Enhanced BASIC for Models I & III

ENHBAS

Lawrence I Charters, Bremerton, Washington

A review of the Cornsoft Group's ENHBAS, in which the reviewer also includes three short programs written using this BASIC enhancement.

Most reviews are just that, but this one departs slightly and gives not only a review of ENHBAS, from the Cornsoft Group, but three sample programs written in ENHBAS as well. If you already have this remarkable package, key in the programs and enjoy them. If you don't have ENHBAS, read on and find out more about it.

ENHBAS is an extension of TRS-80 BASIC, and is worth every penny of its modest cost. Once loaded, you have the ability to use all normal TRS-80 commands, plus such new and powerful additions as WHILE/WEND structured loops, and elegant little flourishes such as the built-in PI constant. Lovers of the exotic will relish commands such as EXEC, which allows you to execute, as if it were a program line, any string expression.

Unlike most BASIC extensions and utilities, ENHBAS is available for Model I and Model III, cassette and disk systems. (An expanded version is available for the Model II). As long as you have at least 16K of RAM and Level II BASIC, there is a version of ENHBAS you can use, and it will allow you to run virtually any ENHBAS program. When this reviewer sent off a check for ENHBAS, it arrived promptly, packed in a sturdy box and padded by sections of the "Indianapolis Star" - a much better paper than the local paper. (I might even subscribe...)

To give you some idea of what you can do with ENHBAS (pronounced "EN - BASE"), note Spectacular Short Program 1 (Listing 1). Written as a single line, it requires next to no memory; the ENHBAS "SIZE" command reports it to be just 160 bytes long. Yet this tiny program prints, in regular, orderly columns, the full binary and hexadecimal equivalents of all decimal numbers between 1 and 500. If this doesn't impress you, it also prints a caption across the top of the video, labels the columns, draws a bar to separate the labels from the columns, and scroll-protects the top three lines of the video. The binary, decimal and hexadecimal numbers neatly scroll up the screen and disappear under the bar, giving a very professional appearance. Finally, as a parting gesture, the program disables the scroll protection. All of this using just three ENHBAS features: SCROLL, BIN$ and HEX$.

```
10      SCROLL=SET:A$=STRING$(32,42):B$="
        Binary           Dec      Hex":PRI
        NTTAB(14)"BINARY / DECIMAL / HEX
        CONVERSIONS":PRINTB$"       "B$:PRIN
        TA$A$:SCROLL=3:FORI=1TO500:PRINTB
        IN$(I);"   "I;"    "HEX$(I),:NEXT:SC
        ROLL=RESET
```

Listing 1

In answer to "what else can it do?" note Spectacular Short Program 2 (listing 2). Again the program is one line long, requiring a mere 181 bytes. This program (1) sets up a string array of 448 one-byte strings (448 fit nicely on the screen); (2) assigns a random character to every element of the array; (3) prints the array (so you can see that it is truly random); (4) waits patiently for you to press a key; (5) sorts the array in ascending order; and (6) prints out the sorted array. In spite of the complexity of the task, the sort is blindingly fast - it takes just six seconds to sort all 448 elements. A program designed to do the same thing, written in standard TRS-80 BASIC, would be much longer, and could very well take *hours* to finish sorting. Note too, such details as the WINKEY$ command. Even the most dedicated Microsoft purist will admit that:

```
10 Z$=WINKEY$:PRINT"YOU TOUCHED
ME!"
```

is much easier to write and understand than

```
10 Z$=INKEY$
20 IF Z$=""GOTO 10
30 PRINT"YOU TOUCHED ME!"
```

Both of these program segments perform identically. Which would you use?

Listing 2

```
10      CLS:CLEAR500:SCLEAR:DIMA$(447):PR
        INT"*** INITIALIZING STRING ARRAY
         ***":FORI=0TO447:A$(I)=CHR$(RND(
        95)+32):PRINTA$(I)" ";:NEXT:PRINT
        "PUSH ANY KEY TO START";:Z$=WINKE
        Y$:CLS:PRINT"START":KEYA$:SORT:PR
        INT"DONE":FORI=0TO447:PRINTA$(I)"
         ";:NEXT
```

You may have noticed that both listing 1 and 2 are very neatly formatted. ENHBAS does this automatically. Listings on both the video and the printer reserve the first six columns for line numbers, resulting in uniform columns of numbers and very readable program lines. ENHBAS also allows you to set the maximum printing width of your printer. For example, if you have an 80 column printer and want your program listing to look identical to what appears on your 64 column screen, simply type "CLM=64" and your printer will think it has only 64 columns. If this isn't enough, ENHBAS automatically sets the maximum page length of your printer at sixty lines, with a three line margin at the top and bottom. Never again will you have to worry about LLISTing your program on the page perforations! If you don't want a sixty line page, you may change the default by setting PAGE to some other value. Finally, if you don't have a printer (or your printer is turned off) the computer will not hang up if asked to LPRINT or LLIST. Instead, after a few seconds, the message "PRINTER NOT READY" will appear and you can continue on.

Other ENHBAS features include: automatic lower case; a user defined cursor; the ability to "RENEW" programs; one-letter edit commands (D, E, I, L, P, R and F replace DELETE, EDIT, AUTO, LIST, LIST-, RUN, and a new command, FIND); short-entry commands - <control> <SHIFT>S, for example, prints "STRING$("; and the ability to print, directly from the keyboard, all the arrow keys, plus the brackets, Yen and Pound signs, and other special characters available with a lower case modification. With a speaker/amplifier connected, ENHBAS provides an audible key "click"; a two-note error tone; and a one-note <BREAK> tone. Simply writing PRINT CHR$(7) or pushing <control> G will activate the "bell" signal, an eight-note rendition of Westminster Chimes.

If you haven't purchased a disk yet, cassette versions of ENHBAS include the Disk BASIC feature MID$, INSTR, LINEINPUT, and octal and hexadecimal constants. Machine language programmers will appreciate the ENHBAS commands WPEEK, WPOKE, and CALL. The first two commands are sixteen bit versions of PEEK and POKE - no more messing around trying to figure out the backward Z-80 instructions and addresses. CALL is similar to USR, except that addresses of machine language routines do not need to be predefined, and arguments may be passed both ways.

In fact, ENHBAS has something for everyone. Several extensions of BASIC are available, such as Microsoft's Level III, Apparat's Bionic Basic, and Modular Software's NewBasic. None offer the power and flexibility of ENHBAS, and ENHBAS is the only such extension with versions for cassette and diskette, Model I and III. ENHBAS also comes with extremely well written documentation, the only problem area being a somewhat fuzzy explanation of the PLAY (play music) command. You won't have to flip through cryptic explanations, poorly printed on off-sized, off-color paper - ENHBAS documentation is written in English, and enclosed in an $8^1/_2$ by 11 inch ring binder. The documentation even includes both a table of contents and an index. Remarkable.

For this review, a disk version of ENHBAS was tested on a 48K Model I using TRSDOS, NEWDOS+, and NEWDOS80. The original version tested, ENHBAS 1.7, worked quite well, with just one trivial bug noted (one command lacked a feature the documentation said it should have). Another minor problem was the use of the down-arrow as a control key for shorthand entry of BASIC keywords - irritating when using the NEWDOS or NEWDOS80 single-line listing functions.

ENHBAS 2.2, the latest version received, added several new features, and fixed the bug noted earlier. The short entry procedure has been changed, allowing all the arrow keys to be used. This latest version also "tokenizes" ENHBAS, resulting in faster operation than 1.7. Probably the most useful feature, though, is the new ZSTEP option. When ZSTEP is set to 1, the programmer can single-step through the program. A pause occurs after each statement is executed, and execution continues only after a key, any key, is pressed. Coupling TRON with ZSTEP makes debugging ridiculously easy.

Some users might find certain aspects of ENHBAS annoying. ENHBAS loads at the top of memory, occupying around 5K that it guards most jealously. Machine language programmers who like to use this area for various routines may find some conflicts, but the WPEEK, WPOKE and CALL command should permit easy adjustments. ENHBAS is also a bit slow. The program "FOR I=0 TO 10000:NEXT" will take 24 seconds to execute under regular disk BASIC, but 29 seconds under ENHBAS 2.2. This, too, is a minor difficulty, since sorts and graphics are the most time-sensitive BASIC operations, and ENHBAS offers routines which *drastically* speed up both types of programs.

The only "defect" noted in ENHBAS 2.2 was the lack of a shorthand entry command for the zero key (it should yield "INSTR", but doesn't). Programs written under older versions of ENHBAS will also require some fairly simple editing, since the "tokenizing" of reserved words in version 2.0 and later releases will create conflicts resulting in syntax errors. If you have an older version of ENHBAS and wish to receive the latest version, the Cornsoft Group will update your disk for around $3.00 (send them the disk). For the very reasonable price of $15.00, they will update your disk and send you the second edition of the manual. The Cornsoft Group appears firmly committed to product support and improvement.

The final sample program (Listing 3) illustrates several ENHBAS features. Borrowing a "turtle graphics" concept, an integer array is used to hold the graphic image of "A Famous Android".

Using the DRAW command, an outline of the android (defined in the array L) is rapidly drawn, rotated at 45 degree intervals, moved all around the screen, inverted from black to white and white to black (using the INVERT command), and expanded and contracted. Two boxes are drawn (with the PLOT command), forming an identification card for the android, and finally the entire display is scrolled off the left edge of the screen (using the LEFT command). While all of this is going on, music vaguely resembling part of Beethoven's Ninth Symphony is played (the command PLAY uses the string S$ for this purpose), together with some other incidental music.

Then, using the RDGOTO and JNAME features, the contents of the array defining the android's outline are replaced with another set of values defining a solid representation of the famous android. In other words, two separate and independent sets of data occupy the same array (L) at different, program-specified times. More androids are drawn, moved and rotated, and more music is played.

Eventually the process is repeated. Note too, that the RDGOTO and JNAME features allow the string S$ to contain, at various times, three different music selections. These features make for a very fast, compact (2963 bytes), and versatile program.

If you do any programming at all, you owe it to yourself to try ENHBAS. The Cornsoft Group also offers an integer BASIC compiler at an attractive price, and is working on a floating point compiler. "Missile Attack", one of Adventure International's latest hits, is another Cornsoft product. Any way you look at it, ENHBAS is a bargain.

```
Listing 3
10      CLS:CLEAR700:DEFINTI-L:DIML(54):P
        OKE16427,1:POKE16409,1
20      PRINT@340,"L APWRROEGNRCAEM MCIHN
        AGR TBEYR S"
30      PRINT@960,CHR$(23):FORI=0TO9:LEFT
        :FORJ=0TO300:NEXTJ,I
40      RDGTO"MUSIC":S$="":GOSUB280:RDGTO
        "ANDY1":FORI=0TO54:READL(I):NEXT
50      IY=12:FORIX=124TO68STEP-14:FORI=1
        TO8:POKE16426,I:CLS:GOSUB270:NEXT
        :IY=IY+4:NEXT
60      PRINT@26,"THE ADVENTURE OF";:PRIN
        T@90,"SECRET 'BOT .037";:GOSUB300
```

```
70    IX=68:IY=28:FORI=2TO11:POKE16427,
      I:GOSUB270:CLS:NEXT
80    CLS:FORI=11TO1STEP-1:POKE16427,I:
      GOSUB270:NEXT
90    RDGTO"MUSIC2":S$="":GOSUB280:FORI
      =0TO11:INVERT:NEXT:GOSUB300
100   CLS:IY=28:FORIX=68TO20STEP-4:GOSU
      B270:IY=IY-1:NEXT
110   CLS:IX=20:IY=16:GOSUB270:PLOTSB,3
      ,2TO34,34:PLOTSB,0,0TO127,47
120   PRINT@88,"SECRETARIAT FOR BUREAUC
      RACY";:PRINT@157,"SECURITY DIVISI
      ON";:PRINT@215,"ANDROID DEPARTMEN
      T, SEC. .037";:PRINT@274,STRING$(
      44,42);
130   PRINT@338,"NAME: Android Nim
       AGE: Two Years";:PRINT@402,"RANK
      : Special Agent    STATUS: Active
      ";:PRINT@466,"WEAPON: Mk 46 Laser
       Blaster, Type III,";:PRINT@538,"
      or Light Saber, 4 Megawatt";
140   PRINT@594,"CLEARANCE: Utmost Top
      Secret, RS-232C";:PRINT@658,"MISS
      ION: Seek out and destroy checksu
      m";:PRINT@731,"errors, bad loads,
       and null sorts.";
150   PRINT@770,"DESCRIPTION-- Wt.: 700
       kg.  Ht.: 200 cm.  Eyes: Two.";:
      PRINT@834,"Hair: None.  Agent Nim
       is fluent in BASIC, Assembler,";
      :PRINT@898,"binary, octal, decima
      l, and hexadecimal.  No oiling re
      quired.";
160   FORI=0TO5:GOSUB300:NEXT
170   RDGTO"ANDY2":FORI=0TO54:READL(I):
      NEXT
180   FORI=0TO63:LEFT:NEXT:POKE16426,4:
      IY=5:FOR IX=15TO127STEP16:GOSUB27
      0:NEXT
190   POKE16426,0:IY=42:FORIX=113TO0STE
      P-16:GOSUB270:NEXT:RDGTO"MUSIC3":
      S$="":GOSUB280
200   IX=114:IY=12:FORI=0TO10:IX=IX-9:I
      Y=IY+2:CLS:GOSUB270:NEXT
210   FORI=5TO8:POKE16426,I:CLS:GOSUB27
      0:NEXT
220   FORI=0TO5:IY=IY-2:CLS:GOSUB270:NE
      XT
230   FORI=0TO5:IX=IX+14:GOSUB270:NEXT:
      INVERT:FORI=0TO3:GOSUB300:NEXT:FO
      RJ=0TO63:LEFT:NEXT
240   CLS:IX=60:IY=12:GOSUB270:POKE1642
      6,2:IX=80:IY=16:GOSUB270
250   IX=74:IY=33:POKE16426,4:GOSUB270:
      IX=54:IY=29:POKE16426,6:GOSUB270
260   PRINTCHR$(7);:RDGTO"MUSIC":S$="":
      GOSUB280:GOTO40
270   DRAWSET@IX,IYUSINGL:RETURN
280   FORI=0TO120:READJ,K:IFJ=0RETURN
290   K2=INT(K/256):K3=K-256*K2:S$=S$+C
      HR$(J)+CHR$(K3)+CHR$(K2):NEXT:RET
      URN
300   PLAYS$:RETURN
310   JNAME"ANDY1"
320   DATA 1307,513,1027,257,519,513,51
      9,513,519,1025,263,1029,1287,1025
      ,263,1029,519,517,519,517,519,261
      ,1027,773,1031,517,519
330   DATA 517,519,261,515,517,515,517,
      1027,1285,1031,261,1283,2049,2307
      ,2053,1283,257,1031,3329,771,2053
      ,771,257,519,2049,1031,513,0
340   JNAME"ANDY2"
350   DATA 257,258,517,1045,259,529,513
      ,273,513,273,257,258,261,277,261,
      789,261,533,261,259,2817,529,259
360   DATA 517,277,1541,259,1025,273,51
      3,259,261,277,1285,259,1281,273,7
      69,276,3077,259,1553,257,273,513
370   DATA 275,261,277,1285,533,261,275
      ,529,257,0
380   JNAME"MUSIC"
390   DATA 111,200,111,200,105,210,94,2
      40,94,240,105,210,111,200,125,190
      ,140,160,140,160,125,190,111,200,
      111,300,125,90,125,260
400   DATA 111,200,111,200,105,210,94,2
      40,94,240,105,210,111,200,125,190
      ,140,160,140,160,125,190,111,200,
      125,300,140,70,140,220
410   DATA 125,190,125,190,111,200,140,
      160,125,190,111,80,105,120,111,18
      0,140,200,125,190,111,80,105,120,
      111,180,125,190,140,160,125,190,1
      88,200
420   DATA 111,200,111,200,105,210,94,2
      40,94,240,105,210,111,200,125,190
      ,140,160,140,160,125,190,111,200,
      125,300,140,70,140,400,0,0
430   JNAME"MUSIC2"
440   DATA 168,100,168,100,149,120,168,
      100,140,140,149,120,125,160,140,1
      40,0,0
450   JNAME"MUSIC3"
460   DATA 188,180,188,180,168,190,168,
      190,158,200,158,200,168,190,168,1
      90,0,0
```

■