Personal Computer

MZ-2200 / 2000

QD BASIC LANGUAGE MANUAL (5Z007)





SHARP®

Personal Computer

MZ-2200/2000

QD BASIC Language Manual (5Z007)

© 1984 SHARP CORPORATION

ご　注　意

このマニュアルは、パーソナルコンピュータ MZ-2200／MZ-2000 のシステムソフトウェア BASIC インタープリタ 5Z007 に基づいて作成されています。

(1) 多目的パーソナルコンピュータ MZ-2200／MZ-2000 では、システムソフトウェアはすべてファイル形態のソフトウェアパック(カセットテープ、クイックディスク、フロッピーディスクなど)によってサポートされます。各システムソフトウェアおよび本書の内容は、改良のため変更することがありますので、ファイルバージョンナンバーには特にご注意されるよう、お願い致します。

(2) 本書には MZ-2000 標準 BASIC の解説は省略されています。内容については、MZ-2000 本体に付属の BASIC／MONITOR マニュアルをご参照ください。なお、MZ-2200 をお持ちの方には一部解説が重複している所がありますが、あらかじめご了承ください。
また GP-IB インターフィイスカードについては、「3. GP-IB ステートメント」をご参照ください。

(3) 本書は内容について十分注意し作成していますが、万一ご不審な点、お気付きの点がありましたら、もよりのシャープお客様ご相談窓口までご連絡ください。

(4) パーソナルコンピュータ MZ-2200／MZ-2000 (以後 MZ-2200 と書かれている場合は、MZ-2200／MZ-2000 と同じ意味です。)のシステムソフトウェアは、すべてシャープ株式会社のオリジナルソフトウェアであり、著作権法上の権利は当社が保有しております。システムソフトウェアならびに本書の内容を無断で転載あるいは複製することは禁止します。

目　　次

第１章　BASIC（5Z007）の概要 …… 7

1.1　BASIC インタープリタ 5Z007 の起動 …… 8
1.2　予約語 …… 9
1.3　初期設定値について …… 10
1.4　ディスプレイ画面の座標 …… 12
1.5　カラーコントロール …… 13
1.6　CRT ディスプレイ間の重ね合せ選択 …… 14
1.7　プライオリティ …… 15
1.8　画面反転表示 …… 16
1.9　MZ-80B との BASIC（5Z007）互換性 …… 16
1.10　ファイルコントロール …… 17
1.11　シーケンシャルアクセスファイルのコントロール …… 18
1.12　プログラムの CHAIN …… 21
1.13　カナ漢字変換機能 …… 22
1.14　プリンタの接続 …… 25

第２章　拡張・追加コマンド、ステートメント …… 27

2.1　基本コマンド …… 28
　2.1.1　ON ERROR GOTO …… 28
　2.1.2　IF ERN …… 28
　2.1.3　IF ERL …… 29
　2.1.4　RESUME …… 29
　2.1.5　DELETE …… 30
　2.1.6　PSEL …… 31
　2,1.7　RENUM …… 32
　2.1.8　SEARCH …… 33
　2.1.9　SEARCH/P …… 34
　2.1.10　TABSET …… 34
　2.1.11　CHR$ …… 35
　2.1.12　POKE …… 35
2.2　ファイルコマンド（クイックディスク） …… 36
　2.2.1　DIR …… 36
　2.2.2　DIR/P …… 37
　2.2.3　SAVE …… 37
　2.2.4　LOAD …… 38
　2.2.5　APPEND …… 38

2.2.6 RUN ……39
2.2.7 CHAIN ……40
2.2.8 WOPEN # ……41
2.2.9 PRINT # ……41
2.2.10 ROPEN # ……42
2.2.11 INPUT # ……42
2.2.12 KILL ……43
2.2.13 CLOSE ……43
2.2.14 INIT ……44
2.2.15 IF EOF (#) THEN ……44
2.3 カラーコマンド……45
2.3.1 CCOLOR ……45
2.3.2 COLOR ……46
2.3.3 GRAPH ……47
2.3.4 CONSOLE ……48
2.3.5 SET ……48
2.3.6 RESET ……49
2.3.7 LINE ……50
2.3.8 BLINE ……51
2.3.9 PATTERN ……52
2.3.10 PAINT ……53
2.3.11 BOX ……54
2.3.12 CIRCLE ……55
2.4 漢字コマンド……57
2.4.1 JTRANS ……57
2.4.2 KCHR$ ……58
2.4.3 KINPUT ……59
2.4.4 KPATTERN ……60
2.4.5 KPRINT/P ……61
2.5 RS-232C コマンド ……62
2.5.1 RSMODE ……62
2.5.2 RSO ……63
2.5.3 RSI ……63

第3章 GP-IB ステートメント ……65

3.1 プログラム作成上の注意……66

3.2 GP-IB ステートメント ……68
3.1.1 ICL ……68
3.2.2 REN ……68
3.2.3 LCL ……69
3.2.4 LCL n……69
3.2.5 LLO ……70
3.2.6 DCL ……70
3.2.7 DCL n ……71
3.2.8 TRG ……71
3.2.9 PCT ……72
3.2.10 WRT ……73
3.2.11 RED ……74
3.2.12 WRT/ ……75
3.2.13 RED/ ……76
3.2.14 CMDW ……77
3.2.15 CMDR ……78
3.2.16 ON SRQ ……79
3.2.17 SPOL ……79
3.2.18 PPC ……80
3.2.19 PPOL ……81
3.2.20 PPU ……82
3.2.21 GPIBM ……82
3.2.22 EOIW ……82
3.2.23 EOIR ……82

第4章 BASIC（5Z007）のまとめ ……83

4.1 コマンド……84
4.2 ファイルコントロール文……86
4.3 データファイルコントロール文……86
4.4 エラー処理文……87
4.5 カセットテープ・データファイル入出力文……88
4.6 代入文……88
4.7 入出力文……88
4.8 ループ文……89
4.9 分岐文……89
4.10 定義文……90

4.11 注釈文とコントロール文……91
4.12 ミュージックコントロール文……92
4.13 カラーコントロール文……92
4.14 漢字コントロール文……97
4.15 機械語プログラムコントロール文……98
4.16 プリンタ・コントロール文……99
4.17 I/O 入出力文……99
4.18 数値関数 ……100
4.19 ストリングコントロール関数 ……101
4.20 タブ関数 ……101
4.21 算術演算子 ……102
4.22 比較・論理演算子 ……102
4.23 その他のシンボル ……102

第5章 付 録 ……105

5.1 QDCOPY ユーティリティ ……106
5.2 DELETE ユーティリティ ……108
5.3 TRANS ユーティリティ ……110
5.4 ASCII コード表 ……114
5.5 エラーメッセージ一覧表 ……116
5.6 起動時のメモリマップ ……118
5.7 JIS 漢字コード表 ……119
5.8 クイックディスク取扱い上の注意 ……131
5.9 索引 ……132

BASIC(5Z007)の概要

Chapter 1

この章は、BASICインタープリタ（5Z007）の特長、ファイルコントロールの概要について解説を行っています。はじめに、BASIC（5Z007）の起動方法が説明されます。

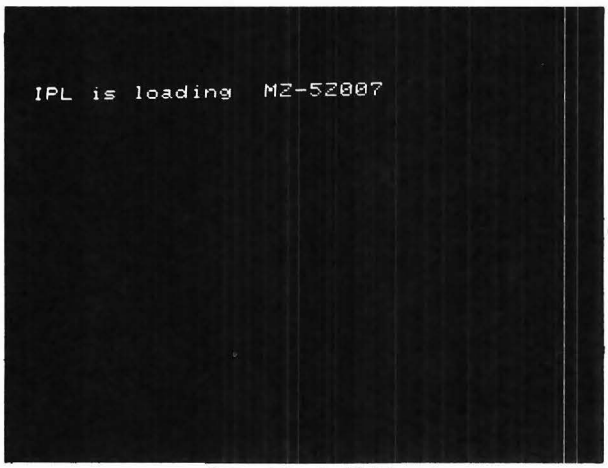
BASIC（5Z007）の持つ新規コマンド、ステートメントの文法解説は第2章に、BASIC（5Z007）の持つ全てのコマンド、ステートメント、関数、オペレータのまとめは第4章です。

1.1 BASIC インタープリタ 5Z007 の起動

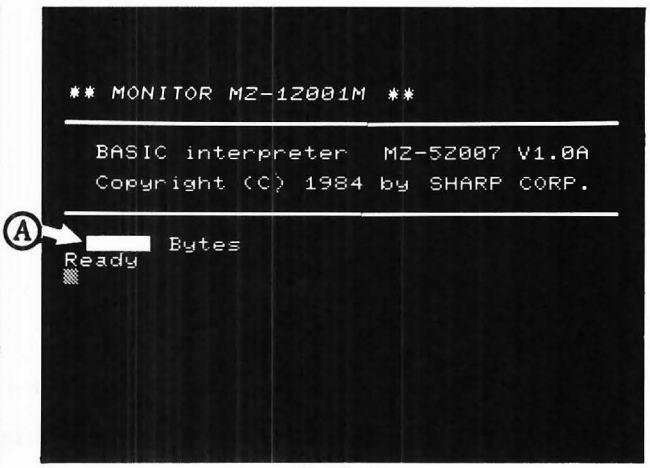
BASIC インタープリタ 5Z007 （および MONITOR MZ-1Z001M）はマスターディスク中にあるので、それを走らせるにはまず、IPL によってイニシャル・ローディングを行わなければなりません。

イニシャル・ローディングは簡単に実行することができます。MZ-2200（電源 OFF）にクイックディスクドライブを接続した状態で、ドライブにマスタークイックディスクをセットします。そして、／（スラッシュ）キーを押しながら MZ-2200 の電源を ON にすると、BASIC（5Z007）が自動的に起動されます。もし、うまく起動できなかった場合は再度実行してください。

図の左側は、BASIC（5Z007）をローディング中であることのメッセージ表示を、右側は、ローディングを終了して、BASIC インタープリタ 5Z007（および MONITOR MZ-1Z001M）が起動し、カーソル点滅によって BASIC コマンドレベルになったことを、それぞれ示しています。



（ローディング中）



（起動時）

BASICテキスト〝AUTO RUN〟が自動的に実行されること

上記のオペレーションのうちに含まれている、BTX〝AUTO RUN〟の実行を解説します。BASIC（5Z007）がローディングされ、起動時の画面のⒶの位置にテキストエリアのバイトサイズが表示されると、再び、クイックディスクドライブがアクセスされるのに気が付かれたと思いますが、BASIC（5Z007）は、イニシャルローディングを終えた時、自動的に、

RUN 〝AUTO RUN〟

コマンドを実行します。即ち、〝AUTO RUN〟というファイル名のプログラムテキストを同じマスターディスクからロードし、その先頭から実行します。サポートされたマスターディスク上には、デファイナブルファンクションキーを定義するプログラムをこのファイル名で登録しています。また、このプログラムの最後に、NEW 文を置いているので実行後、〝Ready〟を表示しカーソル点滅となる前にテキストは消去されているのです。(一度、LOAD〝AUTO RUN〟を実行してリストを調べてみてください。)

もし、BASIC（5Z007）の起動に続けて、あるプログラムをスタートさせたい場合、そのプログラムのファイル名を〝AUTO RUN〟としてマスターディスク上にセーブしておけばよいことになります。

1.2 予約語

BASIC文は予約語（reserved words）——キーワードとも呼ばれます——と、そのオペランド、セパレータ、データによって構成されます。予約語は、BASICインタープリタがそれを解釈し決められた機能を行う特別な語であり、コマンド、ステートメント、関数がそれに相当します。予約語は、特殊な機能を実行させるために決められている語なので、プログラマが、変数名、配列名などに用いることができません。表は、BASIC（5Z007）の全ての予約語をアルファベット順に並べて示しています。（予約語の右の数字は参照ページを示しています。）

A
ABS ……100
APPEND ……38
ASC ……101
ATN……100
AUTO ……85
B
BLINE ……51
BOOT ……86
BOX ……54
C
CCOLOR ……45
CHAIN ……40
CHANGE ……91
CHARACTER$ 101
CHR$ ……35
CIRCLE……55
CLOSE ……43
CLOSE # ……86
CLOSE/T ……88
CLR ……91
CMDR ……78
CMDW ……77
COLOR ……46
CONSOLE ……48
CONT ……86
COPY/P ……99
COS ……100
CSRH……91
CSRV……91
CURSOR ……91
D
DATA ……89
DCL ……70
DEF FN ……90
DEF KEY ……90
DELETE ……30
DIM ……90
DIR……36
DIR/P ……37
E
END ……91
EOIR ……82
EOIW……82
ERL ……29
ERN ……28
ERROR ……28
EXP ……100
F
FAST……91
FOR ……89
G
GET ……89
GOSUB ……89
GOTO ……89
GPIBM ……82
GRAPH……47
I
ICL ……68
IF ……90
IMAGE/P ……99
INIT ……44
INP……99
INPUT ……88
INPUT # ……42
INPUT/T ……88
INT ……100
J
JTRANS ……57
K
KCHR$ ……58
KILL ……43
KINPUT ……59
KLIST ……86
KPATTERN …60
KPRINT/P ……61
L
LCL ……69
LEFT$……101
LEN ……101
LET ……88
LIMIT ……98
LINE ……50
LIST ……85
LIST/P ……85
LLO ……70
LN ……100
LOAD……38
LOAD/T ……84
LOG ……100
M
MID$ …… 101
MON ……86
MUSIC ……92
N
NEW ……86
NEXT ……89
O
ON ……90
OUT ……99
P
PAGE/P ……99
PAINT ……53
PATTERN ……52
PCT ……72
PEEK……98
POINT ……96
POKE……35
POSH……96
POSITION ……94
POSV……96
PPC ……80
PPOL……81
PPU ……82
PRINT ……88
PRINT # ……41
PRINT/P ……99
PRINT/T ……88
PSEL ……31
R
READ……89
RED ……74
RED/ ……76
REM ……91
REN ……68
RENUM ……32
RESET ……49
RESTORE ……89
RESUME ……29
RETURN ……89
REW ……91
RIGHT$ ……101
RND……101
ROPEN # ……42
ROPEN/T ……88
RSI……63
RSMODE ……62
RSO ……63
RUN ……39
S
SAVE……37
SAVE/T ……84
SEARCH ……33
SEARCH/P ……34
SET ……48
SGN ……100
SIN ……100
SIZE ……91
SPACE$ ……101
SPOL ……79
SQR ……100
SRQ ……79
STEP……89
STOP……91
STR$ ……101
STRING$ ……101
T
TAB……101
TAB SET……34
TAN……100
TEMPO……92
THEN ……90
TI$ ……91
TO ……89
TRG ……71
U
USR ……98
V
VAL……101
VERIFY ……85
W
WOPEN #……41
WOPEN/T ……88
WRT ……73
WRT/ ……75

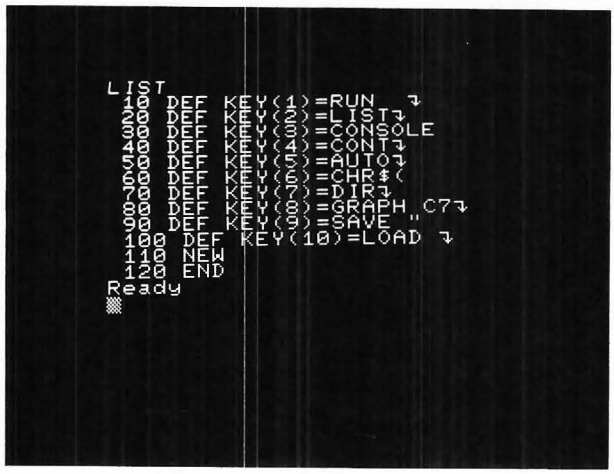
表1—1　BASICインタープリタ5Z007の全ての予約語

1.3 初期設定値について

BASIC インタープリタ 5Z007 が IPL によって起動した時の、システム変数等のデフォルト値はそれぞれ次のように設定されます。

■ キーボード関係

1) 動作モード：ノーマルモード
2) 小文字の入力はノーマルモードのシフトポジション
3) デファイナブルファンクションキーは、BTX" AUTO RUN "によって次のように設定されます。



■CRTディスプレイ関係

1) キャラクタディスプレイモード： ノーマル（バックグラウンド：黒）
2) キャラクタ表示桁数： 40キャラクタ／行
3) キャラクタ表示スクローリングエリア： 最大（第0行から第24行）
4) グラフィックディスプレイ入力モードページ： ページ1
   グラフィックディスプレイ出力モードページ： 全ページともOFF
   ポジションポインタ： POSH＝0、POSV＝0
5) グラフィックディスプレイリゾリューションモード：320×200ドット／画面

■カラーCRTディスプレイ関係

プライオリティ……キャラクタ優先
キャラクタカラー……白色
バックグランドカラー……黒色
} CCOLOR@7, 0のモードに設定

グラフィックカラー……青色
グラフィックディスプレイ出力モードページ……全ページともOFF
カラー機能……W0モード
} COLOR1, O0, W0のモードに設定

■白黒CRTディスプレイ関係

カラーグラフィックデータの白黒画面への重ね合わせを行う。(CONSOLE Mのモードに設定)

■内蔵時計

TI$＝"000000"で初期化してスタートする。

■音楽機能

1) テンポのデフォルト値： 4（中ぐらいのテンポ、Moderato）

2) 音長のデフォルト値 ： 5（4分音符、♩）

■その他

1) 配列変数は全て未定義

2) BASICテキストエリアの上限： $FFFF番地（即ちLIMIT MAX 状態）

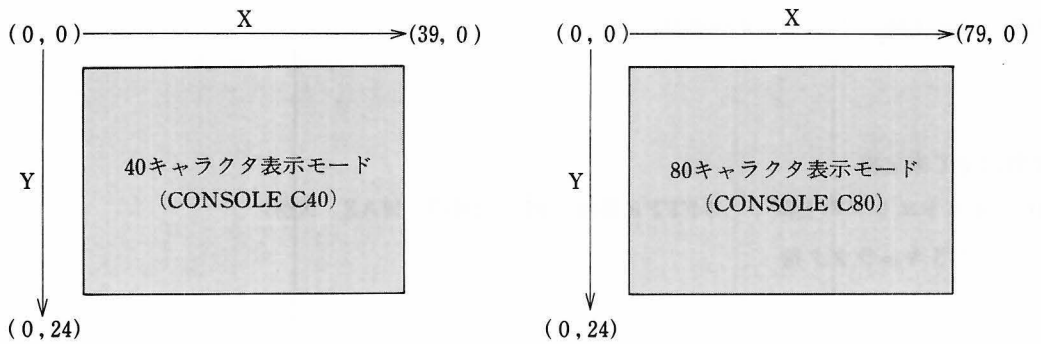
3) タブセット：5キャラクタ毎

1.4 ディスプレイ画面の座標

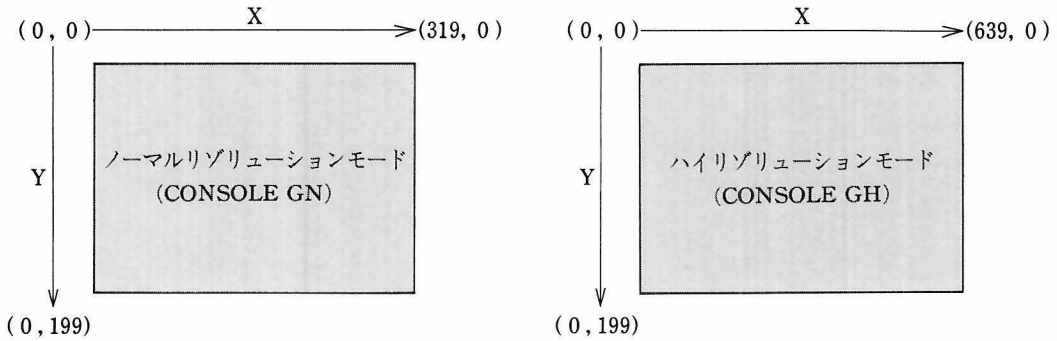
各表示モードにおけるディスプレイ画面上の座標について説明します。

キャラクタ表示におけるディスプレイ画面上の座標を上図に、カラーグラフィック表示におけるディスプレイ画面上の座標を下図に示します。

白/黒CRTディスプレイとカラーCRTディスプレイの座標は同一関係にあります。



キャラクタ表示における画面上の座標



グラフィック表示における画面上の座標

たとえば、ハイリゾリューションモード（640×200ドット/画面）におけるグラフィック座標を指定する場合、画面右下のコーナーであれば（639, 199）となります。このように座標指定とは、X座標、Y座標を指定することです。

例　カラーCRTディスプレイ画面中央に黄色のドットを表示させる。

```
10  PRINT CHR$(6):CONSOLE  GH
20  COLOR, O7, W0
30  SET[6] 320, 100
40  END
```

座標指定

1.5 カラーコントロール

■カラーコード（オペランド：mとn）

このBASICでは、黒を含めて計8色のカラー表示が可能です。表に示すカラーコードにより、各コマンド内で色指定を行います。

カラーコード	0	1	2	3	4	5	6	7
色	黒	青	赤	紫	緑	水色	黄	白

カラーコード

カラーグラフィック表示は、V-RAMグラフィックBLUE、RED、GREENの各エリアにデータを書きこむことにより行われます。これらのV-RAMグラフィックの名称と、BASIC（5Z007）使用時の名称の関係を図に示します。



V-RAM グラフィックの名称

これらの3ページを合成することにより上記8色が画面上に表示されます。たとえば、画面中央のドットに対応した、V-RAMグラフィック上の番地において、BLUEとGREENのページがセットされ、REDのページがリセットされた状態である時、画面上のそのドットには水色が表示されます。(COLOR, 07のモードに設定されているものとします。)

キャラクタおよびバックグランド（背景色）は、これらのグラフィックとは関係なく、その色を指定することができます。ただしキャラクタの色は1画面単位での指定となります。(キャラクタ単位での色指定を行うことはできません。)

■色の重ね合わせ（オペランド：Wa）

カラーグラフィック用のコマンドには、それぞれ色の処理方法を指定するオペランドWa（W0とW1）が用意されています。各モードの機能について説明します。

W0モード　強制的に色の変更を行うモードで、現在どの色が表示されているかにかかわらず、その時点で設定されているカラーコードに従って命令を実行します。たとえば、すでに緑色にセットされているドットに対し、W0モードで赤をセットしにいきます。

```
SET[2,W0]160,100
```

この命令実行後この座標は、赤色にセットされます。

W1モード　色の重ね合わせを行うモードで、現在セットされている色と、その時点で設定されているカラーコードを重ね合わせます。たとえばすでに緑色にセットされているドットに対しW1モードで赤をセットしにいきます。

SET[2, W 1]160, 100

この命令実行後このドットは緑色と赤色が重ね合わされて、黄色となります。

Waの記述を各コマンド内で省略しますと、COLOR文でのWa設定モードに従います。

COLOR文でもWaの設定がされていない場合はW 0モード（初期設定値）となります。

RESET, BLINE文中でのW 1 モードの機能は上記と異なっています。各コマンドの解説の項を参照ください。

なお、キャラクタとグラフィックとの色の重ね合わせは行えません。

1.6 CMTディスプレイ間の重ね合わせ選択

白黒CRTディスプレイとカラーCRTディスプレイとのキャラクタおよびグラフィック表示の重ね合わせの有無を、表のように選択することができます。(キャラクタ、グラフィック共表示データがある場合での組み合わせです。)

組み合わせ	白/黒CRTディスプレイ		カラーCRTディスプレイ	
	キャラクタ表示	グラフィック表示	キャラクタ表示	グラフィック表示
1	○	×	×	×
2	○	×	×	○
3	○	×	○	×
4	○	×	○	○
5	○	○	×	○
6	○	○	○	○

(初期設定モード)

キャラクタおよびグラフィック表示の重ね合わせ選択例

組み合わせ 1　バックグランドとキャラクタの色を同じ色に指定し、プライオリティをグラフィックに指定します。そしてグラフィック出力ページモードを全ページOFFとします。

(例)　10　CCOLOR 1, 1 : COLOR @, O0

組み合わせ 2　バックグランドとキャラクタの色を同じ色に指定し、プライオリティをグラフィックに指定します。そしてグラフィック出力ページモードを全ページONとし、白/黒CRTディスプレイへのグラフィックデータの重ね合わせを禁止します。

(例)　10　CCOLOR 1, 1 : COLOR @, O7 : CONSOLE P

組み合わせ 3　バックグランドとキャラクタの色をそれぞれ異なった色に指定し、グラフィック出力ページモードを全ページ OFF とします（BASIC 起動時、このモードに設定されます。）

(例)　10　CCOLOR 4, 1 : COLOR, O0

組み合わせ 4　バックグランドとキャラクタの色をそれぞれ異なった色に指定し、グラフィック出力ページモードを全ページＯＮとします。そして白/黒CRTディスプレイへのグラフィックデータの重ね合わせを禁止します。（プライオリティは任意）

(例)　10　CCOLOR　@　4,1：COLOR,O7：CONSOLE　P

組み合わせ 5　バックグランドとキャラクタの色を同じ色に指定し、プライオリティをグラフィックに指定します。そして、グラフィック出力ページモードを全ページONとし、白/黒CRTディスプレイへのグラフィックデータの重ね合わせを許可します。

(例)　10　CCOLOR　1,1：COLOR　@,O7：CONSOLE　M

組み合わせ 6　バックグランドとキャラクタの色をそれぞれ異なった色に指定し、グラフィック出力ページモードを全ページＯＮとします。そして、白/黒CRTディスプレイへのグラフィックデータの重ね合わせを許可します。（プライオリティは任意）

(例)　10　CCOLOR　4,1　：COLOR　@,O7：CONSOLE　M

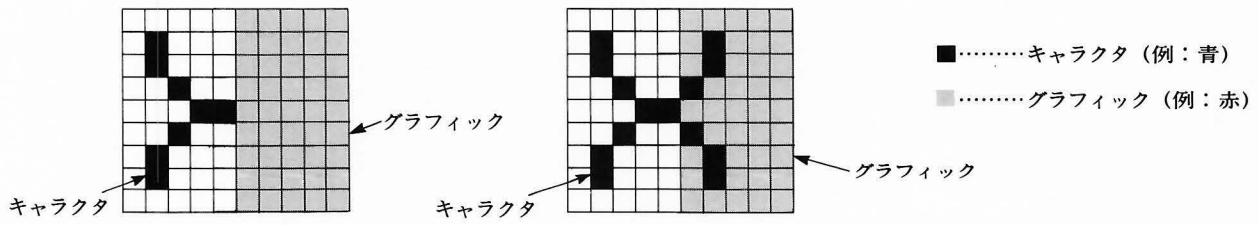
1.7　プライオリティ

プライオリティとは、キャラクタとグラフィックがカラーCRTディスプレイの画面上で重なって表示される場合、どちらを優先して表示させるかを決めるものです。

プライオリティをグラフィックに指定するにはCOLOR　@　を実行させます。

プライオリティをキャラクタに指定するにはCCOLOR　@　を実行させます。

図はプライオリティ指定後の表示状態を示しています。



プライオリティをグラフィックに指定　　プライオリティをキャラクタに指定

プライオリティ指定後の表示状態

1.8 画面反転表示

CONSOLE Rを実行しますと、白/黒CRTディスプレイ画面はリバース表示となりますが、カラーCRTディスプレイ画面は、バックグランド黒一色の表示状態となり、キャラクタ、グラフィックとも表示されなくなります。

1.9 MZ-80B との BASIC（5Z007）互換性

MZ-80B で作成された BASIC テキストプログラムは、MZ-2200 の BASIC インタプリタ MZ-1Z001、MZ-1Z002 および DISK BASIC インタプリタ MZ-2Z001 のどちらでも使用可能です。ただしハードウェアに直接関係した命令（LIMIT、PEEK、POKE、USR、INP および OUT）をプログラム内で使用している場合には、修正を要します。（BASIC テキストエリアおよび各ポートアドレスが MZ-80B と MZ-2200 とでは異なる為）または POINT 命令についてはP.96を参照ください。

また CONSOLE Cn、LIST、LIST/P、DIR、DIR/P 命令において、実行後のグラフィック出力ページモードの処理方法が下記の通りに異なっています。

MZ-80Bの場合

上記命令実行時、グラフィックは画面より消えますが、その後GRAPH I1 (I2、C、F でも同様) を実行しますと、上記命令実行前に設定されたグラフィック出力ページモードが復帰しグラフィックが再び画面上へ表示されます。

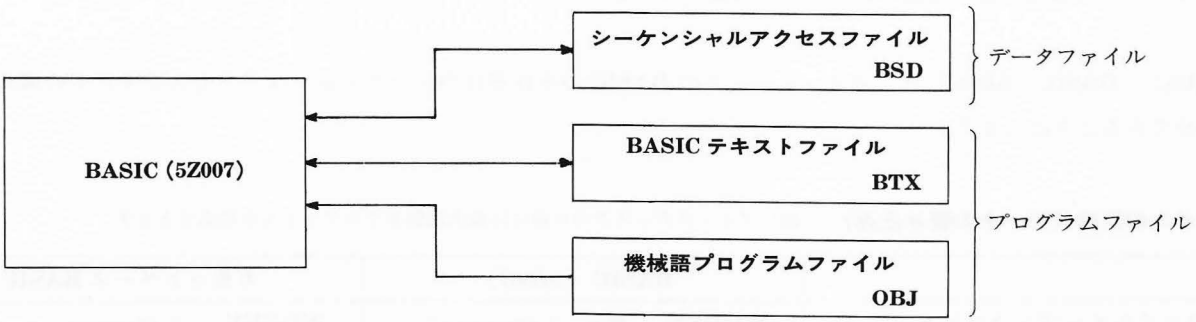
MZ-2200の場合

上記命令実行時、グラフィック出力ページモードが全ページOFF(GRAPH O0またはCOLOR, O0のモード) となります。したがってこの後GRAPH I1 (I2、C、Fでも同様) を実行しても、グラフィックは画面上へ表示されません。

たとえばCONSOLE C80を実行後グラフィックを画面上に表示させる場合は、再度GRAPH O*b*またはCOLOR, Onを実行する必要があります。

1.10 ファイルコントロール

BASIC（5Z007）で作成されるファイルは、シーケンシャルアクセスファイル（BSD）のデータファイルと、BASICテキスト（BTX）のプログラムファイルの2種類があります。もう一種類の機械語プログラムファイル（OBJ）は、MONITORプログラムあるいはFDOSなどで作成したものをクイックディスク上へ登録したもので、それは、単独で走らせるか、BASICの機械語エリアへ乗せて、BASICテキストとリンクして使うなどの目的をもったファイルです。したがって、BASICで利用することはできても、BASICで作成したり内容を変更したりするファイルではありません。



ファイルコントロールコマンドの解説にあたって、はじめに、データファイルの作成方法、利用方法、特徴を説明し、つづいて、プログラムファイルのCHAINコマンドの使い方を説明して行きます。

1.11 シーケンシャルアクセスファイルのコントロール

シーケンシャルアクセスファイルとは、データの登録または呼び出しが、シーケンシャルアクセス形式で行われるデータファイルのことです。**シーケンシャルアクセス(sequential access)**形式とは、データのアクセスを、先頭から順番に行う形式を意味するものです。

すでに、MZ-1Z001シリーズのBASICで、カセットファイル上へ、データファイルを作成する方法を説明していますが、BASIC（5Z007）でのシーケンシャルアクセスは、ちょうどそれと同じことを、クイックディスクに対して行うものです。もちろん、はるかに高速アクセスができ、さらにいくつかの新しいファイルコントロールコマンドを用いることによってファイル管理の面でも幅のある使用が可能となります。

最初に、BASIC（5Z007）と、カセットベースのBASICのそれぞれのシーケンシャルアクセスコマンドの構成を、対比させてみることにします。

ファイルの登録（データの書き込み）　注）クイックディスクの片面には最大32個までのファイルが作成できます。

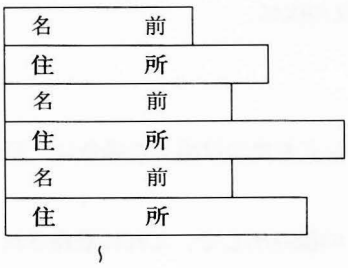
	BASIC（5Z007）	カセットベース BASIC
ファイルオープンコマンド	WOPEN　#*n*, "*filename*"	WOPEN　"*filename*"
データ書き込みコマンド	PRINT　#*n*, *data*	PRINT／T　*data*
ファイルクローズコマンド	CLOSE　#*n*	CLOSE
キャンセルコマンド	KILL　#*n*	―――

ファイルの呼び出し（データの読み出し）

	BASIC（5Z007）	カセットベース BASIC
ファイルオープンコマンド	ROPEN　#*n*, "*filename*"	ROPEN　"*filename*"
データ読み出しコマンド	INPUT　#*n*, *variable*	INPUT／T　*variable*
ファイルクローズコマンド	CLOSE　#*n*	CLOSE
ファイルエンドの検出	IF EOF（#*n*）THEN	―――

それぞれのコマンドの構成を比べてみて、ほとんど、一対一に対応していることがだかると思いますが、BASIC（5Z007）のコマンドにはいつも、#nという要素が含まれていますがこれは、**ロジカルナンバ(logical number)**と呼ばれる番号で、BASIC（5Z007）のファイルアクセスは、常にこの番号を指定して行わなければなりません。

簡単な例として、人名とその自宅の住所とをシーケンシャルアクセスファイルに登録することを考えてみます。手元にある住所録、たとえばあなたが同窓会の幹事をしているとしたらそのクラス全員の名簿を片っ端から全部ファイルにとっておくのです。たとえば、次のようなファイルとして……。



filename＝"3−H、クラスメイボ"

名前も住所も長さをバラバラに書いたのは、**一般にシーケンシャルアクセスで登録されるデータは固定長ではなく、データによって長さが異る**からです。今の場合のように、データが、1まとまりのものとして扱われ(3年H組という1クラス全体)、また住所のようにたいてい32バイトでは足りなく、長さもまちまちの場合、シーケンシャルアクセスファイルに登録するのが向いています。

INPUT文で、名前と住所とを交互にストリング変数に代入して、1人ずつファイルに登録して行き、全部で50人分の住所録"3−H、クラスメイボ"を作成し、次に作成されたファイルを読み出して、10人分ずつ名前と住所とをCRTディスプレイに表示していくプログラムは、次のようにして作ることができます。

〔書き込み〕

```
100 WOPEN #3,"3−H,クラスメイボ"
110 FOR P=1 TO 50
120 INPUT "ナマエ=";NA$
130 INPUT "ジュウショ=";AD$
140 PRINT #3,NA$,AD$
150 NEXT P
160 CLOSE #3
```

〔読み出し〕

```
200 ROPEN #4,"3−H,クラスメイボ"
210 FOR P=1 TO 5 :FOR Q=1 TO 10
220 INPUT #4,NA$,AD$
230 PRINT NA$ :PRINT AD$
240 NEXT Q
250 PRINT "STRIKE ANYKEY"
260 GET X$:IF X$="" THEN 260
270 NEXT P
280 PRINT "END"
290 CLOSE #4
```

■ファイルエンドのみつけ方

ファイルからデータを順番に読み出して行って、登録されているデータを越えた場合にどうなるか？……この場合は、エラーは発生せず、読み出す変数には0か空がセットされることになるのですが、特別な関数、EOF（#*n*）というものがあり、これがファイルエンドを捉えます。**EOF(#*n*)は、INPUT#コマンドでデータの読み出しを行った時、そのファイルエンドに来ていたら条件が真となります。**したがって、INPUT#文の次に、

IF EOF (#*n*) THEN

コマンドを置いておくと、EOF（#*n*）が"真"、すなわちファイルエンドを見つけ出した場合に、THEN以下が実行されることになります。

〔問　題〕　前ページのプログラム例で、"3—H、クラスメイボ"の読み出しで、これに登録されている人数が不明だとして、10人分ずつファイルエンドまで読み出し、表示するプログラムに変更しなさい。

〔例　解〕　たとえば次のようなプログラムが考えられます。

```
300 ROPEN #5,"3—H, クラスメイボ"
310 FOR I=1 TO 10
320 INPUT #5, NA$, AD$
330 IF EOF (#5) THEN 400
340 PRINT NA$:PRINT AD$
350 NEXT I
360 PRINT "STRIKE ANYKEY"
370 GET X$:IF X$="" "THEN 370
380 GOTO 310
400 CLOSE #5
410 PRINT "FILE END":END
```

〔問　題〕　INPUT文でキー入力したストリングをBSDファイルに登録する。ただし、ファイルをクローズする場合"CLOSE"を、キャンセルする場合"KILL"をキー入力するものとします。

〔例　解〕

```
100 WOPEN #30,"SEQ DATAS"
110 INPUT "DATA=";A$
120 IF A$="CLOSE" THEN CLOSE #30:END
130 IF A$="KILL" THEN KILL #30:END
140 PRINT #30,A$:GOTO 110
```

1.12 プログラムの CHAIN

データファイルコントロール命令に続いて、プログラムファイルのコントロールについて説明いたします。ここで説明される命令は、CHAIN コマンドです。このコマンドを用いると、プログラムをジョブ単位でクイックディスク上に登録しておいて、プログラムを走らせながら、別のプログラムを呼び出してそれにコントロールを移すことができます。つまり、プログラムを、クイックディスク上に登録されているプログラムに接続（CHAIN）できるのです。

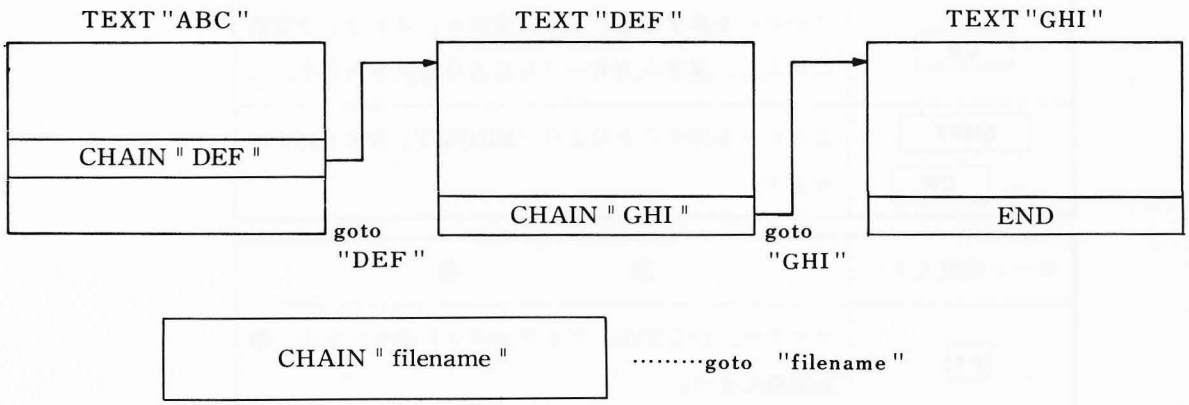
CHAIN コマンドの形式はたとえば次のようになっています。

```
700 CHAIN "TEXT 2"
```

このステートメントは、現在テキストエリア内にあるプログラムを NEW して(ただし変数の値は保持する)、クイックディスク上に登録されている、"TEXT 2" というファイル名のテキストを、オーバレイ（テキストエリアへ重ねて読み出すこと）してそのテキストの先頭へコントロールを移しなさい、という意味です。

このステートメントを実行させると、現在走っている BASIC テキストからコントロールを離れて、新たにテキスト "TEXT 2" を読み出して来てその先頭へコントロールが移ります。**プログラムの CHAIN が実行される時、変数の値と、DEF FN で定義した関数とは、CHAIN 先のプログラムへ受渡されます。**

CHAIN コマンドの機能は、**「goto "filename"」**として捉えることができます。



1.13 カナ漢字変換機能

BASIC (5Z007) のカナ漢字変換機能は、JIS 漢字コードを使用してカナ漢字変換を行うことができるものです。
カナ漢字変換機能は「KINPUT」命令 (P.59 参照) を実行することにより起動します。
次に、カナ漢字変換機能を起動した場合の、操作方法等について、プログラム例をまじえ説明します。

「KINPUT」命令実行時において、次に示すキーについて、その働きを説明します。

注) この機能を実行する場合はオプションの漢字 ROM ボード MZ-1R13 が必要です。

編集キー	説明
BREAK	カタカナ、ひらがなを、このキーを押すことにより、漢字に変換します。
CLR HOME	漢字選択モードにおいてこのキーを押すと、漢字選択モードがキャンセルされ、カナ入力モードにもどります。
INST DEL	このキーを押すことにより、カナ入力カーソルの左側に表示されている文字、1文字を消去し、カナ入力カーソルを1文字分左へ移動します。
SHIFT + CLR HOME	このキーを押すことにより、漢字入力カーソルの左側に表示されている文字、1文字を消去し、漢字入力カーソルを1文字分左へ移動します。
CR	このキーを押すことにより、文字を、ストリング変数に代入し、漢字入力カーソルにより表示されます。
SHIFT + CR	このキーを押すことにより「KINPUT」命令を終了させます。

モード切換えキー	説明
カナ	カタカナ、ひらがな、アルファベットのモードに、順次切換えます。
GRPH	グラフィック、アルファベットのモードに順次切換えます。
SFT LOCK	アルファベット小文字、アルファベット大文字のモードに、順次切換えます。

(注1) JIS第1水準漢字コードに規定されていない非漢字は「KINPUT」命令で入力することはできません。

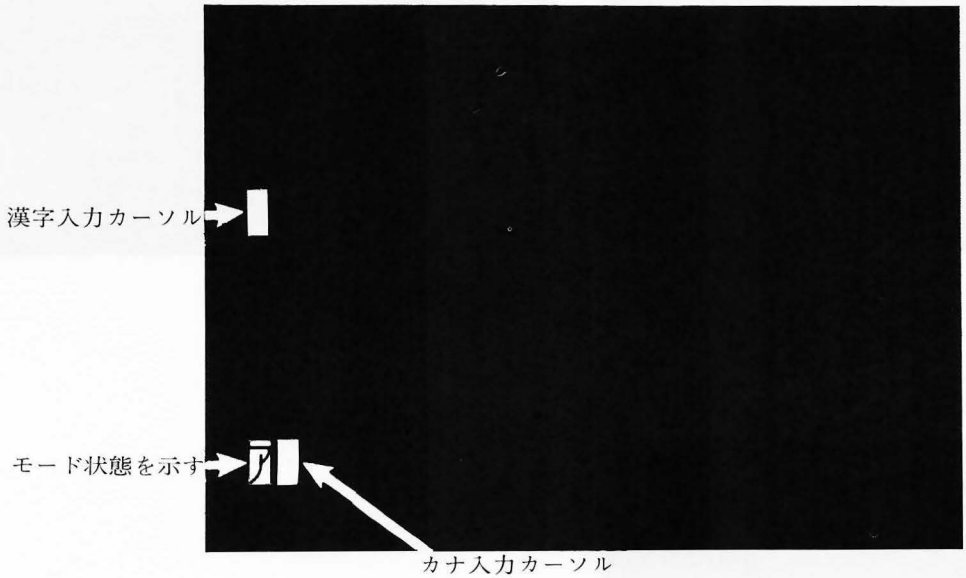
(注2) (+) 記号は SHIFT キーを押しながら (+) 記号の次のキーを押すことを意味します。

実際の操作方法などについて下記プログラム例を使用し説明します。

プログラム例

```
10  PRINT  CHR$(6):CONSOLE GH
20  GRAPH  I1, C, O1
30  POSITION  0, 100
40  KINPUT  A$
```

上記プログラムを実行するとCRT画面は次のようになり、カナ入力待ちの状態になります。

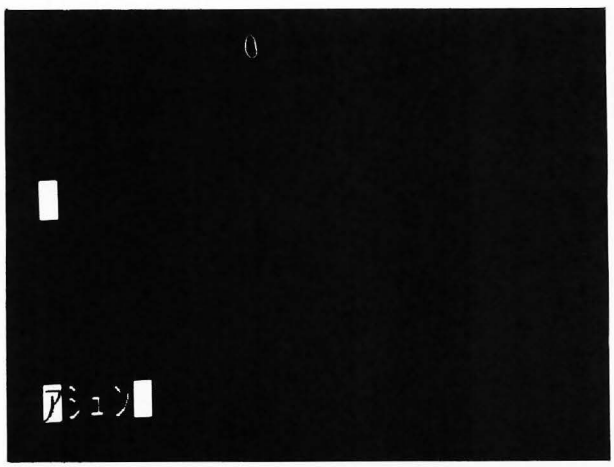


キー操作を次のように実行してください。(“春”を入力する場合)

①カナを入力します。

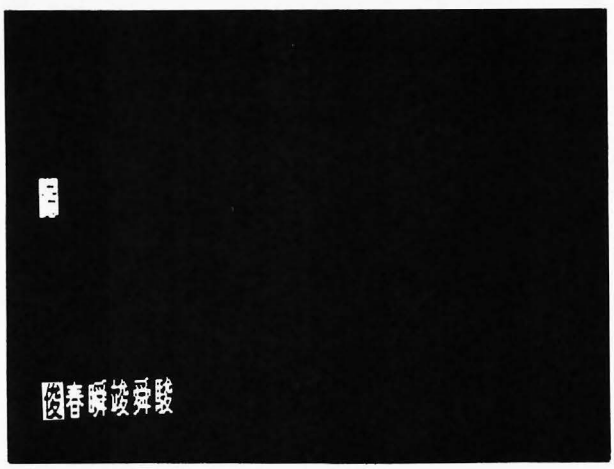
シ　ュ（SHIFT＋ユで入力します）ン

ここまでの操作ではCRT画面は次のようになります。



②シュンをカナ漢字変換します。

BREAK を押すと次のようになります。



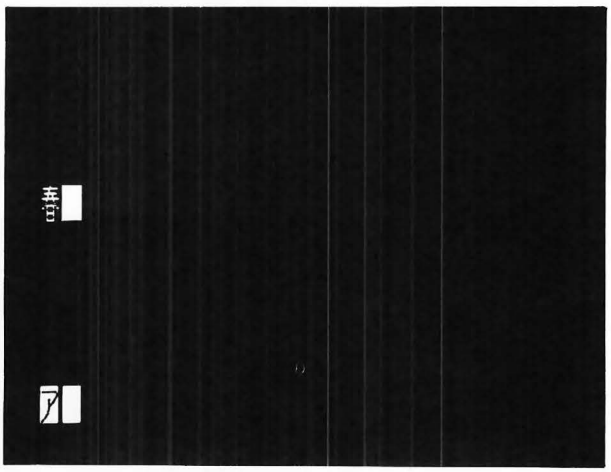
③ 漢字を選択します。

[→] を押すと次のようになります。



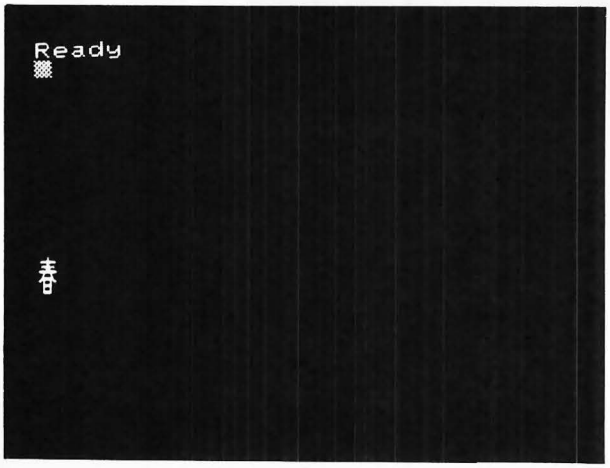
④ 漢字をストリング変数に入力します。

[CR] を押すと次のようになります。



⑤ 命令を終了します。

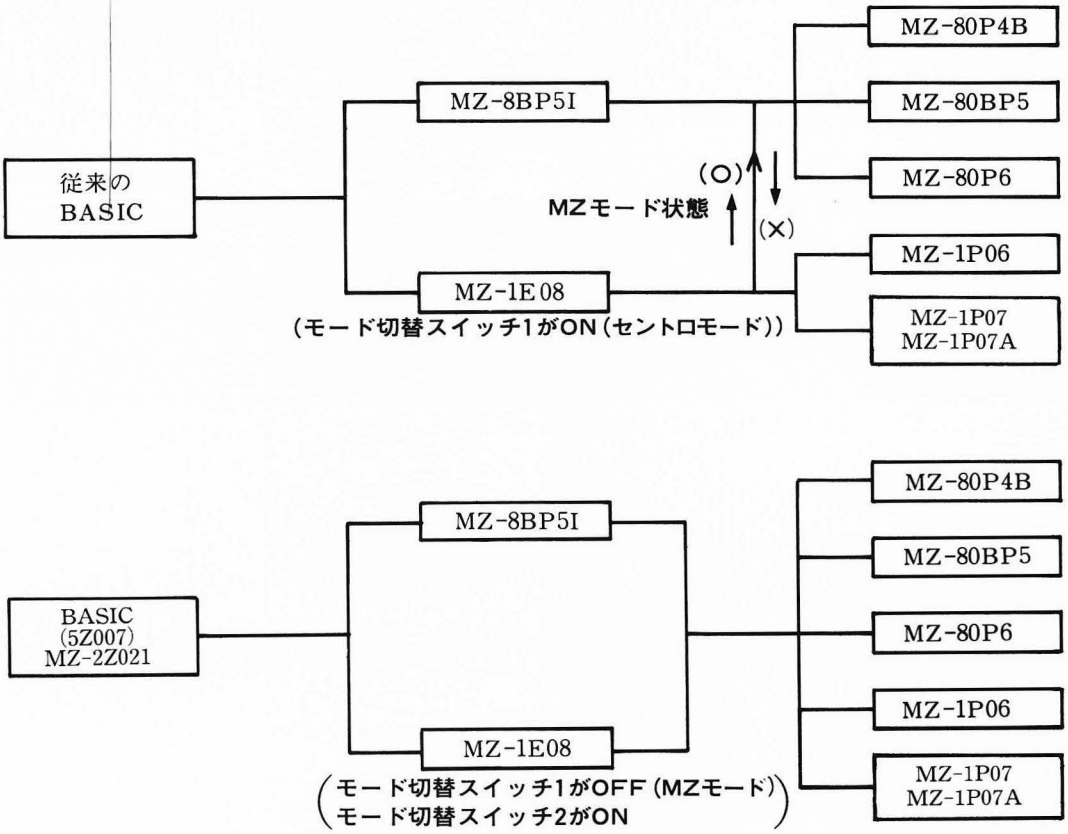
[SHIFT] ＋ [CR] を押すと次のようになります。



(注) 漢字は横16ドット × 縦16ドットで表示されます。

1.14 プリンタの接続

MZ-2200と各プリンタの接続方法は下記の通りです。



● PSEL （31ページ）を参照してください。



Chapter
2

拡張・追加コマンド、ステートメント

コマンドとステートメントの書式

■コマンドおよびステートメントを、アルファベット小文字、反転文字で表記することはできません。

■オペランド中でプログラマが指定すべき箇所は、イタリック体で示しています。

■各コマンドのオペランドで、省略あるいは任意の繰り返し記述が可能な要素については書式例を参照ください。(オペランド省略時におけるコロンの記述について特に注意する必要があります。)

■ここで説明するコマンド・ステートメントは、MZ-2000の標準BASIC MZ-1Z001より追加、拡張されたものです。

2.1 基本コマンド

2.1.1 ON ERROR GOTO

書　式　ON ERROR GOTO *lr*

lr……参照行番号（reference line number）：エラー処理ルーチンの先頭

機　能　エラーが発生した時、エラー処理を行うためにプログラム実行を移す行番号を宣言します。

解　説　ON ERROR GOTO 文によってエラー処理ルーチンを宣言することによって、エラー発生時に、BASIC コマンドレベルへ戻さずに、プログラム内でエラー処理を行うことが可能になります。

ON ERROR GOTO文が実行されていると、プログラム実行中にどのようなエラーが発生した場合もプログラム実行は *lr* で始まるエラー処理ルーチンへ移されて来ますが、このルーチンでIF ERN 文やIF ERL 文を用いることによってエラー番号（ERN）とエラー発生行番号（ERL）の判別ができ、それぞれ適当な処理を行うことができます。更に、エラー発生箇所にプログラム実行を戻すために、RESUME 文を使用することができます。

新たにON ERROR GOTO 文を実行すると、前のON ERROR GOTO 文は無効になります。

ON ERROR GOTOを宣言後、CLR命令を実行すると、この宣言は無効となります。

2.1.2 IF ERN

書　式　IF ERN *expression* THEN *lr*

IF ERN *expression* THEN *statement*

IF ERN *expression* GOTO *lr*

ERN *expression*……ERNの関係式

lr……参照行番号

機　能　発生したエラーの種類を判断して、エラー処理の分岐を行います。（ERN：error number）

解　説　ON ERROR GOTO 文の実行によってエラー処理ルーチンの参照行番号が定義されている場合には、エラー発生時に、システム変数ERN に、エラー番号が代入されそのエラー処理ルーチンへコントロールが移されます。エラー番号は、付録の表 P.116に示されています。

IF ERN文は、エラー処理ルーチン内で、発生したエラーを判別するものであり、IF 文によってERNの値を調べることができます。IF ERN文は、書式に示されているように他のIF文と同 様に3種類の形を使うことができます。

例　エラー処理ルーチンの先頭行番号を1000とし、そこで、エラー番号5（ストリングオーバーフロー ストリング長が255 文字を越えた）であれば更に行番号1200へジャンプさせる例を次に示しています。

```
10 ON ERROR GOTO 1000……エラー処理ルーチンを定義する
……
1000 IF ERN＝5 THEN 1200……ストリングオーバフロー・エラーであれば1200へジ
……                          ャンプする
```

2.1.3 IF ERL

書　式

IF ERL *expression* THEN *lr*

IF ERL *expression* THEN *statement*

IF ERL *expression* GOTO *lr*

ERL *expression*……ERLの関係式

lr……参照行番号

機　能

発生したエラー箇所を判断して、エラー処理の分岐を行います。(ERL：error line number)

解　説

エラー発生時に、システム変数ERLには、エラー発生行番号がセットされるので、ON ERROR GOTO文で宣言されたエラー処理ルーチンで、IF ERL文によるエラー発生箇所の判別がテキストの定義行番号によってできます。

IF ERL文は、IF ERN文と同じくIF～THEN、IF～GOTOのいずれの形でも使えます。

例

エラー発生行番号が250である時、1300行へ分岐するには、次の文を用います。

```
1010 IF ERL＝250 THEN 1300
```

エラー番号が43でかつ、その発生箇所が450行上でなければ、メインプログラム中の行番号520へ戻すには次の文を用います。

```
1020 IF (ERN＝43)＊(ERL＜＞450) THEN RESUME 520
```

2.1.4 RESUME

書　式

RESUME 〈NEXT〉

RESUME *lr*

lr……参照行番号または0

機　能

エラー処理後、プログラム実行をメインプログラムへ戻します。

解　説

エラー発生時にはエラー発生位置が記憶されているので、エラー処理を終了した後、その文あるいは、その次の文へプログラム実行を戻したり、あるいは、他の任意の行へ戻すことができます。

即ち次の4通りの復帰文が記述できます。

RESUME……エラーが発生した文へ復帰します。

RESUME NEXT……エラーが発生した文の次の文へプログラム実行を戻します。

RESUME *lr*…… *lr* で指定する行へプログラム実行を戻します。

RESUME 0 …… プログラムの先頭、即ち最小の行番号をもつ行へプログラム実行を戻します。

エラーが発生していないのにRESUME文に来た場合、ここでError 21 (RESUME-no ERROR)が発生します。

RESUME不能の場合、Error 20 (Can't RESUME) が発生します。

2.1.5 DELETE

書　式　DELETE 〈l_s - l_e〉

l_s……スタート行番号

l_e……エンド行番号

機　能　指定した行番号のプログラムを削除します。

解　説　スタート行番号l_sからエンド行番号l_eまでのプログラムを削除します。

l_sだけ指定した場合は、その行だけを削除し、ハイフォン以下l_eを指定した場合にはプログラムの先頭から指定行までを削除します。

l_sとハイフォンを指定するとl_s以下を削除します。

例

DELETE 10 - 230	行番号10～230までを削除します。
DELETE -150	プログラムの最初から行番号150までを削除します。
DELETE 20-	行番号20以降のプログラムを削除します。
DELETE 5	行番号5を削除します。

(注)クイックディスクのプログラム・データの削除は、DELETEユーティリティ(P.108参照)を使用します。

2.1.6 PSEL

書　式　PSEL　*m*

m……0　従来のMZ方式のプリンタの場合使用します。

m……1　MZ方式のプリンタに対してコントロールコードをそのまま出力します。

m……2　MZ-1P07、MZ-1P07Aのプリンタの場合使用します。

m……3　MZ-1P07、MZ-1P07Aのプリンタに対してコントロールコードをそのまま出力します。

m……4　MZ-1P06のプリンタに対してコントロールコードをそのまま出力します。

機　能　プリンタコマンドの各IOCSをMZ方式またはソフトによりセントロ方式に切換えます。

解　説　初期値は0で、MZ方式のIOCSに設定されています。

PSEL 1、PSEL 3を実行後、ハードコピーをとると行間をつめて印字しますが、それを終了すると、行間は1/6″のピッチに設定されます。

MZ-1P07、MZ-1P07Aの取扱説明書(付録)には機械語サブルーチンを使用してコントロールコードを送っている説明がありますが、PSEL1、PSEL3、PSEL4命令を使用する事により、機械語サブルーチンを作らなくてもコントロールコードを送ることができます。

当ソフトウェアのプリンタ制御はMZ－8BP5Iでも使用できるようになっていますので、インターフェイスMZ－1E08は従来のMZ方式のモードに切換えて使用してください。セントロ方式のプリンタを使用する時もMZ方式のモードで使用できます。

PSELモード	使用するプリンタ	コントロールコード	BASICコマンド	備　考
PSEL 0	MZ-80P4B MZ-80BP5 MZ-80P6	BASICでのコントロールコード(BASICインタープリタによりコードを変換してプリンタへ送信する)	PRINT／P IMAGE／P COPY／P PAGE／P KPRINT／P(注) LIST／P DIR／P SEARCH／P	
PSEL 1		コントロールコードをそのまま出力する		ハードコピー終了後紙送りは1/6″ピッチに設定される
PSEL 2	MZ-1P07 MZ-1P07A	BASICでのコントロールコード(BASICインタープリタによりコントロールコードを変換してプリンタへ送信する)		
PSEL 3		コントロールコードをそのまま出力する		ハードコピー終了後紙送りは1/6″ピッチに設定される
PSEL 4	MZ-1P06			

(注) MZ-1P07、MZ-1P07A、MZ-1P06、MZ-80P6の場合のみ有効です。
KPRINT／P命令を参照してください。

2.1.7 RENUM

書　式

RENUM〈*nls*〉〈,*sv*〉〈,*ols*－*ole*〉

nls……新しくつける行番号の最初の行番号：デフォルト値＝10

sv ……行番号の増分：デフォルト値＝10

ols……行番号のつけ換えをはじめる現在のプログラムの行番号

ole……行番号のつけ換えを終える現在のプログラムの行番号

機　能

プログラムの行番号を整理します。

解　説

オペランドをすべて省略すると現在のプログラムをすべて行番号10から増分10で整理します。

RENUMコマンドは、GOTO、GOSUB、THEN、ON～GOTO、ON～GOSUBおよびERL＝文などで参照している行番号も自動的に変更します。この時、もし参照している行番号が存在しない場合には、

〝UNDEF LINE xxx IN yyy (→zzz)〟

と表示されます。これは、その行は旧番号yyyから新番号zzzへ変更されたが、存在しない行番号xxxがそのまま残ったことを表します。

RENUMの結果、行の順序が入れ換わるような範囲を指定すると、エラーになります。

RENUMの結果、1行の長さが規定より長くなると、LIST命令を実行できなくなります。この場合には、もう一度小さい行番号でRENUMすると、LIST命令を実行できるようになります。

例

```
100 REM RENUM COMMAND SAMPLE PROGRAM
110 REM for SHARP MZ-2200 BASIC
120 REM
130 PRINT "RENUM コマンド ハ ギョウバンゴウヲ"
140 PRINT "ツケカエマス"
150 END

RENUM 200, 20, 130-
Ready
LIST
100 REM RENUM COMMAND SAMPLE PROGRAM
110 REM for SHARP MZ-2200 BASIC
120 REM
200 PRINT "RENUM コマンド ハ ギョウバンゴウヲ"
220 PRINT "ツケカエマス"
240 END
```

2.1.8 SEARCH

書　式

SEARCH $x$$, l_s - l_e

$x$$……ストリングデータ：ストリング変数

l_s ……スタート行番号

l_e ……エンド行番号

機　能

現在BASICテキストエリア内にあるプログラムテキストの全部、または一部に対して指定した文字列を含む行を探し出しCRT上へ表示します。

解　説

オペランドl_s、l_eの記述を省略すれば、プログラムテキストの全部に対して文字列の検索（サーチ）を行います。

プログラムテキストの一部分に対して文字列のサーチを行う場合、サーチスタート行番号l_sとエンド行番号l_eをマイナス記号〝-〟で結ぶかどちらか一方を記述します。

サーチ結果を表示中にスペースキーを押さえると、その間表示がいったん停止し離すとサーチ結果の表示を再開します。また SHIFT キーと BREAK キーを同時に押すとサーチを中断しBASICコマンドレベルへ戻ります。

オペランド$x$$はサーチするストリングデータであり、ダブルクォーテーション（〟）で囲んだストリングを記述するかストリング変数を記述します。

なお、サーチしたいストリングデータ内にダブルクォーテーションを含んでいる場合、たとえば、PRINT〝ABC〟に対しサーチを行う場合、次のように記述します。

SEARCH〝PRINT〟+CHR$ (34)+〝ABC〟+CHR$ (34)

例

SEARCH〝ABC〟……プログラムテキスト全部に対して文字列〝ABC〟を含む行をサーチし、ディスプレイに表示します。

SEARCH〝ABC〟, 100- ……行番号100以降の文に対して文字列〝ABC〟を含む行をサーチし、ディスプレイに表示します。

SEARCH〝ABC〟, -500 ……プログラムの先頭から行番号500までの文に対して文字列〝ABC〟を含む行をサーチし、ディスプレイに表示します。

SEARCH〝ABC〟, 100-500…… 行番号100～500の文に対して文字列〝ABC〟を含む行をサーチし、ディスプレイに表示します。

2.1.9 SEARCH／P

書　式　SEARCH／P　$x$$, $ls-le$

$x$$……ストリングデータ：ストリング変数

ls ……スタート行番号

le ……エンド行番号

機　能　指定した文字列を含む行番号を探し出し、プリンタ上に印字します。

解　説　SEARCH／P文は、SEARCH文の実行と同じ形式でサーチ結果をプリンタ上に印字します。
オペランドの記述も同様です。

例　SEARCH／P＂ABC＂，100－200……行番号 100～500 の文に対して文字列＂ABC＂を含む行を探し出しプリンタ上に印字します。

SEARCH／P＂KANJI＂……すべての行に対して文字列＂KANJI＂を含む行を探し出しプリンタ上に印字します。

2.1.10 TABSET

書　式　TABSET　x_1，x_2，……，x_n

x_i……数値データ（1≦n≦15）

機　能　TABキーのタビュレーション位置を設定します。（TAB：tabulation）

解　説　キー操作時に TAB キーが押されたとき、データ入力位置の表示ポインタであるカーソルがこのコマンドで指定されたタビュレーション位置に移動します。

タビュレーション位置の設定は15個まで可能です。

カーソルの移動は右方向にのみ行われますので、現在のカーソル位置より右方向でもっとも近いタビュレーション位置までカーソルは移動します。

CRTディスプレイの左端が、タビュレーション位置＝0で右端が39（40キャラクタモード）あるいは79（80キャラクタモード）です。

TABSETコマンドを実行すると、それまでに設定されていたタビュレーション位置がすべてクリアされます。

タビュレーションの設定位置を変更する場合はこのコマンドを用いてすべてのタビュレーション位置を設定しなおしてください。

例　TABSET　10，20，30…… TAB キーを押すごとにカーソルは画面の左から11，21，31番目の位置に移動します。

TABSET ……それまでに設定されていたタビュレーション位置はすべてクリアされます。

2.1.11 CHR$

書　式　CHR$ (x_1, x_2, ……,x_n)

x_i……数値および16進データ

機　能　各々の数値データx_iについて、ASCIIコードがx_iである１つのキャラクタを作成し、その作成したキャラクタを順にならべた文字列を与えます。

例　PRINT／P　CHR$ ($1B, $23)……プリンタへ$1B, $23のコードを送ります。

2.1.12 POKE

書　式　POKE　ad, d_1 〈,d_2, d_3, ……, d_n〉

ad……アドレス：数値データ、もしくは４桁の16進数

d_i……数値データ：０～255、もしくは２桁の16進数

機　能　メモリ上の任意のアドレスにオペランドで記述したデータを順次書き込みます。

解　説　adで指定するメモリ位置から順次直接１バイトのデータを書き込んでいく文であり、アクセスするアドレスは任意です。

各データは１バイトデータですので、０～255（16進の$00～$FF）のいずれかであり、数値データであれば０～255の範囲のいずれかの整数でなければなりません。

POKE文は、LIMIT文と関係なくメモリ空間上の任意の位置に対して実行可能ですので、BASICやMONITOR本体を壊す可能性があり、その使用には十分注意が必要です。

機械語プログラムまたはデータをPOKE文で構成する場合、あらかじめLIMIT文を実行して、ユーザエリアを確保しておく必要があります。

2.2 ファイルコマンド（クイックディスク）

2.2.1 DIR

書　式　DIR

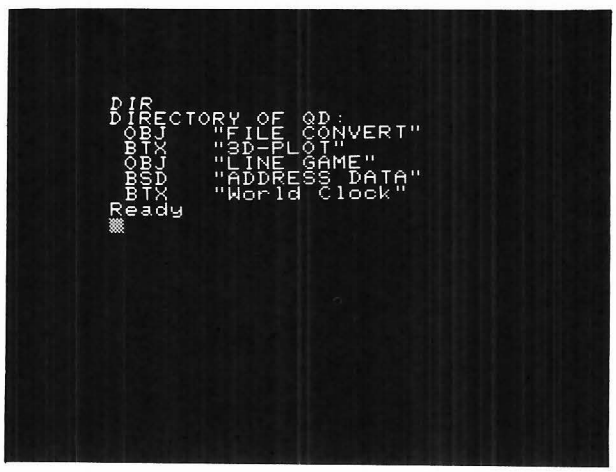
機　能　クイックディスクのディレクトリをCRTディスプレイ上に表示します。
（DIR：directory）

解　説　ディレクトリとは、クイックディスクに既に登録されているファイルの情報を表示します。

登録されている全てのファイル
　各ファイルについて、ファイルモード、ファイル名を示します。
　ファイルモードは3種類あり、次の各3文字のコードによって区別されます。
　　BTX……BASICテキストファイル
　　BSD……BASICシーケンシャルアクセス・データファイル
　　OBJ ……オブジェクトファイル
　ファイル名は、ファイル登録時に指定した名前で、ファイルの呼び出しはいつもその名前によって行います。

ディレクトリ表示は、ファイルを1画面ぶん（20個のファイル）表示すると、そこで一旦停止して、カーソルが現れます。更に続けてディレクトリ表示を行わせるには CR キーを押しますが、その状態から他のコマンドへ移ることもできます。
LOADコマンド、RUNコマンドの実行は、ディレクトリ表示上でカーソルを該当ファイルの行へ移動させ、ファンクションキー（BTX "AUTO RUN" によってLOADコマンドは F10 にRUNコマンドは F1 に定義されています）を押すことによって簡単に行うことができます。
なお、このDIR命令を実行すると、グラフィックディスプレイ出力モードページは、全ページともOFFとなります。（DIR／Pも同様）

例　図は、DIRコマンドによって、クイックディスクのディレクトリを表示させた1例です。



2.2.2 DIR/P

書　式　DIR/P

機　能　クイックディスクのディレクトリをプリンタ上にプリントアウトさせます。この場合、CRT ディスプレイ上への表示は行われません。

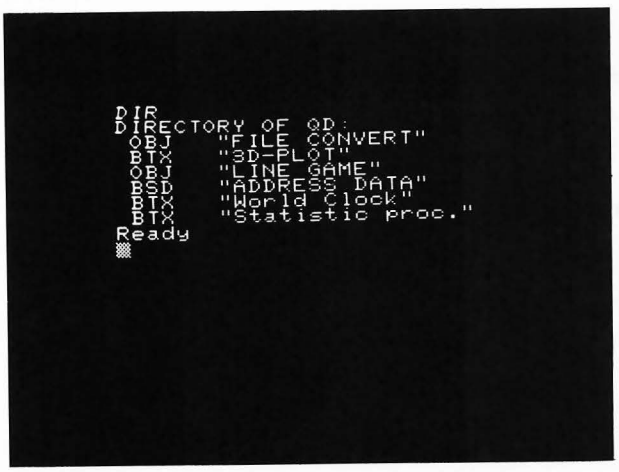
2.2.3 SAVE

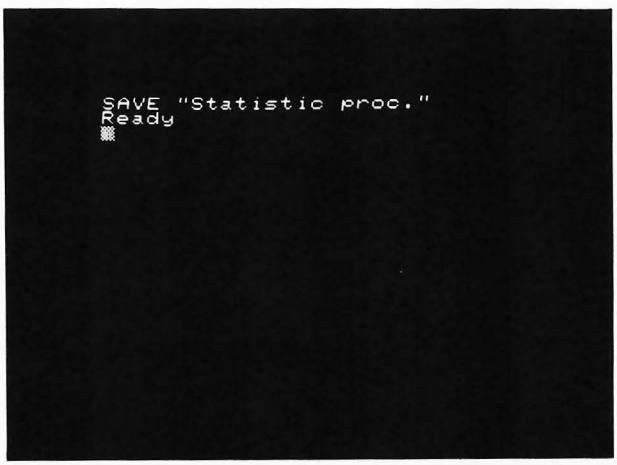
書　式　SAVE　"*file name*"

機　能　現在 BASIC テキストエリアにあるプログラムテキストを、ファイル名を指定してクイックディスク上へセーブします。

解　説　SAVE コマンドを実行することによってクイックディスク上に 1 つの BASIC テキストファイル（BTX）が登録されます。その際、オペランドに必ずファイル名となるストリングデータを記述しなくてはならず、クォーテーションマークで囲った16文字以内のストリングで指定します。
プログラムテキストをカセットテープ上にセーブするには、SAVE/T コマンドを用います。また、クイックディスク上にすでに存在しているファイルのファイル名と同じファイル名を指定することはできません。

例　図は、SAVE コマンドによって、スレーブディスク上に新しく 1 つの BASIC テキストファイルが登録されるもようを示しています。左が、SAVE コマンドを行う前のディレクトリ表示、右が SAVE コマンドの実行とディレクトリ表示であり、BTX ファイル " Statistic proc. " が登録されたのがわかります。†)





2.2.4 LOAD

書　式　LOAD　"*file name*"

機　能　BASICテキストエリアをクリアして、クイックディスク上に登録されているBASICテキストをロードします。

解　説　ロードすべきファイルは、ファイル名の記述によって指定されます。ファイル名は、登録されているファイル名を省略なしで指定しなければなりません。

ディレクトリ表示を利用してカーソル操作を行うと、ファンクションキー10番を用いてワンタッチでファイルローディングを行うことができます。

たとえば、ディレクトリ表示上の次のファイル

BTX　" Statistic proc. "

をロードするには、カーソルを次のようにこの行の先頭へ持って行き、

BTX　" Statistic proc."
↑
カーソル

ファンクションキー10番を押します。DEF KEY文によって新たに定義し直していなければ、ファンクションキー10番は、LOAD ↵が定義されているので、表示は、

LOAD　" Statistic proc."

となり、"↵"記号のファンクション、即ちキャリッジリターンも実行されます。従ってワンタッチでLOADコマンドを与えたことになるわけです。

カセットテープ上のプログラムテキストをロードするには、**LOAD/T**コマンドを用います。

2.2.5 APPEND

書　式　APPEND　"*file name*"

機　能　現在BASICテキストエリアにあるプログラムテキストと、指定したファイルのプログラムテキストを混ぜ合わせます。

解　説　指定したファイルの中に、BASICテキストエリアにあるプログラムと同じ行番号を持つものがあれば、テキストエリア上の行が対応するファイル上の行と置き換えられます。

例　APPEND " TEXT-B "

……テキストエリアに次のプログラムが格納されており

```
10  A=345
30  S=10*A
40  PRINT S
```

ディスク上のファイル＂TEXT-B＂の内容が次のとき

```
20  B=789
30  S=3*A-B
50  END
```

APPEND実行後のテキストエリア内のプログラムは次のようになります

```
10  A=345
20  B=789
30  S=3*A-B
40  PRINT S
50  END
```

2.2.6 RUN

書　式　RUN　＂*file name*＂

＂*file name*＂……BTXファイルまたはOBJファイル

機　能　BASICテキストエリアをクリアして、クイックディスク上に登録されているBASICテキストをロードし、続けてその先頭からプログラムを実行します。

OBJファイル(機械語オブジェクトファイル)へのRUNコマンドの実行は、BASICの使用をやめ、該当OBJファイルにコントロールを移すことになります。

解　説　BTXファイルに対してRUNコマンドを与えると、次のマルチステートメント、

LOAD　＂*file name*＂：RUN

を実行したのと同じく、ファイルのロードに続いて、そのプログラムの先頭から実行が行われます。RUNコマンドは、ファンクションキー1番に定義されているので、LOADコマンドと同様にカーソル操作と共に使用すると便利です。

OBJファイルをRUNさせると、本BASICから、該当する別のシステムプログラムへコントロールが移ります。マスターディスク上の次の3つのファイルは、ユーティリティオブジェクトファイルであり、このRUNコマンドによって起動されます。

```
OBJ    ＂QDCOPY＂
OBJ    ＂DELETE＂
OBJ    ＂TRANS＂
```

2.2.7 CHAIN

書　式　CHAIN "*file name*"

機　能　現在実行中のプログラムテキストから、ディスクファイル上の別のプログラムテキストへプログラム実行をチェインします。

解　説　CHAIN 文は、プログラム中で RUN "*file name*" コマンドを実行したのと似た働きがありますが、チェインする際、もとのプログラムで使用した変数、配列等の内容はそのまま新しいプログラムへ渡されます。従って、CHAIN 文は、GOTO *lr* 文と対照して GOTO "*file name*" のイメージとして捉えることができます。

例　CHAIN "Process 3"

……クイックディスクに登録されている、BTX ファイル〝Process 3〟へプログラム実行をチェインします。

2.2.8 WOPEN

書　式　WOPEN# *l*, "*file name*"

l ……ロジカルナンバ：*l* ＝1～127

機　能　1つのBASICシーケンシャルアクセスデータファイル（BSD）を作成するために書き込み用ファイルをオープンします。（WOPEN：write open）

解　説　WOPEN#文は、シーケンシャルデータ登録のための準備を行う文であり、ファイルアクセス用ロジカルナンバの定義と、ファイル名の指定を行います。

例

```
510  WOPEN#3, "SEQ  DATA1"
```

…… BSD"SEQ DATA1"を登録するため、ロジカルナンバ3番を書き込みオープンします。

※現在、存在するファイルに対しては、WOPEN #文を実行できません。

2.2.9 PRINT

書　式　PRINT# *l*, d_1, <, d_2, ……, d_n>

l ……ロジカルナンバ

d_i ……書き込みデータ並び

機　能　WOPEN#文によって書き込みオープンされているファイル上へ、オペランドで指定するデータを順次書き込んで行きます。

解　説　書き込みを行う対象となるBSDファイルは、ロジカルナンバ#1によって指定します。また、目的とするファイルを、それをオープンした時のロジカルナンバで正しく指示しなくてはなりません。なお、同時に複数のWOPENはできません。できるのは1つだけです。

例

```
510  WOPEN#3, "SEQ  DATA  1"
530  FOR  I=1  TO  30
540  PRINT  #3, A(I), B(I), A$(I)
550  NEXT  I
```

……BSDファイル"SEQ DATA1"をロジカルナンバ3番に書き込みオープンし、Iを1から30まで変えながら順次、A(I)、B(I)、A$(I)の各配列要素の内容を書き込んで行きます。全部で90個のデータを書き込むことになります。

2.2.10 POPEN

書　式　ROPEN#*l*, "*file name*"

l ……ロジカルナンバ：*l* = 1 ～127

機　能　BASIC シーケンシャルアクセスデータファイル（BSD）中のデータを読み出すためにファイルをオープンします。(ROPEN：read open)

解　説　ROPEN# 文は、シーケンシャルなデータ読み出しを行うための準備であり、読み出すべきファイルを指定し"*file name*"によって指定します）それを、ロジカルナンバ *l* 番に設定します。

2.2.11 INPUT

書　式　INPUT#*l*, v_1, <, v_2, ……, v_n>

l ……ロジカルナンバ

v_i ……入力並び：変数または配列要素

機　能　ROPEN# 文によって、読み出しオープンされている BSD ファイルの先頭データから順次データを読み出し、オペランドの入力並びへ代入します。

解　説　データの読み出しを行うファイルは、ロジカルナンバ *l* 番にROPEN# 文を実行しているファイルになります。READ～DATA 文の場合と同様に、データと、入力並びのデータ型が一致しないとエラーが発生します。

例

```
700  ROPEN  #50, "SEQ  DATA  1"
710  FOR  I=1  TO  30
720  INPUT  #50, AA(I), BB(I), AA$ (I)
730  PRINT／P "#No.";AA(I), " Vol." ; BB(I) , " Name： " ; AA$ (I)
740  NEXT  I : CLOSE  #50
```

……BSDファイル"SEQ DATA 1"に登録されているデータを行番号720のINPUT# 文で3個ずつ読み出し、その値をプリンタに整理して打ち出して行きます。

2.2.12 KILL

書　式　KILL　＃*l*

l ……ロジカルナンバ

機　能　ロジカルナンバ*l*番にオープンしたBSDファイルの登録をキルします。即ち、WOPEN＃文によってファイル作成を準備し、あるいはそれに続いてPRINT＃を実行しているファイルの正式登録を途中でキャンセルします。

解　説　ロジカルナンバを指定しない場合は、現在作成中のファイルの正式登録をキャンセルするとともに、すでにオープンされているすべてのファイルをクローズし、そのロジカルナンバをすべて未定義番号に戻します。

なおKILL命令は直接実行命令として使用できますので、クイックディスクを交換する直前にこの命令を実行させ、すべてのファイルをクローズすることによってクイックディスク内容の破壊を保護することができます。

2.2.13 CLOSE

書　式　CLOSE　＃*l*

l ……ロジカルナンバ

機　能　ロジカルナンバ*l*番にオープンされているファイルをクローズして、この番号を未定義番号に戻します。

解　説　CLOSE文は、次のように、3種類のロジカルオープンに対して実行されます。

■ WOPEN＃に対するCLOSE

WOPEN＃、PRINT＃文によるシーケンシャルアクセスデータの書き込みを終了して、このファイルを正式にクイックディスク上へ登録します。使用したロジカルナンバを未定義に戻します。

■ ROPEN＃に対するCLOSE

ROPEN＃によってデータ読み出しオープンしたBSDファイルをクローズし、使用したロジカルナンバを未定義にします。

■ ロジカルナンバを指定しない場合は、現在オープンされているすべてのファイルをクローズして、そのロジカルナンバをすべて未定義番号に戻します。

例

```
510 WOPEN＃3, "SEQ  DATA  1"
530 FOR  I=1  TO  30
540 PRINT  ＃3, A(I), B(I), A$(I)
550 NEXT  I
560 CLOSE  ＃3
```

……行番号560のCLOSE＃文によって、行番号510～550のループでデータを書き込んで行ったBSDファイル"SEQ DATA 1"を正式にファイル登録し、クローズします。

2.2.14 INIT

書　式　INIT " Y "

機　能　クイックディスクの初期化（イニシャライズ）を行います。新しいクイックディスクを使用するときは必ずこの命令を実行してください。INIT だけを指定すると、OK「Y/N？」とたずねてきますので、OK ならば Y キーを押し、NO ならば N キーを押してください。この命令を実行するとディスク上に記憶されていた内容が全て消去されます。

2.2.15 IF EOF (#) THEN

書　式　IF EOF(#*l*) THEN *lr*（または *statement*）

l ……ロジカルナンバ

lr……参照行番号

機　能　BSD ファイルに対して INPUT #文を実行したときに、アウトオブファイルが発生した場合エラーが発生せず、入力並びには、0 または " "（null string）が入ります。IF EOF (#) THEN 文はこの、アウトオブファイルが発生した場合の処理を定めるもので、各 INPUT #文の後に置きます。もし INPUT #文でアウトオブファイルが生じていれば、THEN 以下が実行されることになります。

解　説

```
800  ROPEN  #3 , " DATA "
810  INPUT  #3 , DT$ : PRINT  DT$ ,
820  IF  EOF(#3)  THEN CLOSE : END
830  GOTO  810
```

……BSD ファイル " DATA " に登録されているデータを最初から最後まで全部読み出して CRT ディスプレイ上へ表示させます。

2.3 カラーコマンド

注)カラーコントロールコマンド、ステートメントをMZ-2000で使用される場合は、オプションのグラフィックRAM MZ-1R01、およびMZ-1R02を本体内に設置する必要があります。(詳しくはMZ-2000オーナーズマニュアル参照)

2.3.1 CCOLOR

書　式　CCOLOR@ *m*, *n*

@……プライオリティ

m……キャラクタのカラーコード：数値データ(0～7)

n……バックグランドのカラーコード：数値データ(0～7)

機　能　キャラクタおよびバックグランドのカラーコードと、プライオリティを指定します。

解　説　キャラクタのカラーコードを緑(4)に指定しますと、カラーCRTディスプレイ1画面すべてのキャラクタが緑色に表示されます。

バックグランドのカラーコードを青(1)に指定しますと、カラーCRTディスプレイのバックグランド（背景色）が青色となります。

@の記述のある場合、プライオリティをキャラクタに指定します。@の記述を省略した場合はすでに指定されたプライオリティをそのまま維持します。

プライオリティをキャラクタよりグラフィックに変更する場合には、COLOR@を実行させます。

オペランド*m*または*n*を省略した場合はすでに設定されている色をそれぞれ維持します。

例　CCOLOR@6,1……プライオリティをキャラクタ（文字）優先にし、キャラクタを黄色にバックグランドを青色に指定

CCOLOR@ 5……プライオリティをキャラクタに指定し、キャラクタを水色に指定

CCOLOR 4……キャラクタを緑色に指定

CCOLOR@,3……プライオリティをキャラクタに指定し、バックグランドを紫色に指定*

CCOLOR,2……バックグランドを赤色に指定*

* キャラクタのカラーコード*m*を省略し、バックグランドのカラーコード*n*を記述する場合この例のように、かならずコンマ(,)を記述しなければなりません。

プログラム例

```
10 COLOR,00:CONSOLE C40,M:GRAPH C7
20 PRINTCHR$(6);TAB(10);"### ASCII ｺｰﾄﾞ ﾋｮｳ ###"
30 FOR N=30 TO255
40 PRINT CHR$(N);SPACE$(1);
50 NEXT N
60 FOR B=1 TO 7
70 C=B+4:IFC>7 THEN C=C-7
80 CCOLOR C,B:FOR I=0 TO 2000:NEXT
90 NEXT B
100 CCOLOR 6,1:END
```

2.3.2 COLOR

書　式

COLOR@ *m*, O*n*, W*a*

@……プライオリティ

m……グラフィックのカラーコード：数値データ（0～7）

n……グラフィック出力ページモード：数値データ（0～7）

a……色の重ね合わせ：数値データ
- 0……強制的に*m*で指定した色へ変更
- 1……すでに描かれた色の上に*m*で指定した色を重ね合わせる。

機　能

カラーグラフィック表示についてグラフィックのカラーコード、出力ページモード、プライオリティおよび色の重ね合わせモードを指定します。

解　説

カラーグラフィックコントロール文（GRAPH C、GRAPH F、SET、RESET、LINE、BLINE、PATTERN、BOX、PAINT、CIRCLE）でのカラーコードの指定を省略した場合、COLOR *m*で指定したカラーコードに従って各々の命令を実行します。

*n*はグラフィックの出力ページモードを指定するオペランドで、その関係を右表に示しています。全ページを出力させる場合はCOLOR, O7 を全ページともOFF とする場合はCOLOR, O0 を実行させます。

n	グラフィック出力ページ		
	BLUE	RED	GREEN
0	×	×	×
1	○	×	×
2	×	○	×
3	○	○	×
4	×	×	○
5	○	×	○
6	×	○	○
7	○	○	○

グラフィック出力ページモード

@を記述しますとプライオリティはグラフィックに指定され@の記述を省略しますと、すでに指定されたプライオリティが維持されます。

プライオリティをグラフィックよりキャラクタに変更する場合はCCOLOR@を実行させます。

オペランドWaは、色の重ね合わせについてのモードを指定するものです。

m、*n*、*a*を数値変数として記述される場合には、それぞれ定められた数値の範囲内となるよう配慮してください。

許容数値の範囲外を指定しますとError 3が発生します。

m、*n*、*a*に整数以外の数値を使用しますと、小数点以下は無視されて実行します。

例

COLOR@ ……プライオリティをグラフィックに指定

COLOR 5……グラフィックのカラーコードを水色に指定

COLOR, O1, W1……BLUEのグラフィックエリアのデータのみを表示し、色の重ね合わせのモードに指定

COLOR, , W0……強制的に色を変更するモードに指定

COLOR@ 1, O7, W1……プライオリティをグラフィックに、グラフィックのカラーコードを青色、出力ページモードを全ページ出力とし、色の重ね合わせを行うモードに指定

COLOR 2, , W0……グラフィックのカラーコードを赤色とし、強制的に色を変更するモードに指定

プログラム例

```
10 CONSOLE C40,6H,P:CCOLOR@ 7,0:COLOR,07:C=1
20 PRINTCHR$(6):CURSOR 4,12
30 PRINT"SHARP PERSONAL COMPUTER MZ-2000"
40 FOR X=15TO 627 STEP 32
50 BOX[C]X,50,X+16,150,F
60 C=C+1:IF C=7 THEN C=1
70 NEXT X
80 COLOR@:GOSUB110
90 CCOLOR@:GOSUB110
100 GOTO 80
110 FOR I=0 TO 1500:NEXT:RETURN
```

2.3.3 GRAPH

書　式

GRAPH C*m*, F*n*

m, *n*……カラーコード：数値データ（0～7）

機　能

指定された色のグラフィックエリアのクリアまたはフィリング（埋めつくし）を行います。

解　説

標準BASIC（MZ-1Z001またはMZ-2Z001）でのGRAPH文にC*m*, F*n*の機能が追加となったものです。したがって入力モードの設定(Ia)および出力モードの設定(Ob)がこのGRAPH文で行えます。（BASIC／MONITOR MANUALのGRAPH文参照）

GRAPH I_a, O_b命令の機能は、COLORコマンドに含まれています。どちらもモード設定文であり、同一プログラム内で、これら2つの命令を混用しても何ら支障ありません。ただしこの場合、GRAPH文、COLOR文にかかわらず、常に最後に設定された入力もしくは出力モードが有効となります。たとえば次のプログラムを実行します。

```
10  GRAPH I1
20  COLOR, O7
30  SET 320, 100
40          :
```

SET文以降の入力ページモードは1（ページ1あるいはBLUE）となり、出力ページモードは全ページONのモードに設定されます。

m、*n*の記述を省略しますとCOLOR文で設定された入力モードに対してクリアまたはフィリングが行われることになります。

例

GRAPH C3, F4……青と赤のグラフィックエリアをクリアし、緑のグラフィックエリアをフィリング

GRAPH C……COLOR文で設定された入力モードに対しクリア

GRAPH F……COLOR文で設定された入力モードに対しフィリング

GRAPH CA……数値変数Aで示される色のグラフィックエリアをクリア

2.3.4 CONSOLE

書　式　CONSOLE P……白/黒CRTディスプレイ画面へのグラフィックデータの表示を禁止
CONSOLE M……白/黒CRTディスプレイ画面へグラフィックデータを表示させる

機　能　白/黒CRTディスプレイ画面へグラフィックデータを表示させるか否かを指定します。

解　説　標準BASIC(MZ-1Z001またはMZ-2Z001)でのCONSOLE文に上記機能が拡張されたものです。したがってCONSOLE文の他のオペランド (Cn, Sys, ye, R, N, GH, GN) とともに順不同で記述できます。(BASIC／MONITOR MANUAL の CONSOLE 文参照)
なおCONSOLE GH（またはGN）を実行しますと、同時にCONSOLE Mモードに設定されます。

例　CONSOLE C40, GH, P……40キャラクタモード、ハイリゾリューションモードに設定し、白/黒CRT画面上へのグラフィック画面の表示を禁止します。
CONSOLE C80, M……80キャラクタモードに設定し、白/黒CRT画面上へグラフィック画面を表示させます。

2.3.5 SET

書　式　SET［m, Wa］ x, y
m……カラーコード：数値データ（0～7）
a……色の重ね合わせ：数値データ（0または1）
x……X座標：数値データ
y……Y座標：数値データ

機　能　指定した色で任意の1ドットをセットします。

解　説　セットするドット位置は，X座標：x，Y座標：yで指定します。グラフィック画面の座標については1.4　ディスプレイ画面の座標（P.12）を参照ください。
座標位置の指定範囲は次の通りです。
0≦X座標≦16383
0≦Y座標≦16383
すなわち、各種の演算過程で、各座標が上記数値の範囲内となるように考慮する必要があります。
［m，Wa］を省略しますと、COLOR文にて設定されたカラーコントロールに従って実行します。

例　SET[1 , W0]300, 100……青色をセット
SET[A]300,100……数値変数Aの内容に従った色をセット
SET[, W1]300,100……COLOR文で設定された色をすでに描かれている色に重ね合わせてセット
SET X1, Y1 ……COLOR文で設定されたカラーコントロールに従ってセット

プログラム例
```
10 CONSOLE GH,M:PRINTCHR$(6):COLOR,07,W0:CCOLOR 7,0
20 FOR T=0 TO 16 STEP 0.015
30 C=INT(T/2)+1:IF C=8 THEN C=1
40 SET[C] 125*COS(COS(T*π)+T*π/4)+320,50*COS(COS(-T*π)-T*π/8)+100
50 NEXT
60 END
```

2.3.6 RESET

書　式　RESET［m, Wa］x, y

m……カラーコード：数値データ（0～7）

a……色の重ね合わせ：数値データ（0または1）

x……X座標：数値データ

y……Y座標：数値データ

機　能　指定した色で、任意の1ドットをリセットします。

解　説　リセットするドット位置とX－Y座標の関係および座標位置の指定範囲はSET文と同じです。

［m, Wa］を省略しますと、COLOR文にて設定されたカラーコントロールに従って実行します。

aを1に指定しますとmで指定された色をリセットします。

aを0に指定しますとmで指定された色の反転色をセットします。たとえば

RESET［5, W0］300, 100

を実行しますと水色(5)の反転色である赤色(2)が1ドットセットされます。

例　RESET A, B ……COLOR文で設定されたカラーコントロールに従ってリセット

RESET［3, W1］300, 100 ……紫、すなわち青と赤のグラフィックエリア上のドットをリセット

RESET［C］300, 100……数値変数Cの内容に従ってリセット、WaはCOLOR文に従う。

RESET［, W0］300, 100……COLOR文で設定されたカラーコードの反転色をセット

プログラム例

```
10 CONSOLE GH,P:COLOR7,07,W1:CCOLOR3 1,0:K=7:PRINTCHR$(6):GRAPH F
20 FOR R=4  TO100 STEP 8
30 FOR T=0 TO 2 STEP 0.005
40 X=4*R*COS(T*π)+400:Y=R*SIN(T*π)+120
50 RESET[K]X,Y-1:RESET[K]X,Y:RESET[K]X,Y+1:RESET[K]X,Y-2:RESET[K]X,Y+2
60 NEXT T
70 K=K-1:IF K=0 THEN K=7
80 NEXT R
90 END
```

2.3.7 LINE

書　式　LINE[m , W a] x_1 , y_1 , x_2 , y_2 , …… x_i , y_i

m ……カラーコード：数値データ（０～７）

a ……色の重ね合わせ：数値データ（０または１）

x_i ……Ｘ座標：数値データ

y_i ……Ｙ座標：数値データ

機　能　指定された色で任意のライン（直線）を描きます。

解　説　Ｘ－Ｙ座標を示すx_1 , y_1が１組みとなり、最初の組みの示す座標（x_i , y_i）から次々に線分を描いていきます。表示エリア外に座標が指定された場合は表示エリアの線分のみが描かれます。

Ｘ－Ｙ座標の指定範囲はＳＥＴ文の場合と同じです。

例　LINE 0, 0, 639, 199 ……COLOR文で設定されたカラーコントロールに従ってラインを描く。

LINE[6, W1]0, 100, 639, 100 ……以前に描かれている色の上に重ね合わせて黄色のラインを描く。

LINE[A]X1, Y1, X2, Y2 ……数値変数Ａの内容の色でラインを描く。
（WaはCOLOR文で設定されたモードに従う）

LINE[, W0]600, 0, 600, 199 ……COLOR文で設定された色でラインを描く。

プログラム例

```
10 CONSOLE GH,M:COLOR,07:PRINT CHR$(6):C=1:CCOLOR@ 7,0
20 GOSUB 60
30 X=X+210:C=C+1:IFX=630 THEN Y=Y+100:X=0
40 IF Y=200 THEN END
50 GOTO20
60 FOR I=40 TO 0 STEP -5
70 LINE[C] X,Y+I,X+210,Y+99-I
80 LINE[C] X,Y+99-I,X+210,Y+I
90 NEXT
100 FOR I=0 TO 84 STEP 10
110 LINE[C] X+I,Y,X+210-I,Y+99
120 LINE[C] X+210-I,Y,X+I,Y+99
130 NEXT:RETURN
```

2.3.8 BLINE

書　式　BLINE[m, Wa]x_1, y_1, x_2, y_2……x_i, y_i

m……カラーコード：数値データ（0～7）

a……色の重ね合わせ：数値データ（0または1）

x_i……X座標：数値データ

y_i……Y座標：数値データ

機　能　指定した色で任意のラインを消去します。

解　説　描く線分がリセットされた線分であることの他は、LINE文と同じです。

aを1に指定しますとmで指定された色の線分のみを消去します。

aを0に指定しますとmで指定された色の反転色を描きます。たとえば

BLINE[6, W0]0, 0, 639, 199

を実行しますと青色のラインを描くことになります。

例　LINE文と同じオペランドの記述法となります。

プログラム例

```
10 CONSOLE GH,P:COLOR,07,W1:GRAPH F7:CCOLOR@ 1,0:C=7:PRINTCHR$(6)
20 FOR X1=50 TO639 STEP90
30 FOR X2=X1-45 TO X1+45 STEP2
40 BLINE[C] X1,0,X2,150:NEXT X2
50 C=C-1:NEXT X1
55 C=7
60 FOR X=50 TO639STEP90
70 FOR XX=X-5 TO X+5
90 BLINE[C] XX,151,XX,199
90 NEXT XX
100 C=C-1:NEXT X
110 BOX[4,W0] 0,189,639,199,F
120 END
```

2.3.9 PATTERN

書　式

PATTERN［m_1, Wa_1］x_1, y_1＄,［m_2, Wa_2］x_2, y_2＄……,［m_n, Wa_n］x_n, y_n＄

m_i ……カラーコード：数値データ（0～7）

a_i ……色の重ね合わせ：数値データ（0または1）

x_i …… ドットの積み重ね段数：数値データ（±1～±24の範囲）

y_i＄……8ビット単位のドットパターンを表わすデータ並び：ストリングデータ

機　能

指定された色で任意のグラフィックパターンを描きます。

解　説

複数のストリングデータに対して、それぞれ別個にカラーコントロールを設定することができます。カラーコントロールを省略した場合は、COLOR文で指定されたカラーコントロールに従ってパターンを描きます。

例

PATTERN －8, A＄……COLOR文で設定のカラーコントロールに従って、グラフィックパターンを描く。

PATTERN[2, W0]＋10, B＄ ……色の重ね合わせなしに強制的に赤色でグラフィックパターンを描く。

PATTERN[, W1]＋16, C＄……COLOR文で設定の色に、すでに書かれている色を重ね合わせてグラフィックパターンを描く。

PATTERN[C]－5, D＄ ……変数Cの内容に従った色で、グラフィックパターンを描く。

プログラム例

```
10 CONSOLE GH,M:CCOLOR,0:PRINTCHR$(6):C=1:COLOR,07,W0
20 FOR I=0 TO 63:READ A:A$=A$+CHR$(A):NEXT
30 FOR Y=0 TO 180 STEP 20
40 FOR X=0 TO 600 STEP 40
50 POSITION X,Y
60 PATTERN[C] -16,A$
70 C=C+1:IF C=8 THEN C=1
80 NEXT X,Y
90 GOTO 90
100 DATA0,0,7,16,127,240,198,28,56,0,0,0,1,30,32,7,7,63,228,198,255,1,112,96,127,63,56,127,252,3,15,240
110 DATA225,243,7,30,255,128,224,194,254,252,3,255,30,240,191,3,0,128,196,14,255,14,200,112,56,16,0,192,0,0,254,252
```

2.3.10 PAINT

書　式

PAINT[m] x, y, n_1, n_2, ……n_j

m……カラーコード：数値データ（0～7）

x……X座標：数値データ　許容範囲 { 0～639(640×200ドット/画面) / 0～319(320×200ドット/画面) }

y……Y座標：数値データ　許容範囲　0～199

n_j……境界線のカラーコード（0～7の8色まで指定可）

機　能

指定された境界色で囲まれた範囲内を指定された色でぬりつぶします。

解　説

[m]の記述を省略しますとCOLOR文で設定されたカラーコードに従います。n_jは境界色を指定するオペランドで、最大8色まで指定できます。

PAINT文には色の重ね合わせの機能はありません。したがってCOLOR文でW1モードに設定しても、このPAINT文に対しては無効となります。

x, yによって指定された座標を囲む境界色の範囲内のエリアを、カラーコードmによって指定された色でぬりつぶします。その際そのエリアが指定された境界線によって閉ループ状態となっていなければ、そのエリア外に対してもぬりつぶしを行います。

境界線のカラーコード、n_jの記述を省略しますと、ぬりつぶしのカラーコードmが境界色となります。また、n_jで指定した境界色以外に、ぬりつぶしのカラーコードも境界色となります。

なお、座標（x, y）を境界線上に指定した場合、またはすでに境界色で描かれている場所に指定した場合は、ぬりつぶしを行いません。

例

PAINT 320, 100 ……COLOR文で指定されたカラーコードが赤の場合、座標(320, 100)を囲む赤色の境界線の範囲内を赤色でぬりつぶす。

PAINT[6]320, 100 ……座標(320, 100)を囲む黄色の境界線の範囲内を黄色でぬりつぶす。

PAINT XN, YN, 3……座標(XN, YN)を囲む紫色の境界線の範囲内をCOLOR文で指定されたカラーコードに従ってぬりつぶす。

PAINT[4]320, 100, 1, 6, 7 ……座標(320, 100)の周りの青、黄、緑および白色の境界線によって囲まれた範囲内を緑色でぬりつぶす。

プログラム例

```
10 CONSOLE GH,M:COLOR,07,W0:CCOLOR 6,0:PRINTCHR$(6)
20 LINE[7]10,70,30,50,100,50,100,70,130,50,200,50,200,70,230,50,300,50
30 LINE[7]300,50,300,150,230,150,230,80,200,90,200,150,130,150,130,80,100,90,100,150,30,150,30,80,10,90,10,70
40 LINE[3]320,50,540,50,540,80,400,120,540,120,540,150,320,150,320,120,460,80,320,80,320,50
50 BOX[4] 5,45,565,155
60 PAINT[2] 295,149,7
70 PAINT[1] 539,149,3
80 PAINT[7] 564,154,4,3,7
90 FOR I=1 TO 7:FOR N=0 TO 1500:NEXT N
100 COLOR,0I:NEXT I
110 END
```

2.3.11 BOX

書　式

BOX[m, Wa]x_1, y_1, x_2, y_2, Fn

m……カラーコード：数値データ（0～7）

a……色の重ね合わせ：数値データ（0または1）

x_1, x_2……X座標：数値データ　許容範囲 { 0～639(640×200ドット/画面) / 0～319(320×200ドット/画面) }

y_1, y_2……Y座標：数値データ　許容範囲　0～199

F ……ぬりつぶし

n ……ぬりつぶしのカラーコード（0～7）

機　能

指定された2点（x_1, y_1)と(x_2, y_2）を頂点とする四角形を描きます。

ぬりつぶしが指定されると，その枠内のエリアを指定された色でぬりつぶします。

解　説

[m, Wa] の機能はSET文と同じです。これを省略しますとCOLOR文のカラーコントロールに従います。

Fの記述を省略しますと枠だけを描き、ぬりつぶしは行いません。またFを記述しnを省略しますと、mで指定された色でぬりつぶしを行います。つまり、枠の色と枠内のぬりつぶしの色を同じにする場合は、nの記述を省略すればよいわけです。

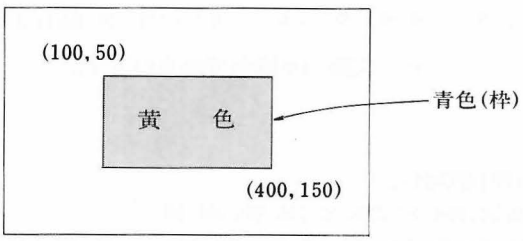
例

BOX 10, 10, 630, 190 ……COLOR文で設定のカラーコントロールに従い四角形の枠のみを描く。

BOX [1]10, 10, 630, 190 ……青色で四角形の枠のみを描く（WaはCOLOR 文での設定に従う）

BOX[, W1]20, 20, 620, 180, F……COLOR文で設定のカラーコードに従い、四角形の枠を描き、枠内を枠と同じ色でぬりつぶす。(色は重ね合わされる)

BOX[1, W0]100, 50, 400, 150, F6 …青色で四角形の枠を描き、その枠内を黄色でぬりつぶす。(下図参照)



プログラム例

```
10 CONSOLE GH,M:COLOR@,07,W0:CCOLOR 6,0:PRINTCHR$(6)
20 X2=639:Y2=199:C=1:K=0
30 FOR X1=0 TO39  STEP 3
40 BOX[7] X1,Y1,X2,Y2,FC
50 Y1=Y1+2:X2=X2-30:Y2=Y2-12
60 IF K=0 THEN CN=C:C=0:K=1:GOTO80
70 C=CN+1:K=0
80 NEXT
90 CURSOR 4,4:PRINT"BOX-SAMPLE"
100 END
```

2.3.12 CIRCLE

書　式

CIRCLE[m, Wa]x, y, r, h, ks, ke, O

m……カラーコード：数値データ（0～7）

a……色の重ね合わせ：数値データ（0または1）

x……X座標
y……Y座標
} 楕円(円)の中心座標：数値データ

r……半径

h……比率（0または正の数値）

ks……開始角度
ke……終了角度
} (単位ラジアン…radian)

O……扇を描く（省略すれば弧を描く）

機　能

画面上の任意の座標を中心として、指定された色で楕円(円)または、円弧(扇)を描きます。

解　説

カラーコントロール［m, Wa］についてはSET文と同じです。

座標(x, y)を中心にして、半径rの円を描きます。座標位置および半径rの指定範囲は次の通りです。

$-16383 \leqq$ X座標 $\leqq 16383$

$-16383 \leqq$ Y座標 $\leqq 16383$

$0 \leqq$ 半径 $r \leqq 16383$

比率hとは、(Y軸方向の半径)/(X軸方向の半径)のことです。

半径rとは、比率hが1以下の場合はX軸方向の半径のことで、比率hによりY軸方向の半径が決まります。

比率hが1以上の場合半径rは、Y軸方向の半径となり、比率hによりX軸方向の半径が決まります。

円を描くための比率hの値は、ハイリゾリューションモード（640×200ドット/画面）の場合、0.41であり、ノーマルリゾリューションモード（320×200ドット/画面）の場合、0.82です。比率hの記述を省略しますと、hは各モードにより上記数値が設定されます。比率hは0または正の数値で指定します。ハイリゾリューションモードの場合、$h>0.41$であれば縦長の楕円を描き、$h<0.41$であれば横長の楕円を描きます。

（このhの値はカラーCRTディスプレイMZ-1D01を対象にしています。白/黒CRTディスプレイ上で円とするには、ハイリゾリューションモードの場合は$h=0.5$に、ノーマルリゾリューションモードの場合は$h=1.0$に指定してください。）

開始角度ks、終了角度keを指定しますと円弧または扇を描きます。角度の単位はラジアン(radian)です。ks, keの記述を省略しますと、$ks=0$、$ke=2\pi$が設定され、楕円(円)を描きます。

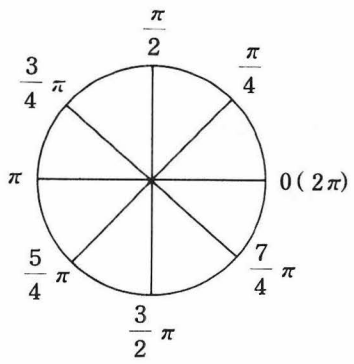
ks, keを指定し、Oを記述しますと扇を描き、Oの記述を省略しますと円弧を描きます。

ks, keの記述の際、度(degree)単位の数値データdを、ラジアン(radian)単位の数値データrに変換するには

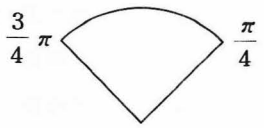
r(ラジアン)$=d$(度)$*\pi/180$

とします。

CRT ディスプレイ画面上での角度の関係を左図に示します。(右図は扇を描かせた場合の例を示しています。)



ks, keで指定する角度（ラジアン）



CIRCLE[7]200, 100, 100, , π/4, 3＊π/4, 0
を実行させた場合の表示例

例

CIRCLE 320, 100, 80 ……COLORのカラーコントロールに従い、半径80の円を描く。

CIRCLE[1, W0]320, 100, 80, , π/4, 3＊π/4 ……π/4～3π/4 ラジアンの円弧を青色で描く。

CIRCLE[2, W0]320, 100, 50, 0.35 ……赤色で楕円を描く。

CIRCLE[4]320, 100, 70, , π/2, π, 0 ……π/2～π ラジアンの扇を緑色で描く。

プログラム例

```
10 CONSOLE 6H,M:COLOR,07,W0:CCOLOR,0:PRINTCHR$(6):C=1
20 FOR H=0.05 TO 1  STEP .075
30 CIRCLE[C] 320,100,100,H
40 C=C+1:IF C=8 THEN C=1
50 NEXT H
60 C=1:K1=7*π/18:K2=11*π/18
70 FOR R=180 TO 60 STEP-20
80 CIRCLE[C] 100,160,R,,K1,K2,0
90 CIRCLE[C] 550,160,R,,π/3,2*π/3
100 K1=K1-π/36:K2=K2+π/36:C=C+1
110 NEXT R
120 END
```

2.4 漢字コマンド

注)漢字コントロールコマンドを使用する場合はオプションのMZ-1R13（漢字ROMボード）が必要です。

2.4.1 JTRANS

JTRANS (*x*)

x……ASCIIコード：数値データ(255以下の正整数)

機　能　ASCIIコードを対応するJIS漢字コード(10進数)に変換します。

解　説　ASCIIコード表にあるキャラクタを漢字パターンとして表示したいときこの関数を用います。コードの値が*x*であるASCIIキャラクタに対応する漢字のJISコードがこの関数で求められます。

ASCII コードについては付録 P.114　アスキーコード表を、JIS 漢字コードについては付録 P.119 JIS 漢字コード表をご参照ください。

*x*は１つのASCIIコードですので255以下の正の数値定数（10進数あるいは16進数）あるいは数値変数でなくてはなりません。

*x*として、コントロールコードやアルファベット反転文字のASCIIコード、あるいは対応する記号がJIS漢字コード表にないASCIIコードを指定するとエラーになります。

この関数による変換結果は10進数の数値データになります。値の範囲は8481（$2121）から 20351（$4F7F）までです。

例

```
10  A=JTRANS ($6D)
20  B=60
30  C=JTRANS (B+4)
40  D$="$%&"
50  E=JTRANS (ASC (D$))
60  F$=KCHR$ (JTRANS (ASC (MID$ (D$, 2, 1))))
70  COLOR  7,07
80  POSITION  0, 100
90  KPATTERN  F$
```

数値変数Aに"m"（ASCIIコード$6D）のJIS漢字コード9069（$236D）が入ります。

数値変数Cに"@"（ASCIIコード64）のJIS漢字コード8567（$2177）が入ります。

数値変数Eに"$"のJIS漢字コード8560（$2170）が入ります。

ストリング変数F$に"%"のJIS漢字キャラクタが入ります。

2.4.2 KCHR$

書　式　KCHR$ (x_1, x_2, ……, x_n)

x_i……JIS漢字コード：数値データ（10進整数）、もしくは4桁の16進数

機　能　JIS漢字コードに対応する漢字キャラクタを与えます。

解　説　数値データx_iについて、JIS漢字コードがx_iである1つの漢字キャラクタを与えます。

したがって、x_iは$2120（8480：10進）から$4F7F（20351：10進）の範囲の数値定数あるいは数値変数でなくてはなりません。

変換結果はストリング変数に代入することができます。1つの漢字キャラクタは2バイトで表現されますので、1つのストリング変数には最大127文字の漢字キャラクタが入ります。

ストリング変数はASCIIキャラクタと漢字キャラクタを区別していません。それゆえ、ストリング変数はその内容に無関係に自由に連結できます。ただし、ASCIIキャラクタは1バイト単位で格納されるのに対して漢字キャラクタは2バイト単位で格納されますので、それぞれを含むストリング変数を結合する場合はトータルの長さが255バイトを越えないように注意してください。

例

```
10  A$=KCHR$ ($3D55,  $3246)
20  B=20014
30  C=17726
40  D$=KCHR$ (B)+KCHR$ (C)
50  COLOR  4 , O6, W0
60  POSITION  20, 10
70  KPATTERN[2]A$, KCHR$ ($3D29, $455F)
```

ストリング変数A$には漢字キャラクタ〝春夏〟が代入されます。

ストリング変数D$には漢字キャラクタ〝流転〟が代入されます。

行番号70において、座標位置（20, 10）から〝春夏秋冬〟と表示されます。ただし、〝春夏〟は赤色で、〝秋冬〟は緑色で表示されます。(KPATTERN 文参照)

2.4.3 KINPUT

書　式　KINPUT［m, Wa］$x$$

m……カラーコード：数値データ（0～7）

a……色の重ね合わせ：数値データ（0または1）

$x$$ ……漢字コードが入力されるストリング変数：ストリングデータ

機　能　JIS漢字コードに対応する1つの漢字をカナ漢字変換をし、その文字列を入力します。

解　説　［m, Wa］の機能はSET文と同じです。これを省略するとCOLOR文のカラーコントロールにしたがいます。

指定されたポジションポインタの位置に漢字入力カーソルが表示されます。この場合最下段にもカナ入力カーソルが表示されます。そこへカナ文字を入力し、カナ漢字変換キー BREAK を押すと下段にそれと対応した漢字が表示されるので、そのカーソルをカーソルコントロールキー ← → ↑ ↓ により移動させることで漢字を選択します。その際、漢字入力カーソルの位置に選択した漢字が表示され、CR キーを押すとストリング変数に代入されます。該当漢字がない場合、CLR HOME キーを押すとその読みの漢字はキャンセルされカナ入力待ちの状態になります。入力した漢字が間違っていた場合、SHIFT キーと CLR HOME キーを同時に押すと消去できます。

入力を終えると SHIFT キーと CR キーを同時に押すことでKINPUTコマンドは終了します。

なお、カナ文字、ひらがな、英数字を入力する場合は、カナ入力カーソルが表示された時に、キーのモード切換えを行い、該当する文字を打ち CR キーを押すと直接ストリング変数に代入されます。キーのひらがな入力へのモード切換えは カナ キーを1回押すとひらがなモードに、2回押すと英数モードになり、3回押すとカナ入力モードに戻ります。画面の左下にある反転文字はキーのモードを表します。

1回に入力される文字数は、640×200ドットモードの場合最大39文字までで320×200ドットモードの場合最大19文字まで入力可能です。

1ラインで入力できる数が最大入力文字数となります。したがって一番左にポジションポインタをおくと39文字または19文字となります。

（注）KINPUT命令を実行するとハイリゾリューションモード時の（0，185）～（639，199）とノーマルリゾリューションモード時の（0，184）～（319，199）までのグラフィックエリア（画面の下側）はクリアされます。

例　KINPUT　A$……漢字入力をし、漢字文字列をA$に代入します。

詳しくはP.22カナ漢字変換機能の項をご参照ください。

2.4.4 KPATTERN

書　式　KPATTERN[m_1, Wa_1]$x_1$$, [m_2, Wa_2]$x_2$$, ……, [m_n, Wa_n]$x_n$$

m_i……カラーコード：数値データ（0～7）

a_i……色の重ね合わせ：数値データ（0または1）

$x_i$$ ……JIS漢字キャラクタを表わすデータの並び：ストリングデータ

機　能　指定した色で指定した漢字文字列のパターンをグラフィックエリアに描きます。

解　説　ストリングデータ$x_i$$で指定した漢字文字列のパターンがカラーコードm_iで指定したグラフィックエリアに描かれます。ストリングデータ$x_i$$は、関数KCHR$によってJIS漢字コードを漢字キャラクタに変換したもの、あるいは変換結果を連結したものでなくてはなりません。グラフィックエリアへの書き込み開始位置はPOSITION文で与えます。ストリングデータから実際に表示される漢字パターンへの変換はシステム中で自動的に行われます。なお、1つの漢字パターンは16×16ドットマトリクスで与えられます。

複数のストリングデータに対して、それぞれ別個にカラーコントロールを設定することができます。カラーコントロールを省略した場合は、COLOR文で指定したカラーコントロールにしたがって漢字パターンを描きます。

例

```
10  A$=KCHR$ ($3348)+KCHR$ ($4425)+KCHR$ ($2121)
20  Z$=KCHR$ (JTRANS (ASC (" 0 ")))
30  C$=KCHR$ (JTRANS (ASC (" M ")))+KCHR$ (JTRANS (ASC (" Z ")))+
        KCHR$ (JTRANS (ASC (" - ")))+KCHR$ (JTRANS (ASC (" 2 ")))+
        Z$+Z$+Z$
40  CONSOLE  GH
50  PRINT  CHR$ (6)
60  COLOR  2, O7, W1
70  POSITION  200, 100
80  KPATTERN  A$,  [5 , W0]C$
```

ハイリゾリューションモードで、画面の（200, 100）の位置から"拡張 MZ-2000"と漢字パターンで表示します。ただし、"拡張"を赤色で、"MZ-2000"を水色で表示します。

2.4.5 KPRINT/P

書式 1　使用するプリンタがMZ-1P06の場合

KPRINT/ P〈 e_1　d_1　e_2　d_2……e_n　d_n 〉

e_i……出力データまたはタビュレーション関数

d_i……セパレータ（；または，）

機　能　オペランドの出力並びで指定された漢字データをセパレータの機能等にしたがって並びの順にプリンタへ出力します。この命令はPSEL4を実行した場合に有効です。

解　説　PRINT／P、PRINT文の実行とほぼ同じ形式でデータ印字をプリンタ上へ行います。セパレータ〃，〃と〃；〃の働き、TAB関数の働きも同じです。

最初このコマンドを実行することによって漢字モードとなっています。

TABおよびセパレータは、キャラクタモード時の動作と同じです。

KPRINT／PとPRINT／Pを使用することにより同一行にキャラクタと漢字を混合印字することが可能です。

（注）漢字ROMボードMZ-1R13にはJIS漢字コードにない漢字が含まれていますが、その文字をKPRINT／P命令を使って漢字プリンタで印字することはできません。

書式 2　使用するプリンタがMZ-80P6、MZ-1P07、MZ-1P07Aの場合

KPRINT／P〈e_1　d_1〉

e_1……出力データ

d_1……セパレータ（;）

機　能　オペランドの出力並びに指定された漢字データをセパレータの機能等にしたがってプリンタへ出力します。この命令はPSEL0、PSEL1、PSEL2、PSEL3を実行した場合に有効です。

解　説　PRINT／P、PRINT文の実行とほぼ同じ形式でデータ印字をプリンタ上へ行います。セパレータ〃;〃の働きは同じです。

KPRINT／PとPRINT／Pを使用することにより同一行にキャラクタと漢字を混合印字することが可能です。

MZ-1P07、MZ-1P07A、MZ-80P6ではKPRINT／Pを実行し終るとプリンタの紙送りピッチは1/6〃に設定されます。

（注）MZ-1P07、MZ-1P07A、MZ-80P6に対してKPRINT／P（書式2）が実行できますが、MZ-80P4B、MZ-80BP5に対してはKPRINT／P（書式1、2）は実行しないでください。

MZ-1P07、MZ-1P07A、MZ-80P6に対し、KPRINT／P命令においてTAB関数は使用できません。

漢字ROMボードMZ-1R13にはJIS漢字コードにない漢字が含まれていますが、その文字をKPRINT／P命令（書式2）を使ってプリンタで印字することが可能です。

またセパレータとして〃，〃は使用できません。

2.5 RS-232C コマンド

注) RS-232C ステートメントを使用する場合はオプションの RS-232C シリアルインターフェイス MZ-8B103 が必要です。

2.5.1 RSMODE

書　式　RSMODE *a*, R*b*, T*c*, M*d*, RX*e*

a ：チャンネル指定

a	チャンネル
A	Aチャンネル
B	Bチャンネル

b ：受信キャラクタのビット数指定

c ：送信キャラクタのビット数指定

b, *c*	ヒット／キャラクタ
5	5
6	6
7	7
8	8

d ：パリティビットの有無とストップビット数の指定

d	パリテイ	ストップビット
69	奇数	1
70	無	
71	偶数	
73	奇数	1½
74	無	
75	偶数	
77	奇数	2
78	無	
79	偶数	

e ：受信可、不可の指定

e	受信
0	不可
1	可

機　能　前記パラメータにより各モードの設定を行います。

解　説
- すべてのパラメータを指定する必要はありませんが、受信可／不可の指定は各パラメータ指定の必ず最後におかなければなりません。
- モード設定パラメータa～eの指定は前記以外のものは使用しないこと。
- BASIC起動時には各チャンネル共、次のモードに設定されています。

b, *c*	8	キャラクタビット数8
d	79	偶数パリティ／ストップビット2
e	0	受信不可

例
```
10 RSMODE A, RX1……チャンネルAを受信可能とする。
20 RSMODE A, R7, T7, M70, RX1……チャンネルAに対して送受信のキャラクタを
                                7ビットとし、パリティ無しのストップビット
                                を設定し受信可能とする。
```

2.5.2 RSO

書　式　RSO *x* *y*$

x：チャンネルの指定(AまたはB)

y$：ストリング変数で転送データを指定。

機　能　チャンネル*x*へ*y*$で指定されるデータを送信します。

例
```
10 X$ = "Demonstration"
20 RSO B X$……チャンネルBへX$のデータを転送します。
```

2.5.3 RSI

書　式　RSI *x* *y*$

x：チャンネルの指定（AまたはB）

y$：受信データを格納するストリング変数

機　能　チャンネル*x*からデータを受信し、ストリング変数*y*$に格納します。

例
```
10 RSMODE A, RX1 ……チャンネルAを受信可能とする。
20 RSI A B$ ……チャンネルAからデータを受信する。
30 PRINT B$
```

GP-IBステートメント

Chapter 3

本章ではGP-IBインタフェースカードMZ-8BI04を用いて、計測器等をコントロールする場合のBASICステートメントに関する内容を含んでいます。
GP-IBに関する詳細は、GP-IBインタフェースの説明を参照ください。

なお、GP-IBインタフェースの解説書は、MZ-80B用に作成されていますが、MZ-2000／2200用としても使用できます。ただし、スレーブモードにおけるサンプルプログラム例の中で、BASICとリンクするための番地が定義されていますが、BASICインタープリタ5Z007を使用の際は、下記の通り、リンキングアドレスおよびインタラプトテーブルのアドレスを変更する必要がありますので、注意願います。

GP-IB　インターフェイス解説書

P.84（リンキングアドレスの変更）

```
45 F000  P      HBRK:   EQU   15A2H
46 F000  P      FGLNK1: EQU   6597H
47 F000  P      FGLNK2: EQU   65B8H
48 F000  P      MLADRS: EQU   65B6H
49 F000  P      MTADRS: EQU   65B7H
```

番地を変更

```
15A6H
7C22H
7C43H
7C41H
7C42H
```

P.85（インタラプトテーブルのアドレス変更）

```
39 F02F 220066     LD (6600H),HL
                番地を変更
39 F02F 22808C     LD (8C80H),HL
```

ステートメントの書式

■ステートメントを、アルファベット小文字、反転文字で表記することはできません。

■かぎカッコ〈 〉中の要素は、省略あるいは、任意回の繰り返し記述が可能な要素であることを示しています。実際のプログラムにこのカッコをタイプしてはいけません。

■セパレータ（コロン、セミコロンなど）は、決められた位置に正しく置かなくてはなりません。

3.1 プログラム作成上の注意

BASICを使用すると、大部分のGP-IBに特有なコマンドを意識することなく、プログラムの作成が可能です。しかし、規格で定められたコマンドには、コマンドを送出する順序が決められており、この順序を誤ると、計測器は動作しません。また、計測器には、それぞれアドレスがつけられており、このアドレスは1つのシステム内に同じアドレスを持つ計測器が2台以上存在すると、不適当な動作をします。

さらに、制御される側の計測器についても、それぞれ、動作に対応したコードが決められており、このコードは同じメーカの同じ計測器以外は、すべて異なっています。従って、プログラムを作成する場合には、測定器の取扱い説明書を良く読み、理解しておくことが必要です。また、データの終了を示すコードについても同様で、EOIだけのものもあれば、EOIをださずASCIIコードで代用するもの、この2つの併用型等、各種あります。これらはすべて各々の計測器に個有のものであり、規格の定める範囲外にあり、また、BASICコマンドにも含まれずデータとして送るストリングデータに含まれますので、これらのことを考えながらプログラムを作成する必要があります。

アドレス

各計測器、機器にはそれぞれ個有のアドレスが設定されており、このアドレスはかえられるようになっています。

1つのシステム内に同じアドレスを持つ計測器があると誤動作の原因になりますので注意が必要です。

データのデリミタ

各計測器には、データの終了を判断するためのコード、すなわちデリミタが設定されていますが、これは各種類あり、極端な場合、同一測定器であっても動作モードによって、デリミタが異なる場合があります。

MZ-2200では初期設定として、0DH-0AHの2バイトがデリミタとして定義されていますが、使用する計測器のデリミタに合せる必要があります。

さもなければ、データ転送が途中で停止し、動作しなくなります。

デリミタは計測器の取扱い説明書に示してありますから、それを参照してください。

ステートメントの実行順序

シリアルポール時の "ON SRQ" パラレルポールの "PPC *n*, *l*" アドレスの設定、デリミタの指定、ICL、REN等はプログラムの最初で実行しておかなければ、全く用をなさない場合がありますので注意してください。

計測器の応答がないとき

BREAKキーによりプログラムを停止させて、プログラムエラーを修正して再度RUN実行してください。

ウェイトルーチン

計測器にはデータの送受の準備完了までの待ち時間を設定する必要があります。

MZ-2200 では WAIT スデートメントがありませんので、FOR NEXT ループで適当な待ち時間を設定してください。

相手側にPCT機能がないとき

PCT ステートメントによって、他のコントローラにコントローラの権利を渡した後、そのコントローラが再度 MZ-2200 にコントローラの権利を戻す機能がないとき、ICL を実行すると相手のコントローラは初期設定され、MZ-2200 にコントローラの権利が戻ります。

ただし MZ-2200 はシステム・コントローラであることが必要です。

GP-IBステートメントリスト

ステートメント	内　　容
1．ICL	MZ-2200がシステムコントローラとして、インタフェース全体を初期状態に設定。
2．REN	MZ-2200がシステムコントローラとして、インタフェース全体をリモート・モードに設定します。
3．LCL	MZ-2200がシステムコントローラとして、インタフェース全体をローカル・モードに設定します。
4．LCL *n*	MZ-2200がコントローラとして、インタフェース・バス上の *n* で指定される機器をローカルに戻します。
5．LLO	MZ-2200がコントローラとして、インタフェース・バス上の機器すべてにローカルに戻ることを禁止します。
6．DCL	MZ-2200がコントローラとして、インタフェース・バス上の機器すべてを、機器側で定められている初期状態に設定します。
7．DCL *n*	MZ-2200がコントローラとして、インタフェース・バス上の *n* で指定される機器を機器で定められている初期状態に設定します。
8．TRG *n*	MZ-2200がコントローラとして、インタフェース・バス上の *n* で指定される機器を動作開始させます。
9．PCT *n*	MZ-2200がコントローラとして、バス上の他のコントローラ機能を持つ機器にコントローラの権利を渡します。その後はコントローラとしての動作はエラー。
10．WRT *n*, *v*	MZ-2200はコントローラとして、*n* で指定される機器をリスナに、トーカを自分に設定し、*v* で指定するデータを送出します。
11．RED *n*, *v*	MZ-2200はコントローラとして、*n* で指定される機器をトーカに、リスナに自分を設定し、*v* に受信データを格納します。
12．WRT/ *v*	MZ-2200がトーカとして、*v* で指定するデータを送出します。
13．RED/ *v*	MZ-2200がリスナとして、*v* で指定される変数に、受信データを格納。
14．CMDW *x*$	MZ-2200がコントローラとして *x*$ で指定するコマンドを送出し、その後MZ-2200はトーカとなります。
15．CMDR *x*$	MZ-2200がコントローラとして *x*$ で指定するコマンドを送出し、その後MZ-2200はリスナとなります。
16．ON SRQ	バス上の機器のどれかがサービス要求をしている場合、その処理ルーチンにジャンプ。
17．SPOL *n*, *v*	バス上の機器 *n* に対して、MZ-2200がコントローラとして、シリアル・ポールを実行し、ステータスバイトを変数 *v* に格納します。
18．PPC *n*, *l*	MZ-2200がパラレル・ポールを実行する時に、バス上の機器が応答するステータスビットを *l* で、機器アドレスを *n* で指定します。(10進で指定)
19．PPOL *v*	MZ-2200がコントローラとしてパラレルポールを実行し、ステータスビットを変数 *v* に格納します。
20．PPU	MZ-2200がパラレル・ポールモードを解除します。
21．GPIBM *n*	MZ-2200が常にシステムコントローラであり、アクティブコントローラであって、自分のアドレスを *n* で指定します。
22．EOIW *n*	データ送出時のデリミタ指定。
23．EOIR *n*	データ受信時のデリミタ指定。

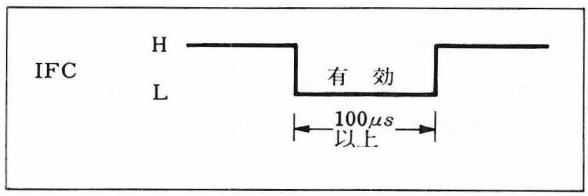
3.2 GP-IB ステートメント

MZ-2200がBASICを起動したとき、システムコントローラ、アクティブコントローラ、アドレスNo. 1に設定されますが、I／O動作は何もしていません。

3.2.1 ICL (Interface Clear)

書　式　ICL

機　能　IFCラインに100μs以上の負のパルスを発生させ、インタフェース全体を初期設定します。

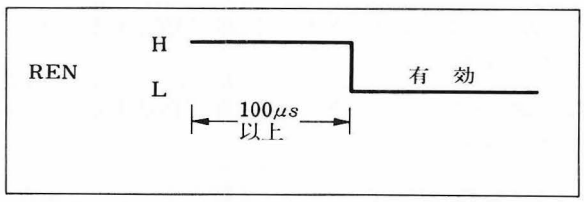


注　意
- MZ-2200がシステムコントローラでないときは、エラーとなります。
- 変数は不要です。

3.2.2 REN (Remote Enable)

書　式　REN

機　能　RENラインを100μs間Highレベルにした後、RENラインをLowレベルに設定し、バス上の機器をリモートモードに設定します。

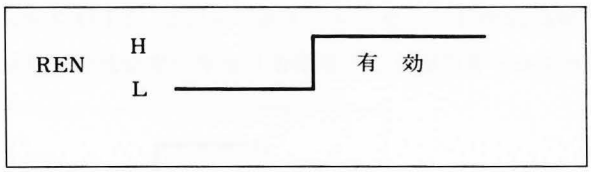


注　意
- MZ-2200がシステムコントローラでないときはエラーとなります。
- 変数は不要です。

3.2.3 LCL (Local)

書　式　LCL

機　能　RENラインをHighレベルに設定し、バス上の機器をすべてローカルモード（手動状態）に戻します。



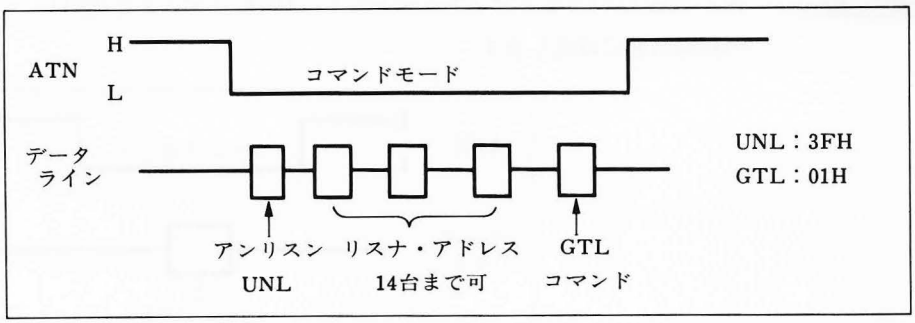
注　意
- MZ-2200がシステムコントローラでないときはエラーとなります。
- 変数は不要です。
- 再度、測定器を制御するときにはRENコマンドの実行が必要です。

3.2.4 LCL n (Go to local No.n)

書　式　LCL n_1 <; n_2 ……; n_k>

n_i：装置番号（リスナ）

機　能　MZ-2200はATNラインをLowレベルに設定し（コマンドモード）、リスナを解除し、n_iで指定された機器のリスナアドレスを送出した後にGTLコマンドを送り、指定されたリスナをローカルモードにします。

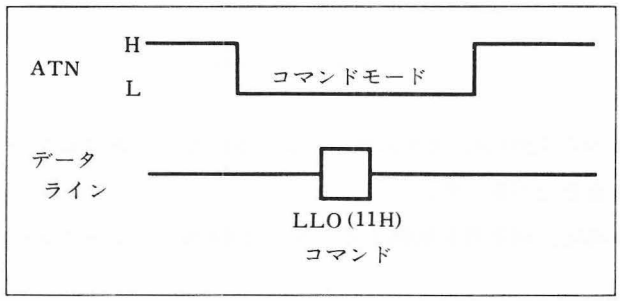


注　意
- n_iは0 ～30までの10進数での指定が可能です。
- 指定可能なリスナの台数は14以内。
- MZ-2200がシステムコントローラでないときはエラーとなります。

3.2.5 LLO (Local Lock-out)

書　式　LLO

機　能　MZ-2200はバスをコマンドモードにし、LLOコマンドを送り、バス上の機器すべてに対してローカルモードに戻ることを禁止します。(すなわち、フロントパネル操作を無効にさせます。)

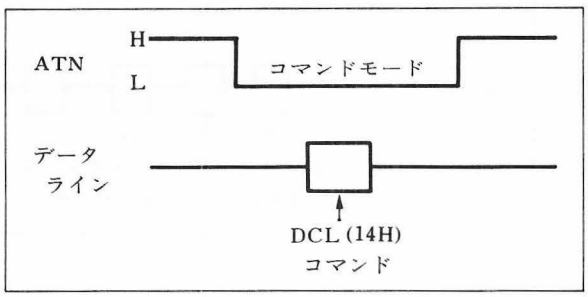


注　意
- MZ-2200がアクティブコントローラでないときはエラーになります。
- 変数は不要です。

3.2.6 DCL (Device Clear)

書　式　DCL

機　能　MZ-2200はバスをコマンドモードにし、DCLコマンドを送出し、バス上の機器すべてを機器個有の初期状態に設定します。



注　意
- MZ-2200がアクティブコントローラでないときはエラーとなります。
- 変数は不要です。
- 機器個有の初期状態は各計測器のマニュアルを参照ください。

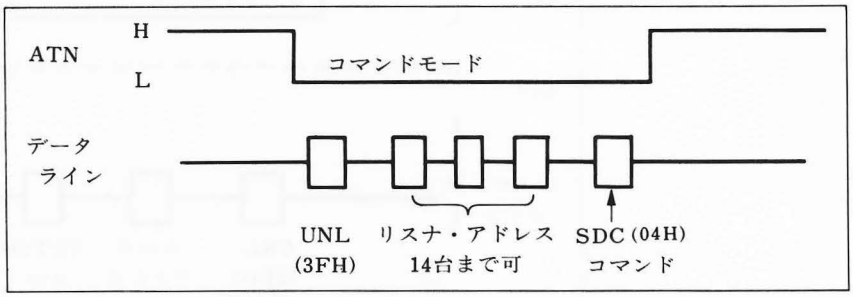
3.2.7 DCL n (Selected Device Clear)

書　式　DCL n_1〈; n_2 ;…… ; n_k〉

n_i：装置番号（リスナ）

機　能　MZ-2200はバスをコマンドモードにし、UNLコードによってすべてのリスナを解除した後、n_iで指定される機器のリスナアドレスを送り、続けてSDCコマンドを送出し、指定したリスナを機器個有の初期状態に設定します。

指定台分のリスナが初期設定されます。



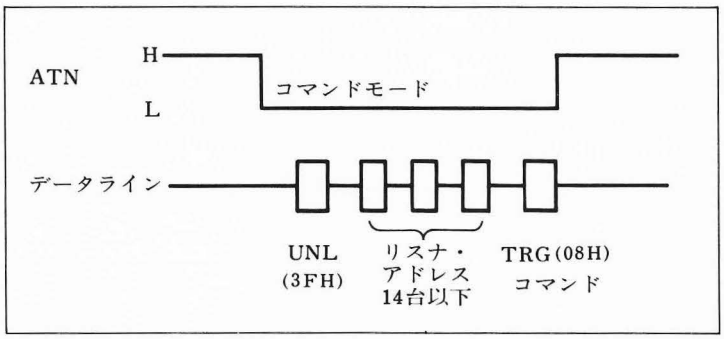
注　意
- n_iは 0 ～30までの10進数で指定します。
- 指定可能なリスナ台数は14以下です。
- MZ-2200がアクティブコントローラでないときはエラーとなります。

3.2.8 TRG (Trigger)

書　式　TRG n_1〈; n_2 ; …… ; n_k〉

n_i：装置番号（リスナ）

機　能　MZ-2200はバスをコマンドモードにし、リスナ解除した後n_iで指定される機器のリスナアドレスに続けてTRGコマンドを送り、指定した台分のリスナを動作開始（測定開始）させます。



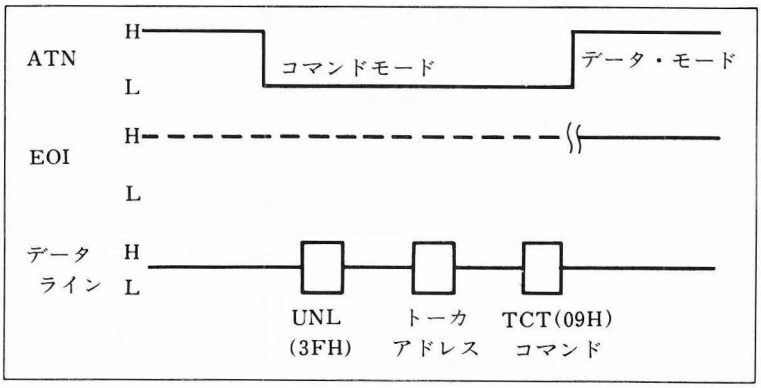
注　意
- n_iは 0 ～30までの10進数が使用可能です。
- 指定するリスナの台数は14以下です。
- MZ-2200がアクティブコントローラでないときはエラーとなります。

3.2.9 PCT (Pass Control)

書　式　PCT *n*

n：装置番号（トーカ）

機　能　MZ-2200はバスをコマンドモードにし、リスナを解除した後、コントローラ機能をもつ*n*で指定される機器のトーカアドレスに続けてTCTコマンドを送り、その機器にコントローラの権利を渡す。



注　意
- *n*はコントローラ機能をもつ機器の番号であり、指定個数は1。
- コマンドモードではMZ-2200はアクティブコントローラですが、データ・モードになったときバス上で単なるリスナまたはトーカに指定されるまでコマンドは送出できません。ICL実行の場合に、アクティブなコントローラに復帰できます。

3.2.10 WRT (Write)

書　式

WRT n_1〈; n_2 ; …… ; n_k〉, v_1〈v_2, ……, v_m〉

n_i：装置番号（リスナ）

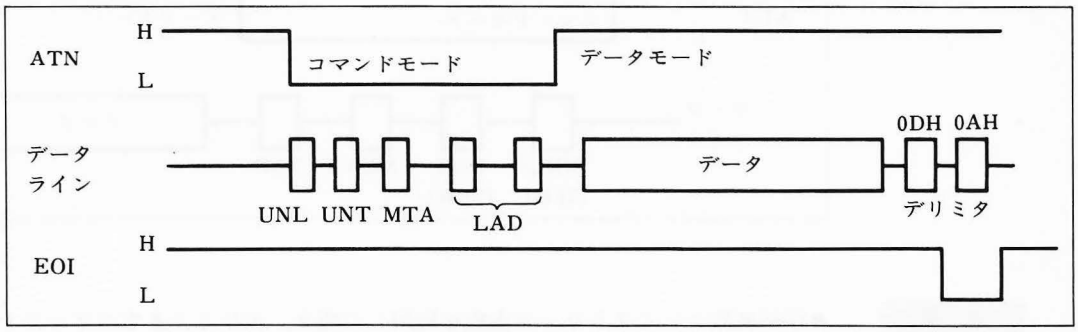
v_j：転送データ（ストリング変数、数値変数または定数）

機　能

指定したリスナに対して指定したデータを転送します。

MZ-2200はコントローラとして、コマンドモードにおいて自分をトーカに指定し、n_iで示される機器をリスナに指定します。

コマンド送出後、MZ-2200はバスをデータモードにし、リスナにv_jで指定されるデータを送出します。データ送出後、自動的にデリミタコード0DH-0AHおよびEOIを下図のタイミングで送出します。



注　意

- 転送データとして数値を用いた場合、ストリングに変換されてASCIIコードで出力されます。転送データが複数個指定された場合は一連のASCIIコードとして連続して出力されます。
- 装置番号n_iは0～30までの10進数が有効です。
- 指定可能なリスナ台数は14以下です。
- コントローラインチャージ（能動になれるコントローラ）の場合に実行可。
  パスコントロール後に実行するとエラーとなります。

例

```
10 WRT 5, "SP1;" ……5番の装置に "SP1;" が送られる。

10 MZ=3 : PC=4 : D$ = "ABC"
20 WRT MZ;PC, D$……3番、4番の装置に "ABC" を送ります。

10 PL=5 : PN=4
20 WRT PL, "SP", PN, ";" ……5番の装置に "SP4;" を送ることになります。
```

3.2.11 RED (Read)

書　式

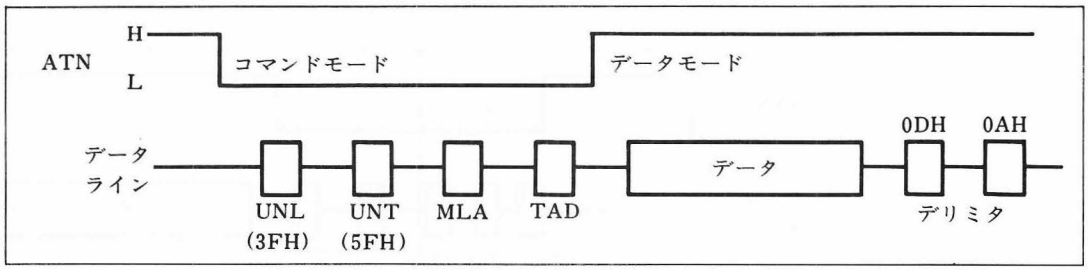
RED n, v_1〈, v_2, ……, v_k〉

n：装置番号（トーカ）

v_i：受信データ格納のための変数

機　能

指定のトーカからデータを受信し、指定の変数に格納します。MZ-2200はコマンドモードにおいて自分をリスナに指定し、nで示される装置をトーカに指定し、データモードにおいてトーカから送られるデータを入力します。EOIラインは1バイトごとにチェックし、EOIがLowレベルであれば、そのときのデータライン上のデータを受信して終了します。しかしEOIがHighレベルであればデリミタコード(0DH-0AH)を受信したときにデータ受信を終了します。



注　意

- 格納変数としてストリング変数を使用した場合、デリミタまでのデータ（ただし253文字以内）を格納するため、ストリング変数は1つのみ使用できます。
- 数値変数を用いた場合、一連のデータのうち、コンマ(,)またはデリミタまでを数値に変換して各々の変数に順次格納します。(変換できない場合はエラーとなります。)
- 複数の数値変数を用いた場合、データの個数と変数の個数が一致している必要があります。
- 測定器から送られてくるデータのフォーマットは統一されていないので、この命令を使ってデータを入力する場合、受信データのフォーマットに注意する必要があります。

例

```
10 RED 5, A$ ……5番の装置からデータを入力し、A$に格納する。

10 FC=24
20 RED FC, F1, F2 ……送られてきたデータが100, 240とすればF1=100,
                      F2=240となる。
```

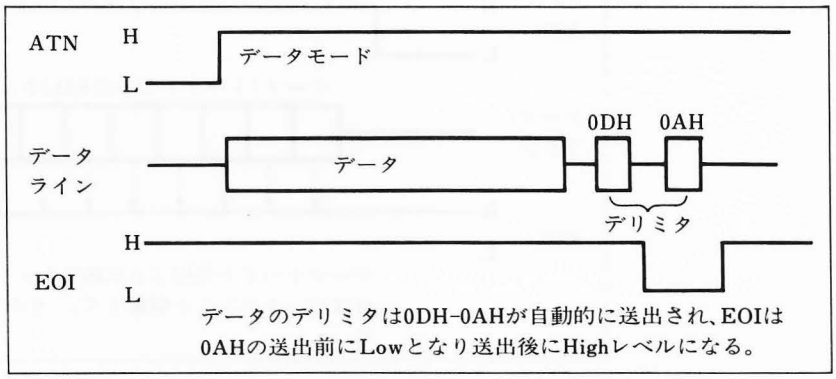
3.2.12 WRT/

書　式　WRT/v_1<, v_2, …, v_k>

v_i：転送データ

機　能　自分がトーカに指定されることを確認し、指定されていればデータを送出します。指定されていなければエラーとなります。

転送データのフォーマットに関してはWRT n, vと同様です。同一の装置に多数のデータを送る場合、一度トーカ、リスナを指定しておけば、そのトーカ，リスナは解除あるいは再指定まで有効ですから、WRT/命令によりデータモードのみでデータの転送ができます。



例

```
10 WRT 9, A$
20 FOR I=1 TO 100
30 WRT/A$
40 NEXT I
```

9番の装置にA$の内容を101回送る。

3.2.13 RED/

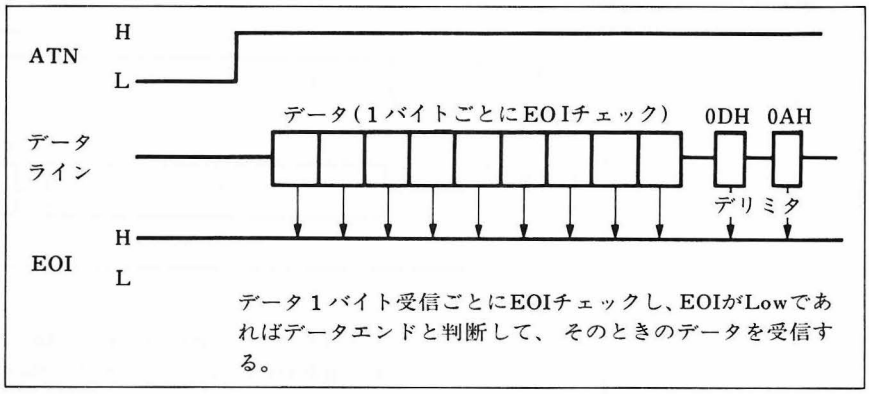
書　式　RED/v_1<, v_2, …, v_k>

v_i：データ格納のための変数

機　能　自分がリスナに指定されていることを確認し、指定されていればデータを入力し、変数に格納します。リスナに指定されていなければエラーとなります。

データ格納変数に関しては、RED n, vの場合と同様です。

WRT/命令の場合と同様に、同一のトーカからのデータを何回も受信する場合に、この命令は有効です。



例

```
10 DIM A$(100)
20 RED 13, A$(0)
30 FOR I=1 TO 100
40 RED/A$(I)
50 NEXT 1
```

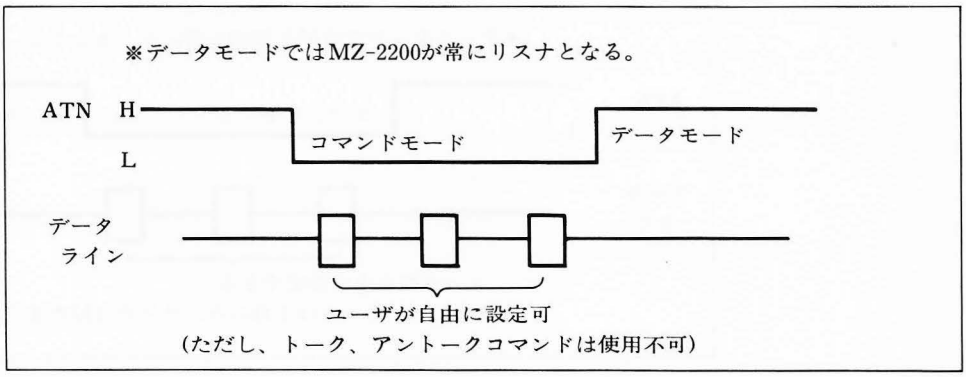
13番の装置からデータを101回入力し、A$(　)に入れる。

3.2.14 CMDW (Command Write)

書　式　CMDW $x$$

$x$$：転送データ

機　能　MZ-2200はコマンドモードにおいて自分自身をトーカに指定し、$x$$で示されるコマンドに対応するストリングをバスに送出した後、データモードに設定する。



注　意

- $x$$中にはUNT(5FH)、TAD(40H～5EH) は使用しないでください。
- 同じリスナと同じトーカ間で何回もデータを転送するときに、このステートメントを使用すると便利です。コマンドの設定が1回ですみます。
- 2次アドレスを持つ機器のアドレスが可能です。
- このステートメントはユーザが自由に使用できますが、規格(IEC,IEEE-488)の内容を理解しないで使用すると全く動作しない場合があります。

例

```
10 A$=CHR$($3F)+CHR$($25) ……UNL、およびプロッタをリスナ
20 B$= "LBAAA" +CHR$($03)+CHR$($3B) ……デリミタ+デリミタ
30 CMDW A$ ……トーカMZ-2200、リスナプロッタに指定
40 FOR I=0 TO 100
50 WRT/B$ ……101回データを書く
60 NEXT
```

3.2.15 CMDR (Command Read)

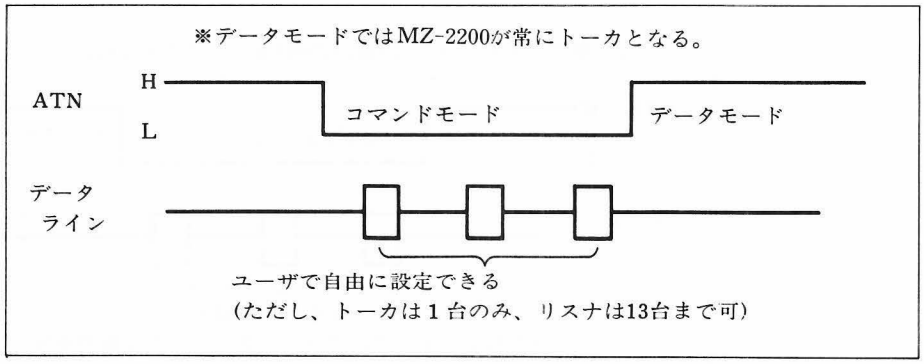
書　式

CMDR $x$$

$x$$ ：受信データ

機　能

MZ-2200はコマンドモードにおいて自分自身をリスナに指定し、$x$$で指定するコマンドに対応するストリングをバスに送出した後、データモードに設定する。



注　意

- MZ-2200がリスナに指定されているので、トーカを1台指定すれば、ユーザが指定できるリスナ台数は13台までとなります。
- 同じリスナ、同じトーカ間でデータを何回も転送する際のコマンド指定にこのステートメントを使用すると便利です。
- 2次アドレスの送出に使用できます。
- このステートメントはユーザが自由に使用できますが、規格(IEC,IEEE-488)の内容を理解しないで使用すると全く動作しない場合があります。

例

```
10 A$=CHR$($3F)+CHR$($5F)+CHR$($51)
          ↑          ↑          ↑
         UNL        UNT     トークアドレス

20 CMDR A$ ……… MZ-2200をリスナ、HP-3325A(F.G)をトーカに指定
30 RED/B$ ……… トーカからのデータをB$に格納
40 PRINT B$
```

3.2.16 ON SRQ

書　式　ON SRQ l_n

l_n：行番号

機　能　このステートメントを最初に実行しておくと、バス上の機器が、サービスを要求してきたときにlnで示される行番号からはじまる処理ルーチンへジャンプします。

注　意　このステートメントは1回のみ有効であり、処理ルーチンの実行完了後、再度必要な場合には改めてこのステートメントを設定しなければなりません。

3.2.17 SPOL（Serial Polling）

書　式　SPOL n, v

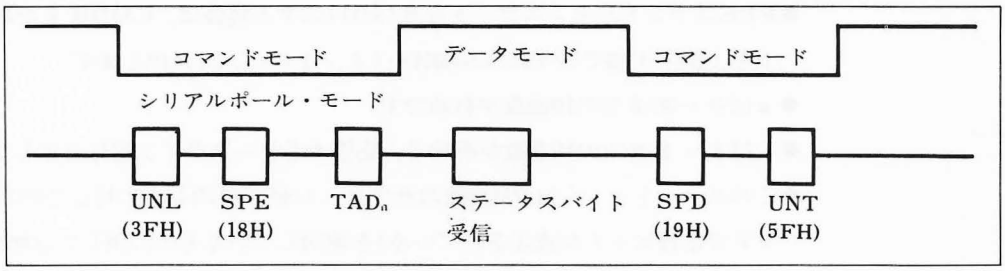
n：装置番号（トーカ）

v：ステータスバイト

機　能　MZ-2200はコントローラとしてリスナ解除し、SPEコマンドにつづけてnで指定される機器のトーカアドレスを送り、データモードにします。

データモードにおいて指定されたトーカからのステータスバイトを受信し変数vに格納します。

再度コマンドモードに設定し、SPEコマンド、UNTコマンドを送り、シリアルポールモードを解除します。



注　意　指定トーカ1台のみで、nで示されるアドレスは0～30までの10進数が有効です。

例

```
10 ON SRQ 150 : N = 5
20 ICL : REN : DCL
30 WRT 5, "IM252, 32, 16;" ……エラーが生じたときSRQを発生させる。
40 WRT 5, "SP1; SP2; SP9;" ……SP9でエラーが発生。
50 WRT 5, "IN;" : GOTO 900 ……エラーランプ消去。
150 SPOL N, A ……Nで指定されるトーカのステータスを変数Aに格納。
160 IF A>=64 THEN 50 ……Aが40H以上であれば要求源確認。
170 N = N+1 : GOTO 150 ……次のトーカをチェックする。
900 END
```

3.2.18 PPC (Parallel Poll Configure)

書　式

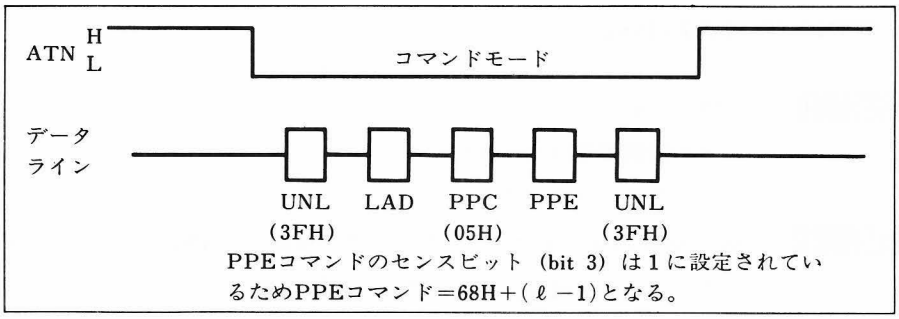
PPC n, l

n：装置番号（リスナ）

l：データライン番号

機　能

MZ-2200はコントローラとしてバスをコマンドモードにしリスナを解除し、nで指定される機器のリスナアドレスを送り、続けてPPCコマンドを送出し、さらに（l－1）+68Hで示されるPPEコマンドを送出し、パラレルポール時にリスナに指定した機器が応答するデータラインを割り振ります。



注　意

- PPCステートメントはプログラムでPPOLを実行する以前に実行しておくことが必要です。
- PPEコマンドのセンスビットを0（ゼロ）にする場合は、CMDWまたはCMDRを用いてコマンドを上図の手順で(PPEのみ60H+(l－1)として)送出します。
- nは0～30までの10進数が有効です。
- lは1～8までの10進数が有効で、応答するデータライン番号を示します。
- このステートメントはPP機能のサブセットがPP1のものに対してのみ有効で、PP2（ハードウェアで応答ビットが決定されている）を使用しているものに関しては無効です。
- PPEコマンドの解除はPPD(70H)コマンドを送出しなければなりません（CMDWまたはCMDRを用いること。）

3.2.19 PPOL (Parallel Poll)

書　式

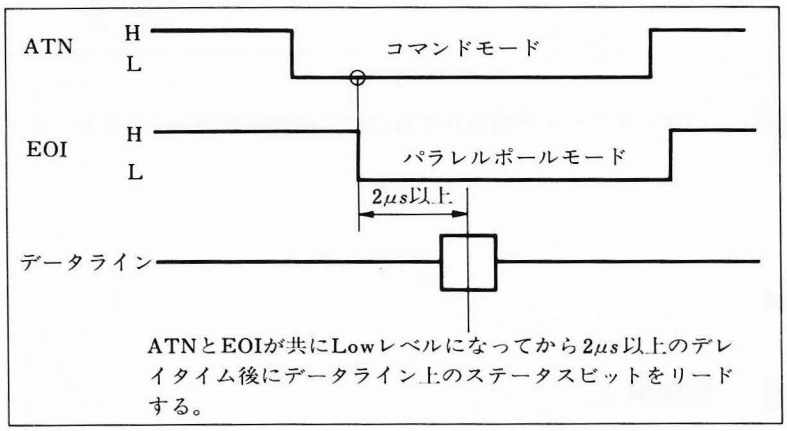
PPOL *v*

v ：ステータスビット格納変数

機　能

MZ-2200はコントローラとして任意の時点でATNとEOIラインを共にLowレベルに設定してパラレルポールを実行します。

この時のみ3線ハンドシェークを使用しないでリスナがステータスビットを設定する時間（約2 μs）だけ待った後、データライン上のビットパターンを変数 *v* に格納します。



注　意

変数 *v* は10進数で 0 ～255までの範囲です。

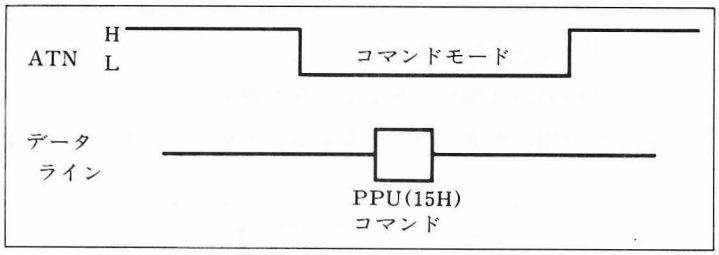
例

```
10 ON SRQ 100
20 ICL : REN : DCL
30 WRT 5, "IM252, 32, 16;" ……エラーが生じたらSRQを発生する。
40 WRT 5, "SP1; SP2; SP7;" ……SP7でエラーが生じ、SRQが発生したので
50 WRT 5, "SP0; IN;" : GOTO 900 …プロッタを初期設定して終了。
100 PPOL A ……パラレルポール実行。
110 IF A=4 THEN 50 ……A＝4ならOK。
120 PRINT "ERROR" ……A≠4ならエラー表示。
900 END
```

3.2.20 PPU (Parallel Poll Unconfigure)

書　式　PPU

機　能　MZ-2200はコマンドモードにおいてPPUコマンドを送出しパラレルポールモードを解除します。



注　意　パラレルポールを再実行するためには再度PPC*n,l* のステートメントを実行しなければなりません。

3.2.21 GPIBM

書　式　GPIBM *n*

n：装置番号

機　能　MZ-2200をシステムコントローラ、アクティブコントローラに設定し、*n*で指定される0～30までの10進数を自分の装置番号とします。

3.2.22 EOIW

書　式　EOIW n_1　または　EOIW n_1, n_2

n_1　n_2：デリミタ

機　能　データ送出時のデリミタとして2個までの10進数で指定します。

注　意
- 3個以上のデリミタは無視します。
- EOIは最後のデリミタの送出前後の間だけLowレベルになります。

3.2.23 EOIR

書　式　EOIR n_1　または　EOIR n_1, n_2

n_1、n_2：デリミタ

機　能　データ受信時のデリミタとして2個までの10進数で指定します。

注　意
- 3個以上のデリミタは無視します。
- EOIラインがLowレベルであれば、そのときのデータライン上のデータを受信後、終了します。

BASIC(5Z007)のまとめ

Chapter 4

この章は、BASIC（5Z007）の全てのコマンド、ステートメント、関数、各種オペレータを、この順にまとめています。

4.1 コマンド

DIR	DIR	ディスクのディレクトリ（directory）をCRT画面上に表示します。ディレクトリ表示上に示される情報は、登録されているファイルのモード、ファイル名です。 注意：CRT画面上のディレクトリ表示は、ファイルを1画面ぶん表示すると一旦止まり、カーソルが現れます。更に続けてディレクトリ表示を行うには CR キーを押しますが、途中で他のコマンドへ移ることもできます。
DIR/P	DIR /P	ディスクのディレクトリをラインプリンタ上に印字します。
LOAD	LOAD " A "	" A " というファイル名のついたBASICテキスト（BTX）を読み出します。
	LIMIT $D000 : LOAD " B "	BASICテキストとリンクするための機械語プログラムファイル（OBJ）を読み出す場合は、LIMIT命令によってBASICエリアと機械語エリアとを分離しておく必要があります。機械語プログラムとのリンク命令を参照のこと。
LOAD/T	LOAD/T " C "	" C " というファイル名のBASICテキストをカセットテープから読み出します。 注意：LOADコマンドまたはLOAD/Tコマンドによって BASICテキストファイルの読み出しを実行すると、それ以前にテキストエリアにあったプログラムは無効になります。
APPEND	APPEND " I "	BASICテキストエリアにあるプログラムテキストと、" I " というファイル名のついたBASICテキストを混ぜ合わせます。
SAVE	SAVE " D "	現在テキストエリアにあるBASICテキストを、" D " というファイル名を付けてディスクに書き込みます。ファイル名" D "、ファイルモードBTXのファイルが1つ登録されます。
SAVE/T	SAVE/T " E "	現在テキストエリアにあるBASICテキストを、" E " というファイル名を付けてカセットテープに書き込みます。
RUN	RUN	現在テキストエリアにあるBASICテキストの先頭からプログラムを実行します。 注意：RUNコマンドでは、プログラムの実行直前に、すべての変数の内容を0または空（null）とします。
	RUN 1000	行番号1000からプログラムを実行します。
	RUN " F " (BTX)	" F " というBASICテキストファイルを読み出し、つづいて、テキストの先頭からプログラムを実行します。
	RUN " G " (OBJ)	ディスクから機械語プログラムテキスト " G " を読み出し、つづいてその指定された実行アドレスからプログラムを実行します。この場合、システムのコントロールは、BASICから離れることになります。

VERIFY	VERIFY "H"	現在BASICテキストエリア内にあるプログラムテキストとファイル名"H"で指定するカセットテープファイルの内容を比較します。
AUTO	AUTO	テキスト作成時に、行番号を、10、20、30……と自動的に発生します。
	AUTO 200, 20	行番号を、200から20おきに、200、220、240……と自動的に発生します。AUTOコマンドは、BREAKキーを押すことにより解除されます。
DELETE ;		プログラム、ファイルの削除を実行します。
	DELETE 10-230	現在テキストエリアにあるBASICテキストの行番号10～230までのプログラムを削除します。
	DELETE -150	プログラムの最初から行番号150までを削除します。
	DELETE 20	行番号20以降のプログラムを削除します。
	DELETE 5	行番号5のプログラムを削除します。
RENUM	RENUM	現在テキストエリア内にあるBASICテキストの行番号を10, 20, 30…という行番号に変更します。
	RENUM 100, 20, 50-150	行番号50～150までを100から20おきの行番号に変更します。
	RENUM 1000	プログラムを行番号を1000から10おきの行番号に変更します。
	RENUM , , 100	行番号100を行番号10に変更します。
	RENUM 50, 10	プログラム行番号を50から10おきの行番号に変更します。
		(注) RENUMの結果、行の順序が入れ換わるような範囲を指定することはできません。
LIST	LIST	現在テキストエリア内にあるBASICテキストの全リストを表示します。
	LIST-500	行番号500までのリストを表示します。
LIST/P	LIST/P	リスト表示をラインプリンタ上へ行います。
SEARCH		現在BASICテキストエリア内にあるプログラムテキストの全部、または一部に対して指定した文字列を含む行を探し出しCRT上へ表示します。
	SEARCH "ABC"	プログラムテキスト全部に対して文字列"ABC"を含む行を探し出しCRT上へ表示します。
	SEARCH "ABC",100-	行番号100以降の文に対して文字列"ABC"を含む行を探し出しCRT上へ表示します。
	SEARCH "ABC", -500	プログラムの先頭から行番号500までの文に対して文字列"ABC"を含む行を探し出しCRT上へ表示します。
	SEARCH "ABC", 100-500	行番号100～500の文に対して文字列"ABC"を含む行を探し出しCRT上へ表示します。
SEARCH／P		指定した文字列を含む行番号を探し出しプリンタ上に印字します。
	SEARCH／P "ABC",100-200	行番号100～200の文に対して文字列"ABC"を含む行を探し出しプリンタ上に印字します。

NEW	NEW	現在テキストエリア内にあるBASICテキストを消去し、変数エリアをクリアします。LIMIT コマンドによって設定した機械語プログラムエリアはクリアされません。
CONT	CONT	プログラム実行を継続します。即ちプログラム中のSTOPステートメントあるいは、BREAK キーによって中断された箇所から実行を再開します。 プログラムの中断時に、BASICテキストのエディションを行うとCONTコマンドは無効になります。
MON	MON	システムのコントロールをBASICからモニタへ移します。(モニタからBASICへの復帰は、モニタコマンド" J "によって行うことができます。)
BOOT	BOOT	本機の IPL を起動して新たにシステムソフトウェアをローディングします。
PSEL		プリンタコマンドの各IOCSをMZ方式またはセントロ方式に切換えます。
	PSEL 1	MZ方式のプリンタに対してコントロールコードをそのまま出力します。
KLIST	KLIST	デファイナブル・ファンクション・キーの定義状態を調べるため、各機能をCRTディスプレイ上にリス　します。

4.2 ファイルコントロール文

INIT	INIT " Y "	クイックディスク上の全ファイルを消去します。
CHAIN	CHAIN " TEXT B "	ディスク上にある BASIC テキスト " TEXT B " にプログラム実行をチェインします。即ち、" TEXT B " を BASIC テキストエリアに読み出し、その先頭からプログラム実行を続けます。 このとき、テキストエリア中にあったもとのプログラムはNEW されますが、変数の値や利用者関数の内容は、CHAIN されるテキストに受け渡されます。 CHAIN 文の働きは、GOTO " file name " として理解することができます。

4.3 データファイルコントロール文

WOPEN#	WOPEN #3, " WSEQ DATA1 "	データファイルを作成するために書き込み用ファイルをオープンします。即ち、作成する BSD のファイル名を " SEQ DATA1 " と定義し、ディスク上に、ロジカルナンバ 3 番としてファイルをオープンします。USR 関数に対する WOPEN #ステートメントは、P.41に示されています。
PRINT#	PRINT#3, A, A$	WOPEN #ステートメントによってロジカルナンバ 3 番にオープンされているファイルに、変数 A、ストリング変数 A$ の内容を順に書き込みます。 ファイルは、CLOSE #ステートメントによってファイルクローズが実行されてはじめて 1 つのファイルとして正式に登録されます。
CLOSE#	CLOSE#3 (corresponding to WOPEN#)	WOPEN #ステートメントでロジカルナンバ 3 番にオープンされたファイルをクローズします。 ファイルクローズによって、WOPEN #ステートメントによって定義されたファイル名をもつ 1 つのファイルが指定ディスク上に作成され、ロジカルナンバ（この場合 3 番）を未定義のものに戻します。
KILL#	KILL#3	WOPEN #ステートメントでロジカルナンバ 3 番にオープンしたファイルをキルします。即ち、ファイルの作成をキャンセルし、ロジカルナンバ（この場合 3 番）を未定義のものに戻します。
	KILL	WOPEN、ROPEN されている全てのファイルをキルします。

ROPEN#	ROPEN #4,"SEQ DATA 1"	データファイル中のデータを読み出すためにファイルをオープンします。即ち、ディスク上のファイル"SEQ DATA 1"を、ロジカルナンバ4番としてファイルオープンします。
INPUT#	INPUT#4,A(1),B$	ROPEN #ステートメントによってロジカルナンバ4番にファイルオープンされているファイルから、順次データを読み出し、配列要素A(1)に数値データを、ストリング変数B$にストリングを代入します。 読み出すデータは、ファイルの先頭から順次シーケンシャルにアクセスされます。
CLOSE#	CLOSE#4 (corresponding to ROPEN#)	ROPEN #ステートメトで、ロジカルナンバ4番にファイルオープンされたファイルをクローズします。 ロジカルナンバ4番は未定義のものに戻します。
	CLOSE	WOPEN, ROPENされている全てのファイルをクローズします。
IF EOF(#)	IF EOF(#5) THEN 700	ファイルに対してINPUT #ステートメントを実行した時にファイルエンドが発生したら、行番号700の処理ルーチンへジャンプせよ、という分岐文です。

4.4 エラー処理文

ON ERROR GOTO	ON ERROR GOTO 1000	プログラム実行中にエラーが発生したら、行番号1000にジャンプせよという宣言文です。
IF ERN	IF ERN=44 THEN 1050	エラー番号が44であれば行番号1050へジャンプせよという命令です。
IF ERL	IF ERL=350 THEN 1090	エラー発生行番号が350であれば行番号1090へジャンプせよという命令です。
	IF (ERN=53)*(ERL=700) THEN END	エラー番号が53で、かつエラー発生行番号が700であるならば、プログラムを終了せよという命令です。 本BASICでは、プログラム中でエラーが発生したら、変数ERN、ERLにそれぞれ、エラー番号、エラー発生行番号がセットされます。
RESUME		エラー処理後、メインプログラムへ復帰する命令です。復帰のし方によって次のようなそれぞれの使い方ができます。
	650 RESUME	エラーが発生した命令へ再びコントロールを移します。
	700 RESUME NEXT	エラーが発生した命令の次の命令へコントロールを移します。
	750 RESUME 400	行番号400へコントロールを移します。
	800 RESUME 0	プログラムの先頭へコントロールを移します。

4.5 カセットテープ・データファイル入出力文

WOPEN/T	10 WOPEN/T " DATA-1 "	カセットテープ・データファイル " DATA-1 " を書き込みオープンします。
PRINT/T	20 PRINT/T A , A$	数値変数A、ストリング変数A$ の内容をWOPEN/Tによってオープンされているカセットテープデータファイルに書き込みます。
CLOSE/T	30 CLOSE/T	WOPEN/T によってオープンしたファイルをクローズします。
ROPEN/T	110 ROPEN/T " DATA-2 "	カセットテープ・データファイル " DATA-2 " を読み出しオープンします。
INPUT/T	120 INPUT/T B , B$	ROPEN/Tによってオープンしたカセットテープ・データファイル中のデータを順に読み出し、数値変数B、ストリング変数B$ に代入します。
CLOSE/T	130 CLOSE/T	ROPEN/T によってオープンしたファイルをクローズします。

4.6 代入文

LET	〈LET〉 A＝X＋3	数値変数Xと数値データ3の加算結果を数値変数Aに代入します。LET は省略できます。

4.7 入出力文

PRINT	10 PRINT A	CRTディスプレイ上に数値変数Aの内容を表示します。
	?A$	CRTディスプレイ上にストリング変数A$の内容を表示します。
	100 PRINT A ; A$, B ; B$	数値変数とストリング変数を混合して使用できます。また区切りでセミコロンが使われると、スペースなしで続けて表示され、コンマが使われると次の表示位置（10文字ごとの区切り）から表示されます。
	110 PRINT " COST＝ " ; CS	クォーテーションマーク「"」で囲まれたストリングはその内容がそのまま表示されます。
	120 PRINT	PRINT だけの場合は、行替えになります。
INPUT	10 INPUT A	キーボードから変数Aに対する数値を入力します。
	20 INPUT A$	キーボードからストリング変数A$ に対するストリングを入力します。
	30 INPUT " VALUE? " ; D	キーボードから入力する前に、ストリングの VALUE? を表示させます。ストリングと変数の区切りはセミコロン「;」を使います。
	40 INPUT X , X$, Y , Y$	数値変数やストリング変数はコンマ「,」で区切れば混合して使用できますが、入力する際には変数の型に合わせる必要があります。

GET	10 GET N	キーボードから数値変数Nに対して、1文字の数値を入力します。そのときキーが押されていないと、0が入力されます。
	20 GET K$	キーボードからストリング変数K$に対して、1個のストリングを入力します。そのときキーが押されていないと、A$は空になります。
READ～DATA		DATA文に置かれた定数またはストリングをREAD文の中に示された変数に代入する命令です。READ文中の変数と、それに対応するDATA文中の各データは、数値変数なら数値データ、ストリング変数ならストリングデータと、変数とデータの形が一致しなくてはなりません。
	10 READ A , B , C 1010 DATA 25 , －0.5 , 500	左のREAD～DATA文の実行によって数値変数A、B、Cのそれぞれに数値データ25、－0.5、500が代入されます。
	10 READ H$, H , S$, S 30 DATA HEART , 3 35 DATA SPADE , 11	READ文の最初の変数、すなわちストリング変数H$にDATA文の最初のデータ、すなわちストリングデータ"HEART"が代入されます。次に2番目の変数Hには、数値データ3が代入され、次々に読み込まれて行きます。
RESTORE		READ…DATA命令では、READ命令につれてDATA文中から読み込むデータが移って行きますが、RESTORE文を使うことによって読み込むデータをテキスト中のDATA文の最初に戻すことができます。
	10 READ A , B , C 20 RESTORE 30 READ D , E 100 DATA 3 , 6 , 9 , 12 , 15	左の例では、行番号10のREAD文によって、変数A、B、Cにそれぞれ、3、6、9の値が代入されますが、次にRESTORE文が置かれているので、行番号30のREADによって変数D、Eに代入される値は、12、15とはならずに、それぞれ3、6が代入されることになります。
	700 RESTORE 200	READ～DATA文におけるデータ読み出しポインタを、行番号200のDATA文の先頭へ移します。

4.8 ループ文

FOR～TO **NEXT**	10 FOR A＝1 TO 10 20 PRINT A 30 NEXT A	行番号10は変数Aを1から10まで変化させよという命令で、最初Aの値は1となります。行番号20でAの値がCRTディスプレイに表示されるので数値1が表示されます。次に行番号30でAの値は2になってこのループを繰り返します。こうしてAの値が10になるまでこのループが繰り返されます。(ループを終了した時点でAには11の値が入っています。)
	10 FOR B＝2 TO 8 STEP 3 20 PRINT B^2 30 NEXT	行番号10は変数Bを2から8まで、3ずつ大きくして変化させよという命令です。STEPの値を負にして変数の値を小さくして行くこともできます。
	10 FOR A＝1 TO 3 20 FOR B＝10 TO 30 30 PRINT A , B 40 NEXT B 50 NEXT A （Bのループ／Aのループ）	変数AとBについてFOR……NEXTループを重ねた例です。BループはAループの内部に置かれているところに注目して下さい。ループは2重、3重……とネスティングすることができますが、内側のループは必ず外側のループ内に閉じていなくてはなりません。 FOR……NEXTのネスティングは15レベルを越えてはなりません。
	60 NEXT B , A 70 NEXT A , B	前の2重ループで、行番号40と50を、左の行番号60のように1つのNEXT文にまとめることができます。しかし行番号70に示したようなオペランドではエラーになります。

4.9 分岐文

GOTO	100 GOTO 200	行番号200へジャンプ（＝プログラム実行を移すこと）します。
GOSUB **～RETURN**	100 GOSUB 700 …… 800 RETURN	行番号700のサブルーチンへ分岐（＝サブルーチンをコールすること）します。RETURN文でサブルーチンの実行を終了し、メインプログラムでGOSUB命令をした次のステートメントへ戻ります。

IF～THEN	10 IF A＞20 THEN 200	変数Aの値が20より大なら、行番号200へジャンプします。Aが20以下なら次の行を実行します。
	50 IF B＜3 THEN B＝B＋3	変数Bの値が3より小なら変数BにB＋3の値を代入します。Bが3以上なら次の行を実行します。
IF～GOTO	100 IF A＞＝B THEN 10	変数Aの値が変数Bの値以上なら行番号10へジャンプします。AがBより小なら次の行を実行します。
IF～GOSUB	30 IF A＝B＊2 GOSUB 90	変数Aの値が変数Bの値の2倍に等しいなら行番号90のサブルーチンへ分岐します。等しくないなら次の行を実行します。 (条件文のあとにマルチステートメントが来る場合、ON 文は条件が成り立たないとき実行されますが、IF 文は条件が成り立たないとき次の行番号へ移りマルチステートメントは無視されます。)
ON～GOTO	50 ON A GOTO 70，80，90	変数Aの値が1なら行番号70へ、2なら行番号80へ、3なら行番号90へジャンプします。Aが0または4以上なら次の文を実行します。ONにはINTの機能が含まれておりAが2.7ならば2の場合と同様に行番号80へジャンプします。
ON～GOSUB	90 ON A GOSUB 700，800	変数Aの値が1なら行番号700、2なら行番号800のサブルーチンへ分岐します。Aが0または3以上なら次の文を実行します。

4.10 定義文

DIM		配列を使う場合には、このDIM (dimensionの略) 文で、配列要素の最大を宣言しておかなければなりません。配列要素は0から、最大255まで使えます。
	10 DIM A(20)	1次元数値配列A( )について、配列要素をA(0)からA(20)まで21個用意します。
	20 DIM B(79，79)	2次元数値配列B( )について、配列要素をB(0，0)からB(79，79)まで6400個用意します。
	30 DIM C1$(10)	1次元ストリング配列C1$( )について、配列要素をC1$(0)からC1$(10)まで11個用意します。
	40 DIM K$(7，5)	2次元ストリング配列K$( )について、配列要素を、K$(0，0)からK$(7，5)まで48個用意します。
DEF FN	100 DEF FNA(X)＝X^2－X 110 DEF FNB(X)＝LOG(X)＋1 120 DEF FNZ(Y)＝LN(Y)	DEF FNで関数の定義をします。行番号100はX^2-XをFNA(X)に、文110は$\log_{10}X+1$をFNB(X)に、文120は$\log_e Y$をFNZ(Y)に定義します。関数は1変数に限ります。
DEF KEY	15 DEF KEY(1)＝LIST↵ 25 DEF KEY(2)＝LOAD!RUN↵	DEF KEY文は、デファイナブル・ファンクションキーの機能定義を行います。行番号15のDEF KEY文は、ファンクションキー1番に、LIST CR の機能を定義し、行番号25では、LOAD：RUN CR の機能が定義されます。

4.11 注釈文とコントロール文

REM	200 REM JOB-1	REMは注釈文であり、プログラム実行の際無視されます。
STOP	850 STOP	プログラムの実行をやめて、命令待ちとなります。ここでCONT命令を与えると、プログラムを続行します。
END	1999 END	プログラムの最後を表わします。プログラムの実行をやめますが、CONT命令を与えると、さらに先のプログラムを実行します。
CLR	300 CLR	数値をとる変数および配列要素の内容をすべて0、ストリングをとる変数および配列要素の内容をすべて空（null）とします。
CURSOR	50 CURSOR 25 , 15 60 PRINT " ABC "	CRTディスプレイの任意の箇所にカーソルを移動させるのがCURSORです。X軸方向には、40キャラクタモードのときは0～39、80キャラクタモードのときは0～79、Y軸方向には上端から0～24の数値または変数で位置を指定します。左の例では、ストリング" ＡＢＣ "は、画面の左端から26番目、上端から16番目のカーソル位置から表示されます。
CSRH		現在のカーソル位置のX座標（水平位置）を示すシステム数値変数です。
CSRV		現在のカーソル位置のY座標（垂直位置）を示すシステム数値変数です。
CONSOLE	10 CONSOLE S10 , 20	CRTディスプレイ上のスクロールエリアを第10行から20行までとします。
	20 CONSOLE C80	CRTディスプレイ上のキャラクタ表示を80キャラクタ／行にします。
	30 CONSOLE C40	CRTディスプレイ上のキャラクタ表示を40キャラクタ／行にします。
	40 CONSOLE R	CRTディスプレイをリバース表示とします。
	50 CONSOLE N	CRTディスプレイをノーマル表示に戻します。
	60 CONSOLE GN	CRTディスプレイ上のグラフィック表示のリゾリューションモードを320×200ドット／画面にします。
	70 CONSOLE GH	CRTディスプレイ上のグラフィック表示のリゾリューションモードを640×200ドット／画面にします。
	100 CONSOLE C40, GH, P	40キャラクタモード、ハイリゾリョーションモードに設定し、白／黒CRT画面上へのグラフィック画面の表示を禁止します。
	110 CONSOLE C80, M	80キャラクタモードに設定し、白／黒CRT画面上へグラフィック画面を表示します。
TABSET		TAB キーのタビュレーション位置を設定します。
	TABSET 10, 20, 30	TAB キーを押すごとにカーソルは画面の左から11, 21, 31番目の位置に移動します。
	TABSET	それまでに設定されていたタビュレーション位置をすべてクリアします。
CHANGE	10 CHANGE	メインキーボード上の26個のアルファベットキーのシフトポジション大文字／小文字を逆にします。
REW	710 REW	カセットテープを巻き戻します。
FAST	720 FAST	カセットテープを早送りします。
SIZE	? SIZE	BASICテキストエリアの未使用バイトサイズを表示させます。
TI$	100 TI$=" 222030 "	内蔵時計を午後10時20分30秒に合わせます。時刻データはクォーテーションマーク内に6桁の数字で表現します。

4.12 ミュージックコントロール文

MUSIC
TEMPO

音楽の自動演奏をさせる命令です。文字通りTEMPO文で指定されたテンポにのって、MUSIC文にあるクォーテーション「"」内の旋律のストリング（音程と音長の指定された音符の集まりに相当します）を音にしてスピーカを鳴らします。

```
300 TEMPO 7
310 MUSIC "DE#FGA"
```

行番号300でテンポが7（最も速い速度です）に指定されます。行番号310で、7のテンポで中音域のレミファの#ソラが続けて演奏されます。TEMPO文が無かったらテンポはデフォルト値で演奏されます。

```
300 M1$="C3EG+C"
310 M2$="BGD-G"
320 M3$="C8R5"
330 MUSIC M1$,M2$,M3$
```

この例では、旋律を3つのストリング変数に代入してMUSIC命令を実行させるものです。五線符で示すと下の音符が演奏されます。なお、TEMPO文がありませんので、デフォルト値のテンポで演奏されます。



4.13 カラーコントロール文

CCOLOR

キャラクタおよびバックグランドのカラーコードと、プライオリティを指定します。

```
100 CCOLOR @6,1
```

プライオリティをキャラクタ（文字）優先にし、キャラクタを黄色にバックグランドを青色に指定します。

```
110 CCOLOR @5
```

プライオリティをキャラクタに指定し、キャラクタを水色に指定します。

```
120 CCOLOR 4
```

キャラクタを緑色に指定します。

```
130 CCOLOR @,3
```

プライオリティをキャラクタに指定し、バックグランドを紫に指定します。(注)

```
140 CCOLOR,2
```

バックグランドを赤色に指定します。(注)

(注) キャラクタのカラーコードを省略し、バックグランドのカラーコードを記述する場合、この例のように必ずコンマ（,）を記述しなければなりません。

COLOR

カラーグラフィック表示について、入力グラフィックエリアをカラーコードで指定します。また出力ページモード、プライオリティおよび色の重ね合わせモードを指定します。

```
200 COLOR @
```

プライオリティをグラフィックに指定します。

```
210 COLOR 5
```

グラフィックのカラーコードを水色に指定します。

```
220 COLOR,O1,W1
```

青色のグラフィックエリアのデータのみを表示し、色の重ね合わせのモードに指定します。

```
230 COLOR,,W0
```

強制的に色を変更するモードに指定します。

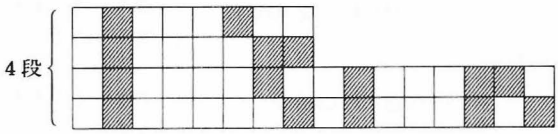
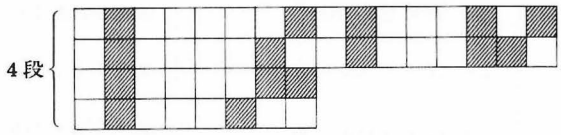
```
240 COLOR @1,O7,W1
```

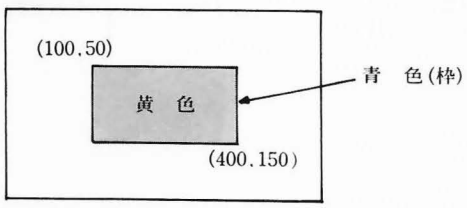
プライオリティをグラフィックに、グラフィックのカラーコードを青色、出力ページモードを全ページ出力とし、色の重ね合わせを行うモードに指定します。

```
250 COLOR 2,,W0
```

グラフィックのカラーコードを赤色とし、強制的に色を変更するモードに指定します。

GRAPH	10 GRAPH I1	グラフィックエリアへのデータ転送モードを、ページ1（グラフィックエリア1）に設定します。
	20 GRAPH O1	グラフィックエリア1を表示モードとします。
	30 GRAPH O23	グラフィックエリア2および3を表示モードとします。
	40 GRAPH O123	グラフィックエリア1、2および3を表示モードとします。
	50 GRAPH O0	グラフィックエリアの表示を行わないようにします。
	60 GRAPH C	GRAPH I 文によってデータ転送モードとなっているエリアをクリアします。
	70 GRAPH F	転送モードとなっているグラフィックエリアをフィル（fill）します。
	80 GRAPH I1，C，O1	3つの命令を行うものではじめに、データ転送をグラフィックエリア1とし、続いてエリア1をクリアして、表示モードもエリア1とします。
	100 GRAPH C	COLOR文で指定した入力グラフィックエリアをクリアします。
	110 GRAPH F	COLOR文で指定した入力グラフィックエリアをフィル（fill）します。
	120 GRAPH C3, F4	青と赤のグラフィックエリアをクリアし、緑のグラフィックエリアをフィルします。
	130 GRAPH CA	数値変数Aで示す色のグラフィックエリアをクリアします。
CONSOLE	100 CONSOLE C40, GH, P	40キャラクタモード、ハイリゾリューションモードに設定し、白／黒CRT画面上へのグラフィック画面の表示を禁止します。
	110 CONSOLE C80, M	80キャラクタモードに設定し、白／黒CRT画面上へグラフィック画面を表示します。
SET	300 SET 159, 199	画面の中央にドットをセットします。(320×200ドット/画面の場合)
	400 SET[1, W0]300, 100	青色をセットします。
	410 SET[A]300, 100	数値変数Aの内容にしたがった色をセットします。
	420 SET[, W1]300, 100	COLOR文で設定した色をすでに描かれている色に重ね合わせてセットします。
	430 SET X1, Y1	COLOR文で設定したカラーコントロールにしたがってセットします。
RESET	310 RESET 159, 99	画面の中央のドットをリセットします。(320×200ドット/画面の場合)
	600 RESET A, B	COLOR文で設定したカラーコントロールにしたがってリセットします。
	610 RESET[3, W1]300, 100	紫、すなわち青と赤のグラフィックエリア上のドットをリセットします。
	620 RESET[C]300, 100	数値変数Cの内容にしたがってリセットします。色の重ね合わせはCOLOR文で設定したモードにしたがいます。
	630 RESET[, W0]300, 100	COLOR文で設定したカラーコードの反転色をセットします。
LINE	400 LINE 110，50，210，50，210，150，110，150，110，50	画面の中央に1辺の長さ100の正方形を描きます。(320×200ドット/画面の場合)
	800 LINE 0, 0, 639, 199	COLOR文で設定したカラーコントロールにしたがってラインを描きます。
	810 LINE[6, W1]0, 100, 639 100	以前に描かれている色の上に重ね合わせて黄色のラインを描きます。

	820 LINE[A]X1, Y1, X2, Y2	数値変数Aの内容の色でラインを描きます。 (色の重ね合わせはCOLOR文で設定したモードにしたがいます)
	830 LINE[, W0]600,0,600,199	COLOR文で設定した色でラインを描きます。
BLINE	900 BLINE[5,W1]0,0,639,199	青と緑のグラフィックエリアを指定の線分に沿ってリセットします。
	910 BLINE[6;W0]50,0,5,150	黄〔6〕の反転色、すなわち青〔7－6＝1〕のラインを描きます。
	920 BLINE[2]20,10,20,100, 300,100,300,10,20,10	COLOR文で設定した色の重ね合わせモードに依存して、赤の色がリセットされた長方形か、あるいは赤の反転色である水色の長方形かのいずれかを描きます。
POSITION		グラフィックエリア上のポジションポインタを設定します。次のPATTERN文は、このポジションポインタの位置から実行されます。
	20 GRAPH I2 , C , O2 30 POSITION 0 , 50 40 PATTERN 8 , A$	行番号20で、グラフィックエリア2を転送モード、また表示モードとして、エリアをクリアし、行番号30、40で座標位置（0 , 50）からグラフィックパターンを表示します。グラフィックパターンは下から上方へ8段重ねで表示されます。
PATTERN		POSITION 文で指定されたポジションポインタ位置から、任意のドットパターンを8ビット単位で表示します。ドットパターンはストリングデータ又は、ストリング変数で与えます。オペランドの第一項は、8ビット単位のドットパターンを上下どちらの方向に何段組にするか決めます。
	10 C$＝" ABCDEF " 20 PATTERN 4 , C$	次のドットパターンが表示されます。 
	30 PATTERN －4 , C$	次のドットパターンを表示します。 
	100 PATTERN －8, A$	COLOR文で設定のカラーコントロールにしたがって、グラフィックパターンを描きます。
	110 PATTERN[2,W0]+10,B$	色の重ね合わせなしに強制的に赤色でグラフィックパターンを描きます。
	120 PATTERN[, W1]16,C$	COLOR文で設定の色に、すでに書かれている色を重ね合わせてグラフィックパターンを描きます。
	130 PATTERN[C]－5,D$	変数Cの内容にしたがった色で、グラフィックパターンを描きます。
	140 PATTERN[1,W0]7, E$, [1, W1]－3,F$, 10, G$	色の重ね合わせなしに緑色でグラフィックパターンを描き、次に、すでに書かれている色に重ね合わせて青色でグラフィックパターンを描き、最後に、COLOR文で設定したカラーコントロールに従ってグラフィックパターンを描きます。

PAINT		指定した座標点を囲む指定した色の境界線の範囲内を指定した色でぬりつぶします。
	300 PAINT 320, 100	COLOR文で指定したカラーコードが赤の場合、座標（320, 100）を囲む赤色の境界線の範囲内を赤色でぬりつぶします。
	310 PAINT[6]320, 100	座標（320,100）を囲む黄色の境界線の範囲内を黄色でぬりつぶします。
	320 PAINT XN, YN, 3	座標（XN, YN）を囲む紫の境界線の範囲内をCOLOR文で指定したカラーコードにしたがってぬりつぶします。
	330 PAINT[4]320,100,1,6,7	座標（320,100）の周りの青、緑、黄および白色の境界線によって囲まれた範囲内を緑色でぬりつぶします。
BOX		指定した2点を対角線の頂点とする四角形を描きます。ぬりつぶしを指定すると、その四角形内のエリアを指定した色でぬりつぶします。
	400 BOX 10, 10, 630, 190	COLOR文で設定のカラーコントロールにしたがい四角形の枠のみを描きます。
	410 BOX[1]10,10,630,190	青色で四角形の枠のみを描きます。
	420 BOX[,W1]20,20,620,180 F	COLOR文で設定のカラーコードにしたがって四角形の枠を描き、枠内を枠と同じ色でぬりつぶします。(色は重ね合わされる)
	430 BOX[1,W0]100,50,400, 150,F6	青色で四角形の枠を描き、その枠内を黄色でぬりつぶします。（下図参照） 
CIRCLE		指定した座標点を中心として、指定した色で楕円（円）または円孤（扇形）を描きます。
	500 CIRCLE 200, 100, 80	COLOR文のカラーコントロールにしたがい、中心（200,100）、半径80の円を描きます。
	510 CIRCLE[1, W0]200, 100, 80,, π/4, 3*π/4	π/4～3π/4ラジアンの円孤を青色で描きます。
	520 CIRCLE[2, W0]200, 100, 50, 0.35	赤色で楕円を描きます。
	530 CIRCLE[4]200, 100, 70,, π/2, π, O	π/2～π（ラジアン）の扇形を緑色で描きます。強制的に緑色で描くことになるか、すでにぬられている色の上に緑色を重ね合わせることになるかは、COLOR文で設定した色の重ね合わせモードに依存します。

POINT

```
100 ON  POINT (X , Y)
    GOTO  10 , 20 , 30 , 40 ,
    50 , 60 , 70
```

グラフィックエリア上の座標点（X，Y）がセットされているかリセットされているかで分岐する文です。

POINT関数の関数値　　ポイント情報

0 …… グラフィックエリア１，２および３ともリセットされている。
1 …… グラフィックエリア１のみセットされている。
2 …… グラフィックエリア２のみセットされている。
3 …… グラフィックエリア1,2のみセットされている。
4 …… グラフィックエリア３のみセットされている。
5 …… グラフィックエリア1,3のみセットされている。
6 …… グラフィックエリア2,3のみセットされている。
7 …… グラフィックエリア1,2,3すべてがセットされている。

（注：上記情報はグラフィックエリア１、２、３すべてが装備されている場合のものであり、いずれかのエリアが、装備されていない場合については、BASIC/MONITOR MANUALを参照ください。）

POSH

グラフィック表示エリア上のポジションポインタの現在のX座標を示すシステム変数。

POSHのとり得る値は
0≦POSH≦319（ノーマルリゾリューションモード）
0≦POSH≦639（ハイリゾリューションモード）
となります。

POSV

グラフィック表示エリア上のポジションポインタの現在のY座標を示すシステム変数。

4.14 漢字コントロール文

JTRANS

ASCIIコード表の文字を漢字パターンに変換して表示させる目的で、KPATTERN文とともに用います。ASCIIコード（1バイト）を、対応する文字のJIS漢字コード（2バイト）に変換します。

```
100 A=JTRANS ($26)
```
ASCIIコードが$26の文字"&"のJIS漢字コード$2175を数値変数Aに代入します。

```
110 B=55
120 C=JTRANS (B)
```
ASCIIコードが55の文字"7"のJIS漢字コード$2337を数値変数Cに代入します。

```
130 D$="KLM"
140 E=JTRANS (ASC (D$))
```
ストリング変数D$の最初の文字"K"のJIS漢字コード$234Bを変数Eに代入します。

```
150 F=JTRANS (232)
```
ASCIIコードが232の文字"■"はJIS漢字コード表にないので、Error 3 (illegal data) が発生します。

KCHR$

JIS漢字コードに対応する漢字キャラクタを与えます。

```
200 A$=KCHR$ ($3250)
```
ストリング変数A$にJIS漢字コードが$3250である漢字キャラクタ"火"を代入します。

```
210 B$=KCHR$ ($3861,
    $3865)
```
ストリング変数B$に漢字文字列"午後"を代入します。

KINPUT

JIS漢字コードに対応する1つの漢字をカナ漢字変換をし、その文字列を入力します。

```
10 KINPUT  X$
```
COLOR文で指定したカラーコントロールにしたがって、漢字をストリング変数X$に漢字キャラクタとして入力します。

```
20 KINPUT [5, W0]
   A$, B$
```
色を水色に指定して漢字をA$, B$に入力します。

```
30 KINPUT [A] X1$, X2$
```
変数Aで指定したカラーコードで漢字をX1$, X2$に入力します。

```
40 KINPUT [, W1] A$
```
COLOR文で指定した色とすでに描かれている色との重ね合わせで漢字を、A$に入力します。

(注) KINPUT命令を実行するとハイリゾリューションモード時の(0, 184)～(639,199)とノーマルリゾーションモード時の (0, 184) ～(319, 199) までのグラフィックエリアはクリアされます。

KPATTERN

COLOR文あるいはGRAPH I文で入力モードに指定しているグラフィックエリアに漢字パターンを描きます。

```
100 A$=KCHR$ (JTRANS
        (ASC ("B")))
      +KCHR$ (JTRANS
        (ASC ("A")))
      +KCHR$ (JTRANS
        (ASC ("S")))
      +KCHR$ (JTRANS
        (ASC ("I")))
      +KCHR$ (JTRANS
        (ASC ("C")))
```
B、A、S、I、Cのそれぞれの文字をアスキーコードに変換し、それをさらに漢字コードに変換してそのコードをA$に代入します。

```
110 B$=KCHR$ ($3840)
```
$3840 ："言"のJIS漢字コード

```
120 C$=KCHR$ ($386C)
```
$386C ："語"のJIS漢字コード

```
130 D$=KCHR$ ($467C)+
      KCHR$ ($4B5C)
```
$467C ："日"のJIS漢字コード
$4B5C ："本"のJIS漢字コード

	140 E$=D$+C$	
	150 CONSOLE GH, M	
	160 PRINT CHR$(6)	
	170 COLOR@ 6, O7, W0	
	180 KPATTERN[2, W1]D$	赤色で〝日本〟と表示します。
	190 KPATTERN E$	黄色で〝日本語〟と表示します。
	200 KPATTERN[1]A$, B$, C$	〝BASIC〟が青色で、それに引き続いて〝言語〟が黄色で、それぞれ表示します。
	210 KPATTERN[4]D$, [5] KCHR$($3324)	〝日本〟が緑色で、〝海〟が水色で、〝日本海〟と表示します。
KPRINT／P		オペランドの出力並びで指定した漢字データをセパレータの機能などにしたがって並びの順にプリンタに印字します。
	10 KPRINT／P A$	ストリング変数A$の内容を漢字データとしてプリンタに印字します。
	20 KPRINT／PKCHR$($2356)	漢字コード$2356の文字をプリンタに印字します。

4.15 機械語プログラムコントロール文

LIMIT	100 LIMIT A	BASIC プログラムで使用するエリアを、変数Aの値の番地に制限します。
	300 LIMIT MAX	BASIC プログラムで使用するエリアを、メモリの最大に戻します。
	200 LIMIT $BFFF 210 LOAD FD2 , " S-R1 "	機械語プログラムファイル（OBJ）" S-R1 "が、ローディングアドレスC000以上のものであれば、左のプログラムによって、機械語リンクエリア内に、" S-R1 "が、ドライブ2番中のディスケットより読み出されます。
POKE	5 POKE 49450, 175 10 POKE 49450, 175, 143, 101 20 POKE AD, DA, DB, DC, DD	10進番地49450にデータ175（10進表現）をセットします。 10進番地49450にデータ175（10進表現）、49451にデータ143、49452にデータ101をセットします。 変数 AD で指定する番地から順次、変数 DA、DB、DC、DD で示される値（0～255の範囲）をセットします。
PEEK	150 A＝PEEK(49450)	10進番地49450にはいっているデータを10進数に直して変数Aに代入します。
	160 B＝PEEK(C)	変数Cで指定される10進番地にはいっているデータを10進数に直して変数Bに代入します。
USR	500 USR(49152)	10進番地49152にプログラムのコントロールを移します。このコントロールの移動は、機械語の CALL コマンドと同じ機能を持っています。従って、機械語プログラムに、RETコマンド、(10進コードで201）があると、BASIC プログラムへリターンします。
	550 USR(AD)	変数ADで指定される10進番地を CALL します。
	570 USR($C000)	16進番地C000をCALLします。
	600 WOPEN#8 , USR ($C000) 610 PRINT#8 , A$ 620 CLOSE#8	USR（$C000）を、ロジカルナンバ8番に定義して書き込みオープンします。行番号610で、ストリング変数A$の内容がセットされているバッファの先頭番地をDEレジスタにセットし、データの長さ（最大255バイト）をBCレジスタにセットして、USR（$C000）を実行します。

	700 ROPEN#9,USR ($C100) 710 INPUT#9,B$ 720 CLOSE#9	USR（$C100）を、ロジカルナンバ9番に定義して読み出しオープンします。行番号710でUSR（$C100）が実行されますが、コールされた機械語ルーチンでコールされた時のDEレジスタの示すアドレスを先頭として、ストリングデータをセットし、その長さをBCレジスタにセットしてリターンすると、B$ にそのストリングデータが代入されます。

4.16 プリンタ・コントロール文

PRINT/P		PRINTと同様の機能を、オプションのプリンタに対して実行します。プリンタが接続されていない時はエラーになります。
	10 PRINT/P A,A$	数値変数Aの内容、続けてストリング変数A$ の内容をプリンタにプリントします。
	20 PRINT/P CHR$(5)	プリンタのフォーム・フィードを行います。(CHR$(5)はプリンタコントロールコードです。)
IMAGE/P	30 IMAGE/P CHR$(255), "UU"	プリンタに任意のイメージドットパターンをプリントします。
COPY/P	10 COPY/P 1	プリンタにCRTディスプレイ上のキャラクタ表示のコピーをとります。
	20 COPY/P 2	プリンタにCRTディスプレイ上のグラフィックエリア1のコピーをとります。
	30 COPY/P 3	プリンタにCRTディスプレイ上のグラフィックエリア2のコピーをとります。
	40 COPY/P 4	プリンタにCRTディスプレイ上のグラフィックエリア1と2のコピーをとります。
	50 COPY/P 5	プリンタにCRTディスプレイ上のグラフィックエリア3のコピーをとります。
	60 COPY/P 6	プリンタにCRTディスプレイ上のグラフィックエリア1と3のコピーをとります。
	70 COPY/P 7	プリンタにCRTディスプレイ上のグラフィックエリア2と3のコピーをとります。
	80 COPY/P 8	プリンタにCRTディスプレイ上のグラフィックエリア1、2および3のコピーをとります。
PAGE/P	100 PAGE/P 20	プリンタの1ページを20行に設定します。

4.17 I/O入出力文

INP		I/Oポート番号を指定して、そのポート上にあるデータの読み出しを行います。
	10 INP @12,A 20 PRINT A	行番号10で、I/Oポート番号12（10進）にあるデータを、変数Aに読み出します。
OUT		外部デバイスにデータを送るため、I/Oポート番号を指定してデータを出力します。
	30 B=A^2+0.3 40 OUT @13,B	行番号40で、変数Bの値をI/Oポート番号13の出力ポートへ出力します。

4.18 数値関数

ABS	100 A＝ABS(X)	変数Xの値の絶対値｜X｜を変数Aに代入します。カッコ内は定数、数式でもかまいません。 (例) ABS (－3)＝3 ABS (12)＝12
INT	100 A＝INT(X)	変数Xの値について、Xを越えない最大の整数を求めて変数Aに代入します。カッコ内は定数、数式でもかまいません。 (例) INT (3．87)＝3 INT (0．6)＝0 INT (－3．87)＝－4
SGN	100 A＝SGN(X)	変数Xの値についてX＜0のとき－1を、X＝0のとき0を、X＞0のとき、1を変数Aに代入します。カッコ内は定数、数式でもかまいません。 (例) SGN (0．4)＝1 SGN (0)＝0 SGN (－400)＝－1
SQR	100 A＝SQR(X)	変数Xの値について、$\sqrt{X}$の値を求めて変数Aに代入します。カッコ内は定数、数式でもかまいませんが、正または0の値でなければなりません。
SIN	100 A＝SIN(X)	変数Xの値（ラジアン）について、sinXの値を求め変数Aに代入します。カッコ内は定数、数式でもかまいません。ラジアンと度の関係は、 1度＝$\frac{\pi}{180}$ラジアン
	110 A＝SIN(30＊π/180)	sin30°の値を変数Aに代入します。
COS	200 A＝COS(X)	変数Xの値（ラジアン）について、cosXの値を求め変数Aに代入します。カッコ内は定数、数式でもかまいません。度で計算するにはSIN関数と同様の方法を使います。行番号210はcos200°の値を変数Aに代入する命令文です。
	210 A＝COS(200＊π/180)	cos200°の値を変数Aに代入します。
TAN	300 A＝TAN(X)	変数Xの値（ラジアン）について、tanXの値を求め変数Aに代入します。カッコ内は定数、数式でもかまいません。度で計算するにはSIN関数と同様の方法を使います。行番号310はtanY°の値を変数Aに代入する命令文です。
	310 A＝TAN(Y＊π/180)	tanY°の値を変数Aに代入します。
ATN	400 X＝ATN(A)	変数Xの値について、$\tan^{-1}X$の値（ラジアン）を求め変数Aに代入します。カッコ内は定数、数式でもかまいません。計算結果は$-\frac{\pi}{2}$と$\frac{\pi}{2}$の間の値となります。行番号410は$\tan^{-1}X$の値を度にして変数Aに代入する命令文です。
	410 Y＝180/π＊ATN(A)	$\tan^{-1}X$の値を度にして変数Aに代入します。
EXP	100 A＝EXP(X)	変数Xの値について、e^Xの値を求めて変数Aに代入します。カッコ内は定数、数式でもかまいません。
LOG	100 A＝LOG(X)	変数Xの値について、常用対数$\log_{10}X$の値を求めて変数Aに代入します。カッコ内は定数、数式でもかまいませんが、正の値でなければなりません。
LN	100 A＝LN(X)	変数Xの値について、自然対数$\log_e X$の値を求めて変数Aに代入します。カッコ内は定数、数式でもかまいませんが、正の値でなければなりません。
	110 A＝LOG(X)/LOG(Y) 120 A＝LN(X)/LN(Y)	対数の底がYのときの対数$\log_Y X$を求めるには行番号110または行番号120によって求められます。

RND		0.00000001から0.99999999までの値をとる擬似乱数を発生する関数です。カッコ内に0または負の整数を与える場合と、正の整数を与える場合とで、2通りの処理が行われます。
	100 A＝RND(1) 110 B＝RND(10)	行番号100または110のようにカッコ内に正の整数を与えるとRND関数を使うたびに、順次0.00000001から0.99999999までの間の値をとる乱数値を発生します。(カッコ内に与える正の整数の値には無関係です。)
	200 A＝RND(0) 210 B＝RND(－3)	行番号200または210のようにカッコ内に0または負の整数を与えると乱数発生のイニシャライズが行なわれて、いつもある特定の数値を発生してAにもBにも同じ値が代入されます。

4.19 ストリングコントロール関数

LEFT $	10 A$＝LEFT$(X$，N)	ストリング変数X$の最初からN文字目までを、ストリング変数A$に代入します。Nは定数でも、変数、数式でもかまいません。
MID $	20 B$＝MID$(X$，M，N)	ストリング変数X$の第M文字目からN文字を、ストリング変数B$に代入します。
RIGHT $	30 C$＝RIGHT $(X$，N)	ストリング変数X$の右からN文字を、ストリング変数C$に代入します。
SPACE $	40 D$＝SPACE $(N)	N個のスペースをストリング変数D$に代入します。
STRING $	50 E$＝STRING $("＊", 10)	10個の連続したアステリスクマークを、ストリング変数E$に代入します。
CHR $	60 F$＝CHR $(A)	ASC関数の逆で、変数Aの値に等しいASCIIコードの文字（キャラクタ）をストリング変数F$に代入します。Aは定数、変数、数式いずれでもかまいません。
ASC	70 A＝ASC(X$)	ストリング変数X$の最初の文字のASCIIコード（10進数）の値を、変数Aに代入します。
STR$	80 N$＝STR$(I)	VAL関数の逆で、変数Iの数値をそのままストリングとして、ストリング変数N$に代入します。
VAL	90 I＝VAL(N$)	ストリング変数N$の数字ストリングを、そのまま数値として変数Iに代入します。
CHARACTER$	85 CR$＝CHARACTER$ (X，Y)	キャラクタ表示位置（X，Y）に現在表示されているキャラクタをストリング変数CR$に代入します。
LEN	100 LX＝LEN(X$)	ストリング変数X$の文字の長さ（文字数）を、変数LXに代入します。
	110 LS＝LEN(X$＋Y$)	ストリング変数X$とY$の文字の長さの和を、変数LSに代入します。

4.20 タブ関数

TAB	10 PRINT TAB(X)；A	画面の左端から数えてX＋1字目に変数Aの値を表示します。

4.21 算術演算子

左端の白ぬきの数字は計算の優先順位、更に優先されるのはカッコ( )内の計算です。

❶^	10 A＝X^Y(べき乗)	変数AにX^{Y}の計算結果を代入します。 (但しX^YでXが負数のとき、Yが整数でなければエラーとなります。)
❷−	10 A＝−B(負号)	0−Bは減算ですが、−Bの「−」は負号であることに注意して下さい。
❸＊	10 A＝X＊Y(乗算)	変数AにXとYの数値の乗算結果を代入します。
❸/	10 A＝X/Y(除算)	変数AにXとYの数値の除算結果を代入します。
❶+	10 A＝X+Y(加算)	変数AにXとYの数値の加算結果を代入します。
❶−	10 A＝X−Y(減算)	変数AにXとYの数値の減算結果を代入します。

4.22 比較・論理演算子

＝	10 IF A＝X THEN…	変数AとXの数値が等しいならば、THEN 以降の命令を実行します。
	20 IF A$＝"XYZ" THEN…	ストリング変数A$ の内容がストリング XYZ であれば、THEN 以降の命令を実行します。
＞	10 IF A＞X THEN…	変数AがXより大きいならば、THEN以降の命令を実行します。
＜	10 IF A＜X THEN…	変数AがXより小さいならば、THEN以降の命令を実行します。
＜＞or＞＜	10 IF A＜＞X THEN…	変数AとXの数値が等しくないならば、THEN 以降の命令を実行します。
＞＝or＝＞	10 IF A＞＝X THEN…	変数AがXより大きいか等しいならば、THEN 以降の命令を実行します。
＜＝or＝＜	10 IF A＜＝X THEN…	変数AがXより小さいか等しいならば、THEN 以降の命令を実行します。
＊	40 IF(A＞X)＊(B＞Y) THEN…	変数AがXより大きく、かつ変数BがYより大きいならば、THEN 以降の命令を実行します。
＋	50 IF(A＞X)＋(B＞Y) THEN…	変数AがXより大きいか、または変数BがYより大きいならば、THEN 以降の命令を実行します。

4.23 その他のシンボル

?	200 ? "A+B＝" ; A+B 210 PRINT "A+B＝" ; A+B	PRINTの代わりに用いることができます。したがって行番号200と210は同等です。
:	220 A＝X : B＝X^2 : ?A , B	命令文の区切りを表わす記号で、多重命令に使用します。行番号 220 の多重命令には、3つの命令文が置かれています。
;	230 PRINT "AB" ; "CD" ; "EF"	PRINTを続けて実行します。行番号230では画面上に、「ABCDEF」とスペースを空けずに続けて表示されます。
	240 INPUT "X＝" ; X$	画面に「X＝」と表示し、ストリング変数X$のデータキー入力を待ちます。
,	250 PRINT "AB" , "CD" , "E"	タブレーションをつけてPRINTを実行します。行番号250の文では、画面上にまずABと表示し、次にAから10文字右の場所よりCDと表示し、次にCから10文字右の場所にEと表示されます。

	300 DIM A(20), B$(3, 6)	変数の区切りに用いた例です。
" "	320 A$="SHARP BASIC" 330 B$="MZ-2000"	" "内がストリングであることを示します。
$	340 C$="ABC"+CHR$(3)	ストリング変数であることを示します。
	500 LIMIT $BFFF	16進数であることを示します。
π	550 S=SIN(X*π/180)	円周率の近似値3.1415927をπで表わします。

付録

Chapter 5

5.1 QDCOPY ユーティリティ

このソフトは、クイックディスクの内容をそっくりそのまま他のクイックディスクにコピーするものです。使用例として同梱されている BASIC を、このソフトを使ってコピーしてみましょう。

ファイル単位のコピーを行う場合は TRANS ユーティリティを使用してください。これは何らかの事故により、クイックディスクが使用不能となってもかわりのクイックディスクに同じものをコピーしておけば助かるからです。ふだん使用するときはコピーした方のディスクを使用し、同梱されていたディスクは保管しておくようにしましょう。

このソフトを動かすには次の命令を実行してください。

```
RUN"QDCOPY" CR
```

上記命令を実行することにより、次のような初期画面が表れます。

```
QD  Format/Copy  Utility  VX.XX

[ Function ]

F...Format disk
C...Copy disk
!...Boot

[ Command area ]
*
```

①まずはじめに、コピーディスクをフォーマット設定状態にしなければなりません。F キーを押すと「Command area」には次の表示が行われます。

```
[ Command area ]
*F
Set format disk, ↵ any key
```

フォーマット設定を行うディスクをディスクドライブにセットして、キー(いずれかのキー)を押してください。すると、コピーディスクのフォーマット設定が行われ、終了すると" Ok ! "が表示され、画面表示は次のようになります。

```
──────────[ Command area ]──────────
＊F
  Set format disk, ↵ any key
  Ok !
＊■
```

②コピーする元のディスクの内容をコンピュータ本体に読み込ませなければなりません。[C]キーを押すと「Command area」には次の表示が行われます。(前記述内容の表示は省略します。)

```
──────────[ Command area ]──────────
＊C
  Insert source disk, ↵ any key
```

コンピュータ本体に読み込ませます。ディスクをディスクドライブにセットしてキー(いずれかのキー)を押してください。するとディスクの内容がコンピュータ本体に読み込まれます。読み込みが終了すると画面は次のようになります。

```
──────────[ Command area ]──────────
＊C
  Insert source disk, ↵ any key
  Insert destination, ↵ any key
```

今度はコピーディスクをディスクドライブへセットしてキー(いずれかのキー)を押してください。するとコンピュータ本体に記憶されている内容がクイックディスクに書き込まれます。書き込みが終了すると〃Ok ! 〃と表示されます。(ただしコピーするファイル容量が大きかったり、ファイル数が多かったりすると一度ではコピーできない場合がありますので、②からの手順を〃Ok ! 〃の表示が出るまでくり返してください。)
コピーが終わると画面は次のようになります。

```
──────────[ Command area ]──────────
＊C
  Insert source disk, ↵ any key
  Insert destination, ↵ any key
  Ok !
＊■
```

クイックディスクのコピーが終了し次のコマンドまち状態になったことを示しています。次に[!]キー、([SHIFT]+[1])を入力すると、ROMモニタの初期画面に戻るか、先頭にあるOBJファイルを読み込みます。

5.2 DELETEユーティリティ

・このソフトはクイックディスクを有効に使用するために不必要なファイルを消去するものです。クイックディスク上に登録されているファイルが不必要になった場合、必要なファイルのみをコンピュータ本体のメモリ上に読み込んで、一時退避させておき、次にクイックディスクをイニシャライズします。その後、メモリ上に読み込んだ内容をクイックディスクに書き込むことにより、不必要なファイルを消去するものです。

・このソフトを動かすには次の命令を実行してください。

```
RUN"DELETE"CR
```

上記命令を実行することにより次のような初期画面が表れます。

```
[ QD  File  Delete  Utility  VX.XX ]

Set  Source  Disk


F1:Run  F2:        F3:        F4:Boot
```

［説明］

ここで消去したいファイルがあるクイックディスクをセットしてF1キーを押してください。F1キーを押すと、セットされたクイックディスクのディレクトリが表示され、画面は次のようになります。（次に１つの例を示します。）

```
[ QD  File  Delete  Utility  VX.XX ]

No. Atr File  Name          Size    Delete
 1  OBJ  ASEM          44651   No
 2  BTX  TEST            299   No
 3  OBJ  DATA           7840   No
 4  BTX  SAMPLE        10342   No
 5  OBJ  DATA2          1064   No


All size 64196
Max size 57088
F1:RUN  F2:        F3:Again  F4:Boot
```

ディレクトリが表示されたらカーソル(▩)が表示されますので、消去したいファイルのところに↑、↓キーを使ってカーソルを移動させ" Y "キーを押してください。" Y "キーを押すと今まで" No "と表示されていたのが" Yes "にかわります。このようにして消去したいファイルを" Yes "に変更してください。その際、画面右下にある「All Size」のバイト数表示が減少します。また、もしまちがえたりして消去したくないファイルを" Yes "にしてしまったら、

ファイルのところにカーソルを移動させ今度は" N "を入力してください。すると" Yes "が" No "に変更されます。次に前の画面で出てきた" DATA "というファイルと" DATA 2 "というファイルを消去したいとすると、[↑]、[↓]キーを使ってカーソルを消去したいファイルのところまで移動させて、" Y "キーを押して" No "を" Yes "に変更すると画面は次のようになります。

```
[ QD  File  Delete  Utility  VX.XX ]

No. Atr File  Name          Size    Delete
1   OBJ  ASEM              44651   No
2   BTX  TEST                299   No
3   OBJ  DATA                7840  Yes
4   BTX  SAMPLE             10342  No
5   OBJ  DATA2               1064  Yes



All size 55292
Max size 57088
F1:Run  F2:        F3:Again F4:Boot
```

" No "を" Yes "にしただけではファイルを消去したことにはなりません。次に[F1]キーを押すと" Yes "にしたファイル以外（" No "と表示しているファイル）を本体メモリ上に読み込み画面右下に次のような表示がでます。

```
File is loading !
```

（この時、All Size＞Max Sizeの条件が成立すると、Memory Capacity errorとなります）。読み込みが終了すると画面右下に次のような表示がでます。

```
Loading ok !
  Set destination disk
```

ここで、書き込むクイックディスクをセットしてください。ただし書き込むクイックディスクとして同じディスクを使用する場合は、入れかえる必要はありません。[F1]キーを押すと、次の表示がでてコンピュータ本体のメモリ上に読み込んだファイルの内容がクイックディスクに書き込まれます。

重要なクイックディスク中の一部のファイルをDELETE（消去）されるときには、Destination diskはSource diskとは別のクイックディスクで行うようにお願い致します。また、Destination diskにSAVE中にエラーが発生した場合に[F3]，[F4]キーを入力すると、コンピュータ本体のメモリ中に記憶されていた内容が消去されますので、別のクイックディスク（フォーマット設定されたもの）を挿入してEnd of jobになるまでやり直してください。

```
File is Saving !
```

書き込みが終了すると、画面右下に次の表示がされます。

```
End of job
```

ここで画面左下に表示されるメッセージについて説明します。

F１：Run ……指定した作業を実行させます。

F２：Cont ……Error（エラー）などで実行が中断された場合、エラーが起こる直前の状態に戻ります。

F３：Again……初期画面（初期状態）に戻ります。

F４：Boot ……ROMモニタの初期画面に戻るか、先頭にあるOBJファイルを読み込みます。

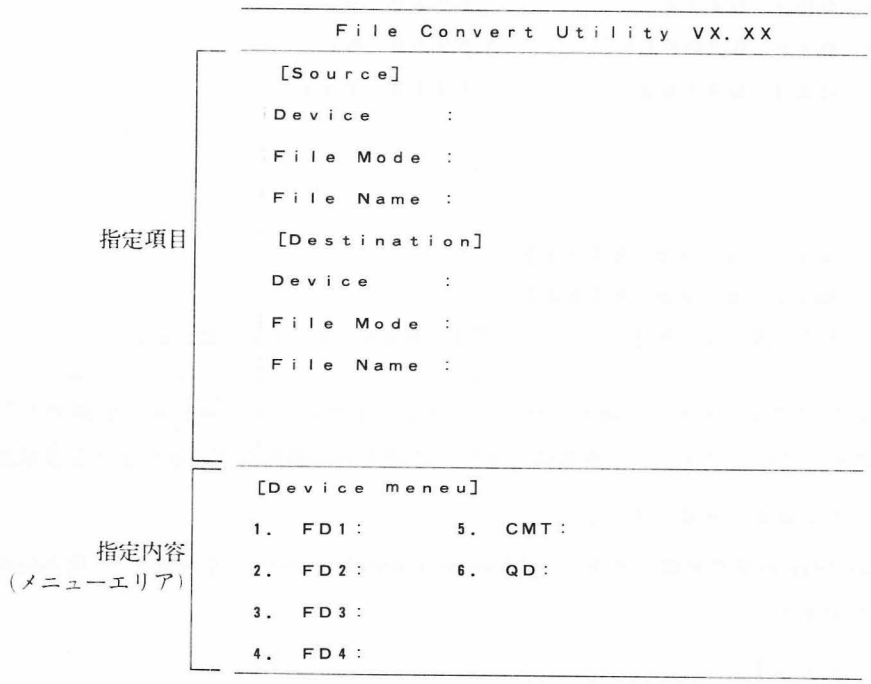
5.3 TRANS ユーティリティ

ファイルコンバータは、BASICのクイックディスク上に、" TRANS "というファイルネームで登録されています。これを実行させるには、

RUN "TRANS" CR

とキーを押してください。

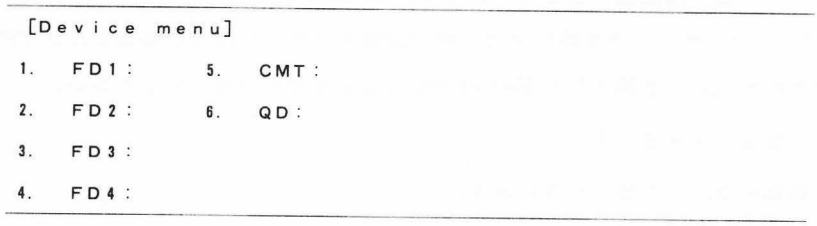
最初の画面に出てくる指定項目を上から順に指定しますが、指定の内容はメニューエリアに表示される中から適切なものを選んでその番号をキー入力してください。指定する項目と最初の画面は次のようになります。



☆ Device menu

・コンバートを行う場合の装置（デバイス）名を、ソースデバイス、ディスティネーションデバイスのそれぞれの場合について指定します。

〔表示内容〕



☆ File mode （ファイルモード）

・System menu と device menu で指定したコンバートの可能なファイルモードが表示されます。ただしディスティネーションのファイルモードは、ソースファイルモードで指定したものから自動的にコンバート可能なファイルモードとなります。

〔表示内容〕

```
[File Mode]
1.  OBJ
2.  BTX
3.  BSD
4.  BRD
```

☆ File name （ファイルネーム）

・コンバートするファイルのファイルネームを入力します。（最大16文字）

また、この状態でディスクのディレクトリを表示させたい場合は SHIFT ＋ CR をキー入力するとディレクトリ表示が行われます。

〔表示内容〕

```
[File Name]
shift + CR ==> Directory
```

［表示様式］ SHIFT ＋ CR キーを入力し、デバイスで6 （QD） を指定した場合の表示例。

```
File Convert Utility VX.XX
    [Directory]
    "DATA"
    "TEST"
    "SAMPLE"
{Function complete, ↵ Any key
 Next page, ↵ Any key
 (エラーメッセージ)
 Convert again [Y/N] ?}  のいずれか
```

・メッセージについて

Function complete, ↵ Any key

……キー入力（どのキーでも良い）を行うとファイルネームの入力待ち状態になります。

Next page , ↵ Any key

……ディレクトリ表示が１画面で足りなかった場合の表示です。もう一度キー入力すると残りのディレクトリ表示を行います。

（エラーメッセージ）

……何らかのエラーでディレクトリ表示ができない場合、エラーメッセージを表示します。

Convert again [Y/N] ?

……" Y "をキー入力するとソースシステムの入力待ち状態になり、" N "を入力すると最初に見つけたOBJファイルを読み込みます。

＊＊＊参　考＊＊＊

ファイルのコンバートを行う場合に、ソースドライブとデスティネーションドライブが同一の場合は次の表示が行われます。

[表示内容]

```
Set Source Disk, ↵ Any key
```

または

```
Set Destination Disk, ↵ Any key
```

［説　明］

デスティネーションファイルネームの入力が終った場合上記の表示が行われます。はじめに" ソースディスクを入れてください "という意味の表示が行われソースディスクをドライブに挿入したらキー(どのキーでも良い)を押してください。キー入力が行われた時点で、ソースディスクファイルの内容がMZ-2200本体のメモリ上に読み込まれます。読み込みが終了すると今度は、" デスティネーションディスクを入れてください "という意味の表示が行われます。ソースディスクを取り出し替りに、デスティネーションディスクをドライブに挿入してください。挿入し終ったらキー(どのキーでも良い)を押してください。キー入力が行われた時点でMZ-2200本体のメモリ上にあるデータがデスティネーションディスクへ書き込まれます。

コンバート終了時の表示について

・コンバート終了後は次のような表示になります。

```
メッセージ { **End of job**
           { Break
           { (エラーメッセージ)   } のどれか

Convert again [Y/N] ?
```

［説　明］

"End of job"のメッセージはコンバートが終りました、という意味です

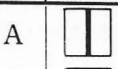
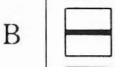
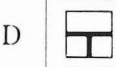
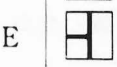
"Break"はコンバートを途中で中断（SHIFT＋BREAK）した場合のメッセージです。

"エラーメッセージ"はコンバートの途中でエラーが発生した場合のメッセージです。いずれもメッセージ表示終了後、もう一度コンバートを行いますか？、それともコンバートを終了しますか、という意味の表示を行います。もう一度コンバートを行うのならば"Y"を入力し、終了ならば"N"を入力してください。"Y"を入力するとコンバートの実行の初期画面へ、"N"を入力すると先頭にあるOBJファイルを読み込みます。

5.4 ASCII コード表

MZ-2200 システムの ASCII コード表を次に示します。

10進	16進	キャラクタ	10進	16進	キャラクタ	10進	16進	キャラクタ	10進	16進	キャラクタ	10進	16進	キャラクタ
0	00	NULL	26	1A		52	34	4	78	4E	N	104	68	h
1	01	↓	27	1B		53	35	5	79	4F	O	105	69	i
2	02	↑	28	1C		54	36	6	80	50	P	106	6A	j
3	03	→	29	1D		55	37	7	81	51	Q	107	6B	k
4	04	←	30	1E	■	56	38	8	82	52	R	108	6C	l
5	05	HOME	31	1F	▚	57	39	9	83	53	S	109	6D	m
6	06	CLR	32	20		58	3A	:	84	54	T	110	6E	n
7	07	DEL	33	21	!	59	3B	;	85	55	U	111	6F	o
8	08	INST	34	22	"	60	3C	<	86	56	V	112	70	p
9	09	GRPH	35	23	#	61	3D	=	87	57	W	113	71	q
10	0A	SET LOCK	36	24	$	62	3E	>	88	58	X	114	72	r
11	0B		37	25	%	63	3F	?	89	59	Y	115	73	s
12	0C	カナ	38	26	&	64	40	@	90	5A	Z	116	74	t
13	0D		39	27	'	65	41	A	91	5B	[	117	75	u
14	0E	L SCRIPT	40	28	(	66	42	B	92	5C	¥	118	76	v
15	0F	カナ CANCEL	41	29	)	67	43	C	93	5D	]	119	77	w
16	10		42	2A	*	68	44	D	94	5E	^	120	78	x
17	11		43	2B	+	69	45	E	95	5F	―	121	79	y
18	12		44	2C	,	70	46	F	96	60	`	122	7A	z
19	13		45	2D	―	71	47	G	97	61	a	123	7B	{
20	14		46	2E	.	72	48	H	98	62	b	124	7C	|
21	15		47	2F	/	73	49	I	99	63	c	125	7D	}
22	16		48	30	0	74	4A	J	100	64	d	126	7E	~
23	17		49	31	1	75	4B	K	101	65	e	127	7F	↵
24	18		50	32	2	76	4C	L	102	66	f			
25	19		51	33	3	77	4D	M	103	67	g			

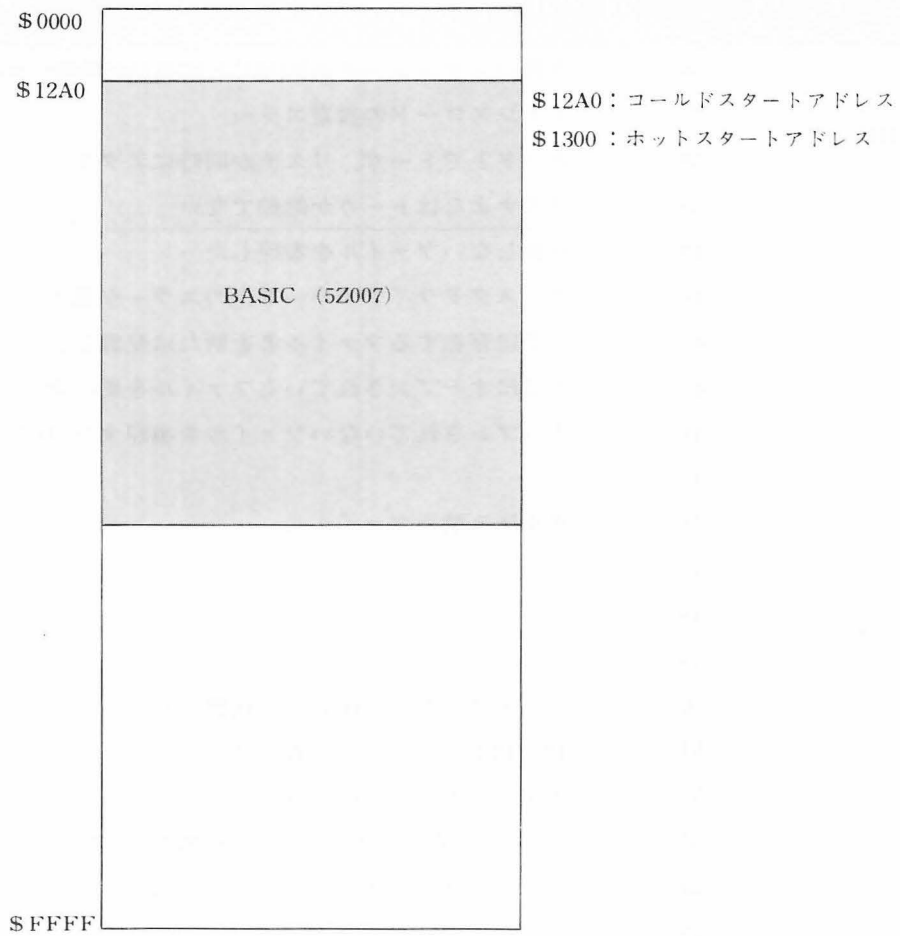
10進	16進	キャラクタ	10進	16進	キャラクタ	10進	16進	キャラクタ	10進	16進	キャラクタ	10進	16進	キャラクタ
128	80		154	9 A		180	B4	エ	206	CE	ホ	232	E8	H
129	81		155	9 B		181	B5	オ	207	CF	マ	233	E9	I
130	82		156	9 C		182	B6	カ	208	D0	ミ	234	EA	J
131	83		157	9 D		183	B7	キ	209	D1	ム	235	EB	K
132	84		158	9 E		184	B8	ク	210	D2	メ	236	EC	L
133	85		159	9 F		185	B9	ケ	211	D3	モ	237	ED	M
134	86		160	A0	V	186	BA	コ	212	D4	ヤ	238	EE	N
135	87		161	A1	｡	187	BB	サ	213	D5	ユ	239	EF	O
136	88		162	A2	｢	188	BC	シ	214	D6	ヨ	240	F0	Ø
137	89		163	A3	｣	189	BD	ス	215	D7	ラ	241	F1	I
138	8A		164	A4	W	190	BE	セ	216	D8	リ	242	F2	2
139	8B		165	A5	X	191	BF	ソ	217	D9	ル	243	F3	3
140	8C		166	A6	ヲ	192	C0	タ	218	DA	レ	244	F4	4
141	8D		167	A7	ア	193	C1	チ	219	DB	ロ	245	F5	5
142	8E		168	A8	イ	194	C2	ツ	220	DC	ワ	246	F6	6
143	8F		169	A9	ウ	195	C3	テ	221	DD	ン	247	F7	7
144	90		170	AA	エ	196	C4	ト	222	DE	゛	248	F8	8
145	91	¥	171	AB	オ	197	C5	ナ	223	DF	°	249	F9	9
146	92	U	172	AC	ヤ	198	C6	ニ	224	E0	Z	250	FA	P
147	93		173	AD	ユ	199	C7	ヌ	225	E1	A	251	FB	Q
148	94		174	AE	ヨ	200	C8	ネ	226	E2	B	252	FC	R
149	95		175	AF	ツ	201	C9	ノ	227	E3	C	253	FD	S
150	96		176	B0	Y	202	CA	ハ	228	E4	D	254	FE	T
151	97		177	B1	ア	203	CB	ヒ	229	E5	E	255	FF	π
152	98		178	B2	イ	204	CC	フ	230	E6	F			
153	99		179	B3	ウ	205	CD	ヘ	231	E7	G			

5.5 エラーメッセージ一覧表

	エラー番号	エラーの内容
	1	文法上の誤り
	2	数値データが範囲外、演算結果がオーバフローした
	3	規定外の数値、変数が使われた
	4	データと変数の型が一致しない
	5	ストリングの長さが255文字を越えた
	6	メモリ容量不足となった
	7	同じ配列変数を前より大きく定義したか、未定義の配列変数を使用した
	8	1行の長さが制限を越えた
	9	KINPUT 文でポジションが範囲外
	10	GOSUB文のネスティングが15を越えた
	11	FOR～NEXT文のネスティングが15を越えた
	12	DEF FN文による関数定義のネスティングが6を越えた
	13	対応するFOR文のないNEXT文が使われた
	14	対応するGOSUB文のないRETURN文が使われた
	15	定義されていない関数が使われた
	16	存在しない行番号を参照しようとした
	17	CONT文によるプログラムの継続ができない
	18	BASICインタープリタの管理エリア内への書き込み要求をした
	19	ダイレクトコマンドとステートメントを混同して使った
	20	RESUMEが実行できない
	21	エラーが発生していないのにRESUMEしようとした
	22	
	23	
	24	対応するDATA文のないREAD文が使われた
	25	
	26	
	27	
	28	
RS-232C	29	フレミングエラー
	30	オーバーランエラー
	31	パリティエラー
	32	データ転送不能（バッファが空でない）
	33	バッファのオーバーフロー
GP-IB	34	システムコントローラでないのにその命令を使用した
	35	カードまたはケーブルが接続されていない

	エラー番号	エラーの内容
GP-IB	36	3線ハンドシェークのNDACとNDTDが同時にHighレベル
	37	アドレスコードの設定エラー
	38	カード上でトーカ、リスナが同時にアクティブ
	39	リスナまたはトーカが能動でない
ディスク	40	存在しないファイルを参照した
	41	ディスクドライブのハード上のエラーが発生した
	42	すでに存在するファイル名を新たに登録しようとした
	43	すでにオープンされているファイルを更にオープンしようとした
	44	オープンされていないファイルを参照またはCLOSE、KILLした
	45	
	46	書き込み禁止ファイル
	47	
	48	
	49	
	50	ディスクドライブがレディ状態でない
	51	32を越すファイルを登録しようとした
	52	ボリュームナンバーエラー
	53	ディスク上のファイルスペースが無くなった
	54	イニシャライズされていないディスク
	55	
	56	ディスクドライブコントロールルーチンでのデータエラー
	57	使用不能ディスク
ディスク または カセット	58	
	59	
	60	ファイル名エラー
	61	ファイルモードエラー
	62	
	63	カセットテープファイルデータ読み込みで、アウトオブファイルが起きた
	64	ロジカルナンバーエラー
プリンタ	65	プリンタが接続されていないか、OFF状態である
	66	プリンタにメカトラブルが起きた
	67	プリンタ用紙ぎれ
	68	
	69	
	70	チェックサムエラー

5.6 起動時のメモリマップ



5.7 JIS漢字コード表 (▒の部分はJIS第一水準以外の漢字及び記号、符号である。)

	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	2120	㈱	㈸	▒	▒	▒	▒	▒	：	；	？	！	▒	▒	▒	▒	▒
	2130	▒	￣	＿	、	〃	〉	〃	〃	仝	々	〆	○	―	―	-	／
	2140	＼	～	∥	｜	…	‥	▒	▒	▒	▒	（	）	〔	〕	［	］
	2150	｛	｝	〈	〉	《	》	「	」	『	』	【	】	＋	－	±	×
	2160	÷	＝	≠	＜	＞	≦	≧	∞	∴	♂	♀	▒	▒	▒	℃	￥
	2170	＄	¢	£	％	＃	＆	＊	＠	§	☆	★	○	●	◎	◇	㈱
	2220	㈲	◆	□	■	△	▲	▽	▼	※	〒	→	←	↑	↓	〓	
非漢字	2330	0	1	2	3	4	5	6	7	8	9	‰	¬	∵	≒	≡	½
	2340	¼	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	2350	P	Q	R	S	T	U	V	W	X	Y	Z	◁	▷	㊗	KK	TEL
	2360	No.	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	2370	p	q	r	s	t	u	v	w	x	y	z	%	⇨	⇦	⇧	⇩
漢字	2420	㈾	ぁ	あ	ぃ	い	ぅ	う	ぇ	え	ぉ	お	か	が	き	ぎ	く
	2430	ぐ	け	げ	こ	ご	さ	ざ	し	じ	す	ず	せ	ぜ	そ	ぞ	た
	2440	だ	ち	ぢ	っ	つ	づ	て	で	と	ど	な	に	ぬ	ね	の	は
	2450	ば	ぱ	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ	ぼ	ぽ	ま	み
	2460	む	め	も	ゃ	や	ゅ	ゆ	ょ	よ	ら	り	る	れ	ろ	ゎ	わ
	2470	ゐ	ゑ	を	ん	lb	ha	oz	yd	ft	in	mb	☎	ms	μs	ns	ps
	2520	㈹	ァ	ア	ィ	イ	ゥ	ウ	ェ	エ	ォ	オ	カ	ガ	キ	ギ	ク
	2530	グ	ケ	ゲ	コ	ゴ	サ	ザ	シ	ジ	ス	ズ	セ	ゼ	ソ	ゾ	タ
	2540	ダ	チ	ヂ	ッ	ツ	ヅ	テ	デ	ト	ド	ナ	ニ	ヌ	ネ	ノ	ハ
	2550	バ	パ	ヒ	ビ	ピ	フ	ブ	プ	ヘ	ベ	ペ	ホ	ボ	ポ	マ	ミ
	2560	ム	メ	モ	ャ	ヤ	ュ	ユ	ョ	ヨ	ラ	リ	ル	レ	ロ	ヮ	ワ
	2570	ヰ	ヱ	ヲ	ン	ヴ	ヵ	ヶ	㎎	㎏	㏄	㎖	㎗	ℓ	㎘	Hz	HP
	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

※コードは16進で表現されています。例えば〝B〟のコードは2340+2=2342となります。

	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
非漢字	2620	㍻	A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ	O
	2630	Π	P	Σ	T	ϒ	Φ	X	Ψ	Ω	㎜	㎝	㎞	㎠	㎡	㎢	㎤
	2640	㎥	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	o
	2650	π	ρ	σ	τ	υ	φ	χ	ψ	ω	Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ
	2660	Ⅷ	Ⅸ	Ⅹ	i	ii	iii	iv	v	vi	vii	viii	ix	x	♧	♡	♤
	2670	◇	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)					
	2720	㍼	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н
	2730	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э
	2740	Ю	Я														
	2750		а	б	в	г	д	е	ё	ж	з	и	й	к	л	м	н
	2760	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э
	2770	ю	я														
	2820		!	”	#	$	%	&	’	(	)	*	+	,	−	.	/
	2830	0	1	2	3	4	5	6	7	8	9	:	;	〈	=	〉	?
	2840	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	2850	P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	-
	2860	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	2870	p	q	r	s	t	u	v	w	x	y	z	{	|	}	‾	
ア	3020	萬	亜	唖	娃	阿	哀	愛	挨	姶	逢	葵	茜	穐	悪	握	渥
	3030	旭	葦	芦	鯵	梓	圧	斡	扱	宛	姐	虻	飴	絢	綾	鮎	或
	3040	粟	袷	安	庵	按	暗	案	闇	鞍	杏						
イ	3040											以	伊	位	依	偉	囲
	3050	夷	委	威	尉	惟	意	慰	易	椅	為	畏	異	移	維	緯	胃
	3060	萎	衣	謂	違	遺	医	井	亥	域	育	郁	磯	一	壱	溢	逸
	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
イ	3070	稲	茨	芋	鰯	允	印	咽	員	因	姻	引	飲	淫	胤	蔭	侑
	3120	噫	院	陰	隠	韻	吋										
ウ	3120							右	宇	烏	羽	迂	雨	卯	鵜	窺	丑
	3130	碓	臼	渦	嘘	唄	欝	蔚	鰻	姥	厩	浦	瓜	閏	噂	云	運
	3140	雲															
エ	3140		荏	餌	叡	営	嬰	影	映	曳	栄	永	泳	洩	瑛	盈	穎
	3150	頴	英	衛	詠	鋭	液	疫	益	駅	悦	謁	越	閲	榎	厭	円
	3160	園	堰	奄	宴	延	怨	掩	援	沿	演	炎	焔	煙	燕	猿	縁
	3170	艶	苑	薗	遠	鉛	鴛	塩									
オ	3170								於	汚	甥	凹	央	奥	往	応	澤
	3220	皓	押	旺	横	欧	殴	王	翁	襖	鴬	鴎	黄	岡	沖	荻	億
	3230	屋	憶	臆	桶	牡	乙	俺	卸	恩	温	穏	音				
力	3230													下	化	仮	何
	3240	伽	価	佳	加	可	嘉	夏	嫁	家	寡	科	暇	果	架	歌	河
	3250	火	珂	禍	禾	稼	箇	花	苛	茄	荷	華	菓	蝦	課	嘩	貨
	3260	迦	過	霞	蚊	俄	峨	我	牙	画	臥	芽	蛾	賀	雅	餓	駕
	3270	介	会	解	回	塊	壊	廻	快	怪	悔	恢	懐	戒	拐	改	絛
	3320	迪	魁	晦	械	海	灰	界	皆	絵	芥	蟹	開	階	貝	凱	劾
	3330	外	咳	害	崖	慨	概	涯	碍	蓋	街	該	鎧	骸	浬	馨	蛙
	3340	垣	柿	蠣	鈎	劃	嚇	各	廓	拡	攪	格	核	殻	獲	確	穫
	3350	覚	角	赫	較	郭	閣	隔	革	学	岳	楽	額	顎	掛	笠	樫
	3360	橿	梶	鰍	潟	割	喝	恰	括	活	渇	滑	葛	褐	轄	且	鰹
	3370	叶	椛	樺	鞄	株	兜	竈	蒲	釜	鎌	噛	鴨	栢	茅	萱	暉
	3420	蕪	粥	刈	苅	瓦	乾	侃	冠	寒	刊	勘	勧	巻	喚	堪	姦
	3430	完	官	寛	干	幹	患	感	慣	憾	換	敢	柑	桓	棺	款	歓
	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
力	3440	汗	漢	潤	灌	環	甘	監	看	竿	管	簡	緩	缶	翰	肝	艦
	3450	莞	観	諌	貫	還	鑑	間	閑	関	陥	韓	館	舘	丸	含	岸
	3460	厳	玩	癌	眼	岩	翫	贋	雁	頑	顔	願					
キ	3460												企	伎	危	喜	器
	3470	基	奇	嬉	寄	岐	希	幾	忌	揮	机	旗	既	期	棋	棄	𥴠
	3520	𠮟	機	帰	毅	気	汽	畿	祈	季	稀	紀	徽	規	記	貴	起
	3530	軌	輝	飢	騎	鬼	亀	偽	儀	妓	宜	戯	技	擬	欺	犠	疑
	3540	祇	義	蟻	誼	議	掬	菊	鞠	吉	吃	喫	桔	橘	詰	砧	杵
	3550	黍	却	客	脚	虐	逆	丘	久	仇	休	及	吸	宮	弓	急	救
	3560	朽	求	汲	泣	灸	球	究	窮	笈	級	糾	給	旧	牛	去	居
	3570	巨	拒	拠	挙	渠	虚	許	距	鋸	漁	禦	魚	亨	享	京	𧏛
	3620	𧘇	供	侠	僑	兇	競	共	凶	協	匡	卿	叫	喬	境	峡	強
	3630	彊	怯	恐	恭	挟	教	橋	況	狂	狭	矯	胸	脅	興	蕎	郷
	3640	鏡	響	饗	驚	仰	凝	尭	暁	業	局	曲	極	玉	桐	粁	僅
	3650	勤	均	巾	錦	斤	欣	欽	琴	禁	禽	筋	緊	芹	菌	衿	襟
	3660	謹	近	金	吟	銀											
ク	3660						九	倶	句	区	狗	玖	矩	苦	躯	駆	駈
	3670	駒	具	愚	虞	喰	空	偶	寓	遇	隅	串	櫛	釧	屑	屈	𠮟
	3720	𧝞	掘	窟	沓	靴	曇	窪	熊	隈	粂	栗	繰	桑	鍬	勲	君
	3730	薫	訓	群	軍	郡											
ケ	3730						卦	袈	祁	係	傾	刑	兄	啓	圭	珪	型
	3740	契	形	径	恵	慶	慧	憩	掲	携	敬	景	桂	渓	畦	稽	系
	3750	経	継	繋	罫	茎	荊	蛍	計	詣	警	軽	頸	鶏	芸	迎	鯨
	3760	劇	戟	撃	激	隙	桁	傑	欠	決	潔	穴	結	血	訣	月	件
	3770	倹	倦	健	兼	券	剣	喧	圏	堅	嫌	建	憲	懸	拳	捲	𤚥
	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ケ	3820	𣏌	検	権	牽	犬	献	研	硯	絹	県	肩	見	謙	賢	軒	遣
	3830	鍵	険	顕	験	鹸	元	原	厳	幻	弦	減	源	玄	現	絃	舷
	3840	言	諺	限													
コ	3840				乎	個	古	呼	固	姑	孤	己	庫	弧	戸	故	枯
	3850	湖	狐	糊	袴	股	胡	菰	虎	誇	跨	鈷	雇	顧	鼓	五	互
	3860	伍	午	呉	吾	娯	後	御	悟	梧	檎	瑚	碁	語	誤	護	醐
	3870	乞	鯉	交	佼	侯	候	倖	光	公	功	効	勾	厚	口	向	𥫱
	3920	𡚴	后	喉	坑	垢	好	孔	孝	宏	工	巧	巷	幸	広	庚	康
	3930	弘	恒	慌	抗	拘	控	攻	昂	晃	更	杭	校	梗	構	江	洪
	3940	浩	港	溝	甲	皇	硬	稿	糠	紅	紘	紋	綱	耕	考	肯	肱
	3950	腔	膏	航	荒	行	衡	講	貢	購	郊	酵	鉱	礦	鋼	閤	降
	3960	項	香	高	鴻	剛	劫	号	合	壕	拷	濠	豪	轟	麹	克	刻
	3970	告	国	穀	酷	鵠	黒	獄	漉	腰	甑	忽	惚	骨	狛	込	躯
	3A20	桧	此	頃	今	困	坤	墾	婚	恨	懇	昏	昆	根	梱	混	痕
	3A30	紺	艮	魂													
サ	3A30				些	佐	叉	唆	嵯	左	差	査	沙	瑳	砂	詐	鎖
	3A40	裟	坐	座	挫	債	催	再	最	哉	塞	妻	宰	彩	才	採	栽
	3A50	歳	済	災	采	犀	砕	砦	祭	斎	細	菜	裁	載	際	剤	在
	3A60	材	罪	財	冴	坂	阪	堺	榊	肴	咲	崎	埼	碕	鷺	作	削
	3A70	咋	搾	昨	朔	柵	窄	策	索	錯	桜	鮭	笹	匙	冊	刷	𠮟
	3B20	𣑊	察	拶	撮	擦	札	殺	薩	雑	皐	鯖	捌	錆	鮫	皿	晒
	3B30	三	傘	参	山	惨	撒	散	桟	燦	珊	産	算	纂	蚕	讃	賛
	3B40	酸	餐	斬	暫	残											
シ	3B40						仕	仔	伺	使	刺	司	史	嗣	四	士	始
	3B50	姉	姿	子	屍	市	師	志	思	指	支	孜	斯	施	旨	枝	止
	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
シ	3B60	死	氏	獅	祉	私	糸	紙	紫	肢	脂	至	視	詞	詩	試	誌
	3B70	諮	資	賜	雌	飼	歯	事	似	侍	児	字	寺	慈	持	時	濱
	3C20	嚢	次	滋	治	爾	璽	痔	磁	示	而	耳	自	蒔	辞	汐	鹿
	3C30	式	識	鴫	竺	軸	宍	雫	七	叱	執	失	嫉	室	悉	湿	漆
	3C40	疾	質	実	蔀	篠	偲	柴	芝	屡	蕊	縞	舎	写	射	捨	赦
	3C50	斜	煮	社	紗	者	謝	車	遮	蛇	邪	借	勺	尺	杓	灼	爵
	3C60	酌	釈	錫	若	寂	弱	惹	主	取	守	手	朱	殊	狩	珠	種
	3C70	腫	趣	酒	首	儒	受	呪	寿	授	樹	綬	需	囚	収	周	皓
	3D20	諄	宗	就	州	修	愁	拾	洲	秀	秋	終	繍	習	臭	舟	蒐
	3D30	衆	襲	讐	蹴	輯	週	酋	酬	集	醜	什	住	充	十	従	戎
	3D40	柔	汁	渋	獣	縦	重	銃	叔	夙	宿	淑	祝	縮	粛	塾	熟
	3D50	出	術	述	俊	峻	春	瞬	竣	舜	駿	准	循	旬	楯	殉	淳
	3D60	準	潤	盾	純	巡	遵	醇	順	処	初	所	暑	曙	渚	庶	緒
	3D70	署	書	薯	藷	諸	助	叙	女	序	徐	恕	鋤	除	傷	償	洸
	3E20	砺	勝	匠	升	召	哨	商	唱	嘗	奨	妾	娼	宵	将	小	少
	3E30	尚	庄	床	廠	彰	承	抄	招	掌	捷	昇	昌	昭	晶	松	梢
	3E40	樟	樵	沼	消	渉	湘	焼	焦	照	症	省	硝	礁	祥	称	章
	3E50	笑	粧	紹	肖	菖	蒋	蕉	衝	裳	訟	証	詔	詳	象	賞	醤
	3E60	鉦	鍾	鐘	障	鞘	上	丈	丞	乗	冗	剰	城	場	壌	嬢	常
	3E70	情	擾	条	杖	浄	状	畳	穣	蒸	譲	醸	錠	嘱	埴	飾	拭
	3F20	廣	拭	植	殖	燭	織	職	色	触	食	蝕	辱	尻	伸	信	侵
	3F30	唇	娠	寝	審	心	慎	振	新	晋	森	榛	浸	深	申	疹	真
	3F40	神	秦	紳	臣	芯	薪	親	診	身	辛	進	針	震	人	仁	刃
	3F50	塵	壬	尋	甚	尽	腎	訊	迅	陣	靱						
ス	3F50											笥	諏	須	酢	図	厨
	3F60	逗	吹	垂	帥	推	水	炊	睡	粋	翠	衰	遂	酔	錐	錘	随
	3F70	瑞	髄	崇	嵩	数	枢	趨	雛	据	杉	椙	菅	頗	雀	裾	邊
	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ス	4020	踪	澄	摺	寸												
セ	4020					世	瀬	畝	是	凄	制	勢	姓	征	性	成	政
	4030	整	星	晴	棲	栖	正	清	牲	生	盛	精	聖	声	製	西	誠
	4040	誓	請	逝	醒	青	静	斉	税	脆	隻	席	惜	戚	斥	昔	析
	4050	石	積	籍	績	脊	責	赤	跡	蹟	碩	切	拙	接	摂	折	設
	4060	窃	節	説	雪	絶	舌	蝉	仙	先	千	占	宣	専	尖	川	戦
	4070	扇	撰	栓	栴	泉	浅	洗	染	潜	煎	煽	旋	穿	箭	線	栄
	4120	崔	繊	羨	腺	舛	船	薦	詮	賤	践	選	遷	銭	銑	閃	鮮
	4130	前	善	漸	然	全	禅	繕	膳	糎							
ソ	4130										噌	塑	岨	措	曾	曽	楚
	4140	狙	疏	疎	礎	祖	租	粗	素	組	蘇	訴	阻	遡	鼠	僧	創
	4150	双	叢	倉	喪	壮	奏	爽	宋	層	匝	惣	想	捜	掃	挿	掻
	4160	操	早	曹	巣	槍	槽	漕	燥	争	痩	相	窓	糟	総	綜	聡
	4170	草	荘	葬	蒼	藻	装	走	送	遭	鎗	霜	騒	像	増	憎	鉄
	4220	輌	臓	蔵	贈	造	促	側	則	即	息	捉	束	測	足	速	俗
	4230	属	賊	族	続	卒	袖	其	揃	存	孫	尊	損	村	遜		
タ	4230															他	多
	4240	太	汰	詫	唾	堕	妥	惰	打	柁	舵	楕	陀	駄	騨	体	堆
	4250	対	耐	岱	帯	待	怠	態	戴	替	泰	滞	胎	腿	苔	袋	貸
	4260	退	逮	隊	黛	鯛	代	台	大	第	醍	題	鷹	滝	瀧	卓	啄
	4270	宅	托	択	拓	沢	濯	琢	託	鐸	濁	諾	茸	凧	蛸	只	尹
	4320	浙	叩	但	達	辰	奪	脱	巽	竪	辿	棚	谷	狸	鱈	樽	誰
	4330	丹	単	嘆	坦	担	探	旦	歎	淡	湛	炭	短	端	箪	綻	耽
	4340	胆	蛋	誕	鍛	団	壇	弾	断	暖	檀	段	男	談			
	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
チ	4340														値	知	地
	4350	弛	恥	智	池	痴	稚	置	致	蜘	遅	馳	築	畜	竹	筑	蓄
	4360	逐	秩	窒	茶	嫡	着	中	仲	宙	忠	抽	昼	柱	注	虫	衷
	4370	註	酎	鋳	駐	樗	瀦	猪	苧	著	貯	丁	兆	凋	喋	寵	𨏍
	4420	𠮟	帖	帳	庁	弔	張	彫	徴	懲	挑	暢	朝	潮	牒	町	眺
	4430	聴	脹	腸	蝶	調	諜	超	跳	銚	長	頂	鳥	勅	捗	直	朕
	4440	沈	珍	賃	鎮	陳											
ツ	4440						津	墜	椎	槌	追	鎚	痛	通	塚	栂	掴
	4450	槻	佃	漬	柘	辻	蔦	綴	鍔	椿	潰	坪	壺	嬬	紬	爪	吊
	4460	釣	鶴														
テ	4460			亭	低	停	偵	剃	貞	呈	堤	定	帝	底	庭	廷	弟
	4470	悌	抵	挺	提	梯	汀	碇	禎	程	締	艇	訂	諦	蹄	逓	𠮷
	4520	𥫱	邸	鄭	釘	鼎	泥	摘	擢	敵	滴	的	笛	適	鏑	溺	哲
	4530	徹	撤	轍	迭	鉄	典	填	天	展	店	添	纏	甜	貼	転	顚
	4540	点	伝	殿	澱	田	電										
ト	4540							兎	吐	堵	塗	妬	屠	徒	斗	杜	渡
	4550	登	菟	贈	途	都	鍍	砥	礪	努	度	土	奴	怒	倒	党	冬
	4560	凍	刀	唐	塔	塘	套	宕	島	嶋	悼	投	搭	東	桃	樋	棟
	4570	盗	淘	湯	涛	灯	燈	当	痘	禱	等	答	筒	糖	統	到	𢘑
	4620	𥳑	董	蕩	藤	討	謄	豆	踏	逃	透	鐙	陶	頭	騰	闘	働
	4630	動	同	堂	導	憧	撞	洞	瞳	童	胴	萄	道	銅	峠	鴇	匿
	4640	得	徳	瀆	特	督	禿	篤	毒	独	読	栃	橡	凸	突	松	届
	4650	鳶	苫	寅	酉	瀞	噸	屯	惇	敦	沌	豚	遁	頓	呑	曇	鈍
ナ	4660	奈	那	内	乍	凪	薙	謎	灘	捺	鍋	楢	馴	縄	畷	南	楠
	4670	軟	難	汝													
	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
二	4670				二	尼	弐	邇	匂	賑	肉	虹	廿	日	乳	入	蓑
	4720	濾	如	尿	韮	任	妊	忍	認								
又	4720									濡							
ネ	4720											禰	祢	寧	葱	猫	熱	年
	4730	念	捻	撚	燃	粘											
ノ	4730							乃	廼	之	埜	嚢	悩	濃	納	能	脳	膿
	4740	農	覗	蚤													
八	4740				巴	把	播	覇	杷	波	派	琶	破	婆	罵	芭	馬
	4750	俳	廃	拝	排	敗	杯	盃	牌	背	肺	輩	配	倍	培	媒	梅
	4760	楳	煤	狽	買	売	賠	陪	這	蝿	秤	短	萩	伯	剝	博	拍
	4770	柏	泊	白	箔	粕	舶	薄	迫	曝	漠	爆	縛	莫	駁	麦	麺
	4820	箸	函	箱	硲	箸	肇	筈	櫨	幡	肌	畑	畠	八	鉢	溌	発
	4830	醗	髪	伐	罰	抜	筏	閥	鳩	噺	塙	蛤	隼	伴	判	半	反
	4840	叛	帆	搬	斑	板	氾	汎	版	犯	班	畔	繁	般	藩	販	範
	4850	禾	煩	頒	飯	挽	晩	番	盤	磐	蕃	蛮					
ヒ	4850												匪	卑	否	妃	庇
	4860	彼	悲	扉	批	披	斐	比	泌	疲	皮	碑	秘	緋	罷	肥	被
	4870	誹	費	避	非	飛	樋	簸	備	尾	微	枇	毘	琵	眉	美	鼻
	4920	黎	鼻	柊	稗	匹	疋	髭	彦	膝	菱	肘	弼	必	畢	筆	逼
	4930	檜	姫	媛	紐	百	謬	俵	彪	標	氷	漂	瓢	票	表	評	豹
	4940	廟	描	病	秒	苗	錨	鋲	蒜	蛭	鰭	品	彬	斌	浜	瀕	貧
	4950	賓	頻	敏	瓶												
	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
フ	4950					不	付	埠	夫	婦	富	冨	布	府	怖	扶	敷
	4960	斧	普	浮	父	符	腐	膚	芙	譜	負	賦	赴	阜	附	侮	撫
	4970	武	舞	葡	蕪	部	封	楓	風	葺	蹈	伏	副	復	幅	服	郁
	4A20	刺	福	腹	複	覆	淵	弗	払	沸	仏	物	鮒	分	吻	噴	墳
	4A30	憤	扮	焚	奮	粉	糞	紛	雰	文	聞						
ヘ	4A30											丙	併	兵	塀	幣	平
	4A40	弊	柄	並	蔽	閉	陛	米	頁	僻	壁	癖	碧	別	瞥	蔑	箆
	4A50	偏	変	片	篇	編	辺	返	遍	便	勉	娩	弁	鞭			
ホ	4A50														保	舗	鋪
	4A60	圃	捕	歩	甫	補	輔	穂	募	墓	慕	戊	暮	母	簿	菩	倣
	4A70	俸	包	呆	報	奉	宝	峰	峯	崩	庖	抱	捧	放	方	朋	敬
	4B20	晨	法	泡	烹	砲	縫	胞	芳	萌	蓬	蜂	褒	訪	豊	邦	鋒
	4B30	飽	鳳	鵬	乏	亡	傍	剖	坊	妨	帽	忘	忙	房	暴	望	某
	4B40	棒	冒	紡	肪	膨	謀	貌	貿	鉾	防	吠	頬	北	僕	卜	墨
	4B50	撲	朴	牧	睦	穆	釦	勃	没	殆	堀	幌	奔	本	翻	凡	盆
マ	4B60	摩	磨	魔	麻	埋	妹	昧	枚	毎	哩	槇	幕	膜	枕	鮪	柾
	4B70	鱒	桝	亦	俣	又	抹	末	沫	迄	儘	繭	麿	万	慢	満	糀
	4C20	國	漫	蔓													
ミ	4C20				味	未	魅	巳	箕	岬	密	蜜	湊	蓑	稔	脈	妙
	4C30	粍	民	眠													
ム	4C30				務	夢	無	牟	矛	霧	鵡	椋	婿	娘			
	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
メ	4C30														冥	名	命
	4C40	明	盟	迷	銘	鳴	姪	牝	滅	免	棉	綿	緬	面	麺		
モ	4C40															摸	模
	4C50	茂	妄	孟	毛	猛	盲	網	耗	蒙	儲	木	黙	目	杢	勿	餅
	4C60	尤	戻	籾	貰	問	悶	紋	門	匁							
ヤ	4C60										也	冶	夜	爺	耶	野	弥
	4C70	矢	厄	役	約	薬	訳	躍	靖	柳	薮	鑓					
ユ	4C70												愉	愈	油	癒	繍
	4D20	趙	諭	輸	唯	佑	優	勇	友	宥	幽	悠	憂	揖	有	柚	湧
	4D30	涌	猶	猷	由	祐	裕	誘	遊	邑	郵	雄	融	夕			
ヨ	4D30														予	余	与
	4D40	誉	輿	預	傭	幼	妖	容	庸	揚	揺	擁	曜	楊	様	洋	溶
	4D50	熔	用	窯	羊	耀	葉	蓉	要	謡	踊	遙	陽	養	慾	抑	欲
	4D60	沃	浴	翌	翼	淀											
ラ	4D60							羅	螺	裸	来	莱	頼	雷	洛	絡	落	酪
	4D70	乱	卵	嵐	欄	濫	藍	蘭	覧								
リ	4D70									利	吏	履	李	梨	理	璃	眞
	4E20	獺	痢	裏	裡	里	離	陸	律	率	立	葎	掠	略	劉	流	溜
	4E30	琉	留	硫	粒	隆	竜	龍	侶	慮	旅	虜	了	亮	僚	両	凌
	4E40	寮	料	梁	涼	猟	療	瞭	稜	糧	良	諒	遼	量	陵	領	力
	4E50	緑	倫	厘	林	淋	燐	琳	臨	輪	隣	鱗	麟				
	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ル	4E50													瑠	璽	涙	累
	4E60	類															
レ	4E60			令	伶	例	冷	励	嶺	怜	玲	礼	苓	鈴	隷	零	霊	麗
	4E70	齢	暦	歴	列	劣	烈	裂	廉	恋	憐	漣	煉	簾	練	聯	𤏋
	4F20	鐐	蓮	連	錬												
ロ	4F20					呂	魯	櫓	炉	賂	路	露	労	婁	廊	弄	朗
	4F30	楼	榔	浪	漏	牢	狼	篭	老	聾	蠟	郎	六	麓	禄	肋	録
	4F40	論															
ワ	4F40		倭	和	話	歪	賄	脇	惑	枠	鷲	亙	亘	鰐	詫	藁	蕨
	4F50	椀	湾	碗	腕	𠮟	𠮷	壓	顆	𨑨	巖	佶	悪	篭	晏	簇	齋
	4F60	鍈	俶	麒	實	暹	寔	滓	銓	瘍	晁	釉	恂	孚	岑	雍	圀
	4F70	絲	鈑	蜷	陞	纈	壽	悳	鞆	纈	靆	楡	巍	鋼	胖	翔	箋
	JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

5.8 クイックディスクの取扱い上の注意

- ディスクを使用する時は、エンベロープ(保護袋)の記載事項をお守りください。
- ディスクを使用しない時は、必ずエンベロープに入れ、収納箱などに垂直に立てて大切に保管してください。ななめに立てかけたり、ディスクがたわむような形での保管はさけてください。
- ディスクを曲げたり、折ったりしないでください。読み書きができなくなる場合もあります。
- ラベルへの書き込みは、ディスクに貼る前に行ってください。すでに貼られたラベルに書き込む場合は、鉛筆・ボールペンなどはさけ、サインペンなど、先の軟らかいものを使用してください。
- ディスクに磁石を近づけることは絶対にさけてください。
  磁気をおびたものを近づけるとディスクに書き込まれているプログラムやデータを破壊しますので、ご注意ください。

5.9 索　　引

A APPEND ····38
B BLINE ····51
BOX ····54
C CCOLOR ····45
CHAIN ····40
CHR$ ····35
CIRCLE ····55
CLOSE ····43
CMDR ····78
CMDW ····77
COLOR ····46
CONSOLE ····48
D DCL ····70
DCL n ····71
DELETE ····30
DIR ····36
DIR/P ····37
E EOIR ····82
EOIW ····82
G GPIBM ····82
GRAPH ····47
I ICL ····68
IF ERL ····29
IF ERN ····28
IF EOF (#) THEN ····44
INIT ····44
INPUT # ····42
J JTRANS ····57
K KCHR$ ····58
KILL ····43
KINPUT ····59
KPATTERN ····60
KPRINT/P ····61
L LCL ····69
LCL n ····69
LINE ····50
LLO ····70
LOAD ····38
O ON ERROR GOTO ····28
ON SRQ ····79
P PAINT ····53
PATTERN ····52
PCT ····72
POKE ····35
PPC ····80
PPOL ····81
PPU ····82
PRINT # ····41
PSEL ····31
R RED/ ····76
RED ····74
REN ····68
RENUM ····32
RESET ····49
RESUME ····29
ROPEN # ····42
RSI ····63
RSMODE ····62
RSO ····63
RUN ····39
S SAVE ····37
SEARCH ····33
SEARCH/P ····34
SET ····48
SPOL ····79
T TABSET ····34
TRG ····71
W WOPEN # ····41
WRT/ ····75
WRT ····73

シャープ株式会社

本　　社　〒545 大阪市阿倍野区長池町22番22号
電話(06)621－1221(大代表)
産業機器事業本部　〒639-11 奈良県大和郡山市美濃庄町492番地
電話(07435)3－5521(大代表)
国内産機営業本部　〒162 東京都新宿区市谷八幡町８番地
電話(03)260－1161(大代表)

お客様ご相談窓口

札　幌	(011)642-4649	仙　台	(0222)88-9141	宇都宮	(0286)35-1155
東　京	(03)893-4649	金　沢	(0762)49-4649	名古屋	(052)322-4649
大　阪	(06)643-4649	広　島	(082)874-4649	高　松	(0878)33-4649
福　岡	(092)572-4649	沖　縄	(0988)62-2231		

シャープエンジニアリング株式会社

本　　社	〒114 東京都北区東田端2丁目13番17号	電話(03)800-1221(代表)
札幌支店	〒063 札幌市西区24軒1条7丁目3番17号	電話(011)641－4649
仙台支店	〒983 仙台市六丁目字本新田東2番地の1	電話(0222)88－9141
宇都宮支店	〒320 宇都宮市不動前4丁目2番41号	電話(0286)35－1155
東京支店	〒114 東京都北区東田端2丁目13番17号	電話 (03)800－1221
金沢支店	〒921 石川県石川郡野々市町字御経塚1096番地の1	電話(0762)49－4649
名古屋支店	〒454 名古屋市中川区山王3丁目5番5号	電話(052)332－2626
大阪支店	〒556 大阪市浪速区恵美須西1丁目2番9号	電話 (06)643－4649
広島支店	〒731-01 広島市安佐南区祇園町大字西原2249番地の1	電話(082)874－2281
高松支店	〒760 高松市木太町1861番地の3	電話(0878)33－4649
福岡支店	〒816 福岡市博多区井相田2丁目12番地の1	電話(092)572－4655
沖縄シャープ電機㈱	〒900 那覇市曙2丁目10番地の1	電話(0988)62－2231

シャープビジネス株式会社

本　　社	〒545 大阪市阿倍野区長池町22番22号	電話(06)621-1221(大代表)
札幌支店	〒063 札幌市西区24軒1条7丁目3番17号	電話(011)641－3631
仙台支店	〒983 仙台市六丁目字本新田東2番地の1	電話(0222)88－9151
東京支店	〒130 東京都墨田区石原2丁目12番3号	電話 (03)625－5111(代表)
千葉支店	〒280 千葉市南町1丁目5番20号	電話(0472)63－4043
横浜支店	〒235 横浜市磯子区中原1丁目2番23号	電話(045)751－3215
埼玉支店	〒330 大宮市宮原町2丁目107番地2号	電話(0486)63－5159
宇都宮支店	〒320 宇都宮市不動前4丁目2番41号	電話(0286)37－3576
新潟支店	〒950 新潟市上所中1丁目7番21号	電話(0252)83－1795
長野支店	〒380 長野市中御所2丁目11番3号	電話(0262)28－4618
名古屋支店	〒454 名古屋市中川区山王3丁目5番5号	電話(052)332－2631(代表)
金沢支店	〒921 石川県石川郡野々市町字御経塚1096番地の1	電話(0762)49－1240
大阪支店	〒556 大阪市浪速区恵美須西1丁目2番9号	電話 (06)643－3021(代表)
京都支店	〒601 京都市南区上鳥羽菅田町48番地	電話(075)661－7335
神戸支店	〒658 神戸市東灘区魚崎北町1丁目6番地18号	電話(078)452－8531
広島支店	〒731-01 広島市安佐南区祇園町大字西原2249番地の1	電話(082)874－4925
高松支店	〒760 高松市木太町1861番地の3	電話(0878)33－4255
福岡支店	〒816 福岡市博多区井相田2丁目12番地の1	電話(092)572－2611
沖縄支店	〒900 那覇市曙2丁目10番地の1	電話(0988)61－7360(代表)

シャープ株式会社

4F 1.8501-I
TINSJ1210ACZZ

© 1984 SHARP CORPORATION