PET USERS GROUP NEWSLETTER

VOLUME 0 NUMBER 4

T-A-B-L-E O-F C-O-N-T-E-N-T-S

Check page 16 for call to meetings!

1.

F-O-R-T-H F-O-R T-H-E P-E-T

by Neil Bussey

I have been very interested in all I could learn about any implementations of Forth for the PET. I recently acquired one implementation, and as a result of some hands-on experience, have formed the opinion that it is the language for the micro computer, particularly from the hobbyist viewpoint.

Forth has been described as like having an HP calculator with a very large stack and an infinite number of keys. It implements on a micro in 6K and includes with in its holistically integrated architecture, facilities found in operating systems, assemblers, compilers, and utilities such as editors and debugging tools.

Melvin Norrel's implementation differs from the concepts of the Forth Interest Group in that his library is self linked in compiled Forth and will run with less overhead than the acknowledged 70 to 100 percent that normal Forth has when running under its outer interpreter. And even more, it can run faster than an equivalent assembler coded program might, in that its discipline forces the programmer to write structured code.

I am very excited by this language, and particularly by PETFORTH from Programma Consultants, primarily because it works! And also because Melvin intends to support it. He backed that contention up by including a "Suspected Error Form" appended to his preliminary documentation.

PETFORTH can be obtained from: Programma Consultants
3400 Wilshire Boulevard
Los Angeles, CA 90010
Price = $35.00

Small world

by Philip Stein

The FORTH dimension: mini language has many faces

Suppose someone came to you with a great new computer language, *Basic*. You had never heard of it before, and had always programmed in *Fortran*. After studying the newcomer, you decided not to convert despite the considerable advantages it presented—because it was different, required retraining, learning new skills, reprogramming, and besides, the old way worked fine. If we all reacted that way, there would be no *Basic*. What would you have done?

Such is the case with *FORTH*. *FORTH* is a language originally developed for use with scientific-type minicomputers used to operate telescopes and to process the data they gather for astronomers. It is now being developed and sold by a private company and is available to run on most minicomputers. *FORTH* is different. It defies simple description because it cannot be easily compared to *Fortran* or *Basic*. It serves as operating system, compiler, assembler, interpreter, virtual memory, file system, etc. all rolled into one without being identifiably any of these. As a natural result, it is very foreign to most minicomputer users, even to professional programmers. Does this mean that *FORTH* will have a hard time being accepted?

2.

Forthright style

FORTH uses concepts and techniques in software engineering such as virtual memory and stack organization which have been around for a long time but have not yet been well implemented or accepted in the minicomputer business. This means that a lot of nice benefits, such as structured, re-entrant programs, are a natural consequence of writing in *FORTH*, not so with other languages.

The key to the use of *FORTH* is a vocabulary. Some of the words are defined by you, the user. Others are supplied with the language—rather, they are the language. A *FORTH* program consists entirely of vocabulary entries, each one stated in terms of other words already defined. To run a "program," you type a word on the computer console and that word is executed. Each definition is therefore separately testable by typing its name, whether it is a simple order or a large subroutine. This capability, coupled with an IF-THEN-ELSE conditional, makes all programs naturally "structured." Writing a *FORTH* program is done by defining a series of words. That's all.

This means that *FORTH* users can define their own set of applications-dependent words, and can easily teach the use of the applications-dependent vocabularly to non-programmers. In that sense it's a good systems-building tool. It's also a good meta-language or macro-language. You can reprogram existing special-purpose languages.

FORTH definitions are compiled at the time that they are defined. This means that programs run quite fast and can even compare favorably to hand-coded assembly programs. The inherently nested structure of the program forces the programmer to make good use of subroutines—usually better than he would have done in other languages—thus getting faster, more compact results. *FORTH* is very small, too. It only takes 3 or 4k in a 16-bit machine. This is partly because it is, itself, written in *FORTH*, and partly because it is compactly structured and not encyclopedic in its ambition.

Nesting assembler programs

Perhaps the most powerful feature is its ability to nest assembler programs right into the definitions. Other languages which allow this flexibility often wind up with the feature unused. But not *FORTH*. You can make your own high-level definitions (vocabularly entries) from machine code, then just use the definitions. A driver for a complicated peripheral such as a magnetic tape drive could easily be written in an afternoon—using about twenty lines of *FORTH*. Because programs are naturally re-entrant, a multi-user system is an almost automatic consequence. Vocabularies, both user-written and *FORTH*-written can be partitioned so that they are partly shared among all users and partly private. This saves lots of space since each user will have mostly shared definitions.

Because *FORTH* is not familiar to most users, you will probably have a hard time fully appreciating how to do things in it. Retraining staff is hard, too. *FORTH* is not designed to work in a conventional *OS* environment. Although some compromise could be worked out so that *FORTH* could share room with other types of programs, its speed and size advantages would be lost. *FORTH* and its proponents like to work in fixed-point integer arithmetic, trading off range for improved speed and accuracy. While most *FORTH* users are accustomed to floating point, it represents another retreading for users' thought processes. Floating point can be done, of course; it's just another compromise which takes away some of the benefits of the language.

FORTH, various scientific and business libraries, applications assistance, and full systems are available from *FORTH*, Inc., 815 Manhattan Ave., Manhattan Beach, CA 90266. □

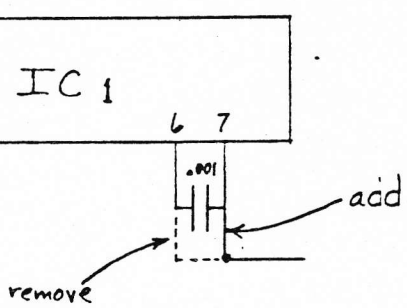
Reprinted from COMPUTER DECISIONS, November 1975, page 10,

C-O-R-R-E-C-T-I-O-N-S

In Issue 1, under PET Stop Disable, page 10, the Commodore Disable Stop 1.3 routine omitted some instructions. The following routine should be used to replace all of routine starting at line 10000:

```
10000 REM SUBROUTINE TO LOAD (NOT EXECUTE) STOP DISABLE/ENABLE
10010 FOR I=832 TO 874: READ W: POKE I,W: NEXT I: RETURN
10020 DATA 120,169,96,141,25,2,169,3
10030 DATA 141,26,2,88,96,0,0,0
10040 DATA 120,169,133,141,25,2,169,230
10050 DATA 141,26,2,88,96,0,0,0
10060 DATA 32,234,255,169,255,141,9,2
10070 DATA 76,136,230,0
```

In Issue 3, page 7, PET Video Mixer, IC#3 should be a 7416 to create a negative pulse. And the schematic at IC#1 should look like this:



N-E-W-S-L-E-T-T-E-R S-U-B-S-C-R-I-P-T-I-O-N

This is the second issue of a PET NEWSLETTER subscription.

A subscription to the PET NEWSLETTER is defined as a six and only six issues commencing with Vol.0, No.3 and terminating with issue 8, the last issue of volume 0, regardless of when you place your subscription. Issues 0, 1 and 2 are back issues and are not included in the subsription.

PET USERS' GROUP NEWSLETTER SUBSCRIPTION AND BACK ISSUE ORDER FORM

Please send me:

( )	Six-issue subscription to PET Newsletter	$4.50
( )	Back Issue, Volume 0, Number 0	.75
( )	Back Issue, Volume 0, Number 1	.75
( )	Back Issue, Volume 0, Number 2	.75

Your Name _____
Address _____
City/State _____ ZIP _____

Send order to: PET Newsletter / Computer Project
Lawrence Hall of Science
University of California
Berkeley, CA 94720

MAKE ALL CHECKS PAYABLE TO: REGENTS OF THE UNIV.OF CALIF.

I-M-A-G-E

by Pete Rowe

In the PET Memory Map in this issue is a location $F6B1 that saves "contiguous bytes of memory onto cassette". Before calling this routine, the user must initialize the following registers:

$F9 - FA	Address of ASCII filename
$EE	Number of characters in filename
$F1	Device number (1=tape#1, 2=tape#2)
$F7 - F8	Starting address
$E5 - E6	Ending address
($F9,$FA)	ASCII filename

A user could POKE values into these registers. However, the following assembly routine prompts the user for the values and stores them in the correct registers before calling $F6B1.

This program can be loaded using TIM or LO- or HI-MONDIS into cassette buffer #2. To execute the program, type SYS(826) from BASIC. The program name should be less than 17 characters, tape unit can be 1 or 2 (or any device on the IEEE), starting address in hex, and ending address in hex plus one. IMAGE's starting and ending addresses are $33A and $3F9. Remember to add one to the ending address. A standard BASIC LOAD will load IMAGE from the cassette into cassette buffer #2 while only altering the pointers $7C-7D, $7E-7F, and $80-81. Typing NEW or LOADing a BASIC program will restore these pointers.

```
033A:  A9 C6     LDA  =$C6
033C:  A0 03     LDY  =$03     Load A,Y with address of "PROGRAM NAME?"
033E:  20 27 CA  JSR  $CA27    Output
0341:  A9 00     LDA  =$00
0343:  85 EE     STA  $EE      Zero $EE
0345:  85 FA     STA  $FA
0347:  A9 0A     LDA  =$0A
0349:  85 F9     STA  $F9      Filename will be at $000A
034B:  20 BB 03  JSR  $03BB    Get a character
034E:  F0 08     BEQ  $0358    Exit if <CR>
0350:  A6 EE     LDX  $EE
0352:  95 0A     STA  $0A,X    Offset store it into $000A
0354:  E6 EE     INC  $EE      Increment character count
0356:  D0 F3     BNE  $034B    Repeat
0358:  A9 D7     LDA  =$D7
035A:  A0 03     LDY  =$03     Load A,Y with address of "TAPE UNIT?"
035C:  20 27 CA  JSR  $CA27    Output
035F:  20 97 03  JSR  $0397    Get a number
0362:  A5 B1     LDA  $B1
0364:  85 F1     STA  $F1      Store it into $F1
0366:  A9 E3     LDA  =$E3
0368:  A0 03     LDY  =$03     "START" address into A,Y
036A:  20 27 CA  JSR  $CA27    Output
036D:  A9 EC     LDA  =$EC
036F:  A0 03     LDY  =$03     "ING ADDRESS?" address into A,Y
0371:  20 27 CA  JSR  $CA27    Output
0374:  20 97 03  JSR  $0397    Get a number
0377:  A5 B1     LDA  $B1
0379:  85 F7     STA  $F7
037B:  A5 B2     LDA  $B2
037D:  85 F8     STA  $F8      Store number into $F7,F8
037F:  A9 E9     LDA  =$E9
0381:  A0 03     LDY  =$03     "ENDING ADDRESS?" address into A,Y
0383:  20 27 CA  JSR  $CA27    Output
0386:  20 97 03  JSR  $0397    Get a number
0389:  A5 B1     LDA  $B1
038B:  85 E5     STA  $E5
038D:  A5 B2     LDA  $B2
038F:  85 E6     STA  $E6      Store number into $E6,E7
0391:  4C B1 F6  JMP  $F6B1    Call SAVE.  (RTS from SAVE to BASIC)

0394:  EA        NOP
0395:  EA        NOP
0396:  EA        NOP
                               INPUT A HEX NUMBER
0397:  A9 00     LDA  =$00
0399:  85 B1     STA  $B1
039B:  85 B2     STA  $B2      Zero $B1,B2
039D:  20 BB 03  JSR  $03BB    Get a character
03A0:  F0 23     BEQ  $03C5    RTS if <CR>
03A2:  C9 40     CMP  =$40
03A4:  90 02     BCC  $03A8
03A6:  69 08     ADC  =$08     ADD 8 if a letter
03A8:  29 0F     AND  =$0F     Mask to lower 4 bits
03AA:  0A        ASL           Rotate into high 4 bits
03AB:  0A        ASL
03AC:  0A        ASL
03AD:  0A        ASL
03AE:  A2 04     LDX  =$04     Rotate into $B1,B2
03B0:  0A        ASL
03B1:  26 B1     ROL  $B1
03B3:  26 B2     ROL  $B2
03B5:  CA        DEX
03B6:  D0 F8     BNE  $03B0
03B8:  4C 9D 03  JMP  $039D    Repeat until <CR>
                               GET A CHARACTER
03BB:  20 E4 FF  JSR  $FFE4    Call GET
03BE:  F0 FB     BEQ  $03BB
03C0:  20 D2 FF  JSR  $FFD2    Echo it on screen
03C3:  C9 0D     CMP  =$0D     Set flag if <CR>
03C5:  60        RTS

03C6:  0D 11 50  <CR><LF>PROGRAM NAME? <00>
03C9:  52 4F 47
03CC:  52 41 4D
03CF:  20 4E 41
03D2:  4D 45 3F
03D5:  20 00
03D7:  54 41 50  TAPE UNIT? <00>
03DA:  45 20 55
03DD:  4E 49 54
03E0:  3F 20 00
03E3:  53 54 41  START<00>
03E6:  52 54 00
03E9:  45 4E 44  ENDING ADDRESS? <00>
03EC:  49 4E 47
03EF:  20 41 44
03F2:  44 52 45
03F5:  53 53 3F
03F8:  20 00
```

HEX DUMP of IMAGE

```
033A: A9 C6 A0 03 20 27 CA A9
0342: 00 85 EE 85 FA A9 0A 85
034A: F9 20 BB 03 F0 08 A6 EE
0352: 95 0A E6 EE D0 F3 A9 D7
035A: A0 03 20 27 CA 20 97 03
0362: A5 B1 85 F1 A9 E3 A0 03
036A: 20 27 CA A9 EC A0 03 20
0372: 27 CA 20 97 03 A5 B1 85
037A: F7 A5 B2 85 F8 A9 E9 A0
0382: 03 20 27 CA 20 97 03 A5
038A: B1 85 E5 A5 B2 S5 E6 4C
0392: B1 F6 EA EA EA A9 00 85
039A: B1 85 B2 20 BB 03 F0 23
03A2: C9 40 90 02 69 08 29 0F
03AA: 0A 0A 0A 0A A2 04 0A 26
03B2: B1 26 B2 CA D0 F8 4C 9D
03BA: 03 20 E4 FF F0 FB 20 D2
03C2: FF C9 0D 60 0D 11 50 52
03CA: 4F 47 52 41 4D 20 4E 41
03D2: 4D 45 3F 20 00 54 41 50
03DA: 45 20 55 4E 49 54 3F 20
03E2: 00 53 54 41 52 54 00 45
03EA: 4E 44 49 4E 47 20 41 44
03F2: 44 52 45 53 53 3F 20 00
```

PET MEMORY MAP

Decimal	Hex	Description
0-2	0000-0002	Jump, User address (lo,hi)
3	0003	Current input device number (0=keyboard)
4	0004	Number of nulls after CR/LF (0=normal)
5	0005	POS of current line (0-255)
6-7	0006-0007	Unused
8-9	0008-0009	Pointer to argument of integer function
10-90	000A-005A	Basic input buffer
91	005B	Used for Quote mode
92	005C	Basic input buffer pointer
94	005E	Current result type: 255=string, 0=numeric
95	005F	Current result type: 80=integer, 0=floating
96-121	0060-0079	OS and Basic working area
122-123	007A-007B	Pointer to start of Basic statements
124-125	007C-007D	Pointer to start of simple variable table
126-127	007E-007F	Pointer to start of array variable table
128-129	0080-0081	Pointer to start of available space
130-131	0082-0083	Pointer to bottom of strings (moving down)
132-133	0084-0085	Pointer to top of strings
134-135	0086-0087	Pointer to top of memory allocated for Basic
136-137	0088-0089	Current program line number
138-139	008A-008B	Current program line number saved by END
140-141	008C-008D	Current program pointer saved by END
142-143	008E-008F	Current line number of READ pointer
144-145	0090-0091	READ pointer
146-147	0092-0093	DATA statement pointer
148-149	0094-0095	Current variable symbols
150-151	0096-0097	Current variable starting pointer
152	0098	EOR operand for WAIT
153	0099	AND operand for WAIT
156	009C	Type of logical comparision
157-158	009D-009E	Pointer to user FN* function
159-161	009F-00A1	Work area used by SQR
163-165	00A3-00A5	Jump, User defined FN*
166-175	00A6-00AF	Work area for transcendental routines
174-175	00AE-00AF	Basic buffer transfer pointer
176	00B0	Exponent + $80 \
177	00B1	Mantissa MSB \
178	00B2	Mantissa \ Floating point
179	00B3	Mantissa / accumulatory (FACC)
180	00B4	Mantissa LSB /
181	00B5	Sign of Mantissa /
184-192	00B8-00C0	Dyadic holding area
184-189	00B8-00BD	Alternate floating point accumulator (AFAC)
194-217	00C2-00D9	Fetch next Basic character routine
201-202	00C9-00CA	Program pointer
218-222	00DA-00DE	Work area for RND
224-225	00E0-00E1	Address of cursor line position on screen
226	00E2	Offset of cursor from current line (E0-E1)
227-228	00E3-00E4	Current tape buffer pointer
229-230	00E5-00E6	End of current program
234	00EA	Quote mode 0=non-quote mode
235	00EB	0=no timer#1 interrupt
238	00EE	Number of characters in filename
239	00EF	GPIB file number
240	00F0	GPIB command
241	00F1	GPIB device number
242	00F2	Maximum number of chrs on current line (39,79)
243-244	00F3-00F4	Start of tape buffer
245	00F5	Current screen line number (0-24)
246	00F6	Running checksum of buffer
247-248	00F7-00F8	Pointer to program during VERIFY, LOAD, SAVE
249-250	00F9-00FA	Filename starting pointer
251	00FB	Remaining count for insert mode
252	00FC	Serial word
253	00FD	Number of blocks remaining to write
254	00FE	Serial word buffer
255	00FF	Basic
256-511	0100-01FF	Hardware stack and tape read work space
512-514	0200-0202	Clock JIFFIES, reset at 2400
515	0203	Matrix coordinate of last key down, 255=none
516	0204	Shift key status, 1=down
517-518	0205-0206	Clock, 1.5 JIFFIES
519	0207	Cassette 1 on switch
520	0208	Cassette 2 ON switch
521	0209	Keyswitch PIA
523	020B	LOAD 0, VERIFY 1
524	020C	Status
525	020D	Number of characters in keyboard buffer
526	020E	Reverse video
527-536	020F-0218	Keyboard input buffer
537-538	0219-021A	Hardware interrupt vector
539-540	021B-021C	Break interrupt vector
542	021E	Number of characters on screen line
544-545	0220-0221	Used in input routine
547	0223	Key image
548	0224	Cursor flash, 0=flash, 1=normal
549	0225	Cursor timing
550	0226	Copy of character at current position
551	0227	0=cursor moved, 1=blink started
552	0228	Tape write
553-577	0229-0241	State of each screen line format
578-587	0242-024B	Logical number of open files
588-597	024C-0255	Device number of open file
598-607	0256-025F	R/W modes of open files (command table)
608	0260	Input from: 0=keyboard, 1=screen
610	0262	GPIB table length
611	0263	Device number of input (0=keyboard)
612	0264	Device number of output (3=screen)
613	0265	Parity
616	0268	Pointer in filename transfer
620	026C	Serial bit count
623	026F	Flip for every bit coming off of tape
624	0270	Tape write countdown
627	0273	Leader counter
629	0275	0=first half byte marker not written
630	0276	0=second half byte marker not written
631	0277	Tape dropout count
632	0278	Tape correction count
633	0279	Checksum working word
634-825	027A-0339	Buffer for Cassette #1
826-1017	033A-03F9	Buffer for Cassette #2
1024-8191	0400-1FFF	Start of Basic statements
8192-32767	2000-7FFF	Expansion memory
32768-33767	8000-83E7	TV screen
36864-49151	9000-BFFF	Expansion memory
49152-57520	C000-E0B0	Microsoft Basic
50059-	C38B-	Ends running program and goes to READY mode
50317-	C48D-	Instruction recognizer
50930-	C6F2-	Get address for execution of Basic statement
51063-	C777-	RUN Basic program now in memory

P-E-T M-E-M-O-R-Y M-A-P

Decimal	Hex	Description
51666-	C9D2-	PRINT # CR LF
51751-51830	CA27-CA76	Output text beginning at (A,Y) & end with 0
51761-	CA31-	Output X characters starting at ($71,$72)
51783-	CA47-	PRINT
53207-	CFD7-	Find variable in table
53415-	D0A7-	Convert floating (FACC) to integer ($B3,$B4)
53880-	D278-	Convert integer (A,Y) to floating (FACC)
55080-	D728-	Dyadic subtraction (AFAC-FACC)
55103-	D73F-	Dyadic addition (AFAC+FACC)
55487-	D8BF-	LOG(FACC)
55552-	D900-	Dyadic multiplication (AFAC*FACC)
55646-	D95E-	Load the AFAC with number addressed by (A,Y)
55650-	D962-	Move variable at ($71,$72) to AFAC
55780-	D9E4-	Dyadic division (AFAC/FACC)
55924-	DA74-	Load the FACC (Floating Point Accumulator)
55928-	DA78-	Move variable at ($71,72) to FACC
55979-	DAAB-	Move variable at FACC to ($71,$72)
56106-	DB2A-	ABS(FACC)
56222-	DB9E-	INT(FACC)
56868-	DE24-	SQR(FACC)
56878-	DE2E-	Raise AFAC to power in FACC, result in FACC
56992-	DEA0-	EXP(FACC)
57246-	DF9E-	COS(FACC)
57253-	DFA5-	SIN(FACC)
57326-	DFEE-	TAN(FACC)
57416-	E048-	ATN(FACC)
57525-57981	E0B5-E27D	System Set-up
57981-	E27D-	Get character from buffer
58004-58986	E294-E66A	Video driver
58185-	E349-	Turns quote mode switch on and off ($EA)
58346-	E3EA-	Print character using $E0,$E1,$E2
58713-	E559-	Scroll screen up one line
58758-	E586-	IRQ Interrupt routine
58987-59012	E66B-E684	Interrupt handler
59013-59227	E685-E75B	Clock Update, Keyboard scan (60 HZ INT)
59199-59227	E73F-E75B	Keyboard scan
59228-59348	E75C-E7D4	Keyboard encoding table
59308-	E7AC-	Put character on screen during screen blank
59392-61439	E800-EFFF	I/O Ports & Expansion I/O (PIA's and VIA)
59408-59411	E810-E813	PIA, Keyboard
59408	E810	I/O Port A & Data direction register: C.R.A2
59409	E811	Control reg. A (C.R.A) 52=screen off, 60=on
59410	E812	I/O Port B & Data direction register: C.R.B2
59411	E813	Control reg. B (C.R.B)
59424-59427	E820-E823	PIA, IEEE Port
59424	E820	I/O Port A & Data direction reg: C.R.A2=1
59425	E821	Control reg. A (C.R.)
59426	E822	I/O Port B & Data direction reg: C.R.B2=1
59427	E823	Control reg. B (C.R.)
59456-59469	E840-E84D	VIA, *USER Port
59456	E840	I/O Port B
59457	E841	I/O Port A data with handshake *USER Port
59458	E842	Data direction reg. for B
59459	E843	Data direction reg. for A: 0=input, 0<output
59460	E844	Timer: 1-LO R/W
59461	E845	Timer: 1-HI R/W & initiate count
59462	E846	Access timer: 1-LO
59463	E847	Access timer: 1-HI
59464	E848	Timer: 2-LO RD & reset INT/WR
59465	E849	Access timer: 2-LO RD/WR & reset INT
59466	E84A	Serial I/O shift reg. CB2

P-E-T M-E-M-O-R-Y M-A-P

Decimal	Hex	Description
59467	E84B	Auxiliary control reg.
59468	E84C	Peripheral control reg: 0=neg, 1=pos
59469	E84D	Interrupt flag
59470	E84E	Interrupt enable
59471	E84F	I/O port A data without handshake *USER Port
60939-	EE0B-	SIGN(FACC)
61440-	F000-	I/O, Diagnostics, Monitor
61622-61990	F0B6-F226	GPIB handler
62026-	F24A-	Close all files
62250-	F32A-	Look for STOP
62278-	F346-	LOAD program from cassette
62481-	F411-	Write filename on screen
62515-	F433-	Get filename and number
62556-	F45C-	Get # in X
62651-	F4BB-	VERIFY
62724-	F504-	Get filename
62741-	F515-	Load current character from tape
62894-	F5AE-	Search for tape header
62947-	F5E3-	Clear 192 bytes addressed in $F3,$F4
62957-	F5ED-	Write end block on tape
63101-	F67D-	Set up tape end pointer
63134-	F69E-	SAVE
63153-	F6B1-	SAVE contiguous bytes of memory onto cassette
63286-	F736-	Update jiffy clock ($205-$206)
63533-64789	F82D-FD15	Tape control
63582-	F85E-	Check cassette on switch
63615-	F87F-	Read 192 bytes into tape buffer from cassette
63684-	F8C4-	Write a block
63765-	F915-	Loop until normal interrupt
64824-65458	FD38-FFB2	Power on diagnostics
64912-	FD90-	Compare tape buffer pointer with EOT buffer
65472-65516	FFC0-FFEC	Jump vectors
65487-65489	FFCF-FFD1	JSR RDT - Input a character (F1DF)
65490-65492	FFD2-FFD4	JSR WRT - Type a character (F230)
65508-65510	FFE4-FFE6	JSR GET - Get a character (F1CC)
65514-	FFEA-	Clock update
65530-65531	FFFA-FFFB	NMI (CA60)
65532-65533	FFFC-FFFD	RESET (FD38)
65534-65535	FFFE-FFFF	IRQ + BRK (E66B)

*** Dynamic Keyboard ***

Mike Louder

July.7,1978 (revised Aug.19,1978)

1148 Thompson Ave. 4, Glendale, Cal. 91201
213/246-0505

Would you like to add GOTO (expression) or GOSUB (expression) to your next BASIC program ? How about being able to add change or delete BASIC statement lines while a program is running ? Variations of a single fundamental procedure provide endless possibilities. The technique could be called programmable keyboard.

The decimal addresses for the keyboard buffer are 527 through 536.

The buffer counter address is 525.

If a BASIC program is interrupted with a STOP or END the keyboard moniter searches the keyboard buffer and executes any "ASCII" instruction that may have been typed in while the program was running. By using BASIC to "POKE" the buffer and appropriately interrupting the run mode, GOTO (expression) is easily implemented. Adding new "line's to your program is a more involved process with a few known disadvantages :

(1) Resident screen graphics will be interfered with.

(2) Changing BASIC "line's" will un-link and reset the variables list to zero.

(3) Subroutine pointers are lost.

If the routine is not used to change your BASIC code, only (1) above applies.

Try this GOTO (expression) demo.

```
10 X = 10
20 L = X*10 : GOTO 50000
100 X = X + 10
300 X = X + 10 : GOTO 20
400 PRINT " THE PET HAS A DYNAMIC KEYBOARD " : END

50000 PRINT "(CLR) (DOWN) (DOWN) (DOWN) GOTO" L "(HOME)" : POKE 525, 1 :
      POKE 527,13 : END
```

REM : Clear screen and position GOTO L so that cursor is located on the left after PRINT "(HOME)" and exiting BASIC with END. "Load" buffer with a count of 1 and a 13 code which represents the same action as pressing the RETURN key.

A simple example using "L = (expression) : GOSUB 50000 ":

```
10 INPUT AREA CODE
20 L = AREA CODE*10 : GOSUB 50000
30 GOTO 10
6060 PRINT " KENTUCKY :
6061 PRINT " ASHLAND, BUTLER, COVINGTON & LEXINGTON." : RETURN
etc.
```

The following utility routine is self explanatory. Call with GOTO or RUN 60000.

```
60000 PRINT "(CLR) DELETE LINE NUMBERS FROM J TO K : 'J,K'
60001 PRINT
60002 PRINT " REM J & K ARE INTEGERS FROM 0 TO 65535.
60003 PRINT
60004 INPUT J,K
60010 PRINT "(CLR) (DOWN) (DOWN)
60011 FOR I = J TO J + 8 : IF I > K THEN 60015
60012 PRINT I
60013 NEXT
60014 PRINT "J = " J + 9 ": K = " K ": GOTO 60010 "
60015 POKE 525, 10 : FOR N = 0 TO 9 : POKE 527 + N, 13 : NEXT
60016 PRINT "(HOME)" : END
```

60010 REM : Clear screen and position each group of 9 consecutive line numbers so that the cursor is in front of the top number after exiting BASIC at line 60016.

60011 REM : Generate nine consecutive line numbers until I > K.

60012 REM : Print line numbers on screen.

60014 REM : 10 th item on list. Since BASIC variables are set to zero, save K and update J on screen. Then return to BASIC line 60010.

60015 REM : "Load" keyboard counter with a maximum count of 10 and fill the buffer with the ASCII code which represents the same action as pressing the RETURN key (13).

60016 REM : Position cursor at HOME then exit BASIC and execute 10 consecutive RETURNs then return to BASIC via GOTO 60010.

Now change line 60012 to :

```
60012 PRINT I  "?"  CHR$ (34) CHR$ (34) CHR$ (20)  "THE PET HAS A DYNAMIC
      KEYBOARD.
```

And add :

```
510 END
```

Watch the action after : RUN 60000; K,J = 400, 500 : Then : RUN 400.

Mike Richter provided the CHR$ ( ) combo which supplies the otherwise missing quotation mark.

Changing 60012 back to "PRINT I " will allow you to delete lines 400 through 510.

As an example of generating new DATA lines as the result of a complex search-compare operation or involved math routine :

Change line 60012 to :

```
60012 PRINT I*10 "DATA" 2 ↑ I " , "  I ↑ 2
```

And add :

```
10 J = 1 : K = 30 : GOTO 60010
```

Using a technique similar to above, the PET "typed in" a large additional DATA table for the Joseph Roehrig 3 - D TIC - TAC - TOE game. (Kilobaud Apr 77). Now it takes only 24 seconds to make the first move.

The following suggests a way to effectively "load" 9 to 12 K programs into the PET's 7K RAM memory ! Many excellent programs lack "user instructions' because of large dimensioned arrays. The program listing may take only 3 to 4K of memory but as soon as the program pointer sees a DIM ( ) statement nearly 7K is needed. The QUBIC game is a good example.

Change lines 60011, 60012 and 60014 :

```
60011 FOR I = J TO J+8 : IF I > K THEN L = 20
60012 PRINT I
60014 PRINT  "J =" J + 9  " : K =" K  " : GOTO " L
```

Add :

```
60006 J = 59000 : K = 60015
60008 L = 60008
```

Program :

```
10 GOTO 59000
20 POKE 59409,60
30 DIM A (255,4)
40 PRINT "(CLR) THE PROGRAM NOW TAKES
50 PRINT 7167-FRE (0) "BYTES TO RUN.
60 PRINT : PRINT "PROMPT USER TO RELOAD PROGRAM TO SEE
70 PRINT "A REPEAT OF INSTRUCTIONS." : END
59000 PRINT "(CLR) THIS PROGRAM PRESENTLY REQUIRES
59010 PRINT 7167-FRE (0) "BYTES OF MEMORY.
59020 PRINT : PRINT "AFTER YOU READ THIS, THE SCREEN WILL
59030 PRINT "BLANK FOR 1 MINUTE 10 SECONDS WHILE
59040 PRINT "LINES 59000 THROUGH 60016 ARE DELETED.
59050 PRINT : PRINT "THAT INCLUDES THE LINE EDITING ROUTINE !
59060 PRINT : PRINT "    PRESS ANY KEY TO CONTINUE
59070 GET A$ : IF A$ = "" THEN 59070
59080 POKE 59409,52 : GOTO 60006
```

Note :

The deleted lines could have been portions of a BASIC program that produced a few intermediate variables which are temporarily stored like J and K or in the second cassette buffer memory location decimal 826 through 1023.

I would like to hear about other Dynamic Keyboard applications.

Useful programming aids :
(1) Pet User Notes : vol. 1 #3 (Mar - Apr 78) : page 9 : CHR$ character codes.
(2) Pet User's Group Newsletter (Sphinx) vol. 0 #1 page 8 : Pet memory map.

Notes : Fundamental idea came from previous SR-52 relocating code programs submitted to HP-65 User Club and the line adding routine illustrated in the Peninsula School "DRAW &K" (People's Computers Mar/Apr 78) program lines 7000 through 7040.
POKE address information was originally provided by Lennie Cooper and Leonard Tramiel of Commodore. Their fast response to my questions is greatly appreciated.
Finally, Mike Richter helped debug the procedure and pointed out how my first effort, which placed the GOTO L in the keyboard buffer, wasn't necessary. The method for handling variables J and K is his latest contribution. Mike has always provided programming tips when I needed it (which is very often).

PET BASIC Documentation

These snippets of the PET Basic interpreter have been annotated for a number of purposes:

1. I wanted to get some practice in 6502 machine coding, and this was the most convenient source of professionally done examples that I had. Deciphering the machine code and annotating it provides practice in abundance.

2. The transcendental functions appear to be not as accurate as they should be; I'm just curious why.

3. I'd like to know in some detail just how data is received and transmitted on the IEEE-488 bus and the user port.

4. I'd like to have something--maybe a software patch--to keep the machine from bombing just before I finish typing in a long program. The information so far--that there's a non-interruptable bit of code somewhere--doesn't help much, and the apparent incompatibility of the 019 ROM with at least some PETs doesn't make me feel very good about trying one in mine.

While the arithmetic routines don't seem to be all that difficult to fathom once they're found, the annotation does take time, and I'd like to encourage others to provide similar documentation for other parts of the system. I myself plan to continue decoding and annotating the arithmetic subroutines through the transcendental functions

I think my comments are accurate--I certainly hope so; but I would be more than pleased to be corrected on any point. Further, I'd appreciate knowing if greater detail would be desirable in future documentation. The experienced programmer will probably say that there's too much explanation, but I'm a novice and I can't afford to put in less detail because then I won't be able to understand it myself afterwards.

My thanks go to all those who have provided clues as to what's really going on inside PET, particularly to SPHINX, those who in the various users group notes have described various applications and actions of assorted SYS and POKE commands, and to those who privately have provided me with documentation that they but not I have received from Commodore.

Charles A. McCarthy
1359 W. Idaho Ave.
St. Paul MN 55108

Subroutine: Load a number from memory into the primary accumulator

DA74-DA98

Assumptions: The number consists of the bytes E, M_1, M_2, M_3, M_4 stored in memory locations

PPQQ through PPQQ+4. The Y register contains PP and the accumulator contains QQ

Upon return: QQ ε 71, PP ε 72, 00 ε BF
B0-B5 contain $E, 1.M_1, M_2, M_3, M_4, M_1$

The accumulator contains E
The X register has not been changed
The Y register contains 00

Code					
DA74	STA 71	STY 72	LDY#04	Sets up acquisition of number through Indirect-Y instructions	
DA7A	LDAIY 71	STA B4	DEY	M_4 ε B4	Y = 3
DA7F	LDAIY 71	STA B3	DEY	M_3 ε B3	Y = 2
DA84	LDAIY 71	STA B2	DEY	M_2 ε B2	Y = 1
DA89	LDAIY 71	STA B5		M_1 ε B5	
	ORA#80	STA B1	DEY	most sig. bit of mantissa inserted, $1.M_1$ ε B1. Y= 0	
DA92	LDAIY 71	STA B0	STY BF	E ε B0	00 ε BF
DA98	RTS				

Notes: $.1,M_1$ denotes the most significant stored bits of the mantissa with the implied 1 inserted. The sign of the mantissa remains available as the most significant bit of location B5

Subroutine: Load a number from memory into the secondary accumulator

D95E-D988

Assumptions: The number consists of the bytes E, M_1, M_2, M_3, M_4 stored in memory locations

PPQQ through PPQQ + 4. The Y register contains PP and the accumulator contains QQ

Upon return: QQ ε 71, PP ε 72
B8-BE contain $E, 1, M_1, M_2, M_3, M_4, M_1$, a sign flag.

The accumulator contains the exponent byte from the primary accumulator
The X register has not been changed
The Y register contains 00

Code

D95E	STA 71 STY 72 LDY#04	Sets up acquisition of number through indirect-Y instructions
D964	LDAIY 71 STA BC DEY	M_4 ε BC Y = 3
D969	LDAIY 71 STA BB DEY	M_3 ε BB Y = 2
D96E	LDAIY 71 STA BA DEY	M_2 ε BA Y = 1
D973	LDAIY 71 STA BD	M_1 ε BD
D977	EOR B5 STA BE	sign flag ε BE (see note)
D97B	LDA BD ORA#80 STA B9 DEY	M_1 recovered from BD. most sig. bit inserted, $1.M_1$ ε B9 Y = 0
D982	LDAIY 71 STA B8	E ε B8
D986	LDA B0	Exp from pri. acc. into acc.
D988	RTS	

Notes: $1.M_1$ denotes the most significant stored bits of the mantissa with the implied 1 inserted. The sign of the mantissa remains available as the most significant bit of location BD. Location BE is the exclusive or of the stored mantissae of the two numbers in the primary and secondary accumulators; thus the most significant bit of BE is 0 if the two mantissae have the same sign, and is 1 if the two mantissae have opposite signs.

Representation of floating point numbers in memory

A floating point number stored in memory consists of five consecutive bytes of data, denoted E, M_1, M_2, M_3, M_4.

E is the base 2 exponent centered at 128. $E = 128 + q$ means that the number lies between 2^q and 2^{q-1}.

E = 0 is taken as sufficent for the number to be zero

The most significant bit of M_1 is the sign of the mantissa: 0 if the mantissa is positive, 1 if the mantissa is negative. The remaining bits of M_1 through M_4 are the least significant 31 bits of the true mantissa. The most significant bit of the mantissa is always taken as 1, and is effected through ORA#80 with M_1; this destroys the sign information, however, which must be preserved elsewhere.

ROBERT PURSER'S REFERENCE LIST OF COMPUTER CASSETTES

POST BOX 466, EL DORADO, CA 95623

Hi!

Here are all the computer programs recorded on cassettes that I know of for the Radio Shack TRS-80, Commodore's PET, and Apple Computer Company's Apple II.

The Reference List of Computer Cassettes is published quarterly. The next edition will be in mid-November. It will cost $2.00. Send $2 by November 15th to guarantee receiving a copy of the November issue. (Robert Elliott Purser, Post Box 466, El Dorado, CA 95623) Later issues will cost $4.00. Until November 15, the subscription rate is $9.00 per year. After November 15, the rate will be $12.00.

If you see any programs you like, write the manufacturer asking them how to order their cassettes. Tell them you saw their name here in Robert Purser's reference list. Do not order directly from this list. This list is not a catalog. I have not seen any of the programs on this list. I do not review the programs listed here. I merely repeat any advertisement I see in various magazines.

REQUEST FOR SOFTWARE

Don't be shy! Someone out there is eagerly looking for your program. Help him!

Do you have any original programs? Please let me know. All listings are free.

I list anyone who has original programs for sale or trade. I will list your name, address, and phone. You can have one line to describe each cassette. At the end of the list, you can include any necessary comments about ordering.

REQUEST FOR ARTICLES

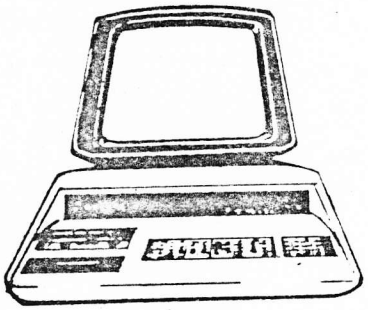
Starting with the February, 1979 edition of this reference list, I will begin printing articles about useful software. Basically, I am looking for an educational article about your program or your cassette of programs.

Preferably the article will be written by the author of the software. It must explain why and how the software is useful. It must deal directly with the software's application to some problem or need.

The article should take up no more than one typeset page. It may include line art. Sorry, no photos. The article may be submitted typed double space or may submitted as a camera-ready copy.

The articles will be printed in the future reference lists on a space available basis. Preference will be given to articles that are camera-ready.

ROBERT PURSER'S REFERENCE LIST OF COMPUTER CASSETTES
POST BOX 466, EL DORADO, CA 95623



12.

PET Cassettes

Edition 3, August 1978

For free information on the PET, send a large SASE with 41¢ postage to Microcomputer Resource Center, 1929 Northport Dr. Room 6, Madison, WI 53704. Ask for the current edition of the free PET Gazzette. An outstanding free public service.

A B Computers
P.O. Box 104
Perkasie, PA 18944

PET Biorythm, Mastermind II, Multi-primer, Othello, Klingon, Capture, Life	$ 8.00
PET Kites - - a unique two player game	$ 7.95
PET Assembler	$ 20.00
PET Link Editor	$ 20.00
PET Text Editor	$ 10.00

Acadiana Software Co.
Route 3, Box 778
St. Martinville, LA 70582
(318) 365-3602

PET Math Practice One	$ 10.00
PET Math 4.1 (Fourth grade lesson 1)	$ 10.00

Don Alan Enterprises
P.O. Box 401
Marlton, NJ 08053
(609) 234-5124

PET Housebreak Your Pet	$ 14.95

PET Star Trek	$ 19.95
PET Star Pilot	$ 14.95
PET Biorhythm	$ 9.95
PET The Dragon	$ 4.95
PET Black Jack	$ 9.95
PET Artillery	$ 9.95
PET Lem	$ 9.95
PET Hangman	$ 9.95
PET Five Card Draw	$ 9.95
PE[illegible]an Amortization	$ 14.95
P[illegible]use Break Your PET-2	$ 9.95
P[illegible]ketch Pad	$ 9.95
PET Osero, Keno, Checkers	$ 9.95
PET 3D Tic-Tac-Toe, Weather & Stock Mkt.	$ 14.95
PET Backgammon, Yahtze and Train Strain	$ 9.95
PET Grand Prix-Golf, Minefield, etc.	$ 9.95
PET Tank Shoot-Horserace, Snake Trap, etc.	$ 9.95
PET Checkbook	$ 9.95

Apparat, Inc.
Box 10324
Denver, CO 80210

Games

PET Horse Race	$ 3.00 + Tape
PET Acey - Ducey	$ 2.00 + Tape
PET Black Jack	$ 5.00 + Tape
PET Craps	$ 3.00 + Tape
PET Roulette	$ 3.00 + Tape
PET One Arm Bandit	$ 3.00 + Tape
PET Backgammon	$ 5.00 + Tape
PET Basketball	$ 3.00 + Tape
PET Football	$ 5.00 + Tape
PET Golf	$ 3.00 + Tape
PET Ski	$ 3.00 + Tape
PET Guess the number	$ 1.00 + Tape
PET Hurkle	$ 2.00 + Tape
PET Mastermind	$ 2.00 + Tape
PET Reverse	$ 1.00 + Tape
PET Lunar Lander	$ 3.00 + Tape
PET Star Trek	$ 5.00 + Tape
PET Wumpus	$ 3.00 + Tape
PET Hammurabi	$ 3.00 + Tape
PET King	$ 5.00 + Tape
PET Life Expectancy	$ 4.00 + Tape
PET Biorhythms	$ 3.00 + Tape
PET Hangman	$ 2.00 + Tape

Investment Programs

PET Apartment Analysis	$ 4.00 + Tape
PET Stock Option	$ 5.00 + Tape
PET Finance	$ 5.00 + Tape

Kitchen Programs

PET Indian Recipes	$ 4.00 + Tape
PET Recipe Cost	$ 2.00 + Tape
PET Diet Planning	$ 3.00 + Tape

Miscellaneous

PET Flight Simulator	$ 5.00 + Tape
PET Metric Conversions	$ 2.00 + Tape
PET Prime Numbers	$ 1.00 + Tape
PET Target Practice	$ 2.00 + Tape

CMS
5115 Menefee Drive
Dallas, TX 75227

Casino Series

PET Casino Blackjack	$ 9.95
PET Casino Roulette	$ 9.95
PET Casino Baccarat (Chemin de Fir)	$ 9.95
PET Casino Craps	$ 9.95

Strategic Boad Series

PET Qubic 4	$ 9.95
PET Go-Moku	$ 9.95

Commodore PET (tm)
(Pet is the trademark of
Commodore International, Ltd.)
Commodore Systems
901 California Avenue
Palo Alto, CA 94303

PET Osero and Reverse	$ 9.95
PET Diet Planner and Biorhythm	$ 14.95
PET Target Pong and Off the Wall	$ 9.95
PET A Treasure Trove of Games	$ 9.95
PET Basic Tutorial	$ 14.95
PET Galexy Games	$ 9.95
PET Mortgage	$ 14.95
PET Draw Poker	$ 9.95
PET Black Jack	$ 9.95
PET Space Flight	$ 9.95
PET Spacetrek	$ 9.95
PET Stock Portfolio Analysis Package	$ 24.95
PET Basic Finance/Investment Analysis	$ 24.95
PET Basic Math Package	$ 29.95
PET The PET Show	$ 9.95

Computer Complements Corp
147 Jericho Turnpike
Floral Park, NY 11001
(516) 775-8700

The Computer Factory
790 Madison Avenue
New York, NY 10021
(212) 249-1666

PET Mortgage	$ 15.95
PET Annual Report Analyzer (Manual)	$ 22.95
PET Stock Analyzer (Automatic Data Entry)	$ 34.95
PET Data Base (12 tapes/yr)	$175.00/yr
PET Stockscreen (requires second recorder)	$ 22.00
PET Options (Manual Data Entry)	$ 24.95
PET Options (Automatic Data Entry)	$ 24.95
PET Finance	$ 12.95
PET Bonds	$ 9.95
PET Checkbook	$ 15.95
PET Inventory Control (requires second recorder)	
PET Accounts Receivable (requires second recorder)	

Computer Project
Peninsula School
Peninsula Way
Menlo Park, CA 94025

PET Pilot Interpreter, Gold, Sky, Hammurabi, Names, Hands	$ 19.9
PET Draw, Quest	$ 9.9
PET Renumber, Lemon, Kaleidoscope, WSFN	$ 14.9

Computers One
#306 Kahala Office Tower
4211 Waialee Avenue
Honolulu, Hawaii 96816
(808) 737-2933

PET Hustler series 1 - General Ledger	$ 21.4
PET Hustler series 1 - Checking Account	$ 21.4
PET Hustler series 1 - Rent Accounts	$ 18.4
PET Hustler series 1 - Legal Diary	$ 18.4
PET Hustler series 1 - Trust Accounts	$ 18.4

Contemporary Marketing Inc.
790 Maple Lane
Bensenville, IL 60106
(800) 323-2273

PET Game programs	$ 9.95 each
PET Business programs	$ 24.95 each
PET Math packages	$ 29.95 each

Creative Software
P.O. Box 4030
Mountain View, CA 94040

PET Household Finance I and II	$ 15.00
PET Math Practice	$ 11.00
PET Space War	$ 11.00
PET Gamepac #1 (5 games)	$ 12.00
PET Gamepac #2 (5 games)	$ 12.00
PET Household Utility #1 (3 programs)	$ 10.00
PET Household Utility #2 (3 programs)	$ 10.00
PET Trap & Craps	$ 10.00
PET Wumpus & Capitals	$ 10.00
PET Osero (Othello) & Tic-Tac-Toe	$ 10.00
PET Hurkle & One-Queen	$ 10.00
PET Breakout	$ 10.00
PET Joystick Sketchpad & Maze (with joystick)	$ 37.00
PET Star Wars Shootout (for joysticks)	$ 10.00
(Shipping Charges extra)	

Dr. Daley
425 Grove Ave.
Berrien Springs, MI 49103

Social Science Simulations

PET Hamurabi	$ 4.00
PET King	$ 7.00
PET Dictator	$ 9.00

Science Fiction Games

PET Capture	$ 4.00
PET Lunar 1	$ 5.00
PET Deepspace	$ 5.00
PET UFO	$ 5.00
PET Enterprise	$ 6.00
PET Spacebattle	$ 9.00
PET Startrek	$ 6.50
PET Trek 3	$ 10.00
PET Swords & Sorcery	$ 5.00
PET Swarms	$ 9.00

Sports Simulations

PET Road Ralley	$ 5.00
PET Football	$ 7.50

Search Games

PET Hurkel	$ 4.00
PET Wumpas	$ 6.00

Arcade-type Games

PET Deflection	$ 5.00
PET Racetrack	$ 3.50
PET Racetrack 2	$ 6.00
PET Swat	$ 3.50
PET Chase	$ 7.50
PET Pong	$ 5.00
PET Breakout	$ 9.00
PET Snake	$ 7.50
PET ZZZAPP!	$ 7.50

Board Games

PET Qubic	$ 5.00
PET Checkers	$ 7.50
PET Othello	$ 7.50
PET Chess	$ 10.00

FUTURE EDITIONS: NOVEMBER 1978 WILL BE $ 2.00 PPD., FEBRUARY 1979 WILL BE $ 4.00

Robert Purser's Reference List Of Computer-Cassettes

Post Box 466, El Dorado, CA 95623

...llaneous

PET Match 23 $ 3.00
PET Dogfight $ 3.00
PET Number $ 3.00
PET Stars $ 3.00
PET Letter $ 3.00
PET Bible Book $ 4.50
PET Bible Quiz $ 6.00
PET Biorhythm $ 7.50
PET Eliza $ 7.50
PET Kaledioscope $ 4.00
PET Life $ 5.00

Games for Two

PET Football 2 $ 10.00
PET Market $ 6.00

Games with Sound Effects

PET Deflection 2 $ 7.50
PET Startrek 2 $ 9.00
PET Amazing Grace $ 4.00

More Serious Stuff

PET File $ 7.50
PET Dissembler $ 10.00
PET Renumber $ 12.50
PET Typewriter $ 5.00
PET Sort $ 5.00

Chemistry

PET Naming Aliphatic Compounds $ 15.00
PET Naming Alkanes (up to 10 carbons) $ 15.00
PET Naming Aromatic Compounds $ 15.00

Data Systems
Box 1873
Rocky Mount, NC 27801

PET Software exchange

Eclectic Corp
...30 Walnut Hill Lane
...llas, TX 75229
...(41) 358-1307

PET 2 Slots, Balckjack, Poke, Craps $ 15.70
PET 3 Golf, Bowling, Hamurabi, etc $ 15.70
PET 7 Finantial Analysis & Amortization $ 15.70

Evan H. Foreman
P.O. Drawer F
Mobile, AL 36601

PET machine Language Monitor with instructions $ 20.00

Andy Fraley
...753 York Road
...eading, PA 19610

PET Seawolf $ 6.00
PET Dogfight $ 6.00
PET Bomber $ 6.00
PET Indy 500 $ 6.00

G2 Program Library
GRT Corp
1286 North Lawrence Station Road
Sunnyvale, CA 94086

PET Beat the House $ 14.95
PET Dailors and Sense $ 14.95
PET Clinic $ 14.95

Home Computer Centre
6101 Yonge Street
Willowdale, Ontario
Canada, M2M 3W2
(416) 222-1166

PET Music $ 12.00
PET Entry $ 12.00
PET Day/Timer $ 12.00
PET Monitor $ 20.00
PET Graph $ 12.00

Kurt Huebner
...266 Valle Vista
...lerton, CA 92631

PET 31 Games for Trade

J K Johnson
9304 Emory Grv Rd
Gaithersburg, MD 20760

PET Black Jack $ 7.00

Randy Julin
15 Poncetta Drive, #322
Daly City, CA 94015

PET Software Exchange

Don Ketchum
313 Van Ness Ave.
Upland, CA 91786

PET Read, Write PET Memory $ 7.95

Kilobaud Book Dept.
Peterborough, NH 03458

PET Weight Control / Biorhythm $ 7.95
PET Mortgage / Finances $ 7.95

A.L.
P.O. Box 7211
Grand Rapids, MI 49510

PET Blackjack $ 6.00
PET Biorhythm $ 6.00
PET Amortization $ 6.00
PET Depthcharge $ 6.00
PET Tic-Tac-Toe 3D $ 6.00
PET Multiplication $ 6.00
PET Hangman $ 6.00

M. H. Major
A to Z Inc.
300 Guaranty Bank Plaza
Corpus Christi, TX 78475

PET Payroll $ 15.00

Magnemedia
17845 Sky Park Circle, Suite H
Irvine, CA 92714
(714) 549-9122

PET True/False Quiz $ 7.50
PET Variable Message $ 7.50
PET Don't Fall $ 7.50
PET Memory Aide $ 7.50
PET Drawing $ 7.50

Mathematical Application Service
Howard Y. Grosman
Box 149
New City, NY 10956
(914) 425-1535

Matrix Magazine
Michael Erlewine
1041 North Main Street
Ann Arbor, Michigan 48104

PET Basic Natel Charts & Houses $ 10.00
PET Aspects, Midpoints, Harmmics $ 10.00
PET Complete Natal Package $ 10.00
PET Games
PET I - CHING

Microcomputer Resource Center
1929 Northport Dr., Room 6
Madison, WI 53704

PET Cassette Exchange
All programs are exchanged on a 1 to 1 basis plus $1.00 handling per program.

PET Acey Deucy (in between)
PET Adding with Blocks
PET Addition Game
PET Air War
PET Air Ace
PET Alphabetizing
PET Animal (with data tape provision)
PET Apartments
PET Art Auction
PET Backgammon Board
PET Battleship
PET Benchmark Series
PET Bid
PET Big Letters
PET Biorhythm for printer
PET Biorythm for TV
PET Blackbox Sound
PET Black Jack
PET Boggle
PET Breakout
PET Brownian Ball
PET Chase
PET Chase with Sound
PET Checkers (3 versions)
PET Chessboard
PET Clock
PET Computer Art
PET Concentration
PET Convert (cartesian/polar)
PET Core Dump
PET Crypto-One
PET Dazzler
PET Deflection with Mines
PET Deflection
PET Demo
PET Dogfight
PET Dot Race
PET Disinterpreter
PET Draw
PET Eliza
PET Employee Tax
PET Expand
PET Football
PET Forest Fire
PET Forms Demo
PET Game Box
PET Global
PET Golf
PET Grades
PET Gomoku
PET Guess (hi or low)
PET Haiku
PET Hamurabi
PET Hangman
PET Hex/Dec
PET Hexpawn
PET Himondiss
PET Household Finance
PET Hunter
PET Hurkel
PET Inventory Control
PET Kalidescope
PET Keyboard display
PET Klingon Capture
PET Letter Matching
PET Letter Sequence
PET Life
PET Loan
PET Lomondis
PET Magic Square
PET Mandala
PET Market
PET Mastermind
PET Math Drill
PET Math Games
PET Math Quiz
PET Maxit
PET Morse Code Tutor
PET Multiple Precision Hex Arithmetic
PET Multiple Regression
PET Music
PET Music Memory
PET Music at Random
PET Osero
PET Othello
PET Patterns
PET Poker
PET Pong
PET Pop Shot with Sound
PET Popcorn, etc.
PET Prime Number Generator
PET Profit Volume Analysis
PET Racetrack
PET Random Character Generator Test (bar graph + %)
PET Readability
PET Renumber & Unlist
PET Reverse
PET Rhymes
PET Road Ralley
PET Run!
PET Shooting Gallery
PET Slot Manchine
PET Snake
PET Snoopy
PET Sort
PET Space Convention (display only)
PET Space Shooter
PET Spades
PET Spell
PET Speed Reading
PET Spelling Test
PET Spiral
PET Stars
PET Startrek
PET State Capitols
PET Submarine Search
PET Swat & Doodle
PET Tac-Tic
PET Tape Copier
PET Tape Library
PET Tic-Tac Toe
PET Tommy Termite
PET Triangles
PET 23 Matches
PET Typewriter (Word Processor)
PET War
PET Widget
PET Wibur
PET Word Backwards?
PET Wrap Trap
PET Wumpus
PET Yahtzee

Future Editions: November 1978 Will Be $ 2.00 ppd., February 1979 Will Be $ 4.00

Robert Purser's Reference List Of Computer Cassettes

Post Box 466, El Dorado, CA 95623

Micr[illegible]al
P.O. Box 161988
Sacramento, CA 95816

PET Voicetrap (requires their Compuvox)	$ 3.95
PET Voicemaze (requires their Compuvox)	$ 3.95
PET Hexmon hexadecimal monitor	$ 3.95
PET Soundtrap (for their Computone)	$ 3.95
PET Soundbreak (for their Computone)	$ 3.95
PET PETypePROG (for their PEType)	$ 3.95

Microtronics
5943 Pioneer Road
Hughson, CA 95326
(209) 634-8888

PET Morse and RTTY with Ham Interface kit	$ 69.95

Mind's Eye Personal Software
P.O. Box 354
Palo Alto, CA 94301
(415) 326-4039

14.

Minikin Knits
Education Department
Robert Elliott Purser
P.O. Box 466
El Dorado, CA 95623

PET CAI Study Drills (Available 9/78)	$ 7.95
PET Custom-made CAI Study Drills	$ 9.95
PET Special Custom-made Study Drills for Schools	Free

NEECO
New England Electronics Co.
248 Bridge Street
Springfield, MA 01103
(413) 739-9626

PET Music Box (with hardware)	$ 49.95
PET Depth Charge	$ 9.95
PET Bombadier	$ 9.95
PET Anti-Aircraft	$ 9.95
PET Dogfight	$ 9.95
PET LEM Lunar Lander	$ 7.95
PET Blackjack	$ 7.95
PET Deflection	$ 9.95
PET Hunt the Wumpus	$ 5.95
PET Masterbrain	$ 9.95
PET NEECO Game Package	$ 24.95
PET Grades (for teachers)	$ 7.95
PET Slot Machine	$ 5.95
PET Statistics	$ 7.95
PET 1 Queen	$ 6.95
PET Depreciation	$ 8.95
PET Biorhythm	$ 7.95
PET Bullfight	$ 5.95
PET Scramble	$ 7.95
PET Space War	$ 9.95
PET Poker	$ 7.95
PET Evasion	$ 7.95
PET Emperor	$ 9.95
PET Addition Game	$ 4.95
PET Entrapment	$ 6.95
PET States and Capitols	$ 4.95
PET Hex/Dec	$ 5.95

(add $2.00 per order)

Newman Computer Exchange
1250 N. Main Street
P.O. Box 8610
Ann Arbor, MI 48107
(313) 994-3200
TWX 810-223-6023

Personal Software
Box 136 - DJ
Cambridge, MA 02138
(617) 783-0694

P[illegible]troductory Special	$ 12.95
PET Contract Bridge and 3D Tic Tac Toe	$ 14.95
PET Poker, Kingdom, Queen, Matadore	$ 9.95
PET Stimulating Simulations	$ 14.95
PET 6502 Assembler in BASIC	$ 24.95
PET Graphics Package	$ 14.95
PET Astrology	$ 14.95
PET Word Processing Package	$ 29.95

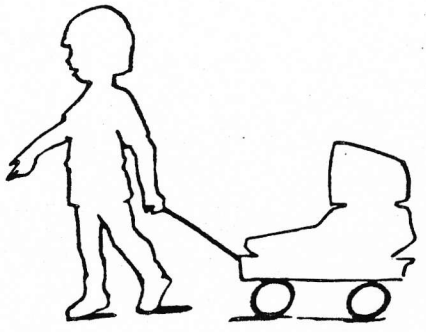
The Pet Paper
P.O. Box 43
Audubon, PA 19407
(215) 631-9052

Flea Market Software Exchange

PET Addition Game	
PET Othello for One	$ 5.00
PET Othello for Two	$ 5.00
PET Codes	$ 5.00
PET Road Rally	$ 5.00
PET Slot Machine	$ 5.00
PET Useful Routines	$ 5.00
PET States and Cities	$ 5.00
PET Emperor	$ 5.00
PET Grades	$ 5.00
PET Hex-Dec	$ 5.00
PET Math Quiz	$ 5.00
PET Flea-Pak I	$ 5.00
PET Pubdomdis	$ 5.00
PET Note	$ 5.00
PET Kalidescope	$ 5.00
PET Keyboard Utility	$ 5.00
PET Renumber & List	$ 5.00
PET Himondis & Himon Intro	$ 5.00
PET Lomondis	$ 5.00
PET Air War	$ 5.00
PET Dog Fight	$ 5.00
PET Prime Numbers	$ 5.00
PET Stars	$ 5.00
PET Morse Code Tutor	$ 5.00
PET Digital Clock	$ 5.00
PET Space Convention	$ 5.00
PET Tic-Tac-Toe	$ 5.00
PET Matches	$ 5.00
PET Number Trap	$ 5.00
PET Biocycle	$ 5.00
PET Amortization	$ 5.00
PET Yahtzee	$ 5.00
PET Vacation Budget	$ 5.00
PET Typewriter-5	$ 5.00
PET Spades	$ 5.00
PET Maxit	$ 5.00
PET Hangman (with super graphics)	$ 5.00
PET Magic Square	$ 5.00
PET Darts	$ 5.00
PET Core Dump	$ 5.00
PET Tommy Termite	$ 5.00

Software Shelf

PET Deflection	$ 10.00
PET Data Retrieval	$ 10.00
PET Data Edit (Companion to Data Retrieval)	$ 10.00
PET Stat	$ 20.00
PET Stat II	$ 10.00
PET Chase	$ 10.00
PET Blockade	$ 10.00
PET Space talk / Space flight	$ 10.00
PET Blackjack	$ 10.00
PET Hostage	$ 7.95



PET-SHACK Software House
Marketing and Research Co.
P.O. Box 960
Mishawaka, IN 46544

PET Machine Language Monitor	$ 9.95
PET Peek-A-Boo & 6502 Disassembler	$ 12.95
PET Mad Libs	$ 5.95
PET World Conquest	$ 5.95
PET Startrek	$ 5.95
PET Mortar	$ 5.95
PET Psyco Annie	$ 5.95
PET Computer Derby	$ 5.95
PET U-Guess-It & Swat	$ 5.95
PET Bounce	$ 5.95
PET Race Car	$ 5.95
PET Numberama	$ 5.95
PET States	$ 5.95
PET Math Tutor	$ 5.95

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

PET Sequence	
PET Statecaps	
PET Racetrack	
PET Deflection	
PET Addition	
PET Othello	
PET Othello for Two	
PET Bagels	
PET Cash Flow/ Return on Investment	
PET Star Trek	
PET Blackjack	
PET Life 40 by 25	
PET Life 64 by 64	
PET File List & Sort	
PET Trap	
PET Super Mastermind	
PET Lunar Lander	
PET Nim	
PET Qubic	
PET List Memory	
PET Machine Language Monitor - like Commodores	
PET Time	
PET Biorhythm	
PET King	
PET Breakout	
PET Swatplot	
PET Curfit - fits data to 6 curves	
PET Triangle - solves any triangle	
PET Metric - does metric conversion	
PET Dates	
PET Trendline - fits forecasts, graphs	
PET Mileage - distance from lat/long	
PET Prime Factors	
PET Mortgage	
PET Finance	
PET Battleship	
PET Moonlander	
PET Cryptogram Solving Aids	
PET Jotto - guess a word	
PET Chase	
PET Poems	
PET Mystery - you can't list it	
PET Concentration	
PET ESP Test	
PET Word Processor	
PET Maxit	
PET Market	
PET Wumpus	

(write for list; 4 programs $5.00)

PROGRAM
Richard Mansfield
P.O. Box 461
Philipsburg, PA 16866

PET PROGRAM Monthly Cassette Magazine	$ 27.00/year

Program Design Inc.
11 Idar Court
Greenwich, CT 06830

PET Preschool IQ Builder	$ 9.50
PET Step-by-Step Basic	$ 29.95
PET IQ Builder: Vocabulary	$ 12.50
PET IQ BUILDER: Number Series	$ 9.50
PET IQ BUILDER: Analogies	$ 9.50

Reich Engineering
635 Grannini Drive
Santa Clara, CA 95051

PET Play-grams	$ 9.95

Michael Richter
2600 Colby Avenue
Los Angeles, CA 90064

PET Software Exchange

PET Boggle	
PET Crypto-sub	
PET Tic-Tac-Toe	
PET Craps	
PET Chess 1	
PET Bridge	
PET Slot Machine	
PET RPN Calculator	
PET 4K RPN Calculator	
PET Fourier Series	
PET Hex Arithmetic	
PET Multi-precision Math	
PET Powers of 2	
PET Editor	
PET Typewriter	
PET Tachistoscope	
PET Stereo System Selection (incomplete)	
PET Audio Speaker Placement (incomplete)	
PET Encryption (not yet available)	
PET Computer Load Model (availibility pending approval)	
PET Big Primes (availibility pending approval)	

Future Editions: November 1978 Will Be $ 2.00 ppd., February 1979 Will Be $ 4.00

Robert Purser's Reference List Of Computer Cassettes

Post Box 466, El Dorado, CA 95623

Walter Rychlewski
603 Spruce
Libery, MO 64068

PET Demo Tape

Sawyer Software
628 Lewis - Rt. 3
Dexter, MO 63841

PET Accounting Pack 1	$ 25.00
PET Financial Pack 1	$ 15.00
PET Chessboard	$ 15.00
PET Payroll	$ 30.00
PET Checkbook Reconciliation	$ 25.00
PET Business Graphic Pack 1	$ 25.00
PET Business Analysis	$ 30.00
PET Schedual Planner	$ 15.00
PET Schedual Planner #2	$ 20.00
PET Calculator	$ 10.00

R. Schmidt
14 Tinker Rd.
Nashua, NH 03060

PET 2K Demo and Games	$ 15.95

Smith Business Services
P.O. Box 1125
Reseda, CA 91335

PET Starterk 8K	$ 7.95
PET Educational Games (Story, Guess, Hammurabi)	$ 7.95
PET Demos & Utilities (Dumps, Mem test, screen)	$ 7.95
PET Blockade	$ 7.95
PET Machine Language Monitor	$ 11.95

Softbyte
315 Dominion Drive
Newport News, VA 23602

PET Federal Income Tax Program	$ 19.50

Softape
10756 Vanowen
North Hollywood, CA 91605
(213) 985-5763

PET Software Exchange

Sof-Touch
Edward A. Crossman
Box 422
Logan, Utah 84321

PET Cannon-Ball / 3D Tic-Tac-Toe	$ 8.95
PET Spin-to-Win / Battleships	$ 8.95
PET Dog Race	$ 4.95

Speakeasy Software Ltd.
Box 1220
Kemptville, Ontario
Canada K0G 1J0
(613) 258-2451

PET Bulls and Bears	$ 12.00
PET Warlords	$ 12.00
PET Microtrivia	$ 12.00

(available only through computer stores)

SPHINX (a PET users group)
C/O Milton Lee
1348 Rudgear Road
Walnut Creek, CA 94598

All tapes are available free for reproduction at meetings which are every second and fourth Thursday at Trinity Lutheran Church, 1323 Central, Alameda, CA. If sufficient interest exists, a collection of programs (20 on each side) will be available for approx $5.00 or trade.

PET Ruler & Emperor
PET Poker & Blackjack
PET Star Trek & Deep Space
PET Shoot & Pong
PET Snake & Osero
PET Yahtzee & Wumpus
PET Brain Strain & Breakout
PET Lunar Lander & Chase
PET Hangman (Adv) and Hangman I
PET Dog Fight & Space Shooter
PET Trap & Stars
PET Road Rallev & Orbit
PET Maxit & Hunter
PET Reverse & Mastermind
PET Hurkle I & Hurkle
PET Planet Lander & Dragon
PET Kaleidoscope & Expanding Window
PET Bagels
PET Spades 1 & 2
PET Loans & Global
PET Grammar 1 & Readability Test
PET Apartments & Draw 2
PET Home Finance & Music
PET Fractions & Organizer
PET Cryptogram & Biorhythm
PET General Utilities 1.3
PET Himon Diss 1.6 & Himon Intro.
PET Himon Diss 1.3 & New Himon Diss 1.3 (0400-09FF)

Warren D. Swan
15933 S. Grove Avenue
Oak Forest, IL 60452

Games

PET Real Time Star Trek	$ 20.00
PET IQ Test	$ 6.00
PET Craps	$ 6.00
PET Baseball	$ 4.00
PET Eliza Doctor	$ 6.00
PET Road Race	$ 2.00
PET Lifetime	$ 2.00
PET Zoop	$ 1.00

Graphics

PET Garge Letter Printer	$ 5.00
PET Life	$ 5.00
PET Calendar	$ 3.00
PEt Pet-Art	$ 2.00
PEt 3D-Plot	$ 2.00
PET Dazzler	$ 2.00
PET Etch-a-Sketch	$ 1.50
PET Bouncing Ball	$ 1.50
PET Screen Blink	$ 1.00
PET Kaleidoscope	$ 1.00

Math

PET Base Converter	$ 3.00
PET Regression	$ 5.00
PET Calculator	$ 5.00

(add $1.50 per order for tape)

15.

Raynor Taylor
Qtrs. 718 Navel Base
Charleston, SC 29408

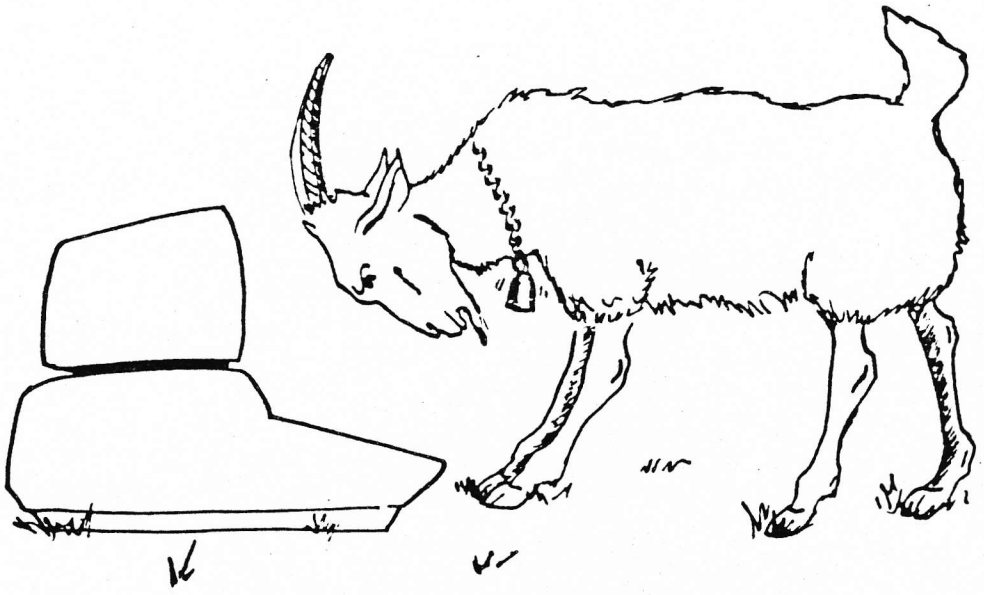
PET Monitor, disassembler,demo,biorhythm,target	$ 7.95
PET Disassembly of Basic,Operating System,Editor	$ 9.95

Total Information Services
P.O. Box 921
Los Alamos, NM 87544

PET Micro Text Editor	$ 9.90

ZZYP Data Processing
2313 Morningside
Bryan, TX 77801
(713) 693-3462

PET ZZYP-PAX #1 (Iron Planet & Hangman)	$ 9.95
PET ZZYP-PAX #2 (Black Bart & Black Bret)	$ 9.95
PET ZZYP-PAX #3 (Block & Football)	$ 9.95



Future Editions: November 1978 Will Be $ 2.00 ppd., February 1979 Will Be $ 4.00

PET SOFTWARE

AMORTIZATION -compute your finances.....$6
BIORHYTHM-plotted before your eyes.....$7
BLACKJACK-full graphics version.....$7
DEPTHCHARGE-seek out enemy subs.....$7
HANGMAN -spelling game for children.....$8
MULTIPLICATION-learn without tables.....$7
TIC-TAC-TOE-in three dimensions.....$6

All programs are on ready-to-load digital cassette for 8K PETS. They are documented and easily modified. 20% discount on order of 3 or more. VISA/BA or MASTERCHARGE cards are accepted(include card# and expir.). Postage paid in U.S.

ABACUS SOFTWARE
P.O. BOX 7211
GRAND RAPIDS, MI. 49510

C-U-R-S-O-R M-A-G-A-Z-I-N-E

CURSOR is a monthly cassette magazine written for 8K PETs. The first issue of CURSOR we received contained two games, a coordination game of throwing a brick thru a disappearing window and a game similar to Chase where you are a shark trying to eat swimmers and evade skin divers and traps. They also included their own version of Squiggle called Wander. The last two programs are a simple graphic utility program and a typewriter simulator.

Each issue of CURSOR is advertised to contain:

* Five or more programs
* A featured game
* Educational and practical programs, programming tricks and pictures
* Recorded on a C-30 cassette
* Mailed First Class
* Use the PET graphics to the fullest

A subscription to CURSOR cost $24.00, covers one year (12 issues), and began July 1978. For more information, contact:

CURSOR Magazine
Box 550
Goleta, CA 93017

P-U-G M-E-E-T-I-N-G (South Bay)

Time: Wednesday, September 6, 7:00 to 10:00 pm
Location: Ford Aero-Space Cafeteria, 3825 Fabian Way
Directions: Take San Antonio Road turnoff from Highway 101, turn right on to Charleston and right on to Fabian. The cafeteria is located BEHIND BUILDING #3.

S-P-H-I-N-X M-E-E-T-I-N-G (East Bay)

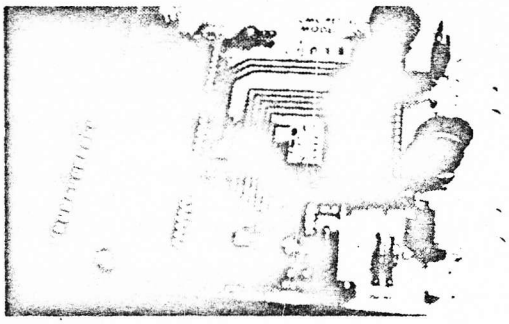
Time: Second Thursday, September 14, 7:00 pm
Location: Trinity Lutheran Church, 1323 Central, Alameda

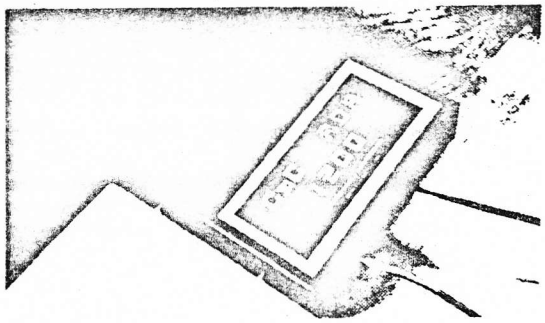
Time: Fourth Thursday, September 28, 7:00 pm
Location: Saint Agnus School, 755 Ashbury St., S.F.
Two blocks above Haight St.

CONNECTICUT microCOMPUTER

150 POCONO ROAD - BROOKFIELD, CONNECTICUT 06804

(203) 775-9659





RS-232 PRINTER ADAPTER FOR THE COMMODORE PET

The CONNECTICUT microCOMPUTER ADApter model 1200 is the first in a line of peripheral adapters for the COMMODORE PET. The CmC ADA 1200 drives an RS-232 printer from the PET IEEE-488 bus. The CmC ADA 1200 allows the PET owner to obtain hard copy program listings, and to type letters, manuscripts, mailing labels, tables of data, pictures, invoices, graphs, checks, needlepoint patterns, etc., using a standard RS-232 printer.

The CmC ADA model 1200B comes assembled and tested, without power supplies, case, or RS-232 connector for $98.50. The CmC ADA 1200C comes complete for $169.00. Specify baud rate when ordering. (300 baud is supplied unless otherwise requested. Instructions for changing the baud rate are included.)

WORD PROCESSOR FOR THE COMMODORE PET

17.

CONNECTICUT microCOMPUTER now has a word processor program for the COMMODORE PET. This program permits composing and printing letters, flyers, advertisements, manuscripts, articles, etc., using the COMMODORE PET and an RS-232 printer.

Script directives include line length, left margin, centering, and skip. Edit commands allow the user to insert lines, delete lines, move lines, change strings, save onto cassette, load from cassette, move up, move down, print and type.

The CmC Word Processor Program addresses an RS-232 printer through a CmC printer adapter.

The CmC Word Processor Program is available for $29.50.

RS-232 TO CURRENT LOOP/TTL ADAPTER

The CmC ADApter model 400 has two circuits. The first converts an RS-232 signal to a 20 ma current loop signal, and the second converts a 20 ma current loop signal to an RS-232 signal. With this device a computer's teletype port can be used to drive an RS-232 terminal, or vice versa, without modification of the port. The CmC ADA 400 can also be paralelled to drive a teletype or RS-232 printer while still using the computer's regular terminal. The CmC ADA 400 can easily be modified to become an RS-232 to TTL and TTL to RS-232 ADApter. The CmC ADA 400 does not alter the baud rate and uses standard power supplies. The current loop is isolated from the RS-232 signal by optoisolators.

The CmC ADA 400 is the perfect partner for KIM if you want to use an RS-232 terminal instead of a current loop teletype.

The CmC ADA 400S comes with drilled, plated through solder pads and sells for $24.50. The CmC ADA 400B comes with barrier strips and screw terminals and sells for $29.50.

This announcement was composed on a COMMODORE PET and printed on a GE TermiNet using a CmC ADA 1200C printer adapter and the CmC Word Processor Program.

Qnty	Description	baud rate	price	total
	CmC ADA 1200B (basic)		$98.50	
	CmC ADA 1200C (complete)		$169.00	
	CmC Word Processor Program (cassette)		$29.50	
	CmC ADA 400S (solder pads)		$24.50	
	CmC ADA 400B (barrier strips)		$29.50	
		Subtotal		
	Connecticut residents add 7% sales tax			
	Handling and shipping - add per order		$3.00	
	Foreign air mail - add $5.00 per order			
		Total included with order		

Mail with remittance or charge information to:

CONNECTICUT microCOMPUTER
150 Pocono Road, Room 12
Brookfield, Conn. 06804

NAME

COMPANY

ADDRESS

CITY

STATE ZIP

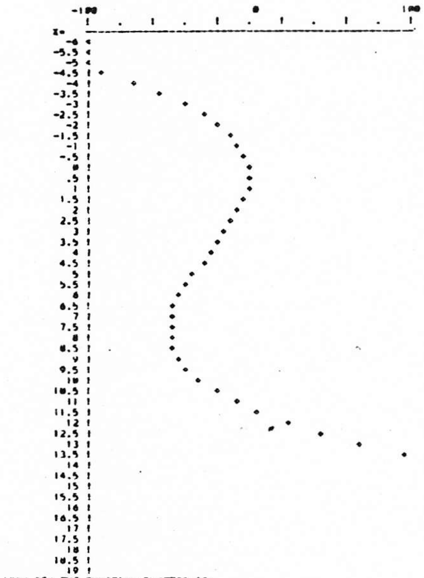
CHARGE TO: VISA MASTER CHARGE M/C INTERBANK NUMBER Expiration date

Credit card number

SIGNATURE

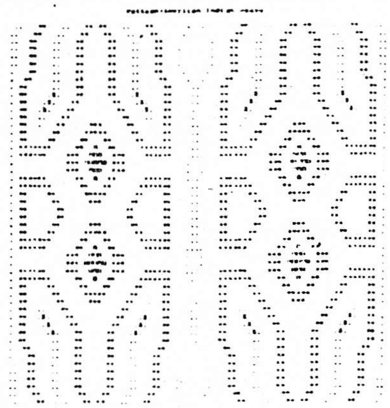
PET prints

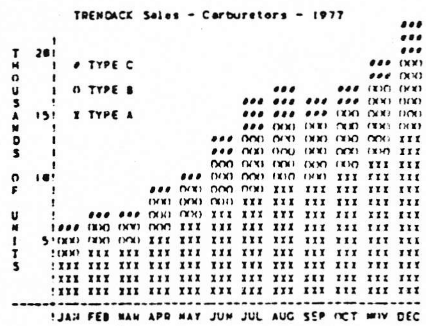
COMMODORE PET HARD COPY OUTPUT USING PET ADA 1200




```
1000 REM THE FUNCTION PLOTTED IS:
1010 Y=X*X*X/4-3*X*X+2*X-1
READY.
```

```
1 OPEN 6,8:CMD 6:LIST
10 REM ARCSIN AND ARCCOS FUNCTIONS FOR THE COMMODORE PET
20 REM WRITTEN BY RICHARD WUSNER
30 REM              CONNECTICUT MICROCOMPUTER
40 REM              150 POCONO ROAD
50 REM              BROOKFIELD, CT 06804
60 REM
70 REM "Listed on a GE TermiNet 300
80 REM "using a CmC ADA 1200.
90 REM
100 REM OPEN OUTPUT FILE ON DEVICE #8.
110 OPEN 5,8
120 REM
500 REM GET A SINE VALUE
510 INPUT S
520 C=S
530 REM
1000 REM THE SINE OF THE ANGLE IS S
1010 REM IF THE SIN IS IN THE RANGE OF -1 TO 1, THEN COMPUTE.
1020 IF S<1 AND S>-1 GOTO 1050
1030 AS=90:GOTO2000
1040 REM THE ARCSINE IS AS
1050 AS=ATN(S/((1-S*S)^.5))
1060 REM THE RESULT IS IN RADIANS. CONVERT TO DEGREES.
1070 AS=AS*180/π
2000 REM THE COSINE OF THE ANGLE IS C
2010 REM IF THE COSINE IS IN THE RANGE OF -1 TO 1,
2011 REM       AND NOT = TO 0, THEN COMPUTE
2020 IF C<>0 THEN 2040
2030 AC=90:GOTO 3000
2040 IF C<1 AND C>-1 THEN 2070
2050 AC=0:GOTO 3000
2060 REM THE ARCCOS IS AC
2070 AC=ATN((1-C*C)^.5/C)
2080 REM THE RESULT IS IN RADIANS. CONVERT TO DEGREES.
2090 AC=AC*180/π
3000 PRINT#5," SIN,COS       ARCSIN       ARCCOS"
3010 PRINT#5,S;
3020 FOR N=1 TO 13-LEN(STR$(S)):PPINT#5," ";:NEXT
3030 PRINT#5,AS;
3040 FOR N=1 TO 13-LEN(STR$(AS)):PRINT#5," ";:NEXT
3050 PRINT#5,AC
3100 GOTO 500
READY.
```

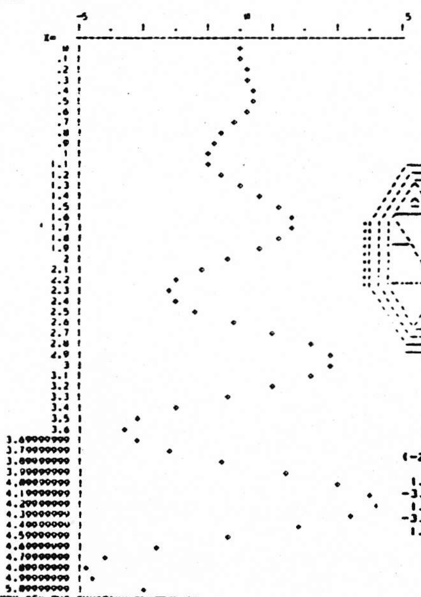




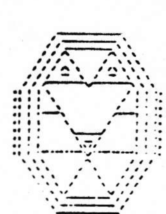
X	Y
0	0
.5	.33773159
1	-.996164609
1.5	1.1965018
2	-.348653564
2.5	-1.34143229
3	2.09697333
3.5	-3.10648462
4	1.37325972
4.5	1.71562719
5	-4.52789179

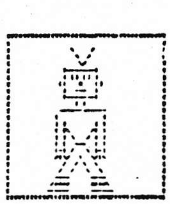
```
1000 REM THE FUNCTION PLOTTED IS:
1010 Y=X*SIN(4.0*X)
READY.
```

SIN,COS	ARCSIN	ARCCOS
.1	5.73917048	84.2608296
SIN,COS	ARCSIN	ARCCOS
.2	11.536959	78.463041
SIN,COS	ARCSIN	ARCCOS
.3	17.4576031	72.5423969
SIN,COS	ARCSIN	ARCCOS
.33333333	19.4712204	70.5287796
SIN,COS	ARCSIN	ARCCOS
.4	23.5781785	66.4218216
SIN,COS	ARCSIN	ARCCOS
.5	30	60
SIN,COS	ARCSIN	ARCCOS
.6	36.8698977	53.1301024
SIN,COS	ARCSIN	ARCCOS
.7	44.427004	45.572996
SIN,COS	ARCSIN	ARCCOS
.72591002	46.5446035	43.4553965
SIN,COS	ARCSIN	ARCCOS
.8	53.1301024	36.8698977
SIN,COS	ARCSIN	ARCCOS
.9	64.1580673	25.8419328
SIN,COS	ARCSIN	ARCCOS
.999999	89.9189618	.0810382363




```
000 REM THE FUNCTION PLOTTED IS:
010 Y=X*SIN(4.0*X)
ADY.
```





Powers of X

X	X^2	X^3	X^PI	X^5.25	X^.5
0	0	0	0	0	0
50	2500	125000	217506.221	830983735	7.07106781
100	10000	1000000	1919487.58	3.16227767E+10	10
150	22500	3375000	6861077.28	2.65753572E+11	12.2474487
200	40000	8000000	16939435.3	1.20339299E+12	14.1421356
250	62500	15625000	34146855.2	3.88315787E+12	15.8113883
300	90000	27000000	60548854.8	1.01131532E+13	17.3205081
350	122500	42875000	98271022.8	2.27173273E+13	18.7082869
400	160000	64000000	149490141	4.57946723E+13	20
450	202500	91125000	216427747	8.49895829E+13	21.2132034
500	250000	125000000	301345240	1.47772127E+14	22.3606798
550	302500	166375000	406540008	2.43727268E+14	23.4520788
600	360000	216000000	534342301	3.84852281E+14	24.4948974
650	422500	274625000	687112634	5.05861722E+14	25.4950976
700	490000	343000000	867239597	8.6449943E+14	26.4575131
750	562500	421875000	1.07713798E+09	1.24185712E+15	27.3861279
800	640000	512000000	1.31924718E+09	1.7426992E+15	28.2842713
850	722500	614125000	1.59602972E+09	2.39579338E+15	29.1547595
900	810000	729000000	1.98997007E+09	3.23424693E+15	30
950	902500	857375000	2.26357349E+09	4.29584822E+15	30.82207
1000	1000000	1E+09	2.65936506E+09	5.62341327E+15	31.6227766

Powers of X

X	X^2	X^3	X^PI	X^5.25	X^.5
0	0	0	0	0	0
50	2500	125000	217506.221	830983735	7.07106781
100	10000	1000000	1919487.58	3.16227767E+10	10
150	22500	3375000	6861077.28	2.65753572E+11	12.2474487
200	40000	8000000	16939435.3	1.20339299E+12	14.1421356
250	62500	15625000	34146855.2	3.88315787E+12	15.8113883
300	90000	27000000	60548854.8	1.01131532E+13	17.3205081
350	122500	42875000	98271022.8	2.27173273E+13	18.7082869
400	160000	64000000	149490141	4.57946723E+13	20
450	202500	91125000	216427747	8.49895829E+13	21.2132034
500	250000	125000000	301345240	1.47772127E+14	22.3606798
550	302500	166375000	406540008	2.43727268E+14	23.4520788
600	360000	216000000	534342301	3.84852281E+14	24.4948974
650	422500	274625000	687112634	5.05861722E+14	25.4950976
700	490000	343000000	867239597	8.64499043E+14	26.4575131
750	562500	421875000	1.07713798E+09	1.24185712E+15	27.3861279
800	640000	512000000	1.31924718E+09	1.7426992E+15	28.2842713
850	722500	614125000	1.59602972E+09	2.39579338E+15	29.1547595
900	810000	729000000	1.98997007E+09	3.23424693E+15	30
950	902500	857375000	2.26357349E+09	4.29584822E+15	30.82207
1000	1000000	1E+09	2.65936506E+09	5.62341327E+15	31.6227766

(-2)^(50-X/10)	(-2)^(50-X/10)
1.12589991E+15	1.12589991E+15
-3.51843721E+13	-3.51843721E+13
1.09951163E+12	1.09951163E+12
-3.43597384E+10	-3.43597384E+10
1.07374182E+09	1.07374182E+09
-33554432	-33554432
1048576	1048576
-32768	-32768
1024	1024
-32	-32
1	1
-.03125	-.03125
9.765625E-04	9.765625E-04
-3.05175781E-05	-3.05175781E-05
9.53674317E-07	9.53674317E-07
-2.98023224E-08	-2.98023224E-08
9.31322575E-10	9.31322575E-10
-2.91038305E-11	-2.91038305E-11
9.09494702E-13	9.09494702E-13
-2.84217094E-14	-2.84217094E-14
8.8817842E-16	8.8817842E-16

CONNECTICUT microCOMPUTER

150 POCONO RD. - BROOKFIELD, CT06804 - (203) 775-9659

PET I/O CA INTO PET ROM - by Chip Bitley

These calls assume that you have opened your files appropriately, and are only for the indicated primitive operations. Keyboard and screen I/O are covered in the PET Monitor document.

PET TAPE I/O FOR DATA

Routines:		
	Get an input channel	FFC6
	Get an output channel	FFC9
	Release channel	FFCC
	Input to BASIC	FFCF
	Output from BASIC	FFD2

To input a character:

```
JSR FFC6      get input channel
JSR FFCF      get character (in A register)
JSR FFCC      release channel
```

To output a character:

```
JSR FFC9      get output channel
JSR FFD2      put character (in A register)
JSR FFCC      release channel
```

When doing output, be sure to close the file to get the buffer onto the tape.

PET IEEE-488 I/O SUBSET

"FA" is the First Address, and is in 00F1 in RAM
"STATUS" is the BASIC "ST" and is in 020C in RAM

Routines:		
	TALK to Device FA	F0B6
	LISTEN to Device FA	F0BA
	UNLISTEN to all	F17E
	Output character w/ handshake	F0BC
	Input character w/handshake	F187

Note: Output buffers 1 char, so last char sent won't go onto Buss. Input char will set STATUS to $ 40 if EOI recieved. Both routines have character in A register.

To input a character:

```
JSR F0B6      talk
JSR F187      get char
JSR F17E      unlisten
```

To output a character:

```
JSR F0BA      listen
JSR F0BC      put char (see note)
JSR F17E      unlisten
```

WARNING: Chip Bitley hasn't had the time to check these out yet, but he believes they will work.

PET INPUT / OUTPUT LINES - by Chip Bitley

These tables relate the PET I/O lines to the internal PIA's and the VIA. See the 6522 and 6520 documents for how to control these lines.

KEYBOARD PIA (6520) Address: E810 59408

PAØ	Keyboard column select	PBØ	Keyboard row
PA1	" " "	PB1	" "
PA2	" " "	PB2	" "
PA3	" " "	PB3	" "
PA4	Switch, Cassette # 1	PB4	" "
PA5	" " # 2	PB5	" "
PA6	$\overline{\text{EOI}}$ in	PB6	" "
PA7	Diagnostic Jumper	PB7	" "
CA1	Read, Cassette # 1	CB1	Video Sync in
CB1	Blank Screen and $\overline{\text{EOI}}$ out	CB2	Motor, Cassette # 1

IEEE-488 PIA (6520) Address: E820 59424

PA0	IEEE Data In - 1	PB0	IEEE Data Out - 1
PA1	" " 2	PB1	" " 2
PA2	" " 3	PB2	" " 3
PA3	" " 4	PB3	" " 4
PA4	" " 5	PB4	" " 5
PA5	" " 6	PB5	" " 6
PA6	" " 7	PB6	" " 7
PA7	" " 8	PB7	" " 8
CA1	$\overline{\text{ATN}}$ in	CB1	$\overline{\text{SRQ}}$ in (not buffered)
CA2	$\overline{\text{NDAC}}$ out	CB2	$\overline{\text{DAV}}$ out

Note: All IEEE related lines are buffered except for SRQ in. This includes the lines in the VIA. CB1 of the keyboard PIA both blanks the screen and pulls EOI out low.

USER PORT VIA (6522) Address: E840 59456

PA0	Lsb of User Port	PB0	$\overline{\text{NDAC}}$ in
PA1	User Port	PB1	$\overline{\text{NRFD}}$ out
PA2	" "	PB2	$\overline{\text{ATN}}$ out
PA3	" "	PB3	Write (Both cassettes)
PA4	" "	PB4	Motor, Cassette # 2
PA5	" "	PB5	Video Sync in
PA6	" "	PB6	$\overline{\text{NRFD}}$ in
PA7	Msb of User Port	PB7	$\overline{\text{DAV}}$ in
CA1	User Port handshake	CB1	Read, Cassette # 2
CA2	Character set select (Graphics vs U/L case)	CB2	User Port handshake

NOTE: The Diagnostic LED will light if the Keyboard PIA is as follows: PAØ High, PA1 High, PA2 Low, PA3 High, Others Don't Care

PS: Chip isn't sure whether the IEEE control lines should have the bar on top or not. Try it yourself and see.