

パーソナル・コンピュータのベーシック (2)

小林 竜一

3. TI ベーシック

これは TI 社のホーム・コンピュータ TI 99/4 用のベーシックで、ANSI規格のミニマル・ベーシックとコンパチブルであり、浮動小数点計算で有効桁数13桁である。初心者用の解説書が市販されている [7]。

プロンプティング記号 (入力促進記号) は > である。CALL CLEAR という文があって CRT ディスプレイの画面をクリアすることができる。変数の名前は英字で始まる15字以内の文字のつらなりである。PRINT 文のリストに算術式を書いてもよい。たとえば以下のようになる。

```
>PRINT W+T;T-W
```

LET 文において LET は略してもよい。CALL SOUND 文がある。周波数が 100~44000 ヘルツの音を作り出せる。

```
>CALL SOUND (1000, 440, 2)
```

これは1000分の1秒間、440ヘルツの音を2 (相当騒がしい) の強さで出す命令である。音の周波数を指定する部分を -1 ~ -8 の負の整数にすると雑音が出せる。

図形操作の命令がある。CALL VCHAR 文と CALL HCHAR 文である。

```
>CALL VCHAR(1, 10, 86, 50)
```

と書くと、第1行、第10列から始めて文字 V (番号86) を50個垂直方向に書く。一番下の行までいくと次の列の先頭にもどって V をまた垂直に書いてゆく。24行の CRT ディスプレイなので2列と2字 (3列目の最初に2つ) V が表示される。

```
>CALL HCHAR(17, 1, 72, 50)
```

と書くと、17行第1列の所から H (72番で指定) を50個水平方向に表示する。1行32字なのでほぼ2行 (第18行では18字) の H の並びが得られる。これらの命令を使っ

て文字を並べて図形を書くことができる。

TI のベーシックでは STOP 文は使わず、END 文で計算の終りとしている。GO TO 文は GOTO と書ける。◆ 上述の CALL SOUND 文や CALL VCHAR 文や CALL HCHAR 文で引数 (括弧の中) は数字でなく変数名を使用してもよい。

CALL COLOR 文という文があって画面に色々の色 (16色) で文字を書くことができる。詳細は略す [7]。

行番号は 1~32767 の範囲と限定されている。

TAB 関数がある。

```
PRINT TAB(10); "HELLO"
```

とすると画面の10桁目から HELLO と印刷が始められる。また RAN 関数に引数がない。

RANDOMIZE 文がある。これを使用すると乱数発生初期値がランダムとなるので、ほとんど2度と同じ乱数を生成することがない。

CALL KEY 文がある。たとえば、

```
CALL KEY(0, NOTE, STATUS)
```

と書く。すると押したキーに対応する文字コードが NOTE という記憶場所に入る。最初の引数は必ず0とし、最後の引数には状態を表わす数 (1, 0, -1) が入る。

```
STATUS = { 1 新たにキーを押したとき  
          -1 キーを押し続けているとき  
          0 キーを押していないとき
```

画面はふつうは明るい緑色をしているが、これを変えたいときは CALL SCREEN 文を使う。たとえば、

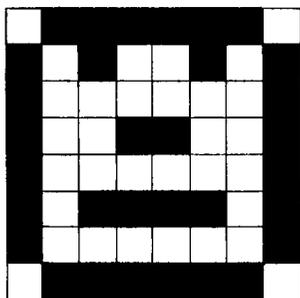
```
CALL SCREEN(11)
```

とすると暗い黄色 (11番) に変化する。

CALL CHAR 文があって 8x8 の区画をオンまたはオフにすることによって新しく自由な文字を作る。たとえば、

```
CALL CHAR(21, "7EA5819981BD817E")
```

と表わすと、21番という文字は16進法の数16個で表わされた 7EA5819981BD817E によって図のような形とな



7 E
A 5
81
99
81
BD
81
7 E

る。なお、NEW という文でプログラム領域もデータ領域も全部クリアすることができる。RUN, LISTのコマンドはあるが DATA文, READ 文, DEF 文がない。

以上の点が標準のベーシックに比べて TI ベーシックの異なる点である。要するにカラーグラフィックを自由にプログラムで作り出すことと音楽の演奏が自由にできる点が特長といえる。これを使っているいろいろとゲームなどを作ることもできようし、もちろん科学技術計算もできる。ただし行列演算の命令文はない。

なお文献[7]は初めて BASICを習う人にとっては実に親切な入門テキストと言えよう。わずかに数カ所、訳が原文に近くて日本語らしく見えない点があるが、概して訳もよくできている。TI の機械を買わない方でも本書を一読されることは有益であろう。

4. ALTAIR 8800 BASIC

米国 mits 社から出された ALTAIR 8800はマイクロコンピュータの始祖というべき傑作で、いまだにその規格 S-100パスがマイクロコンピュータの主流として残っている。

その BASIC は 4 K, 8 K, Extended, DISK の 4つのタイプがあり、それぞれ特色のあるものである。なかでも 4 Kは 4 Kバイトのメモリーの中に入る微小なプログラムにもかかわらず、すぐれたインタープリターである。これが発表されたときは他社の同等のプログラムを寄せつけなかったものである。これら4つのタイプを一括して紹介する。

まず行番号は 0 ~ 65529の範囲である。AUTO コマンドが使える。

AUTO 100, 10

とすれば、行番号が 100 番から始まり 10 番飛びで自動的につけられる。RENUM コマンドが使える。

RENUM 6000, 5000, 1000

5000番の行から始まるすべての行を 6000番から始め 1000番とびの番号を振り直す。もちろんこの行へ飛んでくる GO TO 文の行先番号や GOSUB 文の番号も直す。

RENUM 100, 20

とすると、すべてのプログラムを最初の行を 100 番として刻み 20 で番号をつけ直す。：で区切ることによって 1 行に何個も文を書くことができ、また、EXTENDED, DISK の 2 タイプでは 1 行を復帰改行で何行にもわけて書くこともできる。

100 FOR I=1 TO 10: REM DO THIS LOOP
といったように文のあとにその注釈をつけることができる。編集機能もついており、Del キーによって誤字を消して訂正することができる。1 行消すときは復帰改行のかわりに @ キーを押せばよい。NEW コマンドがある(既出)。

4 K と 8 K の版では数値は 7 桁までである。Extended と DISK 版では数値は 17 桁までである(有効数字)。変数名については 4 K バージョンのときは標準の BASIC と同じであるが 8 K 以上の版では最初の英数字 2 字で識別されるが名前の長さは長くてよくなる。ただし予約語 (LET, GO TO, DIM など……) をその中に含むと誤りとなる。たとえば、

RGOTO

は変数名として不可である。

変数名のうしろに以下のシンボルをつけるると以下に示すような変数の型になる。

シンボル	型
\$	文字型 (0~255字まで)
%	整数型
!	単精度 (有効数字 7 桁)
#	倍精度 (有効数字 16 桁)

また、詳細は省略するが変数の型宣言文も用意されており、これらは FORTRAN JIS 7000 水準のものと機能は同じである。

4 K 版ではべき乗演算はない。論理演算子として、NOT, AND, OR, XOR (排他的 OR), EQV, IMP (含意)

が使える(参考文献 [9] 16頁)。

IF _ THEN 文の他に IF (算術式) GOTO 文がある。IF THEN ELSE 文があり、構造化プログラミングができる強力な文で重ねて使える(文献 [9] 20頁)。

ON _ GOTO 文がある。K の中に 5 が入っているとき

ON K/3 GOTO 1000, 2000, 3000, 4000

という文は $5 \div 3 = 1.66\dots$ 、小数点以下を切捨てて 1 とし、行番号のリストの 1 番目の 1000 番の行番号のところへ飛んでいく。もし K/3 を計算して 0 か、5 以上ならこの文の次の文へゆく(4 K ベーシックを除く、以下の命令は全部 4 K では備っていない)。

ON _ GOSUB 文がある。たとえば K = 8 として、

ON K/3 GOSUB 100, 200, 300, 400, 500
と書くと、8÷3を計算し整数部分が2なので、行先リストの2番目、200番から始まるサブルーチンに飛んでいく。K/3が0または6以上なら次の文へゆく。

INPUT文で入力促進文字を自由に指定できる。たとえば、

```
INPUT "INPUT_「X」"; X
```

と書くと、INPUT_「X」_という入力促進の文がディスプレイに表示されて、KBからXの値が入れられるのが待たれる。WAIT文がある。

```
WAIT 20, 6
```

ポート20(20番目の入力回路、何かの周辺機器がついている)の入力として第1ビット、または第2ビットが1の信号が入ってくるまで待つ。ここで6を2進法で表わすと、

```
0 0 0 0 0 1 1 0
      ↑ ↑ ↑
      第0ビット
      第1ビット
      第2ビット
```

であるからである。また、

```
WAIT 10, 255, 7
```

とすると、ポート10の最高位から5桁までのビットのどれか1つが1となるか、または最下位の3ビットの中のどれかが0となるまでまつ。これは255を2進法で書くと、

```
1 1 1 1 1 1 1 1
  ↑           ↑
  最高位     最低位
```

であり、7は2進法で表わすと、

```
0 0 0 0 0 1 1 1
```

である。ポートからの入力が7と排他的ORされ、次にそれが255とANDされ、それが0でなくなると次の命令へ移るのである。

POKE命令は次のようなものである。

```
POKE I, J
```

ここでI, Jは整数の表式であって、Iの値で示される記憶場所にJを代入する。Jの値は0~255でないといけない。PEEK命令がその逆の命令である。

```
PEEK I
```

とすると、Iで示される番地からその内容を読み出して加算機(Acc)に入れる。OUT文がある。

```
OUT I, J
```

ポートIにJによって示される1バイトの情報を出力する。INP文は次のようなものである。

```
INP I
```

Iによって示されるポートから1バイトをAccに読み込

む。これらはマイコンにつけられた周辺機器や、実験データの取込みに便利に使われる。

その他に、LINE_「INPUT」文、SWAP文、CSAVE*文、CLOAD*文、ERASE文、TRON文、TROFF文、CONSOLE文、WIDTH文、ON_「ERROR」_GOTO文、RESUME文、PRINT_「USING」文などがあるが詳細は省略する。またフロッピーの制御文、ファイル制御文が用意されている。ファイルは順編成ファイルと、ランダムアクセスファイルの両方が利用でき、そのための命令群が用意されている。

また関数として次のものが加わっている。

ASC(X\$) X\$の中の文字列の最初の文字をASCIIコードで表わして与える。
CHR\$(I) ASCIIコードIの文字を作って与える。
LEFT\$(X\$, I) X\$の中の文字列の左方からのI個だけを取り出す。
RIGHT\$(X\$, I) 右方のI個だけを取り出す。
LEN(X\$) X\$の中の文字列の長さを返す。
MID\$(X\$, I) X\$の右方のI字を返す。
MID\$(X\$, I, J) X\$の中の文字列の第I番目の文字から始めてJ個の文字を返す。

この他に関数として、HFX\$, INSTR, CINT, CSNG, CDBL, INP, FRE, LPOS, OCT\$, POS, SPACE\$, SPC, STR\$, TAB, USR, VAL, VARPTRがある。これらの詳細は略すが、興味のある方は文献[9]を見られよ。

なお関数INTはガウスの記号と同じ働きをする。つまり、

```
INT(-2.5)=-3
```

である。この点はFORTRANのIFIXと違う。

また、ALTAIR BASICにはTSSバージョンもあり、これを使用すれば1台のマイコンに8台の端末で(8人のユーザーで)同時使用可能となるが、詳細は略す。

参考文献

- [7] TI社編 入門BASIC 齊藤梅朗訳 共立出版社刊
- [8] IEE社刊 ALTAIR BASIC ABC
- [9] MITS社 altair T.M. 8800 basic REFERENCE MANUAL (VERSION 4.0)
- [10] IEE社 Time Sharing BASIC Documentation (MITS社)