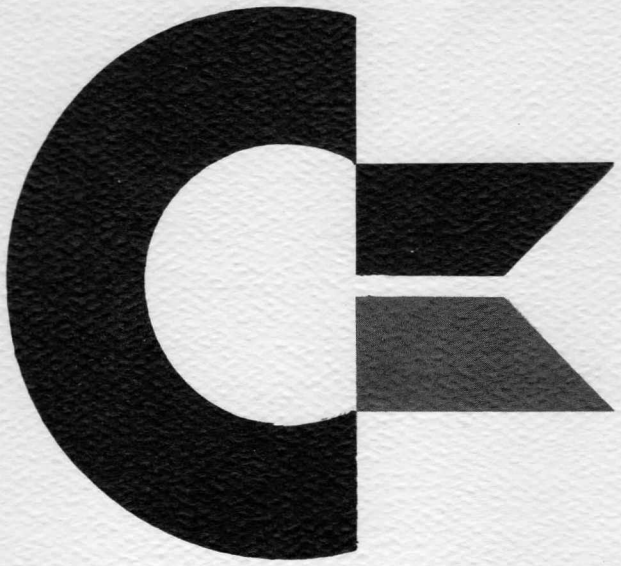


VOLUME I ISSUE 2

commodore

PET USERS CLUB NEWSLETTER

4-7-79

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 PUC Readers:

Our first effort to produce better communication with PET Users', brought on numerous responses. We now realize how necessary this NEWSLETTER is and will continue to do the best job possible.

Along with all the problems of the initial production of the first NEWSLETTER, we ran across further irratations after hours of writing, coordination, and editing. After mass reproduction, we discovered to our dismay poor print. We are pleased to announce however, that the problem has been solved, and we hope it has not caused you any inconvienence.

We are here to accommodate YOU, therefore programs of illegible quality have been reprinted and follow these error corrections.....

- One correction that is necessary was pointed out by Mark Hodes, of Palo Alto, Calif. It is on page 15 Line 1090; It should read as; POKE 59411, 61:Close 1.

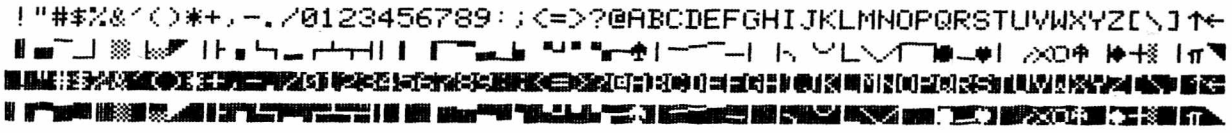
- On page 19 the routine for complete character set, was typed in for printing only on the printer. The Print Out which follows will now display and print.

Please note that to provide a good information service to PET Users, we intend to regularly mention equipment, software and services offered by companies and individuals not directly linked to Commodore. In doing so, we are not making recommendations and cannot be held responsible for the validity of any statements made.

Once again we thank you for your interest and patience, and hope our next issue contains even more interesting information from other PUC members.

The Editor

PAGE 9 ISSUE 1 , reprint



```
10 FORI=32TO95:A$=A$+CHR$(I):NEXT
20 FORI=160TO223:B$=B$+CHR$(I):NEXT
30 C$="■"+A$
40 D$="■"+B$
50 PRINTA$
60 PRINTB$
70 PRINTC$
80 PRINTD$
```

 (by C. Shaw)

PLOTTER

```
1 Z$="■■■■■■■■■■■■■■■■"
2 FORI=1TO15
3 PRINT"■";:FORX=1TO40
4 ONIGOSUB13,14,15,16,17,18,19,20,21,22,23,24,25,26,27
5 Z=C*X+D:ONIGOSUB30,40,50,60,70,80,90,100,110,120,130,140,150,160,170
6 Y=INT(A*Y+B)
7 PRINTZ$"■Y = "A$
8 POKE((24-Y)*40+32807)+X,46
9 NEXT:FORJ=1TO1000:NEXT:NEXT:GOTO2
12 REM**********************************THE CONSTANTS Y=AY+B X=CX+D******
13 A=7:B=12:C=.0019:D=.035:RETURN
14 A=.6:B=0:C=1:D=0:RETURN
15 A=7:B=13:C=.5:D=0:RETURN
16 A=7:B=13:C=.5:D=0:RETURN
17 A=1.1:B=11:C=.3:D=.2:RETURN
18 A=.07:B=2:C=1:D=-20:RETURN
19 A=.0018:B=12:C=1:D=-19:RETURN
20 A=1:B=2:C=.1:D=-.9:RETURN
21 A=7:B=12:C=.1:D=.2:RETURN
22 A=.5:B=12:C=.05:D=-.902:RETURN
23 A=9:B=12:C=.0021:D=.032:RETURN
24 A=100:B=12:C=.6:D=5.5:RETURN
25 A=18:B=5:C=.28:D=0:RETURN
26 A=20:B=2:C=1:D=0:RETURN
27 A=20:B=2:C=.1:D=-2:RETURN
29 REM*******************************************THE FUNCTIONS Y = F(X)********
30 Y=0:A$="0":RETURN
40 Y=Z:A$="AX+B":RETURN
50 Y=COS(Z):A$="COS(X)":RETURN
60 Y=SIN(Z):A$="SIN(X)":RETURN
70 Y=TAN(Z):A$="TAN(X)":RETURN
80 Y=(Z)↑2:A$="X□2":RETURN
90 Y=(Z)↑3:A$="X□3":RETURN
100 Y=EXP(Z):A$="E□X":RETURN
110 Y=LOG(Z):A$="LN(X)":RETURN
120 Y=1/Z:A$="─□■1/■X":RETURN
130 Y=SIN(1/Z):A$="SIN(─)□■■1/■■X":RETURN
140 Y=SIN(Z)/Z:A$="□SIN(X)/■■■■■■■■■■■■■■■■■X":RETURN
150 Y=ABS(SIN(Z)):A$="|SIN(X)|":RETURN
160 Y=RND(1):A$="RND(X)":RETURN
170 Y=EXP(-(Z)↑2):A$="NORMAL CURVE":RETURN
```

```
10 REM DISABLE STOP KEY EXAMPLE
100 GOSUB 10000:REM *** SET UP MACH CODE AND DISABLE STOP ***
200 PRINT "□";:REM *** CLR SCREEN ***
1000 REM*********************************************
1005 REM*                                           *
1010 REM*    SAMPLE PROGRAM WITH NO STOP KEY        *
1015 REM*                                           *
1020 REM*   PROGRAM ECHOS ALL KEYS TYPED AT         *
1030 REM*   THE KEYBOARD. EXIT BY TYPING THE        *
1040 REM*   PASSWORD 'CBM'.                         *
1045 REM*                                           *
1050 REM*********************************************
1100 REM *** BLINK CURSOR ***
1120 PRINT "▒■";:REM *** PRINT SHADY CHAR AND CRSR LEFT ***
1130 FOR I=1 TO 50:REM *** WAIT AND GET ANY CHARACTERS ***
1140 GET A$:IF A$<>"" THEN 3000
1150 NEXT
1200 PRINT " ■";:REM *** PRINT SPACE AND CRSR LEFT ***
1230 FOR I=1 TO 50:REM *** WAIT AND GET ANY CHARACTERS ***
1240 GET A$:IF A$<>"" THEN 3000
1250 NEXT
1300 GOTO 1100
3000 REM **** PASSWORD TEST ****
3010 B$=RIGHT$(B$,2)+A$:REM *** TEST LAST THREE CHARACTERS ***
3020 IF B$<>"CBM" THEN 4000
3040 PRINT "■■■   ":REM *** ERASE PASSWORD ***
3050 SYS(848):REM *** ENABLE STOP KEY AGAIN ***
3060 END
4000 PRINT " ■"A$;:REM *** ECHO LAST CHARACTER ***
4020 GOTO 1100
10000 REM*****************************************
10010 REM*                                       *
10020 REM*       MACHINE LANGUAGE PROGRAM        *
10030 REM*                                       *
10040 REM*   SYS(832) DISABLE STOP KEY FUNCTION  *
10050 REM*   SYS(848) ENABLE STOP KEY FUNCTION   *
10060 REM*                                       *
10070 REM*****************************************
11000 DATA 120,169,96,141,25,2,169,3
11010 DATA 141,26,2,88,96,0,0,0
11020 DATA 120,169,133,141,25,2,169,230
11030 DATA 141,26,2,88,96,0,0,0
11040 DATA 32,234,255,169,255,141,9,2
11050 DATA 76,136,230,0
12000 RESTORE:FORI=832TO874:READW:POKEI,W:NEXT
12010 SYS(832):REM *** DISABLE STOP KEY ***
12020 RETURN
```

Data Exchange

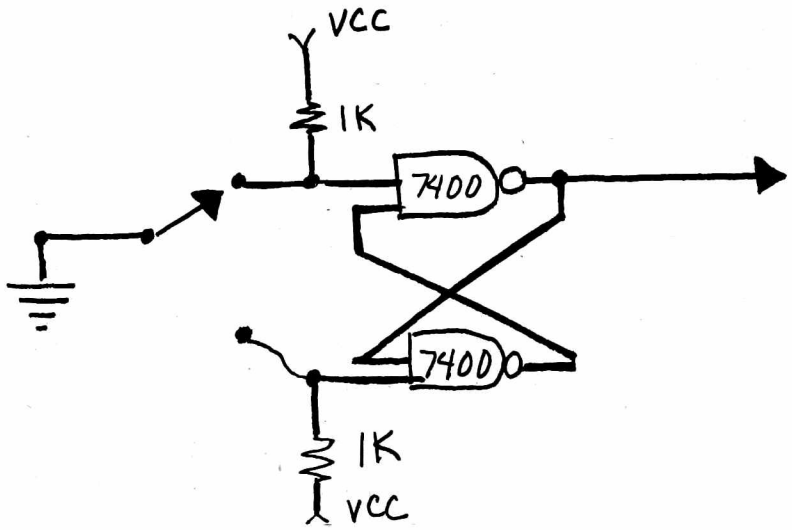
For better accomodation and communication with our USERS', this section will cover answers to your inquires not covered in the NEWSLETTER.

The foll owing three questions are in response to Charles A. McCarthy, of St. Paul, Minn.

Q. The booklet "An introduction to your new PET" that came with my PET states that positions A27 and A28 are RES and IRQ. Can PET be harmed by simply bringing these pins to ground using an external pushbutton? My thought is that this would be preferable to turning the PET on then off when it crashed--at least it should be easier on the CRT filament.

A. Unless you are debugging a machine language program there is no reason why your PET should crash. There is an updated ROM for the original PET ROM set to prevent the "cursor going away when editing" problem. Other RAM and ROM problems can be cured at your service center to prevent "crashes".

In direct answer to your question, both RESET AND IRQ can be brounded safely at A27 and A28 respectively or the memory expansion connector. The RESET signal is debounced by hardware on the logic board. You may have unpredictable results using IRQ unless you debounce it with a circuit such as shown: 019ROM, price $30.00.



Q. The nature of the PIA/VIA addresses at E81x/ E82x/ E84x make me suspect that these addresses are incompletely decoded; true? If so, is there danger of harm to any of the parts if, for instance, one inadvertently addressed location E703 for a read? I have visions of all three I/O chips trying to have its own way with the data bus and trying to destruct each other, while the 6502 becomes confused.

A. The memory locations for the TV display RAM, both PIA's and the VIA are not completely decoded. The result is many images of these devices but they never overlap so it is not possible to address multiple devices.

The TV RAM occupies $8000 to $83FF and has an image $8400 to $87FF.

The PIA's have base addressed $E81Ø and $E82Ø. The VIA occupies 16 locations starting at $E84Ø. The next images of these devices are at E89Ø, E8AØ, E8CØ.

There is no danger to the PET I/O devices no matter what addresses that the processor reads or writes. Reading location $E7Ø3 will give you a character from the TV screen.

Q. There is a footnote on p. 11 of the first NEWSLETTER "retrofit kit required for operation of the 2040 Dual Disc with PET 2001-8". What could that involve? New ROM's? Additional RAM? Complete new main board? In any case, what does it cost?

A. The retrofit kit to operate the new PET 2040 Dual Floppy Disk on an "old" 8K PET simply involves replacing the existing socketed ROMS with a new set. These new ROMS are included in the $1095.00 price.

To use the disk, no additional RAM is required.

Commodore News

Unfortunately, a few USERS' did not receive an ERRATA sheet upon receipt of our newest USERS' MANUAL. Therefore below is a copy of the ERRATA Sheet, including the latest corrections (*) to date.

PET USER MANUAL MODEL 2001-8 AMENDMENTS

Page 5 ,	para	3:	LOCXATION read LOCATION
Page 11,	para	7:	character READ carriage
Page 13,	para	7:	th READ the
Page 14,	para	5:	i.E. READ i.e.
Page 23,	para	6:	effect if a space READ effect of a space
Page 27,	para	5:	Subtration READ Subtraction
Page 28,	para	6:	90 degree READ 90 degrees
Page 31,	para	1:	lanuage READ language
Page 35,	para	2:	adviseable READ advisable
Page 39,	para	5:	O=X =255 READ 0 =X =255
Page 43,			HTPERBOKIC COSINE read HYPERBOLIC COSINE
Page 44,			HTPERBOLIC SECANT read HYPERBOLIC SECANT
Page 45,			Omit colons (:) from lines 10 and 15 of the last program
Page 49,	para	2:	(9$1)*(8$1) read (9+1)*(8+1)
Page 53,	last para	:	are divide in READ are divided in
Page 65,			OPEN#LF,D,S,FN read OPEN#LF,D,SA,FN
Page 67,	para	2:	desireable READ desirable
	para	5:	adviseable READ advisable
Page 75,	para	5:	meesage READ message
Page 91,	para	1:	adviseable READ advisable
Page 91,	para	5:	temoraril READ temporarily
Page 115,		:	delete first two paragraphs titled: "Out of Data...A READ"
Page 117,	last para	:	only ccurs READ only occurs
Page A2	Heading	:	location not specifies READ locations not specified
Page A3,	last para	:	62 byte on READ 62 bytes on
Page B4	para	6:	IF:alwys READ always
Page B6	para	:	PRINT;nest READ next
Page B8,	para	1:	untill READ until
* Page 93,	para	1:	POKE of the decimal equivalent of the lower address to location 2 SHOULD READ POKE of the decimal equivalent of the lower address to location 1

Software

NEW SOFTWARE

Some of you have shown an interest in the purchase of Software from other Commodore divisions, therefore per your requests we have listed the programs and their current cost. These will be available March 30, 1979. For the descriptions of these programs, please see Volume I, issue 1 of the PUC NEWSLETTER.

PROGRAM NAME	PROGRAM NUMBER	PRICE
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

****

As you probably know, Commodore encourages you, the PET user, to submit programs for possible publication. Understandingly, the response has been tremendous! Occasionally we have encountered quality programs which we feel may be of interest to a select group of users. In an effort to meet your software needs, we would like to make these programs available to you.

WEAVE AND DRAFT

DRAFT: Takes patterns up to 30 threads wide by 24 long and gives draft, if possible, on minimum number of harnesses, (maximum is 8). Draft can be stored and passed to WEAVE.

WEAVE: Takes draft from above or one you give it (up to 30 wide x 24 long) and "weaves" pattern on screen like a sample loom. Requires 8K of RAM.

DYNAMIC JOB SCHEDULER

With this program scheduling and controlling jobs and tasks can be done by your PET. Invaluable to businesses where tight schedules are critical to customer satisfaction. Ideal for personal use, also. Requires 8K of RAM.

CONTACT:

Mr. James A. Fowler: P.O. Box L
Coraway Road
Seauket, New York
11733

Inquires to: Mark Stewart
2805 West 2935 South
Salt Lake City, Utah
84119

Please note that these programs are <u>not</u> available through Commodore. Contact the author at the address given.

SOFTWARE REVIEW

The Basic Math Package — Program Number 32 10 14

The BASIC MATH package is a collection of 6 programs dealing with a variety of computational concepts from geometry, numerical analysis and algebra. The purpose is twofold. The programs can be used as a computational tool in research or they can be used in education. The basic theory is part of the curriculum of a first year calculus class.

PLANE GEOMETRY: The Plane Geometry program is used to solve a triangle from a minimum amount of information. A triangle is said to be solved when the measures of all three angles and the lengths of all three sides are known.

The Plane Geometry program will also compute the areas of any triangle and any regular (all sides are equal) polygon.

SPHERICAL GEOMETRY: The Spherical Geometry program demonstrates the analogies between plane and spherical geometry. Spherical triangles are solved just as in the Plane Geometry program. The program works for any radius or angular measure which may be used for the lengths of sides.

ANALYSIS: The Analysis program computes the integral of discrete data using both the trapezoid rule and Simpson's rule. Integration of continuous functions can be done by entering the function as line 1 of the program. Gaussian 10 point quadrature is used to compute the integral. The program also computes the first, second and third derivatives of discrete points.

MATRIX: Matrix 1 and Matrix 2 are two programs that compute the standard matrix operations: add, subtract, multiply, transpose, determinant, inverse and solution to simultaneous equations.

VECTORS: The Vectors program stores up to six vectors in memory and computes the standard vector operations: add, subtract, magnitude, dot product, cross product and angle between vectors.

The package includes a 72 page instruction manual which describes the concepts used in the programs and contains many sample problems and an extensive list of useful formulas.

FEATURE PROGRAM

Break-Even Analysis (by J. Parsons/C. Westfall)

The following program, Break-Even Analysis, contains several techniques you may find useful. It features the 'form' method of entering and displaying data. For many applications the 'form' technique provides the user with a quick and easy method of entering and editing data. This particular program calculates the gross profit along with one of four variables: fixed cost, variable cost, sale price or number of units.

BREAK-EVEN ANALYSIS

```
5 REM (10-30) INITILIZATION OF PROGRAM
10 TE=3:DIMI%(TE,3),E$(6),K$(TE)
11 E$(0)="■TO■LITTLE■DATA":E$(1)="■TO■MUCH■DATA":E$(2)="■NO■DATA■WAS■SENT"
20 GOSUB1000:W$=CHR$(13)+"■■■"+CHR$(20)+"■■■ "+CHR$(148):Y$="1234567890."
30 OPEN3,3:I=0
35 REM (40-90) CONTROLS INPUT FROM THE FORM
40 A=I%(I,0):B=I%(I,1):C=I%(I,2):D=1:IFD$="■■ ■■"ORD$="■■"THENB=B+C-1:D=C
50 GOSUB2000:K(I)=VAL(S$)
78 IFD$="■■ ■■"ORD$="■■"THEN84
80 IFD$="■■ ■■"ORD$="■■"THEN84
81 FORJ=1TO5:IFMID$(W$,J,1)<>D$THENNEXT
82 ONJGOTO90,84,86,88,89,86
84 I=I-1:IFI<0THENI=TE
85 GOTO40
86 I=I+1:IFI>TETHENI=0
87 GOTO40
88 GOSUB3000
89 I=0:GOTO40
90 GOSUB100:I=0:GOTO40
95 REM (100-160) EDITS INPUT FOR INCORRECT DATA
100 E=6:E$(E)=LEFT$(C$,38):GOSUB900
110 FORJ=0TOTE:T=T*K(J):NEXT:IFT<>0THENE=1:GOTO900
120 FORJ=0TOTE:T=T+K(J):NEXT:IFT=0THENE=2:GOTO900
130 I=0:FORJ=0TOTE:IFK(J)=0THENI=I+1
140 NEXT:IFI>1THENE=0:GOTO900
160 FC=K(0):VC=K(1):SP=K(2):UN=K(3)
190 REM (200-290) PERFORMS CALCULATIONS
200 IFUN=0THENUN=FC/(SP-VC)
210 IFFC=0THENFC=(SP-VC)*UN
220 IFVC=0THENVC=(FC-(UN*SP))/-UN:
230 IFSP=0THENSP=(FC+(UN*VC))/UN
240 GP=UN*(SP-VC)-FC
250 K(0)=FC:K(1)=VC:K(2)=SP:K(3)=UN
260 FORJ=0TOTE:S$=LEFT$(STR$(K(J))+C$,10)
270 PRINTLEFT$(A$,I%(J,0))SPC(I%(J,1))S$:NEXT
280 S$=LEFT$(STR$(GP)+C$,10):PRINTLEFT$(A$,18)SPC(20)"■"S$
290 RETURN
895 REM (900-910) PRINTS ERROR MESSAGES
900 PRINTLEFT$(A$,23)SPC(1)LEFT$(C$,38)
910 PRINTLEFT$(A$,23)SPC(1)E$(E):RETURN
995 REM (1000-1150) PRINTS FORM ON SCREEN
1000 A=38:A$="■/":FORI=1TOA:A$=A$+"—":NEXT:A$=A$+"\"
1010 B$="|":FORI=1TOA:B$=B$+" ":NEXT:B$=B$+"|"
1020 C$="\":FORI=1TOA:C$=C$+"—":NEXT:C$=C$+"/"
1025 PRINT"■";
1030 PRINTA$B$B$B$B$C$A$B$B$B$B$B$B$B$B$B$B$B$C$"■■"MID$(A$,2)B$C$"■■";
1040 A$="■":FORI=1TO25:A$=A$+"■":NEXT
1050 PRINTSPC(3)"■■  ■■■■ ■■ ■■■■■■■■ ■■ ■■■■■■■■■■  "
1060 B$="":FORI=1TO25:B$=B$+"■■":NEXT
```

```
1070 READS$:PRINTSPC(10)S$
1080 C$="":FORI=1TO40:C$=C$+" ":NEXT
1090 FORI=0TOTE:READS$,A,B,C,D,E
1110 PRINTLEFT$(A$,A)SPC(B)S$LEFT$(A$,C)SPC(D)LEFT$(C$,E)
1120 I%(I,0)=C:I%(I,1)=D:I%(I,2)=E:NEXT:K=I%(1,2):RETURN
1130 DATA"■■■BREAK-EVEN■■■■■■■■■■ANALYSIS","■FIXED COST■",10,3,10,20,10
1140 DATA"■VARIABLE COST■",12,3,12,20,10,"■SALE PRICE■",14,3,14,20,10
1150 DATA"■NUMBER OF UNITS■■■■■■■■■■■■■■■■■■GROSS PROFIT■",16,3,16,20,10
1995 REM (2000-2230) EDITS INPUT
2000 GETD$:IFD$<>""THEN2000
2005 PRINTLEFT$(A$,A)SPC(B);:GET#3,T$:PRINT"■■";
2010 PRINT"_■■";:FORT=1TO50:GETD$:IFD$=""THENNEXT
2020 IFD$=""THENPRINTT$"■■";:FORT=1TO50:GETD$:IFD$=""THENNEXT
2030 IFD$=""THEN2010
2040 FORT=1TO11:IFMID$(Y$,T,1)<>D$THENNEXT:GOTO2060
2050 GOTO2110
2060 FORT=1TO10:IFMID$(W$,T,1)<>D$THENNEXT:GOTO2010
2070 IFT<6THEN2200
2080 ONT-5GOTO2090,2100,2110,2110,2085,2010
2085 D=D-1:GOSUB4000:GOTO2110
2090 D$="■■ ■■"
2100 D=D-2:IFD<0THEN2200
2110 PRINTT$"■■"D$;:GET#3,T$:PRINT"■■";:D=D+1:IFD<=CTHENGOTO2010
2200 GET#3,S$:IFS$<>T$THENPRINT"■■"T$;
2210 IFD<1THEND=1
2220 PRINTLEFT$(B$,D);:S$="":FORT=1TOC:GET#3,T$
2230 S$=S$+T$:NEXT:RETURN
2995 REM (3000-3020) CLEARS FIELDS
3000 FORJ=0TOTE:PRINTLEFT$(A$,I%(J,0))SPC(I%(J,1))LEFT$(C$,I%(J,2))
3010 K(J)=0:NEXT
3020 PRINTLEFT$(A$,18)SPC(20)"■"LEFT$(C$,10):RETURN
3995 REM (4000) CLEARS FROM CURSOR TO THE END OF THE FIELD
4000 PRINTLEFT$(C$,I%(I,2)-D)LEFT$(B$,I%(I,2)-D-1);:T$="":RETURN
```

The concept of modular programming has been utilized as follows:

Lines	Function
10-30	Initializes program
40-90	Controls input from form
100-160	Edits input for incorrect data
200-290	Performs calculations
900-910	Prints error messages
1000-1150	Prints form on screen
2000-2230	Edits input
3000-3020	Clears field
4000	Clears from cursor to the end of field

REM statements have been inserted to summerize the function of each module. However, in cases where efficient use of memory is required, you may wish to delete these REM statements.

Initialization of the Program

Variable		Definition
TE	=	Total Entries (number of variables that need to be entered).
I%(I,J)	=	Array in which the field positions and lengths are stored.
E$(I)	=	Error messages
K (I)	=	Entered data

Line number	Function
10	This line sets TE=3 which offers you four entries (0, 1, 2, 3). All arrays and matrices which the program utilizes dimensioned. This conserves memory because all dimensions default to 10 unless they have previously been dimensioned.
20	The program jumps to a subroutine located in lines 1000-1150 which prints the form on the screen. (This will be discussed in greater detail later). The program then initializes two more strings W$ and Y$ for later use. W$ = return + cursor up + cursor down + clear home + cursor home + delete + cursor left+ cursor right + insert. Y$ = "1 2 3 4 5 6 7 8 9 Ø."

Printing the form on the Screen (Subroutine called in line 20)

1000 A is set equal to 38 so that the form utilizes all forty columns of the screen.

A$ is set equal to reverse field on +"ר" +(38"-"ʻs) + "ך"

1010 B$ is set equal to "ו" + 38 spaces + "ו"

1020 C$ is set equal to "ט" +(38 "-"ʻs) + "י"

1025 Clears the screen to prepare for the printing of the form.

1030 The first 20 strings (combinations of A$, B$ and C$), print the two reverse field rectangles and their borders. Then a reverse field off and cursor down are printed followed by strings A$, B$ and C$ not in reverse field. The cursor is then moved home and down one row.

Note: The MID$ function is defined with three subscripts. However, MID$ (X$,A) defaults to the string X$ starting with the Ath character. This is useful when you do not know the length of the string and you wish to call the right most portion of a string (the right function becomes cumbersome in this case).

In the next five lines A$, B$ and C$ are reset. The reutilization of these variables helps to conserve memory.

1040 A$ is set equal to cursor home + 25 cursor down's.

1050 Prints the Commodore logo in the upper left-hand corner

1060 B$ is reset to a string of 25 cursor left's.

1070 Reads the first piece of data located in line 1130 as S$. (So S$ = cursor up + cursor up + reverse field on + BREAK-EVEN + cursor down + 9 cursor left's + ANALYSIS").

S$ is then printed, centered on the top of the screen.

1080 C$ is set equal to 40 spaces.

The next three lines contain a FOR NEXT loop which reads in the rest of the data contained in lines 1130-1150.

S$ = field title
A = field title row position
B = field title column position
C = field row position
D = field column position
E = field length

1090 Reads the data S$, A, B, C, D, E

1110 Prints the field title and the spaces for the input (referred to as the 'field').

1120 Stores the field row position, column position and length in the matrix I% (I,J). This information will be needed later in order to receive the input from the screen.

	fixed cost	variable cost	sale price	number of units
J \ I	0	1	2	3
0 Field row Position	10	12	14	16
1 Field column Position	20	20	20	20
2 Field Length	10	10	10	10

1130-1150 data statements called in line 1070 and 1090.

Subsequent issues of this PET NEWSLETTER will contain a similar description of the remaining routines within this program.

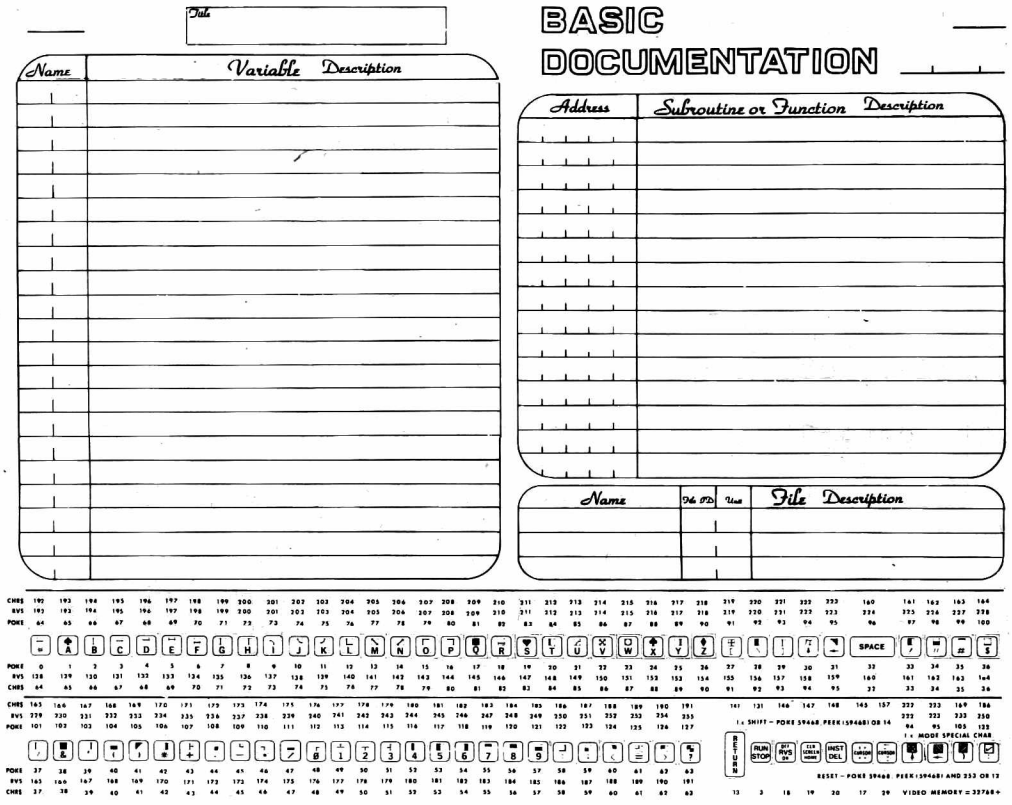
Below is a form which we feel is excellent for recording programs and documentation.
To order write:

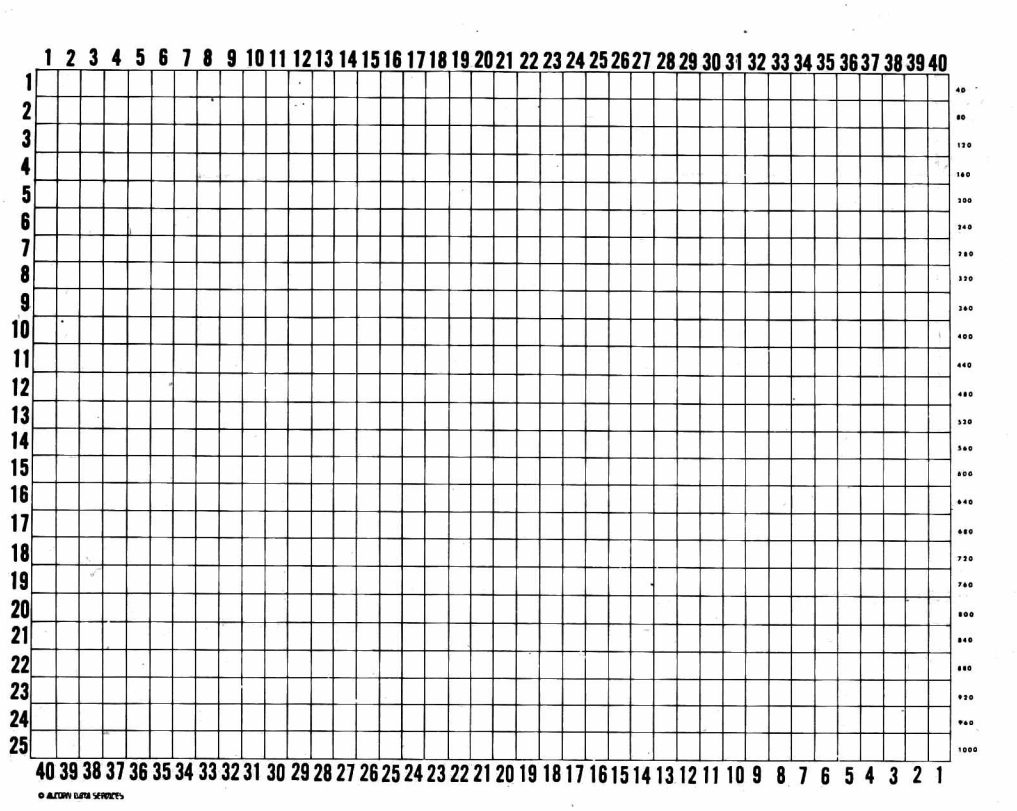
ALCORN DATA SERVICES
P.O. BOX 1535
CORINTH, MISSISSIPPI 38834

Prices:

Pads/aprox. 50 sheets $2.50 per pad
5 Pads/ $7.50

Postpaid





Peripherals & Attachments

This section will list and preview known PET compatible PERIPHERALS and ATTACHMENTS. In addition to standard or commercially available equiptment, we would be interested in any new and unusual hardware interfacing which you may come across.

PERIPHERALS

PET PRINTER PREVIEW

Two of the latest additions to the Commodore line of intelligent IEEE peripherals are the 2022 and 2023 Printers. The 2022 is an 80-column tractor-feed printer and the 2023 is an 80-column friction-feed printer.

Each printer contains a 6504 microprocessor, (smaller cousin of the 6502), and has a program in ROM which enables it to process two classes of data received over the IEEE bus. Special bus commands can be used to differentiate between data which is to be printed and data which is to be used in constructing the print format. Additionally, control characters permit printing of enhanced (double wide characters), and reverse field (white on black).

Basic programs can be listed from a PET just as they would appear on the CRT. There is even an auto line counting mode which spaces to the top of a form after a programmed number of lines.

Furthermore, the 2022 has a variable number of steps between lines. With this feature, continuous graphics can be formed in the vertical axis. This feature can also be used to set up popular 6 lines/inch or 8 lines/inch spacing.

Both the 2022 and 2023 Printers have a built-in diagnostic capability which can be activated even without a PET attached.

****

ATTACHMENTS

Even though the following program has recently been published in other Newsletters, we feel because of its' excellence, it merits republication.

PET, AND THE DUAL JOYSTICKS
by
CHUCK JOHNSON, SPHINX

The joystick manufactured by Atari® for their Home Video Arcade offers a very simple and superbly versatile remote input device for the PET and other micros with a parallel input port. In this article the author will propose an interface which accomodates two Atari joysticks and requires a minimum of electronic parts. A general-purpose subroutine allows the user to fully interpret joystick and button movements and includes ideas for more specialized applications.

WHAT'S A JOYSTICK?

The joystick concept evolved with the technology of airplane maneuvering. From the earliest controlled aircraft (somewhat post-Wright brothers) through the Viper fighter of Battlestar Galactica, the "broomstick-on-the-floor" has been the most popular steering device. Joysticks for video games and home computers are identical in concept and fall into two functional categories:

1. POTENTIOMETER - movement of the joystick turns two variable resistors; one senses up-down movement, the other senses left-right movement. Decoding the joystick position requires a resistance or voltage measurement to be interpreted by the game or computer.
2. SWITCH - Pushing the joystick in one direction closes a simple (SPST) switch. The game or computer senses the joystick position by checking the status (open or closed) of four switches - one each up, down, left, and right.

The Atari joystick is a switch-type device, with an additional push-button switch. This mechanism very easily attaches to the PET's parallel port (User Port), but...

WHAT'S A USER PORT?

The PET has several attachments for accessories on the back. One of these slots is the parallel input/output port, or User Port. In general, the User Port can send or receive eight electronic signals which are "high" or "low," according to TTL convention. (These signals may also be humanly interpreted as on/off, yes/no, or five volts/zero volts.) We will be using the User Port for input and in that mode all eight signals are normally "high;" our joystick(s) switches will force some of the signals to their "low" condition. We can pull any one of the signals "low" by attaching it to the "ground" line of the User Port. The switches in the joysticks will be wired to close the circuit between the "ground" line and one of the data bits.

HERE'S THE PLAN

Each joystick has four position switches and one push-button switch. We'll attach the position switches so that each one closes the circuit to one of the Port data lines. We'll wire the two push-button switches so that each one closes two circuits, instead of one, and pulls two data lines "low". This layout accommodates the full capabilities of two Atari joysticks and is very simple to physically assemble.

THAT'S GREAT, BUT HOW DOES A PET PROGRAM USE IT?

Conveniently, the PET looks at the User Port as just another eight-bit memory location. (Aha! You thought eight data lines were mere coincidence!) PEEKing at memory location 59471 reveals the status of the data lines as a binary number (a decimal range of 0 to 255). If all switches are open, then all lines are "high" and 59741 contains 11111111 (binary) or 255 (decimal). When a switch is closed, the bit corresponding to that line turns to a zero. If a joystick switch closes the first circuit, then 59741 would contain 11111110 (binary) or 254. If two switches on one joystick are closed (the push-button, for example), then 59741 would contain a number with two zero bits, say 11110011 (binary) or 243.

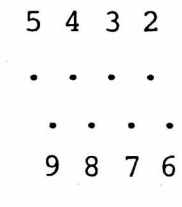
Once a program has performed these decoding steps, it can interpret the meaning of the various switch closures and proceed accordingly.

OK, LET'S PUT IT TOGETHER

To implement the two joystick interface we must establish an attachment standard and a decoding standard. The attachment standard provides the external connection and the physical means for inputting switch closures. The decoding standard is a BASIC subroutine which interprets the meanings of joystick switch closures for a program in execution.

JOYSTICK ATTACHMENT STANDARD

The Atari Joystick terminates with a DB-9S connector; six of the nine circuits are used as follows:



DB-9PIN	FUNCTION
1	Up Switch
2	Down Switch
3	Left Switch
4	Right Switch
5	Not Used
6	Push-Button Switch
7	Not Used
8	Ground (common)
9	Not Used

A DB-25P connector will accept two joystick connectors; the DB-25 may then be wired to a User Port edge connector, according to the following table:

	JOYSTICK	DB-9S PIN	DB-25P PIN	USER PORT PIN	USER PORT FUNCTION
1.	UP	1	9	F	PA3
	DOWN	2	10	E	PA2
	LEFT	3	11	D	PA1
	RIGHT	4	12	C	PA0
	BUTTON	6	22	Diodes to E and F	PA2 and PA3
	GROUND	8	24	A	GROUND
2.	UP	1	1	L	PA7
	DOWN	2	2	K	PA6
	LEFT	3	3	J	PA5
	RIGHT	4	4	H	PA4
	BUTTON	6	14	Diodes to K and L	PA6 and PA7
	GROUND	8	16	N	GROUND

Pin numbers for the DB-25P are marked on the connector. User Port pins are on the bottom row of a 12 position edge connector. Proper polarization of the edge connector is strongly recommended.

Note that the buttons are wired through diodes to both the UP and DOWN functions of their joysticks (pins E and F and pins K & L). When a button is pressed, the PET data lines react as though the joysticks were pushed up and down simultaneously (an impossible condition for the position switches to generate). The decoding standard will, as we shall see, interpret this condition as a button movement. The diodes act as "one-way streets" for current flow and prevent the real UP and DOWN switches from closing both circuits.

Orientation of the diodes is very important. We want current to flow from the data lines to ground. When soldering the diodes in place, orient them with their cathodes ("banded" ends) connected to the joystick buttons (pins 14 and 22 on the DB-25P). If the user is not interested in using the buttons, the button lines may be left unconnected and the diodes omitted.

SOFTWARE DECODING STANDARD

To a great extent, the software to interpret joystick switch closures may be specifically written for each different application. The BASIC subroutine in this section is meant as a general purpose decoder for two joysticks and is by no means the most efficient (i.e. fastest). It performs all of the necessary tasks, however, and should be used as a guideline.

Here's the routine to decode five switches on two joysticks:

```
100 REM INITIALIZE THE LOOKUP TABLE
110 DIM JS (15)
120 FOR I = 0 TO 15
130 READ JS (I)
140 NEXT I
150 DATA 99, 99, 99, 0, 99, 7, 9, 8
160 DATA 99, 1, 3, 2, 99, 4, 6, 5
 .
 .
 .
```

```
19999 REM POLL TWO JOYSTICKS
20000 J1 = JS (PEEK(59471) AND 15)
20010 J2 = JS ((.0625*PEEK(59471)) AND 15)
20100 RETURN
```

The routine returns "lookup table" values in J1 and J2 for joysticks 1 and 2 respectively. The calling program maybe written to take appropriate action based on these values. Readers familiar with User Port control will recognize memory location 56741, as Port A Data without handshake.

This subroutine uses four bits (PAO-PA3 or PA4-PA7) in decimal form (range 0 to 15) as an index to the "lookup table" in array JS. The values in the JS array correspond to the following joystick functions:

DECIMAL INDEX	BINARY	JOYSTICK	EXAMPLE VALUE
0	0000	--	99
1	0001	--	99
2	0010	--	99
3	0011	BUTTON	0
4	0100	--	99
5	0101	UP-LEFT	7
6	0110	UP-RIGHT	9
7	0111	UP	8
8	1000	--	99
9	1001	DOWN-LEFT	1
10	1010	DOWN-RIGHT	3
11	1011	DOWN	2
12	1100	--	99
13	1101	LEFT	4
14	1110	RIGHT	6
15	1111	CENTERED	5

The "example values" relate to the key layout on the PET numeric pad, i.e. joystick centered is "5", joystick UP and LEFT is "7", et cetera. Recall that the BUTTON entry is analogous to UP and DOWN switch closures. Impossible conditions are flagged with "99". Other tables may be developed to suit the user's needs.

These modifications are reasonable for more specialized applications:

1. To use joystick 1 only, delete line 20010.
2. To use both joysticks without their pushbutton functions, change the fourth entry in the lookup table from "0" to "99" (line 150).
3. Make use of ON-GO TO statement.
4. Recode the routine in machine language.

SOUNDS LIKE FUN...WHAT DO I NEED?

The hardware for this project consists of two circuit connectors and four diodes, in addition to one or two Atari joysticks. The joysticks plug into the DB-25P subminiature connector, manufactured by ITT Cannon and several competitors. The User Port connector is a 12 position, dual readout edge connector, widely available. The suggested diode is a 1N914, which is a commonly used fast switching device.

Actually any flea power diodes should work. Assembly is quick and requires soldering wires and diodes between the DB-25P and the edge connector; refer to the attachment standard. Be sure that the diodes are oriented correctly. Improper orientation will not damage the PET, but the joystick buttons won't respond as planned. Plug the edge connector onto the back of your PET (it's the center slot); be sure it's properly oriented. Plug your joysticks into the DB-25P. Turn the PET on and you're off and running!

As a service to interested joystick-ers who lack a source for parts, these items are available from the author:

User Port connector with polarizing keys	$ 3.50
DB-25P wired for 2 joysticks (4 diodes included)	3.50
Atari joystick	15.00
SNAKE program on cassette	2.00
Two joystick, both connectors, four diodes and SNAKE cassette	36.00

The SNAKE program was originally written by Pete Rowe of the Lawrence Hall of Science in Berkeley, and couldn't be better suited for two joysticks. The author has modified the SNAKE coding for joysticks and is distributing it with the permission of Pete Rowe and the SPHINX user group library.

REBUTTAL AND DISCUSSION

...are strongly encouraged. All suggestions should be publicly aired. The author will gladly receive comments and/or parts requests over the telephone (415/278-6595 at reasonable PST hours), at SPHINX meetings, or by mail:

Chuck Johnson
17104 Via Alamitos
San Lorenzo, CA 94580

Applications

We will reprint Issue 1's APPLICATION format until your submissions have been received and reviewed.

There are almost as many applications for a PET as there are PET's themselves. We would like to find out what the PET is being used for so that we can pass on relevent details to other people interested in the same area of use.

We are therefore offering $50.00 worth of free Master Library Software to the best 'Applications" article published in each issue. If you would like to write about what you are doing with your PET, please include the following details:

1. What the nature of the application is.
2. What (if applicable) non-computerized system has the PET replaced.
3. Details of any extra hardware used.
4. Any "special features" of the programs used.
5. Who else this system might be useful to.
6. Any further improvements/modifications intended.

Programming

This section will be dealing with useful routines and "tricks" for using on your PET. Some articles come from users, some from ouselves, and there is a " Hints and Tips" section for smaller yet valuable items.

FORMATTING ROUTINE

(by J. Parsons/ C. Westfall)

Many business applications programs require the display of numberical tables. PET offers the SPC and TAB functions to allow you to easily print the numbers in columns which are left justified. However, it is often desirable to right justify the columns, line up the decimals and/or have the numbers consistently truncated to the same precision.

The following routine allows you to emulate a 'format' statement. This program enables you to print columns of numbers with the decimal points aligned. It also unifies the precision with which each number is displayed. This routine is ideal for business and finance applications.

```
20   C$ = "          "
25   DEF FNA(X) = INT(ABS(X))
26   DEF FNB(X) = (ABS(X) - FNA(X) + 1.0000001
100  INPUT "          "   ; X $:  X = VAL(X$): GO SUB
     600: PRINT X, RIGHT$(C$+X$,10): GO TO 100
600  X$ = "$ " + MID$(STR$(FNA(X)),2) +
     MID$(STR$(FNB(X)), 3, 3) + LEFT$(STR$(X),1)
605  RETURN
```

KEY CODES: ␣ = Space
Shifted Space
← = Cursor Left

VARIABLE NAME		DEFINITION
X$	=	The number the user inputs
X	=	Numerical value of X$ (i.e. x = VAL(X$))
FNA(X)	=	The integer portion of X(the number entered)
FNB(X)	=	The absolute value of the decimal portion of X(the number entered) plus one.
C$	=	10 spaces

PROGRAM EXPLANATION

10 C$ is set up to be a field of ten spaces. C$'s length should be equal to the width of your field.

Example:

If X$ only requires six columns, then C$ would equal six spaces.

25 FNA(X) is defined to be the absolute value of the integer portion of X. The INT function always rounds down. Therefore, INT (-1.9) would be -2. In order to extract the numeral 1, it becomes necessary to take the integer (INT) of the absolute value (ABS) of -1.9.

```
INT  (ABS(-1.9))
INT (1.9)
= 1
```

26 FNB(X) extracts the absolute value of the decimal portion of X. 1.0000001 is added in to the decimal portion of X. By adding 1 (one), the PET will not convert FNB(X) to floating point form. To add trailing zeros to an integer or number of lesser precision than desired .0000001 is added.

100 A protected input (as discussed on page 20 of Volume 1, Issue 1 of this Newsletter is used to enter the number. X is assigned the numerical value of X$ (the number entered). A subroutine is then called which concatenates the appropriate strings. The concatenation is placed in a subroutine so that it can be accessed from anywhere within a program. The program then prints your original number and the formated number. The comma starts the printing in the 11th screen column. Only the 10 rightmost characters of C$ + X$ are printed to ensure you of a 10 character field. In reviewing the code: RIGHT$ (C$+X$,10), "10" indicates the total number of characters in the field.

600 Contains the subroutine which constructs the desired string to be printed

X$ = "$Δ"

plus MID$(STR$(FNA(X)),2) - places the integer portion of X in string form.

plus MID$(STR$(FNB(X)),3,3) - extracts the decimal portion of X along with the decimal point.

The last subscript in the MID$ controls the precision of your output.

Example:

2 rounds to the nearest .1
3 rounds to the nearest .01
4 rounds to the nearest .001

plus LEFT $(STR$(X),1) - the sign of X (positive or negative). A space is printed for a positive X and a negative sign (-) for a negative X.

In this particular program the sign is printed to the right of each number. This aligns the signs to be printed in the same column. The format to be used is only a matter of preference. This can easily be changed by rearranging and modifying the strings in line 600.

605 Returns control of the program back to line 100 (in this case).

BITS AND PIECES

MEMORY LOCATION

When programming in machine language, occassionally it becomes useful to determine the location at which a specific variable is stored. The following routine developed by David M. Duncan of San Francisco, Calif., gives both the value of a variable (V) and the decimal location of it's first character.

```
100  V=2001.8
110  VT=V
120  V=PEEK( 150) +256 * PEEK (151)
130  ML = V
140  V=VT
150  PRINT "V=";V"MEMORY LOCATION IS"; ML
```

****

Simple Non-Stop Programs

(by J. Feagans)

By Poking the RAM interrupt vector to a valve 3 greater than the original, one is able to prevent the stop key action but still allow full keyboard scan and INPUT operation.

Location 537 in the 8K PET contains a 133.

```
POKE 537, 136
```

causes the software to avoid a subrouting call which updates the clock register and checks for the stop key. If the stop key is pressed, the GET command will return a CHR$ (3).

```
10  GET   A$:  if A$ = CHR$ (3) then end
20  GO TO 1Ø
```

Caution: The IRQ RAM vector will be reset if (1) the PET is reset, (2) a cassette is loaded or saved. Furthermore, keep in mind that TI and TI$ no longer keep the time of day.

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 here are PET users who have submitted their specialty or area of interest to further the communication of ideas with PET owners throughout the United States. If you would like to contribute, please fill out the form which follows.

NAME AND ADDRESS	SERVICES OFFERED/SPECIALTIES
Michael Baltay 9900 Milburn Drive Sun Valley, Calif. 91352	Structural Design, Earthquake, Mathematics.
Joseph Banasiak, P.E. 415 Nutt Road Phoenixville, Pa. 19460	EMC Engineering Programs and Analysis. Also would like to form local-user group.
Larry Hauder 4810 East Firestone Blvd. Southgate, Ca. 90280	Software--tutorial.
Capitol Area PET Enthusiasts (CAPE) Robert Karpen (Coordinator) 2054 Eakins Court Reston, Va. 22091	Monthly Meetings on 2nd Saturdays at PATRICK HENRY LIBRARY, 101 Maple Ave., Vienna, Va., at 1:30 p.m.
Dennis Costarakis 3562 Union Street San Diego, Ca. 92103	Securities, investments, stocks, options, bonds, convertibles and hedging techniques.
Charles E. H. Edward 3300 Wake Drive Kensington, Md. 20795	Computer aided engineering design using the PET.
H. A. Elkholy Fairleigh Dickinson University Madison, New Jersey 07940	Education.
Ben Firschein 29 Stowe Lane Menlo Park, Ca. 94025	Personal tutoring in PET programming, writing games and graphics software.
Ron Gutman 13620 Ferncrest Court Saratoga, Ca. 95070	Experienced microprocessor programmer will develop software in basic and/or 6502 assembly. Currently developing PET music system.

Donald F. Hemenway 7805 Klovstad Dr. Oxon Hill, Md. 20022	Tutoring, new owners/radar systems design/personal finance.
JAPS-Jacksonville Area PET Society c/o R. Crockett 401 Monument Road #177 Jax, Florida 32211	PET library. 50+ programs available $2.50 first program tape, $1.50 each additional. SASE for list.
Chuck Johnson 17104 Via Alamitos San Lorenzo, Ca. 94580	Joysticks, Parallel Port, Assembly Language.
Milt Lee 1348 Rudgear Road Walnut Creek, Ca. 94596	PET as a data logger. Telephone modem.
Mark Pleticha P.O. Box 77562 Station 'E' San Francisco, Ca. 94107	"Contract Programming"
Society of Computer Owners and PET Enthusiasts (SCOPE) 1020 Summit Circle Carrollton, Texas 75006	Specialists in every aspect of the PET are available.
Mark Zimmerman Caltech 130-33 Pasadena, Ca. 91125	Would like to help anybody who has questions concerning physics, mathematics, games, FORTH, Assembly language.

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, Maryland 20016

Amateur Computer Group of New Jersey
Box 379,.....South Bound Brook, NJ 0880

Bambug
1450 53rd St..... Emeryville, CA

Lawrence Hall of Science, UC Berekeley
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

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

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

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

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

PET User Group
Texas A & M Microcomputer Club.....Texas A & M Tx.

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

PUG
310 Showers Dr.....Mountain View, CA

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

SCOPE
1020 Summit Circle.....Carrollton, TX 75006

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

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

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

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

COMMODORE BUSINESS MACHINES LIMITED
3370 PHARMACY AVENUE
AGINCOURT, ONTARIO, CANADA M1W2K4

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

COMMODORE BUROMASCHINEN 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 A02