

* VZ - LINK *

VZ - LINK

NEWSLETTER

» AUCKLAND VZ300/200 USERS CLUB «

ISSUE No: 13

OCTOBER 1987

VZ NEWS

As you will notice ,i have changed the format of the front cover. I hope you like it,it will help me to fit just that little more into each issue.Any comment would be great.I hope you found the'Understanding Your VZ'item helpful,part two is in this issue.Also in this issue you will find part 3 of 'silver mountain',and two very handy programs which i hope you will find a use for.I also hope you will find use for the very fine item written by Joe Leon which shows some simple Modifications which can be done on the VZ.

There is not much news on the VZ scene at the moment.I can tell you that some D.S.E stores now have copies of the diskbased "DATABASE" program which cost \$39-95.You will find that this is a good program, but if you really want a super database program then i suggest you read advert. on page10 of this newsletter.I have a copy of "DATA" and it is really worth while .If you want to see before you buy let me know,maybe we can arrange something.(remember add \$2-00 for postage etc when ordering from Australia.)

Peter Paterson has just received some of the programs sold by 'laserlink!' and he inform's me that they are really good(maybe peter will write some reviews for us).

Remember this is your newsletter so don't forget that your input is very important and i will use what ever you send in.As somebody once said "don't be frightened".

Finally,I would like to endorse comment's made by other VZ newsletter editor's,about some of the articles supplied by readers to our various newsletters.In a few cases,these articles appear in other VZ publications In all cases,the articles are reprinted by permission of the author. Although those who subscribe to all such publications may find the repetition annoying ,it must be remembered that there are those who depend on a single source of information.We welcome contributions of any form from readers.

SEE SEPARATE PAGE FOR DETAILS OF VZ MEETING

UNDERSTANDING YOUR VZ . . . PART TWO.

THIS IS A CONTINUATION OF THE ARTICLE STARTED IN THE LAST EDITION.

An address pointer stores the address it 'points' to in a LO, HI format according to the formula:

$$\text{address} = \text{LO} + \text{HI} \times 256$$

where LO is the number (the LO byte) stored in the first of the memory cells making up the pointer and HI is the number (the HI byte) stored in the second cell. To determine the address pointer we need to know the addresses of the two cells of the pointer. Then we can PEEK at those cells to find LO and HI. 30884 30885 are the addresses of the START OF PROGRAM pointer. So, if you switch on your VZ and enter this command:

```
PRINT PEEK(30884) + PEEK(30885) x 256
```

you will discover the start of program address.

You got 31465? That is the usual start address of program memory, which follows on from the end of the communications region. When we enter basic program lines, whether we type them in using the keyboard, CLOAD them from tape or LOAD them from disk, the lines take up residence in RAM from this point on.

31465 is the start address for the line with the lowest line number in a BASIC program. Lines are stored in the sequence of their line numbers. Add a new line, and all the lines with higher line numbers will be pushed up in program memory to create a slot for the new line. Delete a line, and all the higher numbered lines will be pulled down to close the gap.

To explore the structure of program lines in memory, enter this two line program and RUN it:

```
10 FOR A=31465 TO A+37:PRINT PEEK(A);:NEXT
20 PRINT:LIST-20
```

The program has listed itself in two different ways. Line 20 performed the familiar BASIC listing of the program. Line 10 produced a PEEK listing of the program as a sequence of numbers stored in memory cells from 31465 onwards. How do the two compare? In the BASIC listing each program line has a number, a set of BASIC words (FOR, =, TO, PRINT, PEEK, etc.) and various other characters (digits, colons, brackets, A, etc.) which, together with the associated BASIC words, make up BASIC statements.

Now add these lines to the program:

```
30 CLS:FOR A=31465 TO A+37:X=PEEK(A)
40 IF X>31 AND X<95 THEN PRINT CHR$(X);ELSE PRINT X;
50 NEXT:PRINT:PRINT:GOTO 10
```

and RUN 30.

The top PEEK listing provides an ASCII decoding of those bytes that fall within the ASCII range. Variables, digits, brackets, colons, semicolons come through, and if we compare the two PEEK listings it is not hard to figure out where the line numbers occur. But where are the BASIC words? Their positions in the lines seem to be occupied by triple-digit numbers. Apparently more decoding is necessary. Add this line to the program and RUN 30 again:

```
5 REM A TEST OF THE ASCII PEEK LISTING
```

The text of the REM statement gets thorough ASCII decoding, letter by letter, blank by blank. Even the BASIC word PEEK is ASCII decoded, though it was not decoded in the previous RUN.

The text of a BASIC line always begins with the fifth byte for that line. The third and fourth bytes are the line number. Only two bytes (no more, no less) are used to record the line number, using the LO HI format, no matter how many digits the number may have:

$$\text{line number} = \text{byte 3} + \text{byte 4} \times 256$$

Because the line numbers in this program are small, byte 4 is '0', which is 0×256 . The first two bytes of a BASIC line record the start address of the next line in the program, again according to the LO HI formula:

$$\text{start address of line} = \text{byte 1} + \text{byte 2} \times 256$$

If we delete line 5 and RUN 30 we see that the first two bytes of line 10 are '4' and '123' which gives us the start address for line 20 as $4 + 123 \times 256 = 31492$.

PRINT PEEK(31492) gives us 15. If we locate '20' in the PEEK listing as the third byte of line 20, we see that byte 1 of line 20 (two locations back from the number 20) is '15'. But if '15' is the first byte of line 20 then the '0' to the left of '15' must be the last byte of line 10. Indeed, the last byte of every line is a '0', a marker indicating that this line ends here. There is even a pseudo-end of line marker in the memory cell preceding the first byte of the first line of a program -- a zero at address 31464, the last cell of the communications region. If this byte is POKE changed to a non-zero value, your program will not RUN.

Note that in the REM statement PEEK occupied four bytes of memory, one byte for each character:

```
80  69  69  75
P   E   E   K
```

In line 10, however, the position of the PEEK word is occupied by a triple-digit number -- 229.

UNDERSTANDING YOUR VZ . . . PART TWO.

This byte '229' is a token code for PEEK. Every BASIC word is represented in text of program lines (once they have passed into program memory) by a unique token code, a number between 128 and 250. A one byte token uses up three memory cells less than the four character word PEEK. And since BASIC words constitute a major part of most programs, tokenisation results in considerable saving of memory occupied by a program. In the text of a REM, PRINT or LPRINT statement, words like PEEK, FOR, GOSUB are not regarded as BASIC words; they are just strings of characters to be displayed and are not tokenised. Inverse and graphic characters are also coded using numbers greater than 127. They are distinguished from the token codes for BASIC words by their relation to quotes (""). Within quotes such numbers are interpreted as character codes; outside quotes they are token codes.

One peculiarity is the ELSE statement. ELSE is always preceded in memory by a colon (:), but the colon is never displayed when lines are LISTed or LLISTed. You have to count this invisible colon to get an accurate estimate of the length of a line involving ELSE statements.

Enter the TOKEN routine and RUN. The routine will display all the basic words that make up the VZ's BASIC VOCABULARY, along with their token codes. Press any character key to halt the display. Press again to continue. If you have a PP40 (or TP40) printer then Token will LPRINT a hard copy of all the words and their codes if you RUN 100.

T O K E N .

```

10 CLS:D=127:SOUND 0,1:FOR R=5712 TO 6175:B=PEEK(R)
20 A#=INKEY$:A#=INKEY$:IFA#<>" "THEN C=NOT C:SOUND 30,1
30 IF C=-1 THEN 20
40 IF B>128 THEN D=D+1:PRINT:PRINT D;
50 IF B>169 THEN PRINT CHR$(B-128);ELSE IF B=0 THEN 80
60 IF B>31 AND B< 96 THEN PRINT CHR$(B);
70 SOUND 0,1
80 NEXT:END
100 CLS:LPRINT CHR$(18)"S1,"CHR$(17):D=127:FOR R=5712 TO 6175
110 B=PEEK(R):IF B>169THEN D=D+1:PRINT:PRINTD,CHR$(B-128);:C=C+1
120 IF C=1 AND B>169 THEN LPRINT ELSE IF C>1 THEN LPRINT,:C=0
130 IF B>169 THEN LPRINT D;CHR$(B-128);ELSE IF B=129 THEN PRINT
135 PRINT D;:D=D+1
140 IF B>31 AND B<96 THEN PRINT CHR$(B);:LPRINT CHR$(B);
150 NEXT

```

PART 3 NEXT MONTH...


```

2150 IFH=4116,R#="IT IS NOT BIG ENOUGH":RETURN
2160 IFB=18ORB=7,GOSUB2470
2170 IFB=13,GOSUB2730
2180 IFB=19,GOSUB3070
2190 IFB=10,GOSUB2870
2200 IFB=16ORB=6,GOSUB2390
2210 RETURN
2220 IFB=76ORB=38,GOSUB1470
2230 IFH=2030,F(9)=0,R#="OK"
2240 IFH=6030,R#="OK":F(3)=0
2250 IFH=2444ORH=1970,R#="YOU ARE NOT STRONG ENOUGH"
2260 IFH=3756,R#="A PASSAGE!":E$(37)="EW"
2270 IFH=5960,GOSUB3260
2280 IFH=6970,R#="IT FALLS OFF ITS HINGES"
2290 IFH=4870,R#="IT IS LOCKED"
2300 RETURN
2310 IFB>6,R#="IT DOES NOT BURN"
2320 IFB=26,R#="YOU LIT THEM"
2330 IFH=3826,R#="NOT BRIGHT ENOUGH"
2340 IF(B=23ORH=6970)AND(C(26)<>0),R#="OF NBUDIFT":GOSUB4260
2350 IF(B=23AND(C(26)=0)),R#="A BRIGHT "+V$:F(50)=1
2360 IFH=6970AND(C(26)=0,F(43)=1),R#="IT HAS TURNED TO ASHES"
2370 RETURN
2380 IF(B=16ORB=6)AND(R=41ORR=51),R#="YOU CAPSIZED!":F(56)=1
2390 IFH=6516AND(C(16)=0),R#="IT IS NOW FULL":F(34)=1
2400 IFH=656,R#="IT LEAKS OUT!"
2410 RETURN
2420 IFB<>22,R#="YOU CAN'T DO THAT!":RETURN
2422 IF(C(22)<>0),R#="YOU DON'T HAVE THE "+T$:RETURN
2425 IFR<>15,R#="DOES NOT GROW":RETURN
2430 R#="OK":F(37)=1:C(B)=R
2440 RETURN
2450 IFB=23AND(F(37)=1AND(F(34)=1),R#="X2":F(38)=1:GOSUB4260
2460 RETURN
2470 IFB=7ORB=18,R#="THWACK!"
2480 IFH=5918,R#="YOU CLEARED THE WEBS":F(66)=1
2490 IFH=187,R#="THE DOOR BROKE!":E$(18)="NS":E$(28)="NS"
2500 IFH=717,R#="YOU BROKE THROUGH":E$(71)="H"
2510 RETURN
2520 IFB=16,B=22:GOSUB2450
2530 IFH=499,R#="WHERE?"
2540 RETURN
2550 IFH=4337,VB=2:GOSUB800:RETURN
2560 IFR=36,R#="YOU FOUND SOMETHING":F(13)=0
2570 RETURN
2580 IFR=76,VB=4:GOSUB800:RETURN
2590 IFR=75,VB=2:GOSUB800
2600 RETURN
2610 IF(B=3AND(F(29)=1),R#="TAKEN OFF":F(29)=0
2620 IF(B=20AND(F(51)=1),R#="OK":F(51)=0
2630 IFB=36ORB=50,GOSUB2950
2640 RETURN
2650 IFH=3859ORH=3339ORH=1241ORH=2241ORH=751,R#="WITH WHAT?"
2660 RETURN
2670 IFH=2340,R#="IT GOES ROUND"
2680 IFH=2445,R#="UIF HBUDFT POF0, UIF QPPM FNQUJFT":F(33)=1:SL=1
2685 IFSL=1,SL=0:GOSUB4260
2690 RETURN
2700 IFR=14ORR=51,R#="YOU HAVE DROWNED":F(56)=1
2710 RETURN
2720 R#="HOW":RETURN
2730 IFB=6ORB>6,RETURN
2740 C(B)=R:R#="DONE"

```

```

2750 IFH=4180RH=518,R#="YOU DROWNED!":F(56)=1
2760 IFB=8ANDF(30)=1,C(2)=R
2770 IFB=16ANDF(34)=1,R#="YOU LOST THE WATER!":F(34)=0
2780 IFB=2ANDF(30)=1,F(30)=0
2790 RETURN
2800 IFB=62ANDF(44)=0,R#="YOU DO NOT HAVE ANY"
2810 IFH=5762ANDC(1)=0ANDF(44)>0,GOSUB3230
2820 RETURN
2830 IFB=0ORB>6,RETURN
2840 R#="DID NOT GO FAR!":C(B)=R
2850 IFH=3317,R#="ZPV DBYH1U UIF CPBS":F(32)=1:GOSUB4260
2860 RETURN
2870 IFB=10,R#="B OJDF UYDF":GOSUB4260
2880 IFH=5233,R#="WHAT WITH?"
2890 IFB=83,R#="HOW, O MUSICAL ONE?"
2900 IFH=5510,F(35)=1:R#="X1#+ " IS FREE!":E#(56)="NS"
2910 RETURN
2920 IFB=0ORB>6,RETURN
2930 IFB=50RB=24,R#="YUM YUM!":C(B)=81
2940 RETURN
2950 IFR=4ANDB=50,F(45)=1:R#="YOU REVEALED A STEEP PASSAGE"
2960 IFR=3ANDB=50,R#="YOU CANNOT MOVE RUBBLE FROM HERE"
2970 IFH=7136,R#="THEY ARE WEDGED IN!"
2980 RETURN
2990 IF(B=67ORB=68)ANDC(9)=0ANDR=49,R#="OK":F(47)=1
3000 RETURN
3010 IFR<>27ORB<>63,RETURN
3020 PRINT:PRINT"HOW MANY TIMES":INPUTR:IFMR=0,PRINT"A NUMBER":GOTO3020
3030 IFMR=F(42),R#="A ROCK DOOR OPENS":E#(27)="EW":RETURN
3040 R#="ZPV 1BWF NJTUSFBUFE UIF CFNM!":F(56)=1:GOSUB4260:RETURN
3050 IFH=5361,H=5318:GOSUB2470
3060 RETURN
3070 IF(H=4864ORH=4819)ANDC(19)=0,R#="X6#":F(63)=1:GOSUB4260
3080 IFB=27,GOSUB1290
3090 RETURN
3100 IFH=7549ORH=7649,R#="WHAT WITH?"
3110 IFB=10RB=62,GOSUB1750
3120 RETURN
3130 IFH=4870ANDC(21)=0,R#="THE KEY TURNS!":F(70)=1
3140 RETURN
3150 IFH=1870,R#="HOW?"
3160 RETURN
3170 IFR=48,R#="HOW?"
3180 RETURN
3190 R#="ARE YOU THIRSTY?"
3200 RETURN
3210 R#="HE TAKES IT AND SAYS "+STR#(F(42))+ " RINGS ARE ":SL=1
3215 IFSL=1,SL=0:R#="NEEDED":C(25)=81
3220 RETURN
3230 F(44)=F(44)-1:R#="A NUMBER APPEARS - "+STR#(F(41))
3240 IF(F(44)=0,C(1)=81
3250 RETURN
3260 PRINT:R#="X1BU JT UIF DPEF":GOSUB4260:PRINTR#:INPUTC
3270 R#="WRONG!":IFCN=F(41),R#="IT OPENS":F(21)=0
3280 RETURN
3290 T=R:R=F(F(52)+57):GOSUB3310:R=T
3300 R#="X4#+RIGHT#(D#,LEND4)-2":RETURN
3310 RESTORE:FORI=1TOR:READD#:NEXTI
3320 RETURN
3330 RESTORE:FORI=1TOS0:READD#:NEXTI
3340 RETURN

```

BASIC MONITOR FOR THE UZ200 by R.G.DAVIS

A very handy little program if you just want to experiment with machine code programming.

Enter the program and run, you will be given three choices ..

ONE..MEMOD You are asked for start and end address you then step through memory one byte at a time and are given the chance to enter new data in hex, press <RETURN> and step on to the next byte.

TWO..REVIEW displays on the screen 16 lines by 8 columns of hex data. Press <SPACE> to bring up the next 16 lines of data.

THREE..PRINTOUT use your printer to get a hex dump of your program. If you wish to have 16 bytes per line then change the end of line 9 to read..

FOR N=1 TO 16

MEMOD BY R.G.DAVIS 8 AVOCARDO ST MILDURA

```

1 CLS: CLEAR 500
2 PRINT "MEMOD      M"
3 PRINT "REVIEW    R"
4 PRINT "PRINTOUT  P"
5 INPUT Y$
6 IF Y$="M" THEN 12
7 IF Y$="R" THEN 17
8 GOSUB 30
9 GOSUB 61: LPRINT H$ " " B " "; FOR N=1 TO 8
10 GOSUB 36: LPRINT H$ " ";
11 GOSUB 38: NEXT: LPRINT: GOTO 9
12 GOSUB 30: CLS
13 GOSUB 61: PRINT H$ " " : GOSUB 36
14 PRINT H$ " " : INPUT B$
15 IF B$="" THEN 16 ELSE GOSUB 42: POKE M, F
16 GOSUB 38: GOTO 13
17 GOSUB 30: CLS
18 PRINT: FOR T=1 TO 16
19 PRINT: GOSUB 61: PRINT H$ " " : FOR N=1 TO 8
20 GOSUB 36: PRINT H$ " ";
21 GOSUB 38: NEXT: NEXT
22 Z$=INKEY$: W$=INKEY$: IF W$<>" " THEN 22
23 GOTO 18

```

```

30 INPUT "START ADDRESS"; M: B=M
31 IF M<32768 THEN 33 ELSE Q=M
32 GOSUB 40: M=X
33 INPUT "END ADDRESS"; E
34 IF E<32768 THEN RETURN ELSE Q=E
35 GOSUB 40: E=X: RETURN
36 D=PEEK(M): DE=D: IF D<32 THEN DE=46
37 H$="": GOSUB 65: RETURN
38 M=M+1: IF M=32768 THEN GOSUB 41
39 IF M=E THEN END ELSE B=B+1: GOTO 61
40 X=-1*(65536-Q): RETURN
41 Q=M: GOSUB 40: M=X: RETURN
42 C$=B$: F=0: GOSUB 48
43 RETURN
44 IF C<58 THEN C=C-48
45 IF C>63 THEN C=C-55
46 RETURN
48 C=ASC(C$)
49 GOSUB 44: G=16*C: F=F+G
50 C$=RIGHT$(B$, 1): C=ASC(C$)
51 GOSUB 44: F=F+C: RETURN
61 H$="": A=B/4096: A=INT(A): GOSUB 67
62 D=B-4096*A
63 A=D/256: A=INT(A): GOSUB 67
64 D=D-256*A
65 A=D/16: A=INT(A): GOSUB 67
66 A=D-16*A
67 AB=A: IF A>9 THEN A=A+55 ELSE A=A+48
68 A$=CHR$(A): H$=H$+A$: A=AB: RETURN

```

SAMPLE OF PRINTOUT

```

8FF1 36849 21 00 70 11 01 70 01 FF
8FF9 36857 01 36 AA ED B0 C9

```

OTHER VZ USER GROUPS & CLUBS.

AUSTRALIA.

AD LIB Vee Zed MICRO.
Mr. Gordon Brownell, 13 Brooks St., BIGGENDEN. QLD. 4621.

VZ USER.
Mr. Mark Harwood, P.O. Box 154, DURAL. NSW. 2158.

VZ DOWN UNDER.
5 Cameron Court, WANTIRNA. VIC. 3152.

HUNTER VALLEY VZ USERS GROUP.
C/O P.O. Box 161, Jesmond. JESMOND. NSW. 2299.

MAVZ.
Mr. Graeme Bynwater, P.O. Box 388, MORLEY. WA. 6062.

LE VZ OOP

Mr John D'ALTON 39 AGNES St
TOOWONG 4066 QLD.

VZ 200-300 INPUTS, OUTPUTS AND MODIFICATIONS.

The drawings on the next page show the INPUTS and OUTPUTS of the VZ as well as how to make a new DATASSETTE LEAD, install RESET and SHIFT LOCK SWITCHES. None are too difficult to implement and with a little care even the novice should be able to do the mods. Remember, any mods done on a VZ will void your warranty.

VZ 200/300 INPUTS/OUTPUTS.

The original drawing appeared in April issue of VE ZEE NEWS and I thank David Boyce for his permission to reprint it. I added a few more details to the drawing. Anyone contemplating MODS to their VZ will find the information most useful as some published printouts in other magazines were incorrect. All the INPUT/OUTPUT PORTS are self explanatory and need no further comment.

TAPE PLAY/RECORD LEAD.

I'm into my 4th. lead now. The supplied leads just don't stand up to well. Also I believe the VZ 300 leads are way to short. The circuit should help anyone making or repairing their lead.

RESET.

The RESET circuit using SW3 (switch 3) and/or SW4 connects to the MEMORY EXPANSION PORT pins 1 and 2. The 220 ohm resistor is there to prevent possible damage to internal circuitry. Besides the 220 ohm resistor only SW3 is needed for reset. For Persons who accidentally keep bumping the reset button then SW4 can be added in series with SW3, then both switches have to be pressed at the same time to affect reset. Connect only A or B to pin 2. Use N.O. (normally open) push button switches for SW3 and SW4.

POWER OFF RESET.

Members with DISK DRIVES should not use previous RESET as it does not totally reset the VZ. After using previous RESET with a M.L. (machine language) program I could not access the DISK DRIVE till I used the ON/OFF switch. The POWER OFF RESET switch uses a N.C. (normally closed) push button switch and interrupts the power supply to the VZ. It is the preferable of the two and is easier to install.

SHIFT LOCK SWITCH.

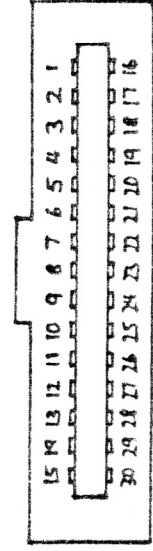
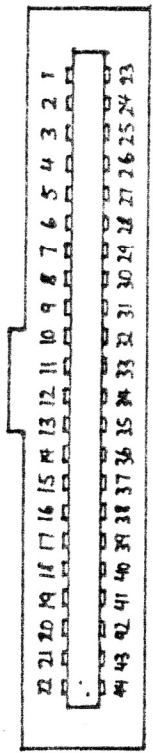
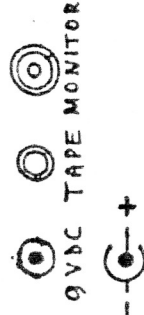
This is a little more difficult to implement as the PCB (printed circuit board) has to be removed from the case and a couple wires soldered to the bottom of the PCB directly under where the KEYBOARD cable is. The RED LED is the POWER ON LED and is shown for clarity. SW5 Should be a DPDT (double pole double throw) switch. You can use a slide, push button or toggle switch depending on Your preference. Only half of SW5 is needed to implement a SHIFT LOCK. A warning LED can be connected to the unused half of SW5 giving visual indication that SHIFT LOCK is operational. The 330R resistor is there to limit the current to the LED. If a FLASHING LED is desired then the resistor is omitted as it works directly off 5 VOLTS. 5 VOLTS should be available either side of the KEYBOARD CABLE on the top side of the PCB. VZ 300 owners should check for themselves as I'm not familiar with it. Do not connect the SHIFT LOCK wires to A 2 and D 2 on the MEM. EXP. or PRINTER PORTS as they are not part of the keyboard matrix. I did not give any construction details as the mods are pretty simple to effect. The Switches and LED can be mounted anywhere it's convenient providing there is sufficient room and doesn't foul any internal pillars.

Happy Hacking

JOE LEON

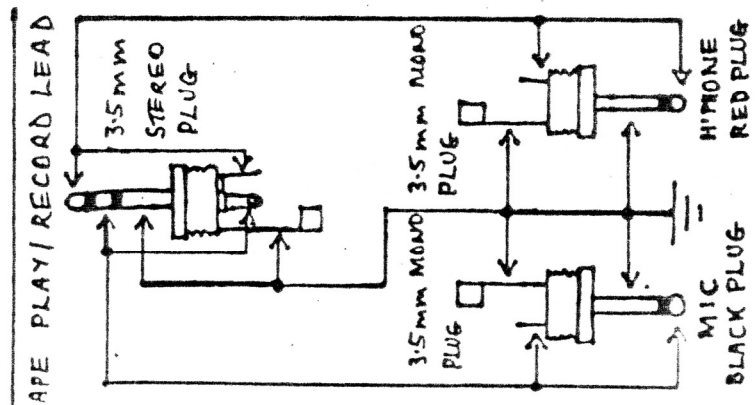
(Technical Advisor).

VZ 200/300 INPUT/OUTPUT CONNECTIONS - REAR VIEW - NOT TO SCALE - ORIGINAL BY D. BOYCE



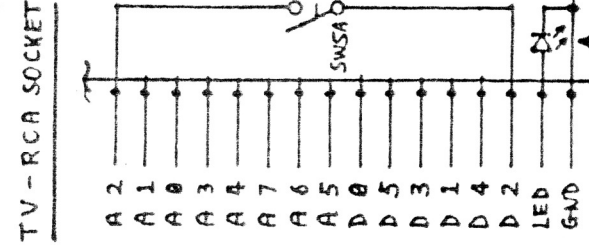
MEMORY EXPANSION - DISK CONTROLLER, ETC. PRINTER - JOYSTICK INTERFACES, ETC.

VDC - 2.1mm POWER SOCKET
APE - 3.5mm STEREO SOCKET
MONITOR - RCA SOCKET

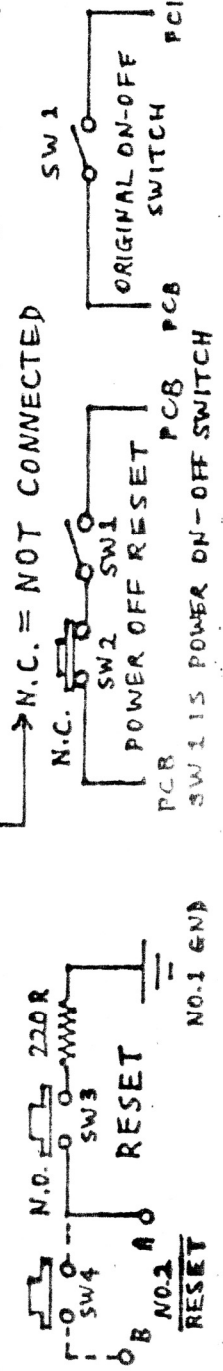


NO. FUNCTION	NO. FUNCTION	NO. FUNCTION
1 - GND	16 - MI	31 - D 3
2 - RESET	17 - WAIT	32 - D 5
3 - A 10	18 - NMI	33 - D 6
4 - A 9	19 - RD	34 - N.C.
5 - A 8	20 - IORQ	35 - A 0
6 - A 7	21 - +5VDC	36 - D 0
7 - A 6	22 - GND	37 - D 1
8 - A 5	23 - GND	38 - INT
9 - A 4	24 - A 11	39 - HALT
10 - A 3	25 - A 12	40 - MERQ
11 - A 2	26 - A 13	41 - WR
12 - A 1	27 - A 14	42 - N.C.
13 - D 2	28 - A 15	43 - +9VDC
14 - D 7	29 - CLK	44 - N.C.
15 - RFSH	30 - D 4	

NO. FUNCTION	NO. FUNCTION
1 - N.C.	16 - N.C.
2 - N.C.	17 - N.C.
3 - N.C.	18 - N.C.
4 - +5VDC	19 - +5VDC
5 - IORQ	20 - N.C.
6 - D 3	21 - D 4
7 - D 6	22 - D 5
8 - D 2	23 - D 7
9 - D 0	24 - D 1
10 - A 5	25 - RD
11 - A 2	26 - A 1
12 - A 6	27 - A 4
13 - A 3	28 - A 0
14 - WR	29 - A 7
15 - GND	30 - GND



RED POWER ON LED
BOTTOM VIEW OF PCB WHERE
KEYBOARD CABLE CONNECTS
SHIFT LOCK SWITCH
WITH OPTIONAL SHIFT
LOCK ON INDICATOR LED



BY JOE LEON

JUNE 1986


```

50 A=ASC(A*):B=A:IFP=-1ANDA>3]ANDA<64THE
NB=B+132
60 IFP=-1ANDA>63ANDA<128THENB=B+128
65 IFK=-1ANDA>63ANDA<95THENA=A+32
70 IFA>127THENGOSUB110:GOTO90
80 LPRINTCHR*(A);
90 IFP=-1ANDA<127ANDA>3]THENLPRINTCHR*(B
);CHR*(95);
95 IFB=13THENPRINT" ";CHR*(8);:D=-1
100 PRINTCHR*(B);:D=D+1:IFD=35THENSOUND3
1,2;20,1
102 IFD=4]THEND=1
105 RETURN
110 IFA=133ORA=138THENLPRINTCHR*(85);CHR
*(8);CHR*(84);:RETURN
120 IFA=131ORA=140THENLPRINTCHR*(85);CHR
*(8);CHR*(69);:RETURN
130 IFA=137THENGOSUB190:LPRINTCHR*(92);:
RETURN
140 IFA=134THENGOSUB190:LPRINTCHR*(47);:
RETURN
150 IFA=143THENLPRINTCHR*(79);CHR*(8);CH
R*(85);:RETURN
160 IFA=128THENGOSUB190:LPRINTCHR*(42);C
HR*(8);CHR*(35);:RETURN
165 IFA=135ORA>138THENGOSUB190:LPRINTCHR
*(43);:RETURN
170 LPRINTCHR*(127);:RETURN
190 LPRINTCHR*(79);CHR*(8);CHR*(85);CHR*
(8);:RETURN
200 SOUND10,1:PRINT" ";CHR*(8);CHR*(8);:
LPRINTCHR*(8);
210 D=D-1:RETURN
300 FORT=28672TO29183:A=PEEK(T)
310 IFA<32THENLPRINTCHR*(A+64);ELSEIFA<6
4THENLPRINTCHR*(A);
320 IFA>63ANDA<96THENLPRINTCHR*(A);CHR*(
8);CHR*(95);
330 IFA>95ANDA<128THENLPRINTCHR*(A-64);C
HR*(8);CHR*(95);
340 IFA>127THENGOSUB370
350 D=D+1:IFD=32THEND=0:LPRINTCHR*(13);
360 NEXT:D=0:LPRINTCHR*(13);:LPRINT:RETU
RN
370 IFA>143THENA=A-16:GOTO370
380 GOSUB110:RETURN

```

FOR SALE - DATABASE - DISK / TAPE

DATA - 16k - VZ DATABASE. Enter data into records thirty characters long (accepts graphic characters). Runs on VZ 200+16k or VZ 300. Available on disk as DISK DATABASE or on tape as CASSETTE DATABASE.

Facilities include data entry into record of choice, into last record chosen, next record, auto-next for fast data entry, edit keys so you don't have to re-enter entire content of a record, delete a record, delete a block of records, gap delete, insert, gap insert, fast alphabetical sort of records--start anywhere in records; number sort; swap any two records; page display--ten records per page; display current page, next page, previous page, flip backward and forward through datafile, swap any two pages, fast search of entire datafile for a sequence of characters--anywhere in records, hardcopy your records--especially suited for VZ printer plotter; menu etc.

Disk DATA has Directory and ERASE commands, saves a datafile or any part thereof as a single binary file which loads back quickly. Cassette DATA CSAVES a datafile as a single T file--no slow loading of multitudes of D files! All instructions for using DATA are stored on disk and tape as datafiles--run DATA, load an instruction file and page through it. This program certainly stands out amongst the crowd of other such programs of it's type.

PRICE - \$20.00 for DISK or CASSETTE DATABASE - Please make all Cheques and Money Orders payable to and is available from :-
SCOTT LE BRUN 5 CAMERON COURT WANTIRNA VIC. 3152

* * FOR SALE * * * * FOR SALE * *
E&F W.P. PATCH3.1 - QUICKWRITE W.I.

PATCH3.1 - COPYRIGHT - H.V.VZ.U.G.

This single Patch will convert your E & F TAPE WORD PROCESSOR for full DISK use while retaining all TAPE functions. It can be used with 1 or 2 DRIVES. Below are the two Menus.

E)DIT TEXT	L)OAD
C)LEAR TEXT	S)AVE
P)RINT TEXT	D)IR
L)OAD FILE	E)RA
S)AVE FILE	R)EN
V)ERIFY FILE	I)NIT
Q)UIT PROGRAM	1-2) DRIVE 1
D)ISK	M)ENU

Fast SAVING and LOADING of TEXT DATA to and from Disk is provided using Block SAVE or LOAD.

Full instructions are supplied together with a Tape to Disk transfer utility for your E & F Tape Word Processor.

This Patch will work with V1.0 or V1.2 Disk Controller. A STATUS facility has been added for V1.0 DOS owners.

SYSTEM REQUIREMENTS :-
DISK DRIVE + V1.0 OR V1.2 DOS
VZ300 + 16K RAM PACK OR
VZ200 + 18K (16K RAM PACK + 2K)

The price - \$10.00, NZ AU\$12.00 and is available from :-

HUNTER VALLEY VZ USERS' GROUP
P.O. BOX 161 JESMOND 2299
N.S.W. AUSTRALIA Phone (049)51 2756

* * * NEW NEW NEW * * *

QUICKWRITE WORDPROCESSOR

DISC BASED WORDPROCESSOR
A\$40.00

QUICKWRITE WORDPROCESSOR IS SUITABLE FOR THE EXPANDED VZ200 AND VZ300 COMPUTERS.

QUICKWRITE is software on disc, so RAM and ROM PACKS do not have to be plugged and unplugged into the VZ which can cause loose port socket connections.

QUICKWRITE runs on either the LASER or VZ DOS disc controller.

QUICKWRITE saves and loads document text (data) to disc.

FEATURES.

- * Fast disc saving and loading of document text (data).
- * Automatic periodic saving of data while in typing mode if required.
- * Tape saving and loading of data as a backup medium.
- * Loading of E&F tape files (data) possible.
- * Printer font changes within the data.
- * Capitals/lower case software lock on/off.
- * Accommodates wide printers - up to 255 columns.
- * A Printer/Plotter can also be used.
- * Four print justify/wragged modes.
- * Adequate operator warnings.
- * Labelling of discs allowable, such as date, code etc.
- * The usual editing facilities:-
Delete, Insert, Find and Replace, Paste, Cut etc.
- * Number 1 or number 2 disc drive selection allowed.
- * The price of A\$40.00. includes surface postage within Australia.

Sold ONLY by VSOFTWAREZ
39 Agnes st., TOOWONG Q/LAND. 4066.
AUSTRALIA.
(07)371 3707.

P.O BOX 1972 C.P.O AUCKLAND N.Z

(C) Auckland VZ 300/200 Computer user's club 1986

All material in this newsletter is subject to copyright ..

Contributed programs & articles can be reproduced on the understanding that they are for the private use of subscriber's only.

Copyright is retained by the author.

This club will not be held responsible for the breeching of copyright caused by the publication of any material sent in under a false name.

Phone 673-967 before 9 pm please