

【 非 公 開 】

TR - A - 0 0 1 1

A T R タキストスコープ制御ソフトウェア

— A T R T S P とグラフィックコンパイラ —

Software for ATR Tachistoscope

— ATRTSP and Graphic Compiler —

横澤 一彦

梅田 三千雄

Kazuhiko YOKOSAWA and Michio UMEMA

1 9 8 7 . 8 . 1 3

A T R 視聴覚機構研究所

## 目次

1.	はじめに	1
2.	ATRタキストスコープとその制御	1
2.1	ATRタキストスコープの特徴	1
2.2	ATRタキストスコープのハードウェア	1
2.3	ATRタキストスコープの制御ソフトウェア	3
3.	ATR T S P (ATR Tachistoscope Subroutine Package)	5
3.1	ATR T S P 概要	5
3.2	初期化と終了化サブルーチン	5
3.3	ディスプレイ語生成サブルーチン	6
3.4	ディスプレイ制御サブルーチン	10
3.5	アテンションサブルーチン	13
3.6	ディスプレイタイプ制御サブルーチン	15
3.7	文字入出力サブルーチン	16
3.8	ビデオ画面制御サブルーチン	18
3.9	スレーブサブルーチン	20
3.10	操作方法	24
3.11	エラーメッセージ	24
3.12	ATR T S P 診断用テストプログラム	26
4.	グラフィックコンパイラ	32
4.1	機能概要	32
4.2	ファイル操作	34
4.3	編集	36
4.4	転送実行	40
4.5	基本図形	41
4.6	属性	43
4.7	終了	44
4.8	割込みメニュー	44
5.	おわりに	45
	謝辞	45
	参考文献	45
	付録 テストプログラム	

## 1. はじめに

高度なマンマシンインターフェースを実現する為には、人間の行動特性を詳細に調べる必要がある。そのためには、人間の視覚、聴覚、記憶、思考などの諸特性を調べていかなければならない。特に、人間の視覚機構ではそれを解剖学的、生理学的に調べる事が難しいので、主に心理物理学的、認知心理学的に調べることになる。

視覚機構を心理物理学的、認知心理学的に調べる為には、被験者としての人間に何等かの視覚パターンを刺激として提示して、被験者から何等かの反応を採取できる装置が必要である。しかも、例えば文字の認識過程などを分析するには、人間が1文字を数十ミリ秒という早さで認識し終わってしまうので、高速の視覚刺激提示装置が必要である。現在のエレクトロニクス技術を駆使すれば、1ミリ秒単位で提示時間を正確に制御できる高速視覚刺激提示装置の開発は可能である。

そこで視覚研究室では、先に高速視覚刺激提示装置としてATRタキストスコープ<sup>1)</sup>を開発した。本報告では、この装置の制御ソフトウェアとその利用法について解説する。

## 2. ATRタキストスコープとその制御

### 2. 1 ATRタキストスコープの特徴

視覚研究室では視覚研究を進める為の基本ツールとして、高速視覚刺激提示装置ATRタキストスコープを開発した。この装置は、

- i) どんな複雑さの視覚刺激パターンでも短時間に提示できる。
- ii) 視覚刺激パターンの作成や変更が容易である。
- iii) 視覚刺激パターンの提示時間や提示時間間隔を高精度で制御できる。
- iv) 被験者の反応データを取込んで分析するのが容易である。

などの特徴を持ち、従来の刺激提示装置それぞれの長所を残し欠点を補った新しい実験装置である。特に、ii)は本報告で述べる制御ソフトウェアによって実現している。

### 2. 2 ATRタキストスコープのハードウェア

ATRタキストスコープのハードウェア構成図は図1に示す。ハードウェアの具体的特徴を挙げると以下の通りとなる。

a. ランダムスキャン方式とラスタースキャン方式(60フレーム/秒)を1台のディスプレイで実現し、更に高速ラスタースキャン方式(120フレーム/秒)のディスプレイも開発した。ランダムスキャン方式には、4つの描画速度、ラスタースキャン方式には標準モードと縮小モードがあり、高速ラスタースキャン方式とあわせて7つの提示方式を自由に選択することができる。従って、提示方式を選ぶことによって、どんな複雑さの視覚刺激パターンでも短時間に提示できる。尚、ランダムスキャン方式の為には16Kバイトのメモリ、ラスタースキャン方式の為には16画面分のビデオメモリを持つ。

b. ATRタキストスコープ本体のマイクロプロセッサ68000MPUでディスプ

# 高速視覚刺激提示装置本体

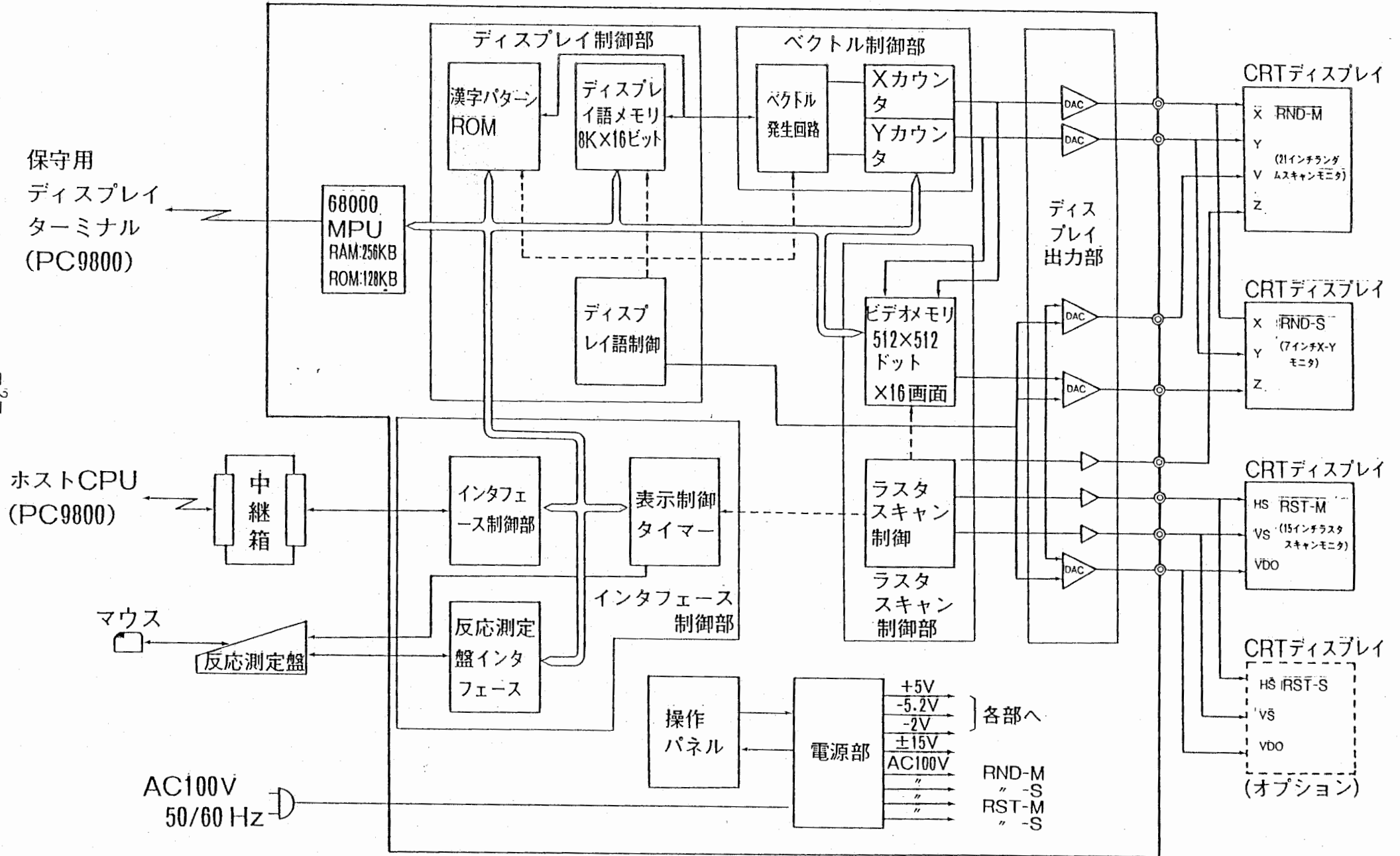


図1 ATRタキストスコープのハードウェア構成図

レイを制御し、短残光のディスプレイ（蛍光体P31）を用いている為、視覚刺激パターンの提示時間や提示時間間隔を高精度（1ミリ秒単位）で制御できる。

c. JIS第1水準の文字のベクトルデータをROMに内蔵し、各漢字のベクトルデータは64×64座標で作成されている。オプションとしてJIS第2水準の文字もROMで内蔵することができる。各文字のベクトルデータは自由に読み出し可能であり、文字の拡大縮小、変形が容易である。

d. 多機能反応測定盤により、反応時間や強制選択実験の選択肢などの反応データを取込める。それらの反応データはATRタキストスコープ本体を制御するホストCPU、PC9801に転送されるので、そのデータ処理が容易である。

e. ラスタースキャン方式のディスプレイに提示する画像はホストCPU、PC9801にイメージスキャナ等を接続し、入力、転送処理を行うことができる。

上述のような特徴を持つATRタキストスコープは、文字認識過程、画像記憶、運動視、アニメーション等の、色情報の研究を除く、ほとんどの視覚情報処理研究の汎用的な実験装置に用いることができる。

## 2. 3 ATRタキストスコープの制御ソフトウェア

ATRタキストスコープを制御するためには、29種類のコマンド、15種類のディスプレイ語及びそれらの組合せを、ホスト計算機PC9800から送らなければならない。これらのコマンドとディスプレイ語をなんらかの高級言語で扱う為には、基本操作のハンドラーが必要である。そこで、我々はまずホスト計算機PC9800上で動作するPC-Fortranで呼出し可能なハンドラー・サブルーチン群ATR T S P (ATR Tachistoscope Subroutine Package)を作成した。このATR T S Pを用いたフォートラン・プログラムによって、ATRタキストスコープを制御することができる。

ATRタキストスコープはこれまでのボックス型タキストスコープやプロジェクション型タキストスコープに代わるものである。従って、実験に用いる刺激をプログラムによって作成するのではなく、紙もしくはスライドを作成するのと同じ要領で実現できたならば、誰にでも使える実験装置になるであろう。そこで我々は、ホスト計算機PC9800のディスプレイ上で実験で用いる刺激図形をマウスを用いて描くと、その図形を自動的にATRタキストスコープを制御するフォートラン・プログラムへ変換するソフトウェア、グラフィックコンパイラを開発した。このグラフィックコンパイラを用いることによって、PC-FortranやATR T S Pを全く知らない実験者でも、ATRタキストスコープを実験に用いることができるようになった。

従って、ATRタキストスコープを利用する為の一連の手順は図2に示す通りである。次章以下では、ATR T S Pとグラフィックコンパイラの詳細を解説する。

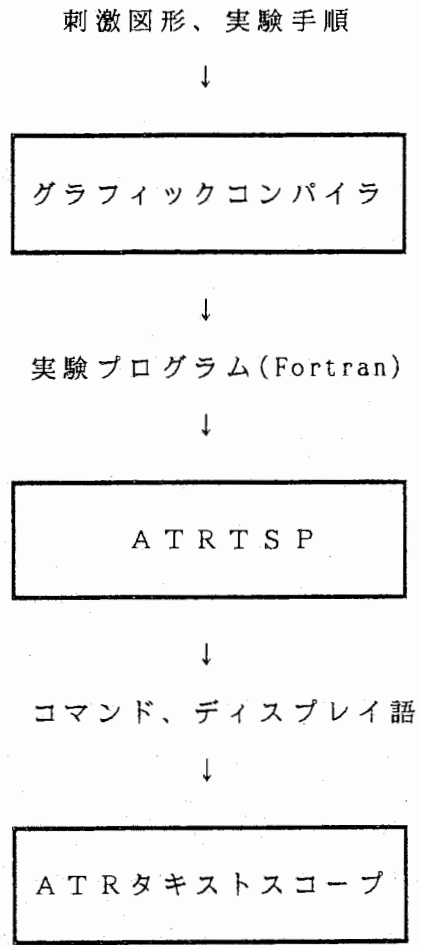


図2 ATRタキストスコープを用いたプログラムの開発手順

### 3. A T R T S P

#### 3. 1 A T R T S P 概要

A T R T S P (ATR Tachistoscope Subroutine Package)は、タキストスコープに図形や文字を表示したり、被験者との交信を制御するサブルーチンから構成される。A T R T S Pは、PC-FortranのC a l l文により使用でき、ユーザプログラムに、リンク時にライブラリーとして組み込まれ動作する。

A T R T S Pは、下記のサブルーチン群から構成される。

- 1) 初期化と終了化サブルーチン
- 2) ディスプレイ語生成サブルーチン
- 3) ディスプレイ制御サブルーチン
- 4) アテンションサブルーチン
- 5) ディスプレイタイプ制御サブルーチン
- 6) 文字入出力サブルーチン
- 7) ビデオ画面制御サブルーチン
- 8) スレーブサブルーチン

#### 3. 2 初期化と終了化サブルーチン

タキストスコープ及びA T R T S Pを初期化したり終了化したりするサブルーチンである。

##### 3. 2. 1 G O P E N (タキストスコープの初期化)

###### 1) 機能

タキストスコープを使用可能状態にし、エレメント管理等の為の制御テーブル (G E C T) を定義する。

###### 2) 呼出し形式

```
C A L L G O P E N ( I G E C T , I T S Z )
```

I G E C T G E C T用の配列名

この配列は次に示す名称で名前付コモン領域を確保しなければならない。

```
C O M M O N / G E C T / I G E C T ( I T S Z )
```

I T S Z 配列G E C Tのワード数 (エレメント数 \* 3ワード)

###### 3) 備考

タキストスコープを使用するプログラムでは、最初に1度C A L Lしなければ他のルーチンは使用できない。

G C L O S EをC A L Lする前に、再び本サブルーチンをC A L Lすると警告エラーが出される。

##### 3. 2. 2 G C L O S E (タキストスコープの終了化)

### 1) 機能

タキストスコープとユーザプログラムの結合を切り離す。

### 2) 呼出し形式

CALL GCLOSE

## 3. 2. 3 GDOA (GDOAの割当て)

### 1) 機能

ディスプレイ語列を格納するメインメモリの領域 (GDOA) を確保する。

### 2) 呼出し形式

CALL GDOA (IGDOA, IASZ)

IGDOA GDOA用の配列名

IASZ 配列IGDOAのワード数 (最大8192ワード)

### 3) 備考

このGDOAには、次節で記述されるディスプレイ御生成ルーチンによりディスプレイ語が格納される。

GDOAを再度CALLすると、そのGDOAは初期状態に戻る。

## 3. 3 ディスプレイ語生成サブルーチン

図形を表示する為に必要なディスプレイ語列を作り、GDOAに格納する。また、これに関連するエレメントについて定義する。

### 3. 3. 1 GEINIT (エレメントの初期化)

#### 1) 機能

エレメントを作り始める準備を行い、このサブルーチンの後で作られるディスプレイ語列からなるエレメントに名前及び特性を与える。

#### 2) 呼出し形式

CALL GEINIT (IGDOA, ID, INV, ITM, IWNK, INTEN, IT)

ID エレメントのID番号

(-32768+32767 但し0を含まない固有の番号でなければならない。)

INV ディスプレイ語列をタキストスコープに転送した時、表示するかどうかの指定

= 1 表示しない

≠ 1 表示する

ITM タイマーカウンットの有無の指定

= 1 カウントする (リアクションキーの入力待ちができる)

≠ 1 カウントしない (マウスによる入力ができる)

IWNK 表示したときの明滅指定



= 1 明滅する

≠ 1 明滅しない

INTEN 輝度指定 (0 - 15) 値が負の時輝度の指定なしとみなされ、それまで表示していた輝度がそのまま使われる。

IT 表示時間指定 (0 - 16383 msec)

< -1 通常モード

≥ -1 実験モード

### 3) 備考

必ず1度CALLする必要がある、GEINITで始まったエレメントが終わりになるのは、別のGEINIT又はGDISP, GXFERをCALLしたときである。

通常モードの時は、GDISP, GRDISPを用いて表示し、実験モードの時は、GXFER, GSTARTを用いて表示する。

IT = -1の場合は、1つ前のGEINITで作られたパターンと重ねあわされて提示される。

## 3. 3. 2 MOVABS (ビームの絶対位置指定)

### 1) 機能

ディスプレイ画面上のビームの絶対位置を決めるディスプレイ語列を作りGDOAに格納する。

### 2) 呼出し形式

CALL MOVEABS (IGDOA, IX, IY)

IX ビームを移動するX軸の絶対アドレス (-512 - +511)

IY ビームを移動するY軸の絶対アドレス (-512 - +511)

## 3. 3. 3 MOVREL (ビームの相対位置指定)

### 1) 機能

ディスプレイ画面上のビームの相対的に移動するディスプレイ語列を作りGDOAに格納する。

### 2) 呼出し形式

CALL MOVEREL (IGDOA, IX, IY)

IX 現在のビーム位置からX方向に移動する相対アドレス

IY 現在のビーム位置からY方向に移動する相対アドレス

## 3. 3. 4 DRWABS (絶対位置指定による直線の作成)

### 1) 機能

現在のビーム位置から指定した絶対位置までの直線をかきディスプレイ語列を作り、

GDOAに格納する。

2) 呼出し形式

CALL DRWABS (IGDOA, IX, IY)

IX 直線の終点のX軸の絶対アドレス

IY 直線の終点のY軸の絶対アドレス

3. 3. 5 DRWREL (相対位置指定による直線の作成)

1) 機能

現在のビーム位置から指定した相対位置までの直線をおくディスプレイ語列を作り、GDOAに格納する。

2) 呼出し形式

CALL DRWREL (IGDOA, IX, IY)

IX 直線の終点のX軸の相対アドレス

IY 直線の終点のY軸の相対アドレス

3. 3. 6 NDRW (相対位置指定による直線の作成の繰り返し)

1) 機能

現在のビーム位置の相対位置で、ある長さの線分を整数回繰り返して連続した直線をおくディスプレイ語列を作り、GDOAに格納する。

2) 呼出し形式

CALL NDRW (IGDOA, IX, IY, N)

IX 直線の移動量のX軸の相対アドレス

IY 直線の移動量のY軸の相対アドレス

N 移動量IX, IYの繰返し回数  $N \leq 0$  のときには  $N = 1$  とみなされる。

3. 3. 7 PNTABS (絶対位置指定による点の作成)

1) 機能

指定した絶対位置に点を表示するディスプレイ語列を作り、GDOAに格納する。

2) 呼出し形式

CALL PNTABS (IGDOA, IX, IY)

IX 点を表示するX軸の絶対アドレス

IY 点を表示するY軸の絶対アドレス

3. 3. 8 PNTREL (相対位置指定による点の作成)

1) 機能

指定した相対位置に点を表示するディスプレイ語列を作り、GDOAに格納する。

## 2) 呼出し形式

CALL PNTREL (IGDOA, IX, IY)

IX 点を表示するX軸の相対アドレス

IY 点を表示するY軸の相対アドレス

### 3. 3. 9 NPNT (相対位置指定による点の作成の繰返し)

#### 1) 機能

現在のビーム位置から相対的な移動量の指定により、複数個の点を表示するディスプレイ語列を作り、GDOAに格納する。

#### 2) 呼出し形式

CALL NPNT (IGDOA, IX, IY, N)

IX 点を表示するX軸の相対アドレス

IY 点を表示するY軸の相対アドレス

N 移動量IX, IYの繰返し回数  $N \leq 0$  のときには  $N = 1$  とみなされる。

### 3. 3. 10 GPINT (割込み発生制御語の作成)

#### 1) 機能

割込みを発生するディスプレイ語を作り、GDOAに格納する。

#### 2) 呼出し形式

CALL GPINT (IGDOA, ICODE)

ICODE インタラプトコード

### 3. 3. 11 GSPACE (スペースの確保)

#### 1) 機能

GDOA中にNO DATA語列を作る命令である。GRDISP(後述)でエレメントにディスプレイ語列を追加していくような場合には、この命令でディスプレイバッファ中にあらかじめNO DATA語列でできた領域を確保しておかなければならない。

#### 2) 呼出し形式

CALL GSPACE (IGDOA, N)

N 発生するNO DATA語の個数

### 3. 3. 12 GEDLY (提示間隔の指定)

#### 1) 機能

実験モードにおいて、続けて表示する幾つかのパターンの間の提示間隔を指定する。

#### 2) 呼出し形式

CALL GEDLY (IGDOA, IT)

### 3. 4 ディスプレイ制御サブルーチン

ディスプレイ語生成サブルーチンによって生成されたGDOA中のディスプレイ語列をディスプレイバッファに送り、CRT上に表示したり、既に表示しているエレメントの表示特性を変更する。

#### 3. 4. 1 G C L E A R (ディスプレイバッファの初期化)

##### 1) 機能

ディスプレイバッファの初期化を行う。

##### 2) 呼出し形式

CALL G C L E A R

#### 3. 4. 2 G D I S P (GDOAからディスプレイバッファへの転送と表示)

##### 1) 機能

GDOA中に用意されたディスプレイ語列をディスプレイバッファに送り、CRT面上に表示する。

##### 2) 呼出し形式

CALL G D I S P ( I G D O A )

I G D O A GDOA用の配列

#### 3. 4. 3 G R D I S P (転送したエレメントの再表示)

##### 1) 機能

既に表示したエレメントを変更して、再表示するとき使用する。この場合、変更後の大きさは変更前の大きさより大きくなってはならない。大きくなると予想されるときは、最初に表示するときGSPACE命令を使用して、みかけ上大きなエレメントとしておく必要がある。

##### 2) 呼出し形式

CALL G R D I S P ( I G D O A , I D )

I G D O A GDOA用配列

このGDOAにはエレメントが複数個存在してはならない。

I D 現在、表示しているエレメントのID番号

このエレメントにGDOAのエレメントがオーバーライトする。

#### 3. 4. 4 G X F E R (GDOAからディスプレイバッファへの転送)

##### 1) 機能

GDOA中に用意されたディスプレイ語列をディスプレイバッファの先頭から転送す

る。

2) 呼出し形式

CALL GXFER (IGDOA)

3. 4. 5 GSTART (ディスプレイバッファの内容の表示開始)

1) 機能

ディスプレイバッファに格納されているディスプレイ語列をCRT面上に表示を開始する。

2) 呼出し形式

CALL GSTART

3. 4. 6 GALSCR (表示中のエレメントのコントロールレジスタの変更)

1) 機能

表示中のエレメントの特性(コントロールレジスタ)を変更する。

2) 呼出し形式

CALL GALSCR (ID, INV, ITM, IWNK)

ID 特性を変更するエレメントのID番号

INV 表示の有無(値が負の場合は変更なし)

ITM タイマーの有無(値が負の場合は変更なし)

IWNK 明滅の有無(値が負の場合は変更なし)

3. 4. 7 GALTME (表示中のエレメントの提示時間の変更)

1) 機能

表示中のエレメントの提示時間のディスプレイ語を変更する。

2) 呼出し形式

CALL GALTME (ID, IT)

ID 変更するエレメントのID番号

IT 変更する提示時間

3. 4. 8 GALINT (表示中のエレメントの輝度の変更)

1) 機能

表示中のエレメントの輝度を変更する。

2) 呼出し形式

CALL GALINT (ID, INTEN)

ID 変更するエレメントのID番号

INTEN 変更する輝度

### 3. 4. 9 GMOVE (表示中のエレメントの移動)

#### 1) 機能

表示中のエレメントを指定した位置に移動する。

#### 2) 呼出し形式

CALL GMOVE (ID, IX, IY)

ID 変更するエレメントのID番号

IX 変更するエレメントの始点のX軸の絶対アドレス

IY 変更するエレメントの始点のY軸の絶対アドレス

### 3. 4. 10 GALDLY (表示中のエレメント間の提示間隔の変更)

#### 1) 機能

ディスプレイバッファに格納されている提示間隔を変更する。

#### 2) 呼出し形式

CALL GALDLY (ID, IT)

ID 変更したい提示間隔を指定したGEDLYの直後のGEINITで指定したID番号

IT 新しく設定する提示間隔 (0-16383msec)

### 3. 4. 11 GSTOP (ディスプレイの表示の停止)

#### 1) 機能

ディスプレイの表示を停止する。

#### 2) 呼出し形式

CALL GSTOP

### 3. 4. 12 GDISPとGXFER, GSTARTについて

GDISPは、通常モードでパターンを表示するときを使用し、現在表示されているパターンに更に別のパターンを追加して表示することもできる。

GXFERとGSTARTは、実験モードにおいてパターンを表示するとき使用する。このモードにおいては、ディスプレイバッファには、数個のパターンを格納し、それを順次表示する。GDOAに格納されているパターンデータは、GXFERによりディスプレイバッファの先頭から格納され、その後GSTARTによりその先頭から表示される。

このように、GXFERでは必ずディスプレイバッファの先頭から格納され、GSTARTにより必ず先頭から表示が開始される。

GDISPの場合には、複数個のGDOAからディスプレイバッファへ転送が可能で

あるが、GXFERの場合には1個のGDOAのみから転送が可能である。

### 3. 5 アテンションサブルーチン

#### 3. 5. 1 GALARM (ブザーのセット)

##### 1) 機能

ディスプレイ装置のブザーを鳴らす。

##### 2) 呼出し形式

CALL GALARM

#### 3. 5. 2 GMRK (リアクションキーのマスク)

##### 1) 機能

リアクションキーL3-L1、R1-R3をマスクする。

##### 2) 呼出し形式

CALL GMRK (L3, L2, L1, R1, R2, R3)

L3, L2, L1, R1, R2, R3 1ならマスク、0ならマスク解除

#### 3. 5. 3 GETFK (ファンクションキー割込みの認識)

##### 1) 機能

押されたファンクションキー番号を知る。

##### 2) 呼出し形式

CALL GETFK (IFK, IWAIT)

IFK 押されたファンクションキー番号 (0-11)

IWAIT = 1 ファンクションキーが押されるまで待つ。

≠ 1 IFK = -1として待たずにすぐ戻る。

#### 3. 5. 4 GETRK (リアクションキー割込みの認識)

##### 1) 機能

押されたリアクションキー番号を知る。

##### 2) 呼出し形式

CALL GETRK (IRK, ITIME)

IRK 押されたリアクションキー番号 (0-5)

ITIME リアクションキーが押されたときまでの経過時間

= 1 リアクションキーが押されるまで待つ。

≠ 1 IRK = -1として待たずにすぐ戻る。

#### 3. 5. 5 GMOUXY (マウス割込みの認識)

1) 機能

マウスの X, Y 座標と押されたボタン情報を得る。

2) 呼出し形式

CALL GMOUXY (ILR, IX, IY)

ILR = 1 左ボタン

= 2 右ボタン

IX X座標

IY Y座標

3. 5. 6 GETINT (プログラム割込みの認識)

1) 機能

GPINTにより発生した割込みコードを知る。

2) 呼出し形式

CALL GETINT (ICODE)

ICODE インタラプトコード

3) 備考

ICODEで指定された割込みが発生するまで待つ。もし割込みが無ければこのルーチンは終了しない。

3. 5. 7 GSTMES (メッセージデータの書込み)

1) 機能

反応測定盤に表示するメッセージデータを書込む。

2) 呼出し形式

CALL GSTMES (NO, TEXT)

NO メッセージ番号 (0-255)

TEXT 書込むメッセージ (8文字固定)

3. 5. 8 GDSPMS (メッセージデータの表示)

1) 機能

反応測定盤にメッセージを表示する。

2) 呼出し形式

CALL GDSPMS (IFLG, INO)

IFLG ≠ 0 ROM登録済みメッセージ

= 0 GSTMESによって追加されたメッセージ

INO メッセージ番号 (0-225)

ROM登録済みメッセージ



番号	メッセージ
0	XXXX MS
1	CLK 50ns
2	CLK 100ns
3	CLK 150ns
4	CLK 200ns
5	INT LV X
6	60FRAME
7	120FRAME
8	VDOSEL 2
9	TIME ON
10	TIME OFF
11	MOUSE ON
12	MOUSE OFF
13	60FRM 1/2
14	PUSH KEY

### 3. 6 ディスプレイタイプ制御サブルーチン

#### 3. 6. 1 GREDTY (表示状態の読み込み)

##### 1) 機能

現在の表示状態を知る。

##### 2) 呼出し形式

CALL GREDTY (ITYPE)

ITYPE ディスプレイ表示状態 (1-7)

- 1 : 50ns
- 2 : 100ns
- 3 : 150ns
- 4 : 200ns
- 5 : 60フレーム 1/2
- 6 : 60フレーム
- 7 : 120フレーム

#### 3. 6. 2 GSLDTY (表示状態の指定)

##### 1) 機能

ディスプレイの表示状態を指定する。

##### 2) 呼出し形式

CALL GSLDTY (ITYPE)

ITYPE ディスプレイ表示状態 (1-7)

- 1 : 50 n s
- 2 : 100 n s
- 3 : 150 n s
- 4 : 200 n s
- 5 : 60 フレーム 1 / 2
- 6 : 60 フレーム
- 7 : 120 フレーム

### 3. 6. 3 GDMODE (表示状態の選択)

#### 1) 機能

表示ディスプレイタイプのメニューを表示し、そのメニュー番号の選択を行う。

#### 2) 呼出し形式

CALL GDMODE (ITYPE)

ITYPE ディスプレイ表示状態 (1-7)

- 1. RANDOM 50 n s
- 2. RANDOM 100 n s
- 3. RANDOM 150 n s
- 4. RANDOM 200 n s
- 5. RASTER 60 FRAME 1 / 2
- 6. RASTER 60 FRAME
- 7. RASTER 120 FRAME

### 3. 7 文字入出力サブルーチン

#### 3. 7. 1 TINPUT (マウスボタンの読み込み)

##### 1) 機能

押されたマウスボタンの読み込み

##### 2) 呼出し形式

CALL TINPUT (ICHAR)

ICHAR = 1 左ボタン  
= 2 右ボタン

#### 3. 7. 2 TOUTPUT (文字出力)

##### 1) 機能

1文字を表示するディスプレイ語列を作り、GDOAに格納する。

2) 呼出し形式

CALL TOUTPT (IGDOA, ICHAR)

ICHAR 出力する文字

3) 備考

ICHARで示されるワードには、右バイトにASCIIコードが格納されている必要がある。

文字は、21X31ラスタユニットで表示され、1文字表示後のビームはX軸方向に31ラスタユニットだけ移動する。

3.7.2 ANSTR (文字列の出力)

1) 機能

配列に格納されている文字列を表示するディスプレイ語列を作り、GDOAに格納する。

2) 呼出し形式

CALL ANSTR (IGDOA, INCHR, IARRY)

INCHR 出力する文字

IARRY 文字列が格納されている配列名

3) 備考

IARRYの配列の各ワードには、左右の順でASCIIコードが格納されている必要がある。

3.7.3 GKANJI (漢字パターンの表示)

1) 機能

漢字パターンを表示する。

2) 呼出し形式

CALL GKANJI (IGDOA, N, KANJI)

N 漢字文字数

KANJI 漢字文字列

3.7.4 GKANJ2 (変形漢字パターンの表示)

1) 機能

幅、高さ、角度を指定して漢字パターンを表示する。

2) 呼出し形式

CALL GKANJ2 (IGDOA, N, KANJI, IH, IW, RR, ISP)

N 漢字文字数

KANJI 漢字文字列

I H 文字高さ  
I W 文字幅  
R R 回転角  
I S P 漢字と漢字のスペース

### 3. 8 ビデオ画面制御サブルーチン

#### 3. 8. 1 GWV MEM (ビデオメモリへの書込み)

##### 1) 機能

指定されたファイルから二値データを読みビデオメモリに書込む。

##### 2) 呼出し形式

CALL GWV MEM (NO, N, FNAME, ISREC, ISBYT)

NO 書込むビデオメモリ番号 (0 - 15)

N ファイル名の文字数  
デバイス、ファイル名、ファイルタイプを含めて14文字以内

FNAME ファイル名

ISREC 読み込む最初のレコード番号 (1 - 96)

ISBYT 読み込む最初のコラム番号  
2バイト単位で指定する

#### 3. 8. 2 GCT LRS (ビデオ画面制御ディスプレイ語の生成)

##### 1) 機能

ビデオ画面を表示する為の制御ディスプレイ語を生成する。

##### 2) 呼出し形式

CALL GCT LRS (IGDOA, NO, INV, ITM, IWNK, INTEN, IT)

NO 表示するビデオメモリ番号 (0 - 15)

INV ディスプレイ語列をタキストスコープに転送した時、表示するかどうか  
の指定

= 1 表示しない

≠ 1 表示する

ITM タイマーカウントの有無の指定

= 1 カウントする (リアクションキーの入力待ちができる)

≠ 1 カウントしない (マウスによる入力ができる)

IWNK 表示したときの明滅指定

= 1 明滅する

≠ 1 明滅しない

INTEN 輝度指定 (0 - 15) 値が負の時輝度の指定なしとみなされ、それま

で表示していた輝度がそのまま使われる。

INT            表示フレーム数又は時間指定  
                  $\geq 0$    フレーム数  
                  $< 0$    時間 (m s e c)

### 3. 8. 3   GRSTAT (ビデオメモリの表示開始)

#### 1) 機能

ビデオメモリ表示制御データをスタートさせる。

#### 2) 呼出し形式

CALL GRSTAT (IGDOA, INT)

INT   指定フレーム数表示終了毎にホスト計算機に割込みを発生するか否かの指定  
      = 0   割込み禁止  
      = 1   割込みあり

### 3. 8. 4   GRVMEM (ビデオメモリからのデータ読み込み)

#### 1) 機能

指定したビデオメモリからデータを読み込む。

#### 2) 呼出し形式

CALL GRVMEM (NO, IADDR, N, MEMORY)

NO            ビデオメモリ番号  
IADDR        ビデオメモリアドレス  
N             読み込むデータ数  
MEMORY      メモリデータ

### 3. 8. 5   GVCLR (ビデオメモリ画面の初期化)

#### 1) 機能

指定したビデオメモリ画面の初期化を行う。

#### 2) 呼出し形式

CALL GVCLR (NO)

NO   ビデオ画面番号 (0 - 15)

### 3. 8. 6   GRDLAY (ビデオ画面提示時間間隔の指定)

#### 1) 機能

ビデオ画面表示後、次の画面を表示するまでの提示時間間隔を指定する。

#### 2) 呼出し形式

CALL GRDLAY (IGDOA, IDTIME)

I D T I M E 提示時間間隔フレーム数又は時間

≥ 0 フレーム数

< 0 時間 ( m s e c )

### 3. 8. 7 G R C O N T (連続表示用ディスプレイ語の生成、実行)

#### 1) 機能

連続表示用ディスプレイ語を生成し実行する。

#### 2) 呼出し形式

C A L L G R C O N T ( I G D O A , I N T E N , I T )

I N T E N 輝度 ( 0 - 1 5 )

I T 提示フレーム数又は時間

≥ 0 フレーム数

< 0 時間 ( m s e c )

### 3. 8. 8 G R S T O P (連続表示の停止)

#### 1) 機能

二値パターン連続表示を停止させる。

#### 2) 呼出し形式

C A L L G R S T O P

### 3. 8. 9 G T R I N T (連続表示終了時の割込みの認識)

#### 1) 機能

二値パターン連続表示動作で1画面の指定提示フレーム数の終了時の割込みを得る。

#### 2) 呼出し形式

C A L L G T R I N T ( N O )

N O 指定提示フレーム数が終了したビデオ画面番号

## 3. 9 スレーブサブルーチン

A T R T S P で用いられる主なスレーブサブルーチンに関して説明する。

### 3. 9. 1 G D D F

#### 1) 機能

A T R タキストスコープを登録し、制御テーブル G D C T を定義する。A T R タキストスコープがオフラインの場合、エラーリターンする。

#### 2) 呼出し形式

C A L L G D D F ( I E R R )

I E R R エラーコード通知領域

### 3. 9. 2 GDRV

#### 1) 機能

ATRタキストスコープをホスト計算機から開放する。

#### 2) 呼出し形式

CALL GDRV (IERR)

IERR エラーコード通知領域

### 3. 9. 3 POSSUB

#### 1) 機能

DRWステップを生成し、GDOAに格納する。

#### 2) 呼出し形式

CALL POSSUB (IGDOA, IX, IY, IDETL, ICNT, IERR)

IGDOA GDOA領域の配列名

IX 移動すべき相対X座標

IY 移動すべき相対Y座標

IDETL 線種 -1 ブランクで移動

0 移動終点到点を描く

1 実線で移動

ICNT 繰返し回数

IERR 本スレーブサブルーチン呼び出したサブルーチンの番号 (エラーコードに用いる。)

### 3. 9. 4 STRGDO

#### 1) 機能

ディスプレイ語をGDOAに格納する。

#### 2) 呼出し形式

CALL STRGDO (IGDOA, IDATA, IWN, INO)

IGDOA GDOA領域の配列名

IDATA 格納すべきディスプレイ語の入っている領域

IWN 格納すべき語数

INO 本スレーブサブルーチン呼び出したサブルーチンの番号 (エラーコードに用いる。)

#### 3) 備考

GDOAに空き領域があるかどうかチェックする。

### 3. 9. 5 GDCMD

ホスト計算機PC9801とATRタキストスコープとの間の入出力処理を行う。コマンド番号 (ICMD) により、種々の入出力処理を行うことができる。

ICMD	コマンド	IDATA (1)	IDATA (2)
1	WDAT	書込みデータ	
2	RDAT	読取りデータ	
3	RAD	アドレス	
4	RPOX, RPOY	X座標	Y座標
5	RID	ID番号	
6	RITC	割込みコード	
7	RTIM	タイムカウント	
8	SAD	アドレス	
9	SADS	アドレス	
10	SALM		
11	SFKB	キースイッチマスク番号	
12	SRAK	メッセージデータ	
13	RKEY		
14	RKEY	ファンクションキーコード	
15	RKEY	リアクションキーコード	
16	SEL DSP CLOCK	クロック番号	
17	READ DSP CLOCK	表示クロック番号	
18	SET VDO SEL	表示モード番号	
19	SET VDO SEL	ビデオメモリ直接表示情報	
20	SEL KANJI CODE	JIS漢字コード	
21	READ KANJI CODE	漢字コード	
22	DISP MESS DATA	メッセージ番号	
23	SET MESS NO.	メッセージ番号	
24	MESS DATA	メッセージデータ	
25	GET MOUSE POS.	座標データ	
26	VIDEO CTRL DATA	ビデオコントロール情報	
27	READ VIDEO MEM	ビデオデータ	
28	WRITE VIDEO MEM	ビデオデータ	
29	MOUSE ST. DATA		

#### 2) 呼出し形式

CALL GDCMD (ICMD, IDATA, IERR)

ICMD      コマンド番号



I D A T A 入出力データ領域

I E R R エラーコード通知領域

### 3. 9. 6 G R D M

#### 1) 機能

ディスプレイ語を読み出す。

#### 2) 呼出し形式

C A L L G R D M ( I A R E A , N , I E R R )

I A R E A 読み出したデータを格納する配列名

N 読み出すデータのワード数

I E R R エラーコード通知領域

### 3. 9. 7 G W R M

#### 1) 機能

ディスプレイ語を格納する。

#### 2) 呼出し形式

C A L L G W R M ( I A R E A , N , I E R R )

I A R E A 書込むデータを格納する配列名

N 書込むデータのワード数

I E R R エラーコード通知領域

### 3. 9. 8 G E S T

#### 1) 機能

配列 G E C T にエレメント管理データを格納する。

#### 2) 呼出し形式

C A L L G E S T ( I D A T A , I E R R )

I D A T A 3ワードの配列で、ディスプレイバッファに格納したエレメントの I D 番号、ディスプレイバッファに置ける先頭アドレス、その大きさが格納される。

I E R R エラーコード通知領域

### 3. 9. 9 G E L D

#### 1) 機能

配列 G E C T からエレメント管理データを読み出す。

#### 2) 呼出し形式

C A L L G E L D ( I D A T A , I E R R )

I D A T A 3ワードの配列で、ディスプレイバッファに格納するエレメントの I D  
番号だけを指定する。

I E R R エラーコード通知領域

### 3. 9. 10 G D E R R

#### 1) 機能

以下のようなエラーメッセージを表示する。

#### 2) 呼出し形式

C A L L G D E R R ( I N O , I E R R )

I N O エラーを起こしたサブルーチン番号

I E R R エラーコード

### 3. 10 操作方法

A T R T S P は、ライブラリ A T R T S P . L I B に保存されており、リンク時に  
以下のように呼出せばよい。

```
A > F O R T R A N X X X
```

```
A > L I N K X X X , , , A T R T S P
```

ここで、X X X はメインプログラム名である。

### 3. 11 エラーメッセージ

#### 3. 11. 1 表示形式

表示は PC9800 のディスプレイに以下の形式で行われる。

```
A T R T S P r r r E R R O R e e e
```

ここで、r r r と e e e は以下の通りである。

r r r ルーチン番号。これは、3章の節番号により以下の要領で決定されたもの  
である。

3. 2. 1 G O P E N → 2 0 1

3. 3. 2 M O V A B S → 3 0 2

3. 3. 1 2 G E D L Y → 3 1 2

e e e エラーコード

#### 3. 11. 2 エラーコード

8 0 0 画面の X 軸方向がオーバー

8 0 1 画面の Y 軸方向がオーバー

- 8 0 2 繰り返し図形の繰り返し回数エラー
- 8 0 3 G D O A 又は G E C T サイズ指定の誤り
- 8 0 4 I D 番号の指定が誤り
- 8 0 5 表示時間の指定が誤り
- 8 0 6 輝度の指定が誤り
- 8 0 7 インタラプトコードの指定が誤り
- 8 0 8 ファンクションキー番号の指定が誤り
- 8 0 9 リアクションキー番号の指定が誤り
- 8 1 0 G D O A 領域が不足する
- 8 1 1 ディスプレイバッファのサイズが不足する
- 8 1 2 ディスプレイバッファ上にエレメントが1つも無い
- 8 1 3 エレメントサイズがオーバー
- 8 1 4 モードエラー
- 8 1 5 ディスプレイバッファ上に変更できるエレメントがない
- 8 1 6 G M O V E できないエレメントである
- 8 1 7 P O S S U B ルーチンのパラメータエラー
- 8 1 8 ディスプレイ表示タイプの異常
- 8 1 9 メッセージ番号の異常
- 8 2 0 ビデオ画面番号の異常
- 8 2 1 ビデオデータファイルオープンエラー
- 8 2 2 ビデオデータファイルを読む開始カラムの異常
- 8 2 3 インタラプトコードが指定されたものと一致しない
  
- 1 3 1 A T R タキストスコープがオフライン
- 1 3 2 A T R タキストスコープが応答時間アウト
- 1 3 3 ディスプレイバッファのパリティエラー
- 1 3 4 A T R タキストスコープコマンドエラー
- 1 3 5 G E L D ルーチンパラメータエラー ( I D 番号エラー)
- 1 3 6 ディスプレイバッファ上に該当エレメントがない
- 1 3 7 G E S T ルーチンパラメータエラー ( I D 番号エラー)
- 1 3 8 ディスプレイバッファ上に同一エレメントがある (重複する)
  - ・ I D E F エラー
  - ・ I R M V エラー

### 3. 1 2 A T R T S P 診断用テストプログラム

#### 3. 1 2. 1 診断テスト概要

A T R T S P は、以下の 8 個のテストプログラムによって、種々の診断テストを行うことができる。テストが合格かどうかは、A T R タキストスコープの画面の目視の他に、メモリの内容をダンプすることによっても可能である。ダンプの手順は以下の通りである。

- ① キーボードから G E T M E M をタイプイン
- ② 終了後、リターンキー押下
- ③ キーボードから D U M P G D M E M をタイプイン

テストプログラムの実行途中でダンプを行うときには、CTRL-C によりテストプログラムを強制終了させる必要がある。

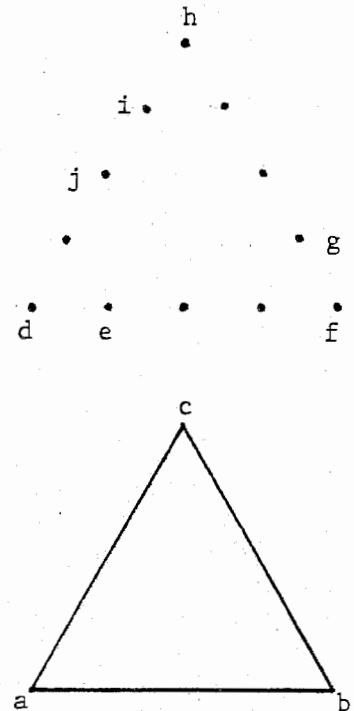
尚、テストプログラムは実験用のアプリケーションプログラムを開発する上で参考になるので、それらのソースプログラムを付録とする。

#### 3. 1 2. 2 T E S T 1

##### (1) 仕様

1 辺 2 0 0 ドットの实線及び点線の正三角形を以下のように表示する。

- ① M O V A B S で a 点にビームを位置させる。
- ② N D R W を 4 回繰返して a b 線を表示させる。
- ③ N D R W を 2 回繰返して b c 線を表示させる。
- ④ D R W R E L で c a 線を表示させる。
- ⑤ M O V R E L で d 点に位置させる。
- ⑥ N P N T を 4 回繰返して e から f の点を表示させる。
- ⑦ N P N T を 4 回繰返して g から h の点を表示させる。
- ⑧ P N T R E L で i 点を表示させる。
- ⑨ N P N T を 3 回繰返して j から d の点を表示させる。
- ⑩ G M O V E により a 点を初期位置 (-500, -500) から (-100, -100) へ移動する。この移動により、2 個の三角形も移動する。
- ⑪ G M O V E により a 点を (-100, -100) から (-300, -500) へ移動する。同じく 2 個の三角形も移動する。



##### (2) 操作法

- a. キーボードから T E S T 1 をタイプイン
- b. 表示ディスプレイタイプ 1 - 7 を選択する。① - ⑨ が実行される。
- c. マウスの左右どちらかのボタンを押すと ⑩ が実行される。
- d. マウスの左右どちらかのボタンを押すと ⑪ が実行される。
- e. マウスの左右どちらかのボタンを押すとプログラムは終了する。

尚、上記 b, c, d の各々について、ATR タキスト スコープメモリ をダンプし、ディスプレイ 語列を確認する。

### 3. 1 2. 3 TEST 2

#### (1) 仕様

ファンクションキーを押すことによって、3-6 角形を個別に、あるいは重ねて表示する。各ファンクションキーの役割は以下の通りである。

FKY-0, 1, 9-10 何もしない。

FKY-2 4 角形が既に表示されているときのみ有効で、GRDISP により 4 角形を 5 角形に変える。

FKY-3 3 角形を今まで表示中の図形に重ねて表示する。

FKY-4 4 角形を今まで表示中の図形に重ねて表示する。

FKY-5 5 角形を今まで表示中の図形に重ねて表示する。

FKY-6 6 角形を今まで表示中の図形に重ねて表示する。

FKY-7 3-6 角形を以下のようなパラメータで重ねて表示する。

	ID	INV	IWNK	INTEN
3 角形	1	1	0	0
4 角形	2	0	1	5
5 角形	3	0	0	10
6 角形	4	0	1	15

FKY-8 GALSCR, GALINT により、以下のようにパラメータを変更して表示する。

	ID	INV	IWNK	INTEN
3 角形	1	0	0	15
4 角形	2	1	1	0
5 角形	3	0	-1	10
6 角形	4	-1	0	5

FKY-11 GSTOP により表示を終了する。

#### (2) 操作法

- キーボードから TEST 2 をタイプイン
- 表示ディスプレイタイプ 1-7 を選択する。
- FKY-3 を押し、3 角形を表示することを確認する。
- FKY-4 を押し、4 角形を 3 角形に重ねて表示することを確認する。
- FKY-5 を押し、更に重ねて 5 角形を表示することを確認する。
- FKY-6 を押し、更に重ねて 6 角形を表示することを確認する。
- FKY-11 を押し、表示が消え、プログラムが終了することを確認する。

- h. 上記 c - f の順序を変えて実行してみる。
- i. TEST 2 を起動し、FKY - 4, FKY - 2 の順に押して、4 角形が 5 角形に変わることを確認する。
- j. FKY - 7 を押して、5 角形が消えた後で 3 - 6 角形が重ねて表示されることを確認する (3 角形は見えない。)
- k. FKY - 8 を押し、3 角形が表示され 4 角形が消えるなどの所定の変更がなされることを確認する (表示ディスプレイがラスタの場合にはこの処理はできない。)

### 3. 1 2. 4 TEST 3

#### (1) 仕様

実験モードの表示を行う。

- ① 3 角形を 3 秒間表示する。
- ② 3 角形が消えた後、3 秒間の間隔をおく。
- ③ 4 角形を 4 秒間表示する。
- ④ 4 角形が消えた後、4 秒間の間隔をおく。
- ⑤ 5 角形を 5 秒間表示する。
- ⑥ 5 角形が消えた後、5 秒間の間隔をおく。
- ⑦ 6 角形を 6 秒間表示する。
- ⑧ インタラプトコード (= 7) を実行し、表示を停止する。
- ⑨ 各図形の提示時間及び提示時間間隔を、GAL TME, GAL DLY を用いて以下のように変更する。

提示時間 (秒)      提示時間間隔 (秒)

3 角形	6	6
4 角形	5	5
5 角形	4	4
6 角形	3	0

- ⑩ 1 つのエレメント (ID = 1) に 3 - 6 角形を組入れ、その各々を 3 - 6 秒ずつ表示する。その後 7 秒の提示時間間隔においてインタラプトコードを実行する。

3 - 6 角形を各々 1 つずつのエレメントとし、同時に重ねて 16.383 秒表示する。

#### (2) 操作法

- a. キーボードから TEST 3 とタイプイン
- b. ディスプレイタイプ 1 - 7 を選択する。
- c. いずれかのファンクションキーを押すと表示を開始する。

### 3. 1 2. 5 TEST 4

#### (1) 仕様

反応測定盤にROM登録済みメッセージと追加登録したメッセージを表示する。

① 0-14のROM登録済みメッセージを表示する。

② 'TEST4', 'PC-9800', 'ATR', 'タキスト', 'スコプ', 'END'のメッセージを0-5に登録する。

③ ②で追加登録したメッセージを表示する。

(2) 操作法

a. キーボードからTEST4とタイプイン

b. 反応測定盤に'END'が表示されるまでいずれかのファンクションキーを押す。

c. 'END'が表示された後、いずれかのファンクションキーを押すとプログラムは終了する。

3. 12. 6 TEST5

(1) 仕様

ビデオメモリに図形を登録し、提示時間、提示時間間隔を与えて表示する。

① ビデオメモリ0、1番に図形を登録する。

② 各図形の提示時間、提示時間間隔を与える。

③ ビデオメモリ0番は明滅なし、1番は明滅有りて表示する。

(2) 操作法

a. キーボードからTEST5をタイプイン

b. 表示ディスプレイタイプ5-7を選択する。

c. ビデオメモリにファイルTMP.DATの図形を登録するかどうかを指定する。

d. ビデオメモリ0、1番の図形について、提示時間、提示時間間隔を入力する。正值はフレーム数を表わし、負値は時間(ミリ秒)を表わす。

e. 指定した時間だけ表示するとプログラムは終了する。

3. 12. 7 TEST6

(1) 仕様

輝度、明滅、大きさ、回転角などのパラメータを与えて、漢字パターンを表示する。

ANSTRも使用して文字パターンを表示する。

① エレメント1に次の文字をGKANJ1を使って登録する。

'この高速視覚刺激提示装置は、'

'漢字、ひらがな、カタカナ、'

'ALPHABET'

輝度8

② エレメント2に次の文字をGKANJ2を使って登録する。

'大' 幅150、高さ100、角度0、スペース10、輝度8

③ エレメント3に次の文字をGKANJ2を使って登録する。

'小' 幅32、高さ32、角度0、スペース10、輝度8

④エレメント4に次の文字をG K A N J Iを使って登録する。

’ 明 ’ 輝度15

⑤エレメント5に次の文字をG K A N J Iを使って登録する。

’ 暗 ’ 輝度5

⑥エレメント6に次の文字をG K A N J 2を使って登録する。

’ 回転 ’ 幅64、高さ64、角度30、スペース10、輝度8

⑦エレメント7に次の文字をG K A N J Iを使って登録する。

’ 明滅 ’ 輝度8、明滅有り

⑧エレメント8に次の文字をG K A N J Iを使って登録する。

’ 提示時間などを自由に設定して ’

’ 提示できます。 ’ 輝度8

⑨エレメント9に次の文字をA N S T Rを使って登録する。

’ W E L O V E A T R ’ 輝度8、明滅有り

⑩エレメント10に次の文字をA N S T Rを使って登録する。

’ 1987 ’ 輝度8

## (2) 操作法

- キーボードからT E S T 6をタイプインすると①-⑩の文字が重ねて表示される。
- マウスのボタンのどちらかを押すとプログラムは終了する。

### 3. 12. 8 T E S T 7

#### (1) 仕様

0-15のビデオメモリを連続して表示する。

- ビデオメモリ0-15番に図形を登録する。
- 輝度10、フレーム数10で連続表示する。

#### (2) 操作法

- キーボードからT E S T 7をタイプイン
- 表示ディスプレイタイプ5-7を選択する。
- ビデオメモリにファイルT M P . D A Tの図形を登録するかどうかを指定する。
- いずれかのファンクションキーを押すとプログラムは終了する。

### 3. 12. 9 T E S T 8

#### (1) 仕様

エラーメッセージを印字する為のプログラムである。

#### (2) 操作法

- キーボードよりT E S T 8をタイプインする。
- エラーメッセージ番号が次の順で出力されるのを確認する。



- 8 1 8 ディスプレイ表示タイプの異常
- 8 1 9 メッセージ番号の異常
- 8 0 3 G D O A または G E C T サイズ指定の誤り
- 8 0 4 I D 番号の指定が誤り
- 8 0 5 表示時間の指定が誤り
- 8 0 6 輝度の指定が誤り
- 8 0 0 画面の X 方向がオーバー
- 8 0 1 画面の Y 方向がオーバー
- 8 0 2 繰返し図形の繰返し回数
- 8 0 7 インタラプトコードの指定が誤り
- 8 1 2 ディスプレイバッファ上にエレメントが1つもない
- 8 1 5 ディスプレイバッファ上に変更できるエレメントがない

#### 4. グラフィックコンパイラ

##### 4.1 機能概要

本プログラムは、視覚刺激提示装置を制御する計算機PC9800上で動作し、以下の3つの機能を有する。

1. 視覚刺激提示装置を提示する視覚パターン（線図形や文字など）をマウスなどにより作成する。
2. 作成された視覚パターンの保存、既存パターンの読み込みなどファイル操作を行う。
3. 作成された視覚パターン系列を、視覚刺激提示装置を制御するFortranプログラムに変換する。

グラフィックコンパイラの起動方法は、日本語版が

A>ATR J

英語版が

A>ATRE

とタイプインすればよい。

グラフィックコンパイラの初期画面は図3に示す通りである。

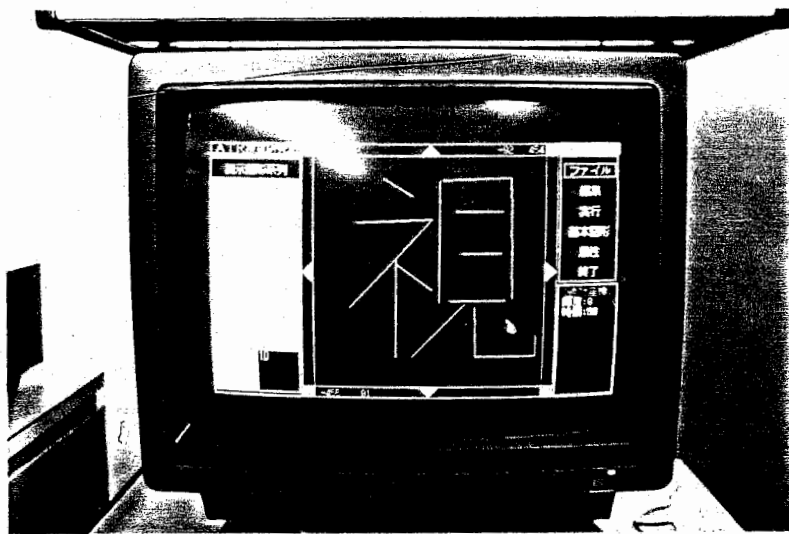


図3 グラフィックコンパイラの初期画面

グラフィックコンパイラ画面は、図4のように、大きく4つの部分に分けることができる。それぞれは以下のような内容である。

- ①提示刺激系列、②編集画面、③メニュー、④属性情報

①は提示刺激系列として編集可能な10図形を示している。編集したい図形をマウスによって指定すると、その図形が②の編集画面に呼出される。③のメニューは以下のような階層構造になっているので、目的のコマンドをマウスで選択の後、図形編集を行う。④の属性情報は、座標モードの他に、編集画面で線分を描くときの始点と終点座標などが随時表示される。

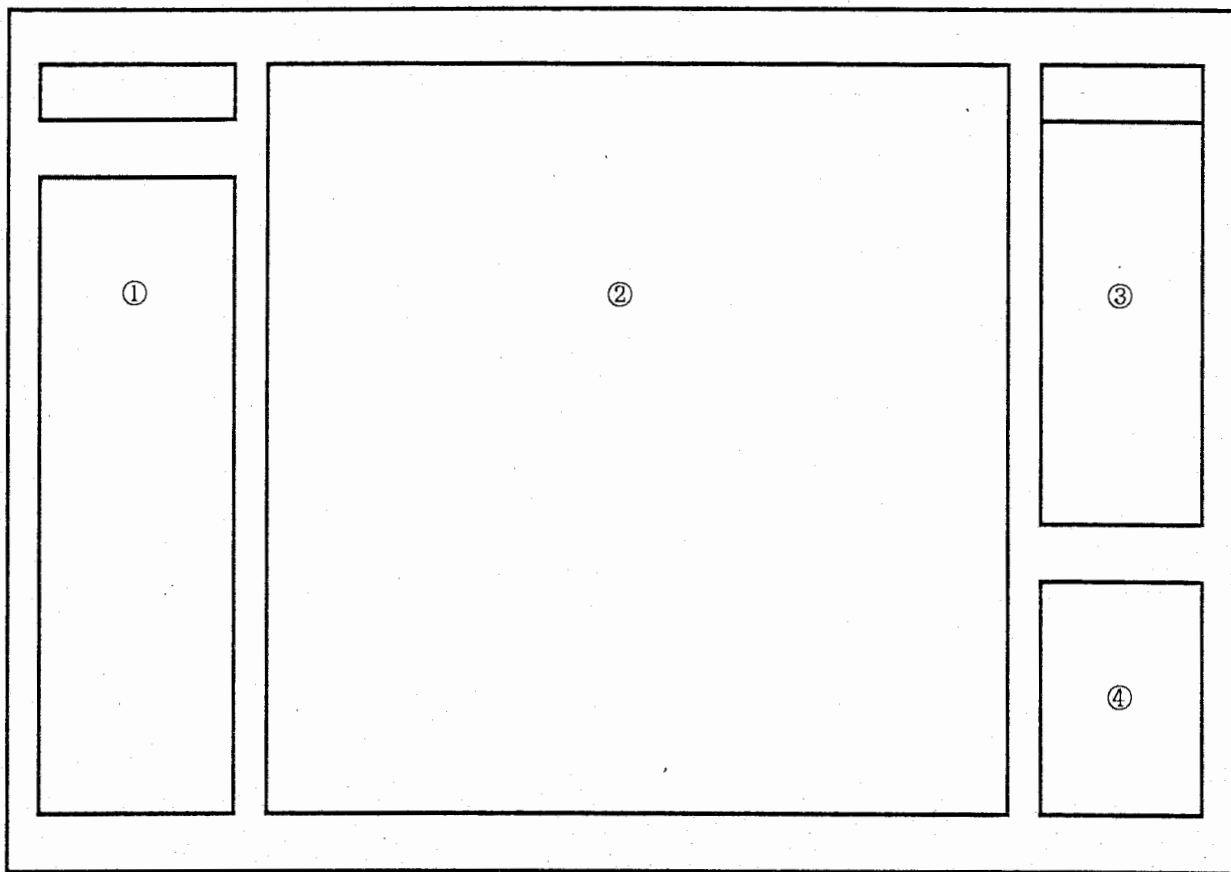
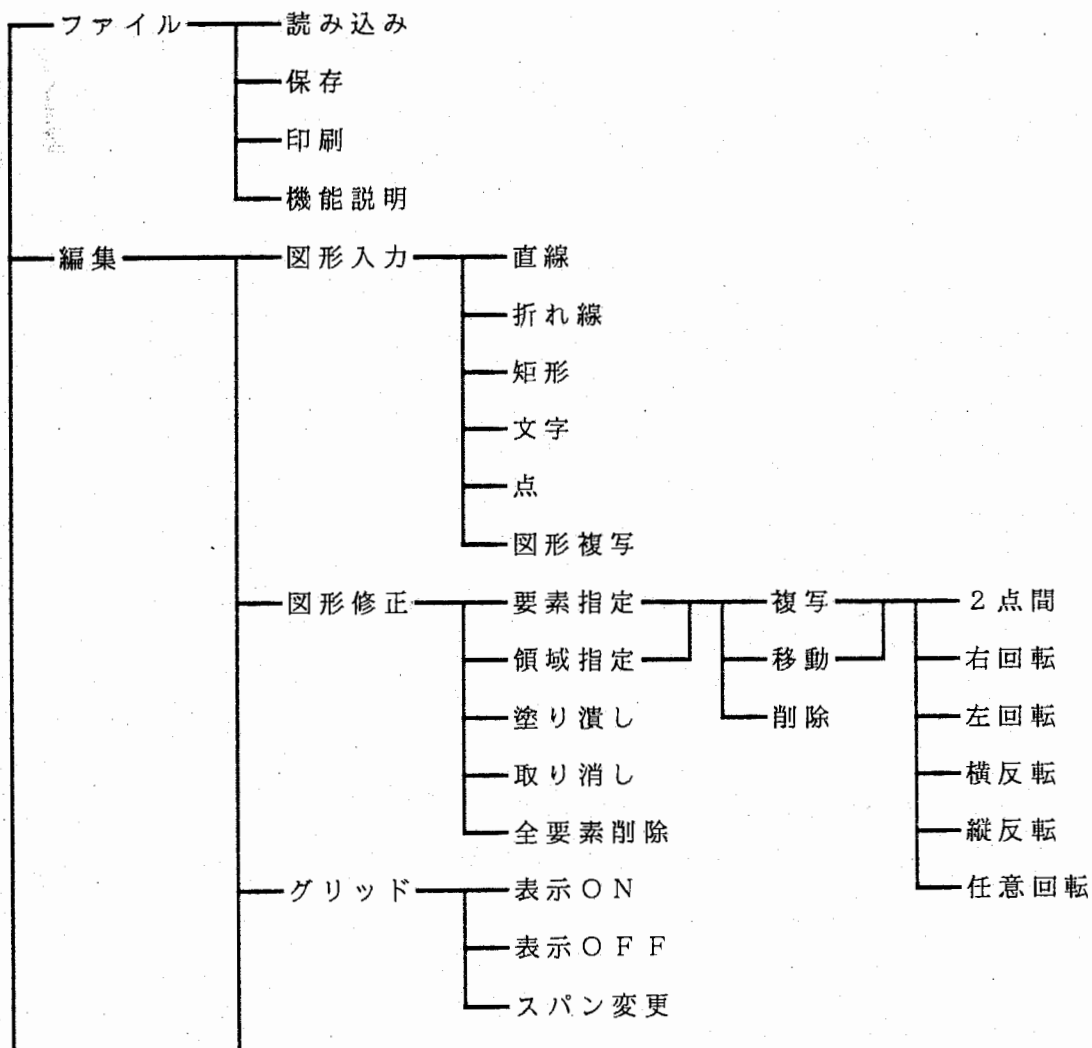
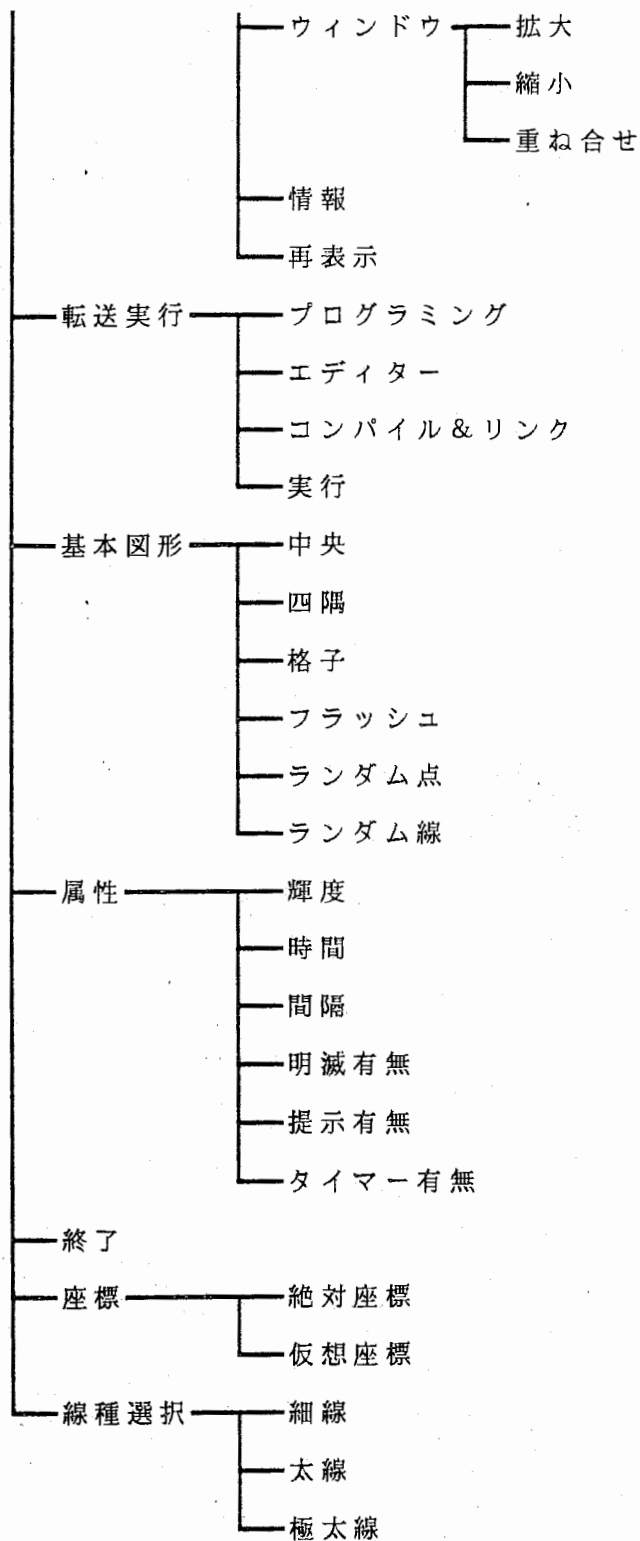


図4 グラフィックコンパイラ画面





## 4. 2 ファイル操作

### 4. 2. 1 読み込み

登録済みの図形データを読み込んで画面に表示する。すでに図形が読み込まれて修正が行われた場合、もしくは新規で作成された場合において、そのデータがファイルに保存されていない場合は、次のメッセージを表示し保存の確認を行う。

編集された図形はまだ保存されていません。

ファイルに残しますか。 ([Y] / N)

Yを入力すれば保存処理を行う。Nを入力すればファイル名の入力を行う。

ファイル名：

読み込むファイル名を入力する。ファイル名が存在すればそのファイルを読み込む。  
ファイル名が存在しなければ確認のメッセージが表示される。

指定されたファイルは存在しません。

新規ですか。（[Y] / N）

Yを入力すれば新規としてファイルをオープンする。Nを入力すればファイル名の入力に戻る。

#### 4. 2. 2 保存

作成、又は修正された図形データの保存を行う。

(1) ファイル名が入力されていれば、次のメッセージが表示される。

現在のファイル名は\*\*です。

このファイル名で保存しますか。

Yを入力すれば保存は現在のファイル名で行う。Nを入力すればファイル名の入力を行う。

ファイル名を入力して下さい。

ファイル名を入力する。

(2) ファイル名が入力されていなければ、次のメッセージが表示される。

ファイル名が指定されていません。

ファイル名を入力してください。

ファイル名を入力する。

上記(1)、(2)の入力を行った後、保存の確認を行う。

保存しますか。（[Y] / N）

Yを入力すればデータは保存される。Nを入力すればメニュー選択に戻る。

#### 4. 2. 3 印刷

図形データのハードコピーを行う。メニューが選択されると画面に図形の全てが属性と共に表示される。ハードコピーをとる場合は左のマウスボタンを、メニュー選択に戻る場合は右のマウスボタンを押す。

#### 4. 2. 4 機能説明

ヘルプファイルを画面に表示する。

↑キーを押すとページ単位でスクロールアップする。

↓キーを押すと1行単位でスクロールダウンする。

RDキーを押すとページ単位でスクロールダウンする。

- リターンキーを押すと機能説明は終了する。
- 0キーを押すとトップメニューを表示する。
- 1キーを押すと機能概要を表示する。
- 2キーを押すとファイル説明の先頭を表示する。
- 3キーを押すと編集説明の先頭を表示する。
- 4キーを押すと転送実行説明の先頭を表示する。
- 5キーを押すと基本図形説明の先頭を表示する。
- 6キーを押すと属性説明の先頭を表示する。
- 7キーを押すと終了説明の先頭を表示する。

#### 4. 3 編集

##### 4. 3. 1 図形入力

###### 4. 3. 1. 1 直線入力

2点を指定する事により直線を生成する。

###### 4. 3. 1. 2 折れ線

連続して座標を指示する事により折れ線を生成する。

###### 4. 3. 1. 3 矩形

2点を対角とする矩形を生成する。(線分は最大1024本まで生成出来る。)

###### 4. 3. 1. 4 文字

文字を生成する。1文字列最高40文字までで、1図形で20文字列生成出来る。

漢字の入力は日本語F P (松茸)を起動することによって可能である。

###### 4. 3. 1. 5 点

点を生成する。座標を指示するとその位置に点が生成され、最大200個まで点が生成出来る。

###### 4. 3. 1. 6 図形複写

現在編集中の図形に他の図形をコピーする。このメニューを選択するとマウスカーソルが赤色になり、その状態で図形列のいずれかを指示します。

##### 4. 3. 2 図形修正

###### 4. 3. 2. 1 要素指定

各要素とマウスカーソルで指示する事により修正の対象となる要素を選択する。右の

マウスボタンを押すと指示は修正する。(修正の対象となる要素は色が赤に変わる。)

#### 4.3.2.2 領域指定

2点を指示し、その2点を対角とする短形の中に含まれる要素を修正の対象とする。  
(修正の対象となる要素は色が赤に変わる。)

#### 4.3.2.3 塗り潰し

閉じた図形要素の中を塗り潰す。閉じた図形をマウスによって指示する事により、その図形を塗り潰す。始点と終点が一致した時点で塗り潰しを行う。

#### 4.3.2.4 取り消し

修正を取り消す。修正を終えた直後であれば、このメニューを選択する事により修正前の状態に戻ることが出来る。但し、塗り潰しと全要素削除については機能しない。

#### 4.3.2.5 全要素削除

現在編集集中の図形要素すべてを削除する。削除する前に次のメッセージによって確認する。

OK:

Yを入力すれば図形要素はすべて削除され、Nを入力すれば何もしないでメニュー選択に戻る。

#### 4.3.2.6 複写

4.3.2.1又は、4.3.2.2によって指示された修正要素を次の6つの方法により複写する。

##### (1) 2点間

2点を指示する事により1点目から2点目の方向に、2点間の距離だけ離れた所に図形を複写する。

##### (2) 右回転

回転の中心を与えることにより、時計回りに90度だけ回転した位置に図形を複写する。

##### (3) 左回転

回転の中心を与えることにより、反時計回りに-90度だけ回転した位置に図形を複写する。

##### (4) 横反転

反転のX座標を与えることにより、Y軸を軸として図形を反転する。

##### (5) 縦反転

反転の Y 座標を与えることにより、X 軸を軸として図形を反転する。

(6) 任意回転

回転の中心と回転角を与える事により回転した図形を複写する。

4. 3. 2. 7 移動

4. 3. 2. 1 又は、4. 3. 2. 2 によって指示された修正要素を次の 6 つの方法により移動する。

(1) 2 点間

2 点を指示する事により 1 点目から 2 点目の方向に、2 点間の距離だけ離れた所に図形を移動する。

(2) 右回転

回転の中心を与えることにより、時計回りに 90 度だけ回転した位置に図形を移動する。

(3) 左回転

回転の中心を与えることにより、反時計回りに -90 度だけ回転した位置に図形を移動する。

(4) 横反転

反転の X 座標を与えることにより、Y 軸を軸として図形を反転する。

(5) 縦反転

反転の Y 座標を与えることにより、X 軸を軸として図形を反転する。

(6) 任意回転

回転の中心と回転角を与える事により図形を移動します。

4. 3. 2. 8 削除

4. 3. 2. 1 又は、4. 3. 2. 2 によって指示された修正要素を削除する。

4. 3. 3 グリッド

4. 3. 3. 1 表示 ON

グリッドを画面に表示する。グリッドの幅は X, Y とも 100 で設定されている。

(0, 0) の位置が必ずグリッドの中央になるように表示する。

このメニューを選択後は、表示 OFF が選ばれるまでマウスの座標はこのグリッドに合わされる。但し、ウィンドウの指定、要素の指示の場合は無視される。

4. 3. 3. 2 表示 OFF

グリッドの表示を取り消す。グリッドを画面から消し、マウスで指示された座標はグリッドに合わされる事はない。



#### 4. 3. 3. 3 スパン変更

グリッド間の幅を指示する。画面右下に次のようなメッセージが表示され、X, Y のスパン幅を入力する。

X 幅 :

Y 幅 :

10未満のスパンは指示する事は出来ない。入力後、グリッドは自動的に再表示される。

#### 4. 3. 4 ウィンドウ

##### 4. 3. 4. 1 拡大

図形の拡大を行う。2点を指示する事により、2点を対角とする矩形を新しいウィンドウとする。拡大を繰り返す事も可能である。

##### 4. 3. 4. 2 縮小

拡大された図形を元に戻す。この機能は、拡大されたウィンドウを元に戻すものであって実際の縮小は行えない。

##### 4. 3. 4. 3 重ね合わせ

2つの座標を重ね合わせて見る。現在表示されている図形に他の図形を重ね合わせることができる。メニューを選択するとマウスの色が赤に変わるのでその状態で図形系列の中から重ねたい図形を指示する。すでに表示されている図形は緑色、重ね合わせる図形は黄色、図形の重なった部分は赤色になる。

#### 4. 3. 5 情報

現在表示されている要素の数、及び点、線の座標を知る。画面右下に各要素の数が表示される。要素の数の中にはランダム点、ランダム線、フラッシュ、塗り潰しの数は含まれない。

点 : 点の数

線分 : 線分の数

文字 : 文字列の数

カーソルは要素指示状態にあり、点、線、を指示すると要素の色が変わり、座標を表示する。

#### 4. 3. 6 再表示

現在表示されている図形をかき直す。グリッド表示時に要素を削除したり、要素を修

正した場合に表示が見にくくなる場合があるが、その時にこのメニューを選択すれば図形を再表示して画面を奇麗にする。

#### 4. 4 転送実行

##### 4. 4. 1 プログラミング

図形データをコンパイルし、フォートランソースプログラムを作成する。まず、表示するディスプレイタイプを指示する。

ディスプレイタイプ番号：

ディスプレイタイプ番号（1-7）又は実行時に選択する方法をとる8番を入力する。次に、コンパイルの条件を入力する。

メインプログラムですか。（[Y] / N）

Yを入力すればメインプログラムとしてコンパイルする。Nを入力すればサブルーチンとしてコンパイルする。

実験モードですか。（[Y] / N）

Yを入力すれば実験モードでコンパイルする。Nを入力すれば通常モードでコンパイルする。

通常モードの場合のみ次の入力要求となる。

重ねますか。（[Y] / N）

Yを入力すれば重ね書きモードでコンパイルする。Nを入力すれば図形を個々に見るモードでコンパイルする。

ファイル名を入力して下さい。

フォートランソースのファイル名を入力します。

説明文を入力して下さい。

フォートランソースに挿入するコメントを入力します（40字以内）。

##### 4. 4. 2 エディター

エディター（RED++）により、生成されたフォートランソースを編集する。フォートランソースが生成されていない場合は次のエラーメッセージを表示する。

図形がコンパイルされていません。

どれかキーを押すと元の画面に戻ります。

##### 4. 4. 3 コンパイル&リンク

生成されたフォートランソースのコンパイル、リンク及び実行を行う。

プログラム\*\*\*.FORをコンパイルしますか。（[Y] / N）

Yを入力すればPC-Fortranコンパイラによってプログラムはコンパイルされる。Nを入力すれば元の画面に戻る。

リンクしますか。 ([Y] / N)

Yを入力すればMS-LINKERによってオブジェクトモジュールはリンクされる。Nを入力すれば何もせず次の処理に進む。

実行しますか。 ([Y] / N)

Yを入力すれば実行を開始する。実行後、確認のメッセージが表示され元の画面に戻る。Nを入力すれば確認メッセージが表示され元の画面に戻る。

#### 4. 4. 4 実行

プログラムの実行を開始する。実行後、確認のメッセージが表示され元の画面に戻る。

#### 4. 5 基本図形

##### 4. 5. 1 中央

画面の中央に点を基本図形として生成する。サブメニューとして次の3つがある。

- ① 新規 画面に新しく基本図形を生成し基本図形セーブファイル (K I H O N Z U . S A V) に登録する。
- ② 参照 基本図形セーブファイルから座標を読み込んで基本図形を生成する。
- ③ 取り消し 基本図形を取り消す。

①, ②は中央に関しては、結果は同じだが、修正機能によって点を中央から移動させて基本図形セーブファイルに登録した場合などは、次回参照した時にはその座標で点を生成する事になる。

##### 4. 5. 2 四隅

画面に (0, 0) を中心とした4点を基本図形として生成する。サブメニューとして次の3つがある。

- ① 新規 画面に新しく基本図形を生成し基本図形セーブファイル (K I H O N Z U . S A V) に登録する。
- ② 参照 基本図形セーブファイルから座標を読み込んで基本図形を生成する。
- ③ 取り消し 基本図形を取り消す。

新規の場合は画面右下に入力要求メッセージが表示される。

幅 :

点と点の間の距離を入力する。

##### 4. 5. 3 格子

画面に (0, 0) を中心とした格子状の基本図形を生成する。サブメニューとして次の3つがある。

- ① 新規 画面に新しく基本図形を生成し基本図形セーブファイル (K I H O N

Z U. S A V) に登録する。

- ② 参照 基本図形セーブファイルから座標を読み込んで基本図形を生成する。
- ③ 取り消し 基本図形を取り消す。

新規の場合は画面右下に入力要求メッセージが表示される。

幅：

格子形状の幅を入力する。

#### 4. 5. 4 フラッシュ

任意領域をフラッシュマスクパターンとして塗り潰した基本図形を生成する。サブメニューとして次の3つがある。

- ① 新規 画面に新しく基本図形を生成し基本図形セーブファイル (K I H O N Z U. S A V) に登録する。
- ② 参照 基本図形セーブファイルから座標を読み込んで基本図形を生成する。
- ③ 取り消し 基本図形を取り消す。

新規の場合はマウスカーソルにより2点を指示しその2点を対角とする矩形の内部をフラッシュさせる。

#### 4. 5. 5 ランダム点

任意領域をランダムな点で埋める基本図形を生成する。サブメニューとして次の3つがある。

- ① 新規 画面に新しく基本図形を生成し基本図形セーブファイル (K I H O N Z U. S A V) に登録する。
- ② 参照 基本図形セーブファイルから座標を読み込んで基本図形を生成する。
- ③ 取り消し 基本図形を取り消す。

新規の場合はまず濃度を入力する。

濃度：

(0.0-1.0)の濃度を入力する。デフォルトは0.5で、入力無しでリターンキーを押すと濃度は0.5になる。濃度を0.5とすれば面積の半分を点で埋めることになるが、ランダムデータを発生する時点で、既に発生したデータを除外することが出来ないので正確に半分ではない。

マウスカーソルにより2点を指示しその2点を対角とする矩形の内部を点で埋める。図形列としての表示は時間の短縮のため矩形内部を塗り潰している。

#### 4. 5. 6 ランダム線

任意領域をランダムな線で埋める基本図形を生成する。サブメニューとして次の3つがある。

- ① 新規 画面に新しく基本図形を生成し基本図形セーブファイル（K I H O N Z U . S A V）に登録する。
- ② 参照 基本図形セーブファイルから座標を読み込んで基本図形を生成する。
- ③ 取り消し 基本図形を取り消す。

新規の場合はまず濃度を入力します。

濃度：

(0.0-1.0)の濃度を入力する。デフォルトは0.5で、入力無しでリターンキーを押すと濃度は0.5になる。濃度を0.5とすれば面積の半分を線で埋めることになるが、ランダムデータを発生する時点で、既に発生したデータを除外することが出来ないので正確に半分ではない。

マウスカーソルにより2点を指示しその2点を対角とする矩形の内部を点で埋める。

#### 4.6 属性

##### 4.6.1 輝度

現在表示されている図形の輝度を変更する。

輝度：

1-15の値を入力する。デフォルトは8で、入力なしでリターンキーを押すと輝度は8となる。

##### 4.6.2 時間

現在表示されている図形の提示時間を変更する。

時間：

1-16383の値を入力する。デフォルトは50で、入力なしでリターンキーを押すと時間は50となる。

##### 4.6.3 間隔

現在表示されている図形の提示時間間隔を変更する。

間隔：

1-16383の値を入力する。デフォルトは100で、入力なしでリターンキーを押すと間隔は100となる。

##### 4.6.4 明滅

現在表示されている図形の明滅を指示する。このメニューはスイッチになっており選択する毎に明滅有り無しが切り換わる。デフォルトは明滅無し。

##### 4.6.5 提示

現在表示されている図形の提示有無を指示する。このメニューはスイッチになっており選択する毎に提示有り無しが切り換わる。デフォルトは提示有り。

#### 4. 6. 6 タイマー

タイマーをカウントするかしないかを指示する。カウントする場合はリアクションキーの待ち受けが可能となる。カウントしない場合はマウスの使用が可能となる。このメニューはスイッチになっており選択する毎に検出有り無しが切り換わる。デフォルトはタイマーカウント有り。

#### 4. 7 終了

本システムの終了を行う。編集されたデータがまだファイルに保存されていない場合は次のメッセージを表示し保存の確認を行う。

編集された図形はまだ保存されていません。

ファイルに残しますか。 ([Y] / N)

Yを入力すれば保存処理を行う。Nを入力すればシステム終了します。

#### 4. 8 割込みメニュー

##### 4. 8. 1 座標モード

[絶対座標]位置をマウスによって指示すると、絶対座標と相対座標が切換わる。

##### 4. 8. 1. 1 絶対座標

図形の表示を絶対座標系で行う。絶対座標系で表示するとすべての図形が表示出来ないため1部分だけを表示する。表示されていない部分を見たい場合は外辺部分をマウスで指示することによって、スクロールしながら見る事ができる。

##### 4. 8. 1. 2 仮想座標

図形の表示を仮想座標系で行う。図形は画面にすべて表示されるが、カーソルから入力した座標はATRタキストコープの座標に1対1に対応していない。なぜならば、PC-9800画面上は400\*400なのに対して、ATRタキストコープは1024\*1024である為である。

##### 4. 8. 2 線種選択

[細線]位置をマウスで指示すると細線、太線、極太線が順に切換わる。細線、太線、極太線の切換えは図形入力時のみ可能である。細線は1本線、太線は3本線、極太線は5本線で線分を表現する。

## 5. おわりに

A T R タキストスコープを制御する為のソフトウェアの解説を行った。特に、グラフィックコンパイラによって、プログラムを意識せずに実験準備が行えるようにした。従って、このA T R タキストスコープが、多くの研究者による様々な実験に利用され、視覚機構の解明に貢献することを望みます。

## 謝辞

日頃御討論頂く視覚研究室の諸氏に感謝致します。本装置のソフトウェアの開発に御協力頂いた株式会社東洋情報システムに感謝致します。

## 参考文献

- 1) 横澤、佐藤、梅田：“高速視覚刺激提示装置A T R タキストスコープ”、A T R テクニカルレポート、TR-A-0009,1987

付録 テストプログラム

```
C*****
C
C PROGRAM NAME : TEST1
C
C FUNCTION : TEST PROGRAM NO.1
C
C CALLING SEQUENCE
C
C*****
C
C PROGRAM TEST1
C *****
C TEST PROGRAM NO.1
C
C COMMON / GECT / IJECT(30)
C
C INTEGER A(256)
C
C WRITE(*,'(A)') ' TEST 1 開始 '
C
C CALL GOPEN(IJECT,30)
C CALL GDOA(A,256)
C
C CALL GDMODE(ITY)
C CALL GSLDTY(ITY)
C
C CALL GEINIT(A,1,0,0,0,15,-2)
C
C CALL MOVABS(A,-500,-500)
C CALL NDRW (A,50,0,4)
C CALL NDRW (A,-50,87,2)
C CALL DRWREL(A,-100,-174)
C CALL MOVREL(A,0,200)
C CALL NPNT (A,50,0,4)
C CALL NPNT (A,-25,43,4)
C CALL PNTREL(A,-25,-43)
C CALL NPNT (A,-25,-43,3)
C
C CALL GDISP(A)
C CALL TINPUT(ILR)
C
C CALL GMOVE(1,-100,100)
C CALL TINPUT(ILR)
C
C CALL GMOVE(1,-300,-500)
C CALL TINPUT(ILR)
C
C CALL GCLOSE
C
C WRITE(*,'(A)') ' テスト 1 終了 '
C
C STOP
C END
```



```

C*****
C
C PROGRAM NAME : TEST2
C
C FUNCTION : TEST PROGRAM NO.2
C
C CALLING SEQUENCE
C
C*****
C
C PROGRAM TEST2
C *****
C TEST PROGRAM NO.2
C
C COMMON / GECT / IJECT(30)
C COMMON / GCT / IGCT(18)
C
C INTEGER V(4,2),W(4,2),N(4,2),A(256)
C
C V(1,1) = 1
C V(2,1) = 0
C V(3,1) = 0
C V(4,1) = 0
C V(1,2) = 0
C V(2,2) = 1
C V(3,2) = 0
C V(4,2) = -1
C W(1,1) = 0
C W(2,1) = 1
C W(3,1) = 0
C W(4,1) = 1
C W(1,2) = 1
C W(2,2) = 0
C W(3,2) = -1
C W(4,2) = 0
C N(1,1) = 0
C N(2,1) = 5
C N(3,1) = 10
C N(4,1) = 15
C N(1,2) = 15
C N(2,2) = 0
C N(3,2) = 10
C N(4,2) = 5
C
C WRITE(*,'(A)') ' T E S T 2 開始 '
C
C CALL GOPEN (IJECT,30)
C
C CALL GDMODE(ITY)
C CALL GSLDTY(ITY)
C
C 1000 CONTINUE
C CALL GDSPMS(1,14)
C CALL GETFK(K,1)
C
C IF(K .EQ. 11) GOTO 6000
C IF(K.LT.2 .OR. K.GT.8) GOTO 9000
C IF(K .EQ. 7) CALL GCLEAR

```

```

C
  CALL GDOA(A,256)
  GOTO (9000,2000,3000,3000,3000,3000,4000,5000) , K
C
2000 CONTINUE
  CALL GEINIT(A,5,0,0,0,8,-2)
  CALL TESTSU(A,3)
  CALL GRDISP(A,2)
  GOTO 1000
C
3000 CONTINUE
  I=K-2
  CALL GEINIT(A,I,0,0,0,8,-2)
  CALL TESTSU(A,I)
  CALL GDISP(A)
  GOTO 1000
C
4000 CONTINUE
  DO 4100 I = 1 , 4
    CALL GEINIT(A,I,V(I,1),0,W(I,1),N(I,1),-2)
    CALL TESTSU(A,I)
4100 CONTINUE
  CALL GDISP(A)
  GOTO 1000
C
5000 CONTINUE
  DO 5100 I = 1 , 4
    CALL GALSCR(I,V(I,2),0,W(I,2))
    CALL GALINT(I,N(I,2))
5100 CONTINUE
  GOTO 1000
C
6000 CONTINUE
  CALL GSTOP
  CALL GCLOSE
C
  WRITE(*,'(A)') ' テスト 2 終了 '
  GOTO 9999
C
9000 CONTINUE
  WRITE(*,'(A)') ' テスト 2 エラー '
C
9999 CONTINUE
C
  STOP
  END

```

```

C*****
C
C PROGRAM NAME : TEST3
C
C FUNCTION : TEST PROGRAM NO.3
C
C CALLING SEQUENCE
C
C*****
C
C PROGRAM TEST3
C ***** ****
C
C TEST PROGRAM NO.3
C
C COMMON / GECT / IJECT(30)
C COMMON / GCT / IGCT(18)
C
C INTEGER A(256)
C
C WRITE(*,'(A)') ' T E S T 3 開始 '
C
C CALL GOPEN(IGECT,30)
C CALL GDOA(A,256)
C
C CALL GDMODE(ITY)
C CALL GSLDTY(ITY)
C
C DO 1000 I = 1 , 4
C   J = 1000 * (I+2)
C   CALL GEINIT(A,I,0,0,0,8,J)
C   CALL TESTSU(A,I)
C   IF(I .NE. 4) CALL GEDLY(A,J)
1000 CONTINUE
C
C CALL GPINT(A,7)
C CALL GXFER(A)
C
C WRITE(6,'(A)') ' ファンクションキーを押して下さい。 '
C CALL GETFK(IFK,1)
C
C CALL GSTART
C CALL GETINT(7)
C DO 2000 I = 1 , 4
C   J = 1000 * (7-I)
C   CALL GALTME(I,J)
C   IF(I .NE. 4) CALL GALDLY(I+1,J)
2000 CONTINUE
C
C CALL GSTART
C CALL GETINT(7)
C CALL GDOA(A,256)
C CALL GEINIT(A,1,0,0,0,8,3000)
C DO 3000 I = 1 , 4
C   CALL TESTSU(A,I)
C   CALL GEDLY(A,1000*(I+3))
3000 CONTINUE
C
C CALL GPINT(A,8)

```

```
CALL GCLEAR
CALL GXFER(A)
CALL GSTART
CALL GETINT(8)
CALL GDOA(A,256)
CALL GEINIT(A,1,0,0,0,8,16383)
DO 4000 I = 1, 4
    CALL TESTSU(A,I)
    CALL GEINIT(A,I+1,0,0,0,8,-1)
4000 CONTINUE
C
    CALL GCLEAR
    CALL GXFER(A)
    CALL GSTART
C
    CALL GCLOSE
C
    WRITE(*,'(A)') 'テスト3終了'
C
    STOP
    END
```

```

C*****
C
C PROGRAM NAME : TEST4
C
C FUNCTION : TEST PROGRAM NO.4
C
C CALLING SEQUENCE
C
C*****
C
C PROGRAM TEST4
C *****
C
C TEST PROGRAM NO.1
C
C COMMON / GECT / IJECT(30)
C
C CHARACTER*8 MS1(15),MS2(6)
C
C DATA MS1/'XXXX MS ','CLK 50NS','CLK100NS','CLK150NS','CLK200NS',
1          'INT LV X',' 60FRAME','120FRAME','VDOSEL 2','TIME ON ',
2          'TIME OFF','MOUSE ON','MOUSEOFF','60FRM1/2','PUSH KEY'/
C
C DATA MS2/'TEST4 ','PC-9800 ','ATR ','タキスト ','スコ-プ ','
1          'END '/
C
C WRITE(6,'(A)') ' T E S T 4 開始 '
C
C CALL GOPEN(IJECT,30)
C
C WRITE(6,'(A)') ' 反応測定盤に E N D が表示されるまで '
C WRITE(6,'(A)') ' 繰り返しファンクションキーを押して下さい。 '
C WRITE(6,'(A)') ' '
C WRITE(6,'(A)') ' ロム内メッセージテスト '
C
C DO 1000 I = 0 , 14
C   WRITE(6,'(A,I4,A,A8)') ' メッセージ',I,' : ',MS1(I+1)
C   CALL GDSPMS(1,I)
C   CALL GETFK(IFK,1)
1000 CONTINUE
C
C CALL GSTMES(0,'TEST4 ')
C CALL GSTMES(1,'PC-9801 ')
C CALL GSTMES(2,'ATR ')
C CALL GSTMES(3,'タキスト ')
C CALL GSTMES(4,'スコ-プ ')
C CALL GSTMES(5,'END ')
C
C WRITE(6,'(A)') ' 追加メッセージテスト '
C
C DO 2000 I = 0 , 5
C   WRITE(6,'(1H ,A8)') MS2(I+1)
C   CALL GDSPMS(0,I)
C   CALL GETFK(IFK,1)
2000 CONTINUE
C
C CALL GCLOSE
C WRITE(6,'(A)') ' テスト 4 終了 '
C

```

STOP  
END

1000

```

C*****
C
C PROGRAM NAME : TEST5
C
C FUNCTION : TEST PROGRAM NO.5
C
C CALLING SEQUENCE
C
C*****
C
C PROGRAM TEST5
C *****
C
C COMMON /GECT/IGECT(30)
C
C INTEGER A(2048)
C CHARACTER*1 IYN
C
C----- OPEN GRAPHIC DISPLAY -----
C
C CALL GOPEN(IGECT,30)
C CALL GDOA (A,2048)
C
C 1000 CONTINUE
C CALL GDMODE(ITY)
C CALL GSLDTY(ITY)
C
C IF(ITY.LT.5) THEN
C WRITE(6,'(A)') ' 5 から 7 までの値を入力をして下さい。'
C GOTO 1000
C ENDIF
C
C WRITE(6,'(A,$)') ' ビデオメモリーに書き込みますか。(Y/N) : '
C READ(5,'(A1)',ERR=9000) IYN
C
C IF( IYN .EQ. 'Y' ) THEN
C CALL GWVMEM(0,7,'TMP.DAT',401,17)
C WRITE(6,'(A)') ' ビデオメモリー 0 番の書き込み終了。'
C CALL GWVMEM(1,7,'TMP.DAT', 1, 1)
C WRITE(6,'(A)') ' ビデオメモリー 1 番の書き込み終了。'
C ENDIF
C
C WRITE(6,'(/,A,$)') ' 0 番の呈示時間 : '
C READ (5,*,ERR=9000) IT1
C WRITE(6,'(A,$)') ' 0 番の呈示間隔 : '
C READ (5,*,ERR=9000) ID1
C
C WRITE(6,'(A,$)') ' 1 番の呈示時間 : '
C READ (5,*,ERR=9000) IT2
C WRITE(6,'(A,$)') ' 1 番の呈示間隔 : '
C READ (5,*,ERR=9000) ID2
C
C CALL GCTLRS(A,0,0,0,0,10,IT1)
C CALL GRDLAY(A,ID1)
C CALL GCTLRS(A,1,0,0,1,10,IT2)
C CALL GRDLAY(A,ID2)
C
C DISPLAY START

```

```
      CALL GRSTAT(A,1)
C
C INTERRUPT
      CALL GTRINT(NO)
      WRITE(6, '(1H ,I4,8N)') NO, '  番の表示終了。'
      CALL GTRINT(NO)
      WRITE(6, '(1H ,I4,8N)') NO, '  番の表示終了。'
C BUFFER CLEAR
      CALL GCLEAR
      CALL GCLOSE
      GOTO 9999
C
9000 CONTINUE
      WRITE(6, '(A)') '  入力された値が異常です。'
C
9999 CONTINUE
      STOP
      END
```



```

C*****
C
C PROGRAM NAME : TEST6
C
C FUNCTION : TEST PROGRAM NO.6
C
C CALLING SEQUENCE
C
C*****
C
C PROGRAM TEST6
C *****
C
C COMMON /GECT/IGECT(30)
C INTEGER A(2048)
C
C----- OPEN GRAPHIC DISPLAY -----
C
C CALL GOPEN(IGECT,30)
C
C*****
C ELEMENTS INITIARIZE (ID = 1)
C*****
C
C CALL GDMODE(ITY)
C CALL GSLDTY(ITY)
C CALL GDOA (A,2048)
C
C CALL GEINIT(A,1,0,0,0,8,-2)
C CALL MOVABS(A,-460,380)
C CALL GKANJI (A,14,'この高速視覚刺激提示装置は、')
C CALL MOVABS(A,-460,230)
C CALL GKANJI (A,13,'漢字、ひらがな、カタカナ、')
C CALL MOVABS(A,-460,80)
C CALL GKANJI (A,9,'ALPHABETを')
C
C CALL GEINIT(A,2,0,0,0,8,-3)
C CALL MOVABS(A,-460,-70)
C CALL GKANJ2 (A,1,'大',150,100,0.0,10)
C
C CALL GEINIT(A,3,0,0,0,8,-4)
C CALL MOVABS(A,-350,-70)
C CALL GKANJ2(A,1,'小',32,32,0.0,10)
C
C CALL GEINIT(A,4,0,0,0,15,-2)
C CALL MOVABS(A,-180,-70)
C CALL GKANJI(A,1,'明')
C
C CALL GEINIT(A,5,0,0,0,5,-2)
C CALL MOVABS(A,-100,-70)
C CALL GKANJI(A,1,'暗')
C
C CALL GEINIT(A,6,0,0,0,8,-2)
C CALL MOVABS(A,80,-70)
C CALL GKANJ2(A,2,'回転',64,64,30.0,10)
C
C CALL GEINIT(A,7,0,0,1,8,-2)
C CALL MOVABS(A,300,-70)
C CALL GKANJI(A,2,'明滅')

```

C

```
CALL GEINIT(A,8,0,0,0,8,-2)
CALL MOVABS(A,-460,-220)
CALL GKANJI (A,14,'提示時間などを自由に設定して')
CALL MOVABS(A,-460,-370)
CALL GKANJI (A,7,'提示できません。')
```

C

```
CALL GEINIT(A,9,0,0,1,8,-2)
CALL MOVABS(A,-460,-470)
CALL ANSTR(A,11,'WE LOVE ATR')
```

C

```
CALL GEINIT(A,10,0,0,0,8,-2)
CALL MOVABS(A,300,-470)
CALL ANSTR(A,4,'1987')
```

C

```
CALL GCLEAR
CALL GDISP(A)
CALL TINPUT(I)
```

C

```
C BUFFER CLEAR
CALL GCLEAR
```

C

```
CALL GCLOSE
```

C

```
STOP
END
```

```

C*****
C
C PROGRAM NAME : TEST7
C
C FUNCTION : TEST PROGRAM NO.7
C
C CALLING SEQUENCE
C
C*****
C
C PROGRAM TEST7
C *****
C
C COMMON / GECT / IJECT(30)
C
C INTEGER A(2048)
C NCHARACTER*2 INUM(16)
C CHARACTER*1 IYN
C
C DATA INUM / ' 0 ',' 1 ',' 2 ',' 3 ',' 4 ',' 5 ',' 6 ',
1 ' 7 ',' 8 ',' 9 ',' 10 ',' 11 ',' 12 ',' 13 ',
2 ' 14 ',' 15 ' /
C
C----- START -----
C
C OPEN GRAPHIC DISPLAY
C
C CALL GOPEN(IJECT,30)
C CALL GDOA (A,2048)
C
C 1000 CONTINUE
C CALL GDMODE(ITY)
C CALL GSLDTY(ITY)
C
C IF(ITY.LT.5) THEN
C WRITE(6,'(1H ,19N)') '5 から 7 までの値を入力をして下さい。'
C GOTO 1000
C ENDIF
C
C 1100 CONTINUE
C WRITE(6,'(1H ,22N,$)')
1 'ビデオメモリーに書き込みますか。(Y/N) : '
C READ(5,'(A1)',ERR=1100) IYN
C
C IST = 1
C IF( IYN .EQ. 'Y' ) THEN
C WRITE(6,'(1H ,22N)')
1 'ファイル” TMP . D A T ” から読み込みます。'
C DO 2000 I = 0 , 14 , 2
C CALL GWVMEM( I,7,'TMP.DAT',IST,1)
C WRITE(6,'(1H ,8N,N2,9N)') 'ビデオメモリー',INUM(I+1),
1 '番の書き込み終了。'
C CALL GWVMEM(I+1,7,'TMP.DAT',IST,33)
C WRITE(6,'(1H ,8N,N2,9N)') 'ビデオメモリー',INUM(I+2),
1 '番の書き込み終了。'
C IST = IST + 512
C 2000 CONTINUE
C ENDIF
C

```

```
CALL GRCONT(IGDOA,10,10)
```

C

```
WRITE(6,'(1H ,17N)') 'どれか、ファンクションキーを押すと'
```

```
WRITE(6,'(1H ,11N)') '連続表示を停止します。'
```

```
CALL GETFK(IFK,1)
```

C

```
CALL GRSTOP
```

```
CALL GCLOSE
```

C

```
STOP
```

```
END
```

```

C*****
C
C PROGRAM NAME : TEST8
C
C FUNCTION : TEST PROGRAM NO.8
C
C CALLING SEQUENCE
C
C*****
C
C PROGRAM TEST8
C *****
C TEST PROGRAM NO.8
C
C COMMON / GCT / IGCT(18)
C COMMON / GECT / IGECT(30)
C
C DIMENSION IGDOA(100),JGDOA(100)
C
C CALL GOPEN (IGECT,30)
C
C 818 CALL GSLDTY(10)
C 819 CALL GSTMES(257,'12345678')
C
C 803 CALL GDOA (IGDOA,5)
C CALL GDOA (IGDOA,100)
C 804 CALL GEINIT(IGDOA,0,0,0,0,7,1000)
C 805 CALL GEINIT(IGDOA,1,0,0,0,7,16384)
C 806 CALL GEINIT(IGDOA,1,0,0,0,16,100)
C CALL GEINIT(IGDOA,1,0,0,0,7,-2)
C 800 CALL MOVABS(IGDOA,513,0)
C 801 CALL MOVABS(IGDOA,0,513)
C 802 CALL NDRW (IGDOA,20,20,-1)
C 807 CALL GPINT (IGDOA,256)
C
C CALL MOVABS(IGDOA,0,0)
C CALL GPINT (IGDOA,1)
C CALL GCLEAR
C CALL GEINIT(IGDOA,2,0,0,0,7,-2)
C 812 CALL GRDISP(IGDOA,2)
C CALL GDISP (IGDOA)
C 815 CALL GALTME(1,10000)
C CALL GDOA (JGDOA,100)
C CALL GSPACE(JGDOA,50)
C

```

CALL GCLEAR  
CALL GCLOSE  
STOP  
END