

PC8001 N-BASIC

リファレンス

* 赤字はDiscBASIC

特殊文字

*イタリックは PasoconMini 非対応

ctr+ B	カーソルを1項目毎に右へずらす
ctr+ C	プログラムの実行を中止して、コマンドモードへ戻る。 (入力待ち時のみ、通常は STOP キー)
ctr+ E	カーソル位置から後ろを抹消する
ctr+ G	ブザーを鳴らす
ctr+ H	1文字を削除
ctr+ I	タブ、タブストップは8カラムごと
ctr+ J	カーソル以降の文字を次の行に送る
ctr+ K	ホーム、カーソルを左上に戻す
ctr+ L	画面全体をクリアする
ctr+ N	カーソルを1項目ごとに右へずらす
ctr+ R	1文字を挿入
&O または &	8進定数を表す
&H	16進定数を表す
? (疑問符)	PRINT と同じ(ただし LPRINT として使えない)
' (アポストロフィ)	REM と同じ
. (ピリオド)	現在の行番号を示す
: (コロン)	マルチステートメントの場合の区切り記号
; (セミコロン)	PRINT 文中での区切り記号(改行無し)

型宣言文字

記号	型	内 容	使用バイト数
\$	ストリング	コードが 0 ~ 255 の文字	3+その文字数
%	整数	-32768 ~ 32767 の整数	2
!	単精度	7.1桁の浮動小数	4
#	倍精度	16.8桁の浮動小数	8

演算子

記号	機能
=	代入または等値判定
-	負数表示または減算
+	加算または文字列の結合
*	乗算
/	除算（結果が浮動小数点の場合）
^	べき乗
¥	除算（結果が整数の場合）
MOD	剰余（結果は整数）
NOT	否定または2の補数（結果は整数）
AND	論理積（結果は整数）
OR	論理和（整数）
XOR	排他的論理和（整数）
EQV	等価（XORの否定、整数）
IMP	インプリケーション（Xの否定またはY、整数）
=、<、>、<=、 =<、>=、=>、 <>、><	条件判定（整数、真ならば-1、偽ならば0）

演算子の優先順位は

- (1) 括弧でくくられた式
- (2) べき乗 (A ^ B)
- (3) 負数 (-A)
- (4) *、/
- (5) ¥
- (6) MOD
- (7) +、-
- (8) 条件判定
- (9) NOT
- (10) AND
- (11) OR
- (12) XOR
- (13) IMP
- (14) EQV

コマンド

[] はオプションで省略可能(2番目以降を使用する時はコンマの省略不可)

コマンド	書式・機能	使用例
AUTO	AUTO [[開始行]、行番号の間隔] 自動的に行番号を発生する	AUTO 10,10
CLOAD	CLOAD "ファイル名" カセットテープからプログラムをロードする CLOAD? "ファイル名" LOAD後のチェック(ベリファイ)を行う	CLOAD "A" CLOAD? "A"
CONT	CONT プログラム停止後の再実行	CONT
CSAVE	CSAVE "ファイル名" プログラムをカセットテープに記録(セーブ)する	CSAVE "ABC"
DELETE	DELETE [開始行番号] [-終了行番号] 指定した行を抹消する	DELETE 40 DELETE 40-100
FILES	FILES ディスク上のファイルの名前と大きさを表示する	FILES
FORMAT	FORMAT ディスクをフォーマットして使用可能にする	FORMAT
KEYLIST	KEYLIST ファンクションキーの内容を画面に全部リストする	KEYLIST
LFILES	LFILES プリンタへFILESを出力	LFILES
LIST	LIST [開始行番号][-終了行番号] プログラムを画面上にディスプレイする	LIST 40-100 LIST
LLIST	LLIST [開始行番号][-終了行番号] プリンタへLISTを出力	LLIST 200- LLIST -1000

LOAD	LOAD “ファイル名”[, R] ディスクからプログラムをメモリ内へロードする	LOAD "A", R
MERGE	MERGE “ファイル名” すでにメモリ上にあるプログラムを重ねてロードする	MERGE "Bプログラム"
MON	MON 機械語モニタへ飛ぶ（コントロールBでもどる）	MON
MOUNT	MOUNT[ドライブ番号[, ドライブ番号··]] ディスクを指定	MOUNT
NAME	NAME “旧ファイル名” AS “新ファイル名” ファイル名を変更する	NAME"OLD"AS"NEW"
NEW	NEW メモリにあるプログラムと、全ての変数をクリアする	NEW
RENUM	RENUM[[新番号][[, [旧番号][, 番号の間隔]]] プログラムの行番号を新しくつけなおす	RENUM RENUM 100,90,20
REMOVE	REMOVE[ドライブ番号[, ドライブ番号··]] ディスクを抜く前に準備（最新配置表書込み）をする	REMOVE 1
RUN	RUN [行番号] プログラムの実行を開始する	RUN RUN 50
SAVE	SAVE“ファイル名”[, A] プログラムをディスクへ保管	SAVE "PC", A
SET	SET ドライブ番号又は#ファイル番号 又は“ファイル名”, “属性文字” ファイルの属性を決定する	SET 1, “R” SET #1, “P” SET "A", ""
TERM	TERM コード, パリティ, ボーレート, フィールドSW ターミナルモードに移る	TERM A,0,1,1

ステートメント

(入出力、スクリーン、特殊ステートメントは除く)

コマンド	書式・機能	使用例
CLEAR	CLEAR [[文字領域の大きさ], メモリ上限] 変数を0または空白にクリアし、文字領域の大きさおよびユーザーメモリの上限を設定	CLEAR 500, 49152 CLEAR 500,&HC000
DATA	DATA 定数 [, 定数...] READ文で読まれる数値や文字を割り当てる	DATA 1, 0, 150, 0
DEF	DEF FN x (引数の並び) = 関数の手儀式 ユーザーの定義関数を定める	DEF FNA(A,Y)=X*Y S=FNA(A,B)
	DEF USR[数字] = 番地 機械語のユーザー・サブルーチンの開始アドレスを定める	DEF USR5 = 0 X = USR5(A)
DEFDBL	DEF x x x 変数名[一変数名]	DEFDBL A
DEFINT	指定した変数の型を、それぞれ倍精度、	DEFINT B-G
DEFSNG	整数、単精度、文字列型に定義	DEFSNG H,K
DEFSTR		DEFSTR X-Z
DIM	DIM 変数名 [要素の数[,要素の数...]] 配列の大きさを指定して、メモリ中に領域を確保する	DIM A(20) DIM X5\$(100,25,10,3)
END	END プログラムの実行を終了する	END
ERASE	ERASE 配列名[,配列名...] 配列を抹消する	ERASE A,B\$
FIELD	FIELD[#] ファイル番号、大きさ AS 文字変数 ランダムファイルバッファ中に変数の領域を割り当てる	FIELD 1, 20 AS M\$,25 AS X\$, 9 AS A\$

FOR	FOR 変数名 = 式 TO 式 [STEP 式] NEXTまでのループを条件に従って繰り返す	FOR J=0 TO 1000 STEP 2
GOSUB	GOSUB 行番号 指定された行から始まるサブルーチンへジャンプする	GOSUB 2000
GOTO	GOTO 行番号 指定された行へプログラム制御を移す	GOTO 10
IF	IF 式 THEN 文または行番号 [ELSE 文または行番号] 条件式が真ならばTHENの後の命令を、 偽ならばELSEの後の命令を実行する	IF X>Y THEN A=Y ELSE A=X
LET	LET 変数名=式 変数に値を代入する	LET A=5 B=0
LSET	LSET 領域名=文字列 PUT文に先立ち、ランダムデータバッファに情報を送る	LSET A\$="GOOD"
NEXT	NEXT[ループ変数[,ループ変数…]] FORループの範囲を定める	NEXT J
ON-GOSUB	ON 式 GOSUB 行番号[,行番号…] 式の値に対応して、相当するサブルーチンをコールする	ON J GOSUB 80,200,990
ON-GOTO	ON 式 GOTO 行番号[,行番号…] 式の値に応じて、相当する行番号へ制御を移す	ON X*Y GOTO 5,300,50
READ	READ 変数[,変数…] DATA文のデータを各変数に読み込む	READ A,B\$
REM	REM 任意の文字、文 プログラム中に注釈を付けるために使う。 REM文の行はマルチステートメント化できない	REM ***START***

RESTORE	RESTORE [行番号] DATA文を、]指定した行番号から読める ようにする	RESTORE 100
RETURN	RETURN サブルーチンから、そのサブルーチンを コールしたGOSUB文の次の行へ戻る	RETURN
RSET	RSET 領域名 = 文字列 ランダム・バッファに文字を右詰めにする	RSET X\$="A"
STOP	STOP プログラムの実行を中止し、ブレークメッ セージを出力してコマンドモードに戻る。	STOP
SWAP	SWAP 変数、変数 2つの変数の値を交換する	SWAP A,B

入出力ステートメント

コマンド	書式・機能	使用例
<i>CLOSE</i>	<i>CLOSE</i> [[#]ファイル番号],[#]ファイル番号…] ディスクファイルへの入出力操作を閉じる	<i>CLOSE</i> #2
<i>DSKO\$</i>	<i>DISKO\$</i> ドライブ番号,トラック番号、セクタ 指定したセクターへ文字列を書き込む	<i>DSKO\$</i> A,B,C
<i>GET</i>	<i>GET</i> [#] ファイル番号[,レコード番号] ランダム・ファイルからランダム・バッファへ 1レコード文を読み込む	<i>GET</i> #1
<i>INPUT</i>	<i>INPUT</i> ["プロンプト文";] 変数 [,変数、…] キーボードから入力する	<i>INPUT</i> "ツギノデータ";N
<i>INPUT#</i>	<i>INPUT#</i> ファイル番号 , 変数 [,変数…] シーケンシャルからデータを読み込み、変数に 代入する	<i>INPUT#</i> 1,A
<i>INPUT#-1</i>	<i>INPUT#-1</i> , 変数 [,変数…] カセットからプログラム中の変数に値を入力する	<i>INPUT#-1</i> ,A
<i>KILL</i>	<i>KILL</i> "ファイル名" ディスク上からファイルを削除する	<i>KILL</i> "NG"
<i>LINE INPUT</i>	<i>LINE INPUT</i> ["プロンプト文";]文字変数 1行全体を特殊文字を含めて区切ることなく入力する	<i>LINE INPUT</i> "NEXT";B\$
<i>LINE INPUT#</i>	<i>LINE INPUT#</i> ファイル番号、文字変数 シーケンシャルファイルより <i>LINE INPUT</i> 動作 を行う	<i>LINE INPUT#</i> 1,C\$
<i>LPRINT</i>	<i>LPRINT</i> 式[,式…]	<i>LPRINT</i> A,B
<i>LPRINT USING</i>	<i>LPRINT USING</i> "フォーマット文";式[,式…] プリンタに対する <i>PRINT</i> [<i>USING</i>]文 (<i>USING</i> の項も参照)	<i>LPRINT USING</i> "**#.#####";A,B

OPEN	OPEN“ファイル名”[FORモード]AS[#]ファイル番号 ファイルを使つての入出力を可能にする	OPEN“ABC”FOR INPUT AS#1
OUT	OUT ポート番号,データ 出力ポートへ信号を送る	OUT 32,100
POKE	POKE メモリ番地、データ メモリの指定された番地にデータを書き込む	POKE 32768,255 POKE &H5A00,&HFA
PRINT	PRINT [,または:または空白または式…] ディスプレイ上に情報を出力する PRINT USING “フォーマット文”;式[,式…] フォーマット文で決められた書式に従つて、 画面に情報を出力する(USINGの項も参照)	PRINT X,X+5;“ABC” PRINT USING “**¥##”;A,B,C
PRINT#	PRINT#ファイル番号[USING “フォーマット文”;]変数 [,または;または空白または式…] シーケンシャルファイルにデータを描く	PRINT#1,A\$;“.”;B\$
PRINT#-1	PRINT#-1,変数 [,または;または式…] カセットへのPRINT	PRINT#-1,A
PUT	PUT[#]ファイル番号[,レコード番号] ランダムファイルにランダムバッファから書き出す	PUT#1,5
WAIT	WAIT ポート番号、マスク[,論理スイッチ] 入力ポートをモニタする。 ($(\text{ポートデータ}) \oplus (\text{論理スイッチ}) \text{ AND } (\text{マスク})$) が真のとき次の行からプログラムの実行を再開する	WAIT 32,&H55,&F0

スクリーンステートメント

コマンド	書式・機能	使用例
COLOR	COLOR [ファンクション][,ヌルキャラクタコード][,グラフィックスイッチ] 画面に対する色や機能を決定する。パラメータはそれぞれ色または白黒の時の機能、クリアしたときのヌルコード、グラフィックモード（1のとき真）を表す	COLOR 2,0,1
CONSOLE	CONSOLE [スクロール開始行][, スクロール長さ] [, ファンクションキー表示スイッチ（1のとき表示）] [, カラー／白黒スイッチ（1のときカラー）]	CONSOLE 0,25,1,1
	スクロールの窓、ファンクションキー内容の表示、カラー／白黒モードの選択をする	
GET@	GET@ (X、Y) - (x,y), [配列] 画面上の情報を指定した配列の中に読み込む GET@A(X,Y)-(x,y), 配列 色やリバーズ機能なども同時に配列の中に取り込む	GET@(0,0)-(10,10),A% GET@(0,0)-(20,40)A%,G GET@A(0,0)-(14,14),B%
LINE	LINE 行数、ファンクションコード カラーの場合に1行単位で機能を決定する LINE(X,Y)-(x,y),文字列[ファンクションコード][,B[F]] 画面に文字を使って詮や箱を描く、色指定も可 LINE(X,Y)-(x,y),P[RE]SET[,ファンクションコード][,B[F]] 画面にグラフィックモードで詮や箱を書く	LINE 5,4 LINE(0,0)-(20,20),"♥",2,B LINE(0,0)-(159,99),PSET,2,B
LOCATE	LOCATE 水平位置、垂直位置[,カーソルスイッチ] 画面上のカーソルを動かす、スイッチは0でカーソル消滅	LOCATE 3,20,0 LOCATE 0.0.1

PRESET	PRESET [水平位置,垂直位置[,ファンクションコード(通常は使わない)]] 指定した位置にドットを消す	PRESET(50,45)
PSET	PRESET [水平位置,垂直位置[,ファンクションコード]] 指定した位置にドットを描く	PSET(10,10,2) PSET(0,0)
PUT@	PUT@(X,Y)-(x,y),配列[,条件] GET@で作成した配列を画面上の 任意の部分に出力する。その際に 条件として前の画面と演算が可能 PUT@A(X,Y)-(x,y),配列 ファンクションコードと共にGET@A したものを画面に出力する	PUT@(0,0)-(15,15),A%,G PUT@(5,5)-(40,40),B%,XOR PUT@A(0,0)-(159,99),A%
WIDTH	WIDTH [異形の桁数][,①画面の行数] 表示する文字数を決定する	WIDTH 80,25

特殊ステートメント

コマンド	書式・機能	使用例
BEEP	BEEP [スイッチ] 内蔵されたスピーカーから音を出す	BEEP BEEP 1
ERROR	ERROR コード エラー発生をシミュレートする ON ERROR GOTOと組み合わせて使用	ERROR 12
KEY	KEY キー番号, 定義文字列 ファンクションキーを定義する	KEY1,"WIDTH80,25"+CHR\$(13)
MOTOR	MOTOR [スイッチ] カセットテープレコーダの内蔵リレーを 制御する (モータ回転ON/OFF)	MOTOR 0
ON ERROR GOTO	ON ERROR GOTO 行番号 エラーが起きたときのジャンプを定義する	ON ERROR GOTO 100 ON ERROR GOTO 0
TROFF	TROFF トレースモードを解除する	TROFF
TRON	TRON トレースモードにはいる	TRON
RESUME	RESUME [0またはNEXTの行番号] エラー処理後、指定した行から プログラムを再開させる	RESUME RESUME 100 RESUME NEXT

ファンクション・コード

COLOR、LINE、PSET文などで使用するアトリビュート

白黒モード (CONSOLE ,,,0)		カラーモード (CONSOLE ,,,1)	
コード	機能	コード	機能
0	ノーマル	0	黒 ■
1	シークレット	1	青 ■
2	ブリンク	2	赤 ■
3	シークレット	3	紫 ■
4	リバーズ	4	緑 ■
5	リバーズシークレット	5	シアン ■
6	リバーズブリンク	6	黄 ■
7	リバーズシークレット	7	白 □

PRINT USINGのフォーマット指定

PRINT USING または PRINT#n USING、LPRINT USING などでは書式制御のためのフォーマット文を書くことができます。以下の記号を組み合わせで使います。

記号	定義	指定の例
#	数字の各文字の位置(桁指定)	###
. (ピリオド)	小数点の位置	##.###
+	数値の最初または最後に 正負の記号を付ける	+## ##.##+
-	数値が負の場合、最初または最後に マイナスを表示する	-#. # ##-
**	桁指定に実際の数値の桁が満たない ときに"*"を前付けする	**#.##
¥¥	数値の最初に円記号を表示	¥¥##, ##
**¥	数値の最初に円記号を表示し、なおかつ 桁指定に満たないときに、"*"を前に埋める。	**¥#, ###-
, (コンマ)	整数部分3桁ごとにコンマを表示	#, ### **¥#, ###, #-
^^^	指数表示にする。小数部分は1桁のみ	#.##^^^
/	文字列のプリントの場合、最初の1文字 だけを出力	/
&n個の空白&	n + 2 個の空白をとる	& &

数値関数

関数	機能	使用例
ABS(x)	xの絶対値	$Y = \text{ABS}(A+B)$
ATN(x)	xのアーктanジェント(xはラジアン)	<code>PRINT ATN(A/360*6.28)</code>
CDBL(x)	xを倍精度形式の数値に変換	$A = \text{CDBL}(Y)$
CINT(x)	xを整数へ変換	$A = \text{CINT}(B)$
COS(X)	xのコサイン (xはラジアン)	$A = \text{COS}(2.3)$
CSNG(x)	xを単精度型へ変換	$C = \text{CSNG}(X+2)$
EXP(x)	eのx乗	$B = \text{EXP}(C)$
FIX(x)	小数点以下を切り捨てた整数	$J = \text{FIX}(A/B)$
INT(x)	xを超えない最大の整数	$C = \text{INT}(X)$
LOG(x)	xの自然対数	$D = \text{LOG}(Y-2)$
RND(x)	乱数を与える x>0のとき 通常の乱数、 x<0のとき 乱数系列を更新、 X=0のとき 同じ乱数	$E = \text{RND}(1)$
SGN(x)	xの正負符号を与える x>0のとき $\text{SGN}(x)=1$ 、 x<0のとき $\text{SGN}(x)=-1$ 、 x=0のとき $\text{SGN}(x)=0$	$B = \text{SGN}(x+Y)$
SIN(x)	xのサイン (xはラジアン)	$S = \text{SIN}(A)$
SQR(x)	xの平方根	$C = \text{SQR}(D)$
TAN(x)	Xのタンジェント (xはラジアン)	$D = \text{TAN}(3.14)$

STRING

(sは文字列、x、y、nは数値を表す)

関数	機能	使用例
ASC(s)	文字列sの最初の文字のコードを与える	PRINT ASC(A\$)
CHR\$(x)	コードxに対応する文字を与える。ASCの逆	PRINT CHR\$(48)
HEX\$(x)	数値xの16進表現	H\$ = HEX\$(3)
INKEY\$	押されているキーの文字を返す(待ち無し)	X\$=INKEY\$
INPUT\$(x,[#]n)	ファイル(番号n)から与えられたx文字分の文字列 (miniでは#n無しのみ可)	x\$=INPUT\$(5,#1) A\$=INPUT\$(1)
INSTR([x,]s1,s2)	s1の中からs2と同じ文字列を位置xから探し、合致位置がs1中の左から何文字目に当たるかを与える	INSTR(A\$,"M") INSTR(3,A\$,B\$)
LEFT\$(s, x)	sの中の左からx文字までの文字列	B\$=LEFT\$(X\$,5)
LEN(s)	sが何文字あるかを与える	K=LEN("BASIC")
MID\$(s, x, y)	sの中の左からx文字目から長さyの文字列 sの中のX文字目からy文字分の文字列を代入	A\$=MID\$(X\$,5,10) MID\$(A\$,3,2)="TO"
OCT\$(s)	xの8進数を表現する文字列を与える	A\$=OCT\$(&HFF)
RIGHT\$(s, x)	sの中の右からx文字分の文字列を与える	C\$=RIGHT\$(X\$,A+1)
SPACE\$(x)	x個の空白を与える	A\$=S\$+SPACE\$(20)+"."
STR\$(x)	数値xを、それを表現する文字列に変換する	PRINT STR\$(350)+"円"
STRING\$(x,sまたはy)	sをx文字分繰り返した文字列を与える 0~255の数yをx回繰り返した文字列を与える	Z\$="Z"+STRING\$(2,"o") Y\$=STRING\$(100,42)
VAL(s)	文字列を数値に変換する	A=5+VAL("2")

ディスク関数

関数	機能	使用例
<i>CVD(s)</i>	8バイト文字列を倍精度の数値に変換	A= CVD(A\$)
<i>CVI(s)</i>	2バイトの文字列を整数の数へ変換	B=CVI(X\$)
<i>CVS(s)</i>	4バイトの文字列を単精度の数値へ変換	C=CVS(Y\$)
<i>DSKI\$(x,y,z)</i>	指定したセクタから文字列を読み出す	A\$=DSKI\$(H,I,J)
<i>EOF(n)</i>	シーケンシャルファイル (番号n) の終わりを示す	PRINT EOF(0)
<i>FPOS(n)</i>	ファイルの物理的セクタ番号	PRINT FPOS(1)
<i>LOC(n)</i>	ランダムファイルのとき、次のレコード番号、 シーケンシャルファイルの時は今まで読み書きした セクター数	PRINT LOC(2)
<i>LOF(n)</i>	ランダムファイルで今までの最大レコード番号	B=LOF(3)
<i>MKD\$(x)</i>	倍精度数値xを8バイトの文字列へ変換	PRINT MKD\$(123456789012)
<i>MKI\$(x)</i>	整数 x を2バイトの文字列へ変換	PRINT MKI\$(A)
<i>MKS\$(x)</i>	単精度xを4バイトの文字列へ変換	PRINT MKS\$(A)

その他の関数

関数	機能	使用例
CSRLIN	画面上のカーソルの縦位置	A=CSRLIN S
DATE\$	内蔵クロックによる日付	PRINT DATE\$
ERL	エラーが起きたときの行番号	E=ERL
ERR	エラー時のコードを与える	PRINT ERR
FRE(x)	ユーザー用のメモリエリアの空きを示す、xはダミー	PRINT FRE(0)
INP(x)	I/Oアドレスxポートから読み込んだデータの値	A=INP(32)
LPOS(x)	プリンタのプリントヘッドの位置、xはダミー	F=LPOS(3)
PEEK(x)	メモリのx番地の内容	A=PEEK(512)
POINT(x,y)	画面上の(x,y)の位置にドットがあるかどうかの判定	PRINT POINT(0,0)
POS(x)	画面上のカーソル位置、xはダミー	PRINT POS(0),CSRLIN
SPC(x)	PRINT文中で使用し、x個の空白を表す	PRINT SPC(5)
TAB(x)	PRINT文中で使い、左端からx文字までスキップする	PRINT TAB(20)
TIME\$	内蔵クロックによる時刻	PRINT TIME\$
USRn(x)	ユーザーが定義したn番目のマシン語ルーチンへ飛ぶ xは引数	USR2(A)
VARPTR(変数名)	変数の割り当てられているアドレスを与える	A=VARPTR(J)

エラーメッセージ

ERR エラーメッセージ コード	意味
1 Next without FOR	FOR-NEX が正しく対応していない。 (NEXT が多すぎる)
2 Syntax erroe	文法がまちがっている。プログラム中に規定のステートメント以外のものがある。
3 RETURN without GOSUB	GOSU-RETURN が正しく対応していない。 (RETURN だけがある)
4 Out of Data	READ 分で読まれるべきデータが DATA 文の中に用意されていない
5 Illegal function call	ステートメントの機能の呼び方が間違っている
6 Over flow	入力された数値や演算結果が許容されている範囲を外れている
7 Out of memory	メモリ容量が足りなくなった。(プログラムが長すぎる、配列が大きすぎるなど)
8 Undefined line number	必要とされるプログラム魚 (GOTO の飛び先など) が定義されていない
9 Subscript out ofrange	配列変数の添字が規定に範囲にない。
10 Redifmentioned array	同じ配列を再定義している (同じ DIM 文が複数回実行された)
11 Division by zero	0 (ゼロ) による割り算が実行された
12 Illegal direct	ダイレクト・ステートメントとして使えないコマンドが入力された
13 Type mismatch	代入文などで式の左右の型が一致していない (数値とストリングなど)

14	Out of string space	CLEAR 文なので指定したストリング変数用メモリエリアが足りなくなった
15	String too long	ストリング（引用符で囲まれた文字列）が長すぎる（256 文字をこえた）
16	String formura too comlex	文字式が複雑すぎる（カッコのネスティングレベルが多すぎる等）
17	Can't continue	CONT コマンドを入力しても続行できない（ポインタが）破壊されている）
18	Undefined usr function	参照されたユーザーの定義（DEF 分による）がなされていない
19	No RESUME	エラー処理後、プログラムの実行を再開できない
20	RESUME without error	エラーと RESUME が対応していない（エラーがないのに RESUME した）
21	Unprintable error	メッセージの定義されていないエラー
22	Missing operand	ステートメント中に必要とされるパラメーターなどが指定されていない
23	Line buffer overflow	BASIC が行を入力する際、1 行の長さが有効範囲を超えている
24	Position not on Screen	指定したカーソル位置などが画面の範囲外になっている
25	Bad File Data	ファイル上にあるデータの形式が間違っている
26	Disk BASIC Feature	ディスクが接続されていないときに、ディスク BASIC の命令を実行した
27	Communications Buffer overflow	周辺機器との入出力のためのバッファがオーバーフローした
28	Port not initialized	インターフェース用 LSI の機器設定がなされていない

29 Tape read ERROR カセットからの入力が行われていない（テープの読み誤りなど）

ディスク BASIC の場合のエラー（ディスクのない場合エラーは 26、21 となる）

50 Field overflow FILED 文で 128 バイト以上の文字を割り当てた

51 Internal error BASIC 内部でのエラー

52 Bad file number オープンされていないファイル番号をアクセスした

53 File not found LOAD、KILL、OPEN など存在しないファイルをアクセスした

54 Bad file mode シーケンシャルファイルで OPEN したファイルに対してランダムアクセスをした、またはその逆

55 File already open すでに開かれたファイルに対して OPEN や KILL を実行した

57 Disk I/O error ディスクにリード・ライト・エラーが生じ読み直しても修正できなくなった

58 File already exists NAME 文によって定義されたファイル名がすでに登録されている

61 Disk Full ディスク上のすべての場所を使い切った

62 Input past end ファイルのすべてのデータを読みだした後に INPUT 文が実行された

63 Bad record number PUT、GET 分でレコード番号が 0 ~ 32767 以内でない

64 Bad filename 不適当なファイル名を使った

66 Direct statement in file アスキーフォーマットでの LOAD 中にダイレクトステートメントがあった

67 Too many files 256 種類以上のファイルを作り出した

モニタコマンド

コマンド	機能	使用例
D	D [表示開始アドレス][, 表示終了アドレス] 指定された番地のメモリ内容を読み出し画面に表示する	DD000,D0FF
G	G [実行開始アドレス],[ブレークポイントアドレス# 1] [, ブレークポイントアドレス# 2] 指定された番地からユーザプログラムを実行する ブレークポイントアドレスでプログラムを中断、 さらに”G”を入力すると、ブレークポイントから再開	GD000,D088,D0A6
L	L カセットテープからマシンプログラムを読み出す	L
S	S [開始アドレス] 指定された番地のメモリにデータをセット（書き込む）	SE000
TM	TM メモリのテストを行う（パソコンミニでは不要）	
W	W セーブ開始アドレス、セーブ終了アドレス カセットテープにプログラムを書き出す	WD000,D0FF
^ B	コントロールキー + "B" BASICモードに戻る	

《 PasoconMiniPC8001 》

※ テープへの書き出し（Wコマンド）、読み込み（L コマンド）の実行前に MEDIA のページで「ファイル名」と「ファイル属性」を事前にセットしておくこと。

※ PC8001キーボード配列



※ PC8001キャラクターコード

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		D _E		0	@	P		p	■	┌		—	タ	ミ	≡	×
1	S _H	D ₁	:	1	A	Q	a	q	■	└	。	ア	チ	ム	⌈	円
2	S _X	D ₂		2	B	R	b	r	■	┌	「	イ	ツ	メ	⌋	年
3	E _X	D ₃	#	3	C	S	c	s	■	└	」	ウ	テ	モ	⌋	月
4	E _T	D ₄	\$	4	D	T	d	t	■	—	,	エ	ト	ヤ	▲	日
5	E _Q	N _K	%	5	E	U	e	u	■	—	・	オ	ナ	ユ	▲	時
6	A _K	S _N	&	6	F	V	f	v	■		ヲ	カ	ニ	ヨ	▲	分
7	B _L	E _B	▼	7	G	W	g	w	■		ア	キ	ヌ	ラ	▲	秒
8	B _S	C _N	(8	H	X	h	x		「	イ	ク	ネ	リ	♠	
9	H _T	E _M)	9	I	Y	i	y		」	ウ	ケ	ノ	ル	♥	
A	L _F	S _B	*	:	J	Z	j	z	■	┌	エ	コ	ハ	レ	♦	
B	H _M	E _C	+	;	K	[k		■	└	オ	サ	ヒ	ロ	♣	
C	C _L	→	,	<	L	¥	l	:	■	┌	ヤ	シ	フ	ワ	●	
D	C _R	←	—	=	M]	m		■	└	ユ	ス	ヘ	ン	○	
E	S _O	↑	.	>	N	^	n	~	■	┌	ヨ	セ	ホ	“	／	
F	S _I	↓	/	?	O	_	o		+	ノ	ツ	ソ	マ	°	＼	