

## 6. リモート・コントロールの概要

本章では、リモート・コントロール・システムの概要について解説します。

### 6.1 リモート・コントロール

#### 6.1.1 システムの種類

インタフェースの違いにより、下表のような2種類のリモート・コントロール・システムを構成することができます。

表 6-1 インタフェースの種類

インタフェース	概要
GPIB (トーカー/リスナ・モード)	外部コントローラから GPIB 接続された U3800 シリーズ、およびその他の機器をコントロールするシステムです。 詳細は「6.2 GPIB リモート・コントロール・システム」(6-2 ページ) を参照してください。
LAN	外部コントローラから LAN 接続された U3800 シリーズ、およびその他の機器をコントロールするシステムです。 詳細は「6.3 LAN リモート・コントロール・システム」(6-6 ページ) を参照してください。

## 6.2 GPIB リモート・コントロール・システム

本器は、IEEE 規格 488.1-1978 および 488.2-1987 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。

以下、GPIB リモート・コントロール機能を用いたコントロール方法について説明します。

### 6.2.1 GPIB とは

GPIB (General Purpose Interface Bus) は、コンピュータと計測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、各機器が固有の互いに異なる機器アドレスを持つことによって、特定の機器を指定します。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する
  - デバイス・メッセージ：測定器をコントロールする

## 6.2.2 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ください。

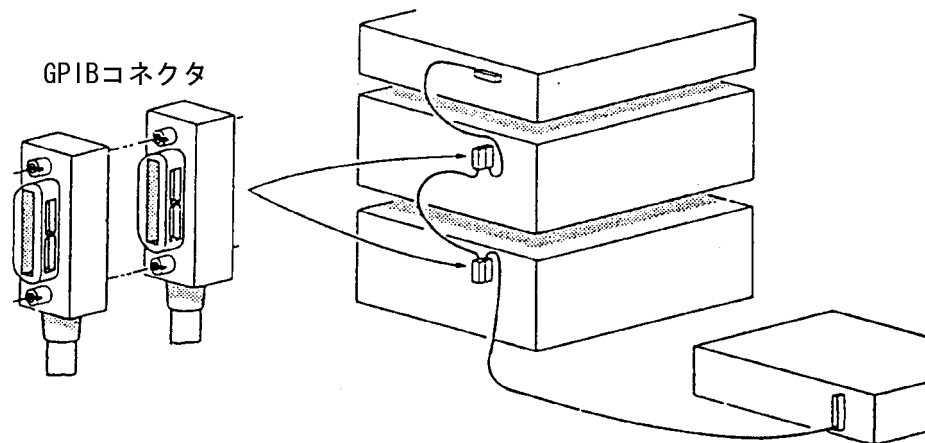


図 6-1 GPIB の接続

GPIB インタフェースの使用時においては、以下のようなことに注意してください。

- 本器背面・パネルの GP-IB 1 コネクタに GPIB ケーブルを接続してください。
- 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、 $2\text{m} \times \{\text{接続される機器の数 (GPIB コントローラも 1 つの機器として数える)}\}$  以下です。  
また、ケーブルの全ケーブル長は 20m 以下とします。
- 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ください。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。

たとえば、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。

- バスに接続されているすべての機器の電源は、必ずオンにしてください。もし、電源をオンにしていない機器があると、システム全体の動作は保証しかねます。
- ケーブルの着脱は、接続されるすべての機器の電源をオフに行ってください。

### 2. GPIB アドレスの設定

GPIB アドレスは、**System** キーを押し、**GPIB Address** で設定します。

## 6.2.3 GPIB バスの機能

### 6.2.3.1 GPIB インタフェース機能

表 6-2 GPIB インタフェース機能

コード	説明
SH1	ソース・ハンドシェイク機能あり
AH1	アクセプタ・ハンドシェイク機能あり
T6	基本的トーカ機能、シリアル・ポール機能、リスナ指定によるトーカ解除機能
TE0	拡張トーカ機能なし
L4	基本的リスナ機能、トーカ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

### 6.2.3.2 インタフェース・メッセージに対する応答

この節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 および 488.2-1987 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取扱説明書を参照してください。

#### 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、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（ローカル・ロック・アウト (LLO) を参照）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. セレクトッド・デバイス・クリア (SDC)

DCL と同一の動作を行います。ただし、SDC は本器がリスナの場合だけ実行されます。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 2 通りあります。

- 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

## 6.3 LAN リモート・コントロール・システム

本器は、IEEE 規格 802.3 に準拠した LAN (Local Area Network) インタフェースを標準装備し、外部コントローラと本器とのソケット通信によりリモート・コントロールが可能です。

以下、LAN リモート・コントロール機能を用いたコントロール方法について説明します。

### 6.3.1 LAN のセットアップ

#### 1. LAN の接続

以下に標準的な LAN の接続を示します。外部コントローラと本器やその他の機器との間で、LAN による通信を行うためには、RJ45 コネクタの 10BASE-T LAN ケーブルを用いて接続します。本器と外部コントローラを直接 LAN ケーブルにて接続する場合には、表 6-3 のような結線をもった LAN ケーブル (クロス・オーバ・ケーブル) を用います。また、本器と外部コントローラ以外に他の機器を LAN にて接続する場合には、イーサネット・ハブなど複数の LAN インタフェースをもった機器を接続するための外部機器を介して接続します。この場合使用する LAN ケーブルは、表 6-4 のような結線を持った LAN ケーブル (ストレート・ケーブル) を用います。

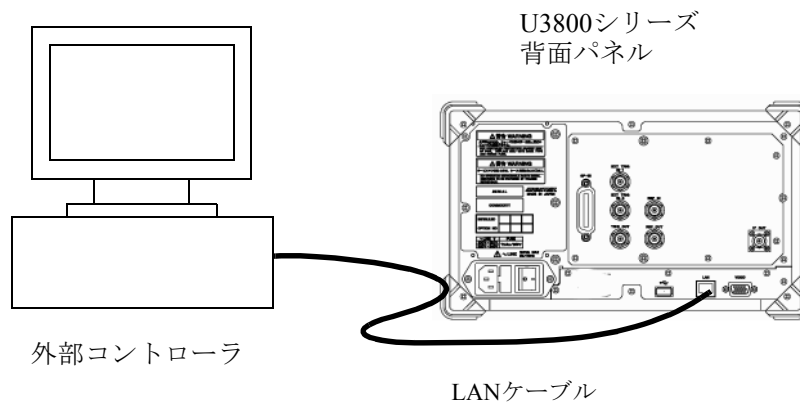


図 6-2 LAN の接続

表 6-3 10BASE-T クロス・オーバ・ケーブルの結線

コネクタ A 側		コネクタ B 側	
信号名	RJ45 ピン番号	RJ45 ピン番号	信号名
RX+	1	3	TX+
RX-	2	6	TX-
TX+	3	1	RX+
TX-	6	2	RX-
Not Used	4	4	Not Used
	5	5	
	6	6	
	7	7	
	8	8	

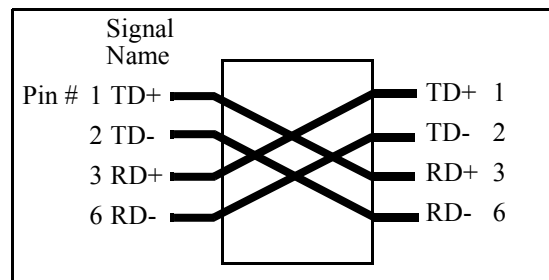


図 6-3 クロス・オーバ・ケーブルの結線図

表 6-4 10BASE-T ストレート・ケーブルの結線

信号名	RJ45 ピン番号	線色	ペア番号
RX+	1	白/橙	2
RX-	2	橙	
TX+	3	白/緑	3
TX-	6	緑	
Not Used	4	青	1
	5	白/青	
	7	白/茶	4
	8	茶	

### 6.3.2 IP アドレスの設定

SYSTEM, Remote Control, LAN IP Address と押します。

1. IP アドレスを手動設定する。  
Use the following IP address をチェックします。

IP Address  
Subnet Mask  
Default Gateway  
を設定します。

The screenshot shows the 'Network Setting' interface. At the top, there are two radio buttons: 'Obtain an IP address automatically' (unchecked) and 'Use the following IP address' (checked). Below the checked option, there are three rows of input fields: 'IP Address' with values 192, 168, 0, 1; 'Subnet Mask' with values 255, 255, 255, 0; and 'Default Gateway' with values 192, 168, 0, 1. At the bottom, the MAC Address is displayed as 00:E0:4B:09:C9:73 and there is an 'Apply' button.

図 6-4 LAN IP Address 設定

Apply ボタンを選択し、Hz を押します。

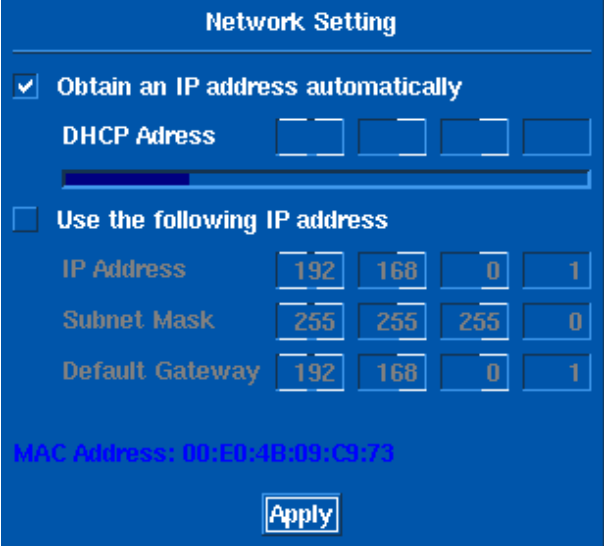
2. IP アドレスを自動取得する。  
Obtain an IP address automatically をチェックします。

The screenshot shows the 'Network Setting' interface. At the top, there are two radio buttons: 'Obtain an IP address automatically' (checked) and 'Use the following IP address' (unchecked). Below the unchecked option, there are three rows of input fields: 'IP Address' with values 192, 168, 0, 1; 'Subnet Mask' with values 255, 255, 255, 0; and 'Default Gateway' with values 192, 168, 0, 1. At the bottom, the MAC Address is displayed as 00:E0:4B:09:C9:73 and there is an 'Apply' button.

図 6-5 LAN IP Address (automatically) 設定 (1)



**Apply** ボタンを選択し、**HZ** を押します。  
アドレスが取得できると、ウィンドウに表示します。



Network Setting

Obtain an IP address automatically

DHCP Address [ ][ ][ ][ ]

Use the following IP address

IP Address [192][168][0][1]

Subnet Mask [255][255][255][0]

Default Gateway [192][168][0][1]

MAC Address: 00:ED:4B:09:C9:73

Apply

図 6-6 LAN IP Address (automatically) 設定 (2)

### 6.3.3 プログラムからのコントロール

外部コントローラのプログラムから本器をコントロールする場合は、ソケット通信のためのポート番号を必要とします。本器側のリモート/コントロール用に用意したソケット通信のためのポート番号は、“5025”です。ソケット通信用のプログラミングを行うには、TCP/IP プロトコルによるネットワーク接続などを行うためのライブラリ（外部コントローラ側の OS 等の環境により異なる）が必要となります。例えば Windows OS 環境では、WinSock が提供されています。

本器とネットワーク接続が完了後、本器に“REN”を送信してリモート制御が可能な状態にします。（このとき、本器正面パネルのリモート・ランプが点灯します。）その後、GPIB と同じコマンドを送ることにより、リモート制御ができます。

また、デリミタは、セミコロン (;) を使用してください。

**GPIB** リモート・コントロール・システムで使用可能な機能の中で、サービス・リクエストなどの GPIB バス特有の一部機能は、LAN リモート・コントロール・システムでは使用できません。

## 6.4 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 バスや LAN 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。

### 6.4.1 各種バッファ

本器にはバッファが 2 つあります。

#### 1. 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 バイトの長さをもつ)

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

#### 2. 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

### 6.4.2 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を、以下に示します。

- クエリの受信によって応答データを生成する（パーサを参照）。
- クエリを実行した順にデータが生成される（応答データ生成を参照）。

パーサ

- 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。

応答データ生成

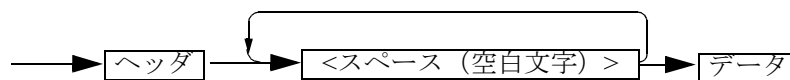
- 本器はパーサがクエリを実行すると、その応答としてデータを出力バッファ上に生成します（つまりデータを出力するにはその直前に必ずクエリを送る必要がある）。

## 6.5 コマンド文法

この章では、コマンド文法について説明します。

### 6.5.1 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、「6.5.2 データ・フォーマット」を参照してください。
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。

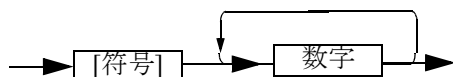
### 6.5.2 データ・フォーマット

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。

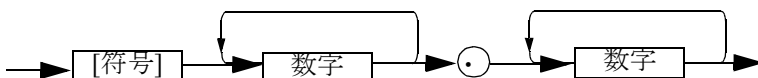
1. 数値データ

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。

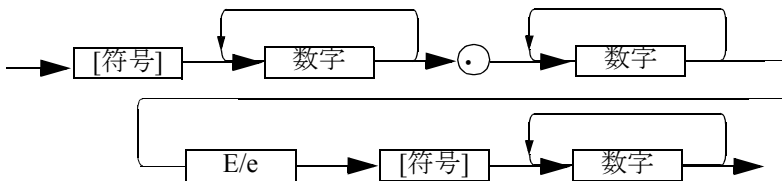
- 整数型：NR1 フォーマット



- 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。

表 6-5 使用可能な単位 (1/2)

単位	指数	意味
GZ	10 <sup>9</sup>	周波数
MZ	10 <sup>6</sup>	周波数
KZ	10 <sup>3</sup>	周波数
HZ	10 <sup>0</sup>	周波数
VOLT	10 <sup>0</sup>	電圧
MV	10 <sup>-3</sup>	電圧
UV	10 <sup>-6</sup>	電圧
NV	10 <sup>-9</sup>	電圧
MW	10 <sup>-3</sup>	電力
DB	10 <sup>0</sup>	dB 関連
MA	10 <sup>-3</sup>	電流
SC	10 <sup>0</sup>	秒
MS	10 <sup>-3</sup>	秒
US	10 <sup>-6</sup>	秒

表 6-5 使用可能な単位 (2/2)

単位	指数	意味
PER	$10^0$	パーセント
%	$10^0$	パーセント

### 6.5.3 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。

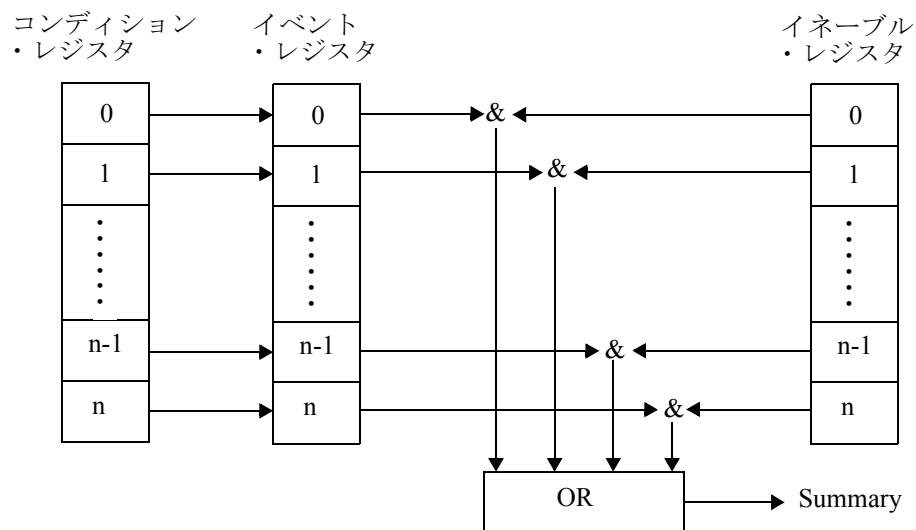


図 6-7 ステータス・バイト・レジスタにセットされる条件

#### 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。

6.5.3 ステータス・バイト

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 6-8 に示します。

ステータス・レジスタの詳細を図 6-9 に示します。

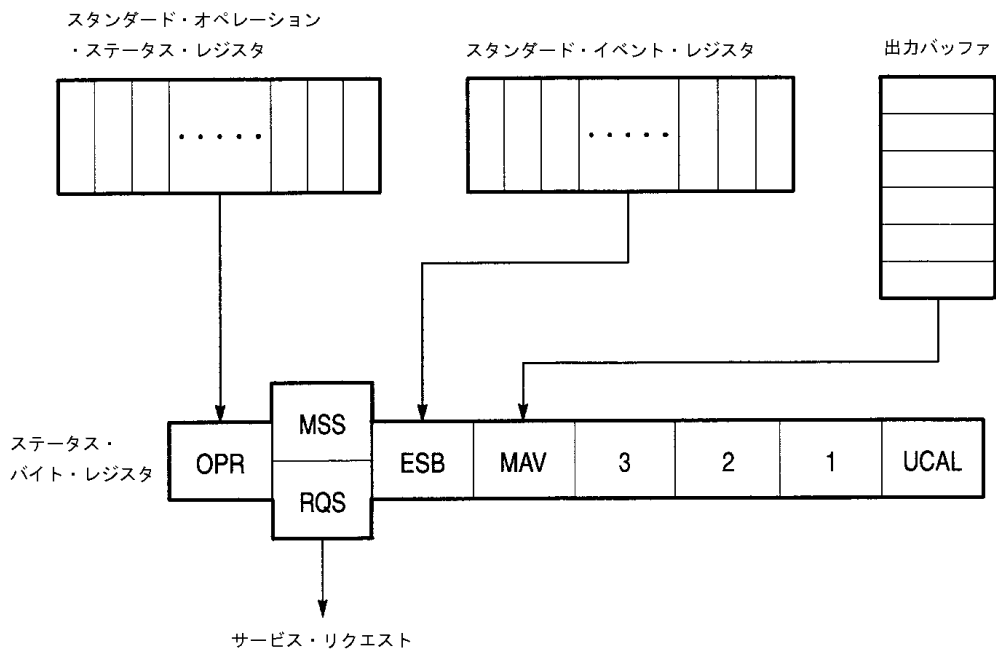


図 6-8 ステータス・レジスタの配置

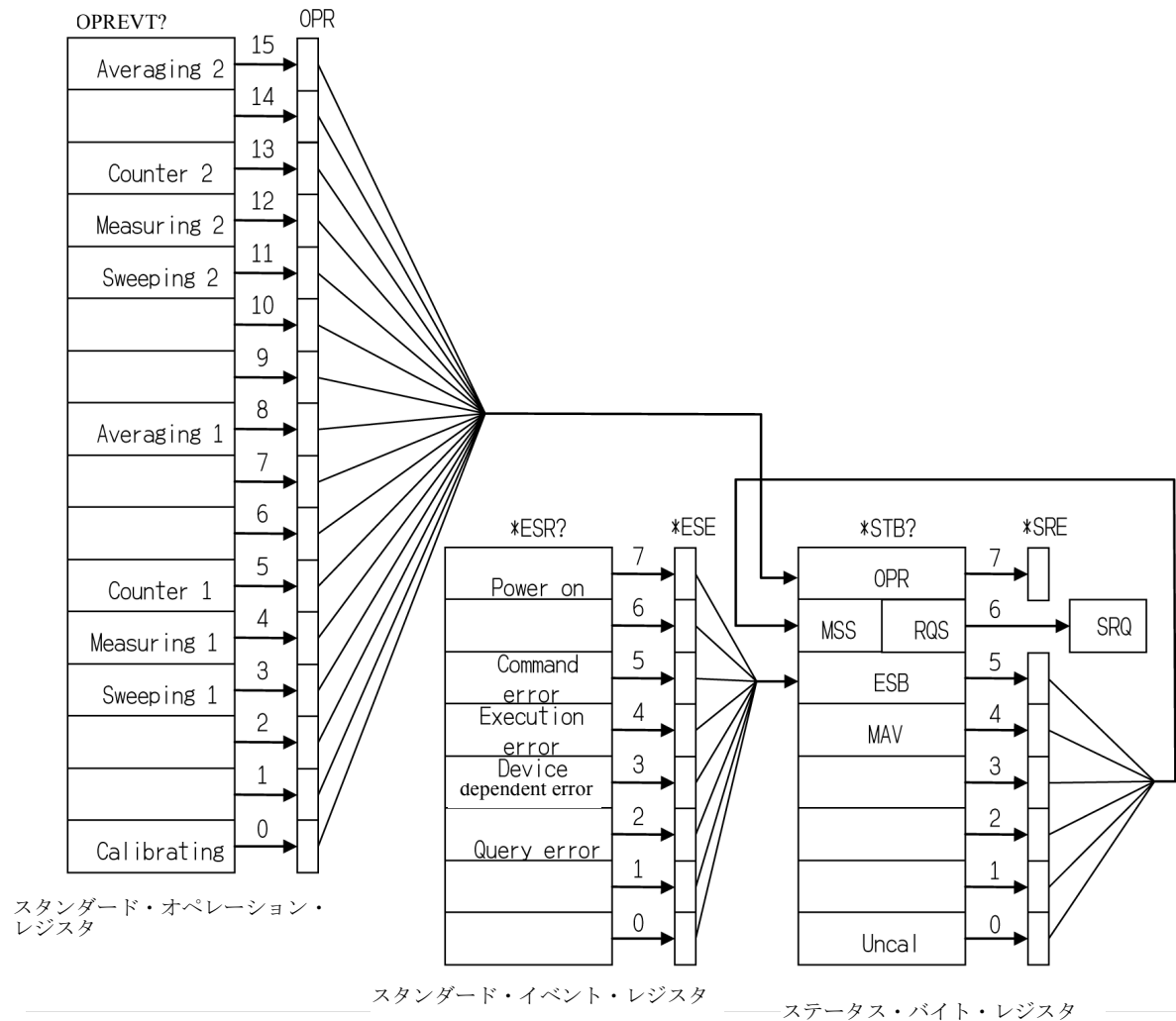


図 6-9 ステータス・レジスタの詳細

## 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

## 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。

## 6.5.3 ステータス・バイト

表 6-6 スタンドアード・オペレーション・ステータス・レジスタ

Bit	機能定義	説明
15	Averaging 2	チャンネル 2 のアベレージ終了時に 1 にセットされる。
14		
13	Counter 2	チャンネル 2 のカウンタ測定終了時に 1 にセットされる。
12	Measuring 2	チャンネル 2 のシーケンス測定終了時に 1 にセットされる。
11	Sweeping 2	チャンネル 2 の掃引終了時に 1 にセットされる。
10		
9		
8	Averaging 1	チャンネル 1 のアベレージ終了時に 1 にセットされる。
7		
6		
5	Counter 1	チャンネル 1 のカウンタ測定終了時に 1 にセットされる。
4	Measuring 1	チャンネル 1 のシーケンス測定終了時に 1 にセットされる。
3	Sweeping 1	チャンネル 1 の掃引終了時に 1 にセットされる。
2		
1		
0	Calibrating	補正データ取得終了時に 1 にセットされる。



## 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 6-10 に示します。

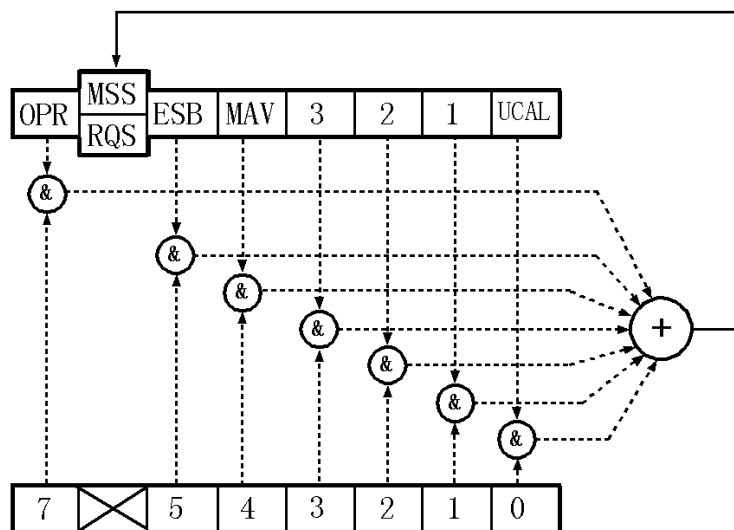


図 6-10 ステータス・バイト・レジスタの構造

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。それにともなって、SRQ ラインも偽になります。

## 6.5.3 ステータス・バイト

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。

表 6-7 ステータス・バイト・レジスタの各ビットの意味

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。

## 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。

表 6-8 スタンダード・イベント・レジスタの各ビットの意味

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。

## 6.6 GPIB リモート・プログラミング

AT コマンド・リストを機能ごとに示します。

- コマンド・コード
  - “\*” は、コードに続いて数値または文字列データを入力するファンクションを表します。
  - [ ] 内のデータは省略可能です。
  
- 出力フォーマット
  - “,” は、複数個のデータを出力することを表します。
  - ON/OFF は、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。
  - レベル単位は設定されている表示単位で出力します。

## 6.7 AT コマンド・インデックス

<n>AVG* <n>GR .....	6-47
<n>AVG* <n>GS .....	6-47
<n>B .....	6-46
<n>G .....	6-46
<n>GC .....	6-47
<n>GCNT .....	6-47
<n>GP .....	6-47
<n>GSGL .....	6-47
<n>MAX .....	6-46
<n>MIN .....	6-46
<n>NORM .....	6-76
<n>PAVG .....	6-47
<n>PGC .....	6-47
<n>PGCNT .....	6-47
<n>PGP .....	6-47
<n>PGSGL .....	6-47
<n>V .....	6-46
<n>W .....	6-46
% PER .....	6-77
*CLS .....	6-73
*ESE .....	6-73
*IDN .....	6-74
*OPT .....	6-74
*SRE .....	6-73
*STB .....	6-73
AA .....	6-42
ACP .....	6-63
ACPCBW .....	6-63
ACPLOAD .....	6-63
ACPNQST .....	6-63

ACPREF .....	6-63
ACPSAVE .....	6-63
ACPSCR .....	6-63
ACPTM .....	6-63
ACTCH .....	6-34
ADG .....	6-63
AL .....	6-43
AMMF .....	6-60
AMMOD .....	6-60
ANNOT .....	6-73
AR .....	6-76
AS .....	6-43
ASV .....	6-68
ASVINTVTM .....	6-68
ASVLMTJDG .....	6-68
ASVMAXSV .....	6-68
ASVMODE .....	6-68
ASVSWPSTEP .....	6-68
ASVTGTCG .....	6-68
ASVTOTALTM .....	6-68
AT .....	6-42
ATMIN .....	6-42
AUNITS .....	6-42
BA .....	6-43
BMP .....	6-69
CA .....	6-40
CALCA .....	6-46
CALCB .....	6-46
CALCC .....	6-46
CALCO .....	6-50
CAPND .....	6-76
CARRBS .....	6-63
CC .....	6-67
CDB .....	6-58
CF .....	6-40
CFCH .....	6-41
CFCHON .....	6-41
CHCALC1 .....	6-41
CHCALC2 .....	6-41
CHCALC3 .....	6-41
CHCON1 .....	6-41
CHCON2 .....	6-41
CHCON3 .....	6-41
CHPDIFON .....	6-40
CHTDEL .....	6-41
CHTIN .....	6-41
CLALL .....	6-67
CLATT .....	6-67
CLCREF .....	6-67
CLDREF .....	6-67
CLGAIN .....	6-67

## 6.7 AT コマンド・インデックス

CLPBW .....	6-67
CLRBW .....	6-67
CLSREF .....	6-67
CN .....	6-61
CN0 .....	6-61
CN1 .....	6-61
CN2 .....	6-61
CN3 .....	6-61
CNPOS .....	6-61
CNPOSA .....	6-61
CNRES .....	6-61
CORS .....	6-43
CORS OFF .....	6-43
CORS ON .....	6-43
COUNT .....	6-61
COVR .....	6-43
COVR OFF .....	6-43
COVR ON .....	6-43
CP .....	6-57
CP1SETTO2 .....	6-34
CP1TBLTO2 .....	6-34
CP1TO2 .....	6-34
CP2SETTO1 .....	6-34
CP2TBLTO1 .....	6-34
CP2TO1 .....	6-34
CPLCH .....	6-34
CR .....	6-66
CR OFF .....	6-42
CR ON .....	6-42
CRDEL .....	6-42, 6-66
CRIN .....	6-42, 6-66
CS .....	6-40
CSBSDEL .....	6-63
CSBSIN .....	6-63
CTXTSEL .....	6-53
DB .....	6-77
DBEMF .....	6-77
DBM .....	6-77
DBMV .....	6-77
DBPW .....	6-77
DBUV .....	6-77
DC0 .....	6-58
DC1 .....	6-58
DC2 .....	6-58
DD .....	6-42
DEL .....	6-68
DET .....	6-47
DETAVG .....	6-48
DETA<n> .....	6-48
DETB .....	6-47
DETC .....	6-48

DISP .....	6-34
DLIM0 .....	6-73
DLIM1 .....	6-73
DLIM2 .....	6-73
DLIM5 .....	6-73
DLN .....	6-52
DLN OFF .....	6-52
DLN ON .....	6-52
DLNFFT .....	6-53
DLNFRE .....	6-53
DLNIQ .....	6-53
DLNPHA .....	6-53
DLNPHF .....	6-38
DLNPWR .....	6-53
DS MTSP .....	6-56
DUAL .....	6-34
DY .....	6-57
E .....	6-77
EMCMEAS .....	6-66
EMCMEASTIM .....	6-66
EMCON EMCDET .....	6-66
ERRNO .....	6-75
FA .....	6-40
FACH .....	6-41
FACHO .....	6-40
FACHON .....	6-41
FACHOON .....	6-40
FB .....	6-40
FBCH .....	6-41
FBCHO .....	6-40
FBCHON .....	6-41
FBCHOON .....	6-40
FC .....	6-67
FILEFORMAT .....	6-68
FILEMEDIA .....	6-68
FINPMD .....	6-40
FMMEAS .....	6-60
FMMODF .....	6-60
FMMODFY .....	6-60
FO .....	6-40
FO ON OFF .....	6-40
FORM1 .....	6-48
FORM2 .....	6-48
FORM3 .....	6-48
FORM4 .....	6-48
FS .....	6-40
FTPOS .....	6-53
FX .....	6-55
GDATA .....	6-69
GIMAG .....	6-69
GTL .....	6-73

## 6.7 AT コマンド・インデックス

GTPOS .....	6-44
GTSLP .....	6-44
GTSRC .....	6-44
GTSWP .....	6-44
GTWID .....	6-44
GZ .....	6-77
HARM .....	6-59
HARMNUM .....	6-59
HCOPY .....	6-72
HRMFND .....	6-59
HRMFND OFF .....	6-59
HRMFND ON .....	6-59
HS .....	6-42
HSCALE .....	6-66
HZ .....	6-77
ID .....	6-74
IMGSP .....	6-41
IMLOAD .....	6-59
IMLS3 .....	6-58
IMLS5 .....	6-58
IMLS7 .....	6-59
IMLS9 .....	6-59
IMM .....	6-58
IMMDF .....	6-58
IMMREF .....	6-58
IMMRES .....	6-58
IMODR .....	6-58
IMPFC .....	6-59
IMSAVE .....	6-59
IP RST .....	6-72
IQBSV .....	6-68
KZ .....	6-77
LARNG .....	6-52
LBRNG .....	6-52
LIMAPOS .....	6-51
LIMAS .....	6-51
LIMASFT .....	6-51
LIMPOS .....	6-51
LIMS .....	6-51
LIMSF .....	6-51
LIMST .....	6-51
LL1 .....	6-42
LLO .....	6-73
LMTA .....	6-51
LMTADELF .....	6-51
LMTADELT .....	6-51
LMTAINF .....	6-51
LMTAINT .....	6-51
LMTB .....	6-51
LMTBDELF .....	6-52
LMTBDELT .....	6-52



LMTBINF .....	6-52
LMTBINT .....	6-52
LMTMRGN .....	6-66
LOF .....	6-72
LON .....	6-72
LTSP LS .....	6-40
M0 MKCS .....	6-56
M1 MTCS .....	6-56
M2 MKMKS .....	6-56
M3 MTMKS .....	6-56
MC MKCF .....	6-56
MDF2 .....	6-54
MDL2 .....	6-54
MEASFUNC .....	6-34
MFL .....	6-54
MFLC .....	6-54
MFLC<n> .....	6-54
MFL<n> .....	6-54
MFR .....	6-54
MF<n> .....	6-54
MIS .....	6-57
MK .....	6-54
MKACF .....	6-56
MKAO .....	6-56
MKBW .....	6-58
MKD .....	6-54
MKDD .....	6-55
MKLST .....	6-55
MKMODE .....	6-55
MKROBJ .....	6-55
MKRSEL .....	6-54
MKSPOS .....	6-57
MKSWID .....	6-57
MKSX .....	6-57
MKSYDL .....	6-57
MKSYLA .....	6-57
MKSYLB .....	6-57
MKTRACE .....	6-55
MKVP .....	6-56
ML .....	6-54
MLN .....	6-54
MLN<n> .....	6-54
MLR .....	6-54
MLSFL .....	6-55
MLTOFF .....	6-54
MLTSCR .....	6-52
ML<n> .....	6-54
MMS .....	6-57
MNRF .....	6-73
MO MKOFF .....	6-54
MPA .....	6-55

## 6.7 AT コマンド・インデックス

MPM .....	6-55
MR MKRL .....	6-56
MS   MSEC .....	6-77
MTAAW .....	6-56
MTASP .....	6-56
MTAW .....	6-56
MTB .....	6-49
MTCF .....	6-56
MTDIV .....	6-56
MTV .....	6-49
MTW .....	6-49
MV .....	6-77
MW .....	6-77
MZ .....	6-77
NI .....	6-58
NIC .....	6-58
NIF .....	6-58
NIM .....	6-58
NION .....	6-58
NIRES .....	6-58
NIU .....	6-58
NORM .....	6-76
NSEC .....	6-77
NV .....	6-77
NXL .....	6-57
NXM .....	6-57
NXP .....	6-57
NXR .....	6-57
OB .....	6-49
OBW .....	6-62
OBWLOAD .....	6-63
OBWON .....	6-62
OBWPER .....	6-62
OBWSAVE .....	6-63
OBWTM .....	6-63
OHM .....	6-42
OPF .....	6-51
OPR .....	6-73
OPREVT .....	6-73
OPT20 .....	6-74
OPT28 .....	6-74
OPT76 .....	6-74
OPT77 .....	6-74
OV .....	6-49
OW .....	6-49
PFC .....	6-51
PFJ .....	6-51
PKACF .....	6-56
PKCF .....	6-56
PKRL .....	6-56
PKZOOM .....	6-41

PLS FREQ .....	6-57
PLS LEVEL .....	6-57
PMEASAVG .....	6-61
PMEASAVGONCE .....	6-61
PMEASMODE .....	6-61
PMEASOFF .....	6-61
PMEASTM .....	6-61
PMEASTRACE .....	6-62
PNG .....	6-69
PPM .....	6-77
PS .....	6-57
PSXDB .....	6-58
PU .....	6-60
PWAVG .....	6-62
PWAVGLOAD .....	6-62
PWAVGON .....	6-62
PWAVGRANGE .....	6-62
PWAVGSAVE .....	6-62
PWAVGTM .....	6-62
PWCH .....	6-62
PWCHLOAD .....	6-62
PWCHON .....	6-62
PWCHPSD .....	6-62
PWCHSAVE .....	6-62
PWCHTM .....	6-62
PWTOTAL .....	6-62
PWTOTALON .....	6-62
PWTOTALPSD .....	6-62
PWTOTALTM .....	6-62
PWTOTLOAD .....	6-62
PWTOTSAVE .....	6-62
QP0 .....	6-66
QP1 .....	6-66
QP2 .....	6-66
QP3 .....	6-66
QPAUTO QA .....	6-66
QRC .....	6-68
QSV .....	6-68
RB .....	6-43
RC .....	6-68
RCSET .....	6-68
RCTBL .....	6-69
RCTRC .....	6-68
RCV CORR .....	6-69
RECALCON .....	6-36, 6-37, 6-39, 6-72
REDLT .....	6-55
REN .....	6-73
RENAME .....	6-68
RF .....	6-72
RFACT .....	6-63
RFC .....	6-72

6.7 AT コマンド・インデックス

RFE .....	6-72
RFI .....	6-72
RFX .....	6-72
RL .....	6-42
RLN .....	6-52
RLN OFF .....	6-52
RLN ON .....	6-52
RLNFFT .....	6-53
RLNFRE .....	6-53
RLNIQ .....	6-53
RLNPHA .....	6-53
RLNPHF .....	6-38
RLNPWR .....	6-53
RO .....	6-42
RO ON OFF .....	6-42
RX .....	6-76
S0 .....	6-73
S1 .....	6-73
S2 .....	6-73
SAM .....	6-60
SC .....	6-77
SCRF .....	6-73
SDV .....	6-60
SEM .....	6-64
SEMCBW .....	6-63
SEMLOAD .....	6-63
SEMNQST .....	6-63
SEMON .....	6-63
SEMRFCALC .....	6-63
SEMRFPOW .....	6-64
SEMSAVE .....	6-63
SEMTDEL .....	6-64
SEMTIN .....	6-64
SEMTM .....	6-63
SETDATE .....	6-74
SETTIME .....	6-74
SFM .....	6-60
SG .....	6-55
SGY .....	6-55
SGY OFF .....	6-55
SGY ON .....	6-55
SIGID .....	6-41
SOF .....	6-60
SON .....	6-60
SP .....	6-40
SPRIN SPRFIN .....	6-64
SPRTBL .....	6-64
SPURI .....	6-64
SQL .....	6-38
SQLLVL .....	6-38
STORE .....	6-47

SUPIP .....	6-72
SV .....	6-68
SW ST .....	6-43
SWM .....	6-43
SWPABORT .....	6-44
SWPCNT .....	6-46
SYMRT .....	6-63
SYNCTRIG .....	6-45
SYSMODE .....	6-34
TAA .....	6-48
TAB .....	6-48
TAC .....	6-48
TAM .....	6-49
TAO .....	6-49
TBA .....	6-48
TBA1 .....	6-48
TBA2 .....	6-48
TBB .....	6-48
TBB1 .....	6-48
TBB2 .....	6-48
TBC .....	6-48
TBC1 .....	6-48
TBC2 .....	6-48
TBLRC .....	6-69
TBLSV .....	6-69
TBM .....	6-49
TBM1 .....	6-49
TBM2 .....	6-49
TBO .....	6-49
TBO1 .....	6-50
TBO2 .....	6-50
TG .....	6-76
TGDLTRLN .....	6-76
TGF .....	6-76
TGL .....	6-76
TGLO .....	6-76
TGLO ON OFF .....	6-76
TGO .....	6-76
TGO ON OFF .....	6-76
THRESH .....	6-38
THRFFT .....	6-38
THRFRE .....	6-38
THRIQ .....	6-38
THRPHA .....	6-38
THRPHF .....	6-38
THRPWR .....	6-38
TN .....	6-41
TPL TP .....	6-47
TPS TP .....	6-47
TRACESEL .....	6-48
TRGDLY .....	6-45

## 6.7 AT コマンド・インデックス

TRGLVL .....	6-45
TRGOUTCH .....	6-45
TRGOUTTHR .....	6-45
TRGSLP .....	6-45
TRGSRC .....	6-45
TRGTTL2LVL .....	6-45
TRGTTLVL .....	6-45
TRSUB .....	6-49
TRXSUB12 .....	6-50
TRXSUB21 .....	6-50
TRXSUBOFF .....	6-49
TST .....	6-74, 6-75
T<n> .....	6-46
US   USEC .....	6-77
UV .....	6-77
V   VOLT .....	6-77
VA .....	6-43
VB .....	6-43
VCDEFCND .....	6-35, 6-37, 6-39
VCFA .....	6-35, 6-36, 6-39, 6-82
VCFB .....	6-35, 6-36, 6-39, 6-82
VCFSEXTABORT .....	6-82
VCFSEXTEXE .....	6-82
VCFSEXTFINISH .....	6-82
VCFSEXTIDX .....	6-82
VCFSEXTINIT .....	6-82, 6-83
VCFSEXTSTPSZ .....	6-82
VCFSTEP .....	6-35, 6-36, 6-39, 6-82
VCINBEXE .....	6-35, 6-37, 6-39
VCINBEXTABORT .....	6-83
VCINBEXTEXE .....	6-83
VCINBEXTFINISH .....	6-83
VCINBEXTFO .....	6-83
VCINBEXTIDX .....	6-83
VCINBEXTINIT .....	6-83
VCINBEXTSTPSZ .....	6-83
VCORR .....	6-35, 6-36, 6-39
VCSIGLVLADJ .....	6-35, 6-37, 6-39
VCSIGSRC .....	6-35, 6-37, 6-39
VCSMPLEXE .....	6-35, 6-36, 6-39
VCSPANEXE .....	6-35, 6-36, 6-39
VCSTAT .....	6-35, 6-37, 6-39
VPOSPHF .....	6-38
VS .....	6-42
VSCLPHF .....	6-38
VSWR .....	6-76
VSWRLOSS .....	6-76
VSWRON .....	6-76
W   WATT .....	6-77
WDOSWP .....	6-52
WDX .....	6-52

WLX .....	6-52
WP .....	6-68
X1 .....	6-73
X2 .....	6-73
XAOFS .....	6-36, 6-39
XATIM .....	6-36, 6-38
XCAPBW .....	6-35, 6-36, 6-37
XCAPTIM .....	6-35, 6-37
XCAPTTIM .....	6-36
XCH1MAG .....	6-37
XCH1PHA .....	6-37
XCH1TIM .....	6-37
XCH2MAG .....	6-37
XCH2PHA .....	6-38
XCH2TIM .....	6-38
XDB .....	6-58
XDIFFON .....	6-36
XDISP .....	6-36, 6-38
XDL .....	6-58
XDOM .....	6-38
XDR .....	6-58
XLHSCH .....	6-38
XMATHON .....	6-37
XMATHOP .....	6-38
XPRPDAOFS .....	6-35
XPRPDATIM .....	6-35
XPRPDON .....	6-35
XREFCH .....	6-35, 6-36
XRHSCH .....	6-38
XSCRSEL .....	6-36, 6-37
ZAT .....	6-42
ZMPOS .....	6-53
ZMWID .....	6-53
ZS .....	6-40

## 6.8 Dual CH 時の TS コマンドの動作

TS (Take Sweep)

TS コマンドを実行すると指定されたチャンネルはシングル掃引モードとなり、掃引を開始します。掃引の間、GPIB バスはホールド状態を保ち、次のコマンドは掃引終了を待って実行されます。

### 6.8.1 動作

#### 1. TS

アクティブ・チャンネルの掃引をスタートさせ、アクティブ・チャンネルの掃引終了まで GPIB バスをホールドします。

CH1 (アクティブCH)

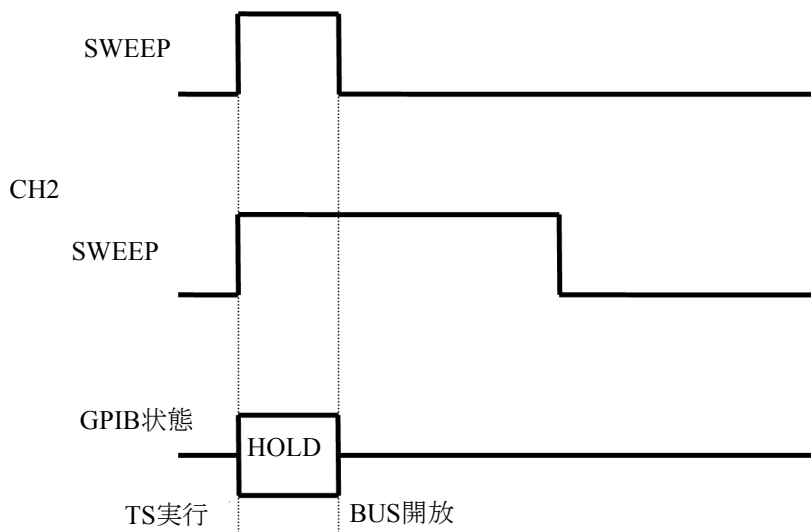


図 6-11 Dual CH 時の TS コマンドのタイミング・チャート

---

注 非アクティブ CH は TS に関係しません。

---



## 2. TSM

CH1, CH2 同時に掃引をスタートさせ、両チャンネルの掃引終了まで GPIB バスをホールドします。

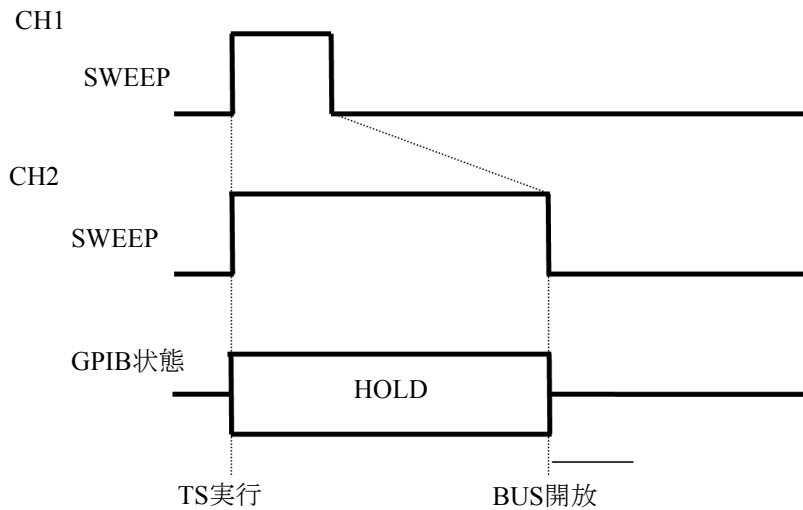


図 6-12 Dual CH 時の TSM コマンドのタイミング・チャート

## 3. TSS

CH1, CH2 同時に掃引をスタートさせ、どちらか掃引終了の早いチャンネルが掃引終了するまで GPIB バスをホールドします。

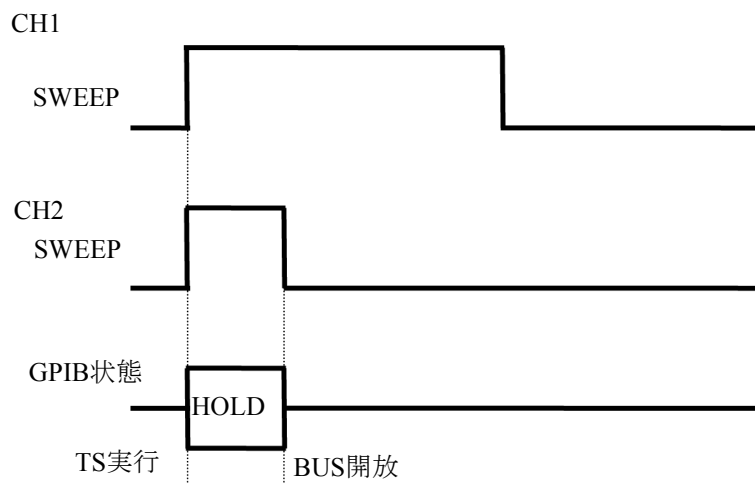


図 6-13 Dual CH 時の TSS コマンドのタイミング・チャート

## 6.9 AT コマンド・コード一覧

## 6.9.1 Mode

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Dual Channel	DUAL*	ON OFF	DUAL?	0 = OFF 1 = ON
2	Active Channel Select	ACTCH*	CH1 CH2	ACTCH?	1 = CH1 2 = CH2
3	Couple Channel	CPLCH*	ON OFF	CPLCH?	0 = OFF 1 = ON
4	Display mode	DISP*	SPLIT FULL1  FULL2 OLAY	DISP?	0 = SPLIT 1 = FULL1 2 = FULL2 3 = OLAY
5	Copy Settings Copy CH1 → CH2 Copy CH2 → CH1	CP1TO2 CP2TO1	---	---	---
6	Copy Table Copy CH1 → CH2 Copy CH2 → CH1 Copy Setup Copy CH1 → CH2 Copy CH2 → CH1	CPITBLTO2 CP2TBLTO1 CP1SETTO2 CP2SETTO1	---	---	---
7	SYSTEM MODE	SYSMODE*	SYNC ASYNCH  SCH1 SCH2	SYSMODE?	0 = DualSyncCH 1 = DualAsyncCH 2 = Single CH1 3 = Single CH2
8	MEAS1	MEASFUNC*	FRE PHA PWR  IQ FFT SPA	MEASFUNCRES?	0 = Freq-Time 1 = Phase-Time 2 = Power-Time 3 = Voltage-Time 4 = Power-Freq 9 = SPA

## 6.9.2 Power Ratio Phase Diff

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Execute	XPRPDON*	ON OFF	XPRPDON?	0 = OFF 1 = ON
2	Ref CH	XREFCH*	CH1 CH2	XREFCH?	1 = CH1 2 = CH2
3	Capture Time	XCAPTIM*	Time	XCAPTIM?	Time
4	Capture BW	XCAPBW*	Frequency	XCAPBW?	Frequency
5	Analysis Window Width	XRPDATIM*	Time	XRPDATIM?	Time
6	Analysis Window Offset	XRPDAOFS*	Time	XRPDAOFS?	Time
7	Vector Correction ON OFF	VCORR*	ON OFF	VCORR?	0 = OFF 1 = ON
8	Vector Correction Current Frequency Get Correction Data	VCSMPLEXE	---	---	---
9	Vector Correction Specific Span Get Correction Data	VCSPANEXE	---	---	---
10	Vector Correction Start Frequency	VCFA*	Frequency	VCFA?	Frequency
11	Vector Correction Stop Frequency	VCFB*	Frequency	VCFB?	Frequency
12	Vector Correction Step Frequency	VCFSTEP*	Frequency	VCFSTEP?	Frequency
13	Vector Correction Default Conditions	VCDEFCND	---	---	---
14	Vector Correction Inband Get Correction Data	VCINBEXE	---	---	---
15	Correction Signal Level Adjust	VCSIGLVLADJ*	ON OFF	VCSIGLVLADJ?	0 = OFF 1 = ON
16	Signal Source INT EXT	VCSIGSRC*	INT EXT	VCSIGSRC?	0 = INT 1 = EXT
17	VCSTAT	---	---	VCSTAT?	0 = OFF 1 = ON <0 = Error

## 6.9.3 Differential

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
18	Re-Calculation ON/OFF	RECALCON*	ON OFF	RECALCON?	0 = OFF 1 = ON

## 6.9.3 Differential

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Execute	XDIFFON*	ON OFF	XDIFFON?	0 = OFF 1 = ON
2	Ref CH	XREFCH*	CH1 CH2	XREFCH?	1 = CH1 2 = CH2
3	Capture Time	XCAPTIM*	Time	XCAPTIM?	Time
4	Capture BW	XCAPBW*	Frequency	XCAPBW?	Frequency
5	Screen Select	XSCRSEL*	SCR1 SCR2	XSCRSEL?	1 = Screen1 2 = Screen2
6	Screen1/2 Display Type	XDISP*	FRE PHA PWR  FFT	XDISP?	0 = Freq-Time 1 = Phase-Time 2 = Power-Time 4 = Power-Freq
7	Screen1/2 Analysis Window Width	XATIM*	Time	XATIM?	Time
8	Screen1/2 Analysis Window Offset	XAOFS*	Time	XAOFS?	Time
9	Vector Correction ON/OFF	VCORR*	ON OFF	VCORR?	0 = OFF 1 = ON
10	Vector Correction Current Frequency Get Correction Data	VCMPLEXE	---	---	---
11	Vector Correction Specific Span Get Correction Data	VCSPANEXE	---	---	---
12	Vector Correction Start Frequency	VCFA*	Frequency	VCFA?	Frequency
13	Vector Correction Stop Frequency	VCFB*	Frequency	VCFB?	Frequency
14	Vector Correction Step Frequency	VCFSTEP*	Frequency	VCFSTEP?	Frequency

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
15	Vector Correction Default Conditions	VCDEFCND	---	---	---
16	Vector Correction Inband Get Correction Data	VCINBEXE	---	---	---
17	Correction Signal Level Adjust	VCSIGLVLADJ*	ON OFF	VCSIGLVLADJ?	0 = OFF 1 = ON
18	Signal Source INT EXT	VCSIGSRC*	INT EXT	VCSIGSRC?	0 = INT 1 = EXT
19	VCSTAT	---	---	VCSTAT?	0 = OFF 1 = ON <0 = Error
20	Re-Calculation ON OFF	RECALCON*	ON OFF	RECALCON?	0 = OFF 1 = ON

## 6.9.4 Math

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Execute	XMATHON*	ON OFF	XMATHON?	0 = OFF 1 = ON
2	Capture Time	XCAPTIM*	Time	XCAPTIM?	Time
3	Capture BW	XCAPBW*	Frequency	XCAPBW?	Frequency
4	Screen Select	XSCRSEL*	SCR1 SCR2	XSCRSEL?	1 = Screen1 2 = Screen2
5	CH1 Config MagnitudeMultiplier	XCH1MAG*	ON OFF	XCH1MAGON?	0 = OFF 1 = ON
		XCH1MAG*	Real	XCH1MAG?	Real
6	CH1 Config PhaseShift	XCH1PHA*	ON OFF	XCH1PHAON?	0 = OFF 1 = ON
		XCH1PHA*	Phase(degree)	XCH1PHA?	Phase(degree)
7	CH1 Config TimeShift	XCH1TIM*	ON OFF	XCH1TIMON?	0 = OFF 1 = ON
		XCH1TIM*	Time	XCH1TIM?	Time
8	CH2 Config MagnitudeMultiplier	XCH2MAG*	ON OFF	XCH2MAGON?	0 = OFF 1 = ON
		XCH2MAG*	Real	XCH2MAG?	Real

## 6.9.4 Math

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
9	CH2 Config PhaseShift	XCH2PHA*	ON OFF	XCH2PHAON?	0 = OFF 1 = ON
		XCH2PHA*	Phase(degree)	XCH2PHA?	Phase(degree)
10	CH2 Config TimeShift	XCH2TIM*	ON OFF	XCH2TIMON?	0 = OFF 1 = ON
		XCH2TIM*	Time	XCH2TIM?	Time
11	Math Config left-hand side	XLHSCH*	CH1 CH2	XLHSCH?	1 = CH1 2 = CH2
12	Math Config math operator	XMATHOP*	ADD SUB MUL  DIV	XMATHOP?	0 = Addition 1 = Subtraction 2 = Multiplication 3 = Division
13	Math Config right-hand side	XRHSCH*	CH1 CH2	XRHSCH?	1 = CH1 2 = CH2
14	Domain	XDOM*	XTIM XFRE	XDOM?	0 = Time 1 = Frequency
15	Display Type	XDISP*	FRE PHA PWR  FFT PHF	XDISP?	0 = Freq-Time 1 = Phase-Time 2 = Power-Time 4 = Power-Freq 5 = Phase-Freq
16	Vertical Scale Phase-Freq	VSCLPHF*	Phase	VSCLPHF?	Phase
17	Vertical Position Phase-Freq	VPOSPHF*	Position	VPOSPHF?	Position
18	Display Line Phase-Freq	DLNPHF*	Phase	DLNPHF?	Phase
19	Reference Line Phase-Freq	RLNPHF*	Phase	RLNPHF?	Phase
20	Threshold	THRIQ*	Volt	THRIQ?	Volt
		THRPWR*	dBm	THRPWR?	dBm
		THRFRE*	Frequency	THRFRE?	Frequency
		THRPHA*	Phase	THRPHA?	Phase
		THRFFT*	dBm	THRFFT?	dBm
		THRPHF*	Phase	THRPHF?	Phase
		THRESH*	ON OFF	THRESHON?	0=OFF, 1=ON
21	Squelch	SQL*	ON OFF	SQL?	0 = OFF 1 = ON
		SQLLVL*	Real	SQLLVL?	Real(Level)
22	Analysis Window Width	XATIM*	Time	XATIM?	Time

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
23	Analysis Window Offset	XAOFS*	Time	XAOFS?	Time
24	Vector Correction ON OFF	VCORR*	ON OFF	VCORR?	0 = OFF 1 = ON
25	Vector Correction Current Frequency Get Correction Data	VCSPMPLEXE	---	---	---
26	Vector Correction Specific Span Get Correction Data	VCSPANEXE	---	---	---
27	Vector Correction Start Frequency	VCFA*	Frequency	VCFA?	Frequency
28	Vector Correction Stop Frequency	VCFB*	Frequency	VCFB?	Frequency
29	Vector Correction Step Frequency	VCFSTEP*	Frequency	VCFSTEP?	Frequency
30	Vector Correction Default Conditions	VCDEFCOND	---	---	---
31	Vector Correction Inband Get Correction Data	VCINBEXE	---	---	---
32	Correction Signal Level Adjust	VCSIGLVLADJ*	ON OFF	VCSIGLVLADJ?	0 = OFF 1 = ON
33	Signal Source INT EXT	VCSIGSRC*	INT EXT	VCSIGSRC?	0 = INT 1 = EXT
34	VCSTAT	---	---	VCSTAT?	0 = OFF 1 = ON <0 = Error
35	Re-Calculation ON OFF	RECALCON*	ON OFF	RECALCON?	0 = OFF 1 = ON

### 6.9.5 Ch Power Diff

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Execute	CHPDIFON*	ON OFF	CHPDIFON?	0 = OFF 1 = ON
2	Ch Pow Diff Result	--	--	CHPDIF?	Lev

### 6.9.6 Frequency

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Center Frequency	CF*	Frequency	CF?	Frequency
2	CF Step Size	CS*	Frequency	CS?	Frequency
3	CF Step Auto	CA[*]	[ON] OFF	CA?	0 = OFF (manual) 1 = ON (auto)
4	Frequency Offset	FO [ON,]* FO ON OFF	Frequency	FO? FOON?	Frequency 0 = OFF 1 = ON
5	Start Frequency	FA*	Frequency	FA?	Frequency
6	Stop Frequency	FB*	Frequency	FB?	Frequency
7	Frequency Span	SP*	Frequency	SP?	Frequency
8	Full Span	FS	---	---	---
9	Zero Span	ZS	---	---	---
10	Last Span	LTSP LS	---	---	---
11	Frequency Setting Mode	FINPMD*	CALC TBL	FINPMD?	0 = CALC 1 = TBL
12	Set Start Channel Offset	FACHO*	Frequency	FACHO?	Frequency
13	Set Stop Channel Offset	FBCHO*	Frequency	FBCHO?	Frequency
14	Start Channel Offset	FACHOON*	ON OFF	FACHOON?	0 = OFF 1 = ON
15	Stop Channel Offset	FBCHOON*	ON OFF	FBCHOON?	0 = OFF 1 = ON



No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
16	Set Center Channel	CFCH*	整数	CFCH?	整数 (チャンネル番号)
17	Set Start Channel	FACH*	整数	FACH?	整数 (チャンネル番号)
18	Set Stop Channel	FBCH*	整数	FBCH?	整数 (チャンネル番号)
19	Center Channel Setting	CFCHON*	ON OFF	CFCHON?	0 = OFF 1 = ON
20	Start Channel Setting	FACHON*	ON OFF	FACHON?	0 = OFF 1 = ON
21	Stop Channel Setting	FBCHON*	ON OFF	FBCHON?	0 = OFF 1 = ON
22	Channel Type 1 Input3 Formulas :	CHCALC1 *,*,*,* CHCALC2 *,*,*,* CHCALC3 *,*,*,*	整数, 整数, 周波数, 周波数, 整数	---	---
23	Formula 1 for Type 1	CHCON1*	ON OFF	CHCON1?	0 = OFF 1 = ON
24	Formula 2 for Type 1	CHCON2*	ON OFF	CHCON2?	0 = OFF 1 = ON
25	Formula 3 for Type 1	CHCON3*	ON OFF	CHCON3?	0 = OFF 1 = ON
26	Channel Type 2 Input	CHTIN*,*	整数, 周波数	---	---
27	Channel Type 2 Deletion	CHTDEL	---	---	---
28	Signal Ident	SIGID*	ON OFF	SIGID?	0 = OFF 1 = ON
29	Image Suppress	IMGSP*	ON OFF	IMGSP?	0 = OFF 1 = ON
30	Auto Tune	TN	---	---	---
31	Peak Zoom	PKZOOM	---	---	---

## 6.9.7 Level

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Reference Level	RL*	Level	RL?	Level
2	Attenuation	AT*	DB (整数)	AT?	DB (整数)
3	Min Attenuation	ATMIN*	DB (整数)	ATMIN?	DB (整数)
4	Zero Attenuation	ZAT	---	---	---
5	Attenuation Auto	AA[*]	[ON] OFF	AA?	0 = OFF (manual) 1 = ON (auto)
6	XdB/Div	DD*	DB (Discr. Val.: 10, 5, 2, 1, 0.5dB)	DD?	0 = 10dB 1 = 5dB 2 = 2dB 3 = 1dB 4 = 0.5dB -1 = others
				DDB?	DB
7	Linear × 1	LL1	---	---	---
8	Vertical Scale	VS*	LIN LOG	VS?	0 = LOG 1 = LIN
9	Level Offset	RO* RO ON OFF	DB	RO?	DB
				ROON?	0 = OFF 1 = ON
10	Hi Sens	HS[*]	[ON] OFF	HS?	0 = OFF 1 = ON
11	Input	OHM*	整数 (Discr. Val.: 50, 75)	OHM?	整数
12	Display Unit (Level Unit)	AUNITS*	DBM DBMV  DBUV DBEMF  DBPW W[ATT]  V[OLT]	AUNITS?	0 = DBM 1 = DBMV 2 = DBUV 3 = DBEMF 4 = DBPW 5 = WATT 6 = VOLT
13	Correction Factor ON OFF	CR ON CR OFF	--- ---	CRON?	0 = OFF 1 = ON
14	Table Input	CRIN*,*	Frequency, Level (DB)	---	---
15	Table Delete	CRDEL	---	---	---

## 6.9.8 Bandwidth

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	RBW	RB*	Frequency	RB?	Frequency
2	RBW Auto	BA[*]	[ON] OFF	BA?	0 = OFF 1 = ON
3	VBW	VB*	Frequency	VB?	Frequency
4	VBW Auto	VA[*]	[ON] OFF	VA?	0 = OFF (manual) 1 = ON (auto)
5	Couple All Auto	AL[*]	[ON] OFF	AL?	0 = OFF 1 = ON (all auto)
6	RBW : Span	CORS* CORS ON[*] CORS OFF	Ratio (float)	CORS?	Ratio (float)
				CORSON?	0 = OFF 1 = ON
7	VBW : RBW	COVR* COVR ON[*] COVR OFF	Ratio (float)	COVR?	Ratio (float)
				COVRON?	0 = OFF 1 = ON

## 6.9.9 Sweep

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Sweep Time	SW* ST*	Time	SW? ST?	Time
2	Sweep Auto	AS[*]	[ON] OFF	AS?	0 = OFF (manual) 1 = ON (auto)
3	Sweep Mode	---	---	SWM?	0 = Single 1 = Normal

## 6.9.9 Sweep

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
4	Sweep Mode Normal				
	active Ch	SN CONTS	---	---	---
	both Ch	SNALL  CONTSALL	---	---	---
	Ch1	SN1 CONTS1	---	---	---
	Ch2	SN2 CONTS2	---	---	---
5	Sweep Mode Single				
	active Ch	SI SNGLS	---	---	---
	both Ch	SIALL  SNGLSALL	---	---	---
	Ch1	SI1 SNGLS1	---	---	---
	Ch2	SI2 SNGLS2	---	---	---
6	Take Sweep				
	active Ch	TS	---	---	---
	Ch1(*1)	TSM	---	---	---
	Ch2(*2)	TSS	---	---	---
7	Sweep Start / Stop				
	active Ch	SR	---	---	---
	both Ch	SRALL	---	---	---
	Ch1(*1)	SR1	---	---	---
	Ch2(*2)	SR2	---	---	---
8	Sweep Abort	SWPABORT	---	---	---
9	Gated Sweep Mode	GTSWP*	ON OFF	GTSWP?	0 = OFF 1 = ON
10	Gate Source	GTSRC*	EXT IF	GTSRC?	2 = EXT 3 = IF 5 = XIF 6 = EXT2
11	Gate Slope	GTSLP*	(FALL NEG -)  (RISE POS +)	GTSLP?	0= RISE POS + 1= FALL NEG -
12	Gate Delay	GTPOS*	Time	GTPOS?	Time
13	Gate Width	GTWID*	Time	GTWID?	Time

(\*1) 両方のチャンネルが Sweep end したら終了

(\*2) どちらかのチャンネルが Sweep end したら終了

## 6.9.10 Trigger

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Trigger Mode	TRGSRC*	FREE IF EXT  VIDEO XVIDEO  XIF EXT2	TRGSRC?	0 = FREE 1 = TRG_VIDEO 2 = TRG_EXT 3 = TRG_IF 4 = XVIDEO 5 = XIF 6 = EXT2
2	Video or IF Trigger Level	TRGLVL*	Level	TRGLVL?	Level
3	External Trigger Level	TRGTLLVL*	Voltage	TRGTLLVL?	Voltage
4	Trigger Slope	TRGSLP*	(FALL NEG -)  (RISE POS +)	TRGSLP?	0 = RISE POS + 1 = FALL NEG -
5	Trigger Delay	TRGDLY*	Time	TRGDLY?	Time
6	Trigger Sync ON/OFF	SYNCTRIG*	ON OFF	SYNCTRIG?	0 = OFF 1 = ON
7	Trigger Out CH Select	TRGOUTCH*	CH1 CH2	TRGOUTCH?	0 = CH1 1 = CH2
8	Trigger Out Through	TRGOUTTHR*	ON OFF	TRGOUTTHR?	0 = OFF 1 = ON
9	External2 Trigger Level	TRGTTL2LVL*	Voltage	TRGTTL2LVL?	Voltage

## 6.9.11 Trace

注 トレース名 A、B、C を、下記コマンドの <n> と置き換えることにより、3 種類のトレースを選択することができます。  
<n> = A | B | C

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Trace Mode	---	---	T<n>?	0 = WRITE 1 = VIEW 2 = BLANK
2	Write	<n>W	---	---	---
3	View	<n>V	---	---	---
4	Blank	<n>B	---	---	---
5	Calc Mode Trace A	CALCA*	WRITE MIN MAX  AVG PAVG  MAXCONT	CALCA?	0 = WRITE 1 = MIN HOLD 2 = MAX HOLD 3 = AVERAGE 4 = POWER AVG 5 = MAX HOLD
6	Calc Mode Trace B	CALCB*	WRITE MIN MAX  AVG PAVG  MAXCONT	CALCB?	0 = WRITE 1 = MIN HOLD 2 = MAX HOLD 3 = AVERAGE 4 = POWER AVG 5 = MAX HOLD
7	Calc Mode Trace C	CALCC*	WRITE MIN MAX  AVG PAVG  MAXCONT	CALCC?	0 = WRITE 1 = MIN HOLD 2 = MAX HOLD 3 = AVERAGE 4 = POWER AVG 5 = MAX HOLD
8	Calc Mode : Max Hold	<n>MAX*	ON OFF	<n>MAX?	0 = OFF 1 = ON
9	Calc Mode : Min Hold	<n>MIN*	ON OFF	<n>MIN?	0 = OFF 1 = ON
10	Averaging and Power Averaging Times	<n>G*	整数	<n>G?	整数
11	Averaging and Power Averaging Times Active Trace	SWPCNT*	整数	SWPCNT?	整数

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
12	Average Start	<n>AVG*  <n>GR	ON	<n>AVG?	0 = OFF 1 = ON
13	Average Stop	<n>AVG*  <n>GS	OFF		
14	Average : Pause	<n>GP	---	<n>GP?	0 = Continue 1 = Pause
15	Average : Continue	<n>GC	---		
16	Average : 1 Time	<n>GSGL	---	<n>GSGL?	0 = sliding 1 = once
17	Average : Continuous	<n>GCNT	---		
18	Power Average Start	<n>PAVG*	ON	<n>PAVG?	0 = OFF 1 = ON
19	Power Average Stop	<n>PAVG*	OFF		
20	Power Average : Pause	<n>PGP	---	<n>PGP?	0 = Continue 1 = Pause
21	Power Average : Continue	<n>PGC	---		
22	Power Average : 1 Time	<n>PGSGL	---	<n>PGSGL?	0 = sliding 1 = once
23	Power Average : Continuous	<n>PGCNT	---		
24	Math : Trace Store (Current trace → trace n)	STORE*	TRA TRB TRC  TRXA  TRXB TRXC	---	---
25	Number of Trace Points	TPS TP*	501	TP?	0 = 501 1 = 1001
26		TPL TP*	1001		
27	Detector Mode Trace A	DET*	NRM POS NEG  SMP AVG QP  EMCAV	DET?	0 = NRM 1 = POS 2 = NEG 3 = SMP 4 = AVG 5 = QP 6 = EMCAV
28	Detector Mode Trace B	DETB*	NRM POS NEG  SMP AVG QP  EMCAV	DETB?	0 = NRM 1 = POS 2 = NEG 3 = SMP 4 = AVG 5 = QP 6 = EMCAV

## 6.9.11 Trace

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
29	Detector Mode Trace C	DETC*	NRM POS NEG  SMP AVG QP  EMCAV	DETC?	0 = NRM 1 = POS 2 = NEG 3 = SMP 4 = AVG 5 = QP 6 = EMCAV
30	Detector Mode Auto	DETA<n>*	ON OFF	DETA<n>?	0 = OFF 1 = ON
31	Detector Average Mode	DETAVG*	RMS VIDEO	DETAVG?	0 = VIDEO 1 = RMS
32	Select Active Trace	TRACESEL*	TRA TRB TRC  TRM TRO	TRACESEL?	0 = TRA 1 = TRB 2 = TRC 3 = TRM 4 = TRO
33	Trace A I/O ASCII	TAA*	DDDDD<DLM>× TRP (*1)	TAA?	DDDDD<DLM>×T RP (*1)
34	Trace A I/O Binary	TBA*	2Bytes×TRP	TBA?	2Bytes×TRP
35	Trace B I/O ASCII	TAB*	DDDDD<DLM>× TRP (*1)	TAB?	DDDDD<DLM>×T RP (*1)
36	Trace B I/O Binary	TBB*	2Bytes×TRP	TBB?	2Bytes×TRP
37	Trace C I/O ASCII	TAC*	DDDDD<DLM>× TRP (*1)	TAC?	DDDDD<DLM>×T RP (*1)
38	Trace C I/O Binary	TBC*	2Bytes×TRP	TBC?	2Bytes×TRP
39	Trace Output Format				
40	16bits Integer	FORM1	---	---	
41	16bits Integer	FORM2	---	---	
42	IEEE 32bits Float	FORM3	---	---	
43	IEEE 32bits Float	FORM4	---	---	
44	Trace Data Binary Out Channel1 TraceA Channel2 TraceA Channel1 TraceB Channel2 TraceB Channel1 TraceC Channel2 TraceC	---	---	TBA1? TBA2? TBB1? TBB2? TBC1? TBC2?	FORM コマンドに 従う

(\*1) TRP: トレースポイント数  
<DLM>: デリミタ



## 6.9.12 Trace Math

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Math Trace Write	MTW	---	---	---
2	Math Trace View	MTV	---	---	---
3	Math Trace Blank	MTB	---	---	---
4	Cross Channel Math Trace Write	OW	---	---	---
5	Cross Channel Math Trace View	OV	---	---	---
6	Cross Channel Math Trace Blank	OB	---	---	---
7	TRACE MATH IN CHANNEL Math OFF	TRSUB*	OFF	TRSUB?	0 = OFF
8	Trace Math R/W IN CH Ascii	TAM		TAM?	
9	Trace Math R/W IN CH Bin	TBM		TBM?	
10	Trace Math R/W IN Channel1 BIN	---		TBM1?	
11	Trace Math R/W IN Channel2 BIN	---		TBM2?	
12	TRACE MATH IN CHANNEL A - B A - C A - DL B - A B - C B - DL C - A C - B C - DL	TRSUB*	AMB AMC AMDL BMA BMC BMDL CMA CMB CMDL	TRSUB?	1 = A - B 2 = A - C 3 = A - DL 4 = B - A 5 = B - C 6 = B - DL 7 = C - A 8 = C - B 9 = C - DL
13	TRACE MATH CROSS CHANNEL Math OFF	TRXSUBOFF	---	TRXSUB?	0 = OFF
14	Trace Math R/W CROSS CH Ascii	TAO		TAO?	
15	Trace Math R/W CROSS CH Bin	TBO		TBO?	

## 6.9.12 Trace Math

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
16	Trace Math R/W Channel1 BIN	---		TBO1?	
17	Trace Math R/W Channel2 BIN	---		TBO2?	
18	TRACE MATH CROSS CHANNEL CH1-CH2 A - A A - B A - C A - M B - A B - B B - C B - M C - A C - B C - C C - M M - A M - B M - C M - M	TRXSUB12*	XAMA XAMB XAMC XAMM XBMA XBMB XBMC XBMM XCMA XCMB XCMC XCMM XMMA XMMB XMMC XMMM	TRXSUB?	0 = OFF 1 = XAMA 2 = XAMB 3 = XAMC 4 = XAMM 5 = XBMA 6 = XBMB 7 = XBMC 8 = XBMM 9 = XCMA 10 = XCMB 11 = XCMC 12 = XCMM 13 = XMMA 14 = XMMB 15 = XMMC 16 = XMMM
19	TRACE MATH CROSS CHANNEL CH2-CH1 A - A A - B A - C A - M B - A B - B B - C B - M C - A C - B C - C C - M M - A M - B M - C M - M	TRXSUB21*	XAMA XAMB XAMC XAMM XBMA XBMB XBMC XBMM XCMA XCMB XCMC XCMM XMMA XMMB XMMC XMMM	TRXSUB?	17 = XAMA 18 = XAMB 19 = XAMC 20 = XAMM 21 = XBMA 22 = XBMB 23 = XBMC 24 = XBMM 25 = XCMA 26 = XCMB 27 = XCMC 28 = XCMM 29 = XMMA 30 = XMMB 31 = XMMC 32 = XMMM
20	TRACE MATH CROSS CHANNEL Calc Mode	CALCO*	OFF MIN MAX	CALCO?	0 = OFF 1 = MIN 2 = MAX

## 6.9.13 Pass/Fail

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Pass/Fail Judgement	PFC*	ON OFF	PFC?	0 = OFF 1 = ON
2	Judgment Result	---	---	PFJ? OPF?	0 = Pass 4 = Error 1 = Fail Limit 1 2 = Fail Limit 2 3 = Fail Limit 1&2
3	X Position Mode	LIMPOS*	ABS LFT CENT	LIMPOS?	0 = ABS 1 = CENT (center freq) 2 = LFT (start freq)
4	Y Position Mode	LIMAPOS*	ABS REF DL	LIMAPOS?	0 = ABS 1 = REF 2 = DL
5	X Offset Activate	LIMS*	ON OFF	LIMS?	0 = OFF 1 = ON
6	X Offset Frequency Domain	LIMSF*	Frequency	LIMSF?	Frequency
7	X Offset Time Domain	LIMST*	Time	LIMST?	Time
8	Y Offset Activate	LIMAS*	ON OFF	LIMAS?	0 = OFF 1 = ON
9	Y Offset	LIMASFT*	DB	LIMASFT?	DB
10	Limit Line 1	LMTA*	ON OFF	LMTA?	0 = OFF 1 = ON
11	Limit Line 1 Frequency Domain Data Input	LMTAINF*,*	Frequency, Level	---	---
12	Limit Line 1 Time Domain Data Input	LMTAINT*,*	Time, Level	---	---
13	Limit Line 1 Frequency Domain Data Erase	LMTADELF	---	---	---
14	Limit Line 1 Time Domain Data Erase	LMTADELT	---	---	---
15	Limit Line 2	LMTB*	ON OFF	LMTB?	0 = OFF 1 = ON

## 6.9.14 Display

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
16	Limit Line 2 Frequency Domain Data Input	LMTBINF*,*	Frequency, Level	---	---
17	Limit Line 2 Time Domain Data Input	LMTBINT*,*	Time, Level	---	---
18	Limit Line 2 Frequency Domain Data Erase	LMTBDELF	---	---	---
19	Limit Line 2 Time Domain Data Erase	LMTBDELT	---	---	---
20	Limit Line 1 Pass Range	LARNG*	ABOVE BELOW	LARNG?	0 = ABOVE 1 = BELOW
21	Limit Line 2 Pass Range	LBRNG*	ABOVE BELOW	LBRNG?	0 = ABOVE 1 = BELOW

## 6.9.14 Display

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Display Line Level	DLN* DLN ON[,*] DLN OFF	Level	DLN?	Level
				DLNON?	0 = OFF 1 = ON
2	Reference Line Level	RLN* RLN ON[,*] RLN OFF	Level	RLN?	Level
				RLNON?	0 = OFF 1 = ON
3	Window Center Position	WLX*	Frequency Time	WLX?	Frequency Time
4	Window Width	WDX*	Frequency Time	WDX?	Frequency Time
5	Window Sweep	WDOSWP*	ON OFF	WDOSWP?	0 = OFF 1 = ON
6	Zoom	MLTSCR*	ZM FT TT OFF	MLTSCR?	0 = OFF 1 = ZM 2 = FT 3 = TT

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
7	Zoom Position	ZMPOS*	Frequency Time	ZMPOS?	Frequency Time
8	Zoom Width	ZMWID*	Frequency Time	ZMWID?	Frequency Time
9	Frequency Pos	FTPOS*	Frequency	FTPOS?	Frequency
10	Select Active Context	CTXTSEL*	整数 (0 1)	CTXTSEL? SCRSEL?	0 = CTXT A 1 = CTXT B
11	Display Line IQ	DLNIQ*	Voltage	DLNIQ?	Voltage
12	Display Line Power	DLNPWR*	Level	DLNPWR?	Level
13	Display Line Freq	DLNFRE*	Frequency	DLNFRE?	Frequency
14	Display Line Phase	DLNPHA*	Phase	DLNPHA?	Phase
15	Display Line FFT	DLNFFT*	Level	DLNFFT?	Level
16	Reference Line IQ	RLNIQ*	Voltage	RLNIQ?	Voltage
17	Reference Line Power	RLNPWR*	Level	RLNPWR?	Level
18	Reference Line Freq	RLNFRE*	Frequency	RLNFRE?	Frequency
19	Reference Line Phase	RLNPHA*	Phase	RLNPHA?	Phase
20	Reference Line FFT	RLNFFT*	Level	RLNFFT?	Level

## 6.9.15 Marker

<n> = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10

注 マーカ0は基準マーカです。

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	All Markers Off	MO MKOFF	---	---	---
2	All Markers Off Except Active Marker	MLTOFF	---	---	---
3	Select Active Marker	MKRSEL*	<n>	MKRSEL?	<n>
4	Active Marker ON / OFF	MLN*	ON OFF	MLN?	0 = OFF 1 = ON
5	Marker ON / OFF	MLN<n>*	[ON] OFF	MLN<n>?	0 = OFF 1 = ON
6	Active Marker Frequency	MK*	Frequency Time	MK? MF?	Frequency Time
7	Marker Frequency	MF<n>*	Frequency Time	MF<n>?	Frequency Time
8	Active Marker Level	---	---	ML?	Level
9	Marker Level	---	---	ML<n>?	Level
10	Active Marker Freq + Lev	---	---	MFL?	Frequency Time, Level
11	Marker Freq + Lev	---	---	MFL<n>?	Frequency Time, Level
12	Active Marker Num + Stauts + Freq + Lev	---	---	MFLC?	Marker Number, Status (1=ON 0=OFF), Frequency Time, Level
13	MarkerNum + Stauts + Freq + Lev	---	---	MFLC<n>?	Marker Number, Status (1=ON  0=OFF), Frequency Time, Level
14	Reference Marker Frequency Absolute Value	---	---	MDF2? MFR?	Frequency Time
15	Reference Marker LevelAbsolute Value	---	---	MDL2? MLR?	Level
16	Delta Mode	MKD*	[ON] OFF	---	---

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
17	Reference Object	MKROBJ*	MARK RLIN	MKROBJ?	0 = MARK 1 = RLIN
18	Fixed ΔMarker	FX*	ON OFF	FX?	0 = OFF 1 = ON
19	Inverse ΔMarker	REDLT*	ON OFF	REDLT?	0 = OFF 1 = ON
20	Marker Step Size	MPM*	Frequency Time	MPM?	Frequency Time
21	Marker Step Auto	MPA[*]	[ON] OFF	MPA?	0 = OFF 1 = ON
22	Signal Track	SG[*]	[ON] OFF	SG?	0 = OFF 1 = ON
23	Signal Track Y Range	SGY[ON,]* SGY ON SGY OFF	Level	SGY?	Level
				SGYON?	0 = OFF 1 = ON
24	Active Marker Trace	MKTRACE*	TRA TRB TRC  TRM TRO	MKTRACE?	0 = TRA 1 = TRB 2 = TRC 3 = TRM 4 = TRO
25	Marker Mode	MKMODE*	INDEX VAL	MKMODE?	0 = INDEX 1 = VAL
26	Display Marker List	MKLST*	ON OFF	MKLST?	0 = OFF 1 = ON
27	Get Marker List	---	---	MLSFL?	Num Marker, Active (1=ON 0=OFF), Frequency Time, Level (, ...)
28	Dual Delta Mode	MKDD*	ON OFF	MKDD?	0 = OFF 1 = ON

**6.9.16 Peak and Marker Move**

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	MKR → CF	MC MKCF	---	---	---
2	MKRΔ → CF	MTCF	---	---	---
3	MKR → REF	MR MKRL	---	---	---
4	PEAK → CF	PKCF	---	---	---
5	PEAK → REF	PKRL	---	---	---
6	MKRΔ → SPAN	DS MTSP	---	---	---
7	MKRΔ → ZOOM (DGT)	MTAW	---	---	---
8	MKR → CF Step	M0 MKCS	---	---	---
9	MKRΔ → CF Step	M1 MTCS	---	---	---
10	MKR → MKR Step	M2 MKMKS	---	---	---
11	MKRΔ → MKR Step	M3 MTMKS	---	---	---
12	Mkr to Alternate CF	MKACF	---	---	---
13	Peak to Alternate CF	PKACF	---	---	---
14	Delta to Alternate Zoom	MTASP	---	---	---
15	Delta to Alternate Zoom (DGT)	MTAAW	---	---	---
16	MKR→Analysis Ofs	MKAO	---	---	---
17	MKR→Vertical Pos	MKVP	---	---	---
18	MKRΔ → V Scale/div	MTDIV	---	---	---



## 6.9.17 Peak

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Peak Search	PS	---	---	---
2	Next Peak	NXP	---	---	---
3	Next Peak Left	NXL	---	---	---
4	Next Peak Right	NXR	---	---	---
5	Min Search	MIS	---	---	---
6	Next Min Peak	NXM	---	---	---
7	Min Max Peak	MMS	---	---	---
8	Continuous Peak	CP*	ON OFF	CP?	0 = OFF 1 = ON
9	Peak $\Delta$ Y Div	DY*	Level Div	DY?	Level Div
10	Peak List Frequency Level	PLS FREQ PLS LEVEL	---	---	---
11	X Peak Area Couple to Window	MKSX*	OFF IN OUT	MKSX?	0 = OFF 1 = IN 2 = OUT
12	X Peak Area Position	MKSPOS*	Frequency Time	MKSPOS?	Frequency Time
13	X Peak Area Width	MKSWID*	Frequency Time	MKSWID?	Frequency Time
14	Y Peak Area Couple to Display Line	MKSYDL*	OFF  ABOVE  BELOW	MKSYDL?	2 = OFF 0 = ABOVE 1 = BELOW
15	Y Peak Area Couple to Limit Line 1	MKSYLA*	OFF  ABOVE  BELOW	MKSYLA?	2 = OFF 0 = ABOVE 1 = BELOW
16	Y Peak Area Couple to Limit Line 2	MKSYLB*	OFF  ABOVE  BELOW	MKSYLB?	2 = OFF 0 = ABOVE 1 = BELOW

## 6.9.18 Measurement

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Noise	NI*	Frequency	NI?	Frequency
2	Noise ON + dBm/Hz	NIM	---	---	---
3	Noise ON + dB $\mu$ V/ $\sqrt$ Hz	NIU	---	---	---
4	Noise ON + dBc/Hz	NIC	---	---	---
5	Noise OFF	NIF	---	---	---
6	Noise Mode Query	---	---	NION?	0 = OFF 1 = dBm/Hz 2 = dB $\mu$ V/ $\sqrt$ Hz 3 = dBc/Hz
7	Noise Value	---	---	NIRES?	Level
8	X dB Down Level	MKBW*	DB	MKBW?	DB
9	X dB Down	XDB	---	---	---
10	X dB Down Left	XDL	---	---	---
11	X dB Down Right	XDR	---	---	---
12	X dB Relative X dB Absolute Left X dB Absolute Right	DC0 DC1 DC2	--- --- ---	DC?	0 = Relative 1 = Absolute Left 2 = Absolute Right
13	Continuous dB Down	CDB[*]	[ON] OFF	CDB?	0 = OFF 1 = ON
14	Peak + X dB Down	PSXDB	---	---	---
15	IM Measurement Mode	IMM[*]	[ON] OFF	IMM?	0 = OFF 1 = ON
16	IM Reference Frequency	---	---	IMMREF?	Frequency, Level
17	IM Delta Frequency	---	---	IMMDF?	Delta Frequency
18	IM Distortion Signal Data Readout	---	---	IMMRES?	n<DLM>LL1, LJ1, UL1, UJ1<DLM>... (*1)
19	IM Order Setting	IMODR*	整数 (3 5 7 9)	IMODR?	整数 (3 5 7 9)
20	IM Criteria Input 3rd Order	IMLS3*	DB	IMLS3?	DB
21	IM Criteria Input 5th Order	IMLS5*	DB	IMLS5?	DB

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
22	IM Criteria Input 7th Order	IMLS7*	DB	IMLS7?	DB
23	IM Criteria Input 9th Order	IMLS9*	DB	IMLS9?	DB
24	IM Pass/Fail Judgement	IMPFC*	ON OFF	IMPFC?	0 = OFF 1 = ON
25	IM Save Setup	IMSAVE	---	---	---
26	IM Restore Setup	IMLOAD	---	---	---
27	Harmonics Measurement	HARM[*]	[ON] OFF	HARMON?	0 = OFF 1 = ON
28	Harmonics Results	---	---	HARM?	n<DLM>Freq1, Level1, DeltaLevel1 <DLM>...(*2)
29	Harmonics Max Order	HARMNUM*	整数	HARMNUM?	整数
30	Harmonics Fundamental	HRMFND[ON,]* HRMFND ON HRMFND OFF	Frequency	HRMFND?	Frequency
				HRMFND ON?	0 = OFF 1 = ON

(\*1)

n: 次数に応じた結果セット数  
 LLn: Lower 周波数信号のレベル差  
 LJn: Lower 周波数信号の Pass/Fail 判定結果  
 0: Pass  
 1: Fail  
 -1: Pass/Fail 判定 OFF 時  
 ULn: Upper 周波数信号のレベル差  
 UJn: Upper 周波数信号の Pass/Fail 判定結果  
 <DLM>: デリミタ

(\*2)

n: 結果セット数  
 Freqn: 高調波周波数  
 Leveln: 高調波レベル  
 DeltaLeveln: 基本波と高調波のレベル差  
 <DLM>: デリミタ

## 6.9.18 Measurement

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
31	AM Modulation Measurement	AMMOD*	ON OFF	AMMODON?	0=OFF 1=ON
32	AM Modulation Depth	---	---	AMMOD?	Real %
33.	AM Modulation	---	---	AMMF?	Frequency
34	FM Measurement	FMMEAS*	ON OFF	FMMEASON?	0=OFF 1=ON
35	FM Frequency Deviation	---	---	FMMEAS?	Frequency
36	Modulation Frequency to Sweep time	FMMODF[ON,]* (*3)	Frequency	FMMODF?	Frequency
		FMMODFY* (*4)	Frequency	FMMODFY?	Frequency
		FMMODF*	ON OFF	FMMODFON?	0=OFF1=ON
37	Sound Mode				
	:ON	SON	---	SD?	0 = OFF 1 = ON (AM) 2 = ON (FM)
	:ON (AM)	SAM			
	:ON (FM)	SFM			
	:OFF	SOF			
38	Sound Volume	SDV*	Integer	SDV?	Integer
39	Demodulation Time	PU*	Time	PU?	Time

(\*3) Modulation Frequency to Sweep Time mode を ON し、FM 周波数偏移値を設定する。

(\*4) Modulation Frequency to Sweep Time mode を ON せず、FM 周波数偏移値を設定する。

## 6.9.19 Counter

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Resolution 1 kHz	CN0	---	---	---
2	Resolution 100 Hz	CN1	---	---	---
3	Resolution 10 Hz	CN2	---	---	---
4	Resolution 1 Hz	CN3	---	---	---
5	Resolution Query	---	---	CN?	0 = 1 kHz 1 = 100 Hz 2 = 10 Hz 3 = 1 Hz
6	Counter Position	CNPOS*	Frequency	CNPOS?	Frequency
7	Counter Position Auto (Position linked to Marker)	CNPOSA[*]	[ON] OFF	CNPOSA?	0 = OFF 1 = ON
8	Counter	COUNT*	ON OFF	COUNT?	0 = OFF 1 = ON
9	Counter Value	---	---	CNRES?	Frequency

## 6.9.20 Power

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Measure Mode			PMEASMODE?	0 = OFF 1 = CHPOW 2 = TOTPOW 3 = AVGPOW 4 = OBW 5 = ACP 6 = SEM 7 = SPU
2	Measure OFF	PMEASOFF	---	---	---
3	Measure Averaging	PMEASAVG*	ON OFF	PMEASAVG?	0 = OFF 1 = ON
		PMEASAVGON CE*	ONCE MULT	PMEASAVGON CE?	0 = ONCE 1 = MULT
4	Measure Averaging Times	---	---	PMEASTM?	整数

## 6.9.20 Power

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
5	Measure Trace	PMEASTRACE*	TRA TRB TRC	PMEASTRACE?	0 = TRA 1 = TRB 2 = TRC
6	Channel Power ON/ OFF	PWCHON[*]	[ON] OFF	PWCHON?	0 = OFF 1 = ON
7	Channel Power Average Times	PWCHTM*	整数	PWCHTM?	整数
8	Channel Power	---	---	PWCH?	Level
9	Channel Power Spectral Density	---	---	PWCHPSD?	dB
10	Power Measure Save	PWCHSAVE	---	---	---
11	Power Measure Restore	PWCHLOAD	---	---	---
12	Total Power ON/OFF	PWTOTALON[*] ]	[ON] OFF	PWTOTALON?	0 = OFF 1 = ON
13	Total Power Average Times	PWTOTALTM*	整数	PWTOTALTM?	Integer
14	Total Power	---	---	PWTOTAL?	Level
15	Total Power Spectral Density	---	---	PWTOTALPSD?	dB
16	Power Measure Save	PWTOTSAVE	---	---	---
17	Power Measure Restore	PWTOTLOAD	---	---	---
18	Average Power ON/ OFF	PWAVGON[*]	[ON] OFF	PWAVGON?	0 = OFF 1 = ON
19	Average Power Average Times	PWAVGTM*	整数	PWAVGTM?	整数
20	Average Power Range	PWAVGRANGE *	FULL WIN	PWAVGRANGE ?	0 = FULL 1 = WIN
21	Average Power	---	---	PWAVG?	Level
22	Power Measure Save	PWAVGSAVE	---	---	---
23	Power Measure Restore	PWAVGLOAD	---	---	---
24	OBW Execution	OBWON[*]	[ON] OFF	OBWON?	0 = OFF 1 = ON
25	OBW Measurement Value	---	---	OBW?	Frequency (Fc), Frequency (OBW)
26	OBW %	OBWPER*	Real %	OBWPER?	Real %

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
27	OBW Average Times	OBWTM*	整数	OBWTM?	整数
28	OBW save setup	OBWSAVE	---	---	---
29	OBW restore setup	OBWLOAD	---	---	---
30	ACP Execution	ACP[*]	[ON] OFF	ACPON?	0 = OFF 1 = ON
31	ACP Measurement Value	---	---	ACP?	n<DLM>fIL, l1L, f1H, l1H<DLM>...(*1)
32	ACP Average Times	ACPTM*	整数	ACPTM?	整数
33	ACP Reference Power Value	---	---	ACPREF?	Level
34	ACP Screen	ACPSCR*	FULL CARR	ACPSCR?	0 = FULL 1 = CARR
35	ACP Carrier Bandwidth	CARRBS* ACPCBW*	Frequency	CARRBS? ACPCBW?	Frequency
36	CS/BS Table Input	CSBSIN*,*	Frequency (CS), Frequency (BS)	---	---
37	CS/BS Table Deletion	CSBSDEL	---	---	---
38	ACP Graphics Mode	ADG[*]	[ON] OFF	ADG?	0 = OFF 1 = ON
39	ACP save setup	ACPSAVE	---	---	---
40	ACP restore setup	ACPLOAD	---	---	---
41	ACP Nyquist Filter	ACPNQST*	ON OFF	ACPNQST?	0 = OFF 1 = ON
42	Nyquist Symbol Rate	SYMRT*	Frequency	SYMRT?	Frequency
43	Nyquist Roll Off Factor	RFACT*	Real	RFACT?	Real
44	Spectrum Emission Mask Execution	SEMON[*]	[ON] OFF	SEMON?	0 = OFF 1 = ON
45	SEM Average Times	SEMTM*	整数	SEMTM?	整数
46	SEM Carrier Bandwidth	SEMCBW*	Frequency	SEMCBW?	Frequency
47	SEM Ref Power	SEMRFCALC*	CHN PEAK	SEMRFCALC?	0 = Channel 1 = Peak
48	SEM Nyquist Filter	SEMNQST*	ON OFF	SEMNQST?	0 = OFF 1 = ON
49	SEM save setup	SEMSAVE	---	---	---
50	SEM restore setup	SEMLOAD	---	---	---

6.9.20 Power

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
51	SEM Band Table Input	SEMTIN *,*,*,*,*,*,*	Frequency, (start) Frequency, (stop) Frequency, (ibw) dB, (limit abs start) dB, (limit abs stop) dB, (limit rel start) dB, (limit rel stop) ABS REL  A_AND_R  A_OR_R(judge)	---	---
52	SEM Band Table Delete	SEMTDEL	---	---	---
53	SEM Reference Power	---	---	SEMRFPOW?	dBm
54	SEM Measurement Value	---	---	SEM?	Channel Number, Start Frequency, Stop Frequency, Frequency, Absolute Power, Relative Power, Judge (,...)
55	Spurious Measurement Execution	SPURI*	[ON FREQ] OFF	SPURION?	0 = OFF 1 = ON
56	SPU Measurement Result	---	---	SPURI?	n<DLM> m1<DLM>f1, 11, j1<DLM>... fm1, lm1, jm1<DLM> m2<DLM>f1, 11, j1<DLM> ... fm2, lm2,jm2<DLM>... mn<DLM>f1, 11, j1<DLM> ... fmn, lmn, jmn<DLM> (*2)
57	SPU Table Selection	SPRTBL*	整数 (0 1 2)	SPRTBL?	整数 (0 1 2)
58	SPU Table Input Freq	SPRIN SPRFIN *,*,*,*	Freq, (start) Freq, (stop) AUTO Freq, (rbw) AUTO Freq, (vbw) AUTO Time, (swp) Level, (ref level) AUTO Level, (att) ON OFF, (preamp) Level (Limit)	---	---



No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
59	SPU Freq Table Deletion	SPRFDEL	---	---	---

(\*1)

n: ポイント数 (0 ~ 5)

fnL: n 次周波数 Low

lnL: n 次レベル Low

fnH: n 次周波数 High

lnH: n 次レベル High

&lt;DLM&gt;: デリミタ

(\*2)

n: 測定ポイント数 (0 ~ 15)

m: スプリアス数 (0 ~ 10)

f: スプリアス周波数

l: スプリアス・レベル

j: スプリアス判定結果 (0:Pass, 1:Fail)

&lt;DLM&gt;: デリミタ

## 6.9.21 EMC

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Correction Factor	CR[*]	[ON] OFF	CRON?	0 = OFF 1 = ON
2	Correction Factor Table Input	CRIN*,*	Frequency, Level(DB)	---	---
3	Correction Factor Table Deletion	CRDEL	---	---	---
4	EMC Trace Detection OFF ON	EMCON* EMC-DET*	NRM OFF PEAK ON	EMCON? EMCDET?	0 = OFF 3 = ON
5	EMC BW Auto 200Hz 9kHz 120kHz 1MHz	QPAUTO QA QP0 QP1 QP2 QP3	---	QPAUTO? QA?	0 = Auto 1 = 200Hz 2 = 9kHz 3 = 120kHz 4 = 1MHz
6	Measure	EMCMEAS[*]	[ON] OFF	EMCMEAS?	0 = OFF 1 = ON
7	Measure Time	EMCMEASTIM* EMCMEASTIM*	[ON] OFF Time	EMCMEASTI-MON? EMCMEAS-TIM?	0 = OFF 1 = ON Time
8	Horizontal Scale	HSCALE*	LIN LOG	HSCALE?	0 = LOG 1 = LIN
9	Limit Line Margin	LMTMRGN* LMTMRGN*	ON OFF dB	LMTMRGNON? LMTMRGN?	0 = OFF 1 = ON dB

## 6.9.22 Calibration

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	U3841/51 Calibration All (CH1) Calibration All (CH2)	CLALL*	CLCH1 CLCH2	---	---
2	U3872 Calibration All (CH1-L) Calibration All (CH1-H) Calibration All (CH2-L) Calibration All (CH2-H)	CLALL*	CLCH1L CLCH1H CLCH2L CLCH2H	---	---
3	Total Gain Calibration Status	---	---	CLGAIN [RFC1]]RFC2? (*1)	整数 0 = OK >0 = Warning <0 = Error
4	RBW Calibration Status	---	---	CLRBW [RFC1]]RFC2? (*1)	整数 0 = OK >0 = Warning <0 = Error
5	PBW Calibration Status	---	---	CLPBW [RFC1]]RFC2? (*1)	整数 0 = OK >0 = Warning <0 = Error
6	ATT step Calibration Status	---	---	CLATT [RFC1]]RFC2? (*1)	整数 0 = OK >0 = Warning <0 = Error
7	CAL 10 M Reference Adjust	CLCREF*	整数	CLCREF?	整数
8	CAL 10 M Reference Default	CLDREF	---	---	---
9	CAL 10 M Reference Store	CLSREF	---	---	---
10	F-Correction	FC*	ON OFF	FC?	0 = OFF 1 = ON
11	CAL-Correction	CC*	ON OFF	CC?	0 = OFF 1 = ON

(\*1) RFC1 = Lo input, RFC2 = Hi input

## 6.9.23 Save/Recall

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Save (File or File Number)	SV[*]	:String  整数	---	---
2	Delete (File)	DEL*	:String	---	---
3	Recall (File)	RC*	:String	---	---
4	Rename (File, New Name)	RENAME*,*	:String, :String	---	---
5	Write Protect (File)	WP*,*	:String, ON OFF	---	---
6	File Format	FILEFORMAT*	BIN CSV XML	FILEFORMAT?	0 = BIN 1 = CSV 2 = XML
7	Media	FILEMEDIA*	FLASH USB	FILEMEDIA?	0 = FLASH 1 = USB
8	Auto Save Execute	ASV*	ON OFF	ASV?	0 = OFF 1 = ON
9	Auto Save Mode	ASVMODE*	LMT SWP TIME	ASVMODE?	0 = Limit 1 = Sweep 2 = Time
10	Auto Save Max Save Count	ASVMAXSV*	整数	ASVMAXSV?	整数
11	Auto Save Target CH	ASVTGTCH*	CH1 CH2	ASVTGTCH?	0:CH1 1:CH2
12	Auto Save Sweep Step	ASVSWPSTEP*	整数	ASVSWPSTEP?	整数
13	Auto Save Interval Time	ASVINTVTM*	時間	ASVINTVTM?	時間
14	Auto Save Total Time	ASVTOTALTM*	時間	ASVTOTALTM?	時間
15	Auto Save Limit Save On	ASVLMTJDG*	PASS FAIL	ASVLMTJDG?	0:PASS 1:FAIL
16	Quick Save	QSV*	0 to 9	---	---
17	Quick Recall	QRC*	0 to 9	---	---
18	Save Sampling Data	IQBSV*	:String	---	---
19	Recall Items Setup	RCSET*	ON OFF	RCSET?	0 = OFF 1 = ON
20	Recall Items Trace	RCTRC*	ON OFF	RCTRC?	0 = OFF 1 = ON

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
21	Recall Items Table	RCTBL*	ON OFF	RCTBL?	0 = OFF 1 = ON
22	Recall Items Vector Correction	RCV CORR*	ON OFF	RCV CORR?	0 = OFF 1 = ON
23	Table Data Save (Type,:Filename)	TBL SV *,:*	TLIM TCHN TACP  TSEM TSPU TANT, :String	---	---
24	Table Data Recall	TBLRC*	:String	---	---

\* ファイル名/フォルダ名を指定する場合には、/adv のフォルダにあるフォルダから指定をして下さい (/adv のパス指定は、不要です)。

(例 1) adv/dat フォルダに abc001.dat ファイルの Save/Recall

- Save  
SV:dat/abc001.dat
- Recall  
RC:dat/abc001.dat

(例 2) adv/dat フォルダの abc001.dat ファイルを Delete

DEL:dat/abc001.dat

(例 3) adv/dat フォルダの abc001.dat ファイルを Rename

RENAME:dat/abc001.dat,:dat/abc002.dat

(例 4) adv/dat フォルダの abc001.dat ファイルに Write Protect ON/OFF

- Write Protect ON  
WP:dat/abc001.dat,ON
- Write Protect OFF  
WP:dat/abc001.dat,OFF

## 6.9.24 File Management

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Reading Bitmap File	---	---	BMP?	Binary data <EOI>
2	Reading Portable Network Graphics File	---	---	PNG?	Binary data <EOI>
3	Reading Image File	---	---	GIMAG :String	Binary data <EOI>
4	Reading Data File	---	---	GDATA :String	Binary data <EOI>

**6.9.25 Vertical Cursor**

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	V-Cursor ON/OFF	VCURS*	ON OFF	VCURS?	0 = OFF 1 = ON
2	V-Cursor Mode	VCMODE*	CSGL CDUAL	VCMODE?	0 = Single 1 = Dual
3	V-Cursor Position	VCSETA* VCSETB*	Frequency, 時間	VCSETA? VCSETB?	Frequency, 時間
4	V-Cursor Result	---	---	VCRES?	CH1 Hor, Ver, CH2 Hor, Ver, Dlt Hor, Ver

**6.9.26 Time Domain Analysis**

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Time Domain Analysis	TDANA*	ON/OFF	TDANAON?	0:OFF, 1:ON
2	Analysis Offset	TDAOFS*	Time	TDAOFS?	Time
3	Analysis Window	TDATIM*	Time	TDATIM?	Time
4	Vertical Scale IQ	VSCLIQ*	Voltage	VSCLIQ?	Voltage
5	Vertical Scale Power	VSCLPWR*	Power	VSCLPWR?	Power
6	Vertical Scale Freq	VSCLFRE*	Frequency	VSCLFRE?	Frequency
7	Vertical Scale Phase	VSCLPHA*	Phase	VSCLPHA?	Phase
8	Vertical Scale FFT	VSCLFFT*	Level	VSCLFFT?	Level
9	Vertical Position IQ	VPOSIQ*	Position	VPOSIQ?	Position
10	Vertical Position Power	VPOSPWR*	Position	VPOSPWR?	Position
11	Vertical Position Freq	VPOSFRE*	Position	VPOSFRE?	Position
12	Vertical Position Phase	VPOSPHA*	Position	VPOSPHA?	Position
13	Vertical Position FFT	VPOSFFT*	Position	VPOSFFT?	Position
14	Measurement Function	TMEAS*	FRE/PHA/PWR/IQ/ FFT	TMEAS?	0:FRE,1:PHA, 2:PWR,3:IQ,4:FFT

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
15	Capture Time	CAPTIM*	Time	CAPTIM?	Time
16	Capture BW	CAPBW*	Frequency	CAPBW?	Frequency
17	RBW	TDRB*	Frequency	TDRB?	Frequency
18	RBW Auto	TDBA[*]	[ON] OFF	TDBA?	0 = OFF 1 = ON
19	Display Mode	TDDISP*	NORM POSV  DUAL	TDDISP?	0=Normal, 1=Position View, 2=Dual Measure
20	Cross point Search Right	CRSPR	---	---	---
21	Cross point Search Left	CRSPL	---	---	---
22	Slope	CRSSLP*	UP DN ALL	CRSSLP?	0=Up,1=Down, 2=All
23	Threshold	THRIQ* THRPWR* THRFRE* THRPHA* THRFFT* THRESH*	Volt dBm Frequency Phase dBm ON OFF	THRIQ? THRPWR? THRFRE? THRPHA? THRFFT? THRESHON?	Volt dBm Frequency Phase dBm 0=OFF,1=ON
24	Measures	DMEAS*	OFF/DTYP/ DTYN FREQ/ WIDP/WIDN/ HILO	DMEAS?	0:OFF, 1:Pos Duty, 2:Neg Duty, 3:Freq, 4:Pos Wid, 5:Neg Wid, 6: HILO
25	Measures Result	---	---	DMEASRES? SWINGRES?	Real H,L,S
26	Mkr to Analysis Offs	MKAO	---	---	---
27	Delta Mkr to Analysis Win	MTAW	---	---	---
28	Mkr to Vertical Pos	MKVP	---	---	---
29	Delta Mkr to scale/ div	MTDIV	---	---	---
30	IQ Output Binary	---	---	IQB?	8 bytes x IQP (*1)
31	IQ Output Scale	---	---	IQS?	Real
32	IQ Output Points	---	---	IQP?	Points
33	IQ Sample Rate	---	---	IQR?	Frequency

## 6.9.27 Config

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
34	ALL IQ Sample Points	---	---	IQPAL?	Points
35	Clear IQP value	IQPRST	---	---	---
36	Re-Calculation ON OFF	RECALCON*	ON OFF	RECALCON?	0 = OFF 1 = ON

(\*1) IQP : IQ Pair Sample Point 数

## 6.9.27 Config

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Title	LON*	/*String*/	LB?	String
2	Erase Title	LOF	---	---	---
3	10 MHz Internal Reference Signal Source	RFI	---	FREF?	0 = INT 1 = EXT 2 = XTL
4	10 MHz External Reference Signal Source	RFE			
5	Xtal (Option)	RFX			
6	Reference Signal Source	RF*	Frequency	RF?	Frequency
7	Input RF Connector	RFC*	RFC1 RFC2	RFC?	1 = RFC1(Lo input) 2 = RFC2(Hi input)
8	Screen Copy	HCOPY	---	---	---

## 6.9.28 Preset

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Preset	IP *RST	---	---	---
2	Factory Init	SUPIP	---	---	---



## 6.9.29 GPIB

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Status Byte Clear	*CLS	---	---	---
2	STB Read	---	---	*STB?	整数
3	SRE Read/Write	*SRE*	整数	*SRE?	整数
4	ESR Read	---	---	*ESR?	整数
5	ESE Read/Write	*ESE*	整数	*ESE?	整数
6	OSR Read	---	---	OPREVT?	整数
7	OSER Read	OPR*	整数	OPR?	整数
8	SRQ Interrupt ON	S0	---	---	---
9	SRQ Interrupt OFF	S1	---	---	---
10	SRQ status clear	S2	---	---	---
11	Delimiter CR LF EOI	DLIM0	---	---	---
12	Delimiter LF	DLIM1	---	---	---
13	Delimiter EOI	DLIM2	---	---	---
14	Delimiter ;	DLIM5	---	---	---
15	Local Lockout	LLO	---	---	---
16	Remote Control	REN	---	---	---
17	Local Control	GTL	---	---	---
18	Refresh Screen in Remote Control	SCR[*]	[ON] OFF	SCR?	ON OFF
19	Open Menus in Remote Control	MNRF[*]	[ON] OFF	MNRF?	ON OFF
20	Annotations	ANNOT[*]	[ON] OFF	ANNOT?	ON OFF
21	Access Channel CH1	X1	---	---	---
22	Access Channel CH2	X2	---	---	---

## 6.9.30 Others

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	Device ID Output	---	---	*IDN? ID?	Maker Name, Device Name, Serial No., Revision
2	Date Setting	SETDATE*	Date (YYMMDD)	SETDATE?	Date
3	Time Setting	SETTIME*	Time (HHMMSS)	SETTIME?	Time
4	Option List	---	---	*OPT?	Opt1,Opt2,..., Optn<DLM>
5	Option				
6	: High Stability	---	---	OPT20?	
7	: EMC Filter	---	---	OPT28?	
8	: TG -60 dBm	---	---	OPT76?	
9	: TG High Freq.	---	---	OPT77?	
10	Execute Self Test CH1	---	---	*TST?	Supply_Voltage, judge<DLM> Memory, judge<DLM> CPU_Registers, judge<DLM> RF_Registers, judge<DLM> RF_PLL_Lock, judge<DLM> LO_Registers, judge<DLM> LO_PLL_Lock, judge<DLM> AIF_Registers, judge<DLM> AIF_PLL_Lock, judge<DLM> TG_Registers, judge<DLM>* Temperature, judge<DLM> RBD_PLL_Lock, judge<DLM> LD_PLL_Lock, judge<DLM>

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
11	Execute Self Test CH2	---	---	*TST?	Supply_Voltage, judge<DLM> Memory, judge<DLM> CPU_Registers,judge<DLM> RF_Registers, judge<DLM> RF_PLL_Lock, judge<DLM> LO_Registers, judge<DLM> LO_PLL_Lock, judge<DLM> AIF_Registers, judge<DLM> AIF_PLL_Lock, judge<DLM>
12	Error Number	---	---	ERRNO?	0 = No Error ≠ 0 = Error Code

\* TG Option

**6.9.31 TG**

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Argument Format	Code	Output Format
1	TG ON	TG	---	TG?	0 = OFF 1 = ON
2	TG OFF	TGF	---	---	---
3	TG Level	TGL*	Level	TGL?	Level
4	TG Frequency Offset	TGO[ON,]* TGO ON OFF	Frequency	TGO? TGOON?	Frequency 0 = OFF 1 = ON
5	TG Level Offset	TGLO[ON,]* TGLO ON OFF	DB	TGLO? TGLOON?	DB 0 = OFF 1 = ON
6	Capture Normalize Data Active Trace	CAPND	---	---	---
7	Normalize Correction Active Trace	NORM*	ON OFF	NORM?	0 = OFF 1 = ON
8	Capture Normalize Data Trace (n = A B C)	<n>RX	---	---	---
9	Normalize Correction Trace (n = A B C)	<n>NORM*	ON OFF	<n>NORM?	0 = OFF 1 = ON
10	Normalize Execute Active Trace	AR	---	---	---
11	Delta Reference Line	TGDLTRLN*	ON OFF	TGDLTRLN?	0 = OFF 1 = ON
12	VSWR ON/OFF	VSWRON [*]	[ON] OFF	VSWRON?	0 = OFF 1 = ON
13	VSWR Measurement Value	---	---	VSWR?	Real
14	VSWR Return Loss Value	---	---	VSWRLOSS?	DB (Real)

**6.9.32 Units**

Function	Code
Exponent	E (see IEEE Std 488.2-1992 page 89)
GHz	GZ
MHz	MZ
KHz	KZ
Hz	HZ
DB	DB
DBM	DBM
DBMV	DBMV
DBUV	DBUV
DBEMF	DBEMF
DBPW	DBPW
Watt	W   WATT
mW	MW
Volt	V   VOLT
Millivolt	MV
Microvolt	UV
Nanovolt	NV
Second	SC
Millisecond	MS   MSEC
Microsecond	US   USEC
Nanosecond	NSEC
%	%   PER
ppm	PPM

## 6.10 I/Q データ出力

### 6.10.1 IQB データ出力

IQB? による出力データは Binary (32bit) 形式で I, Q 交互に出力されます。

表 6-9 取得サンプル数 N 個 (0 ~ N-1) の IQ-Pair データ出力 (8xN bytes)

Sample number	0		1		...		N-2		N-1	
IQB? Data	I0	Q0	I1	Q1	...	...	IN-2	QN-2	IN-1	QN-1
Size Bytes	4	4	4	4	4	4	4	4	4	4

IQB? による出力は I, Q の相対値を表し、位相角の計算に使用できます。

IQS? による出力は I, Q Pair の絶対値換算用スケーリング・データを表します。

IQPAL? による出力は、キャプチャした全 I, Q Pair の数を表します。

IQP? による出力は、次に呼び出す IQB? が返す I, Q Pair のサンプル数 (N) を表します。

IQR? による出力は I, Q Pair のサンプリング周波数を表します。

IQPRST は IQP? の値を初期値に戻します。

I, Q 出力の N 番目データ (IN, QN) に対応する電圧 (VIN, VQN) は次式で求められます。

$$VIN=IN \times IQS [V]$$

$$VQN=QN \times IQS [V]$$

例： IQPAL? の返り値が 4.5M サンプルの場合の全 IQ データを取得します。

1. 1 度目の IQP? では 2Mi を返すため、IQB? で 2M サンプルのデータを取得します。
2. 2 度目の IQP? も 2Mi を返すため、IQB? で 2M サンプルのデータを取得します。
3. 3 度目の IQP? は 0.5Mi を返すため、IQB? で 0.5M サンプルのデータを取得します。
4. 4 度目の IQP? は 0 を返すため、データ取得は終了します。

IQP? が 0 を返すまで IQB? を繰り返し、全データを取得します。

TS または IQPRST マンドで 1 度目の状態に戻ります。

## 6.10.2 IQBSV データ出力

IQBSV? による出力データは Binary 形式でヘッダ・データ部 (24byte)、I, Q データ (IQB? 時に出力されるデータ) が出力されます。

表 6-10 IQBSV データ出力フォーマット

「6.10.1 IQBデータ 出力」を参照	ヘッダ・データ・サイズ	各4byte : 計24byte
	IQR	
	IQP	
	IQS	
	Capture BW	
	Center Frequency	
	I0	
	Q0	
	I1	
	Q1	
	:	
	IN-2	
	QN-2	
	IN-1	
QN-1		

## 6.11 RECALCON による時間軸解析の再計算動作設定

時間軸解析機能およびチャンネル間のベクトル演算解析機能では、RF 信号の記録停止状態で解析機能の切り替えや表示範囲の変更等を行ったときに、本体に記録された測定データを再計算／再表示しますが、リモート・コントロール動作時ではこの再計算機能のために設定動作が遅くなる場合があります。

このような場合、"RECALCON" コマンドを使って再計算動作を禁止することで、設定動作速度が改善されます。

### 用法

RECALCON ON: 再計算動作を行います。(デフォルト設定)

RECALCON OFF: 再計算動作を禁止します。

"RECALCON" コマンドによる再計算動作の設定は、リモート動作中のみ有効です。LOCAL キー操作や GTL メッセージによりリモート・コントロールが解除された場合は、再計算動作を行う状態に設定されます。

## 6.12 UNCAL メッセージ、エラー・メッセージ一覧、制限事項

### 6.12.1 Partial FFT 解析の UNCAL メッセージを消去する方法

Partial FFT 解析は Capture BW 設定値 (表示周波数スパンに等しい) に応じた時間長のデータを使用します。FFT 解析に必要なデータ長が不足する場合、UNCAL メッセージが表示され表示値が正しくない可能性があることを示します。

UNCAL メッセージは、第 5 章、Capture BW 設定値と Partial FFT 周波数分解能の表に示す FFT 解析データ長と Capture Time、Analysis Offset の各設定値の関係が以下の式を満足しない場合に表示されます。

FFT 解析データ長 . (Capture Time - Analysis Offset)

UNCAL メッセージを消去するには、下記 1, 2 のいずれか、または両方を行ってください。

1. Capture Time 設定値を大きくする。
2. Analysis Offset 設定値を小さくする。



### 6.13 外部信号源による複数ポイント補正

外部コントローラから外部信号源と本器をコントロールし、複数ポイントの補正 (Specific Span, InBand) を実現します。

この方式は「5.2.13.1 Power Ratio Phase Diff」、「5.2.13.2 Differential」、「5.2.13.4 Math」で適用されます。

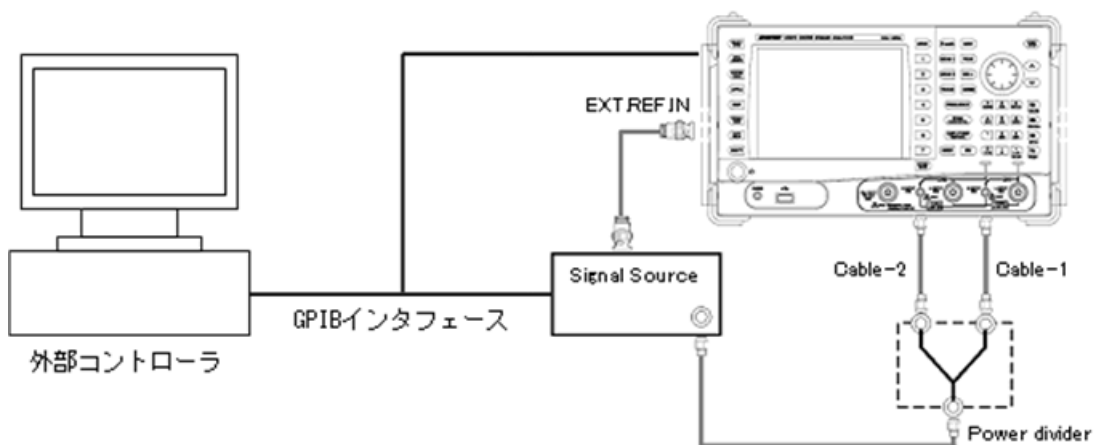


図 6-14 外部コントローラと外部信号源による補正接続 (GPIB インタフェース)

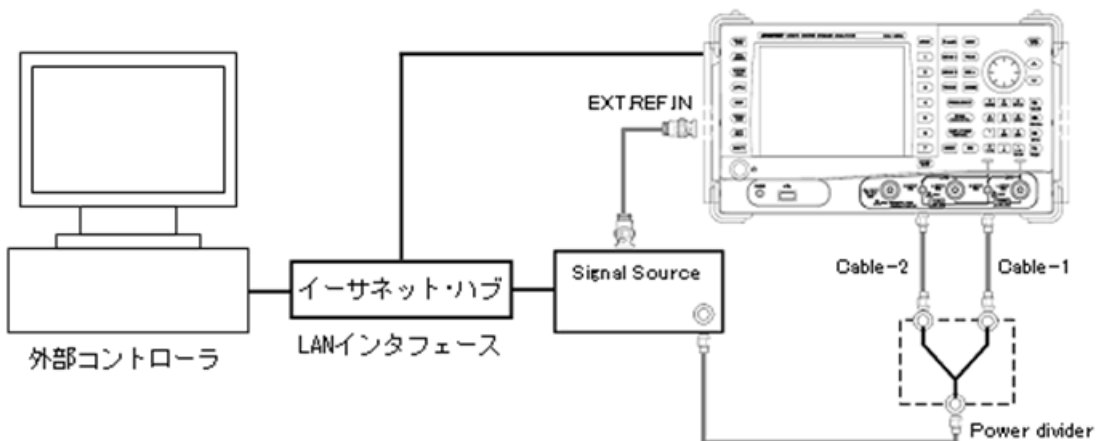


図 6-15 外部コントローラと外部信号源による補正接続 (LAN インタフェース)

周波数の基準信号源 (EXT.REF.IN) は、**SYSTEM**, **Config**, **Frequency Reference**, **Frequency Reference INT|EXT|XTL(EXT)** と押します。周波数の基準信号源が EXT に設定されます。

LAN インタフェースを使用する場合には、イーサネット・ハブなど複数の LAN インタフェースをもった機器を接続するための外部機器を介して接続します。機器間の接続にはストレート・ケーブルを使用します。詳細は「6.3 LAN リモート・コントロール・システム」を参照してください。

### 6.13.1 Specific Span 用リモートコマンド

No.	Function	Command (EXE, SET)		Query (GET)	
		Code	Arguments	Code	Output
1	Vector Correction Specific Span Initialize	VCFSEXTINIT	---	---	---
2	Vector Correction Specific Span Get Step Size	---	---	VCFSEXTSTPSZ?	整数
3	Vector Correction Specific Span Set Index	VCFSEXTIDX*	整数	---	---
4	Vector Correction Specific Span Get Correction Data	VCFSEXTEXE	---	---	---
5	Vector Correction Specific Span Finish	VCFSEXTFINISH	---	---	---
6	Vector Correction Specific Span Abort	VCFSEXTABORT	---	---	---

**重要** 周波数は以下の範囲内で指定してください。

U3841: 1MHz ~ 3GHz

U3851: 1MHz ~ 8GHz

U3872: 1MHz ~ 8GHz(L-Input)

U3872: 10MHz ~ 43GHz(H-Input)

$(\text{Stop Freq} - \text{Start Freq})/1000 \leq \text{Step Freq} \leq 50\text{MHz}$

となるような範囲で Step Freq を指定してください（全機種共通）。

また周波数 (Start Freq, Stop Freq, Step Freq) を設定するリモート・コマンドは以下のようになります。

Function	Command (EXE, SET)		Query (GET)	
	Code	Arguments	Code	Output
Vector Correction Start Frequency	VCFA*	Frequency	VCFA?	Frequency
Vector Correction Stop Frequency	VCFB*	Frequency	VCFB?	Frequency
Vector Correction Step Frequency	VCFSTEP*	Frequency	VCFSTEP?	Frequency

## 6.13.2 InBand 用リモートコマンド

No.	Function	Command (EXE,SET)		Query (GET)	
		Code	Arguments	Code	Output
1	Vector Correction InBand Initialize	VCINBEXTINIT	---	---	---
2	Vector Correction InBand Get Step Size	---	---	VCINBEXTSTPSZ?	整数
3	Vector Correction InBand Set Index	VCINBEXTIDX*	整数	---	---
4	Vector Correction InBand Get Frequency Offset	---	---	VCINBEXTFO?	周波数
5	Vector Correction InBand Get Correction Data	VCINBEXTEXE	---	---	---
6	Vector Correction InBand Finish	VCINBEXTFINISH	---	---	---
7	Vector Correction InBand Abort	VCINBEXTABORT	---	---	---

外部コントローラから外部信号源と本器をコントロールし、複数ポイントの補正を実行する場合のプログラム例は「6.14.2 外部信号源で複数ポイントを補正するプログラム例」を参照してください。

## 6.14 リモート・コントロール プログラム例

この章では、リモート・コントロール用のプログラム例について説明します。

本章のプログラム例では、Microsoft 社製 Visual Basic 6.0 言語 (VB) を使用していますので、他の言語でプログラムする際は、その言語にあった記述に変更してください。

また、GPIB バス・コントローラとして National Instruments 社 (以降 NI 社) 製の GPIB ボードを想定しプログラムの説明を行います。

### 6.14.1 GPIB バス・コントロール用基本ステップ

ここでは GPIB バスを Visual Basic 6.0 / Microsoft Visual Basic 2008 からコントロールするために必要な作業を、順を追って説明します。

Visual Basic 6.0 / Microsoft Visual Basic 2008 に依存した変数等の初期化や、関数ルーチンの定義等は、Visual Basic 6.0 / Microsoft Visual Basic 2008 プログラムの表記ルールに従ってください。

#### 6.14.1.1 GPIB コントロール・ライブラリの読み込み

Visual Basic 6.0 / Microsoft Visual Basic 2008 言語で記述したプログラムから、NI 社製 GPIB ボードをコントロールするためには、NI 社で提供している Visual Basic 6.0 / Microsoft Visual Basic 2008 用 GPIB 通信インタフェースを記述した VBIB-32.BAS ファイルとエラーやタイム・アウト等を定義した NIGLOBAL.BAS ファイルや DLL ファイルを Visual Basic 6.0 / Microsoft Visual Basic 2008 の Project に組み入れる必要があります。

### 6.14.1.2 プログラム例

例 1 本器をマスタ・リセットしたあと、中心周波数を設定する

Visual Basic 6.0

```

*****
'* Example 1 Setting the center frequency after resetting this instrument. *
*****
Private Sub CmdSetup_Click()
    Dim u38 As Integer

    Call ibdev(0, 8, 0, T30s, 1, 0, u38)      ' Initialize
    Call ibclr(u38)                          ' Performs a Device Clear.
    Call ibwrt(u38, "IP")                    ' preset
    Call ibwrt(u38, "CF 30MZ")              ' Sets the center frequency to 30 MHz.
    Call ibonl(u38, 0)
End Sub

```

Microsoft Visual Basic 2008

```

Imports NationalInstruments.NI4882

Module MdlManualSample_01
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    *****
    '* Example 1 Setting the center frequency after resetting this instrument. *
    *****
    Sub Main()
        Dim str As String = Space(1)
        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)

        Call GpibDevice.Clear()                ' Performs a Device Clear.
        System.Console.WriteLine("preset.")
        Call GpibDevice.Write("IP")            ' preset.
        System.Console.WriteLine("Sets the center frequency to 30 MHz.")
        Call GpibDevice.Write("CF 30MZ")      ' Sets the center frequency to 30 MHz.

        Call GpibDevice.Dispose()

        System.Console.WriteLine("Please push Enter key.")
        str = System.Console.ReadLine
    End Sub
End Module

```

## 6.14.1 GPIB バス・コントロール用基本ステップ

例 2 スタート周波数を 300kHz、ストップ周波数を 800kHz に設定し、周波数オフセットを 50kHz 加える

```

*****
'* Example 2 Setting the start frequency to 300 kHz,      *
'*           setting the stop frequency to 800 kHz and *
'*           adding 50 kHz to the frequency offset. *
*****
Private Sub CmdSetup_Click()
    Dim u38 As Integer

    Call ibdev(0, 8, 0, T30s, 1, 0, u38)    ' Initialize
    Call ibclr(u38)                          ' Performs a Device Clear.
    Call ibwrt(u38, "FA 300KZ")             ' Sets the start frequency to 300 kHz.
    Call ibwrt(u38, "FB 800KZ")             ' Sets the stop frequency to 800 kHz.
    Call ibwrt(u38, "FO 50KZ")              ' Adds 50 kHz to the frequency offset.
    Call ibonl(u38, 0)
End Sub

```

## Microsoft Visual Basic 2008

```

Imports NationalInstruments.NI4882

Module MdlManualSample_02
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0

    *****
    *****
    '* Example 2 Setting the start frequency to 300 kHz, setting the stop frequency to 800
    kHz and adding 50 kHz to the frequency offset. *
    *****
    *****

    Sub Main()
        Dim str As String = Space(1)

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear()                ' Performs a Device Clear.
        System.Console.WriteLine("Sets the start frequency to 300 kHz.")
        Call GpibDevice.Write("FA 300KZ")     ' Sets the start frequency to 300 kHz.
        System.Console.WriteLine("Sets the stop frequency to 800 kHz.")
        Call GpibDevice.Write("FB 800KZ")     ' Sets the stop frequency to 800 kHz.
        System.Console.WriteLine("Adds 50 kHz to the frequency offset.")
        Call GpibDevice.Write("FO 50KZ")     ' Adds 50 kHz to the frequency offset.

        Call GpibDevice.Dispose()

        System.Console.WriteLine("Please push Enter key.")
        str = System.Console.ReadLine
    End Sub

End Module

```

例3 リファレンス・レベルを 87dB $\mu$ V、5dB/div、RBW を 100kHz にする

## Visual Basic 6.0

```

*****
'* Example 3 Setting the reference level to 87 dB $\mu$ V (in 5 dB/div) and the RBW to 100 kHz *
*****
Private Sub CmdSetup_Click()
    Dim u38 As Integer

    Call ibdev(0, 8, 0, T30s, 1, 0, u38)      ' Initialize
    Call ibclr(u38)                          ' Performs a Device Clear.
    Call ibwrt(u38, "AUNITS DBUV")           ' Sets the level unit to dB $\mu$ V.
    Call ibwrt(u38, "RL 87DB")              ' Sets the reference level to 87 dB ( $\mu$ V).
    Call ibwrt(u38, "DD 5DB")               ' Sets the vertical scale to 5 dB/div.
    Call ibwrt(u38, "RB 100KZ")            ' Sets the RBW to 100 kHz.
    Call ibonl(u38, 0)
End Sub

```

## Microsoft Visual Basic 2008

```

Imports NationalInstruments.NI4882

Module MdlManualSample_03
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0

    *****
    *
    '* Example 3 Setting the reference level to 87 dB $\mu$ V (in 5 dB/div) and the RBW to 100 kHz. *
    *****
    *

    Sub Main()
        Dim str As String = Space(1)

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear()                ' Performs a Device Clear.
        System.Console.WriteLine("Sets the level unit to dB $\mu$ V.")
        Call GpibDevice.Write("AUNITS DBUV")   ' Sets the level unit to dB $\mu$ V.
        System.Console.WriteLine("Sets the reference level to 87 dB ( $\mu$ V).")
        Call GpibDevice.Write("RL 87DB")      ' Sets the reference level to 87 dB ( $\mu$ V).
        System.Console.WriteLine("Sets the vertical scale to 5 dB/div.")
        Call GpibDevice.Write("DD 5DB")      ' Sets the vertical scale to 5 dB/div.
        System.Console.WriteLine("Sets the RBW to 100 kHz.")
        GpibDevice.Write("RB 100KZ")         ' Sets the RBW to 100 kHz.

        Call GpibDevice.Dispose()

        System.Console.WriteLine("Please push Enter key.")
        str = System.Console.ReadLine
    End Sub

End Module

```

## 例 4 変数を用いた設定の例

## Visual Basic 6.0

```
*****  
 '* Example 4 Setting the instrument using variables *  
*****  
Private Sub CmdSetup_Click()  
    Dim u38 As Integer  
    Dim A As String  
    Dim B As String  
    Dim C As String  
  
    Call ibdev(0, 8, 0, T30s, 1, 0, u38)      ' Initialize  
    A = "10"                                  ' Sets the character string.  
    B = "2"  
    C = "20"  
    Call ibclr(u38)                           ' Performs a Device Clear.  
    Call ibwrt(u38, "CF " & A & "MZ")        ' Sets the center frequency to A MHz.  
    Call ibwrt(u38, "SP " & B & "MZ")        ' Sets the span to B MHz.  
    Call ibwrt(u38, "AT " & C & "DB")        ' Sets the ATT to C dB.  
    Call ibonl(u38, 0)  
End Sub
```



## Microsoft Visual Basic 2008

```
Imports NationalInstruments.NI4882

Module MdlManualSample_04
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    '*****
    '* Example 4 Setting the instrument using variables. *
    '*****
    Sub Main()
        Dim A As String
        Dim B As String
        Dim C As String
        Dim str As String = Space(1)

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        A = "10"      ' Sets the character string.
        B = "2"
        C = "20"
        Call GpibDevice.Clear()           ' Performs a Device Clear.
        System.Console.WriteLine("Sets the center frequency to A MHz.")
        Call GpibDevice.Write("CF " & A & "MZ") ' Sets the center frequency to A MHz.
        System.Console.WriteLine("Sets the span to B MHz.")
        Call GpibDevice.Write("SP " & B & "MZ") ' Sets the span to B MHz.
        System.Console.WriteLine("Sets the ATT to C dB.")
        Call GpibDevice.Write("AT " & C & "DB") ' Sets the ATT to C dB.

        Call GpibDevice.Dispose()

        System.Console.WriteLine("Please push Enter key.")
        str = System.Console.ReadLine
    End Sub
End Module
```

## 例 5 レジスタ 5 への設定値のセーブおよびリコール

## Visual Basic 6.0

```
*****  
'* Example 5 Saving set values in Register 5 and recalling them from Register 5 *  
*****  
Private Sub CmdSetup_Click()  
    Dim u38 As Integer  
    Dim LabelBuff As String          ' Defines the character string buffer for the  
  
    Call ibdev(0, 8, 0, T30s, 1, 0, u38) ' Initialize  
    ' label  
    LabelBuff = "/*Cross Domain Analyzer*/" ' Sets the label.  
    Call ibclr(u38)                    ' Performs a Device Clear.  
    Call ibwrt(u38, "CF 30MZ")         ' Sets the parameter.  
    Call ibwrt(u38, "SP 1MZ")  
    Call ibwrt(u38, "DET POS")  
    Call ibwrt(u38, "LON " & LabelBuff) ' Sets the label.  
    Call ibwrt(u38, "SV 5")           ' Saves the data to Register 5.  
    Call ibwrt(u38, "CF 1GZ")         ' Changes the set parameters.  
    Call ibwrt(u38, "SP 200MZ")  
    Call ibwrt(u38, "RC 5")           ' Recalls the data from Register 5.  
    Call ibonl(u38, 0)  
End Sub
```

## Microsoft Visual Basic 2008

```
Imports NationalInstruments.NI4882

Module MdlManualSample_05
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    '*****
    'Example 5 Saving set values in Register 5 and recalling them from Register 5. *
    '*****
    Sub Main()
        Dim LabelBuff As String ' Defines the character string buffer for the label.
        Dim str As String = Space(1)

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        LabelBuff = "/*Cross Domain Analyzer*" ' Sets the label.
        Call GpibDevice.Clear() ' Performs a Device Clear.
        System.Console.WriteLine("Sets the center frequency to 30 MHz.")
        Call GpibDevice.Write("CF 30MZ") ' Sets the parameter.
        System.Console.WriteLine("Sets the span frequency to 1 MHz.")
        Call GpibDevice.Write("SP 1MZ")
        System.Console.WriteLine("Sets the detector mode trace A positive.")
        Call GpibDevice.Write("DET POS")
        System.Console.WriteLine("Sets the label.")
        Call GpibDevice.Write("LON " & LabelBuff) ' Sets the label.
        System.Console.WriteLine("Saves the data to Register 5.")
        Call GpibDevice.Write("SV 5") ' Saves the data to Register 5.
        System.Console.WriteLine("Sets the center frequency to 1 GHz.")
        Call GpibDevice.Write("CF 1GZ") ' Changes the set parameters.
        System.Console.WriteLine("Sets the span frequency to 200 MHz.")
        Call GpibDevice.Write("SP 200MZ")
        System.Console.WriteLine("Recalls the data from Register 5.")
        Call GpibDevice.Write("RC 5") ' Recalls the data from Register 5.

        Call GpibDevice.Dispose()

        System.Console.WriteLine("Please push Enter key.")
        str = System.Console.ReadLine
    End Sub
End Module
```

## 6.14.1 GPIB バス・コントロール用基本ステップ

## 例 6 リミット・ライン 1 テーブル入力して ON する

## Visual Basic 6.0

```

'*****
'* Example 6 Enter Limit line1 in the table and turn Limit line 1 on *
'*****
Private Sub CmdSetup_Click()
    Dim u38 As Integer

    Call ibdev(0, 8, 0, T30s, 1, 0, u38)      ' Initialize
    Call ibclr(u38)                          ' Performs a Device Clear.
    Call ibwrt(u38, "LMTADEL")              ' Clears the table used for Limit Line 1.
    Call ibwrt(u38, "AUNITS DBUV")         ' Sets the level unit to dBμV.
    Call ibwrt(u38, "LMTAINF 25MZ, 49.5DB")
    ' Enters data used by Limit Line 1.
    Call ibwrt(u38, "LMTAINF 35MZ, 49.5DB")
    Call ibwrt(u38, "LMTAINF 35MZ, 51.5DB")
    Call ibwrt(u38, "LMTAINF 55MZ, 51.5DB")
    Call ibwrt(u38, "LMTAINF 55MZ, 54.3DB")
    Call ibwrt(u38, "LMTAINF 65MZ, 54.3DB")
    Call ibwrt(u38, "LMTAINF 65MZ, 57.0DB")
    Call ibwrt(u38, "LMTAINF 68MZ, 57.0DB")
    Call ibwrt(u38, "LMTAINF 68MZ, 60.0DB")
    Call ibwrt(u38, "LMTAINF 75MZ, 60.0DB")
    Call ibwrt(u38, "LMTAINF 75MZ, 62.5DB")
    Call ibwrt(u38, "LMTAINF 82MZ, 62.5DB")
    Call ibwrt(u38, "LMTAINF 82MZ, 64.7DB")
    Call ibwrt(u38, "FA 0MZ")              ' Sets the start frequency to 0 MHz.
    Call ibwrt(u38, "FB 100MZ")           ' Sets the stop frequency to 100 MHz.
    Call ibwrt(u38, "LMTA ON")            ' Turns Limit line 1 on.
    Call ibonl(u38, 0)
End Sub

```

## Microsoft Visual Basic 2008

```

Imports NationalInstruments.NI4882

Module MdlManualSample_06
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    '*****
    '* Example 6 Enter Limit line1 in the table and turn Limit line 1 on. *
    '*****
    Sub Main()
        Dim str As String = Space(1)
        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear()              ' Performs a device clear.
        Call GpibDevice.Write("LMTADEL")    ' Clears the table used for Limit Line 1.
        Call GpibDevice.Write("AUNITS DBUV") ' Sets the level unit to dBμV.
        ' Enters data used by Limit Line 1.
    End Sub
End Module

```

```
dBμV.") System.Console.WriteLine("Sets the limit frequency to 25 MHz and limit level 49.5
dBμV.") Call GpibDevice.Write("LMTAINF 25MZ, 49.5DB")
dBμV.") System.Console.WriteLine("Sets the limit frequency to 35 MHz and limit level 49.5
dBμV.") Call GpibDevice.Write("LMTAINF 35MZ, 49.5DB")
dBμV.") System.Console.WriteLine("Sets the limit frequency to 35 MHz and limit level 51.5
dBμV.") Call GpibDevice.Write("LMTAINF 35MZ, 51.5DB")
dBμV.") System.Console.WriteLine("Sets the limit frequency to 55 MHz and limit level 51.5
dBμV.") Call GpibDevice.Write("LMTAINF 55MZ, 51.5DB")
dBμV.") System.Console.WriteLine("Sets the limit frequency to 55 MHz and limit level 54.3
dBμV.") Call GpibDevice.Write("LMTAINF 55MZ, 54.3DB")
dBμV.") System.Console.WriteLine("Sets the limit frequency to 65 MHz and limit level 54.3
dBμV.") Call GpibDevice.Write("LMTAINF 65MZ, 54.3DB")
dBμV.") System.Console.WriteLine("Sets the limit frequency to 65 MHz and limit level 57.0
dBμV.") Call GpibDevice.Write("LMTAINF 65MZ, 57.0DB")
dBμV.") System.Console.WriteLine("Sets the limit frequency to 68 MHz and limit level 57.0
dBμV.") Call GpibDevice.Write("LMTAINF 68MZ, 57.0DB")
dBμV.") System.Console.WriteLine("Sets the limit frequency to 68 MHz and limit level 60.0
dBμV.") Call GpibDevice.Write("LMTAINF 68MZ, 60.0DB")
dBμV.") System.Console.WriteLine("Sets the limit frequency to 75 MHz and limit level 60.0
dBμV.") Call GpibDevice.Write("LMTAINF 75MZ, 60.0DB")
dBμV.") System.Console.WriteLine("Sets the limit frequency to 75 MHz and limit level 62.5
dBμV.") Call GpibDevice.Write("LMTAINF 75MZ, 62.5DB")
dBμV.") System.Console.WriteLine("Sets the limit frequency to 82 MHz and limit level 62.5
dBμV.") Call GpibDevice.Write("LMTAINF 82MZ, 62.5DB")
dBμV.") System.Console.WriteLine("Sets the limit frequency to 82 MHz and limit level 64.7
dBμV.") Call GpibDevice.Write("LMTAINF 82MZ, 64.7DB")
System.Console.WriteLine("Sets the start frequency to 0 MHz.")
Call GpibDevice.Write("FA 0MZ") ' Sets the start frequency to 0 MHz.
System.Console.WriteLine("Sets the stop frequency to 100 MHz.")
Call GpibDevice.Write("FB 100MZ") ' Sets the stop frequency to 100 MHz.
System.Console.WriteLine("Turns Limit line 1 on.")
Call GpibDevice.Write("LMTA ON") ' Turns Limit line 1 on.

Call GpibDevice.Dispose()

System.Console.WriteLine("Please push Enter key.")
str = System.Console.ReadLine
End Sub

End Module
```

### 6.14.1.3 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。

表 6-11 データ出力フォーマット

出力フォーマット	
周波数系	$\pm \underline{D.DDDDDDDDDDD} \underline{E \pm DD} \underline{CR LF}$ ↑            ↑            ↑            ↑ 1            2            3            4 ・ データ・サイズ (1~3) は最大 19 バイト、単位は Hz (例) “CF?” を指定し、中心周波数を出力する場合等
レベル系	$\pm \underline{D.DDDDDDD} \underline{E \pm DD} \underline{CR LF}$ ↑    ↑            ↑            ↑ 1    2            3            4 ・ データ・サイズ (1~3) は最大 19 バイト、単位は各 UNIT に従う (例) “ML?” を指定し、マーカ・レベルを出力する場合等
時間系	$\pm \underline{D.DDD} \underline{E \pm DD} \underline{CR LF}$ ↑    ↑            ↑            ↑ 1    2            3            4 ・ データ・サイズ (1~3) は最大 19 バイト、単位は sec (例) “SW?” を指定し、掃引時間を出力する場合等
定数系	$\underline{DDDD} \underline{CR LF}$ ↑            ↑ 2            4 ・ データ・サイズの最大バイトは、出力データの最大による (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

- 【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）

## 例 1 マーカ・レベルを読み込み、表示する

## Visual Basic 6.0

```
*****
'* Example 1 Reading and displaying the marker level. *
*****
Private Sub CmdReadMarkerLevel_Click()
    Dim u38 As Integer
    Dim Rdbuf As String
    Dim sep As Integer

    Call ibdev(0, 8, 0, T30s, 1, 0, u38)      ' Initialize
    Call ibclr(u38)                          ' Performs a Device Clear.
    Call ibwrt(u38, "CF 30MZ")               ' Sets the parameter.
    Call ibwrt(u38, "SP 1MZ")
    Call ibwrt(u38, "MLN ON")               ' Marker ON
    Call ibwrt(u38, "MK 30MZ")              ' Sets the marker to 30 MHz.
    Call ibwrt(u38, "TS")
    Call ibwrt(u38, "ML?")                  ' Requests the value of the marker level.
    Rdbuf = Space(30)                       ' Allocates 30 bytes to the buffer area.
    Call ibrd(u38, Rdbuf)                   ' Reads the data (30 bytes Max.).
    sep = InStr(1, Rdbuf, vbCrLf, 0)
    ' Checks the number of character up to the delimiter.
    Label.Caption = "MarkerLevel = " & Left(Rdbuf, sep - 1)
    ' Displays the data on the screen.
    ' An example display:
    ' MarkerLevel = -88.1875
    Call ibonl(u38, 0)
End Sub
```

## Microsoft Visual Basic 2008

```
Imports NationalInstruments.NI4882

Module MdlManualSample_01
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    '*****
    '* Example 1 Reading and displaying the marker level. *
    '*****
    Sub Main()
        Dim Rdbuff As String
        Dim sep As Integer
        Dim str As String = Space(1)

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear() ' Performs a device clear.
        System.Console.WriteLine("Sets the center frequency to 30 MHz.")
        Call GpibDevice.Write("CF 30MZ") ' Sets the parameter.
        System.Console.WriteLine("Sets the span frequency to 1 MHz.")
        Call GpibDevice.Write("SP 1MZ")
        System.Console.WriteLine("Marker ON.")
        Call GpibDevice.Write("MLN ON") ' Marker ON.
        System.Console.WriteLine("Sets the marker to 30 MHz.")
        Call GpibDevice.Write("MK 30MZ") ' Sets the marker to 30 MHz.
        System.Console.WriteLine("Single sweep.")
        Call GpibDevice.Write("TS")
        System.Console.WriteLine("Requests the value of the marker level.")
        Call GpibDevice.Write("ML?") ' Requests the value of the marker
level.
        Rdbuff = Space(30) ' Allocates 30 bytes to the buffer area.
        Rdbuff = GpibDevice.ReadString ' Reads the data (30 bytes Max.).
        sep = InStr(1, Rdbuff, vbCr & vbLf, 0) ' Checks the number of character up to the
delimiter.
        ' Displays the data on the screen.
        ' An example display:
        ' MarkerLevel = -88.1875
        System.Console.WriteLine("MarkerLevel = " & Left(Rdbuff, sep - 1))

        Call GpibDevice.Dispose()

        System.Console.WriteLine("Please push Enter key.")
        str = System.Console.ReadLine
    End Sub
End Module
```



## 例 2 中心周波数を読み込み、表示する

## Visual Basic 6.0

```
*****  
* Example 2 Reading and displaying the center frequency. *  
*****  
Private Sub CmdReadCenterFreq_Click()  
    Dim u38 As Integer  
    Dim Rdbuff As String  
    Dim sep As Integer  
  
    Call ibdev(0, 8, 0, T30s, 1, 0, u38)      ' Initialize  
    Call ibclr(u38)                          ' Performs a Device Clear.  
    Call ibwrt(u38, "CF?")                   ' Query command for the center frequency.  
    Rdbuff = Space(30)                       ' Allocates 30 bytes to the buffer memory.  
    Call ibrd(u38, Rdbuff)                   ' Reads the data (30 bytes Max.)  
    sep = InStr(1, Rdbuff, vbCrLf, 0)        ' Checks the number of character to the delimiter.  
    Label.Caption = "CenterFreq = " & Left(Rdbuff, sep - 1)  
    ' Displays the data on the screen.  
    ' An example display:  
    ' CenterFreq = 30000000#  
    Call ibonl(u38, 0)  
End Sub
```

## Microsoft Visual Basic 2008

```
Imports NationalInstruments.NI4882

Module MdlManualSample_02
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    '*****
    '* Example 2 Reading and displaying the center frequency. *
    '*****
    Sub Main()
        Dim Rdbuff As String
        Dim sep As Integer
        Dim str As String = Space(1)

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear()           ' Performs a Device Clear.
        System.Console.WriteLine("Query command for the center frequency.")
        Call GpibDevice.Write("CF?")     ' Query command for the center frequency.
        Rdbuff = Space(30)                ' Allocates 30 bytes to the buffer memory.
        Rdbuff = GpibDevice.ReadString   ' Reads the data (30 bytes Max.)
        sep = InStr(1, Rdbuff, vbCr & vbLf, 0) ' Checks the number of character to the
delimiter.
        ' Displays the data on the screen.
        System.Console.WriteLine("CenterFreq = " & Left(Rdbuff, sep - 1))

        Call GpibDevice.Dispose()

        System.Console.WriteLine("Please push Enter key.")
        str = System.Console.ReadLine
    End Sub
End Module
```

## 例3 レベルの表示単位およびレベルを読み込み、表示する

## Visual Basic 6.0

```
*****
'* Example 3 Reading the level and display unit and displaying them *
*****
Private Sub CmdReadRefLevel_Click()
    Dim u38 As Integer
    Dim Rdbuf As String
    Dim sep As Integer

    Call ibdev(0, 8, 0, T30s, 1, 0, u38)      ' Initialize
    Call ibclr(u38)                          ' Performs a Device Clear.
    Call ibwrt(u38, "RL?")                   ' Query command for the reference level.
    Rdbuf = Space(30)                        ' Allocates 30 bytes to the buffer memory.
    Call ibrd(u38, Rdbuf)                    ' Reads the data from the spectrum analyzer.
    sep = InStr(1, Rdbuf, vbCrLf, 0)        ' Checks the number of characters to the delimiter.
    Label.Caption = "RefLevel = " & Left(Rdbuf, sep - 1) ' Display the data on the screen.
    Call ibwrt(u38, "AUNITS?")               ' Requests the level unit.
    Rdbuf = Space(3)
    Call ibrd(u38, Rdbuf)
    sep = InStr(1, Rdbuf, vbCrLf, 0)        ' Checks the number of characters to the delimiter.
    Label.Caption = Label.Caption & vbCrLf & "UNIT = " & Left(Rdbuf, sep - 1)
    ' Displays the previous result, followed by a return mark and the most recent result.
    ' An example display:
    ' RefLevel = 0#
    ' UNIT = 0
    Call ibonl(u38, 0)
End Sub
```

## Microsoft Visual Basic 2008

```
Imports NationalInstruments.NI4882

Module MdlManualSample_03
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0

    '*****
    '* Example 3 Reading the level and display unit and displaying them. *
    '*****
    Sub Main()
        Dim Rdbuff As String
        Dim sep As Integer
        Dim tmp As String = Space(256)
        Dim str As String = Space(1)

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear() ' Performs a Device Clear.
        System.Console.WriteLine("Query command for the reference level.")
        Call GpibDevice.Write("RL?") ' Query command for the reference level.
        Rdbuff = Space(30) ' Allocates 30 bytes to the buffer memory.
        Rdbuff = GpibDevice.ReadString ' Reads the data from the U3800.
        sep = InStr(1, Rdbuff, vbCr & vbLf, 0) ' Checks the number of characters to the
delimiter.
        tmp = "RefLevel = " & Left(Rdbuff, sep - 1)
        System.Console.WriteLine(tmp)
        ' Display the data on the screen.
        System.Console.WriteLine("Requests the level unit.")
        Call GpibDevice.Write("AUNITS?") ' Requests the level unit.
        Rdbuff = Space(3)
        Rdbuff = GpibDevice.ReadString
        sep = InStr(1, Rdbuff, vbCr & vbLf, 0) ' Checks the number of characters to the
delimiter.
        ' Displays the previous result, followed by a return mark and the most recent result.
        System.Console.WriteLine(tmp & vbCrLf & "UNIT = " & Left(Rdbuff, sep - 1))

        Call GpibDevice.Dispose()

        System.Console.WriteLine("Please push Enter key.")
        str = System.Console.ReadLine
    End Sub

End Module
```

例 4 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

Visual Basic 6.0

```
*****  
*****  
* Example 4 Executing the 6 dB-down operation, reading the frequency and level and displaying  
them. *  
*****  
*****  
Private Sub Cmd6dBDownOpe_Click()  
    Dim u38 As Integer  
    Dim Rdbuff As String  
    Dim sep As Integer  
  
    Call ibdev(0, 8, 0, T30s, 1, 0, u38)      ' Initialize  
    Call ibclr(u38)                          ' Performs a Device Clear.  
    Call ibwrt(u38, "CF 30MZ")               ' Sets the parameter.  
    Call ibwrt(u38, "SP 20MZ")  
    Call ibwrt(u38, "MKBW 6DB")             ' Sets a 6 dB down measurement.  
    Call ibwrt(u38, "PS")                   ' Executes the peak search.  
    Call ibwrt(u38, "XDB")                  ' Performs the 6 dB down measurement.  
    Call ibwrt(u38, "MFL?")                 ' Requests the value of the marker level and frequency.  
    Rdbuff = Space(50)                      ' Allocates the buffer memory space to 50 bytes.  
    Call ibrd(u38, Rdbuff)                   ' Reads the data (50 bytes Max.) from the u3800.  
    sep = InStr(1, Rdbuff, vbCrLf, 0)       ' Checks the number of characters to the delimiter.  
    Label.Caption = "Marker Freq && Level = " & Left(Rdbuff, sep - 1)  
    ' Displays the data on the screen.  
    ' An example display:  
    ' Marker Freq & Level = 200000#, 1.0234375  
    Call ibonl(u38, 0)  
End Sub
```

## Microsoft Visual Basic 2008

```
Imports NationalInstruments.NI4882

Module MdlManualSample_04
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0

    '*****
    '* Example 4 Executing the 6 dB-down operation, reading the frequency and level and
    '* displaying them. *
    '*****

    Sub Main()
        Dim Rdbuff As String
        Dim sep As Integer
        Dim str As String = Space(1)

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear() ' Performs a Device Clear.
        System.Console.WriteLine("Sets the center frequency to 30 MHz.")
        Call GpibDevice.Write("CF 30MZ") ' Sets the parameter.
        System.Console.WriteLine("Sets the span frequency to 20 MHz.")
        Call GpibDevice.Write("SP 20MZ")
        System.Console.WriteLine("Sets a 6 dB down measurement.")
        Call GpibDevice.Write("MKBW 6DB") ' Sets a 6 dB down measurement.
        System.Console.WriteLine("Executes the peak search.")
        Call GpibDevice.Write("PS") ' Executes the peak search.
        System.Console.WriteLine("Performs the 6 dB down measurement.")
        Call GpibDevice.Write("XDB") ' Performs the 6 dB down measurement.
        System.Console.WriteLine("Requests the value of the marker level and frequency. ")
        Call GpibDevice.Write("MFL?") ' Requests the value of the marker level and
frequency.
        Rdbuff = Space(50) ' Allocates the buffer memory space to 50 bytes.
        Rdbuff = GpibDevice.ReadString ' Reads the data (50 bytes Max.) from the U3800.
        sep = InStr(1, Rdbuff, vbCr & vbLf, 0) ' Checks the number of characters to the
delimiter.
        ' Displays the data on the screen.
        ' An example display:
        ' Marker(Freq & Level = +200000.0, +1.0234375)
        System.Console.WriteLine("Marker Freq & Level = " & Left(Rdbuff, sep - 1))

        Call GpibDevice.Dispose()

        System.Console.WriteLine("Please push Enter key.")
        str = System.Console.ReadLine
    End Sub

End Module
```

## 例 5 OBW を測定し、表示する

## Visual Basic 6.0

```

*****
'* Example 5 Measuring OBW and displaying it *
*****
Private Sub CmdMeasOBW_Click()
    Dim u38 As Integer
    Dim LENG1 As Integer, LENG2 As Integer
    Dim OBW As String
    Dim FC As String
    Dim Rdbuff As String

    Call ibdev(0, 8, 0, T30s, 1, 0, u38) ' Initialize
    Call ibclr(u38) ' Performs a Device Clear.
    Call ibwrt(u38, "CF 30MZ") ' Sends the command already set.
    Call ibwrt(u38, "SP 1MZ")
    Call ibwrt(u38, "MLN ON") ' Marker ON
    Call ibwrt(u38, "MK 30MZ")
    Call ibwrt(u38, "OBWON ON")
    Call ibwrt(u38, "TS")
    Call ibwrt(u38, "OBW?") ' Sends the query command.
    Rdbuff = Space(60) ' Allocates the area to the read buffer.
    Call ibrd(u38, Rdbuff) ' Reads the read buffer (the maximum number of
bytes to be output is determined by the buffer area size).
    ' Formatting output character string
    LENG1 = InStr(1, Rdbuff, Chr(44), 0) ' Searches for the first comma.
    FC = Mid(Rdbuff, 1, LENG1 - 1) ' Reads the character before the comma.
    DoEvents
    LENG2 = InStr((LENG1 + 1), Rdbuff, vbCr, 0) ' Determines the last data by searching for
the delimiter.
    OBW = Mid(Rdbuff, (LENG1 + 1), (LENG2 - LENG1 - 1)) ' Reads the data between the second
comma and the delimiter.
    Label.Caption = "OBW = " & OBW & vbCrLf & "Fc = " & FC & vbCrLf
    ' Displays the data on the screen.
    ' An example display:
    ' OBW = 981000#
    ' FC = 30002500#
    Call ibonl(u38, 0)
End Sub

```

## Microsoft Visual Basic 2008

```

Imports NationalInstruments.NI4882

Module MdlManualSample_05
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    *****
    '* Example 5 Measuring OBW and displaying it. *
    *****

```

## 6.14.1 GPIB バス・コントロール用基本ステップ

```
Sub Main()
    Dim Rdbuff As String
    Dim LENG1 As Integer, LENG2 As Integer
    Dim OBW As String
    Dim FC As String
    Dim str As String = Space(1)

    GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
    Call GpibDevice.Clear() ' Performs a Device Clear.
    System.Console.WriteLine("Sets the center frequency to 30 MHz.")
    Call GpibDevice.Write("CF 30MZ") ' Sends the command already set.
    System.Console.WriteLine("Sets the span frequency to 1 MHz.")
    Call GpibDevice.Write("SP 1MZ")
    System.Console.WriteLine("Marker ON.")
    Call GpibDevice.Write("MLN ON") ' Marker ON.
    System.Console.WriteLine("Sets the marker frequency to 30 MHz.")
    Call GpibDevice.Write("MK 30MZ")
    System.Console.WriteLine("OBW ON.")
    Call GpibDevice.Write("OBWON ON")
    System.Console.WriteLine("Single Sweep.")
    Call GpibDevice.Write("TS")
    System.Console.WriteLine("Sends the query OBW command.")
    Call GpibDevice.Write("OBW? ") ' Sends the query command.
    Rdbuff = Space(60) ' Allocates the area to the read buffer.
    Rdbuff = GpibDevice.ReadString ' Reads the read buffer (the maximum number of bytes
to be output is determined by the buffer area size).
    ' Formatting output character string.
    LENG1 = InStr(1, Rdbuff, Chr(44), 0)
    ' Searches for the first comma.
    FC = Mid(Rdbuff, 1, LENG1 - 1) ' Reads the character before the comma.
    LENG2 = InStr((LENG1 + 1), Rdbuff, vbCr, 0)
    'Determines the last data by searching for the delimiter.
    OBW = Mid(Rdbuff, (LENG1 + 1), (LENG2 - LENG1 - 1))
    ' Reads the data between the second comma and the delimiter.
    ' Displays the data on the screen.
    ' An example display:
    ' OBW = +981000.0
    ' FC = +30002500.0
    System.Console.WriteLine("OBW = " & OBW & vbCr & vbLf & "FC = " & FC & vbCr & vbLf)

    Call GpibDevice.Dispose()

    System.Console.WriteLine("Please push Enter key.")
    str = System.Console.ReadLine
End Sub

End Module
```



例 6 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

### Visual Basic 6.0

```

*****
'* Example 6 Reading and displaying the three largest peak levels *
*****
Private Sub CmdPeakLevel_Click()
    Dim u38 As Integer
    Dim Rdbuff As String
    Dim pk1, pk2, pk3 As String

    Call ibdev(0, 8, 0, T30s, 1, 0, u38) ' Initialize
    Call ibclr(u38) ' Performs a Device Clear.
    Call ibwrt(u38, "CF 0MZ") ' Applies the settings.
    Call ibwrt(u38, "SP 100MZ")
    Call ibwrt(u38, "TS")
    Call ibwrt(u38, "PS") ' Executes the peak search.
    Call ibwrt(u38, "ML?") ' Query command to search for the marker level
    Rdbuff = Space(25) ' Allocates the buffer memory.
    Call ibrd(u38, Rdbuff) ' Receives the output.
    pk1 = Left(Rdbuff, (InStr(1, Rdbuff, vbCr, 1) - 1)) ' Reads the data between the starting
point and the delimiter.
    Call ibwrt(u38, "NXP") ' Searches for the next peak.
    Call ibwrt(u38, "ML?")
    Rdbuff = Space(25)
    Call ibrd(u38, Rdbuff)
    pk2 = Left(Rdbuff, (InStr(1, Rdbuff, vbCr, 1) - 1)) ' Reads the data between the starting
point and the delimiter.
    Call ibwrt(u38, "NXP")
    Call ibwrt(u38, "ML?")
    Rdbuff = Space(25)
    Call ibrd(u38, Rdbuff)
    pk3 = Left(Rdbuff, (InStr(1, Rdbuff, vbCr, 1) - 1)) ' Reads the data between the starting
point and the delimiter.
    Label.Caption = "1st PK = " & pk1 & vbCrLf & "2nd PK = " & pk2 & vbCrLf & "3rd PK = " &
pk3 & vbCrLf
    ' Displays the data on the screen.
    ' An example display:
    ' 1 st PK = -85.5390625
    ' 2 nd PK = -70.046875
    ' 3 rd PK = -86.5546875
    Call ibonl(u38, 0)
End Sub

```

## Microsoft Visual Basic 2008

```
Imports NationalInstruments.NI4882

Module MdlManualSample_06
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    '*****
    '* Example 6 Reading and displaying the three largest peak levels.      *
    '*****
    Sub Main()
        Dim Rdbuff As String
        Dim pk1 As String, pk2 As String, pk3 As String
        Dim str As String = Space(1)

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear()          ' Performs a device clear.
        System.Console.WriteLine("Sets the center frequency to 0 MHz.")
        Call GpibDevice.Write("CF 0MZ") ' Applies the settings.
        System.Console.WriteLine("Sets the span frequency to 100 MHz.")
        Call GpibDevice.Write("SP 100MZ")
        System.Console.WriteLine("Single Sweep.")
        Call GpibDevice.Write("TS")
        System.Console.WriteLine("Executes the peak search.")
        Call GpibDevice.Write("PS")      ' Executes the peak search.
        System.Console.WriteLine("Requests the value of the marker level.")
        Call GpibDevice.Write("ML?")     ' Requests the value of the marker level.
        Rdbuff = Space(25)                ' Allocates the buffer memory.
        Rdbuff = GpibDevice.ReadString   ' Receives the output.
        pk1 = Left(Rdbuff, (InStr(1, Rdbuff, vbCr, 1) - 1))
        ' Reads the data between the starting point and the delimiter.
        System.Console.WriteLine("Searches for the next peak.")
        Call GpibDevice.Write("NXP")     ' Searches for the next peak.
        System.Console.WriteLine("Requests the value of the marker level.")
        Call GpibDevice.Write("ML?")
        Rdbuff = Space(25)
        Rdbuff = GpibDevice.ReadString
        pk2 = Left(Rdbuff, (InStr(1, Rdbuff, vbCr, 1) - 1))
        ' Reads the data between the starting point and the delimiter.
        System.Console.WriteLine("Searches for the next peak.")
        Call GpibDevice.Write("NXP")
        System.Console.WriteLine("Requests the value of the marker level.")
        Call GpibDevice.Write("ML?")
        Rdbuff = Space(25)
        Rdbuff = GpibDevice.ReadString
        pk3 = Left(Rdbuff, (InStr(1, Rdbuff, vbCr, 1) - 1))
        ' Reads the data between the starting point and the delimiter.
        ' Displays the data on the screen.
        ' An example display:
        ' 1st PK = -8.553906250000E+01
        ' 2:      nd(PK = -70.046875)
        ' 3rd PK = -8.655468750000E+01
    End Sub
End Module
```

```
        System.Console.WriteLine("1st PK = " & pk1 & vbCr & vbLf & "2nd PK = " & pk2 & vbCr
& vbLf & "3rd PK = " & pk3 & vbCr & vbLf)

        Call GpibDevice.Dispose()

        System.Console.WriteLine("Please push Enter key.")
        Str = System.Console.ReadLine

    End Sub

End Module
```

## 6.14.1 GPIB バス・コントロール用基本ステップ

## 例 7 Access Channel プログラム例

リモート・コントロールでチャンネル 1、チャンネル 2 に対し設定、読み出しを行う場合、アクティブ・チャンネルを設定する必要があります。非アクティブ・チャンネル側に 1 コマンドのみ設定を行う場合、**Access Channel** コマンドを使用することにより、“X2”、“コマンド”と、2 つのコマンドで完結できます。**Access Channel** コマンドのチャンネル指定は、“X1”または“X2”に続く 1 つのコマンドのみに適用されます。

## Visual Basic 6.0

```
*****  
* Example 7 Access Channel. *  
*****  
Private Sub CmdReadMarkerFreq_Click()  
    Dim u38 As Integer  
    Dim Rdbuff1 As String, Rdbuff2 As String  
  
    Call ibdev(0, 8, 0, T30s, 1, 0, u38)      ' Initialize  
    Call ibclr(u38)                            ' Performs a Device Clear.  
    Call ibwrt(u38, "IP")                      ' preset  
    Call ibwrt(u38, "TSM")                    ' Sweeps both channels once  
    Call ibwrt(u38, "PS")                     ' Peak search for channel 1  
    Call ibwrt(u38, "X2;PS")                  ' Peak search for channel 2  
    Rdbuff1 = Space(30)                       ' Allocates 30 bytes to the buffer area  
    Rdbuff2 = Space(30)                       ' Allocates 30 bytes to the buffer area  
    Call ibwrt(u38, "MF?")                    ' Reads the marker frequency of channel 1  
    Call ibrd(u38, Rdbuff1)                   ' Reads data  
    Call ibwrt(u38, "X2;MF?")                 ' Reads the marker frequency of channel 2  
    Call ibrd(u38, Rdbuff2)                   ' Reads data  
    TextBox.Text = "CH1 Marker Freq = " & Trim(Rdbuff1) & "CH2 Marker Freq = " & Rdbuff2  
    Call ibonl(u38, 0)  
End Sub
```

## Microsoft Visual Basic 2008

```
Imports NationalInstruments.NI4882

Public Class frmManualSample_07
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    '*****
    '* Example 7 Access Channel. *
    '*****
    Private Sub CmdReadMarkerFreq_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CmdReadMarkerFreq.Click
        Dim Rdbuff1 As String = ""
        Dim Rdbuff2 As String = ""

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear()           ' Performs a Device Clear.
        Call GpibDevice.Write("IP")       ' preset
        Call GpibDevice.Write("TSM")     ' Sweeps both channels once
        Call GpibDevice.Write("PS")      ' Peak search for channel 1
        Call GpibDevice.Write("X2;PS")   ' Peak search for channel 2
        Call GpibDevice.Write("MF?")     ' Reads the marker frequency of channel 1
        Rdbuff1 = GpibDevice.ReadString() ' Reads data
        Call GpibDevice.Write("X2;MF?")  ' Reads the marker frequency of channel 2
        Rdbuff2 = GpibDevice.ReadString() ' Reads data
        Me.RichTextBox.Text = "CH1 Marker Freq = " & Rdbuff1 & "CH2 Marker Freq = " & Rdbuff2
        Call GpibDevice.Dispose()
    End Sub
End Class
```

#### 6.14.1.4 トレース・データ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、1001 ポイントまたは 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 1001/501 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、1792～14592 の整数値で表します。（ただし、スケールの枠から上方へ外れた波形については、14592 を超えた値になります。）

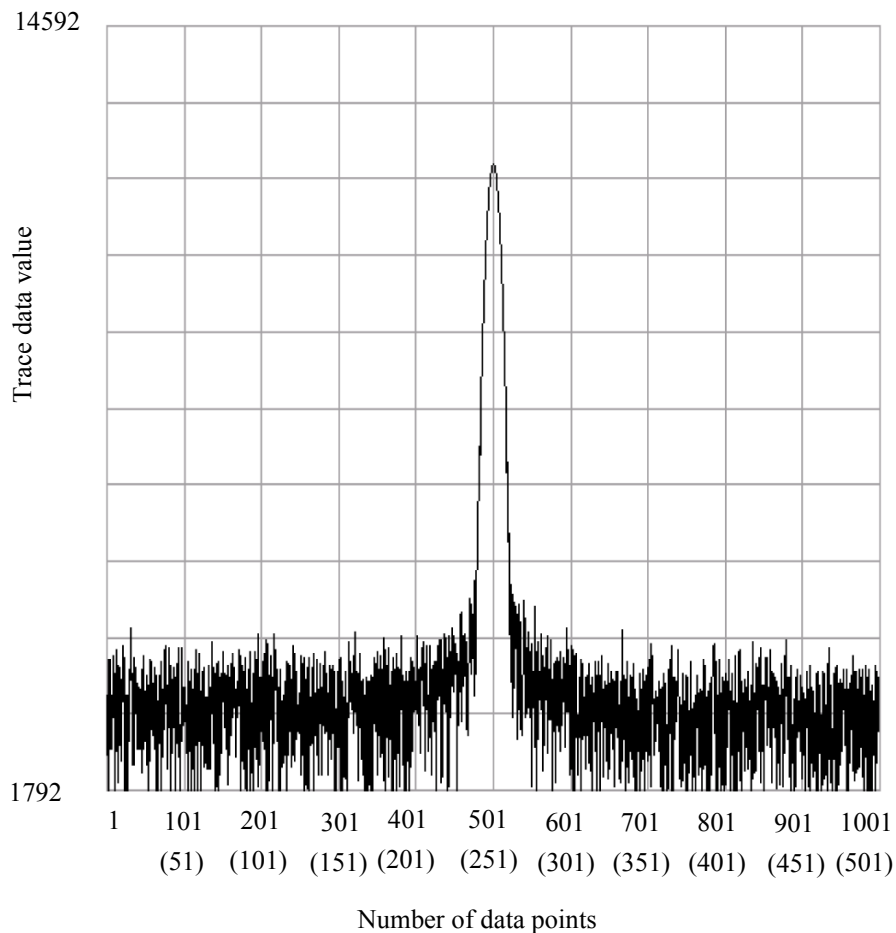


図 6-16 画面格子とトレース・データの関係

トレース・データは、アスキー・データおよびバイナリ・データによる入出力フォーマットがあります。

バイナリ・データで出力するときの出力値は、16bit 整数データまたは絶対値での出力を指定します。フォーマットが“FORM1”または“FORM3”のときはデータの上位バイトから順に、“FORM2”または“FORM4”のときは下位バイトから順に転送します。

---

注 絶対値出力で使用できる単位は“dB”に限られます。  
Units の設定が Watts, Volts のとき、単位は dBm の値が出力されます。

---

表 6-12 トレース・ポイント指定コード

GPIB コード	内容
TPS	測定ポイント数を 501 に設定
TPL	測定ポイント数を 1001 に設定

表 6-13 バイナリ・データ出力フォーマット指定コード

GPIB コード	内容	バイト・オーダ
FORM1 (*1)	1792 ~ 14592 の 16bit 整数値	
FORM2 (*2)	1792 ~ 14592 の 16bit 整数値	順序入れ替え
FORM3 (*2)	絶対値 IEEE 32bit 浮動小数点形式	
FORM4 (*2)	絶対値 IEEE 32bit 浮動小数点形式	順序入れ替え

\*1: 指定がない場合は FORM1

\*2: 入力フォーマットは指定できません。

表 6-14 入出力フォーマット (1/2)

入出力フォーマット	内容		
アスキー・フォーマット	DDDDD CR LF ↑ ↑ 1ポイント分の デリミタ データ  ヘッダの付かない5バイトのデータ		
		入力の GPIB コード	出力の GPIB コード
	A メモリ	TAA	TAA?
	B メモリ	TAB	TAB?
	C メモリ	TAC	TAC?
	O メモリ	TAO	TAO?
M メモリ	TAM	TAM?	

6.14.1 GPIB バス・コントロール用基本ステップ

表 6-14 入出力フォーマット (2/2)

入出力フォーマット	内容	
バイナリ・フォーマット 16bit 整数	<p>FORM1</p> <p style="text-align: center;">DD DD ..... DD DD + EOI</p> <p>1ポイント目の下位バイト      1001/501ポイント目の下位バイト</p> <p>1ポイント目の上位バイト      1001/501ポイント目の上位バイト</p> <p>デリミタ</p> <p>FORM2</p> <p style="text-align: center;">DD DD ..... DD DD + EOI</p> <p>1ポイント目の下位バイト      1001/501ポイント目の下位バイト</p> <p>1ポイント目の上位バイト      1001/501ポイント目の上位バイト</p> <p>デリミタ</p> <p>1ポイントのデータは、バイナリ値が上位と下位の2バイトに分かれている。連続した1001ポイントのデータの終わりには、EOI信号が付加する。</p>	
	入力の GPIB コード	出力の GPIB コード
A メモリ	TBA	TBA?
B メモリ	TBB	TBB?
C メモリ	TBC	TBC?
O メモリ	TBO	TBO?
M メモリ	TBM	TBM?



表 6-15 絶対値出力フォーマット

入出力フォーマット	内容	
バイナリ・ フォーマット 32bit 浮動小数点	<p>FORM3</p> <p style="text-align: center;"> <span style="border-bottom: 1px solid black;">DD</span> <span style="border-bottom: 1px solid black;">DD</span> <span style="border-bottom: 1px solid black;">DD</span> <span style="border-bottom: 1px solid black;">DD</span> ..... <span style="border-bottom: 1px solid black;">DD</span> <span style="border-bottom: 1px solid black;">DD</span> <span style="border-bottom: 1px solid black;">DD</span> <span style="border-bottom: 1px solid black;">DD</span> + EOI                     </p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">                         ↑                          1ポイント目の                          上位バイト                     </div> <div style="text-align: center;">                         ↑                          1ポイント目の                          下位バイト                     </div> <div style="text-align: center;">                         ↑                          1001/501ポイント目の                          上位バイト                     </div> <div style="text-align: center;">                         ↑                          デリミタ                          ↑                          1001/501ポイント目の                          下位バイト                     </div> </div> <p>DDDDDDDD=12345678 のとき、12345678 順で出力</p> <p>FORM4</p> <p style="text-align: center;"> <span style="border-bottom: 1px solid black;">DD</span> <span style="border-bottom: 1px solid black;">DD</span> <span style="border-bottom: 1px solid black;">DD</span> <span style="border-bottom: 1px solid black;">DD</span> ..... <span style="border-bottom: 1px solid black;">DD</span> <span style="border-bottom: 1px solid black;">DD</span> <span style="border-bottom: 1px solid black;">DD</span> <span style="border-bottom: 1px solid black;">DD</span> + EOI                     </p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">                         ↑                          1ポイント目の                          下位バイト                     </div> <div style="text-align: center;">                         ↑                          1ポイント目の                          上位バイト                     </div> <div style="text-align: center;">                         ↑                          1001/501ポイント目の                          下位バイト                     </div> <div style="text-align: center;">                         ↑                          デリミタ                          ↑                          1001/501ポイント目の                          上位バイト                     </div> </div> <p>DDDDDDDD=12345678 のとき、78563412 順で出力</p> <p>1 ポイントのデータは、32 bit (4 バイト) の浮動小数点形式である。連続した 1001 ポイントのデータの終わりには、EOI 信号が付加する。</p>	
		出力の GPIB コード
A メモリ		TBA?
B メモリ		TBB?
C メモリ		TBC?
O メモリ		TBO?
M メモリ		TBM?

## 6.14.1 GPIB バス・コントロール用基本ステップ

## 例 1 トレース・データをアスキーで読み込む

## Visual Basic 6.0

```
*****
'* Example 1 Read the trace data in ASCII format. *
*****
Private Sub CmdReadTraceAscii_Click()
    Dim u38 As Integer
    Dim i As Integer
    Dim res As String
    Dim tr(1000) As String ' Allocates an array in the buffer for 1001 points.

    Call ibdev(0, 8, 0, T30s, 1, 0, u38) ' Initialize
    Call ibclr(u38) ' Performs a Device Clear.
    Call ibwrt(u38, "DLIM0") ' CR LF EOI
    Call ibwrt(u38, "TAA?")
    For i = 0 To 1000 Step 1 ' Repeats the operation for 1001 points.
        tr(i) = Space(7) ' Allocates 7 bytes (5 bytes for the data, and 2
bytes for delimiters).
        Call ibrd(u38, tr(i)) ' Reads the data.
        ' Displays the data on the screen.
        res = res & "tr(" & Str(i) & ") = " & Left(tr(i), 5) & vbCrLf
        DoEvents
    Next i
    TextBox.Text = res
    Call ibonl(u38, 0)
End Sub
```

## Microsoft Visual Basic 2008

```
Imports NationalInstruments.NI4882

Public Class frmManualSample_01
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    '*****
    '* Example 1 Read the trace data in ASCII format. *
    '*****
    Private Sub CmdReadTraceAscii_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CmdReadTraceAscii.Click
        Dim res As String = ""
        Dim tr(1000) As String
        Dim i As Integer

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear()           ' Performs a Device Clear.
        Call GpibDevice.Write("DLIM0")   ' Sets the delimiter to the CR, LF and EOI.
        Call GpibDevice.Write("TAA?")
        For i = 0 To 1000                 ' Repeats the operation for 1001 points.
            tr(i) = GpibDevice.ReadString ' Reads the data.
            res = res & "tr(" & Str(i) & ") = " & Mid(tr(i), 1, 5) & vbCrLf
        Next
        ' Displays the data on the screen.
        Me.RichTextBox.Text = res

        Call GpibDevice.Dispose()
    End Sub
End Class
```

## 6.14.1 GPIB バス・コントロール用基本ステップ

## 例 2 A メモリのデータをバイナリで読み込む

## Visual Basic 6.0

```
*****  
* Example 2 Reading the memory A data in binary format. *  
*****  
Private Sub CmdReadTraceBin_Click()  
    Dim u38 As Integer  
    Dim i As Integer  
    Dim res As String  
    Dim Rslt As Integer, tmp As Integer  
    Dim tr(1000) As Integer ' Allocates an array in the buffer for 1001  
  
    Call ibdev(0, 8, 0, T30s, 1, 0, u38) ' Initialize  
    Call ibclr(u38) ' Performs a Device Clear.  
    Call ibconfig(0, IbcEndBitIsNormal, 0)  
    ' Sets the GPIB-board software so that the end bit of the Ibsta variable is set to 1 only  
    when EOI has been received.  
    Call ibwrt(u38, "DLIM2") ' Sets the delimiter to EOI only.  
    Call ibwrt(u38, "TBA?") ' Requests Trace A in binary data.  
    Call ibrdi(u38, tr(), 1001 * 2) ' Reads 1001 points of binary data.  
    For i = 0 To 1000 Step 1 ' Repeats the operation for 1001 points.  
        tmp = tr(i) ' Swaps the high and low bites.  
        Rslt = (tmp And &HFF&) * 256  
        Rslt = Rslt + ((tmp And &HFF00&) / 256)  
        res = res & Str(Rslt) & vbCrLf  
        'Displays the data on the screen.  
        DoEvents  
    Next i  
    TextBox.Text = res  
    Call ibwrt(u38, "DLIM0") ' Sets the delimiter to the CR, LF and EOI.  
    Call ibconfig(0, IbcEndBitIsNormal, 1)  
    Call ibwrt(u38, "FORM0") ' Switches to FORM0.  
    Call ibonl(u38, 0)  
End Sub
```

## Microsoft Visual Basic 2008

```

Imports NationalInstruments.NI4882

Public Class frmManualSample_02
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    '*****
    '* Example 2 Reading the memory A data in binary format. *
    '*****

    Private Sub CmdReadTraceBin_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CmdReadTraceBin.Click
        Dim res As String = ""
        Dim tmp_buf(1001 * 2) As Byte
        Dim tmp(2) As Byte
        Dim tr(1000) As UShort
        Dim i As Integer

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear() ' Performs a Device Clear.
        ' Sets the GPIB-board software so that the end bit of the Ibsta variable is set to 1
        only when EOI has been received.
        Call GpibDevice.Write("DLIM2") ' Sets the delimiter to EOI only.
        Call GpibDevice.Write("TBA?") ' Requests Trace A in binary data.
        tmp_buf = GpibDevice.ReadByteArray(1001 * 2) ' Reads 1001 points of binary data.
        For i = 0 To 1000 ' Repeats the operation for 1001 points.
            tmp(1) = tmp_buf(i * 2) ' Swaps the high and low bites.
            tmp(0) = tmp_buf(i * 2 + 1)
            tr(i) = System.BitConverter.ToUInt16(tmp, 0)
            res = res & "tr(" & Str(i) & ") = " & Mid(tr(i), 1, 5) & vbCrLf
        Next
        Me.RichTextBox.Text = res
        GpibDevice.Write("DLIM0") ' Sets the delimiter to the CR, LF and EOI.

        Call GpibDevice.Dispose()
    End Sub
End Class

```

## 例3 A メモリにデータをアスキーで入力する

## Visual Basic 6.0

```
*****
'* Example 3 Entering data into memory A in ASCII format *
*****
Private Sub CmdTraceSetAscii_Click()
    Dim u38 As Integer
    Dim trdata(1000) As Integer
    Dim i As Integer

    Call ibdev(0, 8, 0, T30s, 1, 0, u38)    ' Initialize
    trdata(0) = 1792                        ' Creates a temporary test value used to test the
input (*).
    ' If measurement data exists, the steps between the place marked with (*) and this point
are not required.
    For i = 1 To 1000 Step 1
        trdata(i) = Str(Val(trdata(i - 1)) + 12)
        DoEvents
    Next i
    Call ibclr(u38)                          ' Performs a device clear.
    Call ibwrt(u38, "AB")                     ' Sets Trace A to BLANK.
    Call ibwrt(u38, "TAA")                   ' Sets Trace A in ASCII.
    For i = 0 To 1000 Step 1                 ' Repeats the operation for 1001 points.
        ' Sends the value after it has been converted to the ASCII data.
        Call ibwrt(u38, CStr(trdata(i)))
        DoEvents
    Next i
    Call ibwrt(u38, "AV")                     ' Sets Trace A to VIEW.
    Label.Caption = "Sets Trace A to VIEW."
    Call ibonl(u38, 0)
End Sub
```

## Microsoft Visual Basic 2008

```
Imports NationalInstruments.NI4882

Public Class frmManualSample_03
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    '*****
    '* Example 3 Entering data into memory A in ASCII format *
    '*****
    Private Sub CmdTraceSetAscii_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CmdTraceSetAscii.Click
        Dim trdata(1000) As UShort
        Dim i As Integer

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        trdata(0) = 1792          ' Creates a temporary test value used to test the input (*).
        ' If measurement data exists, the steps between the place marked with (*) and this
point are not required.
        For i = 1 To 1000
            trdata(i) = Str(Val(trdata(i - 1) + 12))
        Next
        Call GpibDevice.Clear()          ' Performs a Device Clear.
        Call GpibDevice.Write("AB")      ' Sets Trace A to BLANK.
        Call GpibDevice.Write("TAA")    ' Sets Trace A in ASCII.
        For i = 0 To 1000                ' Repeats the operation for 1001 points.
            ' Sends the value after it has been converted to the ASCII data.
            GpibDevice.Write(CStr(trdata(i)))
        Next
        GpibDevice.Write("AV")          ' Sets Trace A to VIEW.
        Me.Label.Text = "Sets Trace A to VIEW."

        Call GpibDevice.Dispose()
    End Sub
End Class
```

## 6.14.1 GPIB バス・コントロール用基本ステップ

例 4 FORM2 A メモリのデータをバイナリ 16bit 整数（順序入れ替え）で読み込む

Visual Basic 6.0

```
*****  
* Example 4 FORM2 Reading the memory A data in binary 16-bit integer (order swap). *  
*****  
Private Sub CmdReadMemoryA_TraceSwap_Click()  
    Dim u38 As Integer  
    Dim tr(1000) As Integer      ' Allocates an array in the buffer for 1001 points.  
    Dim i As Integer  
    Dim res As String  
  
    Call ibdev(0, 8, 0, T30s, 1, 0, u38)    ' Initialize  
    Call ibclr(u38)                        ' Performs a Device Clear.  
    Call ibwrt(u38, "FORM2")               ' Switches to FORM2.  
    Call ibconfig(0, IbcEndBitIsNormal, 0)  
    ' Sets the GPIB-board software so that the end bit of each Ibsts variable is set to 1 only  
    when EOI has been received.  
    Call ibwrt(u38, "DLIM2")                ' Sets a delimiter to EOI only.  
    Call ibwrt(u38, "TBA?")                 ' Requests Trace A in binary data.  
    Call ibrdi(u38, tr(), 1001 * 2)         ' Reads 1001 points of binary data.  
    For i = 0 To 1000 Step 1                ' Repeats the operation for 1001 points.  
        res = res & Str(tr(i)) & vbCrLf  
        ' Displays the data on the screen.  
        DoEvents  
    Next i  
    TextBox.Text = res  
    Call ibwrt(u38, "DLIM0")                ' Sets the delimiter to the CR, LF and EOI.  
    Call ibconfig(0, IbcEndBitIsNormal, 1)  
    Call ibwrt(u38, "FORM0")               ' Switches to FORM0.  
    Call ibonl(u38, 0)  
End Sub
```



## Microsoft Visual Basic 2008

```
Imports NationalInstruments.NI4882

Public Class frmManualSample_04
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    '*****
    '* Example 4 FORM2 Reading the memory A data in binary 16-bit integer (order swap). *
    '*****
    Private Sub CmdReadTraceA_TraceSwap_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CmdReadTraceA_TraceSwap.Click
        Dim res As String = ""
        Dim tmp_buf(1001 * 2) As Byte
        Dim tr(1000) As UShort
        Dim i As Integer

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear()           ' Performs a Device Clear.
        Call GpibDevice.Write("FORM2")   ' Switches to FORM2.
        Call GpibDevice.Write("DLIM2")   ' Sets a delimiter to EOI only.
        Call GpibDevice.Write("TBA?")    ' Requests Trace A in binary data.
        tmp_buf = GpibDevice.ReadByteArray(1001 * 2) ' Reads 1001 points of binary data.
        For i = 0 To 1000
            tr(i) = System.BitConverter.ToUInt16(tmp_buf, i * 2) ' Repeats the operation
for 1001 points.
            res = res & "tr(" & Str(i) & ") = " & Mid(tr(i), 1, 5) & vbCrLf
        Next
        Me.RichTextBox.Text = res
        GpibDevice.Write("DLIM0")        ' Sets the delimiter to the CR, LF and EOI.
        GpibDevice.Write("FORM0")       ' Switches to FORM0.

        Call GpibDevice.Dispose()
    End Sub
End Class
```

## 6.14.1 GPIB バス・コントロール用基本ステップ

## 例 5 FORM3 A メモリのデータをバイナリ 32bit 浮動小数点で読み込む

## Visual Basic 6.0

```
*****
'* Example 5 FORM3 Reading the memory A data in binary 32-bit floating point *
*****
Private Sub CmdReadTraceFloating_Click()
    Dim u38 As Integer
    Dim tr(1000) As Single      ' Allocates an array in the buffer for 1001 points. 32-bit
floating point type
    Dim i As Integer
    Dim res As String
    Dim tra(4) As Byte         ' Variable which swaps the high-order byte and low-order byte
    Dim tran As Byte          ' Variable which swaps the high-order byte and low-order byte
    Dim tmp As Single          ' Variable which stores converted data

    Call ibdev(0, 8, 0, T30s, 1, 0, u38) ' Initialize
    Call ibclr(u38)                ' Performs a Device Clear.
    Call ibwrt(u38, "FORM3")       ' Switches to FORM3.
    Call ibconfig(0, IbcEndBitIsNormal, 0)
    ' Sets the GPIB-board software so that the end bit of each Ibsta variable is set to 1 only
when EOI has been received.
    Call ibwrt(u38, "DLIM2")       ' Sets a delimiter to EOI only.
    Call ibwrt(u38, "TBA?")        ' Requests Trace A in binary data.
    Call ibrd32(u38, tr(0), 1001 * 4) ' Reads 1001 points of binary data. * 4 for four bytes.
    For i = 0 To 1000 Step 1       ' Repeats the operation for 1001 points.
        Call CopyValtoVal(tra(0), tr(i), 4) ' Copies the 4-byte data stored in tri(i) to tra
(0 to 3).
        tran = tra(0)              ' Swaps the high and low orders.
        tra(0) = tra(3)
        tra(3) = tran
        tran = tra(1)
        tra(1) = tra(2)
        tra(2) = tran
        Call CopyValtoVal(tmp, tra(0), 4) ' Copies the 4-byte data stored in tra (0 to 3),
whose high and low orders are swapped, to tmp.
        res = res & Str(tmp) & vbCrLf
        DoEvents
    Next i
    TextBox.Text = res            ' Displays the data on the screen.
    Call ibwrt(u38, "DLIM0")       ' Sets the delimiter to the CR, LF and EOI.
    Call ibconfig(0, IbcEndBitIsNormal, 1)
    Call ibwrt(u38, "FORM0")       ' Switches to FORM0.
    Call ibonl(u38, 0)
End Sub
```

## Microsoft Visual Basic 2008

```

Imports NationalInstruments.NI4882

Public Class frmManualSample_05
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    '*****
    '* Example 5 FORM3 Reading the memory A data in binary 32-bit floating point *
    '*****

    Private Sub CmdReadTraceFloating_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CmdReadTraceFloating.Click
        Dim res As String = ""
        Dim tmp_buf(1001 * 4) As Byte ' Variable which stores converted allocates an array
in the buffer for 1001 points.
        Dim tmp(4) As Byte ' Variable which swaps the high-order byte and low-order
byte
        Dim tr(1000) As Single ' Allocates an array in the buffer for 1001 points.
32-bit floating point type
        Dim i As Integer

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear() ' Performs a Device Clear.
        Call GpibDevice.Write("FORM3") ' Switches to FORM3.
        Call GpibDevice.Write("DLIM2") ' Sets a delimiter to EOI only.
        Call GpibDevice.Write("TBA?") ' Requests Trace A in binary data.
        ' Reads 1001 points of binary data. * 4 for four bytes.
        tmp_buf = GpibDevice.ReadByteArray(1001 * 4)
        For i = 0 To 1000
            tmp(3) = tmp_buf(i * 4) ' Swaps the high and low orders.
            tmp(2) = tmp_buf(i * 4 + 1)
            tmp(1) = tmp_buf(i * 4 + 2)
            tmp(0) = tmp_buf(i * 4 + 3)
            tr(i) = System.BitConverter.ToSingle(tmp, 0)
            res = res & "tr(" & Str(i) & ") = " & Mid(tr(i), 1, 5) & vbCrLf
        Next
        Me.RichTextBox.Text = res ' Displays the data on the screen.
        Call GpibDevice.Write("DLIM0") ' Sets the delimiter to the CR, LF and EOI.
        Call GpibDevice.Write("FORM0") ' Switches to FORM0.

        Call GpibDevice.Dispose()
    End Sub
End Class

```

## 6.14.1 GPIB バス・コントロール用基本ステップ

例 6 FORM4 A メモリのデータをバイナリ 32bit 浮動小数点（順序入れ替え）で読み込む  
Visual Basic 6.0

```
*****
*
* Example 6 FORM4 Reading the memory A data in binary 32-bit floating point (order swap). *
*****
*
Private Sub CmdReadTraceFloatingOrderSwap_Click()
    Dim u38 As Integer
    Dim tr(1000) As Single ' Allocates an array in the buffer for 1001 points. 32-bit floating
point type
    Dim i As Integer
    Dim res As String

    Call ibdev(0, 8, 0, T30s, 1, 0, u38) ' Initialize
    Call ibclr(u38) ' Performs a Device Clear.
    Call ibwrt(u38, "FORM4") ' Switches to FORM4.
    Call ibconfig(0, IbcEndBitIsNormal, 0)
' Sets the GPIB-board software so that the end bit of each Ibst variable is set to 1 when
EOI has been received.
    Call ibwrt(u38, "DLIM2") ' Sets a delimiter to EOI only.
    Call ibwrt(u38, "TBA?") ' Requests Trace A in binary data.
    Call ibrd32(u38, tr(0), 1001 * 4)
' Reads 1001 points of binary data. * 4 for four bytes.
    For i = 0 To 1000 Step 1 ' Repeats the operation for 1001 points.
        res = res & Str(tr(i)) & vbCrLf
        DoEvents
    Next i
' Displays the data on the screen.
    TextBox.Text = res
    Call ibwrt(u38, "DLIM0") ' Sets the delimiter to the CR, LF and EOI.
    Call ibconfig(0, IbcEndBitIsNormal, 1)
    Call ibwrt(u38, "FORM0") ' Switches to FORM0.
    Call ibonl(u38, 0)
End Sub
```

## Microsoft Visual Basic 2008

```
Imports NationalInstruments.NI4882

Public Class frmManualSample_06
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0

    '*****
    *
    * Example 6 FORM4 Reading the memory A data in binary 32-bit floating point (order swap). *
    '*****
    *

    Private Sub CmdReadTraceFloatingSwap_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CmdReadTraceFloatingSwap.Click
        Dim res As String = ""
        Dim tmp_buf(1001 * 4) As Byte
        Dim tr(1000) As Single
        Dim i As Integer

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear()           'Performs a Device Clear.
        Call GpibDevice.Write("FORM4")    ' Switches to FORM4.
        Call GpibDevice.Write("DLIM2")    ' Sets a delimiter to EOI only.
        Call GpibDevice.Write("TBA?")     ' Requests Trace A in binary data.
        ' Reads 1001 points of binary data. * 4 for four bytes.
        tmp_buf = GpibDevice.ReadByteArray(1001 * 4)
        For i = 0 To 1000                 ' Repeats the operation for 1001 points.
            tr(i) = System.BitConverter.ToSingle(tmp_buf, i * 4)
            res = res & "tr(" & Str(i) & ") = " & Mid(tr(i), 1, 5) & vbCrLf
        Next
        ' Displays the data on the screen.
        Me.RichTextBox.Text = res
        Call GpibDevice.Write("DLIM0")    ' Sets the delimiter to the CR, LF and EOI.
        Call GpibDevice.Write("FORM0")    ' Switches to FORM0.

        Call GpibDevice.Dispose()
    End Sub
End Class
```

### 6.14.1.5 スクリーン・イメージ出力のプログラム例

- 例 1 現在のスクリーン・イメージをビットマップ・データで出力し、ファイルに書き込む。BMP フォーマットでは約 150KB、PNG フォーマットでは約 5KB のデータが出力される。

Visual Basic 6.0

```
*****  
**  
'* Example 1 Outputs a current screen image as bitmap data and writes it to a file.      *  
'* Data, which is approximately 150 KB for the BMP format or approximately 5 KB for the PNG *  
'* format, is output.                                                                *  
'*****  
**  
Private Sub CmdScreenImageOutput_Click()  
    Dim u38 As Integer  
  
    Call ibdev(0, 8, 0, T30s, 1, 0, u38)      ' Initialize  
    Call ibclr(u38)                          ' Performs a Device Clear.  
    Call ibwrt(u38, "DLIM2")                 ' Sets a delimiter to EOI only.  
    Call ibwrt(u38, "BMP?")                 ' Requests bitmap data output.  
    Call ibrdf(u38, "bitmap.bmp")           ' Writes bitmap data to a file.  
    Call ibwrt(u38, "DLIM0")                 ' Returns delimiter setting to CR, LF and EOI.  
    Call ibonl(u38, 0)  
End Sub
```

## Microsoft Visual Basic 2008

```
Imports NationalInstruments.NI4882

Public Class frmManualSample_01
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0

    *****
    **
    '* Example 1 Outputs a current screen image as bitmap data and writes it to a file.      *
    '* Data, which is approximately 150 KB for the BMP format or approximately 5 KB for the PNG *
    '* format, is output.                                                                *
    *****
    **
    Private Sub CmdScreenImageOutput_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CmdScreenImageOutput.Click

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear()                ' Performs a device clear.
        Call GpibDevice.Write("DLIM2")         ' Sets a delimiter to EOI only.
        Call GpibDevice.Write("BMP?")         ' Requests bitmap data output.
        Call GpibDevice.ReadToFile("bitmap.bmp") ' Writes bitmap data to a file.
        Call GpibDevice.Write("DLIMO")        ' Returns delimiter setting to CR, LF and EOI.

        Call GpibDevice.Dispose()
    End Sub
End Class
```

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

例 1 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)

Visual Basic 6.0

```

*****
'* Example 1 Setting the center frequency after resetting this instrument. *
*****
Private Sub CmdSetup_Click()
    Dim u38 As Integer
    Dim Rdbuff As String
    Dim state As Integer
    Dim sep1 As Integer, sep2 As Integer
    Dim i As Integer, j As Integer
    Dim cnt As Integer
    Dim LvlH As String, LvlL As String
    Dim FrqH As String, FrqL As String

    Call ibdev(0, 8, 0, T10s, 1, 0, u38)      ' Initialize
    Call ibclr(u38)                          ' Performs a Device Clear.
    Call ibwrt(u38, "SI")                    ' Sets the single mode.
    Call ibwrt(u38, "CF 1500MZ")            ' Sets the center frequency to 1500 MHz.
    Call ibwrt(u38, "SP 250KZ")            ' Sets the frequency span to 250 kHz.
    Call ibwrt(u38, "RB 1KZ")              ' Sets RBW to 1 kHz.
    Call ibwrt(u38, "VB 3KZ")              ' Sets VBW to 3 kHz.
    Call ibwrt(u38, "ST 5SC")              ' Sets the sweep time to 5 sec.
    Call ibwrt(u38, "CSBSDEL")              ' Clears the channel space and bandwidth previously
set.
    Call ibwrt(u38, "CSBSIN 50KZ,21KZ")    ' Sets CS to 50 kHz, and BS to 21 kHz.
    Call ibwrt(u38, "ACP ON")               ' Starts the ACP measurement.
    For j = 1 To 10 Step 1
        Call ibwrt(u38, "TS")              ' Executes one sweep.
        Call ibwrt(u38, "ACP?")            ' Requests the result of the ACP measurement.
        Rdbuff = Space(3)                  ' Assigns 1 byte for an integer and 2 bytes for
a delimiter before reading the result.
        Call ibrd(u38, Rdbuff)              ' Reads the data.
        cnt = CInt(Rdbuff)                  ' Converts the contents of the buffer into integers.
        For i = 1 To cnt Step 1
            Rdbuff = Space(81)              ' Assigns an area of 81 bytes (Real number x 4 +
', ' x 3 + CRLF).
            Call ibrd(u38, Rdbuff)          ' Reads the data.
            sep1 = InStr(1, Rdbuff, ",", 0) ' Searches for the first comma starting from the
top of the buffer.
            FrqL = Left(Rdbuff, sep1 - 1)   ' Reads the string between the
top and the character string.
            sep2 = InStr(sep1 + 1, Rdbuff, ",", 0) ' Searches for the next comma.
            LvlL = Mid(Rdbuff, sep1 + 1, sep2 - sep1 - 1) ' Reads the string between the
first and second commas.
            sep1 = InStr(sep2 + 1, Rdbuff, ",", 0) ' Searches for the next comma.
            FrqH = Mid(Rdbuff, sep2 + 1, sep1 - sep2 - 1) ' Reads the string between the
second and third commas.
            sep2 = InStr(sep1, Rdbuff, Chr(13), 0) ' Searches for the terminator (CR).
            LvlH = Mid(Rdbuff, sep1 + 1, sep2 - sep1 - 1) ' Reads the string between the
third comma and the CR.
            ' Displays the data on the screen.
            TextBox.Text = TextBox.Text & FrqL & "Hz;" & LvlL & vbCrLf
            TextBox.Text = TextBox.Text & FrqH & "Hz;" & LvlL & vbCrLf
        Next i
    Next j
End Sub

```



```

        Next i
        DoEvents
    Next j
    Call ibon1(u38, 0)
End Sub

```

## Microsoft Visual Basic 2008

```

Imports NationalInstruments.NI4882

Public Class frmManualSample_01
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    *****
    '* Example 1 Setting the center frequency after resetting this instrument. *
    *****
    Private Sub CmdSetup_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles CmdSetup.Click
        Dim j, i As Integer
        Dim cnt As Integer
        Dim Rdbuff As String
        Dim str As String = Space(1)
        Dim sep1 As Integer, sep2 As Integer
        Dim LvlL As String
        Dim FrqH As String, FrqL As String

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear()           ' Performs a device clear.
        Call GpibDevice.Write("SI")       ' Sets the single mode.
        Call GpibDevice.Write("CF 1500MZ") ' Sets the center frequency to 1500 MHz.
        Call GpibDevice.Write("SP 250KZ") ' Sets the frequency span to 250 kHz.
        Call GpibDevice.Write("RB 1KZ")   ' Sets RBW to 1 kHz.
        Call GpibDevice.Write("VB 3KZ")   ' Sets VBW to 3 kHz.
        Call GpibDevice.Write("ST 5SC")   ' Sets the sweep time to 5 sec.
        Call GpibDevice.Write("CSBSDEL")  ' Clears the channel space and bandwidth
previously set.
        Call GpibDevice.Write("CSBSIN 50KZ,21KZ") ' Sets CS to 50 kHz, and BS to 21 kHz.
        Call GpibDevice.Write("ACP ON")    ' Starts the ACP measurement.
        For j = 1 To 10 Step 1
            Call GpibDevice.Write("TS")    ' Executes one sweep.
            Call GpibDevice.Write("ACP?")  ' Requests the result of the ACP measurement.
            Rdbuff = Space(3)              ' Assigns 1 byte for an integer and 2
bytes for a delimiter before reading the result.
            Rdbuff = GpibDevice.ReadString() ' Reads the data.
            cnt = CInt(Rdbuff)              ' Converts the contents of the buffer into
integers.
        Next j
    End Sub
End Class

```

## 6.14.1 GPIB バス・コントロール用基本ステップ

```
        For i = 1 To cnt Step 1
            Rdbuff = Space(81)                ' Assigns an area of 81 bytes
            (Real number x 4 + ',' x 3 + CRLF).
            Rdbuff = GpibDevice.ReadString()  ' Reads the data.
            sep1 = InStr(1, Rdbuff, ",", 0)   ' Searches for the first comma
            starting from the top of the buffer.
            FrqL = Mid(Rdbuff, 1, sep1 - 1)   ' Reads the string between
            the top and the character string.
            sep2 = InStr(sep1 + 1, Rdbuff, ",", 0) ' Searches for the next comma.
            LvlL = Mid(Rdbuff, sep1 + 1, sep2 - sep1 - 1) ' Reads the string between
            the first and second commas.
            sep1 = InStr(sep2 + 1, Rdbuff, ",", 0) ' Searches for the next comma.
            FrqH = Mid(Rdbuff, sep2 + 1, sep1 - sep2 - 1) ' Reads the string between
            the second and third commas.
            sep2 = InStr(sep1, Rdbuff, Chr(13), 0) ' Searches for the terminator (CR).
            LvlL = Mid(Rdbuff, sep1 + 1, sep2 - sep1 - 1) ' Reads the string between
            the third comma and the CR.
            ' Displays the data on the screen.
            RichTextBox.Text = RichTextBox.Text & FrqL & "Hz;" & LvlL & vbCrLf
            RichTextBox.Text = RichTextBox.Text & FrqH & "Hz;" & LvlL & vbCrLf
            RichTextBox.Refresh()
        Next i
    Next

    Call GpibDevice.Dispose()
End Sub
End Class
```

### 6.14.1.7 ステータス・バイトを使用したプログラム例

例 1 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)

Visual Basic 6.0

```
*****  
* Example 1 Execute a single sweep and wait until its finished (when not using SRQ). *  
*****  
Private Sub CmdStatusByteChk_Click()  
    Dim u38 As Integer  
    Dim Rdbuff As String  
    Dim state As Integer  
  
    Call ibdev(0, 8, 0, T30s, 1, 0, u38)      ' Initialize  
    Call ibclr(u38)                          ' Performs a Device Clear.  
    Call ibwrt(u38, "SI")                    ' Turns the single sweep mode on.  
    Call ibwrt(u38, "OPR8")                  ' Enables the sweep-end bit of the operation  
    ' status register.  
    Call ibwrt(u38, "*CLS")                  ' Clears the status byte.  
    Label.Caption = "Sweep Start."  
    Call ibwrt(u38, "SI")                    ' Begins sweeping.  
    Do  
        Call ibwrt(u38, "*STB?")            ' Requests the value of the status byte.  
        Rdbuff = Space(8)                   ' Reserve a maximum of 8 bytes including the delimiter.  
        Call ibrd(u38, Rdbuff)               ' Reads the data.  
        state = Val(Rdbuff)                  ' Converts the character string into numeric values.  
        DoEvents                             ' Checks the loop for other events currently taking place.  
    Loop Until (state And 128)               ' Exits from the loop if the sweep-end bit is set to 1.  
    Label.Caption = "Sweep End."  
    Call ibonl(u38, 0)  
End Sub
```

## Microsoft Visual Basic 2008

```
Imports NationalInstruments.NI4882

Public Class frmManualSample_01
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    '*****
    '* Example 1 Execute a single sweep and wait until its finished (when not using SRQ). *
    '*****
    Private Sub CmdStatusByteChk_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CmdStatusByteChk.Click
        Dim Rdbuff As String
        Dim state As Integer

        GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
        Call GpibDevice.Clear()           ' Performs a Device Clear.
        Call GpibDevice.Write("SI")      ' Turns the single sweep mode on.
        Call GpibDevice.Write("OPR8")    ' Enables the sweep-end bit of the operation
        Call GpibDevice.Write("*CLS")    ' Clears the status byte.
        Label.Text = "Sweep Start."
        Label.Refresh()
        Call GpibDevice.Write("SI")      ' Begins sweeping.
        Do
            Call GpibDevice.Write("*STB?") ' Requests the value of the status byte.
            Rdbuff = Space(8)             ' Reserve a maximum of 8 bytes including the
delimiter.
            Rdbuff = GpibDevice.ReadString ' Reads the data.
            state = Val(Rdbuff)           ' Converts the character string into numeric values.
        Loop Until (state And 128)      ' Exits from the loop if the sweep-end bit is
set to 1.
        Label.Text = "Sweep End."
        Label.Refresh()

        Call GpibDevice.Dispose()
    End Sub
End Class
```

例 2 シングル掃引の終了後、ピーク周波数、レベルを読み込む (SRQ を使用)

### Visual Basic 6.0

```

*****
*****
'* Example 2 Reading the peak frequency and level at the end of a single sweep (when using
SRQ) *
*****
*****
Private Sub CmdSrqWait_Click()
    Dim u38 As Integer
    Dim boardID As Integer
    Dim Rdbuff As String
    Dim res As Integer
    Dim CFLEV As String

    boardID = 0                                ' Sets the board ID.
    Call ibdev(boardID, 8, 0, T30s, 1, 0, u38) ' Initialize
    Call ibclr(u38)                             ' Performs a Device Clear.
    Call ibwrt(u38, "SI")                       ' Turns the single sweep mode on.
    Call ibwrt(u38, "*CLS")                    ' Clears the status byte.
    Call ibwrt(u38, "OPR 8")                   ' Enables the Sweep-end bit of the operation
    ' status register
    Call ibwrt(u38, "*SRE 128")                ' Enables the Operation status bit of the
    ' status byte.
    Call ibwrt(u38, "S0")                      ' Specifies Send mode for the SRQ signal.
    Call ibwrt(u38, "SI")                      ' Begins sweeping.
    Call WaitSRQ(boardID, res)                 ' Waits until SRQ interruption occurs.
    Call ibrsp(u38, res)                       ' Executes serial polling.
    Call ibwrt(u38, "PS")                      ' Executes the peak search.
    Call ibwrt(u38, "MFL?")                   ' Request the values of the marker frequency
and level.
    Rdbuff = Space(43)                         ' Reserves 43 bytes.
    Call ibrd(u38, Rdbuff)                     ' Reads the data.
    CFLEV = Left(Rdbuff, InStr(1, Rdbuff, vbCr, 0) - 1)
    ' Display data on the screen and start a new line.
    TextBox.Text = TextBox.Text & "No. 1/1 Freq ,Level = " & CFLEV & vbCrLf
    Call ibonl(u38, 0)

End Sub

```

### Microsoft Visual Basic 2008

```

Imports NationalInstruments.NI4882

Public Class frmManualSample_02
    Public Delegate Sub GetSrqNotifyDelegate(ByVal readText As String)
    Private GpibDevice As Device
    Private Const U38_GpibBoardId As Integer = 0
    Private Const U38_GpibAddress As Byte = 8
    Private Const U38_GpibSecondaryAddress As Byte = 0
    Private srqNotifyHandler As GetSrqNotifyDelegate

```

## 6.14.1 GPIB バス・コントロール用基本ステップ

```

*****
' Example 2 Reading the peak frequency and level at the end of a single sweep (when using
SRQ)
*
*****
Private Sub CmdSrqrWait_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles CmdSrqrWait.Click
    srqrNotifyHandler = New GetSrqrNotifyDelegate(AddressOf DispRsultData)
    GpibDevice = New Device(U38_GpibBoardId, U38_GpibAddress, U38_GpibSecondaryAddress)
    Call GpibDevice.Clear() ' Performs a Device Clear.
    GpibDevice.Write("SI") ' Turns the single sweep mode on.
    GpibDevice.Write("*CLS") ' Clears the status byte.
    GpibDevice.Write("OPR 8") ' Enables the Sweep-end bit of the operation
    GpibDevice.Write("*SRE 128") ' status register
    GpibDevice.Write("S0") ' status byte.
    GpibDevice.Write("SI") ' Begins sweeping.
    Try
        GpibDevice.Notify(GpibStatusFlags.DeviceServiceRequest, New
NotifyCallback(AddressOf GpibDeviceSrqrNotify), "")
        GpibDevice.Write("SI") ' Begins sweeping.
    Catch exp As Exception
        MessageBox.Show(exp.Message)
    End Try
End Sub

Private Sub GpibDeviceSrqrNotify(ByVal sender As Object, ByVal e As NotifyData)
    Dim Rdbuf As String
    Dim sep As Integer
    Dim CFLEV As String

    Rdbuf = Space(43) ' Allocates 43 bytes to the buffer memory.
    Try
        GpibDevice.SerialPoll()
        GpibDevice.Write("PS") ' Executes the peak search.
        GpibDevice.Write("MFL?") ' Request the values of the marker frequency
and level.
        Rdbuf = GpibDevice.ReadString ' Reads the data (30 bytes Max.)

        sep = InStr(1, Rdbuf, vbCr & vbLf, 0) ' Checks the number of character to the
delimiter.
        CFLEV = "No.1/1 Freq,Level = " & Mid(Rdbuf, 1, InStr(Rdbuf, vbCr, 0) - 1)
        Invoke(srqrNotifyHandler, New Object() {CFLEV})

        Call GpibDevice.Dispose()

    Catch exp As Exception
        MessageBox.Show(exp.Message)
    End Try
End Sub

Private Sub DispRsultData(ByVal disp_dt As String)
    ' Display data on the screen and start a new line.
    RichTextBox.Text = disp_dt
    RichTextBox.Refresh()
End Sub
End Class

```

### 6.14.1.8 LAN のプログラム例

Common function from example 1 to 6

Visual Basic 6.0

```

*****
' Function name
'   SendDtFunc
' Argument
'   dt : transmission command. (string)
' Explanation
'   The transmission command of argument is transmitted to U3800 by TCP/IP.
' Return value
'   0           : Normal
'   0 Excluding: Error number
*****
Private Function SendDtFunc(dt As String) As Integer
    Dim ipErr As Long

    SendDtFunc = 0
    On Error GoTo ErrOcc
    ipErr = 0

    Me.tcpClient.SendData dt + vbCrLf

    If ipErr <> 0 Then
        SendDtFunc = 1
        Exit Function
    End If

    Exit Function

ErrOcc:
    ipErr = Err.Number
    Resume Next

End Function

*****
' Function name
'   ReceiveFunc
' Argument
'   dt : Receive data. (string)
' Explanation
'   It receive it from R3800 by TCP/IP in the argument.
' Return value
'   0           : Normal
'   0 Excluding: Error number
*****

```

## 6.14.1 GPIB バス・コントロール用基本ステップ

```
Private Function ReceiveFunc(ByRef dt As String) As Integer
    Dim ipErr As Long

    ReceiveFunc = 0
    ipErr = 0
    On Error GoTo ErrRecv

    Do While (Me.tcpClient.BytesReceived = 0)
        DoEvents
    Loop
    Me.tcpClient.GetData dt
    Exit Function

ErrRecv:
    ipErr = Err.Number
    ReceiveFunc = ipErr
    Resume Next

End Function

Private Sub Form_Unload(Cancel As Integer)
    Dim ErrRet As Integer

    If tcpClient.State = sckClosed Then Exit Sub

    ErrRet = SendDtFunc("GTL" & vbCrLf)
    tcpClient.Close
End Sub
```



## Microsoft Visual Basic 2008

```
Imports System.Net.Sockets

Public Class frmManualSample_xx
    Private client As TcpClient

    '*****
    ' Function name
    '   SendDtFunc
    ' Argument
    '   dt : transmission command. (string)
    ' Explanation
    '   The transmission command of argument is transmitted to U3800 by TCP/IP.
    ' Return value
    '   0           : Normal
    '   0 Excluding: Error number
    '*****
    Public Function SendDtFunc(ByVal dt As String) As Integer
        Try
            Dim stream As NetworkStream = client.GetStream()
            Dim send_buf() As Byte = System.Text.Encoding.ASCII.GetBytes(dt + ControlChars.CrLf
+ ControlChars.CrLf)
            stream.Write(send_buf, 0, send_buf.Length)
            SendDtFunc = 0
        Catch ex As Exception
            SendDtFunc = 1
        End Try
    End Function

    '*****
    ' Function name
    '   ReceiveFunc
    ' Argument
    '   dt : Receive data. (string)
    ' Explanation
    '   It receive it from U3800 by TCP/IP in the argument.
    ' Return value
    '   0           : Normal
    '   0 Excluding: Error number
    '*****
    Public Function ReceiveFunc(ByRef dt As String) As Integer
        Dim n As Integer
        Try
            Dim stream As NetworkStream = client.GetStream()
            Dim read_buf(4006) As Byte
            n = stream.Read(read_buf, 0, read_buf.Length)
            dt = System.Text.Encoding.Default.GetString(read_buf)
            ReceiveFunc = 0
        Catch ex As Exception
            ReceiveFunc = 1
        End Try
    End Function
End Class
```

## 6.14.1 GPIB バス・コントロール用基本ステップ

```
'*****  
' Function name  
'   QeryCmdFunc  
' Argument  
'   dt : transmission command (character string)  
'   QryDt : Receive data area to transmission command (character string)  
' Explanation  
'   The transmission command of argument is transmitted to U3800 by TCP/IP.  
'   The data reply to QryDt is stored and returned.  
' Return value  
'   0           : Normal  
'   0 Excluding: Error number  
'*****  
Public Function QeryCmdFunc(ByVal dt As String, ByRef QryDt As String) As Integer  
    Dim n As Integer  
    Try  
        Dim stream As NetworkStream = client.GetStream()  
        Dim send_buf() As Byte = System.Text.Encoding.ASCII.GetBytes(dt + ControlChars.CrLf  
+ ControlChars.CrLf)  
        stream.Write(send_buf, 0, send_buf.Length)  
        Dim read_buf(4006) As Byte  
        n = stream.Read(read_buf, 0, read_buf.Length)  
        QryDt = System.Text.Encoding.Default.GetString(read_buf)  
        n = InStr(QryDt, vbCr, vbBinaryCompare)  
        QryDt = Mid(QryDt, 1, n)  
        QeryCmdFunc = 0  
    Catch ex As Exception  
        QeryCmdFunc = 1  
    End Try  
  
End Function  
  
    Private Sub frmManualSample_xx_Disposed(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles Me.Disposed  
        If client.Connected = True Then  
            client.Close()  
        End If  
    End Sub  
  
    Private Sub frmManualSample_xx_Load(ByVal sender As Object, ByVal e As System.EventArgs)  
Handles Me.Load  
        client = New TcpClient  
    End Sub  
End Class
```

## 例 1 Connection LAN Interface

## Visual Basic 6.0

```

*****
'* Example 1 Connection LAN Interface. *
*****
Private Sub CmdLanConnect_Click()
    Dim Ini As String
    Dim ErrRet As Integer

    tcpClient.RemoteHost = "192.168.0.1"      ' Sets IP Address of U38
    tcpClient.Protocol = sckTCPProtocol      ' Sets protocol to TCP
    tcpClient.RemotePort = 5025             ' Sets port no. 5025 of U38
    tcpClient.Connect                       ' Connects to U3800 port
    Do While (tcpClient.BytesReceived = 0)  ' Waits connection
        DoEvents
    Loop
    tcpClient.GetData Ini
    ErrRet = SendDtFunc("REN" & vbCrLf)      ' Remote enable
End Sub

```

## Microsoft Visual Basic 2008

```

Public Class frmManualSample_01
    Private client As TcpClient
    *****
    '* Example 1 Connection LAN Interface. *
    *****
    Private Sub CmdLanConnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CmdLanConnect.Click
        client.Connect("192.168.0.1", "5025")      ' Sets IP Address and port no. "5025"
of U38
        Dim steam As NetworkStream = client.GetStream() ' TCP Connects to U3800 port
        Dim read_buf(1000) As Byte
        Dim intRtn As Integer

        steam.Read(read_buf, 0, read_buf.Length)
        intRtn = SendDtFunc("REN")                  ' Remote enable
    End Sub
End Class

```

## 例 2 Initialize U3800

## Visual Basic 6.0

```
*****
'* Example 2 Initialize U3800. *
*****
Private Sub CmdInitialize_Click()
    Dim Ini As String
    Dim ErrRet As Integer

    tcpClient.RemoteHost = "192.168.0.1"      ' Sets IP Address of U38
    tcpClient.Protocol = sckTCPProtocol      ' Sets protocol to TCP
    tcpClient.RemotePort = 5025              ' Sets port no. 5025 of U38
    tcpClient.Connect                        ' Connects to U3800 port
    Do While (tcpClient.BytesReceived = 0)   ' Waits connection
        DoEvents
    Loop
    tcpClient.GetData Ini
    ErrRet = SendDtFunc("REN" & vbLf)        ' Remote enable

    Call InitU38                             ' Initialize U3800
End Sub

Public Sub InitU38()
    Dim ErrRet As Integer

    ErrRet = SendDtFunc("*CLS" & vbLf)      ' Resets status register
    ErrRet = SendDtFunc("*RST" & vbLf)      ' Resets this instrument
End Sub
```

## Microsoft Visual Basic 2008

```
Public Class frmManualSample_02
    Private client As TcpClient

    '*****
    '* Example 2 Initialize U3800. *
    '*****

    Private Sub CmdInitialize_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CmdInitialize.Click
        client.Connect("192.168.0.1", "5025") ' Sets IP Address and port no. "5025"
of U38
        Dim steam As NetworkStream = client.GetStream() ' TCP Connects to U3800 port
        Dim read_buf(1000) As Byte
        Dim intRtn As Integer

        steam.Read(read_buf, 0, read_buf.Length)
        intRtn = SendDtFunc("REN") ' Remote enable

        Call InitU38() ' Initialize U3800
    End Sub

    Public Sub InitU38()
        Dim intRtn As Integer

        intRtn = SendDtFunc("*CLS") ' Resets status register
        intRtn = SendDtFunc("*RST") ' Resets this instrument
    End Sub
End Class
```

## 例 3 Brief setting of U3800

## Visual Basic 6.0

```
*****
'* Example 3 Brief setting of U3800 *
'* (Set Center freq. to 1.9984 GHz, *
'* Span to 10 MHz and Reference *
'* level to 10 dBm). *
*****
Private Sub CmdSetup_Click()
    Dim Ini As String
    Dim ErrRet As Integer

    tcpClient.RemoteHost = "192.168.0.1" ' Sets IP Address of U38
    tcpClient.Protocol = sckTCPProtocol ' Sets protocol to TCP
    tcpClient.RemotePort = 5025 ' Sets port no. 5025 of U38
    tcpClient.Connect ' Connects to U3800 port
    Do While (tcpClient.BytesReceived = 0) ' Waits connection
        DoEvents
    Loop
    tcpClient.GetData Ini
    ErrRet = SendDtFunc("REN" & vbCrLf) ' Remote enable

    Call U38Setting ' U3800 Setup
End Sub

Public Sub U38Setting()
    Dim ErrRet As Integer

    ErrRet = SendDtFunc("CF 1.9984GZ" & vbCrLf)
    ErrRet = SendDtFunc("SP 10MZ" & vbCrLf)
    ErrRet = SendDtFunc("RL 10DB" & vbCrLf)
End Sub
```

## Microsoft Visual Basic 2008

```
Public Class frmManualSample_03
    Private client As TcpClient

    *****
    '* Example 3 Brief setting of U3800 *
    '* (Set Center freq. to 1.9984 GHz, *
    '* Span to 10 MHz and Reference *
    '* level to 10 dBm). *
    *****
    Private Sub CmdSetup_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles CmdSetup.Click
        client.Connect("192.168.0.1", "5025") ' Sets IP Address and port no. "5025"
of U38
        Dim steam As NetworkStream = client.GetStream() ' TCP Connects to U3800 port
        Dim read_buf(1000) As Byte
        Dim intRtn As Integer

        steam.Read(read_buf, 0, read_buf.Length)
        intRtn = SendDtFunc("REN") ' Remote enable

        Call U38Setting() ' U3800 Setup
    End Sub

    Public Sub U38Setting()
        Dim intRtn As Integer

        intRtn = SendDtFunc("CF 1.9984 GZ")
        intRtn = SendDtFunc("SP 10MZ")
        intRtn = SendDtFunc("RL 10DB")
    End Sub
End Class
```

## 例 4 Read the setting value of U3800

## Visual Basic 6.0

```
*****
'* Example 4 Read the setting value of U3800 *
*****
Private Sub CmdRead_Click()
    Dim Ini As String
    Dim ErrRet As Integer
    Dim CF As String
    Dim SP As String

    CF = Space(20)           ' Prepares the text variable for read
    SP = Space(20)          ' Prepares the text variable for read

    tcpClient.RemoteHost = "192.168.0.1" ' Sets IP Address of U38
    tcpClient.Protocol = sckTCPProtocol  ' Sets protocol to TCP
    tcpClient.RemotePort = 5025          ' Sets port no. 5025 of U38
    tcpClient.Connect                    ' Connects to U3800 port
    Do While (tcpClient.BytesReceived = 0) ' Waits connection
        DoEvents
    Loop
    tcpClient.GetData Ini
    ErrRet = SendDtFunc("REN" & vbCrLf) ' Remote enable

    Call ReadU38Setting(CF, SP)          ' Read U3800
    Label.Caption = "Center freq.: " & CF & "Span freq.: " & SP
End Sub

Public Sub ReadU38Setting(ByRef CF As String, ByRef SP As String)
    Dim ErrRet As Integer

    ErrRet = SendDtFunc("CF?" & vbCrLf) ' Reads request of center freq.
    ErrRet = ReceiveFunc(CF)            ' Reads setting value
    ErrRet = SendDtFunc("SP?" & vbCrLf) ' Reads request of span freq.
    ErrRet = ReceiveFunc(SP)            ' Reads setting value
End Sub
```



## Microsoft Visual Basic 2008

```
Public Class frmManualSample_04
    Private client As TcpClient

    '*****
    '* Example 4 Read the setting value of U3800 *
    '*****
    Private Sub CmdRead_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles CmdRead.Click
        client.Connect("192.168.0.1", "5025") ' Sets IP Address and port no. "5025"
of U38
        Dim steam As NetworkStream = client.GetStream() ' TCP Connects to U3800 port
        Dim read_buf(1000) As Byte
        Dim intRtn As Integer
        Dim CF As String = Space(20)
        Dim SP As String = Space(20)

        steam.Read(read_buf, 0, read_buf.Length)
        intRtn = SendDtFunc("REN") ' Remote enable

        Call ReadU38Setting(CF, SP) ' Read U3800
        Label.Text = "Center freq. = " & CF & "Span freq. = " & SP
    End Sub

    Public Sub ReadU38Setting(ByRef CF As String, ByRef SP As String)
        Dim intRtn As Integer

        intRtn = QeryCmdFunc("CF?", CF) ' Reads request of center freq.
        intRtn = QeryCmdFunc("SP?", SP) ' Reads request of span freq.
    End Sub
End Class
```

## 例 5 Read signal level using the marker function

## Visual Basic 6.0

```
*****
'* Example 5 Read signal level using the marker function *
*****
Private Sub CmdMarkerRead_Click()
    Dim Ini As String
    Dim ErrRet As Integer
    Dim MKFreq As String
    Dim MKLevel As String

    MKFreq = Space(20)           ' Prepares the text variable for read
    MKLevel = Space(20)         ' Prepares the text variable for read

    tcpClient.RemoteHost = "192.168.0.1" ' Sets IP Address of U38
    tcpClient.Protocol = sckTCPProtocol ' Sets protocol to TCP
    tcpClient.RemotePort = 5025 ' Sets port no. 5025 of U38
    tcpClient.Connect           ' Connects to U3800 port
    Do While (tcpClient.BytesReceived = 0) ' Waits connection
        DoEvents
    Loop
    tcpClient.GetData Ini
    ErrRet = SendDtFunc("REN" & vbCrLf) ' Remote enable

    Call ReadMkrSignal(MKFreq, MKLevel) ' Read U3800
    Label.Caption = "Marker freq.: " & MKFreq & "Level: " & MKLevel
End Sub

Public Sub ReadMkrSignal(ByRef MKFreq As String, ByRef MKLevel As String)
    Dim ErrRet As Integer

    ErrRet = SendDtFunc("MLN ON" & vbCrLf) ' Turns on the marker
    ErrRet = SendDtFunc("PS" & vbCrLf) ' Searches peak point of signal
    ErrRet = SendDtFunc("MF?" & vbCrLf) ' Query the marker frequency
    ErrRet = ReceiveFunc(MKFreq) ' Reads marker frequency
    ErrRet = SendDtFunc("ML?" & vbCrLf) ' Query the marker level
    ErrRet = ReceiveFunc(MKLevel) ' Reads marker level
End Sub
```

## Microsoft Visual Basic 2008

```
Public Class frmManualSample_05
    Private client As TcpClient

    '*****
    '* Example 5 Read signal level using the marker function *
    '*****

    Private Sub CmdMarkerRead_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CmdMarkerRead.Click
        client.Connect("192.168.0.1", "5025") ' Sets IP Address and port no. "5025"
of U38
        Dim steam As NetworkStream = client.GetStream() ' TCP Connects to U3800 port
        Dim read_buf(1000) As Byte
        Dim intRtn As Integer
        Dim MKFreq As String = Space(20)
        Dim MKLevel As String = Space(20)

        steam.Read(read_buf, 0, read_buf.Length)
        intRtn = SendDtFunc("REN") ' Remote enable

        Call ReadU38Signal(MKFreq, MKLevel) ' Read U3800
        Label.Text = "Marker freq.: " & MKFreq & "Level : " & MKLevel
    End Sub

    Public Sub ReadU38Signal(ByRef MKFreq As String, ByRef MKLevel As String)
        Dim intRtn As Integer

        intRtn = SendDtFunc("MLN ON") ' Turns on the marker
        intRtn = SendDtFunc("PS") ' Searches peak point of signal
        intRtn = QueryCmdFunc("MF?", MKFreq) ' Query the marker frequency and Reads marker
frequency
        intRtn = QueryCmdFunc("ML?", MKLevel) ' Query the marker level and Reads marker level
    End Sub
End Class
```

## 例 6 Measure Adjacent Channel Power

## Visual Basic 6.0

```

'*****
'* Example 6 Measure Adjacent Channel Power *
'*****
Private Declare Sub Sleep Lib "KERNEL32.DLL" (ByVal dwMilliseconds As Long)
Private Sub CmdAcpRead_Click()
    Dim Ini As String
    Dim ErrRet As Integer
    Dim ResultACP As String

    ResultACP = Space(200) ' Prepares the text variable for read

    tcpClient.RemoteHost = "192.168.0.1" ' Sets IP Address of U38
    tcpClient.Protocol = sckTCPProtocol ' Sets protocol to TCP
    tcpClient.RemotePort = 5025 ' Sets port no. 5025 of U38
    tcpClient.Connect ' Connects to U3800 port
    Do While (tcpClient.BytesReceived = 0) ' Waits connection
        DoEvents
    Loop
    tcpClient.GetData Ini
    ErrRet = SendDtFunc("REN" & vbCrLf) ' Remote enable

    Call MeasACP(ResultACP) ' Read U3800
    Label.Caption = "ACP results : " & ResultACP

End Sub

Public Sub MeasACP(ByRef ResultACP As String)
    Dim ErrRet As Integer
    Dim stb As String

    ErrRet = SendDtFunc("CF 2GZ" & vbCrLf) ' Sets carrier freq.
    ErrRet = SendDtFunc("SP 25MZ" & vbCrLf)
    ErrRet = SendDtFunc("CSBSDEL" & vbCrLf) ' Clears Channel Space param.
    ErrRet = SendDtFunc("ACPSCR CARR" & vbCrLf) ' Sets ACP mode to Carrier
    ErrRet = SendDtFunc("CARRBS 3.84MHz" & vbCrLf) ' Sets Channel Bandwidth
    ErrRet = SendDtFunc("CSBSIN 5MZ,3.84MZ" & vbCrLf) ' Adj. Channel param.
    ErrRet = SendDtFunc("CSBSIN 10MZ,3.84MZ" & vbCrLf) ' Adj. Channel param.
    ErrRet = SendDtFunc("ACPNQST ON" & vbCrLf) ' Sets Nyq. Filter operation to on
    ErrRet = SendDtFunc("SYMRT 3.84MZ" & vbCrLf) ' Sets Symbol rate of filter
    ErrRet = SendDtFunc("RFACT 0.22" & vbCrLf) ' Sets Roll off factor of filter
    ErrRet = SendDtFunc("ACPTM 10" & vbCrLf) ' Sets average times
    ErrRet = SendDtFunc("PMEASAVG ON" & vbCrLf) ' Sets average func. to ON
    ErrRet = SendDtFunc("ACP" & vbCrLf) ' Starts measurement
    ErrRet = SendDtFunc("*CLS;OPR 16;*SRE 128" & vbCrLf) ' Sets status byte
    stb = Space(10)
    Do
        ErrRet = SendDtFunc("*STB?" + vbCrLf)
        ErrRet = ReceiveFunc(stb)
        If CLng(Val(stb)) And 128 Then Exit Do
    Loop
    ErrRet = SendDtFunc("ACP?" + vbCrLf)
    Call Sleep(500)
    ErrRet = ReceiveFunc(ResultACP) ' Reads out all meas. results of ACP
End Sub

```

## Microsoft Visual Basic 2008

```

Public Class frmManualSample_06
  Private client As TcpClient
  '*****
  '* Example 6 Measure Adjacent Channel Power *
  '*****
  Private Sub CmdAcpRead_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles CmdAcpRead.Click
    client.Connect("192.168.0.1", "5025") ' Sets IP Address and port no. "5025"
of U38
    Dim steam As NetworkStream = client.GetStream() ' TCP Connects to U3800 port
    Dim read_buf(1000) As Byte
    Dim intRtn As Integer
    Dim ResultACP As String = Space(200)

    steam.Read(read_buf, 0, read_buf.Length)
    intRtn = SendDtFunc("REN") ' Remote enable

    Call MeasACP(ResultACP) ' Read U3800
    RichTextBox.Text = "ACP result : " & ResultACP
End Sub

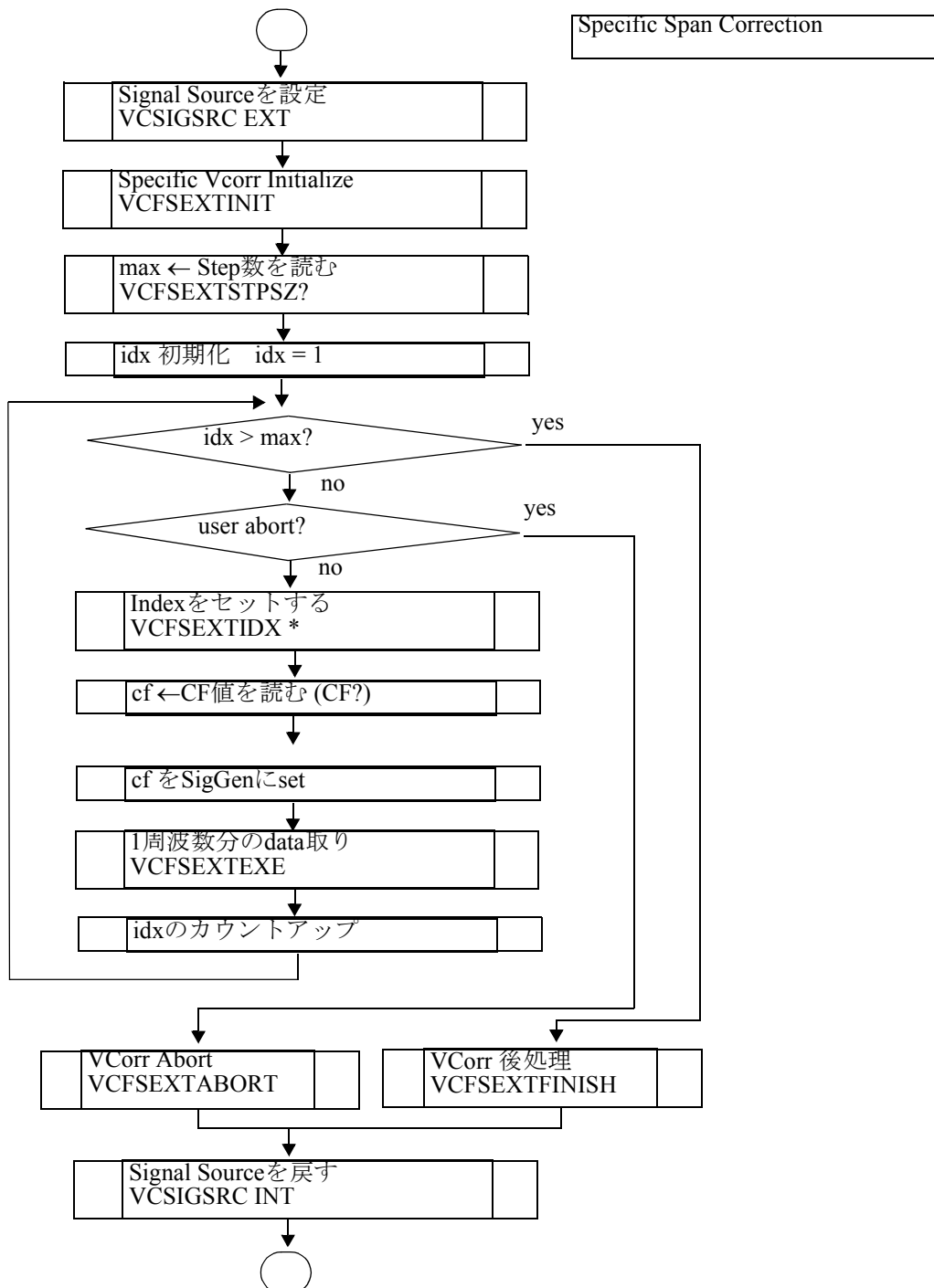
Public Sub MeasACP(ByRef ResultACP As String)
  Dim intRtn As Integer
  Dim stb As String = Space(3)

  intRtn = SendDtFunc("CF 2GZ") ' Sets carrier freq.
  intRtn = SendDtFunc("SP 25MZ")
  intRtn = SendDtFunc("CSBSDEL") ' Clears Channel Space param.
  intRtn = SendDtFunc("ACPSCR CARR") ' Sets ACP mode to Carrier
  intRtn = SendDtFunc("CARRBS 3.84MHz") ' Sets Channel Bandwidth
  intRtn = SendDtFunc("CSBSIN 5MZ,3.84MZ") ' Adj. Channel param.
  intRtn = SendDtFunc("CSBSIN 10MZ,3.84MZ") ' Adj. Channel param.
  intRtn = SendDtFunc("ACPNQST ON") ' Sets Nyq. Filter operation to on
  intRtn = SendDtFunc("SYMRT 3.84MZ") ' Sets Symbol rate of filter
  intRtn = SendDtFunc("RFACT 0.22") ' Sets Roll off factor of filter
  intRtn = SendDtFunc("ACPTM 10") ' Sets average times
  intRtn = SendDtFunc("PMEASAVG ON") ' Sets average func. to ON
  intRtn = SendDtFunc("ACP") ' Starts measurement
  intRtn = SendDtFunc("*CLS;OPR 16;*SRE 128") ' Sets status byte
  stb = Space(10)
  Do
    intRtn = QueryCmdFunc("*STB?", stb)
    If CLng(Val(stb)) And 128 Then Exit Do
  Loop
  intRtn = SendDtFunc("ACP?")
  System.Threading.Thread.Sleep(500)
  intRtn = ReceiveFunc(ResultACP) ' Reads out all meas. results of ACP
End Sub
End Class

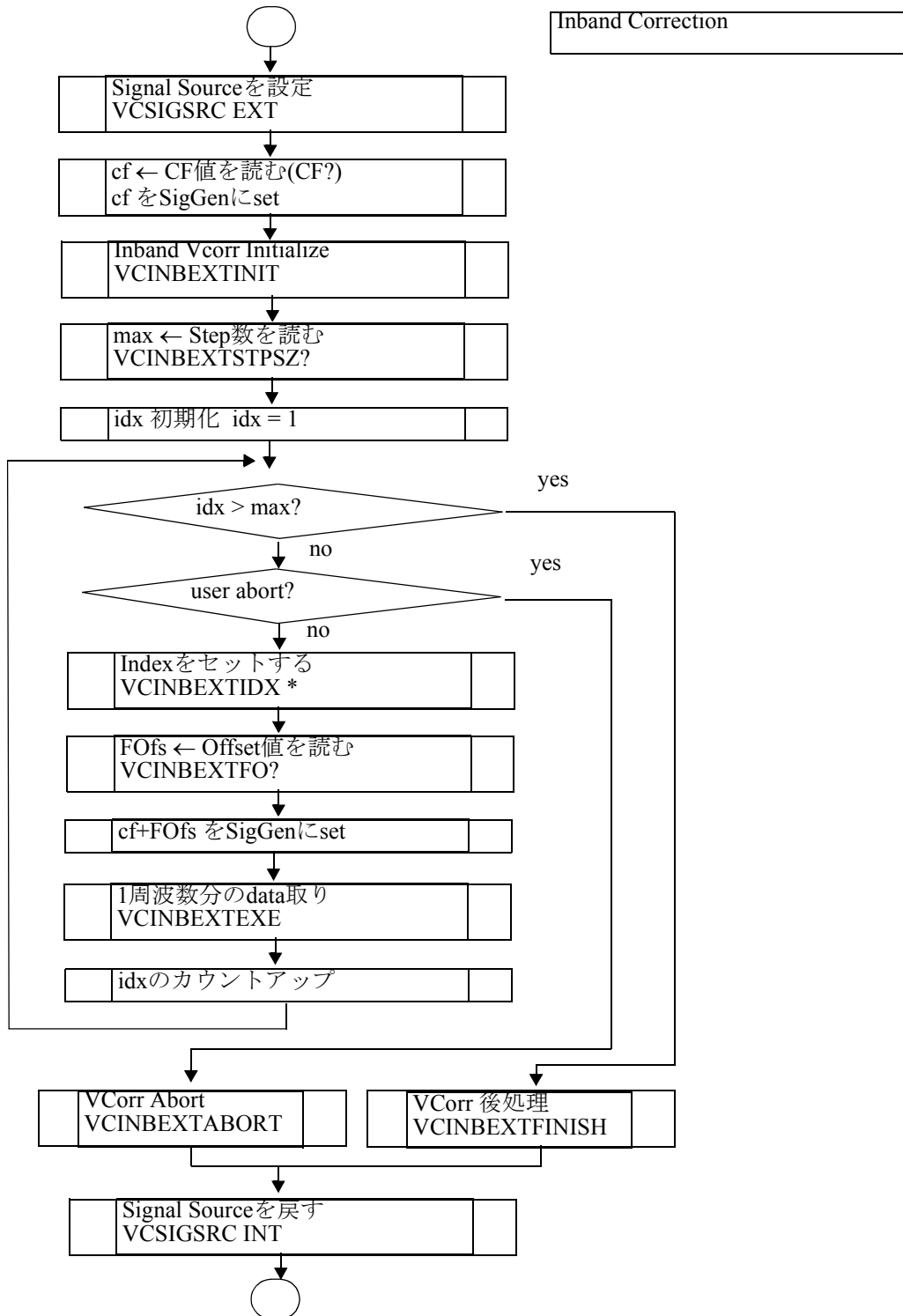
```

### 6.14.2 外部信号源で複数ポイントを補正するプログラム例

外部信号源を使用して複数ポイントを補正するためのリモート・コントロール用プログラムの流れについて説明します。  
 Specific Span 補正を行う場合のリモート・コントロール・プログラムのフローチャートは以下  
 のようになります。



また、Inband 補正を行う場合のリモート・コントロール・プログラムのフローチャートは以下のようになります。



6.14.2 外部信号源で複数ポイントを補正するプログラム例

これら 2 つの補正機能をまとめたプログラムの GPIB 版および LAN 版を例 1、例 2 に示します。

なお、サンプルプログラムは <http://www.advantest.co.jp/products/emi/index.shtml> よりダウンロードできます。

例 1 複数ポイント補正プログラム GPIB 版

- 使用機器  
E8257D (Agilent 社製) PSG アナログ信号発生器 250kHz ~ 50GHz
- GPIB 評価環境  
パソコン  
Microsoft Windows XP SP3 (32bit)  
Professional Ver 2002 (日本語)  
GPIB I/F  
National Instruments GPIB Interfaces ボード  
GPIB コントロール API DLL  
gpiib-32.dll  
ファイル・バージョン 2.5.2.49159  
GPIB API Shell  
開発言語  
Microsoft Visual Basic 6.0(SP6)

```

*****
'* U3800 External SG Vector Correction Start      *
'*                                               *
'* GPIB I/F                                       *
'* National Instruments GPIB Interfaces          *
'*                                               *
'* E8257D (Agilent) PSG Analog Signal Generator *
'* GPIB (SCIP) Command                          *
'* SG_OUTPUT_CMD = "OUTP "                      *
'* SG_OUT_ON_CMD = "ON"                        *
'* SG_OUT_OFF_CMD = "OFF"                      *
'* SG_OUT_LVL_CMD = "SOUR:POW "                *
'* SG_OUT_LVL_UNT = "DBM"                      *
'* SG_SET_FRQ_CMD = "SOUR:FREQ "               *
'* SG_SET_FRQ_UNT = "HZ"                       *
'*                                               *
*****

```



```
Private Sub CmdVCorrStart_Click()
    Dim err As Long
    Dim i As Long
    Dim buf As String
    Dim VCorr_TtlCnt As Long
    Dim strVCorr_TtlCnt As String
    Dim CentFreq As Double
    Dim OfstFreq As Double
    Dim strFreq As String

    Lbl_SG_Start_Freq.Caption = ""
    Lbl_SG_Step_Freq.Caption = ""
    Lbl_SG_End_Freq.Caption = ""
    Lbl_SG_Current_Freq.Caption = ""
    Lbl_Count.Caption = "/"

    Call GetSetting      ' U3800 & SG Setting acquisition of a Textbox.

    '*** GPIB ***
    GpibConnect = True
    If U38_GpibAddress > -1 And SG_GpibAddress > -1 _
        And (U38_GpibAddress <> SG_GpibAddress) Then

        If OpenGpib(U38_GpibAddress) < 0 Then
            err = MsgBox("Can't open the GPIB address(" & CStr(U38_GpibAddress) & ").", _
                vbOKOnly)
            GpibConnect = False
        End If
        If OpenGpib(SG_GpibAddress) < 0 Then
            err = MsgBox("Can't open the GPIB address(" & CStr(SG_GpibAddress) & ").", _
                vbOKOnly)
            GpibConnect = False
        End If
    End If

    If GpibConnect = False Then Exit Sub

    err = PutGpib(U38_GpibAddress, "RFE")           'RefClk=External
    err = PutGpib(U38_GpibAddress, "VCSIGSRC EXT") 'SigSrc EXT

    err = QryGpib(U38_GpibAddress, "CF?", buf)
    If err < 0 Then
        err = MsgBox("CF? query command error.", vbOKOnly)
        GoTo GpibOffline
    End If

    CentFreq = CDBl(buf)
    strFreq = CStr(CentFreq)
    Lbl_SG_Current_Freq.Caption = strFreq
    err = PutGpib(SG_GpibAddress, SG_SET_FRQ_CMD & strFreq & SG_SET_FRQ_UNT)
    Lbl_SG_Start_Freq.Caption = buf
```

## 6.14.2 外部信号源で複数ポイントを補正するプログラム例

```

'Set SG output on
err = PutGpib(SG_GpibAddress, SG_OUT_ON_CMD)

'Set SG output level
err = PutGpib(SG_GpibAddress, SG_OUT_LVL_CMD & CStr(SG_OutputLevel) & SG_OUT_LVL_UNT)
Call Sleep(500)

VCorrAbort = False

If Me.Opt_VCorr_Type1.value = True Then                                     '*** Specific Span ***

    'Xmath 1.Power Ratio Phase Diff Execute ON/OFF
    err = QryGpib(U38_GpibAddress, "XPRPDON?", buf): i = CLng(buf)
    'Xmath 2.Differential Execute ON/OFF
    err = QryGpib(U38_GpibAddress, "XDIFPON?", buf): i = i + CLng(buf)
    'Xmath 4.Power Math Execute ON/OFF
    err = QryGpib(U38_GpibAddress, "XMATHON?", buf): i = i + CLng(buf)
    If i = 0 Then
        err = MsgBox("Xmath Function OFF.", vbOKOnly)
        GoTo GpibOffline
    End If

    err = PutGpib(U38_GpibAddress, "VCFSEXTINIT")                          'Initialize Specific Span
                                                                              'external SG
                                                                              'vector correction.
    err = QryGpib(U38_GpibAddress, "VCFSEXTSTPSZ?", buf)                  'Get Specific Span external
                                                                              'SG vector correction
                                                                              'total count.

    If err < 0 Then
        err = MsgBox("VCFSEXTSTPSZ? query command error.", vbOKOnly)
        Exit Sub
    End If
    VCorr_TtlCnt = CLng(buf)                                               'External SG vector correction
                                                                              'total count.

    strVCorr_TtlCnt = CStr(VCorr_TtlCnt)

    err = QryGpib(U38_GpibAddress, "VCFA?", buf)                          'Start Freq.
    CentFreq = CDBl(buf)
    strFreq = CStr(CentFreq)
    err = PutGpib(SG_GpibAddress, SG_SET_FRQ_CMD & strFreq & SG_SET_FRQ_UNT)
    Call Sleep(1000)
    Lbl_SG_Start_Freq.Caption = buf
    err = QryGpib(U38_GpibAddress, "VCFB?", buf)                          'Stop Freq.
    Lbl_SG_End_Freq.Caption = buf
    err = QryGpib(U38_GpibAddress, "VCFSTEP?", buf)                      'Step Freq.
    Lbl_SG_Step_Freq.Caption = buf

    Lbl_Count.Caption = "0/" & strVCorr_TtlCnt
    Lbl_Count.Refresh

```

```

For i = 1 To VCorr_TtlCnt
  If VCorrAbort = True Then Exit For
  err = PutGpib(U38_GpibAddress, "VCFSEXTIDX " & CStr(i))
  err = QryGpib(U38_GpibAddress, "CF?", buf)
  CentFreq = CDb1(buf)
  strFreq = CStr(CentFreq)
  Lbl_SG_Current_Freq.Caption = strFreq
  err = PutGpib(SG_GpibAddress, SG_SET_FRQ_CMD & strFreq & SG_SET_FRQ_UNT)
  Call Sleep(1000)
  err = PutGpib(U38_GpibAddress, "VCFSEXTEXE")
  Lbl_Count.Caption = CStr(i) & " / " & strVCorr_TtlCnt
  DoEvents
Next i

If VCorrAbort = False Then
  err = PutGpib(U38_GpibAddress, "VCFSEXTFINISH") 'End Specific Span
                                                'vector correction.
Else
  err = PutGpib(U38_GpibAddress, "VCFSEXTABORT") 'Abort Specific Span
                                                'vector correction.
  err = PutGpib(U38_GpibAddress, "VCSIGSRC INT") 'SigSrc INT
End If
Else
  '*** InBand ***
  err = PutGpib(U38_GpibAddress, "VCINBEXTINIT") 'Initialize InBand external
                                                'SG vector correction.
  err = QryGpib(U38_GpibAddress, "VCINBEXTSTPSZ?", buf) 'Get InBand external SG
                                                'vector correction
                                                'total count.

  If err < 0 Then
    err = MsgBox("VCINBEXTSTPSZ? query command error.", vbOKOnly)
    Exit Sub
  End If
  VCorr_TtlCnt = CLng(buf) 'External SG vector correction total count.
  strVCorr_TtlCnt = CStr(VCorr_TtlCnt)

  Lbl_Count.Caption = "0/" & strVCorr_TtlCnt
  Lbl_Count.Refresh

  For i = 1 To VCorr_TtlCnt
    If VCorrAbort = True Then Exit For
    err = PutGpib(U38_GpibAddress, "VCINBEXTIDX " & CStr(i))
    err = QryGpib(U38_GpibAddress, "VCINBEXTFO?", buf)
    OfstFreq = CDb1(buf)
    err = QryGpib(U38_GpibAddress, "CF?", buf)
    CentFreq = CDb1(buf)
    strFreq = CStr(CentFreq + OfstFreq)
    Lbl_SG_Current_Freq.Caption = strFreq
    err = PutGpib(SG_GpibAddress, SG_SET_FRQ_CMD & strFreq & SG_SET_FRQ_UNT)
    Call Sleep(1000)
    err = PutGpib(U38_GpibAddress, "VCINBEXTEXE")
    Lbl_Count.Caption = CStr(i) & " / " & strVCorr_TtlCnt
    DoEvents
  Next i

```

## 6.14.2 外部信号源で複数ポイントを補正するプログラム例

```
    If VCorrAbort = False Then
        err = PutGpib(U38_GpibAddress, "VCINBEXTFINISH")    'End InBand vector
                                                         'correction.
    Else
        err = PutGpib(U38_GpibAddress, "VCINBEXTABORT")    'Abort InBand vector
                                                         'correction.
        err = PutGpib(U38_GpibAddress, "VCSIGSRC INT")    'SigSrc INT
    End If
End If

GpibOffline:
    Call CloseGpib(U38_GpibAddress)
    Call CloseGpib(SG_GpibAddress)

End Sub

'*****
'* U3800 External SG Vector Correction Abort *
'*****
Private Sub CmdVCorrAbort_Click()
    VCorrAbort = True
End Sub
```

## 例2 複数ポイント補正プログラム LAN 版

- 使用機器  
E8257D (Agilent 社製) PSG アナログ信号発生器 250kHz ~ 50GHz
- LAN 評価環境  
パソコン  
Microsoft Windows XP SP3 (32bit)  
Professional Ver 2002 (日本語)  
Ethernet I/F  
3Com 3C920 Integrated Fast Ethernet Controller ボード  
TCP/IP コントロール API DLL  
wsock32.dll  
ファイル・バージョン 5.1.2600.5512  
Windows Socket 32-Bit DLL  
開発言語  
Microsoft Visual Basic 6.0(SP6)

```

*****
'* U3800 External SG Vector Correction Start      *
'*                                               *
'* Ethernet I/F                                *
'*   3Com 3C920 Integrated Fast Ethernet Controller *
'*                                               *
'* E8257D (Agilent) PSG Analog Signal Generator *
'* GPIB (SCPI) Command                          *
'*   SG_OUTPUT_CMD = "OUTP "                    *
'*   SG_OUT_ON_CMD = "ON"                       *
'*   SG_OUT_OFF_CMD = "OFF"                     *
'*   SG_OUT_LVL_CMD = "SOUR:POW "               *
'*   SG_OUT_LVL_UNT = "DBM"                     *
'*   SG_SET_FRQ_CMD = "SOUR:FREQ "              *
'*   SG_SET_FRQ_UNT = "HZ"                      *
'*                                               *
*****
Private Sub CmdVCorrStart_Click()
    Dim err As Long
    Dim i As Long
    Dim buf As String
    Dim VCorr_TtlCnt As Long
    Dim strVCorr_TtlCnt As String
    Dim CentFreq As Double
    Dim OfstFreq As Double
    Dim strFreq As String

```

## 6.14.2 外部信号源で複数ポイントを補正するプログラム例

```
Lbl_SG_Start_Freq.Caption = ""
Lbl_SG_Step_Freq.Caption = ""
Lbl_SG_End_Freq.Caption = ""
Lbl_SG_Current_Freq.Caption = ""
Lbl_Count.Caption = "/"

Call GetSetting      ' U3800 & SG Setting acquisition of a Textbox.

'*** TCP/IP ****
Call InitWinsock

TcpipConnect = True
If OpenSocket(U38_SockID, str_U38_TcpipAddress, U38_PortNo) < 0 Then
    err = MsgBox("Can't open the TCP/IP address(" & str_U38_TcpipAddress & ").", _
                vbOKOnly)
    TcpipConnect = False
Else
    err = GetSocket(U38_SockID, buf)
    err = PutSocket(U38_SockID, "REN")
End If
If OpenSocket(SG_SockID, str_SG_TcpipAddress, SG_PortNo) < 0 Then
    err = MsgBox("Can't open the TCP/IP address(" & str_SG_TcpipAddress & ").", _
                vbOKOnly)
    TcpipConnect = False
End If

If TcpipConnect = False Then Exit Sub

err = PutSocket(U38_SockID, "RFE")                'RefClk=External
err = PutSocket(U38_SockID, "VCSIGSRC EXT")      'SigSrc EXT

err = QrySocket(U38_SockID, "CF?", buf)
If err < 0 Then
    err = MsgBox("CF? query command error.", vbOKOnly)
    GoTo SocketClose
End If

CentFreq = CDb1(buf)
strFreq = CStr(CentFreq)
Lbl_SG_Current_Freq.Caption = strFreq
err = PutSocket(SG_SockID, SG_SET_FRQ_CMD & strFreq & SG_SET_FRQ_UNT)
Lbl_SG_Start_Freq.Caption = buf

'Set SG output on
err = PutSocket(SG_SockID, SG_OUT_ON_CMD)

'Set SG output level
err = PutSocket(SG_SockID, SG_OUT_LVL_CMD & CStr(SG_OutputLevel) & SG_OUT_LVL_UNT)
Call Sleep(500)

VCorrAbort = False
```

```

If Me.Opt_VCorr_Type1.Value = True Then                                     '*** Specific Span ***
    'Xmath 1.Power Ratio Phase Diff Execute ON/OFF
    err = QrySocket(U38_SockID, "XPRPDON?", buf): i = CLng(buf)
    'Xmath 2.Differential Execute ON/OFF
    err = QrySocket(U38_SockID, "XDIFPON?", buf): i = i + CLng(buf)
    'Xmath 4.Power Math Execute ON/OFF
    err = QrySocket(U38_SockID, "XMATHON?", buf): i = i + CLng(buf)
    If i = 0 Then
        err = MsgBox("Xmath Function OFF.", vbOKOnly)
        GoTo SocketClose
    End If

    err = PutSocket(U38_SockID, "VCFSEXTINIT")                            'Initialize Specific Span
                                                                              'external SG vector correction.
    err = QrySocket(U38_SockID, "VCFSEXTSTPSZ?", buf) 'Get Specific Span external SG
                                                                              'vector correction total count.

    If err < 0 Then
        err = MsgBox("VCFSEXTSTPSZ? query command error.", vbOKOnly)
        Exit Sub
    End If

    VCorr_TtlCnt = CLng(buf)                                               'External SG vector correction total count.
    strVCorr_TtlCnt = CStr(VCorr_TtlCnt)

    err = QrySocket(U38_SockID, "VCFA?", buf)                             'Start Freq.
    CentFreq = CDb1(Trim(buf))
    strFreq = CStr(CentFreq)
    err = PutSocket(SG_SockID, SG_SET_FRQ_CMD & strFreq & SG_SET_FRQ_UNT)
    Call Sleep(1000)
    Lbl_SG_Start_Freq.Caption = Trim(buf)
    err = QrySocket(U38_SockID, "VCFB?", buf)                             'Stop Freq.
    Lbl_SG_End_Freq.Caption = Trim(buf)
    err = QrySocket(U38_SockID, "VCFSTEP?", buf)                         'Step Freq.
    Lbl_SG_Step_Freq.Caption = Trim(buf)

    Lbl_Count.Caption = "0/" & strVCorr_TtlCnt
    Lbl_Count.Refresh

    For i = 1 To VCorr_TtlCnt
        If VCorrAbort = True Then Exit For
        err = PutSocket(U38_SockID, "VCFSEXTIDX " & CStr(i))
        err = QrySocket(U38_SockID, "CF?", buf)
        CentFreq = CDb1(Trim(buf))
        strFreq = CStr(CentFreq)
        Lbl_SG_Current_Freq.Caption = strFreq
        err = PutSocket(SG_SockID, SG_SET_FRQ_CMD & strFreq & SG_SET_FRQ_UNT)
        Call Sleep(1000)
        err = PutSocket(U38_SockID, "VCFSEXTEXE")
        Lbl_Count.Caption = CStr(i) & " / " & strVCorr_TtlCnt
        DoEvents
    Next i

```

## 6.14.2 外部信号源で複数ポイントを補正するプログラム例

```

    If VCorrAbort = False Then
        err = PutSocket(U38_SockID, "VCFSEXTFINISH")      'End Specific Span
                                                         'vector correction.
    Else
        err = PutSocket(U38_SockID, "VCFSEXTABORT")      'Abort Specific Span
                                                         'vector correction.
        err = PutSocket(U38_SockID, "VCSIGSRC INT")      'SigSrc INT
    End If
Else
    err = PutSocket(U38_SockID, "VCINBEXTINIT")          '*** InBand ***
                                                         'Initialize InBand external
                                                         'SG vector correction.
    err = QrySocket(U38_SockID, "VCINBEXTSTPSZ?", buf)  'Get InBand external SG
                                                         'vector correction total count.

    If err < 0 Then
        err = MsgBox("VCINBEXTSTPSZ? query command error.", vbOKOnly)
        Exit Sub
    End If
    VCorr_TtlCnt = CLng(buf)                             'External SG vector correction total count.
    strVCorr_TtlCnt = CStr(VCorr_TtlCnt)

    Lbl_Count.Caption = "0/" & strVCorr_TtlCnt
    Lbl_Count.Refresh

    For i = 1 To VCorr_TtlCnt
        If VCorrAbort = True Then Exit For
        err = PutSocket(U38_SockID, "VCINBEXTIDX " & CStr(i))
        err = QrySocket(U38_SockID, "VCINBEXTFO?", buf)
        OfstFreq = CDb1(Trim(buf))
        err = QrySocket(U38_SockID, "CF?", buf)
        CentFreq = CDb1(buf)
        Lbl_SG_Current_Freq.Caption = strFreq
        strFreq = CStr(CentFreq + OfstFreq)
        err = PutSocket(SG_SockID, SG_SET_FRQ_CMD & strFreq & SG_SET_FRQ_UNT)
        Call Sleep(1000)
        err = PutSocket(U38_SockID, "VCINBEXTEXE")
        Lbl_Count.Caption = CStr(i) & " / " & strVCorr_TtlCnt
        DoEvents
    Next i

    If VCorrAbort = False Then
        err = PutSocket(U38_SockID, "VCINBEXTFINISH")    'End InBand vector
                                                         'correction.
    Else
        err = PutSocket(U38_SockID, "VCINBEXTABORT")    'Abort InBand vector
                                                         'correction.
        err = PutSocket(U38_SockID, "VCSIGSRC INT")      'SigSrc INT
    End If
End If

```



```
SocketClose:
    Call CloseConnect(U38_SockID)
    Call CloseConnect(SG_SockID)

    Call CloseWinsock

End Sub

!*****
!* U3800 External SG Vector Correction Abort *
!*****
Private Sub CmdVCorrAbort_Click()
    VCorrAbort = True
End Sub
```