
Post Office Box 1329
Redondo Beach, CA 90278

NEWS80S



Issue Number 6

For HP Series 80 Users

\$ 3.00

* ANATOMY OF A BINARY — A technical Primer
The start of a series on machine language for the Series 80 .

* FILE-IT VS. CARDFILE: Why pay \$100 for something 'free'?

* MENU'D Part One - A program that writes program routines

ANATOMY OF A BINARY by Dale Flanagan

A machine language (or binary) program allows you to do anything the physical limitations of your computer allows. The execution of a machine language routine can be hundreds of times faster than the equivalent set of Basic statements, and a binary program can let you have complete control of the data structures used by your computer. Unfortunately, machine language also requires a great deal more work and knowledge on the part of the person writing it, when compared to Basic or some other higher level language. Still, for many people the speed, control and challenge of machine language makes the writing of binary programs a very attractive alternative to Basic.

With the Series 80, binary programs are usually implemented as extensions to Basic. New Basic reserved words are created by the author of the binary program, and these new reserved words can be used like any other Basic statement or function. Sometimes this new Basic statement actually constitutes the entire program (as with Visicalc), but more often the binary program is designed to be used in conjunction with a regular Basic program; the binary implements commands and functions not found in "normal" HP Basic.

This symbiotic relationship between a binary program and a Basic program is actually a good thing. It makes binary programs very easy to use, once written, and it allows the binary author to concentrate on speeding up or improving just those elements of a program that need it. Unfortunately, it usually also places an extra burden of overhead on the binary author, because the binary must follow a strict format in order for it to successfully execute. This article is the first of a series on binary programs for the Series 80. In this first article we'll examine the structure of a binary program and discuss its various elements. In subsequent articles we hope to delve deeper into HP Series 80 binary programs, because this is one area where there is an enormous lack of available information.

HP Series 80 binary programs are organized around what is referred to as a "shell." The shell is simply a structured collection of elements which must appear in a binary program in order for the binary to successfully execute on an HP 83, 85, 86 or 87. The shells for HP 83/85 and HP 86/87 binary programs are very similar, but there are some differences between the two. However, in both cases the order (and sometimes size) of the various elements of the shell are strictly defined, and must be followed.

A typical binary shell is as follows:

```

        NAM
        DEF RUNTIM
        DEF ASCIIS
        DEF PARSE
        DEF ERMSG
        DEF INIT
PARSE   BYT 0,0
        (Parse routine addresses here)
RUNTIM BYT 0,0
        (Runtime routine addresses here)
        BYT 377,377
ASCIIS' BSZ 0
        (Keyword table)
        BYT 377
ERMSG   BSZ 0
        (Error message table)
        BYT 377
INIT    BSZ 0
        (Initialization routines here)
        RTN
        (Binary program here)
        FIN

```

The faint of heart may wish to stop here (or at least read the next article), but for those intrigued by the meaning of the above cryptic letters, we'll try to explain. The NAN tells the Assembler ROM to create a binary program "control block". For the HP-85 this consists of the program's name and several system reserved bytes, all in a control block 32 (octal) bytes long. For the HP 86/87, the control block is 40 (octal) bytes long, and you must give each binary a number as well as a name. The HP 83/85 manual is notably silent about what goes into the control block, but Section 6 of the HP 86/87 Assembler ROM manual goes into great detail about the contents of an 86/87 control block. The next article in this series will show you what the binary program looks like after it's assembled, so I won't repeat this information here.

After the control block comes five key addresses used in the execution of the binary program. The DEF RUNTIM to DEF INIT section of the shell is to tell the Assembler to put the starting addresses for the runtime (RUNTIM), Ascii keyword (ASCIIS), parsing (PARSE), error message (ERMSG) and initialization (INIT) routines in these spots and in this order.

Following these addresses which point to the key elements of the shell, the binary program then has even more addresses which point to parsing and runtime routines for the individual keywords which make up a binary. This scheme allows a binary to have multiple new statements or functions implemented in one binary. The PARSE BYT 0,0 marks the beginning of labels which define addresses for parsing routines. By putting something like DEF TESTP after the PARSE BYT 0,0, the Assembler will do the work of substituting the address for the parsing routine called TESTPARS in the shell for you. The same holds true for the RUNTIM BYT 0,0 section of the shell. Something DEF TESTR will tell the Assembler to put the address for the runtime code in this section. By the way, if you have more than one command or function in a binary, the order of the parsing addresses, runtime addresses and Ascii keywords should be the same for all three sections of the shell.

Parsing routines tell the system how to interpret Basic statements for storage as machine language tokens. This usually involves getting some parameters, checking for errors and perhaps converting

the parameters to RPN notation. Functions do not require parsing, because you can include some bytes before the function's runtime code which will tell the system how to parse the function.

The runtime code is the binary code which actually does the work when the program is executed. The `BYT 377,377` is a marker used to tell the system where the parsing and runtime routines end. It places two bytes of octal 377 into the shell to mark the end of the routines.

The `ASCIIS BSZ 0` marks the beginning of the Ascii keyword table. This table lists the actually keywords you will type in when writing a program in order to transfer control to the binary program. You can call your binary keywords just about anything you want (i.e. "GEORGE", "XY", "THE THING I KEY IN TO START THE BINARY", etc.), but the order of the keywords must match the order used in the parsing and runtime tables of addresses. By the way, you can even name your new binary routines the same thing as existing Basic statements (i.e. "DISP", "LET", etc.) and the Series 80 will interpret and parse these entries according to the binary, instead of how these words would be interpreted by "normal" HP Basic. The `BYT 377` places an octal 377 in the program shell to mark the end of the keyword table.

The `ERMSG BSZ 0` marks the start of the error message table. If you wish, you can insert custom error messages here, to tell the user what's wrong during parsing or during runtime, if you wish. Once again, the `BYT 377` is a marker to indicate the end of the custom error messages.

The `INIT BSZ 0` marks the beginning of initialization routines, if any. This might be sections of code which put aside memory for later use by the binary, for instance, or it could be code which allows you to "hook in" to the HP's keyboard for later use by the binary. The `RTN` marks the end of the initialization code. It should be noted that both the `ERMSG` and `INIT` parts of the shell must be included in the shell, even if you don't include any special error messages or initialization code in your binary.

Finally you get to the part of the binary where the actual working program code resides. This is where the various parsing and runtime routines are actually found, and where the binary actually does its work.

The `FIN` is another directive to the Assembler to let it know that the binary shell is complete, and that all necessary addresses, parsing code and runtime code are done. If you wish to learn Series 80 Assembly language so you can write your own binary programs, you should be warned that this is not a trivial task. There's a complete lack of novice assembly language material available for the Series 80, and a rather small amount of assembly language material of ANY type for the HP 83/85 or HP 86/87. The Series 80 chip has a rather unusual architecture to it, and if you want to use the internal routines already found in ROM on the Series 80, you'd better be prepared to deal with a variety of numbering schemes and number bases (octal, decimal, BCD and binary) plus RPN notation. The reward for all this effort is the ability to make the Series 80 do what YOU want it to, and to make it do it just as fast as it's physically able to. Many people find assembly language stimulating and intellectually challenging, and you can derive a tremendous sense of accomplishment by getting a binary program to run properly. The first issue of NEWS80S had a review of the HP 83/85 Assembler ROM (see "Do Everything" coupon for back issues). The HP 86/87 Assembler ROM is similar, but the documentation for the HP 86/87 Assembler ROM is much better (although the actual writing of 86/87 binaries is harder). Next month we'll slug it out at the byte level and show you what a binary program looks like assembled.

WHAT DID HE SAY?

If you're new to computing, our lead article on binary programs may have left you in dazed confusion. Binary? Assembled? BCD? Octal? What the heck was he talking about!

This article will give you a brief definition of some of the concepts and terms needed to understand binary programs on the Series 80. It won't teach you all you have to know to write binary programs, but hopefully it will explain some of the jargon to you so you can follow the articles in this series.

Your Series 80 computer has a computer chip (CPU, or Central Processing Unit), permanent memory (ROM, or Read Only Memory), and memory you can read and write to (RAM, or Random Access Memory). To make all this work together, you have a variety of other things in your computer, but these other things don't have to concern us now.

When your CPU is working, it takes "on/off" or "one/zero" instructions from ROM or RAM in groups of eight. Each individual "one/zero" signal is called a BIT. The group of eight bits is called a BYTE. The various combinations of one/zero in each byte can be used as a code that can represent instructions to the CPU, numbers, letters or other pieces of data.

The way you interpret and represent these bytes of data can follow a variety of schemes. Familiar decimal numbers can be represented, and an individual byte can represent numbers 0 to 255, using various unique combinations of eight one/zeros. Octal (base 8) and hexadecimal (base 16) numbering bases are also commonly used to represent a single byte or a group of bytes. Binary Coded Decimal (BCD) is another scheme used, where each byte is broken into two 4-bit pieces (called a nibble, because they're a piece of a byte), and each nibble represents the decimal numbers 0-9. In BCD a byte can represent the decimal numbers 00-99, depending on what each nibble holds. Each byte can also represent letters, and the most common code used for this is known as ASCII.

Your computer can also interpret bytes as instructions to do something or as addresses. An instruction to do something might tell the CPU to add one to the quantity found in a particular byte, for instance, and this instruction to do something could be followed by the address in memory for the byte you want one added to. If you looked at the two bytes or printed their contents on a printer, it's probable that you wouldn't be able to tell the instruction to do something from the memory address (or from a byte coded as a number or letter, for that matter!). All bytes can only hold 256 unique combinations of one/zero, and the CPU depends on the order of bytes looked at to separate instructions to do something from addresses or from data. Naturally, if this order gets messed up, your computer can get completely "confused" and can start doing funny things.

If you tried to code all these one/zeros individually, you could rapidly go nuts (although in the early days of computing hardy souls actually did this). To help you, a computer program called an Assembler can be used. This program "assembles" all the necessary bytes in the required order that results in a binary program that runs on your computer.

On the HP Series 80, the HP supplied assembler is in ROM, and you have to buy this ROM to write assembly language binary programs. The Series 80 requires a particular format for a binary program, and the "shell" described in this issue's lead article describes this format. Instead of using a bunch of numbers or zero/ones to write the program, the assembler lets you use more readable letter combinations (called mnemonics) to indicate what bytes you want in the program. It also does things like translating numbers, letters or addresses into proper byte codes for you, and assemblers usually also provide additional other features to help you make programming easier or more convenient.

CAW!

(for your writing)

CAD, CAM, CAG, CAL, CAZ, CAM...It's a little surprising that CAW is not commonly included among these acronyms. Certainly Computer Aided Writing is as apropos as the somewhat awkward term Word Processing.

You have probably seen our ad recently; I hope you have found them informative. Here I would like to muse with you about this WP phenomenon. In one of several recent books on this topic, the author compared the advent of word processing to the change from hieroglyphics to pen and ink. Overstated I thought; but upon further reflection, perhaps not. Certainly the invention of movable type was the major historical development for the dissemination of information. However the task of the writer hasn't changed much, until now, with the availability of relatively low cost word processing systems. In fact CAW is probably the principle driving application in the exploding phenomenon of personal micros in general. Writing is truly in the throes of a revolution. Concurrently there is much popular interest in the art of calligraphy which is carrying over into the teaching of penmanship in at least some public schools. Interesting times we live in.

TEXT85 is one of a great many WP programs presently in the market, and one of several available for the HP81/85 computers. It is as good as any for your weekly letter to Mom (perhaps with WP your good intentions can be actualized). For larger documents and special format requirements of business and technical professionals, it offers substantially more than any other on these machines. And most any user will tell you it is a pleasure to write with TEXT85.

-Thomas Logan



TEXT85. Designed for you.

- Enclosed is a check/money order for \$_____.
 - Enclosed is an authorized purchase order for the amount \$_____.
- Charge to my MasterCard Visa Cardiac Layman
- Type 81/85 85 Disk Printer
- Please send more information on TEXT85.
 - Send information on other ARC software products as available. I am particularly interested in _____.

NAME _____
CONTACT _____
ADDRESS _____
CITY _____ STATE _____
PHONE _____



© 1985 ARC. All rights reserved. Printed in the United States of America. All other trademarks are the property of their respective owners.



2938 North Madison Ave.
Loveland, Colorado 80537
303/663-3223

Delving into the details of how a CPU works or how various numbering schemes work is suitable for book-length presentation. Tons of books have been written on all this, usually aimed at a specific computer or computer chip. For the Series 80 computers and computer chip, only the manuals that come with the Assembler ROM are available. When we asked HP employees how they learned Series 80 assembly language, we were told that they read the manuals, looked at assembly language listings, and experimented! HP does offer a Series 80 assembly language course, and you might contact your HP sales office to get more information about this. To learn general assembly language principles, we think "How to Program Microcomputers" by William Barden, Jr. (Howard W. Sams Publications, 4300 W. 62nd St., Indianapolis, IN 46268) is a good start. If you can't buy this book locally, try Sams directly or write for the "Yes!" bookshop catalog mentioned in this issue's Rambling column. The Barden book covers several popular microcomputer chips (but not, unfortunately, the Series 80 chip), and the concepts described in the book will form a good foundation for the study of Series 80 assembly language. We plan several more articles in this series to help you explore Series 80 assembly language, and if we continue to receive positive feedback on assembly language information, we'll make Series 80 assembly language an ongoing feature of NEWS80S.

CARDFILE VS. FILE-IT — TWO SIMPLE DATA BASE PROGRAMS

Cardfile, by Hewlett-Packard, is a simple database program that's provided free to owners of the HP 86/87, as part of the "Demo Disk". File-it, by Collins and Associates, is a simple database program for the HP 86/87 that costs \$100. If you're curious as to why you'd want to pay \$100 for something you've already been given "free", then read on.

With Cardfile, the computer screen is set up as an imaginary piece of paper 200 lines long. You only see 16 of these lines at any one time, but you can scroll up and down the "paper" to see all the data held there. Cardfile lets you partition the "paper" into one or more columns. The maximum width available is 77 characters, if you chose to have only a single column. If you have more than one column, you must subtract one character for each additional column you make (for a column separator line), and the total for all the columns must match the width of the screen. With Cardfile, you can load a database from disk or create a new one. You can add lines (but line 200 "drops off" the sheet), delete lines or move lines. You can enter or edit data, using most of the HP's edit keys. You can print data or rearrange (sort) data on any column, either alphabetically or numerically. You can also pick a sub-set of the data using Boolean logic on any one column (i.e. "equal to", "greater than", "less than" some value or string).

To create a database, you first divide up the lines into the various columns you want (up to a maximum of 9). This is done using a very nice graphics display that's the best part of the program. With Cardfile, all commands are given by pressing a function key or a single alpha key (you don't have to press End Line).

The joy of using Cardfile quickly fades, however, the first time you try to sort something. As a test, we set up four columns (2 numeric and 2 alpha). We entered 5 data lines. In one of the alpha fields the data was already in sorted order, in the other it was a random series of 5 characters. We did the same thing for the numeric columns (one column was already sorted, the other was at random). Then we pulled out our trusty stopwatch and did an alpha sort on the alpha column that was already in sorted order. It took one minute and nine seconds to sort 5 items which were already in sorted order! It took 1 minute and 30 seconds to sort the random alpha column.

Remember, this was for 5 items. You could grow old waiting for Cardfile to finish sorting.

Numeric sorts were a bit better (both columns sorted in 53 seconds or so), but considering the fact that only 5 items were involved, the numeric sort was less than impressive, too. With so many nice little features to this program, we were frankly amazed at the slow sorting.

Another problem with Cardfile is the fact that you're limited to 200 lines (or entries), regardless of how much memory you have in your machine. The program uses screen memory to store the data, which is a "clever" programming technique, but one which is limiting (and, perhaps, a cause of the slow sort performance).

When you want to print your database, Cardfile allows you to sort and eliminate lines selectively, using Boolean logic. The remaining "page" is then printed for you. You can't delete selected columns from any line printed, so it's an "all or nothing" proposition if you print a line. If your database contains sensitive information that you don't want printed on a particular output report, I suppose that a pair of scissors and a paste pot is the only answer.

A final point about Cardfile is that the only documentation provided is a file on the Demo Disk that displays a short blurb about the program and the other demos on the disk. I can see some novices having problems with reading this file and working the Cardfile program the first time around.

Since Cardfile is provided to you free, it may be unfair to be too harsh with the program. Cardfile does illustrate some very slick programming techniques, especially in its use of graphics, but even for a "free" program we were disappointed because the sorting and operation of Cardfile didn't match the high standards set by the rest of the program.

Jumping from "free" to \$100 is a big step, but for people in the market for a simple database, this step may be worth making. At \$100 File-It, by Collins and Associates, provides many areas of increased functionality over Cardfile.

File-It operates on a more abstract concept than Cardfile. With Cardfile it's easy to "see" how the imaginary paper looks and can be divided into columns. With File-It, you work with records and fields and files, which may be more difficult for the beginner to deal with. File-It rewards your coming to grips with these concepts in several ways, however. For instance, an individual record can be up to 160 characters long, much longer than Cardfile can display on a line. File-It also provides for use of your HP's memory (up to 96K) by giving you more potential data records than Cardfile's fixed 200. On a 64K machine, File-It could store around 390 records equal to Cardfile's 77 character maximum. On a 96K machine File-It would store around 800 records.

Creating a file requires you to spend a little time thinking about the number and size of the fields you want in each data record. File-It 'doesn't have the nice interactive graphics creation found in Cardfile. With the help of File-It's manual, however, this isn't too hard, and after you've set up a couple of data bases, this should present no problems.

We tried the same sort test that we tried with Cardfile, and we got much more satisfactory results. The Alpha fields took around 2 seconds to sort on, and the numeric fields took 1 to 2 seconds. This was for only 5 records, but it represents a sort speed that's 45-90 times faster than Cardfile!

Besides sorting, File-It allows you to add, delete, or insert a record. You can also find a record, searching for data in any field, or combinations of fields (up to four). Boolean logic can be used when picking which field(s) you want to search on. File-it has a "Tally" command, which allows you to add up numeric data in a field, giving you a very useful way of getting totals from your database.

When you print out your data, File-it allows you to selectively pick which fields it should

print, and the order that you want it printed in. This is considerably more powerful than Cardfile's complete line scheme, but I was a bit disappointed with the format that the reports come out in. The data comes out "stacked", as with the following example:

```
Widget — Blue
1234
36
```

```
-----
Widget — Red
5678
27
-----
```

In the above example the first line is a description, the number is a part number, and the second number is a quantity. File-It will display the headings for you at the start of the report, and you can rearrange the order the data will appear. You can also delete specific data elements from a particular report, so if you also had a price field in the above example database, you could print it or leave it off particular reports, at your option. There is no way to combine data elements into columns on a single line, add column headings, or to change the report format very much. I consider this to be the weakest part of the program, but when you weigh it against File-It's \$100 price tag, you realize that some tradeoffs had to be made.

Collins & Associates provide both a 64K and 96K version of the program. The program is mostly written in Basic, with a binary program. The disk is unsecured, so you don't have the "backup blues" if you have the good sense to make a backup copy that can be placed in a safe place(!).

If you have only trivial need for a database, then Cardfile could match your needs. If you have need for more capability, then at \$100 File-It is worth looking at. File-It doesn't have all the features of some of the other database programs on the market, but it doesn't have the \$250-700 price tag, either. All in all, it's a good choice for the HP-86/87 user looking for a simple, low cost database program.

MENU'D Part One — by Dale Flanagan

Programs that write other programs hold a special fascination for a lot of computer users. For the last couple of years there have been several microcomputer programs which purport to write other programs, given just a few simple inputs from the user. Most often these program generators depend heavily on the user's ability to define the program to be written in a way that fits the capabilities of the program generator. In other words, to loosely quote Maslow, when the only tool you have is a hammer, you have to look at every problem as a nail.

These program generators have brand names like Pearl, CORP, The Last One, and The Creator. They're written for CP/M, IBM PC and Apple micros, and thus far we haven't seen one for the Series 80. In this series of articles we're going to discuss, in simple terms, how this type of program can be written and illustrate our discussion with MENU'D, a program to generate HP Basic screen menus for the HP-85. It's our hope that MENU'D will stir the interest and imagination of readers of NEWS80S, so that other useful Basic routine generators get written and, hopefully, shared. Perhaps some author will even get motivated enough to write a complete program generator.

If you've written more than a few programs, you soon discover that a great deal of

NEW FOR HP87 USERS

PROGRAMS TO INCREASE YOUR HP87's POWER & SAVE YOU TIME & MONEY

*** "FILE-IT" ***

FILE IT IS A USER-FRIENDLY COMPUTERIZED INDEX CARD FILE SYSTEM. IT IS FASTER AND FAR MORE POWERFUL THAN ANY MANUAL SYSTEM. DATA FILE HAS A CAPACITY OF 50,000 CHARACTERS. EACH RECORD CAN HAVE 10 FIELDS OF UP TO 100 CHARACTERS OF ANY TYPE OF ALPHA NUMERIC DATA. NUMBER OF RECORDS IS DEPENDENT ON NUMBER OF FIELDS AND SIZE.

KEY LABEL COMMANDS:

- | | |
|-----------------------|---------------------------------|
| 1. ADD RECORD TO FILE | 8. DISPLAY RECORDS |
| 2. EDIT RECORD | 9. PRINT RECORDS |
| 3. INSERT RECORD | 10. STORE FILE TO DISK |
| 4. DELETE RECORD | 11. CREATE NEW FILE |
| 5. SORT RECORDS | 12. READ FILE FROM DISK |
| 6. FIND A DATA STRING | 13. CLEAR FILE |
| 7. TALLY A FILE (ADD) | 14. READ FILE DIRECTORY ON DISK |

DATA MAY BE SORTED ON ANY FIELD, ADDED, INSERTED, DELETED, DISPLAYED OR PRINTED WITH THE FIELDS IN ANY ORDER OR SUPPRESSED. UP TO 4 KEYS CAN BE USED IN A SEARCH WITH RELATIONS OF =, <, >, OR, NOT OR AND TO FIELD DATA. KEYS CAN BE IN ANY FIELD OR THE THE SAME FIELD AND IN ANY LOCATION IN A FIELD.

REQUIRES 96K RAM FOR 50K FILE, 64K FOR 30K FILE OR 128K FOR A 65K FILE.

UPGRADEABLE BY USER FROM 30K TO 65K FILE SIZE AS MEMORY IS ADDED
\$100.00 PLUS \$3.00 SHIPPING & HANDLING

*** "PLOT-IT" ***

PLOT IT IS THE MOST COMPLETE AND COMPREHENSIVE LINE CURVE PLOTTING PROGRAM AVAILABLE. CURVES OF UP TO 100 POINTS CAN BE PLOTTED ON THE CRT OR PLOTTER. THERE IS NO LIMIT TO THE NUMBER OF CURVES THAT CAN BE PLOTTED ON THE SAME PLOT. DATA AND/OR PLOTTING FORMATS CAN BE STORED AND RECALLED TO UPDATE A PLOT ON THE CRT OR PLOTTER. COMPLETE PLOTS CAN BE STORED AND RECALLED. CRT PLOTS CAN BE PRINTED ON A PRINTER. DATA CAN BE MANIPULATED BY ADDITION, SUBTRACTION, MULTIPLICATION, DIVISION, RAISED TO ANY POWER OR STATISTICAL ANALYSIS CAN BE DONE WITHOUT LEAVING THE PLOT OR PROGRAM.

KEY LABEL COMMANDS:

- | | |
|------------------------------------|------------------------------------|
| 1. KEYBOARD DATA ENTRY | 8. LOAD DATA OR FORMATS FROM DISK |
| 2. SETUP PLOTTING FORMATS | 9. COPY CRT PLOT TO PRINTER |
| 3. PLOT DATA | 10. CLEAR DATA FROM ARRAYS |
| 4. EDIT DATA, INSERT, DELETE | 11. PRINT OR DISPLAY DATA |
| 5. STORE DATA OR PLOT FORMATS | 12. DISPLAY PROGRAM STATUS |
| 6. LABEL PLOTS | 13. STORE OR GET CRT PLOT ON DISK |
| 7. MANIPULATE DATA PLUS STATISTICS | 14. DISPLAY FILE DIRECTORY ON DISK |

DATA MAY BE EDITED, INSERTED OR DELETED. DATA CAN ALSO BE DISPLAYED OR PRINTED ON THE PRINTER. A STATUS REPORT OF ALL PLOTTING PARAMETERS, DATA AND PLOTS IN PROGRESS AT THE TOUCH OF A KEY. COMPLETE FREEDOM IS PROVIDED ON THE PORTIONS OF DATA TO PLOT, LABEL SIZES AND POSITIONS. PLAIN OR GRID PLOTS CAN BE DONE. PLOT IT GIVES YOU THE GREATEST FLEXIBILITY OF ANY HP87 LINE PLOTTING PROGRAM AVAILABLE AT THE LEAST COST. EITHER CRT OR PLOTTER GRAPHS CAN BE UPDATED WITH MORE DATA OR LABELS AT A LATER TIME BY RECALLING TO THE CRT OR REINSERTION IN THE PLOTTER.

REQUIRES 96K RAM, PP ROM, DISK DRIVE, PRINTER AND PLOTTER

\$150.00 PLUS \$3.00 SHIPPING & HANDLING

BOTH PROGRAMS ARE MENU DRIVEN WITH NO CYRPTIC COMPUTER COMMANDS
INCREASE THE USE AND POWER OF YOUR HP87 WITH MINIMUM COST

CASHIERS CHECK OR MONEY ORDER

COLLINS & ASSOCIATES, INC.

187 FLYING MIST ISLE -- FOSTER CITY, CA 94404 (415)571-6991

SPECIAL OFFER! Order BOTH File-It and Plot-It and get a 10% DISCOUNT!

programming is repetitive. Many programs share common functions like screen generation, error checking, file handling, sorting and editing data. After awhile a programmer tends to develop a little library of routines which are used in almost every program written. If you have the HP Advanced Programming ROM, you can actually use the same subprograms as you build new programs with modules like Tinker Toys.

After a moment's reflection, you can also see that many elements in programming can also be similar, but not exactly the same. By this I mean that parts of various programs can be very similar in coding structure, but they don't easily lend themselves to becoming a common subprogram which can be used in a variety of other programs. A good example of this is a menu screen.

Organizing a program around menus is a common and very efficient way to program. The menu screen presents the various choices to the user, and after the user has selected a choice, the control of the program is routed to the sections of the program which really do the work. The coding for menus in various programs is very much alike, but usually the number of menu choices and the messages displayed for each choice are going to be different from program to program.

I suppose you could be clever about writing a common menu subprogram which would take a variety of parameters and display and set the various menu options, but it's more interesting to write a program that will write the Basic code you'll need to handle the creation of menu screens.

When you write a section of a program, you're using a limited vocabulary and a fairly rigid syntax to create the desired result. HP Basic does allow some variations in syntax (i.e. "IF X THEN 125", "IF X THEN GOTO 125", ETC.), but these variations are pretty small. Naturally, there are some things we might want to vary, and by looking at a simple example, we can discover what those things are.

```

10 ON KEY#1 "DO IT" , GOTO 1000 ! ASSIGN KEYS
20 ON KEY#2 "EXIT" , GOTO 2000
30 KEY ON ! DISPLAY CHOICES
40 CLS ! START MENU SCREEN
50 DISP TAB(11 ); "MENU TITLE" ! ENTER TITLE
60 DISP ! SPACE DOWN
70 DISP "K1 = DO SOME WORK"
80 DISP "K2 = EXIT PROGRAM"
90 DISP
100 DISP TAB (6);"YOUR CHOICE (K1-K2)?"
110 GOTO 110 ! WAIT FOR KEY PRESS

```

Probably the first thing we'd like to vary is the line numbering and increment. Next would be assignment of the various function ("K") keys, and what line number they're routed to. It would also be nice to change GOTO to GOSUB, if we wanted to. The menu title would have to be changeable, and it would be nice if the program did the centering for us. Of course, the message displayed on the screen to explain each function key would have to be changeable, and it would be nice to make the number of function keys used variable. The rest of the menu code can be the same for each menu written.

If we wrote a program that produced text that looked just like the menu code shown above, we could use various string variables and HP Basic's string functions to make the text/program as flexible as we need it to be. Once we had the code written to look like text, we'd then be faced with the problem of converting the text into a Basic program. Fortunately, that problem has been solved for us.

On the HP85 we have a binary program called "DGTSAV" which will convert a

word power



WRITE/IDEA
for the HP 83, HP 85,
HP 86 and HP 87
is listed under
the HP+ program



**THRESHOLD
SOFTWARE
INC.**

1032 Irvine Road, Suite
Sacramento, CA 95815
(916) 320-8183

THRESHOLD SOFTWARE INC. and stylized TS are trademarks of THRESHOLD SOFTWARE INC. Identical program names of programs marketed through THRESHOLD SOFTWARE INC. are the property of the program owners and are used by THRESHOLD SOFTWARE INC. with permission. All rights of THRESHOLD SOFTWARE INC. and marketed authors are reserved.

You selected your HP series 80 personal computer for its scientific and technical capabilities. Now, WRITE/IDEA expands your HP system to encompass word processing applications.

WRITE/IDEA is a sophisticated, flexible, easily mastered word processing program to enable the production of documents from notation through finished hard copy. Reports, articles, letters, manuscripts, anything requiring word manipulation can be input and edited with simple commands and interfaced to any HP series 80 compatible printer. You can even make projection slide transparencies of notes, tables, or outlines using a peripheral plotter like the new 7470A. All this with a portable system you can carry to conferences, seminars and meetings.

WRITE/IDEA extends the power of your HP series 80 to increase its flexibility and productivity. An investment in WRITE/IDEA gives you a program that:

URNS YOUR HP SERIES 80 INTO A PORTABLE WORD PROCESSOR

PROVIDES QUICK, CONVENIENT SINGLE TOUCH EDITING COMMANDS

FEATURES FULL CURSOR CONTROL/SEARCH-REPLACE/GLOBAL SEARCH/FILE MERGE/BLOCK MOVEMENT/FILL INSERTION/CENTERING/JUSTIFICATION/DECIMAL & NORMAL TAB AND MUCH MORE

PROVIDES FLEXIBLE FORMAT COMMANDS FOR PROFESSIONAL PRINTOUTS

INTERACTS WITH ANY SERIES 80 COMPATIBLE PRINTER AND THE NEW 7470A PLOTTER

WRITE/IDEA COMES ON EITHER A 5.25 INCH DISK OR A TAPE CARTRIDGE WITH COMPLETE INSTRUCTIONS AND REFERENCE MANUAL.

PRICE: \$250 — \$300 FOR THE HP 86/87

CONTACT YOUR HEWLETT-PACKARD DEALER FOR A DEMONSTRATION
FOR ADDITIONAL INFORMATION, WRITE OR CALL THRESHOLD SOFTWARE

OTHER SERIES 80 SOFTWARE FROM THRESHOLD SOFTWARE:

COSAC - cost accounting and pricing
FILE/IDEA - data base management

\$495.
NEW! call for details

textfile on tape or diskette into a program in the HP's memory. For the HP86/87, the computer's demo disk has a binary called "GETSAVE" that does the same thing. There are a couple of other ways to accomplish the same thing, by the way. One is to display the program lines on the screen, move the cursor under the first line, and press the END LINE key until each of the program lines is fed from screen memory to program memory. Another is to use a binary program which would read the textfile into the keyboard buffer, fooling the computer into thinking that you are keying in the program lines. Since "DGTSAV" and "GETSAVE" are readily available, however, we'll just use them.

To review our approach; what we've done is looked for a programming problem which is commonly found in many programs, but which is not so rigid that it would be better suited as a standardized subroutine or subprogram. We examined how the code would look in an actual program, and then analyzed what parts of the code would be variable. Our intention is to write a program that will produce a textfile that will look like the program lines we want, and then to use the "DGTSAV" (or "GETSAVE") binaries to convert this textfile into an actual program fragment, suitable for merging or incorporation in any program we're writing.

Next month we'll look at the code for MENU'D. This is a program written for an HP-85 using the Advanced Programming ROM, but which can easily be converted for the HP-86/87 with minor changes. The code we'll look at next month will be the core of the program, which does most of the work. The month after that we'll present the sections of the program which add bells, and whistles, allowing you to save menu parameters to tape or disk and to edit parameters, making similar or updated menus easy to write (or rather, to have MENU'D write it for you!).

RAMBLING by Dale Flanagan, Editor

IBM OR HP? Friends who know we do software for both IBM and HP computers often ask me which one I'd recommend. I used to go through a lot of pro and con type analysis on the IBM PC versus the HP-86, but I've now reduced all this to a couple of statements: "If you're going to do your own Basic programming, buy the HP. Otherwise, buy the IBM." In processing speed, the IBM PC will run circles around the HP-86, and there is (and will be) tons of off-the-shelf software for the PC. But raw processing speed and total number of software packages don't tell the whole story. The HP-86/87 has a tremendously powerful Basic available to it, and for people that write their own software, this Basic can drastically cut development time. When you consider the total time between the idea for a program and finally getting it to run, the actual run time of the program is not too important. The time and ease with which you can translate your ideas into programs are.

The HP Basic uses all the memory of the machine (IBM Basic is limited to 64K of memory and data space, no matter how much memory you have in the machine. The IBM DOS and Disk Basic will steal about 26K, so you have to go to a 96K configuration to end up with a user memory of 64K with IBM Basic). Features like timed interrupt processing from HP Basic and named COTO and COSUB calls make it a joy to do programming, and most of the "little" programs we do for our own internal use are now being done on the HP-86, just because it's so much faster to get the program completed. The 86/87 has an adequate core of software, and if HP can be persuaded to open up some on information about the internal workings of the machine, I'm sure a greater variety of software will become available in the future. The IBM PC is a nice machine. But so is your HP, and if you own an HP-86 or HP-87, you have a computer with definitely superior features in many areas.

8 BITS? 16 BITS? While on the subject of the IBM, I find it interesting that the IBM PC is often called a 16 bit computer, even though it has an Intel 8088 microprocessor chip. The 8088 has an 8 bit data bus, and handles memory addressing through bank switching. Under old conventions, this would classify as an 8 bit processor (Intel makes a "true" 16 bit processor, called the 8086, which is compatible with the 8088), but many people are called the IBM PC a 16 bit machine because it uses 16 bit data registers, and internally the chip handles data 16 bits at a time (even if it takes two fetches to bring the data in across the 8 bit data bus) . You might be interested to know that the Series 80 chip also has an 8 bit data bus, but it can handle data 64 bits at a time internally (it has four 8 byte registers). Does this make the entire Series 80 line, including the HP-85, 64 bit computers? Not really, but it does make for interesting conversation when competing for "bragging rights" with other microcomputer owners.

COMPUTER JOURNAL OF PPC. The Personal Programming Center (PPC) is an organization well known for its activity with programmable calculators, especially the HP-41. PPC is now branching out into the HP Series 70 and Series 80 world with the "Computer Journal of PPC", edited by Richard Nelson. This is a bi-monthly publication that presents a variety of topics of interest to owners of HP-75 and Series 80 computers. The first two issues were primarily oriented towards the HP-75, but since HP-75 Basic and Series 80 Basic are similar, many of the Basic programming routines presented are still useful to Series 80 owners. Of specific interest to Series 80 owners in the first two issues was a disassembled listing of HP-87's CP/M BIOS (Basic Input Output System), plans for making an HP-85 carrying case, and information on CP/M upgrades from Systematica. Subscriptions to the "Computer Journal of PPC" are \$25 per year, plus a one time "processing fee" of \$8. This fee also makes you a member of PPC's computer group. If you wish to also receive their calculator journal, there is an additional fee of \$25 per year. Please contact PPC at 2545 West Camden Place, Santa Ana, CA 92704 for more information about PPC and the "Computer Journal".

PASCAL NEWS. For users of UCSD Pascal (TM of the UC Regents) on the HP 86/87, another publication of interest will be "Pascal News" at 2903 Huntington Road, Cleveland, Ohio 44120. "Pascal News" is an irregularly published compilation of letters, articles and program listing dealing with Pascal. It is not aimed specifically at UCSD Pascal, but it does have a variety of Pascal hints and topics useful to all versions of Pascal. In the past, this particular publication has spent an enormous amount of effort on standardizing Pascal and publishing Pascal compiler standards programs, but they've also published a good amount of utility routines and several complete programs. "At least" three issues a year may be had for \$20, and they'll give you three years worth for \$40.

NETWORKING. I don't know whether to thank or curse Curt Adams, author of the Networking program (Gaia Communications, Rt.2 Box 321C, Corvallis, OR 97333). Curt gave us a copy of Networking for a review (featured next issue), and it got us turned on to the possibilities of using data communications in our business. As most of you know, we work with several brands of computers. We've always had them talking to each other through RS232 ports and cables, but we haven't really gotten into data communications via telephone. Several years ago I did a great deal of work on two time sharing systems, and this type of computing didn't appeal to me as much as the personal computers. For various businesses we have three offices, and it got tiring for us to haul around even a personal sized computer when we wanted a computer at one office to "talk" to a computer at another office. Enter Networking and the possibilities of a personal data communications network.

Naturally, with Networking you can also "talk" to large time sharing

systems like The Source (in fact, a free Source subscription comes with Networking if you get it before June 30, 1983), but for us the ability to transmit text files and programs between the computers at our offices is very exciting, especially since you can set things up so the transmission can be done unattended (we have a lot of offices, but not too many people!). So far we've bought a couple of intelligent modems, a communications program for our IBM computer, two programs for our Apples, plus the usual cables and stuff that nickel and dime you to death. And, of course, we have Networking running on our HP-85. The Networking program compares very favorably with the software we've found for our other computers, and Networking has several unique features that we'll call out in our review. What's caused our ambivalent feelings towards Curt about all this is the time we're spending playing with data communications, communications protocols, bulletin board systems, etc. instead of writing software, reviewing software, or writing articles. Curt, by the way, seemed delighted over the ever deepening hole we're digging for ourselves as we become more and more involved in data communications. I guess misery loves company.

COMPUTER BOOKS are even appearing in supermarkets, but sometimes it's hard to find a particular book. We received a catalog from the Yes! Bookshop (1035 31st NW, Washington, DC 20007, phone (202) 338-7874) that had a very comprehensive listing of over 800 titles, plus a short review of each book. This shop has the William Barden book we've mentioned a couple of times, plus most of the others which have been mentioned in our reviews and stories. The catalog costs \$2.00, which is supposed to be refunded with your first order. If you live someplace where computer bookstores aren't springing up like weeds, then this is a catalog worth having.

FEEDBACK — From the Readers

FROM ALBERT HAYES (Fullerton, CA): Thanks for the PACK'R program. I now have it running on my HP-87, including the use of binary programs. I do notice your step 1040 does not pick up the end-of-file with the HP-87 logic. I have had to add two steps, as follows:

```
1035 IF B$= "" THEN 2000
2065 IF B$= "" THEN 4010
```

Also, for the HP-87 and disc drive:

```
3160 (change 6 to 10)
```

FROM RICHARD H. BAKER (SPARKS, NV): Just got and enjoyed Issue #5, particularly the section on the AP ROM, and especially your comments on LINPUT. I had overlooked it in two programs that required the entry of addresses, and with it I no longer have to remember not to enter commas.

A point that might interest some how use that command: the AP ROM manual (p.16) is not strictly correct when it warns "The statement doesn't allow numeric inputs. . ." LINPUT will accept numeric inputs, but only as strings, as the accompanying illustration in the manual shows, and, of course, any such string can be used numerically with the VAL command.

P.S. This is being written on Text85, which I bought as soon as the offer was made to sell it without the copy lock. I am completely satisfied with it, but I must admit the documentation takes some studying.

FROM DR. HEINZ SMIRMAUL (DUNCANVILLE, TX): I have received your magazine for awhile and I am very happy with it. We have used the FOIL'D program and modified it to run with our HP plotter. Your reviews are extremely helpful, especially with respect to the 86 and 87XM series.

One of the problems that has always been with HP is data storage. We have in use at

the present time three HP 85's, and we're using the Winchester drive with 5 1/4" disc. Our problem is that we use the system for patient data control, and since HP Basic permits one to create files with long strings, it is possible to create files up to one megabyte on the Winchester. The big problem is that these files cannot be backed up on the 5 1/4" disc. The solution that HP has is to sell an 8 inch floppy disc with the Winchester, and they sell this combination under one product number. The 8 inch floppy disc, to make matters worse, is in the dual 8 inch drive unit disc case, which is a very large piece of equipment. There is certainly ample space to put the Winchester beside an 8 inch floppy disc. This combination permits the programmer to dump single Winchester discs in one fell swoop onto 8 inch floppies, which are one megabyte storage units. The price that is asked for the combination unit is astronomical and seems to be so set that it tends to discourage users from buying it.

I would very much like to hear your opinion on this problem and to hear whether HP has any future plans of combining Winchester drives with either removable hard discs, or removable floppy discs which have a storage capacity of one megabyte. Until that is accomplished, one still is stuck with the problem of being unable to back up the single Winchester.

NEWS80S: Operating in a professional environment without proper data backup is the data processing equivalent of Russian Roulette. Unfortunately, in the microcomputer area no one seems to have the problem of reliable, low cost Winchester hard disk drive backup solved. There seem to be four alternatives various companies are working on: (1) Ignore the problem; (2) Tell the user to buy a second hard disk for backup (a good, but expensive solution); (3) A variety of tape backup schemes, including a few that use home video tape recorders; and (4) Floppy disks (HP's solution).

Disk drive for the Series 80 are handled by HP's division in Colorado (I believe in Greely), and not in Corvallis. Except for the HP—86's 9130A drives, HP uses many of the same drives in Series 80, Series 100, Series 200 and Series 1000 computers, and the compromises caused by this cause problems. You've mentioned one, with the "9138A Mass Storage Bundled Units", which consists of a 5 1/4" Winchester disk drive in a half empty case and a single 8" disk drive, also in a half empty case (of enormous size). The Winchester does come with either a 3.5" or a 5 1/4" drive in the same case. While this is great for software portability, it does nothing for the backup problem, unless you're willing to copy the hard disk contents to 19 or so diskettes and keep your data files so they can be segmented in 270K chunks.

As owners of single 9121S, 82902M or 9895A drives know, HP often sells you a box big enough for two drives when you buy a single drive. This would be fine if they also offered an upgrade kit, so you can later add the second drive, but current HP catalogs are conspicuous in the absence of any upgrade kits. This may push some customers into buying a dual unit instead of a single unit with their initial orders, but it also provides a point of dissatisfaction because a convenient disk storage upgrade path is not provided.

Regarding HP's disk pricing, the following information may be of interest. With the 9130A drive used on the HP-86, you're given 270K of storage for \$860, and the disk interface and DOS is built into the HP-86. Compare this to, say, the IBM PC. On the IBM you get 320K of storage for \$850, but you also have to buy IBM DOS (\$ 40) and an interface board (about \$350, but it has the capability to handle up to four drives) . The first drive on an IBM costs you about .39 cents per byte of storage and on the HP it costs you .32 cents. For a 2 drive system, the cost per byte is still .32 cents for HP, but also .32 cents for IBM (you only buy DOS and the interface card once). By the way, the generic drive used on the IBM (a 5 1/4" Tandon TM100-2 drive) can be - purchased for around \$300 retail. We believe HP also uses the Tandon TM100-2 drive in the 9130A, the 82901M and 82902M drives, but interfacing the drives may be a problem.

For the HP-85, you must buy a mass storage ROM (\$165), an HPIB interface (\$395), and an 82902M or 82901M (\$1520 or \$2230). This works out to .77 cents per byte for a single 5 1/4" drive system and .52 cents on a dual drive system. If you go with 3.5" drives, the price drops to .61 cents per byte for a single drive system. If you go with the 8" single drive the price goes to .52 cents per byte, but only because you're getting a megabyte of storage.

The 8" drive is not very popular on the Series 80, primarily, we think, because HP is asking \$4580 for a single 8" drive! Since the 5 megabyte Winchester hard drive is only \$4560, our conclusion is that the backup solution for HP's Winchester hard disk drive is a second Winchester hard disk drive. This will not only provide faster backup times, it will also give you the capability to continue to operate if one of your Winchesters has a major problem. Hanging \$9000 worth of hard disk drives off a Series 80 is not an ideal solution from an economic standpoint, but it is one of the best solutions available until HP reviews its disk pricing, introduces new products, or one of the independent hardware manufacturers enters the picture.

CONVERTING CARDFILE FILES TO WORD/80 by Prof. Bruce Liebert

I am sure there are quite a few readers who have used the program Cardfile to do word processing by setting the number of columns to one, as suggested in the HP Cardfile instructions. Now that HP has a "real" word processing program for Hp 86/87 computers (Word/80), I would suspect that a number of people will want to be able to read the Data files generated by Cardfile with the Word/80 software. Unfortunately, in the Word/80 software I have, this is not readily done. With the enclosed program, Data files generated by Cardfile are converted to Data files that can be read by the "GET" command in Word/80.

In order to use this program, which I call "CARD->WORD", you input the present Cardfile Data file name, then the name of a "GET" file that will be eventually read by Word/80. Finally, you must input a unique string for the last line of the present cardfile data file, so the conversion program knows when to stop. If you do not know what this might be then you must inspect the present Cardfile Data file by running the Cardfile program. If, for some reason, there is no string unique to the last line, then edit the Data file and add an appropriate string as the last line, such as "END-OF-FILE", to the last line. This will be the string you input when prompted by the CARD->WORD program. Of course, this string can be deleted once the converted data file is read by Word/80.

```

10 ! 'CARD—>WORD' by Bruce E. Liebert REV. A FEB83
20 ! PROGRAM TO READ 'CARDFILE' DATA FILES AND CONVERT THEM INTO
30 ! 'GET'COMPATIBLE DATA FILES TO BE READ BY WORD/80
40 DIM Q$(80), S$(80)
50 CLEAR
60 DISP "CARDFILE DATA FILE NAME"; @ INPUT D$
70 DISP "WORD/80 'GET' FILE NAME"; @ INPUT G$
80 DISP "INPUT STRING UNIQUE TO LAST LINE OF CARDFILE"; @ INPUT S$
90 DISP "INSERT DISC WITH CARDFILE DATA FILE "; D$;
" AND PRESS 'CONT ' " @ PAUSE
100 DISP "CREATE " ; G$; " (Y/N)"; @ INPUT A$
110 IF UPC$(A$) = "Y" THEN CREATE G$, 65
120 ASSIGN# 1 TO D$
130 ON ERROR GOTO 150
140 READ# 1 ; Q @ GOTO 140 ! strips off leading numbers

```

```

150 ASSIGN #2 TO G$
160 ON ERROR GOTO 200
170 READ# 1 ; Q$ @ DISP Q$ @ PRINT# 2 ; Q$
180 IF POS (Q$,SS) # 0 THEN 200 ! end of cardfile data file
190 GOTO 170
200 ASSIGN# 1 TO * @ ASSIGN# 2 TO *
210 DISP USING "8/5(K)"; "CARDFILE ";D$; " TRANSLATED TO "; G$; "FOR
WORD/80"
220 END

```

"RUNNING" A PROGRAM IN HP BASIC

HP Basic doesn't allow you to RUN a program directly from tape or disk. You must first LOAD the program, then you can RUN it.

You can save some time and a few key—strokes with a little trick. Instead of loading and running a program, use the CHAIN command to load in and run a program directly from tape or disk. The syntax for this is:

CHAIN " filename"

PACK'R for HP-85

PACK'R is a fantastic utility that takes an HP-85 Basic program and removes all remark statements and combines as many statements per line as possible.

The result? A more compact, faster running program that has more data space and that takes up less space on disk or tape.

The exact amount of program compression and program speed-up will vary from program to program, but in test we've had size reductions of 25% and speed improvements of 7%.

Basic program and binary program on HP-85 data tape, complete with instructions, only \$39.95!

See coupon on opposite page.

SC CALC

A Program For calculating The Three Phase And Ground Fault Short-circuit currents In Industrial And commercial Electrical Distribution Systems.

* Per ANSI and IEEE Standards

* Maximum 375 buses

* Will handle future revisions to System

* HP-83, 85, 86 and 87

* Price \$1,325

* User's Manual and Sample Short-circuit Study - \$50

For further information contact:

Jackson & Associates
P.O. Box 562
Spanish Fort, AL 36527
(205) 626-6890

n@w s 80 s

**The Microcomputer Journal
For HP Series 80**

**News80s ran for 12 issues between 1982 and 1984
(#1, #2, Special Issue and #3 thru #11).**

**It was an independent newsletter edited by Dale Flanagan for:
HP-83, HP-85, HP86 and HP87 Personal Computer users.**

**These are used with the permission of Dale Flanagan, who retains the
copyright.**

Scanned and converted by M. A. Cragg

NEWS80S "DO EVERYTHING" COUPON
P.O. BOX 1329, REDONDO BEACH, CA 90278, U.S.A.

NAME:

ADDRESS:

SUBSCRIPTIONS - [] NEW [] RENEWAL

[] 6 ISSUE SUBSCRIPTION (ONE YEAR) - \$15 U.S., \$18 Canada/Mexico, \$30 Others.

[] 12 ISSUE SUBSCRIPTION (TWO YEARS) - \$30 U.S., \$36 Canada/Mexico, \$60 Others.

BACK ISSUES OF NEWS80S

[] ISSUE #1 (HP85 Assembler ROM review, "For/Next" Command, hints; 8 pages) \$2 North America, \$3.50 Elsewhere.

[] ISSUE #2 (HP-87 Issue; '87 Tech Specs, '87 User's View; 12 pages) - \$3 North America, \$4.50 Elsewhere.

[] SPECIAL ISSUE (FOIL'D Program listing, HP86 Introduction, LOADBIN, reprints from Issues 1&2) - \$3.00 North America, \$4.50 Elsewhere.

[] ISSUE #3 (PACK'R Program listing, Anatomy of Basic, What's CP/M?; 16 pages) \$3 North America, \$4.50 Elsewhere.

[] ISSUE #4 (5 HP85 Word Processors reviewed, Pascal for Series 80; 20 pages) \$3 North America, \$4.50 Elsewhere.

SOFTWARE FOR SERIES 80 - All programs are Basic and Binary combinations. Disks have BOTH HP 83/85 and HP 86/87 versions of program. Tape is '85 only.

FOIL'D makes overhead transparency masters, report covers, etc. using an Epson or HP89205A printer. May be modified for use with other printers.

[] 3.5" disk \$39.95 [] 5.25" disk \$34.95 [] HP85 tape \$39.95

PACK'R compresses Basic programs, removing remarks and combining as many statements as possible. The result is a more compact program that can run faster, too. This is a "must have" utility for the Basic programmer.

[] 3.5" disk \$39.95 [] 5.25" disk \$34.95 [] HP85 tape \$39.95

TEBS (Time and Expenses Billing System) is designed specifically for the single practice consultant, programmer or professional who bills on a time and expenses basis. Provides time/activity report and Billing Invoices. Files organized by client. Not for multi-person firms. HP85 version needs 32K RAM & PP or I/O ROM.

[] 3.5" disk \$99.95 [] 5.25" disk \$99.95 [] HP85 tape \$99.95

_____ << SALES TAX (California residents only. No tax on subscriptions.)

_____ << SHIPPING ON SOFTWARE (\$2 per program North America, \$5 Elsewhere)

_____ <<< \$ TOTAL <<<

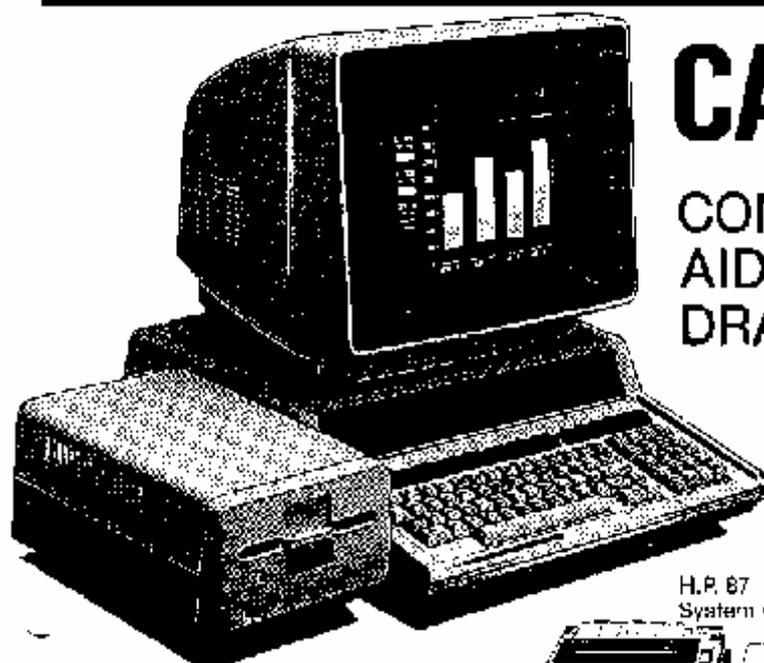
CREDIT CARD PURCHASERS - Mastercard and Visa only.

_____ M/C _____ Visa Card #: _____ Expires _____

Credit card orders may be placed on a 24 hour answering machine by calling (213) 374-3251. Please give us complete information & a daytime phone # for questions.

NON-U.S. ORDERS payable in U.S. currency or International money order. Subscriptions and orders outside the U.S. are sent airmail.

BILLING - We will send you a bill if necessary for your order processing, but no merchandise will be shipped or subscriptions entered without pre-payment.



H.P. 86
Personal Computer

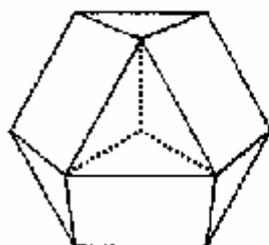
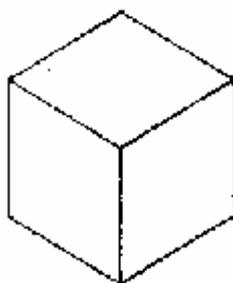
CADD/86-87

COMPUTER
AIDED
DRAFTING

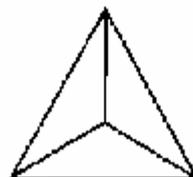
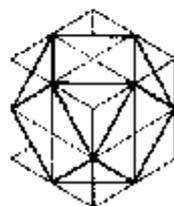
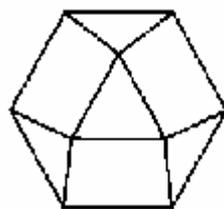
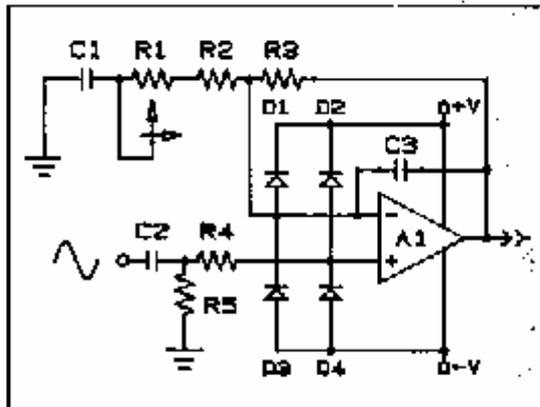
H.P. 87
System Configuration



CADD/86-87 2D DRAFTING



COMPUTER DRAFTING BY TENSEGRITY INC.



CADD/86-87 IS GENERAL PURPOSE...
USEFUL FOR MANY
DESIGN / DRAFTING APPLICATIONS

Tensegrity, Inc. is a registered trademark of Tensegrity, Inc. All rights reserved. This software is provided as a service to our customers. It is not intended for use in any other manner. The user assumes all responsibility for the use of this software. The user agrees to hold Tensegrity, Inc. harmless from any and all claims, damages, or expenses, including reasonable attorneys' fees, arising from the use of this software. The user also agrees to hold Tensegrity, Inc. harmless from any and all claims, damages, or expenses, including reasonable attorneys' fees, arising from the use of this software. The user also agrees to hold Tensegrity, Inc. harmless from any and all claims, damages, or expenses, including reasonable attorneys' fees, arising from the use of this software.

TENSEGRITY, INC.

2424 ADDISON ST. CHICAGO, IL 60618

312-935-9714

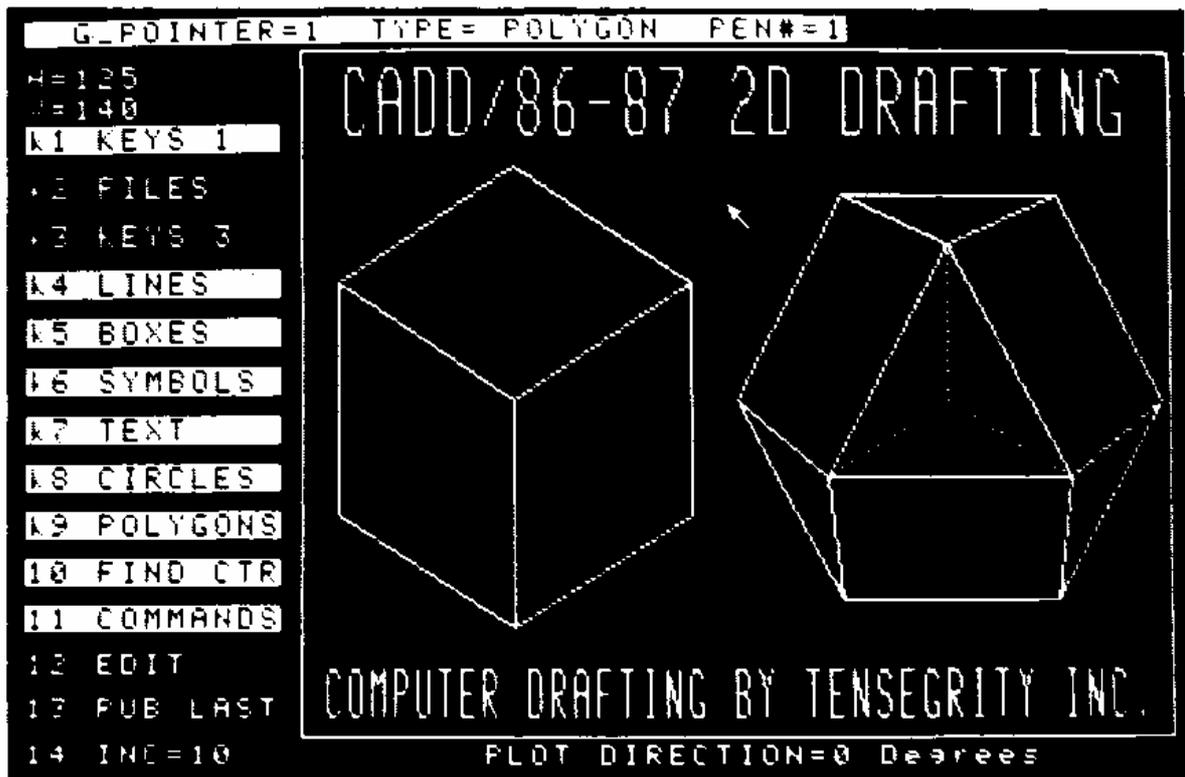
TENSEGRITY, INC.

2424 ADDISON, CHICAGO, IL 60618, 312-935-9714

INTRODUCTION: CADD/87 Computer Aided Drafting & Drawing

CADD/87 provides rapid access to hundreds of high level functions by using both special function keys (softkeys) and global-mode-direct-execute keys. You may work in either metric or inch units and any scale. 1:1 scale relates directly to the actual drawing area of your plotter. By adjusting scale factor, all coordinate information reads out in units corresponding to the actual size of the drafting objective. Any area of a drawing may expand to the full size of your CRT or plotter. All features remain active while drafting in an expanded area (window).

Text options include size, slant, aspect ratio, nine centering positions, and 0 TO 360 degrees of rotation. CADD/87 functions include drawing, editing, redrawing, plotting, windowing, group edits, group moves, group deletes, group replicating, mass storage, digitizing, pen movement, relocating the graphics cursor, searching, pen selection, lines, line-types, vectors, boxes, symbols, text, circles, arcs, polygons, scaling, and more. CADD/87 provides an accurate, time-efficient, and cost-effective solution to a broad spectrum of design & drafting applications.



EQUIPMENT REQUIREMENTS & ORDERING INFORMATION

REQUIRED: HP-86 or HP-87 with at least 160k RAM & Plotter ROM.

PRICE: \$550.00 for single user license. Price includes media cost only program updates and up to one hour phone consulting if necessary.

TERMS: Check in advance or UPS COD. Net 30 is available to qualified customers.

NOTE 1: Volume discounts are available to dealers, schools and industry.

NOTE 2: The word TENSEGRITY is an invention of R. Buckminster Fuller.