

## GPIB 端末 D / A コンバータ

D A M - 7 0 2 G P B (オープンフレーム)  
D A M - 7 0 2 G P C (ケース入り)

### コマンド説明書 for ASCIIモード

エムシ - アイエンジニアリング株式会社  
〒194-0212 東京都 町田市 小山町 7 8 9 - 9  
TEL 042-705-8312 FAX 042-794-8317



<http://www.mci-eng.co.jp/>

【 概要		
[ - 1 ] 概略動作	_____	2
[ - 2 ] フォーマット	_____	2
[ - 3 ] コマンド	_____	2
[ - 4 ] パラメータ	_____	2
[ - 5 ] デリミタ (ターミネータ)	_____	3
[ - 6 ] エラー処理	_____	3
【 共通コマンド		
[ - 1 ] システムデータ・コマンド	_____	4
[ - 2 ] 内部操作・コマンド	_____	4
[ - 3 ] 同期・コマンド	_____	5
[ - 4 ] ステータス/イベント・コマンド	_____	6
[ - 5 ] デバイストリガ・コマンド	_____	7
【 ステータス報告システム		
[ - 1 ] ステータス・ビット・レジスタ	_____	8
[ - 2 ] スタンダード・イベント・ステータス・レジスタ	_____	9
[ - 3 ] DAM - 702GP・外部・ステータス・レジスタ	_____	10
[ - 4 ] ステータス・レジスタの初期値	_____	11
【 SCMCコマンド		
[ - 1 ] 入力端からの入力コマンド	_____	13
[ - 2 ] 電圧出力端への出力コマンド	_____	16
[ - 3 ] バッファメモリ・コマンド	_____	18
[ - 4 ] バッファリングされたデータの電圧出力コマンド	_____	22
[ - 5 ] ステータス操作コマンド	_____	26
[ - 6 ] アボート・コマンド	_____	28
[ - 7 ] 電圧範囲設定・コマンド	_____	29

## 【 】概要

本「コマンド説明書」ではDAM - 702GPB/CをASCIIモードで使用する場合の操作コマンドについて説明します。ハード的な仕様やバイナリモードでの使用方法については、「取扱説明書」を参照して下さい。

DAM - 702GPB/CのASCIIモードの操作コマンドやその機能は、IEEE - Std 488.2 - 1992に準拠すべく構成、構築されています。(488.2規格は488.1規格の上に成り立っています)

なお、本書ではDAM - 702GPBとDAM - 702GPCのことをDAM - 702GPB/Cと記述しています。

### [ - 1 ] 概略動作

本機「DAM - 702GPB/C」は端末機器です。コントローラ機能はありません。  
従って本機を使用する場合は、GPIBコントローラが必要です。  
通常はGPIBコントローラ機能を持ったパソコンをGPIBコントローラとして使用します。

本機にコマンド(メッセージ)を送信するとコマンドの内容により、本機の端末側の電圧出力端に電圧を出力を出力したり、入力信号「TD1~TD8」のデータを読み取ったりすることができます。  
また、ステータス操作コマンドを使って、端末側入力信号「ST1~ST6、ST8」のデータを読み取ることができます。(本書[ - 1 ][ - 5 ]をご参照ください)  
また、本機の端末側に接続された回路の応答スピードにGPIBバス速度が影響されないよう、本機内のメモリを使用した、バッファリング機能を利用することができます。(本書[ - 3 ][ - 4 ]をご参照ください)  
この機能はパソコンなどで計算した任意の電圧波形データを一定間隔で出力するなどにご利用できます。

488.2規格の共通コマンドを使用すると、本機のソフトウェアリセットや内部状態(ステータス)の読み取りなど、きめ細かな状況把握が可能です。(本書【 】【 】をご参照ください)

以上のような操作がすべてASCII文字列のやりとりで行われ、本機のバイナリモードに比べると大変操作しやすくなっています。  
一般的に、大量のデータのやりとりにASCII文字を使用するとデータ転送の時間が大きくなりますが、それを補うべく、バイナリデータ転送方式もサポートしたコマンドも装備した、大変高機能なモードになっています。

### [ - 2 ] フォーマット

コントローラからのメッセージのフォーマットは下記の二つのタイプがあります。

- 1 : コマンド デリミタ  
コマンドのみで、パラメータを必要としないメッセージです。
- 2 : コマンド パラメータ デリミタ  
パラメータを必要とするコマンドのメッセージです。

DAM - 702GPBからの応答メッセージは、無い場合と、パラメータのみを返送する場合との二つのタイプがあります。どちらの場合でもディップスイッチで選択されたデリミタで終了します。  
(本書[ - 5 ]を参照)

### [ - 3 ] コマンド

488.2で規定されている共通コマンド、および、488.2で規定されているフォーマットに基づいたSCMC(Standard Commands for Measurement and Control)コマンドを使うことができます。

SCMCコマンドの二モニクは  
[]の部分は省略可能です。コマンド文字列の小文字の部分は省略してもかまいません。  
省略しない場合はすべて大文字で表記して下さい。

### [ - 4 ] パラメータ

数値パラメータとして、10進数、16進数、8進数、2進数が使用できます。

16進数数値のフォーマットは	#H数値	(数値は 0,1,2,,,9,A,B,C,D,E,F の組み合わせ)
8進数数値のフォーマットは	#Q数値	(数値は 0,1,2,3,4,5,6,7 の組み合わせ)
2進数数値のフォーマットは	#B数値	(数値は 0,1 の組み合わせ)

数値で表現しないパラメータは英大文字(アルファベット)で表現します。  
入出力ポートの名称など、本機に内蔵される信号名や機能名を指定する場合に使用します。  
各コマンドの解説で具体的な名称が列記されています。

[ - 5 ] デリミタ (ターミネータ)

DAM - 702GPBが応答メッセージの最後に付加するデリミタ (ターミネータ) はディップスイッチで下記の4種類の中から選択することができます。(取扱説明書 [ - 1 ] を参照)

SW6	SW7	デリミタ (ターミネータ)
OFF	OFF	CR + EOI
OFF	ON	CR + LF (NL) + EOI
ON	OFF	EOI
ON	ON	LF (NL) + EOI

この表において、SW8はONであることを前提にしています。

DAM - 702GPBがデリミタとして認識して受け取れるデリミタは下記の4種類です。

- 1 : ニューライン (NL) + EOI
- 2 : ニューライン (NL)
- 3 : EOI
- 4 : ディップスイッチで選択されているデリミタ

この4種類を選択する方法はありません。コマンドやパラメータの組み合わせで自動的に認識します。

[ - 6 ] エラー処理

文法エラー : 本機が受け取ったコマンドがフォーマットに適合していない場合や未定義コマンドの場合、文法エラーになります。  
このエラーが発生するとスタンダード・イベント・ステータス・レジスタのbit5 (CME) がON (1) になります。

対処 : 正しいコマンドを再度送って下さい。

実行エラー : コマンドがフォーマットに適合していても、範囲外パラメータの場合、実行エラーになります。  
また、事前のコマンドにより、本機が実行中の作業と排他にしなければならない場合も実行エラーになります。(排他の関係は各コマンドの説明を参照)  
このエラーが発生するとスタンダード・イベント・ステータス・レジスタのbit4 (EXE) がON (1) になります。

対処 : 正しいパラメータに修正して、再度送って下さい。  
または、排他関係を確認し、実行可能な時に送って下さい。

クエリエラー : クエリ (応答) を必要とするコマンドを本機に与えないで本機をトーカーに指定するとクエリエラーが発生し、スタンダード・イベント・ステータス・レジスタのbit2 (QYE) がON (1) になります。  
また、クエリ (応答) を必要とするコマンドを与えた後、クエリ (応答) を必要としないコマンドを与えると前コマンドに対するクエリ (応答) は消滅します。  
消滅した後、本機をトーカーに指定した場合もクエリエラーになります。

対処 : クエリ (応答) を必要とするコマンドを送った後、シリアルポルを行ってステータス・バイト・レジスタのbit4 (MAV) がON (1) になっていることを確認してからトーカーに指定して下さい。

機器エラー : 本機は電源投入直後、プログラムROMとシステムワークRAMをチェックします。  
チェックの結果、異常を発見するとスタンダード・イベント・ステータス・レジスタのbit3 (DDE) をON (1) にします。

対処 : 一度電源を断にし、再度電源を投入してもこのエラーが発生する場合は修理に出して下さい。  
(なお、\*TST?によるセルフテストでの異常の場合も同様に修理が必要です。)

## 【 】 共通コマンド

### [ - 1 ] システムデータ・コマンド

\*IDN? 識別クエリ (Identification Query)

書式 \*IDN?

説明 バスに接続されている機器を識別する文字列を読み出します。

応答 当コマンドを受信した後、本機はトーカに指定されると  
<製造業者>、<モデル番号>、<シリアル番号>、<ファームウェアのバージョン>を表す、  
下記の文字列を返します。

MC1-ENG,DAM-702GP,000000,REV1.00

### [ - 2 ] 内部操作コマンド

\*RST リセット (Reset)

書式 \*RST

説明 機器をリセットします。

下記の内容のリセットを行います。

- \* 2つのD/A変換・電圧出力を0Vにする
- \* GPIBからの受信バッファをクリアする
- \* PLAYコマンドシステム、MEMORYコマンドシステムを初期状態にする
- \* 前に受け取った\*OPCまたは\*OPC?コマンドをクリアする

下記の内容はリセットされません。

- \* GPIBアドレスまたはそのアドレス内容
- \* 出力待ち行列の中のデータ・バイト
- \* ステータス・バイト・レジスタ
- \* サービス・リクエスト・イネーブル・レジスタ
- \* スタンダード・イベント・ステータス・レジスタ
- \* スタンダード・イベント・イネーブル・レジスタ
- \* 外部・ステータス・条件・レジスタ
- \* 外部・ステータス・トランジション・レジスタ
- \* 外部・ステータス・イベント・レジスタ
- \* 外部・ステータス・イネーブル・レジスタ
- \* 電源オン・フラグ

補足：488.1および488.2で定義される3つのリセットの概略を下記に示します。

- レベル1：IFCラインによりリスナ/トーカを解除  
システムコントローラに制御を返す
- レベル2：DCLおよびSDCにより機器の入出力バッファをクリアし、  
新しいコマンドを受け取れるようにする
- レベル3：\*RSTにより機器自体を実際にクリアする

応答 当コマンドに対する応答メッセージはありません。

\*TST? セルフテストクエリ (Self-Test Query)

書式 \*TST?

説明 機器に内部セルフテストを実行させ、テストの結果を報告させます。  
テストの内容は下記の2点です。  
プログラムROMのサムチェック  
ユーザワークRAMのリードライトチェック

現在実行中の作業がある場合はテストの実行はできません。  
テストの実行を行った場合はMEMORYコマンドシステム、PLAYコマンドシステムは初期化されます。  
初期化の結果、メモリに書き込まれていたデータは破棄されます。  
電圧出力端に出力されている電圧値、ステータス報告システムの各レジスタ、は初期化されません。

応答 当コマンドを受信すると本機はセルフテストを実行し、トーカーに指定されると結果を報告します。  
結果の内容は下記の数値(10進数の整数)のいずれかで、エラーがあった場合の数値は負です。

0 テストはすべて正常  
-1 プログラムROMのチェックサムエラー  
-2 ユーザワークRAMのリードライトエラー  
-4 ユーザデータRAMのリードライトエラー  
90 実行中の作業があったため、テストを実行しなかった。

複数のエラーが発生した場合の数値は各エラーの数値の和を報告します。  
(例えば、-1と-2のエラーが発生すると-3を報告します。)

### [ - 3 ] 同期コマンド

\*OPC 動作完了 (Operation Complete)

書式 \*OPC

説明 実行待ち動作がすべて完了したら、スタンダード・イベント・ステータス・レジスタのビット0をセットするように機器に命令します。

応答 当コマンドを受信すると本機は現在実行中の作業がすべて終了したらスタンダード・イベント・ステータス・レジスタのビット0をセットします。

\*OPC? 動作完了 (Operation Complete Query)

書式 \*OPC?

説明 実行待ち動作がすべて完了したら、機器の出力待ち行列 (GP I Bへの送信バッファ) にASCII「1」を入れるように機器に命令します。

応答 当コマンドを受信すると本機は現在実行中の作業がすべて終了したら出力待ち行列にASCII「1」を入れます。その後、トーカーに指定されると、それを送信します。

\*WAI 続行待ち (Wait-to-Continue)

書式 \*WAI

説明 前に受け取ったコマンドやクエリがすべて終了するまで、新たなコマンドの実行を保留させます。

応答 当コマンドを受信すると本機は現在実行中の作業がすべて終了するまで新たなコマンドを実行しません。  
現在実行中の作業がすべて終了するとあらたなコマンドを実行します。

関連 \*OPC, \*OPC?

#### [ - 4 ] ステータス / イベント ・ コマンド

\*CLS        ステータス ・ クリア (Clear Status)

書式 \*CLS

説明 ステータスに関する下記のレジスタをクリアします。  
スタンダード ・ イベント ・ ステータス ・ レジスタのすべてのビット  
外部 ・ ステータス ・ イベント ・ レジスタのすべてのビット

応答 このコマンドに対する応答はありません。

\*ESE        スタンダード ・ イベント ・ ステータス ・ イネーブル (Standard Event Status Enable)

書式 \*ESE 設定値

説明 スタンダード ・ イベント ・ イネーブル ・ レジスタに設定値をセットします。  
設定値は " 0 " から " 2 5 5 " までの値を 1 0 進数または 1 6、 8、 2 進数で表します。

応答 このコマンドに対する応答はありません。

\*ESE?        スタンダード ・ イベント ・ ステータス ・ イネーブル ・ クエリ (Event Status Enable Query)

書式 \*ESE?

説明 スタンダード ・ イベント ・ イネーブル ・ レジスタの内容を読み出します。

応答 戻り値は " 0 " から " 2 5 5 " の範囲の 1 0 進数整数値です。

\*ESR?        イベント ・ ステータス ・ レジスタ ・ クエリ (Event Status Register Query)

書式 \*ESR?

説明 スタンダード ・ イベント ・ ステータス ・ レジスタの内容を読み出します。  
読み出されたスタンダード ・ イベント ・ ステータス ・ レジスタはクリアされます。

応答 戻り値は " 0 " から " 2 5 5 " の範囲の 1 0 進数整数値です。

\*SRE        サービス ・ リクエスト ・ イネーブル (Service Request Enable)

書式 \*SRE 設定値

説明 サービス ・ リクエスト ・ イネーブル ・ レジスタに設定値をセットします。  
設定値は " 0 " から " 2 5 5 " までの値を 1 0 進または 1 6、 8、 2 進数で表します。

応答 このコマンドに対する応答はありません。

\*SRE?        サービス ・ リクエスト ・ イネーブル ・ クエリ (Service Request Enable Query)

書式 \*SRE?

説明 サービス ・ リクエスト ・ イネーブル ・ レジスタの内容を読み出します。

応答 値は " 0 " から " 6 3 "、 " 1 2 8 " から " 1 9 1 " の範囲の 1 0 進数整数値です。

\*STB?        ステータス ・ バイト ・ クエリ (Read Status Byte Query)

書式 \*STB?

説明 ステータス ・ バイトを読み出します。

応答 戻り値は " 0 " から " 2 5 5 " の範囲の 1 0 進数整数値です。

## [ - 5 ] デバイストリガ・コマンド

\*TRG           トリガ (Trigger)

書式   \*TRG

説明   I - E E E 4 8 8 . 1 規格の G E T コマンドと同じです。  
          プレイ動作を起動させます。(本書 [ - 4 ] をご参照ください。)

応答   このコマンドに対する応答はありません。



## 【 】ステータス報告システム

### [ - 1 ]ステータス・バイト・レジスタ

bit 0 : EXS : 外部端末側ステータス・レジスタを代表するサマリ・ビット

bit 1 : : 本機においては常に 0 です。

bit 2 : : 本機においては常に 0 です。

bit 3 : : 本機においては常に 0 です。

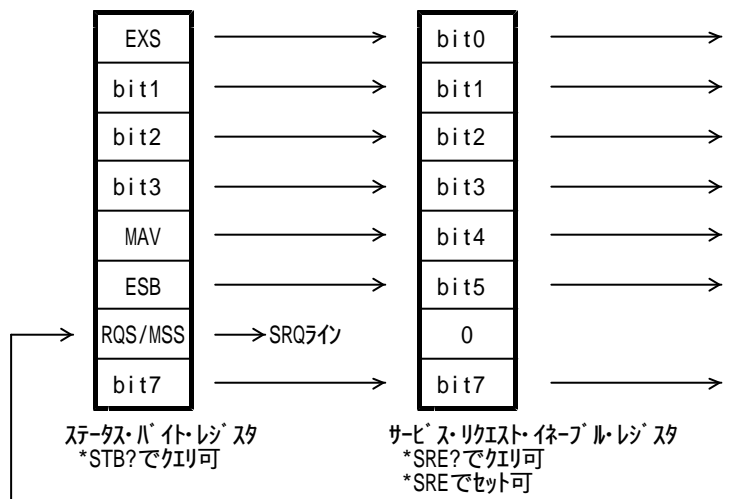
bit 4 : MAV : メッセージ・アベイラブル・ビット  
機器のデータ出力の待ち行列が空であるかどうかを示します。  
本機の GPIB 送信バッファに送信データがある場合、1 にセットされます。

bit 5 : ESB : イベント・ステータス・ビット  
あらかじめ許可された「スタンダード・イベント」が発生した場合、1 にセットされます。

bit 6 : RQS : リクエスト・サービス・ビット  
シリアル・ポールで読み出された場合、本機がサービス・リクエストを発生している場合、1 にセットされています。

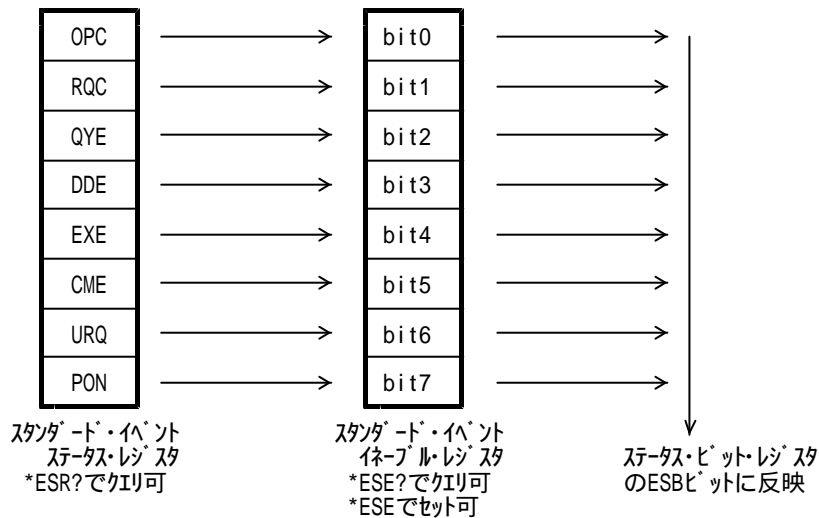
MSS : マスター・ステータス・サマリ  
ステータス・ビット・レジスタの他の 7 ビットの代表。  
過去に本機がサービス・リクエストを発生したかどうかを示します。  
シリアル・ポールによって RQS ビットがクリアされた後も MSS ビットはクリアされません。

bit 7 : : 本機においては常に 0 です。



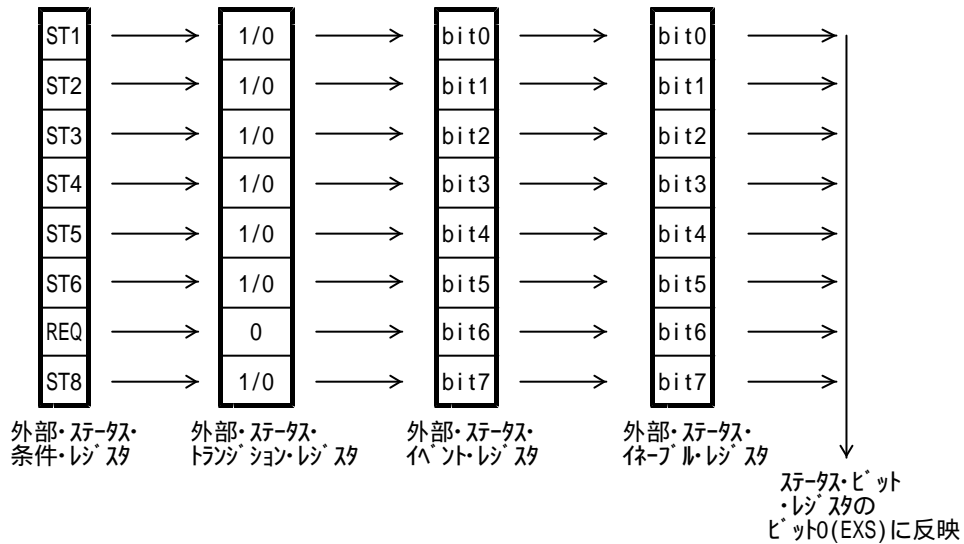
[ - 2 ] スタンダード・イベント・ステータス・レジスタ (SESR)

- bit 0 : OPC : 動作完了  
本機が処理を完了し、新しいコマンドを受け入れる状態であることを示します。  
このビットは動作完了コマンド (\*OPC) の応答として発生します。
- bit 1 : RQC : リクエスト・コントロール  
本機においては常に 0 です。
- bit 2 : QYE : クエリ・エラー  
相手機器が本機のデータ出力待ち行列からデータを読む際にエラーが発生したことを示します。  
原因は、出力待ち行列が空の時に読み出そうとしたが、オーバーフローしている場合です。
- bit 3 : DDE : 機器定義エラー  
本機が電源投入された場合、プログラムROMのサムチェックとシステムワークRAMのリードライトチェックを行い、エラーが発生した場合 1 になります。
- bit 4 : EXE : 実行エラー  
本機がコマンド実行時にエラーが発生したことを示します。  
原因は、本機がサポートしていないコマンドを受け取ったか、現在の本機の状態では実行不可能なコマンドを受け取ったことによります。
- bit 5 : CME : コマンド・エラー  
本機が受け取ったコマンドがフォーマットに適合していない場合に発生します。
- bit 6 : URQ : ユーザ・リクエスト  
本機においては常に 0 です。
- bit 7 : PON : パワー・オン  
スタンダード・イベント・ステータス・レジスタを最後にクエリして以降、本機の電源を入れなおしたことを示します。



[ - 3 ] D A M - 7 0 2 G P ・外部・ステータス・レジスタ

- bit 0 : S T 1 : 外部端末側ステータス入力  $\overline{S T 1}$  の状態  
外部端末側ステータス入力信号 S T 1 の状態を表します。
- bit 1 : S T 2 : 外部端末側ステータス入力  $\overline{S T 2}$  の状態  
外部端末側ステータス入力信号 S T 2 の状態を表します。
- bit 2 : S T 3 : 外部端末側ステータス入力  $\overline{S T 3}$  の状態  
外部端末側ステータス入力信号 S T 3 の状態を表します。
- bit 3 : S T 4 : 外部端末側ステータス入力  $\overline{S T 4}$  の状態  
外部端末側ステータス入力信号 S T 4 の状態を表します。
- bit 4 : S T 5 : 外部端末側ステータス入力  $\overline{S T 5}$  の状態  
外部端末側ステータス入力信号 S T 5 の状態を表します。
- bit 5 : S T 6 : 外部端末側ステータス入力  $\overline{S T 6}$  の状態  
外部端末側ステータス入力信号 S T 6 の状態を表します。
- bit 6 : R E Q : 外部端末側リクエスト入力  $\overline{R E Q}$  の状態  
外部端末側リクエスト入力信号 R E Q の状態を表します。
- bit 7 : S T 8 : 外部端末側ステータス入力  $\overline{S T 8}$  の状態  
外部端末側ステータス入力信号 S T 8 の状態を表します。



外部・ステータス・条件・レジスタ

:STATUS:EXTERNAL:CONDITION? でクエリ可

外部・ステータス・トランジション・レジスタ

:STATUS:EXTERNAL:TRANSITION? でクエリ可

:STATUS:EXTERNAL:TRANSITION 数値 ( 0 ~ 6 3、 1 2 8 ~ 1 9 1 ) で設定可

外部・ステータス・イベント・レジスタ

:STATUS:EXTERNAL:EVENT? でクエリ可

外部・ステータス・イネーブル・レジスタ

:STATUS:EXTERNAL:ENABLE? でクエリ可

:STATUS:EXTERNAL:ENABLE 数値 ( 0 ~ 2 5 5 ) で設定可

(以上のコマンドの説明は、本書 [ - 5 ] をご参照ください)

#### [ - 4 ] ステータス・レジスタの初期値

本機の電源を投入した場合、背面のディップスイッチの状態を変更した場合、ステータス報告システムの各レジスタの初期値は下記のように設定されます。

ステータス・バイト・レジスタ	bit7	RQS/MSS	bit5	bit4	bit3	bit2	bit1	bit0
	0	0	0	0	0	0	0	0
サービス・リクエスト・イネーブル・レジスタ	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	0	0	0	0	0	0	0	1
スタンダード・イベント・ステータス・レジスタ	PON	URQ	CME	EXE	DDE	QYE	RQC	OPC
	1	0	0	0	0	0	0	0
スタンダード・イベント・イネーブル・レジスタ	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	0	0	0	0	0	0	0	0
外部・ステータス・条件・レジスタ	ST8	REQ	ST6	ST5	ST4	ST3	ST2	ST1
	0	0	0	0	0	0	0	0
外部・ステータス・トランジション・レジスタ	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	0	0	0	0	0	0	0	0
外部・ステータス・イベント・レジスタ	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	0	0	0	0	0	0	0	0
外部・ステータス・イネーブル・レジスタ	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	0	1	0	0	0	0	0	0

これは、本機の端末側の $\overline{\text{REQ}}$ 信号がHighからLowに変化するとGPIB上のSRQラインがON（アクティブ）になるように設定されています。

## 【 】SCMCコマンド for DAM-702GPB/C

### コマンド

当SCMCコマンドはI-EEE488.2-1992規格を基に階層構造になっています。  
設定データのほとんどはクエリ(設定値の確認読み出し)する事ができます。

### 数値パラメータ

数値パラメータはASCII文字による10進表記を基本として、16進、8進、2進表記も使用できます。  
10進表記では、符号、小数点、指数部付き表記を使用できませんが、  
16, 18, 2進表記では整数のみを使用します。  
また、2進数の特別な扱いとして論理値(LON, LOFF)を使用することができます。

### ディスクリットパラメータ

数値では表現できない設定データ、または未知の数値データを表すパラメータです。  
例えば、トリガ源として外部トリガ入力を指定(選択)する場合は、EXTERNAL  
例えば、信号の立ち上がりを指定(選択)する場合は、POSITIVE  
例えば、アンプのゲインを最大に取りたい場合は、MAX  
の様に使います。

### ブロックパラメータ

大量のデータを送受するための特別なフォーマットです。  
この中でも、データ個数があらかじめ特定できる場合と、できない場合があります。

確定長・データ・ストリング・フォーマット                    <DAS0>,<DAS1>,<DAS2>,<                    >,<DASm>

<DAS0>: 後に続くデータの個数を表します。  
数値の表現は10進、2進、8進、16進のいずれも使用できます。  
<DAS1>~<DASm>: データです。10進、2進、8進、16進のいずれの表現も使用できます。  
各<DASm>は、で区切られています。

確定長・データ・バイナリ・フォーマット                    #nm<DAB1><DAB2><                    ><DABm>

n: 1桁のASCII数値、データ・バイトのバイト数mの桁数を表します。  
このnは、10進数で表現します。  
m: n桁のASCII数値、データ・バイトのバイト数を表します。  
この後に続く、<DAB1>から<DABm>までの個数をバイト単位で表します。  
このmは、10進数で表現します。  
<DAB1>~<DABm>: データのバイナリ・コードです。  
各<DABm>は、で区切られていません。

不確定長・データ・ストリング・フォーマット                    0,<DAS1>,<DAS2>,<                    >,<DASm>

0: 不確定長ストリングを表す、ASCII文字です。  
<DAS1>~<DASm>: データです。  
数値の表現は10進、2進、8進、16進のいずれも使用できます。  
各<DASm>は、で区切られています。

不確定長・データ・バイナリ・フォーマット                    #0<DAB1><DAB2><                    ><DABm>

#0                    : 不確定長バイナリを表す、ASCII文字です。  
<DAB1>~<DABm>: データのバイナリ・コードです。  
各<DABm>はで区切られていません。  
デリミタはバイナリデータと区別できるよう、E0Iを含んでいなければなりません。

### デリミタ(ターミネータ)

すべてのコマンドメッセージはデリミタで終了させてください。  
本機からの応答メッセージもすべてデリミタで終了します。(本書[                    - 5 ]参照)

[ - 1 ] 入力端からの入力コマンド

INPUTコマンドセット

コマンド	パラメータ	備考	初期値
:INPut [:DATA]?	ビット名称 (BIT0~7), データ数 (0, 1~1000000)		
:FORMat	バイト名称 (BYTE0), データ数 (0, 1~1000000)	データ形式の指定	DECIMAL
:FORMat?	データ形式	データ形式の問い合わせ	

ビット名称: BIT0, BIT1, BIT2, BIT3, BIT4, BIT5, BIT6, BIT7  
 このパラメータが BIT の場合は BIT0 とみなします。  
 BIT0, BIT1, BIT2, ..., BIT7の代わりにTD1, TD2, TD3, TD4, TD5, TD6, TD7, TD8も使用できます。  
 TD1はBIT0と, TD2はBIT1と, TD3はBIT2と, ..., TD8はBIT7と同じとみなします。  
 ただし, TD1はTD1と同じ, とはみなしません。

バイト名称: BYTE0  
 このパラメータがBYTEまたはTDの場合はBYTE0とみなします。

データ形式:  
 A S C I I 文字数値の 2 進数を指定する場合は, BINary と記述します。  
 A S C I I 文字数値の 8 進数を指定する場合は, OCTal と記述します。  
 A S C I I 文字数値の 1 0 進数を指定する場合は, DECimal と記述します。  
 A S C I I 文字数値の 1 6 進数を指定する場合は, HEX と記述します。  
 A S C I I 文字数値の論理を指定する場合は, LOGical と記述します。  
 バイナリコードを指定する場合は, CODE と記述します。

[ - 1 - 1 ]

書式 : INPUT[:DATA]? ビット名称[, データ数]  
 : INPUT[:DATA]? バイト名称[, データ数]

説明 ビット名称またはバイト名称で指定する入力端の信号を、データ数で指定する回数だけ入力し、  
 応答メッセージを作成することを指示します。[]の部分は省略可能です。  
 データ数の指定を省略した場合は 1 とみなされます。  
 応答メッセージのフォーマットは「:INPUT:FORMAT データ形式」で指定されたフォーマットです。  
 「:INPUT[:DATA] バイト名称」の場合で、「:INPUT:FORMAT」で「論理」を指定してあった場合、  
 「BINARY」の表現で応答データを返送します。

データ数:  
 指定できるデータ数は 0 ~ 1 0 0 0 0 0 0 の範囲です。  
 0 を指定すると、データの数を限定せず、EOD信号がアクティブになるまで取り込むこととなります。

応答 このコマンドの後、トーカに指定されると指定された入力端の信号を入力し、  
 指定されたフォーマットで応答メッセージを返送します。  
 この時、本機の端末側信号「READY」はLowでなければなりません。(Highの場合はLowになるのを  
 待ってから入力端の信号を入力します。この間、G P I B 側のハンドシェイクは停止します。)  
 指定されたデータ数を満たすか、本機の端末側信号「EOD」がアクティブになると、  
 その時の入力信号データを G P I B に送出し、終了します。

応答メッセージのフォーマット

不確定長・データ・ストリング・フォーマット 0, <DAS1>, <DAS2>, < >, <DASm>

0: 不確定長ストリングを表す, A S C I I 文字です。  
 <DAS1> ~ <DASm>: 指定されたデータ形式で表した数値のデータです。

「:INPUT:DATA? バイト名称, データ数」コマンドに対する応答の場合、

データの値は 0 ~ 2 5 5 の範囲です。  
 指定データ形式が 2 進数の場合は、例えば#B11011となっています。  
 1 0 進数の場合は、例えば27となっています。  
 1 6 進数の場合は、例えば#H1Bとなっています。  
 8 進数の場合は、例えば#Q27となっています。  
 論理の場合は、例えば#B11011となっています。

「:INPUT:DATA? ビット名称, データ数」コマンドに対する応答の場合、

データの値の範囲は 0 または 1 です。  
 指定データ形式が 2 進数の場合は、#B0または#B1となっています。  
 1 0 進数の場合は、0または1となっています。  
 1 6 進数の場合は、#H0または#H1となっています。  
 8 進数の場合は、#Q0または#Q1となっています。  
 論理の場合は、LOFFまたはLONとなっています。

各<DASm>は、で区切られています。

不確定長・データ・バイナリ・フォーマット #0<DAB1><DAB2>< ><DABm>

#0 : 不確定長バイナリを表す、ASCII文字です。  
<DAB1> ~ <DABm> : 指定入力端から入力したデータのバイナリ・コードです。  
指定入力端の種類によってデータ構造が違います。  
指定入力端名称による<DAB>の内容の例を以下に示します。

「:INPUT:DATA? バイト名称」コマンドに対する応答の場合、

<DAB>はTD1 ~ TD8から入力したデータの8ビット・バイナリ・コードです。  
TD1がLSB、TD8がMSBに対応しています。  
例: TD1 ~ TD8の入力端の信号状態が下表のような場合、

TD8	TD7	TD6	TD5	TD4	TD3	TD2	TD1
Low	High	High	Low	Low	Low	High	Low

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	1	1	0	0	0	1	0

<DAB>は左表のようになります。

「:INPUT:DATA? ビット名称」コマンドに対する応答の場合、

<DAB>はTD1 ~ TD8の中の指定された1ビットのデータを8ビット・バイナリ・コードで表しています。  
指定されたビットのデータがLSBになり、他の7ビットは0です。  
例: TD1 ~ TD8の入力端の信号状態が下表のような場合、

TD8	TD7	TD6	TD5	TD4	TD3	TD2	TD1
Low	High	High	Low	Low	High	High	Low

→

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	0	0	0	1

指定されたビットがBIT2(TD3)であれば<DAB>は左表のようになります。

指定ビットBIT2(TD3)のデータが<DAB>のLSB (bit0)に反映され、他のビット(bit1~bit7)は0が挿入されます。

各<DABm>は、で区切られていません。バイナリデータと区別できるよう、最後は必ずディップスイッチで選択したEOIを含むデリミタで終了します。

どちらのフォーマットもデータ (<DAS>または<DAB>) の個数は、指定されたデータ数が端末側信号EODがアクティブになるまでの数のいずれかです。

「 - 1 - 2 」

書式 :INPUT:FORMAT データ形式

説明 「:INPUT:DATA ビット名称/バイト名称,データ数」コマンドに対する応答メッセージのフォーマットを指定します。

データ形式:

ASCII文字数値の2進数を指定する場合は、BINary と記述します。  
ASCII文字数値の8進数を指定する場合は、OCTal と記述します。  
ASCII文字数値の10進数を指定する場合は、DECimal と記述します。  
ASCII文字数値の16進数を指定する場合は、HEX と記述します。  
ASCII文字数値の論理を指定する場合は、LOGical と記述します。  
バイナリコードを指定する場合は、CODE と記述します。

応答 このコマンドに対する応答はありません。

「 - 1 - 3 」

書式 : INPUT:FORMAT?

説明 「:INPUT:DATA ビット名称/バイト名称,データ数」コマンドに対する応答メッセージのデータ形式の指定状況を問い合わせます。

応答 このコマンドの後、トークンに指定されると下記のいずれかの応答メッセージを返送します。

BINARY  
OCTAL  
DECIMAL  
HEX  
LOGICAL  
CODE



[ - 2 ] 出力端への出力コマンド

OUTPUTコマンドセット

コマンド	パラメータ	備考
:OUTput	チャンネル名称(CH0,1),出力データ	
:OUTput?	チャンネル名称(CH0,1),データ形式	

チャンネル名称 : CH0,CH1

CH0はD/A変換・電圧出力のチャンネル0、CH1はD/A変換・電圧出力のチャンネル1、を指します。  
このパラメータがDA0の場合はCH0とみなし、DA1の場合はCH1とみなします。  
このパラメータがDAの場合はDA0とみなします。

データ形式 : 2進数を指定する場合は、BINary と記述します。  
8進数を指定する場合は、OCTal と記述します。  
10進数を指定する場合は、DECimal と記述します。  
16進数を指定する場合は、HEX と記述します。

「 - 2 - 1 」

書式 : OUTPUT チャンネル名称,出力データ

説明 チャンネル名称で指定するD/A出力CHへ出力データによる電圧を出力させます。

出力データ :

出力データの値は10進数、16進数、8進数、2進数のいずれかで表現したASCII文字で指定します。

基数ヘッダが付加されないと10進数とみなされます。  
基数を2進数とする場合は、例えば#B101などと記述します  
8進数とする場合は、例えば#Q107などと記述します。  
10進数とする場合は、例えば245などと記述します。  
16進数とする場合は、例えば#HE1と記述します。

ただし、「:CONFIGURE:OUTPUT」コマンドで、データ単位を「mV」(V11)、「V」(V00)に設定している場合は、10進数のみが使用でき、2進数、8進数、16進数は「実行エラー」になります。

本機は、このコマンドを受信すると、受信データの値を指定出力端に出力することにより、D/A変換・電圧出力端に電圧を出力させます。  
出力データの値の範囲は下表に示す最小単位で丸められ、結果は下表に示す範囲内であればなりません。(「データ単位」については[ - 7 - 1 ]を参照)

データ単位の記号がC12(設定コード)の場合

出力電圧範囲	出力データの値の範囲	出力データの値の最小単位
0V ~ +10V	0 ~ 4095	1
0V ~ +5V	0 ~ 4095	1
-10V ~ +10V	0 ~ 4095	1
-5V ~ +5V	0 ~ 4095	1
-10V ~ 0V	0 ~ 4095	1
-5V ~ 0V	0 ~ 4095	1

データ単位の記号がV11(mV)の場合

出力電圧範囲	出力データの値の範囲	出力データの値の最小単位
0V ~ +10V	0 ~ +10237.5	2.5
0V ~ +5V	0 ~ +5118.75	1.25
-10V ~ +10V	-10240 ~ +10235	5
-5V ~ +5V	-5120.0 ~ +5117.5	2.5
-10V ~ 0V	-10237.5 ~ 0	2.5
-5V ~ 0V	-5118.75 ~ 0	1.25

データ単位の記号がV00 ( V ) の場合

出力電圧範囲	出力データの値の範囲	出力データの値の最小単位
0 V ~ + 1 0 V	0 ~ + 1 0 . 2 3 7 5	0 . 0 0 2 5
0 V ~ + 5 V	0 ~ + 5 . 1 1 8 7 5	0 . 0 0 1 2 5
- 1 0 V ~ + 1 0 V	- 1 0 . 2 4 0 ~ + 1 0 . 2 3 5	0 . 0 0 5
- 5 V ~ + 5 V	- 5 . 1 2 0 0 ~ + 5 . 1 1 7 5	0 . 0 0 2 5
- 1 0 V ~ 0 V	- 1 0 . 2 3 7 5 ~ 0	0 . 0 0 2 5
- 5 V ~ 0 V	- 5 . 1 1 8 7 5 ~ 0	0 . 0 0 1 2 5

応答 このコマンドに対する応答はありません。

「 - 2 - 2 」

書式 :OUTPUT? チャネル名称[,データ形式]

説明 チャネル名称で指定する D / A 変換・電圧出力端の電圧をモニタし、基数で指定する表現で、応答メッセージを作成させます。  
[]の部分は省略可能です。データ形式の指定を省略した場合は10進数とみなされます。

データ形式:

- 2進数を指定する場合は、BINary と記述します。
- 8進数を指定する場合は、OCTal と記述します。
- 10進数を指定する場合は、DECimal と記述します。
- 16進数を指定する場合は、HEX と記述します。

ただし、「:CONFIGURE:OUTPUT」コマンドで、データ単位を「mV」(V11)、「V」(V00)に設定している場合は、10進数のみが使用でき、2進数、8進数、16進数は「実行エラー」になります。

応答 このコマンドの後、トーカーに指定されると指定されたD/A変換・電圧出力端の電圧をモニタし、指定されたデータ形式の数値で応答メッセージを返送します。

応答メッセージのフォーマットは下記のとおりです。

数値

数値は指定された基数ヘッダが付加されたASCII文字列のデータがひとつです。数値の単位は「:CONFIGURE:OUTPUT 電圧範囲,データ単位」で指定した単位です。  
データ形式が2進数の場合は、例えば#B1000001となっています。  
10進数の場合は、例えば5.123となっています。  
16進数の場合は、例えば#H2A1となっています。  
8進数の場合は、例えば#Q741となっています。

[ - 3 ] バッファメモリ・コマンド

MEMORYコマンドセット

コマンド	パラメータ	備考	初期値
:MEMory :ASSign :ASSign?	ブロック番号(0~1),ワード数 ブロック番号(0~1)	メモリ領域容量を指定確保する メモリ領域の情報の問い合わせ 領域容量,使用容量,空容量を得る	確保されていない
:WRITe :INITialize [:NEXT]	ブロック番号(0~1) ブロック番号(0~1),データ列	指定領域の書込ポインタを初期化 書込ポインタから書込み、 書込ポインタを次へ移す。	
:READ :INITialize [:NEXT]?	ブロック番号(0~1) ブロック番号(0~1),ワード数	指定領域の読出ポインタを初期化 読出ポインタから読出し、 読出ポインタを次へ移す。	
:FORMat :FORMat? :MEMory?	ブロック番号(0~1),データ形式 ブロック番号(0~1)	読出データ形式を指定する 読出データ形式の問い合わせ メモリの情報の問い合わせ 総領域容量,総空容量を得る	DECIMAL

プレイ動作が、STANDBY状態にある場合は同一ブロック番号のメモリ領域に対して以下のことを行うことができません。

:MEMORY:ASSIGN                      ブロック番号,ワード数

プレイ動作が、RUNNING状態にある場合は同一ブロック番号のメモリ領域に対して以下のことを行うことができません。

:MEMORY:ASSIGN                      ブロック番号,ワード数  
:MEMORY:WRITE:INITIALIZE        ブロック番号  
:MEMORY:WRITE:NEXT                ブロック番号,データ列  
:MEMORY:READ:INITIALIZE         ブロック番号  
:MEMORY:READ:NEXT?                ブロック番号,ワード数

ブロック番号 : 0,1

確保した領域はプレイ動作で使用します。

「 - 3 - 1 」

書式 :MEMORY:ASSIGN ブロック番号,ワード数

説明 ブロック番号で指定する領域の容量をワード単位の数で指定確保します。  
読出ポインタと書込ポインタは、この領域の先頭に初期化されます。

ワード数 : 0 または、1 以上、メモリの総空容量以内  
ブロック番号で指定する領域のメモリ領域容量をワード単位で指定します。  
0 を指定した場合は、指定ブロック番号の領域を解放します。

領域の容量を変えたい場合は、一度、「:MEMORY:ASSIGN ブロック番号,0」で領域を開放してから新たな容量で確保します。この時、この領域のデータは消失します。また、「:PLAY:ASSIGN」で割り当てていた同じブロック番号のメモリ領域の割り当ても解放されます。  
確保可能なメモリの総空容量は「:MEMORY?」コマンドで調べることができます。

応答 このコマンドに対する応答はありません。

プレイ動作が、STANDBY状態やRUNNING状態にある場合は  
同一ブロック番号のメモリ領域に対してこのコマンドを受信すると「実行エラー」になります。

このコマンドで指定するブロック番号でメモリの領域がすでに定義確保されている場合は、データ数が0なら領域の解放を行います。0でない場合は「実行エラー」になります。

「 - 3 - 2 」

書式 :MEMORY:ASSIGN? ブロック番号

説明 ブロック番号で指定する領域の情報を問い合わせます。

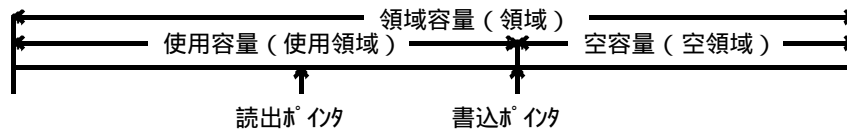
応答 このコマンドの後、トーカーに指定されると下記の応答メッセージを返送します。

領域容量, 使用容量, 空容量

領域容量: 「:MEMORY:ASSIGN ブロック番号,ワード数」で確保されている領域の容量をワード単位で表しています。

使用容量: 「:MEMORY:WRITE:NEXT ブロック番号,データ列」で書き込まれたデータの数をワード単位で表しています。

空容量: 領域容量から使用容量を差し引いた数をワード単位で表しています。



「 - 3 - 3 」

書式 :MEMORY:WRITE:INITIALIZE ブロック番号

説明 ブロック番号で指定する領域への書込ポインタを初期化します。また、今までに書かれていたデータがあればこれを破棄し、読出ポインタも初期化します。

応答 このコマンドに対する応答はありません。プレイ動作が、RUNNING状態にある場合は、同一ブロック番号のメモリ領域に対してこのコマンドを受信すると「実行エラー」になります。

「 - 3 - 4 」

書式 :MEMORY:WRITE:NEXT ブロック番号,データ列

説明 ブロック番号で指定する領域へデータを連続的に書き込みます。この動作の後、書込ポインタは最終書込データの格納された次をポイントします。読出ポインタは変化しません。データ列のフォーマットは下記のどちらの場合でも使用できます。

確定長・データ・ストリング・フォーマット <DAS0>,<DAS1>,<DAS2>,< >,<DASm>

<DAS0>: 書込データの個数を表します。数値の表現は10進、2進、8進、16進のいずれも使用できます。

<DAS1>~<DASm>: 書き込むべきデータです。10進、2進、8進、16進のいずれの表現も使用できます。各<DASm>は、で区切ってください。ただし、「:CONFIGURE:MEMORY」コマンドで、データ単位を「mV」(V11)、「V」(V00)に設定している場合は、10進数のみが使用でき、2進数、8進数、16進数は「実行エラー」になります。

確定長・データ・バイナリ・フォーマット #nm<DAB1><DAB2>< ><DABm>

ただし、「:CONFIGURE:MEMORY」コマンドで、データ単位を「mV」(V11)、「V」(V00)に設定している場合は、このフォーマットは使用できず、「実行エラー」になります。

n: 1桁のASCII数値、データ・バイトのバイト数mの桁数を表します。このnは、10進数で表現します。

m: n桁のASCII数値、データ・バイトのバイト数を表します。この後に続く、<DAB1>から<DABm>までの個数をバイト単位で表します。このmは、10進数で表現し、偶数でなければなりません。基数の場合は「実行エラー」になります。

<DABm>: 端末側へ出力させるデータで、バイナリコードで入れて下さい。(0x0000)~(0x0FFF)の範囲の出力電圧の設定コードを上位バイト, 下位バイトに分けて、上位バイトを先に入れて下さい。

例: (0x0234)と(0x0678)の2ワードを書き込む場合は下記ようになります。  
#14<0x02><0x34><0x06><0x78>

どちらのフォーマットの場合でも、確保されたメモリ領域の領域容量を越えたら途中までで強制的に終了します。

応答 このコマンドに対する応答はありません。プレイ動作が、RUNNING状態にある場合は、同一ブロック番号のメモリ領域に対してこのコマンドを受信すると「実行エラー」になります。

「 - 3 - 5 」

書式 :MEMORY:READ:INITIALIZE プ ロック番号

説明 プ ロック番号で指定する領域からの読出ポ イントを初期化します。  
書込ポ イントは変化しません。

応答 このコマンドに対する応答はありません。  
プレイ動作が、RUNNING状態にある場合は、同一プ ロック番号のメモリ領域に対しての  
このコマンドを受信すると「実行エラー」になります。

「 - 3 - 6 」

書式 :MEMORY:READ:NEXT? プ ロック番号,ワード数

説明 プ ロック番号で指定する領域からデータを連続的に読み出します。  
この動作の後、読出ポ イントは最後に読み出したデータの格納されていた次をポイントします。  
書込ポ イントは変化しません。

ワード数 : 0 または、1 ~ 1 0 0 0 0 0 0

プ ロック番号で指定する領域から読み出したいデータの数をワード単位で指定します。  
指定したプ ロック番号の領域に存在する未読み出しデータよりおおきな数を指定しても  
エラーにはならず、未読み出しデータ全部を正常に読み出す事ができます。  
0 を指定した場合は、指定したプ ロック番号の領域の残りデータ全部を読み出す事になります。

応答 このコマンドの後、トーカーに指定されると「:MEMORY:READ:FORMAT プ ロック番号,データ形式」で  
指定されているフォーマットに従って下記のいずれかで返送します。  
読み出すべきデータが無い場合はデータの個数を 0 として返送します。  
また、指定されたメモリ領域が「:MEMORY:ASSIGN」コマンドで定義確保されていない場合も同様です。

データ形式を BINary、OCTal、DECimal、HEX と指定してある場合は以下のようにになります。

確定長・データ・ストリング・フォーマット <DAS0>,<DAS1>,<DAS2>,< >,<DASm>

<DAS0> : 読み出すデータの個数を表します。数値の表現は 1 0 進整数です。  
「:MEMORY:READ:NEXT プ ロック番号,ワード数」で指定したワード数、  
またはメモリ領域に入っていたデータの数のどちらか小さい方です。  
<DAS1> ~ <DASm> : 読み出したデータです。1 0 進整数で表現しています。  
各<DASm>は、で区切られています。

データ形式を CODE と指定してある場合は以下のようにになります。

確定長・データ・バイナリ・フォーマット #nm<DAB1><DAB2>< ><DABm>

n : 1 桁の A S C I I 数値、データ・バイトのバイト数mの桁数を表します。  
このnは、1 0 進数で表現します。  
m : n 桁の A S C I I 数値、データ・バイトのバイト数を表します。この後に続く、  
<DAB1>から<DABm>までの個数をバイト単位で表します。  
「:MEMORY:READ:NEXT プ ロック番号,ワード数」で指定したワード数、  
またはメモリ領域に入っていたデータの数のどちらか小さい方の 2 倍です。  
このmは、1 0 進数で表現します。  
<DAB1> ~ <DABm> : 各<DABm>は、で区切られていません。  
ワード単位のデータがバイト単位に分割され、上位バイトが先に、  
下位バイトが後に入っています。

プレイ動作が、RUNNING状態にある場合は同一プ ロック番号のメモリ領域に対しての  
このコマンドを受信すると「実行エラー」になります。

「 - 3 - 7 」

書式 :MEMORY:READ:FORMAT ブロック番号,データ形式

説明 「:MEMORY:READ:NEXT? ブロック番号,ワード数」コマンドに対する応答メッセージのデータ形式を指定します。

データ形式:

ASCII文字数値の2進数を指定する場合は、BINary と記述します。  
ASCII文字数値の8進数を指定する場合は、OCTal と記述します。  
ASCII文字数値の10進数を指定する場合は、DECimal と記述します。  
ASCII文字数値の16進数を指定する場合は、HEX と記述します。  
バイナリーコードを指定する場合は、CODE と記述します。  
「論理」は指定できません。

応答 このコマンドに対する応答はありません。

「CONFIGURE:MEMORY」コマンドで、データ単位を「mV」(V11)、「V」(V00)に設定している場合は、10進数のみが使用でき、2進数、8進数、16進数、バイナリーコードは「実行エラー」になります。

「 - 3 - 8 」

書式 :MEMORY:READ:FORMAT? ブロック番号

説明 「:MEMORY:READ:NEXT ブロック番号,ワード数」コマンドに対する応答メッセージのフォーマットの設定選択状況を問い合わせます。

応答 このコマンドの後、トークンに指定されると下記のいずれかの応答メッセージを返送します。

BINARY  
OCTAL  
DECIMAL  
HEX  
CODE

「 - 3 - 9 」

書式 :MEMORY?

説明 メモリの情報を問い合わせます。

応答 このコマンドの後、トークンに指定されると下記の応答メッセージを返送します。

総領域容量,総空容量

総領域容量:「:MEMORY:ASSIGN ブロック番号,ワード数」で確保されているメモリ領域の合計をワードの単位で表しています。

総空容量:MEMORYコマンドシステムで使用できる残りの容量をワードの単位で表しています。  
「:MEMORY:ASSIGN ブロック番号,ワード数」コマンドで確保されているメモリ領域が無い(総領域容量 = 0ワードの場合、262144ワードです。

本機のメモリ管理方法

本機において、メモリ領域は1024ワード単位で管理しています。

「:MEMORY:ASSIGN」コマンドで領域を確保すると総空容量は1024ワード単位で減ります。

例えば、「:MEMORY:ASSIGN 0,10」、「:MEMORY:ASSIGN 1,20」とするとブロック0に10ワード、ブロック1に20ワードが確保され、総空容量は2048ワード減ります。

[ - 4 ] バッファリングされたデータの出力コマンド

PLAYコマンドセット

コマンド	パラメータ	備考	初期値
:PLAY			
:CLOCK			
:LEVel	チャンネル名称 (CH0 ~ 1), クロックソースのレベル値	レベル値の設定 設定値は1 ~ 10000000 (m秒) 精度は± 100μ秒	1 (m秒)
:LEVel?	チャンネル名称 (CH0 ~ 1)	レベル値の問い合わせ 応答は1 ~ 10000000 (m秒)	
:REPeat	チャンネル名称 (CH0 ~ 1), 回数 (0, 1 ~ 1000000)	繰り返し回数の設定 0を指定すると無限	1
:REPeat?	チャンネル名称 (CH0 ~ 1)	繰り返し回数の問い合わせ 応答は0 ~ 1 0 0 0 0 0 0	
:ASSign	チャンネル名称 (CH0 ~ 1), ブロック番号, データ数	プレイ入出力の割り当て	割り当てなし
:ASSign?	チャンネル名称 (CH0 ~ 1)	プレイ入出力の問い合わせ	
[:STARt]	チャンネル名称 (CH0 ~ 1), 指令	指令は下記のいずれか。 ENable, DISable	
:STATe?	チャンネル名称 (CH0 ~ 1)	PLAY動作の状態を返す。 下記のいずれか。 IDLE, STANDBY, RUNNING	IDLE

プレイ動作が STANDBY 状態にある場合は同一ブロック番号のメモリ領域に対して以下のことを行うことができません。

:MEMORY:ASSIGN                      ブロック番号, ワード数

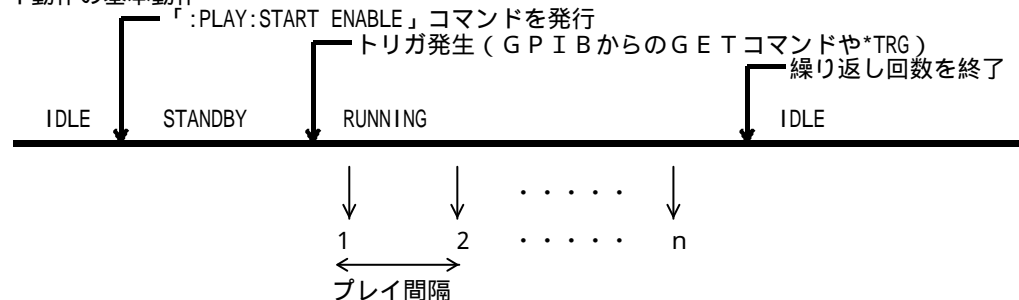
プレイ動作が RUNNING 状態にある場合は同一ブロック番号のメモリ領域に対して以下のことを行うことができません。

:MEMORY:ASSIGN                      ブロック番号, ワード数  
:MEMORY:WRITE:INITIALIZE        ブロック番号  
:MEMORY:WRITE:NEXT                ブロック番号, データ列  
:MEMORY:READ:INITIALIZE         ブロック番号  
:MEMORY:READ:NEXT?                ブロック番号, ワード数

チャンネル名称: CH0, CH1

CH0はD/A変換・電圧出力のチャンネル0、CH1はD/A変換・電圧出力のチャンネル1、を指します。  
このパラメータがDAOの場合はCH0とみなし、DA1の場合はCH1とみなします。  
このパラメータがDAの場合はDAOとみなします。

プレイ動作の基本動作



プレイ間隔は「:PLAY:CLOCK:LEVEL」でのレベル値で指定したタイマー時間の間隔によります。  
プレイ出力するデータの数nは原則として  
$$n = \text{「:PLAY:ASSIGN」で指定したデータ数} \times \text{「:PLAY:REPEAT」で指定した回数}$$
で表されます。

「 - 4 - 1 」

書式 :PLAY:CLOCK:LEVEL ファン礼名称, クロックのレベル値

説明 ファン礼名称で指定する出力端への信号をプレイするためのクロック源のタイマー値を指定します。

クロックのレベル値 : 10 ~ 10000000

出力端へ信号を出力する間隔を指定します。

10 m秒 ~ 10000000 m秒の範囲の1 m秒の整数倍の値で設定します。

範囲以外は「実行エラー」になり、以前の設定値が残ります。

「:PLAY:START ENABLE」コマンドの後、トリガが発生すると、ここで指定された間隔でメモリ領域のデータを出力端へ出力します。

応答 このコマンドに対する応答はありません。  
このコマンドで指定するファン礼名称のプレイ動作がRUNNING状態の時にこのコマンドを受信すると「実行エラー」になります。

「 - 4 - 2 」

書式 :PLAY:CLOCK:LEVEL? ファン礼名称

説明 ファン礼名称で指定する入力端の信号をプレイするためのクロック源のタイマー値の設定選択状況を問い合わせます。

応答 このコマンドの後、トーカーに指定されると下記の応答メッセージを返送します。  
数値は設定されているクロックのレベル値で、プレイ間隔を示しています。  
数値の範囲は1 m秒 ~ 10000000 m秒の範囲の1 m秒の整数倍の値です。

数値

「 - 4 - 3 」

書式 :PLAY:REPEAT ファン礼名称, 回数

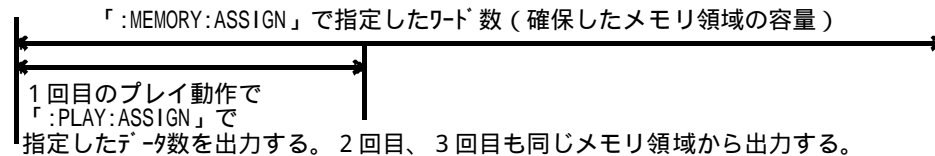
説明 ファン礼名称で指定する出力端へ信号をプレイする繰り返し回数を指定します。

回数 : 0, 1, 2, 3, ..., 1000000

1 ~ 1000000を指定すると一回のプレイ動作を指定した回数、繰り返します。

0を指定すると、\*ABORT, または\*RSTを受信するまで繰り返します。

下図に「:PLAY:REPEAT ファン礼名称, 3」を指定した場合のメモリ領域の使用状況を示します。



この時、「:MEMORY:WRITE:NEXT」で書き込んだデータの数、「:PLAY:ASSIGN」で指定したデータの数より少ない場合、出力したデータの数、「:PLAY:ASSIGN」で指定したデータの数に満たなくてもこの回を終了し、次の回に移ります。

応答 このコマンドに対する応答はありません。  
このコマンドで指定するファン礼名称のプレイ動作がRUNNING状態の時にこのコマンドを受信すると「実行エラー」になります。

「 - 4 - 4 」

書式 :PLAY:REPEAT? ファン礼名称

説明 ファン礼名称で指定する出力端へ信号をプレイする繰り返し回数の設定値を問い合わせます。

応答 このコマンドの後、トーカーに指定されると設定されている回数を10進整数で返送します。



「 - 4 - 5 」

書式 :PLAY:ASSIGN? チャンネル名称, ブロック番号, データ数

説明 チャンネル名称で指定する出力端へ信号をプレイするデータが格納されているメモリ領域の割り当てを行います。

ブロック番号: 0, 1

あらかじめ、「MEMORY:ASSIGN ブロック番号,ワード数」コマンドで、メモリ領域とその容量を定義確保しておかなければなりません。

データ数: 1 以上、メモリ領域容量以内

一回のプレイ動作で出力するデータの数を指定します。

0を指定すると、プレイデータ源とデータ出力先の割り当てを解除（解放）します。

応答 このコマンドに対する応答はありません。

このコマンドのブロック番号で指定するメモリ領域の領域容量が「MEMORY:ASSIGN ブロック番号,ワード数」コマンドのワード数で、定義確保されていない場合は「実行エラー」になります。

このコマンドで指定するチャンネル名称のプレイ動作がSTANDBY状態やRUNNING状態の時にこのコマンドを受信すると「実行エラー」になります。

このコマンドで指定するチャンネル名称に、ブロック番号で指定するメモリ領域とその容量をすでに定義確保している場合は、ワード数が0なら割り当ての解除を行います。  
0でない場合は「実行エラー」になります。

このコマンドで指定するチャンネル名称に、ブロック番号で指定する他のメモリ領域とその容量をすでに定義確保している場合は、「実行エラー」になります。

このコマンドで指定するチャンネル名称に対する「:CONFIGURE:OUTPUT」コマンドによる電圧範囲とこのコマンドで指定するブロック番号に対する「:CONFIGURE:MEMORY」コマンドによる電圧範囲が等しくない場合は「実行エラー」になります。

「 - 4 - 6 」

書式 :PLAY:ASSIGN? チャンネル名称

説明 チャンネル名称で指定する出力端へ信号をプレイするデータが格納されているメモリ領域の割り当て状況を問い合わせます。

応答 このコマンドの後、トーカーに指定されると下記の応答メッセージを返送します。

ブロック番号, データ数

応答メッセージのブロック番号が - 1、データ数が 0 の場合は、指定されたチャンネル名称と指定されたブロック番号のメモリ領域が結び付けられていない（割り当てられていない）ことを示します。

「 - 4 - 7 」

書式 :PLAY[:START] チャンネル名称,指令

説明 チャンネル名称で指定する出力端へ信号のプレイ動作を開始、終了させます。

「:PLAY:START チャンネル名称,ENABLE」の後のトリガ発生でプレイ動作を開始します。メモリ領域からデータを出力し、「:PLAY:CLOCK:LEVEL」で指定した間隔でリレーを動作/復旧させます。書き込まれたデータの数を越えたら終了します。

「:PLAY:START チャンネル名称,DISABLE」でプレイ動作を終了し、IDLE状態になります。

指令: ENABLE,DISABLE

ENABLEで開始します。しかし、このコマンド実行以前に「:PLAY:ASSIGN」コマンドが実行されている必要があります。  
DISABLEで終了します。

応答 このコマンドに対する応答はありません。

「:PLAY:START チャンネル名称,ENABLE」を受信したとき、指定と同じチャンネル名称に対するプレイ動作がSTANDBY状態やRUNNING状態にある時は無視します。

「:PLAY:START チャンネル名称,DISABLE」を受信したとき、指定と同じチャンネル名称に対するプレイ動作がIDLE状態にある時は無視します。

「:PLAY:ASSIGN」コマンドが実行されていないチャンネル名称に対する「:PLAY:START チャンネル名称,ENABLE」を受信すると「実行エラー」になります。

「:PLAY:START チャンネル名称,ENABLE」を受信したとき、指定されたチャンネル名称に割り当てられたメモリ領域に対するプレイ動作がSTANDBY状態やRUNNING状態にある時は「実行エラー」になります。

「:PLAY:START チャンネル名称,ENABLE」を受信したとき、指定されたチャンネル名称に割り当てられたメモリ領域に対するサンプル動作がSTANDBY状態やRUNNING状態にある時は「実行エラー」になります。

「 - 4 - 8 」

書式 :PLAY:STATE? チャンネル名称

説明 チャンネル名称で指定する出力端への信号のプレイ動作の状態を問い合わせます。

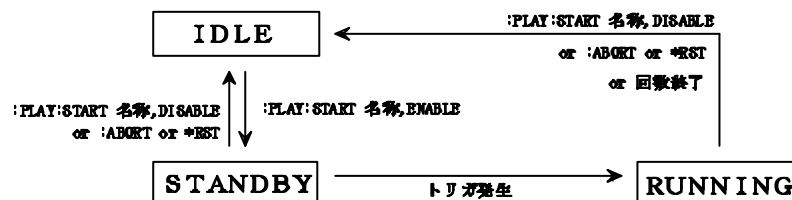
応答 このコマンドの後、トーカーに指定されると下記のいずれかの応答メッセージを返送します。

IDLE  
STANDBY  
RUNNING

IDLE状態: 「:PLAY:START チャンネル名称 ENABLE」コマンドを受信していません。  
または、指定された一連のプレイ動作をすべて終了しています。  
または、「:PLAY:START チャンネル名称 DISABLE」コマンドを受信したか、  
\*RST、\*ABORTなどの受信により、プレイ動作を強制終了しています。

STANDBY状態: 「:PLAY:START チャンネル名称 ENABLE」コマンドを受信し、トリガの発生を待っています。

RUNNING状態: 「:PLAY:START チャンネル名称 ENABLE」コマンドを受信し、トリガが発生し、一連のサンプル動作を行っています。



[ - 5 ] ステータス操作コマンド

STATUSコマンドセット

コマンド	パラメータ	備考
:STATus :EXTernal :TRANSition	数値(0~255)	イベント発生条件を設定する 0 = HighからLowへの変化で発生 1 = LowからHighへの変化で発生
:ENable	数値(0~255)	イベント発生によるSRQ送出を 禁止/許可する 0 = 禁止、1 = 許可
:TRANSition? :EVEnt? :ENable? :CONDition?		イベント発生条件をクエリする イベントの発生状況をクエリする イベント発生によるSRQ送出の 禁止/許可をクエリする 条件レジスタをクエリする

「 - 5 - 1 」

書式 :STATUS:EXTERNAL:TRANSITION 数値

説明 端末側の外部ステータス入力「 $\overline{ST1}$ 、 $\overline{ST2}$ 、 $\overline{ST3}$ 、 $\overline{ST4}$ 、 $\overline{ST5}$ 、 $\overline{ST6}$ 、 $\overline{ST8}$ 」によるイベント発生条件を設定します。設定は、0~255の範囲の数値で行います。REQ信号のイベント発生条件は「HighからLowの変化」に固定のため、ビット6（ $\overline{REQ}$ 信号）はセットされず、常に0になります。例えば、 $\overline{ST5}$ と $\overline{ST8}$ はLowからHighの変化で、他はHighからLowの変化でイベント発生とする場合の数値は、 $16 + 128$ なので、144を設定します。この数値は外部・ステータス・トランジション・レジスタに設定されます。

応答 このコマンドに対する応答はありません。

イネーブル・レジスタがON（1）に設定されている該当ビットのトランジション・レジスタの値によって、「HighからLowの変化」または「LowからHighの変化」を検出し、イベントを発生させます。イベントが発生するとイベント・レジスタの該当ビットがON（1）になります。本機の外部ステータス入力のイベント検出は、REQ信号はハードウェア割込で行っているため、高速の信号（パルス幅100ns以上）でも検出できますが「HighからLowの変化」の検出専用です。「 $\overline{ST1}$ 、 $\overline{ST2}$ 、 $\overline{ST3}$ 、 $\overline{ST4}$ 、 $\overline{ST5}$ 、 $\overline{ST6}$ 、 $\overline{ST8}$ 」はソフトウェアでの監視により行っているため、どちらの変化も検出できますが高速の信号（パルス幅500us以下）には対応できません。

「 - 5 - 2 」

書式 :STATUS:EXTERNAL:ENABLE 数値

説明 端末側の外部ステータス入力「 $\overline{ST1}$ 、 $\overline{ST2}$ 、 $\overline{ST3}$ 、 $\overline{ST4}$ 、 $\overline{ST5}$ 、 $\overline{ST6}$ 、 $\overline{ST8}$ 」および、 $\overline{REQ}$ 信号によるイベント発生でステータス・ビット・レジスタのEXSビット（ビット0）をON（1）にするかどうかを設定します。（EXSビットがONになった時、GPIBのSRQをアクティブにするかどうかは\*SRE（共通コマンド）で設定します。）設定は、0~255の範囲の数値で行います。例えば、REQ信号または $\overline{ST8}$ のイベント発生でEXSビットをONにする場合の数値は、 $64 + 128$ なので、192を設定します。この数値は外部・ステータス・イネーブル・レジスタに設定されます。

応答 このコマンドに対する応答はありません。

「 - 5 - 3 」

書式 :STATUS:EXTERNAL:TRANSITION?

説明 端末側の外部ステータス入力「 $\overline{ST1}$ 、 $\overline{ST2}$ 、 $\overline{ST3}$ 、 $\overline{ST4}$ 、 $\overline{ST5}$ 、 $\overline{ST6}$ 、 $\overline{ST8}$ 」および $\overline{REQ}$ 信号によるイベント発生条件の設定内容を読み出します。

応答 このコマンドの後、トーカーに指定されると応答メッセージとして、外部・ステータス・トランジション・レジスタの内容を、下記のように10進数値で返送します。数値の範囲は0~63、128~191です。数値が飛んでいるのはビット6（REQ信号）はセットできず、0に固定のためです。

数値

「 - 5 - 4 」

書式 :STATUS:EXTERNAL:EVENT?

説明 端末側の外部ステータス入力「 $\overline{ST1}$ ,  $\overline{ST2}$ ,  $\overline{ST3}$ ,  $\overline{ST4}$ ,  $\overline{ST5}$ ,  $\overline{ST6}$ ,  $\overline{ST8}$ 」および $\overline{REQ}$ 信号によるイベント発生条件によるイベントの発生状況を読み出します。  
読み出された外部・ステータス・イベント・レジスタはクリアされます。

応答 このコマンドの後、トーカーに指定されると応答メッセージとして、外部・ステータス・イベント・レジスタの内容を、下記のように10進整数値で返送します。

数値

「 - 5 - 5 」

書式 :STATUS:EXTERNAL:ENABLE?

説明 端末側の外部ステータス入力「 $\overline{ST1}$ ,  $\overline{ST2}$ ,  $\overline{ST3}$ ,  $\overline{ST4}$ ,  $\overline{ST5}$ ,  $\overline{ST6}$ ,  $\overline{ST8}$ 」および $\overline{REQ}$ 信号によるイベント発生条件によるイベント発生でのSRQ送出手の可否設定内容を読み出します。

応答 このコマンドの後、トーカーに指定されると応答メッセージとして、外部・ステータス・イネーブル・レジスタの内容を、下記のように10進整数値で返送します。

数値

「 - 5 - 6 」

書式 :STATUS:EXTERNAL:CONDITION?

説明 端末側の外部ステータス入力「 $\overline{ST1}$ ,  $\overline{ST2}$ ,  $\overline{ST3}$ ,  $\overline{ST4}$ ,  $\overline{ST5}$ ,  $\overline{ST6}$ ,  $\overline{ST8}$ 」および $\overline{REQ}$ 信号を読み出します。

応答 このコマンドの後、トーカーに指定されると応答メッセージとして、外部・ステータス・条件・レジスタの内容を、下記のように10進整数値で返送します。

数値

[ - 6 ] アボート・コマンド

ABORTコマンドセット

コマンド	パラメータ	備考
:ABORT		

トリガ・システムをアイドル・ステートにセットする。

「 - 6 - 1 」

書式 :ABORT

説明 トリガ・システムをアイドル・ステートにし、プレイ動作の状態をIDLEにします。

応答 このコマンドに対する応答はありません。

[ - 7 ] 電圧範囲設定・コマンド

CONFIGUREコマンドセット

コマンド	パラメータ	備考	初期値
:CONFigure :OUTput	チャンネル名称(CH0,1), 電圧範囲, データ単位	D/A変換・電圧出力の範囲の選択を行う。範囲選択状況を問い合わせる。 電圧のデータの形式を選択・指定する。 電圧のデータの形式を問い合わせる。	電圧範囲=P10 データ単位=C12
:OUTput?	チャンネル名称(CH0,1)		
:MEMory	アドレス番号(0,1), 電圧範囲, データ単位		電圧範囲=P10 データ単位=C12
:MEMory?	アドレス番号(0,1)		

本機のボード上にD/A変換・電圧出力範囲を選択するジャンパが実装されています。このジャンパを設定することで電圧出力は希望する範囲に設定されます。(取扱説明書[ - 1 - 2 ]を参照)

しかし、それだけではホストコンピュータ(パソコンなど)とのデータの送受において、電圧値を表す数値を使用することはできません。電圧値を出力電圧の設定コードに変換した数値を使用しなければなりません。

あらかじめ、この「CONFIGUREコマンド」を使用して、ソフト的な「D/A変換・電圧出力範囲」を設定しておく。「出力電圧の設定コード」に変換しなくても、データの送受において「電圧値」を使用することができます。

「 - 7 - 1 」

書式 :CONFIGURE:OUTPUT チャンネル名称, 電圧範囲, データ単位

説明 チャンネル名称で指定するD/A出力CHの電圧出力範囲とパソコンなどと本機との間で送受するデータの単位を設定します。

チャンネル名称: D/A出力CHを指定します。CH0またはCH1のいずれかです。  
このパラメータがDAのばあいにはDA0とみなし、DA0はCH0、DA1はCH1とみなします。

電圧範囲: 本機のボード上のジャンパで選択設定した出力電圧範囲を下表に示す「電圧範囲の記号」で指定します。(取扱説明書[ - 6 ]を参照)

電圧範囲の記号	出力電圧範囲
P10	0V ~ +10V
P05	0V ~ +5V
B10	-10V ~ +10V
B05	-5V ~ +5V
N10	0V ~ -10V
N05	0V ~ -5V

データ単位: 本機とパソコンなどとの間で送受する、チャンネル名称で指定するD/A出力CHの電圧のデータの単位を下表に示す「データ単位の記号」で指定します。

データ単位の記号	送受データの単位
C12	出力電圧の設定コード
V11	mV
V00	V

例: ボード上のジャンパ設定で「-10V ~ +10V」を選択していて、本機と送受するデータの単位を「mV」で扱う場合は下記ようになります。

:CONFIGURE:OUTPUT チャンネル名称,B10,V11

工場出荷時のジャンパ設定はどちらのD/A出力CHとも「0V ~ +10V」になっており、本機と送受するデータの単位は「出力電圧の設定コード」として扱われます。そしてソフトの設定は下記のコマンドを受信した場合と同等になっています。

:CONFIGURE:OUTPUT チャンネル名称,P10,C12

応答 このコマンドに対する応答はありません。

このコマンドで指定するチャンネル名称のプレイ動作がSTANDBY状態やRUNNING状態にある場合には、電圧範囲が変更になるような設定は「実行エラー」になります。

このコマンドで指定するチャンネル名称が「:PLAY:ASSIGN」で割り当てられているメモリの領域に対する「:CONFIGURE:MEMORY」で設定されている電圧範囲が異なるような設定を行うと割り当てが解除されます。

「 - 7 - 2 」

書式 :CONFIGURE:OUTPUT? ファン札名称

説明 ファン札名称で指定する D / A 出力 C H の電圧出力範囲の設定状況を問い合わせます。

ファン札名称 : D / A 出力 C H を指定します。CH0 または CH1 のいずれかです。  
このパラメータが DA のばあいは DA0 とみなし、DA0 は CH0、DA1 は CH1 とみなします。

応答 このコマンドの後、トーカに指定されるとファン札名称で指定する D / A 出力 C H の電圧出力範囲の設定状況を下記のフォーマットで返送します。

電圧範囲, データ単位

電圧範囲とデータ単位の内容は「:CONFIGURE:OUTPUT ファン札名称, 電圧範囲, データ単位」で設定された内容と同じです。

「 - 7 - 3 」

書式 :CONFIGURE:MEMORY ブロック番号, 電圧範囲, データ単位

説明 ブロック番号で指定するメモリの領域に対して読み書きを行う場合の本機との間で送受するデータの電圧値に関する単位を設定します。

ブロック番号 : メモリの領域を指定します。0 または 1 のいずれかです。

電圧範囲 : メモリの領域に対して読み書きする電圧範囲を下表に示す「電圧範囲の記号」で指定します。

電圧範囲の記号	電圧範囲
P 1 0	0 V ~ + 1 0 V
P 0 5	0 V ~ + 5 V
B 1 0	- 1 0 V ~ + 1 0 V
B 0 5	- 5 V ~ + 5 V
N 1 0	0 V ~ - 1 0 V
N 0 5	0 V ~ - 5 V

データ単位 : 本機とパソコンなどとの間で送受する、ブロック番号で指定するメモリの領域に対する電圧のデータの単位を下表に示す「データ単位の記号」で指定します。

データ単位の記号	送受データの単位
C 1 2	電圧の設定コード
V 1 1	m V
V 0 0	V

例 : 電圧範囲を「- 1 0 V ~ + 1 0 V」、本機と送受するデータの単位を「m V」で扱う場合は下記のようになります。

:CONFIGURE:MEMORY ブロック番号, B10, V11

工場出荷時の設定はどちらのメモリの領域に対して読み書きする電圧のデータの電圧範囲は「0 V ~ + 1 0 V」になっており、本機と送受するデータの単位は「電圧の設定コード」として扱われます。これは下記のコマンドを受信した場合と同等です。

:CONFIGURE:MEMORY ブロック番号, P10, C12

応答 このコマンドに対する応答はありません。

指定するメモリの領域にデータが存在する場合には電圧範囲が変更になるような設定を行うことはできません。「実行エラー」になります。「:MEMORY:ASSIGN」または「:MEMORY:WRITE:INITIALIZE」コマンドでメモリの中のデータを消去してから再設定してください。

このコマンドで指定するブロック番号が「:PLAY:ASSIGN」で割り当てられているファン札名称に対する「:CONFIGURE:OUTPUT」で設定されている電圧範囲が異なるような設定を行うと割り当てが解除されます。

「:MEMORY:READ:FORMAT」コマンドで、1 0 進数 (DECIMAL) 以外を設定している場合、このコマンドで「m V」(V11)「V」(V00)を指定すると「実行エラー」になります。

「 - 7 - 4 」

書式 :CONFIGURE:MEMORY? ブロック番号

説明 ブロック番号で指定するメモリの領域に対して読み書きを行う場合の本機との間で送受するデータの電圧値に関する単位の設定状況を問い合わせます。

ブロック番号：メモリの領域を指定します。0または1のいずれかです。

応答 このコマンドの後、トーカーに指定されるとブロック番号で指定するメモリの領域に対する読み書きする電圧値に関する単位の設定状況を下記のフォーマットで返送します。

電圧範囲,データ単位

電圧範囲とデータ単位の内容は「:CONFIGURE:MEMORY ブロック番号,電圧範囲,データ単位」で設定された内容と同じです。