

Moving S1

[Moving] Ver.11-7 By H.Takeishi

システム メモリ- MAP (A000-BFFF)

MA (C000-CFB5) SC (A000-A07F)
PD (A080-A0FF) SW (A100-A16F)
PW (A170-A21F) FR (A280-BFFF)

N(8) S(4) C(2) P(943)

[Comand]

MVSET GSIZE GDSET GDISP GDGET MVMAP



はじめに

“BASICで広がる世界”それはBASICが開発されたときの思惑を大きく越えたものとなった。BASICは今やマイコン界の第一人者たる言語として位置付けられるに至った。汎用性があるため応用範囲が広く今なお新しい世界が開拓されている。

BASICは随時翻訳方式の会話型言語で、その使い易さの面から、いろんな分野で活用されている。しかし、BASICなるがゆえんの欠点がないわけでもない。いくつかの不満点がある。インタープリターと呼ばれる翻訳部が必要で、この部分がメモリー領域のかなりのエリアを占めている。そのためにユーザーの使用メモリー領域が大きく制限される。また逐次翻訳のため処理速度が、かなり遅く、高速処理ができない。など、こういった問題点は通常のプログラミングにおいて表面化しない。が、ゲームまたは、アニメなどを用いた動画シミュレーションなどにおいては致命的なものとなる。賢明なるユーザーはBASICから離れて、アセンブラ・マシン語といったより機械（ハード）に近づいた段階での言語を用い、新境地を切り開くのである。しかし、一般のユーザーにとってこういった言語の世界は近寄りがたいものといってよく、また、飛びこんだとしても駆使できるようになるまでには、かなりのハードな努力を要することになる。結局、多くのユーザーにとってBASICは、わかりやすく、使いやすく、最初で最後の愛すべき言語なのである。そして、このベーシックオンリー派達の前にBASICの欠点という壁は大きく立ちほだかり。BASIC世界の拡大を阻止することとなる。

ところが、新しいツールが開発されBASICが、より強化された形でサポートされると話が変わってくる。“ベーシックマン”達にも新世界へのパスポートが渡されることになる。その努力が各メーカーそして多くのマイコンマニア達によって続けられている。

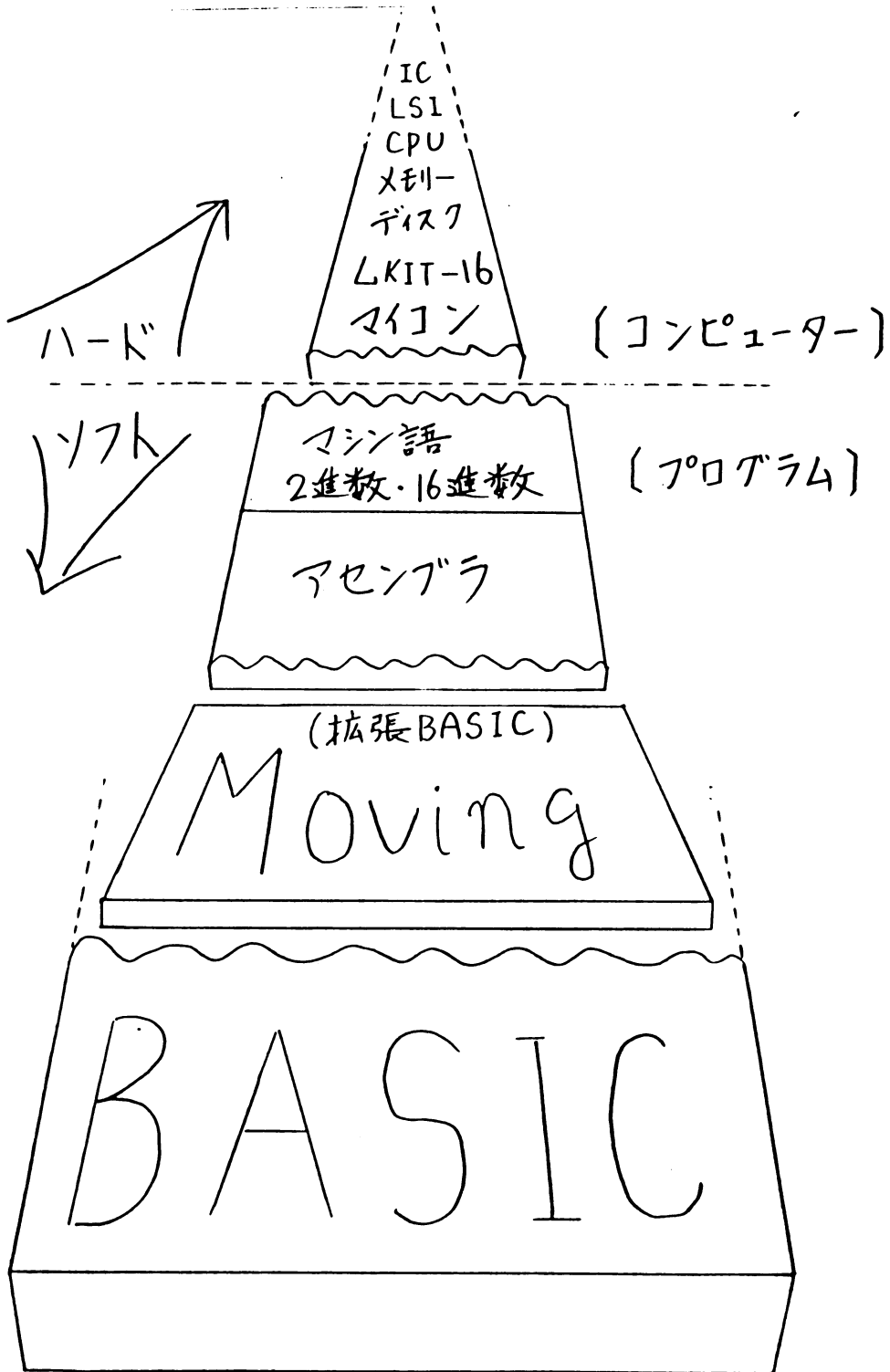
“Moving”はグラフィック・アニメ機能強化のための拡張ベーシックツールである。このツールによって画面表示のための新たなベーシック命令が追加される。とくに動画シミュレーションのための機能が強化されている。画面上のドット単位の位置に任意の色（ドット単位指定可）・大きさのグラフィックパターンを瞬時に表示するもので、画面表示の高速化、鮮明さがとくに考慮されている。さらにスプライト機能と呼ばれるグラフィック画面表示の際の優先順位付加機能は奥行きのある立体的な画面の制作を容易にしてくれる。また漢字表示のための垂直400ラインモードもサポートしているのでワープロ機能も強化できる。4Kバイトとコンパクトなツールではあるが、高度なプログラミングにおいて威力を発揮してくれるものと思う。

1985・7/28

[参考文献]

- システム・コールの使い方 (I/O誌掲載)
- S1ソフト技術マニュアル (有沢博著)

"Moving" は "拡張BASIC" である。



Movingを理解するためには、ハード的な知識が必要である。とくに16進数とメモリ領域に関する十分なる知識が必要となる。

[16進数]

コンピューターが扱うデータ(数値, コード, 状態)や命令はすべて2進数で表わされる。しかし, この2進数は人間がなれない上に, 桁数が長くなってわずらわしい。そこで, 10進数に比較的近くて, 2進数との交換が容易である16進数がよく使われる。2進数を4桁ずつ区切れば, そのまま16進数となる。

1 1 0 0 1 0 0 1 2進数
 ┌── C ─┘ ┌── 9 ─┘ 16進数

16進数で使われる数字は10進数と2進数では次のように対応している。

(16進数) (10進) (2進数)

0	0	0 0 0 0
1	1	0 0 0 1
2	2	0 0 1 0
3	3	0 0 1 1
4	4	0 1 0 0
5	5	0 1 0 1
6	6	0 1 1 0
7	7	0 1 1 1
8	8	1 0 0 0
9	9	1 0 0 1
A	10	1 0 1 0
B	11	1 0 1 1
C	12	1 1 0 0
D	13	1 1 0 1
E	14	1 1 1 0
F	15	1 1 1 1

例 色コード

G (緑) R (赤) B (青)

0	0	0	= 0 (黒)
0	0	1	= 1 (青)
0	1	0	= 2 (赤)
0	1	1	= 3 (マゼンタ)
1	0	0	= 4 (緑)
1	0	1	= 5 (シアン)
1	1	0	= 6 (黄)
1	1	1	= 7 (白)

RGBグラフィック画面指定の際
 1の値のセットされている画面が
 アクセスされる
 GDSET, GDGET文の
 <RGB面指定コード>を参照

[メモリー領域]

コンピューターが扱う情報(プログラム, 表示画面, BASICインタープリター部など)はアドレス(番地)で分類されたメモリー(記憶部)を介して処理される。8ビットマイコンでは普通0000~FFFF (16進数)の65536個のアドレスが設定され, その1つ1つのメモリーに1バイト単位のデータが保存できる。メモリーの種類としてはRAM(ランダム・アクセス・メモリー)とROM(リード・オンリー・メモリー)があり, それらはマイコンのシステムの特徴を生かした形でアドレス配置されている。

アドレス	機能
0000 3FFF	VRAM
4000 6FFF	(未定義領域)
7000 9FFF	(機械語領域)
A000 C000 CFFF	Moving (システム領域)
D000 FFFF	Basic (システム領域)

システム (Moving) 使用の機械語空間
のメモリーマップ例

- ・ CLEAR , &H7000
- ・ DEF MAP=0
- ・ MVSET 8,4,2,&HA000,&HBFFF

設定時

全てのシステムにおける使用不可

←CLEAR文で設定

BasicおよびMovingにおけるユーザー使用可

←MVSET文で設定

(IPL時この領域が設定される)

リロケートブル・マシンゴ・プログラム (約4Kバイト)

(IPL時この領域に設定される)

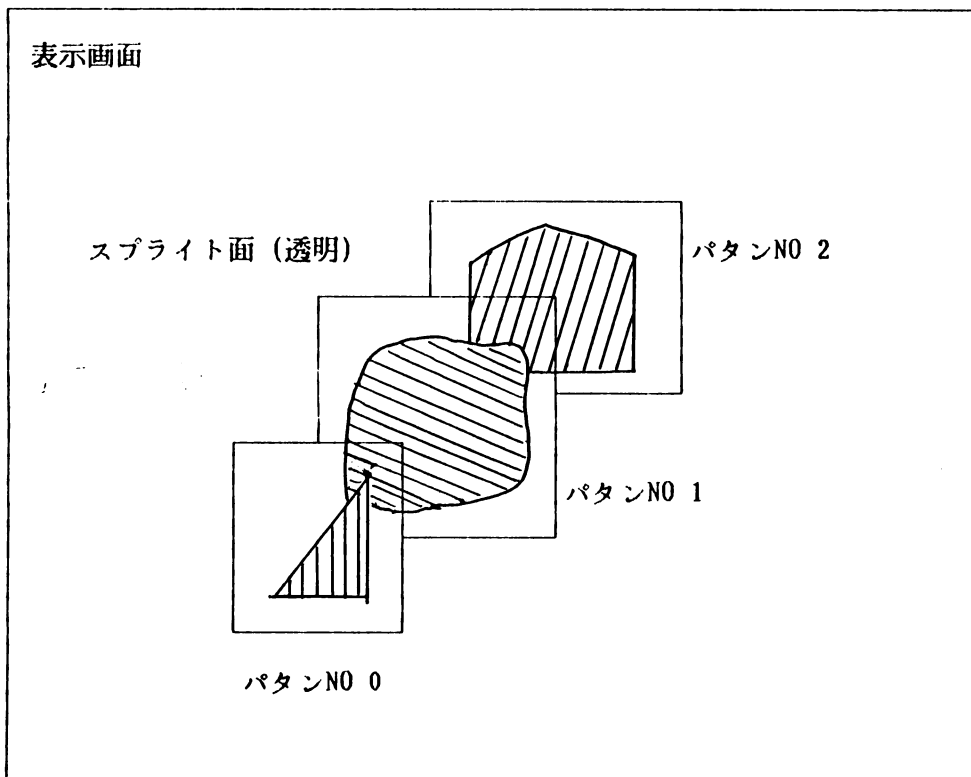
スタック・システムワーク・システムROM・I/O
として使用

BasicおよびMovingにおけるユーザー使用は不可
(D000~DFFFのスタックエリアはユーザーで使用し
てもほとんどの場合システムに支障なし)

スプライト機能・スプライト表示について

本システムによって表示されるグラフィックパターンは、アニメのセル画のような任意のサイズのグラフィック表示面（これをスプライト面と呼ぶ、ドット単位で色指定可能なスプライト面の設定可）の上に描かれたものと考えることができるそして、そのスプライト面1枚1枚に番号がついている。これがパターン番号である。1つ1つのグラフィックパターン表示用のスプライト面を予約・設定しておくことをスプライト機能といい、この機能の備った番号のグラフィックパターンを表示することをスプライト表示と呼ぶ。この種の表示されたパターンは他のグラフィックパターン表示によって消されることはない（ただし、表示優先順位の高いスプライトパターンの影に隠れて見えなくなることはある）。これに対して、スプライト機能の付加されていないグラフィックパターンは他のパターンの表示によって消され復元されることはない。

なお、各グラフィックパターンに対して、スプライト面を設定しておく方式はかなりのメモリーを必要とするため、必要最小限の個数にとどめておくべきである。また、各グラフィックパターンの画面表示の際に、その表示位置のスプライト面の有無をソフト上でチェックしているので、多くのスプライト面を定義すると、表示速度に少なからず影響が出てくることになる。



(スプライト面のサイズ・色は任意に指定できる)

拡張 BASIC 一覧表

MVSET	〈パターン数〉, 〈スプライトパターン数〉, 〈RGBスプライトパターン数〉, 〈スタートアドレス〉, 〈エンドアドレス〉	... 6
G SIZE	(〈水平サイズ〉, 〈垂直サイズ〉), 〈パターン番号〉	... 7
GDSET	〈グラフィックパターンデータ〉, 〈パターンポインター〉, 〈パターン番号〉, [〈RGB面指定コード〉]	... 8
GDSET	, 〈グラフィックパターンポインター〉, 〈パターン番号〉, [〈RGB面指定〉]	
GDSET	, , 〈パターン番号〉	... 9
GDISP	(〈水平位置〉, 〈垂直位置〉), 〈パターン番号〉, [〈色コード〉]	... 10
GDISP	(〈水平位置〉, 〈垂直位置〉), 〈色コード〉, 〈パターン番号〉, 〈水平拡大比〉, 〈垂直拡大比〉	... 11
GDISP	(〈水平位置〉), 〈垂直位置〉, 〈パターン番号〉, [〈色コード〉], , , 〈ポインター〉	... 12
GDISP	〈SGレジスター番号〉, 〈データ〉, 〈SGレジスター番号〉, 〈データ〉, 〈ポインター〉	... 12
GDISP	(〈水平位置〉, 〈垂直位置〉), 〈スタートポインター〉, 〈ポイント数〉, 〈待ち時間〉, , , 9999	... 12
GDGET	(〈水平位置〉, 〈垂直位置〉), 〈パターン番号〉	... 13
GDGET	〈漢字コード〉, 〈パターン番号〉	... 13
MVMAP	〈マップ表示〉, 〈システムアドレス〉, 〈座標系指定モード〉, 〈垂直ライン数〉, [〈色コード〉]	... 14

MVSET (ムーブ・セット)

[機能] システム (Moving) の初期セット

[書式] MVSET 〈パターン数〉, 〈スプライトパターン数〉, 〈RGBスプライトパターン数〉,
〈システムで使用するメモリ番地の初め〉, 〈システムで使用するメモリ番地の終り〉

[解説] 使用できるグラフィックパタンの数, および, その中の表示優先順位 (スプライト機能) を付加するパターン数, さらにその中のドット単位で色指定できるパターン数を設定する。初期セットで定義されるグラフィックパタンの大きさは, 列1ライン, 行8ドットのキャラクターサイズである。グラフィックパタンの内容は保障されない。システムの IPL (イニシャル・プログラム・ロード) により, パターン数は8個, スプライト付加数は4個に, RGBスプライトパターン数は2個に設定される。システムのメモリ番地の初め (下位のアドレス値は00とする), および, 終りはA000~BFFF (16進数) 領域に設定される。よって, このMVSETコマンドで, S1システムにおける各機種のパージョンにあわせて設定し直すこと。各引数値を省略 (" , " によって省略可) した場合は以前に設定した値がそのまま保存される。

パターン数は最大2048 (= &H800) 個まで定義可能。ただし, システムのメモリオーバーに注意 (エラー表示される)。また, スプライト表示は多くのメモリを使うので必要以上にスプライト付加数を定義しないほうがよい (連続表示用のエリアが少なくなる) - システムメモリアップの細解を参照 -

[注] 次の場合には必ずこのMVSETでシステム定数を設定しなおすこと。

(引数は全てを省略できる)

1) 表示画面を変更したとき

・表示画面モード変更 (SCREEN 0 ↔ SCREEN 1)

・表示画面ページ変更 (SCREEN , 1 ↔ SCREEN , 2)

2) システム領域を他のメモリ番地へ変更 (転送) し, 使用するとき

10 CLEAR , &HA000

20 MVSET 8, 4, 2, &HA000, &HBFFF

システムの起動 (IPL) 時の設定に同じ (マシン語をC000~CFFFにロード・実行)
使用できるパターンはパターンNo. 0~7の8つ, その中の0~3の4つはスプライト
機能が付加される, さらにその中の0, 1の2つのパターンはドット単位で色指定可

10 CLEAR 300, &H0000

20 MVSET 256, 128, 16, &H4000, &HBFFF

Movingのフリーエリアを最大にとる場合の設定

G S I Z E (グラフィック・サイズ)

[機能] グラフィックパターンサイズの変更

[書式] G S I Z E (〈水平サイズ〉, 〈垂直サイズ〉), 〈ボタン番号〉

[解説] キャラクターサイズに初期セットされたグラフィックパタンのサイズを拡張または縮小する。

〈水平サイズ〉 横のサイズは1ライン単位(8ドット単位)で指定
SCREEN 0 のとき 1~40
SCREEN 1 のとき 1~80 の範囲で設定

〈垂直サイズ〉 縦のサイズはドット単位で指定
1~200の範囲で設定

G S I Z E (1, 8), 0

ボタン番号0のグラフィックパタンの大きさをキャラクターサイズに設定

G S I Z E (2, 16), 1

ボタン番号1のグラフィックパタンの大きさをキャラクターサイズの2倍に設定

G D S E T (グラフィック・データ・セット)

[機能] グラフィックパタンのデータセット

[書式 1] G D S E T <パタンデータ> , <ポインタ> , <パタン番号> [, <RGB面指定コード>]

[解説] 各パタンのグラフィックデータを1ライン単位でポインタで指定した位置へ格納する。

<パタンデータ> 2進数で表現したものを0~255 (16進数の0~FF)の数值で与える

<ポインタ> 左列より,上位ラインより数える。

<RGB面指定コード> RGBスプライト機能付加のパタンのみ有効

0~15の値で指定。1を指定した場合, B(青)面のみパタンデータをセットする。7を指定した場合, B(青), R(赤), G(緑), 3つの面に同時にデータをセットする (9ページの説明を参照)

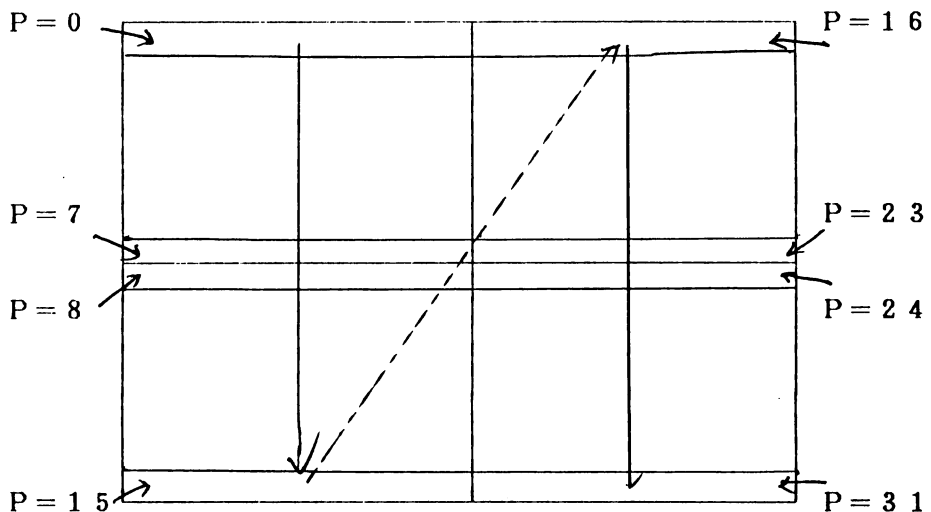
※以下,パタンデータ,ポインタのサンプルを示す。

[サンプル 1] 列1ライン,行8ドットのグラフィックパタン

P (ポインタ)		D (データ)	
P=0	□ □ □ □ □ □ □ ■	0 0 0 0 0 0 0 0	D=&H01 = 1
=1	□ □ □ □ □ □ ■ ■	0 0 0 0 0 0 1 1	=&H03 = 3
=2	□ □ □ □ □ ■ ■ ■	0 0 0 0 0 1 1 1	=&H07 = 7
=3	□ □ □ □ ■ ■ ■ ■	0 0 0 0 1 1 1 1	=&H0F = 15
=4	□ □ □ ■ ■ □ □ □	0 0 0 1 1 0 0 0	=&H18 = 24
=5	□ □ ■ ■ ■ ■ □ □	0 0 1 1 1 1 0 0	=&H3C = 60
=6	□ ■ ■ ■ ■ ■ □	0 1 1 1 1 1 1 0	=&H7E = 126
=7	■ ■ ■ ■ ■ ■ ■ ■	1 1 1 1 1 1 1 1	=&HFF = 255

(1の部分のパターン化される)

[サンプル 2] 列2ライン,行16ドットのグラフィックパタンポインタ



G D S E T (グラフィック・データ・セット)

[機能] グラフィックパタンのデータおよび、サイズの呼出し

[書式 2] G D S E T , <ポインタ> , <パターン番号> [, <RGB面指定コード>]

[解説] 各パタンのグラフィックデータおよびパターンサイズの内容を知りたいときに、この命令で呼出す。

次の3つの変数に各データが受け渡される。

第1変数 DT . . . 指定ポインタのパターンデータ (RGBで指定面が複数のときは各面の論理和が設定される)

第2変数 XZ . . . 列サイズ (水平サイズ)

第3変数 YZ . . . 行サイズ (垂直サイズ)

```
10 CLEAR , &HA000 : DT=0 : XZ=0 : YZ=0
20 MVSET 8,8, &HA000, &HBFFF
30 GSIZE (2,10), 0
40 GDSET 132, 2, 0
50 GDSET , 2, 0
60 PRINT DT; XZ; YZ
```

<RGB面指定コード>

(外形) G (緑) R (赤) B (青)

0	0	0	1	= 1: B面のみ指定
0	0	1	0	= 2: R面のみ指定
0	1	0	0	= 4: G面のみ指定
1	0	0	0	= 8: 外形作成面のみ指定
0	0	1	1	= 3: B面とR面を指定
0	1	0	1	= 5: B面とG面を指定
0	1	1	1	= 7: BRGの3面を指定
1	1	1	1	= 15: BRGの3面と外形作成面を指定

D G S E T (グラフィック・データ・セット)

[機能] RGBグラフィックパタンの外形データの設定

[書式 3] G D S E T , , <パターン番号>

[解説] RGBグラフィックパタンの外形をB (青) 面, R (赤) 面, B (緑) 面のグラフィックデータの論理和 (OR) で設定する

GDISP (グラフィック・ディスプレイ)

[機能] グラフィックパタンの表示

[書式 1] GDISP (〈水平位置〉, 〈垂直位置〉), 〈パターン番号〉 [, 〈色コード〉]

[解説] 任意のサイズのグラフィックパターンを指定した位置へ、指定した色で表示する。
(以前表示したものを消去した後に表示する。)

スプライト表示の優先順位は、パターン番号0が1番手前で番号値が大きくなるほど奥の方への表示となる。

座標位置 グラフィックパタンの左肩を基準とする。

《スクリーン座標系の場合》

- ・水平位置 SCREEN 0 のとき 0～312
- SCREEN 1 のとき 0～632の範囲で指定
- ・垂直位置 垂直200ライン のとき 0～199
- 垂直400ライン のとき 0～399の範囲で指定

色コード 通常の色コード (0～7) に準じる。 (特殊機能あり)

RGBスプライト付加パターンについては指定無効

・特殊機能 (スプライト付加パターンについての説明)

- 1) 色コードに &H08 = 8 を加えた値 8～15 のとき
スプライト表示の優先順位が反転する
- 2) 色コードに &H10 = 16 を加えた値 16～23 のとき
指定したパターン番号のスプライト表示停止を解除し、色コード (&H10 を引いた値) を新たにセットする。
- 3) 負の値のとき
指定したパターン番号のデータを負の値として格納 (これによりスプライト表示は停止する。)

```
GDISP ( 160, 100 ), 0
```

```
FOR X=0 TO 310: GDISP ( X, 100 ), 2, 6: NEXT
```

G Z O O M (グラフィック・ズーム)

[機能] グラフィックパタンの拡大表示

[書式 2] G D I S P (〈水平位置〉, 〈垂直位置〉), 〈色コード〉, 〈パターン番号〉, 〈水平拡大比〉, 〈垂直拡大比〉

[解説] グラフィックパターンを縦・横任意の比で拡大する。

G D I S P 文と違って, 以前表示したものは消去しないで次の表示を行うので同番号のパターンを同時にいくつも表示できる。動画シュミレーションのタイトル表示などに効果的。

この命令は, ほとんどメモリを必要としないので表示画面の大きさを越えない範囲で自由に拡大できる。

引数値 (水平位置, 垂直位置, パターン番号, 色コード) は G D I S P 文と同じ。

〈色コードの特殊機能について〉

通常の色コード (0 ~ 7) に 8 を加えた値 (8 ~ 1 5) を指定した場合, 画面への重ね表示 (他の部分を消去しない) となる。

```
G D I S P ( 1 6 0 , 1 0 0 ) , 0 , , 4 , 3
```

```
10 M V S E T
```

```
20 G D I S P ( 1 6 0 , 1 0 0 ) , 2 , 6 , 8 , 6
```

```
30 G D I S P ( 1 7 6 , 1 0 8 ) , 1 , 1 4 , 2 , 2
```

G D P O K E (グラフィック・データー・ポーク)

[機能] 連続表示用グラフィックデーターの作成

[書式3] GDISP (〈水平位置〉, 〈垂直位置〉), 〈パターン番号〉, [〈色コード〉]
, , , 〈ポインター〉

[解説] BASICの管理下でのグラフィックパタンの表示は、高速性に欠けるので、その欠点をカバーするために、いくつかのパタン表示のためのデーターをあらかじめ格納し、それらを連続的に表示する。このためのデーターを作成し、8バイト1組で指定した領域へ格納する。

〈水平位置〉, 〈垂直位置〉, 〈パターン番号〉, 〈色コード〉は "GDISP文" の引数に同じ。

〈ポインター〉格納領域のポインターで、0より始まりMAPで表示される(P)の値を越えない範囲で指定する。

M D P O K E (ミュージック・データー・ポーク)

[機能] 連続表示用ミュージックデーターの作成

[書式4] GDISP 〈SGレジスター番号〉, 〈データー〉, 〈SGレジスター番号〉
, 〈データー〉, 〈ポインター〉

[解説] 連続表示用格納領域にミュージックデーターを格納する。

G D P E E K (グラフィック・データー・ピーク)

[機能] グラフィックパタン (ミュージック) の連続表示

[書式5] GDISP (〈水平位置〉, 〈垂直位置〉), 〈スタートポインター〉, 〈ポインター
一数〉, 〈待ち時間〉, , 9999

[解説] BASICの管理から離れて、いくつかのグラフィックパタンを連続表示する。

《 GDPOKE (GDISP) 文で格納したデーターを8バイト単位で取り出し
GDISP文をコールする 》

水平および垂直位置は表示画面全体を平行移動させるためのオフセット値で通常は(0,0)に設定する。

待ち時間は、8バイト単位のデーターを取り出す間々の待ち時間で連続表示の際の速さを調節するためのもの。

GDGET (グラフィック・データー・ゲット)

- [機能] 表示画面上のグラフィックをパタンデーターとして格納
- [書式1] GDGET (〈水平位置〉, 〈垂直位置〉), 〈パタン番号〉 [, 〈RGB面指定コード〉]
- [解説] PSET, LINE, PAINT, GDISP, GZOOMなどで作成した表示画面上のグラフィックを, パタンデーターとして指定した番号のグラフィックパタン格納エリアに格納する。(位置は左肩を基準とする。)
- 引数値(水平位置, 垂直位置, パタン番号)はGDISP文に同じ。
- (PSET, LINE文で指定される位置はワールド座標系なので注意)
- 〈RGB面指定コード〉は0~15の範囲で指定, 省略すると15に設定される

```
GDGET (160, 100), 0, 3
```

```
10 SCREEN 0
20 LINE (320, 100) - (327, 107), PSET, 7, BF
30 GDGET (160, 100), 0
```

KJTR (カンジ・トランスファー)

- [機能] 漢字フォントをグラフィックパタンとする
- [書式2] GDGET 〈漢字コード〉, 〈パタン番号〉 [, 〈色コード〉]
- [解説] 漢字ROM (ROMカードを装着している場合) より, 指定した漢字コードの漢字フォント (32 バイト) のデーターを指定したパタン番号のグラフィックパタンデーター部へ転送・格納する。これにより任意の位置に漢字が表示できる。
- 〈色コード〉はRGBスプライトパタンのみ有効 0~7で指定, 省略は7 (白)

漢字はGDISP・GZOOM文により表示する。
グラフィックパタンのサイズは水平2ライン・垂直16ドット (キャラクターの縦横2倍の大きさ) に設定しておくこと。

```
10 GSIZE (2, 16), 0
20 GDGET &H3026, 0
30 GDISP (100, 100), 0
40 GDISP (160, 100), 0, , 4, 4
```

M V M A P (ムーブ・マップ)

[機能] システム (M o v i n g) の定数・メモリー番地の表示または設定

[書式] M V M A P <マップ表示> , <番地設定> , <座標指定モード> , <ラインモード>
[<色コード>]

[解説] <マップ表示> に " 1 " をセットしたとき、システムのメモリー番地、各設定値の内容が画面に表示される。(0 をセットした場合、表示されない)

<番地設定> に " 1 " をセットしたときは、次の2つの変数に次の値が設定される。

(0 をセットした場合、設定されない)

第1変数 S A . . . システムのスタート番地

第2変数 E A . . . システムのエンド番地 (ワークエリアを除く)

この2つの変数で設定されたメモリー番地の中に必要なすべてのデータが存在するので、この部分をマシンゴ・セーブしておくとの他のプログラムでそのまま使うことができる。

マシンゴ・ロード・オフセットで他の番地へロードして使う場合は M V S E T 文で各アドレス値および定数値を設定し直すこと。(C L E A R 文での B A S I C 文領域のメモリー番地の上限設定も間違いのないようにする。)

<座標指定モード> 0 または 1 をセットすることで G D I S P , G D G E T 文で設定する座標系が次のように変更される

0 をセットした場合 . . . スクリーン座標系

1 をセットした場合 . . . ワールド座標系

<ラインモード> 垂直ライン数を 2 0 0 ライン (B a s i c システムでサポート) , または 4 0 0 ライン (モノカラー) に変更する

2 0 0 をセットした場合、2 0 0 ラインに設定される

4 0 0 をセットした場合、4 0 0 ラインに設定される。その場合、次の引数で画面の色指定を行なう。省略したときは白に設定

(G D I S P 文の <色コード> 部でも色指定が行なわれる)

MAP 1

```
10 CLEAR 300, &HA000: SA=0: EA=0
```

```
20 MVNEW 10, 10, &HA000, &HBFFF
```

~

```
70 MVMAP 0, 1: PRINT HEX$(SA); " - "; HEX$(EA)
```

```
80 SAVEM "ヤマト", SA, EA, &HC000
```

```
90 END
```


システム定数 (S C) 部・細解

(DP・AD) (定数内容)

00,01	使用パターン数
02,03	スプライト機能付加パターン数
04,05	R G Bスプライト機能付加パターン数
06,07	グラフィック画面水平表示カラム数 (4 0 OR 8 0)
08,09	V R A M 1 ラインの飛びこしアドレス値
0A,0B	V R A M スタートアドレス
0C,0D	V R A M エンドアドレス
0E,0F	V R A M バックアドレス
10,11	機械語部スタートアドレス
12,13	機械語部エンドアドレス
14,15	システム定数格納スタートアドレス
16,17	パターン属性格納スタートアドレス
18,19	グラフィックパターンデータ格納スタートアドレス
1A,1B	スプライト表示用ワークエリア
1C,1D	グラフィックパターン消去用ワークエリア
1E,1F	グラフィックパターン表示用ワークエリア
20,21	ワークエリア最大サイズ
22,23	連続表示用データ格納スタートアドレス
24,25	システムで使用するメモリーの上限アドレス
26,27	連続表示用ポインター数
28,29	スプライト有無フラッグ・垂直ライン数フラッグ
2A,2B	初期セット判定フラッグ
2C,2D	パターン番号スタックポイント
2E	座標フラッグ (0 : スクリーン OR 1 : ワールド座標)
30-7F	作業域
B・M	B A S I C からシステムへのデータ受け渡し作業域

DP・AD (ダイレクト・ページ・アドレス)

B・M (マシンゴ領域の前部)

P D (パターン・データー) 部 属性・メモリー・ワークエリア 細解

A: パターン属性

00	水平 (横) サイズ [ライン単位]
01	垂直 (縦) サイズ [ドット単位]
02, 03	色コード・スプライトモード (FF, 00, 01)
04, 05	水平位置 (ライン単位, 右シフト数)
06, 07	垂直位置 (ドット単位)
08, 09	グラフィックパターン格納・アドレス
0A, 0B	パターン作成ワークエリア・アドレス
0C, 0D	スプライトパターン作成ワークエリア・アドレス
0E, 0F	R G Bスプライト外形ワークエリア・アドレス

1つのパターンについて
16バイト

B: グラフィックパターン・データー (水平サイズ Xライン, 垂直サイズ Yサイズのパターン)

00	1バイトのグラフィック・データー
↓	
X × Y - 1	*

モノカラーパタンの場合
X × Y バイト
R G Bフルカラーパタンの場合
4 × X × Y バイト
* 次ページ参照

C: スプライト表示ワークエリア (水平サイズ Xライン, 垂直サイズ Yサイズのパターン)

00	1バイト
↓	
(X + 1) × Y - 1	*

モノカラー・スプライトパタンの場合
(X + 1) × Y バイト
R G Bスプライトパタンの場合
4 × (X + 1) × Y バイト
* 次ページ参照

※ 1つのパターンにスプライト機能が定義されると C部のメモリーが追加必要となる。

R G BスプライトグラフィックパターンについてのB, C部の補足説明

B: グラフィックパターン・データ (水平サイズ Xライン, 垂直サイズ Yサイズのパターン)

00 ~ X × Y - 1	B面表示グラフィック・データ
00 ~ X × Y - 1	R面表示グラフィック・データ
00 ~ X × Y - 1	G面表示グラフィック・データ
00 ~ X × Y - 1	外形表示グラフィック・データ

4 × X × Y バイト

通常は G D S E T 文で上の R G B 面のデータの論理和 (O R) を設定

C: スプライト表示ワークエリア (水平サイズ Xライン, 垂直サイズ Yサイズのパターン)

00 ~ (X+1) * Y - 1	B (青) 面表示用ワークエリア
00 ~ (X+1) * Y - 1	R (赤) 面表示用ワークエリア
00 ~ (X+1) * Y - 1	G (緑) 面表示用ワークエリア
00 ~ (X+1) * Y - 1	パターン外形表示用ワークエリア

4 × (X + 1) × Y バイト

システム使用の際の注意 (補足)

(1) 拡張ベーシックで使用する数値は、0から始まるものと1から始まるものがあるので、設定の際、間違いのないように。

(A) 0から始まるもの

- ・ ボタン番号
- ・ ボタン表示位置
- ・ グラフィックボタンデータポインター
- ・ 連続表示用のポインター

(B) 1から始まるもの

- ・ ボタン数
- ・ スプライト機能付加ボタン数
- ・ RGBスプライト機能付加ボタン数
- ・ ボタンサイズ

(2) DISK・BASIC使用時とそうでない場合 (ROMベーシック) では、ベーシックプログラムの中間コードが異なるので、バージョンの違う機種間のベーシックプログラム転送はアスキー形式でセーブしたものをを使用すること。

(3) 拡張ベーシック命令使用の際のエラー表示として以下の3つがある。

(A) M V S E T コマンドによるシステムの設定がされていない

<< Moving Error >>

Undefined user function

(B) システムのメモリーがオーバーした

<< Moving Error >>

Out Of Memory

(C) 入力された数値が許容されている範囲からはずれている

<< Moving Error >>

Overflow

Moving (ムービング) の入力と実行について

本システムは日立のS1 (モデル10, 20, 30, 40) で実行できる拡張ベーシック・ツールである。プログラムは4 Kバイトのマシン語から成り、リロケートブルにつくられている。以下、例としてメモリー番地のC000~CFFF (16進数) の領域に格納・実行する場合を説明する。

(1) NEWON 0 を実行

すでに、このモードになっている場合はこの操作は不要

マシン語の打ち込みだけなら、NEWON 1 などの、ノーマルモードでもよい

(2) CLEAR ,&HA000 を実行

(3) マシン語モニターを起動し、メモリーダンプの通りに C000 番地よりマシン語を打ち込む。 (チェックサム機能の付いた他のツールを用いた方がスピーディーかつ確実である)

(4) 打ち込み終わったらベーシックにもどり、マシン語部をカセットテープまたはディスクへセーブする。

[操作] SAVEM "MVマシン-S1",&HC000,&HCFFF,&HC000 (RETURN)

格納が完了したら本システムを起動させる。(EXEC &HC000 を実行)

本システムを起動した場合、ベーシックコマンド待ちで終了してしまうので、ゲーム・シミュレーションなどで不便を感じる。そのような場合、起動アドレスを1つ大きくしてやるとプログラム中で起動しても拡張ベーシック設定後、続いて次のコマンドを実行する(EXEC &HC001)

メモリーマップが表示されたら、拡張ベーシック状態となり、6 個の新たな命令がベーシックコマンドとして追加されたことになる。

本システムはDISKベーシックを標準としているため拡張コマンドを最大6個までしか定義していない。したがって、多数の機能を使用するために1つのコマンドを多機能化している。区別は書式の違いによって行っている(GDSET, GDISP文など参照)

カセットテープまたはディスクからのロードおよび起動

(1) NEWON 0 を実行

(2) CLEAR ,&HA000 を実行

(3) LOADM "MVマシン-S1",,R を実行

(またはLOADM "MVマシン-S1":EXEC &HC000 を実行)

プログラム中では

10 CLEAR ,&HA000

10 LOADM "MVマシン-S1"

20 EXEC &HC001 (次のコマンドを引き続き実行する)

～