端末組込用ＧＰＩＢアダプタ

ＵＩＯ－７８８ＧＰＢ
ＵＩＯ－７８８ＧＰＢ－Ｒｏ

コマンド説明書
for ＡＳＣＩＩモード

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/

第５版　2011年12月05日

【Ｉ】概要

本「コマンド説明書」ではＵＩＯ－７８８ＧＰＢやＵＩＯ－７８８ＧＰＢ－Ｒｏ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。両機種とも性能・機能は同じですので本書の中では両機種を「本機」または「ＵＩＯ－７８８ＧＰＢ」と表現しています。
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。

ＵＩＯ－７８８ＧＰＢのＡＳＣＩＩモードの操作コマンドやその機能は、ＩＥＥＥ－Ｓｔｄ４８８．２－１９９２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「ＵＩＯ－７８８ＧＰＢ」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。
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－１］［Ⅳ－２］［Ⅳ－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－３］～［Ⅳ－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【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、本機のバイナリー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。

ＵＩＯ－７８８ＧＰＢ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）

ＵＩＯ－７８８ＧＰＢ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。

ＵＩＯ－７８８ＧＰＢ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－２－１］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、シリアルポールを行って
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。）

【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＞，＜ファームウェアのリビジョン＞を表す、
下記の文字列を返します。
MCI-ENG,UIO-788GP,000000,REV1.00

［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。
＊　ＬＤ１～ＬＤ８の出力をＬｏｗ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）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。）

［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”２５５”までの値を１０進数または１６、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”２５５”の範囲の１０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”２５５”の範囲の１０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”２５５”までの値を１０進または１６、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”６３”、”１２８”から”１９１”の範囲の１０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”２５５”の範囲の１０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－ＥＥＥ４８８．１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－４］［Ⅳ－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。

【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

ｂｉｔ０：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

ｂｉｔ１：　　：本機においては常に０です。

ｂｉｔ２：　　：本機においては常に０です。

ｂｉｔ３：　　：本機においては常に０です。

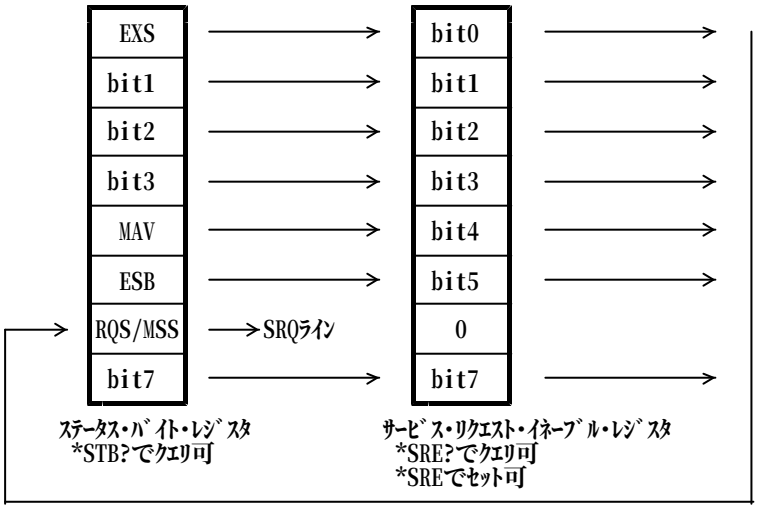
ｂｉｔ４：ＭＡＶ：メッセージ・アベイラブル・ビット
機器のデータ出力の待ち行列が空であるかどうかを示します。
本機のＧＰＩＢ送信バッファに送信データが有る場合、１にセットされます。

ｂｉｔ５：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。

ｂｉｔ６：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。

ｂｉｔ７：　　：本機においては常に０です。



［Ⅲ－２］スタンダード・イベント・ステータス・レジスタ（ＳＥＳＲ）

ｂｉｔ０：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。

ｂｉｔ１：ＲＱＣ：リクエスト・コントロール
本機においては常に０です。

ｂｉｔ２：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か、オーバーフローしている場合です。

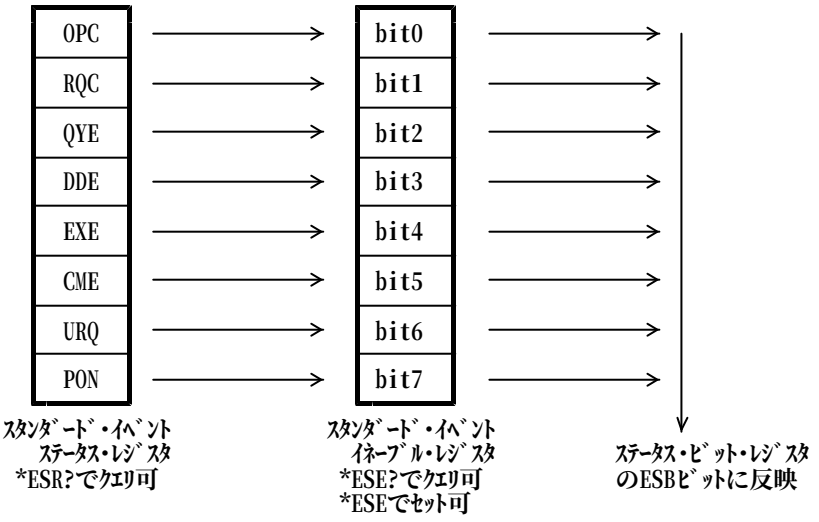
ｂｉｔ３：ＤＤＥ：機器定義エラー
本機が電源投入された場合、プログラムＲＯＭのサムチェックとシステムワークＲＡＭの
リードライトチェックを行い、エラーが発生した場合１になります。

ｂｉｔ４：ＥＸＥ：実行エラー
本機がコマンド実行時にエラーを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。

ｂｉｔ５：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。

ｂｉｔ６：ＵＲＱ：ユーザ・リクエスト
本機においては常に０です。

ｂｉｔ７：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。



［Ⅲ－３］ＵＩＯ－７８８ＧＰ・外部・ステータス・レジスタ

ｂｉｔ０：ＳＴ１：外部端末側ステータス入力$\overline{\text{ＳＴ１}}$の状態
外部端末側ステータス入力信号$\overline{\text{ＳＴ１}}$の状態を表します。

ｂｉｔ１：ＳＴ２：外部端末側ステータス入力$\overline{\text{ＳＴ２}}$の状態
外部端末側ステータス入力信号$\overline{\text{ＳＴ２}}$の状態を表します。

ｂｉｔ２：ＳＴ３：外部端末側ステータス入力$\overline{\text{ＳＴ３}}$の状態
外部端末側ステータス入力信号$\overline{\text{ＳＴ３}}$の状態を表します。

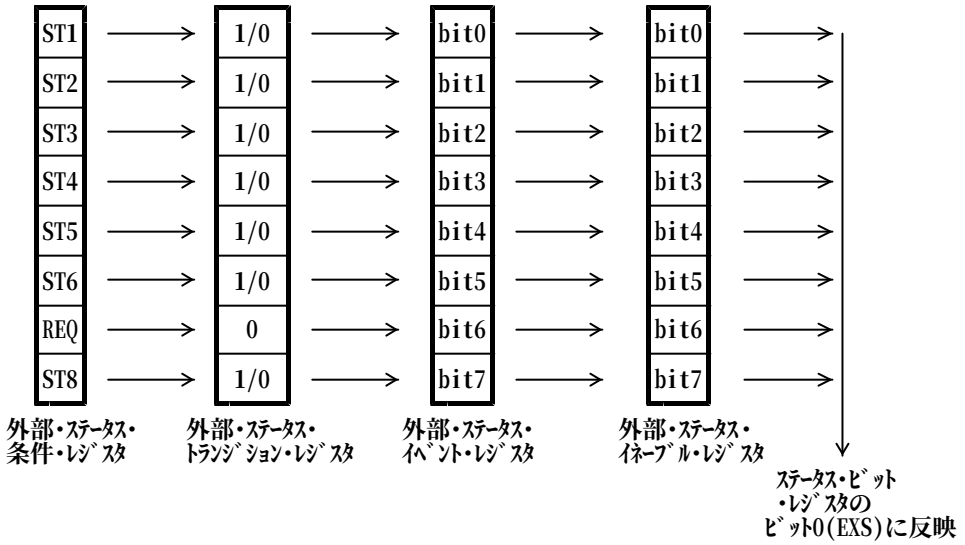
ｂｉｔ３：ＳＴ４：外部端末側ステータス入力$\overline{\text{ＳＴ４}}$の状態
外部端末側ステータス入力信号$\overline{\text{ＳＴ４}}$の状態を表します。

ｂｉｔ４：ＳＴ５：外部端末側ステータス入力$\overline{\text{ＳＴ５}}$の状態
外部端末側ステータス入力信号$\overline{\text{ＳＴ５}}$の状態を表します。

ｂｉｔ５：ＳＴ６：外部端末側ステータス入力$\overline{\text{ＳＴ６}}$の状態
外部端末側ステータス入力信号$\overline{\text{ＳＴ６}}$の状態を表します。

ｂｉｔ６：ＲＥＱ：外部端末側リクエスト入力$\overline{\text{ＲＥＱ}}$の状態
外部端末側リクエスト入力信号$\overline{\text{ＲＥＱ}}$の状態を表します。

ｂｉｔ７：ＳＴ８：外部端末側ステータス入力$\overline{\text{ＳＴ８}}$の状態
外部端末側ステータス入力信号$\overline{\text{ＳＴ８}}$の状態を表します。



外部・ｽﾃｰﾀｽ・条件・ﾚｼﾞｽﾀ
:STATUS:EXTERNAL:CONDITION? でクエリ可

外部・ｽﾃｰﾀｽ・ﾄﾗﾝｼﾞｼｮﾝ・ﾚｼﾞｽﾀ
: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・ｲﾍﾞﾝﾄ・ﾚｼﾞｽﾀ
:STATUS:EXTERNAL:EVENT? でクエリ可

外部・ｽﾃｰﾀｽ・ｲﾈｰﾌﾞﾙ・ﾚｼﾞｽﾀ
: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－６］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。

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ステータス・バイト・レジスタ	RQS/MSS							
	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　ＵＩＯ－７８８ＧＰＢ

◎ 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。

◎ ディスクリートパラメータ
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

INPUTコマンドセット

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

ﾋﾞｯﾄ名称：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、TDはTD1と同じ、とはみなしません。

ﾊﾞｲﾄ名称：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。

ﾃﾞｰﾀ形式：
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－１－１」

書式　:INPUT[:DATA]? ﾋﾞｯﾄ名称[,ﾃﾞｰﾀ数]
:INPUT[:DATA]? ﾊﾞｲﾄ名称[,ﾃﾞｰﾀ数]

説明　ﾋﾞｯﾄ名称またはﾊﾞｲﾄ名称で指定する入力端の信号を、ﾃﾞ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。
ﾃﾞ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 ﾃﾞｰﾀ形式」で指定されたフォーマットです。
「:INPUT[:DATA] ﾊﾞｲﾄ名称」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。

ﾃﾞｰﾀ数：
指定できるﾃﾞｰﾀ数は０～１０００００００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になります。

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、ＧＰＩＢ側のハンドシェイクは停止します。）
指定されたﾃﾞｰﾀ数を満たすか、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

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>：指定されたﾃﾞｰﾀ形式で表した数値のデータです。

☆「:INPUT:DATA? ﾊﾞｲﾄ名称,ﾃﾞｰﾀ数」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。
指定ﾃﾞｰﾀ形式が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? ﾋﾞｯﾄ名称,ﾃﾞｰﾀ数」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。
指定ﾃﾞｰﾀ形式が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? ﾊﾞｲﾄ名称」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? ﾋﾞｯﾄ名称」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、指定されたﾃﾞｰﾀ数か
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－１－２」

書式　:INPUT:FORMAT ﾃﾞｰﾀ形式

説明　「:INPUT:DATA ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,ﾃﾞｰﾀ数」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。

ﾃﾞｰﾀ形式：
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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,ﾃﾞｰﾀ数」コマンドに対する応答メッセージのﾃﾞ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］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 :OUTput?	ﾋﾞｯﾄ名称(BIT0～7),出力ﾃﾞｰﾀ列 名称(BYTE0),出力ﾃﾞｰﾀ列 ﾋﾞｯﾄ名称(BIT0～7),ﾃﾞｰﾀ形式 ﾊﾞｲﾄ名称(BYTE0),ﾃﾞｰﾀ形式	

ﾋﾞｯﾄ名称：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。

ﾊﾞｲﾄ名称：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。

ﾃﾞｰﾀ形式：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－２－１」

書式　:OUTPUT ﾋﾞｯﾄ名称,出力ﾃﾞｰﾀ列
:OUTPUT ﾊﾞｲﾄ名称,出力ﾃﾞｰﾀ列

説明　ﾋﾞｯﾄ名称またはﾊﾞｲﾄ名称で指定する出力端へ出力ﾃﾞｰﾀ列を出力させます。

出力ﾃﾞｰﾀ列：
出力ﾃﾞ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。

出力ﾃﾞｰﾀ列は下記の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,出力ﾃﾞｰﾀ」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－２－２」

書式　:OUTPUT? ﾋﾞｯﾄ名称[,ﾃﾞｰﾀ形式]
　　:OUTPUT? ﾊﾞｲﾄ名称[,ﾃﾞｰﾀ形式]

説明　ﾋﾞｯﾄ名称またはﾊﾞｲﾄ名称で指定する出力端の信号をモニタし、ﾃﾞ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。ﾃﾞ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。

ﾃﾞｰﾀ形式：
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、指定されたﾃﾞｰﾀ形式の
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。
ﾃﾞｰﾀ形式が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。
ﾃﾞｰﾀ形式が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－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	ﾌﾞﾛｯｸ番号(0～1),ﾊﾞｲﾄ数	メモリ領域容量を指定確保する	確保されていない
:ASSign?	ﾌﾞﾛｯｸ番号(0～1)	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	
:WRITe			
:INITialize	ﾌﾞﾛｯｸ番号(0～1)	指定領域の書込ﾎﾟｲﾝﾀを初期化	
[:NEXT]	ﾌﾞﾛｯｸ番号(0～1),ﾃﾞｰﾀ列	書込ﾎﾟｲﾝﾄから書込み、 書込ﾎﾟｲﾝﾄを次へ移す。	
:READ			
:INITialize	ﾌﾞﾛｯｸ番号(0～1)	指定領域の読出ﾎﾟｲﾝﾀを初期化	
[:NEXT]?	ﾌﾞﾛｯｸ番号(0～1),ﾊﾞｲﾄ数	読出ﾎﾟｲﾝﾄから読出し、 読出ﾎﾟｲﾝﾀを次へ移す。	
:FORMat	ﾌﾞﾛｯｸ番号(0～1),ﾃﾞｰﾀ形式	読出ﾃﾞｰﾀ形式を指定する	DECIMAL
:FORMat?	ﾌﾞﾛｯｸ番号(0～1)	読出ﾃﾞｰﾀ形式の問い合わせ	
:MEMory?		メモリの情報の問い合わせ 総領域容量,総空容量を得る	

サンプル動作やプレイ動作が、ＳＴＡＮＤＢＹ状態にある場合は同一ﾌﾞ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          ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数
```

サンプル動作やプレイ動作が、ＲＵＮＮＩＮＧ状態にある場合は同一ﾌﾞ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          ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数
:MEMORY:WRITE:INITIALIZE ﾌﾞﾛｯｸ番号
:MEMORY:WRITE:NEXT      ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ列
:MEMORY:READ:INITIALIZE  ﾌﾞﾛｯｸ番号
:MEMORY:READ:NEXT?      ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数
```

ﾌﾞﾛｯｸ番号：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－３－１」

書式　:MEMORY:ASSIGN ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数

説明　ﾌﾞﾛｯｸ番号で指定するメモリ領域の容量をﾊﾞｲﾄ数で指定確保します。
　　読出ﾎﾟｲﾝﾀと書込ﾎﾟｲﾝﾀは、この領域の先頭に初期化されます。

　　ﾊﾞｲﾄ数：０または、１以上、メモリの総空容量以内
　　ﾌﾞ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、指定ﾌﾞ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 ﾌﾞﾛｯｸ番号,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ﾌﾞ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ﾌﾞ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。

　　このコマンドで指定するﾌﾞ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、ﾃﾞｰﾀ数が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－３－２」

書式　:MEMORY:ASSIGN? ﾌﾞﾛｯｸ番号

説明　ﾌﾞﾛｯｸ番号で指定する領域の情報を問い合わせます。

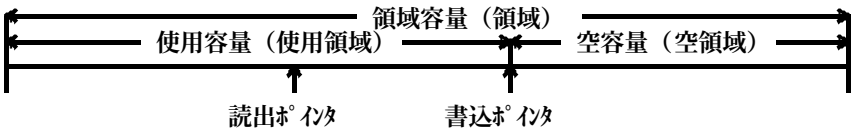
応答　このコマンドの後、トーカに指定されると下記の応答メッセージを返送します。

領域容量,使用容量,空容量

領域容量：「:MEMORY:ASSIGN ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数」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 ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ列」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。



「Ⅳ－３－３」

書式　:MEMORY:WRITE:INITIALIZE ﾌﾞﾛｯｸ番号

説明　ﾌﾞﾛｯｸ番号で指定する領域への書込ﾎﾟ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、読出ﾎﾟ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
同一ﾌﾞ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－３－４」

書式　:MEMORY:WRITE:NEXT ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ列

説明　ﾌﾞ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、書込ﾎﾟｲﾝﾀは
最終書込データの格納された次をポイントします。
読出ﾎﾟｲﾝﾀは変化しません。
ﾃﾞ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
同一ﾌﾞ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－３－５」

書式　:MEMORY:READ:INITIALIZE ﾌﾞﾛｯｸ番号

説明　ﾌﾞﾛｯｸ番号で指定する領域からの読出ﾎﾟｲﾝﾀを初期化します。
書込ﾎﾟ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
同一ﾌﾞ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－３－６」

書式　:MEMORY:READ:NEXT? ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数

説明　ﾌﾞ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、読出ﾎﾟ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。
書込ﾎﾟｲﾝﾀは変化しません。

ﾊﾞｲﾄ数：０または、１～１０００００００
ﾌﾞ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。
指定したﾌﾞ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、指定したﾌﾞ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、トーカに指定されると「:MEMORY:READ:FORMAT ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ形式」で
指定されているﾃﾞ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。

ﾃﾞｰﾀ形式を 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。数値の表現はﾃﾞｰﾀ形式に従います。
「:MEMORY:READ:NEXT ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数」で指定したﾊﾞｲﾄ数、
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。数値の表現はﾃﾞｰﾀ形式に従います。
各<DASm>は，で区切られています。

ﾃﾞｰﾀ形式を CODE と指定してある場合は以下のようになります。
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 ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数」で指定したﾊﾞｲﾄ数、
またはメモリ領域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
同一ﾌﾞ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－３－７」

書式　:MEMORY:READ:FORMAT ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ形式

説明　「:MEMORY:READ:NEXT? ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数」コマンドに対する応答メッセージのﾃﾞｰﾀ形式を
指定します。

ﾃﾞｰﾀ形式：
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－３－８」

書式　:MEMORY:READ:FORMAT? ﾌﾞﾛｯｸ番号

説明　「:MEMORY:READ:NEXT ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数」コマンドに対する応答メッセージのﾃﾞ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－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 ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数」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 ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数」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、１０２４バイトです。

［Ⅳ－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			
:REPeat	ﾊﾞｲﾄ名称(BYTE0),回数(0,1～1000000) ﾋﾞｯﾄ名称(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?	ﾊﾞｲﾄ名称(BYTE0) ﾋﾞｯﾄ名称(BIT0～7)	繰り返し回数の問合わせ 応答は０～１０００００００	
:ASSign	ﾊﾞｲﾄ名称(BYTE0),ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ数 ﾋﾞｯﾄ名称(BIT0～7),ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ数	サンプル入出力の割当て	割り当てなし
:ASSign?	ﾊﾞｲﾄ名称(BYTE0) ﾋﾞｯﾄ名称(BIT0～7)	サンプル入出力の問合せ	
[:STARt]	ﾊﾞｲﾄ名称(BYTE0),指令 ﾋﾞｯﾄ名称(BIT0～7),指令	指令は下記のいずれか。 ENAble,DISable	
:STATe?	ﾊﾞｲﾄ名称(BYTE0) ﾋﾞｯﾄ名称(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状態にある場合は同一ﾌﾞ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	ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数

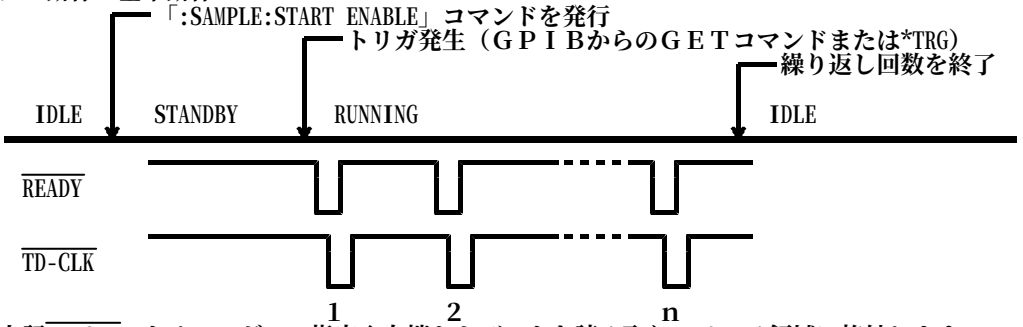
サンプル動作が ＲＵＮＮＩＮＧ状態にある場合は同一ﾌﾞﾛｯｸ番号のメモリ領域に対して以下のことを行うことができません。

:MEMORY:ASSIGN	ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数
:MEMORY:WRITE:INITIALIZE	ﾌﾞﾛｯｸ番号
:MEMORY:WRITE:NEXT	ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ列
:MEMORY:READ:INITIALIZE	ﾌﾞﾛｯｸ番号
:MEMORY:READ:NEXT?	ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数

ﾋﾞｯﾄ名称：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、TDはTD1と同じ、とはみなしません。

ﾊﾞｲﾄ名称：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。

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 ＝ 「:SAMPLE:ASSIGN」で指定したﾃﾞｰﾀ数 × 「: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ﾎﾟｲﾝﾀと読出ﾎﾟ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ﾎﾟ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。

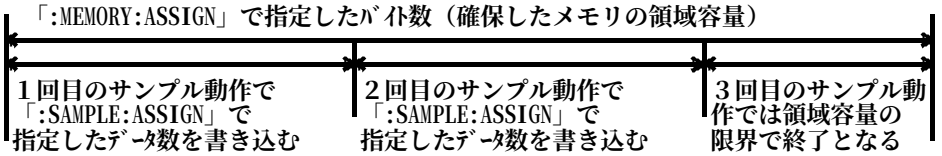
「Ⅳ－４－１」

書式　:SAMPLE:REPEAT ﾋﾞｯﾄ名称,回数
:SAMPLE:REPEAT ﾊﾞｲﾄ名称,回数

説明　ﾋﾞｯﾄ名称、またはﾊﾞｲﾄ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
１～１０００００００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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ数」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 ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数」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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,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」で指定したﾃﾞｰﾀ数に満たないうちに$\overline{EOD}$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ﾋﾞｯﾄ名称／ﾊﾞ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。

「Ⅳ－４－２」

書式　:SAMPLE:REPEAT? ﾋﾞｯﾄ名称
:SAMPLE:REPEAT? ﾊﾞｲﾄ名称

説明　ﾋﾞｯﾄ名称、またはﾊﾞｲﾄ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－４－３」

書式　:SAMPLE:ASSIGN ﾋﾞｯﾄ名称,ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ数
:SAMPLE:ASSIGN ﾊﾞｲﾄ名称,ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ数

説明　ﾋﾞｯﾄ名称、またはﾊﾞ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、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。

ﾌﾞﾛｯｸ番号：0,1
あらかじめ、「MEMORY:ASSIGN ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数」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。

ﾃﾞｰﾀ数：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 ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数」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ﾋﾞｯﾄ名称／ﾊﾞ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ﾋﾞｯﾄ名称／ﾊﾞｲﾄ名称に、ﾌﾞ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、ﾃﾞｰﾀ数が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ﾋﾞｯﾄ名称／ﾊﾞｲﾄ名称に、ﾌﾞ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－４－４」

書式　:SAMPLE:ASSIGN? ﾋﾞｯﾄ名称
:SAMPLE:ASSIGN? ﾊﾞｲﾄ名称

説明　ﾋﾞｯﾄ名称、またはﾊﾞ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、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、トーカに指定されると下記の応答メッセージを返送します。

ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ数

応答メッセージのﾌﾞﾛｯｸ番号が－１、ﾃﾞｰﾀ数が０の場合は、指定されたﾋﾞｯﾄ名称／ﾊﾞｲﾄ名称と
指定されたﾌﾞ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－４－５」

書式　:SAMPLE[:START] ﾋﾞｯﾄ名称,指令
:SAMPLE[:START] ﾊﾞｲﾄ名称,指令

説明　ﾋﾞｯﾄ名称、またはﾊﾞ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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,ENABLE」の後のトリガ発生でサンプル動作を開始します。
「:SAMPLE:ASSIGN」コマンドのﾃﾞｰﾀ数で指定した数のデータを、ﾌﾞ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$\overline{\text{READY}}$信号がLowの時の$\overline{\text{TD-CLK}}$の立ち下がりの時です。
「:SAMPLE:ASSIGN」コマンドのﾃﾞ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、行うと終了し、ＩＤＬＥ状態になります。

「:SAMPLE:START ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,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ﾃﾞ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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,ENABLE」を受信したとき、指定と同じﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称に
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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,DISABLE」を受信したとき、指定と同じﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称に
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ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称に対する
「:SAMPLE:START ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,ENABLE」を受信すると「実行エラー」になります。

「:SAMPLE:START ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,ENABLE」を受信したとき、指定されたﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称に
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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,ENABLE」を受信したとき、指定されたﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称に
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。

「Ⅳ－４－６」

書式　:SAMPLE:STATE? ﾋﾞｯﾄ名称
　　:SAMPLE:STATE? ﾊﾞｲﾄ名称

説明　ﾋﾞｯﾄ名称、またはﾊﾞ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。

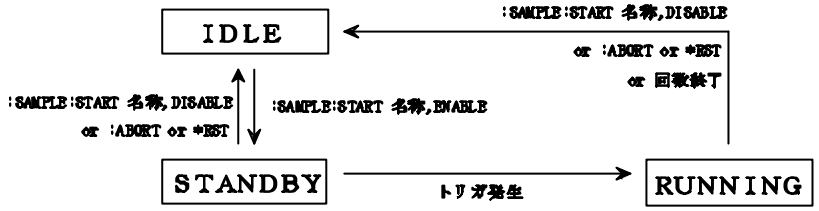
応答　このコマンドの後、トーカに指定されると下記のいずれかの応答メッセージを返送します。

IDEL
STANDBY
RUNNING

ＩＤＬＥ状態：「:SAMPLE:START ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称 ENABLE」コマンドを受信していません。
または、指定された回数のサンプル動作をすべて終了しています。
または、「:SAMPLE:START ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称 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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称 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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称 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、
書込ﾎﾟｲﾝﾀ、読出ﾎﾟｲﾝﾀがリセットされます。



［Ⅳ－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			
:REPeat	ﾊﾞｲﾄ名称(BYTE0),回数(0,1～1000000) ﾋﾞｯﾄ名称(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?	ﾊﾞｲﾄ名称(BYTE0) ﾋﾞｯﾄ名称(BIT0～7)	繰り返し回数の問合わせ 応答は０～１０００００００	
:ASSign	ﾊﾞｲﾄ名称(BYTE0),ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ数 ﾋﾞｯﾄ名称(BIT0～7),ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ数	プレイ入出力の割り当て	割り当てなし
:ASSign?	ﾊﾞｲﾄ名称(BYTE0) ﾋﾞｯﾄ名称(BIT0～7)	プレイ入出力の問合わせ	
[:STARt]	ﾊﾞｲﾄ名称(BYTE0),指令 ﾋﾞｯﾄ名称(BIT0～7),指令	指令は下記のいずれか。 ENAble,DISable	
:STATe?	ﾊﾞｲﾄ名称(BYTE0) ﾋﾞｯﾄ名称(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ﾌﾞ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	ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数

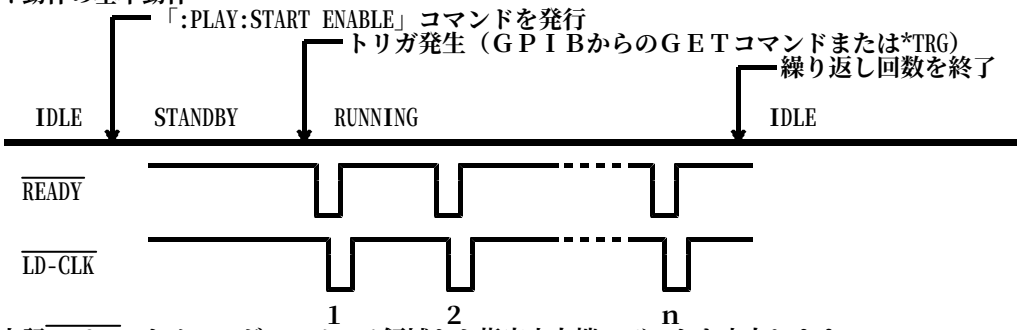
プレイ動作が ＲＵＮＮＩＮＧ状態にある場合は同一ﾌﾞﾛｯｸ番号のメモリ領域に対して以下のことを行うことができません。

:MEMORY:ASSIGN	ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数
:MEMORY:WRITE:INITIATE	ﾌﾞﾛｯｸ番号
:MEMORY:WRITE:NEXT	ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ列
:MEMORY:READ:INITIATE	ﾌﾞﾛｯｸ番号
:MEMORY:READ:NEXT?	ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数

ﾋﾞｯﾄ名称：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。

ﾊﾞｲﾄ名称：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 ＝ 「:PLAY:ASSIGN」で指定したﾃﾞｰﾀ数 × 「:PLAY:REPEAT」で指定した回数
で表されます。

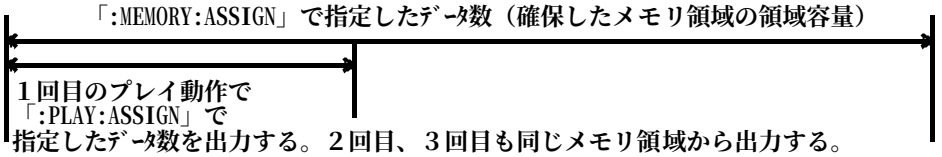
「Ⅳ－５－１」

書式　:PLAY:REPEAT ﾋﾞｯﾄ名称,回数
:PLAY:REPEAT ﾊﾞｲﾄ名称,回数

説明　ﾋﾞｯﾄ名称、またはﾊﾞ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
１～１０００００００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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,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が「: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ﾋﾞｯﾄ名称／ﾊﾞ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－５－２」

書式　:PLAY:REPEAT? ﾋﾞｯﾄ名称
:PLAY:REPEAT? ﾊﾞｲﾄ名称

説明　ﾋﾞｯﾄ名称、またはﾊﾞ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－５－３」

書式　:PLAY:ASSIGN ﾋﾞｯﾄ名称,ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ数
:PLAY:ASSIGN ﾊﾞｲﾄ名称,ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ数

説明　ﾋﾞｯﾄ名称、またはﾊﾞ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。

ﾌﾞﾛｯｸ番号：0,1
あらかじめ、「MEMORY:ASSIGN ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数」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。

ﾃﾞｰﾀ数：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。

このコマンドのﾌﾞﾛｯｸ番号で指定するメモリ領域の領域容量が「MEMORY:ASSIGN ﾌﾞﾛｯｸ番号,ﾊﾞｲﾄ数」コマンドのﾊﾞｲﾄ数で、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ﾋﾞｯﾄ名称／ﾊﾞ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ﾋﾞｯﾄ名称／ﾊﾞｲﾄ名称に、ﾌﾞ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、ﾃﾞｰﾀ数が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ﾋﾞｯﾄ名称／ﾊﾞｲﾄ名称に、ﾌﾞ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－５－４」

書式　:PLAY:ASSIGN? ﾋﾞｯﾄ名称
　　:PLAY:ASSIGN? ﾊﾞｲﾄ名称

説明　ﾋﾞｯﾄ名称、またはﾊﾞｲﾄ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、トーカに指定されると下記の応答メッセージを返送します。

　　ﾌﾞﾛｯｸ番号,ﾃﾞｰﾀ数

　　応答メッセージのﾌﾞﾛｯｸ番号が－１、ﾃﾞｰﾀ数が０の場合は、指定されたﾋﾞｯﾄ名称／ﾊﾞｲﾄ名称と
　　指定されたﾌﾞ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－５－５」

書式　:PLAY[:START] ﾋﾞｯﾄ名称,指令
　　:PLAY[:START] ﾊﾞｲﾄ名称,指令

説明　ﾋﾞｯﾄ名称、またはﾊﾞ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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,ENABLE」の後のトリガ発生でプレイ動作を開始します。
　　「:PLAY:ASSIGN」コマンドのﾃﾞｰﾀ数で指定した数のデータを、ﾌﾞ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$\overline{\text{READY}}$信号がLowの時の$\overline{\text{LD-CLK}}$の立ち下がりの時です。
　　「:PLAY:ASSIGN」コマンドのﾃﾞ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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,ENABLE」を受信したとき、指定と同一のﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称に
　　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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,DISABLE」を受信したとき、指定と同一のﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称に
　　対するプレイ動作がＩＤＬＥ状態にある時は無視します。

　　「:PLAY:START ﾋﾞｯﾄ名称,ENABLE」を受信したとき、指定のﾋﾞｯﾄ名称が含まれるﾊﾞｲﾄ名称に対する
　　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 ﾊﾞｲﾄ名称,ENABLE」を受信したとき、指定のﾊﾞｲﾄ名称に含まれるﾋﾞｯﾄ名称に対する
　　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ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称に対する
　　「:PLAY:START ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,ENABLE」を受信すると「実行エラー」になります。

　　「:PLAY:START ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,ENABLE」を受信したとき、指定されたﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称に
　　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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称,ENABLE」を受信したとき、指定されたﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称に
　　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。

「Ⅳ－５－６」

書式　:PLAY:STATE? ﾋﾞｯﾄ名称
　　:PLAY:STATE? ﾊﾞｲﾄ名称

説明　ﾋﾞｯﾄ名称、またはﾊﾞ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。

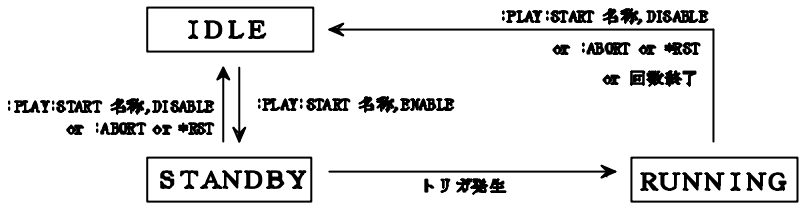
応答　このコマンドの後、トーカに指定されると下記のいずれかの応答メッセージを返送します。

IDLE
STANDBY
RUNNING

ＩＤＬＥ状態：「:PLAY:START ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称 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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称 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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称 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 ﾋﾞｯﾄ名称/ﾊﾞｲﾄ名称 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－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?		イベント発生条件をクエリする
:EVEnt?		イベントの発生状況をクエリする
:ENAble?		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－６－１」

書式　:STATUS:EXTERNAL:TRANSITION 数値

説明　端末側の外部ステータス入力「ST1,ST2,ST3,ST4,ST5,ST6,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、ST5と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ので、
高速の信号（パルス幅１００ｎＳｅｃ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。
「ST1,ST2,ST3,ST4,ST5,ST6,ST8」はソフトウェアでの監視により行っているので、どちらの変化も
検出できますが高速の信号（パルス幅５００ｕＳｅｃ以下）には対応できません。

「Ⅳ－６－２」

書式　:STATUS:EXTERNAL:ENABLE 数値

説明　端末側の外部ステータス入力「ST1,ST2,ST3,ST4,ST5,ST6,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－６－３」

書式　:STATUS:EXTERNAL:TRANSITION?

説明　端末側の外部ステータス入力「ST1,ST2,ST3,ST4,ST5,ST6,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進整数値で返送します。
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

「Ⅳ－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－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－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－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。

「Ⅳ－８－１」

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。