

4. リモート・プログラミング

4.1 GPIB リモート・プログラミング

本器は、IEEE 規格 488.1-1978 に準拠した GPIB (General Purpose Interface Bus) を標準装備し、外部コントローラによるリモート・コントロールが可能です。

4.1.1 GPIB とは

GPIB は、コンピュータと測定器を統合する高性能のバスを提供します。

この GPIB の動作は IEEE 規格 488.1-1978 によって定義されています。GPIB はバス構造のインタフェースのため、各機器に固有の機器アドレスを持たせることによって、機器を指定します。これらの機器は 1 つのバスに 15 台まで並列に接続できます。GPIB 機器は、以下の機能のうち 1 つ以上を備えています。

- トーカ： バスにデータを送信するために指定された機器を「トーカ」と呼びます。GPIB バス上では、一台の機器のみがアクティブ・トーカとして動作します。
- リスナ： バスのデータを受信するために指定された機器を「リスナ」と呼びます。アクティブなリスナ機器は、GPIB バス上に複数存在することができます。
- コントローラ： トーカ、リスナを指定する機器を「コントローラ」と呼びます。GPIB バス上では一台の機器のみがアクティブ・コントローラとして動作します。これらのコントローラのうち、IFC、および REN のメッセージをコントロールできる機器を特に「システム・コントローラ」と呼びます。

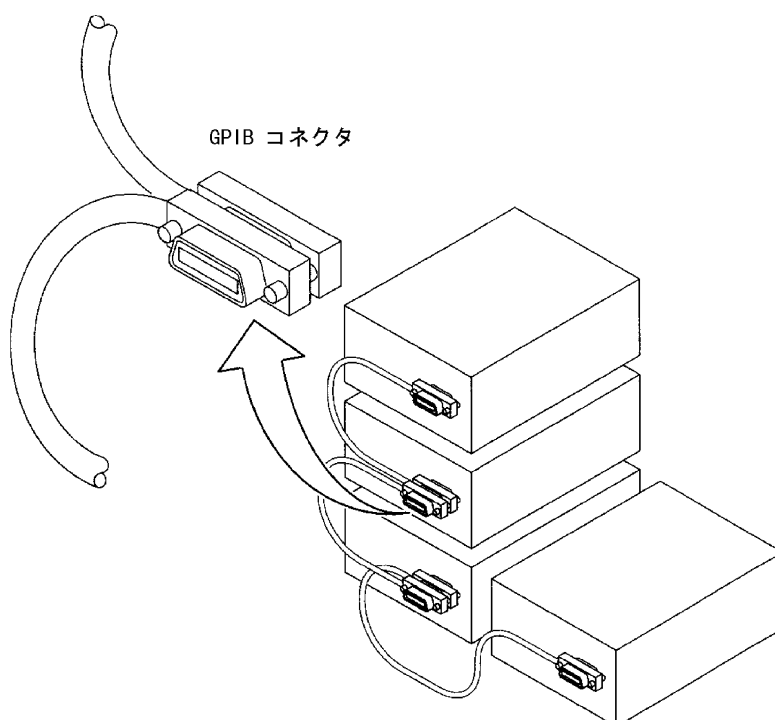
システム・コントローラは、GPIB バス上に一台だけ許されます。バス上に複数のコントローラがある場合、システム起動時にはシステム・コントローラがアクティブ・コントローラとなり、その他のコントローラ能力を持つ機器はアドレスサブル機器として動作します。その他のコントローラをアクティブ・コントローラにするには Take Control (TCT) インタフェース・メッセージを用います。そのとき自分はノンアクティブ・コントローラとなります。コントローラはインタフェース・メッセージ、またはデバイス・メッセージを各測定器に送ってシステム全体をコントロールします。それぞれ以下の役目を果たします。

- インタフェース・メッセージ： GPIB バスをコントロールします。
- デバイス・メッセージ： 測定器をコントロールします。

4.1.2 GPIB のセット・アップ

(1) GPIB の接続

以下に標準的な GPIB の接続を示します。GPIB コネクタは 2 本のねじでしっかり固定して、使用中にゆるむことがないように注意して下さい。



GPIB インタフェースの使用時には、以下のことに注意して下さい。

- 1つのバス・システムで使われる GPIB ケーブルの全ケーブル長は、20m 以下かつ、 $2\text{m} \times$ 接続される機器の数以下です。GPIB コントローラも 1つの機器として数えます。
 - 1つのバス・システムに接続できる機器の数は、最高 15 台です。
 - ケーブル間の接続方法には制限はありません。ただし、1 台の機器上に 4 個以上の GPIB コネクタを重ねないで下さい。4 個以上重ねるとコネクタの取り付け部に過度の力が加わり、破損することがあります。
- (例) 5 台の機器から構成されるシステムで使用できる全ケーブル長は、10m 以下 ($5 \text{台} \times 2\text{m}/\text{台} = 10\text{m}$) です。全ケーブル長が許容最大長を超えない範囲で、自由に分配することができます。ただし、10 台以上の機器を接続する場合は、何台かの機器を 2m 以下のケーブルで接続して、全ケーブル長が 20m を超えないようにする必要があります。

(2) GPIB アドレスの設定

1. **CONFIG, GPIB** と押します。
GPIB メニューが表示されます。
2. **Address** と押します。
アクティブ・エリアに現在の GPIB アドレスが表示されます。
3. データ・ノブ、ステップ・キーまたはテン・キーで、本器の GPIB アドレスを入力します。
4. **ENTER (Hz)** を押して、アドレスを設定します。

(3) 表示の Off

リモート・コントロール時、画面表示を OFF にすると高速な測定を行うことができます。

1. **CONFIG, GPIB** と押します。
GPIB メニューが表示されます。
2. **Display ON/OFF** を押します。
OFF に設定され、トレース以外の表示が消去されます。

4.1.3 GPIB インタフェース機能

コード	説明
SH1	ソース・ハンドシェーク機能あり
AH1	アクセプタ・ハンドシェーク機能あり
T6	基本的トーカー機能、シリアル・ポール機能、リスナ指定によるトーカー解除機能
TE0	拡張トーカー機能なし
L4	基本的リスナ機能、トーカー指定によるリスナ解除機能
LE0	拡張リスナ機能なし
SR1	サービス・リクエスト機能あり
RL1	リモート機能、ローカル機能、ローカル・ロック・アウト機能
PP0	パラレル・ポール機能なし
DC1	デバイス・クリア機能
DT0	デバイス・トリガ機能なし
C0	システム・コントローラ機能なし
E1	オープン・コレクタ・バス・ドライバを使用

4.1.4 インタフェース・メッセージに対する応答

この項で説明するインタフェース・メッセージに対する本器の応答は、IEEE 規格 488.1-1978 で定義されています。

インタフェース・メッセージの本器への送り方は、使用するコントローラの手取扱説明書を参照して下さい。

(1) インタフェース・クリア (IFC)

このメッセージは、本器へ直接信号線で送られてきます。

このメッセージによって本器は GPIB バスの動作を停止します。すべての入/出力を停止しますが、入出力バッファはクリアされません(クリアは DCL で実行される)。

(2) リモート・イネーブル (REN)

このメッセージは、本器へ直接信号線で送られてきます。このメッセージが真のとき、本器がリスナに指定されるとリモート状態になります。この状態は GTL を受けとるか、REN が偽になるか、または LOCAL キーを押すまで続きます。本器は、ローカル状態のとき、すべての受信データを無視します。

リモート状態のとき、LOCAL キーを除くすべてのキー入力を無視します。ローカル・ロック・アウト状態のとき、すべてのキー入力を無視します。

(3) シリアル・ポール・イネーブル (SPE)

本器はこのメッセージを外部から受信すると、シリアル・ポール・モードになります。このモードでは、トーカーに指定されると通常のメッセージではなくステータス・バイトを送信します。このモードはシリアル・ポール・ディセーブル (SPD) メッセージを受信するか、IFC メッセージを受信するまで続きます。

本器がサービス・リクエスト (SRQ) メッセージをコントローラに送信しているときには、応答データの bit6 (RQS bit) が 1 (TRUE) になります。送信が終了後、RQS bit は 0 (FALSE) になります。サービス・リクエスト (SRQ) メッセージは、直接信号線で送ります。

(4) デバイス・クリア (DCL)

本器は DCL を受け取ったときに、以下のことを実行します。

- 入力バッファと出力バッファのクリア
- 構文解析部、実行コントロール部、応答データ生成部のリセット
- 次に実行するリモート・コマンドを妨げる全コマンドのキャンセル
- 他のパラメータを待つため一時停止されているコマンドのキャンセル

以下のことは実行しません。

- 本器に設定または格納されているデータの変更
- 正面パネル操作の中断
- 実行中の本器の動作への影響や中断
- MAV を除くステータス・バイトの変更 (MAV は出力バッファのクリアの結果として 0 になる)

(5) セレクテッド・デバイス・クリア (SDC)

DCL と同一の動作を行います。ただし、SDC は本器がリスナるときだけ実行されます。その他の場合は無視されます。

(6) ゴー・トゥ・ローカル (GTL)

このメッセージは、本器をローカル状態にします。ローカル状態になると、正面パネル操作がすべて有効になります。

(7) ローカル・ロック・アウト (LLO)

このメッセージは、本器をローカル・ロック・アウト状態にします。この状態で本器がリモート状態になると、正面パネル操作はすべて禁止されます (通常のリモート状態では、**LOCAL** キーで正面パネル操作ができる)。

このとき本器をローカル状態にする方法は、以下の3通りあります。

- GTL メッセージを本器に送る
- REN メッセージを偽にする (このときローカル・ロック・アウト状態も解除される)
- 電源を再投入する

4.1.5 メッセージ交換プロトコル

本器は、コントローラやその他の機器から GPIB バスを通じてプログラム・メッセージを受け取り、応答データを発生します。プログラム・メッセージには、コマンド、クエリ (応答データを問い合わせるコマンドのことを、特に「クエリ」と呼ぶ)、データが含まれています。これらのデータのやりとりには手順があります。ここではその手順について説明します。

(1) GPIB 各種バッファ

本器には、以下の2つバッファがあります。

(a) 入力バッファ

コマンド解析をするために一時的にデータを貯めておくバッファです (1024 バイトの長さをもちますが、それ以上の入力は無視されます)。

入力バッファのクリア方法は、2通りあります。

- 電源投入
- DCL または SDC の実行

(b) 出力バッファ

コントローラからデータを読まれるまでデータを貯めておくバッファです (1024 バイトの長さをもつ)。

出力バッファのクリア方法は、2通りあります。

- 電源投入
- DCL または SDC の実行

4.1 GPIB リモート・プログラミング

(2) メッセージ交換

この他のコントローラや機器がメッセージを本器から受信するときに、特に重要な項目であるを、クエリの受信と応答データの生成を以下に説明します。

(a) パーサー

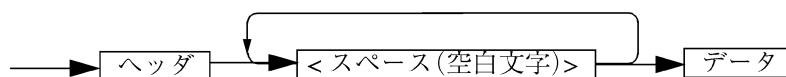
入力バッファから受信した順序通りにコマンド・メッセージを受け取り、構文解析を実行し、受け取ったコマンドがどんな内容の実行を行うのかを決定します。

(b) 応答データ生成

本器はパーサーがクエリを実行すると、その応答としてデータを出力バッファ上に生成します（つまりデータを出力するにはその直前に必ずクエリを送る必要がある）。

4.1.6 コマンド文法

コマンド文法は、以下のフォーマットで定義されています。



(1) ヘッダ

ヘッダには、共通コマンド・ヘッダと単純ヘッダがあります。共通コマンド・ヘッダは、ニーモニックの先頭にアスタリスク(*)を付けたものです。

単純ヘッダは、階層構造を持たない、機能的に独立した命令です。

ヘッダの直後に?を付けるとクエリ・コマンドになります。

(2) スペース (空白文字)

1文字分以上のスペースが可能です（スペースを省略しても構いません）。

(3) データ

コマンドが複数のデータを必要とするときは、データをカンマ(,)で区切って複数並べます。カンマ(,)の前後にスペース(空白文字)を入れても構いません。データ・タイプの詳細については、[4.1.7 データ・フォーマット]を参照して下さい。

(4) 複数のコマンドの記述

本器は、複数のコマンドをセミコロン(;)で区切って1行で記述することが可能です。

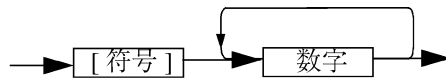
4.1.7 データ・フォーマット

本器は、ここで示すデータ・タイプをデータの入出力で使用します。

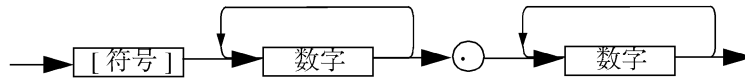
(1) 数値データ

数値データには以下の3つのフォーマットがあり、本器に対する数値の入力では、どれを用いても構いません。また、コマンドによっては入力時に単位を付けられます。

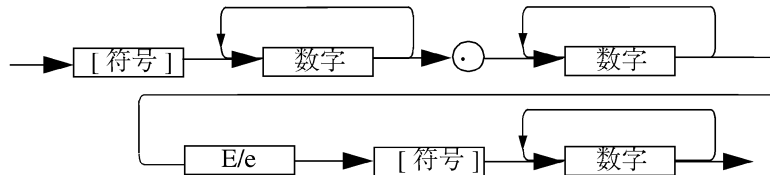
- 整数型：NR1 フォーマット



- 固定小数点型：NR2 フォーマット



- 浮動小数点型：NR3 フォーマット



(2) 単位

使用可能な単位の一覧を以下に示します。

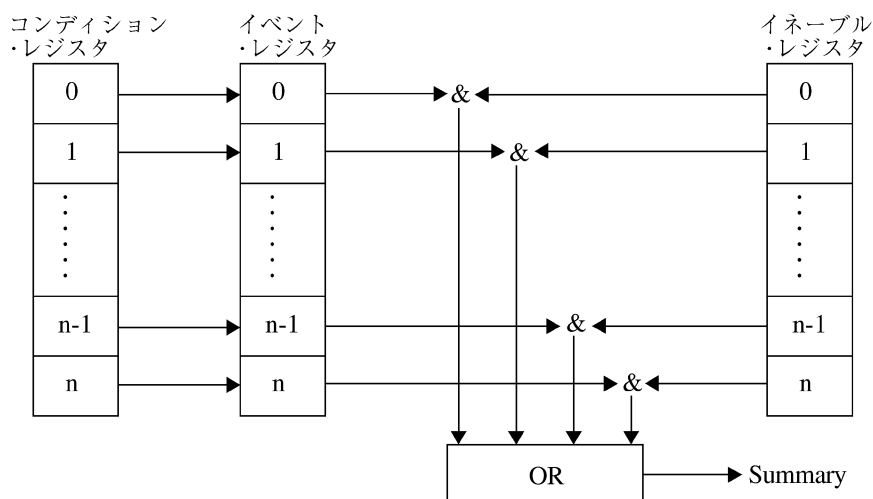
単位	指数	意味
GZ	10^9	周波数
MZ	10^6	周波数
KZ	10^3	周波数
HZ	10^0	周波数
VOLT	10^0	電圧
MV	10^{-3}	電圧
UV	10^{-6}	電圧
NV	10^{-9}	電圧
MW	10^{-3}	電力
DB	10^0	dB 関連
MA	10^{-3}	電流
SC	10^0	秒
MS	10^{-3}	秒
US	10^{-6}	秒
PER	10^0	パーセント
%	10^0	パーセント

4.1.8 ステータス・バイト

本器では IEEE 規格 488.2-1987 に適合した階層化されたステータス・レジスタ構造をもち、機器の様々な状態をコントローラへ送信できます。ここではこのステータス・バイトの動作モデルと、イベントの割当を説明します。

(1) ステータス・レジスタ

本器は、IEEE 規格 488.2-1987 で定義されたステータス・レジスタのモデルを採用し、コンディション・レジスタ、イベント・レジスタ、イネーブル・レジスタから構成されています。



(a) コンディション・レジスタ

コンディションレジスタは、機器のステータスを常に監視しています。つまり、このレジスタには常に最新の機器のステータスが保持されています。ただし、コンディション・レジスタは内部情報として保持しているため、データの読み書きはできません。

(b) イベント・レジスタ

イベント・レジスタは、コンディション・レジスタからのステータスをラッチして保持します(変化を保持する場合もある)。このレジスタがセットされると、クエリで読み出されるか、*CLS でクリアされるまでセットされたままです。イベント・レジスタにデータを書き込むことはできません。

(c) イネーブル・レジスタ

イネーブル・レジスタは、イベント・レジスタのどのビットを有効なステータスとしてサマリを生成するのか指定します。イネーブル・レジスタはイベント・レジスタと AND をとられ、その結果の OR がサマリとして生成されます。サマリはステータス・バイト・レジスタに書き込まれます。イネーブル・レジスタはデータを書き込めます。

本器のステータス・レジスタは、以下の3種類があります。

- ステータス・バイト・レジスタ
- スタンダード・イベント・レジスタ
- スタンダード・オペレーション・ステータス・レジスタ

本器のステータス・レジスタの配置を図4-1に示します。

ステータス・レジスタの詳細を図4-2に示します。

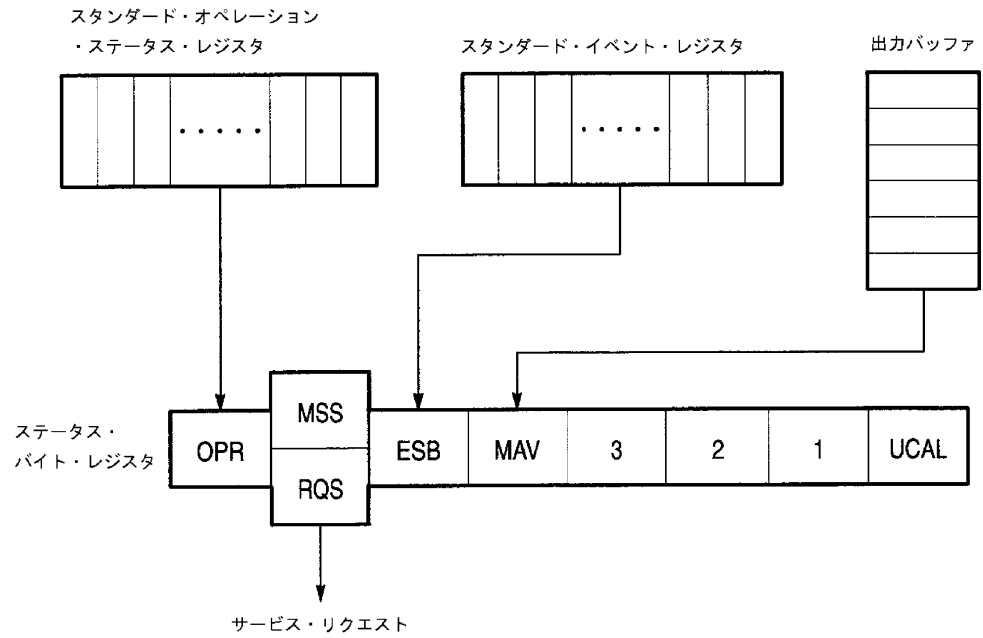


図 4-1 ステータス・レジスタの配置

4.1 GPIB リモート・プログラミング

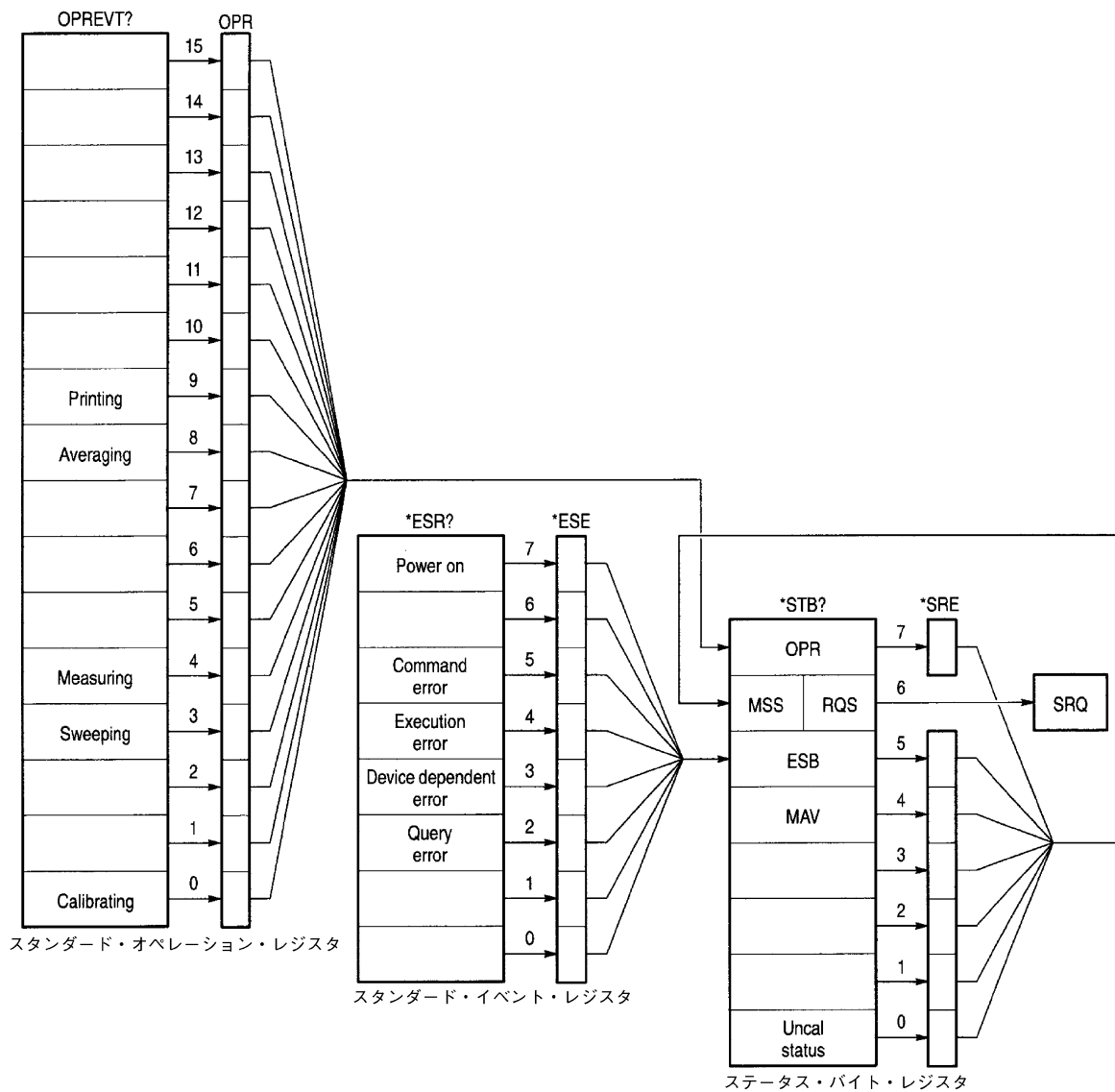


図 4-2 ステータス・レジスタの詳細

(2) イベント・イネーブル・レジスタ

各イベント・レジスタには、どのビットを有効にするかを決定するイネーブル・レジスタがあります。イネーブル・レジスタは、対応するビットを10進値で設定します。

- サービス・リクエスト・イネーブル・レジスタのセット:*SRE
- スタンダード・イベント・ステータス・イネーブル・レジスタのセット:*ESE
- オペレーション・ステータス・イネーブル・レジスタのセット:OPR

(例) オペレーション・ステータス・レジスタの Measuring ビットのみを有効にします。
オペレーション・ステータス・レジスタの Measuring ビットが1にセットされると、ステータス・バイト・レジスタの OPR ビットが1にセットされます。

PRINT @ 8;"OPR16" (N88BASIC のプログラム例)

OUTPUT 708;"OPR16" (HP200、300 シリーズのプログラム例)

(例) ステータス・バイト・レジスタの OPR (Operation Status Register のサマリ) ビットと ESB (Event Status Register のサマリ) ビットを有効にします。

OPR ビットまたは ESB ビットが1にセットされると、ステータス・バイト・レジスタの MSS ビットが1にセットされます。

PRINT @ 8;"*SRE160" (N88BASIC のプログラム例)

OUTPUT 708;"*SRE160" (HP200、300 シリーズのプログラム例)

(3) スタンダード・オペレーション・ステータス・レジスタ

スタンダード・オペレーション・ステータスのイベント・レジスタの割り当てを、以下に示します。

bit	機能定義	説明
15 ~ 10		常に 0
9	Printing	プリンタ出力終了時に 1 にセットされる。
8	Averaging	アベレージ終了時に 1 にセットされる。
7 ~ 5		常に 0
4	Measuring	シーケンス測定終了時に 1 にセットされる。
3	Sweeping	掃引終了時に 1 にセットされる。
2 ~ 1		常に 0
0	Calibrating	補正データ取得終了時に 1 にセットされる。

4.1 GPIB リモート・プログラミング

(4) ステータス・バイト・レジスタ

ステータス・バイト・レジスタは、ステータス・レジスタからの情報を要約しています。また、このステータス・バイト・レジスタのサマリがサービス・リクエストとしてコントローラに送信されます。そのため、ステータス・バイト・レジスタは、ステータス・レジスタ構造とは若干違った動作を行います。ここではステータス・バイト・レジスタに関して説明をします。

ステータス・バイト・レジスタの構造を、図 4-3 に示します。

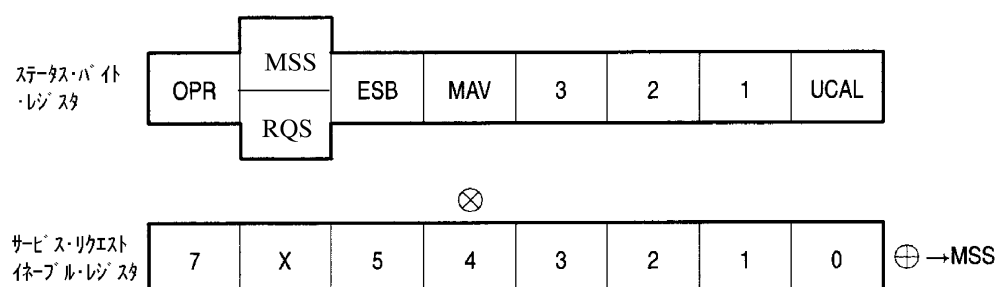


図 4-3 ステータス・バイト・レジスタの構造

このステータス・バイト・レジスタは、以下の3点を除くとステータス・レジスタに従います。

- ステータス・バイト・レジスタのサマリが、ステータス・バイト・レジスタの bit6 に書き込まれます。
- イネーブル・レジスタの bit6 は、常に有効で変更できません。
- ステータス・バイト・レジスタの bit6 (MSS) が、サービス・リクエスト要求の RQS を書き込みます。

このレジスタが、コントローラからのシリアル・ポールに対して応答します。シリアル・ポールに対して応答するときには、ステータス・バイト・レジスタの bit0 ~ 5、bit7 および RQS が読み出され、その後に RQS は 0 にリセットされます。その他のビットはそれぞれの要因が 0 になるまでクリアされません。

ステータス・バイト・レジスタ、RQS、MSS は、“*CLS”、“S2” を実行するとクリアできます。それにとまって、SRQ ラインも偽になります。

ステータス・バイト・レジスタの各ビットの意味を、以下に示します。

bit	機能定義	説明
7	OPR	OPR は、スタンダード・オペレーション・ステータス・レジスタのサマリである。
6	MSS	RQS は、ステータス・バイト・レジスタの MSS が 1 になったとき TRUE になるが、その MSS はすべてのステータス・データ構造のサマリ・ビットになっている。 MSS は、シリアル・ポールでは読めない(ただし、RQS が 1 のときは MSS が 1 であることがわかる)。 MSS を読むには、共通コマンド *STB? を用いる。 *STB? ではステータス・バイト・レジスタの bit0 ~ 5、bit7 および MSS が読み出される。 この場合ステータス・バイト・レジスタと MSS はクリアされない。 MSS は、ステータス・レジスタ構造のすべてのマスクされていない要因がクリアされるまで 0 にならない。
5	ESB	ESB は、スタンダード・イベント・レジスタのサマリである。
4	MAV	出力バッファの要約ビット 本器では、対応していません。
3 ~ 1		常に 0
0	UCAL	掃引が早すぎて信号のレベルに誤差が生じる場合 1 にセットされる。

4.1 GPIB リモート・プログラミング

- (5) スタンダード・イベント・レジスタ
スタンダード・イベント・レジスタの割り当てを、以下に示します。

bit	機能定義	説明
7	Power on	電源投入で 1 になる。
6		常に 0
5	Command Error	パーサーが文法エラーを見つけたときに 1 にセットされる。
4	Execution Error	GPIB コマンドとして受け取った命令の実行を何らかの理由 (パラメータが範囲外など) で失敗すると 1 にセットされる。
3	Device Dependent Error	Command Error、Execution Error、Query Error 以外のエラーが発生したとき 1 にセットされる。
2	Query Error	コントローラが本器からデータを読み出そうとしたときに、データが存在しない、またはデータが消失していると 1 にセットされる。
1	Request Control	本器では、対応していません。
0	Operation Complete	本器では、対応していません。

4.1.9 GPIB コード一覧

GPIB コマンド・リストを機能ごとに示します。

- リスナ・コード欄： * は、コードに続いて数値データを入力するファンクションであることを表します。
/**/ は、コードに続いて文字列データを入力するファンクションであることを表します。
[ON]、[ON,] および 数値データは、省略可能です。
ファイル名、ラベルなどの文字列データは、コマンド直後の文字からデリミタ直前の文字までを入力として受け取ります。ただし、最初の文字が / で始まる場合は、/ と / で挟まれた部分を入力として受け取ります。
- 出力フォーマット欄： , は、複数のデータを出力することを表します。
ON/OFF および Auto/Manual は、それぞれ 1/0 を出力します。
周波数単位は Hz、時間単位は sec で出力します。また、レベル単位は設定されている表示単位で出力します。

表 4-1 周波数コマンド

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
中心周波数	CF *	CF?	Center 周波数
CF ステップ・サイズ	CS *	CS?	CF Step 周波数
CF ステップ・オート	CA	CA?	0:Manual/1:Auto
スタート周波数	FA *	FA?	Start 周波数
ストップ周波数	FB *	FB?	Stop 周波数
周波数スパン	SP *	SP?	Span 周波数
フル・スパン	FS	—	—
ゼロ・スパン	ZS	—	—
ピーク・ズーム	PKZOOM	—	—
ラスト・スパン	LTSP	—	—
	LSPAN	—	—

4.1 GPIB リモート・プログラミング

表 4-2 レベル

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
レファレンス・レベル	RL *	RL?	Reference レベル
アッテネータ	AT*	AT?	ATT レベル
アッテネータ・オート	AA	AA?	0:Manual/1:Auto
入力インピーダンス			
50Ω	OHM50	OHM?	0: 50ohm 1: 75ohm
75Ω	OHM75		
XdB/div	DD *	DD?	0: 10dB 1: 5dB 2: 2dB 3: 1dB
リニア×1	LL1	—	—
レベル・オフセット	RO *,(RON*)	RO?	レベル
ON	RO *,(RON*)	ROON?	0:OFF 1:ON
OFF	ROF		

表 4-3 ユニット

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
表示単位	—	UNIT?	0: dBm
dBm	AUNITS DBM	UN? AUNITS?	1: dBmV
	KSA		2: dBμV
	UB		5: V
	AUNITS DBMV		6: W
dBmV	KSB		
	UM		
dBμV	AUNITS DBUV		
	KSC		
	UU		
Volts	AUNITS V		
	KSD		
Watts	AUNITS W		

表 4-4 掃引モード

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
掃引モード ノーマル	CONTS	SWM?	0:Normal & Full 1:Normal & Window 20:Single & Full 21:Single & Window
	SN		
	SI		
	SNGLS		
リセット & スタート (掃引終了待ち)	TS	---	---
リセット & スタート	SR	---	---

表 4-5 掃引時間

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
掃引時間	SW*	SW?	Sweep Time
	ST*	ST?	Sweep Time
掃引オート	AS	AS?	0:Manual/ 1:Auto
Gate Sig External	ON	GEXON?	0:OFF 1:ON
	OFF		

表 4-6 バンド幅

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
RBW	RB*	RB?	RBW 周波数
RBW オート	BA	BA?	0:Manual/ 1:Auto
VBW	VB*	VB?	VBW 周波数
VBW オート	VA	VA?	0:Manual/ 1:Auto
カップル・オール・オート	AL	AL?	0:Manual/ 1:Auto

4.1 GPIB リモート・プログラミング

表 4-7 マーカ

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
マーカ ON	MN* (※ 1)	MN?	0:OFF 1:Normal 2:Delta
マーカ 周波数	—	MF?	Marker 周波数 (※ 2)
マーカ・レベル	—	ML?	Marker レベル (※ 2)
周波数 + レベル	—	MFL?	MKR 周波数、レベル (※ 2)
マーカ OFF	MKOFF	—	—
	MO	—	—
ノーマル・マーカ (デルタ・マーカ OFF)	MKN*	—	—
	MK*	—	—
デルタ・マーカ ON	MKD*	—	—
	MT*	—	—
ノーマル・マーカ絶対値 周波数	—	MDF1?	Normal MKR 周波数
レベル	—	MDL1?	Normal MKR レベル
デルタ・マーカ絶対値 周波数	—	MDF2?	Delta MKR 周波数
レベル	—	MDL2?	Delta MKR レベル
フィックスド・デルタ・マーカ ON	FXN*	FX?	0:OFF
OFF	FXF		1:ON
マーカ・ステップ・サイズ	MPM*	MPM?	周波数
マーカ・ステップ・オート	MPA	MPA?	0:Manual/ 1:Auto
シグナル・トラック	ON	SGN	0:OFF 1:ON
	OFF	SGF	
カップル	ON	CPLMK ON	0:OFF 1:ON
	OFF	CPLMK OFF	
マルチ・マーカ	ON OFF	MLT ON MLT OFF	MLT? 0:OFF 1:ON
マルチ・マーカ No1	ON	MLN1 * (※ 3) MLF1	— —
	OFF		
マルチ・マーカ No2	ON	MLN2 * (※ 3) MLF2	— —
	OFF		
マルチ・マーカ No3	ON	MLN3 * (※ 3) MLF3	— —
	OFF		
マルチ・マーカ No4	ON	MLN4 * (※ 3) MLF4	— —
	OFF		
マルチ・マーカ No5	ON	MLN5 * (※ 3) MLF5	— —
	OFF		

表 4-7 マーカ (続き)

ファンクション	リスナコード	トーカー・リクエスト	
		クエリ	出力フォーマット
マルチ・マーカ No6	ON OFF	MLN6 * (※ 3) MLF6	— —
マルチ・マーカ No7	ON OFF	MLN7 * (※ 3) MLF7	— —
マルチ・マーカ No8	ON OFF	MLN8 * (※ 3) MLF8	— —
マルチ・マーカ No9	ON OFF	MLN9 * (※ 3) MLF9	— —
マルチ・マーカ No10	ON OFF	MLN10 * (※ 3) MLF10	— —
マルチ・マーカ周波数	—	MLSF?	n<DLM> f1<DLM>… (※ 4)
マルチ・マーカ・レベル	—	MLSL?	n<DLM> l1<DLM>… (※ 5)
ピーク・リスト 周波数 レベル	PLS FREQ PLS LEVEL	— —	— —
ピーク・リストのクエリ	—	PKLST?	n<DLM> f1,l1<DLM>… (※ 6)

- (※ 1) デルタ・モード時は、ノーマル・マーカとの周波数差を入力して下さい。
- (※ 2) デルタ・モード時は、周波数差またはレベル差となります。
- (※ 3) 数値データを入力しないときは、アクティブ・マーカの番号選択として動作します。
- (※ 4) n = 11 固定 fn = 周波数 10 個分 + Δ MKR、<DLM> = デリミタ
- (※ 5) n = 11 固定 ln = レベル 10 個分 + Δ MKR、<DLM> = デリミタ
- (※ 6) n = Peak の個数 fn,ln = 周波数 (時間)、レベル <DLM> = デリミタ

4.1 GPIB リモート・プログラミング

表 4-8 ピーク・サーチ

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
ピーク・サーチ	MKPK	——	——
	PS	——	——
ネクスト・ピーク	NXP	——	——
ネクスト・ピーク・レフト	NXL	——	——
ネクスト・ピーク・ライト	NXR	——	——
Min サーチ	MIS	——	——
Max-Min サーチ	MMS	——	——
連続ピーク ON	CPN	CP?	0:OFF
	OFF		CPF
ピーク ΔY div	DY*	DY?	ΔY (実数)
ピーク範囲 ノーマル	PSN	PKRNG?	0:ALL
	上側		1:Up
	下側		2:Low

表 4-9 サウンド

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
サウンド ON (AM または FM)	SON	SD?	0:OFF
	(AM)	SDMD?	1:ON(AM)
	(FM)		2:ON(FM)
	OFF		
音量	SDV*	SDV?	音量 (整数)
ポーズ時間	PU*	PU?	ポーズ時間 (時間)
スケルチ レベル	SQE [ON]*	SQE?	Squelch レベル
	ON	SQE ON	0:OFF
	OFF	SQE OFF	1:ON

表 4-10 マーカ →

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
MKR → CF	MKCF	---	---
	MC	---	---
MKRA → CF	MTCF	---	---
MKR → REF	MKRL	---	---
	MR	---	---
ピーク → CF	PKCF	---	---
ピーク → REF	PKRL	---	---
MKRA → SPAN	MTSP	---	---
	DS	---	---
MKR → CF ステップ	MKCS	---	---
	M0	---	---
MKRA → CF ステップ	MTCS	---	---
	M1	---	---
MKR → マーカ・ステップ	MKMKS	---	---
	M2	---	---
MKRA → マーカ・ステップ	MTMKS	---	---
	M3	---	---
マーカ → A トレース	MKTRACE TRA	MKTRACE?	0:Blank 1:A Trace 2:B Trace
マーカ → B トレース	MKTRACE TRB		

4.1 GPIB リモート・プログラミング

表 4-11 トリガ

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
トリガ・モード	——	TM?	0:Free run 1:Line 2:Video 5:Ext
フリー・ラン	TM FREE	——	——
	FR	——	——
ライン・トリガ	TM LINE	——	——
	LI	——	——
ビデオ・トリガ	TM VID	——	——
	VI*	VI?	Level(整数)
外部トリガ	TM EXT	——	——
	EX*	EX?	Level(実数)
トリガ・スロープ +	TRIGSLP+	TRIGSLP?	0:+
	TRIGSLP-		1:-

表 4-12 トレース

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
トレース A	—	TA?	0:Write 1:View 2:Blank 3:Max Hold 4:Min Hold 5:Averaging
A write	AW	—	—
A view	AV	—	—
A blank	AB	—	—
A max hold	AM	—	—
A min hold	AMIN	—	—
A アベレージ 回数	AG*	AG?	アベレージ回数
スタート	AGR	—	—
ストップ	AGS	—	—
Pause	AGP	AGP?	0:Continue 1:Pause
Continue	AGC		
1 time	AGSGL	AGSGL?	0:Continuous 1:1 time
	AG1		
Continuous	AGCNT		
	AG0		
トレース B	—	TB?	0:Write 1:View 2:Blank 3:Max Hold 4:Min Hold 5:Averaging
B write	BW	—	—
B view	BV	—	—
B blank	BB	—	—
B max hold	BM	—	—
B min hold	BMIN	—	—

4.1 GPIB リモート・プログラミング

表 4-12 トレース (続き)

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
B アベレージ 回数	BG*	BG?	アベレージ回数
スタート	BGR	——	——
ストップ	BGS	——	——
Pause	BGP	BGP?	0:Continue
Continue	BGC		1:Pause
1 time	BGSGL	BGSGL?	0:Continuous
	BG1		1:1 time
Continuous	BGCNT		
	BG0		
ストア A → B	BSTORE	——	——
ストア B → A	ASTORE	——	——
ディテクタ・モード A			
ノーマル	DET NRM	DET?	0:Normal
	DTN	DM?	1:Positive
ポジティブ	DET POS		2:Negative
	DTP		3:Sample
ネガティブ	DET NEG		
	DTG		
サンプル	DET SMP		
	DTS		
ディテクタ・モード B			
ノーマル	DETBNRM	DETBN?	0:Normal
	DTBN	DTBN?	1:Positive
ポジティブ	DETBN POS		2:Negative
	DTBNP		3:Sample
ネガティブ	DETBN NEG		
	DTBNG		
サンプル	DETBN SMP		
	DTBNS		

表 4-13 ディスプレイ

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
ディスプレイ・ライン	DL*(,DLN*)	DL?	レベル
ON	DL*(,DLN*)	DLON?	0:OFF
OFF	DLF		1:ON
レファレンス・ライン	RLN*	RLN?	レベル
ON	RLN*	RLON?	0:OFF
OFF	RLF		1:ON
ウィンドウ	ON	WDO ON	WDO?
		WN	WN?
OFF	WDO OFF		0:OFF
	WF		1:ON
ウィンドウ 中心位置	WLX*	WLX?	周波数
幅	WDX*	WDX?	周波数
開始位置	WTF*	WTF?	周波数
終了位置	WPF*	WPF?	周波数
ズーム イン	WDOZM IN	—	—
アウト	WDOZM OUT	—	—
ウィンドウ掃引 ON	WDOSWP ON	—	—
OFF	WDOSWP OFF	—	—
タイトル ON	LON/***/	LB?	ラベル
OFF	LOF		

4.1 GPIB リモート・プログラミング

表 4-14 パス／フェイル

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
Pass/Fail 1(Window) 2(Limit Line) OFF	PF1 PF2 PFF	PFON?	1:1(Window) 2:2(Limit Line) 0:OFF
リミット 上限位置 (Window) 下限位置 (Window)	WUL* WLL *	WUL? WLL?	レベル レベル
判定結果 (Window/Limit Line)	—	PCM?	1:Pass/0:Fail
詳細判定結果 (Limit Line)	—	OPF?	0:Pass 1:Fail(Upper) 2:Fail(Lower) 3:Fail(Both) 4>Error
Upper Fail Point Lower Fail Point	— —	FPU? FPL?	N<DLM>f1,l1<DLM>... N<DLM>f1,l1<DLM>... (*1)
Line1 ON OFF Data 入力 Data 消去	LMTA ON LMTA OFF LMTAIN*,* (*2) LMTADEL	LMTA?	0:OFF/1:ON (*3) (*3)
Line2 ON OFF Data 入力 Data 消去	LMTB ON LMTB OFF LMTBIN*,* (*2) LMTBDEL	LMTB?	0:OFF/1:ON (*3) (*3)
周波数 Domain 入力選択 時間 Domain 入力選択	LIMTYP FREQ LIMTYP TIME	LIMTYP?	0:Freq/1:Time (*3)
X 位置モード 絶対 相対 (Left) 相対 (Center)	LIMPOS ABS LIMPOS REL LIMPOS CENT	LIMPOS?	0: 絶対 1: 相対 (Left)2: 相対 (Center)
Y 位置モード 絶対 相対 (Top) 相対 (Bottom)	LIMAPOS ABS LIMAPOS REL LIMAPOS BOTM	LIMAPOS?	0: 絶対 1: 相対 (Top) 2: 相対 (Bottom)
X オフセット	LIMSFT*	LIMSFT?	周波数/時間
Y オフセット	LIMASFT*	LIMASFT?	レベル

(※ 1) N= ポイント数 fn,ln= 周波数 (時間)、レベル <DLM> デリミタ

(※ 2) * は周波数、レベルの順にパラメータを指定して下さい。

(※ 3) テーブル入力を行う前に LIMTYP コマンドでドメインを選択して下さい。

表 4-15 メジヤメント

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
Noise/Hz	NI*	NI?	周波数
dBm/Hz	NIM	NION?	0:OFF
dB μ V/ $\sqrt{\text{Hz}}$	NIU		1:dBm/Hz
dBc/Hz	NIC		2:dB μ V/ $\sqrt{\text{Hz}}$
OFF	NIF		3:dBc/Hz
Noise/Hz 値	—	NIRES?	レベル
XdB ダウン幅	MKBW*	MKBW?	レベル
XdB ダウン	XDB	—	—
XdB ダウン・レフト	XDL	—	—
XdB ダウン・ライト	XDR	—	—
XdB relative	DC0	DC?	0: 相対
abs. レフト	DC1		1: 絶対 (左側)
abs. ライト	DC2		2: 絶対 (右側)
連続 dB ダウン	ON	CDB?	0:OFF
OFF	CDB OFF		1:ON
3rd Order meas	PKTHIRD	—	—
AM 変調度 (%AM)	—	AMMOD?	%
ON	AMMOD [ON]	AMMODON?	0:OFF
OFF	AMMOD OFF		1:ON

表 4-16 オート・チューニング

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
Auto Tune	TN	—	—

4.1 GPIB リモート・プログラミング

表 4-17 カウンタ

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
分解能 1kHz	CN0	CNORD?	0:1kHz
100Hz	CN1		1:100Hz
10Hz	CN2		2:10Hz
1Hz	CN3		3:1Hz
カウンタ ON	COUNT ON	COUNT?	0:OFF
OFF	COUNT OFF		1:ON
	CNF		
カウンタ値	—	CNRES?	周波数

表 4-18 電力測定

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
チャンネル・パワー	PWCH	PWCH?	レベル
		PWCHON?	0:OFF/ 1:ON
トータル・パワー	PWTOTAL	PWTOTAL?	レベル
		PWTOTALON?	0:OFF/ 1:ON
アベレージ・パワー	PWAVG	PWAVG?	レベル
		PWAVGON?	0:OFF/ 1:ON
キャリア・パワー	PWCARR	PWCARR?	レベル
アベレージ回数	PWTM*	PWTM?	1 - 999
パワー OFF	PWM	—	—

表 4-19 OBW

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
OBW 実行 (%)	OBW [ON]*	OBW?	Center,OBW
ON	OBW [ON]	OBWON?	0:OFF
OFF	OBW OFF		1:ON
OBW パーセント	OBWPER*	OBWPER?	OBW%
OBW 即時実行	OBWEXE	—	—

表 4-20 ACP

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
ACP 実行	ACP	ACP?	Lower,Upper
ON	ACP	ACPON?	0:OFF
OFF	ACP OFF		1:ONACPACPA
ACP 即時実行	ACPEXE	—	—
ACP CS	ADCH*	ADCH?	チャンネル間隔
ACP BS	ADBS*	ADBS?	規定帯域幅
ACP(2次)	ADCHSC [ON]*	ADCHSC?	チャンネル間隔 (2次)
	—	ACPSC?	Lower1,Upper1, Lower2,Upper2
	ON	ADCHSC [ON]	ADCHSCON? 0:OFF 1:ON
	OFF	ADCHSC OFF	
ACP グラフ	ON	ADG [ON]	ADG? 0:OFF 1:ON
	OFF	ADG OFF	

表 4-21 TG (オプション 74)

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
TG ON	TG	TG?	0:OFF
OFF	TGF		1:ON
TG レベル	TGL*	TGL?	TG 出力レベル
トラッキング調整	TGM*	TGM?	周波数
トラッキング調整オート	TGA	—	—
ノーマライズ実行	NORM EX	—	—
ノーマライズ ON	NORM ON	NORM?	0:OFF
OFF	NORM OFF		1:ON

4.1 GPIB リモート・プログラミング

表 4-22 EMC

ファンクション	リスナコード	トーカー・リクエスト			
		コード	出力フォーマット		
EMC トレース・ ディテクション					
QP	EMCDET QP	EMCDET?	0:Normal		
ピーク	EMCDET PEAK		1:QP		
ノーマル	EMCDET NRM		3:PEAK		
QP BW 9kHz	QP1	QPAUTO?	0:AUTO		
120kHz	QP2	QA?	2:9kHz		
オート	QPAUTO		3:120kHz		
	QA				
アンテナ選択		ANT?	0: OFF 1: ダイポール 2: ログペリ 3: バイコニカル 4: バイログ 5: ユーザ補正		
ダイポール (TR1722)	ANT0 AN0				
ログペリ (UHALP9107)	ANT1 AN1				
バイコニカル (BBA9106)	ANT2 AN2				
バイログ (EMCO3142)	ANT3 AN3				
ユーザ補正	ANT4 AN4				
アンテナ OFF	ANT OFF AF				
ユーザ補正テーブル					
テーブル入力	CRIN *,* (※)			——	——
テーブル消去	CRDEL			——	——
アンテナ・モード	CR ANT	CR?	0:Antenna		
レベル・モード	CR LVL		1:Level		

(※) * は周波数、レベルの順にパラメータを指定して下さい。

表 4-23 キャリブレーション

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
CAL オール	CLA	---	---
トータル・ゲイン	CLG	---	---
IF ステップ・アンプ	CLSTEP	---	---
	IT1	---	---
RBW スイッチ	CLRBW	---	---
	IT2	---	---
Log リニアリティ	CLLOG	---	---
	IT3	---	---
AMPTD MAG	CLMAG	---	---
	IT4	---	---
PBW	CLPBW	---	---
	IT6	---	---
CAL レベル	CLN*	CL?	レベル
ON	CLN*	CLON?	0:OFF
OFF	CLF		1:ON
f 特補正 ON	FC ON	FC?	0:OFF 1:ON
	FCN		
	FC OFF		
	FCF		
CAL 補正 ON	CC ON	CC?	0:OFF 1:ON
	CCN		
	CC OFF		
	CCF		

4.1 GPIB リモート・プログラミング

表 4-24 データの保存／読み出し

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
セーブ・レジスタ	SV1	---	---
	SV2	---	---
	SV3	---	---
	SV4	---	---
	SV5	---	---
	SV6	---	---
	SV7	---	---
	SV8	---	---
	SV9	---	---
	SV10	---	---
セーブ・ファイル	SV FD: ファイル名 (※)	---	---
デリート・レジスタ	DEL1	---	---
	DEL2	---	---
	DEL3	---	---
	DEL4	---	---
	DEL5	---	---
	DEL6	---	---
	DEL7	---	---
	DEL8	---	---
	DEL9	---	---
	DEL10	---	---
デリート・ファイル	DEL FD: ファイル名 (※)	---	---
リコール・レジスタ	RC1	---	---
	RC2	---	---
	RC3	---	---
	RC4	---	---
	RC5	---	---
	RC6	---	---
	RC7	---	---
	RC8	---	---
	RC9	---	---
	RC10	---	---

(※) “FD:” はフロッピー・ディスク・ドライブを指定します。

表 4-24 データの保存／読み出し (続き)

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
リコール・ファイル	RC FD: ファイル名 (※1)	—	—
セーブ項目			
設定条件	ON SVSET ON	SVSET?	0:OFF
	OFF SVSET OFF		1:ON
トレース	ON SVTRC ON	SVTRC?	0:OFF
	OFF SVTRC OFF		1:ON
アンテナ補正	ON SVANT ON	SVANT?	0:OFF
	OFF SVANT OFF		1:ON
トレース・ レベル	ON SVLVL ON	SVLVL?	0:OFF
	OFF SVLVL OFF		1:ON
ノーマライズ・ データ (※2)	ON SVNRM ON	SVNRM?	0:OFF
	OFF SVNRM OFF		1:ON
リミット・ ライン	1 ON SVLIM 1	SVLIM?	0:OFF
	2 ON SVLIM 2		1:1 ON
	1/2 ON SVLIM 3		2:2 ON
	OFF SVLIM OFF		3:1/2 ON

(※1) “FD:” はフロッピー・ディスク・ドライブを指定します。

(※2) オプション 74 搭載時のみ有効

表 4-25 ハード・コピー

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
コマンド・セレクト			
ESC/P	PRTCMD ESC	PRTCMD?	0:PCL
ESC/P ラスタ	PRTCMD ESCPR		1:ESCP
PCL	PRTCMD PCL		2:ESCP-R
プリント・サイズ			
ラージ	PSIZE LRG	PSIZE?	0:Small
スモール	PSIZE SML		1:Large
モード・セレクト			
Gray	HCIMAG GRY	HCIMAG?	0:Gray
monochrome	HCIMAG MON		1:B&W
デバイス選択			
プリンタ	HCDEV PRT	HCDEV?	0:Printer
フロッピー	HCDEV FDD		1:Floppy
ビットマップ・ファイル 番号	HCFILE*	HCFILE?	番号 (0~999)

4.1 GPIB リモート・プログラミング

表 4-25 ハード・コピー (続き)

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
ビットマップ・ファイル オーバーライト ON OFF	HCOVWRT ON HCOVWRT OFF	HCOVWRT?	0:OFF 1:ON
実行	HCOPY	—	—

表 4-26 プリセット

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
プリセット	IP	—	—

表 4-27 トレース・データ入出力

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
A トレース入出力 ASCII	TAA	TAA?	ASCII Trace
バイナリ	TBA	TBA?	BIN Trace
B トレース入出力 ASCII	TAB	TAB?	ASCII Trace
バイナリ	TBB	TBB?	BIN Trace

表 4-28 ステータス・バイト

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
ステータス・バイト・ クリア	*CLS	—	—
STB の読み出し	—	*STB?	0-255
SRE の読み書き	*SRE*	*SRE?	0-255
ESR の読み出し	—	*ESR?	0-255
ESE の読み書き	*ESE*	*ESE?	0-255
OSR の読み出し	—	OPREVT?	0-65535
OSER の読み出し	OPR	OPR?	0-65535
SRQ 割り込み ON	S0	—	—
SRQ 割り込み OFF	S1	—	—
SRQ ステータス・クリア	S2	—	—
サービス・リクエスト・ マスク	RQS*	RQS?	0-255

表 4-29 その他のコマンド

ファンクション	リスナコード	トーカー・リクエスト	
		コード	出力フォーマット
デリミタ CR LF <EOI>	DL0	—	—
LF	DL1	—	
<EOI>	DL2	—	
CR LF	DL3	—	
LF <EOI>	DL4	—	
↑ステップアップ	UP	—	—
↓ステップダウン	DN	—	—
ノブアップ coarse	CU	—	—
fine	FU	—	—
ノブダウン coarse	CD	—	—
fine	FD	—	—
表示 ON	ANNOT ON	ANNOT?	0:OFF
OFF	ANNOT OFF		1:ON
日付	SETDATE	SETDATE?	YYMMDD (例 :980528)
時間	SETTIME	SETTIME?	HHMMSS (例 :130530)
機器 ID の出力	—	*IDN?	メーカー名,機種タイプ,シリアル番号,レビジョン
セルフ・テスト	—	*TST?	1 : CPU (※ 1) 2 : A/D 4 : Lock Detector 8 : Total Gain 16 : Cal Signal 32 : RF ATT 64 : Variable Gain AMP 128 : IF Step AMP 256 : RBW SWB
エラー番号出力	—	ERRNO?	整数 (※ 2)

(※ 1) エラーの発生した項目は、bit OR の値が戻値となります。

(※ 2) A.1 節のエラー・メッセージ一覧のエラー番号を参照して下さい。

表 4-30 データ・エントリ

ファンクション	パラメータ
数値	0
	1
	2
	3
	4
	5
	6
	7
	8
	9
小数点	.
負符合	-
正符合	+
指数	EXP
	E
GHz	GZ
MHz	MZ
KHz	KZ
Hz	HZ
mW	MW
DB 関係	DB
mA	MA
秒	SC
m 秒	MS
	MSEC
μ 秒	US
	USEC
n 秒	NSEC
Enter	ENT
Volt	VOLT
mVolt	MV
μVolt	UV
nVolt	NV
%	PER
	%

4.1.10 測定条件の設定および読み込みのプログラム例

注意 記述したサンプル・プログラムは、言語として Visual Basic 4.0（以降 VB と記述）および HP BASIC for Windows（以降 HP BASIC と記述）を使用しています。また、GPIB 用コントロール・ボードとして National Instruments 社（以降 NI 社と記述）製 GPIB ボードを、コントロール・ドライバとして NI 社のドライバを使用しています。

(1) VB のプログラム例

例 VB-1 本器をマスタ・リセットしたあと、中心周波数の設定

```
Call ibclr(spa)           ' デバイス・クリア
Call ibwrt(spa, "IP")    ' preset
Call ibwrt(spa, "CF 30MZ") ' 中心周波数を 30MHz に設定
```

例 VB-2 リファレンス・レベルを 87dB μ V、5dB/div、RBW を 100kHz にする

```
Call ibclr(spa)           ' デバイス・クリア
Call ibwrt(spa, "AUNITS DBUV") ' レベル単位を dB $\mu$ V に設定
Call ibwrt(spa, "RL 87DB")  ' リファレンス・レベルを 87dB( $\mu$ V) に設定
Call ibwrt(spa, "DD 5DB")   ' 縦軸目盛りを 5dB/div に設定
Call ibwrt(spa, "RB 100KZ") ' RBW を 100kHz に設定
```

4.1 GPIB リモート・プログラミング

例 VB-3 変数を用いた設定の例

```
Dim A As String
Dim B As String
Dim C As String

A = "10"
B = "2"
C = "20"

Call ibclr(spa)

Call ibwrt(spa, "CF " & A & "MZ")
Call ibwrt(spa, "SP " & B & "MZ")
Call ibwrt(spa, "AT " & C & "DB")
```

・ 文字列の設定
・ デバイス・クリア
・ 中心周波数を A MHz に設定
・ スパンを B MHz に設定
・ ATT を C dB に設定

例 VB-4 レジスタ 5 への設定値のセーブおよびリコール

```
Dim LabelBuff As String

LabelBuff = "SPECTRUM Analyzer"

Call ibclr(spa)

Call ibwrt(spa, "CF 30MZ")
Call ibwrt(spa, "SP 1MZ")
Call ibwrt(spa, "DET POS")
Call ibwrt(spa, "LON " & LabelBuff)

Call ibwrt(spa, "SV5")

Call ibwrt(spa, "CF 1GZ")
Call ibwrt(spa, "SP 200MZ")

Call ibwrt(spa, "RC5")
```

・ ラベル用の文字列バッファ
・ ラベルの設定
・ デバイス・クリア
・ パラメータの設定
・ ラベルの設定
・ レジスタ 5 へセーブ
・ 設定パラメータの変更
・ レジスタ 5 からのリコール

例 VB-5 リミット・ライン 1 テーブル入力して ON する

Call ibclr(spa)	・ デバイス・クリア
Call ibwrt(spa, "LMTADEL")	・ リミット・ライン 1 のテーブルを消去
Call ibwrt(spa, "AUNITS DBUV")	・ レベル単位を dB μ V に設定
Call ibwrt(spa, "LMTAIN 25MZ, 49.5DB")	・ リミット・ライン 1 のデータを入力
Call ibwrt(spa, "LMTAIN 35MZ, 49.5DB")	
Call ibwrt(spa, "LMTAIN 35MZ, 51.5DB")	
Call ibwrt(spa, "LMTAIN 55MZ, 51.5DB")	
Call ibwrt(spa, "LMTAIN 55MZ, 54.3DB")	
Call ibwrt(spa, "LMTAIN 65MZ, 54.3DB")	
Call ibwrt(spa, "LMTAIN 65MZ, 57.0DB")	
Call ibwrt(spa, "LMTAIN 68MZ, 57.0DB")	
Call ibwrt(spa, "LMTAIN 68MZ, 60.0DB")	
Call ibwrt(spa, "LMTAIN 75MZ, 60.0DB")	
Call ibwrt(spa, "LMTAIN 75MZ, 62.5DB")	
Call ibwrt(spa, "LMTAIN 82MZ, 62.5DB")	
Call ibwrt(spa, "LMTAIN 82MZ, 64.7DB")	
Call ibwrt(spa, "FA 0MZ")	・ スタート周波数を 0MHz に設定
Call ibwrt(spa, "FB 100MZ")	・ ストップ周波数を 100MHz に設定
Call ibwrt(spa, "LMTA ON")	・ リミット・ライン 1 を ON

4.1 GPIB リモート・プログラミング

(2) HP BASIC のプログラム例 (GPIB アドレス = 1)

- HP-1 本器をマスタ・リセットし、中心周波数を 30MHz にする
- ```
10 OUTPUT 701;"IP"
20 OUTPUT 701;"CF30MZ"
30 END
```
- HP-2 基準レベルを -20 dBm(5 dB/div)、分解能帯域幅を 100 kHz、ディテクタ・モードを posi に設定する
- ```
10 OUTPUT 701;"RL-20DB"  
20 OUTPUT 701;"DD5DB"  
30 OUTPUT 701;"RB100KZ"  
40 OUTPUT 701;"DTP"  
50 END
```
- HP-3 トリガモードをシングル、掃引時間を 2 秒に設定し、掃引のたびに最大レベルへマークをのせる
- ```
10 OUTPUT 701;"SI"
20 OUTPUT 701;"SW2SC"
30 OUTPUT 701;"SR" ! 掃引の開始
40 WAIT 2.5 ! 掃引の終了を待つ (またはサービス・リクエストを使う)
50 OUTPUT 701;"PS" ! マークのピークサーチ
60 GOTO 30
70 STOP
80 END
```
- HP-4 MAX HOLD (A) に設定する
- ```
OUTPUT 701;"AM"           ! ダイレクトに設定する
```
- HP-5 ファイル・アクセス関連
- ```
OUTPUT 701;"RC5" ! レジスタ 5 をリコールする
OUTPUT 701;"RC/FD:FILE01.DAT/" ! カードからリコールする
```

---

注 RC、DEL、SV コマンドでのファイル・アクセス方法は、同一形式です。

---



#### 4.1.11 データ読み込みのプログラム例

測定データや設定状態などの内部データを出力させるには、“xx?” コマンドで出力させたいデータの指定をしておきます。そして本器がトーカーになったときに指定したデータを読み込みます。出力のフォーマットは、大きく分けると下表のようになります。最終データとなるデリミタは、5種類の指定ができます (GPIB コード一覧のその他を参照)。なお、一度設定した“xx?” コマンドは変更があるまで有効です。

| 出力フォーマット |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 周波数系     | $\pm \text{D.DDDDDDDDDDD} \text{ E } \pm \text{DD} \text{ CR LF}$ <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">↑<br/>1</div> <div style="text-align: center;">↑<br/>2</div> <div style="text-align: center;">↑<br/>3</div> <div style="text-align: center;">↑<br/>4</div> </div> <ul style="list-style-type: none"> <li>データサイズ (1 ~ 3) は最大 18 バイト、単位は Hz</li> </ul> <p>(例) ”CF?” を指定し、中心周波数を出力する場合等</p>      |
| レベル系     | $\pm \text{D.DDDDDDD} \text{ E } \pm \text{DD} \text{ CR LF}$ <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">↑<br/>1</div> <div style="text-align: center;">↑<br/>2</div> <div style="text-align: center;">↑<br/>3</div> <div style="text-align: center;">↑<br/>4</div> </div> <ul style="list-style-type: none"> <li>データサイズ (1 ~ 3) は最大 18 バイト、単位は各 UNIT に従う</li> </ul> <p>(例) ”ML?” を指定し、マーカ・レベルを出力する場合等</p> |
| 時間系      | $\pm \text{D.DDD} \text{ E } \pm \text{DD} \text{ CR LF}$ <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">↑<br/>1</div> <div style="text-align: center;">↑<br/>2</div> <div style="text-align: center;">↑<br/>3</div> <div style="text-align: center;">↑<br/>4</div> </div> <ul style="list-style-type: none"> <li>データサイズ (1 ~ 3) は最大 18 バイト、単位は sec</li> </ul> <p>(例) ”SW?” を指定し、掃引時間を出力する場合等</p>              |
| 定数系      | $\text{DDDD} \text{ CR LF}$ <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">↑<br/>2</div> <div style="text-align: center;">↑<br/>4</div> </div> <ul style="list-style-type: none"> <li>データサイズの最大バイトは、出力データの最大による</li> </ul> <p>(例) ON/OFF 状態を出力またはアベレージ回数を出力する場合等</p>                                                                                                                                           |

- 【補足】
- 1= 符号 (正はスペース、負は - が入る)
  - 2= データ仮数部
  - 3= データ指数部
  - 4= デリミタ (初期設定時 CR/LF, “DLn” コードで変更可能)

## 4.1 GPIB リモート・プログラミング

## (1) VB のプログラム例

例 VB-6 マーカ・レベルを読み込み、表示する

```

Dim sep As Integer

Call ibclr(spa) ' デバイス・クリア

Call ibwrt(spa, "CF 30MZ") ' パラメータ設定
Call ibwrt(spa, "SP 1MZ")
Call ibwrt(spa, "MK 30MZ") ' マーカを 30MHz に設定
Call ibwrt(spa, "TS")

Call ibwrt(spa, "ML?") ' マーカ・レベルのクエリ

Rdbuf = Space(30) ' バッファ領域を 30 バイト確保

Call ibrd(spa, Rdbuf) ' データを読み込む (MAX は 30 バイトになる)
sep = InStr(1, Rdbuf, vbCrLf, 0) ' デリミタまでの文字数をチェック
RichTextBox1.Text = "MarkerLevel = " & Left(Rdbuf, sep - 1)
' 画面に出力

```

結果例

MarkerLevel = -8.818750000000E+01

例 VB-7 中心周波数を読み込み、表示する

```

Dim sep As Integer

Call ibclr(spa) ' デバイス・クリア

Call ibwrt(spa, "CF?") ' 中心周波数値のクエリ・コマンド

Rdbuf = Space(30) ' バッファ領域 30 バイト確保
Call ibrd(spa, Rdbuf) ' 最大 30 バイト分を読み込む
sep = InStr(1, Rdbuf, vbCrLf, 0) ' デリミタまでの文字数をチェック
RichTextBox1.Text = "CenterFreq = " & Left(Rdbuf, sep - 1)
' 画面に出力

```

結果例

CenterFreq = +3.000000000000E+7

## 例 VB-8 レベルの表示単位およびレベルを読み込み、表示する

```

Dim sep As Integer

Call ibclr(spa) ' デバイス・クリア

Call ibwrt(spa, "RL?") ' リファレンス・レベルのクエリ

Rdbuf = Space(30) ' バッファ領域を 30 バイト取る
Call ibrd(spa, Rdbuf) ' スペクトラム・アナライザからの読み込み
sep = InStr(1, Rdbuf, vbCrLf, 0) ' デリミタまでの文字数をチェック
RichTextBox1.Text = "RefLevel = " & Left(Rdbuf, sep - 1)
 ' 画面に出力

Call ibwrt(spa, "AUNITS?") ' レベル単位のクエリ

Rdbuf = Space(3)
Call ibrd(spa, Rdbuf)
sep = InStr(1, Rdbuf, vbCrLf, 0) ' デリミタまでの文字数をチェック
RichTextBox1.Text = RichTextBox1.Text & vbCrLf & "UNIT = " & Left(Rdbuf, sep - 1)
 ' 前回の結果に、改行と今回の出力を追加して、画面に出力

```

```

結果例
RefLevel = +0.000000000000E+00
UNIT = 0

```

## 例 VB-9 6dB ダウンを実行後、その周波数とレベルを読み込み、表示する

```

Dim sep As Integer

Call ibclr(spa) ' デバイス・クリア

Call ibwrt(spa, "CF 30MZ") ' パラメータを設定
Call ibwrt(spa, "SP 20MZ")

Call ibwrt(spa, "MKBW 6DB") ' 6dB ダウンを設定
Call ibwrt(spa, "PS") ' ピーク・サーチ
Call ibwrt(spa, "XDB") ' 設定した dB ダウンを実行
Call ibwrt(spa, "MFL?") ' マーカ・レベル&周波数値のクエリ

Rdbuf = Space(50) ' バッファ領域 50 バイト確保
Call ibrd(spa, Rdbuf) ' データ読み出し (MAX 50 バイト)

sep = InStr(1, Rdbuf, vbCrLf, 0) ' デリミタまでの文字数をチェック

RichTextBox1.Text = "Marker Freq & Level = " & Left(Rdbuf, sep - 1)
 ' 画面に出力

```

```

結果例
Marker Freq & Level = +2.000000000000E+05, +1.023437500000E+00

```

## 4.1 GPIB リモート・プログラミング

例 VB-10 OBW を測定し、表示する

```

Dim LENG1 As Integer, LENG2 As Integer
Dim OBW As String
Dim FC As String
Dim searchchar As String

Call ibclr(spa) ' デバイス・クリア

Call ibwrt(spa, "CF 30MZ") ' 設定コマンドの送信
Call ibwrt(spa, "SP 1MZ")
Call ibwrt(spa, "MK 30MZ")
Call ibwrt(spa, "OBW")
Call ibwrt(spa, "TS")

Call ibwrt(spa, "OBW?") ' クエリ・コマンドの送信
Rdbuf = Space(60) ' 読み取りバッファの領域確保
Call ibrd(spa, Rdbuf) ' 読み出す (MAX 出力バイト数はバッファの領域範囲)

' 出力文字列の整形
LENG1 = InStr(1, Rdbuf, Chr(44), 0) ' 1 番目のコンマの位置を検索
FC = Mid(Rdbuf, 1, LENG1 - 1) ' コンマまでの文字列を取る

DoEvents

LENG2 = InStr((LENG1 + 1), Rdbuf, Chr(13), 0) ' 最後のデータはデリミタの位置を検索
OBW = Mid(Rdbuf, (LENG1 + 1), (LENG2 - LENG1 - 1)) ' 2 番目のコンマとデリミタの間の文字列を取る

RichTextBox1.Text = "OBW = " & OBW & vbCrLf & "Fc = " & FC & vbCrLf
' 画面に出力

結果例
OBW= +9.810000000000E+05
FC = +3.000250000000E+07

```

例 VB-11 信号の最大および第 2、第 3 のピークのレベル値を読み込み、表示する

```
Dim pk1 As String, pk2 As String, pk3 As String
```

```
Call ibclr(spa) ' デバイス・クリア
Call ibwrt(spa, "CF 0MZ") ' 設定
Call ibwrt(spa, "SP 100MZ")
```

```
Call ibwrt(spa, "PS") ' ピーク・サーチ
Call ibwrt(spa, "ML?") ' マーカ・レベル値のクエリ・コマンド
Rdbuf = Space(25) ' バッファの領域確保
Call ibrd(spa, Rdbuf) ' 出力を受け取る
pk1 = LeftB(Rdbuf, (InStrB(1, Rdbuf, Chr(13), 1) - 1))
 ' バッファからデリミタの一文字前を取り出す
```

```
Call ibwrt(spa, "NXP") ' ネクスト・ピーク・サーチ
Call ibwrt(spa, "ML?")
Rdbuf = Space(25)
Call ibrd(spa, Rdbuf)
pk2 = LeftB(Rdbuf, (InStrB(1, Rdbuf, Chr(13), 1) - 1))
 ' バッファからデリミタの一文字前を取り出す
```

```
Call ibwrt(spa, "NXP")
Call ibwrt(spa, "ML?")
Rdbuf = Space(25)
Call ibrd(spa, Rdbuf)
pk3 = LeftB(Rdbuf, (InStrB(1, Rdbuf, Chr(13), 1) - 1))
 ' バッファからデリミタの一文字前を取り出す
```

```
RichTextBox1.Text = "1st PK = " & pk1 & vbCrLf & "2nd PK = " & pk2 & vbCrLf & "3rd PK = " & pk3 & vbCrLf
 ' 画面に出力
```

結果例

```
1st PK = -8.553906250000E+01
2nd PK = -7.004687500000E+01
3rd PK = -8.655468750000E+01
```

## 4.1 GPIB リモート・プログラミング

## (2) HP BASIC のプログラム例 (GPIB アドレス = 1)

HP-6 マーカ周波数を出力する (整数値)

```
10 OUTPUT 701;"MF?"
20 ENTER 701;A
30 END
```

結果例 A=1.8E+9

HP-7 中心周波数を出力する (文字列)

```
10 DIM A$ [30]
20 OUTPUT 701;"CF?"
30 ENTER 701;A$
40 END
```

結果例 A\$= 1.234567E+9

HP-8 ユニットの状態を出力する

```
10 OUTPUT 701;"UN?"
20 ENTER 701;A
30 END
```

結果例 A=2 (dBuV)

HP-9 マーカの周波数とレベルを同時に出力する (複数個の出力)

```
10 OUTPUT 701;"MFL?"
20 ENTER 701;Mf,MI
30 END
```

結果例 Mf=1.8E+9 MI=-65.15

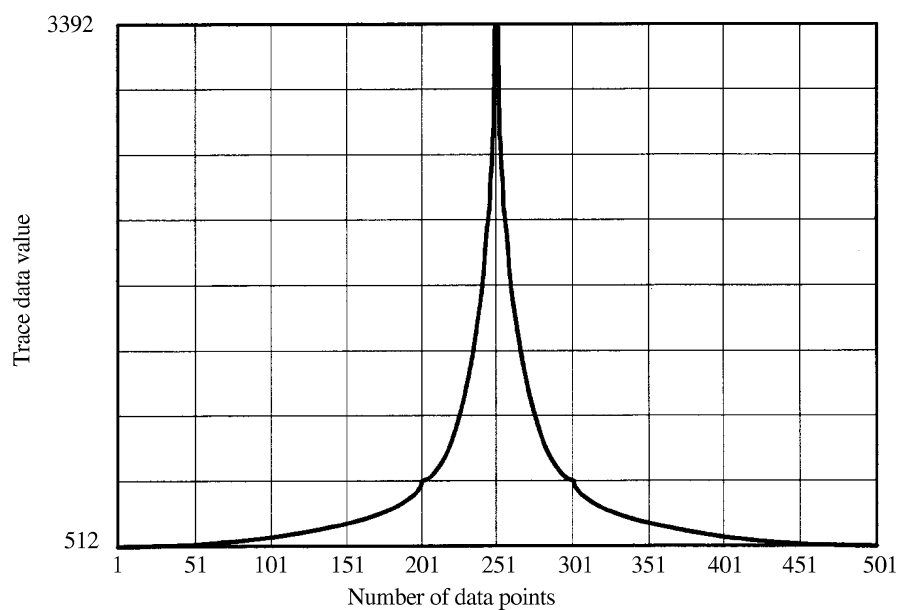
HP-10 NEXT PEAK を使用し、信号の第 2 ピーク・レベル から 10 個のピーク・レベル を読み取る

```
10 DIM MI(9)
20 OUTPUT 701;"PS"
30 FOR I=0 TO 9
40 OUTPUT 701;"NXP"
50 OUTPUT 701;"ML?"
60 ENTER 701;MI(I)
70 NEXT I
80 END
```

結果例 MI(0)=-55.01 MI(1)=-58.22 .... MI(9)=-70.26

#### 4.1.12 トレース・データの入出力のプログラム例

画面上のトレース・データは周波数軸上で、501 ポイントのデータで構成しています。このデータを入出力するには左（スタート周波数）から順に 501 ポイント分のデータを転送します。各ポイントのレベル値は、512 ~ 3392 の整数値で表わします（ただし、スケールの枠から上方へはずれた波形については、3392 を越えた値になります）。







## (1) VB のプログラム例

## 例 VB-12 トレース・データを ASCII で読み込む

```

Dim tr(500) As String ' 501 ポイント分のバッファの配列
Dim i As Integer
Dim res As String

Call ibclr(spa) ' デバイス・クリア

Call ibwrt(spa, "DL0") ' CR LF EOI
Call ibwrt(spa, "DET NEG") ' ネガティブ・ディテクタに設定
Call ibwrt(spa, "TAA?")

For i = 0 To 500 Step 1 ' 501 ポイント分繰り返す
 tr(i) = Space(6) ' データ 4 バイト + デリミタ 2 バイトで 6 バイト確保
 Call ibrd(spa, tr(i)) ' 読み込み
 ' 画面に出力
 res = res & "tr(" & Str(i) & ") = " & Left(tr(i), 4) & vbCrLf

 DoEvents
Next i

RichTextBox1.Text = res

```

## 例 VB-13 A メモリのデータをバイナリで読み込む

```

Dim tr(500) As Integer ' 501 ポイント分の配列
Dim i As Integer
Dim res As String

Call ibclr(spa) ' デバイス・クリア
Call ibconfig(0, IbcEndBitIsNormal, 0) ' EOI を受け取ったときのみ Ibsta 変数の End ビットが立つ
 ' ように GPIB ボードのソフト設定をする
Call ibconfig(spa, IbcReadAdjust, 1) ' 読み取り処理中にデータの上位バイトと下位バイトを入れ
 ' 替える

Call ibwrt(spa, "DL2") ' EOI のみのデリミタに設定
Call ibwrt(spa, "DET NEG") ' ネガティブ・ディテクタに設定

Call ibwrt(spa, "TBA?") ' トレース A のバイナリ・データでのクエリ

Call ibrdi(spa, tr(), 501 * 2) ' 501 ポイント分のバイナリ・データ読み込み

For i = 0 To 500 Step 1 ' 501 ポイント分繰り返す
 res = res & Str(tr(i)) & vbCrLf ' 画面に出力
 DoEvents
Next i
RichTextBox1.Text = res

Call ibwrt(spa, "DL0") ' CR LF EOI にデリミタを設定

Call ibconfig(0, IbcEndBitIsNormal, 1) ' GPIB のソフト設定を標準に戻す
Call ibconfig(spa, IbcReadAdjust, 0)

```

## 4.1 GPIB リモート・プログラミング

例 VB-14 A メモリにデータを ASCII で入力する

```
Dim trdata(500) As Integer
Dim i As Integer

trdata(0) = 512
For i = 1 To 500 Step 1
 trdata(i) = Str(Val(trdata(i - 1)) + 12)
 DoEvents
Next i

Call ibclr(spa)
Call ibwrt(spa, "AB")
Call ibwrt(spa, "TAA")

For i = 0 To 500 Step 1
 Call ibwrt(spa, CStr(trdata(i)))
 DoEvents
Next i

Call ibwrt(spa, "AV")
```

’ 入力用のテスト用仮データを作成 (※)

’ データがある場合、(※) からここまでの記述は不要

’ デバイス・クリア

’ トレース A を BLANK に設定

’ トレース A を ASCII 入力設定

’ 501 ポイント分のデータを送信

’ 数値を ASCII に変換して送信

’ トレース A を VIEW に設定

## (2) HP BASIC のプログラム例 (GPIB アドレス = 1)

## HP-11 A メモリのデータを ASCII で出力する

```

10 DIM Tr(500) !変数を 501 個確保
20 OUTPUT 701;"DL3" !デリミタを CR LF にする
30 OUTPUT 701;"TAA?" !A メモリ ASCII 指定
40 FOR I=0 TO 500 !データの取込みを 501 回繰り返す
50 ENTER 701;Tr(I) !
60 NEXT I !
70 END

```

## HP-12 B メモリのデータをバイナリで出力する

```

10 DIM Tr (500) !変数を 501 個確保
20 OUTPUT 701;"DL2" !デリミタを EOI にする
30 OUTPUT 701;"TBB?" !B メモリ バイナリ指定
40 ENTER 701 USING"%W";Tr(*) !EOI がくるまでワード型変換してデー
50 END !タを取り込む

```

---

注 データが ASCII の場合は、入出力する回数は必ず 501 回分の指定をして下さい。  
またデータがバイナリの場合も、501 個のデータを確保し、デリミタは必ず EOI 指定を行って下さい。

---

## HP-13 A メモリにデータを ASCII で入力する

```

10 INTEGER Tr(500) !
20 OUTPUT 701;"TAA" !A メモリ ASCII 指定
30 FOR I=0 TO 500 !501 個確保された変数 Tr の入力を 501
40 OUTPUT 701;Tr(I) !回繰り返す
50 NEXT I !
60 END

```

---

注 プログラム実行前に VIEW モードに設定する必要があります。実行後に再び VIEW キーを押すと入力した結果が確認できます。

---

## HP-14 B メモリにデータをバイナリで入力する

```

10 INTEGER Tr(500) !
20 OUTPUT 701;"TBB" !B メモリ バイナリ指定
30 OUTPUT 701 USING"#W";Tr(*),END !501 個のデータをワードサイズで入力し
40 END !最終に EOI を付加する

```

## 注

1. プログラム実行前に VIEW モードに設定する必要があります。実行後に再び VIEW キーを押すと入力した結果が確認できます。
  2. データが ASCII の場合は、入出力する回数は必ず 501 回分の指定をして下さい。  
またデータがバイナリの場合も、501 個のデータを確保し、デリミタは必ず EOI 指定を行って下さい。
-

## 4.1.13 TS コマンド (Take Sweep) を使用したプログラム例

## (1) VB のプログラム例

例 VB-15 ACP 測定を行い、測定終了後に結果を読み出す (TS コマンドを使用)

```

Dim state As Integer
Dim sep1 As Integer, sep2 As Integer
Dim j As Integer

Dim LvlH As String, LvlL As String

Call ibclr(spa)

Call ibwrt(spa, "SI")
Call ibwrt(spa, "CF 1500MZ")
Call ibwrt(spa, "SP 250KZ")
Call ibwrt(spa, "RB 1KZ")
Call ibwrt(spa, "VB 3KZ")
Call ibwrt(spa, "ST 5SC")
Call ibwrt(spa, "ADCH 50KZ")
Call ibwrt(spa, "ADBS21KZ")

Call ibwrt(spa, "ACP")

For j = 1 To 10 Step 1
 Call ibwrt(spa, "TS")
 Call ibwrt(spa, "ACP?")

 Rdbuf = Space(41)
 Call ibrd(spa, Rdbuf)

 sep1 = InStr(1, Rdbuf, ",", 0)
 LvlL = Left(Rdbuf, sep1 - 1)

 sep2 = InStr(sep1, Rdbuf, Chr(13), 0)
 LvlH = Mid(Rdbuf, sep1 + 1, sep2 - sep1 - 1)

 RichTextBox1.Text = RichTextBox1.Text & "-50kHz:" & LvlL & vbCrLf
 RichTextBox1.Text = RichTextBox1.Text & "50kHz:" & LvlH & vbCrLf

 DoEvents
Next j

```

' シングル掃引モードに設定  
 ' 中心周波数を 1500MHz に設定  
 ' 周波数スパンを 250kHz に設定  
 ' RBW を 1kHz に設定  
 ' VBW を 3kHz に設定  
 ' 掃引時間を 20 秒に設定  
 ' チャンネルスペースを 50kHz に設定  
 ' 帯域幅を 21kHz に設定  
 ' ACP 測定を開始  
 ' 実数 19×2 + ', ' + CRLF = 41 バイトの領域を確保  
 ' 読み込む  
 ' バッファの頭からのカンマの位置を得る  
 ' 先頭からカンマの位置までの文字列を取り出す  
 ' ターミネータの位置を得る  
 ' セパレータの文字列を得る  
 ' 画面に出力

#### 4.1.14 ステータス・バイトを使用したプログラム例

##### (1) VB のプログラム例

例 VB-16 シングル掃引をして、掃引の終了を待つ (SRQ を使用しない場合)

```

Dim state As Integer

Call ibclr(spa)
Call ibwrt(spa, "SI")
Call ibwrt(spa, "OPR8")

Call ibwrt(spa, "*CLS")
Call ibwrt(spa, "SI")

Do

 Call ibwrt(spa, "*STB?")
 Rdbuff = Space(8)
 Call ibrd(spa, Rdbuff)
 state = Val(Rdbuff)

 DoEvents
Loop Until (state And 128)

```

- ・ デバイス・クリア
- ・ シングル掃引モードに設定
- ・ オペレーション・ステータス・レジスタの掃引終了ビットを有効にする
- ・ 現状のステータス・バイトをクリア
- ・ 掃引を開始
- ・ ステータス・バイト値のクエリ・コマンド
- ・ デリミタも含めて最大 8 バイトの領域を確保
- ・ 読み込む
- ・ 文字列を数値に変換する
- ・ ループ内に起こっている他のイベントをチェック
- ・ 掃引終了ビットが立っていればループを抜ける

例 VB-17 ACP 測定を行い、測定終了後に結果を読み出す (SRQ 信号を使用しない場合)

```

Dim state As Integer
Dim cnt As Integer
Dim i As Integer
Dim sep1 As Integer, sep2 As Integer
Dim LOL As String, UPL As String

Call ibclr(spa)

Call ibwrt(spa, "CF 1500MZ")
Call ibwrt(spa, "SP 250KZ")
Call ibwrt(spa, "RB 1KZ")
Call ibwrt(spa, "VB 3KZ")
Call ibwrt(spa, "ST 20SC")
Call ibwrt(spa, "ADCH 50KZ")
Call ibwrt(spa, "ADBS 21KZ")

Call ibwrt(spa, "OPR8")

Call ibwrt(spa, "*CLS")
Call ibwrt(spa, "ACP")

Do

 Call ibwrt(spa, "*STB?")
 Rdbuff = Space(8)
 Call ibrd(spa, Rdbuff)
 state = Val(Rdbuff)

 DoEvents
Loop Until (state And 128)

```

- ・ デバイス・クリア
- ・ 中心周波数を 1500MHz に設定する
- ・ スパンを 250kHz に設定する
- ・ RBW を 1kHz に設定する
- ・ VBW を 3kHz に設定する
- ・ 掃引時間を 20 秒に設定
- ・ チャンネルスペースを 50kHz に設定
- ・ 帯域幅を 21kHz に設定
- ・ オペレーション・ステータス・レジスタの Sweep-end ビットをイネーブルにする
- ・ ステータス・バイトをクリアする
- ・ ACP 測定を開始
- ・ ステータス・バイトのクエリ
- ・ 8 バイトの領域確保
- ・ 読み込む
- ・ アスキーを数値に変える
- ・ 他の Windows イベントがあればこのとき実行する
- ・ Measuring ビットが立つまで Do に戻る

## 4.1 GPIB リモート・プログラミング

```

Call ibwrt(spa, "ACP?") ' ACP の測定結果をクエリ

Rdbuf = Space(81) ' 実数 (MAX19Byte)×2 + ', '×1 + CRLF = 41 バイトの領域を
 ' 確保
Call ibrd(spa, Rdbuf) ' 読み込む

sep1 = InStr(1, Rdbuf, ",", 0) ' バッファの頭からの項目セパレータ（ここではカンマ）の
 ' 位置を得る
LOL = Left(Rdbuf, sep1 - 1) ' バッファの先頭からセパレータの前までの文字列を取り出
 ' す

sep2 = InStr(sep1 + 1, Rdbuf, Chr(13), 0) ' ターミネータの位置を得る
UPL = Mid(Rdbuf, sep1 + 1, sep2 - sep1 - 1) ' セパレータ間の文字列を取り出す

 ' 画面に出力
RichTextBox1.Text = "-50kHz:" & LOL & vbCrLf & "50kHz: " & UPL & vbCrLf

```

例 VB-18 シングル掃引の終了ごとにピーク周波数、レベルを読み込む（SRQ を使用）

```

Dim boardID As Integer
Dim I As Integer
Dim res As Integer
Dim CFLEV As String

boardID = 0 ' ボードの ID を設定

Call ibclr(spa) ' デバイス・クリア

Call ibwrt(spa, "SI") ' シングル掃引モードにする

Call ibwrt(spa, "*CLS") ' ステータス・バイト・クリア
Call ibwrt(spa, "OPR 8") ' オペレーション・ステータス・レジスタの掃引終了ビット
 ' を有効にする
Call ibwrt(spa, "*SRE 128") ' ステータス・バイトの Operation status ビットを有効にする
Call ibwrt(spa, "S0") ' SRQ 信号送出モードに設定

For I = 1 To 10 Step 1 ' 10 回のループ
 Call ibwrt(spa, "SI") ' 掃引実行
 Call WaitSRQ(boardID, res) ' SRQ 信号が送信されるまで待つ
 Call ibrsp(spa, res) ' シリアルボール実行

 Call ibwrt(spa, "PS") ' ピーク・サーチ
 Call ibwrt(spa, "MFL?") ' マーカの周波数、レベルのクエリ

 Rdbuf = Space(43) ' 43 バイトの領域を確保
 Call ibrd(spa, Rdbuf) ' 読み込む

 CFLEV = Left(Rdbuf, InStr(1, Rdbuf, Chr(13), 0) - 1)
 RichTextBox1.Text = RichTextBox1.Text & "Freq ,Lebel = " & CFLEV & vbCrLf
 ' 画面に出力して改行

 DoEvents ' その他の Windows のイベントがあれば実行
Next I

```

## (2) HP BASIC のプログラム例 (GPIB アドレス = 8)

## HP-15 シングル掃引を実行し、掃引の終了を待つ (SRQ 信号を使用しない場合)

```

10 Spa=708 ! GPIB アドレス (8) を変数に設定
20 OUTPUT Spa;"SI" ! シングル掃引モード に設定
30 OUTPUT Spa;"OPR8" ! オペレーション・ステータス・レジスタの
40 ! Sweep-end ビットをイネーブルにする
50 OUTPUT Spa;"*CLS" ! ステータス・バイトをクリアする
60 OUTPUT Spa;"SR" ! 掃引を開始
70 Mloop: !
80 OUTPUT Spa;"*STB?" ! ステータス・バイトの出力要求
90 ENTER Spa;S ! ステータス・バイトを読み込む
100 IF BIT(S,7)=0 THEN GOTO Mloop ! オペレーション・ステータス・ビット
110 ! (掃引終了) が 1 にセットされるまで待つ
120 STOP
130 END

```

## HP-16 ACP 測定を行い、測定終了後に結果を読み出す (SRQ 信号を使用しない場合)

```

10 Spa=708 ! GPIB アドレス (8) を変数に設定
20 OUTPUT Spa;"CF1500MZ" ! 中心周波数を 1500MHz に設定
30 OUTPUT Spa;"SP250KZ" ! 周波数スパンを 250kHz に設定
40 OUTPUT Spa;"RB1KZ;VB10KZ" ! RBW:1kHz,VBW:10kHz に設定
50 OUTPUT Spa;"ST20SC" ! 掃引時間を 20 秒に設定
60 OUTPUT Spa;"DTP" ! デテクタ・モードをポジティブに設定
70 OUTPUT Spa;"ADCH50KZ" ! チャンネル・スペースを 50kHz に設定
80 OUTPUT Spa;"ADBS21KZ" ! 帯域幅を 21kHz に設定
90 OUTPUT Spa;"OPR8" ! オペレーション・ステータス・レジスタの
100 ! Sweep-end ビットをイネーブルにする
110 OUTPUT Spa;"*CLS" ! ステータス・バイトをクリアする
120 OUTPUT Spa;"SI" ! シングル掃引モード に設定
130 OUTPUT Spa;"ACP" ! ACP 測定を開始
140 OUTPUT Spa;"SR" ! 掃引を開始
150 Mloop: !
160 OUTPUT Spa;"*STB?" ! ステータス・バイトの出力要求
170 ENTER Spa;S ! ステータス・バイトを読み込む
180 IF BIT(S,7)=0 THEN GOTO Mloop ! ACP 測定終了を待つ
190 OUTPUT Spa;"ACP?" ! ACP 測定結果の出力要求
200 ENTER Spa;Lo,Up ! ACP 測定結果を読み込む
210 PRINT "-50K: ";Lo;" ,+50K: ";Up ! 測定結果を表示
220 END

```

## 4.1 GPIB リモート・プログラミング

HP-17 シングル掃引の終了ごとにピーク周波数、レベルを読み込む  
(SRQ 信号を使用する場合)

|                                                |                                  |
|------------------------------------------------|----------------------------------|
| 10 Spa=708                                     | ! GPIB アドレス (8) を変数に設定           |
| 20 OUTPUT Spa;"SI"                             | ! シングル掃引モードに設定                   |
| 30 ON INTR 7 GOSUB Ssrq                        | ! SRQ 割り込み処理ルーチンを定義              |
| 40 OUTPUT Spa;"*CLS"                           | ! ステータス・バイトをクリアする                |
| 50 OUTPUT Spa;"OPR8"                           | ! オペレーション・ステータス・レジスタの            |
| 60                                             | ! Sweep-end ビットをイネーブルにする         |
| 70 OUTPUT Spa;"*SRE128"                        | ! ステータス・バイトの Operation Status ビッ |
| 80                                             | ! トをイネーブルにする                     |
| 90 OUTPUT Spa;"S0"                             | ! SRQ 信号送出モードを指定                 |
| 100 Mloop:                                     | !                                |
| 110 Mend=0                                     | ! 掃引終了フラグをクリア                    |
| 120 OUTPUT Spa;"SR"                            | ! 掃引を開始                          |
| 130 ENABLE INTR 7;2                            | ! SRQ 割り込みをイネーブルにする              |
| 140 Wint:                                      | !                                |
| 150 IF Mend = 0 THEN GOTO Wint                 | ! SRQ 割り込みが発生するまで待つ              |
| 160 OUTPUT Spa;"PS"                            | ! ピークサーチを実行                      |
| 170 OUTPUT Spa;"MFL?"                          | ! マーカ・データの出力要求                   |
| 180 ENTER Spa:MF,ML                            | ! ピーク周波数, レベルを読み込む               |
| 190 PRINT "Peak Freq: ";MF;" ,Peak Level: ";ML | ! 読み込んだデータを表示                    |
| 200 GOTO Mloop                                 | ! 掃引を繰り返す                        |
| 210                                            | !                                |
| 220 Ssrq:                                      | ! SRQ 割込処理ルーチン                   |
| 230 S=SPOLL(Spa)                               | ! ステータス・バイトを読み込む                 |
| 240 Mend=1                                     | ! 掃引終了フラグを 1 にセット                |
| 250 RETURN                                     | ! メインルーチンに復帰                     |
| 260                                            | !                                |
| 270 END                                        |                                  |



## 4.2 RS-232 リモート・コントロール機能

GPIB インタフェースを装備していないコントローラ（パーソナル・コンピュータなど）でも、RS-232 インタフェースを用いて本器をコントロールすることができます。

### 4.2.1 GPIB リモート・コントロールとの互換性

シリアル・コントロールで使用できるコントロール・コードは、GPIB に特有なコード、機能といくつかのコマンドを除き、本体の GPIB コードと同じものを使用できます。

### 4.2.2 制御可能な機能

シリアル・コントロールを使用すると、以下の機能が制御できます。

- 測定条件の設定：パネル上のキー操作と同様に、各種測定条件の入力ができます。
- 設定状態の出力：本器の各種設定状態と、データの読み出しができます。
- ステータス出力：GPIB と同様に、本器の現在の状態を示すステータス・バイトの読み出しが行えます。

### 4.2.3 パラメータ設定画面

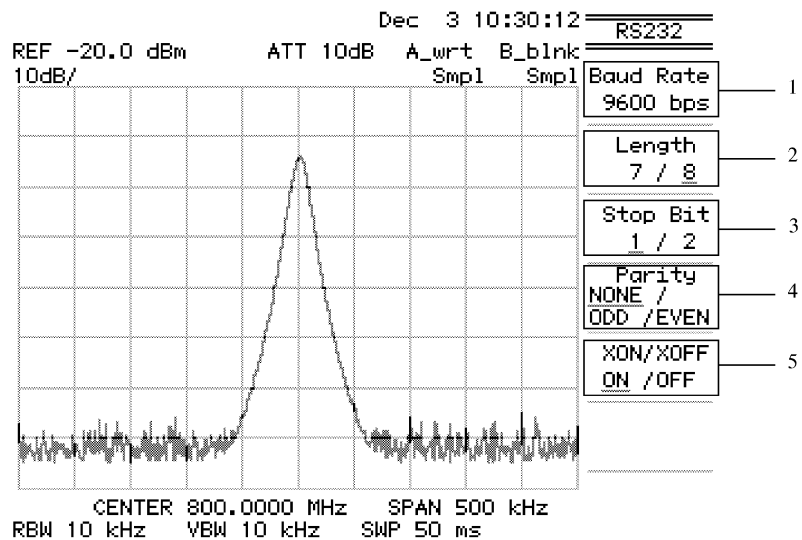


図 4-4 パラメータ設定

1. 転送速度を 600, 1200, 2400, 4800, 9600, 19200 から選択します。
2. データのビット数を 7 ビット、8 ビットのいずれかに選択します。
3. ストップ・ビットを 1 ビット、2 ビットのいずれかに選択します。
4. パリティ・チェックを NONE, ODD, EVEN から選択します。
5. XON/XOFF を使用するかしないかを選択します。

#### 4.2.4 接続方法

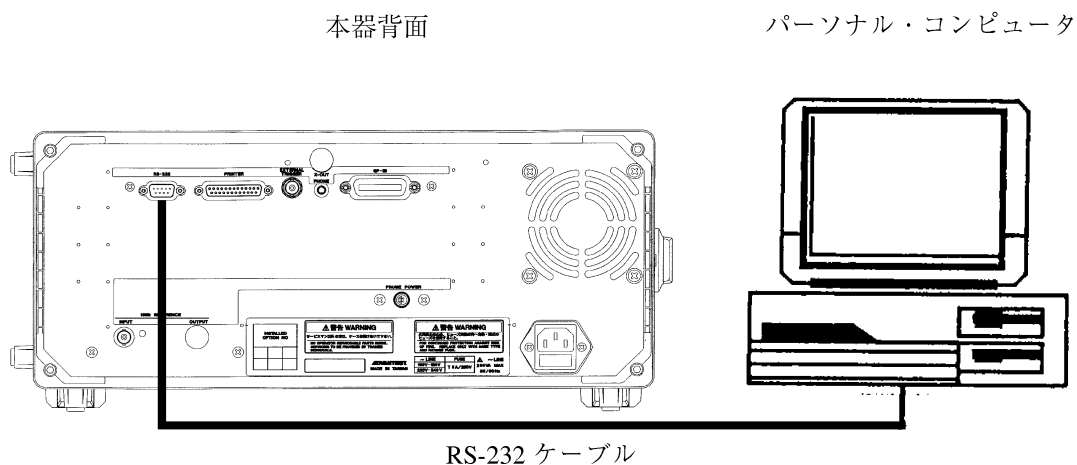


図 4-5 本体とコントローラの接続

本器側は3線ですが、コントロール側（パーソナル・コンピュータ等）は3線では入出力できません。

#### 注意

1. 図 4-6 のケーブル結線図でデータの送受信を実行する場合は、XON/XOFF を有効 (ON) にして本器を使用して下さい。
2. 本器では DCD、DTR、DSR は使用しません。CTS、RTS を使用する場合は、コントローラと本器をクロス結線されたケーブルで接続して下さい。ただし、CTS/RTS でフロー制御は行いません。フロー制御を行う場合は、XON/XOFF を有効にして使用して下さい。



注

1. 転送データは ASCII で行って下さい。
  2. コントローラからのデータの区切りは (CR) または (CR, LF) で送信して下さい。  
クエリ・データは、 GPIB のデリミタと同じになります。そのため、シリアル・ポートをオープンしたあとに DL0 または DL3 を送って下さい (RS-232 リモート・プログラム例参照)。
- 

• 送受信の例

PC からは、

CF 30.0MZCR

CF 30.0MZCR LF

のいずれでも認識します。

クエリ・データのフォーマットは、

+3.0000000000E+07 CR LF

となります (DL0 または DL3 を送る)。

データの区切り (CR, LF) を除く出力データの文字数は、 GPIB と同じです。

#### 4.2.6 GPIB との相違点

- コマンド・コード  
トレース・データの入出力 ASCII フォーマットのみ可能です。
- 

注意 使用できないコマンド :TBA, TBB

---

#### 4.2.7 パネル・コントロール

リモート・コントロール実行時は、以下の仕様になります。

- リモート・ランプを点灯しない。
- キーのロックはされません。コントロール中にキー操作を行って設定を変更した場合、コントロール動作が不安定になる場合があります。

### 4.2.8 リモート・コントロール・プログラム例

実際のプログラムで、リモート・コントロール機能を使用した例です。なお、本項に記載しているプログラム例はすべてマイクロソフト社『Microsoft Quick BASIC』でのプログラム例です。

プログラム例中にある OPEN "COM1:9600,N,8,1,ASC" FOR RANDOM AS #1 は、ボーレート: 9600bps、パリティ: なし、データ長: 8bit、ストップ・ビット: 1bit、ASCII フォーマット、ランダム・アクセス・モードでオープンするコマンドです。

例           ステータス・バイトで掃引終了を待つ

```

OPEN "COM1:9600,N,8,1,ASC" FOR RANDOM AS #1
PRINT #1, "DL3" ' GPIB のデリミタを CR LF にする
PRINT #1, "SI" ' シングル掃引をする
PRINT #1, "OPR8" ' GPIB のオペレーション・レジスタの掃引終了ビットをセット
PRINT #1, "*CLS" ' ステータス・バイトのクリア
PRINT #1, "TS" ' シングル掃引をする
MEAS.LOOP:
PRINT #1, "*STB?" ' ステータス・バイトを読み出す
INPUT #1, STAT
IF (STAT AND 128) = 0 THEN GOTO MEAS.LOOP
PRINT #1, "PS" ' ピーク・サーチ
PRINT #1, "ML?" ' ピークのレベルを読み出す
INPUT #1, MLEVEL
PRINT MLEVEL
END

```

