

プログラミング・ガイド
ダイナミック測定dcソース
Agilentモデル66312A, 66332A
システムDC電源
Agilentモデル6631B, 6632B, 6633B, 6634B
6611C, 6612C, 6613C, 6614C



Agilent Technologies

Innovating the HP Way

Agilent Part No. 5962-8231

Microfiche No. 5962-8232

Printed in USA April, 2000

安全性に関するガイドライン

『操作ガイド』の初めに「安全性について」のページがあります。コントローラからdcソースの操作設定を行う前に、そのページの内容をよくお読みください。

出版履歴

本書の版と現在のリビジョンを以下に示します。再販にあたって大きな訂正や改訂がなければ、再版の印刷日は変更されない場合もあります。改訂版の場合、印刷日は変更されます。改訂した版には、前回の印刷日以降に新たに追加されたり、訂正された情報がすべて盛り込まれています。次のリビジョンの出版までに変更が生じた場合、本書には変更シートが添付されます。

本書には、著作権によって保護された所有権のある情報が含まれています。すべての権利は留保されています。アジレント・テクノロジー社の事前の同意なしに本書を複写、複製あるいは他の言語に翻訳することは禁止されています。本書の内容は、予告なしに変更される場合があります。

© Copyright 1997 Agilent Technologies

第1版 1997年11月

改訂第2版 2000年1月

納入後の保証について

- ★ 保証の期間は、ご購入時に当社よりお出しした見積書に記載された期間とします。保証サービスは、当社の定める休日を除く月曜日から金曜日までの、午前8時45分から午後5時30分の範囲で無料で行います。当社で定めたシステム製品については出張修理を行い、その他の製品については当社へご返却いただいた上での引取り修理となります。当社が定める地域以外における出張修理対象製品の修理は、保証期間中においても技術者派遣費が有料となります。
- ★ ソフトウェア製品の保証は上記にかかわらず、下記に定める範囲とさせていただきます。
 - ソフトウェア製品及びマニュアルは当社が供給した媒体物の破損、資料の落丁およびプログラム・インストラクションが実行できない場合のみ保証いたします。
 - バグ及び前記以外の問題の解決は、別に締結するソフトウェア・サポート契約に基づいて実施されます。
- ★ 次のような場合には、保証期間内でも修理が有料となります。
 - 取扱説明書等に記載されている保証対象外部品の故障の場合。
 - 当社が供給していないソフトウェア、ハードウェア、または補用品の使用による故障の場合。
 - お客様の不適合または不十分な保守による故障の場合。
 - 当社が認めていない改造、酷使、誤使用または誤操作による故障の場合。
 - 納入後の移設が不適切であったための故障または損傷の場合。
 - 指定外の電源 (電圧、周波数) 使用または電源の異常による故障の場合。
 - 当社が定めた設置場所基準に適合しない場所での使用、および設置場所の不適合な保守による故障の場合。
 - 火災、地震、風水害、落雷、騒動、暴動、戦争行為、放射能汚染、およびその他天災地変等の不可抗力的事故による故障の場合。
- ★ 当社で取扱う製品は、ご需要先の特定目的に関する整合性の保証はいたしかねます。また、そこから生ずる直接的、間接的損害に対しても責任を負いかねます。
- ★ 当社で取扱う製品を組み込みあるいは転売される場合は、最終需要先における直接的、間接的損害に対しては責任を負いかねます。
- ★ 製品の保守、修理用部品の供給期間は、その製品の製造中止後最低5年間とさせていただきます。本製品の修理については取扱説明書に記載されている最寄りの事業所へお問合わせください。

— 原 典 —

本書は"Programming Guide Dynamic Measurement DC Source Agilent Models 66312A, 66332A System DC Power Supply Agilent Models 6631B, 6632B, 6633B, 6634B 6611C, 6612C, 6613C, 6614C" (Part No. 5962-8198) (Printed in USA, January 2000)を翻訳したものです。

詳細は上記の最新マニュアルを参照して下さい。

— ご 注 意 —

- 本書に記載した内容は、予告なしに変更することがあります。
- 当社は、お客様の誤った操作に起因する損害については、責任を負いかねますのでご了承ください。
- 当社では、本書に関して特殊目的に対する適合性、市場性などについては、一切の保証をいたしかねます。
- また、備品、パフォーマンス等に関連した損傷についても保証いたしかねます。
- 当社提供外のソフトウェアの使用や信頼性についての責任は負いかねます。
- 本書の内容の一部または全部を、無断でコピーしたり、他のプログラム言語に翻訳することは法律で禁止されています。
- 本製品パッケージとして提供した本マニュアル、フレキシブル・ディスクまたはテープ・カートリッジは本製品用だけにお使いください。プログラムをコピーをする場合はバックアップ用だけにしてください。プログラムをそのままの形で、あるいは変更を加えて第三者に販売することは固く禁じられています。

アジレント・テクノロジー株式会社

許可なく複製、翻案または翻訳することを禁止します。

Copyright © Agilent Technologies, Inc. 2000

Copyright © Agilent Technologies Japan, Ltd. 2000

All rights reserved. Reproduction, adaptation, or translation without prior written permission is prohibited.

目次

安全性に関するガイドライン	2
出版履歴	2
目次	5
1 - はじめに	9
本書の内容	9
マニュアルの概要	9
その他の参考文献	10
GPIBに関する参考文献	10
SCPIリファレンス	10
2 - プログラミング入門	11
VXIplug&play電源測定器ドライバ	11
サポートするアプリケーション	11
システム要件	11
ドライバのダウンロードとインストール	11
オンライン・ヘルプへのアクセス	12
dcソースのGPIB機能	12
GPIBアドレス	12
dcソースのRS-232機能	12
RS-232データ・フォーマット	13
RS-232フロー制御	13
RS-232プログラミング例	13
RS-232に関するトラブルシューティング	14
SCPIの概要	15
本書で使用されている表記法	15
SCPIコマンドの種類	16
複数のコマンドから成る単一のメッセージ	16
サブシステム間の移動	17
共通コマンドの挿入	17
クエリの使用法	17
SCPIメッセージの種類	18
メッセージ・ユニット	18
ヘッダ	18
クエリ・インジケータ	18
メッセージ・ユニット・セパレータ	19
ルート指示子	19
メッセージ・ターミネータ	19
SCPIデータ・フォーマット	20
数値データ・フォーマット	20
接尾語と乗数	20
応答データ型	20
SCPIコマンドの完了	21
デバイス・クリアの使用法	21
3 - dcソースの設定	23
はじめに	23
出力の設定	23
電源投入時の初期化	23
出力のイネーブル	23
出力電圧	24
出力電流	24
出力変動のトリガ	25

SCPIにおけるトリガの名称	25
出力トリガ・システム・モデル	25
電圧または電流トリガ・レベルの設定	25
出力トリガ・システムの起動	26
トリガの生成	26
測定の実施	27
電圧および電流の測定	27
内部トリガ測定	29
SCPIにおけるトリガの名称	29
測定トリガ・システム・モデル	29
測定トリガ・システムの起動(Agilent 66312A, 66332Aの場合のみ)	29
測定トリガ源の選択(Agilent 66312A, 66332Aの場合のみ)	30
測定トリガの生成(Agilent 66312A, 66332Aの場合のみ)	30
出力パルスの測定(Agilent 66312A, 66332Aの場合のみ)	32
電流ディテクタ	32
パルス測定クウェリ	32
測定サンプルの制御	33
電圧または電流サンプリング・レートの変更	33
複数測定(Agilent 66312A, 66332Aの場合のみ)	33
プリイベントおよびポストイベント・トリガ(Agilent 66312A, 66332Aの場合のみ)	34
パルス測定例(Agilent 66312A, 66332Aの場合のみ)	34
ステータス・レジスタの設定	36
パワーオン条件	36
動作ステータス・グループ	38
クエスチョナブル・ステータス・グループ	39
標準イベント・ステータス・グループ	39
ステータス・バイト・レジスタ	40
サービス割込みの原因の究明	40
動作ステータス・イベントおよびクエスチョナブル・ステータス・イベントの処理	41
ステータス遷移の両相のモニタ	41
インヒビット/フォールト・インジケータ	42
リモート・インヒビット(RI)	42
個別フォールト・インジケータ(DFI)	42
インヒビット/フォールト・ポートをデジタルI/Oとして使用するには	42
DFIプログラミング例	43
4 - 言語辞書	45
概要	45
サブシステム・コマンド	45
共通コマンド	49
設定パラメータ	49
校正コマンド	50
CALibrate:CURRent	50
CALibrate:CURRent:NEGative	50
CALibrate:CURRent:MEASure:LOWRange	50
CALibrate:CURRent:MEASure:AC	50
CALibrate:DATA	51
CALibrate:LEVel	51
CALibrate:PASSword	51
CALibrate:SAVE	51
CALibrate:STATe	52
CALibrate:VOLTag	52
CALibrate:VOLTag:PROTection	52
測定コマンド	53
MEASure:ARRay:CURRent? FETCh:ARRay:CURRent?	53
MEASure:ARRay:VOLTag? FETCh:ARRay:VOLTag?	53

MEASure:CURRent? FETCh:CURRent?	54
MEASure:CURRent:ACDC? FETCh:CURRent:ACDC?	54
MEASure:CURRent:HIGH? FETCh:CURRent:HIGH?	54
MEASure:CURRent:LOW? FETCh:CURRent:LOW?	55
MEASure:CURRent:MAXimum? FETCh:CURRent:MAXimum?	55
MEASure:CURRent:MINimum? ETCh:CURRent:MINimum?	55
MEASure:VOLTagE? FETCh:VOLTagE?	56
MEASure:VOLTagE:ACDC? FETCh:VOLTagE:ACDC?	56
MEASure:VOLTagE:HIGH? FETCh:VOLTagE:HIGH?	56
MEASure:VOLTagE:LOW? FETCh:VOLTagE:LOW?	57
MEASure:VOLTagE:MAXimum? FETCh:VOLTagE:MAXimum?	57
MEASure:VOLTagE:MINimum? FETCh:VOLTagE:MINimum?	57
SENSe:CURRent:RANGe	58
SENSe:CURRent:DETEctor	58
SENSe:FUNCTion	59
SENSe:SWEep:OFFSet:POINts	59
SENSe:SWEep:POINts	59
SENSe:SWEep:TINterval	59
SENSe:WINDow	60
出力コマンド	61
OUTPut	61
OUTPut:DFI	61
OUTPut:DFI:SOURce	61
OUTPut:PON:STATe	62
OUTPut:PROTEction:CLEar	62
OUTPut:PROTEction:DELay	62
OUTPut:RELay	63
OUTPut:RELay:POLarity	63
OUTPut:RI:MODE	63
[SOURce:]CURRent	64
[SOURce:]CURRent:TRIGger	64
[SOURce:]CURRent:PROTEction:STATe	64
[SOURce:]DIGital:DATA	65
[SOURce:]DIGital:FUNCTion	65
[SOURce:]VOLTagE	65
[SOURce:]VOLTagE:ALC:Bandwidth? [SOURce:]VOLTagE:ALC:BWIDth?	66
[SOURce:]VOLTagE:TRIGger	66
[SOURce:]VOLTagE:PROTEction	66
ステータス・コマンド	67
STATus:PRESet	67
STATus:OPERation?	67
STATus:OPERation:CONDition?	68
STATus:OPERation:ENABle	68
STATus:OPERation:NTR STATus:OPERation:PTR	68
STATus:QUEStionable?	69
STATus:QUEStionable:CONDition?	69
STATus:QUEStionable:ENABle	69
STATus:QUEStionable:NTR STATus:QUEStionable:PTR	70
*CLS	70
*ESE	71
*ESR?	71
*OPC	71
*PSC	72
*SRE	72
*STB?	73

*WAI	73
システム・コマンド	74
DISPlay	74
DISPlay:MODE	74
DISPlay:TEXT	74
SYSTem:ERRor?	75
SYSTem:LANGuage	75
SYSTem:VERsion?	75
SYSTem:LOCal	76
SYSTem:REMote	76
SYSTem:RWLock	76
*IDN?	76
*OPT?	77
*RCL	77
*RST	77
*SAV	78
*TST?	78
トリガ・コマンド	79
ABORt	79
INITiate:SEQuence INITiate:NAME	79
INITiate:CONTInuous:SEQuence1 INITiate:CONTInuous:NAME	79
TRIGger	80
TRIGger:SOURce	80
TRIGger:SEQuence2 TRIGger:ACQuire	80
TRIGger:SEQuence2:COUNT:CURRent TRIGger:ACQuire:COUNT:CURRent	81
TRIGger:SEQuence2:COUNT:VOLTage TRIGger:ACQuire:COUNT:VOLTage	81
TRIGger:SEQuence2:HYSTeresis:CURRent TRIGger:ACQuire:HYSTeresis:CURRent	82
TRIGger:SEQuence2:HYSTeresis:VOLTage TRIGger:ACQuire:HYSTeresis:VOLTage	82
TRIGger:SEQuence2:LEVel:CURRent TRIGger:ACQuire:LEVel:CURRent	83
TRIGger:SEQuence2:LEVel:VOLTage TRIGger:ACQuire:LEVel:VOLTage	83
TRIGger:SEQuence2:SLOPe:CURRent TRIGger:ACQuire:SLOPe:CURRent	84
TRIGger:SEQuence2:SLOPe:VOLTage TRIGger:ACQuire:SLOPe:VOLTage	84
TRIGger:SEQuence2:SOURce TRIGger:ACQuire:SOURce	85
TRIGger:SEQuence1:DEFine TRIGger:SEQuence2:DEFine	85
*TRG	85
A - SCPIとの適合性について	87
SCPIバージョン	87
SCPI準拠コマンド	87
非SCPIコマンド	88
B - 互換言語	89
概要	89
C - エラー・メッセージ	95
エラー番号のリスト	95
D - サンプル・プログラム	99
概要	99
プログラムにおけるGPIBアドレスの割当て	99
DOSドライバの種類	99
エラー処理	100
Agilent BASICコントローラ	100
例1. HP Vectra PCコントローラ(Agilent 82335インタフェース搭載)	100
例2. IBMコントローラ(ナショナル・インタフェース搭載)	102
例3. Agilent BASIC搭載のコントローラ	104
索引	105

はじめに

本書の内容

本書では、以下のGPIBプログラマブルDC電源シリーズのリモート・プログラミング方法について説明します。

- Agilent 66312A
- Agilent 66332A
- Agilent 6631B/6632B/6633B/6634B
- Agilent 6611C/6612C/6613C/6614C

各章と付録の内容を以下に示します。

第1章 本書の概要

第2章 SCPIメッセージの構造、構文およびデータ・フォーマットの概要。SCPIプログラムの例

第3章 SCPIコマンドを使用したdcソースの設定方法の概要

第4章 SCPIコマンドの辞書

付録A SCPIの適合性について

付録B 代替互換プログラミング言語の使用法

付録C エラー・メッセージ

マニュアルの概要

本『プログラミング・ガイド』のほか、以下のマニュアルがdcソースを使用する際に役立ちます。

- 『Agilent 66312AおよびAgilent 6611C/6612C/6613C/6614Cユーザーズ・ガイド』。仕様と補足特性、フロント・パネルの使用法、測定器への接続方法および校正手順が記載されています。
- 『Agilent 66332AおよびAgilent 6631B/6632B/6633B/6634Bユーザーズ・ガイド』。仕様と補足特性、フロント・パネルの使用法、測定器への接続方法および校正手順が記載されています。

その他の参考文献

GPIBに関する参考文献

GPIBに関しては、『Agilent BASIC』『GPIB Command Library for MS DOS』などコントローラのプログラミング・マニュアルが大変参考となります。SCPI以外のコマンド(Local Lockoutなど)については、これらのマニュアルを参照してください。

GPIBインタフェースに関しては、次の2冊の公式文書が発行されています。

- ANSI/IEEE Std. 488.1-1987 IEEE Standard Digital Interface for Programmable Instrumentation。GPIBインタフェースの技術的な詳細が定義されています。ここに掲載されている情報の多くは、ほとんどのプログラマには不要なものですが、本書や関連文書に使用されている専門用語を明確に理解するためには有用です。
- ANSI/IEEE Std. 488.2-1987 IEEE Standard Codes, Formats, Protocols, and Common Commands。高度なプログラムを作成する場合に参考とされることをお勧めします。ある特定の種類のSCPIメッセージ・フォーマット、データ型または共通コマンドの正確な定義を知りたい場合に便利です。

上記の2冊のドキュメントをご希望の場合には、IEEE(Institute of Electrical and Electronics Engineers: アメリカ電気電子技術者協会), 345 East 47th Street, New York, NY 10017, USAまでお問い合わせください。WEBのアドレスはwww.ieee.orgです。

SCPIリファレンス

SCPIのプログラミングの参考文献は、以下のとおりです。

- Standard Commands for Programmable Instruments Volume 1, Syntax and Style
- Standard Commands for Programmable Instruments Volume 2, Command References
- Standard Commands for Programmable Instruments Volume 3, Data Interchange Format
- Standard Commands for Programmable Instruments Volume 4, Instrument Classes

上記のドキュメントの入手先は以下のとおりです。

Fred Bode, Executive Director, SCPI Consortium, 8380 Hercules Drive, Suite P3, Ls Mesa, CA 91942, USA

プログラミング入門

VXIplug&play電源測定器ドライバ

Microsoft Windows 9.5およびWindows NT用VXIplug&play電源測定器ドライバは、ウェブ・サイト<http://www.agilent.com/find/drivers>で入手可能になりました。この測定器ドライバは、アジレント・テクノロジーの測定器にハイレベルのプログラミング・インタフェースを提供します。VXIplug&play測定器ドライバを使えば、SCPIコマンド・ストリングで測定器をプログラミングする必要がなくなります。測定器のファンクション・コールはVISA I/Oライブラリ上で同時に機能するため、1つの測定器ドライバを複数のアプリケーション環境で使用できます。

サポートするアプリケーション

- Agilent VEE
- Microsoft Visual BASIC
- Microsoft Visual C/C++
- Borland C/C++
- National Instruments LabVIEW
- National Instruments LabWindows/CVI

システム要件

- Microsoft Windows 95
- Microsoft Windows NT 4.0
- HP VISA revision F.01.02
- National Instruments VISA 1.1

ドライバのダウンロードとインストール

注記： VXIplug&play測定器ドライバをインストールする前に、上記のサポート対象アプリケーションのどれかがコンピュータにインストール済みであり、起動していることを確認してください。

1. アジレント・テクノロジーのウェブ・サイト<http://www.agilent.com/find/drivers>にアクセスします。
2. ドライバをインストールする測定器を選択します。
3. Windows 9.5またはWindows NT用のドライバをクリックし、PCに実行ファイルをダウンロードします。
4. ウェブ・サイトからダウンロードしたファイルを実行します。**Start**メニューから**Run** <path>:\agxxxx.exeを選択します。<path>はファイルが収容されたディレクトリ・パス、agxxxxはダウンロードした測定器ドライバです。

2 - プログラミング入門

5. 画面の指示に従ってソフトウェアをインストールします。ほとんどの場合、インストール時の選択項目はデフォルトです。readme.txtファイルには、オンライン・ヘルプに記載されていない製品のアップデートや修正情報が収容されています。このファイルはインストール後、どのテキスト・エディタでもオープンして読むことができます。
6. *VXIplug&play*測定器ドライバを使用する際は、*VXIplug&play*オンライン・ヘルプ「プログラミングの紹介」にある説明に従ってください。

オンライン・ヘルプへのアクセス

ドライバには、オンライン・ヘルプ・リファレンスが付属しています。リファレンスは、Agilent VEE、LabVIEW、LabWindowsで測定器ドライバを使用する方法について述べたものです。全ファンクション・コールの詳細な説明に加えて、C/C++およびVisual BASICのプログラム例が納められています。

- デフォルトのVxipnpスタート・フォルダを選択しているときにオンライン・ヘルプにアクセスするには、StartボタンをクリックしてPrograms | Vxipnp | Agxxxx Help (32ビット)を選択します。agxxxxは測定器ドライバです。

dcソースのGPIB機能

GPIBアドレスの設定を除くすべてのdcソース機能は、GPIBを介して設定できます。dcソースのIEEE 488.2機能については、『ユーザーズ・ガイド』の「仕様」の表を参照してください。

GPIBアドレス

dcソースは、フロント・パネルで設定した1つのGPIBアドレスによって制御することができます。GPIBアドレスを設定するには、フロント・パネルの**Address**キーを押してから、エントリ・キーを使用してアドレスを入力します。GPIBアドレスは不揮発性メモリにストアされます。

dcソースのRS-232機能

dcソースにはRS-232プログラミング・インタフェースがあり、フロント・パネルの**Address**キーの下にあるコマンドによってアクティブになります。SCPIコマンドやCOMPatibilityコマンドはすべて、RS-232プログラミングで使用できます。RS-232インタフェースを選択すると、GPIBインタフェースがディスエーブルになります。

EIAのRS-232規格にはデータ端末機器(DTE)とデータ通信機器(DCE)の相互接続が定義されています。dcソースは、DTEとして設計されており、ヌル・モデム・ケーブルを使ってPCのCOMポートなど別のDTEに接続することができます。

注記： プログラム内のRS-232の設定はフロント・パネルのAddressメニューで指定した設定と整合している必要があります。設定を変更する必要がある場合は、フロント・パネルのAddressキーを押してください。

RS-232データ・フォーマット

RS-232データは10ビット・ワードで、開始ビットと停止ビットが1つずつあります。開始ビットと停止ビットの数を設定することはできません。しかし、フロント・パネルのAddressキーを使用して、以下のパリティ・オプションを選択することができます。

EVEN	7データ・ビット+偶数パリティ
ODD	7データ・ビット+奇数パリティ
MARK	7データ・ビット+マーク・パリティ (パリティは常に真)
SPACE	7データ・ビット+スペース・パリティ (パリティは常に偽)
NONE	パリティなしの8ビット

パリティ・オプションは不揮発性メモリにストアされます。

ボーレート

フロント・パネルのAddressキーを使用して、以下のボーレートのうちの1つを選択します。

300 600 1200 2400 4800 9600

ボーレートは不揮発性メモリにストアされます。

RS-232フロー制御

RS-232インタフェースは、複数のフロー制御オプションをサポートしています。これらのフロー制御オプションは、フロント・パネルのAddressキーを使用して選択します。いずれの場合も、dcソースは、コントローラがホールドオフをアサートした後に最大5個の文字を送信します。また、dcソースは、ホールドオフをアサートした後に15個の追加文字を受け取ることができます。

XON-XOFF	ASCII制御コードDC3(10進法の19)を使用してホールドオフをアサートし、制御コードDC1(10進法の17)を使用してホールドオフを解除するソフトウェア・ハンドシェイク
RTS-CTS	dcソースは、入力バッファがほぼ一杯である場合には送信要求(RTS)ラインをアサートしてホールドオフ信号を送信し、送信クリア(CTS)ラインをコントローラからのホールドオフ信号として解釈します。
DTR-DSR	dcソースは、入力バッファがほぼ一杯である場合にはデータ端末レディ (DTR) ラインをアサートしてホールドオフ信号を送信し、データ・セット・レディ (DSR) ラインをコントローラからのホールドオフ信号として解釈します。
NONE	フロー制御はありません。

フロー制御オプションは不揮発性メモリにストアされます。

RS-232プログラミング例

以下のプログラムは、RS-232を使用して出力電圧および電流を設定したり、モデル番号や出力電圧をリードバックする電源をプログラムするための方法を示したものです。このプログラムは、Microsoft QBasicを使用するすべてのコントローラ上で実行できるように書かれています。

注記： 本電源はRS232用に構成してください。また、ボーレートとパリティはコントローラと同じ値に設定してください。

2 - プログラミング入門

```
' RS232を介した読み書き用プログラム
' 電源を9600ボー、偶数パリティ、RS232用に構成します
' シリアル・ポートを構成します
'   9600ボー
'   7ビット・データ
'   2ストップ・ビット
'   送信要求を無視します
'   キャリア検出を無視します
'   偶数パリティ           ャ Vectra basicでは必要とされ、QBasicでは無視されます
'   改行文字を送信します
'   シリアルI/O用に1000文字バッファを確保します
'
DECLARE FUNCTION gets$()
CLS                               ' 画面をクリアします
LOCATE 1, 1                       ' カーソルを左上に位置づけます
                                ' COMポートを構成します
OPEN "com1:9600,e,7,2,rs,cd,pe,lf" FOR RANDOM AS #1 LEN = 1000
PRINT #1, "OUTPUT ON"           ' 出力をオンにしてから、電圧と電流を設定します
PRINT #1, "VOLT 6"              ' 電圧を6ボルトに設定します
PRINT #1, "CURR .5"             ' 電流を0.5アンペアに設定します
PRINT #1, "*IDN?"              ' 電源識別文字列を問い合わせます
PRINT gets$                     ' gets$関数に進み、返されたデータをプリントします
PRINT #1, MEAS"VOLT?"; volt     ' 電源電圧を問い合わせます
Volt = VAL(gets$)              ' gets$文字列を値に変換します
PRINT gets$                     ' 電圧の値をプリントします
END                               ' メイン・プログラムの終わり

FUNCTION gets$                   ' デバイス#1から復帰改行終端文字列を入手します
C$ = ""                          ' C$をヌルに設定します
WHILE c$ <> CHR$(10)             ' 改行で停止するようにループを設定します
    C$ = INPUT$(1, #1)          ' 1ビットをファイル#1に読み込みます
    Resp$ = resp$ + c$         ' ビットをそれ以前のビットと連結します
WEND                              ' WHILEループの終わり
gets$ = resp$                    ' 応答をgets$に割り当てます
END FUNCTION
```

RS-232に関するトラブルシューティング

RS-232インタフェースを介した通信に問題がある場合には、以下のことを確認します。

- コンピュータとDC電源は、同じボーレート、パリティ、データ・ビット数、およびフロー制御オプション用に構成されていなければなりません。DC電源は1スタート・ビットおよび1ストップ・ビット用に構成してください(これらの値は固定です)。
- RS-232コネクタに示されている適切なインタフェース・ケーブル、またはアダプタが使用されていなければなりません。ご使用のシステムに適合のコネクタが付いているケーブルでも、内部の配線が誤っている場合もありますので注意してください。
- インタフェース・ケーブルは、コンピュータの適切なシリアル・ポート(COM1、COM2など)に接続されていなければなりません。

SCPIの概要

SCPI(Standard Commands for Programmable Instruments)は、測定器の諸機能をGPIBを介して制御するためのプログラミング言語です。SCPIは、IEEE 488.2のハードウェア部分の上にある層です。同一のSCPIコマンドとパラメータで、異なるクラスの測定器の同じ機能を制御しています。例えば、同じDISPlayコマンドを使用して、dcソースのディスプレイとSCPI互換マルチメータのディスプレイを制御します。

本書で使用されている表記法

かぎっこ <>	山括弧内の項目はパラメータを表します。例えば、<NR1>は、特定の形式の数値データを示します。
縦棒	縦棒によって複数の代替パラメータを区切ります。例えば、NORM TEXTは、"TEXT"と"NORM"のどちらかをパラメータとして使用できることを示します。
角っこ []	角っこ内の項目はオプションです。[SOURce:]VOLTageと表記されている場合、SOURce:は省略してもかまいません。
中かっこ {}	中かっこは、0回またはそれ以上繰り返すことができるパラメータを示します。特に、配列を示す場合に使用されています。<A>{<,B>}と表記されている場合、パラメータ"B"は省略したり、複数回入力することが可能であるのに対して、パラメータ"A"の入力は必須です。
ボールド・フォント	ボールド・フォントは、コマンド定義の構文を強調するために使用されています。 TRIGger:COUNT:CURRent <Nrf> は、コマンド定義であることを示します。
コンピュータ・フォント	コンピュータ・フォントは、テキストのプログラム行を示す場合に使用されています。TRIGger:COUNT:CURRent 10は、プログラム行であることを示します。

SCPIコマンドの種類

SCPIには、共通コマンドとサブシステム・コマンドの2種類のコマンドがあります。

- 共通コマンドは、一般に、特定の処理ではなく、リセット、ステータス、同期などdcソースの全体的な機能の制御に関係します。共通コマンドはすべて、次のように前にアスタリスクが付いた3文字のニモニックから成ります。 *RST *IDN? *SRE 8
- サブシステム・コマンドは、特定のdcソース機能を実行します。サブシステム・コマンドは、ルートから枝分かれする1つの逆ツリー構造を作り上げています。下の図に、サブシステム・コマンド・ツリーの一部分を示します。このツリーからさまざまなパス上にあるコマンドにアクセスします。ツリーの全体については、表4-1を参照してください。

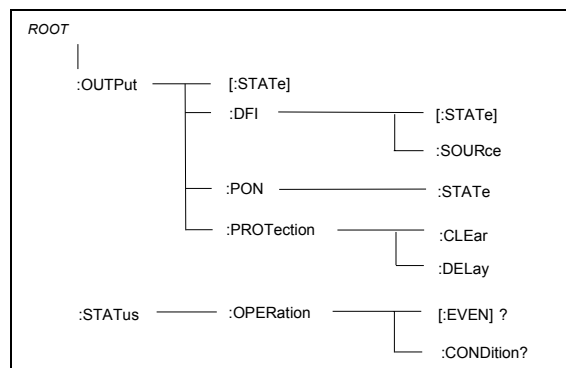


図2-1. コマンド・ツリーの一部分

複数のコマンドから成る単一のメッセージ

複数のSCPIコマンドを結合して、1つのメッセージ・ターミネータで終わる単一のメッセージとして送信することができます。複数のコマンドを単一のメッセージとして送信する場合には、次の2点に留意しなければなりません。

- セミコロンを使用してメッセージ内のコマンドを区切る。
- dcソースがコマンドを解釈する方法に影響を及ぼす暗黙のヘッダ経路が存在する。

ヘッダ経路は、メッセージ内の各コマンドの**前に**挿入される1つの文字列と考えることができます。メッセージ内の最初のコマンドに対するヘッダ経路は、ヌル文字列になります。後続の各コマンドに対するヘッダ経路は、メッセージ内にある前のコマンドのヘッダを構成する文字(最後のコロン・セパレータを含む)と定義されます。例として、以下に2つのコマンドから成る単一のメッセージを示します。

```
CURR:LEV 3;PROT:STAT OFF
```

この例は、2つのコマンドを区切るセミコロンの使用法とヘッダ経路の概念を示すものです。2番目のコマンドに対し、先行するヘッダ "CURR" が省略されていることに注意してください。これは、"CURR:LEV 3" コマンドの後のヘッダ経路は "CURR" となることが定義されているため、測定器は2番目のコマンドを次のように解釈するからです。

```
CURR:PROT:STAT OFF
```

実際、"CURR" を2番目のコマンドに明示的に含めると、2番目のコマンドとヘッダ経路を結合したときに次のような誤った結果が得られ、構文誤りとなります。

```
CURR:CURR:PROT:STAT OFF
```


サブシステム間の移動

各種サブシステムのコマンドを結合するためには、メッセージ内のヘッダ経路をヌル文字列にリセットできなければなりません。この場合、コマンドをコロン(:)で開始すれば、前のヘッダ経路はすべて除去できます。例えば、ルート指示子を次のように用いることによって、1つのメッセージ内の出力保護をクリアし、演算条件レジスタのステータスをチェックすることができます。

```
OUTPut:PROtEction:CLear;:STATus:OPERation:CONDition?
```

以下のメッセージからは、同一サブシステム内だけでなく、異なるサブシステムのコマンドの結合方法も分かります。

```
VOLTage:LEVel 20;PROtEction 28;:CURRent:LEVel 3;PROtEction:STATe ON
```

電圧サブシステムや電流サブシステム内で正しいパスを保持するためにオプション・ヘッダLEVelを使用し、サブシステム間を移動するためにルート指示子を使用しています。

共通コマンドの挿入

共通コマンドとシステム・コマンドを同一のメッセージに結合することができます。共通コマンドは、セミコロン(メッセージ・ユニット・セパレータ)で区切ることによって、メッセージ・ユニットとして扱います。共通コマンドはヘッダ経路に影響を及ぼすことはないので、メッセージ内の任意の場所に挿入することができます。

```
VOLTage:TRIGgered 17.5;:INITialize;*TRG  
OUTPut OFF;*RCL 2;OUTPut ON
```

クエリの使用法

クエリを使用して、以下の予防措置を講じてください。

- 戻りデータに対して正しい数の変数を設定します。
- 別のコマンドを dc ソースに送信する前に、クエリの結果をすべてリードバックします。さもないと、**Query Interrupted(クエリ割込み)**エラーが発生し、戻っていないデータが失われてしまいます。

SCPIメッセージの種類

SCPIメッセージには、プログラムと応答の2種類があります。

- **プログラム・メッセージ**は、コントローラからdcソースに送られる1つまたは複数の正しくフォーマットされたSCPIコマンドから成ります。このメッセージはいつでも送信することができ、dcソースに何らかの処理を実行するように要求します。
- **応答メッセージ**は、dcソースからコントローラに送られる、ある特定のSCPIフォーマットのデータから成ります。dcソースは、「クエリ」と呼ばれるプログラム・メッセージによって命令された場合にだけ、このメッセージを送信します。

下の図に、SCPIメッセージの構造を示します。

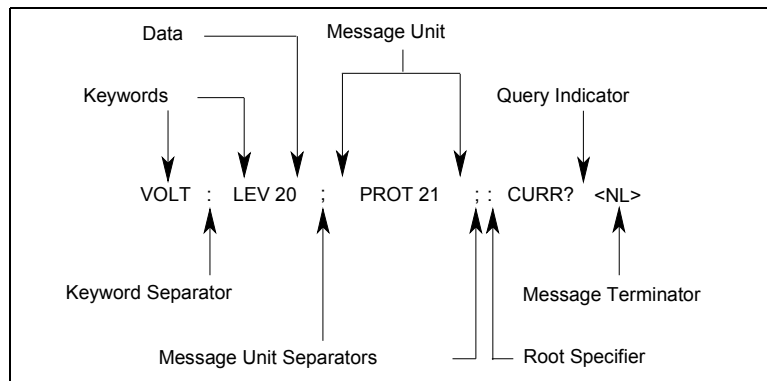


図2-2. コマンド・メッセージの構造

メッセージ・ユニット

最も単純なSCPIコマンドは単一のメッセージ・ユニットで、1つのコマンド・ヘッダ(キーワード)とメッセージ・ターミネータから成ります。このメッセージ・ユニットにはヘッダの後にパラメータを指定することができます。パラメータは、数値または1つの文字列です。

```
ABORt<NL>
VOLTage 20<NL>
```

ヘッダ

ヘッダは、dcソースによって認識される命令で、キーワードとも呼ばれます。ヘッダにはショートとロングの2つの形式があります。ロング形式では、VOLTAGE、STATUS、DELAYなど、ヘッダのスペルを省略せずに完全な形で指定します。一方、ショート形式の場合、VOLT、STAT、DELなど、ヘッダとして最初の3文字や4文字を指定します。

クエリ・インジケータ

ヘッダの後に疑問符を付ければ、ヘッダがクエリに変わります(VOLTage?, VOLTage:PROTection?)。クエリにパラメータが含まれている場合、クエリ・インジケータは最後のヘッダの終わりに挿入します(VOLTage:PROTection? MAX)。

メッセージ・ユニット・セパレータ

2つまたはそれ以上のメッセージ・ユニットが結合されて1つの複合メッセージを構成している場合には、セミコロンでユニットを区切ります(STATUS:OPERation?;QUESTIONable?)。

ルート指示子

メッセージ・ユニットの最初のヘッダの前に置かれているコロンがルート指示子になります。ルート指示子は、それがコマンド・ツリーのルートまたは一番上のノードであることをコマンド構文解析プログラムに知らせます。

メッセージ・ターミネータ

ターミネータは、メッセージの終わりに達したことをSCPIに知らせます。次の3種類のメッセージ・ターミネータが使用できます。

- 改行(<NL>)、ASCII 10進法の10または16進法の0A
- 終わりまたは識別(<END>)
- その両方(<NL><END>)

本書に示す例では、各メッセージの終わりにメッセージ・ターミネータがあると仮定しています。

注記： dcソースによって送信されるRS-232応答データはすべて、<carriage return><newline>のASCII文字ペアで終わります。GPIB応答データはこれと異なり<newline>1文字で終わり、EOIがアサートされません。

SCPIデータ・フォーマット

dcソースに設定されるデータやdcソースから返されるデータはすべてASCIIです。データは数値文字列またはキャラクタ文字列から成ります。

数値データ・フォーマット

シンボル	データ形式
	<u>トーク・フォーマット</u>
<NR1>	最下位桁の右側に、黙示の10進小数点があると仮定している数字。例: 273
<NR2>	明示の10進小数点を持つ数字。例: .0273
<NR3>	明示の10進小数点と指数を持つ数字。例: 2.73E+2
	<u>リッスン・フォーマット</u>
<NRf>	<NR1>、<NR2>、<NR3>を含む拡張フォーマット。例: 273 273. 2.73E2
<NRf+>	<NRf>とMIN MAXを含む拡張10進フォーマット。例: 273 273. 2.73E2 MAX 。 MINおよびMAXは最小値と最大値で、パラメータのレンジ仕様に黙示的に含まれます。
<ブール>	ブール・データ。例: 0 1 または ON OFF

接尾語と乗数

種類	接尾語	単位	単位と乗数
電流	A	アンペア	MA(ミリアンペア)
振幅	V	ボルト	MV(ミリボルト)
時間	S	秒	MS(ミリ秒)
		常用乗数	
	1E3	K	キロ
	1E-3	M	ミリ
	1E-6	U	マイクロ

応答データ型

クウェリ文によって返される文字列は、返される文字列の長さに応じて、次のいずれかの形をとります。

- <CRD> 文字応答データ。キャラクタ文字列を返します。
- <AARD> 任意ASCII応答データ。区切り文字のない7ビットのASCIIを返します。このデータ型には、黙示のメッセージ・ターミネータが含まれます。
- <SRD> 文字列応答データ。二重引用符で囲まれた文字列パラメータを返します。

SCPIコマンドの完了

dcソースに送られたSCPIコマンドは、順次、または並列に処理されます。シーケンシャル・コマンドは、後続のコマンドが開始される前に実行を終了します。パラレル・コマンドの場合には、コマンドの実行中に別のコマンドの実行を開始することができます。パラレル・コマンドは、トリガ動作に影響を及ぼす場合があります。

*WAI、*OPC、*OPC?といった共通コマンドは、伝送されたすべてのコマンド(パラレル・コマンドを含む)が動作を完了した時点と、さまざまな方法で知らせます。これらのコマンドの構文とパラメータについては、第4章で説明します。コマンドを使用する際には、以下の点に留意してください。

- *WAI このコマンドは、待機中の処理がすべて完了するまで、dcソースが後続のコマンドを処理しないようにします。
- *OPC? このコマンドは、待機中の処理がすべて完了すると、出力待ち行列に1を置きます。*OPC?コマンドは、戻り値を読み取ってから次のプログラム文を実行するようにプログラムに要求するので、このコマンドを使用すれば、コントローラがコマンドの完了を待って次のプログラムに進むようにできます。
- *OPC このコマンドは、待機中の処理がすべて完了するとOPCステータス・ビットをセットします。プログラムは割り込みが発生した時にこのステータス・ビットを読み取ることができるので、*OPCコマンドによって後続のコマンドを実行させることができます。

注記： ステータスOPCビットが真になるには、トリガ・サブシステムがアイドルでなければなりません。したがって、トリガに関する限り、トリガ・サブシステムが起動ステートにある場合、OPCは常に偽になります。

デバイス・クリアの使用法

デバイス・クリアを送信すれば、GPIBインタフェースをハングアップさせる恐れのあるSCPIコマンドを、いつでもアボートすることができます。デバイス・クリア・メッセージの受信によって、ステータス・レジスタ、エラー待ち行列、設定ステートが変わることはありません。デバイス・クリアは、次のような処理を実行します。

- dcソースの入出力バッファをクリアします。
- dcソースが新しいコマンド文字列を受け入れるように準備します。

以下のステートメントは、Agilent BASICを使用して、GPIBインタフェースを介してデバイス・クリアを送信する方法を示します。

```
CLEAR 705                      IEEE-488デバイス・クリア
```

以下のステートメントは、CまたはQuickBASIC用GPIBコマンド・ライブラリを使用して、GPIBインタフェースを介してデバイス・クリアを送信する方法を示します。

```
IOCLEAR (705)
```

注記： RS-232を使用する場合、Breakを送信すると、IEE-488デバイスのクリア・メッセージと同じ処理が行われます。

dcソースの設定

はじめに

本章では、dcソースの設定方法を簡単な例をあげて説明します。以下の設定例を紹介しています。

- 電圧や電流などの出力機能
- 内部トリガと外部トリガ
- 測定機能
- ステータス機能と保護機能

注記： 本章の例からは、特定の機能を実行するためにどのコマンドを使用するかは分かりますが、ある特定のプログラミング環境でどのコマンドが使用されているかを知ることはできません。特定のプログラミング環境で使用するSCPIコマンドの例については、付録Dを参照してください。

出力の設定

電源投入時の初期化

最初に電源を入れた時、dcソースは出力ステートがOFFに設定された状態で起動されます。このステートでは、出力電圧は0に設定されます。電源を入れると、以下のコマンドが黙示的に送信されます。

```
*RST
*CLS
STATus:PRESet
*SRE 0
*ESE 0
```

*RSTは、すべてのパラメータを既知のステートに設定するための便利な方法です。第4章の「*RSTコマンド」の項で、*RSTコマンドを使ってプログラム可能なパラメータを設定する方法について説明します。*ESEレジスタと*SREレジスタの電源投入時の初期化の詳細については、第4章の「*PSCコマンド」の項を参照してください。

出力のイネーブル

出力をイネーブルにするには、次のコマンドを使用します。

```
OUTPut ON
```

3 - dcソースの設定

出力電圧

出力電圧は、VOLTageコマンドによって制御します。例えば、出力電圧を25ボルトに設定するには、次のコマンドを使用します。

```
VOLTage 25
```

出力電圧が設定されたピーク電圧リミットを超えた時に、dcソースの出力がオフになるよう設定できます。この保護機能は、VOLTage:PROTECTIONコマンドによって実現されます。詳細については、第4章を参照してください。

最大電圧

設定可能な最大出力電圧は、次のコマンドを使用して問い合わせることができます。

```
VOLTage? MAX
```

出力電流

すべてのモデルに電流設定機能が備わっています。以下に出力電流を設定するためのコマンドを示します。

```
CURRENT <n>
```

ここで、<n>は電流リミット(単位:アンペア)です。

負荷が設定リミットを超える電流を流し込もうとすると、出力電圧が低下して電流がリミット内に保たれます。

最大電流

設定可能な最大出力電流は、次のコマンドを使用して問い合わせることができます。

```
CURRENT? MAX
```

過電流保護

電流リミットに達した時にdcソースの出力がオフになるよう設定できます。この保護機能は、次のコマンドによって実現されます。詳細については、第4章を参照してください。

```
CURRENT:PROTECTION:STATE ON | OFF
```

注記： OUP:PROT:DELを使用すれば、出力設定の変更によって一時的に電流リミット条件が変わっても、過電流保護機能が作動するのを防ぐことができます。

出力変動のトリガ

dcソースには2つの独立したトリガ・システムがあります。1つは出力変動の生成用であり、もう1つは測定のトリガ用です。本項では、出力トリガ・システムについて取り上げ、測定トリガ・システムについては「測定のトリガ」の項で説明します。

SCPIにおけるトリガの名称

SCPIでは、トリガ・システムをシーケンスと呼んでいます。トリガ・システムが複数存在する場合、SEquence1とSEquence2という名前で区別します。SEquence1は過渡トリガ・システムを、SEquence2は測定トリガ・システムをそれぞれ表します。dcソースは、これらのシーケンスに対して、もっと内容を示す名前を別名として使用します。これらの別名は、シーケンス・フォームの代わりに使用することができます。

シーケンス・フォーム	別名
SEquence1	TRANsient
SEquence2	ACQuire

出力トリガ・システム・モデル

図3-1は、出力トリガ・システムの1つのモデルを示したものです。長方形のボックスは状態を表します。矢印は状態間の遷移を示します。矢印には、遷移を発生させた入力またはイベントが示されています。

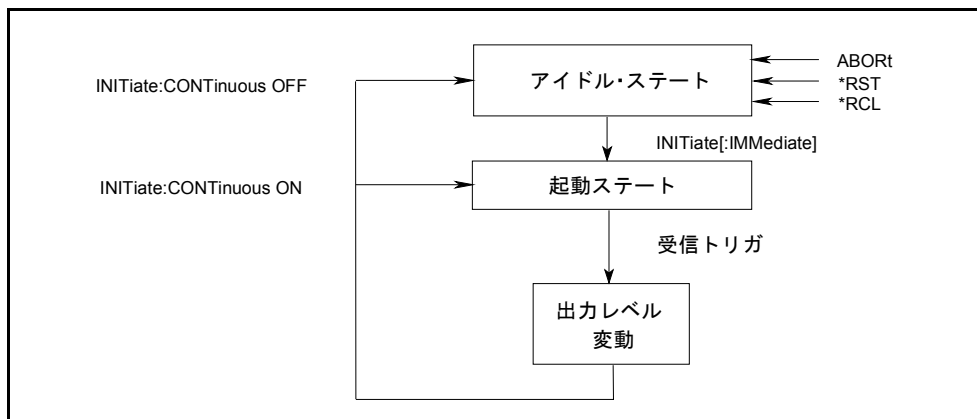


図3-1. 出力トリガのモデル

電圧または電流トリガ・レベルの設定

トリガの出力レベルを設定するには、まず、トリガ信号を受信した時に出力が達する電圧トリガ・レベルまたは電流トリガ・レベルを指定する必要があります。以下のコマンドを使用して、出力トリガ・レベルを設定します。

```
VOLTage:TRIGgered <n> または
CURRent:TRIGgered <n>
```

注記：

トリガ・レベルが初期設定されていない場合、設定を行うまでは対応する即時レベルがトリガ・レベルと見なされます。例えば、dcソースの電源がオンで、VOLTage:LEVelが6に設定されている場合、別の値に設定するまではVOLTage:LEVel:TRIGgerも6になります。VOLTage:LEVel:TRIGgerをある値に設定すると、後でVOLTage:LEVelの設定を変えても、VOLTage:LEVel:TRIGgerの値は変わりません。

3 - dcソースの設定

出力トリガ・システムの起動

dcソースをオンにしたとき、トリガ・サブシステムはアイドル・ステートになっています。アイドル・ステートでは、トリガ・サブシステムは、すべてのトリガを無視します。以下のコマンドを送信すると、トリガ・システムはいつでもアイドル・ステートに戻ります。

```
ABORt
*RST
*RCL
```

INITiateコマンドは、トリガ・システムをアイドル・ステートから起動ステートに変えます。これによって、dcソースがトリガを受信できるようになります。トリガ・システムを起動して1つのトリガ動作を行わせるには、次のコマンドを使用します。

```
INITiate:SEquence1 または
INITiate:NAME TRANSient
```

トリガが受信され、動作が完了すると、トリガ・システムはアイドル・ステートに戻ります。このため、トリガ動作が必要になるたびに、システムを起動させなければなりません。

次のコマンドを使用すれば、トリガごとに起動コマンドを送信しなくてもトリガ・システムが起動ステートに保たれ、複数のトリガ動作が実行できます。

```
INITiate:CONTinuous:SEquence1 ON または
INITiate:CONTinuous:NAME TRANSient, ON
```

トリガの生成

出力トリガは、GPIBバスを介してのみプログラムできます。BUSはトリガ出力の唯一のトリガ源ですが、念のために以下のコマンドを記載しておきます。

```
TRIGger:SOURce BUS
```

適切なトリガ源を指定すれば、次の手順でトリガを発生させることができます。

シングル・トリガ 以下のコマンドのうちの1つをGPIBを介して送信します。

```
TRIGger:IMMediate
*TRG
グループ実行トリガ
```

連続トリガ 以下のコマンドをGPIBを介して送信します。

```
INITiate:CONTinuous:SEquence1 ON
```

トリガ・システムがトリガを受信して出力変動ステートに入ると(図3-1を参照)、トリガされた機能は、プログラムで指定したトリガ・レベルに設定されます。トリガ動作が完了すると、トリガ・システムはアイドル・ステートに戻ります。

測定の実施

dcソースには、数種類の電圧および電流測定機能が備わっています。Agilent 66312AとAgilent 66332Aモデルの測定機能は、電流をパルスで流し込む負荷に特に有用です。

注記： 出力電圧と出力電流は、同時に測定できません。

測定では、定義したサンプル数やサンプリング間隔における瞬時出力電圧をデジタル化し、結果をバッファにストアした後、測定結果を算出します。多くの測定パラメータが設定可能です。設定可能なパラメータとして、サンプル数、サンプリング間隔、帯域幅、トリガ方法があります。雑音がある場合、これらの値を大きくすると、測定の速度、確度、安定性が低下します。

測定には、次の2通りの方法があります。

- MEASureコマンドを使用して、即座に新しい電圧データまたは電流データの収集を開始し、バッファが一杯になるとすぐに、このデータから算出した測定結果を返します。この方法は、明示的なトリガの設定が不要であるため、最も簡単な測定方法であると言えます。
- 捕捉トリガを使用してデータを収集します。次に、FETChコマンドを使用して、捕捉トリガによって取り出されたデータから算出した測定結果を返します。この方法では、データの捕捉と出力電圧や出力電流の遷移を同期させることができます。FETChコマンドによって新しい測定データの捕捉をトリガすることはできませんが、捕捉トリガによって取り出されたデータを基に、多種多様な計算結果を返すことができます。

捕捉トリガ・システムを使用したトリガ測定の実行方法については、「測定のトリガ」を参照してください。

注記： MEASure形式のクエリにはそれぞれ、ヘッダFETChで始まる対応するクエリがあります。FETChクエリでは、対応するMEASureクエリと同じ計算が行われますが、新しいデータの捕捉は行われません。明示的なトリガまたは前にプログラムされたMEASureコマンドによって収集されたデータが使用されます。

電圧および電流の測定

SCPI言語には、電圧波形や電流波形の各種測定パラメータを返すMEASureクエリやFETChクエリが多数あります。

DC測定

DC出力電圧またはDC出力電流を測定するには、次のコマンドを使用します。

```
MEASure:VOLTage? または
MEASure:CURRent?
```

DC電圧やDC電流を測定するには、選択した時間間隔で多くの値を読み取り、読取り値にハニング窓関数を適用して、読取り値を平均化します。ウィンドイングは、電源リップルなどの周期信号がある場合にDC測定の誤差を減らす、信号調節処理です。電源を入れ、*RSTコマンドを送信すると、次のパラメータが設定されます。

```
SENSe:SWEep:TINterval 15.6E-6
SENSe:SWEep:POINTs 2048
```

3 - dcソースの設定

これによって、データの捕捉時間は32ミリ秒になります。約20ミリ秒のコマンド処理オーバーヘッドを加えると、合計の測定時間は1測定サンプル当たり約50ミリ秒になります。

リップル除去は、捕捉窓に含まれるリップル周波数のサイクル数の関数です。捕捉窓内のサイクルの数が多ければ、リップルを減らすことができます。例えば、各測定の時間間隔を45マイクロ秒に延長した場合、約70dBのリップル除去に必要な捕捉窓内のサイクル数は、60Hzで5.53サイクルです。

測定の速度は、サンプル点数の数を減らすことによって速めることができます。例えば、次のコマンドでは、捕捉時間は16ミリ秒に短縮されますが、測定精度は低下します。

```
SENSe:SWEep:TINterval 15E-6  
SENSe:SWEep:POINts 1024
```

実効値測定(Agilent 66312A, 66332Aの場合のみ)

電圧波形や電流波形の実効値の内容を読み取るには、次のコマンドを使用します。

```
MEASure:VOLTage:ACDC? または  
MEASure:CURREnt:ACDC?
```

このコマンドは、DC部分を含めた総実効測定値を返します。

捕捉したAC波形のデータ・サイクル数が整数でないと、AC波形で実効値測定を行った場合、捕捉データの最後のサイクルが不完全であるため測定誤差が生じます。測定器は、測定を行う際にハニング窓関数を用いることによってこの誤差を減少させます。

最小値および最大値の測定(Agilent 66312A, 66332Aの場合のみ)

パルスやAC波形の最大電圧、最小電圧、最大電流、最小電流を測定するには、次のコマンドを使用します。

```
MEASure:VOLTage:MAXimum?  
MEASure:VOLTage:MINimum?  
MEASure:CURREnt:MAXimum?  
MEASure:CURREnt:MINimum?
```

電流レンジ

dcソースには2つの電流測定レンジがあります。電流レンジを制御するには、次のコマンドを使用します。

```
SENSe:CURREnt:RANGe MIN | MAX
```

レンジをMINに設定した場合に測定可能な最大電流は、20ミリアンペアです。

データ・バッファからの測定データの戻り(Agilent 66312A, 66332Aの場合のみ)

MEASureクエリとFETChクエリは、瞬時電圧または電流バッファ内のデータ値をすべて返すこともできます。この場合、次のコマンドを使用します。

```
MEASure:ARRay:CURREnt?  
MEASure:ARRay:VOLTage?
```

内部トリガ測定

データ捕捉トリガ・システムを用いることによって、電圧または電流データ捕捉のタイミングをBUSまたは内部トリガ源と同期させることができます。次に、FETChコマンドを使用して、測定トリガによって捕捉されたデータを基にさまざまな計算結果を返します。

SCPIにおけるトリガの名称

前述の「出力変動のトリガ」の項でも説明したように、dcソースは、測定トリガ・システムに対し以下のシーケンス名と別名を使用します。別名は、シーケンス・フォームの代わりに使用することができます。

シーケンス・フォーム	別名
SEquence2	ACquire

測定トリガ・システム・モデル

図3-2に、測定トリガ・システムの1つのモデルを示します。長方形のボックスは状態を表します。矢印はステート間の遷移を示します。矢印には、遷移を発生させた入力またはイベントが示されています。

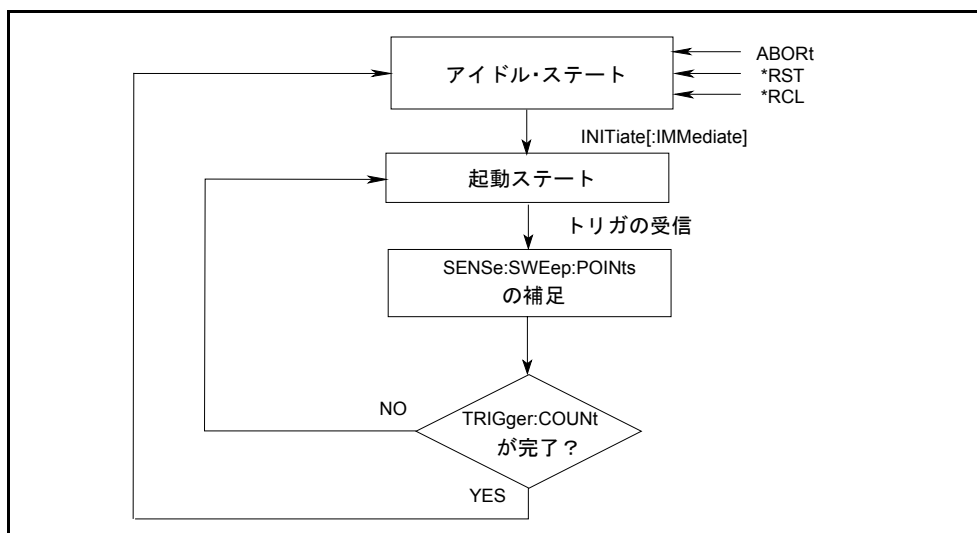


図3-2. 測定トリガのモデル

測定トリガ・システムの起動(Agilent 66312A, 66332Aの場合のみ)

dcソースの電源を入れた時には、トリガ・システムはアイドル・ステートにあります。このステートでは、トリガ・システムはトリガをすべて無視します。以下のコマンドを送信すると、トリガ・システムはいつでもアイドル・ステートに戻ります。

```

ABORT
*RST
*RCL

```

INITiateコマンドは、トリガ・システムをアイドル・ステートから起動ステートに変えます。これによって、dcソースがトリガを受信できるようになります。測定トリガのためにトリガ・システムを起動させるには、次のコマンドを使用します。

3 - dcソースの設定

```
INITiate:SEquence2 または  
INITiate:NAME ACquire
```

トリガを受信してデータの捕捉を完了すると、トリガ・システムは、複数の測定が必要とされない限りアイドル・ステートに戻ります。このため、トリガ捕捉が必要になるたびに、システムを起動させる必要があります。

注記： 測定トリガを連続的に発生させることはできません。連続して発生させると、トリガ測定のたびに、データ・バッファ内の測定データが上書きされてしまいます。

測定トリガ源の選択(Agilent 66312A, 66332Aの場合のみ)

トリガ・システムは、起動ステートではトリガ信号を待っています。トリガを生成するためには、まず、トリガ源を選択しなければなりません。以下の測定トリガ源が選択可能です。

BUS - GPIBバス・トリガを選択します。
INternal - dcソースの出力を測定トリガとして選択します。

GPIBバス・トリガ(グループ実行トリガ、デバイス・トリガまたは*TRGコマンド)を選択するには、次のコマンドを使用します。

```
TRIGger:SEquence2:SOURce BUS または  
TRIGger:ACquire:SOURce BUS
```

内部トリガ(出力信号でトリガを発生させる測定)を選択するには、次のコマンドを使用します。

```
TRIGger:SEquence2:SOURce INternal または  
TRIGger:ACquire:SOURce INternal
```

測定トリガの生成(Agilent 66312A, 66332Aの場合のみ)

dcソースには、測定変換器が1つだけあります。測定トリガを発生させるには、まず、電圧測定捕捉または電流測定捕捉を指定しなければなりません。測定捕捉を指定するには、次のコマンドを使用します。

```
SENSe:FUNction "CURRENT" または  
SENSe:FUNction "VOLTage"
```

適切なトリガ源と測定捕捉を指定すれば、次の手順でトリガを発生させることができます。

GPIBトリガ 以下のコマンドのうち1つをGPIBを介して送信します。
TRIGger:IMMediate(トリガ源の設定に影響されません)
*TRG
グループ実行トリガ

内部トリガ 出力信号のトリガに対し、トリガ条件を限定するためにトリガを発生させる出力レベル、スロープの立ち上がり端または立下がり端、ヒステリシスを指定する必要があります。図3-3を参照してください。

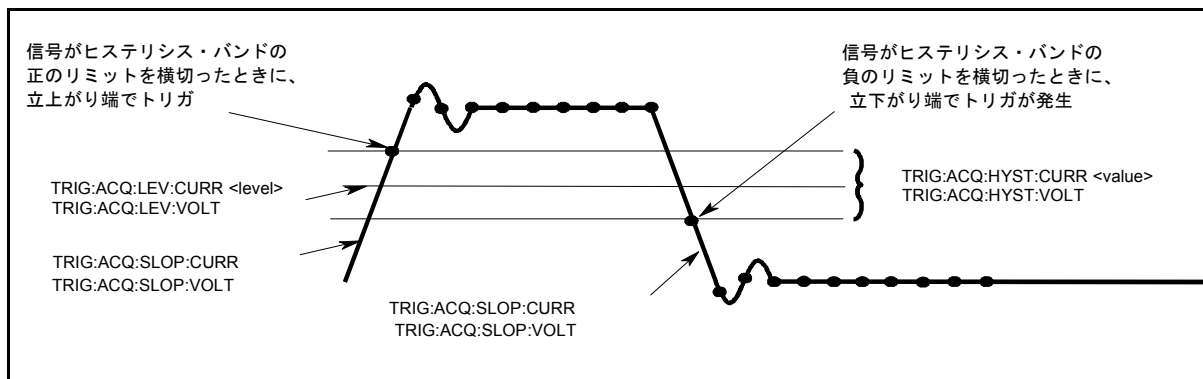


図3-3. 出力パルスを測定するためのトリガ・コマンド

立上がり信号と立下がり信号の両方に対してトリガを発生させる出力レベルを指定するには、次のコマンドを使用します。

```
TRIGger:SEquence2:LEVel:CURRent <値> または
TRIGger:ACquire:LEVel:CURRent <値>
```

トリガをかけるスロープを指定するには、次のコマンドを使用します。指定可能なスロープの種類として、POSitive、NEGative、EITHerがあります。

```
TRIGger:SEquence2:SLOPe:CURRent <スロープ> または
TRIGger:ACquire:SLOPe:CURRent <スロープ>
```

立上がり信号または立下がり信号を限定するためのヒステリシス・バンドを指定するには、次のコマンドを使用します。

```
TRIGger:SEquence2:HYSTeresis:CURRent <値> または
TRIGger:ACquire:HYSTeresis:CURRent <値>
```

注記： 内部トリガを使用する場合には、必ず、スロープ、レベルおよびヒステリシスを指定してから測定を開始(INITiateコマンドを送信)してください。

捕捉が終了したら、任意のFETChクウェリを使用して捕捉結果を返すことができます。測定トリガが一旦開始されると、データ捕捉のトリガまたは完了前にFETChクウェリが送信された場合、トリガが発生し捕捉が完了するまで応答データが遅延します。このため、トリガ条件がすぐに整わないと、コントローラが拘束されてしまう場合があります。

コントローラを拘束することなく結果を待つ1つの方法として、SCPIコマンドの完了コマンドを使用します。例えば、INITializeの後に*OPCコマンドを送信し、他のタスクの実行中に標準イベント・ステータス・レジスタのOPCステータス・ビットを時々ポーリングして、ステータスの完了を確かめます。また、OPCステータス・ビットが真になるときのSRQ条件を設定し、SRQ割込みが発生するまで他のタスクを実行することも可能です。

出力パルスの測定(Agilent 66312A, 66332Aの場合のみ)

電流ディテクタ

数kHzより大きい周波数成分を持つ電流パルスや他の波形を測定する時は、電流ディテクタがACDCに設定されていることをチェックします。

```
SENSe:CURRent:DETECT ACDC
```

DC電流測定だけを行い、ハイ電流測定レンジで2mA以上の測定オフセットを必要とする場合にのみ、測定ディテクタとしてDCを選択します。DCを選択した場合、AC成分を持つ電流波形に関する結果は不正確になります。

```
SENSe:CURRent:DETECT DC
```

パルス測定クウェリ

図3-4に示すように、dcソースにはパルス波形の主要パラメータを返す測定クウェリがいくつかあります。

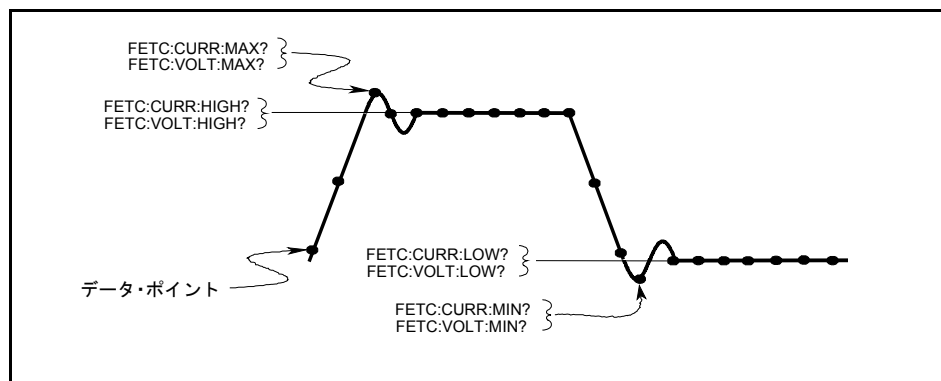


図3-4. パルス・データを返すための測定コマンド

パルス波形の最大値または最小値を返すには、次のコマンドを使用します。

```
FETCh:VOLTage:MAXimum? または
FETCh:VOLTage:MINimum?
FETCh:CURRent:MAXimum? または
FETCh:CURRent:MINimum?
```

ハイ・レベルまたはロー・レベルのパルスの平均値を測定することもできます。ハイ・レベルのパルスの平均値を返すには、次のコマンドを使用します。

```
FETCh:CURRent:HIGH? または
FETCh:VOLTage:HIGH?
```

ロー・レベルのパルスの平均値を返すには、次のコマンドを使用します。

```
FETCh:CURRent:LOW? または
FETCh:VOLTage:LOW?
```


測定サンプルの制御

電圧または電流サンプリング・レートの変更

測定サンプルのデータ・ポイント数だけでなくサンプリング間隔も変更することができます。図3-5を参照してください。

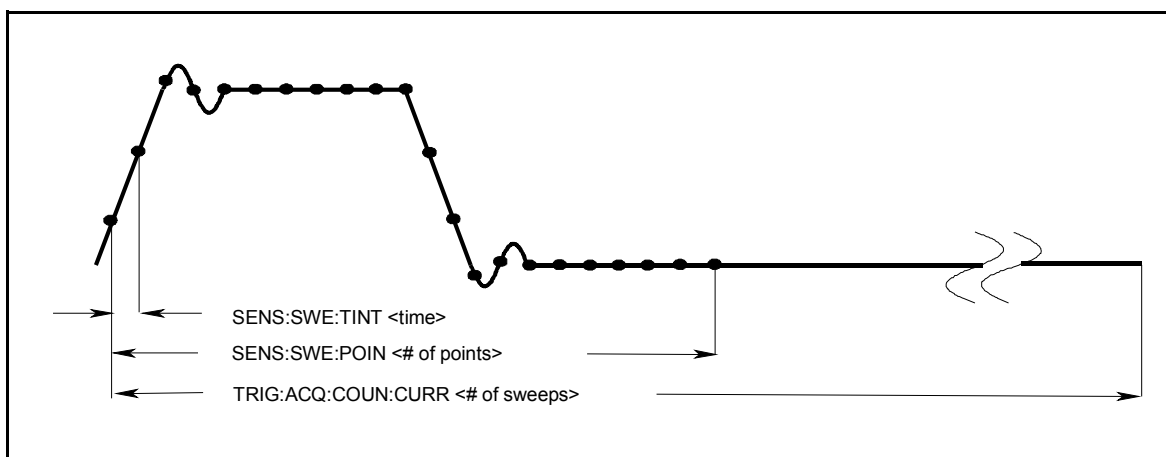


図3-5. サンプリング・レートを変更するためのセンス・コマンド

*RSTの場合、出力電圧または出力電流のサンプリング・レートは15.6マイクロ秒です。したがって、データ・バッファに2048のデータ・ポイントをストアするには約32ミリ秒かかります。このデータ・サンプリング・レートは、次のコマンドによって変更することができます。

`SENSE:SWEep:TINTerval <サンプリング周期>`

`SENSE:SWEep:POINTs <ポイント>`

例えば、サンプル数が1500で、時間間隔を46.8マイクロ秒/サンプルに設定するには、`SENSE:SWEep:TINTerval 46.8E-6;POINTs 1500`を使用します。

複数測定(Agilent 66312A, 66332Aの場合のみ)

測定器には、いくつかの捕捉トリガを連続して設定し、返された測定値の各捕捉結果を平均する機能もあります。トリガ・システムを複数のシーケンシャル捕捉を行うように設定するには、次のコマンドを使用します。

`TRIGger:ACQuire:COUNT:CURRENT <数値> または`

`TRIGger:ACQuire:COUNT:VOLTage <数値>`

この設定では、測定器は各捕捉を順次実行し、デジタル化した読取り値を内部測定バッファにストアします。測定の初期化は、初めに1回だけ行います。測定器は、捕捉が完了すると、次の有効なトリガ条件を待って別の捕捉を開始します。MEASureまたはFETChによって返される結果は、収集された全データの平均値となります。

注記： データ・ポイントの総数は、4096を超えてはなりません。つまり、トリガ・カウントと掃引点の積が4096を超えてはなりません。4096を超えると、エラーが発生します。

プリイベントおよびポストイベント・トリガ(Agilent 66312A, 66332Aの場合のみ)

測定が開始されると、dcソースは、瞬時出力電圧または瞬時出力電流を連続的にサンプリングします。図3-6に示すように、ユーザは、捕捉バッファに読み込まれているデータ・ブロックを捕捉トリガを基準に移動させることができます。これによって、プリイベント・データやポストイベント・データのサンプリングが可能になります。

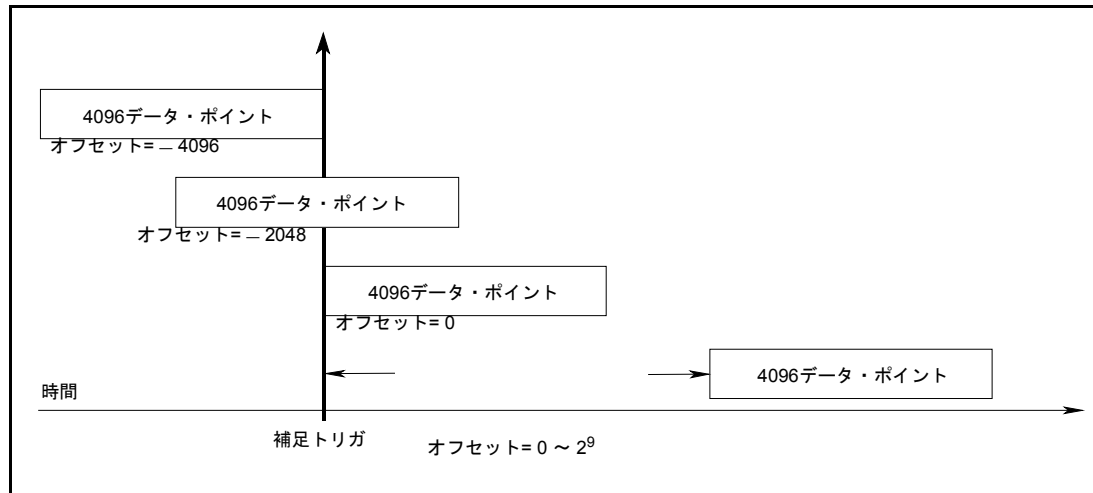


図3-6. プリイベントおよびポストイベント・トリガ

捕捉バッファの開始位置を捕捉トリガに対してオフセットするには、次のコマンドを使用します。

```
SENSE:SWEp:OFFSet:POINts <オフセット>
```

オフセットの範囲は、-4096~2,000,000,000ポイントです。図に示すように、オフセット値が負の時、データ・レコードの始めの値は、トリガの前に取り込まれたサンプルを表します。オフセット値が0の時、値はすべてトリガ後に取り込まれています。トリガを受信してからバッファに入れられるデータ・ポイントが有効になるまでの遅延時間を設定するには、0より大きい値を使用します(遅延時間 = オフセット X サンプリング周期)。

パルス測定例(Agilent 66312A, 66332Aの場合のみ)

以下のプログラムは、GPIBを介したパルスの測定方法を示したものです。測定機能がACDCに設定されているので、AC成分が含まれた電流波形の最高の測定結果が得られます。測定結果には、2ミリ秒の総測定時間内に20マイクロ秒の時間間隔で読み取られた100の値が含まれています。パルス測定のトリガ点は、電流パルスの立上がりスロープ上に0.1アンペアごとに発生しています。測定オフセットは、トリガ前の20測定ポイントも測定サンプルの一部として返されるようにプログラムされています。

測定トリガは電流パルスによって開始されるため、測定データを返すのにFETChコマンドが使用されています。FETChコマンドは、測定のMAXimum、MINimum、HIGH、およびLOW値を返すのにも使用されています。

注記： MEASureコマンドは、使用される度に必ず新しい測定データを捕捉するので、本例の場合はデータを返すのに使用することはできません。

このプログラムは、Agilent BASICで動作するすべてのコントローラ上で実行することができます。出力パルスを生成するためには、1000Hzで100マイクロ秒のデューティ・サイクルで3アンペア・パルスを生成するように電子負荷をプログラムします。電源アドレスは705、負荷アドレスは706です。必要に応じて、該当するステートメント内のこれらのパラメータを変更してください。

```

10      !Rev A.00.00
20      OPTION BASE 1
30      DIM Curr_array(100)
40      !
50      ASSIGN @Ps TO 705
60      ASSIGN @Ld TO 706
80      OUTPUT @Ps; "*RST"
90      OUTPUT @Ps; "OUTP ON"
100     OUTPUT @Ps; "VOLT 5"
110     !
120     OUTPUT @Ld; "CURR:LEVEL 0"
130     OUTPUT @Ld; "CURR:TLEVEL 3"
140     !
150     OUTPUT @Ld; "TRAN:FREQ 1000"
160     OUTPUT @Ld; "TRAN:DCYCLE 10"
170     OUTPUT @Ld; "TRAN:MODE CONT"
180     OUTPUT @Ld; "TRAN:STATE ON"
190     !
200     OUTPUT @Ps; "SENS:CURR:DET ACDC"
210     OUTPUT @Ps; "SENS:CURR:RANG MAX"
220     OUTPUT @Ps; "TRIG:ACQ:SOUR INT"
230     OUTPUT @Ps; "SENS:FUNC " "CURR" ""
240     OUTPUT @Ps; "TRIG:ACQ:LEV:CURR .1"
250     OUTPUT @Ps; "TRIG:ACQ:SLOPE:CURR POS"
260     OUTPUT @Ps; "TRIG:ACQ:HYST:CURR .05"
270     OUTPUT @Ps; "SENS:SWE:TINT 20E-6"
280     OUTPUT @Ps; "SENS:SWE:POIN 100"
290     OUTPUT 705; "SENS:SWE:OFFS:POIN -20"
300     OUTPUT @Ps; "INIT:NAME ACQ"
310     !
320     OUTPUT @Ps; "FETCH:ARRAY:CURR?"
330     !
340     ENTER @Ps; Curr_array(*)
350     PRINT Curr_array(*)
360     !
370     OUTPUT @Ps; "FETCH:CURR:MAX?"
380     ENTER @Ps; Curr_max
390     PRINT "MAX CURRENT", Curr_max
400     !
410     OUTPUT @Ps; "FETCH:CURR:MIN?"
420     ENTER @Ps; Curr_min
430     PRINT "MIN CURRENT", Curr_mi
440     !
450     OUTPUT @Ps; "FETCH:CURR:HIGH?"
460     ENTER @Ps; Curr_hi
470     PRINT "HIGH CURRENT", Curr_hi
480     !
490     OUTPUT @Ps; "FETCH:CURR:LOW?"
500     ENTER @Ps; Curr_low
510     PRINT "LOW CURRENT", Curr_low
520     !
530     END

```

! 電源をデフォルト値に設定します

! 電源出力をオンにします

! 電源を5ボルトにプログラムします

! パルスを生成するように電子負荷を設定します

! メータをACDCに設定します

! 最大電流レンジ

! パルス上でトリガするように設定します

! 電流データを捕捉します

! 0.1アンペアでトリガします

! 立上がりスロープ上でトリガします

! トリガのヒステリシスを設定します

! サンプリング時間間隔を20μsに設定します

! 掃引の測定サンプル数を設定します

! トリガ前のサンプリング点の数

! トリガ・システムを起動します

! コントローラはトリガが発生するのを待っています

! 測定が完了したらデータを取り出します

! 100のデータ・ポイントをすべて入力します

! すべてのデータ・ポイントをプリントします

! 前の測定からさらにデータを取り出します

3 - dcソースの設定

このプログラムを実行すると、MIN、MAX、HIGH、およびLOWデータに加えて、100の測定データ・ポイントが以下のフォーマットで返されます。

```
.030585      .031869      .0344369    .031655      .0320829    .0325109    .0333669    .0340089
.0320825    .031449      .031227     .031441      .0337949    .0327249    .031869      .031655
.0327249    .031013      .0325109    .0333669    3.09751     3.1814      3.14266     3.13667
3.13817     3.13624     .977283     .0667496     .0245932    .0280171    .031013      .031655
.0331529    .0350788     .0348648    .0327249    .031227     .0327249    .031227     .030799
.031869     .0329389     .030371     .031655      .031869     .0329389    .031869     .0322869
.0320829    .0325109     .0333669    .0340089     .0348648    .0327249    .031227     .0327249
.0320829    .030371      .031449     .031227     .031441     .0337949    .031449     .0333669
.031441     .0337949     .030371     .031655      .031869     .0329389    .031869     .0293011
.031441     .0337949     .0327249    .031869      .031655     .031655     .0320829    .031227
.0322969    .031655      .0327249    .0340089     2.97661     3.18632     3.14523     3.13496
3.13453     3.13731     1.32438     .0836549     .0258772    .0284451    .0275891    .0329389
.0329389    .0333669     .0322969    .0333669
MAX CURRENT  3.18632
MIN CURRENT  .0245932
HIGH CURRENT 3.1371
LOW CURRENT  .0314077
```

ステータス・レジスタの設定

ステータス・レジスタの設定によって、dcソースの動作条件をいつでも定義することができます。例えば、電流リミットなどのイベントが発生した時に割込みを発生させる(SRQをアサートする)ように、dcソースを設定することもできます。割込みが発生すると、プログラムはそのイベントに適切な方法で対処します。

図3-7に、dcソースのステータス・レジスタの構造を示します。表3-1には、ステータス・ビットの定義が示されています。標準イベント・レジスタ、ステータス・バイト・レジスタ、サービス・リクエスト・イネーブル・レジスタおよび出力待ち行列は、IEEE 488.2 Standard Digital Interface for Programmable Instrumentationの定義に従って、標準のGPIB機能を実行します。動作ステータス・レジスタとクエスチョナブル・ステータス・レジスタは、dcソースに固有の機能を実行します。

パワーオン条件

ステータス・レジスタのパワーオン条件については、第4章の*RSTコマンドに関する説明を参照してください。

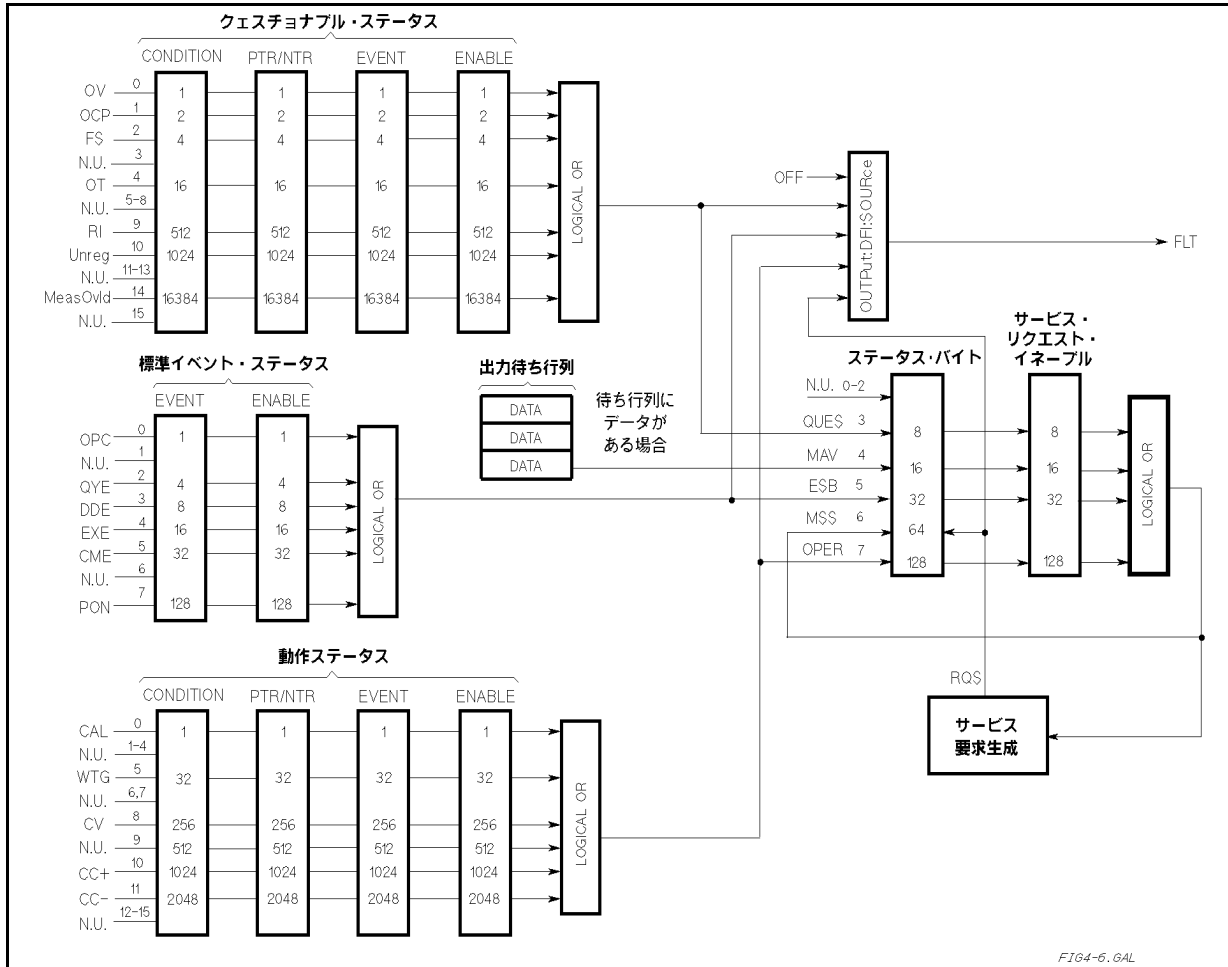


図3-7. dcソースのステータス・モデル

表3-1. ステータス・レジスタのビット構成

ビット	信号	意味
0	CAL	動作ステータス・グループ dcソースは新しい校正定数を算出中です。 dcソースはトリガ待ち状態にあります。 dcソースは定電圧モードにあります。 dcソースは定電流モードにあります。 dcソースは負の定電流モードにあります。
5	WTG	
8	CV	
10	CC+	
11	CC-	
0	OV	クwestionナブル・ステータス・グループ 過電圧保護が作動しました。 過電流保護が作動しました。 ヒューズが飛んでいます。 過熱保護が作動しました。 リモート・インヒビット・ステートにあります。 出力がレギュレーションされていません。 電流測定がロー・レンジの能力を超えています。
1	OCP	
2	FS	
4	OT	
9	RI	
10	Unreg	
14	MeasOvld	
0	OPC	標準イベント・ステータス・グループ 動作完了 クウェリ・エラー デバイス固有エラー 実行エラー コマンド・エラー パワーオン
2	QYE	
3	DDE	
4	EXE	
5	CME	
7	PON	
3	QUES	
4	MAV	
5	ESB	
6	MSS	
	RQS	
7	OPER	

動作ステータス・グループ

動作ステータス・レジスタは、正常な動作中に発生する信号を記録します。以下に示すように、このグループは、条件、PTR/NTR、イベントおよびイネーブル・レジスタから構成されます。動作ステータス・レジスタ・グループの出力を論理和(OR)演算した結果が、ステータス・バイト・レジスタのOPER(ation)サマリ・ビット(7)になります。

レジスタ	コマンド	内容
条件	STAT:OPER:COND?	モニタしている回路のリアルタイム・ステータスを保持するレジスタ。これは、リード・オンリ・レジスタです。
PTRフィルタ	STAT:OPER:PTR <n>	第4章のSTAT:OPER:NTR PTRコマンドの項で説明したような働きをする正方向遷移フィルタ。これは、リード/ライト・レジスタです。
NTRフィルタ	STAT:OPER:NTR <n>	第4章のSTAT:OPER:NTR PTRコマンドの項で説明したような働きをする負方向遷移フィルタ。これは、リード/ライト・レジスタです。
イベント	STAT:OPER:EVEN?	PTRまたはNTRフィルタを通過するすべての条件をラッチするレジスタ。これは、リード・オンリ・レジスタで、読取りが行われるとクリアされます。
イネーブル	STAT:OPER:ENAB <n>	イベント・レジスタの特定のビットをイネーブルにするためのマスクとして働くレジスタ。これは、リード/ライト・レジスタです。

クwestionナブル・ステータス・グループ

クwestionナブル・ステータス・レジスタは、dcソースの異常動作を示す信号を記録します。図3-7に示すように、このグループは、動作ステータス・グループと同じ種類のレジスタから構成されています。クwestionナブル・ステータス・グループの出力を論理和(OR)演算した結果が、ステータス・バイト・レジスタのQUESTionableサマリ・ビット(3)になります。

レジスタ	コマンド	内容
条件	STAT:QUES:COND?	モニタしている回路のリアルタイム・ステータスを保持するレジスタ。これは、リード・オンリ・レジスタです。
PTRフィルタ	STAT:QUES:PTR <n>	第4章のSTAT:QUES:NTR PTRコマンドの項で説明したような働きをする正方向遷移フィルタ。これは、リード/ライト・レジスタです。
NTRフィルタ	STAT:QUES:NTR <n>	第4章のSTAT:QUES:NTR PTRコマンドの項で説明したような働きをする負方向遷移フィルタ。これは、リード/ライト・レジスタです。
イベント	STAT:QUES:EVEN?	PTRまたはNTRフィルタを通過するすべての条件をラッチするレジスタ。これは、リード・オンリ・レジスタで、読取りが行われるとクリアされます。
イネーブル	STAT:QUES:ENAB <n>	イベント・レジスタの特定のビットをイネーブルにするためのマスクとして働くレジスタ。これは、リード/ライト・レジスタです。

標準イベント・ステータス・グループ

このグループは、共通コマンドによって設定されるイベント・レジスタとイネーブル・レジスタから構成されます。標準イベント・イベント・レジスタは、測定器の通信ステータスに関連するイベントをラッチします(図3-7を参照)。これは、リード・オンリ・レジスタで、読取りが行われるとクリアされます。標準イベント・イネーブル・レジスタは、動作ステータス・グループやクwestionナブル・ステータス・グループのイネーブル・レジスタと同じように機能します。

コマンド	動作
*ESE	標準イベント・イネーブル・レジスタ内の特定のビットを設定します。
*PSC ON	電源投入時に、標準イベント・イネーブル・レジスタをクリアします。
*ESR?	標準イベント・イベント・レジスタを読み取り、クリアします。

PON(パワー・オン)ビット

標準イベント・イベント・レジスタのPONビットは、dcソースの電源がオンになるたびにセットされます。PONの最も一般的な使用法は、予期せぬ電力損失がある場合、電源投入時にSRQを生成することにあります。そのためには、標準イベント・イネーブル・レジスタのビット7を、パワーオン・イベントがESB(標準イベント・サマリ・ビット)に記憶されるように設定し、サービス・リクエスト・イネーブル・レジスタのビット5をSRQが生成されるように設定し、*PSC OFFを送信しなければなりません。これらの条件を満たすコマンドを以下に示します。

```
*PSC OFF *ESE 128 *SRE 32
```

3 - dcソースの設定

ステータス・バイト・レジスタ

このレジスタは、IEEE 488.2 Standard Digital Interface for Programmable Instrumentationの定義に従って、他のあらゆるステータス・グループからの情報をまとめます。ビット構成については、表3-1を参照してください。

コマンド	動作
*STB?	レジスタ内のデータを読み取りますが、クリアはしません(ビット6にMSSを返します)。
シリアル・ポーリング	レジスタ内のRQSをクリアし、応答のビット位置6に返します。

MSSビット

これは、サービス・リクエスト・イネーブル・レジスタによってイネーブルにされるステータス・バイト・レジスタの全ビットのリアルタイムの(ラッチされていない)サマリです。MSSは、dcソースにサービスを要求する理由が1つ以上ある場合には必ずセットされます。*STB?は、応答のビット位置6のMSSを読み取りますが、ステータス・バイト・レジスタ内のビットをクリアすることはありません。

RQSビット

RQSビットは、MSSビットがラッチされたものです。dcソースは、サービスを要求するたびに、SRQ割込みラインを真にセットし、RQSをステータス・バイト・レジスタのビット6にラッチします。コントローラがシリアル・ポーリングを行うと、レジスタ内のRQSがクリアされ、応答のビット位置6に返されます。ステータス・バイト・レジスタの残りのビットには何の影響もありません。

MAVビットと出力待ち行列

出力待ち行列は、dcソースからコントローラへ送ったメッセージをコントローラが読み取るまでストアしておく、先入れ先出し(FIFO)データ・レジスタです。待ち行列に1バイトまたはそれ以上格納されている場合には、待ち行列はステータス・バイト・レジスタのMAVビット(4)をセットします。

サービス割込みの原因の究明

以下の動作を実行することによって、SRQの原因を確認することができます。

- ステップ1 どのサマリ・ビットがアクティブであるか確認します。次のコマンドを使用してください。
*STB? または シリアル・ポーリング
- ステップ2 対応するイベント・レジスタの各サマリ・ビットを読み取り、どのイベントによってサマリ・ビットがセットされたかを確認します。次のコマンドを使用してください。
STATus:QUEStionable:EVENT?
STATus:OPERation:EVENT?
ESR?
イベント・レジスタは、読み取られると、クリアされます。この場合、対応するサマリ・ビットもクリアされます。
- ステップ3 イベントを引き起こした特定の条件を除去します。これが不可能である場合には、ステータス・グループのイネーブル・レジスタまたはNTR|PTRフィルタの対応ビットを設定することによって、イベントをディスエーブルにすることもできます。割込みを防ぐ簡単な方法としては、サービス・リクエスト・イネーブル・レジスタの該当ビットを設定することによってサービス要求をディスエーブルにします。

動作ステータス・イベントおよびクェスチョナブル・ステータス・イベントの処理

本例では、dcソースがCC(定電流)動作モードに切り替わった時か、dcソースの過電圧回路、過電流回路または過熱回路が作動した時に、常にサービス要求が生成されます。図3-7には、動作ステータス・レジスタのビット10(CC)の条件によってステータス・バイト・レジスタのビット6(RQS)をセットするための経路が示されています。また、ビット0、ビット1、ビット4のクェスチョナブル・ステータス条件によって、ステータス・バイト・レジスタでサービス要求(RQS)を生成するための経路も示されています。レジスタを次のように設定する必要があります。

ステップ1 動作ステータスPTRレジスタのビット10の正方向遷移が動作ステータス・イベント・レジスタにラッチされ、ラッチされたイベントが動作サマリ・ビットに加算されるように、動作ステータスPTRレジスタを設定します。次のコマンドを使用してください。

```
STATUS:OPERation:PTR 1024;ENABle 1024
```

ステップ2 クェスチョナブル・ステータスPTRレジスタのビット0、ビット1またはビット4の正方向遷移がクェスチョナブル・ステータス・イベント・レジスタにラッチされ、ラッチされたイベントがクェスチョナブル・サマリ・ビットに加算されるように、クェスチョナブル・ステータスPTRレジスタを設定します。次のコマンドを使用してください。

```
STATUS:QUEStionable:PTR 19;ENABle 19 (1+2+16=19)
```

ステップ3 ステータス・バイト・レジスタからの動作サマリ・ビットとクェスチョナブル・サマリ・ビットの両方がRQSを生成するように、サービス・リクエスト・イネーブル・レジスタを設定します。次のコマンドを使用してください。

```
*SRE 136 (8+128=136)
```

ステップ4 要求を処理する場合には、イベント・レジスタを読み取って、動作ステータス・イベント・レジスタとクェスチョナブル・ステータス・イベント・レジスタのどのビットがセットされているかを確認し、次のイベントに備えてレジスタをクリアします。次のコマンドを使用してください。

```
STATUS:OPERation:EVENT;QUEStionable:EVENT?
```

ステータス遷移の両相のモニタ

ステータス信号の正方向遷移と負方向遷移の両方をモニタすることができます。例えば、dcソースがCC+(定電流)状態に入るか、CC+状態を出た場合にRQSを生成するには、動作ステータスPTR/NTRフィルタを次のように設定します。

```
STATUS:OPERational:PTR 1024;NTR 1024
```

```
STATUS:OPERational:ENABle 1024;*SRE 128
```

PTRフィルタによって、CC+が発生した時にOPERationalサマリ・ビットがRQSをセットします。コントローラが、次にSTATUS:OPERational:EVENT?コマンドによってイベント・レジスタを読み取ると、レジスタはクリアされます。CC+が偽になると、NTRフィルタによってOPERationalサマリ・ビットが再びRQSをセットします。

インヒビット/フォールト・インジケータ

リモート・インヒビット(INH)インジケータと個別フォールト(FLT)インジケータが、それぞれ**INH**および**FLT**として裏面パネル上に実装されています。電気パラメータについては、表1-2を参照してください。

リモート・インヒビット(RI)

リモート・インヒビットは、裏面パネルのINHから入力され、外部機器がフォールトを信号で知らせることができるようになります。リモート・インヒビット信号の動作モードを選択するには、以下を使用します。

```
OUTPut:RI:MODE LATChing | LIVE | OFF
```

個別フォールト・インジケータ(DFI)

個別フォールト・インジケータは、裏面パネルのFLTに接続されているオープン・コレクタ論理信号で、フォールト条件が検出された場合に外部機器に信号を送るのに使用することができます。この信号をドライブする内部フォールト・ソースを選択するには、以下を使用します。

```
OUTPut:DFI:SOURce QUEStionable | OPERation | ESB | RQS | OFF
```

DFI出力をイネーブル/ディスエーブルするには、以下を使用します。

```
OUTPut:DFI:STATe ON | OFF
```

インヒビット/フォールト・ポートをデジタルI/Oとして使用するには

カスタム・デジタル・インタフェース回路またはリレー回路と一緒に使用されるデジタル入出力を提供するように、インヒビット/フォールト・ポートを構成することができます。工場出荷時には、このポートはインヒビット/フォールト処理用に構成されています。以下のコマンドを用いることによって、汎用デジタル入出力ポートとして動作するようにポートの構成を変更することができます。

```
[SOURce:]DIGital:FUNction RIDFi | DIGio
```

下表は、デジタルI/Oモードだけでなく、RI/DFIモードで使用した場合のプラグのピン割当てを示したものです。ポートの電気特性については、表1-2を参照してください。

ピン	フォールト/禁止	デジタルI/O	ビット重み
1	FLT出力	OUT 0	0
2	FLT出力	OUT 1	1
3	INH入力	IN/OUT 2	2
4	INHコモン	コモン	プログラム不可能

デジタルI/Oポートをプログラムするには、以下を使用します。

```
[SOURce:]DIGital:DATA <データ>
```

この場合、データは、データの2進重みに応じてピン1～3を設定する0～7の範囲の整数です。詳細については、DIGital:DATAコマンドを参照してください。

DFIプログラミング例

以下のプログラムは、OCP条件によって装置の出力がオフにされている場合に、DFIポートをローになるようにプログラムするための方法を示したものです。過電流条件をクリアするには、過電流状態の原因をまず取り除いてから、OUTPUT:PROTECTION:CLEARコマンドを送信してください。ステータス・イベント・レジスタは、読み取られるまで、DFIポートをクリアしないということに注意してください。

```

10  !Rev A.00.00
20  ASSIGN @Ps TO 705
30  OUTPUT @Ps;"*RST"                ! 電源をデフォルト値に設定します
40  OUTPUT @Ps;"OUTP ON"             ! 電源出力をオンにします
50  OUTPUT @Ps;"VOLT 10;CURR .1"     ! 電源の電圧および電流をプログラムします
60  !
70  OUTPUT @Ld;"CURR:PROT:STAT ON"   ! 過電流保護をオンにします
80  OUTPUT @Ld;"OUTP:DFI:STAT ON"   ! DFIポートをオンにします
90  OUTPUT @Ld;"OUTP:DFI:SOUR QUES"  ! クェスチョナブル・ステータス・レジスタのDFIビットを選択します。
100 OUTPUT @Ld;"STAT:QUES:ENAB 2;PTR 2" ! 正方向遷移のビット2(OCP)のマスクを除去します
110 !
120 OUTPUT @Ld;"OUTP:PROT:CLE"       ! 保護回路をクリアします
130 OUTPUT @Ld;"STAT:QUES:EVENT?"    ! イベント・レジスタおよびDFIをクリアします
140 OUTPUT @Ld;EVENT                 ! イベントを読み取り、パッファをクリアします
190 !

```


言語辞書

概要

本項では、dcソースで使用されるIEEE 488.2 SCPIコマンドと共通コマンドの構文およびパラメータについて説明します。本項は、第2章「リモート・プログラミング」の内容を十分に理解された上でお読みください。第2章では、本項で使用されている用語、シンボル、構文構造のほか、設定の概要についても説明しています。dcソースの機能を理解するためには、第4章「フロント・パネル操作」(操作ガイド)の内容も十分に理解しておく必要があります。

設定例は、SCPIコマンドの簡単なアプリケーションです。SCPI構文はすべてのプログラミング言語に共通しているため、各コマンドに対する例は一般的なものです。

構文形式構文定義はロング形式を使用しますが、例ではショート形式のヘッダ(キーワード)だけが示されています。ロング形式を用いると、プログラムがそのまま内容の説明として利用できます。

パラメータ ほとんどのコマンドがパラメータを必要とします。また、すべてのクエリがパラメータを返します。パラメータの範囲は、dcソースのモデルによって異なる場合があります。その場合には、操作ガイドの仕様の表を参照してください。

モデル コマンドが特定モデルにしか当てはまらない場合には、<モデル>の場合のみの欄にそれらのモデルが示されています。<モデル>の場合のみという項目がない場合には、そのコマンドはすべてのモデルに適用されます。

関連コマンド 関連するコマンドまたはクエリがある場合、そのコマンドやクエリが記載されています。関連コマンドは、本来のコマンドやクエリと直接、機能的に関係している場合、または本来のコマンドやクエリを理解する上で役に立つ場合に記載されています。

掲載の順序 辞書は、校正、測定、出力、ステータス、システム、トリガの順で機能別に作成されています。サブシステム・コマンドと共通コマンドは、機能ごとにアルファベット順に並んでいます。

サブシステム・コマンド

サブシステム・コマンドは、dcソースの機能に固有の、単一コマンドまたはコマンド・グループです。コマンド・グループは、ルートから1レベル以上、下にあるコマンドから成ります。

サブシステム・コマンド・グループは、機能別に校正、測定、出力、ステータス、システム、トリガに分かれています。また、各機能のコマンドは、アルファベット順に分けられています。後に疑問符(?)が付くコマンドは、クエリ形式しかとりません。コマンドがコマンド形式とクエリ形式の両方をとる場合には、構文説明にそのことが明記されています。表4-1に、すべてのサブシステム・コマンドをアルファベット順でリストアップします。

表4-1. サブシステム・コマンドの構文

ABORt	トリガ・システムをアイドル・ステートにリセットします。
CALibrate	
:CURRent	
[:SOURce]	
[:DC] [:POSitive]	正の出力電流と高電流測定レンジを校正します。
:NEGative	負の出力電流を校正します。
:MEASure	
[:DC] :LOWRange	低電流測定レンジを校正します。
:AC	AC電流測定回路を校正します。
:DATA <n>	校正測定値を入力します。
:LEVel <レベル>	次の校正ステップに進みます(P1 P2)。
:PASSword <n>	校正パスワードを設定します。
:SAVE	新しい校正定数を不揮発性メモリにセーブします。
:STATE <ブール> [, <n>]	校正モードをイネーブル/ディスエーブルにします。
:VOLTag	
[:DC]	出力電圧と電圧リードバックを校正します。
:PROTECTio	電圧保護校正シーケンスを開始します。
DISPlay	
[:WINDow]	
[:STATE] <ブール>	フロント・パネルの表示をイネーブル/ディスエーブルにします。
:MODE <モード>	表示モードを設定します(NORM TEXT)。
:TEXT [:DATA] <文字列>	表示するテキストを設定します。
INITiate	
[:IMMediate]	
:SEquence [<n>]	特定の番号のシーケンスを開始します(1 2)。
:NAME <名前>	特定の名前のシーケンスを開始します(TRAN ACQ)。
CONTinuous	
:SEquence[1], <ブール>	連続初期化を設定します。
:NAME TRANsient, <ブール>	連続初期化を設定します。
MEASure FETCh	
:ARRay	
:CURRent [:DC]?	デジタル化された瞬時電流を返します。
:VOLTag [:DC]?	デジタル化された瞬時電圧を返します。
[:SCALar]	
:CURRent [:DC]?	DC電流を返します。
:ACDC?	総実効値電流を返します(AC+DC)。
:HIGH?	電流パルスのHIGHレベルを返します。
:LOW?	電流パルスのLOWレベルを返します。
:MAX?	最大電流を返します。
:MIN?	最小電流を返します。
:VOLTag [:DC]?	DC電圧を返します。
:ACDC?	総実効値電圧を返します(AC+DC)。
:HIGH?	電圧パルスのHIGHレベルを返します。
:LOW?	電圧パルスのLOWレベルを返します。
:MAX?	最大電圧を返します。
:MIN?	最小電圧を返します。

表4-1. サブシステム・コマンドの構文 (続き)

OUTPut		
[:STATe] <ブール> [,NORelay]		dcソースの出力をイネーブル/ディスエーブルにします。
:DFI		
[:STATe] <ブール>		DFIの出力をイネーブル/ディスエーブルにします。
:SOURce <ソース>		イベント源を選択します(QUES OPER ESB RQS OFF)。
:PON		
:STATe <ステート>		パワーオン・ステートを設定します(*RST RCL0)。
:PROTection		
:CLEar		ラッチ保護をリセットします。
:DELAy <n>		設定後/保護前に遅延させます。
:RELAy		
[:STATe] <ブール>		外部リレー接点をオープン/クローズします。
:POLarity <ポラリティ>		外部リレー・ポラリティを設定します(NORM REV)。
:RI		
:MODE <モード>		リモート・インヒビット入力を設定します(LATC LIVE OFF)。
SENSE		
:CURRent		
[:DC]		
RANGe [:UPPer] <n>		高電流測定レンジを選択します。
:DETeCTOR <ディテクタ>		電流測定ディテクタを選択します(ACDC DC)。
:FUNction <機能>		測定センサを設定します("VOLT" "CURR")。
:SWEep		
:OFFSet		
:POINts <n>		データ掃引のオフセットを定義します。
:POINts <n>		掃引のデータ・ポイント数を定義します。
:TINterval <n>		デジタルのサンプリング間隔を設定します。
:WINDow [:TYPE] <タイプ>		測定ウィンドウ関数(HANN RECT)を設定します。
[SOURce:]		
CURRent		
[:LEVel]		
[:IMMediate][:AMPLitude] <n>		出力電流レベルを設定します。
:TRIGgered [:AMPLitude] <n>		トリガ出力電流レベルを設定します。
:PROTection		
:STATe <ブール>		電流リミット保護をイネーブル/ディスエーブルにします。
DIGital		
:DATA [:VALue] <n>		デジタル制御ポートを設定し、読み取ります。
:FUNction <機能>		デジタル制御ポートを設定します(RIDF DIG)。
VOLTage		
[:LEVel]		
[:IMMediate][:AMPLitude] <n>		DC電圧レベルを設定します。
:TRIGgered [:AMPLitude] <n>		過渡電圧レベルを設定します。
:ALC		
:BANDwidth? :BWIDth?		出力モード・スイッチの設定を返します。
:PROTection [:LEVel] <n>		過電圧保護しきい値を設定します。

表4-1. サブシステム・コマンドの構文 (続き)

STATus	
:PRESet	すべてのイネーブルおよび遷移レジスタをパワーオンにプリセットします。
:OPERation	
[:EVENT]?	イベント・レジスタの値を返します。
:CONDition?	条件レジスタの値を返します。
:ENABle <n>	イベント・レジスタの特定のビットをイネーブルにします。
:NTRansition<n>	負方向遷移フィルタを設定します。
:PTRansition<n>	正方向遷移フィルタを設定します。
:QUEStionable	
[:EVENT]?	イベント・レジスタの値を返します。
:CONDition?	条件レジスタの値を返します。
:ENABle <n>	イベント・レジスタの特定のビットをイネーブルにします。
:NTRansition<n>	負方向遷移フィルタを設定します。
:PTRansition<n>	正方向遷移フィルタを設定します。
SYSTem	
:ERRor?	エラー番号とエラー文字列を返します。
:LANGuage <言語>	プログラミング言語を設定します(SCPI COMP)。
:VERSion?	SCPIバージョン番号を返します。
:LOCal	ローカル・モードに入ります(RS-232の場合)。
:REMOte	リモート・モードに入ります(RS-232の場合)。
:RWLock	ローカルをロックアウトした状態でリモート・モードに入ります(RS-232の場合)。
TRIGger	
:SEQuence2 :ACQuire	
[:IMMediate]	測定を即座にトリガします。
:COUNt	
:CURRent <n>	電流測定当たりの掃引数を設定します。
:VOLTage <n>	電圧測定当たりの掃引数を設定します。
:HYSTeresis	
:CURRent <n>	電流を測定する際のトリガを指定します。
:VOLTage <n>	電圧を測定する際のトリガを指定します。
:LEVel	
:CURRent <n>	電流を測定する際のトリガ・レベルを設定します。
:VOLTage <n>	電圧を測定する際のトリガ・レベルを設定します。
:SLOPe	
:CURRent <スロープ>	トリガ電流スロープを設定します(POS NEG EITH)。
:VOLTage <スロープ>	トリガ電圧スロープを設定します(POS NEG EITH)。
:SOURce <ソース>	トリガ源を設定します(BUS INT)。
[:SEQuence1 :TRANsient]	
[:IMMediate]	出力を即座にトリガします。
:SOURce <ソース>	トリガ源を設定します(BUS)。
:SEQuence1	
:DEFine TRANsient	SEQ1名を設定/問い合わせします。
:SEQuence2	
:DEFine ACQuire	SEQ2名を設定/問い合わせします。

共通コマンド

共通コマンドは、*で始まり、3文字(コマンド)または3文字+(クエリ)から成ります。共通コマンドは、IEEE 488.2規格によって共通のインタフェース機能を実行するために定義されています。共通コマンドとクエリは、機能別にシステム、ステータス、トリガに分類され、各グループの終わりにリストアップされています。dcソースは、以下のコマンドに応答します。

表4-2. 共通コマンドの構文

*CLS	ステータスをクリアします。
*ESE <n>	標準イベント・ステータス・イネーブル
*ESE?	標準イベント・ステータス・イネーブルを返します。
*ESR?	イベント・ステータス・レジスタを返します。
*IDN?	測定器識別を返します。
*OPC	ESRの「動作完了」ビットをイネーブルにします。
*OPC?	動作が完了した時に "1" を返します。
*OPT?	オプション番号を返します。
*PSC <ブール>	パワーオン・ステータス・クリア・ステートの設定/リセット
*PSC?	パワーオン・ステータス・クリア・ステートを返します。
*RCL <n>	機器ステートをリコールします。
*RST	リセットします。
*SAV <n>	機器ステートをセーブします。
*SRE <n>	サービス・リクエスト・イネーブル・レジスタをセットします。
*SRE?	サービス・リクエスト・イネーブル・レジスタを返します。
*STB?	ステータス・バイトを返します。
*TRG	トリガします。
*TST?	セルフテストを実行し、結果を返します。
*WAI	デバイス・コマンドがすべて完了するまでバスをホールドオフします。

設定パラメータ

下の表に、各モデルの出力設定パラメータの一覧を示します。

表4-3. 出力設定パラメータ

パラメータ	値					
	66312A	66332A	6631B 6611C	6632B 6612C	6633B 6613C	6634B 6614C
[SOUR:]CURR[:LEV][:IMM] MAX および [SOUR:]CURR[:LEV]:TRIG MAX *RST 電流値	2.0475	5.1188	<u>10.237</u> 5.1188	<u>5.1188</u> 2.0475	<u>2.0475</u> 1.0238	<u>1.0238</u> 0.5118
	MAX値の10%(全モデル)					
[SOUR:]VOLT[:LEV][:IMM] MAX および [SOUR:]VOLT[:LEV]:TRIG MAX *RST 電圧値	20.475	20.475	8.190	20.475	51.188	102.38
	0 V(全モデル)					
[SOUR:]VOLT:PROT[:LEV] MAX *RST OVP値	22	22	12	22	55	110
	MAX(全モデル)					
OUTP:PROT:DEL MAX *RST 保護遅延値	2,147,483.647秒(全モデル) 0.08秒					
SENS:CURR:RANG *RST 電流レンジ値	ロー・レンジ = 0 – 20 mA(全モデル) ハイ・レンジ = 20 mA – MAX(全モデル) MAX(全モデル)					

校正コマンド

校正コマンドには、次の機能があります。

- 校正モードをイネーブル/ディスエーブルにします。
- 校正パスワードを変更します。
- 電流および電圧の設定と測定を校正し、新しい校正定数を不揮発性メモリにストアします。

注記： 校正モードを CALibrate:STATe でイネーブルにしなかった場合、校正コマンドを設定するとエラーになります。

CALibrate:CURRent

このコマンドは、正のDC出力電流とハイレンジ電流測定回路の校正を開始します。

コマンド構文	CALibrate:CURRent[:SOURce][:DC][:POSitive]
パラメータ	なし
例	CAL:CURRCAL:CURR:SOUR:DC:POS
関連コマンド	CAL:CURR:NEG

CALibrate:CURRent:NEGative

このコマンドは、負のDC出力電流の校正を開始します。

コマンド構文	CALibrate:CURRent[:SOURce][:DC]:NEGative
パラメータ	なし
例	CAL:CURR:NEGCAL:CURR:SOUR:DC:NEG
関連コマンド	CAL:CURR

CALibrate:CURRent:MEASure:LOWRange

このコマンドは、ローレンジ電流測定回路の校正を開始します。

コマンド構文	CALibrate:CURRent:MEASure[:DC]:LOWRange
パラメータ	なし
例	CAL:CURR:MEAS:LOWR
関連コマンド	CAL:CURR

CALibrate:CURRent:MEASure:AC

Agilent 66312A, 66332Aの場合のみ

このコマンドは、高帯域幅(AC)測定回路の校正を開始します。

コマンド構文	CALibrate:CURRent:MEASure:AC
パラメータ	なし
例	CAL:CURR:MEAS:AC

CALibrate:DATA

このコマンドは、外部メータの読取りによって得た校正値を入力します。まず、入力する値に対し校正レベルを選択する必要があります(CALibrate:LEVelを使用)。

コマンド構文	CALibrate:DATA <NRf>
パラメータ	<外部読取り値>
単位	A(アンペア)
例	CAL:DATA 3222.3 MA CAL:DATA 5.000
関連コマンド	CAL:TAT CAL:LEV

CALibrate:LEVel

このコマンドは、校正シーケンスの次のステートに進む場合に使用します。

P1: 最初の校正ポイント

P2: 2番目の校正ポイント

コマンド構文	CALibrate:LEVel <ポイント>
パラメータ	P1 P2
例	CAL:LEV P2

CALibrate:PASSword

このコマンドで校正パスワードを変更することができます。新しいパスワードは、不揮発性メモリに自動的にストアされるので、CALibrate:SAVEを使用してストアする必要はありません。

パスワードを0に設定した場合、パスワード保護が解除されるので、自由に校正モードに入ることができます。

コマンド構文	CALibrate:PASScode <NRf>
パラメータ	<モデル番号>(デフォルト)
例	CAL:PASS 6812 CAL:PASS 6.1994
関連コマンド	CAL:SAV

CALibrate:SAVE

このコマンドは、校正手順の完了後に、すべての新しい校正定数を不揮発性メモリにセーブします。CALibrate:SAVEを使わずにCALibrate:STATe OFFを設定した場合、前の校正定数が復元されます。

コマンド構文	CALibrate:SAVE
パラメータ	なし
例	CAL:SAVE
関連コマンド	CAL:PASS CAL:STAT

CALibrate:STATe

このコマンドは、校正モードをイネーブル/ディスエーブルにします。校正モードをイネーブルにしないと、dcソースは他の校正コマンドを受け入れることができません。

最初のパラメータは、ステートのイネーブルまたはディスエーブルを指定します。2番目のパラメータはパスワードです。校正モードがイネーブルで、現在のパスワードが0でない場合に必要となります。パスワードを入力しなかったり、パスワードが間違っている場合、エラーが発生し、校正モードはディスエーブルのままになります。クウェリ文は、パスワードでなく、ステートだけを返します。

注記： 校正ステートをイネーブルからディスエーブルに変更すると、新しい校正定数は、CALibrate:SAVEでストアしない限り失われてしまいます。

コマンド構文	CALibrate:STATe<ブール>[,<NRf>]
パラメータ	0 1 OFF ON [<パスワード>]
*RST値	OFF
例	CAL:STAT 1,6812 CAL:STAT OFF
クウェリ構文	CALibrate:STATe?
戻りパラメータ	<NR1>
関連コマンド	CAL:PASS CAL:SAVE *RST

CALibrate:VOLTage

このコマンドは、出力電圧および電圧リードバック回路の校正を開始します。

コマンド構文	CALibrate:VOLTage[:DC]
パラメータ	なし
例	CAL:VOLT CAL:VOLT:DC

CALibrate:VOLTage:PROTection

このコマンドは、dcソースの過電圧保護(OV)回路を校正することができます。dcソースは、校正を自動的に実行します。CALibrate:VOLTage:PROTectionはシーケンシャル・コマンドで、完了までに数秒かかります。

コマンド構文	CALibrate:VOLTage:PROTection
パラメータ	なし
例	CAL:VOLT:PROT

測定コマンド

測定コマンドには、測定コマンドとセンス・コマンドがあります。

測定コマンドは、出力電圧と出力電流を測定します。測定では、定義したサンプル数やサンプル間隔における瞬時出力電圧または電流をデジタル化し、結果をバッファにストアして、測定結果を計算します。MEASureとFETChの2種類の測定コマンドが使用可能です。MEASureは、新しいデータの捕捉をトリガしてから読取り値を返します。FETChは、前に収集したデータから計算した読取り値を返します。

- 測定を他のイベントと同期させる必要がない場合、MEASureを使用します。
- 測定がトリガか出力波形の特定部分と同期していることが重要な場合、FETChを使用します。

センス・コマンドは、電流測定レンジ、dcソースの帯域幅ディテクタおよびデータ捕捉シーケンスを制御します。

MEASure:ARRay:CURRent?

FETCh:ARRay:CURRent?

Agilent 66312A, 66332Aの場合のみ

これらのクエリは、アンペア単位の瞬時出力電流を含む配列を返します。出力電圧および電流は、測定コマンドを受け取るか、捕捉トリガが発生するたびにデジタル化されます。時間間隔は、SENSe:SWEEp:TINTervalによって設定します。データ・バッファの開始位置に相対したトリガ位置は、SENSe:SWEEp:OFFSetによって指定します。返すポイントの数は、SENSe:SWEEp:POINtsによって設定します。

クエリ構文	MEASure:ARRay:CURRent[:DC]?	
	FETCh:ARRay:CURRent[:DC]?	
パラメータ	なし	
例	MEAS:ARR:CURR?	FETC:ARR:CURR?
戻りパラメータ	<NR3>	
関連コマンド	SENS:SWEE:TINT	SENS:SWEE:OFFS SENS:SWEE:POIN

MEASure:ARRay:VOLTage?

FETCh:ARRay:VOLTage?

Agilent 66312A, 66332Aの場合のみ

これらのクエリは、ボルト単位の瞬時出力電圧を含む配列を返します。出力電圧および電流は、測定コマンドを受け取るか、捕捉トリガが発生するたびにデジタル化されます。時間間隔は、SENSe:SWEEp:TINTervalによって設定します。データ・バッファの開始位置に相対したトリガ位置は、SENSe:SWEEp:OFFSetによって指定します。返すポイントの数は、SENSe:SWEEp:POINtsによって設定します。

クエリ構文	MEASure:ARRay:VOLTage[:DC]?	
	FETCh:ARRay:VOLTage[:DC]?	
パラメータ	なし	
例	MEAS:ARR:VOLT?	FETC:ARR:VOLT?
戻りパラメータ	<NR3>	
関連コマンド	SENS:SWEE:TINT	SENS:SWEE:OFFS SENS:SWEE:POIN

4 - 言語辞書

MEASure:CURRent?

FETCh:CURRent?

FETCh:CURRent?はAgilent 66312A, 66332Aの場合のみ適用

これらのクエリは、DC出力電流を返します。

クエリ構文	MEASure[:SCALar]:CURRent[:DC]?
	FETCh[:SCALar]:CURRent[:DC]?
パラメータ	なし
例	MEAS:CURR? MEAS:CURR:DC?
戻りパラメータ	<NR3>
関連コマンド	MEAS:VOLT?

MEASure:CURRent:ACDC?

FETCh:CURRent:ACDC?

Agilent 66312A, 66332Aの場合のみ

これらのクエリは、AC+DC実効値出力電流を返します。

クエリ構文	MEASure[:SCALar]:CURRent:ACDC?
	FETCh[:SCALar]:CURRent:ACDC?
パラメータ	なし
例	MEAS:CURR:ACDC? FETC:CURR:ACDC?
戻りパラメータ	<NR3>
関連コマンド	MEAS:VOLT:ACDC?

MEASure:CURRent:HIGH?

FETCh:CURRent:HIGH?

Agilent 66312A, 66332Aの場合のみ

これらのクエリは、電流パルス波形のハイ・レベル電流を返します。測定器は、まず、パルス波形の最小および最大データ・ポイントを測定します。次に、最大データ・ポイントと最小データ・ポイントの間の1024ビンを使用して、パルス波形のヒストグラムを生成します。ほとんどのデータ・ポイントが50%ポイントを超えるビンがハイ・ビンです。ハイ・ビン内のすべてのデータ・ポイントの平均がハイ・レベルとして返されます。捕捉ポイントの総数の1.25%以上を含むハイ・ビンがない場合には、これらのクエリは最大値を返します。

クエリ構文	MEASure[SCALar]:CURRent:HIGH?
	FETCh[:SCALar]:CURRent:HIGH?
パラメータ	なし
例	MEAS:CURR:HIGH? FETC:CURR:HIGH?
戻りパラメータ	<NR3>
関連コマンド	MEAS:CURR:LOW? CALC:REF:HIGH

MEASure:CURRent:LOW?

FETCh:CURRent:LOW?

Agilent 66312A, 66332Aの場合のみ

これらのクエリは、電流パルス波形のロー・レベル電流を返します。測定器は、まず、パルス波形の最小および最大データ・ポイントを測定します。次に、最大データ・ポイントと最小データ・ポイントの間の1024ビンを使用して、パルス波形のヒストグラムを生成します。ほとんどのデータ・ポイントが50%より下であるビンがロー・ビンです。ロー・ビン内のすべてのデータ・ポイントの平均がロー・レベルとして返されます。捕捉ポイントの総数の1.25%以上を含むロー・ビンがない場合には、これらのクエリは最小値を返します。

クエリ構文	MEASure[SCALar]:CURRent:LOW?
	FETCh[SCALar]:CURRent:LOW?
パラメータ	なし
例	MEAS:CURR:LOW? FETC:CURR:LOW?
戻りパラメータ	<NR3>
関連コマンド	MEAS:CURR:HIGH? CALC:REF:LOW

MEASure:CURRent:MAXimum?

FETCh:CURRent:MAXimum?

Agilent 66312A, 66332Aの場合のみ

これらのクエリは、最大出力電流を返します。

クエリ構文	MEASure[:SCALar]:CURRent:MAXimum?
	FETCh[SCALar]:CURRent:MAXimum?
パラメータ	なし
例	MEAS:CURR:MAX? FETC:CURR:MAX?
戻りパラメータ	<NR3>
関連コマンド	MEAS:CURR:MIN?

MEASure:CURRent:MINimum?

ETCh:CURRent:MINimum?

Agilent 66312A, 66332Aの場合のみ

これらのクエリは、最小出力電流を返します。

クエリ構文	MEASure[:SCALar]:CURRent:MINimum?
	FETCh[SCALar]:CURRent:MINimum?
パラメータ	なし
例	MEAS:CURR:MIN? FETC:CURR:MIN?
戻りパラメータ	<NR3>
関連コマンド	MEAS:CURR:MAX?

MEASure:VOLTage?

FETCh:VOLTage?

FETCh:VOLTage?はAgilent 66312A, 66332Aの場合のみ適用

これらのクエリは、DC出力電圧を返します。

クエリ構文	MEASure[:SCALar]:VOLTage[:DC]?
	FETCh[:SCALar]:VOLTage[:DC]?
パラメータ	なし
例	MEAS:VOLT? FETC:VOLT:DC?
戻りパラメータ	<NR3>
関連コマンド	MEAS:CURR?

MEASure:VOLTage:ACDC?

FETCh:VOLTage:ACDC?

Agilent 66312A, 66332Aの場合のみ

これらのクエリは、AC+DC実効値出力電圧を返します。

クエリ構文	MEASure[:SCALar]:VOLTage:ACDC?
	FETCh[:SCALar]:VOLTage:ACDC?
パラメータ	なし
例	MEAS:VOLT:ACDC? FETC:VOLT:ACDC?
戻りパラメータ	<NR3>
関連コマンド	MEAS:CURR:ACDC

MEASure:VOLTage:HIGH?

FETCh:VOLTage:HIGH?

Agilent 66312A, 66332Aの場合のみ

これらのクエリは、電圧パルス波形のハイ・レベル電圧を返します。測定器は、まず、パルス波形の最小および最大データ・ポイントを測定します。次に、最大データ・ポイントと最小データ・ポイントの間の1024ビンを使用して、パルス波形のヒストグラムを生成します。ほとんどのポイントが50%ポイントを超えるビンがハイ・ビンです。ハイ・ビン内のすべてのデータ・ポイントの平均がハイ・レベルとして返されます。捕捉ポイントの総数の1.25%以上を含むハイ・ビンがない場合には、これらのクエリは最大値を返します。

クエリ構文	MEASure[SCALar]:VOLTage:HIGH?
	FETCh[SCALar]:VOLTage:HIGH?
パラメータ	なし
例	MEAS:VOLT:HIGH? FETC:VOLT:HIGH?
戻りパラメータ	<NR3>
関連コマンド	MEAS:VOLT:LOW? CALC:REF:HIGH

MEASure:VOLTage:LOW?

FETCh:VOLTage:LOW?

Agilent 66312A, 66332Aの場合のみ

これらのクエリは、電圧パルス波形のロー・レベル電圧を返します。測定器は、まず、パルス波形の最小および最大データ・ポイントを測定します。次に、最大データ・ポイントと最小データ・ポイントの間の1024ビンを使用して、パルス波形のヒストグラムを生成します。ほとんどのデータ・ポイントが50%ポイントより下のビンがロー・ビンです。ロー・ビン内のすべてのデータ・ポイントの平均がロー・レベルとして返されます。捕捉ポイントの総数の1.25%以上を含むロー・ビンがまったくない場合には、これらのクエリは最小値を返します。

クエリ構文	MEASure[SCALar]:VOLTage:LOW?
	FETCh[SCALar]:VOLTage:LOW?
パラメータ	なし
例	MEAS:VOLT:LOW? FETC:VOLT:LOW?
戻りパラメータ	<NR3>
関連コマンド	MEAS:VOLT:HIGH? CALC:REF:LOW

MEASure:VOLTage:MAXimum?

FETCh:VOLTage:MAXimum?

Agilent 66312A, 66332Aの場合のみ

これらのクエリは、最大出力電圧を返します。

クエリ構文	MEASure[:SCALar]:VOLTage:MAXimum?
	FETCh[SCALar]:VOLTage:MAXimum?
パラメータ	なし
例	MEAS:VOLT:MAX? FETC:VOLT:MAX?
戻りパラメータ	<NR3>
関連コマンド	MEAS:VOLT:MIN?

MEASure:VOLTage:MINimum?

FETCh:VOLTage:MINimum?

Agilent 66312A, 66332Aの場合のみ

これらのクエリは、最小出力電圧を返します。

クエリ構文	MEASure[:SCALar]:VOLTage:MINimum?
	FETCh[SCALar]:VOLTage:MINimum?
パラメータ	なし
例	MEAS:VOLT:MIN? FETC:VOLT:MIN?
戻りパラメータ	<NR3>
関連コマンド	MEAS:VOLT:MAX?

SENSe:CURRent:RANGe

このコマンドは、DC電流測定レンジを選択します。すべてのモデルが2つの電流測定レンジを備えています。

ハイ・レンジ: 0~MAX(表4-3参照)

ロー・レンジ: 0~0.02 A(全モデル)

ハイ・レンジは、測定器の全電流測定機能をカバーします。また、ロー・レンジでは、20mAまで電流が測定できます。これによって、低電流測定感度が高まり、確度と分解能が向上します。SENSe:CURRent:RANGeで設定する値は、測定する電流の最大値でなければなりません。測定器は、最高の分解能を実現するレンジを選択します。クロスオーバー値は20mAです。クウェリを行った場合、現在設定されているレンジで測定可能な最大電流が返されます。

コマンド構文	SENSe:CURRent[:DC]:RANGe[:UPPer]<NRf+>
パラメータ	0~MAX(表4-3参照)
単位	A(アンペア)
*RST値	MAX(ハイ・レンジ)
例	SENS:CURR:RANG 4.0
クウェリ構文	SENSe:CURRent:RANGe?
戻りパラメータ	<NR3>

SENSe:CURRent:DETEctor**Agilent 66312A, 66332Aの場合のみ**

このコマンドを使用することによって、出力電流測定に使用するディテクタの種類を選択することができます。電流測定値を検出するには、次の2通りの方法があります。

ACDC これは、すべてのダイナミック電流測定に適しています。ACDCを選択すると、測定器の出力コンデンサ内を流れる電流も出力電流として測定されます。数kHzを超える周波数成分を持つパルスやその他の波形を測定する場合、特に、ACDC検出を選択することが大切です。

DC DC電流測定の実行中に、高電流測定レンジで2mAを超えるDC測定オフセットが必要になった場合にだけDCを選択してください。DCを選択すると、測定器の出力フィルタが供給する出力電流成分は検出されません。したがって、DCを選択した場合、数kHzを超える周波数成分を持つ電流波形は不正確になります。

注記: このコマンドは、高電流測定レンジにのみ使用できます。

コマンド構文	SENSe:CURRent:DETEctor<ディテクタ>
パラメータ	ACDCまたはDC
*RST値	ACDC
例	SENS:CURR:DET ACDC SENS:CURR:DET DC
クウェリ構文	SENSe:CURRent:DETEct?
戻りパラメータ	<CRD>

SENSe:FUNctIon

Agilent 66312A, 66332Aの場合のみ

このコマンドは、捕捉トリガを使用する際、測定センサが電圧と電流のどちらかを測定するように設定します。クウェリは、VOLTまたはCURRの機能設定を返します。

コマンド構文	SENSe:FUNctIon<機能>
パラメータ	"VOLTage" "CURRent"
例	SENS:FUNC "VOLT"
クウェリ構文	SENSe:FUNctIon?
戻りパラメータ	<SRD>

SENSe:SWEp:OFFSet:POINts

Agilent 66312A, 66332Aの場合のみ

このコマンドは、捕捉トリガを使用する際のデータ掃引のオフセットを定義します。負の値は、トリガ前に収集されたデータ・サンプルを表します。

コマンド構文	SENSe:SWEp:OFFSet:POINts <NRf+>
パラメータ	-4095 ~ 2,000,000,000
*RST値	0
例	SENS:SWE:OFFS:POIN -2047
クウェリ構文	SENSe:SWEp:OFFSet:POINts?
戻りパラメータ	<NR3>
関連コマンド	SENS:SWE:TINT SENS:SWE:POIN MEAS:ARR

SENSe:SWEp:POINts

このコマンドは、データ掃引のポイント数を定義します。

コマンド構文	SENSe:SWEp:POINts<NRf+>
パラメータ	0 ~ 4096
*RST値	2048
例	SENS:SWE:POIN 1024
クウェリ構文	SENSe:SWEp:POINts?
戻りパラメータ	<NR3>
関連コマンド	SENS:SWE:TINT SENS:SWE:OFFS MEAS:ARR

SENSe:SWEp:TINTerval

このコマンドは、サンプル間の周期を定義します。

コマンド構文	SENSe:SWEp:TINTerval<NRf+>
パラメータ	15.6 μ s ~ 31200 s
*RST値	15.6 μ s
例	SENS:SWE:TINT 31.2E-6
クウェリ構文	SENSe:SWEp:TINTerval?
戻りパラメータ	<NR3>
関連コマンド	SENS:SWE:POIN SENS:SWE:OFFS MEAS:ARR

SENSe:WINDow

このコマンドは、出力測定計算に使用されるウィンドウ関数を設定します。選択可能な関数としては以下のものがあります。

HANNing ライン・リップルなどの周期信号が存在する場合に、DC や実効値測定の計算誤差を減らすための信号調整ウィンドウ。逐次パルス波形を測定する際には、ジッタも減少させます。ハニング窓は、測定サンプルの各ポイントに関数 \cos^4 を掛けます。単発パルス波形を測定する場合には、ハニング窓を使用しないでください。

RECTangular 信号調整をまったく行わずに測定計算値を返すウィンドウ。このウィンドウは、パルス波形の正確な周期が分かっており、それに応じてSENSe:SWEep:TINTervalコマンドを使用して測定間隔を設定することができるパルス測定に使用できます。

注記： いずれのウィンドウも、測定配列に戻される瞬時電圧データや瞬時電流データには何の影響も及ぼしません。

コマンド構文	SENSe:WINDow [:TYPE] <タイプ>
パラメータ	HANNing RECTangular
*RST値	HANNing
例	SENS:WIND RECT
クエリ構文	SENSe:WINDow [:TYPE]?
戻りパラメータ	<CRD>

出力コマンド

出力コマンドには、出力コマンドとソース・コマンドがあります。

出力コマンドは、dcソースの出力とデジタル・ポート機能を制御します。また、オプション760出力リレーも制御します。

ソース・コマンドは、実電圧、電流およびデジタル・ポート出力を設定します。

OUTPut

このコマンドは、dcソース出力をイネーブル/ディスエーブルにします。出力がディスエーブルのステートとは、出力電圧がゼロ、ソース電流が最小(モデルによって異なる)の状態を言います(*RSTを参照)。NORelayコマンドが設定されていないならば、OUTputコマンドはAgilentモデル66332A、6632B、6633Bおよび6634Bのオプション760出力リレーも制御します。NORelayコマンドが送信された場合には、出力リレー・ステートは変化しません。

コマンド構文	OUTPut[:STATe] <ブール> [,NORelay]
パラメータ	0 OFF 1 ON
*RST値	0
例	OUTP 1 OUTPUT:STATE ON
クウェリ構文	OUTPut[:STATe]?
戻りパラメータ	<NR1> 0 または 1
関連コマンド	*RST *RCL *SAV

OUTPut:DFI

このコマンドは、dcソースからのディスクリート・フォルト・インジケータ(DFI)出力をイネーブル/ディスエーブルにします。

コマンド構文	OUTPut:DFI[:STATe]<ブール>
パラメータ	0 1 OFF ON
*RST値	OFF
例	OUTP:DFI 1 OUTP:DFI ON
クウェリ構文	OUTPut:DFI[:STATe]?
戻りパラメータ	0 1
関連コマンド	OUTP:DFI:SOUR

OUTPut:DFI:SOURce

このコマンドは、ディスクリート・フォルト・インジケータ(DFI)イベントのソースを選択します。選択肢としては以下のものがあります。

QUESTIONable	クエスチョナブル・サマリ・ビット
OPERation	動作サマリ・ビット
ESB	標準イベント・サマリ・ビット
RQS	サービス要求サマリ・ビット
OFF	真になることはなし

コマンド構文	OUTP:DFI:SOUR<ソース>
パラメータ	QUES OPER ESP RQS OFF
*RST値	OFF

4 - 言語辞書

例	OUTP:DFI:SOUR OPER
クウェリ構文	OUTPut:DFI:SOUR?
戻りパラメータ	<CRD>
関連コマンド	OUTP:DFI

OUTPut:PON:STATe

このコマンドは、dcソースのパワーオン・ステートを選択します。この情報は不揮発性メモリにセーブされます。選択可能なステートとしては以下のものがあります。

RST パワーオン・ステートを*RSTに設定します。詳細については、本章の*RSTコマンドに関する説明を参照してください。

RCL0 パワーオン・ステートを*RCL 0に設定します。詳細については、本章の*RCLコマンドに関する説明を参照してください。

コマンド構文	OUTPut:PON:STATE <ステート>
パラメータ	RST RCL0
例	OUTP:PON:STAT RST
クウェリ構文	OUTPut:PON:STATe?
戻りパラメータ	<CRD>
関連コマンド	*RST *RCL

OUTPut:PROTection:CLEar

このコマンドは、過電圧、過電流、過熱、リモート・インヒビットまたはヒューズ・ステータス条件が検出された時に出力をディスエーブルにするラッチをクリアします。障害の原因をすべて取り除かなければ、ラッチをクリアすることはできません。出力は、障害状態が発生する前のステートに復元されます。

コマンド構文	OUTPut:PROTection:CLEar
パラメータ	なし
例	OUTP:PROT:CLE
関連コマンド	OUTP:PROT:DEL *RCL *SAV

OUTPut:PROTection:DELAy

このコマンドは、定電圧または定電流条件を生成する出力変動が設定されてから、動作ステータス条件レジスタが条件を記録するまでの時間を設定します。遅延設定によって、再設定中に発生するdcソースのステータスの瞬時変動が、ステータス・サブシステムによってイベントとして記憶されることはありません。遅延は定電流ステータスに適用されるため、過電流保護機能も遅延されます。過電圧保護はこのコマンドによる影響を受けません。

コマンド構文	OUTPut:PROTection:DELAy <NRf+>
パラメータ	0 ~ 2,147,483,647
単位	秒
*RST値	0.08(ノーマル)
例	OUTPUT:PROTECTION:DELAY 75E-1
クウェリ構文	OUTPut:PROTection:DELAy?
戻りパラメータ	<NR3>
関連コマンド	OUTP:PROT:CLE *RCL *SAV

OUTPut:RELAy

Agilent 66332A, 6632B, 6633B, 6634B, 6611C, 6612C, 6613C, 6614Cの場合のみ

このコマンドは、オプション760が装備された装置に対してのみ有効です。それ以外の場合には、エラーが発生します。ONを設定すると出力リレー接点がクローズし、OFFを設定すると出力リレー接点がオープンします。リレーは、出力ステートと無関係に制御されます。dcソースが負荷に電力を供給している場合には、スイッチング中にリレー接点に電力が生じます。

コマンド構文	OUTPut:RELAy[:STATe]<ブール>
パラメータ	0 1 OFF ON
*RST値	0
例	OUTP:REL 1OUTP:REL OFF
クウェリ構文	OUTPput:RELAy?
戻りパラメータ	0 1
関連コマンド	OUTP *RCL *SAV

OUTPut:RELAy:POLarity

Agilent 66332A, 6632B, 6633B, 6634B, 6611C, 6612C, 6613C, 6614Cの場合のみ

このコマンドは、オプション760が装備された装置に対してのみ有効です。それ以外の場合には、エラーが発生します。NORMAlを設定するとリレー出力ポラリティがdcソース出力と同じになり、REVerseを設定するとリレー出力ポラリティがdcソース出力と反対になります。どちらかのリレー・コマンドを送信した時にOUTPut=ONの場合、リレーのポラリティが変化している間、dcソースの出力電圧は0に設定されます。

コマンド構文	OUTPut:RELAy:POLar ity<CRD>
パラメータ	NORMAl REVerse
*RST値	NORM
例	OUTP:REL:POL NORM
クウェリ構文	OUTPput:RELAy:POLar ity?
戻りパラメータ	NORM REV
関連コマンド	OUTP *RCL *SAV

OUTPut:RI:MODE

このコマンドは、リモート・インヒビット保護の動作モードを選択します。RIモードは不揮発性メモリにストアされます。選択可能なモードとして以下のものがあります。

LATChing INH入力のTTLロー信号で出力をディスエーブルします。ラッチをクリアする唯一の方法は、INH入力がハイである間にOUTPut:PROTection:CLEARを送信することです。

LIVE INH入力を使ってラッチを行わずに出力をディスエーブルにします。つまり、INH入力ステートの状態に応じて出力が行われます。INHがローの真になると、出力はディスエーブルされます。INHがハイの場合、出力は影響を受けません。

OFF INH入力がディスエーブルになります。

コマンド構文	OUTPut:RI:MODE <モード>
パラメータ	LATChing LIVE OFF
例	OUTP:RI:MODE LIVE
クウェリ構文	OUTPut:RI:MODE?
戻りパラメータ	<CRD>
関連コマンド	OUTP:PROT:CLE

[SOURce:]CURRent

このコマンドは、dcソースの即時電流レベルを設定します。即時レベルは、出力端子用に設定された電流です。

コマンド構文	[SOURce]:CURRent[:LEVel][:IMMediate][:AMPLitude]<NRf+>
パラメータ	表4-3を参照
デフォルトの接尾語	A(アンペア)
*RST値	MAXの10%
例	CURR 200 MA CURRENT:LEVEL 200 MA
クエリ構文	[SOURce]:CURRent[:LEVel][:IMMediate][:AMPLitude]?
戻りパラメータ	<NR3>
関連コマンド	CURR:TRIG

[SOURce:]CURRent:TRIGger

このコマンドは、dcソースの保留トリガ電流レベルを設定します。保留トリガ・レベルは、トリガが発生した時に出力端子に転送されるストアされた電流値です。トリガを発生させるには、トリガ・サブシステムが起動されていない必要があります(トリガ・サブシステムのINITiateコマンドを参照してください)。

コマンド構文	[SOURce]:CURRent[:LEVel]:TRIGgered[:AMPLitude]<NRf+>
パラメータ	表4-3を参照
デフォルトの接尾語	A(アンペア)
*RST値	MAXの10%
例	CURR:TRIG 1CURRENT:LEVEL:TRIGGERED 1
クエリ構文	[SOURce]:CURRent[:LEVel]:TRIGgered[:AMPLitude]?
戻りパラメータ	<NR3>
関連コマンド	INIT CURR

[SOURce:]CURRent:PROTection:STATe

このコマンドは、dcソースの過電流保護(OCP)機能をイネーブル/ディスエーブルにします。過電流保護機能がイネーブルになり、dcソースが定電流動作に入ると、出力がディスエーブルになり、クエスチョナブル条件ステータス・レジスタのOCビットがセットされます(第3章の「ステータス・レジスタの設定」の項を参照してください)。過電流状態は、その原因を取り除いた後、OUTPut:PROTection:CLEarコマンドでクリアすることができます。

注記： OUTP:PROT:DELを使用することにより、出力設定の変更によって一時的に電流リミット条件が変わっても、過電流保護機能が作動するのを防ぐことができます。

コマンド構文	[SOURce]:CURRent:PROTection:STATe <ブール>
パラメータ	0 1 OFF ON
*RST値	OFF
例	CURR:PROT:STAT 0CURRENT:PROTECTION:STATE OFF CURR:PROT:STAT 1CURRENT:PROTECTION:STATE ON
クエリ構文	[SOURce]:CURRent:PROTection:STATe?
戻りパラメータ	<NR1> 0または1
関連コマンド	OUTP:PROT:CLE *RST

[SOURce:]DIGital:DATA

このコマンドは、dcソースのデジタル制御ポートがデジタルI/O処理用に設定されている時に、ポートの設定と読取りに使用します。ポートには、3つの信号ピンと1つのデジタル・グラウンド・ピンがあります。ピン1と2は、ビット0と1によって制御される出力ピンです。ピン3は、ビット2によって制御され、入力と出力のどちらかの働きをするように設定することができます。このピンは、通常、出力として機能します。ピン3を入力として使用するには、ビット2をハイに設定する必要があります。ピン4はデジタル・グラウンドです。クウェリは、ビット0と1にある最後に設定した値と、ビット2にあるピン3の読取り値を返します。

プログラム 値	ビット設定			ピン設定			
	2	1	0	4	3	2	1
0	0	0	0	GND	出力	Lo	Lo
1	0	0	1	GND	出力	Lo	Hi
2	0	1	0	GND	出力	Hi	Lo
3	0	1	1	GND	出力	Hi	Hi
4	1	0	0	GND	入力	Lo	Lo
5	1	0	1	GND	入力	Lo	Hi
6	1	1	0	GND	入力	Hi	Lo
7	1	1	1	GND	入力	Hi	Hi

コマンド構文 [SOURce:]DIGital:DATA[:VALue] <NRf>
 パラメータ 0~7
 *RST値 0
 例 DIG:DATA 7
 クウェリ構文 [SOURce:]DIGital:DATA?
 戻りパラメータ <NR1>
 関連コマンド DIG:FUNC

[SOURce:]DIGital:FUNCtion

このコマンドは、dcソースのデジタル制御ポートを設定します。設定値は不揮発性メモリにセーブされます。

RIDFi ポートをリモート・インヒビット/ディスクリット・フォールト割込み処理用に設定します。
DIGio ポートをデジタル入出力処理用に設定します(DIG:DATAを参照してください)。

コマンド構文 [SOURce:]DIGital:FUNCtion <CRD>
 パラメータ RIDFi | DIGio
 例 DIG:FUNC DIG
 クウェリ構文 [SOURce:]DIGital:FUNC?
 戻りパラメータ <CRD>
 関連コマンド DIG:DATA

[SOURce:]VOLTage

このコマンドは、dcソースの出力電圧レベルを設定します。

コマンド構文 [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]<NRf+>
 パラメータ 表4-3を参照
 デフォルトの接尾語 V(ボルト)
 *RST値 0

4 - 言語辞書

例	VOLT 2	VOLTAGE:LEVEL 200 MV
クエリ構文	[SOURce]:VOLTage[:LEVel][:IMMediate][:AMPLitude]?	
戻りパラメータ	<NR3>	
関連コマンド	VOLT:TRIG	

[SOURce:]VOLTage:ALC:BANDwidth?
[SOURce:]VOLTage:ALC:BWIDth?

Agilent 66332A, 6631B, 6632B, 6633B, 6634Bの場合のみ

これらのクエリは、出力モード・スイッチの設定を返します。戻り値は、スイッチがNormalに設定されている場合には15,000、スイッチがFastに設定されている場合には60,000になります。

クエリ構文	[SOURce]:VOLTage:ALC:BANDwidth? [SOURce]:VOLTage:ALC:BWIDth?
例	VOLT:ALC:BAND? VOLTAGE:ALC:BWIDth?
戻りパラメータ	<NR3>

[SOURce:]VOLTage:TRIGger

このコマンドは、dcソースの保留トリガ電圧レベルを設定します。保留トリガ・レベルは、トリガが発生した時に出力端子に転送されるストアされた電圧値です。トリガを発生させるためには、トリガ・サブシステムが起動されていないとなりません(トリガ・サブシステムのINITiateコマンドを参照してください)。

コマンド構文	[SOURce][:VOLTage[:LEVel]:TRIGgered[:AMPLitude]<NRf+>
パラメータ	表4-3を参照
デフォルトの接尾語	V(ボルト)
*RST値	0
例	VOLT:TRIG 20 VOLTAGE:LEVEL:TRIGGERED 20
クエリ構文	[SOURce]:VOLTage[LEVel]:TRIGgered[:AMPLitude]?
戻りパラメータ	<NR3>
関連コマンド	VOLT *RST

[SOURce:]VOLTage:PROTect ion

このコマンドは、dcソースの過電圧保護(OVP)レベルを設定します。出力電圧がOVPレベルを超えると、dcソースの出力がディスエーブルになり、クエリ条件ステータス・レジスタのOVビットがセットされます(第3章の「ステータス・レジスタの設定」の項を参照してください)。過電圧状態は、OVPを作動させた条件を取り除いた後、OUTP:PROT:CLE コマンドを使用してクリアすることができます。OVP は常にゼロ遅延で作動し、OUTP:PROT:DELコマンドによる影響を受けません。

コマンド構文	[SOURce]:VOLTage:PROTect ion[:LEVel]<NRf+>
パラメータ	表4-3を参照
デフォルトの接尾語	V(ボルト)
*RST値	MAX
例	VOLT:PROT 21.5 VOLT:PROT:LEV MAX
クエリ構文	[SOURce]:VOLTage:PROTect ion[:LEVel]?
戻りパラメータ	<NR3>
関連コマンド	OUTP:PROT:CLE OUTP:PROT:DEL

ステータス・コマンド

ステータス・コマンドは、dcソースのステータス・レジスタを設定します。dcソースには、動作、クエスチョナブル、標準イベントの3組のステータス・レジスタがあります。標準イベント・グループは、後に説明するように、共通コマンドによって設定します。動作ステータス・グループとクエスチョナブル・ステータス・グループは、それぞれ、条件レジスタ、イネーブル・レジスタ、イベント・レジスタと、NTRフィルタおよびPTRフィルタから成ります。特定のレジスタ・ビットの読取り方法や、ビットが返す情報の使用方法については、第3章の「ステータス・レジスタの設定」の項を参照してください。

共通コマンドは、ステータス機能も実行します。本項では次の共通コマンドについて説明します：

*CLS、*ESE、*ESR?、*OPC、*PSC、*SRE、*STB、*WAI。

STATus:PRESet

このコマンドは、ステータス・サブシステムのPTRレジスタに定義されているすべてのビットをセットし、ステータス・サブシステムのNTRレジスタとイネーブル・レジスタ内のすべてのビットをクリアします。

コマンド構文 STATus:PRESet
 パラメータ なし
 例 STAT:PRESet STATUS:PRESET

表4-4. ステータス動作レジスタのビット構成

ビット位置	15-12	11	10	9	8	7-6	5	4-1	0
ビット名	未使用	CC-	CC+	未使用	CV	未使用	WTG	未使用	CAL
ビットの重み		2048	1024		256		32		

CAL=dcソースは新しい校正定数を計算中です。
 WTG=dcソースはトリガ待ち状態です。
 CV=dcソースは定電圧モードで動作中です。
 CC+=dcソースは定電流モードで動作中です。
 CC-=dcソースは負の定電流モードで動作中です。

STATus:OPERation?

このクエリは、動作イベント・レジスタの値を返します。イベント・レジスタはリード・オンリ・レジスタで、動作NTRフィルタまたは動作PTRフィルタ、あるいはその両方によって渡されるすべてのイベントを保持(ラッチ)します。動作イベント・レジスタは、読取りが完了するとクリアされます。

クエリ構文 STATus:OPERtion[:EVENT]?
 パラメータ (なし)
 戻りパラメータ <NR1> (レジスタ値)
 例 STAT:OPER? STATUS:OPERATIONAL:EVENT?
 関連コマンド *CLS STAT:OPER:NTR STAT:OPER:PTR

STATus:OPERation:CONDition?

このクエリは、動作条件レジスタの値を返します。条件レジスタはリード・オンリ・レジスタで、dcソースのリアルタイムの(ラッチされていない)動作ステータスを保持します。

クエリ構文 STATus:OPERation:CONDition?
 パラメータ (なし)
 例 STAT:OPER:COND? STATUS:OPERATION:CONDITION?
 戻りパラメータ <NR1> (レジスタ値)

STATus:OPERation:ENABLE

このコマンドとそのクエリは、動作イネーブル・レジスタの値の設定と読取りに使用します。動作イネーブル・レジスタは、動作イベント・レジスタの特定ビットをイネーブルにし、ステータス・バイト・レジスタの動作サマリ・ビット(OPER)をセットするためのマスクです。動作サマリ・ビット(ビット7)は、ステータス動作イネーブル・レジスタによってイネーブルにされた動作イベント・レジスタの全ビットの論理和です。

コマンド構文 STATus:OPERation:ENABLE<NRf>
 パラメータ 0~32727
 初期設定値 0
 例 STAT:OPER:ENAB 1312 STAT:OPER:ENAB 1
 STATUS:OPERATION:ENABLE?
 クエリ構文 STATus:OPERation:ENABLE?
 戻りパラメータ <NR1> (レジスタ値)
 関連コマンド STAT:OPER:EVEN

STATus:OPERation:NTR

STATus:OPERation:PTR

これらのコマンドは、動作NTR(負方向遷移)レジスタとPTR(正方向遷移)レジスタの値の設定と読取りに使用します。これらのレジスタは、動作イネーブル・レジスタと動作イベント・レジスタ間のポラリティ・フィルタとして働き、次の動作を発生させます。

- 動作NTRレジスタ内の任意のビットを1に設定すると、動作条件レジスタ内の対応するビットが1から0に遷移し、動作イベント・レジスタ内のそのビットがセットされます。
- 動作PTRレジスタ内の任意のビットを1に設定すると、動作条件レジスタ内の対応するビットが1から0に遷移し、動作イベント・レジスタ内のそのビットがセットされます。
- NTRとPTRの両方のレジスタの同じビットを1に設定した場合、動作条件レジスタのそのビットのすべての遷移で、動作イベント・レジスタ内の対応ビットがセットされます。
- NTRとPTRの両方のレジスタの同じビットを0に設定した場合、動作条件レジスタのそのビットに何の遷移もないと、動作イベント・レジスタ内の対応ビットをセットすることができます。

コマンド構文 STATus:OPERtion:NTRansition<NRf>
 STATus:OPERtion:PTRansition<NRf>
 パラメータ 0~32727
 初期設定値 NTRレジスタ=0; PTRレジスタ=32727
 例 STAT:OPER:NTR 32 STAT:OPER:PTR 1312
 クエリ構文 STAT:OPER:NTR? STAT:OPER:PTR?
 戻りパラメータ <NR1> (レジスタ値)
 関連コマンド STAT:OPER:ENAB

表4-5. ステータス・クwestionablE・レジスタのビット構成

ビット位置	15	14	13-11	10	9	8-5	4	3	2	1	0
ビット名	未使用	Meas Ovld	未使用	Unreg	RI	未使用	OT	未使用	FS	OCP	OV
ビットの重み		16384		1024	512		16		4	2	1
OV=過電圧保護が作動しました。 OCP=過電流保護が作動しました。 FS=ヒューズが飛びました。 OT=過熱保護が作動しました。 RI=リモート・インヒビットがアクティブです。 Unreg=出力がレギュレーションされていません。 Meas Ovld=測定過負荷											

STATus:QUEStionable?

このクウェリは、クwestionablE・イベント・レジスタの値を返します。イベント・レジスタはリード・オンリ・レジスタで、クwestionablENTRフィルタまたはPTRフィルタ、あるいはその両方によって渡されるすべてのイベントを保持(ラッチ)します。クwestionablE・イベント・レジスタは、読取りが完了するとクリアされます。

クウェリ構文 STATus:QUEStionable[:EVENT]?

パラメータ なし

例 STAT:QUES? STATUS:QUESTIONABLE:EVENT?

戻りパラメータ <NR1> (レジスタ値)

関連コマンド *CLS STAT:QUES:ENAB
STAT:QUES:NTR STAT:QUES:PTR

STATus:QUEStionable:CONDition?

このクウェリは、クwestionablE条件レジスタの値を返します。条件レジスタはリード・オンリ・レジスタで、dcソースのリアルタイムの(ラッチされていない)クwestionablE・ステータスを保持します。

クウェリ構文 STATus:QUEStionable:CONDition?

パラメータ なし

例 STAT:QUES:COND? STATUS:QUESTIONABLE:CONDITION?

戻りパラメータ <NR1> (レジスタ値)

STATus:QUEStionable:ENABle

このコマンドとそのクウェリは、クwestionablE・イネーブル・レジスタの値の設定と読取りに使用します。クwestionablE・イネーブル・レジスタは、クwestionablE・イベント・レジスタの特定ビットをイネーブルにし、ステータス・バイト・レジスタのクwestionablE・サマリ・ビット(QUES)をセットするためのマスクです。クwestionablE・サマリ・ビット(ビット3)は、クwestionablE・ステータス・イネーブル・レジスタによってイネーブルにされたクwestionablE・イベント・レジスタの全ビットの論理和です。

コマンド構文 STATus:QUEStionable:ENABle<NRf>

パラメータ 0~32767

初期設定値 0

例 STAT:QUES:ENAB 20 STAT:QUES:ENAB 16

クウェリ構文 STATus:QUEStionable:ENABle?

戻りパラメータ <NR1> (レジスタ値)

関連コマンド STAT:QUES?

STATus:QUEStionable:NTR

STATus:QUEStionable:PTR

これらのコマンドは、クエスチョナブルNTR(負方向遷移)レジスタやPTR(正方向遷移)レジスタの値の設定や読取りに使用できます。これらのレジスタは、クエスチョナブル・イネーブル・レジスタとクエスチョナブル・イベント・レジスタ間のポラリティ・フィルタとして働き、次の動作を発生させます。

- クエスチョナブルNTRレジスタ内の任意のビットを1に設定すると、クエスチョナブル条件レジスタ内の対応するビットが1から0に遷移し、クエスチョナブル・イベント・レジスタ内のそのビットがセットされます。
- クエスチョナブルPTRレジスタ内の任意のビットを1に設定すると、クエスチョナブル条件レジスタ内の対応するビットが1から0に遷移し、クエスチョナブル・イベント・レジスタ内のそのビットがセットされます。
- NTRとPTRの両方のレジスタの同じビットを1に設定した場合、クエスチョナブル条件レジスタのそのビットのすべての遷移で、クエスチョナブル・イベント・レジスタ内の対応するビットがセットされます。
- NTRとPTRの両方のレジスタの同じビットを0に設定した場合、クエスチョナブル条件レジスタのそのビットに何の遷移もないと、クエスチョナブル・イベント・レジスタ内の対応するビットをセットすることができます。

コマンド構文	STATus:QUEStionable:NTRansition<NRf> STATus:QUEStionable:PTRansition<NRf>
パラメータ	0~32727
初期設定値	NTRレジスタ=0; PTRレジスタ=32727
例	STAT:QUES:NTR 16 STATUS:QUESTIONABLE:PTR 512
クウェリ構文	STAT:QUES:NTR? STAT:QUES:PTR?
戻りパラメータ	<NR1> (レジスタ値)
関連コマンド	STAT:QUES:ENAB

*CLS

このコマンドは、以下の動作を引き起こします(各レジスタの詳細については、第3章の「ステータス・レジスタの設定」の項を参照してください)。

- 以下のレジスタをクリアします。
 - 標準イベント・ステータス
 - 動作ステータス・イベント
 - クエスチョナブル・ステータス・イベント
 - ステータス・バイト
- エラー待ち行列をクリアします。
- プログラム・メッセージ・ターミネータ(<NL>)の直後に*CLSがある場合には、出力待ち行列とMAVビットもクリアされます。

コマンド構文	*CLS
パラメータ	なし

***ESE**

このコマンドは、標準イベント・ステータス・イネーブル・レジスタのビットを設定します。設定によって、ステータス・バイト・レジスタのESB(イベント・サマリ・ビット)をセットする標準イベント・ステータス・イベント・レジスタのイベント(*ESR?を参照)を決定します。ビット位置の"1"は、対応するイベントをイネーブルにします。標準イベント・ステータス・イベント・レジスタのイネーブルになったイベントがすべて論理和され、ステータス・バイト・レジスタのイベント・サマリ・ビット(ESB)がセットされます。クウェリは、標準イベント・ステータス・イネーブル・レジスタを読み取ります。

表4-6. 標準イベント・ステータス・イネーブル・レジスタのビット構成

ビット位置	7	6	5	4	3	5	1	0
ビット名	PO	0	CME	EXE	DDE	QUE	0	OPC
ビットの重み	128	64	32	16	8	4	2	1
PON=電源がオンになりました。 CME=コマンド・エラー EXE=実行エラー				DDE=デバイス固有のエラー QUE=クウェリ・エラー OPC=動作完了				

コマンド構文 *ESE <NRf>
 パラメータ 0 ~ 255
 パワーオン値 (*PSCを参照)
 例 *ESE 129
 クウェリ構文 *ESE?
 戻りパラメータ <NR1> (レジスタ値)
 関連コマンド *ESR? *PSC *STB?

注記： *PSCが0に設定されていると、*ESEコマンドは不揮発性メモリにデータを書き込みます。不揮発性メモリはメモリ・サイズが決まっています。不揮発性メモリに繰り返し書き込みが生じるプログラムは最終的にはメモリの最大書き換え回数を超え、メモリ不良が発生します。

***ESR?**

このクウェリは、標準イベント・ステータス・イベント・レジスタを読み取ります。このレジスタは、読み取られるとクリアされます。ビット構成は、標準イベント・ステータス・イネーブル・レジスタと同じです(*ESEを参照)。

クウェリ構文 *ESR?
 パラメータ なし
 戻りパラメータ <NR1> (レジスタ2進値)
 関連コマンド *CLS *ESE *ESE *OPC

***OPC**

このコマンドを使用した場合、dcソースが保留中の処理をすべて完了すると、測定器は標準イベント・ステータス・レジスタのOPCビット(ビット0)をセットします。(標準イベント・ステータス・レジスタのビット構成については、*ESEを参照してください)。保留中の処理は、次のような場合に完了します。

- *OPCの前に送られたコマンドがすべて実行された場合。これには、重複するコマンドも含まれます。ほとんどのコマンドがシーケンシャルであり、次のコマンドの実行前に完了します。重複コマンドは、他のコマンドと並行して実行されます。出力電圧、電流またはステート、リレーおよびトリガ動作に影響を及ぼすコ

4 - 言語辞書

マンドは、dcソースに送られる後続のコマンドと重複します。*OPCコマンドは、重複するコマンドがすべて完了したことを知らせます。

- すべてのトリガ動作が完了した場合。

*OPCは、後続コマンドの処理を妨げませんが、保留中の処理がすべて完了するまでビット0はセットされません。

*OPC?を使用した場合、測定器は、保留中の処理がすべて完了すると、出力待ち行列にASCII "1"を入れます。*OPCと違って、*OPC?は後続のすべてのコマンドの処理を妨げます。アプリケーション・プログラムがdcソースの出力待ち行列から"1"を受け取るまでバスでデータをモニタできるように、*OPC?はコマンド行の終わりに使用します。

コマンド構文	*OPC
パラメータ	なし
クウェリ構文	*OPC?
戻りパラメータ	<NR1> 1
関連コマンド	*OPC *TRIG *WAI

*PSC

このコマンドは、サービス・リクエスト・イネーブル・レジスタおよび標準イベント・ステータス・イネーブル・レジスタの電源投入時の自動クリアを制御します。

***PSC ON | 1** 電源投入時にこれらのレジスタをクリアします。これによって、電源投入時にPONイベントによってSRQが生成されるのを防ぐことができます。

***PSC OFF | 0** 標準イベント・イネーブル・レジスタとサービス・リクエスト・イネーブル・レジスタの内容が不揮発性メモリにセーブされ、電源投入時にリコールされます。これによって、電源投入時にPONイベントによってSRQが生成できます。

クウェリは、*PSCのカレント・ステートを返します。

コマンド構文	*PSC <ブール>
パラメータ	0 1 OFF ON
例	*PSC 0 *PSC 1
クウェリ構文	*PSC?
戻りパラメータ	<NR1> 0 1
関連コマンド	*ESE *SRE

注記： *PSC は不揮発性メモリにデータを書き込みます。不揮発性メモリはメモリ・サイズが決まっています。不揮発性メモリに繰り返し書き込みを生じるプログラムは最終的にはメモリ・サイズをオーバーし、メモリ不良を発生させます。

*SRE

このコマンドは、サービス・リクエスト・イネーブル・レジスタの条件を設定します。このレジスタは、マスタ・ステータス・サマリ(MSS)ビットとサービス要求(RQS)サマリ・ビットをセットするステータス・バイト・レジスタのビット(ビット構成については*STBを参照)を決定します。サービス・リクエスト・イネーブル・レジスタのビット位置の1は、対応するステータス・バイト・レジスタのビットをイネーブルにします。すべてのイネーブルになったビットが論理和演算され、ステータス・バイト・レジスタのビット6がセットされます。

コントローラがSRQに応答してシリアル・ポーリングを実行した時には、RQSビットはクリアされますが、MSSビットはクリアされません。*SREがクリアされると(0に設定することによって)、dcソースはコントローラに対するSRQを生成することができません。

クウェリは、*SREのカレント・ステートを返します。

コマンド構文 *SRE <NRf>
 パラメータ 0~255
 電源投入時値 *PSCを参照
 例 *SRE 20
 クウェリ構文 *SRE?
 戻りパラメータ <NR1> (レジスタ2進値)
 関連コマンド *ESE *ESR *PSC

注記： *PSCが0に設定されていると、*SREコマンドは不揮発性メモリにデータを書き込みます。不揮発性メモリはメモリ・サイズが決まっています。不揮発性メモリに繰り返し書き込みが生じるプログラムは最終的にはメモリの最大書き換え回数を超え、メモリ不良が発生します。

***STB?**

このクウェリは、ステータス・バイト・レジスタを読み取ります。このレジスタには、ステータス・サマリ・ビットと出力待ち行列MAVビットが含まれています。ステータス・バイト・レジスタは、読み取ってもクリアされません。該当するイベント・レジスタが読み取られると、入力サマリ・ビットがクリアされます。MAVビットは、電源投入時に、*CLS*によってクリアされるか、使用可能な応答データがそれ以上ない場合にクリアされます。

シリアル・ポーリングは、ステータス・バイト・レジスタの値も返します。ただし、ビット6はマスタ・ステータス・サマリ(MSS)でなくサービス要求(RQS)を返します。シリアル・ポーリングは、RQSはクリアしますが、MSSはクリアしません。MSSがセットされている場合、dcソースがサービスを要求する理由が、1つ以上あることを示します。

表4-7. ステータス・バイト・レジスタのビット構成

ビット位置	7	6	5	4	3	2	1	0
ビット名	OPER	MSS(RQS)	ESB	MAV	QUES	0	0	0
ビットの重み	128	64	32	16	8	4	2	1
ESB=イベント・ステータス・バイト・サマリ MAV=使用可能なメッセージ MSS=マスタ・ステータス・サマリ				OPER=動作ステータス・サマリ QUES=クwestionナブル・ステータス・サマリ RQS=サービス要求				

クウェリ構文 *STB?
 戻りパラメータ <NR1> (レジスタ2進値)

***WAI**

このコマンドは、保留中の処理がすべて完了するまで別のコマンドの処理をしないようにdcソースに命令します。「保留中の処理」の定義については、*OPCコマンドで説明しています。*WAIは、dcソースにGPIB DCL(デバイス・クリア)コマンドを送信することによってのみクリアすることができます。

コマンド構文 WAI?
 パラメータ なし
 関連コマンド *OPC *OPC?

システム・コマンド

システム・コマンドには、システム・コマンドと表示コマンドがあります。

システム・コマンドは、出力制御機能や測定機能に直接関係のないシステム機能を制御します。

表示コマンドは、dcソースのフロント・パネルの表示を制御します。

共通コマンドもシステム機能を実行します。本項では、以下の共通コマンドについて説明します：

*IDN?、*OPT?、*RCL、*RST、*SAV、*TST?。

DISPlay

このコマンドは、フロント・パネル表示のオン/オフの切替えを行います。オフに設定すると、フロント・パネルには何も表示されません。表示アナウンシエータは、このコマンドによる影響を受けません。

コマンド構文	DISPlay[:WINDow][:STATe] <ブール>
パラメータ	0 1 OFF ON
*RST値	ON
例	DISP ON DISPLAY:STATE ON
クエリ構文	DISPlay[:WINDow][STATe]?
戻りパラメータ	<NR1> 0または1
関連コマンド	DISP:MODE DISP:TEXT *RST

DISPlay:MODE

通常の測定器機能とユーザが送ったテキストを表示するモードとの間で表示の切替えを行います。テキスト・メッセージは、DISPlay:TEXTコマンドで定義します。

コマンド構文	DISPlay[:WINDow]:MODE NORMal TEXT
パラメータ	<CRD> NORMal TEXT
*RST値	NORM
例	DISP:MODE NORM DISPLAY:MODE TEXT
クエリ構文	DISPlay[:WINDow]:MODE?
戻りパラメータ	<CRD> NORMALまたはTEXT
関連コマンド	DISP DISP:TEXT *RST

DISPlay:TEXT

このコマンドは、表示モードがTEXTに設定されている場合に、キャラクタ文字列をディスプレイに送信します。キャラクタ文字列は、大文字と小文字が区別されます。また、一重引用符(')か二重引用符(")で囲む必要があります。ディスプレイには、14文字まで表示することができます。文字列の15文字以降は切り捨てられます。

コマンド構文	DISPlay[:WINDow]:TEXT [:DATA] <表示文字列>
パラメータ	<表示文字列>
*RST値	ヌル文字列
例	DISP:TEXT "DEFAULT_MODE" DISPLAY:WINDOW:TEXT:DATA `533.2E-1VOLTS`
クエリ構文	DISPlay[:WINDow]:TEXT?
戻りパラメータ	<STR> (最後に設定されたテキスト文字列)
関連コマンド	DISP DISP:MODE

SYSTem:ERRor?

このクエリは、リモート・プログラミング・エラー待ち行列から、次のエラー番号と対応するエラー・メッセージ文字列を返します。待ち行列は、エラーが発生した時にストアするFIFO(先入れ先出し)バッファです。各エラーは、読み取られると、待ち行列から取り除かれます。すべてのエラーが読み取られると、クエリは0,NO ERRORを返します。待ち行列の保持能力を超えるエラーが蓄積された場合には、待ち行列の最後のエラーは-350,TOO MANY ERRORSになります(他のエラー・コードについては、付録Cを参照してください)。

dcソースのフロント・パネルにある Error キーを使用して、待ち行列のエラーを読み取ることができます。フロント・パネルで生成されたエラーは、待ち行列に入らず、即座にディスプレイ上に表示されます。

クエリ構文	SYSTem:ERRor?
パラメータ	(なし)
戻りパラメータ	<NR1>, <SRD>
例	SYST:ERR? SYSTEM:ERROR?

SYSTem:LANGuage

このコマンドは、測定器のSCPIコマンド言語と互換言語の切替えを行います。互換言語は、旧モデルのdcソース・システムをエミュレーションするために提供されています。互換言語については、付録Bを参照してください。このコマンドを送信すると、次の動作が行われます。

- 選択された言語がアクティブになり、不揮発性メモリにストアされます。
- dcソースがパワーオン・ステートにリセットされます。

dcソースの電源が遮断した場合、電源が回復すると、最後に選択されていた言語で処理が再開されます。このコマンドとクエリは、現在選択されている言語に関係なく使用することができます。

コマンド構文	SYSTem:LANGuage<文字列>
パラメータ	SCPI COMPatibility
パワーオン値	最後に選択された言語
例	SYST:LANG SCPI SYSTEM:LANGUAGE COMPATIBILITY
クエリ構文	SYSTem:LANGuage?
戻りパラメータ	<CRD>

SYSTem:VERSion?

このクエリは、dcソースが準拠しているSCPIバージョン番号を返します。戻り値は、YYYY.Vという形式をとります。ここで、YYYYは年を、Vはその年のリビジョン番号をそれぞれ表します。

クエリ構文	SYSTem:VERSion?
パラメータ	(なし)
戻りパラメータ	<NR2>
例	SYST:VERS? SYSTEM:VERSION?

4 - 言語辞書

SYSTem:LOCal

RS-232動作の場合のみ

このコマンドは、RS-232動作中に、dcソースをローカル・モードにします。フロント・パネルのキーを使用することができます。

コマンド構文	SYSTem:LOCal
パラメータ	なし
例	SYST:LOC
関連コマンド	SYST:REM SYST:RWL

SYSTem:REMOte

RS-232動作の場合のみ

このコマンドは、RS-232動作中に、dcソースをリモート・モードにします。これによって、Localキーを除くすべてのフロント・パネル・キーがディスエーブルになります。リモート・ステートの時にLocalキーを押すと、フロント・パネルがローカル・ステートに戻ります。

コマンド構文	SYSTem:REMOte
パラメータ	なし
例	SYST:REM
関連コマンド	SYST:LOC SYST:RWL

SYSTem:RWLock

RS-232動作の場合のみ

このコマンドは、RS-232動作中に、dcソースをリモート・モードにします。Localキーを含めたすべてのフロント・パネル・キーがディスエーブルになります。SYSTem:LOCalを使用して、フロント・パネルをローカル・ステートに戻してください。

コマンド構文	SYSTem:RWLock
パラメータ	なし
例	SYST:RWL
関連コマンド	SYST:REM SYST:LOC

*IDN?

このクエリは、dcソースに自己を識別するように要求します。また、カンマによって区切られた4つのフィールドから成る文字列を返します。

クエリ構文	*IDN?		
戻りパラメータ	<AARD>	フィールド	情報
		Agilent Technologies	メーカー
		xxxxxA	モデル番号と1文字の接尾語
		nnnnA-nnnnn	10文字から成るシリアル番号または0
		<A>.xx.xx	ファームウェアのリビジョン・レベル
例	AGILENT,66312A,0,A.00.01		

***OPT?**

このコマンドは、dcソースにインストールされているオプションをすべて識別するように要求します。オプションは番号Aによって識別されます。0はオプションがインストールされていないことを示します。

クウェリ構文 *OPT?
戻りパラメータ <AARD>

***RCL**

警告： 以前にストアしたステートをリコールすると、dcソース出力に危険な電圧が生じる可能性があります。

このコマンドは、dcソースを、以前に*SAVコマンドでメモリ内の指定した場所にストアしたステータに復元します。以下の場合を除き、すべてのステータがリコールされます。

- トリガ・システムは、黙示のABORtコマンドによってアイドル・ステータに設定されます(これによって、完了していないトリガ動作はすべてキャンセルされます)。
- 校正機能は、CAL:STATeをOFFに設定することによりディスエーブルになります。

注記： 0位置にストアされているデバイス・ステータは、OUTPut:PON:STATeがRCL0に設定されている場合、電源投入時に自動的にリコールされます。

コマンド構文 *RCL <NRf>
パラメータ 0 | 1 | 2 | 3
例 *RCL 3
関連コマンド *PSC *RST *SAV

***RST**

このコマンドは、dcソースを以下に定義する出荷時のステータにリセットします。*RSTもまたABORtコマンドを強制的に実行します。

コマンド構文 *RST
パラメータ なし
関連コマンド *PSC *SAV

表4-8. *RST設定

CAL:STAT	OFF	[SOUR:]CURR	MAXの10%*
DIG:DATA	0	[SOUR:]CURR:TRIG	MAXの10%*
DISP:STAT	ON	[SOUR:]CURR:PROT:STAT	OFF
DISP:MODE	NORM	[SOUR:]LIST:COUN	0
DISP:TEXT	‘ ’	[SOUR:]VOLT	0
INIT:CONT	OFF	[SOUR:]VOLT:TRIG	0
OUTP	OFF	[SOUR:]VOLT:PROT	MAX*
OUTP:DFI	OFF	TRIG:ACQ:COUN:CURR	1
OUTP:DFI:SOUR	OFF	TRIG:ACQ:COUN:VOLT	1
OUTP:PROT:DEL	.08 Norm; .008 Fast	TRIG:ACQ:HYST:CURR	0
OUTP:REL	OFF	TRIG:ACQ:HYST:VOLT	0
OUTP:REL:POL	NORM	TRIG:ACQ:LEV:CURR	MAX*
SENS:CURR:RANG	MAX	TRIG:ACQ:LEV:VOLT	MAX*
SENS:CURR:DET	ACDC	TRIG:ACQ:SLOP:CURR	POS
SENS:FUNC	VOLT	TRIG:ACQ:SLOP:VOLT	POS
SENS:SWE:OFFS:POIN	0	TRIG:ACQ:SOUR	INTERNAL
SENS:SWE:POIN	2048	TRIG:TRAN:SOUR	BUS
SENS:SWE:TINT	15.6μs		

* 最大値はモデルによって異なります。表4-3を参照してください。

*SAV

このコマンドは、dcソースの現在のステートを不揮発性メモリの指定された場所にストアします。最高4つのステートをストアすることができます。電源投入時にある特定のステートに設定したい場合、そのステートを0位置にストアします。OUTPut:PON:STATeがRCL0に設定されていれば、電源投入時にこのステートが自動的にリコールされません。*RCLは、機器ステートを検索します。

```

コマンド構文  *SAV <NRf>
  パラメータ  0 | 1 | 2 | 3
  例          *SAV 3
  関連コマンド *RCL *RST

```

注記： *SAV は不揮発性メモリにデータを書き込みます。不揮発性メモリはメモリ・サイズが決まっています。不揮発性メモリに繰り返し書き込みが生じるプログラムは最終的にはメモリの最大書き換え回数を超え、メモリ不良が発生します。

*TST?

このクウェリは、dcソースにセルフテストを実行させ、エラーをすべて報告させます。0は、dcソースがセルフテストに合格したことを示します。1は、1つ以上のテストが失敗したことを示します。セルフテスト・エラーは、エラー待ち行列に書き込まれます(付録Cを参照)。

```

クウェリ構文  *TST?
戻りパラメータ <NR1>

```

トリガ・コマンド

トリガ・コマンドは、トリガ・コマンドと起動コマンドから成ります。

トリガ・コマンドは、dcソースのリモート・トリガを制御します。トリガ・コマンド(および起動コマンド)は、名前か番号によって参照されます。名前と番号の対応関係を以下に示します。

シーケンス番号	シーケンス名	内容
1(デフォルト)	TRANsient	出力過渡トリガ・シーケンス
2	ACQuire	測定捕捉トリガ・シーケンス

起動コマンドは、トリガ・システムを起動します。

ABORt

このコマンドは、処理中のトリガ動作をすべてキャンセルします。保留中のトリガ・レベルは、対応する即値にリセットされます。ABORtは、動作条件ステータス・レジスタのWTGビットもリセットします(第3章の「ステータス・レジスタの設定」の項を参照してください)。INITiate:CONTInuous ONが設定されると、トリガ・サブシステムはABORt後に即座に動作を開始し、WTGが設定されます。ABORtは、電源投入時に*RCLまたはRSTを実行すると実行されます。

コマンド構文	ABORt
パラメータ	なし
例	ABOR
関連コマンド	INIT *RST *TRG TRIG

INITiate:SEQuence

INITiate:NAME

INIT:SEQ2またはINIT:NAME ACQはAgilent 66312A, 66332Aにのみ適用

INITiateコマンドは、出力トリガと測定トリガの両方の開始を制御します。トリガがイネーブルになると、選択されたトリガ源のイベントによって、指定されたトリガ動作が発生します。トリガ・サブシステムがイネーブルになっていないと、トリガ・コマンドはすべて無視されます。

コマンド構文	INITiate[:IMMediate]:SEQuence[1 2] INITiate[:IMMediate]:NAME<名前>
パラメータ	INIT:NAME の場合 TRANsient ACQuire
例	INIT:SEQ2 INIT:NAME TRAN
関連コマンド	ABOR INIT:CONT TRIG TRIG:SEQ:DEF *TRG

INITiate:CONTInuous:SEQuence1

INITiate:CONTInuous:NAME

これらのコマンドは、出力過渡トリガ・システムを制御します。

1または**ON**は、出力トリガ・システムを連続的に起動します。

0または**OFF**は、連続トリガをオフにします。このステートでは、INITiate:SEQuenceを使用して、出力トリガ・システムをトリガごとに起動する必要があります。

コマンド構文	INITiate:CONTInuous:SEQuence1<ブール> INITiate:CONTInuous:NAME TRANsient,<ブール>
--------	--

4 - 言語辞書

パラメータ	0 1 OFF ON
*RST値	OFF
例	INIT:CONT:SEQ ON INIT:CONT:NAME TRAN, 1
関連コマンド	ABOR INIT TRIG TRIG:SEQ:DEF *TRG

TRIGger

トリガ・サブシステムが起動されると、このコマンドは出力トリガ信号を生成します。このトリガによって、以下のことが起こります。

1. 保留中のレベル変更をCURRent:TRIGgerまたはVOLTage:TRIGgerの指定に従って開始します。
2. 過渡トリガ・シーケンスと捕捉トリガ・シーケンスの両方が完了したら、ステータス動作条件レジスタのWTGビットをクリアします(WTGは、過渡シーケンスと捕捉シーケンスの両方の論理和です)。
3. INITiate:CONTinuous ONが設定されている場合、トリガ・サブシステムが後続のトリガに対して即座に再イネーブルになります。クリアされるとすぐに、WTGビットは再度1に設定されます。

コマンド構文	TRIGger[:SEquence1][:IMMEDIATE] TRIGger[:TRANSient][:IMMEDIATE]
パラメータ	なし
例	TRIG TRIG:IMM
関連コマンド	ABOR CURR:TRIG INIT *TRG VOLT:TRIG

TRIGger:SOURce

このコマンドは、以下のように過渡トリガのトリガ源を選択します。

BUS **GPIO**デバイス、*TRGまたは<GET>(グループ実行トリガ)

コマンド構文	TRIGger[:SEquence1]:SOURce<ソース> TRIGger[:TRANSient]:SOURce<ソース>
パラメータ	BUS
*RST値	BUS
例	TRIG:SOUR BUS
クエリ構文	TRIGger[:SEquence1]:SOURce? TRIGger[:TRANSient]:SOURce?
戻りパラメータ	<CRD>

TRIGger:SEquence2

TRIGger:ACQuire

Agilent 66312A, 66332Aの場合のみ

トリガ・サブシステムが起動されると、これらのコマンドは測定トリガ信号を生成します。この測定トリガによって、dcソースは出力電圧と電流を測定し、その結果をバッファにストアします。

コマンド構文	TRIGger:SEquence2[:IMMEDIATE] TRIGger:ACQuire[:IMMEDIATE]
パラメータ	なし
例	TRIG:SEQ2 TRIG:ACQ
関連コマンド	TRIG:SOUR TRIG:SEQ2:DEF TRIG:SEQ2:COUN TRIG:SEQ2:LEV:VOLT TRIG:SEQ2:SLOP:CURR

TRIGger:SEQuence2:COUnT:CURRent
 TRIGger:ACQuire:COUnT:CURRent

Agilent 66312A, 66332Aの場合のみ

このコマンドは、電流データを測定する際のトリガの連番を設定します。このコマンドを使用する場合には、収集期間の始めに1度だけトリガ・システムを初期化する必要があります。1つの測定が完了すると、測定器は、新たにトリガ条件が有効になるのを待って次の測定を開始します。これは、指定カウントに達するまで続けられます。

コマンド構文	TRIGger:SEQuence2:COUnT:CURRent<NRf> TRIGger:ACQuire:COUnT:CURRent<NRf+>
パラメータ	1~100
*RST値	1
例	TRIG:SEQ2:COUnT:CURR 5 TRIG:ACQ:COUnT:CURR 1
クウェリ構文	TRIGger:SEQuence2:COUnT:CURRent? TRIGger:ACQuire:COUnT:CURRent?
戻りパラメータ	<NR3>
関連コマンド	TRIG:SEQ2 TRIG:ACQ

TRIGger:SEQuence2:COUnT:VOLTagE
 TRIGger:ACQuire:COUnT:VOLTagE

Agilent 66312A, 66332Aの場合のみ

このコマンドは、電圧データを測定する際のトリガの連番を設定します。このコマンドを使用する場合には、収集期間の始めに1度だけトリガ・システムを初期化する必要があります。1つの測定が完了すると、測定器は、新たにトリガ条件が有効になるのを待って次の測定を開始します。これは、指定カウントに達するまで続けられます。

コマンド構文	TRIGger:SEQuence2:COUnT:VOLTagE<NRf+> TRIGger:ACQuire:COUnT:VOLTagE<NRf+>
パラメータ	1~100
*RST値	1
例	TRIG:SEQ2:COUnT:VOLT 5 TRIG:ACQ:COUnT:VOLT 1
クウェリ構文	TRIGger:SEQuence2:COUnT:VOLTagE? TRIGger:ACQuire:COUnT:VOLTagE?
戻りパラメータ	<NR3>
関連コマンド	TRIG:SEQ2 TRIG:ACQ

TRIGger:SEQuence2:HYSTeresis:CURRent
 TRIGger:ACQuire:HYSTeresis:CURRent

Agilent 66312A, 66332Aの場合のみ

このコマンドは、内部測定が発生する前に信号が通過しなければならないトリガ・レベル付近の帯域を定義します。トリガ・レベルの上と下にある帯域限界値は、トリガ・レベルにヒステリシス値の1/2を加えた値か、トリガ・レベルからヒステリシス値の1/2を引いた値になります。

立上がりトリガが発生させるためには、出力波形の正方向への偏位がヒステリシス・バンドの下限値の下から始まり、ヒステリシス・バンドの上限値を通過しなければなりません。一方、立下がりトリガが発生させるためには、出力波形の負方向への偏位がヒステリシス・バンドの上限値の上から始まり、ヒステリシス・バンドの下限値を通過しなければなりません。

コマンド構文	TRIGger:SEQuence2:HYSTeresis:CURRent<NRf+> TRIGger:ACQuire:HYSTeresis:CURRent<NRf+>
パラメータ	0~MAX(表4-3を参照)
単位	A(アンペア)
*RST値	0
例	TRIG:SEQ2:HYST:CURR 0.5 TRIG:ACQ:HYST:CURR 0.5
クエリ構文	TRIGger:SEQuence2:HYSTeresis:CURRent? TRIGger:ACQuire:HYSTeresis:CURRent?
戻りパラメータ	<NR3>
関連コマンド	TRIG:SEQ2:HYST:VOLT TRIG:SEQ2:LEV:CURR

TRIGger:SEQuence2:HYSTeresis:VOLTage
 TRIGger:ACQuire:HYSTeresis:VOLTage

Agilent 66312A, 66332Aの場合のみ

このコマンドは、内部測定が発生する前に信号が通過しなければならないトリガ・レベル付近の帯域を定義します。トリガ・レベルの上と下にある帯域限界値は、トリガ・レベルにヒステリシス値の1/2を加えた値か、トリガ・レベルからヒステリシス値の1/2を引いた値になります。

立上がりトリガが発生させるためには、出力波形の正方向への偏位がヒステリシス・バンドの下限値の下から始まり、ヒステリシス・バンドの上限値を通過しなければなりません。一方、立下がりトリガが発生させるためには、出力波形の負方向への偏位がヒステリシス・バンドの上限値の上から始まり、ヒステリシス・バンドの下限値を通過しなければなりません。

コマンド構文	TRIGger:SEQuence2:HYSTeresis:VOLTage<NRf+> TRIGger:ACQuire:HYSTeresis:VOLTage<NRf+>
パラメータ	0~MAX(表4-3を参照)
単位	V(ボルト)
*RST値	0
例	TRIG:SEQ2:HYST:VOLT 2 TRIG:ACQ:HYST:VOLT 2
クエリ構文	TRIGger:SEQuence2:HYSTeresis:VOLTage? TRIGger:ACQuire:HYSTeresis:VOLTage?
戻りパラメータ	<NR3>
関連コマンド	TRIG:SEQ2:HYST:CURR TRIG:SEQ2:LEV:VOLT

TRIGger:SEQuence2:LEVel:CURRent
 TRIGger:ACQuire:LEVel:CURRent

Agilent 66312A, 66332Aの場合のみ

このコマンドは、内部トリガ電流測定トリガ・レベルを設定します。立上がり電流トリガは、電流レベルがヒステリシス・バンドの下限値より小さい値からヒステリシス・バンドの上限値より大きい値に変動した場合に発生します。同様に、立下がり電流トリガは、電流レベルがヒステリシス・バンドの上限値より大きい値からヒステリシス・バンドの下限値より小さい値に変動した場合に発生します。

コマンド構文	TRIGger:SEQuence2:LEVel:CURRent<NRf+> TRIGger:ACQuire:LEVel:CURRent<NRf+>
パラメータ	0~MAX(表4-3を参照)
単位	A(アンペア)
*RST値	0
例	TRIG:SEQ2:LEV:CURR 5 TRIG:ACQ:LEV:CURR MAX TRIG:ACQ:LEV 2
クウェリ構文	TRIGger:SEQuence2:LEVel:CURRent? TRIGger:ACQuire:LEVel:CURRent?
戻りパラメータ	<NR3>
関連コマンド	TRIG:SEQ2:LEV:VOLT TRIG:SEQ2:HYST:CURR

TRIGger:SEQuence2:LEVel:VOLTagE
 TRIGger:ACQuire:LEVel:VOLTagE

Agilent 66312A, 66332Aの場合のみ

このコマンドは、内部トリガ電圧測定トリガ・レベルを設定します。立上がり電圧トリガは、電圧レベルがヒステリシス・バンドの下限値より小さい値からヒステリシス・バンドの上限値より大きい値に変動した場合に発生します。同様に、立下がり電圧トリガは、電圧レベルがヒステリシス・バンドの上限値より大きい値からヒステリシス・バンドの下限値より小さい値に変動した場合に発生します。

コマンド構文	TRIGger:SEQuence2:LEVel:VOLTagE<NRf+> TRIGger:ACQuire:LEVel:VOLTagE<NRf+>
パラメータ	0~MAX(表4-3を参照)
単位	V(ボルト)
*RST値	0
例	TRIG:SEQ2:LEV:VOLT 5 TRIG:ACQ:LEV:VOLT MAX TRIG:ACQ:LEV 2
クウェリ構文	TRIGger:SEQuence2:LEVel:VOLTagE? TRIGger:ACQuire:LEVel:VOLTagE?
戻りパラメータ	<NR3>
関連コマンド	TRIG:SEQ2:LEV:CURR TRIG:SEQ2:HYST:VOLT

4 - 言語辞書

TRIGger:SEquence2:SLOPe:CURRent
TRIGger:ACQuire:SLOPe:CURRent

Agilent 66312A, 66332Aの場合のみ

このコマンドは、内部トリガ電流測定のスロープを設定します。

POSitive トリガが立上がり端で発生します。

NEGative トリガが立下がり端で発生します。

EITHer トリガがどちらかのエッジで発生します。

コマンド構文	TRIGger:SEquence2:SLOPe:CURRent<スロープ> TRIGger:ACQuire:SLOPe:CURRent<スロープ>
パラメータ	EITHer POSitive NEGative
*RST値	EITHer
例	TRIG:SEQ2:SLOP:CURR POS TRIG:ACQ:SLOP:CURR EITH
クウェリ構文	TRIGger:SEquence2:SLOPe:CURRent? TRIGger:ACQuire:SLOPe:CURRent?
戻りパラメータ	<CRD>
関連コマンド	TRIG:SEQ2:SLOP:VOLT

TRIGger:SEquence2:SLOPe:VOLTage
TRIGger:ACQuire:SLOPe:VOLTage

Agilent 66312A, 66332Aの場合のみ

このコマンドは、内部トリガ電圧測定のスロープを設定します。

POSitive トリガは立上がり端で発生します。

NEGative トリガは立下がり端で発生します。

EITHer トリガはどちらかのエッジで発生します。

コマンド構文	TRIGger:SEquence2:SLOPe:VOLTage<スロープ> TRIGger:ACQuire:SLOPe:VOLTage<スロープ>
パラメータ	EITHer POSitive NEGative
*RST値	EITHer
例	TRIG:SEQ2:SLOP:VOLT POS TRIG:ACQ:SLOP:VOLT EITH
クウェリ構文	TRIGger:SEquence2:SLOPe:VOLTage? TRIGger:ACQuire:SLOPe:VOLTage?
戻りパラメータ	<CRD>
関連コマンド	TRIG:SEQ2:SLOP:CURR

TRIGger:SEQuence2:SOURce
 TRIGger:ACQuire:SOURce

Agilent 66312A, 66332Aの場合のみ

これらのコマンドは、以下のように測定トリガのトリガ源を選択します。

BUS GPIBデバイス、*TRGまたは<GET>(グループ実行トリガ)

INTernal 測定波形が選択されたスロープでトリガ・レベルをクロスした場合に、トリガが内部的に生成されます。

コマンド構文	TRIGger:SEQuence2:SOURce<ソース> TRIGger:ACQuire:SOURce<ソース>
パラメータ	BUS INTernal
*RST値	INTernal
例	TRIG:ACQ:SOUR BUS
クウェリ構文	TRIGger:SEQuence2:SOURce? TRIGger:ACQuire:SOURce?
戻りパラメータ	<CRD>

TRIGger:SEQuence1:DEFine
 TRIGger:SEQuence2:DEFine

TRIGger:SEQuence2:DEFineはAgilent 66312A, 66332Aにのみ適用

これらのコマンドは、トリガ・シーケンス1と2に付ける別名を定義します。コマンドは、定義済みの名前として、ACQuire(シーケンス2の場合)とTRANSient(シーケンス1の場合)だけを受け付けます。クウェリを使用することによって、ユーザはシーケンス1および2の別名の測定器名を問い合わせることができます。

コマンド構文	TRIGger:SEQuence1:DEFine TRANSient TRIGger:SEQuence2:DEFine ACQuire
パラメータ	TRANSient, ACQuire
例	SEQ1:DEF ACQ SEQ2:DEF TRAN
クウェリ構文	TRIGger:SEQuence1:DEFine? TRIGger:SEQuence2:DEFine?
戻りパラメータ	<CRD>
関連コマンド	TRIG:SEQ2:ACQ TRIG:SEQ1:TRAN

***TRG**

この共通コマンドは、トリガ・サブシステムがトリガ源として選択されたバスを持っている場合にトリガを発生させます。このコマンドは、グループ実行トリガ(<GET>)コマンドと同じ効果を持ちます。

RS-232モードでは、このコマンドはIEEE-488グループ実行コマンドの機能の一部をエミュレーションします。

コマンド構文	*TRG
パラメータ	なし
関連コマンド	ABOR INIT TRIG[:IMM] <GET>

SCPIとの適合性について

SCPIバージョン

Agilentダイナミック測定dcソースは、SCPIバージョン1995.0に準拠しています。

SCPI準拠コマンド

ABOR	SENS:SWE:POIN
CAL:DATA	SENS:SWE:TINT
CAL:STAT	STAT:OPER[:EVEN]?
DISP[:WIND][:STAT]	STAT:OPER:COND?
DISP[:WIND]:TEXT[:DATA]	STAT:OPER:ENAB
INIT[:IMM]:SEQ NAME	STAT:OPER:NTR
INIT:CONT:SEQ NAME	STAT:OPER:PTR
MEAS FETC:ARR:CURR[:DC]?	STAT:PRES
MEAS FETC:ARR:VOLT[:DC]?	STAT:QUES[:EVEN]?
MEAS FETC:SCAL:CURR[:DC]?	STAT:QUES:COND?
MEAS FETC:SCAL:CURR:HIGH?	STAT:QUES:ENAB
MEAS FETC:SCAL:CURR:LOW?	STAT:QUES:NTR
MEAS FETC:SCAL:CURR:MAX?	STAT:QUES:PTR
MEAS FETC:SCAL:CURR:MIN?	SYST:ERR?
MEAS FETC:SCAL:VOLT[:DC]?	SYST:LANG
MEAS FETC:SCAL:VOLT:HIGH?	SYST:VERS?
MEAS FETC:SCAL:VOLT:LOW?	TRIG[:SEQ1 :TRAN][:IMM]
MEAS FETC:SCAL:VOLT:MAX?	TRIG[:SEQ1 :TRAN]:SOUR
MEAS FETC:SCAL:VOLT:MIN?	TRIG:SEQ2 ACQ[:IMM]
OUTP[:STAT]	TRIG:SEQ2 ACQ:SOUR
OUTP:PROT:CLE	TRIG:SEQ:DEF
OUTP:PROT:DEL	*CLS
[SOUR]:CURR[:LEV][:IMM][:AMPL]	*ESE *ESE? *ESR?
[SOUR]:CURR[:LEV]:TRIG[:AMPL]	*IDN?
[SOUR]:CURR:PROT:STAT	*OPC *OPC? *OPT?
[SOUR]:VOLT[:LEV][:IMM][:AMPL]	*PSC *PSC?
[SOUR]:VOLT[:LEV]:TRIG[:AMPL]	*RCL *RST
[SOUR]:VOLT:PROT	*SAV *SRE *STB?
SENS:CURR[:DC]:RANG[:UPP]	*TRG *TST?
SENS:FUNC	*WAI
SENS:SWE:OFFS:POIN	

A - SCPIとの適合性について

非SCPIコマンド

CAL:CURRE[:SOUR][:DC][:POS]	OUTP:DFI:SOUR
CAL:CURRE[:SOUR][:DC]:NEG	OUTP:PON:STAT
CAL:MEAS[:DC]:LOWR	OUTP:REL[:STAT]
CAL:MEAS:AC	OUTP:REL:POL
CAL:LEV	OUTP:RI:MODE
CAL:PASS	SENS:CURRE:DET
CAL:SAVE	[SOUR]:DIG:DATA[:VAL]
CAL:VOLT[:DC]	[SOUR]:DIG:FUNC
CAL:VOLT:PROT	TRIG:SEQ2 ACQ:COUN:CURRE :VOLT
DISP[:WIND]:MODE	TRIG:SEQ2 ACQ:HYST:CURRE :VOLT
MEAS FETC[:SCAL]:CURRE:ACDC?	TRIG:SEQ2 ACQ:LEV:CURRE :VOLT
MEAS FETC[:SCAL]:VOLT:ACDC?	TRIG:SEQ2 ACQ:SLOP:CURRE :VOLT
OUTP:DFI[:STAT]	

互換言語

概要

本書で取り扱っているAgilent電源は、Agilent 6632A、6633Aおよび6634A DC電源とプログラムの互換性があります。つまり、 GPIBを介しCOMPatibility言語モードで、COMPatibilityコマンドを使って、これらの新しいdcソースを設定できます。

SCPI コマンドからCOMPatibility コマンド(あるいはその反対)に切り替えるには、第4章に説明したように、SYST:LANGコマンドを使用します。言語設定は不揮発性メモリにセーブされます。

表B-2に、電源の設定に使用するCOMPatibilityコマンドの概要を示します。COMPatibilityコマンドの詳細については、『Agilent Series 6632, 6633A, and 6634A Operating Guide』(Agilentマニュアル番号5957-6360)を参照してください。

本付録の残りの部分では、COMPatibility言語ステータス・システムとCOMPatibility言語のエラー・コードについて説明します。

注記： Compatibilityプログラミング言語について詳しくお知りになりたい場合は、『Agilent 6632A/6633A/6634A Operating manual』(Agilentマニュアル番号5957-6360)を購入してください。

表B-1. COMPatibilityの電源投入時の設定

コマンド	設定	コマンド	設定
DC	1(ON)	POL	1(ノーマル)
DLY	8 ms(高速) 80 ms(ノーマル)	PON	最後にストアされた値
DSP	1(ON)	RELAY	1(クローズ)
ISSET	0.04 A(6631B) 0.02 A(6632B) 0.008 A(6633B) 0.004 A(6634B)	RLYPO	1(クローズ)
OCP	OFF	SRQ	0
OUT	1(ON)	UNMASK	0
OVSET	MAX	VSET	0 V

表B-2. COMPAtibilityコマンド

互換コマンド	内容	類似するSCPIコマンド
ASTS?	このコマンドは、累積ステータス・レジスタの内容を読み取ります。このレジスタには、累積ステータス・レジスタの最後の読取り以降にステータス・レジスタに入れられたビット条件が、その条件がなおも存在しているか否かに関係なくストアされています。データ表記: ZZZZD	STAT:OPER?、 STAT:QUES?、 *ESE?
CLR	このコマンドは、dcソースをパワーオン・ステートに初期設定します。また、シリアル・ポール・レジスタのPONビットをリセットします。このコマンドは、デバイス・クリア(DCL)インタフェース・メッセージと同じ機能を実行します。	*RST
DC 0 1	オプション760が装備された装置にのみ適用されます。このコマンドは、出力リレーのステートに影響することなく、出力をイネーブル/ディスエーブルにします。初期条件: DC 1	OUTP:STAT[:NOR] 0 1 OFF ON
DLY <n>	このコマンドは、CV, CCあるいはUNREG条件を生成する出力変動の設定と、ステータス・レジスタによる条件の記録との間の遅延時間を設定します。このコマンドは、過電流保護機能(OCP)の間違ったトリガを防ぐために使用することができます。 初期遅延: 0.08s(ノーマル); 0.008s(高速)	OUTP:PROT:DEL
DSP 0 1	このコマンドは、dcソースのフロント・パネル表示をイネーブル/ディスエーブルにします。初期条件: DSP 1	DISP 0 1 OFF ON
ERR?	このコマンドは、dcソースによって検出されるプログラミング・エラーのタイプを決定します。リモート・プログラミング・エラーは、ステータス・レジスタにERRビットをセットします。このビットは、UNMASKによってイネーブルにし、サービスを要求することができます。	SYST:ERR?
FAULT?	このコマンドは、フォルト・レジスタにどのビットがセットされているかを読み取ります。ビットは、ステータス・レジスタ内の対応するビットが非アクティブからアクティブに変わり、マスク・レジスタ内の対応するビットがイネーブルになっている場合にだけ、フォルト・レジスタにセットされます。フォルト・レジスタは、読取り後にのみリセットされます。全イネーブル・ビットの総ビット重みの等価10進値が返されます。データ表記: ZZZZD	STAT:OPER?、 STAT:QUES?、 *ESE
ID?	このコマンドは、dcソースにモデル番号と、dcソースの出力に影響を及ぼすオプションを報告させます。データ表記: Agilent663XA	*IDN?
IOUT?	このコマンドは、実出力電流を測定し、その値を返します。 データ表記: SD.DDDD	MEAS:CURR?
ISSET <n>	このコマンドは、出力電流を設定します。このコマンドの設定レンジについては、表4-3を参照してください。初期条件: 表B-1を参照。	CURR
OCP 0 1	このコマンドは、dcソースの過電流保護をイネーブル/ディスエーブルにします。この機能がイネーブルになり、dcソースがCCモードに入ると、dcソースの出力がディスエーブルになります。初期条件: OCP 0	CURR:PROT:STAT 0 1 OFF ON
OUT 0 1	このコマンドは、dcソースの出力をイネーブル/ディスエーブルにします。dcソースは、出力がディスエーブル状態にある間も、コマンドを実行できるようになります。初期条件: OUT 1	OUTP:STAT 0 1 OFF ON

表B-2. COMPatibilityコマンド (続き)

互換コマンド	内容	類似するSCPIコマンド
OVSET <n>	このコマンドは、過電圧保護を設定します。このコマンドの設定レンジについては、表4-3を参照してください。初期条件: MAX	VOLT:PROT
POL 0 1	オプション760が装備された装置にのみ適用されます。このコマンドは、出力リレーのポラリティをノーマル(1)か反転(0)のいずれかに設定します。初期条件: POL 1	OUTP:REL:POL 0 1
PON 0 1	このコマンドは、電源投入時にSRQをイネーブル(1)またはディスエーブル(0)にします。初期条件: 最後に設定された値	PSC 0 1
RELAY 0 1	オプション760が装備された装置にのみ適用されます。このコマンドは、装置の設定された出力ステートに影響を及ぼすことなく、出力リレーをオープン(0)/クローズ(1)します。初期条件: RELAY 1	OUTP:REL 0 1
RLYPON 0 1	オプション760が装備された装置にのみ適用されます。このコマンドは、装置の設定された出力ステートに影響を及ぼすことなく、電源投入時に出力リレーをオープン(0)/クローズ(1)します。初期条件: RLYPON 1	RCL 0
ROM?	このコマンドは、ROMバージョンのdcソースを返します。 データ表記: AAA AAA	*IDN?
RST	このコマンドは、出力が出力保護回路によってディスエーブルにされている場合に、dcソースをリセットします。	OUTP:PROT:CLE
SENS:CURRE:RANG <n>	このコマンドは、dcソースの電流測定レンジを設定します。このコマンドの設定レンジについては、表4-3を参照してください。初期条件: MAX	SENS:CURRE:RANG
SENS:SWE:POIN <n>	このコマンドは、測定掃引のデータ・ポイント数を定義します。 初期条件: 32	SENS:SWE:POIN
SENS:SWE:TINT <n>	このコマンドは、測定サンプル間の周期を定義します。初期条件: 15.6μs	SENS:SWE:TINT
SRQ 0 1	これらのコマンドは、フォルト条件に対しコントローラからサービスを要求するdcソースの機能をイネーブル/ディスエーブルにします。UNMASKは、フォルトとして定義する条件を定義します。 初期条件: SRQ 0	*SRE
STS?	このコマンドは、ステータス・レジスタの内容を読み取ります。ステータス・レジスタには、dcソースの現在のステータスが保存されています。 データ表記: ZZZZD	STAT:OPER:COND?、 STAT:QUES:COND?、 *ESE?
SYST:LANG	このコマンドは、代替言語をアクティブにし、不揮発性メモリにストアします。この場合、コマンドは等価です。dcソースの電源が遮断された場合、dcソースは電源が回復すると、最後に選択された言語で再開します。COMPかSCPIのどちらかのパラメータを指定します。	SYST:LANG
TEST?	このコマンドは、dcソースにセルフテストを実行させ、検出された障害をすべて報告させます。データ表記: ZZZZD	*TST?

表B-2. COMPatibilityコマンド (続き)

互換コマンド	内容	類似するSCPIコマンド
UNMASK xxx	これらのコマンドは、フォルト・レジスタにビットをセットする条件を決定します。これによって、オペレータはフォルトとして報告される条件を定義することができます。フォルト条件は、全イネーブル条件の総ビット重みの等価10進値を送信することにより、イネーブルにすることができます。	STAT:OPER:ENAB、 STAT:QUES:ENAB、 *ESE
VOUT?	このコマンドは、実出力電圧を測定し、返します。 データ表記: SZZD.DD;(SZZD.DDDは6634Bの場合のみ)	MEAS:VOLT?
VSET <n>	このコマンドは、出力電圧を設定します。このコマンドの設定レンジについては、表4-3を参照してください。初期条件: 0 V	VOLT

A アルファベット

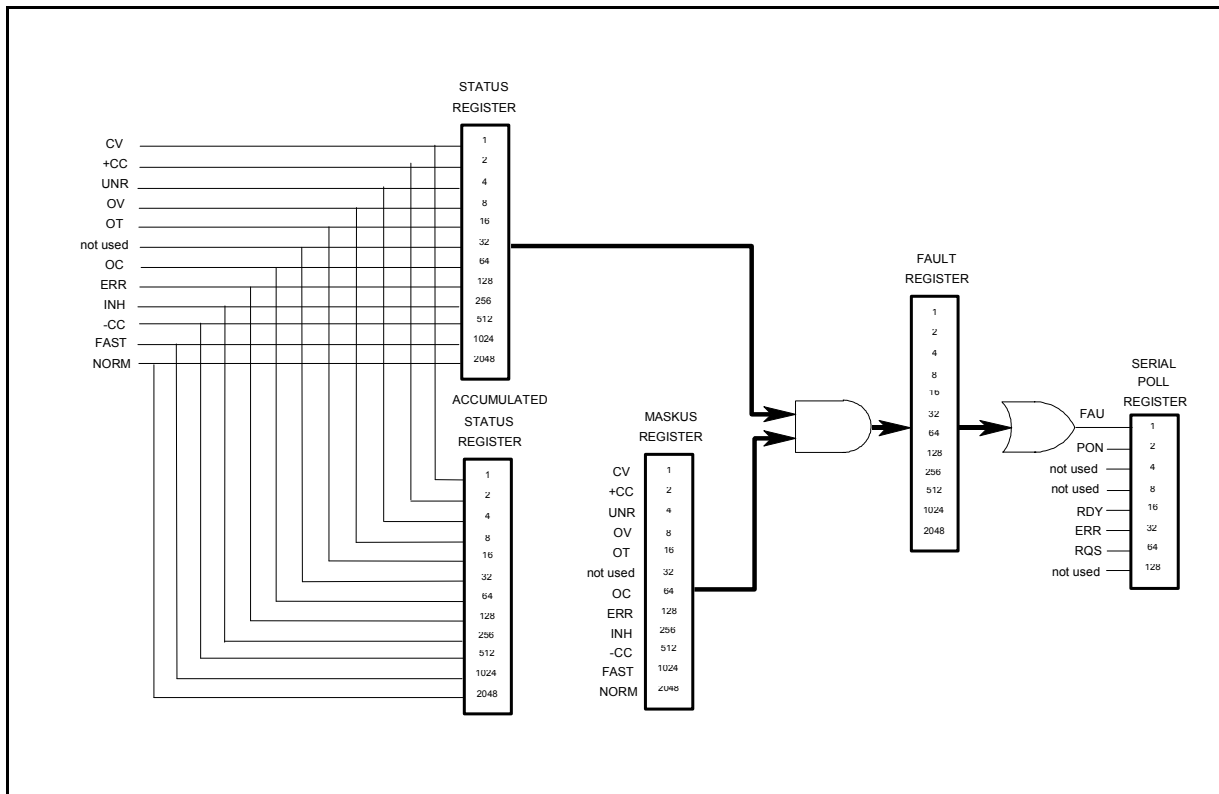
D 桁(数字)

S 符号 (正は空白、負は-)

Z 先頭の桁が0のときは、スペースが入る

表B-3. COMPatibilityエラー

エラー番号	エラー文字列 [内容/説明/例]
ERR 0	エラーなし
ERR 1	EEPROMのセーブが失敗 [不揮発性メモリへのデータの書込みが失敗しました。]
ERR 2	電源投入後の第2のPON [電源投入後に複数のPONコマンドを受信しました。<newline>1つしか受信できません。]
ERR 4	リレー・オプションなしにRLYPONを送信 [RLYPONコマンドをリレー・オプションなしで送信しました。]
ERR 5	リレー・オプションなし [リレー・オプション・コマンドがリレー・オプションなしに送信されました。]
ERR 8	トークにアドレス指定したが、応答がない [装置が最初にクエリを受け取らないでトークにアドレス指定されました。]
ERR 10	ヘッダを予想 [ヘッダを予想していたのに、英字以外の文字を受信しました。]
ERR 11	ヘッダが認識されない [受け取った英字から成る文字列が有効なコマンドではありません。]
ERR 20	数値を予想 [数字を予想していたのに、数字以外の文字を受信しました。]
ERR 21	数構文 [受け取った数字が適切な数ではありません。]
ERR 22	内部レンジ外の数値 [受け取った数値が内部形式で表すには長過ぎるか、短過ぎます。]
ERR 30	カンマ [予想した場所にカンマがありませんでした。]
ERR 31	ターミネータを予想 [予想した場所にターミネータがありませんでした。]
ERR 41	レンジ外のパラメータ [受け取った数値が関連コマンドのリミットを超えています。]
ERR 42	電圧設定エラー [設定された値が有効な電圧リミットを超えています。]
ERR 43	電流設定エラー [設定された値が有効な電流リミットを超えています。]
ERR 44	過電圧設定エラー [設定された値が有効な過電圧リミットを超えています。]
ERR 45	遅延設定エラー [設定された値が有効な遅延リミットを超えています。]
ERR 46	マスク設定エラー [設定された値がフォルト・マスク・リミットを超えています。]
ERR 51	EEPROMチェックサム [EEPROMが失敗したか、未校正のEEPROMが新たにインストールされました。]



図B-1. COMpatibilityステータス・モデル

表B-4. ステータス、Aステータス、フォルトおよびマスク・レジスタのビット割当て

ビット位置	11	10	9	8	7	6	5	4	3	2	1	0
ビット名	NORM	FAST	-CC	INH	ERR	OC	未使用	OT	OV	UNR	+CC	CV
ビットの重み	2048	1024	512	256	128	64	32	16	8	4	2	1

CV=装置は定電圧モードで動作しています。
 CC+=装置は定電流モードで動作しています。
 UNR=装置の出力はレギュレーションされていません。
 OV=過電圧保護回路が作動しました。
 OT=過熱保護回路が作動しました。
 OC=過電流保護回路が作動しました。
 ERR=設定エラーが発生しました。ERR?を使用してクリアしてください。
 CC-=装置は負定電流モードで動作しています。
 INH=外部リモート・インヒビット信号が出力をオフにしました。
 FAST=出力がFast動作モードにあります(Agilent 66332A, 6631B, 6632B, 6633B, 6634Bの場合のみ)。
 NORM=出力がNormal動作モードにあります(Agilent 66332A, 6631B, 6632B, 6633B, 6634Bの場合のみ)。

表B-5. シリアル・ポール・レジスタのビット構成

ビット位置	7	6	5	4	3	2	1	0
ビット名	未使用	RQS	ERR	RDY	未使用	未使用	PON	FAU
ビットの重み		64	32	16			2	1
<p>RQS=dcソースがサービス要求を生成しました。シリアル・ポーリングを使用してクリアしてください。</p> <p>ERR=ステータス・レジスタのERRビットと同じです。ERR?を使用してクリアしてください。</p> <p>RDY=このビットは、装置がコマンド処理中はクリアされます。処理が完了したらセットしてください。</p> <p>PON=電源が投入されました。CLRを使用してクリアしてください。</p> <p>FAU=ビットがフォルト・レジスタにセットされています。FAULT?を使用してクリアしてください。</p>								

エラー・メッセージ

エラー番号のリスト

本付録では、dcソースによって返されるエラー番号とその内容について説明します。エラー番号は、次の2通りの方法で返されます。

- エラー番号がフロント・パネルに表示されます。
- エラー番号とメッセージがSYSTEM:ERRor?クエリによって返されます。SYSTEM:ERRor?は、エラー番号を変数に戻し、2つのパラメータNR1と文字列を返します。

以下の表にSCPI構文エラーとインタフェースに関連するエラーをリストします。また、デバイス固有のエラーもリストします。角括弧内の情報は、標準エラー・メッセージの一部ではなく、エラー・メッセージに関する説明です。

エラーが発生すると、標準イベント・ステータス・レジスタは発生したエラーを次のように記録します。

ビット・セット	エラー・コード	エラーのタイプ	ビット・セット	エラー・コード	エラーのタイプ
5	-100~-199	コマンド	3	-300~-399 または1~32767	デバイス固有
4	-200~-299	実行	2	-400~-499	クエリ

表C-1. エラー番号

エラー番号	エラー文字列 [概要/説明/例]
-100	コマンド・エラー [包括的]
-101	無効な文字
-102	構文エラー [認識されないコマンドまたはデータ・タイプ]
-103	無効な区切り記号
-104	データ・タイプ・エラー [たとえば、数値または文字列を受け取るはずなのに、ブロック・データを受け取った]
-105	GETは不可
-108	パラメータは不可 [パラメータが多すぎる]
-109	パラメータが抜けている [パラメータが少なすぎる]
-112	プログラムのニモニックが長すぎる [最大12文字]
-113	ヘッダが定義されていない [このデバイスでは操作は不可]
-121	数字に無効な文字がある [8進法のデータに "9" があるなど]
-123	数値オーバフロー [指数が大きすぎる。指数の大きさ>32 k]
-124	桁数が多すぎる [数字が長すぎる。255桁以上を受け取った]
-128	数値データは不可
-131	接尾語が無効 [認識されない単位、または単位が適切でない]
-138	接尾語は不可

表C-1. エラー番号 (続き)

エラー番号	エラー文字列 [概要/説明/例]
– 141	無効な文字データ [不良文字、または認識されない文字]
– 144	文字データが長すぎる
– 148	文字データは不可
– 150	文字データ・エラー
– 151	無効な文字列データ [たとえば、閉じ引用符の前にENDを受け取った]
– 158	文字列データは不可
– 160	ブロック・データ・エラー
– 161	無効なブロック・データ [たとえば、指定の長さに達する前にENDを受け取った]
– 168	ブロック・データは不可
– 170	式エラー
– 171	無効な式
– 178	式データは不可
– 200	実行エラー [包括的]
– 222	データがレンジ外 [たとえば、デバイスに対してデータが大きすぎる]
– 223	データが多すぎる [メモリが不足。ブロック、文字列、または式が長すぎる]
– 224	違法なパラメータ値 [デバイス固有]
– 225	メモリ不足
– 270	マクロ・エラー
– 272	マクロ実行エラー
– 273	違法なマクロ・ラベル
– 276	マクロ反復エラー
– 277	マクロ再定義は不可
– 310	システム・エラー
– 350	エラーが多すぎる [9個を超えたエラーは、待ち行列オーバーフローのために消えます]
– 400	クエリ・エラー [包括的]
– 410	クエリが中断された [クエリの後、応答が完了する前にDABまたはGETを受け取った]
– 420	クエリが終了しない [トークにアドレスされ、不完全なプログラミング・メッセージを受け取った]
– 430	クエリがデッドロックになった [コマンド文字列内のクエリ件数が多すぎる]
– 440	クエリが終了しない [無期限の応答の結果]
0	エラーなし
1	不揮発性RAMのRD0セクションのチェックサムが失敗
2	不揮発性RAMのCONFIGセクションのチェックサムが失敗
3	不揮発性RAMのCALセクションのチェックサムが失敗
4	不揮発性RAMのSTATEセクションのチェックサムが失敗
5	不揮発性RSTセクションのチェックサムが失敗
10	RAMセルフテスト
11	VDAC/IDACセルフテスト1
12	VDAC/IDACセルフテスト2
13	VDAC/IDACセルフテスト3
14	VDAC/IDACセルフテスト4
15	OVDACセルフテスト
80	デジタルI/Oのセルフテスト・エラー

表C-1. エラー番号 (続き)

エラー番号	エラー文字列 [概要/説明/例]
213	Ingrdレシーバのバッファがオーバーラン
216	RS-232レシーバのフレーミング・エラー
217	RS-232レシーバのパリティ・エラー
218	RS-232レシーバのオーバーラン・エラー
220	フロント・パネルuartのオーバーラン
221	フロント・パネルuartのフレーミング
222	フロント・パネルuartのパリティ
223	フロント・パネル・バッファのオーバーラン
224	フロント・パネルのタイムアウト
401	CALスイッチによる校正のディスエーブル
402	CALパスワードが正しくない
403	CALがイネーブルになっていない
404	算出されたリードバック校正定数が正しくない
405	算出されたプログラミング校正定数が正しくない
406	校正コマンドの順番が正しくない
407	CVまたはCCステータスがコマンドに対して正しくない
408	出力モード・スイッチがNORMALの位置でなければならない
601	掃引ポイントが多すぎる
602	コマンドはRS-232インタフェースにだけ適用される
603	CURRENTまたはVOLTageのフェッチが最新の捕捉と互換性がない
604	測定がレンジを越えている

D - サンプル・プログラム

National InstrumentsのGPIBドライバ

プログラムには、National Instrumentsのヘッダ・ファイルDECL.BASが含まれていなければなりません。このファイルには、インタフェースの初期化コードが含まれています。アプリケーション・プログラムを実行する前に、必ず、コンフィギュレーション・プログラム(IBCONF.EXE)とのインタフェースを構築してください。

アプリケーション・プログラムにはdcソースのシンボル名やGPIBアドレスは含まれないので、設定中(IBCONF.EXEの実行時)に指定する必要があります。1次アドレスのレンジは0~30ですが、2次アドレスを96~126のレンジで指定しなければなりません。dcソースは、メッセージがEOIまたは改行で終了するものと期待するため、EOI w/last byte of Writeと設定してください。また、Disable Auto Serial Pollingに設定することもお勧めします。

関数呼出しはすべて、ステータス・ワードIBSTA%を返します。これには、呼出しがエラーに終わった場合に設定されるビット(ERR)が含まれています。ERRが設定されると、該当するコードが変数IBERR%に割り当てられます。関数呼出しの後には、必ず、IBSTA%をチェックしてください。ゼロでなければ、IBERR%を読み取って特定のエラーを抽出するエラー・ハンドラに分岐します。

エラー処理

プログラムにエラー処理コードが含まれていないと、検出されなかったエラーによって予期せぬ問題が生じる可能性があります。例えば、コントローラの「ハングアップ」や、システムのリセットの強制などです。前述のDOSドライバにはどちらも、プログラム実行エラーを検出するためのルーチンがあります。すべてのサブルーチン呼出しの後に、エラー検出を必ず入れてください。

Agilent BASICコントローラ

Agilent BASICプログラミング言語は、オペレーティング・システム・レベルでのGPIB機能へのアクセスを可能にします。このため、DOSアプリケーション・プログラムのフロントに必要なヘッダ・ファイルが不要になります。さらに、プログラムにタイムアウト・ステートメントが含まれていれば、コントローラの「ハングアップ」を心配する必要もありません。dcソースはエラー発生時にSRQを生成するように設定できるため、検出したエラーをデコードするためのSRQサービス・ルーチンをプログラムに使用することができます。検出可能なエラーについては、付録Cを参照してください。

例1. HP Vectra PCコントローラ(Agilent 82335インタフェース搭載)

```
5      '----- Merge SETUP.BAS here -----
1000  MAX.ELEMENTS=2 :ACTUAL.ELEMENTS=0 :MAX.LENGTH=80 :ACT.LENGTH=0
1005  DIM OUTPUTS(2) :CODES$=SPACE$(40)
1010  ISC=7 :PS=706
1015  '
1020  'Set up the DC Source Interface for DOS driver
1025  CALL IORESET (ISC)          'Reset the interface
1030  IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
1035  TIMEOUT=3
1040  CALL IOTIMEOUT (ISC, TIMEOUT)  'Set timeout to 3 seconds
1045  IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
1050  CALL IOCLEAR (ISC)          'Clear the interface
1055  IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
1060  CALL IOREMOTE (ISC)        'Set dc source to remote mode
1065  IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
1070  '
1075  'Program dc source to CV mode with following voltage and current
1080  CODES$ = "VOLTAGE MAX;CURRENT MAX" :GOSUB 2000
```

```

1085 '
1090 'Query dc source outputs CURRENT?" :GOSUB 2000 :GOSUB 3000
1100 VOUT = OUTPUTS(1)
1105 IOUT = OUTPUTS(2)
1110 PRINT "The output levels are "VOUT" Volts and "IOUT" Amps"
1115 '
1120 'Program triggered current level to value insufficient to maintain
1125 'supply within its CV operating characteristic
1130 CODES$ = "CURR:TRIG MIN" :GOSUB 2000
1135 '
1140 'Set operation status mask to detect mode change from CV to CC
1145 CODES$ = "STAT:OPER:ENAB 1024;PTR 1024" :GOSUB 2000
1150 '
1155 'Enable Status Byte OPER summary bit
1160 CODES$ = "*SRE 128" :GOSUB 2000
1165 '
1170 'Arm trigger circuit and send trigger to dc source
1175 CODES$ = "INITIATE:SEQUENCE1;TRIGGER" :GOSUB 2000
1180 '
1185 'Wait for supply to respond to trigger
1190 FOR I= 1 to 100 :NEXT I
1195 '
1200 'Poll for interrupt caused by change to CC mode and print to screen
1205 CALL IOS POLL (PS,RESPONSE)
1210 IF (RESPONSE AND 128) <> 128 THEN GOTO 1240 'No OPER event to report
1215 CODES$ = "STATUS:OPER:EVEN?" :GOSUB 2000 'Query status oper register
1220 CALL IOENTER (PS,OEVENT) 'Read back event bit
1225 IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
1230 IF (OEVENT AND 1024) = 1024 THEN PRINT "Supply switched to CC mode."
1240 'Clear the status circuit
1245 CODES$ = "*CLS" :GOSUB 2000
1250 FOR I = 1 TO 100 :NEXT I 'Wait for supply to clear
1255 '
1260 'Disable output and save present state in location 2
1265 CODES$ = "OUTPUT OFF;*SAV 2" :GOSUB 2000
1270 END
1275 '
2000 'Send command to dc source
2005 LENGTH = LEN(CODES$)
2010 CALL IOOUTPUTS (PS,CODES$,LENGTH) 'Send command to interface
2015 IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR 'SETUP.BAS error trap
2020 RETURN
2025 '
3000 'Get data from dc source
3005 CALL IOENTERA (PS,OUTPUTS(1),MAX.ELEMENTS,ACTUAL.ELEMENTS)
3010 IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
3015 RETURN

```

例2. IBMコントローラ(ナショナル・インタフェース搭載)

```

990  '----- Merge DECL.BAS here -----
1000 'DC Source Variable = PS% ; Stand-Alone Address = 706
1005 CODES$=SPACE$(50):MODE$=SPACE$(5):OEVENT$=SPACE$(20)
1010 D$=SPACE$(60):OUTPUT$=SPACE$(40):BDNAME$="PS%"
1015 DIM OUTPUT(2)
1020 '
1025 'Set up dc source interface for DOS driver
1030 CALL IBFIND(BDNAME$,PS%)
1035 IF PS%
1040 CALL IBCLR(PS%)
1045 '
1050 'Program dc source to CV mode with following voltage and current
1055 CODES$ = "VOLTAGE MAX;CURRENT MAX" :GOSUB 2000
1060 '
1065 'Query dc source outputs and print to screen
1070 CODES$ = "MEASURE:VOLTAGE?;CURRENT?" :GOSUB 2000 :GOSUB 3000
1075 VOUT = OUTPUT(1)
1080 IOUT = OUTPUT(2)
1085 PRINT"The programmed levels are "VOUT" Volts and "IOUT" Amps"
1090 '
1095 'Program triggered current level to value insufficient to maintain
1100 'supply within its CV operating characteristic
1105 CODES$ = "CURR:TRIG MIN" :GOSUB 2000
1110 '
1115 'Set operation status mask to detect mode change from CV to CC
1120 CODES$ = "STAT:OPER:ENAB 1024;PTR 1024" :GOSUB 2000
1125 '
1130 'Enable Status Byte OPER summary bit
1135 CODES$ = "*SRE 128" :GOSUB 2000
1140 '
1145 'Arm trigger circuit and send trigger to dc source
1150 CODES$ = "INITIATE:SEQUENCE1;TRIGGER" :GOSUB 2000
1160 'Wait for supply to respond to trigger
1165 FOR I= 1 to 100 :NEXT I
1170 '
1175 'Poll for interrupt caused by change to CC mode and print to screen
1180 SPOL%=0
1185 CALL IBRSP(PS%,SPOL%)
1190 IF (SPOL% AND 128) = 128 THEN POLL = 1 'Set interrupt flag on OPER bit
1195 IF POLL <> 1 THEN GOTO 1230 'No interrupt to service
1200 "CODES$ = "STAT:OPER:EVEN?" :GOSUB 2000 'Query status oper register
1205 CALL IBRD(PS%,OEVENT$) 'Read back event bit
1210 IF IBSTA%
1215 OEVENT=VAL(OEVENT$)
1220 IF (OEVENT AND 1024) = 1024 THEN PRINT "Supply switched to CC mode."
1225 '
1230 'Clear status circuit
1235 CODES$="*CLS" :GOSUB 2000
1240 FOR I=1 TO 50 :NEXT I 'Wait for supply to clear
1245 '
1250 'Disable output and save present state to location 2
1255 CODES$ = "OUTPUT OFF;*SAV 2" :GOSUB 2000
1260 END
1265 '

```

```

2000 'Send command to dc source
2005 CALL IBWRT(PS%,CODES$)
2010 IF IBSTAT%
2015 RETURN
1250 'Disable output and save present state to location 2
1255 CODES$ = "OUTPUT OFF;*SAV 2" :GOSUB 2000
1260 END
1265 '
2000 'Send command to dc source
2005 CALL IBWRT(PS%,CODES$)
2010 IF IBSTAT%
2015 RETURN
2020 '
2100 'Error detection routine
2105 PRINT "GPIB error. IBSTAT% = HEX$(IBSTAT%)
2110 PRINT "          IBERR% = ";IBERR%" in line ";ERL
2115 STOP
2120 '
3000 'Get data from dc source
3005 CALL IBRD(PS%,OUTPUT$)
3010 IF IBSTA%
3015 I=1 'Parse data string
3020 X=1
3025 C=INSTR(I,OUTPUT$,";")
3030 WHILE C <> 0
3035 D$=MID$(OUTPUT$,I,C-I)
3040 OUTPUT(X)=VAL(D$) 'Get values
3045 I=C+1
3050 C=INSTR(I,OUTPUT$,";")
3055 X=X+1
3060 WEND
3065 D$=RIGHT$(OUTPUT$,LEN(OUTPUT$)-(I-1))
3070 OUTPUT(X)=VAL(D$)
3075 OUTPUT$=SPACE$(40) 'Clear string
3080 RETURN

```

例3. Agilent BASIC搭載のコントローラ

```
1000 !Dc source at stand-alone address = 706
1005 OPTION BASE 1
1010 DIM Codes$(80),Response$(80),Mode$(32)
1015 !
1020 !Program dc source to CV mode with following voltage and current
1025 OUTPUT 706;"VOLTAGE MAX;CURRENT MAX"
1030 !
1035 !Query dc source outputs and print to screen
1040 OUTPUT 706;"MEASURE:VOLTAGE?;CURRENT?"           !Query output levels
1045 ENTER 706;Vout,Iout
1050 PRINT "The output levels are ";Vout;" Volts and ";Iout" Amps"
1055 !
1060 !Program current triggered level to a value insufficient to maintain
1065 !supply within its CV operating characteristic
1070 OUTPUT 706;"CURR:TRIG MIN"
1075 !
1080 !Set operation status mask to detect mode change from CV to CC
1085 OUTPUT 706;"STAT:OPER:ENAB 1024;PTR 1024"
1090 !
1095 !Enable Status Byte OPER summary bit
1100 OUTPUT 706;"*SRE 128"
1105 !
1110 !Arm trigger circuit and send trigger to dc source
1115 OUTPUT 706;"INITIATE:SEQUENCE1;TRIGGER"
1130 !Poll for interrupt caused by change to CC mode and print to screen
1135 Response=SPOLL(706)
1140 IF NOT BIT (Response,7) THEN GOTO 1130           !No OPER event to report
1145 OUTPUT 706;"STAT:OPER:EVENT?"                 !Query status operation register
1150 ENTER 706;Oevent                               !Read back event bit
1155 IF BIT(Oevent,10) THEN PRINT "Supply switched to CC mode."
1160 !
1165 !Clear status
1170 OUTPUT 706;"*CLS"
1175 !
1180 !Disable output and save present state in location 2
1185 OUTPUT 706;"OUTPUT OFF;*SAV 2"
1190 END
```


索引

— R —

— A —

AARD, 20
ABORT, 79
ACDC, 56
Agilent 8235A ドライバ, 99
Agilent BASICコントローラ, 100

— C —

CRD, 20

— D —

DC, 56
DC測定, 27
DFI, 42
DFIプログラミング例, 43
DOSドライバの種類, 99
DTR-DSR, 13

— F —

FLT, 42

— G —

GPIB
 dcソースの機能, 12
 MS DOS用コマンド・ライブラリ, 10
 アドレス, 12
 コントローラ・プログラミング, 10
 参考文献, 10
 トリガ, 30
 標準コードのIEEE規格, 10
 標準デジタル・インタフェースのIEEE規格, 10

— I —

INH, 42

— M —

MAVビット, 40
MSSビット, 40

— N —

National Instruments GPIBドライバ, 100

— O —

OCP, 24

— P —

PON(パワーオン)ビット, 39

RI, 42
RQSビット, 40
RS-232
 dcソースの機能, 12
 データ・フォーマット, 13, 14
 データ・ターミネータ, 19
 フロー制御, 13
RTS-CTS, 13

— S —

SCPI
 応答メッセージ, 18
 共通コマンド, 16
 コマンド・ツリー, 16
 コマンド構文, 45
 コマンドの完了, 21
 サブシステム・コマンド, 16, 45
 参考文献, 10
 データ・フォーマット, 20
 デバイス・クリア, 21
 適合性, 87
 トリガの名称, 25, 29
 非SCPIコマンド, 88
 複数のコマンド, 16
 プログラム・メッセージ, 18
 ヘッダ経路, 16
 メッセージ・ユニット, 18
 メッセージの構造, 18
 メッセージの種類, 18
 SCPIコマンドの種類, 16
 SRD, 20

— X —

XON-XOFF, 13

— あ —

アクティブ, 69
安全性に関するガイドライン, 2
いずれか, 84
印刷日, 2
エラー処理, 100
エラー番号, 92
オプション・ヘッダ
例, 17

— か —

過電流保護, 24
起動コマンド, 79
 INIT CONT NAME, 79
 INIT CONT SEQ, 79
 INIT NAME, 79
 INIT SEQ, 79
 キャラクタ文字列, 20
 共通コマンド, 49, 74

- *CLS, 70
- *ESE, 71
- *ESR?, 71
- *IDN?, 76
- *OPC, 71
- *OPT?, 77
- *PSC, 72
- *RCL, 77
- *RST, 77
- *SAV, 78
- *SRE, 72
- *STB?, 73
- *TRG, 85
- *TST, 78
- *WAI, 73
- 共通コマンドの構文, 49
- クウェリ, 17
 - インジケータ, 18
- クエスチョナブル・ステータス・イベントの処理, 41
- クエスチョナブル・ステータス・グループ, 39
- 言語, 89
- 言語辞書, 45
- 校正コマンド, 50
 - CAL CURR, 50
 - CAL CURR MEAS AC, 50
 - CAL CURR MEAS LOWR, 50
 - CAL CURR NEG, 50
 - CAL DATA, 51
 - CAL LEV, 51
 - CAL PASS, 51
 - CAL SAVE, 51
 - CAL STAT, 52
 - CAL VOLT, 52
 - CAL VOLT PROT, 52
- 互換
 - エラー, 92
 - 言語, 89
 - コマンド, 90
 - ステータス・モデル, 93
 - 電源投入時の設定, 89
- 個別フォールト・インジケータ, 42
- コマンドの完了, 21
- コマンドの結合
 - 共通コマンド, 17
 - 異なるサブシステムのコマンド, 17
 - ルート指示子, 17

— さ —

- 最小値測定, 28
- 最大値測定, 28
- サブシステム・コマンドの構文, 45
- サブシステム間の移動, 17
- システム・エラー, 95
- システム・コマンド, 74
 - SYST ERR?, 75
 - SYST LANG, 75, 89
 - SYST LOC, 76

- SYST REM, 76
- SYST RWL, 76
- SYST VERS?, 75
- 実効値測定, 28
- 出力コマンド, 61
 - OUTP, 61
 - OUTP DFI, 61
 - OUTP DFI SOUR, 61
 - OUTP PON STAT, 62
 - OUTP PROT CLE, 62
 - OUTP PROT DEL, 62
 - OUTP REL, 63
 - OUTP REL POL, 63
 - OUTP RI MODE, 63
- 出力トリガ・システム・モデル, 25
- 出力トリガ・システムの起動, 26
- 出力トリガ・システムの設定, 25
- 出力のイネーブル, 23
- 出力のプログラミング, 23
- 出力パルスの測定, 32
- 出力変動のトリガ, 25
- 出力待ち行列, 40
- 初期化, 23
- 数値データ・フォーマット, 20
- ステータス・コマンド, 67
 - STAT OPER COND?, 68
 - STAT OPER ENAB, 68
 - STAT OPER NTR, 68
 - STAT OPER PTR, 68
 - STAT OPER?, 67
 - STAT PRES, 67
 - STAT QUES COND?, 69
 - STAT QUES ENAB, 69
 - STAT QUES NTR, 70
 - STAT QUES PTR, 70
 - STAT QUES?, 69
- ステータス・バイト・レジスタ, 40
- ステータス・ビット構成, 38
- ステータス・モデル, 37
- ステータス・レジスタのプログラミング, 36
- ステータス遷移の両相のモニタ, 41
- 正, 84
- 接尾語, 20
- センス・コマンド, 53
 - SENS CURR DET, 58
 - SENS CURR RANG, 58
 - SENS FUNC, 59
 - SENS SWE OFFS POIN, 59
 - SENS SWE POIN, 59
 - SENS SWE TINT, 59
 - SENS WIND, 60
- ソース・コマンド, 64
 - [SOUR] CURR, 64
 - [SOUR] CURR PROT STAT, 64
 - [SOUR] CURR TRIG, 64
 - [SOUR] DIG DATA, 65
 - [SOUR] DIG FUNC, 65
 - [SOUR] VOLT, 65

[SOUR] VOLT ALC BAND?, 66
[SOUR] VOLT PROT, 66
[SOUR] VOLT TRIG, 66
測定結果待ち, 31
測定コマンド, 27, 53
 MEAS ARRay CURR?, 53
 MEAS ARRay VOLT?, 53
 MEAS CURR ACDC?, 54
 MEAS CURR HIGH?, 54
 MEAS CURR LOW?, 55
 MEAS CURR MAX?, 55
 MEAS CURR MIN?, 55
 MEAS CURR?, 54
 MEAS VOLT ACDC?, 56
 MEAS VOLT HIGH?, 56
 MEAS VOLT LOW?, 57
 MEAS VOLT MAX?, 57
 MEAS VOLT MIN?, 57
 MEAS VOLT?, 56
測定トリガ・システム・モデル, 29
測定トリガ・システムの起動, 29
測定トリガ・ソースの選択, 30
測定トリガの生成, 30
測定の実施, 27

— た —

長方形, 60
デジタルI/Oポート, 42
デバイス・クリア, 21
電圧, 24
 最大, 24
 測定, 27
電圧データまたは電流データの戻り, 27
電圧または電流のサンプリングの変更, 33
電源投入時の初期化, 23
電流, 24
 最大, 24
 測定, 27
 測定レンジ, 28
電流測定ディテクタ, 32, 58
電流測定レンジ, 58
動作ステータス・グループ, 38
動作ステータスの処理, 41
トリガ・コマンド, 79
 TRIG, 80
 TRIG ACQ, 80
 TRIG ACQ COUN CURR, 81
 TRIG ACQ COUN VOLT, 81
 TRIG ACQ HYST CURR, 82
 TRIG ACQ HYST VOLT, 82
 TRIG ACQ LEV CURR, 83
 TRIG ACQ LEV VOLT, 83
 TRIG ACQ SLOP CURR, 84
 TRIG ACQ SLOP VOLT, 84
 TRIG ACQ SOUR, 85
 TRIG SEQ1 DEF, 85
 TRIG SEQ2, 80

TRIG SEQ2 COUN CURR, 81
TRIG SEQ2 COUN VOLT, 81
TRIG SEQ2 DEF, 85
TRIG SEQ2 HYST CURR, 82
TRIG SEQ2 HYST VOLT, 82
TRIG SEQ2 LEV CURR, 83
TRIG SEQ2 LEV VOLT, 83
TRIG SEQ2 SLOP CURR, 84
TRIG SEQ2 SLOP VOLT, 84
TRIG SEQ2 SOUR, 85
TRIG SOUR, 80
 シングル, 26
 連続, 26
トリガの生成, 26

— な —

内部, 85
内部トリガ, 29
内部トリガ測定, 29

— は —

はじめに, 9
バス, 85
ハニング, 60
パラメータのプログラミング, 49
パルス測定クウェリ, 32
パルス測定例, 34
パルス波形, 32
パワーオン条件, 36
表示コマンド, 74
 DISP, 74
 DISP MODE, 74
 DISP TEXT, 74
標準イベント・ステータス・グループ, 39
負, 84
フェッチ・コマンド, 27, 53
フォールト・インジケータ
 個別, 42
 リモート・インヒビット, 42
複数測定, 33
プライベート・トリガ, 34
プログラム例, 99
ヘッダ, 18
 ショート形式, 18
 ロング形式, 18
ポストイベント・トリガ, 34
本書で使用されている表記法, 15

— ま —

メッセージ・ターミネータ, 19
 終わりまたは識別, 19
 改行, 19
メッセージ・ユニット
 セパレータ, 19

— ら —

ラッチ, 62

リモート・インヒビット, 42

履歴, 2

ルート指示子, 19

例

Agilent BASIC使用コントローラ, 104

Agilent 82335インタフェース装備HP Vectra, 100

DFIプログラミング, 43

Nationalインタフェース使用IBMコントローラ, 102

パルス測定, 34

プログラム, 99

— わ —

割込みの原因の究明, 40