

# THE PAPER

FOR OWNERS OF THE COMMODORE PET™ PERSONAL COMPUTER

Volume III, Issue 8/9/10

A CSD-LIPS PUBLICATION

\$2.50

## Table of Contents

General Info		2
Volume 4?		3
Background/Foreground		4
Reader I/O		6
Machine Language for Beginners	R Bressler	9
GOTO XXX	D Haluza	16
Assembly Language III	J Fowler	17
Micros and the School Media Center	C Winkler	18
Auto Repeat: Version 3	E Volcheck	19
Review: Matrix Sort Eprom	R Bressler	21
Face Lift for the 'Old' PET	C Werner	22
Am I Being Ripped Off?	B Batcher	23
Merge	J McArthur	25
Shift Lock	J McArthur	27
Defining Programmable Characters	B Batcher	28
Designer	J Feinstein	29
Machine Language Reference Guide	D Haluza	center
Survey Form		center
Fun with PETs and POKes	A Stezsyn	31
Observations on Vol 3, Issue 6/7	R Busdiecker	32
Feedback on Vol 3, Issue 6/7	F Turco	33
Printout Tips	R Bressler	35
Hard Copy for the PET	R Bressler	37
Review: Disk Master	R Bressler	40
PET's Real Cursor	D Haluza	41
Review: MTU's Visible Memory	R Bressler	44
Review: PET/CBM Guide 2nd Edition	R Bressler	45
WordPro IV and the Spinwriter	Dr G Badeau	46
Review: PETSET1	E McDaniels	47
Review: Axiom Ex-801P	A Breed	48
Review: IDPC Forth	R Bressler	49
Review: Spacemaker II	R Bressler	50
PET Magazines	R Bressler	51
PET Memory Map	D Conley	53

**We are ready for Volume IV -- Are You?**

Tell us if you want more by filling out and mailing the survey in the centerfold.

### **NEXT LIPS MEETING**

Monday, May 18  
Harborfields High School  
4:00 P.M.

## General Information

The PAPER is published 10 times per year by Centerbrook Software Designs and the Long Island PET Society at 98 Emily Dr., Centereach, NY 11720. Telephone: (516) 585-2402.

New subscribers will receive all issues of the current volume. Single copy price is \$2 and the subscription price is \$15 for all the issues of the current volume. Subscription orders should be mailed to The PAPER, Box 524, East Setauket, NY 11733.

Third class postage is paid at East Setauket, NY 11733 (Permit #96). POSTMASTER: Mail all address changes to the address above.

The PAPER, Centerbrook Software Designs and the Long Island PET Society are in no way associated with Commodore Business machines. CBM is not responsible for any of the contents of The PAPER unless otherwise noted. PET and CBM are trademarks of Commodore Business Machines.

All readers are encouraged to submit articles of general interest to PET users. Materials submitted must be free of copyright restrictions. All contents of The PAPER are not copyrighted. All articles remain the property of the author and may be reprinted with their permission. When reprinting please include a note stating that the article was originally published in The PAPER.

### Subscription Rates:

USA third class: \$15/volume

Non-USA first class surface: \$20/volume

Payment in check or money order in US funds must accompany all orders. Only prepaid purchase orders will be accepted. All checks should be made out to The PAPER. Sorry, we cannot accept bank or credit cards.

### Advertising

Advertising rates are \$25 per quarter page per issue. Copy must be camera ready or there will be an additional charge. Special rates may be negotiated.

### Circulation

Volume 3 Issue 6/7:

Printed: 1000      USA: 591      Canada: 47      Foreign: 41

### Software Exchange

Software published in The PAPER or distributed through the Exchange is believed to be free of copyright restrictions. It is meant to work on the machine indicated. Many programs were originally designed for the old ROM PETs but efforts have been made to convert them to work on BASIC 2.0 and 3.0. Write for a complete list of programs.

### Staff:

Publisher: Ralph Bressler

Editor: Doug Haluza

Assoc. Ed: Roy Busdiecker

Vic SantaLucia

Exchange : Charlotte Deschamps

Staff writers: Bill Batcher

JoAnn Comito

Gerry Eisner

Jim Fowler

## Volume 4 ?

When we took over from ARESCO almost a year ago we knew we were in for a lot of work. Maybe we didn't realize how much work. I feel we have published a lot of useful information and answered some questions for our readers. The readership has increased by almost 200 and there are more advertisers. Beginning with this issue we will be using a letter quality printer to produce more readable copy and get more information in an issue. On the other side of the coin, I am still dissatisfied with the length of time between issues, the small number of subscribers and the number of typos that seem to crop up. In any case this will be the last issue of Volume 3. I know we really 'owe' one more single issue but we are nearly \$1000 down and I do not feel we can afford another issue. Please remember we started with no money and have been able to publish 3 double issues.

It is now time to decide whether or not there will be a Volume 4. This decision is up to you, the readers. It is hard to make the promises we would like to make. All we can say is that we will try to shorten the time between issues and correct the errors. We will also begin printing single issues instead of doubles as we have due to lack of funds. This means we can be more current with our information and you will get more for your money. We have received many thank you's and words of encouragement but now we really need your support.

We feel that to begin a new volume at least 500 people must renew. To find out how many people would be interested in renewing we have included a survey in the center of this issue. The most important question on the survey is the one that asks if you will renew. We have also included some other questions to help form a reader profile. Please take a moment to fill out the survey, rip out the centerfold and return the survey to us; but SEND NO MONEY NOW. Notice that the other half of the survey is a machine language reference card with a coding sheet on the back. Feel free to photocopy this as many times as you want. Your bill for Volume 4 will be the first issue if we decide to publish it. This issue will come out as soon as 500 people have promised to renew.

Since the costs of producing, printing and mailing The PAPER have risen we must charge \$20 for Volume 4, but if you renew your subscription the price will be only \$15. If you have friends who have not subscribed, they can get the \$15 rate by sending for the back issues of volume 3 for \$15. As always you can help by hiding your copy of The PAPER and only letting out small bits of interesting information to your friends. We need more subscribers! You can also alert potential advertisers who might like to include their products in our pages. Above all you can write articles or letters to help vary our content.

So please fill out the entire survey and indicate whether or not you will renew. We would appreciate your response even if you may not renew. Notice that your name is not on the survey so that all information will be confidential. Remember we're counting on you!

### RENEWAL SUBSCRIPTIONS

<u>United States</u> (second/third class)	\$15
<u>United States, Canada, Mexico</u> (first class)	\$20
<u>Foreign</u> (surface)	\$25
<u>Foreign</u> (airmail)	\$35

All amounts are in US funds. Canadians add 20% if paying in Canadian funds.

Back issues of Volume 3 are \$15

## Background/Foreground

So many small things come to mind I really don't know where to start. A lot of these items are short and not particularly related. So, at the risk of seeming disjointed, here goes.

### Eventually Something Breaks

I obviously think that the PET is the best value microcomputer on the market today. With the 3 for 2 deal for schools noone can match it. Software and hardware is available and the dealer network is growing. My experience shows that after a 2 or 3 week breakin period PETs show good reliability. Still, eventually something breaks and this is where the problems begin.

One of the most important things you can do is develop a good relationship with a local dealer. When things go wrong this dealer may be able to fix the problem right at his shop. Perhaps he might even loan you a machine to use if the repair takes some time. The problem arises when a dealer is unable to repair the PET. At this point the machine or PC board must be sent back to Commodore.

I have a PET which has been in my possession approximately three weeks since its arrival in October of last year. Its been to Commodore three times for relatively minor problems. Some dealers tell me that they have a many as 20 PC boards hung up somewhere for four months. Dealers also tell me they constantly get a runaround when they try to track down their equipment. Commodore's excuse is always that they have recently had a shake up. This wears thin since they seem to have a 'shake-up' on the average of once every two months. This has had many negative effects. Good people get fed up and leave the company. Witness the latest and, perhaps, final departure of Chuck Peddle. Newspapers like the Wall Street Journal (dec. 18, 1980) pick up the story which is bound to discourage investors and potential buyers. Finally, dealers get fed up and drop the Commodore line. Come on Commodore get your act together, drop the excuses, and begin to support the fine machine you produce.

### Shop Around

We all know that when you make a major purchase you should get several prices on the same or similar items and buy the cheapest item with the best quality. It is a little harder when buying computer hardware and a lot harder when purchasing software. Most of us buy programs sight unseen based on a catalog description, magazine ad or a friend's recommendation. I suggest you always try to see the software work or at least get some comments from people that have used it. At the very least try to get an idea of the different products available. Here are some examples.

Bob Baker sells a program which will remove all spaces and REMs from a program. It will also combine short lines without destroying the program. It is called Compactor and sells for \$3. Another company is selling a similar program that does less for \$9.95. Bob tells me has seen this kind of program for as much as \$30.

A recent issue of COMPUTE! contained a File Recovery program which will get back programs accidentally scratched from a disk. It works and costs the price of the magazine and the time to type it in. Similar programs sell for between \$20 and \$50.

Micro recently published UNASSEMBLER by Jim Strasma. This is a symbolic disassembler which, again, might cost as much as \$50. You can have it for the cover price of the magazine.

Someone just told me they paid \$50 for a box of 10 Scotch diskettes. I've seen these for as low as \$30 a box.

These are just some examples but there are many others. Its your money but I encourage you to take the time to shop around.

## Disk Drive Woes

There is nothing more frustrating than having a disk drive crash and take with it many hours of work. Lately, several people have complained about problems with the Commodore disk drive. It's hard to diagnose the problem over the phone but here are some things people may overlook.

Whenever you turn off any hardware connected to the disk drive be sure to open the drive doors. Any small surge of power on the line may cause problems.

When initializing a disk give the command and THEN wait for the disk to start spinning before closing the drive door gently. This helps to seat the disk properly and get the best alignment.

Never pile books, papers or anything else on top of your disk drive. These drives tend to run a little hot and blocking the vents prevents the vital circulation of air.

Be kind to your disks. Store them in a disk box or special protective pages. Never let them lay out in the open on top of the computer or drive. Dirt, dust and stray magnetic fields spell disaster. Remember that the recording is done on the underside of the disk.

I have used almost every brand of diskette from the most to least expensive. I have found no discernable difference. Any failure I have had can be traced back to a power problem or my mistake.

Spend the \$35 or so to buy a head cleaning kit and use it at least once a week.

When you slide a non-write protected disk into a drive there should be an initial click as the write protect switch is depressed. This is followed by another click as the switch comes back up when you push the disk all the way in. With a write protected disk there should be no second click. I have found several drives that have the write protect switch permanently in the non-write protect position. This means that even write protected disks will be written on. See your dealer right away since the problem can be serious but is easily fixed.

Never use the same ID for any of your disks. Keep a record and always use a different to character combination. Bob Baker's Disk Master program is great for this. If you have a disk in each drive and both have the same ID, there is a chance only one BAM may be written. This will destroy valuable data.

Do not use the @ to replace data in a file. This warning has been around a long time but people still do it. Write a new data file and then scratch the old one.

## Newsletters

I have recently received three PET newsletters. I would like to give their addresses and a brief description here.

KEY BITS AND PIECES is published by Jerry Key, 141 Flint Ridge, Gahanna, OH 43230. Jerry's first issue was four pages and summarizes tips from various sources. It is available for a self-addressed stamped envelope and is well worth it. Let's support anyone trying to get the word out.

MIDNIGHT SOFTWARE GAZETTE is printed by the Central Illinois PET Users, 3838 Benton Dr., Decatur, IL 62526. This newsletter contains numerous short reviews of PET software and hardware. It is amazing how much you can learn from a brief, concise review. You can get this one for postage also.

TORPET is from the Toronto PET User's Group, Chris Bennett, 381 Lawrence Ave. Wesy, Toronto, Ont. M5M 1B9 and the last issue was 8 typeset pages. It costs \$10 for the rest of the year. There were several good points mentioned and it is probably a good buy. I assume ALL PET owners in the area are members of this model PET user's group.

## Computers in Color

I just got a brand new copy of Shelly and Cashman's INTRODUCTION to COMPUTERS and DATA PROCESSING published by Anaheim Publishing Co., 1120 E. Ash, Fullerton, CA 92631. This book is not meant as a textbook for a basic high school course but rather for specialized college level instruction. It does have a nice overview of today's computer technology and almost every page has a color picture or diagram. I think it would be a good reference for teachers. It came with a lot of ancillary material but no price.

## Reviews

Any comment about a computer product must contain some bias from the author. Always take reviews with a grain of salt. I hope to be getting some new products to review as I have requested many. If you write a review try to keep it short and to the point and use our format. Please review only those products you have used and those that you have no special interest in.

## Reader I/O

Several people suggested that we simply print letters from readers with minimal editing. We have done so this time around and followed the letter with the name and address of the reader. Our reply follows this.

I have a problem which I would appreciate you addressing. I have a 2022 printer which was purchased prior to Dec. of 1979, and thus contained the old ROM. I ordered the new ROM for the printer and have installed it. However, I am now having difficulty getting my programs to print in upper case. In my business I have occasion to print numerous amortization schedules showing loan payments. Using the old ROM, the print out was in upper case. When using the new ROM the first two lines print in upper case but subsequent lines appear in lower case. The only change in the program is a loop of numerical calculations which should not effect the upper/lower case condition. All runs after this will be entirely in lower case. I have tried just about everything I can think of to get the printer to continue to print out in upper case including a separate print statement to the printer, placing a cursor up before each word, placing a cursor down before each word, running the program in both upper case and lower case, all to no avail. Thomas I. White, Jr., PO Box 952, Manteo, NC 27954

Sometimes the printer will jump into lowercase after formatting data has been sent via secondary address 2. To prevent this use the following code just prior to opening the formatting channel.

```
10 OPEN 4,4,0 : PRINT#4: T=TI
20 IF TI-T < 25 THEN 20
30 OPEN 5,4,2,"***format string***"
```

If you want to do a listing in lowercase on a printer with the new ROM, you can force it into lowercase by typing: 'OPEN7,4,7: PRINT#7: CLOSE7' To get back to uppercase you'll have to reset the printer by turning it on and off. - ed.

In my opinion, what is needed by us non-hardware types is very specific information on where to buy the best hardware for the least money. Also, we need to know how to put the components together into a guaranteed headache free system. For example, I would greatly appreciate knowing how to convert my basic 8K 2001 into an intelligent terminal operating at 1200 baud. But how? Will Hendricks, 5736 W 81st Terrace, Prairie Village, KS 66208

First of all there is really no such thing as a 100% trouble free system. One way to reduce headaches considerably is to only buy products and peripherals made by Commodore. It would also help greatly to develop a good relationship with a local dealer or a good mail order firm who will also answer your questions. We really cannot take the place of a local dealer or computer user's group. If you want to save money or tailor a system to your needs by buying specialized peripherals you'll have to deal with different vendors. Any issue of a good PET magazine lists many of these dealers and their PET compatible products. In an effort to help people like yourself we have an article in this issue which reviews many of the printers available for the PET. Please remember that it is hard for us to review hardware items like this since few vendors are willing to send us printers, modems or disk drives. It would be most helpful if readers who are using various pieces of hardware would write us about them.

To operate your PET as an intelligent terminal you'll need a PET terminal package and a modem. Running at 1200 baud is unusual since phone line only handle about 300. However, to do this you must be sure to buy a 1200 baud modem. This modem will probably use RS-232 serial communication so you will need a 2-way PET-IEEE to RS-232 converter such as the "SADI" from Connecticut microComputer or one of the TNW models from The Networks. To run at lower baud rates means you can use several different modems such as the Novation CAT or Livermore STAR. With these modems you can use a terminal package like McTerm from Madison Computers, PETTERM from FSS or the package from NEECO. Again, all of these will be listed in issues of good micro oriented magazines. - ed

I have a 16K PET with new ROMs and I'd like to expand to 32K. I heard that the big C drilled up some boards to prevent in-computer expansion. My PET doesn't have any unnatural holes, but is there a way to be sure expansion is possible?

My PET has a row of eight 8116's soldered directly to the board above another row of contacts for eight more chips. How do I expand to 32K? Do any chips have to be removed? Where do the chips go? How many chips? What kind? Evan Heit, 3530-15 Long Beach Rd., Oceanside, NY 11572

Commodore began drilling holes in their 8 and 16K boards to prevent dealers from buying the 'smaller' machines, upgrading them to 32K with the chips and selling them for an added profit. Commodore charges \$300 for 16K of memory while the chips only cost about \$30. A markup of \$270 seems extreme for labor but we won't comment here on who was wrong. It seems Commodore has given up mutilating its boards for now.

If your 8K or 16K machine does not have large 1/2" holes drilled in the board it can be expanded to 32K. If you have an 8K with one row of 8K (4108) chips or a 16K with two rows of 4108's they'll have to be removed. You must carefully desolder the chips using a soldering iron with a solder sucker and solder wick. This usually destroys the original chips. You'll need 16 4116 chips to add the 32K and it is best to buy sockets for these chips also. The chips are available for about \$3 and the sockets are about 20 cents. Sockets are nice since a faulty RAM can be removed more easily. After removing the chips simply solder in the sockets and seat the chips.

If your 16K PET has a single row of 4116's, you'll have to remove the solder from the holes on the blank row of chips. Then just solder in the sockets, insert the chips and change the jumpers. To set up the jumpers you can cut or solder them to match the pattern below. There are two sets of jumpers. The first set near the front of the memory expansion connector has six jumpers. The other set is farther toward the back of the PC board and has 10 jumpers. They should look like this:

```

A B C D E F   H I J K L M N P R S
! ! ! ! ! !   ! ! ! ! ! ! ! ! ! !
      !       ! ! !           ! !
! ! ! ! ! !   ! ! ! ! ! ! ! ! ! !
o o x o o x   x x x o o o o x o x   o - open   x - closed

```

You SHOULD NOT try this unless you know what you are doing because the results could be dangerous. - ed.

There is a problem with the DOPEN statement in PET BASIC 4.0. A string variable cannot be used as a file name so that '100 DOPEN#1,A\$' will not work. However, if the filename can be expressed as two parts with one part as a string constant and the other as a variable, things work out correctly. For example, if the filename is ABCDEF you might use:

```

100 A$ = "DEF"
110 DOPEN#1, "ABC" + A$

```

If you really want to put the filename exclusively in a string variable then '100 DOPEN#1,""+A\$' will work. Here the use of a null string allows the entire filename to reside in A\$. Ben Meyer, PO Box 1280, White Plains, NY 10602

Thanks for the hint Ben. Ben is the coordinator of the Westchester (NY) PET User's Group which meets the last Tuesday of the month at various locations. Call him at 914-428-7872 or write: PET User Group of Westchester, Ben Meyer, PO Box 1280, White Plains, NY 10602. We'll be compiling a complete list of PET user's groups. If you have a user's group or know of one, please include them on the survey form in this issue. - ed.

I think that one of the most important tasks of PET user's groups is to exchange software. One reason there is so much Apple software is that their user's groups are VERY active. I would like to get all PET user groups in contact to exchange as much software as possible. I would like to put together a disk library for the PET, which all PET user's groups who have contributed may access for the cost of disks and postage. I am willing to send disks and return postage to each PET user's group to get copies of all the programs they have. I will then compile a list of all the software that I have received and mail this list to all user's groups. I will be willing to copy disks for any group that contributed. I prefer to deal with user groups rather than individuals since I have classes to attend. I realize this will be quite a chore, but by dealing with entire disks rather than single programs I can save time. Philip Chao, PO Box 387, Rochester, NY 14642

We think this sounds like a big job but the Long Island PET Society is willing to contribute. The only reservation is that we must be careful not to distribute copyrighted material. - pub.

I am an instructor in Computer BASIC at Sewanahaka HS. The school recently went Apple and I feel I must sell my PET system. I have the following equipment to sell as a package: PET 2001-8 with Skyles 8K, Skyles Big Keyboard, PR-40 printer, M-65 Ham Morse/RTTY adapter, large software collection, manuals and PET magazines. The total cost of replacement would be well over \$1500 but I will accept the best offer over \$900. I wish to sell it locally so that I can deliver it myself and set it up for the new owner. Allan Adams, 1533 Jasmine Ave., New Hyde Park, NY 11040

We hope someone takes advantage of this good deal. We will include requests like this under letters unless the demand warrants a separate column. - pub.



It is with a great deal of doubt that I start this series of articles. I have just started really programming in machine language and, therefore, am not an expert. However, since I have just started I think I can help you avoid some pitfalls and problems. At a recent user's group meeting, I gave a short lesson and I have also started some simple instruction in my advanced programming course in high school. Both of these efforts have been well received so here goes.

### Why?

I hope some of you got passed the title! Just the mention of machine language is enough to make some people turn the page. I know the feeling since I used to do the same thing. Even if you've gotten this far you may be thinking 'I program in BASIC pretty well. Why should I worry about machine language?'. There are several reasons.

Machine language is MUCH faster than BASIC and some tasks require this speed. Also, programming in machine language gives you a better understanding of how the computer really works. BASIC and machine language routines may be combined; each doing what it is best at. Learning machine language gives you a break from BASIC and is really quite simple if approached step by step.

### How?

The best book now available for 6502 (PET) machine language is titled 6502 Assembly Language Programming by Lance Levanthal published by Osbourne/McGraw Hill. There are other books also available. All the books, however, take the wrong approach to teaching machine/assembly language. They start by talking about binary and hex and one's complement and two's complement and decimal modes and 9 addressing modes and status flags and 150 commands and ... By the time you've read two chapters you're sure you'll never understand any of it. Nothing seems of any practical value either. These books don't start talking about actual things you can do and program until chapter 6!

Most of us learned at least part of our programming by looking at programs and seeing how they work. The approach I want to take is to show some programs and explain how they work and what they do. When the time comes I will introduce and explain what is new.

### Introduction

There are some concepts that are necessary before beginning actual programming. When programming in ML you should consider the PET to be a large collection of empty spaces or "memory locations". Each space can hold ML commands, an operand, or data. The ML commands are called "op codes" and the location of each space is designated by an "address". The "operand" is simply the thing the op code will operate on. It may be data or an address. Each op code also has a "mnemonic" associated with it. The mnemonic is an abbreviation that is usually easier to remember than the op code and tells you something about the ML command. For example, the op code A9 has the mnemonic LDA and this means "load accumulator".

The first real problem we run into is that in ML we must work with base 16 or hexadecimal or just hex. All op codes are in hex as are all the addresses and all the data. In base 16 the digits 0 to 9 represent the same values as in our own decimal system. However, 10 always represents the base, in this case 16. That

means there is a gap between 9 which represents the same value in both bases and 10 which represents ten in decimal but 16 in hex. To fill the gap we use the letters A through F in hex to represent the values 10 to 15 in decimal. So E in hex represents 14 in decimal. Of course, the place values also change! In decimal the situation is as follows:

```

... 1000 100 10 1
      1   5  4  2

```

So the decimal number 1542 means 1 thousand plus 5 hundreds plus 4 tens plus 2 ones. In hex the situation is similar but the places are:

```

... 4096 256 16 1
      0   3  3  A

```

So the hex number 033A means 3 256's plus 3 16's plus A 1's. In decimal this would be 3 X 256 + 3 X 16 + 10 X 1 or 826. At this point this is all the hex you have to know. It might be good to practice a little though.

### Addresses and Contents

All addresses in the PET are represented by 4 hex digits. For example, 033A or A204 both represent different addresses. The contents of each address may only be two hex digits. So an address might contain A9 or EA or 01. The PET is called an 8 bit microcomputer. A "bit" is a binary digit. Binary is base 2 and the only digits available are 0 and 1. This corresponds to the idea that an electronic circuit can either be on or off. Being an 8 bit computer means that the 6502 microprocessor in the PET can only handle information that consists of 8 binary digits at a time. These 8 bits make up one "byte" and this is the amount of information that each memory location can hold. This means the biggest number that a location can hold is 1111 1111 binary or FF hex or 255 decimal. Since each memory location is designated by 4 hex digits, this means that each address is represented by two bytes or 16 bits.

The memory of the PET is divided into RAM and ROM and further divided as to what it does. ROM is permanent memory that we cannot alter. It stores the BASIC language interpreter and PET operating system. The RAM or user memory is what we are concerned with. PET memory is further divided into 255 byte "pages". Page zero, therefore, runs from \$0000 to \$00FF and page 1 starts at \$0100 and ends at \$01FF. (The \$ means a hexadecimal number.) This concept is important since many books speak particularly of page zero memory.

Any part of this may be used but some is easier to use than others. When programming in BASIC all programs begin at \$0400. We will work in the address or memory locations allocated to the PET's second cassette buffer. This is an area of memory that is not used unless the second cassette deck is plugged in or your machine has BASIC 4.0. The addresses run from \$033A to \$03F9.

### The Monitor

To write an ML program we use a facility of the PET known as the TIM monitor. On old (small keyboard) PETs this is a program that must be loaded. On new (big keyboard) PETs the monitor is already in the machine. This monitor will help us see the contents of memory locations, write programs, execute the programs and see the results. To access the monitor type SYS 1024 directly on the screen on new machines or LOAD and RUN the monitor program on old PETs. When you do this you should see something like:

B\*

```
PC  IRQ  SR AC XR YR SP
.; 0401 E62E 32 04 5E 00 F6
```

You are now in the monitor and may use some special monitor commands. The line of information you see is a display of the various "registers". Registers are special memory locations that are used in ML programming. Here is what you have to know about these registers.

#### PC - Program counter

Whenever a program is executed this register 'points' to the next step to be handled. If a program's last instruction was at \$0342 the PC will show \$0343. Each ML instruction effects this register but each may have a slightly different effect. Instructions may increase the register by 1, 2 or 3.

#### IRQ - Maskable Interrupt

When doing simple programming we do not have to worry about this register. It is only displayed on new PETs and contains the starting address of PET's interrupt routine. This routine updates the clock and scans the keyboard.

#### SR - Status register

When the microprocessor performs certain operations various "flags" may be set. If the operation results in a negative number or a zero a particular flag is set to record this. Based on these flags decision may be made to branch to another part of the program. This is similar to IF...THEN.. in BASIC. What these flags are and how to tell their status will be discussed later.

#### AC - Accumulator

The 6502 has one of these special registers. This is where most mathematical operations take place. It is easily examined by the monitor and many instructions can manipulate this register directly.

#### XR and YR - X and Y Registers

These two registers are also used for temporary data storage. They are many times used as counters to control how many times a particular operation is done. In more advanced techniques these registers are treated differently by certain commands.

#### SP - Stack pointer

This important register is a pointer to a set of special memory locations in page one called the "stack". The stack can be used for temporary data storage and is also important in handling subroutines. For now it is not important.

#### Monitor Commands

Whenever the monitor shows a . you may type in a monitor command. The following list of commands includes only those you need for simple programming. Other commands will be mentioned later.

#### M - Display memory

To use this command you might type:

```
.M 0390,03A0
```

and you would see something like:

```
.:0390 00 FF 00 FF 00 FF 00 FF
.:0398 00 FF 00 FF 00 FF 00 FF
.:03A0 00 FF 00 FF 00 FF 00 FF
```

This shows you what each memory location presently contains. 0390 contains 00, 0391 contains FF, 0392 has 00 and so on. Remember that in hex we count 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. When you display these locations on your machine they may have different contents.

### R - Registers

This command displays the registers mentioned before.

### G - Go (Execute)

To execute or run an ML program you might type:

```
.G 0390
```

The number after the G represents the beginning address of your program. After a G command the program will execute. What happens after execution depends on the program but most simple programs will just return to the monitor and show the registers. To see the effect of your program on memory locations you must use the M command.

### : - Write

There is no special write command in this monitor. To write a program, enter data or just change memory location you should move the cursor to the appropriate location displayed on the screen. Now you just type the correct value into the location. When you get to the end of a line hit RETURN and continue to type on the next line.

### A Real Program

We are now ready (finally!) to write a real program. As in any language your first few "words", or in this case, programs will be simple. Our first program will add two numbers from two different memory locations and store the result in the third. The first step is to write the program in assembly language. This means we can use the easier to remember op codes and symbolic addresses rather than real ones. Later we will translate the assembly to ML using op codes and real addresses. This translation is called "hand assembly".

Our program might look something like this:

<u>Mnemonic</u>	<u>Operand</u>	<u>Comment</u>
CLC		* clear the carry
LDA	#\$00	* load accumulator with 0
ADC	LOC1	* add w/ carry the contents of LOC1 to the accumulator
ADC	LOC2	* add w/ carry the contents of LOC2 to the accumulator

```

STA      LOC3      * store accumulator at LOC3
BRK      * return to monitor

```

Perhaps the foggiest notion here is the idea of carry. Actually this is quite straightforward. When we add we sometimes get a carry. In this case the carry we are worried about occurs when two numbers are added and the answer will not fit into one byte. Adding 2 and A is no problem since it results in C but adding FF and 2 gives 101 which is too big to fit in one byte. This would set the carry. The PET has only one addition command and that one is ADD W/ CARRY. Therefore, always CLEAR CARRY before the beginning of an addition. Also, for now, add only small numbers since adding large ones will give an apparent incorrect result as we will see. The LDA or load accumulator is followed by a #00. Remember that the \$ means a hex number. The # indicates IMMEDIATE which means LOAD ACCUMULATOR with the number right after the #, in this case a 0. This technique is used to clear the accumulator before an operation such as addition. Advanced programmers may find this awkward but I wanted to introduce immediate instructions. The ADC and STA instructions are both ABSOLUTE or DIRECT in this case. This means that the operation indicated will involve the address after the operation, in this case the symbolic addresses LOC1, LOC2 and LOC3. The BRK instruction simply returns the program to control of the monitor.

Now we are ready to hand assemble our program. The first thing we have to do is choose where to locate the program and the data it will handle. We already said that we would use the second cassette buffer which runs from \$033A to \$03FF. We'll start at \$0390 and work through \$03F9 to avoid the area used by BASIC 4.0. Let's use \$0390 as LOC1, \$0391 for LOC2 and \$0392 as LOC3. We can start our program at \$0398. One more thing that's important to know. When writing addresses the 6502 requires that we write the "low order" byte first and then the "high order" byte which seems backwards. For example, \$0390 would be written 90 03. Our hand assembled program would then look like this:

```

0398 18
0399 A9 00
039B 6D 90 03
039E 6D 91 03
03A1 8D 92 03
03A4 00

```

We now have to display these locations using M and enter the program by typing the program into these locations. The monitor displays eight locations on a line. Type the program into these locations by moving the cursor to the first location and typing over the numbers already there. Move to the next location and continue typing to the end of the line. Hit return to get to the next line and continue until the program is finished. Now enter the data to be added by typing the numbers we want to add into \$0390 and \$0391. Say we just try using 4 and 6, respectively. Now all that remains is to type G 0398. The program will run in no time and then display the registers. To see the answer stored in \$0392 you will have to use M to display the locations. You should see something like this when all is done:

```

.G 0398
B*
      PC  IRQ  SR  AC  XR  YR  SP
.; 03A5 E62E 30 0A 5E 00 F6

.M 0390,03A0
.: 0390 04 06 0A FF 00 FF 00 FF
.: 0398 18 A9 00 6D 90 03 6D 91
.: 03A0 03 8D 92 03 00 FF 00 FF

```

You should notice that the answer to 4 plus 6 or A is stored in \$0392 and also remains in the accumulator. Also notice that our program ended at \$03A4 with a 00 or BRK. The PC or program counter, therefore, is \$03A5 or one more than the end of the program. The other registers are unaffected since our program did not reference them.

Another important point becomes clear if you look back at our original hand assembled list. Each 6502 assembly instruction may have several ADDRESSING MODES and each different addressing mode has a different op code and is considered a different instruction. Some commands only take one memory location and these are called 1 byte instructions. Examples of these are the CLC and BRK instructions and their addressing mode is IMPLIED which means they require no data or address to operate. The LDA instruction, as explained before, here loads the accumulator with the data immediately following the op code. This makes it a 2 byte instruction and the addressing mode is IMMEDIATE. This is not the only way to use LDA which has 8 different addressing modes and 8 different op codes. The two ADC's and one STA used all require 3 bytes and operate with the address which comes after the op code. This addressing mode is called ABSOLUTE or DIRECT but these instructions also have many different addressing modes when used in different programs. The final word here is that there are 56 mnemonics which can each have from 1 to 9 addressing modes. Since not all instructions have all the addressing modes we have around 150 actual commands. This is a very rich language!

### More Programs

I hope by now that you are ready to begin reading Levanthal's book and typing in some of the programs. He uses a nice progression of programs arranged from simple to complex from which you can pick and choose. There are some problems though with this book that will make using it impossible without some extra knowledge on your part. These problems stem from the fact that the book is written for the 6502 microprocessor in general and not specifically for the PET.

A sample program from this book might be like the one below found on page 4-4. This program simply transfers the contents of one memory location to another and looks like this:

<u>Location</u>	<u>Contents</u>	<u>Instruction</u>
0000	A5	LDA \$40
0001	40	
0002	85	STA \$41
0003	41	
0004	00	BRK

Even if you are just beginning, careful study and comparison to our last program should reveal two major problems. First, the locations used \$0000 to \$0004 are right at the beginning of page zero in the PET. Page zero is filled with important information on the PET and disturbing it may make it take an extended vacation. We will have to change the location of this program to the range \$033A to \$03FF. Second, the LDA command is NOT immediate but it only takes 2 bytes. This addressing mode is called ZERO PAGE since it can only be used with addresses in the \$0000 to \$00FF range. It is really only a shortened form of ABSOLUTE addressing but it drops the leading two zeroes. For example, \$00A0 becomes \$A0 and \$0040 is represented as \$40. If we are going to stay in the \$0390 to \$03F9 'safe' range then we will have to change this addressing from ZERO PAGE to ABSOLUTE. This requires using a different op code and the full 2 byte form of the address. This is quite easy since the book contains a well

organized and alphabetized list of the instructions and the various op codes for different addressing modes. Our converted program might look something like this:

<u>Location</u>	<u>Contents</u>	<u>Instruction</u>
0398	AD	LDA \$0390
0399	90	
039A	03	
039B	8D	STA \$0391
039C	91	
039D	03	
039E	00	BRK

Notice that this has increased the program from 5 locations to seven. Make sure to enter some data in \$0390. At this point I will close and let you try some more things for yourself. Try some programs from the book and look through back issues of The PAPER and others magazines and read the machine language articles you skipped over. Some articles in The PAPER contained short programs you might want to try. If you think you are ready try this assignment. Write a program which will:

- 1) Exchange the contents of two locations. (Use a third location for temporary storage.)
- 2) Exchange the contents of two locations. (DO NOT disturb any other locations.)
- 3) Add \$101 and \$3FF and store the result.

In the second part of this article I will discuss some more advanced subjects. The next article will include the status register, branch and jump instructions, subtraction, some different addressing modes, subroutines and the use of the PET's built in ROM routines. Good luck!

### More People Proofing

The PET, unlike some other micros, has the annoying habit of dropping the user out of a program when a return is hit with no input. There are several ways to avoid this.

To keep the user from dropping out of a program on a null input use INPUT with a prompt. At the end of the prompt add three shifted spaces followed by three left cursors. If the user just hits RETURN the input string will be a shifted space and the user will not be staring at a READY wondering what to do.

```
10 INPUT "ANSWER [3 shift spaces][3 left cursors]";A$
20 IF ASC(A$)=96 THEN PRINT"[up cursor]";:GOTO10
```

Another method to prevent unwanted 'fallout' is to provide a default answer. This is similar to the trick above but does not force the user to type in an answer.

```
10 INPUT "ARE YOU SURE [2 spaces]NO[4 left cursors]";A$
20 IF LEFT$(A$,1) = "Y" THEN 100
30 IF LEFT$(A$,1) <> "N" THEN PRINT"[up cursor]";: GOTO 10
```

## GOTO XXX

by Doug Haluza

Have you ever found that ON...GOTO just won't solve your programming problem? Have you wished that you could GOTO a computed line number? This handy little machine language routine will let you do this on any version PET (old, upgrade or 4.0).

The PET's built in GOTO routine first calls a routine to read and evaluate a constant, checks to see if it's in the range 0-63999 and then executes some code to find the line (if it exists) and actually does the GOTO. This machine language program calls a routine that evaluates an expression, calls another routine to check that it's in the range 0-65535 and then drops into the GOTO routine. It then does the line search and completes the GOTO. By doing this you can call the routine with a variable or an expression and the PET will then evaluate the expression and GOTO that line number.

The program which follows is really three programs. By checking location \$C353 (50003 decimal) it can determine what type of PET it's running on. On old PETs it's \$82, on upgrade PETs it's \$01 and on 4.0 it's \$A0. By testing the value of this location the program sees what version of BASIC it's working with and runs the appropriate part of the program.

```
A0 53 C3   START:  LDA $C353
C9 82      CMP #$82      ;Is it an old PET?
D0 09      BNE NEW      ;No, skip this routine
20 A4 CC   JSR $CCA4    ;Evaluate expression
20 D0 D6   JSR $D6D0    ;Check if 0-65535 and store in 8,9
4C A0 C7   JMP $C7A0    ;Return to GOTO routine
C9 01      NEW:      CMP #$01      ;Is it upgrade?
D0 09      BNE FOUR    ;No, it must be 4.0
20 8B CC   JSR $CC8B    ;Evaluate expression
20 D2 D6   JSR $D6D2    ;Check if 0-65535 and store
4C B0 C7   JMP $C7B0    ;Return to GOTO routine
20 84 BD   FOUR:     JSR $BD84    ;Evaluate expression
20 2D C9   JSR $C92D    ;Check if 0-65535 and store
4C 33 B8   JMP $B833    ;Return to GOTO routine
```

To call the routine from basic use 'SYS(897)expression' where expression has the line number. Putting the SYS address in parentheses leaves the line scanning pointer at the first character in the expression so when the expression evaluation routine is called the expression can be evaluated. A sample program might look like:

```
10 DATA173,83,195,201,130,208,9,32,164,204,32,208,214,76,160
20 DATA199,201,1,208,9,32,139,204,32,210,214,76,176,199,32,132
30 DATA189,32,45,201,76,51,184:FORI=897TO929:READN:POKEI,N:NEXT
40 INPUT"WHAT ROUTINE  "J###";N
50 SYS(897)N*1000
1000 REM ROUTINE 1
1010 :STOP
2000 REM ROUTINE 2
2020 :STOP
3000 REM ROUTINE 3
3030 :STOP
```



## Assembly Language Programming

### III. Style and Debugging

by Jim Fowler

By now, if you have been reading this series of articles, I hope you have tried to write some short assembly or machine language programs. Perhaps you've included your routines or someone else's in your BASIC program. These routines can be called by a SYS or USR command. Anyway the real rewards come when you begin to write many routines or whole programs in assembly language. To do that you may want to use an ASSEMBLER program. The first assemblers available to PET owners were written in BASIC. These assemblers are long and leave only a few hundred bytes for the assembly program. BASIC assemblers are also slow and take a long time to translate the assembly to machine language. Finally, most of these assemblers required the entry of mnemonics in some unnatural way - such as placing them in data statements. The newest assemblers for the PET are written in assembly or machine language. The MAE/TED assemblers from Eastern House Software are loaded into RAM with a version for any PET you might have. The MacroTEA from Skyles Electric Works is resident in ROM. There are advantages and disadvantages in these methods but they should be discussed elsewhere. Both assemblers, as well as, others now available have Editors to make the writing and updating of assembly programs easier. An editor allows you to insert new lines of code or to delete old ones. To use an assembler/editor you must have this program, your assembly language program (the "source"), and the output in machine language (the "object") all in memory at once. Comments are also useful since assembly language is not quite as understandable as a higher level language. This takes a lot of RAM or user memory and may simply be beyond the size of your machine. Don't despair! Substantial programs may be written in assembly on paper and then 'hand assembled'. This simply means you translate the program into machine language yourself and type it direct into the PET using the TIM monitor.

Without an assembler you will receive no "error messages" when you make a mistake. You will have to keep very careful and accurate records as you go along. Beginners should be particularly careful to plan out their routines ahead of time. Even experienced programmers will benefit from some planning before sitting down to do the actual coding. Even with an assembler you will make mistakes in some program and will have to do a lot of debugging on your own. The best way to keep your sanity and develop compact and easily understood programs is to develop a style of programming that is easy to read. One way to do this is to write your programs in little sections that perform one job or make one decision. These may be called subroutines if this same job has to be done in more than one part of the program. Write and debug one section, add it to the main program and check the overall operation before going on to the next. You may have to put in some "dummy" sections of code so the main part can be tested before finishing all the parts and subroutines. I prefer doing all the subroutines first; later putting them together with a main program.

Debugging is best done with the TIM monitor or with the MON program included in the last issue. These programs show you the value of the accumulator, status register, program counter, stack pointer and X and Y registers whenever a BRK (break) instruction is encountered. At this point you can look at memory locations and study the registers to see exactly what the routine has done. If the sequence you are testing is short and does some simple task, debugging is easy. If you write complex code and try to "kill two birds with one stone", you may hit the jackpot, but if you don't get it right the first time you can be in deep trouble. So keep it short!

Finally, it is best to make each section of code so that there is only one entry point and one exit. Don't try to write code that jumps from the middle of

one routine into the middle of another. That is a trap to be avoided! Changes in the program can hopelessly destroy the routine. Therefore, keep it simple! "Short and simple", that is the key.

It isn't hard to write assembly but it takes practice (150 commands, remember). There are usually many ways to do the same thing. Some are short, other longer; some are simple, others complex. It is a lot like writing English prose. You develop a style, sometimes a specialty - something you do better than other programmers. When that happens you are on the road to becoming a real pro.

This series of three articles ought to get you well on the way. Now how about hearing from you? Lets hear about your successes. Have you gotten stuck on a problem? If we share experiences we will also learn faster. Good luck!

### The Microcomputer and The School Media Center

by Charolette Winkler

The school Library Media Center is an ideal place to house the school's microcomputers. Their uses are many and varied which means they are best used if they are situated in a central location. The Media Center is such a place and allows the entire school easy access to the microcomputers.

In the Media Center the microcomputer serves many functions. It acts as a reinforcement tool for some educational areas. Students particularly enjoy certain game programs which also teach some skills. These games are highly motivational and the students enjoy spending time on the computer. Some games reinforce simple library skills like encyclopedia use, library vocabulary, call numbers and so on. Similar games can be created in any subject area. The microcomputer also acts as a remedial tool. Students can improve their understanding of mathematics by choosing from an almost endless supply of programs in this area. Spelling and grammar also lend themselves to the use of the microcomputer. Time on the computer can also be offered as a reward for some other task well done. Games like "Snake" or "Breakout" are fun and students will do a lot for the chance to play. The microcomputer is also a tool to teach programming. Gifted students under the guidance of a District Coordinator, interested teachers and media specialists have created their own programs. Micros may also be used for some clerical chores such as overdue books or simple bibliographies.

The microcomputer is portable and can adapt, with the right program, to many different disciplines and situations. If a school is fortunate enough to have several, one can be put on a cart to move from classroom to classroom. An inexpensive interface can be used to connect the micro to any monitor to allow large group viewing.

All of the above are only possible if there is someone in a District-wide coordinating position. This person must create and obtain software, teach teachers the use of the micro and programming, and teach gifted students. A district should have some goals and objectives for the use of the microcomputer and back them up with a person's time and salary. Media specialists understand that no new technology can be forced upon teachers. The introduction of the microcomputer, like any new teaching tool, must be gradual. It will be the job of the media specialist to become acquainted with this new technology and then to work with teachers toward the use of the micro in their classes.

### Disk Diversions

To load and run the first file on your disk drive you just type LOAD "\*",8. To run another named file you may type LOAD "NAME",8: and then hit the shift-run key sequence. The LOAD command drops to BASIC and ignores anything after the colon. When it returns to BASIC it finds the run command in the buffer and then runs the program it just loaded.

## Auto Repeat Keys: Version Three

by Emil Volcheck  
1046 General Allen Ln.  
West Chester, PA 19380

The saga of auto-repeat keys for the PET continues. I have been interested in the idea since I've been using a couple of programming aids (BEXMON extended monitor and Mike Riley's PAPERMATE) which have the feature built in. So, a couple of months ago, I ran across the original reference to THE PAPER (Vol 1, Issue 10, page 27) in THE TRANSACTOR. I did not learn about the second article in THE PAPER (Vol 2, Issue 6, page 1) until the Fall, 1980, issue of THE new PAPER came a few days ago. So, in November, I had worked up a modification to the original version - along the lines of Sung's article.

Then, as I said, the latest issue of THE new PAPER came with Roy Busdiecker's article (Vol 3, Issue 6/7, page 22) - Version Two. I looked at it with interest; then decided that Version Three could be justified. So, here it is; it offers the following improvements over Version Two:

1. The routine is stored completely in the second cassette buffer, so operation of the first cassette unit is not effected.
2. The routine can be initialized and disabled with the same easy to remember command (SYS 826).
3. The routine has been shortened slightly by removing the excess 00 bytes with zero page addressing when possible. Since the routine is so short, now only 65 bytes, this is not really necessary. There is ample room for the Programmer's Toolkit to operate.

The delay before keys begin to repeat is found at location 855. The delay is set to 20 jiffies or 1/3 second. The rate of repeat is found at location 873. The rate is now set at 4 jiffies or 1/15 second. These factors can easily be changed by POKing new values into these locations or changing them in the program. The listings on the next page include the BASIC source program, a hex dump of the machine language, and a disassembly listing of the program.

The disassembly listing was done with the aid of BEXMON which is a slightly modified version of EXTRAMON. The later is an extended monitor and both allow the D (disassembly) output to be directed to the printer via CMD. The DATA and REM statements in the BASIC program were generated directly with a utility program that I developed. This makes it easier and more error-free to get programs in BASIC to poke the ML program into memory. The BASIC program was generated from a master or model program that I APPEND the DATA and REM statements to from the above, using the Toolkit. So far I have gotten along with the simple assembler which is part of BEXMON. I expect to try DASM from Robert Baker and HESBAL soon. At this time the MAE assembler is too expensive for me! I am anxious to learn how the EARL package from AB Computers works. It reportedly generates fully relocateable code.

Since I am unskilled in the art of assembly language programming, I do not pretend to fully understand how the program operates. Still, I will try to answer any questions that readers have about what I have done.

(Even though Emil claims to be unskilled, I think we could all benefit from the utility routines he has developed. They seem to be handy and work nicely. Will we see another article about them soon? Also, Gene Beals of AB Computers has promised a review copy of EARL so look for a review soon. I use MAE even though I don't do a lot of assembly programming. It is so easy to use and so versatile I feel its worth the price.-pub)

```

..: 033A 78 38 A9 E9 E5 91 85 91 10 REM * AUTO REPEAT NEW ROMS *
..: 0342 38 A9 79 E5 90 85 90 58 20 REM THE PAPER V I, ISS 10, P 27
..: 034A 60 A5 97 C9 FF D0 OC A9 30 :
..: 0352 00 8D 78 03 A9 14 8D 79 40 REM ADAPTED BY EMIL VOLCHECK
..: 035A 03 D0 18 EE 78 03 AD 79 50 REM 1046 GENERAL ALLEN LA
..: 0362 03 CD 78 03 B0 OD A9 04 60 REM WEST CHESTER, PA 19380
..: 036A 8D 79 03 A2 FF 86 97 E8 70 REM 1981 JANUARY 18
..: 0372 8E 78 03 4C 2E E6 EA EA 80 REM 1441 BYTES 65 BYTES OF ML
90 :
100 PRINT "(clr)":CLR
110 PRINT "ML SECOND CASSETTE PRG"
120 FOR I=1 TO 40: PRINT "-";:NEXT
130 ?:?"AUTO REPEAT - NEW ROMS
140 FOR I=1 TO 40: PRINT "-";:NEXT
150 PRINT: PRINT "NOW LOADING"
160 BE=826: EN=887: GOSUB270
170 PRINT "PROGRAM IS LOADED"
180 PRINT "START WITH SYS 826"
190 PRINT "CHANGE REPEAT DELAY BY"
200 PRINT "POKING # OF JIFFIES"
210 PRINT "TO 855. DELAY NOW 20"
220 PRINT "JIFFIES. REPEAT RATE"
230 PRINT "SET BY POKING # OF"
240 PRINT "JIFFIES TO 873. RATE"
250 PRINT "NOW 4 JIFFIES. DISBALE"
260 PRINT "WITH SYS 826 BEFORE"
265 PRINT "USING TAPE":END
270 REM POKE SUBROUTINE
280 READH$: IF H$="ZZ" THEN RETURN
290 D=VAL(H$)
300 POKE BE,D: BE=BE+1: GOTO280
310 REM
320 DATA 120,56,169,233,229,145,133
330 DATA 145,56,169,121,229,144,133
340 DATA 144,88,96,165,151,201,255
350 DATA 208,12,169,0,141,120,3
360 DATA 169,20,141,121,3,208,24
370 DATA 238,120,3,173,121,3,205
380 DATA 120,3,176,13,169,4,141
390 DATA 121,3,162,255,134,151,232
400 DATA 142,120,3,76,46,230,234
410 DATA 234,234,ZZ
420 REM 78 38 A9 E9 E5 91 85 91
430 REM 38 A9 79 E5 90 85 90 58
440 REM 60 A5 97 C9 FF D0 OC A9
450 REM 00 8D 78 03 A9 14 8D 79
460 REM 03 D0 18 EE 78 03 AD 79
470 REM 03 CD 78 03 B0 OD A9 79
480 REM 8D 79 03 A2 FF 86 97 E8
490 REM 8E 78 03 4C 2E E6
.., 033A 78 SEI
.., 033B 38 SEC
.., 033C A9 E9 LDA #$E9
.., 033E E5 91 SBC $91
.., 0340 85 91 STA $91
.., 0342 38 SEC
.., 0343 A9 79 LDA #$79
.., 0345 E5 90 SBC $90
.., 0347 85 90 STA $90
.., 0349 58 CLI
.., 034A 60 RTS
.., 034B A5 97 LDA $97
.., 034D C9 FF CMP #$FF
.., 034F D0 OC BNE $035D
.., 0351 A9 00 LDA #$00
.., 0353 8D 78 03 STA $0378
.., 0356 A9 14 LDA #$14
.., 0358 8D 79 03 STA $0379
.., 035B D0 18 BNE $0375
.., 035D EE 78 03 INC $0378
.., 0360 AD 79 03 LDA $0379
.., 0363 CD 78 03 CMP $0378
.., 0366 B0 OD BCS $0375
.., 0368 A9 04 LDA #$04
.., 036A 8D 79 03 STA $0379
.., 036D A2 FF LDX #$FF
.., 036F 86 97 STX $97
.., 0371 E8 INX
.., 0372 8E 78 03 STX $0378
.., 0375 4C 2E E6 JMP $E62E

```

MATRIX SORT EPROM  
A Product Review

Type: Firmware  
Model PET: Any BASIC 2.0 or 4.0 machine  
Source: MATRIX Software  
315 Marion Ave.  
Big Rapids, MI 49307  
Price: \$55 in single quantity

by Ralph Bressler

I must admit that when I asked for a review copy of this 2716 EPROM I was skeptical. The ad claimed that the sorting algorithm written in 6502 machine language would sort data, integers or strings, over 100 times faster than BASIC. This is a hefty claim but I was interested since I have a mailing list of nearly 1000 to sort. The chip arrived several days after I mailed my request indicating a company trying to please the customer. Their phone number is also included on the literature which means they are willing to answer phone queries.

The package contained the EPROM, 4 pages of installation information and 8 pages of instructions on using the chip. The installation directions were complete from opening the PET to inserting the chip. Good diagrams were also included. The instructions for using the chip were also well written and right to the point. To sort an array of numbers you must give three POKE commands and then a SYS. The POKES tell the chip which array in the program you wish to sort, the dimension you want to sort on and how many keys you want to use. For integers, floating point numbers and single key strings the number of keys is 1. However, with string data you may sort on up to 20 different keys. Each key requires two POKES to tell the chip what position in the string to start at and how many characters to compare. A record might contain LAST NAME, FIRST NAME, STREET ADDRESS, TOWN, STATE and ZIP CODE. You could tell the chip to sort first by zip code, then by last name within each zip code, then by first name within each last name. This can go on for up to 20 levels. This may sound complex but, believe me, the instructions make it clear by giving examples and a summary table.

So the instructions make it sound great, but what about speed? You may find the following numbers hard to believe but they are real. If you don't believe them, buy the chip yourself.

<u>Data type</u>	<u># of keys</u>	<u># of items</u>	<u>Time (sec.)</u>
integer	1	100	.50
integer	1	1000	7.99
string (14 char.)	1	100	.43
string (14 char.)	1	1000	6.92
string (30 char.)	3	500	4.60
string (120 char.)	3	200	2.84

All these times are averages based on at least three trials. The last example is from a real application in sorting my mailing list. I keep The PAPER's mailing list sorted into separate files by zip code. I have 8 fields per record and a maximum of about 200 in any one file. To sort the data I first concatenate all fields into one string. After concatenation I sort by last name, first name and town. The biggest limitation here is the number of strings that will fit in memory. This may require some creative use of memory and disk for your application.

I would recommend this EPROM to anyone who needs to sort information fast. The chip also contains a screen dump as a bonus. Documentation and company support is very good.

## Face Lift for the 'Old' PET

by Chris Werner  
PO Box 61795  
Sunnyvale, CA 94088

The addition of a second, large keyboard to the PET (small keyboard version) has been the subject of numerous articles. All of these keyboard additions have one thing in common, they are an addition or extension to the existing keyboard. They come installed in a box, the width of the PET, connect to the existing keyboard cable and sit directly in front of the PET. They range in price from \$125 to \$250. I consider this type of add-on bulky and it defeats the original purpose of keeping the PET together in one piece.

The keyboard that comes with the 'new' PETs is very attractive since it is an integral part of the PET and has large, good quality keys which are clearly labeled. The obvious solution here is to bite the bullet, trade in your old PET, and shell out the bucks to buy a new one. But this means giving up a machine that is paid for, probably works great and may have extras like added memory, a Toolkit and so on.

So, back to square one, you like your PET but would rather have a large keyboard. You might want to do as I did. Go to your favorite computer store that deals in Commodore products and purchase the new PET keyboard and PET upper case. The keyboard costs \$80, the case is \$40 and the label \$2.50. The large keyboard is a direct replacement (electrically) for the smaller keyboard right down to the polarizing pin in the interconnect cable.

### INSTALLATION

First, mount the keyboard in the new case using 6-32X3/8 machine screws with lock washers. Also attach the decal or label to the new case.

Next, open up the case of the PET by removing the two screws on each side. Swing down the support on the left side to hold the case open. Unplug the cable from the monitor to the main circuit board. Disconnect the two brown, twisted wires that connect the monitor to terminals 7 and 8 of the transformer. To do this you must use a soldering iron. You'll need your spouse or a friend to hold the monitor housing as you remove the four 10-32 nuts that hold this assembly in place. Now unplug the cassette deck and keyboard cables from the main circuit board. Remove the rivets securing the case to the base hinge. Use a 3/16" drill bit to partially drill each rivet and then knock them out with a center punch. Make sure that you recover each rivet to prevent electrical shorts. This frees the case for removal.

Mount the new case in the same position using 6-32X3/8 machine screws instead of rivets. Mount the monitor housing assembly on the new case using the four 10-32 nuts and reconnect the two brown wires to terminals 7 and 8 on the transformer. Reconnect the monitor and keyboard cables as before. Remove the cassette recorder from the old case. Connect the recorder to the same connector as before and set it next to the PET with the cable draped under the case. Otherwise, you may connect the recorder to the second cassette port at the rear of the PET.

When you close the PET and replace the four screws which hold the case closed you have completed the face lift. All of this should take about twenty minutes. Remember that you still have an 'old' PET which still requires shifting for lower case unlike the newer models. Depending on the exact model of your PET you may be able to correct this 'problem'. Ask your dealer about replacing the character generator ROM which controls what characters are created when the keys are hit.

## Am I Being Ripped Off

by Bill Batcher

As a developer of educational software, I recognize that there are different programming styles. I frequently exchange programs with others and enjoy seeing how someone else has written a particular type of math drill or spelling program. While I might not have written it the same way, I still am grateful for any new title that can help my pupils. Like all programmers, I will frequently edit such programs and put in the features I like to have.

When I am purchasing software, however, I expect a complete and professional program. Such programs should not have to be edited or improved by the buyer. There are several clues I have learned to look for in a program which cause me to question the wisdom of my purchase. Of course, I can't do much after I've already spent my money, but I can be wary about buying from the same vendor or programmer in the future.

Clue 1 - The program won't load. I once purchased a package of 6 games from a very famous software company that was one of the pioneers in the industry. Although I had access to 60 different PETs of all different types, I could not get any of the games to load. I returned the tape and eventually got a replacement. However, this one also would not load. After several such exchanges, the fifth copy I received finally loaded. By this time I had spent so much time, postage and mental energy that I doubt I will order from them again. I know it is possible to reliably mass produce tapes. I have never gotten a copy of Cursor that would not load.

Clue 2 - The program fails to clear the screen when run. After typing RUN and pressing RETURN, the word RUN, SEARCHING, and FOUND remain on the screen. Of course, this may really be judging a program by its cover, but I am highly suspect of commercial programs that fail to have a simple "clear screen". This omission displays a general disregard for the average user or an oversight which may indicate something about the rest of the program. Non-programmers can be easily distracted by this problem.

Clue 3 - No directions are given. Unless the programmer has exhausted the available memory in a clever or complex program, there is no excuse for not including directions in the program. Little pieces of paper with two paragraphs of instructions quickly get lost. Preferably, the user should have the option of seeing the directions or skipping over them. After using the program several times it becomes boring having to sit through the directions.

Clue 4 - Information disappears before it can be read and understood. Some programmers seem to think they can predict how long it will take all users to read an amount of text. This attitude fails to take into account my distractions and my need to reread sentences. The simple solution is a line like PRESS C TO CONTINUE at the bottom of the screen, so I can read the text at my leisure. And if it does say PRESS C, I don't want it to:

- a) continue when any key is pressed
- b) have to press RETURN after the C

Clue 5 - Text or problems scroll off the screen. A program should anticipate when the screen is filled with text, problems or pictures and should clear the screen and start at the top again. Programs on a video screen should not look like they are running on a teletype or printer.

Clue 6 - The program requires pressing 0 for no and 1 for yes. This technique was once necessary when computers would not handle strings many years ago. Programmers who use this technique probably have not learned anything else new either.

Clue 7 - The user is never told to press anything. It is unbelievable that

some commercial software requires no user interaction. A series of programs I have seen asks the user to write the answers on a sheet of paper and then shows the answers at the end for comparison! This is using a sophisticated microcomputer as little more than a ditto sheet.

Clue 8 - Subsequent runs show the same questions in the same order. Any good CAI or drill program should be randomized. The order of the questions should be random and multiple choice answers should vary. The relative ease with which this is accomplished leaves no reason for not doing it.

Clue 9 - Little use is made of the PET's excellent graphics. Checker games that use the letter X and O for the checkers and the '.' for an empty space are disgraceful. This feature combined with scrolling boards make it clear little was done to convert an old teletype game. It would have been simple to at least have used the PET's shifted Q and W for the checkers. Homing the cursor and reprinting the board in the same place would have given the impression of movement.

Clue 10 - The program contains errors. Educational software should at least be accurate and free of errors. I've seen commercial software with misspelled words and, worse yet, incorrect concepts. One program that purports to teach analogies includes errors like:

dog : cat :: kitten : puppy

Clue 11 - The program doesn't tolerate user errors. An accidental RETURN without an input, drops me out of the program. Then, if I don't realize I am out and type my answer, 50, I replace a line of the program. When I'm asked for a numeric solution the program shouldn't respond with REDO FROM START if I accidentally type A. The typical user assumes START means the beginning of the program and has no idea what to do.

Am I getting ripped off? Are we receiving less for our money than we should? If the programs you buy contain one or more of the 'clues' above the answer is probably 'yes'. If a program fails to make use of the many excellent and unique features of the PET the answer is 'yes'. If the software seems to be copied from a school workbook the answer is 'yes'. Why, then, is this low quality product still on the market? How is it possible for companies to exist peddling inferior merchandise? How can we force software developers to improve the quality of software? The answers to these questions are not always easy. If people buy a product then it will continue to be produced and the companies distributing it will thrive. When the hardware arrives at a school, teachers and administrators panic and will buy anything. To improve program quality we must be better consumers! Refuse to buy programs without knowing something about them. Complain if they do not live up to your expectations. Refuse to buy from companies that produce inferior products and do not support their software. Lastly, support good programs and programmers by encouraging others to buy the product. Do not 'loan' or 'borrow' copies of educational software. This single fact is the biggest disincentive to good programmers producing a good product.

### Keeping Track of Program Versions

Ever wonder which version of a program was your latest? Do you know how long you have been working on a program? To solve both these problems try SAVING your program (on tape or disk) with TI\$ added to the end. When you start working on a program for the first time set TI\$ to 0. Each time you start to work on the program again set TI\$ to the time on the last version. Keep resaving the program and your problem is solved.



## Merge

by James F. MacArthur  
3240 Blundell Rd.  
Falls Church, VA 22042

Here is a MERGE program for the old ROM 8K PETs. The essential part of MERGE consists of lines 280 through 440. Lines 10000 and up are needed only if you have difficulty making it work. These lines do a checksum on each string, as an aid in locating errors, and should be deleted after it is debugged.

MERGE makes it possible to load a program and "dump" it to high memory so that another program may be loaded, after which lines from the first program can be listed or merged into the second program. The total length of the two programs must be short enough to fit into 8k of memory along with MERGE which occupies the last 512 bytes. The first step is to load and run MERGE, then load the program containing the lines you want to merge. Next type the following command: ?USR(D)

When you hit the RETURN key the screen will clear and NEW will appear in the upper left hand corner with a READY below it. The cursor will be on the next line down. Your second program, the one into which lines are to be merged, can now be loaded. To list lines from the first program, type the following command:

?USR(first line number) where "first line number" is the first line to be listed. For example, to list starting at line 149 you would type: ?USR(149). This is equivalent to typing LIST 149- for a regular program. The listing can be halted with the STOP key. The lines on the screen can now be edited into the program in the usual manner. Just place the cursor over the first line to be merged and then hit RETURN for every line to be added. A complete merge may be effected by continuing this procedure.

If you want to merge a large quantity of lines, there is an easier way, since MERGE also has an automatic mode of operation. To accomplish this type the command: ?offset number, USR(first line number) where "first line number" is the first line to be merged and "offset number" is the number to be added to the merged line numbers. This offset may be positive, negative or zero. For example, if the first line to be merged is line 573 and you want to add 2000 to the line numbers, you would type: ?2000,USR(573) which would result in the first merged line having the number 2573. Note that although line numbers can be changed, numbers contained within lines, such as GOTO statements, will stay the same. You will have to seek out these lines and edit them manually.

Anyway, when you hit RETURN the screen will clear, and you will see one line at a time appear and disappear from the top of the screen. To stop the merging operation, wait until the last line you want merged has flashed on and off the screen and hit STOP. If you want to merge all the way to the end, wait until FINISHED appears at the top of the screen.

Occasionally, there may be times when the automatic merging stops by itself without reaching the end. This happens when a line is too long to merge and is usually caused by the line number having more digits than it had originally. To restart the merging operation, simply shorten the line to 79 characters or less (changing PRINT to ? and deleting blanks works well). Hit RETURN and move the cursor down to the preprinted SYS7876 which may have to be retyped and hit RETURN. Whenever the cursor is visible you are in BASIC and free to do anything you normally could.

That just about covers all there is to know about using MERGE, but the following information may be helpful at times. Bytes 134 and 135, which determine how much memory is available, are controlled by MERGE, to protect the program being merged, as well as, MERGE itself. When in the automatic mode, it checks to see how much free memory is available prior to merging each line. If the amount of free memory is about to become too small, it resets these bytes so that only MERGE is protected. In this way the first program is unprotected only

if it is necessary to continue the merging operation. Therefore, these two bytes may or may not have been reset during the merging process, depending on how long the programs were. If you get OUT OF MEMORY ERROR when trying to run or edit your program after using MERGE just type: POKE 134,0:POKE 135,30. This will remove the protection from the first program which is probably no longer needed anyway. If you still get OUT OF MEMORY ERROR, POKE byte 135 to 32 instead of 30. This will make the remainder of memory available but will destroy MERGE.

This version of MERGE can be used on old ROM PETs only but should work for any memory size. However, only the first 8k of memory is usable. I have not completely checked out the changes required to adapt the program for a larger memory but will make this information available as soon as it is ready.

```

22 PRINT" JAMES F. MC ARTHUR
23 PRINT" 3240 BLUNDELL RD.
24 PRINT" FALLS CHURCH, VA. 22042
30 PRINT"␣ JULY 15, 1979"
40 PRINT"␣ LAST UPDATE JAN. 18, 1980"
50 PRINT"␣STAND BY.":PRINT
55 PRINT"THE MACHINE CODE IS BEING LOADED."
280 IF ST<>0 THEN PRINT"␣STATUS ERROR - LOAD AGAIN.":END
290 POKE134,0:POKE135,30:CLR:POKE1,0:POKE2,30:DIM M$(19)
300 M$(0)="38A50DE944D049851CE57C8DFE1F8586A91EE57D8DFF1F"
301 M$(1)="69038587A21D861DA57C851AA67DC861BA0FFB11A911C88"
302 M$(2)="C0FFD0F7C61BC61DA51DC903D0EDC88C0004A2058E0D02"
303 M$(3)="BDE01F9D0E02CAD0F76020DAD6A5B38595A5B48594A900"
304 M$(4)="A885792078D2A40BC0ABD003E8A9FF48E8E50A38E930090A"
305 M$(5)="B01786794820B4D920E1DA68A8A9002078D22044D7A679D00F"
306 M$(6)="6885B520DAD6A5B38578A5B48577A2098E0D02B0D71F"
307 M$(7)="9D0E02CAD0F760A004A99120D2FF88D0FAA92091E0C810FB"
308 M$(8)="A201A90486758576A575186D0FE1F8575A5766DFF1F8576"
309 M$(9)="20E1FFF01AA000B175851CC81175D010A479F00BA2F4"
310 M$(10)="B0FE1E20D2FFE8D0F760B175851DC818A594F175C8A595"
311 M$(11)="F17590034CC31F88B175657785B2C8B175657885B1A579"
312 M$(12)="F01338A586E57CA587E57DD008A9008586A91E8587A29038"
313 M$(13)="201B0B20B1DCE8BDFE00D0FA8542A9209D0FE00A003C8E8303E"
314 M$(14)="B1759D0FE00C922D002E642A5422901D029B1751025C9CBB021"
315 M$(15)="8444A000C8B990C010FA0EFE0030F5CAE8C8B990C0297F"
316 M$(16)="9D0FE00B990C010F1A444B175D08BEA9009D0FE00A479F00B"
317 M$(17)="A2FDB0E91E20D2FFE8D0F7A000C8B9FE0020D2FFC90DD00F5"
318 M$(18)="A479F018A2F7BDF21E20D2FFE8D0F7A9028D0D02A90D"
319 M$(19)="8D0F028D1002A61CA51D86758576B005A479F001604CC41E"
330 S=7680:FOR I=0 TO 19:FOR N=0 TO LEN(M$(I))/2-1
340 J=ASC(MID$(M$(I),2*N+1,1))-48
350 K=ASC(MID$(M$(I),2*N+2,1))-48
360 IF J>9THENJ=J-7
370 IF K>9THENK=K-7
380 POKE S+N,16*J+K:NEXT N:S=S+N:NEXT I
420 P$="SYS 7851"+CHR$(13)+"␣NEW"+CHR$(13)
430 P$=P$+"␣SYS 7876␣FINISHED.␣"
440 FORN=1 TO 38:POKE 8151+N,ASC(MID$(P$,N,1)):NEXT
490 POKE59468,14
500 PRINT"␣MERGE IS READY TO USE."
505 PRINT"␣AFTER LOADING YOUR FIRST PROGRAM, TYPE"
510 PRINT"?USR<D> TO DUMP IT TO HIGH MEMORY. THEN"
515 PRINT"IF YOU WISH YOU CAN LOAD ANOTHER PROGRAM
550 PRINT"TO LIST THE FIRST PROGRAM, TYPE ?USR"
555 PRINT"FOLLOWED BY THE FIRST LINE NUMBER YOU"
560 PRINT"WANT, IN PARENTHESES. EXAMPLE: TO LIST"
565 PRINT"STARTING AT LINE 149, TYPE AS FOLLOWS:"
570 PRINTTAB(15);"?USR(149)

```

```

575 PRINT"TO MERGE LINES FROM THE FIRST PROGRAM,"
580 PRINT"TYPE A QUESTION MARK FOLLOWED BY THE"
585 PRINT"NUMBER YOU WANT TO ADD TO THE MERGED"
590 PRINT"LINE NUMBERS, FOLLOWED BY A COMMA,"
595 PRINT"FOLLOWED BY USR, FOLLOWED BY THE NUMBER"
600 PRINT"OF THE FIRST LINE YOU WANT MERGED, IN"
605 PRINT"PARENTHESES.  EXAMPLE: TO MERGE"
610 PRINT"STARTING AT LINE 573, WITH THE MERGED"
615 PRINT"LINE NUMBERS INCREASED BY 2000, TYPE:"
620 PRINTTAB(12);"?2000,USR(573)"
10000 REM ** DELETE LINES 10000 & UP AFTER DEBUGGING
10010 DATA 2734,2767,2678,2664,2782,2824,2536
10020 DATA 2774,2624,2485,2697,2582,2715,2910
10030 DATA 2786,2668,2673,2878,2582,2682,2427
10040 PRINT:PRINT:FOR I=0TO19:C=0:FOR N=1 TO LEN(M$(I))
10050 C=C+ASC(MID$(M$(I),N,1))
10060 NEXT:READ CC:IF C<>CC THEN PRINT"M$(I);I;" BAD"
10070 NEXT:C=0:FOR N=1 TO 38:C=C+ASC(MID$(P$,N,1))
10080 NEXT:READ CC:IF C<>CC THEN PRINT"P$ BAD"
10090 END

```

### Shift Lock

by James F. MacArthur  
3240 Blundell Rd.  
Falls Church, VA 22042

SHIFT LOCK is a program which allows typing of lowercase letters on old ROM PETs without holding down shift. When the program is activated, the left shift key acts as the shift lock key with the right shift key acting as the shift lock release. The program works a little like the repeating keys program that appeared a while ago in The PAPER.

```

0 POKE59468,14:PRINT"SHIFT LOCK BY JAMES F. MCARTHUR
5 S$="78AD190249C88D1902AD1A0249E58D1A0"
6 S$=S$+"25860AD2502C914D00034E2502"
10 S$=S$+"CE10E8AD12E8CD12E8D0F8EE10E8"
15 S$=S$+"A200C9FED002A201C9DFD002A200A"
16 S$=S$+"D0302C91BD002A2000E6603A4E288C000"
20 S$=S$+"D0168AF013AD0302C949F00CB1E0F0"
21 S$=S$+"08C91BB004094091E0C88C83034C85E6"
25 FORN=0TO104
30 J=ASC(MID$(S$,2*N+1,1))-48
40 K=ASC(MID$(S$,2*N+2,1))-48
50 IFJ>9THENJ=J-7
60 IFK>9THENK=K-7
70 POKE 826+N,16*J+K:NEXT
105 PRINT"SYS826 ENABLES AND DISABLES"
110 PRINT"THE SHIFT LOCK FUNCTION."
115 PRINT"WHEN ENABLED, THE CURSOR BLINKS"
120 PRINT"TWICE AS FAST AS NORMAL."
125 PRINT"SHIFT LOCK IS PERFORMED BY THE"
130 PRINT"LEFT SHIFT KEY. IN THIS"
135 PRINT"MODE, THE LETTER KEYS ARE SHIFTED."
145 PRINT"SHIFT RELEASE OCCURS WHENEVER THE"
150 PRINT"RETURN OR RIGHT SHIFT KEY IS PRESSED."
155 PRINT"BE SURE THE CURSOR IS BLINKING NORMALLY"
160 PRINT"BEFORE DOING ANY TAPE OPERATIONS.":END

```

## DEFINING PROGRAMABLE CHARACTERS

by Bill Batcher

So you got yourself a 2022 or 2023 Commodore printer. Congratulations! Now you can list your program on hard copy, you can generate catalogs, print notices, etc., etc. And that letter enhancing really jazzes things up. But have you ever used secondary address #5? With your new printer you can design your own characters! It may look a little cumbersome at first but it's worth taking the time to learn about it. In foreign languages you can create special punctuations and accents. However, only one character can be defined at time so it is difficult to write an entire character set like a Greek or Russian alphabet. Still, individual letters can be done. You can also combine the defined character with the printer enhanced character.

I think I'll be using the user-defined characters, mainly for decorating my printouts. The owner's manual for the printer explains the mechanics of doing the defining but you'll need some ideas for interesting characters. First, I'll explain the theory and then show you some examples of interesting characters and border designs.

The explanation of programmable characters is found in the Commodore printer manual on pages 29 to 31. The characters produced by the printer are in a 7 vertical by 6 horizontal matrix. Set up a grid that is 7 by 6 and write the numbers 1, 2, 4, 8, 16, 32, 64 along the right hand side starting at the bottom. Now plan out your character by darkening in the squares you want. Now total each column by adding up the numbers from the right hand side for each square you darkened. The six numbers you just calculated are data which when passed to the printer using secondary address #5 will set the printer for your character. Suppose the total for the columns were 28,34,65,54,34 from right to left. The program to define and print this character should look like this.

```
10 DATA 28,34,65,54,34
20 OPEN 5,4,5
30 FOR I=1 TO 6: READ A: A$=A$+CHR$(A): NEXT
40 PRINT#5, A$
50 OPEN 4,4
60 PRINT#4,CHR$(254)
70 CLOSE 5: CLOSE 4
```

In this program line 10 is the data that will define our character. Line 20 opens file #5, to device 4 which is the printer and uses secondary address 5. Line 30 reads in the DATA and builds the defined character in A\$. Line 40 then sends the character to the printer where it is stored in the printer's memory. (Remember, these peripherals are smart!) Line 50 then OPENS the printer for printing. Line 60 prints our character on the printer. Whenever you want to print the defined character you must use CHR\$(254). Line 70 cleans things up by CLOSING all files. At times you may have to enhance a character to make it look better. To do this send a CHR\$(1) as many times as you want to enhance the character before printing it.

Now here are some suggestions for characters. Since this article is produced with WordPro III actually printing the characters is impossible.

American flag (enhanced twice)	125,125,125,85,85,85
Candle (enhanced once)	2,5,3,95,1,1
House (enhanced twice)	31,55,124,55,31,5
New York (enhanced once)	24,24,56,124,126,2
Infinity (enhanced once)	24,36,24,36,21,0
G Clef (unenhanced)	14,17,127,37,10,4

Eyes (enhanced once)	38,73,73,75,73,38
Greek (unenhanced)	23,21,29,1,31,16
Brotherhood (unenhanced)	17,18,124,124,18,17
Railroad (enhanced once)	126,36,126,36,126,36
Music (enhanced once)	2,5,5,126,32,16
Waves (enhanced once)	4,2,33,25,22,8
Braid (unenhanced)	95,110,117,123,119,47
Derby (enhanced twice)	12,30,60,28,14,12

### Designer

by Jeff Feinstein

I am a junior in high school and wrote the following program after my teacher showed my how to do programmable characters on the PET printer. The program allows three different choices. First, you may choose a predefined character already stored in DATA in the program. Second, you may input the six character programming numbers like the ones given in Bill Batcher's article. Lastly, you may choose to design your own character. The program will print a 6 x 7 grid on the screen and allow you to 'light up' various points to define the character.

When you send the character to the printer you may choose the number of times to enhance it and how many characters to print. If you find a new character you like you may enter it in DATA statements in the program. This will also require some other small program changes. I hope this helps you in designing characters on your printer.

(We know the program is long but it is very useful for what it does. If you want a copy and don't want to do the typing, you should know that it will be available on the program exchange.-pub.)

```

100 REM***JEFF FEINSTEIN***
110 POKE59467,12:FORI=1TO7:READBY$(I):NEXT
120 DATA 64,32,16," 8"," 4"," 2"," 1"
130 DATA FLAG,125,125,125,85,85,85,CANDLE,2,5,3,95,1,1
140 DATA HOUSE,31,55,124,55,31,5,NEW YORK,24,24,56,124,126,2
150 DATA INFINITY,24,36,24,36,21,0,6 CLEF,14,17,127,37,10,4
160 DATA EYE,38,73,73,75,73,38,TANK,7,62,63,62,23,22
170 DATA GREEK,23,21,29,1,31,16,TINY MEN,17,18,124,124,18,17
180 DATA RAILROAD,126,36,126,36,126,36,MUSIC,2,5,5,126,32,16
190 DATA BRAID,95,110,117,123,119,47,DERBY,12,30,60,28,14,12,*
200 DATA DERBY,12,30,60,28,14,12
210 :
220 PRINT"***DESIGN-A-CHARACTER***"
230 PRINT"DESIGN, INPUT, PREDEFINED OR QUIT?"
240 GETSP$:IFSP$=""THEN240
250 IFSP$="1"THEN390
260 IFSP$="D"THEN420
270 IFSP$="Q"THENPRINT"GOODBYE!":END
280 :
290 PRINT"PREDEFINED CHARACTERS"
300 RESTORE:C=0:FORI=1TO7:READA$:NEXT
310 READC$:IFC$="*"THEN340
320 C=C+1:PRINTCHR$(C+64)" C#"
330 FORI=1TO6:READA$:NEXT:GOTO310
340 PRINT"WHICH CHARACTER? "
350 GETCH$:IFCH$=""THEN350
360 RESTORE:FORK=1TOASC(CH$)-64:READCH$:FORI=1TO6:READA$:NEXTI,K
370 READCH$:FORI=1TO6:READA$(I):NEXT:GOTO730

```

```

390 REM INPUT PRG CHR
400 PRINT"INPUT THE VALUES FOR EACH CHARACTER:"
410 FORI=1TO6:PRINT"VALUE #":INPUTA(I):NEXTI:GOTO730
420 PRINT"      A B C D E F
425 PRINT"
430 FORX=1TO7
435 PRINT"
440 PRINT("BY$(X)") "STR$(X)" | | | | | |
450 PRINT"
455 NEXTX
460 IFA#=""THEN550
470 PRINT"USE OLD CHR ?"
480 GETSP#:IFSP#=""THEN480
490 PRINT"
500 FORI=1TO6:B(I)=A(I):A(I)=0:NEXT:G=81:IFSP#="N"THEN550
510 FORI=1TO6:C=0
520 FOR J=6TO0STEP-1:C=C+1:IFB(I)-2↑J<0THEN540
530 Q#=CHR$(I+64):Q=C:GOSUB860:B(I)=B(I)-2↑J
540 NEXTJ,I
550 PRINT"          "TAB(28)"O/P?
560 GET PE#:IFPE#=""THEN560
570 IFPE#="P"THEN730
580 PRINT"          "TAB(28)"DESIGN?
590 PRINT"          "
600 GETA#:IFA#=""THEN600
610 IFASC(A#)=13THEN730
620 IFA#>"F"ORVAL(A#)>7THEN600
630 PRINTA#:
640 GETB#:IFB#=""THEN640
650 IFASC(B#)=13THEN730
660 IFVAL(A#)=0ANDVAL(B#)=0THEN640
670 IFVAL(A#)<>0ANDVAL(B#)<>0THEN640
680 PRINTB#:
690 Q=VAL(A#):IFQ=0THENQ=VAL(B#)
700 Q#=A#:IFVAL(Q#)<>0THENQ#=B#
710 GOSUB860:GOTO590
730 REM PRINT CHARACTER
740 INPUT"# TIMES TO PRINT (0 TO STOP)":NT:IFNT=0THEN220
750 INPUT"# OF TIMES TO ENHANCE":ET
755 IFNT=1THENNS=0:GOTO770
760 INPUT"# SPACES BETWEEN":NS
770 OPENS,4,5:A#="":FORI=1TO6:A#=A#+CHR$(A(I)):NEXTI:PRINT#5,A#:OPEN4,4
780 IFET<>0THENFORI=1TOET:PRINT#4,CHR$(1):NEXTI
790 FORI=1TONT:PRINT#4,CHR$(254):IFNS=0THEN810
800 FORJ=1TONS:PRINT#4," ":NEXTJ
810 NEXTI:PRINT#4
820 PRINT#4:PRINT#4,"CODE IS:":FORI=1TO6:PRINT#4,A(I):NEXT
830 PRINT#4:PRINT#4,"ENHANCED"ET"TIMES"
840 PRINT#4:PRINT#4:CLOSE5:CLOSE4:GOTO740
860 REM POSITION POINT FOR PLOT
865 Z=40+Q*120:ZY=ASC(Q#)-64:Y=6+3*ZY
870 G=81:IFPEEK(32768+Z+Y)<>32THENG=32
880 POKE 32768+Z+Y,G
890 IFG=81THENA(ZY)=A(ZY)+2↑(7-Q)
900 IFG=32THENA(ZY)=A(ZY)-2↑(7-Q)
910 RETURN

```



Addressed Instructions:

BYTES:	2	3	2	1	2	2	2	2	3	3	2	STATUS		
MODE:	IMM	ABS	ZP	AC	(ZP,X)	(ZP),Y	ZP,X	ZP,Y	ABS,X	ABS,Y	REL	NV	BD	IZC
ADC	69 2	6D 4	65 3		61 6	71 5	75 4		7D 4	79 4		**	---	**
AND	29 2	2D 4	25 3		21 6	31 5	35 4		3D 4	39 4		*-	---	*-
ASL		0E 6	06 5	0A 2			16 6		1E 7			*-	---	**
BCC											90 2	--	---	---
BCS											B0 2	--	---	---
BEQ											F0 2	--	---	---
BIT		2C 4	24 3									76	---	*-
BMI											30 2	--	---	---
BNE											D0 2	--	---	---
BPL											10 2	--	---	---
BVC											50 2	--	---	---
BVS											70 2	--	---	---
CMP	C9 2	CD 4	C5 3		C1 6	D1 5	D5 2		DD 4	D9 4		*-	---	**
CPX	E0 2	EC 4	E4 3									*-	---	**
CPY	C0 2	CC 4	C4 3									*-	---	**
DEC		CE 6	C6 5				D6 6		DE 7			*-	---	*-
EOR	49 2	4D 4	45 3		41 6	51 5	55 4		5D 4	59 4		*-	---	*-
INC		EE 6	E6 5				F6 6		FE 7			*-	---	*-
JMP		4C 3										---	---	---
JSR		20 6										---	---	---
LDA	A9 2	AD 4	A5 3		A1 6	B1 5	B5 4		BD 4	B9 4		*-	---	*-
LDX	A2 2	AE 4	A6 3					B6 4		BE 4		*-	---	*-
LDY	A0 2	AC 4	A4 3				B4 4		BC 4			*-	---	*-
LSR		4E 6	46 5	4A 2			56 2		5E 7			0-	---	**
ORA	09 2	0D 4	05 3		01 6	11 5	15 4		1D 4	19 4		*-	---	*-
ROL		2E 6	26 5	2A 2			36 2		3E 7			*-	---	**
ROR		6E 6	66 5	6A 2			76 6		7E 7			*-	---	**
SBC	E9 2	ED 4	E5 3		E1 6	F1 5	F5 4		FD 4	F9 4		**	---	**
STA		8D 4	85 3		81 6	91 6	95 4		9D 5	99 5		---	---	---
STX		8E 4	86 3					96 4				---	---	---
STY		8C 4	84 3				94 4					---	---	---

Implied Instructions: (all 1 byte)

BRK	00 7	-- 1-1--	INY	C8 2	*-	---	RTI	40 6	RESTORED	TSX	BA 2	*-	---	*-
CLC	18 2	-- --0-	INY	C8 2	*-	---	RTS	60 6	--	TXA	8A 2	*-	---	*-
CLD	D8 2	-- -0---	NOP	EA 2	--	---	SEC	38 2	--	TXS	9A 2	--	---	---
CLI	58 2	-- --0--	PHA	48 3	--	---	SED	F8 2	-- -1--	TYA	98 2	*-	---	*-
CLV	B8 2	-0	PHP	08 3	--	---	SEI	78 2	-- --1--					
DEX	CA 2	*-	PLA	68 4	*-	---	TAX	AA 2	*-					
DEY	88 2	*-	PLP	08 3	RESTORED		TAY	A8 2	*-					

Indirect Instr:

JMP 6C 5 --

Op-code table

	0	1	2	4	5	6	8	9	A	C	D	E	
\$0	BRK	ORAIX			ORAZP	ASLZP	PHP	ORAIM	ASLAC		ORA AB	ASLAB	\$0
\$1	BPL	ORAIY			ORAZX	ASLZX	CLC	ORAA Y			ORA AX	ASLAX	\$1
\$2	JSR	ANDIX		BITZP	ANDZP	RQLZP	PLP	ANDIM	ROLAC	BITAB	ANDAB	ROLAB	\$2
\$3	BMI	ANDIY			ANDZX	RQLZX	SEC	ANDAY			ANDAX	ROLAX	\$3
\$4	RTI	EORIX			EORZP	LSRZP	PHA	EORIM	LSRAC	JMPAB	EORAB	LSTAB	\$4
\$5	BVC	EOR IY			EORZX	LSRZX	CLI	EORAY			EORAX	LSRAX	\$5
\$6	RTS	ADCIX			ADCZP		PLA	ADCIM		JMPIN	ADCAB		\$6
\$7	BVS	ADCIY			ADCZX		SEI	ADCAY			ADCAX		\$7
\$8		STAIX		STYZP	STAZP	STXZP	DEY		TXA	STYAB	STAAB	STXAB	\$8
\$9	BCC	STAIY		STYZX	STAZX	STXZY	TYA	STAA Y	TXS		STAAX		\$9
\$A	LDYIM	LDAIX	LDXIM	LDYZP	LDAZP	LDXZP	TAY	LDAIM	TAX	LDYAB	LDA AB	LDXAB	\$A
\$B	BCS	LDAIY		LDYZX	LDAZX	LDXZY	CLV	LDAAY	TSX	LDYAX	LDA AX	LDXAY	\$B
\$C	CPYIM	CMPIX		CPYZP	CMPZP	DECZP	INY	CMPIM	DEX	CPYAB	CMPAB	DECAB	\$C
\$D	BNE	CMPIY			CMPZX	DECZX	CLD	CMPAY			CMPAX	DECAX	\$D
\$E	CPXIM	SBCIX		CPXZP	SBCZP	INCZP	INX	SBCIM	NOP	CPXAB	SBCAB	INCAB	\$E
\$F	BEQ	SBCIY			SBCZX	INCZX	SED	SBCAY			SBCAX	INCAX	\$F



The PAPER Reader Profile

Will you subscribe to Volume 4? (Y/N) \_\_\_\_\_  
 How many friends can you get to subscribe? \_\_\_\_\_  
 Are you Male or Female? (M/F) \_\_\_\_\_

Your age	Your education	Your income	Your occupation
_____ <18	_____ some HS	_____ < \$10,000	_____ student
_____ 18-21	_____ HS grad	_____ \$10-20,000	_____ teacher
_____ 22-30	_____ some college	_____ \$20-30,000	_____ engineer
_____ 31-40	_____ 2yr/tech deg	_____ \$30-40,000	_____ programmer
_____ 41-50	_____ 4yr college deg	_____ \$40-50,000	_____ computer dealer
_____ 51-60	_____ master's deg	_____ \$50-100,000	_____ other blue col.
_____ 61+	_____ PHd	_____ \$100,000+	_____ other white col

How much money did you spend on computers and related materials last year? \_\_\_\_\_  
 What computer and/or peripherals do you own or use? \_\_\_\_\_

	at home	at work/school		at home	at work/school
\$ <100	_____	_____	old PET	_____	_____
\$100-250	_____	_____	new PET ( K)	_____	_____
\$250-500	_____	_____	PET printer	_____	_____
\$500-1000	_____	_____	Other printer	_____	_____
\$1000-2500	_____	_____	PET disk	_____	_____
\$2500-5000	_____	_____	other disk	_____	_____
\$5000-10,000	_____	_____	Apple	_____	_____
\$10-25,000	_____	_____	Atari	_____	_____
\$25,000+	_____	_____	TRS-80	_____	_____
other micros	_____	_____			

Which of the following magazines do you enjoy reading (R) or subscribe to (S)?  
 COMPUTE! \_\_\_\_\_ Microcomputing \_\_\_\_\_ BYTE \_\_\_\_\_ Micro \_\_\_\_\_ On Computing  
 Creative Computing \_\_\_\_\_ Personal Computing \_\_\_\_\_ Recreational Computing \_\_\_\_\_

Do you belong to a computer users group? (Y/N) \_\_\_\_\_  
 Is this group PET oriented? (Y/N) \_\_\_\_\_  
 If it's an active PET user's group please give the name & address of a contact.

My primary use of the PET is business \_\_\_\_\_ pleasure \_\_\_\_\_ scientific \_\_\_\_\_  
 education \_\_\_\_\_  
 I think the PET is the best \_\_\_\_\_ one of the best \_\_\_\_\_ a good \_\_\_\_\_ micro.  
 About \_\_\_\_\_ others read my copy of The PAPER.  
 I have \_\_\_\_\_ have not \_\_\_\_\_ encouraged others to subscribe.

We would be happy to here any suggestions you have. Are you happy with the article quality and level? Would you like to see more/less of certain things? What about a new name starting with Volume 4? Please feel free to write your comments below. If you like include further comments on an additional sheet of paper.

1974

Faint, illegible text, possibly a header or introductory paragraph.

**THE PAPER**

**Box 524**

**East Setauket, NY 11733**

Main body of faint, illegible text, likely the address label or a list of items.

## Fun with PETs and POKEs

by Andrew Stezsyn

Location 101 (decimal) on the old ROM PETs is supposed to point to the "variable psuedostack". On the new ROM PETs it is not clear if there is a location that does the same thing. When you poke certain numbers to 101 some interesting things happen. Here are the values and their results. In most cases you should do the POKE first and then LIST to see the result.

POKE 101,X : LIST where X =

104	normal mode
226	program is shifted 5 spaces to the right when listed
240	extra space between lines with less than 40 characters
225	program not displayed, only blank space
183	line #'s are in decimal form
208	need colon after command for it to be executed
150	list program line #'s only
114	no line numbers and no linefeed after each line

In many cases a program will be destroyed after one of these POKEs. It will not run properly even if the "normal" value of 104 is POKed back.

We found that location 19 (decimal) in the new ROMs corresponds to location 101 in the old ROMs. We tried POKing from 1 to 255 in this location and found some strange results; some have to be seen to be believed. As before some bomb the PET while some allow you to recover control by hitting certain keys (especially STOP and HOME). The program usually will RUN but this often clears the effect of the POKE. - ed & pub

5,6,30	removes line numbers and adds some garbage
7	try this yourself!
11,12	drops some RETURNS
18,254,255	LIST program behind bars
21,38-40,93	scrambles program
97,107,108	
22	normal mode
32,96,98,109	LIST program without line #'s
68,69	LIST line numbers w/o program
95	LISTs program with no #'s, behind bars
99	LISTs with decimal line #'s
110,111,119	PET ignores all commands
120-122	
157,159	LIST in reverse field
173	clears screen permanently
174,175	LOAD ? (try STOP)
194	LISTs in strange places (?)
195,197	disables LIST
198	indents LISTing
211-213	space between short lines
214,215	PET waits for STOP or HOME
240-242	
224-227	switch program lines around
237-239	
244	drops first line

by Roy Busdiecker  
PO Box 1442  
Woodbridge, VA 22193

My toes are frozen solid! Winter in Woodbridge is bitter cold this year. This afternoon, with wind chill, the effective temperature was -10 to -20 degrees Fahrenheit. It's January 10 and The PAPER just arrived (at least THAT gave me a little warm feeling, and I appreciate all the warmth I can get!) One consolation is that by the time another issue arrives we should be past the worst of the cold weather.

It may not make sense to everyone, but I'm going to write the Observations generally in the order in which articles appear in The PAPER. This will make it easier to compare the remarks to the articles about which they're written.

The first favorable impression this issue came with the table of contents. There was a good mix of both subjects and authors. A growing interest in assembly/machine language programming was balanced by articles on BASIC and "real-world" applications. It was good to see some familiar names (MacArthur, Wachtel, Swan, Turco) among many which appear to be part of an energetic and enthusiastic LIPS group (we're going to have a lot of fun with that acronym!).

Publication of circulation figures (p2) is commendable. I hope that all readers will accept the challenge to make those figures grow. The experience of the past few years has convinced me that "word of mouth" is a very effective mode of advertising, and there are many PET/CBM owners who would enjoy "The PAPER" if they only knew about it.

Thanks to the publisher for my title of Associate Editor. It's an honor and privilege that I really appreciate (I would also appreciate your correcting of my last name in the masthead). It will be a pleasure to add the title to my resume.

With this second double issue, the obvious question is "are they all going to be double?". While it would be nice to get The PAPER twice as often, it's also easy to understand the extra effort and expense involved in putting out a publication more frequently. As long as the quality and "folksy flavor" of The PAPER keep up, I'm willing to wait a little longer to get it.

Ralph and Doug's thoughts on software protection were well-presented. The moral issue of illicit copying, versus legitimate right of the consumer to have a backup copy of the program he paid for, is a tough problem that we're going to have to face and resolve, sooner or later.

The review of DASM/TASM by Ralph Bressler is a good example for aspiring review-writers. It's obvious that Ralph knows something about the product area he's reviewing, and he's presented a reasonably detailed description of the package. In addition, he's identified both strong and weak points. It was a good review, and a welcome contrast to the views of another publisher some time back, who responded to my suggestions by saying that his reviews were valuable just because they proved that the product existed.

Reader I/O is great, the kind of direct user-to-user exchange that can be so valuable. May I offer a few suggestions? Why not quote parts of the reader letters (if they're too long to publish in their entirety), rather than summarizing them (remember the "Dear Terry" letters in the early issues of The PAPER?). It would also be nice to publish the address of each contributor, to allow direct responses.

In response to Don Condon's suggestion, let me resurrect one of my own: call this publication "The PET PAPER", as it was originally titled. COMPUTE continues to run a section called "The PET Gazette", and the old "PET Users Notes" didn't seem to have any trouble with the name.

The Osbourne book, "PET/CBM Personal Computer Guide", as mentioned by Ralph, is a winner. Even those who've followed all the newsletters and magazines may

find some new lessons in it. I understand that a new Second Edition should be out momentarily. (It is out. See the review in this issue.-Ed.)

George Spilich's article, Microcomputers in Psychology, provided just enough information to arouse my curiosity (which seems to be what he intended). Tell us more, George! Many of us would be interested in the particulars of your lab use of PETs in cognitive processes. Are there simple examples that we can use to learn more about the field?

In my article on page 12, either a word got dropped or I got carried away with nostalgia. The "old ways" of PET lore were late '77 and early '78, not early '77-78 (the first PETs started appearing in Oct-Nov 77).

Congratulations to Pub (p15), for passing Gerry Eisner's "final exam". As a reward, we'll let you publish another issue!

It was good to see Jim MacArthur and SEARCH again. Those who don't have the PAICS Toolkit will look forward to seeing the version of Jim's article that will work on new (BASIC 2.0) ROMS. (Jim, if you still need to experiment with a new ROM machine come to Virginia Micro Systems. Call us at (703) 491-6502, and we'll give you directions.)

J. R. Bromley did a nice job with the "Screen Copy" article. Providing both old and new ROM locations, as J. R. did, should be a requirement for all programs of this sort. We should start seeing more on 8000 and 4000 series computers, too.

Reading Sherwood's article on tape care reminded me of Roy Buchanan's suggestions (published some time ago) to rewind cassettes just before use, rather than just after use. This results in less tape tension, which reduces the likelihood of print-through and loss of signal.

Jim Fowler's review of ASSM/TED should have been published next to Ralph's review of Baker's product (p5). The products represent two worthwhile ends of the spectrum ... the professional, full feature system, priced fairly, and the inexpensive starter system for the beginner whose needs are limited. I've used the EHS products (both tape and disk), and found very little more that I could ask for. EHS also has a new disk package for the 8032 (more features have been added, too).

All in all, issue 6/7 continued the good impression created by the first one out of the LIPS. The articles were good and interesting. The only sour note was the continued presence of too many typographical errors. Could LIPS find some volunteer proofreaders to help out?

That's all for now. Enjoy your PETs, experiment, and (most important) WRITE ... share your discoveries and questions.

#### Feedback on Volume III, Issue 6/7

by Fran Turco  
c/o Vega Precision Labs  
800 Follin Ave.  
Vienna, VA 22180

Page 5, Review of DASM/TASM

According to the write up, DASM/TASM has the following advantages and disadvantages:

#### Advantages

Low cost

#### Disadvantages

Slow speed

Non-standard mnemonics

No macros

Does not store object code directly

The summary states "...you won't find a faster, workable assembler anywhere."

Does that mean "for \$10.00 or under", "written in BASIC", or unconditionally? I have not had experience with this particular assembler, but Dan Fylstra's "6502 Assembler in BASIC" is painfully slow. (This assembler is fast for one written in BASIC. It is the best assembler I have seen FOR THE PRICE. It does, however, require two tape decks which may be a problem for some.-Pub.)

Carl Moser's ASSM/TED and MAE have none of the disadvantages listed above. They are written entirely in machine language and assemble code very rapidly. In addition, they offer conditional assembly and a fine editor with such niceties as SEARCH & REPLACE. Cost is \$50.00 for cassette based ASSM/TED or \$170 for disk-based MAE with a 35 page manual.

For beginning assembly language programmers on a tight budget DASM/TASM seems to be the answer. For more advanced programmers, ASSM/TED seems superior.

Page 12 & 13, Observations, Volume 3, Issue 1, 2/3

I stand corrected. Roy is right. If you purchase a program it should work. PIMS was written for the TRS-80. Mr. Gupta lists some corrections which he says will allow the program to work on the PET. It does not.

If anyone is interested, I will be glad to send a copy of PIMS as I have corrected it for an old ROM PET. It should work on any PET since it is in BASIC. Of course, proof of ownership of PIMS is necessary. The easiest approach is to send me your cassette copy of PIMS. I will record on the back of it and return it to you.

While it is true that the keywords in lines 63988-63995 are stored in ROM, I chose not to use them. The program becomes non-transportable if you rely on ROM locations for your keywords. Written entirely in BASIC it can be used on any PET and probably on other microcomputers using Microsoft BASIC, such as the APPLE.

The tables in the March/April issue were written specifically at the suggestion of Roy Busdiecker in an earlier issue. There are two different tables. The first, printed on pages 4-9, is in order of mnemonic. The second on pages 10-14, is in order of code.

Page 23, Auto Repeat Keys, Version Two

Under the narrative, item 2, the numbers 845 and 034D should be 827 and 033B. Also, isn't it possible to relocate this routine so that it does not overlap both cassette buffers. The original version was entirely in the second cassette buffer and the owner could still load and save using the first tape drive.

Page 24, Program Exchange

How can a person get a copy of one or two of the programs if he does not have a program to exchange for it? (This is indeed a problem. These programs were contributed with the understanding that they would be exchanged only. For those that want a program and have none to give, I suggest writing an article on some aspect of the PET or its use.-pub)

Pages 32, Observations Resumed

It is my opinion that the use of the @ to set off comments falls in the same category as NEXT without specifying the variable, failing to close a quotation with the second set of quote marks, etc. There are things you can "get away with" on the PET because Microsoft BASIC is forgiving. However, they are poor programming practice and generally will not work on other versions of BASIC.

## Printout

Somewhere between The PAPER and COMPUTE

by Ralph Bressler

Printout is an all-PET magazine published in England. It is slick but also folksy and runs about 40 pages every month. To understand some of the reasons why Printout is so good you have to understand the English and Continental PET scene. In England Commodore actually speaks, and understandably, to us end users. Much of the interesting software originates there also. Commodore probably has over 50% of the micro market. Also, there are many happy and knowledgeable PET users in England that support their user's groups and publications.

Printout has just completed Volume 1 and are rolling on the second volume. Back issues are available for \$3.50 each. Volume 2 is \$36.00 surface or \$45.00 airmail in US funds. The address is Printout, PO Box 48, Newbury RG16 OBD, Berkshire, England. I highly recommend airmail since surface is so slow.

The following bits of information are only a brief summary of what may be found in the last volume of Printout. They reflect my biases but I thought you might find them informative.

### Strange Invaders

I have tried to disassemble Invaders to see how it works and to control certain features. It is incredibly complex and hand assembled and impossible to really understand. Paul Higginbottom of Commodore England has been able to find out something about the program and suggests the following changes.

Do NOT let the game run for long in the "display" mode. For some reason this causes the program to abort in mid game when someone begins playing. Simply hit a key so that the screen says 'PUSH ANY KEY TO START' and this solves the problem. You may also want to change certain memory locations which control the speeds of various game elements. For example changing location 0E01 from 2 to 1 doubles the speed of your base across the bottom of the screen. Changing it to 4 halves the speed. Location 0E09 controls the speed at which the invaders drop bombs. Changing this from 4 to 2 increases this speed. 0E0E is the location for the speed of the mystery craft. Decreasing the value of this location from 6 to 2 increases the mystery craft speed. Finally, location 0623 is the character used to erase your missile as it is fired. It contains 20 hex or 32 decimal which is a space. Changing this to A0 hex will leave behind a wall which you can fire through, but the Invaders cannot.

All these changes can be POKEd using the decimal values or entered using the TIM monitor.

### Program Recovery

Gavin Sanders, Printout Contributing Editor, suggests the following fix for all of us who might just have typed NEW without SAVING our latest creation. This fix only works if you have a Programmer's Toolkit (for \$35 who doesn't?). To recover your program type: POKe 1026,4: SYS50242. After hitting RETURN type: FIND IT. Your program will have returned from the 'land of the lost'.

### Cursor Positioning

This one comes from Tommy Turnbull the Technical Editor for Printout. I feel it's easier to position the cursor other ways but you might like it.

```

10 X%=226:Y%=245:Z=58843:IF PEEK(50003) THEN X%=198:Y%=216:Z=57949
20 READ X,Y: IF X=-1 THEN 60
30 POKE X%,X: POKE Y%,Y: SYS(Z): PRINT "##": GOTO 20
40 DATA 1,1,2,2,3,3,4,4,5,5,6,6,7,7,
50 DATA 16,6,15,5,16,6,15,5,14,4,13,3,12,2,-1,-1
60 GOTO 60

```

X% is the horizontal position of the cursor while Y% is the vertical. Line 10 automatically adjusts these values for old or new ROMs but not for BASIC 4.

### Reverse List

Gavin Sanders offers the following bit of PET info. This is one of those that is interesting but has little practical use in itself. Type in the following short program and then give the POKES that follow in direct mode. After this try RUN and LIST.

```

1 PRINT "IT'S RALPH AGAIN (2up)"
2 G=G-1
3 FOR I=1 TO 50: NEXT
4 IF G=1 THEN PRINT "(rev)";
5 GOTO 5

```

```
POKE 1027,5: POKE 1052,4: POKE1076,2: POKE 1090,1
```

This should work on any PET but it must be typed in exactly as you see it!

### Title Retrieval

Gavin Sanders also gives us this tip for finding out the exact title of a program after a cassette LOAD or SAVE.

```
POKE 158,32 - reveals a short title
POKE 158,204 - gives a the complete title even if more
                than 16 characters
```

### Fast Graphics

Lindsay Doyle has taken the trouble to modify Warren Swan's excellent Fast Graphics program to work on the new ROMs. Only a few data items need to be changed and its really fine utility. A revised copy will be available on our program exchange.

### PET Screen Clock

Mark Valentine presents a little machine language routine that places a 12 hour clock in the upper right hand corner of the screen. The clock stays there whether you clear the screen or not. It can be set using a few POKES to change the time.

In summary, Printout contains an interesting but odd mix of articles dealing with information I have never seen and things I knew months ago. The ads for chips, hardware, and software are also intriguing but the prices are outrageously high. Printout also gives an interesting insight into what is happening with Commodore in other countries and what may be coming our way. At times, though the inside commentary about figures in the British PET world is bothersome.



by Ralph Bressler

The PET is a fine microcomputer which can be used in many different settings. However, the PET like many other micros cannot produce hardcopy without the addition of a printer. Printers come in many sizes, shapes, prices, and print qualities. Price is, to a great extent, determined by the versatility and print quality of the printer. Print quality runs from standard dot matrix to "correspondence quality" dot matrix to "letter quality" printing indistinguishable from the best IBM typewriter. The following list is meant to be a brief summary of printers available for the PET with a brief description and approximate price. The prices are only approximate since they depend greatly on who you buy them from and the options you choose.

There are basically three categories of printers available. Some printers are either plug compatible with the PET or come with an interface installed. Other printers require that you buy an interface from another source. The last type claim that they could be interfaced to a PET. Most of these in the last category can, in fact, be connected with no problem. However, when you do buy a printer be sure you are buying a printer that plugs directly into the PET or that an interface is included which accomplishes this since this is the surest way to be up and running immediately. Some of these printers will output the entire PET graphics set. In most cases a graphics option means that the printer either has special graphics symbols or a high resolution mode. High resolution options mean that you are able to control the wires of the dot matrix printhead individually. It may also mean you can control the spacing between lines using very small increments.

I have not seen all these printers work but have talked to people who have. In many cases I have received letters printed on these machines or have gotten notes on their use with the PET. In any case, this is NOT A REVIEW of any of these printers. This is only a summary gleaned from brief comments of a few users, magazine advertisements and catalogs. I suggest that you only buy equipment after seeing it work and, preferably, after using it yourself. I hope this article may stimulate some of you to write about your favorite PET compatible printer.

1.) Axiom Ex 801 (\$299) Plugs directly into PET. Dot matrix print is hard to read but makes excellent photocopies. Prints electrostatically on special aluminized paper which can be expensive. It uses the full PET graphics set and is, therefore, great for program listings. Prints 40 characters or 80 characters across without sacrificing readability.

2.) Trendcom 100 or 200 (\$450, \$675) Plugs directly into PET's IEEE port. Readable 5x7 dot matrix print. Prints on white thermal paper and is very quiet. Features bidirectional printing and a 96 character set including upper and lower case and graphics characters. Speed is 40 cps at 10 characters per inch. Also marketed as PAL 40 and 80.

3.) Seikosha GP 80 (\$450) PET interface available for \$80. Readable 5x7 dot matrix print. Character set includes 128 characters with graphics. Prints at 80 cps. Sold by Systems Formulate which has brought us other interesting products.

4.) Epson MX-70 (\$450) Plugs directly into PET's IEEE port. Readable 5x7 dot matrix print. This is a "stripped" version of the MX-80 below which retains most of the necessary features of that model. It has tractor feed, a disposable printhead, unidirectional printing at 80 cps, and programmable form handling. The Graftrax II option allows 480 dots across or 60 dots per inch resolution.

- 5.) Okidata Microline 80 (\$550) Can be interfaced to PET. Good quality 9x7 dot matrix. Features include tractor feed, 64 different characters, double width characters, 6 or 8 lines per inch and unidirectional printing. Speed is 80 cps.
- 6.) Comprint 912 (\$560) Plugs into PET with available interface. Correspondence quality print. Features true lower case descenders and compatibility with most word processors.
- 7.) Epson MX-80 and TX-80 (\$595 and \$730) Plugs directly into PET's IEEE port. Good quality 9x9 dot matrix print. This company makes the PET printer mechanism. The printhead on all models is disposable and therefore easily replaced by anyone. Features include 64 special (not PET) graphics characters, adjustable width tractor feed, multi pass printing, true descenders on lower case, and bidirectional, logic seeking printing. Print speed is 80 cps. The TX has a graphics option with variable line spacing.
- 8.) Base 2 (\$649) Plugs into PET with available interface. Good quality dot matrix print. Features an excellent graphics option, tractor or friction feed, programmable and expanded characters, programmable horizontal and vertical tabs and line spacing and bidirectional, logic seeking printing. Speed is 100 cps. Graphics option is \$100 extra.
- 9.) MPI 88G (\$749) Can be interfaced to the PET. Correspondence quality 11x7 dot matrix or 7x7 good quality dot matrix. Features tractor or friction feed, unidirectional or bidirectional printing, double width printing, upper and lower case and a 1K receive/output buffer. Graphics option is \$50.
- 10.) IDS Paper Tiger 445 (\$795) Can be interfaced to the PET. Good quality dot matrix print. Features variable character sizes, 80 to 132 columns, DotPlot graphics option and unidirectional printing. Speed is 190 cps. Company has reputation for printers that never die. More features available on other models.
- 11.) PET 2022 (\$795 + \$40 cable) Plugs directly into PET's IEEE port. Good quality 7x6 dot matrix print. Features full set of PET characters including upper and lower case and all graphics. Lower case has no descenders. Also has tractor feed and a full range of formatting commands. Ability to create programmable characters. This is the best choice for producing PET program listings but is not the best choice for many other applications.
- 12.) Centronics 737 (\$995 + \$125 for interface) Plugs into PET with available interface. Correspondence quality 18x9 dot matrix with 3 different type styles which can be chosen under program control. Features include proportional spacing, underlining, justification, adjustable tractor or friction feed and true lower case descenders. Prints from 80 to 132 columns at speeds of 50 to 80 cps. No graphics option.
- 13.) IDS Paper Tiger 460 (\$1295) Can be interfaced to the PET. Correspondence quality 9 wire overlap dot matrix. Features include 3 typefaces, proportional spacing, justification, programmable horizontal and vertical tabs, forms control, graphics, bold printing, upper and lower case with true descenders and bidirectional printing.
- 14.) C. Itoh Starwriter (\$1550 w/ parallel interface) Plugs into Pet with available interface. Letter quality printing produced with a daisy wheel. Features include proportional spacing and bidirectional printing with tractor feed optional. Prints at 25 cps. Marketed as Vista or TEC V300.
- 15.) IDS Paper Tiger 560 (\$1695) Can be interfaced to the PET. Correspondence

quality 9 wire overlap dot matrix. Features tractor feed, justification, fixed or proportional spacing, programmable tabbing, DotPlot high resolution graphics and bidirectional printing in up to 132 columns. Speed is 150 cps.

16.) IBM 550 (\$2000) Plugs into PET with available interface. Letter quality type with standard Selectric type ball. This modified Selectric Electronic Typewriter has the features of this machine including a replaceable type ball. It can also act as a stand alone typewriter. Print speed is 10 cps (slow) and does not include any graphics.

17.) PET-Tally (\$2100) Plugs directly into the PET's IEEE port. Good quality dot matrix print. Features a pedestal mounting, bidirectional printing, two print sizes and true descenders on lower case. Prints up to 200 lines per minute. Very heavy duty model.

18.) LA-120 (\$2450 + \$100 interface) Plugs into PET with available interface. An extremely heavy duty pedestal mounted workhorse. Prints at 1200 baud in 5 different type styles. Will not support PET graphics.

19.) Typrinter 221 (\$2850) Can be interfaced to the PET. Letter quality print using a daisywheel. Features plain English commands embedded in the text, centering, columned printing, reverse and boldface printing, justification, proportional spacing and three separate pitches. Also acts as a stand alone typewriter.

20.) NEC Spinwriter (\$3100) Plugs into PET with available interface. Letter quality printing. Features an interchangeable print thimble and a full range of formatting features. Heavy duty and reliable. Serviced by several US firms. Recommended by Commodore and works beautifully with any WordPro.

21.) Diablo 630 (\$3200 w/ interface) Plugs into PET with available interface. Letter quality printer uses daisywheel for the best quality print. Features full formatting commands, single pass or reversible ribbons. Heavy duty, made and serviced by Xerox. Makes great printer for business form letters or advertisement.

Many different interfaces can be purchased to use with the printers that need them. The following are a few that seem to get consistently good reviews. As with printers you should not buy one unless you have strong evidence that it will successfully connect the printer of your choice to the PET. Strong evidence would be seeing it work or talking to someone who has one up and running.

1.) PIE-C (\$120) Can be used for Centronics, Spinwriter, "Paper Tigers", Anadex and others. Not very versatile but works well. Connects PET IEEE to a parallel printer.

2.) Complications C-101 (\$250) IEEE to parallel interface with stackable IEEE connectors. Switch selects for upper case only or upper and lower case.

3.) CmC SADI (\$295) Two way serial and parallel interface for IEEE or RS-232. Will connect almost any printer to the PET's IEEE or user port.

4.) CmC ADA 1450 (\$139) Serial adaptor to connect serial printers to the IEEE port.

5.) CmC ADA 1600 (\$129) Connects standard parallel printers like the Spinwriter or Centronics to the PET IEEE.

6.) CmC ADA 730 (\$129) Parallel to IEEE adapter to connect Centronics 730 or 737.

All the Connecticut microComputer interfaces have stackable connectors and do not tie up the PET busses. Contact the company for the appropriate interface.

Finally, there is a way to connect any typewriter to your PET. This means that printing is slow but you do get letter quality type. The device below sits on top of the typewriter keyboard and has plungers which hit each key. This is primitive but effective.

1.) I/O Pack (\$585) Fits over a typewriter keyboard and connects directly to the PET. Can run at 50 cps but is usually set at 12 cps.

Disk Master  
A Product Review

Type: Software

by Ralph Bressler

Model PET: BASIC 2.0 and 2040 disk

Source: Baker Enterprises

15 Windsor Dr.

Atco, NJ 08004

Price: \$10

Bob Baker has done it again. He has produced a useful program at a very low cost. Disk Master can be used to catalog all of the disks in your growing library. All you need to do is format a new disk and record disk master as the first program on it. When you want to add a disk to the catalog you simply place this Master disk in drive 0 and run the program. The disk to be cataloged is placed in drive 1 and then Disk Master takes over.

Five major functions are available from the main menu. The UPDATE option adds a disk to the catalog after asking you for verification. Programs are read from the directory and alphabetized. A warning will occur if the disk ID has already been entered in the master catalog. If everything is OK the program will wait for another disk to be placed in drive 1 or you can return to the main menu. DELETE allows you to remove all records of a disk from the catalog. Disks may be specified by name or ID.

The DISPLAY option will print the directory for a selected disk identified by name or ID. All entries are alphabetized. FIND will let you search the entire catalog of disk for a specific program or data file. Character matching can be used by typing a '\*' after a set of characters. For example, DOS\* would find all files beginning with DOS. This function is VERY useful but it is also quite slow. This is partially due to the fact that it is all done in BASIC and that character matching is used.

Finally, you may use the LIST option to output the names and/or ID's of disks that have been cataloged. There are three choices within this option. The simplest option prints just the IDs in use in alphabetical order. A second option prints the IDs along with the names of the disks. Again this list is alphabetized. The last option requires a printer. It prints a chart which shows all the IDs in use consisting of the digits 0 to 9 and the letters A to Z. This can then be used as a check sheet when assigning new disk IDs.

The single reservation I have about this program is that it is slow when searching for a program in the catalog. It is important to use an ID only once since repeated IDs may cause the destruction of a valuable disk. This program makes disk cataloging easy with adequate speed and reasonable price.

## PET's Real Cursor

by Doug Haluza

Many programs lend themselves well to the use of the GET statement. The problem with GET is that it "turns off" the PET's cursor. There has been much talk about "counterfeit cursors" to use with a prompt with GET. The PET's real cursor can do the job much better with far less effort.

The hardware interrupt (IRQ) routine will flash the cursor into the opposite field once every 20 interrupts (1/3 sec.) as long as location 548 (old) or 167 (new) contains 0. To get the cursor flashing during the GET just POKE 0 into this location. There are some problems with this, however. Try the routine below and you'll see:

```
10 CO=548:IF PEEK(50003) THEN CO=167
20 POKE CO,0
30 GETC$: PRINT C$: GOTO30
```

You may have noticed that the cursor gets "left behind", especially if you used the cursor controls. DELEte and INSErT also do some funny things. These problems are not solved by many of the "phoney cursor" routines and will mess them up if not properly checked. The following routine does that checking and corrects any problems by ignoring any control characters except DELEte and RETURN:

```
0 CO=548: BF=525: QU=234: IF PEEK(50003) THEN CO=167: BF=158: QU=205
1 POKE CO,0: A$="": PRINT"INPUT TEXT:";
2 WAIT BF,7: GETC$: A=ASC(C$): IF A=13 THEN 6
3 IF A=20 AND LEN(A$)>0 THEN A$=MID$(A$,1,LEN(A$)-1):
  POKE QU,0: PRINT"(sp)(2left)(sp)(left)";: GOTO2
4 IF (A AND 127) < 32 THEN 2
5 A$=A$+C$: PRINT C$;: GOTO2
6 POKE CO,1: PRINT"(sp)": RETURN
```

Some explanation is in order:

POKE CO,0 turns on the cursor by setting a flag checked by the IRQ routine.

WAIT BF,7 waits until location BF (keyboard buffer pointer) logically ANDed with 7 does not equal 0. (e.g. it'll wait until a character is typed) WAIT BF,1 could be used, but it waits for an odd number of characters to be typed.

IF A=13 then a carriage return was typed, so we're finished (almost).

IF A=20 AND LEN(A\$)>0 then delete was hit, AND some characters still remain, so it can be safely handled.

A\$=MID\$(A\$,1,LEN(A\$)-1) will take the 1st to the next to the last characters of A\$ which is where the string being collected is kept (RIGHT\$(A\$,LEN(A\$)-1) won't always work here)

POKE QU,0: PRINT"(sp)(2left)(sp)(left)" turns off the quote mode flag (just in case it was set by typing a ") and erases BOTH the last character AND the cursor 'blob'.

IF (A AND 127)<32 then the character is some other control character and we want to ignore it. A AND 127 drops bit 8 which would be set if it was a shifted machine command like (up) or (off).

A\$=A\$+C\$: PRINT C\$; adds the characters to the running tally and prints it on the screen.

POKE CO,1: PRINT"(sp)" turns off the cursor and erases the 'blob' if it's there.

Here's a special application of the GET input routine that takes a string and prints it out backwards...Note how the string is assembled in line 50 and how (del) is handled in line 30.

```
5 CO=548: BF=525: QU=234: IF PEEK(50003) THEN CO=167: BF=158: QU=205
10 POKE CO,0: PRINT"(clr)(dn)YOUR NAME IS:";
20 WAIT BF,7: GETC$: A=ASC(C$): IF A=13 THEN 60
30 IF A=20 AND LEN(N$)>0 THEN N$=MID$(N$,2): POKE QU,0:
   PRINT"(sp)2left)(sp)(left)";: GOTO 20
40 IF (A AND 127)<32 THEN 20
50 N$=C$+N$: PRINTC$;: GOTO20
60 PRINT"(sp)":IF N$="" THEN PRINT "YOU MUST HAVE A NAME:";: GOTO 20
70 PRINT"YOUR NAME SPELLED BACKWARDS IS:"; PRINT N$
```

And lastly let me leave you with yet another special application of the GET input routine. This one takes in a number in any base set by the variable B (e.g. for HEX input B=16, binary B=2 or for decimal B=10). The string is not really a string, but rather each character's ASCII value is the value of each of the digits typed in. The values are corrected if the digits are letters (as in HEX numbers where A=10, B=11,...F=15) and checked to see if they're legal in line 4. Note the formula used to do the corrections. Seven must be subtracted from the ASCII value of the character if it's greater than 57 (e.g. if its a letter). The formula '(A>57)\*7' is either -7 or 0. By adding it to the rest of the formula, seven will be subtracted if A is greater than 57.

After the string is collected the number is obtained by repeated multiplication. The process goes like this, take the number you have and multiply it by the base, than add the next digit, and repeat the process until you have all the digits. It's really much simpler than it sounds, say, for example you typed in '1234' and the base was set to 10. The process goes something like this:

Number	*10	+next digit
0	0	1
1	10	12
12	120	123
123	1230	1234
1234	no more digits	

And there you have it, so here's the listing:

```
0 CO=548: BF=525: QU=234: IF PEEK(50003) THEN CO=167: BF=158: QU=205
1 B=10: POKE CO,0: A$="": PRINT"INPUT A NUMBER IN BASE"B":";
2 WAIT BF,7: GETC$: A=ASC(C$): IF A=13 AND A$<>"" THEN 6
3 IF A=20 AND LEN(A$)>0 THEN A$=MID$(A$,1,LEN(A$)-1):
   POKE QU,0: PRINT"(sp)(2left)(sp)(left)";:GOTO2
4 A=A-48+(A>57)*7: IF A<0 OR A>B THEN 2
5 A$=A$+CHR$(A): PRINT C$;: GOTO 2

6 POKE CO,1: PRINT"(sp)":A=0: FOR B=1 TO LEN(A$):
   A=A*B+ASC(MID$(A$,B)): NEXT
```

```

10 REM * PET'S CURSOR *
20 REM BY D. HALUZA
30 REM FROM THE NEW PAPER PROGRAM EXCHANGE
40 REM :
50 REM MODIFIED FOR NEW OR OLD ROMS
60 REM AND DOLLED UP BY EMIL VOLCHECK
70 REM 1981 JANUARY 26
80 POKE59468,12:CF=548:KB=525:DC=234:CL=245:PT=1
90 IFPEEK(50003)THENCF=167:KB=158:DC=205:CL=216
100 PRINT"EXAMPLES OF USING PET'S CURSOR WITH GET"
110 PRINT"-----"
120 PRINT"EXAMPLES :"
130 PRINT"  1) BASE CONVERSION
140 PRINT"  2) NAME BACKWARDS
150 PRINT"  3) TEXT TYPING
160 PRINT"  4) QUIT
170 PRINT"CHOICE # ? ";
180 POKECF,0:GETCH$:IFCH$=""THEN180
190 POKECF,1:CH=VAL(CH$):IFCH<1ORCH>4THENRUN
200 ONCHGOTO220,330,410,999
220 INPUT"BASE TO CONVERT TO DECIMAL ";BA:IFBA>36ORBA<2THEN220
230 POKECF,0:A$="":PRINT"INPUT NUMBER BASE"BA:"";
240 WAITKB,7:GETC$:A=ASC(C$):IFA=13AND A$<>""THEN300
250 IFA=20ANDLEN(A$)>0THENA$=MID$(A$,1,LEN(A$)-1):GOTO270
260 GOTO280
270 POKEDC,0:PRINT"  ||| ";:GOTO240
280 A=A-48+(A>57)*7:IFA<00RA>=BATHEN240
290 A$=A$+CHR$(A):PRINTC$:GOTO240
300 POKECF,1:PRINT"  ":A=0:FORB=1TOLEN(A$):A=A*BA+ASC(MID$(A$,B)):NEXTB
310 PRINT"IN DECIMAL YOUR NUMBER IS"A
320 GOSUB520:GOTO220
330 N$="":POKECF,0:PRINT"YOUR NAME IS:";
340 WAITKB,7:GETC$:A=ASC(C$):IFA=13THEN380
350 IFA=20ANDLEN(N$)>0THENN$=MID$(N$,2):POKEDC,0:PRINT"  ||| ";:GOTO340
360 IF(AAND127)<32THEN340
370 N$=C$+N$:PRINTC$:GOTO340
380 PRINT"  ":IFN$=""THENPRINT"YOU MUST HAVE A NAME:";:GOTO340
390 PRINT"YOUR NAME BACKWARDS IS:":PRINT""TAB(15-LEN(N$)/2)""N$
400 GOSUB520:GOTO330
410 PRINT""":GOTO500
420 POKECF,0:A$="":PRINT"INPUT TEXT:"
430 WAITKB,7:GETC$:A=ASC(C$):IFA=13THEN490
440 IFA=95THENPRINT""":GOTO520
450 IFA=20ANDLEN(A$)>0THENA$=MID$(A$,1,LEN(A$)-1):GOTO470
460 GOTO480
470 POKEDC,0:PRINT"  ||| ";:GOTO430
480 A$=A$+C$:PRINTC$:GOTO430
490 PRINT"  ":RETURN
500 GOSUB420:PRINT"YOUR TEXT WAS: "":PRINT:PRINTCHR$(34)A$CHR$(34)
510 GOSUB520:GOTO410
520 POKECL,20:PRINT:PRINT"AGAIN ? ";
530 POKECF,0:WAITKB,7:GETSP$:POKECF,1:PRINT"  ":IFSP$="Y"THENRETURN
540 IFSP$="N"THENRUN
550 PRINT""":GOTO530
999 PRINT:END

```

MTU Visible Memory  
Keyword Graphics Software  
A Product Review

Type: Hardware and software  
Model PET: 8K and up, all ROM versions  
Source: Micro Technology Unlimited  
2806 Hillsborough St.  
PO Box 12106  
Raleigh, NC 27605  
Price: \$434 for K-1008-40 package (old ROMs)  
\$483 for K-1008-41 package (new ROMs)

by Ralph Bressler

Much has been said about this excellent product from MTU in many different magazines. This is meant as a summary of the best points, a mention of some problems and a look at some things I have not seen mentioned before.

Each of the packages above includes all the parts you need to produce fine high resolution graphics on your PET screen. This includes a high-res board with 8K of RAM memory, a connector board, mounting brackets and software programs. This allows you to individually control 64,000 points on the screen rather than the 1000 available normally. The boards also have 5 expansion ROM slots which replace the three found on the PET main logic board. Installing the package is not for the complete novice but is easily done by someone experienced with simple circuit diagrams and a soldering iron. Installing the boards inside the PET blocks some air circulation and adds some heat. This may not be a problem but a small boxer fan will prevent heat build up. When you are not using the high-res function, the 8K of RAM is available for program storage on machines with less than 32K. Some initial hookup and operation problems were solved by several phone calls to MTU. The people at MTU are very qualified and know how to help. The documentation includes a hardware manual explaining the installation, testing and operation of the board. It includes extensive directions and detailed circuit diagrams. The software comes with its own manual which is equally good. In general, this kind of equipment is not for the real beginner but is great for those that have a specific need.

Keyword Graphics

This is the new software package for the MTU hi-res board and is far superior to the old package. The software is written in 6502 machine language which makes it super fast. Also included is an excellent demonstration program. The manual is extensive but assumes some knowledge which not all of us may have. It has unbelievable versatility and adds 45 new commands to BASIC. Lines can easily be drawn, points plotted, and text manipulated. Drawings can be scaled or rotated and coordinates offset. Up to four independent windows can be defined. What occurs in each window is independent of the others. Shapes may be predefined and stored in a shape table for future use. In fact, one shape can be defined as a collection of other shapes. The commands are relatively easy to use but getting the full potential out of the board requires a lot of experience. The points on the screen are labeled from 0,0 at the lower left to 319,199 in the upper right. Some examples of the commands used to control the board might be:

LINE 35,21,117,73	- draws a line from point 35,21 to 117,73
RVSDSP	- displays black points on a green background
SCFLIP 10,10,100,100	- flips the region defined by points 10,10 and 100,100
MOVE 200,10	- moves the drawing cursor to point 200,10
DRAW 319,199	- draws a line from the cursor to the point 319,199
AUTEXT "HELLO"	- displays the characters indicated using the



CHROT 2  
CHSCALE 1

character set supplied by MTU  
- rotate character printing by 2 \* 90 degrees  
- increase size of characters by a factor of 2

### ROM Slots and RAM

As mentioned before, there are 5 ROM slots on the MTU board. These slots can easily be set for addresses \$9000, \$A000, or \$B000 by simply changing a jumper wire. Any number of ROMs can be at the same location and are selected or enabled by a simple POKE command. This means you can have ROMs with the same address on the board and use them when you wish.

Eventually, you may run out of ROM slots even though 5 are provided on the board. My PET is a 32K which means the 8K of RAM on the MTU board is not doing any work when I am not using hi-res graphics. This RAM is located at addresses \$9000 and \$A000, the same locations as many of the ROMs now on the market. What I have done is to use the ML monitor to save the code contained in certain ROMs such as WordPro and the Monjana monitor onto disk. When I need to use the ML found in these ROMs I simply load the code from disk into RAM. This totally eliminates the need for the chip and frees up a previously used slot. These features are an extra added feature which can be very handy.

I would recommend this MTU product and any other to those who have a need to use them and some technical knowledge to get the most out of them.

### PET/CBM Personal Computer Guide Second Edition A Product Review

Type: Book

by Ralph Bressler

Model PET: All from 'old' to BASIC 4.0

Source: OSBORNE/McGraw-Hill  
630 Bancroft Way  
Berkeley, CA 94710

Price: \$15

This book is the second edition of the popular PET/CBM Guide. Adam Osborne and Carroll Donahue have revised the first edition and added a lot of new material. The primary difference is the inclusion of material for ALL PET/CBM computers. Common material is presented and then the differences are pointed out. Information about the PET peripherals is provided in chapter 6. The information is remarkably complete and well presented. There really is something for everyone from the beginner to the more advanced programmer. Not only is programming information provided but the maintenance and testing of equipment is also included. Even the 8000 series machines are dealt with! Chapter 7 contains SYSTEM INFORMATION including the system organization, memory map, assembly language programming and random access files. All program listings are done on a PET printer so that you can see exactly what to type in on your own machine. The appendices contain invaluable information such as character codes, error messages, conversion tables, and memory maps. Remember that this information is included for the three major releases of PET BASIC.

This book is well worth the \$15 price and I have only one major complaint. In their effort to provide information on three BASICS and three or four different machines the authors have made things a bit choppy. In places there is a short paragraph for YOUR machine followed by several paragraphs for other versions. The reader simply has to watch for the sections which apply to all machines or his specifically. However, sections are clearly marked and a complete index is provided.

## Using WordPro 4 Plus & NEC Spinwriter 5530P

ADA1600 Interface, CBM 8032+4040, Skyles Socket 2ME, Nestar Toolkit

by "Dr. Guenther Badeau"

Yesterday at Winter Consumer Electronics Show a Commodore territorial sales manager was uncanning WordCraft, a text editor which he said is from England and which Commodore would distribute in this country. I saw nothing in that program superior to WordPro 4 Plus, which I've had 2 or 3 months and like a lot. You can see for yourself in the title above that WordPro 4 Plus permits bold on the NEC; also super- and sub-scripting and very easy output to disk file which is then readable by your BASIC program.

I should have gotten a Spinwriter with keyboard so as to print continuous forms using WordPro 4 Plus, do a "pause" to go Local on the Spinwriter while I typewrite through it onto the paper, then back to Remote and let WordPro 4 Plus finish up the print. I want to do that with accounts receivables and on my school's kids' report cards. Bob Crowell at NEECO furnished this round of equipment in a gentlemanly way but hasn't responded to my request to spend a little more with him and change printers. The NEC is a very marvelous product. Print quality compares with IBM typewriters' best. NEC seems built to last and last. If you can buy something that just goes on working indefinitely you get less concerned with the cost.

"Our" printer is NEBS (just starting advertising in micro publications) and they are excellent. I made a mistake while ordering continuous checks and they replaced them free. They insisted it was to be free if I couldn't use what I had. I prepay orders to them and in return they U.P.S. it from Massachusetts to me in California at no extra cost. Door-to-door turnaround time is 14 calendar days or less. WordPro 4 Plus and the NEC are happy to process hand-fed consumer-sized separate cut checks, and did fine the other day on year end W-2 forms which are free from the government 3 to page and require manual insertion of five carbons. Page 6's carbon was entirely legible.

All the Commodore hardware has worked fine; this has also been true on several earlier sets of machines. Their documentation has improved from non-existent up to minimally acceptable (8032 manuals are entirely silent on the whole business of a screen window, and as always lack an index). We gave up the 2022 Printer for improved print quality and "real" descenders on the lower-case characters, 2 aspects we think important in our elementary school usage. The kids, all ages, use all this stuff every day. That's trial by fire.

Why do a piece for the slick magazines? They pay about enough per page to cover the postage getting it to them, so what can be expected? At CES there must have been AT LEAST forty trade and special-interest magazines shown, a mudslide matched in volume only by the porno video cassette hawkers. I neither write nor register at shows with my real name, incidentally, to facilitate junk-mail processing. That leaves more time for kids and computers.

Bob Skyles is the best of the bunch, the human-equivalent of the NEC in reliability. His prime advantage is that he will talk to you on the phone, besides which what he sells works, he ships it as promised and has a few other curious nice habits.

PETSET1  
A Product Review

Type: Hardware  
Model PET: 8K and up  
Source: Connecticut microComputer, Inc.  
34 Del Mar Drive  
Brookfield, CT 06804  
Price: \$295

by Ed McDaniels

The PETSET1 is a system that converts quantities such as temperature and light into digital bits of information that is fed directly into the computer and can be immediately processed by the PET. This type of peripheral is known as an analog to digital converter or an A to D. In essence this converter takes any input that can be represented as a voltage and changes it to a value from 0 to 255 that the PET can read as data.

In its literature, the company lists a large number of physical, mechanical and chemical quantities that can be measured with the appropriate sensors. To date the only two sensors that the company has developed has been for temperature and light. Once again reality has not lived up to promise. The trick for the experimentalist is to be able to represent quantities such as velocity, acceleration, humidity, pH, etc. as voltages. Both circuits and software must be designed to first create the voltage (0-5.1V) and then to evaluate the information. The lack of available sensors is a serious limitation. It mandates that the computerist be both knowledgeable and inventive in constructing his own sensors and sensing circuits as well as being able to create the programs that will use these digital bits.

The documentation of the PETSET1 is adequate. CmC explains what an A to D converter is, how the process is accomplished and a sample program for measuring temperature.

Physically the PETSET1 is a system consisting of a 16 channel-8 bit Analog Input Module (AIM-16), a Power Module (POW1), a Manifold Module (MANMOD), a PET interface (PETMOD) and a connecting cable. The PETMOD plugs into the J1 and J2 ports but does not restrict their access since there are extra edge connectors that still allow you to connect peripherals such disk or printer.

The 16 channels of the AIM-16 can be separately monitored so that the temperature at 16 different locations can be measured and recorded. Since the scanning of all 16 channels is accomplished in less than 1.5 milliseconds an almost continuous readout is possible.

The latest literature from CmC lists the price of the temperature sensor at \$69.95 and the light sensor at \$89.95. There is also a SUPER X10 MOD that will act as a remote controller of devices connected to BSR remote modules and has a programmable clock and calendar built in.

I teach Physics and Electronics on the high school level so that the PETSET1 has a number of applications in my classroom. One use is in measuring the rate of charging in a circuit containing resistance and capacitance. It allows me to measure the percentage of charge on a capacitor at one second intervals. The results are extremely reproduceable and exact. To duplicate these results by other methods would be difficult. Another exciting application combines a linear air track with the PETSET1 to make very quick and accurate measurements of velocity and accelerations. The students don't get lost in a maze of dotted paper and the computer can be programmed to either allow the student to calculate everything on their own or have the computer do the computations.

I see the PETSET1 as a good beginning by the industry to directly connect the computer to the real world. Why not have the computer monitor all those tedious, intermittent or brief measurements. I'm all for the machine doing those things that machines do best and letting the people express their creativity in other activities.



IDPC FORTH  
A Product Review

Type: Software  
Model PET: New ROMs with 2040 disk  
Source: IDPC Co.  
Box 11594  
Philadelphia, PA 19116  
Price: \$65

by Ralph Bressler

AB Computers  
115 E. Stump Rd  
Montgomeryville, PA 18936

This software package contains a disk and four separate pieces of documentation. The disk has a full implementation of the FORTH language for the PET. This includes the language itself, an editor and a full 6502 assembler. Integers, floating point numbers, double precision numbers and strings are all handled by this package. It conforms to the FORTH Interest Group standards. The documentation includes an Installation Guide which explains how to get the language up and running on the PET. It also includes some of the 'words' that have been added to the language to handle the peculiarities of the PET. The second 'manual' is called GETTING STARTED and it walks you slowly through the beginnings of FORTH. The USER'S GUIDE goes into more depth using more complex examples including disk operations. Finally, the GLOSSARY defines all of the 'words' and gives examples of their use.

It is impossible to review this package without saying something about FORTH itself. Before working with this package I knew NOTHING about FORTH. I am far from an expert now but this package has taught me an immense amount about FORTH. Using the package is as simple as loading it from disk. After this you are ready to write in FORTH.

FORTH is called a 'threaded language'. It has 'words' which have definitions and tell the computer to perform some action. Certain words come in the standard FORTH vocabulary but users may combine these words to define new words. It is possible, although not advisable, to create one word that will play an entire game! FORTH is a stack oriented language. The stacks are a 'last in first out' or LIFO structure. The stacks are the data, dictionary and return stack. The data stack operates like RPN notation. If you wanted to add two numbers you might type:

```
3 2 + .
```

FORTH would then respond with 5 since the + means add and the . means output the top value on the stack. There are numerous 'words' which manipulate the stack by swapping, duplicating, and copying values. Defining a word might look something like this:

```
: WAIT 6 23 CPOS ." HIT SPACE TO GO ON" CR  
  BEGIN GET 32 = UNTIL CR ;
```

This defines the word WAIT but does nothing until the word WAIT is used later. When we use the word WAIT the cursor will be positioned (CPOS) at line 23 and column 6 and the message, HIT SPACE TO GO ON, will be printed. The BEGIN-UNTIL structure will wait until the character gotten by GET is ASCII 32; a space.

To write a program in FORTH you use the word EDIT which calls the FORTH editor. A program is written one screen at a time and is, therefore, called a screen. Screens may be linked together for long programs. When writing a screen you may use predefined words, define your own new words or use the 6502 assembler. To run the program the word LOAD is used.

In summary FORTH is very different from any other language I have experienced. It is interesting because of the ability to define a whole new set of words for special applications.

This implementation of FORTH is very complete and reasonably well documented. By itself it will not teach you everything about FORTH but it gives you a good start. The examples included are extensive and clear. The assembler has a strange syntax but is very versatile since you may use certain looping structures. The \$65 price for a completely new language is fantastic. This is not a game or fun piece of software. It is a genuine implementation of a full language.

Some of my criticisms are of FORTH and some of this implementation. In the direct mode typing certain keys, some graphics and the RVS key, will permanently hang up the PET. This is annoying since mistakes are common. There is no facility to generate random numbers and my efforts have not been successful. This eliminates most games. When writing a screen to disk a direct block-write is used. This means that the disk appears to have only one program, FORTH, when a directory is done. To really find what is stored on the disk you must load FORTH. The range of numbers that this FORTH handles is from -32767 to 32767. This means that working with the screen RAM locations can be a pain.

Overall, I would say that any serious programmer should purchase this FORTH package. Others are available and will be reviewed soon. Users should also contact the FORTH Interest Group and get a copy of the August 1980 BYTE. This issue of BYTE was devoted to FORTH and gives a lot of useful information.

### Spacemaker II A Product Review

Type: Hardware  
Model PET: New model PETs  
Source: CGRS Microtech  
PO Box 102  
Langhorne, PA 19047  
Price: \$39

by Ralph Bressler

Are you running out of ROM space? Are your sockets crammed to the hilt? What will you do? SPACEMAKER II to the rescue. You will remember that the original Spacemaker was a small PC board with two ROM sockets. This board plugged into any ROM socket and allowed you to put two ROMs at the same address. A small toggle switch let you access either ROM. It had a few problems though. It would not handle the popular 2716 EPROM without some modifications and, after all, it only had two slots. Enter SPACEMAKER II with four ROM sockets. This makes it twice as good as the original right? Wrong! Its more than twice as good.

SPACEMAKER II has four ROM slots and plugs into any ROM socket, but it is the same size as the original. A four position slide switch allows you to manually choose which ROM to use. Not only will it handle 2716's but small jumpers above each socket allow you to have a mixture of any of 9 different chips on the same board. This board will even work in the character generator ROM slot. This means you can have old PET and new PET characters on the same machine. It also means you can use the Foreign language and Math character generators from West River Electronics.

One thing you may note about the SPACEMAKER is several pins sticking up at the top of the board. These are connected to the control lines on the board. CGRS has hardware and software accessories which will allow you to switch ROMs from the keyboard or under software control. ROMDRIVER is a PC board which allows you to do this switching. If you don't want to write your own software to use ROMDRIVER then CGRS will supply ROM I/O. This is a software package that allows you to more easily use the ROMDRIVER and SPACEMAKER together. These latter products will be more fully reviewed in another issue.

## PET Magazines

by Ralph Bressler

The following list represents the magazines that deal with the PET. There are no magazines devoted entirely to the PET and the only independent US newsletter that is exclusively PET is The PAPER. The comments included are my own as I perceive the magazines usefulness to PET owners. The only magazines included are ones I have seen and read consistently. The list is in order of most to least interesting to PET users.

### COMPUTE

Published: monthly  
Price: \$16.00/yr  
Address: Circulation Dept.  
COMPUTE Magazine  
PO Box 5406  
Greensboro, NC 27403

The best magazine for general 6502 and PET users. Format includes a section which applies to 6502 machines in general and also has some machine specific articles. Separate sections have articles for Apple, Atari, PET, OSI and single board computers. Articles are by top authors and experts in the field. Includes programs, tricks and reviews of software and hardware. Many articles lean a little toward more advanced programmers. On going series dedicated to computers in education.

### Cursor

Published: bimonthly  
Price: \$27  
Address: The Code Works  
Box 550  
Goleta, CA 93116

A tape cassette and several pages of notes come to your house every other month. The tape always loads and has a "cover" and four or five programs. Programs run on all ROMs and most new ones even work on 80 column screens. Back issues available for \$4.95 each. One of the best ways to learn BASIC programs is to look at good programs. These are good programs and will teach you a lot. Most are either enjoyable or useful or both.

### Microcomputing

Published: monthly  
Price: \$25  
Address: Microcomputing  
PO Box 997  
Farmingdale, NY 11737

A nice balance of articles on hardware C and software. Regular PET column by Bob Baker which includes in depth reviews, programs and tips. Frequent 6502 and PET articles. Many ideas appear that are not found other places. Extensive advertisements are interesting.

## MICRO

Published: monthly  
Price: \$15  
Address: Micro  
PO Box 6502  
Chelmsford, MA 01824

Completely 6502 oriented with many PET specific articles. A mixture of hardware, software, BASIC and machine language. Not for real beginners. Recent overabundance of Apple articles but publisher is trying to solve this "problem". Good PET column.

## Creative Computing

Published: monthly  
Price: \$20  
Address: Creative Computing  
PO Box 789-M  
Morristown, NJ 07960

Includes a little bit of everything from philosophy to cartoons to BASIC and other languages. Emphasis on micros. Most programs and programming ideas are adaptable to the PET. Interesting Legal Forum dealing with the laws governing computers and software copyrights. Some issues have themes (education, artificial intelligence). PET column by Greg Yob is variable and is not always included.

## Recreational Computing

Published: bimonthly  
Price: \$12  
Address: Recreational Computing  
Subscription Service  
PO Box E Dept L6  
Menlo Park, CA 94025

Concentrates on computer theory and applications of micros. A few articles related to PET. Published by People's Computer Company, a pioneer. This organization has several other interesting projects including a phone communications systems for the PET.

## BYTE

Published: monthly  
Price: \$19  
Address: BYTE Subscriptions  
POB 590  
Martinsville, NJ 08836

No regular PET column and few specific PET related articles. Lots of general computer information but little relates directly to the 6502. Traditional hardware orientation is changing to software. Many issues theme oriented. Magazine is thick (350+ pages) but must be 50% ads. Ads are interesting and informative.



# PET Memory Map

- 10=A
- 11=B
- 12=C
- 13=D
- 14=E
- 15=F

HEX    DECIMAL

\$FFF	4095
\$F00	3840
\$E00	3584
\$D00	3328
\$C00	3072
\$B00	2816
\$A00	2560
\$900	2304
\$800	2048
\$700	1792
\$600	1536
\$500	1280
\$400	1024
\$300	768
\$200	512
\$100	256
\$000	0

4K

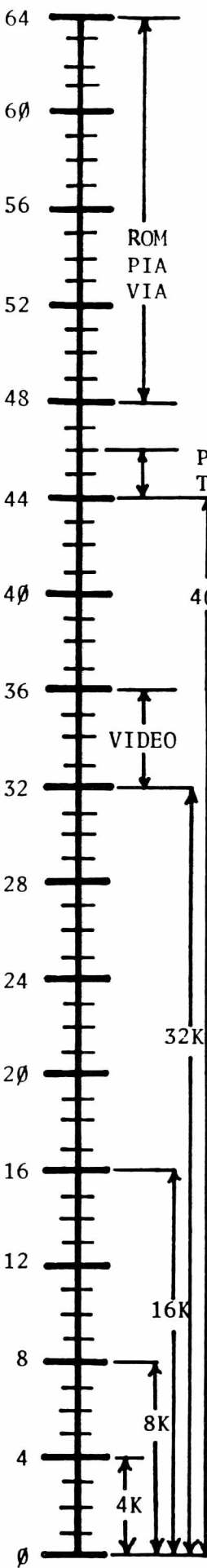
\$FF	255
\$F0	240
\$E0	224
\$D0	208
\$C0	192
\$B0	176
\$A0	160
\$90	144
\$80	128
\$70	112
\$60	96
\$50	80
\$40	64
\$30	48
\$20	32
\$10	16
\$00	0

Page 0

HEX    DECIMAL

\$FFFF	65535
\$F000	61440
\$E000	57344
\$D000	53248
\$C000	49152
\$B000	45056
\$A000	40960
\$9000	36864
\$8000	32768
\$7000	28672
\$6000	24576
\$5000	20480
\$4000	16384
\$3000	12288
\$2000	8192
\$1000	4096
\$0000	0

K



## EDUCATIONAL ACTIVITIES

Bridge the Classroom Literacy Gap. Two series of full color filmstrips with audio cassettes and teacher's manuals present a systematic approach to programming. The programs provide a set of step-by-step instructions to guide students through the BASIC language. These filmstrips can be used at any grade level to teach students the concepts of programming a microcomputer in BASIC. The manuals include program listings, comprehension questions and program assignments. Both sets are designed by Arnold and Seth Friedman.

### BASIC for Microcomputers

Includes *Getting Started*, *Mathematical Operations*, *Loops and Subroutines*, *Original Programming Techniques* and *Flowcharting*. Designed for beginners in programming.

5 Filmstrips and Audio Cassettes - \$89.00

### Advanced BASIC Techniques

A sequel to the popular series shown above. Topics discussed are *String Functions*, *Data Handling and Arrays*, *Computer Graphics* and *Advanced Math Functions*. This series is meant for students who understand the concepts presented in *BASIC for Microcomputers*.

4 Filmstrips and Audio Cassettes - \$74.00

Both series available for preview.

We also have many educational programs for the PET, Apple and TRS-80.

P. O. Box 392  
Freeport, NY 11520

## Finally, MAE — A PET DISK-Based MACRO ASSEMBLER/TEXT EDITOR

Works with 32K PET

- Works with 2040 Disk, and can drive 2022/2023 Printer, and/or RS232/20 ma Device thru User Port.
- 100% Disk Based, 100% Machine Language.
- Macros, Conditional Assembly, and a new feature we developed called Interactive Assembly.
- Coexists with Basic, Auto character repeat, Sorted Symbol Table.
- 27 Commands, 26 Pseudo Ops, 5 Conditional Ops, 38 Error Codes.
- Creates relocatable object code on disk.
- Assemble from Memory or Disk.
- String search, search and replace, and inter-line edit.
- Auto line #-ing, move, copy, delete, renumber.
- Labels up to 31 characters — user specifies length.
- Includes extension to PET monitor (disassemble, trace, etc.), Library of PET ROM locations, Relocating Loader, plus more.

Manual, Diskette, U. S. postage — \$169.95  
(Requires completion of License Agreement —  
Write for details)

## Eastern House Software

3239 Linda Drive

Winston-Salem, N. C. 27106

## PET RABBIT



Load, Save, Verify, Execute  
8 K in 38 seconds versus  
PETs 2 Minute 45 seconds,  
plus more!

High-speed Cassette Routines work with 8K, 16K, or 32K new ROM PETs which have the new Commodore cassette deck (like the external version which sells for \$95.00). Note: If you have a new ROM PET with the old style lift-top deck, everything but the high-speed cassette routines will work.

- Auto repeat of any key held down, toggle character set.
- RAM Memory Test, convert #'s to hex and decimal.

### 12 Rabbit Commands

Note: Rabbit is 2K of machine code at \$1800 for 8K PETS, \$3000 or \$3800 for 16K PETS, or \$7000 or \$7800 for 32K PETS. (Specify one of the 5 versions.)

Cassette and Manual — \$29.95 (Add \$5.00 for foreign)

## Eastern House Software

3239 Linda Dr.

Winston-Salem, N. C. 27106

Microphys is pleased to announce the release of a series of twelve programs designed for use in *introductory calculus courses* on both the high school and college levels.

The programs are intended for use with a Commodore PET/CBM microcomputer having at least 8K of storage. Each program is recorded on a C-10 cassette and is accompanied by simple descriptive instructions. The programs retail for \$20 each. A diskette containing all 12 programs may be obtained at a cost of \$180.

Each program generates a unique set of problems for each student. Answers may be generated so that the student may check his own work or these answers may be suppressed. The student then solves his set of problems away from the computer. When his work is completed, the student enters his code number and answers and the computer will then grade his work, displaying the answers to those questions which were incorrectly solved; a percent score and a brief comment reflecting an overall evaluation are also given.

### Individual Program Content

- PC726- Differentiation of Algebraic Functions
- PC727- Maxima/Minima Problems: Part I
- PC728- Maxima/Minima Problems: Part II
- PC729- Relative Rate Problems: Part I
- PC730- Relative Rate Problems: Part II
- PC731- Integration of Algebraic Functions
- PC732- Differentiation of Trigonometric Functions
- PC733- Integration of Trigonometric Functions
- PC734- Integration: Areas of Plane Figures
- PC735- Integration: Volumes of Solids
- PC736- Integration: Arc Lengths
- PC737- Integration: Surface Areas of Solids

Note: All programs are available from your local computer dealer. They may also be obtained directly from Microphys.

Educators: Be sure to write for our free educational software catalogue which describes over 140 programs for use in Chemistry, Physics, Math, English Vocabulary and Spelling. These programs are designed for use on the PET with a minimum of 8K of storage.

Dealer Inquires Invited

MICROPHYS PROGRAMS

2048 Ford St.

Bklyn, N. Y. 11229

(212) 646-0140

# AFFORDABLE QUALITY



Heavy duty SELECTRIC printers

Typewriter (15" carriage)	\$ 599
for CBM, Apple	\$1195
for TRS-80, Atari	\$ 995
NEC Spinwriter #5530	\$2695
Xymec typewriter/printer	\$2499
Centronics 737	\$ 839
Microline 80 (Okidata)	\$ 599
Epson MX-80	\$ 549
ESCON Parallel Interface	\$ 549
TU-65C (CBM/Centronics)	\$ 129.95
TU-8014 (TRS-80/RS-232)	\$ 69.95

\*prices may be changed without notice

## **Virginia Microsystems**

14415 Jefferson Davis Highway      VA add 4%  
Woodbridge, Virginia    22191    (703) 491-6502

*Did You Complete  
Our Survey?  
See Center!*

## CENTERBROOK SOFTWARE DESIGNS

*"Learning is Fun"*

Remedial Math and Reading  
Science, Utilities, and Games

We have over 70 programs in the topics listed above. All programs work in 8k and there are versions for all PETs. Prices for most programs are in the \$10 to \$20 range and special packages are available. New programs are produced every month and custom programming is available.

### SPECIALS

(Until June 15, 1981)

Any 6 Math Programs

normally \$90, **now only \$60**

Science Package - biology & chemistry

8 programs sell for \$95

now only \$50

**NEW** Programming Pack SCARDIAC-simulates  
Bell Tel CARDIAC-intro to machine language  
programming

POGOG - a LOGO-like or turtle graphics  
language

Regular \$50 **NOW ONLY \$20**

Both work on 8K of memory

*For a catalog and more information write or call:*

**Centerbrook Software Designs**

98 Emily Drive

Centereach, NY 11720

516-585-2402

## DISK MASTER

**\$12.00 On Disk**

The first disk cataloging program to be introduced for the Commodore PET/CBM with a 2040 disk. Now you can easily catalog as many as 140 diskettes, forming a large master directory on a single diskette. The program automatically reads the directory blocks of any disk being cataloged, so no typing is involved. File searching, disk ID lists, and other useful functions are also provided.

## RIBBONS

for your PET/CBM 2022/2023 printers

**\$20.00/Dozen**

## MEMOREX DISKETTES

Single side, single density, w/hub rings

**\$25/Box of 10**

\$2.00 Shipping Per Order

VISA/MC Accepted - NJ Residents Add 5% Tax

SASE For Current Program List

## BAKER ENTERPRISES

15 Windsor Drive

Atco, NJ 08004

## B. C. COMMUNICATIONS

*The Comprehensive Computer Store*

Authorized dealers and factory service Center for Commodore and Apple.  
Factory trained personnel.

A full line of Centronics printers and NEC Spinwriters.

Before you buy come see what you're buying!

- All models of Commodore computers and peripherals including the new 8000 series 80 column business machines.
- Apple II and the new Apple III business computer. Also Apple peripherals.
- Hands-on word processing demonstrations on the CBM or Apple.
- Most major magazines and newsletters as well as manuals and other computer supplies.

We also offer:

- Word processing services.
- Consultations on systems configurations.
- A place to meet and swap ideas and exchange programs.

### SYSTEMS FOR EDUCATIONAL, BUSINESS OR PERSONAL USE

**10% Discount On All Computer Equipment % Expires June 30, 1981**

**B. C. COMMUNICATIONS**

**207 Depot Road, P.O. Box 228**

**Huntington Station, NY 11746**

**(516) 549-8833 or 692-2735**

## PET™ 2001 STUDENT WORKBOOKS

For Classes Beginning to Program

### COMPLETE TEACHING UNITS

- Step-By-Step Instruction (Teachers Can Be Beginners, Too!)
- Daily Lesson Plans, Classwork & Homework Worksheets, Quizzes
- Hands-On and Hands-Off Exercises
- Group and Individual Assignments

**FEED ME, I'M YOUR PET™** (Beginner-1)

**LOOKING GOOD WITH YOUR PET** (Intermediate-1)

**TEACHER'S PET** (Plans, Quizzes, Answer Key)

Workbooks are \$4.35. Quantity discounts available.

**TEACHER'S PET** is \$4.00 or **FREE** with workbook orders of 25 or more.

### EDUCATIONAL SOFTWARE

- High Quality Teaching Skills and Programming Skills
- Graphics Used to Enhance Learning

**REMEDIAL MATH PACKAGE** (84 Concepts)

**MATH IN SCIENCE SERIES**

Signed Numbers, Physics, Primary Primer, Elementary Grade topics and More. We Will Send a Price List.

## COW BAY COMPUTING

Box 515 • Manhasset, NY 11030

F.O.B. Manhasset, NY

PET & APPLE II USERS

## TINY PASCAL

Plus+  
GRAPHICS



The TINY Pascal System turns your APPLE II micro into a 16-bit P-machine. You too can learn the language that is slated to become the successor to BASIC. TINY Pascal offers the following:

- **LINE EDITOR** to create, modify and maintain source
- **COMPILER** to produce P-code, the assembly language of the P-machine
- **INTERPRETER** to execute the compiled P-code (has TRACE)
- **Structured programmed constructs:** CASE-OF-ELSE, WHILE-DO, IF-THEN-ELSE, REPEAT-UNTIL, FOR-TO/DOWNTO-DO, BEGIN-END, MEM, CONST, VAR ARRAY

Our new TINY Pascal PLUS+ provides graphics and other builtin functions: GRAPHICS, PLOT, POINT, TEXT, INKEY, ABS AND SQR. The PET version supports double density plotting on 40 column screen giving 80 x 50 plot positions. The APPLE II version supports LORES and for ROM APPLESOFT owners the HIRES graphics plus other features with: COLOR, HGRAPHICS, HCOLOR, HPLOT, PDL and TONE. For those who do not require graphics capabilities, you may still order our original Tiny Pascal package.

#### TINY Pascal PLUS+ GRAPHICS VERSION:

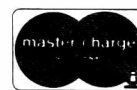
PET 32K NEW Roms cassette.....\$55  
 PET 32K NEW Roms diskette.....\$50  
 APPLE II 32K/48K w/DOS 3.2 or 3.3.....\$50

#### TINY Pascal NON-GRAPHICS VERSIONS:

PET 16K/32K NEW Roms cassette.....\$40  
 PET 16K/32K NEW Roms diskette.....\$35  
 APPLE II w/ROM Applesoft 32K w/DOS.....\$35  
 APPLE II w/ROM Applesoft 48K w/DOS.....\$35

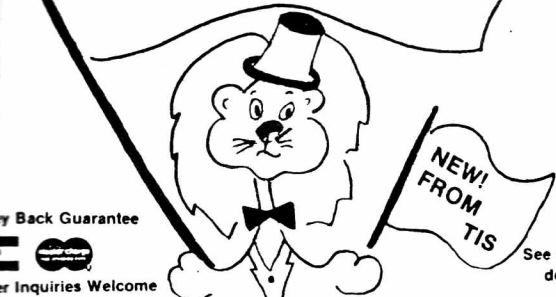
USER'S Manual (refundable with software order).....\$10  
 6502 Assembly Listing of INTERPRETER-graphics.....\$25  
 6502 Assembly Listing of INTERPRETER-non graphics.....\$20

FREE postage in U.S. and CANADA. Orders may be prepaid of by bankcard (include card number and expiration date). Michigan residents include 4% state sales tax. Orders accepted via THE SOURCE - CLO852.



**ABACUS SOFTWARE**  
 P. O. Box 7211  
 Grand Rapids, Michigan 49510

## UNDERSTANDING YOUR PET/CBM Volume I: Basic Programming



Money Back Guarantee



Dealer Inquiries Welcome

See your dealer

New 248-page book includes all the former TIS workbooks except "PET Graphics." Provides information for both ROMs and a comprehensive index. Only \$14.95.

Also from TIS

WB-3 PET Graphics \$4.95

Software products on cassette or floppy disk with complete instruction manual. Each \$24.95 (cassette), \$29.95 (diskette).

- SW-1 MAIL B mailing list system
- SW-2 CHECKBOOK record
- SW-3 ACCOUNTS keep track of who owes you how much
- SW-4 MEDIT create and maintain date files
- SW-5 CALENDAR appointments, meetings at-a-glance

**TIS**  
 P.O. Box 921, Dept. C  
 Los Alamos, NM 87544

Add \$2 (\$5 foreign orders)  
 shipping and handling

PET and CBM are trademarks of Commodore Business Machines

## COMPUTER LAND of Nassau "The Full Service Computer Store"

Come **SEE** what you've only **READ** about.

Hardware... Apple... PET... Atari... Texas Instruments... Cromemco... Dynabyte... computers for personal and professional needs... Diablo... Centronics... Spinwriter... printers and disk drives to complete a system.

Software... programs for **ALL** the machines we carry... programs for recreation, education and business.

Books and Magazines... Osborne... Blacksburg Group... Hayden... Sybex... Scelbi... or pick up a copy of your favorite magazine.

*Watch For New  
 Store In Suffolk.  
 Coming Soon!*

*We Now Carry  
 Vector Graphics  
 Products*

**Computer land**  
**79 Westbury Avenue**  
**Carle Place, NY 11514**



# Skyles Electric Works

**BASIC Programmer's, Toolkit™, Disk-O-Pro™, Command-O™**

## **For PET™ Owners Who Want More Fun And Fewer Errors with Their Programs**

Here are thirty-five commands you'll need, all on dual chips you can install in two minutes without tools, **on any PET or PET system**. 2KB or 4KB of ROM firmware on each chip with a collection of machine language programs available from the time you turn on your PET to the time you shut it off. No tape to load or to interfere with any running programs.

For PET/CBM 2001-8, -8N, -16N/B, -32N/B, 3016 and 3032

### **BASIC Programmers Toolkit™ commands**

**AUTO<sup>ed</sup> DELETE<sup>ed</sup> RENUMBER<sup>ed</sup> HELP<sup>ed</sup> TRACE<sup>ed</sup>  
STEP<sup>ed</sup> OFF<sup>ed</sup> APPEND<sup>ed</sup> DUMP<sup>ed</sup> FIND<sup>ed</sup>**

### **BASIC Programmers Disk-O-Pro™**

**CONCAT<sup>B80</sup> DOPEN<sup>B80</sup> DCLOSE<sup>B80</sup> RECORD<sup>B80</sup> HEADER<sup>B80</sup> COLLECT<sup>B80</sup>  
BACKUP<sup>B80</sup> COPY<sup>B80</sup> APPEND<sup>B80</sup> DSAVE<sup>B80</sup> DLOAD<sup>B80</sup> CATALOG<sup>B80</sup>  
RENAME<sup>B80</sup> SCRATCH<sup>B80</sup> DIRECTORY<sup>B80</sup> INITIALIZE<sup>BS</sup> MERGE<sup>BS</sup> EXECUTE<sup>BS</sup>  
SCROLL<sup>ed</sup> OUT<sup>ed</sup> SET<sup>ed</sup> KILL<sup>ed</sup> EAT<sup>ed</sup> PRINT USING<sup>BS</sup> SEND<sup>BS</sup> BEEP<sup>BS</sup>**

```
RUN
?DIVISION BY ZERO ERROR IN 500
READY
HELP
500 J = SORIA*B/J)
READY
```

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

```
RUN
READY
DUMP
A1 = 10
BW = -61
CS = 'M'
READY
```

### **NOTES:**

**ed** — a program editing and debugging command  
**B80** — a BASIC command also available on Commodore CBM™ 8016 and 8032 computers.  
**BS** — a Skyles Electric Works added value BASIC command.  
BASIC Programmers Toolkit™ is a trademark of Palo Alto IC's.  
BASIC Programmers Disk-O-Pro™, Command-O™ are trademarks of Skyles Electric Works.  
PET™, CBM™ are trademarks of Commodore Business Machines.

Phone or write for information. We'll be delighted to answer any questions  
and to send you the complete information package.



## **Skyles Electric Works**

231 E South Whisman Road  
Mountain View, CA 94041  
(415) 965-1735



# Skyles Electric Works

**BASIC Programmer's, Toolkit™, Disk-O-Pro™, Command-O™**

## For CBM™ Owners Who Want More Fun And Fewer Errors with Their Programs

Here are nineteen commands you'll need, on a single chip you can install in two minutes without tools, on any **CBM** or **CMB** system. 4KB of ROM firmware on each chip with a collection of machine language programs available from the time you turn on your PET to the time you shut it off.

For CBM 8016 and 8032; BASIC 4.0

### BASIC Programmers Command-O™

**AUTO<sup>ed</sup> DUMP<sup>ed</sup> DELETE<sup>ed</sup> FIND<sup>ed</sup> (improved) HELP<sup>ed</sup> KILL<sup>ed</sup> OFF<sup>ed</sup> TRACE<sup>ed</sup> (improved) RENUMBER<sup>ed</sup> (improved) INITIALIZE<sup>BS</sup> MERGE<sup>BS</sup> MOVE<sup>BS</sup> EXECUTE<sup>BS</sup> SCROLL<sup>ed</sup> OUT<sup>ed</sup> SET<sup>ed</sup> SEND<sup>BS</sup> PRINT USING<sup>BS</sup> BEEP<sup>BS</sup>**

```
100 GOSUB 180
105 PRINT USING CS, A, BS
130 INPUT "TIME", DS
131 INPUT "DAY", ES
160 IFB C THEN 105
180 FOR X IT09
183 PRINT Y(X), NEXT
184 RETURN
200 I X 19
READY
RENUMBER 110, 10, 105-184
READY
LIST
100 GOSUB 150
110 PRINT USING CS, A, BS
120 INPUT "TIME", DS
130 INPUT "DAY", ES
140 IFB C THEN 110
150 FOR X IT09
160 PRINT Y(X), NEXT
170 RETURN
200 I X 19
READY
```

```
MERGE D1 "BUY NOW"
SEARCHING FOR BUY NOW
LOADING
READY
RENUMBER 100, 10
READY
FIND BS
110 PRINT USING AS, SS, SS, CS, DS
280 SS "NOW IS THE TIME"
READY
```

```
580 BA = BA 1
590 RA = 123 * 5X.92 - BA * 10
600 IF BA - 143 THEN 580
610 RETURN
620 CS PROFIT $#, ***** DAILY
630 PRINT USING CS, PI
640 DS LOSS $#, ***** DAILY
650 PRINT USING DS, LI
RUN
PROFIT $1, 238.61 DAILY
LOSS $ 0.00 DAILY
READY
```



**NOTICE**



When you order **Command-O**, we will loan you a **Toolkit** until we deliver **Command-O**.



**NOTICE**



### PRICES:

- BASIC Programmers **Toolkit™** (chip only) \$40.00
- BASIC Programmers **Disk-O-Pro™** (chip only) \$75.00
- BASIC Programmers **Command-O™** (chip only) \$75.00
- Interface boards (needed sometimes) \$20.00-\$50.00
- Instruction Manual (with redeemable \$5.00 coupon) \$5.00

*Shipping and handling \$2.50 USA/Canada, \$10.00 Europe/Asia*

*California residents please add 6% or 6-1/2% sales tax as required*

*Reserve your **Disk-O-Pro**, **Command-O** today*

*Toolkit™ immediate delivery, Disk-O-Pro delivery in December, Command-O delivery in January*

**VISA, MASTERCARD ORDERS CALL (800) 538-3083 (except California residents)**

**CALIFORNIA ORDERS PLEASE CALL (408) 257-9140**



# Skyles Electric Works

**231 E South Whisman Road  
Mountain View, CA 94041  
(415) 965-1735**

# AMERICAN PERIPHERALS

*First In Education*

*Stop in at the showroom in Lindenhurst to see and try the collection of 6 brands of printers (including an IBM Selectric) hooked to the PET's. Try out the CBM VOICE Synthesizer. Use the Wantanbe Plotter for artistic equation art.*

## Printers For PET/CBM (all cables and interfaces included)

IBM Selectric Model 50 - IBM letter quality output .....	\$2,200
Xerox Diablo 630 - letter quality, daisy wheel printhead .....	\$3,200
NEC Spinwriter - letter quality, thimble printhead .....	\$3,300
Centronics 737 - high density dot matrix, near letter quality .....	\$1,200
Commodore Mannesmann Tally 8024 - 132 columns, high speed, bi-directional printing, business applications .....	\$1,750
Wantanabe MiPlot - flatbed plotter connects to PET .....	\$1200

Come in and examine our wide collection of textbooks and workbooks for the PET. Or maybe you are interested intrading-up your existing equipment to a larger system?

## Upgrade Options

Trade up from your old ROM PET to the new ROMs.  
You get a large keyboard and an improved BASIC.

Old 8K to New 8k .....	\$400
Old 8K to New 16K .....	\$600
Old 8K to New 32K .....	\$800

*Many, Many Other Upgrade Deals Available!*

*Our new 46-page educational Software Catalog is out!  
Call Alice or Jackie at 516-226-5849 for your copy.*

### Family Oriented Computer Games

Available for use with the PET,  
BINGO, DRAW POKER,  
WHEEL OF FORTUNE  
CRYPTOGRAMS, WORD GAMES

### Educational Software

Free educational catalog listing over  
140 programs in math, physics,  
chemistry, vocabulary, spelling.  
Write to:

**MICROPHYS**  
2048 Ford Street  
Brooklyn, NY 11229

### Programs!!



- Business Research ..... \$50
- Home & Small Business ..... \$15-\$40
- Games & Simulations ..... \$15 ea.
- Education (HS) Pack ..... \$15

- |            |                    |                |
|------------|--------------------|----------------|
| • Shopper  | • Home Addresser   | • Inventory    |
| • Museum!  | • Deluxe Addresser | • Dinner's On! |
| • Mansion! | • Fur Trapper      | • Pentagon!    |
|            | • High Seas        |                |

Send for complete catalog!

**HARRY R. BRILEY**  
P.O. Box 2913 • Livermore, CA 94550  
(415) 455-9139

### THE PAPER

Box 524  
East Setauket, NY 11733

THIRD CLASS POSTAGE

PAID

PERMIT NO. 96  
EAST SETAUKET, NY