THE PAPER

FOR OWNERS OF THE COMMODORE PET™ PERSONAL COMPUTER

VOLUME II, ISSUE 3 An ARESCO Publication April, 1979 $2.00

INTRODUCTION TO ASSEMBLY LANGUAGE PROGRAMMING

By Rick Simpson

We published a questionnaire in Volume 1, issues no. 5 through no. 7. About 500 people responded - and the overwhelming response to our question "What do you want to know?" was "MACHINE LANGUAGE PROGRAMMING!"

There are several good books on the market which purport to teach 6502 machine language programming, so we obtained a few of them and set about the business of examining them - with an eye to possibly reviewing them and/or recommending them. Our major complaint with every 6502 machine language programming book we saw was that the author moved too quickly over the most fundamental material. The less experienced reader was left behind before completing even the first chapter!

In view of this turn of events, we decided to offer a sort of "micro" course to teach these fundamental concepts, because we believe that if you can grasp the basic concepts, the rest of the material, if not easy, is at least well covered elsewhere.

I have got to admit that this insistence on learning machine level programming on the part of PET owners amuses me a bit. My first job in the microcomputer industry was as the Product Manager for the MOS Technology (now Commodore) KIM-1 micro-computer. The KIM has a 20 key calculator-style keyboard, a six-digit LED display, and 1K of RAM. If you wanted to program it, you programmed in machine language. Period. Day after day, I received telephone calls asking where the users could get a BASIC interpreter - so they could escape from the tedium of machine language. Now I am confronted by a group of users who have one of the most powerful BASIC interpreters available - and what do they want? They want to program in machine language!

The whole machine programming business begins with the machine (processor) environment. The heart of the PET is the 6502 microprocessor, a single integrated circuit containing literally thousands of logic circuits, connected through to the outside world by 40 leads or pins. Although tremendously powerful, the microprocessor is useless without a variety of other circuits. The 6502 is the control element for the system, but it contains no memory to speak of, and has no way of communicating directly with the keyboard, the screen, or any other device (such as printers, disks, cassette recorders, etc.).

The forty leads on the 6502 can be broken into three groups: the address bus, the data bus, and the control lines. (A bus, in computer terminology, means a group of wires, or "lines" which carry signals that are related to each other.) For our purposes, we can ignore the control signals and concentrate on the eight data lines and the sixteen address lines.

Every signal line in the computer is limited to two states of being. Each line, at any point in time, will carry either a voltage near zero volts or a voltage near five volts. The levels may change back and forth between these two signal levels as many as a million times a second, but these two voltages are the only ones the computer recognizes as "legal". By an arbitrary convention, the signal near zero volts is referred to as "logic 0" and the signal near five volts is called "logic 1". Thus all the internal computation done in a computer is done with only two digits - zero and one.

"But wait a minute!" you reply. "I have the digits 2, 3, 4, and so forth on my keyboard - and the letters of the alphabet; and all sorts of other characters!" And so you have. But all the PET can act upon are the groups of 0's and 1's.

Because the PET is restricted to using just two numbers, it can't use our human "base 10" (decimal) numbering system, which uses the digits 0 through 9. Instead, the PET uses a "base 2" (binary) numbering system, in which any number and letter of the alphabet can be represented by groups of zeros and ones. To count to the decimal number ten in binary, for example, the binary groups are 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, and 1010. The largest number which will fit in eight binary digits (bits) is 11111111 - which is 256 in decimal. All the characters on the keyboard can be represented by rearranging the pattern of 1's and 0's in eight binary digits - and can thus be transmitted across the data lines to and from the 6502 processor.

The largest binary number we can store in 16 binary digits is 1111111111111111, which is over 65,000 in decimal. Thus, your PET can handle up to 64K addresses across the 16 address lines.

In the decimal system, "1K" means "one thousand", so when you see "64K", you rather automatically think "sixty-four thousand". But we have just said that the 16 address lines can carry over sixty-five thousand addresses! How can this be?

In binary, "1K" is (again arbitrarily) defined as being "2↑10", or 1024. 64*1024 is 65,535. When you're talking about an 8K PET, then, you're talking about a PET with 8192 (since 8*1024=8192) different addresses into which you can store data. In actual practice, however, you only get 7167 addresses, because PET uses 1K of memory for its own housekeeping chores.

---

*Elsewhere in each issue during the course of our study of machine language, you will be able to find answers to questions raised by the previous month's "lesson". If you are left confused by some part of our discussion - write and ask questions! Our intention is to help you learn - not to confuse you. If we don't know (from your responses) when we move too fast, we can't achieve our objective! So help us out - let us know how you receive the information presented, and ask any questions you have.

Continued on Page 4

EDITORIAL

by Terry L. Laudereau

This particular issue has been one of our most challenging. First, we were late with last month's issue (not in getting it out to you, but in getting to press. . . .you should have seen the flurry of activity!), and then we changed typesetters (notice the different typeface?), and then we're in the process of getting moved to Columbia (What a tangle of details to get around to taking care of), and many of you sent your letters and articles to Columbia instead of to Audubon, so we don't even have access to your comments!

This business of editing a user-newletter gets more interesting every month. The latest suggestion we've received is for a form of classified advertising. The suggestion read "three lines of type (say 20 characters or so a line) for $10.00 seems fair. Then those of us who aren't in the business of selling PET products can at least sell off a second copy of this or an extra part, or something." Sounds like a good deal to me. Are you interested? We'll have to have copy by the 1st of each month to give to the typesetter. Please PRINT your ad - so we don't confuse "9" and "g" (which happens sometimes when you write letters). To get it started, however, I'll make you a deal you can't refuse: Send in your ad - with a check. If you are one of the first ten people to advertise in the new "classified" section, I'll return your check immediately. If you doubt it, you can tell by the number of ads that appear in the next issue. . .and they'll be listed in the order received (at least until we have enough ads to truly "classify" them).

Notice, in this issue, the column "Observations", by Roy Busdiecker. Roy, who is owner and president of Micro Software Systems (which you've probably guessed, since he plugs it in all his articles), is also vice-president and director of International Technical Systems. And he's a consultant to Edumatics Corp. He holds a Master's degree (Stanford) in Electrical Engineering. He has taught computer systems courses at the college

Continued on back cover

SUBSCRIPTION, ADVERTISING, & MISCELLANEOUS INFORMATION

The PAPER is published ten times per year by ARESCO, 6303 Golden Hook, Columbia, MD 21044. Mailed to subscribers on the last day of each month except June and December. Single copy price is $2.00, subscription price is $15.00. Foreign $25.00. One subscription includes all ten issues of one volume. A subscription does not carry over from one volume to another. Subscribers wishing to order less than the full ten issues should remit $2.00 per issue desired. Renewals for subscriptions are accepted from November 1, and the first issue of each volume is published in February.

Second class postage paid in Columbia, MD 21044 (USPS 450-930). Postmaster: Send all address changes to ARESCO, P.O. Box 1142, Columbia, MD 21044.

Editor: Terry L. Laudereau
Contributing Editor: Rick S. Simpson
Corresponding Editors: Roy Busdiecker
Dr. Hinckley W. Deardon

We do not knowingly reprint any material protected by copyright. Material submitted to THE PAPER for publication is considered free of copyright restrictions and becomes the property of THE PAPER. Any copyrighted material submitted must include permission to reprint the material or it will not be considered for publication.

Payment for subscriptions, software, hardware, and advertising orders must accompany the order. We have no facilities for C.O.D. or billing.

Advertising rates are:

¼ page.	$25.00
½ page.	$45.00
full page.	$85.00

Advertisers should submit camera ready copy (not photocopies) in the exact page size of the ad. Payment must accompany the copy.

No more than 25% of the total pages of any one issue will carry advertising. Copy should reach us by the 5th day of the month prior to the month in which the ad is to be published. Ad placement is on a first-come, first-served basis.

Manufacturers of hardware, software, and paperware for the PET personal computer are encouraged to send us a sample of their products for review. The product will be returned after 30 days, but before 120 days, when accompanied by a letter requesting return.

Subscribers wishing to submit software for consideration in the Flea Market Exchange or in the Software Shelf Library should send a copy of the program and appropriate documentation to the Software Editor, c/o THE PAPER. Programs accepted will not be returned. Programs accepted for the Software Shelf will be copyrighted by ARESCO. 20% royalties are paid quarterly on all Software Shelf programs. A sample contract will be sent upon request.

PET is a registered tradmark of Commodore Business Machines, Inc. THE PAPER is not associated in any way with Commodore and is not responsible for any action or inaction on Commodore's part. Commodore is not responsible for any opinion or material published in THE PAPER.

Readers are encouraged to submit articles on interest to PET owners. (See comments regarding copyrights, above.)

Sawyer Software

Business Software for your 8K PET tm

★★★ NEW ★★★

ACCOUNTING PACK 1 —
Accounting Pack 1 is a general ledger package designed for small businesses and homeowners. It contains check journal, general ledger, income statement (current ytd, previous month ytd and current month), balance sheet (current month and previous month). There are 15 commands and 6 reports that can be generated. The system uses an unique single-entry bookkeeping system and can hold up to 50 entries per period (month, week, day) and up to 40 different accounts. Each period's data is kept on convenient cassette tapes. Utilizing the general ledger command the user can view the general ledger entries for the month from Assets to Expenses or stop in midstream and view one particular account. Or the user can type in an account name such as "Advertising" and view the entries for that month. The Accounting Pack 1 program includes a checkbook reconcilation routine which aids in finding checkbook errors. **Sawyer Software** plans updates to Accounting Pack 1 to enable the user to use a printer, floppy disk or more memory. Accounting Pack 1 is well documented, with a newly updated User's Manual. Several businessmen are using Accounting Pack 1 and have written to us their satisfaction with the program along with their purchase of other software.
..... **$25.00**

SCHEDULE PLANNER —
Schedule Planner can be used by secretaries, receptionists, housewives or anyone wanting to plan and have at their fingertips their own schedule. Data entered is data: time, priority and description. The commands allow the schedule to be shown for a particular day, request of time or the "viewing" of appointments according to importance.
..... **$15.00**

SCHEDULE PLANNER #2 —
Schedule Planner #2 includes all the features of Schedule Planner, but is used for one or more individuals. Utilizing Schedule Planner #2 a customer can call in asking when his appointment with Dr. Jones is and in seconds the receptionist can give the date and time. Or Dr. Jones can find out his schedule for the day. With the viewing command, an appointment at 12:00 on a particular day will display on the screen at that time allowing receptionists and secretaries to validate appointments.
..... **$20.00**

COMING SOON —
Business Software for your 16K TRS-80. Write or call for details and availability.

Dealer inquires invited.

ACCOUNTING PACK II —
Accounting Pack II is a much more powerful version of Accounting Pack I. It has all the features of Accounting Pack I, plus up to 250 entries per period, MENU, Optional debit-credit of entries, easy addition or deletion of accounts, formatter for reports, intelligent report generator and single entry for sales transactions. Accounting Pack II requires at least 8K additional memory for your PET. OPTIONS: 1. Accounting Pack I to Accounting Pack II data file converter (free to previous owners of Accounting Pack I).
..... **$45.00**

BUSINESS ANALYSIS —
Business Analysis allows management to have available to them information for financial planning decisions. Up to 4 years of balance sheet and income statement information can be entered with resultant analysis in the areas of liquidity, leverage, profitability and activity. The ratios generated for each year are the: current ratio, acid test, debt-nw, profit-nw, profit margin, sales-rec, sales-inv, sales-wc, with a brief explaination of each. The growth analysis gives the yearly growth in 5 areas and the average growth in 5 areas. Future growth analysis projects figures for the next year.
..... **$30.00**

FINANCIAL PACK I —
Financial Pack I includes Amortization Schedule, Calendar (which figures days between two dates), Rebate (which is calculated on the Rule of 78's), Compound, Annuities and Monthly payment.
..... **$15.00**

CHESSBOARD —
Chessboard allows two players to play chess on your own PET graphic chessboard. This program displays each player's time and records each move to allow you to playback the game at another time.
..... **$15.00**

PAYROLL—
Especially designed with the small businessman in mind. Utilizing cassettes can record data for any number of employees (8 employees per cassette). Computes tax information and updates totals for quarterly and yearly reports. Employees can be salaried or hourly and pay periods can be either weekly, bi-weekly, semi-monthly or monthly.
..... **$30.00**

BUSINESS GRAPHIC PACK 1 —
Business Graphic Pack 1 is a simple program to use, but professional in output. The graph includes title, labeling of axis, dual graphic ability, whether the data is in Mill's, 100's or 1000's, and an optional x-axis = date and labeling of the x-axis with month and year. Entry is as easy as typing the title, # of entries, the X,Y value (Jan. 15, 1978 would be entered as 115.78), entering if the x-axis = date, if the user wants crosshatching and then graphing. The program also includes Nth order and Geometric regression to give the user a formula for his set of data (if possible).
..... **$25.00**

CHECKBOOK RECONCILIATION —
Designed for ease of use and to find those troublesome checkbook errors, this program balances your checkbook and bank statement from month to month. It locates over ten different types of errors and instructs the user on correcting them, including bank statement errors made by your bank. Records data on cassette for next month's balancing - saving re-entry of figures.
..... **$25.00**

CALCULATOR —
Especially for the accountant-bookkeeper. This program not only allows your PET to function as a calculator, but also has such features as Lister (which verifies and displays errors between two tape listings) and Matrix (which adds columns vertically and horizontally).
..... **$10.00**

All Programs include documentation, are in BASIC and on cassette.

*PET is a trademark of Commodore Business Machines, Inc.

Sawyer Software

201 Worley Road
Dexter, Mo. 63841
314-624-7611

We honor master charge THE INTERBANK CARD

VISA

PROCESSOR OPERATION

Although the things a microprocessor can do are very complex, its basic operation is very simple. The fundamental rule of microprocessor operation is that only one thing can happen at a time. At any given instant, the processor is performing only one function. For instance, it cannot add two numbers together at the same time it's storing a third number.

In order to make sure that all parts of the processor system operate together smoothly (are synchronized), the microprocessor contains a clock circuit which sends out an endless series of pulses; it alternates between the logic 0 and logic 1 states at a fixed rate (frequency). To make that frequency as precise as possible, the clock line is regulated by a quartz crystal (which we won't digress to explain at this point).

Figure 1 is a block diagram of the PET microprocessor, although a similar diagram would apply to almost any other microsystem on the market. Although the clock circuitry is part of the processor, the output of the clock is fed to every other component (element) of the system as well.

The figure shows the 16 lines of the address bus and the eight lines of the data bus, all connected from the processor to each of the other elements of the system controlled by the processor.

The address bus is always controlled by the processor; that is, the processor itself decides what address values will be put on the address bus. On the other hand, the data on the data bus may be generated (provided) by either the processor or by the other elements in the system. The processor selects the element which will place the data on the bus. This "selection" is exercised through a control line called the "READ/WRITE" line.

If the processor is instructed to write data into memory, it puts the address of the memory location to be changed on the address bus. Then it puts the data to be written into that location on the data bus, and indicates that the data is to be written through the use of the READ/WRITE line. If the processor is told to read the value of data stored in a memory location, it puts the location's address on the address bus, then activates the READ/WRITE line so that the addressed location puts its data onto the data bus. The processor then "reads" the data off the bus.

In other words, the address bus determines which location in the whole processor system is to be activated, and the data bus acts as the transmission path for the data, either from the processor to the system or from the system to the processor.

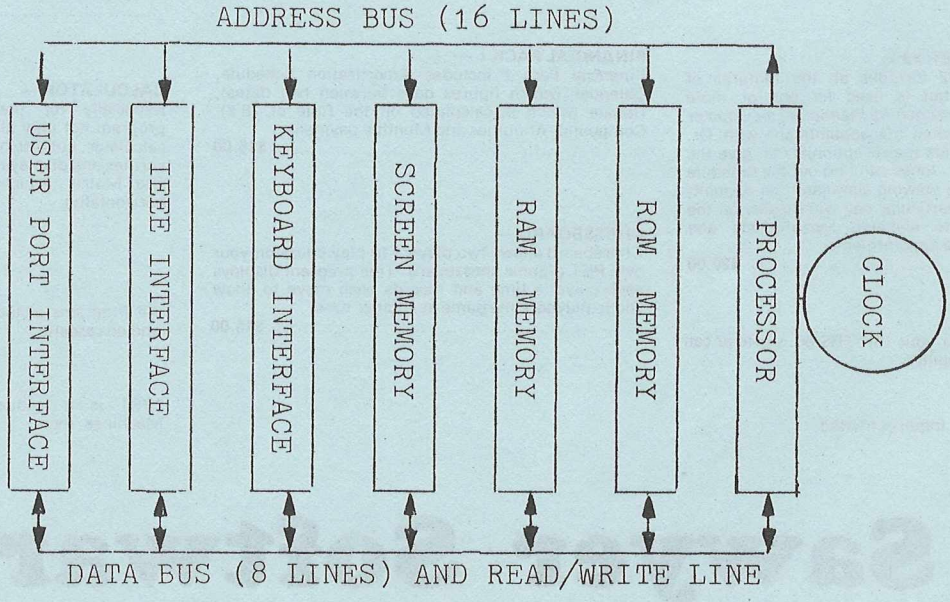


Figure 1

STARTING THE SYSTEM

It is important to understand that the microprocessor itself has no program stored within it, although it does have space for data storage (more about this later). All the instructions for the processor must be stored in the computer system's memory. When your PET is first turned on, a uniques series of actions take place, called the RESET sequence. When the microprocessor is reset, a specific address is placed on the address bus and the processor activates the READ/WRITE line. The data in that address is placed on the data bus so the processor can "read" it, and is then transferred to a special counter within the processor called the program counter.

The function of the program counter is to hold the address of the next instruction to be "read" by the processor. Having determined where the instruction resides, the processor puts the address of this instruction on the address bus, increments the program counter to "point to" the next address, activates the READ/WRITE line, and "reads" the data residing in the address which was placed on the address bus.

This whole process of reading a memory location is called a "fetch". The processor examines the pattern of ones and zeros transmitted to it on the data bus. Each microprocessor instruction is formed by a unique combination of ones and zeros. Once the processor has understood ("decoded") that particular pattern (called an "opcode"), it executes the instruction. The processor then "goes back", transfers the updated address in the program counter to the address bus, increments the program counter, activates the READ/WRITE line, and so on, for every instruction in your program.

Some of the 6502 instructions require only a single data word (byte) for the entire instruction. Others may require that additional information be fetched from memory before the instruction can be executed. For instance, there is an instruction which tells the processor to load its Accumulator (which is one of the storage areas inside the microprocessor itself) with data from another memory location. This requires the processor to fetch the next two bytes of data from the program. These next two bytes must contain the address of the data to be loaded into the accumulator. In other words, the processor must first fetch the instruction, and then it must fetch the address in which the data resides, and then it must fetch the data itself.

Notice that in this example, it is necessary to load two bytes from memory. This is because a memory address is composed of 16 bits of data, but each memory location can store only eight bits of information.

Each time another byte is fetched from the program, the program counter is updated (incremented) so that it "automatically" points to the next instruction. Once all the data required for an instruction is fetched from memory, the processor then executes the instruction. In the case of our example, the address of the instruction byte is placed on the address bus, having been taken from the program counter. The program counter is updated to point to the next address, and the processor "knows" it has to have data in order to execute the "Load The Accumulator" instruction. Back it goes to the program counter, places the address (this time, the address will contain data) on the address bus, updates the program counter again, and fetches the data from the address it has placed on the address bus. Only then - after all the data it needs is available - does the processor execute the instruction and actually load the Accumulator.

The processor continually cycles between these fetch and execute operations. It will continue to draw its instructions from successively higher memory locations until a "jump" or "branch" instruction is encountered. Such an instruction tells the processor to draw its next instruction not from the location pointed to (stored in) the program counter, but rather to calculate a new value (or fetch it from somewhere in memory) to be placed in the program counter, and draw its next instruction from the new address.

There are several important consequences of this method of processor operation. The first point is that the processor can only access devices as though they were memory locations. In other words, the processor can only put an address on the address bus and read or write the contents of that address via the data bus. There are no special instructions which provide for input or output by the processor except by manipulating these busses.

The second important point is that the processor will continue to fetch and execute instructions as long as power is applied to the system. Even when the computer appears to be "doing nothing", it is constantly executing instructions (usually in a loop) waiting for some external event to occur which will change the data it is reading.

MEMORY ORGANIZATION

Referring again to Figure 1, you can see that memory is divided into different areas for different purposes. We call the total range of possible memory addresses (i.e., the decimal numbers from 0 to 65,535) the computer's "memory space". A diagram showing which devices are placed, or addressed from, the various parts of the memory space is called a "memory map". In normal operation of the computer, there will never be two devices in the system which have the same memory address. Each address references a unique location within the system, although many addresses may be assigned to any device (i.e., it doesn't matter which device uses those addresses). Figure 1 shows all the devices which exist within the PET's memory map.

"ROM" MEMORY

Part of the address space in the PET is dedicated to devices which have data permanently built into them when they are manufactured. This is called "ROM", or Read Only Memory space. This means that regardless of what the processor does with the READ/WRITE line when these locations are addressed, it can only read data from them. It can never write data in them or in any manner alter the contents of these addresses. Even if power to the PET is turned off and back on, the data stored in these locations will never change. Almost all microsystems must have some ROM, because when the

system is first turned on, there must be instructions permanently stored somewhere which tell the processor what to do at initial power-up (RESET).

In some microprocessor systems, these instructions simply tell the processor how to load more instructions from some external device (such as a tape cassette or disk system). The PET is more sophisticated than this, and its ROM contains all the instructions necessary for normal operation of the PET, such as reading tapes, entering BASIC programs, running those programs, writing programs on tape, and so on. So, on the PET, at least, it isn't necessary to load other programs in order to have full operating capability of your computer. About twelve thousand bytes of the PET's memory space is devoted to this ROM.

Having large amounts of ROM on a system is advantageous because it means that the system can perform complex operations without having to be fed any further instructions. The disadvantage of large ROM programs is that they cannot be altered. Since the data is permanently embedded in the ROMs, any mistakes made in the ROM program can never be changed. To correct the mistakes, new ROMs, with a different program, have to be manufactured.

"RAM" MEMORY

RAM is an acronym for Random Access Memory. This is a misnomer, since ROM can also be accessed randomly. A better term for RAM would be Read/Write Memory (but RWM is hard to pronounce). The distinguishing characteristic of RAM is that data in RAM memory locations can be written (modified) by the processor, whereas ROM memory cannot be altered at all. Both RAM and ROM memory can be accessed randomly, in that it isn't necessary to read data from memory in any particular sequence. The RAM memory in most giant computers and many minicomputers is made of ferrite beads (cores) shaped like microscopic doughnuts. The magnetization characteristics of the beads can be changed by sending a small current through a coil surrounding each one. This characteristic magnetization will remain with each core, even if power is removed from the entire system. Thus, when power is restored, the data contents of the memory would be unaltered.

Without exception, microcomputers use semiconductor elements instead of ferrite cores for their data storage. When power is removed from the system, the contents of RAM memory are completely lost. When power is reapplied, there is no way to predict the contents of RAM - which is why some ROM is required. If the system contained only RAM, then when power is first applied, the processor would attempt to fetch instructions from the RAM, in which there is nothing (at power up) but random bit patterns. The processor would run amok - out of control.

In most applications for the PET, the entire program and all the data used by the program are stored simultaneously in RAM. Therefore, it stands to reason that the more RAM there is, the larger program it can run. When the PET was first introduced, 8K of RAM was considered sufficient for most programs. Because of the sizeable decrease in the cost of RAM components, however, more RAM is affordable and easily attached to the PET via the Memory Expansion Connector. Many of the other microsystems being offered today have provisions for easy expansion to up to 48K of RAM.

So far, we have concentrated on the internal operation of the processor and its associated memory. Next month, we'll discuss the circuitry which makes microprocessors useful to people; the input and output from the real world.

PET PROSE

There are many people who want specialized application programs to use on their PET's, but who are not knowledgeable enough to design or to write them.

If you can write significant software in some specialized field, and are willing to do so, we may be able to help you find the people who are willing to pay you to write programs for them. Send $25. with your name, address, and field of expertise. The information will be published in all the remaining issues of this volume.

Michael J. McCann (716) 837-8068
28 Ravenswood Terrace
Cheektowaga, New York 14225
*Scientific and Systems Programming

OBSERVATIONS
By Roy Busdiecker

Since my comments are late, I would suspect that by now you've received a resounding vote of approval on Volume 2. The Paper (could we try calling it The PET Paper again. . . since others seen to be using the PET name with impunity?) has achieved a new level in quality, with respect to both appearance and content. Congratulations, and keep up the good work.

Also noticed the ad for The Paper in the March issue of Creative Computing - a step I applaud. There are many PET owners (and dealers, too!) who don't know what's available. Perhaps your readers could help spread the word.

Gary Maier's article on machine-language programming was a good initial effort in that area. Hope you won't go too long without including two items: 1) a picture of the 6502 logical elements and data/control transfer paths (like the one on the summary card), and 2) a summary of opcode mnemonics, opcodes, and meanings (both alphabetically by opcode mnemonics and numerically by opcode. . .with opcodes in both hex and decimal, for those who POKE).

Since Commodore is now reviewing The Paper before you publish, perhaps they will accept advice from you regarding copyright. . .the first issue of their PET User Club paper contained a short utility routine which I contributed to the PET User Notes (Gene Beals), without permission or credit. That's not responsible!

I wince everytime a "complete" list of software suppliers (like one of the big "slick" magazines had recently) shows up in print without including Micro Software Systems (P.O. Box 1442, Woodbridge, VA 22193). Since MSS supplies software only for the PET (at this time, anyway), it really hurts to see a statement like Dennis Costarakis's that "only three 'software sales' firms. . .have provided prompt delivery. . .". We ship via First Class Mail (except quantity orders to dealers), normally the day after an order is received. Hope you've heard from some of our customers. We invite comparison, both in service and in product quality!

Robert Buxton's article ("Fast Forward to Find Your Program") was interesting, but it's much easier just to use C-10 cassettes at one program per cassette.

After more than a year, my built-in cassette failed in the manner described by Tyson Richmond. Problem was an open capacitor in the RECORD circuitry. Also bought one of the new external cassette units (with the name "Commodore" molded in the plastic). It performs excellently. . .improved electronics give a more stable signal (watch it on a good scope!).

John Hirsch's problem with the "endless" cassette may have been that he never closed his file. When the file header is encountered the second time (without a trailer having been found), it should make the PET hiccup!

NEW PRODUCT ANNOUNCEMENTS

The 3G Company, Rt 3, Box 28A, Gaston, OR, 97119 announces a self-contained light pen, which plugs directly into the PET's User Port. The light pen makes it possible to bypass the PET's keyboard and interact directly with the information displayed on the screen. The company says that the light pen will aid CAI oriented programs immeasurably, since a child doesn't need to know how to type to use the pen. A "menu" can be displayed on the screen, in graphics, and the student can make a selection from that "menu" with the light pen. The pen is completely assembled and ready plug into the PET, with a sample program and programming instructions. The price is $29.95 (plus $1.50 for postage and handling) within the US, and $6.00 more for foreign orders. For more information, contact Imants Golts at 3G.

Concordia Designs, P.O. Box 219, Station D, Scarborough, Ontario, M1E 5B7, Canada, announced the successful interfacing of a PET with a Houston Instruments Digital Plotter. Full hardcopy graphs of data or functions can be automatically plotted with pen accuracy of up to 0.005 inches. Software is also available to issue plotting commands in simple BASIC (interactive) steps. For more information (pricing, delivery, etc.,) contact Andrew G. Hwang, Interface Division, at the above address.

CONDUIT announces the availability of eleven CAI packages for the PET. The packages are primarily simulations that cover basic concepts in biology, chemistry, and physics. The units are supplementary to regular classroom instruction, and are intended to broaden the user's experience in areas not usually covered in basic laboratory classes because of time and expense. Printed guides for students and instructors complement the computer programs in each package. Authors of computer based materials in biology, chemistry, economics, education, geography, humanities, management science, mathematics, physics, political science, psychology, sociology, and statistics are encouraged to submit their materials to CONDUIT for review. For more information about CONDUIT, and for information on how to order the available instructional material, write for a free copy of the CONDUIT magazine, Pipeline. CONDUIT, THE PAPER Offer, P.O. Box 388, Iowa City, IA 52240. Be sure to mention "THE PAPER Offer" to get your free copy of Pipeline.

California MicroComputer Company, P.O. Box 3199, Chico, CA 95927, offers "complete business application software for general ledger, accounts receivable, accounts payable, and payroll". The company literature says the software is designed with the inexperienced computer user in mind. Be sure to write for price and delivery information before ordering.

NEW PRODUCTS

AT LAST – AN $800 DISK FOR THE PET

CGRS Microtech announces the PETDISK System - a high-speed floppy disk and an S100 expansion chassis designed to give the PET owner an as-yet unheard of potential for expansion. The floppy disk provides the PET with the ability to instantly load and save program and data files; the optional S100 expansion chassis permits the PET owner to take advantage of the great variety of S100 products already on the market, such as additional memory, telephone interface cards, analog I/O, printer I/O, voice generators - and even voice input.

According to the CGRS literature, the PETDISK System offers an economical solution to the problem of expanding the PET. It offers complete expansion compatibility with other systems (such as the TRS-80, for example).

CGRS provides the PETDISK in three different packages, so each user has the opportunity to select the package that fits both his need and his budget.

Package 1 contains the EXS100 disk controller, mini-floppy disk drive, cabinet, and disk power supply - for $799.95. It's a ready-to-go disk system, holding 80Kbytes of data. It can be expanded by adding two more disk drives, for a total of 240 Kbytes. The EXS-100 board can be fully assembled into a combination board, and the optional S100 chassis can also be added.

Package 2 contains the ESX100 combination disk/S100 board, disk drive, cabinet, power supply, and a 3-slot S100 chassis, for $895. The two extra S100 slots can be used for memory, printer, or any of the already available S100 boards. Two additional disk drives can be added to the disk controller.

Package 3 has the ESX100 combination disk/S100 board, disk drive, full S100 mainframe and power supply (+8v @10A, ±16v @ 1A), and a 10 slot S100 motherboard for $1195. Nine unused slots give the PET owner tremendous expansion potential, and two additional disk drives can be added to this package, too.

The EXS100 board is available separately, of course as an S100 adapter only ($199.95), as a disk controller only ($299.95) or as a combination disk/S100 adapter ($349.95). CGRS also provides, for $75.00, the KM3 software package, which permits the user to LOAD, SAVE, RUN, and UPDATE files on the disk. Revision 2 of this software package will allow the user to OPEN, CLOSE, INPUT, and PRINT a disk file, and the commands will execute directly as well as from a program. KM3 is an excellent disk file support package from WILSERV.

CGRS offers a variety of other microsystem products, and you can get information by writing to them at P.O. Box 368 Southampton PA 18966. Or call Joe Swope at (215) 757-0284.

CORRECTION: CLUSTER/ONE

The price indicated in the February issue for the Nestar Systems Cluster/One (which connects up to 15 PETs together to a single central disk system) was incorrect. Prices begin at $4,500. Further information is available from Nestar Systems, 430 Sherman Avenue, Palo Alto CA 94306

PETPILOT

Commodore PET owners can get full standard PILOT (the most popular computer-assisted instruction language) on a minimum size PET. Dave Gomberg, of the PETPILOT Project, announced not only a PETPILOT Language processor, but also an Editor suitable for preparing long programs of up to about 80,000 characters.

The new product features full BASIC in compute statements as well as two new keywords designed to make PILOT programming easier and faster. All language features of the most recent PILOT standard are implemented. In addition, the system has been designed to be easy to learn and use.

Because PET tapes can only move forward, these are limits, depending on memory size, to the distance a PETPILOT program may jump upward, but this does not create a barrier to the effective use of PETPILOT.

Only the tape drive supplied with the PET is required to run any PETPILOT program. While simple PET-PILOT programs can be created on a one tape PET, authors writing long programs will need the second cassette drive offered by Commodore International of Palo Alto, CA, manufacturer of the PET.

The package offered by the PETPILOT Project contains both programs, a sample PETPILOT program, a teacher's manual, a quick reference card, and licenses to run the programs on a single PET. A tutorial course of two one hour lessons in effective use of PETPILOT is also available. The basic package costs $12, the tutorial is an extra $8. Both products can be ordered by specifing the PET serial number to be licensed to Dave Gomberg, 7 Gateview Court, San Francisco CA 94116.

ATTENTION! IMPORTANT NOTICE

When you see something in THE PAPER that prompts you to write to a manufacturer or vendor (whether it's a new product announcement or an advertisement), be sure to mention that you discovered the product (or, at least, saw it mentioned) in THE PAPER. More advertising means more income, which means more quality. You who have been with us from volume 1 issue 1 can see how much progress we've made - and it's all directly a result of your support! Do let vendors and manufacturers know that it pays to advertise in THE PAPER!

DR. DALEY'S SOFTWARE FOR THE PET

DR. DALEY's software continues to expand offerings. Listed below are our most popular programs. No PET owner should be without these. Dealers, you should stock them as well.

PET TREK 3 — Like STARTREK, but has several UNIQUE features. For example, the unpredictable EXPERIMENTAL RAY, who knows what it will do ..... **$ 7.95**

BACKGAMMON — It's you vs the PET with an exciting game of BACKGAMMON ..... **$ 7.95**

MASTER MIND — Plays two simultaneous games, one where you guess PET's secret code, and another where PET guesses yours **$ 7.95**

RENUMBER — Will renumber your BASIC programs, including all jump statements. For a 6K source code requires less than 5 seconds **$12.95**

PILOT — A BASIC coded PILOT interpreter. A second high level language for the PET. Simple to use, even a ten year old can learn to use PILOT quickly. With sample PILOT programs and documentation **$12.95**

CHECKBOOK — Will balance your checkbook and save totals in 16 categories on tape. Will produce end of month and year to date summaries. Categories can easily be changed to suit your own purposes..... **$12.95**

MAIL LIST — Keeps a mailing list and will sort the list into sub groups using up to three search parameters **$12.95**

All of our programs are available on tape or for the Compu-Think disk. We charge $5.00 for the disk and shipping, but you can subtract $1.00 for each program which we place on the disk. Order 5 programs and you get the disk free!

MAIL LIST — The above program has been modified for disk files. Will be placed on a disk by itself which you can then use for your mailing list **$19.95**

FLASH! — We have just acquired the rights to distribute a linking loader for BASIC programs! This will allow you to link exclusively numbered BASIC subroutines in memory. No serious programmer should be without this useful programming tool..... **$12.95**

An ideal companion to the linking loader will be our library of useful subroutines which can be linked into your own program. Currently over 25 useful routines are included. These range from plotting utilities to a beautiful display of rolling dice. Write or call for a list or order the set for only **$49.95**

* * *

Remember that we GUARANTEE that your order will be shipped within four business days from receipt or you will receive a coupon for a discount on a future purchase.

Charge your order to MC/VISA

master charge THE INTERBANK CARD

VISA

DR. DALEY, 425 Grove Avenue, Berrien Springs, Michigan 49103

Phone (616) 471-5514 Sun. to Thurs. noon to 9 p.m. eastern time

Dr. Hinkley W. Deardon is a world famous atomic physicist, a brilliant, all-around genius. He has been forced to relinquish his unsurpassed professional career because of a chronic ulcer condition, and has retired to a remote mountain cabin which he affectionately refers to as "the pits". Because of his ulcer, Dr. Deardon has been forced to sacrifice all the pleasures of wine, women, and song (he subsists entirely on cream of chicken soup heated over Sterno cans), and his physician has advised him that writing vitriolic articles is the best form of therapy for his ulcerous condition.

Considering that it would be a tremendous loss to the world if Dr. Deardon's ulcer should kill him off before his envious colleagues have had a chance to decipher the coded notebook he mischievously left taped under the center drawer of his laboratory desk, THE PAPER has agreed to publish Dr. Deardon's diatribes - purely in the interest of restoring Dr. Deardon's health in the hopes that he will eventually be ferretted out and forced to reveal the secret of. . . .well, his research.

Readers should be aware of two things, however. First, we do not edit Dr. Deardon's articles. He, like most geniuses, is convinced that he has done all the editing needed, and he would be irretrievably insulted should we alter so much as a single word. This might cause him to retreat even further into the wildernesses, and be even more difficult to locate.

Second, we are not permitted (by orders of Dr. Deardon's physician) to agree or disagree with Dr. Deardon's opinions. Only the readers of THE PAPER are entitled to comment. If we, the editors, disagree with Dr. Deardon, he will fly into an uncontrollable rage. If you, the reader, disagree with Dr. Deardon, he will patiently explain his views again - and again - and again - until you see the light. If we, the editors, agree with Dr. Deardon, he will fly into an uncontrollable rage - for he will not be able to feel that is a "voice in the wilderness" revealing unthought of, unheard of truths to the world at large. If you, the reader, agree with Dr. Deardon, however, he will be appeased, feeling that at last he has managed to reach a person with true foresight, vision, insight, and wisdom.

In view of all these factors, we urge you to write to us, giving us your comments about Dr. Deardon's articles. He sends his trained pigeon in once a month for mail, and we can forward your letters to him then. We call upon all of you to help us in our efforts to locate Dr. Deardon - to make him feel less vitriolic, less bilious, less obnoxious, and less sarcastic. Help us by responding to his articles!

FROM THE PITS
By Dr. Hinkley W. Deardon
A TWO-BIT SERIAL INTERFACE FOR THE SELECTRIC

In a recent issue of THE PAPER, a reader wanted to know why PET-to-Selectric interfaces are not common. My question is: "Why would anyone ever want to interface a Selectric typewriter to a microcomputer?"

I remember when the first Selectric typewriters were introduced to the anxiously waiting world. They came complete with service manuals. This turned out to be very fortunate, for it wasn't long until IBM-style servicemen, wearing three-piece suits and carrying little attache cases full of intricate little tools, were seen scurrying around in all parts of the country on almost daily repair missions.

Someone once described the Selectric as "a triumph of development over bad design". This may well be true, since later models of the Selectric were somewhat more reliable. But Selectrics were never sold or repaired by ordinary typewriter stores. Those folks unfortunate enough to own a Selectric without owning a Service Contract were forced to resort to the nearest IBM temple, where a high priest would pick at its entrails and mutter healing incantations for a slight forty three bucks or so (which always seemed to be the minimum, no matter how the receipt was itemized).

Selectric mechanisms have been used in many computer terminals, even some which were not manufactured by IBM. Some of these terminals are now available reconditioned, at attractive prices. ("Reconditioned" means that someone has replaced the platen and gone over the innards with a damp rag.)

These surplus Selectrics usually have some features in common. Their transmission codes are EBCDIC (Extended Binary Coded Decimal Interchange Code, if I remember rightly), or Correspondence Code, or even PTTC (Paper Tape Transmission Code). But not in ASCII. Which the rest of the computer world uses. And there is a different set of typeballs for each code - to ensure complete incompatability.

Another feature of these Surplus Selectrics is that they are usually not serviced by IBM. That means the high priest at the IBM temple will reject you. And your unusable Selectric. And they'll smile as they turn you away, too.

Now, I will admit that Selectrics have some pretty typefaces, and can turn out nice hard-copy if you're patient. And members of the U.S. Robotics Society will enjoy building a Selectric interface with typefaces selectable under software control, since the only way to do it, so far as I can tell, is to build a little guy with tiny hands who changes typeballs very fast.

I read in the newspaper not long ago that a gentleman in New Jersey was found at the bottom of a lake with a Selectric tied to his ankles for ballast. It was assumed by authorities that the unknown parites who deposited the gentleman in the lake had temporarily run out of concrete. But I'm not so sure.

Perhaps the victim was a Selectric repairman, and the guilty party was an enraged customer. Depositing his victim at the bottom of a lake, tied with 25 cents worth of rope to an equally dead Selectric, would seem to have been an appropriately symbolic revenge.

Like I said: A two-bit serial interface.

PET USERS

Glenn Schwartz
807 Avon
Philadelphia, PA 19116

John Loofbourrow
ACGNJ
(201) 233-7068

United PET Users
1929 Northport Dr. No. 6
Madison, WI 53704

Twin Cities PET Users
(John Fung)
(612) 376-5465

Peter Oakes
2235 Lakeshore Drive
Muskegon, MI 49441

Sacramento PET Workshop
P.O. Box 28314
Sacramento, CA

Midpeninsula PUG
Ford Aerospace Cafeteria
3939 Fabian Way
Palo Alto, CA
(415)328-7745 (Harry Saal)

BAMUG
1450 53rd Street
Emeryville, CA

David Smith - NOCCC
3030 Topaz No. A
Fullerton, CA 92631

St. Louis PET Users
(Mary Perkinson)
(314) 432-5225

Northern VA PET Users
2054 Eakins Ct.
Reston, VA 22091

Shelly Wernikoff
2731 N. Milwaukee Ave.
Chicago, IL 60647

SPHINX
(415) 451-6364

Vancouver PET User Group
Box 35353, Station E
Vancouver, BC, Canada

Lincoln Computer Club
750 E. Yosemite
Manteca, Ca 95336

PET NET I
14.34 MHz Sundays
1500 UCT (10 AM CDT)

PET NET II
7.24 MHz Thursdays
1600 UCT (9 AM PST)

Independent PET Group
22 Firs Walk, Tewin Wood
Welsyn, Herts., UK

PET Users Group
2001 Bryan Tower Suite 3800
Dallas, TX 75201

Capital District PET Users
(Ben Green)
(518) 370-1820

PET Users of Japan
Soichiro Moridaira
Shinsen Park Himu, Rm. 150
4-13 Shensencho
Shibuyaku, Tokyo, Japan 150

David Liem
14361 Warwick Street
Detroit, MI 48223

Larry Williams
P.O. Box 652
San Antonio, TX 78293

Richard Prestien
6278 SW 14th Street
Miami, FL 33144

Have we missed anyone? Let us know who anc where you are!

SOFTWARE

DATA RETRIEVAL $10.00
Create unlimited numbers of data files of unlimited length. Each record in each file consists of up to 255 characters in as many fields as required; the first three fields are keyed for selection and sorting purposes. Up to 50 records at a time from any combination of existing data files can be selected and sorted in either ascending or descending order. Listing provided. Eight pages of documentation.

DATA EDIT $10.00
Companion program to our DATA RETRIEVAL program. Edit records, create a new file, add to existing file from the keyboard or from another file. Listing provided. Uses same documentation as DATA RETRIEVAL.

STAT I $20.00
A real statistical package for up to 100 data points. Basic stats, linear regression, moving averages, plotting, and transformation. Listing provided. Documentation.

STAT II $20.00
Seven programs in one package to carry out most of the statistical tests you might need. All the programs are interactive and conversational:
a. Descstats takes a set of data, either as single values or as grouped data, and calculates descriptive statistics. . .mean, variance, standard deviation, maximum and minimum values.
b. Probf accepts test statistics produced from statistical tests: the variance ratio, F, Student's standardized mormal deviation or chi square. . . and calculates the probability of such a value. It replaces looking up the values in tables, and is also used as a subroutine in all the programs below.
c. 1wayanova performs analysis of variance on two or more sets of data, each set being the responses of a group of cases to a treatment (for example, two drugs and a control - three treatments). The number of cases for each treatment may be the same or different. In the special case of two treatments, it is identical with the well-known T-test.
d. 2wayanova is similar to 1wayanova, but each case is subjected to two treatments A and B simultaneously. For example, treatment A might be different drugs (or different dosages of the same drug) and treatment B different diets; alternatively, the treatment B can be blocks of a block design.
e. Anova2wayr is the same as 2wayanova, but there are two or more cases (replicates) for each treatment combination. All the analyses of variance programs including tests of significance.
f. Slr calculates linear regression, correlation coefficients, and coefficients of determination and tests the significance of all statistics. Designed for grouped data, i.e., several values of Y (the dependent variable) for each value of X (the independent variable).
g. Xyslr is similar to Slr, but the X and Y values are in pairs.

SPACE FIGHT $10.00
A two player game in which each player tries to destroy the other player's spacecraft by firing bursts of energy. If a player fires - and misses - he's in danger of running in to his own energy burst and destroying himself instead! Written instructions.

CHASE $10.00
A real-time version of the popular Chase game, in which the robot guards can be as intelligent or as dumb as the user wishes. Try to escape!

MICROMAZE $7.95
A maze is drawn randomly on the screen - your job is to find your way out in the shortest possible time! No two mazes alike. Documented sound is available.

BLACKJACK $10.00
Excellent graphic game. Displays the cards as they are turned up, keeps track of your winnings (and losses), and wises off when you lose. It's the best version of Blackjack we've seen. No instructions, since all instructions are incorporated into the program itself.

BLOCKADE $10.00
A two-person game in which each player tries to avoid the other while also avoiding the wall around their enclosure. Excellent graphics. Documentation.

DEFLECTION $10.00
A fast-paced, action game in which you attempt to hit all the targets. The ball moves horizontally and vertically until it runs into a deflection shield you place on the screen using the M and N key graphics. Timed success to encourage greater skills.

HOSTAGE $7.95
A group of terrorists have seized a planeload of innocent people, holding them as hostages until their demands are met. You, as governor of the realm, are supposed to do something! But what? Your chances of re-election depend on your ability to make the correct decision in time. Non-graphic.

STARTREK 2001 $10.00
America's favorite computer game is available with sound! It uses the sound circuit described in issue no. 1, Volume 1 of THE PAPER. And, even without sound, it's a fun game. Search through the galaxies for Klingons, refuel your ship at the starbases, and try to avoid being destroyed by the enemy. Documentation.

AIR-SEA WAR $7.95
Your naval fleet is at war with enemy aircraft! Will you destroy them before they sink your entire fleet? Good graphics.

XMON $15.00
Machine language monitor and disassembler with

an exciting difference: The user is actually encouraged to make use of the program. A "menu" is displayed after each command is executed, so you don't have to remember each of the commands. Each command prompts you for the correct entries. The entire program is written in BASIC, not in machine language, so you can see how it is done. Functions: Write to memory, read memory/disassemble, save memory on tape, load from tape to memory, execute a machine language program, exit to BASIC. Four pages of documentation.

TUTOR $19.95
A beginner's guide to PET BASIC and to the cursor control features which make PET the No. 1 personal computer available. Over 400 screensful of information are presented in 20 lessons, provided on both sides of each of two C-45 cassette tapes, five lessons per side. All the major BASIC keywords, plus exercises, quizzes, and a few wisecracks. A must for all beginners.

PET BASIC COMPLEAT $29.95
This 170+ page manual is the companion for the TUTOR tapes. The manual is drilled for a standard 3-hold binder, is indexed for easy reference, and reproduces (so far as is possible without a PET printer) exactly the data presented in the tape and displayed on the screen.

PET TUTORIAL PACKAGE $39.95
Because we feel that the TUTOR tape and the PET BASIC COMPLEAT manual constitute an unbeatable learning tool for beginning PET person, we offer the set as a package deal - you get the 170+ page manual (as of this writing, there are 183 pages) plus the two C-45 cassette tapes, for under $40. Save almost $10.00 on the set.

CMC/WPP $29.50
The Connecticut Microcomputer Word Processing Program turns your PET into a powerful text processing system. You can enter text in upper and lower case letters, edit the text, and format the text for finished printing. Due to the limit of available memory, you can enter approximately 2000 characters in an 8K PET - about the length of a full page letter. Complete input and editing features: enter text, delete lines, insert lines between the lines of existing text, do substitutions within a line, or replace one line with another. You can also move a block of lines from one place to another in the text. You can alter the left margin for indentation, or adjust the right margin (although there is no provision for right-hand justification). There are provisions for filling a line with a specified character, or centering a repeated character on a line. Twenty-five pages of documentation provided.

BAZAAR $7.95
Greetings from the ancient city of Karazama! You are welcome, stranger. Borrow money from Gragr, the Innkeeper, and go into business for yourself, as a merchant in the Karazama Bazaar. Buy your goods from dealers, and haggle with your customers for the most profit! Earn enough to pay off your debts and buy an inn of your own! Of course, there are thieves to contend with. . . and the tax collector for Lord Incatazam, the ruler of Karazama. . .and someday, Gragr will want his money back - with interest. Call it an "economic simulation" game if you like, but what it really is is just plain fun! The program tape and five pages of documentation are available from THE PAPER for just $7.95. Use the order form on the inside back cover of this issue.

STAT III $7.95
A good program for calculating elementary stats. 200 data points, edit after entry. Conversational, no DATA statements, Stats provided are: mean, median, mode, high value, low value, range, variance, standard deviation, sample std. deviation, average deviation, and an automatically scaled bar chart.

WAREHOUSE $7.95
Who said running a warehouse is easy? There's never enough help, they're never where you need them when you need them there, things pile up on the loading dock, and if you make a mistake, Mr. Farrell, your boss, sends you nasty memos! You sit at your PET and dispatch workers, review orders, and manage incoming shipments. You get a real feeling of accomplishment reviewing your work at the end of an eight minute "day", but beware! One mistake too many and Farrell will fire you! Warehouse is another fine simulation from the author of Bazaar, and he provides four pages of documentation with the program tape. Order from THE PAPER, for $7.95, using the order from on the inside back cover.

THE PAPER guarantees that each of these programs will perform as claimed herein. If your copy of any program fails to load, send it back to us - we'll ship you a new copy. Record a program of your own on the back of the tape, however, so we can check the tape head alignment.

Albert Einstein wasn't talking about computers when he said "No amount of experimentation will ever prove me right - but a single experiment can prove me wrong." - was he??

Kenneth Tong writes that there is a new PET user group being formed in the Seattle, Washington area. PET owners in that region should contact Ken at the Northwest PET Users Group, 1800 Taylor Avenue North, No. 102, Seattle WA 98109.

From Larry Riley: "I can't stand it any longer! Enclosed is my check for $15.00 for volume 2. I cannot miss an issue of THE PAPER!. I'd like to know of other PET owners in the Portsmouth, NH area. I'd be interested in attending a group or club, if there is one. I have a new PET (series II), and plan on getting a good printer and a floppy disk. Any recommendations?"

(Editor's note: We don't have any other subscribers in your zip code, Larry. Write and give us the zips nearby, and we'll see what we can do. As for printers and floppies, there are several good ones on the market - since we haven't personally used them, however, I cannot in good conscience recommend one.)

We recently received a letter from Charles Nadovich of Children's Village, a school for emotionally disturbed children. He would like to use a PET in his classroom administration, and is look for some help in writing a program he has outlined. If you would like to assist him, write to him at Children's Village, Dobbs Ferry, NY 10522. At least, contact him and find out about his program, then decide whether you'd like to help! Send him a SASE, since we think he'll get a lot of responses to this request.

John D. Hirsch purchased the Skyles Electric Works 8K memory board, and offers these observations: "This 8K static memory expansion board was shipped promptly by insured mail and was well packed. Following the clear instructions enclosed with the board, it took me about two minutes to attach it above the PET's main board with a screw, spacer, and three plastic standoffs, and plug it into the expansion and power buses. Then I turned on my PET - and lo! 15359 BYTES FREE! A test tape that came with the board was run, and assured me that everything was working properly. For those of you who think $250.00 is too much to pay for 8K of memory, let me assure you, you get what you pay for. The Skyles 8K Memory Board is well worth it!"

From Steven Kahn, some observations about ECLECTIC's Cassette No. 2 - the one with Slots, Blackjack, Poker, and Craps on it: "In a fit of curiosity I bought Cassette No. 2 from ECLECTIC - to see what they had for the slot machine. So much for curiosity and expansiveness. I was disappointed. From a practical standpoint, the ECLECTIC program barely brushes shoulders with reality. There are no "wheels" as such, spinning around; new random numbers are constantly being generated in the top row and moving downwards, so the effect is like playing with infinitely large wheels, instead of with the standard 20 character wheels. There is no way to calculate the percentage payoff is such a system. In my opinion, this is a drastic drawback to the program. Other problems include the payoff odds and the POKE system. The payoff odds tie in with the infinite wheel - they are apparently arbitrary and meaningless. The actual odds are just as great for the highest payoff as for the lowest three-of-a-kind payoff.

The POKE system leaves much to be desired, in that while the PET is POKeing the data onto the screen, all sorts of random noise flashes on the screen; white lines on black background and black lines on white background. While this enables the wheels to "spin" it detracts from the overall graphics. I have written a program which has features ECLECTIC's program doesn't have: a coin drops in the slot, the handle is pulled, the wheels (sort of) spin, and the final characters are printed. If there is a win, the proper number of coins drop into the payoff tray and the payoff odds flash. All this without a single POKE. Furthermore, it has a 20-character wheel with a calculated payout of 87.375%. In actual practice, two runs of 10,000 pulls each yielded 87.53% and 91.02%. These compare with the best in the casinos, which run from 80% to 93%. Unfortunately, I haven't been able to come up with a non-POKE routine which will show the three lines of the wheels, with the top line dropping to the middle and the middle line dropping to the bottom. I suspect this is more than an engineering detail!"

(Editor's note: Steve enclosed a copy of his program for us to look at - it's certainly the best of the Slot Machine programs we've seen! We'll keep you posted on his progress.)

Jack Clark writes "I bought the Checkbook Program from the Computer Factory last year - the ads didn't say that an 8K PET will only hold 20 checks! Very misleading advertising, I think. And the program itself leaves much to be desired. For each check entered, the user has to make nine data entries - and press the RETURN key each time. No provisions are made to prevent "fall out" if there's any RETURN key "bounce" - and if you fall out of the program, there aren't any instructions telling you how to recover without losing your data! At nine entires per check, for 20 checks, you have 180 opportunities to lose everything you've already entered. Very frustrating. I also bought Commodore's Stock Analysis Program. It's a fancy program with great graphics, but it doesn't do much, and it's hard to use. You have to have one data tape containing all the stock information like the names of the stocks, etc., and another data tape containing all the transaction information like buys and sells. One part of the program even requires a printer to use it - and Commodore hasn't even delivered any yet (that I know of)!"

EDITING

When you have typed a quotation mark, and are trying to enter cursor control characters, the PET prints the characters as part of the string. Unfortunately, when you are in "quote mode", you can't backspace to delete typing errors. There are only two ways you can correct errors from quote mode — and both of them involve leaving that mode.

The most commonly used method is to hit RETURN and get completely out of the input line. Then, by manipulating the cursor, you get back to the offending character and correct it.

The problem with this method is that you can't continue to enter cursor control characters as part of your string — because you're out of quote mode, and PET will interpret the cursor keys as an intention to move the cursor, not to print the character as part of the input string. The usual way to deal with this is to press shift and DEL — (INST) to insert empty spaces into which you can enter the cursor characters you want.

The second way of correcting errors inside quotation marks involves simply typing the closing quotes. No matter where you are on the program line, typing the second quotation mark will get you out of quote mode, and you can backspace to correct typing errors. When you have finished your corrections, you can type over the quotes (with another quotation mark) to get back into quote mode. The problem with this method is that you'll eventually have to go back and delete the embedded quotes.

The rule is that one (or an odd number of) quotes puts you into quote mode, and all the cursor controls can be made a part of your string. Two (or an even number of) quotes gets you out of quote mode, and the cursor will move when you press the cursor keys. They will not be incorporated into your string.

DELETING LINE NUMBERS

It is sometimes tedious to have to enter (one number at a time) a whole block of line numbers you wish to delete from your program. If you follow the general practice of numbering lines in increments of 10, then the procedure can be somewhat eased by this little technique:

Suppose you want to delete the block of lines from 600 - 700. In direct mode, type a FOR/NEXT loop like this one

```
FOR I=600 TO 700 STEP 10: PRINT I: NEXT I
```

PET will oblige cheerfully. When the cursor reappears, "home" it, and move it down the screen to the number 600. Press the RETURN key. Now line 600 has been deleted. Press the RETURN key for each number on the screen, then ask PET to LIST the numbers between 600 and 700 - just in case you had inserted a line while you weren't looking. If all the lines between 600 and 700 have been deleted, PET will say READY. If there are any remaining lines, PET will list them, and you can delete them singly.

Of course, if you're in the habit of assigning line numbers in increments of five, you'd use a STEP 5 in the FOR/NEXT loop!

*****

Jim Yost offers this tip for use if your PET has trouble reading cassette tapes: "On the recorder PC board is a 270 ohm (red, violet, black) resistor, between two capacitors, and a 22K resistor (red, red, orange). Paralleling this 270 with another 270 doubles the gain of the first head signal amplifier, compensating for the low tape output, recorder playback azimuth misalignment, etc. It's been working fine for me for quite awhile!"

IMPORTANT NOTICE

Aresco has moved! Our new address and telephone number should be printed all over this issue of the newsletter - in the masthead information, as the return address, in the banner line.

We'll still be running this "business" out of our home - but the move makes it possible for us to run it out of the basement instead of out of the living-room (a situation our family didn't like very much). We'll still be available by telephone on a 24 hour basis, seven days a week. If we should be out of the office, however (because we have to go to the printer, the typesetter, the postoffice, and our local computer store) (besides, we are hopelessly addicted to pinball machines), we have graduated to a new system of getting your messages.

If we happen to be at a specific place, where we can receive your calls, the calls will be forwarded to us automatically. If we aren't in a place where we can receive calls, our answering device takes over. It is a persnickety machine, however, and at this point in time, it's back at Radio Shack, getting fixed. We got too many messages like ". . . .Smith. Call anytime between eight and four, my time, at area code two-one-three, four-six-. . . . . ", and messages like ". . . . . one, four-nine-one, about the article I read in . . . . ". So we're getting it adjusted. We hope.

Another new thing is that we'll be able to handle more than one call at a time, if you're willing to be put on "hold" while we answer another call. And, if you have a question we can't answer, we can put you on "hold", call up the expert on the topic, and you and I can both talk to the expert!

All in all, the improvements should benefit us all. We're looking forward to meeting new printers, new typesetters, new people. . . . . .hopefully, some of them will be PET people!

REPLY TO THE QUESTIONS REGARDING THE FEBRUARY, 1979 (VOL. 2, No. 1)
INTRO TO MACHINE LANGUAGE
By Terry L. Laudereau

When using 6502 assembly language, it is important to realize that the mnemonics are instructions. They serve the same purpose in assembly language as the BASIC keywords serve in the BASIC language. For example, LDA and STA and ADC are assembly language instructions; LOAD, PRINT, and INPUT are BASIC language instructions.

Each of the assembly language mnemonics is representable in a hexadecimal form; called the "op-codes". The 650x Programming Manual (available from most KIM-1 distributors) gives a hex values for each instruction, and there is a "quick-reference chart" which gives the opcodes (the hex values) for each of the mnemonics.

For those who don't have a MONITOR program with their PET, it is necessary to convert the hex values to decimal values, and put the decimal values into a BASIC program to be POKEd into memory, where they will function as a "machine language program". The general format of such a program is given at the top of page 5 in issue no. 1 (February, 1979).

Line number	DATA statement containing the decimal values to be POKEd into memory
Line number	FOR statement to select the addresses into which the data will be POKEd. The first (lowest) limit of the FOR statement is the first address in which the data will be POKEd, and the last (highest) limit is the last address in which data will be POKEd
Line number	READ statement, to fetch the data from the DATA statement
Line number	POKE statement, to poke the data into memory
Line number	NEXT statement, to close the FOR loop.

Then, to run the machine language program, the user must type SYS (xxxx), and press the RETURN key. The xxxx must be the same number as the lower limit on the FOR loop.* The example in issue no. 1, for instance, POKEd data into addresses 6500 to 6512, and the user had to type SYS (6500) to run the program.

Converting the hex values to decimal numbers can be tedious, although it isn't difficult. The PET can do this very well, and if you'd rather do it yourself, there is a conversion table printed in issue no. 3, volume 1 of THE PAPER, along with instructions for using it. If you want to type in the code, it's listed here, and if you prefer, you can buy or exchange the program on our Flea Market.

```
100 REM * HEX TO DECIMAL CONVERSION
110 REM * BY RICK SIMPSON
120 REM * FOR THE PAPER
200 DIM D$(16)
210 INPUT"/CDD/HEX NUMBER TO CONVERT"; H$
220 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
230 H$=RIGHT$("0000"+H$,4)
240 DATA 4096,256,16,1
```

*This isn't always true, of course, but for the purpose of this article, we will assume it is. It is certainly the most common way of doing things!

```
250 FOR J=0 TO 15:READ D$(J):NEXT J
260 FOR I=1 TO 4:REM * FOR FOUR DIGITS
270 READ M:REM * MULTIPLIER
280 C$=MID$(H$,I,1)
290 FOR V=0 TO 15:IF C$=D$(V)THEN X=V
300 NEXTV
310 T=T+X*M
320 NEXT I
330 PRINT:PRINT T
340 RESTORE
350 GOTO 210
```

```
100 REM * DECIMAL TO HEX CONVERSION
110 REM * BY RICK SIMPSON
120 REM * FOR THE PAPER
200 DIM H$(15)
210 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
220 FOR I=0 TO 15: READ H$(I):NEXT I
230 INPUT"/CDD/DECIMAL NUMBER TO CONVERT";D
240 IF D>65535 THEN 230
250 IF D>=4096 THEN WUN=WUN+1:D=D-4096:GOTO 250
260 IF D>=256 THEN TWO=TWO+1:D=D-256:GOTO 260
270 IF D>= 16 THEN THREE=THREE+1:D=D-16:GOTO 270
280 PRINT:PRINT H$(WUN);H$(TWO);H$(THREE);H$(D)
290 GET A$:IF A$="" THEN 290
300 WUN=0:TWO=0:THREE=0
310 GOTO 220
```

Question: Do you realize why we had to use WUN as a variable name rather than ONE?

ANIMATION

By Terry L. Laudereau

Unless you want to go into PEEKing and POKEing around inside your PET's memory, it's hard to get true animation. You can bounce a ball around on the screen; even get a "paddle" to move. But it's hard to get all these things to happen at effectively the same time without the POKE statement.

Nevertheless; it's fun to experiment, and that's what this is all about - fun. So here's a short program to bounce a ball around on the screen of your PET:

```
10 FOR I=1 TO 39: PRINT"/  /●/L/";:NEXT I
20 FOR I=1 TO 39: PRINT"/  /L/L/●/L/";:NEXT I
30 FOR I=1 TO 25: PRINT"/  /L/D/●/L/";:NEXT I
40 FOR I=1 TO 25: PRINT"/  /L/U/●/L/";.NEXT I
50 GOTO 10
```

CURSOR KEYS
/C/ = Clear Screen
/ / = Space
/L/ = Cursor Left
/D/ = Cursor right
/●/ = Shifted Q

This will only bounce a ball across the bottom of the screen, but if you know that screen memory starts at 32568 and officially ends at 33792, then you can do the rest - and get your "ball" to bounce all over the place!

Another way to bounce a ball is with the POKE statement. Be careful of using POKE, however, because you can fill the screen with "snow" or "hash" if your timing isn't impeccable. To help the timing a bit, we'll use the WAIT keyword in this program:

```
10 PRINT"/C/":C=81:D=32
20 A=33568:B=33607:E=1:GOSUB 50
30 H=A:A=B:B=H:E=-1:GOSUB 50
40 GOTO 20
50 FOR I= A TO B STEP E
60 WAIT 59456,32,32:POKE I,C
70 WAIT 59456,32,32:POKE I,D
80 NEXT I:RETURN
```

This program will bounce the ball from left to right, then from right to left; down the screen and back up. Over and over, until you press STOP. It might be an interesting exercise to get the ball to bounce "around" the screen - or to get it to bounce off the edges of the screen in a relatively random direction. Care to try it?

DATA FILES CONTAINING STRINGS

By Paul M. Julich

The article by Jim Butterfield, "Memory Usage And Garbage Collection" (Volume 1, issue no. 7, PET User Notes) provides a key to another problem with the reading of data files.

To illustrate the problem, type in the following program. Place a fresh cassette in the recorder, and RUN the program to write the string variables on the tape.

```
10 OPEN 1,1,1
20 T$="ONE"
30 PRINT  # 1, T$
40 T$="TWO"
50 PRINT  #  1, T$
60 T$="THREE"
70 PRINT  #  1, T$
80 CLOSE 1
90 END
```

Now, rewind the tape, remove the above program from memory by typing NEW, then enter the program here:

```
10 OPEN 1,1,0
20 INPUT  # 1, A$
30 INPUT  # 1, A$
40 INPUT  #1, A$
50 CLOSE 1
60 END
```

RUN this program to read the data from the tape, then (following Butterfield's article) look at the upper end of memory (string storage space) by entering

```
FOR J=8161 TO 8191 : PRINT CHR$(PEEK(J) );:NEXT J
```

You'll see the appearance of the desired string "THREE", in addition to the "garbage strings" "ONE" and "TWO". If, in the course of reading a tape, you fill up the string space, the PET will execute the "garbage collection" routine to free up some of the string storage space. Since this routine is time-consuming and complex, the PET will miss the interrecord gap or necessary control characters which may be being INPUT while the routine is being executed. This may cause the PET to "hang up", lose its cursor, and necessitate your RESETing (by power-down and power up) to free it.

Two points are therefore of importance:

1. Always include an instruction such as A=FRE(0) to force the PET to collect its garbage immediately prior to reading a data file. This will assure you that you have the maximum amount of string storage space available when entering the tape read routine.

2. Be careful with the structure of data files consisting of string variables. This is especially important in large programs, in which string space may be extremely limited. Use as few string variables as possible, and if you can, use the same variable designations in different parts of your program (where the usage won't conflict with anything).

CORRECTIONS TO ROBERT BUXTON'S FAST FORWARD PROGRAM, ISSUE No. 1

Even with the marvel of electronic typesetting, we seem to be having problems with proofreading out text! We do apologize, and offer our thanks to Robert Buxton, who sent us the corrections:

Lines 20-30: The words "Program name and load point" are not part of the program. They were put there to help you know what the name and numbers in the data section are.

Line 510 IF PEEK(59411)<>53 GOTO 510
Line 540 IF TI$<>R$ GOTO 540
Line 600 Omit the E at the end of the typed line. The word should be PRESS, not PREESS.
Line 610 IF PEEK(519)<>0 GOTO 610

Then, in the section of the article describing the graphics used, include these two:

/F/ Reverse Field on
/O/ Reverse Field off

That's all of them, I think. Again, we apologize to all.

LATEST UPDATE - CASSETTE FILES

By Terry L. Laudereau

During the course of Volume 1, we received a lot of information from readers about tricks they used to read and write data files reliably. Here is a collection of those ideas - the ones we have adopted for use here at ARESCO:

1. Before opening a file for writing, the cassette's start address has to be initialized. You do this by including a couple of POKEs before your OPEN statement:

```
For cassette no. 1:    POKE 243,122:POKE 244,2: OPEN 1,1,1
For cassette no. 2:    POKE 243,58:POKE 244,3:OPEN 2,2,1
```

2. When writing data to tapes, it is necessary to write a gap between each block of data lest PET somehow miss its own IRG when reading the file back later. In your main program, include the following lines for each and every PRINT # statement. ("xxx" is a line number of your choice, "n" is the logical file number, and "X$" is the variable you want to write to tape.)

```
xxx T=TI:PRINT#n,X$
xxx IF TI-T>10 THEN GOSUB 9000
```

Then include one of these two subroutines, depending on which cassette unit you're using in your program. If you're using both cassette units for writing, of course, you'll have to re-number one of the routines - and change the GOSUB address in the lines above.

```
9000 REM * CASSETTE # 1
9010 POKE 59411, 53
9020 FOR C1=1 TO 100: NEXT C1
9030 POKE 59411,61:RETURN
```

```
9000 REM * CASSETTE # 2
9010 POKE 59456,207
9020 FOR C2=1 TO 100:NEXT C2
9030 POKE 59456,223:RETURN
```

3. Don't use more than one variable in a PRINT # statement. And use only one PRINT # statement per program line.

4. Don't PRINT # any strings more than 80 characters long. The 80th character should be a carriage return, so that if you type PRINT # 1, X$, X$ should only be 79 characters. After you type X$, you press the carriage return, and that will be written to the tape as the 80th character.

5. Always CLOSE the file when you're done with it. Data may still be in the tape buffer - and if you don't close the file, it won't be written to tape.

6. When reading a tape, always check the STatus - after every read. If ST=0, then no errors were found (but even that does not guarantee you a good read!). If ST=64, you've come to the end of the file, and if ST=128, you've come to the end of the tape. Any other value is indicative of an error somewhere, and you should try again.

7. If you have opened the file for writing to tape, and then encounter a ?SYNTAX ERROR in the program somewhere, PET BASIC aborts the program, leaving your file open - for writing. This could be disastrous! CLOSE the file in direct mode, and then start over from RUN. (You don't have to close the file, it says here in fine print, since RUN automatically closes it for you. Still. . . . .)

8. Some PETs cause the cursor to misbehave if the file is not closed "normally" - that is, during the program run. If your cursor should ever get ideas of its own about where it should appear on the screen, follow these three steps:

a. Clear the screen
b. Type PRINT #
c. Press RETURN

One of my PETs gives me back a well-behaved cursor at this point, with no further comment. The other PET insists on getting the last word in, and generates another ?SYNTAX ERROR before it begrudgingly announces that it's READY.

9. PET is a real garbage collector, and collects all sorts of strange things in space that is supposed to re-reserved for string storage. I have found it usefull to force PET to collect all that garbage by typing

```
xxx ZZ=FRE(0)
```

This works because in order to tell you how many bytes are available to you, PET has to do a little housecleaning and get rid of all the garbage.

10. There are a number of (tedious) routines you can use to determine the number of characters that have been written to the tape file. Location 625 contains that number - just PEEK at it while writing your tape (between PRINT # statements). If it contains 190 characters, write a carriage return, then turn on the cassette motor (as in item no. 2 above) to write an IRG. Or you can count the characters as you write them. But to do that, you must remember to count the carriage return at the end of each PRINT # statement, and you must remember that the PET puts a "cursor-right" at the end of each number, and you must remember that each number has a "sign-space" in front of it, and you must remember. . . . . .

All in all, I think we've got the data file problem licked at this point - thanks much to all of you who have contributed to the fund of knowledge. Of course, we'll keep you posted it anything new turns up.

"PROGRAMMING" A FLASHING CURSOR

By Terry L. Laudereau

One of life's biggest frustrations (for a program user) is to "fall out" of a program by accidentally pressing the RETURN key. You, as the programmer, can protect the user from such a frustration by programming a "flashing cursor" into your game. You should be should be sure to use some character other than the white square used by the PET, so the user won't get confused and think he's fallen out.

```
1000 GET A$
1010 PRINT"/▒ /L/";:FORI=1 TO 100:NEXT I
1020 PRINT"/  /L/";:FORI=1 TO 100:NEXT I
1030 IF A$=" "GOTO 1000
1040 PRINT A$;
```

To test this one, add a line 1050 GOTO 1000. Your characters will be input and printed, albeit a little more slowly, as if the PET itself were flashing the cursor.

This is all well and good, if your program requires a simple "Y/N" type of response. But what if you want your user to enter a longer string - like his name, for instance?

```
1000 GET A$
1010 PRINT"/▒ /L/";:FOR I=1 TO 100: NEXT I
1020 PRINT"/  /L/";:FOR I=1 TO 100: NEXT I
1030 IF A$=" " GOTO 1000
1040 IF A$=CHR$(13) GOTO 1080
1050 PRINT A$;
1060 B$=B$+A$
1070 GOTO 1000
1080 PRINT B$
```

This program will continue to input characters until a RETURN is pressed. True, if the user accidentally hits the RETURN key before he gets all the data entered, he could "fall" out but, at the worst, he'd fall into your program, not out of it. You can check for the length of the string he entered, for example, and if it isn't long enough, you can put him right back into the routine.

```
1090 B=LEN(B$):IF B>5 THEN 1110
1100 B=B-1: B$=LEFT$(B$,B): GOTO 1000
1110 . . . . . .
```

And, of course, line 1110 can contain a RETURN to the main program. Line 1080 should be deleted after you've tested the routine - it's there for your benefit, not for the user's.

Kvenich & Associates announces that they are now distributing the Micro-Systems Engineering Group WAVECOM Interface, a stand-alone device which will interface PET to a Selectric typewriter. The WAVECOM interface contains a processor and ROM, so the user needn't use the second cassette buffer except for holding his own applications programs!

WAVECOM plugs into the PET via the IEEE-488 bus. While installation of solenoids and plugs are required for the Selectric all the necessary parts are provided. The user or his dealer must install them. WAVECOM can also be used with a Selectric terminal - communication is via a telephone handset, through an acoustic coupler. WAVECOM will allow the use of TAB and backspace keys, as well as all cursor control keys.

Designed to permit comprehensive word processing, the WAVECOM Interface retails for $1495.00, from Kvenich & Associates, 151 Carlingview Drive No. 5, Rexdale, Ontario, M9W 5E7, CANADA. Be sure to write for ordering and delivery information.

MORE ABOUT EXTENDED GRAPHICS

(from The Software Shoppe)

By Roy Busdiecker

In the hands of a programmer who understands its capabilities, this package provides the basis for impressive displays on the PET. It does everything it promises, and does it well.

Without detracting from the powerful potential of the program, several warnings should be noted.

– The package is not for non-programmers or beginning programmers, unless they are satisfied to watch the fixed (but extensive) sequence in the demonstration program. It is intended to be called as a subroutine from the program that you write.

– The package does not provide protection against invalid numbers. Once you've run the program with a number outside the allowed range, it no longer performs as advertised. The only way I could restore normal operation was to re-initialize the PET, reload EXTENDED GRAPHICS, run it, then reload my program. . . .a heavy penalty for the error!

– In operation, the program does not seem to be "human-engineered". Operator responses are obtained via INPUT statements, not protected. Numbers assigned to functions did not seem to have any pattern to make them easier to remember. Instructions are minimal, and in some cases, misleading (as in the case of determining where to load the machine-language routine).

Extended Graphics operates on a "strip" of screen. The strip may be horizontal or vertical, and may be as narrow as one row or column or as wide as the full screen. The important fact is that the strip extends from one edge of the screen to the other, and every screen location in the strip is affected just the same as every other.

Here are the operations that the package will perform:

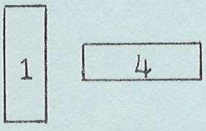
– fill the strip with a specified character (1 for vertical, 4 for horizontal).

– shift the strip with "wrap-around" (as a character is shifted off one edge of the screen, it is shifted back on to the opposite side).

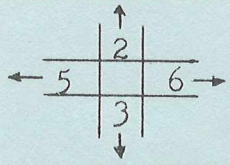
– shift the strip without "wrap-around" (when it is shifted off the screen, it disappears).

This information is summarized below, with function numbers.

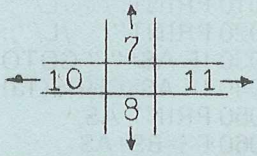
FILL



SHIFT WITH WRAP



SHIFT OFF SCREEN

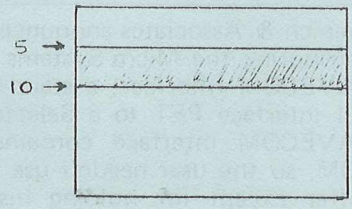


Extended Graphics is controlled by the values POKEd into five locations. If we set SP equal to the value of the Starting Point in memory where the program is loaded (we note how to calculate this at the end of the article), then

ADDRESS	CONTAINS
SP	Function (from above diagrams)
SP+1	Starting row/column
SP+2	Ending row/column
SP+3	Magnitude of shift (for functions 2,3,5,6,7,8,10,11)
SP+4	Value which determines character (for functions 1,4) NOTE: This is the "screen code", NOT the ASCII value determined by the ASC ( ) function!

For example to fill rows 5 through 10 with asterisks, a program like this

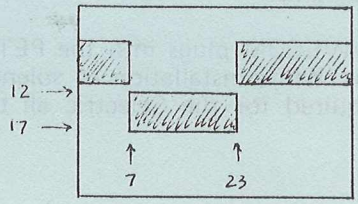
```
10  FOR I=ØTO4:READ C:POKE SP+I,C:NEXT I
12  SYS Ø
15  DATA 4,5,1Ø,Ø,42
16  REM FUNCTION, START, END, MAGNITUDE(DUMMY), CHAR
```



will do the job (in an instant), leaving the display shown at right.

Now to pull a "chunk" down from the middle of the display, use

```
20  FOR I=ØTO4:READ C: POKE SP+I,C:NEXT I
22  SYS Ø
25  DATA 8,7,23,7,Ø
26  REM FUNCTION, START, END MAGNITUDE, CHAR(DUMMY)
```



If a series of ten data statement is prepared, than the program segment which follows can execute the functions so quickly that they will give a "motion picture" effect (get rid of the previous examples before entering these lines). Data statements you develop can be numbered 10,20,30. . .

```
5   FOR J=1TO1Ø
6   FOR I=ØTO4:READ C: POKE SP+I, C:NEXT I
7   SYSØ
8   NEXT J
```

To make the ten functions repeat endlessly, add

```
9   RESTORE:GOTO5
```

Finally, the value of SP can be calculated by

```
2   SP=PEEK(1)+256*PEEK(2)-94
```

assuming that the computer has not been turned off or reset since the Extended Graphics program was run.

SOFTWARE MAINTENANCE BULLETIN

MICRO-SET I (for PET)

Version	Date	Comment

NOTE: Customers with versions prior to 1.7 may obtain a current version by sending their original cassette with $2 and their name and address to MSS.

1.7	14 Dec 78	Changes line-number allocation for compatibility with SOFTOOL KIT products.
1.71	31 Jan 79	Discrepancy: Appearance of the word "READY" in a program causes the ADD FROM TAPE function to terminate prematurely.

Correction: Add the segments underlined below.

```
63875 IF B$ = "READY" AND POS( Ø )<=1 THEN 63845
```

1.72	10 Feb 79	Discrepancy: Tape reproduction error caused appearance of two Ø/Ø entries between each valid entry in the NEW LINE/NEW TARGET table in RENUMBER function. Program functions properly, but produces extraneous printout.

Correction: At the end of line 60544, change T7=T7+3 to read T7=T7+1

Randall Lockwood writes about his experiences with vendors:

1) Computer Factory (NY) - Dennis notes that he had bad delivery and lousy tapes. My only contact with this company has been to drop in in person. Then they were very friendly and let me try out any software I wanted on the PET's set up in the store. The drawback was that they only had a few of the simpler games, and as Dennis indicates, said that the stuff I really wanted to see would be in "soon"

2) AB Computers, Perkasie PA. - So far this company has been incredible compared to the rest of the field. I first ordered some blank tapes and edge connectors. I paid using a personal check. The order arrived UPS 4 days later. The tapes had dropped in price so they had sent a few extra. On my next order I requested more blanks and Microchess. The order arrived in 5 days, everything worked well. I sent in a third order for "Tunnel Vision" and more blanks and received it in 4 days along with a nice personal note and some extra cassettes (the price had dropped again). The program was nice and very reasonably priced. All in all. . . .complete satisfaction.

3) Creative Computing (NJ) - You would think this group would be particularly good. In January I ordered "Graphics Games and Sports Games". About a month later I received a note saying they were back ordered. Two weeks later I got another note indicating that "Sports Games" was now "Action Games" and had new programs but was not going to be available for several months. A refund certificate was enclosed for that tape. The other is still on back order as of now (end of March). . .so I have not yet seen any of the software.

APPLICATION NOTE NO. 1

Software Delay for Slow Carriage Return Printers Using the CmC ADA 1200 C — Courtesy of Fred C. Winograd

```
40 PRINT"CLR CARRIAGE RETURN DELAY ROUTINE"
50 PRINT"COURTESY OF FRED C WINOGRAD"
60 PRINT
70 PRINT"     THIS PROGRAM WILL LOAD A SPECIAL  ROUTINE THAT WILL DELAY ";
71 PRINT"OUTPUT AFTER     EACH CARRIAGE RETURN OUTPUTTED THROUGH     THE IEEE";
72 PRINT" PORT.  THIS DELAY WILL GIVE A SERIAL PRINTER ENOUGH TIME";
73 PRINT" TO PHYSICALLYRETURN BEFORE ADDITIONAL CHARACTERS ARE"
74 PRINT"PRESENTED FOR PRINTING."
75 PRINT
76 PRINT"  TO ENABLE THE DELAY ENTER THE COMMAND 'SYS(832)'."
77 PRINT"  TO DISABLE THE DELAY ENTER THE COMMAND 'SYS(845)'."
78 PRINT
79 PRINT"  AFTER THE ROUTINE IS LOADED, THE       SECOND CASSETTE CANNOT";
80 PRINT" BE USED."
81 PRINT
82 PRINT"  ****************************************"
83 PRINT"  ALSO **PLEASE NOTE** THE DELAY MUST BE";
84 PRINT"DISABLED WHILE USING CASSETTE #1."
85 PRINT" ****************************************"
86 PRINT
87 PRINT"HIT ANY KEY TO CONTINUE";
88 GETC$:IFC$=""THEN  88
110 X$="0123456789ABCDEF"
120 L$="2223223332112"
122 PRINT
125 PRINT"CLR LOADING DELAY ROUTINE INTO MEMORY!!!"
130 GOTO700
300 PRINT:PRINT"ROUTINE HAS BEEN LOADED!!!"
310 PRINT:PRINT"THIS ROUTINE WILL REMAIN IN THE"
320 PRINT"SECOND CASSETTE BUFFER UNTIL THE"
330 PRINT"PET IS TURNED OFF."
340 PRINT:PRINT"LOADING ANOTHER BASIC PROGRAM WILL"
350 PRINT"NOT AFFECT THE DELAY PROGRAM."
600 END
700 REM
710 READH$
720 A$=LEFT$(H$,4):K=1:GOSUB950:S=A
730 IFS=0THEN300
740 K=3:FORJ=0TO(LEN(H$)-4)/2-1
750 A$=MID$(H$,5+J*2,2):GOSUB950
760 POKES+J,A
770 NEXTJ
780 GOTO710
950 A$=RIGHT$("0000"+A$,4):A=0
955 FORJ1=KTO4
960 C$=MID$(A$,J1,1)
965 IFC$>="A"ANDC$<="F"THENC=ASC(C$)-ASC("A")+10
970 IFC$>="0"ANDC$<="9"THENC=ASC(C$)-ASC("0")
975 A=A*16+C
980 NEXTJ1
985 RETURN
1000 DATA 034078A9638D1902A9038D1A02586078A9858D1902A9E68D1A025860A50048484848
1001 DATA 03604C85E678AD3F03C900F012AD22E8C9F2D0154C85E6A9018D3F034C85E6AD22E8
1002 DATA 0380C9F2F0F14C85E6A9FF8D3E03A9FFE901D0FCAD3E03E9018D3E03D0F0A9008D3F
1003 DATA 03A0034C85E69884891C99208881B91998898935080DBC999A9CF8A88DAAAD880C2D
1004 DATA 0000
READY.
```

Application Note Number 2.
Output Formatting Routines for the Commodore PET.
Using the CmC ADA-1200

```
2 REM FORMAT PROGRAM
3 REM WRITTEN FOR THE COMMODORE PET
4 REM BY RICHARD ROSNER
5 REM     CONNECTICUT MICROCOMPUTER
6 REM     150 POCONO ROAD
7 REM     BROOKFIELD, CT 06804
8 REM
30 REM THIS PROGRAM PRINTS FUNCTIONS OF X AS
31 REM X GOES FROM 0 TO 1000 IN STEPS OF 50.
115 REM THIS PROGRAM WAS WRITTEN TO ILLUSTRATE A METHOD OF PRODUCING
116 REM RIGHT JUSTIFIED OUTPUT, OR ALIGNED DECIMAL OUTPUT, ON A
117 REM HARD COPY UNIT FROM THE PET.
118 REM THE HARD COPY UNIT USED IS A GE TERMINET AND THE ADAPTER IS
119 REM A CMC PET ADA 1200.
120 REM
6000 REM REQUEST INPUT FOR RIGHT JUSTIFY OR DECIMAL OPTIONS.
6080 PRINT"TYPE 'R' FOR RIGHT JUSTIFY OR 'D' "
6090 INPUT"TO ALIGN DECIMAL POINT. ";F$
6100 F$=LEFT$(F$,1)
8100 REM CONVERT FROM GRAPHICS TO LOWER CASE LETTERS FOR THE PET SCREEN.
8102 POKE 59468,14
8109 REM OFILE IS THE OUTPUT FILE NUMBER.
8110 OFILE=5
8119 REM DEVICE IS THE OUTPUT DEVICE ADDRESS.
8120 DEVICE=5
8128 REM NULL$ IS THE NULL CHARACTER PRINTED AT THE BEGINNING OF EACH LINE FOR
8129 REM PRINTERS WITH LONG CARRIAGE RETURN TIMES.
8130 NULL$=CHR$(0)
8140 REM NNULL IS THE NUMBER OF NULL CHARACTERS PRINTED AT THE BEGINNING
8141 REM OF EACH EACH LINE
8142 NNULL=5
8149 REM OPEN THE OUTPUT FILE.
8150 OPEN OF,DE,1
8159 REM PRINT THE HEADINGS.
8160 GOSU 13000
10100 FOR N1=0 TO 1000 STEP 50
10199 REM CALCULATE THE FUNCTIONS OF X.
10200 XO(0)=N1
10210 XO(1)=N1*N1
10215 IF N1=0 THEN XO(2)=0:GOTO 10230
10220 XO(2)=1000/N1-3.2
10230 XO(3)=N1
10240 XO(4)=(-2)  (50-N1/10)
10250 XO(5)=(XO(0))  .5
10299 REM PRINT A FORMATTED LINE OF FUNCTIONS.
10300 IF F$="R" THEN GOSUB 15000
10310 IF F$="D" THEN GOSUB 16000
10350 NEXT N1
10399 REM CLOSE THE OUTPUT FILE
10400 CLOSE OF
10900 END
12999 REM PRINT THE HEADING.
13000 PRINT#OF:PRINT#OF:PRINT#OF,"Functions of X"
13010 GOSUB 18000
```

```
13020 PRINT#OF
13030 PRINT#OF,"     X     X 2     1000/X-3.2     X PI     (-2) (";
13040 PRINT#OF,"50-X/10)     X .5"
13050 GOSUB 18000
13100 PRINT#OF
13150 GOSUB 18000
13999 RETURN
15000 REM RIGHT JUSTIFY SUBROUTINE
15009 REM THE TX(N) ARE THE TAB SETTINGS
15010 TX(0)=7
15011 TX(1)=16
15012 TX(2)=31
15013 TX(3)=47
15014 TX(4)=64
15015 TX(5)=77
15019 REM PSITIN IS THE POSITION OF THE TYPE HEAD ON THE OUTPUT LINE.
15020 PSITIN=0:FL=1
15030 FORN=0 TO 5
15039 REM MOVE THE TYPE HEAD TO THE STARTING POSITION OF THE NEXT NUMBER.
15040 TX=TX(N)-LEN(STR$(XO(N)))-PS
15050 IF TX  =0 GOTO15070
15060 PRINT#OF," ";:TX=TX-1:PS=PS+1:GOTO15050
15069 REM PRINT THE NUMBER.
15070 PRINT#OF,FL*XO(N);
15080 PS=PS+LEN(STR$(XO(N)))
15090 NEXTN
15099 REM CARRIAGE RETURN-LINE FEED.
15100 PRINT#OF
15110 GOSUB 18000
15200 RETURN
16000 REM ALIGN DECIMAL POINT SUBROUTINE
16009 REM THE TX(N) ARE THE TAB SETTINGS
16010 TX(0)=6
16011 TX(1)=15
16012 TX(2)=20
16013 TX(3)=44
16014 TX(4)=61
16015 TX(5)=70
16019 REM PSITIN IS THE POSITION OF THE TYPE HEAD ON THE OUTPUT LINE.
16020 PSITIN=0
16030 FOR N=0 TO 5
16031 REM MOVE THE TYPE HEAD TO THE STARTING POSITION OF THE NEXT NUMBER.
16035 FL=1
16038 IF XO(N)  0 THEN XO(N)=-XO(N):FL=-1
16040 IF XO(N)  =1E9 THEN L=LEN(STR$(XO(N)))  -6:GOTO 16070
16045 IF XO(N)  .01 THEN L=LEN(STR$(XO(N)))-6:GOTO 16070
16050 IF XO(N)=0 THEN L=0:GOTO 16070
16060 L=INT(LOG(XO(N))/LOG(10))
16070 IF L  0 THEN L=-1
16080 TX=TX(N)-L-PS-3
16090 IF TX  =0 GOTO15070
16100 PRINT#OF," ";:TX=TX-1:PS=PS+1:GOTO16090
16101 REM PRINT THE NUMBER.
16120 PRINT#OF,FL*XO(N);
16130 PS=PS+LEN(STR$(XO(N)))
16140 NEXTN
16150 GOTO 15100
18000 FOR N=1 TO NN:PRINT#OF,NU$;:NEXTN:RETURN
READY
```

YES! Please enter my

New Subscription Order for all ten issues of Volume 1 _____

Renewal Subscription order for all ten issues of Volume 2 _____

New Subscription Order for all ten issues of Volume 2 _____

Full payment of $15.00 per volume ordered is enclosed.**

NAME (Please print or type _____

ADDRESS (Street, not P.O. Box) _____

CITY _____ STATE _____ ZIP _____

Please charge my /MC/VISA/BAC/ # _____

Master Charge Interbank # _____ Exp. Date _____

Required Credit Card Signature _____

Please ship the following Software Shelf Programs to me within two weeks. I enclose payment in full.**

PDPS FILE CREATE	$15.00 ____	WAREHOUSE	$ 7.95 ____
STAT III	$ 7.95 ____	CHASE	$10.00 ____
DATA RETRIEVAL	$10.00 ____	MICROMAZE	$ 7.95 ____
DATA EDIT	$10.00 ____	BLACKJACK	$10.00 ____
STAT I	$20.00 ____	BLOCKADE	$10.00 ____
STAT II	$20.00 ____	DEFLECTION	$10.00 ____
SPACE FIGHT	$10.00 ____	STARTREK 2001	$10.00 ____
HOSTAGE	$ 7.95 ____	XMON	$15.00 ____
AIR-SEA WAR	$ 7.95 ____	CMC/WPP	$29.50 ____
TUTOR	$19.95 ____	TUTOR PACKAGE	$39.95 ____
BASIC COMPLEAT	$29.95 ____	BAZAAR	$ 7.95 ____

I understand that if any of the programs fail to load, THE PAPER will send me a replacement as soon as I return the defective tape.

NAME (Please print or type) _____

ADDRESS (Street, not P.O. Box) _____

CITY _____ STATE _____ ZIP _____

Please charge my /MC/VISA/BAC/ # _____

Master Charge Interbank # _____ Exp. Date _____

Required Credit Card Signature _____

** No facilities are available for billing or for C.O.D. delivery. Orders submitted without payment will be returned. Sorry folks, but nearly 1/10th of the subscribers we have so far asked for billing — and haven't paid up. If everyone paid in a timely fashion, we might be able to reduce the subscription price!

EDITORIAL - Continued from page 2

level and in the commercial (short-course) environment. He's developed some of those courses himself. He's a member of the IEEE Computer Society, and the Association for Computing Machinery. Roy has been a PET owner since December, 1977, and has written a number of articles for us (and for other publications) about the PET. We are delighted to be able to "claim" him as out Corresponding Editor.

We have shipped off several new Software Shelf programs for review by some of our most critical readers. One of them (who shall be unnamed at this point in time, for reasons of personal security) has never written a kind work about any program-that we know of! We hope to have that material back by next issue. And, of course, we have more columns by Dr. Deardon - more new products to tell you about - and more on machine language programming.

THE PAPER

P.O. BOX 1142
COLUMBIA, MD 21044

SECOND CLASS POSTAGE
PAID IN COLUMBIA, MD 21044
(USPS 450-930)

TO: