y>:SOURce {<NRf> MATH<y> PODA<y> PODB<y> PODC<y> PODD<y>} :SERIALbus<x>:SPI[:ANALyze]:SETup: DATA<y>:SOURce? <x> = 1 ~ 4 <y> = 1、2 <NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4) MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4) PODA<y>、PODB<y>、PODC<y>、PODD<y> の <y> = 0 ~ 7
例	:SERIALBUS1:SPI:ANALYZE:SETUP:DATA1: SOURCE 1 :SERIALBUS1:SPI:ANALYZE:SETUP:DATA1: SOURCE? -> :SERIALBUS1:SPI:ANALYZE:SETUP: DATA1:SOURCE 1

:SERIALbus<x>:SPI[:ANALyze]:SETup:FIELD

機能	SPI バス信号解析のデータの Field Size を設定 / 問い合わせします。
構文	:SERIALbus<x>:SPI[:ANALyze]:SETup: FIELD {<NRf>} :SERIALbus<x>:SPI[:ANALyze]:SETup: FIELD? <x> = 1 ~ 4 <NRf> = 4 ~ 32
例	:SERIALBUS1:SPI:ANALYZE:SETUP: FIELD 4 :SERIALBUS1:SPI:ANALYZE:SETUP:FIELD? -> :SERIALBUS1:SPI:ANALYZE:SETUP: FIELD 4

SERIALbus<x>:SPI[:ANALyze]:SETup:GROUPing

機能	SPI バス信号解析のチップセレクト信号がない場合のグルーピングの ON/OFF を設定 / 問い合わせします。
構文	:SERIALbus<x>:SPI[:ANALyze]:SETup: GROUPing {<Boolean>} :SERIALbus<x>:SPI[:ANALyze]:SETup: GROUPing? <x> = 1 ~ 4
例	:SERIALBUS1:SPI:ANALYZE:SETUP: GROUPING ON :SERIALBUS1:SPI:ANALYZE:SETUP: GROUPING? -> :SERIALBUS1:SPI:ANALYZE:SETUP: GROUPING 1

:SERIALbus<x>:SPI[:ANALyze]:SETup:ITIME

機能	SPI バス信号解析のチップセレクト信号がない場合のアイドルタイムを設定 / 問い合わせします。
構文	:SERIALbus<x>:SPI[:ANALyze]:SETup: ITIME {<時間>} :SERIALbus<x>:SPI[:ANALyze]:SETup: ITIME? <x> = 1 ~ 4 <時間> = 10ns ~ 1ms
例	:SERIALBUS1:SPI:ANALYZE:SETUP: ITIME 2US :SERIALBUS1:SPI:ANALYZE:SETUP:ITIME? -> :SERIALBUS1:SPI:ANALYZE:SETUP: ITIME 2.00000E-06

:SERIALbus<x>:SPI[:ANALyze]:SETup:MODE

機能 SPIバス信号解析のデータ信号の結線方式(3線式/4線式)を設定/問い合わせします。

構文 :SERIALbus<x>:SPI[:ANALyze]:SETup:
MODE {WIRE3|WIRE4}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
MODE?

例 <x> = 1 ~ 4
:SERIALBUS1:SPI:ANALYZE:SETUP:
MODE WIRE3
:SERIALBUS1:SPI:ANALYZE:SETUP:MODE?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:
MODE WIRE3

:SERIALbus<x>:SPI[:ANALyze]:SETup:MSBLSb

機能 SPIバス信号解析のデータのMSB/LSBのビットを設定/問い合わせします。

構文 :SERIALbus<x>:SPI[:ANALyze]:SETup:
MSBLSb {<NRF>,<NRF>}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
MSBLSb?

例 <x> = 1 ~ 4
<NRF> = 0 ~ 31
:SERIALBUS1:SPI:ANALYZE:SETUP:
MSBLSB 7,0
:SERIALBUS1:SPI:ANALYZE:SETUP:
MSBLSB?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:
MSBLSB 7,0

:SERIALbus<x>:SPI[:ANALyze]:SETup:TYPE

機能 SPIバス信号解析のデータのタイプを設定/問い合わせします。

構文 :SERIALbus<x>:SPI[:ANALyze]:SETup:
TYPE {BYTE|USER}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
TYPE?

例 <x> = 1 ~ 4
:SERIALBUS1:SPI:ANALYZE:SETUP:
TYPE BYTE
:SERIALBUS1:SPI:ANALYZE:SETUP:TYPE?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:TYPE
BYTE

:SERIALbus<x>:SPI:DETAil?

機能 SPIバス信号の解析結果リストに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPI:DETAil?
<x> = 1 ~ 4

:SERIALbus<x>:SPI:DETAil:DISPlay

機能 SPIバス信号の解析結果リストの表示モードを設定/問い合わせします。

構文 :SERIALbus<x>:SPI:DETAil:
DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:SPI:DETAil:DISPlay?
<x> = 1 ~ 4

例 :SERIALBUS1:SPI:DETAIL:DISPLAY FULL
:SERIALBUS1:SPI:DETAIL:DISPLAY?
-> :SERIALBUS1:SPI:DETAIL:
DISPLAY FULL

:SERIALbus<x>:SPI:DETAil:LIST:ITEM?

機能 SPIバス信号の解析結果リストに表示される項目を問い合わせます。

構文 :SERIALbus<x>:SPI:DETAil:LIST:ITEM?
<x> = 1 ~ 4

例 :SERIALBUS1:SPI:DETAIL:LIST:ITEM?
-> :SERIALBUS1:SPI:DETAIL:LIST:
ITEM "No.,Time(ms),Data,,"

:SERIALbus<x>:SPI:DETAil:LIST:VALue?

機能 SPIバス信号の解析結果リストの指定した解析番号の全データを問い合わせます。

構文 :SERIALbus<x>:SPI:DETAil:LIST:
VALue? {<NRF>|MAXimum|MINimum}
<x> = 1 ~ 4

例 <NRF> = 本体ユーザーズマニュアル機能編参照。
:SERIALBUS1:SPI:DETAIL:LIST:VALUE? 0
-> :SERIALBUS1:SPI:DETAIL:LIST:
VALUE "0,0.077002,Data1,
00 00 00 00 00 00 00 00,"

:SERIALbus<x>:SPI:SEARCh?

機能 SPIバス信号の検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPI:SEARCh?
<x> = 1 ~ 4

:SERIALbus<x>:SPI:SEARCh:ABORt

機能 SPIバス信号検索を中止します。

構文 :SERIALbus<x>:SPI:SEARCh:ABORt
<x> = 1 ~ 4

例 :SERIALBUS1:SPI:SEARCH:ABORT

:SERIALbus<x>:SPI:SEARCh:EXECute

機能 SPIバス信号検索を実行します。

構文 :SERIALbus<x>:SPI:SEARCh:EXECute
<x> = 1 ~ 4

例 :SERIALBUS1:SPI:SEARCH:EXECUTE

5.25 Serialbus グループ

:Serialbus<x>:SPI:SEARCh:SElect

機能 SPI バス信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。

構文 :Serialbus<x>:SPI:SEARCh:
SElect {<NRf>|MAXimum}
:Serialbus<x>:SPI:SEARCh:SElect?
<x> = 1 ~ 4
<NRf> = 0 ~ 49999

例 :SERIALBUS1:SPI:SEARCH:SELECT 1
:SERIALBUS1:SPI:SEARCH:SELECT?
-> :SERIALBUS1:SPI:SEARCH:
SELECT 1.50000000

解説 検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:Serialbus<x>:SPI:SEARCh:SElect? MAXimum

機能 SPI バス信号検索の検索点の総数を問い合わせます。

構文 :Serialbus<x>:SPI:SEARCh:
SElect? MAXimum
<x> = 1 ~ 4

例 :SERIALBUS1:SPI:SEARCH:
SELECT? MAXIMUM
-> :SERIALBUS1:SPI:SEARCH:SELECT 100

解説 検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:Serialbus<x>:SPI:SEARCh:SETup?

機能 SPI バス信号検索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 :Serialbus<x>:SPI:SEARCh:SETup?
<x> = 1 ~ 4

:Serialbus<x>:SPI:SEARCh[:SETup]:DATA<y>?

機能 SPI バス信号検索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 :Serialbus<x>:SPI:SEARCh[:SETup]:
DATA<y>?
<x> = 1 ~ 4
<y> = 1, 2

:Serialbus<x>:SPI:SEARCh[:SETup]:DATA<y>:BCount

機能 SPI バス信号検索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 :Serialbus<x>:SPI:SEARCh[:SETup]:
DATA<y>:BCount {<NRf>}
:Serialbus<x>:SPI:SEARCh[:SETup]:
DATA<y>:BCount?
<x> = 1 ~ 4
<y> = 1, 2
<NRf> = 0 ~ 9999

例 :SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
BCOUNT 0
:SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
BCOUNT?
-> :SERIALBUS1:SPI:SEARCH:SETUP:
DATA1:BCOUNT 0

:Serialbus<x>:SPI:SEARCh[:SETup]:DATA<y>:CONDition

機能 SPI バス信号検索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 :Serialbus<x>:SPI:SEARCh[:SETup]:
DATA<y>:CONDition {FALSE|TRUE}
:Serialbus<x>:SPI:SEARCh[:SETup]:
DATA<y>:CONDition?
<x> = 1 ~ 4
<y> = 1, 2

例 :SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
CONDITION DONTCARE
:SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
CONDITION?
-> :SERIALBUS1:SPI:SEARCH:SETUP:
DATA1:CONDITION DONTCARE

:Serialbus<x>:SPI:SEARCh[:SETup]:DATA<y>:DBYTe

機能 SPI バス信号検索の各データのデータサイズ (バイト数) を設定 / 問い合わせします。

構文 :Serialbus<x>:SPI:SEARCh[:SETup]:
DATA<y>:DBYTe {<NRf>}
:Serialbus<x>:SPI:SEARCh[:SETup]:
DATA<y>:DBYTe?
<x> = 1 ~ 4
<y> = 1, 2
<NRf> = 1 ~ 4

例 :SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
DBYTE 1
:SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
DBYTE?
-> :SERIALBUS1:SPI:SEARCH:SETUP:
DATA1:DBYTE 1

:SERIALbus<x>:SPI:SEARCH[:SETUP]:DATA<y>:DPATTERN?

機能 SPIバス信号検索の各データに関わるすべての設定値を問い合わせます。

構文 :SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:DPATTERN?
<x> = 1 ~ 4
<y> = 1、2

:SERIALbus<x>:SPI:SEARCH[:SETUP]:DATA<y>:DPATTERN:HEXA

機能 SPIバス信号検索の各データを16進数で設定します。

構文 :SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:DPATTERN:HEXA {<文字列>}
<x> = 1 ~ 4
<y> = 1、2

例 :SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
DPATTERN:HEXA "AB"

解説 「:SERIALbus<x>:SPI:SEARCH[:SETUP]:DATA<y>:DBYTE」の設定値によって設定できるバイト数が増減します。

:SERIALbus<x>:SPI:SEARCH[:SETUP]:DATA<y>:DPATTERN:PATTERN

機能 SPIバス信号検索の各データを2進数で設定/問い合わせします。

構文 :SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:DPATTERN:PATTERN {<文字列>}
:SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:DPATTERN:PATTERN?
<x> = 1 ~ 4
<y> = 1、2

例 :SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
DPATTERN:PATTERN "11001010"
:SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
DPATTERN:PATTERN?

-> :SERIALBUS1:SPI:SEARCH:SETUP:
DATA1:DPATTERN:PATTERN "11001010"

解説 「:SERIALbus<x>:SPI:SEARCH[:SETUP]:DATA<y>:DBYTE」の設定値によって設定できるバイト数が増減します。

:SERIALbus<x>:SPI:SEARCH[:SETUP]:DATA<y>:MODE

機能 SPIバス信号検索のデータ条件(有効/無効)の設定/問い合わせします。

構文 :SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:MODE {<Boolean>}
:SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:MODE?
<x> = 1 ~ 4
<y> = 1、2

例 :SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
MODE ON
:SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
MODE?
-> :SERIALBUS1:SPI:SEARCH:SETUP:
DATA1:MODE 1

:SERIALbus<x>:TYPE

機能 検索タイプを設定/問い合わせします。

構文 :SERIALbus<x>:TYPE {CANBus|CANFDbus|
CXPIbus|FLEXray|I2Cbus|LINbus|PSI5|
SENT|SPATTERN|SPIbus|UART}
:SERIALbus<x>:TYPE?
<x> = 1 ~ 4

例 :SERIALBUS1:TYPE CANBUS
:SERIALBUS1:TYPE?
-> :SERIALBUS1:TYPE CANBUS

:SERIALbus<x>:UART?

機能 UART信号の解析/検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:UART?
<x> = 1 ~ 4

:SERIALbus<x>:UART:ANALYZE?

機能 UART信号解析に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:UART:ANALYZE?
<x> = 1 ~ 4

:SERIALbus<x>:UART[:ANALYZE]:SETUP?

機能 UART信号解析のバス設定に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:UART[:ANALYZE]:SETUP?
<x> = 1 ~ 4

5.25 SERIALbus グループ

:SERIALbus<x>:UART[:ANALyze]:SETup:BITOrder

機能 UART 信号解析のビットオーダーを設定 / 問い合わせします。

構文 :SERIALbus<x>:UART[:ANALyze]:SETup:BITOrder {MSBFirst|LSBFirst}
:SERIALbus<x>:UART[:ANALyze]:SETup:BITOrder?
<x> = 1 ~ 4

例 :SERIALBUS1:UART:ANALYZE:SETUP:BITORDER MSBFIRST
:SERIALBUS1:UART:ANALYZE:SETUP:BITORDER?
-> :SERIALBUS1:UART:ANALYZE:SETUP:BITORDER MSBFIRST

:SERIALbus<x>:UART[:ANALyze]:SETup:BRATe

機能 UART 信号解析のビットレート (データ転送速度) を設定 / 問い合わせします。

構文 :SERIALbus<x>:UART[:ANALyze]:SETup:BRATe {<Nrf>|USER,<Nrf>}
:SERIALbus<x>:UART[:ANALyze]:SETup:BRATe?
<x> = 1 ~ 4
<Nrf> = 1200、2400、4800、9600、19200

USER の <NRF> = 本体ユーザズマニュアル機能編参照。

例 :SERIALBUS1:UART:ANALYZE:SETUP:BRATE 2400
:SERIALBUS1:UART:ANALYZE:SETUP:BRATE?
-> :SERIALBUS1:UART:ANALYZE:SETUP:BRATE 2400

:SERIALbus<x>:UART[:ANALyze]:SETup:BSpace

機能 UART 信号解析のバイトスペースを設定 / 問い合わせします。

構文 :SERIALbus<x>:UART[:ANALyze]:SETup:BSpace {<時間>|MINimum}
:SERIALbus<x>:UART[:ANALyze]:SETup:BSpace?
<x> = 1 ~ 4
<時間> = 本体ユーザズマニュアル機能編参照。

例 :SERIALBUS1:UART:ANALYZE:SETUP:BSpace 0.1S
:SERIALBUS1:UART:ANALYZE:SETUP:BSpace?
-> :SERIALBUS1:UART:ANALYZE:SETUP:BSpace 1.000E-01

:SERIALbus<x>:UART[:ANALyze]:SETup:GRouping

機能 UART 信号解析のグルーピングの ON/OFF を設定 / 問い合わせします。

構文 :SERIALbus<x>:UART[:ANALyze]:SETup:GRouping {<Boolean>}
:SERIALbus<x>:UART[:ANALyze]:SETup:GRouping?
<x> = 1 ~ 4

例 :SERIALBUS1:UART:ANALYZE:SETUP:GROUPING ON
:SERIALBUS1:UART:ANALYZE:SETUP:GROUPING?
-> :SERIALBUS1:UART:ANALYZE:SETUP:GROUPING 1

:SERIALbus<x>:UART[:ANALyze]:SETup:POLarity

機能 UART 信号解析の極性を設定 / 問い合わせします。

構文 :SERIALbus<x>:UART[:ANALyze]:SETup:POLarity {NEGative|POSitive}
:SERIALbus<x>:UART[:ANALyze]:SETup:POLarity?
<x> = 1 ~ 4

例 :SERIALBUS1:UART:ANALYZE:SETUP:POLARITY NEGATIVE
:SERIALBUS1:UART:ANALYZE:SETUP:POLARITY?
-> :SERIALBUS1:UART:ANALYZE:SETUP:POLARITY NEGATIVE

:SERIALbus<x>:UART[:ANALyze]:SETup:SOURce

機能 UART 信号解析の信号を設定 / 問い合わせします。

構文 :SERIALbus<x>:UART[:ANALyze]:SETup:SOURce {<Nrf>|MATH<y>|PODA<y>|PODB<y>|PODC<y>|PODD<y>}
:SERIALbus<x>:UART[:ANALyze]:SETup:SOURce?
<x> = 1 ~ 4
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<y>、PODB<y>、PODC<y>、PODD<y> の <y> = 0 ~ 7

例 :SERIALBUS1:UART:ANALYZE:SETUP:SOURCE 1
:SERIALBUS1:UART:ANALYZE:SETUP:SOURCE?
-> :SERIALBUS1:UART:ANALYZE:SETUP:SOURCE 1

:SERIALbus<x>:UART[:ANALyze]:SETup:SPOint

機能 UART 信号解析のサンプルポイントを設定 / 問い合わせします。

構文 :SERIALbus<x>:UART[:ANALyze]:SETup:
SPOint {<NRf>}
:SERIALbus<x>:UART[:ANALyze]:SETup:
SPOint?
<x> = 1 ~ 4
<NRf> = 18.8 ~ 90.6

例 :SERIALBUS1:UART:ANALYZE:SETUP:
SPOINT 18.8
:SERIALBUS1:UART:ANALYZE:SETUP:
SPOINT?
-> :SERIALBUS1:UART:ANALYZE:SETUP:
SPOINT 18.8

:SERIALbus<x>:UART:DETail?

機能 UART 信号の解析結果リストに関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:UART:DETail?
<x> = 1 ~ 4

:SERIALbus<x>:UART:DETail:DISPlay

機能 UART 信号の解析結果リストの表示モードを設定 / 問い合わせします。

構文 :SERIALbus<x>:UART:DETail:
DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:UART:DETail:DISPlay?
<x> = 1 ~ 4

例 :SERIALBUS1:UART:DETAIL:DISPLAY FULL
:SERIALBUS1:UART:DETAIL:DISPLAY?
-> :SERIALBUS1:UART:DETAIL:
DISPLAY FULL

:SERIALbus<x>:UART:DETail:DTIME

機能 UART 信号の解析結果リストのグルーピングを設定 / 問い合わせします。

構文 :SERIALbus<x>:UART:DETail:
DTIME {<Boolean>}
:SERIALbus<x>:UART:DETail:DTIME?
<x> = 1 ~ 4

例 :SERIALBUS1:UART:DETAIL:DTIME ON
:SERIALBUS1:UART:DETAIL:DTIME?
-> :SERIALBUS1:UART:DETAIL:DTIME 1

:SERIALbus<x>:UART:DETail:LIST:ITEM?

機能 UART 信号の解析結果リストに表示される項目を問い合わせます。

構文 :SERIALbus<x>:UART:DETail:LIST:ITEM?
<x> = 1 ~ 4

例 :SERIALBUS1:UART:DETAIL:LIST:ITEM?
-> :SERIALBUS1:UART:DETAIL:LIST:
ITEM "No.,Time(ms),Data(HEX),
Data(ASCII),Information"

:SERIALbus<x>:UART:DETail:LIST:VALue?

機能 UART 信号の解析結果リストの指定した解析番号の全データを問い合わせます。

構文 :SERIALbus<x>:UART:DETail:LIST:
VALue? {<NRf>|MAXimum|MINimum}
<x> = 1 ~ 4

例 :SERIALBUS1:UART:DETAIL:LIST:
VALUE? 0
-> :SERIALBUS1:UART:DETAIL:LIST:
VALUE "0,-0.5720,3A 74 69 D2,":ti.,,,

:SERIALbus<x>:UART:SEARCh?

機能 UART 信号の検索に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:UART:SEARCh?
<x> = 1 ~ 4

:SERIALbus<x>:UART:SEARCh:ABORT

機能 UART 信号検索を中止します。

構文 :SERIALbus<x>:UART:SEARCh:ABORT
<x> = 1 ~ 4

例 :SERIALBUS1:UART:SEARCH:ABORT

:SERIALbus<x>:UART:SEARCh:EXECute

機能 UART 信号検索を実行します。

構文 :SERIALbus<x>:UART:SEARCh:EXECute
<x> = 1 ~ 4

例 :SERIALBUS1:UART:SEARCH:EXECUTE

5.25 SERIALbus グループ

:SERIALbus<x>:UART:SEARCH:SElect

機能 UART 信号検索のズームウィンドウに表示する検索点の設定とその検索点のズーム位置を問い合わせます。

構文 :SERIALbus<x>:UART:SEARCH:SElect {<Nrf>|MAXimum}
:SERIALbus<x>:UART:SEARCH:SElect?<x> = 1 ~ 4<Nrf> = 0 ~ 49999

例 :SERIALBUS1:UART:SEARCH:SELECT 1
:SERIALBUS1:UART:SEARCH:SELECT?
-> :SERIALBUS1:UART:SEARCH:SELECT 1.50000000

解説 検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:SERIALbus<x>:UART:SEARCH:SElect?MAXimum

機能 UART 信号検索の検索点の総数を問い合わせます。

構文 :SERIALbus<x>:UART:SEARCH:SElect? MAXimum<x> = 1 ~ 4

例 :SERIALBUS1:UART:SEARCH:SELECT? MAXIMUM
-> :SERIALBUS1:UART:SEARCH:SELECT 100

解説 検索された位置が無い場合は、"NAN" が問い合わせの応答として返ってきます。

:SERIALbus<x>:UART:SEARCH:SETup?

機能 UART 信号検索の検索条件に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:UART:SEARCH:SETup?<x> = 1 ~ 4

:SERIALbus<x>:UART:SEARCH:SETup:DATA?

機能 UART 信号検索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 :SERIALbus<x>:UART:SEARCH:SETup:DATA?<x> = 1 ~ 4

:SERIALbus<x>:UART:SEARCH:SETup:DATA:AScii

機能 UART 信号検索のデータを ASCII で設定します。

構文 :SERIALbus<x>:UART:SEARCH:SETup:DATA:AScii {<文字列>}<x> = 1 ~ 4

例 :SERIALBUS1:UART:SEARCH:SETUP:DATA:ASCII "TEST"

:SERIALbus<x>:UART:SEARCH:SETup:DATA:CONDITION

機能 UART 信号検索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 :SERIALbus<x>:UART:SEARCH:SETup:DATA:CONDITION {DONTcare|TRUE}<x> = 1 ~ 4

例 :SERIALBUS1:UART:SEARCH:SETUP:DATA:CONDITION DONTCARE
:SERIALBUS1:UART:SEARCH:SETUP:DATA:CONDITION?
-> :SERIALBUS1:UART:SEARCH:SETUP:DATA:CONDITION DONTCARE

:SERIALbus<x>:UART:SEARCH:SETup:DATA:CSENSitive

機能 UART 信号検索の ASCII データの大文字、小文字を区別する / しないの設定 / 問い合わせします。

構文 :SERIALbus<x>:UART:SEARCH:SETup:DATA:CSENSitive {<Boolean>}<x> = 1 ~ 4

例 :SERIALBUS1:UART:SEARCH:SETUP:DATA:CSENSITIVE ON
:SERIALBUS1:UART:SEARCH:SETUP:DATA:CSENSITIVE?
-> :SERIALBUS1:UART:SEARCH:SETUP:DATA:CSENSITIVE 1

:SERIALbus<x>:UART:SEARCH:SETup:DATA:DBYTE

機能 UART 信号検索のデータのバイト数を設定 / 問い合わせします。

構文 :SERIALbus<x>:UART:SEARCH:SETup:DATA:DBYTE {<Nrf>}<x> = 1 ~ 4<Nrf> = 1 ~ 4(byte)

例 :SERIALBUS1:UART:SEARCH:SETUP:DATA:DBYTE 1
:SERIALBUS1:UART:SEARCH:SETUP:DATA:DBYTE?
-> :SERIALBUS1:UART:SEARCH:SETUP:DATA:DBYTE 1

:SERIALbus<x>:UART:SEARCH:SETup:DATA:DPATtern?

機能 UART バス信号検索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 :SERIALbus<x>:UART:SEARCH:SETup:
DATA:DPATtern?
<x> = 1 ~ 4

:SERIALbus<x>:UART:SEARCH:SETup:DATA:DPATtern:HEXa

機能 UART バス信号検索のデータを 16 進数で設定します。

構文 :SERIALbus<x>:UART:SEARCH:SETup:
DATA:DPATtern:HEXa {<文字列>}
<x> = 1 ~ 4

例 :SERIALBUS1:UART:SEARCH:SETUP:DATA:
DPATtern:HEXa "12"

:SERIALbus<x>:UART:SEARCH:SETup:DATA:DPATtern:PATtern

機能 UART 信号検索のデータを 2 進数で設定 / 問い合わせします。

構文 :SERIALbus<x>:UART:SEARCH:SETup:
DATA:DPATtern:PATtern {<文字列>}
:SERIALbus<x>:UART:SEARCH:SETup:
DATA:DPATtern:PATtern?
<x> = 1 ~ 4

例 :SERIALBUS1:UART:SEARCH:SETUP:DATA:
DPATtern:PATtern "00110101"
:SERIALBUS1:UART:SEARCH:SETUP:DATA:
DPATtern:PATtern?
-> :SERIALBUS1:UART:SEARCH:SETUP:
DATA:DPATtern:PATtern "00110101"

:SERIALbus<x>:UART:SEARCH:SETup:DATA:PFORMAT

機能 UART 信号検索の ID とデータ条件のデータの入力形式を設定 / 問い合わせします。

構文 :SERIALbus<x>:UART:SEARCH:SETup:
DATA:PFORMAT {ASCII|BINary|HEXa}
:SERIALbus<x>:UART:SEARCH:SETup:
DATA:PFORMAT?
<x> = 1 ~ 4

例 :SERIALBUS1:UART:SEARCH:SETUP:DATA:
PFORMAT BINARY
:SERIALBUS1:UART:SEARCH:SETUP:DATA:
PFORMAT?
-> :SERIALBUS1:UART:SEARCH:SETUP:
DATA:PFORMAT BINARY

:SERIALbus<x>:UART:SEARCH:SETup:ERROR?

機能 UART 信号検索の Error に関するすべての設定値を問い合わせます。

構文 :SERIALbus<x>:UART:SEARCH:SETup:
ERROR?
<x> = 1 ~ 4

:SERIALbus<x>:UART:SEARCH:SETup:ERROR:FRAMing

機能 UART 信号検索の Framing Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:UART:SEARCH:SETup:
ERROR:FRAMing {<Boolean>}
:SERIALbus<x>:UART:SEARCH:SETup:
ERROR:FRAMing?
<x> = 1 ~ 4

例 :SERIALBUS1:UART:SEARCH:SETUP:ERROR:
FRAMing ON
:SERIALBUS1:UART:SEARCH:SETUP:ERROR:
FRAMing?
-> :SERIALBUS1:UART:SEARCH:SETUP:
ERROR:FRAMing 1

:SERIALbus<x>:UART:SEARCH:SETup:ERROR:PARity

機能 UART 信号検索の Parity Error を設定 / 問い合わせします。

構文 :SERIALbus<x>:UART:SEARCH:SETup:
ERROR:PARity {<Boolean>}
:SERIALbus<x>:UART:SEARCH:SETup:
ERROR:PARity?
<x> = 1 ~ 4

例 :SERIALBUS1:UART:SEARCH:SETUP:ERROR:
PARity ON
:SERIALBUS1:UART:SEARCH:SETUP:ERROR:
PARity?
-> :SERIALBUS1:UART:SEARCH:SETUP:
ERROR:PARity 1

5.25 SERIALbus グループ

:SERIALbus<x>:UART:SEARCh:SETup:ERRor:PMODE

機能 UART 信号検索の Parity モードを設定 / 問い合わせします。

構文 :SERIALbus<x>:UART:SEARCh:SETup:
ERRor:PMODE {EVEN|ODD}
:SERIALbus<x>:UART:SEARCh:SETup:
ERRor:PMODE?
<x> = 1 ~ 4

例 :SERIALBUS1:UART:SEARCH:SETUP:ERROR:
PMODE EVEN
:SERIALBUS1:UART:SEARCH:SETUP:ERROR:
PMODE?
-> :SERIALBUS1:UART:SEARCH:SETUP:
ERROR:PMODE EVEN

:SERIALbus<x>:UART:SEARCh:SETup:FORMat

機能 UART 信号検索のフォーマットを設定 / 問い合わせします。

構文 :SERIALbus<x>:UART:SEARCh:SETup:
FORMat {BIT7Parity|BIT8Parity|
BIT8Noparity}
:SERIALbus<x>:UART:SEARCh:SETup:
FORMat?
<x> = 1 ~ 4

例 :SERIALBUS1:UART:SEARCH:SETUP:
FORMAT BIT7PARITY
:SERIALBUS1:UART:SEARCH:SETUP:
FORMAT?
-> :SERIALBUS1:UART:SEARCH:SETUP:
FORMAT BIT7PARITY

:SERIALbus<x>:UART:SEARCh:SETup:MODE

機能 UART 信号検索のモードを設定 / 問い合わせします。

構文 :SERIALbus<x>:UART:SEARCh:SETup:
MODE {DATA|EDATA|ERRor}
:SERIALbus<x>:UART:SEARCh:SETup:
MODE?
<x> = 1 ~ 4

例 :SERIALBUS1:UART:SEARCH:SETUP:
MODE DATA
:SERIALBUS1:UART:SEARCH:SETUP:MODE?
-> :SERIALBUS1:UART:SEARCH:SETUP:
MODE DATA

:SERIALbus<x>:ZLINKage

機能 シリアルバス信号解析の解析結果の解析番号とズーム位置との連動 (ON/OFF) を設定 / 問い合わせします。

構文 :SERIALbus<x>:ZLINKage {<Boolean>}
:SERIALbus<x>:ZLINKage?
<x> = 1 ~ 4

例 :SERIALBUS1:ZLINKAGE ON
:SERIALBUS1:ZLINKAGE?
-> :SERIALBUS1:ZLINKAGE 1

5.26 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

5.27 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 ~ 36000(100ms 単位: 待ち時間、START して待つ)
<NRf> = 0(START するだけ。待ちなし)
<NRf> = -36000 ~ -1(100ms 単位: 待ち時間、START しないで待つ)

例 :SSTART?

-> :SSTART 1

解説

- ・ 指定時間が + 値の場合は、指定時間内において、SINGLETRIGGER 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。

5.28 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。

5.29 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?

:STATus:CONDition?

機能 状態レジスタの内容を問い合わせます。

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 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: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 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: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"

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:FILTer<x>

機能 遷移フィルターを設定 / 問い合わせします。

構文 :STATus:FILTer<x> {BOTH|FALL|NEVer|RISE}

:STATus:FILTer<x>?

<x> = 1 ~ 16

例 :STATUS:FILTER2 RISE

:STATUS:FILTER2?

-> :STATUS:FILTER2 RISE

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: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 シリアルポーリングを実行します。

構文 :STATus:SPOLl?

例 :STATUS:SPOLL?

-> :STATUS:SPOLL 0

5.30 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。

5.31 STORe グループ

:STORe?

機能 内部メモリへ保存された設定データに関するすべての情報を問い合わせます。

構文 :STORe?

:STORe:SETup<x>?

機能 指定された番号の内部メモリに保存された設定データに関する情報を問い合わせます。

構文 :STORe:SETup<x>?

<x> = 1 ~ 3

:STORe:SETup<x>:COMMeNt

機能 指定された番号の内部メモリに保存する設定データのコメントを設定 / 問い合わせします。

構文 :STORe:SETup<x>:COMMeNt {<文字列>}

:STORe:SETup<x>:COMMeNt?

<x> = 1 ~ 3

<文字列> = 16 文字以内

例 :STORE:SETUP1:COMMENT "WAVE1"

:STORE:SETUP1:COMMENT?

-> :STORE:SETUP1:COMMENT "WAVE1"

:STORe:SETup<x>:DATE?

機能 指定された番号の内部メモリに保存した設定データの日付 / 時間を問い合わせます。

構文 :STORe:SETup<x>:DATE?

<x> = 1 ~ 3

例 :STORE:SETUP1:DATE?

-> :STORE:SETUP1:

DATE "2008/09/30 10:56:22"

:STORe:SETup<x>:EXECute

機能 指定された番号の内部メモリに設定データを保存します。

構文 :STORe:SETup<x>:EXECute

<x> = 1 ~ 3

例 :STORE:SETUP1:EXECUTE

:STORe:SETup<x>:LOCK

機能 指定された番号の内部メモリに保存した設定データの保護の ON/OFF を設定 / 問い合わせします。

構文 :STORe:SETup<x>:LOCK {<Boolean>}

:STORe:SETup<x>:LOCK?

<x> = 1 ~ 3

例 :STORE:SETUP1:LOCK ON

:STORE:SETUP1:LOCK?

-> :STORE:SETUP1:LOCK 1

5.32 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?

:SYSTem:BEEP

機能 ビープ音を鳴らします。

構文 :SYSTem:BEEP

例 :SYSTEM:BEEP

: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>}

:SYSTem:CLICk?

例 :SYSTEM:CLICk ON

:SYSTEM:CLICk?

-> :SYSTEM:CLICk 1

:SYSTem:CLOCK?

機能 日付・時刻に関するすべての設定値を問い合わせます。

構文 :SYSTem:CLOCK?

:SYSTem:CLOCK:DATE

機能 日付を設定 / 問い合わせします。

構文 :SYSTem:CLOCK:DATE {<文字列>}

:SYSTem:CLOCK:DATE?

<文字列> = YYYY/MM/DD、ユーザズマニュアル機能編参照。

例 :SYSTEM:CLOCK:DATE "2023/03/30"

:SYSTEM:CLOCK:DATE?

-> :SYSTEM:CLOCK:DATE "2023/03/30"

:SYSTem:CLOCK:FORMat

機能 日付の表示順を設定 / 問い合わせします。

構文 :SYSTem:CLOCK:FORMat {<Nrf>}

:SYSTem:CLOCK:FORMat?

<Nrf> = 1 ~ 4

例 :SYSTEM:CLOCK:FORMat 1

:SYSTEM:CLOCK:FORMat?

-> :SYSTEM:CLOCK:FORMat 1

:SYSTem:CLOCK:MODE

機能 日付・時刻表示の 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:CLOCK:SNTP?

機能 SNTP による日付時刻設定を問い合わせます。

構文 :SYSTem:CLOCK:SNTP?

:SYSTem:CLOCK:SNTP:EXECute

機能 SNTP による日付時刻を設定します。

構文 :SYSTem:CLOCK:SNTP:EXECute

例 :SYSTEM:CLOCK:SNTP:EXECUTE

:SYSTem:CLOCK:SNTP:GMTTime

機能 グリニッジ標準時との時差を設定 / 問い合わせします。

構文 :SYSTem:CLOCK:SNTP:GMTTime {<文字列>}

:SYSTem:CLOCK:SNTP:GMTTime?

<文字列> = HH:MM、ユーザズマニュアル機能編参照。

例 :SYSTEM:CLOCK:SNTP:GMTTIME "09:00"

:SYSTEM:CLOCK:SNTP:GMTTIME?

-> :SYSTEM:CLOCK:SNTP:GMTTIME "09:00"

:SYSTem:CLOCK:TIME

機能 時刻を設定 / 問い合わせします。

構文 :SYSTem:CLOCK:TIME {<文字列>}

:SYSTem:CLOCK:TIME?

<文字列> = HH:MM:SS、ユーザズマニュアル機能編参照。

例 :SYSTEM:CLOCK:TIME "14:30:00"

:SYSTEM:CLOCK:TIME?

-> :SYSTEM:CLOCK:TIME "14:30:00"

:SYSTem:DCANcel (Delay Cancel)

機能 設定した遅延時間を時間測定値に反映する (ON)/ しない (OFF) を設定 / 問い合わせします。

構文 :SYSTem:DCANcel {<Boolean>}

:SYSTem:DCANcel?

例 :SYSTEM:DCANCEL ON

:SYSTEM:DCANCEL?

-> :SYSTEM:DCANCEL 1

5.32 SYSTem グループ

:SYSTem:DLMSync?

機能 同期運転に関するすべての設定値を問い合わせます。

構文 :SYSTem:DLMSync?

:SYSTem:DLMSync:ADESkew

機能 同期運転時のオートスキュー補正を実行します。

構文 :SYSTem:DLMSync:ADESkew

例 :SYSTEM:DLMSYNC:ADESKEW

:SYSTem:DLMSync:CONNect?

機能 同期運転を開始し、結果を問い合わせます。

構文 :SYSTem:DLMSync:CONNect?

例 :SYSTEM:DLMSYNC:CONNECT?

-> :SYSTEM:DLMSYNC:CONNECT 0

解説 同期運転が成功した場合は「0」、失敗した場合は「1」を返します。

:SYSTem:DLMSync:DESKew

機能 同期運転時のスキュー補正を設定 / 問い合わせします。

構文 :SYSTem:DLMSync:DESKew {<時間>}

:SYSTem:DLMSync:DESKew?

<時間> = 15 ~ 35ns、0.05ns ステップ

例 :SYSTEM:DLMSYNC:DESKEW 20.2E-09

:SYSTEM:DLMSYNC:DESKEW?

-> :SYSTEM:DLMSYNC:DESKEW 20.2000E-09

:SYSTem:DLMSync:DISConnect?

機能 同期運転を終了し、結果を問い合わせます。

構文 :SYSTem:DLMSync:DISConnect?

例 :SYSTEM:DLMSYNC:DISCONNECT?

-> :SYSTEM:DLMSYNC:DISCONNECT 0

解説 成功した場合は「0」、失敗した場合は「1」を返します。

: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|FRENch|GERMan|ITALian|JAPANese|KOREan|SPANish}

:SYSTem:LANGuage?

例 :SYSTEM:LANGUAGE JAPANESE

:SYSTEM:LANGUAGE?

-> :SYSTEM:LANGUAGE JAPANESE

:SYSTem:LCD?

機能 液晶画面に関するすべての設定値を問い合わせます。

構文 :SYSTem:LCD?

:SYSTem:LCD:AUTO?

機能 液晶画面のバックライトのオートオフに関するすべての設定値を問い合わせます。

構文 :SYSTem:LCD:AUTO?

:SYSTem:LCD:AUTO:MODE

機能 液晶画面のバックライトのオートオフの ON/OFF を設定 / 問い合わせします。

構文 :SYSTem:LCD:AUTO:MODE {<Boolean>}

:SYSTem:LCD:AUTO:MODE?

例 :SYSTEM:LCD:AUTO:MODE ON

:SYSTEM:LCD:AUTO:MODE?

-> :SYSTEM:LCD:AUTO:MODE 1

:SYSTem:LCD:AUTO:TIME

機能 液晶画面のバックライトのオートオフまでの時間を設定 / 問い合わせします。

構文 :SYSTem:LCD:AUTO:TIME {<NRf>}

:SYSTem:LCD:AUTO:TIME?

<NRf> = 1 ~ 60(分)

例 :SYSTEM:LCD:AUTO:TIME 1

:SYSTEM:LCD:AUTO:TIME?

-> :SYSTEM:LCD:AUTO:TIME 1

:SYSTem:LCD:BRIGhtness

機能 液晶画面の明るさを設定 / 問い合わせします。

構文 :SYSTem:LCD:BRIGhtness {<NRf>}

:SYSTem:LCD:BRIGhtness?

<NRf> = 1 ~ 8

例 :SYSTEM:LCD:BRIGHTNESS 2

:SYSTEM:LCD:BRIGHTNESS?

-> :SYSTEM:LCD:BRIGHTNESS 2

:SYSTem:LCD:MODE

機能 液晶画面のバックライトの消灯 (OFF) / 点灯 (ON) を設定 / 問い合わせします。

構文 :SYSTem:LCD:MODE {<Boolean>}

:SYSTem:LCD:MODE?

例 :SYSTEM:LCD:MODE ON

:SYSTEM:LCD:MODE?

-> :SYSTEM:LCD:MODE 1

: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|FRENch|GERMan|ITALian|JAPANese|KOREan|SPANish}

:SYSTem:MLAnguage?

例 :SYSTEM:MLANGUAGE JAPANESE

:SYSTEM:MLANGUAGE?

-> :SYSTEM:MLANGUAGE JAPANESE

:SYSTem:MPARameter?

機能 波形パラメータの自動測定値・カーソル測定値に関するすべての設定値を問い合わせます。

構文 :SYSTem:MPARameter?

:SYSTem:MPARameter:FSIZE

機能 波形パラメータの自動測定値・カーソル測定値の表示文字の大きさを設定 / 問い合わせします。

構文 :SYSTem:MPARameter:FSIZE {LARGE|SMALL}

例 :SYSTem:MPARameter:FSIZE?

:SYSTEM:MPARAMETER:FSIZE SMALL

:SYSTEM:MPARAMETER:FSIZE?

-> :SYSTEM:MPARAMETER:FSIZE SMALL

:SYSTem:MPARameter:MROws

機能 波形パラメータの自動測定値・カーソル測定値をメニュー表示エリアから何行表示するかを設定 / 問い合わせします。

構文 :SYSTem:MPARameter:MROws {<NRf>}
:SYSTem:MPARameter:MROws?

<NRf> = 2 ~ 30

例 :SYSTEM:MPARAMETER:MROWS 4

:SYSTEM:MPARAMETER:MROWS?

-> :SYSTEM:MPARAMETER:MROWS 4

:SYSTem:OCANcel (Offset Cancel)

機能 設定したオフセット電圧を測定結果や演算結果に反映する (ON)/ しない (OFF) を設定 / 問い合わせします。

構文 :SYSTem:OCANcel {<Boolean>}
:SYSTem:OCANcel?

例 :SYSTEM:OCANCEL ON

:SYSTEM:OCANCEL?

-> :SYSTEM:OCANCEL 1

:SYSTem:POWeroFF:EXECute

機能 本機器の電源を切ります。また、次に主電源スイッチを ON したときの動作を設定します。

構文 :SYSTem:POWeroFF:EXECute [<NRf>]
[,<Boolean>]]

<NRf> = 1 ~ 60

例 :SYSTEM:POWEROFF:EXECUTE

解説

- <NRf> には、本機器の電源を切るまでの時間を秒で設定します。<NRf> を省略すると、5 秒後に電源を切ります。ここで、電源を切るとは、フロントパネルの電源スイッチをオフにすることを示します。

コントローラと通信したまま、本機器の電源を切ると、コントローラ上で通信エラーが発生する場合があります。通信エラーを避けるため、次の手順での使用をおすすめします。

1. コントローラの通信を終了するために必要な時間を入力して、本コマンドを送信する。
2. 入力した時間内に、コントローラと本機器の間の通信を終了する。

- <Boolean> には、次に主電源スイッチを ON したときの動作を設定します。<Boolean> が ON の場合は、主電源スイッチを一度 OFF し、その後 ON したときに、本機器の電源が入ります。<Boolean> が OFF の場合は、主電源スイッチを一度 OFF し、その後 ON しても、本機器の電源は切れたままです。<Boolean> を省略すると、OFF に設定した場合と同じ動作になります。

:SYSTem:PPOWer:STATus?

機能 プローブ電源の状態を問い合わせます。

構文 :SYSTem:PPOWer:STATus?

例 :SYSTEM:PPOWER:STATUS?

-> :SYSTEM:PPOWER:STATUS NORMAL

解説 応答の内容は、次のとおりです。

NORMAL: プローブ電源は正常です。

WARNING: プローブ電源が過負荷です。

FAILURE: プローブへの電源供給を停止しています。

:SYSTem:TOUCh

機能 タッチパネル機能の ON/OFF を設定 / 問い合わせします。

構文 :SYSTem:TOUCh {<Boolean>}
:SYSTem:TOUCh?

例 :SYSTEM:TOUCH ON

:SYSTEM:TOUCH?

-> :SYSTEM:TOUCH 1

5.32 SYSTem グループ

:SYSTem:TOUT?

機能 トリガ出力に関するすべての設定値を問い合わせます。

構文 :SYSTem:TOUT?

:SYSTem:TOUT:POLarity

機能 トリガ出力の極性を設定 / 問い合わせします。

構文 :SYSTem:TOUT:POLarity {NEGative|POSitive}
:SYSTem:TOUT:POLarity?

例 :SYSTEM:TOUT:POLARITY NEGATIVE
:SYSTEM:TOUT:POLARITY?
-> :SYSTEM:TOUT:POLARITY NEGATIVE

:SYSTem:TOUT:RTIMing

機能 トリガ出力をリセットするタイミングを設定 / 問い合わせします。

構文 :SYSTem:TOUT:RTIMing {AEND|NASTart}
:SYSTem:TOUT:RTIMing?

例 :SYSTEM:TOUT:RTIMING AEND
:SYSTem:TOUT:RTIMING?

-> :SYSTEM:TOUT:RTIMING AEND

解説

- ・「AEND」を指定すると、波形の取り込み終了時に出力をリセットします。
- ・「NASTart」を指定すると、次の波形の取り込み開始時に出力をリセットします。

:SYSTem:TRANsparency?

機能 メニューの透過モードに関するすべての設定値を問い合わせます。

構文 :SYSTem:TRANsparency?

:SYSTem:TRANsparency:LEVel

機能 メニューの透過モードの透過率を設定 / 問い合わせします。

構文 :SYSTem:TRANsparency:LEVel {<NRf>}
:SYSTem:TRANsparency:LEVel?
<NRf> = 1 ~ 5

例 :SYSTEM:TRANSPARENCY:LEVEL 3
:SYSTEM:TRANSPARENCY:LEVEL?
-> :SYSTEM:TRANSPARENCY:LEVEL 3

:SYSTem:TRANsparency:MODE

機能 透過モードの有効 / 無効を設定 / 問い合わせします。

構文 :SYSTem:TRANsparency:MODE {<Boolean>}
:SYSTem:TRANsparency:MODE?

例 :SYSTEM:TRANSPARENCY:MODE ON
:SYSTEM:TRANSPARENCY:MODE?
-> :SYSTEM:TRANSPARENCY:MODE 1

:SYSTem:TSYNchro?

機能 時刻同期に関するすべての設定値を問い合わせます。

構文 :SYSTem:TSYNchro?

:SYSTem:TSYNchro:IEEE1588?

機能 IEEE 1588 時刻同期に関するすべての設定値を問い合わせます。

構文 :SYSTem:TSYNchro:IEEE1588?

:SYSTem:TSYNchro:IEEE1588:DNUMber

機能 IEEE 1588 時刻同期のドメイン番号を設定 / 問い合わせします。

構文 :SYSTem:TSYNchro:IEEE1588:DNUMber {<NRf>}
:SYSTem:TSYNchro:IEEE1588:DNUMber?
<NRf> = 0 ~ 255

例 :SYSTEM:TSYNCHRO:IEEE1588:DNUMBER 0
:SYSTEM:TSYNCHRO:IEEE1588:DNUMBER?
-> :SYSTEM:TSYNCHRO:IEEE1588:DNUMBER 0

:SYSTem:TSYNchro:IEEE1588:LSECond

機能 IEEE 1588 時刻同期のうるう秒を設定 / 問い合わせします。

構文 :SYSTem:TSYNchro:IEEE1588:LSECond {<NRf>}
:SYSTem:TSYNchro:IEEE1588:LSECond?
<NRf> = 0 ~ 255

例 :SYSTEM:TSYNCHRO:IEEE1588:LSECOND 0
:SYSTEM:TSYNCHRO:IEEE1588:LSECOND?
-> :SYSTEM:TSYNCHRO:IEEE1588:LSECOND 0

:SYSTem:TSYNchro:IEEE1588:NLAYer

機能 IEEE 1588 時刻同期のネットワークレイヤーを設定 / 問い合わせします。

構文 :SYSTem:TSYNchro:IEEE1588:NLAYer {LAYER2|LAYER3}
:SYSTem:TSYNchro:IEEE1588:NLAYer?

例 :SYSTEM:TSYNCHRO:IEEE1588:NLAYER LAYER2
:SYSTEM:TSYNCHRO:IEEE1588:NLAYER?
-> :SYSTEM:TSYNCHRO:IEEE1588:NLAYER LAYER2

:SYSTem:TSYNchro:IEEE1588:PRIority**<x>**

機能 IEEE 1588 時刻同期の Priority を設定 / 問い合わせします。

構文 :SYSTem:TSYNchro:IEEE1588:

PRIority<x> {<Nrf>}

:SYSTem:TSYNchro:IEEE1588:

PRIority<x>?

<x> = 1、2

<Nrf> = 0 ~ 255

例 :SYSTEM:TSYNCHRO:IEEE1588:

PRIORITY1 255

:SYSTEM:TSYNCHRO:IEEE1588:PRIORITY1?

-> :SYSTEM:TSYNCHRO:IEEE1588:

PRIORITY1 255

:SYSTem:TSYNchro:IEEE1588:PState?

機能 IEEE 1588 時刻同期の状態を問い合わせます。

構文 :SYSTem:TSYNchro:IEEE1588:PState?

例 :SYSTEM:TSYNCHRO:IEEE1588:PSTATE?

-> :SYSTEM:TSYNCHRO:IEEE1588:

PSTATE LISTENING

解説 応答の内容は、次のとおりです。

OFF : 時刻同期機能が OFF

INITIALIZING、FAULTY、DISABLED、
LISTENING、PMASER、MASTER、PASSIVE、
UNCALIBRATED、SLAVE : IEEE 1588-2008 で定
義されたステート

:SYSTem:TSYNchro:IEEE1588:SONLy

機能 IEEE 1588 時刻同期の Slave Only を設定 / 問い合わせします。

構文 :SYSTem:TSYNchro:IEEE1588:

SONLy {<Boolean>}

:SYSTem:TSYNchro:IEEE1588:SONLy?

例 :SYSTEM:TSYNCHRO:IEEE1588:ONLY ON

:SYSTEM:TSYNCHRO:IEEE1588:STATE?

-> :SYSTEM:TSYNCHRO:IEEE1588:ONLY 1

:SYSTem:TSYNchro:IEEE1588:StAtE

機能 IEEE 1588 時刻同期の ON/OFF を設定 / 問い合わせします。

構文 :SYSTem:TSYNchro:IEEE1588:

StAtE {<Boolean>}

:SYSTem:TSYNchro:IEEE1588:StAtE?

例 :SYSTEM:TSYNCHRO:IEEE1588:STATE ON

:SYSTEM:TSYNCHRO:IEEE1588:STATE?

-> :SYSTEM:TSYNCHRO:IEEE1588:STATE 1

:SYSTem:TSYNchro:StAtE?

機能 本機器の時刻同期の状態を問い合わせます。

構文 :SYSTem:TSYNchro:StAtE?

例 :SYSTEM:TSYNCHRO:STATE?

-> :SYSTEM:TSYNCHRO:STATE OFF

解説 応答の内容は、次のとおりです。

OFF : 時刻同期機能が OFF

UNLock : マスターと通信ができていません。サン
プリングクロックは内部発振器を基
準として自走しています。

LOCK : マスターと同期を開始し、時刻とサン
プリングクロックを調整してる状態です。

StABle : マスターと同期が成立し、時刻とサン
プリングクロックが安定した状態
です。

: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 JAPANESE

:SYSTEM:USBKEYBOARD?

-> :SYSTEM:USBKEYBOARD JAPANESE

5.33 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?

:TImebase:SRAt? (Sample RATE)

機能 サンプルレートを問い合わせます。

構文 :TImebase:SRAt?

例 :TIMEBASE:SRATE?

-> :TIMEBASE:SRATE 12.50E+06

:TImebase:TDIV

機能 Time/div 値を設定 / 問い合わせします。

構文 :TImebase:TDIV {< 時間 >}

:TImebase:TDIV?

< 時間 > = 500ps ~ 500s

例 :TIMEBASE:TDIV 2NS

:TIMEBASE:TDIV?

-> :TIMEBASE:TDIV 2.000E-06

5.34 TRIGger グループ

トリガの種類に関する通信コマンドと、操作キーを押して表示されるトリガタイプの対応を以下に示します。

トリガの種類	通信コマンド	設定メニュー	
		操作キー	トリガタイプ
エッジ	SIMPlE	EDGE キー	—
複数のエッジトリガの OR(エッジ OR)	OR	ENHANCED キー	Edge OR
複数入力の組み合わせ (パターン)	PATtern		Pattern
パルス幅	PULSe		Pulse Width
立ち上がり / 立ち下がり時間	RFTime		Rise/Fall Time
ラント	RUNT		Runt
タイムアウト	TIMEout		Timeout
ウィンドウ	WINDow		Window
ウィンドウ OR	WOR		Window OR
エッジ間隔 (インターバル)	INTerval		Interval
Serial	CAN*		CAN
	CANFD*		CAN FD
	CXPI*		CXPI
	FLEXray*		FlexRay
	I2C*		I2C
	LIN*		LIN
	PSI5*		PSI5 Airbag
	SENT*		SENT
	SPATtern		User Define
	SPI*		SPI
	UART*		UART
	TV:HDTV		HDTV
	TV:NTSC		NTSC
	TV:PAL		PAL
	TV:SDTV		SDTV(480/60p)
	TV:USERdefine		UserDef TV
コンビネーション	COMBination	B TRIG キー	—
強制	FORCe	SHIFT+B TRIG(FORCE TRIG) キー	—

* CAN、CANFD、CXPI、FLEXray、I2C、LIN、PSI5、SENT、SPI、UART はシリアルバスオプションです。シリアルバスオプション付でないモデルでは、これらのシリアルバスに関するコマンドは使用できません。

: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?

:TRIGger:ABN?

機能 A->B(N) 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:ABN?

:TRIGger:ABN:COUNT

機能 A->B(N) トリガの条件 B の成立回数を設定 / 問い合わせします。

構文 :TRIGger:ABN:COUNT {<NRf>}

:TRIGger:ABN:COUNT?

<NRf> = 1 ~ 1000000000

例 :TRIGGER:ABN:COUNT 1

:TRIGGER:ABN:COUNT?

-> :TRIGGER:ABN:COUNT 1

: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?

: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 1

:TRIGGER:ACTION:ACQCOUNT?

-> :TRIGGER:ACTION:ACQCOUNT 1

5.34 TRIGger グループ

: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通知に関するすべての設定値を問い合わせます。

構文 :TRIGger:ACtion:MAIL?

:TRIGger:ACtion:MAIL:COUNT

機能 アクション時のメール通知のメール件数の上限を設定 / 問い合わせします。

構文 :TRIGger:ACtion:MAIL:COUNT {<Nrf>}
:TRIGger:ACtion:MAIL:COUNT?
<Nrf> = 1 ~ 1000

例 :TRIGGER:ACTION:MAIL:COUNT 1
:TRIGGER:ACTION:MAIL:COUNT?
-> :TRIGGER:ACTION:MAIL:COUNT 1

:TRIGger:ACtion:MAIL:MODE

機能 アクション時にメールで通知するかしないかを設定 / 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: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
:TRIGGER:ACTION:START

解説 トリガアクションを中断するには、
「:TRIGger:ACtion:STOP」を使います。

: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
:TRIGGER:ACTION:STOP

解説 トリガアクションを中止するだけで、設定のキャンセルはしません。
「:TRIGger:ACtion:START」でトリガアクションを再開します。

:TRIGger:ADB?

機能 A Delay B 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:ADB?

:TRIGger:ADB:DElay

機能 A Delay B トリガの条件 B のディレイ時間を設定 / 問い合わせします。

構文 :TRIGger:ADB:DElay {<時間>}
:TRIGger:ADB:DElay?
<時間> = 10ns ~ 10s、2ns ステップ

例 :TRIGGER:ADB:DELAY 10ns
:TRIGGER:ADB:DELAY?
-> :TRIGGER:ADB:DELAY 10ns

:TRIGger:{ATRIgger|BTRigger}?

機能 トリガ条件に関するすべての設定値を問い合わせます。

構文 :TRIGger:{ATRIgger|BTRigger}?

:TRIGger{[:ATRIgger]|:BTRigger}:CAN?

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{[:ATRIgger]|:BTRigger}:CAN?

:TRIGger{[:ATRIgger]|:BTRigger}:CAN: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{[:ATRIgger]|:BTRigger}:CAN:

BRATe {<Nrf>|USER,<Nrf>}

:TRIGger{[:ATRIgger]|:BTRigger}:CAN:

BRATe?

<Nrf> = 33300、83300、125000、250000、500000、1000000

USER の <Nrf> = 本体ユーザズマニュアル機能編参照。

例 :TRIGGER:ATRIGGER:CAN:BRATE 83300

:TRIGGER:ATRIGGER:CAN:BRATE?

-> :TRIGGER:ATRIGGER:CAN:BRATE 83300

:TRIGger{[:ATRIgger]|:BTRigger}:CAN:EFramE?

機能 CAN バス信号トリガの Error に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRIgger]|:BTRigger}:CAN:

EFramE?

:TRIGger{[:ATRIgger]|:BTRigger}:CAN:EFramE:CRC

機能 CAN バス信号トリガの CRC Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRIgger]|:BTRigger}:CAN:

EFramE:CRC {<Boolean>}

:TRIGger{[:ATRIgger]|:BTRigger}:CAN:

EFramE:CRC?

例 :TRIGGER:ATRIGGER:CAN:EFramE:CRC ON

:TRIGGER:ATRIGGER:CAN:EFramE:CRC?

-> :TRIGGER:ATRIGGER:CAN:EFramE:

CRC 1

:TRIGger{[:ATRIgger]|:BTRigger}:CAN:EFramE[:MODE]

機能 CAN バス信号トリガの Error Frame を設定 / 問い合わせします。

構文 :TRIGger{[:ATRIgger]|:BTRigger}:CAN:

EFramE[:MODE] {<Boolean>}

:TRIGger{[:ATRIgger]|:BTRigger}:CAN:

EFramE:MODE?

例 :TRIGGER:ATRIGGER:CAN:EFramE:MODE ON

:TRIGGER:ATRIGGER:CAN:EFramE:MODE?

-> :TRIGGER:ATRIGGER:CAN:EFramE:

MODE 1

:TRIGger{[:ATRIgger]|:BTRigger}:CAN:EFramE:STUFF

機能 CAN バス信号トリガの STUFF Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRIgger]|:BTRigger}:CAN:

EFramE:STUFF {<Boolean>}

:TRIGger{[:ATRIgger]|:BTRigger}:CAN:

EFramE:STUFF?

例 :TRIGGER:ATRIGGER:CAN:EFramE:

STUFF ON

:TRIGGER:ATRIGGER:CAN:EFramE:STUFF?

-> :TRIGGER:ATRIGGER:CAN:EFramE:

STUFF 1

:TRIGger{[:ATRIgger]|:BTRigger}:CAN:IDData?

機能 CAN バス信号トリガの ID とデータ条件に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRIgger]|:BTRigger}:CAN:

IDData?

:TRIGger{[:ATRIgger]|:BTRigger}:CAN[IDData]:ACK?

機能 CAN バス信号トリガの ID とデータ条件の ACK に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRIgger]|:BTRigger}:

CAN[:IDData]:ACK?

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:ACK:MODE

機能 CAN バス信号トリガの ID とデータ条件の ACK モードを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:ACK:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:ACK:MODE?

例 :TRIGGER:ATRIGGER:CAN:IDDATA:ACK:
MODE ON
:TRIGGER:ATRIGGER:CAN:IDDATA:ACK:
MODE?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:ACK:
MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:ACK:TYPE

機能 CAN バス信号トリガの ID とデータ条件の ACK 条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:ACK:TYPE {ACK|ACKBoth|
NONAck}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:ACK:TYPE?

例 :TRIGGER:ATRIGGER:CAN:IDDATA:ACK:
TYPE ACK
:TRIGGER:ATRIGGER:CAN:IDDATA:ACK:
TYPE?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:ACK:
TYPE ACK

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA?

機能 CAN バス信号トリガの ID とデータ条件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{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA?

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:CONDition

機能 CAN バス信号トリガの ID とデータ条件のデータ判定条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:
CONDition {BETween|EQUal|FALSe|
GREater|LESS|NOTBetween|NOTEqaul|
TRUE}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:CONDition?

例 :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
CONDITION BETWEEN
:TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
CONDITION?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
DATA:CONDITION BETWEEN

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:DECimal<x>

機能 CAN バス信号トリガの ID とデータ条件の判定データを 10 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:DECimal<x> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:DECimal<x>?
<x> = 1 ~ 2
<Nrf> = 本体ユーザズマニュアル機能編参照。

例 :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
DECIMAL1 1
:TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
DECIMAL1?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
DATA:DECIMAL1 1.000E+00

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:DLC

機能 CAN バス信号トリガの ID とデータ条件の有効バイト数 (DLC) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:DLC {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:DLC?
<Nrf> = 0 ~ 8

例 :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
DLC 0
:TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
DLC?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
DATA:DLC 0

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:DPATtern?

機能 CAN バス信号トリガの ID とデータ条件の判定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{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:DPATtern?

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:DPATtern:HEXa

機能 CAN バス信号トリガの ID とデータ条件の判定データを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:DPATtern:HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:DPATTERN:HEXa "12"

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:DPATtern:PATtern

機能 CAN バス信号トリガの ID とデータ条件のデータを 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:DPATtern:PATtern {<文字列>}

例 :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:DPATTERN:PATTERN "00110101"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:DLC」の設定値によって設定できるバイト数が変動します。

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:ENDian

機能 CAN バス信号トリガの ID とデータ条件の判定データのエンディアンを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:ENDian {BIG|LITTLE}

例 :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:ENDIAN BIG

解説 「:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:DLC」の設定値によって設定できるバイト数が変動します。

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:MODE

機能 CAN バス信号トリガの ID とデータ条件のデータ条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:MODE {<Boolean>}

例 :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:MODE ON

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:MSBLsb

機能 CAN バス信号トリガの ID とデータ条件のデータの MSB/LSB のビットを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:MSBLsb {<Nrf>, <Nrf>}

例 :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:MSBLSB 1,0

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:SIGN

機能 CAN バス信号トリガの ID とデータ条件のデータの符号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:SIGN {SIGN|UNSign}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:SIGN?

例 :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
SIGN SIGN
:TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
SIGN?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
DATA:SIGN SIGN

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:IDENTifier?

機能 CAN バス信号トリガの ID とデータ条件の識別子に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier?

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:IDENTifier:ID?

機能 CAN バス信号トリガの ID とデータ条件の ID に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:ID?

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:IDENTifier:ID:HEXa

機能 CAN バス信号トリガの ID とデータ条件の ID を 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:ID:
HEXa{<文字列>}

例 :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:ID:HEXa "1AB"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:IDENTifier:MFormat」の設定値によって設定できるバイト数が変動します。

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:IDENTifier:ID:MODE

機能 CAN バス信号トリガの ID とデータ条件の ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:ID:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:ID<x>:MODE?

例 :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:ID:MODE ON
:TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:ID:MODE?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:ID:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:IDENTifier:ID:PATtern

機能 CAN バス信号トリガの ID とデータ条件の ID 条件を 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:ID:
PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:ID:PATtern?

例 :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:ID:PATTERN "11100001111"
:TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:ID:PATTERN?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:ID:PATTERN "11100001111"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:IDENTifier:MFormat」の設定値によって設定できるバイト数が変動します。

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:IDENTifier:MFormat

機能 CAN バス信号トリガの ID とデータ条件の ID のフレームフォーマット (標準 / 拡張) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:
MFormat {EXTended|STANDARD}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:MFormat?

例 :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:MFORMAT EXTENDED
:TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:MFORMAT?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:MFORMAT EXTENDED

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:IDENtifier:PFORmat

機能 CAN バス信号トリガの ID とデータ条件の ID の入力形式を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:IDENtifier:PFORmat {MESSagE|PATtern}
:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:IDENtifier:PFORmat?

例 :TRIGGER:ATRIGGER:CAN:IDDATA:IDENTIFIER:PFORMAT PATTERN
:TRIGGER:ATRIGGER:CAN:IDDATA:IDENTIFIER:PFORMAT?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:IDENTIFIER:PFORMAT PATTERN

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal?

機能 CAN バス信号トリガの ID とデータ条件のメッセージング信号に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal?

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal:MESSagE:ITEM

機能 CAN バス信号トリガの ID とデータ条件のメッセージのアイテムを設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal:MESSagE:ITEM {<文字列>}
<文字列> = 32 文字以内

例 :TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:MESSAGE:ITEM "TEST"

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal:SELEct

機能 CAN バス信号トリガのメッセージング信号の条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal:SELEct {MESSagE|SIGnal}
:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal:SELEct?

例 :TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SELECT MESSAGE
:TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SELECT?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SELECT MESSAGE

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal:SIGnal?

機能 CAN バス信号トリガの ID とデータ条件の信号に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal:SIGnal?

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal:SIGnal:CONDition

機能 CAN バス信号トリガの ID とデータ条件の信号のデータ条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal:SIGnal:CONDition {BETween|EQual|GREater|LESS|NOTbetween|NOTEqual}
:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal:SIGnal:CONDition?

例 :TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SIGNAL:CONDITION BETWEEN
:TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SIGNAL:CONDITION?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SIGNAL:CONDITION BETWEEN

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal:SIGnal:DECimal<x>

機能 CAN バス信号トリガの ID とデータ条件の信号の判定データを 10 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal:SIGnal:DECimal<x>{<NRF>}
:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal:SIGnal:DECimal<x>?

<x> = 1 ~ 2
<NRF> = 本体ユーザズマニュアル機能編参照。

例 :TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SIGNAL:DECIMAL1 1
:TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SIGNAL:DECIMAL1?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SIGNAL:DECIMAL1 1.000E+00

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal:SIGnal:ITEM

機能 CAN バス信号トリガの ID とデータ条件のシグナルのアイテムを設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:MSIGnal:SIGnal:ITEM {<文字列>,<文字列>}
<文字列> = 32 文字以内

例 :TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SIGNAL:ITEM "Sig _ Test",
"Mess _ Test"

解説 設定する順番は、シグナルアイテム、メッセージアイテムの順です。

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:RTR

機能 CAN バス信号トリガの ID とデータ条件の RTR を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:RTR {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:RTR?

例 :TRIGGER:ATRIGGER:CAN:IDDATA:RTR ON
:TRIGGER:ATRIGGER:CAN:IDDATA:RTR?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:RTR 1

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR?

機能 CAN バス信号トリガの ID 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{[:ATRigger]|:BTRigger}:CAN:IDOR?

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK?

機能 CAN バス信号トリガの ID OR 条件の ACK に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK?

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK:MODE

機能 CAN バス信号トリガの ID OR 条件の ACK モードを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK:MODE?

例 :TRIGGER:ATRIGGER:CAN:IDOR:ACK:MODE ON
:TRIGGER:ATRIGGER:CAN:IDOR:ACK:MODE?
-> :TRIGGER:ATRIGGER:CAN:IDOR:ACK:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK:TYPE

機能 CAN バス信号トリガの ID OR 条件の ACK 条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK:TYPE {ACK|ACKBoth|NONack}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK:TYPE?

例 :TRIGGER:ATRIGGER:CAN:IDOR:ACK:TYPE ACK
:TRIGGER:ATRIGGER:CAN:IDOR:ACK:TYPE?
-> :TRIGGER:ATRIGGER:CAN:IDOR:ACK:TYPE ACK

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:DATA?

機能 CAN バス信号トリガの ID OR 条件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{[:ATRigger]|:BTRigger}:CAN:IDOR:DATA?

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:DATA[:MODE]

機能 CAN バス信号トリガの ID OR 条件のデータ条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:DATA[:MODE] {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:DATA[:MODE]?

例 :TRIGGER:ATRIGGER:CAN:IDOR:DATA:MODE ON
:TRIGGER:ATRIGGER:CAN:IDOR:DATA:MODE?
-> :TRIGGER:ATRIGGER:CAN:IDOR:DATA:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier?

機能 CAN バス信号トリガの ID OR 条件の識別子に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier?

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:ID<x>?

機能 CAN バス信号トリガの ID OR 条件の ID に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:ID<x>?
<x> = 1 ~ 4

**:TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:ID<x>:HEXa**

機能 CAN バス信号トリガの ID OR 条件の ID を 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:ID<x>:HEXa {<文字列>}
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:CAN:IDOR:
IDENTIFIER:ID1:HEXa "1AB"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:CAN:
:IDOR:IDENtifier:MFORMAT」の設定値によっ
て設定できるバイト数が増減します。

**:TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:ID<x>:MODE**

機能 CAN バス信号トリガの ID OR 条件の ID 条件 (有
効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:ID<x>:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:ID<x>:MODE?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:CAN:IDOR:
IDENTIFIER:ID1:MODE ON
:TRIGGER:ATRIGGER:CAN:IDOR:
IDENTIFIER:ID1:MODE?
-> :TRIGGER:ATRIGGER:CAN:IDOR:
IDENTIFIER:ID1:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:ID<x>:PATTERN**

機能 CAN バス信号トリガの ID OR 条件の ID 条件を 2
進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:ID<x>:
PATTERN {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:ID<x>:PATTERN?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:CAN:IDOR:
IDENTIFIER:ID1:PATTERN "11100001111"
:TRIGGER:ATRIGGER:CAN:IDOR:
IDENTIFIER:ID1:PATTERN?
-> :TRIGGER:ATRIGGER:CAN:IDOR:
IDENTIFIER:ID1:PATTERN "11100001111"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:CAN:
:IDOR:IDENtifier:MFORMAT」の設定値によっ
て設定できるバイト数が増減します。

**:TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:MFORMAT**

機能 CAN バス信号トリガの ID OR 条件の ID のフレ
ームフォーマット (標準 / 拡張) を設定 / 問い合わ
せします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:MFORMAT {EXTended|
STANDARD}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:MFORMAT?

例 :TRIGGER:ATRIGGER:CAN:IDOR:
IDENTIFIER:MFORMAT EXTENDED
:TRIGGER:ATRIGGER:CAN:IDOR:
IDENTIFIER:MFORMAT?
-> :TRIGGER:ATRIGGER:CAN:IDOR:
IDENTIFIER:MFORMAT EXTENDED

**:TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:MODE**

機能 CAN バス信号トリガの ID OR 条件の識別子条件
(有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:MODE {<Boolean>}

例 :TRIGGER:ATRIGGER:CAN:IDOR:
IDENTIFIER:MODE ON
-> :TRIGGER:ATRIGGER:CAN:IDOR:
IDENTIFIER:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:PFORMAT**

機能 CAN バス信号トリガの ID OR 条件の ID の入力形
式を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:PFORMAT {MESSAGE|
PATTERN}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:
IDOR:IDENtifier:PFORMAT?

例 :TRIGGER:ATRIGGER:CAN:IDOR:
IDENTIFIER:PFORMAT PATTERN
:TRIGGER:ATRIGGER:CAN:IDOR:
IDENTIFIER:PFORMAT?
-> :TRIGGER:ATRIGGER:CAN:IDOR:
IDENTIFIER:PFORMAT PATTERN

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:MSIGnal<x>:MESSAge:ITEM

機能 CAN バス信号トリガの ID OR 条件のメッセージのアイテムを設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:MSIGnal<x>:MESSAge:ITEM {<文字列>}

<x> = 1 ~ 4

<文字列> = 32 文字以内

例 :TRIGGER:ATRIGGER:CAN:IDOR:MSIGNAL1:MESSAGE:ITEM "TEST"

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:RTR

機能 CAN バス信号トリガの ID OR 条件の RTR を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:RTR {<Boolean>}

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:RTR?

例 :TRIGGER:ATRIGGER:CAN:IDOR:RTR ON
:TRIGGER:ATRIGGER:CAN:IDOR:RTR?
-> :TRIGGER:ATRIGGER:CAN:IDOR:RTR 1

:TRIGger{[:ATRigger]|:BTRigger}:CAN: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{[:ATRigger]|:BTRigger}:CAN:MODE {EFrame|IDData|IDOR|SOF}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:MODE?

例 :TRIGGER:ATRIGGER:CAN:MODE EFRAME
:TRIGGER:ATRIGGER:CAN:MODE?
-> :TRIGGER:ATRIGGER:CAN:MODE EFRAME

:TRIGger{[:ATRigger]|:BTRigger}:CAN:REcessive

機能 CAN バス信号トリガのリセッシブ電位を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:REcessive {HIGH|LOW}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:REcessive?

例 :TRIGGER:ATRIGGER:CAN:RECESSIVE HIGH
:TRIGGER:ATRIGGER:CAN:RECESSIVE?
-> :TRIGGER:ATRIGGER:CAN:RECESSIVE HIGH

:TRIGger{[:ATRigger]|:BTRigger}:CAN:SOURce

機能 CAN バス信号トリガの信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CAN:SOURce {<Nrf>}

:TRIGger{[:ATRigger]|:BTRigger}:CAN:SOURce?

<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :TRIGGER:ATRIGGER:CAN:SOURCE 1
:TRIGGER:ATRIGGER:CAN:SOURCE?
-> :TRIGGER:ATRIGGER:CAN:SOURCE 1

:TRIGger{[:ATRigger]|:BTRigger}:CAN: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{[:ATRigger]|:BTRigger}:CAN:SPOint {<Nrf>}

:TRIGger{[:ATRigger]|:BTRigger}:CAN:SPOint?

<Nrf> = 18.8 ~ 90.6

例 :TRIGGER:ATRIGGER:CAN:SPOINT 18.8
:TRIGGER:ATRIGGER:CAN:SPOINT?
-> :TRIGGER:ATRIGGER:CAN:SPOINT 18.8

:TRIGger{[:ATRigger]|:BTRigger}:CANFD?

機能 CAN FD 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{[:ATRigger]|:BTRigger}:CANFD?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:BRATe

機能 CAN FD 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{[:ATRigger]|:BTRigger}:CANFD:BRATe {<Nrf>|USER,<Nrf>}

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:BRATe?

<Nrf> = 250000、500000、1000000

USER の <Nrf> = 20000 ~ 1000000(設定分解能 100)

例 :TRIGGER:ATRIGGER:CANFD:BRATE 500000
:TRIGGER:ATRIGGER:CANFD:BRATE?
-> :TRIGGER:ATRIGGER:CANFD:BRATE 500000

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:DBRate

機能 CAN FD バス信号トリガのデータフェーズのビットレート（データ転送速度）を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CANFD:DBRate {<NRf>|USER,<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:CANFD:DBRate?
<NRf> = 500000、1000000、2000000、4000000、5000000、8000000
USER の <NRf> = 250000 ~ 10000000(設定分解能 100)

例 :TRIGGER:ATRIGGER:CANFD:
DBRATE 1000000
:TRIGGER:ATRIGGER:CANFD:DBRATE?
-> :TRIGGER:ATRIGGER:CANFD:
DBRATE 1000000

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:DSPoint

機能 CAN FD バス信号トリガのデータフェーズのサンプルポイントを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CANFD:DSPoint {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:CANFD:DSPoint?
<NRf> = 18.8 ~ 90.6(設定分解能 0.1)

例 :TRIGGER:ATRIGGER:CANFD:DSPPOINT 18.8
:TRIGGER:ATRIGGER:CANFD:DSPPOINT?
-> :TRIGGER:ATRIGGER:CANFD:
DSPPOINT 18.8

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе?

機能 CAN FD バス信号トリガの Error に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе:CRC

機能 CAN FD バス信号トリガの CRC Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе:CRC {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе:CRC?

例 :TRIGGER:ATRIGGER:CANFD:EFramе:
CRC ON
:TRIGGER:ATRIGGER:CANFD:EFramе:CRC?
-> :TRIGGER:ATRIGGER:CANFD:EFramе:
CRC 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе:CRCEFactor?

機能 CAN FD バス信号トリガの CRC エラー要因に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе:CRCEFactor?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе:CRCEFactor:CRCSequence

機能 CAN FD バス信号トリガの CRC エラー要因の CRC Sequence を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе:CRCEFactor:
CRCSequence {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе:CRCEFactor:CRCSequence?

例 :TRIGGER:ATRIGGER:CANFD:EFramе:
CRCEFACTOR:CRCSEQUENCE ON
:TRIGGER:ATRIGGER:CANFD:EFramе:
CRCEFACTOR:CRCSEQUENCE?
-> :TRIGGER:ATRIGGER:CANFD:EFramе:
CRCEFACTOR:CRCSEQUENCE 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе:CRCEFactor:SCount

機能 CAN FD バス信号トリガの CRC エラー要因の Stuff Count を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе:CRCEFactor:
SCount {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе:CRCEFactor:SCount?

例 :TRIGGER:ATRIGGER:CANFD:EFramе:
CRCEFACTOR:SCOUNT ON
:TRIGGER:ATRIGGER:CANFD:EFramе:
CRCEFACTOR:SCOUNT?
-> :TRIGGER:ATRIGGER:CANFD:EFramе:
CRCEFACTOR:SCOUNT 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе:FSTuff

機能 CAN FD バス信号トリガの固定 STUFF Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе:FSTuff {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFramе:FSTuff?

例 :TRIGGER:ATRIGGER:CANFD:EFramе:
FSTUFF ON
:TRIGGER:ATRIGGER:CANFD:EFramе:
FSTUFF?
-> :TRIGGER:ATRIGGER:CANFD:EFramе:
FSTUFF 1

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFrame[:MODE]

機能 CAN FD バス信号トリガの Error Frame を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:EFrame[:MODE] {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:Eframe:mode?

例 :TRIGGER:ATRIGGER:CANFD:EFRAME:
MODE ON
:TRIGGER:ATRIGGER:CANFD:EFRAME:MODE?
-> :TRIGGER:ATRIGGER:CANFD:EFRAME:
MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFrame:STUFF

機能 CAN FD バス信号トリガの STUFF Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:EFrame:STUFF {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:EFrame:STUFF?

例 :TRIGGER:ATRIGGER:CANFD:EFRAME:
STUFF ON
:TRIGGER:ATRIGGER:CANFD:EFRAME:
STUFF?
-> :TRIGGER:ATRIGGER:CANFD:EFRAME:
STUFF 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:FDF?

機能 CAN FD バス信号トリガの FDF に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:FDF?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:FDF:CONDition

機能 CAN FD バス信号トリガの FDF を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:FDF:CONDition {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:FDF:CONDition?

例 :TRIGGER:ATRIGGER:CANFD:EFRAME:FDF:
CONDition ON
:TRIGGER:ATRIGGER:CANFD:EFRAME:FDF:
CONDition?
-> :TRIGGER:ATRIGGER:CANFD:EFRAME:
FDF:CONDition 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:FDStandard

機能 CAN FD バス信号トリガが、ISO 標準か否かを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:FDStandard {ISO|NISO}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:FDStandard?

例 :TRIGger:ATRIGGER:CANFD:
FDSTANDARD ISO
:TRIGger:ATRIGGER:CANFD:FDSTANDARD?
-> :TRIGger:ATRIGGER:CANFD:
FDSTANDARD ISO

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDData?

機能 CAN FD バス信号トリガの ID/Data 条件に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDData?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[IDData]:ACK?

機能 CAN FD バス信号トリガの ID/Data 条件の ACK に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[IDData]:ACK?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[IDData]:ACK:MODE

機能 CAN FD バス信号トリガの ID/Data 条件の ACK モードを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[IDData]:ACK:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[IDData]:ACK:MODE?

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:ACK:
MODE ON
:TRIGGER:ATRIGGER:CANFD:IDDATA:ACK:
MODE?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
ACK:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:ACK:TYPE

機能 CAN FD バス信号トリガの ID/Data 条件の ACK 条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:ACK:TYPE {ACK|
ACKBoth|NONack}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:ACK:TYPE?

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:ACK:
TYPE ACK
:TRIGGER:ATRIGGER:CANFD:IDDATA:ACK:
TYPE?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
ACK:TYPE ACK

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA?

機能 CAN FD バス信号トリガの ID/Data 条件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{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:BCOUNT

機能 CAN FD バス信号トリガの ID/Data 条件のデータ判定条件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{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:BCOUNT {<Nrf>}
<Nrf> = 0 ~ 63

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
BCOUNT 4
:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
BCOUNT?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
DATA:BCOUNT 4

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:CONDITION

機能 CAN FD バス信号トリガの ID/Data 条件のデータ判定条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:
CONDition {BETween|EQual|FALSe|
GREater|LESS|NOTBetween|NOTEqual|
TRUE}

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
CONDITION BETWEEN
:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
CONDITION?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
DATA:CONDITION BETWEEN

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:DBYTE

機能 CAN FD バス信号トリガの ID/Data 条件のデータのバイト数を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:DBYTE {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:DBYTE?

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
DBYTE 2
:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
DBYTE?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
DATA:DBYTE 2

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:DECimal<x>

機能 CAN FD バス信号トリガの ID/Data 条件の判定データ (a,b) を 10 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:
DECimal<x> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:DECimal<x>?
<x> = 1 ~ 2
<Nrf> = -9E+18 ~ 9E+18

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
DECIMAL1 1
:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
DECIMAL1?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
DATA:DECIMAL1 1.000E+00

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:DPATtern?

機能 CAN FD バス信号トリガの ID/Data 条件の判定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{[:ATRigger]|:BTRigger):CANFD[:IDData]:DATA:DPATtern?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:DPATtern:HEXa

機能 CAN FD バス信号トリガの ID/Data 条件の判定データを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger):CANFD[:IDData]:DATA:DPATtern:HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:DPATTERN:HEXa "12"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:DBYTe」の設定値によって設定できるバイト数が変動します。

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:DPATtern:PATtern

機能 CAN FD バス信号トリガの ID/Data 条件のデータを 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):CANFD[:IDData]:DATA:DPATtern:PATtern {<文字列>}:TRIGger{[:ATRigger]|:BTRigger):CANFD[:IDData]:DATA:DPATtern:PATtern?

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:DPATTERN:PATTERN "00110101":TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:DPATTERN:PATTERN?

-> :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:DPATTERN:PATTERN "00110101"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:DBYTe」の設定値によって設定できるバイト数が変動します。

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:ENDian

機能 CAN FD バス信号トリガの ID/Data 条件の判定データのエンディアンを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):CANFD[:IDData]:DATA:ENDian {BIG|LITTLE}:TRIGger{[:ATRigger]|:BTRigger):CANFD[:IDData]:DATA:ENDian?

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:ENDIAN BIG

:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:ENDIAN?

-> :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:ENDIAN BIG

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:MODE

機能 CAN FD バス信号トリガの ID/Data 条件のデータ条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):CANFD[:IDData]:DATA:MODE {<Boolean>}:TRIGger{[:ATRigger]|:BTRigger):CANFD[:IDData]:DATA:MODE?

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:MODE ON

:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:MODE?

-> :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:MSBLsb

機能 CAN FD バス信号トリガの ID/Data 条件のデータの MSB/LSB のビットを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):CANFD[:IDData]:DATA:MSBLsb {<NRf>, <NRf>}:TRIGger{[:ATRigger]|:BTRigger):CANFD[:IDData]:DATA:MSBLsb? <NRf> = 0 ~ 63

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:MSBLSB 1,0

:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:MSBLSB?

-> :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:MSBLSB 1,0

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:SIGN

機能 CAN FD バス信号トリガの ID/Data 条件のデータの符号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:SIGN {SIGN|
UNSign}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:SIGN?

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
SIGN SIGN
:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
SIGN?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
DATA:SIGN SIGN

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDENTifier?

機能 CAN FD バス信号トリガの ID/Data 条件の識別子に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDENTifier?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDENTifier:ID?

機能 CAN FD バス信号トリガの ID/Data 条件の ID に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDENTifier:ID?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDENTifier:ID:HEXa

機能 CAN FD バス信号トリガの ID/Data 条件の ID を 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDENTifier:ID:
HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:ID:HEXa "1AB"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDENTifier:MFormat」の設定値によって設定できるバイト数が変動します。

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDENTifier:ID:MODE

機能 CAN FD バス信号トリガの ID/Data 条件の ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDENTifier:ID:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDENTifier:ID<x>:
MODE?

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:ID:MODE ON
:TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:ID:MODE?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:ID:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDENTifier:ID:PATtern

機能 CAN FD バス信号トリガの ID/Data 条件の ID 条件を 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDENTifier:ID:
PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDENTifier:ID:
PATtern?

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:ID:PATTERN "11100001111"
:TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:ID:PATTERN?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:ID:PATTERN "11100001111"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDENTifier:MFormat」の設定値によって設定できるバイト数が変動します。

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDENTifier:MFormat

機能 CAN FD バス信号トリガの ID/Data 条件の ID のフレームフォーマット (標準 / 拡張) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDENTifier:
MFormat {EXTended|STANdard}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDENTifier:MFormat?

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:MFORMAT EXTENDED
:TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:MFORMAT?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:MFORMAT EXTENDED

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDENTifier:PFORMAT

機能 CAN FD バス信号トリガの ID/Data 条件の ID の入力形式を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CANFD[:IDData]:IDENTifier:
PFORMAT {MESSAge|PATtern}
:TRIGger{[:ATRigger]|:BTRigger):
CANFD[:IDData]:IDENTifier:PFORMAT?

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:PFORMAT PATTERN
:TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:PFORMAT?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:PFORMAT PATTERN

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:MSIGNAL?

機能 CAN FD バス信号トリガの ID/Data 条件のメッセージシグナルに関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CANFD[:IDData]:MSIGNAL?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:MSIGNAL:MESSAGE:ITEM

機能 CAN FD バス信号トリガの ID/Data 条件のメッセージのアイテムを設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CANFD[:IDData]:MSIGNAL:MESSAGE:
ITEM <文字列>
<文字列> = 32 文字以内

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:MESSAGE:ITEM "TEST"

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:MSIGNAL:SELECT

機能 CAN FD バス信号トリガの ID/Data 条件のメッセージシグナルの条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CANFD[:IDData]:MSIGNAL:
SELEct {MESSAge|SIGNal}
:TRIGger{[:ATRigger]|:BTRigger):
CANFD[:IDData]:MSIGNAL:SELEct?

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SELECT MESSAGE
:TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SELECT?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SELECT MESSAGE

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:MSIGNAL:SIGNAL?

機能 CAN FD バス信号トリガの ID/Data 条件のシグナルに関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CANFD[:IDData]:MSIGNAL:SIGNAL?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:MSIGNAL:SIGNAL:CONDition

機能 CAN FD バス信号トリガの ID/Data 条件のシグナルのデータ条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CANFD[:IDData]:MSIGNAL:SIGNAL:
CONDition {BETWeen|EQual|GREater|
LESS|NOTBetween|NOTEqual}
:TRIGger{[:ATRigger]|:BTRigger):
CANFD[:IDData]:MSIGNAL:SIGNAL:
CONDition?

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SIGNAL:CONDITION BETWEEN
:TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SIGNAL:CONDITION?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SIGNAL:CONDITION BETWEEN

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:MSIGNAL:SIGNAL:DECimal<x>

機能 CAN FD バス信号トリガの ID/Data 条件のシグナルの判定データを 10 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CANFD[:IDData]:MSIGNAL:SIGNAL:
DECimal<x> {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger):
CANFD[:IDData]:MSIGNAL:SIGNAL:
DECimal<x>?

<x> = 1 ~ 2
<NRf> = -9E+18 ~ 9E+18
例 :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SIGNAL:DECIMAL1 1
:TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SIGNAL:DECIMAL1?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SIGNAL:DECIMAL1 1.000E+00

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:MSIGNAL:SIGNAL:ITEM

機能 CAN FD バス信号トリガの ID/Data 条件のシグナルのアイテムを設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:MSIGNAL:SIGNAL:
ITEM {<文字列>,<文字列>}
<文字列> = 32 文字以内

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SIGNAL:ITEM "Sig_Test",
"Mess_Test"

解説 設定する順番は、シグナルアイテム、メッセージアイテムです。

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:RTR

機能 CAN FD バス信号トリガの ID/Data 条件の RTR を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:RTR {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:RTR?

例 :TRIGGER:ATRIGGER:CANFD:IDDATA:
RTR ON
:TRIGGER:ATRIGGER:CANFD:IDDATA:RTR?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
RTR 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR?

機能 CAN FD バス信号トリガの ID 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{[:ATRigger]|:BTRigger}:
CANFD:IDOR?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:ACK?

機能 CAN FD バス信号トリガの ID OR 条件の ACK に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:ACK?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:ACK:MODE

機能 CAN FD バス信号トリガの ID OR 条件の ACK モードを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:ACK:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:ACK:MODE?

例 :TRIGGER:ATRIGGER:CANFD:IDOR:ACK:
MODE ON
:TRIGGER:ATRIGGER:CANFD:IDOR:ACK:
MODE?
-> :TRIGGER:ATRIGGER:CANFD:IDOR:ACK:
MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:ACK:TYPE

機能 CAN FD バス信号トリガの ID OR 条件の ACK 条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:ACK:TYPE {ACK|ACKBoth|
NONAck}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:ACK:TYPE?

例 :TRIGGER:ATRIGGER:CANFD:IDOR:ACK:
TYPE ACK
:TRIGGER:ATRIGGER:CAN:IDOR:ACK:TYPE?
-> :TRIGGER:ATRIGGER:CANFD:IDOR:ACK:
TYPE ACK

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:DATA?

機能 CAN FD バス信号トリガの ID OR 条件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{[:ATRigger]|:BTRigger}:
CANFD:IDOR:DATA?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:DATA[:MODE]

機能 CAN FD バス信号トリガの ID OR 条件のデータ条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:DATA[:MODE] {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:DATA[:MODE]?

例 :TRIGGER:ATRIGGER:CANFD:IDOR:DATA:
MODE ON
:TRIGGER:ATRIGGER:CANFD:IDOR:DATA:
MODE?
-> :TRIGGER:ATRIGGER:CANFD:IDOR:
DATA:MODE 1

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier?

機能 CAN FD バス信号トリガの ID OR 条件の識別子に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CANFD:IDOR:IDENTifier?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier:ID<x>?

機能 CAN FD バス信号トリガの ID OR 条件の ID に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CANFD:IDOR:IDENTifier:ID<x>?
<x> = 1 ~ 4

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier:ID<x>:HEXa

機能 CAN FD バス信号トリガの ID OR 条件の ID を 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CANFD:IDOR:IDENTifier:ID<x>:
HEXa {<文字列>}
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:ID1:HEXa "1AB"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier:MFormat」の設定値によって設定できるバイト数が変動します。

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier:ID<x>:MODE

機能 CAN FD バス信号トリガの ID OR 条件の ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CANFD:IDOR:IDENTifier:ID<x>:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger):
CANFD:IDOR:IDENTifier:ID<x>:MODE?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:ID1:MODE ON
:TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:ID1:MODE?
-> :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:ID1:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier:ID<x>:PATtern

機能 CAN FD バス信号トリガの ID OR 条件の ID 条件を 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CANFD:IDOR:IDENTifier:ID<x>:
PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger):
CANFD:IDOR:IDENTifier:ID<x>:PATtern?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:ID1:PATTERN "11100001111"
:TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:ID1:PATTERN?

-> :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:ID1:PATTERN "11100001111"
解説 「:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier:MFormat」の設定値によって設定できるバイト数が変動します。

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier:MFormat

機能 CAN FD バス信号トリガの ID OR 条件の ID のフレームフォーマット (標準 / 拡張) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CANFD:IDOR:IDENTifier:
MFormat {EXTended|STANDARD}
:TRIGger{[:ATRigger]|:BTRigger):
CANFD:IDOR:IDENTifier:MFormat?

例 :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:MFORMAT EXTENDED
:TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:MFORMAT?
-> :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:MFORMAT EXTENDED

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier:MODE

機能 CAN FD バス信号トリガの ID OR 条件の識別子条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CANFD:IDOR:IDENTifier:
MODE {<Boolean>}

例 :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:MODE ON
-> :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier:PFORMAT

機能 CAN FD バス信号トリガの ID OR 条件の ID の入力形式を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:IDENTifier:
PFORMAT {MESSAge|PATtern}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:IDENTifier:PFORMAT?

例 :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:PFORMAT PATTERN
:TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:PFORMAT?
-> :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:PFORMAT PATTERN

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:MSIGNAL<x>:MESSAGE:ITEM

機能 CAN FD バス信号トリガの ID OR 条件のメッセージのアイテムを設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:MSIGNAL<x>:MESSAGE:
ITEM {<文字列>}
<x> = 1 ~ 4
<文字列> = 32 文字以内

例 :TRIGGER:ATRIGGER:CAN:IDOR:MSIGNAL1:
MESSAGE:ITEM "TEST"

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:RTR

機能 CAN FD バス信号トリガの ID OR 条件の RTR を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:RTR {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:RTR?

例 :TRIGGER:ATRIGGER:CANFD:IDOR:RTR ON
:TRIGGER:ATRIGGER:CANFD:IDOR:RTR?
-> :TRIGGER:ATRIGGER:CANFD:IDOR:
RTR 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:MODE

機能 CAN FD 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{[:ATRigger]|:BTRigger}:
CANFD:MODE {EFrame|ESI|FDF|IDData|
IDOR|SOF}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:MODE?

例 :TRIGGER:ATRIGGER:CANFD:MODE EFRAME
:TRIGGER:ATRIGGER:CANFD:MODE?
-> :TRIGGER:ATRIGGER:CANFD:
MODE EFRAME

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:REcessive

機能 CAN FD バス信号トリガのリセッシブ電位を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:REcessive {HIGH|LOW}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:REcessive?

例 :TRIGGER:ATRIGGER:CANFD:
REcessive HIGH
:TRIGGER:ATRIGGER:CANFD:REcessive?
-> :TRIGGER:ATRIGGER:CANFD:
REcessive HIGH

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:SOURce

機能 CAN FD バス信号トリガの信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:SOURce {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:SOURce?

例 :TRIGGER:ATRIGGER:CANFD:SOURce 1
:TRIGGER:ATRIGGER:CANFD:SOURce?
-> :TRIGGER:ATRIGGER:CANFD:SOURce 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:SPOint

機能 CAN FD 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{[:ATRigger]|:BTRigger}:
CANFD:SPOint {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:SPOint?

例 :TRIGGER:ATRIGGER:CANFD:SPOint 18.8
:TRIGGER:ATRIGGER:CANFD:SPOint?
-> :TRIGGER:ATRIGGER:CANFD:
SPOint 18.8

:TRIGger{[:ATRigger]|:BTRigger}:CXPI?

機能 CXPI 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{[:ATRigger]|:BTRigger}:
CXPI?

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:BRATE

機能 CXPI 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{[:ATRigger]|:BTRigger}:
CXPI:BRATE {<Nrf>|USER,<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:BRATE?

例 <Nrf> = 4800、9600、19200
USER の <Nrf> = 4000 ~ 50000
:TRIGGER:ATRIGGER:CXPI:BRATE 19200
:TRIGGER:ATRIGGER:CXPI:BRATE?
-> :TRIGGER:ATRIGGER:CXPI:
BRATE 19200

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:CTOLerance

機能 CXPI バス信号トリガのクロック許容差を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:CTOLerance {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:CTOLerance?

例 <Nrf> = ± 0.5% ~ ± 5.0% (設定分解能 0.1%)
:TRIGGER:ATRIGGER:CXPI:
CTOLERANCE 0.04
:TRIGGER:ATRIGGER:CXPI:CTOLERANCE?
-> :TRIGGER:ATRIGGER:CXPI:
CTOLERANCE 0.04

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ERROr?

機能 CXPI バス信号トリガの Error に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERROr?

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ERROr:CLOCK

機能 CXPI バス信号トリガの Clock Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERROr:CLOCK {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERROr:CLOCK?

例 :TRIGGER:ATRIGGER:CXPI:ERROR:
CLOCK ON
:TRIGGER:ATRIGGER:CXPI:ERROR:CLOCK?
-> :TRIGGER:ATRIGGER:CXPI:ERROR:
CLOCK 1

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ERROr:CRC

機能 CXPI バス信号トリガの CRC Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERROr:CRC {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERROr:CRC?

例 :TRIGGER:ATRIGGER:CXPI:ERROR:CRC ON
:TRIGGER:ATRIGGER:CXPI:ERROR:CRC?
-> :TRIGGER:ATRIGGER:CXPI:ERROR:
CRC 1

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ERROr:DLENGth

機能 CXPI バス信号トリガの Data Length Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERROr:DLENGth {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERROr:DLENGth?

例 :TRIGGER:ATRIGGER:CXPI:ERROR:
DLENGTH ON
:TRIGGER:ATRIGGER:CXPI:ERROR:
DLENGTH?
-> :TRIGGER:ATRIGGER:CXPI:ERROR:
DLENGTH 1

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ERROr:FRAMing

機能 CXPI バス信号トリガの Framing Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERROr:FRAMing {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERROr:FRAMing?

例 :TRIGGER:ATRIGGER:CXPI:ERROR:
FRAMING ON
:TRIGGER:ATRIGGER:CXPI:ERROR:
FRAMING?
-> :TRIGGER:ATRIGGER:CXPI:ERROR:
FRAMING 1

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ERROR:IBS

機能 CXPI バス信号トリガの IBS Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERROR:IBS {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERROR:IBS?

例 :TRIGGER:ATRIGGER:CXPI:ERROR:IBS ON
:TRIGGER:ATRIGGER:CXPI:ERROR:IBS?
-> :TRIGGER:ATRIGGER:CXPI:ERROR:
IBS 1

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ERROR:PARity

機能 CXPI バス信号トリガの Parity Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERROR:PARity {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERROR:PARity?

例 :TRIGGER:ATRIGGER:CXPI:ERROR:
PARITY ON
:TRIGGER:ATRIGGER:CXPI:ERROR:PARITY?
-> :TRIGGER:ATRIGGER:CXPI:ERROR:
PARITY 1

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:IDData?

機能 CXPI バス信号トリガの ID とデータ条件に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData?

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:IDData:DATA?

機能 CXPI バス信号トリガの ID とデータ条件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{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA?

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:IDData:DATA:BCount

機能 CXPI バス信号トリガの ID とデータ条件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{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:BCount {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:BCount?

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
BCOUNT 1
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
BCOUNT?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:BCOUNT 1

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:IDData:DATA:CONDition

機能 CXPI バス信号トリガの ID とデータ条件のデータ判定条件の設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:CONDition {BETween|
EQUAL|FALSE|GREATER|LESS|NOTBetween|
NOTEQUAL|TRUE}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:CONDition?

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
CONDITION BETWEEN
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
CONDITION?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:CONDITION BETWEEN

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:IDData:DATA:DBYTE

機能 CXPI バス信号トリガの ID とデータ条件のデータのバイト数を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:DBYTE {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:DBYTE?

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
DBYTE 1
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
DBYTE?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:DBYTE 1

5.34 TRIGger グループ

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:DATA:DECimal<x>**

機能 CXPI バス信号トリガの ID とデータ条件の判定データを 10 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:DECimal<x> {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:DECimal<x>?
<x> = 1、2
<NRf> = -9E+18 ~ 9E+18

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
DECIMAL1 1
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
DECIMAL1?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:DECIMAL1 1.0000000E+00

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:DATA:DPATtern?**

機能 CXPI バス信号トリガの ID とデータ条件の ID に関わるすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:DPATtern?

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:DATA:DPATtern:HEXa**

機能 CXPI バス信号トリガの ID とデータ条件の ID を 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:DPATtern:
HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
DPATTERN:HEXa "12"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:CXPI
I:IDData:DATA:DBYTE」の設定値によって設定できるバイト数が変動します。

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:DATA:DPATtern:PATtern**

機能 CXPI バス信号トリガの ID とデータ条件の ID 条件を 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:DPATtern:
PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:DPATtern:PATtern?

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
DPATTERN:PATTERN "00110101"
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
DPATTERN:PATTERN?

-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:DPATTERN:PATTERN "00110101"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:CXPI
I:IDData:DATA:DBYTE」の設定値によって設定できるバイト数が変動します。

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:DATA:ENDian**

機能 CXPI バス信号トリガの ID とデータ条件の判定データのエンディアンの設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:ENDian {BIG|LITTLE}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:ENDian?

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
ENDIAN BIG
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
ENDIAN?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:ENDIAN BIG

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:DATA:MODE**

機能 CXPI バス信号トリガの ID とデータ条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:MODE?

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
MODE ON
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
MODE?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:DATA:MSBLSb**

機能 CXPI バス信号トリガの ID とデータ条件のデータの MSB/LSB のビットを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:MSBLSb {<Nrf>,
<Nrf>}
<Nrf> = 0 ~ 63

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
MSBLSB 7,0
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
MSBLSB?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:MSBLSB 7,0

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:DATA:SIGN**

機能 CXPI バス信号トリガの ID とデータ条件のデータの符号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:SIGN {SIGN|UNSign}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:SIGN?

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
SIGN SIGN
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
SIGN?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:SIGN SIGN

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:FINformation**

機能 CXPI バス信号トリガの ID とデータ条件のフレーム情報に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:FINformation?

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:FINformation:CT**

機能 CXPI バス信号トリガの ID とデータ条件のフレーム情報のカウンタ値を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:FINformation:CT {<Nrf>|
DONTcare}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:FINformation:CT?
<Nrf> = 0 ~ 3

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFORMATION:CT 3
:TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFORMATION:CT?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFORMATION:CT 3

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:FINformation:MODE**

機能 CXPI バス信号トリガの ID とデータ条件のフレーム情報 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:FINformation:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:FINformation:MODE?

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:MODE ON
:TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:MODE?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:FINformation:SLEEP**

機能 CXPI バス信号トリガの ID とデータ条件のフレーム情報の Sleep ビットを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:FINformation:SLEEP {0|1|
X}

例 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:FINformation:SLEEP?
:TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:SLEEP 1
:TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:SLEEP?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:SLEEP 1

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:FINformation:WAKEup**

機能 CXPI バス信号トリガの ID とデータ条件のフレーム情報の Wakeup ビットを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:FINformation:WAKEup {0|
1|X}

例 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:FINformation:WAKEup
:TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:WAKEUP 1
:TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:WAKEUP?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:WAKEUP 1

5.34 TRIGger グループ

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:ID?**

機能 CXPIバス信号トリガのIDとデータ条件のIDに関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CXPI:IDData:ID?

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:ID:HEXa**

機能 CXPIバス信号トリガのIDとデータ条件のIDを16進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CXPI:IDData:ID:HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
HEXA "1E"

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:ID:MODE**

機能 CXPIバス信号トリガのIDとデータ条件のID条件(有効/無効)を設定/問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CXPI:IDData:ID:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger):
CXPI:IDData:ID:MODE?

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
MODE ON
:TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
MODE?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:ID:PATtern**

機能 CXPIバス信号トリガのIDとデータ条件のIDを2進数で設定/問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CXPI:IDData:ID:PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger):
CXPI:IDData:ID:PATtern?

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
PATTERN "0010000"
:TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
PATTERN?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
PATTERN "0010000"

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDData:ID:PTYPE**

機能 CXPIバス信号トリガのIDとデータ条件のPTYPEの判定条件を設定/問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CXPI:IDData:ID:PTYPE {DONTcare|NO|
YES}
:TRIGger{[:ATRigger]|:BTRigger):
CXPI:IDData:ID:PTYPE?

例 :TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
PTYPYE DONTCARE
:TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
PTYPYE?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
PTYPYE DONTCARE

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDOR?**

機能 CXPIバス信号トリガのIDOR条件に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CXPI:IDOR?

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDOR:IDENTifier?**

機能 CXPIバス信号トリガのIDOR条件の識別子に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CXPI:IDOR:IDENTifier?

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDOR:IDENTifier:ID<x>?**

機能 CXPIバス信号トリガのIDOR条件のIDに関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CXPI:IDOR:IDENTifier:ID<x>?
<x> = 1 ~ 4

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDOR:IDENTifier:ID<x>:HEXa**

機能 CXPIバス信号トリガのIDOR条件のIDを16進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger):
CXPI:IDOR:IDENTifier:ID<x>:
HEXa {<文字列>}
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:ID1:HEXA "1E"

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDOR:IDENTifier:ID<x>:MODE**

機能 CXPI バス信号トリガの IDOR 条件の ID 条件 (有効 / 無効) の設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:ID<x>:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:ID<x>:MODE?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:ID1:MODE ON
:TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:ID1:MODE?
-> :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:ID1:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDOR:IDENTifier:ID<x>:PATTERN**

機能 CXPI バス信号トリガの IDOR 条件の ID 条件を 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:ID<x>:
PATTERN {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:ID<x>:PATTERN?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:ID1:PATTERN "0010000"
:TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:ID1:PATTERN?
-> :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:ID1:PATTERN "0010000"

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDOR:IDENTifier:MODE**

機能 CXPI バス信号トリガの IDOR 条件の識別子条件 (有効 / 無効) の設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:MODE?

例 :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:MODE ON
:TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:MODE?
-> :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:IDOR:IDENTifier:PTYPE**

機能 CXPI バス信号トリガの IDOR 条件の PTYPE の判定条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:
PTYPE {DONTcare|NO|YES}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:PTYPE?

例 :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:PTYPE DONTCARE
:TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:PTYPE?
-> :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:PTYPE DONTCARE

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:MODE**

機能 CXPI バス信号トリガの種類を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:MODE {ERROR|IDData|IDOR|PTYPE|
SOF|WAKEupsleep}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:MODE?

例 :TRIGGER:ATRIGGER:CXPI:MODE SOF
:TRIGGER:ATRIGGER:CXPI:MODE?
-> :TRIGGER:ATRIGGER:CXPI:MODE SOF

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:PTYPE?**

機能 CXPI バス信号トリガの PTYPE に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:PTYPE?

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI
:PTYPE:CONDition**

機能 CXPI バス信号トリガの PTYPE の判定条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:PTYPE:CONDition {EPTYPE|
NOResponse}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:PTYPE:CONDition?

例 :TRIGGER:ATRIGGER:CXPI:PTYPE:
CONDITION EPTYPE
:TRIGGER:ATRIGGER:CXPI:PTYPE:
CONDITION?
-> :TRIGGER:ATRIGGER:CXPI:PTYPE:
CONDITION EPTYPE

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:SOURce

機能 CXPI バス信号トリガの信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:SOURce {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:SOURce?
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:TSAMple

機能 CXPI バス信号トリガの論理値 1/0 判定の閾値を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:TSAMple <NRf>
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:TSAMple?
<NRf> = 0.01Tbit ~ 0.3Tbit(設定分解能0.001Tbit)
例 :TRIGGER:ATRIGGER:CXPI:TSAMPLE 0.04
:TRIGGER:ATRIGGER:CXPI:TSAMPLE?
-> :TRIGGER:ATRIGGER:CXPI:
TSAMPLE 0.04

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:WAKEupsleep?

機能 CXPI バス信号トリガの Wakeup と Sleep に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:CXPI:
WAKEupsleep?

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:WAKEupsleep:TYPE

機能 CXPI バス信号トリガの Wakeup と Sleep の条件を設定 / 問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:WAKEupsleep:TYPE {SFrame|SLEEP|
WAKEup|WPULse}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:WAKEupsleep:TYPE?
例 :TRIGGER:ATRIGGER:CXPI:WAKEUPSLEEP:
TYPE SFRAME
:TRIGGER:ATRIGGER:CXPI:WAKEUPSLEEP:
TYPE?
-> :TRIGGER:ATRIGGER:CXPI:
WAKEUPSLEEP:TYPE SFRAME

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray?

機能 FlexRay 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{[:ATRigger]|:BTRigger}:
FLEXray?

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:BCHannel

機能 FlexRay バス信号トリガのチャンネルのバスタイプを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:BCHannel {A|B}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:BCHannel?

例 :TRIGGER:ATRIGGER:FLEXRAY:BCHANNEL A
:TRIGGER:ATRIGGER:FLEXRAY:BCHANNEL?
-> :TRIGGER:ATRIGGER:FLEXRAY:
BCHANNEL A

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:BRATe

機能 FlexRay 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{[:ATRigger]|:BTRigger}:
FLEXray:BRATe {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:BRATe?
<NRf> = 2500000、5000000、10000000
例 :TRIGGER:ATRIGGER:FLEXRAY:
BRATE 5000000

:TRIGGER:ATRIGGER:FLEXRAY:BRATE?
-> :TRIGGER:ATRIGGER:FLEXRAY:
BRATE 5000000

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:ERROr?

機能 FlexRay バス信号トリガの Error に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:ERROr?

例 :TRIGGER:ATRIGGER:CXPI:SOURCE 1
:TRIGGER:ATRIGGER:CXPI:SOURCE?
-> :TRIGGER:ATRIGGER:CXPI:SOURCE 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:ERROr:BSS

機能 FlexRay バス信号トリガの BSS Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:ERROr:BSS {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:ERROr:BSS?

例 :TRIGGER:ATRIGGER:FLEXRAY:ERROR:
BSS ON
:TRIGGER:ATRIGGER:FLEXRAY:ERROR:BSS?
-> :TRIGGER:ATRIGGER:FLEXRAY:ERROR:
BSS 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:ERROR:CRC

機能 FlexRay バス信号トリガの CRC Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:ERROR:CRC {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:ERROR:CRC?

例 :TRIGGER:ATRIGGER:FLEXRAY:ERROR:
CRC ON
:TRIGGER:ATRIGGER:FLEXRAY:ERROR:CRC?
-> :TRIGGER:ATRIGGER:FLEXRAY:ERROR:
CRC 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:ERROR:FES

機能 FlexRay バス信号トリガの FES Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:ERROR:FES {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:ERROR:FES?

例 :TRIGGER:ATRIGGER:FLEXRAY:ERROR:
FES ON
:TRIGGER:ATRIGGER:FLEXRAY:ERROR:FES?
-> :TRIGGER:ATRIGGER:FLEXRAY:ERROR:
FES 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:IDData?

機能 FlexRay バス信号トリガの ID とデータ条件に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDData?

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:CCOUNT?

機能 FlexRay バス信号トリガの ID とデータ条件の Cycle Count に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCOUNT?

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:CCOUNT:CONDITION

機能 FlexRay バス信号トリガの ID とデータ条件の Cycle Count のデータ条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCOUNT:
CONDITION {BETween|EQUal|GREater|
LESS|NOTBetween|NOTEqual}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCOUNT:CONDITION?
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:CONDITION BETWEEN
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:CONDITION?

例 -> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:CONDITION BETWEEN

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:CCOUNT:COUNT<x>

機能 FlexRay バス信号トリガの ID とデータ条件の Cycle Count を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCOUNT:
COUNT<x> {<NRF>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCOUNT:COUNT<x>?
<x> = 1 ~ 2
<NRF> = 0 ~ 63

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:COUNT1 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:COUNT1?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:COUNT1 1

解説

- 「:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCOUNT:CONDITION
{EQUal|GREater|NOTEqual}」のときは
「:TRIGger{[:ATRigger]|:BTRigger}:FL
EXray[:IDData]:CCOUNT:COUNT1」で設定
します。
- 「:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCOUNT:CONDITION
LESS」のときは「:TRIGger{[:ATRigger]|
:BTRigger}:FLEXray[:IDData]:CCOUNT:C
OUNT2」で設定します。
- 「:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCOUNT:CONDITION
{BETween|NOTBetween}」のときは小さい
値を「:TRIGger{[:ATRigger]|:BTRigge
r}:FLEXray[:IDData]:CCOUNT:COUNT1」、
大きい値を「:TRIGger{[:ATRigger]|:BT
Rigger}:FLEXray[:IDData]:CCOUNT:COU
Nt2」で設定します。

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:CCOUNT:MODE

機能 FlexRay バス信号トリガの ID とデータ条件の Cycle Count 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCOUNT:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCOUNT:MODE?

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:MODE ON
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:MODE?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>?

機能 FlexRay バス信号トリガの ID とデータ条件の Data Field に関するすべての設定値を問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>?
<x> = 1 ~ 2

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:BCOUNT

機能 FlexRay バス信号トリガの ID とデータ条件の Data Field 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{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:
BCOUNT {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:BCOUNT?
<x> = 1 ~ 2
<NRf> = 0 ~ 253

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:BCOUNT 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:BCOUNT?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:BCOUNT 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:CONDITION

機能 FlexRay バス信号トリガの ID とデータ条件の Data Field のデータ条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:
CONDition {BETWEEN|EQUAL|FALSE|
GREATER|LESS|NOTBetween|NOTEQUAL|
TRUE}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:CONDition?
<x> = 1 ~ 2

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:CONDITION BETWEEN
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:CONDITION?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:CONDITION BETWEEN

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:DBYTE

機能 FlexRay バス信号トリガの ID とデータ条件の Data Field のデータのバイト数を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:
DBYTE {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:DBYTE?
<x> = 1 ~ 2
<NRf> = 1 ~ 8

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:DBYTE 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:DBYTE?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:DBYTE 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:DECimal<y>

機能 FlexRay バス信号トリガの ID とデータ条件の判定データを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:
DECimal<y> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:DECimal<y>?
<x> = 1 ~ 2
<y> = 1 ~ 2
<Nrf> = 本体ユーザーズマニュアル機能編参照。

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:DECIMAL1 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:DECIMAL1?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:DECIMAL1 1.000E+00

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA:DPATtern?

機能 FlexRay バス信号トリガの ID とデータ条件の Data Field の判定データに関わるすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA:DPATtern?

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA:DPATtern:HEXa

機能 FlexRay バス信号トリガの ID とデータ条件の Data Field の判定データを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA:DPATtern:
HEXa {<文字列>}
<x> = 1 ~ 2

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:DPATTERN:HEXa "12"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:DBYTE」の設定値によって設定できるバイト数が変動します。

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA:DPATtern:PATtern

機能 FlexRay バス信号トリガの ID とデータ条件の Data Field のデータを BINARY で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA:DPATtern:
PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA:DPATtern:
PATtern?

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:DPATTERN:PATTERN "00110101"
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:DPATTERN:PATTERN?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:DPATTERN:PATTERN "00110101"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:DBYTE」の設定値によって設定できるバイト数が変動します。

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:ENDian

機能 FlexRay バス信号トリガの ID とデータ条件の Data Field のエンディアンを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:
ENDian {BIG|LITTLE}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:ENDian?
<x> = 1 ~ 2

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:ENDIAN BIG
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:ENDIAN?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:ENDIAN BIG

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:MODE

機能 FlexRay バス信号トリガの ID とデータ条件の Data Field 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
FLEXray[:IDData]:DATA<x>:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger):
FLEXray[:IDData]:DATA<x>:MODE?
<x> = 1 ~ 2

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:MODE ON
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:MODE?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:MSBLSb

機能 FlexRay バス信号トリガの ID とデータ条件の Data Field 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{[:ATRigger]|:BTRigger):
FLEXray[:IDData]:DATA<x>:
MSBLSb {<NRf>,<NRf>}
:TRIGger{[:ATRigger]|:BTRigger):
FLEXray[:IDData]:DATA<x>:MSBLSb?
<x> = 1 ~ 2
<NRf> = 本体ユーザーズマニュアル機能編参照。

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:MSBLSB 7,0
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:MSBLSB?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:MSBLSB 7,0

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:SIGN

機能 FlexRay バス信号トリガの ID とデータ条件の Data Field のデータの符号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
FLEXray[:IDData]:DATA<x>:SIGN {SIGN|
UNSign}
:TRIGger{[:ATRigger]|:BTRigger):
FLEXray[:IDData]:DATA<x>:SIGN?
<x> = 1 ~ 2

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:SIGN SIGN
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:SIGN?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:SIGN SIGN

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:FID?

機能 FlexRay バス信号トリガの ID とデータ条件の Frame ID に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger):
FLEXray[:IDData]:FID?

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:FID:CONDition

機能 FlexRay バス信号トリガの ID とデータ条件の Frame ID のデータ条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
FLEXray[:IDData]:FID:
CONDition {BETWEEN|EQUAL|GREATER|
LESS|NOTBetween|NOTEQUAL}
:TRIGger{[:ATRigger]|:BTRigger):
FLEXray[:IDData]:FID:CONDition?
例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:CONDITION BETWEEN
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:CONDITION?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:CONDITION BETWEEN

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:FID:ID<x>

機能 FlexRay バス信号トリガの ID とデータ条件の Frame ID の値を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
FLEXray[:IDData]:FID:ID<x> {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger):
FLEXray[:IDData]:FID:ID<x>?
<x> = 1 ~ 2
<NRf> = 1 ~ 2047
例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:ID1 100
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:ID1?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:ID1 100

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:FID:MODE

機能 FlexRay バス信号トリガの ID とデータ条件の Frame ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:FID:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:FID:MODE?

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:MODE ON
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:MODE?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:INDicator?

機能 FlexRay バス信号トリガの ID とデータ条件の Indicator に関するすべての設定値を問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator?

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:INDicator:MODE

機能 FlexRay バス信号トリガの ID とデータ条件の Indicator 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:MODE?

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDicator:MODE ON
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDicator:MODE?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDicator:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:INDicator:NFRame

機能 FlexRay バス信号トリガの ID とデータ条件の Indicator の Null frame を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:
NFRame {0|1|X}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:NFRame?

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDicator:NFRAME 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDicator:NFRAME?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDicator:NFRAME 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:INDicator:PPReamble

機能 FlexRay バス信号トリガの ID とデータ条件の Indicator の Payload preamble を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:
PPReamble {0|1|X}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:
PPReamble?

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDicator:PPREMBLE 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDicator:PPREMBLE?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDicator:PPREMBLE 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:INDicator:STFRame

機能 FlexRay バス信号トリガの ID とデータ条件の Indicator の Start frame を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:
STFRame {0|1|X}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:STFRame?

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDicator:STFRAME 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDicator:STFRAME?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDicator:STFRAME 1

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:INDIcator:SYFRame

機能 FlexRay バス信号トリガの ID とデータ条件の Indicator の Sync frame を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDIcator:
SYFRame {0|1|X}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDIcator:SYFRame?

例 :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:SYFRAME 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:SYFRAME?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:SYFRAME 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:IDOR?

機能 FlexRay 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{[:ATRigger]|:BTRigger}:
FLEXray:IDOR?

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:IDOR:ID<x>?

機能 FlexRay バス信号トリガの OR 条件の各 Frame ID に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR:ID<x>?
<x> = 1 ~ 4

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:IDOR:ID<x>:CCOunt?

機能 FlexRay バス信号トリガの OR 条件の各 Frame ID の Cycle Count に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR:ID<x>:CCOunt?
<x> = 1 ~ 4

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:IDOR:ID<x>:CCOunt:CONDition

機能 FlexRay バス信号トリガの OR 条件の各 Frame ID の Cycle Count の判定条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR:ID<x>:CCOunt:
CONDition {BETWeen|DONTcare|EQUAL|
GREater|LESS|NOTBetween|NOTEqual}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR:ID<x>:CCOunt:CONDition?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
CCOUNT:CONDITION BETWEEN
:TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
CCOUNT:CONDITION?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDOR:
ID1:CCOUNT:CONDITION BETWEEN

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:IDOR:ID<x>:CCOunt:COUNt<y>

機能 FlexRay バス信号トリガの OR 条件の各 Frame ID の Cycle Count の判定データを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR:ID<x>:CCOunt:
COUNt<y> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR:ID<x>:CCOunt:COUNt<y>?
<x> = 1 ~ 4
<y> = 1 ~ 2
<Nrf> = 0 ~ 63

例 :TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
CCOUNT:COUNt1 100
:TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
CCOUNT:COUNt1?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDOR:
ID1:CCOUNT:COUNt1 100

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:IDOR:ID<x>:FID?

機能 FlexRay バス信号トリガの OR 条件の各 Frame ID の ID 条件に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR:ID<x>:FID?
<x> = 1 ~ 4

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:IDOR:ID<x>:FID:CONDition

機能 FlexRay バス信号トリガの OR 条件の各 Frame ID の ID の判定条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR:ID<x>:FID:
CONDition {BETween|EQual|GREater|
LESS|NOTBetween|NOTEqual}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR:ID<x>:FID:CONDition?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
FID:CONDITION BETWEEN
:TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
FID:CONDITION?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDOR:
ID1:FID:CONDITION BETWEEN

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:IDOR:ID<x>:FID:ID<y>

機能 FlexRay バス信号トリガの OR 条件の各 Frame ID の ID の判定データを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR:ID<x>:FID:ID<y> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR:ID<x>:FID:ID<y>?
<x> = 1 ~ 4
<y> = 1 ~ 2
<Nrf> = 1 ~ 2047

例 :TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
FID:ID1 1
:TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
FID:ID1?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDOR:
ID1:FID:ID1 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:IDOR:ID<x>:MODE

機能 FlexRay バス信号トリガの OR 条件の各 Frame ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR:ID<x>:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR:ID<x>:MODE?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
MODE ON
:TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
MODE?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDOR:
ID1:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:MODE

機能 FlexRay 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{[:ATRigger]|:BTRigger}:
FLEXray:MODE {Error|FStart|IDData|
IDOR}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:MODE?

例 :TRIGGER:ATRIGGER:FLEXRAY:
MODE FSTART
:TRIGGER:ATRIGGER:FLEXRAY:MODE?
-> :TRIGGER:ATRIGGER:FLEXRAY:
MODE FSTART

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:SOURce

機能 FlexRay 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{[:ATRigger]|:BTRigger}:
FLEXray:SOURce {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:SOURce?
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :TRIGGER:ATRIGGER:FLEXRAY:SOURCE 1
:TRIGGER:ATRIGGER:FLEXRAY:SOURCE?
-> :TRIGGER:ATRIGGER:FLEXRAY:
SOURCE 1

:TRIGger{[:ATRigger]|:BTRigger}:I2C?

機能 I2C 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{[:ATRigger]|:BTRigger}:I2C?

:TRIGger{[:ATRigger]|:BTRigger}:I2C:ADDRESS?

機能 I2C バス信号トリガのアドレス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 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS?

:TRIGger{[:ATRigger]|:BTRigger}:I2C:ADDRESS:ADDRESS?

機能 I2C バス信号トリガのアドレスパターン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{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS?

5.34 TRIGger グループ

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT10Address?**

機能 I2C バス信号トリガの 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{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT10Address?

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT10Address:DIREcti
on**

機能 I2C バス信号トリガの 10bit アドレスのデータ方向を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT10Address:
DIREction {READ|WRITE|X}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT10Address:
DIREction?

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT10ADDRESS:DIRECTION READ
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT10ADDRESS:DIRECTION?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT10ADDRESS:DIRECTION READ

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT10Address:HEXa**

機能 I2C バス信号トリガの 10bit アドレスを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT10Address:
HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT10ADDRESS:HEXA "1AB"

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT10Address:PATtern**

機能 I2C バス信号トリガの 10bit アドレスを 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT10Address:
PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT10Address:
PATtern?

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT10ADDRESS:
PATTERN "11010111001"
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT10ADDRESS:PATTERN?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT10ADDRESS:
PATTERN "11010111001"

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7Address?**

機能 I2C バス信号トリガの 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{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7Address?

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS[:BIT7Address]:DIREct
ion**

機能 I2C バス信号トリガの 7bit アドレスのデータ方向を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS[:BIT7Address]:
DIREction {READ|WRITE|X}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS[:BIT7Address]:
DIREction?

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7ADDRESS:DIRECTION READ
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7ADDRESS:DIRECTION?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7ADDRESS:DIRECTION READ

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS[:BIT7Address]:HEXa**

機能 I2C バス信号トリガの 7bit アドレスを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7Address:
HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7ADDRESS:HEXA "5C"

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS[:BIT7Address]:PATte
rn**

機能 I2C バス信号トリガの 7bit アドレスを 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS[:BIT7Address]:
PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS[:BIT7Address]:
PATtern?

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7ADDRESS:
PATTERN "11100110"
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7ADDRESS:PATTERN?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7ADDRESS:
PATTERN "11100110"

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub?**

機能 I2C バス信号トリガの 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{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub?

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS?**

機能 I2C バス信号トリガの 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{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS?

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:DI
Rection**

機能 I2C バス信号トリガの 7bit+Sub アドレスのデータ方向を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
DIRrection {READ|WRITE|X}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
DIRrection?

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:ADDRESS:
DIRECTION READ
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:ADDRESS:DIRECTION?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:ADDRESS:
DIRECTION READ

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:HE
Xa**

機能 I2C バス信号トリガの 7bit+Sub アドレスの 7bit アドレスを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:ADDRESS:HEXA "AB"

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:PA
TTern**

機能 I2C バス信号トリガの 7bit+Sub アドレスの 7bit アドレスを 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
PATtern?

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:ADDRESS:
PATTERN "00111010"
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:ADDRESS:PATTERN?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:ADDRESS:
PATTERN "00111010"

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:SADDRESS?**

機能 I2C バス信号トリガの 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{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:SADDRESS?

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:SADDRESS:H
EXa**

機能 I2C バス信号トリガの 7bit+Sub アドレスの Sub アドレスを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:SADDRESS:
HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:SADDRESS:HEXA "EF"

5.34 TRIGger グループ

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:SADDRESS:P
ATtern**

機能 I2C バス信号トリガの 7bit+Sub アドレスの Sub
アドレスを 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:SADDRESS:
PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:SADDRESS:
PATtern?

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:SADDRESS:
PATTERN "00111010"
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:SADDRESS:PATTERN?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:SADDRESS:
PATTERN "00111010"

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:MODE**

機能 I2C バス信号トリガのアドレス条件の有効 / 無効
を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:MODE?

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:MODE ON
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:MODE?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:TYPE**

機能 I2C バス信号トリガのアドレス条件のアドレス形
式を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:TYPE {BIT10Address|
BIT7Address|BIT7APsub}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:TYPE?

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:TYPE BIT10ADDRESS
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:TYPE?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:TYPE BIT10ADDRESS

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA?**

機能 I2C バス信号トリガのアドレスパターンのデー
タに関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA?

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:BCount**

機能 I2C バス信号トリガのデータのパターンを比較す
る位置を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:BCount {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:BCount?
<Nrf> = 0 ~ 9999

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
BCOUNT 0
:TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
BCOUNT?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
DATA:BCOUNT 0

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:BMode**

機能 I2C バス信号トリガのデータのパターン比較する
位置の ON/OFF を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:BMode {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:BMode?

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
BMODE ON
:TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
BMODE?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
DATA:BMODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:CONDition**

機能 I2C バス信号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{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:CONDition {FALSE|TRUE}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:CONDition?

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
CONDITION FALSE
:TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
CONDITION?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
DATA:CONDITION FALSE

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:DBYTE**

機能 I2C バス信号トリガの設定データ数を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:DBYTE {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:DBYTE?
<NRf> = 1 ~ 4

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
DBYTE 1
:TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
DBYTE?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
DATA:DBYTE 1

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:DPATtern?**

機能 I2C バス信号トリガのデータに関する設定値をすべて問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:DPATtern?

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:DPATtern:HEXa**

機能 I2C バス信号トリガのデータを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:DPATtern:HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
DPATtern:HEXa "AB"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:I2C:
:ADDRESS:DATA:DBYTE」の設定値によって設定
できるバイト数が変動します。

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:DPATtern:PATtern**

機能 I2C バス信号トリガのデータ条件のデータを 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:DPATtern:
PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:DPATtern:PATtern?

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
DPATtern:PATtern "10001101"
:TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
DPATtern:PATtern?

-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
DATA:DPATtern:PATtern1 "10001101"
解説 「:TRIGger{[:ATRigger]|:BTRigger}:I2C:
:ADDRESS:DATA:DBYTE」の設定値によって設定
できるバイト数が変動します。

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:MODE**

機能 I2C バス信号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{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:DATA:MODE?

例 :TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
MODE ON
:TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
MODE?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
DATA:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
GENeralcall?**

機能 I2C バス信号トリガのゼ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{[:ATRigger]|:BTRigger}:I2C:
GENeralcall?

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
GENeralcall:BIT7Maddress?**

機能 I2C バス信号トリガのゼネラルコールの 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{[:ATRigger]|:BTRigger}:I2C:
GENeralcall:BIT7Maddress?

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
GENeralcall:BIT7Maddress:HEXa**

機能 I2C バス信号トリガのゼネラルコールの 7bit マスタアドレスを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
GENeralcall:BIT7Maddress:
HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:I2C:GENERALCALL:
BIT7MADDRESS:HEXa "AB"

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:I2C:GENeralcall:BIT7Maddress:PATtern

機能 I2C バス信号トリガのゼネラルコールの 7bit マスタアドレスを 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:GENeralcall:BIT7Maddress:PATtern {<文字列>}

:TRIGger{[:ATRigger]|:BTRigger}:I2C:GENeralcall:BIT7Maddress:PATtern?

例 :TRIGGER:ATRIGGER:I2C:GENERALCALL:BIT7MADDRESS:PATTERN "0010110"

:TRIGGER:ATRIGGER:I2C:GENERALCALL:BIT7MADDRESS:PATTERN?

-> :TRIGGER:ATRIGGER:I2C:

GENERALCALL:BIT7MADDRESS:

PATTERN "0010110"

:TRIGger{[:ATRigger]|:BTRigger}:I2C:GENeralcall:SBYTE

機能 I2C バス信号トリガのゼ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{[:ATRigger]|:BTRigger}:I2C:GENeralcall:SBYTE {BIT7Maddress|DONTcare|H04|H06}

:TRIGger{[:ATRigger]|:BTRigger}:I2C:GENeralcall:SBYTE?

例 :TRIGGER:ATRIGGER:I2C:GENERALCALL:SBYTE BIT7MADDRESS

:TRIGGER:ATRIGGER:I2C:GENERALCALL:SBYTE?

-> :TRIGGER:ATRIGGER:I2C:

GENERALCALL:SBYTE BIT7MADDRESS

:TRIGger{[:ATRigger]|:BTRigger}:I2C:INCLuderw

機能 I2C バス信号トリガのアドレス R/W の有効 (ON)/無効 (OFF) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:INCLuderw {<Boolean>}

:TRIGger{[:ATRigger]|:BTRigger}:I2C:INCLuderw?

例 :TRIGGER:ATRIGGER:I2C:INCLUDERW ON

:TRIGGER:ATRIGGER:I2C:INCLUDERW?

-> :TRIGGER:ATRIGGER:I2C:INCLUDERW 1

:TRIGger{[:ATRigger]|:BTRigger}:I2C:MODE

機能 I2C 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{[:ATRigger]|:BTRigger}:I2C:MODE {ADRData|EVERystart|GENeralcall|HSMode|NONack|STARTbyte}:TRIGger{[:ATRigger]|:BTRigger}:I2C:MODE?

例 :TRIGGER:ATRIGGER:I2C:MODE ADRDATA

:TRIGGER:ATRIGGER:I2C:MODE?

-> :TRIGGER:ATRIGGER:I2C:

MODE ADRDATA

:TRIGger{[:ATRigger]|:BTRigger}:I2C:NONack?

機能 I2C バス信号トリガの 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{[:ATRigger]|:BTRigger}:I2C:NONack?

:TRIGger{[:ATRigger]|:BTRigger}:I2C:NONack:HSMode

機能 I2C バス信号トリガのハイスピードモードで NONACK を無視する / しないを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:NONack:HSMode {<Boolean>}

:TRIGger{[:ATRigger]|:BTRigger}:I2C:NONack:HSMode?

例 :TRIGGER:ATRIGGER:I2C:NONACK:

HSMODE ON

:TRIGGER:ATRIGGER:I2C:NONACK:HSMODE?

-> :TRIGGER:ATRIGGER:I2C:NONACK:

HSMODE 1

:TRIGger{[:ATRigger]|:BTRigger}:I2C:NONack:READaccess

機能 I2C バス信号トリガのリードアクセスバイトで NON ACK を無視する / しないを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:NONack:READaccess {<Boolean>}

:TRIGger{[:ATRigger]|:BTRigger}:I2C:NONack:READaccess?

例 :TRIGGER:ATRIGGER:I2C:NONACK:

READACCESS ON

:TRIGGER:ATRIGGER:I2C:NONACK:

READACCESS?

-> :TRIGGER:ATRIGGER:I2C:NONACK:

READACCESS 1

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
NONack:STARTbyte**

機能 I2C バス信号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{[:ATRigger]|:BTRigger}:I2C:
NONack:STARTbyte {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
NONack:STARTbyte?

例 :TRIGGER:ATRIGGER:I2C:NONACK:
STARTBYTE ON
:TRIGGER:ATRIGGER:I2C:NONACK:
STARTBYTE?
-> :TRIGGER:ATRIGGER:I2C:NONACK:
STARTBYTE 1

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
SCL**

機能 I2C バス信号トリガのクロック信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
SCL {<Nrf>|PODA<x>|PODB<x>|PODC<x>|
PODD<x>}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
SCL?
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<x>、PODB<x>、PODC<x>、PODD<x>
の <x> = 0 ~ 7

例 :TRIGGER:ATRIGGER:I2C:SCL 1
:TRIGGER:ATRIGGER:I2C:SCL?
-> :TRIGGER:ATRIGGER:I2C:SCL 1

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:
SDA**

機能 I2C バス信号トリガのデータ信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:I2C:
SDA {<Nrf>|PODA<x>|PODB<x>|PODC<x>|
PODD<x>}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
SDA?
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<x>、PODB<x>、PODC<x>、PODD<x>
の <x> = 0 ~ 7

例 :TRIGGER:ATRIGGER:I2C:SDA 1
:TRIGGER:ATRIGGER:I2C:SDA?
-> :TRIGGER:ATRIGGER:I2C:SDA 1

解説 「:TRIGger{[:ATRigger]|:BTRigger}:I2C:
:SCL」の設定によって設定できるデータ信号が異
なります。詳しくは本体ユーザーズマニュアル機
能編をご覧ください。

:TRIGger{[:ATRigger]|:BTRigger}:LIN?

機能 LIN バス信号トリガに関するすべての設定値を問
い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN?

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
BLENgth**

機能 LIN バス信号トリガの Break length を設定 / 問
い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
BLENgth {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:
BLENgth?

例 <Nrf> = 10 ~ 13
:TRIGGER:ATRIGGER:LIN:BLENGTH 10
:TRIGGER:ATRIGGER:LIN:BLENGTH?
-> :TRIGGER:ATRIGGER:LIN:BLENGTH 10

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
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{[:ATRigger]|:BTRigger}:LIN:
BRATe {<Nrf>|USER,<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:
BRATe?

<Nrf> = 1200、2400、4800、9600、19200
USER の <Nrf> = 本体ユーザーズマニュアル機能
編参照。

例 :TRIGGER:ATRIGGER:LIN:BRATE 2400
:TRIGGER:ATRIGGER:LIN:BRATE?
-> :TRIGGER:ATRIGGER:LIN:BRATE 2400

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
ERRor?**

機能 LIN バス信号トリガの Error に関するすべての設
定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
ERRor?

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
ERRor:PARity**

機能 LIN バス信号トリガの Parity Error を設定 / 問
い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
ERRor:PARity {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:
ERRor:PARity?

例 :TRIGGER:ATRIGGER:LIN:ERROR:
PARITY ON
:TRIGGER:ATRIGGER:LIN:ERROR:PARITY?
-> :TRIGGER:ATRIGGER:LIN:ERROR:
PARITY 1

5.34 TRIGger グループ

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
ERROR:SYNCh**

機能 LIN バス信号トリガの Synch Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
ERROR:SYNCh {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:
ERROR:SYNCh?

例 :TRIGGER:ATRIGGER:LIN:ERROR:SYNCH ON
:TRIGGER:ATRIGGER:LIN:ERROR:SYNCH?
-> :TRIGGER:ATRIGGER:LIN:ERROR:
SYNCH 1

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData?**

機能 LIN バス信号トリガの ID と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{[:ATRigger]|:BTRigger}:LIN:
IDData?

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA?**

機能 LIN 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{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA?

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:CONDition**

機能 LIN 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{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:CONDition {BETween|
EQUAL|FALSE|GREATER|LESS|NOTBetween|
NOTEQUAL|TRUE}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:CONDition?

例 :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
CONDITION BETWEEN
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
CONDITION?
-> :TRIGGER:ATRIGGER:LIN:IDDATA:
DATA:CONDITION BETWEEN

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:DBYTe**

機能 LIN バス信号トリガのデータのバイト数を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:DBYTe {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:DBYTe?
<NRf> = 1 ~ 8

例 :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
DBYTE 1
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
DBYTE?
-> :TRIGGER:ATRIGGER:LIN:IDDATA:
DATA:DBYTE 1

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:DECimal<x>**

機能 LIN バス信号トリガのデータを 10 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:DECimal<x> {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:DECimal<x>?
<x> = 1 ~ 2

例 <NRf> = 本体ユーザズマニュアル機能編参照。
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
DECIMAL1 1
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
DECIMAL1?
-> :TRIGGER:ATRIGGER:LIN:IDDATA:
DATA:DECIMAL1 1.000E+00

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:DPATtern?**

機能 LIN 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{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:DPATtern?

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:DPATtern:HEXa**

機能 LIN バス信号トリガのデータを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:DPATtern:HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
DPATTERN:HEXa "12"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:DBYTe」の設定値によって設定
できるバイト数が変動します。

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:DPATtern:PATtern**

機能 LIN バス信号トリガのデータを 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:DPATtern:PATtern {<文字列>}

:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:DPATtern:PATtern?

例 :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
DPATTERN:PATTERN "00110101"
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
DPATTERN:PATTERN?

-> :TRIGGER:ATRIGGER:LIN:IDDATA:
DATA:DPATTERN:PATTERN "00110101"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:DBYTe」の設定値によって設定
できるバイト数が変動します。

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:ENDian**

機能 LIN バス信号トリガのデータのエンディアンを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:ENDian {BIG|LiTTle}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:ENDian?

例 :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
ENDIAN BIG
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
ENDIAN?

-> :TRIGGER:ATRIGGER:LIN:IDDATA:
DATA:ENDIAN BIG

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:MODE**

機能 LIN バス信号トリガのデータ条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:MODE?

例 :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
MODE ON
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
MODE?

-> :TRIGGER:ATRIGGER:LIN:IDDATA:
DATA:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:MSBLsb**

機能 LIN 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{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:MSBLsb {<Nrf>,<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:MSBLsb?

<Nrf> = 0 ~ 63

例 :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
MSBLSB 0
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
MSBLSB?

-> :TRIGGER:ATRIGGER:LIN:IDDATA:
DATA:MSBLSB 0

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:SIGN**

機能 LIN 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{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:SIGN {SIGN|UNSign}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:DATA:SIGN?

例 :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
SIGN SIGN
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:
SIGN?

-> :TRIGGER:ATRIGGER:LIN:IDDATA:
DATA:SIGN SIGN

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:IDENTifier?**

機能 LIN バス信号トリガの識別子に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:IDENTifier?

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:IDENTifier:ID?**

機能 LIN バス信号トリガの ID に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:IDENTifier:ID?

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:IDENTifier:ID:HEXa**

機能 LIN バス信号トリガの ID を 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:IDENTifier:ID:HEXa {<文字列>}
:TRIGGER:ATRIGGER:LIN:IDDATA:
IDENTIFIER:ID:HEXa "1E"

5.34 TRIGger グループ

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:IDENtifier:ID:MODE**

機能 LIN バス信号トリガの ID とデータ条件の ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:IDENtifier:ID:
MODE {<Boolean>}

:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:IDENtifier:ID:MODE?

例 :TRIGGER:ATRIGGER:LIN:IDDATA:
IDENTIFIER:ID:MODE ON
:TRIGGER:ATRIGGER:LIN:IDDATA:
IDENTIFIER:ID:MODE?
-> :TRIGGER:ATRIGGER:LIN:IDDATA:
IDENTIFIER:ID:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:IDENtifier:ID:PATtern**

機能 LIN バス信号トリガの ID を 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:IDENtifier:ID:PATtern {<文字列
>}

:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDData:IDENtifier:ID:PATtern?

例 :TRIGGER:ATRIGGER:LIN:IDDATA:
IDENTIFIER:ID:PATTERN "101100"
:TRIGGER:ATRIGGER:LIN:IDDATA:
IDENTIFIER:ID:PATTERN?
-> :TRIGGER:ATRIGGER:LIN:IDDATA:
IDENTIFIER:ID:PATTERN "101100"

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDOR?**

機能 LIN バス信号トリガの ID 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{[:ATRigger]|:BTRigger}:LIN:
IDOR?

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDOR:IDENtifier?**

機能 LIN バス信号トリガの ID OR 条件の ID に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDOR:IDENtifier?

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDOR:IDENtifier:ID<x>?**

機能 LIN バス信号トリガの ID OR 条件のそれぞれの ID に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDOR:IDENtifier:ID<x>?
<x> = 1 ~ 4

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDOR:IDENtifier:ID<x>:HEXa**

機能 LIN バス信号トリガの ID OR 条件の ID を 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDOR:IDENtifier:ID<x>:HEXa {<文字列>}
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:LIN:IDOR:
IDENTIFIER:ID1:HEXa "1E"

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDOR:IDENtifier:ID<x>:MODE**

機能 LIN バス信号トリガの ID OR 条件の ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDOR:IDENtifier:ID<x>:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDOR:IDENtifier:ID<x>:MODE?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:LIN:IDOR:
IDENTIFIER:ID1:MODE ON
:TRIGGER:ATRIGGER:LIN:IDOR:
IDENTIFIER:ID1:MODE?
-> :TRIGGER:ATRIGGER:LIN:IDOR:
IDENTIFIER:ID1:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDOR:IDENtifier:ID<x>:PATtern**

機能 LIN バス信号トリガの ID OR 条件の ID 条件を 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDOR:IDENtifier:ID<x>:
PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:
IDOR:IDENtifier:ID<x>:PATtern?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:LIN:IDOR:
IDENTIFIER:ID1:PATTERN "101100"
:TRIGGER:ATRIGGER:LIN:IDOR:
IDENTIFIER:ID1:PATTERN?
-> :TRIGGER:ATRIGGER:LIN:IDOR:
IDENTIFIER:ID1:PATTERN "101100"

:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDOR:IDENTifier:MODE

機能 LIN バス信号トリガの ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:IDOR:IDENTifier:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDOR:IDENTifier:MODE?

例 :TRIGGER:ATRIGGER:LIN:IDOR:IDENTIFIER:MODE ON
:TRIGGER:ATRIGGER:LIN:IDOR:IDENTIFIER:MODE?
-> :TRIGGER:ATRIGGER:LIN:IDOR:IDENTIFIER:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:LIN:MODE

機能 LIN 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{[:ATRigger]|:BTRigger}:LIN:MODE {BSYNch|ERRor|IDData|IDOR}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:MODE?

例 :TRIGGER:ATRIGGER:LIN:MODE BSYNCH
:TRIGGER:ATRIGGER:LIN:MODE?
-> :TRIGGER:ATRIGGER:LIN:MODE BSYNCH

:TRIGger{[:ATRigger]|:BTRigger}:LIN:SOURce

機能 LIN バス信号トリガの信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:LIN:SOURce {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:SOURce?

<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例 :TRIGGER:ATRIGGER:LIN:SOURCE 1
:TRIGGER:ATRIGGER:LIN:SOURCE?
-> :TRIGGER:ATRIGGER:LIN:SOURCE 1

:TRIGger{[:ATRigger]|:BTRigger}:LIN:SPOint

機能 LIN 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{[:ATRigger]|:BTRigger}:LIN:SPOint {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:SPOint?

<Nrf> = 18.8 ~ 90.6
例 :TRIGGER:ATRIGGER:LIN:SPOINT 18.8
:TRIGGER:ATRIGGER:LIN:SPOINT?
-> :TRIGGER:ATRIGGER:LIN:SPOINT 18.8

:TRIGger{[:ATRigger]|:BTRigger}:OR?

機能 エッジ OR 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{[:ATRigger]|:BTRigger}:OR?

:TRIGger{[:ATRigger]|:BTRigger}:OR:ALL

機能 すべてのチャネルを一括して、エッジ OR トリガのトリガソースの対象にする / しないを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:OR:ALL {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:OR:ALL?

例 :TRIGGER:ATRIGGER:OR:ALL ON
:TRIGGER:ATRIGGER:OR:ALL?
-> :TRIGGER:ATRIGGER:OR:ALL 1

:TRIGger{[:ATRigger]|:BTRigger}:OR:CHANNEL<x>

機能 エッジ OR トリガの各チャネルのスロープを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:OR:CHANNEL<x>{BOTH|FALL|OFF|RISE}
:TRIGger{[:ATRigger]|:BTRigger}:OR:CHANNEL<x>?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:OR:CHANNEL1 RISE
:TRIGGER:ATRIGGER:OR:CHANNEL1?
-> :TRIGGER:ATRIGGER:OR:CHANNEL1 RISE

:TRIGger{[:ATRigger]|:BTRigger}:PATTERN?

機能 複数入力の組み合わせ (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{[:ATRigger]|:BTRigger}:PATTERN?

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:PATTERN:CHANNEL<x>

機能 複数入力の組み合わせ (パターン) トリガのときの各チャンネルのステートを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:CHANNEL<x> {DONTcare|HIGH|LOW}
:TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:CHANNEL<x>?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:PATTERN:
CHANNEL1 DONTCARE
:TRIGGER:ATRIGGER:PATTERN:CHANNEL1?
-> :TRIGGER:ATRIGGER:PATTERN:
CHANNEL1 DONTCARE

:TRIGger{[:ATRigger]|:BTRigger}:PATTERN:CLOCK?

機能 複数入力の組み合わせ (パターン) トリガの時のクロック信号に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:CLOCK?

:TRIGger{[:ATRigger]|:BTRigger}:PATTERN:CLOCK:SLOPe

機能 複数入力の組み合わせ (パターン) トリガの時のクロック信号のスロープを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:CLOCK:SLOPe {BOTH|FALL|RISE}
:TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:CLOCK:SLOPe?

例 :TRIGGER:ATRIGGER:PATTERN:CLOCK:
SLOPe FALL
:TRIGGER:ATRIGGER:PATTERN:CLOCK:
SLOPe?
-> :TRIGGER:ATRIGGER:PATTERN:CLOCK:
SLOPe FALL

解説 「:TRIGger:PATTERN:CLOCK:SOURCE NONE」以外になっていないと、エラーになります。

:TRIGger{[:ATRigger]|:BTRigger}:PATTERN:CLOCK:SOURCE

機能 複数入力の組み合わせ (パターン) トリガの時のクロック信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:CLOCK:SOURCE {<Nrf>|NONE|
PODA<x>|PODB<x>|PODC<x>|PODD<x>}
:TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:CLOCK:SOURCE?
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<x>、PODB<x>、PODC<x>、PODD<x>
の <x> = 0 ~ 7

例 :TRIGGER:ATRIGGER:PATTERN:CLOCK:
SOURCE 1
:TRIGGER:ATRIGGER:PATTERN:CLOCK:
SOURCE?
-> :TRIGGER:ATRIGGER:PATTERN:CLOCK:
SOURCE 1

:TRIGger{[:ATRigger]|:BTRigger}:PATTERN:CONDITION

機能 複数入力の組み合わせ (パターン) トリガのときのトリガ条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:CONDITION {ENTER|EXIT|FALSE|TRUE}
:TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:CONDITION?

例 :TRIGGER:ATRIGGER:PATTERN:
CONDITION ENTER
:TRIGGER:ATRIGGER:PATTERN:CONDITION?
-> :TRIGGER:ATRIGGER:PATTERN:
CONDITION ENTER

:TRIGger{[:ATRigger]|:BTRigger}:PATTERN:LOGic

機能 複数入力の組み合わせ (パターン) トリガのときの組み合わせ条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:LOGic {AND|OR}
:TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:LOGic?

例 :TRIGGER:ATRIGGER:PATTERN:LOGIC AND
:TRIGGER:ATRIGGER:PATTERN:LOGIC?
-> :TRIGGER:ATRIGGER:PATTERN:
LOGIC AND

:TRIGger{[:ATRigger]|:BTRigger}:PATTERN:{PODA|PODB|PODC|PODD}?

機能 複数入力の組み合わせ (パターン) トリガのロジック入力に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:{PODA|PODB|PODC|PODD}?

:TRIGger{[:ATRigger]|:BTRigger}:PATTERN:{PODA|PODB|PODC|PODD}:HEXa

機能 複数入力の組み合わせ (パターン) トリガのロジック入力の状態を 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:{PODA|PODB|PODC|PODD}:
HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:PATTERN:PODA:
HEXA "1F"

:TRIGger{[:ATRigger]|:BTRigger}:PATTERN:{PODA|PODB|PODC|PODD}:PATTERN

機能 複数入力の組み合わせ (パターン) トリガのロジック入力のパターンで設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:{PODA|PODB|PODC|PODD}:
PATTERN {<文字列>}

:TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:{PODA|PODB|PODC|PODD}:
PATTERN?

例 :TRIGGER:ATRIGGER:PATTERN:PODA:
PATTERN "10111000"
:TRIGGER:ATRIGGER:PATTERN:PODA:
PATTERN?
-> :TRIGGER:ATRIGGER:PATTERN:PODA:
PATTERN "10111000"

:TRIGger{[:ATRigger]|:BTRigger}:PATTERN:TQUALify

機能 複数入力の組み合わせ (パターン) トリガの時間条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:TQUALify {INSide|LTHan|
MTHan|OUTSide|TIMEout}

:TRIGger{[:ATRigger]|:BTRigger}:
PATTERN:TQUALify?

例 :TRIGGER:ATRIGGER:PATTERN:
TQUALIFY INSIDE
:TRIGGER:ATRIGGER:PATTERN:TQUALIFY?
-> :TRIGGER:ATRIGGER:PATTERN:
TQUALIFY INSIDE

:TRIGger{[:ATRigger]|:BTRigger}:PSI5?

機能 PSI5 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{[:ATRigger]|:BTRigger}:
PSI5?

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:DATA?

機能 PSI5 信号のデータ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{[:ATRigger]|:BTRigger}:
PSI5:DATA?

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:DATA:CONDition

機能 PSI5 信号のデータトリガの判定条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:DATA:CONDition {EQUAL|FALSe|
GREater|INSide|LESS|NOTEQual|
OUTSide|TRUE}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:DATA:CONDition?

例 :TRIGGER:ATRIGGER:PSI5:DATA:
CONDition EQUAL
:TRIGGER:ATRIGGER:PSI5:DATA:
CONDition?
-> :TRIGGER:ATRIGGER:PSI5:DATA:
CONDition EQUAL

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:DATA:DECimal<x>

機能 PSI5 信号のデータトリガのデータを 10 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:DATA:DECimal<x> {<NRF>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:DATA:DECimal<x>?

<x> = 1, 2
<NRF> = 10 ビットデータのとき、-512 ~ 511
16 ビットデータのとき、-32768 ~ 32767

例 :TRIGGER:ATRIGGER:PSI5:DATA:
DECIMAL1 -10
:TRIGGER:ATRIGGER:PSI5:DATA:
DECIMAL1?
-> :TRIGGER:ATRIGGER:PSI5:DATA:
DECIMAL1 -10

解説 「:TRIGger[:ATRigger]:PSI5:DATA:CONDition」が、「EQUAL」、「GREater」、「INSide」、「LESS」、「NOTEQual」、「OUTSide」のときに有効です。

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:DATA:DPATTERN?

機能 PSI5 信号のデータ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{[:ATRigger]|:BTRigger}:
PSI5:DATA:DPATTERN?

5.34 TRIGger グループ

**:TRIGger{[:ATRigger]|:BTRigger}:PSI5
:DATA:DPATtern:HEXa**

機能 PSI5 信号のデータトリガのデータを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:DATA:DPATtern:HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:PSI5:DATA:
DPATtern:HEXa "3A0F"

解説 「:TRIGger[:ATRigger]:PSI5:DATA:CONDi
tion」が「TRUE」、「FALSE」のときに有効です。

**:TRIGger{[:ATRigger]|:BTRigger}:PSI5
:DATA:DPATtern:PATtern**

機能 PSI5 信号のデータトリガのデータを 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:DATA:DPATtern:PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:DATA:DPATtern:PATtern?

例 :TRIGGER:ATRIGGER:PSI5:DATA:
DPATtern:PATtern "0011010100001111"
:TRIGGER:ATRIGGER:PSI5:DATA:
DPATtern:PATtern?

-> :TRIGGER:ATRIGGER:PSI5:DATA:
DPATtern:PATtern "0011010100001111"
解説 「:TRIGger[:ATRigger]:PSI5:DATA:CONDi
tion」が「TRUE」、「FALSE」のときに有効です。

**:TRIGger{[:ATRigger]|:BTRigger}:PSI5
:DATA:FRAMeinslot?**

機能 PSI5 信号のデータトリガのスロット指定に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:DATA:FRAMeinslot?

**:TRIGger{[:ATRigger]|:BTRigger}:PSI5
:DATA:FRAMeinslot:MODE**

機能 PSI5 信号のデータトリガのスロット指定条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:DATA:FRAMeinslot:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:DATA:FRAMeinslot:MODE?

例 :TRIGGER:ATRIGGER:PSI5:DATA:
FRAMEINSLOT:MODE ON
:TRIGGER:ATRIGGER:PSI5:DATA:
FRAMEINSLOT:MODE?
-> :TRIGGER:ATRIGGER:PSI5:DATA:
FRAMEINSLOT:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:PSI5
:DATA:FRAMeinslot:SNUMber**

機能 PSI5 信号のデータトリガのスロット番号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:DATA:FRAMeinslot:
SNUMber {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:DATA:FRAMeinslot:SNUMber?
<Nrf> = 1 ~ 6

例 :TRIGGER:ATRIGGER:PSI5:DATA:
FRAMEINSLOT:SNUMBER 1
:TRIGGER:ATRIGGER:PSI5:DATA:
FRAMEINSLOT:SNUMBER?
-> :TRIGGER:ATRIGGER:PSI5:DATA:
FRAMEINSLOT:SNUMBER 1

**:TRIGger{[:ATRigger]|:BTRigger}:PSI5
:ERROR?**

機能 PSI5 信号の Error 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{[:ATRigger]|:BTRigger}:
PSI5:ERROR?

**:TRIGger{[:ATRigger]|:BTRigger}:PSI5
:ERROR:FNUMber**

機能 PSI5 信号の Frame Number Error トリガを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:ERROR:FNUMber {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:ERROR:FNUMber?

例 :TRIGGER:ATRIGGER:PSI5:ERROR:
FNUMBER ON
:TRIGGER:ATRIGGER:PSI5:ERROR:
FNUMBER?
-> :TRIGGER:ATRIGGER:PSI5:ERROR:
FNUMBER 1

**:TRIGger{[:ATRigger]|:BTRigger}:PSI5
:ERROR:FRAMe**

機能 PSI5 信号の Frame Error トリガを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:ERROR:FRAMe {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:ERROR:FRAMe?

例 :TRIGGER:ATRIGGER:PSI5:ERROR:
FRAME ON
:TRIGGER:ATRIGGER:PSI5:ERROR:FRAME?
-> :TRIGGER:ATRIGGER:PSI5:ERROR:
FRAME 1

**:TRIGger{[:ATRigger]|:BTRigger}:PSI5
:ERROR:MANChester**

機能 PSI5 信号の Manchester Error トリガを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:ERror:MANChester {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:ERror:MANChester?

例 :TRIGGER:ATRIGGER:PSI5:ERROR:
MANCHESTER ON
:TRIGGER:ATRIGGER:PSI5:ERROR:
MANCHESTER?
-> :TRIGGER:ATRIGGER:PSI5:ERROR:
MANCHESTER 1

**:TRIGger{[:ATRigger]|:BTRigger}:PSI5
:ERROR:PCRC**

機能 PSI5 信号の Parity/CRC Error トリガを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:ERror:PCRC {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:ERror:PCRC?

例 :TRIGGER:ATRIGGER:PSI5:ERROR:PCRC ON
:TRIGGER:ATRIGGER:PSI5:ERROR:PCRC?
-> :TRIGGER:ATRIGGER:PSI5:ERROR:
PCRC 1

**:TRIGger{[:ATRigger]|:BTRigger}:PSI5
:ERROR:SBIT**

機能 PSI5 信号の Start Bit Error トリガを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:ERror:SBIT {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:ERror:SBIT?

例 :TRIGGER:ATRIGGER:PSI5:ERROR:SBIT ON
:TRIGGER:ATRIGGER:PSI5:ERROR:SBIT?
-> :TRIGGER:ATRIGGER:PSI5:ERROR:
SBIT 1

**:TRIGger{[:ATRigger]|:BTRigger}:PSI5
:ERROR:SBOundary**

機能 PSI5 信号の Slot Boundary Error トリガを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:ERror:SBOundary {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:ERror:SBOundary?

例 :TRIGGER:ATRIGGER:PSI5:ERROR:
SBOUNDARY ON
:TRIGGER:ATRIGGER:PSI5:ERROR:
SBOUNDARY?
-> :TRIGGER:ATRIGGER:PSI5:ERROR:
SBOUNDARY 1

**:TRIGger{[:ATRigger]|:BTRigger}:PSI5
:FRAMeinslot?**

機能 PSI5 信号のスロット指定トリガのすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:FRAMeinslot?

**:TRIGger{[:ATRigger]|:BTRigger}:PSI5
:FRAMeinslot:SNUMber**

機能 PSI5 信号のスロット指定トリガのスロット番号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:FRAMeinslot:SNUMber {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:FRAMeinslot:SNUMber?

例 :TRIGGER:ATRIGGER:PSI5:FRAMEINSLOT:
SNUMBER 1
:TRIGGER:ATRIGGER:PSI5:FRAMEINSLOT:
SNUMBER?
-> :TRIGGER:ATRIGGER:PSI5:
FRAMEINSLOT:SNUMBER 1

**:TRIGger{[:ATRigger]|:BTRigger}:PSI5
:MODE**

機能 PSI5 信号トリガの種類を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:MODE {DATA|ERROR|FRAMeinslot|
SBIT|SYNC}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:MODE?

例 :TRIGGER:ATRIGGER:PSI5:MODE DATA
:TRIGGER:ATRIGGER:PSI5:MODE?
-> :TRIGGER:ATRIGGER:PSI5:MODE DATA

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup?

機能 PSI5 信号トリガのバス設定に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup?

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup:DATA?

機能 PSI5 信号トリガのデータ信号に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA?

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup:DATA:BRAt

機能 PSI5 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{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:BRAt {<NRf>|USER,
<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:BRAt?

例 <NRf> = 125000、189000
USER の <NRf> = 10000 ~ 1000000
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
BRAt 125000
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
BRAt?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
DATA:BRAt 125000

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup:DATA:CTOLerance

機能 PSI5 信号トリガのクロック許容範囲を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:CTOLerance {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:CTOLerance?

例 <NRf> = ± 0.5% ~ ± 33.3%(設定分解能 0.1%)
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
CTOLERANCE 5
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
CTOLERANCE?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
DATA:CTOLERANCE 5.00E+00

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup:DATA:DBITs

機能 PSI5 信号トリガのデータ長を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:DBITs <NRf>
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:DBITs?
<NRf> = 10、16

例 :TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
DBITs 10
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
DBITs?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
DATA:DBITs 10

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup:DATA:EDETection

機能 PSI5 信号トリガのエラー検出方式を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:EDETection {CRC|
PARity}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:EDETection?

例 :TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
EDETECTION CRC
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
EDETECTION?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
DATA:EDETECTION CRC

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup:DATA:SNRejection?

機能 PSI5 信号トリガの同期ノイズ除去に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:SNRejection?

:TRIGger{[:ATRigger]|:BTRigger}:PSI5 :SETup:DATA:SNRejection:ETIME

機能 PSI5 信号トリガの同期ノイズ除去の終了時間を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:SNRejection:
ETIME {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:SNRejection:ETIME?
<Nrf> = 0 ~ 0.02s(0.1us ステップ)

例 :TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
SNREJECTION:ETIME 66.0000E-06
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
SNREJECTION:ETIME?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
DATA:SNREJECTION:ETIME 66.0000E-06

:TRIGger{[:ATRigger]|:BTRigger}:PSI5 :SETup:DATA:SNRejection:MODE

機能 PSI5 信号トリガの同期ノイズ除去の ON/OFF を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:SNRejection:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:SNRejection:MODE?

例 :TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
SNREJECTION:MODE ON
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
SNREJECTION:MODE?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
DATA:SNREJECTION:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:PSI5 :SETup:DATA:SOURce

機能 PSI5 信号トリガのデータソースを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:SOURce {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:SOURce?
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
SOURCE 1
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
SOURCE?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
DATA:SOURCE 1

:TRIGger{[:ATRigger]|:BTRigger}:PSI5 :SETup:NUMBerofslot

機能 PSI5 信号トリガのスロット数を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:NUMBerofslot {AUTO|<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:NUMBerofslot?
<Nrf> = 1 ~ 6

例 :TRIGGER:ATRIGGER:PSI5:SETUP:
NUMBEROFSLOT AUTO
:TRIGGER:ATRIGGER:PSI5:SETUP:
NUMBEROFSLOT?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
NUMBEROFSLOT AUTO

:TRIGger{[:ATRigger]|:BTRigger}:PSI5 :SETup:SLOT<x>

機能 PSI5 信号トリガのスロットの開始 / 終了時間を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:SLOT<x> {<Nrf>[,<Nrf>]}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:SLOT<x>?
<x> = 1 ~ 6(スロット番号)
<Nrf> = 0 ~ 0.02s(0.1us ステップ)

例 :TRIGGER:ATRIGGER:PSI5:SETUP:
SLOT1 44.0000E-06,181.300E-06
:TRIGGER:ATRIGGER:PSI5:SETUP:SLOT1?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
SLOT1 44.0000E-06,181.300E-06

解説 最初の <Nrf> にスロットの開始時間、2 目目の <Nrf> が終了時間です。
「:TRIGger[:ATRigger]:PSI5:SETup:NUMBerofslot」が 1 ~ 6 の場合は、最後のスロット番号が <x> で指定されたときに、2 目目の <Nrf> が有効になります。
「:TRIGger[:ATRigger]:PSI5:SETup:NUMBerofslot」が「AUTO」の場合は、問い合わせできません。

:TRIGger{[:ATRigger]|:BTRigger}:PSI5 :SETup:SYNC

機能 PSI5 信号トリガの同期信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:SYNC {<Nrf>|NONE}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:SYNC?
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :TRIGGER:ATRIGGER:PSI5:SETUP:SYNC 1
:TRIGGER:ATRIGGER:PSI5:SETUP:SYNC?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
SYNC 1

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:SENT

機能 SENT 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{[:ATRigger]|:BTRigger}:
SENT?

:TRIGger{[:ATRigger]|:BTRigger}:SENT **:CEFactor?**

機能 SENT 信号トリガのエラー要因に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:CEFactor?

:TRIGger{[:ATRigger]|:BTRigger}:SENT **:CEFactor:SAComm?**

機能 SENT 信号トリガの Status and Communication のエラー要因に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:CEFactor:SAComm?

:TRIGger{[:ATRigger]|:BTRigger}:SENT **:CEFactor:SAComm:BIT<x>**

機能 SENT 信号トリガの Status and Communication のエラー要因のビットを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:CEFactor:SAComm:
BIT<x> {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:CEFactor:SAComm:BIT<x>?
<x> = 0、1

例 :TRIGGER:ATRIGGER:SENT:CEFACTOR:
SACOMM:BIT1 ON
:TRIGGER:ATRIGGER:SENT:CEFACTOR:
SACOMM:BIT1?
-> :TRIGGER:ATRIGGER:SENT:CEFACTOR:
SCPULSES 1

:TRIGger{[:ATRigger]|:BTRigger}:SENT **:CEFactor:SCPulses**

機能 SENT 信号トリガの連続キャリブレーションパルスのエラー要因を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:CEFactor:SCPulses {OFF|OPT2|
POPTION}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:CEFactor:SCPulses?

例 :TRIGGER:ATRIGGER:SENT:CEFACTOR:
SCPULSES OFF
:TRIGGER:ATRIGGER:SENT:CEFACTOR:
SCPULSES?
-> :TRIGGER:ATRIGGER:SENT:CEFACTOR:
SCPULSES OFF

:TRIGger{[:ATRigger]|:BTRigger}:SENT **:CRCType**

機能 SENT 信号トリガの CRC 演算方式を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:CRCType {LEGacy|RECommended}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:CRCType?

例 :TRIGGER:ATRIGGER:SENT:
CRCTYPE LEGACY
:TRIGGER:ATRIGGER:SENT:CRCTYPE?
-> :TRIGGER:ATRIGGER:SENT:
CRCTYPE LEGACY

:TRIGger{[:ATRigger]|:BTRigger}:SENT **:CTICK**

機能 SENT 信号トリガのクロックティック値を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:CTICK {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:CTICK?
<Nrf> = 1.00us ~ 100.00us(設定分解能 0.01us)

例 :TRIGGER:ATRIGGER:SENT:
CTICK 0.000001
:TRIGGER:ATRIGGER:SENT:CTICK?
-> :TRIGGER:ATRIGGER:SENT:
CTICK 1.000000E-06

:TRIGger{[:ATRigger]|:BTRigger}:SENT:CTOLerance

機能 SENT 信号トリガのクロック許容差を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:CTOLerance {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:CTOLerance?
<NRf> = ± 1.0% ~ ± 30.0% (設定分解能 0.1%)

例 :TRIGGER:ATRIGGER:SENT:
CTOLERANCE 25.0
:TRIGGER:ATRIGGER:SENT:CTOLERANCE?
-> :TRIGGER:ATRIGGER:SENT:
CTOLERANCE 25.0E+00

:TRIGger{[:ATRigger]|:BTRigger}:SENT:DNIBbles

機能 SENT 信号トリガのデータニブル数を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:DNIBbles {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:DNIBbles?
<NRf> = 1 ~ 6

例 :TRIGGER:ATRIGGER:SENT:DNIBBLES 6
:TRIGGER:ATRIGGER:SENT:DNIBBLES?
-> :TRIGGER:ATRIGGER:SENT:DNIBBLES 6

:TRIGger{[:ATRigger]|:BTRigger}:SENT:DTYPe

機能 SENT 信号トリガの Fast CH のデータ形式を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:DTYPe {NIBble|USER}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:DTYPe?

例 :TRIGGER:ATRIGGER:SENT:DTYPE NIBBLE
:TRIGGER:ATRIGGER:SENT:DTYPE?
-> :TRIGGER:ATRIGGER:SENT:
DTYPE NIBBLE

:TRIGger{[:ATRigger]|:BTRigger}:SENT:ERRor?

機能 SENT 信号トリガの Error に関するすべての設定値を問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:ERRor?

:TRIGger{[:ATRigger]|:BTRigger}:SENT:ERRor:FCRC

機能 SENT 信号トリガの Fast CH の CRC エラーを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:ERRor:FCRC {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:ERRor:FCRC?

例 :TRIGGER:ATRIGGER:SENT:ERROR:FCRC ON
:TRIGGER:ATRIGGER:SENT:ERROR:FCRC?
-> :TRIGGER:ATRIGGER:SENT:ERROR:
FCRC 1

:TRIGger{[:ATRigger]|:BTRigger}:SENT:ERRor:NDValue

機能 SENT 信号トリガのニブルデータバリュエラーを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:ERRor:NDValue {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:ERRor:NDValue?

例 :TRIGGER:ATRIGGER:SENT:ERROR:
NDVALUE ON
:TRIGGER:ATRIGGER:SENT:ERROR:
NDVALUE?
-> :TRIGGER:ATRIGGER:SENT:ERROR:
NDVALUE 1

:TRIGger{[:ATRigger]|:BTRigger}:SENT:ERRor:NNUmber

機能 SENT 信号トリガのニブルデータ数エラーを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:ERRor:NNUmber {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:ERRor:NNUmber?

例 :TRIGGER:ATRIGGER:SENT:ERROR:
NNUmber ON
:TRIGGER:ATRIGGER:SENT:ERROR:
NNUmber?
-> :TRIGGER:ATRIGGER:SENT:ERROR:
NNUmber 1

5.34 TRIGger グループ

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:ERROr:SAComm**

機能 SENT 信号トリガの Status and Communication
エラーを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:ERROr:SAComm {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:ERROr:SAComm?

例 :TRIGGER:ATRIGGER:SENT:ERROR:
SACOMM ON
:TRIGGER:ATRIGGER:SENT:ERROR:SACOMM?
-> :TRIGGER:ATRIGGER:SENT:ERROR:
SACOMM 1

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:ERROr:SCPulses**

機能 SENT 信号トリガの連続キャリブレーションパル
スエラーを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:ERROr:SCPulses {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:ERROr:SCPulses?

例 :TRIGGER:ATRIGGER:SENT:ERROR:
SCPULSES ON
:TRIGger:ATRIGGER:SENT:ERROR:
SCPULSES?
-> :TRIGGER:ATRIGGER:SENT:ERROR:
SCPULSES 1

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:ERROr:SCRC**

機能 SENT 信号トリガの Slow CH の CRC エラーを設
定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:ERROr:SCRC {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:ERROr:SCRC?

例 :TRIGGER:ATRIGGER:SENT:ERROR:SCRC ON
:TRIGGER:ATRIGGER:SENT:ERROR:SCRC?
-> :TRIGGER:ATRIGGER:SENT:ERROR:
SCRC 1

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:FDATa?**

機能 SENT 信号トリガの Fast CH 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{[:ATRigger]|:BTRigger}:
SENT:FDATa?

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:FDATa:DATA<x>?**

機能 SENT 信号トリガの Fast CH の各ユーザーデー
タに関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:FDATa:DATA<x>?
<x> = 1 ~ 4

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:FDATa:DATA<x>:CONDition**

機能 SENT 信号トリガの Fast CH の各ユーザーデー
タの判定条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:FDATa:DATA<x>:
CONDition {BETWEE|EQUAL|GREATER|
LESS|NOTBETWEEN|NOTEQUAL}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:FDATa:DATA<x>:CONDition?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:SENT:FDATA:DATA1:
CONDITION BETWEEN
:TRIGGER:ATRIGGER:SENT:FDATA:DATA1:
CONDITION?
-> :TRIGGER:ATRIGGER:SENT:FDATA:
DATA1:CONDITION BETWEEN

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:FDATa:DATA<x>:DECimal<y>**

機能 SENT 信号トリガの Fast CH の各ユーザーデー
タのデータを 10 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:FDATa:DATA<x>:
DECimal<y> {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:FDATa:DATA<x>:DECimal<y>?
<x> = 1 ~ 4
<y> = 1、2

<NRf> = 本体ユーザーマニュアル機能編参照。

例 :TRIGGER:ATRIGGER:SENT:FDATA:DATA1:
DECIMAL1 1
:TRIGGER:ATRIGGER:SENT:FDATA:DATA1:
DECIMAL1?
-> :TRIGGER:ATRIGGER:SENT:FDATA:
DATA1:DECIMAL1 1

:TRIGger{[:ATRigger]|:BTRigger}:SENT:FDATa:DATA<x>:MODE

機能 SENT 信号トリガの Fast CH の各ユーザーデータのデータ条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:FDATa:DATA<x>:MODE{<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:FDATa:DATA<x>:MODE?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:SENT:FDATa:DATA1:
MODE ON
:TRIGGER:ATRIGGER:SENT:FDATa:DATA1:
MODE?
-> :TRIGGER:ATRIGGER:SENT:FDATa:
DATA1:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:SENT:FDATa:DNIBbles?

機能 SENT 信号トリガの Fast CH のニブル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{[:ATRigger]|:BTRigger}:
SENT:FDATa:DNIBbles?

:TRIGger{[:ATRigger]|:BTRigger}:SENT:FDATa:DNIBbles:CONDition

機能 SENT 信号トリガの Fast CH のニブルデータの判定条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:FDATa:DNIBbles:
CONDition {FALSE|TRUE}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:FDATa:DNIBbles:CONDition?

例 :TRIGGER:ATRIGGER:SENT:FDATa:
DNIBBLES:CONDITION TRUE
:TRIGGER:ATRIGGER:SENT:FDATa:
DNIBBLES:CONDITION?
-> :TRIGGER:ATRIGGER:SENT:FDATa:
DNIBBLES:CONDITION TRUE

:TRIGger{[:ATRigger]|:BTRigger}:SENT:FDATa:DNIBbles:HEXa

機能 SENT 信号トリガの Fast CH のニブルデータのデータを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:FDATa:DNIBbles:HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:SENT:FDATa:
DNIBBLES:HEXa "112233"

:TRIGger{[:ATRigger]|:BTRigger}:SENT:FDATa:DNIBbles:PATtern

機能 SENT 信号トリガの Fast CH のニブルデータのデータを 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:FDATa:DNIBbles:PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:FDATa:DNIBbles:PATtern?

例 :TRIGGER:ATRIGGER:SENT:FDATa:
DNIBBLES:
PATTERN "111100001111000011110000"
:TRIGGER:ATRIGGER:SENT:FDATa:
DNIBBLES:PATTERN?
-> :TRIGGER:ATRIGGER:SENT:FDATa:
DNIBBLES:
PATTERN "111100001111000011110000"

:TRIGger{[:ATRigger]|:BTRigger}:SENT:FSAComm?

機能 SENT 信号トリガの Fast CH の Status and Communication ニブルに関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:FSAComm?

:TRIGger{[:ATRigger]|:BTRigger}:SENT:FSAComm:HEXa

機能 SENT 信号トリガの Fast CH の Status and Communication ニブルのデータを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:FSAComm:HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:SENT:FSACOMM:
HEXa "F"

:TRIGger{[:ATRigger]|:BTRigger}:SENT:FSAComm:PATtern

機能 SENT 信号トリガの Fast CH の Status and Communication ニブルのデータを 2 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:FSAComm:PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:FSAComm:PATtern?

例 :TRIGGER:ATRIGGER:SENT:FSACOMM:
PATTERN "1111"
:TRIGGER:ATRIGGER:SENT:FSACOMM:
PATTERN?
-> :TRIGGER:ATRIGGER:SENT:FSACOMM:
PATTERN "1111"

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:SENT:MODE

機能 SENT 信号トリガのモードを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:MODE {EFASt|ERRor|ESLow|FDATa|
FSAComm|SDATa}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:MODE?

例 :TRIGGER:ATRIGGER:SENT:MODE ERRor
:TRIGGER:ATRIGGER:SENT:MODE?
-> :TRIGGER:ATRIGGER:SENT:MODE ERRor

:TRIGger{[:ATRigger]|:BTRigger}:SENT:PPULse

機能 SENT 信号トリガのポーズパルスの有無を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:PPULse {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:PPULse?

例 :TRIGGER:ATRIGGER:SENT:PPULSE ON
:TRIGGER:ATRIGGER:SENT:PPULSE?
-> :TRIGGER:ATRIGGER:SENT:PPULSE 1

:TRIGger{[:ATRigger]|:BTRigger}:SENT:SDATa?

機能 SENT 信号トリガの Slow CH 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{[:ATRigger]|:BTRigger}:
SENT:SDATa?

:TRIGger{[:ATRigger]|:BTRigger}:SENT:SDATa:ENHanced?

機能 SENT 信号トリガの Slow CH の Enhanced タイプに関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDATa:ENHanced?

:TRIGger{[:ATRigger]|:BTRigger}:SENT:SDATa:ENHanced:CBIT

機能 SENT 信号トリガの Slow CH の Enhanced タイプの Configuration ビットを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDATa:ENHanced:CBIT {D12Bit|
D16Bit}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDATa:ENHanced:CBIT?

例 :TRIGGER:ATRIGGER:SENT:SDATa:
ENHANCED:CBIT D12Bit
:TRIGger:ATRIGGER:SENT:SDATa:
ENHANCED:CBIT?
-> :TRIGGER:ATRIGGER:SENT:SDATa:
ENHANCED:CBIT D12Bit

:TRIGger{[:ATRigger]|:BTRigger}:SENT:SDATa:ENHanced:D12Bit?

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDATa:ENHanced:D12Bit?

:TRIGger{[:ATRigger]|:BTRigger}:SENT:SDATa:ENHanced:D12Bit:DATA?

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID 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{[:ATRigger]|:BTRigger}:
SENT:SDATa:ENHanced:D12Bit:DATA?

:TRIGger{[:ATRigger]|:BTRigger}:SENT:SDATa:ENHanced:D12Bit:DATA:CONDition

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータの判定条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDATa:ENHanced:D12Bit:DATA:
CONDition {BETWeen|EQual|FALSe|
GREater|LESS|NOTBetween|NOTEqual|
TRUE}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDATa:ENHanced:D12Bit:DATA:
CONDition?

例 :TRIGGER:ATRIGGER:SENT:SDATa:
ENHANCED:D12BIT:DATA:CONDITION FALSE
:TRIGger:ATRIGGER:SENT:SDATa:
ENHANCED:D12BIT:DATA:CONDITION?
-> :TRIGGER:ATRIGGER:SENT:SDATa:
ENHANCED:D12BIT:DATA:CONDITION FALSE

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:ENHanced:D12Bit:DATA:DECimal
<x>**

機能 SENT 信号トリガの Slow CH の Enhanced タイプ
の 12 ビットデータ、8 ビット ID のデータを 10
進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:SDaTa:ENHanced:D12Bit:DATA:
DECimal<x> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger):
SENT:SDaTa:ENHanced:D12Bit:DATA:
DECimal<x>?

<x> = 1、2
<Nrf> = 本体ユーザズマニュアル機能編参照。

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:DECIMAL1 123
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:DECIMAL1?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:DECIMAL1 123

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:ENHanced:D12Bit:DATA:HEXa**

機能 SENT 信号トリガの Slow CH の Enhanced タイプ
の 12 ビットデータ、8 ビット ID のデータを 16
進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:SDaTa:ENHanced:D12Bit:DATA:
HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:HEXA "123"

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:ENHanced:D12Bit:DATA:MODE**

機能 SENT 信号トリガの Slow CH の Enhanced タイプ
の 12 ビットデータ、8 ビット ID のデータ条件 (有
効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:SDaTa:ENHanced:D12Bit:DATA:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger):
SENT:SDaTa:ENHanced:D12Bit:DATA:
MODE?

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:MODE ON
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:MODE?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:ENHanced:D12Bit:DATA:PATtern**

機能 SENT 信号トリガの Slow CH の Enhanced タイプ
の 12 ビットデータ、8 ビット ID のデータを 2 進
数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:SDaTa:ENHanced:D12Bit:DATA:
PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger):
SENT:SDaTa:ENHanced:D12Bit:DATA:
PATtern?

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:
PATTERN "111101010101"
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:PATTERN?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:
PATTERN "111101010101"

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:ENHanced:D12Bit:ID?**

機能 SENT 信号トリガの Slow CH の Enhanced タイプ
の 12 ビットデータ、8 ビット ID の ID に関する
すべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:SDaTa:ENHanced:D12Bit:ID?

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:ENHanced:D12Bit:ID:CONDition**

機能 SENT 信号トリガの Slow CH の Enhanced タイプ
の 12 ビットデータ、8 ビット ID の ID 判定条件
を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:SDaTa:ENHanced:D12Bit:ID:
CONDition {BETween|EQual|GREater|
LESS|NOTBetween|NOTEqual}
:TRIGger{[:ATRigger]|:BTRigger):
SENT:SDaTa:ENHanced:D12Bit:ID:
CONDition?

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:ID:CONDITION BETWEEN
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:ID:CONDITION?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:ID:CONDITION BETWEEN

5.34 TRIGger グループ

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDATA:ENHanced:D12Bit:ID:DECimal<x>**

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID の ID を 10 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDATA:ENHanced:D12Bit:ID:
DECimal<x> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDATA:ENHanced:D12Bit:ID:
DECimal<x>?
<x> = 1、2
<Nrf> = 本体ユーザーズマニュアル機能編参照。

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:ID:DECIMAL1 0
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:ID:DECIMAL1?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:ID:DECIMAL1 0

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDATA:ENHanced:D12Bit:ID:MODE**

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID の ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDATA:ENHanced:D12Bit:ID:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDATA:ENHanced:D12Bit:ID:MODE?

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:ID:MODE ON
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:ID:MODE?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:ID:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDATA:ENHanced:D12Bit:PFORMAT**

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 12 ビットデータ、8 ビット ID のデータの入力形式を設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDATA:ENHanced:D12Bit:
PFORMAT {DECimal|HEXa}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDATA:ENHanced:D12Bit:PFORMAT?

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:PFORMAT HEXA
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:PFORMAT?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:PFORMAT HEXA

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDATA:ENHanced:D16Bit?**

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDATA:ENHanced:D16Bit?

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDATA:ENHanced:D16Bit:DATA?**

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID 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{[:ATRigger]|:BTRigger}:
SENT:SDATA:ENHanced:D16Bit:DATA?

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDATA:ENHanced:D16Bit:DATA:CONDition**

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータの判定条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDATA:ENHanced:D16Bit:DATA:
CONDition {BETween|EQual|FALSe|
GREater|LESS|NOTBetween|NOTEqual|
TRUE}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDATA:ENHanced:D16Bit:DATA:
CONDition?

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:CONDITION FALSE
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:CONDITION?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:CONDITION FALSE

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:ENHanced:D16Bit:DATA:DECimal**

<x>

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータを 10 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
DECimal<x> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
DECimal<x>?
<x> = 1, 2
<Nrf> = 本体ユーザズマニュアル機能編参照。

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:DECIMAL 123
:TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:DECIMAL?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:DECIMAL 123

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:ENHanced:D16Bit:DATA:HEXa**

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:HEXA "0000"

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:ENHanced:D16Bit:DATA:MODE**

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータ条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
MODE?

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:MODE ON
:TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:MODE?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:ENHanced:D16Bit:DATA:PATtern**

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータを 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
PATtern?

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:
PATTERN "1111000011110000"
:TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:PATTERN?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:
PATTERN "1111000011110000"

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:ENHanced:D16Bit:ID?**

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:ID?

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:ENHanced:D16Bit:ID:CONDition**

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID 判定条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:ID:
CONDition {BETween|EQual|GREater|
LESS|NOTBetween|NOTEqual}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:ID:
CONDition?

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:ID:CONDITION BETWEEN
:TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:ID:CONDITION?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:ID:CONDITION BETWEEN

5.34 TRIGger グループ

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDATA:ENHanced:D16Bit:ID:DECimal<x>**

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID を 10 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:SDATA:ENHanced:D16Bit:ID:
DECimal<x> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger):
SENT:SDATA:ENHanced:D16Bit:ID:
DECimal<x>?
<x> = 1, 2
<Nrf> = 本体ユーザーマニュアル機能編参照。

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:ID:DECIMAL1 0
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:ID:DECIMAL1?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:ID:DECIMAL1 0

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDATA:ENHanced:D16Bit:ID:MODE**

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID の ID 条件 (有効 / 無効) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:SDATA:ENHanced:D16Bit:ID:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger):
SENT:SDATA:ENHanced:D16Bit:ID:MODE?

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:ID:MODE ON
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:ID:MODE?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:ID:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDATA:ENHanced:D16Bit:PFORMAT**

機能 SENT 信号トリガの Slow CH の Enhanced タイプの 16 ビットデータ、4 ビット ID のデータの入力形式を設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:SDATA:ENHanced:D16Bit:
PFORMAT {DECimal|HEXa}
:TRIGger{[:ATRigger]|:BTRigger):
SENT:SDATA:ENHanced:D16Bit:PFORMAT?

例 :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:PFORMAT HEXA
:TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:PFORMAT?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:PFORMAT HEXA

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDATA:SHORT?**

機能 SENT 信号トリガの Slow CH の Short タイプに関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:SDATA:SHORT?

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDATA:SHORT:DATA?**

機能 SENT 信号トリガの Slow CH の Short タイプ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{[:ATRigger]|:BTRigger):
SENT:SDATA:SHORT:DATA?

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDATA:SHORT:DATA:CONDition**

機能 SENT 信号トリガの Slow CH の Short タイプのデータの判定条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:SDATA:SHORT:DATA:
CONDition {BETween|EQual|FALSe|
GREater|LESS|NOTBetween|NOTEqual|
TRUE}
:TRIGger{[:ATRigger]|:BTRigger):
SENT:SDATA:SHORT:DATA:CONDition?

例 :TRIGGER:ATRIGGER:SENT:SDATA:SHORT:
DATA:CONDITION FALSE
:TRIGger:ATRIGGER:SENT:SDATA:SHORT:
DATA:CONDITION?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
SHORT:DATA:CONDITION FALSE

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDATA:SHORT:DATA:DECimal<x>**

機能 SENT 信号トリガの Slow CH の Short タイプのデータを 10 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:SDATA:SHORT:DATA:
DECimal<x> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger):
SENT:SDATA:SHORT:DATA:DECimal<x>?
<x> = 1, 2
<Nrf> = 本体ユーザーマニュアル機能編参照。

例 :TRIGGER:ATRIGGER:SENT:SDATA:SHORT:
DATA:DECIMAL1 123
:TRIGGER:ATRIGGER:SENT:SDATA:SHORT:
DATA:DECIMAL1?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
SHORT:DATA:DECIMAL1 123

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:SHORt:DAta:HEXa**

機能 SENT信号トリガのSlow CHのShortタイプのデータを16進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORt:DAta:HEXa {<文字列>}

例 :TRIGGER:ATRIGGER:SENT:SDaTa:SHORt:
DAta:HEXa "00"

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:SHORt:DAta:MODE**

機能 SENT信号トリガのSlow CHのShortタイプのデータ条件(有効/無効)を設定/問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORt:DAta:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORt:DAta:MODE?

例 :TRIGGER:ATRIGGER:SENT:SDaTa:SHORt:
DAta:MODE ON
:TRIGger:ATRIGGER:SENT:SDaTa:SHORt:
DAta:MODE?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
SHORt:DAta:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:SHORt:DAta:PATtern**

機能 SENT信号トリガのSlow CHのShortタイプのデータを2進数で設定/問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORt:DAta:
PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORt:DAta:PATtern?

例 :TRIGGER:ATRIGGER:SENT:SDaTa:SHORt:
DAta:PATtern "11110000"
:TRIGger:ATRIGGER:SENT:SDaTa:SHORt:
DAta:PATtern?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
SHORt:DAta:PATtern "11110000"

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:SHORt:ID?**

機能 SENT信号トリガのSlow CHのShortタイプのIDに関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORt:ID?

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:SHORt:ID:CONDition**

機能 SENT信号トリガのSlow CHのShortタイプのID判定条件を設定/問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORt:ID:
CONDition {BETween|EQUal|GREater|
LESS|NOTBetween|NOTEQUAL}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORt:ID:CONDition?

例 :TRIGGER:ATRIGGER:SENT:SDaTa:SHORt:
ID:CONDition BETWEEN
:TRIGger:ATRIGGER:SENT:SDaTa:SHORt:
ID:CONDition?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
SHORt:ID:CONDition BETWEEN

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:SHORt:ID:DECimal<x>**

機能 SENT信号トリガのSlow CHのShortタイプのIDを10進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORt:ID:
DECimal<x> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORt:ID:DECimal<x>?
<x> = 1、2

例 :TRIGGER:ATRIGGER:SENT:SDaTa:SHORt:
ID:DECimal1 0
:TRIGger:ATRIGGER:SENT:SDaTa:SHORt:
ID:DECIMAL1?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
SHORt:ID:DECIMAL1 0

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:SHORt:ID:MODE**

機能 SENT信号トリガのSlow CHのShortタイプのID条件(有効/無効)を設定/問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORt:ID:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORt:ID:MODE?

例 :TRIGGER:ATRIGGER:SENT:SDaTa:SHORt:
ID:MODE ON
:TRIGger:ATRIGGER:SENT:SDaTa:SHORt:
ID:MODE?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
SHORt:ID:MODE 1

5.34 TRIGger グループ

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SDaTa:SHORt:PFORmat**

機能 SENT 信号トリガの Slow CH の Short タイプのデータの入力形式を設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:SDaTa:SHORt:PFORmat {DECimal|
HEXa}
:TRIGger{[:ATRigger]|:BTRigger):
SENT:SDaTa:SHORt:PFORmat?

例 :TRIGGER:ATRIGGER:SENT:SDaTa:SHORt:
PFORMAT HEXA
:TRIGGER:ATRIGGER:SENT:SDaTa:SHORt:
PFORMAT?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
SHORt:PFORMAT HEXA

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:SOURce**

機能 SENT 信号トリガのソースを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:SOURce {<NRF>|PODA<x>|PODB<x>|
PODC<x>|PODD<x>}
:TRIGger{[:ATRigger]|:BTRigger):
SENT:SOURce?

<NRF> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<x>、PODB<x>、PODC<x>、PODD<x>
の <x> = 0 ~ 7
例 :TRIGGER:ATRIGGER:SENT:SOURce 1
:TRIGger:ATRIGGER:SENT:SOURce?
-> :TRIGGER:ATRIGGER:SENT:SOURce 1

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:STYPe**

機能 SENT 信号トリガの Slow CH 形式を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:STYPe {ENHanced|SHORt}
:TRIGger{[:ATRigger]|:BTRigger):
SENT:STYPe?

例 :TRIGGER:ATRIGGER:SENT:
STYPe ENHANCED
:TRIGger:ATRIGGER:SENT:STYPe?
-> :TRIGGER:ATRIGGER:SENT:
STYPe ENHANCED

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:USETup?**

機能 SENT 信号トリガの Fast CH のユーザー定義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{[:ATRigger]|:BTRigger):
SENT:USETup?

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:USETup:DAta<x>?**

機能 SENT 信号トリガの Fast CH の各ユーザー定義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{[:ATRigger]|:BTRigger):
SENT:USETup:DAta<x>?
<x> = 1 ~ 4

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:USETup:DAta<x>:MODE**

機能 SENT 信号トリガの Fast CH の各ユーザー定義データの有効 / 無効を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:USETup:DAta<x>:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger):
SENT:USETup:DAta<x>:MODE?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:SENT:USETUP:DAta1:
MODE ON
:TRIGger:ATRIGGER:SENT:USETUP:DAta1:
MODE?
-> :TRIGGER:ATRIGGER:SENT:USETUP:
DAta1:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:SENT
:USETup:DAta<x>:ORDER**

機能 SENT 信号トリガの Fast CH の各ユーザー定義データのエンディアンを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SENT:USETup:DAta<x>:ORDER {BIG|
LITtle}
:TRIGger{[:ATRigger]|:BTRigger):
SENT:USETup:DAta<x>:ORDER?
<x> = 1 ~ 4

例 :TRIGGER:ATRIGGER:SENT:USETUP:DAta1:
ORDER BIG
:TRIGger:ATRIGGER:SENT:USETUP:DAta1:
ORDER?
-> :TRIGGER:ATRIGGER:SENT:USETUP:
DAta1:ORDER BIG

:TRIGger{[:ATRigger]|:BTRigger}:SENT:USESetup:DATA<x>:SIZE

機能 SENT 信号トリガの Fast CH の各ユーザー定義データのデータサイズを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:USESetup:DATA<x>:SIZE {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:USESetup:DATA<x>:SIZE?
<x> = 1 ~ 4
<Nrf> = 0 ~ 24

例 :TRIGGER:ATRIGGER:SENT:USESetup:DATA1:
SIZE 24
:TRIGger:ATRIGGER:SENT:USESetup:DATA1:
SIZE?
-> :TRIGGER:ATRIGGER:SENT:USESetup:
DATA1:SIZE 24

:TRIGger{[:ATRigger]|:BTRigger}:SENT:USESetup:MULTiplexing

機能 SENT 信号トリガの Fast CH のユーザー定義データで、Multiplexing の有効 / 無効を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:USESetup:MULTiplexing {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:USESetup:MULTiplexing?

例 :TRIGGER:ATRIGGER:SENT:USESetup:
MULTIPLEXING ON
:TRIGger:ATRIGGER:SENT:USESetup:
MULTIPLEXING?
-> :TRIGGER:ATRIGGER:SENT:USESetup:
MULTIPLEXING ON

:TRIGger{[:ATRigger]|:BTRigger}:SENT:VERsion

機能 SENT 信号トリガの仕様バージョンを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SENT:VERsion {APR2016|FEB2008|
JAN2010}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:VERsion?

例 :TRIGGER:ATRIGGER:SENT:
VERSION JAN2010
:TRIGger:ATRIGGER:SENT:VERSION?
-> :TRIGGER:ATRIGGER:SENT:
VERSION JAN2010

:TRIGger{[:ATRigger]|:BTRigger}:SIMPlE?

機能 エッジ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{[:ATRigger]|:BTRigger}:
SIMPlE?

:TRIGger{[:ATRigger]|:BTRigger}:SIMPlE:HFRejection(HighFrequencyREJECTION)

機能 エッジトリガのトリガソースのローパスフィルター (HF リジェクション) の ON/OFF を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SIMPlE:HFRejection {<周波数>|OFF}
:TRIGger{[:ATRigger]|:BTRigger}:
SIMPlE:HFRejection?

例 <周波数> = 15KHz ~ 20MHz
:TRIGGER:ATRIGGER:SIMPlE:
HFREJECTION 15KHZ
:TRIGger:ATRIGGER:SIMPlE:
HFREJECTION?
-> :TRIGGER:ATRIGGER:SIMPlE:
HFREJECTION 15000

解説 「:TRIGger{[:ATRigger]|:BTRigger}:SIMPlE:SOURce」が、「EXTERNAL」、「LINE」のときは、設定 / 問い合わせできません。

:TRIGger{[:ATRigger]|:BTRigger}:SIMPlE:LEVel

機能 エッジトリガのトリガソースのトリガレベルを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SIMPlE:LEVel {<電圧>}
:TRIGger{[:ATRigger]|:BTRigger}:
SIMPlE:LEVel?

<電圧> = 本体ユーザーズマニュアル機能編参照。

例 :TRIGGER:ATRIGGER:SIMPlE:LEVel 0V
:TRIGger:ATRIGGER:SIMPlE:LEVel?
-> :TRIGGER:ATRIGGER:SIMPlE:
LEVel 0.000E+00

解説 • 「:TRIGger{[:ATRigger]|:BTRigger}:SIMPlE:SOURce」が、「LINE」のときは、設定 / 問い合わせできません。
• 電流プローブ設定時は、<電流> 値の設定 / 問い合わせとなります。

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:SIMPlе:NREJection

機能	エッジトリガのトリガソースのレベルのノイズリジェクションを設定 / 問い合わせします。
構文	:TRIGger{[:ATRigger] :BTRigger}:SIMPlе:NREJection {HIGH LOW MIDDLE} :TRIGger{[:ATRigger] :BTRigger}:SIMPlе:NREJection?
例	:TRIGGER:ATRIGGER:SIMPLE:NREJECTION HIGH :TRIGGER:ATRIGGER:SIMPLE:NREJECTION? -> :TRIGGER:ATRIGGER:SIMPLE:NREJECTION HIGH
解説	「:TRIGger{[:ATRigger] :BTRigger}:SIMPlе:SOURce」が、「EXTErnal」、「LINE」のときは、設定 / 問い合わせできません。

:TRIGger{[:ATRigger]|:BTRigger}:SIMPlе:PROBe

機能	エッジトリガの外部トリガソースの、プローブを設定 / 問い合わせします。
構文	:TRIGger{[:ATRigger] :BTRigger}:SIMPlе:PROBe {<NRf>} :TRIGger{[:ATRigger] :BTRigger}:SIMPlе:PROBe? <NRf> = 1、10
例	:TRIGGER:ATRIGGER:SIMPLE:PROBE 1 :TRIGGER:ATRIGGER:SIMPLE:PROBE? -> :TRIGGER:ATRIGGER:SIMPLE:PROBE 1
解説	「:TRIGger{[:ATRigger] :BTRigger}:SIMPlе:SOURce」が、「EXTErnal」でない場合は、設定 / 問い合わせできません。

:TRIGger{[:ATRigger]|:BTRigger}:SIMPlе:RANGe

機能	エッジトリガの外部トリガソースの入力レンジを設定 / 問い合わせします。
構文	:TRIGger{[:ATRigger] :BTRigger}:SIMPlе:RANGe {<電圧>} :TRIGger{[:ATRigger] :BTRigger}:SIMPlе:RANGe? <電圧> = 2 ~ 200(V)
例	:TRIGGER:ATRIGGER:SIMPLE:RANGE 2 :TRIGGER:ATRIGGER:SIMPLE:RANGE? -> :TRIGGER:ATRIGGER:SIMPLE:RANGE 2

:TRIGger{[:ATRigger]|:BTRigger}:SIMPlе:SLOPe

機能	エッジトリガのトリガソースのトリガスロープを設定 / 問い合わせします。
構文	:TRIGger{[:ATRigger] :BTRigger}:SIMPlе:SLOPe {BOTH FALL RISE} :TRIGger{[:ATRigger] :BTRigger}:SIMPlе:SLOPe?
例	:TRIGGER:ATRIGGER:SIMPLE:SLOPE RISE :TRIGGER:ATRIGGER:SIMPLE:SLOPE? -> :TRIGGER:ATRIGGER:SIMPLE:SLOPE RISE
解説	「:TRIGger{[:ATRigger] :BTRigger}:SIMPlе:SOURce」が、「LINE」のときは、設定 / 問い合わせできません。

:TRIGger{[:ATRigger]|:BTRigger}:SIMPlе:SOURce

機能	エッジトリガのトリガソースを設定 / 問い合わせします。
構文	:TRIGger{[:ATRigger] :BTRigger}:SIMPlе:SOURce {<NRf> EXTErnal LINE PODA<x> PODB<x> PODC<x> PODD<x>} :TRIGger{[:ATRigger] :BTRigger}:SIMPlе:SOURce? <NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4) PODA<x>、PODB<x>、PODC<x>、PODD<x>の <x> = 0 ~ 7
例	:TRIGGER:ATRIGGER:SIMPLE:SOURCE 1 :TRIGGER:ATRIGGER:SIMPLE:SOURCE? -> :TRIGGER:ATRIGGER:SIMPLE:SOURCE 1

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern?

機能	ユーザー定義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{[:ATRigger] :BTRigger}:SPATtern?

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:BITSize

機能 ユーザー定義バス信号トリガのビット長を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:BITSize {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:BITSize?

<Nrf> = 1 ~ 128

例 :TRIGGER:ATRIGGER:SPATTERN:BITSIZE 1
:TRIGGER:ATRIGGER:SPATTERN:BITSIZE?
-> :TRIGGER:ATRIGGER:SPATTERN:
BITSIZE 1

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:BRATe

機能 ユーザー定義バス信号トリガのビットレートを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:BRATe {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:BRATe?

<Nrf> = 1000 ~ 50000000

例 :TRIGGER:ATRIGGER:SPATTERN:
BRATE 1000
:TRIGGER:ATRIGGER:SPATTERN:BRATE?
-> :TRIGGER:ATRIGGER:SPATTERN:
BRATE 1000

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:CLOCK?

機能 ユーザー定義バス信号トリガのクロック信号に関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CLOCK?

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:CLOCK:MODE

機能 ユーザー定義バス信号トリガのクロック信号を使用する / しなないを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CLOCK:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CLOCK:MODE?

例 :TRIGGER:ATRIGGER:SPATTERN:CLOCK:
MODE ON
:TRIGGER:ATRIGGER:SPATTERN:CLOCK:
MODE?
-> :TRIGGER:ATRIGGER:SPATTERN:CLOCK:
MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:CLOCK:POLarity

機能 ユーザー定義バス信号トリガのクロック信号の極性を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CLOCK:POLarity {FALL|RISE}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CLOCK:POLarity?

例 :TRIGGER:ATRIGGER:SPATTERN:CLOCK:
POLARITY FALL
:TRIGGER:ATRIGGER:SPATTERN:CLOCK:
POLARITY?
-> :TRIGGER:ATRIGGER:SPATTERN:CLOCK:
POLARITY FALL

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:CLOCK:SOURce

機能 ユーザー定義バス信号トリガのクロック信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CLOCK:SOURce {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CLOCK:SOURce?

例 :TRIGGER:ATRIGGER:SPATTERN:CLOCK:
SOURCE 1
:TRIGGER:ATRIGGER:SPATTERN:CLOCK:
SOURCE?
-> :TRIGGER:ATRIGGER:SPATTERN:CLOCK:
SOURCE 1

解説 「:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:DATA:SOURce」の設定によって設定できるクロック信号が異なります。詳しくは本体ユーザーズマニュアル機能編をご覧ください。

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:CS?

機能 ユーザー定義バス信号トリガのチップセレクト信号のすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CS?

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:CS:ACTive

機能 ユーザー定義バス信号トリガのチップセレクト信号のアクティブ状態を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SPATtern:CS:ACTive {HIGH|LOW}
:TRIGger{[:ATRigger]|:BTRigger):
SPATtern:CS:ACTive?

例 :TRIGGER:ATRIGGER:SPATTERN:CS:
ACTIVE HIGH
:TRIGGER:ATRIGGER:SPATTERN:CS:
ACTIVE?
-> :TRIGGER:ATRIGGER:SPATTERN:CS:
ACTIVE HIGH

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:CS:SOURce

機能 ユーザー定義バス信号トリガのチップセレクト信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SPATtern:CS:SOURce {<Nrf>|NONE}
:TRIGger{[:ATRigger]|:BTRigger):
SPATtern:CS:SOURce?

例 <Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
:TRIGGER:ATRIGGER:SPATTERN:CS:
SOURCE 1
:TRIGGER:ATRIGGER:SPATTERN:CS:
SOURCE?
-> :TRIGGER:ATRIGGER:SPATTERN:CS:
SOURCE 1

解説 「:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:DATA:SOURce」の設定によって設定できるチップセレクト信号が異なります。詳しくは本体ユーザーズマニュアル機能編をご覧ください。

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:DATA?

機能 ユーザー定義バス信号トリガのデータ信号のすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SPATtern:DATA?

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:DATA:ACTive

機能 ユーザー定義バス信号トリガのデータ信号のアクティブ状態を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SPATtern:DATA:ACTive {HIGH|LOW}
:TRIGger{[:ATRigger]|:BTRigger):
SPATtern:DATA:ACTive?

例 :TRIGGER:ATRIGGER:SPATTERN:DATA:
ACTIVE HIGH
:TRIGGER:ATRIGGER:SPATTERN:DATA:
ACTIVE?
-> :TRIGGER:ATRIGGER:SPATTERN:DATA:
ACTIVE HIGH

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:DATA:SOURce

機能 ユーザー定義バス信号トリガのデータ信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SPATtern:DATA:SOURce {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger):
SPATtern:DATA:SOURce?

例 <Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
:TRIGGER:ATRIGGER:SPATTERN:DATA:
SOURCE 1
:TRIGGER:ATRIGGER:SPATTERN:DATA:
SOURCE?
-> :TRIGGER:ATRIGGER:SPATTERN:DATA:
SOURCE 1

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:HEXa

機能 ユーザー定義バス信号トリガのデータ条件を 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SPATtern:HEXa {<文字列>}
:TRIGGER:ATRIGGER:SPATTERN:HEXa "12"

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:LATCh?

機能 ユーザー定義バス信号トリガのラッチ信号のすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger):
SPATtern:LATCh?

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:LATCh:POLarity

機能 ユーザー定義バス信号トリガのラッチ信号の極性を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:LATCh:POLarity {FALL|RISE}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:LATCh:POLarity?

例 :TRIGGER:ATRIGGER:SPATTERN:LATCH:
POLARITY FALL
:TRIGGER:ATRIGGER:SPATTERN:LATCH:
POLARITY?
-> :TRIGGER:ATRIGGER:SPATTERN:LATCH:
POLARITY FALL

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:LATCh:SOURce

機能 ユーザー定義バス信号トリガのラッチ信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:LATCh:SOURce {<NRF>|NONE}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:LATCh:SOURce?

例 :TRIGGER:ATRIGGER:SPATTERN:LATCH:
SOURCE 1
:TRIGGER:ATRIGGER:SPATTERN:LATCH:
SOURCE?
-> :TRIGGER:ATRIGGER:SPATTERN:LATCH:
SOURCE 1

解説 「:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:DATA:SOURce」の設定によって設定できるラッチ信号が異なります。詳しくは本体ユーザーズマニュアル機能編をご覧ください。

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:PATtern

機能 ユーザー定義バス信号トリガのデータ条件を2進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:PATtern?

例 :TRIGGER:ATRIGGER:SPATTERN:
PATTERN "00110101"
:TRIGGER:ATRIGGER:SPATTERN:PATTERN?
-> :TRIGGER:ATRIGGER:SPATTERN:
PATTERN "00110101"

:TRIGger{[:ATRigger]|:BTRigger}:SPI?

機能 SPI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{[:ATRigger]|:BTRigger}:SPI?

:TRIGger{[:ATRigger]|:BTRigger}:SPI:BITOrder

機能 SPIバス信号トリガデータのビットオーダーを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:SPI:
BITOrder {LSBFirst|MSBFirst}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:
BITOrder?

例 :TRIGGER:ATRIGGER:SPI:
BITORDER LSBFIRST
:TRIGGER:ATRIGGER:SPI:BITORDER?
-> :TRIGGER:ATRIGGER:SPI:
BITORDER LSBFIRST

:TRIGger{[:ATRigger]|:BTRigger}:SPI:CLOCK?

機能 SPIバス信号トリガのクロック信号のすべての設定値を問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:SPI:
CLOCK?

:TRIGger{[:ATRigger]|:BTRigger}:SPI:CLOCK:POLarity

機能 SPIバス信号トリガのクロック信号の極性を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:SPI:
CLOCK:POLarity {FALL|RISE}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:
CLOCK:POLarity?

例 :TRIGGER:ATRIGGER:SPI:CLOCK:
POLARITY FALL
:TRIGGER:ATRIGGER:SPI:CLOCK:
POLARITY?
-> :TRIGGER:ATRIGGER:SPI:CLOCK:
POLARITY FALL

:TRIGger{[:ATRigger]|:BTRigger}:SPI:CLOCK:SOURce

機能 SPIバス信号トリガのクロック信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:SPI:
CLOCK:SOURce {<NRF>|PODA<x>|PODB<x>|
PODC<x>|PODD<x>}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:
CLOCK:SOURce?

<NRF> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<x>、PODB<x>、PODC<x>、PODD<x>
の <x> = 0 ~ 7
例 :TRIGGER:ATRIGGER:SPI:CLOCK:SOURCE 1
:TRIGGER:ATRIGGER:SPI:CLOCK:SOURCE?
-> :TRIGGER:ATRIGGER:SPI:CLOCK:
SOURCE 1

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:SPI:CS?

機能 SPIバス信号トリガのチップセレクト信号のすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:SPI:CS?

:TRIGger{[:ATRigger]|:BTRigger}:SPI:CS:Active

機能 SPIバス信号トリガのチップセレクト信号のアクティブ状態を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:SPI:CS:Active {HIGH|LOW}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:CS:Active?

例 :TRIGGER:ATRIGGER:SPI:CS:ACTIVE HIGH
:TRIGGER:ATRIGGER:SPI:CS:ACTIVE?
-> :TRIGGER:ATRIGGER:SPI:CS:ACTIVE HIGH

:TRIGger{[:ATRigger]|:BTRigger}:SPI:CS:SOURce

機能 SPIバス信号トリガのチップセレクト信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:SPI:CS:SOURce {<Nrf>|PODA<x>|PODB<x>|PODC<x>|PODD<x>}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:CS:SOURce?

<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<x>、PODB<x>、PODC<x>、PODD<x>の <x> = 0 ~ 7

例 :TRIGGER:ATRIGGER:SPI:CS:SOURCE 1
:TRIGGER:ATRIGGER:SPI:CS:SOURCE?
-> :TRIGGER:ATRIGGER:SPI:CS:SOURCE 1

解説 「:TRIGger{[:ATRigger]|:BTRigger}:SPI:CLOCK:SOURce」の設定によって設定できるチップセレクト信号が異なります。詳しくは本体ユーザーズマニュアル機能編をご覧ください。

:TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>?

機能 SPIバス信号トリガのデータ信号のすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>?
<x> = 1 ~ 2

:TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:BCount

機能 SPIバス信号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{[:ATRigger]|:BTRigger}:SPI:DATA<x>:BCount {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:BCount?
<x> = 1 ~ 2
<Nrf> = 0 ~ 9999

例 :TRIGGER:ATRIGGER:SPI:DATA1:BCOUNT 0
:TRIGGER:ATRIGGER:SPI:DATA1:BCOUNT?
-> :TRIGGER:ATRIGGER:SPI:DATA1:BCOUNT 0

:TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:CONDition

機能 SPIバス信号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{[:ATRigger]|:BTRigger}:SPI:DATA<x>:CONDition {FALSE|TRUE}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:CONDition?
<x> = 1 ~ 2

例 :TRIGGER:ATRIGGER:SPI:DATA1:CONDITION FALSE
:TRIGGER:ATRIGGER:SPI:DATA1:CONDITION?
-> :TRIGGER:ATRIGGER:SPI:DATA1:CONDITION FALSE

:TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:DBYTe

機能 SPIバス信号トリガの各データのデータサイズ (バイト数) を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:DBYTe {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:DBYTe?
<x> = 1 ~ 2
<Nrf> = 1 ~ 4

例 :TRIGGER:ATRIGGER:SPI:DATA1:DBYTE 1
:TRIGGER:ATRIGGER:SPI:DATA1:DBYTE?
-> :TRIGGER:ATRIGGER:SPI:DATA1:DBYTE 1

:TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:DPATtern?

機能 SPIバス信号トリガの各データをに関するすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:DPATtern?
<x> = 1 ~ 2

**:TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:DPATtern:HEXa**

機能 SPIバス信号トリガの各データを16進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:DPATtern:HEXa {<文字列>}
<x> = 1 ~ 2

例 :TRIGGER:ATRIGGER:SPI:DATA1:
DPATTERN:HEXA "AB"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:SPI:
:DATA<x>:DBYTe」の設定値によって設定できる
バイト数が増減します。

**:TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:DPATtern:PATtern**

機能 SPIバス信号トリガの各データを2進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:DPATtern:PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:DPATtern:PATtern?
<x> = 1 ~ 2

例 :TRIGGER:ATRIGGER:SPI:DATA1:
DPATTERN:PATTERN "11001010"
:TRIGGER:ATRIGGER:SPI:DATA1:
DPATTERN:PATTERN?
-> :TRIGGER:ATRIGGER:SPI:DATA1:
DPATTERN:PATTERN "11001010"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:SPI:
:DATA<x>:DBYTe」の設定値によって設定できる
バイト数が増減します。

**:TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:MODE**

機能 SPIバス信号トリガのデータ信号の有効(ON)/ 無効(OFF)を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:MODE?
<x> = 1 ~ 2

例 :TRIGGER:ATRIGGER:SPI:DATA1:MODE ON
:TRIGGER:ATRIGGER:SPI:DATA1:MODE?
-> :TRIGGER:ATRIGGER:SPI:DATA1:
MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:SOURce**

機能 SPIバス信号トリガのデータ信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:SOURce {<NRF>|PODA<y>|
PODB<y>|PODC<y>|PODD<y>}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:SOURce?
<x> = 1 ~ 2
<NRF> = 1 ~ 8(4chモデルでは1 ~ 4)
PODA<y>、PODB<y>、PODC<y>、PODD<y>
の<y> = 0 ~ 7

例 :TRIGGER:ATRIGGER:SPI:DATA1:SOURCE 1
:TRIGGER:ATRIGGER:SPI:DATA1:SOURCE?
-> :TRIGGER:ATRIGGER:SPI:DATA1:
SOURCE 1

解説 「:TRIGger{[:ATRigger]|:BTRigger}:SPI:
:CLOCK:SOURce」の設定によって設定できるデータ
信号が異なります。詳しくは本体ユーザーズマ
ニュアル機能編をご覧ください。

**:TRIGger{[:ATRigger]|:BTRigger}:SPI:
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{[:ATRigger]|:BTRigger}:SPI:
MODE {WIRE3|WIRE4}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:
MODE?
例 :TRIGGER:ATRIGGER:SPI:MODE WIRE3
:TRIGGER:ATRIGGER:SPI:MODE?
-> :TRIGGER:ATRIGGER:SPI:MODE WIRE3

:TRIGger{[:ATRigger]|:BTRigger}:TYPE

機能 トリガの種類を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
TYPE {INTERval|OR|PATtern|PULSe|
RFTIME|RUNT|SIMple|TIMEout|TV|
WINDow|WOR|CANBus|CANFDbus|CXPiBus|
FLEXray|I2Cbus|LINbus|PSI5|SENT|
SPATtern|SPIbus|UART}
:TRIGger{[:ATRigger]|:BTRigger}:
TYPE?
例 :TRIGGER:ATRIGGER:TYPE SIMPLE
:TRIGGER:ATRIGGER:TYPE?
-> :TRIGGER:ATRIGGER:TYPE SIMPLE

解説 「:TRIGger:BTRigger:TYPE」は {OR|PATtern
|SIMple|WINDow|WOR|CANBus|CANFDbus|CXPiB
us|FLEXray|I2Cbus|LINbus|PSI5|SENT|SPATtern|SPI
Bus|UART} の選択肢が有効です。

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}:UART?

機能 UART 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{[:ATRigger]|:BTRigger}UART?

:TRIGger{[:ATRigger]|:BTRigger}:UART:BITOrder

機能 UART 信号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{[:ATRigger]|:BTRigger):
UART:BITOrder {LSBFirst|MSBFirst}
:TRIGger{[:ATRigger]|:BTRigger):
UART:BITOrder?

例 :TRIGGER:ATRIGGER:UART:
BITORDER LSBFIRST
:TRIGGER:ATRIGGER:UART:BITORDER?
-> :TRIGGER:ATRIGGER:UART:
BITORDER LSBFIRST

:TRIGger{[:ATRigger]|:BTRigger}: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{[:ATRigger]|:BTRigger):
UART:BRATe {<Nrf>|USER,<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger):
UART:BRATe?
<Nrf> = 1200、2400、4800、9600、19200
USER の <Nrf> = 本体ユーザーズマニュアル機能
編参照。

例 :TRIGGER:ATRIGGER:UART:BRATE 2400
:TRIGGER:ATRIGGER:UART:BRATE?
-> :TRIGGER:ATRIGGER:UART:BRATE 2400

:TRIGger{[:ATRigger]|:BTRigger}:UART:DATA?

機能 UART 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{[:ATRigger]|:BTRigger):
UART:DATA?

:TRIGger{[:ATRigger]|:BTRigger}:UART:DATA:ASCIi

機能 UART 信号トリガのデータを ASCII で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger):
UART:DATA:ASCIi {<文字列>}

例 :TRIGGER:ATRIGGER:UART:DATA:
ASCII "TEST"

:TRIGger{[:ATRigger]|:BTRigger}:UART:DATA:CONDition

機能 UART 信号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{[:ATRigger]|:BTRigger):
UART:DATA:CONDition {DONTcare|TRUE}
:TRIGger{[:ATRigger]|:BTRigger):
UART:DATA:CONDition?

例 :TRIGGER:ATRIGGER:UART:DATA:
CONDITION DONTCARE
:TRIGGER:ATRIGGER:UART:DATA:
CONDITION?
-> :TRIGGER:ATRIGGER:UART:DATA:
CONDITION DONTCARE

:TRIGger{[:ATRigger]|:BTRigger}:UART:DATA:CSensitive

機能 UART 信号トリガの ASCII データの大文字、小文字を区別する / しないの設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
UART:DATA:CSensitive {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger):
UART:DATA:CSensitive?

例 :TRIGGER:ATRIGGER:UART:DATA:
CSENSITIVE ON
:TRIGGER:ATRIGGER:UART:DATA:
CSENSITIVE?
-> :TRIGGER:ATRIGGER:UART:DATA:
CSENSITIVE 1

:TRIGger{[:ATRigger]|:BTRigger}:UART:DATA:DBYTe

機能 UART 信号トリガのデータのバイト数を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger):
UART:DATA:DBYTe {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger):
UART:DATA:DBYTe?
<Nrf> = 1 ~ 4(byte)

例 :TRIGGER:ATRIGGER:UART:DATA:DBYTE 1
:TRIGGER:ATRIGGER:UART:DATA:DBYTE?
-> :TRIGGER:ATRIGGER:UART:DATA:
DBYTE 1

:TRIGger{[:ATRigger]|:BTRigger}:UART:DATA:DPATtern?

機能 UART 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{[:ATRigger]|:BTRigger):
UART:DATA:DPATtern?

**:TRIGger{[:ATRigger]|:BTRigger}:UART
:DATA:DPATtern:HEXa**

機能 UART 信号トリガのデータを 16 進数で設定します。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:DPATtern:HEXa {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:DPATtern:HEXa?

例 :TRIGGER:ATRIGGER:UART:DATA:
DPATTERN:HEXA "12"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:UART:DATA:DBYte?」の設定値によって設定できるバイト数が変動します。

**:TRIGger{[:ATRigger]|:BTRigger}:UART
:DATA:DPATtern:PATtern**

機能 UART 信号トリガのデータを 2 進数で設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:DPATtern:PATtern {<文字列>}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:DPATtern:PATtern?

例 :TRIGGER:ATRIGGER:UART:DATA:
DPATTERN:PATTERN "00110101"
:TRIGGER:ATRIGGER:UART:DATA:
DPATTERN:PATTERN?

-> :TRIGGER:ATRIGGER:UART:DATA:
DPATTERN:PATTERN "00110101"

解説 「:TRIGger{[:ATRigger]|:BTRigger}:UART:DATA:DBYte?」の設定値によって設定できるバイト数が変動します。

**:TRIGger{[:ATRigger]|:BTRigger}:UART
:DATA:PFORmat**

機能 UART 信号トリガの ID とデータ条件のデータの
入力形式を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:PFORmat {ASci|PATtern}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:PFORmat?

例 :TRIGGER:ATRIGGER:UART:DATA:
PFORMAT PATTERN
:TRIGGER:ATRIGGER:UART:DATA:PFORMAT?
-> :TRIGGER:ATRIGGER:UART:DATA:
PFORMAT PATTERN

**:TRIGger{[:ATRigger]|:BTRigger}:UART
:ERROR?**

機能 UART 信号トリガの Error に関するすべての設定
値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
UART:ERROR?

**:TRIGger{[:ATRigger]|:BTRigger}:UART
:ERROR:FRAMing**

機能 UART 信号トリガの Framing Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
UART:ERROR:FRAMing {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:ERROR:FRAMing?

例 :TRIGGER:ATRIGGER:UART:ERROR:
FRAMING ON
:TRIGGER:ATRIGGER:UART:ERROR:
FRAMING?
-> :TRIGGER:ATRIGGER:UART:ERROR:
FRAMING 1

**:TRIGger{[:ATRigger]|:BTRigger}:UART
:ERROR:PARity**

機能 UART 信号トリガの Parity Error を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
UART:ERROR:PARity {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:ERROR:PARity?

例 :TRIGGER:ATRIGGER:UART:ERROR:
PARITY ON
:TRIGGER:ATRIGGER:UART:ERROR:PARITY?
-> :TRIGGER:ATRIGGER:UART:ERROR:
PARITY 1

**:TRIGger{[:ATRigger]|:BTRigger}:UART
:ERROR:PMODE**

機能 UART 信号トリガの Parity モードを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
UART:ERROR:PMODE {EVEN|ODD}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:ERROR:PMODE?

例 :TRIGGER:ATRIGGER:UART:ERROR:
PMODE EVEN
:TRIGGER:ATRIGGER:UART:ERROR:PMODE?
-> :TRIGGER:ATRIGGER:UART:ERROR:
PMODE EVEN

5.34 TRIGger グループ

:TRIGger{[:ATRigger]|:BTRigger}: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{[:ATRigger]|:BTRigger}:
UART:FORMat {BIT7Parity|BIT8Parity|
BIT8Noparity}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:FORMat?

例 :TRIGGER:ATRIGGER:UART:
FORMAT BIT7PARITY
:TRIGGER:ATRIGGER:UART:FORMAT?
-> :TRIGGER:ATRIGGER:UART:
FORMAT BIT7PARITY

:TRIGger{[:ATRigger]|:BTRigger}:UART:MODE

機能 UART 信号トリガのトリガの種類を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
UART:MODE {DATA|EDATa|ERRor}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:MODE?

例 :TRIGGER:ATRIGGER:UART:MODE DATA
:TRIGGER:ATRIGGER:UART:MODE?
-> :TRIGGER:ATRIGGER:UART:MODE DATA

:TRIGger{[:ATRigger]|:BTRigger}:UART:POLarity

機能 UART 信号解析の極性を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
UART:POLarity {NEGative|POSitive}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:POLarity?

例 :TRIGGER:ATRIGGER:UART:
POLARITY NEGATIVE
:TRIGGER:ATRIGGER:UART:POLARITY?
-> :TRIGGER:ATRIGGER:UART:
POLARITY NEGATIVE

:TRIGger{[:ATRigger]|:BTRigger}:UART:SOURce

機能 UART 信号解析の信号を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
UART:SOURce {<Nrf>|PODA<x>|PODB<x>|
PODC<x>|PODD<x>}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:SOURce?
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<x>、PODB<x>、PODC<x>、PODD<x>
の <x> = 0 ~ 7

例 :TRIGGER:ATRIGGER:UART:SOURCE 1
:TRIGGER:ATRIGGER:UART:SOURCE?
-> :TRIGGER:ATRIGGER:UART:SOURCE 1

:TRIGger{[:ATRigger]|:BTRigger}:UART:SPOint

機能 UART 信号解析のサンプルポイントを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
UART:SPOint {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:SPOint?
<Nrf> = 18.8 ~ 90.6

例 :TRIGGER:ATRIGGER:UART:SPOINT 18.8
:TRIGGER:ATRIGGER:UART:SPOINT?
-> :TRIGGER:ATRIGGER:UART:
SPOINT 18.8

:TRIGger{[:ATRigger]|:BTRigger}:WINDOW?

機能 ウィンドウトリガのすべての設定値を問い合わせます。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
WINDOW?

:TRIGger{[:ATRigger]|:BTRigger}:WINDOW:POLarity

機能 ウィンドウトリガのトリガソースの極性を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
WINDOW:POLarity {IN|OUT}
:TRIGger{[:ATRigger]|:BTRigger}:
WINDOW:POLarity?

例 :TRIGGER:ATRIGGER:WINDOW:POLARITY IN
:TRIGGER:ATRIGGER:WINDOW:POLARITY?
-> :TRIGGER:ATRIGGER:WINDOW:
POLARITY IN

:TRIGger{[:ATRigger]|:BTRigger}:WINDOW:SOURce

機能 ウィンドウトリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
WINDOW:SOURce {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
WINDOW:SOURce?
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :TRIGGER:ATRIGGER:WINDOW:SOURCE 1
:TRIGGER:ATRIGGER:WINDOW:SOURCE?
-> :TRIGGER:ATRIGGER:WINDOW:SOURCE 1

:TRIGger{[:ATRigger]|:BTRigger}:WINDOW:TQualify

機能 ウィンドウトリガの時間条件を設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:
WINDOW:TQualify {INSide|LTHan|MTHan|
NONE|OUTSide|TIMEout}
:TRIGger{[:ATRigger]|:BTRigger}:
WINDOW:TQualify?

例 :TRIGGER:ATRIGGER:WINDOW:
TQUALIFY INSIDE
:TRIGGER:ATRIGGER:WINDOW:TQUALIFY?
-> :TRIGGER:ATRIGGER:WINDOW:
TQUALIFY INSIDE

:TRIGger{[:ATRigger]|:BTRigger}:WOR?

機能 ウィンドウ OR 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{[:ATRigger]|:BTRigger}:WOR?

:TRIGger{[:ATRigger]|:BTRigger}:WOR:ALL

機能 すべてのチャンネルを一括して、ウィンドウ OR トリガのトリガソースの対象にする / しないを設定 / 問い合わせします

構文 :TRIGger{[:ATRigger]|:BTRigger}:WOR:
ALL {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:WOR:
ALL?

例 :TRIGGER:ATRIGGER:WOR:ALL ON
:TRIGGER:ATRIGGER:WOR:ALL?
-> :TRIGGER:ATRIGGER:WOR:ALL 1

:TRIGger{[:ATRigger]|:BTRigger}:WOR:CAHNnel<x>

機能 ウィンドウ OR トリガの各チャンネルのスロープを設定 / 問い合わせします。

構文 :TRIGger{[:ATRigger]|:BTRigger}:WOR:
CAHNnel<x> {IN|OFF|OUT}
:TRIGger{[:ATRigger]|:BTRigger}:WOR:
CAHNnel<x>?
<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :TRIGGER:ATRIGGER:WOR:CAHNNEL1 IN
:TRIGGER:ATRIGGER:WOR:CAHNNEL1?
-> :TRIGGER:ATRIGGER:WOR:CAHNNEL IN

:TRIGger[:ATRigger]:INTERval?

機能 インターバル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[:ATRigger]:INTERval?

:TRIGger[:ATRigger]:INTERval:SLOPe

機能 インターバルトリガのトリガスロープを設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:INTERval:
SLOPe {FALL|RISE}
:TRIGger[:ATRigger]:INTERval:SLOPe?

例 :TRIGGER:ATRIGGER:INTERVAL:
SLOPE FALL
:TRIGGER:ATRIGGER:INTERVAL:SLOPE?
-> :TRIGGER:ATRIGGER:INTERVAL:
SLOPE FALL

:TRIGger[:ATRigger]:INTERval:SOURce

機能 インターバルトリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:INTERval:
SOURce {<Nrf>|PODA<x>|PODB<x>|
PODC<x>|PODD<x>}
:TRIGger[:ATRigger]:INTERval:SOURce?
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<x>, PODB<x>, PODC<x>, PODD<x>
の <x> = 0 ~ 7

例 :TRIGGER:ATRIGGER:INTERVAL:SOURCE 1
:TRIGGER:ATRIGGER:INTERVAL:SOURCE?
-> :TRIGGER:ATRIGGER:INTERVAL:
SOURCE 1

:TRIGger[:ATRigger]:INTERval:TQQualify

機能 インターバルトリガの時間条件を設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:INTERval:
TQQualify {INSide|LTHan|MTHan|
OUTSide|TIMEout}
:TRIGger[:ATRigger]:INTERval:
TQQualify?
例 :TRIGGER:ATRIGGER:INTERVAL:
TQUALIFY INSIDE
:TRIGGER:ATRIGGER:INTERVAL:TQUALIFY?
-> :TRIGGER:ATRIGGER:INTERVAL:
TQUALIFY INSIDE

:TRIGger[:ATRigger]:PULSe?

機能 パルス幅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[:ATRigger]:PULSe?

5.34 TRIGger グループ

:TRIGger[:ATRigger]:PULSe:POLarity

機能 パルス幅トリガのトリガソースの極性を設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:PULSe:
POLarity {NEGative|POSitive}
:TRIGger[:ATRigger]:PULSe:POLarity?

例 :TRIGGER:ATRIGGER:PULSE:
POLARITY NEGATIVE
:TRIGGER:ATRIGGER:PULSE:POLARITY?
-> :TRIGGER:ATRIGGER:PULSE:
POLARITY NEGATIVE

:TRIGger[:ATRigger]:PULSe: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[:ATRigger]:PULSe:
SOURce {<NRf>|PODA<x>|PODB<x>|
PODC<x>|PODD<x>}
:TRIGger[:ATRigger]:PULSe:SOURce?
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<x>、PODB<x>、PODC<x>、PODD<x>
の <x> = 0 ~ 7

例 :TRIGGER:ATRIGGER:PULSE:SOURCE 1
:TRIGGER:ATRIGGER:PULSE:SOURCE?
-> :TRIGGER:ATRIGGER:PULSE:SOURCE 1

:TRIGger[:ATRigger]:PULSe:TQQualify

機能 パルス幅トリガの時間条件を設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:PULSe:
TQQualify {INSide|LTHan|MTHan|
OUTSide|TIMEout}
:TRIGger[:ATRigger]:PULSe:TQQualify?

例 :TRIGGER:ATRIGGER:PULSE:
TQUALIFY INSIDE
:TRIGGER:ATRIGGER:PULSE:TQUALIFY?
-> :TRIGGER:ATRIGGER:PULSE:
TQUALIFY INSIDE

:TRIGger[:ATRigger]:RFTime?

機能 立ち上がり / 立ち下がり時間トリガのすべての設定値を問い合わせます。

構文 :TRIGger[:ATRigger]:RFTime?

:TRIGger[:ATRigger]:RFTime:SLOPe

機能 立ち上がり / 立ち下がり時間トリガのトリガスロープを設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:RFTime:
SLOPe {BOTH|FALL|RISE}
:TRIGger[:ATRigger]:RFTime:SLOPe?
例 :TRIGGER:ATRIGGER:RFTIME:SLOPE FALL
:TRIGGER:ATRIGGER:RFTIME:SLOPE?
-> :TRIGGER:ATRIGGER:RFTIME:
SLOPE FALL

:TRIGger[:ATRigger]:RFTime:SOURce

機能 立ち上がり / 立ち下がり時間トリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:RFTime:
SOURce {<NRf>}
:TRIGger[:ATRigger]:RFTime:SOURce?
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :TRIGGER:ATRIGGER:RFTIME:SOURCE 1
:TRIGGER:ATRIGGER:RFTIME:SOURCE?
-> :TRIGGER:ATRIGGER:RFTIME:SOURCE 1

:TRIGger[:ATRigger]:RFTime:TQQualify

機能 立ち上がり / 立ち下がり時間トリガの時間条件を設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:RFTime:
TQQualify {INSide|LTHan|MTHan|
OUTSide}
:TRIGger[:ATRigger]:RFTime:TQQualify?

例 :TRIGGER:ATRIGGER:RFTIME:
TQUALIFY INSIDE
:TRIGGER:ATRIGGER:RFTIME:TQUALIFY?
-> :TRIGGER:ATRIGGER:RFTIME:
TQUALIFY INSIDE

:TRIGger[:ATRigger]:RUNT?

機能 ラン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[:ATRigger]:RUNT?

:TRIGger[:ATRigger]:RUNT:POLarity

機能 ラントトリガのトリガソースの極性を設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:RUNT:
POLarity {EITHer|NEGative|POSitive}
:TRIGger[:ATRigger]:RUNT:POLarity?
例 :TRIGGER:ATRIGGER:RUNT:
POLARITY NEGATIVE
:TRIGGER:ATRIGGER:RUNT:POLARITY?
-> :TRIGGER:ATRIGGER:RUNT:
POLARITY NEGATIVE

:TRIGger[:ATRigger]:RUNT:SOURce

機能 ラントトリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:RUNT:
SOURce {<NRf>}
:TRIGger[:ATRigger]:RUNT:SOURce?
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
例 :TRIGGER:ATRIGGER:RUNT:SOURCE 1
:TRIGGER:ATRIGGER:RUNT:SOURCE?
-> :TRIGGER:ATRIGGER:RUNT:SOURCE 1

:TRIGger[:ATRigger]:RUNT:TQQualify

機能 ラントリガの時間条件を設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:RUNT:
TQQualify {INSide|LTHan|MTHan|NONE|
OUTSide}

例 :TRIGGER:ATRIGGER:RUNT:TQQualify?
TQUALIFY INSIDE
:TRIGGER:ATRIGGER:RUNT:TQUALIFY?
-> :TRIGGER:ATRIGGER:RUNT:
TQUALIFY INSIDE

:TRIGger[:ATRigger]:TIMEout?

機能 タイムアウトトリガのすべての設定値を問い合わせます。

構文 :TRIGger[:ATRigger]:TIMEout?

:TRIGger[:ATRigger]:TIMEout:POLarity

機能 タイムアウトトリガのトリガソースの極性を設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:TIMEout:
POLarity {EITHer|NEGative|POSitive}
:TRIGger[:ATRigger]:TIMEout:
POLarity?

例 :TRIGGER:ATRIGGER:TIMEOUT:
POLARITY NEGATIVE
:TRIGGER:ATRIGGER:TIMEOUT:POLARITY?
-> :TRIGGER:ATRIGGER:TIMEOUT:
POLARITY NEGATIVE

:TRIGger[:ATRigger]:TIMEout:SOURce

機能 タイムアウトトリガのトリガソースを設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:TIMEout:
SOURce {<Nrf>|PODA<x>|PODB<x>|
PODC<x>|PODD<x>}
:TRIGger[:ATRigger]:TIMEout:SOURce?
<Nrf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<x>、PODB<x>、PODC<x>、PODD<x>
の <x> = 0 ~ 7

例 :TRIGGER:ATRIGGER:TIMEOUT:SOURCE 1
:TRIGGER:ATRIGGER:TIMEOUT:SOURCE?
-> :TRIGGER:ATRIGGER:TIMEOUT:
SOURCE 1

:TRIGger[:ATRigger]: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[:ATRigger]:TV?

:TRIGger[:ATRigger]:TV:FIELD

機能 TV トリガをかけるフィールドを設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:TV:
FIELD {DONTcare|<Nrf>}
:TRIGger[:ATRigger]:TV:FIELD?
<Nrf> = 1、2

例 :TRIGGER:ATRIGGER:TV:FIELD 1
:TRIGGER:ATRIGGER:TV:FIELD?
-> :TRIGGER:ATRIGGER:TV:FIELD 1

:TRIGger[:ATRigger]: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[:ATRigger]:TV:FRAME {<Nrf>}
:TRIGger[:ATRigger]:TV:FRAME?
<Nrf> = 1、2、4、8

例 :TRIGGER:ATRIGGER:TV:FRAME 2
:TRIGGER:ATRIGGER:TV:FRAME?
-> :TRIGGER:ATRIGGER:TV:FRAME 2

:TRIGger[:ATRigger]: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[:ATRigger]:TV:{HDTV|NTSC|
PAL|SDTV|USERdefine}?

:TRIGger[:ATRigger]: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[:ATRigger]:TV:{HDTV|NTSC|
PAL|SDTV|USERdefine}:LINE {<Nrf>}
:TRIGger[:ATRigger]:TV:{HDTV|NTSC|
PAL|SDTV|USERdefine}:LINE?
<Nrf> = 2 ~ 2251

例 以下は、HDTV についての例です。
:TRIGGER:ATRIGGER:TV:HDTV:LINE 2
:TRIGGER:ATRIGGER:TV:HDTV:LINE?
-> :TRIGGER:ATRIGGER:TV:HDTV:LINE 2

5.34 TRIGger グループ

:TRIGger[:ATRigger]: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[:ATRigger]:TV:HDTV:
POLarity {POSitive|NEGative}
:TRIGger[:ATRigger]:TV:HDTV:
POLarity?

例 以下は、HDTV についての例です。
:TRIGGER:ATRIGGER:TV:HDTV:
POLARITY POSITIVE
:TRIGGER:ATRIGGER:TV:HDTV:POLARITY?
-> :TRIGGER:ATRIGGER:TV:HDTV:
POLARITY POSITIVE

:TRIGger[:ATRigger]: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[:ATRigger]:TV:LEVel {<NRf>}
:TRIGger[:ATRigger]:TV:LEVel?
<NRf> = 0.1(div) ~ 2.0(div)、0.1div ステップ
例 :TRIGGER:ATRIGGER:TV:LEVEL 0.1
:TRIGGER:ATRIGGER:TV:LEVEL?
-> :TRIGGER:ATRIGGER:TV:LEVEL 0.1

:TRIGger[:ATRigger]:TV:LFORMAT

機能 TV トリガをかけるラインの指定方式を設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:TV:LFORMAT {ALL|
ONE}
:TRIGger[:ATRigger]:TV:LFORMAT?
例 :TRIGGER:ATRIGGER:TV:LFORMAT ALL
:TRIGGER:ATRIGGER:TV:LFORMAT?
-> :TRIGGER:ATRIGGER:TV:LFORMAT ALL

:TRIGger[:ATRigger]: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[:ATRigger]:TV:
SOURce {<NRf>}
:TRIGger[:ATRigger]:TV:SOURce?
<NRf> = 1 ~ 8
例 :TRIGGER:ATRIGGER:TV:SOURCE 1
:TRIGGER:ATRIGGER:TV:SOURCE?
-> :TRIGGER:ATRIGGER:TV:SOURCE 1

:TRIGger[:ATRigger]: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[:ATRigger]:TV:TYPE {NTSC|
HDTV[, {I1080 _ 50|I1080 _ 60|P1080 _ 24|
P1080 _ 25|P1080 _ 60|P720 _ 60|
SF1080 _ 24}]]|PAL|SDTV|USERdefine}
:TRIGger[:ATRigger]:TV:TYPE?

例 :TRIGGER:ATRIGGER:TV:TYPE NTSC
:TRIGGER:ATRIGGER:TV:TYPE?
-> :TRIGGER:ATRIGGER:TV:TYPE NTSC

:TRIGger[:ATRigger]:TV:USERdefine?

機能 ユーザー定義 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[:ATRigger]:TV:USERdefine?

:TRIGger[:ATRigger]:TV:USERdefine:DEFinition

機能 ユーザー定義 TV トリガの解像度を設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:TV:USERdefine:
DEFinition {HD|SD}
:TRIGger[:ATRigger]:TV:USERdefine:
DEFinition?

例 :TRIGGER:ATRIGGER:TV:USERDEFINE:
DEFINITION HD
:TRIGGER:ATRIGGER:TV:USERDEFINE:
DEFINITION?
-> :TRIGGER:ATRIGGER:TV:USERDEFINE:
DEFINITION HD

:TRIGger[:ATRigger]:TV:USERdefine:HFRejection (HighFrequencyREJECTION)

機能 ユーザー定義 TV トリガのローパスフィルター (HF リジェクション) を設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:TV:USERdefine:
HFRejection {<周波数>|OFF}
:TRIGger[:ATRigger]:TV:USERdefine:
HFRejection?

<周波数> = 300kHz
例 :TRIGGER:ATRIGGER:TV:USERDEFINE:
HFREJECTION 300KHZ
:TRIGGER:ATRIGGER:TV:USERDEFINE:
HFREJECTION?
-> :TRIGGER:ATRIGGER:TV:USERDEFINE:
HFREJECTION 300000

:TRIGger[:ATRigger]:TV:USERdefine:HSYNc (Hsync Freq)

機能 ユーザー定義 TV トリガ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[:ATRigger]:TV:USERdefine:HSYNc {<周波数>}

:TRIGger[:ATRigger]:TV:USERdefine:HSYNc?

例 <周波数> = 10k ~ 200k(Hz)

:TRIGGER:ATRIGGER:TV:USERDEFINE:HSYNC 10KHZ

:TRIGGER:ATRIGGER:TV:USERDEFINE:HSYNC?

-> :TRIGGER:ATRIGGER:TV:USERDEFINE:HSYNC 10.000E+03

:TRIGger[:ATRigger]:TV:USERdefine:SGUard

機能 ユーザー定義 TV トリガの Sync Guard を設定 / 問い合わせします。

構文 :TRIGger[:ATRigger]:TV:USERdefine:SGUard {<Nrf>}

:TRIGger[:ATRigger]:TV:USERdefine:SGUard?

例 <Nrf> = 60 ~ 90(%)

:TRIGGER:ATRIGGER:TV:USERDEFINE:SGUARD 60

:TRIGGER:ATRIGGER:TV:USERDEFINE:SGUARD?

-> :TRIGGER:ATRIGGER:TV:USERDEFINE:SGUARD 60

:TRIGger:COMBination

機能 トリガの組み合わせを設定 / 問い合わせします。

構文 :TRIGger:COMBination {ABN|ADB|AORB|OFF}

:TRIGger:COMBination?

例 :TRIGGER:COMBINATION ABN

:TRIGGER:COMBINATION?

-> :TRIGGER:COMBINATION ABN

:TRIGger:DElay?

機能 デ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?

:TRIGger:DElay:TIME

機能 デレイ (トリガ点からトリガポジションまでの時間) を設定 / 問い合わせします。

構文 :TRIGger:DElay:TIME {<時間>}

:TRIGger:DElay:TIME?

<時間> = 本体ユーザズマニュアル機能編参照。

例 :TRIGGER:DELAY:TIME 2US

:TRIGGER:DELAY:TIME?

-> :TRIGGER:DELAY:TIME 2.00E-06

解説 タイムベースが外部クロックのときは 0 固定です。本機器では、トリガデレイはトリガ点からトリガポジションまでの時間で設定します。トリガポジションは「:TRIGger:POSition」で設定 / 問い合わせします。

:TRIGger:DEtect?

機能 トリガがかかったかどうかを問い合わせます。

構文 :TRIGger:DEtect? [{<Nrf>}]

<Nrf> = 0 ~ 360000 (タイムアウト時間、10 ms 単位)

例 :TRIGGER:DETECT? 0

-> :TRIGGER:DETECT 1

解説

- ・トリガがかかると「1」、かからなかった場合は「0」を返します。本コマンドの送信前にトリガがかかっていた場合は「1」を返します。
- ・タイムアウト時間を設定した場合は、トリガがかかったときに「1」を返します。タイムアウト時間内にトリガがかからなかった場合は「0」を返します。タイムアウト時間を「0」に設定した場合は、本コマンドを送信したときに、トリガがかかっていると「1」、かかっていないと「0」を返します。
- ・タイムアウト時間は省略可能です。省略した場合は、<Nrf>=0 のときと同様の動作になります。
- ・次の動作をしたとき、トリガが検出されてない (本コマンドで「0」と判定される) 状態にリセットします。
 - ・本コマンドを送信したとき。
 - ・波形の取り込みをスタートしたとき。
 - ・波形の取り込み中に波形の取り込み条件を変更したとき。

:TRIGger:FORCE

機能 強制的にトリガをかけます。

構文 :TRIGger:FORCE

例 :TRIGGER:FORCE

5.34 TRIGger グループ

:TRIGger:HOLDoff?

機能 ホールドオフに関するすべての設定値を問い合わせます。

構文 :TRIGger:HOLDoff?

:TRIGger:HOLDoff:TIME

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:TIME {<時間>}

:TRIGger:HOLDoff:TIME?

<時間> = 20ns ~ 10s

例 :TRIGGER:HOLDOFF:TIME 20NS

:TRIGGER:HOLDOFF:TIME?

-> :TRIGGER:HOLDOFF:TIME 20.0E-09

: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 {AUTO|ALeVel|NORMal|NSINgle}

:TRIGger:MODE?

例 :TRIGGER:MODE AUTO

:TRIGGER:MODE?

-> :TRIGGER:MODE AUTO

解説 「NSINgle」時のカウントは、「:TRIGger:SCOUNT」で設定 / 問い合わせします。

: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(%), 0.1% ステップ

例 :TRIGGER:POSITION 50

:TRIGGER:POSITION?

-> :TRIGGER:POSITION 50.000

:TRIGger:SCOUNT

機能 トリガモードがN Single のとき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> = 1 ~ 100000

例 :TRIGGER:SCOUNT 100

:TRIGGER:SCOUNT?

-> :TRIGGER:SCOUNT 100

:TRIGger:SOURce?

機能 ENHANCED トリガの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?

:TRIGger:SOURce:CHANnel<x>?

機能 ENHANCED トリガの指定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:CHANnel<x>?

<x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:TRIGger:SOURce:CHANnel<x>:HFRejection (HighFrequencyREJECTION)

機能 ENHANCED トリガのときの指定トリガソースのローパスフィルターを設定 / 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 ~ 8(4ch モデルでは 1 ~ 4)

<周波数> = 15KHz ~ 20MHz

例 :TRIGGER:SOURCE:CHANNEL1:

HFREJECTION 15KHZ

:TRIGGER:SOURCE:CHANNEL1:

HFREJECTION?

-> :TRIGGER:SOURCE:CHANNEL1:

HFREJECTION 15000

:TRIGger:SOURce:CHANnel<x>:LEVel

機能 ENHANCED トリガのときの指定トリガソース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>}

:TRIGger:SOURce:CHANnel<x>:LEVel?

1 ~ 8(4ch モデルでは 1 ~ 4)

<電圧> = 本体ユーザーズマニュアル機能編参照。

例 :TRIGGER:SOURCE:CHANNEL1:LEVEL 0V

:TRIGGER:SOURCE:CHANNEL1:LEVEL?

-> :TRIGGER:SOURCE:CHANNEL1:

LEVEL 0.000E+00

解説 電流プローブ設定時は、<電流> 値の設定 / 問い合わせとなります。

**:TRIGger:SOURce:CHANnel<x>:NREJecti
on**

機能 ENHANCED トリガのときの指定トリガソースの
 ノイズリジェクションを設定 / 問い合わせしま
 す。

構文 :TRIGger:SOURce:CHANnel<x>:
 NREJection {HIGH|LOW|MIDDLE}
 :TRIGger:SOURce:CHANnel<x>:
 NREJection?
 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :TRIGGER:SOURCE:CHANNEL1:
 NREJECTION HIGH
 :TRIGGER:SOURCE:CHANNEL1:NREJECTION?
-> :TRIGGER:SOURCE:CHANNEL1:
 NREJECTION HIGH

:TRIGger:SOURce:CHANnel<x>:ULLeVel

機能 ENHANCED トリガのトリガソースのトリガレベ
 ルの上下限値を設定 / 問い合わせします。

構文 :TRIGger:SOURce:CHANnel<x>:
 ULLeVel {< 電圧>,< 電圧>}
 :TRIGger:SOURce:CHANnel<x>:ULLeVel?
 1 ~ 8(4ch モデルでは 1 ~ 4)
 < 電圧> = 本体ユーザーズマニュアル機能編参
 照。

例 :TRIGGER:SOURCE:CHANNEL1:
 ULLEVEL 0.5V,-0.5V
 :TRIGGER:SOURCE:CHANNEL1:ULLEVEL?
-> :TRIGGER:SOURCE:CHANNEL1:
 ULLEVEL 0.500E+00,-0.500E+00

:TRIGger:TIME<x>

機能 ENHANCED トリガのときの時間を設定 / 問い合
 わせします。

構文 :TRIGger:TIME<x> {<NRf>}
 :TRIGger:TIME<x>?
 <x> = 1 ~ 2
 <NRf> = 本体ユーザーズマニュアル機能編参照。

例 :TRIGGER:TIME1 1S
 :TRIGGER:TIME1?
-> :TRIGGER:TIME1 1.000E+00

5.35 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?

:WAVEform:ALL?

機能 「:WAVEform:ALL:SEND?」に関するすべての設定値を問い合わせます。

構文 :WAVEform:ALL?

:WAVEform:ALL:SEND?

機能 「:WAVEform:ALL:TRACe」で指定した波形データを問い合わせます。

構文 :WAVEform:ALL:SEND? {<NRF>}

<NRF> = 0 ~ (アクイジションカウント)

例 :WAVEFORM:ALL:SEND? 0

-> #800007558

03 00 85 00 00 00 00 00 00 00 00 00

00 00 E2 04

00 00 00 00 00 00 E2 04 00 00 62 00

AC 11 44 00

6C 00 26 00 08 00 3C 18 3E 18 4E 18

0A 18 18 18 ...

- 解説
- 測定中に波形データの問い合わせが可能です。
 - トリガモードが Single、NSingle のときは問い合わせできません。
 - ロールモードで動作中のときは問い合わせできません。
 - 波形パラメータの自動測定の継続統計処理、またはユーザー定義演算が有効なときは問い合わせできません。
 - データフォーマットは次のとおりです。
#8(8桁(10進)の送信データバイト数) <- ブロックデータのヘッダ
(トレース数 [2 バイト])(アクイジションカウント [8 バイト]) <- 共通情報
(トレース番号 [4 バイト])(リザーブ領域 [8 バイト])(データポイント数 [4 バイト]) <- トレース情報 1
(データ列)...
(トレース番号 [4 バイト])(リザーブ領域 [8 バイト])(データポイント数 [4 バイト]) <- トレース情報 2
(データ列)...
..... (トレース数分)
 - 共通情報、トレース情報のバイトオーダーはリトルエンディアンです。
 - データ列のフォーマットは「WORD」だけです。バイトオーダーはリトルエンディアンです。
 - レコード長は 12.5kPoints まで問い合わせできます。
 - <NRF>(アクイジションカウント)に 0 を指定すると、最新の波形データを取得できます。
 - 同じ波形データを取得しないように波形を取り込むには、前回、本コマンドで取得した共通情報のアクイジションカウントを、<NRF>で指定します。問い合わせた時点で、指定した値と本体アクイジションカウントが同じ場合、#8000000000 を返します。
 - フォーマットが「DWORD」の時、標準 16Bit モデルの応答は下位 16Bit に取り込みます。

:WAVEform:ALL:TRACE

機能 「:WAVEform:ALL:SEND?」で対象となる波形を設定 / 問い合わせます。

構文 :WAVEform:ALL:TRACE {<NRf>|ALL|LOGic|MATH<x>}:WAVEform:ALL:TRACE?
 <NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
 <x> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :WAVEFORM:ALL:TRACE 1
 :WAVEFORM:ALL:TRACE?
 -> :WAVEFORM:ALL:TRACE 1

: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 波形データが 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 ~ 499999999

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 送信する波形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: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?

-> :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。
 ・「:SYSTem:OCANcel」が「ON」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?

-> :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。

: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 {<NRf>|AVERage|MINimum}
 :WAVEform:RECORD?

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、アキュ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ズマニュアル機能編をご覧ください。

5.35 WAVEform グループ

:WAVEform:RECOrd? MINImum

機能 対象チャネルのヒストリの最小レコード番号を問い合わせます。

構文 :WAVEform:RECOrd? MINImum

例 :WAVEFORM:RECORD? MINIMUM
-> :WAVEFORM:RECORD -49999

: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 ~ 100000
レコード長の設定に応じて、<NRf> の設定範囲が異なります。

例 :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。
 - 「ASCIi」にしたとき <電圧>、<電圧>、...<電圧> の形式で返されます。
 - 「BYTe」、「WORD」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
「WORD」: Division = 3200
 - 「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> から <NRf> 回、波形データを順に問い合わせます。
- 「:CHANnel<x>:LScale:MODE」が「ON」のときは、次の式で、リニアスケール値に換算できます。
リニアスケール値 = 電圧(演算値) × スケーリング係数 A* + オフセット値 B*
* スケーリング係数 A = 「:CHANnel<x>:LScale:AVALue?」の返値
オフセット値 B = 「:CHANnel<x>:LScale:BVALue?」の返値
- (4) レコード長は 500MPoints まで問い合わせできます。

: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

:WAVEform:SRAt? (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:SRAt?

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 ~ 499999999

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>|LOGic|
MATH<x>}
:WAVEform:TRACe?
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<x> = 1 ~ 8(4ch モデルでは 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

5.36 WPARameter グループ

: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 :WPARameter<x>?
<x> = 1、2

:WPARameter<x>:DISPlay

機能 波形パラメータ測定の表示の ON/OFF を設定 / 問い合わせします。

構文 :WPARameter<x>:DISPlay {<Boolean>}
:WPARameter<x>:DISPlay?
<x> = 1、2

例 :WPARAMETER1:DISPLAY ON
:WPARAMETER1:DISPLAY?
-> :WPARAMETER1:DISPLAY 1

: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 :WPARameter<x>:HISTogram?
<x> = 1、2

: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 :WPARameter<x>:HISTogram:MEASure?
<x> = 1、2

:WPARameter<x>:HISTogram:MEASure:MODE

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モードを設定 / 問い合わせします。

構文 :WPARameter<x>:HISTogram:MEASure:
MODE {OFF|PARAMeter}
:WPARameter<x>:HISTogram:MEASure:
MODE?
<x> = 1、2

例 :WPARAMETER1:HISTOGRAM:MEASURE:
MODE OFF
:WPARAMETER1:HISTOGRAM:MEASURE:MODE?
-> :WPARAMETER1:HISTOGRAM:MEASURE:
MODE OFF

:WPARameter<x>:HISTogram:MEASure:PARameter?

機能 波形パラメータ測定の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。

構文 :WPARameter<x>:HISTogram:MEASure:
PARameter?
<x> = 1、2

:WPARameter<x>:HISTogram:MEASure:PARameter:ALL

機能 波形パラメータ測定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。

構文 :WPARameter<x>:HISTogram:MEASure:
PARameter:ALL {<Boolean>}
<x> = 1、2

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

:WPARameter<x>:HISTogram:MEASure:PARameter:<パラメータ>?

機能 波形パラメータ測定の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。

構文 :WPARameter<x>:HISTogram:MEASure:
PARameter:<パラメータ>?
<x> = 1、2
<パラメータ> = {C1|C2|DC|MAXimum|MEAN|M
EDian|MINimum|PEAK|SD2integ|SD3integ|SDEVia
tion|SDInteg}

:WPARameter<x>:HISTogram:MEASure:PARameter:<パラメータ>:STATE

機能 波形パラメータ測定の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。

構文 :WPARameter<x>:HISTogram:MEASure:
PARameter:<パラメータ>:STATE
{<Boolean>}

:WPARameter<x>:HISTogram:MEASure:
PARameter:<パラメータ>:STATE?
<x> = 1、2
<パラメータ> = {C1|C2|DC|MAXimum|MEAN|M
EDian|MINimum|PEAK|SD2integ|SD3integ|SDEVia
tion|SDInteg}

例 以下は、最大値についての例です。
:WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER1:MAXIMUM:STATE ON
:WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER1:MAXIMUM:STATE?
-> :WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER1:MAXIMUM:STATE 1

5.36 WPARameter グループ

:WPARameter<x>:HISTogram:MEASure:PARameter:<パラメータ>:VALue?

機能 波形パラメータ測定の手istogramパラメータの測定値を問い合わせます。

構文 :WPARameter<x>:HISTogram:MEASure:PARameter:<パラメータ>:VALue?
<x> = 1、2
<パラメータ> = {C1|C2|DC|MAXimum|MEAN|MEDian|MINimum|PEAK|SD2integ|SD3integ|SDEVia tion|SDInteg}

例 以下は、最大値についての例です。
:WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER:MAXIMUM:VALUE?
-> :WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER:MAXIMUM:VALUE 1.000E+00

:WPARameter<x>:HISTogram:MEASure:PARameter:POSition<y>

機能 波形パラメータ測定の手istogramの各パラメータの位置を設定 / 問い合わせします。

構文 :WPARameter<x>:HISTogram:MEASure:PARameter:POSition<y> {<NRF>}
:WPARameter<x>:HISTogram:MEASure:PARameter:POSition<y>?
<x> = 1、2
<y> = 1、2
<NRF> = -5 ~ 5(div)

例 :WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER1:POSITION 1
:WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER1:POSITION?
-> :WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER1:POSITION 1.000E+00

:WPARameter<x>:ITEM

機能 波形パラメータ測定の手istogramパラメータを設定 / 問い合わせします。

構文 :WPARameter<x>:ITEM {<NRF>|MATH<y>|PODA<y>|PODB<y>|PODC<y>|PODD<y>,<パラメータ>[,2]}
:WPARameter<x>:ITEM?
WPARameter<x>の<x> = 1、2
<NRF> = 1 ~ 8(4ch モデルでは 1 ~ 4)
PODA<y>|PODB<y>|PODC<y>|PODD<y>の<y> = 0 ~ 7
MATH<y>の<y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<パラメータ> = {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDTH|DElay|DT|DUTYcycle|ENUMber|FALL|FREQuency|I2T|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWIDTH|PABS|PERiod|PN|PNUMber|PON|POVershoot|PP|PTOff|PTON|PTOPeak|PWIDTH|RISE|RMS|SDEviation|TY1Integ|TY2Integ|V1|V2|WHABs|WHN|WHP|WHTOff|WHTON|WHTOTa||Z}

例 :WPARAMETER1:ITEM 1,AVERAGE
:WPARAMETER1:ITEM?
-> :WPARAMETER1:ITEM 1,AVERAGE

解説

- BIT<y>の場合、設定可能な<パラメータ>は {AVGFreq|DElay|DUTYcycle|FREQuency|PERiod|PNUMBER} になります。
- CALC<y>の場合、パラメータは必要ありません。
- <NRF>、MATH<y>の場合、設定可能な<パラメータ>は {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDTH|DElay|DT|DUTYcycle|ENUMber|FALL|FREQuency|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWIDTH|PERiod|PNUMBER|POVershoot|PTOPeak|PWIDTH|RISE|RMS|SDEviation|TY1Integ|TY2Integ|V1|V2} になります。
- <パラメータ>に {PON|PTOff|PTON|PTOTAL|WHON|WHTOff|WHTON|WHTOTAL} を設定する場合、<パラメータ>のあとの[,2]は設定できません。

:WPARameter<x>:MODE

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

構文 :WPARameter<x>:MODE {HISTogram|TREND}
:WPARameter<x>:MODE?
<x> = 1、2

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

:WPARameter<x>:TREND?

機能 波形パラメータ測定の特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 :WPARameter<x>:TREND?
<x> = 1、2

:WPARameter<x>:TREND:AScale

機能 波形パラメータ測定の特レンド表示のオートスケールを実行します。

構文 :WPARameter<x>:TREND:AScale
<x> = 1、2

例 :WPARAMETER1:TREND:ASCALE

:WPARameter<x>:TREND:CURSOr?

機能 波形パラメータ測定の特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。

構文 :WPARameter<x>:TREND:CURSOr?
<x> = 1、2

:WPARameter<x>:TREND:CURSOr:C<y>?

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ます。

構文 :WPARameter<x>:TREND:CURSOr:C<y>?
<x> = 1、2
<y> = 1、2

例 :WPARAMETER1:TREND:CURSOr:C1?
-> :WPARAMETER1:TREND:CURSOr:
C1 1.000E+00

:WPARameter<x>:TREND: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値を問い合わせます。

構文 :WPARameter<x>:TREND:CURSOr:DC?
<x> = 1、2

例 :WPARAMETER1:TREND:CURSOr:DC?
-> :WPARAMETER1:TREND:CURSOr:
DC 1.000E+00

:WPARameter<x>:TREND:CURSOr:MODE

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

構文 :WPARameter<x>:TREND:CURSOr:
MODE {<Boolean>}
:WPARameter<x>:TREND:CURSOr:MODE?
<x> = 1、2

例 :WPARAMETER1:TREND:CURSOr:MODE ON
:WPARAMETER1:TREND:CURSOr:MODE?
-> :WPARAMETER1:TREND:CURSOr:MODE 1

:WPARameter<x>:TREND:CURSOr:POSition<y>

機能 波形パラメータ測定の特レンドの各カーソルの位置を設定 / 問い合わせします。

構文 :WPARameter<x>:TREND:CURSOr:
POSition<y> {<NRF>}
:WPARameter<x>:TREND:CURSOr:
POSition<y>?

<x> = 1、2
<y> = 1、2
<NRF> = -5 ~ 5(div)
例 :WPARAMETER1:TREND:CURSOr:
POSITION1 1
:WPARAMETER1:TREND:CURSOr:POSITION1?
-> :WPARAMETER1:TREND:CURSOr:
POSITION1 1.000E+00

:WPARameter<x>:TREND:HRANge

機能 波形パラメータ測定の特レ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 :WPARameter<x>:TREND:HRANge {MAIN|
Z1|Z2}
:WPARameter<x>:TREND:HRANge?
<x> = 1、2

例 :WPARAMETER1:TREND:HRANge MAIN
:WPARAMETER1:TREND:HRANge?
-> :WPARAMETER1:TREND:HRANge MAIN

:WPARameter<x>:TREND:HSPan

機能 波形パラメータ測定の特レンド表示の水平スパンを設定 / 問い合わせします。

構文 :WPARameter<x>:TREND:HSPan {<NRF>}
:WPARameter<x>:TREND:HSPan?
<x> = 1、2
<NRF> = 1 ~ 100000

例 :WPARAMETER1:TREND:HSPAN 1
:WPARAMETER1:TREND:HSPAN?
-> :WPARAMETER1:TREND:HSPAN 1

5.36 WPARameter グループ

:WPARameter<x>:TREND:VERTical

機能 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。

構文 :WPARameter<x>:TREND:
VERTical {<NRf>,<NRf>}
:WPARameter<x>:TREND:VERTical?
<x> = 1, 2

例 <NRf> = -1.0000E+31 ~ 1.0000E+31
:WPARAMETER1:TREND:VERTICAL 1
:WPARAMETER1:TREND:VERTICAL?
-> :WPARAMETER1:TREND:
VERTICAL 1.00000E+00

:WPARameter<x>: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 :WPARameter<x>:VTDisplay {<Boolean>}
:WPARameter<x>:VTDisplay?
<x> = 1, 2

例 :WPARAMETER1:VTDISPLAY ON
:WPARAMETER1:VTDISPLAY?
-> :WPARAMETER1:VTDISPLAY 1

:WPARameter<x>: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 :WPARameter<x>:WAIT? {<NRf>}
<x> = 1, 2
<NRf> = 1 ~ 36000(タイムアウト時間、10ms)

例 :WPARAMETER1:WAIT?
-> :WPARAMETER1: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。

5.37 XY グループ

:XY<x>?

機能 XY 表示に関するすべての設定値を問い合わせます。

構文 :XY<x>?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:XY<x>:DISPlay

機能 XY 表示を画面に表示する (ON)/ しない (OFF) を設定 / 問い合わせします。

構文 :XY<x>:DISPlay {<Boolean>}
:XY<x>:DISPlay?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :XY1:DISPlay ON
:XY1:DISPlay?
-> :XY1:DISPlay 1

: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 :XY<x>:MEASure?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:XY<x>:MEASure:CURSor?

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 :XY<x>:MEASure:CURSor?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:XY<x>:MEASure:CURSor:DX?

機能 X-Y 表示の水平カーソル間の電圧差に関するすべての設定値を問い合わせます。

構文 :XY<x>:MEASure:CURSor:DX?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:XY<x>:MEASure:CURSor:DX:STATe

機能 X-Y 表示の水平カーソル間の電圧差の ON/OFF を設定 / 問い合わせします。

構文 :XY<x>:MEASure:CURSor:DX:
STATe {<Boolean>}
:XY<x>:MEASure:CURSor:DX:STATe?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
例 :XY1:MEASURE:CURSOR:DX:STATE ON
:XY1:MEASURE:CURSOR:DX:STATE?
-> :XY1:MEASURE:CURSOR:DX:STATE 1

:XY<x>:MEASure:CURSor:DX:VALue?

機能 X-Y 表示の水平カーソル間の電圧差を問い合わせます。

構文 :XY<x>:MEASure:CURSor:DX:VALue?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :XY1:MEASURE:CURSOR:DX:VALUE?
-> :XY1:MEASURE:CURSOR:DX:
VALUE 6.000E+00

:XY<x>:MEASure:CURSor:DY?

機能 X-Y 表示の垂直カーソル間の電圧差に関するすべての設定値を問い合わせます。

構文 :XY<x>:MEASure:CURSor:DY?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:XY<x>:MEASure:CURSor:DY:STATe

機能 X-Y 表示の垂直カーソル間の電圧差の ON/OFF を設定 / 問い合わせします。

構文 :XY<x>:MEASure:CURSor:DY:
STATe {<Boolean>}
:XY<x>:MEASure:CURSor:DY:STATe?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :XY1:MEASURE:CURSOR:DY:STATE ON
:XY1:MEASURE:CURSOR:DY:STATE?
-> :XY1:MEASURE:CURSOR:DY:STATE 1

:XY<x>:MEASure:CURSor:DY:VALue?

機能 X-Y 表示の垂直カーソル間の電圧差を問い合わせます。

構文 :XY<x>:MEASure:CURSor:DY:VALue?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :XY1:MEASURE:CURSOR:DY:VALUE?
-> :XY1:MEASURE:CURSOR:DY:
VALUE 6.000E+00

:XY<x>:MEASure:CURSor:X<y>?

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 :XY<x>:MEASure:CURSor:X<y>?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<y> = 1, 2

5.37 XY グループ

:XY<x>:MEASure:CURSor:X<y>:POSition

機能 XY 表示の水平カーソルの位置を設定 / 問い合わせます。

構文 :XY<x>:MEASure:CURSor:X<y>:
POSition {<NRf>}
:XY<x>:MEASure:CURSor:X<y>:POSition?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<y> = 1, 2
<NRf> = -4 ~ 4(div)

例 :XY1:MEASURE:CURSOR:X1:POSITION 1
:XY1:MEASURE:CURSOR:X1:POSITION?
-> :XY1:MEASURE:CURSOR:X1:
POSITION 1.000E+00

:XY<x>:MEASure:CURSor:X<y>:STATe

機能 X-Y 表示の水平カーソルの電圧値の ON/OFF を設定 / 問い合わせします。

構文 :XY<x>:MEASure:CURSor:X<y>:
STATe {<Boolean>}
:XY<x>:MEASure:CURSor:X<y>:STATe?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<y> = 1, 2

例 :XY1:MEASURE:CURSOR:X1:STATE ON
:XY1:MEASURE:CURSOR:X1:STATE?
-> :XY1:MEASURE:CURSOR:X1:STATE 1

:XY<x>:MEASure:CURSor:X<y>: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 :XY<x>:MEASure:CURSor:X<y>:VALue?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<y> = 1, 2

例 :XY1:MEASURE:CURSOR:X1:VALUE?
-> :XY1:MEASURE:CURSOR:X1:
VALUE 1.000E+00

:XY<x>:MEASure:CURSor:Y<y>?

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 :XY<x>:MEASure:CURSor:Y<y>?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<y> = 1, 2

:XY<x>:MEASure:CURSor:Y<y>:POSition

機能 XY 表示の垂直カーソルの位置を設定 / 問い合わせます。

構文 :XY<x>:MEASure:CURSor:Y<y>:
POSition {<NRf>}
:XY<x>:MEASure:CURSor:Y<y>:POSition?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<y> = 1, 2
<NRf> = -4 ~ 4(div)

例 :XY1:MEASURE:CURSOR:Y1:POSITION 1
:XY1:MEASURE:CURSOR:Y1:POSITION?
-> :XY1:MEASURE:CURSOR:Y1:
POSITION 1.000E+00

:XY<x>:MEASure:CURSor:Y<y>:STATe

機能 X-Y 表示の垂直カーソルの電圧値の ON/OFF を設定 / 問い合わせします。

構文 :XY<x>:MEASure:CURSor:Y<y>:
STATe {<Boolean>}
:XY<x>:MEASure:CURSor:Y<y>:STATe?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<y> = 1, 2

例 :XY1:MEASURE:CURSOR:Y1:STATE ON
:XY1:MEASURE:CURSOR:Y1:STATE?
-> :XY1:MEASURE:CURSOR:Y1:STATE 1

:XY<x>:MEASure:CURSor:Y<y>: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 :XY<x>:MEASure:CURSor:Y<y>:VALue?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)
<y> = 1, 2

例 :XY1:MEASURE:CURSOR:Y1:VALUE?
-> :XY1:MEASURE:CURSOR:Y1:
VALUE 1.000E+00

:XY<x>:MEASure:INTeg?

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 :XY<x>:MEASure:INTeg?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

:XY<x>:MEASure:INTeg:LOOP

機能 XY 表示の積分方法を設定 / 問い合わせます。

構文 :XY<x>:MEASure:INTeg:LOOP {CLOSE|
OPEN}
:XY<x>:MEASure:INTeg:LOOP?
<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :XY1:MEASURE:INTEG:LOOP CLOSE
:XY1:MEASURE:INTEG:LOOP?
-> :XY1:MEASURE:INTEG:LOOP CLOSE

:XY<x>:MEASure:INTEg:POLarity

機能 XY 表示の積分方向を設定 / 問い合わせます。

構文 :XY<x>:MEASure:INTEg:POLarity {CCW|CW}
:XY<x>:MEASure:INTEg:POLarity?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :XY1:MEASURE:INTEG:POLARITY CCW

:XY1:MEASURE:INTEG:POLARITY?

-> :XY1:MEASURE:INTEG:POLARITY CCW

解説 「:XY<x>:MEASURE:INTEG:LOOP」が「CLOSE」のときのみ有効な設定です。

:XY<x>:MEASure:INTEg:VALue?

機能 XY 表示の積分値を問い合わせます。

構文 :XY<x>:MEASure:INTEg:VALue?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :XY1:MEASURE:INTEG:VALUE?

-> :XY1:MEASURE:INTEG:

VALUE 1.000E+00

:XY<x>:MEASure:MODE

機能 XY 表示の自動測定モードを設定 / 問い合わせます。

構文 :XY<x>:MEASure:MODE {CURSor|INTEg|OFF}

:XY<x>:MEASure:MODE?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :XY1:MEASURE:MODE CURSOR

:XY1:MEASURE:MODE?

-> :XY1:MEASURE:MODE CURSOR

:XY<x>:RANGe

機能 X-Y 表示する T-Y 波形の対象ウインドウを設定 / 問い合わせします。

構文 :XY<x>:RANGe {MAIN|Z1|Z2}

:XY<x>:RANGe?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :XY1:RANGE MAIN

:XY1:RANGE?

-> :XY1:RANGE MAIN

:XY<x>:SPLit

機能 XY 表示で XY1/XY2 の表示画面を分割する / しないを、設定 / 問い合わせます。

構文 :XY<x>:SPLit {<Boolean>}

:XY<x>:SPLit?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :XY1:SPLIT ON

:XY1:SPLIT?

-> :XY1:SPLIT 1

:XY<x>:TRANge (Time Range)

機能 XY 表示する T-Y 波形の範囲を設定 / 問い合わせします。

構文 :XY<x>:TRANge {<NRF>,<NRF>}

:XY<x>:TRANge?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

<NRF>,<NRF> = -5 ~ 5(div)、(10div/ 表示レコード長) ステップ

例 :XY1:TRANge -4,4

:XY1:TRANge?

-> :XY1:TRANge -4.00,4.00

:XY<x>:VTDispay

機能 XY 表示の VT 波形の表示の ON/OFF を設定 / 問い合わせします。

構文 :XY<x>:VTDispay {<Boolean>}

:XY<x>:VTDispay?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

例 :XY1:VTDISPLAY ON

:XY1:VTDISPLAY?

-> :XY1:VTDISPLAY 1

:XY<x>:XTRace

機能 XY 表示の X 軸に割り当てるチャンネルを設定 / 問い合わせします。

構文 :XY<x>:XTRace {<NRF>|MATH<y>}

:XY<x>:XTRace?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

<NRF> = 1 ~ 8(4ch モデルでは 1 ~ 4)

MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :XY1:XTRACE 1

:XY1:XTRACE?

-> :XY1:XTRACE 1

解説 • XY<x> = 1 ~ 2 のときは、<NRF> = 1 ~ 4、MATH<y> = 1 ~ 4 を設定できます。

• XY<x> = 3 ~ 4 のときは、<NRF> = 5 ~ 8、MATH<y> = 5 ~ 8 を設定できます。

:XY<x>:YTRace

機能 XY 表示の Y 軸に割り当てるチャンネルを設定 / 問い合わせします。

構文 :XY<x>:YTRace {<NRF>|MATH<y>}

:XY<x>:YTRace?

<x> = 1 ~ 4(4ch モデルでは 1 ~ 2)

<NRF> = 1 ~ 8(4ch モデルでは 1 ~ 4)

MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :XY1:YTRACE 1

:XY1:YTRACE?

-> :XY1:YTRACE 1

解説 • XY<x> = 1 ~ 2 のときは、<NRF> = 1 ~ 4、MATH<y> = 1 ~ 4 を設定できます。

• XY<x> = 3 ~ 4 のときは、<NRF> = 5 ~ 8、MATH<y> = 5 ~ 8 を設定できます。

5.38 ZOOM グループ

: ZOOM<x>?

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<x>?
<x> = 1、2

: ZOOM<x>:ALLocation?

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<x>:ALLocation?
<x> = 1、2

: ZOOM<x>:ALLocation:{CHANnel<y>|LOGic|MATH<y>}?

機能 ズーム対象波形を設定 / 問い合わせします。

構文 :ZOOM<x>:ALLocation:{CHANnel<y>|LOGic|MATH<y>} {<Boolean>}
:ZOOM<x>:ALLocation:{CHANnel<y>|LOGic|MATH<y>}?
<x> = 1、2
<y> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :ZOOM1:ALLOCATION:CHANNEL1 ON
:ZOOM1:ALLOCATION:CHANNEL1?
-> :ZOOM1:ALLOCATION:CHANNEL1 1

: ZOOM<x>:DISPlay

機能 ズーム波形表示の ON/OFF を設定 / 問い合わせします。

構文 :ZOOM<x>:DISPlay {<Boolean>}
:ZOOM<x>:DISPlay?
<x> = 1、2

例 :ZOOM1:DISPLAY ON
:ZOOM1:DISPLAY?
-> :ZOOM1:DISPLAY 1

: ZOOM<x>:FORMat

機能 ズーム波形表示フォーマットを設定 / 問い合わせします。

構文 :ZOOM<x>:FORMat {DUAL|HEXa|MAIN|OCTal|QUAD|SINGle|TRIad}
:ZOOM<x>:FORMat?
<x> = 1、2

例 :ZOOM1:FORMAT DUAL
:ZOOM1:FORMAT?
-> :ZOOM1:FORMAT DUAL

: ZOOM<x>:MAG

機能 対象ズーム波形表示画面の倍率を設定 / 問い合わせします。

構文 :ZOOM<x>:MAG {<NRf>}
:ZOOM<x>:MAG?
<x> = 1、2
<NRf> = 本体ユーザーズマニュアル機能編参照。

例 :ZOOM1:MAG 2.5
:ZOOM1:MAG?
-> :ZOOM1:MAG 2.5

: ZOOM<x>:MAGFine

機能 対象ズーム波形表示画面の倍率 (Fine) を設定 / 問い合わせします。

構文 :ZOOM<x>:MAGFine {<NRf>}
:ZOOM<x>:MAGFine?
<x> = 1、2
<NRf> = 本体ユーザーズマニュアル機能編参照。

例 :ZOOM1:MAGFINE 2.5
:ZOOM1:MAGFINE?
-> :ZOOM1:MAGFINE 2.5

: ZOOM<x>:MAIN

機能 ズーム波形表示の表示比率を設定 / 問い合わせします。

構文 :ZOOM<x>:MAIN {20|50|OFF}
:ZOOM<x>:MAIN?
<x> = 1、2

例 :ZOOM1:MAIN 20
:ZOOM1:MAIN?
-> :ZOOM1:MAIN 20

: ZOOM<x>:POSition

機能 対象ズームボックスの位置を設定 / 問い合わせします。

構文 :ZOOM<x>:POSition {<NRf>}
:ZOOM<x>:POSition?
<x> = 1、2
<NRf> = -5 ~ 5(div)、(10div/ 表示レコード長) ステップ

例 :ZOOM1:POSITION 2
:ZOOM1:POSITION?
-> :ZOOM1:POSITION 2.00E+00

: ZOOM<x>:VERTical?

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<x>:VERTical?
<x> = 1、2

:ZOOM<x>:VERTical:{CHANnel<y>|MATH<y>}?

機能 垂直軸方向のズーム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 :ZOOM<x>:VERTical:{CHANnel<y>|
MATH<y>}?
<x> = 1、2
<y> = 1 ~ 8(4ch モデルでは 1 ~ 4)

:ZOOM<x>:VERTical:{CHANnel<y>|MATH<y>} :MAG

機能 垂直軸方向のズーム倍率を設定 / 問い合わせします。

構文 :ZOOM<x>:VERTical:{CHANnel<y>|
MATH<y>}:MAG {<NRf>}
:ZOOM<x>:VERTical:{CHANnel<y>|
MATH<y>}:MAG?
<x> = 1、2
<y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = 本体ユーザーズマニュアル機能編参照。

例 :ZOOM1:VERTICAL:CHANNEL1:MAG 2.5
:ZOOM1:VERTICAL:CHANNEL1:MAG?
-> :ZOOM1:VERTICAL:CHANNEL1:MAG 2.5

:ZOOM<x>:VERTical:{CHANnel<y>|MATH<y>} :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<x>:VERTical:{CHANnel<y>|
MATH<y>}:POSITION {<NRf>}
:ZOOM<x>:VERTical:{CHANnel<y>|
MATH<y>}:POSITION?
<x> = 1、2
<y> = 1 ~ 8(4ch モデルでは 1 ~ 4)
<NRf> = -4 ~ 4(div)

例 :ZOOM1:VERTICAL:CHANNEL1:POSITION 2
:ZOOM1:VERTICAL:CHANNEL1:POSITION?
-> :ZOOM1:VERTICAL:CHANNEL1:
POSITION 2.00E+00

:ZOOM<x>:VERTical:RESet

機能 垂直軸方向のズームに関するすべての設定値をリセットします。

構文 :ZOOM<x>:VERTical:RESet
<x> = 1、2

例 :ZOOM1:VERTICAL:RESET

:ZOOM<x>:VERTical:TRACe

機能 垂直軸方向のズーム画面に表示するトレースを設定 / 問い合わせします。

構文 :ZOOM<x>:VERTical:TRACe {<NRf>|
LOGic|MATH<y>}
:ZOOM<x>:VERTical:TRACe?
<x> = 1、2
<NRf> = 1 ~ 8(4ch モデルでは 1 ~ 4)
MATH<y> の <y> = 1 ~ 8(4ch モデルでは 1 ~ 4)

例 :ZOOM1:VERTICAL:TRACE 1
:ZOOM1:VERTICAL:TRACE?
-> :ZOOM1:VERTICAL:TRACE 1

5.39 共通コマンドグループ

共通コマンドグループは、USBTCM-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ネータ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」が起こって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,710120,08ELP1024,F1.01

解説 < 製造者 >、< 機種 >、< シリアル No.>、< ファームウェアのバージョン > の形式で返されます。< 機種 > は、DLM5034HD の場合は「DLM5034HD」、DLM5038HD の場合は「DLM5038HD」、DLM5054HD の場合は「DLM5054HD」、DLM5058HD の場合は「DLM5058HD」、が返ります。

*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 *OPC

例 *OPC

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。

***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	*OPC?
例	*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	*OPT?
例	*OPT? -> CH125MW,PRINTER,LOGIC,GPIB, STORAGE,I2C,CAN,LIN,SPIUART, PROBEPOWER4
解説	・ 本機器が装備している各種オプション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。「*OPT?」のあとにほかのクエリがあるときは、エラーになります。

***RST (ReSeT)**

機能	設定の初期化（イニシャライズ）をします。
構文	*RST
例	*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 175 -> 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、無視されます。 ・ 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	*STB?
例	*STB? -> 4
解説	・ 各ビットの和が 10 進数で返されます。 ・ シリアルボ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、6-2 ページを参照してください。

5.39 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、内部の各メモリテストです。
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。

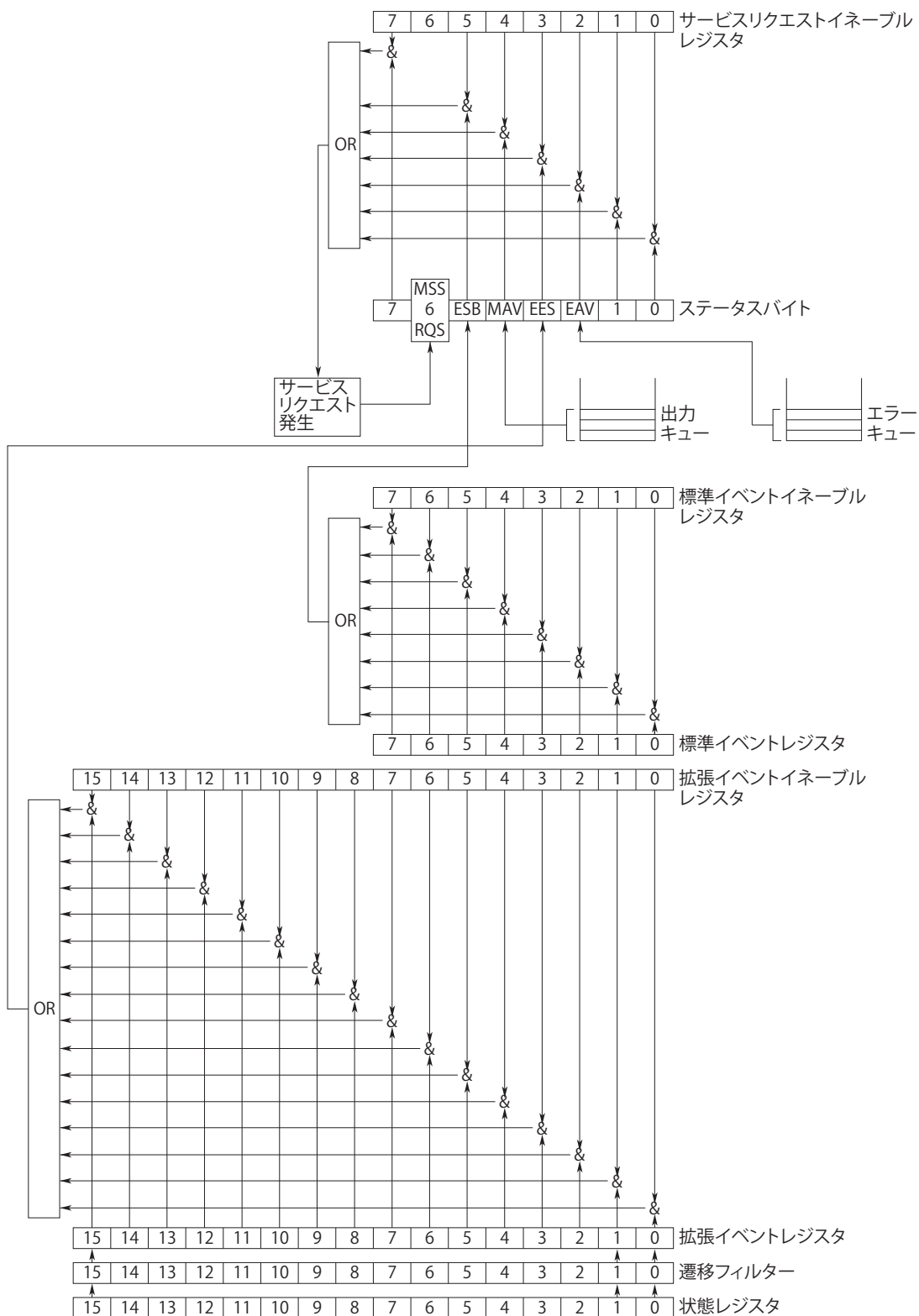
***WAI (WAIt)**

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、4-8 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。

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。



各レジスタ・キューの概要

名称	機能	書き込み	読み出し
ステータスバイト	—	—	シリアルポール (RQS), *STB? (MSS)
サービスリクエストイネーブルレジスタ	ステータスバイトのマスク	*SRE	*SRE?
標準イベントレジスタ	機器の状態の変化	—	*ESR?
標準イベントイネーブルレジスタ	標準イベントレジスタのマスク	*ESE	*ESE?
拡張イベントレジスタ	機器の状態の変化	—	STATus:ESR?
拡張イベントイネーブルレジスタ	拡張イベントレジスタのマスク	STATus:EESE	STATus:EESE?
状態レジスタ	現在の機器の状態	—	STATus:CONDition?
遷移フィルター	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。

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各ビットをマスク
標準イベントレジスタ	標準イベントイネーブルレジスタにより、各ビットをマスク
拡張イベントレジスタ	拡張イベントイネーブルレジスタにより、各ビットをマスク

各レジスタの書き込み / 読み出し

たとえば、標準イベントイネーブルレジスタの各ビットを 1 または 0 にするには、*ESE コマンドを使います。また、標準イベントイネーブルレジスタの各ビットが 1 であるか 0 であるかを確認するには、*ESE? コマンドを使います。これらの各コマンドについては、第 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

	RQS						
7	6	ESB	MAV	EES	EAV	1	0
	MSS						

- **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和が 1 になったときは、ビット 5(ESB) が 1 にセットされます。このとき、サービスリクエストイネーブルレジスタのビット 5 が 1 であれば、ビット 6(MSS) が 1 にセットされ、コントローラにサービスを要求します。

また、ステータスバイトの内容を読むことにより、どんな種類のイベントが起こったのかを確認することができます。

ステータスバイトの読み出し

ステータスバイトの内容を読み出すには、次の 2 つの方法があります。

- ***STB? による問い合わせ**
*STB? で問い合わせると、ビット 6 は MSS になります。したがって、MSS を読み出すことになります。読み出したあとは、ステータスバイトのどのビットもクリアしません。
- **シリアルポール**
シリアルポールを実行すると、ビット 6 は RQS になります。したがって、RQS を読み出すことになります。読み出したあと、RQS だけをクリアします。シリアルポールでは MSS を読み出すことはできません。

ステータスバイトのクリア

ステータスバイトの全ビットを強制的にクリアする方法はありません。各動作に対してクリアされるビットを以下に示します。

- ***STB? で問い合わせたとき**
どのビットもクリアされません。
- **シリアルポールを実行したとき**
RQS ビットだけがクリアされます。
- ***CLS コマンドを受信したとき**
*CLS コマンドを受信すると、ステータスバイト自体はクリアされませんが、各ビットに影響する標準イベントレジスタなどの内容がクリアされます。その結果、それに対応したステータスバイトのビットがクリアされます。ただし、出力キューは *CLS コマンドではクリアできないので、ステータスバイトのビット 4(MAV) は影響を受けません。ただし、*CLS コマンドをプログラムメッセージターミネータのすぐあとに受信したときは、出力キューもクリアされます。

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) 電源オン**
本機器の電源がオンになったときに、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。

	FILTer<x>→															
状態レジスタ :STATus:CONDition?	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	AN	0	NSG	SCH	NGO	0	HST	MES	ACS	PRN	TST	CAL	TRG	CUR	RUN
遷移フィルタ :STATus:FILTer<x> {RISE FALL BOTH NEVer}	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
拡張イベントレジスタ :STATus:EESR?	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

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 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)	波形検索実行中に 1 になります (シリアルバス信号の検索実行中も含む)。
ビット 12	NSG(N-Single)	トリガモードが N シングルのときの連続アクイジション中に 1 になります。
ビット 14	AN(Analysis)	解析実行中に 1 になります (シリアルバス信号の解析実行中も含む)。

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

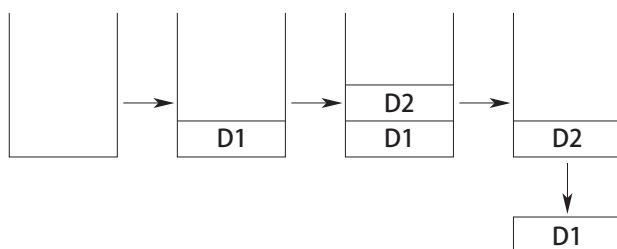
出力キュー

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。

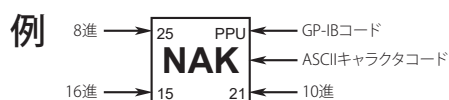
- ・ `*CLS` コマンドを受信したとき
- ・ 電源の再投入

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 キャラクタコード

ここでは、ASCII のキャラクタ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	リスナー 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、PC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 DLM5058HD-03JA 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) 300～399
 - ・ 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
100	Command error.	コマンドのつづりを確認してください。	5 章
102	構文が間違っています。 Syntax error.	通信文法エラーコード (100 ～ 199) 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。	3-7
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.	ニーモニック (アルファベットと数 1 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

コード	メッセージ	対処方法	ページ
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.	{...} 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
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。	4-6、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> の右の括弧がありません。 Missing Right.	演算式は使用できません。	—
172	<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
181	プレースホルダがマクロの外にあります。 Invalid outside macro definition.	IEEE488.2 のマクロ機能には対応していません。	—

通信実行エラー (200 ~ 299)

Error in communication execution

コード	メッセージ	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
225	オーバーフロー OverFlow.	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
226	データの格納領域が足りません。 Out Of Memory.	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
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。	—
270	マクロのネストが深すぎます。 Macro error.	IEEE488.2 のマクロ機能には対応していません。	—
272	マクロでは使用できません。 Macro execution error.	IEEE488.2 のマクロ機能には対応していません。	—
273	マクロラベルが不適当です。 Illegal macro label.	IEEE488.2 のマクロ機能には対応していません。	—

付録2 エラーメッセージ

コード	メッセージ	対処方法	ページ
275	マクロが長すぎます。 Macro definition too long.	IEEE488.2 のマクロ機能には対応していません。	—
276	マクロが再帰呼び出しされました。 Macro recursion error.	IEEE488.2 のマクロ機能には対応していません。	—
277	マクロの二重定義はできません。 Macro redefinition not allowed.	IEEE488.2 のマクロ機能には対応していません。	—
278	そのようなマクロは定義されていません。 Macro header not found.	IEEE488.2 のマクロ機能には対応していません。	—

通信クエリエラー (400 ~ 499)

Error in communication Query

コード	メッセージ	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) (300、399)

Error in System Operation

コード	メッセージ	対処方法	ページ
300	通信デバイスエラー。 Communication device-specific error.	サービスが必要です。	—
399	通信ドライバエラー。 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、390)

コード	メッセージ	対処方法	ページ
350	エラーキューがオーバーフローを起こしました。 Queue overflow.	エラーキューを読み出してください。	6-6
390	オーバーランエラーが発生しました。 Communication overrun error.	ボーレートを下げて実行してください。	—

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 IEEE 488.2-1992 について

本機器の GP-IB インタフェースは、IEEE 488.2-1992 規格に準じています。この規格では、以下の 23 の項目について「ドキュメントに記載しなければならない」としています。ここでは、これらについて説明しています。

- (1) **IEEE 488.1 インタフェース機能のうち、サポートしているサブセット**
「3.2 GP-IB インタフェースの機能と仕様」を参照してください。
- (2) **アドレスが 0 ～ 30 以外に設定されたときのデバイスの動作**
本機器では、アドレスを 0 ～ 30 以外に設定することはできません。
- (3) **ユーザーがアドレス変更をしたときの動作**
アドレスの変更は UTILITY キー → Remote Control メニューでアドレスを設定した時点で行われます。設定したアドレスは、次に変更するまで有効です。
- (4) **電源オン時のデバイスのセッティング。電源オン時に使用可能なコマンド**
基本的には、以前の設定（その前に電源をオフにしたときの設定）になります。電源オフ時に実行を制限されるコマンドはありません。
- (5) **メッセージ交換のオプション**
 - (a) **入力バッファのサイズ**
16384 バイト
 - (b) **複数の応答メッセージユニットを返すクエリ**
5 章の各コマンドの例を参照してください。
 - (c) **構文解析時に応答データを作成するクエリ**
すべてのクエリは、構文を解析すると応答データを作成します。
 - (d) **受信時に応答データを作成するクエリ**
コントローラが受信する時点で応答データを作成するクエリはありません。
 - (e) **制限しあうパラメータを有するコマンド**
:CHANnel<x>:PROBe[:MODE] と
:CHANnel<x>:VDIV のように一方的に制限を与えるものはありますが、相互に制限を与えるものはありません。
- (6) **コマンドを構成する機能エレメントおよび複合ヘッダのエレメントに含まれるもの**
4 章および 5 章を参照してください。
- (7) **ブロックデータの転送に影響するバッファのサイズ**
ブロックデータの送信時には、そのサイズに合わせて出力キューを拡張します。
- (8) **演算式で使えるプログラムデータのエレメントの一覧と、そのネストの制限**
演算式は使えません。
- (9) **各問い合わせに対する応答の構文**
5 章の各コマンドの例を参照してください。
- (10) **応答の文法に従わないデバイス間の通信について**
本機器では、サポートしていません。
- (11) **応答データのブロックデータのサイズ**
1 ～ 999999998
- (12) **サポートしている共通コマンドの一覧**
「5.39 共通コマンドグループ」を参照してください。
- (13) **キャリブレーション正常終了時のデバイスの状態**
測定を実行中の状態になります。
- (14) ***DDT のトリガマクロの定義で利用できるブロックデータの最大長**
サポートしていません。
- (15) **マクロ定義のマクロラベルの最大長、マクロ定義で利用できるブロックデータの最大長、マクロ定義で再帰を使ったときの処理**
マクロ機能は対応していません。
- (16) ***IDN? に対する返送**
「5.39 共通コマンドグループ」を参照してください。
- (17) ***PUD、*PUD? のプロテクトユーザーデータの保存エリアのサイズ**
*PUD、*PUD? はサポートしていません。
- (18) ***RDT、*RDT? のリソース名の長さ**
*RDT、*RDT? はサポートしていません。
- (19) ***RST、*LRN?、*RCL、*SAV による状態の変化**
*RST
「5.39 共通コマンドグループ」を参照してください。
*RCL、*SAV、*LRN?
これらの共通コマンドはサポートしていません。
- (20) ***TST? によるセルフテストの実行範囲**
UTILITY の Self Test メニューの MEMORY テストのすべて（内部の各メモリー）を実行します。
- (21) **拡張されたリターンステータスの構造**
6 章を参照してください。
- (22) **各コマンドの処理がオーバーラップするか、シーケンシャルに行われるか**
「4.5 コントローラとの同期」および 5 章を参照してください。
- (23) **各コマンドの実行内容**
5 章の各コマンドの機能とユーザーズマニュアル IMDLM5058HD-01JA/IMDLM5058HD-02JA を参照してください。

索引

記号	ページ
<Boolean>.....	4-7
<NRf>.....	4-6, 4-7
< 時間 >.....	4-6
< 周波数 >.....	4-6
< 電圧 >.....	4-6
< 電流 >.....	4-6
Δ D 値.....	5-90
Δ T カーソル.....	5-95
Δ V カーソル.....	5-92
Δ V 値.....	5-90

A	ページ
A->B(N) トリガ.....	5-261
ACQuire グループ.....	5-60
A Delay B トリガ.....	5-262
ANALysis グループ.....	5-61
ASCII のキャラクタコード.....	付 -1
ASETup グループ.....	5-81

C	ページ
Calc アイテム.....	5-152
CALibrate グループ.....	5-82
CAN FD バストリガ.....	5-270
CAN FD バスの解析.....	5-171
CAN FD バスの検索.....	5-173
CAN バストリガ.....	5-263
CAN バスの解析.....	5-163
CAN バスの検索.....	5-165
CHANnel グループ.....	5-83
CHUtil グループ.....	5-86
CLEar グループ.....	5-87
COMMunicate グループ.....	5-88
CURSor グループ.....	5-90
CXPI バストリガ.....	5-279
CXPI バスの解析.....	5-180
CXPI バスの検索.....	5-182

D	ページ
DCL.....	3-6
DISPlay グループ.....	5-98
DLM シリーズ用ライブラリ.....	ii

F	ページ
FFT 解析.....	5-101
FFT 解析結果リスト.....	5-105
FFT グループ.....	5-101
FFT 点数.....	5-102
FILE グループ.....	5-109
FlexRay バストリガ.....	5-286
FlexRay バスの解析.....	5-190

G	ページ
GONogo グループ.....	5-113
GO/NO-GO 判定	
警告音.....	5-113
ゾーン.....	5-114
ハードコピー.....	5-113
波形ゾーン.....	5-117
パラメータ.....	5-114

判定終了.....	5-114
方形ゾーン.....	5-116
保存.....	5-113
ポリゴンゾーン判定.....	5-115
メール通知.....	5-113
モード.....	5-114
GO/NO-GO 判定.....	5-113
GP-IB インタフェースの機能.....	3-2
GTL.....	3-6

H	ページ
HCOPy グループ.....	5-119
HISTory グループ.....	5-121

I	ページ
I2C バストリガ.....	5-293
I2C バスの解析.....	5-198
I2C バスの検索.....	5-199
IEEE 488.2-1992 の 23 項目.....	付 -5
IEEE 1588 時刻同期.....	5-258
IFC.....	3-6
IMAGe グループ.....	5-127
INITialize グループ.....	5-129

L	ページ
LIN バストリガ.....	5-299
LIN バスの解析.....	5-205
LIN バスの検索.....	5-207
LLO.....	3-6
LOGic グループ.....	5-130

M	ページ
MATH グループ.....	5-137
MEASure グループ.....	5-143

N	ページ
NO-GO 時の動作.....	5-113

P	ページ
PSI5 トリガ.....	5-305
PSI5 の解析.....	5-211
PSI5 の検索.....	5-214

R	ページ
RECall グループ.....	5-154
REFeRence グループ.....	5-155
REN.....	3-6

S	ページ
SDC.....	3-6
SEARch グループ.....	5-156
SENT トリガ.....	5-310
SENT の解析.....	5-219
SERialbus グループ.....	5-162
SNAP グループ.....	5-249
SNTP.....	5-255
SPD.....	3-6
SPE.....	3-6
SPI バストリガ.....	5-325

索引

SPI バスの解析	5-239
SPI バスの検索	5-241
SStart グループ	5-250
STARt グループ	5-251
STATus グループ	5-252
STOP グループ	5-253
STORe グループ	5-254
SYSTem グループ	5-255

T	ページ
TCP/IP	2-6
Timebase グループ	5-260
Time/div 値	5-260
TMCTL	ii
TRIGger グループ	5-261
TV トリガ	5-333

U	ページ
UART トリガ	5-328
UART の解析	5-243
UART の検索	5-245
USB インタフェース	1-2
USB キーボードの種類	5-259

W	ページ
WAVeform グループ	5-338
Web サイト	ii
WPARameter グループ	5-341

X	ページ
XY グループ	5-345
XY 表示	5-345

Z	ページ
ZOOM グループ	5-348

ア	ページ
明るさ	5-256
アキュムレート	5-98
空き容量	5-109
アクイジションモード	5-60, 5-340
アクションオントリガ	5-261
圧縮保存	5-111
アベレージング	5-60
アベレージ回数	5-60
減衰定数	5-60
指数化平均	5-60
単純平均	5-60
アベレージング (FFT の)	5-101
安全動作領域	5-68

イ	ページ
イーサネットインタフェース	2-2
移動 (ファイルの)	5-110
移動平均	5-138
イニシャライズ	5-129, 5-351
色	5-98
インターバル (エッジ間隔) トリガ	5-331
インタフェースメッセージ	3-7
インバート	5-83

ウ	ページ
ウィンドウ OR トリガ	5-331
ウィンドウトリガ	5-330

エ	ページ
液晶画面	5-256
エッジ OR トリガ	5-303
エッジカウント	5-137
エッジ検索	5-156
エッジトリガ	5-321
エラーキュー	6-6
エラーキューの先頭の問い合わせ	5-252
エラーのコードの問い合わせ	5-252
エラーメッセージ	付-2
演算子	5-139
演算 (波形)	5-137
エンベロープモード	5-60

オ	ページ
応答	4-5
応答メッセージ	4-1
オートオフ	5-256
オートキャリブレーション	5-82
オートスケュー補正	5-66, 5-77
オートスクロール	5-156
オートスケール	5-83
オートセットアップ	5-81, 5-162
オーバーラップコマンド	5-88
オプションの問い合わせ	5-351
オフセット	5-84
オフセット電圧の反映	5-257

カ	ページ
カーソル (XY 表示の)	5-345
カーソル測定	5-90
カーソルの種類	5-95
解析	
CAN FD バス	5-171
CAN バス	5-163
CXPI バス	5-180
FlexRay バス	5-190
I2C バス	5-198
LIN バス	5-205
SENT	5-219
SPI バス	5-239
UART	5-243
電源	5-63
ユーザー定義バス	5-235
拡張イベントイネーブルレジスタの設定	5-252
拡張イベントレジスタ	6-5
角度カーソル	5-90
位置	5-91
角度差	5-90
基準角度	5-91
単位	5-91
電圧	5-91
Δ D 値	5-90
Δ V 値	5-90
角度差	5-90
各部の名称	1-1, 2-1, 3-1
画面イメージ	5-127
外部プリンタへの出力	5-119
色調	5-128
出力形式	5-127
内蔵プリンタへの出力	5-120
ネットワークプリンタへの出力	5-120
バックグラウンド (透過)	5-127
ファイル名	5-127
保存	5-127

索引	
キ	ページ
機種.....	5-350
基準角度.....	5-91
輝度.....	5-99
キャリブレーション.....	5-82, 5-350
強制トリガ.....	5-335
共通コマンド.....	4-3
共通コマンドグループ.....	5-350
ク	ページ
クエリ.....	4-1, 4-4
グラティクル.....	5-99
クリアトレース.....	5-87
クリア (レジスタの).....	5-350
クリック音.....	5-255
グリニッジ.....	5-255
グループ.....	4-3
ケ	ページ
警告音.....	5-113
検索.....	5-156
CAN FD バス信号.....	5-173
CAN バス信号.....	5-165
CXPI バス信号.....	5-182
FlexRay バス信号.....	5-192
I2C バス信号.....	5-199
LIN バス信号.....	5-207
SENT 信号.....	5-223
SPI バス信号.....	5-241
UART 信号.....	5-245
エッジ.....	5-156
タイムアウト.....	5-156
パルス幅.....	5-159
ヒストリ波形.....	5-121
複数入力の組み合わせ (パターン).....	5-157
ユーザー定義バス信号.....	5-238
検索タイプの設定.....	5-161, 5-243
減衰定数.....	5-60
減衰比.....	5-84
コ	ページ
高調波解析.....	5-63
高分解能モード.....	5-60
構文の記号.....	v
コピー (ファイルの).....	5-109
サ	ページ
サービスリクエストイネーブルレジスタの値の設定.....	5-351
サイクル統計処理.....	5-148
最小レコード番号.....	5-340
サイドパネル.....	3-1
削除 (ファイルの).....	5-109
サンプリングモード.....	5-60
サンプルプログラム.....	ii
サンプルレート.....	5-260, 5-340
シ	ページ
時刻.....	5-255
指数化平均.....	5-60
システム.....	5-255
自動測定.....	5-143
自動測定 (ロジックの).....	5-149
周期モード.....	5-145
ジュール積分.....	5-65
出力キュー.....	6-6
上位クエリ.....	4-4
状態レジスタ.....	6-5
状態レジスタの問い合わせ.....	5-252
省略形.....	4-5
初期化.....	5-351
シリアルバス信号の解析 / 検索.....	5-162
シリアルポートの実行.....	5-252
シングルスタート.....	5-250
ス	ページ
垂直軸.....	5-83
垂直ポジション.....	5-84
スイッチング損失.....	5-69
ズーム.....	5-348
ズーム位置の連動.....	5-248
ズーム波形表示フォーマット.....	5-348
ズームポジション.....	5-349
ズームボックス.....	5-348
スキップモード.....	5-160
スキュー補正.....	5-67, 5-77, 5-83, 5-132
スケーリング (演算).....	5-140
スケーリング係数.....	5-84
スケール値表示.....	5-100
スタート.....	5-251
ステータス.....	5-252
ステータスバイト.....	6-3
ステータスバイトレジスタの値の問い合わせ.....	5-351
ステータスレポート.....	6-1
ストップ.....	5-253
ストレージ.....	5-109
スナップショット.....	5-249
スペクトラム.....	5-107
セ	ページ
成立回数 (トリガの).....	5-336
積分 (MATH の).....	5-138
積分 (XY 表示の).....	5-346
設定コピー.....	5-86
設定の初期化.....	5-351
セルフテスト.....	5-352
ゼロ補正.....	5-85
遷移フィルター.....	5-252, 6-5
ソ	ページ
ゾーン判定.....	5-115
測定アイテム.....	5-143
測定値の文字の大きさ.....	5-257
測定範囲 (自動測定の).....	5-152
ソケットインタフェース.....	2-3
タ	ページ
タイムアウト検索.....	5-161
タイムアウト時間.....	2-3
タイムアウトトリガ.....	5-333
タイムベース.....	5-260
立ち上がり / 立ち下がり時間トリガ.....	5-332
チ	ページ
チャンネル間ディレイ.....	5-145
チャンネル間のコピー.....	5-86
ツ	ページ
通信に関する設定.....	5-88
通信のステータス.....	5-252
通信ライブラリ.....	ii

索引

テ	ページ
ディスタル	5-69, 5-74, 5-146
ディレイ (チャネル間の)	5-145
ディレイ (トリガの)	5-262, 5-335
ディレクトリ	5-109
データ	4-6
データストレージ	5-109
データ点数	5-101
データ転送速度	1-2, 2-2, 3-3
デッドロック状態	4-2
電圧軸感度	5-85
電源解析	5-63
電源解析の入力	5-66
電力測定	5-73
電力量	5-73, 5-79
ト	ページ
同期運転	5-256
統計処理 (サイクルの)	5-148
統計処理 (自動測定の)	5-143
統計処理 (通常の)	5-147
統計処理 (ヒストリ波形の)	5-148
トータル損失	5-69
ドライバ	ii
トリガ	5-261
A->B(N)	5-261
A Delay B	5-262
CAN FD バス	5-270
CAN バス	5-263
CXPI バス	5-279
FlexRay バス	5-286
I2C バス	5-293
LIN バス	5-299
SENT	5-310
SPI バス	5-325
TV	5-333
UART 信号	5-328
アクションオントリガ	5-261
インターバルトリガ (エッジ間隔)	5-331
ウインドウ	5-330
ウインドウ OR	5-330
エッジ	5-321
エッジ OR	5-303
強制	5-335
組み合わせ	5-335
タイムアウト	5-333
立ち上がり / 立ち下がり時間	5-332
ディレイ	5-335
パルス幅	5-331
複数入力の組み合わせ (パターン)	5-304
ホールドオフ	5-336
ユーザー定義 TV	5-334
ユーザー定義バス	5-322
ラント	5-332
トリガ出力	5-258
トリガ条件	5-263
トリガ成立回数	5-336
トリガタイプ	5-261
トリガディレイ	5-335
トリガの組み合わせ	5-335
トリガの種類の設定	5-327
トリガポジション	5-336
トリガモード	5-336
トリガレベル	5-336
トレンドの色	5-98
トレンド (波形パラメータの)	5-343

ニ	ページ
入力カップリング	5-83
入力フィルター	5-83

ノ	ページ
ノーマルモード	5-60

ハ	ページ
ハードコピー	5-119
倍率	5-348
ハイレゾリレーションモード	5-60
波形演算	5-137
波形ゾーン判定	5-117
波形データのフォーマット	5-339
波形の検索	5-156
波形の取り込み	5-60
波形の取り込みのスタート	5-251
波形の取り込みのストップ	5-253
波形のヒストグラム表示	5-61
波形パラメータ	5-143
波形パラメータ測定	5-341
波形パラメータのヒストグラム表示	5-341
波形パラメータ (ロジックの)	5-149
波形ラベル	5-83
バックライト	5-256
パラメータ (自動測定)	5-143
パルス幅検索	5-159
パルス幅トリガ	5-331
反転表示	5-83

ヒ	ページ
ビープ音	5-255
ヒストグラム表示 (波形の)	5-61
ヒストグラム表示 (波形パラメータの)	5-341
ヒストリ	5-121
ヒストリ波形の検索	5-121
ヒストリ波形のユーザー定義演算	5-142
日付	5-255
表記法	v
表示色	5-98
表示に関する設定	5-98
表示フォーマット	5-99
標準イベントイネーブルレジスタの値	5-350
標準イベントレジスタ	6-4

フ	ページ
ファイル	
移動	5-110
コピー	5-109
削除	5-109
保存	5-110
呼び出し	5-110
リネーム	5-110
ファイングリッド	5-99
フィルター	5-83, 5-137
フォーマット (ズーム波形表示の)	5-348
フォーマット (波形データの)	5-339
フォーマット (表示の)	5-99
フォーマット (分割の)	5-99
複数入力の組み合わせ (パターン) 検索	5-157
複数入力の組み合わせ (パターン) トリガ	5-304
符号の有無	5-340
物理値変換	5-339
プリンタ	5-120
プローブの減衰比	5-84
プロキシマル	5-69, 5-74, 5-146

プログラムメッセージ	4-1
ブロックデータ	4-7
プロテクト	5-110
フロントパネル	1-1, 2-1, 3-1
分割フォーマット	5-99
へ	ページ
平均	5-60
→アベレージング も参照	
ヘッダ	4-5
ホ	ページ
ホールドオフ	5-336
ホールドオフ検出	5-160
補間	5-99
保存 (ストア)	5-254
保存 (ファイルの)	5-110
ポリゴンゾーン判定	5-115
本機器のマニュアル	iii
マ	ページ
マーカーカーソル	5-93, 5-103
窓関数	5-108
マニュアルで使用している記号	v
マニュアルの構成	iii
マルチターゲット	5-119
マルチラインメッセージ	3-7
メ	ページ
命令	4-3
メシアル	5-69, 5-74, 5-146
メッセージ	4-1
メッセージ (エラーキューの先頭) の問い合わせ	5-252
メッセージ言語	5-256
メディア	5-109
メニュー言語	5-256
目盛り	5-99
モ	ページ
文字データ	4-7
文字の表記	v
文字列データ	4-7
ユ	ページ
ユーザー定義 TV トリガ	5-334
ユーザー定義演算	5-141
ユーザー定義バストリガ	5-322
ユーザー定義バスの解析	5-235
ユーザー定義バスの検索	5-238
ユニラインメッセージ	3-7
ヨ	ページ
呼び出し (内部メモリー)	5-154
呼び出し (ファイルの)	5-110
ラ	ページ
ライブラリ	ii
ラベル名	5-83
ラントトリガ	5-332

リ	ページ
リアパネル	1-1, 2-1
リニアスケーリング	5-84
リネーム (ファイルの)	5-110
リファレンス波形	5-155
リモート	5-88
リモート / ローカル切り替え	1-3, 2-3, 3-3
レ	ページ
レコード長	5-60
レジスタクリア	5-350
ロ	ページ
ローカル	5-88
ローカルロックアウト	5-88
ロータリカウント	5-139
ロジック入力	5-130
ロックアウト	5-88