VOLUME 1 ISSUE 4
VOLUME 1 ISSUE 5

commodore

PET USERS CLUB NEWSLETTER

Newsletter Contents

MEMBERSHIP/SUBSCRIPTION

The Charter of the COMMODORE PET USER CLUB is to provide a method of sharing up to date information, applications and programs relating to the PET Computer between the many PET owners and users.

We would like to publish features from PET Users concerning specific applications, interesting discoveries or even bits worthy of sharing. If you would like to contribute to future NEWSLETTERS, please send your article, letter or comments to:

THE EDITOR
COMMODORE U.S. PET USERS' CLUB
COMMODORE BUSINESS MACHINES, INC
3330 SCOTT BLVD.
SANTA CLARA, CALIF. 95050

Editor Notes

Dear PET User Club Readers:

Due to a not so well known correlary to Murphy's law, we have combined Issue 4 and 5 into one JUMBO BONUS NEWSLETTER, containing over 44 pages of useful information. Because of these factors, please excuse the delay in receiving your NEWSLETTER.

In our Data Exchange Section we will continue to answer your questions as presented. COMMODORE NEWS contains product information on our new PETs and Floppy Disk peripheral. Be sure to review in detail Part 3 of the BREAK-EVEN ANALYSIS program to be found in the SOFTWARE Section. Issue 6 will contain the concluding article. The PERIPHERALS AND ATTACHMENT Section features a HAM Radio Interface which can turn your PET into a mini-communication center. Have you ever wanted to concatenate your programs or subroutines? If so, the PROGRAMMING Section contains the code you'll need (with Documentation). We have included information as provided by Len Linsay of the PET GAZETTE on quality tapes and of every programmers major concern, PROTECTING PROGRAMS!

As always, your comments are welcomed in our continuing efforts to mold this NEWSLETTER into YOUR NEWSLETTER.

The EDITOR

P.S. In Issue 3 one error should be recognized:

Page 3, Figure 1 is missing two labels:

LONG
510uS

BYTE
680uS

Data Exchange

IN PURSUING BETTER COMMUNICATION WITH OUR USERS , THIS SECTION WILL COVER ANSWERS TO YOUR INQUIRIES NOT COVERED IN THIS NEWSLETTER.

→ Mr. John F. Garbarino of Mystic, Conn., has two questions for us,

Q. Is there any problem attaching the Single Drive Floppy Disk to the older PETS?

A. To attach either the Dual Floppy (Model 2040) or Single Floppy (Model 2040A has replaced 2041) all that is required is a "Retrofit Kit" consisting of replacement ROMs. The Dealer who supplies you with your Floppy Disk can easily install these ROMs if you bring in your PET.

Q. A small sample program, featured in one of your earliest brochures, was illustrating the PETs ability to Read cassette #1 and then PRINT (copy) on cassette #2. I have not been able to make it work. I did hear that the ST should be (ST) and the 64 should be (64) in the line where it is doing a status check for read errors. I have the second cassette and would like to see a simple program to be able to copy a tape on the second cassette from the first cassette.

A. The ST, you refer to, should be a 64 (EOF) when the copy is finished. Any other value means that the tape is bad. Please see page 80 of your PET USER MANUAL for the other status. Per your request a program for a tape copy follows:

```
 10   OPEN1,1,0 :PRINT"FOUND FILE"
 20   OPEN2,2,2,"TEST FILE"
 90   GET#1,A$ :IF ST<>0 THE 100
 40   PRINT#2,A$; :PRINTA$;
 50   GOTO 30
100   PRINT ST
110   CLOSE 1 :CLOSE 2
```

→ The following question comes from a User in Winter Park, Florida, Mr. Jeffery Lewis Vida.

Q. When I POKE 59411,60, it not only turns off the motor (of cassette) it shuts down the PET completely if I have any type of INPUT or GET afterwards. If a POKE 59411,53 is in the program after the POKE 59411,60 and before an INPUT or GET statement, operation is normal.

A. When you PEEK (59411) you will find a 61. When you POKE 59411 with a 53 you change the 8th Bit of the location. But you Poked a 60 when you were at that location. This not only changed the 8th Bit, but also the 1st. The first Bit is used to set interrupts. You turned off the interrupts so the keyboard stopped working. POKE a 61 instead, this will solve the problem.

→ Our final question comes from Danny Johnson of Hampton, VA.

Q. I am using a read statement in a FOR-NEXT loop that goes from 1 to 680 with a lot of data statements that hold the 680 numbers that I am reading. There is also a POKE in the loop. The purpose of the program is a hockey game with all the surface markings. The data statements hold the character numbers of the position of the symbols in the loop, it prints it out perfectly. But, when I try to get a particular symbol out, they are all wrong. I wrote a little test at the end of the program, and I found out that only the last few hundred symbols are right. Why?

A. There is a bug in the BASIC in your PET. You cannot have an array with more than 256 (0-255) items. The wraparound is caused by the index Byte overflowing. When the Byte hits 255 then it starts back to zero (0).

The problem has been fixed in the new ROMs that you can buy in mid-Summer. If you do need more items than 255 use more than one array. (For more information on New ROMs see Newsletter No. 3)

~~~~~~~~~~ ~~~~~~~~~~~~~ ~~~~~~

Commodore News

2040 FLOPPY DISK

During April, New Product deliveries continued ahead of schedule with the release of our Model 2040 Dual Drive Floppy Disk. If you have already purchased one from your local PET Dealer, please make note that the following items have been enclosed:

PRELIMINARY USERS MANUAL (53 PAGES)
BUSINESS REPLY CARD
WARRANTY FORM
TEST/DEMO DISKETTE

The final Users Manual is now being diligently constructed and will be shipped to those Floppy owners who send in their Business Reply Cards to COMMODORE. Be sure to send us your completed Wanrranty Form also. In Addition to being guaranteed your entitled warranty period, you will be kept informed of anything significant relating to this new product -bugs, attachments, and new Software.

The TEST/DEMO DISKETTE contains :

DUM 3.4
DIAGNOSTIC BOOT
COPY DISK FILES
CHECK DISK
PET DISK
~~~~~~~~~~

2040 FLOPPY DISK

The first program, DUM 3.4 (Disk Utility Maintanence) permits you to perform the 2040 Disk Commands - NEW INITIALIZE, VERIFY, DUPLICATE, COPY, RENAME, SCRATCH by simply responding to PET inquiries. This program will be extremely useful during your "START UP" period while learning the Disk Command Statement Formats.

DIAGNOSTIC BOOT loads a program into the 2040 internal devices. Different combinations of the 2040's three LED's will lite to indicate a specific failure.

The BASIC Program, COPY DISK FILES assists you in selectively copying any or all files from one drive to the other. This program features complete error handling, pattern matching and alphabetizing.

CHECK DISK can be employed to adjust a diskette with a stubborn "Hard" error. During this process, all files are verified with the DOS (Disk Operating System) VERIFY COMMAND. Unused blocks are tested for error and if bad, they will be allocated so that DOS will ignore them during use. The CHECK DISK PROGRAM contained in the first few 2040's delivered had a small bug. A listing of the current version follows. If you wish to make note of the changes, please see your Preliminary Floppy Manual, page A-4.

CHECK DISK VER 1.3

```
1 REM CHECK DISK -- VER 1.3
2 DN=8:REM FLOPPY DEVICE NUMBER
5 DIMT(100):DIMS(100):REM BAD TRACK, SECTOR ARRAY
10 PRINT"■■■"TAB(9)"■CHECK DISK PROGRAM"
20 INPUT"■DRIVE NUMBER TO BE VERIFIED   ■■■";D$
30 OPEN15,DN,15
35 PRINT#15,"V"D$
40 PRINT"■VERIFYING DRIVE "D$
45 N%=RND(TI)*255
50 A$="":FORI=1TO255:A$=A$+CHR$(255AND(I+N%)):NEXT
60 GOSUB900
70 OPEN2,DN,2,"#"
80 PRINT:PRINT#2,A$;
85 T=1:S=0
90 PRINT#15,"B-A:"D$;T;S
100 INPUT#15,EN,EM$,ET,ES
110 IFEN=0THEN130
115 IFET=0THEN200:REM END
120 PRINT#15,"B-A:"D$;ET;ES:T=ET:S=ES
130 PRINT#15,"U2:2,"D$;T;S
134 NB=NB+1:PRINT"NUMBER OF BLOCKS CHECKED:    ■■■"NB
135 PRINT"CHECKING TRACK    ■■■■"T", SECTOR    ■■■■"S"■■"
140 INPUT#15,EN,EM$,ES,ET
150 IF EN=0THEN85
160 T(J)=T:S(J)=S:J=J+1
165 PRINT"■■■BAD BLOCK:■■",T,S
170 GOTO85
200 PRINT#15,"I"D$
210 PRINT"■■■INIT DRIVE "D$:GOSUB900
212 CLOSE2
```

```
 215 IFJ=0THENPRINT"■NO BAD BLOCKS!":END
 217 OPEN2,DN,2,"#"
 218 PRINT"■■BAD BLOCKS■","■TRACK■","■SECTOR■"
 220 FORI=0TOJ-1
 230 PRINT#15,"B-A:";D$,T(I);S(I)
 240 PRINT,,T(I),S(I)
 250 NEXT
 260 PRINT"■"J"BAD BLOCKS HAVE BEEN ALLOCATED"
 270 CLOSE2:END
 900 INPUT#15,EN,EM$,ET,ES
 910 IF EN=0 THEN RETURN
 920 PRINT"■ERROR #"EN,EM$,ET,ES
 930 PRINT#15,"I"D$
READY.
```

The last program, PET DISK, displays a continuous demonstration of the 2040's key features - it may be the easiest way to explain to your friends or associates what this product is about.

In order for our Users to use the Floppy as effectively as possible, future Newsletters will contain further information on this new peripheral. Next months issue will disclose addresses of some important variables and useable subroutines. A program to concatenate BASIC programs will also be provided.

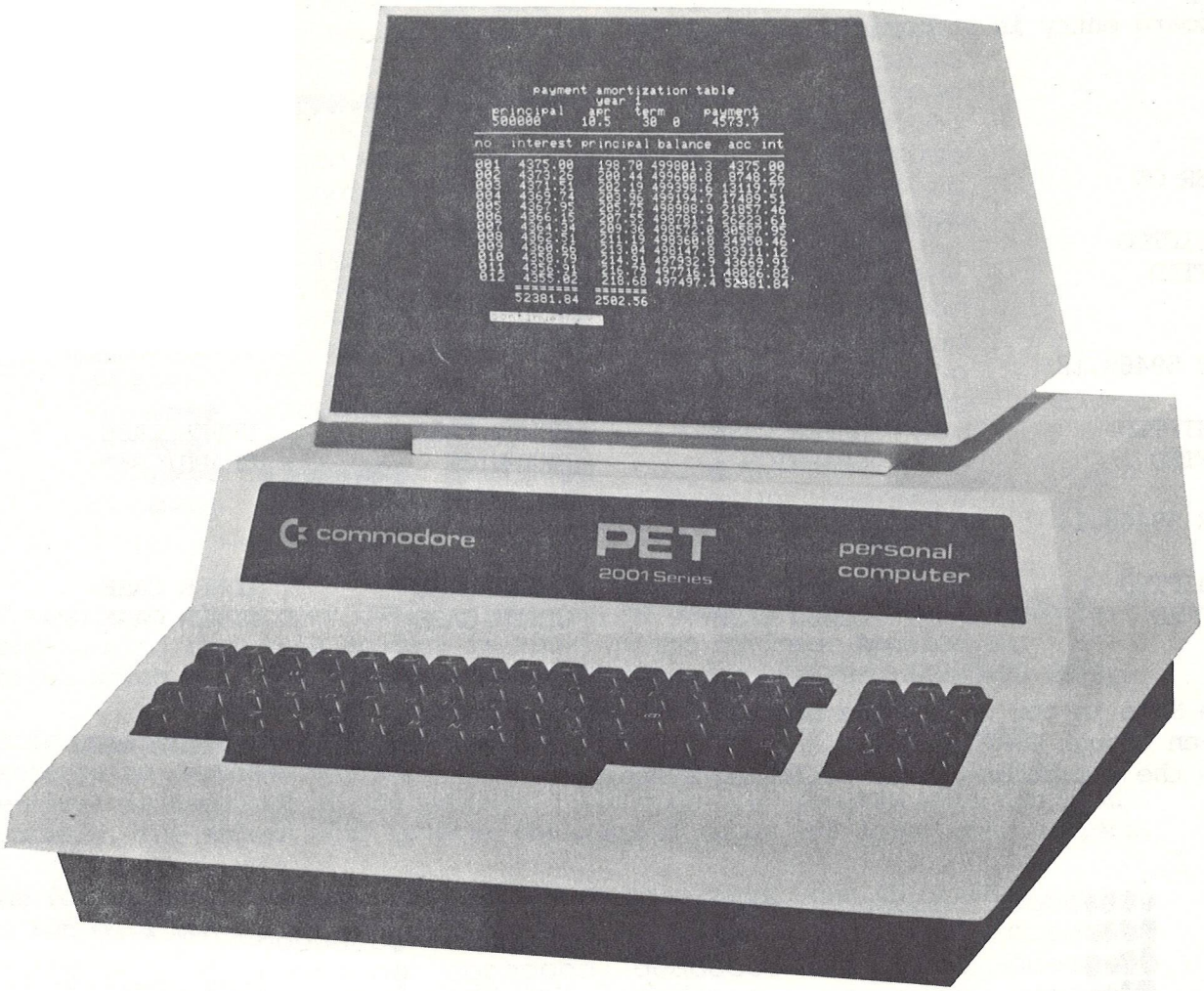
If you have developed an interesting program for the Floppy, or if you have questions concerning the 2040, please send us your thoughts. We will be more than happy to integrate them into future Newsletters.

NEW PETS

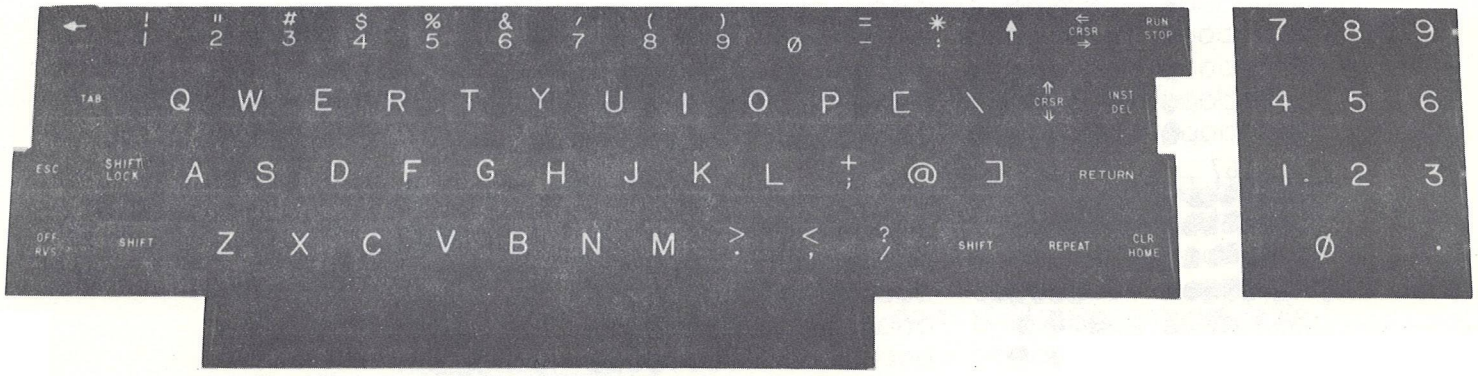
4/8K VERSUS THE 16/32K

Market response to the new GRAPHICS KEYBOARD PET (Model 2001-16N or 2001-32N) has been overwhelming. Withthe BUSINESS KEYBOARD PETS nearing production delivery,our Newsletter will begin to include information pertinent to those new products. There are significant differences between these units and 4K or 8K PET. Once the obvious external differences have been noted (expanded PET keyboard or Standard Typewriter Keyboard - external cassette option) internal operating system differences are a bit more subtle.

The CASSETTE interfacing addressshas been reversed on the new PETS so that the external cassette plug in the back panel is for CASSETTE 1 (rather than 2, as on the 4K or 8K units). If a second cassette is required on the new PETS simply open the PET and attach the Cassette Cable to the left side of the Main PCB.



CBM Model 2001-32B

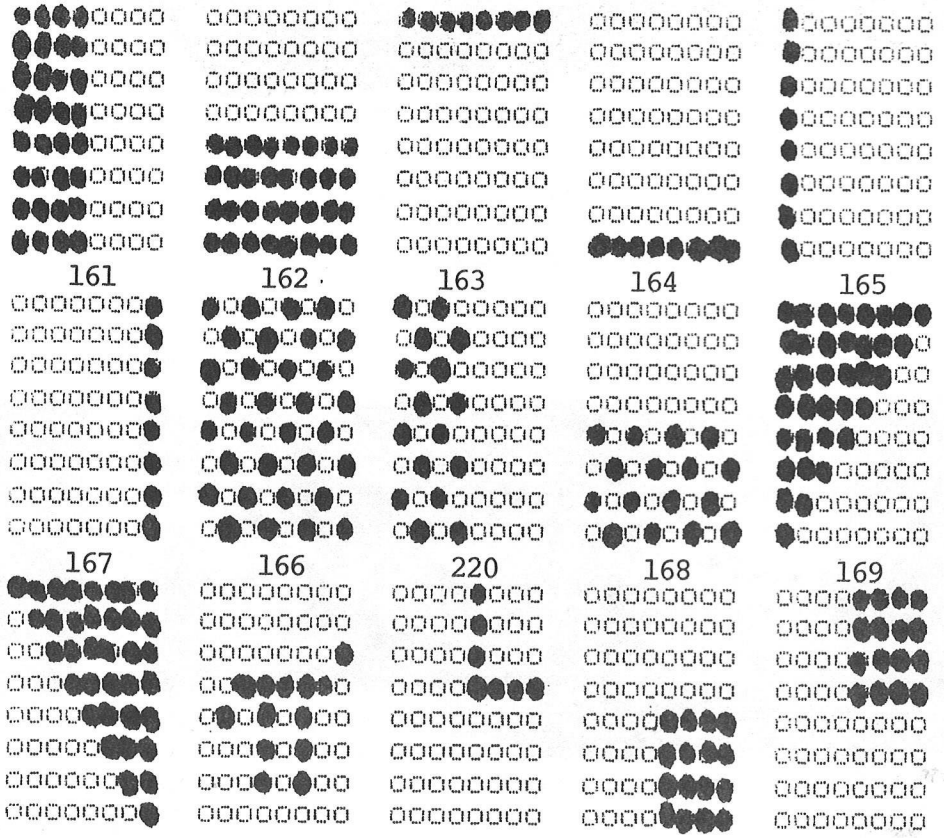


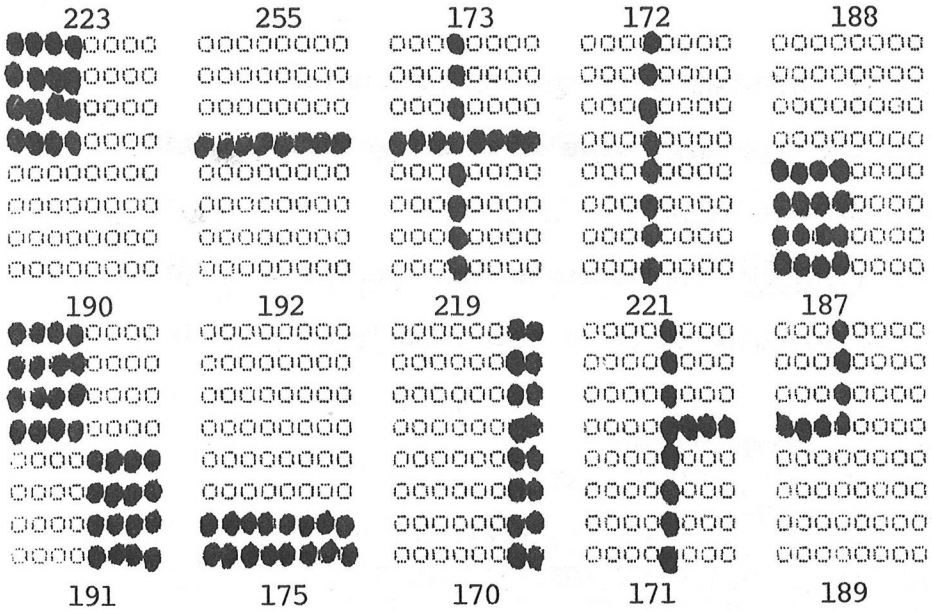
BUSINESS KEYBOARD MODEL 2001-16 or 32

Keyboard entry is summarized below:

	4K/8K	GRAPHIC 16K/32K	BUSINESS* 16K/32K
POWER ON			
UNSHIFTED	UPPER CASE	UPPER CASE	LOWER CASE
SHIFTED	GRAPHICS	GRAPHICS	UPPER CASE
POKE 59468,12			
UNSHIFTED	UPPER CASE	UPPER CASE	UPPER CASE
SHIFTED	GRAPHICS	GRAPHICS	GRAPHICS**
POKE 59468,14			
UNSHIFTED	UPPER CASE	LOWER CASE	LOWER CASE
SHIFTED	LOWER CASE	UPPER CASE	UPPER CASE

*See More in our next ussue on this Business Product.
**Even though GRAPHICS keys are unlabelled, they are accessable in this mode with the exception of the following symbols:





If required, the proper CHR$ (value) can be used to access these symbols. Keep in mind that this Product is aimed at the Business Application Market which will use canned programs developed on GRAPHIC KEYBOARD PETS.

Basic program compatibility between the 8K and GRAPHIC - 16/32K PET, presents no problems unless a POKE 59468,14 (you have used lower case letters) is executed by the program. Although your program will function properly, the screen will reverse the whole of upper and lower case letters.

The following proceedure will enable you to convert your 4K or 8K programs to run on a 16K or 32K GRAPHICS PET if:

1. Only POKE 59468,14 is in your program.
2. Your program does not contain any other Machine Language commands which address memory locations that have been changed. (See revised Memory Map in the PROGRAMMING Section)

To convert your program, just follow these steps:

1. If this is your first time through this procedure, procede with step 3.
2. If you have previously performed steps 3-9, LOAD the program you saved on cassette into PET Memory in the usual manner. Instructions for loading are as follows:

   A. Place the cassette tape in the cassette unit with the desired side facing up.

   B. Push down the REW button on the cassette unit to rewind the tape to the beginning. When the cassette stops rewinding, push the STOP button on the cassette unit.

   C. Clear the screen by holding [SHIFT] down while you press [CLR HOME].

   D. Type: LOAD [RETURN]

   E. Press down the PLAY button on the cassette unit.

   F. After the program has loaded, READY appears on the screen along with the cursor (the winking square.)

G. Press the STOP button on the cassette unit.

H. Rewind the tape and remove it from the cassette drive.

Next, procede with step 10.

3. Type: NEW [RETURN] (this assures that the PET is cleared)

4. Type in the following lines of code exactly as shown below.

```
100 FOR I=0TO99:READA$
110 FORE826+I,VAL(A$):NEXT
120 SY826:STOP
1000 DATA169,4,133,202,169,1,133,201
1010 DATA32,89,3,160,0,196,202,240,13
1020 DATA177,201,170,200,177,201,134
1030 DATA201,133,202,76,66,2,96,160,4
1040 DATA177,201,240,44,201,34,240,4
1050 DATA200,76,91,3,200,177,201,240
1060 DATA31,201,34,240,23,201,65,144
1070 DATA243,201,91,144,8,201,192,144
1080 DATA235,201,219,176,231,73,128
1090 DATA145,201,76,103,3,200,76,91,3
1100 DATA96,255,255,255,255,255,255
1110 DATA255,255,255,255,255,255,255
1120 DATA255,255,255,255,255
```

BE SURE TO PRESS THE [RETURN] KEY AFTER TYPING
THE LAST CHARACTER IN EACH LINE!

5. Clear the screen by holding [SHIFT] down while you press [CLR HOME].

6. TYPE : LIST [RETURN]

7. The lines you just typed in should be displayed on the screen. Check each line carefully for typographical errors. If you find an error, correct it. Refer to Chapter 3, screen editor section of your PET Users Manual Model 2001-16/32 if you require additional information.

8. Repeat steps 5-7 until step 6 yields an exact duplicate of the listing given in step 4. This sequence guarantees that the program has been properly entered into PET Memory.

9. Save and verify this program on a blank cassette tape according to the following directions.

   A. Place the cassette tape in unit with the desired blank side facing up.

   B. Push down the REW button on the cassette unit to rewind the tape to the beginning. When the cassette stops rewinding, push the STOP button on the cassette unit.

C. Clear the screen by holding [SHIFT] down while you press [CLR HOME].

D. Type: SAVE [RETURN]

E. PET will display:
PRESS PLAY & RECORD ON TAPE #1
Do exactly that.

F. PET will then display:
OK
WRITING

As soon as the program is saved, PET will add the word READY to the display and the cursor will return. When this happens push down the STOP key on the cassette unit.

G. Rewind the tape back to beginning, and then push the STOP key on the cassette unit.

H. Clear the screen by holding [SHIFT] down while you press [CLR HOME].

I. Type: VERIFY [RETURN]

J. The PET will then display
PRESS PLAY ON TAPE #1
After you do so, PET will display
OK
SEARCHING

Then after a moment,
FOUND
VERIFYING
will be added. Shortly thereafter the screen will show:
OK
READY
and the cursor will return.
Note: Should the screen display
? VERIFY ERROR
repeat steps B-J again using the same cassette.

K. Rewind the tape and remove it from the cassette deck. Label it 8K to 16/32K Modification Program.

10. Type: RUN [RETURN]

This results in a Machine Language subroutine being stored in the second cassette buffer.

If this step causes your PET to malfunction, it most likely indicates a typographical error in one of the data statements. Turn your PET off and on. Load the program from cassette and procede with step 5 to correct the mistake.

11. After the cursor returns, type:
NEW [RETURN]

12. To load the program you wish to modify, follow the loading instructions A through H in step 2 of this procedure.

13. When the program has finished loading

READY

will be printed on the screen and the cursor will return.

Should a LOAD ERROR be encountered reload the program to be modified as instructed in step 2 of this article.

14. After the program to be modified has been loaded successfully typed:

SYS826 [RETURN]

15. Instantly,

READY

will be displayed and the cursor will return.

16. In order to prevent having to repeat this process next time you want to use the modified program, save the revised program on a blank cassette, as shown in step 9. From then on, all you need to do is load the revised program from cassette and type RUN.

Note: Unless you use the second cassette, or turn your PET off, the Machine Language Modification Program will remain in the second cassette buffer. Therefore you can revise other programs by executing steps 11-16.

The revised Program is now ready to run. While the program is running, your keyboard will work like a typewriter: i.e. upper case.

BUSINESS KEYBOARD VS. GRAPHICS KEYBOARD

Upon scanning the Business Keyboard photograph, you will note the following new keys:

TAB
ESC
REPEAT

TAB and ESC generate legitimate control characters according to ASCII code. These keys may be scanned with the GET command and a value processed in a string variable.

TAB (ASCII CODE 9)
ESC (ASCII CODE 28)

REPEAT has not been implemented at this time and does not correspond to any ASCII code.

The major difference between the business and graphics keyboard is in the location of numerics and punctuation. Numerics are not only located in a pad on the right hand side but in the top row of the keyboard as well. The symbols located in the top row on the graphics keyboard are accessed by shifting the top row numeric keys on the business keyboard.

Cursor and screen editing keys are grouped around the RETURN key on the main keyboard.

[←] now controls scrolling speed as opposed to [OFF RVS] as on the graphics keyboard.

[SHIFT] [RUN STOP] still generates a LOAD RUN sequence but this is now displayed as Lo and Ru respectively.

[SHIFT LOCK] has been added. When this key is down, BASIC will ignore all commands because it recognizes only the ASCII codes which print as lower case.

PRINTERS

In our next issue, we will preview the Model 2022 Tractor Feed Printer and the Model 2023 Friction Feed Printer. As projected in our first issue, Model 2023 deliveries will begin in May.

~~~~~~

Software

NEW SOFTWARE

IN CONTINUING TO BRING YOU AN EXPANSIVE LINE OF SOFTWARE, WE HAVE LISTED NEW OVERSEAS PROGRAMS AND THEIR CURRENT COST. FOR THE DESCRIPTION OF THESE PROGRAMS PLEASE SEE VOLUMN 1 ISSUE 3 OF THE NEWSLETTER.

PROGRAM NAME	PROGRAM NUMBER	PRICE
BOOKS	321040	$19.95
BACKGAMMON	321041	$ 9.95
USER PORT COOKBOOK	321042	$ 9.95
BASIC STATISTICS 1	321044	$24.95
BASIC STATISTICS 2	321045	$14.95
STRATHCLYDE BASIC COURSE	321046	$14.95
STRATHCLYDE BASIC WORKBOOK	321047	$ 5.95
~~~~~~

OTHER OVERSEAS SOFTWARE WE OFFER ARE:

ROCKSTOCK	321019	$29.95
ARDENSTOCK	321023	$24.95
COSTING	321024	$19.95
DATA BASE UTILITY	321025	$24.95
SURVEY ANALYSIS	321026	$14.95
SNARK	321027	$19.95
DISASSEMBLER	321028	$24.95
MACHINE CODE HANDLER	321029	$ 9.95
HEX EDITOR AND LOADER	321030	$ 9.95
LEAST SQUARES	321031	$ 9.95

SOFTWARE REVIEW

SNARK, A SUMMARY

by M. Pipes

The program SNARK originally comes from our overseas office, but is now available in the United States. The program SNARK allows you to write, assemble and execute programs in the language of the SNARK Machine.

It emulates a sixteen-bit computer, with two sixteen bit accumulators and a nine-bit program counter.

The Machine has sixteen instructions:

LOAD	LDA	OR	ORA
STORE	STA	AND	AND
ADD	ADD	JUMP	JMP
SUBTRACT	SUB	BRANCH ZERO	BZE
BRANCH not ZERO	BNZ	BRANCH POSITIVE	BPL
BRANCH NEGATIVE	BM1	RIGHT SHIFT	LRS
NEGATE	NEG	INPUT	INA
OUTPUT	OUT	END	END

There are four addressing schemes, immediate, absolute, indexed by accumulator A, and indexed by accumulator B.

When you write a program you enter an address, and an operation, For Example:

```
Ø  INA
1  OUT
2  END
```

This SNARK Program will print out any number that is input to it. You may list and modify a program at any time.

When the program is assembled SNARK informs you of any errors you may have in the program, and allows you to make required changes.

One of the best features of this program is the capacity to display and single-step the program during execution.

The display is a trace which prints the address, instruction, and the contents of both accumulators.

The single-step allows you to run the program one instruction at a time; a useful debugging tool.

We have found SNARK to be a very useful program for understanding machine-language programs. It's IS (Instruction Set) represents a part of the IS of nearly every computer. A SNARK Program could easily be converted and re-assembled to run on almost any machine allowing easy development of support software. For ordering SNARK, and other overseas programs, please see our Software Section.

FEATURE PROGRAM

BREAK-EVEN ANALYSIS Part 3 by J. Parsons/C. Westfall

In Volume 1, Issue 2 of our Newsletter, Break-Even Analysis was listed on page 10. The program features the 'form' method of entering and displaying data; which provides a quick and easy method of entering and editing data. The programs contain several modules which with slight modifications can serve many functions. The printing of the 'form' was described in Issue 2. Issue 3 discussed the control of data from the form.

Variable name	Value of description of it's function
A	Field row position on screen
B	Field column position on screen
C	Field length
D	Position of the cursor within a field
I	The number of the field
TE	The total number of field
A$	Home plus 25 cursor downs
B$	25 cursor left's
C$	40 spaces

D$	Input from the keyboard
S$	Input from the screen when a field is excited. It equals the number printed on the screen.
W$	Return + cursor up + cursor down + clear home + home + delete + cursor left + cursor right + space + insert
4$	1 2 3 4 5 6 7 8 9

Edits the Input

Line Number	Function
2000	This line clears the keyboard buffer. It continues to 'GET' characters from the buffer until the GET yields a null.
2005	LETT$(A$,A) moves the cursor down the field row position. SPC(B) moves the cursor to the right, B spaces to the field column position. GET#3 stores the character in row A, column B of the screen in T$. The cursor is then moved back one space so it remains in row A, column B. The semicolon surpassed the carriage return for the next PRINT statement.
2010	A- is printed in row A, column B on the screen and the cursor is again moved back to its original position. The FOR NEXT loop scans the keyboard 50 times for input. If a character is received it is stored as D$ and the IF THEN conditions of 2020 and 2030 are not met. In this case the program jumps to line 2040.
2020	However, if no input is received (from the keyboard); this line prints the character T$ (taken from the screen) and a cursor back. The keyboard is again scanned 50 times for input.
2030	If no input is received in 2010 or 2020 the program loops back to line 2010. These three lines simulate a cursor using a (-) rather than the blinking square. In lines 2010 and 2020 the GET command is executed 50 times. 50 is an arbitrary setting the flash rate of the simulated cursor.

2040 D$ (the unput from the keyboard) is compared to each character of the string Y$ i.e. it is checking to see if the input was numeric (a decimal being numeric).

2050 It only executed if D$ is numeric and transfers control to 2110.

2060 Compares D$ with each character in W$ which contains cursor movements, editing functions and the return.

Should D$ not be contained in W$, then T will have a value of 11 when the FOR NEXT loop has been completed.

The ':GOTO 2010' at the end of this line is not needed and should be deleted.

2070 If D$ is one of the first five characters in W$, then this line will jump you to 2200, as the value of T will be less than five.

2080

Value of T	Character	Control jump to line
6	delete	2090
7	cursor left	2100
8	cursor right	2110
9	space	2110
10	insert	2085
11	if D$ is not equal to any character in W$.	2010

2085 Is only executed when insert is typed. Insert clears the character under the cursor and all characters to the right of it. D is set equal to D-1 to keep track of how many spaces remain in that field.

GOSUB4000 - contains the routine which prints the spaces to clear the field.

GOTO2010 begins the simulated cursor routine and allows input from the keyboard.

2090 D$ is set equal to a cursor left; space and cursor left. This simulates a delete by printing a space over the character to be deleted.

2100 This line is executed for both a delete and cursor back. It sets D=D-2 which points two spaces to the left of the cursor position BEFORE delete or cursor back was typed.

If D becomes less than zero, the cursor back or delete moved you out of the field and program control jumps to 2200.

2110 If the cursor remains in the same field it prints the last token taken from the screen (T$ in line 2005), a cursor back, and D$ (character received from keyboard).

Next it GET's T$ from the screen and checks to see if it's at the end of the field. If it's not, control goes to 2010 to allow more input.

2200 Should the field be filled to capacity or your command requires a field change, lines 2200-2230 are executed.

If the last token taken from the screen has been changed it is flashed on the screen.

2210 If the cursor movement moved you out,the left end of the field D would be less than one. So it is reset to one for use in the next lines.

2220 LEFT$(B$,D) tabs back to the beginning of the field being excited. S$ is set equal to a null set.

The FOR NEXT loop contained in lines 2220 and 2230 GET'S each character from the screen in that field as T$ and concatenates them in a string of S$.

RETURN return control to line 50.

~~~~~~
~~~~~~

PUBLICATIONS

In continuing with last months PUBLICATIONS Section we have more books of personal computer information for you. If your local PET dealer or bookstore does not carry the title you're interested in, contact the publisher directly. Also, if you've read a book regarding personal computers and would like to send us your review, we'd be more than happy to look at it!

A CONSUMER'S GUIDE TO PERSONAL COMPUTING AND MICROCOMPUTERS
by Stephen Freiberger and Paul Chew

Unlike most books reviewed in the past, A CONSUMER'S GUIDE TO PERSONAL COMPUTING AND MICROCOMPUTERS, involves reviews and evaluations of 64 microcomputer products from over 50 manufacturers. It also informs you of what to look for when purchasing a microcomputer and its' peripherals. As if this isn't enough to fill one book, it also expands on the fundamental principals and definitions in conjunction with the personal computer.

$7.95
164 pages

Publisher: Hayden Book Co., Inc.
Rochelle Park, New Jersey

BASIC PROGRAMMING FOR BUSINESS
by Irvine H. Forkner

As the title denotes, BASIC PROGRAMMING FOR BUSINESS deals mostly with Business Applications. The development of this book gives the novice an understanding and appreciation of the electronic computer. It would be an excellent source for teachers, as it keys on learning BASIC and not on deciphering the problem and its solution algorithm.

$11.50
237 pages

Publisher: Prentice-Hall, Inc.
Englewood Cliffs
New Jersey 07632

PERSONAL COMPUTING, HARDWARE AND SOFTWARE BASICS
An Electronics Book Series

If you are a serious personal computer buff, this book shows you the contemporary phenomenon now in the microcomputer field. It represents a wealth of technical details and know-how, telling you how to make use of this technology, what methods are available to perform various tasks, what other engineers are doing and just how they are doing it.

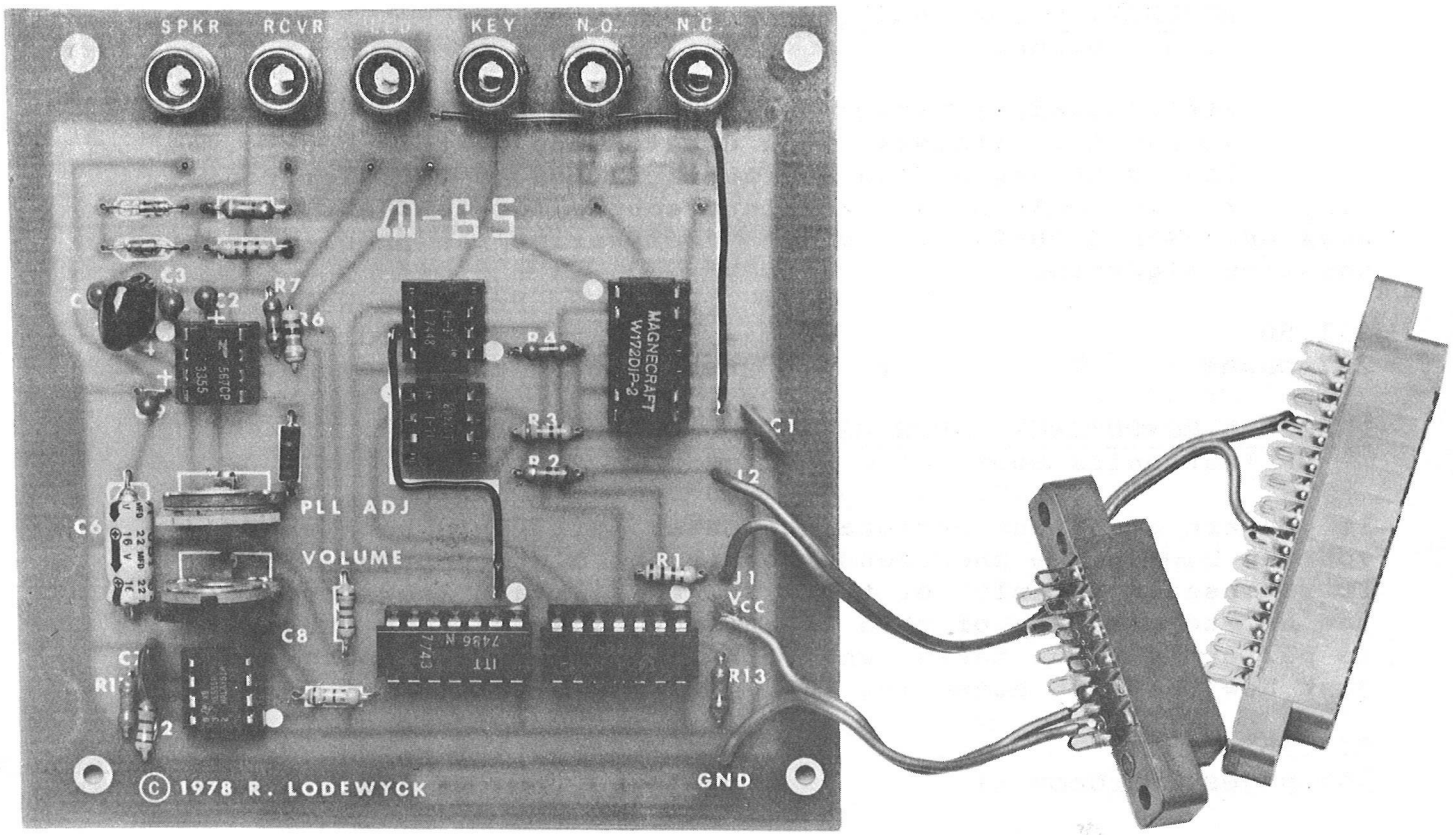
$14.95
266 pages (Hardcover)

Publisher: Mc Graw-Hill
1221 Avenue of the Americans
Suite 26-1 N.Y. N.Y.10020

```
ING THE "INSTANT REPLAY" FEATURE TO SHOW
 YOU HOW IT WORKS. . . ← WHY DONT YOU PUT
ON THE "FOR BEAN LOVERS ONLY" TAPE?  I'L
L ALSO SAVE IT ON CASSETTE HERE.← N6LK
DE N6EE RON IN HUGHSON  ← K K K← CW ID ←
 0230 UTC ←←← ▮

DE N6EE RON IN HUGHSON    K K K  CW ID
 0230 UTC
         SENDING   BAUDOT

ULD DUPLICATE  THOSE
NOISES THEY HAD ON THE BAND THAT WAS TT
Y . .    OF COURSE THE
DISADVANTAGE OF THIS STUFF IS THAT YOU
HAVE TO USE IT TO SEE IF
IT WORKS . . NEVER DID CARE FOR TYPING E
ITHER    BUT GOT FORCED
INTO DINGXXX DOING IT . .  THAT IS THINK
 IN THIS MODE YOU EVENTUALY
GET SB YOU CAN TYPE FAIRLY FAST  , ALSO
 MAKE ERROS FAST TOO
WHERE   YONARE NOT TOO ACCURATE .
```



HAM INTERFACE

Peripherals & Attachments

ADDING THE PET TO A HAM SHACK

According to Ron Lodewyck, President of Macrotronics, the PET makes an ideal HAM communications terminal for several reasons. First, the compact integrated design eliminates exposed wiring cables, thereby reducing susceptibility to RF interference. Second, and most important, the metal cabinet of the PET shields the RF generated by the microprocessor and virtually eliminates any RF "HASH" from being picked up in the receiver. This factor, together with the compact size of the PET, makes it convenient to place the PET within easy reach of the station's radio equipment.

M650 HAM INTERFACE

The M650 is a deluxe RTTY and MORSE system which converts the PET into one of the finest communications terminals ever developed for the amateur radio operator. It provides both send and receive capabilities in MORSE, BAUDOT and ASCII Modes. Let's take a look at some of its features.

RTTY SOFTWARE -- written by Wayne Reindollar

1. Three level split screen display. A "Transmit Buffer" is displayed on top 10 lines. A "Receive Buffer" is displayed on bottom 10 lines. The middle 2 lines display characters as they are being sent over the air. With this system, you can type ahead into the transmit buffer while you are simultaneously receiving an incoming message. You have full edit control of the transmit buffer while in the receive mode, allowing the composed message to be corrected or deleted before it is transmitted. When the incoming station turns it over to you, enter the send mode (one key) -- the transmit buffer will start sending and will be displayed in the center of the screen as it is sent over the air. You may continue typing ahead into the trasmit buffer which will still be displayed on the top of the screen.

2. Word Oriented Editor. The transmit buffer is sent a word at a time. The word will not be transmitted until the space character is entered. Thus, even in the send mode, you will be able to edit the last typed word before it is sent over the air.

3. "Instant Replay". You can send all or part of the received message back to the sending station at the press of a single key!

4. Message Library. This feature allows you to save incoming messages on the PET's built-in cassette and play them back at a later time or date. . Traffic handlers (especially 'MARS' operators) take note!

5. ASCII or BAUDOT. The M650 lets you select ASCII at 110 BAUD or BAUDOT at 60, 67, 75, or 100 wpm. If you have a good general coverage receiver you will be able to copy both amateur and commercial RTTY stations.

6. Auto CW ID. Sends 10 message (keyboard programmable) in MORSE code at the end of the transmission, then automatically transfers to the receive mode.

7. Eight message memories. You can program CQs, test messages (RY, quick brown fox, etc.) and station descriptions for instant one key recall.

8. Automatic time. Sends the present time in UTC at the press of a key.

9. Auto transmitter control. Keys push-to-talk line automatically on send, unkeys on receive. Permits full station control from the keyboard.

10. Auto 10 minute ID. Automatically inserts CW ID message every 10 minutes from start of send mode. The buffer will be preserved and automatically resumes sending where it was interrupted. Keeps you legal during those 'long-winded' QSOs!

11. Auto CR/LF. Automatically sends carriage return and line feed every 72 characters. No need to ever manually type carriage return!

12. Auto Didle. Sends "LTRS" character whenever transmit buffer is empty. May be turned off from the keyboard if desired.

13. Mark/Space Tone Reverse. If the incoming station is "Upside down", you can reverse his signal from the keyboard.

MORSE SOFTWARE -- written by Ron Lodewyck

1. Speed. The speed is entered from the keyboard and can be any number in the range from 1 to 100 words per minute. This setting determines the sending speed and also initializes the receiving speed. The algorithm used on receiving automatically adjusts to incoming speed variations of approximately plus or minus 10 wpm., depending on the quality of the signal. The program will compensate for variations in the dit, dah and space ratios to permit copying most hand sent code. The translated code appears as text on the screen of your PET.

2. Message Memories. You can program any of ten messages of up to 255 characters each at any time for later replay. Common uses would be for CQ's station descriptions, I.D. messages and contest exchanges.

3. Code practice. A Morse code trainer is also included as part of the CW package. You select the speed and whether you want random characters or random five letter words. The PET then sends the code continuously on the built-in sidetone oscillator. The words are ham radio related and are randomly selected from a table. Examples of the words generated

include plate, final, morse, diode, Italy, tubes. Since the words appear in random order, It is not possible to memorize them (as is the case with cassette oriented code practice tapes). Furthermore, the speed is continuously adjustable, permitting fine upward adjustments as code proficiency is gradually developed.

4. Special characters. In addition to the alphabet (a-z), and standard punctuation (-?,.), the CW program can also transmit the following special characters: AR AS SK KN ERROR.

Hardware

The M650 is enclosed in an attractive cabinet and is easy to install with the connectors provided to the PET user port and second cassette port. AN led on the cabinet panel is synchronized with the incoming signal to provide a visual indication of proper tuning. Connection to amateur radio equipment involves simply attaching to the receiver's headphone jack or speaker terminals and the transmitter's CW jack. For RITY, connect to FSK or AFSK keying circuits (the AFSK unit is not included), or connect in series with a local loop using the optically isolated loop module (included). An external terminal unit is recommended for optimal performance, especially on HF amateur and commercial RTTY.

A complete 20 page instruction manual is part of the package. It includes setting up, operation, adjustments, schematics, hints and kinks, trouble shooting and even a program to use the M-650/PET as an alarm clock!

M65 HAM Interface

The M65 is a lower priced version of the M650 HAM interface. It uses exactly the same Morse software, but a less sophisticated RITY program. The hardware interface is sold "bareboard" (not enclosed in a cabinet). It is available in either kit or wired and tested.

Both the M650 and M65 HAM interfaces are available from many PET dealers or by writing direct to:

Macrotronics, Inc.
P.O. Box 747C
Keyes, CA 95328

Applications

In our previous NEWSLETTER we offered $100.00 worth of free Master Library Software to the best "HOUSEHOLD" Application received before May 25, 1979. Next months application, which will be due by June 26, 1979, should deal with the creation of a "DESK TOP CALENDAR".

An explanation for those of you who did not subscribe in time to receive the previous issue of the NEWSLETTER follows. Each month we will be soliciting specific categories of Software. After they have been received they will be evaluated and the "winner" will receive $100.00 worth of free Master Library Software. Or you may wish to compete internally with your local USER Group, or compete with another group in your city.

This by no means should stop you from submitting your Application Program just because it doesn't fall within this month's category -send it in and it could be published on its own merit.

Please send all programs that are competing to:

THE EDITOR
COMMODORE BUSINESS MACHINES
3330 SCOTT BLVD.
SANTA CLARA, CA 95050

Programming

THE APPEND WEDGE (For 8Ks only) by B. Seiler

The APPEND WEDGE is an excellent program to append one BASIC Program to another. This special characteristic allows you to have a set of general purpose subroutines and 'tack' them onto any program. One draw-back though; the line numbers must be in order.

Because the edification of this listing is quite complete, you may wish to read through the listing first, and then commence to programming.

ENTERING THE "APPEND WEDGE"

1. First load the MACHINE LANGUAGE MONITOR Commodore Part No.321000
2. Use the MONITOR to enter the machine code into the second cassette buffer. Hex Ø33A to Ø3FF
3. Use the "X" command to return to BASIC . Type "NEW" (return).
4. Enter the BASIC PROGRAM. Lines 1Ø thru 23Ø are all that are necessary to LOAD, RUN, and SAVE "APPEND WEDGE".
5. Basic Lines 1ØØØ thru 9ØØØ are just the instructions.
6. To SAVE the original copy type RUN 1ØØ. This will save the Machine Language along with the BASIC Program.

```
10 SYS826:NEW
20 REM ************************
30 REM *                      *
40 REM * TO SAVE TYPE RUN 100 *
50 REM *                      *
60 REM ************************
100 POKE241,1
110 POKE247,58:POKE248,3
120 B=PEEK(124):POKE229,B
130 B=PEEK(125):POKE230,B
140 REM *** FIND SAVE NAME ***
150 A$=""
160 A$=STR$(PEEK(150)+256*PEEK(151))
170 A=VAL(A$)
180 A$="APPEND WEDGE"
190 B=PEEK(A):POKE238,B
200 B=PEEK(A+1):POKE249,B
210 B=PEEK(A+2):POKE250,B
220 SYS63153
230 END
1000 REM ******************************
1010 REM *                            *
1020 REM * FOR INSTRUCTIONS RUN 1000 *
1030 REM *                            *
1040 REM ******************************
1050 PRINT"■";
1100 PRINT"       ■ APPEND  WEDGE   COMMAND "
1110 PRINT:PRINT:PRINT
1120 PRINT"    THIS PROGRAM ADDS AN EXTRA COMMAND
1130 PRINT"TO PET BASIC.  THE EXTRA COMMAND IS
1140 PRINT"IS CALLED ■APPEND■.  ■APPEND■ ALLOWS THE
1150 PRINT"USER TO JOIN SEPERATE BASIC PROGRAMS.
1160 PRINT"■APPEND■ COULD BE USED TO LINK TESTED
1170 PRINT"SUBROUTINES TO A NEW MAIN PROGRAM.
1180 PRINT"    THE ■APPEND■ COMMAND IS ADDED TO
1190 PRINT"BASIC BY PLACING A WEDGE IN THE ZERO-
1200 PRINT"PAGE CODE USED TO SCAN ALL LINES.
1210 PRINT"THE WEDGE IS FORCED BY LINE 10 AND THE
1220 PRINT"PROGRAM AREA IS CLEARED BY A NEW.
1230 PRINT"    THE MACHINE CODE FOR ■APPEND■ SITS
1240 PRINT"IN THE SECOND CASSETTE BUFFER.  THIS
1250 PRINT"BUFFER IS FROM 033A HEX TO 0400 HEX OR
1260 PRINT"JUST BEFORE THE BASIC SOURCE.  TO SAVE
1270 PRINT"■APPEND■ THE SECOND CASSETTE BUFFER
1280 PRINT"MUST BE SAVED WITH THE BASIC SOURCE.
1300 GOSUB9000
1320 PRINT"■■■■    THE BASIC LINES 100 TO 230 PERFORM
1330 PRINT"THE TOTAL SAVE.  LINE 100 SETS THE
1340 PRINT"FIRST ADDRESS FOR CASSETTE #1.  LINE
1350 PRINT"110 SETS THE LO AND HI BYTES FOR THE
1360 PRINT"START ADDRESS OF THE SAVE TO 033A HEX.
1370 PRINT"LINES 120 AND 130 SET THE END ADDRESS
1380 PRINT"FOR THE SAVE TO VARTAB.  VARTAB POINTS
1390 PRINT"TO THE END OF BASIC SOURCE.
1400 PRINT"    A SPECIAL TRICK IS USED TO MAKE THE
1410 PRINT"NAME FOR THE SAVE.  LINES 140 THRU 170
1420 PRINT"LOCATE THE LENGTH AND ADDERSS POINTER
1430 PRINT"USED BY BASIC FOR STRING A$.
1440 PRINT"LINE 180 MAKES A$ EQUAL TO THE NAME FOR
```

```
 1450 PRINT"THE SAVE.  LINE 190 SETS THE LENGTH OF
 1460 PRINT"A$ FOR THE SAVE.  LINES 200 AND 210 SET
 1470 PRINT"THE ADDRESS OF A$ FOR THE SAVE NAME.
 1480 PRINT"FINALLY LINE 220 CALLS THE OPERATING
 1490 PRINT"SYSTEM ROUTINE TO DO THE SAVE.
 1500 GOSUB9000
 1600 PRINT"□■■■    TO ACTIVATE THE ■APPEND■ COMMAND
 1610 PRINT"TYPE ■RUN■.
 1620 PRINT"■    TO SAVE THE ■APPEND■ COMMAND AND
 1630 PRINT"INSTRUCTIONS TYPE ■RUN 100■.
 1640 PRINT"■■■■ WARNING "
 1650 PRINT"■    THE ■APPEND■ COMMAND DOES NOT FIX
 1660 PRINT"LINE NUMBERS!  APPENDING PROGRAMS WITH
 1670 PRINT"LINE NUMBERS OUT OF ORDER WILL HAVE
 1680 PRINT"STRANGE RESULTS WHEN RUN.
 1690 PRINT"    USE PET RENUMBER TO FIX SEGMENTS
 1700 PRINT"BEFORE APPENDING.
 1900 GOSUB9000
 2000 PRINT"□SYNTAX FOR ■APPEND■ COMMAND
 2010 PRINT
 2020 PRINT">APPEND "CHR$(34)"PROGRAM NAME"CHR$(34)
 2030 PRINT"↑  ↑      ↑
 2040 PRINT"|  |      |┌──────────────────────────┐
 2050 PRINT"|  |      └┤ NAME OF PROGRAM ON TAPE  |
 2060 PRINT"|  |       | #1 YOU WISH TO APPEND    |
 2070 PRINT"|  |       | TO THE PRESENT PROGRAM   |
 2080 PRINT"|  |       | IN PET MEMORY            |
 2090 PRINT"|  |       |                          |
 2100 PRINT"|  |       | IF OMITTED THE NEXT      |
 2110 PRINT"|  |       | PROGRAM ON TAPE #1 WILL  |
 2120 PRINT"|  |       | BE APPENDED              |
 2130 PRINT"|  |       └──────────────────────────┘
 2140 PRINT"|  |
 2150 PRINT"|  |┌──────────────────────────┐
 2160 PRINT"|  └┤ COMMAND NAME - A FOR SHORT |
 2170 PRINT"|   └──────────────────────────┘
 2180 PRINT"|┌──────────────────────────┐
 2190 PRINT"└┤ PROMPT CHARACTER MUST BE   |
 2200 PRINT" | IN FIRST COLUMN OF LINE    |
 2210 PRINT" └──────────────────────────┘
 2900 GOSUB9000
 5000 GOTO1000
 9000 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■";
 9010 PRINT"       ■ HIT ANY KEY TO CONTINUE ■";
 9020 GETA$:IFA$=""THEN9020
 9030 RETURN
READY.
```

```
LINE #  LOC      CODE        LINE

0002   0000                ;*************************************
0003   0000                ;*
0004   0000                ;*  APPEND BASIC PROGRAMS
0005   0000                ;*
0006   0000                ;*    FOR LEVEL 1 BASIC
0007   0000                ;*
0008   0000                ;*    3-29-79
0009   0000                ;*
0010   0000                ;*
0011   0000                ;*************************************


0013   0000                ;BASIC VARABLES
0014   0000                TXTPTR=$C9
0015   0000                BUF=$0A
0016   0000                VARTAB=$7C


0018   0000                ;BASIC ROUTINES
0019   0000                CHRGET=$00C2            ;INC TXTPTR AND GETS CHAR
0020   0000                CHRGOT=$00C8            ;GETS LAST CHAR
0021   0000                FINI=$C430              ;FIXES LINKS


0023   0000                ;OP SYSTEM VARABLES
0024   0000                TEMP1=$50
0025   0000                EAL=$E5                 ;END ADR FOR SAVE
0026   0000                EAH=$E6
0027   0000                FNLEN=$EE               ;LENGTH OF FILE NAME
0028   0000                FA=$F1
0029   0000                STAL=$F7                ;START ADR FOR SAVE
0030   0000                STAH=$F8
0031   0000                FNADR=$F9               ;ADDRESS OF FILE NAME
0032   0000                VERCK=$020B
0033   0000                SATUS=$020C
0034   0000                TAPE1=$027A
0035   0000                TAPE2=$033A


0037   0000                ;OP SYSTEM ROUTINES
0038   0000                FAH=$F5AE               ;READ ANY TAPE HEADER
0039   0000                LDAD2=$F64D             ;COPY PTRS FROM HEADER
0040   0000                SAVE=$F6B1              ;OP SYSTEM TAPE SAVE
0041   0000                PRT=$FFD2               ;PRINT CHAR IN A
0042   0000                LD410=$F42B             ;PRINT 'ING' MSG
0043   0000                TRD=$F88A               ;READ DATA FROM TAPE
0044   0000                TWAIT=$F913             ;WAIT FOR KEYBRD IRQ
0045   0000                LERR=$F3DB              ;PRT ERROR MSG
0046   0000                LD210=$F3E5             ;PRT READY,FIX VARTAB
0047   0000                ZZZ=$F667               ;SETS TBUF PTR
0048   0000                CSTE1=$F83B             ;ISSUE TAPE MSG
```

```
LINE # LOC      CODE        LINE

0049  0000                LD300=$F3FF             ;PRINTS FILE NAME
0050  0000                FAF=$F495               ;SEARCHS FOR FILE BY NAME
0051  0000                OP160=$F579             ;PRT 'FILE NOT FOUND ERROR'



0053  0000                      *=TAPE2
0054  033A              ;SET WEDGE CMD IN Z-PAGE CODE
0055  033A              ;
0056  033A  A9 4C        SETW  LDA #$4C          ;JMP INSTRUCTION
0057  033C  85 CB              STA CHRGOT+3
0058  033E  A9 47              LDA #<WEDGE       ;SET LO
0059  0340  85 CC              STA CHRGOT+4
0060  0342  A9 03              LDA #>WEDGE       ;SET HI
0061  0344  85 CD              STA CHRGOT+5
0062  0346  60          STRTS RTS



0064  0347              ;APPEND WEDGE CMD
0065  0347              ;
0066  0347  C9 3E        WEDGE CMP #'>           ;A WEDGE CMD?
0067  0349  D0 08              BNE WG200         ;NO
0068  034B  48                 PHA               ;MAYBE
0069  034C  A5 C9              LDA TXTPTR        ;WAS '>' IN COLUMN 1
0070  034E  C9 0A              CMP #<BUF
0071  0350  F0 08              BEQ WCMD          ;YES-WEDGE CMD



0073  0352  68          WG100 PLA               ;FINISH CHRGOT
0074  0353  C9 3A        WG200 CMP #':
0075  0355  B0 EF              BCS STRTS
0076  0357  4C CF 00           JMP CHRGOT+7



0078  035A  20 C2 00    WCMD  JSR CHRGET        ;AN APPEND CMD?
0079  035D  C9 41              CMP #'A
0080  035F  D0 F1              BNE WG100         ;NO



0082  0361  A2 01              LDX #1            ;YES-SET FOR CASSETTE #1
0083  0363  86 F1              STX FA
0084  0365  CA                 DEX               ;X=$00
0085  0366  86 EE              STX FNLEN         ;ZERO LENGTH OF FILE NAME
0086  0368  86 FA              STX FNADR+1       ;POINT INTO Z-PAGE
0087  036A  8E 0B 02           STX VERCK         ;SET FOR A LOAD
```

```
LINE # LOC      CODE        LINE

0089  036D  20 C2 00    WC100  JSR CHRGET       ;GET NEXT CHAR
0090  0370  AA                  TAX              ;IS THIS THE END
0091  0371  F0 17                BEQ WC210        ;YES-LOAD ANYTHING
0092  0373  C9 22                CMP #$22         ;A (")?
0093  0375  D0 F6                BNE WC100        ;NO-LOOP
0094  0377  A6 C9                LDX TXTPTR       ;START FILE NAME ONE+
0095  0379  E8                   INX
0096  037A  86 F9                STX FNADR



0098  037C  20 C2 00    WC200  JSR CHRGET       ;FIND END OF THE NAME
0099  037F  AA                  TAX              ;THIS THE END?
0100  0380  F0 08                BEQ WC210        ;YES
0101  0382  C9 22                CMP #$22         ;AN END DOUBLE QUOTE?
0102  0384  F0 04                BEQ WC210        ;YES
0103  0386  E6 EE                INC FNLEN        ;NO-KEEP CHARACTER
0104  0388  D0 F2                BNE WC200        ;BRANCH ALWAYS



0106  038A  20 67 F6    WC210  JSR ZZZ          ;SET TBUF PTRS
0107  038D  20 3B F8           JSR CSTE1        ;ISSUE TAPE MSG
0108  0390  20 FF F3           JSR LD300        ;PRT FILE NAME
0109  0393  A5 EE              LDA FNLEN        ;LOADING ANY FILE
0110  0395  F0 08              BEQ WC250        ;YES
0111  0397  20 95 F4           JSR FAF          ;NO-SEARCH FOR IT
0112  039A  D0 08              BNE WC270        ;SKIP IF FOUND
0113  039C  4C 79 F5    WC220  JMP OP160        ;FILE NOT FOUND MSG
0114  039F  20 AE F5    WC250  JSR FAH          ;READ ANY FILE
0115  03A2  F0 F8              BEQ WC220        ;ERROR IF NOT FOUND



0117  03A4              ;CALC NEW ADDRESS FOR APPEND SOURCE
0118  03A4              ;
0119  03A4  AD 7D 02    WC270  LDA TAPE1+3      ;GET LO END ADR
0120  03A7  38                 SEC              ;CALC DELTA
0121  03A8  ED 7B 02           SBC TAPE1+1      ;SUB BEGIN LO
0122  03AB  AA                 TAX              ;SAVE IN X
0123  03AC  AD 7E 02           LDA TAPE1+4      ;GET END ADR HI
0124  03AF  ED 7C 02           SBC TAPE1+2      ;SUB BEGIN ADR HI
0125  03B2  A8                 TAY              ;SAVE IN Y
0126  03B3  A5 7C              LDA VARTAB       ;CALC APPEND BEGIN
0127  03B5  38                 SEC
0128  03B6  E9 04              SBC #4
0129  03B8  8D 7B 02           STA TAPE1+1      ;SET LOAD NEW START
0130  03BB  A5 7D              LDA VARTAB+1
0131  03BD  E9 00              SBC #0
0132  03BF  8D 7C 02           STA TAPE1+2      ;NEW START HI
0133  03C2  8A                 TXA              ;GET DELTA LO
0134  03C3  18                 CLC              ;CALC NEW LOAD END ADR
0135  03C4  6D 7B 02           ADC TAPE1+1
0136  03C7  8D 7D 02           STA TAPE1+3
0137  03CA  98                 TYA              ;GET DELTA HI
```

```
APPEND 120.....PAGE 0004

LINE # LOC      CODE        LINE

0138  03CB  6D 7C 02           ADC TAPE1+2
0139  03CE  8D 7E 02           STA TAPE1+4



0141  03D1  20 4D F6           JSR LDAD2        ;COPY POINTERS FOR LOAD
0142  03D4  A2 00              LDX #0           ;PRT 'APPENDING' MSG
0143  03D6  8E 0B 02           STX VERCK        ;CLEAR FOR A LOAD
0144  03D9  BD F8 03    WC300  LDA MSG1,X       ;DONE WITH MSG
0145  03DC  F0 06              BEQ WC400        ;YES
0146  03DE  20 D2 FF           JSR PRT
0147  03E1  E8                 INX
0148  03E2  D0 F5              BNE WC300        ;BRANCH ALWAYS



0150  03E4  20 2E F4    WC400  JSR LD410+3      ;PRINT 'ING' MSG
0151  03E7  20 8A F8           JSR TRD          ;READ DATA FROM TAPE
0152  03EA  20 13 F9           JSR TWAIT        ;WAIT FOR KEY IRQ RESTORE
0153  03ED  AD 0C 02           LDA SATUS        ;GOOD LOAD?
0154  03F0  F0 03              BEQ WC500        ;YES
0155  03F2  4C DB F3           JMP LERR         ;NO-LOAD ERROR



0157  03F5  4C E5 F3    WC500  JMP LD210        ;GO FIX BASIC LINKS
0158  03F8  0D           MSG1   .BYTE $D,'APPEND',0
0158  03F9  41 50
0158  03FF  00
0159  0400              AAAA
0160  0400                     .END


ERRORS = 0000


SYMBOL TABLE

SYMBOL    VALUE

AAAA      0400    BUF       000A    CHRGET    00C2    CHRGOT    00C8
CSTE1     F83B    EAH       00E6    EAL       00E5    FA        00F1
FAF       F495    FAH       F5AE    FINI      C430    FNADR     00F9
FNLEN     00EE    LD210     F3E5    LD300     F3FF    LD410     F42B
LDAD2     F64D    LERR      F3DB    MSG1      03F8    OP160     F579
PRT       FFD2    SATUS     020C    SAVE      F6B1    SETW      033A
STAH      00F8    STAL      00F7    STRTS     0346    TAPE1     027A
TAPE2     033A    TEMP1     0050    TRD       F88A    TWAIT     F913
TXTPTR    00C9    VARTAB    007C    VERCK     020B    WC100     036D
WC200     037C    WC210     038A    WC220     039C    WC250     039F
WC270     03A4    WC300     03D9    WC400     03E4    WC500     03F5
WCMD      035A    WEDGE     0347    WG100     0352    WG200     0353
ZZZ       F667
END OF ASSEMBLY
```

SOME PET ROUTINES

by J. Butterfield

IN LAST MONTHS ISSUE WE FEATURED MR. BUTTERFIELD'S MEMORY MAP. (PG.27) TO FOLLOW IS A CONTINUATION OF THAT MAP, WHICH INCLUDES THE ADDRESSES OF C2AC THRU CDCO.

C2AC-C2D9	peeks at the stack for an active FOR loop
C2AD-C31C	'opens up' a space in Basic for insertion of a new line.
C31D-C329	tests for stack-too-deep and aborts if found.
C32A-C356	
C357-C388	sends a canned error message from C190 area, then drops into:
C389-C391	Signals 'ready'
C394-C3A9	gets a line of input, analyzes it, executes it
C3AC-C42E	handles a new line of Basic from keyboard; deletes old line, etc.
C430-C460	corrects the chaining between Basic lines after insert/ delete
C462-C476	receives a line from the keyboard into the Basic buffer
C479-C48C	gets each character from keyboard
C48D-C521	looks up the keywords in an input lines and changes to "tokens"
C522-C550	searches for the location of a Basic line from number in 8, 9
C551-C599	implements NEW command - clears everything
C59A-C5A7	sets the Basic pointer to start-of-programs
C5A8-C647	performs LIST command
C649-C647	executes a FOR statement
C692-C6B4	continues to build FOR vectors
C6B5-C6EF	reads and executes the next Basic statement, finds next line, etc.
C6F2-C70A	executes the Basic Command as a subroutine
C70D-C71B	performs RESTORE
C71C-C742	handles STOP, END, and BREAK procedures.
C745-C75E	performs CONT
C75F-C76D	
C770-C772	performs CLR
C775-C77D	performs RUN
C780-C79A	performs GOSUB
C79D-C7C9	performs GOTO
C7CA-C7FD	performs RETURN
C7FE-C81E	scans for start of next Basic Line
C820-C840	performs IF
C843-C862	performs ON
C863-C89A	gets a fixed-point number from Basic and stores in 8, 9
C89D-C91B	performs LET
C91C-C97E	check numeric digit/move string pointer
C97F-C982	performs PRINT#
C985-C996	performs CMD
C999-CA24	performs PRINT
CA27-CA41	prints string from address in Y, A
CA44-CA76	prints a character
CA77-CA9E	handles bad input data
CA9F-CAC5	performs GET
CAC6-CADF	performs INPUT#
CAEO-CB14	performs INPUT
CB17-CB21	prompts and receives the input
CB24-CC11	performs READ
CC12-CC35	canned messages: EXTRA IGNORED; REDO FROM START
CC36-CC8F	performs NEXT
CC92-CCB5	checks Basic format, data type, flags TYPE MISMATCH
CCB8-CD38	inputs and evaluates any expression (numeric or string)
CD3A-Cd9C	pushes a partially-evaluated argument to the stack
CD9D-CDB9	evalues a numeric, variable, or pi, or identifies other symbol
CDBC-CDCO	value of pi in floating binary

THE FOLLOWING MEMORY MAP REFLECTS ROM CHANGES IN THE NEW 16K/32K PETS.

Appendix A.

Detailed CBM Memory Map

CBM Memory Allocation By 4K Blocks

BLOCK #	TYPE	START ADDRESS	FUNCTION
*0	RAM	$0000	Working, text, variable storage.
1	RAM	$1000	Test variable storage (8K only)
2	---	$2000	Expansion RAM
3	---	$3000	Expansion RAM
4	---	$4000	Expansion RAM
5	---	$5000	Expansion RAM
6	---	$6000	Expansion RAM
7	---	$7000	Expansion RAM
8	RAM	$8000	Screen memory (1K)
9	---	$9000	Expansion ROM
10	---	$A000	Expansion ROM
11	---	$B000	Expansion ROM
12	ROM	$C000	BASIC (principally statement interpreter).
13	ROM	$D000	BASIC (principally math package).
*14	ROM I/O	$E000 $E800	Screen editor. All internal CBM I/O.
15	ROM	$F000	OS diagnostics

*see expanded description

Block 0 By 256 Byte Pages

PAGE	TYPE	START ADDRESS	FUNCTION
**0	RAM	0000	BASIC OS working storage
**1	RAM	0100	Stack
**2	RAM	0200	O S working storage
**3	RAM	0300	Cassette buffers.
4-15	RAM	0400	BASIC text area

** see expanded description by page

Block 14 By 2K Segment

PAGE	TYPE	START ADDRESS	FUNCTION
0	ROM	$E000	Screen editor
1	I/O	$E800	CBM I/O

I/O Device Base Addresses

PAGE	TYPE	START ADDRESS	FUNCTION
0	PIA	$E810	Keyboard
1	PIA	$E820	IEEE-488
2	VIA	$E840	USR PORT cassette

CBM PAGE ZERO MEMORY MAP

FROM	TO	DESCRIPTION
000	--	$4C constant (6502 JMP instruction).
001	002	USR function address lo, hi.
3	--	Starting delimiter
4	--	Ending delimiter
5	--	General counter for BASIC.
Evaluation of variables		
6	--	Flag to remember dimensioned variables.
7	--	Flag for variable type; 0#numeric; 1 ÷ string.
8	--	Flag for integer tape.
9	--	Flag to crunch reserved words (protects '& remark).
10	--	Flag which allows subscripts in syntax.
11	--	Flags INPUT or READ.
12	--	Flag sign of TAN.
13	--	Flag to suppress OUTPUT (+ normal; – suppressed).
14	--	Active I/O channel #.
15	--	Terminal width (unused).
16	--	Limit for scanning source columns (unused).
17	18	Line number storage
13	--	Flag to suppress OUTPUT (+ normal; – suppressed).
19	--	Index to next available descriptor.
20	21	Pointer to last string temporary lo; hi.
22	29	Table of double byte descriptors which point to vaiables.
30	31	Indirect index #1 lo; hi.
32	33	Indirect index #2 lo; hi.
34	39	Pseudo register for function operands.
Data storage maintenance		
40	41	Pointer to start of BASIC text area lo; hi byte.
42	43	Pointer to start of variables lo; hi byte.
44	45	Pointer to array table lo; hi byte.
46	47	Pointer to end of variables lo; hi byte.
48	49	Pointer to start of strings lo; hi byte.
50	51	Pointer to top string space lo; hi byte.
52	53	Highest RAM adr lo; hi byte.
54	55	Current line being executed. A two in 54 means statement executed in a direct command.
56	57	Line # for continue command lo; hi.
58	59	Pointer to next STMNT to execute lo; hi.
60	61	Data line # for errors lo; hi.
62	63	Data statement pointer lo; hi.

Expression evaluation

64	65	Source of INPUT lo; hi.
66	67	Current variable name.
68	69	Pointer to variable in memory lo; hi.
70	71	Pointer to variable referred to in current FOR-NEXT.
72	73	Pointer to current operator in table lo, hi.
74	--	Special mask for current operator.
75	76	Pointer to function definition lo; hi.
77	78	Pointer to a string description lo; hi.
79	--	Length of a string of above string.
80	--	Constant used by garbage collect routine.
81	--	$4C constant (6502 JMP inst).
82	83	Vector for function dispatch lo; hi.
84	89	Floating accumulator #3.
90	91	Block transfer pointer #1 lo; hi.
92	93	Block transfer pointer #2 lo; hi.
94	99	Floating accumulator #1. (USR function evaluated here).
100	--	Duplicate copy of sign of mantissa of FAC #1.
101	--	Counter for # of bits of shift to normalize FAC # 1.
102	107	Floating accumulator #2.
108	--	Overflow byte for floating argument.
109	--	Duplicate copy of sign of mantissa.
110	111	Pointer to ASCII rep of FAC in conversion routine lo; hi.

RAM subroutines

112	--	CHRGOT RAM code. Gets next character from BASIC text.
118	--	CHRGOT RAM code regets current characters.
119	120	Pointer to source text lo; hi.
136	140	Next random number in storage.

Operating System page zero storage

141	143	;24 Hr clock in 1/60 sec.
144	145	;IRQ RAM vector
146	147	:BRK inst ram vector
148	149	;NMI RAM vector
150	--	;I/O operation status byte
151	--	;last key index
152	--	
153	154	correction factor for clock
157	--	Verify flag
158	--	Index to keyboard queue
159	--	Reverse field on
160	166	Multiply defined
167	--	Cursor on flag
168	--	Count of jiffies to blink cursor
169	--	Multiply defined
170	--	Character saved during blink
171	173	Multiply defined
174	--	;Pointer into logical file table
175	--	;Default input device #
176	--	;Default output device #
177	--	;Vertical parity for tape
178	185	Multiply defined
186	--	SYNC on tape header count
187	188	Pointer to active cassette
189	--	Multiply defined
190	--	Bit/byte tape error
191	--	Reading shorts

192		Index to addresses for tape error correction
193	--	Multiply defined
194	--	Flag for cassette read..tolls*
195	--	Count of second of shorts to write before data
196	197	Pointer to cursor position
197	198	Multiply defined
199	200	Load start address
201	202	Load end address
203	--	
204	--	
205	--	Quote mode flag
206	208	Multiply defined
209	--	;Length current file name Str
210	--	;Current file logical addr
211	--	;Current file 2nd addr
212	--	;Current file primary addr
213	217	Multiply defined
218	219	;Addr current file name str
220	221	Multiply defined
222	--	;Cassette read block count
223	--	Multiply defined
224	248	Table of LBB of start addr of video display lines
249	--	
250	--	
251	--	
252	--	
253	254	

Page 1

62 byte on bottom are used for error correction in tape reads. Also, buffer for ASCII when BASIC is expanding the FAC into a printable number. The rest of page 1 is used for storage of BASIC GOSUB and FOR NEXT context and hardware stack for the machine.

CBM PAGE TWO MEMORY MAP

FROM	TO	DESCRIPTION
512	592	;Basic input buffer
512	513	program counter
514	--	processor status
515		accumulator
516		X index
517		Y index
518		Stack pointer
519	520	;User modifiable IRQ
593	602	;Logical file numbers
603	612	;Primary device numbers
613	622	;Secondary addresses
623	633	Keyboard Buffer
634	825	Tape buffer #1
826	1017	Tape buffer #2
1018	1019	Unused

PASSWORD ROUTINE

by F.L. Peters

```
  5 PRINT"  ":Y=0
 10 POKE 525,0:WAIT 525,4
 20 GET A$,B$,C$,D$
 30 IF A$+B$+C$+D$="TEST" GOTO 60
 40 Y=Y+1:IF Y=2 GOTO 999
 50 GOTO 10
 60 REM***PROTECTED PROGRAM STARTS HERE***
999 END
```

This short password routine can be modified for any password, and is very useful to stop unauthorized access to programs or files, and the password will not be printed. Here is what each step does:

LINE 5: Clears the screen and sets the value of variable Y to zero. Y is the variable used to count the number of attempts at inputing the password.

LINE 10: Location 525 is the keyboard counter and keeps track of the number of keys that has been typed since the keyboard was last strobed. By POKING zero into location 525 the PET is fooled into thinking no keys have been typed. The WAIT statement makes the PET wait until a set number of keys has been typed. Example: WAIT 525,X where X is the number of keys to be typed before the input is used by the PET. The value is the number of letters in the password.

LINE 20: This is the use of the GET command to input characters from the keyboard. The number of GET variables will be the number of letters in the password with each variable corresponding to each letter of the password.

LINE 30: Here the GET variables are assembled into a word to be tested against the password. If the input word matches the password the program goes to line 60 where the protected program starts. The password is contained in the " " marks in this statement.

LINE 40: Here the Y variable is incremented by one each try and then tested to see if the limit of attempts has been reached. In the following example the "X" indicates the number of tries at inputing the password and if exceeded the program goes to line 999 and stops. An example: IF Y="X" GOTO 999

LINE 50: Loops back to line 10 for another try if the number of attempts is below the number allowed in line 40.

LINE 999:Contains the END command to stop the program.

BITS AND PIECES

SWITCHING FROM THE PRINTER TO THE SCREEN

The following program illustrates a technique for switching a CMD mode from the Printer to the screen and back. This avoids the necessity of saying 'PRINT#:CLOSE' every time you wish to exit a CMD mode.

```
100 REM***THIS IS A METHOD OF SWITCHING FROM THE PRINTER TO THE SCREEN
110 REM***WITHOUT REVERTING TO A 'SYNTAX ERROR' OR SOME OTHER ABNORMAL
120 REM***END OF THE PRINT COMMAND.
130 REM***GLENN HOELSCHER
140 REM***5773 DEXTER CIRCLE
150 REM***ROHNERT PARK. CA. 94928
160 REM***707+542-6773
170 PRINT:PRINT
200 OPEN 4,4:CMD 4:REM***OPENS PRINTER IN NORMAL WAY
210 PRINT "THIS SHOULD COME OUT ON THE PRINTER":REM***THIS IS THE MESSAGE
220 CLOSE 4:REM***NORMAL CLOSE OF ABOVE COMMAND
230 REM***NOTICE. NO PRINT # WAS NEEDED AND '?' IS A VALID PRINT COMMAND
300 OPEN 4,3:CMD 4:REM***NOW WE DIRECT OUR PRINT TO THE SCREEN***
310 PRINT "THIS SHOULD COME OUT ON THE SCREEN"
320 REM***AGAIN, NO NEED FOR A PRINT # AND '?' IS A VALID COMMAND
330 REM***SINCE WE WANT TO GO BACK TO PRINTER, WE MUST CLOSE THE SCREEN
340 CLOSE 4:GOTO 200
READY.
```

PROTECTING PROGRAMS

For those of you concerned about unauthorized copying of your programs, Len Lindsay of the PET GAZETTE has fowarded to us the name of a vendor who may be able to solve your problem. Mr. Lindsay visited BC COMPUTING and was fortunate enough to spend an hour with the main PET programmer. He was very impressed by their system and is pleased to announce that it works!

According to Mr. Lindsay, you can load a protected program by typing L-O-A-D, (RETURN). You will then be amazed to see the program LOAD and immediately RUN! If for some reason you manage to break out of the program (the stop key does not work) the program will not LIST correctly and you lose control of your PET and have to turn it off and on again. A SAVE will not work, neither will the SYS equivalent.

Adding this protection is very complex so BC COMPUTING charges a nominal fee for the service of protecting your program.

For further information please write to;

BC COMPUTING
2124 Colorado Ave.
Sun Prairie, WI 53590

Users' Directory &

Announcements

One of the major advantages in being a member of the PET USERS' CLUB is the ability to get hold of PET related Software and ideas. Although our Master Library of programs is now growing, we get frequent Software inquires for a wide range of applications.

In this issue, we have included the current Users' Directory, containing lists of people writing software, importing literature or starting local PET Groups. If you would like to use your PET for fun and profit, why not offer personal tutoring in PET programming to new PET owners. Alternatively, if you require a program to be written for you, ask for contacts via the USERS' DIRECTORY. The possibilities are endless. Please write to the EDITOR, U.S. PET USERS' CLUB, at our NEW address below.

To include your name in the USERS' DIRECTORY, please complete the following form:

TO: THE EDITOR, U.S. PET USERS' CLUB, Commodore Business Machines Inc., 3330 Scott Blvd., Santa Clara, Calif. 95050.

NAME:_____

ADDRESS_____

_____

SERVICES OFFERED/SPECIALIST AREA OF INTEREST:_____

_____

_____

To include as many contacts as possible, we must restrict each USER to only one line of description.

COMMODORE reserves the right to edit or withdraw any entry.

****

LISTED BELOW ARE PET USERS WHO HAVE RECENTLY SUBMITTED THEIR SPECIALTY OR AREA OF INTEREST TO FURTHER COMMUNICATION WITH PET OWNERS THROUGHOUT THE UNITED STATES. IF YOU WOULD LIKE TO OFFER YOUR SERVICES TO OTHERS, PLEASE FILL OUT THE "USER DIRECTORY" FORM ON THE PREVIOUS PAGE.

NAME AND ADDRESS	SERVICES OFFERED/SPECIALTIES
Charles & Robert Ames 1717 Penn Joplin, MO 64801	Programming in BASIC, FORTRAN, ASSEMBLY ASSEMBLY LANGUAGE on all systems.
Nilan R. Beardall Brigham Young Univ. 175 TMCB Provo, Ut. 84602	General purpose, Games
Charles Branson Box 3061 University Station Moscow, ID 83843	EDUCATION/GAMES/etc.
Dave Caulkins The PCNOT Committe 340 E. Middlefield Rd. Mtn. View, CA	ELECTRONIC MAIL SYSTEMS VIA PETs with MODEMS.
Russell Grokett PET Library 401 Monument Rd. Apt 177 Jacksonville, FL	JAPS-Jax Area PET Society offers PET Software. $1.50 per program SASE for list to PET LIBRARY
Innovision P.O. Box 1317 Los Altos, CA 94022	Innovision offers both Software and Hardware (such as low cost device to allow handwritten character input to the PET).
Marvin Mallon 6914 Berquist Ave. Canoga Park, CA 91307	Business and Engineering Applications.
Microsette Co. 777 Palomar Ave. Sunnyvale, Ca 94086	Duplication of cassette programs. 100 quantity minumum.
Bob Nathanson Kings Park High School Route 25A Kings Park, NY 11754	Using PETS for both high school students and Adult Education classes in computer literacy.
Carl C. Peck, M.D. 56 Hillcrest Larkspur, CA 94939	Math modeling ad, simulation, pharmacokinetics, biostatistics.

NAME AND ADDRESS	SPECIALTY
Princeton Energy Group 729 Alexander Rd. Princeton, NJ 08540	Passive solar heating and cooling, building design
Thomas B. Simpson 5557 Coddington, Apt. 1C Kalamazoo, MI 49009	Home Applications Programming
Rev. James Strasma 120 W. King St. Decatur, IL 62521	Church related programs and programming service. (Central Illinois PET Users)
Ed Zwieback 175 Cordova Walk Long Beach, CA 90803	Real Estate Fianance, Acoustics

~~~~~~

TAPES

If you have found your blank cassettes of inferior quality for creating programs you may wish to try AGFA's. These were recommended to us by Len Lindsay of the PET GAZETTE. He feels they produce one tape which is far superior to ALL others which can be used with the PET. These cassettes use "AGFA Premium" tape. You should be careful of companies advertising AGFA tapes for there are several different grades. "AFGA Premium" tape in the highest quality cassette housing are available from COMPUTER WAY. Their prices on AGFA Premium C-10 cassettes are lower than others advertised.

25 at $1.00 each (total $25)

50 at $0.96 each (total $48)

100 at $0.85 each (total $85)

200 at $0.80 each (total $160)

400 at $0.75 each (total $300)

Approved Computer Clubs receive a 10% discount on orders of 200 or more. Shipping and quick delivery anywhere in the USA is included but boxes are extra. Order directly from:

COMPUTER WAY
P.O. BOX 7006
Madison, WI 53707
~~~~~~

THE LIST OF PET USER GROUPS LISTED BELOW IS BY NO MEANS COMPLETE. PLEASE NOTIFY US IF WE OMITTED YOUR GROUP.

Association of Personal Computer Users
5014 Rodman Rd. . . . . Bethesda, MD 20016

Amateur Computer Group of New Jersey
Box 379 . . . . . South Bound Brook, NJ 07076

Bambug
1450 53rd St. Emeryville, CA 94608

JAPS-JACKSONVILLE AREA PET SOCIETY
401 Monument Road #177 . . . . Jax, Florida, 32211

Lawrence Hall of Science, UC Berkeley
Computer Project, Room 254. . Berkeley, CA 94720

Las Vegas PET Users
4884 Iron Ave . . . . . Las Vegas, Nev. 89110

Lincoln Computer Club
750 E. Yosemite . . . . . Manteca, CA 95336

Madison PET Users
1400 East Washington Ave. Madison, WI 53703

Northern New England Computer Society
P.O. Box 69 . . . . . Berlin, NH 03570

North Orange County Computer Club
3030 Topaz, Apt. A . . . Fullerton, CA 92361

Northwest PET User Group
P.O. Box 482 . . . . . Vashon, WA 98070

PET User Club (CAPE)
2054 Eakins C7 . . . . . Reston, VA 22091

PET User Group
2235 Lakeshore Dr. . . . . .Muskegon, MI 49441

PET User Group c/o Meyer
35 Barker Ave. . . .White Plains, NY 10610

PET User Group . . . . .
Texas A & M Microcomputer Club, Tex. A & M, Tex.

PET User Group
P.O. Box 371 . . . . Montgomeryville, PA 18936

PET User Group
2323 Washington Blvd. . . . Ogden, Utah 84401

PUG
7170 S.W. 11th St., West Hollywood, Fla. 33023

PUG of the Silicone Valley
22355 Rancho Verta.....Cupertino, Calif. 95014

Sacramento PET Workshop
P.O. Box 26314 . . . . . Sacramento CA 95826

SCOPE
1020 Summit Circle . . . Carrollton, Texas 75006

SPHINX
314 10th Ave. . . . . Oakland, CA 94606

St. Louis Club
40 Westwood Court . . . . . St. Louis, Mo. 63131

The Human Society--United PET Users
1929 Northport Dr. #6 . . . . Madison WI 53704

Valley Computer Club
P.O. Box 6545 . . . . . Burbank, CA 91510

NEW DEALERS

IN LAST MONTH'S ISSUE WE PUBLISHED A 4 PAGE LIST OF OUR CURRENT DEALERS. SINCE THAT DATE WE HAVE ACQUIRED SEVERAL MORE DEALERS AND ARE PASSING THEM ONTO YOU. PLEASE NOTE THOUGH THAT A FEW ARE SIMPLY A CHANGE OF ADDRESS.

ALASKA

Klopf Words
SR 51367
Fairbanks 99701
907-452-8502

CALIFORNIA

Active Business Forms
1148 Alpine Rd., Suite A
Walnut Creek 94596
415-938-1230

The Base Station
2101 Pacheco St.
Concord 94520
415-685-7388

Computer Age, Inc.
4688 Convoy St., Suite 105
San Siego 92111
714-565-4042

Computerland/Belmont
1625 El Camino Real
Belmont 94002
415-595-4232

Computerland/Marin
1930-4th St.
San Rafael 94901
415-459-1767

Computerland/Pasadena
81 N. Lake St.
Pasadena 91101
213-449-3205

Computerland/San Diego East
2992 Navajo Rd.
El Cajon 92020
714-464-5656

Computerland/Santa Maria
223 S. Broadway
Santa Maria 93454
805-928-1919

Computer World, Inc.
3808 W. Verdugo Ave.
Burbank 91505
213-848-5521

Computer World, Inc.
5848 Sepulveda Blvd.
Van Nuys 91411
213-786-7411

Computer World, Inc.
6789 Westminster Ave.
Westminster 92683
714-898-8330

COLORADO

The Neighborhood Store
13045 W. Alameda Pky.
Lakewood 80215

FLORIDA

Computerland/Jacksonville
2777-6 University Blvd.
Jacksonville 32217
904-731-2471

GEORGIA

Olson Electronics
3131 Campbellton Rd.
Atlanta 30311
404-349-3628

ILLINOIS

Appletree Stereo, Inc.
1022 W. Lincoln Hwy.
DeKalb 60115
815-758-2442

Appletree Stereo, Inc.
1645 N. Alpine
Rockford 61107
815-226-9826

Appletree Stereo, Inc.
117 E. Beaufort St.
Normal 61761
309-452-4215

Computerland/Arlington Hgt.
50 E. Rand Rd.
Arlington Heights 60004
312-255-6488

Computerland/Mundelein
1500 S. Lake St.
Mundelein 60060
312-949-1300

Computerland/Downers Grove
136 W. Ogden Ave
Downers Grove 60515
312-964-7762

Orcutt Business Machines
431 First St.
LaSalle 61301
815-224-2774

INDIANA

Stewart Business Machines
4778 Broadway
Gary 46408
219-884-9474

KANSAS

Personal Computer Center
3819 W. 95th St.
Overland Park 66026
913-649-5942

MARYLAND

Your Own Computer, LTD.
10678 Campus Way South
Upper Marlboro, 20870
301-350-3500

MASSACHUSETTS

New England Electronics Co.
679 Highland Ave.
Needham 02192
617-449-1760

MICHIGAN

Computermart of Royal Oak
560 W. 14 Mile Rd.
Clawson 48017
313-288-0040

Lafayette Radio Elec.
3127 W. Huron
Pontiac 48054
313-681-7400

Olson Electronics
14243 Gratiot
Detroit 48205
313-372-1317

Lafayette Radio Elec.
34208 Van Dyke
Sterling Heights 48077
313-268-8550

Main System, Inc.
1161 N. Ballenger Hwy, S.8
Flint 48504
313-232-3130

MINNESOTA

Johnsons RIC Sales
1325-97th Ave. N.W.
Coon Rapids 55433
612-755-7037

Zim Computers
5717 Xeres Ave
Brooklyn Center 55429
612-560-0336

NEW JERSEY

Cardiotronics
639 Ridge Rd.
Lyndhurst 07071
201-935-6677

Saturn Management, Inc.
600 Washington Ave.
Carlstadt 07072
201-939-4800

NEW YORK

United Photocopy Co., Inc.
41 Union Square
NY. NY 10003
212-929-4826

S.S. Computer Enterprises
224 E. Madison Ave.
Cresskill 07626
201-567-8076

Thor Electronics Corp.
321 Pennsylvania Ave.
Linden 07036
201-486-3300

NEW YORK

Computerland/Nassau Cnty.
79 Westbury Ave.
Carle Place, L.I. 11514
516-742-2262

Olson Electronics
711 Main St.
Buffalo 14203
716-856-2504

Olson Electronics
4401 Transit Rd.
Buffalo 14221
716-633-6644

NY Astrology Ctr. Book Store
127 Madison Ave.
NY City 10016
212-679-5676

Olson Electronics
3259 Sheridan Dr.
Buffalo 14226
716-837-6300

Olson Electronics
3768 Seneca St.
West Seneca 14224
716-675-4330

OHIO

Computerland/Columbus
6429 Busch Rd.
Columbus 43229
614-888-2215

Olson Electronics
414 Northfield Rd.
Bedford 44146
216-663-5970

Olson Electronics
36212 Euclid Ave.
Willoughby 44094
216-946-5457

Inductive Components
1200 Ferris Rd.
Amelia 45102
513-752-4731

Olson Electronics
1193 W. Pleasant Valley Rd.
Parma 44134
216-888-6366

OKLAHOMA

Audio Horizons
3707 E. Frank Phillips
Bartlesville 74003
918-333-7748

Bradford Brothers
507 S. Main
Tulsa 74103
918-584-4558

OREGON

Computerland/Portland
12020 S.W. Main St.
Tigard 97223
503-620-6170

PENNSYLVANIA

The Computer House
1000 Greentree Rd.
Pittsburgh 15220
412-343-1339

TEXAS

Foleys
2103 Ernestine
Houston 77023
713-651-6070

Computerland/Houston Bay A.
17647 El Camino Real
Houston 77058
713-488-8153

UTAH

The Computer Works
740 S. State St.
Provo 84601
801-374-0204

WASHINGTON

Omega Computers
839 - 106th Ave. NE
Bellevue 98004
206-455-1138

Omega Computers
5421 - 196th SW
Lynwood 98036
206-775-7585

COMMODORE BUSINESS MACHINES, INC.
3330 SCOTT BLVD.
SANTA CLARA, CA 95050

COMMODORE/MOS
VALLEY FORGE CORPORATE CENTER
950 RITTENHOUSE ROAD
NORRISTOWN, PENN 19401, USA

COMMODORE SYSTEMS DIVISION
360 EUSTON RD.
LONDON NW1 3BL, ENGLAND

COMMODORE BUROMASHNINEN GmbH
FRANKFURTER STRASSE 171-175
6078 NEW ISENBURG WEST GERMANY

COMMODORE JAPAN LIMITED
TAISEI-DENSHI BUILDING
8-14 1kue 1-CHOMEASAHI-KU, OSAKA 535 JAPAN

COMMODORE ELECTRONICS (HONG KONG) LTD.
WATSONS ESTATES
BLOCK C, 11th FLOOR
HONG KONG, HONG KONG

P/N 320851-A04 and A05

Commodore Business Machines, Inc.
3330 Scott Road
Santa Clara, California 95050

Bulk Rate
U.S. Postage
PAID
Permit No. 2196
San Jose, CA
95131

Dated Material

Do not forward