

**ファンクションジェネレータ**

**SG-4200 シリーズ**

**SG-4212/4208/4204**

**プログラミングガイド**

---

---

## 履 歴

◇2024年4月 第1版発行

KMLA01251

© 2024 IWATSU ELECTRIC CO.,LTD All rights reserved.

---

## 目次

1. プログラミングの概要.....	5
1.1. VISAによるコミュニケーション構築.....	5
1.1.1. NI-VISAのインストール.....	5
1.1.2. SG-4200を接続する.....	8
1.2. リモートコントロール.....	8
1.2.1. NI-MAXでSCPIを使用する.....	8
1.2.2. TelnetでSCPIを使用する.....	9
1.2.3. SCPI over Socketの使用.....	12
2. SCPI言語の紹介.....	13
2.1. コマンドとクエリについて.....	13
2.1.1. 掲載方法.....	13
2.1.2. 記述方法.....	13
2.1.3. 使用方法.....	13
2.1.4. コマンド表記.....	13
2.2. コマンドとクエリーの表.....	14
2.3. IEEE488.2 共通コマンド.....	15
2.3.1. *IDN.....	15
2.3.2. *OPC.....	15
2.3.3. *CLS.....	16
2.3.4. *ESE.....	16
2.3.5. *ESR.....	17
2.3.6. *RST.....	17
2.3.7. *SRE.....	17
2.3.8. *STB.....	18
2.3.9. *TST.....	18
2.3.10. *WAI.....	19
2.4. OUTP 出力コマンド.....	19
2.5. BSWV 基本波形コマンド.....	20
2.6. MDWV 変調波形コマンド.....	21
2.7. SWWV スイープ波形コマンド.....	24
2.8. BTWV バースト波形コマンド.....	26
2.9. PACP パラメータコピーコマンド.....	28
2.10. ARWV 任意波形コマンド.....	29
2.11. SYNC 同期コマンド.....	30
2.12. NBFM 数値フォーマットコマンド.....	30
2.13. SCFG コンフィグレーションコマンド.....	31
2.14. BUZZ ブザーコマンド.....	31
2.15. SCSV スクリーンセーバーコマンド.....	31
2.16. ROSC クロックソースコマンド.....	32
2.17. FCNT 周波数カウンタコマンド.....	32
2.18. INVT インバートコマンド.....	33
2.19. COUP カップリングコマンド.....	33
2.20. VOLTPRT 過電圧保護コマンド.....	34
2.21. STL 保存リストコマンド.....	35
2.22. VKEY 仮想キーコマンド.....	35
2.23. SYST:COMM:LAN:IPAD IP コマンド.....	36
2.24. SYST:COMM:LAN:SMAS サブネットマスクコマンド.....	37
2.25. SYST:COMM:LAN:GAT ゲートウェイコマンド.....	37
2.26. SRATE サンプリングレートコマンド.....	38

---

---

2.27. HARM ハーモニックコマンド.....	38
2.28. MODE モードコマンド.....	39
2.29. インデックス .....	39

---

# 1. プログラミングの概要

USB および LAN インタフェースを使用し、NI-VISA やプログラミング言語と組み合わせることで、ユーザーは SG-4200 をリモートで制御することができます。LAN インタフェースを介して、VXI-11、ソケット、および Telnet プロトコルを使用して、SG-4200 と通信することができます。この章では、SG-4200 と PC の間の通信を構築する方法を紹介します。また、SG-4200 のリモート・コントロールのためのシステムの構成方法についても紹介します。

## 1.1. VISA によるコミュニケーション構築

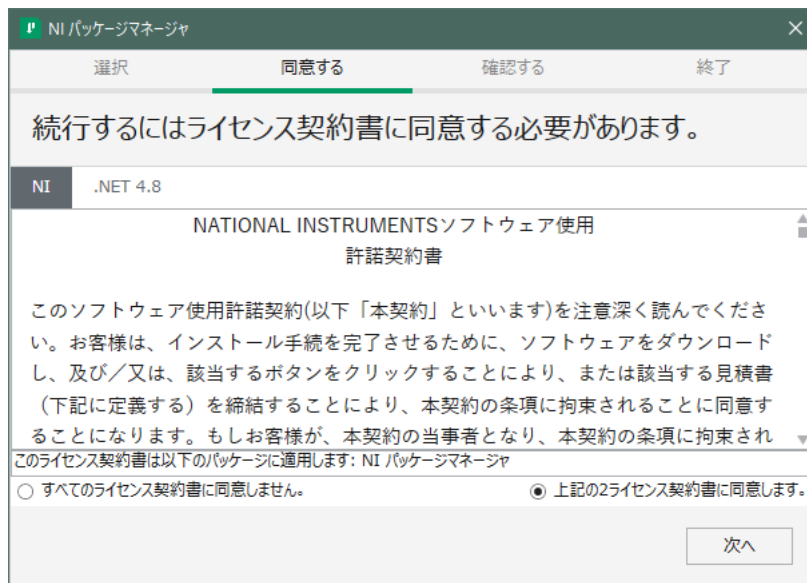
### 1.1.1. NI-VISA のインストール

プログラミングを行う前に、National Instruments NI-VISA Software が正しくインストールされていることを確認してください。

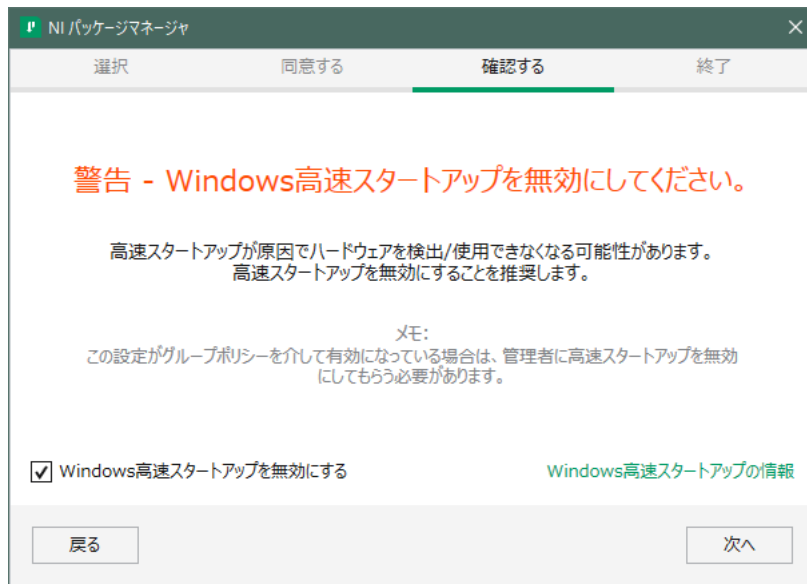
NI-VISA は、コンピュータから計測器への通信を可能にする通信ライブラリです。インストーラには、NI デバイスドライバと、デバイスを制御するためのユーザーインターフェイスである NI-MAX というツール等が含まれています。ドライバと NI MAX は便利ですが、リモートコントロールに必須ではありません。ランタイムエンジンだけでリモートコントロールは可能です。

以下の手順で NI-VISA をインストールします。

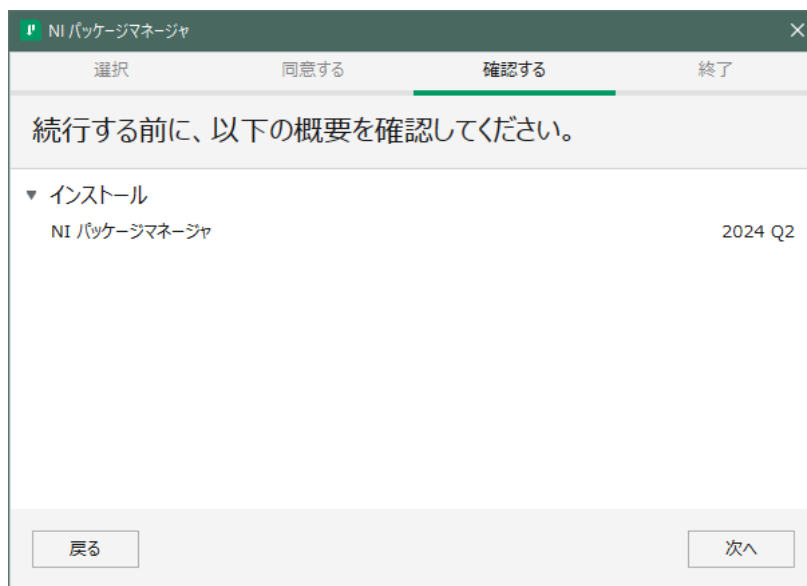
- a. NI-VISA をダウンロードします。(使用する OS に対応したバージョンを選択してください。この例では NI-VISA 2024Q1 を使用しています。)
- b. ダウンロードした ni-visa\_24.0\_online.exe を実行します。ライセンス契約書を読み、「上記の 2 ライセンス契約書に同意します。」を選択し、「次へ」をクリックします。



c. そのまま「次へ」をクリックします。



d. 「次へ」をクリックします。

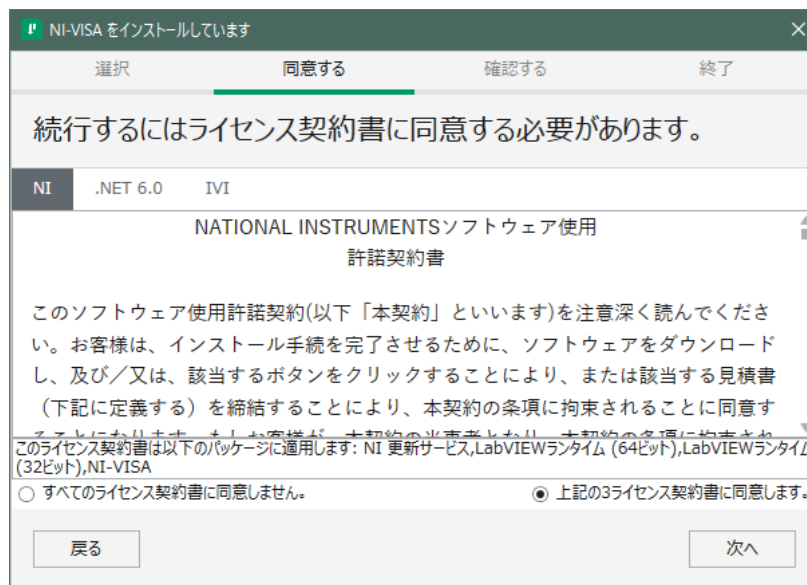


e. パッケージマネージャがインストールされた後、インストールする項目を選択します。そのまま「次へ」をクリックします。

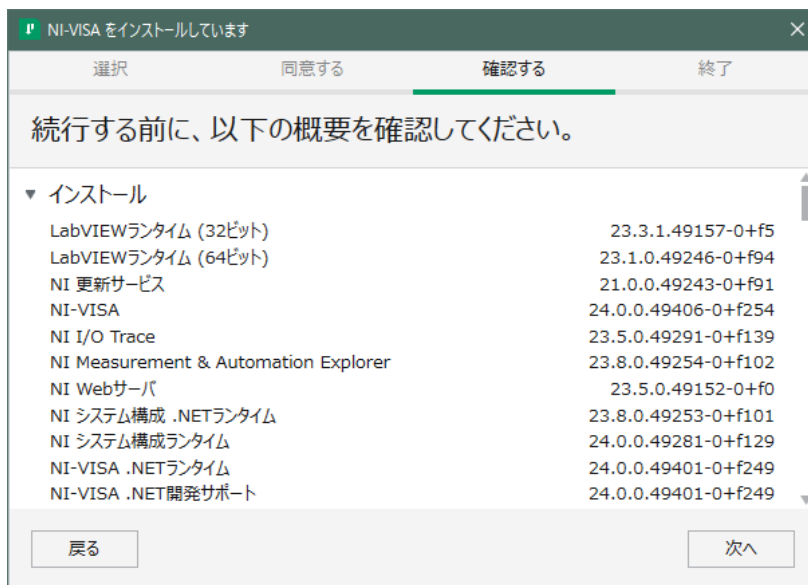
(ランタイムエンジンのみインストールする場合は、「すべて選択解除」をクリックしてから“NI-VISA .NET ランタイム”のみチェックを入れ、「次へ」をクリックします。)



f. ライセンス契約書を読み、「上記の3ライセンス契約書に同意します。」を選択し、「次へ」をクリックします。



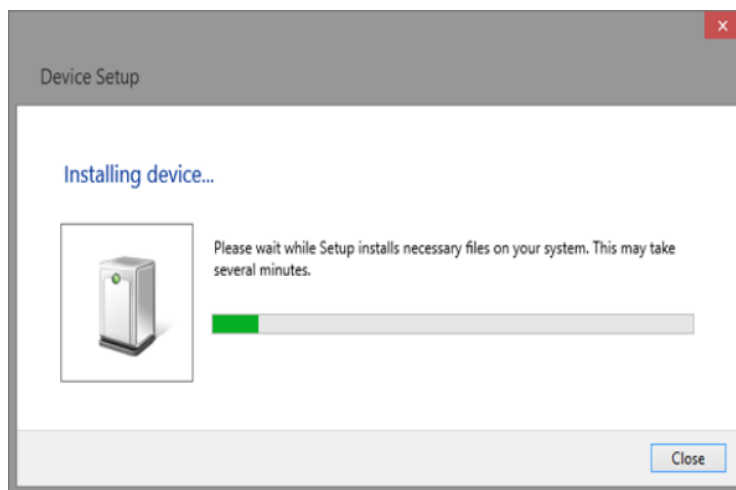
g. 次の画面で、「次へ」をクリックすると、インストールが開始されます。



h. これでインストールは完了です。PC を再起動してください。

### 1.1.2. SG-4200 を接続する

SG-4200 と PC の USB ホストインターフェイスを USB ケーブルで接続します。PC の電源は入っているものとし、SG-4200 の電源を入れると、PC に「デバイスセットアップ」画面が表示され、下図のようにデバイスドライバが自動的にインストールされます。



インストールが完了するのを待ち、次のステップに進みます。

## 1.2. リモートコントロール

### 1.2.1. NI-MAX で SCPI を使用する

NI-MAX はナショナルインスツルメンツ社によって作成、維持されているプログラムです。VXI、LAN、USB、GPIB、シリアル通信の基本的なリモートコントロールインターフェイスを提供します。NI-MAX 経由で SCPI コマンドを送信することにより、SG-4200 をリモートで制御することができます。



### 1.2.2. Telnet で SCPI を使用する

Telnet は、LAN 経由で SG-4200 と通信するための手段を提供します。Telnet プロトコルは、PC から SG-4200 に SCPI コマンドを送信し、USB 経由で SG-4200 と通信するのと似ています。一度に 1 つのコマンドを送信し、インタラクティブに情報を送受信します。Windows オペレーティングシステムでは、Telnet クライアントにコマンドプロンプトスタイルのインターフェイスを使用します。手順は次のとおりです。

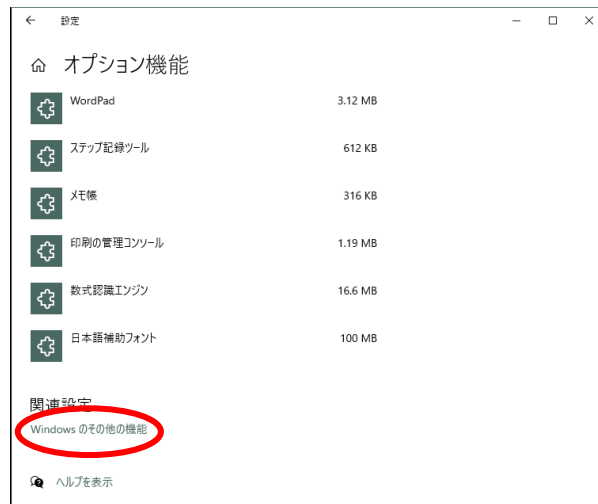
1. 最初に Telnet クライアントを有効にします。Windows の「設定」アプリから「アプリ」を選択します。



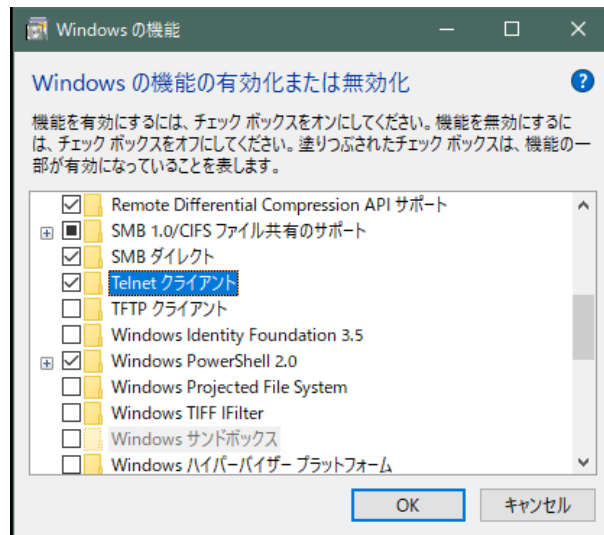
2. 「アプリと機能」から「オプション機能」をクリックします。



3. 下方へスクロールして、関連設定から「Windows のその他の機能」をクリックします。

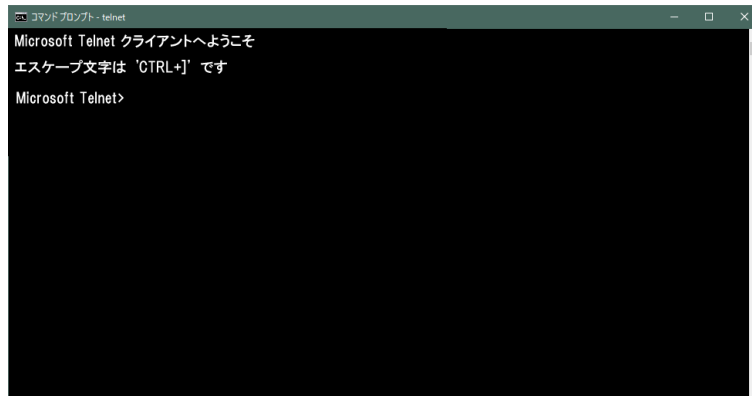


4. 「Windows の機能」から「Telnet クライアント」をチェックし、「OK」をクリックします。

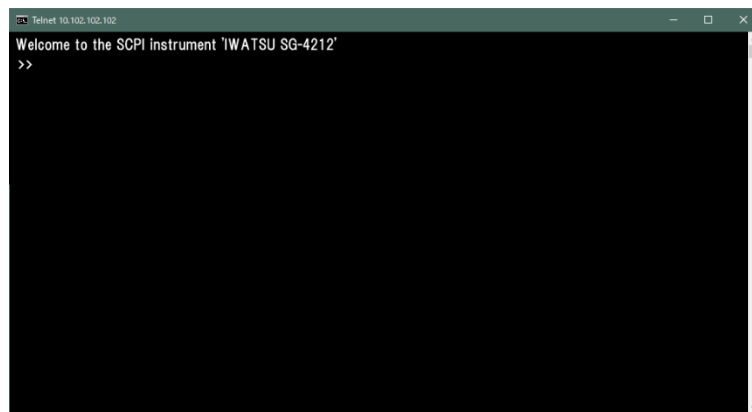


これで、Telnet クライアントが有効になりました。

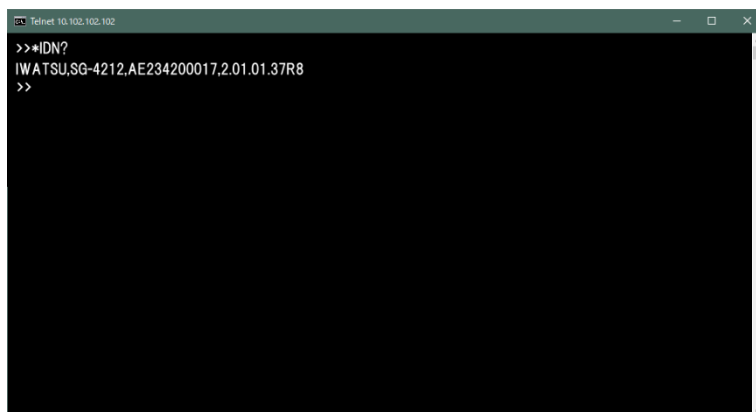
5. PC の「スタート」→「Windows システムツール」→「コマンドプロンプト」をクリックします。
6. コマンドプロンプトで、「telnet」と入力します。
7. Enter キーを押します。Telnet の表示画面が表示されます。



8. Telnet コマンドラインで、open XXX.XXX.XXX 5024 と入力します。ここで、XXX.XXX.XXX.XXX は SG-4200 の IP アドレス、5024 はポートです。次のような応答が表示されます。



9. \*IDN? などの SCPI コマンドを入力すると、メーカー名、モデル名、シリアル番号、ソフトウェア・バージョンが返されます。



10. セッションを終了するには、Ctrl+] キーを同時に押してください。
11. プロンプトで quit と入力するか、Telnet ウィンドウを閉じると、測定器との接続が終了し、Telnet が終了します。

---

### 1.2.3. SCPI over Socket の使用

Socket API を使用することで、他のライブラリをインストールすることなく、SG-4200 を LAN で制御することができます。これにより、プログラミングの複雑さを軽減することができます。

ソケットアドレス	IP アドレス+ポート番号
IP アドレス	SG-4200 の IP アドレス
ポート番号	5025

---

## 2. SCPI 言語の紹介

### 2.1. コマンドとクエリについて

このセクションでは、測定器が認識するリモートコントロールコマンドとクエリのリストと説明をします。すべてのコマンドとクエリは、ローカルまたはリモートのいずれかの状態で実行することができます。

各コマンドやクエリには、シンタックスやその他の情報があり、いくつかの例が掲載されています。コマンドは "COMMAND SYNTAX" で長い形式と短い形式の両方が示され、主語はコマンドかクエリかその両方であることが示されています。クエリは情報を得るなどの動作を行い、ヘッダーの後にクエスチョンマーク(?)が付いていることで認識されます。

#### 2.1.1. 掲載方法

説明文は、その短い形式に従ってアルファベット順に並んでいます。

#### 2.1.2. 記述方法

記述そのものは、実行される機能についての簡単な説明です。これに続いて、正式な構文が示され、ヘッダは大文字と小文字で、そこから派生した短縮形はすべて大文字で示されます。該当する場合、問い合わせの構文とその応答の形式が示されます。

#### 2.1.3. 使用方法

ここに掲載されているコマンドやクエリは、SG4200 シリーズの関数/任意波形発生器に使用することができます。

#### 2.1.4. コマンド表記

コマンドには次のような表記があります。

< > 角括弧はプレースホルダーとして使われる単語を囲むもので、ヘッダーパスとコマンドのデータパラメータの 2 種類あります。

:= コロンと等号は、プレースホルダーの代わりにコマンドで使用できる値の種類と範囲を区切ります。

{ } 中括弧は選択肢のリストを囲むもので、そのうちの 1 つは必ず選択する必要があります。

[ ] 角括弧は、オプションの項目を囲みます。

... 省略記号は、その左右の項目が何度でも繰り返される可能性があることを示す。

## 2.2. コマンドとクエリーの表

ショート	ロングフォーム	サブシステム	コマンド/クエリができること
<a href="#">*IDN</a>	*IDN	システム	デバイスから識別情報を取得します。
<a href="#">*OPC</a>	*OPC	システム	イベントステータスレジスタ (ESR) の OPC ビット (0) を取得または設定します。
<a href="#">*CLS</a>	*CLS	システム	全てのステータスデータレジスタをクリアします。
<a href="#">*ESE</a>	*ESE	システム	標準イベントステータスイネーブルレジスタ (ESE) を設定・取得します。
<a href="#">*ESR</a>	*ESR	システム	イベントステータスレジスタ (ESR) の内容を読み出し、クリアします。
<a href="#">*RST</a>	*RST	システム	デバイスのリセットを開始します。
<a href="#">*SRE</a>	*SRE	システム	サービスリクエストイネーブルレジスタ (SRE) を設定します。
<a href="#">*STB</a>	*STB	システム	IEEE488.2 で定義されたステータスレジスタの内容を取得します。
<a href="#">*TST</a>	*TST	システム	内部セルフテストを実施します。
<a href="#">*WAI</a>	*WAI	システム	コマンドの継続を待ちます。
<a href="#">OUTP</a>	OUTPUT	信号	出力状態を設定または取得します。
<a href="#">BSWV</a>	BASIC_WAVE	信号	基本波形を設定・取得します。
<a href="#">MDWV</a>	MODULATEWAVE	信号	変調波形を設定・取得します。
<a href="#">SWWV</a>	SWEEPWAVE	信号	スイープ波形を設定・取得します。
<a href="#">BTWV</a>	BURSTWAVE	信号	バースト波形を設定・取得します。
<a href="#">PACP</a>	PARACOPY	信号	パラメータをチャンネル間でコピーします。
<a href="#">ARWV</a>	ARBWAVE	データ	任意波形を変更します。
<a href="#">SYNC</a>	SYNC	信号	同期信号を設定・取得します。
<a href="#">NBFM</a>	NUMBER_FORMAT	システム	データフォーマットを設定・取得します。
<a href="#">SCFG</a>	SYS_CFG	システム	電源投入時のシステム設定値を設定・取得します。
<a href="#">BUZZ</a>	BUZZER	システム	ブザーの状態を設定・取得します。
<a href="#">SCSV</a>	SCREEN_SAVE	システム	画面の保存状態を設定・取得します。
<a href="#">ROSC</a>	ROSCILLATOR	信号	クロックソースの状態を設定・取得します。
<a href="#">FCNT</a>	FREQCOUNTER	信号	周波数カウンタのパラメータを設定・取得します。
<a href="#">INVT</a>	INVERT	信号	現在のチャンネルの極性を設定・取得します。
<a href="#">COUP</a>	COUPLING	信号	カップリングを設定・取得します。

<a href="#">VOLTPRT</a>	VOLTPRT	システム	過電圧の保護の状態設定・取得します。
<a href="#">STL</a>	STORELIST	信号	保存されているすべての波形を一覧表示します。
<a href="#">VKEY</a>	VIRTUALKEY	システム	仮想キーを設定します。
<a href="#">SYST:COMM:LAN:IPAD</a>	SYSTEM:COMMUNICATE:LAN:IPADDRESS	システム	システムの IP アドレスを設定・取得します。
<a href="#">SYST:COMM:LAN:SMASK</a>	SYSTEM:COMMUNICATE:LAN:SMASK	システム	システムのサブネットマスクを設定・取得します。
<a href="#">SYST:COMM:LAN:GATEWAY</a>	SYSTEM:COMMUNICATE:LAN:GATEWAY	システム	システムのゲートウェイを設定・取得します。
<a href="#">SRATE</a>	SAMPLERATE	信号	TrueArb モード時にサンプリングレートを設定・取得します。
<a href="#">HARM</a>	HARMONIC	信号	高調波パラメータを設定または取得します。
<a href="#">MODE</a>	MODE	システム	フェーズモードを設定または取得します。

## 2.3. IEEE488.2 共通コマンド

IEEE 規格では、測定器の基本情報の問い合わせや基本的な操作の実行に使用する共通コマンドを定義しています。これらのコマンドは通常 "\*" で始まり、コマンドのキーワードの長さは通常 3 文字です。

### 2.3.1. \*IDN

説明 IDN? クエリーは、測定器自身を識別するために使用されます。応答は、メーカー名、モデル名、シリアル番号、ソフトウェア・バージョンから構成されます。

クエリ構文 \*IDN?

応答フォーマット \*IDN, <device id>, <model>, <serial number>, <software version>.

<device id>:=メーカー名。  
 <model>:= モデル名。  
 <serial number>:=機器のシリアル番号。  
 <software version>:=ソフトウェアのバージョン。

実施例 バージョンを読み取ります。  
 戻り値：  
 IWATSU,SG-4212,AE234200027,2.01.01.37R8

### 2.3.2. \*OPC

---

説明	<p>OPC (オペレーションコンプリート) コマンドは、標準のイベントステータスレジスタ (ESR) の OPC ビット (ビット 0) を設定します。</p> <p>本機は、直前のコマンドまたはクエリを完全に処理した後のみ、コマンドまたはクエリの解析を開始するため、このコマンドは本機の動作に他の影響を与えません。</p> <p>OPC?クエリは、前のコマンドが完全に実行されたとき、常に ASCII 文字 1 で応答します。</p>
----	---

コマンド構文	*OPC
--------	------

クエリ構文	*OPC?
-------	-------

応答フォーマット	*OPC 1
----------	--------

関連コマンド	*WAI
--------	------

### 2.3.3. \*CLS

説明	<p>*CLS コマンドは、全てのステータスデータレジスタをクリアします。</p>
----	---

コマンド構文	*CLS
--------	------

実施例	<p>次のコマンドを実行すると、すべてのステータスデータレジスタがクリアされます。</p> <p>*CLS</p>
-----	---

関連コマンド	CMR、DDR、*ESR、*STB
--------	-------------------

### 2.3.4. \*ESE

説明	<p>*ESE コマンドは、標準イベントステータスイネーブルレジスタ (ESE) を設定します。このコマンドにより、ESR レジスタ内の 1 つまたは複数のイベントが STB レジスタの ESB サマリーメッセージビット (ビット 5) に反映されるようになります。</p> <p>*ESE? クエリは、ESE レジスタの内容を読み出します。</p>
----	---

コマンド構文	<p>*ESE &lt;value&gt; &lt;value&gt; := 0~255 とする。</p>
--------	---

クエリ構文	*ESE?
-------	-------

応答フォーマット	*ESE <value>
----------	--------------

実施例	<p>次の命令では、ユーザリクエスト (URQ ビット 6、すなわち 10 進数 64) および/またはデバイス依存エラー (DDE ビット 3、すなわち 10 進数 8) が発生した場合に ESB ビットを設定することができます。これらの値を合計すると、ESE レジスタのマスクは 64+8=72 となります。</p> <p>*ESE?</p>
-----	---

---



---

	戻り値 :
	ESE 72
関連コマンド	*ESR
<b>2.3.5. *ESR</b>	
説明	*ESR?クエリは、イベント・ステータス・レジスタ (ESR) の内容を読み取り、クリアするものです。レスポンスは、レジスタのビット 0 から 7 までのバイナリ値の合計を表します。
クエリ構文	*ESR?
応答フォーマット	*ESR <value> <value> := 0~255
実施例	次の命令では、ESR レジスタの内容を読み出し、クリアします *ESR? 戻り値 : *ESR 0
<b>2.3.6. *RST</b>	
説明	*RST コマンドは、デバイスのリセットを開始します。RST は、全てのトレースを GND ラインに設定し、デフォルトのセットアップをリコールします
コマンド構文	*RST
実施例	この例では、信号発生器をリセットしています。 *RST
<b>2.3.7. *SRE</b>	
説明	*SRE コマンドは、サービスリクエストイネーブルレジスタ (SRE)を設定します。このコマンドにより、STB レジスタ内のどのサマリーメッセージビットがサービスリクエストを生成するかが指定されます。 サマリーメッセージビットは、対応するビットの場所に'1'を書き込むことで有効になります。逆に、'0'を書き込むと、関連するイベントがサービスリクエスト(SRQ)を生成しないようにします。SRE レジスタをクリアすると SRQ 割り込みが禁止されます。 *SRE?クエリは、SRE レジスタのビット設定を 2 進数に変換した値を返します。ビット 6 (MSS) は設定できないため、返される値は常に 0 であることに注意してください。
コマンド構文	*SRE <value> <value> := 0~255 とする。
クエリ構文	*SRE?

---

---

応答フォーマット	<b>*SRE &lt;value&gt;</b>
実施例	<p>次の命令では、STB レジスタの MAV サマリービット（ビット 4、すなわち 10 進数 16）または INB サマリービット（ビット 0、すなわち 10 進数 1）、あるいはその両方が設定されるとすぐに SRQ を生成することができるようにします。これら 2 つの値を合計すると、SRE マスク 16+1=17 となります。</p> <p><b>*SRE?</b> 戻り値 <b>*SRE 17</b></p>

### 2.3.8. \*STB

説明	<p><b>*STB?</b>クエリは、488.2 定義のステータスレジスタ (STB)、およびマスターサマリーステータス (MSS) の内容を読み取ります。</p> <p>応答は、ステータスバイトレジスタのビット 0~5 及び 7 の値と MSS サマリーメッセージを表します。</p> <p><b>*STB?</b>クエリの応答は、以下の応答と同じです。</p> <p>シリアルポールの場合、RQS メッセージの代わりに MSS サマリーメッセージがビット 6 に表示されることを除けば、RQS メッセージの代わりに MSS サマリーメッセージが表示されます。</p>
クエリ構文	<b>*STB?</b>
応答フォーマット	<p><b>*STB &lt;value&gt;</b> <b>&lt;value&gt; := 0~255</b></p>
実施例	<p>以下は、ステータスバイトレジスタの読み出しです。</p> <p><b>*STB?</b> 戻り値： <b>*STB 0</b></p>
関連コマンド	<b>*CLS、*SRE</b>

### 2.3.9. \*TST

説明	<p><b>*TST?</b>クエリは、内部のセルフ・テストを実行し、その応答として、セルフ・テストがエラーを検出したかどうかを示されます。セルフ・テストには、全チャンネルのハードウェア、タイムベース、トリガ回路のテストが含まれます。</p> <p>ハードウェアの故障は、返された&lt;status&gt;番号に含まれる一意のバイナリコードで識別されます。0 "応答は、障害が発生しなかったことを示します。</p>
クエリ構文	<b>*TST?</b>
応答フォーマット	<p><b>*TST&lt;status&gt;</b> <b>&lt;status&gt; := 0 セルフテスト成功</b></p>
実施例	以下のようにすると、セルフテストが実行されます。

---

---

\*TST?  
戻り値：(失敗しない場合)  
\*TST 0

関連コマンド \*CAL

### 2.3.10. \*WAI

説明 IEEE488.2 規格で要求される \*WAI (WAIT to continue) コマンドは、直前のコマンドが完全に実行されたときにのみコマンドの処理を開始するため、機器に影響を与えません。

コマンド構文 \*WAI

関連コマンド \*OPC

## 2.4. OUTP 出力コマンド

説明 チャンネルに対応する前面パネルの[Output]コネクタの出力を有効または無効にします。  
クエリでは、出力状態、負荷、極性パラメータの値が返されません。

コマンド構文 <チャンネル> : OUTP (出力) <パラメータ>  
<チャンネル>:= {C1, C2}  
<パラメータ >:= {下表のパラメータ}。

パラメータ	値	説明
ON	-	出力オン
OFF	-	出力オフ
LOAD	<load>	負荷
PLRT	<NOR,INVT>	極性

<load>:= {下記の注意事項をご覧ください。}

クエリ構文 <channel>: OUTP(OUTPut)?

応答フォーマット <channel>: OUTP <load>

実施例 チャンネル 1 をオンにします。  
C1: OUTP ON

チャンネル 1 の出力状態を読み出します。  
C1: OUTP?

戻り値：  
C1: OUTP ON, LOAD, HZ, PLRT, NOR

負荷を 50Ω に設定します。  
C1: OUTP LOAD, 50

負荷をハイインピーダンスに設定します。

C1: OUTP LOAD, HZ

極性を正に設定します。  
C1: OUTP PLRT, NOR

極性を負に設定します。  
C1: OUTP PLRT, INVT

注 <load>:=50~100000、HZ

## 2.5. BSWV 基本波形コマンド

説明 基本波形のパラメータを設定または取得します。

コマンド構文 <channel>:BSWV(BaSic\_WaVe) <parameter>  
<channel>:={C1, C2}  
<parameter>:= {下表のパラメータ}

パラメータ	値	説明
WVTP	<type>	波形の種類。
FRQ	<frequency>	周波数。波形の種類がノイズまたは DC の場合、このパラメータを設定することはできません。
PERI	<period>	周期。波形の種類がノイズまたは DC の場合、このパラメータを設定することはできません。
AMP	<amplitude>	振幅。波形の種類がノイズまたは DC の場合、このパラメータを設定することはできません。
OFST	<offset>	オフセット値。波形の種類がノイズまたは DC の場合、このパラメータを設定することはできません。
SYM	<symmetry>	シンメトリ。波形の種類がランプの時のみ、このパラメータを設定することができます。
DUTY	<duty>	デューティサイクル。波形の種類が方形波でパルスの時のみ、このパラメータを設定することができます。
PHSE	<phase>	位相。波形の種類がノイズまたはパルスまたは DC の場合、このパラメータを設定することはできません。
STDEV	<stdev>	ノイズ波形の分散値。波形の種類がノイズの時のみ、このパラメータを設定することができます。
MEAN	<mean>	ノイズ波形の平均値。波形の種類がノイズの時のみ、このパラメータを設定することができます。
WIDTH	<width>	パルス幅。波形の種類がパルスの時のみ、このパラメータを設定することができます。
RISE	<rise>	立ち上がり時間。波形の種類がパルスの時のみ、このパラメータを設定することができます。
FALL	<fall>	立下り時間。波形の種類がパルスの時のみ、このパラメータを設定することができます。
DLY	<delay>	ディレイ値。波形の種類がパルスの時のみ、このパラメータを設定することができます。

HLEV	<high level>	ハイレベル値。波形の種類がノイズまたは DC の場合、このパラメータを設定することはできません。
LLEV	<low level>	ロウレベル値。波形の種類がノイズまたは DC の場合、このパラメータを設定することはできません。

注 : コマンドで基本波形を設定しない場合、WVPT パラメータは現在の波形を設定します。

ここで <type> := {SINE、SQUARE、RAMP、PULSE、NOISE、ARB、DC}  
 <frequency> := {デフォルトの単位は "Hz" です。値は機種に依存します。}  
 <amplitude> := {デフォルトの単位は "V" です。値は機種に依存します。}  
 <offset> := {デフォルトの単位は "V" です。値は機種に依存します。}  
 <duty> := {0%~100%。値は周波数に依存します。}  
 <symmetry> := {0%~100%}  
 <phase> := {値は機種によって異なります。}  
 <stdev> := {デフォルトの単位は "V" です。値は機種に依存します。}  
 <mean> := {デフォルトの単位は "V" です。値は機種に依存します。}  
 <width> := {Max\_width < (Max\_duty \* 0.01 \* period.and Min\_width > (Min\_duty \* 0.01 \* period.)} とします。}  
 <rise> := {値は機種に依存します。}  
 <fall> := {値は機種に依存します。}  
 <delay> := {単位は S、最大値はパルス周期、最小値は 0}

クエリ構文 <channel>: BSWV (BaSic\_WaVe)?  
 <channel>:= {C1, C2}

応答フォーマット <channel>:BSWV<type>,<frequency>,<amplitude>,<offset>,<duty>,<symmetry>,<phase>,<variance>,<mean>,<width>,<rise>,<fall>,<delay>.

実施例 チャンネル 1 の波形をランプに変更します。  
 C1: BSWV WVTP, RAMP

チャンネル 1 の周波数を 2000Hz に変更します。  
 C1 : BSWV FRQ, 2000

チャンネル 1 の振幅を 3Vpp に設定します。  
 C1 : BSWV AMP, 3

チャンネル 1 の基本波形パラメータを読み出します。  
 C1 : BSWV?

戻り値 :  
 C1: BSWV WVTP, SINE,FRQ,100HZ,PERI,0.01S,AMP,2V,OFST,0V,HLEV,1V,LLEV,-1V,PHSE,0

関連コマンド ARWV、BTWV、MDWV、SWWV

## 2.6. MDWV 変調波形コマンド

説明 変調波形のパラメータを設定または取得します。

コマンド構文 <channel>:MDWV(MoDulateWaVe)<parameter>

<channel>:={C1, C2}  
 <parameter>:= {下表のパラメータ}

パラメータ	値	説明
STATE	<state>	変調をオンまたはオフにします。注：変調の他のパラメータを設定または読み取る場合は、最初に STATE を ON に設定する必要があります。
AM, SRC	<src>	AM 信号のソース。
AM, MDSP	<mod wave shape>	AM 変調の波形。変調ソースに INT(内部) を設定した場合のみ、設定可能です。
AM, FRQ	<AM frequency>	AM 変調の周波数。変調ソースに INT (内部) を設定した場合のみ、設定可能です。
AM, DEPTH	<depth>	AM 変調の変調度。変調ソースに INT (内部) を設定した場合のみ、設定可能です。
DSBAM, SRC	<src>	DSBAM 信号のソース。
DSBAM, MDSP	<mod wave shape>	DSBAM 変調の波形。変調ソースに INT(内部) を設定した場合のみ、設定可能です。
DSBAM, FRQ	<DSB-AM frequency>	DSBAM 変調の周波数。変調ソースに INT(内部) を設定した場合のみ、設定可能です。
FM, SRC	<src>	FM 信号のソース。
FM, MDSP	<mod wave shape>	FM 変調の波形。変調ソースに INT(内部) を設定した場合のみ、設定可能です。
FM, FRQ	<FM frequency>	FM 変調の周波数。変調ソースに INT (内部) を設定した場合のみ、設定可能です。
FM, DEVI	<FM frequency deviation >	FM 変調の周波数偏差。変調ソースに INT (内部) を設定した場合のみ、設定可能です。
PM, SRC	<src>	PM 信号のソース。
PM, MDSP	<mod wave shape>	PM 変調の波形。変調ソースに INT(内部) を設定した場合のみ、設定可能です。
PM, FRQ	<PM frequency>	PM 変調の周波数。変調ソースに INT (内部) を設定した場合のみ、設定可能です。
PWM, FRQ	<PWM frequency>	PWM 変調の周波数。キャリアがパルスの場合のみ、設定可能です。
PWM, DEVI	<PWM dev>	デューティサイクル偏差値。キャリアがパルスの場合のみ、設定可能です。
PWM, MDSP	<mod wave shape>	PWM 変調の波形。キャリアがパルスの場合のみ、設定可能です。
PWM, SRC	<src>	PWM 信号のソース。
PM, DEVI	<PM phase offset>	FM 変調の周波数偏差。キャリアがパルスの場合のみ、設定可能です。
ASK, SRC	<src>	ASK 信号のソース。
ASK, KFRQ	<ASK key frequency>	ASK 変調のキー周波数。変調ソースに INT (内部) を設定した場合のみ、設定可能です。
FSK, KFRQ	<FSK frequency>	FSK 変調のキー周波数。変調ソースに INT (内部) を設定した場合のみ、設定可能です。
FSK, HFRQ	<FSK hop frequency>	FSK 変調のホップ周波数。
FSK, SRC	<src>	FSK 信号のソース。

CARR, WVTP	<wave type>	キャリア波形。
CARR, FRQ	<frequency>	キャリア周波数。
CARR, AMP	<amplitude>	キャリアの振幅。
CARR, OFST	<offset>	キャリアのオフセット値。
CARR, SYM	<symmetry>	キャリアのシンメトリ値。パルスの場合のみ、設定可能です。
CARR, DUTY	<duty>	デューティサイクル値。方形波でパルスの場合のみ、設定可能です。
CARR, PHSE	<phase>	位相値。
CARR, RISE	<rise>	立ち上がり時間。パルスの場合のみ、設定可能です。
CARR, FALL	<fall>	立ち下がり時間。パルスの場合のみ、設定可能です。
CARR, DLY	<delay>	キャリア信号の遅延値。パルスの場合のみ、設定可能です。

注 キャリアがノイズの場合、変調をオンにする設定はできません。

AM、FM、PM、CARR、STATE を設定する場合、最初のパラメータはそれらのいずれかにする必要があります

ここで <state>:={ON、OFF}  
 <src>:={INT、EXT}  
 <mod wave shape>:={SINE、SQUARE、TRIANGLE、UP RAMP、DNRAMP、NOISE、ARB}  
 <am frequency>:={デフォルトの単位は"Hz"です。値は機種に依存します。}  
 <depth>:={0%~120%}  
 <fm frequency>:={デフォルトの単位は"Hz"です。値は機種に依存します。}  
 <fm frequency deviation>:={0~キャリア周波数,キャリア信号の周波数と帯域幅周波数の差に依存します。}  
 <pm frequency>:={デフォルトの単位は"Hz"です。値は機種に依存します。}  
 <pm phase deviation>:={0~360。}  
 <pwm frequency>:={デフォルトの単位は"Hz"です。値は機種に依存します。}  
 <pwm dev>:={デフォルトの単位は"%"です。値はキャリア信号のデューティサイクルに依存します。}  
 <ask key frequency>:={デフォルトの単位は"Hz"です。値は機種に依存します。}  
 <fsk frequency>:={デフォルトの単位は"Hz"です。値はバージョンに依存します。}  
 <fsk jump frequency>:={基本波の周波数と同じです。}  
 <wave type>:={SINE、SQUARE、RAMP、ARB、PULSE}。  
 <frequency>:={デフォルトの単位は"Hz"です。値は機種に依存します。}  
 <amplitude>:={デフォルトの単位は"V"です。値は機種に依存します。}  
 <offset>:={デフォルトの単位は"V"です。値は機種に依存します。}  
 <duty>:={0%~100%}。  
 <symmetry>:={0%~100%}  
 <phase>:={値は機種に依存します。}  
 <rise>:={値は機種に依存します。}  
 <fall>:={値は機種に依存します。}  
 <delay>:={デフォルトの単位は "S"です。}

注 パラメータは機種によって異なりますので、各機種の仕様をご参照ください。

クエリ構文

<channel>: MDWV (MoDulateWaVe)?

	<channel>:={C1, C2}
応答フォーマット	<channel>:MDWV <parameter> <parameter> := {現在の変調パラメータの全てを返します。}
実施例	<p>チャンネル 1 の変調方式を AM に設定します。 C1: MDWV AM</p> <p>変調を AM に、AM 変調波の種類を正弦波に設定します。 C1: MDWV AM, MDSP, SINE</p> <p>STATE が ON のチャンネル 1 の変調パラメータを読み出します。 C1: MDWV? 戻り値 : C1: MDWV STATE, ON, AM, MDSP, SINE, SRC, INT, FRQ, 100HZ, DEPTH, 100, CARR, WVTP, RAMP, FRQ, 1000HZ, AMP, 4V, OFST, 0V, PHSE, 0, SYM, 50</p> <p>STATE が OFF のチャンネル 1 の変調波パラメータを読み出します。 C1: MDWV? 戻り値 : C1: MDWV STATE, OFF</p> <p>チャンネル 1 の FM 周波数を 1000Hz に設定します。 C1: MDWV FM, FRQ, 1000</p> <p>チャンネル 1 のキャリア波形を SINE に設定します。 C1: MDWV CARR, WVTP, SINE</p> <p>チャンネル 1 のキャリア周波数を 1000Hz に設定します。 C1: MDWV CARR, FRQ, 1000</p>
関連コマンド	ARWV、BTWV、SWWV、BSWV

## 2.7. SWWV スイープ波形コマンド

説明	変調波形のパラメータを設定または取得します。
コマンド構文	<channel>:SWWV(SweepWaVe)<parameter> <channel>:={C1, C2} <parameter>:= {下表のパラメータ}

パラメータ	値	説明
STATE	<state>	スイープのオン/オフを切り替えます。注：設定する場合、または他のパラメータを読む場合は、最初に STATE を ON に設定する必要があります



TIME	<time>	スイープ時間。
STOP	<stop frequency>	停止周波数。
START	<start frequency>	開始周波数。
TRSR	<trigger src>	トリガーソース。
TRMD	<trigger mode>	トリガ出力の状態。TRSR が EXT の場合、パラメータは無効です。
SWMD	<sweep mode>	スイープスタイル。
DIR	<direction>	スイープ方向。
EDGE	<edge>	エッジ方向。TRSR が EXT の場合、パラメータは無効です。
MTRIG	<manual trigger>	マニュアルトリガー。TRSR が MAN の時のみ、パラメータは有効です。
CARR, WVTP	<wave type>	キャリア波形。
CARR, FRQ	<frequency>	キャリア周波数。
CARR, AMP	<amplitude>	キャリアの振幅。
CARR, OFST	<offset>	キャリアのオフセット値。
CARR, SYM	<symmetry>	キャリアのシンメトリ値。パルスの場合のみ、設定可能です。
CARR, DLY	<delay>	キャリア信号の遅延値。方形波の場合のみ、設定可能です。
CARR, PHSE	<phase>	位相値。

注：キャリアがパルスまたはノイズの場合、スイープをオンにすることはできません。CARR と STATE を設定する場合、最初のパラメータはどちらか一方でなければいけません。

ここで <state>:= {ON, OFF}  
 <time>:= {デフォルトの単位は "S" です。値は機種に依存します。}  
 <stop frequency> := { 基本波の周波数と同じです。 }  
 <start frequency> := { 基本波の周波数と同じです。 }  
 <trigger src>:= {EXT, INT, MAN}  
 <trigger mode>:= {ON, OFF}  
 <sweep mod>:= {LINE, LOG}  
 <direction>:= {UP, DOWN}  
 <edge>:= {RISE, FALL}  
 <wave type>:= {SINE, SQUARE, RAMP, ARB}  
 <frequency> := { デフォルトの単位は "Hz" です。値は機種に依存します。 }  
 <amplitude> := { デフォルトの単位は "V" です。値は機種に依存します。 }  
 <offset> := { デフォルトの単位は "V" です。値は機種に依存します。 }  
 <duty>:= {0% to 100 %}  
 <symmetry>:= {0% to 100%}  
 <phase>:= {値は機種に依存します。 }

注 パラメータは機種によって異なりますので、各機種の仕様をご参照ください。

クエリ構文 <channel>: SWWV (SWeepWaVe)?  
 <channel>:= {C1, C2}

応答フォーマット <parameter> := { 現在のスイープ波の全パラメータを返します。 }

実施例 チャンネル 1 のスイープタイムを 1S に設定します。

C1: SWWV TIME, 1

チャンネル 1 のスイープ停止周波数を 1000Hz に設定します。

C1: SWWV STOP, 1000

STATE が ON のチャンネル 2 のスイープパラメータを読み出します。

C2: SWWV?

戻り値 :

C2: SWWV STATE, ON, TIME, 1S, STOP, 100HZ, STAR  
100HZ, TRSR, MAN, TRMD, OFF, SWMD, LINE, DIR, U  
CARR, WVTP, SQUARE, FRQ, 1000HZ, AMP, 4V, OFST, 0V,  
DUTY, 50, PHSE, 0

STATE が OFF のチャンネル 2 スイープパラメータを読み出します。

C2: SWWV?

戻り値 :

C2: SWWV STATE, OFF

## 2.8. BTWV バースト波形コマンド

説明 バースト波形のパラメータを設定または取得します。

コマンド構文 <channel>:BTWV(BursTWaVe) <parameter>  
<channel>:={C1, C2}  
<parameter>:={下表のパラメータ}

パラメータ	値	説明
STATE	<state>	バーストをオンまたはオフにします。注: バーストの他のパラメータを設定したり読み出したりする場合は、最初に STATE を ON にする必要があります。
PRD	<period>	バースト期間の設定値。キャリアがノイズの場合は設定できません。トリガソースが EXT の場合、設定できません。
STPS	<start phase>	キャリアの開始位相。キャリアがノイズまたはパルスの場合、設定できません。
GATE_NCYC	<gate Ncycle>	バーストモードを GATE または NCYC に設定します。キャリアがノイズの場合、設定できません。
TRSR	<trigger source>	トリガソースを設定します。
DLAY	<delay>	遅延時間。キャリアがノイズの場合、設定できません。NCYC が選択されている場合に設定できます。
PLRT	<polarity>	極性。GATE が選択されている場合に設定できます。キャリアがノイズの場合、唯一のパラメータです。
TRMD	<trig mode>	トリガモード。キャリアがノイズの場合、設定で

		きません。NCYC が選択されている場合に設定できません。TRSR が EXT に設定されている場合は、設定できません。
EDGE	<edge>	エッジの選択。キャリアがノイズの場合、設定できません。NCYC が選択されている場合に設定できます。TRSR が EXT に設定されている場合は、設定できません。
TIME	<circle time>	サイクル数。キャリアがノイズの場合、設定できません。NCYC が選択されている場合に設定できます。
MTRIG	<manual trig>	マニュアルトリガー。TRSR が MAN の時のみ、パラメータは有効です。
CARR, WVTP	<wave type>	キャリア波形。
CARR, FRQ	<frequency>	キャリア周波数。
CARR, AMP	<amplitude>	キャリアの振幅。
CARR, OFST	<offset>	キャリアのオフセット値。
CARR, SYM	<symmetry>	キャリアのシンメトリ値。パルスの場合のみ、設定可能です。
CARR, DUTY	<duty>	デューティサイクル値。方形波でパルスの場合のみ、設定可能です。
CARR, PHSE	<phase>	位相値。
CARR, RISE	<rise>	立ち上がり時間。パルスの場合のみ、設定可能です。
CARR, FALL	<fall>	立ち下がり時間。パルスの場合のみ、設定可能です。
CARR, MEAN	<mean>	キャリアの平均値。キャリアがノイズの場合のみ有効です。
CARR, STDEV	<stdev>	キャリアの分散値。キャリアがノイズの場合のみ有効です。
CARR, DLY	<delay>	キャリア信号の遅延値。パルスの場合のみ、設定可能です。

注：CARR と STATE を設定する場合、最初のパラメータはどちらか一方でなければいけません。

ここで <state>:= {ON, OFF}  
 <period>:= {デフォルトの単位は "S" です。値は機種に依存します。}  
 <start phase>:= {0 to 360}  
 <gate ncycle>:= {GATE, NCYC}  
 <trigger source>:= {EXT, INT, MAN}  
 <delay>:= {デフォルトの単位は "S" です。値は機種に依存します。}  
 <polarity>:= {NEG, POS}  
 <trig mode >:= {RISE, FALL, OFF}  
 <edge>:= { RISE, FALL}  
 <circle time> := { 値は機種に依存します。 ("INF" は無限です。)}  
 <wave type>:= {SINE, SQUARE, RAMP, PULSE, NOISE, ARB}  
 <frequency> := { デフォルトの単位は "Hz" です。値は機種に依存します。}  
 <amplitude>:= { デフォルトの単位は "V" です。値は機種に依存します。}  
 <offset>:= { デフォルトの単位は "V" です。値は機種に依存します。}  
 <duty>:= {0% to 100%.}  
 <symmetry> := { 0% to 100%}  
 <phase>:= { 値は機種に依存します。}  
 <stdev>:= {デフォルトの単位は "V" です。値は機種に依存します。}

---

<mean>:= {デフォルトの単位は"V"です。値は機種に依存します。}  
 <width> := { Max\_width < (Max\_duty \* 0.01) \* period ,  
 Min\_width > (Min\_duty \* 0.01) \* period }  
 <rise>:= {値は機種に依存します。}  
 <fall>:= {値は機種に依存します。}  
 <delay>:= {デフォルトの単位は "S"です。}

注 パラメータは機種によって異なりますので、各機種の仕様をご参照ください。

クエリ構文	<pre> &lt;channel&gt;: BTWV (BurstWaVe)? &lt;parameter&gt; &lt;channel&gt;:= {C1, C2} &lt;parameter&gt;:= &lt;period&gt;..... </pre>
応答フォーマット	<pre> &lt;channel&gt;:BTWV &lt;type&gt;,&lt;state&gt;,&lt;period&gt;..... </pre>
実施例	<p>チャンネル 1 のバースト周期を 1S に設定します。  C1: BTWV PRD, 1</p> <p>チャンネル 1 のバーストディレイを 1s に設定します。  C1: BTWV DLAY, 1</p> <p>チャンネル 1 のバーストを無限大にします。  C1:BTWV TIME, INF</p> <p>STATE が ON のチャンネル 2 のバーストパラメータを読み出します。  C2: BTWV?  戻り値 :  C2: BTWV STATE,ON,PRD,0.01S,STPS,0,TRSR,INT,  TRMD,OFF,TIME,1,DLAY,2.4e-07S,GATE_NCYC,NCYC,  CARR,WVTP,SINE,FRQ,1000HZ,AMP,4V,OFST,0V,PHSE,0</p> <p>STATE が OFF のチャンネル 2 のバーストパラメータを読み出します。  C2: BTWV?  戻り値 :  C2: BTWV STATE, OFF</p>

## 2.9. PACP パラメータコピーコマンド

説明	一方のチャンネルから他方のチャンネルにパラメータをコピーします。
コマンド構文	<pre> PACP(ParaCoPy) &lt;destination channel&gt;, &lt;src channel&gt; &lt; destination channel&gt;:= {C1, C2} &lt;src channel&gt;:= {C1, C2} </pre> <p>注 : パラメータ C1、C2 は一緒に機器に設定する必要があります。</p>

実施例                   チャンネル 1 からチャンネル 2 へパラメータをコピーします。  
PACP C2, C1

## 2.10. ARWV 任意波形コマンド

説明                    任意の波形を設定または取得します。

コマンド構文           <channel>:ARWV(ArbWaVe)INDEX,<value1>,  
NAME,<value2>  
<channel>:={C1, C2}  
<value1>: (インデックス番号の意味は下表の通りです)  
<value2>: 下表参照。

クエリ構文            <channel>: ARWV (ARbWaVe)?  
<channel>:={C1, C2}

応答フォーマット      <channel>:ARWV <index>

実施例                   StairUp 任意波形出力をインデックスで設定します。  
C1:ARWV INDEX, 2

現在の任意波形を読みます。  
ARWV?  
戻り値 :  
ARWV INDEX,2,NAME,StairUp

Cardiac 任意波出力を名前で設定します。  
ARWV NAME, Cardiac

関連コマンド           BSWV

<表>:

インデックス	名称	インデックス	名称	インデックス	名称	インデックス	名称
0	StairUp	12	Sqrt	24	Cardiac	36	Bartlet
1	StairDn	13	Root3	25	Quake	37	Tan
2	Stairud	14	X^2	26	Chirp	38	Cot
3	Ppulse	15	X^3	27	Twotone	39	Sec
4	Npulse	16	Sinc	28	Snr	40	Csc
5	Trapezia	17	Gaussian	29	Hamming	41	Asin
6	Upramp	18	Dlorentz	30	Hanning	42	Acos
7	Dnramp	19	Haversine	31	Kaiser	43	Atan
8	Exp_fall	20	Lorentz	32	Blackman	44	Acot
9	Exp_rise	21	Gauspuls	33	Gausswin		
10	Logfall	22	Gmonopuls	34	Triang		
11	Logrise	23	Tripuls	35	Harris		

この表について: この表は一例であり、インデックスはモデルによって異なる可能性があります

---

ります。STL?"コマンドを実行することで、正確なインデックスを得ることができます。

## 2.11. SYNC 同期コマンド

説明	同期信号を設定します。
コマンド構文	<code>&lt;channel&gt;: SYNC &lt;parameter&gt;</code> <code>&lt;channel&gt;:= {C1, C2}</code> <code>&lt;parameter&gt;:= {ON, OFF}</code>
クエリ構文	<code>&lt;channel&gt;: SYNC?</code> <code>&lt;channel&gt;:= {C1, C2}</code>
応答フォーマット	<code>&lt;channel&gt;: SYNC &lt;parameter&gt;</code>
実施例	チャンネル 1 の同期機能を ON にします。 C1 : SYNC ON  チャンネル 1 の同期状態を読み出します。 C1 : SYNC? 戻り値 : C1: SYNC OFF

## 2.12. NBFM 数値フォーマットコマンド

説明	数値のフォーマットを設定または取得します。									
コマンド構文	<code>NBFM(NumBer_ForMat) &lt;parameter&gt;</code> <code>&lt;parameter&gt; := { 下表にあるパラメータ。 }</code>									
	<table border="1"><thead><tr><th>パラメータ</th><th>値</th><th>説明</th></tr></thead><tbody><tr><td>PNT</td><td><code>&lt;pnt&gt;</code></td><td>小数点形式</td></tr><tr><td>SEPT</td><td><code>&lt;sept&gt;</code></td><td>区切り形式</td></tr></tbody></table> <code>&lt;pnt&gt;:= {Dot, Comma}.</code> <code>&lt;sept&gt; := { Space, Off, On}.</code>	パラメータ	値	説明	PNT	<code>&lt;pnt&gt;</code>	小数点形式	SEPT	<code>&lt;sept&gt;</code>	区切り形式
パラメータ	値	説明								
PNT	<code>&lt;pnt&gt;</code>	小数点形式								
SEPT	<code>&lt;sept&gt;</code>	区切り形式								
クエリ構文	<code>NBFM(NumBerForMat)?</code>									
応答フォーマット	<code>NBFM &lt;parameter&gt;</code>									
実施例	小数点の形式をドットに設定します。 NBFM PNT, DOT  区切り形式を ON にします。 NBFM SEPT, ON  現在の設定を読みます。 NBFM? 戻り値 :									

---

## 2.13. SCFG コンフィグレーションコマンド

説明	電源投入時のシステム設定を設定または取得します。
コマンド構文	SCFG(Sys_CFG)<parameter> <parameter>:= {DEFAULT, LAST}
クエリ構文	SCFG (Sys_CFG)?
応答フォーマット	SCFG <parameter>
実施例	電源投入時の設定を最後の設定にします。 SCFG LAST

## 2.14. BUZZ ブザーコマンド

説明	ブザーのオン／オフを切り替えます。
コマンド構文	BUZZ(BUZZer) <parameter> <parameter>:= {ON, OFF}
クエリ構文	BUZZ (BUZZer)?
応答フォーマット	BUZZ <parameter>
実施例	ブザーを ON にします。 BUZZ ON

## 2.15. SCSV スクリーンセーバーコマンド

説明	スクリーンセーバーを設定します。
コマンド構文	SCSV (SCreen_SaVe) <parameter> <parameter>:= {OFF, 1, 5, 15, 30, 60, 120, 300} (単位は分です。)
クエリ構文	SCSV (SCreen_SaVe)?
応答フォーマット	SCreen_SaVe <parameter>
実施例	スクリーンセーバーを 5 分に設定します。 SCSV 5  スクリーンセーバーの時間を読みます。

---

---

Sscreen\_SaVe?

戻り値 :

SCSV 5MIN

## 2.16. ROSC クロックソースコマンド

説明	クロックソースを設定または取得します。
コマンド構文	ROSC (ROSCillator) <parameter> <parameter>:= {INT, EXT}
クエリ構文	ROSC (ROSCillator)?
応答フォーマット	ROSC <parameter>
実施例	内部タイムベースをクロックソースとして設定します。 ROSC INT

## 2.17. FCNT 周波数カウンタコマンド

説明	周波数カウンタのパラメータを設定または取得します。
コマンド構文	FCNT(FreqCouNter) <parameter> <parameter>:= {下表のパラメータ}

パラメータ	値	説明
STATE	<state>	周波数カウンタの状態。
FRQ	<frequency>	周波数の値。設定不可。
PW	<position width>	正のパルス幅の値。設定不可。
NW	<negative width>	負のパルス幅の値。設定不可。
DUTY	<duty>	デューティサイクルの値。
FRQDEV	<freq deviation>	周波数偏差の値。設定不可。
REFQ	<ref freq>	リファレンス周波数の値。
TRG	<triglev>	トリガレベルの値。
MODE	<mode>	モードの値。
HFR	<HFR>	HFR の状態。

ここで <state>:= {ON, OFF}

<frequency>:= {デフォルトの単位は "Hz "です。値の範囲は機種に依存します。}

<mode>:= {AC, DC}

<HFR>:= {ON, OFF}



---

クエリ構文	FCNT (FreqCouNter)?
応答フォーマット	FCNT < state ><frequency><duty> <ref freq><triglev><position width><negative width> <freq deviation><mode><HFR>
実施例	周波数カウンターをオンにします。 FCNT STATE,ON  周波数カウンターの情報を問い合わせます。 FCNT ? 戻り値 : FCNT STATE,ON,FRQ,0HZ,DUTY,0,REFQ,1e+07HZ,TRG,0V,PW,0 S,NW,0S,FRQDEV,-1e+06ppm,MODE,AC,HFR,OFF

## 2.18. INVT インバートコマンド

説明	現在のチャンネルの極性を設定または取得します。
コマンド構文	<channel>:INVT(INVerT) <parameter> <channel>:={C1, C2} <parameter>:={ON, OFF}
クエリ構文	<channel>: INVT (INVerT)? <channel>:={C1, C2}
応答フォーマット	<channel>:INVerT <parameter>
実施例	チャンネル 1 の反転を ON にします。 C1: INVT ON  チャンネル 1 の極性を読み取ります。 C1: INVT? 戻り値 : C1: INVT ON

## 2.19. COUP カップリングコマンド

説明	チャンネルカップリングのパラメータを設定または取得します。TRACE をオフにすると、他のパラメータを設定できます。
コマンド構文	COUPling <parameter>,<value> <parameter>:={下表のパラメータ} <value>:={下表の値}

パラメータ	値	説明
TRACE	<track_enable>	同期を有効/無効にします。

FCOUP	<fcoup>	周波数結合を有効／無効にします。
FDEV	<frq_dev>	チャンネル間の周波数偏差。
FRAT	<frat>	チャンネル間の周波数比。
PCOUP	<pcoup>	位相結合を有効／無効にします。
PDEV	<pha_dev>	チャンネル間の位相偏差。
PRAT	<prat>	チャンネル間の位相比。
ACOUP	<acoup>	振幅結合を有効／無効にします。
ADEV	<adev>	チャンネル間の振幅偏差。
ARAT	<arat>	チャンネル間の振幅比。

ここで <track\_enable>:={ON, OFF}  
 <fcoup>:={ON, OFF}  
 <frq\_dev>:={単位は"Hz"です。}  
 <pcoup>:={ON, OFF}  
 <pha\_dev>:={単位は"°"です。}  
 <acoup>:={ON, OFF}  
 <adev>:={単位は"V"です。}

クエリ構文

COUPLing?

応答フォーマット

COUP <parameter>  
 <parameter>:={ All parameters of coupling}

実施例

周波数結合を有効にします。  
 COUP FCOUP,ON

周波数偏差を 5Hz に設定します。  
 COUP FDEV,5

カップリング設定を取得します。  
 COUP?  
 戻り値：  
 COUP TRACE,OFF,FCOUP,ON,PCOUP,ON,ACOUP,ON,  
 FDEV,5HZ,PRAT,1,ARAT,2

## 2.20. VOLTPRT 過電圧保護コマンド

説明

過電圧保護の状態を設定または取得します。

コマンド構文

VOLTPRT<parameter>  
 <parameter>:={ON,OFF}

クエリ構文

VOLTPRT?

応答フォーマット

VOLTPRT<parameter>



説明 前面パネルのキーを押したときの動作をシミュレートするコマンドを送信します。

コマンド構文 VKEY (VirtualKEY) VALUE,<value>,STATE,<state>  
 <value>:= {下表のパラメータ}  
 <state>:=<0,1>(“1” は仮想値として有効、“0” は無効)

実施例 VKEY VALUE,15, STATE,1  
 VKEY VALUE,KB\_SWEEP, STATE,1

KB_FUNC1	28	KB_NUMBER_4	52
KB_FUNC2	23	KB_NUMBER_5	53
KB_FUNC3	18	KB_NUMBER_6	54
KB_FUNC4	13	KB_NUMBER_7	55
KB_FUNC5	8	KB_NUMBER_8	56
KB_FUNC6	3	KB_NUMBER_9	57
KB_MOD	15	KB_POINT	46
KB_SWEEP	16	KB_NEGATIVE	43
KB_BURST	17	KB_LEFT	44
KB_WAVES	4	KB_RIGHT	40
KB_UTILITY	11	KB_OUTPUT1	153
KB_PARAMETER	5	KB_OUTPUT2	152
KB_STORE_RECALL	70	KB_KNOB_RIGHT	175
KB_NUMBER_0	48	KB_KNOB_LEFT	177
KB_NUMBER_1	49	KB_KNOB_DOWN	176
KB_NUMBER_2	50	KB_HELP	12
KB_NUMBER_3	51	KB_CHANNEL	72

## 2.23. SYST:COMM:LAN:IPAD IP コマンド

説明 システムの IP アドレスを設定・取得することができます。

コマンド構文 SYST:COMM:LAN:IPAD  
 (SYSTem:COMMunicate:LAN:IPADdress)  
 <parameter1>.<parameter2>.<parameter3>.<parameter4>  
 <parameter1>:= {1～223 の整数値}  
 <parameter2>:= {1～255 の整数値}  
 <parameter3>:= {1～255 の整数値}  
 <parameter4>:= {1～255 の整数値}

クエリ構文 SYST:COMM:LAN:IPAD  
 (SYSTem:COMMunicate:LAN:IPADdress)?

実施例 IP アドレスを 10.11.13.203 に設定します。  
 SYSTem: COMMunicate: LAN:IPADdress 10.11.13.203

IP アドレスを取得します。  
 SYST:COMM:LAN:IPAD?  
 戻り値 :

---

“10.11.13.203”

## 2.24. SYST:COMM:LAN:SMAS サブネットマスクコマンド

説明	システムのサブネットマスクを設定・取得することができます。
コマンド構文	<pre>SYST:COMM:LAN:SMAS (SYSTem:COMMunicate:LAN:SMASk) &lt;parameter1&gt;.&lt;parameter2&gt;.&lt;parameter3&gt;.&lt;parameter4&gt;  &lt;parameter1&gt;:={0~255 の整数値} &lt;parameter2&gt;:={0~255 の整数値} &lt;parameter3&gt;:={0~255 の整数値} &lt;parameter4&gt;:={0~255 の整数値}</pre>
クエリ構文	<pre>SYSTem:COMMunicate:LAN:SMASk?</pre>
実施例	<p>サブネットマスクを 255.0.0.0 に設定します。</p> <pre>SYSTem:COMMunicate:LAN:SMASk 255.0.0.0</pre> <p>サブネットマスクを取得します。</p> <pre>SYSTem:COMMunicate:LAN:SMASk?</pre> <p>戻り値 :</p> <pre>“255.0.0.0”</pre>

## 2.25. SYST:COMM:LAN:GAT ゲートウェイコマンド

説明	システムゲートウェイの設定と取得が可能です。
コマンド構文	<pre>SYST:COMM:LAN:GAT(SYSTem:COMMunicate:LAN:GATeway) &lt;parameter1&gt;.&lt;parameter2&gt;.&lt;parameter3&gt;.&lt;parameter4&gt;  &lt;parameter1&gt;:={0~223 の整数値} &lt;parameter2&gt;:={0~255 の整数値} &lt;parameter3&gt;:={0~255 の整数値} &lt;parameter4&gt;:={0~255 の整数値}</pre>
クエリ構文	<pre>SYSTem:COMMunicate:LAN:GATeway?</pre>
実施例	<p>ゲートウェイを 10.11.13.5 に設定します。</p> <pre>SYSTem:COMMunicate:LAN:GATeway 10.11.13.5</pre> <p>ゲートウェイを取得します。</p> <pre>SYSTem:COMMunicate:LAN:GATeway?</pre> <p>戻り値 :</p> <pre>“10.11.13.5”</pre>

---

## 2.26. SRATE サンプルレートコマンド

説明	任意波形モードとサンプルレートを設定または取得します。  注：サンプルレートの設定・取得は、TrueArb モード時のみ可能です。
コマンド構文	<code>&lt;channel&gt;:SRATE(SampleRATE) MODE &lt;parameter1&gt;, VALUE, &lt;parameter2&gt;</code>  <code>&lt;channel&gt; := &lt;C1, C2&gt;</code> <code>&lt;parameter1&gt; := &lt; DDS, TARB &gt;</code> <code>&lt;parameter2&gt; := {1e-6~75000000 の整数値, (デフォルトの単位は Sa/s です)}</code>
クエリ構文	<code>&lt;channel&gt;: SRATE?</code>
実施例	チャンネル 1 のサンプルレート値を取得します。 <code>C1: SRATE?</code> 戻り値： <code>C1: SRATE MODE, DDS</code>  チャンネル 1 を TrueArb モードに設定します。 <code>C1: SRATE MODE, TARB</code>  チャンネル 1 のサンプルレート値を 1000000Sa/s に設定します。 <code>C1: SRATE VALUE, 1000000</code>

## 2.27. HARM ハーモニックコマンド

説明	高調波パラメータを設定または取得します。基本波が SINE のときのみ使用可能です。
コマンド構文	<code>&lt;channel&gt;:HARMonic HARMSTATE,&lt;state&gt;,HARMTYPE,&lt;type&gt;,HARMORDER,&lt;order&gt;,&lt;unit&gt;,&lt;value&gt;,&lt;unit&gt;,&lt;value&gt;,HARMPHASE,&lt;phase&gt;</code>  <code>&lt;state&gt;:= &lt;ON, OFF&gt;.</code> <code>&lt;type&gt;:= &lt;EVEN, ODD, ALL&gt;.</code> <code>&lt;order&gt;:= {1,2,...,10}</code> <code>&lt;unit&gt;:= &lt; HARMAMP, HARMDBC &gt;.</code> <code>&lt;value&gt;:= 高調波の振幅。値は機種に依存します。&lt;unit&gt;= HARMAMP のとき、単位は"V","Vpp",&lt;unit&gt;= HARMDBC のとき、単位は"dBc"です。</code> <code>&lt;phase&gt;:= {0~360},単位は"°"です。</code>
クエリ構文	<code>&lt;channel&gt;:HARMonic?</code> <code>&lt;channel&gt; := {C1, C2}</code>

---

---

実施例	<p>チャンネル1のハーモニック機能を有効にします。 C1:HARM HARMSTATE,ON</p> <p>チャンネル1の2次高調波を-6dBcに設定します。 C1:HARM HARMORDER,2,HARMDBC,-6</p> <p>チャンネル1の高調波情報を取得します。 C1:HARM? 戻り値: C1:HARM ,HARMSTATE,ON,HARMTYPE,EVEN, HARMORDER,2,HARMAMP,2.004748935V,HARMDBC,-6dBc, HARMPHASE,0</p>
-----	---

## 2.28. MODE モードコマンド

説明	フェーズモードを設定または取得します。
コマンド構文	MODE <parameter> <parameter>:= {PHASE-LOCKED, INDEPENDENT}
クエリ構文	MODE?
実施例	位相モードを INDEPENDENT に設定します。 MODE INDEPENDENT

## 2.29. インデックス

[\\*IDN](#)  
[\\*OPC](#)  
[\\*CLS](#)  
[\\*ESE](#)  
[\\*ESR](#)  
[\\*RST](#)  
[\\*SRE](#)  
[\\*STB](#)  
[\\*TST](#)  
[\\*WAI](#)

### A

[ARWV](#) ARBWAVE

### B

[BSWV](#) BASIC\_WAVE

[BTWV](#) BURSTWAVE

[BUZZ](#) BUZZER

---

---

## C

[COUP](#) COUPLING

## F

[FCNT](#) FREQCOUNTER

## H

[HARM](#) HARMONIC

## I

[INVT](#) INVERT

## M

[MDWV](#) MODULATEWAVE

[MODE](#)

## N

[NBFM](#) NUMBER\_FORMAT

## O

[OUTP](#) OUTPUT

## P

[PACP](#) PARACOPY

## R

[ROSC](#) ROSCILLATOR

## S

[SCFG](#) Sys\_CFG

[SCSV](#) SCREEN\_SAVE

[SWWV](#) SWEEPWAVE

[SYNC](#) SYNC

[STL](#) STORELIST

[SYST:COMM:LAN:IPAD](#) SYSTEM:COMMUNICATE:LAN:IPADDRESS

[SYST:COMM:LAN:SMAS](#) SYSTEM:COMMUNICATE:LAN:SMASK

[SYST:COMM:LAN:GAT](#) SYSTEM:COMMUNICATE:LAN:GATEWAY

[SRATE](#) SAMPLERATE

## V

[VKEY](#) VIRTUALKEY

[VOLTPRT](#) VOLTPRT



---

**岩崎通信機株式会社**