

MP1762D
誤り検出器
取扱説明書
(GPIB プログラミング)

初版

- ・ 製品をご使用前に必ず本取扱説明書をお読みください。
- ・ 安全にお使いいただくための重要事項は、MP1762D 誤り検出器取扱説明書に記載してありますのでそちらをお読みください。
- ・ 本書は製品とともに保管してください。

アンリツ株式会社

安全情報の表示について

当社では人身事故や財産の損害を避けるために、危険の程度に応じて下記のようなシグナルワードを用いて安全に関する情報を提供しています。記述内容を十分理解して機器を操作するようにしてください。

下記の表示およびシンボルは、そのすべてが本器に使用されているとは限りません。また、外観図などが本書に含まれるとき、製品に張り付けたラベルなどがその図に記入されていない場合があります。

説明書中の表示について



危険

回避しなければ、死亡または重傷に至る切迫した危険状況があることを警告しています。



警告

回避しなければ、死亡または重傷に至る可能性がある潜在的危険について警告しています。



注意

回避しなければ、軽度または中程度の人体の傷害に至る可能性がある潜在的危険、または、物的損害の発生のみが予測されるような危険状況について警告しています。

機器に表示または説明書に使用されるシンボルについて

機器の内部や操作箇所の近くに、または説明書に、安全上あるいは操作上の注意を喚起するための表示があります。

これらの表示に使用しているシンボルの意味についても十分理解して、注意に従ってください。



禁止行為を示します。丸の中や近くに禁止内容が描かれています。



守るべき義務的の行為を示します。丸の中や近くに守るべき内容が描かれています。



警告や注意を喚起することを示します。三角の中や近くにその内容が描かれています。



注意すべきことを示します。四角の中にその内容が書かれています。



このマークを付けた部品がリサイクル可能であることを示しています。

MP1762D

誤り検出器

取扱説明書（GPIBプログラミング）

2004年（平成16年）3月22日（初 版）

- ・予告なしに本書の内容を変更することがあります。
- ・許可なしに本書の一部または全部を転載・複製することを禁じます。

Copyright © 2004, ANRITSU CORPORATION

Printed in Japan

品質証明

アンリツ株式会社は、本製品が出荷時の検査により公表機能を満足することを証明します。

品質保証

- ・ アンリツは、本ソフトウェアが付属のマニュアルに従った使用方法にもかかわらず、実質的に動作しなかった場合に、無償で補修または交換します。
- ・ その保証期間は、購入から1年間とします。
- ・ 補修または交換後の本ソフトウェアの保証期間は、購入時から一年内の残余の期間、または補修もしくは交換後から30日のいずれか長い方の期間とします。
- ・ 本ソフトウェアの不具合の原因が、天災地変などの不可抗力による場合、お客様の誤使用の場合、またはお客様の不十分な管理による場合は、保証の対象外とさせていただきます。

また、この保証は、原契約者のみ有効で、再販売されたものについては保証しかねます。

アンリツ株式会社は、本製品の欠陥に起因する損害のうち、予見できない特別の事情に基づき生じた損害およびお客様の取引上の損失については、責任を負いかねます。

当社へのお問い合わせ

本製品の故障については、本説明書(紙版説明書では巻末、CD版説明書では別ファイル)に記載の「本製品についてのお問い合わせ窓口」へすみやかにご連絡ください。

国外持出しに関する注意

1. 本製品は日本国内仕様であり、外国の安全規格などに準拠していない場合もありますので、国外へ持ち出して使用された場合、当社は一切の責任を負いかねます。
2. 本製品および添付マニュアル類は、輸出および国外持ち出しの際には、「外国為替及び外国貿易法」により、日本国政府の輸出許可や役務取引許可を必要とする場合があります。また、米国の「輸出管理規則」により、日本からの再輸出には米国政府の再輸出許可を必要とする場合があります。

本製品や添付マニュアル類を輸出または国外持ち出しする場合は、事前に必ず弊社の営業担当までご連絡ください。

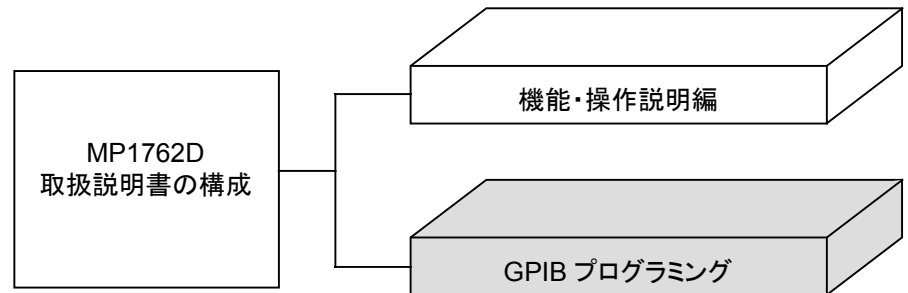
輸出規制を受ける製品やマニュアル類を廃棄処分する場合は、軍事用途等に不正使用されないように、破砕または裁断処理していただきますようお願い致します。

商標・登録商標

- 「HP Basic」は Hewlett-Packard Company の登録商標です。
- 「HP」は Hewlett-Packard Company の登録商標です。
- 「MS-DOS」は Microsoft Corporation の登録商標です。
- 「Quick Basic」はマイクロソフト株式会社の登録商標です。

MP1762D 取扱説明書の構成

MP1762D 誤り検出器の取扱説明書は、下記の 2 冊で構成されています。利用目的に合わせて使い分けてください。



機能・操作説明編 : MP1762D の概要、使用前の準備、パネル説明、規格、性能、機能および操作法を解説してあります。

GPIB プログラミング : MP1762D は IEEE488.2 対応機種ですので IEEE488.2 にもとづいた GPIB リモート制御について解説されています。プログラム例は、HP9000 シリーズ用 HP-BASIC、マイクロソフト社製 Quick Basic を適用しています。

目 次

1 章 概 要	1-1
GPIB 規格変遷	1-3
MP1762D GPIB 機能	1-4
2 ポート GPIB 機能概要	1-4
GPIB 1 / GPIB 2 を利用したシステムアップ例	1-5
2 章 規 格	2-1
インタフェースファンクション	2-3
デバイスメッセージリスト	2-5
IEEE488.2 共通コマンドとサポート対象コマンド	2-7
ステータスメッセージ	2-9
MP1762D デバイスメッセージ	2-11
3 章 バス接続とアドレス設定	3-1
GPIB ケーブルによるデバイスの接続	3-3
アドレスの確認および設定方法	3-4
アドレスの設定	3-5
MP1761B/C とのトラッキング動作時の接続	3-6
外部プリンタとの接続	3-9
4 章 イニシャル設定	4-1
IFC ステートメントによるバスの初期化	4-4
DCL, SDC バスコマンドによるメッセージ交換の初期化	4-6
*RST コマンドによるデバイスの初期化	4-8
INI コマンドによるデバイスの初期化	4-10
電源投入時のデバイスの状態	4-11

5章 リスナ入力フォーマット	5-1
リスナ入力プログラムメッセージ文法表記の要点	5-4
セパレータ, ターミネータ, ヘッダ前置スペース	5-4
プログラムコマンドメッセージの一般形式	5-6
問合せメッセージの一般形式	5-7
プログラムメッセージの機能要素	5-8
<TERMINATED PROGRAM MESSAGE>	5-8
<PROGRAM MESSAGE TERMINATOR>	5-9
<white space>	5-10
<PROGRAM MESSAGE>	5-10
<PROGRAM MESSAGE UNIT SEPARATOR>	5-11
<PROGRAM MESSAGE UNIT>	5-11
<COMMAND MESSAGE UNIT>/<QUERY MESSAGE UNIT>	5-12
<COMMAND PROGRAM HEADER>	5-13
<QUERY PROGRAM HEADER>	5-15
<PROGRAM HEADER SEPARATOR>	5-16
<PROGRAM DATA SEPARATOR>	5-16
プログラムデータのフォーマット	5-17
< DECIMAL NUMERIC PROGRAM DATA>	5-18
< NON-DECIMAL NUMERIC PROGRAM DATA>	5-20
6章 トーカ出力フォーマット	6-1
リスナ入力とトーカ出力フォーマットの文法上の相違点	6-4
レスポンスメッセージの機能要素	6-5
<TERMINATED RESPONSE MESSAGE>	6-5
<RESPONSE MESSAGE TERMINATOR>	6-5
<RESPONSE MESSAGE>	6-6
<RESPONSE MESSAGE UNIT SEPARATOR>	6-7
<RESPONSE MESSAGE UNIT>	6-7
<RESPONSE HEADER SEPARATOR>	6-8
<RESPONSE DATA SEPARATOR>	6-8
<RESPONSE HEADER>	6-8
<RESPONSE DATA>	6-10

7 章 共通コマンド	7-1
MP1762D サポート 共通コマンドのグループ機能別分類	7-3
サポートコマンドの分類とリファレンス	7-4
8 章 ステータス・ストラクチャー	8-1
IEEE488.2 標準ステータスのモデル	8-4
ステータスバイト (STB) レジスタ	8-6
ESB および MAV サマリメッセージ	8-6
装置固有のサマリメッセージ	8-7
STB レジスタの読み出しとクリア	8-8
SRQ のイネーブル	8-10
標準イベントステータス・レジスタ	8-12
標準イベントステータス・レジスタのビット定義	8-12
問合せエラーの詳細	8-13
標準イベントステータス・レジスタの読み取り・書き込み・クリア	8-14
標準イベントステータス・イネーブルレジスタの 読み取り・書き込み・クリア	8-14
拡張イベントステータス・レジスタ	8-15
END イベントステータス・レジスタのビット定義	8-16
ERROR イベントステータス・レジスタのビット定義	8-17
拡張イベントステータス・レジスタの読み取り・書き込み・クリア	8-18
拡張イベントステータス・イネーブルレジスタの 読み取り・書き込み・クリア	8-18
キュー (待ち行列) モデル	8-19
デバイスとコントローラ間の同期テクニック	8-21
シーケンシャル実行の強制	8-21
デバイスの出力キュー応答待ち	8-22
サービスリクエスト待ち	8-23

9 章 デバイス・メッセージの詳細 9-1

 デバイス・メッセージ一覧表 9-3

 デバイス・メッセージ一覧表 (アルファベット順) 9-3

 デバイス・メッセージ一覧表 (パネルとの対応) 9-9

 デバイス・メッセージの詳細説明 9-24

10 章 プログラム作成例 10-1

 HP9000を用いたサンプル・プログラム例 10-6

 DEC_{pc}を用いたサンプル・プログラム例 10-70

付録 A 従来器とのコンパチビリティ A-1

付録 B パターンの DMA 転送 B-1

付録 C 初期値一覧表 C-1

付録 D トラッキング項目一覧表 D-1

1 章

概 要

この章では、 **GPIB** 規格変遷の概要および MP1762D 誤り検出器の **GPIB** 機能の概要について説明します。

目 次

GPIB 規格変遷	1-3
MP1762D GPIB 機能	1-4
2 ポート GPIB 機能概要	1-4
GPIB 1 / GPIB 2 を利用したシステムアップ例	1-5

1 章 概 要

GPiB 規格変遷

MP1762D 誤り検出器は、外部コントローラと組み合わせて、システムバスを構築し、測定の自動化を提供します。このために使用する計測バスとして、本器は、GPiBインタフェースバス(IEEE std 488.2-1987)を第1インタフェースとして標準装備しています。

GPiB (General Purpose Interface Bus)は、1975年IEEE (Institute of Electric and Electronics Engineers 米国電気電子学会) で制定されたプログラマブル計測機器用標準デジタルインタフェースバス規格で、そのオリジナルは1975年に、IEEE Std 488-1975 の名で発表されました。

このIEEE Std 488-1975は、ドキュメント上の編集・改訂を経て、1978年にIEEE Std 488-1978として発行されました。このIEEE Std 488-1978は、インタフェース側の仕様を定めたハードウェア規格のみであったため、1982年に、デバイス側の仕様を定めたソフトウェア規格として、IEEE Std 728-1982が加えられました。

このIEEE Std 728-1982は、デバイスメッセージの転送フォーマットを規格化したものですが、利用者側にたったソフトウェア共有化のコンセプトに欠けるところがありました。そこで、このコンセプト実現のためIEEE Std 728-1982の改訂版として1987年に、IEEE Std 488.2-1987(これ以降、IEEE 488.2と記す)が登場することとなり、メッセージ交換プロトコル、メッセージデータコード、デバイス入出力フォーマット、共通コマンドの標準化が強化されました。

このIEEE 488.2の登場により、これまで運用してきたIEEE Std 488-1978(これ以降、IEEE 488と記す)は、IEEE Std 488.1-1987(これ以降、IEEE 488.1と記す)と改名されることになりました。以上述べてきたGPiB規格の変遷を整理すると次のようになります。

規格対象	旧規格	新規格	備 考
ハードウェア	IEEE 488	IEEE 488.1	IEEE 488.1はIEEE 488 と同じ
ソフトウェア	IEEE 728	IEEE 488.2	IEEE 488.2はIEEE 728 の改訂版

IEEE 488.2をサポートする機器は、IEEE 488.1とも互換性があることが必須条件ですが、IEEE 488.1 (IEEE 488) をサポートする機器は、IEEE 488.2への適合を保証されていません。

MP1762D GPIB 機能

MP1762D には、次のような GPIB 機能があります。

- (1) 電源スイッチ および LOCAL キー等の一部を除くすべての機能の制御
- (2) すべての設定条件の読み出し
- (3) 割込み機能とシリアルポール動作
- (4) パーソナルコンピュータやその他の測定器と組み合わせて自動計測システムを構成できます。
- (5) GPIB は GPIB 1, GPIB 2 と 2 ポートによって構成

最後の (5) については、下記で見本例を挙げて説明します。

2 ポート GPIB 機能概要

MP1762D には GPIB のポートが 2 ポート用意されています。第 1 インタフェースとしての GPIB 1 側のポートは主として外部ホストコンピュータにより MP1762D のリモート制御を主とし、第 2 インタフェースとしての GPIB 2 側のポートは外部プリンタへの測定データの出力を主目的としております。したがって GPIB 1 側はデバイスポートとして、GPIB 2 側はシステムコントローラポートとして使用することにより、より効率の良いシステムアップが可能となります。

(1) GPIB 1 の機能

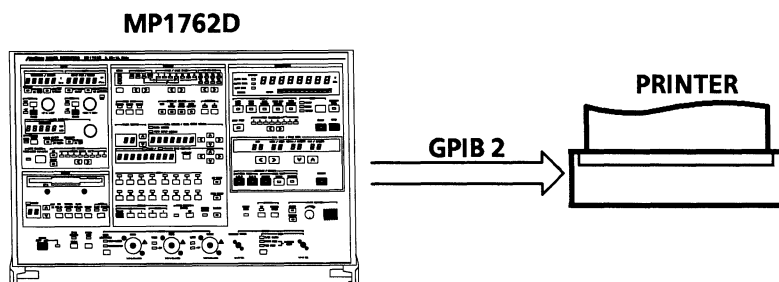
GPIB 1 ポートは従来の 1 ポート GPIB を備えた測定器と同様に扱うことができます。したがってパワー ON 時や通常の測定状態ではデバイスポートとしての機能を行い、またトラッキング動作時においてはシステムコントローラの設定を行うことにより、MP1761B/C パルスパターン発生器を制御するためのシステムコントローラポートとして機能を行います。

(2) GPIB 2 の機能

GPIB 1 ポートとは独立して、GPIB 2 ポートに接続される各デバイスの制御ポートとして使用します。したがって GPIB 2 ポートは常にシステムコントローラポートとして機能し、デバイスポートとはなり得ません。

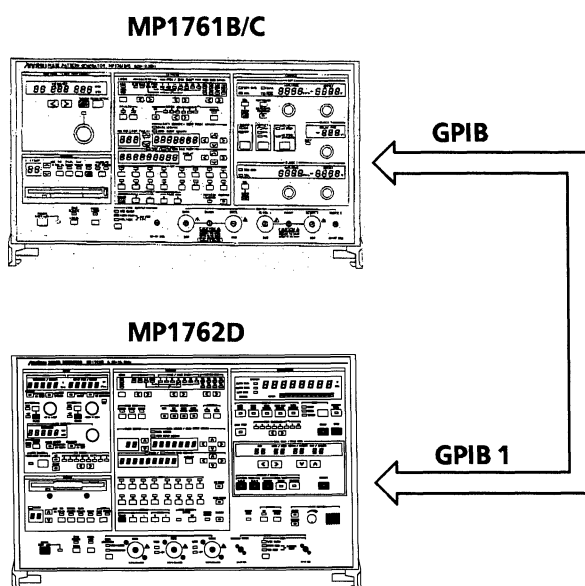
GPIB 1 / GPIB 2 を利用したシステムアップ例

(1) スタンドアロン方式 (その1)……パネル操作



MP1762Dで測定したデータをパネル操作によりプリンタへ出力します。

(2) スタンドアロン方式 (その2)……トラッキング動作



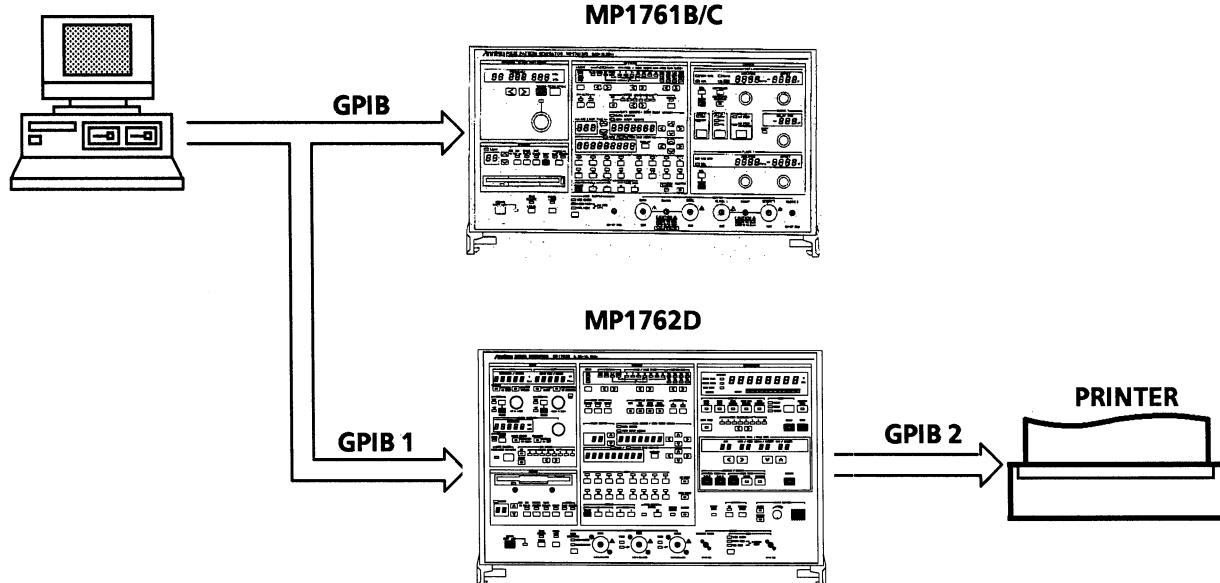
① 送信部の設定の一部が受信部の設定に連動します。このトラッキング動作時は、外部コントローラを接続できません。

② 受信部の設定の一部が送信部の設定に連動します。このトラッキング動作時は、外部コントローラを接続できません。

※ 本トラッキング動作でマスター(コントローラ)となるのは、MP1761B/CもしくはMP1762Dのどちらか一方のみです。

(3) ホストコンピュータ制御

ホストコンピュータ



ホストコンピュータより **GPIB 1**ポートを介して、**MP1761B/C**と **MP1762D**を制御します。
このとき、**GPIB 2**ポートよりプリンタヘデータを出力します。

2 章

規 格

この章では、MP1762D の **GPIB** 規格としてインタフェースファンクション、およびデバイスメッセージリストについて説明します。

目 次

インタフェースファンクション	2-3
デバイスメッセージリスト	2-5
IEEE 488.2 共通コマンドとサポート対象コマンド	2-7
ステータスメッセージ	2-9
MP1762D デバイスメッセージ	2-11

2章 規 格

インタフェースファンクション

IEEE 488.2 では、IEEE 488.1で規定されている GPIB インタフェース機能の中から各測定器が備えるべき必要最小限のサブセットを決め、すくなくともシステム用として使えるレベルにしています。MP1762D の GPIB 1 および GPIB 2 は、下表のコード欄に示すサブセットを備えています。

GPIB 1 インタフェースファンクション

コード	インタフェースファンクション	IEEE 488.2 規定
SH1	ソースハンドシェイクの全機能有り。 データ送信のタイミングをとります。	全機能標準装備。デバイスは完全なソースハンドシェイク機能を有すること。
AH1	アクセプタハンドシェイクの全機能有り。 データ受信のタイミングをとります。	全機能標準装備。デバイスは完全なアクセプタ・ハンドシェイク機能を有すること。
T6	基本的トーカ機能有り。 シリアルポール機能有り。 トークオンリ機能無し。 MLAによるトーカ解除機能有り。	デバイスは、T5, T6, TE5, TE6いずれかのサブセットを有すること。トークオンリ機能はIEEE 488.2 規定の規定外とする。
L4	基本的リスナ機能有り。 リスンオンリ機能無し。 MTAによるリスナ解除機能有り。	デバイスは、L3, L4, LE3, LE4いずれかのサブセットを有すること。リスンオンリ機能はIEEE 488.2 規定の規定外とする。
SR1	サービスリクエスト、ステータスバイトの全機能有り。	全機能標準装備
RL1	リモート/ローカルの全機能有り。 ローカルロックアウトの機能有り。	RL0 (機能無し)またはRL1 (全機能)
PP0	パラレルポール機能無し。	PP0 (機能無し)またはPP1 (全機能)
DC1	デバイスクリアの全機能有り。	全機能標準装備。
DT1	デバイストリガ機能有り。	DT0 (機能無し)またはDT1 (全機能)
C1,C2, C3,C4, C7	コントローラ機能有り。 トラッキング動作時のみコントローラにすることが可能。	C0 (機能無し) もしくはC4とC5、またはC7, C9, C11 のいずれか

GPIB 2 インタフェースファンクション

コード	インタフェースファンクション
SH1	ソースハンドシェイクの全機能有り。 データ送信のタイミングをとります。
AH1	アクセプタハンドシェイクの全機能有り。 データ受信のタイミングをとります。
T6	基本的トーカ機能有り。 シリアルポール機能有り。 トークオンリ機能無し。 MLAによるトーカ解除機能有り。
L4	基本的リスナ機能有り。 リスンオンリ機能無し。 MTAによるリスナ解除機能有り。
SR0	サービスリクエスト、ステータスバイトの機能無し。
RL0	リモート/ローカルの機能無し。 ローカルロックアウトの機能無し。
PP0	パラレルポール機能無し。
DC0	デバイスクリアの機能無し。
DT0	デバイストリガ機能無し。
C1,C2,C3,C4, C28	コントローラ機能有り。

デバイスメッセージリスト

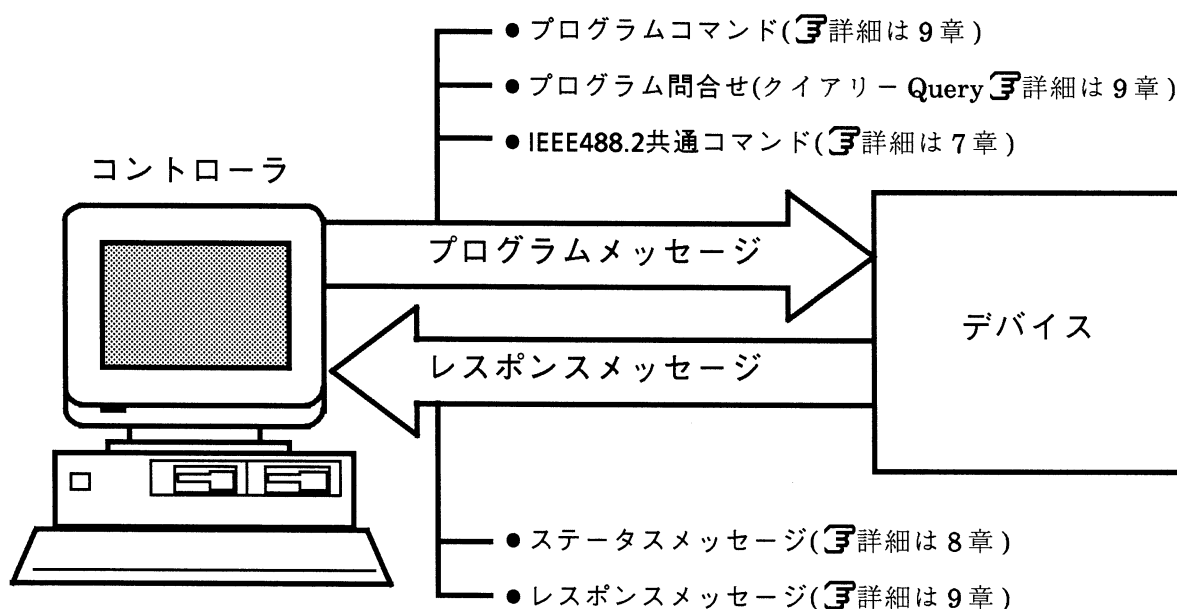
デバイスメッセージは、バスのモードすなわちATNラインが偽の時、システムインタフェースをととしてコントローラとデバイス間で送受されるデータメッセージで、プログラムメッセージとレスポンスメッセージの二つがあります。

プログラムメッセージは、コントローラからデバイスへ転送されるASCIIデータメッセージです。プログラムメッセージには、プログラム命令(コマンドー **command**)およびプログラム問合せ(クイアリー **Query**)の二つがあります。この二つを次ページ以降でリストします。

プログラム命令には、MP1762D制御専用に使される装置固有のコマンドの他に、IEEE 488.2 共通コマンドがあります。IEEE 488.2 共通コマンドは、MP1762Dを含み、バス上に接続されたその他のIEEE 488.2 対応測定器にも共通に適用されるプログラム命令です。

プログラム問合せは、デバイスからレスポンスメッセージを得るためのコマンドであって、あらかじめコントローラからデバイスへ転送しておき、その後にデバイスからのレスポンスメッセージをコントローラでアクセプトします。

レスポンスメッセージは、デバイスからコントローラへ転送されるASCIIデータメッセージです。ここでは、その中からステータスメッセージおよびプログラム問合せに対応するレスポンスメッセージを次ページ以降でリストします。



以上述べたメッセージはデバイスの入出力バッファを介してやりとりされます。出力バッファについては出力キューとも言います。入力バッファ、出力バッファについて簡単に説明しておきます。

入力バッファ	出力キュー
DAB (プログラムメッセージや問合せメッセージ)の構文を解析し、実行する前に、それらのメッセージを一次的に蓄えておく FIFO (First in First out) タイプのメモリエリアです。 MP1762Dの入力バッファサイズは 256 バイトあります。	FIFO タイプの待ち行列メモリエリアです。デバイスからコントローラへ出力するすべての DAB (レスポンスメッセージ)は、コントローラがそれを読み終わるまでは、このメモリに蓄えられます。 MP1762Dの出力キューサイズは 256 バイトあります。

IEEE 488.2 共通コマンドとサポート対象コマンド

下表に IEEE 488.2 規格で定められている 39 種類の共通コマンドをリストします。この中から MP1762D で使用される IEEE 488.2 共通コマンドを◎印で示します。

ニーモニック	コマンド・フルスペル名	IEEE488.2規定	当社サポートコマンド (MP1762D)
*AAD	Accept Address Command	任意	
*CAL?	Calibration Query	任意	
*CLS	Clear Status Command	必須	◎
*DDT	Define Device Trigger Command	任意	
*DDT?	Define Device Trigger Query	任意	
*DLF	Disable Listener Function Command	任意	
*DMC	Define Macro Command	任意	
*EMC	Enable Macro Command	任意	
*EMC?	Enable Macro Query	任意	
*ESE	Standard Event Status Enable Command	必須	◎
*ESE?	Standard Event Status Enable Query	必須	◎
*ESR?	Standard Event Status Register Query	必須	◎
*GMC?	Get Macro Contents Query	任意	
*IDN?	Identification Query	必須	◎
*IST?	Individual Status Query	任意	
*LMC?	Learn Macro Query	任意	
*LRN?	Learn Device Setup Query	任意	
*OPC	Operation Complete Command	必須	◎
*OPC?	Operation Complete Query	必須	◎
*OPT?	Option Identification Query	任意	◎
*PCB	Pass Control Back Command	C0以外なら必須	
*PMC	Purge Macro Command	任意	
*PRE	Parallel Poll Register Enable Command	任意	
*PRE?	Parallel Poll Register Enable Query	任意	
*PSC	Power On Status Clear Command	任意	◎
*PSC?	Power On Status Clear Query	任意	◎
*PUD	Protected User Data Command	任意	
*PUD?	Protected User Data Query	任意	
*RCL	Recall Command	任意	
*RDT	Resource Description Transfer Command	任意	
*RDT?	Resource Description Transfer Query	任意	
*RST	Reset Command	必須	◎
*SAV	Save Command	任意	
*SRE	Service Request Enable Command	必須	◎
*SRE?	Service Request Enable Query	必須	◎

ニーモ ニック	コマンド・フルスペル名	IEEE488.2規定	当社サポートコマンド (MP1762D)
*STB?	Read Status Byte Query	必須	◎
*TRG	Trigger Command	DT1なら必須	◎
*TST?	Self Test Query	必須	◎
*WAI	Wait to Continue Command	必須	◎

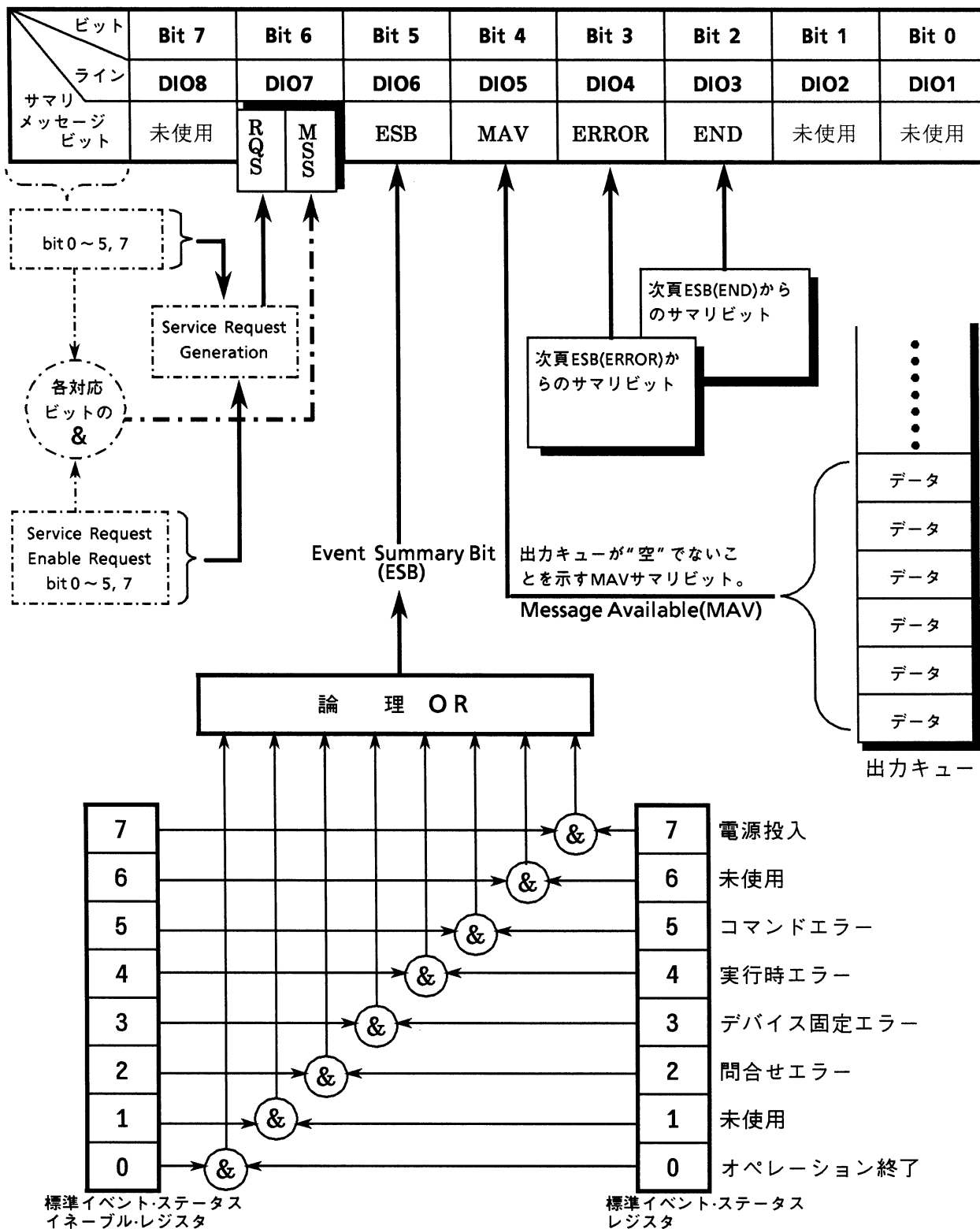


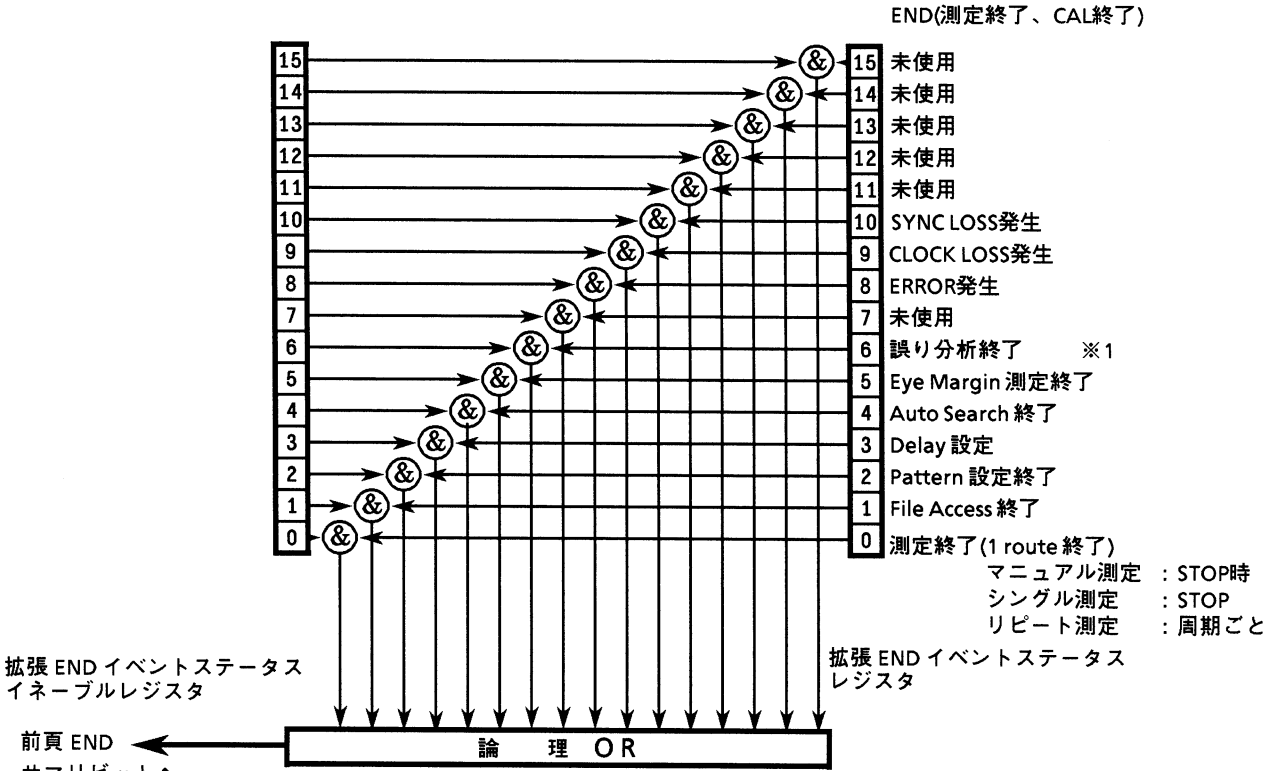
IEEE488.2 共通コマンドは、必ず*で始まります。詳細については、7章を参照してください。

ステータスメッセージ

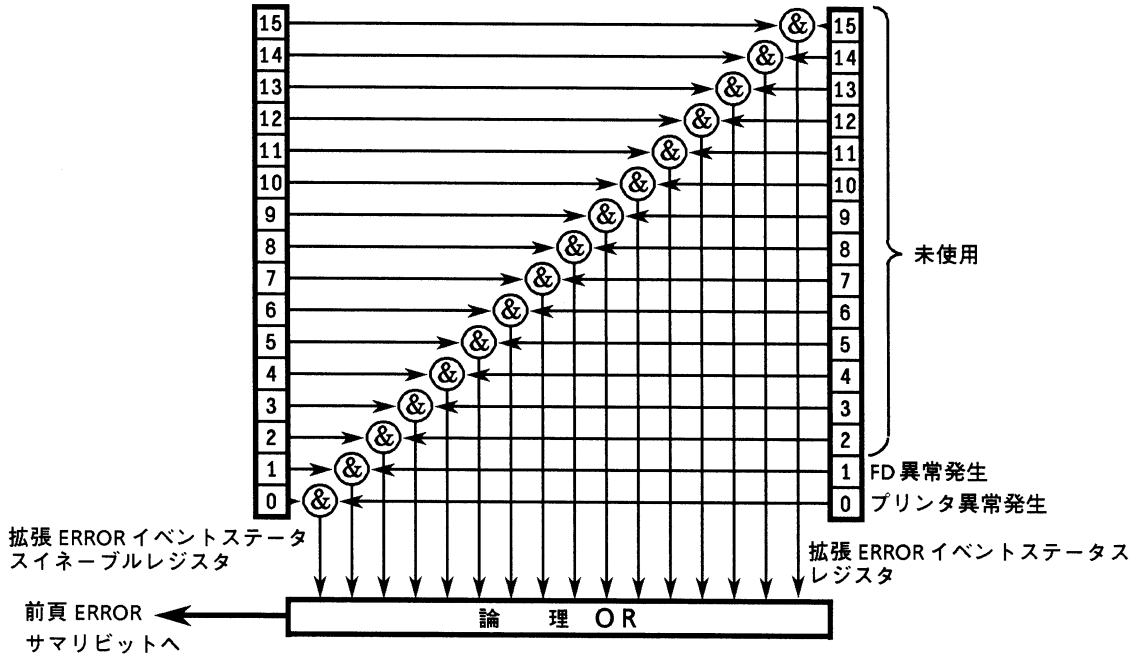
MP1762Dで使用されるステータスバイト・レジスタのサービス要求用サマリメッセージの構造を下記に示します。

ステータスバイト・レジスタのサマリビット構成





※1) 誤り分析はOPTION-01 を実装している場合に限られます。



MP1762D デバイスメッセージ

MP1762D固有のプログラムコマンド、クイアリ(問合せ)、およびレスポンスメッセージです。デバイスメッセージの一覧表および説明を**9章**に示します。

3 章

バス接続とアドレス設定

GPIBシステムインタフェース上に接続されたデバイスに対するリモート制御は、制御手順のパラメータとしてのアドレスを参照することから開始されます。

以下、この章では、**GPIB**使用前の準備として必ず行わなければならない**GPIB**ケーブルの接続およびアドレス設定について説明します。

目 次

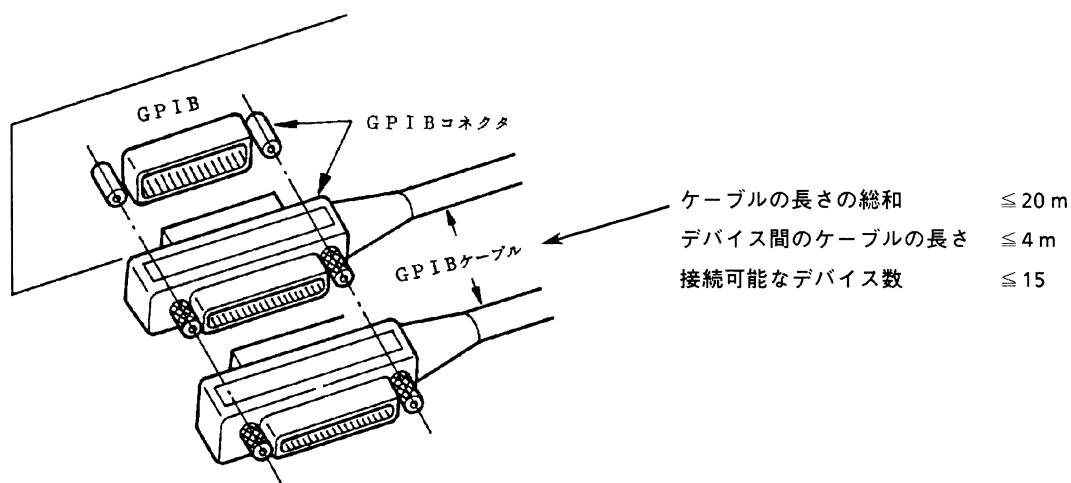
GPIBケーブルによるデバイスの接続	3-3
アドレスの確認および設定方法	3-4
アドレスの設定	3-5
MP1761B/C とのトラッキング動作時の接続	3-6
外部プリンタとの接続	3-9

3 章 バス接続とアドレス設定

GPIB ケーブルによるデバイスの接続

GPIB ケーブル接続用コネクタは、背面パネル上に取り付けられています。GPIB ケーブルの接続は、必ず電源を投入する前に行ってください。

一つのシステムに接続可能なデバイス台数は、コントローラを含めて、最大15台までですが、その場合、下図右側に示した条件に従って接続してください。

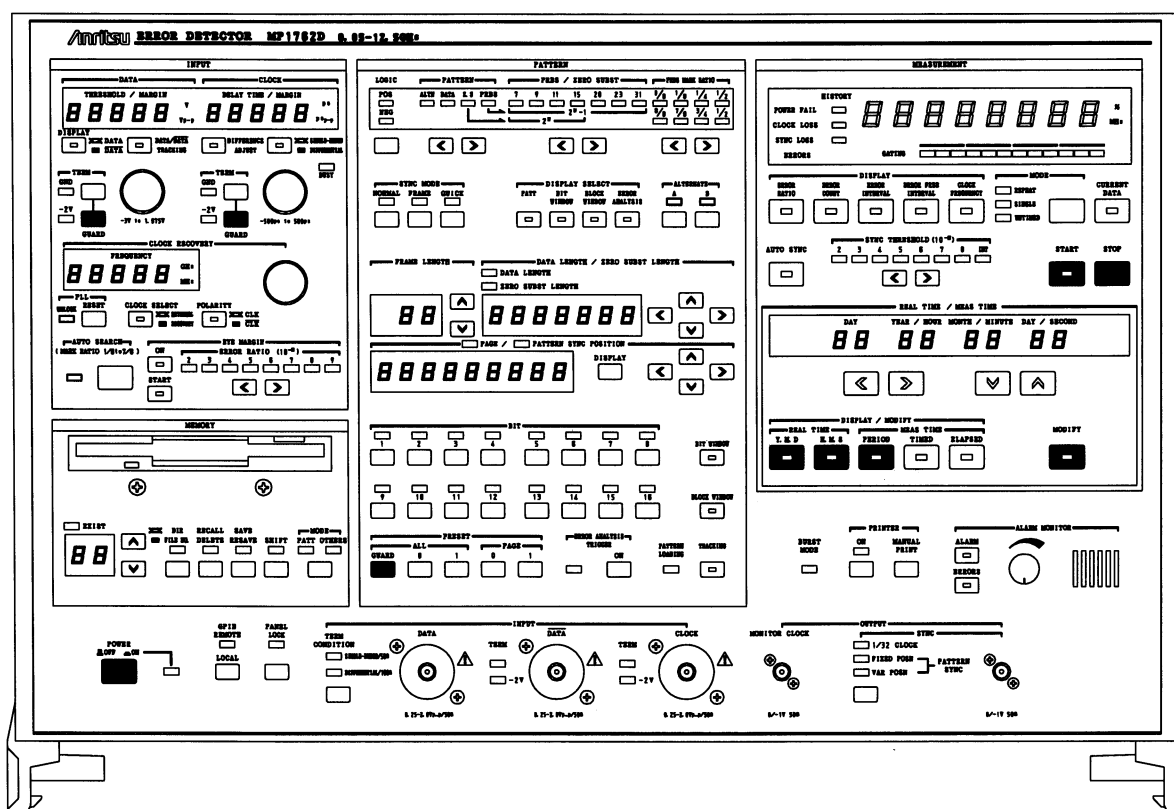


アドレスの確認および設定方法

MP1762D の GPIB アドレスは、電源投入前後に設定します。GPIB 1 (デバイス用) のアドレスは、工場出荷時に 0 番が背面パネル GPIB ADDRESS スイッチによって設定済みです。アドレスを 0 番のまま使う場合には、アドレスを設定する必要はありません。アドレスを設定する場合は、MP1762D をローカル状態にしておき、背面パネルの GPIB ADDRESS スイッチによって設定します。一般に、電源投入時点では、GPIB 上のデバイスは、ローカル状態となります。

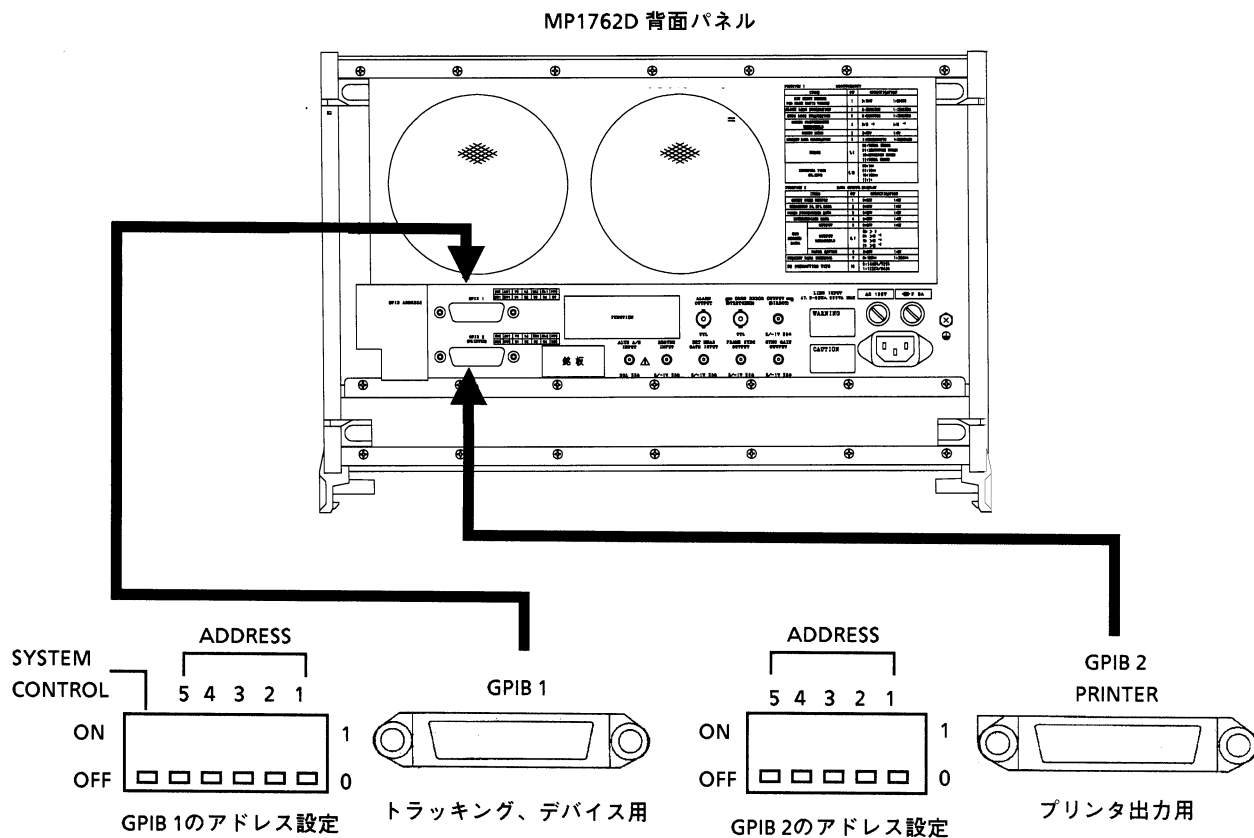
注: ● 本器は、通電中常時 GPIB の「ADDRESS」スイッチの設定状態をチェックして、自分自身のアドレスを決定しています。そのため、アドレスの変更は、本器がリモート状態の時以外は常時受け付けられません。

- 本器をデバイスとして外部コントローラから制御する場合は、GPIB 1 アドレス・スイッチの「SYSTEM CONTROL」を OFF (0) にしてください。



アドレスの設定

MP1762D の GPIB 2ポートの GPIB アドレス設定は背面パネルのディップスイッチで各々行います。

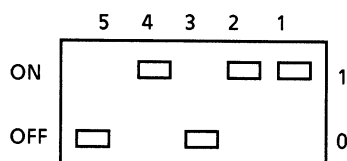


GPIB 1および GPIB 2のアドレスとしては各々0~30が設定できます。5個のスイッチは、それぞれ重みが異なっており「5」は16、「4」は8、「3」は4、「2」は2、「1」は1の重みを持っています。

たとえば、アドレスを11に設定するには、

$$11 = 8 + 2 + 1$$

で下記のようにスイッチの「4」、「2」、「1」を ON にします。



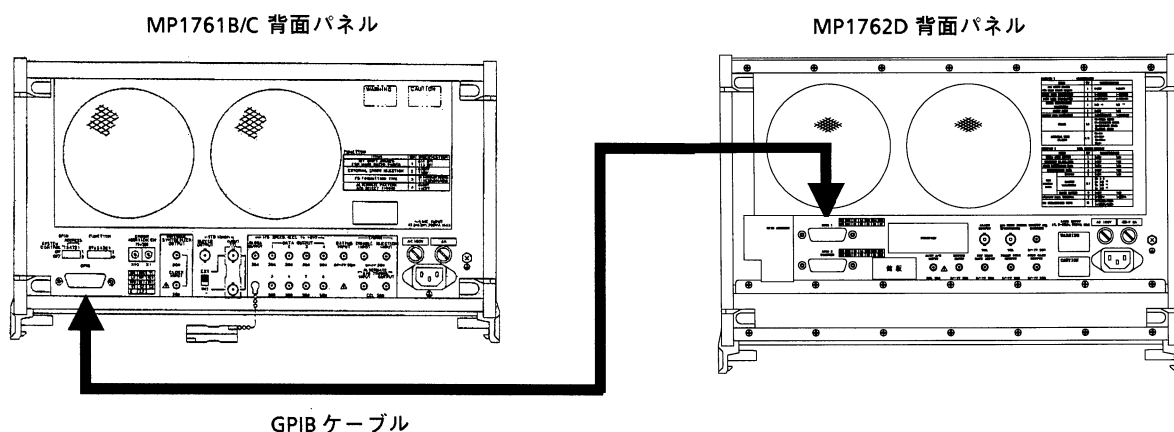
ただし、すべてのスイッチを ON としたアドレス31の場合は、アドレス0とみなします。

MP1761B/C とのトラッキング動作時の接続

トラッキング動作とは、MP1761B/C と MP1762D の間でパターン 設定を同じにする機能です。MP1761B/C と MP1762D どちらか一方が **Master** となり、残りの一方が **Slave** となって **Master** 側の設定内容に **Slave** 側の設定をあわせます。

(1) MP1761B/C が Master となり MP1762D を制御する場合

MP1761B/C で設定された内容を MP1762D に GPIB を経由して設定する場合の設定方法と接続を以下に示します。

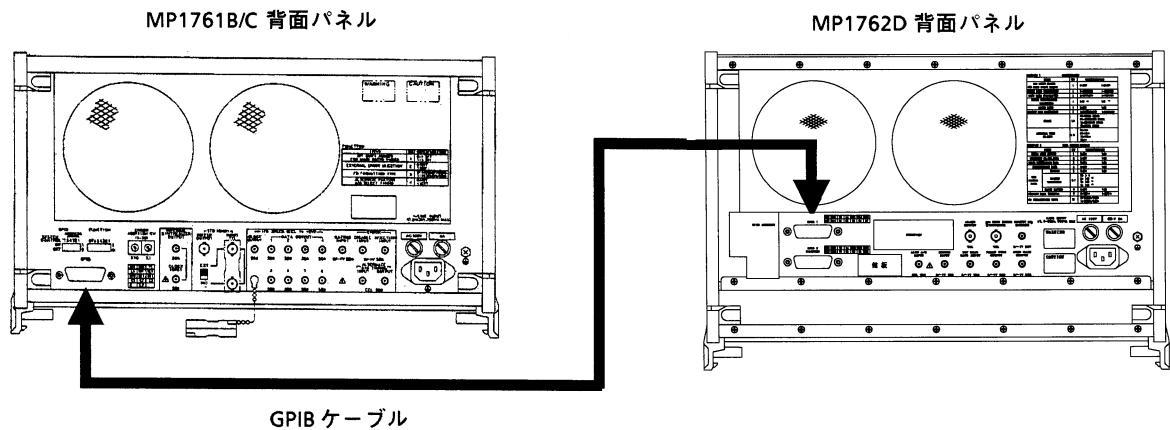


- a) 図のように MP1761B/C の背面パネル GPIB コネクタと MP1762D の GPIB1 コネクタ間を GPIB ケーブル (添付されています。) で接続します。
- b) MP1761B/C 背面パネルの GPIB アドレス SW にある「SYSTEM CONTROL」を ON (1) にします。
- c) MP1762D 背面パネルの GPIB 1 アドレス SW の値を MP1761B/C の GPIB アドレス +2 に設定します。
- d) MP1761B/C の電源を再投入します。
- e) MP1761B/C 正面パネルの TRACKING キーを ON にします。

これでパターントラッキングが可能となります。

(2) MP1762Dが Master となり MP1761B/C を制御する場合

MP1762Dで設定された内容を MP1761B/C に GPIB を経由して設定する場合の設定方法と接続を以下に示します。



- a) 図のように MP1762D の背面パネル GPIB 1コネクタと MP1761B/C の GPIB コネクタ間を GPIB ケーブル (添付されています。) で接続します。
- b) MP1762D背面パネルの GPIB 1アドレス SW にある「SYSTEM CONTROL」を ON (1)にします。
- c) MP1761B/C 背面パネルの GPIB 1アドレス SW の値を MP1762D の GPIB アドレス +2に設定します。
- d) MP1762D の電源を再投入します。
- e) MP1762D 正面パネルの TRACKING キーを ON にします。

これでパターントラッキングが可能となります。

(3) MP1761B/C と MP1762D との間でトラッキングを行う項目

パターントラッキング機能でトラッキングを行う設定項目を以下に示します。

MP1762D 正面パネルのパターン設定部

The diagram shows the front panel of the MP1762D, specifically the PATTERN settings section. The panel is organized into several functional areas:

- LOGIC:** Includes buttons for POS, NEG, and a selection switch.
- PATTERN:** Includes buttons for ALIN, DATA, Z/S, and PRBS, along with a selection switch.
- PRBS/ZERO SUBST:** Includes buttons for PRBS, ZERO, and SUBST, along with a selection switch.
- PRBS MARK/RATIO:** Includes buttons for PRBS, MARK, and RATIO, along with a selection switch.
- SYNC MODE:** Includes buttons for NORMAL, FRAME, and QUICK.
- DISPLAY SELECT:** Includes buttons for PATT, BIT, BLOCK, and ERROR, along with a selection switch.
- ALTN:** Includes buttons for A and B, along with a selection switch.
- FRAME LENGTH:** Includes a digital display and up/down arrow buttons.
- DATA LENGTH/ZERO SUBST LENGTH:** Includes buttons for DATA LENGTH and ZERO SUBST LENGTH, along with a digital display and up/down arrow buttons.
- PAGE / PATTERN SYNC POSITION:** Includes a digital display and up/down arrow buttons.
- BIT:** Includes buttons for BIT 1 through BIT 16, along with a selection switch.
- BIT WINDOW:** Includes a button for BIT WINDOW.
- BLOCK WINDOW:** Includes a button for BLOCK WINDOW.
- PRESET:** Includes buttons for ALL, 0, and 1, along with a selection switch.
- PAGE:** Includes buttons for 0 and 1, along with a selection switch.
- ERROR ANALYSIS TRIGGER:** Includes a button for TRIGGER.
- PATTERN TRACKING:** Includes buttons for PATTERN LOADING and TRACKING, along with a selection switch.

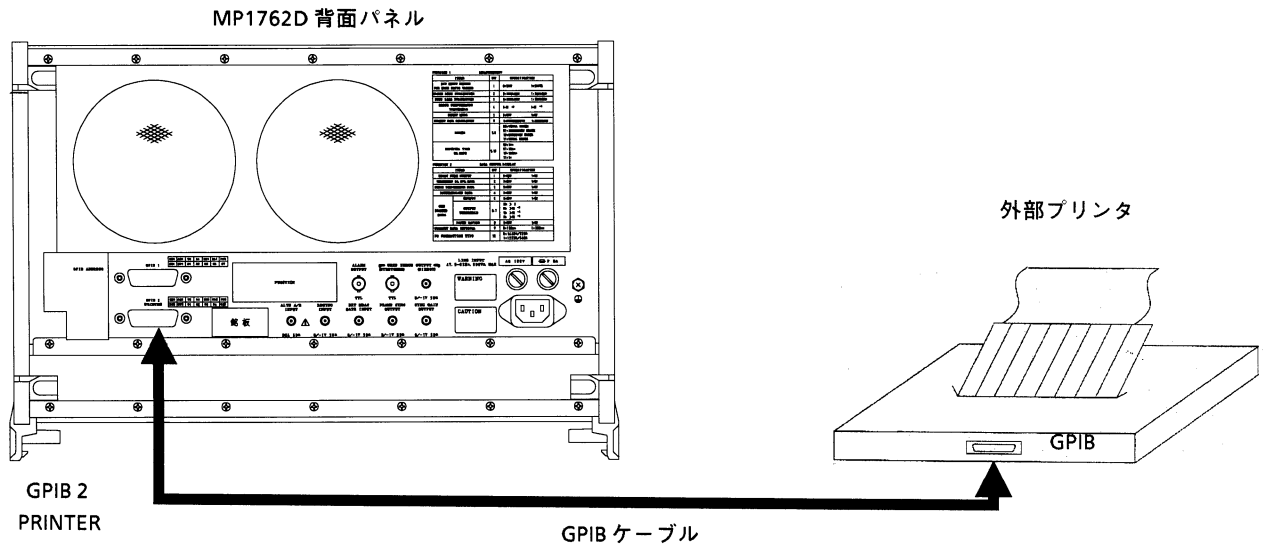
トラッキングできる項目は、上図の網かけ部分のパターン設定についてです。

ただし、この中で MP1761B/C と MP1762D の設定が共通で無い所 (たとえば、誤り分析データ) は、パターン・トラッキングの対象となりません。

 パターン・トラッキングの対象となる設定詳細は、『付録 D トラッキング項目一覧表』を参照してください。

外部プリンタとの接続

MP1762D は、測定データを外部プリンタへ出力するための **GPIB** ポートを有しています。外部プリンタに測定データを出力するときの接続方法とアドレス設定を以下に示します。



- 図のように MP1762D 背面パネル **GPIB 2, PRINTER** コネクタと外部プリンタ **GPIB** コネクタ間を **GPIB ケーブル** で接続します。
- MP1762D 背面パネルにある **GPIB 2, PRINTER** アドレス **SW** の設定をします。
- 外部プリンタのアドレスを b) で設定した MP1762D **GPIB 2, PRINTER** アドレス + 2 に設定します。
- MP1762D の正面パネル「 **PRINTER ON** 」キー、または「 **MANUAL PRINT** 」キーを **ON** にします。

これで測定データは出力可能な状態になります。

ただし、測定データが無い場合、(たとえば測定停止中)は出力されません。

4 章

イニシャル設定

GPIB インタフェースシステムは、**3**段階のレベルに分けて初期化されます。第**1**のレベルは、「バスの初期化」で、システムのバスをアイドル状態にします。第**2**のレベルは、「メッセージ交換の初期化」で、デバイスをプログラムメッセージ受信可能な状態にします。

第**3**のレベルは、「デバイスの初期化」で、デバイス特有の機能を初期化します。これら初期化レベル**1, 2, 3**は、いわばデバイスの動作を開始させるための準備に相当します。

目 次

IFC ステートメントによるバスの初期化	4-4
DCL, SDC バスコマンドによるメッセージ交換の初期化	4-6
*RST コマンドによるデバイスの初期化	4-8
INI コマンドによるデバイスの初期化	4-10
電源投入時のデバイスの状態	4-11

4章 イニシャル設定

GPIB システムの初期化については、従来IEEE 488.1では、次の2点が規定されていました。

- バスの初期化 コントローラからの**IFC**メッセージによって、バスに接続されたすべてのインタフェース機能を初期化する。
- デバイスの初期化 GPIB バスコマンド**DCL**によって GPIB 上の全デバイス、または GPIB バスコマンド**SDC**によって、指定したデバイスだけをデバイスごとに定められている初期状態に戻す。

IEEE 488.2では、上記を3つのレベルに分け、第1レベルを『バスの初期化』とし、最も高位のレベルに位置付けました。『デバイスの初期化』は、さらに第2レベル『メッセージ交換の初期化』と第3レベル『デバイスの初期化』の2階層にわけて規定されると共に、電源投入時のデバイスの状態についても、ある既知の状態へ設定することが定められました。

以上のことをまとめると下表のようになります。

レベル	初期化の種類	概 要
1	バスの初期化	コントローラからの IFC メッセージによって、バスに接続されたすべてのインタフェース機能を初期化します。
2	メッセージ交換の初期化	GPIB バスコマンド DCL によって GPIB 上の全デバイス、または GPIB バスコマンド SDC によって、指定したデバイスのメッセージ交換の初期化やオペレーションが終了したことをコントローラへ報告する機能を無効にします。
3	デバイスの初期化	*RST または INI リセットコマンドによって GPIB 上の指定したデバイスだけを、過去の使用状態に関係なく、そのデバイス固有の、既知の状態に戻します。

以下、レベル1, 2, 3については、これらを実行する命令およびその結果である初期化対象項目を中心に説明します。また、電源投入時に設定される既知の状態について説明します。

IFC @

IFC ステートメントによるバスの初期化

■ 書 式

IFC△@セレクトコード

■ 使用例

IFC @1

■ 解 説

指定したセレクトコードの GPIB において、IFC ラインを約100 μ S の間アクティブ状態 (電氣的に **Low** レベルの状態)にします。IFC@を実行すると指定したセレクトコードの GPIB バスラインに接続されているすべてのデバイスのインタフェース機能が初期化されます。システムコントローラのみが送信できます。

インタフェース機能の初期化とは、コントローラによって設定されているデバイスのインタフェース機能の状態 (トーカ, リスナ, その他) を解除して初期状態に戻すもので、下表の中で○印の各ファンクションを初期化します。△印は、その一部を初期化します。

No	ファンクション	記 号	IFCでの初期化
1	ソース・ハンドシェイク	SH	○
2	アクセプタ・ハンドシェイク	AH	○
3	トーカまたは拡張トーカ	TまたはTE	○
4	リスナまたは拡張リスナ	LまたはLT	○
5	サービス要求	SR	△
6	リモート・ローカル	RL	
7	パラレル・ボール	PP	
8	デバイス・クリア	DC	
9	デバイス・トリガ	DT	
10	コントローラ	C	○

IFCステートメントがTrue (IFC@文の実行によって下図の IFC ラインが Low レベル) でも、レベル2, 3に対する初期化は除かれます。したがって、デバイスの動作状態 (パラメータの設定値、ランプの ON/OFF など) には影響を与えません。

前頁の表の中から IFCメッセージによるデバイスの状態をいくつか挙げます。

- ① トーカ/リスナ すべてのトーカ、すべてのリスナは、100 μ S 以内にアイドル状態 (TIDS, LIDS) になります。
- ② コントローラ コントローラがアクティブ (SACS: System control Active State) でなければ、100 μ S 以内に、コントローラはアイドル状態 CIDS (Controller Idle State) になります。
- ③ コントロール権の戻し IFC@を実行した時、もしシステムコントローラ (GPiB上で最初にコントローラになるよう定められている装置) が他の装置にコントローラとしての機能を移譲している状態であれば、システムコントローラとしての機能が戻されます。なお、一般には、システムコントローラの [RESET] キーを押せば、システムコントローラから IFCメッセージが出力されます。
- ④ サービス要求のデバイス デバイスがコントローラに SRQメッセージを送信している状態 (下図の SRQ ラインがデバイスによって LOW レベルに設定されている状態) は、解除されませんが、これによって、コントローラがシステムバス下の全デバイスをシリアルボールモード下においている状態は解除されます。
- ⑤ リモート状態のデバイス 現在リモート状態にあるデバイスは、IFCメッセージによって、リモート状態を解除されることはありません。

DCL @

DCL, SDC バスコマンドによるメッセージ交換の初期化

■ 書 式

DCL△@セレクトコード[プライマリアドレス][セカンダリアドレス]

■ 使用例

DCL @1 バス下の全デバイスのメッセージ交換の初期化 (DCL送出)
DCL @103 アドレス3番のデバイスのみのメッセージ交換の初期化 (SDC 送出)

■ 解 説

指定したセレクトコードの GPIB 上の全デバイス、または指定したデバイスだけの、メッセージ交換に関する初期化を行うステートメントです。

メッセージ交換の初期化の目的は、パネルの設定状態を変える必要はないが、デバイス内部のメッセージ交換に関係する部分が他のプログラムの実行などで、コントローラから制御するには不適當な状態になっている場合に、メッセージ交換の初期化を行うことによって、コントローラから新しい命令を送れるように準備を整えることにあります。

■ セレクトコードのみ指定した場合

指定したセレクトコードの GPIB 上のすべてのデバイスのメッセージ交換の初期化を行います。DCL@は GPIB に DCL (Device Clear) バスコマンドを出力します。

■ アドレスまで指定した場合

指定したデバイスに対するメッセージ交換の初期化を行います。指定したセレクトコードの GPIB においてリスナを解除した後、指定したデバイスだけをリスナに設定し SDC (Selected Device Clear) バスコマンドを出力します。

■ メッセージ交換の初期化対象項目

- | | |
|-------------------------|--|
| ① 入力バッファと出力キュー | クリアされます。 |
| ② 構文解析部・実行制御部・応答作成部 ... | リセットされます。 |
| ③ *RSTを含むデバイスコマンド | これらのコマンドの実行を妨げるすべてのコマンドをクリアします。 |
| ④ 対パラメータ・プログラムメッセージ ... | 対パラメータのため、実行が延期されている部分のコマンドおよび間合せもすべて捨てます。 |
| ⑤ *OPC コマンドの処理 | デバイスを OCIS ステート (Operation Complete Command Idle State) にします。この結果、オペレーション終了ビットを標準イベントステータス・レジスタに立てることはできません。 |

- ⑥ ***OPC?**問合せの処理 デバイスを **OQIS** ステート (**Operation Complete Query Idle State**) にします。この結果、オペレーション終了ビット1を出力キューに立てることができません。**MAV** ビットはクリアされます。
- ⑦ システム構築の自動化 これを実行する ***ADD** と ***DLF** 共通コマンドを無効にします。(MP1762Cでは、これらのコマンドをサポートしていません。)
- ⑧ デバイスファンクション メッセージ交換に関する部分は、すべてアイドル状態におかれます。デバイスは、コントローラからのメッセージを待ち続けます。

⚠ 注意

下記事項は、デバイスクリアによって処理することを禁じられています。

- ① 現在のデバイスの設定データやストアされているデータを変えること。
- ② フロントパネルI/Oへの割り込み
- ③ 出力キューのクリアにおいて、**MAV** ビットクリア以外に他のステータスバイトを変えること。
- ④ 現在進行中のデバイスの動作に影響を与えたり、割り込みを行うこと。

■ DCL @文による GPIB バスコマンド送出順序

DCL@ 文の GPIB バスコマンド **DCL**, **SDC** の送出順序を下記に示します。

ステートメント	バスコマンド送出順序 (ATN ライン "LOW")	データ (ATN ライン "HIGH")
DCL @ セレクトコード	UNL, DCL	_____
DCL @ 装置番号	UNL, LISTEN アドレス, [2次アドレス], SDC	_____

*RST

*RSTコマンドによるデバイスの初期化

■ 書 式

*RST

■ 使用例

WRITE @103:"*RST" アドレス3番のデバイスのみをレベル3で初期化

■ 解 説

***RST (Reset)** コマンドは **IEEE488.2** 共通コマンドの一つで、デバイスをレベル3でリセットします。

一般にデバイスは装置固有のコマンド(デバイスメッセージ)を使って、さまざまな状態に設定されています。***RST** コマンドは、それらの中でデバイスのある特定の既知の状態を再現するのに使用されます。なお、デバイスのオペレーションの終了を無効にすることについては、レベル2の場合と同じです。

■ WRITE @文の装置番号の指定

指定したアドレスのデバイスをレベル3で初期化します。

バスコマンドの送出順序は、**ATN** ラインがこの期間に、指定したセレクトコードの **GPIB** においてリスナを解除した後、指定したデバイスだけをリスナに設定します。**ATN** ラインが偽になると***RST** コマンドを送出します。

■ デバイス初期化対象項目

- | | |
|-------------------------|--|
| ① デバイス固有の機能・状態 | それまでの来歴に関わらずある既知の状態に戻します。 |
| ② *OPC コマンドの処理 | デバイスを OCIS ステート (Operation Complete Command Idle State) にします。この結果、オペレーション終了ビットを標準イベントステータス・レジスタに立てることはできません。 |
| ③ *OPC? 問合せの処理 | デバイスを OQIS ステート (Operation Complete Query Idle State) にします。この結果、オペレーション終了ビット1を出力キューに立てることができません。 MAV ビットはクリアされます。 |
| ④ マクロコマンド | マクロ動作を禁止し、マクロコマンドを受け付けないモードにします。また、マクロ定義を設計者が示す状態に戻します。 |

NOTE

***RST** コマンドは、下記事項には影響を与えません。

- ① IEEE488.1 インタフェースの状態 ② デバイスアドレス
- ③ 出力キュー ④ Service Request Enable レジスタ
- ⑤ Standard Event Status Enable レジスタ
- ⑥ Power-on-status-clear フラグ設定
- ⑦ デバイスの規格に影響する校正データ
- ⑧ DMC (Define Macro Contents) コマンドで定義したマクロ
- ⑨ PUD (Protected User Data) 問合せに対するレスポンスメッセージ
- ⑩ RDT (Resource Description Transfer) 問合せに対するレスポンスメッセージ

このほか、MP1762D 固有のものとして時計、カレンダーに関する設定パラメータ等があります。

(**㊦** ③, ④, ⑤は 8 章を参照してください。なお、⑧～⑩については、MP1762D ではサポートしていません。)

前頁①デバイス固有の機能・状態について、MP1762D 固有の初期設定一覧表を次頁に示します。

装置固有初期設定一覧表

グループ	初期設定	備 考
設定状態	初期化する	初期値は付録 C 初期値一覧表を参照してください。
GPIB アドレス	初期化されない	
時計、日付	初期化されない	

INI

INIコマンドによるデバイスの初期化

■ 書 式

INI

■ 使用例 (プログラムメッセージ)

WRITE @103:"INI" アドレス3番のデバイスのみをレベル3で初期化

■ 解 説

INI コマンドは MP1762D 固有のデバイスメッセージの一つで、デバイスをレベル3でリセットするため、そのコマンドをプログラムメッセージとしてコントローラからデバイスへ送ります。

機能としては、前述、*RST 共通コマンドと同じです。

■ WRITE @文の装置番号の指定

指定したアドレスのデバイスをレベル3で初期化します。

コマンド送出順序は、ATN ラインが真の期間に、指定したセレクトコードの GPIB に
おいてリスナを解除した後、指定したデバイスだけをリスナに設定します。ATN ライ
ンが偽になると、INI コマンドをプログラムメッセージとして、指定したリスナへ出力
します。

■ デバイス初期化対象項目

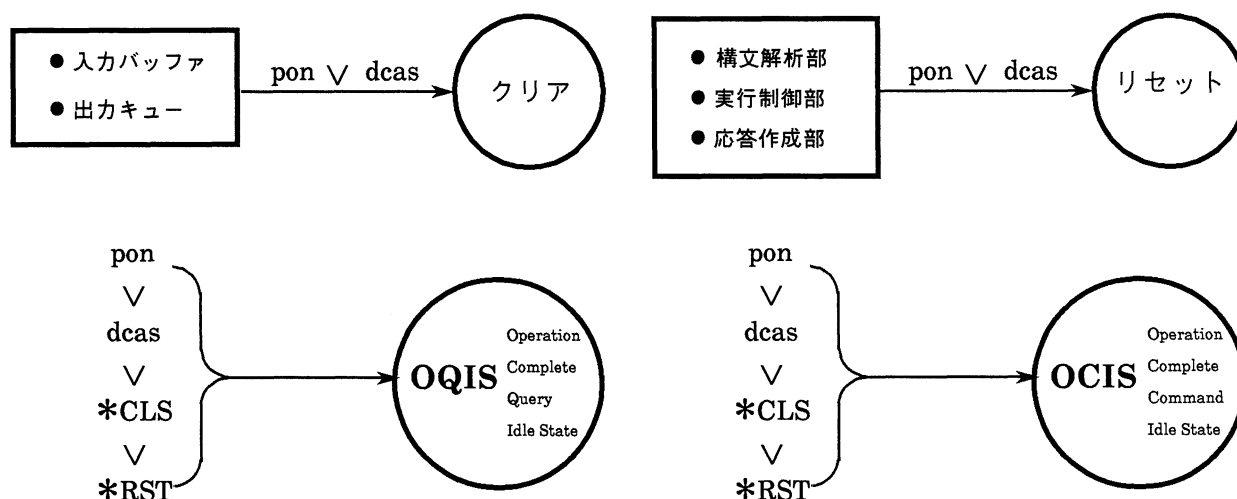
デバイス初期化対象項目は、前述 *RST コマンドの場合と同じです。

電源投入時のデバイスの状態

電源が投入されると：

- ① 最後に電源を **OFF** した時の状態に設定されます。
- ② 入力バッファと出力キューは、クリアされます。
- ③ 構文解析部・実行制御部・応答作成部は、リセットされます。
- ④ デバイスを **OCIS** ステート (**Operation Complete Command Idle State**) にします。
- ⑤ デバイスを **OQIS** ステート (**Operation Complete Query Idle State**) にします。
- ⑥ **MP1762D** は、***PSC** コマンドを用意していますので、もし **PSC** フラグが **True** の時すべてのイベント・ステータス・イネーブル・レジスタは、クリアされます。
イベントはクリア後に記録されます。

①の特別な場合として、**MP1762D**の工場出荷後、最初にその電源投入した時には、前述で示した初期設定一覧表 (付C-1) のとおりに再現されます。②～⑤は電源投入以外でも、この状態に設定されますのでそのステートダイアグラムを下図に示します。



■ 電源投入時に変えてはいけない項目

- ① アドレス
- ② 関連するキャリブレーションデータ (**MP1762D**ではキャリブレーションデータはありません)
- ③ 以下のコモン疑問符に対するレスポンスで変化するデータやステート

*IDN?
 *OPT?
 *PSC?
 *PUD? (MP1762Dでは、サポートされていません)
 *RDT? (MP1762Dでは、サポートされていません)

■ POWER ON STATUS CLEAR (PSC) フラグに関係する項目

PSC フラグが **False** の時、サービス・リクエスト・イネーブルレジスタ、標準イベント・ステータス・イネーブルレジスタおよび拡張イベント・ステータス・イネーブルレジスタは影響されません。

また、PSC フラグが **True** か、***PSC** コマンドが実行されていない時は、前記レジスタはクリアされます。

■ 電源投入時に変わってもよい項目

- ① カレントデバイスファンクションステート
- ② ステータス情報
- ③ ***SAV/*RCL**レジスタ
- ④ ***DDT** コマンドで定義されたマクロ定義 (MP1762Dではサポートされていません)
- ⑤ ***DMC** コマンドで定義されたマクロ定義 (MP1762Dではサポートされていません)
- ⑥ ***EMC** コマンドで可能となったマクロ (MP1762Dではサポートされていません)
- ⑦ ***PCB** コマンドで受信したアドレス (MP1762Dではサポートされていません)

5 章

リスナ入力フォーマット

バスのデータモードすなわち**ATN**ラインが偽の時に、システムインタフェースをと
おしてコントローラとデバイス間で送受されるデータメッセージには、プログラム
メッセージとレスポンスメッセージの二つがあります。以下、この章では、リスナ
が受信するプログラムメッセージの書式について説明します。

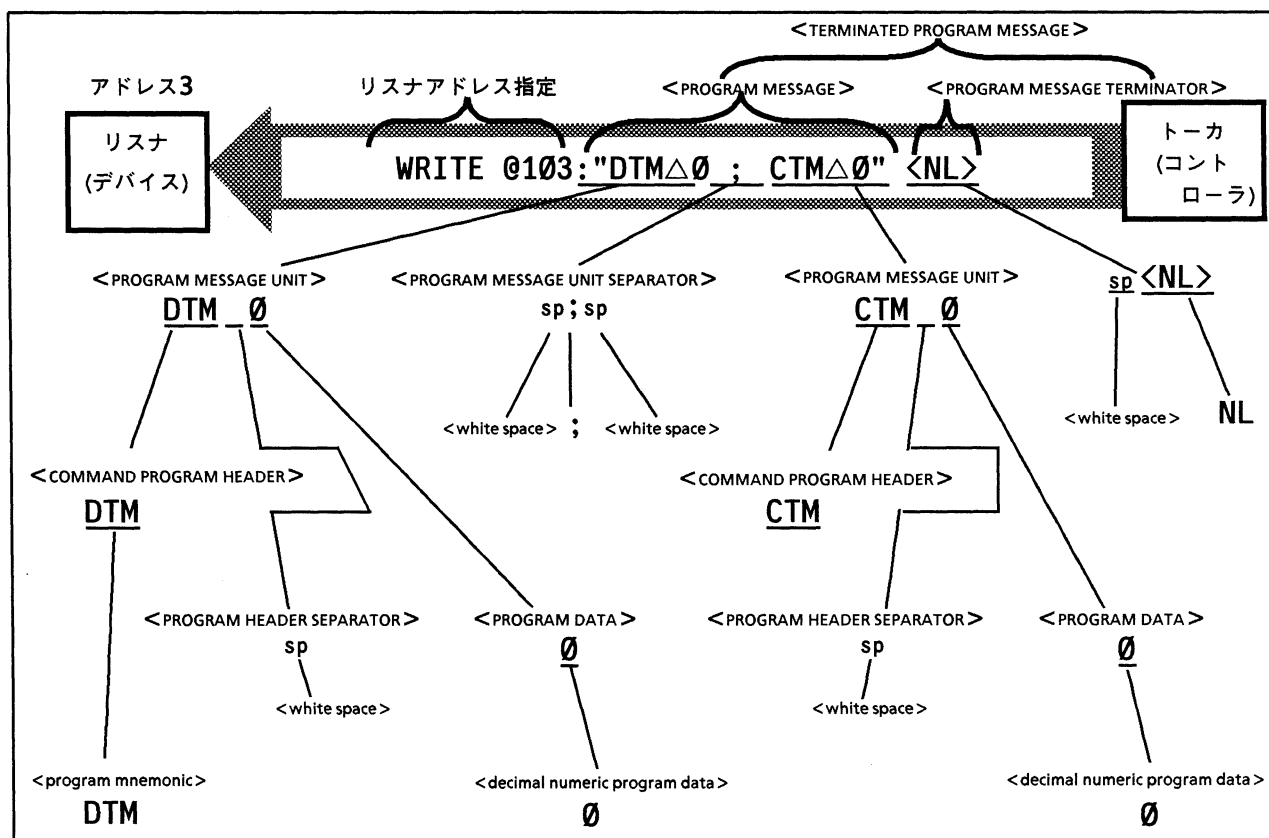
目 次

リスナ入力プログラムメッセージ文法表記の要点	5-4
セパレータ, ターミネータ, ヘッダ前置スペース	5-4
プログラムコマンドメッセージの一般形式	5-6
問合せメッセージの一般形式	5-7
プログラムメッセージの機能要素	5-8
<TERMINATED PROGRAM MESSAGE>	5-8
<PROGRAM MESSAGE TERMINATOR>	5-9
<white space>	5-10
<PROGRAM MESSAGE>	5-10
<PROGRAM MESSAGE UNIT SEPARATOR>	5-11
<PROGRAM MESSAGE UNIT>	5-11
<COMMAND MESSAGE UNIT>/<QUERY MESSAGE UNIT>	5-12
<COMMAND PROGRAM HEADER>	5-13
<QUERY PROGRAM HEADER>	5-15
<PROGRAM HEADER SEPARATOR>	5-16
<PROGRAM DATA SEPARATOR>	5-16
プログラムデータのフォーマット	5-17
<DECIMAL NUMERIC PROGRAM DATA>	5-18
<NON-DECIMAL NUMERIC PROGRAM DATA>	5-20

5章 リスナ入力フォーマット

プログラムメッセージは、プログラムメッセージ・ユニットのシーケンスで構成されており、各々のユニットは、プログラム命令またはプログラム問合せです。

下図は、データ入力およびクロック入力終端電圧を **GND** に設定するため、2つのプログラムメッセージ・ユニット **DTM Δ 0**と **CTM Δ 0**をプログラムメッセージ・ユニットセパレータで結び、一つのプログラムメッセージとしてコントローラからデバイスへ送出していることを示しています。



プログラムメッセージの書式は、機能を表すことのできる最小レベルの単位まで分割した機能要素のシーケンスから構成されます。上図でカギカッコ<>で囲まれた英大文字が機能要素の例です。機能要素を更に分割したものをコード化要素と呼びます。上図でカギカッコ<>で囲まれた英小文字がコード化要素の例です。

特定の経路の機能要素の選択を図で表したものを機能文法図と言います。また、特定の経路のコード化要素の選択を図で表したものをコード化文法図と言います。次ページからこの機能文法図・コード化文法図を使ってプログラムメッセージの書式を説明します。

コード化要素は、機能要素のデータバイトをデバイスに送るに必要な実際のバスのコード化を表しています。機能要素のデータバイトを受信したリスナは、各々の要素がコード化文法のルールに正しく従っているかどうかを解釈し、もし違反しているならば、機能要素と解釈することなくコマンド・エラーを発生します。

リスナ入力プログラムメッセージ文法表記の要点

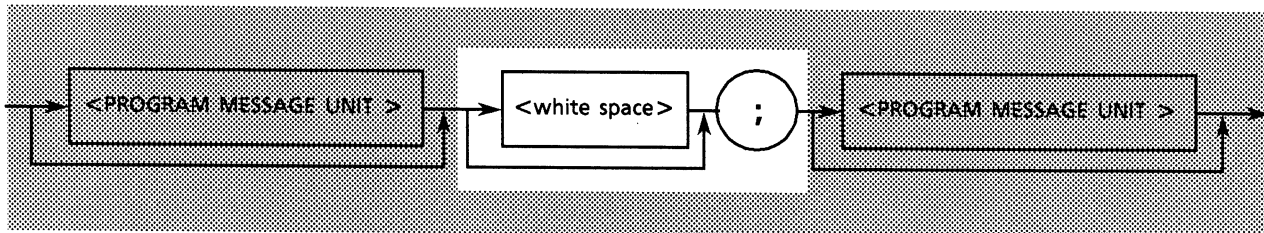
プログラムメッセージの機能要素およびプログラムデータのフォーマットの解説の要点を以下に示します。(複合コマンドと共通コマンドは省略)

セパレータ, ターミネータ, ヘッダ前置スペース

(1) PROGRAM MESSAGE UNIT SEPARATOR

複数のプログラムメッセージ・ユニットは、0個以上のスペース + セミコロンで連結されます。

<例1> 2つのプログラムメッセージユニットの連結一般形式



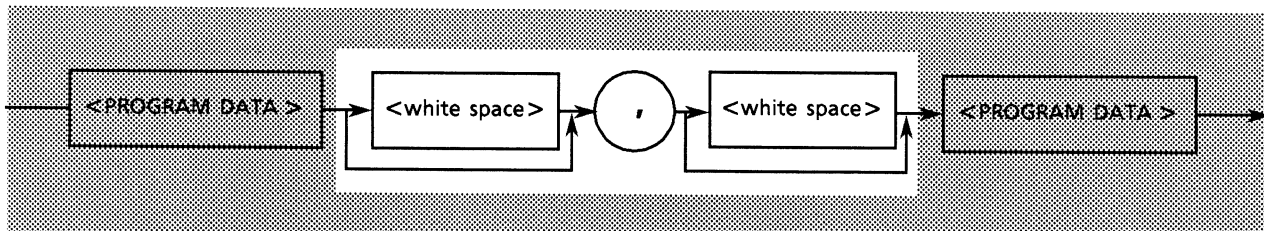
<例2> スペース1個 + セミコロン

DTM 0△; CTM 0

(2) PROGRAM DATA SEPARATOR

複数のプログラムデータは、0個以上のスペース + コンマ + 0個以上のスペースで区切ります。

<例1> 2つのプログラムデータの区切り一般形式



<例2> コンマのみ <例3> コンマ + スペース1個

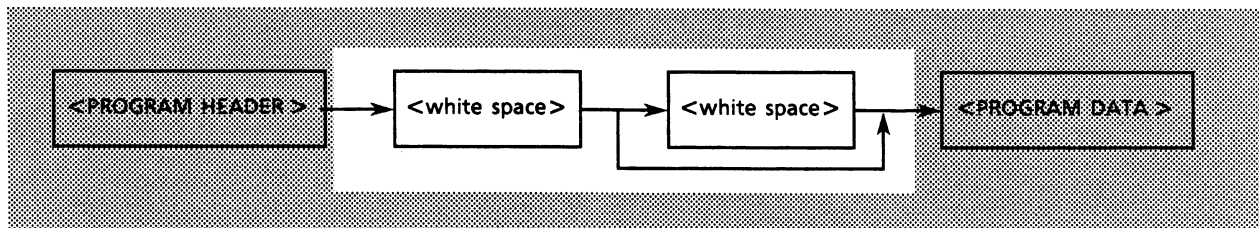
WRT 1,0

WRT 1,△0

(3) PROGRAM HEADER SEPARATOR

プログラムヘッダとプログラムデータの間を1個のスペース + 0個以上のスペースで区切ります。

<例1> 単一コマンドプログラムヘッダ一般形式

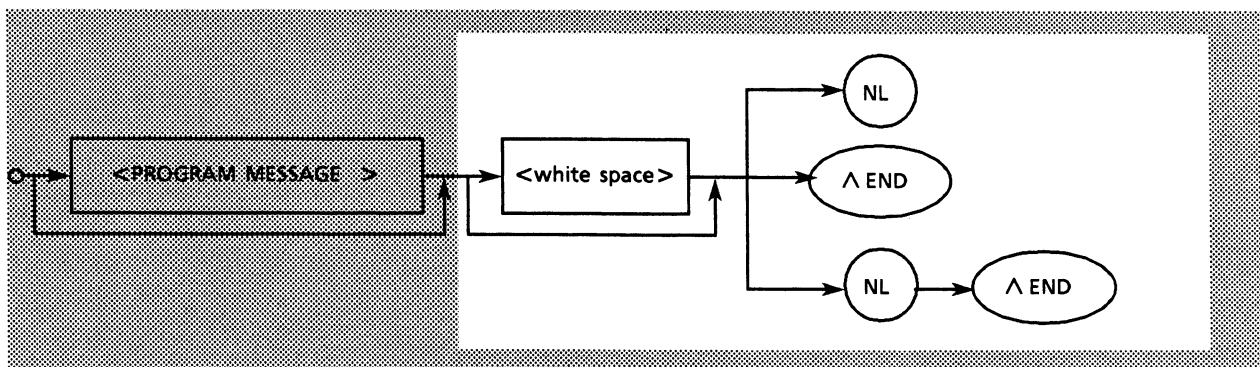


<例2> スペース1個

DTM△∅

(4) PROGRAM MESSAGE TERMINATOR

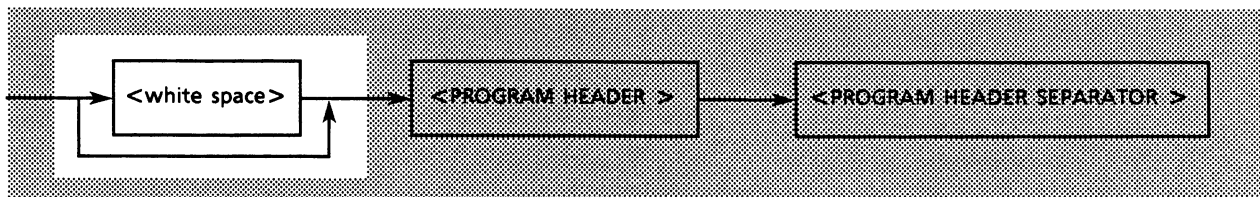
プログラムメッセージの最後には、0個以上のスペース + $\left\{ \begin{array}{l} \text{NL} \\ \text{EOI} \\ \text{NL} + \text{EOI} \end{array} \right\}$ のいずれかを付加
<一般形式>



(5) ヘッダ前置スペース

プログラムヘッダの前に、0個以上のスペースをおくことができます。

<一般形式>

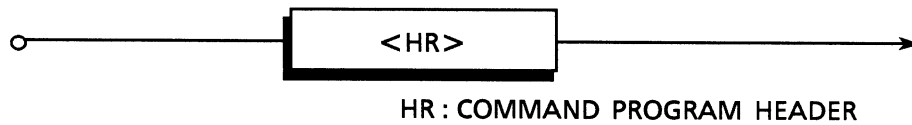


<例> 2番目のプログラムヘッダ SPF の前にスペース1個

DTM ∅ ;△CTM ∅

プログラムコマンドメッセージの一般形式

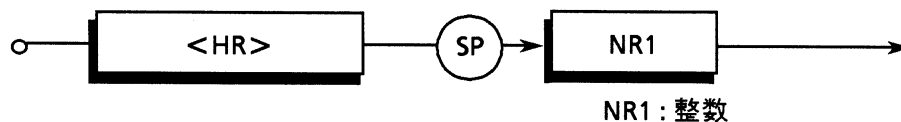
(1) データの指定を伴わないメッセージ



<例>

INI 初期設定

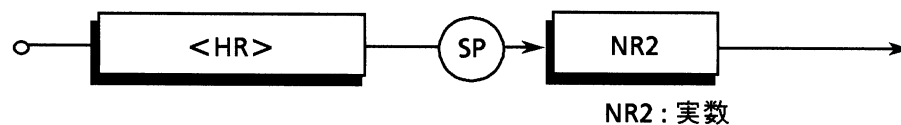
(2) 整数データを伴うメッセージ



<例>

DMS△0	Error 測定表示単位 Ratio の設定
DMS△1	Error 測定表示単位 Count の設定
DMS△2	Error 測定表示単位 EI の設定
DMS△3	Error 測定表示単位 % EF1 の設定
DMS△4	Error 測定表示単位 CLOCK FREQUENCY の設定

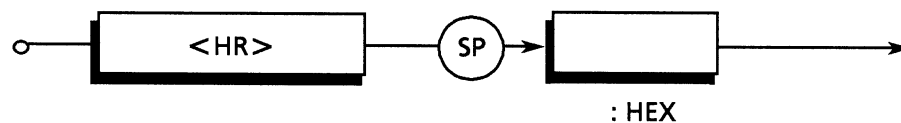
(3) 実数を伴うメッセージ



<例>

DTH△-3.000 入力データスレッシュホールドレベル設定

(4) HEX (16進数) を伴うメッセージ



<例>

BIT△#H FFFF

(5) 複数のプログラムデータを伴うメッセージ



<例>

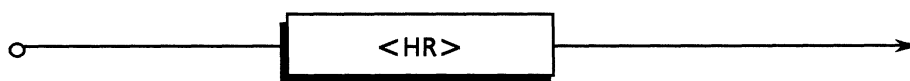
PRD△99,23,59,59

測定時間を99日23時間59分59秒に設定

問合せメッセージの一般形式

問合せ PROGRAM HEADER は、コマンド PROGRAM HEADER の末尾に? を付けます。

(1) 問合せデータの指定を伴わないメッセージ



<例>

DTM?

データ入力終端電圧データを要求する

(2) 問合せデータの指定を伴うメッセージ



<例>

FSH? 1

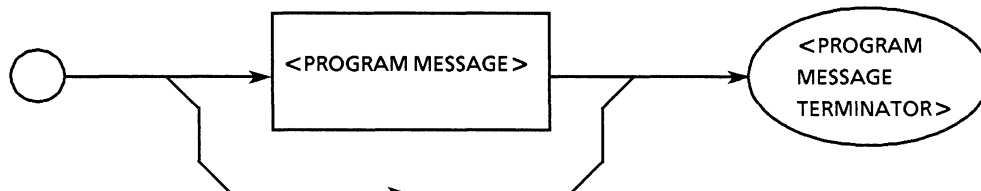
フロッピーディスク中の測定条件が格納されているファイルのうちファイル No. が51以降のファイル情報を要求する。

プログラムメッセージの機能要素

デバイスは、プログラムメッセージの最後にあるターミネータを検出することによりプログラムメッセージをアクセプトします。以下、このプログラムメッセージの各機能要素を説明します。

<TERMINATED PROGRAM MESSAGE>

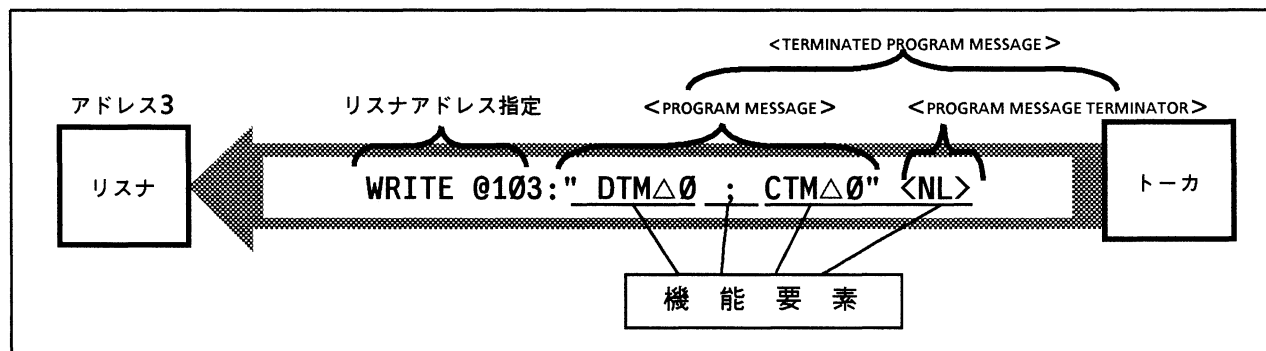
<TERMINATED PROGRAM MESSAGE> は、次のように定義されます。



<TERMINATED PROGRAM MESSAGE> は、コントローラからリスナデバイスに送るに必要なすべての機能要素を満たしたデータ・メッセージです。

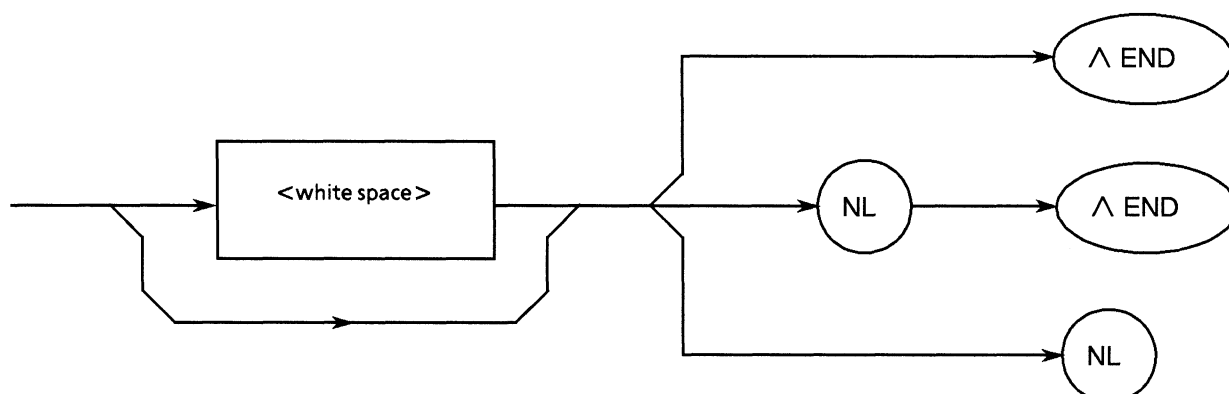
<PROGRAM MESSAGE> の転送を完了させるため、<PROGRAM MESSAGE> の最後には、<PROGRAM MESSAGE TERMINATOR> が付加されます。

<例> WRITE 文で2個の命令を送る <TERMINATED PROGRAM MESSAGE>



<PROGRAM MESSAGE TERMINATOR>

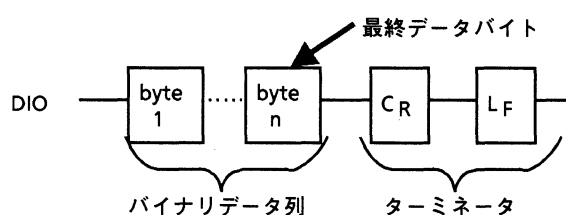
<PROGRAM MESSAGE TERMINATOR>は、次のように定義されます。



<PROGRAM MESSAGE TERMINATOR>は、一つ、またはそれ以上の一定の長さの<PROGRAM MESSAGE UNIT>要素のシーケンスを終了させます。

NL 単一の ASCII コードバイト 0A (10進の10) として定義されます。すなわち、ASCII 制御キャラクタ LF (Line Feed) であって、印字位置を次の行の同じキャラクタ位置へ戻す復帰改行動作を行います。これによって、新しい行からスタートするので NL (New Line) とも呼ばれます。**WRITE @** 文で<PROGRAM MESSAGE>を送る場合、**WRITE @**文は、自動的に CR・LF を送りますので、プログラム作成時に CR・LF コードを発生させる記述は必要ありません。この場合、LF コードのみを発生させるには、下記のステートメントをプログラムの初めで実行します。**TERM IS CHR\$(10)**

END GPIB 管理バスの一つ、EOI ラインを TRUE (LOW レベル) にすることにより、EOI 信号を発生することができます。

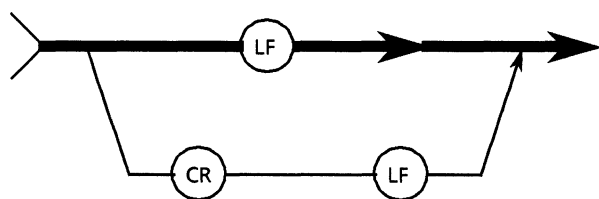


EOI ラインを制御するステートメントに **EOI ON/OFF** 文があります。デフォルトでは、**EOI OFF** を実行した状態と同じで、EOI ラインの制御は行われませんが、あらかじめ **EOI ON** を実行しておけば、**WRITE @** 文の最終データバイト送出時に、ターミネータ LF と同時に EOI 信号が送出されます。



LF を送らないで、END 信号だけで<PROGRAM MESSAGE>を終了させる場合があります。

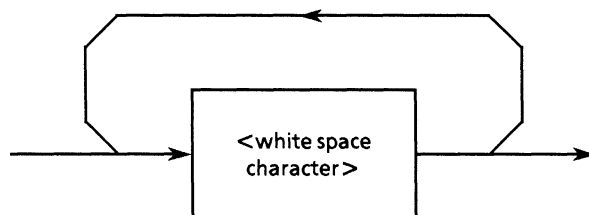
NOTE



CR は、印字位置を同じ行の最初の文字へ戻す復帰動作を行います。リスナ側では、一般には無視されます。しかし、すでに世に出回っている製品の多くは、CR・LF コードを使用している場合もあるので、たいていのコントローラは、CR コードに続いて LF コードを出力する方式が多い。

<white space>

<white space>は、次のように定義されます。

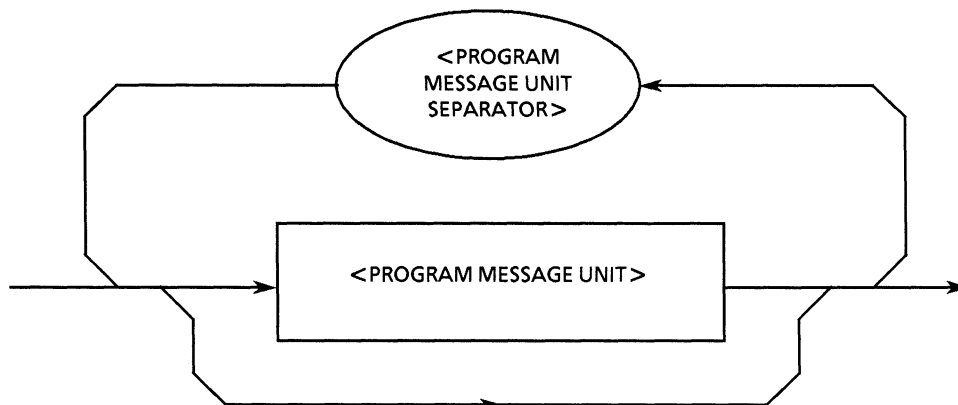


<white space character>は、ASCII コードバイト 00 ~ 09, 0B ~ 20 (10 進数、0 ~ 9, 11 ~ 32) の範囲の中で、単一の ASCII コードバイトとして定義されます。

その範囲は、ニューラインを除き、ASCII コントロール記号およびスペース信号を含みますが、デバイスは、これらを ASCII コントロール記号の意味として解釈せずに、単にスペースとして処理するか、読み飛ばします。

<PROGRAM MESSAGE>

<PROGRAM MESSAGE>は、次のように定義されます。

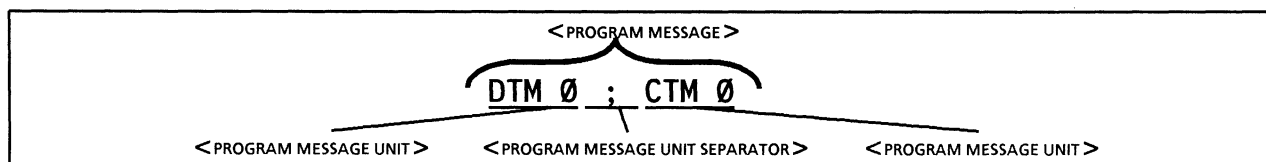


<PROGRAM MESSAGE>とは、それらはゼロであるか、1 個の<PROGRAM MESSAGE UNIT>要素、または、より多くの<PROGRAM MESSAGE UNIT>要素のシーケンスです。<PROGRAM MESSAGE UNIT>要素は、コントローラからデバイスに送られるプログラミング命令か、データを意味しています。<PROGRAM MESSAGE UNIT SEPARATOR>要素は、複数の<PROGRAM MESSAGE UNIT>を区切るためのセパレータとして使用されます。

<例 1> データ入力終端電圧を GND に設定するプログラム・メッセージ

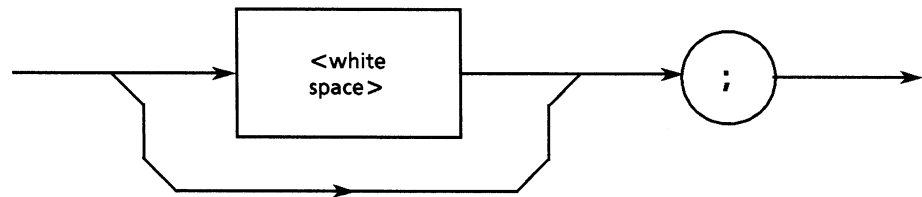
DTM 0

<例 2> 上記設定に続き、クロック入力終端電圧を GND に設定するプログラム・メッセージ

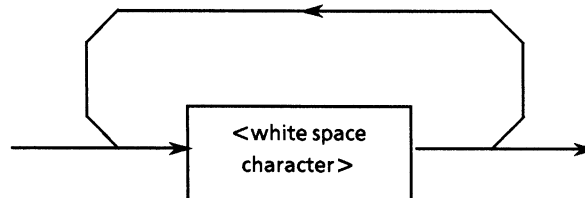


<PROGRAM MESSAGE UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT SEPARATOR>は、次のように定義されます。



<white space>は、次のように定義されます。

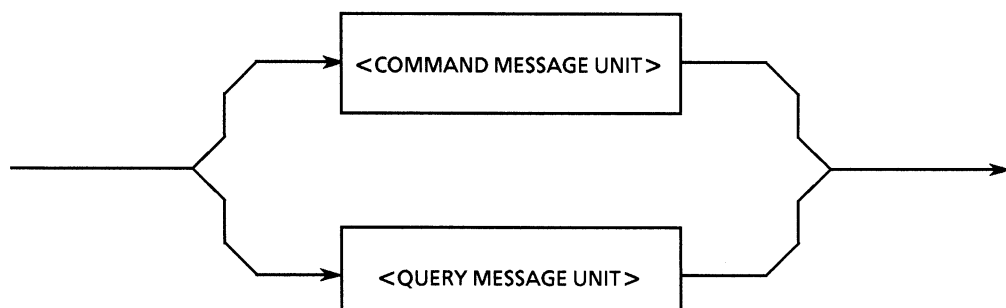


<PROGRAM MESSAGE UNIT SEPARATOR> は、<PROGRAM MESSAGE UNIT>要素のシーケンスを<PROGRAM MESSAGE>の範囲で分割します。

デバイスはセミコロン“;”を<PROGRAM MESSAGE UNIT>のセパレータとして解釈します。したがってセミコロン“;”の前後の<white space character>は読み飛ばされます。ただし、<white space character>は、プログラムを読み易くするためには有用です。なお、セミコロンの後に<white space>がある場合は、次のプログラムヘッダの前におかれた<white space>です。

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT>は、次のように定義されます。

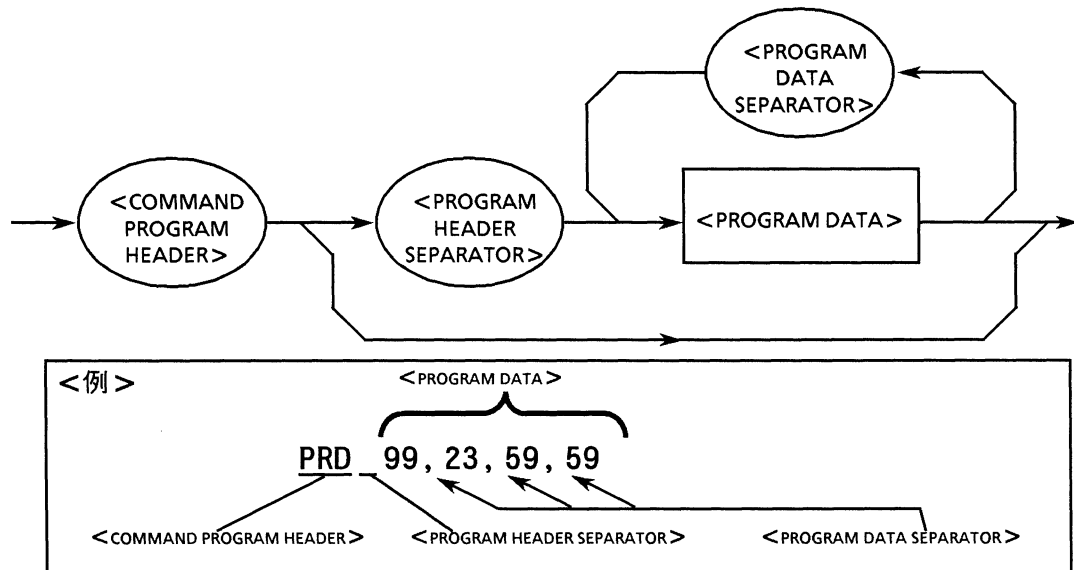


<PROGRAM MESSAGE UNIT>は、デバイスで受信される単一のコマンドメッセージである<COMMAND MESSAGE UNIT>または単一の問合せメッセージである<QUERY MESSAGE UNIT>から成ります。

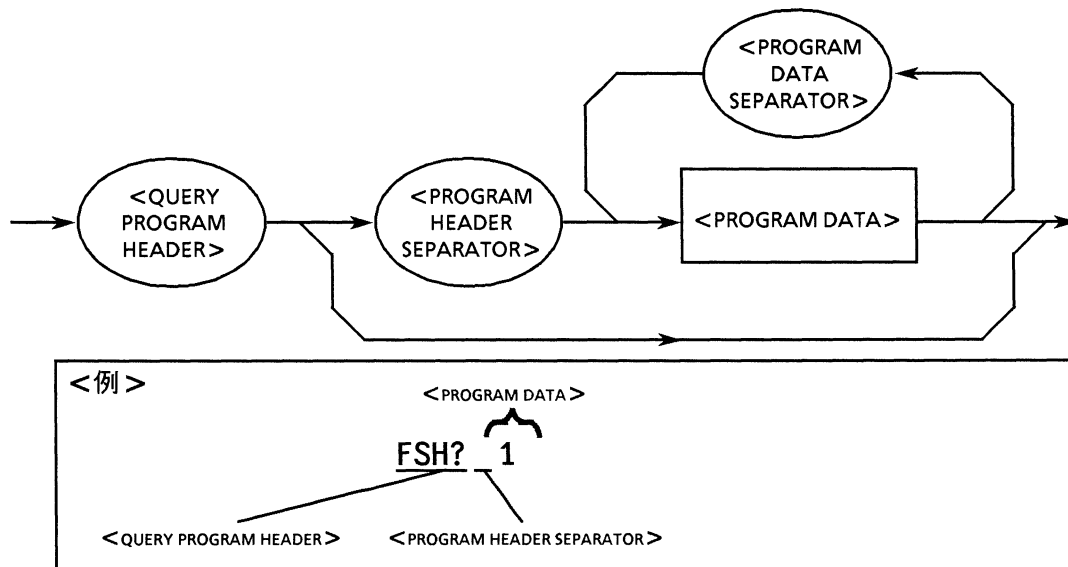
<COMMAND MESSAGE UNIT>と<QUERY MESSAGE UNIT>の詳細は、次ページで説明します。

<COMMAND MESSAGE UNIT>/<QUERY MESSAGE UNIT>

- 1) <COMMAND MESSAGE UNIT>は、次のように定義されます。



- 2) <QUERY MESSAGE UNIT>は、次のように定義されます。

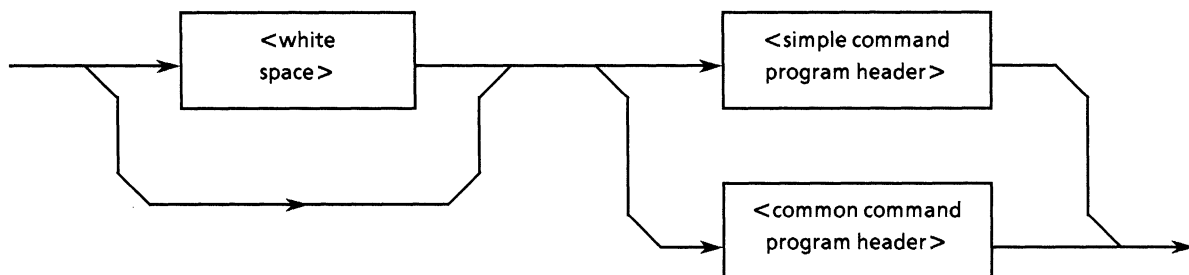


<COMMAND MESSAGE UNIT>も<QUERY MESSAGE UNIT>も、それぞれプログラムヘッダの次にプログラムデータが続く場合は、必ずその間にスペースが1個、セパレータとしてはいります。プログラムヘッダによって、プログラムデータの用途・機能・動作が分かります。プログラムデータが付かない場合は、ヘッダだけでデバイスの中で実行される用途・機能・動作を表します。

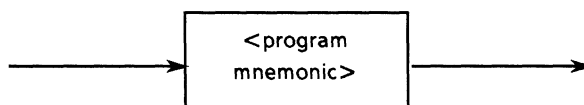
プログラムヘッダの中で、<COMMAND PROGRAM HEADER>は、コントローラからデバイスを制御するコマンドであり、<QUERY PROGRAM HEADER>は、コントローラがデバイスからレスポンスメッセージを受信するため、あらかじめコントローラからデバイスへ送る問合せ用コマンドです。そのヘッダの末尾には、かならず、問合せインジケータ?がつけられるのが特徴です。

<COMMAND PROGRAM HEADER>

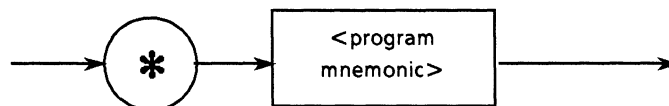
<COMMAND PROGRAM HEADER>は、次のように定義されます。
各ヘッダの前には<white space>をおくことができます。



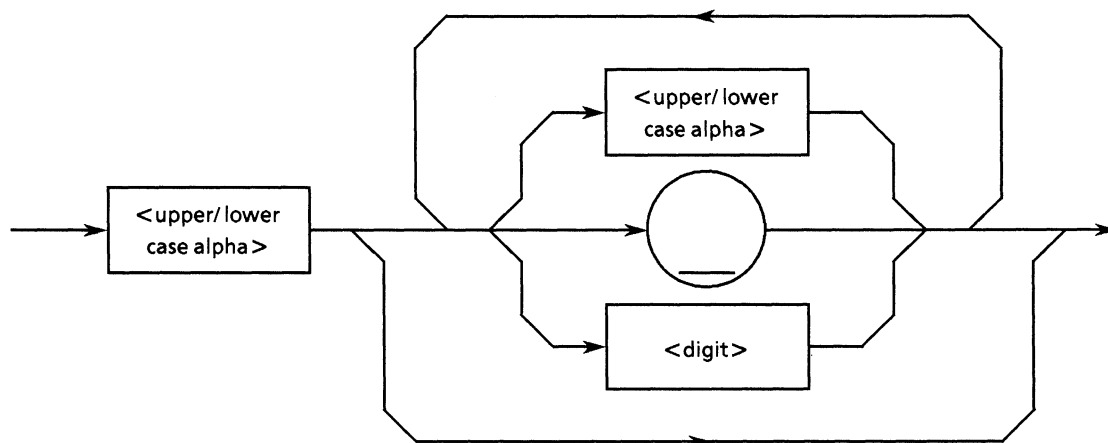
1) <simple command program header>は、次のように定義されます。



2) <common command program header>は、次のように定義されます。



3) <program mnemonic>は、次のように定義されます。



■ <COMMAND PROGRAM HEADER>

デバイスが実行するプログラムデータの用途・機能・動作を表すもので、プログラムデータが付かない場合は、ヘッダだけでデバイスの中で実行される用途・機能・動作を表します。

それらの意味を ASCII コード文字で表したのが<program mnemonic>で、一般には、ニーモニックと呼ばれています。以下、ニーモニックの規定および上記 1), 2) について説明します。

■ <program mnemonic>

ニーモニックの先頭は、必ず英大文字または英小文字で始まります。その後は、英大文字“A～Z”/英小文字“a～z”、アンダーライン“_”、数字“0～9”の任意の組合せが続きます。ニーモニックの最大長は、12文字ですが、一般には3～4文字の英大文字が多用されます。文字と文字の間にスペースは含まれません。

- <upper / lower case alpha> ASCII コードバイト 41 ～ 5A, 61 ～ 7A (10進数、65 ～ 90, 97 ～ 122 = 英大文字 A ～ Z, 英小文字 a ～ z) の範囲の中で、単一の ASCII コードバイトとして規定されます。
- <digit> ASCII コードバイト 30 ～ 39 (10進数、48 ～ 57 = 数値 0 ～ 9) の範囲の中で、単一の ASCII コードバイトとして規定されます。
- (_) ASCII コードバイト 5F (10進数、95 = アンダーライン) を示し、単一の ASCII コードバイトとして規定されます。

■ <simple command program header>

上で述べた<program mnemonic>の規定がそのまま適用されます。

■ <common command program header>

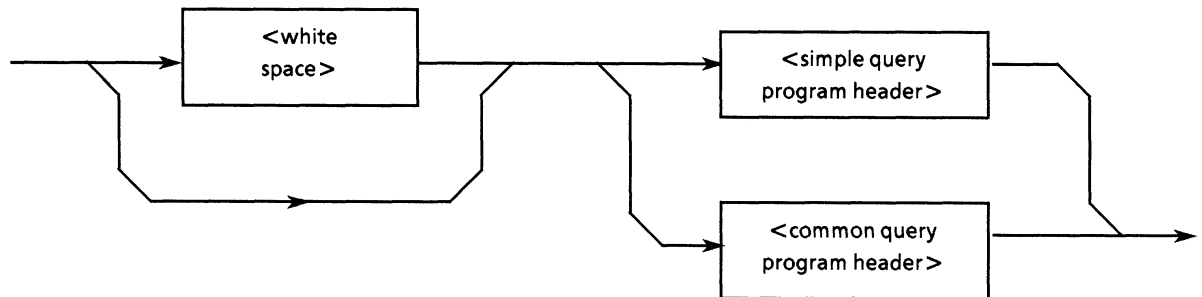
<common command program header>は、<program mnemonic>の前に必ずアスタリスク“*”が付けられます。このコマンドは、バス上に接続されたその他の IEEE 488.2 対応測定器にも共通に適用されるプログラムコマンドであるため common の名が付けられています。

- <例> セレクトコード1の GPIB インタフェースに接続されているアドレス3のデバイスのオペレーション終了をアイドルにし、各デバイスを、決められた固有の状態に初期設定する。

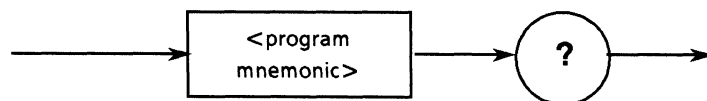
WRITE @103 : "*RST" 引用符“ ”の中が IEEE 488.2 共通コマンド *RST で、上記を実行します。

<QUERY PROGRAM HEADER>

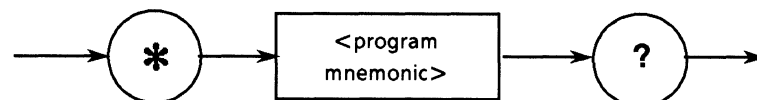
<QUERY PROGRAM HEADER>は、次のように定義されます。
各ヘッダの前には<white space>をおくことができます。



1) <simple query program header>は、次のように定義されます。



2) <common query program header>は、次のように定義されます。



■ <QUERY PROGRAM HEADER>

<QUERY PROGRAM HEADER>は、コントローラがデバイスからレスポンスメッセージを受信するため、あらかじめコントローラからデバイスへ送る問合せ用コマンドです。そのヘッダの末尾には、必ず、問合せインジケータ?がつけられるのが特徴です。

 以上述べた<QUERY PROGRAM HEADER>の形式は、ヘッダの末尾に問合せインジケータ?が付けられる以外は、<COMMAND PROGRAM HEADER>に同じです。

<PROGRAM HEADER SEPARATOR>

<PROGRAM HEADER SEPARATOR>は、次のように定義されます。



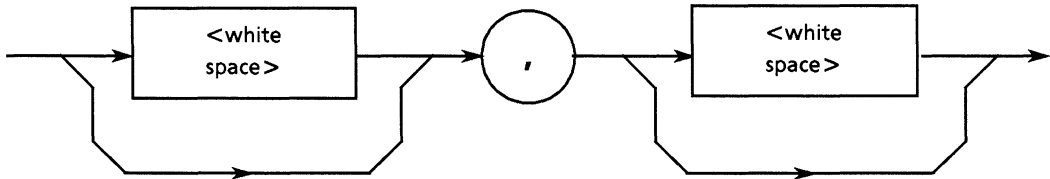
<PROGRAM HEADER SEPARATOR> は、<COMMAND PROGRAM HEADER> または<QUERY PROGRAM HEDADER>と<PROGRAM DATA>の間のセパレータとして使用されます。

プログラムヘッダとプログラムデータの間に複数の<white space character>がある場合は、最初の<white space character>がセパレータとして解釈され、残りの<white space character>は、読み飛ばされます。ただし、<white space character>は、プログラムを読みやすくするためには有用です。

すなわち、ヘッダ・セパレータは、ヘッダとデータの間に1個だけ必ず存在し、プログラムヘッダの終わりであると同時にプログラムデータの始まりを示しています。

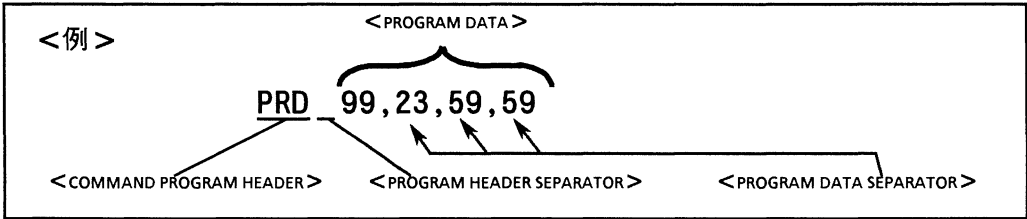
<PROGRAM DATA SEPARATOR>

<PROGRAM DATA SEPARATOR>は、次のように定義されます。



<PROGRAM DATA SEPARATOR> は、<COMMAND PROGRAM HEADER> または<QUERY PROGRAM HEADER>が多数のパラメータを持つ場合に、それらを区切るために使用されます。

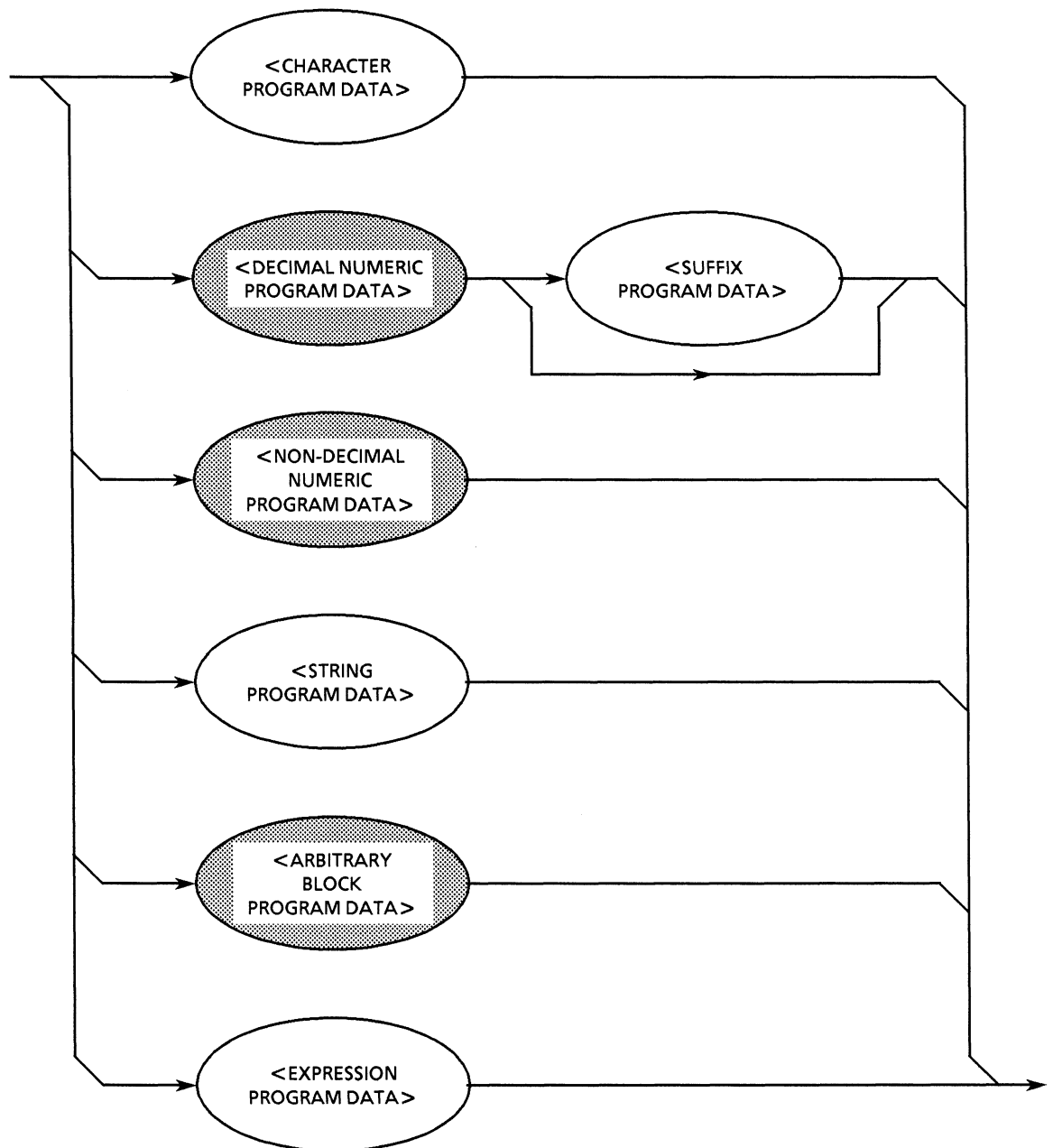
このデータセパレータを使う場合、コンマは必ず必要ですが、<white space character>は、必ずしも必要ではありません。コンマの前または後の<white space character>は、読み飛ばされます。ただし、<white space character>は、プログラムを読みやすくするためには、有用です。



プログラムデータのフォーマット

前述、ターミネイトされたプログラムメッセージのフォーマット体系の中から、機能文法図で示されている<PROGRAM DATA>のフォーマットを説明します。

<PROGRAM DATA>の機能要素は、プログラム・ヘッダに関連したいろいろなタイプのパラメータを伝送するのに使用されます。下図で、それらのプログラムデータの種類を示します。MP1762Dは、この中からアミで囲まれ、白抜きされたプログラムデータをアクセプトします。



<DECIMAL NUMERIC PROGRAM DATA>

<DECIMAL NUMERIC PROGRAM DATA>は、10進で表現される数値定数を伝送するプログラムデータです。10進数値の表現形式には、『整数形式』『固定小数点形式』『浮動小数点形式』の3種類があります。浮動小数点形式について MP1762D では使用していません。

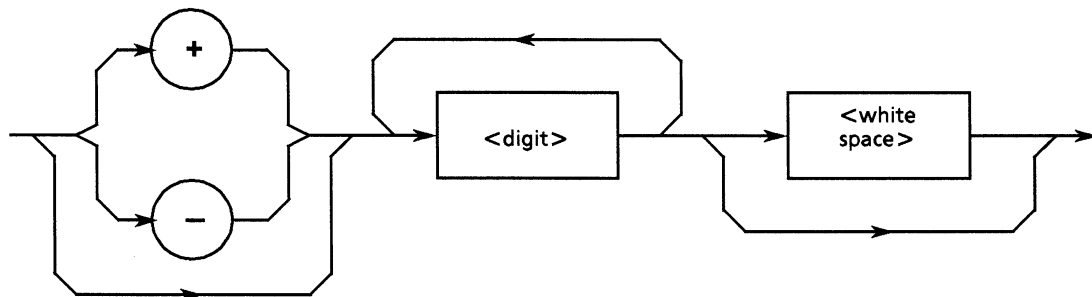
MP1762Dで使用している整数形式、固定小数点形式のプログラムデータ伝送について説明します。

なお、どの形式の伝送においても下記の処理がなされますので注意してください。

- 数値要素の丸め デバイスは、その内部で扱えるよりも桁数の多い<DECIMAL NUMERIC PROGRAM DATA>の要素を受け取った場合は、その数のサインを無視して、4捨5入を行います。
- レンジ外データ <DECIMAL NUMERIC PROGRAM DATA>要素の値がプログラム・ヘッダとの関連において、許されているレンジ外の場合は、実行エラーが報告されます。

(1) 整数形式－NR1伝送

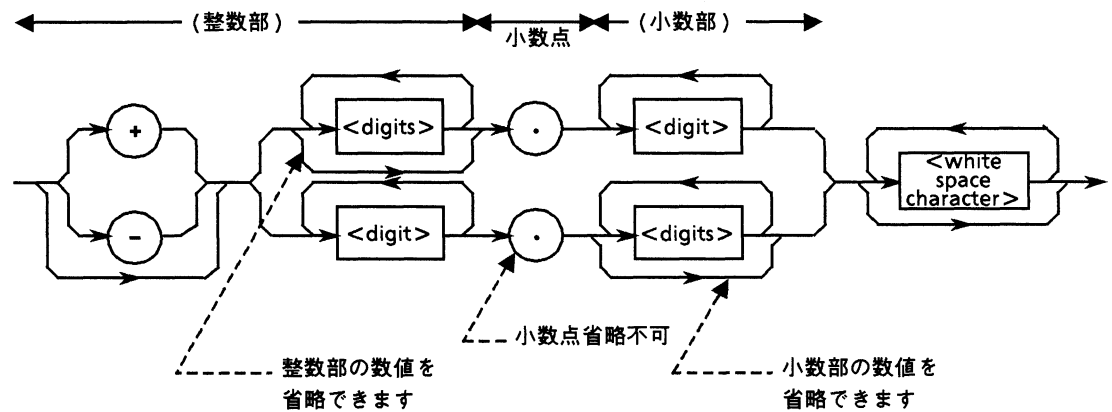
小数点や指数表現を含まない10進数値、すなわち実数の中の整数(NR1)を伝送します。



- 先頭に0挿入可 → 005, +000045
- 符号(+または-)と数字の間にスペース挿入不可 → +5, + △5 (×)
- 数字の後ろにスペース挿入可 → +5△△△
- +符号は、付けても付けなくてもかまいません。 → +5, 5
- 桁区切りにコンマは使用できません。 → 1,234,567 (×)

(2) 固定小数点形式－NR2伝送

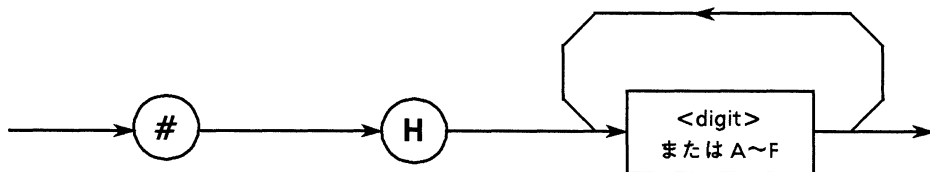
小数点以下の桁を持つ10進数値、すなわち整数および指数表現を除く実数 (NR2) を伝送します。文法図は、(整数部)、小数点、(小数部) から成ります。



- (整数部) は、整数形式の数値表現が適用されます。
- 数字と小数点の間にスペース挿入不可 → +753△.123 (×)
- (小数部) の数字の後ろにスペース挿入可 → +753.123△△△△
- 小数点の前に数値がなくてもかまいません。→ .05
- 小数点の前に符号がおけます。→ +.05, -.05
- 小数点で終わることも可。→ 12.

<NON-DECIMAL NUMERIC PROGRAM DATA>

非10進数値として、16進数値データを伝送するプログラムデータです。非10進データは、必ず#マークから始まります。下図に示すコード化文法図のように定義されます。指定した文字列以外の並びが送られるとコマンド・エラーとなります。



#Hに続く文字は、符号付きでない16進数としてデバイスで受け付けられます。
()内は、対応する10進数を示します。

<例> #HABCD (43,981 D)

6 章

トーカ出力フォーマット

バスのデータモードすなわち**ATN**ラインが偽の時に、システムインタフェースをと
おしてコントローラとデバイス間で送受されるデータメッセージには、プログラム
メッセージとレスポンスメッセージの二つがありますが、この章では、トーカデバ
イスからコントローラへ送出するレスポンスメッセージの書式について説明しま
す。

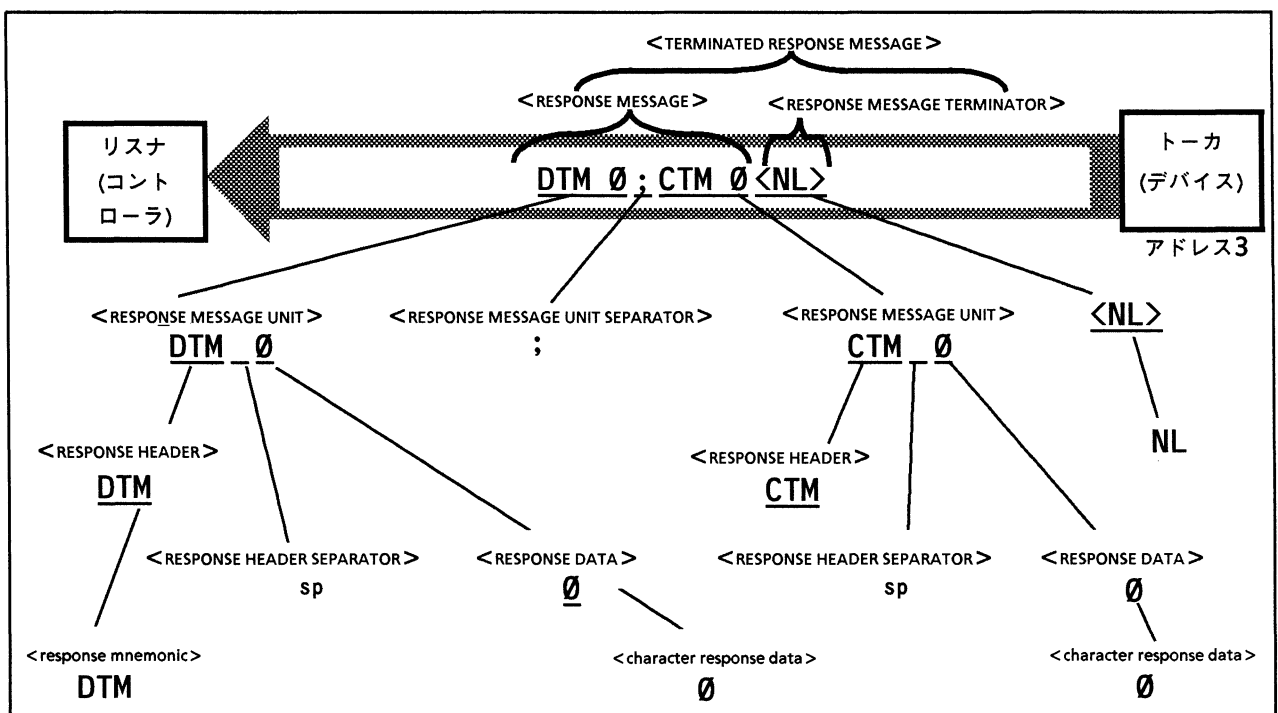
目 次

リスナ入力とトーカ出力フォーマットの文法上の相違点	6-4
レスポンスメッセージの機能要素	6-5
<TERMINATED RESPONSE MESSAGE>	6-5
<RESPONSE MESSAGE TERMINATOR>	6-5
<RESPONSE MESSAGE>	6-6
<RESPONSE MESSAGE UNIT SEPARATOR>	6-7
<RESPONSE MESSAGE UNIT>	6-7
<RESPONSE HEADER SEPARATOR>	6-8
<RESPONSE DATA SEPARATOR>	6-8
<RESPONSE HEADER>	6-8
<RESPONSE DATA>	6-10

6章 トーカ出力フォーマット

レスポンスメッセージの代表的なものとして、測定結果、設定状態、ステータス情報などがあり、ヘッダ付きで返されるレスポンスメッセージとヘッダ無しで返されるレスポンスメッセージがあります。

下図は、データ入力終端電圧問合せメッセージユニット **DTM?** およびクロック入力終端電圧問合せメッセージユニット **CTM?** に対して、それぞれのレスポンスメッセージがヘッダ付き ASCII 文字列として、デバイスからコントローラへ送出されていることを示しています。



以上の動作部分のみをプログラムで示せば、次のとおりとなります。

```
100 WRITE @103:"DTM? "!
```

データ入力終端電圧問合せメッセージユニット

```
110 READ @103:A$!
```

ターミネータ NL が検出されると、レスポンスメッセージ DTM△0 が A\$ へ読み込まれます。

```
120 WRITE @103:"CTM? "!
```

クロック入力終端電圧問合せメッセージユニット

```
130 READ @103:B$!
```

クロック入力終端電圧のレスポンスメッセージ CTM△0 が B\$ へ読み込まれます。

レスポンスメッセージの書式は、プログラムメッセージの場合と同様に、機能を表すことのできる最小レベルの単位まで分割した機能要素のシーケンスから構成されます。上図でカギカッコ<>で囲まれた英大文字が機能要素の例です。機能要素をさらに分割したものをコード化要素と呼びます。上図でカギカッコ<>で囲まれた英小文字がコード化要素の例です。したがって、文法図表記法は、トーカもリスナも同じです。

以下、リスナ装置入力フォーマットとの相違点を中心にトーカ装置の出力フォーマットを説明します。

リスナ入力とトーカ出力フォーマットの文法上の相違点

リスナ装置の入力フォーマットとトーカ装置の出力フォーマットに関する文法上の最も大きな相違点は、

- リスナフォーマット コントローラからのプログラムメッセージをデバイスが容易にアクセプトできるようにするため、柔軟性を持ったプログラム作成が意図されています。したがって、プログラムメッセージに記述上の多少の違いがあっても、それらのプログラムメッセージは、同じ機能を発揮することができます。たとえば、セパレータやターミネータには、**<white space>**を好きなだけジョイントできるため、読みやすいプログラムを作成することができます。
- トーカフォーマット デバイスから出力されるレスポンスメッセージをコントローラが容易にアクセプトできるようにするため、厳格に定められた文法に従って出力メッセージは送り出されます。したがって、上記とは逆にレスポンスメッセージの文法は、一つ機能に対しては一つ表記法しかありません。

下表は、リスナ装置の入力フォーマットとトーカ装置の出力フォーマットの相違点を要約したものです。なお、表中の0/1個以上のスペースは**<white space>**を指します。

項目	リスナ入力プログラムメッセージ文法	トーカ出力レスポンスメッセージ文法
特 性	(柔軟)	(厳格)
英文字	大文字も小文字も同じ意味に使える。 ただしヘッダについては大文字のみ。	大文字のみ
NR3 指数部Eの前後	<u>0個以上のスペース + E/e + 0個以上のスペース</u> ただしサポート無し	大文字 E のみ
NR3 指数部の+符号	省略可能 ただしサポート無し	省略不可
<white space>	セパレータ前後やターミネータの前に複数おける。	不使用
メッセージユニット	①プログラムデータ付き <u>ヘッダ</u> ②プログラムデータ無し <u>ヘッダ</u>	①ヘッダ付き <u>データ</u> ②ヘッダ無し <u>データ</u>
ユニットセパレータ	<u>0個以上のスペース</u> + セミコロン	セミコロンのみ
ヘッダ前置スペース	<u>0個以上のスペース</u> + ヘッダ	ヘッダのみ
ヘッダセパレータ	ヘッダ + <u>1個以上のスペース</u>	ヘッダ + 1個の\$20*
データセパレータ	<u>0個以上のスペース</u> + コンマ + <u>0個以上のスペース</u>	コンマのみ
ターミネータ	<u>0個以上のスペース</u> + $\left\{ \begin{array}{l} \text{NL} \\ \text{EOI} \\ \text{NL} + \text{EOI} \end{array} \right\}$ のいずれか	NL + EOI

* ASCII コードバイト 20(10進数32 = ASCII 文字SP, スペース)

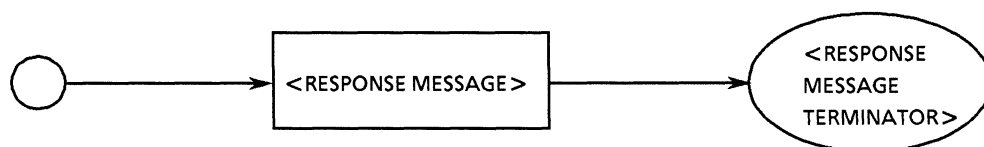
レスポンスメッセージの機能要素

トーカから出力されるレスポンスメッセージは、NL^END 信号でターミネイトされることにより、コントローラでアクセプトされます。以下、このレスポンスメッセージの各機能要素について説明します。

文法図の表記規定については、プログラムメッセージの場合と同じなので、5章を参照してください。また、機能要素やコード化要素の説明についても、プログラムメッセージの場合と重複するものについては、説明を省略していますので必要があれば、5章を参照してください。

<TERMINATED RESPONSE MESSAGE>

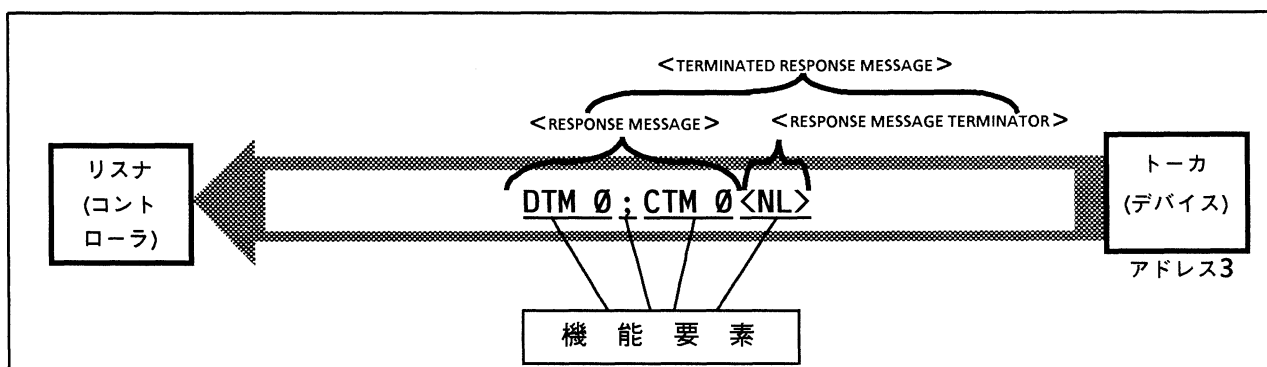
<TERMINATED RESPONSE MESSAGE>は、次のように定義されます。



<TERMINATED RESPONSE MESSAGE>は、トーカデバイスからコントローラに送るに必要なすべての機能要素を満たしたデータ・メッセージです。

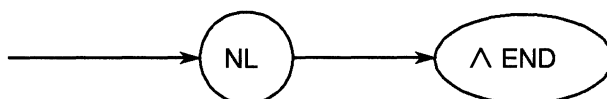
<RESPONSE MESSAGE>の転送を完了させるため、<RESPONSE MESSAGE>の最後には、<RESPONSE MESSAGE TERMINATOR>が付加されます。

<例> 2個のメッセージユニットを連結した<TERMINATED RESPONSE MESSAGE>



< RESPONSE MESSAGE TERMINATOR>

< RESPONSE MESSAGE TERMINATOR>は、次のように定義されます。



<RESPONSE MESSAGE TERMINATOR> は、最後の<RESPONSE MESSAGE UNIT>の次に置かれ、一つ、またはそれ以上の一定の長さの<RESPONSE MESSAGE UNIT>要素のシーケンスを終了させます。

NL^END は次のステートメントをプログラムの開始部分で実行しておけば、最終データバイト送出時にターミネータLFと同時に END 信号として EOI 信号が送出されます。

- NL に対しては、TERM IS CHR\$(10)
- END に対しては、EOI ON

<例> 現在、設定されているアラーム付加を読み出す

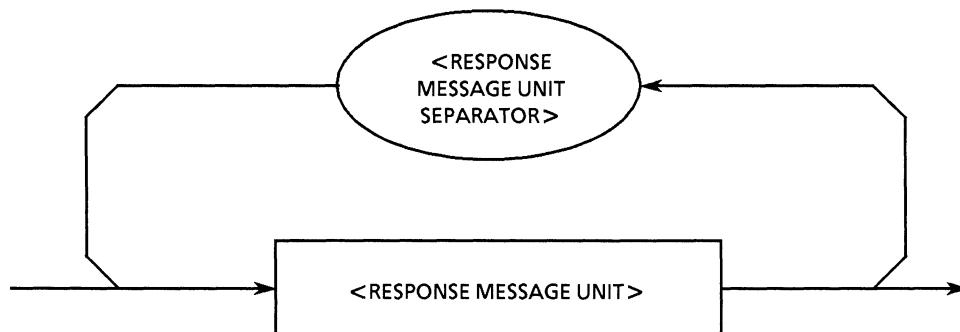
```

10 LET ADR=101
20 TERM IS CHR$(10)! ..... ターミネータコードを LF (New Line) にする
30 EOI ON! ..... 最終データバイト送出時に EOI ラインを TRUE とする EOI 信号を出力
40 WRITE @ADR:"DTM?"! ..... データ入力終端電圧読み出し用問合せ
50 READ @ADR:A$! ..... レスポンスデータの読み込みを EOI 信号によって終了させる
60 PRINT A$
70 END

```

<RESPONSE MESSAGE>

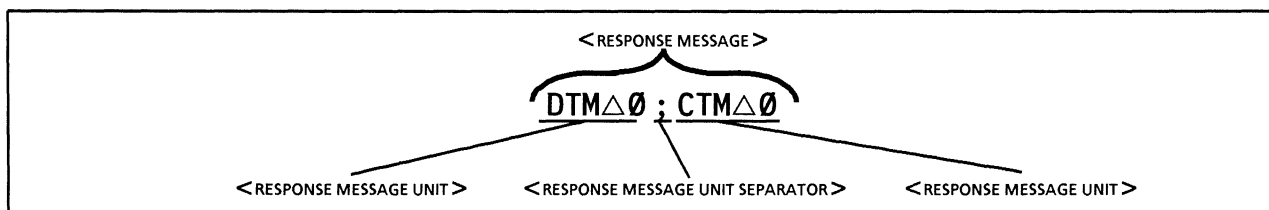
<RESPONSE MESSAGE> は、次のように定義されます。



<RESPONSE MESSAGE> とは、1個の<RESPONSE MESSAGE UNIT>要素、または、より多くの<RESPONSE MESSAGE UNIT>要素のシーケンスです。

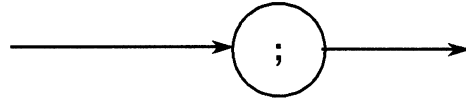
<RESPONSE MESSAGE UNIT>要素は、デバイスからコントローラに送られる単一のメッセージを意味しています。<RESPONSE MESSAGE UNIT SEPARATOR>要素は、複数の<RESPONSE MESSAGE UNIT>を区切るためのセパレータとして使用されます。

<例> データ入力終端電圧およびクロック入力終端電圧にヘッダDTMとCTMを付け、それらのデータを1文字の固定フォーマットで送出する



<RESPONSE MESSAGE UNIT SEPARATOR>

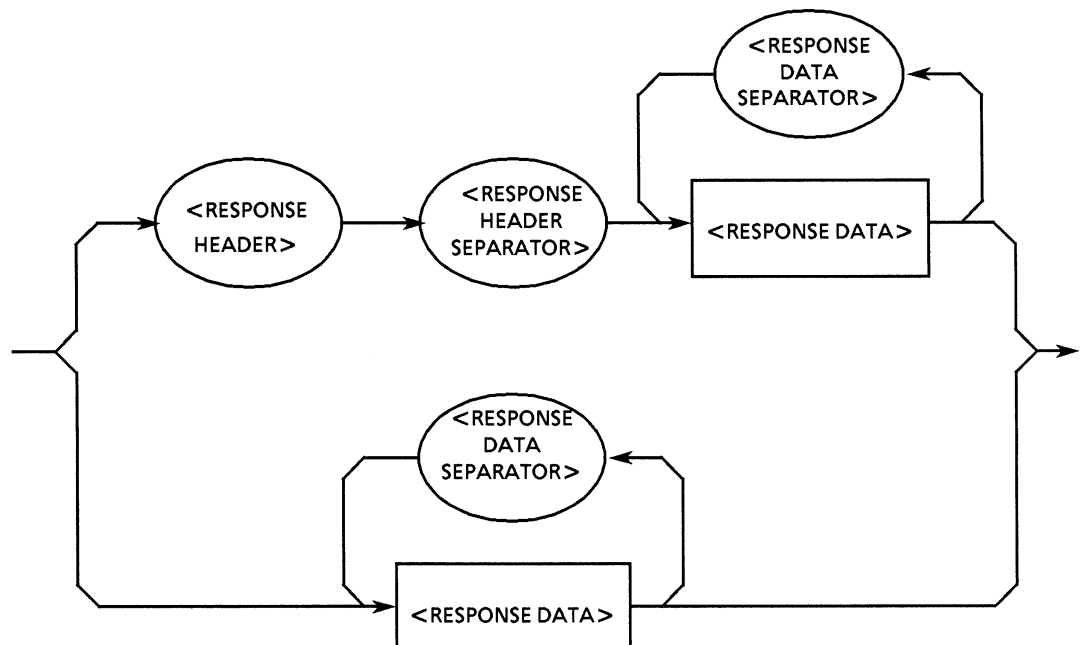
<RESPONSE MESSAGE UNIT SEPARATOR>は、次のように定義されます。



<RESPONSE MESSAGE UNIT SEPARATOR>は、複数の<RESPONSE MESSAGE UNIT>要素のシーケンスを一つの<RESPONSE MESSAGE>として、出力する場合に、<RESPONSE MESSAGE UNIT>要素を<UNIT SEPARATOR>セミコロン“;”で分割します。

<RESPONSE MESSAGE UNIT>

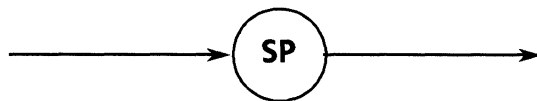
<RESPONSE MESSAGE UNIT>は、次のように定義されます。



<RESPONSE MESSAGE UNIT>は二つの基本的なシンタックスから成っています。前者はヘッダ付きレスポンス・メッセージ・ユニットで、プログラムメッセージでセットした情報の処理結果を正確に返します。後者はヘッダ無しレスポンス・メッセージ・ユニットで、測定結果のデータだけを無駄なく返します。

< RESPONSE HEADER SEPARATOR >

<RESPONSE HEADER SEPARATOR>は、次のように定義されます。



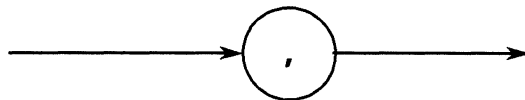
<RESPONSE HEADER SEPARATOR>は、<RESPONSE HEADER>の次に1個のスペースを置き、<RESPONSE HEADER>と<RESPONSE DATA>を分離します。

スペース SP は、ASCII コードバイト 20 (10進数の32) です。

すなわち、ヘッダ付きレスポンスメッセージでは、レスポンス・ヘッダ・セパレータとしてスペースがヘッダとデータの間に1個だけ必ず存在し、レスポンス・ヘッダの終わりであると同時にレスポンスデータの始まりを示しています。

< RESPONSE DATA SEPARATOR >

<RESPONSE DATA SEPARATOR>は、次のように定義されます。



<RESPONSE DATA SEPARATOR>は、複数の<RESPONSE DATA>を出力する場合に、データとデータの間に置き、それらを区切るために使用されます。

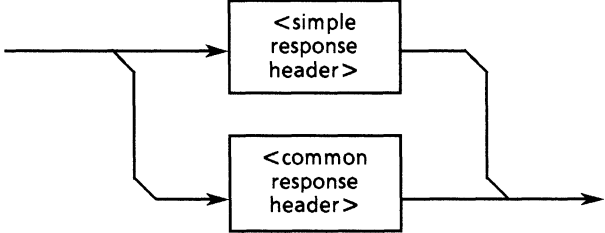
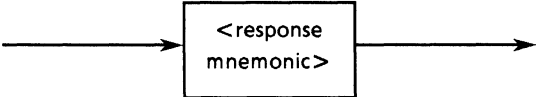
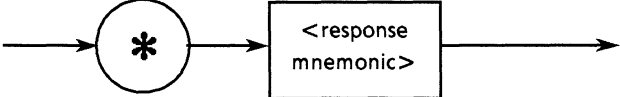
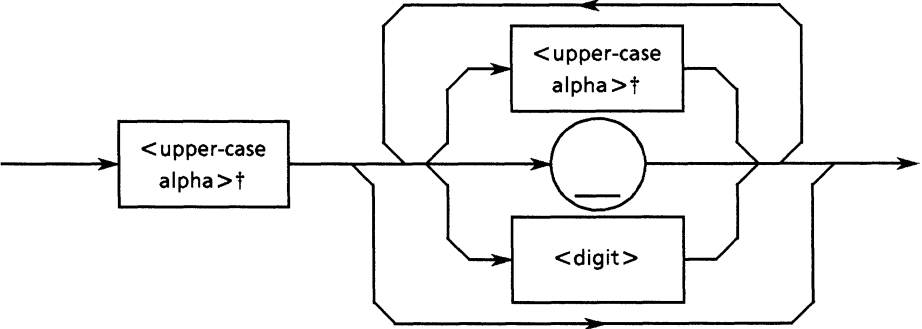
< RESPONSE HEADER >

<RESPONSE HEADER>は、フォーマット表現上においては、次の3点を除き、<COMMAND PROGRAM HEADER>に同じです。

- ① <response mnemonic>で使用文字が定められており、その中で英文字については、大文字のみを使用すること。その他は<program mnemonic>に同じ。
- ② プログラム・ヘッダの前にスペースがおけましたが、レスポンス・ヘッダ前には、スペースはおけません。
- ③ プログラム・ヘッダの後には、複数のスペースがおけましたが、レスポンス・ヘッダ後には、1個のスペースしかおけません。

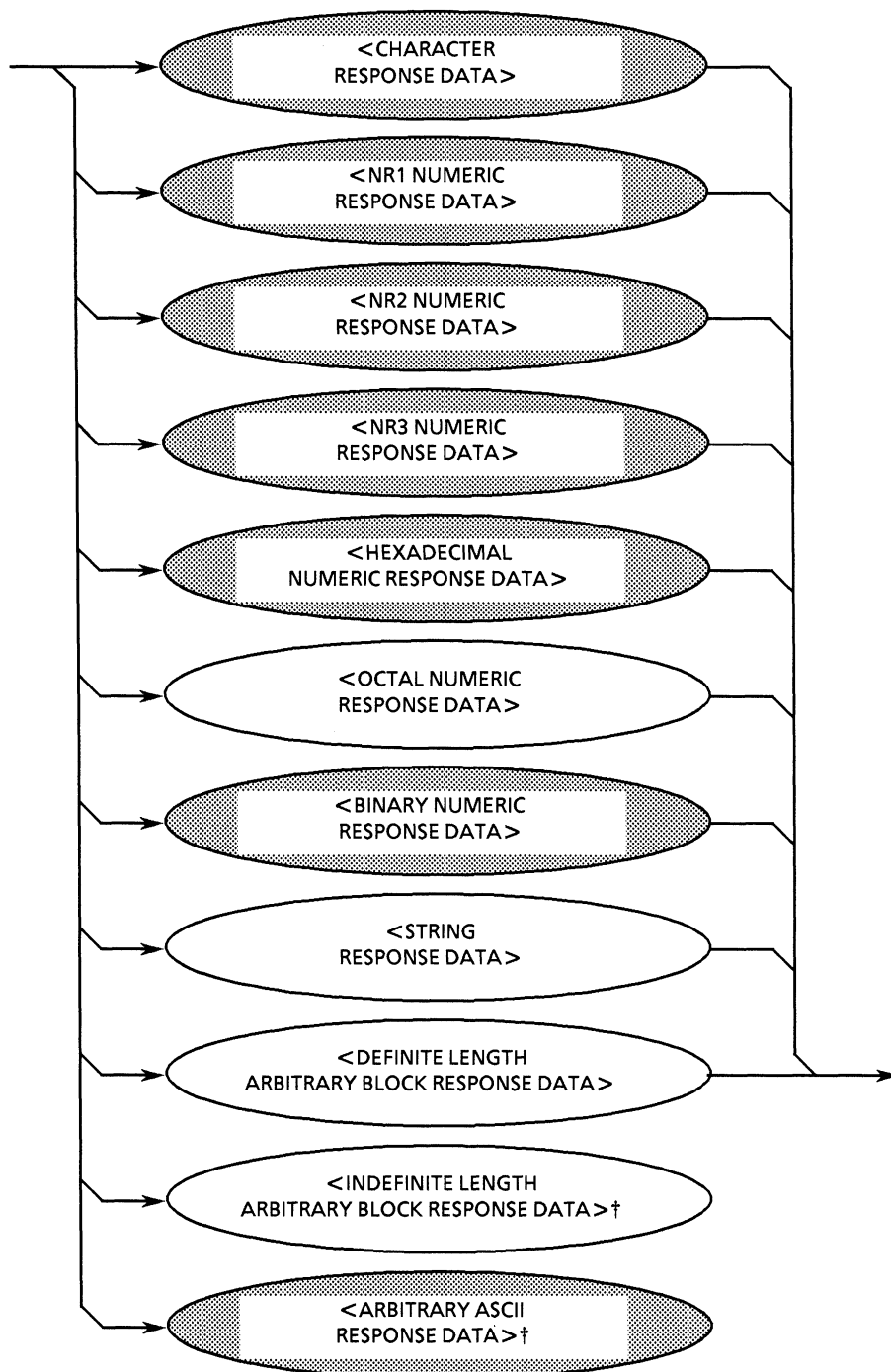
次ページで、<response mnemonic>までを一括して示します。

( <response mnemonic>で使用する文字の中で、英字は大文字のみとなりますが、その他は<program mnemonic>に同じです)

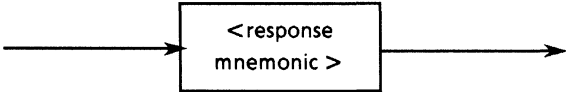
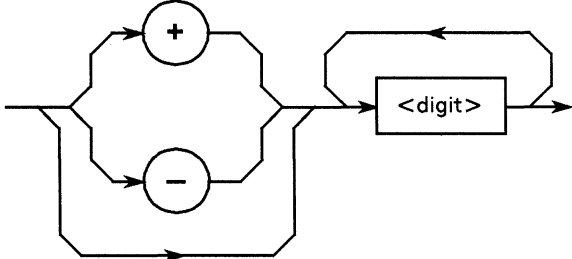
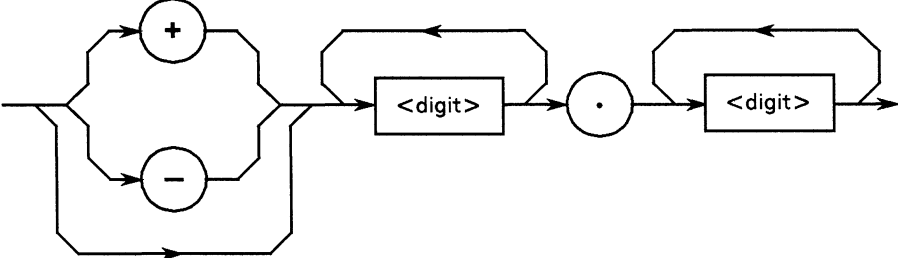
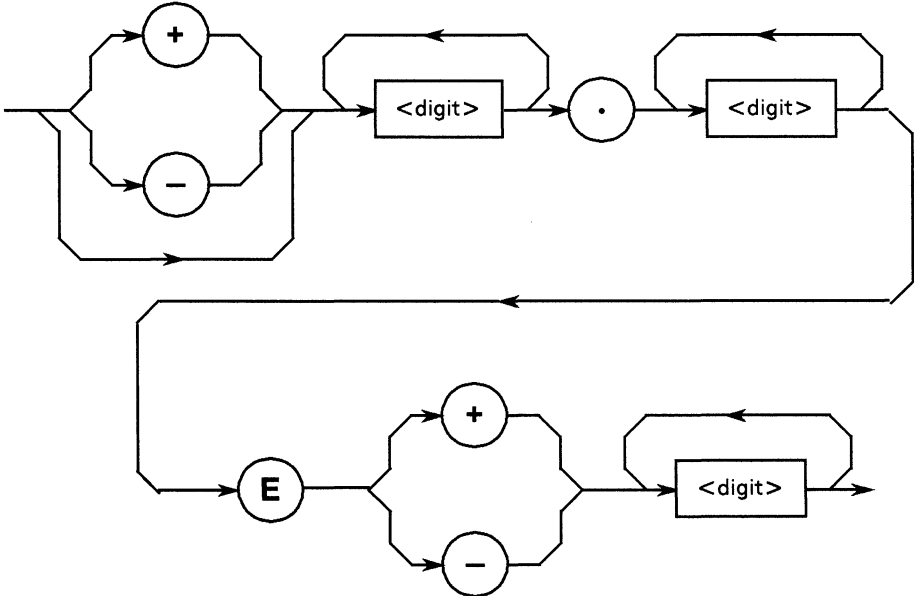
要素	機能
RESPONSE HEADER	<p>ヘッダは、レスポンス・データの機能を表すもので、英大文字から始まる12文字以内の英大文字、数字、アンダーラインのいずれか、それらの組合せから成る<response mnemonic>で、その語義が示されます。</p>  <p>1) <simple response header>は、次のように定義されます。</p>  <p>2) <common response header>は、次のように定義されます。</p>  <p>3) <response mnemonic>は、次のように定義されます。</p>  <p>†<upper-case alpha> ASCIIコードバイト41～5A (10進数65～90 = 英大文字A～Z)</p>

<RESPONSE DATA>

<RESPONSE DATA>は11種類あり、その中でMP1762Dは、アミで囲まれているレスポンスデータをコントローラへ送出します。どのレスポンスデータが返されるかは、問い合わせメッセージによって決定されます。



† <INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> および <ARBITRARY ASCII RESPONSE DATA> は、それ自身の最後のデータバイトの次はNL&ENDでターミネイトされます。

要素	機能
<p>(1) CHARACTER RESPONSE DATA</p> <p><例> AAT2_AUTO AAT2_MANUAL</p>	<p><response mnemonic>と同じ文字列の構成から成るデータです。したがって、文字列の先頭は必ず英大文字から始まり、文字列の長さは12文字以内です。数値パラメータの使用は適当ではありません。</p> 
<p>(2) NR1 NUMERIC RESPONSE DATA</p> <p><例> 123 +123 -1234</p>	<p>整数形式データ、すなわち小数点や指数表現を含まない整数の10進数値</p> 
<p>(3) NR2 NUMERIC RESPONSE DATA</p> <p><例> 12.3 +12.34 -12.345</p>	<p>固定小数点形式データ、すなわち整数および指数表現を除く10進数値</p> 
<p>(4) NR3 NUMERIC RESPONSE DATA</p> <p><例> 12.3E + 4 +12.34E - 5 -12.345 E + 6</p> <ul style="list-style-type: none">● Eに小文字は使えません。● Eの前後のスペースは不可● 指数部の+は省略できません。● 仮数部の+は省略可。	<p>浮動小数点形式データ、すなわち指数表現の桁を持つ10進数値</p> 

要素	機能
<p>(5) HEXADECIMAL NUMERIC RESPONSE DATA</p> <p><例> #HABC123 #H2DC3 #H8301</p>	<p>16進数値データです。</p>
<p>(6) OCTAL NUMERIC RESPONSE DATA</p> <p><例> #Q37 #Q26703 #Q30562</p>	<p>8進数値データです。</p>
<p>(7) BINARY NUMERIC RESPONSE DATA</p> <p><例> #B011101 #B1011 #B1011</p>	<p>2進数値データです。 ※MP1762Dでは“+”、“-”を含むデータを扱っています。</p>

要素	機能
(8) STRING RESPONSE DATA <例> "This is a text" "Say," "Hello" "."	<p>ASCII 7ビット・コードのすべてが使えます。文字列の両端は、必ずダブル引用符で囲まれます。文字列中のダブル引用符は、引用符一個に付き、同じ引用符が2個連続した2連引用符となります。CR, LF, スペースが使えるので、テキストをプリンタやCRTへ出力するのに適しています。</p>
(9) DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA <例> 11256099Dを 4バイトで転送 ↓ #1400ABC123	<p>固定長8ビットバイナリのブロックデータです。大量なデータ、8ビット拡張 ASCII コード、非ディスプレイデータなどの転送に適しています。</p>
(10) INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA <例> - 250, - 50, 120, ...を不定長転送 ↓ #0FF06FFCE0078	<p>不定長8ビットバイナリのブロックデータです。このため、最初のデータの前に#0をおきます。また、最後のデータの次はNL^ENDでターミネートされます。</p>
(11) ARBITRARY ASCII RESPONSE DATA <例1> <ASCII Byte> <ASCII Byte> NL^END <例2> NL^END	<p>NL 文字を除いた ASCII データバイトを区切らないで送ります。このため、最後のデータの次はNL^ENDでターミネートされます。</p>

7 章

共通コマンド

この章では、**IEEE488.2**で規定されている共通コマンドと共通問合せコマンドについて説明します。これら共通コマンドは、インタフェース・メッセージで使用するバスコマンドではありません。デバイスメッセージと同様に、共通コマンドは、バスのデータモードすなわち**ATN**ラインが偽の時に使用されるデータメッセージの一つであって、**IEEE488.2**対応機種であれば、他社の製品を含むすべての測定器に共通に使用することができます。**IEEE488.2**共通コマンドは、必ず*で始まります。

目 次

MP1762D サポート共通コマンドのグループ機能別分類	7-3
サポートコマンドの分類とリファレンス	7-4

7 章 共通コマンド

MP1762D サポート 共通コマンドのグループ機能別分類

MP1762D サポート IEEE 488.2 共通コマンドのグループ機能別分類を以下に示します。
サポート対象コマンドについては、アルファベット順にリストされています。

サポートコマンドの分類とリファレンス

前ページで示した **MP1762D** に対する当社サポート対象コマンドの機能グループ別の説明を下表に示します。各コマンドの説明は、次ページからアルファベット順で示します。

グループ	グループ別機能	ニーモニック
システムデータ	GPIO システムに接続されているデバイス固有の情報、たとえば、そのデバイスの製造メーカー名・形名・シリアル番号などのデータを知ることができます。	*IDN?
内部オペレーション	デバイス内部の制御：① デバイスをレベル 3 でリセット ② デバイス内部のセルフテストとエラー有無の検知	*RST *TST?
同期	デバイスとコントローラの同期を① サービスリクエスト待ち ② デバイスの出力キュー応答待ち ③ シーケンシャル実行の強制により行います。	*OPC *OPC? *WAI
ステータス & イベント	ステータスバイトは、ステータス・サマリメッセージによって構成されています。そのメッセージの個々のサマリビットは、標準イベントレジスタ、出力キュー、および拡張イベントレジスタまたは拡張キューから供給されます。そこで、これらのレジスタやキューにあるデータをセット・クリアー・有効化・無効化、さらにはレジスタの設定状況を問合せによって知るため、4個のコマンド、5個の問い合わせが用意されています。	*CLS *ESE *ESE? *ESR? *PSC *PSC? *SRE *SRE? *STB?
デバイストリガ	IEEE488.2 の GET バスコマンドをデバイスが受信したとき、実行すべきコマンドシーケンスを定義します。	*TRG
オプション情報	実装されているオプション情報を知ることができます。	*OPT?

*CLS Clear Status Command

(ステータスバイト・レジスタのクリア)

■ 書 式

*CLS

■ 使用例

```
30 WRITE @103:"*CLS"
40 WRITE @103:"DTM△0;CTM△0;*CLS"
```

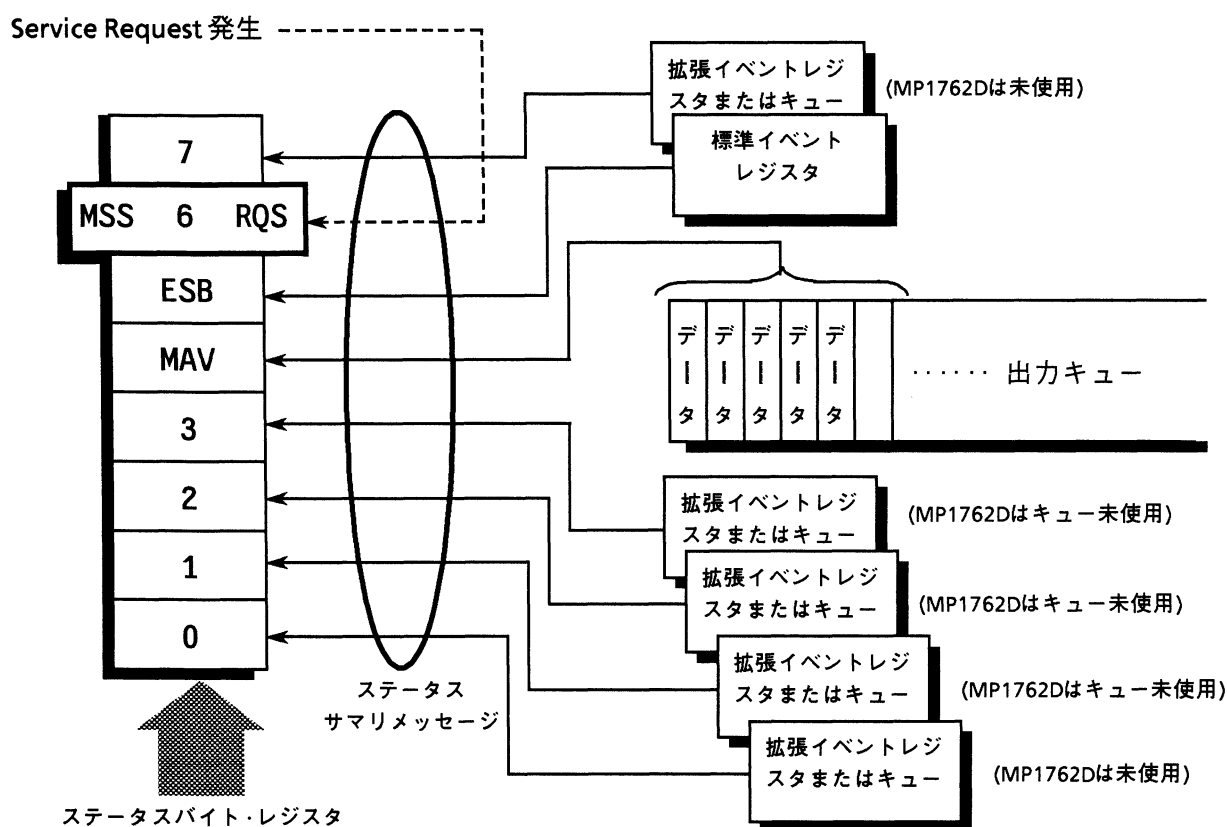
■ 解 説

***CLS** 共通コマンドは、出力キューとその **MAV** サマリメッセージを除くすべてのステータスデータ構造(すなわち、これらのイベントレジスタおよびキュー)をクリアし、これに応じてそれらに対応するサマリメッセージもクリアします。

出力キューとその **MAV** サマリメッセージも、下例の場合はクリアされます。

```
30 WRITE @103:"DTM△0;CTM△0"
40 WRITE @103:"*CLS;DTM?"
```

すなわち、<PROGRAM MESSAGE TERMINATOR>の後、そして何らかの<Query MESSAGE UNIT>要素の前に***CLS** コマンドを送出すると、すべてのステータスバイトはクリアされます。この方法により出力キューは、すべての未読み出しのメッセージもクリアされます。なお、各イネーブル・レジスタの設定値については、***CLS** によって影響されません。



*ESE コマンド/問合せ

*ESE Standard Event Status Enable Command

(標準イベントステータス・イネーブルレジスタのセットまたはクリア)

■ 書 式

***ESE** <HEADER SEPARATOR> <DECIMAL NUMERIC PROGRAM DATA>

本書式において、

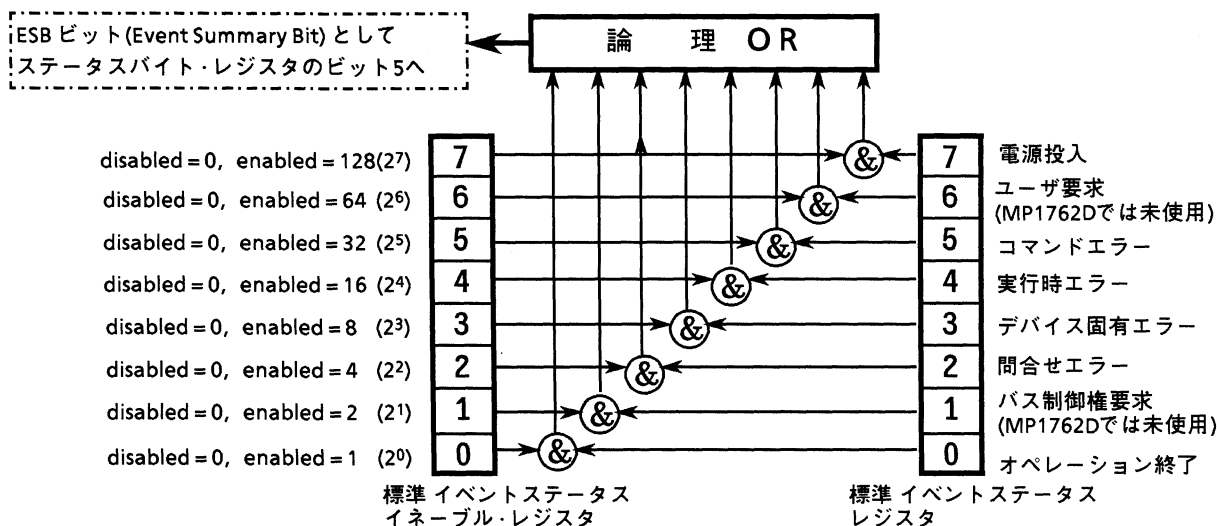
<DECIMAL NUMERIC PROGRAM DATA> = 0 ~ 255の整数に丸められ数値(2を底としてバイナリで重み付けされていること)

■ 使用例

WRITE @103:"*ESE 20"! イネーブルレジスタのビット2,4をセットします。

■ 解 説

標準イベントステータス・イネーブルレジスタのビット0, 1, 2, 3, 4, 5, 6, 7に対応する値 $2^0=1$, $2^1=2$, $2^2=4$, $2^3=8$, $2^4=16$, $2^5=32$, $2^6=64$, $2^7=128$ の中から、**enabled**にしたいビットを選択したときのビット桁値の総和がプログラムデータとなります。**disabled**にしたいビット桁値は、0となります。



*ESE? Standard Event Status Enable Query

(標準イベントステータス・イネーブルレジスタの現在値をレスポンス)

■ 書 式

***ESE?**

■ 使用例

ESE 20** 実行後、ESE?**を送ると、**20**がレスポンスされます。

■ 解説

標準イベントステータス・イネーブルレジスタの値である **NR1**を返します。

■ レスポンスメッセージ

NR1 = 0 ~ 255

*ESR? **Standard Event Status Register Query**

(標準イベントステータス・レジスタの現在値を返す)

■ 書 式

*ESR?

■ 使用例

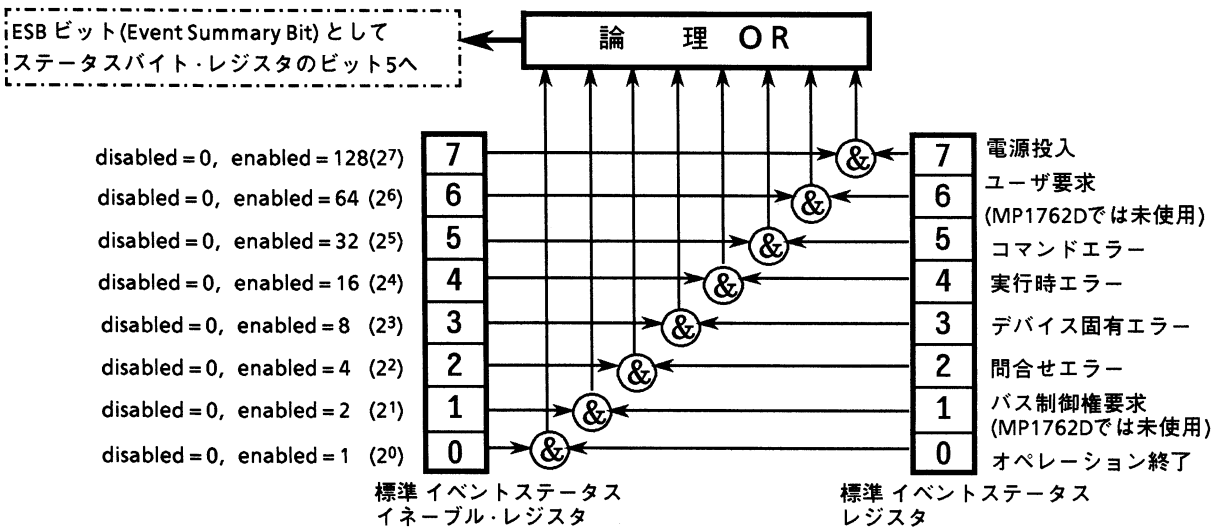
```
30 WRITE @103:"*ESR?"
40 READ @103:STEvet
50 PRINT STEvet
```

■ レスポンスメッセージ

NR1=0 ~ 255

■ 解 説

標準イベントステータス・レジスタの現在値をNR1で返します。標準イベントステータス・レジスタのビット0, 1, 2, 3, 4, 5, 6, 7に対応する値 $2^0=1$, $2^1=2$, $2^2=4$, $2^3=8$, $2^4=16$, $2^5=32$, $2^6=64$, $2^7=128$ に対して、標準イベントステータス・イネーブルレジスタによって**enabled**されているビットに対応するビット桁値の総和がNR1となります。値はレスポンスが読み取られると(たとえば、行40)、このレジスタはクリアされます。



*IDN? 問合せ

*IDN? Identification Query

(製品のメーカー名・形名等を返す。)

■ 書 式

*IDN?

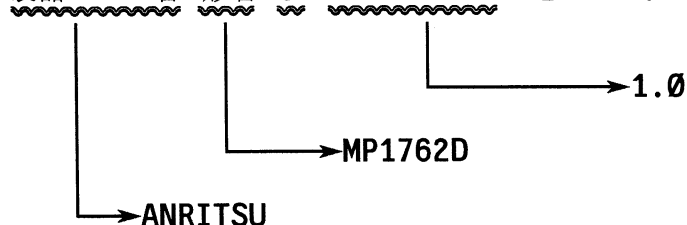
■ 使用例

30 WRITE @103:"*IDN?"

40 READ @103:IDN\$! メーカー名・形名等を格納

■ 解 説

製品メーカー名・形名・0・バージョンNo.を返します。



形名 **MP1762D**の製造メーカーがアンリツで、ソフトウェアまたはハードウェアのバージョン No. が1の場合、***IDN?** 共通問い合わせをデバイスに送ると、上記に示した4つのフィールドから成るレスポンスメッセージを返します。

- ①フィールド1 製品メーカー名(当社の場合、ANRITSU)
- ②フィールド2 形名
- ③フィールド3 (常に0)
- ④フィールド4 バージョン No.

■ レスポンスメッセージ

上記4つのフィールドをコンマで区切って構成したレスポンスメッセージを<ARBITRARY ASCII RESPONSE DATA>で送ります。

<フィールド1>,<フィールド2>,<フィールド3>,<フィールド4>

解説の例では、

ANRITSU,MP1762D,0,1.0となります。

レスポンスメッセージの全長は、≤72文字です。

*OPC Operation Complete Command

(デバイス動作が終了すると、標準イベントステータス・レジスタのビット0をセット)

■ 書 式

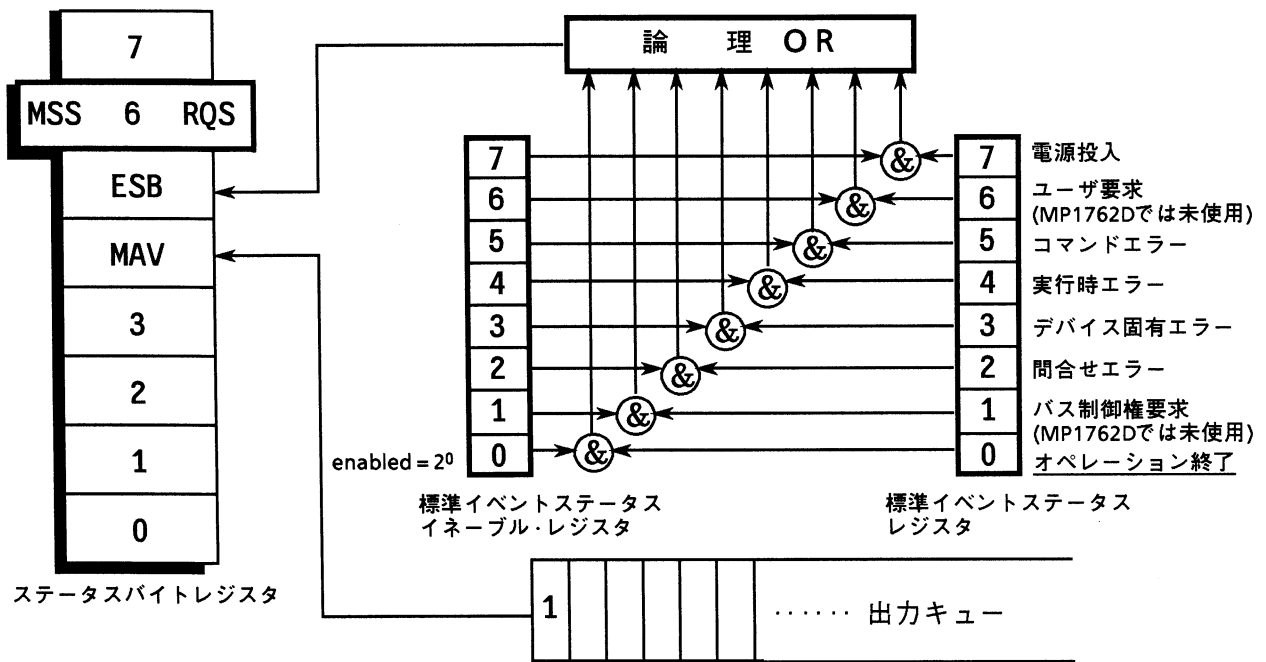
*OPC

■ 使用例

WRITE @103:"*OPC"

■ 解 説

選択されたペンディング中のデバイス動作がすべて終了したら、標準イベントステータスレジスタの中のビット0、すなわち『オペレーション終了ビット』をセットします。



*OPC? Operation Complete Query

(デバイス動作が終了すると、出力キューに"1"を立て MAV サマリメッセージを発生させる。)

■ 書 式

*OPC?

■ 使用例

WRITE @103:"*OPC?"

■ 解説

選択されたペンディング中のデバイス動作がすべて終了したら、出力キューに"1"を立て MAV サマリメッセージが発生するまで待ちます。

■ レスポンスメッセージ

1を<NR1 NUMERIC RESPONSE DATA>で返します。

***OPT?** 問合せ

***OPT? Option Query**

(実装されている Option 情報を出力)

■ 書 式

***OPT?**

■ 使用例

```
30 WRITE @103:"*OPT?"  
40 READ:OPTI
```

■ 解 説

***OPT** 問い合わせは、実装されている **Option** 情報を出力します。

■ レスポンスメッセージ

<ARBITRARY ASCII RESPONSE DATA>

オプションに対応したキャラクタをカンマで区切って返します。

0 : オプション無し

OPT01 : MP1762D-01 Error Analysis

***PSC Power-on Status Clear Command**

(ステータス報告部のイネーブルレジスタを電源ON時にクリアするかどうかを決定)

■ 書 式

<HEADER><HEADER SEPARATOR><DECIMAL NUMERIC PROGRAM DATA>

本書式において、 <HEADER> = ***PSC**

<DECIMAL NUMERIC PROGRAM DATA> = **0** (クリアしない場合)

0を除く1～255の整数 (クリアする場合)

■ 使用例

WRITE @103:"*PSC 0;*SRE 32;*ESE 128"!クリアしないでSRQ ON

■ 解 説

***PSC** 共通コマンドはステータス報告部のサービスリクエスト・標準イベントステータスおよびパラレルポールの各イネーブルレジスタを電源ON時にクリアするかどうかを決定します。

<DECIMAL NUMERIC PROGRAM DATA>の値がパワーON・ステータス・フラグの真/偽を決定します。この値が整数値**0**に丸められる場合は、フラグは偽にセットされますので、イネーブルレジスタはクリアされません。***PSC 0**によって、デバイスは電源ON後に**SRQ**を発生することができます。上記の例は、電源ONのイベントをコントローラに伝えることを示しています。

<DECIMAL NUMERIC PROGRAM DATA>の値が**0**でない整数1～255に丸められる場合は、フラグは真にセットされますのでイネーブルレジスタはクリアされます。***PSC 1**を実行すると、パワーONクリアを可能にしますが、**SRQ**の発生は許されません。

プログラムデータの整数値が0～255の範囲外であれば、実行エラーが発生します。

***PSC? Power-on Status Clear Query**

(パワーONステータスフラグの真/偽を返す)

■ 書 式

***PSC?**

■ 使用例

30 WRITE @103:"*PSC?"

40 READ:POWF

■ 解 説

***PSC?** 共通問合せを実行すると、パワーONステータスフラグが真であれば**1**を返し、偽であれば**0**を返します。

■ レスポンスメッセージ

NR1=1 (パワーONステータスフラグ真), **0** (パワーONステータスフラグ偽)

*RST コマンド

*RST Reset Command

(デバイスをレベル3でリセット)

■ 書 式

*RST

■ 使用例

WRITE @103:"*RST" アドレス3番のデバイスのみを初期化

■ 解 説

***RST (Reset)** コマンドは、デバイスをレベル3でリセットします。レベル3における初期化対象項目は、次のとおりです。

- ① デバイス固有の機能・状態をそれまでの来歴に関わらず、ある既知の状態に戻します。
- ② マクロ動作を禁止し、マクロコマンドを受け付けないモードにします。また、マクロ定義を設計者が示す状態に戻します。
- ③ デバイスを **OCIS** ステート (**Operation Complete Command Idle State**) にします。この結果、オペレーション終了ビットを標準イベント・ステータスレジスタに立てることはできません。
- ④ デバイスを **OQIS** ステート (**Operation Complete Query Idle State**) にします。この結果、オペレーション終了ビット1を出力キューに立てることができません。**MAV** ビットはクリアされます。

***RST** コマンドは、下記事項には影響を与えません。

- ① **IEEE488.1** インタフェースの状態
- ② デバイスアドレス
- ③ 出力キュー
- ④ サービスリクエスト・イネーブル レジスタ
- ⑤ 標準イベントステータス・イネーブルレジスタ
- ⑥ **Power-on-status-clear** フラグ設定

*SRE Service Request Enable Command

(サービスリクエスト・イネーブルレジスタのビットをセット)

書式

***SRE** <HEADER SEPARATOR> <DECIMAL NUMERIC PROGRAM DATA>

本書式において、

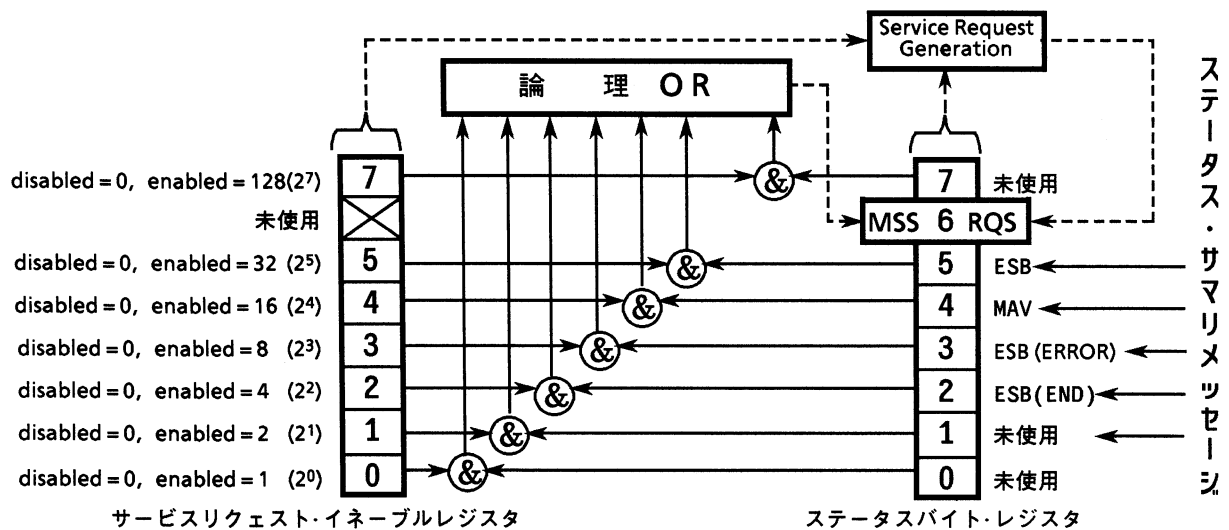
<DECIMAL NUMERIC PROGRAM DATA> = 0 ~ 255の整数に丸められ数値 (2を底としてバイナリで重み付けされていること)

使用例

WRITE @103:"*SRE 16"! イネーブルレジスタのビット4をセットします。

解説

サービスリクエスト・イネーブルレジスタのビット0, 1, 2, 3, 4, 5, 7に対応する値 $2^0=1$, $2^1=2$, $2^2=4$, $2^3=8$, $2^4=16$, $2^5=32$, $2^7=128$ の中から、**enabled**にしたいビットを選択したときのビット桁値の総和がプログラムデータとなります。**disabled**にしたいビット桁値は、0となります。



*SRE? Service Request Enable Query

(サービスリクエスト・イネーブルレジスタの現在値を返す)

書式

***SRE?**

使用例

SRE 16** 実行後、SRE?**を送ると、**16**がレスポンスされます。

解説

サービスリクエスト・イネーブルレジスタの値である **NR1**を返します。

レスポンスメッセージ

NR1=ビット6(RQSビット)はセットできないので、NR1=0~63 or 128~191

*STB? Read Status Byte Command

(MSSビットを含むステータスバイトの現在値を返す)

書 式

*STB?

使用例

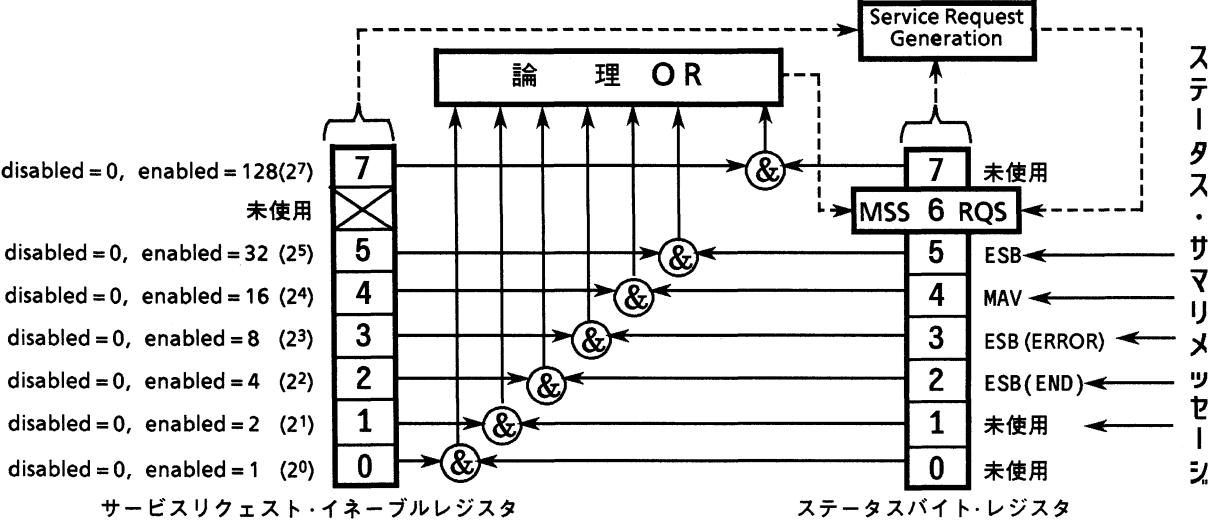
```
30 WRITE @103:"*STB?"
40 READ @103:STBV
50 PRINT STBV
```

解 説

*STB? 問合せは、バイナリで重み付けされたステータスバイト・レジスタの値と MSS サマリメッセージの値の総和を<NR1 NUMERIC RESPONSE DATA>として返します。

レスポンスメッセージ

レスポンスメッセージは、<NR1 NUMERIC RESPONSE DATA>=0～255の整数で、ステータスバイト・レジスタの各ビット桁値の総和。ステータスバイト・レジスタのビット0～5と7はそれぞれ1, 2, 4, 8, 16, 32および128に、また MSS (Master Summary Status) ビットは64に重み付けされています。MSS はサービスをリクエストする原因を少なくとも一つあることを示します。下表に、MP1762Dステータスバイト・レジスタの条件を示します。



ビット	ビットの重み	ビット名	ステータスバイトレジスタの条件	
7	128	—	0=使用せず	
6	64	MSS	0=サービスを要求していない。	1=サービスを要求している。
5	32	ESB	0=イベントステータス発生せず。	1=イベントステータス発生
4	16	MAV	0=出力キューにデータなし。	1=出力キューにデータあり。
3	8	ESB(ERROR)	0=イベントステータス発生せず。	1=イベントステータス発生
2	4	ESB(END)	0=イベントステータス発生せず。	1=イベントステータス発生
1	2	—	0=使用せず	
0	1	—	0=使用せず	

***TRG** **Trigger Command**

(IEEE488.1 の GET – Group Execute Trigger バスコマンドと同一の機能)

■ 書 式

***TRG**

■ 使用例

WRITE @103:"*TRG"

■ 解 説

***TRG** 共通コマンドは、IEEE488.1 の **GET – Group Execute Trigger** バスコマンドと同一の機能を提供します。MP1762D の場合には ***DDT** コマンドはサポートしていません。

MP1762D の場合は、***TRG** 共通コマンドを実行すると測定スタートを行います。

WRITE @103:"*TRG"

***TST** 問合せ

***TST? Self-Test Query**

(内部セルフテストのエラーの有無を返す)

■ 書 式

***TST?**

■ 使用例

```
30 WRITE @103:"*TST?"  
40 READ @103:TEST  
50 PRINT TEST
```

■ 解 説

***TST?** 問合せは、デバイス内部のセルフテストを実行します。テスト結果は出力キューにおかれます。出力キューのデータは、エラーを起こすことなくテストが完了したかどうかを示します。セルフテストの実行にあたっては、オペレータの介入を必要としません。

MP1762Dでは電源 ON 時のセルフテスト結果を出力します。

■ レスポンスメッセージ

レスポンスメッセージは、<NR1 NUMERIC RESPONSE DATA>で送ります。データの範囲=-32767 ~ 32767

NR1=0 エラー無しを示します。

NR1≠0 エラーが発生したことを示します。

***WAI** **Wait-to-Continue Command**

(デバイスがコマンド実行中は、次のコマンドを待機させる)

■ 書 式

***WAI**

■ 使用例

WRITE @103:"*WAI"

■ 解 説

***WAI** 共通コマンドは、オーバーラップコマンドをシーケンシャルコマンドとして実行します。

コントローラから送られてくるコマンドまたは問合せがデバイスで何かを実行している間でも、次に送られてくるコマンドが実行を開始することができれば、最初に実行中のコマンドまたは問合せをオーバーラップコマンドと言います。

オーバーラップコマンドの次に、***WAI** 共通コマンドを実行しますと、デバイスがコマンド実行中は、次のコマンドを待機させ、実行が終了してから、次のコマンドの実行を許します。これは、シーケンシャルコマンドと同じ動作です。

8 章

ステータス・ストラクチャー

この章では、**IEEE488.2**規格で定義されているデバイスのステータス報告とそのデータ構造およびデバイスとコントローラ間の同期テクニックについて説明します。

IEEE488.2では、**IEEE488.1**に比べて、より詳しいステータス情報を得るために、共通コマンドおよび共通問い合わせが追加されていますが、これらの詳細については、7章を参照してください。

目 次

IEEE488.2標準ステータスのモデル	8-4
ステータスバイト (STB) レジスタ	8-6
ESB および MAV サマリメッセージ	8-6
装置固有のサマリメッセージ	8-7
STB レジスタの読み出しとクリア	8-8
SRQ のイネーブル	8-10
標準イベントステータス・レジスタ	8-12
標準イベントステータス・レジスタのビット定義	8-12
問合せエラーの詳細	8-13
標準イベントステータス・レジスタの読み取り・書き込み・クリア	8-14
標準イベントステータス・イネーブルレジスタの 読み取り・書き込み・クリア	8-14

目 次 (つづき)

拡張イベントステータス・レジスタ	8-15
END イベントステータス・レジスタのビット定義	8-16
ERROR イベントステータス・レジスタのビット定義	8-17
拡張イベントステータス・レジスタの読み取り・書き込み・クリア	8-18
拡張イベントステータス・イネーブルレジスタの 読み取り・書き込み・クリア	8-18
キュー (待ち行列) モデル	8-19
デバイスとコントローラ間の同期テクニック	8-21
シーケンシャル実行の強制	8-21
デバイスの出力キュー応答待ち	8-22
サービスリクエスト待ち	8-23

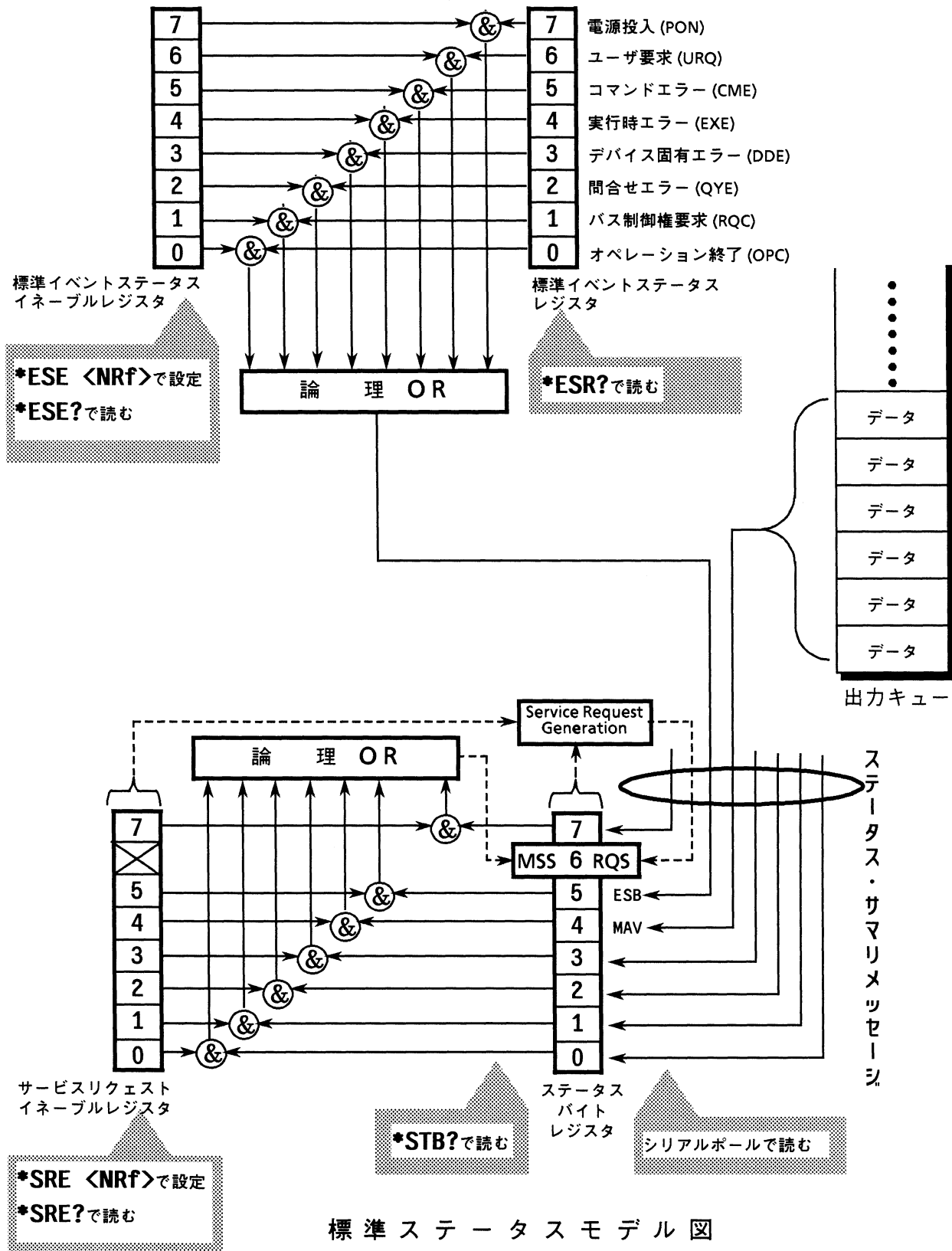
8章 ステータス・ストラクチャー

コントローラに送るステータスバイト (STB – Status Byte) は、IEEE488.1 規格に基づいていますが、その構成ビットはステータスサマリ・メッセージと呼ばれ、レジスタやキュー (待ち行列) に蓄えられたデータの現在の内容を要約して表したものです。

以下、このステータスサマリ・メッセージビットおよびこのステータスサマリ・メッセージビットを生成するためのステータスデータ構造、並びにこのステータスメッセージを使ったデバイスとコントローラ間の同期テクニックについて説明します。

IEEE488.2標準ステータスのモデル

下図に IEEE488.2で定められているステータスデータ構造の標準モデル図を示します。



ステータスモデルでは、IEEE488.1ステータスバイトが使用されます。そのステータスバイトは、ステータスデータ構造から供給される7個のサマリメッセージビットで構成されます。これらのサマリメッセージビットを生成するため、ステータスデータ構造は、レジスタモデルとキューモデルの2種類から構成されます。

レジスタモデル	キューモデル
デバイスの遭遇した事象 (event) および状態 (condition) を記録するための一組のレジスタ、これをレジスタモデル (register-model) といいます。その構造はイベントステータス・レジスタ (Event Status Register) とイベントステータス・イネーブルレジスタ (Event Status Enable Register) とから構成され、両者の AND が0でないとき、ステータスビットの対応ビットが1となります。それ以外の場合は0となります。そして、それらの論理 OR の結果が1であれば、サマリメッセージビットは、1となります。論理 OR の結果が0であれば、サマリメッセージビットは、0となります。	順序を待つ状態値または情報をシーケンシャルに記録するための待ち行列で、これをキューモデル (queue-model) といいます。キュー構造では、キューにデータがある時だけ対応ビットが1となり、キューが空であれば0となります。

以上、説明したレジスタモデルとキューモデルをもとに、IEEE488.2のステータスデータ構造の標準モデルは、2種類のレジスタモデルと1個のキューモデルから構成されています。:

- ①標準イベントステータス・レジスタと標準イベントステータス・イネーブルレジスタ
- ②ステータスバイト・レジスタとサービスリクエスト・イネーブルレジスタ
- ③出力キュー

標準イベントステータス・レジスタ (Standard Event Status Register)	ステータスバイト・レジスタ (Status Byte Register)	出力キュー (Output Queue)
これは前記のレジスタモデルの構造を持ち、この内容はデバイスが遭遇する事象の中で、8種類の事象 (①電源投入、②ユーザ要求、③コマンドエラー、④実行時エラー、⑤デバイス固有エラー、⑥問い合わせエラー、⑦バス制御権要求、⑧オペレーション終了) の各ビットを標準事象として、標準イベントステータス・レジスタに立てます。論理 OR 出力ビットは、Event Status Bit (ESB) サマリメッセージとして、ステータスバイト・レジスタの bit 5 (DI06) に要約表示されます。	ステータスバイト・レジスタは、RQS ビットおよびステータスデータ構造からの7個のサマリメッセージビットがセット可能なレジスタで、サービスリクエスト・イネーブルレジスタと組で使用され、両者の OR が0でないとき SRQ を ON にします。この時のステータスバイト・レジスタの bit 6 (DIO7) は、RQS ビットとしてシステム予約されており、このビットにより外部コントローラにサービス要求の有ることを報告します。この SRQ の仕組みは IEEE488.1 の規格に従っています。	これは前記キューモデルの構造を持ち、この内容は出力バッファにデータの有ることを知らせる Message Available (MAV) サマリメッセージとしてステータスバイト・レジスタの bit 4 (DI05) に要約表示されます。

ステータスバイト (STB) レジスタ

STB レジスタは、デバイスの STB と RQS (または MSS) メッセージから構成されます。IEEE488.1では、STB と RQS メッセージの伝達 (reporting) 方法については定義していますが、セットおよびクリアのプロトコルと STB の意味については定義していません。IEEE488.2では、デバイスのステータスサマリメッセージおよび*STB? 共通問い合わせに応じて、STB と共に bit 6に送出される Master Summary Status (MSS) について定義しています。

ESB および MAV サマリメッセージ

ESB サマリメッセージおよび MAV サマリメッセージについて説明します。

(1) ESB サマリメッセージ

ESB (Event Summary Bit) サマリビットは、IEEE488.2で定義されたメッセージで、STB レジスタの bit 5に現れます。このbitの状態は、標準イベントステータスレジスタを最後にリード後またはクリア後において、イベント発生が有効となるようにサービスリクエスト・イネーブルレジスタを設定した状態で、IEEE488.2で定義された事象が少なくとも1つ以上発生したかどうかを示すものです。ESB サマリメッセージビットは、イベント発生が有効となるように設定された状態で、標準イベントステータスレジスタに登録されたイベントが一つでも TRUE にセットされれば、TRUE となります。逆に ESB サマリビットは、イベント発生が有効となるように設定された状態でも、登録されたイベントの発生が一つもないときに FALSE となります。

(2) MAV サマリメッセージ

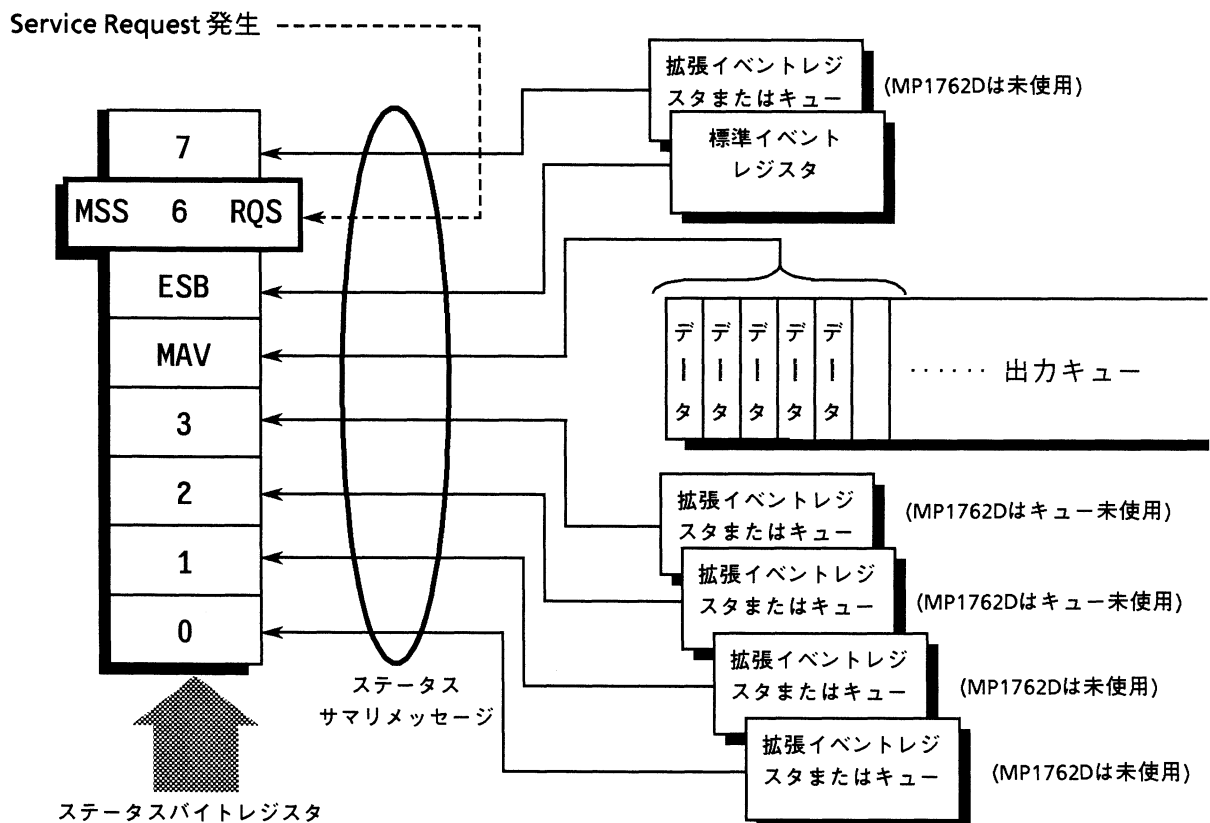
MAV (Message Available) サマリビットは、IEEE488.2で定義されたメッセージで、STB レジスタの bit 4に現れます。このbitの状態は、出力キューが“空”であるかどうかを示します。デバイスがコントローラからレスポンスメッセージの送出要求を受け付ける用意ができているときに、MAV サマリメッセージビットは1 (TRUE) となり、出力キューが“空”のときに0 (FALSE) となります。このメッセージはコントローラとの情報交換に同期を取るために利用されます。たとえば、コントローラがデバイスに問合せコマンドを送り、MAV が TRUE になるのを待つというように使うことができます。そして、デバイスが応答をするのを待つ間、他の処理をすることができます。もし、初めに MAV をチェックすることなしに出力キューを読み取り始めた場合は、すべてのシステムバス動作はデバイスが応答するまで待たされます。

装置固有のサマリメッセージ

IEEE488.2では、ステータスバイト・レジスタのbit 7 (DIO8), bit 3 (DIO4) ~ bit 0 (DIO1) をステータスレジスタのサマリビットとして使うか、キューにデータの有ることを知らせるビットとして使うかは、決められていません。これらのビットは、装置固有のサマリメッセージとして利用することができます。

装置固有サマリメッセージは、それぞれレジスタモデルまたはキューモデルのステータスデータ構造を持ちます。すなわち、このステータスデータ構造は事象および状態を並列的に報告する一組のレジスタであるか、または状態および情報を順次報告する一個のキューです。サマリビットは対応するステータスデータ構造の現在の状態を要約表示します。レジスタモデルの場合は、一つ以上の **TRUE** の発生が有効となるように設定された事象が存在するとき、またキューモデルの場合は、キューが空でないときサマリメッセージは **TRUE** となります。

MP1762Dでは下記に示すように、bit 0, bit 1およびbit7を未使用とし、bit 2およびbit 3の3ビットをステータスレジスタのサマリビットとして使っていますので、レジスタモデルは全部で5種類(拡張3種類)、キューモデルは拡張なしで、出力キューの一種類となっています。



STB レジスタの読み出しとクリア

STB レジスタの内容は、シリアルポール、または ***STB?** 共通問い合わせを使って読み取ります。どちらの方法でも IEEE488.1 の STB メッセージを読みとりますが、bit 6 (位置) に送られる値はその方法によって異なります。

STB レジスタの内容は、***CLS** コマンドによってクリアすることができます。

(1) シリアルポールを使って読む

IEEE488.1 によるシリアルポールが行われた場合、デバイスは7ビットのステータスバイトと、IEEE488.1 による RQS メッセージビットを返送しなければなりません。IEEE488.1 によれば、RQS メッセージはデバイスが SRQ を TRUE で送出していたかどうかを示します。ステータスバイトの値は、シリアルポールを行っても変化しません。デバイスは、ポーリングされた直後 rsv メッセージを FALSE にセットしなければなりません。これにより、新たなサービス要求のための原因が発生する前に、再度デバイスがポーリングされた場合、RQS メッセージは FALSE となっています。

(2) *STB 共通問い合わせを使って読む

STB?** 共通問い合わせは、デバイスに STB レジスタの内容と MSS (Master Summary Status) サマリメッセージからの一つの <NR1 NUMERIC RESPONSE DATA> を送出させます。応答はバイナリで重み付けされた STB レジスタの値と MSS サマリメッセージの値の総和を表します。STB レジスタの bit 0 ~ 5 と 7 はそれぞれ 1, 2, 4, 8, 16, 32 および 128 に、また MSS は 64 に重み付けされます。これにより、RQS メッセージの替わりに MSS サマリメッセージが bit 6 位置に現れることを除いては、STB?** に対する応答は、シリアルポールに対する対応と一致します。

(3) MSS (Master Summmary Status) の定義

デバイスに少なくとも一つのサービスを要求する原因があることを示します。MSS メッセージは ***STB?** 問い合わせに対するデバイスの応答の中でビット 6 に現れますが、シリアルポールに対する応答としては現れません。また、IEEE488.1 のステータスバイトの一部とみなしてはなりません。MSS は STB レジスタと SRQ イネーブル (SRE) レジスタのビットの組合せによる総合的 OR により構成されます。これを具体的に示すと、結局 MSS は以下のように定義されます。

(STB Register bit0 AND SRE Register bit0)

OR

(STB Register bit1 AND SRE Register bit1)

OR

:

:

(STB Register bit5 AND SRE Register bit5)

OR

(STB Register bit7 AND SRE Register bit7)

MSS の定義において、STB レジスタと、SRQ イネーブルレジスタ双方の bit 6 の状態を無視していますので、MSS の値を算出するに当たっては、ステータスバイトを bit 6 が常に 0 である 8 ビットの値として取り扱ってかまいません。

(4) *CLS 共通コマンドによる STB レジスタのクリア

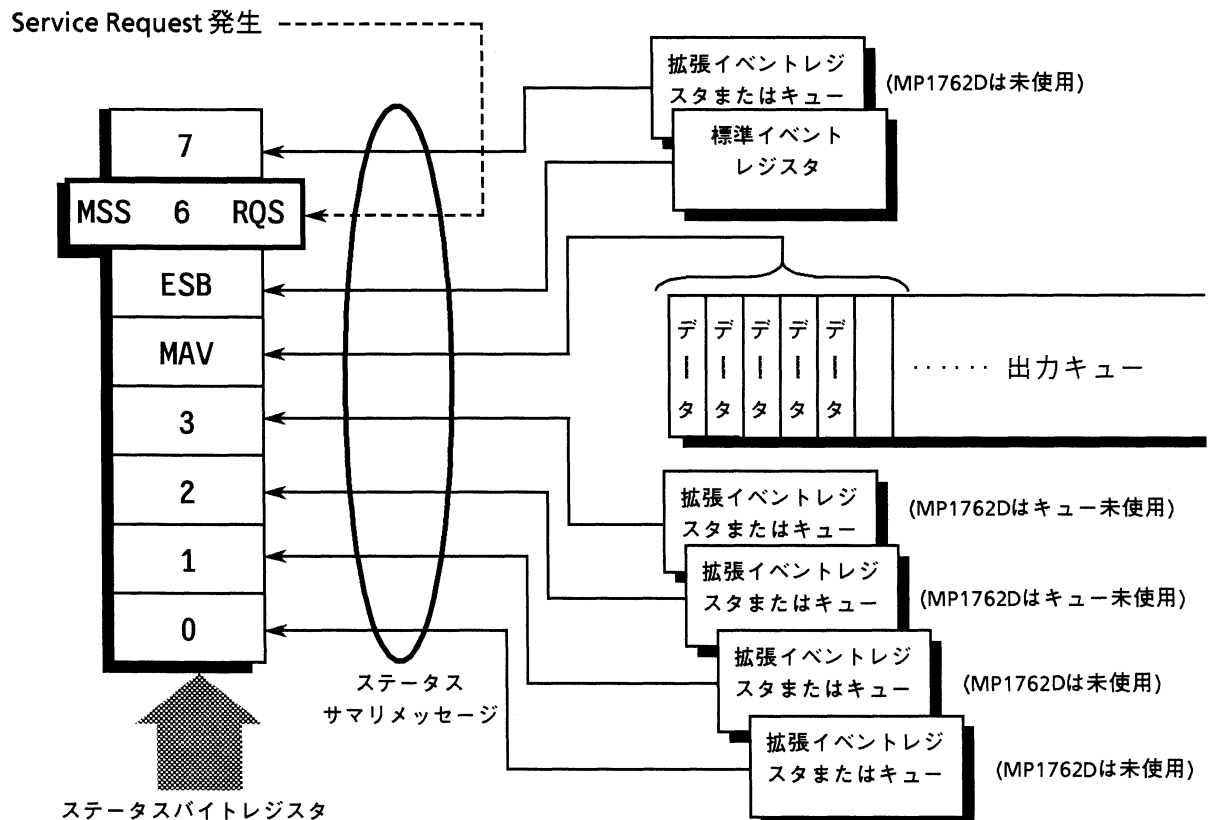
*CLS 共通コマンドは、出力キューとその MAV サマリメッセージを除くすべてのステータスデータ構造(すなわち、これらのイベントレジスタおよびキュー)をクリアし、これに応じてそれらに対応するサマリメッセージもクリアします。

出力キューとその MAV サマリメッセージも、下例の場合はクリアされます。

```
30 WRITE @103:"DTMΔ0;CTMΔ0"
```

```
40 WRITE @103:"*CLS;DTM?"
```

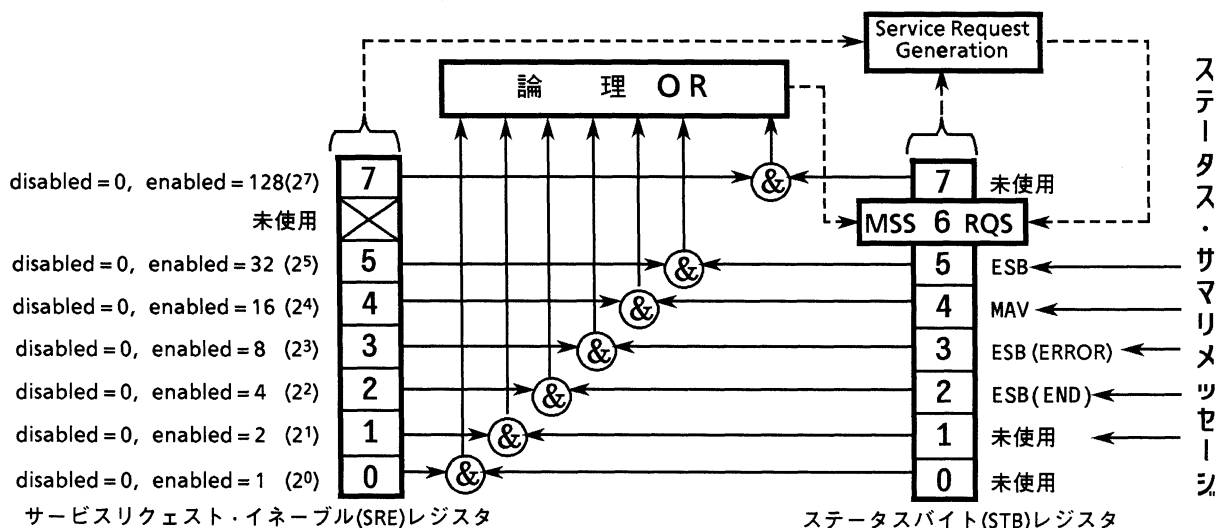
すなわち、<PROGRAM MESSAGE TERMINATOR> の後、そして何らかの <Query MESSAGE UNIT> 要素の前に *CLS コマンドを送出すると、すべてのステータスバイトはクリアされます。この方法により出力キューのすべての未読み出しのメッセージはクリアされ、MAV メッセージは FALSE となります。また、*STB? に対する応答時、MSS メッセージも FALSE になります。なお、各イネーブル・レジスタの設定値については、*CLS によって影響されません。



SRQ のイネーブル

SRQ のイネーブルにより、STB レジスタの中のどのサマリメッセージをサービスリクエストに対して有効にするか無効にするかを選択できます。下図で示すサービスリクエストイネーブル (SRE) レジスタがサマリメッセージを選択する手段として使用されます。

サービスリクエスト・イネーブルレジスタ上のビットは、ステータスバイト・レジスタ上のビットと対応しています。サービスリクエスト・イネーブルレジスタ上の有効なビットに対応するステータスバイト中のビットに1が立つと、デバイスは、RQS ビットを1として、サービスリクエストをコントローラに対して行います。たとえば、サービスリクエスト・イネーブルレジスタのビット4をイネーブルにセットしておく、出力キューにデータがあれば、MAV ビットに1が立つ度に、サービスリクエストをコントローラに対して行います。



(1) SRE レジスタの読み出し

SRE レジスタの内容は、***SRE?** 共通問合せを使って読み出せます。この問合せに対するレスポンスメッセージは、**<NR1 NUMERIC RESPONSE DATA> = 0 ~ 255** の整数で、サービスリクエスト・イネーブルレジスタの各ビット桁値の総和となります。サービスリクエスト・イネーブルレジスタのビット0~5と7はそれぞれ1, 2, 4, 8, 16, 32および128に重み付けされています。使用されないビット6は、常に0でなければなりません。

(2) SRE レジスタの更新

SRE レジスタは、***SRE** 共通命令を使って書き込まれます。***SRE** 共通命令の後には**<DECIMAL NUMERIC PROGRAM DATA>**要素が続きます。**<DECIMAL NUMERIC PROGRAM DATA>**は整数に丸められ、2を基数としてバイナリで表現され、SRE レジスタの各ビット桁値(ウェイト値)の総和を表します。このbit値は、1が**enabled**の状態を表し、0が**disabled**の状態を表します。ビット6の値は常に無視しなければなりません。

(3) SRE レジスタのクリア

***SRE** 共通コマンドの実行または電源再投入によって、**SRE** レジスタのクリアをクリアできます。

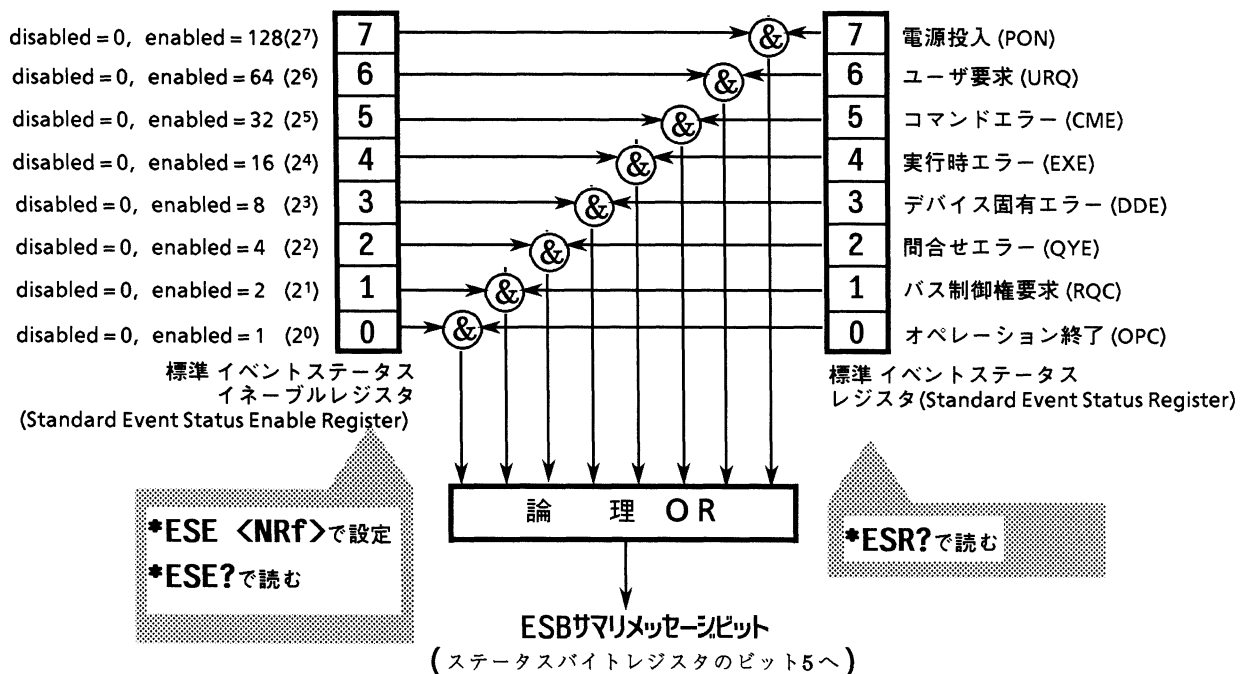
***SRE** 共通コマンドの場合は、<**DECIMAL NUMERIC PROGRAM DATA**>の要素の値を0にすれば、**SRE** レジスタをクリアすることができます。レジスタをクリアすることで、ステータス情報が **rsv** ローカルメッセージを発生することを禁止できますので、この結果サービス要求は発生しなくなります。

MP1762D は***PSC** コマンドを用意していますので、もし電源投入時 **psc** フラグが **True** の時 **SRE** レジスタはクリアされます。

標準イベントステータス・レジスタ

標準イベントステータス・レジスタのビット定義

標準イベントステータス・レジスタは、IEEE488.2対応機種であれば、すべてのデバイスが装備しなければならないイベントステータス・レジスタです。下図に、標準イベント・ステータス・レジスタモデルの動作を示します。レジスタモデルの動作それ自身は、これまでに説明してきたのと同じなので、ここでは、標準イベント・ステータス・レジスタの各ビットの意味について、IEEE488.2の定義を説明します。



ビット	イベント名	説明
7	電源投入 (PON—Power on)	電源投入が OFF から ON へと変化した。
6	ユーザ要求 (URQ—User Request)	ローカル制御 (rtl) を要求しています。 このビットは、デバイスのリモート/ローカル状態とは無関係に発生します。MP1762D では使用しておりませんので常に0となります。
5	コマンドエラー (CME—Command Error)	文法に従わないプログラムメッセージ、ミススペルのコマンド、またはプログラムメッセージの中で GET コマンドを受信した。(ヘッダパラメータの文法のミス、パラメータの個数ミス)
4	実行時エラー (EXE—Execution Error)	文法に問題はないが、実行できないプログラムメッセージを受信した。(パラメータの範囲外)
3	デバイス固有エラー (DDE—Device-dependent Error)	CME, EXE, QYE 以外の原因によるエラーが発生した。 (現在のデバイスの状態により、実行不可)
2	問合せエラー (QYE—Query Error)	出力キューにデータがないのに、出力キューからデータを読もうとした。または出力キューのデータがなんらかの原因、たとえばオーバーフローなどで失われた。

ビット	イベント名	説明
1	バス制御権要求 (RQC-Request Control)	自らがアクティブコントローラになることを要求している。MP1762Dでは使用しておりませんので常に0となります。
0	オペレーション終了 (OPC-Operation Complete)	デバイスが、ペンディング中の、指定した動作を終了して、新しい命令を受ける準備ができている。このビットは、*OPCコマンドに対してのみ応答し、オペレーション終了ビットを立てます。

問合せエラーの詳細

No.	項目	説明
1	不完全なプログラムメッセージ	デバイスがプログラムメッセージを受信中に、プログラムメッセージ・ターミネータを受信する前にコントローラからMTAを受信した場合、デバイスはそれまでに入力した不完全なプログラムメッセージを破棄し、次のプログラムメッセージを待ちます。不完全なプログラムメッセージの破棄の動作では、デバイスは入出力バッファをクリアし、問合せエラーをステータス報告部に伝え、標準ステータス・レジスタのビット2に問合せエラービットをセットします。
2	レスポンスメッセージの出力の中断	デバイスがレスポンスメッセージを送信中で、レスポンスメッセージ・ターミネータを転送し終わる前にコントローラからMLAを受信した場合には、デバイスは自動的にレスポンスメッセージ出力中断動作を行い、次のプログラムメッセージを待ちます。レスポンスメッセージ出力中断動作では、デバイスは出力バッファをクリアし、問合せエラーをステータス報告部に伝え、標準ステータス・レジスタのビット2に問合せエラービットをセットします。
3	レスポンスメッセージを読まないで次のプログラムメッセージを送信した場合	コントローラが問合せメッセージを含むプログラムメッセージの送信に続いて、さらに次のプログラムメッセージを送信したためにデバイスがレスポンスメッセージの出力をできなかった場合、デバイスはレスポンスメッセージの破棄を行い、次のプログラムメッセージを待ちます。8と同じように問合せエラーをステータス報告部に伝えます。
4	出力キューのオーバーフロー	問合せメッセージを多数含むプログラムメッセージを実行していくとき、出力キュー(256バイト)に入りきらないほど多くのレスポンスメッセージが発生することがあります。出力キューが満杯になっても、まだ問合せメッセージが入力され、それに伴いレスポンスメッセージを出力しなければいけない時、出力キューがオーバーフロー状態になります。出力キューがオーバーフローすると、デバイスは出力キューをクリアし、レスポンスメッセージ作成部をリセットします。また、問合せエラービットをステータス報告部の標準イベントステータスレジスタのビット2にセットします。

標準イベントステータス・レジスタの読み取り・書き込み・クリア

読み取り	<p>*ESR? 共通問い合わせにより破壊的に読み取られます。</p> <p>すなわち、読み取られた後、クリアされます。レスポンスメッセージは、イベントビットに2進数の重みを付けて10進数変換した<NR1>です。</p>
書き込み	<p>クリアすることを除き、外部から書き込みは行えません。</p>
クリア	<p>次の場合にのみクリアされます。</p> <ul style="list-style-type: none"> ① *CLS コマンド受信 ② 電源 ON ステータス・クリア・フラグが真ならば、電源 ON のとき。 ③ *ESR? 問合せコマンドに対して、イベントが読み込まれた。

標準イベントステータス・イネーブルレジスタの読み取り・書き込み・クリア

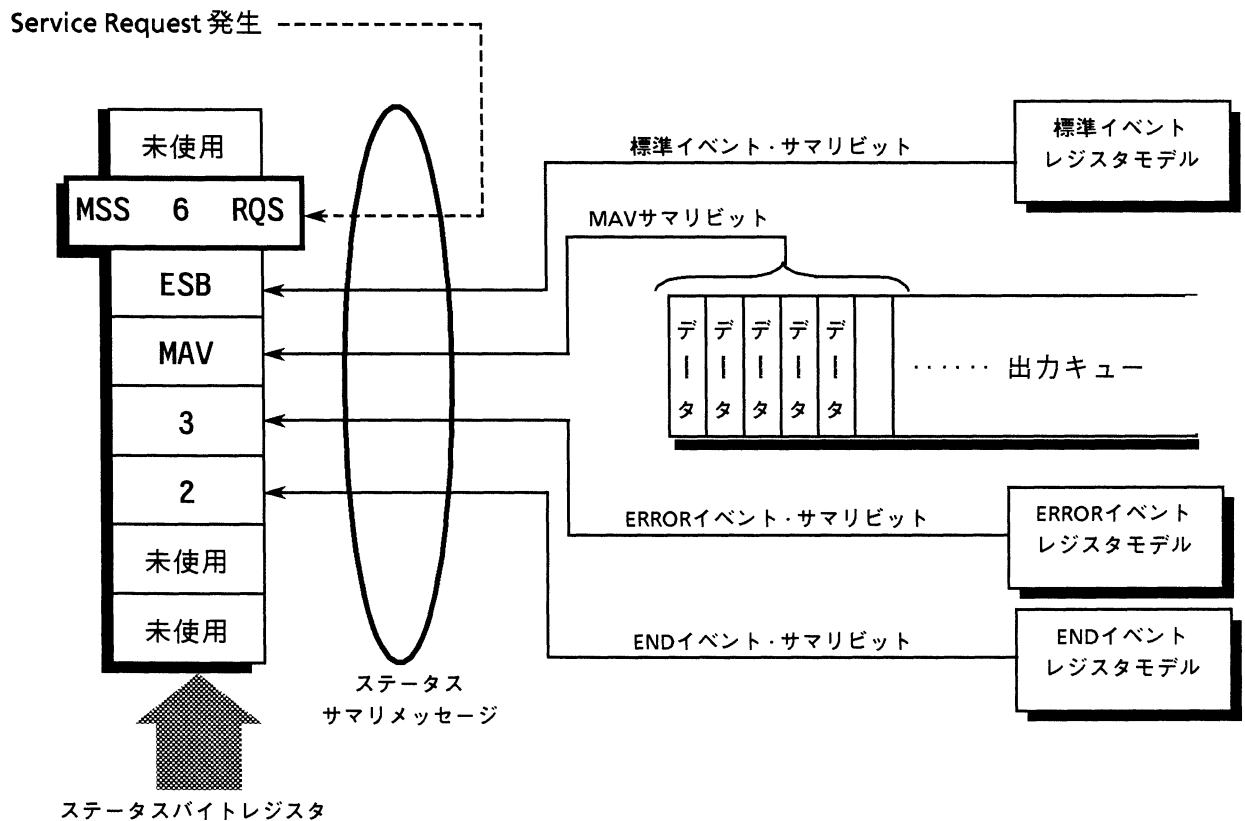
読み取り	<p>*ESE? 共通問い合わせにより非破壊的に読み取られます。</p> <p>すなわち、読み取られた後も、クリアされません。レスポンスメッセージは、2進数の重みを付けて2進-10進変換された<NR1>で返されます。</p>
書き込み	<p>*ESE 共通コマンドによって書き込まれます。レジスタのビット0～7は、それぞれ1, 2, 4, 8, 16, 32, 64, 128に重み付けされていますので、書き込みデータは、その中から希望のビット桁値を合計した<10進数値プログラムデータ>で送ります。</p>
クリア	<p>次の場合にクリアされます。</p> <ul style="list-style-type: none"> ① データ値0の*ESE コマンドを受信 ② 電源 ON ステータス・クリア・フラグが真の状態での電源 ON 時。 <p>標準イベントステータス・イネーブルレジスタは、下記事項に影響されません。</p> <ul style="list-style-type: none"> ① IEEE488.1のデバイスクリア・ファンクションの状態変化 ② *RST 共通コマンドの受信 ③ *CLS 共通コマンドの受信

拡張イベントステータス・レジスタ

IEEE488.2対応機種において、これまで説明したイネーブルレジスタを含むステータスバイト・レジスタおよび標準イベントステータス・レジスタの各レジスタモデルは、必須のものとなっています。

IEEE488.2では、ステータスバイト・レジスタの bit 7 (DIO8), bit 3 (DIO4) ~ bit 0 (DIO1) を拡張レジスタモデルまたは拡張キューモデルから供給されるステータスサマリビット用に割り当てています。

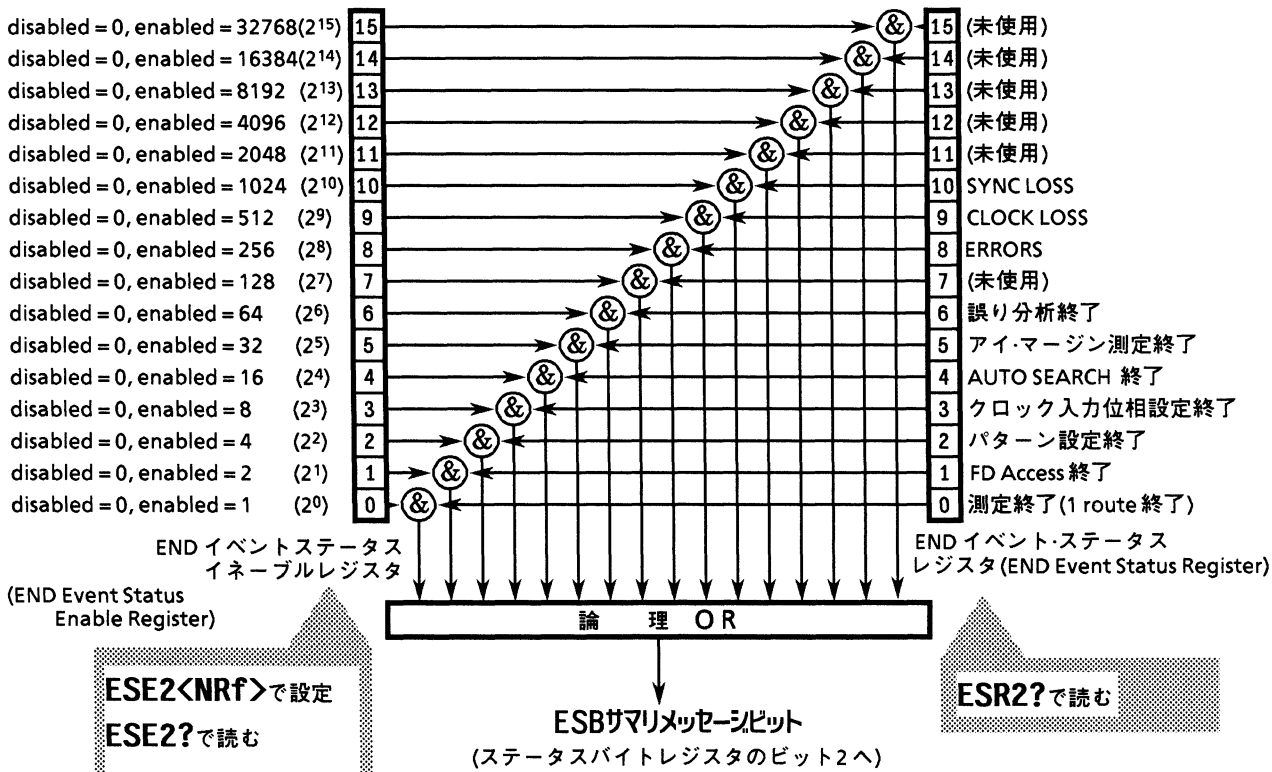
MP1762D では、下記に示すように、bit 0, bit 1, bit 7 を未使用とし、bit 2, bit 3 の2ビットを、拡張レジスタモデルから供給されるステータスサマリビット用として、ERROR および END サマリビットに割り当てています。したがって、キューモデルは拡張されていないので、出力キューの一種類となっています。



以下、END, ERROR 各々の拡張イベントレジスタ・モデルのビットの定義、読み取り・書き込み・クリアについて説明します。

END イベントステータス・レジスタのビット定義

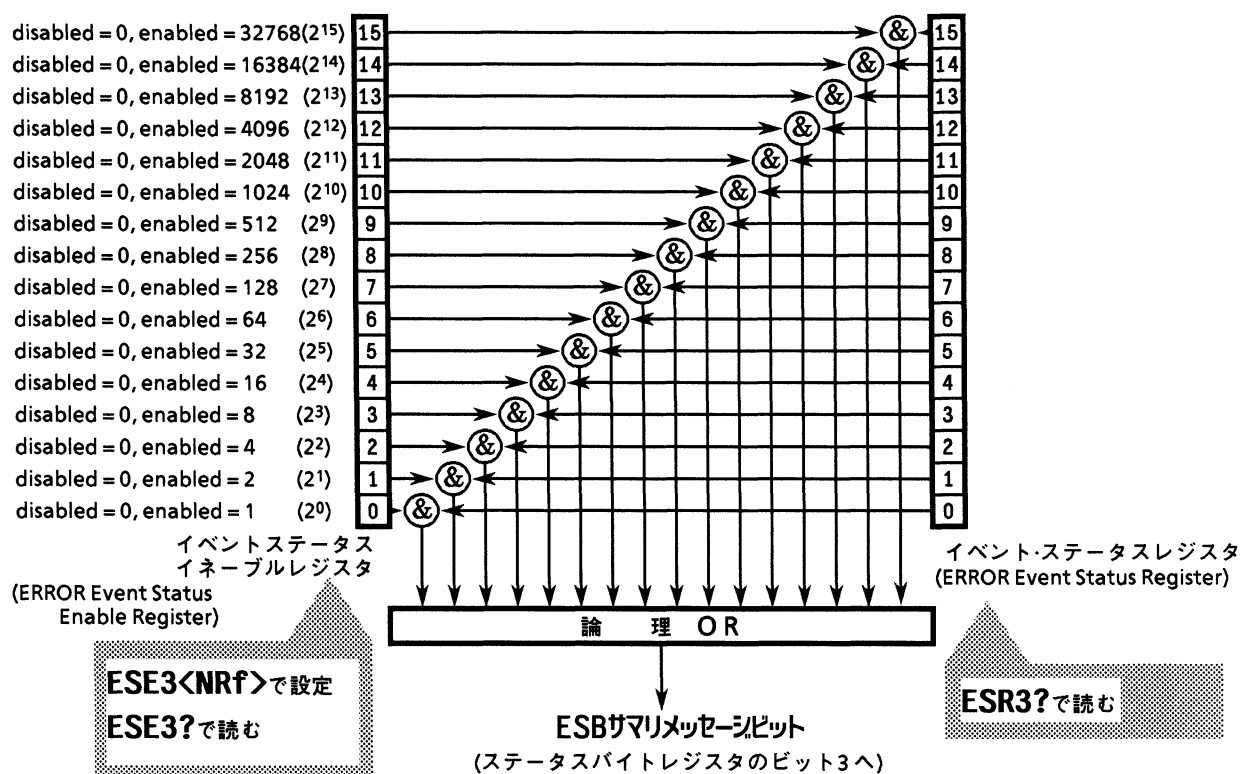
下記に、END イベントステータス・レジスタモデルの動作、イベントビット名およびその意味について説明します。



ビット	イベント名	説 明
15	(未使用)	(未使用)
14	(未使用)	(未使用)
13	(未使用)	(未使用)
12	(未使用)	(未使用)
11	(未使用)	(未使用)
10	SYNC LOSS	同期はずれ発生、または同期はずれ回復した。
9	CLOCK LOSS	クロック断発生、またはクロック断回復した。
8	ERRORS	誤り無しから誤りを検出した。
7	(未使用)	(未使用)
6	誤り分析終了	誤り分析が終了した。(オプション-01実装時のみ)
5	アイ・マージン測定終了	アイ・マージン測定が終了した。
4	AUTO SEARCH 終了	オート・サーチが終了した。
3	クロック入力位相設定終了	クロック入力位相のサーボ回路が BUSY から READY 状態に変化した。
2	パターン 設定終了	プログラマブル・パターンの設定が終了した。
1	FD Access 終了	フロッピーのアクセスが終了した。
0	測定終了	マニュアル測定で測定を STOP した。シングル測定で測定が終了した。また、リピート測定で測定終了の周期ごと。

ERROR イベントステータス・レジスタのビット定義

下記に、**ERROR** イベントステータス・レジスタモデルの動作、イベントビット名およびその意味について説明します。



ビット	イベント名	説 明
15	(未使用)	(未使用)
14	(未使用)	(未使用)
13	(未使用)	(未使用)
12	(未使用)	(未使用)
11	(未使用)	(未使用)
10	(未使用)	(未使用)
9	(未使用)	(未使用)
8	(未使用)	(未使用)
7	(未使用)	(未使用)
6	(未使用)	(未使用)
5	(未使用)	(未使用)
4	(未使用)	(未使用)
3	(未使用)	(未使用)
2	(未使用)	(未使用)
1	異常発生	FD異常発生
0	異常発生	プリンタ異常発生

拡張イベントステータス・レジスタの読み取り・書き込み・クリア

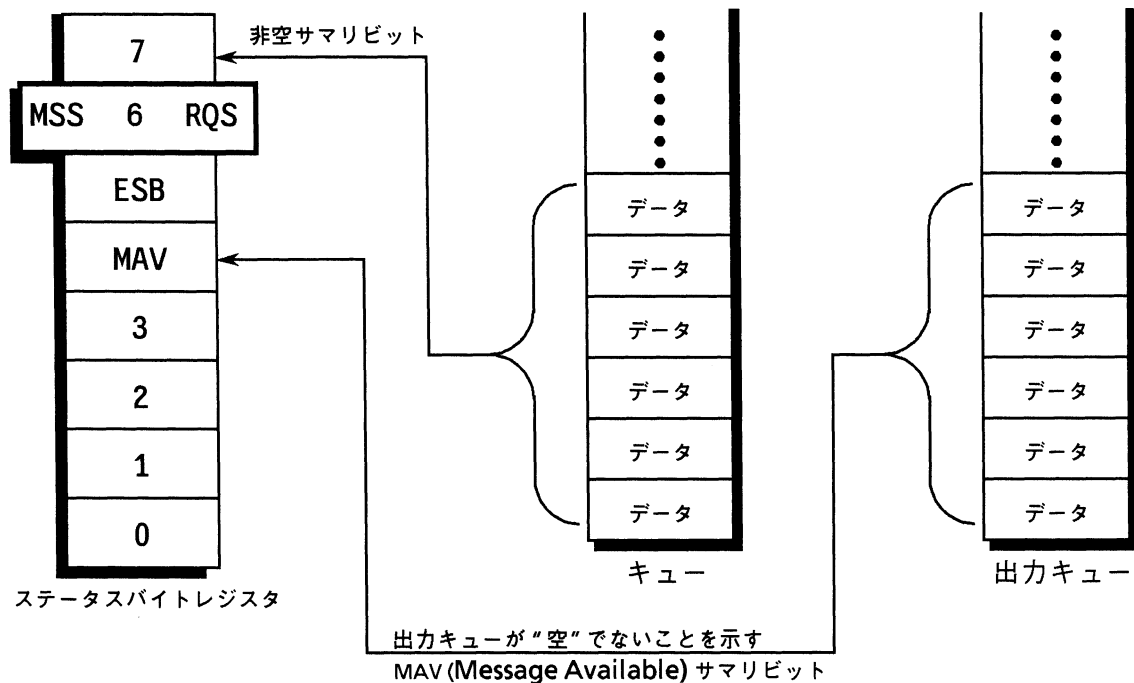
読み取り	問合わせにより破壊的に読み取られます。すなわち、読み取られた後、クリアされます。 END, ERROR の各々のイベントステータス・レジスタに対しては、 ESR2?, ESR3? の問合わせで読み取ります。その値はイベントビットに2進数の重みを付けて10進数変換した<NR1>です。
書き込み	クリアすることを除き、外部から書き込みは行えません。
クリア	次の場合にクリアされます。 ① *CLS コマンド受信 ② 電源 ON ステータス・クリア・フラグが真ならば、電源 ON のとき。 ③ 問合せコマンドに対して、イベントが読み込まれた。

拡張イベントステータス・イネーブルレジスタの読み取り・書き込み・クリア

読み取り	問合わせにより非破壊的に読み取られます。すなわち、読み取られた後も、クリアされません。 END, ERROR の各々のイベントステータス・イネーブル・レジスタに対しては、 ESE2?, ESE3? の問合わせで読み取ります。その値は、2進数の重みを付けて2進-10進変換された<NR1>で返されます。
書き込み	END, ERROR の各々のイベントステータス・イネーブル・レジスタに対しては、 ESE2, ESE3 プログラムコマンドによって書き込まれます。 レジスタのビット0～7は、それぞれ1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768に重み付けされていますので、書き込みデータは、その中から希望のビット桁値を総和した<10進数値プログラムデータ>で送ります。
クリア	次の場合にクリアされます。 ① END, ERROR の各々のイベントステータス・イネーブル・レジスタに対してはデータ値0の ESE2, ESE3 , プログラムコマンドを受信 ② 電源 ON ステータス・クリア・フラグが真の状態での電源 ON 時。 拡張イベントステータス・イネーブルレジスタは、下記事項に影響されません。 ① *IEEE488.1 のデバイスクリア・ファンクションの状態変化 ② *RST 共通コマンドの受信 ③ *CLS 共通コマンドの受信

キュー (待ち行列) モデル

下図右側に、ステータスデータ構造のキュー・モデルを示します。キューとは、順番に並べられた情報リストを含むデータ構造で、シーケンシャル・ステータスその他の情報を報告する手段を提供します。そうした情報がキューの中に存在することは、サマリ・メッセージに要約表示されます。キューの内容は、デバイスがトーカーアクティブ・ステート (TACS) にある時、ハンドシェークにより読み取られます。



サマリ・メッセージの中で、**MAV** サマリ・メッセージをステータス・バイトのビット4へ出力するキューは、『出力キュー』と呼ばれ、必須となっています。**MAV** サマリ・メッセージをステータスバイト・レジスタのビット0～3, 7のいずれかへ出力可能なキューは、単に『キュー』と呼ばれ、オプションとなっています。ステータスバイト・レジスタのビット0～3, 7には、レジスタモデルからのサマリ・メッセージも接続可能ですので、サマリ・メッセージの種類は、デバイスごとに相違します。

当社では、ステータスバイト・レジスタのビット7を『キュー』からのサマリ・メッセージビット用に当てていますが、『出力キュー』で間に合う場合は、特に『キュー』を使用していませんので、ステータスバイト・レジスタのビット7は未使用となっています。

ここでは、『出力キュー』と一般のキューを、次ページで比較して示します。

出力キューとキューの比較表

項 目	出力キュー	キュー
データ 入出力方式	FIFO 形	必ずしも FIFO 形 である必要はありません。
読出し	IEEE 488.2で定義されたプロトコルを通じてのみ読み出されます。読み出されるレスポンスメッセージユニットのタイプは、問合せによって決定されます。	装置固有の問合せコマンドによって読み出されます。読み出されるレスポンスメッセージユニットは同じタイプでなければなりません。
書込み	プログラムメッセージ要素が直接書き込まれることはありません。 IEEE 488.2メッセージ交換プロトコルを通じてのみシステムインタフェースとやりとりされます。	プログラムメッセージ要素が直接書き込まれることはありません。 コード化されたデバイスの情報を示します。
サマリ メッセージ	出力キューが空でない時には TRUE (1)、空の時には FALSE (0) となります。MAV サマリ・メッセージは、デバイスとコントローラとの情報交換の同期に用いられます。	キューが空でない時には TRUE (1)、空の時には FALSE (0) となります。
クリア	次のいずれかが発生した時にはクリアされます。 ① キューの中のすべてのアイテムが読み取られた。 ② メッセージ交換初期化のため、DCL バスコマンドを受信した。 ③ 電源投入で、pon が True となった。	次のいずれかが発生した時にはクリアされます。 ① キューの中のすべてのアイテムが読み取られた。 ② *CLS コマンドを受信した。 ③ 装置固有のその他の手段。

デバイスとコントローラ間の同期テクニック

デバイスとコントローラ間の同期をとるには3つの方法があります。

- ① シーケンシャル実行の制御 (***WAI?** コマンドによる)
- ② デバイスの出力キュー応答待ち (***OPC?** 問合せによる)
- ③ サービスリクエスト待ち (***OPC** コマンド/***OPC?** 問合せによる)

シーケンシャル実行の強制

デバイス固有のコマンドは、シーケンシャルコマンドとオーバラップコマンドの2つのタイプに分けられます。

- シーケンシャルコマンド コントローラから送られてくるコマンドまたは問合せで、デバイスが何かを実行している間、次に送られてくるコマンドの実行を開始しないコマンドまたは問合せをシーケンシャルコマンドと言います。
- オーバラップコマンド コントローラから送られてくるコマンドまたは問合せで、デバイスが何かを実行している間でも、次に送られてくるコマンドの実行を開始するコマンドまたは問合せをオーバラップコマンドと言います。

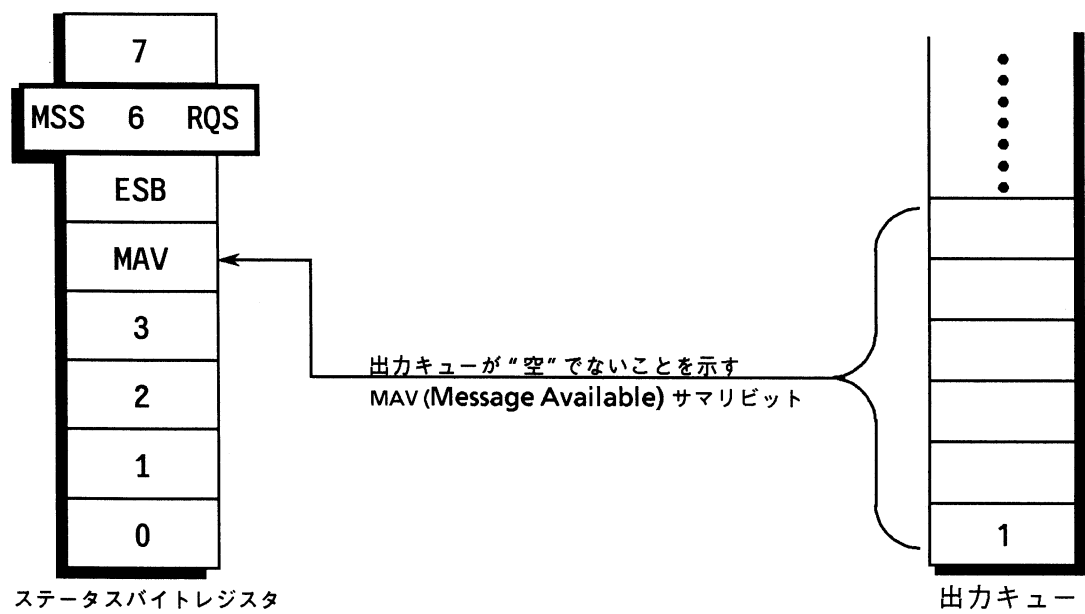
シーケンシャル実行の強制とは、本来オーバラップコマンドとして振る舞うコマンドを強制的にシーケンシャルに動作させ、一つの処理終了を合図に、次の処理を行う同期テクニックを言います。この同期テクニックには、***WAI** コマンドを使用します。

デバイスの出力キュー応答待ち

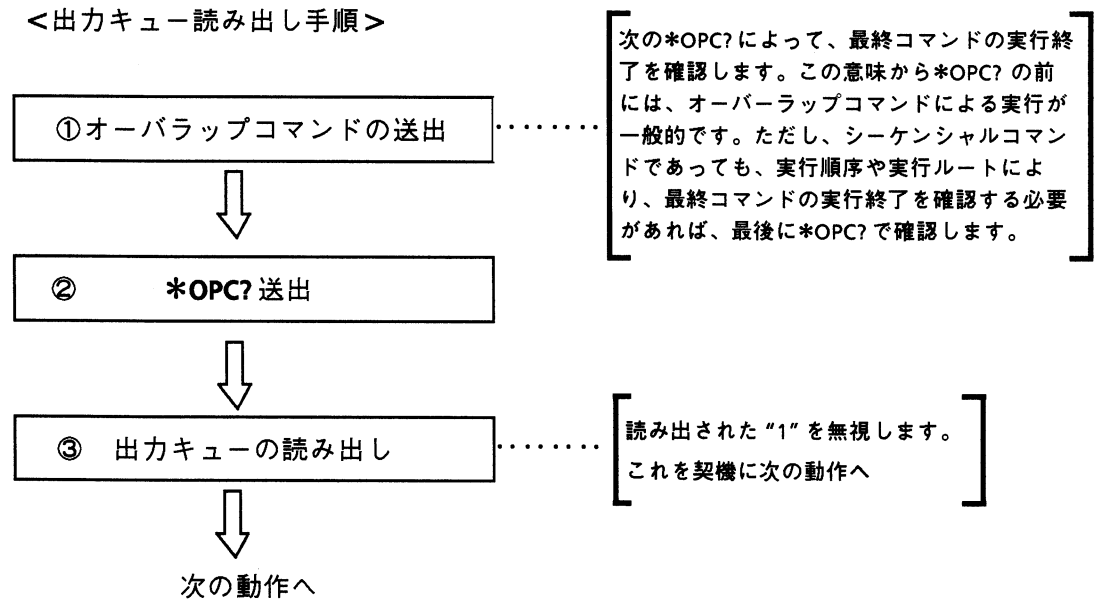
***OPC?** 問合せを実行すると、ペンディング中の、指定した動作をデバイスがすべて終了したら、出力キューに "1" を立て、**MAV** サマリメッセージを発生させます。

出力キュー応答待ちによる同期とは、上記の結果によって得られた出力キューの中のデータ1を読み出すか、または **MAV** サマリメッセージビットを読み出すことによって、デバイスとコントローラ間の同期をとる方法です。

MAV サマリメッセージビットの利用は、『サービスリクエスト待ちによる方法』に属するため、そこで説明します。ここでは、『出力キューの読み出しによる方法』について説明します。



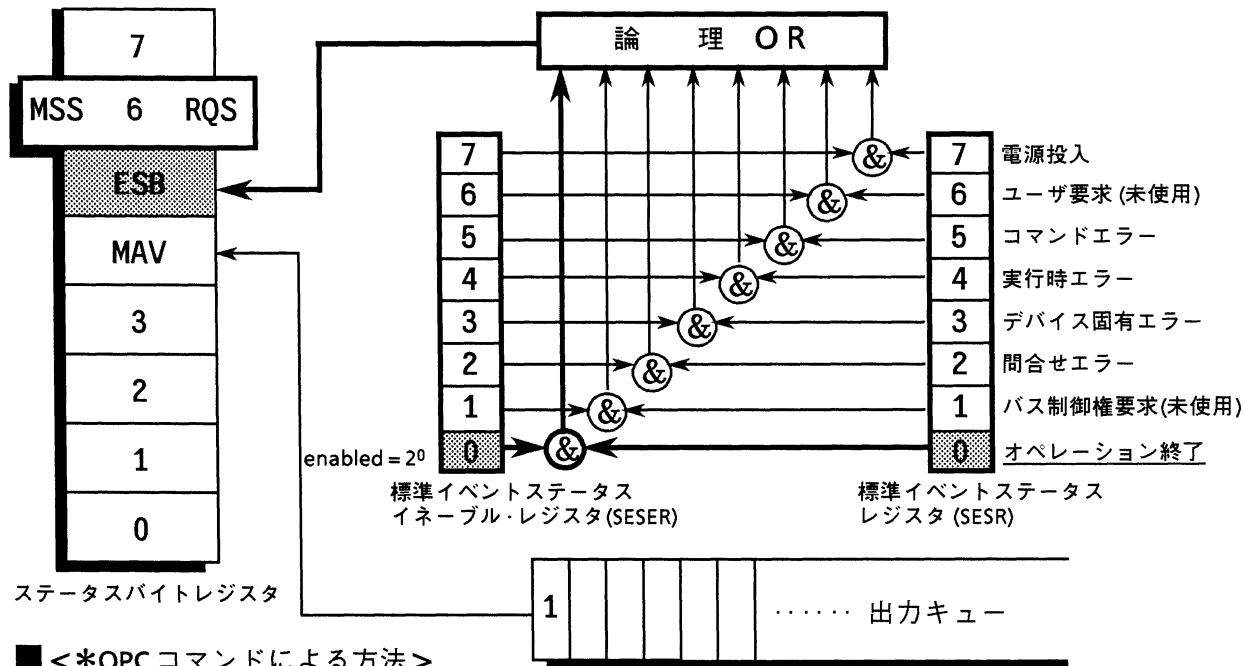
<出力キュー読み出し手順>



サービスリクエスト待ち

サービスリクエスト待ちとは、通常の処理を行っているコントローラがデバイスの **SRQ** 信号による割り込みのため、その通常の処理を一時中断し、デバイスのステータスメッセージの条件に対応した処理を行うことです。

本来の割り込み処理は、コントローラが現在どんな処理をやっているかにはおかまいなくデバイス側が要求してくるものですが、デバイスとコントローラ間の同期テクニックにおいては、あらかじめコントローラがデバイスの動作が終了したかどうかを確認するため、***OPC** コマンドまたは***OPC?** 問合せをデバイスに送ります。そして、動作終了イベントによる **SRQ** 信号を待つ間、コントローラは、何か別の有用な仕事を進め、動作終了イベントを検出すると、指定された処理をするというのが、サービスリクエスト待ちによる同期テクニックです。



<*OPC コマンドによる方法>

- ①標準イベントステータス・イネーブルレジスタの 2^0 ビットをイネーブルにする。

WRITE @103:"*ESE 1"
- ↓
- ②サービスリクエスト・イネーブルレジスタの 2^5 ビットをイネーブルにする。

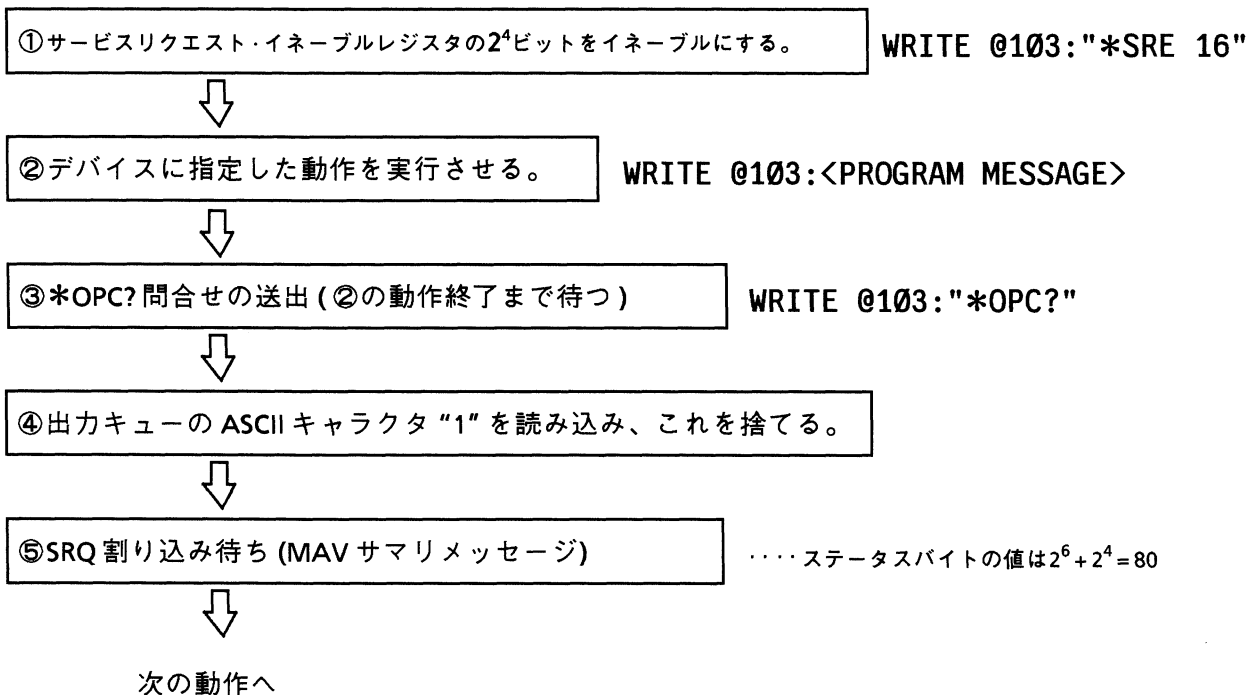
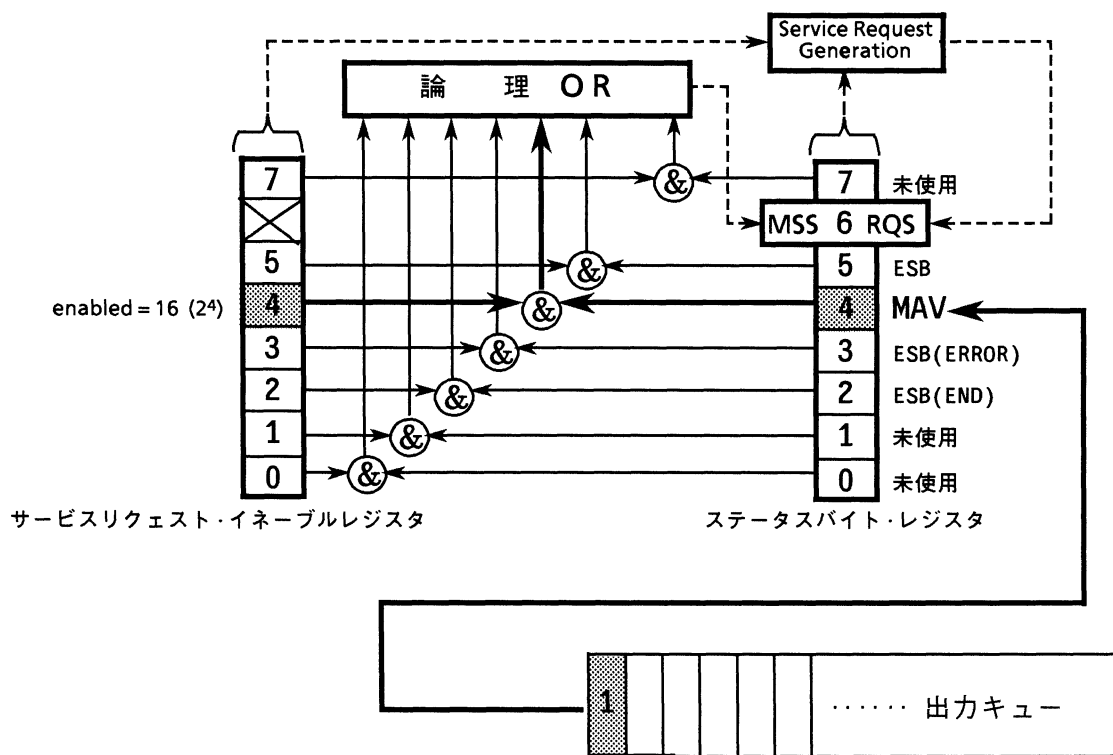
WRITE @103:"*SRE 32"
- ↓
- ③デバイスに指定した動作を実行させる。

WRITE @103:<PROGRAM MESSAGE >
- ↓
- ④***OPC** コマンド実行

WRITE @103:"*OPC"
- ↓
- ⑤**SRQ** 割り込み待ち (ESBサマリメッセージ)

・・・ステータスバイトの値は $2^6 + 2^5 = 96$

■ <*OPC? 問合せによる方法>



9 章

デバイス・メッセージの詳細

この章では、各デバイス・メッセージの一覧表と各デバイス・メッセージの詳細について説明します。

本書での書式、使用例はヒューレット・パッカード社 **HP9000** シリーズの **HP-BASIC** で説明しています。

目 次

デバイス・メッセージ一覧表	9-3
デバイス・メッセージ一覧表 (アルファベット順)	9-3
デバイス・メッセージ一覧表 (パネルとの対応)	9-9
デバイス・メッセージの詳細説明	9-24

9章 デバイス・メッセージの詳細

この章では、各デバイスメッセージをグループごとに説明します。各グループは、MP1762Dの正背面パネルに対応しているので、設定または要求の内容によりグループを選択して参照します。

デバイス・メッセージ一覧表

MP1762Dで規定される、コントロール・メッセージおよびデータ・リクエスト・メッセージについて一覧表の形で説明します。

各コマンドについての詳細は一覧表内のページにしたがって「デバイス・メッセージの詳細説明」の所で確認して下さい。

デバイス・メッセージ一覧表(アルファベット順)

各コントロール・メッセージおよびデータ・リクエスト・メッセージについてアルファベット順にした一覧表を表9-1に示します。

表9-1 デバイス・メッセージ一覧表 (アルファベット順)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細	
	ヘッダ部	数値データ部	ヘッダ部	セクション	ページ
ページ数	ADR	NR1形式	ADR?	PATTERN	P9-63
パターンデータ・プリセット (全ページ、全ビット)	ALL	NR1形式	—	PATTERN	P9-82
アラーム・モニタ (アラーム検出)	ALM	NR1形式	ALM?	その他	P9-124
オルタネート・パターン A/B 切替選択	ALT	NR1形式	ALT?	PATTERN	P9-58
アラーム測定結果	—	—	AMD?	MEASUREMENT	P9-112
パターン・ビット	BIT	NR1形式 HEX 形式	BIT?	PATTERN	P9-64
BURST 測定モード	BST	NR1形式	BST?	その他	P9-132
測定途中データ算出機能	CAL	NR1形式	CAL?	その他	P9-133
クロック個数測定結果	—	—	CC?	MEASUREMENT	P9-107
ビット・ウインドウ・パターン	CHM	NR1形式 HEX 形式	CHM?	PATTERN	P9-66
クロック断状態	—	—	CLI?	MEASUREMENT	P9-91
クロック断処理機能	CLS	NR1形式	CLS?	その他	P9-129
クロック入力位相	CPA	NR1形式	CPA?	INPUT	P9-28
クロック入力極性	CPL	NR1形式	CPL?	INPUT	P9-34
クロック入力終端電圧	CTM	NR1形式	CTM?	INPUT	P9-31
途中結果表示機能	CUR	NR1形式	CUR?	MEASUREMENT	P9-95
フロッピー・データ・デリート	DEL	NR1形式	—	MEMORY	P9-44
データ長	DLN	NR1形式	DLN?	PATTERN	P9-60
ディレイ状態	—	—	DLY?	INPUT	P9-32
測定結果表示モード	DMS	NR1形式	DMS?	MEASUREMENT	P9-94
1秒データ印字スレッシュホールド 選択	DOT	NR1形式	DOT?	その他	P9-141
表示選択	DSP	NR1形式	DSP?	PATTERN	P9-57
データ入力スレッシュホールド電圧	DTH	NR2形式	DTH?	INPUT	P9-26
データ入力終端電圧	DTM	NR1形式	DTM?	INPUT	P9-30
誤り分析データ ※1	—	—	EAB?	PATTERN	P9-71
誤り分析ページ ※1	EAP	NR1形式	EAP?	PATTERN	P9-72

表9-1 デバイス・メッセージ一覧表 (アルファベット順: つづき)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細	
	ヘッダ部	数値データ部	ヘッダ部	セクション	ページ
誤り分析トリガ ※1	EAT	NR1形式	EAT?	PATTERN	P9-75
誤り個数測定結果	—	—	EC?	MEASUREMENT	P9-106
測定終了データのバッファをクリア	EDC	—	—	MEASUREMENT	P9-114
測定終了データをバッファにストア	EDS	—	—	MEASUREMENT	P9-113
%EFI 測定結果	—	—	EFI?	MEASUREMENT	P9-109
EI 個数測定結果	—	—	EI?	MEASUREMENT	P9-108
EI, EFI インタバル時間	EIT	NR1形式	EIT?	その他	P9-135
アイ・マージン測定表示切替	EME	NR1形式	EME?	INPUT	P9-35
測定終了データの出力	—	—	END?	MEASUREMENT	P9-115
エラー・パフォーマンス・データ印字	EPF	NR1形式	EPF?	その他	P9-138
誤り率測定結果	—	—	ER?	MEASUREMENT	P9-105
誤り検出状態	—	—	ERS?	MEASUREMENT	P9-93
アイ・マージン測定開始	EST	NR1形式	EST?	INPUT	P9-36
エラー・パフォーマンス・データ・スレッシュホールド選択機能	ETH	NR1形式	ETH?	その他	P9-131
誤り検出モード	ETY	NR1形式	ETY?	その他	P9-134
アイ・マージン測定 (エラー・レシオ選択)	EYT	NR1形式	EYT?	INPUT	P9-37
FD エラーメッセージ	—	—	FDE?	MEMORY	P9-49
FD フォーマット	FDF	—	—	MEMORY	P9-50
ファイル No. / ディレクトリ モード切替	FIL	NR1形式	FIL?	MEMORY	P9-42
フレーム・ビット長	FLN	NR1形式	FLN?	PATTERN	P9-59
メモリ・FD モード	—	—	FMD?	MEMORY その他	P9-39
データ印字フォーマット	FMT	NR1形式	FMT?	その他	P9-136
クロック周波数データ	—	—	FRQ?	MEASUREMENT	P9-110
ファイル内容の検索	—	—	FSH?	MEMORY	P9-40

表9-1 デバイス・メッセージ一覧表(アルファベット順: つづき)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細	
	ヘッダ部	数値データ部	ヘッダ部	セクション	ページ
GPIB2 アドレス	GPA	NR1形式	GPA?	その他	P9-127
ビット・ウインドウ・プリセット (全ページ、全ビット)	HAL	NR1形式	—	PATTERN	P9-86
ビット・ウインドウ・プリセット (1ページ、全ビット)	HPS	NR1形式	—	PATTERN	P9-87
測定中間データのバッファを クリア	IMC	—	—	MEASUREMENT	P9-118
測定中間データの出力	—	—	IMD?	MEASUREMENT	P9-119
測定中間データをバッファに ストア	IMS	—	—	MEASUREMENT	P9-117
測定中間データ印字	ITM	NR1形式	ITM?	その他	P9-139
測定インタバル時間	ITV	NR1形式	ITV?	その他	P9-143
パターン論理	LGC	NR1形式	LGC?	PATTERN	P9-52
フロッピー・アクセス状態	—	—	MAC?	MEMORY	P9-48
ブロック・ウインドウ・ プリセット (全ページ、全ビット)	MAL	NR1形式	—	PATTERN	P9-84
メモリ・モード切替	MEM	NR1形式	MEM?	MEMORY	P9-47
ブロック・ウインドウ・パターン	MGB	NR1形式 HEX 形式	MGB?	PATTERN	P9-69
ブロック・ウインドウ ON/OFF	MGE	NR1形式	MGE?	PATTERN	P9-74
測定モード	MOD	NR1形式	MOD?	MEASUREMENT	P9-96
アラーム・モニタ (誤り検出)	MON	NR1形式	MON?	その他	P9-125
ブロック・ウインドウ・ プリセット (1ページ、全データ)	MPS	NR1形式	—	PATTERN	P9-85
ブロック・ウインドウ・データ 出力バイト数	—	—	MRD?	PATTERN	P9-80
PRBS マーク率	MRK	NR1形式	MRK?	PATTERN	P9-55
ビット・ウインドウ ON/OFF	MSE	NR1形式	MSE?	PATTERN	P9-73
ビット・ウインドウ・ページ	MSK	NR1形式	MSK?	PATTERN	P9-68
測定状態	—	—	MSR?	MEASUREMENT	P9-99
ブロック・ウインドウ・データ 入力バイト数	MWT	NR1形式	—	PATTERN	P9-78

表9-1 デバイス・メッセージ一覧表(アルファベット順: つづき)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細	
	ヘッダ部	数値データ部	ヘッダ部	セクション	ページ
1秒データ印字	OSC	NR1形式	OSC?	その他	P9-140
1秒データ測定結果	—	—	OSD?	MEASUREMENT	P9-111
ページ数	PAG	NR1形式	PAG?	PATTERN	P9-63
アイ・マージン測定結果 (フェーズ)	—	—	PHM?	INPUT	P9-29
PAGE / PATTERN SYNC POSITION切替	PPD	NR1形式	PPD?	PATTERN	P9-89
測定時間	PRD	NR1形式	PRD?	MEASUREMENT	P9-104
プリンタ ON / OFF	PRN	NR1形式	PRN?	その他	P9-122
マニュアル・プリント	PSA	—	—	その他	P9-123
PATTERN SYNC POSITION	PSP	NR1形式	PSP?	PATTERN	P9-88
パターン・データ・プリセット (1ページ、全ビット)	PST	NR1形式	—	PATTERN	P9-83
紙節約機能	PSV	NR1形式	PSV?	その他	P9-142
ZERO SUBST および PRBS 段数	PTN	NR1形式	PTN?	PATTERN	P9-54
測定パターン	PTS	NR1形式	PTS?	PATTERN	P9-53
フロッピー・データ・リコール	RCL	NR1形式	—	MEMORY	P9-43
パターン・データ出力バイト数	—	—	RED?	PATTERN	P9-77
フロッピー・データ・リセーブ	RSV	NR1形式	—	MEMORY	P9-46
内部タイマ設定	RTM	NR1形式	RTM?	MEASUREMENT	P9-103
フロッピー・データ・セーブ	SAV	NR1形式	—	MEMORY	P9-45
マーク率の AND ビットシフト数	SFT	NR1形式	SFT?	その他	P9-128
同期外れ状態	—	—	SLI?	MEASUREMENT	P9-92
同期外れ処理機能	SLS	NR1形式	SLS?	その他	P9-130
同期信号出力	SOP	NR1形式	SOP?	その他	P9-126
自動位相スレッシュホールド・サーチ 機能	SRH	NR1形式	SRH?	INPUT	P9-33
測定開始および再スタート	STA	—	—	MEASUREMENT	P9-97
測定停止	STO	—	—	MEASUREMENT	P9-98

表9-1 デバイス・メッセージ一覧表(アルファベット順: つづき)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細	
	ヘッダ部	数値データ部	ヘッダ部	セクション	ページ
自動同期スレッシュホールド	SYE	NR1形式	SYE?	MEASUREMENT	P9-101
同期方式	SYM	NR1形式	SYM?	PATTERN	P9-56
自動同期機能	SYN	NR1形式	SYN?	MEASUREMENT	P9-100
アイ・マージン測定結果 (スレッシュホールド)	—	—	THM?	INPUT	P9-27
スレッシュホールド EI, EFI データ 印字	THR	NR1形式	THR?	その他	P9-137
リアルタイム・測定時間表示切替	TIM	NR1形式	TIM?	MEASUREMENT	P9-102
終端コード切り替え	TRM	NR1形式	TRM?	その他	P9-144
パターン・データ入力バイト数	WRT	NR1形式	—	PATTERN	P9-76
ZERO SUBST 長	ZLN	NR1形式	ZLN?	PATTERN	P9-62

※1) オプション-01 : 誤り分析機能を搭載している場合にのみ有効となるコマンドです。

MP1762D : 付属新規コマンド

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細	
	ヘッダ部	数値データ部	ヘッダ部	セクション	ページ
入力データ切り替え	DSL	NR1形式	DCD?	INPUT	P9-145
Tracking/Adjust 操作	DAC	NR1形式	DCD?	INPUT	P9-146
入力信号数の設定	DSD	NR1形式	DCD?	INPUT	P9-147
供給クロックの選択	RRC	NR1形式	DRC?	INPUT	P9-148
PLL リセット/ PLL ステータス	RPL	—	RPL?	INPUT	P9-148
Recovery 周波数設定	RFQ	NR1形式	DRC?	INPUT	P9-150
終端の差動選択	DTC	NR1形式	DCD?	INPUT	P9-152

デバイス・メッセージ一覧表 (パネルとの対応)

各コントロール・メッセージおよびデータ・リクエスト・メッセージについてパネル上のキーに対応した形で図9-1-(1)～(7), および表9-2-(1)～(7)に示します。

● INPUT セクション

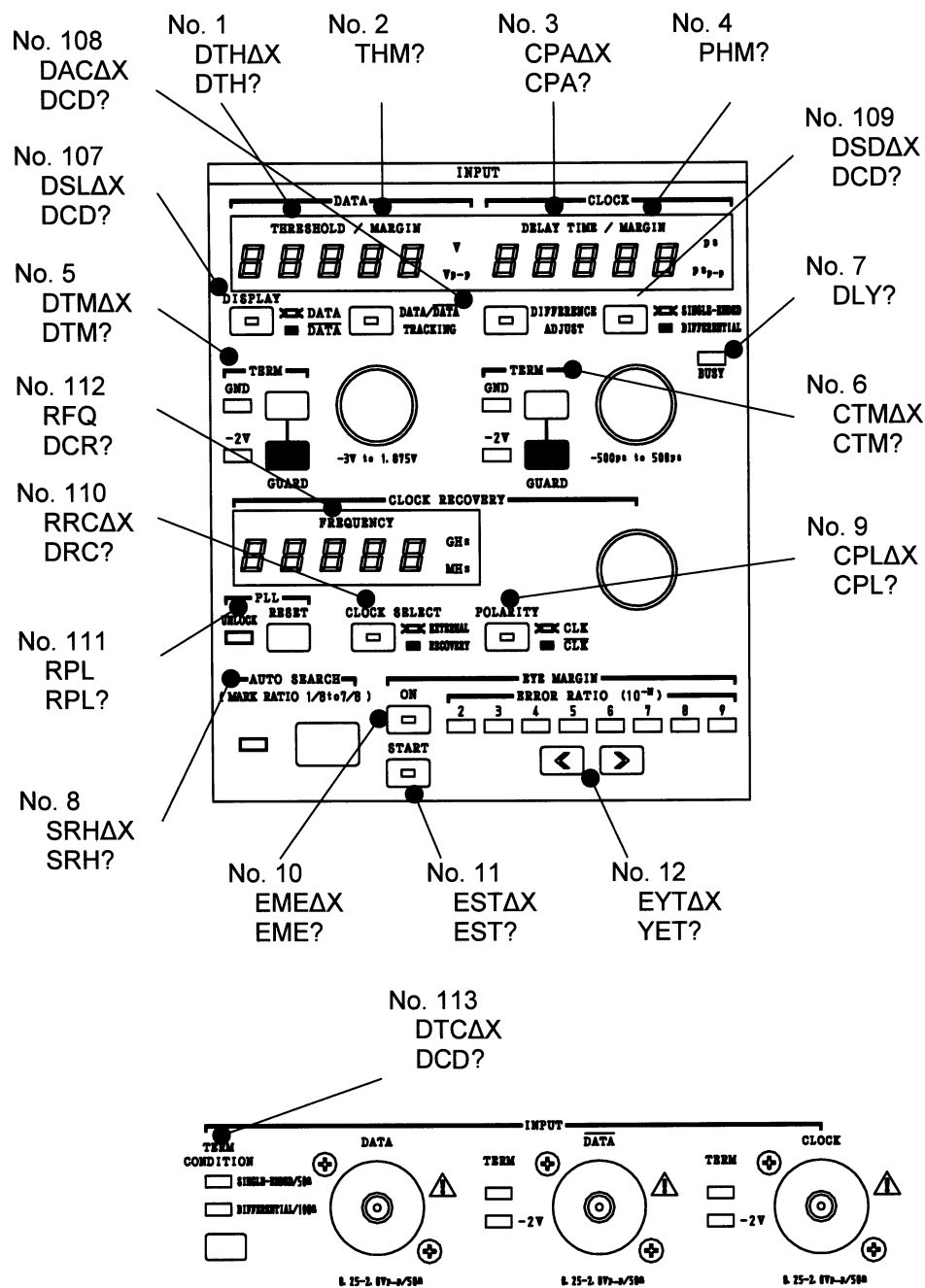
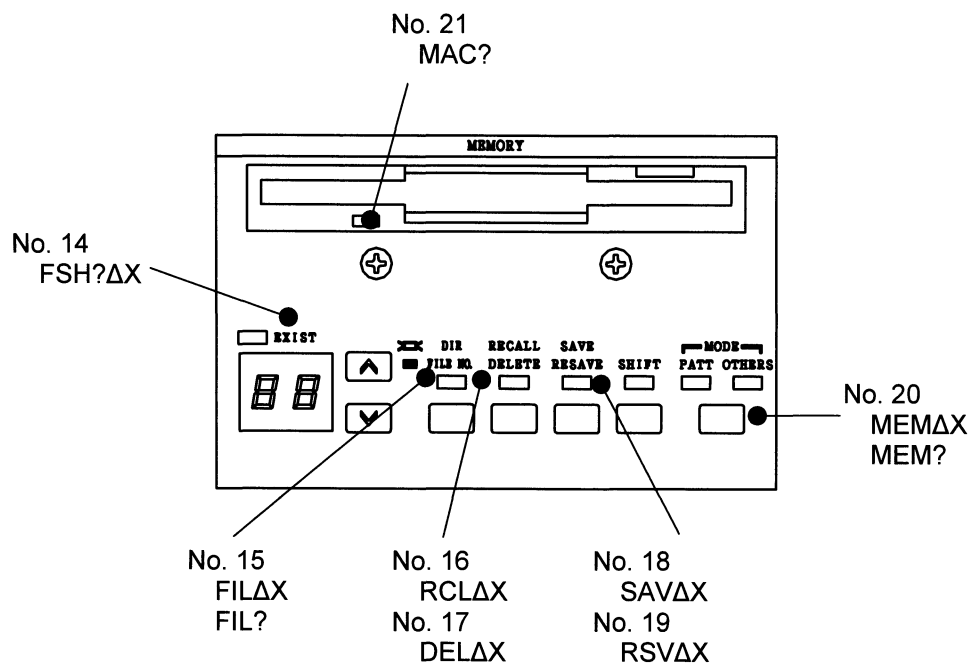


表9-2-(1) デバイス・メッセージ一覧表 (INPUT セクション)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細	
	ヘッダ部	数値データ部	ヘッダ部	項目 No.	ページ
● INPUT セクション					
データ入力スレッシュOLD電圧	DTH	NR2形式	DTH?	1	P9-26
アイ・マージン測定結果 (スレッシュOLD)	—	—	THM?	2	P9-27
クロック入力位相	CPA	NR1形式	CPA?	3	P9-28
アイ・マージン測定結果 (フェーズ)	—	—	PHM?	4	P9-29
データ入力終端電圧	DTM	NR1形式	DTM?	5	P9-30
クロック入力終端電圧	CTM	NR1形式	CTM?	6	P9-31
ディレイ状態	—	—	DLY?	7	P9-32
自動位相スレッシュOLD・サーチ機能	SRH	NR1形式	SRH?	8	P9-33
クロック入力極性	CPL	NR1形式	CPL?	9	P9-34
アイ・マージン測定表示切替	EME	NR1形式	EME?	10	P9-35
アイ・マージン測定開始	EST	NR1形式	EST?	11	P9-36
アイ・マージン測定 (エラー・レシオ選択)	EYT	NR1形式	EYT?	12	P9-37
入力データ切替	DSL	NR1形式	DCD?	107	P9-145
Tracking/Adjust 操作	DAC	NR1形式	DCD?	108	P9-146
入力信号数の設定	DSD	NR1形式	DCD?	109	P9-147
供給クロックの選択	RRC	NR1形式	DRC?	110	P9-148
PLL リセット/ PLL ステータス	RPL	—	RPL?	111	P9-149
Recovery 周波数設定	RFQ	NR1形式	DRC?	112	P9-150
終端の差動選択	DTC	NR1形式	DCD?	113	P9-152

● MEMORY セクション



- ・ FD mode : No. 13 FMD?
- ・ FD error message : No. 22 FDE?
- ・ FD format : No. 23 FDF

図9-1-(2) MEMORY セクション

表9-2-(2) デバイス・メッセージ一覧表 (MEMORY セクション)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細	
	ヘッダ部	数値データ部	ヘッダ部	項目 No.	ページ
● MEMORY セクション					
メモリ・FD モード	—	—	FMD?	13	P9-39
ファイル内容の検索	—	—	FSH?	14	P9-40
ファイル No./ディレクトリモード切替	FIL	NR1形式	FIL?	15	P9-42
フロッピー・データ・リコール	RCL	NR1形式	—	16	P9-43
フロッピー・データ・デリート	DEL	NR1形式	—	17	P9-44
フロッピー・データ・セーブ	SAV	NR1形式	—	18	P9-45
フロッピー・データ・リセーブ	RSV	NR1形式	—	19	P9-46
メモリ・モード切替	MEM	NR1形式	MEM?	20	P9-47
フロッピー・アクセス状態	—	—	MAC?	21	P9-48
FD エラーメッセージ	—	—	FDE?	22	P9-49
FD フォーマット	FDF	—	—	23	P9-50

● PATTEERN セクション

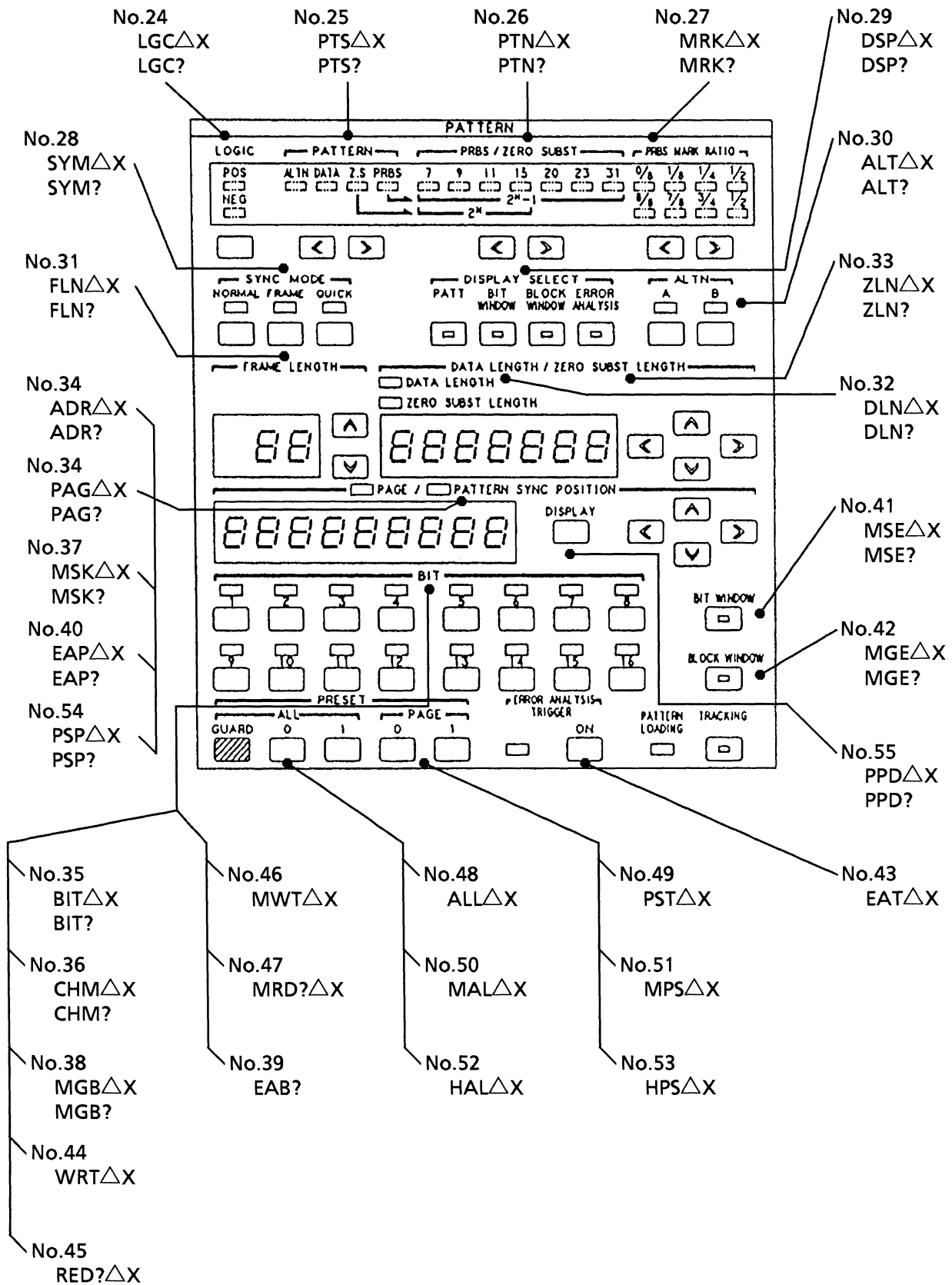


図9-1-(3) PATTERN セクション

表9-2-(3) デバイス・メッセージ一覧表 (PATTERN セクション)

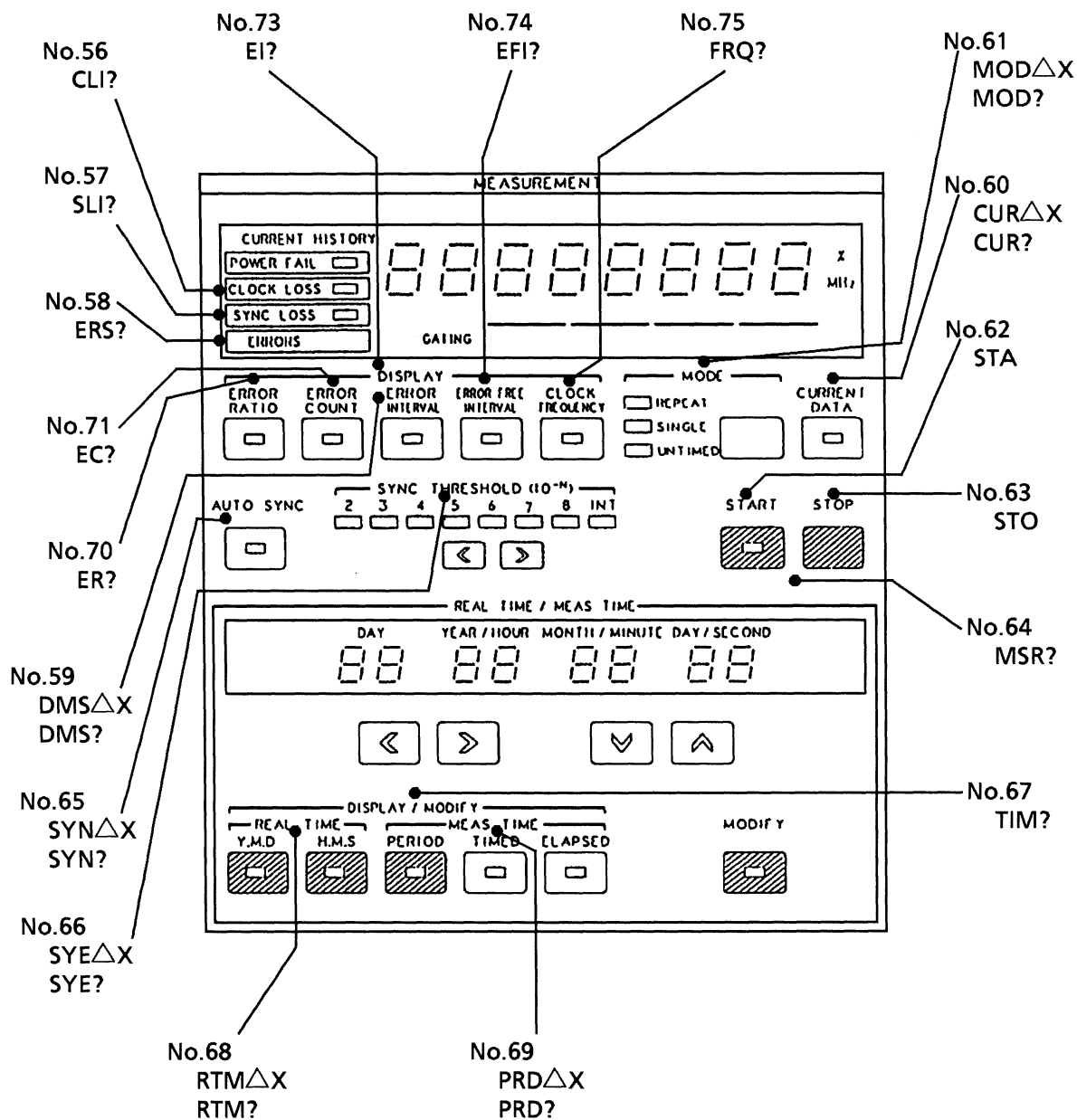
機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細	
	ヘッダ部	数値データ部	ヘッダ部	項目 No.	ページ
● PATTERN セクション					
パターン論理	LGC	NR1形式	LGC?	24	P9-52
測定パターン	PTS	NR1形式	PTS?	25	P9-53
ZERO SUBST および PRBS 段数	PTN	NR1形式	PTN?	26	P9-54
PRBS マーク率	MRK	NR1形式	MRK?	27	P9-55
同期方式	SYM	NR1形式	SYM?	28	P9-56
表示選択	DSP	NR1形式	DSP?	29	P9-57
オルタネート・パターン A/B 切替	ALT	NR1形式	ALT?	30	P9-58
フレーム・ビット長	FLN	NR1形式	FLN?	31	P9-59
データ長	DLN	NR1形式	DLN?	32	P9-60
ZERO SUBST 長	ZLN	NR1形式	ZLN?	33	P9-62
ページ数	PAG ADR	NR1形式 NR1形式	PAG? ADR?	34	P9-63
パターン・ビット	BIT	NR1形式 HEX 形式	BIT?	35	P9-64
ビット・ウインドウ・パターン	CHM	NR1形式 HEX 形式	CHM?	36	P9-66
ビット・ウインドウ・ページ	MSK	NR1形式	MSK?	37	P9-68
ブロック・ウインドウ・パターン	MGB	NR1形式 HEX 形式	MGB?	38	P9-69
誤り分析データ ※1	—	—	EAB?	39	P9-71
誤り分析ページ ※1	EAP	NR1形式	EAP?	40	P9-72
ビット・ウインドウ ON/OFF	MSE	NR1形式	MSE?	41	P9-73
ブロック・ウインドウ ON/OFF	MGE	NR1形式	MGE?	42	P9-74
誤り分析トリガ ※1	EAT	NR1形式	EAT?	43	P9-75
パターン・データ入力バイト数	WRT	NR1形式	—	44	P9-76

表9-2-(3) デバイス・メッセージ一覧表 (PATTERN セクション : つづき)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細	
	ヘッダ部	数値データ部	ヘッダ部	項目 No.	ページ
● PATTERN セクション (つづき)					
パターン・データ出力バイト数	—	—	RED?	45	P9-77
ブロック・ウインドウ・データ入力 バイト数	MWT	NR1形式	—	46	P9-78
ブロック・ウインドウ・データ出力 バイト数	—	—	MRD?	47	P9-80
パターンデータ・プリセット (全ページ、全ビット)	ALL	NR1形式	—	48	P9-82
パターンデータ・プリセット (1ページ、全ビット)	PST	NR1形式	—	49	P9-83
ブロック・ウインドウ・プリセット (全ページ、全ビット)	MAL	NR1形式	—	50	P9-84
ブロック・ウインドウ・プリセット (1ページ、全ビット)	MPS	NR1形式	—	51	P9-85
ビット・ウインドウ・プリセット (全ページ、全ビット)	HAL	NR1形式	—	52	P9-86
ビット・ウインドウ・プリセット (1ページ、全ビット)	HPS	NR1形式	—	53	P9-87
パターン・同期トリガ位置	PSP	NR1形式	PSP?	54	P9-88
ページ/パターン・同期トリガ位置 表示切替	PPD	NR1形式	PPD?	55	P9-89

※1) オプション-01 : 誤り分析機能を搭載している場合にのみ有効となるコマンドです。

● MEASUREMENT セクション



- ・ One-second data output : No.76 OSD? Δ X
- ・ Alarm measurement results : No.77 AMD? Δ X
- ・ Result data buffer clear : No.79 EDC
- buffer store : No.78 EDS
- output : No.80 END? Δ X
- ・ Intermediate data buffer clear : No.82 IMC
- buffer store : No.81 IMS
- output : No.83 IMD? Δ X

図9-1-(4) MEASUREMENT セクション

表9-2-(4) デバイス・メッセージ一覧表 (MEASUREMENT セクション)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細	
	ヘッダ部	数値データ部	ヘッダ部	項目 No.	ページ
● MEASUREMENT セクション					
クロック断状態	—	—	CLI?	56	P9-91
同期外れ状態	—	—	SLI?	57	P9-92
誤り検出状態	—	—	ERS?	58	P9-93
測定結果表示モード	DMS	NR1形式	DMS?	59	P9-94
途中結果表示機能	CUR	NR1形式	CUR?	60	P9-95
測定モード	MOD	NR1形式	MOD?	61	P9-96
測定開始および再スタート	STA	—	—	62	P9-97
測定停止	STO	—	—	63	P9-98
測定状態	—	—	MSR?	64	P9-99
自動同期機能	SYN	NR1形式	SYN?	65	P9-100
自動同期スレッシュホールド	SYE	NR1形式	SYE?	66	P9-101
リアルタイム・測定時間表示切替	TIM	NR1形式	TIM?	67	P9-102
内部タイマ設定	RTM	NR1形式	RTM?	68	P9-103
測定時間設定	PRD	NR1形式	PRD?	69	P9-104
誤り率測定結果	—	—	ER?	70	P9-105
誤り個数測定結果	—	—	EC?	71	P9-106
クロック個数測定結果	—	—	CC?	72	P9-107
EI 個数測定結果	—	—	EI?	73	P9-108
%EFI 測定結果	—	—	EFI?	74	P9-109
クロック周波数データ	—	—	FRQ?	75	P9-110
1秒データ測定結果	—	—	OSD?	76	P9-111
アラーム測定結果	—	—	AMD?	77	P9-112
測定終了データをバッファにストア	EDS	—	—	78	P9-113
測定終了データのバッファをクリア	EDC	—	—	79	P9-114

表9-2-(4) デバイス・メッセージ一覧表 (MEASUREMENT セクション : つづき)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細	
	ヘッダ部	数値データ部	ヘッダ部	項目 No.	ページ
● MEASUREMENT セクション (つづき)					
測定終了データの出力	—	—	END?	80	P9-115
測定中間データをバッファにストア	IMS	—	—	81	P9-117
測定中間データのバッファをクリア	IMC	—	—	82	P9-118
測定中間データの出力	—	—	IMD?	83	P9-119

● その他のセクション

(正面パネル)

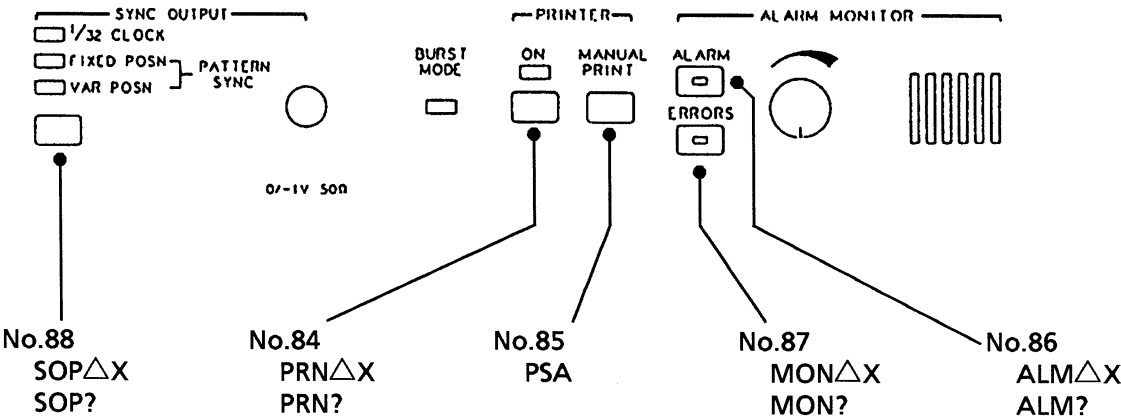


図9-1-(5) その他のセクション (正面パネル)

(背面パネル)

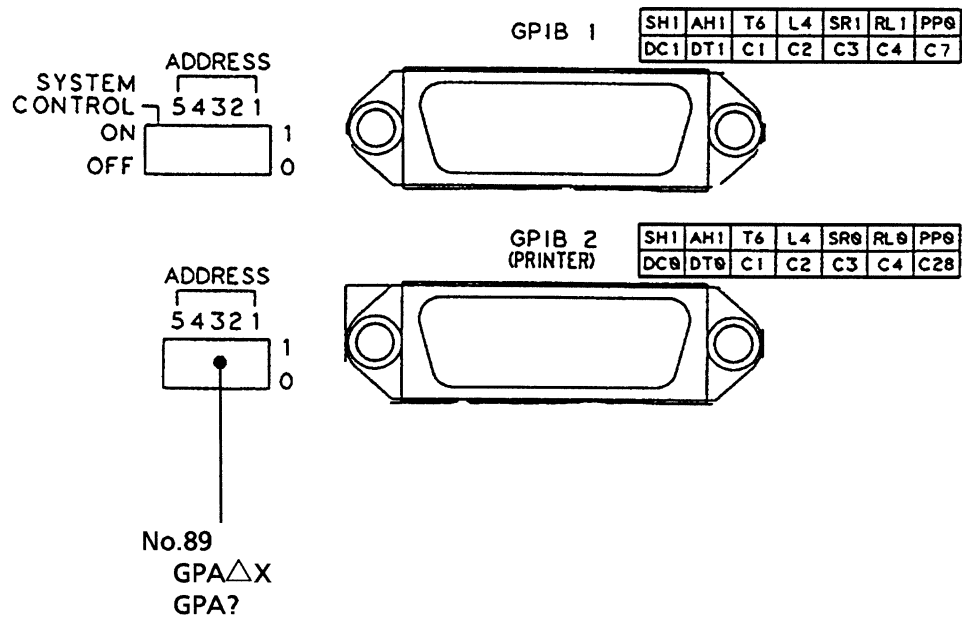


図9-1-(6) その他のセクション (背面パネル・GPIB)

NOTE

GPIB2のアドレス設定は、GPIB 経由で設定する場合と背面のアドレス SW で設定する場合の2通りの設定方法があります。

GPIB 経由で “GPA” コマンドによってアドレス設定された場合、本器がリモート状態の間は設定内容を保持します。(ただし、イニシャライズコマンドの INI, *RST を送出した場合はのぞきます。)

しかし、いったんローカル状態になった場合は、背面のアドレス SW の状態を優先してリモート状態での設定内容は無効となります。

表9-2-(5) デバイス・メッセージ一覧表 (その他のセクション : 正面パネル)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細	
	ヘッダ部	数値データ部	ヘッダ部	項目 No.	ページ
● その他のセクション (正面パネル)					
プリンタ ON/OFF	PRN	NR1形式	PRN?	84	P9-122
マニュアル・プリント	PSA	NR1形式	—	85	P9-123
アラーム・モニタ (アラーム検出)	ALM	NR1形式	ALM?	86	P9-124
アラーム・モニタ (誤り検出)	MON	NR1形式	MON?	87	P9-125
同期信号出力	SOP	NR1形式	SOP?	88	P9-126

表9-2-(6) デバイス・メッセージ一覧表 (その他のセクション : 背面パネル GPIB)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細	
	ヘッダ部	数値データ部	ヘッダ部	項目 No.	ページ
● その他のセクション (背面パネル GPIB)					
GPIB2アドレス	GPA	NR1形式	GPA?	89	P9-127

(背面パネル・ファンクション SW)

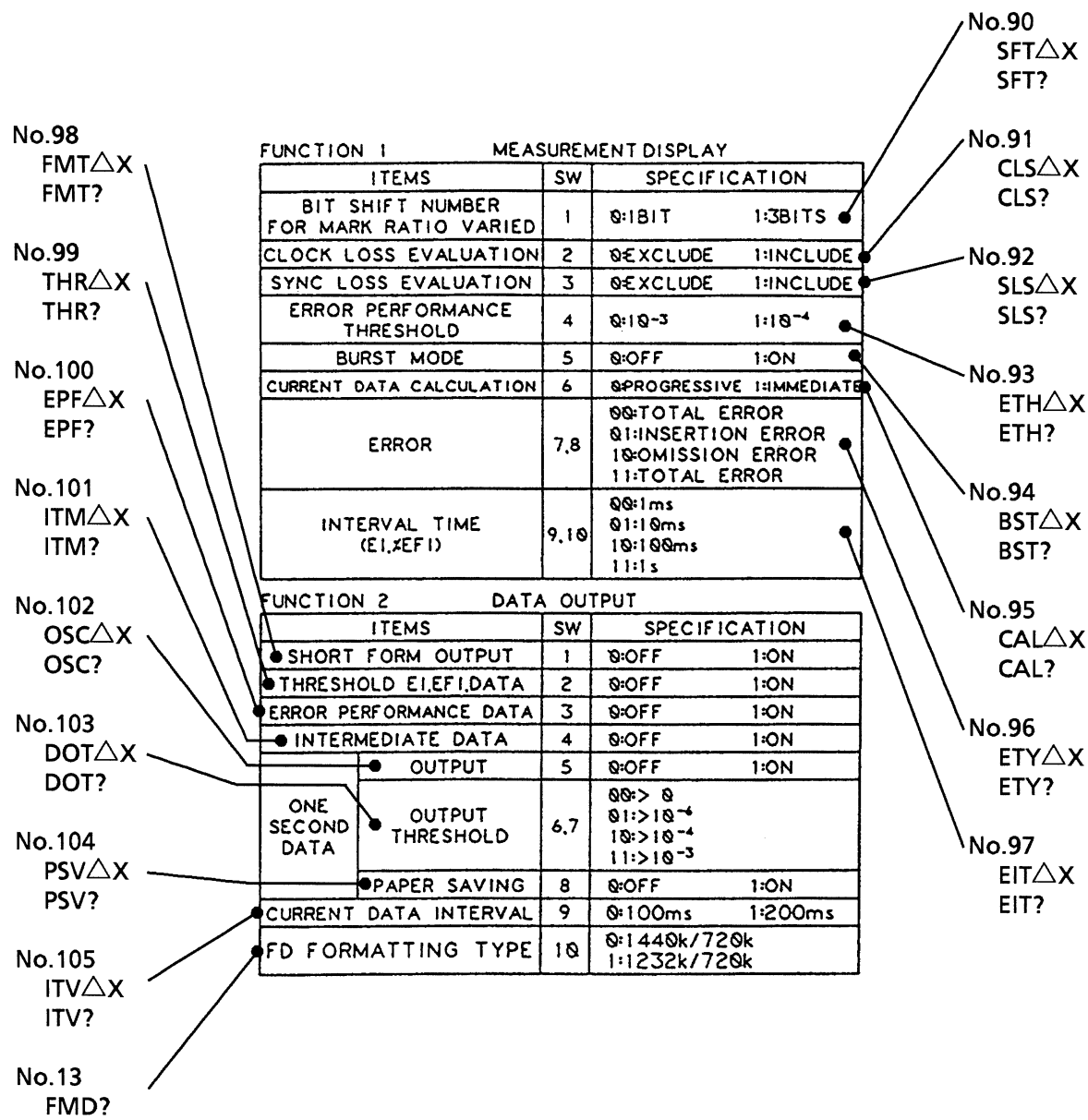


図9-1-(7) その他のセクション (背面パネル・ファンクション SW)

NOTE

背面ファンクション SW はリモート状態で設定された値と背面のスイッチの値が異なる場合は、以下の処理をしています。

- ・ リモートで設定された値は、本器がリモート状態を継続中は保持されます。(但し、イニシャライズコマンドのINI, *RSTを送出した場合は初期化されます。)しかし、いったん本器がローカル状態になると背面のファンクションSWの設定状態を優先して、リモート時の設定内容は無効となります。

表9-2-(7) デバイス・メッセージ一覧表
(その他のセクション：背面パネル・FUNCTION スイッチ)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細	
	ヘッダ部	数値データ部	ヘッダ部	項目 No.	ページ
● その他のセクション (背面パネル・FUNCTION スイッチ)					
マーク率の AND ビットシフト数	SFT	NR1形式	SFT?	90	P9-128
クロック断処理機能	CLS	NR1形式	CLS?	91	P9-129
同期外れ処理機能	SLS	NR1形式	SLS?	92	P9-130
エラー・パフォーマンス・データ・スレッシュホルド選択機能	ETH	NR1形式	ETH?	93	P9-131
BURST 測定モード	BST	NR1形式	BST?	94	P9-132
測定途中データ算出機能	CAL	NR1形式	CAL?	95	P9-133
誤り検出モード	ETY	NR1形式	ETY?	96	P9-134
EI, EFI インターバル時間	EIT	NR1形式	EIT?	97	P9-135
データ印字フォーマット	FMT	NR1形式	FMT?	98	P9-136
スレッシュホルド EI、EFI データ印字	THR	NR1形式	THR?	99	P9-137
エラー・パフォーマンス・データ印字	EPF	NR1形式	EPF?	100	P9-138
中間データ印字	ITM	NR1形式	ITM?	101	P9-139
1秒データ印字	OSC	NR1形式	OSC?	102	P9-140
1秒データ印字スレッシュホルド	DOT	NR1形式	DOT?	103	P9-141
紙節約機能	PSV	NR1形式	PSV?	104	P9-142
測定インターバル時間	ITV	NR1形式	ITV?	105	P9-143
メモリ・FD モード	—	—	FMD?	13	P9-39
終端コード切り替え	TRM	NR1形式	TRM?	106	P9-144

デバイス・メッセージの詳細説明

MP1762Dの個々のコントロール・メッセージおよびデータ・リクエスト・メッセージについて詳細な説明をします。

以下の例は、ヒューレット・パッカード社 **HP9000**シリーズの **HP-BASIC** で説明しています。

- INPUT セクション

次のページより、**INPUT** セクションの各コントロール・メッセージについて示します。なお、文中の△はスペースを意味します。

1) DTH データ入カスレッシュヨルド電圧 (Data THreshold)

■ 機能

入力されるデータのスレッシュヨルド電圧設定を行います。
設定分解能は、0.001 V です。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
DTH	DTH△m	DTH?	DTH△m (FIX 6)

■ m の値

データ入カスレッシュヨルド電圧設定範囲である1.875 V ～ -3.000 V までの値を設定します。

数値範囲 : 最大値 1.875
 最小値 -3.000
 ステップ 0.001

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュヨルド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : データ入カスレッシュヨルド電圧の設定を-3.000 V に設定する場合

OUTPUT△700;"DTH△-3"

問い合わせ : データ入カスレッシュヨルド電圧の設定値が1.000 V の場合

OUTPUT△700;"DTH?"

ENTER△700;B\$

PRINT△B\$

↓

DTH△1.000 (CR/LF) を出力します。

データ入カスレッシュヨルド電圧の設定値が-3.000 V の場合

OUTPUT△700;"DTH?"

ENTER△700;B\$

PRINT△B\$

↓

DTH△-3.000 (CR/LF) を出力します。

■ Note

入力データの切り替え (DATA,DATA) で選択されている側の設定およびクエリを実施します。

2) THM?

アイ・マージン測定結果(スレッシュヨルド)
(THreshold Margin ?)

■ 機能

アイ・マージン測定結果データの内、スレッシュヨルド・マージン測定結果を出力します。

測定分解能は、0.001 Vp-p です。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
THM	無し	THM?	THM△m (FIX 6)

■ m の値

アイ・マージン測定結果である 0.000 Vp-p ~ 4.875 Vp-p までの値を出力します。

数値範囲 : 最大値 4.875
 最小値 0.000
 分解能 0.001

但し、アイ・マージン測定結果が無い場合は、次のデータ出力を行います。

−9.999

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

問い合わせ : 無し

■ 使用例

問い合わせ : アイ・マージン測定結果が 1.875 Vp-p の場合

```
OUTPUT△700;"THM?"
ENTER△700;B$
PRINT△B$
```

↓

THM△△1.875 (CR/LF) を出力します。

アイ・マージン測定結果が無い場合

```
OUTPUT△700;"THM?"
ENTER△700;B$
PRINT△B$
```

↓

THM△−9.999 (CR/LF) を出力します。

■ Note

アイ・マージン測定結果データは、バッテリー・バック・アップの対象外となっています。そのため、電源 OFF 時には測定データが消えてしまいますので注意して下さい。

3) CPA クロック入力位相 (Clock Phase Adjust)

- 機能 入力されるクロック位相を設定します。
設定分解能は、1 ps です。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
CPA	CPA△m	CPA?	CPA△m (FIX 4)

- mの値 クロック入力位相の設定範囲である -500 ps ~ 500 ps までの値を設定します。

数値範囲 : 最大値 500
最小値 -500
分解能 1

- コマンド種別 シーケンシャル・コマンド

- 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

- 使用例 プログラム : クロック入力位相を0 ps に設定する場合
OUTPUT△700;"CPA△0"

問い合わせ : クロック入力位相の設定値が0 ps の場合
OUTPUT△700;"CPA?"
ENTER△700;B\$
PRINT△B\$

↓

CPA△△△△0 (CR/LF) を出力します。

クロック入力位相の設定値が-20 ps の場合

OUTPUT△700;"CPA?"
ENTER△700;B\$
PRINT△B\$

↓

CPA△△-20 (CR/LF) を出力します。

4) PHM? アイ・マージン測定結果(フェーズ) (PHase Margin ?)

■ 機 能

アイ・マージン測定結果データの内、フェーズ・マージン測定結果を出力します。

測定分解能は、1 psp-p です。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
PHM	無 し	PHM?	PHM△m (FIX 6)

■ m の値

アイ・マージン測定結果である 0 psp-p ~ 1000 psp-p までの値を出力します。

数値範囲 : 最大値 1000
 最小値 0
 分解能 1

但し、アイ・マージン測定結果が無い場合は、次のデータ出力を行います。
 -9999

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

問い合わせ : 無し

■ 使用例

問い合わせ : アイ・マージン測定結果が100 psp-p の場合

OUTPUT△700;"PHM?"

ENTER△700;B\$

PRINT△B\$

↓

PHM△△△100 (CR/LF) を出力します。

アイ・マージン測定結果が無い場合

OUTPUT△700;"PHM?"

ENTER△700;B\$

PRINT△B\$

↓

PHM△-9999 (CR/LF) を出力します。

■ Note

アイ・マージン測定結果データは、バッテリー・バック・アップの対象外となっています。そのため、電源 OFF 時には測定データが消えてしまいますので注意して下さい。

5) DTM データ入力終端電圧 (Data TerMination)

■ 機 能 データ入力の終端電圧を設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
DTM	DTM△m	DTM?	DTM△m (FIX 1)

■ mの値 0 : GND
1 : -2V

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例 プログラム : データ入力終端電圧を GND に設定する場合
OUTPUT△700;"DTM△0"

問い合わせ : データ入力終端電圧の設定値が-2Vの場合
OUTPUT△700;"DTM?"
ENTER△700;B\$
PRINT△B\$

↓

DTM△1 (CR/LF) を出力します。

注意

データ入力終端電圧設定値である GND / -2V の設定を入力されたデータの信号条件と異なって設定すると本器並びに被測定物を破損する恐れがありますので設定を誤らないように十分注意をして下さい。

6) CTM クロック入力終端電圧 (Clock TerMination)

- 機能 クロック入力の終端電圧を設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
CTM	CTM△m	CTM?	CTM△m (FIX 1)

- m の値 0 : GND
 1 : -2V

- コマンド種別 シーケンシャル・コマンド

- 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

- 使用例 プログラム : クロック入力終端電圧を GND に設定する場合
 OUTPUT△700;"CTM△0"

問い合わせ : クロック入力終端電圧の設定値が-2V の場合
 OUTPUT△700;"CTM?"
 ENTER△700;B\$
 PRINT△B\$

↓

CTM△1 (CR/LF) を出力します。

⚠ 注意

クロック入力終端電圧設定値である GND/-2V の設定を入力されたデータの信号条件と異なって設定すると本器並びに被測定物を破損する恐れがありますので設定を誤らないように十分注意して下さい。

7) DLY? ディレイ状態 (DeLaY ?)

■ 機 能 クロック入力位相を設定するサーボ・モーターの動作状態を出力します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
DLY	無 し	DLY?	DLY△m (FIX 1)

■ mの値 0 : READY 状態
 1 : BUSY 状態

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。
問い合わせ : 無し

■ 使用例 問い合わせ : クロック入力位相を設定するサーボ・モーターが動作停止中の場合
 OUTPUT△700;"DLY?"
 ENTER△700;B\$
 PRINT△B\$
 ↓
 DLY△0(CR/LF)を出力します。

■ Note READYとは、クロック入力位相を設定するサーボ・モーター回路が動作停止状態であることを示します。
 BUSYとは、クロック入力位相を設定するサーボ・モーター回路が動作状態であることを示します。
 そのため、BUSY 状態の場合は、本器がクロック入力位相を設定中であることを示しています。

8) SRH 自動位相スレッシュホールド・サーチ機能 (auto SeaRch)

機能

本器に入力されたクロック・データ間の位相関係およびデータのスレッシュホールド電圧を最適な状態に設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
SRH	SRH△m	SRH?	SRH△m (FIX 1)

mの値

- 0 : 自動位相スレッシュホールド・サーチ機能 OFF
 - 1 : 自動位相スレッシュホールド・サーチ機能 ON
 - 2 : 自動位相スレッシュホールド・サーチ機能失敗
- 2は、問い合わせに対するレスポンスのみ有効です。

コマンド種別

シーケンシャル・コマンド

使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動同期機能が OFF の場合
 測定パターンのマーク率が1/8～7/8で無い場合
 クロック断が発生している場合
 フロッピー・ディスクをアクセス中の場合
 自動位相スレッシュホールド・サーチ機能が実行中の場合
 自動位相スレッシュホールド・サーチ機能が失敗している状態での ON 設定
 アイ・マージン測定が実行中の場合

問い合わせ : 無し

使用例

プログラム : 自動位相スレッシュホールド・サーチ機能を ON に設定する場合
OUTPUT;"SRH△1"

問い合わせ : 自動位相スレッシュホールド・サーチ機能が失敗している場合
OUTPUT△700;"SRH?"
ENTER△700;B\$
PRINT△B\$

↓

SRH△2 (CR/LF) を出力します。

Note

- ① 自動位相スレッシュホールド・サーチ機能は、
 - (1) 動作中にクロック断が発生した場合
 - (2) サーチした設定状態で同期引き込みができなかった場合
 失敗と判定して、正面パネルの **AUTO SERACH** キー **LED** を点滅表示し、**SRH△2**の状態となります。
- ② 自動位相スレッシュホールド・サーチ機能が失敗している状態を解除するためには **OFF** の設定を行うようにして下さい。
 失敗状態では、再度自動位相スレッシュホールド・サーチ機能を **ON** することはできません。

9) CPL クロック入力極性 (Clock PoLarity)

■ 機 能 クロック入力の極性を設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
CPL	CPL△m	CPL?	CPL△m (FIX 1)

■ mの値 0 : CLOCK
 1 : CLOCK

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例 プログラム : クロック入力極性を CLOCK に設定する場合
 OUTPUT△700;"CPL△0"

問い合わせ : クロック入力極性の設定値が CLOCK の場合
 OUTPUT△700;"CPL?"
 ENTER△700;B\$
 PRINT△B\$

↓

CPL△1 (CR/LF) を出力します。

10) EME

アイ・マージン測定表示切替 (Eye Margin Enable)

■ 機能

INPUT セクションのデータ及びクロックに対する 7 seg 表示器の表示内容を切替ます。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
EME	EME△m	EME?	EME△m (FIX 1)

■ m の値

0 : 7 seg 表示器の表示値は、データ入力スレッシュホールド電圧設定値およびクロック入力位相設定値を示します。

1 : 7 seg 表示器の表示値は、アイ・マージン測定結果を示します。

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : 7 seg 表示器の表示値が、データ入力スレッシュホールド電圧設定値およびクロック入力位相設定値の場合

OUTPUT△700;"EME△0"

問い合わせ : 7 seg 表示器の表示値が、アイ・マージン測定結果の場合

OUTPUT△700;"EME?"

ENTER△700;B\$

PRINT△B\$

↓

EME△1 (CR/LF) を出力します。

■ Note

本設定は、パネル上での操作時にアイ・マージン測定を有効にするためのものです。

GPIB コマンドでアイ・マージン測定を制御する場合には、本設定状態に無関係にアイ・マージン測定関連の設定および問い合わせが可能です。

11) EST アイ・マージン測定開始 (Eye margin SStart)

■ 機 能 アイ・マージン測定の開始・停止を設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
EST	EST△m	EST?	EST△m (FIX 1)

■ **mの値**

- 0** : アイ・マージン測定停止
- 1** : アイ・マージン測定開始
- 2** : アイ・マージン測定失敗

2は、問い合わせに対するレスポンスのみ有効です。

■ コマンド種別 シーケンシャル・コマンド

■ **使用制限** 次の設定条件の場合は、無効となります。

プログラム : 自動同期機能が **OFF** の場合
測定パターンのマーク率が $1/8 \sim 7/8$ で無い場合
クロック断が発生している場合
フロッピー・ディスクをアクセス中の場合
自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が失敗している状態での **ON** 設定
アイ・マージン測定が実行中の場合

問い合わせ：無し

■ 使用例

プログラム : アイ・マージン測定を OFF に設定する場合
OUTPUT:"EST△0"

```

問い合わせ：アイ・マージン測定が失敗している場合
OUTPUT△700;"EST?"
ENTER△700;B$
PRINT△B$

```

↓

EST△2 (CR/LF) を出力します。

■ **Note**

アイ・マージン測定は、

- (1) 動作中にクロック断が発生した場合
- (2) アイ・マージン測定した設定状態で同期引き込みができなかった場合

失敗と判定して、正面パネルの **EYE MARGIN START** キー LED を点滅表示し、**EST△2**の状態となります。

アイ・マージン測定が失敗している状態を解除するためには **OFF** の設定を行うようにして下さい。

失敗状態では、再度アイ・マージン測定スタートをすることはできません。

12) EYT

アイ・マージン測定 (エラー・レシオ選択)
(EYe margin Threshold)

■ 機 能

アイ・マージン測定範囲をエラー・レシオで設定します。
設定されたエラー・レシオ範囲内のアイ・マージンが測定されます。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
EYT	EYT△m	EYT?	EYT△m (FIX 1)

■ m の値

0 : $\leq 1.0E-2$
 1 : $\leq 1.0E-3$
 2 : $\leq 1.0E-4$
 3 : $\leq 1.0E-5$
 4 : $\leq 1.0E-6$
 5 : $\leq 1.0E-7$
 6 : $\leq 1.0E-8$
 7 : $\leq 1.0E-9$

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : アイ・マージン測定エラー・レシオ範囲を $\leq 1.0E-5$ とする場合
OUTPUT;"EYT△3"

問い合わせ : アイ・マージン測定エラー・レシオ範囲の設定値が $\leq 1.0E-7$ の場合

OUTPUT△700;"EYT?"
ENTER△700;B\$
PRINT△B\$

↓

EYT△5 (CR/LF) を出力します。

- MEMORY セクション

次のページより、**MEMORY** セクションの各コントロール・メッセージについて示します。なお、文中の△はスペースを意味します。

13) FMD? メモリ・FDモード (memory Fd MoDe?)

■ 機 能

フロッピー・ディスクのフォーマットタイプを出力します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
FMD	無 し	FMD?	FMD△m (FIX 1)

■ mの値

0 : 1440 k
1 : 720 k
2 : 1232 k
3 : 640 k

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

問い合わせ : 無し

■ 使用例

問い合わせ : 挿入されている FD が 2DD で 1440 kB / 720 k のフォーマット形式が設定されている場合

OUTPUT△700;"FMD?"

ENTER△700;B\$

PRINT△B\$

↓

FMD△1 (CR/LF) を出力します。

■ Note

MS-DOS フォーマット形式の選択は、背面 FUNCTION2 SW10で行います。この設定は電源投入時の状態で決定するため、電源投入後に設定を変更することはできません。

● FUNCTION2 SW10 = 'OFF'の時

フォーマット時の容量	セクタ長 [バイト/セクタ]	セクタ数 [セクタ/トラック]	トラック数 [トラック/サイド]
1440 kB	512	18	80
720 kB	512	9	80

● FUNCTION2 SW10 = 'ON'の時

フォーマット時の容量	セクタ長 [バイト/セクタ]	セクタ数 [セクタ/トラック]	トラック数 [トラック/サイド]
1232 kB	1024	8	77
640 kB	512	8	80

フロッピー・ディスクが挿入されていない時に本問い合わせを実行すると現在設定されている FDD の状態 (FUNCTION2 SW10) に従って 1440 kB もしくは 1232 kB 時の情報を出力します。

14) FSH? ファイル内容の検索 (File Search?)

■ 機 能 フロッピー・ディスク内に保存されているデータの情報を出力します。
対象となるファイル名は、下記の3種類です。

TT***.PTN
RR***.PTN RR***.OTH

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
FSH	無 し	FSH?△m1	FSH△m2,m3,m4,m5, m2 (FIX 7) m3 (FIX 7) m4 (FIX 2) m5 (各FIX 2)

■ m の値 m1 : ファイル No. の前後半の選択
 0 : 前半 (No. 0 ~ No. 49)
 1 : 後半 (No. 50 ~ No. 99)
 m2 : Unused size
 m3 : Used size
 m4 : File 数
 m5 : 前半もしくは後半の File No. (対象となるファイル名のみ)

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

問い合わせ : 無し

■ 使用例 問い合わせ : フロッピー・ディスク内にファイル No. 1 ~ 10のファイルが存在する場合 (Unused size および Used size は一例です。)

```
OUTPUT△700;"FSH?△0"  
ENTER△700;B$  
PRINT△B$  
↓  
FSH△△72294,△△18132,10,01,02,03,04,05,06,  
07,08,09,10 (CR/LF) を出力します。  
  
フロッピー・ディスク内に対象となるファイルが存在しない場合  
OUTPUT△700;"FSH?△0"  
ENTER△700;B$  
PRINT△B$  
↓  
FSH△△723968,△△△6144,△0,--(CR/LF) を出力します。
```

■ Note

ファイル内容の検索は、メモリ・モード切替 (**PATT/OTHERS**) の設定状態に従っています。

(1) メモリ・モード切替が **PATT** の場合

TT*.PTN** および **RR***.PTN** のファイルを検索します。

(2) メモリ・モード切替が **OTHERS** の場合

RR*.OTH** のファイルを検索します。

MP1761B/C で保存した **PATT** ファイルと **MP1762D** で保存した **PATT** ファイルが同一ファイル名でフロッピー・ディスクに保管されている場合は **MP1762D** で保存されたファイル名を優先して出力します。

ファイル内容の検索は、その直前のディレクトリ情報を出力しています。

また本器はフロッピーの挿入検出機能がないため、フロッピー・ディスク交換時に以前のディレクトリ情報が更新されません。

そのため、フロッピー・ディスク挿入および交換時には、必ず **DIR** モードに切替をして、ディレクトリ情報の更新を行って下さい。

15) FIL ファイルNo./ディレクトリモード切替 (FILE no. / directory mode)

■ 機 能 ファイルNo.・モードとディレクトリ・モードの切替設定をします。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
FIL	FIL△m	FIL?	FIL△m (FIX 1)

■ m の値 0 : File No.モード
1 : DIR モード

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例 プログラム : DIR モードに設定を行う場合
 OUTPUT△700;"FIL△1"

この時、挿入されているフロッピー・ディスクをアクセスし、
DIR 情報を内部メモリに格納します。

問い合わせ : File No. モードに設定されている場合
 OUTPUT△700;"FIL?"
 ENTER△700;B\$
 PRINT△B\$

↓

FIL△0 (CR/LF) を出力します。

■ Note ファイル・アクセス時にエラーが発生した場合は、エラーとなり MEMORY
表示器にエラー情報を表示します。

このエラー表示は、下記の設定を行った場合にクリアされます。

ファイル No./ディレクトリモード切替 (No. 15)
フロッピー・データ・リコール (No. 16)
フロッピー・データ・デリート (No. 17)
フロッピー・データ・セーブ (No. 18)
フロッピー・データ・リコール (No. 19)
メモリ・モード切替 (No. 20)
FD フォーマット (No. 23)

16) RCL フロッピー・データ・リコール (ReCaLI)

■ 機 能 フロッピー・ディスクの内容を本器に設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
RCL	RCL△m	無 し	無 し

■ m の値 ファイル名を0～99の範囲で設定します。

数値範囲： 最大値 99
 最小値 0
 ステップ 1

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム： 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

■ 使用例 プログラム： ファイル No. が9のファイル内容を本器に設定する場合
 OUTPUT△700;"RCL△9"

■ Note 指定したファイルが存在しない場合には、エラーとなり MEMORY 表示器にエラー情報を表示します。

このエラー表示は、下記の設定を行った場合にクリアされます。

ファイル No./ディレクトリモード切替 (No. 15)
 フロッピー・データ・リコール (No. 16)
 フロッピー・データ・デリート (No. 17)
 フロッピー・データ・セーブ (No. 18)
 フロッピー・データ・リコール (No. 19)
 メモリ・モード切替 (No. 20)
 FD フォーマット (No. 23)

また、拡張イベント・ステータス・レジスタ ESR3 (ERROR) bit 1 の FD 異常発生ビットを立てます。

メモリ・モードの設定状態によって下記のファイルを読みだします。

PATT モードの場合 : RR**.PTN もしくは TT**.PTN
 OTHERS モードの場合 : RR**.OTH

17) DEL フロッピー・データ・デリート (DElete)

■ 機能

フロッピー・ディスクの指定されたファイル No. を削除します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
DEL	DEL△m	無 し	無 し

■ m の値

ファイル名を0～99の範囲で設定します。

数値範囲： 最大値 99
最小値 0
ステップ 1

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

■ 使用例

プログラム : ファイル No. が9のファイル内容を削除する場合
OUTPUT△700:"DEL△9"

Note

指定したファイルが存在しない場合には、エラーとなり **MEMORY** 表示器にエラー情報を表示します。

このエラー表示は、下記の設定を行った場合にクリアされます。

- ① ファイル No./ディレクトリモード切替 (No. 15)
 フロッピー・データ・リコール (No. 16)
 フロッピー・データ・デリート (No. 17)
 フロッピー・データ・セーブ (No. 18)
 フロッピー・データ・リコール (No. 19)
 メモリ・モード切替 (No. 20)
FD フォーマット (No. 23)

また、拡張イベント・ステータス・レジスタ **ESR3 (ERROR) bit 1** の **FD 異常発生ビット** を立てます。

本器では、MP1761B/Cで保存したファイルの削除できません。

メモリ・モードの設定状態によって下記のファイルを削除します。

PATT モードの場合 : RR**.PTN

OTHERS モードの場合： RR**.OTH

18) SAV**フロッピー・データ・セーブ (SAVe)****機能**

本器の設定内容をフロッピー・ディスクにセーブします。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
SAV	SAV△m	無 し	無 し

m の値

ファイル名を0～99の範囲で設定します。

数値範囲： 最大値 99
 最小値 0
 ステップ 1

コマンド種別 シーケンシャル・コマンド**使用制限**

次の設定条件の場合は、無効となります。

プログラム：自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

使用例

プログラム：ファイル No. 9で本器の設定内容を保存する場合
OUTPUT△700;"SAV△9"

Note

指定したファイルが存在する場合には、エラーとなり **MEMORY** 表示器にエラー情報を表示します。

このエラー表示は、下記の設定を行った場合にクリアされます。

ファイル No./ディレクトリモード切替 (No. 15)
 フロッピー・データ・リコール (No. 16)
 フロッピー・データ・デリート (No. 17)
 フロッピー・データ・セーブ (No. 18)
 フロッピー・データ・リコール (No. 19)
 メモリ・モード切替 (No. 20)
 FDフォーマット (No. 23)

また、拡張イベント・ステータス・レジスタ ESR3 (ERROR) bit 1の FD 異常発生ビットを立てます。

メモリ・モードの設定状態によって下記のファイルを保存します。

PATT モードの場合 : RR***.PTN もしくは TT***.PTN
 OTHERS モードの場合: RR***.OTH

19) RSV フロッピー・データ・リセーブ (ReSaVe)

■ 機 能 フロッピー・ディスクの内容を本器に設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
RSV	RSV△m	無 し	無 し

■ **m の値** ファイル名を0～99の範囲で設定します。

数値範囲： 最大値 99
最小値 0
ステップ 1

■ コマンド種別 シーケンシャル・コマンド

■ **使用制限** 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

■ 使用例 プログラム : ファイル No. が9のファイル内容を上書きする場合
 OUTPUT△700:"RSV△9"

■ **Note** 指定したファイルが存在しない場合には、エラーとなり **MEMORY** 表示器にエラー情報を表示します。

このエラー表示は、下記の設定を行った場合にクリアされます。

ファイル No./ディレクトリモード切替	(No. 15)
フロッピー・データ・リコール	(No. 16)
フロッピー・データ・デリート	(No. 17)
フロッピー・データ・セーブ	(No. 18)
フロッピー・データ・リコール	(No. 19)
メモリ・モード切替	(No. 20)
FDフォーマット	(No. 23)

また、拡張イベント・ステータス・レジスタ **ESR3 (ERROR) bit 1**の **FD 異常発生ビット**を立てます。

メモリ・モードの設定状態によって下記のファイルを再保存します。

PATT モードの場合 : RR**.PTN もしくは TT**.PTN
OTHERS モードの場合 : RR**.OTH

20) MEM メモリ・モード切替 (**MEM**ory mode)

■ 機 能 PATT/OTHERS の切替設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
MEM	MEM△m	MEM?	MEM△m (FIX 1)

■ m の値 0 : PATT モード
 1 : OTHERS モード

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例 プログラム : メモリ・モードを PATT に設定する場合
 OUTPUT△700;"MEM△0"

問い合わせ : メモリ・モードが OTHERS に設定されている場合
 OUTPUT△700;"MEM?"
 ENTER△700;B\$
 PRINT△B\$

↓

MEM△1 (CR/LF) を出力します。

■ Note PATT モードとは、PATTern モードを意味します。
 この場合は、メモリの内容は **PATTERN** セクションの内容を対象とします。
 OTHERS モードとは、上記以外の部分で測定データを含めない内容をメモリの対象とします

ファイル・アクセス等でエラーが発生した場合には、**MEMORY** 表示器にエラー情報を表示します。

このエラー表示は、下記の設定を行った場合にクリアされます。

- ファイル No./ディレクトリモード切替 (No. 15)
- フロッピー・データ・リコール (No. 16)
- フロッピー・データ・デリート (No. 17)
- フロッピー・データ・セーブ (No. 18)
- フロッピー・データ・リコール (No. 19)
- メモリ・モード切替 (No. 20)
- FD フォーマット (No. 23)

また、拡張イベント・ステータス・レジスタ **ESR3 (ERROR)** bit 1 の FD 異常発生ビットを立てます。

メモリ・モードの設定状態によって下記のファイルを再保存します。

- PATT モードの場合 : **RR**.PTN** もしくは **TT**.PTN**
- OTHERS モードの場合 : **RR**.OTH**

21) MAC?**フロッピー・アクセス状態 (Memory Access Condition?)****機能**

フロッピー・ディスクのアクセス状態を出力します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
MAC	無し	MAC?	MAC△m (FIX 1)

mの値

0 : ノンアクセス状態

1 : アクセス状態

コマンド種別

シーケンシャル・コマンド

使用制限

次の設定条件の場合は、無効となります。

問い合わせ : 無し

使用例

問い合わせ : フロッピーがアクセス中で無い場合

OUTPUT△700;"MAC?"

ENTER△700;B\$

PRINT△B\$

↓

MAC△0 (CR/LF) を出力します。

22) FDE?**FD エラー・メッセージ (FD Error message?)**

■ 機 能

FD のエラー情報を出力します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
FDE	無 し	FDE?	FDE△m (FIX 2)

■ m の値

エラー・メッセージ

- 0 : E0 (フォーマット形式の違いによるエラー)
- 1 : E1 (書き込み時にライトプロテクトされている)
- 2 : E2 (書き込み領域が不足している)
- 3 : E3 (読み出し時に指定されたファイル名が存在しない)
- 4 : E4 (同一ファイル名でセーブしようとした)
- 5 : E5 (書き込み障害エラー)
- 6 : E6 (読み出し障害エラー)
- 7 : E7 (DMA転送エラー)
- 8 : E8 (その他のエラー)
- 9 : E9 (ハードウェア・トラブル・エラー)
- 10 : No error

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

問い合わせ : 無し

■ 使用例

問い合わせ : ハードウェア・トラブル・エラーが発生している場合

```
OUTPUT△700;"FDE?"
ENTER△700;B$
PRINT△B$
```

↓

FDE△9 (CR/LF) を出力します。

■ Note

このエラー表示は、下記の設定を行った場合にクリアされます。

- ファイル No./ディレクトリモード切替 (No. 15)
- フロッピー・データ・リコール (No. 16)
- フロッピー・データ・デリート (No. 17)
- フロッピー・データ・セーブ (No. 18)
- フロッピー・データ・リコール (No. 19)
- メモリ・モード切替 (No. 20)
- FD フォーマット (No. 23)

また、拡張イベント・ステータス・レジスタ ESR3 (ERROR) bit 1 の FD 異常発生ビットを立てます。

23) FDF FD フォーマット (FD Format)

- 機能
- フロッピー・ディスクのフォーマットを行います。
フォーマット形式の選択は、背面 **FUNCTION2 SW10**で行って下さい。
また、**2HD/2DD** の判定は挿入された **FD** に対して自動的に検出し判定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
FDF	FDF	無し	無し

- mの値
- 無し
- コマンド種別
- シーケンシャル・コマンド
- 使用制限
- 次の設定条件の場合は、無効となります。
プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合
ファイル **No./ディレクトリ**モード切替がディレクトリ・モードの場合
- 使用例
- プログラム : フロッピー・ディスクのフォーマットを行う。
OUTPUT△700;"FDF"
- Note
- ファイル・アクセス等でエラーが発生した場合には、**MEMORY** 表示器にエラー情報を表示します。
このエラー表示は、下記の設定を行った場合にクリアされます。
ファイル **No./ディレクトリ**モード切替 (No. 15)
フロッピー・データ・リコール (No. 16)
フロッピー・データ・デリート (No. 17)
フロッピー・データ・セーブ (No. 18)
フロッピー・データ・リコール (No. 19)
メモリ・モード切替 (No. 20)
FD フォーマット (No. 23)
また、拡張イベント・ステータス・レジスタ **ESR3 (ERROR)** bit 1の **FD** 異常発生ビットを立てます。

- **PATTERN** セクション

次のページより、**PATTERN** セクションの各コントロール・メッセージについて示します。なお、文中の△はスペースを意味します。

24) LGC

パターン論理 (LoGiC mode)

■ 機能

データの論理設定を行います。

論理と実際のデータ出力の関係は、ALTERNATE / DATA / ZERO SUBST の場合と PRBS の場合で異なります。(機能・操作編取扱説明書を参照して下さい。)

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
LGC	LGC△m	LGC?	LGC△m (FIX 1)

■ m の値

0 : Positive

1 : Negative

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : パターン論理を正論理 (Positive) に設定を行う場合

OUTPUT△700;"LGC△0"

問い合わせ : パターン論理が負論理 (Negative) に設定されている場合

OUTPUT△700;"LGC?"

ENTER△700;B\$

PRINT△B\$

↓

LGC△1 (CR/LF) を出力します。

■ Note

パターンが PRBS モードの場合、パターン論理を設定すると、パターン・マーク率も論理に対応して切替わります。

● 正論理 0/8, 1/8, 1/4, 1/2

● 負論理 8/8, 7/8, 3/4, $\overline{1/2}$

25) PTS**測定パターン (PaTtern Select)****機能**

測定パターンの設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
PTS	PTS△m	PTS?	PTS△m (FIX 1)

mの値

0 : Alternate
 1 : Data
 2 : Zero subst
 3 : PRBS

コマンド種別 シーケンシャル・コマンド**使用制限**

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

使用例

プログラム : 測定パターンを **Alternate** に設定する場合
OUTPUT△700;"PTS△0"

問い合わせ : 測定パターン論理が **Data** に設定されている場合
OUTPUT△700;"PTS?"
ENTER△700;B\$
PRINT△B\$

↓

PTS△1 (CR/LF) を出力します。**Note**

測定パターンを切替えると、各測定パターンでの前回の設定状態が復帰します。

例えば、**PRBS** パターンに設定すると **PRBS** パターンの段数とパターン・マーク率の設定が復帰します。

26) PTN

ZERO SUBST / PRBS 段数 (zero subst / prbs PaTterN mode)

機能

ZERO SUBST / PRBS パターンの場合の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
PTN	PTN△m	PTN?	PTN△m (FIX 1)

m の値	ZERO SUBST の場合	PRBS の場合
	2 : 2 ⁷	2 : 2 ⁷ - 1
	3 : 2 ⁹	3 : 2 ⁹ - 1
	5 : 2 ¹¹	5 : 2 ¹¹ - 1
	6 : 2 ¹⁵	6 : 2 ¹⁵ - 1
		7 : 2 ²⁰ - 1
		8 : 2 ²³ - 1
		9 : 2 ³¹ - 1

コマンド種別

シーケンシャル・コマンド

使用制限

次の設定条件の場合は、無効となります。

プログラム

自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合
測定パターンが Alternate、Data の場合

問い合わせ

次の場合は無効となり、ERR (CR / LF) を出力します。
測定パターンが Alternate、Data の場合

使用例

プログラム : 測定パターンを PRBS 2⁷ - 1 に設定する場合
OUTPUT△700;"PTN△2"

問い合わせ : 測定パターン論理が PRBS 2³¹ - 1 に設定されている場合
OUTPUT△700;"PTN?"
ENTER△700;B\$
PRINT△B\$
↓
PTN△9 (CR / LF) を出力します。
:
測定パターンが DATA に設定されている場合
OUTPUT△700;"PTN?"
ENTER△700;B\$
PRINT△B\$
↓
ERR (CR / LF) を出力します。

27) MRK パターン・マーク率 (MaRK ratio mode)

■ 機 能

PRBS パターンの場合のマーク率設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
MRK	MRK△m	MRK?	MRK△m (FIX 1)

■ m の値

正論理 (Positive) の場合

0 : 0/8

1 : 1/8

2 : 1/4

3 : 1/2

負論理 (Negative) の場合

8/8

7/8

3/4

1/2

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合
 測定パターンが Alternate、Data、Zero subst の場合

問い合わせ : 次の場合は無効となり、ERR (CR/LF) を出力します。
 測定パターンが Alternate、Data、Zero subst の場合

■ 使用例

プログラム : パターン・マーク率を0/8に設定する場合
OUTPUT△700;"MRK△0"

問い合わせ : パターン・マーク率が1/8に設定されている場合
OUTPUT△700;"MRK?"
ENTER△700;B\$
PRINT△B\$

↓

MRK△1 (CR/LF) を出力します。

: 測定パターンが DATA に設定されている場合
OUTPUT△700;"MRK?"
ENTER△700;B\$
PRINT△B\$

↓

ERR (CR/LF) を出力します。

28) SYM 同期方式 (SYnc Mode)

■ 機 能 同期方式の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
SYM	SYM△m	SYM?	SYM△m (FIX 1)

■ mの値 0 : NORMAL
 1 : FRAME
 2 : QUICK

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

 プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合
 測定パターンが PRBS の場合
 測定パターンが DATA で、かつビット長が127ビット以下の
 場合の FRAME の選択
 測定パターンが ALTERNATE の場合の QUICK の選択

 問い合わせ : 次の場合は無効となり、ERR (CR/LF) を出力します。
 測定パターンが PRBS の場合

■ 使用例 プログラム : 同期方式を NORMAL に設定する場合
 OUTPUT△700;"SYM△0"

 問い合わせ : 同期方式が FRAME に設定されている場合
 OUTPUT△700;"SYM?"
 ENTER△700;B\$
 PRINT△B\$
 ↓
 SYM△1 (CR/LF) を出力します。

 : 測定パターンが PRBS に設定されている場合
 OUTPUT△700;"SYM?"
 ENTER△700;B\$
 PRINT△B\$
 ↓
 ERR (CR/LF) を出力します。

29) DSP 表示切替 (DiSPlay select)

■ 機 能 パターン部の表示切替の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
DSP	DSP△m	DSP?	DSP△m (FIX 1)

■ m の値 0 : PATTERN
 1 : BIT WINDOW
 2 : BLOCK WINDOW
 3 : ERROR ANALYSIS

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

 プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合
 次の場合は **BLOCK WINDOW** の設定は無効になります。
 測定パターンが **PRBS** の場合
 測定パターンが **DATA** の場合でビット長が**32**の倍数でない場合
 同期方式が **QUICK** の場合
 次の場合は **ERROR ANALYSIS** の設定は無効になります。
 OPTION-01を実装していない場合
 測定パターンが **ALTERNATE** の場合
 同期方式が **QUICK** の場合

問い合わせ : 無し

■ 使用例

 プログラム : 表示切替を **PATTERN** に設定する場合
 OUTPUT△700;"DSP△0"

 問い合わせ : 表示切替が **BIT WINDOW** に設定されている場合
 OUTPUT△700;"DSP?"
 ENTER△700;B\$
 PRINT△B\$
 ↓
 DSP△1 (CR/LF) を出力します。

30) ALT オルタネート・パターン A/B 表示切替 (ALTernate a / b)

■ 機能 オルタネート・パターン時の A/B 表示切替の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
ALT	ALT△m	ALT?	ALT△m (FIX 1)

■ m の値
 0 : A パターン
 1 : B パターン

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合
 測定パターンが DATA、ZERO SUBST、PRBS の場合

問い合わせ : 次の場合には無効となり、ERR (CR/LF) を出力します。
 測定パターンが DATA、ZERO SUBST、PRBS の場合

■ 使用例
 プログラム : オルタネート表示を A に切替える場合
OUTPUT△700;"ALT△0"

問い合わせ : オルタネート表示を A に切替える場合
OUTPUT△700;"ALT?"
ENTER△700;B\$
PRINT△B\$

↓

ALT△1 (CR/LF) を出力します。

: 測定パターンが PRBS に設定されている場合
OUTPUT△700;"ALT?"
ENTER△700;B\$
PRINT△B\$

↓

ERR (CR/LF) を出力します。

31) FLN**FRAME BIT 長 (Frame bit LeNght)**

■ 機 能

フレーム・ビット長の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
FLN	FLN△m	FLN?	FLN△m (FIX 2)

■ m の値

フレーム・ビット長を数値範囲4～32ビットまでの範囲で値を設定します。

最大値 : 32

最小値 : 4

ステップ : 4

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合
測定パターンが PRBS の場合
同期方式が NORMAL、QUICK の場合
フレーム・ビット長の設定値が4ステップで無い場合

問い合わせ : 次の場合には無効となり、ERR (CR/LF) を出力します。
測定パターンが PRBS の場合
同期方式が NORMAL、QUICK の場合

■ 使用例

プログラム : フレーム・ビット長を4ビットに設定する場合
OUTPUT△700;"FLN△4"

問い合わせ : フレーム・ビット長が32ビットに設定されている場合
OUTPUT△700;"FLN?"
ENTER△700;B\$
PRINT△B\$

↓

FLN△32 (CR/LF) を出力します。

: 測定パターンが PRBS に設定されている場合
OUTPUT△700;"FLN?"
ENTER△700;B\$
PRINT△B\$

↓

ERR (CR/LF) を出力します。

32) DLN

データ長 (Data Length)

■ 機能

測定パターンが ALTERNATE、DATA の場合にデータ長の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
DLN	DLN△m	DLN?	DLN△m (FIX 7)

■ m の値

データ長の設定を下記の範囲内で設定します。

- オルタネートパターンの場合
 - 最大値 : 4194304
 - 最小値 : 128
 - ステップ : 128
- データパターンの場合
 - 最大値 : 8388608
 - 最小値 : 2
 - ステップ : データ長によって下記の様に分類されます。

データ長

2 ~	65536ビット / ステップ	1ビット
65536 ~	131072ビット / ステップ	2ビット
131072 ~	262144ビット / ステップ	4ビット
262144 ~	524288ビット / ステップ	8ビット
524288 ~	1048576ビット / ステップ	16ビット
1048576 ~	2097152ビット / ステップ	32ビット
2097152 ~	4194304ビット / ステップ	64ビット
4194304 ~	8388608ビット / ステップ	128ビット

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合
 測定パターンが ZERO SUBST、PRBS の場合

問い合わせ : 次の場合には無効となり、ERR (CR/LF) を出力します。
 測定パターンが ZERO SUBST、PRBS の場合

■ 使用例

プログラム : データ長を4ビットに設定する場合
OUTPUT△700;"DLN△4"

問い合わせ : データ長が32ビットに設定されている場合
OUTPUT△700;"DLN?"
ENTER△700;B\$
PRINT△B\$

↓

DLN△△△△△32 (CR/LF) を出力します。

: 測定パターンが PRBS に設定されている場合
OUTPUT△700;"DLN?"
ENTER△700;B\$
PRINT△B\$

↓

ERR (CR/LF) を出力します。

■ Note

データ長の設定値が存在しない値を設定した場合はデータ長を下記の要領で最適化します。

入力した値よりも小さく、しかも入力した値に最も近い値に変更します。

例) ● 入力したデータ長		● 設定されるデータ長
131075	→→→	131072

33) ZLN ZERO SUBST 長 (Zero subst LeNgtH)

機能 測定パターンが ZERO SUBST の場合にゼロ挿入ビット長の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
ZLN	ZLN△m	ZLN?	ZLN△m (FIX 5)

m の値 ゼロ挿入ビット長の設定を下記の範囲内で設定します。

最大値 : ZERO SUBST 段数によって以下の範囲となります。

- 2⁷ : 127
- 2⁹ : 511
- 2¹¹ : 2047
- 2¹⁵ : 32767

最小値 : 1
ステップ : 1

コマンド種別 シーケンシャル・コマンド

使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合
測定パターンが ALTERNATE、DATA、PRBS の場合
同期方式が QUICK の場合

問い合わせ : 次の場合には無効となり、ERR (CR / LF) を出力します。
測定パターンが ALTERNATE、DATA、PRBS の場合
同期方式が QUICK の場合

使用例 プログラム : ZERO SUBST 長を1ビットに設定する場合
OUTPUT△700;"ZLN△1"

問い合わせ : ZERO SUBST 長が127ビットに設定されている場合
OUTPUT△700;"ZLN?"
ENTER△700;B\$
PRINT△B\$

↓

ZLN△△△127 (CR / LF) を出力します。

: 測定パターンが PRBS に設定されている場合
OUTPUT△700;"ZLN?"
ENTER△700;B\$
PRINT△B\$

↓

ERR (CR / LF) を出力します。

34) ADR
PAG

ページ数 (ADdRess / PAGe)

■ 機能

ページ数の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
ADR	ADR△m	ADR?	ADR△m (FIX 9)
PAG	PAG△m	PAG?	PAG△m (FIX 9)

■ mの値

ページ数の設定を下記の範囲内で設定します。

最大値 : 1342177278

最小値 : 1

ステップ : 1

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : ページを1ページに設定する場合

OUTPUT△700;"PAG△1"

問い合わせ : ページが16000ページに設定されている場合

OUTPUT△700;"PAG?"

ENTER△700;B\$

PRINT△B\$

↓

PAG△△△△△16000 (CR/LF) を出力します。

■ Note

ページ数の設定コマンドは、“ADR”と“PAG”の2種類ありますが、機能は同じです。

但し、設定できる最大ページ数は、設定されている測定パターンやデータ長の設定値によって異なります。

また、上記 m の最大値である1342177278を越えない範囲で、設定可能な最大ページ数を越えるページ数を入力すると、その時の最大ページ数に変更します。

例) データ長=32、表示しているページ数=1、最大ページ数は2ページですが、PAG△3を入力すると、表示しているページ数は2ページとなります。

ページの最大値は、データ長 ÷ 16を越えない値までで、余りがある場合は左記の商 + 1までとなります

35) BIT パターン・ビット (pattern BIT)

■ 機能 ビット・パターンの設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
BIT	<ul style="list-style-type: none">NR1形式 BIT△mHEX形式 BIT△#Hm	BIT?	<p>設定されているページ数から最大8ページ分までのビットの内容を、最大パターン設定ページまで、以下の形式で出力します。</p> <p>PAG△*****: BIT△#H****, #H****, ..., #H****</p> <p>ビット・パターンを出力先頭ページとともに HEX で表示する。 最大8ページ分 (各 FIX 4)</p>

■ m の値 ビット・パターンを下記の範囲内で設定します。

- | | |
|-------------|------------|
| ● NR1形式 | ● HEX形式 |
| 最大値 : 65535 | 最大値 : FFFF |
| 最小値 : 0 | 最小値 : 0 |
| ステップ : 1 | ステップ : 1 |

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合
測定パターンが ZERO SUBST、PRBS の場合
同期方式が QUICK の場合

問い合わせ : 次の場合には無効となり、ERR (CR/LF) を出力します。
同期方式が QUICK の場合

■ 使用例 プログラム : 現在設定されているページから3ページ分のビット・パターンを設定する場合

OUTPUT△700;"BIT△10,20,30"
OUTPUT△700;"BIT△#HFFFF, #H1000, #H2000"

データとデータの間をコンマ (,) で区切るにより連続したページのパターン・ビットを設定することができます

ページ数を設定し、そのページから4ページ分のパターン・ビットの設定をする場合は

OUTPUT 700;"PAG△10;BIT△10,20,30,40"
OUTPUT 700;"PAG△10;BIT△#HFFFF, #H1000, #H2000, #H3000"

問い合わせ : 表示ページが1、取りうる最大ページ数が10までとされている場合

1ページから8ページまでのデータを読みだします。

OUTPUT△700;"BIT?"

ENTER△700;B\$

PRINT△B\$

↓

PAG△△△△△△△△1; BIT△#H0000,#H0000,
#H0000,#H0000,#H0000,
#H0000,#H0000,#H0000

(CR/LF)を出力します。

: 同期方式が **QUICK** に設定されている場合

OUTPUT△700;"BIT?"

ENTER△700;B\$

PRINT△B\$

↓

ERR (CR/LF)を出力します。

■ Note

NR1または HEX 部のデータとデータの間をコンマ(,)で区切ることによって連続したページのパターン・ビットを設定することができます。(最大8ページ分)

ビット表示器の bit 1を LSB、bit 16を MSB として設定・レスポンスを行います。

例えば、32768を設定すると最上位ビット (bit 16) が1となります。

36) CHM

BIT WINDOW パターン・ビット
(bit window **CH Mask pattern**)

機能

BIT WINDOW パターンの設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
CHM	<div><div>● NR1形式</div><div>CHM△m</div><div>● HEX 形式</div><div>CHM△#Hm</div></div>	CHM?	<div>設定されているページ数から最大8ページ分までのビットの内容を、最大パターン設定ページまで、以下の形式で出力します。</div> <div><div>MSK△*;CHM△#H****, #H****</div></div> <div>ビット・パターンを先頭ページとともに HEX で表示</div> <div>最大2ページ分 (各 FIX 4)</div>

m の値

BIT WINDOW パターンを下記の範囲内で設定します。

- NR1形式

最大値 : 65535

最小値 : 0

ステップ : 1
- HEX 形式

最大値 : FFFF

最小値 : 0

ステップ : 1

コマンド種別

シーケンシャル・コマンド

使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合

アイ・マージン測定が実行中の場合

フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

使用例

プログラム : 現在設定されているページが1ページで2ページ分のビット・パターンを設定する場合

OUTPUT△700;"CHM△10,20"

OUTPUT△700;"CHM△#HFFFF, #H1000"

データとデータの間をコンマ(,)で区切ることで連続したページのパターン・ビットを設定することができます

ページ数を設定し、そのページから最大ページ分のパターン・ビットの設定をする場合は

OUTPUT 700;"MSK△1;CHM△10,20"

OUTPUT 700;"MSK△1;CHM△#HFFFF, #H1000"

問い合わせ : 表示ページが1と設定されている場合

1ページから2ページまでのデータを読みだします。

OUTPUT△700;"CHM?"

ENTER△700;B\$

PRINT△B\$

↓

MSK△1;CHM△#H0000,#H0000 (CR/LF) を出力します。

■ **Note**

NR1またはHEX部のデータとデータの間をコンマ(,)で区切ることによって連続したページのパターン・ビットを設定することができます。(最大2ページ分)

37) MSK BIT WINDOW ページ (bit window MaSK page)

機能 ページ数とパターン同期位置の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
MSK	MSK△m	MSK?	MSK△m (FIX 10)

m の値 ページ数の設定を下記の範囲内で設定します。

最大値 : 2
最小値 : 1
ステップ : 1

コマンド種別 シーケンシャル・コマンド

使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

使用例 プログラム : ページを1ページに設定する場合
OUTPUT△700;"MSK△1"

問い合わせ : ページが2ページに設定されている場合
OUTPUT△700;"MSK?"
ENTER△700;B\$
PRINT△B\$

↓
MSK△2 (CR/LF) を出力します。

38) MGB

BLOCK WINDOW パターン・ビット
(block window Mask Gate pattern Bit)

■ 機 能

BLOCK WINDOW ビット・パターンの設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
MGB	<ul style="list-style-type: none"> ● NR1形式 MGB△m ● HEX 形式 MGB△#Hm 	MGB?	<p>設定されているページ数から最大8ページ分までのビットの内容を、最大パターン設定ページまで、以下の形式で出力します。</p> <p>PAG△*****; MGB△#H****, #H****, ..., #H****</p> <p>ビット・パターンを先頭ページとともに HEX で表示 最大8ページ分 (各 FIX 4)</p>

■ m の値

ビット・パターンを下記の範囲内で設定します。

● NR1形式

最大値 : 65535

最小値 : 0

ステップ : 1

● HEX 形式

最大値 : FFFF

最小値 : 0

ステップ : 1

※ 1 ~ 65535 の場合は32ビット単位で測定区間が MASK されます。

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合
測定パターンが PRBS の場合
同期方式が QUICK の場合
データ長が32の倍数でない場合

問い合わせ : 次の場合には無効となり、ERR (CR/LF) を出力します。
測定パターンが PRBS の場合
同期方式が QUICK の場合
データ長が32の倍数でない場合

■ 使用例

プログラム : 現在設定されているページから3ページ分のビット・パターンを設定する場合

```
OUTPUT△700;"MGB△00,01,00"
```

```
OUTPUT△700;"MGB△#HFFFF,#H1000,#H2000"
```

データとデータの間をコンマ(,)で区切ることで連続したページのパターン・ビットを設定することができます

ページ数を設定し、そのページから4ページ分のパターン・ビットの設定をする場合は

```
OUTPUT 700;"PAG△10;MGB△10,20,30,40"
```

```
OUTPUT 700;"PAG△10;MGB△#HFFFF,#H1000,
#H2000,#H3000"
```

問い合わせ : 表示ページが1と設定されている場合

1ページから8ページまでのデータを読みだします。

```
OUTPUT△700;"MGB?"
```

```
ENTER△700;B$
```

```
PRINT△B$
```

↓

```
PAG△△△△△△△△1; MGB△#HFFFF,#HFFFF,#HFF
FF,
#HFFFF,#H0000,#H0000,
#H0000,#H0000,#H0000
```

(CR/LF)を出力します。

■ Note

NR1またはHEX部のデータとデータの間をコンマ(,)で区切ることによって連続したページのパターン・ビットを設定することができます。(最大8ページ分)

BLOCK WINDOW パターンは、32ビット単位で設定されます。

そのため、1ページと2ページ、3ページと4ページといった32ビットのデータはALL'0'かALL'1'のどちらかに設定されます。

例えば、設定値が1～65535の場合は32ビット分ALL'1'に設定され測定区間がMASKされます。

ALTERNATE パターンの場合は、A/B表示切替(No. 30)の状態によってAもしくはBの**BLOCK WINDOW**パターンの設定を行って下さい。

ページの設定は、(No. 34)と同一コマンドになります。

39) EAB?

誤り分析データ (Error Analysis pattern Bit ?)

■ 機能

誤り分析の結果を読みとります。(オプション-01を実装している場合のみ有効です。)

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
無し	無し	EAB?	<p>設定されているページ数の誤り分析結果の内容を以下の形式で出力します。</p> <pre> EAP△**;EAB△*****, #B***** ↑ ↑ ページ (FIX 2) モニター・データ (ページ値) (FIX 9) 誤り分析データ (FIX 16) 0 or 1 : No error + : 挿入エラー - : 消失エラー </pre>

■ m の値

ページ : 1～16までの表示切替

モニター・データ： 設定されたパターンの、先頭からのページ位置

誤り分析データ : 誤り分析結果

0もしくは1 : エラー無し

+記号 : 挿入エラー (0 → 1に変化したエラー)

一記号 : 送出エラー (1 → 0 に変化したエラー)

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

問い合わせ：次の場合には無効となり、ERR (CR/LF) を出力します。

同期方式が QUICK の場合

測定データが無い場合

測定データが無効の場合

測定パターンがAlternate の場合

■ 使用例

問い合わせ：表示ページが10、モニターのページ値が218ページ目の誤り分析結果が下記の場合

bit 1 bit 16

0110110100001001

↑ エラービット

OUTPUT△700:"EAB?"

ENTER $\Delta 700$; BS

PRINT△BS\$

EAP△10;EAB△△△△△△△218,#B10010000010110+10
(CR/LF)を出力します。

Note

誤り分析結果の取り込みは16ビット単位で行いますが、ページ設定を自動的にインクリメントしません。そのため、各ページのデータを取り込む場合はその都度誤り分析ページ設定を読みとりたいページ値に設定して下さい。

(ページ設定は次ページの **EAP** コマンドで行います。)

誤り分析データは内蔵メモリにバックアップされません。ご注意ください。

40) EAP 誤り分析ページ (Error Analysis Page)

機能 誤り分析のページ切替を行います。(オプション-01を実装している場合のみ有効です。)

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
EAP	EAP△m	EAP?	EAP△m (FIX 2)

mの値 誤り分析結果は16ビット単位でBIT表示部に表示されます。
そのページを1~16の範囲で設定します。

数値範囲： 最大値 16
最小値 1
ステップ 1

コマンド種別 シーケンシャル・コマンド

使用制限 次の設定条件の場合は、無効となります。

プログラム： 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合
誤り分析 OPTION-01が未実装の場合
同期方式が QUICK の場合

問い合わせ： 次の場合には無効となり、ERR (CR / LF) を出力します。
同期方式が QUICK の場合
次の場合には、無効となります
誤り分析 OPTION-01が未実装の場合

使用例

プログラム： 誤り分析表示ページを5ページに設定する場合
OUTPUT△700;"EAP△5"

問い合わせ： 誤り分析表示ページが1と設定されている場合
OUTPUT△700;"EAP?"
ENTER△700;B\$
PRINT△B\$
↓
EAP△△1 (CR / LF) を出力します。

41) MSE**BIT WINDOW ON / OFF (bit window ch MaSk Enable)****■ 機 能**

BIT WINDOW の ON / OFF 制御を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
MSE	MSE△m	MSE?	MSE△m (FIX 1)

■ m の値

BIT WINDOW の ON / OFF 制御を行います。

0 : OFF

1 : ON

■ コマンド種別 シーケンシャル・コマンド**■ 使用制限**

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : BIT WINDOW を ON に設定する場合“
OUTPUT△700;"MSE△1"

問い合わせ : BIT WINDOW が OFF に設定されている場合
OUTPUT△700;"MSE?"
ENTER△700;B\$
PRINT△B\$

↓

MSE△0 (CR / LF) を出力します。

42) MGE

BLOCK WINDOW ON / OFF
(block window M eas. G ate E nable)

機能

BLOCK WINDOWのON / OFF制御を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
MGE	MGE△m	MGE?	MGE△m (FIX 1)

mの値

BLOCK WINDOW の ON / OFF 制御を行います。

- 0 : OFF
- 1 : ON

コマンド種別

シーケンシャル・コマンド

使用制限

次の設定条件の場合は、無効となります。

- プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合
測定パターンが PRBS の場合
測定パターンが ALTERNATE, DATA の場合で、かつ
データ長が32の倍数で無い場合
同期方式が QUICK の場合
- 問い合わせ : 次の場合は無効となり、ERR (CR / LF) を出力します。
測定パターンが PRBS の場合
測定パターンが ALTERNATE, DATA の場合で、かつ
データ長が32の倍数で無い場合
同期方式が QUICK の場合

使用例

プログラム : BLOCK WINDOW を ON に設定する場合
OUTPUT△700;"MGE△1"

問い合わせ : BLOCK WINDOW が OFF に設定されている場合
OUTPUT△700;"MGE?"
ENTER△700;B\$
PRINT△B\$
↓
MGE△0 (CR / LF) を出力します。

43) EAT**誤り分析トリガ (Error Analysis Trigger)**

■ 機 能

誤り分析のトリガ設定を行います。(オプション-01を実装している場合のみ有効です。)

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
EAT	EAT△m	EAT?	EAT△m (FIX 1)

■ mの値

誤り分析のトリガ状態の設定を行います。

0 : OFF (強制終了)

1 : プログラム時

START

問い合わせ時

AWITTING (トリガ待ち状態)

2 : プログラム時

無効

問い合わせ時

TRIGGERD (トリガがかかった状態)

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合
 同期方式が **QUICK** の場合
OPTION-01を実装していない場合
 測定パターンが **Alternate** の場合

問い合わせ : 次の場合は無効となり、**ERR (CR/LF)**を出力します。
 同期方式が **QUICK** の場合
 次の場合は無効となります。
OPTION-01を実装していない場合

■ 使用例

プログラム : 誤り分析トリガ **START**に設定する場合
OUTPUT△700;"EAT△1"

問い合わせ : 誤り分析トリガがかかった状態の場合
OUTPUT△700;"EAT?"
ENTER△700;B\$
PRINT△B\$

↓

EAT△2 (CR/LF)を出力します。

44) WRT**パターン・データ入力バイト数 (pattern data WRiTe)****機能**

パターンデータの DMA 転送を行うバイト数とスタートアドレスを設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
WRT	WRT△m1,m2	無 し	無 し

m の値

m1 : パターン転送バイト数
 数値範囲 : 最大値 1048376
 最小値 1
 ステップ 1

m2 : パターン入力先頭アドレス
 数値範囲 : 最大値 524287
 最小値 0
 ステップ 1

測定パターンが **ALTERNATE** パターンの場合は、上記最大値の半分となります。

コマンド種別 シーケンシャル・コマンド**使用制限** 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合
 同期方式が **QUICK** の場合
 測定パターンが **ZERO SUBST** もしくは **PRBS** の場合

使用例 プログラム : 測定パターンが **DATA** の場合で1ページから10ページまでのデータを設定する場合

```
DIM△B(9)
READ△B(*)
DATA△1,2,4,8,16,32,64,128,256,512
OUTPUT△700;"WRT△20,0"
OUTPUT△700△USING△"W";B(*)
```

1～10ページのデータを設定します。

Note

本器は **NR** 部の各値から、DMA 転送するパターン・データの必要なバイト数及び入力先頭アドレスを定義し、DMA モードの切替、内部 **RAM** エリアへの格納アドレスを定義します。

パターン先頭アドレスと設定される実際のページとの関係は、

(パターン先頭アドレス + 1) = 実際のページ数

となります。

なお、パターン・データの転送終了後は **DMA** モードを解除します。

パターン・データの DMA 転送については、付録のパターン・データの DMA 転送を参照して下さい。

また、データ長と先頭アドレスによって設定可能なバイト数が決まりますが、設定可能な範囲を越えるデータが転送された場合は、範囲を越えたデータは無効となります。

45) RED? パターン・データ出力バイト数 (pattern data REaD ?)

■ 機能

パターンデータの DMA 転送を使用した読みとりを行うバイト数とスタートアドレスを設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
RED	無し	RED?△m1,m2	データパターン列 (m1による)

■ m の値

m1 : パターン転送バイト数

数値範囲 : 最大値 1048376
 最小値 1
 ステップ 1

m2 : パターン出力先頭アドレス

数値範囲 : 最大値 524287
 最小値 0
 ステップ 1

測定パターンが **ALTERNATE** パターンの場合は、上記最大値の半分となります。

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となり ERR (CR / LF) を出力します。

問い合わせ : 同期方式が **QUICK** の場合
 測定パターンが **ZERO SUBST** もしくは **PRBS** の場合

■ 使用例 問い合わせ : 測定パターンが DATA の場合で1ページから10ページまでのデータを設定する場合

```
DIM△B(9)
OUTPUT△700;"RED?△20,0"
ENTER△700△USING△"W";B(*)
PRINT△B(*)
```

1～10ページのデータを印字します。

■ Note

本器は NR 部の各値から、DMA 転送するパターン・データの必要なバイト数及び入力先頭アドレスを定義し、DMA モードの切替、内部 RAM エリアへの格納アドレスを定義します。

パターン先頭アドレスと設定される実際のページとの関係は、

(パターン先頭アドレス + 1) = 実際のページ数

となります。

なお、パターン・データの転送終了後は DMA モードを解除します。

パターン・データの DMA 転送については、付録のパターン・データの DMA 転送を参照して下さい。

また、データ長と先頭アドレスによって出力するバイト数が決まりますが、有効なデータ数を越える問い合わせを行なった場合は、有効なバイト数のみを出力します。

46) MWT BLOCK WINDOW・データ入力バイト数
(block window M meas. gate pattern data WriTe)

■ 機 能 BLOCK WINDOW パターンデータの DMA 転送を行うバイト数とスタートアドレスを設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
MWT	MWT△m1,m2	無 し	無 し

■ m の値 m1 : BLOCK WINDOW パターン転送バイト数 (32ページ分のデータ)
 数値範囲 : 最大値 32768
 最小値 1
 ステップ 1

 m2 : BLOCK WINDOW パターン入力先頭アドレス
 数値範囲 : 最大値 16383
 最小値 0
 ステップ 1

測定パターンが **ALTERNATE** パターンの場合は、上記最大値の半分となります。

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合
 同期方式が **QUICK** の場合
 測定パターンが **PRBS** の場合
 測定パターンが **ALTERNATE** もしくは **DATA** の場合でかつ
 データ長が32の倍数で無い場合

■ 使用例 プログラム : 測定パターンが **DATA** の場合で1ページから32ページまでのデータを設定する場合

DIM△B(0)
READ△B(*)
DATA△1
OUTPUT△700;"MWT△2,0"
OUTPUT△700△USING△"W";B(*)

1～32ページのデータを設定します。(17, 18ページを **MASK** します。)

■ Note

本器は **NR** 部の各値から、**DMA** 転送するパターン・データの必要なバイト数及び入力先頭アドレスを定義し、**DMA** モードの切替、内部 **RAM** エリアへの格納アドレスを定義します。

パターン先頭アドレスと設定される実際のページとの関係は、

$$(\text{パターン先頭アドレス} \times 32 + 1) = \text{実際のページ数}$$

となります。

なお、パターン・データの転送終了後は **DMA** モードを解除します。

パターン・データの **DMA** 転送については、付録のパターン・データの **DMA** 転送を参照して下さい。

BLOCK WINDOW は、**32**ビット単位でのデータとなるため **DMA** 転送する際は**32**ビット単位を**1**ビットとして扱います。

各ビットの設定値が**0**の場合は測定対象ブロックとなり、**1**の場合は測定をマスクするようになります。

設定値には下記のような重み付けがされています。

設定値	BLOCK WINDOW 設定ページ (先頭アドレスを 0 にした場合)
1	17, 18
2	19, 20
4	21, 22
8	23, 24
16	25, 26
32	27, 28
64	29, 30
128	31, 32
256	1, 2
512	3, 4
1024	5, 6
2048	7, 8
4096	9, 10
8192	11, 12
16384	13, 14
32768	15, 16

先頭アドレスを **N** にすると、上記ページ値に **32 × N** を加えたページに設定されます。

ALTERNATE パターンの場合は **ALTERNATE** 表示切替の状態によって **A** もしくは **B** のパターンが設定されます。

また、データ長の先頭アドレスによって設定可能なバイト数が決まりますが、設定可能な範囲を越えるデータが転送された場合は範囲を越えた分のデータは無効となります。

47) MRD? BLOCK WINDOW・データ出力バイト数
(block window Meas. gate pattern data ReaD?)

■ 機能 BLOCK WINDOW パターンデータの DMA 転送を使用した読みとりを行う
バイト数とスタートアドレスを設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
MRD	無し	MRD?△m1,m2	データパターン列 (m1による)

■ m の値 m1 : BLOCK WINDOW パターン転送バイト数
数値範囲 : 最大値 32768
最小値 1
ステップ 1

m2 : BLOCK WINDOW パターン出力先頭アドレス
数値範囲 : 最大値 16383
最小値 0
ステップ 1

測定パターンが ALTERNATE パターンの場合は、上記最大値の半分となります。

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

問い合わせ : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合
同期方式が QUICK の場合
測定パターンが PRBS の場合
測定パターンが ALTERNATE もしくは DATA の場合でかつ
データ長が32の倍数で無い場合

■ 使用例 問い合わせ : 測定パターンが DATA の場合で1ページから32ページまでの
データをよみとる場合
DIM△B(0)
OUTPUT△700;"MRD?△2,0"
ENTER△700△USING△"W";B(*)
PRINT△B(*)
1～32ページのデータを印字します。

■ Note

本器は NR 部の各値から、DMA 転送するパターン・データの必要なバイト数及び入力先頭アドレスを定義し、DMA モードの切替、内部 RAM エリアへの格納アドレスを定義します。

パターン先頭アドレスと設定される実際のページとの関係は、

$$(\text{パターン先頭アドレス} + 1) = \text{実際のページ数}$$

となります。

なお、パターン・データの転送終了後は DMA モードを解除します。

パターン・データの DMA 転送については、付録のパターン・データの DMA 転送を参照して下さい。

BLOCK WINDOW は、32ビット単位でのデータとなるため DMA 転送する 32 ビット単位のデータが 0 以外の場合は、全ての 32 ビットデータは 1 となります。

各ビットの設定値が 0 の場合は測定対象ブロックとなり、1 の場合は測定をマスクするようになります。

設定値には下記のような重み付けがされています。

設定値	BLOCK WINDOW 設定ページ (先頭アドレスを 0 にした場合)
1	17, 18
2	19, 20
4	21, 22
8	23, 24
16	25, 26
32	27, 28
64	29, 30
128	31, 32
256	1, 2
512	3, 4
1024	5, 6
2048	7, 8
4096	9, 10
8192	11, 12
16384	13, 14
32768	15, 16

先頭アドレスを N にすると、上記ページ値に $32 \times N$ を加えたページに設定されます。

ALTERNATE パターンの場合は **ALTERNATE** 表示切替の状態によって A もしくは B のパターンが読み出されます。

また、データ長と先頭アドレスによって出力するバイト数が決まりますが、有効なデータ数を越える問い合わせを行なった場合には、有効なバイト数のみを出力します。

48) ALL パターン・データ・プリセット (全ページ・全ビット) (**prest ALL**)

■ 機 能 パターンデータ全ページの全ビットを0または1に設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
ALL	ALL△m1	無 し	無 し

■ mの値 m1 :
 0 : 全ページ、クリア
 1 : 全ページ、セット

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

 プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合
 同期方式が **QUICK** の場合
 測定パターンが **ZERO SUBST** もしくは **PRBS** の場合

■ 使用例 プログラム : 測定パターンが **DATA** の場合で全ページをクリアする場合

 OUTPUT△700;"ALL△0"

 全ページのデータをクリアします。

■ Note 測定パターンが **ALTERNATE** パターンの場合は **A/B** 表示切替 (No. 30) の状態によって **A** もしくは **B** のパターンについてプリセットを行います。

 例えば、**A** パターンを表示している場合は、本コマンドを実行すると **A** のパターンのみプリセットされます。

49) PST パターン・データ・プリセット (1ページ・全ビット) (PreST)**■ 機 能**

パターンデータ1ページの全ビットを0または1に設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
PST	PST△m1	無 し	無 し

■ m の値

m1 :

0 : 1ページ、クリア

1 : 1ページ、セット

■ コマンド種別 シーケンシャル・コマンド**■ 使用制限**

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合
 同期方式が **QUICK** の場合
 測定パターンが **ZERO SUBST** もしくは **PRBS** の場合

■ 使用例

プログラム : 測定パターンが **DATA** の場合で1ページをクリアする場合

OUTPUT△700;"PST△0"

設定されているページの1ページ分のデータをクリアします。

■ Note

測定パターンが **ALTERNATE** パターンの場合は **A/B** 表示切替 (No. 30) の状態によって **A** もしくは **B** のパターンについてプリセットを行います。

例えば、**A** パターンを表示している場合は、本コマンドを実行すると **A** のパターンのみプリセットされます。

50) MAL

BLOCK WINDOW・データ・プリセット (全ページ・全ビット)
(block window Meas. gate pattern preset ALL)

■ 機能

BLOCK WINDOW パターンデータ全ページの全ビットを0または1に設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
MAL	MAL△m1	無 し	無 し

■ m の値

m1 :

0 : 全ページ、クリア

1 : 全ページ、セット

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合
同期方式が **QUICK** の場合
測定パターンが **PRBS** の場合
測定パターンが **ALTERNATE** もしくは **DATA** の場合でかつ
データ長が32の倍数で無い場合

■ 使用例

プログラム : 測定パターンが **DATA** の場合で **BLOCK WINDOW** データの
全ページをクリアする場合

```
OUTPUT△700:"MAL△0"
```

全ページのデータをクリアします。

Note

測定パターンが **ALTERNATE** パターンの場合は **A/B 表示切替 (No. 30)** の状態によって **A** もしくは **B** のパターンについてプリセットを行います。

例えば、A パターンを表示している場合は、本コマンドを実行すると A のパターンのみプリセットされます。

51) MPS**BLOCK WINDOW・データ・プリセット (1ページ・全ビット)
(block window M meas. gate pattern PreSet)**

■ 機 能

BLOCK WINDOW パターンデータ1ページの全ビットを0または1に設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
MPS	MPS△m1	無 し	無 し

■ m の値

m1 :

0 : 1ページ、クリア

1 : 1ページ、セット

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合

アイ・マージン測定が実行中の場合

フロッピー・ディスクをアクセス中の場合

同期方式が **QUICK** の場合

測定パターンが **PRBS** の場合

測定パターンが **ALTERNATE** もしくは **DATA** の場合でかつ

データ長が32の倍数で無い場合

■ 使用例

プログラム : 測定パターンが **DATA** の場合で **BLOCK WINDOW** データの1ページをクリアする場合

OUTPUT△700;"MPS△0"

設定されているページの1ページ分のデータをクリアします。

■ Note

測定パターンが **ALTERNATE** パターンの場合は **A/B** 表示切替 (No. 30) の状態によって **A** もしくは **B** のパターンについてプリセットを行います。

例えば、**A** パターンを表示している場合は、本コマンドを実行すると **A** のパターンのみプリセットされます。

52) HAL

BIT WINDOW・データ・プリセット (全ページ・全ビット)
(bit window ch mask pattern preset ALL)

機能

BIT WINDOW パターンデータ全ページの全ビットを0または1に設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
HAL	HAL△m1	無 し	無 し

mの値

m1 :
0 : 全ページ、クリア
1 : 全ページ、セット

コマンド種別

シーケンシャル・コマンド

使用制限

次の設定条件の場合は、無効となります。

プログラム :

自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

使用例

プログラム : 測定パターンが DATA の場合で BIT WINDOW データの
全ページをクリアする場合

OUTPUT△700;"HAL△0"

全ページのデータをクリアします。

53) HPS**BIT WINDOW・データ・プリセット (1ページ・全ビット)
(bit window cH mask pattern PreSet)****■ 機 能**

BIT WINDOW パターンデータ1ページの全ビットを0または1に設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
HPS	HPS△m1	無 し	無 し

■ m の値

m1 :

0 : 1ページ、クリア

1 : 1ページ、セット

■ コマンド種別 シーケンシャル・コマンド**■ 使用制限**

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

■ 使用例

プログラム : 測定パターンが DATA の場合で BIT WINDOW データの1ページをクリアする場合

OUTPUT△700;"HPS△0"

設定されているページの1ページ分のデータをクリアします。

54) PSP パターン同期トリガ位置 (Pattern Sync Position)

■ 機能 パターン同期トリガが **Variable** の場合にトリガ位置の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
PSP	PSP△m	PSP?	PSP△m (FIX 9)

■ mの値 パターン同期位置の設定を下記の範囲内で設定します。

最大値 : 134217728

最小値 : 1

ステップ : 1

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例
プログラム : パターン同期トリガ位置を1ページ目に設定する場合
OUTPUT△700;"PSP△1"

問い合わせ : パターン同期トリガ位置が16000ページ目に設定されている場合

OUTPUT△700;"PSP?"
ENTER△700;B\$
PRINT△B\$

↓

PSP△△△△△16000 (CR/LF) を出力します。

■ Note 設定できるパターン同期トリガ位置の最大ページ数は、設定されている測定パターンやデータ長の設定値によって異なります。

また、上記 m の最大値である134217728を越えない範囲で、設定可能な最大ページ数を越えるページ数を入力すると、その時の最大ページ数に変更します。

例) データ長=32、トリガ位置=1、トリガ位置最大数は2ですが、**PSP△3**を入力すると、トリガ位置は2となります。

トリガ位置の最大値は、データ長 ÷ 16を越えない値までで、余りがある場合は左記の商 + 1までとなります

55) PPD**ページ・パターン同期トリガ位置表示切替
(Page / Pattern sync positon Display)****機 能**

7セグ表示器の表示内容を、ページとパターン同期トリガ位置に切り替えます。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
PPD	PPD△m	PPD?	PPD△m (FIX 1)

mの値

0 : ページ数を表示します。

1 : パターン同期トリガ位置を表示します。

コマンド種別

シーケンシャル・コマンド

使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

使用例

プログラム : ページ数を表示する場合
OUTPUT△700;"PPD△0"

問い合わせ : パターン同期トリガ位置が表示されている場合
OUTPUT△700;"PPD?"
ENTER△700;B\$
PRINT△B\$

↓

PPD△1 (CR / LF) を出力します。

● MEASUREMENT セクション

次のページより、**MEASUREMENT** セクションの各コントロール・メッセージについて示します。なお、文中の△はスペースを意味します。

56) CLI? クロック断状態 (Clock Loss Intervals ?)

■ 機能 クロックの入力状態を読みとります。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
CL1	無し	CL1?	CL1△m (FIX 1)

■ mの値
 0 : クロック断でない状態
 1 : クロック断状態

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

問い合わせ : 無し

■ 使用例 問い合わせ : クロック断でない状態の場合

OUTPUT△700;"CLI?"

ENTER△700;B\$

PRINT△B\$

↓

CLI△0 (CR/LF) を出力します。

57) SLI? 同期外れ状態 (Sync Loss Intervals ?)

■ 機 能 同期状態を読みとります。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
SLI	無 し	SLI?	SLI△m (FIX 1)

■ mの値 0 : 同期外れでない状態
 1 : 同期外れ状態

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

問い合わせ : 無し

■ 使用例 問い合わせ : 同期外れでない状態の場合

OUTPUT△700;"SLI?"

ENTER△700;B\$

PRINT△B\$

↓

SLI△0 (CR/LF) を出力します。

58) ERS? 誤り検出状態 (ERrorS ?)

■ 機 能 誤りの検出状態を読みとります。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
ERS	無 し	ERS?	ERS△m (FIX 1)

■ mの値 0： 誤りを検出しなかった状態
 1： 誤りを検出した状態

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。
問い合わせ： 無し

■ 使用例 問い合わせ： 同期外れでない状態の場合
 OUTPUT△700;"ERS?"
 ENTER△700;B\$
 PRINT△B\$
 ↓
 ERS△0 (CR/LF) を出力します。

59) DMS

測定表示モード (Display or Measurement)

■ 機能

測定表示器に表示するデータの選択を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
DMS	DMS△m	DMS?	DMS△m (FIX 1)

■ m の値

- 0 : ERROR RATIO
- 1 : ERROR COUNT
- 2 : ERROR INTERVAL
- 3 : ERROR FREE INTERVAL
- 4 : CLOCK FREQUENCY

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : 表示モードを ERROR RATIO に設定する場合
 OUTPUT△700;"DMS△0"

問い合わせ : 表示モードが ERROR COUNT に設定されている場合
 OUTPUT△700;"DMS?"
 ENTER△700;B\$
 PRINT△B\$

↓

DMS△1 (CR/LF) を出力します。

60) CUR**測定途中結果表示機能 (CURrent data)****■ 機 能**

測定途中結果を表示する機能を制御します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
CUR	CUR△m	CUR?	CUR△m (FIX 1)

■ mの値

0 : OFF

1 : ON

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : 測定途中結果表示機能を **OFF** に設定する場合
OUTPUT△700;"CUR△0"

問い合わせ : 測定途中結果表示機能が **ON** に設定されている場合
OUTPUT△700;"CUR?"
ENTER△700;B\$
PRINT△B\$

↓

CUR△1 (CR/LF) を出力します。

61) MOD 測定モード (measurement MODe)

■ 機 能 測定モードの設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
MOD	MOD△m	MOD?	MOD△m (FIX 1)

■ m の値 0 : REPEAT
 1 : SINGLE
 2 : UNTIMED

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例 プログラム : 測定モードを REPEAT に設定する場合
 OUTPUT△700;"MOD△0"

問い合わせ : 測定モードが SINGLE に設定されている場合
 OUTPUT△700;"MOD?"
 ENTER△700;B\$
 PRINT△B\$

↓

MOD△1 (CR/LF) を出力します。

62) STA**測定開始 (STArt)****機 能**

測定開始の設定を行います。

測定動作中は、測定再スタート指示となります。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
STA	STA	無 し	無 し

コマン ド種別 シーケンシャル・コマンド**使用制限**

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッショルド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

使用例

プログラム : 測定開始または測定再スタート指示を行う場合
OUTPUT△700;"STA"

63) STO 測定終了 (STOp)

■ 機 能 測定終了または、測定中止の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
STO	STO	無 し	無 し

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッショルド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

■ 使用例 プログラム : 測定終了または測定中止を行う場合
 OUTPUT△700;"STO"

64) MSR?

測定状態 (MeaSuRement in progress or stop ?)

■ 機 能

測定状態 (測定中または、測定停止中) の読みとりを行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
MSR	無 し	MSR?	MSR△m (FIX 1)

- mの値
- 0： 測定停止中
1： 測定中
- コマンド種別
- シーケンシャル・コマンド
- 使用制限
- 次の設定条件の場合は、無効となります。
問い合わせ： 無し
- 使用例
- 問い合わせ： 測定停止中の場合
OUTPUT△700;"MSR?"
ENTER△700;B\$
PRINT△B\$
↓
MSR△0 (CR/LF) を出力します。

65) SYN 自動同期機能 (auto SYNC)

■ 機 能 自動同期機能の制御を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
SYN	SYN△m	SYN?	SYN△m (FIX 1)

■ m の値 0 : OFF
1 : ON

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例 プログラム : 自動同期機能を OFF に設定する場合
OUTPUT△700;"SYN△0"

問い合わせ : 自動同期機能が ON に設定されている場合
OUTPUT△700;"SYN?"
ENTER△700;B\$
PRINT△B\$

↓

SYN△1 (CR/LF) を出力します。

66) SYE**自動同期機能スレッシュールド (auto SYNC threshold)****機能**

自動同期機能のスレッシュールド設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
SYE	SYE△m	SYE?	SYE△m (FIX 1)

mの値

0 : 1E-2
 1 : 1E-3
 2 : 1E-4
 3 : 1E-5
 4 : 1E-6
 5 : 1E-7
 6 : 1E-8
 8 : INT

コマンド種別 シーケンシャル・コマンド**使用制限**

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

使用例

プログラム : 自動同期機能スレッシュールドを1E-2に設定する場合
OUTPUT△700;"SYE△0"

問い合わせ : 自動同期機能スレッシュールドが1E-3に設定されている場合
OUTPUT△700;"SYE?"
ENTER△700;B\$
PRINT△B\$

↓

SYE△1 (CR/LF) を出力します。**Note**

自動同期スレッシュールドの設定値は、別冊の機能・操作取扱説明書を参照して下さい。

67) TIM

リアル・タイム、測定時間表示切替
(real TIME or measurement time)

機能

時間表示器の表示内容を切替を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
TIM	TIM△m	TIM?	TIM△m (FIX 1)

- mの値
- 0

: 内蔵タイマの Y. M. D
- 1

: 内蔵タイマの H. M. S
- 2

: PERIOD
- 3

: TIMED
- 4

: ELAPSED

コマンド種別

シーケンシャル・コマンド

使用制限

次の設定条件の場合は、無効となります。

プログラム

自動位相スレッシュホールド・サーチ機能が実行中の場合

アイ・マージン測定が実行中の場合

フロッピー・ディスクをアクセス中の場合

問い合わせ

無し

使用例

プログラム

表示項目を Y. M. D (内蔵タイマの年, 月, 日) 表示に設定する場合

OUTPUT△700;"TIM△0"

問い合わせ

表示項目が H. M. S (内蔵タイマの時, 分, 秒) に設定されている場合

OUTPUT△700;"TIM?"

ENTER△700;B\$

PRINT△B\$

↓

TIM△1 (CR/LF) を出力します。

68) RTM**内部タイマ設定 (Real TiMe setting)****機能**

内部時計の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
RTM	RTM△m1,m2,m3,m4,m5,m6	RTM?	RTM△m1,m2,m3,m4,m5,m6 (各FIX 2)

m の値

m1 : 年 0 ~ 99
m2 : 月 1 ~ 12
m3 : 日 1 ~ 31 (日の最大値は月によって変わります)
m4 : 時 0 ~ 23
m5 : 分 0 ~ 59
m6 : 秒 0 ~ 59

コマンド種別 シーケンシャル・コマンド**使用制限**

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

使用例

プログラム : 内部時計を '95年1月27日13時59分59秒に設定する場合
OUTPUT△700;"RTM△95,01,27,13,59,59"

問い合わせ : 内部時計が '95年1月27日14時0分0秒に設定されている場合
OUTPUT△700;"RTM?"
ENTER△700;B\$
PRINT△B\$

↓

RTM△95,01,27,14,00,00 (CR/LF) を出力します。

Note

プログラムで設定するm1、m2等のNR1部分は省略できません。

69) PRD 測定時間設定 (measurement PeRioD)

機能 測定時間の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
PRD	PRD△m1,m2,m3,m4	PRD?	PRD△m1,m2,m3,m4 (各FIX 2)

m の値

m1 : 日 0 ~ 99
m2 : 時間 0 ~ 23
m3 : 分 0 ~ 59
m4 : 秒 0 ~ 59

但し、すべて00には設定できません。

コマンド種別 シーケンシャル・コマンド

使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合
測定モードが **UNTIME** の場合

問い合わせ : 無し

使用例

プログラム : 測定時間を99日13時間59分59秒に設定する場合
OUTPUT△700;"PRD△99,13,59,59"

問い合わせ : 測定時間が99日14時間0分0秒に設定されている場合
OUTPUT△700;"PRD?"
ENTER△700;B\$
PRINT△B\$

↓
PRD△99,14,00,00 (CR/LF) を出力します。

Note プログラムで設定する m1、m2等の NR1部分は省略できません。

70) ER?

誤り率測定結果 (Error ratio ?)

■ 機能

誤り率測定結果を、出力形式に従って出力します。

但し、ここで出力する誤り率は、7セグメント表示器に表示している値(終了データまたは、カレント・データ)を出力します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
ER	無し	ER?	ER△△*.****E-*** (FIX 10) ER△△*.****E-* (FIX 9) *は誤り率を示し、小数点以下4桁、指数部付きとします。指数部は2桁もしくは1桁となります。

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

問い合わせ : 誤り率表示器が‘-’表示の場合は下記の値を出力します。

ER△△0.0000E-00

■ 使用例

問い合わせ : 誤り率測定結果が 1.05×10^{-6} の場合

OUTPUT△700;"ER?"

ENTER△700;B\$

PRINT△B\$

↓

ER△△1.0500E-6 (CR/LF) を出力します。

: 誤り率測定結果が 1.05×10^{-10} の場合

OUTPUT△700;"ER?"

ENTER△700;B\$

PRINT△B\$

↓

ER△△1.0500E-10 (CR/LF) を出力します。

: 誤り率測定結果が“-”の場合

OUTPUT△700;"ER?"

ENTER△700;B\$

PRINT△B\$

↓

ER△△0.0000E-00 (CR/LF) を出力します。

71) EC? 誤り 個数測定結果 (Error Count ?)

■ 機 能 誤り 個数測定結果を、出力形式に従って出力します。
但し、ここで出力する誤り 個数は、7セグメント表示器に表示している値 (終了データまたは、カレント・データ) を出力します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
EC	無 し	EC?	<div>● 誤り 個数が1E + 8未満の場合 (FIX 10)</div> <div>EC△△△*****</div> <div>*は、誤り 個数を示し8桁の固定長とします。</div> <div>● 誤り 個数が1E+8以上の場合</div> <div>EC△△*.****E**</div> <div>*は誤り 個数を示し、小数点以下4桁指数部付きの固定長とします。</div>

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

問い合わせ : 誤り率表示器が‘-’表示の場合は下記の値を出力します。
EC△△1.0000E-99 (文字数11)

■ 使用例 問い合わせ : 誤り 個数測定結果が1.05 × 10⁶の場合
OUTPUT△700;"EC?"
ENTER△700;B\$
PRINT△B\$
↓
EC△△△△1050000 (CR/LF) を出力します。
: 誤り 個数測定結果が1.05 × 10⁹の場合
OUTPUT△700;"EC?"
ENTER△700;B\$
PRINT△B\$
↓
EC△△1.0500E09 (CR/LF) を出力します。
: 誤り率測定結果が“-”の場合
OUTPUT△700;"EC?"
ENTER△700;B\$
PRINT△B\$
↓
EC△△1.0000E-99 (CR/LF) を出力します。

72) CC?

クロック個数測定結果 (Clock Count ?)

■ 機能

クロック個数測定結果を、出力形式に従って出力します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
CC	無し	CC?	<ul style="list-style-type: none"> クロック個数が$1E+8$未満の場合 (FIX 10) CC△△△***** *は、クロック個数を示し8桁の固定長とします。 クロック個数が$1E+8$以上の場合 CC△△*.*****E** *はクロック個数を示し、小数点以下4桁指数部付きの固定長とします。

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

問い合わせ : 測定データの存在しない場合は下記の値を出力します。

CC△△1.0000E-99 (文字数11)

■ 使用例

問い合わせ : クロック個数測定結果が 1.05×10^6 の場合

```
OUTPUT△700;"CC?"
ENTER△700;B$
PRINT△B$
```

↓

CC△△△△1050000 (CR/LF) を出力します。

: クロック個数測定結果が 1.05×10^9 の場合

```
OUTPUT△700;"CC?"
ENTER△700;B$
PRINT△B$
```

↓

CC△△1.0500E09 (CR/LF) を出力します。

: クロック率測定結果が“-”の場合

```
OUTPUT△700;"CC?"
ENTER△700;B$
PRINT△B$
```

↓

CC△△1.0000E-99 (CR/LF) を出力します。

73) EI? エラー・インターバル個数測定結果 (Error Interval ?)

- 機能
- エラー・インターバル個数測定結果を、出力形式に従って出力します。
但し、ここで出力するエラー・インターバル個数は、7セグメント表示器に表示している値 (終了データまたは、カレント・データ) を出力します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
EI	無し	EI?	<div>● EI 個数が1E + 8未満の場合 (FIX 10) EI△△△***** *は、EI 個数を示し8桁の固定長とします。</div> <div>● EI 個数が1E + 8以上の場合 EI△△*.****E** *は EI 個数を示し、小数点以下4桁指数部付きの固定長とします。</div>

- コマンド種別 シーケンシャル・コマンド
- 使用制限 次の設定条件の場合は、無効となります。

- 使用例
- 問い合わせ : EI 個数測定結果が 1.05×10^6 の場合
OUTPUT△700;"EI?"
ENTER△700;B\$
PRINT△B\$
↓
EI△△△1050000 (CR/LF) を出力します。
- : EI 個数測定結果が 1.05×10^9 の場合
OUTPUT△700;"EI?"
ENTER△700;B\$
PRINT△B\$
↓
EI△△1.0500E09 (CR/LF) を出力します。
- : 誤り率測定結果が“－”の場合
OUTPUT△700;"EI?"
ENTER△700;B\$
PRINT△B\$
↓
EI△△1.0000E-99 (CR/LF) を出力します。

74) EFI?

エラー・フリー・インターバル率測定結果
(Error Free Interval ?)

- 機能
- エラー・フリー・インターバル率測定結果を、出力形式に従って出力します。
但し、ここで出力するエラー・インターバル個数は、7セグメント表示器に表示している値 (終了データまたは、カレント・データ) を出力します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
EFI	無し	EFI?	EFI△△△***.*** (FIX 10) *は、%EFIを示し、小数点以下4桁の固定長

- コマンド種別 シーケンシャル・コマンド
- 使用制限 次の設定条件の場合は、無効となります。
問い合わせ : 誤り率表示器が‘-’表示の場合は下記の値を出力します。
EFI△△△999.9999 (CR/LF)
- 使用例 問い合わせ : EFI率測定結果が99.01%の場合
OUTPUT△700;"EFI?"
ENTER△700;B\$
PRINT△B\$
↓
EFI△△△△99.0100 (CR/LF) を出力します。
: EFI率測定結果が‘-’の場合
OUTPUT△700;"EFI?"
ENTER△700;B\$
PRINT△B\$
↓
EFI△△△999.9999 (CR/LF) を出力します。

75) FRQ? クロック周波数測定結果 (clock FReQuency ?)

- 機能
- クロック周波数測定結果を、出力形式に従って出力します。
但し、ここで出力するエラー・インターバル個数は、7セグメント表示器に表示している値 (終了データまたは、カレント・データ) を出力します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
FRQ	無し	FRQ?	FRQ△△*****.*** (FIX 10) *は、クロック周波数を示し、小数点以下3桁の固定長 (単位MHz)

- コマンド種別
- シーケンシャル・コマンド

- 使用制限
- 次の設定条件の場合は、無効となります。

問い合わせ : クロック断が発生している場合は下記の値を出力します。
FRQ△△△△△0.000 (CR / LF)

- 使用例
- 問い合わせ : クロック周波数が50 MHz の場合
OUTPUT△700;"FRQ?"
ENTER△700;B\$
PRINT△B\$
↓
FRQ△△△△50.000 (CR / LF) を出力します。
:
クロック断が発生している場合
OUTPUT△700;"FRQ?"
ENTER△700;B\$
PRINT△B\$
↓
FRQ△△△△0.000 (CR / LF) を出力します。

76) OSD? 1秒データ測定結果 (One-Second Data ?)

機能 1秒測定結果を以下の出力形式に従って出力します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
OSD	無し	OSD?	<div>● 1秒間誤り個数が1E + 8未満の場合</div> <div>OSD△*.****E-**, 1秒間平均誤り率 △***** 1秒間誤り個数</div> <div>● 1秒間誤り個数が1E + 8以上の場合</div> <div>OSD△*.****E-**, *.****E** (FIX 19 or 20)</div>

コマンド種別 シーケンシャル・コマンド

使用制限 次の設定条件の場合は、無効となりERR (CR / LF) を出力します。

問い合わせ : 1秒データ印字選択機能が OFF の場合

下記の場合はデータ無しとして ER?, EC? の ‘-’ の場合と同様のデータを出力します。

問い合わせ : 測定データ無し

使用例

問い合わせ : 1秒間誤り率が1.0E-4、1秒間誤り個数が1000個の場合

```
OUTPUT△700;"OSD?"
ENTER△700;B$
PRINT△B$
↓
OSD△1.0000E-4,△△△△△1000 (CR / LF) を出力します。
```

: 1秒間誤り率が1.0E-10、1秒間誤り個数が1個の場合

```
OUTPUT△700;"OSD?"
ENTER△700;B$
PRINT△B$
↓
OSD△1.0000E-10,△△△△△△△△1 (CR / LF) を出力します。
```

: EFI率測定結果が ‘-’ の場合

```
OUTPUT△700;"OSD?"
ENTER△700;B$
PRINT△B$
↓
OSD△0.0000E-00,1.0000E-99 (CR / LF) を出力します。
```

77) AMD? アラーム測定結果 (AlarM Data ?)

■ 機 能 アラーム測定データを、下記の出力形式に従って出力します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
AMD	無 し	AMD?m	下記参照

■ m の値 m : アラーム測定項目を選択します。
 0 : 下記1～6まで全項目を出力
 1 : 電源断発生時刻
 2 : 電源断回復時刻
 3 : クロック断発生時刻
 4 : クロック断回復時刻
 5 : 同期外れ発生時刻
 6 : 同期外れ回復時刻

■ 出力フォーマット

全項目出力の場合 : **-*-*-**△**:*:*:**
 年 月 日 時 分 秒
 を上記1～6の順番で','で区切って出力します。
1～6の場合 : 選択項目1項目分の時刻データを以下のフォーマットで出力します。
 -*-*-△**:*:*:**
 年 月 日 時 分 秒

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となり99-99-99△99:99:99 (CR/LF) を出力します。
問い合わせ : アラームが1度も発生していない場合の発生・回復時刻
 アラームが回復していない場合の回復時刻
 測定停止中に起こった発生・回復時刻

■ 使用例 問い合わせ : 電源断発生時刻を問い合わせる場合
 OUTPUT△700;"AMD?△1"
 ENTER△700;B\$
 PRINT△B\$
 ↓
 95-01-28△13:52:59 (CR/LF) を出力します。
 : 無効条件の場合
 OUTPUT△700;"AMD?△2"
 ENTER△700;B\$
 PRINT△B\$
 ↓
 99-99-99△99:99:99 (CR/LF) を出力します。

78) EDS 測定終了データをバッファにストア (End-Data buffer Store)

■ 機 能

測定終了データをバッファにストアします。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
EDS	EDS	無 し	無 し

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合
 測定中で無い場合

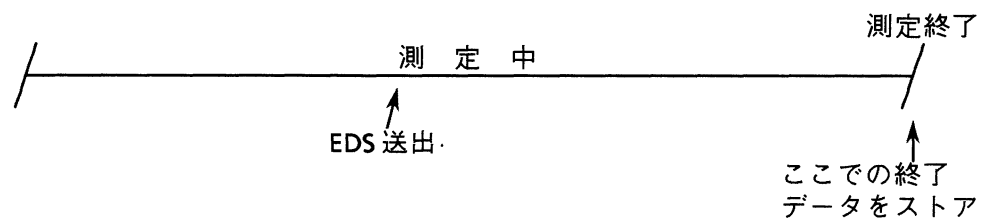
■ 使用例

プログラム : 測定終了データをバッファにストアする場合
OUTPUT△700;"EDS"

■ Note

測定終了データのストアを行うタイミングは、本コマンド実行時に継続中の測定データについて測定終了時にストアします。

測定終了データをストアする場合は、いったんバッファのクリアを行って下さい。



79) EDC 測定終了データのバッファをクリア (End-Data buffer Clear)

■ 機 能 測定終了データのバッファをクリアします。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
EDC	EDC	無 し	無 し

■ コマンド種別 シーケンシャル・コマンド

■ **使用制限** 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

■ 使用例 プログラム : 測定終了データのバッファをクリアする場合
 OUTPUT△700;"EDC"

■ **Note** 測定終了データのクリアを行うと、EDSによってストアされている測定終了データが消去されます。

80) END?**測定終了データの読みだし (END-Data buffer read ?)****機 能**

測定終了データの読みだしをします。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
END	無 し	END?m1,m2	測定終了データを下記の出力フォーマットにしたがって出力します。

m の値**m1** : データの種別

- 0 : 時刻データ
- 1 : アラーム測定データ
- 2 : エラー測定データ
- 3 : スレッシュホールド EI, EFI データ
- 4 : エラー・パフォーマンス・データ

m2 : 出力数

- 0 : 全項目出力

=> 1 : 1項目のデータを出力 (下記の出力フォーマットを参照)

出力フォーマット**m1= 0** : 時刻データ

- m2= 1** : 測定開始時刻 (FORM 1)
- 2 : 測定終了時刻 (FORM 1)
- 3 : 測定経過時刻 (FORM 1)
- 4 : 測定残り時刻 (FORM 1)

m1= 1 : アラーム測定データ

- m2= 1** : POWER FAIL インタバル回数 (FORM 2)
- 2 : CLOCK LOSS インタバル回数 (FORM 2)
- 3 : SYNC LOSS インタバル回数 (FORM 2)

m1= 2 : エラー測定データ

- m2= 1** : ERROR RATIO (FORM 3)
- 2 : ERROR COUNT (FORM 4)
- 3 : CLOCK COUNT (FORM 4)
- 4 : EI (FORM 4)
- 5 : %EFI (FORM 5)

m1= 3 : スレッシュホールド EI, EFI データ

- m2= 1** : $>10^{-3}$ (FORM 6)
- 2 : $>10^{-4}$ (FORM 6)
- 3 : $>10^{-5}$ (FORM 6)
- 4 : $>10^{-6}$ (FORM 6)
- 5 : $>10^{-7}$ (FORM 6)
- 6 : $>10^{-8}$ (FORM 6)
- 7 : $\leq 10^{-8}$ (FORM 6)

m1= 4 : エラー・パフォーマンス・データ

m2= 1 : ES (FORM 5)
 2 : EFS (FORM 5)
 3 : SES (FORM 5)
 4 : DM (FORM 5)
 5 : US (FORM 5)

(FORM 1) 時刻データ型

-*--*△**:*:*:* (文字数17)
 年 月 日 時 分 秒

(FORM 2) 数値データ型

***** (文字数10)

(FORM 3) 指数データ型

*.*****E-* (文字数10)
 *.*****E-* (文字数9)

(FORM 4) 数値・指数データ型

● 1E+8未満
 △***** (文字数9)

● 1E+8以上
 *.*****E** (文字数9)

(FORM5) %データ型

. (文字数8)

(FORM6) 混合データ型

● 1E+8未満
 △*****,***.*** (文字数18)

● 1E+8以上
 *.*****E**,***.*** (文字数18)

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となりERR (CR/LF) を出力します。

問い合わせ : 測定データバッファにデータが無い場合

■ 使用例 問い合わせ : 測定終了データのうち時刻データの測定開始時刻を読み出す場合

OUTPUT△700;"END?△0,1"

ENTER△700;B\$

PRINT△B\$

↓

95-01-30△15:24:59 (CR/LF) を出力します。

81) IMS

測定中間データをバッファにストア
(InterMediate-data buffer Store)

機能

測定中間データをバッファにストアします。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
IMS	IMS	無し	無し

コマンド種別

シーケンシャル・コマンド

使用制限

次の設定条件の場合は、無効となります。

プログラム

: 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合
測定中で無い場合

使用例

プログラム : 測定中間データをバッファにストアする場合
OUTPUT△700;"IMS"

82) IMC

測定中間データのバッファをクリア
(InterMediate-data buffer Clear)

機能

測定中間データのバッファをクリアする。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
IMC	IMC	無 し	無 し

コマンド種別

シーケンシャル・コマンド

使用制限

次の設定条件の場合は、無効となります。

プログラム

： 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

使用例

プログラム： 測定中間データのバッファをクリアする場合
OUTPUT△700;"IMC"

Note

測定中間データのクリアを行うと、IMSによってストアされている測定中間データが消去されます。

83) IMD?**測定中間データの読みだし
(InterMediate-Data buffer read ?)****機 能**

測定中間データの読みだしをします。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
IMD	無 し	IMD?m1,m2	測定中間データを下記の出力フォーマットにしたがって出力します。

m の値**m1** : データの種別

- 0 : 時刻データ
- 1 : アラーム測定データ
- 2 : エラー測定データ
- 3 : スレッシュホールド EI, EFI データ
- 4 : エラー・パフォーマンス・データ

m2 : 出力数

0 : 全項目出力

=> 1 : 1項目のデータを出力 (下記の出力フォーマットを参照)

出力フォーマット**m1 = 0** : 時刻データ

- m2 = 1** : 測定開始時刻 (FORM 1)
- 2 : 測定途中時刻 (FORM 1)
- 3 : 測定経過時刻 (FORM 1)
- 4 : 測定残り時刻 (FORM 1)

m1 = 1 : アラーム測定データ

- m2 = 1** : POWER FAIL インタバル回数 (FORM 2)
- 2 : CLOCK LOSS インタバル回数 (FORM 2)
- 3 : SYNC LOSS インタバル回数 (FORM 2)

m1 = 2 : エラー測定データ

- m2 = 1** : ERROR RATIO (FORM 3)
- 2 : ERROR COUNT (FORM 4)
- 3 : CLOCK COUNT (FORM 4)
- 4 : EI (FORM 4)
- 5 : %EFI (FORM 5)

m1 = 3 : スレッシュホールド EI, EFI データ

- m2 = 1** : $>10^{-3}$ (FORM 6)
- 2 : $>10^{-4}$ (FORM 6)
- 3 : $>10^{-5}$ (FORM 6)
- 4 : $>10^{-6}$ (FORM 6)
- 5 : $>10^{-7}$ (FORM 6)
- 6 : $>10^{-8}$ (FORM 6)
- 7 : $\leq 10^{-8}$ (FORM 6)

m1= 4 : エラー・パフォーマンス・データ

m2= 1 : ES	(FORM 5)
2 : EFS	(FORM 5)
3 : SES	(FORM 5)
4 : DM	(FORM 5)
5 : US	(FORM 5)

(FORM 1) 時刻データ型

--**△**:**:** (文字数17)
年 月 日 時 分 秒

(FORM 2) 数値データ型

***** (文字数10)

(FORM 3) 指数データ型

*.****E-** (文字数10)

*.****E-* (文字数9)

(FORM 4) 数値・指数データ型

● 1E + 8未満
△***** (文字数9)

● 1E + 8以上
*.****E** (文字数9)

(FORM 5) %データ型

. (文字数8)

(FORM 6) 混合データ型

● 1E + 8未満
△*****,***.*** (文字数18)

● 1E + 8以上
*.****E**,***.*** (文字数18)

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となりERR (CR/LF) を出力します。

問い合わせ : 測定中間データバッファにデータが無い場合
測定時間が1秒の場合は中間データは生成されません。

■ 使用例 問い合わせ : 測定中間データのうち時刻データの測定開始時刻を読み出す場合

```
OUTPUT△700;"IMD?△0,1"
ENTER△700;B$
PRINT△B$
```

↓

95-01-30△15:24:59 (CR/LF) を出力します。

● その他のセクション

次のページより、その他のセクションの各コントロール・メッセージについて示します。なお、文中の△はスペースを意味します。

■ Note

本章の背面ファンクションスイッチの設定内容は **Remote** 時はコマンド設定された内容を優先しますが、**Local** 状態になると背面のファンクションスイッチの設定内容に戻ります。

注意して下さい。

84) PRN プリンタ ON/OFF (PRiNter enable)

■ **機 能** プリンタの出力 ON/OFF の制御を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
PRN	PRN△m	PRN?	PRN△m (FIX 1)

■ **m の値** 0 : OFF
1 : ON

■ **コマンド種別** シーケンシャル・コマンド

■ **使用制限** 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ **使用例** プリンタを OFF に設定する場合
OUTPUT△700;"PRN△0"

問い合わせ : プリンタが ON に設定されている場合
OUTPUT△700;"PRN?"
ENTER△700;B\$
PRINT△B\$

↓

PRN△1 (CR/LF) を出力します。

85) PSA マニュアルプリント (Printer StArt)

■ **機 能** マニュアル・プリントの開始指示を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
PSA	PSA	無 し	無 し

■ **コマンド種別** シーケンシャル・コマンド

■ **使用制限** 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合
 測定開始から最初の1秒間の場合
 測定停止中
 プリント機能がOFFの場合

■ **使用例** プログラム : マニュアルプリントを開始する場合
OUTPUT△700;"PSA"

86) ALM アラーム検出機能 (ALarm Monitor on / off)

■ 機 能 アラーム検出機能のON/OFFの制御を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
ALM	ALM△m	ALM?	ALM△m (FIX 1)

■ mの値 0 : OFF
 1 : ON

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

 プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

 問い合わせ : 無し

■ 使用例 プログラム : アラーム検出機能を OFF に設定する場合
 OUTPUT△700;"ALM0"

 問い合わせ : アラーム検出機能が ON に設定されている場合
 OUTPUT△700;"ALM?"
 ENTER△700;B\$
 PRINT△B\$
 ↓
 ALM△1 (CR/LF) を出力します。

87) MON**誤り検出機能 (error MONitor on / off)**

■ 機 能

誤り検出機能の ON / OFF の制御を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
MON	MON△m	MON?	MON△m (FIX 1)

■ m の値

0 : OFF

1 : ON

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : 誤り検出機能を OFF に設定する場合

OUTPUT△700;"MON△0"

問い合わせ : 誤り検出機能が ON に設定されている場合

OUTPUT△700;"MON?"

ENTER△700;B\$

PRINT△B\$

↓

MON△1 (CR / LF) を出力します。

88) SOP 同期信号出力選択 (Sync OutPut)**■ 機 能**

同期信号出力の制御を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
SOP	SOP△m	SOP?	SOP△m (FIX 1)

■ mの値

0 : 1/32 CLOCK

1 : PATTERN SYNC (FIXED)

2 : PATTERN SYNC (VARIABLE)

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : 同期信号出力を1/32 CLOCK に設定する場合
OUTPUT△700;"SOP△0"

問い合わせ : 同期信号出力が PATTERN SYNC (FIXED) に設定されている
 場合

OUTPUT△700;"SOP?"
ENTER△700;B\$
PRINT△B\$

↓

SOP△1 (CR/LF) を出力します。

89) GPA**GPIB 2アドレス (GPIB 2 Address)****■ 機 能**

GPIB 2(プリンタ専用出力ポート)のアドレス設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
GPA	GPA△m	GPA?	GPA△m (FIX 2)

■ mの値

GPIB 2のアドレス1～30までを設定します。

数値範囲 : 最大値 30
 最小値 0
 ステップ 1

■ コマンド種別 シーケンシャル・コマンド**■ 使用制限**

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : GPIB 2アドレスを0に設定する場合
OUTPUT△700;"GPA△0"

問い合わせ : GPIB 2アドレスが1に設定されている場合
OUTPUT△700;"GPA?"
ENTER△700;B\$
PRINT△B\$

↓

GPA△△1 (CR/LF) を出力します。

90) SFT マーク率の AND ビットシフト数 (mark ratio and bit ShiFT)**■ 機 能**

PRBS マーク率の AND ビットシフト数を設定します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
SFT	SFT△m	SFT?	SFT△m (FIX 1)

■ m の値

0 : 1 bit シフト

1 : 3 bit シフト

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合
 測定パターンが ALTERNATE、DATA、ZERO SUBST の場合

問い合わせ : 次の場合は無効となり、ERR (CR/LF) を出力します。
 測定パターンが ALTERNATE、DATA、ZERO SUBST の場合

■ 使用例

プログラム : マーク率の AND ビットシフトを 1 bit に設定する場合
OUTPUT△700;"SFT△0"

問い合わせ : マーク率の AND ビットシフトが 1 bit に設定されている場合
OUTPUT△700;"SFT?"
ENTER△700;B\$
PRINT△B\$

↓

SFT△1 (CR/LF) を出力します。

: 測定パターンが ALTERNATE、DATA、ZERO SUBST に設定されている場合

OUTPUT△700;"SFT?"
ENTER△700;B\$
PRINT△B\$

↓

ERR (CR/LF) を出力します。

91) CLS クロック断処理機能 (CLock loSs)

■ 機 能 クロック断処理機能の選択を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
CLS	CLS△m	CLS?	CLS△m (FIX 1)

■ m の値 0 : EXCLUDE
1 : INCLUDE

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例 プログラム : クロック断の状態を計算から除外する場合
OUTPUT△700;"CLS△0"

問い合わせ : クロック断の状態を計算に含めるように設定されている場合
OUTPUT△700;"CLS?"
ENTER△700;B\$
PRINT△B\$

↓

CLS△1 (CR/LF) を出力します。

■ Note EXCLUDE の場合は、クロック断の場合の処理を計算から除外します。
INCLUDE の場合は、クロック断の場合の処理を計算に含めます。

92) SLS 同期外れ処理機能 (Sync LoSs)

■ 機 能 同期外れ処理機能の選択を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
SLS	SLS△m	SLS?	SLS△m (FIX 1)

■ mの値 0 : EXCLUDE
 1 : INCLUDE

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例 プログラム : 同期外れの状態を計算から除外する場合
 OUTPUT△700;"SLS△0"

問い合わせ : 同期外れの状態を計算に含めるように設定されている場合
 OUTPUT△700;"SLS?"
 ENTER△700;B\$
 PRINT△B\$

↓

SLS△1 (CR/LF) を出力します。

■ Note EXCLUDE の場合は、同期外れの場合の処理を計算から除外します。
 INCLUDE の場合は、同期外れの場合の処理を計算に含めます。

93) ETH**エラー・パフォーマンス・スレッシュホールド選択機能
(Error performance THreshold)****機 能**

エラー・パフォーマンス・データの印字スレッシュホールドの設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
ETH	ETH△m	ETH?	ETH△m (FIX 1)

m の値

0 : 1.0E-3

1 : 1.0E-4

コマンド種別 シーケンシャル・コマンド**使用制限**

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

使用例

プログラム : エラー・パフォーマンスの印字スレッシュホールドを1.0E-3に設定する場合

OUTPUT△700;"ETH△0"

問い合わせ : エラー・パフォーマンスの印字スレッシュホールドが1.0E-4に設定されている場合

OUTPUT△700;"ETH?"**ENTER△700;B\$****PRINT△B\$**

↓

ETH△1 (CR/LF) を出力します。

94) **BST** **BURST 測定モード (BurST mode)**

■ 機 能 BURST 測定モードの設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
BST	BST△m	BST?	BST△m (FIX 1)

■ m の値 0 : OFF
 1 : ON

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例 プログラム : BURST モードを OFF に設定する場合
 OUTPUT△700;"BST△0"

問い合わせ : BURST モードを OFF に設定されている場合
 OUTPUT△700;"BST?"
 ENTER△700;B\$
 PRINT△B\$

↓
BST△1 (CR/LF) を出力します。

■ Note BURST モードが ON の場合は、次のアラーム検出方法が変わります。

- クロック断 : クロック断は100 msec 区間クロックが入力されなかった場合にクロック断となります。それ以外の場合は、クロック断となりません。
- 同期外れ : 同期外れは、検出しません。

95) CAL

測定途中データ算出機能 (current data CALculatation)

■ 機 能

測定途中データ (カレントデータ) の算出方式の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
CAL	CAL△m	CAL?	CAL△m (FIX 1)

■ mの値

0 : 累積計算データ
1 : 即値計算データ

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッショルド・サーチ機能が実行中の場合
アイ・マージン測定が実行中の場合
フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : 測定途中データの算出を累積計算データに設定する場合
OUTPUT△700;"CAL△0"

問い合わせ : 測定途中データの算出が即値計算データに設定されている場合

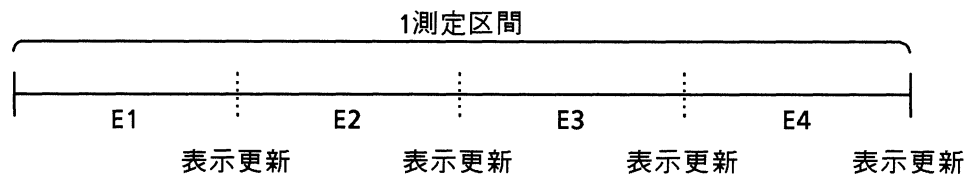
OUTPUT△700;"CAL?"
ENTER△700;B\$
PRINT△B\$

↓

CAL△1 (CR/LF) を出力します。

■ Note

累積データとは、カレントデータ表示更新周期 (100 msec / 200 msec) 毎にデータを加算していきます。



- 累積計算データ = $E1 + E2 + E3 + E4 + \dots$
- 即値計算データ = $E1$
 $E2$
 $E3$
 $E4$

即値計算データは、カレントデータ表示更新周期 (100 msec / 200 msec) 区間での測定データを表示します。(各データは加算されません)

96) ETY 誤り検出モード選択 (Error TYpe)

■ 機 能 誤り検出モードの選択を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
ETY	ETY△m	ETY?	ETY△m (FIX 1)

■ mの値 0 : トータル・エラー
 1 : 挿入エラー
 2 : 消失エラー

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例 プログラム : 誤り検出モードをトータル・エラーに設定する場合
 OUTPUT△700;"ETY△0"
 問い合わせ : 誤り検出モードが挿入エラーに設定されている場合
 OUTPUT△700;"ETY?"
 ENTER△700;B\$
 PRINT△B\$
 ↓
 ETY△1 (CR/LF) を出力します。

97) EIT**EI / %EFI インターバル時間 (Ei, %efi Interval Time)****■ 機 能**

EI, %EFI のインターバル時間の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
EIT	EIT△m	EIT?	EIT△m (FIX 1)

■ m の値

0 : 1 msec
 1 : 10 msec
 2 : 100 msec
 3 : 1 sec

■ コマンド種別 シーケンシャル・コマンド**■ 使用制限**

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : インターバル時間を 1 msec に設定する場合

```
OUTPUT△700;"EIT△0"
```

問い合わせ : インターバル時間が 10 msec に設定されている場合

```
OUTPUT△700;"EIT?"
```

```
ENTER△700;B$
```

```
PRINT△B$
```

↓

```
EIT△1(CR/LF) を出力します。
```

98) FMT データ印字フォーマット (output data ForMaT)

■ 機 能 プリンタ印字フォーマットの設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
FMT	FMT△m	FMT?	FMT△m (FIX 1)

■ mの値 0 : 標準フォーマット
 1 : 短縮フォーマット

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例 プログラム : プリンタ印字フォーマットを標準フォーマットに設定する場合
 OUTPUT△700;"FMT△0"

問い合わせ : プリンタ印字フォーマットを短縮フォーマットに設定する場合
 OUTPUT△700;"FMT?"
 ENTER△700;B\$
 PRINT△B\$
 ↓
 FMT△1 (CR/LF) を出力します。

99) THR**スレッシュヨルド EI, %EFI データ印字選択
(THReshold ei / %efi data output)****■ 機 能**

スレッシュヨルド EI, %EFI データの印字設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
THR	THR△m	THR?	THR△m (FIX 1)

■ m の値

0 : 印字しない

1 : 印字する

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュヨルド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : スレッシュヨルド EI, %EFI データを印字しない場合
OUTPUT△700;"THR△0"

問い合わせ : スレッシュヨルド EI, %EFI データを印字するに設定されている
 場合

OUTPUT△700;"THR?"
ENTER△700;B\$
PRINT△B\$

↓

THR△1 (CR/LF) を出力します。

100) EPF エラー・パフォーマンス・データ印字選択 (Error PerFormance data output)

■ 機 能 エラー・パフォーマンス・データの印字選択を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
EPF	EPF△m	EPF?	EPF△m (FIX 1)

■ mの値 0 : 印字しない
 1 : 印字する

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ:無し

■ 使用例 プログラム : エラー・パフォーマンス・データを印字しない場合
 OUTPUT△700;"EPF△0"

問い合わせ : エラー・パフォーマンス・データを印字する場合に設定されている場合
 OUTPUT△700;"EPF?"
 ENTER△700;B\$
 PRINT△B\$

↓

EPF△1 (CR/LF) を出力します。

101) ITM**中間データ印字選択 (InTerMediate data output)****■ 機 能**

中間データの印字選択を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
ITM	ITM△m	ITM?	ITM△m (FIX 1)

■ mの値

0 : 印字しない

1 : 印字する

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : 中間データを印字しない場合

OUTPUT△700;"ITM△0"

問い合わせ : 中間データを印字するに設定されている場合

OUTPUT△700;"ITM?"

ENTER△700;B\$

PRINT△B\$

↓

ITM△1 (CR/LF) を出力します。

102) OSC**1秒データ印字選択 (One-SeCond data output)****■ 機 能**

1秒データの印字選択を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
OSC	OSC△m	OSC?	OSC△m (FIX 1)

■ m の値

0 : 印字しない

1 : 印字する

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : 1秒データを印字しない場合

OUTPUT△700;"OSC△0"

問い合わせ : 1秒データを印字するに設定されている場合

OUTPUT△700;"OSC?"**ENTER△700;B\$****PRINT△B\$**

↓

OSC△1 (CR/LF) を出力します。

103) DOT**1秒データ印字スレッシュホールド選択 (Data Output Threshold)**

■ 機 能

1秒データの印字スレッシュホールド選択を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
DOT	DOT△m	DOT?	DOT△m (FIX 1)

■ mの値

0 : 誤り > 0
 1 : 誤り > 1.0E-6
 2 : 誤り > 1.0E-4
 3 : 誤り > 1.0E-3

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッシュホールド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : 1秒データ印字スレッシュホールドが誤り > 0に設定する場合
OUTPUT△700;"DOT△0"

問い合わせ : 1秒データ印字スレッシュホールドが誤り > 1.0E-6に設定されている場合

OUTPUT△700;"DOT?"
ENTER△700;B\$
PRINT△B\$

↓

DOT△1 (CR/LF) を出力します。

104) PSV 紙節約機能 (Paper SaVe)

■ 機 能 紙節約機能の選択を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
PSV	PSV△m	PSV?	PSV△m (FIX 1)

■ mの値 0 : OFF
 1 : ON

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッショルド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例 プログラム : 紙節約機能を OFF に設定する場合
 OUTPUT△700;"PSV△0"

問い合わせ : 紙節約機能が ON に設定されている場合
 OUTPUT△700;"PSV?"
 ENTER△700;B\$
 PRINT△B\$

↓

PSV△1 (CR/LF) を出力します。

105) ITV**測定インターバル時間選択 (measurement InTerVal time)****■ 機 能**

測定インターバル時間選択を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
ITV	ITV△m	ITV?	ITV△m (FIX 1)

■ m の値

0 : 100 msec

1 : 200 msec

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

プログラム : 自動位相スレッショルド・サーチ機能が実行中の場合
 アイ・マージン測定が実行中の場合
 フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : 測定インターバル時間の設定を100 msec に設定する場合
OUTPUT△700;"ITV△0"

問い合わせ : 測定インターバル時間の設定が200 msec に設定されている場合

OUTPUT△700;"ITV?"**ENTER△700;B\$****PRINT△B\$**

↓

ITV△1 (CR/LF) を出力します。

106) TRM 終端コード切り替え (TeRMination select)**■ 機 能**

データリクエストコマンドに対するレスポンスデータの終端コードを切り替えます。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
TRM	TRM△m	TRM?	TRM△m (FIX 1)

■ m の値

0 : LF+EOI

1 : CR+LF+EOI

※イニシャライズ時、電源投入時は**TRM 0** 状態で起動します。

■ コマンド種別 シーケンシャル・コマンド**■ 使用制限**

次の設定条件の場合は、無効となります。

プログラム : フロッピー・ディスクをアクセス中の場合

問い合わせ : 無し

■ 使用例

プログラム : レスポンスデータの終端コードをLF+EOIに設定を行う場合
OUTPUT△700;"TRM△0"

問い合わせ : レスポンスデータの終端コードがCR+LF+EOIに設定されている場合

OUTPUT△700;"LGC?"

ENTER△700;B\$

PRINT△B\$

↓

TRM△1 (CR+LF+EOI) を出力します。

107)DSL 入力データ切り替え(Data Select)

■ 機 能 入力信号(Data, $\overline{\text{Data}}$)の切り替えを行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
DSL	DSL△m	DCD?	DCD△m1,m2,m3,m4 m1 (FIX1) m2 (FIX1) m3 (FIX1) m4 (FIX1)

■ m の値 0: DATA
 1: NDATA
 ※ イニシャライズ時、電源投入時は DSL 0 状態で起動します。

■ コマンド種別 シーケンシャル・コマンド

■ 使用制限 次の設定条件の場合は、無効となります。

 (1) Auto search を実施中の場合
 (2) Eye margin 測定を実施中の場合
 (3) FD にアクセスを実施中の場合

■ 使用例 プログラム: 入力信号の選択を Data に設定を行う場合
 OUTPUT△700;"DSL△0"

 問い合わせ: 入力信号が Data に設定されている場合
 OUTPUT△700;"DCD?"
 ENTER△700;B\$
 PRINT△B\$
 ↓
 DCD△0, 0, 0, 0 を出力します。

問い合わせ時のレスポンスパラメータは次の通りです。

DCD△m1,m2,m3,m4

m1:DATA/NDATA (0:DATA／1:NDATA)

 m2:Tracking/Adjust
 (0:Tracking/Adjust OFF／1:Tracking ON／2: Adjust ON／-1:無効)

 m3:Single/Differential (0:Single／1:Differential)

 m4:Term. Condition (0:Single/50Ω／1:Differential/100Ω)

108)DAC Tracking/Adjust操作の選択(Data tracking/Adjust Control)

機能 入力信号(Data, $\overline{\text{Data}}$)の Tracking/Adjust 操作の選択を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
DAC	DAC△m	DCD?	DCD△m1,m2,m3,m4 m1 (FIX1) m2 (FIX1) m3 (FIX1) m4 (FIX1)

m の値 0: Tracking/Adjust OFF
1: Tracking ON (Adjust OFF)
2: Adjust ON (Tracking OFF)
※ イニシャライズ時、電源投入時は DAC 1 状態で起動します。

コマンド種別 シーケンシャル・コマンド

使用制限 次の設定条件の場合は、無効となります。

- (1) Auto search を実施中の場合
- (2) Eye margin 測定を実施中の場合
- (3) FD にアクセスを実施中の場合

使用例 プログラム: 入力信号の操作指示を Tracking/Adjust OFF に設定を行う場合
OUTPUT△700;"DAC△0"

問い合わせ: 入力信号の操作指示が Tracking/Adjust OFF に設定されている場合
OUTPUT△700;"DCD?"
ENTER△700;B\$
PRINT△B\$
↓
DCD△0, 0, 0, 0 を出力します。

問い合わせ時のレスポンスパラメータは次の通りです。

DCD△m1,m2,m3,m4
m1:DATA/NDATA (0:DATA／1:NDATA)
m2:Tracking/Adjust
(0:Tracking/Adjust OFF／1:Tracking ON／2:Adjust ON／-1:無効)
m3:Single/Differential (0:Single／1:Differential)
m4:Term. Condition (0:Single/50Ω／1:Differential/100Ω)

109)DSD 入力信号数の選択(Data Single/Differential select)

■ 機 能

対象入力信号の数を選択します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
DSD	DSD△m	DCD?	DCD△m1,m2,m3,m4 m1 (FIX1) m2 (FIX1) m3 (FIX1) m4 (FIX1)

■ m の値

0: Single

1: Differential

※ イニシャライズ時、電源投入時は DSD 1 状態で起動します。

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

- (1) Auto search を実施中の場合
- (2) Eye margin 測定を実施中の場合
- (3) FD にアクセスを実施中の場合

■ 使用例

プログラム: 対象入力信号を Single に設定を行う場合

OUTPUT△700;"DSD△0"

問い合わせ: 入力信号の操作指示が Single に設定されている場合

OUTPUT△700;"DCD?"

ENTER△700;B\$

PRINT△B\$

↓

DCD△0, 0, 0, 0 を出力します。

問い合わせ時のレスポンスパラメータは次の通りです。

DCD△m1,m2,m3,m4

m1:DATA/NDATA (0:DATA／1:NDATA)

m2:Tracking/Adjust

(0:Tracking/Adjust OFF／1:Tracking ON／2:Adjust ON／-1:無効)

m3:Single/Differential (0:Single／1:Differential)

m4:Term. Condition (0:Single/50Ω／1:Differential/100Ω)

110)RRC 供給クロック選択(external/Recovery Clock select)

機能 供給クロック源の選択を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
RRC	RRC△m	DRC?	DRC△m1,m2 m1 (FIX1) m2 (FIX1)

m の値 0: External
1: Recovery
※ イニシャライズ時、電源投入時は RRC 0 状態で起動します。

コマンド種別 シーケンシャル・コマンド

使用制限 次の設定条件の場合は、無効となります。
(1) Auto search を実施中の場合
(2) Eye margin 測定を実施中の場合
(3) FD にアクセスを実施中の場合

使用例 プログラム: クロックを External に設定を行う場合
OUTPUT△700;"RRC△0"
問い合わせ: クロックが External に設定されている場合
OUTPUT△700;"DRC?"
ENTER△700;B\$
PRINT△B\$
↓
DRC△0, 0 を出力します。

問い合わせ時のレスポンスパラメータは次の通りです。
DRC△m1,m2
m1:External/Recovery (0:External／1:Recovery)
m2:Recovery Frequency(0 to 637／-1:無効)

111) RPL PLLリセット(Recovery PLL reset)

■ 機能

PLLリセットを実施します。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
RPL	RPL	RPL?	RPL△m (FIX1)

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

- (1) Auto search を実施中の場合
- (2) Eye margin 測定を実施中の場合
- (3) FD にアクセスを実施中の場合

■ 使用例

プログラム: PLLリセットを実施する場合

OUTPUT△700;"RPL"

問い合わせ: PLL のステータスを問い合わせる場合

OUTPUT△700;"RPL?"**ENTER△700;B\$****PRINT△B\$**

↓

RPL△0 を出力します。

問い合わせ時のレスポンスパラメータは次の通りです。

RPL△m (0:Lock／1:Unlock／-1:無効)

112)RFQ Recovery周波数設定(Recovery FreQuency)

機能 Recovery 周波数の設定を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
RFQ	RFQ△m	DRC?	DRC△m1,m2 m1 (FIX1) m2 (FIX1)

- m の値
- 0: 62.5MHz/3G CR
 - 1: 70.3MHz/3G CR
 - 2: 78.1MHz/3G CR
 - 3: 85.9MHz/3G CR
 - 4: 93.8MHz/3G CR
 - 5: 100.0MHz/3G CR
 - 6: 125.0MHz/3G CR
 - 7: 140.6MHz/3G CR
 - 8: 156.3MHz/3G CR
 - 9: 171.9MHz/3G CR
 - 10: 187.5MHz/3G CR
 - 11: 200.0MHZ/3G CR
 - 12: 250.0MHz/3G CR
 - 13: 281.3MHz/3G CR
 - 14: 312.5MHz/3G CR
 - 15: 343.8MHz/3G CR
 - 16: 375.0MHz/3G CR
 - 17: 400.0MHz/3G CR
 - 18: 500.0MHz/3G CR
 - 19: 562.5MHz/3G CR
 - 20: 625.0MHz/3G CR
 - 21: 687.5MHz/3G CR
 - 22: 750.0MHz/3G CR
 - 23: 800.0MHz/3G CR
 - 24: 1.000GHz/3G CR
 - 25: 1.125GHz/3G CR
 - 26: 1.250GHz/3G CR
 - 27: 1.375GHz/3G CR
 - 28: 1.500GHz/3G CR
 - 29: 1.600GHz/3G CR
 - 30: 2.000GHZ/3G CR
 - 31: 2.250GHZ/3G CR
 - 32: 2.500GHz/3G CR
 - 33: 2.750GHz/3G CR
 - 34: 3.000GHz/3G CR
 - 35: 3.200GHz/3G CR (Default:3G Unit が最大の場合)
 - 36: 4.250GHz/4G CR (Default:4G Unit が最大の場合)
 - 37: 9.900GHz/10G CR (Default:10G Unit が最大の場合)
 - 38: 以降 2MHz Step で 11.1GHzまで(~637)

※ イニシャライズ時、電源投入時は上記 Default 状態で起動します。

■ コマンド種別

シーケンシャル・コマンド

■ 使用制限

次の設定条件の場合は、無効となります。

- (1) Auto search を実施中の場合
- (2) Eye margin 測定を実施中の場合
- (3) FD にアクセスを実施中の場合

■ 使用例

プログラム: 周波数を 62.5 MHz に設定を行う場合

OUTPUT△700;"RFQ△0"

問い合わせ: 周波数が 62.5 MHz に設定されている場合

OUTPUT△700;"DRC?"

ENTER△700;B\$

PRINT△B\$

↓

DRC△1, 0 を出力します。

問い合わせ時のレスポンスパラメータは次の通りです。

DRC△m1,m2

m1:External/Recovery(0:External／1:Recovery)

m2:Recovery Frequency(0 to 637／-1:無効)

113)DTC 終端の差動選択(Data Term. Condition)

機能 終端の差動選択を行います。

ヘッダ	プログラム	問い合わせ	レスポンス (文字数)
DTC	DTC△m	DCD?	DCD△m1,m2,m3,m4 m1 (FIX1) m2 (FIX1) m3 (FIX1) m4 (FIX1)

m の値 0: Single/50Ω
1: Differential/100Ω
※ イニシャライズ時、電源投入時は DTC 0 状態で起動します。

コマンド種別 シーケンシャル・コマンド

使用制限 次の設定条件の場合は、無効となります。
(1) Auto search を実施中の場合
(2) Eye margin 測定を実施中の場合
(3) FD にアクセスを実施中の場合

使用例 プログラム: 差動選択を Single/50Ωに設定を行う場合
OUTPUT△700;"DTC△0"

問い合わせ: 差動選択が Single/50Ωに設定されている場合
OUTPUT△700;"DCD?"
ENTER△700;B\$
PRINT△B\$
↓
DCD△0, 0, 0, 0 を出力します。

問い合わせ時のレスポンスパラメータは次の通りです。
DCD△m1,m2,m3,m4
m1:DATA/ NDATA (0:DATA／1:NDATA)
m2:Tracking/Adjust
(0:Tracking/Adjust OFF／1: Tracking ON／2:Adjust ON／-1 無効)
m3: Single/Differential (0:Single／1:Differential)
m4:Term. Condition (0:Single/50Ω／1:Differential/100Ω)

10 章

プログラム作成例

目 次

HP9000を用いたサンプル・プログラム例	10-6
DECpcを用いたサンプル・プログラム例	10-70

10 章 プログラム作成例

この章では、MP1762Dにおけるプログラム作成例を示します。

ホストコントローラとして、HP社のHP9000シリーズおよびIBM-PC/ATまたはその互換機にナショナル・インスツルメント社(N.I.)社製 GPIB インタフェース・カードを付加したものを想定して作成しました。

記述言語は、HP-BASIC およびマイクロ・ソフト社 QUICK-BASIC です。

プログラムの検証は、HP-9000-200/300, HP-BASIC V5.12 および DECpc, N.I. 社 GPIB インタフェース・カード, QUICK-BASIC V4.50 で行いました。

プログラム例は、

- (1) 入力信号の設定
- (2) 自動スレッシュホールドサーチ(オートサーチ)設定
- (3) アイ・マージンの測定
- (4) 測定パターン・BIT WINDOW・BLOCK WINDOW の設定
- (5) 誤り分析
- (6) 測定結果の表示(シリアル・ボールで測定結果を表示)
- (7) 測定結果の表示(リクエスト・コマンドで測定結果を表示)
- (8) 測定中間データの表示
- (9) フロッピー・ディスクからのファイル情報読みだし
- (10) フロッピー・ディスクの操作
- (11) ステータス・バイトのチェック
- (12) パターン・データの DMA 転送
- (13) BLOCK WINDOW パターン・データの DMA 転送

プログラムの実行にあたっての準備を、使用コントローラ別に表10-1に示します。

表10-1 サンプル・プログラムの実行にあたっての準備 (1/2)

使用コントローラ	プログラム実行の準備																																																												
HP9000	<ul style="list-style-type: none"> ● 本器の GPIB アドレスを “1” に設定します。 ● MP1761B/C の GPIB アドレスを “2” に設定します。 ● 本器と MP1761B/C と HP9000 を GPIB ケーブルで接続します。 																																																												
DEC pc	<ul style="list-style-type: none"> ● 本器の GPIB アドレスを “1” に設定します。 ● MP1761B/C の GPIB アドレスを “2” に設定します。 ● IBCONF の設定の一例 (但し、御使用される機器の状態によって設定は異なります。) <p>① <Board Characteristics></p> <p>Board : GPIB 0 (Board を “GPIB 0” とします。)</p> <table> <tr><td>Primary GPIB Address</td><td>0</td></tr> <tr><td>Secondary GPIB Address</td><td>NONE</td></tr> <tr><td>Timeout setting</td><td>1000 sec</td></tr> <tr><td>Terminate Read on EOS</td><td>Yes</td></tr> <tr><td>Set EOI with EOS on Writes</td><td>Yes</td></tr> <tr><td>Type of Compare on EOS</td><td>7-Bit</td></tr> <tr><td>EOS byte</td><td>0AH</td></tr> <tr><td>Send EOI at end of Write</td><td>Yes</td></tr> <tr><td>System Controller</td><td>Yes</td></tr> <tr><td>Assert REN when SC</td><td>No</td></tr> <tr><td>Enable Auto Serial Polling</td><td>Yes</td></tr> <tr><td>Enable CIC Protocol</td><td>No</td></tr> <tr><td>Bus timing</td><td>500 nsec</td></tr> <tr><td>Cable Length for High Speed</td><td>off</td></tr> <tr><td>Parallel Poll Duration</td><td>Default</td></tr> <tr><td>Use this GPIB interface</td><td>Yes</td></tr> <tr><td>Base I / O Address</td><td>02C0h</td></tr> <tr><td>Interrupt Level</td><td>11</td></tr> <tr><td>DMA Channel</td><td>5</td></tr> <tr><td>DMA Transfer Mode</td><td>Demand</td></tr> </table> <p>② <Device Characteristics></p> <p>Device : ED (デバイス名を “ED” とします。)</p> <table> <tr><td>Primary GPIB Address</td><td>1</td></tr> <tr><td>Secondary GPIB Address</td><td>NONE</td></tr> <tr><td>Timeout setting</td><td>1000 sec</td></tr> <tr><td>Serial Poll Timeout</td><td>1 sec</td></tr> <tr><td>Terminate Read on EOS</td><td>Yes</td></tr> <tr><td>Set EOI with EOS on Writes</td><td>Yes</td></tr> <tr><td>Type of compare on EOS</td><td>7-Bit</td></tr> <tr><td>EOS byte</td><td>0Ah</td></tr> <tr><td>Send EOI at end of Write</td><td>Yes</td></tr> <tr><td>Enable Repeat Addressing</td><td>No</td></tr> </table>	Primary GPIB Address	0	Secondary GPIB Address	NONE	Timeout setting	1000 sec	Terminate Read on EOS	Yes	Set EOI with EOS on Writes	Yes	Type of Compare on EOS	7-Bit	EOS byte	0AH	Send EOI at end of Write	Yes	System Controller	Yes	Assert REN when SC	No	Enable Auto Serial Polling	Yes	Enable CIC Protocol	No	Bus timing	500 nsec	Cable Length for High Speed	off	Parallel Poll Duration	Default	Use this GPIB interface	Yes	Base I / O Address	02C0h	Interrupt Level	11	DMA Channel	5	DMA Transfer Mode	Demand	Primary GPIB Address	1	Secondary GPIB Address	NONE	Timeout setting	1000 sec	Serial Poll Timeout	1 sec	Terminate Read on EOS	Yes	Set EOI with EOS on Writes	Yes	Type of compare on EOS	7-Bit	EOS byte	0Ah	Send EOI at end of Write	Yes	Enable Repeat Addressing	No
Primary GPIB Address	0																																																												
Secondary GPIB Address	NONE																																																												
Timeout setting	1000 sec																																																												
Terminate Read on EOS	Yes																																																												
Set EOI with EOS on Writes	Yes																																																												
Type of Compare on EOS	7-Bit																																																												
EOS byte	0AH																																																												
Send EOI at end of Write	Yes																																																												
System Controller	Yes																																																												
Assert REN when SC	No																																																												
Enable Auto Serial Polling	Yes																																																												
Enable CIC Protocol	No																																																												
Bus timing	500 nsec																																																												
Cable Length for High Speed	off																																																												
Parallel Poll Duration	Default																																																												
Use this GPIB interface	Yes																																																												
Base I / O Address	02C0h																																																												
Interrupt Level	11																																																												
DMA Channel	5																																																												
DMA Transfer Mode	Demand																																																												
Primary GPIB Address	1																																																												
Secondary GPIB Address	NONE																																																												
Timeout setting	1000 sec																																																												
Serial Poll Timeout	1 sec																																																												
Terminate Read on EOS	Yes																																																												
Set EOI with EOS on Writes	Yes																																																												
Type of compare on EOS	7-Bit																																																												
EOS byte	0Ah																																																												
Send EOI at end of Write	Yes																																																												
Enable Repeat Addressing	No																																																												

表10-1 サンプル・プログラムの実行にあたっての準備 (2 / 2)

使用コントローラ	プログラム実行の準備
DEC pc	<div>③ <Device Characteristics> Device : PPG (デバイス名を “PPG” とします。) Primary GPIB Address2 Secondary GPIB AddressNONE Timeout setting1000 sec Serial Poll Timeout1 sec Terminate Read on EOSYes Set EOI with EOS on WritesYes Type of compare on EOS7-Bit EOS byte0Ah Send EOI at end of WriteYes Enable Repeat AddressingNo</div> <div>④ GPIB Device Map で②のデバイスと③のデバイスを①の GPIB 0 に接続します。</div> <div>● 本器と MP1761B/C と DEC pc を GPIB ケーブルで接続します。</div>

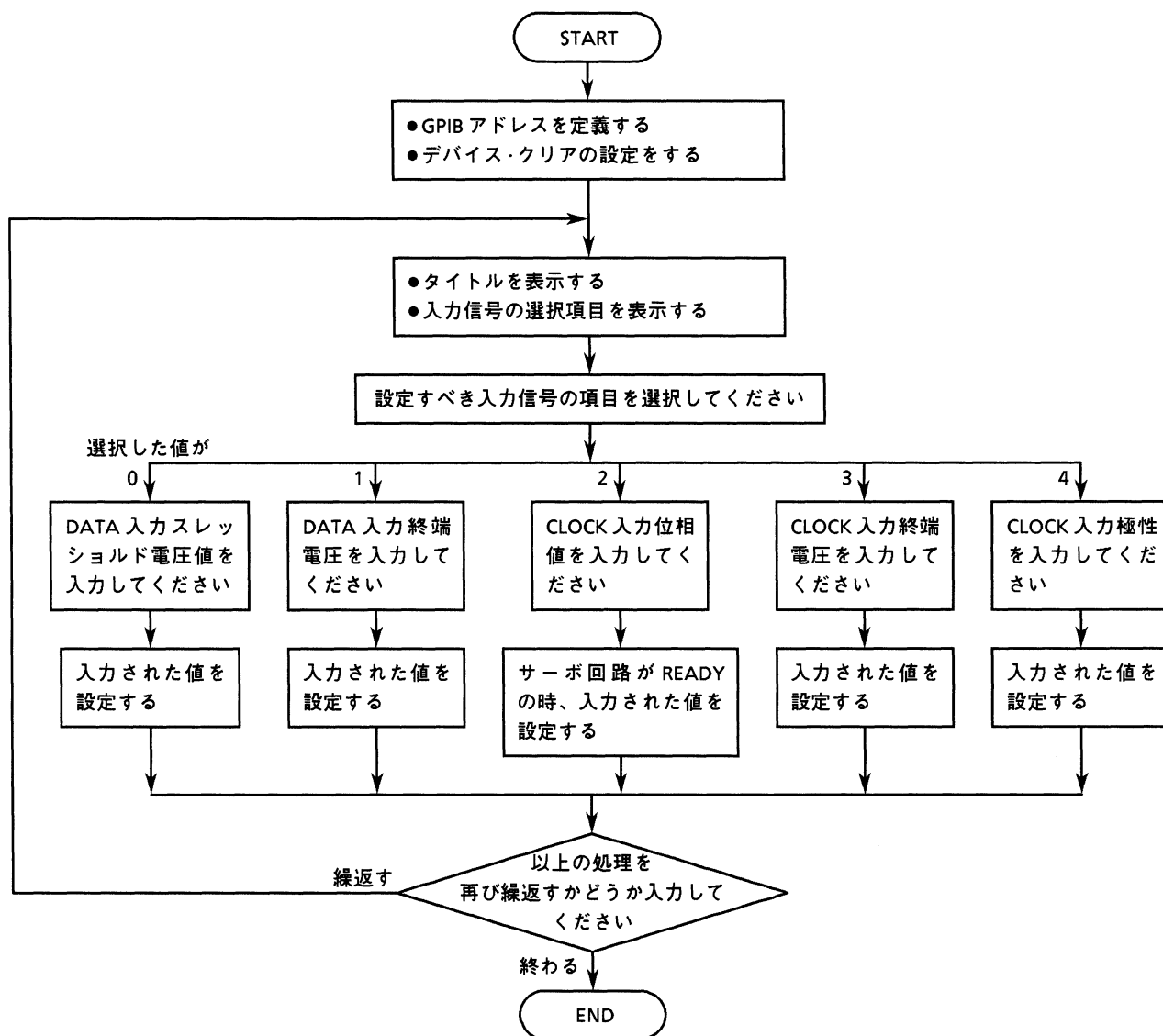
HP9000 を用いたサンプル・プログラム例

(1) 入力信号の設定

本プログラムは、入力信号関係の **GPIB** 制御を行います。

入力信号 (**DATA**, **CLOCK** の電圧, 位相, 極性など) を選択してメッセージにしたがった値を入力すると、それを本器に設定します。

なお、クロック入力位相の設定では、ディレイ状態をチェックし、**READY** になっていることを確かめてから設定しています。また、プログラム例では、差動入力を **DATA** (default) のみとして扱っています。



● プログラム・リスト

```

10  !*****
20  !*
30  !*          MP1762D/MP1764D INPUT SIGNAL SAMPLE PROGRAM          *
40  !*
50  !*****
60  !
70  Add=701                      !MP1762D/MP1764D ADDRESS
80  CLEAR Add                    !DEVICE CLEAR
90  !
100 LOOP
110  CLEAR SCREEN
120  !
130  PRINT "** MP1762D/MP1764D INPUT SIGNAL SAMPLE PROGRAM **"
140  PRINT
150  PRINT "INPUT SIGNAL * DATA THRESHOLD      = [0] "
160  PRINT "          * DATA TERMINATION      = [1] "
170  PRINT "          * CLOCK PHASE ADJUST      = [2] "
180  PRINT "          * CLOCK TERMINATION      = [3] "
190  PRINT "          * CLOCK POLARITY        = [4] "
200  PRINT
210  INPUT "Choose function [0 to 4]:",Sel$
220  !
230  IF Sel$<>"0" AND Sel$<>"1" AND Sel$<>"2" AND Sel$<>"3" AND Sel$<>"4" T
HEN
240      PRINT "Wrong chosen number!"
250      PRINT "Please enter correct number"
260  END IF
270  !
280  SELECT Sel$
290  !
300      CASE "0"
310          PRINT "Please type number for the DATA THRESHOLD"
320          INPUT "Possible data range is -3.000 to +1.875V STEP 0.001V",D
th$
330          OUTPUT Add;"DTH "&Dth$
340          !
350      CASE "1"
360          INPUT "Choose DATA TERMINATION.[GND:0, -2V:1]",Dtm$
380          OUTPUT Add;"DTM "&Dtm$
390          !
400      CASE "2"
410          PRINT "Please type number for the CLOCK PHASE ADJUST"
420          INPUT "Possible data range is -500 to +500ps STEP 1ps",Cpa$
430          !
440          LOOP
450              OUTPUT Add;"DLY?"          !REQUEST Delay unlock
460              ENTER Add;Dly$
470              EXIT IF Dly$="DLY 0"
480          END LOOP
490          !
500          OUTPUT Add;"CPA "&Cpa$
510          !
520      CASE "3"
530          INPUT "Choose CLOCK TERMINATION.[GND:0, -2V:1]",Ctm$
540          OUTPUT Add;"CTM "&Ctm$
550          !
560      CASE "4"
570          INPUT "Choose CLOCK POLARITY.[CLK:0, NCLK:1]",Cp1$
580          OUTPUT Add;"CPL "&Cp1$
590          !
600  END SELECT
610  !
620  INPUT "Do you set another data?[Yes:0, No:1]",Loop$
630  EXIT IF Loop$="1"
640  END LOOP
650  END

```

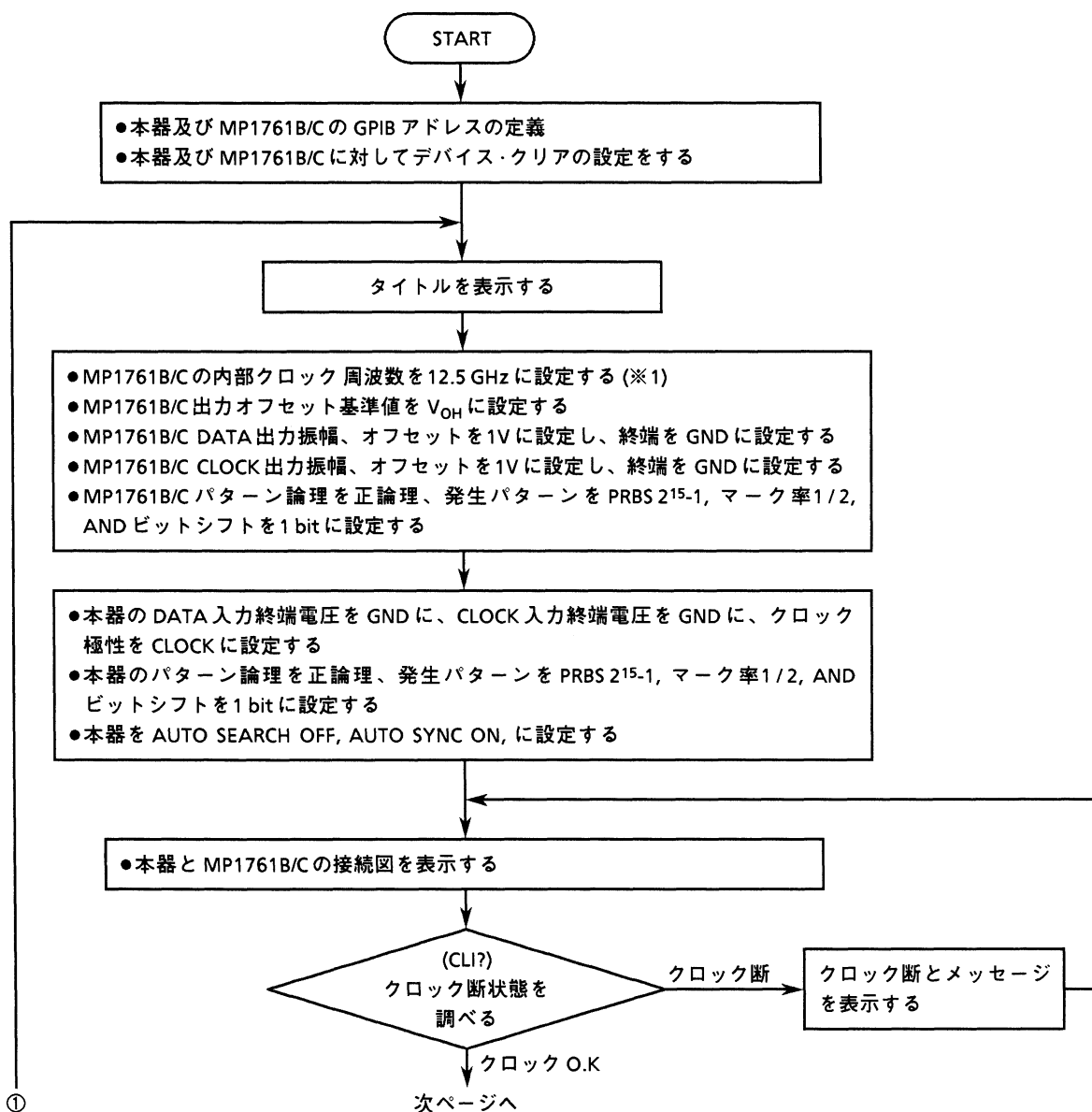
(2) 自動入力スレッシュホールド・サーチ (オート・サーチ) の設定

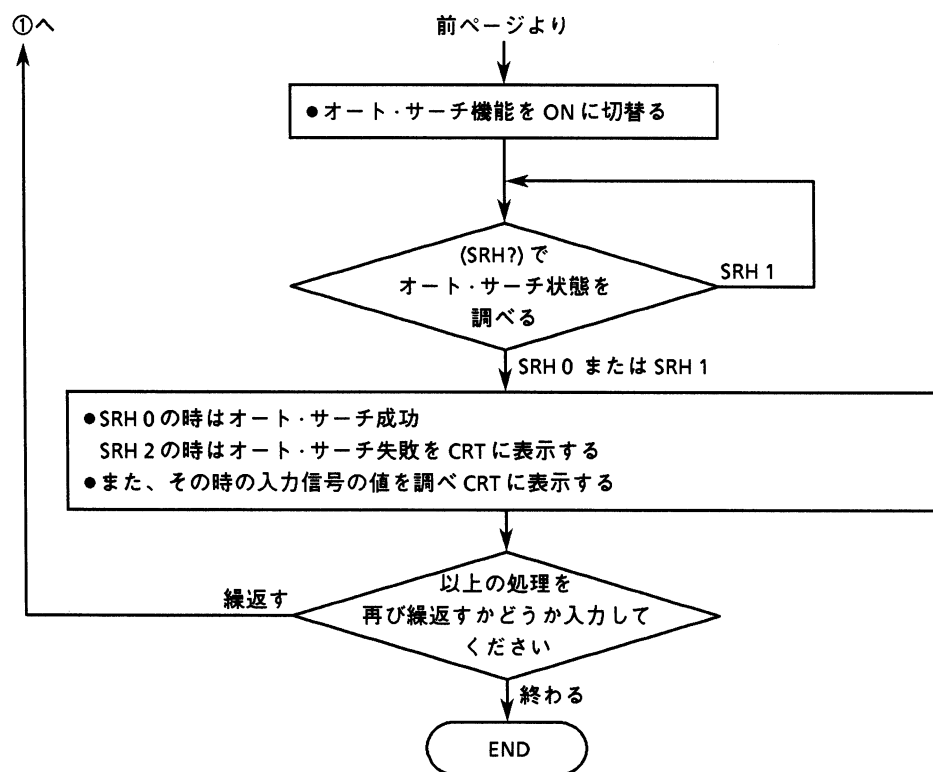
本プログラムは、本器と MP1761B/C を接続して、オートサーチを実行するプログラムです。

まず始めに、オート・サーチを実行する条件を満たすよう、必要な設定条件を本器と MP1761B/C に設定します。

次にクロック断の時はオート・サーチは無効なので、クロック断でないことを確認して、オート・サーチ機能を ON に切替えます。そして、リクエスト・コマンド (SRH?) でオート・サーチが成功したか失敗したかを読み出し CRT に表示します。

また、その時の入力信号の値を調べ CRT に表示します。





■ Note :

- (※1) MP1761B/C の内部クロック周波数の設定はMP1761B/C OPTION-01 内蔵シンセサイザ実装時にのみ有効となります。

● プログラム・リスト

```

10  !*****
20  !*
30  !*      MP1762D/MP1764D AUTO SEARCH SAMPLE PROGRAM      *
40  !*
50  !*****
60  !
70  Add1=701
80  Add2=702
90  CLEAR Add1
100 CLEAR Add2
110 !
120 !
130 LOOP
140 !
150 CLEAR SCREEN
160 PRINT "*** MP1762D/MP1764D AUTO SEARCH SAMPLE PROGRAM *** "
170 PRINT
180 !
190 GOSUB D_set
200 GOSUB Clock
210 GOSUB Brch
220 GOSUB Result
230 !
240 INPUT " Next data set[Yes:0, No:1],Loop$
250 EXIT IF Loop$="1"
260 END LOOP
270 !
280 STOP
290 !
300 !***** MP1761B/C,MP1763B/C/MP1762D,MP1764D DATA SETTING *****
310 D_set:
320 !MP1761B/C/MP1763B/C DATA SETTING
330 !
340 OUTPUT Add2;"CLK 1;RES 1;FRQ 12500"
350 OUTPUT Add2;"OFS 0"
360 OUTPUT Add2;"DAP 1;DOS 1;DTM 0"
370 OUTPUT Add2;"CDL 100;CAP 1;COS 1"
380 OUTPUT Add2;"LGC 0;PTS 3;PTN 6;MRK 3;SFT 0"
390 !
400 !MP1762D/MP1764D DATA SETTING
410 !
420 OUTPUT Add1;"DTM 0;CTM 0;CPL 0"
430 OUTPUT Add1;"LGC 0;PTS 3;PTN 6;MRK 3;SFT 0"
440 OUTPUT Add1;"SRH 0;SYN 1"
450 !
460 RETURN
470 !
480 !
490 !***** Check Connection *****
500 Clock:
510 !
520 LOOP
530 !
540 GOSUB Connect
550 !
560 OUTPUT Add1;"CLI?"
570 ENTER Add1;Cli$
580 IF Cli$="CLI 1" THEN
590 PRINT "***** CLOCK LOSS ***** "
600 END IF
610 EXIT IF Cli$="CLI 0"

```

```

620 END LOOP
630 !
640 RETURN
650 !
660 !
670 !***** AUTO SEARCH FUNCTION *****
680 Srch: !
690 !
700 OUTPUT Add1;"SRH 1"
710 !
720 LOOP
730 !
740 OUTPUT Add1;"SRH?"
750 ENTER Add1;Srch$
760 !
770 EXIT IF Srch$="SRH 0" OR Srch$="SRH 2"
780 END LOOP
790 !
800 IF Srch$="SRH 0" THEN
810 PRINT "***** AUTO SEARCH OK ***** "
820 ELSE
830 PRINT "***** Failed in AUTO SEARCH ***** "
840 END IF
850 !
860 RETURN
870 !
880 !
890 !***** DISPLAY RESULT FUNCTION *****
900 Result: !
910 !
920 OUTPUT Add1;"DTH?"
930 ENTER Add1;Dth$
940 OUTPUT Add1;"DTM?"
950 ENTER Add1;Dtm$
960 IF Dtm$="DTM 0" THEN
970 Dtm$="GND"
980 ELSE
990 Dtm$="-2V"
1000 END IF
1010 !
1020 OUTPUT Add1;"CPA?"
1030 ENTER Add1;Cpa$
1040 !
1050 OUTPUT Add1;"CTM?"
1060 ENTER Add1;Ctm$
1070 IF Ctm$="CTM 0" THEN
1080 Ctm$="GND"
1090 ELSE
1100 Ctm$="-2V"
1110 END IF
1120 !
1130 OUTPUT Add1;"CPL?"
1140 ENTER Add1;Cpl$
1150 IF Cpl$="CPL 0" THEN
1160 Cpl$="CLK"
1170 ELSE
1180 Cpl$="NCLK"
1190 END IF
1200 !
1210 PRINT "DATA THRESHOLD = "&Dth$[5,10]&" V"
1220 PRINT "DATA TERMINATION = "&Dtm$
1230 PRINT "CLOCK PHASE ADJUST = "&Cpa$[6,9]&" ps"
1240 PRINT "CLOCK TERMINATION = "&Ctm$
1250 PRINT "CLOCK POLARITY = "&Cpl$
1260 PRINT
1270 !

```

```

1290 !
1300 !
1310 !***** DISPLAY CONNECTION *****
1320 Connect: !
1330 !
1340 PEN 3
1350 VIEWPORT 70,140,50,100
1360 SHOW 0,70,0,50
1370 !
1380 CLIP 0,70,5,70
1390 FRAME
1400 !
1410 CSIZE 3,.4
1420 MOVE 25,45
1430 LABEL "<< CONNECTION >>"
1440 !
1450 CSIZE 3,.35
1460 MOVE 6,39
1470 LABEL " MP1761A/MP1763A MP1762D/MP1764D"
1480 !
1490 MOVE 7,20
1500 RECTANGLE 25,18
1510 !
1520 MOVE 38,20
1530 RECTANGLE 25,18
1540 !
1550 MOVE 26,14
1560 IDRAW 0,9
1570 !
1580 FOR I=0 TO PI*2 STEP PI/12
1590 IDRAW .2*COS(I),.2*SIN(I)
1600 NEXT I
1610 !
1620 MOVE 26,14
1630 IDRAW 21,0
1640 IDRAW 0,9
1650 !
1660 FOR I=0 TO PI*2 STEP PI/12
1670 IDRAW .2*COS(I),.2*SIN(I)
1680 NEXT I
1690 !
1700 MOVE 21,17
1710 IDRAW 0,6
1720 !
1730 FOR I=0 TO PI*2 STEP PI/12
1740 IDRAW .2*COS(I),.2*SIN(I)
1750 NEXT I
1760 !
1770 MOVE 21,17
1780 IDRAW 21,0
1790 IDRAW 0,6
1800 !
1810 FOR I=0 TO PI*2 STEP PI/12
1820 IDRAW .2*COS(I),.2*SIN(I)
1830 NEXT I
1840 !
1850 MOVE 16,25
1860 CSIZE 2.3,.5
1870 LABEL "DATA CLOCK1 DATA CLOCK"
1880 !
1890 INPUT "Are you ready ? Press return key to start.",A
1900 !
1910 RETURN
1920 !
1930 !
1940 END

```

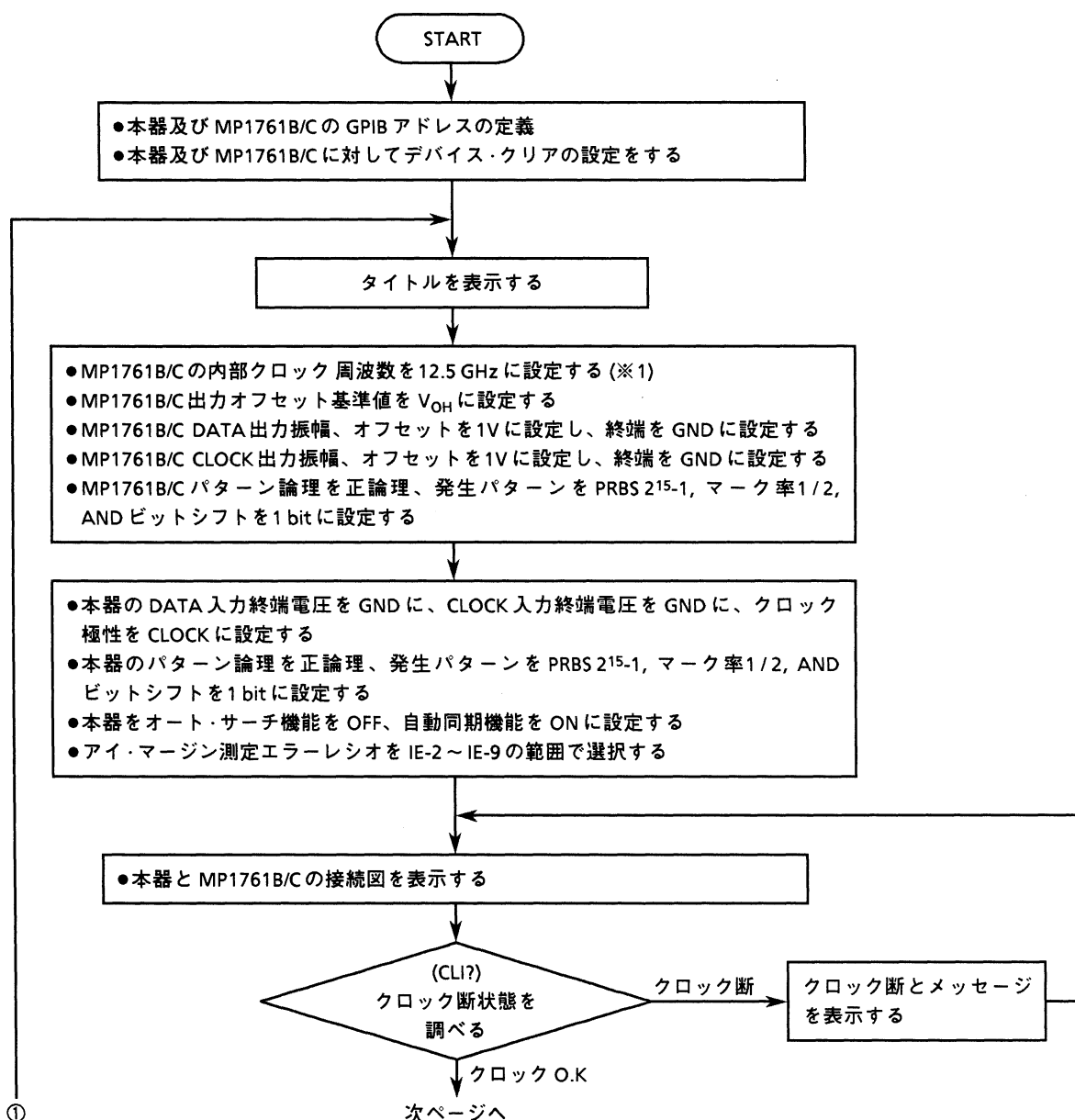
(3) アイ・マージンの測定

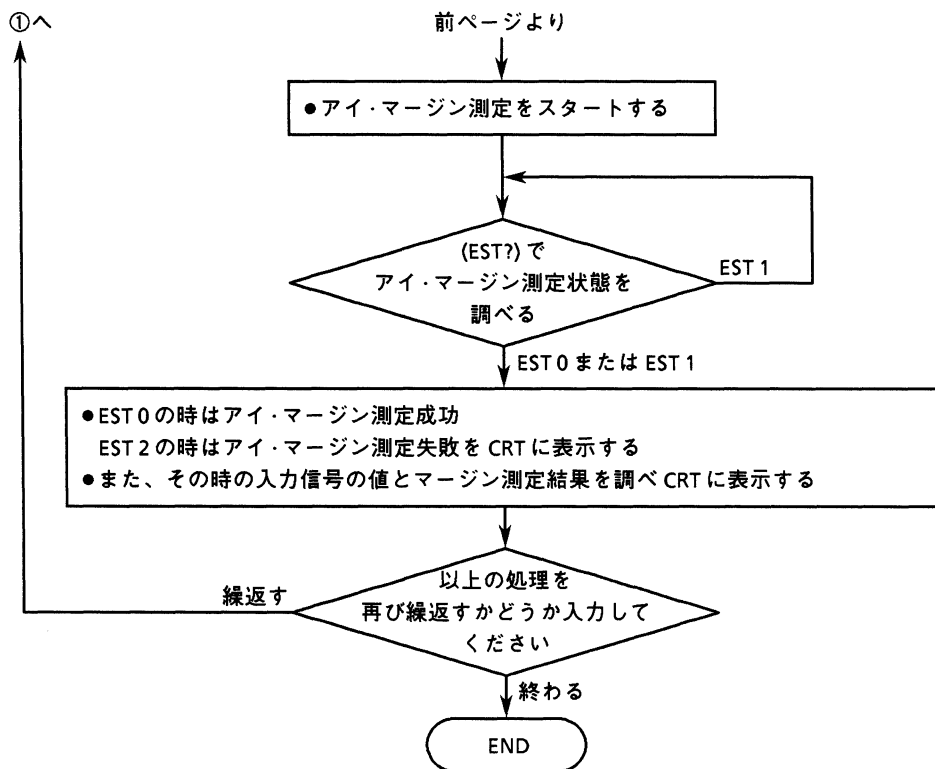
本プログラムは、本器と MP1761B/C を接続して、アイ・マージン測定を実行するプログラムです。

まず始めに、アイ・マージン測定を実行する条件を満たすよう、必要な設定条件を本器と MP1761B/C に設定します。

次にクロック断の時はアイ・マージン測定は無効なので、クロック断でないことを確認して、アイ・マージン測定をスタートします。そして、リクエスト・コマンド (EST?) でアイ・マージン測定が成功したか失敗したかを読み出し CRT に表示します。

また、その時の入力信号の値と測定結果を調べ CRT に表示します。





■ Note :

- (※1) MP1761B/C の内部クロック周波数の設定はMP1761B/C OPTION-01 内蔵シンセサイザ実装時にのみ有効となります。

● プログラム・リスト

```

10  !*****
20  !*
30  !*          MP1762D/MP1764D EYE MARGIN SAMPLE PROGRAM          *
40  !*                                          EYE_MRGN          *
50  !*****
60  !
70  Add1=701                                !MP1762D/MP1764D ADDRESS
80  Add2=702                                !MP1761B/C/MP1763B/C ADDRESS
90  CLEAR Add1                             !DEVICE CLEAR(ED)
100 CLEAR Add2                             !DEVICE CLEAR(PPG)
110  !
120  !
130  LOOP
140  !
150      CLEAR SCREEN
160      PRINT "*** MP1762D/MP1764D EYE MARGIN SAMPLE PROGRAM ** "
170      PRINT
180      !
190      GOSUB D_set                        !DATA SETTING
200      GOSUB Clock                       !CHECK CLOCK LOSS
210      GOSUB Eye_mrgn                   !EYE MARGIN START
220      GOSUB Result                     !DISPLAY RESULT
230      !
240      INPUT "    Next data set[Yes:0, No:1]",Loop$
250      EXIT IF Loop$="1"
260      END LOOP
270      !
280      STOP
290      !
300  !***** MP1761B/C,MP1763B/C/MP1762D,MP1764D DATA SETTING *****
310  D_set: !
320      !MP1761B/C/MP1763B/C DATA SETTING
330      !
340      OUTPUT Add2;"CLK 1;RES 1;FRQ 12500"          !FREQUENCY
350      OUTPUT Add2;"OFS 0"                          !VOH
360      OUTPUT Add2;"DAP 1;DOS 1;DTM 0"              !DATA SET
370      OUTPUT Add2;"CDL 100;CAP 1;COS 1"            !CLOCK SET
380      OUTPUT Add2;"LGC 0;PTS 3;PTN 6;MRK 3;SFT 0"  !PATTERN
390      !
400      !MP1762D/MP1764D DATA SETTING
410      !
420      OUTPUT Add1;"DTM 0;CTM 0;CPL 0"              !INPUT
430      OUTPUT Add1;"LGC 0;PTS 3;PTN 6;MRK 3;SFT 0"  !PATTERN
440      OUTPUT Add1;"SRH 0;SYN 1;SYM 0"
450      PRINT " ** SELECT EYE MARGIN ERROR RATIO ** "
460      PRINT "      ERROR RATIO [ 0 to 7 ]          "
470      PRINT "      0:<=1.0E-2          4:<=1.0E-6          "
480      PRINT "      1:<=1.0E-3          5:<=1.0E-7          "
490      PRINT "      2:<=1.0E-4          6:<=1.0E-8          "
500      PRINT "      3:<=1.0E-5          7:<=1.0E-9          "
510      INPUT "      Select number of the ERROR RATIO = ",Ert$
520      OUTPUT Add1;"EME 1;EYT "&Ert$
530      !
540      RETURN
550      !
560      !
570      !***** Check Connection *****
580      Clock: !
590      !
600      LOOP
610      !

```

```

620      GOSUB Connect
630      !
640      OUTPUT Add1;"CLI?"
650      ENTER Add1;Cli$
660      IF Cli$="CLI 1" THEN
670          PRINT "***** CLOCK LOSS ***** "
680      END IF
690      EXIT IF Cli$="CLI 0"
700      END LOOP
710      !
720      RETURN
730      !
740      !
750      !***** EYE MARGIN FUNCTION *****
760      Eye_mrgn: !
770      !
780      OUTPUT Add1;"EST 1"
790      !
800      LOOP
810      !
820          OUTPUT Add1;"EST?"
830          ENTER Add1;Est$
840          !
850      EXIT IF Est$="EST 0" OR Est$="EST 2"
860      END LOOP
870      !
880      IF Est$="EST 0" THEN
890          PRINT "***** EYE MARGIN OK ***** "
900      ELSE
910          PRINT "***** Failed in EYE MARGIN ***** "
920      END IF
930      !
940      RETURN
950      !
960      !
970      !***** DISPLAY RESULT FUNCTION *****
980      Result: !
990      !
1000     OUTPUT Add1;"DTH?"
1010     ENTER Add1;Dth$
1020     !
1030     OUTPUT Add1;"THM?"
1040     ENTER Add1;Thm$
1050     IF Thm$="THM -9.999" THEN
1060         Thm$="THM No data "
1070     END IF
1080     !
1090     OUTPUT Add1;"DTM?"
1100     ENTER Add1;Dtm$
1110     IF Dtm$="DTM 0" THEN
1120         Dtm$="GND"
1130     ELSE
1140         Dtm$="-2V"
1150     END IF
1160     !
1170     OUTPUT Add1;"CPA?"
1180     ENTER Add1;Cpa$
1190     !
1200     OUTPUT Add1;"PHM?"
1210     ENTER Add1;Phm$
1220     IF Phm$="PHM -9999" THEN
1230         Phm$="PHM No data "
1240     END IF
1250     !
1260     OUTPUT Add1;"CTM?"
1270     ENTER Add1;Ctm$

```

```

1280 IF Ctm$="CTM 0" THEN
1290     Ctm$="GND"
1300 ELSE
1310     Ctm$="-2V"
1320 END IF
1330 !
1340 OUTPUT Add1,"CPL?"
1350 ENTER Add1,Cpl$
1360 IF Cpl$="CPL 0" THEN
1370     Cpl$="CLK"
1380 ELSE
1390     Cpl$="NCLK"
1400 END IF
1410 !
1420 PRINT "DATA THRESHOLD ="&Dth$[5,10]&" V"
1430 PRINT "THRESHOLD MARGIN = "&Thm$[5,10]&" Vp-p"
1440 PRINT "DATA TERMINATION = "&Dtm$
1450 PRINT "CLOCK PHASE ADJUST = "&Cpa$[6,9]&" ps"
1460 PRINT "PHASE MARGIN = "&Phm$[5,10]&" psp-p"
1470 PRINT "CLOCK TERMINATION = "&Ctm$
1480 PRINT "CLOCK POLARITY = "&Cpl$
1490 PRINT
1500 !
1510 RETURN
1520 !
1530 !
1540 !***** DISPLAY CONNECTION *****
1550 Connect: !
1560 !
1570 PEN 3
1580 VIEWPORT 70,140,50,100
1590 SHOW 0,70,0,50
1600 !
1610 CLIP 0,70,5,70
1620 FRAME
1630 !
1640 CSIZE 3,.4
1650 MOVE 25,45
1660 LABEL "<< CONNECTION >>"
1670 !
1680 CSIZE 3,.35
1690 MOVE 6,39
1700 LABEL " MP1761B/C/MP1763B/C          MP1762D/MP1764D"
1710 !
1720 MOVE 7,20
1730 RECTANGLE 25,18
1740 !
1750 MOVE 38,20
1760 RECTANGLE 25,18
1770 !
1780 MOVE 26,14
1790 IDRAW 0,9
1800 !
1810 FOR I=0 TO PI*2 STEP PI/12
1820     IDRAW .2*COS(I),.2*SIN(I)
1830 NEXT I
1840 !
1850 MOVE 26,14
1860 IDRAW 21,0
1870 IDRAW 0,9
1880 !
1890 FOR I=0 TO PI*2 STEP PI/12
1900     IDRAW .2*COS(I),.2*SIN(I)
1910 NEXT I
1920 !
1930 MOVE 21,17

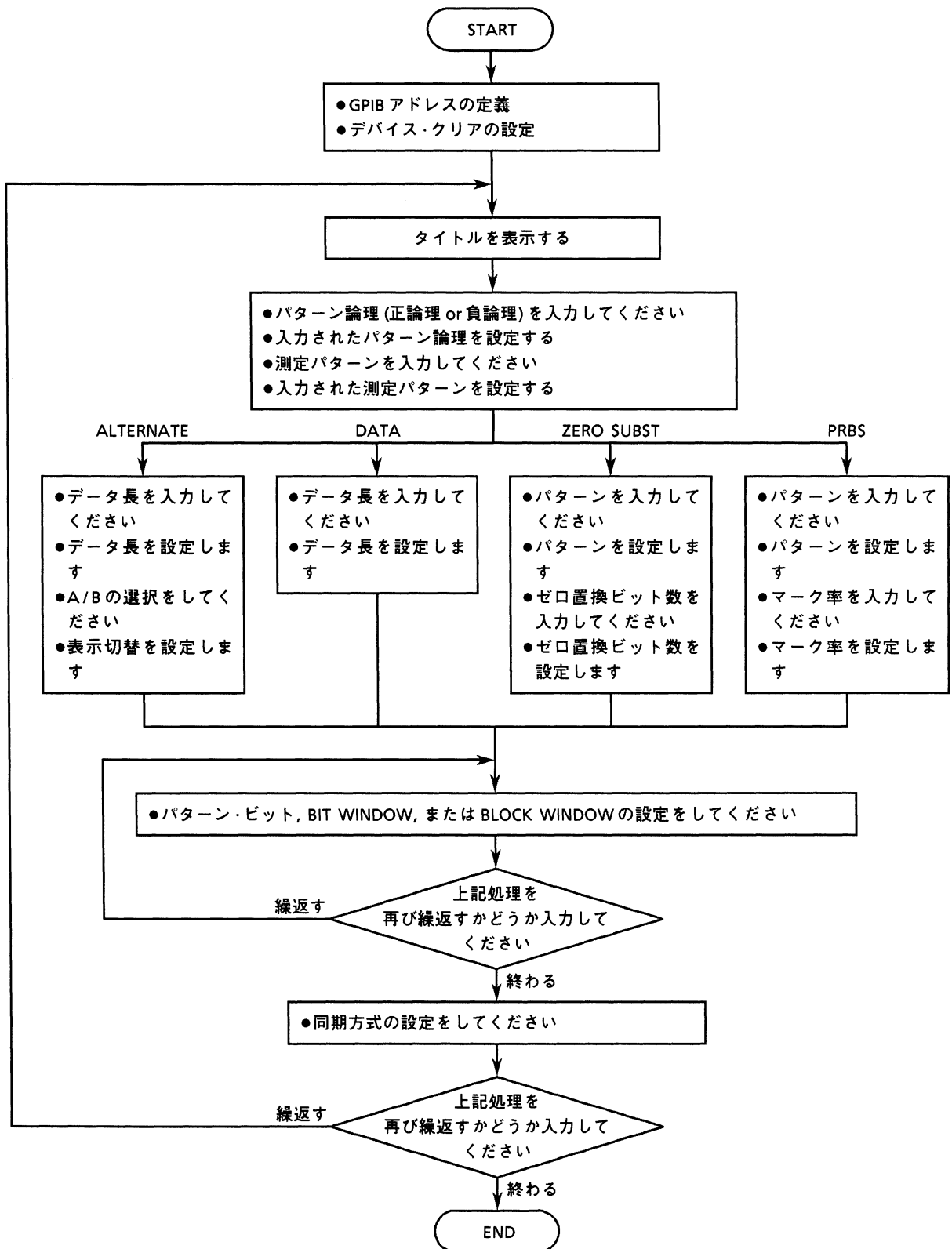
```

```
1940 IDRAW 0,6
1950 !
1960 FOR I=0 TO PI*2 STEP PI/12
1970     IDRAW .2*COS(I),.2*SIN(I)
1980 NEXT I
1990 !
2000 MOVE 21,17
2010 IDRAW 21,0
2020 IDRAW 0,6
2030 !
2040 FOR I=0 TO PI*2 STEP PI/12
2050     IDRAW .2*COS(I),.2*SIN(I)
2060 NEXT I
2070 !
2080 MOVE 16,25
2090 CSIZE 2.3,.5
2100 LABEL "DATA CLOCK1          DATA  CLOCK"
2110 !
2120 INPUT "Are you ready ? Press return key to start.",A
2130 !
2140 RETURN
2150 !
2160 !
2170 END
```

(4) 測定パターン・BIT WINDOW・BLOCK WINDOW の設定

本プログラムは測定パターン関係の GPIB 制御を行います。

まず始めに、パターン論理と測定パターンの選択を行います。選択したパターンに対する各種設定を行います。



● プログラム・リスト

```

10  !*****
20  !*
30  !*          MP1762D/MP1764D PATTERN SAMPLE PROGRAM          *
40  !*
50  !*****
60  !
70  Add=701
80  !
90  CLEAR Add
100 !
110 LOOP
120   CLEAR SCREEN
130   PRINT "*** MP1762D/MP1764D PATTERN SAMPLE PROGRAM ** "
140   PRINT
150   !
160   GOSUB Logic_set
170   GOSUB Pattern_set
180   !
190   INPUT "Next data set[Yes:0, No:1]",Loop$
200   EXIT IF Loop$="1"
210   END LOOP
220   !
230   STOP
240   !
250   !***** MP1762D/MP1764D LOGIC SETTING *****
260   Logic_set: !
270   !
280   LOOP
290       INPUT "Choose LOGIC [Positive:0, Negative:1]",Lgc$
300       IF Lgc$<>"0" AND Lgc$<>"1" THEN
310           PRINT "Wrong chosen number!! Please select a correct LOGIC"
320       END IF
330   EXIT IF Lgc$="0" OR Lgc$="1"
340   END LOOP
350   CLEAR SCREEN
360   OUTPUT Add;"LGC "&Lgc$
370   !
380   RETURN
390   !
400   !***** MP1762D/MP1764D PATTERN SETTING *****
410   Pattern_set: !
420   !
430   LOOP
440       INPUT "Choose measure PATTERN [Alternate:0, Data:1, Zero subst:2, PRBS
:31",Patt$
450       IF Patt$<>"0" AND Patt$<>"1" AND Patt$<>"2" AND Patt$<>"3" THEN
460           PRINT "Wrong chosen number!! Please select a correct PATTERN"
470       END IF
480   EXIT IF Patt$="0" OR Patt$="1" OR Patt$="1" OR Patt$="2" OR Patt$="3"
490   END LOOP
500   CLEAR SCREEN
510   OUTPUT Add;"PTS "&Patt$
520   !
530   SELECT Patt$
540       CASE "0"
550           LOOP
560               INPUT "Choose SYNC MODE [Normal:0, Frame:1]",Sync$
570               IF Sync$<>"0" AND Sync$<>"1" THEN
580                   PRINT "Wrong chosen number!! Please select a correct SYNC
MODE"
590               END IF

```

```

600         EXIT IF Sync$="0" OR Sync$="1"
610         END LOOP
620         CLEAR SCREEN
630         OUTPUT Add;"SYM "&Sync$
640         GOSUB Altn
650     CASE "1"
660     LOOP
670         INPUT "Choose SYNC MODE [Normal:0, Frame:1, Quick:2]",Sync$
680         IF Sync$<>"0" AND Sync$<>"1" AND Sync$<>"2" THEN
690             PRINT "Wrong chosen number!! Please select a correct SYNC
MODE"
700             END IF
710         EXIT IF Sync$="0" OR Sync$="1" OR Sync$="2"
720         END LOOP
730         CLEAR SCREEN
740         OUTPUT Add;"SYM "&Sync$
750         GOSUB Dat
760     CASE "2"
770     LOOP
780         INPUT "Choose SYNC MODE [Normal:0, Frame:1, Quick:2]",Sync$
790         IF Sync$<>"0" AND Sync$<>"1" AND Sync$<>"2" THEN
800             PRINT "Wrong chosen number!! Please select a correct SYNC
MODE"
810             END IF
820         EXIT IF Sync$="0" OR Sync$="1" OR Sync$="2"
830         END LOOP
840         CLEAR SCREEN
850         OUTPUT Add;"SYM "&Sync$
860         GOSUB Zer
870     CASE "3"
880         GOSUB Prbs
890         !
900     END SELECT
910     !
920     RETURN
930     !
940     !***** ALTERNATE PATTERN SETTING *****
950 Altn: !
960     LOOP
970         INPUT "Set alternate pattern length [128 to 4194304]",Alt_dln$
980         Alt_dln=VAL(Alt_dln$)
990         IF Alt_dln<128 OR Alt_dln>4194304 THEN
1000             PRINT "Wrong input ALTERNATE LENGTH!! Please set a correct number"
1010         END IF
1020     EXIT IF Alt_dln>=128 AND Alt_dln<=4194304
1030     END LOOP
1040     CLEAR SCREEN
1050     OUTPUT Add;"DLN "&Alt_dln$
1060     !
1070     LOOP
1080         INPUT "Choose ALTERNATE A or B [A:0, B:1] ",Alt_dsp$
1090         IF Alt_dsp$<>"0" AND Alt_dsp$<>"1" THEN
1100             PRINT "Wrong input ALTERNATE DISPLAY!! Please set a correct number
"
1110         END IF
1120     EXIT IF Alt_dsp$="0" OR Alt_dsp$="1"
1130     END LOOP
1140     CLEAR SCREEN
1150     OUTPUT Add;"ALT "&Alt_dsp$
1160     !
1170     LOOP
1180     LOOP
1190         INPUT "Which PATTERN do you set?[Pattern:0, Bit_window:1, Block_wi
ndow:2],Dsp_sel$
1200         IF Dsp_sel$<>"0" AND Dsp_sel$<>"1" AND Dsp_sel$<>"2" THEN
1210             PRINT "Wrong input setting DISPLAY SELECT!! Please set a corre

```

```

ct number"
1220      END IF
1230      EXIT IF Dsp_sel$="0" OR Dsp_sel$="1" OR Dsp_sel$="2"
1240      END LOOP
1250      CLEAR SCREEN
1260      OUTPUT Add;"DSP "&Dsp_sel$
1270      SELECT Dsp_sel$
1280      CASE "0"
1290          GOSUB Bit_patt
1300      CASE "1"
1310          GOSUB Bit_win
1320      CASE "2"
1330          GOSUB Block_win
1340      END SELECT
1350      INPUT "Do you wish to continue to set another ALTERNATE pattern?[Yes:0
, No:1]",Cont_alt$
1360      EXIT IF Cont_alt$="1"
1370      END LOOP
1380      CLEAR SCREEN
1390      RETURN
1400      !
1410      !***** DATA PATTERN SETTING *****
1420      Dat: !
1430      LOOP
1440          INPUT "Set data pattern length [2 TO 8388608]",Dat_dln$
1450          Dat_dln=VAL(Dat_dln$)
1460          IF Dat_dln<2 OR Dat_dln>8388608 THEN
1470              PRINT "Wrong input DATA LENGTH!! Please set a correct number"
1480          END IF
1490          EXIT IF Dat_dln>=2 AND Dat_dln<=8388608
1500      END LOOP
1510      CLEAR SCREEN
1520      OUTPUT Add;"DLN "&Dat_dln$
1530      !
1540      LOOP
1550          LOOP
1560              IF Dat_dln MOD 32=0 THEN
1570                  INPUT "Which PATTERN do you set?[Pattern:0, Bit_window:1, Bloc
k_window:2]",Dsp_sel$
1580                  Flg=0 !TRUE
1590                  IF Dsp_sel$<>"0" AND Dsp_sel$<>"1" AND Dsp_sel$<>"2" THEN
1600                      PRINT "Wrong input setting DISPLAY SELECT!! Please set a c
orrect number"
1610                      Flg=1 !FALSE
1620                  END IF
1630              ELSE
1640                  INPUT "Which PATTERN do you set?[Pattern:0, Bit_window:1]",Dsp
_sel$
1650                  Flg=0 !TRUE
1660                  IF Dsp_sel$<>"0" AND Dsp_sel$<>"1" THEN
1670                      PRINT "Wrong input setting DISPLAY SELECT!! Please set a c
orrect number"
1680                      Flg=1 !FALSE
1690                  END IF
1700              END IF
1710          EXIT IF Flg=0
1720          END LOOP
1730          CLEAR SCREEN
1740          OUTPUT Add;"DSP "&Dsp_sel$
1750          !
1760          SELECT Dsp_sel$
1770          CASE "0"
1780              GOSUB Bit_patt
1790          CASE "1"
1800              GOSUB Bit_win
1810          CASE "2"

```



```

1820          GOSUB Block_win
1830      END SELECT
1840      INPUT "Do you wish to continue to set another DATA pattern?[Yes:0, No:
1J],Cont_dat$
1850      EXIT IF Cont_dat$="1"
1860      END LOOP
1870      CLEAR SCREEN
1880      RETURN
1890      !
1900      !***** ZERO SUBST PATTERN SETTING *****
1910      Zer: !
1920      LOOP
1930          INPUT "Set zero subst pattren [2^7:0, 2^9:1, 2^11:2, 2^15:3]",Zsub_dan
$
1940          IF Zsub_dan$<>"0" AND Zsub_dan$<>"1" AND Zsub_dan$<>"2" AND Zsub_dan$<
>"3" THEN
1950              PRINT "Wrong input ZERO SUBST. PATTERN!! Please set a correct numb
er"
1960          END IF
1970          EXIT IF Zsub_dan$="0" OR Zsub_dan$="1" OR Zsub_dan$="2" OR Zsub_dan$="3"
1980          END LOOP
1990          CLEAR SCREEN
2000          SELECT Zsub_dan$
2010              CASE "0"
2020                  OUTPUT Add;"PTN 2"
2030                  LOOP
2040                      INPUT "Set zero subst length [1 to 127]",Zsub_len$
2050                      Zsub_len=VAL(Zsub_len$)
2060                      IF Zsub_len<1 OR Zsub_len>127 THEN
2070                          PRINT "Wrong input ZERO SUBST LENGTH!! Please set a correc
t number"
2080                      END IF
2090                      EXIT IF Zsub_len>=1 AND Zsub_len<=127
2100                      END LOOP
2110                      CLEAR SCREEN
2120                      OUTPUT Add;"ZLN "&Zsub_len$
2130                      !
2140                  CASE "1"
2150                      OUTPUT Add;"PTN 3"
2160                      LOOP
2170                          INPUT "Set zero subst length [1 to 511]",Zsub_len$
2180                          Zsub_len=VAL(Zsub_len$)
2190                          IF Zsub_len<1 OR Zsub_len>511 THEN
2200                              PRINT "Wrong input ZERO SUBST LENGTH!! Please set a correc
t number"
2210                          END IF
2220                          EXIT IF Zsub_len>=1 AND Zsub_len<=511
2230                          END LOOP
2240                          CLEAR SCREEN
2250                          OUTPUT Add;"ZLN "&Zsub_len$
2260                          !
2270                  CASE "2"
2280                      OUTPUT Add;"PTN 5"
2290                      LOOP
2300                          INPUT "Set zero subst length [1 to 2047]",Zsub_len$
2310                          Zsub_len=VAL(Zsub_len$)
2320                          IF Zsub_len<1 OR Zsub_len>2047 THEN
2330                              PRINT "Wrong input ZERO SUBST LENGTH!! Please set a correc
t number"
2340                          END IF
2350                          EXIT IF Zsub_len>=1 AND Zsub_len<=2047
2360                          END LOOP
2370                          CLEAR SCREEN
2380                          OUTPUT Add;"ZLN "&Zsub_len$
2390                          !
2400                  CASE "3"

```

```

2410      OUTPUT Add;"FTN 6"
2420      LOOP
2430          INPUT "Set zero subst length [1 to 32767]",Zsub_len$
2440          Zsub_len=VAL(Zsub_len$)
2450          IF Zsub_len<1 OR Zsub_len>32767 THEN
2460              PRINT "Wrong input ZERO SUBST LENGTH!! Please set a correc
t number"
2470          END IF
2480          EXIT IF Zsub_len>=1 AND Zsub_len<=32767
2490      END LOOP
2500      CLEAR SCREEN
2510      OUTPUT Add;"ZLN "&Zsub_len$
2520      !
2530  END SELECT
2540      !
2550  LOOP
2560      LOOP
2570          INPUT "Which PATTERN do you set?[Bit_window:1, Block_window:2]",Ds
p_sel$
2580          IF Dsp_sel$<>"1" AND Dsp_sel$<>"2" THEN
2590              PRINT "Wrong input setting DISPLAY SELECT!! Please set a corre
ct number"
2600          END IF
2610          EXIT IF Dsp_sel$="1" OR Dsp_sel$="2"
2620      END LOOP
2630      CLEAR SCREEN
2640      OUTPUT Add;"DSP "&Dsp_sel$
2650      !
2660      SELECT Dsp_sel$
2670          CASE "1"
2680              GOSUB Bit_win
2690          CASE "2"
2700              GOSUB Block_win
2710      END SELECT
2720      INPUT "Do you wish to continue to set another ZERO SUBST.pattern?[Yes:
0, No:1]",Cont_zsub$
2730      EXIT IF Cont_zsub$="1"
2740  END LOOP
2750      CLEAR SCREEN
2760      RETURN
2770      !
2780      !***** PRBS PATTERN SETTING *****
*
2790  Prbs: !
2800      LOOP
2810          LOOP
2820              PRINT "Select PRBS pattern [2^7-1:0, 2^9-1:1, 2^11-1:2, 2^15-1:3]"
2830              PRINT "                                [2^20-1:4, 2^23-1:5, 2^31-1:6
] "
2840              PRINT
2850              INPUT "Set PRBS pattren ",Prbs_dan$
2860              IF Prbs_dan$<>"0" AND Prbs_dan$<>"1" AND Prbs_dan$<>"2" AND Prbs_d
an$<>"3" AND Prbs_dan$<>"4" AND Prbs_dan$<>"5" AND Prbs_dan$<>"6" THEN
2870                  PRINT "Wrong input PRBS PATTERN!! Please set a correct number"
2880              END IF
2890              EXIT IF Prbs_dan$="0" OR Prbs_dan$="1" OR Prbs_dan$="2" OR Prbs_dan$="
3" OR Prbs_dan$="4" OR Prbs_dan$="5" OR Prbs_dan$="6"
2900          END LOOP
2910          CLEAR SCREEN
2920          SELECT Prbs_dan$
2930              CASE "0"
2940                  OUTPUT Add;"PTN 2"
2950              CASE "1"
2960                  OUTPUT Add;"PTN 3"
2970              CASE "2"
2980                  OUTPUT Add;"PTN 5"
2990              CASE "3"

```

```

3000         OUTPUT Add;"PTN 6"
3010     CASE "4"
3020         OUTPUT Add;"PTN 7"
3030     CASE "5"
3040         OUTPUT Add;"PTN 8"
3050     CASE "6"
3060         OUTPUT Add;"PTN 9"
3070 END SELECT
3080 !
3090 LOOP
3100     PRINT
3110     PRINT "Mark ratio (Positive)[0/8:0, 1/8:1, 1/4:2, 1/2:3]"
3120     PRINT "          (Negative)[8/8:0, 7/8:1, 3/4:2, 1/2:3]"
3130     PRINT
3140     INPUT "Choose MARK RATIO",Mrk$
3150     IF Mrk$<>"0" AND Mrk$<>"1" AND Mrk$<>"2" AND Mrk$<>"3" THEN
3160         PRINT "Wrong input setting PRBS MARK RATIO!! Please set a corre
ct number"
3170     END IF
3180     EXIT IF Mrk$="0" OR Mrk$="1" OR Mrk$="2" OR Mrk$="3"
3190 END LOOP
3200 CLEAR SCREEN
3210 OUTPUT Add;"MRK "&Mrk$
3220 !
3230 LOOP
3240     INPUT "Do you wish to set BIT-WINDOW PATTERN?[Yes:0, No:1]",Dsp_sel$
3250     IF Dsp_sel$<>"0" AND Dsp_sel$<>"1" THEN
3260         PRINT "Wrong input setting DISPLAY SELECT!! Please set a corre
ct number"
3270     END IF
3280     EXIT IF Dsp_sel$="0" OR Dsp_sel$="1"
3290 END LOOP
3300 CLEAR SCREEN
3310 !
3320 SELECT Dsp_sel$
3330     CASE "0"
3340         OUTPUT Add;"DSP 1"
3350         GOSUB Bit_win
3360     END SELECT
3370     INPUT "Do you wish to continue to set another PRBS pattern?[Yes:0, No:
1]",Cont_prbs$
3380     EXIT IF Cont_prbs$="1"
3390 END LOOP
3400 CLEAR SCREEN
3410 RETURN
3420 !
3430 !***** BIT PATTERN SETTING *****
3440 Bit_patt: !
3450 !
3460 DIM Bit$(255)
3470 LOOP
3480     PRINT "You are able to choice data format of HEXadecimal or DECimal"
3490     PRINT "Default data format is HEXdecimal"
3500     INPUT "Which do you choice format [HEXdecimal:0, DECimal:1]",Fmt$
3510     INPUT "Where do you set start page[1 to LENGTH/16]",Page$
3520     !
3530     Bit$=""
3540     A$=""
3550     FOR K=0 TO 7
3560         PRINT "<Do you set bit-pattern of = "&VAL$(VAL(Page$)+K)&" PAGE? [
Yes:0, No:1]"
3570         INPUT "",A$
3580         IF A$="1" THEN
3590             IF K=0 THEN
3600                 GOTO Jump out

```

```

3610         ELSE
3620             GOTO Jump
3630         END IF
3640     END IF
3650     PRINT
3660     IF K<>0 THEN
3670         Bit$=Bit$&","
3680     END IF
3690     IF Fmt$="1" THEN
3700         LOOP
3710             PRINT "Enter "&VAL$(VAL(Page$)+K)&"PAGE pattern [ 0 to 655
3720             INPUT B$
3730             B=VAL(B$)
3740             EXIT IF B>=0 AND B<=65535
3750             PRINT "Wrong number!!Please input a correct number"
3760         END LOOP
3770         CLEAR SCREEN
3780     ELSE
3790         LOOP
3800             PRINT "Enter "&VAL$(VAL(Page$)+K)&"PAGE pattern [ 0 to FFF
3810             INPUT B$
3820             B=DVAL(B$,16)
3830             B$="#H"&B$
3840             EXIT IF B>=0 AND B<=65535
3850             PRINT "Wrong number!!Please input a correct number"
3860         END LOOP
3870         CLEAR SCREEN
3880     END IF
3890     Bit$=Bit$&B$
3900 NEXT K
3910 Jump: !
3920     PRINT
3930     PRINT "You set data is :PAG "&Page$&"; BIT "&Bit$
3940     OUTPUT Add;"PAG "&Page$&";BIT "&Bit$
3950     !
3960 Jump_out: !
3970     INPUT "Do you want to continue setting BIT-PATTERN? [Yes:0, No:1]",Loop$
3980     EXIT IF Loop$="1"
3990     CLEAR SCREEN
4000 END LOOP
4010 CLEAR SCREEN
4020 RETURN
4030 !
4040 !***** BIT WINDOW SETTING *****
4050 Bit_win: !
4060 !
4070 DIM Bit_win$(255)
4080 LOOP
4090     PRINT "You are able to choice data format of HEXadecimal or DECimal"
4100     PRINT "Default data format is HEXdecimal"
4110     INPUT "Which do you choice format [HEXdecimal:0, DECimal:1]",Fmt$
4120     INPUT "Where do you set start page[1 to 21]",Page$
4130     IF Page$<>"1" AND Page$<>"2" THEN
4140         GOTO Jump1
4150     END IF
4160     !
4170     Bit_win$=""
4180     A$="0"
4190     FOR K=0 TO 1
4200         IF VAL(Page$)+K>2 THEN
4210             GOTO Jump1
4220         END IF
4230         PRINT "<Do you set bit-WINDOW-pattern of = "&VAL$(VAL(Page$)+K)&"

```

```

PAGE? [Yes:0, No:1]"
4240      INPUT A$
4250      IF A$="1" THEN
4260          IF K=0 THEN
4270              GOTO Jump_out1
4280          ELSE
4290              GOTO Jump1
4300          END IF
4310      END IF
4320      PRINT
4330      IF K<>0 THEN
4340          Bit_win$=Bit_win$&","
4350      END IF
4360      IF Fmt$="1" THEN
4370          LOOP
4380              PRINT "Enter "&VAL$(VAL(Page$)+K)&"PAGE BIT-WINDOW pattern
[ 0 to 65535 ]"
4390              INPUT B$
4400              B=VAL(B$)
4410              PRINT B
4420              EXIT IF B>=0 OR B<=65536
4430              PRINT "Wrong number!!Please input a correct number"
4440          END LOOP
4450          CLEAR SCREEN
4460      ELSE
4470          LOOP
4480              PRINT "Enter "&VAL$(VAL(Page$)+K)&"PAGE BIT-WINDOW pattern
[ 0 to FFFF ]"
4490              INPUT B$
4500              PRINT B$
4510              B=DVAL(B$,16)
4520              PRINT B
4530              B$="#H"&B$
4540              EXIT IF B>=0 AND B<=65535
4550          END LOOP
4560          CLEAR SCREEN
4570      END IF
4580      Bit_win$=Bit_win$&B$
4590      NEXT K
4600      Jump1: !
4610      PRINT
4620      PRINT "You set data is :MSK "&Page$&"; CHM "&Bit_win$
4630      OUTPUT Add;"MSK "&Page$&";CHM "&Bit_win$
4640      !
4650      Jump_out1: !
4660      LOOP
4670          INPUT "Do you wish to set BIT-WINDOW ON/OFF[OFF:0, ON:1]",Ena$
4680          EXIT IF Ena$="0" OR Ena$="1"
4690          PRINT "Wrong number!!Please input a correct number"
4700      END LOOP
4710      CLEAR SCREEN
4720      OUTPUT Add;"MSE "&Ena$
4730      INPUT "Do you want to continue setting BIT-WINDOW-PATTERN? [Yes:0, No:
1]",Loop$
4740      EXIT IF Loop$="1"
4750      CLEAR SCREEN
4760      END LOOP
4770      CLEAR SCREEN
4780      RETURN
4790      !
4800      !***** BLOCK WINDOW SETTING *****
*****
4810      Block_win: !
4820      !
4830      DIM Block_win$(255)
4840      LOOP

```

```

4850 PRINT "You aer able to choice data format of HEXadecimal or DECimal"
4860 PRINT "Default data format is HEXdecimal"
4870 INPUT "Which do you choice foramt [HEXdecimal:0, DECimal:1]",Fmt$
4880 INPUT "Where do you set start page[1 to LENGTH/16]",Page$
4890 !
4900 Block_win$=""
4910 A$="0"
4920 FOR K=0 TO 7
4930 PRINT "<Do you set BLOCK-WINDOW-pattern of = "&VAL$(VAL(Page$)+K)&
" PAGE? [Yes:0, No:1]"
4940 INPUT A$
4950 IF A$="1" THEN
4960 IF K=0 THEN
4970 GOTO Jump_out2
4980 ELSE
4990 GOTO Jump2
5000 END IF
5010 END IF
5020 PRINT
5030 IF K<>0 THEN
5040 Block_win$=Block_win$&","
5050 END IF
5060 IF Fmt$="1" THEN
5070 LOOP
5080 PRINT "Enter "&VAL$(VAL(Page$)+K)&"PAGE BLOCK-WINDOW patte
rn [ 0 to 65535 ]"
5090 INPUT B$
5100 B=VAL(B$)
5110 EXIT IF B<0 OR B>65535
5120 PRINT "Wrong number!!Please input a correct number"
5130 END LOOP
5140 CLEAR SCREEN
5150 ELSE
5160 LOOP
5170 PRINT "Enter "&VAL$(VAL(Page$)+K)&"PAGE BLOCK-WINDOW patte
rn [ 0 to FFFF ]"
5180 INPUT B$
5190 B=DVAL(B$,16)
5200 B$="#H"&B$
5210 EXIT IF B>=0 AND B<=65535
5220 END LOOP
5230 CLEAR SCREEN
5240 END IF
5250 Block_win$=Block_win$&B$
5260 NEXT K
5270 Jump2: !
5280 PRINT
5290 PRINT "You set data is :PAG "&Page$&"; MGB "&Block_win$
5300 OUTPUT Add;"PAG "&Page$&";MGB "&Block_win$
5310 !
5320 Jump_out2: !
5330 LOOP
5340 INPUT "Do you wish to set BLOCK-WINDOW ON/OFF[OFF:0, ON:1]",Ena$
5350 EXIT IF Ena$="0" OR Ena$="1"
5360 PRINT "Wrong number!!Please input a correct number"
5370 END LOOP
5380 CLEAR SCREEN
5390 INPUT "Do you want to continue setting BLOCK-WINDOW-PATTERN? [Yes:0, N
o:1]",Loop$
5400 EXIT IF Loop$="1"
5410 END LOOP
5420 CLEAR SCREEN
5430 RETURN
5440 !
5450 PAUSE
5460 END

```

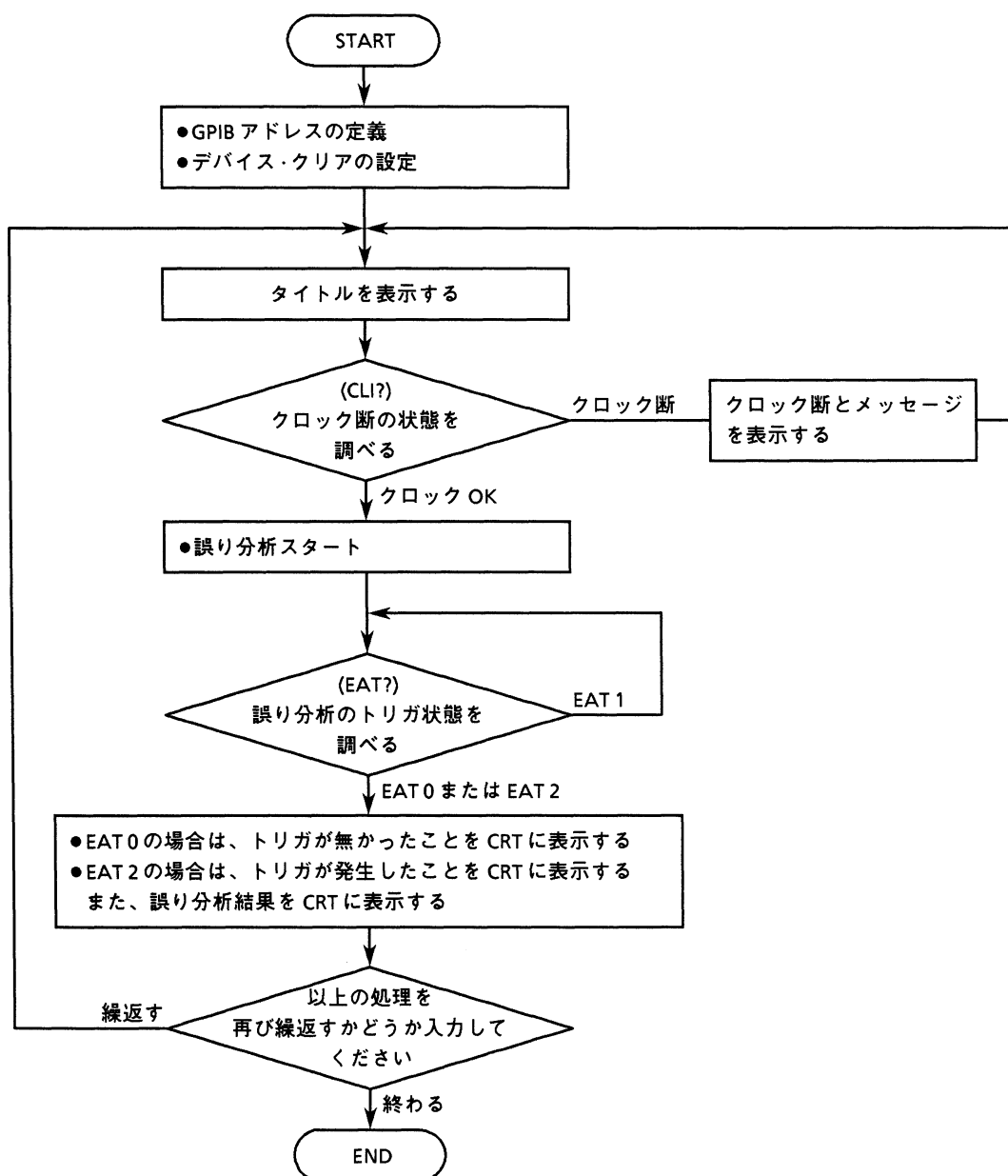
(5) 誤り分析

本プログラムは、誤り分析の **GPIB** 制御を行います。

誤り分析は、クロック断の時に無効なので、クロック断でないことを確認して、誤り分析をスタートします。

そして、リクエスト・コマンド (**EAT?**) で誤り分析中かどうかを読み出し、**CRT** に表示します。

また、誤り分析終了後、結果を **CRT** に表示します。



● プログラム・リスト

```

10  !*****
20  !*
30  !*          MP1762D/MP1764D ERROR ANALYSIS PROGRAM          *
40  !*
50  !*****
60  !
70  Add=701
80  CLEAR Add
90  !
100 LOOP
110  !
120  CLEAR SCREEN
130  PRINT "** MP1762D/MP1764D ERROR ANALYSIS SAMPLE PROGRAM ** "
140  PRINT
150  !
160  GOSUB Clock
170  GOSUB Trig
180  GOSUB Result
190  !
200  INPUT " Try again?[Yes:0, No:1]",Loop$
210  EXIT IF Loop$="1"
220  END LOOP
230  !
240  STOP
250  !
260  !***** Check Clock loss *****
270 Clock: !
280  !
290  LOOP
300  !
310  OUTPUT Add;"CLI?"
320  ENTER Add;Cli$
330  IF Cli$="CLI 1" THEN
340  PRINT "***** CLOCK LOSS ***** "
350  WAIT .5
360  CLEAR SCREEN
370  END IF
380  EXIT IF Cli$="CLI 0"
390  END LOOP
400  !
410  RETURN
420  !
430  !
440  !***** ERROR ANALYSIS TRIGGER *****
450 Trig: !
460  !
470  OUTPUT Add;"EAT 1"
480  !
490  LOOP
500  !
510  OUTPUT Add;"EAT?"
520  ENTER Add;Eat$
530  !
540  IF Eat$="EAT 1" THEN
550  PRINT "AWITTING TRIGGER!!"
560  WAIT .5
570  CLEAR SCREEN
580  END IF
590  EXIT IF Eat$="EAT 0" OR Eat$="EAT 2"
600  END LOOP
610  !

```



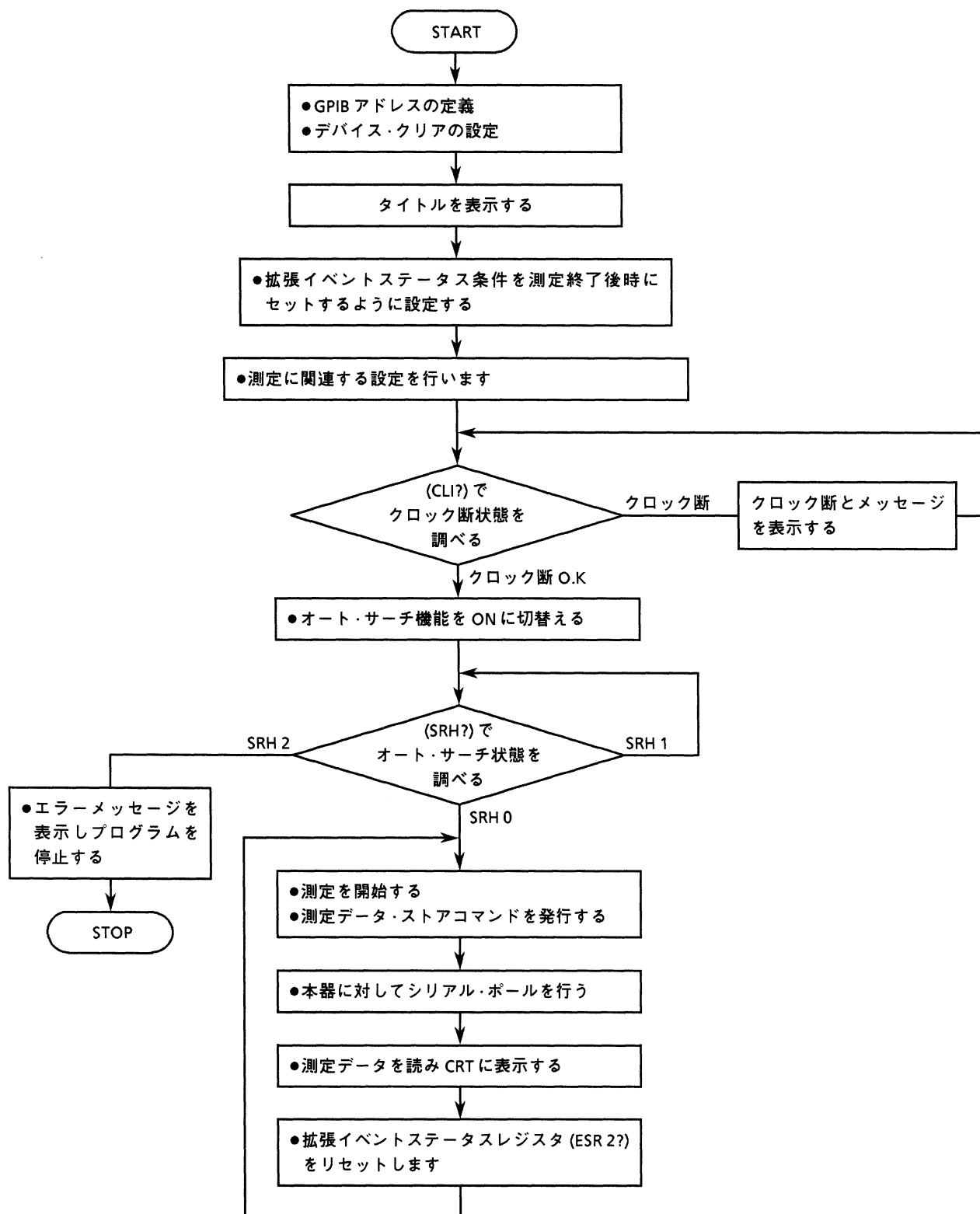
```

620 IF Eat$="EAT 0" THEN
630     PRINT "***** NO TRRIGER FOUND ***** "
640 ELSE
650     PRINT "***** TRIGGERD ***** "
660 END IF
670 !
680 RETURN
690 !
700 !
710 !***** DISPLAY RESULT FUNCTION *****
720 Result: !
730 !
740 DIM Ana$(16)[255]
750 CSIZE 3,.7
760 MOVE 20,90
770 LABEL "<< ERROR ANALYSIS RESULT >> "
780 !
790 CSIZE 3,.5
800 MOVE 0,85
810 LABEL "<< PAGE >>"
820 !
830 FOR J=1 TO 16
840     CSIZE 3,.5
850     MOVE 0,(80-J*4)
860     LABEL " "&VAL$(J)
870 NEXT J
880 !
890 CSIZE 3,.5
900 MOVE 20,85
910 LABEL "<< MONITOR >>"
920 !
930 CSIZE 3,.5
940 MOVE 50,85
950 LABEL "<< ERROR ANALYSIS DATA >>"
960 !
970 CSIZE 3,.5
980 MOVE 50,80
990 LABEL " BIT16 BIT1"
1000 !
1010 FOR I=1 TO 16
1020     OUTPUT Add;"EAP "&VAL$(I)
1030     OUTPUT Add;"EAB?"
1040     ENTER Add;Ana$(I)
1050     CSIZE 3,.5
1060     MOVE 20,(80-I*4)
1070     LABEL Ana$(I)[12,20]
1080     !
1090     CSIZE 3,.7
1100     MOVE 52,(80-I*4)
1110     LABEL Ana$(I)[24,41]
1120     !
1130 NEXT I
1140 !
1150 RETURN
1160 !
1170 !
1180 END

```

(6) 測定結果の表示 (シリアル・ポールで測定結果を表示)

本プログラムは、測定終了データをシリアル・ポールを使用して **CRT** に表示します。



● プログラム・リスト

```

10  !*****
20  !*
30  !*          MP1762D/MP1764D MEASUREMENT RESULT SAMPLE PROGRAM          *
40  !*
50  !*****
60  !
70  Add=701                      !MP1762D/MP1764D ADDRESS
80  CLEAR Add                    !DEVICE CLEAR(ED)
90  !
100 LOOP
110 !
120   CLEAR SCREEN
130   PRINT "*** MP1762D/MP1764D MEASUREMENT RESULT SAMPLE PROGRAM ** "
140   PRINT
150   !
160   OUTPUT Add;"ST0"
170   GOSUB Status_set           !STATUS RESISTOR SET
180   GOSUB Meas_cnd            !MEAS.CONDITION SET
190   GOSUB Closs               !CHECK CLOCK LOSS
200   GOSUB Auto_srh           !AUTO SEARCH ON
210   GOSUB Polling            !SERIAL POLE
220   !
230   INPUT "    Try again?[Yes:0, No:1]",Loop$
240   EXIT IF Loop$="1"
250   END LOOP
260   !
270   STOP
280   !
290   !***** STATUS SETTING *****
300   Status_set: !
310   !
320   OUTPUT Add;"*SRE 4"       !ESR2 ENABLE
330   OUTPUT Add;"ESE2 1"      !MEAS. END
340   !
350   RETURN
360   !
370   !***** MEASUREMENT CONDITION SET *****
380   Meas_cnd: !
390   !
400   DIM Prd$(255)
410   LOOP
420   !
430   LOOP
440       INPUT "MEAS.MODE?[Repeat:0, Single:1,Untimed:2]",M_mode$
450       EXIT IF M_mode$="0" OR M_mode$="1" OR M_mode$="2"
460       PRINT "Wrong chosen number!! Please select a correct MEAS.MODE"
470   END LOOP
480   OUTPUT Add;"MOD "&M_mode$
490   CLEAR SCREEN
500   !
510   IF M_mode$<>"2" THEN
520       INPUT "MEAS.TIME=[DAY,HOUR,MINUTE,SECOND]",Prd1$,Prd2$,Prd3$,Prd4$
530   END IF
540   OUTPUT Add;"FRD "&Prd1$&","&Prd2$&","&Prd3$&","&Prd4$
550   !CLEAR SCREEN
560   !
570   LOOP
580       INPUT "AUTO SYNC CONDITION=[OFF:0, ON:1]",Auto_sync$
590       EXIT IF Auto_sync$="0" OR Auto_sync$="1"
600       PRINT "Wrong chosen number!! Please select a correct AUTO SYNC CON
DITON"

```

```

610      END LOOP
620      OUTPUT Add;"SYN "&Auto_sync$
630      CLEAR SCREEN
640      !
650      IF Auto_sync$="1" THEN
660          LOOP
670              PRINT "SYNC THRESHOLD=[1E-2:0, 1E-3:1, 1E-4:2]"
680              PRINT "                [1E-5:3, 1E-6:4, 1E-7:5]"
690              PRINT "                [1E-8:6,          , INT :8]"
700              PRINT
710              INPUT Sync_th$
720              EXIT IF Sync_th$="0" OR Sync_th$="1" OR Sync_th$="2" OR Sync_th$="
3" OR Sync_th$="4" OR Sync_th$="5" OR Sync_th$="6" OR Sync_th$="8"
730              PRINT "Wrong chosen number!! Please select a correct SYNC THRE
SHOLD"
740          END LOOP
750          OUTPUT Add;"SYE "&Sync_th$
760          CLEAR SCREEN
770          END IF
780          INPUT "Do you change meas.condition?[Yes:0, No:1],M_cond$
790          EXIT IF M_cond$="1"
800      END LOOP
810      CLEAR SCREEN
820      RETURN
830      !
840      !
850      !***** CHECK CLOCK LOSS *****
860      Closs:
870      !
880      LOOP
890          OUTPUT Add;"CLI?"
900          ENTER Add;Cli$
910          IF Cli$="CLI 1" THEN
920              PRINT "** CLOCK LOSS **"
930              WAIT .5
940          END IF
950          CLEAR SCREEN
960          EXIT IF Cli$="CLI 0"
970      END LOOP
980      !
990      RETURN
1000     !
1010     !***** AUTO SEARCH FUNCTION *****
1020     Auto_srh:
1030     !
1040     OUTPUT Add;"SRH 1"
1050     !
1060     LOOP
1070         OUTPUT Add;"SRH?"
1080         ENTER Add;Srh$
1090         IF Srh$="SRH 1" THEN
1100             PRINT "** SEARCHING **"
1110             WAIT .5
1120         END IF
1130         CLEAR SCREEN
1140         EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1150     END LOOP
1160     !
1170     IF Srh$="SRH 2" THEN
1180         PRINT "Failed in auto search!! Program STOP!!"
1190         STOP
1200     END IF
1210     RETURN
1220     !
1230     !
1240     !***** POLLING *****

```

```

1250 Polling: !
1260 OUTPUT Add; "MOD?"
1270 ENTER Add; Mod$
1280 IF Mod$="MOD 0" THEN
1290     OUTPUT Add; "ESR2?"
1300     ENTER Add; Esr2$
1310     OUTPUT Add; "STA"
1320 END IF
1330 !
1340 LOOP
1350     IF Mod$<>"MOD 0" THEN
1360         OUTPUT Add; "STA"
1370     END IF
1380     !
1390     OUTPUT Add; "EDS"
1400     WAIT .1
1410     LOOP
1420         A=SPOLL(Add)
1430     EXIT IF BIT(A,2)=1
1440     WAIT .1
1450     END LOOP
1460     GOSUB Result_read
1470     OUTPUT Add; "ESR2?"
1480     ENTER Add; Esr2$
1490 END LOOP
1500 !
1510 RETURN
1520 !
1530 !***** DISPLAY RESULT FUNCTION *****
1540 Result_read: !
1550 !
1560 DIM Tme$[255], Arm$(3)[255], Err$(4)[255]
1570 DIM Arm_dat$[255], Err_dat$[255]
1580 !
1590 OUTPUT Add; "END? 0,0"
1600 ENTER Add; Tme$
1610 OUTPUT Add; "END? 1,0"
1620 ENTER Add; Arm_dat$
1630 OUTPUT Add; "END? 2,0"
1640 ENTER Add; Err_dat$
1650 Arm$(1)=" "
1660 Arm$(2)=" "
1670 Arm$(3)=" "
1680 K=1
1690 IF Arm_dat$<>"ERR" THEN
1700     Max_len=LEN(Arm_dat$)
1710     FOR J=1 TO 3
1720         LOOP
1730             EXIT IF K=(Max_len+1) OR Arm_dat$[K,K]=","
1740             Arm$(J)=Arm$(J)&Arm_dat$[K,K]
1750             K=K+1
1760         END LOOP
1770         K=K+1
1780     NEXT J
1790 ELSE
1800     FOR J=1 TO 3
1810         Arm$(J)=" NO DATA "
1820     NEXT J
1830 END IF
1840 !
1850 Err$(1)=" "
1860 Err$(2)=" "
1870 Err$(3)=" "
1880 Err$(4)=" "
1890 L=1
1900 IF Err_dat$<>"ERR" THEN

```

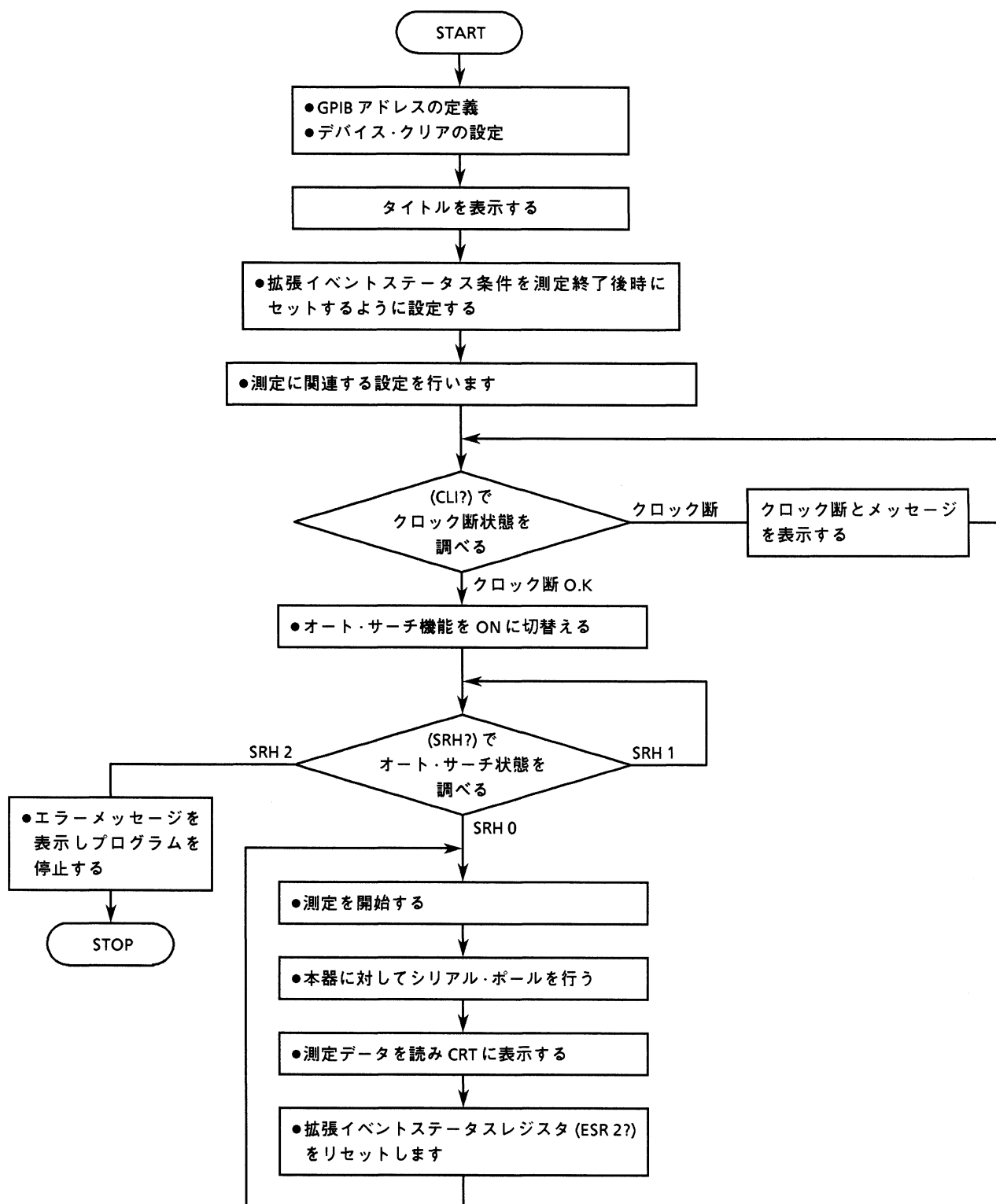
```

1910      Max_len2=LEN(Err_dat$)
1920      FOR M=1 TO 4
1930          LOOP
1940              EXIT IF Err_dat$[L,L]="," OR L=(Max_len2+1)
1950                  Err$(M)=Err$(M)&Err_dat$[L,L]
1960                  L=L+1
1970          END LOOP
1980          L=L+1
1990      NEXT M
2000  ELSE
2010      FOR M=1 TO 4
2020          Err$(M)=" NO DATA "
2030      NEXT M
2040  END IF
2050  !
2060  !
2070  IF Tme$="ERR" THEN
2080      GOTO Jump
2090  END IF
2100  !
2110  PRINT " << START TIME >>    "&Tme$[1,17]&" << STOP TIME >>    "&Tme$[19,35]
2120  !
2130  PRINT " << ARLAM DATA >>  "
2140  PRINT " << POWER FAIL INTVL>> "&Arm$(1)
2150  PRINT " << CLOCK LOSS INTVL>> "&Arm$(2)
2160  PRINT " << SYNC LOSS INTVL >> "&Arm$(3)
2170  !
2180  PRINT " << ERROR DATA >>"
2190  PRINT " << ERROR RATIO >>    "&Err$(1)
2200  PRINT " << ERROR COUNT    >> "&Err$(2)
2210  PRINT " << EI              >> "&Err$(3)
2220  PRINT " << %EFI             >>  "&Err$(4)
2230  !
2240  PRINT
2250  Jump: !
2260  RETURN
2270  !
2280  !
2290  END

```

(7) 測定結果の表示 (リクエスト・コマンドで測定結果を表示)

本プログラムは、測定終了データをリクエスト・コマンドを使用して **CRT** に表示します。



● プログラム・リスト

```

10  !*****
20  !*
30  !*          MP1762D/MP1764D MEASUREMENT RESULT SAMPLE PROGRAM          *
40  !*
50  !*****
60  !
70  Add=701                      !MP1762D/MP1764D ADDRESS
80  CLEAR Add                    !DEVICE CLEAR(ED)
90  !
100 LOOP
110 !
120   CLEAR SCREEN
130   PRINT "*** MP1762D/MP1764D MEASUREMENT RESULT SAMPLE PROGRAM ** "
140   PRINT
150   !
160   OUTPUT Add;"ST0"
170   GOSUB Status_set           !STATUS RESISTOR SET
180   GOSUB Meas_cnd             !MEAS.CONDITION SET
190   GOSUB Closs                !CHECK CLOCK LOSS
200   GOSUB Auto_srh             !AUTO SEARCH ON
210   GOSUB Polling              !SERIAL POLE
220   !
230   INPUT "    Try again?[Yes:0, No:1]",Loop$
240   EXIT IF Loop$="1"
250   END LOOP
260   !
270   STOP
280   !
290   !***** STATUS SETTING *****
300 Status_set: !
310   !
320   OUTPUT Add;"*SRE 4"        !ESR2 ENABLE
330   OUTPUT Add;"ESE2 1"        !MEAS. END
340   !
350   RETURN
360   !
370   !***** MEASUREMENT CONDITION SET *****
380 Meas_cnd: !
390   !
400   DIM Prd$(255)
410   LOOP
420   !
430   LOOP
440       INPUT "MEAS.MODE?[Repeat:0, Single:1,Untimed:2]",M_mode$
450       EXIT IF M_mode$="0" OR M_mode$="1" OR M_mode$="2"
460       PRINT "Wrong chosen number!! Please select a correct MEAS.MODE"
470   END LOOP
480   OUTPUT Add;"MOD "&M_mode$
490   CLEAR SCREEN
500   !
510   IF M_mode$<>"2" THEN
520       INPUT "MEAS.TIME=[DAY,HOUR,MINUTE,SECOND]",Prd1$,Prd2$,Prd3$,Prd4$
530   END IF
540   OUTPUT Add;"PRD "&Prd1$&","&Prd2$&","&Prd3$&","&Prd4$
550   !CLEAR SCREEN
560   !
570   LOOP
580       INPUT "AUTO SYNC CONDITION=[OFF:0, ON:1]",Auto_sync$
590       EXIT IF Auto_sync$="0" OR Auto_sync$="1"
600       PRINT "Wrong chosen number!! Please select a correct AUTO SYNC CON
DITON"

```



```

610     END LOOP
620     OUTPUT Add;"SYN "&Auto_sync$
630     CLEAR SCREEN
640     !
650     IF Auto_sync$="1" THEN
660         LOOP
670             PRINT "SYNC THRESHOLD=[1E-2:0, 1E-3:1, 1E-4:2]"
680             PRINT "                [1E-5:3, 1E-6:4, 1E-7:5]"
690             PRINT "                [1E-8:6,          , INT :8]"
700             PRINT
710             INPUT Sync_th$
720             EXIT IF Sync_th$="0" OR Sync_th$="1" OR Sync_th$="2" OR Sync_th$="
3" OR Sync_th$="4" OR Sync_th$="5" OR Sync_th$="6" OR Sync_th$="8"
730             PRINT "Wrong chosen number!! Please select a correct SYNC THRE
SHOLD"
740         END LOOP
750         OUTPUT Add;"SYE "&Sync_th$
760         CLEAR SCREEN
770         END IF
780         INPUT "Do you change meas.condition?[Yes:0, No:1]",M_cond$
790         EXIT IF M_cond$="1"
800     END LOOP
810     CLEAR SCREEN
820     RETURN
830     !
840     !
850     !***** CHECK CLOCK LOSS *****
860     Closs:!
870     !
880     LOOP
890         OUTPUT Add;"CLI?"
900         ENTER Add;Cli$
910         IF Cli$="CLI 1" THEN
920             PRINT "** CLOCK LOSS **"
930             WAIT .5
940         END IF
950         CLEAR SCREEN
960         EXIT IF Cli$="CLI 0"
970     END LOOP
980     !
990     RETURN
1000    !
1010    !***** AUTO SEARCH FUNCTION *****
1020    Auto_srh:!
1030    !
1040    OUTPUT Add;"SRH 1"
1050    !
1060    LOOP
1070        OUTPUT Add;"SRH?"
1080        ENTER Add;Srh$
1090        IF Srh$="SRH 1" THEN
1100            PRINT "** SEARCHING **"
1110            WAIT .5
1120        END IF
1130        CLEAR SCREEN
1140        EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1150    END LOOP
1160    !
1170    IF Srh$="SRH 2" THEN
1180        PRINT "Failed in auto search!! Program STOP!!"
1190        STOP
1200    END IF
1210    RETURN
1220    !
1230    !
1240    !***** POLLING *****

```

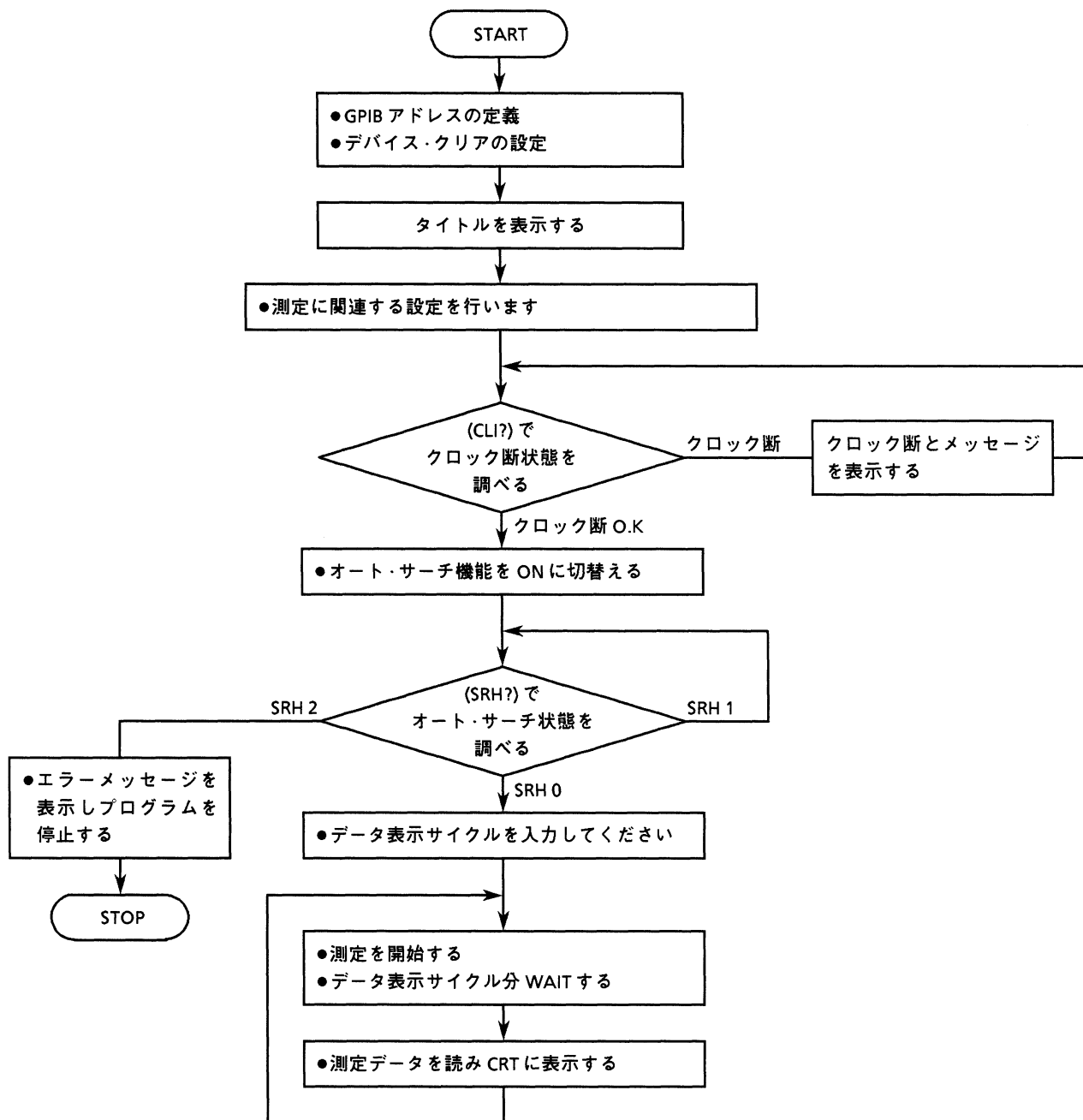
```

1250 Polling: !
1260 OUTPUT Add; "MOD?"
1270 ENTER Add; Mod$
1280 IF Mod$="MOD 0" THEN
1290     OUTPUT Add; "ESR2?"
1300     ENTER Add; ESR2$
1310     OUTPUT Add; "STA"
1320 END IF
1330 !
1340 LOOP
1350     IF Mod$<>"MOD 0" THEN
1360         OUTPUT Add; "STA"
1370     END IF
1380     !
1390     OUTPUT Add; "EDS"
1400     WAIT .1
1410     LOOP
1420         A=SPOLL(Add)
1430         EXIT IF BIT(A,2)=1
1440         WAIT .1
1450     END LOOP
1460     GOSUB Result_read
1470     OUTPUT Add; "ESR2?"
1480     ENTER Add; ESR2$
1490 END LOOP
1500 !
1510 RETURN
1520 !
1530 !***** DISPLAY RESULT FUNCTION *****
1540 Result_read: !
1550 !
1560 DIM Err$(4)[255]
1570 !
1580 OUTPUT Add; "ER?"
1590 ENTER Add; Err$(1)
1600 OUTPUT Add; "EC?"
1610 ENTER Add; Err$(2)
1620 OUTPUT Add; "EI?"
1630 ENTER Add; Err$(3)
1640 OUTPUT Add; "EFI?"
1650 ENTER Add; Err$(4)
1660 !
1670 PRINT " << ERROR DATA >>"
1680 PRINT " << ERROR RATIO >>      "&Err$(1)
1690 PRINT " << ERROR COUNT      >> "&Err$(2)
1700 PRINT " << EI              >> "&Err$(3)
1710 PRINT " << %EFI            >> "&Err$(4)
1720 !
1730 PRINT
1740 RETURN
1750 !
1760 !
1770 END

```

(8) 測定中間データの表示

本プログラムは、測定中間データを CRT に表示します。



● プログラム・リスト

```

10  !*****
20  !*
30  !*      MP1762D/MP1764D MEASUREMENT INTERMEDIATE DATA SAMPLE PROGRAM
40  !*
50  !*****
60  !
70  Add=701                      !MP1762D/MP1764D ADDRESS
80  CLEAR Add                    !DEVICE CLEAR(ED)
90  !
100 LOOP
110 !
120  CLEAR SCREEN
130  PRINT "*** MP1762D/MP1764D MEASUREMENT INTERMEDIATE DATA SAMPLE PROGRAM
140  PRINT
150  !
160  OUTPUT Add;"STO"
170  GOSUB Meas_cnd                !MEAS.CONDITION SET
180  GOSUB Closs                  !CHECK CLOCK LOSS
190  GOSUB Auto_srh               !AUTO SEARCH ON
200  GOSUB Int_dat                !INTERMEDIATE DATA
210  !
220  INPUT " Try again?[Yes:0, No:1]",Loop$
230  EXIT IF Loop$="1"
240  END LOOP
250  !
260  STOP
270  !
280  !***** MEASUREMENT CONDITION SET *****
290  Meas_cnd: !
300  !
310  DIM Prd$(255)
320  LOOP
330  !
340  LOOP
350  INPUT "MEAS.MODE?[Repeat:0, Single:1,Untimed:2]",M_mode$
360  EXIT IF M_mode$="0" OR M_mode$="1" OR M_mode$="2"
370  PRINT "Wrong chosen number!! Please select a correct MEAS.MODE"
380  END LOOP
390  OUTPUT Add;"MOD "&M_mode$
400  CLEAR SCREEN
410  !
420  IF M_mode$<>"2" THEN
430  INPUT "MEAS.TIME=[DAY,HOUR,MINUTE,SECOND]",Prd1$,Prd2$,Prd3$,Prd4$
440  END IF
450  OUTPUT Add;"PRD "&Prd1$&","&Prd2$&","&Prd3$&","&Prd4$
460  !LEAR SCREEN
470  !
480  LOOP
490  INPUT "AUTO SYNC CONDITION=[OFF:0, ON:1]",Auto_sync$
500  EXIT IF Auto_sync$="0" OR Auto_sync$="1"
510  PRINT "Wrong chosen number!! Please select a correct AUTO SYNC CON
DITON"
520  END LOOP
530  OUTPUT Add;"SYN "&Auto_sync$
540  CLEAR SCREEN
550  !
560  IF Auto_sync$="1" THEN
570  LOOP
580  PRINT "SYNC THRESHOLD=[1E-2:0, 1E-3:1, 1E-4:2]"
590  PRINT " [1E-5:3, 1E-6:4, 1E-7:5]"

```

```

600          PRINT "          [IE-8:6,          , INT :8]"
610          PRINT
620          INPUT Sync_th$
630          EXIT IF Sync_th$="0" OR Sync_th$="1" OR Sync_th$="2" OR Sync_th$="
3" OR Sync_th$="4" OR Sync_th$="5" OR Sync_th$="6" OR Sync_th$="8"
640          PRINT "Wrong chosen number!! Please select a correct SYNC THRE
SHOLD"
650          END LOOP
660          OUTPUT Add;"SYE "&Sync_th$
670          CLEAR SCREEN
680          END IF
690          INPUT "Do you change meas.condition?[Yes:0, No:1]",M_cond$
700          EXIT IF M_cond$="1"
710          END LOOP
720          CLEAR SCREEN
730          RETURN
740          !
750          !
760          !***** CHECK CLOCK LOSS *****
770 Closs:
780          !
790          LOOP
800              OUTPUT Add;"CLI?"
810              ENTER Add;Cli$
820              IF Cli$="CLI 1" THEN
830                  PRINT "*** CLOCK LOSS ***"
840                  WAIT .5
850              END IF
860              CLEAR SCREEN
870              EXIT IF Cli$="CLI 0"
880          END LOOP
890          !
900          RETURN
910          !
920          !***** AUTO SEARCH FUNCTION *****
930 Auto_srh:
940          !
950          OUTPUT Add;"SRH 1"
960          !
970          LOOP
980              OUTPUT Add;"SRH?"
990              ENTER Add;Srh$
1000             IF Srh$="SRH 1" THEN
1010                 PRINT "*** SEARCHING ***"
1020                 WAIT .5
1030             END IF
1040             CLEAR SCREEN
1050             EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1060         END LOOP
1070         !
1080         IF Srh$="SRH 2" THEN
1090             PRINT "Failed in auto search!! Program STOP!!"
1100             STOP
1110         END IF
1120         CLEAR SCREEN
1130         RETURN
1140         !
1150         !
1160         !***** INTERMEDIATE DATA *****
1170 Int_dat:
1180         OUTPUT Add;"MOD?"
1190         ENTER Add;Mod$
1200         IF Mod$="MOD 0" THEN
1210             OUTPUT Add;"ESR2?"
1220             ENTER Add;Esr2$
1230             OUTPUT Add;"STA"

```

```

1240 END IF
1250 !
1260 INPUT " CYCLE TIME ? ",Int_time
1270 LOOP
1280     IF Mod$((">"MOD 0" THEN
1290         OUTPUT Add;"STA"
1300     END IF
1310     !
1320     OUTPUT Add;"IMS"
1330     WAIT Int_time
1340     GOSUB Int_dat_read
1350 END LOOP
1360 !
1370 RETURN
1380 !
1390 !***** DISPLAY RESULT FUNCTION *****
1400 Int_dat_read: !
1410 !
1420 DIM Tme$(255),Arm$(3)(255),Err$(4)(255)
1430 DIM Arm_dat$(255),Err_dat$(255)
1440 !
1450 OUTPUT Add;"IMD? 0,0"
1460 ENTER Add;Tme$
1470 OUTPUT Add;"IMD? 1,0"
1480 ENTER Add;Arm_dat$
1490 OUTPUT Add;"IMD? 2,0"
1500 ENTER Add;Err_dat$
1510 PRINT Err_dat$
1520 Arm$(1)=" "
1530 Arm$(2)=" "
1540 Arm$(3)=" "
1550 K=1
1560 IF Arm_dat$(">"ERR" THEN
1570     Max_len=LEN(Arm_dat$)
1580     FOR J=1 TO 3
1590         LOOP
1600             EXIT IF K=(Max_len+1) OR Arm_dat$[K,K]=","
1610             Arm$(J)=Arm$(J)&Arm_dat$[K,K]
1620             K=K+1
1630         END LOOP
1640         K=K+1
1650     NEXT J
1660 ELSE
1670     FOR J=1 TO 3
1680         Arm$(J)=" NO DATA "
1690     NEXT J
1700 END IF
1710 !
1720 Err$(1)=" "
1730 Err$(2)=" "
1740 Err$(3)=" "
1750 Err$(4)=" "
1760 L=1
1770 IF Err_dat$(">"ERR" THEN
1780     Max_len2=LEN(Err_dat$)
1790     FOR M=1 TO 4
1800         LOOP
1810             EXIT IF Err_dat$[L,L]="," OR L=(Max_len2+1)
1820             Err$(M)=Err$(M)&Err_dat$[L,L]
1830             L=L+1
1840         END LOOP
1850         L=L+1
1860     NEXT M
1870 ELSE
1880     FOR M=1 TO 4
1890         Err$(M)=" NO DATA "

```

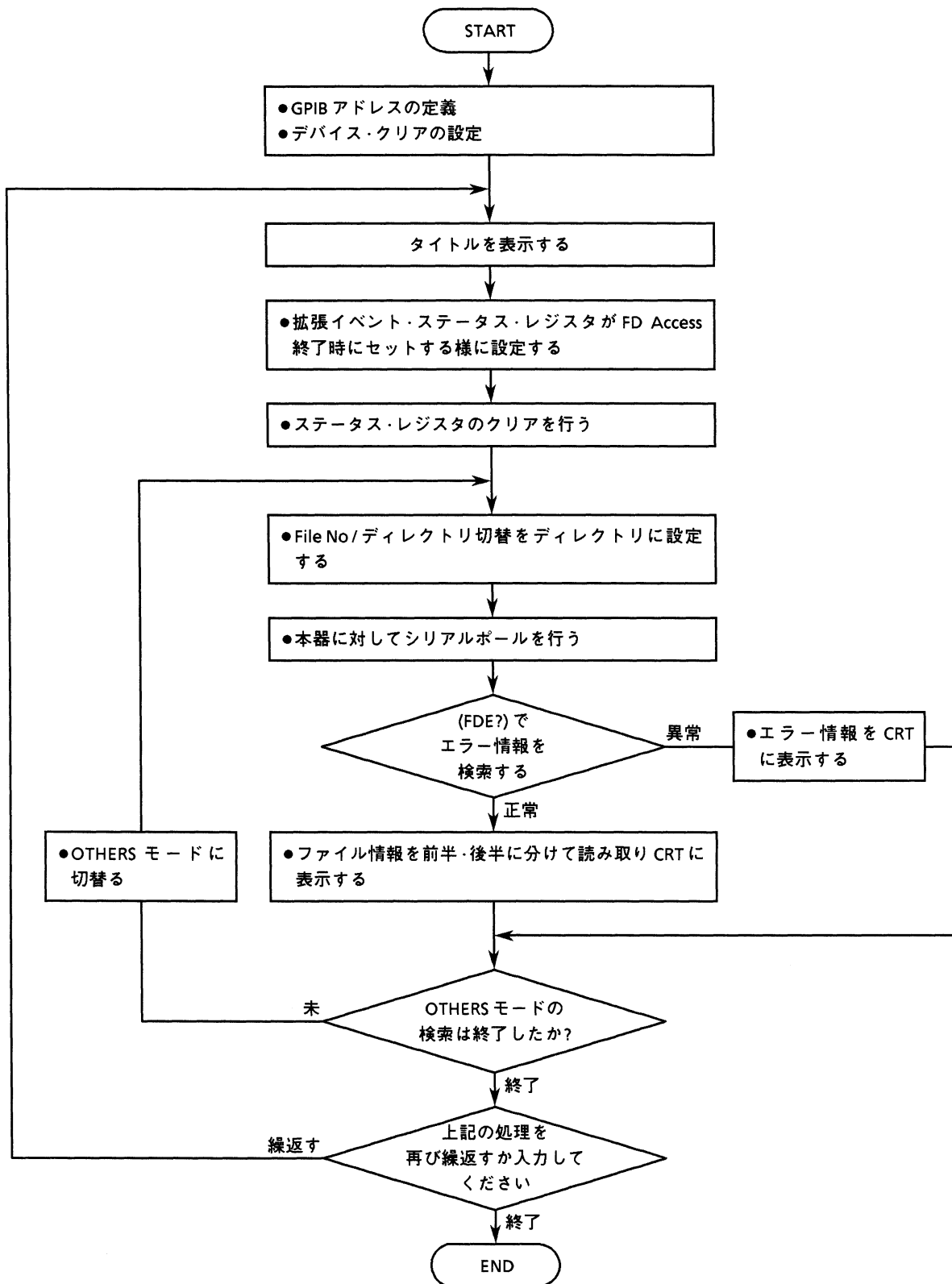
```

1900     NEXT M
1910 END IF
1920 !
1930 !
1940 IF Tme$="ERR" THEN
1950     GOTO Jump
1960 END IF
1970 !
1980 PRINT " << START TIME >>    "&Tme$[1,17]&" << INT TIME >>    "&Tme$[19,35]
1990 !
2000 PRINT " << ARLAM DATA >>  "
2010 PRINT " << POWER FAIL INTVL>> "&Arm$(1)
2020 PRINT " << CLOCK LOSS INTVL>> "&Arm$(2)
2030 PRINT " << SYNC LOSS INTVL >> "&Arm$(3)
2040 !
2050 PRINT " << ERROR DATA >>"
2060 PRINT " << ERROR RATIO >>    "&Err$(1)
2070 PRINT " << ERROR COUNT    >> "&Err$(2)
2080 PRINT " << EI              >> "&Err$(3)
2090 PRINT " << %EFI            >> "&Err$(4)
2100 !
2110 PRINT
2120 Jump: !
2130 RETURN
2140 !
2150 !
2160 END

```

(9) フロッピー・ディスクからのファイル情報読み出し

本プログラムは、フロッピー・ディスク内のファイル情報検索を行います。



● プログラム・リスト

```

10 !*****
20 !*
30 !*   MP1762D/MP1764D FLOPPY DISK OPERATION SAMPLE PROGRAM   *
40 !*                                     ED_FD1                  *
50 !*****
60 !
70 Add=701                                !MP1762D/MP1764D GPIB ADDRESS
80 CLEAR Add                             !DEVICE CLEAR
90 !
100 DIM Fil_pat$(2)[255],Fil_oths$(2)[255],Fde$(255)
110!
120 LOOP
130 !
140   CLEAR SCREEN
150   PRINT "*** MP1762D/MP1764D FD OPERATION SAMPLE PROGRAM ** "
160   !
170   GOSUB Status_set                    !STATUS RESISITOR SET
180   GOSUB Fd_ope                       !FD OPERATION SET
190   !
200   INPUT "   Try again?[Yes:0, No:1]",Loop#
210 EXIT IF Loop#="1"
220 END LOOP
230 !
240 STOP
250 !
260 !***** STATUS RESISTOR SET *****
270 Status_set: !
280 !
290 OUTPUT Add;"*SRE 4"                  !ESR2 ENABLE
300 OUTPUT Add;"ESE 2"                  !FD ACCESS END
310 !
320 RETURN
330 !
340 !***** FD OPERATION SET *****
350 Fd_ope: !
360 !
370 OUTPUT Add;"*STB?"
380 ENTER Add;Stb$
390 OUTPUT Add;"ESR2?"
400 ENTER Add;Esr2$
410 !
420 FOR I=0 TO 1
430   OUTPUT Add;"MEM "&VAL$(I)
440   !
450   OUTPUT Add;"FIL 1"
460   !
470   LOOP
480     A=SPOLL(Add)                    !SERIAL POLLING
490     EXIT IF BIT(A,2)=1              !ESR2 ENABLE
500     WAIT .1
510   END LOOP
520   GOSUB Fd_dir_dsp                  !FD DIR INFORMATION
530   OUTPUT Add;"ESR2?"
540   ENTER Add;Esr2$
550 NEXT I
560 !
570 RETURN
580 !
590 !***** FD DIR INFORMATION *****
600 Fd_dir_dsp: !
610 !

```

```

620 OUTPUT Add;"FDE?"
630 ENTER Add;Fde$
640 IF Fde$="FDE 10" THEN
650   OUTPUT Add;"FSH? 0"
660   IF I=0 THEN
670     ENTER Add;Fil_pat$(1)
680   ELSE
690     ENTER Add;Fil_oths$(1)
700   END IF
710   OUTPUT Add;"FSH? 1"
720   IF I=0 THEN
730     ENTER Add;Fil_pat$(2)
740   ELSE
750     ENTER Add;Fil_oths$(2)
760   END IF
770   !
780   IF I=0 THEN
790     PRINT " << Unused size      >> "&Fil_pat$(1)[5,11]
800     PRINT " << Used size      >> "&Fil_pat$(1)[13,19]
810     PRINT " << PATTERN FILES  >> "
820     PRINT " << PATT File count >>      "&Fil_pat$(1)[21,22]
830     K=24
840     Max_fil1=LEN(Fil_pat$(1))
850     Max_fil2=LEN(Fil_pat$(2))
860     IF Fil_pat$(1)[K,Max_fil1+1]<>"--" AND Fil_pat$(2)[K,Max_fil2+1]<>"-
--" THEN
870       PRINT " << File name      >>      "&Fil_pat$(1)[K,Max_fil1+1]&
,"&Fil_pat$(2)[K,Max_fil2+1]
880     ELSE
890       IF Fil_pat$(1)[K,Max_fil1+1]<>"--" THEN
900         PRINT " << File name      >>      "&Fil_pat$(1)[K,Max_fil1+
1]
910       ELSE
920         PRINT " << File name      >>      "&Fil_pat$(2)[K,Max_fil2+
1]
930       END IF
940     END IF
950   ELSE
960     PRINT " << OTHERS FILES      >>"
970     PRINT " << OTHS File count >>      "&Fil_oths$(1)[21,22]
980     K=24
990     Max_fil1=LEN(Fil_oths$(1))
1000    Max_fil2=LEN(Fil_oths$(2))
1010    IF Fil_oths$(1)[K,Max_fil1+1]<>"--" AND Fil_oths$(2)[K,Max_fil2+1]<>
"---" THEN
1020      PRINT " << File name      >>      "&Fil_oths$(1)[K,Max_fil1+1]&
,"&Fil_oths$(2)[K,Max_fil2+1]
1030    ELSE
1040      IF Fil_oths$(1)[K,Max_fil1+1]<>"--" THEN
1050        PRINT " << File name      >>      "&Fil_oths$(1)[K,Max_fil1
+1]
1060      ELSE
1070        PRINT " << File name      >>      "&Fil_oths$(2)[K,Max_fil2
+1]
1080      END IF
1090    END IF
1100  END IF
1110 ELSE
1120   SELECT Fde$[6,6]
1130   CASE "0"
1140     PRINT " << E0:Media error      >> "
1150   CASE "1"
1160     PRINT " << E1:Write protection error >> "
1170   CASE "2"
1180     PRINT " << E2:File full      >> "
1190   CASE "3"

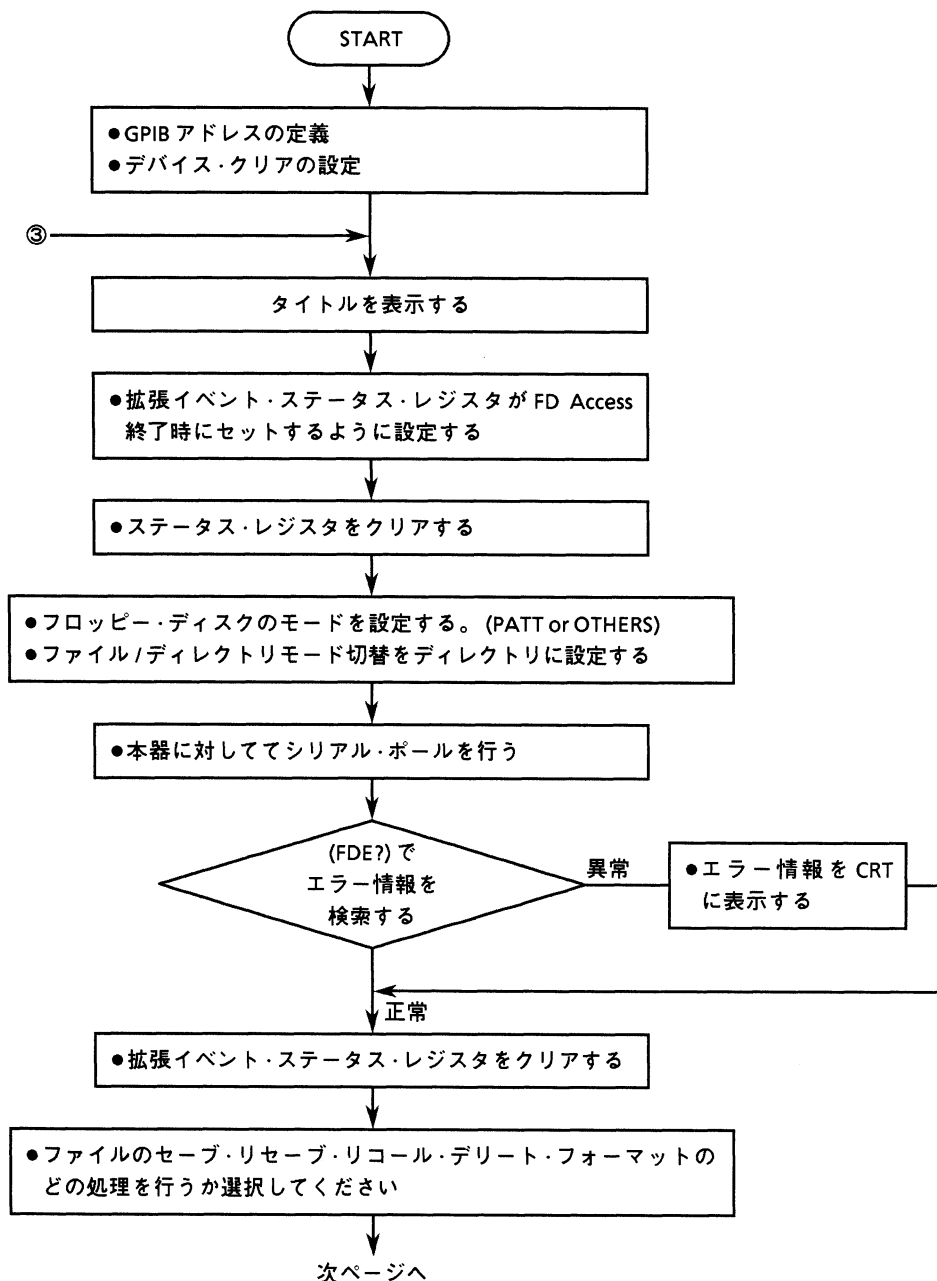
```

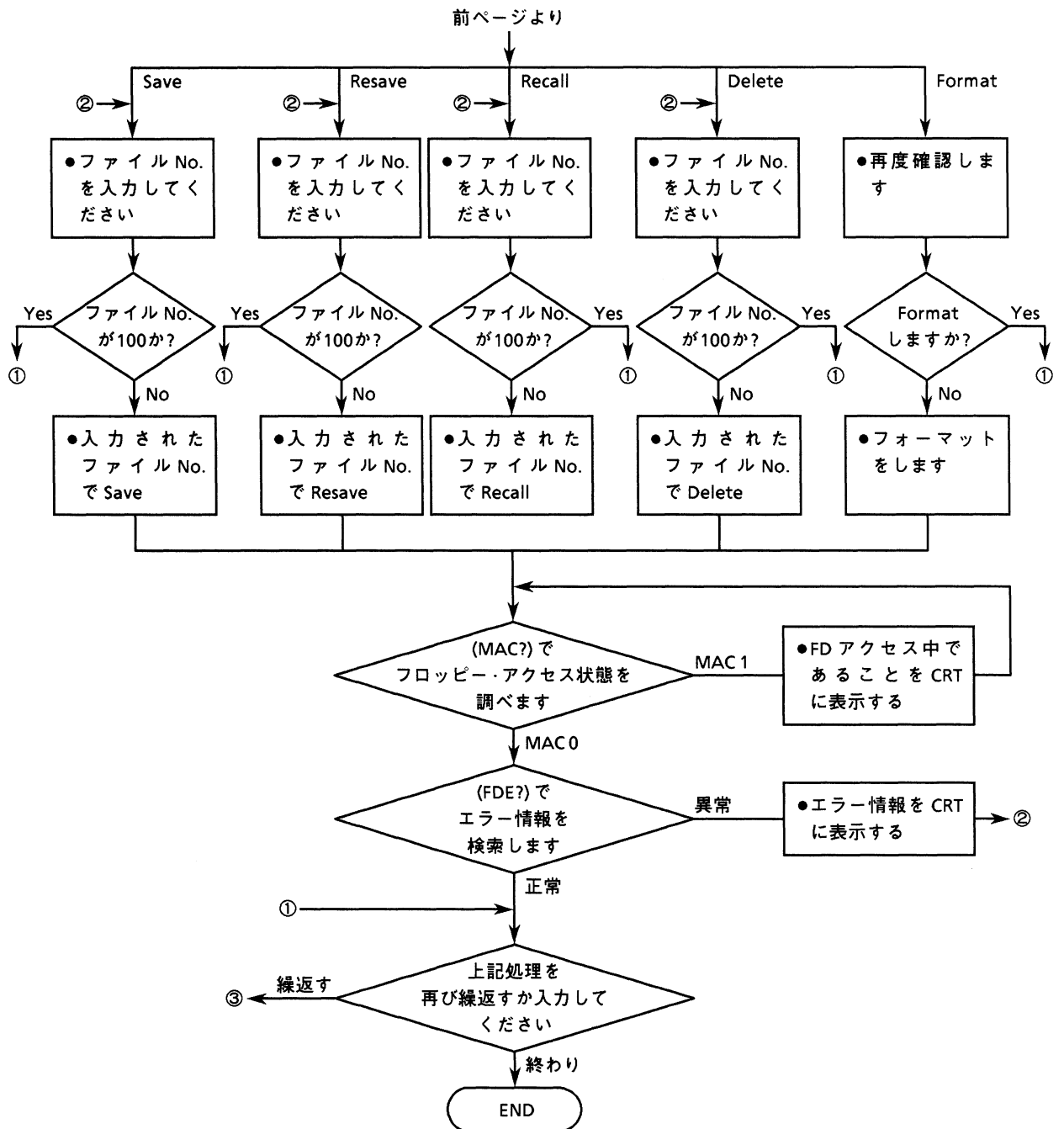
```
1200         PRINT " << E3:File not found          >> "
1210     CASE "4"
1220         PRINT " << E4:File already exists error >> "
1230     CASE "5"
1240         PRINT " << E5:Write error              >> "
1250     CASE "6"
1260         PRINT " << E6:Read error                >> "
1270     CASE "7"
1280         PRINT " << E7:File type , File error    >> "
1290     CASE "8"
1300         PRINT " << E8:FD error                  >> "
1310     CASE "9"
1320         PRINT " << E9:Hardware error            >> "
1330     END SELECT
1340 END IF
1350!
1360 RETURN
1370!
1380 END
```

(10) フロッピー・ディスクの操作

本プログラムは、フロッピー・ディスクの **Save/Recall/Resave/Delete/Format** の操作を行います。

フロッピー・ディスクのアクセス状態のチェックは、シリアル・ボールとリクエスト・コマンド (MAC?) の両方を使用しています。





● プログラム・リスト

```

10 !*****
20 !*
30 !*   MP1762D/MP1764D FLOPPY DISK OPERATION SAMPLE PROGRAM
40 !*                                     ED_FD2
50 !*****
60 !
70 Add=701                                !MP1762D/MP1764D GPIB ADDRESS
80 CLEAR Add                              !DEVICE CLEAR
90 !
100 DIM Fil_pat$(2)[255],Fde$(255)
110 !
120 LOOP
130 !
140   CLEAR SCREEN
150   PRINT "*** MP1762D/MP1764D FD OPERATION SAMPLE PROGRAM ** "
160   !
170   GOSUB Status_set                    !STATUS RESISITOR SET
180   GOSUB Fd_ope                        !FD OPERATION SET
190   GOSUB Fd_fil                        !FD FILE OPERATION
200   !
210   !
220   INPUT "   Try again?[Yes:0, No:1]",Loop$
230 EXIT IF Loop$="1"
240 END LOOP
250 !
260 STOP
270 !
280 !***** STATUS RESISTOR SET *****
290 Status_set: !
300 !
310 OUTPUT Add;"*SRE 4"                  !ESR2 ENABLE
320 OUTPUT Add;"*ESE2 2"                 !FD ACCESS END
330 !
340 RETURN
350 !
360 !***** FD OPERATION SET *****
370 Fd_ope: !
380 !
390 OUTPUT Add;"*STB?"
400 ENTER Add;Stb$
410 OUTPUT Add;"*ESR2?"
420 ENTER Add;Esr2$
430 !
440 LOOP
450   INPUT " Memory mode select [PATT:0, OTHERS:1]",Mmod$
460 EXIT IF Mmod$="0" OR Mmod$="1"
470   PRINT " Wrong chosen number!! Please select a correct number "
480 END LOOP
490 OUTPUT Add;"MEM "&Mmod$
500 !
510 OUTPUT Add;"FIL 1"
520 !
530 LOOP
540   A=SPOLL(Add)                        !SERIAL POLLING
550 EXIT IF BIT(A,2)=1                    !ESR2 ENABLE
560   WAIT .1
570 END LOOP
580 GOSUB Fd_err                          !FD DIR INFORMATION
590 OUTPUT Add;"*ESR2?"
600 ENTER Add;Esr2$
610 !

```

```

620 RETURN
630 !
640 !***** FD FILE OPERATION *****
650 Fd_fil: !
660 !
670 LOOP
680     INPUT " Select [ Save:0, Resave:1, Recall:2, Delete:3, Format:4 ],Ope
690 $
690 EXIT IF Ope$="0" OR Ope$="1" OR Ope$="2" OR Ope$="3" OR Ope$="4"
700 CLEAR SCREEN
710 PRINT "Wrong chosen number!! Please select a correct number"
720 END LOOP
730 !
740 SELECT Ope$
750     CASE "0"
760         GOSUB Dsave
770     CASE "1"
780         GOSUB Dresave
790     CASE "2"
800         GOSUB Drecall
810     CASE "3"
820         GOSUB Ddelete
830     CASE "4"
840         GOSUB Dformat
850 END SELECT
860 !
870 RETURN
880 !
890 !***** DATA SAVE OPERATION *****
900 Dsave: !
910 !
920 LOOP
930 !
940     Num$=""
950 !
960     INPUT "** DATA SAVE ** FILE NUMBER [0 to 99],[Exit:100] ",Num$
970 EXIT IF Num$="100"
980     OUTPUT Add;"SAV "&Num$
990     GOSUB Access
1000 EXIT IF Fde$="FDE 10"
1010 END LOOP
1020 CLEAR SCREEN
1030 !
1040 RETURN
1050 !
1060 !***** DATA RESAVE OPERATION *****
1070 Dresave: !
1080 !
1090 LOOP
1100 !
1110     Num$=""
1120 !
1130     INPUT "** DATA RESAVE ** FILE NUMBER [0 to 99],[Exit:100] ",Num$
1140 EXIT IF Num$="100"
1150     OUTPUT Add;"RSV "&Num$
1160     GOSUB Access
1170 EXIT IF Fde$="FDE 10"
1180 END LOOP
1190 CLEAR SCREEN
1200 !
1210 RETURN
1220 !
1230 !***** DATA RECALL OPERATION *****
1240 Drecall: !
1250 !
1260 LOOP

```

```

1270      !
1280      Num$=""
1290      !
1300      INPUT "** DATA RECALL ** FILE NUMBER [0 to 99],[Exit:100] ",Num$
1310      EXIT IF Num$="100"
1320      OUTPUT Add;"RCL "&Num$
1330      GOSUB Access
1340      EXIT IF Fde$="FDE 10"
1350      END LOOP
1360      CLEAR SCREEN
1370      !
1380      RETURN
1390      !
1400      !***** DATA DELETE OPERATION *****
1410      Ddelete: !
1420      !
1430      LOOP
1440      !
1450      Num$=""
1460      !
1470      INPUT "** FILE DELETE ** FILE NUMBER [0 to 99],[Exit:100] ",Num$
1480      EXIT IF Num$="100"
1490      OUTPUT Add;"DEL "&Num$
1500      GOSUB Access
1510      EXIT IF Fde$="FDE 10"
1520      END LOOP
1530      CLEAR SCREEN
1540      !
1550      RETURN
1560      !
1570      !***** FD FORMAT OPERATION *****
1580      Dformat: !
1590      INPUT " Format disk [Yes:0, No:1]",Fmt$
1600      IF Fmt$="0" THEN
1610          OUTPUT Add;"FIL 0"
1620          OUTPUT Add;"FDF"
1630          GOSUB Access
1640      END IF
1650      CLEAR SCREEN
1660      !
1670      RETURN
1680      !
1690      !***** FD ERROR CHECK *****
1700      Fd_err: !
1710      !
1720      OUTPUT Add;"FDE?"
1730      ENTER Add;Fde$
1740      IF Fde$<>"FDE 10" THEN
1750          SELECT Fde$[6,6]
1760              CASE "0"
1770                  PRINT " << E0:Media error          >> "
1780              CASE "1"
1790                  PRINT " << E1:Write protection error    >> "
1800              CASE "2"
1810                  PRINT " << E2:File full              >> "
1820              CASE "3"
1830                  PRINT " << E3:File not found         >> "
1840              CASE "4"
1850                  PRINT " << E4:File already exists error >> "
1860              CASE "5"
1870                  PRINT " << E5:Write error            >> "
1880              CASE "6"
1890                  PRINT " << E6:Read error             >> "
1900              CASE "7"
1910                  PRINT " << E7:File type , File error  >> "
1920              CASE "8"

```



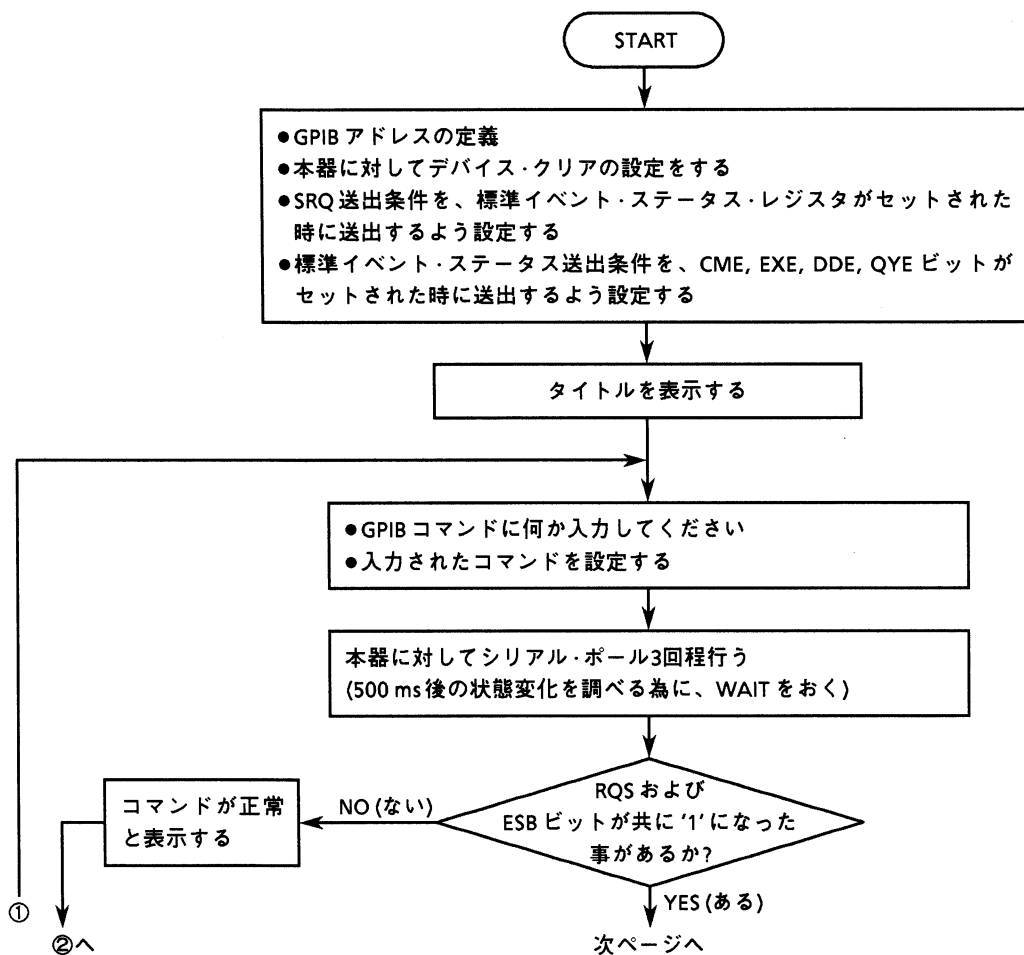
```
1930          PRINT " << E8:FD error                >> "
1940          CASE "9"
1950              PRINT " << E9:Hardware error        >> "
1960          END SELECT
1970      ELSE
1980          PRINT " << Operation complete !! >> "
1990      END IF
2000      !
2010      RETURN
2020      !
2030      !***** FD ACCESS CHECK *****
2040      Access: !
2050      !
2060      LOOP
2070          OUTPUT Add;"MAC?"
2080          ENTER Add;Mac$
2090          EXIT IF Mac$="MAC 0"
2100          PRINT " FD ACCESS "
2110          WAIT .5
2120          CLEAR SCREEN
2130      END LOOP
2140      !
2150      GOSUB Fd_err
2160      !
2170      RETURN
2180      !
2190      END
```

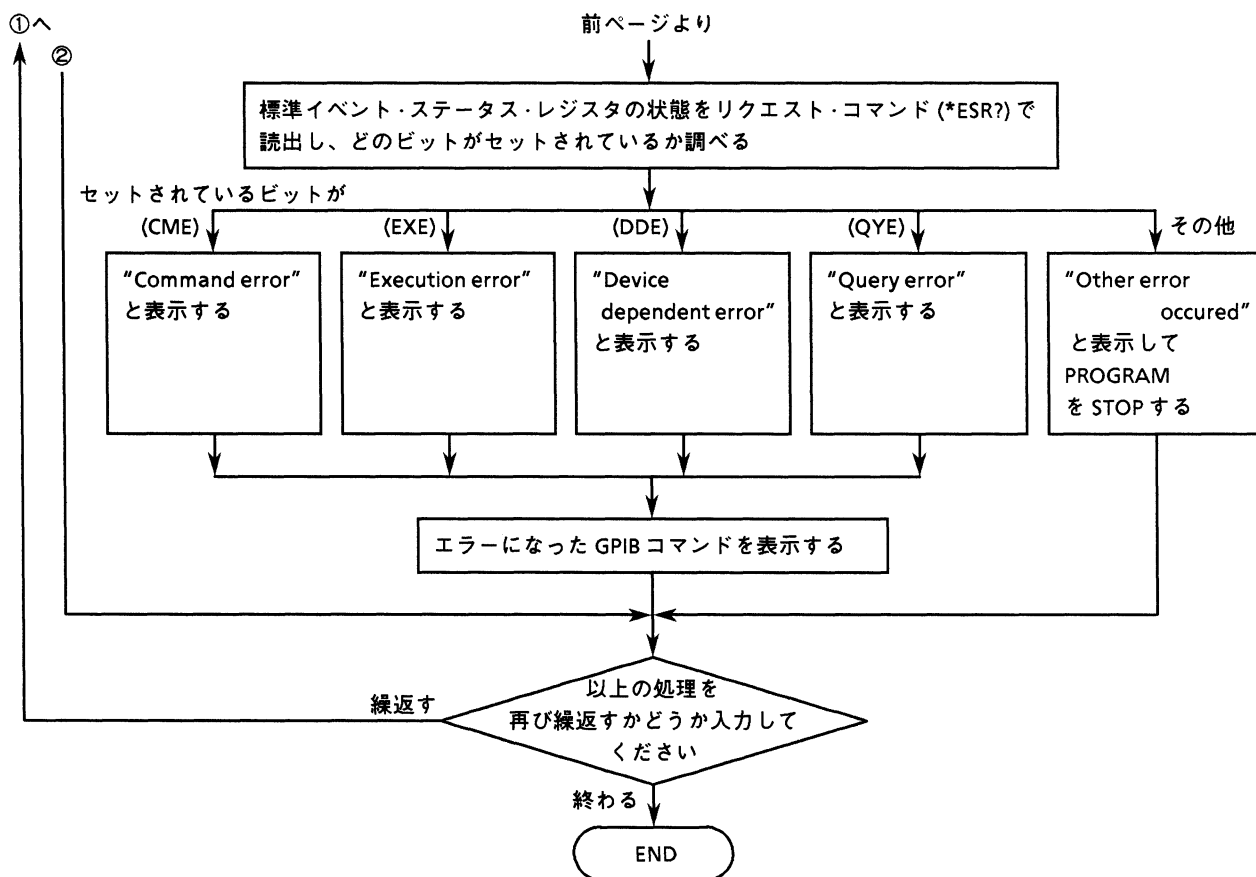
(11) ステータス・バイトのチェック

本プログラムは、標準イベント・ステータス・バイトの **CME, EXE, DDE, QYE** ビットを調べ、入力された **GPIB** コマンドが正常かどうかを **CRT** に表示するものです。

なお、異常な時は、その時のエラーの意味を表示します。

また、**GPIB** ステータス・バイトのチェックは、シリアル・ポーリングと標準イベント・ステータス・レジスタの状態を、データ・リクエスト・コマンドによって調べます。





● プログラム・リスト

```

10 !*****
20 !*
30 !*      MP1762D/MP1764D STANDARD EVENT STATUS REGISTOR CHECK      *
40 !*      SAMPLE PROGRAM      ED_ESR      *
50 !*****
60 !
70 Add=701                                !MP1762D/MP1764D GPIB ADDRESS
80 CLEAR Add                              !DEVICE CLEAR
90 !
100 OUTPUT Add;"*SRE 32"                  !SRQ ON ESR bit
110 OUTPUT Add;"*ESE 60"                  !ESR ON CME,EXE,DDE,GYE
120 OUTPUT Add;"ESE2 0"                  !DISABLE ESR2
130 OUTPUT Add;"ESE3 0"                  !DISABLE ESR3
140 !
150 PRINT " **      MP1762D/MP1764D      ** "
160 PRINT " ** STANDARD EVENT STATUS REGISTOR CHECK ** "
170 !
180 LOOP
190 !
200 INPUT " Input any GPIB command ? ",Cmd$
210 OUTPUT Add;Cmd$
220 !
230 GOSUB S_poll
240 !
250 INPUT " Next command set ? [Yes:0, No:1] ",Loop$
260 !
270 EXIT IF Loop$="1"
280 !
290 END LOOP
300 !
310 STOP
320 !
330 !***** POLLING *****
340 S_poll:
350 !
360 Byt=0
370 !
380 FOR I=0 TO 2
390 !
400 A=SPOLL(Add)
410 IF BIT(A,6)=1 AND BIT(A,5)=1 THEN
420 Byt=A
430 END IF
440 WAIT .5
450 NEXT I
460 !
470 IF BIT(Byt,6)=1 AND BIT(Byt,5)=1 THEN
480 GOSUB Err
490 ELSE
500 PRINT " GPIB command is OK!! "
510 PRINT
520 END IF
530 !
540 RETURN
550 !
560 !***** ESR CHECK *****
570 Err:
580 !
590 OUTPUT Add;"*ESR?"
600 ENTER Add;Esr
610 IF BIT(Esr,2)=1 THEN PRINT " << Query error >> "

```

```
620 IF BIT(Esr,3)=1 THEN PRINT " << Device dependent error >> "
630 IF BIT(Esr,4)=1 THEN PRINT " << Execution error >> "
640 IF BIT(Esr,5)=1 THEN PRINT " << Command error >> "
650 !
660 PRINT " Input command = "&Cmd$
670 PRINT
680 !
690 IF BIT(Esr,0)=1 OR BIT(Esr,1)=1 OR BIT(Esr,6)=1 OR BIT(Esr,7)=1 THEN
700     PRINT " Other error occurred !! "
710 END IF
720 !
730 RETURN
740 !
750 END
```

(12) パターン・データの DMA 転送

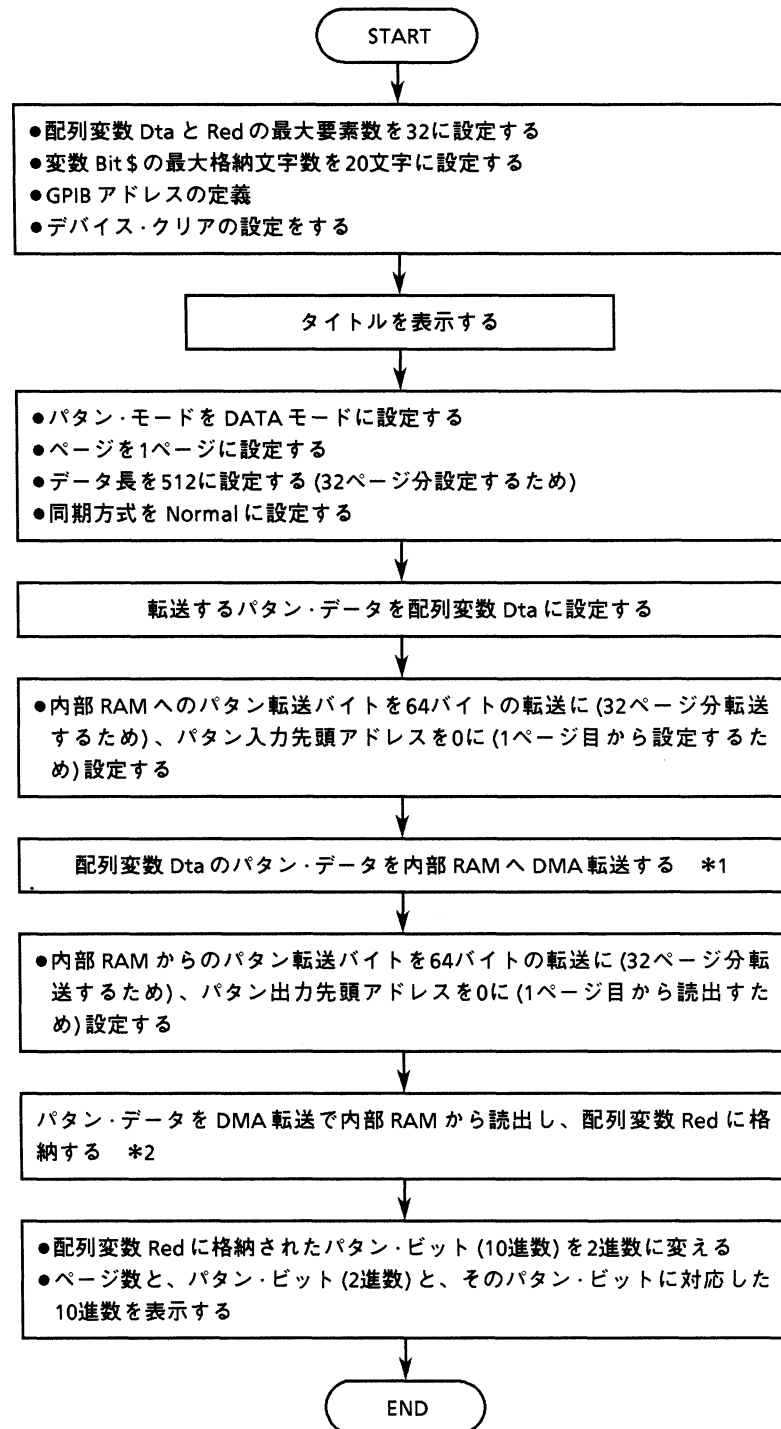
コントローラとして HP 9000 シリーズを対象とした、パターン・データの DMA 転送を行うプログラムです。

本器に DMA 転送されたパターン・データの出力を DMA 転送によって行っています。

なお、実際の実行結果を以下に示してありますが、各配列変数 **Dta** とそれに設定する数値との関係 (10進数表現, 2進数表現, 16進数表現) を表12-1に示しますが、この関係は、使用するコントローラによって多少異なります。

表12-1 配列変数と設定値の関係

配列変数	設定値 10進数	2進数及び BIT LED No.																16進数	対応 ページ数
		16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1		
Dta (0)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1H	1
Dta (1)	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2H	2
Dta (2)	4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4H	3
Dta (3)	8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	8H	4
Dta (4)	16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	10H	5
Dta (5)	32	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	20H	6
Dta (6)	64	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	40H	7
Dta (7)	128	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	80H	8
Dta (8)	256	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	100H	9
Dta (9)	512	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	200H	10
Dta (10)	1024	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	400H	11
Dta (11)	2048	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	800H	12
Dta (12)	4096	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000H	13
Dta (13)	8192	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000H	14
Dta (14)	16384	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000H	15
Dta (15)	32767	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	7FFFH	16
Dta (16)	-32768	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000H	17
Dta (17)	-16384	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C000H	18
Dta (18)	-8192	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	E000H	19
Dta (19)	-4096	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	F000H	20
Dta (20)	-2048	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	F800H	21
Dta (21)	-1024	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	FC00H	22
Dta (22)	-512	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	FE00H	23
Dta (23)	-256	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	FF00H	24
Dta (24)	-128	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	FF80H	25
Dta (25)	-64	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	FFC0H	26
Dta (26)	-32	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	FFE0H	27
Dta (27)	-16	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	FFF0H	28
Dta (28)	-8	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	FFF8H	29
Dta (29)	-4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	FFFCH	30
Dta (30)	-2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	FFFEH	31
Dta (31)	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFH	32



1 パタン・データの転送 <OUTPUT Add USING "W" ; Dta() >

W : 16ビットの2の補数の整数を出力します。

GPIB インターフェイス・ボードは8ビットのI/Oなので、上位バイトが先で2バイト送られます。

* : 定義した配列 Dta 全体を出力します。

2 パタン・データの転送 <ENTER Add USING "W" ; Red() >

W : 16ビットの2の補数の整数を出力します。

GPIB インターフェイス・ボードは8ビットのI/Oなので、上位バイトが先で2バイト送られます。

* : 定義した配列 Red 全体に格納されます。

● プログラム・リスト

```

10 !*****
20 !*
30 !*   MP1762D/MP1764D PROGRAMMABLE PATTERN DATA DMA TRANSFER   *
40 !*                               SAMPLE PROGRAM                     ED_DMA   *
50 !*****
60 !
70 DIM Red(31)
80 DIM Dta(31)
90 DIM Bit$(20)
100!
110 Add=701                                !MP1762D/MP1764D GPIB ADDRESS
120 CLEAR Add                              !DEVICE CLEAR
130 CLEAR SCREEN
140 !
150 PRINT " **           MP1762D/MP1764D           ** "
160 PRINT " **           PATTERN DATA DMA TRANSFER           ** "
170 !
180 OUTPUT Add;"PTS 1"
190 OUTPUT Add;"PAG 1"
200 !
210 OUTPUT Add;"SYM 0"
220 !
230 OUTPUT Add;"DLN 512"
240 !
250 !***** DATA SET *****
260 Dta(0)=1
270 Dta(1)=2
280 Dta(2)=4
290 Dta(3)=8
300 Dta(4)=16
310 Dta(5)=32
320 Dta(6)=64
330 Dta(7)=128
340 Dta(8)=256
350 Dta(9)=512
360 Dta(10)=1024
370 Dta(11)=2048
380 Dta(12)=4096
390 Dta(13)=8192
400 Dta(14)=16384
410 Dta(15)=32767
420 Dta(16)=-32768
430 Dta(17)=-16384
440 Dta(18)=-8192
450 Dta(19)=-4096
460 Dta(20)=-2048
470 Dta(21)=-1024
480 Dta(22)=-512
490 Dta(23)=-256
500 Dta(24)=-128
510 Dta(25)=-64
520 Dta(26)=-32
530 Dta(27)=-16
540 Dta(28)=-8
550 Dta(29)=-4
560 Dta(30)=-2
570 Dta(31)=-1
580 !
590 !
600 OUTPUT Add;"WRT 64,0"
610 !

```



```
620 OUTPUT Add USING "W";Dta(*)
630 !
640 OUTPUT Add;"RED? 64,0"
650 !
660 ENTER Add USING "W";Red(*)
670 !
680 FOR I=1 TO 32
690     Bit#=IVAL$(Red(I-1),2)
700     !
710     IMAGE "PATTERN BIT PAGE=",AA,XXX,AAAAAAAAAAAAAAAA,XX,DDDDDD
720     PRINT USING 710;VAL$(I);Bit#;Red(I-1)
730     !
740 NEXT I
750 !
760 END
```

実行結果

PATTERN BIT PAGE=1	0000000000000001	1
PATTERN BIT PAGE=2	0000000000000010	2
PATTERN BIT PAGE=3	0000000000000100	4
PATTERN BIT PAGE=4	0000000000001000	8
PATTERN BIT PAGE=5	0000000000010000	16
PATTERN BIT PAGE=6	0000000000100000	32
PATTERN BIT PAGE=7	0000000001000000	64
PATTERN BIT PAGE=8	0000000010000000	128
PATTERN BIT PAGE=9	0000000100000000	256
PATTERN BIT PAGE=10	0000001000000000	512
PATTERN BIT PAGE=11	0000010000000000	1024
PATTERN BIT PAGE=12	0000100000000000	2048
PATTERN BIT PAGE=13	0001000000000000	4096
PATTERN BIT PAGE=14	0010000000000000	8192
PATTERN BIT PAGE=15	0100000000000000	16384
PATTERN BIT PAGE=16	0111111111111111	32767
PATTERN BIT PAGE=17	1000000000000000	-32768
PATTERN BIT PAGE=18	1100000000000000	-16384
PATTERN BIT PAGE=19	1110000000000000	-8192
PATTERN BIT PAGE=20	1111000000000000	-4096
PATTERN BIT PAGE=21	1111100000000000	-2048
PATTERN BIT PAGE=22	1111110000000000	-1024
PATTERN BIT PAGE=23	1111111000000000	-512
PATTERN BIT PAGE=24	1111111100000000	-256
PATTERN BIT PAGE=25	1111111110000000	-128
PATTERN BIT PAGE=26	1111111111000000	-64
PATTERN BIT PAGE=27	1111111111100000	-32
PATTERN BIT PAGE=28	1111111111110000	-16
PATTERN BIT PAGE=29	1111111111111000	-8
PATTERN BIT PAGE=30	1111111111111100	-4
PATTERN BIT PAGE=31	1111111111111110	-2
PATTERN BIT PAGE=32	1111111111111111	-1

(13) BLOCK WINDOW パターン・データの DMA 転送

コントローラとして HP9000 シリーズを対象とした、BLOCK WINDOW パターンの DMA 転送を行うプログラムです。

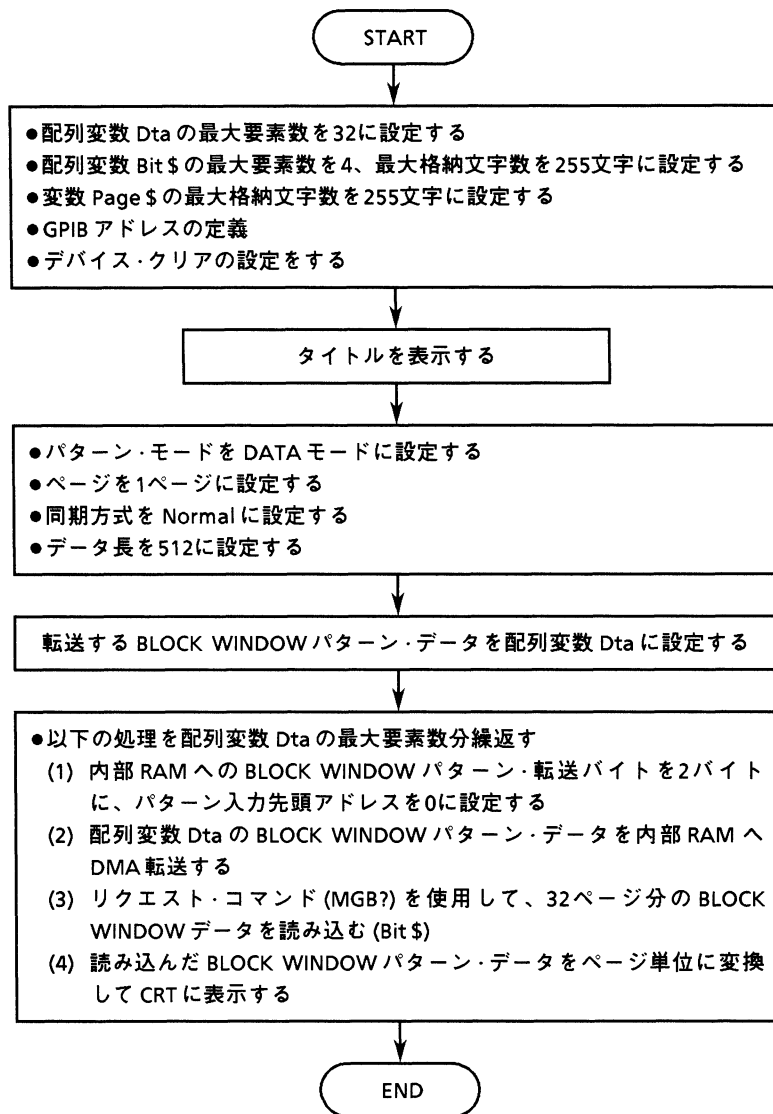
本器に DMA 転送された BLOCK WINDOW パターン・データの出力をリクエスト・コマンド (MGB?) によって行います。

なお、実行結果を以下に示してありますが、各配列 Dta とそれに設定する数値殿関係 (10進数、16進数) を表13-1に示しますが、この関係は使用するコントローラによって多少異なります。

また、対応するページ番号は DMA 転送先頭アドレスを0にした場合の値です。(付録Bを参照してください。)

表13-1 配列変数と設定値との関係

配列変数	設定値 10進数	先頭を1ページとした時の対応するページ番号							設定値 16進数
		1	8	9	16	17	24	25	
Dta (0)	1	00000000	00000000		11000000		00000000		1H
Dta (1)	2	00000000	00000000		00110000		00000000		2H
Dta (2)	4	00000000	00000000		00001100		00000000		4H
Dta (3)	8	00000000	00000000		00000011		00000000		8H
Dta (4)	16	00000000	00000000		00000000		11000000		10H
Dta (5)	32	00000000	00000000		00000000		00110000		20H
Dta (6)	64	00000000	00000000		00000000		00001100		40H
Dta (7)	128	00000000	00000000		00000000		00000011		80H
Dta (8)	256	11000000	00000000		00000000		00000000		100H
Dta (9)	512	00110000	00000000		00000000		00000000		200H
Dta (10)	1024	00001100	00000000		00000000		00000000		400H
Dta (11)	2048	00000011	00000000		00000000		00000000		800H
Dta (12)	4096	00000000	11000000		00000000		00000000		1000H
Dta (13)	8192	00000000	00110000		00000000		00000000		2000H
Dta (14)	16384	00000000	00001100		00000000		00000000		4000H
Dta (15)	32767	11111111	11111100		11111111		11111111		7FFFH
Dta (16)	-32768	00000000	00000011		00000000		00000000		8000H
Dta (17)	-16384	00000000	00001111		00000000		00000000		C000H
Dta (18)	-8192	00000000	00111111		00000000		00000000		E000H
Dta (19)	-4096	00000000	11111111		00000000		00000000		F000H
Dta (20)	-2048	00000011	11111111		00000000		00000000		F800H
Dta (21)	-1024	00001111	11111111		00000000		00000000		FC00H
Dta (22)	-512	00111111	11111111		00000000		00000000		FE00H
Dta (23)	-256	11111111	11111111		00000000		00000000		FF00H
Dta (24)	-128	11111111	11111111		00000000		00000011		FF80H
Dta (25)	-64	11111111	11111111		00000000		00001111		FFC0H
Dta (26)	-32	11111111	11111111		00000000		00111111		FFE0H
Dta (27)	-16	11111111	11111111		00000000		11111111		FFF0H
Dta (28)	-8	11111111	11111111		00000011		11111111		FFF8H
Dta (29)	-4	11111111	11111111		00001111		11111111		FFFCH
Dta (30)	-2	11111111	11111111		00111111		11111111		FFFEH
Dta (31)	-1	11111111	11111111		11111111		11111111		FFFFH



● プログラム・リスト

```

10 !*****
20 !*
30 !*      MP1762D/MP1764D BLOCK WINDOW PATTERN DATA DMA TRANSFER      *
40 !*                      SAMPLE PROGRAM                      ED_DMA2      *
50 !*****
60 !
70 !
80 DIM Dta(31)
90 DIM Bit$(3)[255]
100 DIM Page$(32)
110 !
120 Add=701                                !MP1762D/MP1764D GPIB ADDRESS
130 CLEAR Add                              !DEVICE CLEAR
140 CLEAR SCREEN
150 !
160 PRINT " **      MP1762D/MP1764D      ** "
170 PRINT " **      BLOCK WINDOW DATA DMA TRANSFER      ** "
180 !
190 OUTPUT Add;"PTS 1"
200 OUTPUT Add;"PAG 1"
210 !
220 OUTPUT Add;"SYM 0"
230 !
240 OUTPUT Add;"DLN 512"
250 !
260 !***** DATA SET *****
270 Dta(0)=1
280 Dta(1)=2
290 Dta(2)=4
300 Dta(3)=8
310 Dta(4)=16
320 Dta(5)=32
330 Dta(6)=64
340 Dta(7)=128
350 Dta(8)=256
360 Dta(9)=512
370 Dta(10)=1024
380 Dta(11)=2048
390 Dta(12)=4096
400 Dta(13)=8192
410 Dta(14)=16384
420 Dta(15)=32768
430 Dta(16)=-32768
440 Dta(17)=-16384
450 Dta(18)=-8192
460 Dta(19)=-4096
470 Dta(20)=-2048
480 Dta(21)=-1024
490 Dta(22)=-512
500 Dta(23)=-256
510 Dta(24)=-128
520 Dta(25)=-64
530 Dta(26)=-32
540 Dta(27)=-16
550 Dta(28)=-8
560 Dta(29)=-4
570 Dta(30)=-2
580 Dta(31)=-1
590 !
600 !

```

```

610 FOR I=1 TO 32
620     OUTPUT Add;"MWT 2,0"
630     !
640     OUTPUT Add USING "W";Dta(I-1)
650     !
660     GOSUB Page_ana
670     IMAGE "BLOCK WINDOW PAGE = ",XXX,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,XX,DDD
680     PRINT USING 670;Page$;Dta(I-1)
690     !
700 NEXT I
710 !
720 STOP
730 !
740 !***** BLOCK WINDOW PAGE ANALYSIS *****
750 Page_ana: !
760 !
770 FOR J=1 TO 4
780     !
790     OUTPUT Add;"FAG "&VAL$((J-1)*8+1)
800     OUTPUT Add;"MBB?"
810     ENTER Add;Bit$(J-1)
820     FOR K=1 TO 8
830         !
840         IF Bit$(J-1)[21+7*(K-1),24+7*(K-1)]="FFFF" THEN
850             Page$[K+(J-1)*8,K+(J-1)*8]="1"
860         ELSE
870             Page$[K+(J-1)*8,K+(J-1)*8]="0"
880         END IF
890     NEXT K
900 NEXT J
910 !
920 RETURN
930 !
940 END

```

実行結果

```

BLOCK WINDOW PAGE = 00000000000000000110000000000000 1
BLOCK WINDOW PAGE = 00000000000000000011000000000000 2
BLOCK WINDOW PAGE = 00000000000000000000110000000000 4
BLOCK WINDOW PAGE = 00000000000000000000001100000000 8
BLOCK WINDOW PAGE = 00000000000000000000000011000000 16
BLOCK WINDOW PAGE = 00000000000000000000000000110000 32
BLOCK WINDOW PAGE = 00000000000000000000000000001100 64
BLOCK WINDOW PAGE = 00000000000000000000000000000011 128
BLOCK WINDOW PAGE = 1100000000000000000000000000000000 256
BLOCK WINDOW PAGE = 001100000000000000000000000000000000 512
BLOCK WINDOW PAGE = 000011000000000000000000000000000000 1024
BLOCK WINDOW PAGE = 000000110000000000000000000000000000 2048
BLOCK WINDOW PAGE = 000000001100000000000000000000000000 4096
BLOCK WINDOW PAGE = 000000000011000000000000000000000000 8192
BLOCK WINDOW PAGE = 000000000000110000000000000000000000 16384
BLOCK WINDOW PAGE = 11111111111111001111111111111111 32767
BLOCK WINDOW PAGE = 000000000000000110000000000000000000 -32768
BLOCK WINDOW PAGE = 000000000000011110000000000000000000 -16384
BLOCK WINDOW PAGE = 000000000001111110000000000000000000 -8192
BLOCK WINDOW PAGE = 000000000111111110000000000000000000 -4096
BLOCK WINDOW PAGE = 000000011111111110000000000000000000 -2048
BLOCK WINDOW PAGE = 000011111111111110000000000000000000 -1024
BLOCK WINDOW PAGE = 001111111111111110000000000000000000 -512
BLOCK WINDOW PAGE = 111111111111111000000000000000000000 -256
BLOCK WINDOW PAGE = 111111111111111110000000000000000011 -128
BLOCK WINDOW PAGE = 1111111111111110000000000000001111 -64
BLOCK WINDOW PAGE = 11111111111111100000000000111111 -32
BLOCK WINDOW PAGE = 11111111111111100000000011111111 -16
BLOCK WINDOW PAGE = 11111111111111100000001111111111 -8
BLOCK WINDOW PAGE = 11111111111111100000111111111111 -4
BLOCK WINDOW PAGE = 11111111111111100111111111111111 -2
BLOCK WINDOW PAGE = 11111111111111111111111111111111 -1

```

DECpc を用いたサンプル・プログラム例

<プログラム共通部分の説明>

Quick Basic Ver4.50およびナショナル・インストゥル・メント社製 GPIB ボードを使用して、以降のサンプルプログラムは作成されています。(Quick Basic および NI 社製 GPIB ドライバに関する詳細は、それぞれの取扱説明書を参照してください。)

サンプル・プログラムで必要な共通関数を下記の2種類のプログラムにまとめてあります。

- COMMON.BAS
- ACS_GPIB.BAS

サンプル・プログラムを実行する際には、上記のプログラムが必要となるので前もって準備しておいてください。

また、使用されるアプリケーションに必要な関数のみを作成されてもけっこうです。

2種類の共通関数について、まず説明します。

<COMMON.BASの説明>

COMMON.BASは下記の7種類の関数で構成されています。

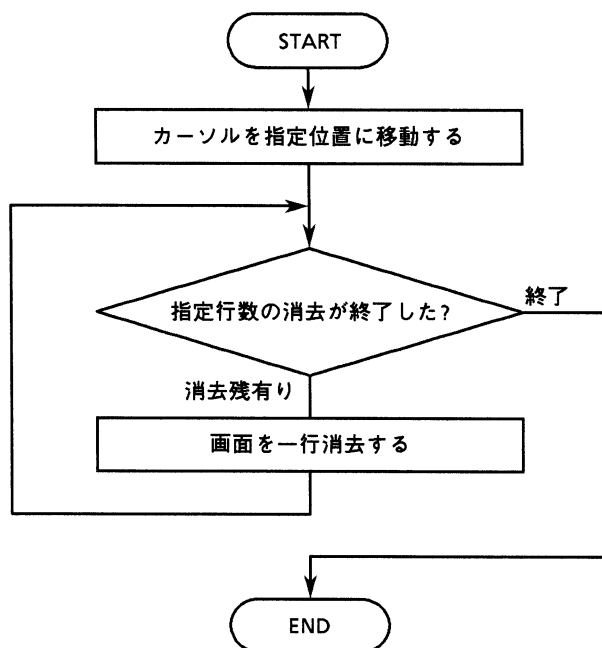
表10-2 COMMON.BASの関数表

モジュール 番号	関 数 名	処 理 概 要
1.1	SUB ClearDisp (p%, l%)	画面を行単位で消去する p% : 消去開始行番号 l% : 消去する行数
1.2	SUB Connect (ttl\$)	接続図を表示する ttl\$: 接続図と同時に表示するタイトル文字列
1.3	FUNC Exchange% (i%)	単精度整数をビット・パターンとするデータの上位、下位をバイト単位で交換する i% : ビット・パターン・データ
1.4	FUNC itob\$ (l%, v%)	単精度整数を LSB から指定されたビット長の2進文字列に変換する ただし、出力文字長は16文字固定とする。 l% : 2進文字列長 v% : 変換データ
1.5	SUB SelItem (dist\$())	測定項目選択のキー入力処理を行う dist\$() : 発行するコマンドの引き数文字配列 (出力)
1.6	SUB Disp1 ()	SelItem () で設定されたコマンドによって読み出されたデータを表示する
1.7	SUB waidly (tim)	指定時間 (秒) wait する tim : 指定時間 (秒) (入力)

次ページ以降に、各関数のフローチャートとプログラム・リストを掲載します。

(1.1) SUB ClearDisp (p%, l%): 画面を消去する

● フローチャート



● プログラム・リスト

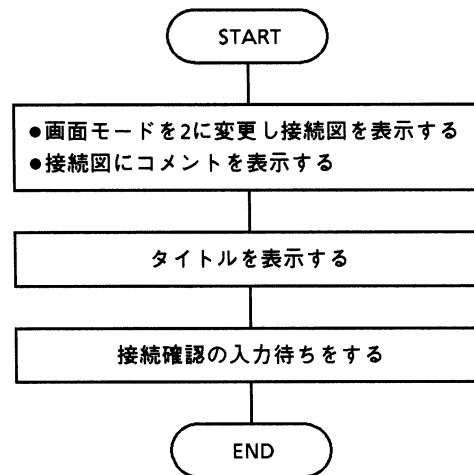
```

' ---- Procedure for Clear display ----
' in   p%:Location line number
'      l%:clear line count
'
SUB ClearDisp (p%, l%)
  LOCATE p%, 1
  FOR i% = 0 TO (l% - 1)
    PRINT "
  NEXT i%
END SUB

```

(1.2) SUB Connect (ttl\$): タイトルと接続図を表示する

● フローチャート



● プログラム・リスト

```

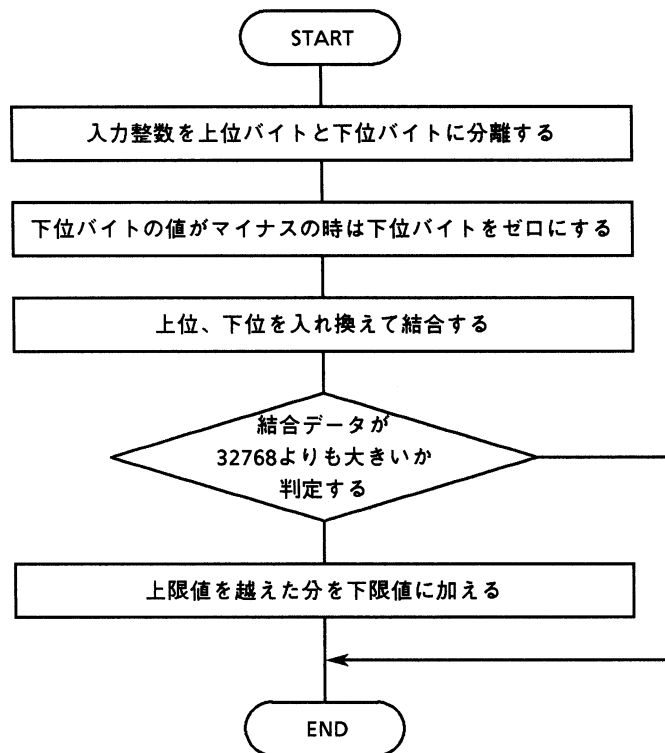
' ---- Connection layout ----
' in   ttl$:Title message for method
'
' This is drawing connection line layout for MP1762D with other PPG.
'
SUB Connect (ttl$)
  CLS
  SCREEN 2
  WINDOW (-600, -500)-(600, 500)

  LINE (-50, -50)-(600, 500), 14, B
  LINE (10, 100)-STEP(235, 200), , B
  LINE (300, 100)-STEP(235, 200), , B
  '
  CIRCLE (130, 130), 8
  LINE (130, 125)-STEP(0, -55)
  LINE (130, 70)-STEP(220, 0)
  LINE (350, 70)-STEP(0, 55)
  CIRCLE (350, 130), 8
  '
  CIRCLE (190, 130), 8
  LINE (190, 125)-STEP(0, -100)
  LINE (190, 25)-STEP(220, 0)
  LINE (410, 25)-STEP(0, 100)
  CIRCLE (410, 130), 8
  '
  LOCATE 2, 45: PRINT "<< CONNECTION Layout >>"
  LOCATE 4, 42: PRINT "MP1761B/C/MP1763B/C      MP1762D/MP1764D"
  LOCATE 9, 45: PRINT "DATA CLOCK1"
  LOCATE 9, 62: PRINT "DATA CLOCK"
  '
  LOCATE 1, 1: PRINT ttl$
  LOCATE 22, 1: PRINT "You must confirm connection line."
  LOCATE 23, 1: INPUT "Aer You ready to start? Press 'Enter' to continue.", a
  '
  LOCATE 22, 1: PRINT "
  LOCATE 23, 1: PRINT "
END SUB

```

(1.3) FUNCTION Exchange (i%): 16-BIT 整数データをバイト単位で交換する

● フローチャート



● プログラム・リスト

```

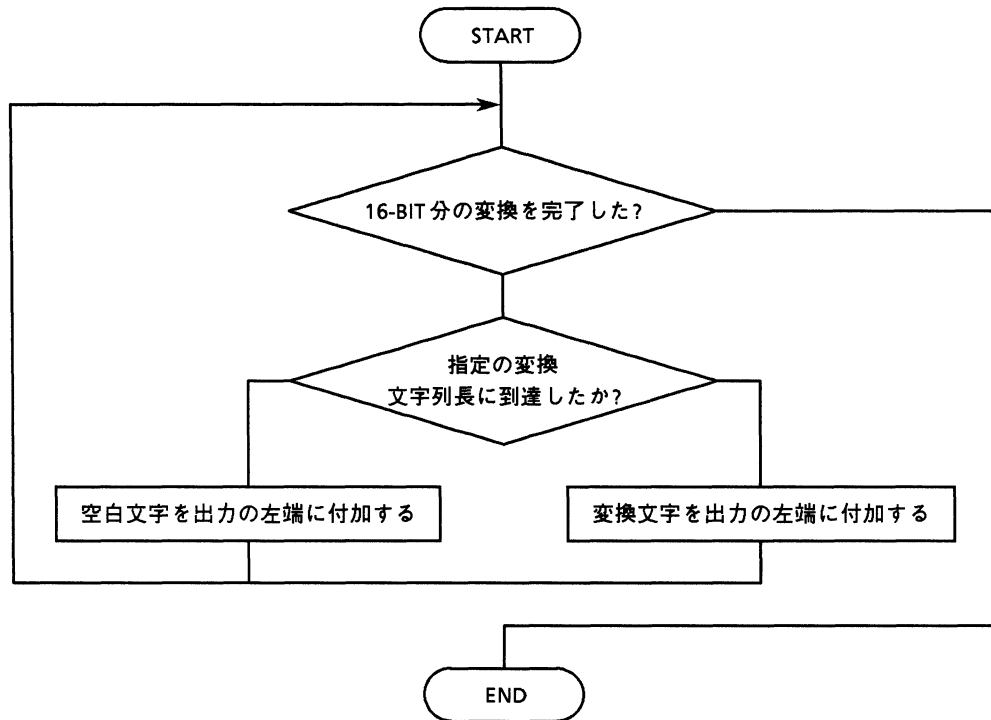
' ---- Exchange 16-bits pattern data ----
' In   i%:16bits pattern data (used integer)
'
' Procedure for swap of low byte and high byte .
' This program is bit manipulation of integer value. Why this program used
' real value because one is overflow detect on bit manipulation of integer
' value, another one is internal manipulation by real value although input
' parameter is integer. And integer declare value is same operation.
'
FUNCTION Exchange% (i%)
  h = i% AND &HFF
  l = i% AND &HFF00
  IF h < 0 THEN
    h = 0
  END IF

  a = INT(h * 256) + ((l ¥ 256) AND &HFF)
  IF a >= 32768 THEN
    b = a - 32768
    a = -32768 + b
  END IF
  Exchange% = a
END FUNCTION

```

(1.4) itob\$(l%, v%): 整数を2進文字列に変換する

● フローチャート



● プログラム・リスト

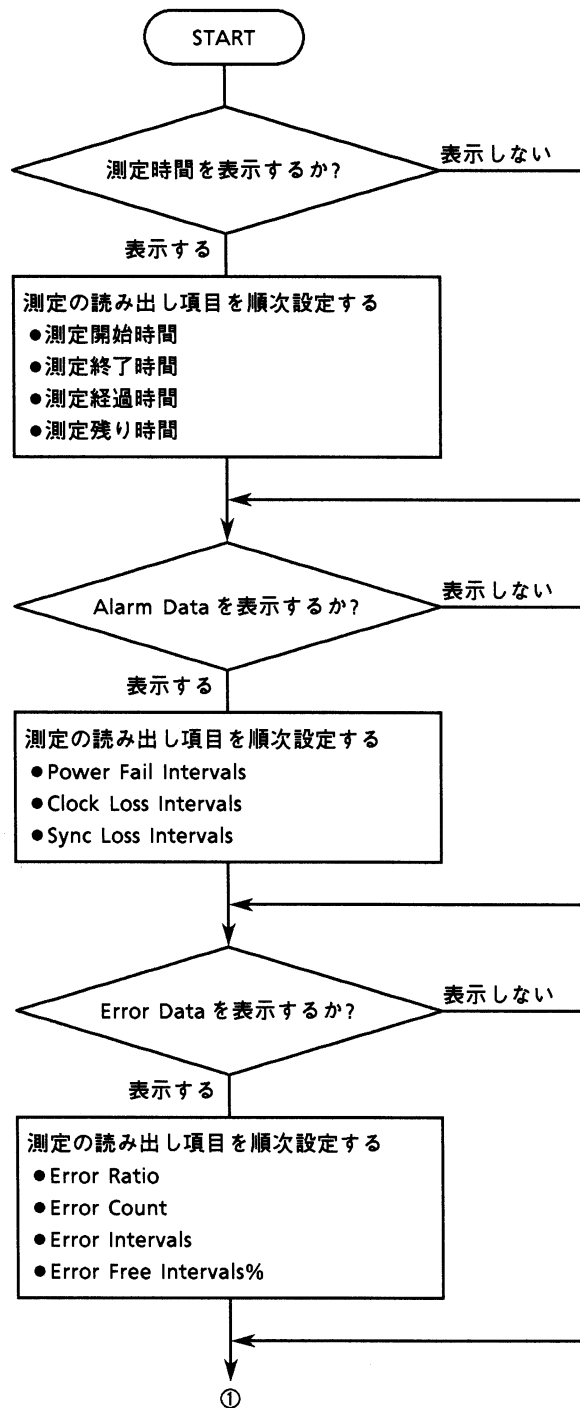
```

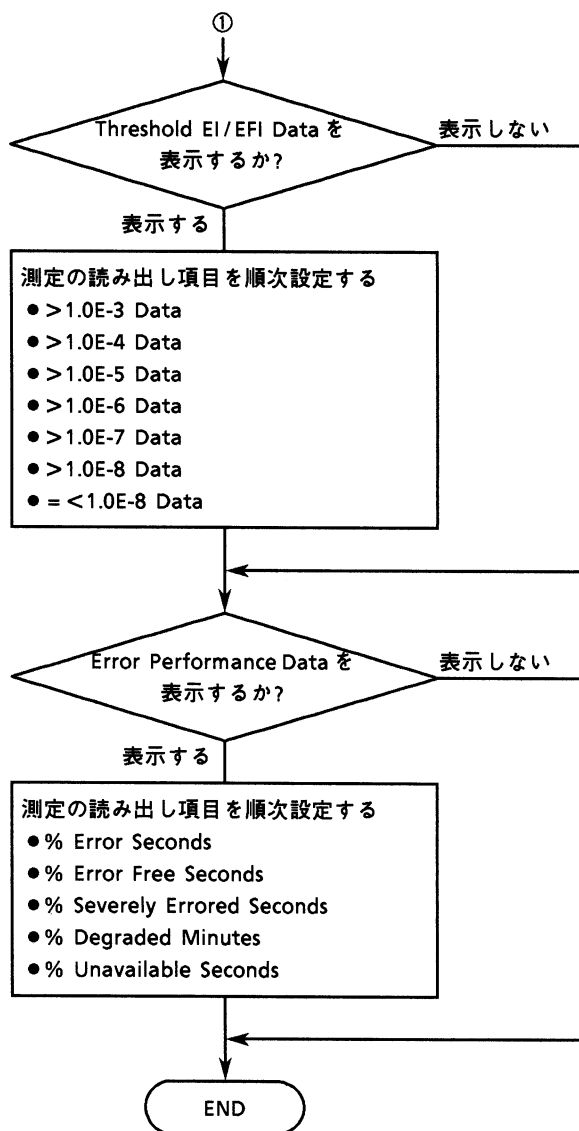
' ---- Interger convert to binary strings ----
' in    l%:convert binary length
'       v%:convert value
'
' This convert binary strings is always possession 16-character field.
'
FUNCTION itob$ (l%, v%)
b$ = ""
FOR i% = 1 TO 16
  IF i% <= l% THEN
    IF v% AND &H1 THEN
      b$ = "1" + b$
    ELSE
      b$ = "0" + b$
    END IF
    v% = INT(v% / 2)
  ELSE
    b$ = " " + b$
  END IF
NEXT i%
itob$ = b$
END FUNCTION

```

(1.5) Selltem(dst\$): 測定結果読み出しコマンドの引き数を設定する

● フローチャート





● プログラム・リスト

```

SUB SelItem (dst$())
LOCATE 16, 1          '===== Select measurement item(s) =====
PRINT "Choose item for measure time.          "
LOCATE 17, 1
INPUT "Do you wish output measure TIME data?      [Yes/No]:"; a$
IF a$ = "y" OR a$ = "Y" THEN
    dst$(0, 0) = "0,1"
    dst$(0, 1) = "0,2"
    dst$(0, 2) = "0,3"
    dst$(0, 3) = "0,4"
END IF

CALL ClearDisp(16, 2)
LOCATE 16, 1
PRINT "Choose item for alarm data          "
LOCATE 17, 1
INPUT "Do you wish output ALARM data?            [Yes/No]:"; a$
IF a$ = "y" OR a$ = "Y" THEN
    dst$(1, 0) = "1,1"
    dst$(1, 1) = "1,2"
    dst$(1, 2) = "1,3"
END IF

CALL ClearDisp(16, 2)
LOCATE 16, 1
PRINT "Choose item for ERROR measurement data. "
LOCATE 17, 1
INPUT "Do you wish output ERROR data?          [Yes/No]:"; a$
IF a$ = "y" OR a$ = "Y" THEN
    dst$(2, 0) = "2,1"
    dst$(2, 1) = "2,2"
    dst$(2, 2) = "2,3"
    dst$(2, 3) = "2,4"
END IF

CALL ClearDisp(16, 2)
LOCATE 16, 1
PRINT "Choose item for THRESHOLD EI/EFI data.  "
LOCATE 17, 1
INPUT "Do you wish output THR. EI/EFI data ?    [Yes/No]:"; a$
IF a$ = "y" OR a$ = "Y" THEN
    dst$(3, 0) = "3,1"
    dst$(3, 1) = "3,2"
    dst$(3, 2) = "3,3"
    dst$(3, 3) = "3,4"
    dst$(3, 4) = "3,5"
    dst$(3, 5) = "3,6"
    dst$(3, 6) = "3,7"
END IF

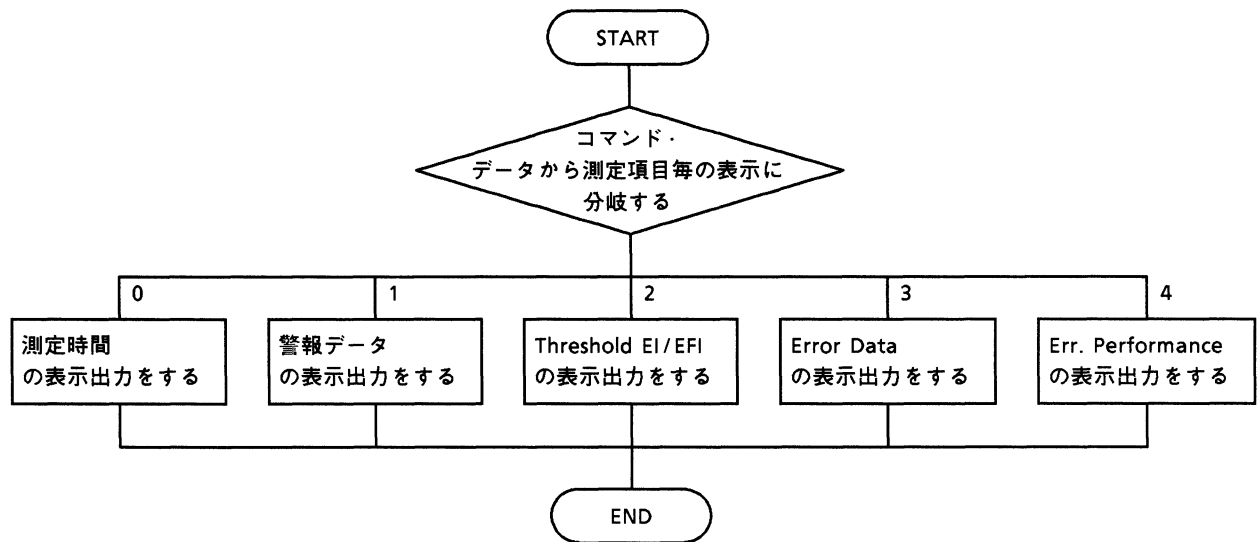
CALL ClearDisp(16, 2)
LOCATE 16, 1
PRINT "Choose item for ERROR PERFORMANCE data. "
LOCATE 17, 1
INPUT "Do you wish output EP data data ?        [Yes/No]:"; a$
IF a$ = "y" OR a$ = "Y" THEN
    dst$(4, 0) = "4,1"
    dst$(4, 1) = "4,2"
    dst$(4, 2) = "4,3"
    dst$(4, 3) = "4,4"
    dst$(4, 4) = "4,5"
END IF

PRINT " -- End select measurement item -- "
END SUB

```


(1.6) Disp1 (cmd\$,dt\$) : 読み出した測定結果を項目毎に表示する

- フローチャート



● プログラム・リスト

```

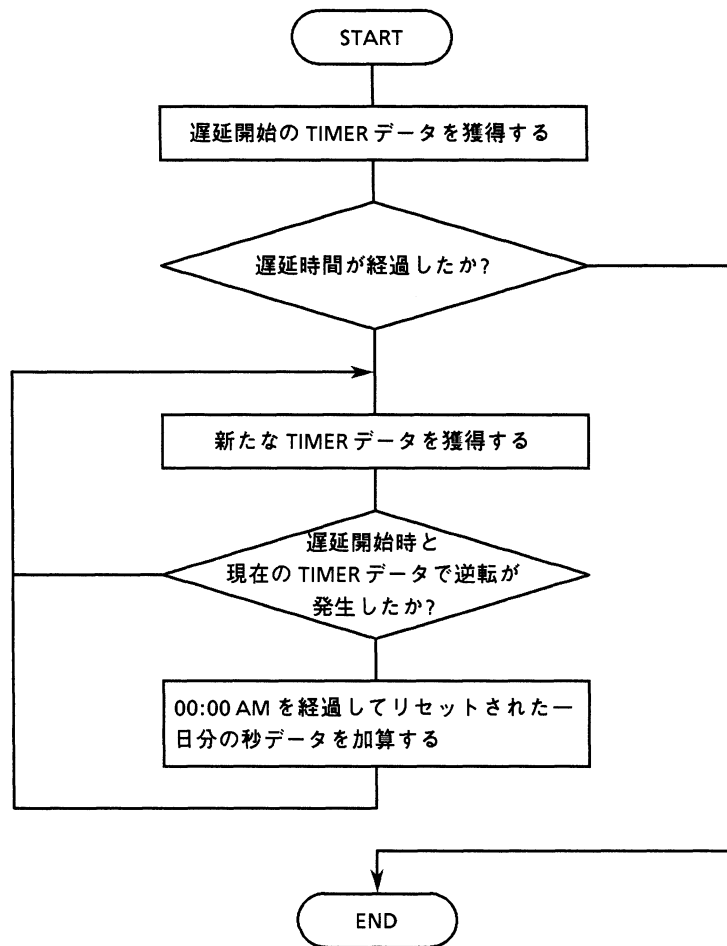
' ---- Procedure for Measure result display ----
' in      cmd$:parameter strings of query
'      dt$:response message from MP1762D/MP1764D(ED)
'
SUB Displ (CMD$, DT$)
  unit$ = ""
  SELECT CASE VAL(MID$(CMD$, 1, 1))
  CASE 0
    ' Time data
    SELECT CASE VAL(MID$(CMD$, 3, 1))
    CASE 1
      ttl$ = "Start time" : "
    CASE 2
      ttl$ = "Stop time" : "
    CASE 3
      ttl$ = "Elapsed time" : "
    CASE 4
      ttl$ = "Remain time" : "
    END SELECT
  CASE 1
    ' Alarm data
    SELECT CASE VAL(MID$(CMD$, 3, 1))
    CASE 1
      ttl$ = "Power Fail Intervals" : "
    CASE 2
      ttl$ = "Clock Loss Intervals" : "
    CASE 3
      ttl$ = "Sync Loss Intervals" : "
    END SELECT
  CASE 2
    ' Error data
    SELECT CASE VAL(MID$(CMD$, 3, 1))
    CASE 1
      ttl$ = "Error Ratio" : "
    CASE 2
      ttl$ = "Error Count" : "
    CASE 3
      ttl$ = "Error Intervals" : "
    CASE 4
      ttl$ = "Error Free Intervals%" : "
      unit$ = "%"
    END SELECT
  CASE 3
    ' Threshold EI/EFI data
    unit$ = "%"
    SELECT CASE VAL(MID$(CMD$, 3, 1))
    CASE 1
      ttl$ = "Threshold EI/EFI >1.0E-3:"
    CASE 2
      ttl$ = "Threshold EI/EFI >1.0E-4:"
    CASE 3
      ttl$ = "Threshold EI/EFI >1.0E-5:"
    CASE 4
      ttl$ = "Threshold EI/EFI >1.0E-6:"
    CASE 5
      ttl$ = "Threshold EI/EFI >1.0E-7:"
    CASE 6
      ttl$ = "Threshold EI/EFI >1.0E-8:"
    CASE 7
      ttl$ = "Threshold EI/EFI =<1.0E-8:"
    END SELECT
  CASE 4
    ' Error Performance data
    unit$ = "%"
    SELECT CASE VAL(MID$(CMD$, 3, 1))
    CASE 1
      ttl$ = "Error Performance %ES" : "
    CASE 2

```

```
        ttl$ = "Error Performance %EFS    :"  
CASE 3  
        ttl$ = "Error Performance %SES    :"  
CASE 4  
        ttl$ = "Error Performance %DM     :"  
CASE 5  
        ttl$ = "Error Performance %US     :"  
END SELECT  
END SELECT  
IF DT$ = "ERR" THEN  
    DT$ = " No data "  
END IF  
PRINT ttl$ + DT$ + unit$ + "  "  
END SUB
```

(1.7) waidly (tim!) : 遅延時間を作る

● フローチャート



● プログラム・リスト

```

' ---- Make a timing delay ----
'in   tim:wait time length (unit is seconds)
'
SUB waidly (tim)
  stm = TIMER
  etm = TIMER
  WHILE etm - stm < tim
    etm = TIMER
    IF etm < stm THEN etm = etm + 86400
  WEND
END SUB

```

<ACS_GPIB.BASの説明>

ACS_GPIB.BASは下記の15種類の関数で構成されています。

表10-3 ACS_GPIB.BASの関数表

(1/2)

モジュール 番号	関 数 名	処 理 概 要
2.1	SUB wrtcmd1 (w\$)	PPGにコマンドを発行する w\$: 発行するコマンド文字列 (入力)
2.2	SUB wrtcmd2 (w\$)	EDにコマンドを発行する w\$: 発行するコマンド文字列 (入力)
2.3	FUNC readcmd2\$()	EDからメッセージをリードする readcmd2\$: メッセージの文字列 (戻り値)
2.4	SUB dmawrt (w%(), i%)	EDへのDMA転送処理をする w%() : 転送するパターン・データの整数配列 i% : 整数配列の要素数
2.5	SUB EndPoll()	MSSステータスレジスタのENDビットをポーリングする
2.6	SUB SRQPoll()	MSSステータスレジスタのERRORビットおよびSRQビットをポーリングする
2.7	SUB StatusMask(sre%, ese%, ese2%, ese3%)	ステータス・レジスタ、イベント・レジスタおよび拡張イベント・レジスタのマスク・レジスタを設定する sre% : ステータス・レジスタのマスク・パターン ese% : 標準イベント・レジスタのマスク・パターン ese2% : 拡張イベント・レジスタ-2のマスク・パターン ese3% : 拡張イベント・レジスタ-3のマスク・パターン
2.8	SUB StatusDisp(stb%, esr%, esr2%, esr3%)	ステータス・レジスタ、イベント・レジスタおよび拡張イベント・レジスタの設定状態を表示する、読み出したデータは引き数に設定され呼出側に渡される stb% : ステータス・レジスタの設定状態パターン esr% : 標準イベント・レジスタの設定状態パターン esr2% : 拡張イベント・レジスタ-2の設定状態パターン esr3% : 拡張イベント・レジスタ-3の設定状態パターン
2.9	SUB MeasStart()	測定を開始する
2.10	SUB MeasStop()	測定の停止または中断をする
2.11	SUB ChecClk()	CLOCK LOSSを判定し、復帰する迄処理をホールドする

表10-3 ACS_GPIB.BAS の関数表

(2/2)

モジュール 番号	関 数 名	処 理 概 要
2.12	FUNC AutoSrc%()	Auto Search を実行し、その結果を関数値として返す 0 (False): 何らかの原因で異常終了した 1 (True): 正常に処理を終了した
2.13	FUNC gpinit%()	GPIB の初期化を実行し、その結果を関数値として返す 0 (False): 設定に異常が有り、GPIB の初期化に失敗した 1 (True): ED, PPG の一方または双方共に初期化が完了した
2.14	SUB trap()	システムエラー時の処理を行う
2.15	SUB gpiberr()	National Instruments Cop. の提供する GPIB サンプル・プログラムに含まれる内部エラー、ステータス情報の表示処理

次ページ以降に各関数のフローチャートとプログラム・リストを掲載します。

また本モジュールの先頭に、

```
REM $INCLUDE: 'C:\at-gbib\qbbasic\qbdecl.bas' ..... ①
COMMON SHARED DEV%, GPIBØ%, PPG%, ED% ..... ②
```

と記述してください。

①は、NI 社の GPIB ドライバで提きょうされている NI-488関数の定義をロードしています。

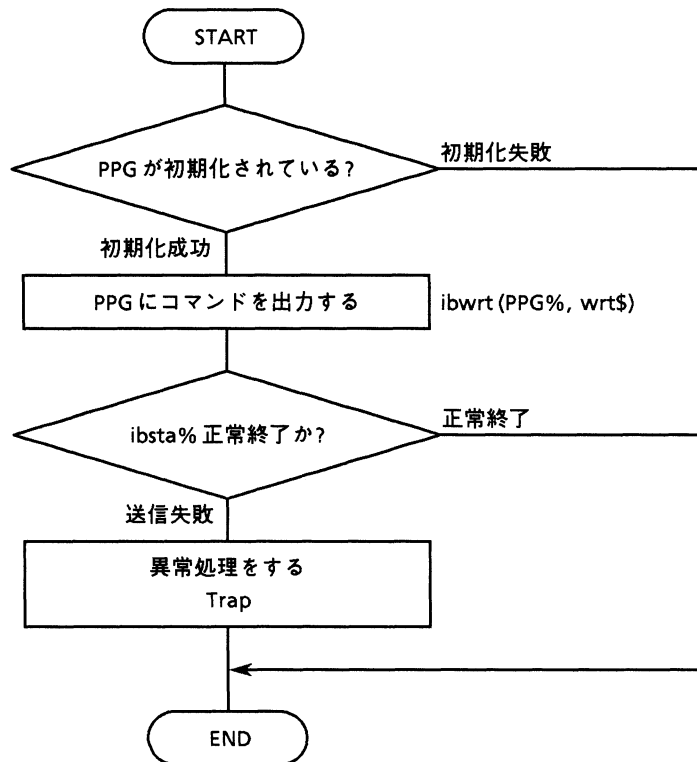
実際に御使用される為は、'qbdecl. bas' が存在しているディレクトリを指定してください。

②は、Quick Basic のステートメントで、複数モジュール間で共有する変数の定義をしています。

■ Note : ①に関してはお使いになられる GPIB ドライバおよび使用環境によって異なりますのでご注意ください。

(2.1) SUB wrtcmd1 (w\$) : PPG にコマンドを送信する

● フローチャート



● プログラム・リスト

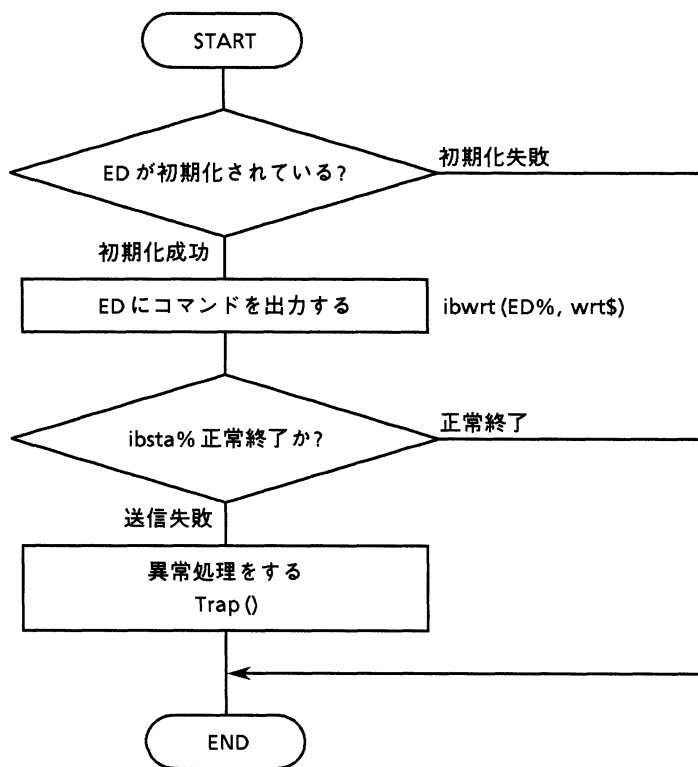
```

' ---- Procedure for command write to PPG ----
'
SUB wrtcmd1 (WRT$)
  IF DEV% = 1 OR DEV% = 3 THEN
    WRT$ = WRT$ + CHR$(13) + CHR$(10)
    CALL IBWRT(PPG%, WRT$)
    IF IBSTA% < 0 THEN CALL trap
  END IF
END SUB
'write command(ppg)
'call trap if illegal end

```

(2.2) SUB wrtcmd2 (w\$) : ED にコマンドを送信する

● フローチャート



● プログラム・リスト

```

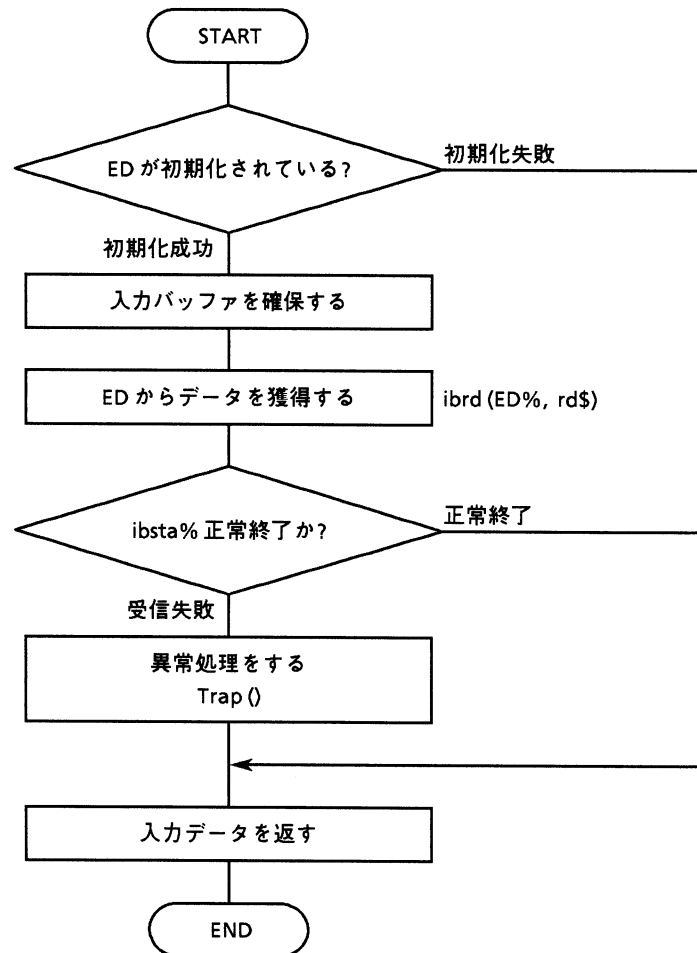
' ---- Procedure for command write to ED ----
'
SUB wrtcmd2 (WRT$)
  IF DEV% = 2 OR DEV% = 3 THEN
    WRT$ = WRT$ + CHR$(13) + CHR$(10)
    CALL IBWRT(ED%, WRT$)
    IF IBSTA% < 0 THEN CALL trap
  END IF
END SUB

```

' write command to ED
' call trap if illegal end

(2.3) FUNCTION readcmd2\$(): EDから別に送信したコマンドに対応するデータを獲得する

● フローチャート



● プログラム・リスト

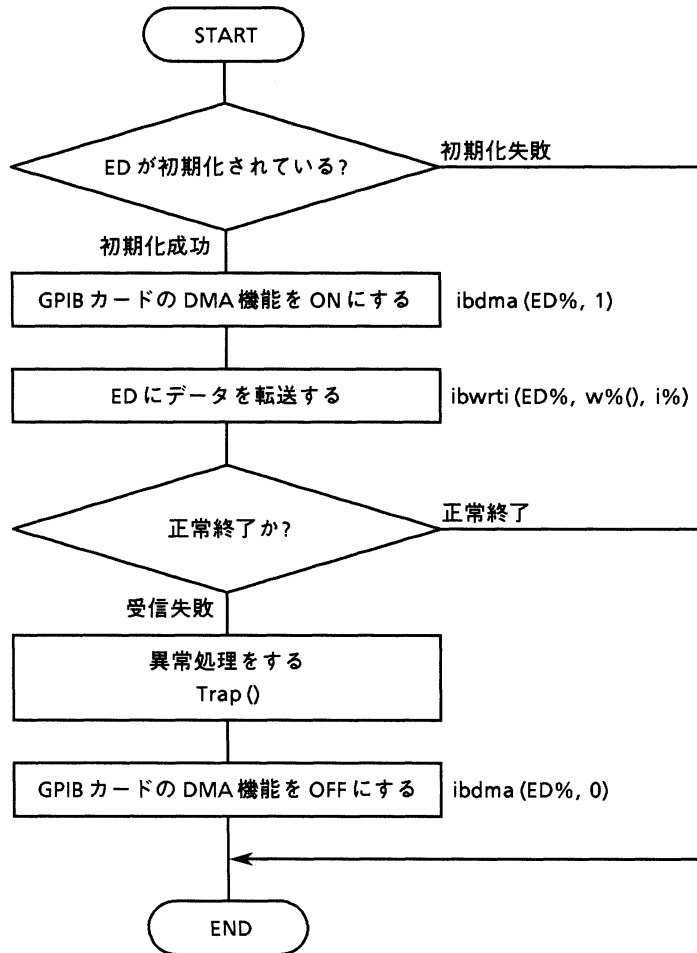
```

' ---- Procedure for data read from ED ----
,
FUNCTION readcmd2$
  IF DEV% = 2 OR DEV% = 3 THEN
    r$ = SPACE$(256)
    CALL IBRD(ED%, r$)          ' Read data from ED%
    IF IBSTA% < 0 THEN CALL trap
  ,
    readcmd2$ = r$
  END IF
END FUNCTION

```

(2.4) SUB dmawrt (w%, i%) : ED にデータを DMA 転送する

● フローチャート



● プログラム・リスト

```

' --- Procedure for DMA transfer ----
' in   w%():Transmit data pattern of integer array
'      i   :length count for integer array
SUB dmawrt (w%(), i%)
  IF DEV% = 2 OR DEV% = 3 THEN
    CALL IBDMA(ED%, 1)          ' DMA enable

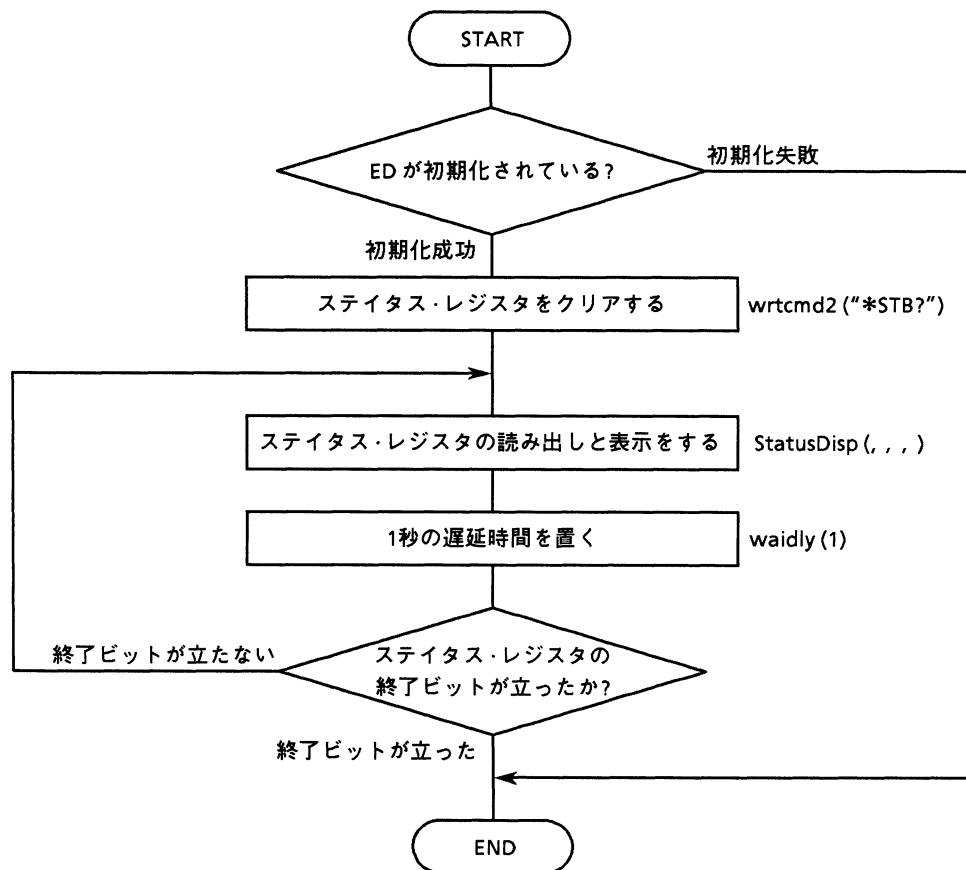
    i% = i% * 2 + 1             ' make up to a byte count
    CALL IBWRTI(ED%, w%(), i%)
    IF IBSTA% < 0 THEN CALL trap ' call trap if illegal end

    CALL IBDMA(ED%, 0)          ' DMA disable
  END IF
END SUB

```

(2.5) SUB EndPoll () : ステータスの終了ビットが設定される迄待つ

● フローチャート



● プログラム・リスト

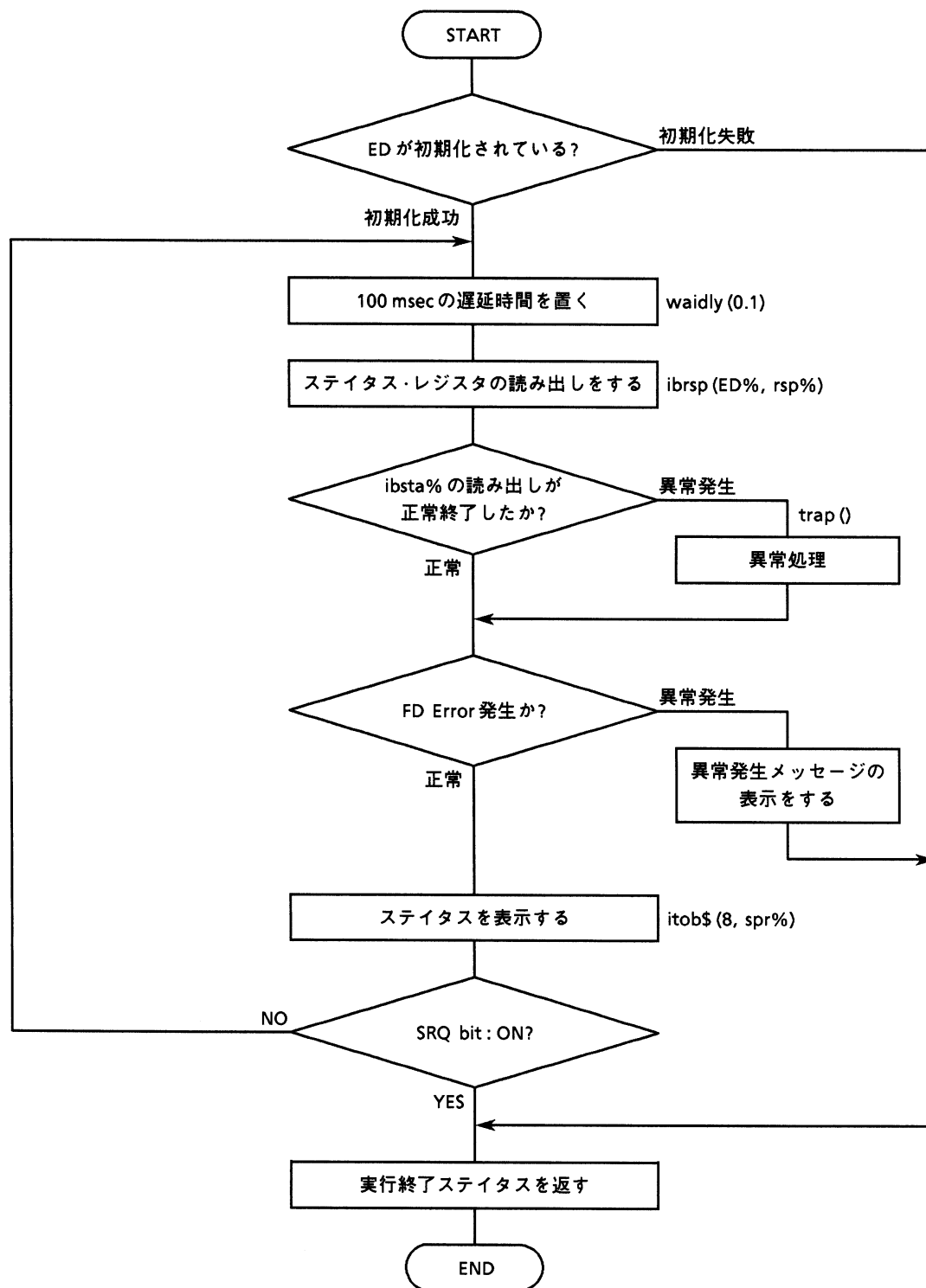
```

' ---- Procedure for judgement of Measurement end ----
'
SUB EndPoll
  IF DEV% = 2 OR DEV% = 3 THEN
    CALL wrtcmd2 ("*STB?")          ' reset event flag
    RD$ = LEFT$(readcmd2$, IBCNT% - 1)
    DO
      CALL StatusDisp(reg%, dmy%, dmy2%, dmy3%)
      waidly (1)
    LOOP UNTIL reg% AND &H4
  END IF
END SUB

```

(2.6) FUNCTION SRQPoll () : SRQ bit と Error bit の判定をする

● フローチャート



● プログラム・リスト

```

' ---- Procedure for Serial poll with SRQ bit ----
'
FUNCTION SRQPoll%
  IF DEV% = 2 OR DEV% = 3 THEN
    exe% = 1
    DO
      waidly (.1)

      CALL IBRSP(ED%, SPR%)
      IF IBSTA < 0 THEN CALL trap
      srq = SPR% AND &H40
      esr1 = SPR% AND &H4
      esr2 = SPR% AND &H8

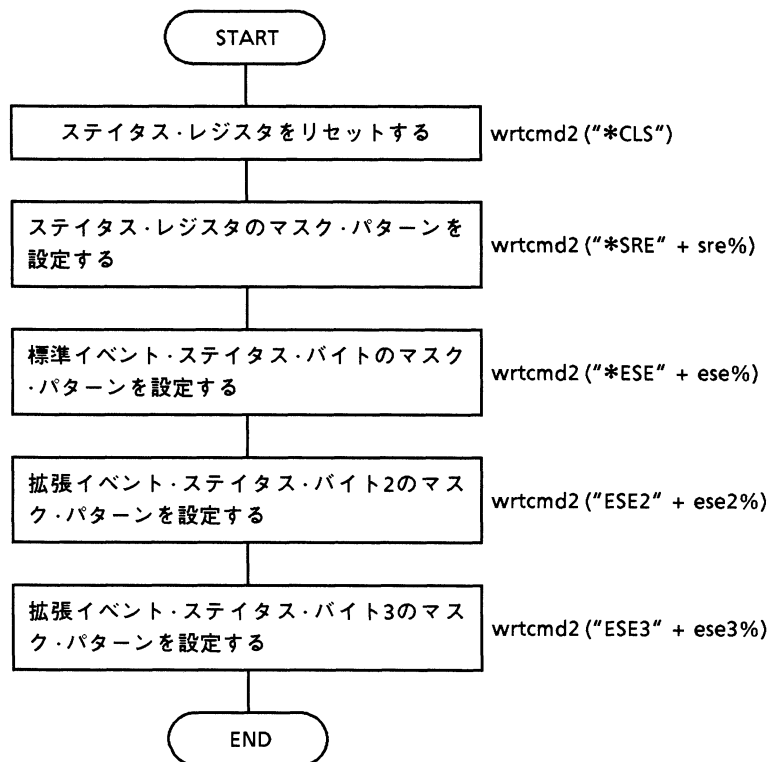
      IF esr2 = &H8 THEN          ' Output warning message, if error detect
        LOCATE 12, 35
        PRINT "FD error detect!!"
        exe% = 0
        EXIT DO
      END IF

      sta$ = itob$(8, SPR%)
      LOCATE 1, 60
      PRINT "*STB:"; sta$
    LOOP UNTIL srq = &H40 AND esr1 = &H0
  END IF
  SRQPoll% = exe%
END FUNCTION

```

(2.7) SUB StatusMask (sre%, ese2%, ese3%) :
ステータス・マスク・レジスタの設定

● フローチャート



● プログラム・リスト

```

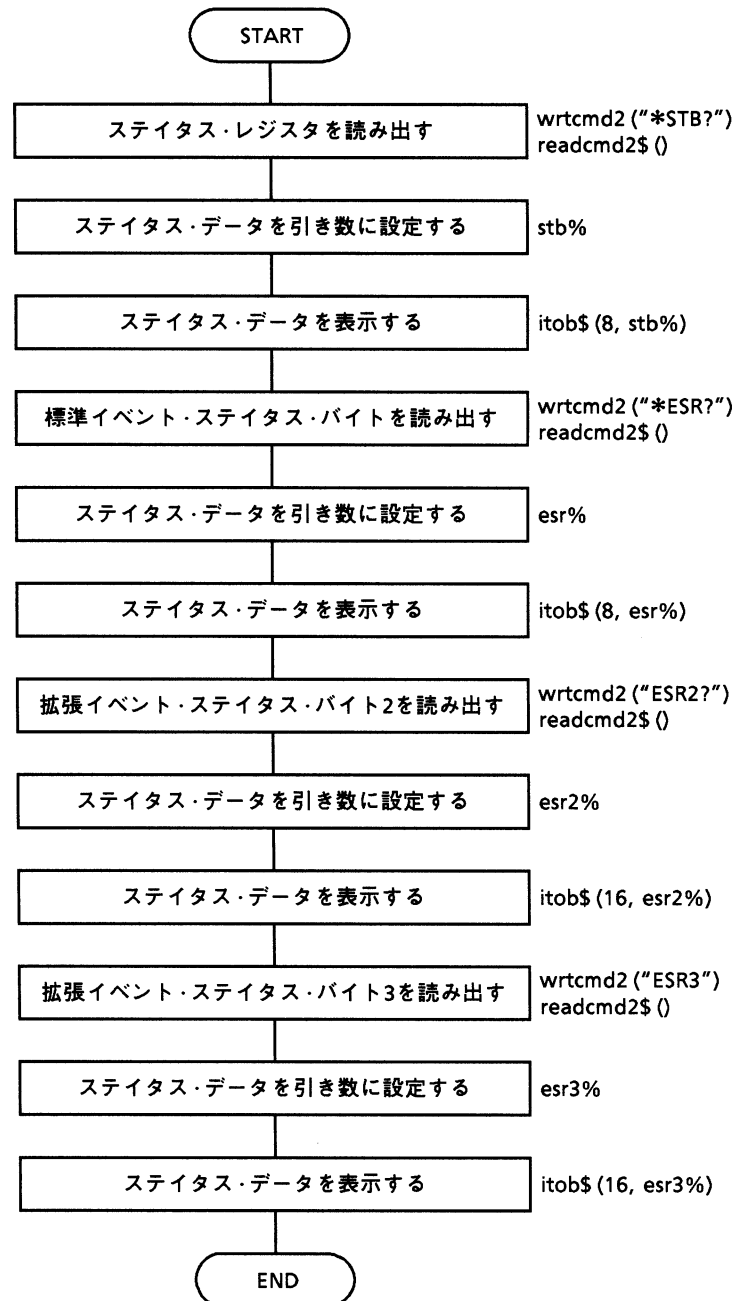
' ---- Procedure for set status mask pattern ----
' in      s0%:status byte enable register mask pattern
'         s1%:normal event status enable register mask pattern
'         s2%:Extend event status enable register-2 mask pattern
'         s3%:Extend event status enable register-3 mask pattern
'
SUB StatusMask (s0%, s1%, s2%, s3%)
    wrtcmd2 ("*CLS")
    wrtcmd2 ("*SRE " + STR$(s0%))
    wrtcmd2 ("*ESE " + STR$(s1%))
    wrtcmd2 ("ESE2 " + STR$(s2%))
    wrtcmd2 ("ESE3 " + STR$(s3%))
END SUB

```

(2.8) SUB StatusDisp (stb%, esr%, esr2%, esr3%) :

ステータス・レジスタの読み出しおよび表示をする

● フローチャート



● プログラム・リスト

```

' ---- Procedure for status byte display ----
' out   stb% :Status byte
'       esr% :Normal event status byte
'       esr2%:Extend event-2 status byte
'       esr3%:Extend event-3 status byte
'
SUB StatusDisp (stb%, esr%, esr2%, esr3%)
  CALL wrtcmd2("*STB?")
  RD$ = LEFT$(readcmd2$, IBCNT% - 1)
  stb% = VAL(RD$)
  sta$ = itob$(8, VAL(RD$))
  LOCATE 1, 60
  PRINT "*STB:"; sta$

  CALL wrtcmd2("*ESR?")
  RD$ = LEFT$(readcmd2$, IBCNT% - 1)
  esr% = VAL(RD$)
  sta$ = itob$(8, VAL(RD$))
  LOCATE 2, 60
  PRINT "*ESR:"; sta$

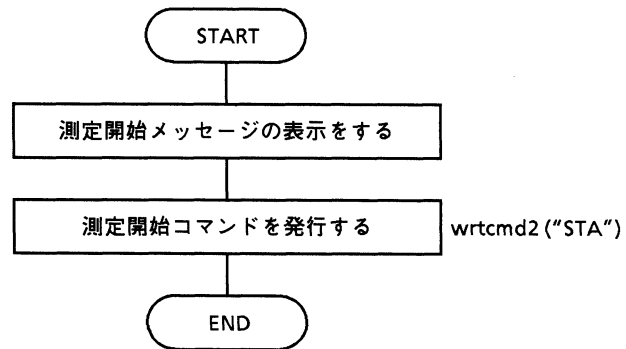
  CALL wrtcmd2("ESR2?")
  RD$ = LEFT$(readcmd2$, IBCNT% - 1)
  esr2% = VAL(MID$(RD$, 6, 5))
  sta$ = itob$(16, VAL(MID$(RD$, 6, 5)))
  LOCATE 3, 60
  PRINT "ESR2:"; sta$

  CALL wrtcmd2("ESR3?")
  RD$ = LEFT$(readcmd2$, IBCNT% - 1)
  esr3% = VAL(MID$(RD$, 6, 5))
  sta$ = itob$(16, VAL(MID$(RD$, 6, 5)))
  LOCATE 4, 60
  PRINT "ESR3:"; sta$
END SUB

```


(2.9) SUB MeasStart ()

● フローチャート



● プログラム・リスト

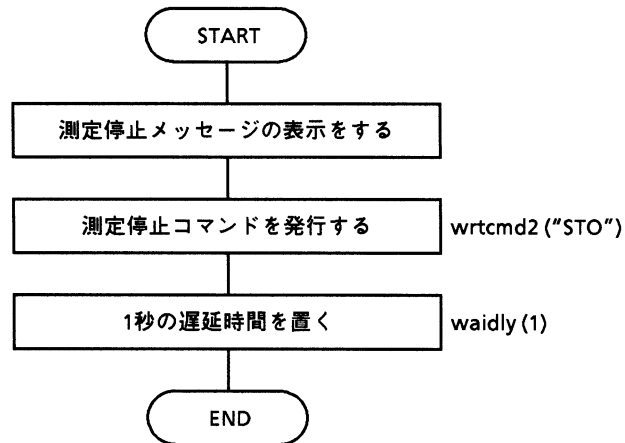
```

' ---- Procedure for Measurement start ----
'
SUB MeasStart
  CLS
  LOCATE 1, 1
  PRINT "***** Measure START *****"
  CALL wrtcmd2("STA")
END SUB

```

(2.10) SUB MeasStop ()

- フローチャート



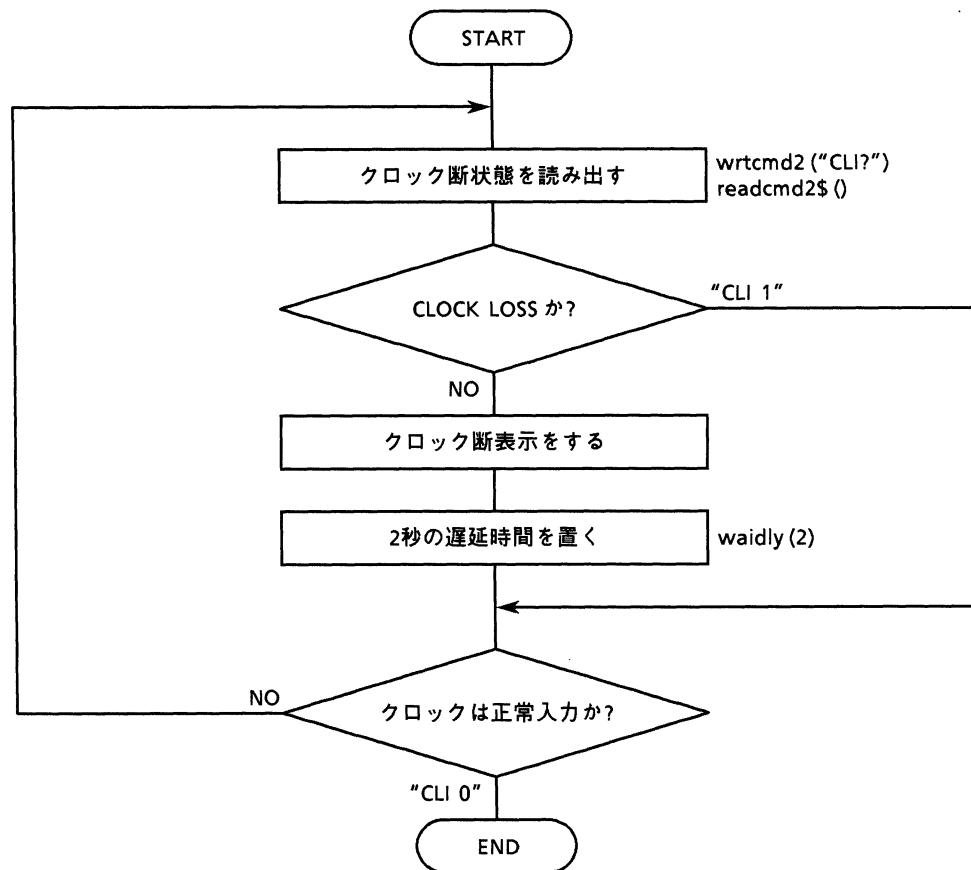
- プログラム・リスト

```

! ---- Procedure for Measurement stop ----
!
SUB MeasStop
  LOCATE 1, 1
  PRINT "***** Measure STOP *****"
  CALL wrtcmd2("STO")
  waidly (1)
END SUB
  
```

(2.11) SUB ChecClk () : クロック断の状態確認をする

● フローチャート



● プログラム・リスト

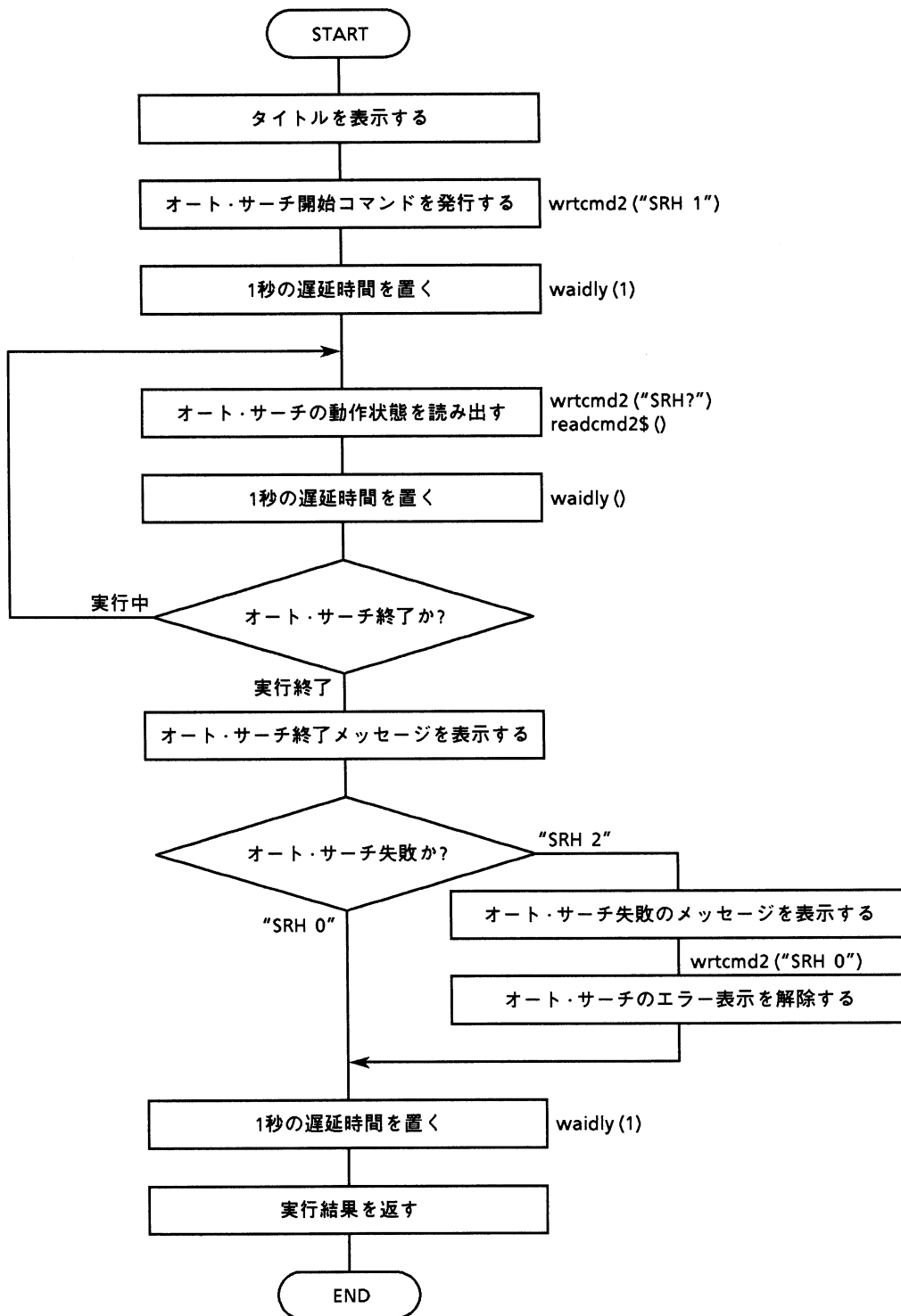
```

' ---- Procedure for clock status ----
' This program is loop until to clock detect.
'
SUB ChecClk
DO
  CALL wrtcmd2("CLI?")
  RD$ = LEFT$(readcmd2$, IBCNT% - 1)
  IF MID$(RD$, 1, 5) = "CLI 1" THEN
    LOCATE 4, 1
    PRINT "*** CLOCK LOSS ***"
    waidly(2)
  END IF
  LOOP UNTIL MID$(RD$, 1, 5) = "CLI 0"
  LOCATE 4, 1
  PRINT "          "
END SUB

```

(2.12) FUNCTION AutoSrc% ()

● フローチャート



● プログラム・リスト

```

' ---- Procedure for Auto Search ----
' out   AutoSrc%:Auto Search execution status
'       0(false):illegal termination
'       1(true) :normal end
'
FUNCTION AutoSrc%
  LOCATE 4, 1
  PRINT "*** Auto Search START ***"

  '===== Auto search ON =====
  CALL wrtcmd2("SRH 1")
  waidly (1)

  '===== Polling =====
  DO
    CALL wrtcmd2("SRH?")
    RD$ = LEFT$(readcmd2$, IBCNT% - 1)
    waidly (1)
  LOOP UNTIL MID$(RD$, 1, 5) = "SRH 0" OR MID$(RD$, 1, 5) = "SRH 2"
  rs1% = 1                                ' Auto Search success

  LOCATE 4, 1
  PRINT "*** Finish Auto Search ***"

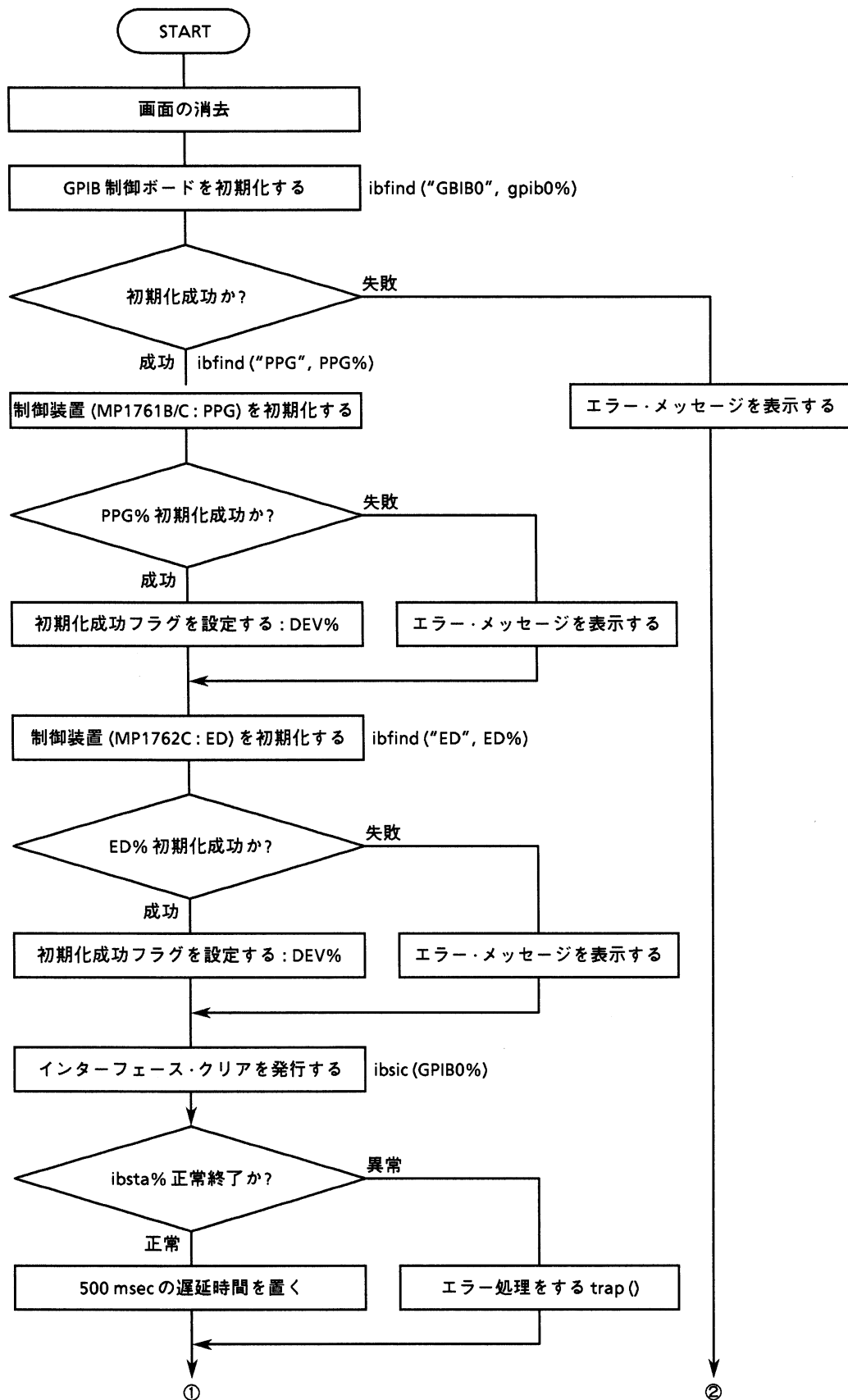
  '===== Fail Auto Search =====
  IF MID$(RD$, 1, 5) = "SRH 2" THEN
    PRINT "<<< Failed on AUTO SEARCH ! >>>"
    CALL wrtcmd2("SRH 0")
    rs1% = 0                                ' Auto Search fail
  END IF
  waidly (1)

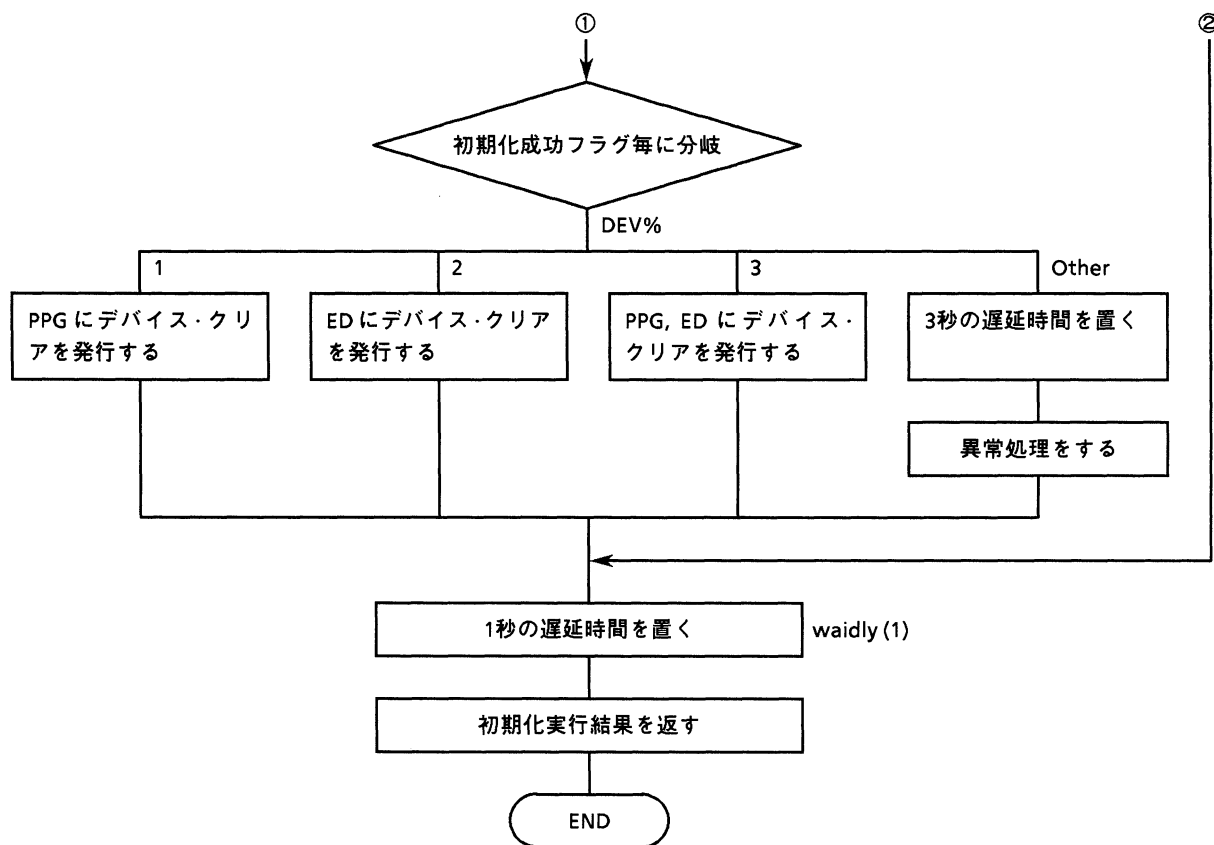
  AutoSrc% = rs1%
END FUNCTION

```

(2.13) FUNCTION gpinit () : GPIB 制御環境の初期化をする

● フローチャート





● プログラム・リスト

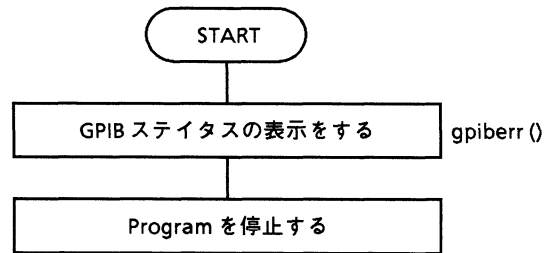
```

' ---- Procedure for initialize equipments and interface board ----
'
FUNCTION gpinit%
  CLS
  CALL IBFIND("GPIB0", GPIB0%)      'Open DEvIce (GPiB0)
  IF GPIB0% < 0 THEN
    PRINT "Configuration fail!!"
    PRINT "You need verify are hardware condition, and try again."
    ret% = 0
  ELSE
    CALL IBFIND("PPG", PPG%)        'Open DEvIce (PPG)
    IF PPG% < 0 THEN
      PRINT "Lost PPG address!!"
      PRINT "If you use a PPG, then verify are configuration and environmen
t."
      DEV% = 0
    ELSE
      DEV% = 1
    END IF
    CALL IBFIND("ED", ED%)          'Open DEvIce (ED)
    IF ED% < 0 THEN
      PRINT "Lost ED address!!"
      PRINT "If you use a PPG, then verify are configuration and environmen
t."
    ELSE
      IF DEV% = 0 THEN
        DEV% = 2
      ELSE
        DEV% = 3
      END IF
    END IF
    CALL IBSIC(GPIB0%)              'Interface clear
    IF IBSTA% < 0 THEN CALL trap
    CALL waidly(.5)                  '500ms wait
    SELECT CASE DEV%
      CASE 1
        CALL IBCLR(PPG%)            'DEvIce clear (PPG)
      CASE 2
        CALL IBCLR(ED%)             'DEvIce clear (ED)
      CASE 3
        CALL IBCLR(PPG%)            'DEvIce clear (PPG)
        CALL IBCLR(ED%)            'DEvIce clear (ED)
      CASE ELSE
        waidly (3)
        CALL trap
    END SELECT
    ret% = 1
  END IF
  waidly (1)
  CLS
  gpinit% = ret%                    ' set Execution status
END FUNCTION

```


(2.14) SUB trap (msg\$) : 異常処理をする

● フローチャート



● プログラム・リスト

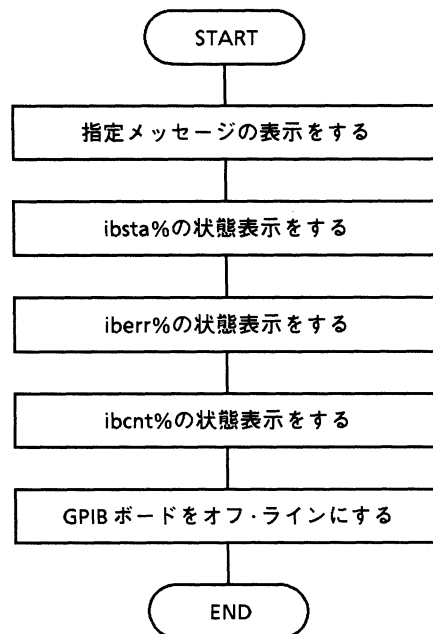
```

' ---- Procedure for illegal process trap ----
' This subroutine, call on illegal execution or fatal error detect.
' And, you will get are status condition by presented NI-488 function.
'
SUB trap
  CALL gpiberr("Program trap condition.") ' call NI subroutine
  STOP
END SUB

```

(2.15) SUB gpiberr (msg\$) STATIC : GPIB ステータスの表示をする

- フローチャート



● プログラム・リスト

```

=====
Subroutine GPIBERR
This subroutine will notify you that a NI-488 function failed by printing
an error message. The status variable IBSTA% will also be printed
in hexadecimal along with the mnemonic meaning of the bit position.
The status variable IBERR% will be printed in decimal along with the
mnemonic meaning of the decimal value. The status variable IBCNT% will
be printed in decimal.

The NI-488 function IBONL is called to disable the hardware and software.

The STOP command will terminate this program.
=====
SUB gpiberr (msg$) STATIC

PRINT msg$

PRINT "ibsta = &H"; HEX$(IBSTA%); " <";
IF IBSTA% AND EERR THEN PRINT " ERR";
IF IBSTA% AND TIMO THEN PRINT " TIMO";
IF IBSTA% AND EEND THEN PRINT " END";
IF IBSTA% AND SRQI THEN PRINT " SRQI";
IF IBSTA% AND RQS THEN PRINT " RQS";
IF IBSTA% AND SPOLL THEN PRINT " SPOLL";
IF IBSTA% AND EEVENT THEN PRINT " EVENT";
IF IBSTA% AND CMPL THEN PRINT " CMPL";
IF IBSTA% AND LOK THEN PRINT " LOK";
IF IBSTA% AND RREM THEN PRINT " REM";
IF IBSTA% AND CIC THEN PRINT " CIC";
IF IBSTA% AND AATN THEN PRINT " ATN";
IF IBSTA% AND TACS THEN PRINT " TACS";
IF IBSTA% AND LACS THEN PRINT " LACS";
IF IBSTA% AND DTAS THEN PRINT " DTAS";
IF IBSTA% AND DCAS THEN PRINT " DCAS";
PRINT " >"

PRINT "iberr = "; IBERR%;
IF IBERR% = EDVR THEN PRINT " EDVR <DOS Error>"
IF IBERR% = ECIC THEN PRINT " ECIC <Not CIC>"
IF IBERR% = ENOL THEN PRINT " ENOL <No Listener>"
IF IBERR% = EADR THEN PRINT " EADR <Address error>"
IF IBERR% = EARG THEN PRINT " EARG <Invalid argument>"
IF IBERR% = ESAC THEN PRINT " ESAC <Not Sys Ctrlr>"
IF IBERR% = EABO THEN PRINT " EABO <Op. aborted>"
IF IBERR% = ENEB THEN PRINT " ENEB <No GPIB board>"
IF IBERR% = EOIP THEN PRINT " EOIP <Async I/O in prg>"
IF IBERR% = ECAP THEN PRINT " ECAP <No capability>"
IF IBERR% = EFSO THEN PRINT " EFSO <File sys. error>"
IF IBERR% = EBUS THEN PRINT " EBUS <Command error>"
IF IBERR% = ESTB THEN PRINT " ESTB <Status byte lost>"
IF IBERR% = ESRQ THEN PRINT " ESRQ <SRQ stuck on>"
IF IBERR% = ETAB THEN PRINT " ETAB <Table Overflow>"

PRINT "ibcnt = "; IBCNT%

' Call the IBONL function to disable the hardware and software.

CALL IBONL(dvm%, 0)
END SUB

```

<プログラムの起動>

前述の共通関数と(1)～(13)のサンプル・プログラムを起動する手順を以下に記述します。

(手順1): メニューバーの **File** をオープンして、“**Load File. . .**” を選択します。

次に **Load** するファイル名 (共通関数の) **COMMON.BAS** をロードします。

(手順2): 手順1と同様にして、**ACS_GPIB.BAS** をロードします。

(手順3): 手順1と同様にして、サンプル・プログラムをロードします。

(手順4): メニューバーの **Run** をオープンして“**Set Main Module**”を選択し、手順3でロードしたサンプル・プログラムを **Main Module** とします。

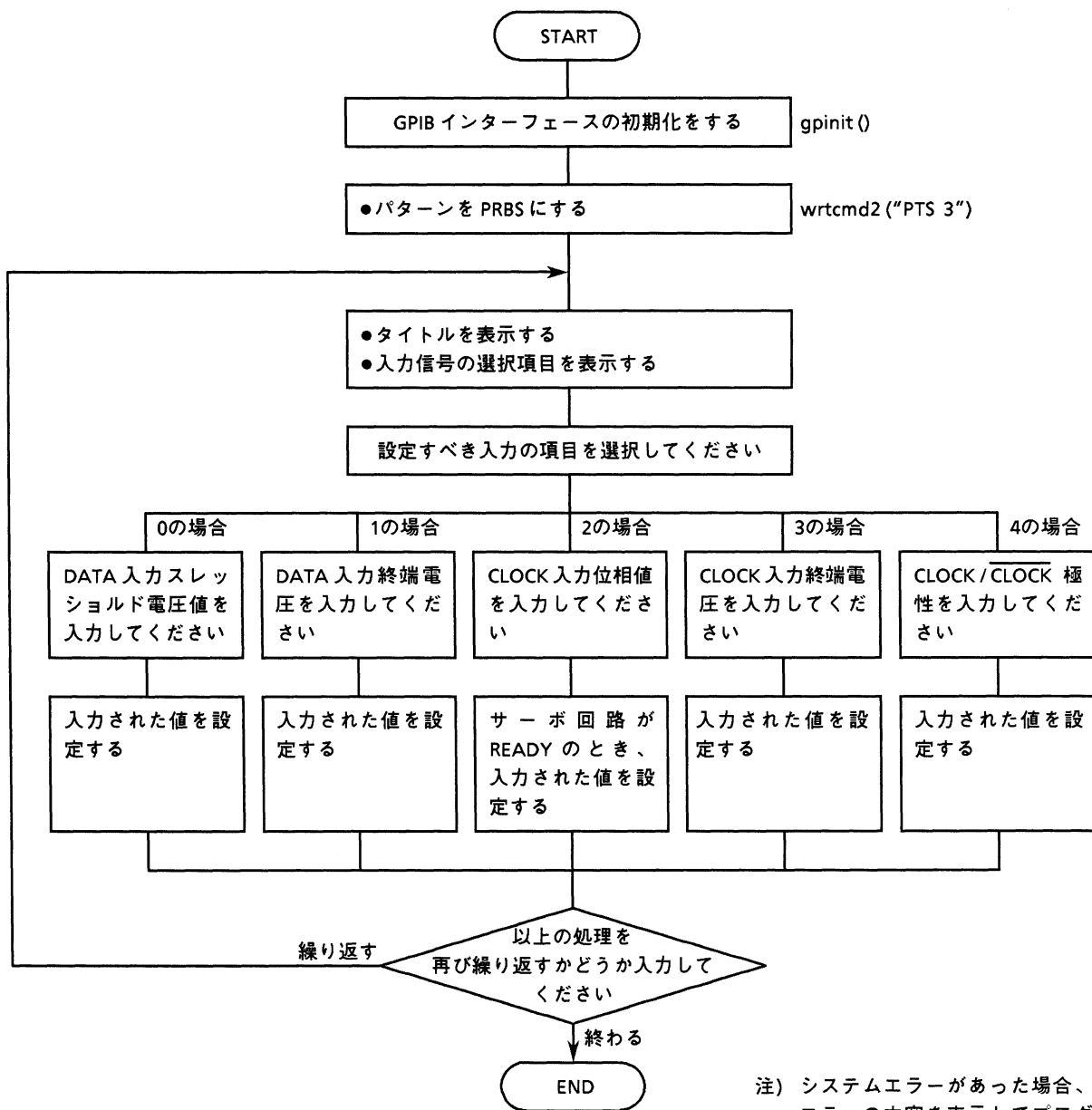
(手順5): メニューバーの **Run** をオープンして“**Start**”を実行するとプログラムがスタートします。

( 詳細は Quick Basic の取扱説明書を参照してください。)

(1) 入力信号の設定

本プログラムは入力信号関係 GPIB 制御を行います。

なおプログラム例では、入力を **DATA** のみとして扱っています。



注) システムエラーがあった場合、エラーの内容を表示してプログラムを STOP します。

● プログラム・リスト

```

REM $INCLUDE: 'c:\vat-gpib\qbasic\qbdecl.bas'

COMMON SHARED DEV%, GPIB0%, PPG%, ED%

DECLARE SUB waitdly (tim!)
DECLARE SUB wrtcmd2 (w$)
DECLARE FUNCTION gpinit% ()
DECLARE FUNCTION readcmd2$ ()

CLS
IF gpinit% <> 0 THEN          ' Initialize GPIB environment
DO
    CALL wrtcmd2("PTS 3")
    DO
        PRINT "** MP1762/MP1764A INPUT SIGNAL SAMPLE PROGRAM ** "
        PRINT
        PRINT " INPUT SIGNAL * DATA THRESHOLD      = [0] "
        PRINT "                  * DATA TERMINATION   = [1] "
        PRINT "                  * CLOCK PHASE ADJUST  = [2] "
        PRINT "                  * CLOCK TERMINATION    = [3] "
        PRINT "                  * CLOCK POLARITY       = [4] "
        PRINT
        INPUT "Choose function [0 to 4]:"; sel%

        IF sel% < 0 OR sel% > 4 THEN
            CLS
            PRINT "Wrong chosen number!!"
            PRINT "Please, enter correct number."
        END IF

        LOOP UNTIL sel% >= 0 AND sel% <= 4

        SELECT CASE sel%
        CASE 0
            PRINT "Please, type number for the DATA THRESHOLD."
            INPUT "Possible data range is -3.000 to +1.875V, STEP 0.001V."; dth$
            CALL wrtcmd2("DTH " + dth$)

        CASE 1
            INPUT "Choose DATA TERMINATION.[GND:0, -2V:1] "; dtm$
            CALL wrtcmd2("DTM " + dtm$)

        CASE 2
            PRINT "Please, type number for the CLOCK PHASE ADJUST."
            INPUT "Possible data range is -500 to 500ps, STEP by 1ps"; cpa$

            DO
                CALL wrtcmd2("DLY?")
                RD$ = readcmd2$
                IF MID$(RD$, 1, 5) = "DLY 0" THEN
                    EXIT DO
                ELSE
                    ' Wait status read timing delay for clock phase adjust
                    CALL waitdly(1)
                END IF
            LOOP

            WRT$ = "CPA " + cpa$: CALL wrtcmd2(WRT$)

        CASE 3
            INPUT "Choose CLOCK TERMINATION.[GND:0, -2V:1] "; ctm$
            CALL wrtcmd2("CTM " + ctm$)

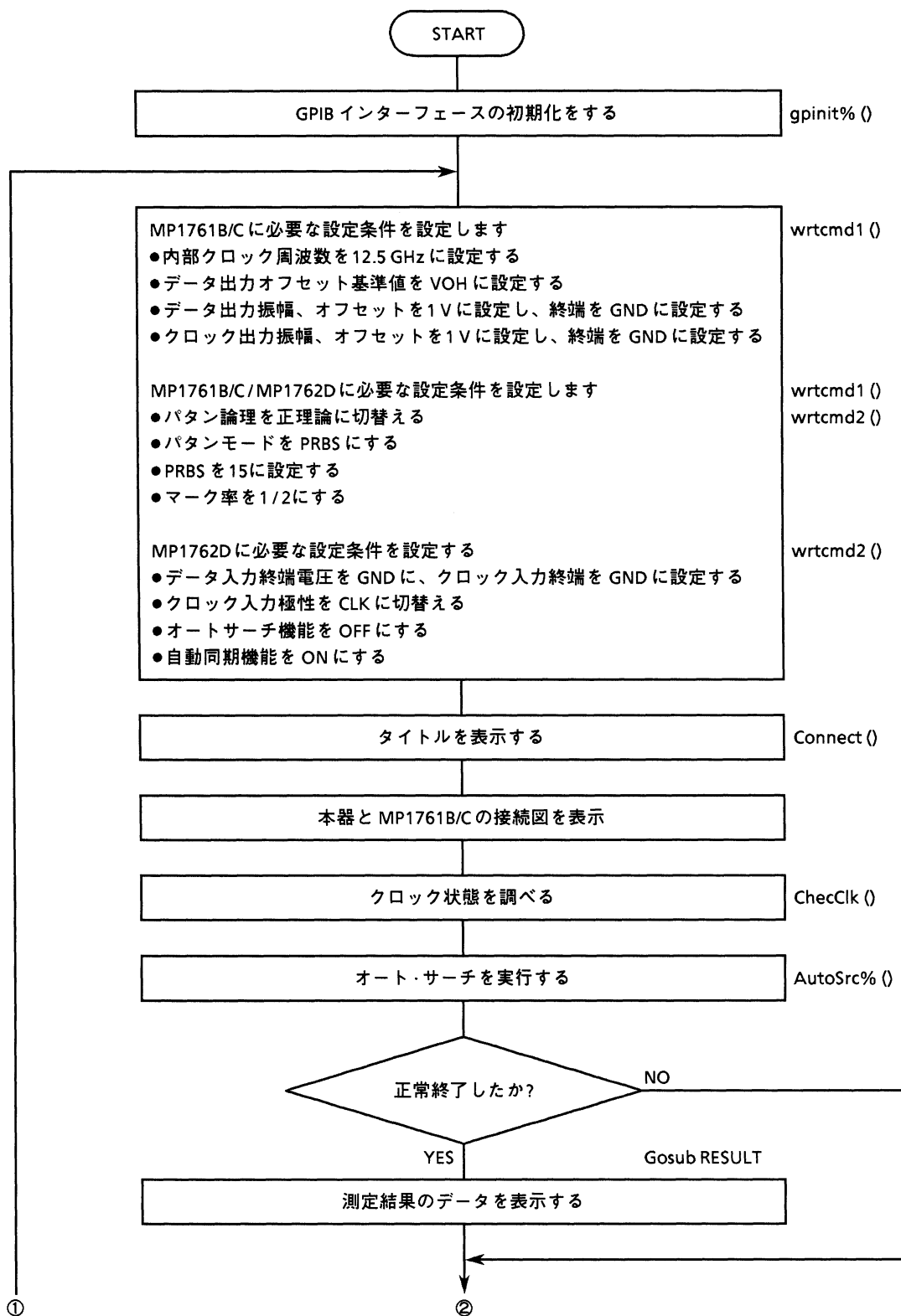
        CASE 4
            INPUT "Choose CLOCK POLARITY.[CLK:0, NCLK:1] "; cpl$
            CALL wrtcmd2("CPL " + cpl$)

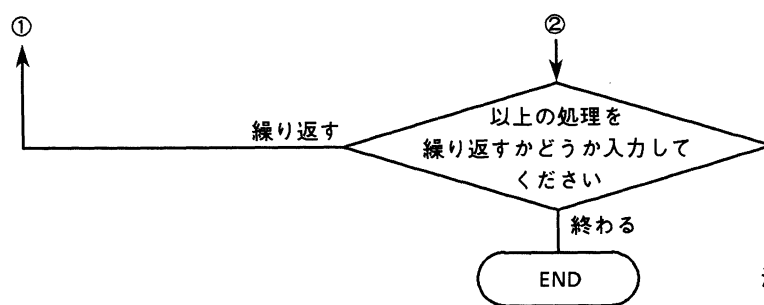
```

```
        ,  
        END SELECT  
    ,  
    INPUT "Do you set are another data? [Yes:0, No:1]:"; loop$  
    LOOP UNTIL loop$ = "1"  
END IF  
,  
STOP
```

(2) 自動スレッシュホールド・サーチ (オート・サーチ) の設定

本プログラムはオート・サーチを実行するプログラムです。





注) システムエラーがあった場合、エラーの内容を表示してプログラムを STOP します。

● プログラム・リスト

```

DECLARE SUB wrtcmd1 (WRT$)
REM $INCLUDE: 'c:\wat-gpib\qbasic\qbdecl.bas'

COMMON SHARED DEV%, GPIB0%, PPG%, ED%
'
DECLARE SUB ChecClk ()
DECLARE SUB Connect (ttl$)
DECLARE SUB wrtcmd2 (w$)
DECLARE FUNCTION gpinit% ()
DECLARE FUNCTION AutoSrc% ()
DECLARE FUNCTION readcmd2$ ()
'
CLS
IF gpinit% <> 0 THEN                                'Setup interface
DO
    ' Setup to PPG
    CALL wrtcmd1("CLK 1;RES 1;FRQ 12500") 'FREQUENCY
    CALL wrtcmd1("OFS 0") 'Offset
    CALL wrtcmd1("DAP 1;DOS 1;DTM 0") 'Data
    CALL wrtcmd1("CDL 100;CAP 1;COS 1") 'Clock
    ' Setup to ED
    CALL wrtcmd2("DTM 0;CTM 0;CPL 0") 'Input
    CALL wrtcmd2("SRH 0;SYN 1")
    ' Setup to PPG/ED
    CALL wrtcmd1("LGC 0") 'Pattern Logic :POSITIVE
    CALL wrtcmd2("LGC 0")
    CALL wrtcmd1("PTS 3") 'Pattern :PRBS
    CALL wrtcmd2("PTS 3")
    CALL wrtcmd1("PTN 6") 'PRBS :PN15
    CALL wrtcmd2("PTN 6")
    CALL wrtcmd1("MRK 3") 'Mark ratio :1/2
    CALL wrtcmd2("MRK 3")
    '
    CALL Connect("*** AUTO SEARCH SAMPLE PROGRAM ***")

    CALL ChecClk 'Check Clock loss

    IF AutoSrc% <> 0 THEN
        GOSUB RESULT 'Display Result
    END IF
    '
    PRINT
    INPUT "Try again [Yes/No]"; loop$
    LOOP UNTIL loop$ = "n" OR loop$ = "N"
END IF
STOP

'
RESULT: '----- Display Result -----
'
CALL wrtcmd2("DTH?")
RD$ = LEFT$(readcmd2$, IBCNT% - 1)
PRINT "DATA THRESHOLD = " + MID$(RD$, 5, 6) + " V"

CALL wrtcmd2("DTM?")
RD$ = LEFT$(readcmd2$, IBCNT% - 1)
IF MID$(RD$, 1, 5) = "DTM 0" THEN
    RD$ = " GND"
ELSE
    RD$ = " -2V"
END IF
PRINT "DATA TERMINATION = " + RD$

```

```

CALL wrtcmd2("CPA?")
RD$ = LEFT$(readcmd2$, IBCNT% - 1)
PRINT "CLOCK PHASE ADJUST    = " + MID$(RD$, 5, 4) + " ps"

CALL wrtcmd2("CTM?")
RD$ = LEFT$(readcmd2$, IBCNT% - 1)
IF MID$(RD$, 1, 5) = "CTM 0" THEN
    RD$ = "    GND"
ELSE
    RD$ = "    -2V"
END IF
PRINT "CLOCK TERMINATION    = " + RD$

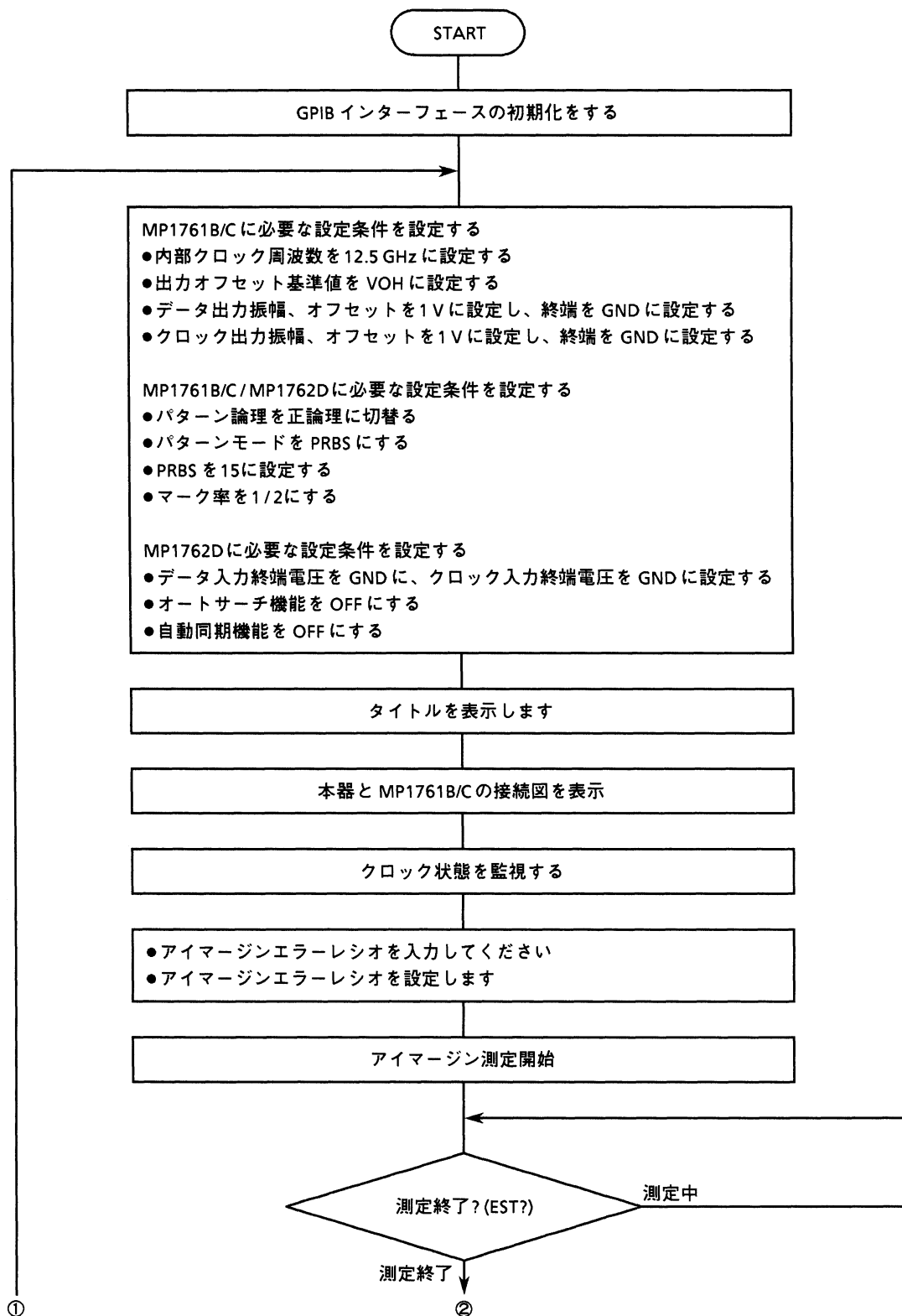
CALL wrtcmd2("CPL?")
RD$ = LEFT$(readcmd2$, IBCNT% - 1)
IF MID$(RD$, 1, 5) = "CPL 0" THEN
    RD$ = "    CLK"
ELSE
    RD$ = "    N CLK"
END IF
PRINT "CLOCK POLARITY        = " + RD$

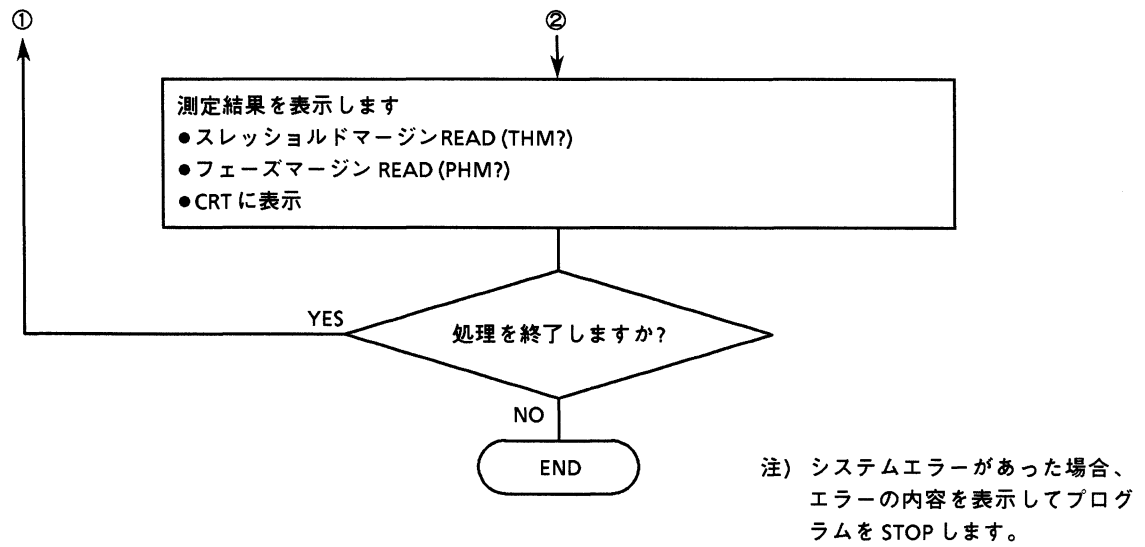
RETURN

```

(3) アイ・マージンの測定

本プログラムはアイ・マージンの測定を行います。





● プログラム・リスト

```

DECLARE SUB wrtcmd1 (WRT$)
REM $INCLUDE: 'c:\wat-gpib\qbasic\qbdecl.bas'

COMMON SHARED DEV%, GPIB0%, PPG%, ED%

DECLARE SUB ChecClk ()
DECLARE SUB ClearDisp (p%, l%)
DECLARE SUB waitly (tim!)
DECLARE SUB Connect (ttl$)
DECLARE SUB wrtcmd2 (w$)
DECLARE FUNCTION gpinit% ()
DECLARE FUNCTION readcmd2$ ()

IF gpinit% <> 0 THEN                                'Setup interface
DO
    '===== Eye margin start =====
    ' Setup to PPG
    CALL wrtcmd1("CLK 1;RES 1;FRQ 12500") 'FREQUENCY
    CALL wrtcmd1("OFS 0") 'Offset
    CALL wrtcmd1("DAP 1;DOS 1;DTM 0") 'Data
    CALL wrtcmd1("CDL 100;CAP 1;COS 1") 'Clock
    ' Setup to ED
    CALL wrtcmd2("DTM 0;CTM 0;CPL 0") 'Input
    CALL wrtcmd2("SRH 0;SYN 1")
    ' Setup to PPG/ED
    CALL wrtcmd1("LGC 0") 'Pattern Logic :POSITIVE
    CALL wrtcmd2("LGC 0")
    CALL wrtcmd1("PTS 3") 'Pattern :PRBS
    CALL wrtcmd2("PTS 3")
    CALL wrtcmd1("PTN 6") 'PRBS :PN15
    CALL wrtcmd2("PTN 6")
    CALL wrtcmd1("MRK 3") 'Mark ratio :1/2
    CALL wrtcmd2("MRK 3")
    CALL wrtcmd1("SFT 0") 'AND bit shift :1bit
    CALL wrtcmd2("SFT 0")

    CALL Connect("*** EYE MARGIN SAMPLE PROGRAM ***")

DO
    '==== Set error ratio ====
    LOCATE 20, 1
    PRINT "INPUT ERROR RATIO [ 1.0E-2:0 , 1.0E-3:1 , 1.0E-4:2 , 1.0E-5:3
]"
    PRINT " [ 1.0E-6:4 , 1.0E-7:5 , 1.0E-8:6 , 1.0E-9:7
]"
    PRINT
    INPUT "Choose error ratio: "; ratio%
    IF ratio% < 0 OR ratio% > 7 THEN
        LOCATE 18, 1
        PRINT "Wrong chosen number!! Please, enter a correct number."
        LOCATE 23, 1
        PRINT " "
    END IF
    LOOP UNTIL ratio% >= 0 AND ratio% <= 7
    CALL wrtcmd2("EME 1;EYT " + STR$(ratio%))

    CALL ChecClk '==== test clock loss ====

    CALL ClearDisp(18, 6)

    CALL wrtcmd2("EST 1") '==== Eye margin start ====
    LOCATE 4, 1
    PRINT "*** Eye Margin Start *** "

```

```

'===== Polling end of Measurement =====
DO
  CALL wrtcmd2("EST?")
  RD$ = LEFT$(readcmd2$, IBCNT% - 1)
  IF MID$(RD$, 1, 5) = "EST 0" THEN
    LOCATE 4, 1
    PRINT "*** Eye Margin finish ***  "
    waidly (1)

    '===== Display result =====
    CALL wrtcmd2("THM?")
    RD1$ = LEFT$(readcmd2$, IBCNT% - 1)
    CALL wrtcmd2("PHM?")
    RD2$ = LEFT$(readcmd2$, IBCNT% - 1)
    PRINT "THRESHOLD MARGIN : " + MID$(RD1$, 5, 6) + " Vp-p"
    PRINT "PHASE MARGIN      : " + MID$(RD2$, 5, 6) + " psp-p"
    EXIT DO
  ELSE
    IF MID$(RD$, 1, 5) = "EST 1" THEN
      waidly (1)
    ELSE
      'if execute fail then restart measure
      LOCATE 4, 1
      PRINT "*** Eye Margin Execution fail ***  "
      CALL wrtcmd2("EST 0")
      EXIT DO
    END IF
  END IF
END IF
LOOP

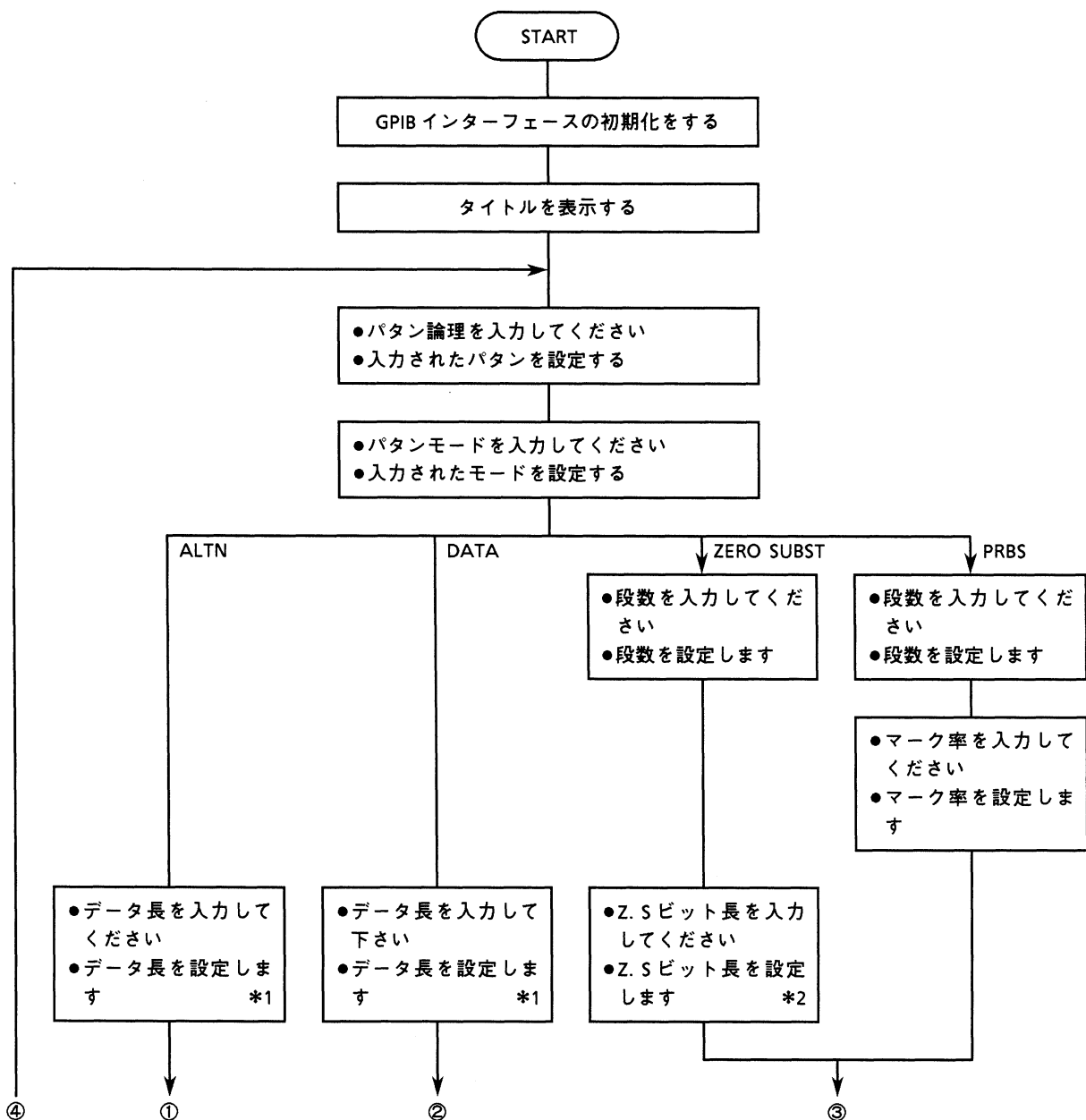
PRINT
INPUT "Do you wish try again [Yes/No]"; loop$
LOOP UNTIL loop$ = "n" OR loop$ = "N"

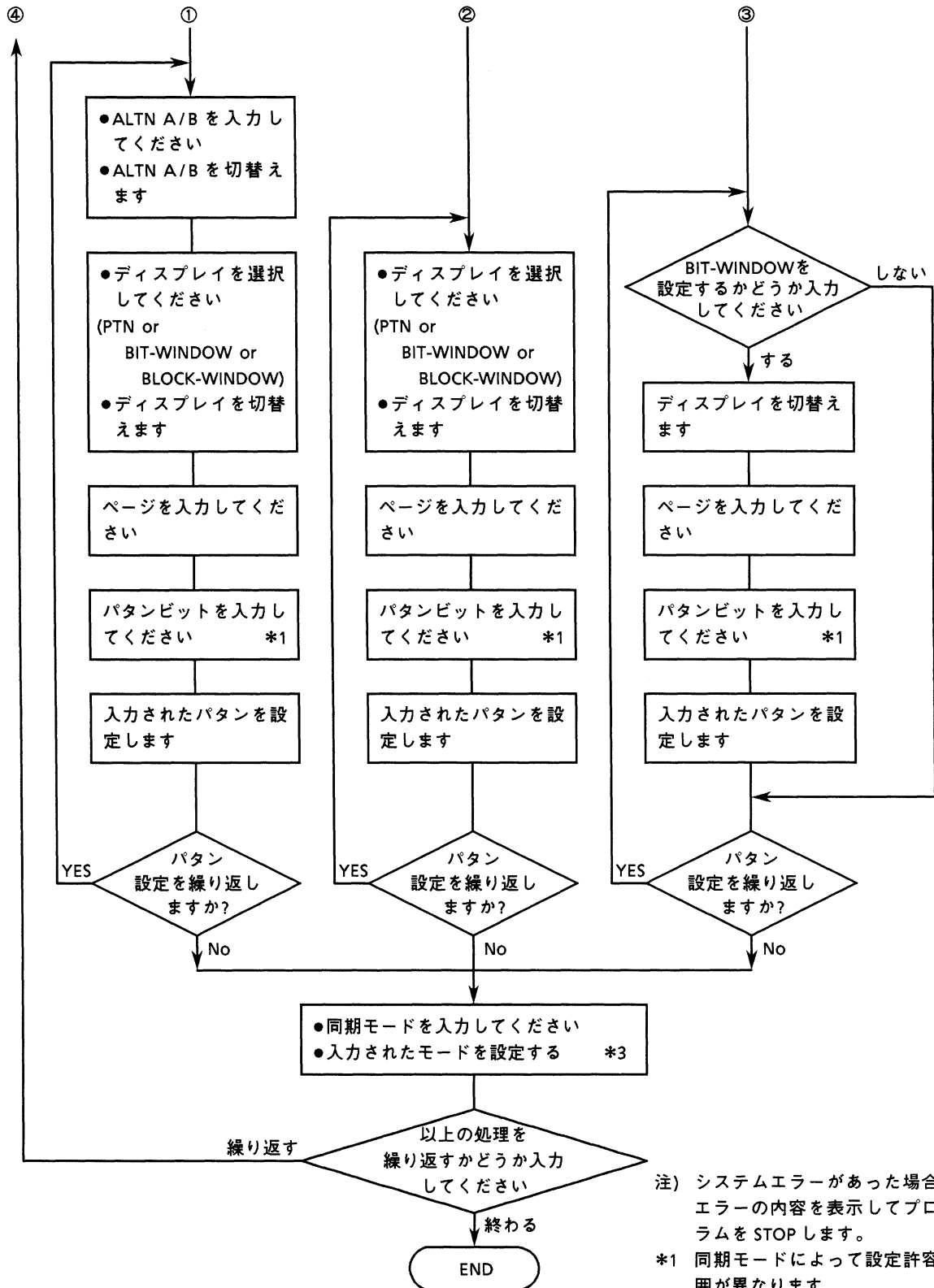
CALL wrtcmd2("EME 0")      ' ==== Eye margin OFF ====
END IF
'
STOP

```

(4) 測定パターン、BIT WINDOW、BLOCK WINDOW の設定

本プログラムは測定パターン、BIT WINDOW、BLOCK WINDOW の設定を行います。





注) システムエラーがあった場合、エラーの内容を表示してプログラムをSTOPします。

*1 同期モードによって設定許容範囲が異なります

*2 PRBS/ZERO SUBST(段数)の設定によって設定許容範囲は異なります

*3 パタンモードによって選択項目が違います

● プログラム・リスト

```

REM $INCLUDE: 'c:\wat-gpib\qbasic\qbdecl.bas'

COMMON SHARED DEV%, GPIB0%, PPG%, ED%

DECLARE SUB wrtcmd2 (w%)
DECLARE FUNCTION gpinit% ()

CLS
IF gpinit% <> 0 THEN                                'Setup GPIB interface
DO
CLS

PRINT "** MP1762D/MP1764D PATTERN SAMPLE PROGRAM **"
PRINT

'===== Set Pattern Logic =====
DO
INPUT "Choose LOGIC polarity [ Positive:0 , Negative:1 ] "; lg%
IF lg% <> 0 AND lg% <> 1 THEN
CLS
PRINT "Wrong Chosen number!! Please select a correct LOGIC polar
ity."
END IF
LOOP UNTIL lg% = 0 OR lg% = 1
CALL wrtcmd2("LGC " + STR$(lg%))

'===== Set Pattern mode =====
DO
INPUT "Choose Measure PATTERN [ ALTERNATE:0 , DATA:1 , ZERO Subst.:2
, PRBS:3 ] "; ptn%
IF ptn% < 0 OR ptn% > 3 THEN
CLS
PRINT "Wrong Chosen number!! Please select a correct new PATTERN
."
END IF
LOOP UNTIL ptn% >= 0 AND ptn% <= 3
CALL wrtcmd2("PTS " + STR$(ptn%))

SELECT CASE ptn%
CASE 0
'===== Set Altn pattern =====
GOSUB SetAltn

'===== Set Sync mode =====
DO
INPUT "Choose SYNC MODE [ NOMAL:0 , FRAME:1 ] "; sync%
IF sync% <> 0 AND sync% <> 1 THEN
CLS
PRINT "Wrong Chosen number!! Please select a correct pattern
sync."
END IF
LOOP UNTIL sync% = 0 OR sync% = 1
wrtcmd2 ("SYM " + STR$(sync%))

CASE 1
'===== Set data pattern =====
GOSUB SetData

'===== Set Sync mode =====
DO
INPUT "Choose SYNC MODE [ NOMAL:0 , FRAME:1 , QUICK:2 ] "; sync%
IF sync% < 0 OR sync% > 2 THEN
CLS

```

```

                                PRINT "Wrong Chosen number!! Please select a correct pattern
sync."
                                END IF

                                LOOP UNTIL sync% >= 0 AND sync% <= 2
                                wrtcmd2 ("SYM " + STR$(sync%))

                                CASE 2
                                '===== Set zero subst pattern =====
                                GOSUB SetZero

                                '===== Set Sync mode =====
                                DO
                                INPUT "Choose SYNC MODE [ NOMAL:0 , FRAME:1 , QUICK:2 ] "; sync%
                                IF sync% < 0 OR sync% > 2 THEN
                                CLS
                                PRINT "Wrong Chosen number!! Please select a correct pattern
sync."
                                END IF

                                LOOP UNTIL sync% >= 0 AND sync% <= 2
                                wrtcmd2 ("SYM " + STR$(sync%))

                                CASE 3
                                '===== Set prbs pattern =====
                                GOSUB SetPrbs
                                '
                                END SELECT

                                '===== continue ? =====
                                INPUT "Do you wish to set other pattern set? [Yes/No]"; loop$
                                LOOP UNTIL loop$ = "n" OR loop$ = "N"
                                END IF
                                '
                                STOP
                                '
                                '
                                SetAltn: '----- Set Altn Pattern -----
                                '
                                '===== Data length =====
                                DO
                                INPUT "Set DATA LENGTH [128 to 4194304] "; length
                                IF length < 128 OR length > 4194304 THEN
                                CLS
                                PRINT "Wrong input data length!! Please to set a correct number."
                                END IF
                                LOOP UNTIL length >= 128 AND length <= 4194304
                                wrtcmd2 ("DLN " + STR$(length))
                                '
                                '===== Set pattern =====
                                DO
                                INPUT "Choose ALTERNATE A or B:"; alt$
                                IF alt$ = "a" OR alt$ = "A" THEN
                                wrtcmd2 ("ALT 0")
                                ELSE
                                IF alt$ = "b" OR alt$ = "B" THEN
                                wrtcmd2 ("ALT 1")
                                END IF
                                END IF
                                DO
                                INPUT "Choose DISPLAY type[ PATTERN:0 , BIT-WINDOW:1 , BLOCK-WINDOW:

```

```

2 ] "; disp%
      IF disp% < 0 OR disp% > 2 THEN
        CLS
        PRINT "Wrong Chosen number!! Please select a correct correct DIS
PLAY type."
      END IF

      LOOP UNTIL disp% >= 0 AND disp% <= 2
      wrtcmd2 ("DSP " + STR$(disp%))

      GOSUB SetBit

      INPUT "Do you wish to continue to set another pattern? [Yes/No] "; loop$
      LOOP UNTIL loop$ = "n" OR loop$ = "N"
      RETURN
    ,
SetData: '----- Set DATA Pattern -----
    ,
    '===== Set data length =====
    DO
      INPUT "Set DATA LENGTH [2 to 8388608] "; length
      IF length < 2 OR length > 8388608 THEN
        CLS
        PRINT "Wrong length number!! Please set correct number."
      END IF
      LOOP UNTIL length >= 2 AND length <= 8388608
      wrtcmd2 ("DLN " + STR$(length))

    DO
      '===== Set display =====
      DO
        INPUT "Choose DISPLAY type[ PATTERN:0 , BIT-WINDOW:1 , BLOCK-WINDOW:
2 ] "; disp%
        IF disp% < 0 OR disp% > 2 THEN
          CLS
          PRINT "Wrong Chosen number!! Please select a correct correct DIS
PLAY type."
        END IF
        LOOP UNTIL disp% >= 0 AND disp% <= 2
        wrtcmd2 ("DSP " + STR$(disp%))

        '===== Set pattern =====
        GOSUB SetBit

        INPUT "Do you wish continue another pattern set? [Yes/No] "; loop$
        LOOP UNTIL loop$ = "n" OR loop$ = "N"
        RETURN
      ,
SetZero: '----- Set ZERO SUBST PATTERN -----
    ,
    '===== Set zero subst =====
    DO
      PRINT "ZERO SUBSTITUTION [2^7-1:0 , 2^9-1:1 , 2^11-1:2 , 2^15-1:3 ]"
      PRINT
      INPUT "Choose ZERO Substitution lenght:"; dan%
      IF dan% < 0 OR dan% > 3 THEN
        CLS
        PRINT "Wrong Chosen number!! Please put things right length."
      END IF
      LOOP UNTIL dan% >= 0 AND dan% <= 3

```

```

DO
'===== Set zero-sub length =====
SELECT CASE dan%
CASE 0
  wrtcmd2 ("PTN 2")
  DO
    INPUT "Set ZERO Substitution BIT LENGTH [1 to 127]"; length%
    IF length% < 1 OR length > 127 THEN
      CLS
      PRINT "Wrong input for out of range limit!! Please enter a c
correct bit length."
    END IF
    LOOP UNTIL length% >= 1 AND length% <= 127

CASE 1
  wrtcmd2 ("PTN 3")
  DO
    INPUT "Set ZERO Substitution BIT LENGTH [1 to 511]"; length%
    IF length% < 1 OR length > 511 THEN
      CLS
      PRINT "Wrong input for out of range limit!! Please enter a c
correct bit length."
    END IF
    LOOP UNTIL length% >= 1 AND length% <= 511

CASE 2
  wrtcmd2 ("PTN 5")
  DO
    INPUT "Set ZERO Substitution BIT LENGTH [1 to 2047]"; length%
    IF length% < 1 OR length > 2047 THEN
      CLS
      PRINT "Wrong input for out of range limit!! Please enter a c
correct bit length."
    END IF
    LOOP UNTIL length% >= 1 AND length% <= 2047

CASE 3
  wrtcmd2 ("PTN 6")
  DO
    INPUT "Set ZERO Substitution BIT LENGTH [1 to 32767]"; length%
    IF length% < 1 OR length > 32767 THEN
      CLS
      PRINT "Wrong input for out of range limit!! Please enter a c
correct bit length."
    END IF
    LOOP UNTIL length% >= 1 AND length% <= 32767

END SELECT
wrtcmd2 ("ZLN " + STR$(length%))

'===== Chanel mask =====
INPUT "Do you set Mask Chanel ? [Yes/No] "; ans$
IF ans$ = "y" OR ans$ = "Y" THEN
  wrtcmd2 ("DSP 1")
  disp% = 1
  GOSUB SetBit
ELSE
  wrtcmd2 ("DSP 0")
  disp% = 0
END IF

INPUT "Do you wish continue another pattern set? [Yes/No] "; loop$
LOOP UNTIL loop$ = "n" OR loop$ = "N"

```

```

RETURN
'
SetPrbs: '----- Set PRBS PATTERN -----
'===== Set prbs =====
DO
    PRINT
    PRINT "SELECT PRBS [ 2^7 :0 , 2^9 :1 , 2^11:2 , 2^15:3 ]"
    PRINT "          [ 2^20:4 , 2^23:5 , 2^31:6          ]"
    PRINT
    INPUT "Choose PRBS lenght:"; dan%
    IF dan% < 0 OR dan% > 6 THEN
        CLS
        PRINT "Wrong Chosen number!! Please put things right length."
    END IF
LOOP UNTIL dan% >= 0 AND dan% <= 6

SELECT CASE dan%
CASE 0
    wrtcmd2 ("PTN 2")          ' PRBS 2^7
CASE 1
    wrtcmd2 ("PTN 3")          ' PRBS 2^9
CASE 2
    wrtcmd2 ("PTN 5")          ' PRBS 2^11
CASE 3
    wrtcmd2 ("PTN 6")          ' PRBS 2^15
CASE 4
    wrtcmd2 ("PTN 7")          ' PRBS 2^20
CASE 5
    wrtcmd2 ("PTN 8")          ' PRBS 2^23
CASE 6
    wrtcmd2 ("PTN 9")          ' PRBS 2^31
END SELECT

DO
    '===== Set mark ratio =====
    PRINT
    PRINT "MARK RATIO (Positive)[ 0/8:0 , 1/8:1 , 1/4:2 , 1/2:3 ]"
    PRINT "          (Negative)[ 8/8:0 , 7/8:1 , 3/4:2 , 1/2:3 ]"
    PRINT
    INPUT "Choose MARK RATIO:"; m$
    wrtcmd2 ("MRK " + m$)

    '===== Chanel mask =====
    INPUT "Do you set Mask Chanel? [Yes/No] "; ans$
    IF ans$ = "y" OR ans$ = "Y" THEN
        wrtcmd2 ("DSP 1")
        disp% = 1
        GOSUB SetBit
    ELSE
        wrtcmd2 ("DSP 0")
        disp% = 0
    END IF

    INPUT "Do you wish continue another pattern set? [Yes/No] "; loop$
LOOP UNTIL loop$ = "n" OR loop$ = "N"
RETURN
'
'
SetBit: '----- Set Page and Bit Pattern -----
'

```

```

SELECT CASE disp%
CASE 0
  '===== Set page =====
  INPUT "to set top PAGE:"; page

  '===== Set pattern bit =====
  j = 8
  GOSUB Inbit
  PRINT "PAG " + STR$(page) + ";BIT " + BIT$
  wrtcmd2 ("PAG " + STR$(page) + ";BIT " + BIT$)

CASE 1
  '===== Set page =====
  INPUT "Set BIT-WINDOW PAGE ( 1 or 2 ):"; page

  '===== Set mask bit =====
  j = 2
  GOSUB Inbit
  wrtcmd2 ("MSK " + STR$(page) + ";CHM " + BIT$)

CASE 2
  '===== Set page =====
  INPUT "Set top PAGE:"; page

  '===== Set mask bit =====
  j = 8
  GOSUB Inbit
  wrtcmd2 ("PAG " + STR$(page) + ";MGB " + BIT$)
END SELECT
RETURN

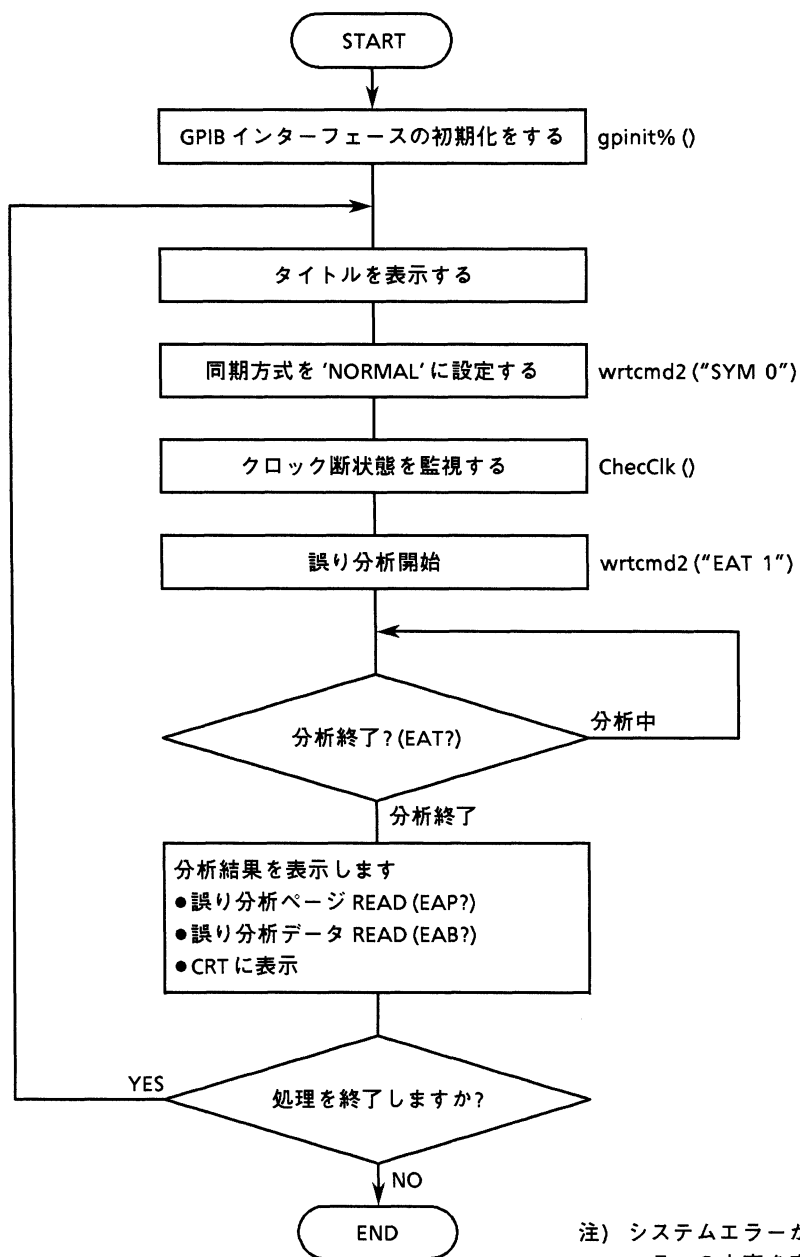
Inbit:
  PRINT "You are able to choice data format of Hexadecimal or Decimal."
  PRINT "Default data format is Hexadecimal."

  BIT$ = ""
  FOR k = 0 TO j - 1
    PRINT "< Do you set bit-pattern of " + STR$(page + k) + " PAGE ? [Yes/No
] > ";
    INPUT ; a$
    IF a$ = "n" OR a$ = "N" THEN
      EXIT FOR
    END IF
    IF k <> 0 THEN BIT$ = BIT$ + ","
    PRINT " "
    INPUT "Which do you choice format? [ Hex/Dec ]"; type$
    IF type$ = "d" OR type$ = "D" THEN
      PRINT "Enter " + STR$(page + k) + " PAGE pattern [0 to 65535]:";
      INPUT ; b$
    ELSE
      PRINT "Enter " + STR$(page + k) + " PAGE pattern [0 to FFFF]:";
      INPUT ; b$
      b$ = "#H" + b$
    END IF
    BIT$ = BIT$ + b$
  NEXT k
  PRINT " "
RETURN

```

(5) 誤り分析

本プログラムは誤り分析を行います。



注) システムエラーがあった場合、エラーの内容を表示してプログラムを STOP します。

● プログラム・リスト

```

REM $INCLUDE: 'c:\wat-gpib\qbasic\qbdecl.bas'

COMMON SHARED DEV%, GPIBO%, PPG%, ED%

DECLARE SUB ChecClk ()
DECLARE SUB waidly (tim!)
DECLARE SUB wrtcmd2 (w$)
DECLARE FUNCTION gpinit% ()
DECLARE FUNCTION readcmd2$ ()

IF gpinit% <> 0 THEN                                'Setup interface
DO
  CLS
  PRINT "** MP1762D/MP1764D ERROR ANALYSIS SAMPLE PROGRAM ** "
  PRINT
  CALL wrtcmd2("SYM 0")          ' Sync mode set to normal

  CALL ChecClk                   ' Test Clock loss

  '===== Error analysis start =====
  CALL wrtcmd2("EAT 1")

  '===== Polling end of analysis =====
DO
  CALL wrtcmd2("EAT?")
  RD$ = readcmd2$
  waidly (1)
  LOOP UNTIL MID$(RD$, 1, 5) = "EAT 2"
  CALL wrtcmd2("EAT 0")

  '===== Display result =====
  PRINT "Error Analysis data"
  FOR j = 1 TO 16
    CALL wrtcmd2("EAP " + STR$(j))
    CALL wrtcmd2("EAB?")
    RD$ = LEFT$(readcmd2$, IBCNT% - 1)

    PRINT "page:" + MID$(RD$, 5, 9) + ", Data:" + MID$(RD$, 17, 16)
  NEXT j
  PRINT

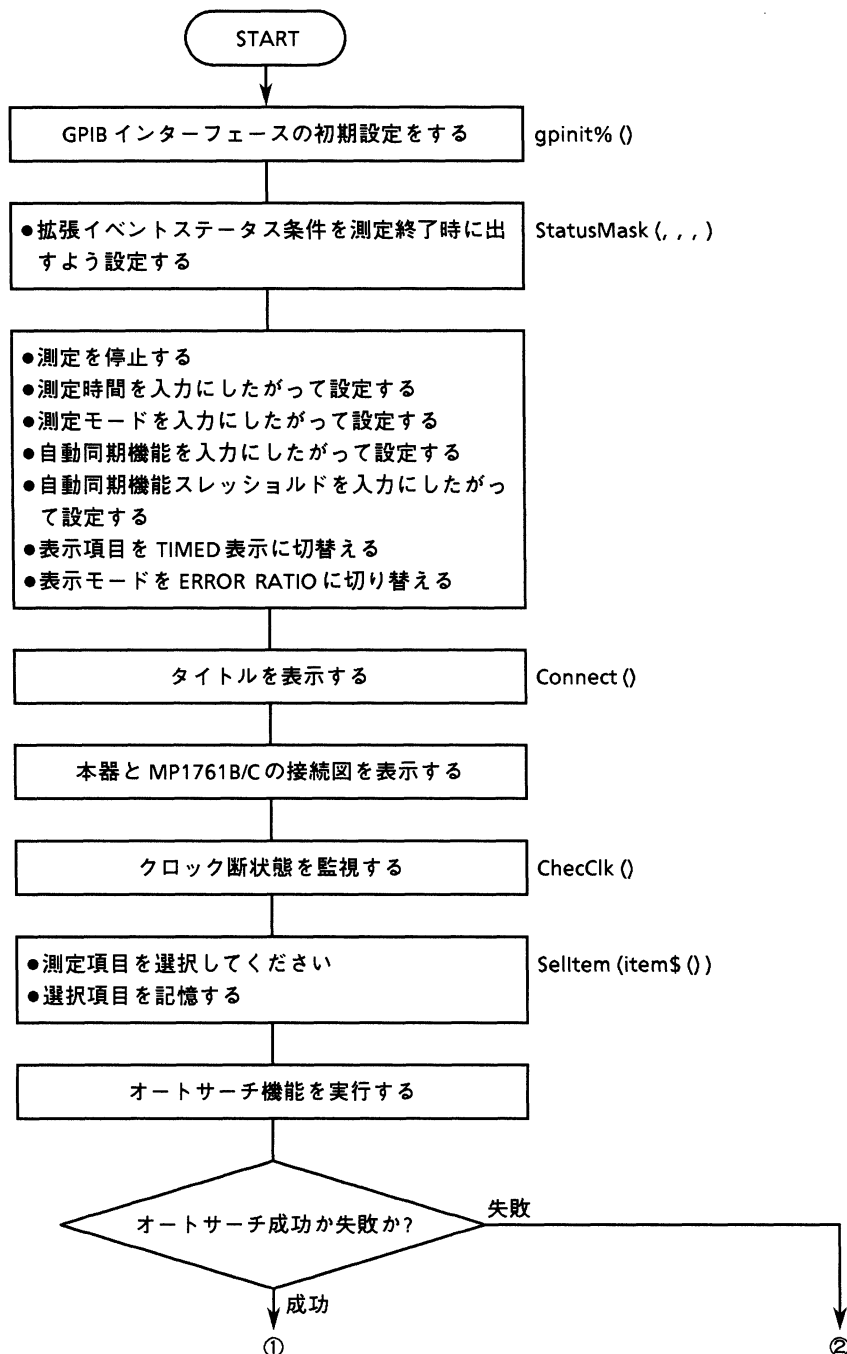
  INPUT "Do you want to try again? [Yes/No]:"; loop$
  LOOP UNTIL loop$ = "n" OR loop$ = "N"
END IF

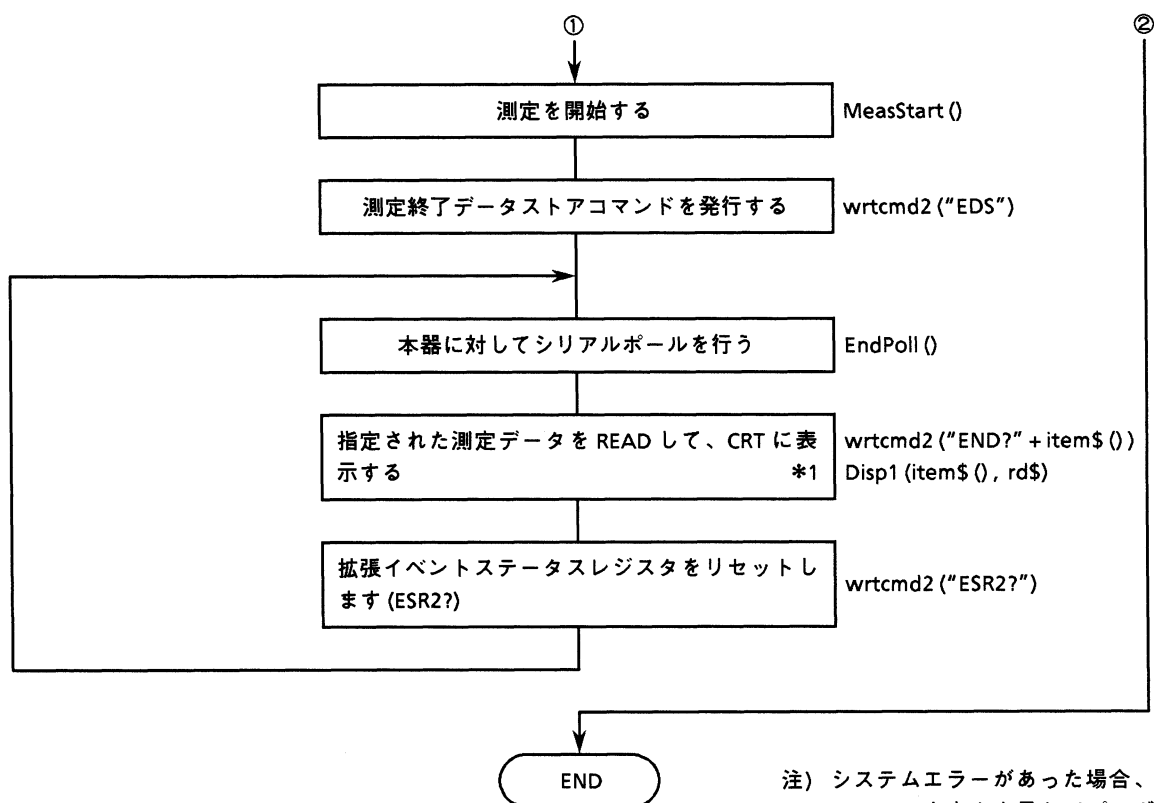
STOP

```

(6) 測定結果の表示その1(シリアルポールで測定結果を表示)

本プログラムは測定終了結果を CRT に表示します。





注) システムエラーがあった場合、エラーの内容を表示してプログラムを STOP します。

*1 指定した項目のみ、データの結果を読み出します。

● プログラム・リスト

```

REM $INCLUDE: 'c:\wat-gpib\qbasic\qbdecl.bas'

COMMON SHARED DEV%, GPIB0%, PPG%, ED%

DECLARE SUB waidly (tim!)
DECLARE SUB ClearDisp (p%, l%)
DECLARE SUB Disp1 (CMD$, RD$)
DECLARE SUB ChecC1k ()
DECLARE SUB MeasStart ()
DECLARE SUB MeasStop ()
DECLARE SUB EndPoll ()
DECLARE SUB SelItem (item$())
DECLARE SUB wrtcmd2 (w$)
DECLARE SUB Connect (ttl$)
DECLARE SUB StatusMask (s0%, s1%, s2%, s3%)
DECLARE FUNCTION gpinit% ()
DECLARE FUNCTION AutoSrc% ()
DECLARE FUNCTION readcmd2$ ()

DIM item$(5, 7)                                'Command string

CLS
IF gpinit% <> 0 THEN                             'Setup interface
'===== Set event status enable register =====
CALL StatusMask(&H0, &H0, &H1, &H0)

'===== Set mode =====
CALL MeasStop
DO
LOCATE 17, 1
INPUT "MEAS.MODE? [Repeat:0, Single:1, Untimed:2]", mmode%
IF mmode% < 0 OR mmode% > 2 THEN
LOCATE 16, 1
PRINT "Wrong chosen number!! Please select corrcet threshold."
CALL ClearDisp(17, 4)
END IF
LOOP UNTIL mmode% >= 0 AND mmode% <= 8
CALL wrtcmd2("MOD " + STR$(mmode%))

CALL ClearDisp(16, 2)
LOCATE 17, 1
IF mmode% <> 2 THEN
INPUT "MEAS.TIME? [DAY, HOUR, MINUTE, SECOND]", prd1%, prd2%, prd3%, prd4%
END IF
CALL wrtcmd2("PRD " + STR$(prd1%) + "," + STR$(prd2%) + "," + STR$(prd3%) +
", " + STR$(prd4%))

CALL ClearDisp(16, 2)
DO
LOCATE 17, 1
INPUT "AUTO SYNC CONDITION? [OFF:0, ON:1]", async%
IF async% < 0 OR async% > 1 THEN
LOCATE 16, 1
PRINT "Wrong chosen number!! Please select corrcet threshold."
CALL ClearDisp(17, 4)
END IF
LOOP UNTIL async% >= 0 AND async% <= 1
CALL wrtcmd2("SYN " + STR$(async%))

CALL ClearDisp(16, 2)
DO
LOCATE 17, 1
PRINT "AUTO SYNC Threshold Ratio"
PRINT "[ 1E-2:0 , 1E-3:1 , 1E-4:2 , 1E-5:3 , 1E-6:4 ] "
PRINT "[ 1E-7:5 , 1E-8:6 ,          , INT :8 ] "
INPUT "Choose Auto Sync Threshold: "; sye%

```

```

IF sye% < 0 OR sye% > 8 THEN
    LOCATE 16, 1
    PRINT "Wrong chosen number!! Please select corrcet threshold."
    CALL ClearDisp(17, 4)
END IF
LOOP UNTIL sye% >= 0 AND sye% <= 8
CALL ClearDisp(16, 5)
CALL wrtcmd2("SYE " + STR$(sye%))

CALL wrtcmd2("DMS 0")           'Error Display mode      : ERROR RATIO
CALL wrtcmd2("TIM 3")           'Real time display mode : TIMED

'===== Draw layout box =====
CALL Connect("** MEASUREMENT SAMPLE PROGRAM **")

'===== Check clock =====
CALL ChecClk

'===== Select display item =====
CALL SelItem(item$())           'item: command string (output)
                                'sort: command number (output)

'===== Auto Search ON =====
IF AutoSrc% = 1 THEN

    '===== Measurement start =====
    'CLS
    SCREEN 0
    CALL MeasStart
    CALL EndPoll                 'Reset SRQ
    LOCATE 24, 48
    PRINT "*** Report Measure execution. ***"
    LOCATE 24, 48
    PRINT "*** Push ESC key then stop.   ***"

    '===== Display result =====
    DO
        CALL wrtcmd2("MSR?")
        RD$ = LEFT$(readcmd2$, IBCNT% - 1)
        IF RD$ = "MSR 0" THEN
            CALL wrtcmd2("STA")
        END IF

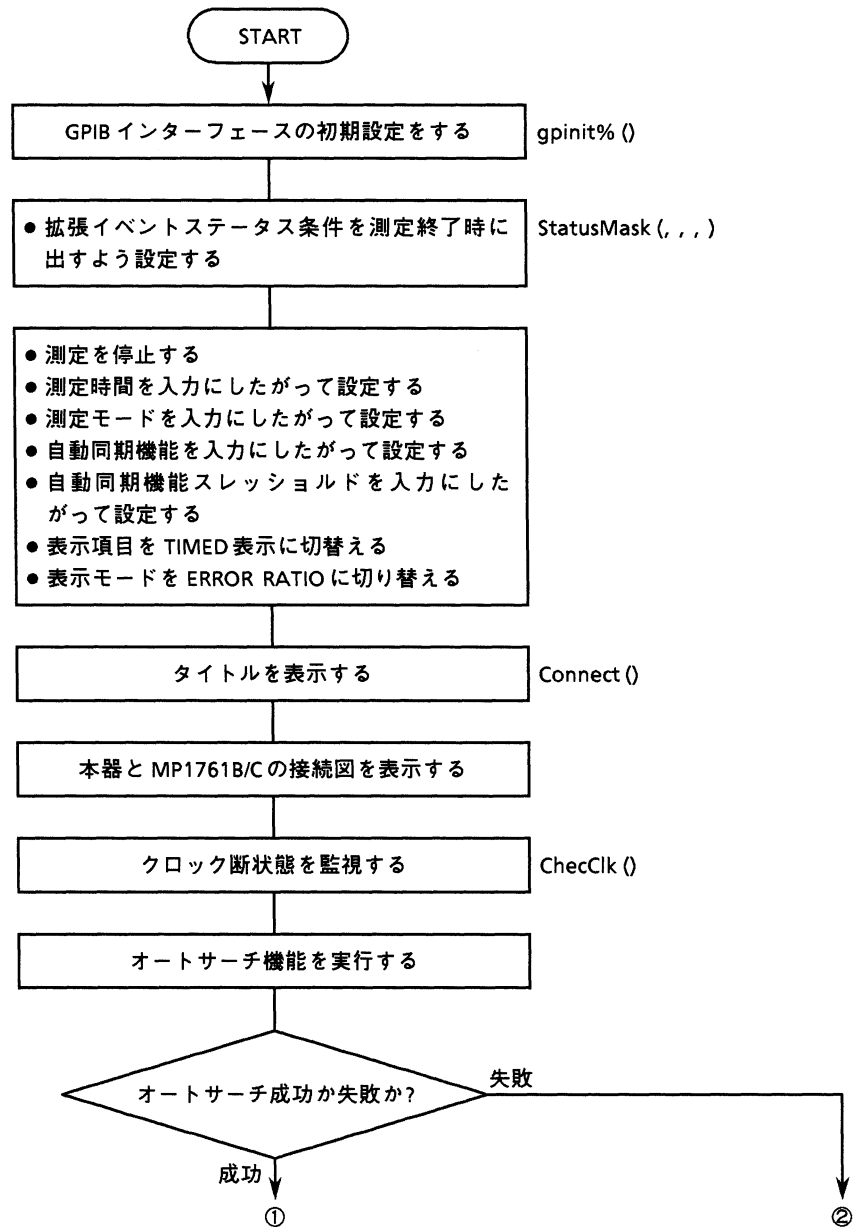
        CALL wrtcmd2("EDS")       'Data store
        CALL EndPoll             'Polling END bit

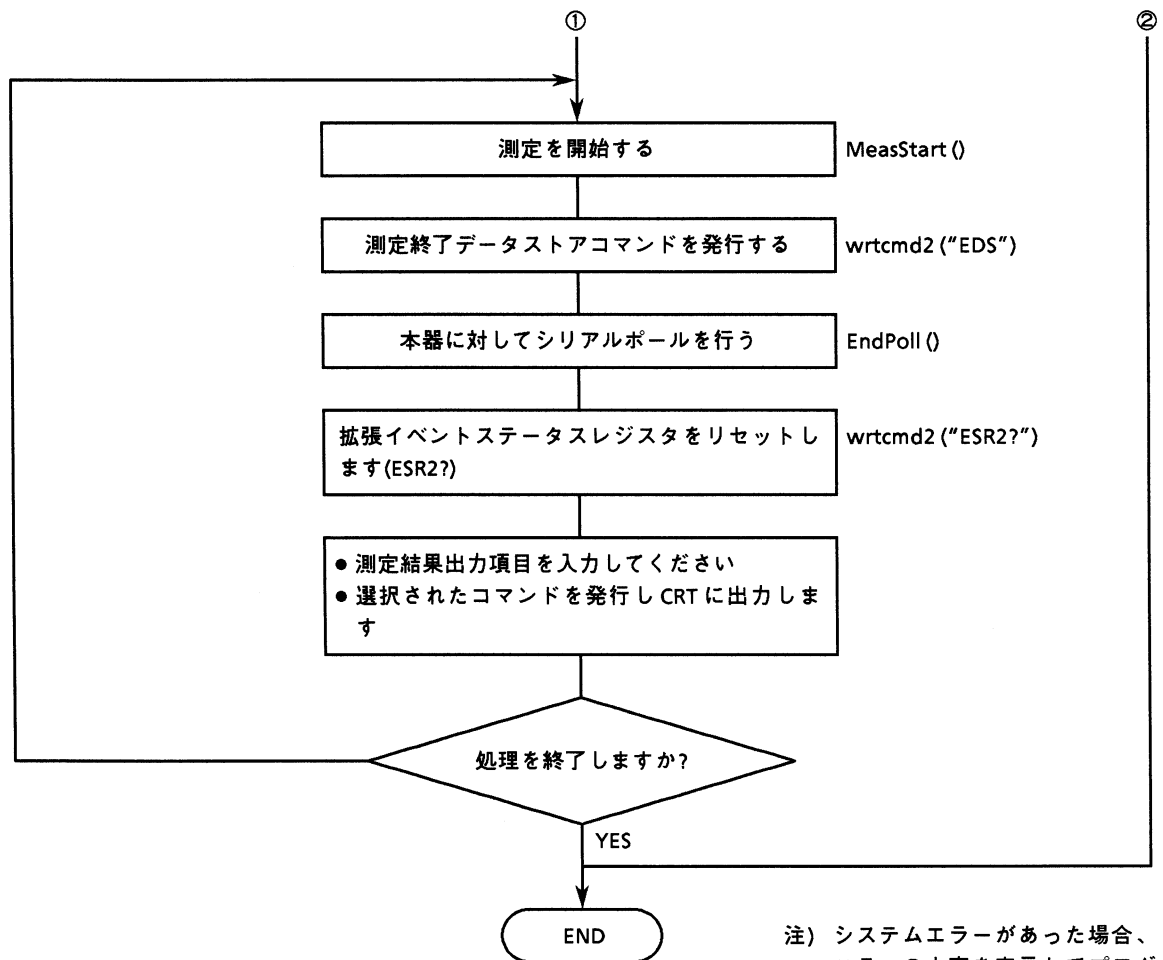
        LOCATE 1, 1
        FOR i = 0 TO 4
            FOR j = 0 TO 6
                IF item$(i, j) <> "" THEN
                    CALL wrtcmd2("END? " + item$(i, j)) 'Write command
                    RD$ = LEFT$(readcmd2$, IBCNT% - 1)
                    IF RD$ <> "ERR" THEN
                        CALL Displ(item$(i, j), RD$)      'Display data
                    END IF
                END IF
            NEXT j
        NEXT i
        CALL wrtcmd2("ESR2?")
        RD$ = readcmd2$
        LOOP UNTIL INKEY$ = CHR$(&H1B)                  'Reset ESR1
                                                        'ESC(&H1B)
    END IF
END IF
STOP

```

(7) 測定結果の表示その2(リクエスト・コマンドで測定結果を表示)

本プログラムは測定終了結果を CRT に表示します。





注) システムエラーがあった場合、エラーの内容を表示してプログラムを STOP します。

*1 指定した項目のみ、データの結果を読み出します。

● プログラム・リスト

```

REM $INCLUDE: 'c:\wat-gpib\qbasic\qbdecl.bas'

COMMON SHARED DEV%, GPIB0%, PPG%, ED%

DECLARE SUB StatusMask (s0%, s1%, s2%, s3%)
DECLARE SUB MeasStop ()
DECLARE SUB wrtcmd2 (w$)
DECLARE SUB EndPoll ()
DECLARE SUB ClearDisp (p%, l%)
DECLARE SUB Connect (ttl$)
DECLARE SUB ChecClk ()
DECLARE SUB MeasStart ()
DECLARE SUB waidly (tim!)
DECLARE FUNCTION gpinit% ()
DECLARE FUNCTION AutoSrc% ()
DECLARE FUNCTION readcmd2$ ()

CLS
IF gpinit% <> 0 THEN 'Setup interface
'===== Set event status enable register =====
CALL StatusMask(&H0, &H0, &H1, &H0)

'===== Set mode =====
CALL MeasStop
DO
LOCATE 17, 1
INPUT "MEAS.MODE? [Repeat:0, Single:1, Untime:2]", mmode%
IF mmode% < 0 OR mmode% > 2 THEN
LOCATE 16, 1
PRINT "Wrong chosen number!! Please select corrcct MEAS.MODE."
CALL ClearDisp(17, 4)
END IF
LOOP UNTIL mmode >= 0 AND mmode <= 2
CALL ClearDisp(17, 4)
CALL wrtcmd2("MOD " + STR$(mmode%))

IF mmode% <> 2 THEN
LOCATE 17, 1
INPUT "MEAS.TIME?[DAY,HOUR,MIN,SEC]", prd1%, prd2%, prd3%, prd4%
END IF
CALL ClearDisp(17, 4)
CALL wrtcmd2("PRD " + STR$(prd1%) + "," + STR$(prd2%) + "," + STR$(prd3%) +
", " + STR$(prd4%))

DO
LOCATE 17, 1
INPUT "AUTO SYNC CONDITION? [OFF:0, ON:1]", async%
IF async% < 0 OR async% > 1 THEN
LOCATE 16, 1
PRINT "Wrong chosen number!! Please select corrcct AUTO SYNC CONDITIO
N"
CALL ClearDisp(17, 4)
END IF
LOOP UNTIL async% >= 0 AND async% <= 1
CALL ClearDisp(17, 4)
CALL wrtcmd2("SYN " + STR$(async%))

DO
LOCATE 17, 1
PRINT "AUTO SYNC Threshold Ratio"
PRINT "[ 1E-2:0 , 1E-3:1 , 1E-4:2 , 1E-5:3 , 1E-6:4 ] "
PRINT "[ 1E-7:5 , 1E-8:6 , 1E-9:7 , INT :8 ] "
INPUT "Choose Auto Sync Threshold: "; sye%
IF sye% < 0 OR sye% > 8 THEN
LOCATE 16, 1
PRINT "Wrong chosen number!! Please select corrcct threshold "

```



```

        CALL ClearDisp(17, 4)
    END IF
    LOOP UNTIL sye% >= 0 AND sye% <= 8
    CALL ClearDisp(16, 5)
    CALL wrtcmd2("SYE " + STR$(sye%))

    CALL wrtcmd2("DMS 0")           'Error Display mode      : ERROR RATIO
    CALL wrtcmd2("TIM 3")           'Real time display mode : TIMED
    CALL wrtcmd2("CUR 1")           'Current data          : ON

    '===== Draw layout box =====
    CALL Connect("** MEASUREMENT SAMPLE PROGRAM **")

    '===== Check clock =====
    CALL ChecClk

    IF AutoSrc% = 1 THEN
        SCREEN 0
        DO
            '===== Measurement start =====
            CALL MeasStart

            '===== Polling END bit =====
            CALL EndPoll

            '===== Reset ESR2 =====
            CALL wrtcmd2("ESR2?")
            RD$ = LEFT$(readcmd2$, IBCNT% - 1)

            PRINT "          << Print measure data >> "
            PRINT "          Push any key then END !! "

            '===== Serect request =====
            DO

                CALL wrtcmd2("ER?")
                RD$ = LEFT$(readcmd2$, IBCNT% - 1)
                LOCATE 10, 10
                PRINT "Error Ratio  -> "; MID$(RD$, 5)

                CALL wrtcmd2("EC?")
                RD$ = LEFT$(readcmd2$, IBCNT% - 1)
                LOCATE 11, 10
                PRINT "Error Count  -> "; MID$(RD$, 5)

                CALL wrtcmd2("EI?")
                RD$ = LEFT$(readcmd2$, IBCNT% - 1)
                LOCATE 12, 10
                PRINT "EI          -> "; MID$(RD$, 5)

                CALL wrtcmd2("EFI?")
                RD$ = LEFT$(readcmd2$, IBCNT% - 1)
                LOCATE 13, 10
                PRINT "%EFI          -> " + MID$(RD$, 5) + " "

                CALL wrtcmd2("FRQ?")
                RD$ = LEFT$(readcmd2$, IBCNT% - 1)
                LOCATE 14, 10
                PRINT "Clock Cycle  -> "; MID$(RD$, 5)

                CALL wrtcmd2("MSR?")
                RD$ = LEFT$(readcmd2$, IBCNT% - 1)

```

```
        IF RD$ = "MSR 0" THEN
            CALL wrtcmd2("STA")
        END IF

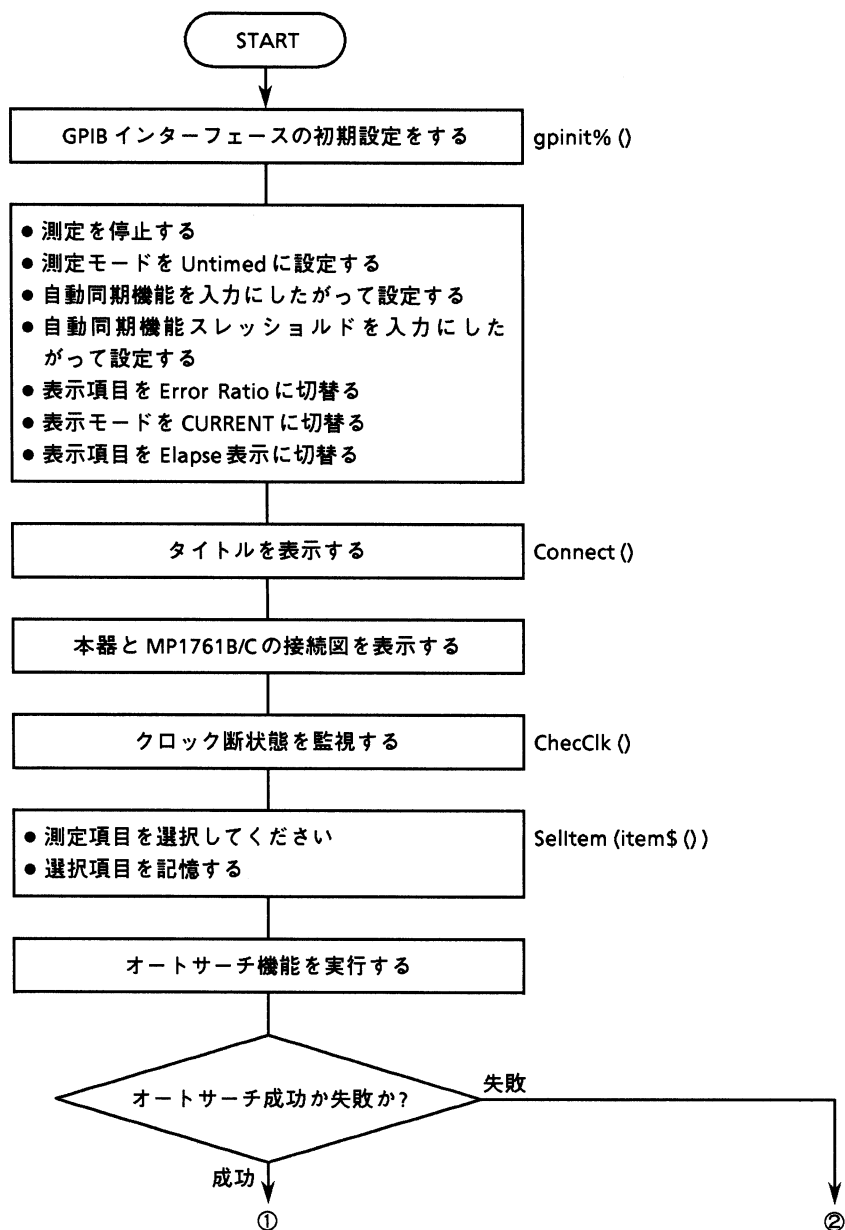
        CALL waidly(1)

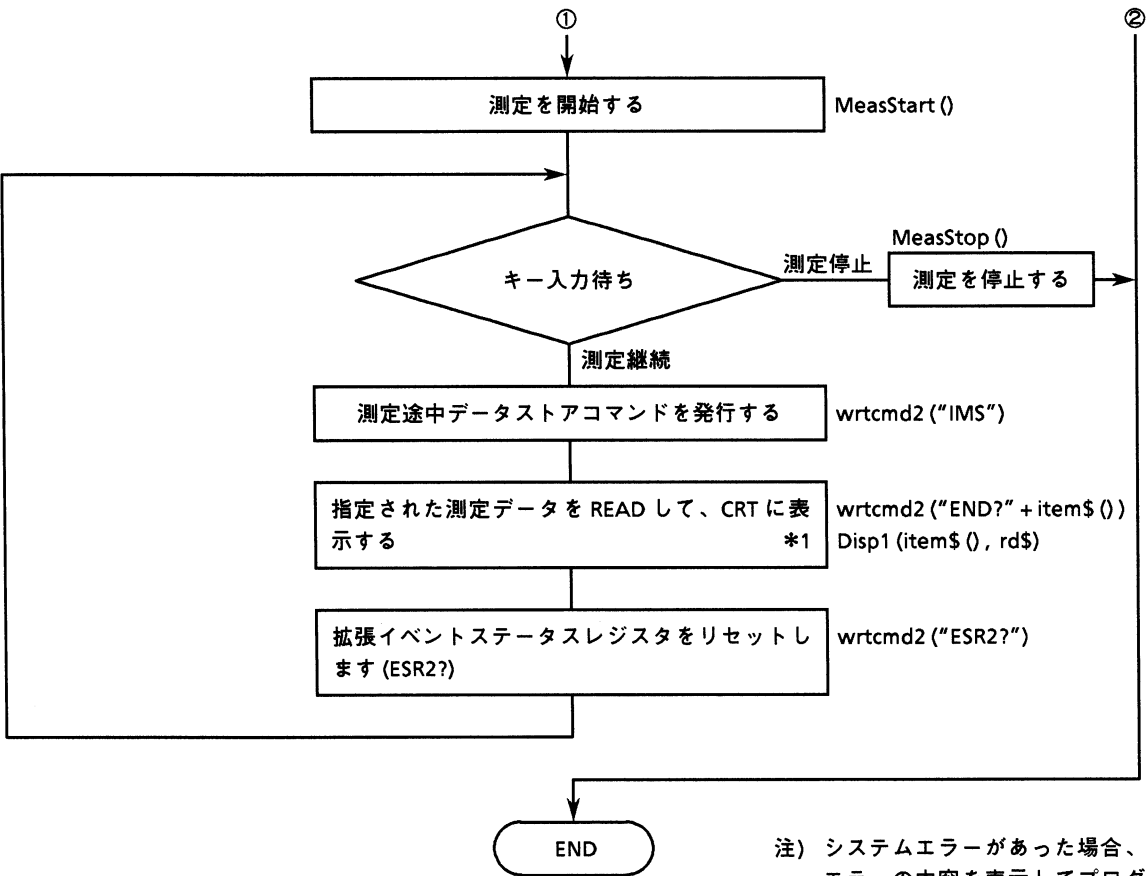
        LOOP UNTIL INKEY$ <> ""

        INPUT "Do you want measure try again? [Yes/No]:"; loop$
        LOOP UNTIL loop$ = "n" OR loop$ = "N"
    END IF
END IF
STOP
```

(8) 測定中間データの表示

本プログラムは測定中間結果を CRT に表示します。





注) システムエラーがあった場合、エラーの内容を表示してプログラムを STOP します。

*1 指定した項目のみ、データの結果を読み出します。

● プログラム・リスト

```

REM $INCLUDE: 'c:\wat-gpib\qbasic\qbdecl.bas'

COMMON SHARED DEV%, GPIB0%, PPG%, ED%

DECLARE SUB waitly (tim!)
DECLARE SUB ClearDisp (p%, l%)
DECLARE SUB Displ (CMD$, RDS)
DECLARE SUB ChecClk ()
DECLARE SUB MeasStart ()
DECLARE SUB MeasStop ()
DECLARE SUB EndPoll ()
DECLARE SUB SelItem (item$())
DECLARE SUB wrtcmd2 (w$)
DECLARE SUB Connect (ttl$)
DECLARE SUB StatusMask (s0%, s1%, s2%, s3%)
DECLARE FUNCTION gpinit% ()
DECLARE FUNCTION AutoSrc% ()
DECLARE FUNCTION readcmd2$ ()

DIM item$(5, 7)                                'Command string

CLS
IF gpinit% <> 0 THEN                                'Setup interface
'===== Set event status enable register =====
CALL StatusMask(&H0, &H0, &H1, &H0)

'===== Set mode =====
CALL MeasStop
CALL wrtcmd2("MOD 2")                            'Meas mode                :Untimed

CALL ClearDisp(16, 2)
DO
LOCATE 17, 1
INPUT "AUTO SYNC CONDITION? [OFF:0, ON:1]", async%
IF async% < 0 OR async% > 1 THEN
LOCATE 16, 1
PRINT "Wrong chosen number!! Please select corrcr threshold."
CALL ClearDisp(17, 4)
END IF
LOOP UNTIL async% >= 0 AND async% <= 1
CALL wrtcmd2("SYN " + STR$(async%))

CALL ClearDisp(16, 2)
DO
IF async% = 1 THEN
LOCATE 17, 1
PRINT "AUTO SYNC Threshold Ratio"
PRINT "[ 1E-2:0 , 1E-3:1 , 1E-4:2 , 1E-5:3 , 1E-6:4 ] "
PRINT "[ 1E-7:5 , 1E-8:6 , , INT :8 ] "
INPUT "Choose Auto Sync Threshold:"; sye%
IF sye% < 0 OR sye% > 8 THEN
LOCATE 16, 1
PRINT "Wrong chosen number!! Please select corrcr threshold."
CALL ClearDisp(17, 4)
END IF
END IF
LOOP UNTIL sye% >= 0 AND sye% <= 8
CALL ClearDisp(16, 5)
CALL wrtcmd2("SYE " + STR$(sye%))

CALL wrtcmd2("DMS 0")                            'Error Display mode                : ERROR RATIO
CALL wrtcmd2("CUR 1")                            'Current data                      : ON
CALL wrtcmd2("TIM 4")                            'Real time display mode           : ELAPSED

'===== Draw layout box =====

```

```

CALL Connect("*** MEASUREMENT SAMPLE PROGRAM ***")

'===== Check clock =====
CALL ChecClk

'===== Select display item =====
CALL SelItem(item$())          'item: command string (output)
                                'sort: command number (output)

'===== Auto Search ON =====
IF AutoSrc% = 1 THEN

    '===== Measurement start =====
    CLS
    SCREEN 0
    CALL MeasStart

    DO
        '===== wait trigger =====

        LOCATE 23, 48
        PRINT "*** Report Measure execution. ***"
        LOCATE 23, 48
        PRINT "*** Push ESC key then stop. ***"

        CALL wrtcmd2("IMS")          'Data store
        CALL waidly(1)

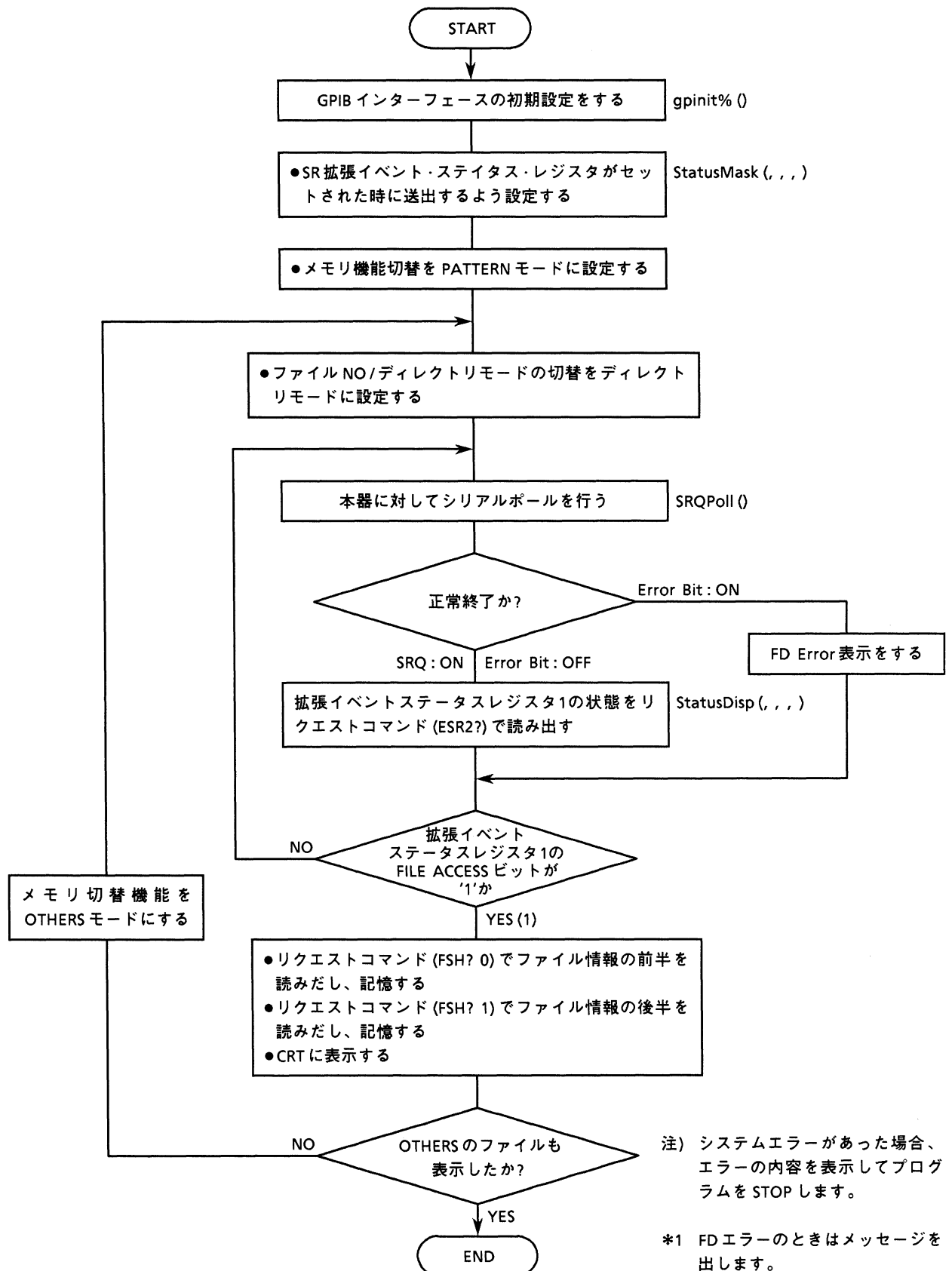
        '===== Display result =====

        LOCATE 1, 1
        FOR i = 0 TO 4
            FOR j = 0 TO 6
                IF item$(i, j) <> "" THEN
                    CALL wrtcmd2("IMD? " + item$(i, j)) 'Write command
                    RD$ = LEFT$(readcmd2$, IBCNT% - 1)
                    CALL Displ(item$(i, j), RD$)         'Display data
                END IF
            NEXT j
        NEXT i
        CALL wrtcmd2("ESR2?")
        RD$ = readcmd2$
        LOOP UNTIL INKEY$ = CHR$(&H1B)                  'Reset ESR1
                                                         'ESC(&H1B)
    END IF
END IF
STOP

```

(9) フロッピーディスクのファイル情報読みだし

本プログラムはフロッピーディスクに格納されているファイルのディレクトリ情報を CRT に表示します。



● プログラム・リスト

```

REM $INCLUDE: 'c:\wat-gpib\qbasic\qbdecl.bas'

COMMON SHARED DEV%, GPIB0%, PPG%, ED%

DECLARE SUB wrtcmd2 (w$)
DECLARE SUB ErrPoll ()
DECLARE SUB StatusDisp (stb%, esr%, esr2%, esr3%)
DECLARE SUB StatusMask (s0%, s1%, s2%, s3%)
DECLARE FUNCTION itob$ (i%, v%)
DECLARE FUNCTION SRQPoll% ()
DECLARE FUNCTION gpinit% ()
DECLARE FUNCTION readcmd2$ ()

IF gpinit% <> 0 THEN                                'Setup interface
    '===== Set MSS status byte register =====
    CALL StatusMask(&H4, &H0, &H2, &H2)

    FOR i = 0 TO 1
        '===== Set memory mode Pattern/Others =====
        wrtcmd2 ("MEM " + STR$(i))

        '===== Set FILE DIR mode =====
        wrtcmd2 ("FIL 1")

        '===== Polling FILE ACCESS bit =====
        DO
            IF SRQPoll% <> 0 THEN
                CALL StatusDisp(dmy%, dmy1%, reg%, dmy3%)
            ELSE
                LOCATE 12, 35
                PRINT "FD error detect!!"
                EXIT DO
            END IF
        LOOP UNTIL reg% AND &H2

        '===== Read FD infomation =====
        wrtcmd2 ("FDE?")
        rd1$ = LEFT$(readcmd2$, IBCNT% - 1)
        IF rd1$ <> "FDE 10" THEN
            LOCATE 1, 1
            SELECT CASE VAL(MID$(rd1$, 5, 2))
                CASE 0
                    PRINT "<<E0:Media error                >>"
                CASE 1
                    PRINT "<<E1:Write protection error      >>"
                CASE 2
                    PRINT "<<E2:File full                    >>"
                CASE 3
                    PRINT "<<E3:File not found                >>"
                CASE 4
                    PRINT "<<E4:File already exists error   >>"
                CASE 5
                    PRINT "<<E5:Write error                  >>"
                CASE 6
                    PRINT "<<E6:Read error                    >>"
                CASE 7
                    PRINT "<<E7:File type , File error      >>"
                CASE 8
                    PRINT "<<E8:FD error                      >>"
                CASE 9
                    PRINT "<<E9:Hardware error                >>"
            END SELECT
            LOCATE 23, 1
            INPUT "End of FD analyze. Press 'Enter' to fin."; f$
        END IF
    NEXT i
END IF

```



```

        STOP
    END IF
    wrtcmd2 ("FSH? 0")
    rd1$ = LEFT$(readcmd2$, IBCNT% - 1)
    wrtcmd2 ("FSH? 1")
    rd2$ = LEFT$(readcmd2$, IBCNT% - 1)

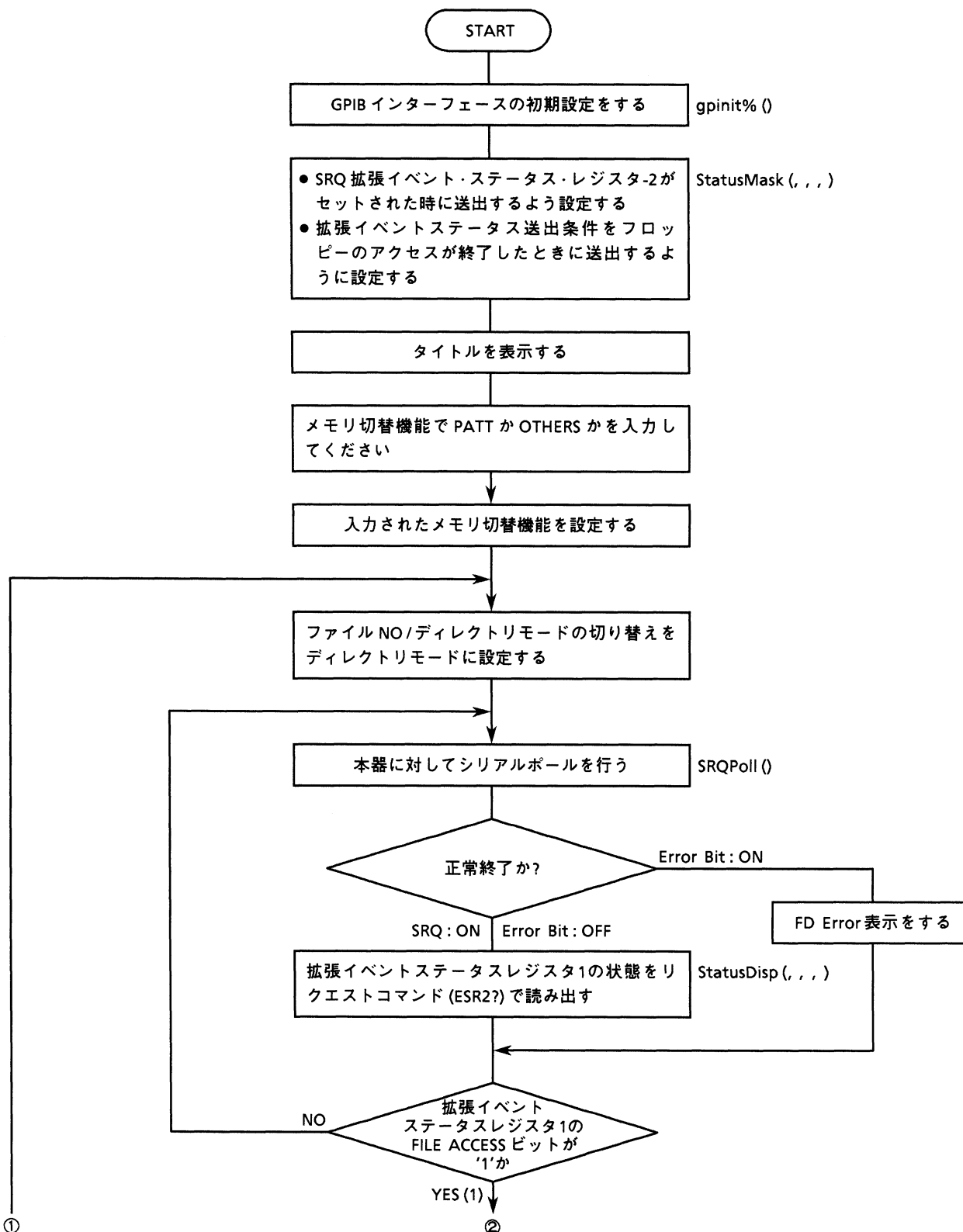
    '===== Output CRT =====
    IF i = 0 THEN
        LOCATE 4, 1
        PRINT "Pattern directory data."
    ELSE
        LOCATE 11, 1
        PRINT "Others directory data."
    END IF
    PRINT "Unused size : " + MID$(rd1$, 5, 7)           'print unused size
    PRINT "Used size   : " + MID$(rd1$, 13, 7)          'print used size
    PRINT "File count  : "; VAL(MID$(rd1$, 21, 22))      'print file num
                                                    'print file no
    PRINT "File name   : "; MID$(rd1$, 24) + "," + MID$(rd2$, 24)
NEXT i
PRINT
INPUT "End of FD analyze. Press 'Enter' to fin."; f$
END IF

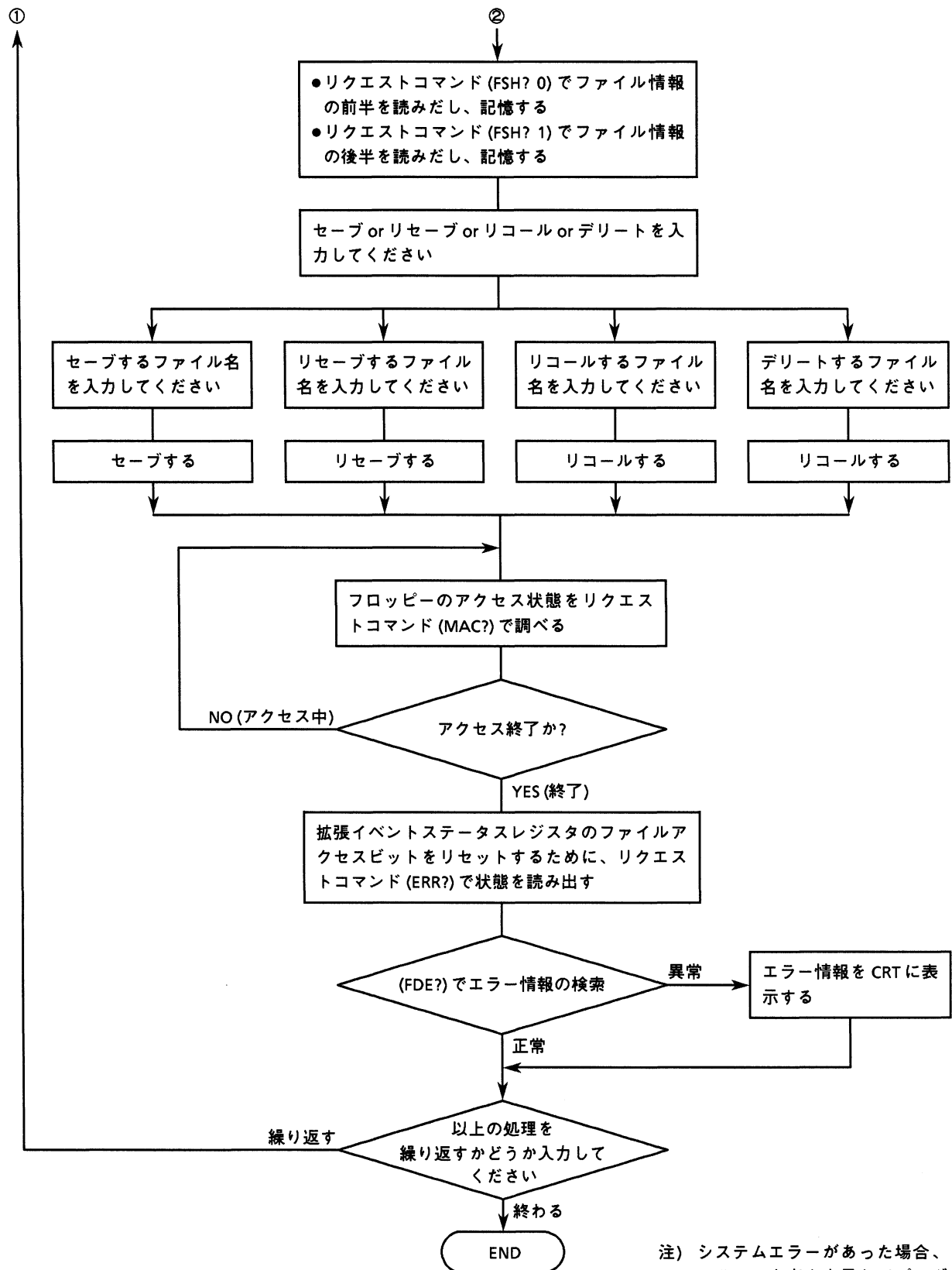
STOP

```

(10) FD のオペレーション (データのセーブ、リセーブ、リコール)

本プログラムはフロッピーディスクへのデータセーブ、リセーブ、またフロッピーディスクのデータリコールを行います。





注) システムエラーがあった場合、
エラーの内容を表示してプログラ
ムを STOP します。

*1 FDエラーの場合はメッセージを
出します。

● プログラム・リスト

```

DECLARE SUB ClearDisp (p%, l%)
REM $INCLUDE: 'c:\wat-gpib\qbasic\qbdecl.bas'

COMMON SHARED DEV%, GPIB0%, PPG%, ED%

DECLARE SUB waibly (tim!)
DECLARE SUB wrtcmd2 (w$)
DECLARE SUB ErrPoll ()
DECLARE SUB StatusDisp (stb%, esr%, esr2%, esr3%)
DECLARE SUB StatusMask (s0%, s1%, s2%, s3%)
DECLARE FUNCTION itob$ (l%, v%)
DECLARE FUNCTION SRQPoll% ()
DECLARE FUNCTION gpinit% ()
DECLARE FUNCTION readcmd2$ ()

IF gpinit% <> 0 THEN                                'Setup interface

'===== Set MSS status byte register =====
CALL StatusMask(&HC, &H0, &H2, &H2)

DO
CLS
PRINT "** MP1762D/MP1764D FD OPERATION PROGRAM ** "
'
'===== Select PTN/OTHERS =====
DO
LOCATE 17, 1
INPUT "Memory mode select [ PATTERN:0 , OTHERS:1 ]"; mem$
IF mem$ <> "0" AND mem$ <> "1" THEN
LOCATE 16, 1
PRINT "Wrong chosen number!! Please select a correct number"
END IF
CALL ClearDisp(16, 2)
LOOP UNTIL mem$ = "0" OR mem$ = "1"
wrtcmd2 ("MEM " + mem$)

'===== Set FILE DIR mode =====
wrtcmd2 ("FIL 1")

'===== Polling FILE ACCESS bit =====
DO
IF SRQPoll% <> 0 THEN
CALL StatusDisp(dmy%, dmy1%, reg%, dmy3%)
ELSE
GOSUB Fderr
GOTO jump
END IF
LOOP UNTIL reg% AND &H2

'==== Save, Resave, Recall or Delete? ====
DO
LOCATE 17, 1
INPUT "Choose function [ SAVE:0 , RESAVE:1 , RECALL:2 , DELEAT:3 ]";
op%
IF op% < 0 OR op% > 3 THEN
LOCATE 16, 1
PRINT "Wrong chosen number!! Please select correct function."
END IF
LOOP UNTIL op% >= 0 AND op% <= 3
CALL ClearDisp(16, 2)

LOCATE 17, 1
SELECT CASE op%

```

```

CASE 0
    INPUT "Enter file number for SAVE:"; NO$
    wrtcmd2 ("SAV " + NO$)
CASE 1
    INPUT "Enter file number for RESAVE:"; NO$
    wrtcmd2 ("RSV " + NO$)
CASE 2
    INPUT "Enter file number for RECALL:"; NO$
    wrtcmd2 ("RCL " + NO$)
CASE 3
    INPUT "Enter file number for DELEAT:"; NO$
    wrtcmd2 ("DEL " + NO$)

END SELECT
GOSUB Faccess
GOSUB Fderr
CALL ClearDisp(17, 1)

'===== Reset EventStatusRegister1 =====
jump: CALL StatusDisp(dmy%, dmy1%, dmy2%, dmy3%)
LOCATE 17, 1
INPUT "Do you more test another function? [Yes/No]"; loop$
LOOP UNTIL loop$ = "n" OR loop$ = "N"
END IF
STOP

Faccess: '===== FD access end ? =====
DO
    CALL StatusDisp(dmy%, dmy1%, dmy2%, dmy3%)
    waidly (1)
    wrtcmd2 ("MAC?")
    RD$ = LEFT$(readcmd2$, IBCNT% - 1)
    LOOP UNTIL MID$(RD$, 1, 5) = "MAC 0"

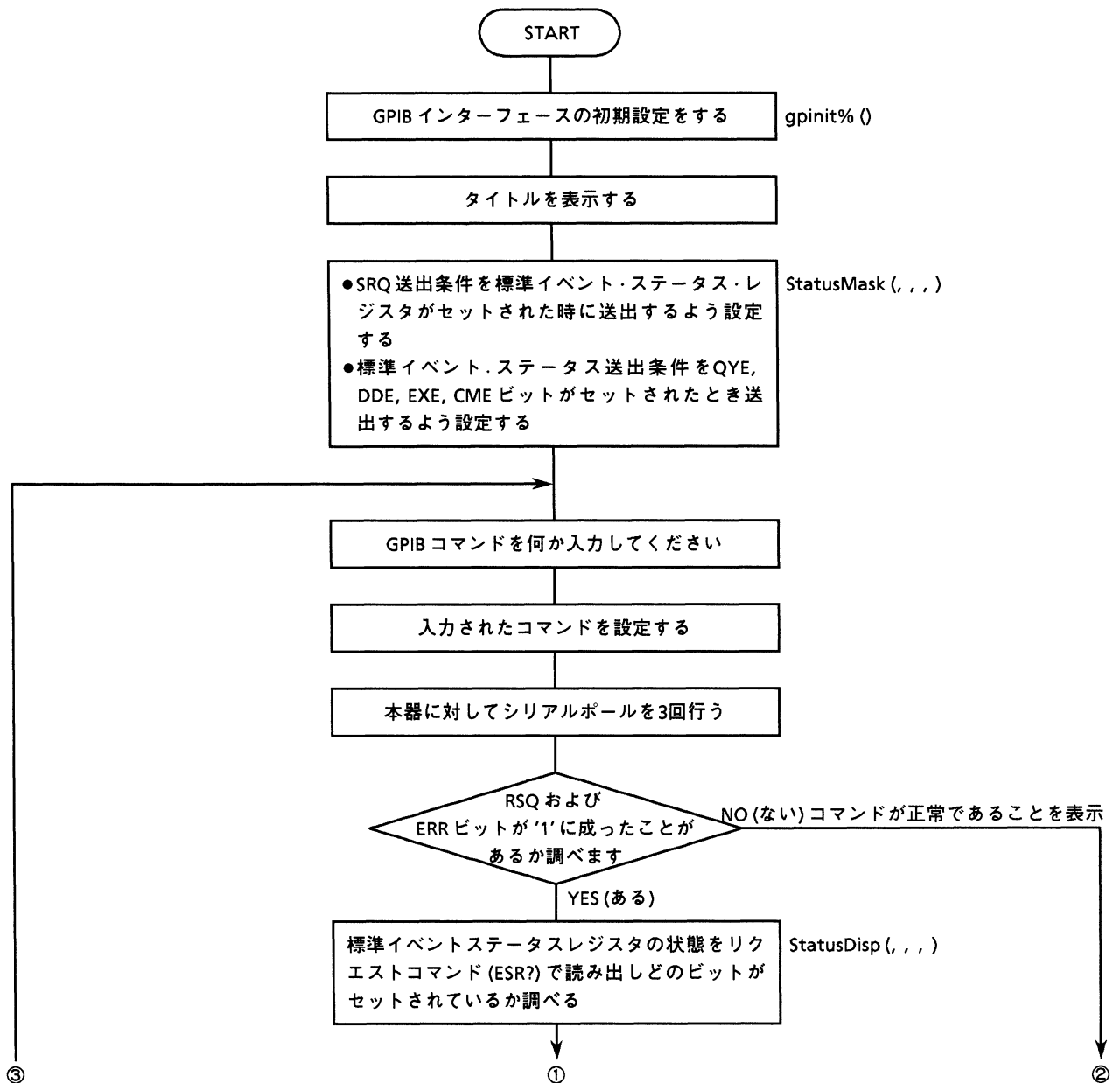
RETURN

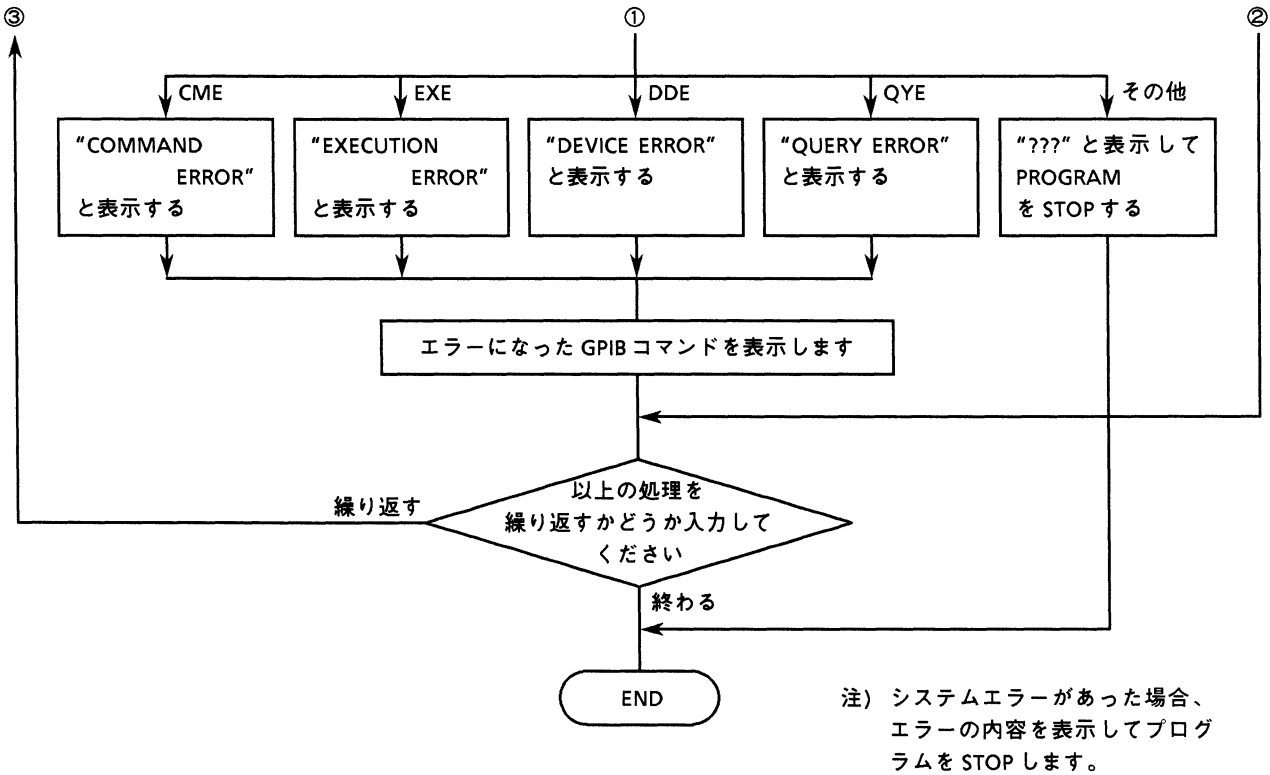
Fderr: '===== FD error message =====
CALL wrtcmd2("FDE?")
RD$ = LEFT$(readcmd2$, IBCNT% - 1)
LOCATE 10, 1
IF RD$ <> "FDE 10" THEN
    PRINT "FD error occuerd!! "
    SELECT CASE MID$(RD$, 6, 1)
        CASE "0"
            PRINT "E0:Media error"
        CASE "1"
            PRINT "E1:Write protection error"
        CASE "2"
            PRINT "E2:File full"
        CASE "3"
            PRINT "E3:File not found"
        CASE "4"
            PRINT "E4:File already exists error"
        CASE "5"
            PRINT "E5:Write error"
    
```

```
        CASE "6"
            PRINT "E6:Read error"
        CASE "7"
            PRINT "E7:File type , File error"
        CASE "8"
            PRINT "E8:FD error"
        CASE "9"
            PRINT "E9:Hardware error"
    END SELECT
ELSE
    PRINT "<<  ** FD operation complete!! **  >>"
    PRINT "<<  ** Accept file number is " + NO$ + ". **  >>"
END IF
RETURN
```

(11) 標準ステータスバイト (4種類) のチェック

本プログラムは標準ステータスバイト (QYE, DDE, EXE, CME ビット) のチェックをします。





● プログラム・リスト

```

REM $INCLUDE: 'c:\wat-gpib\qbasic\qbdecl.bas'

COMMON SHARED DEV%, GPIB0%, PPG%, ED%

DECLARE SUB waidly (tim!)
DECLARE SUB wrtcmd2 (WRT$)
DECLARE SUB trap ()
DECLARE SUB ClearDisp (p%, l%)
DECLARE SUB StatusDisp (stb%, esr%, esr2%, esr3%)
DECLARE SUB StatusMask (s0%, s1%, s2%, s3%)
DECLARE FUNCTION itob$ (l%, V%)
DECLARE FUNCTION gpinit% ()
DECLARE FUNCTION readcmd2$ ()

CLS
IF gpinit% <> 0 THEN                                'Setup interface

PRINT "*** MP1762D/MP1764D STANDARD STATUS REGISTER CHECK ***"
PRINT

'===== Set MSS status byte register =====
CALL StatusMask(&H3C, &H7E, &H77F, &H3)

DO
  CALL ClearDisp(5, 15)
  LOCATE 5, 1
  INPUT "Please enter some GPIB command(s):"; com$
  length% = LEN(com$)
  CALL wrtcmd2(com$)
  LOCATE 5, 1
  PRINT "Please enter some GPIB command(s):"

  sta% = 0
  FOR i = 0 TO 2
    CALL IBRSP(ED%, SPR%)
    IF IBSTA < 0 THEN CALL trap
    sta% = sta% OR SPR%

    sta$ = itob$(8, SPR%)
    LOCATE 1, 60
    PRINT "*SRE: "; sta$

    waidly (.1)
  NEXT i

  IF (sta% AND &H20) THEN
    LOCATE 7, 1
    PRINT "Execution command(s) fail of '"
    IF length% > 0 THEN
      LOCATE 7, 1
      PRINT "Execution command(s) fail of '"; LEFT$(com$, length%); "'
    END IF

    CALL StatusDisp(dmy%, reg%, dmy2%, dmy3%)

    CALL ClearDisp(8, 6): LOCATE 8, 1
    IF (reg% AND &H2) OR (reg% AND &H40) THEN PRINT " ??? ": STOP
    IF reg% AND &H4 THEN PRINT "* QUERY ERROR *"
    IF reg% AND &H8 THEN PRINT "* DEVICE ERROR *"
    IF reg% AND &H10 THEN PRINT "* EXECUTION ERROR *"
    IF reg% AND &H20 THEN PRINT "* COMMAND ERROR *"
  ELSE

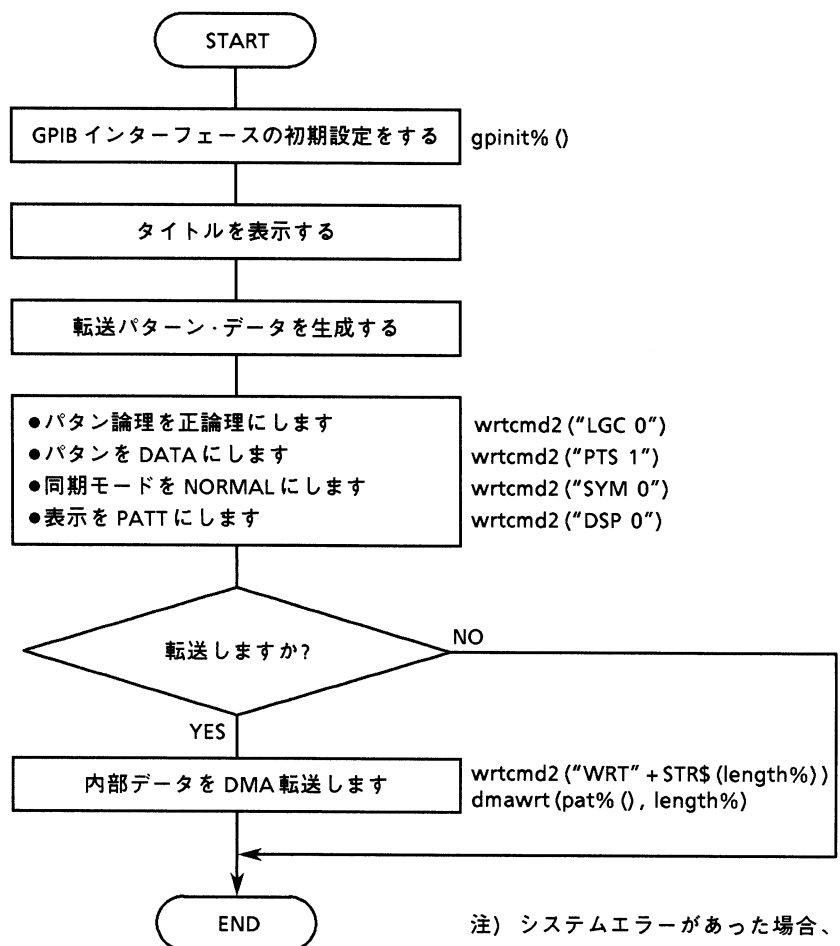
```

```
        LOCATE 7, 1
        PRINT "Command succed execution."
        CALL ClearDisp(9, 6)
    END IF

    LOCATE 15, 36: PRINT "
    LOCATE 15, 1
    INPUT "Do you test other command? [Yes/No] "; loop$
    LOOP UNTIL loop$ = "n" OR loop$ = "N"
END IF
'
STOP
```

(12) パターンデータの DMA 転送処理

本プログラムはパターンデータの DMA 転送処理を行います



注) システムエラーがあった場合、エラーの内容を表示してプログラムを STOP します。

● プログラム・リスト

```

REM $INCLUDE: 'c:\wat-gpib\qbasic\qbdecl.bas'

COMMON SHARED DEV%, gpib0%, PPG%, ED%

DECLARE SUB StatusMask (s0%, s1%, s2%, s3%)
DECLARE SUB StatusDisp (stb%, esr%, esr2%, esr3%)
DECLARE SUB wrtcmd2 (w$)
DECLARE SUB dmawrt (w$(), i%)
DECLARE SUB gpiberr (msg$)
DECLARE FUNCTION gpinit% ()
DECLARE FUNCTION Exchange% (i%)

DIM pat%(302)

IF gpinit% <> 0 THEN          'Setup interface
  CLS
  PRINT "** MP1762D/MP1764D DMA(pattern data) SAMPLE PROGRAM ** "
  PRINT

  CALL StatusMask(&H0, &H0, &H0, &H0)
  '
  '===== Table =====
  ' Test pattern set and swap data.
  ' if you use ibconfig() swap function,
  ' then don't call Exchange() function. Because, its same operation.
  '
  Dlength% = 300                      ' :Max Page
  j% = 0
  FOR i% = 0 TO Dlength% - 1
    pat%(i%) = Exchange(j%)
    j% = j% + 1
  NEXT i%
  pat%(i%) = &HA

  '===== initial =====
  CALL wrtcmd2("LGC 0")      'Pattern logic      : positive
  CALL wrtcmd2("PTS 1")      'Pattern            : data
  CALL wrtcmd2("SYM 0")      'Sync mode       : normal
  CALL wrtcmd2("DSP 0")      'Display         : pattern
  CALL wrtcmd2("DLN " + STR$(Dlength% * 16)): DATA Length

  CALL StatusDisp(dmy%, dmy1%, dmy2%, dmy3%)

  INPUT "Do you wish transmit are PATTERN data? [Yes/No]:"; a$
  IF a$ = "y" OR a$ = "Y" THEN
    CALL wrtcmd2("WRT " + STR$(Dlength% * 2) + ",0")

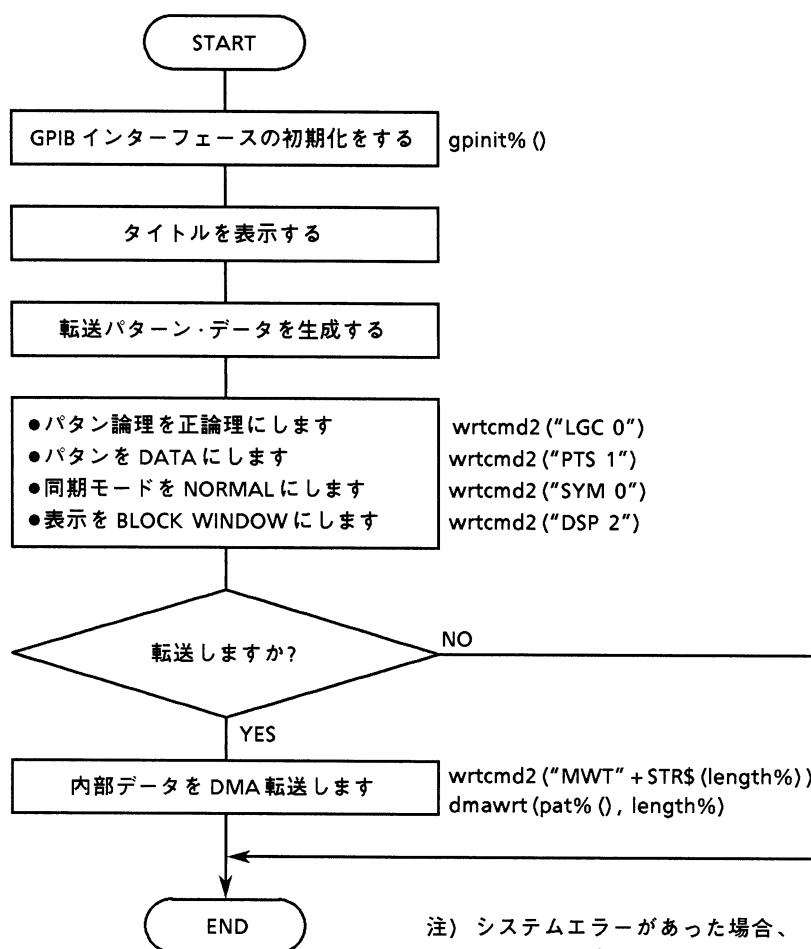
    ' CALL ibconfig(gpib0%, 20, 1)
    ' CALL gpiberr("'ibconfig' execute status")

    CALL dmawrt(pat%(), Dlength%)
  END IF
END IF
STOP

```

(13) BLOCK WINDOW データの DMA 転送処理

本プログラムはBLOCK WINDOW データの DMA 転送処理を行います



注) システムエラーがあった場合、エラーの内容を表示してプログラムを STOP します。

● プログラム・リスト

```

' ---- Note ----
' If you are use to control command of 'MWT' then you must set a data length
' to multiplication by integral number of 32-bit.
'
REM $INCLUDE: 'c:\wat-gpib\qbasic\qbdecl.bas'

COMMON SHARED DEV%, GPIB0%, PPG%, ED%

DECLARE SUB StatusMask (s0%, s1%, s2%, s3%)
DECLARE SUB StatusDisp (stb%, esr%, esr2%, esr3%)
DECLARE SUB wrtcmd2 (WRT$)
DECLARE SUB dmawrt (w%(), i%)
DECLARE FUNCTION gpinit% ()

DIM pat%(310)

IF gpinit% <> 0 THEN          'Setup interface
  CLS
  PRINT "** MP1762D/MP1764D DMA(block window data) SAMPLE PROGRAM ** "
  PRINT

  CALL StatusMask(&H0, &H0, &H0, &H0)
  '
  '===== Table =====
  ' Test pattern set and swap data.
  ' if you use ibconfig() swap function,
  ' then don't call Exchange() function. Because, its same operation.
  '
  Dlength% = 300
  FOR i% = 0 TO Dlength% - 1
    pat%(i%) = &H5555
  NEXT i%
  pat%(i%) = &HA              'Append termination code (LF)

  '===== initial =====
  CALL wrtcmd2("LGC 0")      'Pattern logic : positive
  CALL wrtcmd2("PTS 1")      'Pattern      : data
  CALL wrtcmd2("SYM 0")      'Sync mode    : normal
  CALL wrtcmd2("DSP 2")      'Display      : pattern

  CALL StatusDisp(dmy%, dmy1%, dmy2%, dmy3%)

  INPUT "Do you wish transmit are BLOCK WINDOW data? [Yes/No]:"; a$
  IF a$ = "y" OR a$ = "Y" THEN
    CALL wrtcmd2("MWT " + STR$(Dlength% * 2) + ",0")

    CALL dmawrt(pat%(), Dlength%)
  END IF
END IF
STOP

```

付 録 A 従来器とのコンパチビリティ

従来器(ここでは、MP1702A/MP1609A/MP1651Aを示します。)で用いていたプログラムで本器を実行する場合、以下に示す項目について、多少プログラムの再編集が必要となります。

(1) ステータス・共通コマンド構成

MP1762Dでは488.2に準拠した共通コマンドの一部をサポートしていますが、MP1702Aではこの共通コマンドを*無しで表現していました。

コ マ ン ド	MP1702A	MP1762C
サービス・リクエスト	STB?	*STB?
サービス・リクエスト・イネーブル・レジスタ	SRQ	*SRE
標準イベント・ステータス・レジスタ	ESR?	*ESR?
標準イベント・ステータス・イネーブル・レジスタ	ESE	*ESE
拡張イベント・ステータス・レジスタ	EER?	ESR2? ESR3?
拡張イベント・ステータス・イネーブル・レジスタ	EES	ESE2? ESE3?

その他の共通コマンド、およびステータス構成は「第8章 ステータス・ストラクチャー」を参照してください。

(2) 測定データの集計・読みだし

MP1762Dの測定データの読みだしコマンドは、データリクエストメッセージに対する応答の形で出力されます。

また、測定結果データに関しても、内部バッファに保存したデータを出力します。

測定データの集計に関しては、プログラムの再編集を行ってください。

(3) その他の GPIB コマンド

表 A-1に MP1702A 等との GPIB コマンド対応表を示します。

各機器の GPIB 取扱説明書を見比べて、プログラムの再編集をお願いします。

○ : MP1702A と共通のコマンド

× : MP1702A と共通で無いコマンド

表 A-1 デバイス・メッセージ一覧表

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細		MP1702A との コンパチ ビリティ
	ヘッダ部	数値データ部	ヘッダ部	No.	ページ	
● INPUT セクション						
データ入力スレッシュOLD電圧	DTH	NR2 形式	DTH?	1	P9-26	○
アイ・マージン測定結果 (スレッシュOLD)	—	—	THM?	2	P9-27	×
クロック入力位相	CPA	NR1形式	CPA?	3	P9-28	○
アイ・マージン測定結果(フェーズ)	—	—	PHM?	4	P9-29	×
データ入力終端電圧	DTM	NR1形式	DTM?	5	P9-30	○
クロック入力終端電圧	CTM	NR1形式	CTM?	6	P9-31	○
ディレイ状態	—	—	DLY?	7	P9-32	○
自動位相スレッシュOLD・サーチ機能	SRH	NR1形式	SRH?	8	P9-33	○
クロック入力極性	CPL	NR1形式	CPL?	9	P9-34	○
アイ・マージン測定表示切替	EME	NR1形式	EME?	10	P9-35	×
アイ・マージン測定開始	EST	NR1形式	EST?	11	P9-36	×
アイ・マージン測定 (エラー・レシオ選択)	EYT	NR1形式	EYT?	12	P9-37	×
● MEMORY セクション						
メモリ・FD モード	—	—	FMD?	13	P9-39	×
ファイル内容の検索	—	—	FSH?	14	P9-40	×
ファイル No./ディレクトリモード切替	FIL	NR1形式	FIL?	15	P9-42	×
フロッピー・データ・リコール	RCL	NR1形式	—	16	P9-43	○
フロッピー・データ・デリート	DEL	NR1形式	—	17	P9-44	×
フロッピー・データ・セーブ	SAV	NR1形式	—	18	P9-45	○
フロッピー・データ・リセーブ	RSV	NR1形式	—	19	P9-46	○
メモリ・モード切替	MEM	NR1形式	MEM?	20	P9-47	○
フロッピー・アクセス状態	—	—	MAC?	21	P9-48	○
FD エラーメッセージ	—	—	FDE?	22	P9-49	×
FD フォーマット	FDF	—	—	23	P9-50	×
● PATTERN セクション						
パターン論理	LGC	NR1形式	LGC?	24	P9-52	○
測定パターン	PTS	NR1形式	PTS?	25	P9-53	×

表 A-1 デバイス・メッセージ一覧表(つづき)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細		MP1702A との コンパチ ビリティ
	ヘッダ部	数値データ部	ヘッダ部	No.	ページ	
● PATTERN セクション(つづき)						
ZERO SUBST および PRBS 段数	PTN	NR1形式	PTN?	26	P9-54	○
PRBS マーク率	MRK	NR1形式	MRK?	27	P9-55	○
同期方式	SYM	NR1形式	SYM?	28	P9-56	×
表示選択	DSP	NR1形式	DSP?	29	P9-57	×
オルタネート・パターン A/B 切替	ALT	NR1形式	ALT?	30	P9-58	×
フレーム・ビット長	FLN	NR1形式	FLN?	31	P9-59	○
データ長	DLN	NR1形式	DLN?	32	P9-60	×
ZERO SUBST 長	ZLN	NR1形式	ZLN?	33	P9-62	×
ページ数・パターン同期トリガ位置	PAG ADR	NR1形式 NR1形式	PAG? ADR?	34	P9-63	×
パターン・ビット	BIT	NR1形式 HEX 形式	BIT?	35	P9-64	○
ビット・ウインドウ・パターン	CHM	NR1形式 HEX 形式	CHM?	36	P9-66	×
ビット・ウインドウ・ページ	MSK	NR1形式	MSK?	37	P9-68	×
ブロック・ウインドウ・パターン	MGB	NR1形式 HEX 形式	MGB?	38	P9-69	×
誤り分析データ	—	—	EAB?	39	P9-71	×
誤り分析ページ	EAP	NR1形式	EAP?	40	P9-72	×
ビット・ウインドウ ON/OFF	MSE	NR1形式	MSE?	41	P9-73	×
ブロック・ウインドウ ON/OFF	MGE	NR1形式	MGE?	42	P9-74	×
誤り分析トリガ	EAT	NR1形式	EAT?	43	P9-75	×
パターン・データ入力バイト数	WRT	NR1形式	—	44	P9-76	×
パターン・データ出力バイト数	—	—	RED?	45	P9-77	×
ブロック・ウインドウ・データ入力 バイト数	MWT	NR1形式	—	46	P9-78	×
ブロック・ウインドウ・データ出力 バイト数	—	—	MRD?	47	P9-80	×
パターンデータ・プリセット (全ページ、全ビット)	ALL	NR1形式	—	48	P9-82	○

表 A-1 デバイス・メッセージ一覧表(つづき)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細		MP1702A との コンパチ ビリティ
	ヘッダ部	数値データ部	ヘッダ部	No.	ページ	
● PATTERN セクション(つづき)						
パターンデータ・プリセット (1ページ、全ビット)	PST	NR1形式	—	49	P9-83	○
ブロック・ウインドウ・プリセット (全ページ、全ビット)	MAL	NR1形式	—	50	P9-84	×
ブロック・ウインドウ・プリセット (1ページ、全ビット)	MPS	NR1形式	—	51	P9-85	×
ビット・ウインドウ・プリセット (全ページ、全ビット)	HAL	NR1形式	—	52	P9-86	×
ビット・ウインドウ・プリセット (1ページ、全ビット)	HPS	NR1形式	—	53	P9-87	×
パターン同期トリガ位置	PSP	NR1形式	PSP?	54	P9-88	—
ページ・パターン同期トリガ位置表示 切替	PPD	NR1形式	PPD?	55	P9-89	—
● MEASUREMENT セクション						
クロック断状態	—	—	CLI?	56	P9-91	○
同期外れ状態	—	—	SLI?	57	P9-92	○
誤り検出状態	—	—	ERS?	58	P9-93	○
測定結果表示モード	DMS	NR1形式	DMS?	59	P9-94	○
途中結果表示機能	CUR	NR1形式	CUR?	60	P9-95	○
測定モード	MOD	NR1形式	MOD?	61	P9-96	○
測定開始および再スタート	STA	—	—	62	P9-97	○
測定停止	STO	—	—	63	P9-98	○
測定状態	—	—	MSR?	64	P9-99	○
自動同期機能	SYN	NR1形式	SYN?	65	P9-100	○
自動同期スレッシュホールド	SYE	NR1形式	SYE?	66	P9-101	×
リアルタイム・測定時間表示切替	TIM	NR1形式	TIM?	67	P9-102	○
内部タイマ設定	RTM	NR1形式	RTM?	68	P9-103	○
測定時間設定	PRD	NR1形式	PRD?	69	P9-104	○
誤り率測定結果	—	—	ER?	70	P9-105	○
誤り個数測定結果	—	—	EC?	71	P9-106	○
クロック個数測定結果	—	—	CC?	72	P9-107	○

表 A-1 デバイス・メッセージ一覧表(つづき)

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細		MP1702A との コンパチ ビリティ
	ヘッダ部	数値データ部	ヘッダ部	No.	ページ	
● MEASUREMENT セクション(つづき)						
EI 個数測定結果	—	—	EI?	73	P9-108	○
%EFI 測定結果	—	—	EFI?	74	P9-109	○
クロック周波数データ	—	—	FRQ?	75	P9-110	○
1秒データ測定結果	—	—	OSD?	76	P9-111	×
アラーム測定結果	—	—	AMD?	77	P9-112	×
測定終了データをバッファにストア	EDS	—	—	78	P9-113	×
測定終了データのバッファをクリア	EDC	—	—	79	P9-114	×
測定終了データの出力	—	—	END?	80	P9-115	×
測定中間データをバッファにストア	IMS	—	—	81	P9-117	×
測定中間データのバッファをクリア	IMC	—	—	82	P9-118	×
測定中間データの出力	—	—	IMD?	83	P9-119	×
● その他のセクション(正面パネル)						
プリンタ ON/OFF	PRN	NR1形式	PRN?	84	P9-122	○
マニュアル・プリント	PSA	NR1形式	—	85	P9-123	○
アラーム・モニタ(アラーム検出)	ALM	NR1形式	ALM?	86	P9-124	○
アラーム・モニタ(誤り検出)	MON	NR1形式	MON?	87	P9-125	○
同期信号出力	SOP	NR1形式	SOP?	88	P9-126	×
● その他のセクション(背面パネル GPIB)						
GPIB2アドレス	GPA	NR1形式	GPA?	89	P9-127	×
● その他のセクション (背面パネル・FUNCTION スイッチ)						
マーク率の AND ビットシフト数	SFT	NR1形式	SFT?	90	P9-128	○
クロック断処理機能	CLS	NR1形式	CLS?	91	P9-129	○
同期外れ処理機能	SLS	NR1形式	SLS?	92	P9-130	○
エラー・パフォーマンス・データ・ スレッシュホールド選択機能	ETH	NR1形式	ETH?	93	P9-131	○
BURST 測定モード	BST	NR1形式	BST?	94	P9-132	×
測定途中データ算出機能	CAL	NR1形式	CAL?	95	P9-133	○
誤り検出モード	ETY	NR1形式	ETY?	96	P9-134	○

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細		MP1702A との コンパチ ビリティ
	ヘッダ部	数値データ部	ヘッダ部	No.	ページ	
● その他のセクション (背面パネル・FUNCTION スイッチ) (つづき)						
EI, EFI インターバル時間	EIT	NR1形式	EIT?	97	P9-135	×
データ印字フォーマット	FMT	NR1形式	FMT?	98	P9-136	○
スレッシュホールド EI、EFI データ印字	THR	NR1形式	THR?	99	P9-137	○
エラー・パフォーマンス・データ印字	EPF	NR1形式	EPF?	100	P9-138	○
中間データ印字	ITM	NR1形式	ITM?	101	P9-139	○
1秒データ印字	OSC	NR1形式	OSC?	102	P9-140	○
1秒データ印字スレッシュホールド	DOT	NR1形式	DOT?	103	P9-141	○
紙節約機能	PSV	NR1形式	PSV?	104	P9-142	○
測定インタバル時間	ITV	NR1形式	ITV?	105	P9-143	×
メモリ・FD モード	—	—	FMD?	13	P9-39	×
終端コード切り替え	TRM	NR1形式	TRM?	106	P9-144	○

MP1762D : 付属新規コマンド

機 能 項 目	コントロール・メッセージ		データ・リクエスト・メッセージ	デバイス・メッセージ詳細		MP1702A との コンパチ ビリティ
	ヘッダ部	数値データ部	ヘッダ部	No.	ページ	
入力データ切り替え	DSL	NR1形式	DCD?	107	P9-145	×
Tracking/Adjust 操作	DAC	NR1形式	DCD?	108	P9-146	×
入力信号数の設定	DSD	NR1形式	DCD?	109	P9-147	×
供給クロックの選択	RRC	NR1形式	DRC?	110	P9-148	×
PLL リセット/ PLL ステータス	RPL	—	RPL?	111	P9-148	×
Recovery 周波数設定	RFQ	NR1形式	DRC?	112	P9-150	×
終端の差動選択	DTC	NR1形式	DCD?	113	P9-152	×

付 録 B パターンの DMA 転送

DMA とは、**Direct Memory Access** の略で、大量のデータ転送を高速に行うことです。

本器には、DMA 転送コマンドを4種類もうけています。この4種類の転送方法について説明します。

(1) パターンデータの DMA 転送コマンド (WRT)

測定パターンが **ALTERNATE, DATA** の場合にプログラム・パターンの **DMA** 転送内容を前もって送出するコマンドです。

本コマンドでは、

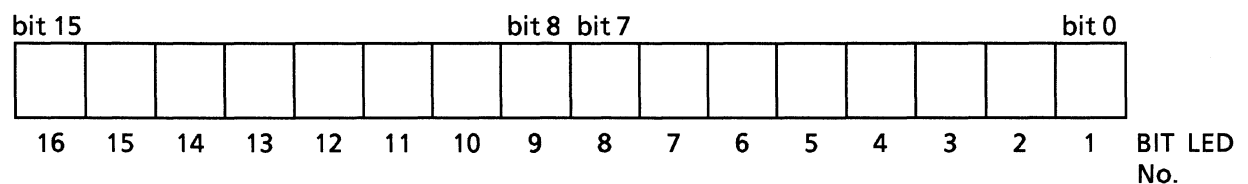
- 1：何バイトのデータを転送するのか？
- 2：転送したパターンデータを本器内部 **RAM** エリアの何番目のアドレスから格納するのか？

の情報を **WRT** をもちいて本器に知らせます。

1) 何バイトのデータを転送するのか？

本器は**16 bit** のパターン構成になっています。そのため通常表示されている **BIT** 表示器 1ページ (**16 bit**) に対して2バイトの転送を行います。

パターンデータとビットの対応は、以下の様になっています。



奇数バイト転送の場合は、上位ビット (bit 15～bit 8) のみ設定されます。

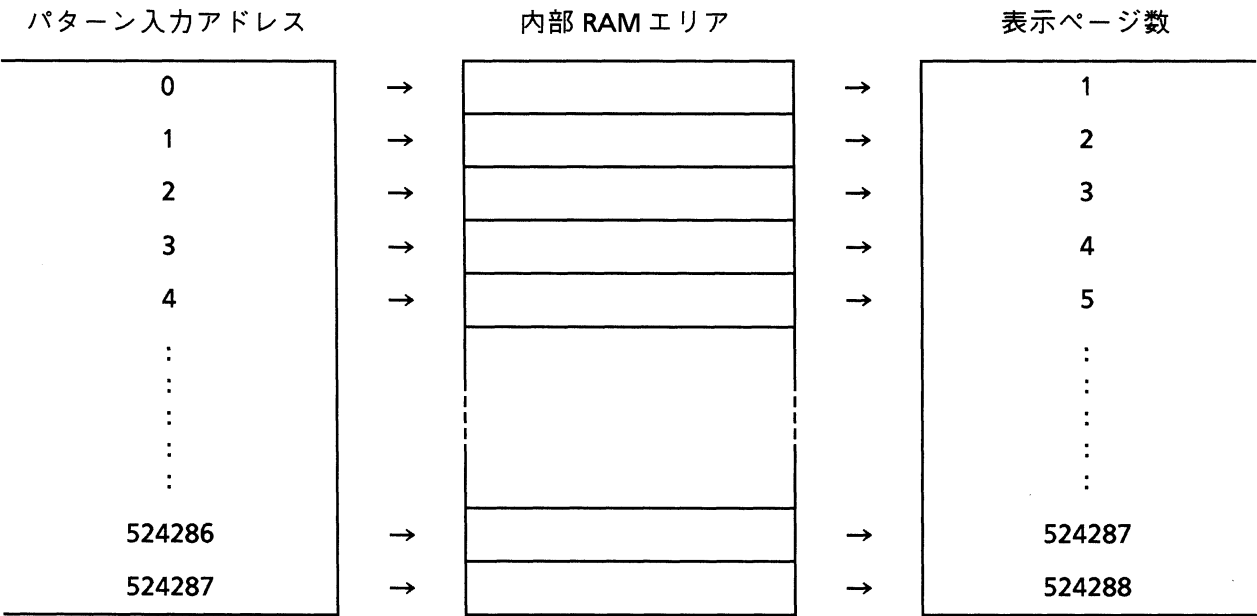
2) 転送したパターンデータを本器内部 **RAM** エリアの何番目のアドレスから格納するのか?

本器の内部 **RAM** アドレスは、

DATA の場合 : 0 ~ 524287
ALTERNATE の場合 : 0 ~ 262143

で、**ALTERNATE** の場合は **A** パターン / **B** パターンの2種類で内部 **RAM** エリアを2分割しています。(A/B 表示切替の状態によって **A** パターン・**B** パターンの転送を選択します。)

また、以下に実際のページ数とアドレスとの関係を示します。



(2) パターンデータの DMA 転送コマンド (RED?)

測定パターンが **ALTERNATE, DATA** の場合にプログラム・パターンの **DMA** 転送内容を前もって送出するコマンドです。

本コマンドでは、

1：何バイトのデータを転送するのか？

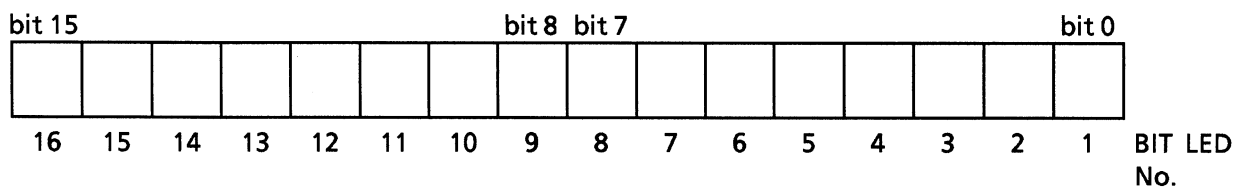
2：転送するパターンデータを本器内部 **RAM** エリアの何番目のアドレスから出力するのか？

の情報を **RED?** をもちいて本器に知らせます。

1) 何バイトのデータを転送するのか？

本器は**16 bit** のパターン構成になっています。そのため通常表示されている **BIT** 表示器 **1** ページ (**16 bit**) に対して**2** バイトの転送を行います。

パターンデータとビットの対応は、以下のようにになっています。



奇数バイト転送の場合は、上位ビット (**bit 15~bit 8**) のみ設定されます。

2) 転送したパターンデータを本器内部 **RAM** エリアの何番目のアドレスから出力するのか?

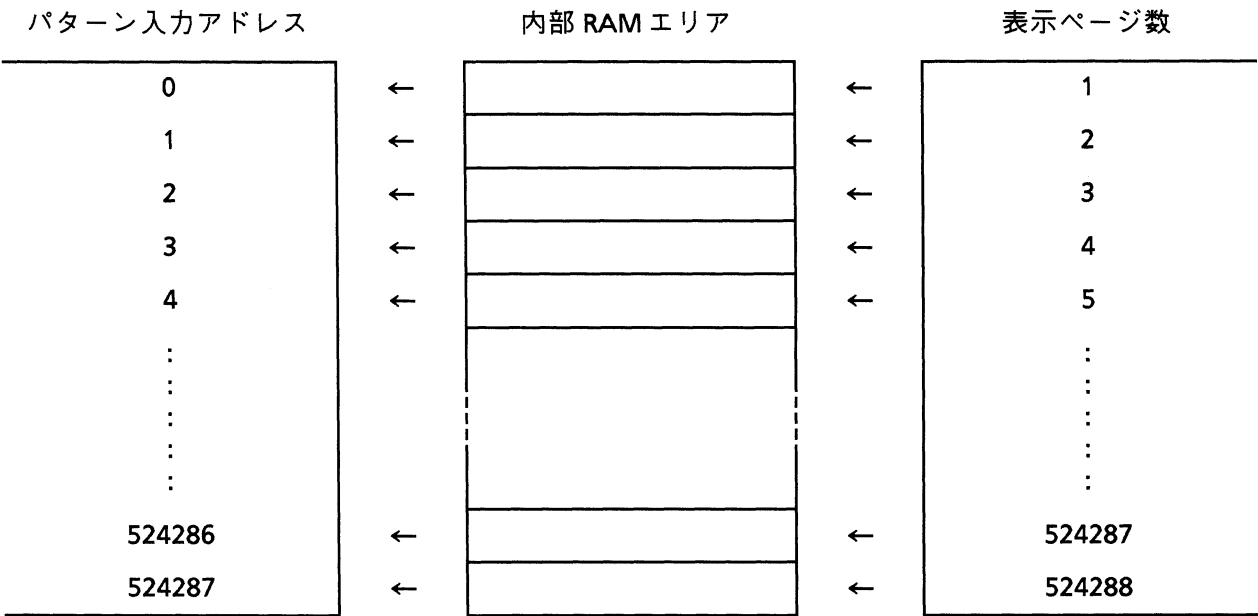
本器の内部 **RAM** アドレスは、

DATA の場合 : 0 ~ 524287

ALTERNATE の場合 : 0 ~ 262143

で、**ALTERNATE** の場合は **A** パターン / **B** パターンの2種類で内部 **RAM** エリアを2分割しています。(A/B 表示切替の状態によって **A** パターン・**B** パターンの転送を選択します。)

また、以下に実際のページ数とアドレスとの関係を示します。



(3) BLOCK WINDOW データの DMA 転送コマンド (MWT)

BLOCK WINDOW データの場合に BLOCK WINDOW パターンの DMA 転送内容を前もって送出するコマンドです。

本コマンドでは、

- 1: 何バイトのデータを転送するのか?
- 2: 転送するパターンデータを本器内部 RAM エリアの何番目のアドレスに格納するのか?

の情報を MWT をもちいて本器に知らせます。

1) 何バイトのデータを転送するのか?

BLOCK WINDOW データは32ビット単位であるため内部 RAM エリアには BIT 表示器 32ビット分 (2ページ) 分のデータを1 bit として保持しています。

そのため、上記 (1)、(2) とは異なり、設定値の1 bit が本器の BLOCK WINDOW データ 32ビット分を表します。

パターンデータとビットの対応は、以下のようにになっています。

bit 15				bit 8				bit 7				bit 0				
15	13	11	9	7	5	3	1	31	29	27	25	23	21	19	17	PAGE No.
16	14	12	10	8	6	4	2	32	30	28	26	24	22	20	18	

上記は、パターン先頭アドレスを0とした場合のページ値で、アドレスが1増える毎に32上記ページ値に加えて下さい。

奇数バイト転送の場合は、上位ビット (bit 15~bit 8) のみ設定されます。

2) 転送したパターンデータを本器内部 RAM エリアの何番目のアドレスから格納するのか？

本器の内部 RAM アドレスは、

DATA の場合 : 0 ～ 16383

ALTERNATE の場合 : 0 ～ 8192

で、ALTERNATE の場合は A パターン / B パターンの2種類で内部 RAM エリアを2分割しています。(A/B 表示切替の状態によって A パターン・B パターンの転送を選択します。)

また、以下に実際のページ数とアドレスとの関係を示します。

パターン入力アドレス		内部 RAM エリア		表示ページ数
0	→		→	1 ～ 32ページ
1	→		→	33 ～ 64ページ
2	→		→	65 ～ 96ページ
3	→		→	97 ～ 128ページ
4	→		→	129 ～ 160ページ
⋮				⋮
⋮				⋮
⋮				⋮
⋮				⋮
16382	→		→	524225 ～ 524256ページ
16383	→		→	524257 ～ 524288ページ

(4) BLOCK WINDOW データの DMA 転送コマンド (MRD?)

BLOCK WINDOW データの場合に BLOCK WINDOW パターンの DMA 転送内容を前もって送出するコマンドです。

本コマンドでは、

1 : 何バイトのデータを転送するのか?

2 : 転送するパターンデータを本器内部 RAM エリアの何番目のアドレスから出力するのか?

の情報を MRD? をもちいて本器に知らせます。

1) 何バイトのデータを転送するのか?

BLOCK WINDOW データは32ビット単位であるため内部 RAM エリアには BIT 表示器 32ビット分 (2ページ) 分のデータを1 bit として保持しています。

そのため、上記 (1)、(2) とは異なり、設定値の1 bit が本器の BLOCK WINDOW データ 32ビット分を表します。

パターンデータとビットの対応は、以下のようになっています。

bit 15				bit 8				bit 7				bit 0				
15	13	11	9	7	5	3	1	31	29	27	25	23	21	19	17	PAGE No.
16	14	12	10	8	6	4	2	32	30	28	26	24	22	20	18	

上記は、パターン先頭アドレスを0とした場合のページ値で、アドレスが1増える毎に32上記ページ値に加えて下さい。

奇数バイト転送の場合は、上位ビット (bit 15~bit 8) のみ設定されます。

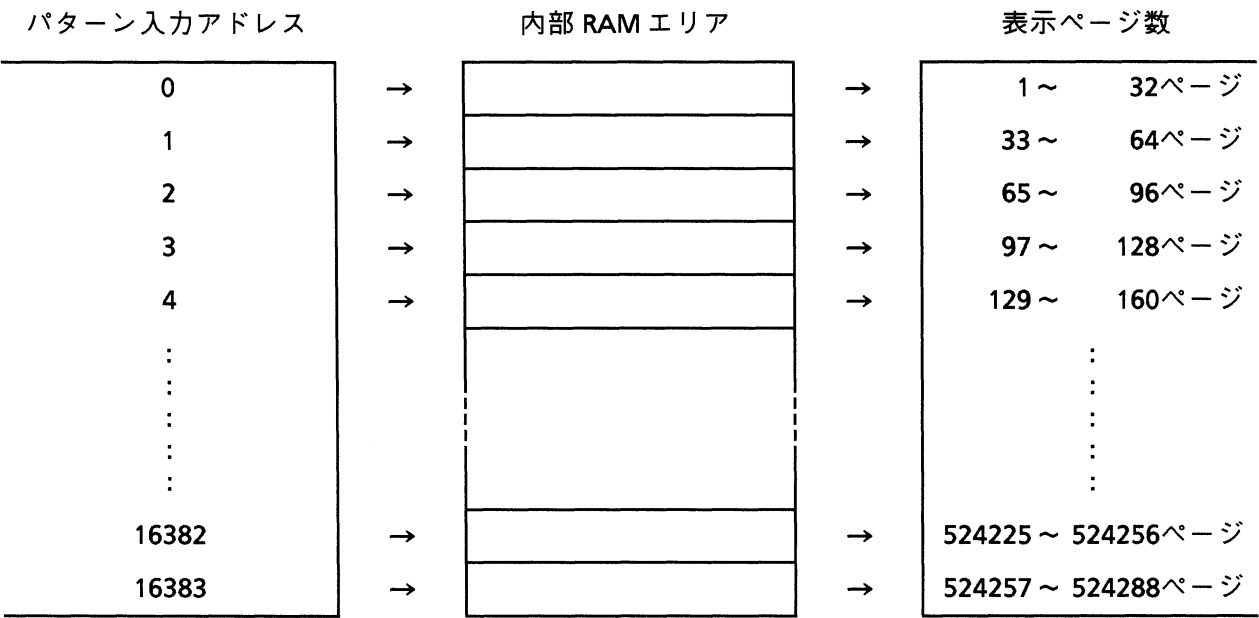
2) 転送したパターンデータを本器内部 **RAM** エリアの何番目のアドレスから出力するのか？

本器の内部 **RAM** アドレスは、

DATA の場合 : 0 ~ 16383
ALTERNATE の場合 : 0 ~ 8192

で、**ALTERNATE** の場合は **A** パターン / **B** パターンの2種類で内部 **RAM** エリアを2分割しています。(**A** / **B** 表示切替の状態によって **A** パターン・**B** パターンの転送を選択します。)

また、以下に実際のページ数とアドレスとの関係を示します。



付 録 C 初期値一覧表

MP1762Dの工場出荷時の初期値を以下の表 C-1 ～ C-5に示します。

表 C-1 INPUT セクション初期値一覧表

機 能 項 目	ヘッダ	初 期 値	デバイス・メッセージ詳細	
			項目 No.	ページ
● INPUT セクション				
データ入力スレッシュホールド電圧	DTH	-0.500V	1	P9-26
アイ・マージン測定結果 (スレッシュホールド)	THM	測定結果無し	2	P9-27
クロック入力位相	CPA	0 ps	3	P9-28
アイ・マージン測定結果 (フェーズ)	PHM	測定結果無し	4	P9-29
データ入力終端電圧	DTM	GND	5	P9-30
クロック入力終端電圧	CTM	GND	6	P9-31
ディレイ状態	DLY	READY 状態	7	P9-32
自動位相スレッシュホールド・サーチ機能	SRH	OFF	8	P9-33
クロック入力極性	CPL	CLOCK	9	P9-34
アイ・マージン測定表示切替	EME	入力スレッシュホールド / 入力位相	10	P9-35
アイ・マージン測定開始	EST	アイマージン測定停止	11	P9-36
アイ・マージン測定 (エラー・レシオ選択)	EYT	1.0E-2	12	P9-37
入力データ切替	DSL	DATA	107	P9-145
Tracking/Adjust 操作	DAC	Tracking ON (Adjust OFF)	108	P9-146
入力信号数の設定	DSD	Differential	109	P9-147
供給クロックの選択	RRC	External	110	P9-148
Recovery 周波数設定	RFQ	9.900 GHz	111	P9-149
終端の差動選択	DTC	Single / 50 Ω	113	P9-152

表 C-2 MEMORY セクション初期値一覧表

機 能 項 目	ヘッダ	初 期 値	デバイス・ メッセージ詳細	
			項目 No.	ペー ジ
● MEMORY セクション				
メモリ・FD モード	FMD	1440 K	13	P9-39
ファイル内容の検索	FSH	データ無し	14	P9-40
ファイル No. / ディレクトリモード 切替	FIL	File No.	15	P9-42
フロッピー・データ・リコール	RCL	—	16	P9-43
フロッピー・データ・デリート	DEL	—	17	P9-44
フロッピー・データ・セーブ	SAV	—	18	P9-45
フロッピー・データ・リセーブ	RSV	—	19	P9-46
メモリ・モード切替	MEM	PATT	20	P9-47
フロッピー・アクセス状態	MAC	ノンアクセス	21	P9-48
FD エラーメッセージ	FDE	No error	22	P9-49
FD フォーマット	FDF	—	23	P9-50

表 C-3 PATTERN セクション初期値一覧表

機 能 項 目	ヘッダ	初 期 値	デバイス・メッセージ詳細	
			項目 No.	ページ
● PATTERNセクション				
パターン論理	LGC	Positive	24	P9-52
測定パターン	PTS	PRBS	25	P9-53
ZERO SUBST および PRBS 段数	PTN	PRBS : $2^{15}-1$ Zero : 2^{15}	26	P9-54
PRBS マーク率	MRK	1 / 2	27	P9-55
同期方式	SYM	Alternate : Normal Data : Normal Zero sunbst: Normal PRBS : 無し	28	P9-56
表示選択	DSP	Pattern	29	P9-57
オルタネート・パターン A/B 切替	ALT	A	30	P9-58
フレーム・ビット長	FLN	32ビット	31	P9-59
データ長	DLN	Alternate : 128ビット Data : 2ビット	32	P9-60
ZERO SUBST 長	ZLN	1ビット	33	P9-62
ページ数・パターン同期トリガ位置	PAG ADR	1ページ	34	P9-63
パターン・ビット	BIT	全ビット 0	35	P9-64
ビット・ウインドウ・パターン	CHM	全ビット 0	36	P9-66
ビット・ウインドウ・ページ	MSK	1ページ	37	P9-68
ブロック・ウインドウ・パターン	MGB	全ビット 0	38	P9-69
誤り分析データ ※1	EAB	測定データ無し	39	P9-71
誤り分析ページ ※1	EAP	1ページ	40	P9-72
ビット・ウインドウ ON/OFF	MSE	OFF	41	P9-73
ブロック・ウインドウ ON/OFF	MGE	OFF	42	P9-74
誤り分析トリガ ※1	EAT	OFF	43	P9-75
パターン同期トリガ位置	PSP		54	P9-88
ページ・パターン同期トリガ位置表示切替	PPD		55	P9-89

表 C-4 MEASUREMENT セクション初期値一覧表

機 能 項 目	ヘッダ	初 期 値	デバイス・ メッセージ詳細	
			項目 No.	ペー ジ
● MEASUREMENT セクション				
測定結果表示モード	DMS	ERROR RATIO	59	P9-94
途中結果表示機能	CUR	OFF	60	P9-95
測定モード	MOD	Repeat	61	P9-96
測定状態	MSR	測定停止	64	P9-99
自動同期機能	SYN	OFF	65	P9-100
自動同期スレッシュホールド	SYE	10^{-2}	66	P9-101
リアルタイム・測定時間表示切替	TIM	Period	67	P9-102
内部タイマ設定	RTM	現在時刻	68	P9-103
測定時間設定	PRD	1秒	69	P9-104

表 C-5 その他のセクション初期値一覧表

機 能 項 目	ヘッダ	初 期 値	デバイス・メッセージ詳細	
			項目 No.	ページ
● その他のセクション (正面パネル)				
プリンタ ON/OFF	PRN	OFF	84	P9-122
マニュアル・プリント	PSA	停止	85	P9-123
アラーム・モニタ (アラーム検出)	ALM	OFF	86	P9-124
アラーム・モニタ (誤り検出)	MON	OFF	87	P9-125
同期信号出力	SOP	1 / 32 CLOCK	88	P9-126
● その他のセクション (背面パネル GPIB)				
GPIB 2アドレス	GPA	0	89	P9-127
● その他のセクション (背面パネル・FUNCTION スイッチ)				
マーク率の AND ビットシフト数	SFT	1ビットシフト	90	P9-128
クロック断処理機能	CLS	Exclude	91	P9-129
同期外れ処理機能	SLS	Exclude	92	P9-130
エラー・パフォーマンス・データ・ スレッシュホールド選択機能	ETH	1.0E-3	93	P9-131
BURST 測定モード	BST	OFF	94	P9-132
測定途中データ算出機能	CAL	累積計算データ	95	P9-133
誤り検出モード	ETY	トータルエラー	96	P9-134
EI, EFI インターバル時間	EIT	1 msec	97	P9-135
データ印字フォーマット	FMT	標準フォーマット	98	P9-136
スレッシュホールド EI、EFI データ印字	THR	印字しない	99	P9-137
エラー・パフォーマンス・データ印字	EPF	印字しない	100	P9-138
中間データ印字	ITM	印字しない	101	P9-139
1秒データ印字	OSC	印字しない	102	P9-140
1秒データ印字スレッシュホールド	DOT	誤り > 0	103	P9-141
紙節約機能	PSV	OFF	104	P9-142
測定インタバル時間	ITV	100 msec	105	P9-143
メモリ・FD モード	FMD	1440 K / 720 K	13	P9-39
終端コード切り替え	TRM	LF+EOI	106	P9-144

付 録 D トラッキング項目一覧表

ここでは、MP1762Dのトラッキング項目について表 D-1に示します。

トラッキングとは、MP1762Dの設定状態を MP1761B/C に GPIB 経由で送信する機能で接続方法は、「3章 バス接続とアドレス設定」を参照してください。

また、トラッキング項目は測定パターンによってこととなります。

表 D-1 トラッキング項目一覧表

測 定 パ タ ー ン	ト ラ ッ キ ン グ 項 目
Alternate パターン	1 : LOGIC 2 : 測定パターン (Alternate) 3 : A/B 表示切替 4 : ページ設定 5 : パターンビット (DMA 転送、A/B とともに送出)
Data パターン	1 : LOGIC 2 : 測定パターン (Data) 3 : データ長 4 : ページ設定 5 : パターンビット (DMA 転送)
Zero subst. パターン	1 : LOGIC 2 : 測定パターン 3 : Zero subst. 段数 4 : Zero subst 長 5 : ページ設定
PRBS パターン	1 : LOGIC 2 : 測定パターン 3 : PRBS 段数 4 : PRBS マーク率 5 : ページ設定

