PET USERS GROUP NEWSLETTER

Volume 0 Number 5

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

P-A-T-C-H T-O "H-I-M-O-N-D-I-S"

A popular variation to Commodore's TIM or Machine Language Monitor program is HIMONDIS. HIMONDIS stands for the "HIgh memory MONItor and DISassembler" which emanated from the San Francisco Bay Area more than a year ago. No doubt your local user's group has a copy. If not, you will need to contact, preferably in person, either the SPHINX Users Group or PUG Users Group.

The original program had a bug; it could not LOAD binary programs like TIM. Here is the patch. Follow it exactly!

(1) Put your current version of HIMONDIS into your PET tape unit.
(2) Rewind the tape.
(3) Type: LOAD
(4) Press: "PLAY" on the tape unit.
(5) When PET types READY, press: "STOP" on the tape unit.
(6) Type: POKE 135,24
(7) Type: NEW
(8) Type: POKE 7377,10
(9) Type: POKE 7487,10
(10) Type: SYS 6671
(11) Now you are in HIMONDIS
(12) Put a new cassette into your PET tape unit.
(13) Rewind the tape.
(14) Type: S,01,HIMONDIS,1A00,2000
(15) Press: "RECORD" and "PLAY" on your tape unit.
(16) When PET is finished recording HIMONDIS, press: "STOP" on the tape unit.

Now you have the patched version of HIMONDIS. To use the L(oad) command, follow the instructions given in the TIM manual.

To load HIMONDIS, do steps (1) through (10) above, <u>without</u> (8) and (9).

THE BOOLEAN EXCLUSIVE-OR FUNCTION

Grimes Slaughter

Microprocessors implement the exclusive-or function in a variety of ways and for many purposes. The Boolean exclusive-or function of two binary variables is defined as true if and only if one or the other of the variables is true. For example, if the two binary functions are x and y, the exclusive-or for the different respective combinations (0,0); (0,1); (1,0); and (1,1) is defined as true for only the middle two combinations (0,1) and (1,0). One of the very necessary uses of the exclusive-or is for complementing the content of a register or memory location (changing ones to zeros and zeros to ones). This is done by taking the exclusive-or between the register or memory "word" to be complemented and a "word" which consists of all ones. This is done on a bit for bit basis, and the result gives zeros where the original word had ones, and ones where the original word had zeros.

Another ingenious use of the exclusive-or is described in BYTE magazine for November, 1978, p. 141. The exclusive-or is used to exchange the content of two memory locations without the use of a third temporary location. This is done in the following manner. Let the two memory locations have the content X and Y respectively, where X and Y are both n-bit words. Let x represent the i'th bit of X, and y represent the i'th bit of Y. Negation or the "NOT" function will be represented by underlines rather than overlines, and the "exclusive or" symbol by xor. The "NOT" of an expression in parentheses will be represented by not( ). The Boolean AND will be indicated by "x.y", and the OR by "x + y". These restrictions are a direct result of the limitations of the word processor used to type this manuscript (are you listening, word processor soft-ware writers?). According to BYTE, to interchange two memory locations all you do is

(1) $$X = X \text{ xor } Y$$
(2) $$Y = X \text{ xor } Y$$
(3) $$X = X \text{ xor } Y$$

and the job is done. The symbol "=" in this case does not mean "equal", but rather that the expression on the right replaces the expression on the left. Let me derive this result and see if it is correct. I will need a definition of the exclusive-or, and De Morgan's Theorem. There are several definitions of the exclusive-or, all of which are equivalent. I will choose to use the one which states that the exclusive-or between the i'th bits of two words X and Y is given by

(4) $$x.\underline{y} + \underline{x}.y.$$

The form of De Morgan's Theorem which I will use is the following:

(5) $$\underline{x + y + z} = \underline{x}.\underline{y}.\underline{z},$$

where a product is implied on the right side (word processor limitations again). This states in equation form the premise that if the OR of a group of binary variables is untrue (none of them is true) then each one must be untrue, and the AND of their NOT's must also be untrue. For example, the NOT of the "exclusive-or" is the other two states, or $x.y + \underline{x}.\underline{y}$. I will apply the "exclusive-or" to equations 1, 2, and 3 in turn.

The left side of equation one is replaced by

(6) $$x = x.\underline{y} + \underline{x}.y.$$

The left side of equation two is replaced by the exclusive-or between the new x above and y, which is given by

(7) $$y = y\ \mathrm{not}(x.\underline{y} + \underline{x}.y) + \underline{y}(x.\underline{y} + \underline{x}.y)$$

$$= y(x.y + \underline{x}.\underline{y}) + \underline{y}(x.\underline{y} + \underline{x}.y)$$

$$= y.x.y + y.\underline{x}.\underline{y} + \underline{y}.x.\underline{y} + \underline{y}.\underline{x}.y$$

Since $y.\underline{y} = 0$, $y.y = y$, and $(y + \underline{y}) = 1$, the above reduces to

$$y = (y + \underline{y})x = x$$

and the location which contained Y now contains X. Now I invoke equation three, remembering that y is now x:

(8) $$x = x\ \mathrm{not}(x.\underline{y} + \underline{x}.y) + \underline{x}(x.\underline{y} + \underline{x}.y)$$

$$= x.x.y + \underline{x}.x.\underline{y} + \underline{x}.x.y + \underline{x}.\underline{x}.y$$

$$= (x + \underline{x})y = y$$

and the interchange is complete without the use of an auxiliary memory location. I will give an example to show the process more clearly. The initial values of X and Y will be, in hex,

X = D3
Y = 7A

I will give the operations in both hex and binary. Remember that the values of X and Y change from line to line:

X = D3 xor 7A = A9
  = 11010011 xor 01111010 = 10101001

Y = A9 xor 7A = D3
  = 10101001 xor 01111010 = 11010011

X = A9 xor D3 = 7A
  = 10101001 xor 11010011 = 01111010

The numeric values are now interchanged, as promised by the derivation.

11010011
01111010
10101001

A BULLETPROOF (almost) STOP CODE FOR THE PET

Grimes Slaughter
Oak Ridge, TN

For applications in which the public interacts with a canned program, disabling of the STOP key has been described in the elegant bit of machine language programming by Richard Tobey (PUGN No. 1, p.11). However, when I am interacting with the machine, there are times when I fervently desire STOP key action, and it is completely ineffective. In such cases PET is still responding to the 60 Hz interrupt and things look normal, but the cursor has disappeared. The PRECODE in the abovementioned program can include a forced strobe of the STOP key, with a subsequent branch to the BASIC Ready routine if the key is down. Before listing the code I will describe the keyboard scanning routine.

The PIA (6520) dedicated to the keyboard and a few other items has, upon system initialization, the lowest four A-side lines defined as outputs. The remainder of the A-side lines and all of the B-side lines are defined as inputs. All of the B lines have 10k pull up resistors to the positive power supply, so location E812 will read FF unless some of the lines are pulled down by some means. The lowest four A lines are multiplexed by a four line to ten line decoder. One of the ten output lines goes low for each number from X0 to X9 which is loaded into location E810 (X means "don't care", but the value F is present in this location). No line is brought low for the "illegal" codes XA through XF. It is now apparent what needs to be done to interrogate the STOP key and branch out of the hang-up. The proper number has to be entered in the A-side lower four bits to strobe the correct one of ten lines, and the proper B-side bit has to be masked and tested for a zero. In the case of the STOP key, location E810 should be loaded with F9 and the number four bit in location E812 masked and tested for a zero. On the keyboard connector contents of F0 through F9 for location E810 correspond to lines 1 through 10, and B lines 0 through 7 correspond to letters A through H (PUGN No. 0, p. 4). The listing should now be more or less self-explanatory. For this particular purpose the POSTCODE is not needed, so the PRECODE branches directly back to the interrupt handler. The location C38B seems to be a good branch target for returning to the Ready mode. If you hold your pinky on the STOP key, it will branch to Ready 60 times a second. I believe that you will be glad enough to see the cursor back that you can ignore the low grade fireworks. SYS832 sets the STOP strobe mode, and SYS845 returns things to normal. Note that these transitions must be done under program control, since the change must be complete before the next interrupt. The code can recognize any combination of keys on the same 4-to-10 multiplexor line. If 18 is substituted in location 0369, then both the STOP key and the "less than" key two to the left must be pushed. However, the interrupt vector must be set back to normal before the cassette is used for input or output.

The "almost" in the title means that if you diddle with the interrupt vector, the code itself which resides in the second tape buffer, the control registers A or B in the PIA, or do an SEI, then all bets are off. I have found that the code will arrest most lapses and allow main memory to be saved. Sometimes BASIC returns in less than usable form, and that is another problem. If the initialization items are zapped, then main memory won't be worth much anyhow, and a system reset will be necessary. I strongly recommend a hardware reset switch which triggers the 555 one shot (PET User Notes No. 6, p. 10). Not only does the hardware reset prevent stress on the power supply and the CRT, but it does not erase the tape buffers.

Location	Content	Interp.	Remarks
0340	78	SEI	Set interrupt disable
0341	A9,63	LDA=63	Load A with value 63
0343	8D,19,02	STA at 0219	Store A at location 0219
0346	A9,03	LDA=03	
0348	8D,1A,02	STA at 021A	
034B	58	CLI	Clear interrupt disable
034C	60	RTS	Return from subroutine
034D	78	SEI	
034E	A9,85	LDA	
0350	8D,19,02	STA at 0219	
0353	A9,E6	LDA=E6	
0355	8D,1A,02	STA at 021A	
0358	58	CLI	
0359	60	RTS	
•			
•			
0363	A9,F9	LDA=F9	
0365	8D,10,E8	STA at E810	
0368	A9,10	LDA=10	
036A	2D,12,E8	AND	AND A with contents of E812
036D	F0,03	BEQ to 0372	Branch if equal to zero to 0372
036F	4C,85,E6	JMP	Jump to location E685 (interrupt rout.)
0372	4C,8B,C3	JMP	Jump to BASIC Ready routine

The start locations again are SYS832 for setting strobe and SYS845 for resetting the normal mode. I assume that the reader has access to some monitor program for putting the above into one of the cassette buffers and reading it onto tape. If not, the hex values shown can be converted to decimal and POKEd one at a time.

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

The following is an abridged list of publications that should be of interest to PET owners. As a group, they deal with both software and hardware applications of the PET 2001 and are rich in information. This is obviously not an exhaustive list; it represents a selection on the basis of quality of ten percent or less of PET-related publications. The list is divided into two categories:
(1) Resource Publications: non-periodic guides, manuals and compendia.
(2) Periodic Publications: journals and newsletters, (printed and on cassette).

(1) RESOURCE PUBLICATIONS

Title	Source	Price
Pet User Manual 2001-8 (150+ pages) P/N #320845	Commodore Inc. 901 California Ave. Palo Alto CA 94304	$30.00*
Pet Communication With the Outside World (included above as Chapter 7) (53 pages)	Commodore (as above)	?
Best of the Pet Gazette (96 pages)	Pet Gazette 1929 Northport Dr. Room 6 Madison WI 53704	10.00

(2) PERIODICALS

Title	Source	Price
Commodore Pet Users Club Newsletter (2 issues to date)	Commodore (as above)	15.00 (? issues)
Commodore Pet Users Club Newsletter (more than 3 issues to date)	Commodore 360, Euston Road London, N.W.1. 3BL United Kingdom	£ 10.00 (? issues)
Pet Gazette (6 issues to date)	Pet Gazette (as above)	Free! (bi-monthly)
Pet User Notes (7 issues to date)	Pet User Group P.O.Box 371 Montgomeryville PA 18936	6.00 (6-issues)
The XXX Paper (more than 6 issues to date)	The XXX Paper Box 43 Audubon PA 19407	15.00 (10-issues)
Computer Cassettes for the TRS-80, PET, Apple II	R.E.Pursor P.O.Box 466 El Dorado CA 95623	12.00 (quarterly)
Cursor Magazine (more than 6 issues to date in form of programs on a cassette)	Cursor Box 550 Goleta CA 93017	24.00 (monthly)

* - $3.50 with PET proof-of-purchase. Send Commodore your name and your PET's serial number.

P-E-R-I-P-H-E-R-A-L H-A-R-D-W-A-R-E

Commodore and a host of other manufacturers are producing peripherals to attach, both inside and outside, to your PET. The peripherals can be categorized into four types of hardware:

(1) Primary Storage, i.e., added RAM

(2) Secondary Storage
  (A) Cassettes, extra
  (B) Disks
  (C) Other, i.e., paper tape, punch cards

(3) Interfaces, i.e., PET to RS232, PET to S-100

(4) Special Input/Output
  (A) Input, i.e., light pen, switches, A to D
  (B) Output, i.e., printers, lights, D to A

It is not the editor's intent to review equipment from all the above types of peripherals. Each manufacturer usually can send a "Spec Sheet" to potential customers which should contain pertinent operating specifications. It is highly advisable before you buy a peripheral that,

i. You see it in operation.
ii. There exists a specified warranty period.
iii. You have access to a known repair station.
iv. You can return the equipment for a full refund within a stated period of time.

The single most useful peripheral in the editor's opinion is a fast secondary storage device, namely a disk. Bear in mind that disk hardware alone is not sufficient to provide for loading and saving of program and data files, and that the quality of a manufacturer's software (Disk Operating System) is critically important for the performance of the disk. Products listed below are by hardware name/software name. We know of four disk systems for the PET; three are in production:

DISK DRIVER/DISKMON

Compu/Think
3260 Alpine Road
Menlo Park CA 94025
(415) 854-2577

Dual disk, single head with DISKMON	200K bytes user space	$1295.00
Dual disk, double head with DISKMON	400K bytes user space	$1645.00

Both need an "Expandamem" (formerly named Expandapet) internal additional RAM memory. Both include a Disk Operating System called DISKMON by Compu/Think that adds 16 commands to PET Basic including data files. The strength of the Compu/Think system is in the completeness and flexibility of their DOS.

DISK/KM3

CGRS Microtech
P.O.Box 368
Southampton PA 18966
(215) 757-0284

Single disk with KM3	80K bytes user space	$ 799.95

CGRS offers more expensive models that extend the S-100 chassis to 3 and 10 slots. The KM3 DOS minimally extends PET Basic to load and save program and data files.

204X NOT RELEASED AS OF THIS DATE

Commodore Business Machines, Inc.
901 California Avenue
Palo Alto CA 94304 USA
(415) 326-4000

2041 Single drive (a)	170K bytes user space	$ 595.00
2040 Dual drive (b)	340K bytes user space	$1095.00

(a) A DOS is provided on two ROMs that plug inside a 16K or 32K PET made by Commodore or are retrofited to a standard 8K PET.
(b) The 2040 is a self-contained, internal 6502 & 6504 micro-processor-driven, intelligent peripheral. It is plug compatible with an 8K, 16K and 32K PET.

CLUSTER/ONE

Nestar Systems Inc.
430 Sherman Ave.
Palo Alto CA 94306
(415) 327-0125

Dual 8 inch drives	315K bytes users space	$4500.00 +

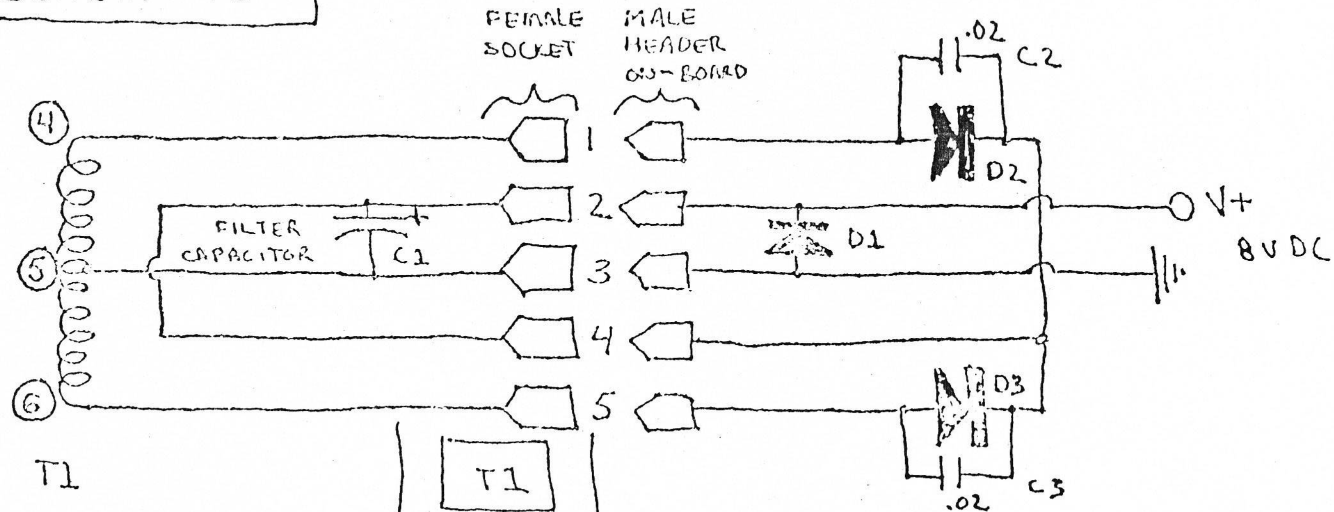
The CLUSTER/ONE supports 15 to 30 PETs, Apples and TRS-80s. It features shared program files and a very reliable data bus with transfer rates equaling the single user systems listed above. Data files not yet available.

CLUSTER/ONE is clearly more expensive than the other disk systems. Economic advantages to CLUSTER/ONE would be experienced in a lab with four or more disk-based PETs and/or Apples and/or TRS-80s all sharing a common dual disk file system.

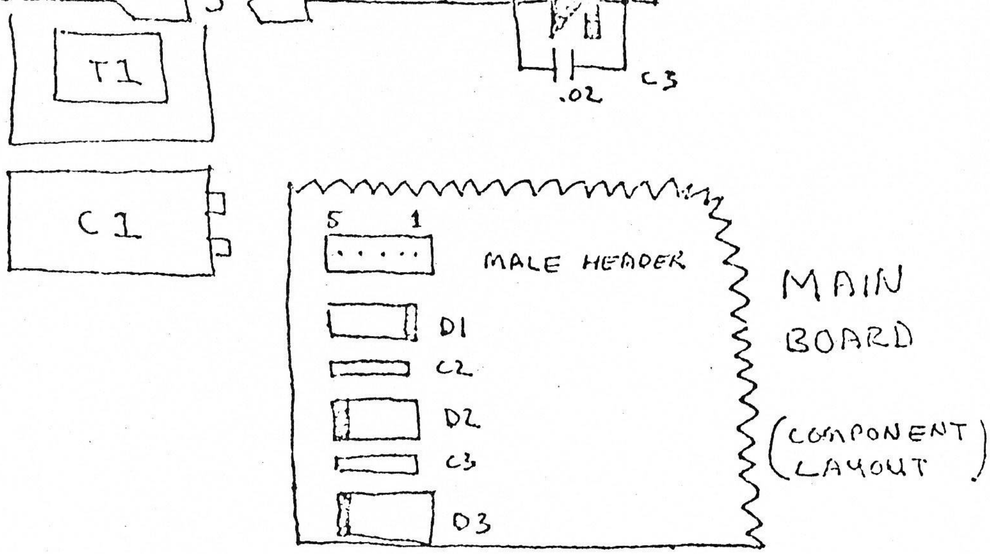
P-E-T P-O-W-E-R S-U-P-P-L-Y

Here are two schematics. The first one is a simplified schematic of your PET's simple power supply. The second schematic offers the user a way to add an extra 5 volt output for external applications using less than one amp. CAUTION: The added power supply connects to the innards of your PET. Result: your warranty is voided.

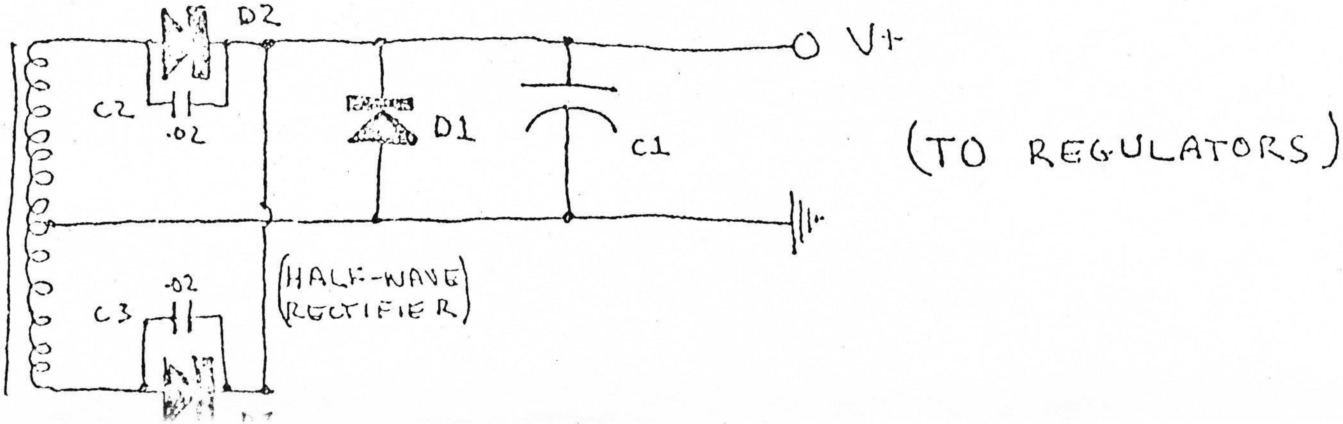
CONNECTOR SCHEMATIC



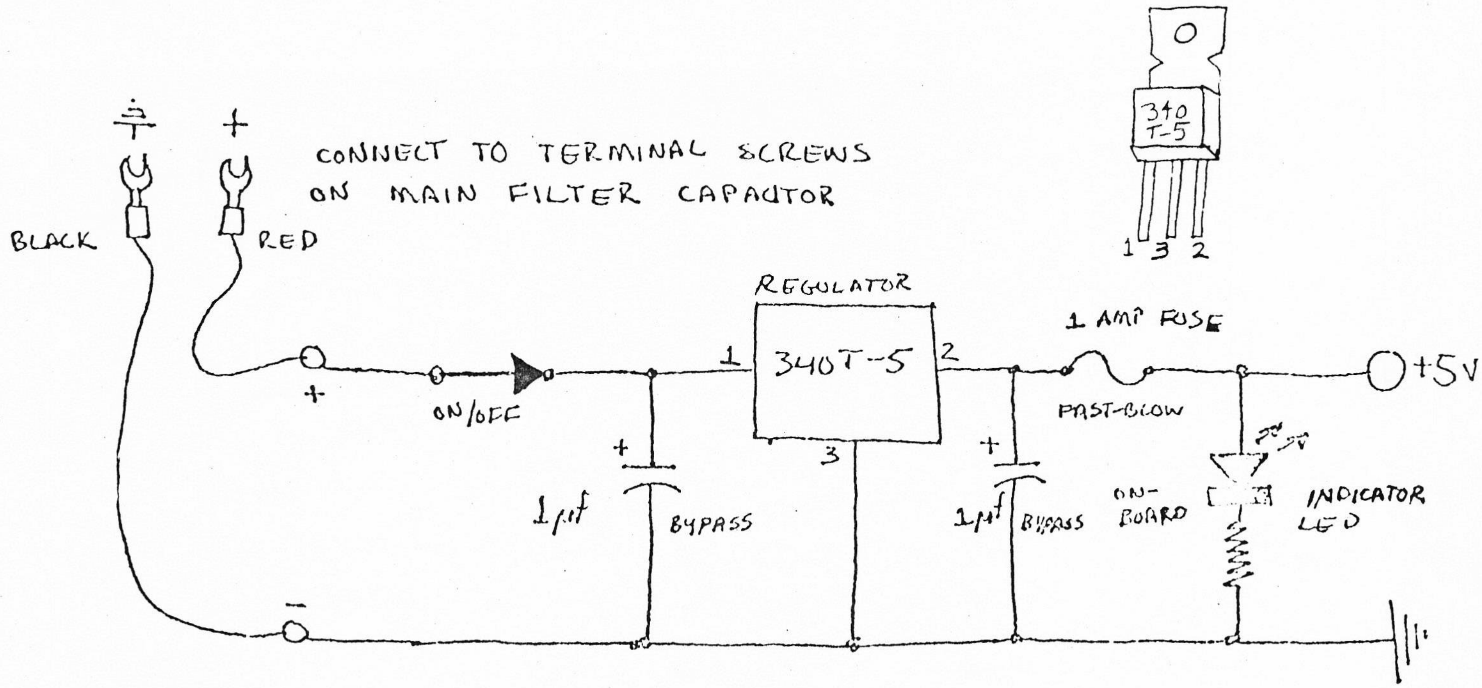
COMPONENT LAYOUT



SIMPLIFIED SCHEMATIC



PET POWER SUPPLY: 1 AMP EXPANSION



RESET BUTTON FOR YOUR PET

by Milt Lee

Many of us have been wearing out our on/off switch to restart our machines when the operating system gets lost. Playing with machine code can be especially hazardous as no errors are tolerated and all of our code is lost.

There is a way to reset the PET and save one page (256 bytes) of data! By grounding the $\overline{\text{RES}}$ line of the microprocessor the operating system is reset as if on a power up except page 3, locations 768 to 1024 decimal or 0300 to 03FF Hex, are left untouched. Since the second cassette buffer is on this page, machine language programs using this space can be saved even if they bomb the machine.

Connecting a normally open pushbutton between expansion port pins A27 and B27 is all that is needed. Wires could also be soldered on the main board to pick up the $\overline{\text{RES}}$ circuit. ($\overline{\text{RES}}$ is pin 40 or the farthest right pin on the 6502 chip as seen from the front of the PET.

Try SYS(64824) to get the same effect; however that doesn't do much good if the PET is not scanning the keyboard!