

# True BASICにおける特異性

榊原 正明

物質工学科

(1995年8月28日受理)

Peculiarities on True BASIC

by

Masaaki SAKAKIBARA

Department of Materials Science

(Received August 28, 1995)

Peculiarities on True BASIC are discussed in comparison with N88-BASIC. Arguments of internal functions and internal subroutines are found to be global. Window and view commands in N88-BASIC can be related to window and screen commands in True BASIC.

The simple program of conversion from N88-BASIC to True BASIC is proposed.

Key words : True BASIC N88-BASIC conversion

## True BASIC での 特異性

### 1. はじめに

われわれはNECのパソコン上でN88-BASIC言語を用いて教育用プログラムの作成を行ってきた<sup>1)</sup>。これは使っているということがあったからであるが、プログラムのデバッグや学生に対するプログラミングの指導には苦勞してきた。そこで数年前からMS-DOS上で動くBASIC言語の1つであるPC-9801用の日本語版 True BASIC ver. 3.0J を使い始め、学生の講義にも使用することにした。講義では文献[2]、[6]に基づいたテキストを自作して使用した。

1991年に True BASIC を土台として Full BASIC の国際規格が制定され、1993年には Full BASIC が日本工業規格(JIS)の新しい標準となった。True BASIC は Full BASIC に最も近いものとして BASIC の創始者である Kemeny と Kurtz が開発したものである。しかし True BASIC プログラムの本は、われわれの知り得たところ文献[2]から[11]だけである。しかも文献[2]から[5]までは True BASIC のマニュアルの訳本である。これはN88-BASICはマイクロソフト社のMicrosoft BASIC 5と同等と考えられ、マイクロソフト社がQuick BASIC を開発し、さらにQuick BASIC と同系統の Visual BASIC を Windows上で実行するのが主流になってきているからだと思われる。

Quick BASIC のほうが N88-BASIC より新しく、構造化言語となっているが、両者は非常に似通っている。(N88-BASIC と Quick BASIC を、それぞれN88、QB と省略して表す)。このため QB を中心に述べている本の中にも N88 については参考として触れているものがある<sup>1)</sup>。今回は、True BASIC とN88との違いを重点的に述べたい。これはTrue BASICを他のプログラム言語と比較することにより、その特徴(特異性)を捉え直し、新しく使う人の便をはかるためである。ただし、True BASIC の特異性としては、N88 とともに QB と異なる点を重点的にとりあげることとする。従ってSELECT CASE とかブロックIFなどはとりあげないこととする。

N88 から QB への変換ソフトは既に発表されており、真野<sup>2)</sup>はN88からQBへの変換ソフトを作成している。その

内容は CLEAR, CHILD, WIDTH, CLS, KINPUT, INPUT WAIT, CONSOLE, LOCATE, SCREEN, ON STOP, ON HELP, ON KEY GOSUB の12個のコマンドについて、対応するN88→QBの変換を行い、THEN、ELSE、RESTORE、RETURN、RESUME、GOTO、GOSUB文で行き先に指定された行番号を除いて、残りの行番号を取り去る。行ラベルをQBの書式に書き換えるというものである。一方、小池、森の変換ソフト<sup>3)</sup>は行番号の処理だけでなく、さらにLFコードがある場合はそれを取り去り不要のスペースを除くという方式を取っている。

これに倣って N88 からTrue BASICへの簡易変換ソフトを作成したのであわせてこれも紹介する。

### 2. True BASIC の 概要

True BASIC では QB と同じように総合環境にあり、メニューの選択はWindowライクな方法でマウスを使用するか、あるいはGRAPHキーを押してプルダウンメニューを開く方法がある。

1つの文は1行に割り当て、マルチステートメント文は使えない。しかし QB と違って継続行が使用できる。例えば2つの行を1行として扱うには、上の行の最後と下の行の最初に & を入力すればよい。

MAT 文を使って、配列の要素を一度に処理できる。例えば配列変数を A とすれば、MAT READ A で配列の読み込み、MAT PRINT A で配列の要素の表示を行うことができる。

DIM name\$(N) のようにカッコ内に変数は使えない、したがって配列の大きさを変更する場合は MAT REDIM name\$(N) として配列を再定義するしかない。N88 では再定義によって内容が失われるが、配列を再定義しても内容は失われない。単純変数Nと配列変数N()とは同時には使えない。

行番号はN88と同じで、実行順番を示し単なるラベルではない (QBではラベルである。)

THEN、ELSE、GOTO、GOSUB文にはラベルは使えず、行番号だけしか使えない。GOSUB文に使用したサブルーチンの変数はグローバルである。なお行番号を付けるには DO NUM、行番号を除くには DO UNNUM というコマンドが準備されている。

INPUT TIMEOUT <数値式> は N88 での INPUT WAIT文に

対応する。

注釈は"!"あるいはREMである。REMはキーワードなので文の先頭でしか使えない。!はキーワードではないので、どこにでも置くことができる。

N88とQBではRUNは文であり、メモリ上のプログラムと指定したプログラムを実行できるが、True BASICではコマンドでありプログラムの中では使用できない、またメモリ上のプログラムであるカレントプログラムだけを実行する。

KEY INPUTは何かキーを押せば真となる。GET KEY文は入力された文字のアスキーコードを、<数値変数>に代入する。キー入力を待つためにKEY INPUTと組み合わせる。N88、QBでのINKEY\$に対応させるには以下のようにする。

PRINT "スペースキーを押さない";

```
DO
  IF KEY INPUT THEN
    GET KEY X
    IF X=32 THEN EXIT DO
  END IF
LOOP
```

X=A=2とかPRINT A=2というような使い方ができない。これはN88、QBでの[真]は[-1]、[偽]は[0]ということを利用した使い方ができないということである。

部分文字列指定m\$(a,b)はm\$のa番目からb番目の文字を示す。N88、QBでのLEFT\$(A\$,3)はA\$[1:3]となり、RIGHT\$(A\$,3)はa\$[len(a\$)-3+1:len(a\$)]のようになる。文字型変数(定数)の加法は"&"である。

N88と文字が異なるものを以下に示す。

N88	True BASIC
ASC	ORD
STRING\$	REPEAT\$
KILL	UNSAVE
PAINT	FLOOD
LOAD	OLD
SYSTEM	BYE

## 2.1 厳密な用法

代入ではLETが必ず必要で、省略するためにはプログラ

ムの先頭にOPTION NOLET文を付けなくてはいけない(N88とQBでは省略可)。NEXT文の指標変数は省略できない(N88とQBでは省略可)。N88とQBでは配列の大きさが10以下であれば省略可能であるが、配列の宣言は省略できない。DIM A(10.3)という使用はできない。( )の中は整数でなければならぬ、しかもDIM A(INT(10.3))でも使えない。

INPUT文で文字列を表示する場合はPROMPT句は省略できない。

```
INPUT PROMPT"デ-タ数は":N
```

END文は1つのプログラムに1回だけ使用して省略できない(N88、QBでは省略できる)。

## 2.2 数値の処理

True BASICでは整数型と実数型との区別をしないので、DEFINTなどがなく、また型宣言子(%,&,!,#)もない。

Full BASICでは小数は十進法で扱われることになっているが、True BASICでは小数は二進法で扱っており、表示は8桁である。

仮数部は48桁、機械のエプシロンは $3.5527137 \times 10^{-1}$ °であり<sup>7)</sup>、文献[7]に示されているプログラムで調べると数値演算コプロセッサが内蔵してあればそれぞれ53桁、 $1.110223 \times 10^{-1}$ °となる。文献[9]では二進小数を求めるプログラムで計算精度を示している。

## 2.3 算術関数

N88でのA=10 MOD 3はA=MOD(10,3)のように使用する。

わり算の商を求める¥はなくてDIVIDEサブルーチンを使う。

自然対数LOG以外に、常用対数LOG10と2を底とする対数LOG2が用意されている。

OPTION ANGLE degrees文で角度の単位として度を使用でき、OPTION ANGLE radians文で角度の単位としてラジアンを使用できる。度をラジアンに変換する関数RAD( )、ラジアンを度に変換する関数DEG( )が用意されている。円周率PI関数があり、引数はない。四捨五入の関数(ROUND(x,n)小数点以下n+1位を四捨五入して小数点以下n位までを求める)も準備されている。転置行列、逆行列、単位行列、ベクトル内積、行列式などの関数が用意されている。MAT文を使って行列の和、差、積の計算を行う

ことができる。

### 3. 関数とサブルーチンにおける変数

QBの定義関数 DEF と FUNCTION 中の変数は、デフォルトでそれぞれグローバル、ローカルであり、また変数の引き渡しは前者は値により、後者は参照によってなされるという違いがある。True BASIC では、定義関数 DEF と FUNCTION は機能的に差は見られず、変数の引き渡しは値によりなされる。つまり仮引数(パラメータ)の値を変更しても実引数(引数)の値は変化しない。True BASIC での SUB構文は変数の引き渡しは参照によりなされる。つまり仮引数(パラメータ)の値を変更したら実引数(引数)の値も変化することになる。ところが CALL文の実引数(引数)を括弧でくくれば(例 CALL T((X))), 変数の引き渡しは値によりなされる(これは QB と同じである)。

END文より後の関数定義、サブルーチンはそれぞれ外部関数、外部サブルーチンである。外部関数(外部サブルーチン)とメインプログラムでは変数はローカルとなる。END文より前の関数定義、サブルーチンはそれぞれ内部関数、内部サブルーチンである。内部関数、内部サブルーチンとメインプログラムでは変数はグローバルとなる。今回、引数について調べると内部関数(内部サブルーチン)では仮引数(パラメータ)はローカルであるが、実引数(引数)は、他の変数と同じようにグローバルであることを見いだした。これは内部関数、内部サブルーチンで、実引数の値を変更すると、メインでの実引数の値も変更されていることになる。

なお内部関数であっても、関数定義が関数の引用より後の場合はDECLARE DEF文は必要である。

ローカル変数は自動変数であり、静的変数に設定することはできない。つまりローカル変数を含む関数、サブルーチンが呼び出されたときその変数は初期化されることになる。

行番号のあるプログラムでの GOSUB の注意事項は、END文の後にサブルーチンがくると"行番号にあやまりがあります"とエラー表示がでることである。この解決策としては END をプログラムの最後に持ってくる。なお GOSUB文には引数は使えない。

従って移植する場合もN88の変数がグローバルなので内

部関数、内部サブルーチンを使用したほうが容易であろう。文献[6][7][8][11]でもほとんど内部関数、内部サブルーチンを使用している。

QB では統合環境でサブルーチン・関数(QB ではサブプロシジャー)を作成し保存すると、自動的にENDの後にサブプロシジャーが置かれる。メインプログラムの先頭にDECLARE文が作られる。True BASICにはそのような機能はない。

## 4. 両面表示

### 4.1 テキスト表示

テキスト画面は WIDTH 80, 25 しかなく、しかも WIDTH 2文はない。N88 での CONSOLE文に対応する命令は存在しない。

SET CURSOR "OFF"文はカーソルを消し、"OFF"以外の文字式はカーソルを表示する。これらは N88、QB では LOCATE文の第3パラメータの値で行う。カーソルの位置を設定するのは SET CURSOR であり、N88 の LOCATE X, Y は QB と同じように SET CURSOR Y+1, X+1に対応する。

TABはPRINT文のなかでだけで使用でき、TAB(r, c) は SET CURSOR r, c と同じで、TAB の引数が1つの場合は桁数を意味し、行数は現在のカーソルのある行になる。

### 4.2 グラフィック

グラフィック画面の左下は(0,0)で右上が(1,1)となる。N88 との違いは縦方向が逆になる。

SET WINDOW文はグラフィック画面のウインドウ座標を設定する。これは N88 の WINDOWに対応している。N88 での 640×400 ドットのグラフィック(SCREEN 3)を移植する場合 SET WINDOW 0, 639, 399, 0のようにすればよい。下から上の方向にする場合は SET WINDOW 0, 639, 0, 399のようにする。

OPEN #<整数式>;SCREEN X1, X2, Y1, Y2 で画面での表示範囲を決める。N88 での VIEW に相当する。

N88 での VIEW(NVX1, NVY1)-(NVX2, NVY2) と WINDOW(NWX1, NWY1)-(NWX2, NWY2) と True BASIC の OPEN #1: SCREEN TSX1, TSX2, TSY1, TSY2 と SET WINDOW TWX1, TWX2, TWY1, TWY2 とに対応させると、以下のような関係が得られる。但しグラフィック画面は 640×400 ドットとする。

TSX1=NVX1/639	TSX2=NVX2/639
TSY1=1-NVY2/399	TSY2=1-NVY1/399
TWX1=NWX1	TWX2=NWX2
TWY1=NWY2	TWY2=NWY1

他のグラフィック命令の N88 との対応を以下に示す。

N88	True BASIC
PSET(X, Y)	PLOT X, Y
LINE(X1, Y1)-(X2, Y2)	PLOT X1, Y1; X2, Y2
LINE(X1, Y1)-(X2, Y2), .B	BOX LINES X1, X2, Y1, Y2
LINE(X1, Y1)-(X2, Y2), .BF	BOX AREA X1, X2, Y1, Y2

#### 4.3 両面モード

グラフィックモードにするためにはPLOTなどのグラフィック命令を実行するか、SET WINDOW 文、SET MODE "GRAPHICS"を実行すればよい。テキストモードからグラフィックモードになった場合に文字は消去されない。グラフィックモードでSET MODE "GRAPHICS"を実行すると図形は消去される。

テキストモードにするためには、SET MODE "TEXT"を実行すればよい。編集画面からであれば OPEN #1:SCREEN文、SET CURSOR文、CLEAR文を実行してもテキストモードにすることができる。

SET MODE "TEXT" の実行でグラフィックモードからテキストモードにすると、すでに描かれていた図形は消去される。

画面モードの切り替えの取扱いは、あまり厳格でないことがわかる。

#### 4.4 COLOR文

SET COLOR文はフォアグラウンドカラー(表示色)を設定する。

コマンド画面では色指定をしても色の変化は無い。テキストモードでは色指定をしてもグラフィックの色は変化しない。テキストモードでグラフィックを描くとグラフィックモードになり、その後で色指定をするとグラフィックの色は変化する。

#### 4.5 混合色の設定 SET COLOR MIX (c) r, g, b

cは作った色につける番号で、r, g, bは赤、緑、青の輝

度で0から1の間の値で指定する。0は点灯しない(輝度0%)、1は輝度100%の状態を示す。64色の場合は0, 1/3, 2/3, 1の4種に近似される。

SET MODE、SET WINDOW を実行するとSET COLOR MIX文の設定が無視される。

PLOT文によるグラフィック画面への変更とSET MODE "GRAPHICS"によるグラフィック画面への変更とは違う、後者はSET COLOR MIXを初期設定にし、文字の表示の色もとに戻っている。SET MODE "TEXT" ではSET COLOR MIXを初期設定にして、文字の表示の色は本来に戻っている。PLOT文の色は変更したままである。

N88でと同じ色番号に設定するには以下のようにする。このSET COLOR MIX文が設定してあれば、N88での例えばCOLOR 4 は SET COLOR 4 とすればよい。

SET COLOR MIX(1) 0, 0, 1	!明るい青
SET COLOR MIX(2) 1, 0, 0	!明るい赤
SET COLOR MIX(3) 1, 0, 1	!明るい紫
SET COLOR MIX(4) 0, 1, 0	!明るい緑
SET COLOR MIX(5) 0, 1, 1	!明るい水色
SET COLOR MIX(6) 1, 1, 0	!明るい黄色
SET COLOR MIX(7) 1, 1, 1	!明るい白
!	
SET COLOR MIX(8) 1/3, 1/3, 1/3	!灰色
SET COLOR MIX(9) 0, 0, 2/3	!青
SET COLOR MIX(10) 2/3, 0, 0	!赤
SET COLOR MIX(11) 2/3, 0, 2/3	!紫
SET COLOR MIX(12) 0, 2/3, 0	!緑
SET COLOR MIX(13) 0, 2/3, 2/3	!水色
SET COLOR MIX(14) 2/3, 2/3, 0	!黄色
SET COLOR MIX(15) 2/3, 2/3, 2/3	!白

## 5. ファイル

### 5.1 ファイル名の変更

RENAME ファイル名 はカレントファイル(現在表示されているファイル)の名前を変更する。拡張子を指定しなければ、TRUがつき、指定すればTRU、TRC 以外の拡張子にすることができる。

実行例 REN TEST.TXT

RENAME OLDNAME, NEWNAME はディスクに保存されているファイルの名を変更できる。ドライブ名も指定できるので、カレントドライブ以外にあるファイルの名前も変更できる。

OLD TEST を実行して ファイル TEST をオープンする、続いて OLD DEMO を実行して ファイル DEMO をオープンする。ファイル TEST はオープンされているが、カレントファイルではない。

ここで REN TEST, SIKEN と実行すれば、ディスクに保存されているファイルの名が変更される。つまりディスク上には TEST というファイルは存在しなくなる。しかし SWITCH TEST と実行すれば、ファイル TEST がカレントファイルになる。つまりファイル TEST はメモリ上にだけあることになる。

SET NAME コマンドはカレントファイルだけの名前を変更できる。拡張子の指定をしなれば、TRUがつき、指定すればTRU、TRC 以外の拡張子にすることができる。

```
SET NAME "FILES.TXT"
```

RENAMEとSET NAMEはコマンドであり、プログラムの中では使えない。

## 5.2 ファイルの読み書き

テキストファイルの上書きができないので、ERASE文でファイルの内容をメモリから消去する必要がある。また APPENDモードがないので RESET #1:END でポインタを最後に移動させてから書き込みを行う。N88とQBのようなEOF関数がないので DO WHILE MORE #1 という使い方をする。

ランダムファイルについて、N88 では文字列しか書き込めないが、True BASICでは文字列と数値を書き込めるし、読みだしもできる。WRITE文でセミコロンは使えない。

## 6. 簡易変換プログラム

N88 のプログラムをTrue BASIC用に変換するものである。その機能は注釈行の ' か REM を ! に変換し、コロンによるマルチステートメント文に対して改行を行うものである。変換するのにこれだけでは当然不十分である。まず問題となるのはGOTO、GOSUB文の行番号とラベルの処理である。True BASICでは、ラベルは使用できず、

行番号を一部だけ残すこともできない。つまり行番号とラベルについては真野<sup>12)</sup>と小池、森<sup>13)</sup>のやりかたは使えないということである。今回の変換プログラムには行番号とラベルの処理は含めなかった。

```
OPTION NOLET
```

```
INPUT prompt "read file name?":r_file$
```

```
INPUT prompt "write file name?":w_file$
```

```
OPEN #1:name r_file$,create newold.org text,access
  outin
```

```
OPEN #2:name w_file$,create newold.org text,access
  outin
```

```
ERASE #2 !すでにファイルが存在している場合のために
```

```
DO while more #1
```

```
  LINE INPUT #1:line$
```

```
  l=1cn(line$)
```

```
  ! ' を ! に変更
```

```
  s=l
```

```
DO
```

```
  p=pos(line$,"'",s)
```

```
  IF p=0 then EXIT DO
```

```
  LINE$(p:p)="!"
```

```
  s=p+1
```

```
LOOP
```

```
! REM を ! に変更
```

```
s=l
```

```
DO
```

```
  p=pos(line$,"REM",s)
```

```
  IF p=0 then EXIT DO
```

```
  LINE$(p:p+2)="!"
```

```
  s=p+1
```

```
  l=l-2
```

```
LOOP
```

```
! コロンで改行
```

```
s=l
```

```
DO
```

```

p=pos(line$,":",s)
IF p=0 then EXIT DO
out$=line${s:p-1}
s=p+1
PRINT #2:out$
LOOP

out$=line${s:1}
PRINT #2:out$
LOOP

CLOSE #1
CLOSE #2
END

```

## 7. まとめ

True BASIC の特徴を、主に N88-BASIC との比較から考察した。True BASIC の特異性は、その厳密な用法、行列演算機能、グラフィック命令、ファイルの取扱いにあることがわかる。

内部サブルーチン、内部関数では変数はグローバルであり、引数のうち実引数だけがグローバルであることがわかった。N88-BASIC ではすべての変数はグローバルであることから、N88-BASIC からの移植は、内部サブルーチン、内部関数だけとなることがわかる。外部関数、外部サブルーチンを用いて移植を行うことは非常に手間がかかるが、引数が重要となるのでこのような移植を自習させることで教育効果は向上すると思われる。

画面モードの切り替えと図形と文字の表示の関係、混合色設定との関係も明らかになった。変換プログラム作成には堀尾忠教君の協力を得た。

True BASICを使い始めてその基礎的な部分をようやく理解したが、今後はそれをどのように初心者に教えるのかという問題が残されている。

## 参考文献

- 2) J. G. Kemeny, T. E. Kurtz(フエムテック訳): True BASIC リファレンスマニュアル 啓学出版 1991.
  - 3) R. T. Bogart, B. Elliot(フエムテック訳): True BASIC プログラミング徹底入門 啓学出版 1992.
  - 4) J. G. Kemeny, T. E. Kurtz(フエムテック訳): True BASIC プログラミング徹底入門 活用編 啓学出版 1993.
  - 5) J. G. Kemeny, T. E. Kurtz(フエムテック訳): True BASIC プログラミング徹底入門 演習編 啓学出版 1994.
  - 6) 田中廣: True BASIC プログラミング500 題 日刊工業新聞社 1992.
  - 7) 森口繁一、伊理正夫、武市正人編: Full BASIC による算法通論 東京大学出版 1991.
  - 8) 森口繁一: Full BASIC 統計プログラム集 日本規格協会 1992.
  - 9) 佐藤公作、白石和夫、高橋雅信: 数学とコンピュータ 1 森北出版 1994.
  - 10) 佐藤公作、白石和夫、高橋雅信: 数学とコンピュータ 2 森北出版 1995.
  - 11) 和田正信: コンピュータ物理の功罪 共立出版 1994.
  - 12) 真野倅一: JAPC, Vol. 12, No. 3, p. 333 (1990).
  - 13) 小池慎一、森博: Quick Basic 中級プログラミング 技術評論社 1990.
  - 14) 水島賢太郎: プログラミング演習 BASIC ソフトバンク 1995.
- 1) 樽原正明, 松本法子: 鳥取大学教養部紀要, Vol. 28, P. 185 (1994).

