

端末組込用GPIBアダプタ

DIO-5432GPB (オープンフレーム)

コマンド説明書

エムシー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/>



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

目次

【I】概要		
[I-1]	概略動作	2
[I-2]	フォーマット	2
[I-3]	コマンド	2
[I-4]	パラメータ	2
[I-5]	デリミタ (ターミネータ)	3
[I-6]	エラー処理	3
【II】共通コマンド		
[II-1]	システムデータ・コマンド	4
[II-2]	内部操作・コマンド	4
[II-3]	同期・コマンド	5
[II-4]	ステータス/イベント・コマンド	6
[II-5]	デバイストリガ・コマンド	7
【III】ステータス報告システム		
[III-1]	ステータス・ビット・レジスタ	8
[III-2]	スタンダード・イベント・ステータス・レジスタ	9
[III-3]	DIO-5432GP・外部・ステータス・レジスタ	10
[III-4]	ステータス・レジスタの初期値	11
【IV】SCMCコマンド		
[IV-1]	入力端からの入力コマンド (5432モード)	13
[IV-2]	入力端からの入力コマンド (788Asci iモード)	15
[IV-3]	出力端への出力コマンド (5432モード)	18
[IV-4]	出力端への出力コマンド (788Asci iモード)	20
[IV-5]	バッファメモリ・コマンド (788Asci iモード)	23
[IV-6]	入力端データのバッファリング・コマンド (788Asci iモード)	27
[IV-7]	バッファリングされたデータの出力・コマンド (788Asci iモード)	31
[IV-8]	ステータス操作コマンド (5432モード 788Asci iモード)	35
[IV-9]	アボート・コマンド (788Asci iモード)	37

改版履歴	改版日付	改版内容
第1版	2024年05月01日	初版

【I】概要

本「コマンド説明書」ではDI0-5432GPBを「5432モード」や「788Ascii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。
 本書の説明の中では「DI0-5432GPB」を「本機」と表現している場合があります。
 また、「5432モード」や「788Asciiモード」を「ASCIIモード」と総称しています。
 ハード的な仕様やバイナリモード（788Binary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。

DI0-5432GPB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）

【I-1】概略動作

本機「DI0-5432GPB」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。

（本書【IV-1】～【IV-4】をご参照ください）
 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【IV-8】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【IV-5】～【IV-7】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【II】【III】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。

【I-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。

DI0-5440GPB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【I-5】を参照）

【I-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コマンドのニーモニックは
 []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。

【I-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。

[I - 5] デリミタ (ターミネータ)

DIO-5432GPB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 [II - 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。

DIO-5432GPB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。

通常、デリミタはコマンドやパラメータとは切り離して取り扱われますが、出力端への出力コマンドの場合、注意が必要です。(本書 [IV - 4 - 1] をご参照ください)

[I - 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bit 5 (CME) がON (1) になります。

対処 : 正しいコマンドを再度送ってください。

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bit 4 (EXE) がON (1) になります。

対処 : 正しいパラメータに修正して、再度送ってください。

または、排他関係を確認し、実行可能な時に送ってください。

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bit 2 (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bit 4 (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ください。

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bit 3 (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。)

【Ⅱ】 共通コマンド

[Ⅱ-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。

5 4 3 2モードの場合、
 出力16ビットの時は MCI-ENG, DIO-5432GP/016, 000000, REV1.01
 出力24ビットの時は MCI-ENG, DIO-5432GP/024, 000000, REV1.01
 出力32ビットの時は MCI-ENG, DIO-5432GP/032, 000000, REV1.01
 7 8 8 A s c i iモードの時は MCI-ENG, UIO-5488GP, 000000, REV1.01
 (7 8 8 B i n a r yの時は応答しません。)
 (5 4 3 2モードでの出力ビット数の選択は取扱説明書を参照して下さい。)

[Ⅱ-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などのデータ出力ポートを全てLowにする
- * 負論理の制御出力ポートを全てHigh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 (S T B)
- * サービス・リクエスト・イネーブル・レジスタ (S R E)
- * スタンダード・イベント・ステータス・レジスタ (E S R)
- * スタンダード・イベント・イネーブル・レジスタ (E S E)
- * 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：4 8 8. 1および4 8 8. 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
 ◎ 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 (LD 1 ~ LD 8) などの端末側への出力信号の状態、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 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。)

[II - 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 (ESR) 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
 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?

[Ⅱ-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。

[Ⅱ－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 [Ⅳ－6] [Ⅳ－7] 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。

【Ⅲ】ステータス報告システム

[Ⅲ-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。

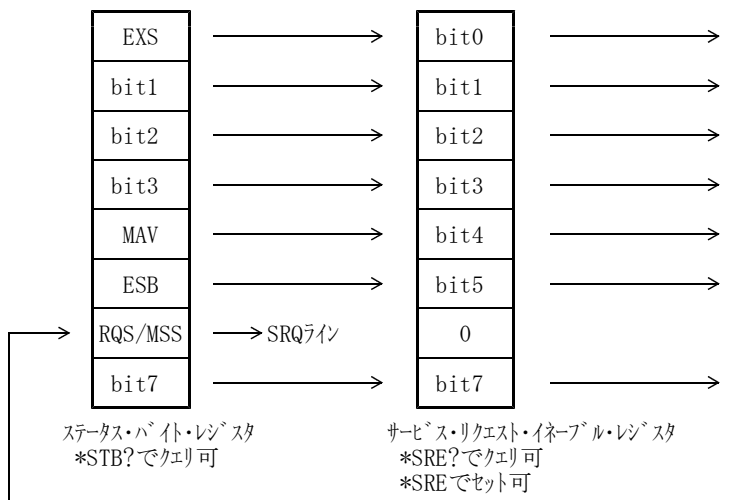
bit 4 : MAV : メッセージ・アベイラブル・ビット
機器のデータ出力の待ち行列が空であるかどうかを示します。
本機のGPIB送信バッファに送信データが有る場合、1にセットされます。

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。

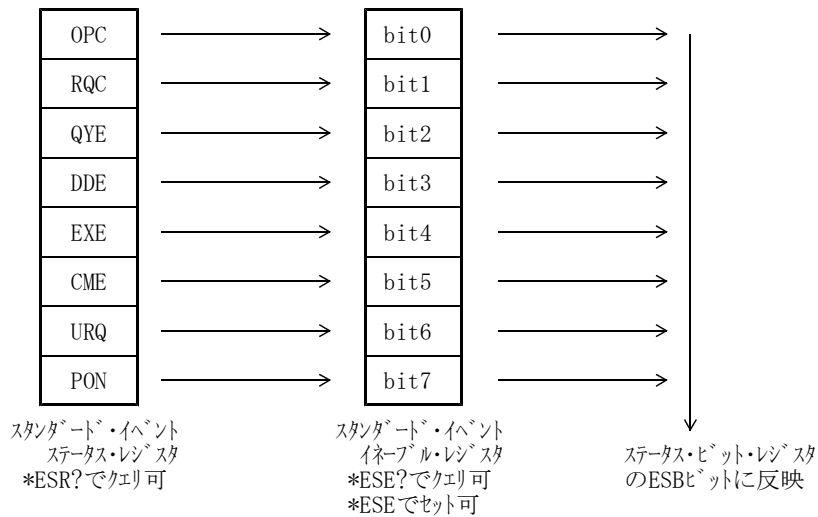
MSS : マスター・ステータス・サマリ
ステータス・ビット・レジスタの他の7ビットの代表。
過去に本機がサービス・リクエストを発生したかどうかを示します。
シリアル・ポールによってRQSビットがクリアされた後もMSSビットはクリアされません。

bit 7 : : 本機においては常に0です。



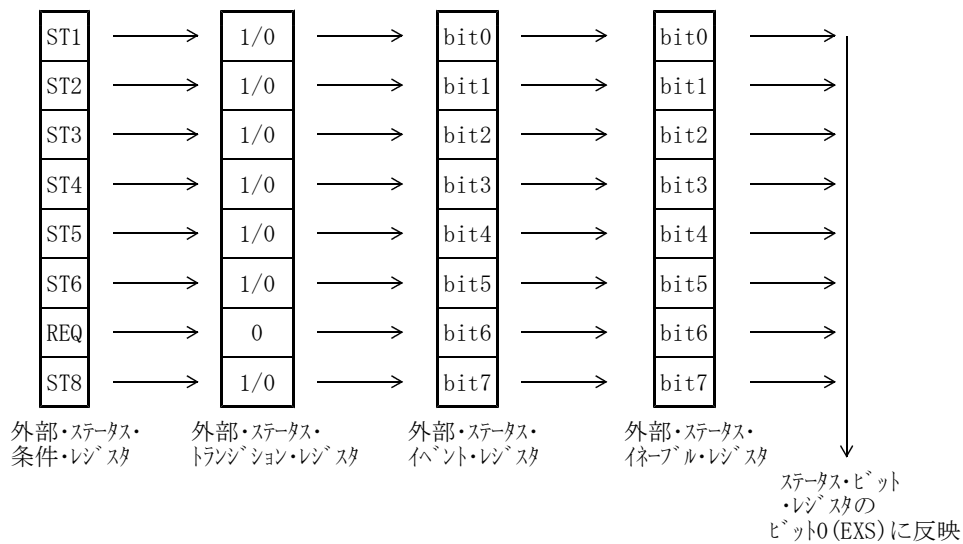
[III-2] スタンド・イベント・ステータス・レジスタ (SESR)

- 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か、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。



[III-3] DIO-5440GP・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 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 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 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 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 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 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 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 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 (0~63、128~191) 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 (0~255) で設定可

(以上のコマンドの説明は、本書 [IV-8] をご参照ください)

[Ⅲ-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。

【IV】SCMCコマンド for DIO-5432GPB

◎ コマンド

当SCMCコマンドはIEEE488.2-1992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すが、
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、EOI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 [I-5] 参照）

[IV-1] 入力端からの入力コマンド (5 4 3 2モード)

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 (BIT0~23) バイト名称 (BYTE0~2) ワード名称 (WORD0~1)		
:FORMat	データ形式	データ形式の指定	DECIMAL
:FORMat?		データ形式の問い合わせ	

ビット名称 : BIT0, BIT1, BIT2, ~ BIT31
 このパラメータが BIT の場合は BIT0 とみなします。
 BIT0, BIT1, ..., BIT7の代わりにTD11, TD12, ..., TD18も使用できます。
 BIT8, BIT9, ..., BIT15の代わりにTD21, TD22, ..., TD28も使用できます。
 BIT16, BIT17, ..., BIT23の代わりにTD31, TD32, ..., TD38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, TDはTD1と同じ, とはみなしません。

バイト名称 : BYTE0, BYTE1, BYTE2
 このパラメータがBYTEのまたはTDの場合はBYTE0とみなします。

ワード名称 : WORD0, WORD1
 このパラメータがWORDの場合はWORD0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。
 バイナリーコードを指定する場合は, CODE と記述します。

[IV-1-1]

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

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、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。
 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/ワード名称」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、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。

応答メッセージのフォーマット

数値

「:INPUT:DATA? ワード名称」コマンドに対する応答の場合、データの値は0~65535の範囲です。
 「: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。
 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」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。

「IV-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/ワード名称」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。
 バイナリー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。

「IV-1-3」

書式 :INPUT:FORMAT?

説明 「:INPUT:DATA ビット名称/バイト名称/ワード名称,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
 LOGICAL
 CODE

[IV-2] 入力端からの入力コマンド (788Asciモード)

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 (BIT0~13), データ数 (0, 1~1000000) バイト名称 (BYTE0~1), データ数 (0, 1~1000000) ワード名称 (WORD0), データ数 (0, 1~1000000)		
:FORMat	データ形式	データ形式の指定	DECIMAL
:FORMat?		データ形式の問い合わせ	

ビット名称 : BIT0, BIT1, BIT2, BIT3, BIT4, BIT5, BIT6, BIT7, BIT8, BIT9, BIT10, BIT11, BIT12, BIT13
 このパラメータが BIT の場合は BIT0 とみなします。
 BIT0, BIT1, ..., BIT7の代わりにTD11, TD12, ..., TD18も使用できます。
 BIT8, BIT9, ..., BIT13の代わりにTD21, TD22, ..., TD26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、TDはTD1と同じ、とはみなしません。

バイト名称 : BYTE0, BYTE1
 このパラメータがBYTEまたはTDの場合はBYTE0とみなします。

ワード名称 : WORD0
 このパラメータがWORDの場合はWORD0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。
 バイナリーコードを指定する場合は、CODE と記述します。

[IV-2-1]

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

説明 ビット名称またはバイト名称、ワード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します。

データ数 :
 指定できるデータ数は0~1000000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、トーカー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、GPIB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GPIBに送出し、終了します。

応答メッセージのフォーマット

「:INPUT:FORMAT データ形式」でデータ形式が
 「CODE」の場合は 不確定長・データ・バイナリ・フォーマット
 「CODE」以外なら 不確定長・データ・ストリング・フォーマット
 で応答メッセージを返します。

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> : 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0~255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,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。
 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。

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? ビット名称」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。

☆「:INPUT:DATA? ワード名称」コマンドに対する応答の場合、

<DAB1>~<DABm>は指定したデータ数の2倍の個数が返ります。
 TD1~TD8の値が先に、その直後にTD21~TD26の値が、交互に送られてきます。

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か端末側信号EODがアクティブになるまでの数のいずれかです。

「IV-2-2」

書式 :INPUT:FORMAT データ形式

説明 「:INPUT:DATA ビット名称/バイト名称/ワード名称,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。
 バイナリー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。

「IV-2-3」

書式 :INPUT:FORMAT?

説明 「:INPUT:DATA ビット名称/バイト名称/ワード名称,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

[IV-3] 出力端への出力コマンド (5 4 3 2モード)

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 (BIT0~31), 出力データ 名称 (BYTE0~3), 出力データ 名称 (WORD0~1), 出力データ	
:OUTput?	ビット名称 (BIT0~31), データ形式 バイト名称 (BYTE0~3), データ形式 バイト名称 (WORD0~1), データ形式	

ビット名称 : BIT0, BIT1, BIT2, BIT3, BIT4, BIT5, BIT6, BIT7, BIT8, ..., BIT30, BIT31
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LD11, LD12, LD13, ..., LD18も使用できます。
BIT8, BIT9, BIT10, ..., BIT15の代わりにLD21, LD22, LD23, ..., LD28も使用できます。
BIT16, BIT17, BIT18, ..., BIT23の代わりにLD31, LD32, LD33, ..., LD38も使用できます。
BIT24, BIT25, BIT26, ..., BIT31の代わりにLD41, LD42, LD43, ..., LD48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, BYTE1, BYTE2, BYTE3
このパラメータが BYTE または LD の場合は BYTE0 とみなします。

ワード名称 : WORD0, WORD1
このパラメータが WORD の場合は WORD0 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。

「IV-3-1」

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

説明 ビット名称またはバイト名称、ワード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:
出力データは
出力先がビットの場合は、各データの値は0～1の範囲でなければなりません
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。
出力先がワードの場合は、データの値は0～65535の範囲でなければなりません。

本機は、このコマンドを受信すると受信データを指定出力端に出力します。

応答 このコマンドに対する応答はありません。

「IV-3-2」

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

説明 ビット名称またはバイト名称、ワード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やワードの場合、
 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。

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。

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。

モニタ先がワードの場合、データは0から65535の範囲の正の整数値です。

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。

[IV-4] 出力端への出力コマンド (788Asciモード)

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), 出力データ列 名称 (BYTE0), 出力データ列	
:OUTput?	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, ..., 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。

バイト名称 : 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。

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。

「IV-4-1」

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

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列を出力させます。

出力データ列 :

出力データ列は複数のデータを並べたものです。並べるデータの数に制限はありません。
出力先がビットの場合は、データ列中の各データの値は0~1の範囲でなければなりません
出力先がバイトの場合は、データ列中の各データの値は0~255の範囲でなければなりません。

本機は、このコマンドを受信しながら受信データを指定出力端に次々に出力します。
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
待ってから受信データを出力端に出力します。この間、GPIB側のハンドシェイクは停止します。)
GPIBのEOIラインがON (アクティブ) になると、一連のデータの受信・出力の動作を終了します。

出力データ列は下記の2通りのフォーマットが使用できます。

不確定長・データ・ストリング・フォーマット 0, <DAS1>, <DAS2>, <DAS3>, < >, <DASm>
0 : 不確定長ストリングを表す、ASCII文字です。
<DASm> : 基数ヘッダが付加されたASCII文字列のデータです。
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。

☆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。

不確定長・データ・バイナリ・フォーマット #0<DAB1><DAB2><DAB3>…<DABm>
 #0 : 不確定長バイナリを表す、ASCII文字です。
 <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。

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

例：LD1 ~ LD8 の出力端の信号状態を下表のようにしたい場合、

LD8	LD7	LD6	LD5	LD4	LD3	LD2	LD1
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>は左表のようなデータにしてください。

☆出力先がビットの場合、(0x00) か (0x01) のデータを入れて下さい。

<DAB>はLD1 ~ LD8 中の指定された1ビットのデータを8ビット・バイナリ・コードで表します。
 指定されたビットのデータをLSBに入れ、他の7ビットは0にします。

例：LD3をLowに、他は変化させたくないような場合、
 (コマンドは「:OUTPUT LD3,出力データ」となります。)

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

左表のようなデータを<DAB>に入れます。

出力結果は下表のようになります。

LD8	LD7	LD6	LD5	LD4	LD3	LD2	LD1
*	*	*	*	*	High	*	*

コマンドで指定したLD3のみが更新され、他のビットは以前の状態を保っています。

- デリミタ：本書 [I-5] で示すデリミタの内、EOIを伴うデリミタを入れてください。
 このデリミタに、NLのみを使用しても<DAB1>~<DABm>中の(0x0A)と
 区別することができませんので注意が必要です。
 下記に、使用できるデリミタを使用した場合の端末側への出力例を記します。
- A：ディップスイッチでLF(NL)+EOIまたはCR+LF(NL)+EOIを選択している場合、
 本機ではLF(NL)+EOIまたはEOIのみを検出しますので、受信データの
 デリミタがLF(NL)+EOIであれば <DAB1>~<DABm>までが出力されます。
 デリミタがEOIであれば <DAB1>~<DABm>までが出力されます。
 デリミタがCR+EOIであれば <DAB1>~<DABm>CRまでが出力されます。
 デリミタがCR+LF(NL)+EOIであれば<DAB1>~<DABm>CRまでが出力されます。
 - B：ディップスイッチでCR+EOIを選択している場合、
 本機ではCR+EOIまたはEOIのみを検出しますので、受信データの
 デリミタがLF(NL)+EOIであれば <DAB1>~<DABm>LFまでが出力されます。
 デリミタがEOIであれば <DAB1>~<DABm>までが出力されます。
 デリミタがCR+EOIであれば <DAB1>~<DABm>までが出力されます。
 デリミタがCR+LF(NL)+EOIであれば<DAB1>~<DABm>CR+LFまでが出力されます。
 - C：ディップスイッチでEOIを選択している場合、
 本機ではEOIのみを検出しますので、受信データの
 デリミタがLF(NL)+EOIであれば <DAB1>~<DABm>LFまでが出力されます。
 デリミタがEOIであれば <DAB1>~<DABm>までが出力されます。
 デリミタがCR+EOIであれば <DAB1>~<DABm>CRまでが出力されます。
 デリミタがCR+LF(NL)+EOIであれば<DAB1>~<DABm>CR+LFまでが出力されます。

応答 このコマンドに対する応答はありません。

「IV-4-2」

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。

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。

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。

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。

[IV-5] バッファメモリ・コマンド (788Asciiモード)

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

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

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

確保した領域はサンプル動作やプレイ動作で使用します。

[IV-5-1]

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メモリ領域の容量をワード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メモリ領域の領域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。

「IV-5-2」

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。

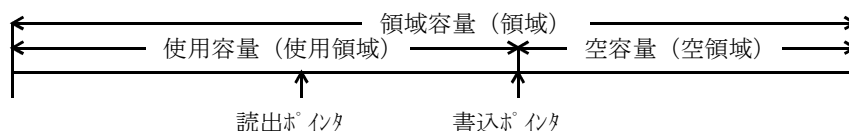
応答 このコマンドの後、トーカーに指定されると下記の応答メッセージを返送します。

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。



「IV-5-3」

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。

「IV-5-4」

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>

書き込むデータはワード単位のデータですので<DABn>と<DABn+1>の2バイトで1ワードになります。書き込むワード数の2倍をバイト数「m」とします。

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) (0x00)か(0x00) (0x01)しかあり得ません。出力先がバイトの場合は、(0x00) (0x00)~(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。

「IV-5-5」

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。

「IV-5-6」

書式 :MEMORY:READ:NEXT? 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。

ワード数: 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ワード単位で指定します。
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、トーカーに指定されると「: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,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。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,ワード数」で指定したワード数の2倍、
またはメモリ領域に入っていたデータのワードの単位での数の2倍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。

「IV-5-7」

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。

「IV-5-8」

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

「IV-5-9」

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 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、512ワードです。

[IV-6] 入力端データのバッファリング・コマンド (788Asciモード)

SAMPLEコマンドセット

コマンド	パラメータ	備考	初期値
:SAMPle			
:REPeat	ワード名称(WORD0), 回数(0, 1~1000000) バイト名称(BYTE0), 回数(0, 1~1000000) ビット名称(BIT0~13), 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?	ワード名称(WORD0) バイト名称(BYTE0) ビット名称(BIT0~13)	繰り返し回数の問い合わせ 応答は0~1000000	
:ASSign	ワード名称(WORD0), ブロック番号, データ数 バイト名称(BYTE0), ブロック番号, データ数 ビット名称(BIT0~13), 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?	ワード名称(WORD0) バイト名称(BYTE0) ビット名称(BIT0~13)	サンプル入出力の問い合わせ	
[:STARt]	ワード名称(WORD0), 指令 バイト名称(BYTE0), 指令 ビット名称(BIT0~13), 指令	指令は下記のいずれか。 ENAbLe, DISAbLe SAMPLE動作の状態を返す。	
:STATe?	ワード名称(WORD0) バイト名称(BYTE0) ビット名称(BIT0~13)	下記のいずれか。 IDLE, STANDBY, RUNNING	IDLE

サンプル動作が 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

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。

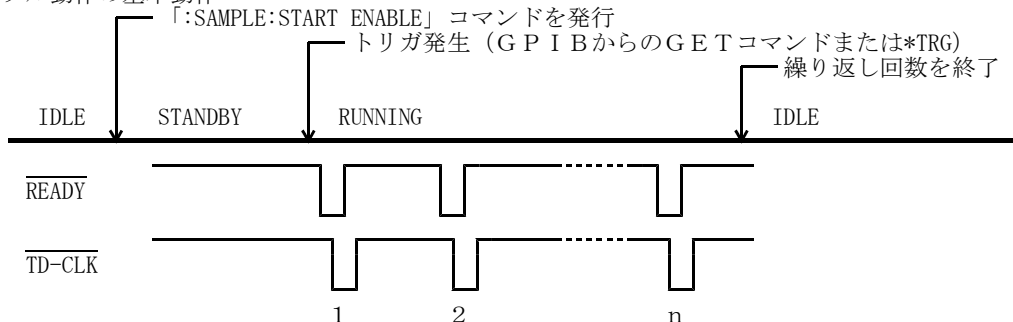
:MEMORY:ASSIGN ブロック番号, ワード数
:MEMORY:WRITE:INITIALIZE ブロック番号
:MEMORY:WRITE:NEXT ブロック番号, データ列
:MEMORY:READ:INITIALIZE ブロック番号
:MEMORY:READ:NEXT? ブロック番号, ワード数

ビット名称: BIT0, BIT1, BIT2, BIT3, BIT4, BIT5, BIT6, BIT7, ..., BIT13
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TD11, TD12, TD13, ..., TD18も使用できます。
BIT8, BIT9, BIT10, ..., BIT13の代わりにTD21, TD22, TD23, ..., TD26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, TD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。

ワード名称: WORD0
このパラメータが WORD の場合は WORD0 とみなします。

サンプル動作の基本動作



上記TD-CLKのタイミングで、指定入力端からデータを読み取り、メモリ領域に格納します。
読み取って格納するデータの数nは原則として
 $n = \text{「:SAMPLE:ASSIGN」で指定したデータ数} \times \text{「:SAMPLE:REPEAT」で指定した回数}$
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ポインタと読出ポインタを初期化して、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ポインタ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。

「IV-6-1」

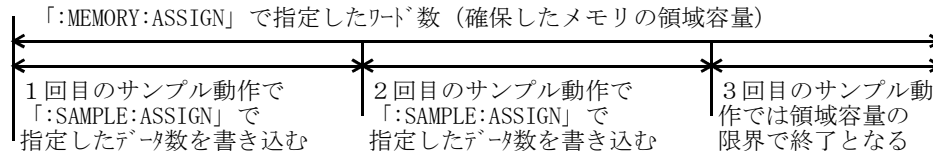
書式 :SAMPLE:REPEAT ビット名称,回数
 :SAMPLE:REPEAT バイト名称,回数
 :SAMPLE:REPEAT ワード名称,回数

説明 ビット名称、またはバイト名称、ワード名称で指定する入力端の信号をサンプルする繰り返し回数を指定します。

回数 : 0, 1, 2, 3, ..., 1000000

1 ~ 1 0 0 0 0 0 を指定すると一回のサンプル動作を指定した回数、繰り返します。
 一回の動作とは、「:SAMPLE:ASSIGN ビット名称/バイト名称/ワード名称,ブロック番号,データ数」で指定したデータの数をサンプルすることを言います。
 0 を指定すると、*ABORT,または*RSTを受信するか、「MEMORY:ASSIGN ブロック番号,ワード数」で指定したメモリの領域容量がいっぱいになるまで繰り返します。

下図に「:SAMPLE:REPEAT ビット名称/バイト名称,3」を指定した場合のメモリの使用状況を示します。



ただし、「:SAMPLE:ASSIGN」で指定したデータ数に満たないうちにEOD信号がアクティブになった場合はその時のデータの書き込みで終了とし、IDLE状態になります。

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/バイト名称/ワード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。

「IV-6-2」

書式 :SAMPLE:REPEAT? ビット名称
 :SAMPLE:REPEAT? バイト名称
 :SAMPLE:REPEAT? ワード名称

説明 ビット名称、またはバイト名称、ワード名称で指定する入力端の信号をサンプルする繰り返し回数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。

「IV-6-3」

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
 :SAMPLE:ASSIGN ワード名称,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、ワード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、一回の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,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 : 0, 1 ~ メモリ領域のメモリ領域容量以内

一回の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,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/ワード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/ワード名称に、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/ワード名称に、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。

「IV-6-4」

書式 :SAMPLE:ASSIGN? ビット名称
 :SAMPLE:ASSIGN? バイト名称
 :SAMPLE:ASSIGN? ワード名称

説明 ビット名称、またはバイト名称、ワード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、一回の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/ワード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。

「IV-6-5」

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

説明 ビット名称、またはバイト名称、ワード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/ワード名称, 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領域に取り込みます。

取り込むタイミングはREADY信号がLowの時のTD-CLKの立ち下がりの時です。

「:SAMPLE:ASSIGN」コマンドのデータ数で指定した数のデータの取り込みを「:SAMPLE: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。

「:SAMPLE:START ビット名称/バイト名称/ワード名称, 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データ数に満たない内にEODがアクティブになった場合や、確保されたメモリ領域の領域容量を越えた場合も、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/ワード名称, ENABLE」を受信したとき、指定と同じビット名称/バイト名称/ワード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/ワード名称, DISABLE」を受信したとき、指定と同じビット名称/バイト名称/ワード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/ワード名称に対する

「:SAMPLE:START ビット名称/バイト名称/ワード名称, 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/ワード名称, ENABLE」を受信したとき、指定されたビット名称/バイト名称/ワード名称に割り当てられたメモリ領域に対するブレイ動作が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/ワード名称, ENABLE」を受信したとき、指定されたビット名称/バイト名称/ワード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。

「IV-6-6」

書式 :SAMPLE:STATE? ビット名称
 :SAMPLE:STATE? バイト名称
 :SAMPLE:STATE? ワード名称

説明 ビット名称、またはバイト名称、ワード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。

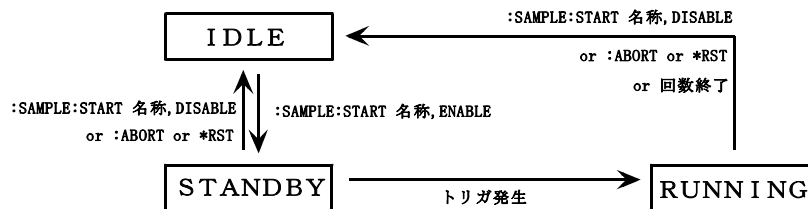
応答 このコマンドの後、トーカーに指定されると下記のいずれかの応答メッセージを返送します。

IDEL
 STANDBY
 RUNNING

IDLE状態：「:SAMPLE:START ビット名称/バイト名称/ワード名称, ENABLE」コマンドを受信していません。
 または、指定された回数の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/ワード名称, 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/ワード名称, 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/ワード名称, 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、書込ホインタ、読出ホインタがリセットされます。



[IV-7] バッファリングされたデータの出力コマンド (788Asciiモード)

PLAYコマンドセット

コマンド	パラメータ	備考	初期値
:PLAY			
:REPeat	バイト名称 (BYTE0), 回数 (0, 1~1000000)	繰り返し回数の設定 0を指定すると無限 繰り返し回数の問合わせ 応答は0~1000000	1
:REPeat?	ビット名称 (BIT0~7), 回数 (0, 1~1000000)		
:ASSIgn	バイト名称 (BYTE0), 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?	ビット名称 (BIT0~7), ブロック番号, データ数		
[:STARt]	バイト名称 (BYTE0), 指令	プレイ入出力の問合わせ	
:STATe?	ビット名称 (BIT0~7), 指令		
	バイト名称 (BYTE0)	指令は下記のいずれか。 ENable, DISable PLAY動作の状態を返す。 下記のいずれか。 IDLE, STANDBY, RUNNING	IDLE
	ビット名称 (BIT0~7)		

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

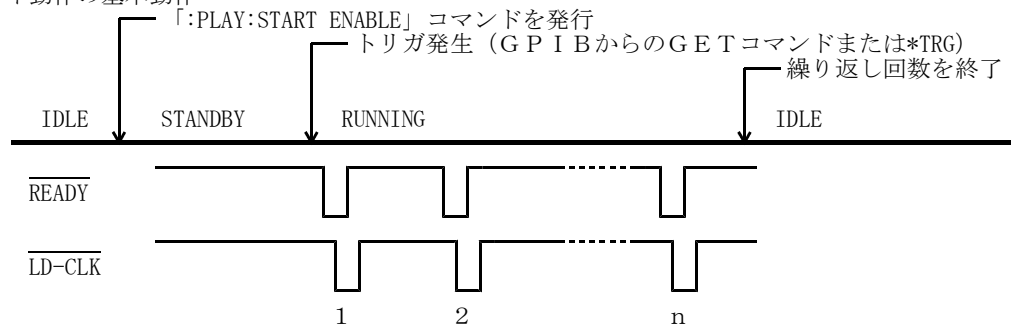
プレイ動作が RUNNING状態にある場合は同一ブロック番号のメモリ領域に対して以下のことを行うことができません。

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

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LD11, LD12, LD13, , , , LD18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
 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



上記LD-CLKのタイミングで、メモリ領域から指定出力端へデータを出力します。
 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。

「IV-7-1」

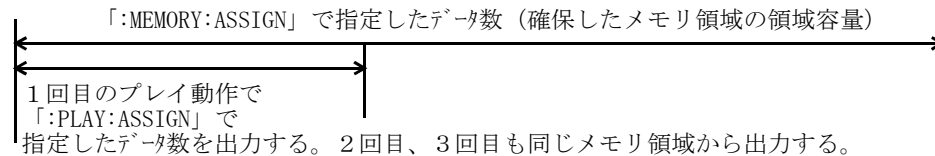
書式 :PLAY:REPEAT ビット名称,回数
 :PLAY:REPEAT バイト名称,回数

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 ビット名称/バイト名称,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「: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が「: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/バイト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。

「IV-7-2」

書式 :PLAY:REPEAT? ビット名称
 :PLAY:REPEAT? 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。

「IV-7-3」

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,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、メモリ領域の領域容量以内

一回のプレイ動作で出力するデータ数を指定します。
 0を指定すると、プレイデータ源とデータ出力端の割り当てを解除（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。

「IV-7-4」

書式 :PLAY:ASSIGN? ビット名称
:PLAY:ASSIGN? バイオ名称

説明 ビット名称、またはバイオ名称で指定する出力端へ信号をプレイ出力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、指定されたビット名称/バイオ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。

「IV-7-5」

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

説明 ビット名称、またはバイオ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/バイオ名称, 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。
出力するタイミングはREADY信号がLowの時のLD-CLKの立ち下がりの時です。
「: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、行うと終了し、IDLE状態になります。

「:PLAY:START ビット名称/バイオ名称, 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/バイオ名称, ENABLE」を受信したとき、指定と同一のビット名称/バイオ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/バイオ名称, DISABLE」を受信したとき、指定と同一のビット名称/バイオ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オ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 バイオ名称, ENABLE」を受信したとき、指定のバイオ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/バイオ名称に対する「:PLAY:START ビット名称/バイオ名称, ENABLE」を受信すると「実行エラー」になります。

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

「IV-7-6」

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。

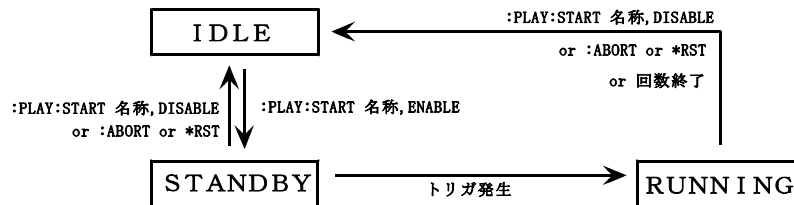
応答 このコマンドの後、トーカーに指定されると下記のいずれかの応答メッセージを返送します。

IDLE
 STANDBY
 RUNNING

IDLE状態：「:PLAY:START 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、指定された回数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、プレイ動作を行っています。



[IV-8] ステータス操作コマンド (5432モード、788Asci iモード)

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?		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
:CONDITION?		条件レジスタをクエリする

[IV-8-1]

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。

[IV-8-2]

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、GP1B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。

[IV-8-3]

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 (\overline{REQ} 信号) はセットできず、0に固定のためです。

数値

「IV-8-4」

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

「IV-8-5」

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

「IV-8-6」

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

[IV-9] アボート・コマンド (788Asciiモード)

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。

「IV-9-1」

書式 :ABORT

説明 トリガ・システムをアイドル・ステートにし、プレイ動作やサンプル動作の状態を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。