

# パーソナル・コンピュータのベーシック (3)

小林 竜一

## 5. PETのベーシック

コモドール社から発売された PET というマイクロコンピュータはマイコン界のベストセラーの1つである。PET で使われるベーシックはオリジナルはマイクロソフト社の BASIC であると聞いている。

以下に PET の BASIC と標準 BASIC との差異を解説しよう。まず、NEXT 文が2つ続くとき、これを1つで書くことができる。たとえば、

```
NEXT V
NEXT W
```

の代りに、

```
NEXT V,W
```

と書くことができる。

また、NEXT とだけ書いた NEXT 文が許され、この文だけで適当な FOR 文 (もちろん上方の行の)へ帰る命令となる。

GET 文が使える。GET 文は INPUT 文や INPUT # 文と同じような働きであるが、それらと違う点は復帰改行のキーが押される必要がなく、押されたキーの情報を取込む点である。使用例を示すと、

```
GET A (キーボードから数字を入れるとき)
```

```
GET A$ (キーボードから文字を入れるとき)
```

などを書くことにより、キーボードから数字(Aのとき)または文字(A\$のとき)を1文字よむ。もしキーボードが押されないと A\$ の中には null (8ビット全部0)、A の中には0が入る。したがって、

```
10 GET A$: IF A$=" " THEN 10
```

と書いておくと、キーボードから何か押されて情報が A\$ に入ってくるまで待っているということになる。

PET のベーシックにも IF GOTO 文がある。INPUT 文の中で好きな入力促進記号(文)が書ける。たとえば、

```
5 INPUT "VALUE";V
```

という書き方により、Display に VALUE という文字が表示されてメモリーVへのデータ入力を促進していることを示せる。

ON GOTO 文が使用できる。ON GOTO 文については前回の解説を参照されたい。

ON GOSUB 文が使用できる。これも ALTAIR ベーシックと同じである。

POKE 文が使用できる。これも ALTAIR ベーシックと同じである。

PEEK 文はないが PEEK 関数がある。実例で示すと、

```
10 A=PEEK(I)
```

と書き、これで I で示される番地にある1バイトの情報をもってきてAという名のメモリーに代入する。

WAIT 文がある。実例で示そう。

```
30 WAIT I,J,K
```

と書くと、I で示される場所のメモリーの内容とKとを排他的ORをとり、それとJとANDを取ってそれが0でなくなるまで待つ。

```
0 ≤ I ≤ 65536
```

```
0 ≤ J ≤ 255
```

```
0 ≤ K ≤ 255
```

STOP キーはこの WAIT 文の働きを中断することはできない。この WAIT 文も ALTAIR ベーシックと同じである。標準 BASIC にはない関数としては上記の PEEK のほかに以下のものがある。

```
ASC("XYZ")
```

```
CHR$(N)
```

```
LEFT$(X$,A)
```

```
LEN(X$)
```

```
MID$(X$,A,B)
```

```
RIGHT$(X$,A)
```

```
STR$(A)
```

```
VAL(A$)
```

これらの関数は最後の2つを除いてすべて ALTAIR ベーシックとして紹介済みのものである。STR\$(A) は

A(数字)を文字列(ストリング)にして表わす関数である。VAL(A\$)は文字列 A\$ を数値に直す関数である。

また、IF-THEN 文において次の使用法ができる。

```
IF A>0 THEN B=0
IF A<0 THEN RETURN
```

また、PRINT 文の代わりに?を使用できる(これは最近ほとんどマイクロコンピュータで使用できる)。

? A, B (これで PRINT A, B と同じ)

ファイルを開いたり閉じたりする命令がある。まずファイルを開く命令である。

```
OPEN LF, D, SA, FN
```

と書く。LFは論理ファイル番号(ユーザーが好きな番号を割当てればよい)でDは装置の番号(これはPETの場合0がKB, 1がカセット No. 1, 2がカセット No. 2, 3がビデオスクリーンと決まっている。SAはセコンダリーアドレスで、これはIEEEバスの設計思想にもとづいた概念で以下のようなものである。

```
SA=0 普通の印刷
=1 フォーマット指定のある印刷
=2 CPU からフォーマットを指定
=3 1ページ当りの行数のセット
=4 拡張された警告メッセージの使用
=5 プログラブルな特性に対するバイト・データ
```

というようにPETのPrinterでは決まっている。要するに、このSAの情報でI/O機器へコントロール指令を送るのであり、この番号はマイクロコンピュータの装置によって違った約束がつけられている。同じPETでもテープ装置だとまた別の約束である。FNはファイル・ネームであって、ランダム記憶装置のときは2つ以上のファイル名があるから、その中の使用するファイル・ネームをここに書くのである。たとえば、

```
20 OPEN 1, 2, 1
```

とすると論理番号1番のファイルが開かれ(使用できる準備がととのったこと)、それは実際の装置としてはカセット装置2であって、それはテープに書き込む(これが上の命令の最後の1の意味)ために準備されたことを示す。なおFNがないのはファイル名なしであることを示す。このOPENのステートメントのあとでカセット装置2に書き込むときには、

```
700 PRINT #1, X
```

というような命令で書くことになる。そしてプログラムの最後(正確には計算などの作業の最後)で、

```
980 CLOSE 1
```

とすると論理ファイル1が閉じられる。

もう1つOPENの例を書くと、以下のようなになる。

```
10 OPEN 1, 1, 0 "VECTOR"
```

これで、論理ファイル番号1のファイルはカセットテープ装置1で、それは読むために開かれ、その読むファイルは“VECTOR”という名前のついたファイルであることを示している。

《注》磁気テープ装置1にVECTORというファイル名をつけた記録が記憶されているテープをセットしておくことが論理番号1番のファイルとされて読むことができるのである。

LOAD コマンドはOPEN コマンドの特殊のものと考えればよい。

```
LOAD, 名前, 装置番号
```

という形で書かれる。装置番号で指定される装置にかかっている磁気テープ(またはフロッピーなど記録媒体)から、指定された名前のプログラムをメモリーの所定のところ(これは磁気テープの中にすでに記入してある)に読み込む。

VERIFY というコマンドがある。これはSAVE コマンドのあとで使用され、SAVE したプログラムが正しいかどうかチェックするのに使用される。これはLOAD コマンドの変形と考えればよい。SAVE コマンドは以下のように使用される。

```
SAVE
SAVE "PROG 1"
SAVE "PROG 2", 2
SAVE "PROG 3", 2, 1
```

第1行のコマンドではBASICのプログラム(CPUの中に作られた)を磁気テープ装置1に唯しまう(プログラムの名前はつけない)。第2行のコマンドでは同じくBASICのプログラムを磁気テープ装置1に“PROG 1”という名前をつけてしまう。第3行のコマンドではBASICのプログラムを磁気テープ装置2に“PROG 2”という名前をつけてしまう。第4行のコマンドではBASICのプログラムを磁気テープ装置2に“PROG 3”という名前をつけてしまい、そのあとにテープ・ブロックの終りの印を書き込む。

VERIFY コマンドは以下のように使用する。

```
VERIFY "PROG 3"
```

ただしこれはSAVEのコマンド(第4行)のあとでこれを使用するのである。

DIM 文についての制限は要素が255個までしか許されないこと以外はない。次元は3次元でも4次元でもよい。たとえば、

```
DIM A(254), B(2, 80)
```

は正しいが、

```
DIM A(256), B(2, 128)
```

などは間違いである。なお、BASICでは添字は0からなので、 $B(2,80)$ と書くと要素でいうと、 $B(0,0)$ ,  $B(0,1)$ ,  $B(0,2)$ , ...,  $B(0,80)$ ,  $B(1,0)$ ,  $B(1,1)$ , ..., となる。これは一般のBASICでいえることである。またコモドールのBASICも行列演算の命令(MATのついた命令)はない。この点はTSS用の標準BASICと違う。

## 6. シャープ PC-1210/1211 の BASIC

シャープからポケット型の電卓でBASICでプログラミングできるものが発売されている。BASICの使える機械で最も安価なものであり、かつ持って歩けるものとして唯一のものである。

このBASICの文法について述べよう。まず、第1に変数名は英文字一文字である。A, B, ..., Zの26個までしか使えない。計算は10進法10桁(仮数部)で指数部は2桁別に取りられている。命令で標準のBASICと比較して欠けているものを以下に示す。

READ文, DATA文, DEF文, RESTORE文, DIM文 およびすべての行列演算命令(MAT, ATREAD文, MAT+文など)

また、関数としては欠けるものとして、

RND

があるが、逆に標準のBASICにはない次のものが加わっており、

ASN( $\sin^{-1}$ ), ACS( $\cos^{-1}$ ), LOG( $\log_{10}$ ), DMS(10進数 $\rightarrow$ 度分秒), DEG(度分秒 $\rightarrow$ 10進数)

また、自然対数はLN(x)とあらわす点が標準関数のLOGの表わし方と異なっている。また、標準関数でSQR(平方根)とかく所をPC-1210/1211 BASICでは $\sqrt{\quad}$ で表わす点が違っている。

DIM文がないけれども配列A( )というものが用意されており、たとえば、

```
110 FOR I=1 TO 10
```

```
120 A(I+26)=I^2 (Iは乗べきを示す)
```

```
130 NEXT I
```

というようなプログラムが書ける。つまり配列は必ず、

A( ) (1次元配列)

と考えればよい。ただし、

A(1)——A

A(2)——B

A(26)——Z

となっており、A(1)とAは同じメモリーであり、A(26)はZと同じメモリーである。そこで単独変数を使用したときは、そこを避けて配列A( )を使わないといけな

PC-1210/1211 BASICでは文字でラベルをつけることができ、これを使ってプログラムの任意の場所からプログラムをスタートすることができる。たとえば、

```
50 "A"
```

```
60 FOR I=0 TO 1 STEP 0.1
```

```
70 PRINT I, SIN(I)
```

```
80 NEXT I
```

と書いてあると、計算機のモードをDEFモードにしておく(そういうキーがある)と、SHIFTキー、Aキーを押すと、このラベルのところからプログラムの実行が開始される。また、USING文があって、これで書式を指定することができる。

```
10 A=-123.456
```

```
20 USING "###.##"
```

```
30 PRINT A
```

とかくと表示は、-123.4となる。

PAUSE文があって、文字を約0.85秒表示することができる。もちろん表示後次の命令の実行に入っていく。

またCLOAD, CSAVE文があって、プログラムをカセットテープからロードしたり、テープにセーブしたりすることができる。またCHAIN文があって、プログラムの終わりにこれを書けば、これで次のプログラムをカセットテープから読み出してきて、この新しいプログラムに制御を移すことができるので、長いプログラムでも短かく切って次々に実行させるようプログラムすることもできる。

さらにINPUT#文やPRINT#文があって、カセットテープからデータを計算機のデータメモリーの所に読み込んだり、データメモリーの内容をカセットテープに書き込んだりできる。なお、プログラムもデータもカセットに書き込むときラベルをつけて書き込み、読み出すときはラベルを使って読み出す。

それ以外にCLOAD?文があって、カセットテープからロードしたプログラムが正しいかどうかチェックできるし、DEBUGコマンドがあってデバックに便利になっているなど、ポケット・コンピュータとしては異色のものである。

## 参 考 文 献

[11] CBM 2001-16, 16N, 32, 32N PERSONAL COMPUTER USER MANUAL Commodore Business Machines 3330 Scott Blvd. Santa Clara, Ca. 95051

[12] シャープ ポケットコンピュータ ピタゴラス エルシーメイト PC-1210/1211 取扱説明書