

GPIBアイソレートアダプタ

P I C - 7 8 9 G P B (オープンフレーム)
P I C - 7 8 9 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789-9
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]	P I C - 7 8 9 G P ・外部・ステータス・レジスタ	_____	1 0
[- 4]	ステータス・レジスタの初期値	_____	1 1
【 】	S C M C コマンド		
[- 1]	入力端からの入力コマンド	_____	1 3
[- 2]	出力端への出力コマンド	_____	1 5
[- 3]	バッファメモリ・コマンド	_____	1 7
[- 4]	入力端データのバッファリング・コマンド	_____	2 1
[- 5]	バッファリングされたデータの出力・コマンド	_____	2 5
[- 6]	ステータス操作コマンド	_____	2 9
[- 7]	アボート・コマンド	_____	3 1

【 】概要

本「コマンド説明書」ではP I C - 7 8 9 G P B / CをA S C I I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。

P I C - 7 8 9 G P B / CのA S C I Iモードの操作コマンドやその機能は、I E E E - S t d 4 8 8 . 2 - 1 9 9 2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P I C - 7 8 9 G P BとP I C - 7 8 9 G P CのことをP I C - 7 8 9 G P B / Cと記述しています。

[- 1] 概略動作

本機「P I C - 7 8 9 G P B / 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、G P I Bコントローラが必要です。
通常はG P I Bコントローラ機能を持ったパソコンをG P I B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、本機の端末側出力信号「LD1~LD8」にデータを出力したり、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][- 2][- 6]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G P I B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]~[- 5]をご参照ください)

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A S C I I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A S C I I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。

P I C - 7 8 9 G P B / C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S C M C (Standard Commands for Measurement and Control) コマンドを使うことができます。

S C M Cコマンドの二モニックは
[]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)

P I C - 7 8 9 G P B / C 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。

P I C - 7 8 9 G P B / C 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,PIC-789GP,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。

- * LD 1 ~ LD 8 の出力を OFF (OPEN) にする
- * GPIB からの受信バッファをクリアする
- * INPUT コマンドシステム、SAMPLE コマンドシステム、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、
 SAMPLE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。
 出力データ(LD1~LD8)の状態、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
 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 (GPIBへの送信バッファ) に
 ASCIIR「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
 ASCIIR「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GETコマンドと同じです。
 サンプル動作やプレイ動作を起動させます。(本書 [- 4] [- 5] 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。

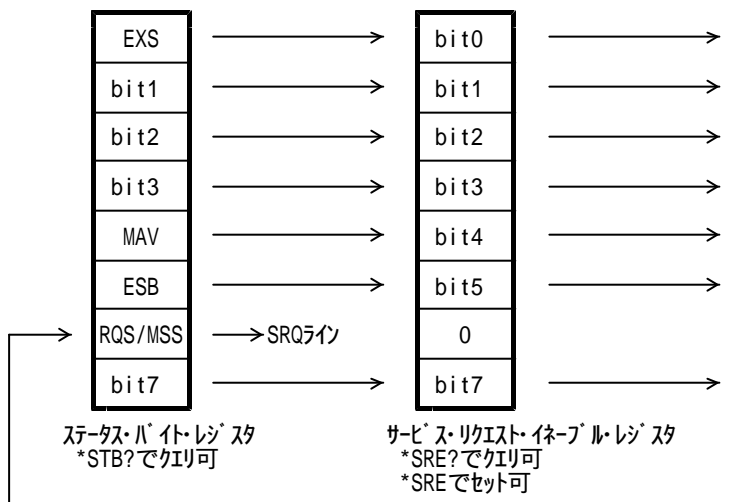
bit 4 : MAV : メッセージ・アベイラブル・ビット
機器のデータ出力の待ち行列が空であるかどうかを示します。
本機の GPIB 送信バッファに送信データがある場合、1 にセットされます。

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。

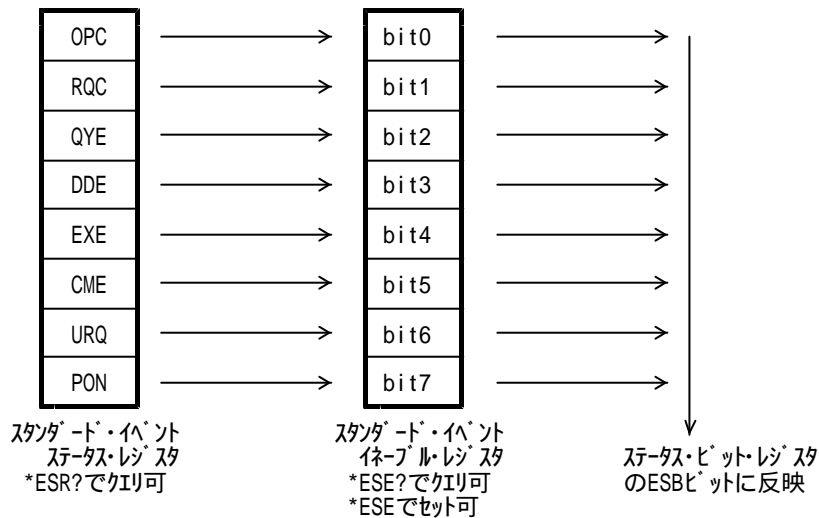
MSS : マスター・ステータス・サマリ
ステータス・ビット・レジスタの他の 7 ビットの代表。
過去に本機がサービス・リクエストを発生したかどうかを示します。
シリアル・ポールによって RQS ビットがクリアされた後も MSS ビットはクリアされません。

bit 7 : : 本機においては常に 0 です。



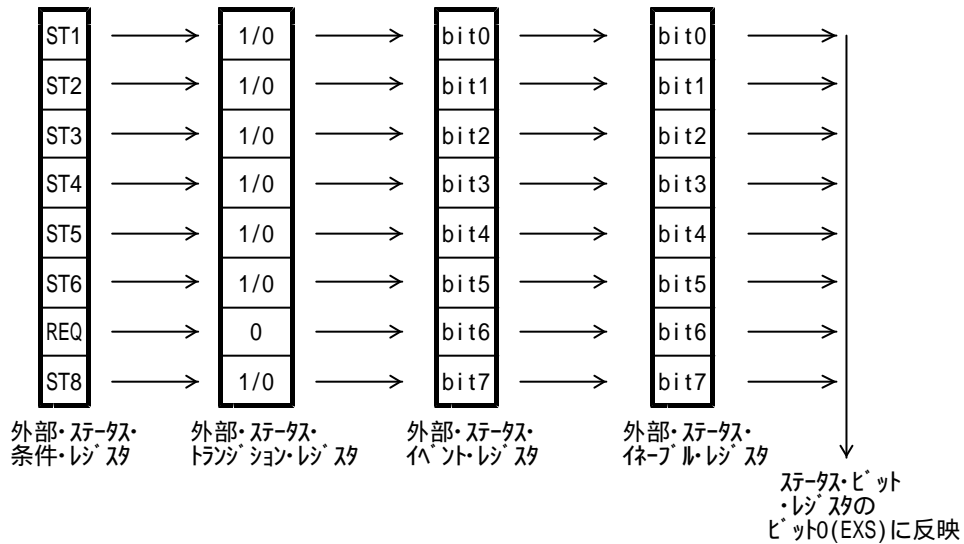
[- 2] スタンダード・イベント・ステータス・レジスタ (SESR)

- 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
相手機器が本機から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] PIC - 7 8 9 GP ・外部・ステータス・レジスタ

- bit 0 : ST 1 : 外部端末側ステータス入力 $\overline{ST\ 1}$ の状態
外部端末側ステータス入力信号 ST 1 の状態を表します。
- bit 1 : ST 2 : 外部端末側ステータス入力 $\overline{ST\ 2}$ の状態
外部端末側ステータス入力信号 ST 2 の状態を表します。
- bit 2 : ST 3 : 外部端末側ステータス入力 $\overline{ST\ 3}$ の状態
外部端末側ステータス入力信号 ST 3 の状態を表します。
- bit 3 : ST 4 : 外部端末側ステータス入力 $\overline{ST\ 4}$ の状態
外部端末側ステータス入力信号 ST 4 の状態を表します。
- bit 4 : ST 5 : 外部端末側ステータス入力 $\overline{ST\ 5}$ の状態
外部端末側ステータス入力信号 ST 5 の状態を表します。
- bit 5 : ST 6 : 外部端末側ステータス入力 $\overline{ST\ 6}$ の状態
外部端末側ステータス入力信号 ST 6 の状態を表します。
- bit 6 : REQ : 外部端末側リクエスト入力 \overline{REQ} の状態
外部端末側リクエスト入力信号 REQ の状態を表します。
- bit 7 : ST 8 : 外部端末側ステータス入力 $\overline{ST\ 8}$ の状態
外部端末側ステータス入力信号 ST 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 [- 6] 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 PIC-789GPB/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。

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]? :FORMat :FORMat?	ビット名称 (BIT0~7) バイト名称 (BYTE0) データ形式	データ形式の指定 データ形式の問い合わせ	DECIMAL

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
TD1~TD8の8ビットを総称する名称です。
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 :
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。

「 - 1 - 1 」

書式 : INPUT[:DATA]? ビット名称
: INPUT[:DATA]? バイト名称

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入力し、「:INPUT:FORMAT」で指定する表現で、
応答メッセージを作成することを指示します。
「: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。

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入力端の信号を入力し、そのデータを
ASCII文字の数値で下記のように返送します。

数値

「:INPUT:DATA? バイト名称」コマンドに対する応答の場合、データの値は0~255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。
10進数の場合は、例えば27となっています。
16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」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。
10進数の場合は、0または1となっています。
16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。

「 - 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」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。

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」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

[- 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	ビット名称 (BIT0~7), 出力データ	
:OUTput?	バイト名称 (BYTE0), 出力データ ビット名称 (BIT0~7), データ形式 バイト名称 (BYTE0), データ形式	

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LD1, LD2, LD3, LD4, LD5, LD6, LD7, LD8も使用できます。
LD1はBIT0と, LD2はBIT1と, LD3はBIT2と, ..., LD8はBIT7と同じとみなします。
ただし, LDはLD1と同じ, とはみなしません。

バイト名称 : BYTE0
LD1~LD8の8ビットを総称する名称です。
このパラメータが BYTE または LD の場合は BYTE0 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。
論理を指定する場合は, LOGical と記述します。

「 - 2 - 1 」

書式 : OUTPUT ビット名称, 出力データ
:OUTPUT バイト名称, 出力データ

説明 ビット名称またはバイト名称で指定する出力端へ出力データを出力させます。

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。
出力先がバイトの場合は、データの値は0~255の範囲でなければなりません。
出力先がビットの場合は、データの値は0~1の範囲でなければなりません

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。
出力先がビットの場合に限って、論理表現、LONまたはLOFFと記述してもかまいません。

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します。

出力先がバイトの場合、データが整数でない場合は整数になるよう、四捨五入されます。
その結果のデータは0から255の範囲の正の値でなければなりません。
範囲外はエラーになります。
出力先には四捨五入した整数値が出力されます。

出力先がビットの場合、データが整数でない場合は整数になるよう、四捨五入されます。
その結果のデータは0または1の範囲の正の値でなければなりません。
範囲外はエラーになります。
出力先には四捨五入した整数値が出力されます。

応答 このコマンドに対する応答はありません。

書式 :OUTPUT? ビット名称[,データ形式]
:OUTPUT? バイト名称[,データ形式]

説明 ビット名称またはバイト名称で指定する出力端の信号をモニタし、データ形式で指定する表現で、
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:

モニタする出力端がビットの場合、
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。
論理を指定する場合は、LOGical と記述します。

モニタする出力端がバイトの場合、
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。

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出力端の信号をモニタし、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。

モニタ先がビットの場合、データは0か1の正の整数値です。

基数が2進数の場合は、#B0または#B1となっています。
10進数の場合は、0または1となっています。
16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。

モニタ先がバイトの場合、データは0から255の範囲の正の整数値です。

基数が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#65となっています。
16進数の場合は、例えば#H41となっています。
8進数の場合は、例えば#Q141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),バイト数 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	指定領域の書込バイトを初期化 書込バイトから書込み、 書込バイトを次へ移す。	
:READ :INITialize [:NEXT]?	ブロック番号(0~1) ブロック番号(0~1),バイト数	指定領域の読出バイトを初期化 読出バイトから読出し、 読出バイトを次へ移す。	
: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

サンプル動作やプレイ動作が、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,バイト数

サンプル動作やプレイ動作が、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,バイト数
: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,バイト数

ブロック番号 : 0,1

確保した領域はサンプル動作やプレイ動作で使用します。

「 - 3 - 1 」

書式 :MEMORY:ASSIGN ブロック番号,バイト数

説明 ブロック番号で指定するメモリ領域の容量をバイト数で指定確保します。
読出バイトと書込バイトは、この領域の先頭に初期化されます。

バイト数 : 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容量をバイト単位で指定します。
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、「:SAMPLE:ASSIGN」や「: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。

サンプル動作やプレイ動作が、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。

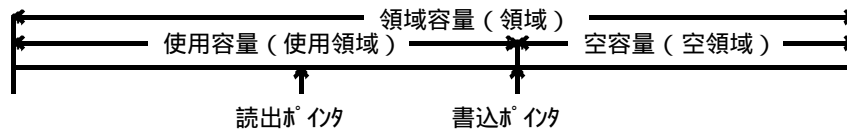
応答 このコマンドの後、トーカーに指定されると下記の応答メッセージを返送します。

領域容量, 使用容量, 空容量

領域容量: 「:MEMORY:ASSIGN ブロック番号, バイト数」で確保されている領域の容量をバイト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バイト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バイト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。サンプル動作やプレイ動作が、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 - 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。

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。

<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。

出力先がビットの場合は、(0x00)か(0x01)しかあり得ません。

出力先がバイトの場合は、(0x00)~(0xFF)の範囲のデータを入れて下さい。

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。サンプル動作やプレイ動作が、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 - 5 」

書式 :MEMORY:READ:INITIALIZE ブロック番号

説明 ブロック番号で指定する領域からの読出ポインタを初期化します。
書込ポインタは変化しません。

応答 このコマンドに対する応答はありません。
サンプル動作やプレイ動作が、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? ブロック番号,バイト数

説明 ブ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、読出ポインタ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。
書込ポインタは変化しません。

バイト数: 0 または、1 ~ 1000000

ブ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バイト単位で指定します。
指定したブロック番号の領域に存在する未読み出しデータの数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、指定したブ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、トークンに指定されると「:MEMORY:READ:FORMAT ブロック番号,データ形式」で
指定されているデータ形式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。数値の表現はデータ形式に従います。

「:MEMORY:READ:NEXT ブロック番号,バイト数」で指定したバイト数、

またはメモリ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。数値の表現はデータ形式に従います。
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。

「:MEMORY:READ:NEXT ブロック番号,バイト数」で指定したバイト数、

またはメモリに入っていたデータのバイトの単位での数のどちらか小さい方です。

このmは、10進数で表現します。

<DAB1> ~ <DABm>: 各<DABm>は、で区切られていません。

サンプル動作やプレイ動作が、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,バイト数」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。

「 - 3 - 8 」

書式 :MEMORY:READ:FORMAT? プ ロック番号

説明 「 :MEMORY:READ:NEXT プ ロック番号,バ イット数」コマンドに対する応答メッセージのデータ形式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。

総領域容量,総空容量

総領域容量 : 「 :MEMORY:ASSIGN プ ロック番号,バ イット数」で確保されているメモリ領域の合計をバイト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バイトの単位で表しています。
「 :MEMORY:ASSIGN プ ロック番号,バ イット数」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 バイトの)場合、1 0 2 4 バイトです。

[- 4] 入力端データのバッファリング・コマンド

SAMPLEコマンドセット

コマンド	パラメータ	備考	初期値
:SAMPle			
:CLOCK			
:LEVel	ハ` 1 名称 (BYTE0), クロックのレベル値 ビット名称 (BIT0 ~ 7), クロックのレベル値	レベル値の設定 設定は 1 ~ 10000000 (m秒) 設定精度は ±100u秒	1 (m秒)
:LEVel?	ハ` 1 名称 (BYTE0) ビット名称 (BIT0 ~ 7)	レベル値の問い合わせ 応答は 1 ~ 10000000 (m秒)	
:ASSign	ハ` 1 名称 (BYTE0), ブロック番号, データ数 ビット名称 (BIT0 ~ 7), ブロック番号, データ数	サンプル入出力の割当て	割り当てなし
:ASSign?	ハ` 1 名称 (BYTE0) ビット名称 (BIT0 ~ 7)	サンプル入出力の問合せ	
[:STARt]	ハ` 1 名称 (BYTE0), 指令 ビット名称 (BIT0 ~ 7), 指令	指令は下記のいずれか。 ENABle, DISABle	
:STATe?	ハ` 1 名称 (BYTE0) ビット名称 (BIT0 ~ 7)	SAMPLE動作の状態を返す。 下記のいずれか。 IDLE, STANDBY, RUNNING	IDLE

サンプル動作が STANDBY の状態にある場合は同一ブロック番号のメモリ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, ハ` 1 数

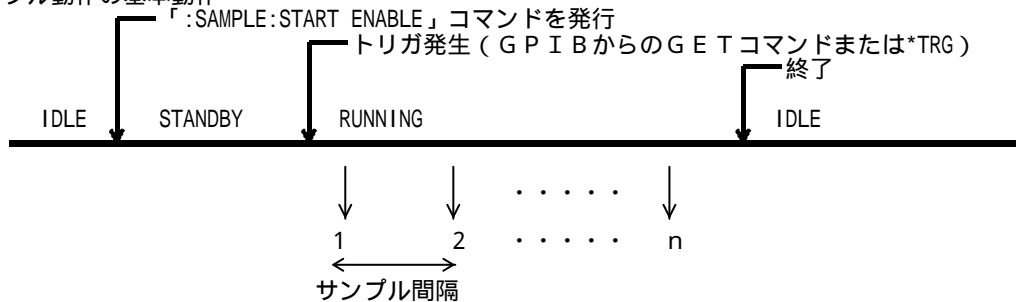
サンプル動作が RUNNING 状態にある場合は同一ブロック番号のメモリに対して以下のことを行うことができません。

:MEMORY:ASSIGN ブロック番号, ハ` 1 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, ハ` 1 数

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 BIT0 と同じ, とはみなしません。

ハ` 1 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。

サンプル動作の基本動作



サンプル間隔は「:SAMPLE: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。
サンプルしてメモリに格納するデータの数 n は原則として
 $n = \text{「:SAMPLE:ASSIGN」で指定したデータ数}$
で表されます。
トリガが発生した時点で指定されたメモリ領域の書込ハ` 1 と読出ハ` 1 を初期化して、領域の先頭からサンプルしたデータを保存します。サンプル動作が終了すると書込ハ` 1 はサンプルし保存した最後のデータの次をポイントしています。

「 - 4 - 1 」

書式 :SAMPLE:CLOCK:LEVEL ビット名称, クラックのレベル値
:SAMPLE:CLOCK:LEVEL バイト名称, クラックのレベル値

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サンプルするためのクロック源のタイマー値を指定します。

クラックのレベル値: 1 ~ 10000000

出力端へ信号を出力する間隔を指定します。

1 m秒 ~ 1 0 0 0 0 0 0 0 m秒の範囲の 1 m秒の整数倍の値で設定します。

範囲以外はエラーになり、以前の設定値が残ります。

「:SAMPLE: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入力端のデータを読み取りメモリに格納します。

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ビット名称/バイト名称のサンプル動作が RUNNING 状態にある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 :SAMPLE:CLOCK:LEVEL? ビット名称
:SAMPLE:CLOCK:LEVEL? バイト名称

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サンプルするためのクロック源のタイマーの設定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ラックのレベル値で、サンプル間隔を示しています。
数値の範囲は 1 m秒 ~ 1 0 0 0 0 0 0 0 m秒の範囲の 1 m秒の整数倍の値です。

数値

「 - 4 - 3 」

書式 :SAMPLE:ASSIGN ビット名称, ブロック番号, データ数
:SAMPLE:ASSIGN バイト名称, ブロック番号, 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サンプルして得られるデータを格納するメモリ領域の割り当てと、サンプル動作で得られるデータの数を指定します。

ブロック番号: 0, 1

あらかじめ、「MEMORY:ASSIGN ブロック番号, バイト数」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: 0, 1 ~ メモリ領域のメモリ容量以内

サンプル動作で得られるデータの数を指定します。

0 を指定すると、サンプルデータ入力端とデータ格納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。

あらかじめ、「MEMORY:ASSIGN ブロック番号, バイト数」コマンドで、メモリ領域とその容量を定義確保していない場合は「実行エラー」になります。

このコマンドで指定するビット名称/バイト名称のサンプル動作が STANDBY 状態や RUNNING 状態にある時に、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ビット名称/バイト名称に、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、データ数が 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ビット名称/バイト名称に、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。

「 - 4 - 4 」

書式 :SAMPLE:ASSIGN? ビット名称
:SAMPLE:ASSIGN? バイト名称

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サンプルして得られるデータを格納するメモリ領域の割り当てと、サンプル動作で得られるデータの数の指定状況を問い合わせます。

応答 このコマンドの後、トリガ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、指定されたビット名称/バイト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 - 5 」

書式 :SAMPLE[:START] ビット名称,指令
:SAMPLE[:START] バイト名称,指令

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のサンプル動作を開始、終了させます。

「:SAMPLE:START ビット名称/バイト名称,ENABLE」の後のトリガ発生でサンプル動作を開始します。
「:SAMPLE:ASSIGN」コマンドのデータ数で指定した数のデータを、ブロック番号で指定したメモリに取り込みます。取り込む間隔は「:SAMPLE:CLOCK:LEVEL」のレベル値で指定したタイマーの間隔です。
「:SAMPLE:ASSIGN」コマンドのデータ数で指定した数のデータを取り込みを行うと終了し、IDLE状態になります。

「:SAMPLE:START ビット名称/バイト名称,DISABLE」でサンプル動作を終了し、IDLE状態になります。取り込んだデータの数が確保されたメモリ領域のメモリ容量に達した場合も、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「:SAMPLE: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。

「:SAMPLE:START ビット名称/バイト名称,ENABLE」を受信したとき、指定と同じビット名称/バイト名称に対するサンプル動作がSTANDBY状態やRUNNING状態にある時は無視します。
「:SAMPLE:START ビット名称/バイト名称,DISABLE」を受信したとき、指定と同じビット名称/バイト名称に対するサンプル動作がIDLE状態にある時は無視します。

「:SAMPLE:ASSIGN」コマンドが実行されていないビット名称/バイト名称に対する「:SAMPLE:START ビット名称/バイト名称,ENABLE」を受信すると「実行エラー」になります。

「:SAMPLE:START ビット名称/バイト名称,ENABLE」を受信したとき、指定されたビット名称/バイト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。

「:SAMPLE:START ビット名称/バイト名称,ENABLE」を受信したとき、指定されたビット名称/バイト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 - 6 」

書式 :SAMPLE:STATE? ビット名称
:SAMPLE:STATE? バイト名称

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のサンプル動作の状態を問い合わせます。

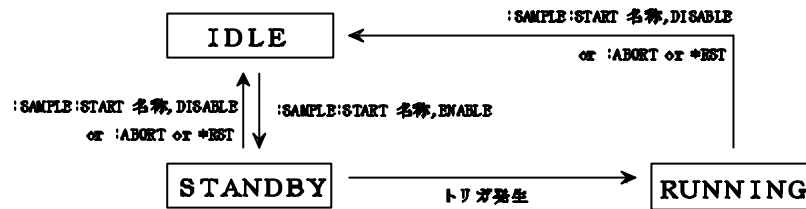
応答 このコマンドの後、トークンに指定されると下記のいずれかの応答メッセージを返送します。

IDLE
STANDBY
RUNNING

IDLE状態: 「:SAMPLE:START ビット名称/バイト名称 ENABLE」コマンドを受信していません。
または、サンプル動作をすべて終了しています。
または、「:SAMPLE:START ビット名称/バイト名称 DISABLE」コマンドを受信したか、
*RST、*ABORTなどの受信により、サンプル動作を強制終了しています。

STANDBY状態: 「:SAMPLE:START ビット名称/バイト名称 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: 「:SAMPLE:START ビット名称/バイト名称 ENABLE」コマンドを受信し、
トリガが発生し、サンプル動作を行っています。
トリガが発生した時「:MEMORY:WRITE:INITIALIZE」と同じ操作が行われるため、
書込ポインタ、読出ポインタがリセットされます。



[- 5] 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	ハ` 1 名称 (BYTE0), クロックのレベル値 ビット名称 (BIT0~7), クロックのレベル値	レベル値の設定 設定値は 1 ~ 10000000 (m秒) 精度は ± 100u秒	1 (m秒)
:LEVel?	ハ` 1 名称 (BYTE0) ビット名称 (BIT0~7)	レベル値の問い合わせ 応答は 1 ~ 10000000 (m秒)	
:REPeat	ハ` 1 名称 (BYTE0), 回数 (0, 1 ~ 1000000) ビット名称 (BIT0~7), 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?	ハ` 1 名称 (BYTE0) ビット名称 (BIT0~7)	繰り返し回数の問い合わせ 応答は 0 ~ 1 0 0 0 0 0 0	
:ASSign	ハ` 1 名称 (BYTE0), ブロック番号, データ数 ビット名称 (BIT0~7), 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?	ハ` 1 名称 (BYTE0) ビット名称 (BIT0~7)	プレイ入出力の問い合わせ	
[:STARt]	ハ` 1 名称 (BYTE0), 指令 ビット名称 (BIT0~7), 指令	指令は下記のいずれか。 ENABle, DISABle	
:STATe?	ハ` 1 名称 (BYTE0) ビット名称 (BIT0~7)	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, ハ` 1 数

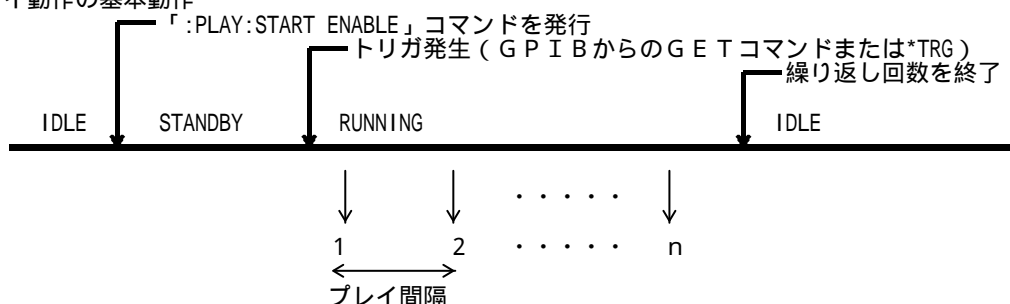
プレイ動作が RUNNING 状態にある場合は同一ブロック番号のメモリ領域に対して以下のことを行うことができません。

:MEMORY:ASSIGN ブロック番号, ハ` 1 数
:MEMORY:WRITE:INITIATE ブロック番号
:MEMORY:WRITE:NEXT ブロック番号, データ列
:MEMORY:READ:INITIATE ブロック番号
:MEMORY:READ:NEXT? ブロック番号, ハ` 1 数

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 LD1, LD2, LD3, LD4, LD5, LD6, LD7, LD8 も使用できます。
LD1 は BIT0 と, LD2 は BIT1 と, LD3 は BIT2 と, ..., LD8 は BIT7 と同じとみなします。
ただし, LD1 は LD1 と同じ, とはみなしません。

ハ` 1 名称: BYTE0
このパラメータが BYTE または LD の場合は BYTE0 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。

「 - 5 - 1 」

書式 :PLAY:CLOCK:LEVEL ビット名称, クロックのレベル値
:PLAY:CLOCK:LEVEL パッケージ名称, クロックのレベル値

説明 ビット名称、またはパッケージ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 : 1 ~ 10000000

出力端へ信号を出力する間隔を指定します。

1 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ビット名称/パッケージ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。

「 - 5 - 2 」

書式 :PLAY:CLOCK:LEVEL? ビット名称
:PLAY:CLOCK:LEVEL? パッケージ名称

説明 ビット名称、またはパッケージ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

「 - 5 - 3 」

書式 :PLAY:REPEAT ビット名称, 回数
:PLAY:REPEAT パッケージ名称, 回数

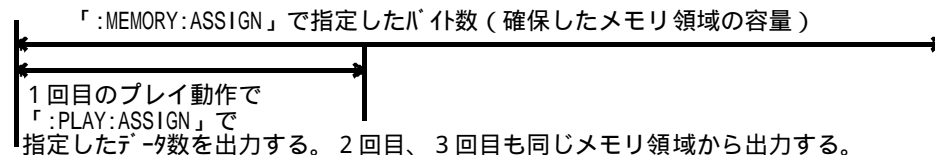
説明 ビット名称、またはパッケージ名称で指定する出力端へ信号をプレイする繰り返し回数を指定します。

回数 : 0, 1, 2, 3, ..., 1000000

1 ~ 1000000を指定すると一連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 ビット名称/パッケージ名称, 3」を指定した場合のメモリ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と「: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と「: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ビット名称/パッケージ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。

「 - 5 - 4 」

書式 :PLAY:REPEAT? ビット名称
:PLAY:REPEAT? パッケージ名称

説明 ビット名称、またはパッケージ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。

「 - 5 - 5 」

書式 :PLAY:ASSIGN ビット名称,ブロック番号,データ数
:PLAY:ASSIGN バイト名称,ブロック番号,データ数

説明 ビット名称、またはバイト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,バイト数」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,バイト数」コマンドのバイト数で、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ビット名称/バイト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ビット名称/バイト名称に、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、データ数が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ビット名称/バイト名称に、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。

「 - 5 - 6 」

書式 :PLAY:ASSIGN? ビット名称
:PLAY:ASSIGN? バイト名称

説明 ビット名称、またはバイト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の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、指定されたビット名称/バイト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。

「 - 5 - 7 」

書式 :PLAY[:START] ビット名称,指令
:PLAY[:START] バイト名称,指令

説明 ビット名称、またはバイト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 ビット名称/バイト名称,ENABLE」の後のトリガ発生でプレイ動作を開始します。
「:PLAY:ASSIGN」コマンドのデータ数で指定した数のデータを、ブロック番号で指定したメモリ領域から出力端へ出力します。

「:PLAY:ASSIGN」コマンドのデータ数で指定した数のデータの出力を「:PLAY:REPEAT」の回数で指定した回数、行くと終了し、IDLE状態になります。

「:PLAY:START ビット名称/バイト名称,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 ビット名称/バイト名称,ENABLE」を受信したとき、指定と同じビット名称/バイト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 ビット名称/バイト名称,DISABLE」を受信したとき、指定と同じビット名称/バイト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:START ビット名称,ENABLE」を受信したとき、指定のビット名称が含まれるバイト名称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 バイト名称,ENABLE」を受信したとき、指定のバイト名称に含まれるビット名称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:ASSIGN」コマンドが実行されていないビット名称/バイト名称に対する「:PLAY:START ビット名称/バイト名称,ENABLE」を受信すると「実行エラー」になります。

「:PLAY:START ビット名称/バイト名称,ENABLE」を受信したとき、指定されたビット名称/バイト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 ビット名称/バイト名称,ENABLE」を受信したとき、指定されたビット名称/バイト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。

「 - 5 - 8 」

書式 :PLAY:STATE? ビット名称
:PLAY:STATE? バイト名称

説明 ビット名称、またはバイト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。

IDLE
STANDBY
RUNNING

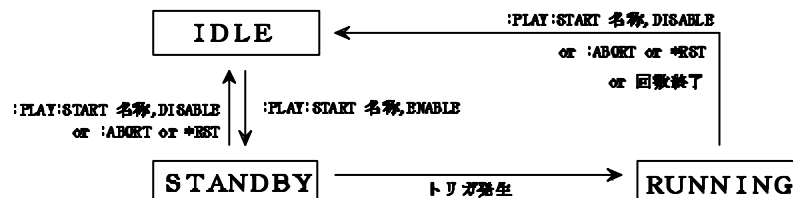
IDLE状態: 「:PLAY:START ビット名称/バイト名称 ENABLE」コマンドを受信していません。

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 ビット名称/バイト名称 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 ビット名称/バイト名称 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 ビット名称/バイト名称 ENABLE」コマンドを受信し、トリガが発生し、一連のプレイ動作を行っています。



[- 6] 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

「 - 6 - 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 (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。

「 - 6 - 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、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。

「 - 6 - 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

「 - 6 - 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

「 - 6 - 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送付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

「 - 6 - 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

[- 7] 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。

「 - 7 - 1 」

書式 :ABORT

説明 トリガ・システムをアイドル・ステートにし、プレイ状態やサンプル状態を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。