

## リモート動作

---

- ・ リモート動作とローカル動作の差異—リモート動作の拡張とリモート動作からローカル動作への移行について説明します。
- ・ インタフェースの選択— GPIB と RS-232 インタフェースの選択方法を説明します。
- ・ GPIB の動作— GPIB のバス規格、バス接続、一次アドレス選択について説明します。
- ・ 汎用バスコマンド—基本的な GPIB 制御に使用するバスコマンドを説明します。
- ・ 前面パネル GPIB 操作— GPIB エラーメッセージ、ステータス標識、SCPI コマンドの使用について説明します。
- ・ プログラミングシンタクス—コモンコマンドと SCPI コマンドについての基本的なプログラミングシンタクスを説明します。
- ・ RS-232 インタフェース動作— RS-232 インタフェースを使って、リモート経由でソース・メータを制御する方法の概要を説明します。

## リモート動作とローカル動作の差異

### 動作の拡張(リモート動作)

前面パネルからは実行不可能ですが、IEEE-488 バスと RS-232 インタフェースを介して実行可能ないくつかのソース-メジャー動作があり、それを以下に要約します。

#### 数式

前面パネル (FCTN) 動作、リモート動作の両方に利用可能な事前定義数式は、4 個あります。しかし、リモート動作の場合には、5 個までのユーザ定義数式を定義し、合計 9 個の数式を使用することができます (CALCulate 1 サブシステム参照)。

#### 同時測定

TOGGLE キーを使用すれば、2 つの機能を同時に測定 (ディスプレイ) することができます。リモート動作を利用すれば、3 つの機能 (電圧、電流、抵抗) すべてについて、同時測定を行うことができます。詳細については SENSE サブシステムを参照してください。

**注記** 2430 型ではパルスモードに入っている間は同時測定は不可能です。パルスモード動作の詳細は、第 5 部を参照してください。

### ローカル動作からリモート動作への移行

ローカル動作からリモート動作に移行する時には、次のアクションが起こります。

- ・ ソースメータはソース-メジャー動作を停止し、アイドル状態 (ARM アナナシーエタ消灯) に戻ります。
- ・ すべてのスweep動作が途中で停止します。
- ・ すべてのメニューから抜け出します。
- ・ すべての実行中パネルコマンドが途中で停止します。
- ・ ソース編集とコンプライアンス編集が、使用禁止になります。
- ・ サンプルバッファの中のデータが失われます (すなわち:FETCn?、:CALC1:DATA?、:CALC2:DATA?は、リモート動作中に読取りが行われるまで、どのデータも返しません)。
- ・ 同時測定が使用可能になります。
- ・ そのほかの設定は、:TRACe バッファの設定 (データストア) を含めて、すべて影響を受けません。

## リモート動作からローカル動作への移行

リモート動作からローカル動作に移行する時には、次のアクションが起こります。

- ・ ソースメータはソース-メジャー動作を停止し、アイドル状態（ARM アナライザ消灯）に戻ります。
- ・ すべてのスイープ動作が途中で停止します。
- ・ すべてのユーザ定義ディスプレイメッセージが取消となります。
- ・ ディスプレイが入ります（切っている場合）。
- ・ ソースオートレンジ設定が解除状態になります。
- ・ 同時測定が使用状態になります。
- ・ 抵抗測定が使用状態になれば、ソースリードバックが使用状態になります。
- ・ ディスプレイは、デフォルトであるトグル状態に設定されます。
- ・ 読取りは継続して行われます（OUTPUT がオン状態の場合）。

## インタフェースの選択

2400 型は、下記の 2 つの内蔵リモートインタフェースをサポートします。

- ・ GPIB バス
- ・ RS-232 インタフェース

2 つのインタフェースを同時に使用することはできません。工場出荷時のインタフェース選択は、GPIB バスです。インタフェースの選択ができるのは、前面パネルからに限られます。インタフェースの選択は、不揮発性メモリに格納されます。この選択は電源を切った状態でも、またリモートインタフェースをリセットしたあとでも、変化しません。

GPIB バスは IEEE-488 インタフェースです。2400 型のための、固有のアドレスを選択しなければなりません。このアドレスは、ソースメータの電源が入った時にディスプレイされます。工場出荷時には、このアドレスは、24 に設定されています。

RS-232 インタフェースは、シリアルインタフェースです。このインタフェースのプログラム可能な項目は次のとおりです。括弧の中は、工場出荷時のデフォルト値です。

- ・ ボーレート (9600)
- ・ データビット (8)
- ・ パリティ (なし)
- ・ ターミネータ (CR)
- ・ フロー制御 (なし)

インタフェースは、メインメニューの COMMUNICATIONS オプションから選択し、設定します（第 3 部の「メインメニュー」参照）。各インタフェースのプログラム可能な項目の詳細については、「GPIB バスの動作と基準、一次アドレス」と「RS-232 の動作」を参照してください。

**注記** インタフェース選択を変更する場合、ソースメータはリセットされて、メニュー体系から抜け出します。選択したインタフェースの確認または変更を行うためには、メニュー体系に再度入る必要があります。

## GPIB バスの動作

第 14 部では、GPIB 規格、バス接続、一次アドレス選択について説明します。

### GPIB バス規格

GPIB バスは、IEEE-488 計測データバスで、1975 年に IEEE（米国電気学会）が初めて採用したハードウェアとプログラミング規格によるものです。2400 型は、これらの規格に合致します。

- ・ IEEE-488.1-1987
- ・ IEEE-488.2-1992

この規格が定義するのは、計測器へと計測器からのデータの送信に関するシンタクス、計測器がこのデータを解釈する方法、計測器の状態を記録するために存在するレジスタの種類、コマンドのグループです。

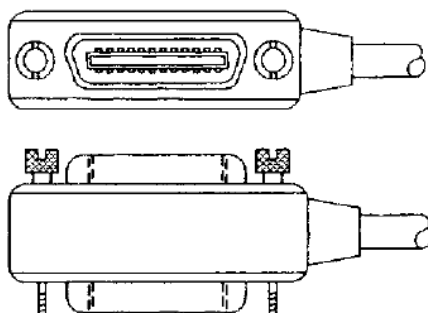
- ・ SCPI 1996.0（プログラマブル計測器用標準コマンド）

この規格は、共通言語プロトコルを定義します。この規格は、IEEE-488-1987.2 よりも、一歩先を進んでおり、計測器のプログラム可能な項目すべてを制御するためのコマンドの、標準的なセットを定義します。

### GPIB バス接続

2400 型を GPIB バスに接続するには、図 14-1 に示すような標準 IEEE-488 コネクタを備えるケーブルを使用してください。

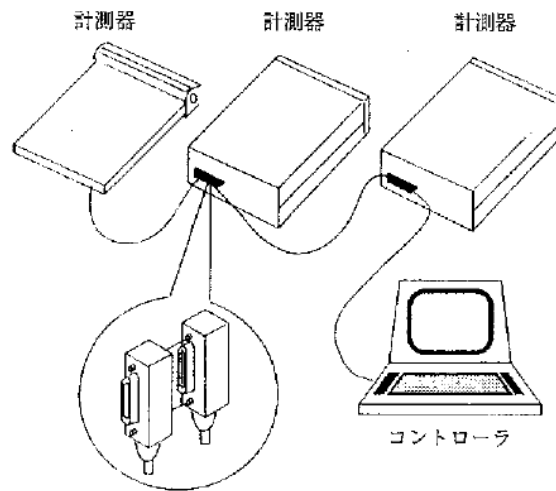
図 14-1  
IEEE-488 コネクタ



1 個の計測器に多数の並列接続を行えるようにするには、コネクタをスタックしてください。それぞれのコネクタは 2 個のねじを備えており、確実な接続状態を維持するようになっています。現在の規格はメートルねじの使用を規定しています。これには識別のために濃い色のねじがついています。初期のバージョンは、これと異なるねじを使っていますが、これは銀色です。この種のコネクタを 2400 型では使用しないでください。2400 型はメートルねじ用に設計されています。

図 14-2 は、複数ユニット用試験システムの代表的な接続方式を示します。

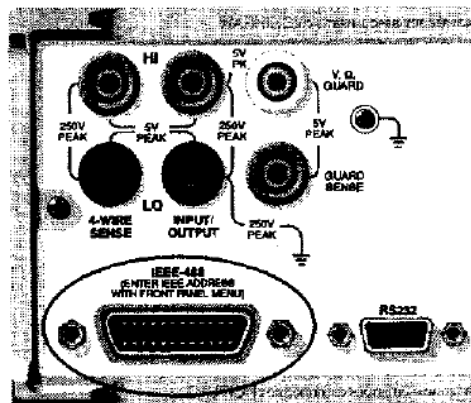
図 4-2  
IEEE-488 接続



機械的な損傷の発生を避けるために、どのようなユニットにコネクタをスタックする場合も、3 個までとしてください。

注記 電磁放射による干渉を最小限に抑えるために、シールド付き IEEE-488 ケーブルだけを使用してください。ケースレーが用意しているシールドケーブルは 7001-1 型と 7007-2 型です。

図 14-3  
IEEE-488 コネクタの位置



2400 型を IEEE-488 バスに接続するには、次のステップに従ってください。

1. 背面パネルにあるコネクタに、ケーブルコネクタを合わせてください。コネクタは、合せ方が一通りしかないように設計されています。図14-3は、IEEE-488コネクタの位置を示します。
2. ねじを確実に締めます。締めすぎに注意してください。
3. ご使用の目的に必要なほかの計測器からの追加コネクタを接続してください。
4. ケーブルのもう一方の端がコントローラに正しく接続されていることを確認してください。ほとんどのコントローラはIEEE-488式のコネクタを備えていますが、一部のコントローラでは、別の種類の接続ケーブルが必要になることがあります。IEEE-488バスへの正しい接続方法についての詳細は、ご使用のコントローラの取扱説明書を参照してください。

**注記** 1本のIEEE-488バスに接続可能なデバイスの数は、コントローラを含め、最大15台です。最大ケーブル長は、20m か、デバイスの数×2m の、どちらか小さい方です。この限界を守らないと、バスが誤動作することがあります。

## 一次アドレス

2400 型は、 GPIB のアドレスを 24 として、工場から出荷されます。電源投入時には、一次アドレスが瞬間的にディスプレイされます。アドレスは、0-30 の値に設定することができます。同じアドレスを別のデバイスに、または同一 GPIB 上のコントローラに割り当てないようにしてください。（通常は、コントローラのアドレスは、0 か 21。）

一次アドレスの点検と変更、またはどちらでも、メインメニューの COMMUNICATIONS オプションを使って実行することができます（第1部の「メインメニュー」参照）。

## 汎用バスコマンド

汎用バスコマンドとは、DCLのように、計測器とは無関係に同一の汎用的な意味を持つコマンドです。表 14-1 に汎用バスコマンドを示します。

表 14-1  
汎用バスコマンド

コマンド	ソース・メータへの影響
REN	次回リスンするように呼びかけられたときに有効になります。
IFC	トーカーとリスナのアイドル状態に入ります。
LLO	LOCAL キーがロックアウトされています。
GTL	リモート動作を取り消します。ソース・メータの前面パネル操作が復旧します。リモート動作を取り消します。すべてのデバイスについて前面パネル操作が復旧します。
DCL	すべてのデバイスを既知の条件に戻します。
SDC	ソース・メータを既知の条件に戻します。
GET	トリガを開始します。
SPE, SPD	ソース・メータをシリアルボールします

## REN (remote enable)

リモートイネーブルコマンドは、計測器をリモート操作できるようにセットアップするために、コントローラが2400型に送るコマンドです。一般に、バスを介して計測器をプログラムする前に、計測器をリモートモードに入れる必要があります。RENを正しく設定するだけでは、計測器は実際にはリモート状態に入りません。RENを正しく設定したあと、計測器をリモート動作に入れる前には、計測器にlistenするように呼びかけなければなりません。

下記のコマンドを使い、読み取りをトリガし、読み取り値を取得するには、ソース・メータはリモート状態に入っていないなければなりません。

- ・ :INITiate 続いて:FETCh
- ・ :READ?
- ・ :MEASure?

## IFC (interface clear)

IFC コマンドはコントローラによって送られ、ソース・メータをローカル、トーカー、リスナーアイドルの各状態に入れます。ユニットはIFC コマンドに応答し、計測器がこれまでに上記の状態の1つに入ったことがあれば、前面パネル TALK または LSTN 表示灯を消灯します。

このコマンドは計測器のステータスを変更しないことに注目してください。設定値、データ、イベントレジスタは変わりません。

オート出力オフが使用可能な状態では (:SOURce1:CLEar:AUTO ON)、出力が自動的にオフ状態になる前に動作が終了すると、出力はオン状態に留まります。

IFC コマンドを送るには、最小 100  $\mu$  の間、コントローラが IFC ラインを正しく設定すれば十分です。

## LLO (local lockout)

LLO コマンドを使用して、計測器のローカル操作を防止してください。ユニットが LLO を受信したあと、OUTPUT OFFを除いて、前面パネル制御はすべて無効になります。この状態では、LOCAL を押しても前面パネルには制御機能は復活しません。GTL コマンドが前面パネルに制御を復活させます。電源を入れ直しても、ローカルロックアウトを解除することができます。

## GTL (go to local)

GTL コマンドを使用して、リモートモードの計測器をローカルモードにします。GTL コマンドは、また、前面パネルキー操作を復活させます。

## DCL (device clear)

DCL コマンドを使用して、**GPIB** インタフェースをクリアし、既知の状態に戻してください。**DCL** コマンドはアドレスを指定されたコマンドではないので、**DCL** を実施するようにされたすべての計測器が同時にデバイスクリアされることに留意してください。

2400 型が **DCL** コマンドを受けると、入力バッファと出力待ち行列をクリアし、保留コマンドを取り消し、ほかのデバイスのコマンドの処理を妨げるコマンドをすべてクリアします。**DCL** は、計測器の設定値と格納データに影響を与えません。

## SDC (selective device clear)

**SDC** コマンドは、**DCL** コマンドと基本的には同じ機能を実行する、アドレスを指定されたコマンドです。しかし、それぞれのデバイスが個別にそのアドレスの指定を受けていなければならないので、**SDC** コマンドは、**DCL** の場合のようにすべての計測器を同時にクリアするのではなくて、選択したものだけをクリアする方法を提供します。

## GET (group execute trigger)

**GET** は、動作を制御するためのアームイベントとして利用される **GPIB** トリガです。**GET** が、プログラムされたアーム制御ソースになっていれば、2400 型はこのトリガに対して反応します。下記のコマンドは、**GPIB** アーム制御ソースを選択します。

**:ARM:SOURce BUS**

**注記** **:ARM:SOURce BUS** を選択した場合は、ソース・メジャー動作を行っている間は (**ARM** アナライザ点灯)、どのコマンド (**GET**、**DCL**、**SDC**、**\*TRG**、**:ABORt** を除く) も送らないでください。もし送れば、誤動作が発生します。

計測器のプログラムが終わり、**GPIB** トリガを待っている時は、下記のプログラムフラグメントが **GET** を供給します。

## SPE、SPD (serial polling)

シリアルポーリングシーケンスを使用して、ソース・メータのシリアルボールバイトを求めてください。シリアルボールバイトには、内部機能に関する重要な情報が含まれています。(第 15 部の「ステータス体系」参照) 一般に、シリアルポーリングシーケンスは、コントローラが使用し、いくつかの計測器のうち、どれが **SRQ** ラインを使用するかを決定します。しかし、ソース・メータから状態ステータスバイトを求めようとするときは、いつでもシリアルポーリングシーケンスを実行することができます。



## 前面パネル GPIB 動作

この部では、GPIB 動作の一部となっている前面パネルの動作項目を説明します。これにはメッセージ、ステータスインジケータ、LOCAL キーが含まれます。

### エラーメッセージとステータスメッセージ

IEEE-488 のプログラミングと関連するエラーメッセージ、ステータスメッセージの一覧については、付録 B を参照してください。計測器は、SRQ を発生するようにプログラムすることができます。特定のエラー条件の有無をチェックするために、コマンド照会を行うことができます。

### GPIB の状態標識

REM (remote)、TALK (talk)、LSTN (listen)、SRQ (service request) の各アナウンシエータは、GPIB のバスステータスを示します。これらのステータス・インジケータそれぞれについて、以下に説明します。

#### REM

このインジケータは、計測器がリモート状態にある時に現れます。REM は必ずしも REM ラインの状態を表示しません。その理由は、REM 表示が現れる前に、計測器は、REM が true になり、listen するようにアドレスされなければならないからです。計測器がリモート状態にある場合には、LOCAL キーを除いて、すべての前面パネルキーはロックアウトされます。REM が消えた場合には、計測器はローカル状態にあり、前面パネル操作が復活します。

**注記** LLO が有効な場合は、LOCAL はロックアウトされます。OUTPUT ON/OFF は、リモート状態では依然として働いています。ARM:SOUR が手動に設定されていると、TRIG キーはリモート状態ではアクティブになります。

#### TALK

このインジケータが現れるのは、計測器がトーカー活動状態にある場合です。正しい MTA (My Talk Address) コマンドを使用して、ユニットに対し、talk するようにアドレスされ、ユニットをトーク状態にしてください。ユニットがトーカーアイドル状態にある場合には、TALK はオフ状態となります。UNT (Untalk) コマンドを送り、ユニットに listen するようにアドレスされ、または IFC (インタフェースクリア) コマンドを送ることにより、ユニットをトーカーアイドル状態にしてください。

#### LSTN

この標識が現れるのは、2400 型がリスナ活動状態にある場合です。この状態を起動するには、正しい MLA (My Listen Address) を使用して、計測器に対し、listen するように呼びかけます。ユニットがリスナアイドル状態にある場合には、LSTN は消えます。UNL (Unlisten) を送ることにより、ユニットに talk するように呼びかけるか、またはバスを経由して IFC (インタフェースクリア) コマンドを送ることにより、ユニットをリスナアイドル状態にしてください。

#### SRQ

1 つ以上のエラー条件が発生した場合にサービスリクエスト (SRQ) を発生させるように計測器をプログラムすることができます。この標識が現れる場合は、サービスリクエストが発生しています。この標識が消えるのは、シリアルポールの読取りが終った時、または SRQ 発生の原因となったすべての条件がなくなった時です。詳しくは、第 15 部「ステータス構造」を参照してください。

## LOCAL キー

LOCAL キーは、計測器のリモート状態を取り消し、ローカル動作を復活させます。

また LOCAL キーを押すと、REM 標識が消え、ユーザ定義メッセージがディスプレイされていればこれを通常状態に戻します。

LLO（ローカルロックアウト）コマンドが有効な場合は、LOCAL キーは無効になります。

安全上の理由から、LLO では、OUTPUT キーは依然としてアクティブです。

## プログラミング文法

この部の情報は、コモンコマンドと SCPI コマンド、両方の文法に関するものです。ここに記載されていない情報については、IEEE-488.2 規格と SCPI 規格を参照してください。コモンコマンドと SCPI コマンドについての詳細は、それぞれ第 16 部と第 18 部を参照して下さい。

### コマンド語

プログラムメッセージは、1 語以上のコマンド語で設定されます。

#### コマンドとコマンドパラメータ

コモンコマンドと SCPI コマンドは、パラメータを使用することもあれば、使用しないこともあります。いくつかの例を下記に示します。

*SAV <NRf>	パラメータ (NRf) が要求されています。
*RST	パラメータは使用されていません。
:CALCulate:STATe <b>	パラメータ <b> が要求されています。
:SYSTem:PRESet	パラメータは使用されていません。

**注記** コマンド語とパラメータとの間には、少なくとも 1 スペースが必要です。

ブラケット[]—一部のコマンド語は、ブラケット ([]) で囲まれています。このようなブラケットを使用するのは、プログラムメッセージの中に含める必要のないオプションコマンド語を表すためです。

例

```
:INITiate[:IMMediate]
```

ここで使うブラケットは、:IMMediate がオプションであり、必ずしも使用する必要のないことを示します。したがって、上記のコマンドは 2 つの形式のどちらか 1 つで送ることができます。

```
:INITiate
または
:INITiate:IMMediate
```

オプションコマンドがブラケットなしで使用されることに注目してください。お客様のプログラムの中でオプションコマンドを使用する場合は、ブラケットを含めないでください。

パラメータの種類—最も広く使用されるパラメータ種類の一部を、以下に示します。

- <b>** プーリアン—これを使って、計測器の動作を使用状態または解除状態にします。0 または OFF は動作を解除状態に、1 または ON は動作を使用状態にします。例  
:CALCulate1:STATe ON Calc 1 数式を使用状態にせよ。
- <name>** ネームパラメータ—リストに挙げた1つのグループからパラメータネームを選択します。例  
:name = NEVer  
= NEXt  
:TRACe:FEED: CONTrol NEXt
- <NRf>** 数値表現フォーマット—このパラメータは数値であり、整数（たとえば8）、実数（たとえば23.6）、または指数（2.3E6）で表現することができます。例  
:SYSTem:KEY 11 バスを経由してEXITを押せ。
- <n>** 数値—数値パラメータは、1個のNRfパラメータ、または下記のネームパラメータ、すなわちDEFault、MINimum、MAXimumのうちの1つで設定することができます。DEFaultパラメータを使用する場合は、計測器は\*RSTデフォルト値にプログラムされます。MINimumパラメータを使用する場合は、計測器は最小許容値にプログラムされます。MAXimumパラメータを使用する場合は、計測器は最大許容値にプログラムされます。  
:ARM:TIMer 0.1 タイマを100msecに設定せよ。  
:ARM:TIMer DEFault タイマを0.1secに設定せよ。  
:ARM:TIMer MINimum タイマを1msecに設定せよ。  
:ARM:TIMer MAXimum タイマを99999.99secに設定せよ。
- <numlist>** Numlist—リストのために1個以上の演算を指定してください。例  
:STATus:QUEue:ENABle (-110:-222) -110から-222までのエラーを使用可能にします。
- <NDN>** 非10進数—このパラメータを使用して2進、8進、または16進フォーマットの値を送ります。プレフィックスは、フォーマットの種類を表します。
- #Bxx...X #Bは2進フォーマットを指定します。  
xx...xは2進数です(0と1を使う)。
- #Qxx...X #Qは8進フォーマットを指定します。  
xx...xは8進数です(値は0から7まで)。

#Hxx...X	#H は 16 進フォーマットを指定します。 xx...x は 16 進数です (値は 0 から 9 までと A から F まで)。
2 進値の 36 を非 10 進フォーマットで送る例	
*ESE#b100100	2 進フォーマット
*ESE#q44	8 進フォーマット
*ESE#h24	16 進フォーマット

アングルブラケット<>ーアングルブラケット(<>) は、パラメータの種類を表示するために使います。ブラケットをプログラムメッセージに含めないでください。例：

```
:OUTPut <b>
```

ここで<b> は、ブール型パラメータが要求されることを示します。したがって、選択したソースを使用可能にするには、次のようにコマンドをパラメータ ON または 1 とともに送る必要があります。

```
:OUTPut ON
```

```
:OUTPut 1
```

## 照会コマンド

この種類のコマンドは、現在プログラムされているステータスを要求照会します。このコマンドを識別するために、コマンドの基本形の最後に、疑問符をつけます。ほとんどのコマンドが照会形を持っています。例

```
:ARM:TiMer?          タイマインターバルを照会せよ。
```

数値パラメータ (<n>) を要求するほとんどのコマンドは、また、DEFAult、MINimum、MAXimum も照会形として使用することがあります。これらの照会形を使用するのは、基本コマンドの\*RST デフォルト値と、上限、下限を決めるためです。例

:ARM:TiMer? DEFAult	*RST デフォルト値を照会せよ。
:ARM:TiMer? MINimum	最小許容値を照会せよ。
:ARM:TiMer? MAXimum	最大許容値を照会せよ。

## 大文字と小文字の識別

コモンコマンドと SCPI コマンドは、大文字と小文字を識別しません。大文字を使うことも、小文字を使うことも、あるいは組合せて使うこともできます。例

*RST	= *rst
:DATA?	= :data?
:SYSTem:PRESet	= :system:preset

**注記**     すべて大文字を使用すると、コマンド応答時間がいくらか短くなります。

## ロングフォームバージョンとショートフォームバージョン

SCPI コマンド語は、ロングフォームバージョンでもショートフォームバージョンでも送ることができます。第18部のコマンドサブシステム表は、ロングフォームバージョンを使ってコマンドを示します。ただし、ショートフォームバージョンは、大文字で表示されます。例

*SYSTem:PRESet	ロングフォーム
*SYST:PRES	ショートフォーム
*SYSTem:PRES	ロングフォームとショートフォームの組合せ

それぞれのコマンド語は、ロングフォームか、ショートフォームか、どちらかでなければならぬことに留意してください。その中間ということはありません。たとえば、:SYSTem:PRESe はイリーガルであり、エラーを発生させます。そのコマンドは実行されません。

## ショートフォームのルール

下記のルールに従って、SCPI コマンドのショートフォームバージョンを決めてください。

- ・ コマンド語の長さが4文字以下であれば、ショートフォームバージョンは存在しません。  
:auto = :auto

以下のルールは5文字以上のコマンド語に適用されます。

- ・ コマンド語の4番目の文字が母音("y"を含む)であれば、その文字とそのあとのすべての文字を削除してください。例  
:immediate = :imm
- ・ コマンド語の4番目の文字が子音であれば、その文字は残し、その後ろのすべての文字を落としてください。例  
:format = :form
- ・ コマンドに疑問符(?, 照会)か、コマンド語に含まれる非オプション数がある場合には、それをショートフォームバージョンに含めなければなりません。例  
:delay? = :del?
- ・ ブラケット ([]) で囲まれるコマンド語または文字はオプションですから、プログラムメッセージに含める必要はありません。

**注記** コマンドに対する応答を速めるには、必ずショートフォームを使ってください。

## プログラムメッセージ

プログラムメッセージを設定するのは、コンピュータから計測器に送られた1語以上のコマンド語です。それぞれのコモンコマンドは、アスタリスク (\*) を先頭とする3文字の頭文字です。SCPI コマンドは、:STATus サブシステムに分類され、プログラムメッセージを公式化するためにコマンド語がどのように設定されるかを説明するのに使われます。

:STATus	パス (ルート)
:OPERation	パス
*ENABle <NRf>	コマンドとパラメータ
*ENABle?	照合コマンド
:PRESet	コマンド

### 単一コマンドメッセージ

上記のコマンド構造には、3段のレベルがあります。最初のレベルは、ルートコマンド (:STATus) で設定され、パスとして使用されます。第2のレベルは、別のパス (:OPERation) と1つのコマンド (:PRESet) で設定されます。第3のパスは、パス \*OPERation のための1つの命令で設定されます。この構造の中の3個のコマンドを実行するには、次のような3個の独立したプログラムメッセージを送ります。

```
:stat:oper:enab <NRf>
:stat:oper:enab?
:stat:pres
```

上記のプログラムメッセージのそれぞれにおいて、パスポイントはルートコマンド (:stat) から出発し、コマンドレベルを下に進み、最後にコマンドが実行されます。

### 複数コマンドメッセージ

複数のコマンドメッセージがセミコロン (;) で分離してあれば、同じプログラムメッセージの中でこれらのメッセージを送ることができます。以下に示すのは、1個のプログラムメッセージの中の2個の命令です。

```
:stat:oper;:stat:oper:enab <NRf>
```

上記のメッセージを送信すると、最初のコマンド語はルートコマンド (:stat) として認識されます。次のコロンが検出されると、パスポイントは下の方の次のコマンドレベルに移行し、コマンドを実行します。パスポイントがセミコロン (;) の後ろにコロンを見れば、パスポイントはルートレベルにリセットし、もう一度ははじめから動作を開始します。

同じコマンドレベルにあるコマンドは、完全なコマンドパスを再度キー入力しなくても実行することができます。例

```
:stat:oper:enab <NRf>;enab?
```

最初のコマンド (:enab) が実行されたあとは、バスポインタはコマンド構造の第3レベルにあります。:enab?も第3レベルにあるので、完全なバスネームを反復しなくても、これをキー入力することができます。enab?の先頭のコロンが、プログラムメッセージには含まれていないことに注目してください。もしコロンが含まれていれば、バスポインタはルートレベルにリセットして、ルートコマンドが来るのを待つことになるでしょう。:enab?はルートコマンドではないので、エラーが起こるでしょう。

### コマンドパスに関するルール

- ・ ルートコマンドがオプションコマンド (たとえば[:SENSe]) でない場合は、それぞれの新しいプログラムメッセージは、ルートコマンドで始まらなければなりません。ルートがオプションコマンドの場合は、次のレベルのコマンド語をルートとして扱えば十分です。
- ・ プログラムメッセージの頭のコロン (:) はオプションであり、必ずしも使用する必要はありません。例  
:stat:pres = stat:pres
- ・ バスポインタは、コロン (:) を検出すると、次のコマンドレベルに下がります。例外は、バスポインタがセミコロン (;) を検出したときです。セミコロンはプログラムメッセージ内の複数のコマンドを分離するために使用します (次のルールを参照)。
- ・ バスポインタは、セミコロン (;) の直後のコロン (:) を検出すると、ルートレベルに戻り、リセットします。
- ・ バスポインタは、下への移動だけが可能です。バスポインタは、上のレベルに移動することはできません。上のレベルでコマンドを実行するためには、もう一度ルートコマンドから始める必要があります。

### 同じメッセージの中でコモンコマンドと SCPI メッセージを使う

コモンコマンドと SCPI コマンドがセミコロン (;) で分離してあれば、同じメッセージの中でこれらのコマンドを使用することができます。コモンコマンドはどのコマンドレベルでも実行することができ、バスポインタに影響を与えません。例

```
:stat:oper:enab <NRf>;*ESE<NRf>
```

### プログラムメッセージターミネータ (PMT)

それぞれのプログラムメッセージは、LF (line feed ラインフィード)、EOI (end or identify 終了または識別)、または LF+EOI で終止しなければなりません。いま使用しているコンピュータがこの終止符を与えなければ、バスは立ち往生します。下記の例は、複数コマンドプログラムメッセージをどのようにして終止しなければならないか、その方法を示します。

```
:outp on <PMT>
```

## コマンド実行に関するルール

- ・ コマンドを実行する順序は、プログラムメッセージの中でコマンドが現れる順序です。
- ・ 無効コマンドはエラーを発生します。実行されないのはもちろんです。
- ・ 複数コマンドメッセージの中で、無効コマンドに先行する有効コマンドは実行されます。
- ・ 複数コマンドメッセージの中で、無効コマンドに続く有効コマンドは無視されます。

## 応答メッセージ

応答メッセージとは、照会コマンドプログラムメッセージに対する応答として計測器がコンピュータに送ったメッセージです。

### 応答メッセージを送る

照会メッセージを送ったあと、応答メッセージが出力待ち行列に置かれます。続いて 2400 型は talk するように呼びかけられ、応答メッセージが出力待ち行列からコンピュータに送られます。

### 複数応答メッセージ

同じプログラムメッセージで 2 個以上の照会コマンドを送る場合（「複数コマンドメッセージ」の項を参照）には、2400 型が talk するように呼びかけられると、すべての照会についての複数応答メッセージがコンピュータに送られます。応答は、照会コマンドを送信した順序で送られ、セミコロン (;) で分離されます。同じ照会の中の複数の項目は、コンマ (,) で分離されます。下記に示す例は、4 個の単一項目照会コマンドで設定されるプログラムメッセージに対する応答メッセージです。

0;1;1;0

### 応答メッセージターミネータ (RMT)

それぞれの応答は、LF (line feed ラインフィード) と BOI (end or identify 終止または識別) で終止します。下記に示す例は、複数応答メッセージの終止方法です。

0;1;1;0 <RMT>



## メッセージ交換プロトコル

メッセージ交換プロトコルは、次の2つのルールに要約されます。

ルール1. 2400型に、何をコンピュータに送るかを必ず命令しなければなりません。

計測器からコンピュータに情報を送るには、下記の2つのステップを必ず実行しなければなりません。

1. 関連する照会コマンドを1つのプログラムメッセージで送る。
2. 2400型にtalkするように呼びかける。

ルール2. まずコンピュータが全部の応答メッセージを受けてから、ほかのプログラムメッセージを2400型に送ってください。

## RS-232のインタフェース動作

**注記** RS-232に関してプログラム可能な項目（ボーレート、データビット、パリティ、ターミネータ）は、メインメニューのCOMMUNICAIONオプションを使って設定することができます（第1部の「メインメニュー」を参照）。

### データの送受

RS-232インタフェースは、8個のデータビット、1個のストップビット、ノーパリティを使用して、データを転送します。これから接続しようとするデバイスもこのような設定を使用することを確認してください。

^C (decimal 3) または ^X (decimal 18) で設定される文字列をメーターに送ることにより、データ伝送を中断することができます。これは、未実行の動作をすべてクリアし、未出力の出力をすべて廃棄します。

### ボーレート

ボーレートとは、2400型とプログラミング端末が通信を行う速度を指します。以下の使用可能な通信速度から1つを選んでください。

- ・ 57600
- ・ 38400
- ・ 19200
- ・ 9600
- ・ 4800
- ・ 2400
- ・ 1200
- ・ 600
- ・ 300

工場出荷時の選択ボーレートは9600です。

ボーレートを選ぶときには、2400型に接続するプログラミング端末またはプリンタが、選択したボーレートをサポートできることを確認してください。2400型もほかのデバイスも、同一ボーレートに合わせて機器設定しなければなりません。

## データビットとパリティ

RS-232 インタフェースは、偶数パリティまたは奇数パリティを使って、またはパリティなしで、長さ7または8ビットのデータを送受するように設定することができます。

## ターミネータ

2400 型の設定を変更して、コントローラに伝送するそれぞれのプログラムメッセージを、下記に示す <CR> と <LF> の組合せのどれかを使って終了させることができます。

<CR>	キャリッジリターン
<CR+LF>	キャリッジリターンとラインフィード
<LF>	ラインフィード
<LF+CR>	ラインフィードとキャリッジリターン

## フロー制御（信号ハンドシェーキング）

コントローラと計測器との間の信号ハンドシェーキングを使えば、データ受信の準備ができているかどうかについて、2 個のデバイスに相互に通信させることができます。2400 型は、ハードウェアのハンドシェーキング（フロー制御）をサポートしません。

ソフトウェアのフロー制御は、XON と XOFF という文字の形をとり、RS-232 の FLOW CONTROL メニューから XON-XOFF が選択されると、使用可能になります。2400 型の入力待ち行列が 3/4 以上満たされると、計測器は XOFF コマンドを出します。制御プログラムはこれに応答して、2400 型が XON コマンドを出すまで、文字の送り出しを中止する必要があります。入力バッファのレベルが半分以下まで下がれば、2400 型は直ちに XON を出します。2400 型は、コントローラが送った XON と XOFF を認識します。XOFF は、2400 型が XON を受けるまで、文字の出力を中止させます。到来コマンドが処理されるのは、コントローラから <CR> 文字を受けてからになります。

NONE というフロー制御を選択すれば、コントローラと 2400 型との間には信号ハンドシェーキングは起こりません。受信側デバイスの準備が整う前に伝送されたデータは、失われます。

## RS-232 の接続

RS-232 シリアルポートをコンピュータまたはリスニングデバイス（すなわちシリアルプリンタ）のシリアルポートに接続するために、DB-9 コネクタを端末とするストレート接続 RS-232 ケーブルを使用します。ヌルモデムケーブルは、使用しないでください。シリアルポートは、RS-232 規格の、送信ライン（TXD）、受信ライン（RXD）、信号接地（GND）を使用します。図 4-16 には RS-232 用の背面パネルコネクタを示し、表 4-3 はコネクタのピン接続を示します。

ご使用のコンピュータの RS-232 インタフェース用のコネクタが DB-25 であれば、一方の端末には DB-25 コネクタを、他方の端末には DB-9 コネクタをストレート接続（ヌルモデムではありません）したケーブルまたはアダプタを使用する必要があります。

図 14-4  
RS-232 インタフェース  
コネクタ

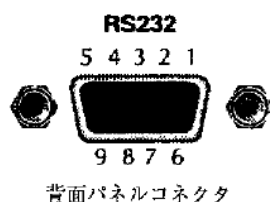


表 14-2  
RS-232 コネクタのピン接続

ピン番号	用途
1	使用しません
2	RXD、データ送信
3	TXD、データ受信
4	使用しません
5	GND、信号接地
6	使用しません
7	RTS、クリアして送信
8	CTS、送信準備完了
9	使用しません

注記 CTS と RTS はいっしょにくくられています。

表 14-3 に示すのは、9 ピン (DB-9) または 25 ピン (DB-25) シリアルポートコネクタをコンピュータ (PC) で使用する場合の、ピン接続識別です。

表 14-3  
PC シリアルポートピン接続

信号	DB-9 ピン番号	DB-25 ピン番号
DCD、データキャリア検出	1	8
RXD、データ受信	2	3
TXD、データ送信	3	2
DTR、データ端末準備完了	4	20
GND、信号接地	5	7
DSR、データセット準備完了	6	6
RTS、送信リクエスト	7	4
CTS、クリアして送信	8	5
RI、リングインディケータ	9	22

## エラーメッセージ

RS-232 のエラーメッセージについては、付録 B を参照してください。

## プログラミングの例

下記のクイックペーシック 4.5 のプログラミングの例は、RS-232 COM2 ポートを経由してソース・メータを制御します。前面パネルメインメニューからの操作で、ソース・メータを RS-232 モードに入れてください (MENU を押し、COMMUNICATION を選択し、RS-232 を選択してください)。通信設定を変更したときは、ソース・メータはリセットしてこの時のモードに入ります。

```
RD$ = SPACE$ (1500)      'ストリングスペースを設定せよ。
CLS                      'スクリーンをクリアせよ。

"Set COM2 baud rate to 9600"
"Set no flow control, and CR as terminator"

'シリアルポートパラメータを設定せよ。
'下記の値はソース・メータのデフォルト設定値です。
ComOpen$ = "COM2: 9600, N, 8, 1, ASC, CD0, CS0, DS0, LF, OP0, RS, TB8192, RB192"
OPEN ComOpen$ FOR RANDOM AS #1

'ソース・メータセットアップコマンド
PRINT #1, "*RST"          '計測器をデフォルトパラメータにリセットせよ。
PRINT #1, "SENS:FUNC RES" '抵抗測定機能を選択せよ。
PRINT #1, "SENS:RES:NPLC 1" '測定速度を 1PLC に設定せよ。
PRINT #1, "SENS:RES:MODE MAN" '手動抵抗測定モードを選択せよ。
PRINT #1, "SOUR:FUNC CURR" '電流ソース機能を選択せよ。
PRINT #1, "SOUR:CURR 0.01" '10mA を出力できるようにソースを設定せよ。
PRINT #1, "SOUR:CLE:AUTO ON" 'ソースオート出力オフを有効にせよ。
PRINT #1, "SENS:VOLT:PROT 10" '10V のコンプライアンスリミットを設定せよ。
PRINT #1, "TRIG:COUN 1" '1 つの測定を行うように設定せよ。
PRINT #1, "FORM:FLEM RES" '抵抗読み取り値を PC に出力するように設定せよ。

'1 回の読み取りを開始し、結果を印刷せよ。
PRINT #1, "READ?"        'トリガを行い 1 つの読み取り値を取得せよ。
LINE INPUT #1, RD$
RD# = "Resistance: " + RD$
PRINT RD$

'クリーンアップを行い、終了せよ。
終了:
CLOSE #1                  'ファイルを閉じよ。
CLEAR                    'インタフェースクリア
END
```