DC パワーサプライ

PSM シリーズ

プログラムマニュアル

ISO-9001 CERTIFIED MANUFACTURER

GW INSTEK

2018 年 9 月

このマニュアルに記載された情報は印刷時点のものです。製品の仕様、機器、および保守手順は、いつでも予告なしで変更することがありますので、予めご了承ください。

Good Will Instrument Co., Ltd.
No. 7-1, Jhongsing Rd., Tucheng Dist., New Taipei City 236, Taiwan.

目次

1 はじめに

先進的な自動計測システムでは、計測器とコンピュータとの間の通信が不可欠です。測定手法をユーザーの試験プログラムに応じて変更することができます。このため、プログラマブル電源を計測コントローラあるいはコンピュータから RS232 インタフェース(オプション)または GPIB(オプション)経由で制御することができます。.

インタフェースの選択と設定

GPIB アドレスは通常動作状態で変更することができます。前面パネルの[SHIFT]キーと[LOCAL]キーを押すと、最後に使用されていた通信インタフェース設定が表示されます。インタフェースを選択して[ENTER]を押した後、ボーレート(または GPIB アドレス)選択して[ENTER] を押してジョグダイアルによる設定を確認します。最後に「save」を選択して[ENTER]を押し、設定を保存します。

2 GPIB インタフェース

GPIB インタフェースの機能

プログラマブル電源の GPIB インタフェースは IEEE488.1-1987, IEEE488.2-1992 および SCPI-1994 の規格に対応します。GPIB インタフェースの機能を以下に示します。

SH1(ソースハンドシエーク)	プログラマブル電源は GPIB にマルチレーンメッセージを送信できます。
AH1(アクセプタハンドシエーク)	プログラマブル電源は GPIB からマルチレーンメッセージを受信できます。
T6(トーカー)	トーカーインタフェース機能には、基本トーカー、シリアルポール、およびアンアドレスイフ MLA 機能が含まれます。トークオンリーモードの機能は含まれていません。
L4 (リスナー)	コントローラが ATN (アテンション) ラインをアサートしてリスンアドレスを送信したときには、プログラマブル電源はリスナーになります。プログラマブル電源はリスンオンリー機能を持っていません。
SR1 (サービスリクエスト)	プログラマブル電源は、サービスを必要とするときには SRQ (サービスリクエスト) ラインをアサートしてコントローラに知らせます。
RL1 (リモート/ローカル)	プログラマブル電源は GTL(ゴートゥローカル) と LLO(ローカルロックアウト) のインタフェースメッセージの両方に応答します。
PP0 (パラレルポール)	プログラマブル電源はパラレルポール機能を持っていません。

DC1 (デバイスクリア)	プログラマブル電源はデバイスを電源投入直後の状態に戻すデバイスクリア機能を持っています。
DT0 (デバイストリガ)	プログラマブル電源はデバイストリガのインタフェース機能を持っていません。
C0 (コントローラ)	プログラマブル電源は他のデバイスを制御することはできません。

GPIB 接続に関する注記

プログラマブル電源を GPIB システムに接続するときには、以下のことに注意してください。

- 1つの GPIB バスに接続できるデバイスの数は最大 15 です。.
- デバイスとの接続ケーブルは 20m 以内にしてください。
- 使用するケーブル 2m ごとに 1 つのデバイスを接続してください。.
- バス上の各デバイスは独自のデバイスを必要とします。1つのデバイスを 2 つのデバイスが共用することはできません。
- GPIB システムを使用するときにはシステムのデバイスの少なくとも 2/3 の電源をオンにしてください。
- GPIB システムの接続形態をループあるいはパラレルにしないください。.

パソコンの接続

プログラマブル電源を GPIB インタフェース経由で制御するためには GPIB カード付きのパソコンが必順です。

プログラマブル電源とパソコンの間を次のように接続します。

1. GPIB ケーブルの片方の端をパソコンに接続します。
2. GPIB ケーブルの他方の端をプログラマブル電源の GPIB ポートに接続します。.
3. プログラマブル電源の電源をオンにします。
4. パソコンの電源をオンにします。

GPIB 接続の試験

GPIB 接続が動作しているかどうかを調べたいときには、パソコンから GPIB コマンドを送って調べることができます。例えば、クエリコマンド

*idn?

を送ると製造業者、型番、シリアル番号、およびファームウェアバージョンを以下のフォーマットで返すはずです。

GW Inc, PSM-2010, A000000, FW1.00W

プログラマブル電源から適切な応答が得られない場合には、電源がオンであるか、GPIB アドレスが適切か、ケーブル接続先がすべてアクティブかを調べてください。

3 RS-232 インタフェース

RS-232 インタフェース機能

RS-232 インタフェースはパンコンとプログラマブル電源などの 2 台の機器間をポイントトゥポイントで接続するインタフェースです。両方の側でいくつかのパラメータを設定する必要があります。パラメータを設定した後、RS232 インタフェースを通してプログラマブル電源を制御することができます。

- ボーレート: 1200, 2400, 4800、または 9600 ボーのボーレートを選択することができます。
- パリティビット: なし.
- データビット: 8 ビット.
- ストップビット: 1 ビット.
- データフロー制御: なし

RS232 接続に関する注記

プログラマブル電源は後面パネルに 9 ピン D 型 RS232 コネクタを備えた *DTE 機器です。9ピンコネクタ(オス型)のピン番号の割当てを図 2 に示します。プログラマブル電源を* RS232 インタフェースで接続したときには以下の点を確認してください。

- DTE デバイスの出力ラインを他 DTE デバイスの出力ラインに接続してはいけません。
- 多くの機器では 1 つあるいは複数の入力ピンを定常的に HI にしておく必要があります。
- 機器のシグナルグラウンドが外部機器のシグナルグラウンドと接続されていることを確認してください。
- 機器のシャーシグラウンドが外部機器のシャーシグラウンドと接続されていることを確認してください。
- 機器とパソコンとの接続ケーブルは 15m 以内にしてください。.

- 機器で使用するボーレートとパソコン端末で使用するボーレートを合わせてください。.
- ケーブル両端のコネクタと内部接続ラインが機器の要求に合致することを確認してください。.

1. 接続なし
2. 受信データ (RxD)　　(入力)
3. 送信データ(TxD)　　(出力)
4. 接続なし
5. シグナルグラウンド (GND)
6. 接続なし
7. 接続なし
8. 接続なし
9. 接続なし

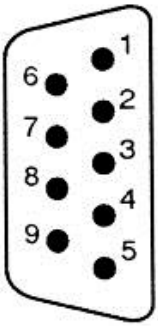

図 1 後面パネル DB-9-D RS232 コネクタのピン割当て

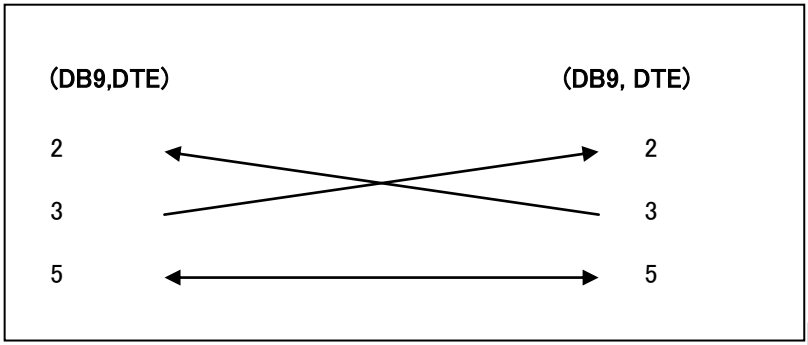

図 2 DB9 から DB9 への配線接続

パソコンの接続

プログラマブル電源を RS232 インタフェース経由で制御するためには COM ポート付きのパソコンが必須です。

プログラマブル電源とパソコンとの間の接続は以下のようにします。

1. RS232 ケーブルの片方の端をパソコンに接続します。
2. ケーブルの他方の端をプログラマブル電源の RS232 ポートに接続します。
3. プログラマブル電源の電源をオンにします。
4. パソコンの電源をオンにします。

RS232 接続の試験

RS232 の接続が正常に動作しているかどうかを調べたいときには、

パソコンからコマンドを送って調べることができます。例えば、ターミナルプログラムを使ってクエリコマンド

*idn?

を送ると製造業者、型番、シリアル番号、およびファームウェアバージョンを以下のフォーマットで返すはずです。

GW. INC,PST-3202,A000000,FW1.00

プログラマブル電源から適切な応答が得られない場合には、電源がオンであるか、RS232 のボーレートが適切か、ケーブル接続先がすべてアクティブかを調べてください。

4 入力キューと出力キュー

128 バイトの入力キューと128バイトの出力キューが組み込まれており、保留中のコマンドとりターンメッセを保存して、リモートコントロールのための送信コマンドとりターンメッセージが失われないようにします。エラー/イベントキューの最大容量が20メッセージ分であるため、これらのバッファを使うことにより容量を超える入力データがあればデータが失われることに注意願います。

5 コマンドと構文

プログラマブル電源の GPIB コマンドは IEEE-488.2 と SCPI 規格に適合しています。

SCPI

SCPI (プログラマブル計測器標準コマンド)は試験計測器の主要メーカーの国際団体が作成した規格です。IEEE-488.2 の構文が SCPI で採用されており各種のプログラマブル計測器で同じ機能について同一のコマンドを利用できるようにします。

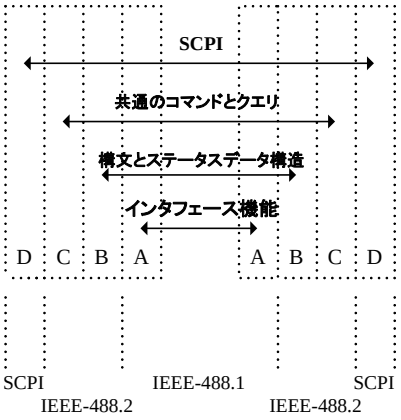

図 3 IEEE-488.1, IEEE-488.2, および SCPI の関係

図 3 に示すように、IEEE-488.1 規格はレイヤ A に位置します。レイヤ A は GPIB バスのインタフェース機能のプロトコルです。ソースハンドシエーク(SH)、アクセプタハンドシエーク(AH)、およびトーカーがこのレイヤに含まれています(合計 10 のインタフェース機能)。レイヤ B では、構文とデータ構造が IEEE-488.2 規格全体の核心となることがあります。構文ではメッセージ通信の機能が定義されており、<プログラムメージ>(すなわち、「コマンド」)と<応答メッセージ>が含まれます。2 種類のメッセージ>(すなわち、「コマンド」)と<応答メッセージ>が含まれます。2 種類のメッセージはデバイスのコマンドと戻り値の構文構造を表します。データ構造は IEEE-488.2 規格によって定義される標準レポーテイングの構成です。

共通のコマンドとクエリがレイヤ C に含まれています。コマンドとクエリは、必須と任意の 2 つの種類に区分することができます。コマンドは制御設定の変更あるいは特定の機能の実行を計測器の命じます。クエリによって計測器がデータまたはステータス情報をパソコンに送り返します。コマンド末尾に疑問符(?)を付けると、クエリであることを示します。

レイヤ D は機器の情報に関連します。別々の機器は別々の機能を持ちます。SCPI コマンドセットはこのレイヤに属します。

コマンドの構文

SCPI に従って計測器に命令を伝えたいときには、以下の 3 つの基本要素を含める必要があります。

- コマンドヘッダ
- パラメータ(必要な場合)
- メッセージターミネータまたはメッセージセパレータ

コマンドヘッダ

コマンドヘッダは階層構造を持ち、コマンドツリー（図 4）によって表されます。

ツリーの再上位はルートレベルです。ルートノードがルートレベルに存在します。ルートノードと 1 つまたは複数の下位ノードがリーフノードと呼ばれる最終ノードへのヘッダパスを構成します。

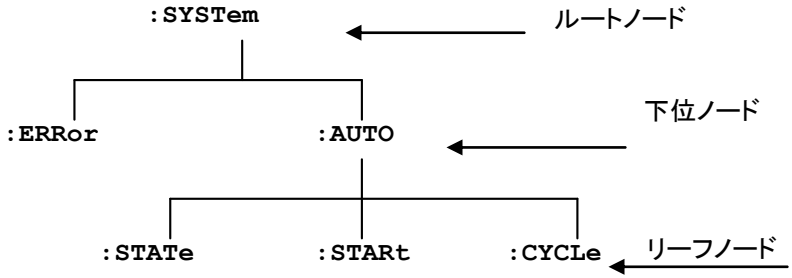

図 4: ツリー階層

コマンドヘッダはヘッダパスとリーフノードから構成されます。図 4 に示された
リーフノードのコマンドヘッダを図 5 に示します。

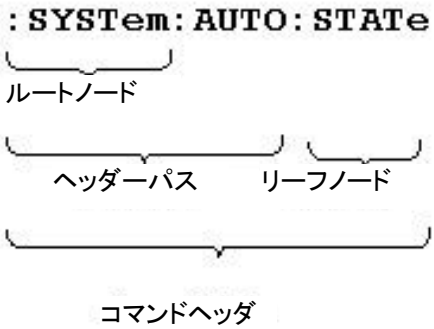

図 5 コマンドヘッダー

パラメータ

コマンドがパラメータを持つときには、その値を含める必要があります。コマニュアルではコマンドの構文を説明するときにパラメータの種類を示すために< >シンボルを使います。例えば、図 6 のコマンドの構文はブーリアン型のパラメータを含みます。

注記: パラメータの実際の値を入力するときには< >、あるいはシンボルを含めないでください。

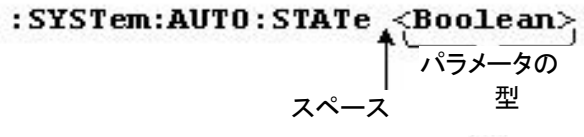

図 6 パラメータ付きのコマンドヘッダ

プログラマブル電源のブーリアン型および他の型のパランメータの定義を表 1 に示します。

パラメータの型	定義	例
ブーリアン	ブール代数値	0, 1
NR1	整数	0, 1, 18
NR2	小数	1.5, 3.141, 8.4
NR3	浮動小数点数	4.5E-1, 8.25E+1
ストリング	英数字	“No error”

表 1:構文記述のためのパラメータの型

メッセージターミネータおよびメッセージセパレータ

GPIB メッセージターミネータ

IEEE 488.2 規格に従って、以下に示す任意のメッセージターミネータが利用可能です。

LF^END	改行コード(16 進 0A)および END メッセージ
LF	改行コード
<dab>^END	最終データバイトおよび END メッセージ

これらのターミネータはほとんどのアプリケーションプログラムと適合します。1 つの行に複数のコマンドを記述するときにはセミコロンによってコマンドどうしを分離します。

RS-232 メッセージターミネータ

RS-232 バスではメッセージの終わりを示す信号がないため、LF（改行）をメッセージターミネータとして利用します。一連のコマンドを計測器に送った後、メッセージターミネータの印として LF を付加する必要があります。クエリコマンドについては、計測器のリターンメッセージにも LF を付加してパソコンがメッセージターミネータを判別できるようにします。

コマンドの入力

プログラマブル電源のコマンドセットを規定する規格では、コマンド入力時にある程度の柔軟性を許容しています。例えば、多くのコマンドを短縮することができ、また複数のコマンドを組み合わせて 1 つのメッセージにしてプログラマブル電源に送ることができ。「フレンドリリスニング」と呼ばれる柔軟性によってプログラミング時間を節約してコマンドセットを覚えやすくし、また利用しやすくします。

コマンドの文字

プログラマブル電源はコマンドの大文字と小文字を区別しません。コマンドを大文字で入力しても小文字で入力してもかまいません。.

スペース文字を含む任意のコマンドを実行することができます。しかし、パラメータとコマンドヘッダと間には少なくとも 1 個のスペースを入れなければなりません。

コマンドの短縮

ほとんどのコマンドには長形式と短形式があります。この節に示す各コマンドでは省略可能部分を小文字で示します。例えば、:CHANnel1:VOLTage 1.23 というクエリを： :CHAN1:VOLT 1.23 と省略することができます。

プログラマブル電源はコマンドがルートから始まっていると仮定するため、最初のコマンドヘッグをコロン(:).から始めることができます。

コマンドの結合

セミコロン(;)を使ってコマンドを結合することができます。しかし、連続したクエリコマンドを使うとメッセージが失われるおそれがあります。例：
CHAN1:VOLT ?;CURR ?

セミコロンの後のコマンドがルートレベルとは別のヘッグパスを持つときには、コロンを使ってルートレベルに戻らなければなりません。

:CHAN1:VOLT 1.23;:OUTP:COUP:TRAC 1

セミコロンの後のコマンドが同一のヘッダパスを持つときには、コロンとパスを省略して新しいリーフノードのみを示すことができます。例えば、

:CHAN1:VOLT 12.34;CHAN1:CURR 1.55

は次のコマンドと等価です。

:CHAN1:VOLT 12.34;CURR 1.55

複数のコマンドやクエリを結合して 1 つのメッセージにすることができます。例えば、以下のようにすることができます。

:CHAN1:VOLT 12.34;VOLT ?

コマンドの概要

この節の表ではプログラマブル電源のコマンドの概要を示します。表ではコマンドを以下の 3 種類に分類しています。

- 一般設定コマンド
- ステータスコマンド
- 各種コマンド

表では各コマンドの簡単な説明も含みます。

一般設定コマンド

プログラマブル電源の設定のコマンドとクエリを行う一般設定コマンドを示します。

APPLy {<voltage>|DEF|MIN|MAX}[,{<current>|DEF|MIN|MAX}]

APPLy?

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] {<current>|MIN|MAX|UP|DOWN}

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]? [MIN|MAX]

[SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement] {<numeric value>|DEFault}

[SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement]? [DEFault]

[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude] {<current>|MIN|MAX}

[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]? [MIN|MAX]

[SOURce:]CURRent:PROTection[:LEVel] {<current>|MIN|MAX}

[SOURce:]CURRent:PROTection[:LEVel]? [MIN|MAX]

[SOURce:]CURRent:PROTection:STATe {0|1|OFF|ON}

[SOURce:]CURRent:PROTection:STATe?

[SOURce:]CURRent:PROTection:TRIPped?

[SOURce:]CURRent:PROTection:CLEar

[SOURce:]CURRent:PROTection:DELay {<numeric value> |MIN|MAX}**

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] {<voltage>|MIN|MAX|UP|DOWN}

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]? [MIN|MAX]

[SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement] {<numeric value>|DEFault}

[SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]? [DEFault]

[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] {<current>|MIN|MAX}

[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]? [MIN|MAX]

[SOURce:]VOLTage:PROTection[:LEVel] {<current>|MIN|MAX}

[SOURce:]VOLTage:PROTection[:LEVel]? [MIN|MAX]

[SOURce:]VOLTage:PROTection:STATe {0|1|OFF|ON}

[SOURce:]VOLTage:PROTection:STATe?

[SOURce:]VOLTage:PROTection:TRIPped?

[SOURce:]VOLTage:PROTection:CLEar

[SOURce:]VOLTage:RANGe {P8V|P20V|P15V|P30V|P60V|LOW|HIGH}*

[SOURce:]VOLTage:RANGe?

MEASure[:SCALar][:VOLTage][:DC]?

MEASure[:SCALar]:CURRent[:DC]?

OUTPut[:STATe]{0|1|OFF|ON}

OUTPut[:STATe]?

* P8V, P20V: PSM-2010, P15V, P30V: PSM-3004, P30V, P60V: PSM-6003.

** Delay 設定幅: 0.0~10.0s

トリガーコマンド

トリガーの各種設定を行うコマンドを示します。

TRIGger[:SEQuence]:DELay {<seconds>|MIN|MAX}

TRIGger[:SEQuence]:DELay?

TRIGger[:SEQuence]:SOURce {BUS|IMMediate}

TRIGger[:SEQuence]:SOURce?

INITiate[:IMMediate]

*TRG

ステータスコマンド

プログラマブル電源のステータスとイベント構造を構成する各種のレジスタとキューの設定と間合せを行うステータスコマンドを示します。

STATus:QUEStionable:CONDition?

STATus:QUEStionable:ENABle {0|1}

STATus:QUEStionable:ENABle?

STATus:QUEStionable[:EVENt]?

STATus:OPERation:CONDition?

STATus:OPERation:ENABle {0|1}

STATus:OPERation:ENABle?

STATus:OPERation[:EVENt]?

STATus:PRESet

システムコマンド

プログラマブル電源の管理機能を制御する各種コマンドを示します。

*IDN?

*RST

*TST?

*SAV {0|1|2|…..|99}

*RCL {0|1|2|…..|99}

DISPlay:CONTrast {0|1|2|3|4}*

DISPlay:CONTrast?

DISPlay[:WINDow][:STATe] {0|1|OFF|ON }

DISPlay[:WINDow][:STATe]?

DISPlay[:WINDow]:TEXT[:DATA] <quoted string>

DISPlay[:WINDow]:TEXT[:DATA]?

DISPlay[:WINDow]:TEXT:CLEar

SYSTem:BEEPer:STATe {0|1|OFF|ON }

SYSTem:BEEPer::STATe?

SYSTem:BEEPer:[:IMMediate]

SYSTem:ERRor[:NEXT]?

SYSTem:VERSion?

SYSTem:MEMory?

*設定範囲: 0(明るい) ~ 4(暗い)

RS-232C コマンド

SYSTem:LOCal

SYSTem:REMote

SYSTem:RWLock

IEEE488.2 共通コマンド

*CLS

*ESE {enable value}

*ESE?

*ESR?

*IDN?

*OPC

*OPC?

*RCL {0|1|2|…..|99}

*RST

*SAV {0|1|2|…..|99}

*SRE {enable value}

*SRE?

*STB?

*TRG

*TST?

*WAI

その他コマンド

SYSTem:AUTO[:STATe] {OFF|ON}
SYSTem:AUTO[:STATe]?
SYSTem:AUTO:STARt {0|1|2|…..|99}
SYSTem:AUTO:STARt?
SYSTem:AUTO:CEASe {0|1|2|…..|99}
SYSTem:AUTO:CEASe?
SYSTem:AUTO:CYCLe {0*|1|2|…..|99999}
SYSTem:AUTO:CYCLe?
SYSTem:AUTO:DELay {0|1|2|…..|59999}**
SYSTem:AUTO:DELay?
＊0 : Auto の場合無限に繰返し (∞).
**0.1s 単位

6 詳細コマンドリファレンス

この章では各コマンドの詳細な説明を行います。各コマンドの例を示すとともにクエリと応答の例を示します。

IEEE-488.2 共通コマンド

*CLS (クエリ形式なし)

機能:

イベントステータスデータレジスタをすべてクリアします。出力キュー、オペレーションイベントステータスレジスタ、クエスチョナブルイベントステータスレジスタ、および標準イベントステータスレジスタを含みます。

構文:

*CLS

例:

*CLS は、イベントレジスタをすべてクリアします。

*ESE

機能:

イベントステータスイネーブルレジスタ(ESER)のビットの設定または問合わせを行います。ESER は標準イベントステータスレジスタ(SESR)をイネーブルしてステータスバイトレジスタ(SBR)のビット 5 (ESB)に要約します。

構文:

*ESE <NR1>

*ESE?

<NR1>は、0～255.

戻り値:

<NR1> は 0～255の数値であり、ESER の 2 進ビットの 10 進値を示します。

例:

*ESE 65 により、ESER を 2 進の 01000001 に設定します。

ESER が2進の10000010を含む場合には、*ESE?によって 130 という値が返されます。

*ESR? (クエリのみ)

機能:

標準イベントステータスレジスタ (SESR)の内容を返してクリアします。.

構文:

*ESR?

戻り値:

<NR1>は0～255の数値であり ESER のバイナリビットの10進値を示します。

例:

ESER が 2 進の 11000110 を含む場合には、*ESR?によって 198 という値が返されます。

*IDN? (クエリのみ)

機能:

プログラマブル電源の固有の識別コードを返します。

構文:

*IDN?

戻り値:

<string>には製造業者、型番、シリアル番号、およびファームウェアバージョンが含まれます。

例:

*IDN?により、GW,PSM-2010,A1234567,FW1.00 を返します。

*OPC

機能:

コマンド形式 (*OPC)では、保留中のオペレーションがすべて完了したとき標準イベントステータスレジスタ(SESR)のオペレーション完了ビット(ビット 0)をセットします。

クエリ形式(*OPC?)では、プログラマブル電源が保留中のオペレーションを完了したときに出力キューに ASCII コードの 1 を出力するようにプログラマブル電源に命じます。

構文:

*OPC

*OPC?

戻り値:

1

*RCL

機能:

以前に保存した設定データをメモリーから呼び出します (PSS シリーズと PSH シリーズはこの機能を持っていません)。

構文:

*RCL <NR1>

<NR1>は、0～99の範囲内です。

例:

*RCL 12 は、メモリーアドレス 12 に保存した設定データを呼び出します。

*RST (クエリ形式なし)

機能:

プログラマブル電源のすべての制御定をデフォルト値にしますが保存した設定は消去されません。デフォルト状態と等価なパネル設定を以下に示します。

前面パネルの操作部	デフォルト設定
OUTPUT	OFF
CURRENT SET	PSM-2010: 20.000A PSM-3004: 7.000A PSM-6003: 6.000A
VOLTAGE SET VOLTAGE TRIGGER	0
DELAY	0.1 sec
OCP DELAY	0 sec
AUTO SET	OFF
RECALL (memory location)	00
OVP SET	PSM-2010: 22.000V PSM-3004: 32.000V PSM-6003: 65.000V
OCP SET	PSM-2010: 22.000A PSM-3004: 7.700A PSM-6003: 6.600A
STEP SET	0.001V, 0.001A
RECALL RANGE	START 00 CEASE 99 CYCLE 2

構文:

*RST

*SAV

機能:

設定データを特定のメモリーアドレスに保存します(PSS シリーズオよび PSH シリーズはこの機能を持っていません)。

構文:

*SAV <NR1>

<NR1>は、0 ～ 99 の範囲内です。.

例:

*SAV 01 により現在の設定データをメモリーアドレス 1 に保存します。

*SRE

機能:

サービスリクエストイネーブルレジスタ(SRER)の内容を設定します。クエリ形式では SRER の内容を返します。SRER のビット 6 は常にゼロです。SRER のビットは SBR のビットに対応します。.

構文:

*SRE <NR1>

*SRE?

戻り値:

<NR1>は、0～255の範囲内です。.

例:

*SRE 7 は、SRER のビットを00000111に設定します。

*SRE? が3返すときには、SRER は 0000 0011 に設定されています。.

*STB? (クエリのみ)

機能:

(SBR) *STB?によるステータスバイトレジスタ(SBR)の問合わせを行うと、ステータスレジスタ内で 1 にセットされたビットに対応する 10 進数を返します。

構文:

*STB?

戻り値:

<NR1>は、0～255 の範囲内の値です。.

例:

*STB?は SBR が 2 進値 0101 0001 を含む場合.には 81 を返します。

*TRG

機能:

バスをトリガソースととらえ、トリガ信号をトリガサブシステムに送ります。

構文:

*TRG

*PSC

機能:

電源オンステータスクリーンフラグ(Power-On-Status-Clean-Flag)の内容を設定します。クエリ形式では内容を返します。

構文:

*PSC <0|1>

*PSC?

例:

*PSC 1 は、SESER, SRER, OSER, QSER を削除します。

*TST? (クエリのみ)

機能:

セルフテストと RAM および ROM のテスト

構文:

*TST?

戻り値:

0|-300

例:

*TST?により、試験に合格であれば 0 が返されます。

*TST? により、試験に不合格であれば-300 が返されます。

*WAI (クエリ形式なし)

機能:

保留中のオペレーションが完了するまでプロラミング計測器がその先のコマンドあるいはクエリを実行しないようにさせます。.

構文:

*WAI

一般設定用コマンド

:APPLy

機能:

出力電圧と出力電流を設定します。

構文:

:APPLy {<Voltage>|DEF|MIN|MAX}[,{<Current>|DEF|MIN|MAX}]

:APPLy?

例:

:APPLy DEF, MAX

DEF, MIN, MAX を電圧、電流の数値の代わりに用いることができます。

1) DEF を選択すると、電圧、電流レベルは初期値に設定されます。詳細は表 1-1, 1-2, 1-3 を参照してください。
2) MIN を選択すると、電圧、電流共に 0 に設定されます。
3) MAX を選択すると、電圧、電流共に指定レンジ内の最大値に設定されます。詳細は表 1-1, 1-2, 1-3 を参照してください。
4) 入力パラメータが一つのみの場合、電圧レベルとみなされます。
5) :APPLy ? を入力すると、現在の電圧、電流設定値が返されます。

		0-8V/20A レンジ	0-20V/10A レンジ
電圧	設定範囲	0V ～ 8.24V	0V ～ 20.6V
	MAX 値	8.24V	20.6V
	MIN 値	0V	0V
	DEFault 値	0V	0V
	*RST 値	0V	
	MAX Ovp 値	22V	
	MIN 値	0V	
電流	設定範囲	0A ～ 20.6A	0A ～ 10.3A
	MAX 値	20.6A	10.3A

	MIN 値	0A	0A
	DEFault 値	20A	10A
	*RST 値	20A	
	MAX Ovp 値	22A	
	MIN 値	0A	

表 1-1. PSM-2010 設定内容

		0-15V/7A レンジ	0-30V/4A レンジ
電圧	設定範囲	0V ～15.45V	0V ～ 30.9V
	MAX 値	15.45V	30.9V
	MIN 値	0V	0V
	DEFault 値	0V	0V
	*RST 値	0V	
	MAX Ovp 値	32V	
	MIN 値	0V	
電流	設定範囲	0A ～ 7.21A	0A ～ 4.12A
	MAX 値	7.21A	4.12A
	MIN 値	0A	0A
	DEFault 値	7A	4A
	*RST 値	7A	
	MAX Ovp 値	7.7A	
	MIN 値	0A	

表 1-2. PSM-3004 設定内容

		0-30V/6A レンジ	0-60V/3A レンジ
電圧	設定範囲	0V ～ 30.9V	0V ～ 61.8V
	MAX 値	30.9V	61.8V
	MIN 値	0V	0V
	DEFault 値	0V	0V
	*RST 値	0V	
	MAX Ovp 値	65V	
	MIN 値	0V	
電流	設定範囲	0A ～ 6.18A	0A ～ 3.4A
	MAX 値	6.18A	3.4A
	MIN 値	0A	0A
	DEFault 値	6A	3A
	*RST 値	6A	
	MAX Ovp 値	6.6A	
	MIN 値	0A	

表 1-3. PSM-6003 設定内容

:CURRent

機能:

出力電流のリミット値を設定します。

構文:

:CURRent {<current>|MIN|MAX|UP|DOWN}

1) <current> 電流の設定範囲は上記の表 1-1, 1-2, 1-3 を参照してください。

2) MIN, MAX 値の詳細は上記の表 1-1, 1-2, 1-3 を参照してください。

3) Up は電流を1ステップ増加させ、Down は1ステップ減少させます。
CURRent:STEP コマンドを使用してステップ幅を設定します。

例:

:CURRent 2.0 は電流リミットを 2.0000A(PSM-2010)に設定します。

:CURRent MAX は電流リミットを 20.6000A(PSM-2010)に設定します。

:CURRent:STEP 0.1;CURRent UP は電流リミットを 0.1A 増加させます。

:CURRent?

機能:

出力電流のリミット値を返します。

構文:

:CURRent? [MIN|MAX]

戻り値:

浮動小数点形式

例:

:CURRent? は電流リミットが 0.012A の際に +1.20000000E-02 を返します。

:CURRent? MIN は PSM-2010 の場合 +0.00000000E+00 を返します。

:CURRent? MAX は PSM-2010 の場合 +2.06000000E+01 を返します。

:VOLTage

機能:

出力電圧のリミット値を設定します。設定範囲の詳細は上記の表 1-1, 1-2, 1-3 を参照してください。

構文:

:VOLTage {<voltage>|MIN|MAX|UP|DOWN}

1) Up は電流を１ステップ増加させ、Down は１ステップ減少させます。VOLTage:STEP コマンドを使用してステップ幅を設定します。

例:

:VOLTage 2.0 は電圧リミットを 2.0000V(PSM-2010)に設定します。

:VOLTage MAX は電圧リミットを 20.6000V(PSM-2010)に設定します。

:VOLTage:STEP 0.1;VOLTage UP は電圧リミットを 0.1V 増加させます。

:VOLTage?

機能:

出力電圧のリミット値を返します。設定範囲の詳細は上記の表 1-1, 1-2, 1-3 を参照してください。

構文:

:VOLTage? [MIN|MAX]

戻り値:

浮動小数点形式

例:

:VOLTage? は電圧リミットが 0.012V の際に+1.20000000E-02 を返します。

:VOLTage? MIN は PSM-2010 の場合 +0.00000000E+00 を返します。

:VOLTage? MAX は PSM-2010 の場合 +2.06000000+01 を返します。

:CURRent:STEP

機能:

電流のステップ幅を設定または返します。設定範囲の詳細は上記の表 1-1, 1-2, 1-3 を参照してください。

構文:

:CURRent:STEP {<Decimal number>|DEFault}

:CURRent:STEP? [DEFault]

例:

1) :CURRent:STEP 1.0 は電流ステップ幅を 1.0A に設定します。CURRent:UP または CURRent:DOWN コマンドを使用すると、この設定幅で電流値が増加、または減少します。

2) :CURRent:STEP DEF はステップ幅を最小値 0.5mA(PSM-2010 の場合)に設定します。

3) :CURRent:STEP? は現在のステップ幅を返します。

4） :CURRent:STEP? DEF は最小のステップ幅 0.5mA(PSM-2010 の場合)を返します。

:VOLTage:STEP

機能:

電圧ステップ幅を設定または返します。設定範囲の詳細は上記の表 1-1, 1-2, 1-3 を参照してください。

構文:

:VOLTage:STEP {<Decimal number>|DEFault}

:VOLTage:STEP? [DEFault]

例:

1） :VOLTage:STEP 1.0 は電圧ステップ幅を 1.0V に設定します。VOLTage:UP または VOLTage:DOWN コマンドを使用すると、この設定幅で電圧が増加または減少します。

2） :VOLTage:STEP DEF はステップ幅を最小の 0.5mV(PSM-2010 の場合)に設定します。

3） :VOLTage:STEP? は現在のステップ幅を返します。

4） :VOLTage:STEP? DEF は最小ステップ幅 0.5mV(PSM-2010 の場合)を返します。

:CURRent:TRIGgered

機能:

電流のトリガ値を設定または返します。設定範囲の詳細は上記の表 1-1, 1-2, 1-3 を参照してください。

構文:

:CURRent:TRIGgered {<Decimal number>|MIN|MAX}

:CURRent:TRIGgered? [MIN|MAX]

例:

1） :CURRent:TRIGgered 1.0 はトリガ値を 1.0000A に設定します。

2） :CURRent:TRIGgered MAX はトリガ値を最大値に設定します。

3） :CURRent:TRIGgered MIN はトリガ値を最小値に設定します。

4） :CURRent:TRIGgered? は現在のトリガ値を返します。

5） :CURRent:TRIGgered? MAX は最大トリガ値を返します。

6） :CURRent:TRIGgered? MIN は最小トリガ値を返します。

:VOLTage:TRIGgered

機能:

電圧のトリガ値を設定または返します。設定範囲の詳細は上記の表 1-1, 1-2, 1-3 を参照してください。

構文:

:VOLTage:TRIGgered {<Decimal number>|MIN|MAX}

:VOLTage:TRIGgered? [MIN|MAX]

例:

1） :VOLTage:TRIGgered 1.0 はトリガ値を 1.0000V に設定します。

2） :VOLTage:TRIGgered MAX はトリガ値を最大値に設定します。

3） :VOLTage:TRIGgered MIN はトリガ値を最小値に設定します。

4） :VOLTage:TRIGgered? は現在のトリガ値を返します。

5） :VOLTage:TRIGgered? MAX は最大トリガ値を返します。

6） :VOLTage:TRIGgered? MIN は最小トリガ値を返します。

:MEASure:CURRent?（クエリのみ）

機能:

実際の出力電流値を返します。設定範囲の詳細は上記の表 1-1, 1-2, 1-3 を参照してください。

構文:

:MEASure:CURRent?

戻り値:

浮動小数点形式

例:

:MEASure:CURRent? は実際の出力電流が 1.234A の場合、+1.23400000E+00 を返します。

:MEASure?(クエリのみ)

機能:

実際の出力電圧値を返します。設定範囲の詳細は上記の表 1-1, 1-2, 1-3 を参照してください。

構文:

:MEASure?

戻り値:

浮動小数点形式

例:

:MEASure? は実際の出力電圧が 11.55V の場合、+1.15500000E+01 を返します。

:CURRent:PROTection

機能:

過電流保護の値を設定します。設定範囲の詳細は上記の表 1-1, 1-2, 1-3 を参照してください。

構文:

:PROTection:CURRent {<Decimal numbers>|MIN|MAX}

例:

:CURRent:PROTection 10 は過電流保護を 10.000A に設定します。

:CURRent:PROTection MIN は過電流保護を最小値に設定します。

:CURRent:PROTection MAX は過電流保護を最大値に設定します。

:CURRent:PROTection?

機能:

過電流保護の値を返します。設定範囲の詳細は上記の表 1-1, 1-2, 1-3 を参照してください。

構文:

:CURRent:PROTection? [MIN|MAX]

戻り値:

浮動小数点形式

例:

:CURRent:PROTection? は過電流保護の値を返します。

:CURRent:PROTection? MIN は過電流保護の最小値を返します。

:CURRent:PROTection? MAX は過電流保護の最大値を返します。

:CURRent:PROTection:STATe

機能:

過電流保護のオン/オフを設定または返します。

構文:

:CURRent:PROTection:STATe {0|1|OFF|ON}

:CURRent:PROTection?

例:

:CURRent:PROTection:STATe 0 は過電流保護をオフにします。

過電流保護がオンの場合、CURRent:PROTection? コマンドは 1 を返します。

:CURRent:PROTection:TRIPped?（クエリのみ）

機能:

過電流保護の作動状況を返します。

構文:

:CURRent:PROTection:TRIPped?

戻り値:

0|1

例:

過電流保護が作動していると 1 を返し、作動していない場合は 0 を返します。

:CURRent:PROTection:DELay

機能:

過電流保護の作動遅延時間を設定します。

構文:

:CURRent:PROTection:DELay{<Decimal number>|MIN|MAX}

例:

:CURRent:PROTection:DELay 3 は遅延時間を 3 秒に設定します。

:CURRent:PROTection:DELay MIN は遅延時間を最小値 0.1 秒に設定します。
:CURRent:PROTection:DELay MAX は遅延時間を最大値 10.0 秒に設定します。

:CURRent:PROTection:DELay?

機能:

過電流保護の作動遅延時間を返します。

構文:

:CURRent:PROTection:DELay?[MIN|MAX]

例:

:CURRent:PROTection:DELay? は遅延時間を返します。

:CURRent:PROTection:DELay? MIN は最小値 0.1 秒を返します。
:CURRent:PROTection:DELay? MAX は最大値 10.1 秒を返します。

:VOLTage:PROTection

機能:

過電圧保護の値を設定します。設定範囲の詳細は上記の表 1-1, 1-2, 1-3 を参照してください。

構文:

:PROTection:VOLTage {<Decimal numbers>|MIN|MAX}

例:

:VOLTage:PROTection 10 は過電圧保護を 10.000V に設定します。

:VOLTage:PROTection MIN は過電圧保護を最大値 0.000V(PSM-2010 の場合)に設定します。

:VOLTage:PROTection MAX は過電圧保護を最大値 22.000V(PSM-2010 の場合)に設定します。

:VOLTage:PROTection?

機能:

過電圧保護の設定値を返します。設定範囲の詳細は上記の表 1-1, 1-2, 1-3 を参照してください。

構文:

:VOLTage:PROTection? [MIN|MAX]

戻り値:

浮動小数点形式

例:

:VOLTage:PROTection? は過電圧保護の設定値を返します。

:VOLTage:PROTection? MIN は過電圧保護の最小値を返します。

:VOLTage:PROTection? MAX は過電圧保護の最大値を返します。

:VOLTage:PROTection:STATe

機能:

過電圧保護のオン/オフを設定または返します。

構文:

:VOLTage:PROTection:STATe {0|1|OFF|ON}

:VOLTage:PROTection?

例:

:VOLTage:PROTection:STATe 0 は過電圧保護をオフにします。

過電圧保護がオンの場合、VOLTage:PROTection?は 1 を返します。

:VOLTage:PROTection:TRIPped?

機能:

過電圧保護の作動状況を返します。

構文:

:VOLTage:PROTection:TRIPped?

戻り値:

0|1

例:

過電圧保護が作動している場合 1 を返し、そうでない場合は 0 を返します。

:CURRent:PROTection:CLEar

機能:

過電流保護の作動をリセットします。

構文:

:CURRent:PROTection:CLEar

過電流保護が作動すると、メッセージがディスプレイに表示されて他の設定を受け付けなくなります。このコマンドを利用してメッセージを解除し、パネル設定を再度可能にします。

例:

:CURRent:PROTection:CLEar

:VOLTage:PROTection:CLEar

機能:

過電圧保護の作動をリセットします。

構文:

:VOLTage:PROTection:CLEar

過電圧保護が作動すると、メッセージがディスプレイに表示されて他の設定を受け付けなくなります。このコマンドを利用してメッセージを解除し、パネル設定を再度可能にします。

例:

:VOLTage:PROTection:CLEar

:VOLTage:RANGe

機能:

出力電圧のレンジを設定します。

構文:

:VOLTage:RANGe {P8V|P20V|LOW|HIGH} (PSM-2010)

:VOLTage:RANGe {P15V|P30V|LOW|HIGH} (PSM-3004)

:VOLTage:RANGe {P30V|P60V|LOW|HIGH} (PSM-6003)

例:

:VOLTage:RANGe P8V は PSM-2010 の出力レンジを 8V/20A に設定します。

:VOLTage:RANGe HIGH は出力レンジをハイに設定します。

:VOLTage:RANGe?

機能:

出力電圧のレンジを返します。

構文:

:VOLTage:RANGe?

例:

:VOLTage:RANGe? は現在のレンジを返します。

PSM-2010 の場合 “P8V” または “P20V”になります。

PSM-3004 の場合 “P15V” または “P30V”になります。

PSM-6003 の場合 “P30V” または “P60V”になります。

:OUTPut

機能:

出力のオン/オフを設定または返します。

構文:

OUTPut:STATe {0|1|OFF|ON}

OUTPut:STATe?

例:

OUTPut:STATe 1(または “ON”) は出力をオンにします。
出力がオフの場合、OUTPut:STATe? は 0 を返します。

ステータスコマンド

STATus:OPERation:CONDition? (クエリのみ)

機能:

オペレーションレジスタの内容を返します。しかし、プログラマブル電源は状態通知のためにオペレーションレジスタを利用しません。

構文:

STATus:OPERation:CONDition?

戻り値:

<NR1>

例:

STATus:OPERation:CONDition? によって、0 という値を返します。

STATus:OPERation:ENABle

機能:

イネーブルマスクの設定あるいは問合わせを行います。イネーブルマスクによって、イベントレジスタ内のマスクされた状態がサマリービットとして報告されるようにします。イネーブルレジスタ内のビットは1（真）であり関連のイベントビットが1（真）に変化した場合には、関連のサマリービットは1（真）に変化します。これは 16 ビットのレジスタですが、15 ビット（ビット 0～14）のみを使っています。ビット 15 は常に 0 です。

構文:

STATus:OPERation:ENABle <NR1>

STATus:OPERation:ENABle?

<NR1> は、0～32767 の整数です。.

戻り値:

<NR1>

例:

STATus:OPERation:ENABle 32767 により、レジスタの全15ビットを1にセットします。

STATus:OPERation:ENABle?によって 0 という値が返されるときには、レジスタの全15ビットはゼロです。

STATus:OPERation? (クエリのみ)

機能:

オペレーションレジスタの内容を返してレジスタをクリアします。

構文:

STATus:OPERation:CONDition?

戻り値:

<NR1>

例:

STATus:OPERation:CONDition? によって、0 という値を返します。

STATus:PRESet

機能:

オペレーションレジスタとクエスチョナブルレジスタをゼロに設定します。

構文:

STATus:PRESet

STATus:QUEStionable:CONDition? (query only)

機能:

クエスチョナブルレジスタの内容を返します。コンデイションレジスタの読み出し後、内容は保持されます。

構文:

STATus:QUEStionable:CONDition?

戻り値:

<NR1>

例:

STATus:QUEStionable:CONDition? により、0という値を返します。

STATus:QUEStionable?

機能:

クエスチョナブルレジスタの内容を返してレジスタをクリアします。

構文:

STATus:QUEStionable?

<NR1>は、0～ 32767 の整数です。.

戻り値:

<NR1>

その他コマンド

SYSTem:AUTO:CYCLe

機能:

実行の繰返し回数を設定あるいは返します。

構文:

SYSTem:AUTO:CYCLe <NR1>

SYSTem:AUTO:CYCLe?

<NR1>は、0～ 99999 の範囲あるいは無限大です。

戻り値:

<NR1>

例:

SYSTem:AUTO:CYCLe 8 により、自動サイクルをオンにし、設定を8回繰り返します。

SYSTem:AUTO:CYCLe 0 により自動サイクルをオンにし、設定を無限に繰り返します。

If the command SYSTem:AUTO:CYCLe?コマンドがOという値を返す場合には、無限を意味します。

SYSTem:AUTO:DELay

機能:

現在応答しているメモーリテータスの遅延時間を設定します。

構文:

SYSTem:AUTO:DELay <NR1>

SYSTem:AUTO:DELay?

<NR1>は、1～59999 の範囲内で、単位は 100ms です。.

戻り値:

<NR1>

例:

SYSTem:AUTO:DELay 1 は、メモリーの特定部分について自動遅延時間を 100ms に設定します。

SYSTem:AUTO:DELay 1000 は、メモリーの特定部分について自動選択時間を 100 秒に設定します。以前の自動遅延が完了するまで次のメモリーアドレスの自動遅延は実行されません。SYSTem:AUTO:DELay? のコマンドによって 5 という数値が返された場合には、LCD に表示される現在のメモリーアドレスの遅延が 500ms であることを示します。

SYSTem:AUTO:CEASE

機能:

連続自動実行の最終アドレスを設定します。

構文:

SYSTem:AUTO:CEASE <NR1>

SYSTem:AUTO:CEASE?

<NR1>は、1～99 の範囲内であり、START の値以上でなければなりません。.

戻り値:

<NR1>

例:

SYSTem:AUTO:CEASE 8 は、現在のデバイスのメモリーアドレス 8 を自動実行の最終アドレスに設定します。

SYSTem:AUTO:CEASE? のコマンドが 99 という値を返す場合には、アドレス 99 が最終アドレスとして設定されていることを示します。

SYSTem:AUTO:STARt

機能:

連続自動実行の最終アドレスを設定します。

構文:

SYSTem:AUTO:STARt <NR1>

SYSTem:AUTO:STARt?

<NR1>は、0～99 の範囲内であり、END の値以下でなければなりません。.

戻り値:

<NR1>

例:

SYSTem:AUTO:STARt 0 は、現在のデバイスのメモリーアドレス 0 を自動実行の開始アドレスに設定します。

SYSTem:ATUO:STARt?のコマンドが 2 という値を返す場合には、アドレス 2 が開始アドレスとして設定されていることを示します。

SYSTem:AUTO

機能:

自動実行設定の設定または問合わせを行います。

構文:

SYSTem:AUTO:STATe <0|1|OFF|ON>

SYSTem:AUTO:STATe?

<Boolean>は、0(オフ)または 1(オン)の値を取り得ます。

戻り値:

0|1

例:

SYSTem:AUTO:STATe 1 により自動実行をオンにします。

システムコマンド

SYSTem:ERRor? (クエリのみ)

機能:

エラー/イベントキューから次のエラーメッセージを取り出します。問合わせの結果はエラー番号とエラーメッセージ本文です。

構文:

SYSTem:ERRor?

戻り値:

<string>

例:

SYSTem:ERRor? が 0 を返す場合は、「エラーなし」です。

SYSTem:MEMory? (クエリのみ)

機能:

パネルに表示されている現在のメモリーアドレスを読み出します。

構文:

SYSTem:MEMory?

戻り値:

<NR1>

SYSTem:VERSion? (クエリのみ)

機能:

デバイスの SCPI バージョンを返します。

構文:

SYSTem:VERSion?

戻り値:

1994.0

SYSTem:BEEPer:STATe

機能:

警告ブザーのオン/オフを設定または返します。

構文:

SYSTem:BEEPer:STATe {0|1|OFF|ON}

SYSTem:BEEPer:STATe?

例:

SYSTem:BEEPer:STATe ON はブザーをオンにします。

SYSTem:BEEPer

機能:

警告ブザーを一度鳴らします。

構文:

SYSTem:BEEPer

リモート操作モードコマンド

SYSTem:LOCal

機能:

リモート操作モードから、パネル操作モードへ切替えます。フロントパネル上のキー操作が可能になります。

構文:

SYSTem:LOCal

SYSTem:REMote

機能:

パネル操作モードから、リモート操作モードへ切替えます。フロントパネル上のキー操作は LOCAL キーを除いて無効になり、全ての操作はリモート形式で行われます。

構文:

SYSTem:REMote

SYSTem:RWLock

機能:

パネル操作モードから、リモート操作モードへ切替えます。フロントパネル上のキー操作は LOCAL キーを含めて無効になり、全ての操作はリモート形式で行われます。

構文:

SYSTem:RWLock

ディスプレイコマンド

DISPlay

機能:

ディスプレイのオン/オフを設定または返します。

構文:

DISPlay {0|1|OFF|ON}

DISPlay?

例:

DISPlay ON はディスプレイをオンにします。

ディスプレイがオフになると、ERROR メッセージ以外の表示がオフになります。

リモート操作からパネル操作に戻ると、ディスプレイは自動的にオンになります。

DISPlay:TEXT

機能:

ディスプレイ上にメッセージを表示させます。

構文:

DISPlay:TEXT <メッセージ>

DISPlay:TEXT?

例:

DISPlay:TEXT “ABCD” はディスプレイ上に“ABCD”を表示させます。

DISPlay:TEXT? は“ABCD”を返します。

DISPlay:TEXT:CLEar

機能:

ディスプレイ上のメッセージをクリアします。

構文:

DISPlay:TEXT:CLEar

DISPlay:CONTrast

機能:

ディスプレイの輝度を調節します。

構文:

DISPlay:CONTrast {0|1|2|3|4}は輝度を 0(明るい)から 4(暗い)まで調節します。

トリガコマンド

リモート操作でトリガをかけることができます。トリガ操作は複数のコマンドを利用して行います。

1. トリガのソースを指定します(TRIGger:SOURce コマンド)。
2. トリガがかかってから出力を変化させるまでの遅延時間を指定します(TRIGger:DELay コマンド)。
3. トリガ条件後の出力レベルを指定します(INITiate コマンド)。

TRIGger:SOURce

機能:

トリガのソースを設定または返します。

構文:

TRIGger:SOURce {BUS|IMMediate}

TRIGger:SOURce?

例:

TRIGger:SOURce BUS は*TRG コマンドが送られるまで機器をトリガ待ち状態に置きます(ソフトウェアトリガ)。

TRIGger:SOURce IMMediate は出力レベル設定後、直ちにトリガをかけます。

TRIGger:DELay

機能:

トリガ条件後の遅延時間を設定または返します。

構文:

TRIGger:DELay {<Integers>|MIN|MAX}

TRIGger:DELay?

<Integers> は時間を指定します。0~3600s の範囲です。

MIN: 0、MAX: 3600s を指します。

INITiate

機能:

トリガ条件後の出力レベルを指定または返します。

構文:

INITiate

例:

TRIG:SOUR IMM

VOLT:TRIG 5.0

CURR:TRIG 3.0

INIT

上記の設定を実行すると、出力レベルは即座に(IMMidiate)、5V/3A に変化します。

TRIG:SOUR BUS

VOLT:TRIG 5.0

CURR:TRIG 3.0

TRIG:DEL 3.0

INIT

*TRG

上記の設定を実行すると、出力レベルは 3 秒待った後（Delay コマンド）、
*TRG コマンドを受けて 5V/3A に変化します。

7 ステータスとエラーの報告

組になったステータスレジスタによってユーザーがプログラマブル電源内部処理状態を迅速に知ることができます。ステータスレジスタおよびステータスとイベントの報告の方法については SCPI の推奨に従っています。.

システムの構成

ステータスとイベントの報告方法の概要を図 7 に示します。図中の各構成要素は、システム内でのある種のイベントの読み出し、報告、およびイネーブルを行うことのできるいくつかのレジスタとキューを表します。

プログラマブル電源内の特定のイベントがステータスレジスタ内のビットをセットするときには、どのビットがセットされているかを読み取ることによってどの種類のイベントが発生したかを知ることができます。.

ステータスレジスタの各ビットはイネーブルレジスタのビットに対応します。イベントがステータスバイトレジスタに報告されるためには、イネーブルビットが 1 でなければなりません。

サービスリクエスト(SRQ)は最も起こりにくいイベントです。SRQ は GPIB に割込みをリクエストしてシステムコントローラにイベントを知らせます。

ステータスレジスタ

プログラマブル電源には 2 種類のステータスレジスタが組み込まれています。

オペレーションステータスレジスタ(コンデイション、イベント、およびイネーブル)

クエスチョナブルステータスレジスタ(コンデイション、イベント、およびイネーブル)

下位ノードであるクエスチョナブルとオペレーションのそれぞれには、コンデイション、イベント、およびイネーブルの 3 個の 16 ビットレジスタがあります。これら 3 種類のレジスタと各レジスタに関連するコマンドの順序関係を図 8 に示します。.

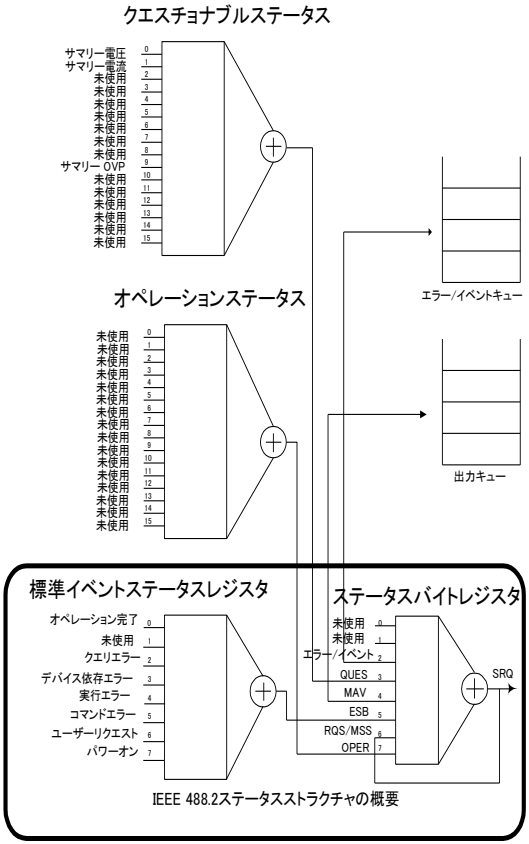

図7、ステータスレジスタと関連の連続との図式的関係

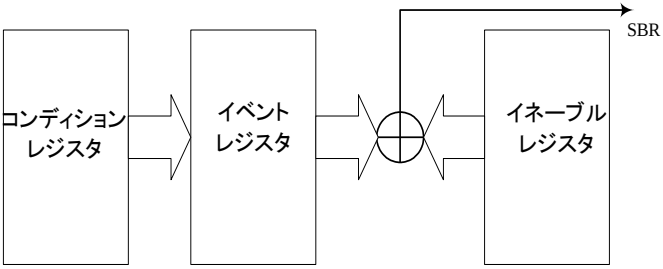

図 8:ステータスレジスタと関連コマンド

コンデイションレジスタは読み出し専用のレジスタであり、計測器の現在の状態をモニタします。コンデイションレジタはリアルタイムで更新され、入力のラッチあるいはバッファは行われません。コンデイションレジスタがモニタする状態が真になると、その状態のビットも真（1）になります。状態が偽のときにはビットは 0 です。読み出し専用のイベントレジスタは、偽から真への状態変化があればラッチします。イベントレジスタのビットがセットされると、コンデイションレジスタの対応するビットが変化しても影響を受けません。ビットはコントローラが読み出すまでセットされたままです。*CLS（クリアステータス)コマンドによってイベントレジスタをクリアします。

クエスチョナブルステータスレジスタ.

16 ビットのクエスチョナブルステータスレジスタのビット割当てを表 4 に示します。

表 4:クエスチョナブルステータスレジスタ

ビット 15	ビット 14	ビット 13	ビット 12	ビット 11	ビット 10	ビット 9	ビット 8
	*未使用	*未使用	*未使用	*未使用	*未使用	サマリー OVP	*未使用
ビット 7	ビット*6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
未使用	未使用	未使用	未使用	未使用	未使用	サマリー 電流	サマリー 電圧

STATus:QUEStionable:CONDtion? コマンドはクエスチョナブルコンディションレジスタを読み出しますが、クリアはしません。

STATus:QUEStionable:EVENt?コマンドはクエスチョナブルイベントステータスレジスタを読み出し、クリアします。

オペレーションステータスレジスタ

16 ビットのオペレーションステータスレジスタのビット割当てを表 5 に示します。

表 5:オペレーションステータスレジスタ

ビット 15	ビット 14	ビット 13	ビット 12	ビット 11	ビット 10	ビット 9	ビット 8
	*未使用	*未使用	*未使用	*未使用	*未使用	未使用	*未使用
ビット 7	ビット*6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
未使用	未使用	未使用	未使用	未使用	未使用	未使用	未使用

ステータスレジスタ

プログラマブル電源には IEEE-488.2 の規格で定義されている 2 つのステータスレジスタが含まれています。

ステータスバイトレジスタ (SBR)

標準イベントステータスレジスタ (SESR)

ステータスバイトレジスタ (SBR):SBR (表 6) は他のすべてのレジスタおよびキューのステータスを要約します。

表 6:ステータスバイトレジスタ (SBR)

ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
OPER	RQS/MSS	ESB	MAV	QUES	E/E	未使用	未使用

ビット 0 とビットは使用されていません。このため、これらのビットは常に 0 です。ビット 2（エラーおよびイベント）はエラーイベントキュー内に読み出しを待っているエラーコードがあることを示します。ビット 3 (QUES（クエスチョナブル）)は QESR(クエスチョナブルイベントステータスレジスタ)の要約ビットです。このビットが 1 のときには、ステータスがイネーブルされ, QUES に存在することを示します。ビット 4（MAV (メッセージアベィラプル)）は出力が出力キュー内で利用可能であることを示します。ビット 5 (ESB（イベントステータスビット）)は標準イベントステータスレジスタ(SESR)の要約ビットです。このビットが 1 のときには、ステータスがイネーブルされ SESR 内に存在することを示します。ビット 6 (RQS,（リクエストサービス）)はシリアルポールから得られ、プログラマブル電源が GPIB コントローラからのサービスを要求していることを表示します。ビット 7 (OPER（オペレーション）)は OESR (オペレーションイベントステータスレジスタ)) の要約ビットです。

SBR.の内容を読み出すためにはシリアルポールあるいは*STB? クエリを使います。SBR 内のビットが標準イベントステータスレジスタ(SESR)および出力キューの内容に従って設定およびクリアされます。

標準イベントステータスレジスタ*(SESR)*: SESR を表 7 に示します。

表 7: 標準イベントステータスレジスタ (SESR): SESR を表 7 に示します。

ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
PON	URQ	CME	EXE	DDE	QYE	未使用	OPC

ビット 0 (OPC（オペレーション完了）)はオペレーションが完了したことを示します。このビットは*OPC コマンドの後、保留中のオペレーションがすべて完了したときにアクティプになります。ビット1は常に 0 です。ビット 2 (QYE（クエリエラー）)はコマンドまたはクエリのプロトコルエラーを示します。ビット 3 (DDE（デバイスエラー）)はデバイスエラーが起こったことを示します。ビット 4(EXE（実行エラー）)はプログラマブル電源がコマンドまたはクエリを実行しているときにエラーが発生したことを示します。ビット 5(CME（コマンドエラー）)はプログラマブル電源がコマンドまたはエラーを解析しているときにエラーが発生したことを示します。ビット 6 (USR（ユーザーリクエスト）)は LOCAL ボタンが押されたことを示します。ビット7 (PON（パワーオン）)は電源がオンにされたことを示します。

SESR を読み出すためには *ESR?クエリを使用します。SESR を読み出して、新しいイベントの情報を入れることができるようにレジスタのビットをクリアします。

イネーブルレジスタ

イネーブルレジスタは、特定のイベントをステータスバイトレジスタおよび SRQ に報告するかどうかを判定します。プログラマブル電源には以下のイネーブルレジスタがあります。

イベントステータスイネーブルレジスタ (ESER)

(オペレーションイネーブルレジスタ)

(クエスチョナブルイネーブルレジスタ)

サービスリクエストイネーブルレジスタ(SRER)

イネーブルレジスタのビットの 1 つが 1 であり、ステータスレジスタ内の対応するビットが 1 のときには、イネーブルレジスタ論理 AND 演算を行い、ステータスバイレジスタ内でセットするビットを制御する出力は 1 です。.

各種のコマンドによりイネーブルレジスタ内のビットがセットされます。イネーブルレジスタとセットするコマンドを以下に示します。.

イベントステータスイネーブルレジスタ (ESER):ESER はどの種類のイベントが SBR 内のイベントステータスビット(ESB)に要約されるかを制御します。ESER 内のビットは SESR 内のビットに対応します。.

ESER のビットを設定するためには*ESE コマンドを使用します。 ESER を読み出すためには*ESE?クエリを使用します。

オペレーションイネーブルレジスタ:オペレーションイネーブルレジスタがプログラマブル電源内にありますが、オペレーションレジスタは状態を報告しません。

クエスショナプルイネーブルレジス: クエスショナプルイネーブルレジスタはどの種類のイベントが SBR 内の QUES ステータスビットに要約されるかを制御します。クエスチョナプナプルイネーブルレジスタ内のビットを設定するためには STATus:QUEStionable:ENABle コマンドを使用します。読み出すためには STATus:QUEStionable:ENABle?クエリを使用します。

サービスリクエストイネーブルレジスタ (SRER): SRER は SBR 内のどのビットがサービスリクエストを生成するかを制御します。.

SRER を設定するためには*SRE コマンドを使用します。SRER を読み出すためには*SRE?クエリを使用します。

キュー

出力キューがプログラマブル電源に組み込まれています。

出力キュー:プログラマブル電源はクエリに対する応答を IEEE 488.2 プログラマブル電源がメッセージターミネータの後で新しいコマンドまたはクエリのメッセージを受け取ると、プログラマブル電源はその都度このキューをクリアしてリセットします。以前のクエリに対する応答が失われないようにするために、パソコンがクエリの応答を読み出した後で新しいコマンド(またはクエリ)を送信しなければなりません。

エラー/イベントキュー

エラーまたはイベントが発生すると、出力キューがメッセージを保存します。出力キューはメッセージを FIFO (ファーストインファーストアウト)方式で保存して報告します。

SYSTem:ERRor?クエリにより出力キューから次の項目を読み出します。出力キューがオーバフローすると、エラーメッセージが-350(キューオーバフロー)になり、キューが読み出されてクリアされるまで次のメッセージの保存と報告を行うことができません。

エラーメッセージ

エラーメッセージプログラマブル電源の SCPI エラーメッセージを表 8 に示します。

表 8 -プログラマブル電源のエラーメッセージ

エラーコード	説明
-101	文字エラー
-102	構文エラー
-103	セパレータエラー
-104	データタイプエラー
-105	GET 利用不可
-108	パラメータ不可
-109	パラメータ不在
-112	構文長さ規定外
-113	ヘッダー無指定

-121	数字の箇所に文字入力
-123	数字の長さ規定外
-124	桁数規定外
-128	数字入力不可
-131	末尾の形式不一致
-134	末尾の長さ規定外
-138	末尾不可
-141	文字形式規定外
-144	文字の長さ規定外
-148	文字入力不可
-151	データストリング形式規定外
-158	データストリング入力不可
-160～-168	ブロックデータエラー
-170～-178	表現形式エラー
-211	トリガ無視
-213	トリガ開始無視
-221	設定矛盾
-222	範囲外データ
-223	データ過大
-224	パラメータ規定外
-330	自己テスト不合格
-350	エラー過大
-410	クエリが割り込みを受けた
-420	クエリが未定
-430	クエリのデッドロック発生
-440	初期反応の後、クエリ未定

製品についてのご質問等につきましては下記までお問い合わせください。

お問合せ先

株式会社テクシオ・テクノロジー

本社：〒222-0033　横浜市港北区新横浜 2-18-13
藤和不動産新横浜ビル 7F

[HOME PAGE] ：http://www.texio.jp/

E-Mail：info@texio.co.jp

アフターサービスに関しては下記サービスセンターへ

サービスセンター：

〒222-0033　横浜市港北区新横浜 2-18-13
藤和不動産新横浜ビル 8F

TEL.　045-620-2786　　FAX.045-534-7183