

PALO ALTO ICs

810 Garland Drive
Palo Alto, California 94303

An open letter to PET owners:

I love my PET as much as you love yours. But don't you sometimes find it frustrating to create or modify programs?

I've owned my PET for over a year; I'm impressed with its features and its power. Yet when I program I find a great deal of difficulty in modifying, polishing, simplifying and adding new features.

But now, with the BASIC PROGRAMMER'S TOOLKIT™ that I and my associates have created, I'm having more fun, making far fewer errors and I'm completing programs much faster.


I'd like to share my discovery with you.

The BASIC PROGRAMMER'S TOOLKIT has two kilobytes of ROM firmware on a single chip and contains a collection of machine language programs available from the time you turn on your PET to the time you shut it off. Your PET will now have a significant set of additional commands—programmer's tools to enhance your ability to create new programs, to modify old programs, to discover how programs work . . . to enjoy your PET even more. And to do it all so much faster and so much more accurately.

There are no tapes to load or to interfere with any running programs. And it installs in minutes, without tools.

Once you have your BASIC PROGRAMMER'S TOOLKIT in place, you'll know how much more pleasure your programming can give you. I know: I've been using the TOOLKIT for almost two months. I think I love it as much as I love my Pet.

Cordially,
PALO ALTO ICs



Harry J. Saal
President

WARNING
TO ALL PET OWNERS
THE NEW BASIC PROGRAMMER'S
TOOLKIT™ WILL NOT NECESSARILY
MAKE YOU A BETTER PROGRAMMER. NO
CLAIM—IMPLIED OR OTHERWISE—IS
MADE. BUT THE BASIC PROGRAM-
MER'S TOOLKIT MAY VERY WELL
MAKE YOU A BETTER PROGRAMMER.
AT THE VERY LEAST IT WILL MAKE
PROGRAMMING YOUR PET MORE
FUN. THIS IS BOTH AN IMPLICIT
AND EXPLICIT CLAIM.

The Tools in the Basic Programmer's Tool Kit™

VERY SPECIAL COMMANDS. Power up your PET, execute the command **SYS** with the appropriate address and your *new* PET is off and running with these commands:

AUTO This command is followed by a series of optional parameters specifying where you want to enter lines and how far apart you want them. Your PET will automatically respond with a line number prompt. You won't have to enter the line numbers; you won't have to worry about errors of screen editing.

```
AUTO 100,25
100 FOR I = 1 TO 10
125 GOSUB 300
150 ■
```

DELETE Like **LIST**, this command is followed by a range of line numbers . . . and every one of the lines within the range of numbers will be removed instantly from your program. No longer will you have to type a line number, press **RETURN**, type the next line number, **RETURN**, next line. . . .

```
DELETE 200-350
```

```
READY.
LIST 200-350
```

```
READY.
```

RENUMBER Now you can change all line numbers— and all references to those numbers—instantly, as, for instance by evenly spaced increments of 100 or 25 or 10. . . .

LIST

```
10 GOSUB 99
15 PRINT I
16 GOTO 10
99 INPUT J
100 IF J = 0 THEN END
200 I = SQR(J):RETURN
READY
```

```
RENUMBER 100,10
```

```
READY.
LIST
```

```
100 GOSUB 130
110 PRINT I
120 GOTO 100
130 INPUT J
140 IF J = 0 THEN END
150 I = SQR(J):RETURN
READY.
```

HELP How many times have you wanted to scream "HELP!" when your PET couldn't interpret your program and all it would say was: **?SYNTAX ERROR?** Fred no longer: Now just type in **HELP**. The line on which the error occurs will be shown and the erroneous portion of the line will be indicated in reverse video on the screen. Truly a great help in any learning or school situation.

RUN

```
?DIVISION BY ZERO ERROR IN 500
READY.
HELP
500 J = SQR(A*B/C)
```

```
READY
```


TRACE Now you can see precisely the order and sequence in which your program is being executed. You can also stop the program at any point and record the sequence. Type in this command and your PET will keep a record of the line numbers of the last six statements executed. These last six statement numbers will appear in a small rectangular window in the upper right hand corner of your screen.

```
TRACE
READY.
RUN
ENTER YOUR NAME? JIM
HI JIM.
HOW OLD ARE YOU?
```

#100
#110
#150
#160
#175
#200

STEP Again your line numbers are displayed in the upper right hand corner of the screen in this version of **TRACE**. But now your PET executes just one statement and pauses until you press **SHIFT**. Then it proceeds to the next statement.

OFF This command will stop either **TRACE** or **STEP**.

APPEND You've already worked hard to develop a number of programs and, of course, you've saved them on tape. You're working on a program that's now in memory. Just type in **APPEND** "*program name*" and all statements in that program will now follow the program in memory. No need to retype; no opportunity for errors. Save any set of statements or subroutines onto a tape, using the normal **SAVE** command; then recall them with the **APPEND** command and add them permanently to your program.

```
APPEND "INPUT"
PRESS PLAY ON TAPE #1
OK
SEARCHING FOR INPUT
FOUND INPUT
APPENDING
READY.
```

DUMP During or after running a program, this command will display the names and values of all variables used in the execution of your program. You'll know at once what simple variables, arrays and strings you've used and what values you've assigned to each.

```
RUN
READY.
DUMP
A1 = 10
BW = -6.1
C$ = "HI"
AR(1) = 2.3
AR(2) = 7.6
READY.
```

FIND Like **LIST**, this command will show a set of lines. But the **FIND** command is followed by specifying a character string. Those lines, and only those lines, containing a desired character string will be listed on your PET's screen. If you were to type **FIND A\$, 100-500** your PET's screen would display all lines between line numbers 100 and 500 that contain **A\$**.

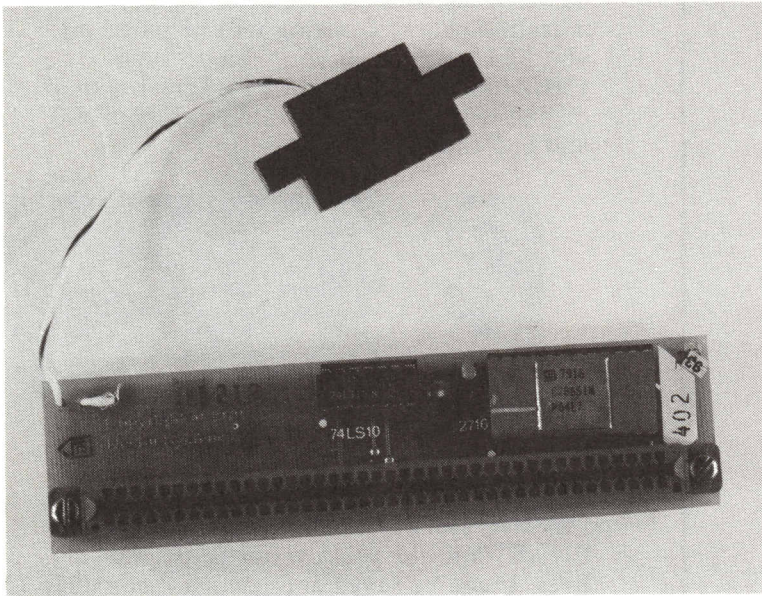
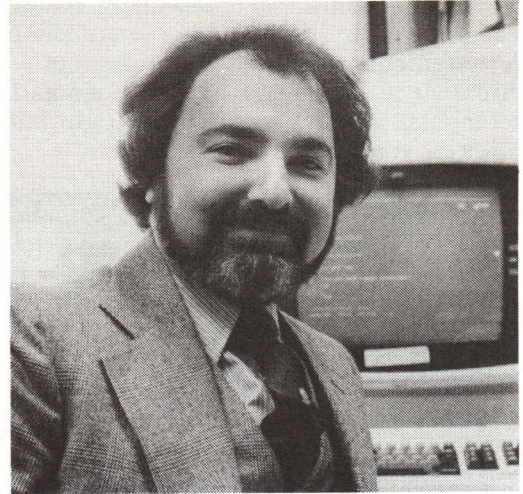
```
FIND A$,100 - 200
110 A$ = "HELD" + B$
180 B$ = MID(A$,1,12)
200 INPUT A$
READY.
```

UNLIST An extraordinary command; very necessary in certain applications. Type **UNLIST** and the text cannot be displayed. Ever. It can be saved and used subsequently on any PET but neither you nor anyone else can ever type **LIST** and see the program. It provides confidentiality when necessary . . . as in answers to a test. . . .

```
UNLIST
READY.
LIST
10
20
30
40
50
READY.
```

HARRY SAAL, with 20 years of computing experience, has become a leading expert in the field of personal computers.

He holds a PhD in Physics from Columbia University, has taught Computer Science at two major universities and has been a researcher and systems programmer with IBM. He is currently president of Nestar Systems, Inc., and its subsidiary, Palo Alto ICs, both devoted to the development of personal computer peripherals and systems. The Nestar System is considered by experts to be the ultimate multiple microcomputer program storage system.



THE BASIC PROGRAMMER'S TOOLKIT™

Two KB of ROM firmware on a single chip. For PET 2001-8, the 8K RAM PET, the printed circuit board shown is attached, with the connectors shown, to the memory expansion interface at the right side of your PET and to the second cassette interface located at the rear right of your PET.

Memory addresses in ROM from \$9800 to \$9FFF and addresses in RAM from \$03E0 to \$03FF are used by the TOOLKIT.

For the 16K and 32K PET, the BASIC PROGRAMMER'S TOOLKIT consists of the ROM firmware chip which is to be installed in the right-most empty socket inside the PET.

The TOOLKIT is custom designed to interface, at your request, with either the Skyles or ExpandaPet memory expansion systems. The TOOLKIT may also be ordered for interfacing with the 8K PET but without connectors. Please indicate when ordering.

A comprehensive user's manual is supplied at no additional charge.
