# PET USER'S GROUP NEWSLETTER

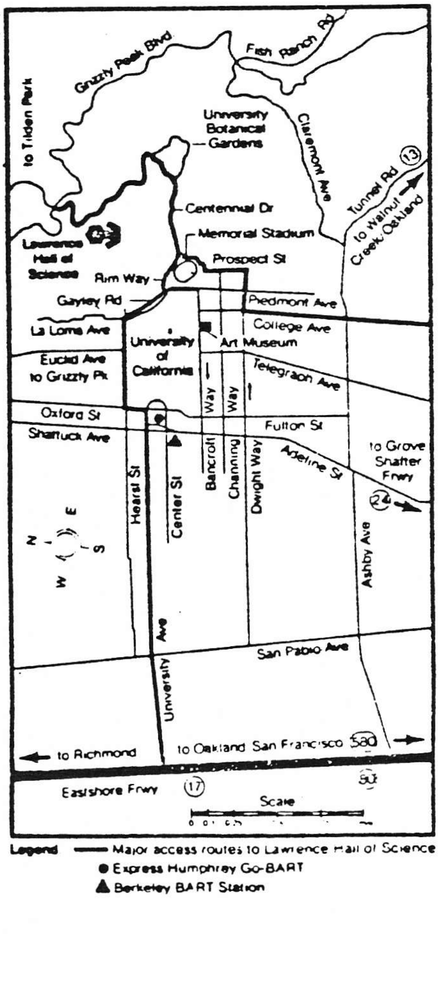
VOLUME 0 NUMBER 0

On January 25, the PET User's Group had its second meeting in Cupertino at the VanderKooi's home. The Group attempted to create a name for itself, but was unable to reach a decision and has postponed further discussion until the next meeting. Some people from Commodore were again present that gave some software insight and made overtures to discuss hardware I/O structure at a later time.

The next meeting will be held at Lawrence Hall of Science, above the University of California campus and above the Lawrence Berkeley Labs, on February 8, Wednesday at 7:00 p.m. Directions are included below for finding the Hall. Allow 80 minutes travel time from the Palo Alto, Mt. View, Sunnyvale areas. Please bring your ideas for a name for the group.

## TABLE OF CONTENTS

1. How to Sense Held-Down Keys
2. Peninsula School Software Ad
3. Principal Pointers into PET RAM
4. PET BASIC Statement Chain
5. PET Matrix-Decoded Keyboard
6. Interesting Locations in BASIC
7. ASC/CHR, PEEK/POKE Character Set
8. Insertion Mode, Special Characters
9. PET Hardware Notes
10. Brief Sketch of PET I/O



## HOW TO SENSE HELD-DOWN KEYS

by Harry Saal

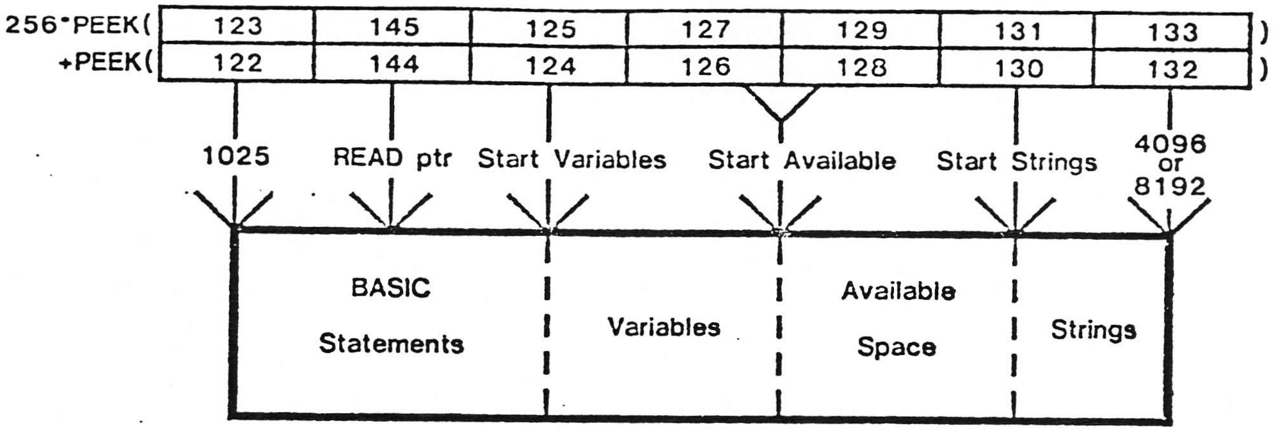
By detecting a key that is held down you can implement full cursor controls. Location 515 (hex 203) contains a number which indicates the keyboard state during the last screen retrace (every 60th of a second). A value of 255 indicates no key is down. Otherwise a value from 1 to 80 indicates a key is down. The values relate to the particular sequence keys are scanned by the PET monitor.

To determine the codes PET uses, run this program and push keys:

```
100 A=PEEK(515): IF A=255 THEN 100
200 PRINT A: GOTO 100
```

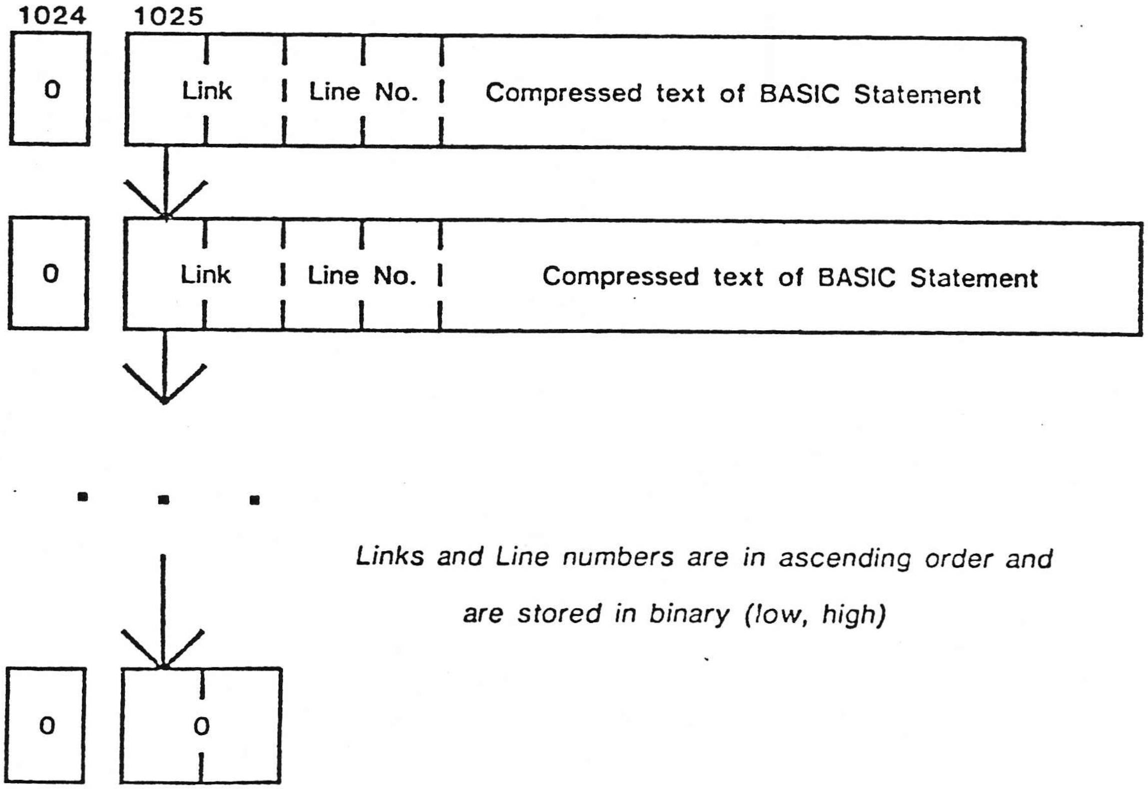
NOTE: You cannot sense the STOP key due to the manner the PET monitor handles scanning. Check the PET Matrix-Decoded Keyboard in this newsletter for all the codes and location 516 for the SHIFT key.

## Principal Pointers into PET RAM



*The READ pointer starts at 1024*
*When it is not pointing into a DATA statement*
*it resides at the 0-byte after a DATA statement*

## PET BASIC Statement Chain


1024
1025
0
Link
Line No.
Compressed text of BASIC Statement
0
Link
Line No.
Compressed text of BASIC Statement
0
0

*Links and Line numbers are in ascending order and*

*are stored in binary (low, high)*

## PET Matrix-Decoded Keyboard

*See 515 & 516 in table below*

| | | 8 | | 7 | | 6 | | 5 | | 4 | | (3) | | 2 | | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 64 | ! | " | # | S | % | ' | & | \ | ( | ) | ← | | ho | ↑↑ | ⇄ | de |
| 48 | Q | W | E | R | T | Y | U | I | O | P | ↑ | | 7 | 8 | 9 | / |
| 32 | A | S | D | F | G | H | J | K | L | : | | | 4 | 5 | 6 | • |
| 16 | Z | X | C | V | B | N | M | , | : | ? | re | | 1 | 2 | 3 | + |
| 0 | sh | rv | @ | [ | ] | sp | | < | > | st | sc | | 0 | . | - | = |
| | 16 | | 15 | | 14 | | 13 | | 12 | | 11 | | 10 | | 9 | |

## Interesting Locations Accessible from BASIC

| Location (decimal) | Contents |
|---|---|
| 225, 224<br>226 | Byte address of screen line with Cursor<br>Character position of Cursor (0 to 79) |
| 515 | Matrix-coordinate (row+column) of last key down<br>255 if no key down |
| 516 | 1 if shift down, 0 if shift up |
| 525<br>526-534 | No. of characters in Keyboard Buffer<br>Keyboard Buffer |
| 578 to 587<br>588 to 597<br>598 to 607<br>610 | Logical numbers of open files<br>Device numbers of open files<br>Read/write modes of open files<br>How many open files |
| 512, 513, 514<br>518, 517<br>59465, 59464 | Clock that increments 60 times a second<br>Clock that increments 30 times a second?<br>Clock that decrements every microsecond |
| 59456 | WAIT 59456,32,32 waits for vertical retrace of display |
| 64824 | SYS(64824) simulates power-on reset |
| 59469 | Interrupt Flag Register; e.g., to input user port CA1:<br>I=PEEK(59469) AND 2: POKE 59469,I: IF I=0 THEN *CA1 low* |
| 59411 | IEEE PIA B Control, e.g., to run cassette#1 motor N jiffies:<br>100 POKE 59411,53: T=TI<br>200 IF TI-T<N GOTO 200<br>300 POKE 59411,61<br>ADVICE: *Run motor at least 3 jiffies per 191 output chars* |

| Character | ASC/CHR | PEEK/POKE | Character | ASC/CHR | PEEK /POKE | Character | ASC/CHR | PEEK/POKE | Character | ASC/CHR | PEEK/POKE |
|---|---|---|---|---|---|---|---|---|---|---|---|
|  | 0 |  | @ | 64 | 0 |  | 128 |  | ─ 5 | 192 | 64 |
|  | 1 |  | A | 65 | 1 |  | 129 |  | ♠, a | 193 | 65 |
|  | 2 |  | B | 66 | 2 |  | 130 |  | 4 │, b | 194 | 66 |
|  | 3 |  | C | 67 | 3 |  | 131 |  | 4 ─, c | 195 | 67 |
|  | 4 |  | D | 68 | 4 |  | 132 |  | 3 ─, d | 196 | 68 |
|  | 5 |  | E | 69 | 5 |  | 133 |  | 2 ─, e | 197 | 69 |
|  | 6 |  | F | 70 | 6 |  | 134 |  | 6 ─, f | 198 | 70 |
|  | 7 |  | G | 71 | 7 |  | 135 |  | 3 │, g | 199 | 71 |
|  | 8 |  | H | 72 | 8 |  | 136 |  | 6 │. h | 200 | 72 |
|  | 9 |  | I | 73 | 9 |  | 137 |  | ╮, i | 201 | 73 |
|  | 10 |  | J | 74 | 10 |  | 138 |  | ╰, j | 202 | 74 |
|  | 11 |  | K | 75 | 11 |  | 139 |  | ╯, k | 203 | 75 |
|  | 12 |  | L | 76 | 12 |  | 140 |  | ⌞, l | 204 | 76 |
| RETURN | 13 |  | M | 77 | 13 | RETURN | 141 |  | ╲, m | 205 | 77 |
|  | 14 |  | N | 78 | 14 |  | 142 |  | ╱, n | 206 | 78 |
|  | 15 |  | O | 79 | 15 |  | 143 |  | ⌜, o | 207 | 79 |
|  | 16 |  | P | 80 | 16 |  | 144 |  | ⌝, p | 208 | 80 |
| ⇩ | 17 |  | Q | 81 | 17 | ⇧ | 145 |  | ●, q | 209 | 81 |
| RVS | 18 |  | R | 82 | 18 | RVSoff | 146 |  | 2 ─, r | 210 | 82 |
| HOME | 19 |  | S | 83 | 19 | CLEAR | 147 |  | ♥, s | 211 | 83 |
| DEL | 20 |  | T | 84 | 20 | INST | 148 |  | 2 │, t | 212 | 84 |
|  | 21 |  | U | 85 | 21 |  | 149 |  | ╭, u | 213 | 85 |
|  | 22 |  | V | 86 | 22 |  | 150 |  | ╳, v | 214 | 86 |
|  | 23 |  | W | 87 | 23 |  | 151 |  | ○, w | 215 | 87 |
|  | 24 |  | X | 88 | 24 |  | 152 |  | ♣, x | 216 | 88 |
|  | 25 |  | Y | 89 | 25 |  | 153 |  | 7 │, y | 217 | 89 |
|  | 26 |  | Z | 90 | 26 |  | 154 |  | ♦, z | 218 | 90 |
|  | 27 |  | [ | 91 | 27 |  | 155 |  | ┼ | 219 | 91 |
|  | 28 |  | \ | 92 | 28 |  | 156 |  | ▒ | 220 | 92 |
| ⇨ | 29 |  | ] | 93 | 29 | ⇦ | 157 |  | 5 │ | 221 | 93 |
|  | 30 |  | ↑ | 94 | 30 |  | 158 |  | π, ▒ | 222 | 94 |
|  | 31 |  | → | 95 | 31 |  | 159 |  | ◥, ▨ | 223 | 95 |
| space | 32 | 32 | space | 96 | 32 |  | 160 |  | □ | 224 | 96 |
| ! | 33 | 33 | ! | 97 | 33 | ▌ | 161 | 97 | ▌ | 225 | 97 |
| " | 34 | 34 | " | 98 | 34 | ▄ | 162 | 98 | ▄ | 226 | 98 |
| # | 35 | 35 | # | 99 | 35 | ▔ 1 | 163 | 99 | ▔ 1 | 227 | 99 |
| $ | 36 | 36 | $ | 100 | 36 | ▁ 8 | 164 | 100 | ▁ 8 | 228 | 100 |
| % | 37 | 37 | % | 101 | 37 | ▏ 1 | 165 | 101 | ▏ 1 | 229 | 101 |
| & | 38 | 38 | & | 102 | 38 | ▒ | 166 | 102 | ▒ | 230 | 102 |
| ' | 39 | 39 | ' | 103 | 39 | ▕ 8 | 167 | 103 | ▕ 8 | 231 | 103 |
| ( | 40 | 40 | ( | 104 | 40 | ▒ | 168 | 104 | ▒ | 232 | 104 |
| ) | 41 | 41 | ) | 105 | 41 | ◤, ▨ | 169 | 105 | ◤, ▨ | 233 | 105 |
| * | 42 | 42 | * | 106 | 42 | ▕ 1/4 | 170 | 106 | ▕ 1/4 | 234 | 106 |
| + | 43 | 43 | + | 107 | 43 | ├ | 171 | 107 | ├ | 235 | 107 |
| , | 44 | 44 | , | 108 | 44 | ▗ | 172 | 108 | ▗ | 236 | 108 |
| - | 45 | 45 | - | 109 | 45 | └ | 173 | 109 | └ | 237 | 109 |
| . | 46 | 46 | . | 110 | 46 | ┐ | 174 | 110 | ┐ | 238 | 110 |
| / | 47 | 47 | / | 111 | 47 | ▂ 1/4 | 175 | 111 | ▂ 1/4 | 239 | 111 |
| 0 | 48 | 48 | 0 | 112 | 48 | ┌ | 176 | 112 | ┌ | 240 | 112 |
| 1 | 49 | 49 | 1 | 113 | 49 | ┴ | 177 | 113 | ┴ | 241 | 113 |
| 2 | 50 | 50 | 2 | 114 | 50 | ┬ | 178 | 114 | ┬ | 242 | 114 |
| 3 | 51 | 51 | 3 | 115 | 51 | ┤ | 179 | 115 | ┤ | 243 | 115 |
| 4 | 52 | 52 | 4 | 116 | 52 | ▎ 1/4 | 180 | 116 | ▎ 1/4 | 244 | 116 |
| 5 | 53 | 53 | 5 | 117 | 53 | ▍ 1/3 | 181 | 117 | ▍ 1/3 | 245 | 117 |
| 6 | 54 | 54 | 6 | 118 | 54 | ▐ 1/3 | 182 | 118 | ▐ 1/3 | 246 | 118 |
| 7 | 55 | 55 | 7 | 119 | 55 | ▔ 1/4 | 183 | 119 | ▔ 1/4 | 247 | 119 |
| 8 | 56 | 56 | 8 | 120 | 56 | ▀ 1/3 | 184 | 120 | ▀ 1/3 | 248 | 120 |
| 9 | 57 | 57 | 9 | 121 | 57 | ▃ 1/3 | 185 | 121 | ▃ 1/3 | 249 | 121 |
| : | 58 | 58 | : | 122 | 58 | ▕, ☑ | 186 | 122 | ▕, ☑ | 250 | 122 |
| ; | 59 | 59 | ; | 123 | 59 | ▖ | 187 | 123 | ▖ | 251 | 123 |
| < | 60 | 60 | < | 124 | 60 | ▝ | 188 | 124 | ▝ | 252 | 124 |
| = | 61 | 61 | = | 125 | 61 | ┘ | 189 | 125 | ┘ | 253 | 125 |
| > | 62 | 62 | > | 126 | 62 | ▘ | 190 | 126 | ▘ | 254 | 126 |
| ? | 63 | 63 | ? | 127 | 63 | ▚ | 191 | 127 | ▚ | 255 | 127 |

CURSOR CONTROL
PET SPECIAL CHARACTERS INSERTION MODE

Notation of special char.
Jan-Feb '78 PCC Phyllis Cole

AFTER TYPING AN ODD NUMBER OF QUOTE MARKS

| KEY | SCREEN SHOWS | ASC No. | NOTATION OF SPECIAL CHAR. |
|---|---|---|---|
| OFF<br>RVS | ⊟ | 146 | OFF |
| OFF<br>RVS | R | 18 | RVS |
| RUN<br>STOP | C | 3 | STOP |
| CLR<br>HOME | ♥ | 147 | CLR |
| CLR<br>HOME | S | 19 | HOME |
| ⇧<br>CRSR<br>⇩ | ● | 145 | UP |
| ⇧<br>CRSR<br>⇩ | Q | 17 | DOWN |
| ⇐<br>CRSR<br>⇒ | ▕ | 157 | LEFT |
| ⇐<br>CRSR<br>⇒ | ] | 29 | RIGHT |
| INST<br>DEL | ▏ | 148 | INST |

[ROUGH DRAFT] CORRECTIONS, ADDITIONS, COMMENTS TO ME

PET HARDWARE NOTES

by Richard Tobey
408 733 0688

## I. I/O HARDWARE

There is much more to the PET I/O than meets the eye ( or is mentioned in the owner's manual!).

The PET has three LSI I/O chips -- two of the well known PIAs (6820, 6520, etc.) and one of the lesser known but more powerful VIAs (6522).

The PIAs look to the rest of the PET like 4 memory locations each, the VIA like 16 memory locations (see table 1).

For further details, consult the appropriate data sheets.(The VIA alone takes up 24 pages of heavy material to describe all its capabilities. These notes just scratch the surface.)

### A. User Port

The user port connected to one half of the VIA, PA0 to 7, are bidirectional data lines and CA1 is a bidirectional control line for handshaking. After reset it is set up as an input port and the appropriate register must be poked to set it up for output.

Example: POKE 59459,15 sets PA0 to 3 as outputs, PA4 to 7 as inputs.

CB2 is designed for use with the shift register in the VIA. (Anyone working out a good routine to use it thus, please let me know!) Incidentally, the shift register in the VIA is <u>not</u> used by the PET.

### B. IEEE Port

The IEEE Port uses most of a PIA, having the A register initialized for input and the B register initialized for output.

The IEEE Port uses some lines from other I/O chips as well. 12 of the 15 IEEE lines (REN is not implemented) pass through Bus-Driver /Receiver strips (see schematic). The drivers are always enabled and are open collector drivers having considerable sink capability (48 MA). As a source, however, they have only termination resistance.

Example: PEEK (59424) reads the IEEE Port. If not connected,
1 it will read all 1s due to the terminations.

Example: POKE 59426,A writes to the Port.
2

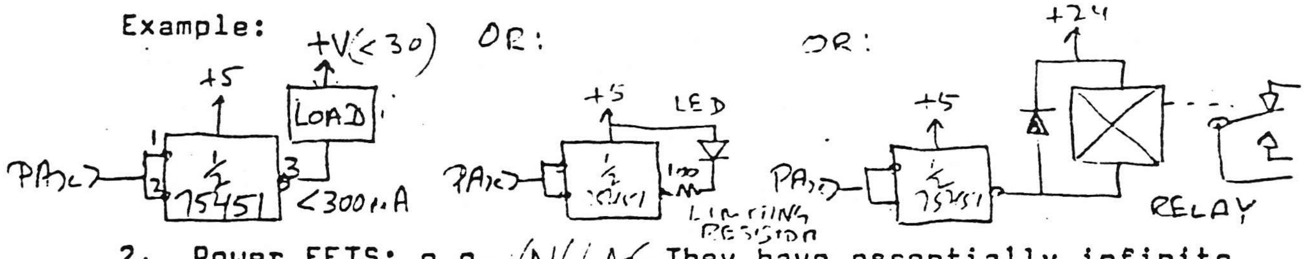
The second PIA scans the keyboard, and the other half of the VIA performs miscellaneous functions such as upper/lower case control and cassette I/O.

## II. INCREASING I/O DRIVE CAPABILITY

Except for the IEEE Bus, the I/O lines are limited to driving about one TTL load or equivalent. Useful buffers are:

1. Peripheral driver chips such as the 75451. These can sink 300 MA for control of relays, small lamps, etc. They can stand 30V when off.

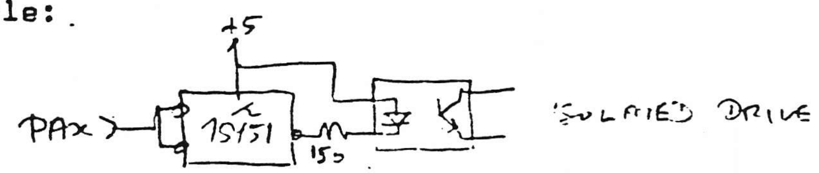
Example:



2. Power FETS: e.g. VN66AF They have essentially infinite input impedance, 80V breakdown and about 3 ohm on resistance.

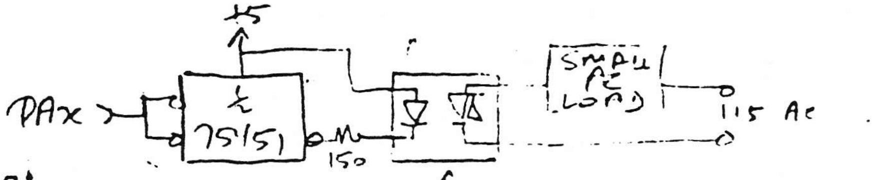
3. Opto couplers: These devices are useful when driving something which cannot share a common ground with the PET. Most types need more input current than can be supplied by the VIA. Use a peripheral Driver as well!

Example:

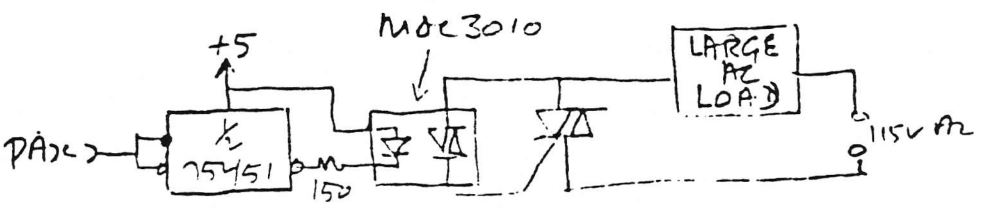


An interesting new part is the Opto coupled triac MOC3010 (Motorola)

Example:
1



Example:
2



Don't miswire or take my pin connections as gospel -- your PET won't like 115V AC on the I/O.

## III. INTERRUPT STRUCTURE

Interrupts (including Break or Software Interrupts) are handled by software polling.

When the processor recognizes an interrupt it vectors through FFFE, FFFF in ROM to a routine that first inspects the processor status register to see if it was caused by a Break instruction or by hardware (IRQ line low).

If it was caused by a Break instruction, a Jump Indirect is executed through locations 021B, C. If by a hardware interrupt then a Jump Indirect is taken through locations 0219, A.

These locations being in RAM may be user-modified to point to extra user code ahead of normal interrupt processing.

Note, however that the IRQ pointer is used by the cassette routines and should be restored to standard values before the cassette Save or Load functions are called.

Various sections of the I/O chips can be set up to cause interrupts through the IRQ line.

Example: POKE 59470,2 enables a negative edge on the user port CA1 line to cause an interrupt.

However, have your code set up to handle it when it happens!

Also note that each pass through the regular interrupt code increments the time register.

## IV. EXPANSION PORT

This port is an extension of the Address and Data Bus of the PET together with some timing signals and a hardware reset input.

The 4 high-order Addresses have already been decoded into 4K block selects.

The lines are not dedicated for expansion or as strong as S100 Bus lines -- so beware! Anything hanging on the expansion port should be fully buffered, preferably with low power Schottky.

The PET regulated power supply does not appear to have anything extra for expansion devices so they will need their own supply if they draw more than say 100 MA.

However, with these problems taken care of, Memory cards such as S100 8K by Godbout (and others) will work well with minor modification.

IV. (continued)

Data should be enabled onto the Data Bus through Tri State drivers during BOARDSELECT.ϕ2.R/W.

The memory write pulse should be generated during BOARDSELECT.$\phi 2.\overline{R/W}.\overline{PROTECT}$.

Memory sizing is automatic to the top of contiguous memory on Power up.

## V. I/O REGISTERS

VIA (Includes User Port) E840 - E84F

| | | |
|---|---|---|
| 59457 | ORA | Output register for User Port with handshake. |
| 59459 | DDRA | Data Direction reg. for ORA. 0=IN 1=OUT. |
| 59466 | SR | Shift Register (can connect to CB2). |
| 59468 | PCR | Peripheral Control Register. B0 is CA1 control, 0=Neg. edge, 1=Pos. edge sensitive. |
| 59469 | IFR | Interrupt Flag Register. CA1 sets B1, reset by read of 59457. |
| 59470 | IER | Interrupt Enable Register. B1=1 enables interrupt by CA1. |
| 59471 | ORA | Output register without handshake. |

Other registers are associated with the timers and other functions. See MCS6522 data sheets.

PIAs E810 - E813 and E820 - E823

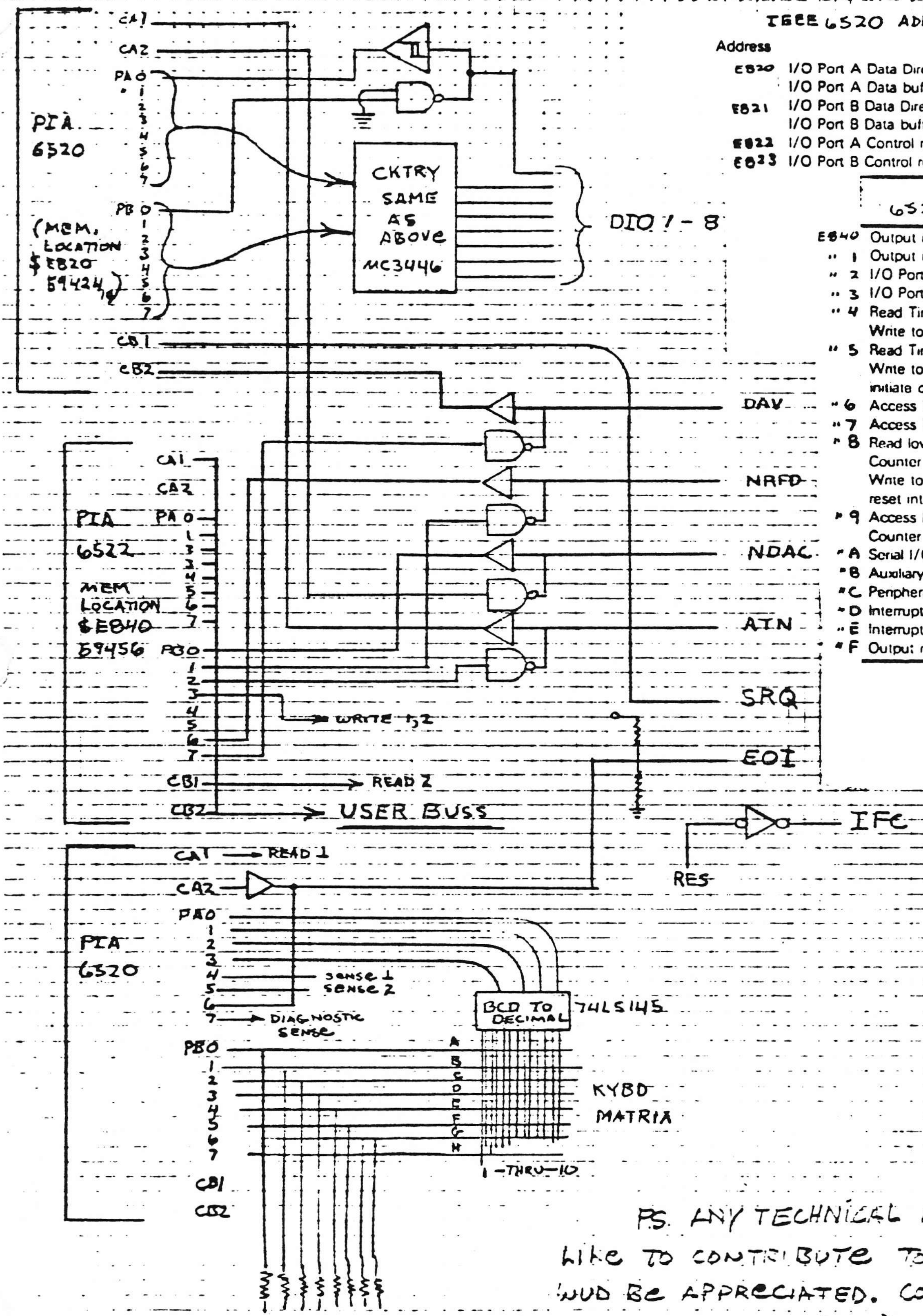
| KEYBOARD | IEEE | |
|---|---|---|
| 59408 | 59424 | A Register (also A Direction with CRA2=1). |
| 59409 | 59425 | A Control. |
| 59410 | 59426 | B Register (Also B Direction with CRB2=1). |
| 59411 | 59427 | B Control. |

The IEEE PIA is initialized for the A Register 59424 to read, and the B Register 59426 to output DIO1 to 8.

TABLE 1

# BRIEF SKETCH OF PET I/O

SNIFFED OUT BY RICHARD TOBEY



IEEE 6520 ADDRESSES

| Address | Selected |
|---|---|
| E820 | I/O Port A Data Direction register, if C002$_{16}$ bit 2 = 0 |
| | I/O Port A Data buffer, if C002$_{16}$ bit 2 = 1 |
| E821 | I/O Port B Data Direction register, if C003$_{16}$ bit 2 = 0 |
| | I/O Port B Data buffer, if C003$_{16}$ bit 2 = 1 |
| E822 | I/O Port A Control register |
| E823 | I/O Port B Control register |

| 6522 | ADDRESSED LOCATION |
|---|---|
| E840 | Output register for I/O Port B |
| " 1 | Output register for I/O Port A, with handshaking |
| " 2 | I/O Port B Data Direction register |
| " 3 | I/O Port A Data Direction register |
| " 4 | Read Timer 1 Counter low order byte<br>Write to Timer 1 Latch low order byte |
| " 5 | Read Timer 1 Counter high order byte<br>Write to Timer 1 Latch high order byte and initiate count |
| " 6 | Access Timer 1 Latch low order byte |
| " 7 | Access Timer 1 Latch high order byte |
| " 8 | Read low order byte of Timer 2 and reset Counter interrupt<br>Write to low order byte of Timer 2 but do not reset interrupt |
| " 9 | Access high order byte of Timer 2; reset Counter interrupt on write |
| " A | Serial I/O Shift register |
| " B | Auxiliary Control register |
| " C | Peripheral Control register |
| " D | Interrupt Flag register |
| " E | Interrupt Enable register |
| " F | Output register for I/O Port A, without handshaking |

P.S. ANY TECHNICAL ADDITIONS YOU LIKE TO CONTRIBUTE TO THE GROUP WUD BE APPRECIATED. CONTACT:

RICHARD TOBEY (408) 733-0688
NEIL BUSSEY (415) 451-6364