

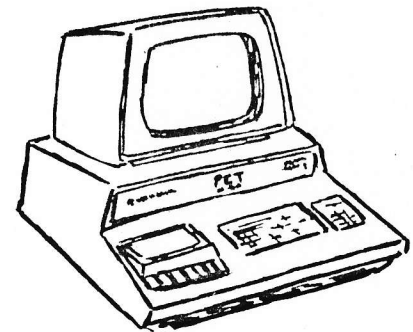
**BOX 43
AUDUBON
PA 19407**

THE



PAPER

**BOX 43
AUDUBON
PA 19407**



Editorial

Since this is being written only about two weeks after we mailed the first issue of The [redacted] Paper, we have just started to get feedback from you. It's nice to get subscriptions - the trip to the post office is an exciting part of each day - but what we look forward to most is getting a letter, so we know how you feel about what we are trying to do.

We started The PET Paper for the same reasons that Americans start hundreds of organizations every day - in the belief that if we work together and share what we know, all of us will benefit. It doesn't matter whether the subject is breeding goldfish, raising funds for medical research, collecting postage stamps, or swapping antiques. Americans have a passion for information, and a desire to share what they have learned with others.

We have already heard from several local User Groups which have sprung up spontaneously; Terry attended such a meeting in Berkeley, California, the other night which was attended by about fifty PET enthusiasts.

We have been very happy with the response we have received from you to the software tapes we offered. No one has been unable to load the tapes due to cassette reading problems, although some people have had to try several times. That bears out our suspicion that most tape problems we have heard about have been due to problems with tapes "mass duplicated" on commercial recording equipment. We will probably have to use these methods ourselves for tapes such as 'Useful Routines' which contain a large number of programs. The time involved in recording and verifying so many programs on a single \$5 tape just can't be spent. However, we will continue to 'hand-duplicate' all major programs in the library.

Speaking of the tape library, we have been pleased with the number and quality of tapes you have submitted. We do not know whether there are legal problems with our distribution of programs which have appeared in such books as "101 Computer Games" and have been only slightly modified to work on the PET. We do know that if you simply use the game concept, without using the actual BASIC coding that is printed in the books we are on safe legal grounds. The real challenge for a PET programmer is to take common games and adapt them to PET's unique graphic capabilities. Games like Deflection just couldn't be put on a computer which used an ordinary terminal, but I have seen at least four versions of the game Chase, which has appeared in several computer magazines for use on regular printing terminals, using the special capabilities of the PET. Each programmer used his own ingenuity in applying the game to the special capabilities of PET. The best one I have seen is a new addition to the Software Shelf this month.

One of the things which has surprised us most about your response to our software offerings is the tremendous interest you have shown in 'practical' as opposed to 'game' programs. Address File and Data Retrieval For The Home are by far the most requested programs. We apologize to those who ordered Address File and got an early version with several serious bugs. All of you who ordered it and got the wrong version have been notified and should return the tape for replacement. Both Address File and Home Data Retrieval incorporate

NEW PRODUCT ANNOUNCEMENT

THE NET WORKS, TNW 488/232, Serial Interface

INTERFACES IEEE 488 BUS TO RS232C AND MIL STD 188C

Now you can use RS232C and Mil Std 188C serial peripherals with your Commodore PET or other IEEE 488 bus computer system. The TNW488/232 provides two serial I/O ports to interface both commercial and military surplus printers, modems, CRT terminals, plotters, paper tape readers and punches, and even other computers to your system. Because power supplies are built right on the board, you can just plug the power cord into a standard outlet, connect cables to your PET and peripherals, and start to use your system. A wide range of baud rates accomodates both fast and slow devices.

The interface is available as assembled and tested board with documentation and 90 day warranty in dual port (\$280) or single port (\$240) configurations, or as bare board (no components) with documentation (\$50). Cabinet (\$35). IEEE 488 Interface cable (1 meter long, "PET style" connectors) (\$20)

The Net Works, 5924 Quiet Slope Dr., San Diego, CA 92120
Represented by Astronics, 4805 Mercury St., San Diego, CA 92111
(714) 278-5441

New Product Announcement

The Net Works has also announced their TNW 488/103 low-speed modem interface for the PET. This unit allows your PET to automatically dial another PET or receive such a call and transmit and receive data over phone lines. It connects directly to a "CBT" type Data Access Arrangement available from your phone company. A complete power supply is included. Shipment is within 60 days. Prices: completely assembled: \$320 - bare printed circuit board and documentation: \$60 - Cabinet: \$35 - PET cable: \$20 - Telephone cable: \$15.

Editor's note: I think this telephone interface offers fantastic possibilities for PET-to-PET communication. I would like to hear from anyone who has one in use and would like to publish information on how you are using it.

NEW PRODUCT ANNOUNCEMENT

International Technical Systems announces an 8K fully assembled memory expansion unit, ready to plug in to your PET. The attractive housing is 10" x 10 1/2" x 4 1/2". It weighs 5 pounds and costs \$279.00.

International Technical systems, Inc. ; Box 264 ; Woodbridge, VA, 22194

Product Review
The Connecticut MicroComputer ADA-1200
Printer Adapter

When the PET was first introduced I wasn't particularly bothered by the lack of peripherals (printers, telephone interface, memory expansion, etc.) because I was sure that if Commodore didn't offer peripherals but did build thousands of PETs, American entrepreneurship would prevail and other people would see the opportunity and build PET-compatible products. One of the first such products offered was the Connecticut MicroComputer Printer Adapter.

Many PET owners have or have access to a standard printing communications terminal such as the Texas Instruments Silent 700, the GE Terminat 300, or Univac DCT-500. Most of these terminals, originally designed for use with commercial time-sharing services, use a standard electrical interface designated as RS-232C. This interface was adopted by the EIA (Electronics Industry Association) so that there would be a standard plug and set of electric signal levels for communication terminals. These were usually connected to a modem (modulator-demodulator) which allowed data transmission over normal telephone lines. The RS-232C standard has also been adopted by many microcomputer manufacturers for use with the serial input-output ports on their equipment. Commodore chose, for a variety of good reasons, not to use RS-232C but rather the IEEE-488 standard. Thus an adapter such as this is required to convert the parallel data supplied by the IEEE-488 port to serial data used by RS-232C. Since I have a printer with an RS-232C interface, I contacted Connecticut Micro Computer (CMC) and got one of their interface adapters, which sells for \$98.50 completely assembled. The unit I received requires both + and - 12 volt DC power, which I had available from an auxiliary power supply. CMC also can provide a power supply as an option.

The unit is assembled on a single 4" X 6" printed circuit card, with the plug which connects to the PET attached right to the card. The purchaser must provide his own cable to his printer which is connected to two points on the printed circuit.

I attached the unit to my PET power supply and printer, and just three short days later I was getting printed copies of my PET programs! The reasons for the three-day delay would make an article by themselves, but none of them was the fault of the CMC adapter.

The CMC adapter can be adjusted to work with printers running from 110 baud (10 characters per second) to 1200 baud (120 characters per second). Specify the rate you intend to use when you order and CMC will preset it for you.

There are two limitations you should be aware of. First, the adapter (and your printer) doesn't handle PET graphic characters or cursor control symbols. Sometimes a letter or number will be printed, sometimes nothing. Second, the CMC adapter does not allow any other device to be connected to the PET IEEE port. (For our technically sophisticated user, this is because the CMC adapter does not recognize the IEEE port device addresses and will respond to ANYTHING put on the port.) But if you want a quick, inexpensive way to put a printer on your pet, the CMC adapter is a good way to go. You can contact CMC at 120 Pocono Rd., Brookfield, CT.

special cassette read and write routines which get around the problem of missing records which Commodore mentions in their software bulletin on cassette tape files. If you have not received a package of bulletins from Commodore, they are intending to issue updated versions to all PET owners sometime in May. Contact your dealer if you did not order your PET directly from Commodore.

Because of the strong response to Home Date Retrieval, we would like to encourage those of you who want to create software for the PET community to construct more programs for use with the file structure used in that program. We have received several suggestions for enhancements such as the ability to edit entries placed in memory and add new entries directly to memory. A program which allows the user to automatically select tape records by rule, such as 'all records with a value greater than 100 in field two' would be very useful too.

So here is issue #2! There is still less technical information than we would like, but we have just received a large quantity which will begin to appear in issue #3. This issue marks the first appearance of reprinted material; in this case, two excellent articles by Dan Fylstra of Personal Software giving some basic information on the use of the User Port. Enjoy!

CORRECTIONS TO PREVIOUS ISSUES

Pardon us, people, but when our first issue was barely back from the printers, we found an error in the GET article. The article contains a 5-line program on its first page:

```
10 GETA$:IFA$=""THEN 10
20 IF A$="O"THEN 50
30 PRINTA$
40 PRINT"DONE":END
50 PRINT"PRESS THE SPACE KEY":GOTO 10
```

Line 50 should read; 50 PRINT"PRESS ANOTHER KEY":GOTO 10

And, in the description below the program, the second sentence starts "Line 20 tells PET that if A\$ does not contain a space ...". The sentence SHOULD say: "Line 20 tells PET that if A\$ contains a space....".

Sorry, folks. We're trying to keep such errors to a minimum. If you find others, please let us know so we can print corrections. Thanks for your indulgence and patience.

* * F L A S H * *

Commodore Business Machines feels that the word "PET" is proprietary, and prefers that we (and others who are not associated in any way with the company) refrain from using the letters P.E.T. and the word PET in the names of our products. In deference to Commodore's sensitivity, we have blacked out the word PET on the banner of this issue and in all references to the PAPER throughout the publication. We're searching for a new name.

If any of our readers has a neat name in mind, we'd like to know about it. If we use a name sent to us by a reader, that reader will receive a Software Shelf program of his/her choice as a token of our appreciation.

-- The Editors

P E T P A R A D E

As requested, subscribers have sent in a variety of questions - even before our first issue came off the press! Due to the necessity of sending each issue to press within two weeks of mailing the previous issue, your letters will be published in the second issue following the one which prompted you to write.

Q: So the PAPER has all the answers! So maybe you can tell me why in the #!@#* I can't get a tutorial tape and instruction book from Commodore. - Dennie Bassham

A: Mostly due to the lack of time. However, we've heard that Commodore does have the tape tutorial in-house now, and that there is an updated version of the introductory manual in the works. The bigger User's Guide will take a while yet. We understand that Commodore is planning to sell the tutorial tape for \$19.95 and that it covers most of the BASIC commands. It's a well-thought-out program and well worth the cost. We recommend that you buy it.

Q: In describing programs for sale or exchange, I do hope that a more complete description of what the program does will be possible. Also, it would be quite helpful to know to what extent the PET graphics are used by the program. -Edward Crossman

A: More complete descriptions begin with this issue, Ed. And we've begun a graphics rating system - from 'non-graphic' to 'excellent graphics'. Thanks for the suggestion.

Q: Does the PAPER have any relationship to the PET User Notes? - Randall Julin

A: Nope.

Q: Enclosed is \$2.00 for the sample issue. If I subscribe, will the \$2.00 be applied to the subscription cost? - Jules Whitehill

A: Absolutely.

Q: I'd be interested in knowing the serial number at which Commodore changed tape units. - Dr. John Dilt

A: Even Commodore doesn't know exactly which PET received the first of the new electronics. It isn't the tape unit that matters - it's the circuitry. However, we have a tendency to think many problems are due less to circuitry differences than to high-speed mass duplication procedures. Which is why we save each of our tapes ourselves instead of sending them out for duplication processing.

Q: Does anyone have information on developing a dual model PERTEC FD200 microfloppy or similar Shugart disk system? I'm in the process of looking into the development, but would prefer to buy rather than develop. - Robert L. Hinckley

which reads the VIA's Port A and stores the data in the BASIC variable D as an unsigned integer between 0 and 255. A side effect of the PEEK is to reset the CA1 flag bit in the Interrupt Flag Register, thereby setting things up for execution of the next WAIT statement.

Thus, to read a whole line of ASCII characters ending with a carriage return (binary 00001101, or 13) into a string variable, you could use the following program segment:

```
10 A$=""
20 FOR I=1 TO 72
30 WAIT 59469,2
40 D=PEEK(59457) AND 127
50 IF D=13 THEN 80
60 A$=A$+CHR$(D)
70 NEXT I
80 ?A$
```

Here, statement 20 simply limits the number of characters read to 72; statement 40 masks the data read to 7 bits to eliminate any parity bit; and statement 60 uses string concatenation to convert the data into a single string. Although the PET's internal character code is essentially ASCII, some character code translations will be needed in most practical applications. This can easily be done with an array in BASIC.

To use the 8 bit port for output, you must first program the data lines to act as outputs with the BASIC statement:

```
POKE 59459,255
```

which sets all bits of the VIA's Data Direction Register A to 1s. Handshaking is considerably less convenient, since only the CB2 line is brought to the edge connector. You can force CB2 to a logic 1 with the BASIC statement:

```
POKE 59468, PEEK(59468) OR 224
```

and reset it to 0 with:

```
POKE 59468, PEEK(59468) AND 31 OR 192
```

HARDWARE FOR YOUR PET

And these people produce and market hardware for the PET: (the "-" in the description means "PET to". For example, "-RS232" means "PET to RS232".)

```
-RS232 Connecticut Microcomputers 150 Pocono Rd., Brookfield, CT 06804
-S-100 Forethought Products Box 386, Coburg, OR 97401
-S-100 H U H Electronics Box 259, Fairfax, CA 94930
-modem
-RS232 The Net Works 5014 Narragansett #6, San Diego, CA 92107
```

USING THE PET'S 8 BIT PARALLEL I/O PORT

by Dan Fylstra, 22 Weitz St., Boston, MA 02134
Reprinted from the PET User Notes by permission of the author

The PET employs the IEEE 488 bus for general purpose interfacing of external devices, however, for "quick and dirty" interfacing problems, it may be simpler and cheaper to use the 8 bit parallel I/O port. This port is capable of handling many common peripherals including an ASCII keyboard, a printer, or a paper tape reader, but only one device can be connected to the port at a time without some external switching logic.

The 8 bit port is actually part of an MOS Technology MCS 6522 Versatile Interface Adapter (VIA). You can get a copy of the VIA data sheet from MOS Technology, 950 Rittenhouse Rd., Norristown, PA 19401, (215) 666-7950. Most of the VIA's features apparently are used for the PET itself, leaving only a 8 bit port and two handshake lines, which are really quite simple to use. This discussion will limit itself to input through the 8 bit port, which I have actually tested with a REACO optical paper tape reader, but the essential information for output through the port will be included.

The new PET user manual briefly describes the 8 bit port edge connector: pins A and N are grounded, pin B is CAL, the input handshake line; pin M is CB2, the output handshake line, and pins C through L are the 8 data lines, with C being the high order (left most) and L the low order bit. When the PET is turned on, the 8 data bits are programmed to act as inputs and CAL is programmed to recognize a negative transition (from 1 to 0). If the handshake or data strobe line on your peripheral device produces a positive transition, you can reprogram CAL with the BASIC statement:

```
POKE 59468, PEEK(59468) OR 1
```

which changes the CAL control bit in the VIA's Peripheral Control Register (address 59468) from 0 to 1.

When a transition occurs on CAL, meaning that data is ready to be read from the data lines, the next to low order bit in the VIA's Interrupt Flag Register (the CAL flag bit) will be set. You can test for this with the BASIC statement:

```
WAIT 59469,2
```

which takes the contents of the Interrupt Flag Register, ANDs it with 2 or binary 00000010, and tests the result, repeating the test until the result is nonzero. (Note that execution of the WAIT statement cannot be interrupted with the RUN/STOP key, so you should have a way of manually creating a transition on CAL when you're testing the interface.)

After execution of the WAIT statement, the data present at the 8 bit port is ready to be read with the BASIC statement:

```
D=PEEK(59457)
```

A: Well, Readers??

Q: We have been barraged with PET mail! We cannot afford to take advantage of all these offers, but... - John Vito

A: BARRAGED??? We'd like to know: by whom? Please share your list of offers with the rest of us so we can find out who has what and where it is! Thanks!

Q: How can I purchase the better version of the Basic ROM - the one without the flaws in handling arrays? - R. L. Sosnowski

A: You can't - right now. Arrays with more than 256 elements will not work correctly in PET. Commodore says they intend to offer ROMs which fix this and several other more minor bugs, but no date has been mentioned.

Q: How about memory expansion? I already have three programs that cannot run within my 8K. - R.L.S.

A: Try chaining (explained in this issue) to reduce program size. Commodore has not made any announcements on memory expansion. At least two firms offer (or will offer) PET to S-100 bus adapters. This allows you to use the readily-available S-100 memory cards to expand your PET. We would appreciate a letter from anyone who has actually done this.

Q: I don't know what an RS-232 printer is, who makes it, or what it costs. Can you tell me?

A: Our apologies to all of you who thought (and why not?) that RS-232 was the model number of a specific printer. RS-232 is actually the designation for an industry-standard for serial communication. Thus, theoretically, any RS-232 device looks the same to a computer output circuit. Unfortunately, the standard only refers to the voltages which have to be passed from the computer to the RS-232 device and back to the computer. It doesn't say anything about how fast the data is transmitted or what computer codes should be used to transmit data (although ASCII is usually assumed). Almost all modern data terminals are available with an RS-232 interface. You would have to consult the documentation for any particular terminal to see if it has this kind of interface. The vagueness of this interface 'standard' was one of the reasons the designers of PET chose the IEEE bus (which is much more specifically defined) for all peripheral communication.

Q: Where can I get copies of the Blackjack and Lunar Lander programs Commodore has used in demonstrating the PET at computer shows? - Mark Forbes

A: Both of these programs will be among the first group of programs offered by Commodore. You should get a notice from Commodore in the next month giving their prices and ordering information.

PET letters

Dear Sir:

This is to inform you that the North Orange County Computer Club has recently organized a PET User's Group for PET owners in the club. We plan on operating a PET Service Center for Orange County as soon as we obtain the necessary authorization from Commodore. This will allow us access to the information necessary to run the built in diagnostic routines in the PET. Our biggest concern right now is to try to yank out of Commodore as much technical information about the PET as we possibly can.

At the present, we have five members who actually have PET's in their homes. We hope to encourage more members of the club to buy PET's but only if we can pry some more information out of Commodore. Many of the members are technically competent and want to be able to service the PET themselves, as well as expand the PET and write their own machine language programs for the PET.

We will be happy to keep you informed of our activities and hope that between all the PET groups we can put together a decent hardware and software manual that will provide PET owner's with the information they are so hungry for. As I have been selected to head up the user's group, you may address any correspondence to me at the address below.

Thank you

David Smith
 PET User's Group
 North Orange County Computer Club
 3030 Topax, Apt. A
 Fullerton, Ca. 92631

Dear Sirs:

The program enclosed is a math practice program for individuals of all ages. I ask you to try it. If you like it and decide to sell it, I would greatly appreciate it. I am 16 years old, saved up enough money to buy the PET and I enjoy it a great deal. I have noticed that my PET is one of the older models with a tape-recording problem. If, when you try my program, it will not load properly, then that is the problem. If you would like to see any changes in the program, please contact me and I will do whatever is necessary.

Sincerely,
 Dan L. Davidson, San Raphael, CA

Thank you, Dan. Your tape loaded perfectly. - Editor

hex, find the value in a box in the table, and follow the column up to the top to find the second digit and the row to the right to find the first digit. Locate decimal 163; the hex equivalent is A3. And the decimal equivalent of EC is 236. Good luck!

HEX TO DECIMAL CONVERSION

		Second Hex Digit																
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
F i r s t D i g i t	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0
	1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1
	2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	2
	3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	3
	4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	4
	5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	5
	6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	6
	7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	7
	8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	8
	9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	9
	A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	A
	B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	B
	C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	C
	D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	D
	E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	E
	F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	F

INTRO TO BASIC

Last time we talked about line numbers and the PRINT, READ, DATA, RESTORE, IF and GOTO keywords. This time we'll discuss PRINT in more detail. First we have to have some definitions so we'll all be talking about the same things when we use similar terms.

Data is variable. It can change from run to run. It can even be changed during a run. PET uses letters of the alphabet and limited combinations of letters and numbers as variable names. While this is a tongue twisting phrase, it means "data names", which is not only easier to say, but easier to remember. The variable names allow PET to store data somewhere in its memory while it's working on other things or while it's processing the data itself. When you type:

```
10 DATA 10,20,30
20 READ X
30 PRINT X
40 IF X=30 GOTO 60
50 GOTO 20
60 PRINT"DONE"
```

"X" is the variable name. Each time PET reads a value from the DATA statement, it stores the value in memory and calls it X. Later, when you tell PET to do something with the value it read, it hunts around in memory for the value called X and then does whatever you instructed as soon as it finds the value. Obviously only one item of data can be called "X" at one time, or PET would have no way of knowing which X you are asking for. So, when PET reads a new data item that you have specified to be called "X", PET destroys the old value and calls the new value X. Valid names for variable data are restricted to the following characters:

A - Z Any letter of the alphabet
A0 - Z9 Any letter of the alphabet followed by any digit
AA - ZZ Any letter of the alphabet followed by any other letter.

PET can store three types of data: integers, floating point (decimal numbers) and string (non-numeric) data. You tell PET which type of data you are using by giving the data a name which has an appropriate identifier appended to the end of it. Floating point numbers do not require an identifier, and PET will assume, if you don't specify otherwise, that all data is of the floating point type. The identifier for integer numbers is the % sign, and the identifier for string data is the \$ sign. If you tell PET to read a number and to store that number in a variable identified with a \$ as alphabetic, PET will complain: ?TYPE MISMATCH ERROR and will refuse to process the data until you correct the problem.

Let's look again at valid variable names, and identify the three types of data:

A% - Z%	A0% - Z9%	AA% - ZZ%	Integer variable names
A\$ - Z\$	A0\$ - Z9\$	AA\$ - ZZ\$	String variable names
A - Z	A0 - Z9	AA - ZZ	Floating point variable names

SOFTWARE SHELF: Please send the following programs. Payment in full is enclosed.

Name _____ Cash, check or money order \$ _____

Street or Box # _____ MC/VISA (circle 1) # _____

City, State, Zip _____ MC interbank # _____ Exp date _____

A signature is required for all credit card orders: I authorize this purchase on my _____ credit card.

Signed: _____

PET PROSE: Please publish my name as an author of specific application software. I enclose \$25 for publication in the next 10 issues of the PET PAPER.

Name _____ Application fields: _____

Street or Box # _____

City, State, Zip _____

Cash, check, money order \$ _____ MC/VISA (circle 1) # _____

MC Interbank # _____ Exp date _____ Signature _____

This is the last issue in which ADDRESS FILE will appear in our Flea Market Exchange Library. Beginning with the May issue, ADDRESS FILE will be on the Software Shelf and will sell for \$10.00.

We have a new 5-game program tape containing HAIKU, HANGMAN, HI-LO, COMPUTER HAIKU, AND CHARACTER (which isn't really a game, but which is fun to play with anyway). The package sells for \$10.00 without any exchange tape (it costs more because of the extra time involved in saving more than one program on a tape).

P E T ' S H E X

If you plan to do any machine language coding, you'll need to convert all the hex values in your program to decimals before PET can use them. We have a program in our Flea Market exchange to do that for you, but here's a table (complete with instructions on how to use it) to help out. If you want to do your own calculations, here's how to do that:

To convert from Hex to decimal, you must first give each hex digit a 'place' value, from right to left, like this

F	E	0	A	These are the Hex values (we'll call them H as we convert them to decimal values)
3	2	1	0	These are the 'place' values (we'll call them P)

$D = H * 16 \quad P + D$

Step 1.	A=10	D=10*16	0	D=10*1+0	D=10
Step 2.	O=0	D=0*16	1+10	D=0+10	D=10
Step 3.	E=14	D=14*16	2+10	D=3584+10	D=3594
Step 4.	F=15	D=15*16	3+3594	D=61440+3594	D=65034

So the final decimal value of Hex FE0A is 65034.

To convert from decimal to Hex is easier:

Let A = memory location in decimal
Let B = INT(A/256) This is the high order byte.
Let C = B*256
Let D = A-C This is the low order byte.

Then concatenate B and D to get the right Hex value. An example:

Let A = 32800	
Let B = INT (32800/256)	B will equal 128, or Hex 80
Let C = B*256	C will equal 32768
Let D = A-C	D will equal 32, or Hex 20

Now concatenate B and D to get 8020, the Hex equivalent of decimal 32800.

It's easier to use the table if your decimal value is less than 255. Just follow the right or left hand column down until you find the first hex digit in your value, then follow the row until you come to the intersection of the second hex digit. The number in the square is the decimal equivalent of the hex value you're looking up. On the other hand, if you have a decimal value and you want to convert it to

Editor's note: This letter was so interesting that we wanted to share it with you. Please write to Tom Ryan directly if you are interested in his concepts or can help him.

Dear Sirs,

I am writing you about the possibility of obtaining as much information and software possible for the Commodore PET. We have just been donated such a computer by the Commodore people, and we would like to start a users group here also.

The Little Singer School project is for Navajo children, and so we would be using the PET for instruction and bilingual education as well as to handle the many solar and wind equations and other engineering applications inherent in our appropriate technology work. I have developed a complete line of software already for computing solar efficiency, top losses, reflection, solar angle for any day and time, and so forth, which I could make available to other PET users in return for help in other areas.

Little Singer School is totally energy self sufficient, using efficient structural designs, solar heating, and wind power for electricity. All of this installation and fabrication was accomplished by the Navajo community of Birdsprings using my designs and supervision. We need to also monitor the energy systems at the school and would like to use the PET for this with appropriate IEEE interfaces and data logger. Anyone who could advise us about microprocessor controls and interfacing the solar and wind monitors would be most appreciated. We also would like good graphics capability if possible - for the children, and for visual implementation of data.

Thank you for your interest in promoting the computer arts.

Sincerely,

Tom Ryan, Project Director
Navajo Technology Project
P. O. Drawer 700
Winslow, Arizona 86047
(602) 289-4025

PET PAPER SOFTWARE SHELF - SOFTWARE SALES

Have you written something significant or original that you'd like to share with others, but which you believe requires more than average expertise to write? Send us your tape! If we agree with you, we'll send you a contract specifying royalty terms (normally 20% of the retail price). As soon as we receive a signed copy of the contract, we'll offer your program on the SOFTWARE SHELF. All royalties are paid quarterly.

So you see a program on our SOFTWARE SHELF that you'd like? Send us the purchase price and we'll send you the tape. All our non-game SOFTWARE SHELF programs that require explanation come with written documentation.

HOME DATA RETRIEVAL Create unlimited numbers of data files of unlimited length. Each record in each file consists of up to 256 characters in as many fields as required; the first 3 fields are keyed for selection and sorting purposes. Select up to 50 records at a time from any combination of existing data files and sort them by any of the 3 key fields in either ascending or descending order. Save selected records on either ascending or descending order. Save selected records on either of the cassette units or on an external IEEE device. Written documentation shipped separately. Non-graphic. \$10.00.

STAT Statistical analysis of up to 100 data points. The user enters the data into DATA statements before program execution. Basic statistics, linear regression, moving averages, transformations, and plotting. Good graphics. \$20.00

SPACE-TALK/SPACE-FIGHT A two program set in which SPACE-TALK is the instructions and SPACE-FIGHT is a two-player game. Each player controls a space ship which moves around on the screen trying to "shoot down" the other. A well-executed and fascinating game, full of tension and laughs. Excellent graphics. \$10.00

DEFLECTION A fast-paced action game for one player. It may seem easy to hit the targets, but your control over the missile is dependent on where - and how fast - you place the deflection shields. Excellent graphics. \$10.00

BLOCKADE An adaptation of the popular arcade game for two players. Each player's piece is in constant motion, leaving a trail behind. Don't bump into the wall or either player's trail. Fast and fun with excellent graphics. \$10.00

CHASE You are trapped in a room with electrified obstacles and robot guardians trying to grab you. Elude the robots and get them to run into the obstacles. You control the speed of the robots and their intelligence in avoiding obstacles. All movement in real time. Excellent graphics. Unlike any other version available. \$10.00.

And let's discuss each type of data very briefly. Integer data consists only of whole numbers like 1, 23, 876, -567, and -4. You must never use a decimal point when you identify data using an integer variable name. Floating point data consists of all the other numbers - like 1.25, 43.006, 8.00, -45.0, and -.07. You must not attempt to use integer or string names when using data containing a decimal point (there's an exception to this rule and we'll get to it in a little while). String data consists of all the characters on the keyboard which are non-numeric. Letters of the alphabet, graphics, special symbols like #, @, and *, and the characters representing the cursor control functions. String data must almost always be enclosed in quotation marks so PET will know you aren't trying to simply make up an extra long variable name. Numbers can be included in the string, if they are enclosed in quotes, too. (This is the exception mentioned earlier.) Here are some examples of data and their data types:

A	floating point	12.0082	floating point
"HI"	string	"12.0082"	string
42	integer	12	integer

Notice that sometimes the only difference between a string value and a floating point value is the quotation marks. It is very important to remember that there is a difference - because PET gives string data very special treatment. It prints strings exactly the way you type them in, with no deviations whatever. Try these two statements, and see what happens:

```
10 PRINT .0001
20 PRINT ".0001"
```

Now we know what variable names are (data names!) and what integers, strings, and floating point numbers are. So we can go on to bigger and better things.

PRINT You already know how to use the PRINT keyword to display a number on the screen. You can print several numbers on the screen - and on the same line of print on the screen - by separating the print items with a comma. The comma permits PET to print each item in the next available pre-selected print position and suppress the line-feed and carriage return. The pre-selected print positions are columns 0, 10, 20, 30, etc. (notice: increments of 10). You might like to note that PET prints all numbers in this format:

sign-space	number	Ⓜ
------------	--------	---

If the number is negative, the minus sign will appear in the sign-space and the first digit of the number will appear in the next column. Try typing and RUNNING this program:

```
10 PRINT 12,24,36
20 PRINT -12,-24,-36
```

You'll be able to see the effect of the sign-space as well as the effect of the comma.

You can use a semi-colon to change the spacing of the printed items, too. It will look like the numbers are printed two spaces apart, but they are actually printed immediately adjacent to each other. One of the apparent spaces is occupied by the cursor-right of the first number, and the other apparent space is occupied by the sign-space of the next number.

```
10 PRINT -12;12
20 PRINT 12;-12
```

PET will display string data separated by commas or semi-colons, too. Again, the print items are either begun in the next available pre-selected print position (if you separate them with a comma) or immediately adjacent to each other (if you separate them with a semi-colon). Because string data is usually alphabetic, you may want to remember to add an extra space at the end of each string to provide a space between it and the next string, if you are using the semi-colon as the separator.

```
10 PRINT "HELLO";"SAM"
```

or

```
10 PRINT "HELLO";" SAM"
```

It is not necessary to use the semi-colon, however. PET will always default to semi-colon spacing if you don't specifically use a comma. So not only do you save a keystroke by not using the semi-colon, you also save a space in memory. Just remember that the items are printed immediately adjacent to each other, and that numeric data only looks like it's separated by two spaces.

Another neat feature of the PRINT statement is the TAB(X) function. You can decide exactly which column you want something to be printed in, and, using the TAB(X) keyword, you can tell PET to print it there. The X (as you've probably guessed) represents a column number.

```
10 PRINT TAB(15);"HELLO";
20 PRINT TAB(27);"SAM"
```

The results of the above code may surprise you. The semi-colon following the string "HELLO" in line 10 caused the line feed and carriage return to be suppressed, so that the string "SAM" in line 20 was printed on the same screen line.

Notice that a semi-colon was used between the function and the string item (between TAB(15) and "HELLO"). While in this kind of PRINT statement as in many others, the semi-colon is not necessary, it is important to remember to not use a comma. If you do use a comma as a separator after TAB(X), PET will be confused: Do you want it to print the item in the specified column or do you want it to print the

ROAD RALLY Previously listed as ROAD RACE, this is a fun version of a car rally. Race your selected vehicle - but don't run out of gas or get hit by a bus. Moderately good graphics.

CODES Based on the board game "Mastermind". PET is the encoder and the user is decoder. Best version of the game we've seen so far. Non-graphic.

FLEA MARKET: Please send the following program tapes. I enclose \$_____ in full payment or _____ exchange programs; one for each program I select.

Name _____ Cash, check or money order \$ _____

Street or Box # _____ MC/VISA (Circle 1) # _____

City, State, Zip _____ MC interbank # _____ Exp. _____

Credit cards accepted on orders over \$10.00. Signature required on all credit card orders: I authorize this purchase on my _____ card.

Signed: _____

SUBSCRIPTIONS, ADVERTISING,
and Other Money Matters

The PET PAPER is published ten times per year. Each ten issues constitutes a volume, and subscriptions are for one volume only. Your subscription will not span two volumes, so if you subscribe at issue #8, volume 1, you'll still receive all ten issues of that volume.

You may, however, purchase a copy of each issue separately for \$2 each.

If, during any publication year, you wish to buy back issues, send us \$15 per volume (or \$2 per issue) and we'll ship your order ASAP.

Subscription rate increases will not occur during any given volume, and subscribers will be notified if the rates are being increased for the next volume.

Advertising rates are:*

1/4 Page -- \$ 50

1/2 Page -- 90

1 Page -- 150

*For other rates,
please write.

Payable with the camera-ready copy for the ad. Ads not accompanied by payment will not be published. If we misprint an ad, the next issue will carry the same ad at no cost, provided the advertiser notifies us within 15 days of the mailing date of the issue in which the error occurred.

PET PAPER FLEA MARKET - SOFTWARE EXCHANGE LIBRARY

Exchange: Send us a program - we'll send you a coupon good for any program in the Flea Market. Return the coupon and \$2 (for duplication and postage charges) and we'll send you the program you specify on the coupon.

Sales: Send us \$5 and we'll send you the program you specify.

HEX/DEC Convert numbers from hexadecimal to decimal or from decimal to hex. Non-graphic.

USEFUL ROUTINES Nine programs designed to help you avoid re-inventing the wheel: FRAME * GET* X\$, Y\$ FOR CURSOR CONTROL * CURSOR DEMO * SORT * SORT DEMO * FORMS * FORMS DEMO * NUMBER EDIT. Good graphics.

ADDRESS FILE Create, sort, edit, and save up to 50 names and addresses at once. Named data files. Non-graphic.

STATES & CAPITOLS Well-executed drill on the 50 states and their capitols. Makes learning fun rather than tedious. Non-graphic.

ADDITION GAME Ten timed skill-levels permit self-paced learning. Non-graphic.

HAIKU Traditional Haiku poetry created by the PET. Non-graphic.

COMPUTER HAIKU Haiku style poetry using computer terminology. Non-graphic. Funny.

HANGMAN For two players. One player selects a word and the other guesses the letters and the word. PET draws the hangman. Limited graphics.

HI-LO The user selects a range of numbers, then either the PET or the user selects a secret number from that range. The other tries to guess the secret number. PET catches cheaters. Non-graphic.

OTHELLO FOR ONE Board display and user-selected tokens. PET plays a good corner game in this version, as well as keeping a running score. Limited graphics.

OTHELLO FOR TWO Same game as OTHELLO FOR ONE, but PET's role is merely score-keeper. Limited graphics.

CHARACTER Prints PET's equivalent of ASCII code and the characters represented. Non-graphic.

SLOT-MACHINE Real Las Vegas style slots! PET's unique graphics permit the illusion of spinning wheels. In this version, however, you keep your money in your pocket. Good graphics.

item in the pre-selected print position as indicated by the comma? Oh, my! Such decisions! So PET gripes ?SYNTAX ERROR at you and sits and sulks until you make up your mind what you want and insert a semi-colon in place of the comma - or just delete the comma - or delete the TAB(X) part of the statement and the comma.

The PRINT statement has yet another very handy feature: the SPC(X) function. SPC(X) causes PET to print a specified number of SPaCes to the right of the previous print item before printing the next print item:

```
10 PRINT X;SPC(8);Y
```

PET will print the value stored by the name X (beginning with the sign-space in column 0 and the first digit of the number in column 1). After completing that, PET will print eight spaces. Then it will print the value stored in B. (The terms "stored in" and "stored by the name of", "value in" and "value of" all refer to the confusion programmers get themselves into because they haven't made up their minds to clearly differentiate between the terms "variable" and "variable name". To make things neat and simple for us, we'll refer to data as being stored in a variable name; and we'll refer to the value of a variable name when we mean the data stored in it, i.e. "Data stored in A" and "the value of A".)

And one final bit of good news: because the most frequently used keyword in most BASIC programs is the keyword PRINT, the PET's creators (bless their infinite wisdom!) provided a token for the user to type instead of having to type PRINT every time the word is needed:

```
10 ?"HELLO"  
20 ?A
```

PET recognizes the question mark as the PRINT keyword, and when you LIST your program, PET will dutifully display the word PRINT every place you typed a question mark. Just think of the hundreds of saved keystrokes!

We intended to cover the INPUT statement, more about the DATA statement, and a bit about the FOR and NEXT keywords, but since it turned out that there was so much to say about PRINT, those topics will have to wait until another time. Patience, people! We'll get there yet!

If you're in the middle of a program and accidentally press the Ⓢ, you fall out of the program and PET proudly announces READY. You can get back into your program at the same place you fell out by typing CONT Ⓢ (for CONTINUE).

STANDARD SYMBOLS FOR PET FUNCTIONS

People around the country are crying for a standardization of symbols to depict the PET special function keys, but no one has done anything about it. Here are the symbols we plan to use in THE PAPER. We'll reprint them in every issue so you'll always have them handy.

Ⓜ	Cursor HOME
Ⓒ	CLEAR Screen
Ⓡ	Cursor RIGHT
Ⓛ	Cursor LEFT
Ⓤ	Cursor UP
Ⓣ	Cursor DOWN
ⓕ	Reverse FIELD on
Ⓞ	Reverse field OFF
Ⓔ	Carriage Return
↑	Shift On
↓	Shift Off
/	Delete
Ⓜ	INSERT
Ⓢ	Space
*	Stop

Here's a sample program to show you how the code looks:

```
10 PRINT "Ⓜ ";           Send the cursor HOME
20 FOR I=1 TO 10
30 PRINT "Ⓣ "           Print "cursor down" 10 times
40 NEXT I
50 FOR I=1 TO 10
60 PRINT "Ⓡ "           Print "cursor right" 10 times
70 NEXT I
80 PRINT "HI!"           Print a message
```

We chose these symbols because they are easily drawn, typesettable, and some of them are mnemonic. If you have a better idea, let us know about it!

If you feel that you absolutely must write programs or subroutines in machine language, this assembler will be a great help. But be very sure that BASIC can't do what you want to do before you invest a lot of time in learning machine language programming.

PET PROSE

There are many people who want specialized application programs to use on their PETs, but who are not knowledgeable enough to write them. These people are willing to pay for the software they want. If you can write significant software in some specific field (i.e., education, business, engineering, etc.) and are willing to do so, we can help you find the people who are willing to pay you to write programs for them. Send us \$25 and we'll print your name, address, phone number, and your field of expertise, in the next 10 issues of the PAPER.

Wes Fasnacht (215) 436-2886
West Chester State College
West Chester, PA 19380

Programming and Systems Development

SOFTWARE FOR YOUR PET

The following people produce and market software for the PET:

AB Computers	Box 104,	Perkasie,	PA 18944
Creative Software	Box 4030,	Mountain View,	CA 94040
Don Alan Enterprises	Box 401,	Marlton,	NJ 08053
Mind's Eye Personal Software	Box 354,	Palo Alto,	CA 94301
Personal Software	Box 136,	Cambridge,	MA 02138
Silver State Enterprises	Box 27111,	Lakewood,	CO 80227
Softbyte	315 Dominion Dr.,	Newport News,	VA 23602

USER GROUP NOTES

ACGNJ (the Amateur Computer Group of New Jersey) has a PET User subgroup that meets on the 4th Friday of every month at the Union County Technical Institute on Terrill Road in Scotch Plains, NJ. For information (like how to get there and when) call Ike Rodman (coordinator) at (201) 671-3507 or Roy O'Brien (software librarian) at (201) 246-1000.

Software Review

6502 Assembler in BASIC for the PET

One of the most persistent inquiries about the PET from experienced microcomputer enthusiasts is whether there is an assembler available. The answer is now yes. Dan Fylstra of Personal Software (P.O. Box 136, Cambridge, MA 02138) is offering a package of programs which includes both one-pass and two-pass assemblers, an editor and an execution-time monitor. Price for the set of four programs plus 27 pages of documentation is \$25.

When Commodore has spent so much time and effort to provide BASIC for the PET, why would you want to write programs in assembly language, a far more difficult programming task? Speed is almost always the answer. One of the first programs I wrote for my PET was an implementation of Conway's game of LIFE, published several years ago in Scientific American. The program requires a large number of logical decisions (typically several thousand) for each new 'generation' of life displayed on the screen. Written in BASIC, the calculations for a new generation required about a minute. A similar program written in machine language requires about a second.

Since assembly language programs are so difficult to write, the best approach is to write most of your program in BASIC and use machine language subroutines for the few areas that need the speed. Fylstra's assembler gives you that capability.

Some further observations:

1. Because it is written in BASIC, the assembler is slow - one to three seconds per line of code assembled.
2. The documentation is good, but if you haven't written assembler programs before you will need the MOS Technology 6502 Programming Manual and a good book on assembly language programming.
3. Because of memory limitations, the text editor is limited, and all source statements must be stored on tape for later processing by the assembler.
4. A sample assembly language program is included on the tape, giving you an example to work with.
5. Unlike several assemblers available for other 6502-based systems, Fylstra's assembler does use the standard MOS Technology addressing notation conventions.
6. There are two severe restrictions on the size of programs which may be written using this assembler. First, no more than 198 bytes of machine code can be generated. Second, no more than ten symbols can be used in your source program. As Fylstra points out, the only solution is to add more memory to your system.
7. The documentation contains two pages of 'Machine Language Programming Hints' which are quite useful.

Save \$\$\$!

NOW!

ONLY \$15./YEAR

YES! I Want the  **PAPER** for a whole YEAR! I enclose \$15.⁰⁰ for **10** issues,

Name _____

Address _____

City, State, Zip _____

cash mc# _____

check VISA _____

money order exp. date _____

Master Charge Interbank # _____

Mail to:  **PAPER**

P. O. BOX 43

AUDUBON, PA. 19407

CHAINING

or How To Run Programs Too Large For Your Memory
by Dan Fylstra, 22 Weitz St., Boston, MA 02134
Reprinted from the PET User Notes by permission of the author

What can you do when your program is too long for your 4K or 8K of user memory? One alternative is to read text messages or initial data from a cassette data file rather than, say, DATA statements. If the program text itself is very long, however, you may have to resort to chaining, or having one program load another program from cassette to continue execution. You can do this on the PET and preserve data values in memory from one cassette load to another, provided that you watch out for a few restrictions. The basic idea is simple, however: when you reach the point in the program logic where the next cassette load is needed, just include a statement such as:

```
100 LOAD"NEXTPART"
```

After loading NEXTPART, the PET seems to continue execution from the first BASIC statement loaded. All BASIC statements previously in memory are erased, but data values are preserved (if you're careful). You can chain to additional program segments with LOAD statements in the newly loaded program.

The basic restriction you must obey is that the largest program segment must be loaded first in order to preserve data values. This is because variables and arrays are allocated memory space immediately following the BASIC program at the time execution is begun, so loading a longer BASIC program in place of the existing one would wipe out some of the data values.

Another restriction is more subtle, since it depends on the implementation of strings. A string variable such as A\$ is really a pointer to a character string somewhere in memory. Many character strings are actually stored within the BASIC program itself. So if you execute a statement such as

```
10 A$="THIS IS A STRING"
```

the variable A\$ will thereafter point to a constant string "THIS IS A STRING" within the BASIC program. If you load a new BASIC program in place of the old one, "THIS IS A STRING" will be erased, and A\$ will be left pointing at something invalid. To avoid this problem, you can write

```
20 A$=A$
```

which is deliberately implemented so that "THIS IS A STRING" will be copied to a "safe place" (the free string area in the upper part of memory), and A\$ will be left pointing to this copy.

Another point to remember is that any DIM statements should be executed only once; they should not reappear in the newly loaded program segment, or you'll get a REDIM'ED ARRAY ERROR. The following program, which you can run to try out program chaining, includes logic

to avoid re-executing the DIM statement. The first time the program is loaded and run, each A(I) will contain 0 and each A\$(I) will contain a null string, which prints as nothing. After the program is reloaded, however, it should print two columns of the first 20 integers and their squares.

```
10 IF A THEN 40
20 DIM A(20), A$(20)
30 A=-1
40 FOR I=1 TO 20
50 ?A(I),A$(I)
60 A(I)=1:A$(I)=STR$(I*I)
70 NEXT I
80 IF B THEN STOP
90 B=-1
100 LOAD"PROG"
```

You should either SAVE"PROG" twice, or SAVE it once and be prepared to rewind the tape during program execution so that PROG can be found and loaded again. You'll notice that the message PRESS PLAY ON TAPE #1 is displayed if PLAY is not pressed on the cassette drive, but the messages FOUND PROG and LOADING are suppressed. In the above examples, there are no constant strings assigned to variables to worry about; the statement A\$(I)=STR\$(I*I) causes A\$(I) to point to a newly created string such as "1", "4", etc., in the free string area.

FROM THE COMMODORE HIMSELF

We were fortunate enough to catch a glimpse of the new Commodore printer in action this month. The printer handles PET graphics beautifully (see the section of code below) and is really an attractively packaged machine. It is a serial-input unit using 8 1/2 inch wide paper, running at 120 characters per second. Commodore is taking orders (\$595) and delivery is scheduled to begin in June.

The BASIC tutorial tape will soon be available at \$19.95. It's an excellent set of programs written by Ron Lodewyck and is well worth the price. The tape covers almost all the BASIC commands and is interesting even to those of us who are already familiar with BASIC. We recommend that you get it as soon as Commodore releases it.

THESE ITEMS REACHED US TOO LATE FOR TYPESET AND BEFORE WE WENT TO PRESS:

Demo tapes are available from Walter Rychlewski (603 Spruce, Liberty, MO, 64068).

Ten programs demonstrate most of PET's features. Educational and fun.

Cassette interface connectors (CINCH #250-06-90-170) and IEEE-488/User Port interface connectors (CINCH #251-12-90-160) are available from Rico Enterprises, 8J Lakeside Dr.; Ledyard, CT, 06339. Cost is \$2.50 & \$3.50, respectively.