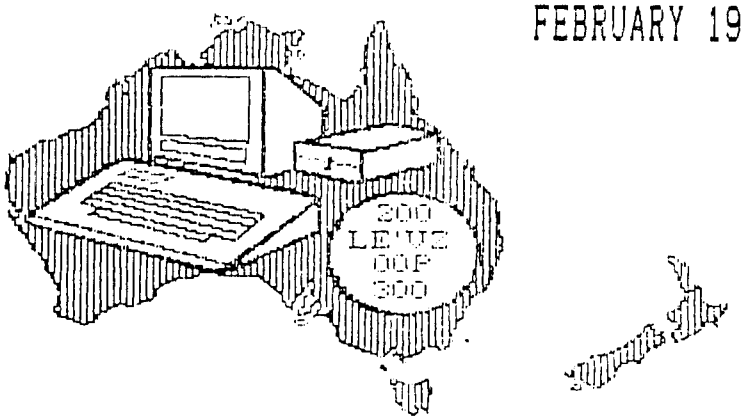
* LE'V2 200/300 *

Ouner's Operators Programmers

** THE SOUTH PACIFIC MAGAZINE FOR VZ COLOUR COMPUTERS **

FEBRUARY 1989. #22 A$2.00.



EDITORIAL

Hullo VZers, and others.

Firstly a piece of very exciting news that I should have mentioned in my editorial of LE'VZ #21. It is personal news tut we are sure that many readers will like to share it with us. Marie and I are now Grandparents, for the first time. Joshua Timothy Nicholas, a healthy son for our son John and his wife Lyn, born on the 19th of July 1988.
At present he is in India with his parents, as they go there each year for a few months on mission work.
I thank all those OOPs and others who sent Christmas cards, more than other years. GOD bless you and yours.

WORLD EXPO 88 is now a memory, but what a wonderful memory. Brisbane folk should be back to normal now. Much of the exterior works are still in place, but when the holidays are over I should imagine that will change.

The Computer Exhibition was held in the new Royal National Associations' building during November, but as with the last two or three years has developed into a big business sales expo. One finds no small hobby computers at these shows. D.S.E. and Tandy didn't even bother to display this year. Its OK if one wants to spend a couple of thousand dollars on equipment and readymade software, but not for me. If one wants to get into the basics, M/L etc, it seems to be a thing of the past. We (Australia) are following the trail of what I mentioned in LE'VZ #21 about the USA.

Now Mr Mark Harwood is also departing the VZ world. He is going to print only one more ***VZ USER.*** He hopes that some one will take it over and print it, so if anyone is interested, please contact him at his address on page 16. I will continue to publish ***LE'VZ 200/300 OOP*** as long as there are enough readers to make it worthwhile. It is up to *you* to contribute so that I can continue.

As from November 1988 as I printed in LE'VZ #21, I have discontinued to sell much of our software; only a selected group of good quality software, see page 14. I will continue to sell books, see page 16.

Mr Joe Leon must have had GOD by his side during and after a bad car crash he had. Joe is OK now. He suffered some cuts and bruises, but his car was a "write off." Joe says that life is really quite precious, one can always obtain another car but not another life.

For newcomers to VZ world, I can assure you that the little VZ is alive and well. There is so much happening, enough for another five years perhaps. A life span of ten years would be wonderfull for a machine in this ever changing world of high technology.

The ***H.V.VZ.U.G*** editor, Joe Leon has many modifications information available for those who are interested.

I hope that some of you obtain the December 1988/January 1989 bumper 100th issue of ***Australian Personal Computer.*** It comes in two parts. The smaller part mentions very briefly what was in each issue. It is quite interesting for newcomers to computing (and others) how the various machines have evolved. The VZ is not mentioned but the VIC 20 is. Then again the SC/MP, 2650,6800 D2 units aren't either.

That's all for folks, I wish you all a safe and blessed 1989.

John D'Alton.

CONTENTS

LE'VZ 200/300 OOP IS PUBLISHED BY MR J.C.E. D'ALTON OF 39 AGNES ST, TOOWONG, QLD. AUSTRALIA. TELEPHONE (07) 371 3707.

OUR QUICKWRITE WORD PROCESSOR-TEXT EDITOR IS USED EXTENSIVELY.

TARGET

This little BASIC game was

contributed by Mr Darryl G Lynch
of 17 Hopetoun St., DUBBO. 2830.

```
10 CLS:MODE(1)
20 COLOR2,1
30 P=0:Y=1:Z=2:W=3:X=4:H=0:HH=0
40 A=63:B=62:C=61:D=64:E=65
50 SOUND6,4;5,4;4,4;3,4;2,4;1,4
60 REM DRAW SHIP
70 FORF=59TO63
80 SET(A,F)
90 NEXTF
100 SET(B,61):SET(B,62)
110 SET(D,61):SET(D,62)
120 SET(C,62):SET(C,63)
130 SET(E,62):SET(E,63)
140 GOSUB600
150 I$=INKEY$:I$=INKEY$
160 IFI$=" "THEN190
170 IFI$=","THEN270
180 IFI$="M"THEN320ELSE650
190 REM FIRE
200 COLOR4:HH=HH+1
210 FORFF=58TO10STEP-4
220 SET(A,FF):SET(A,FF-1):SET(A,FF-2)
230 RESET(A,FF):RESET(A,FF-1):RESET(A,FF-2):SOUND1,1
240 IFA=Z AND FF=14THENGOTO460
250 NEXTFF
251 FORFF=58TO10STEP-8
252 RESET (A,FF)
260 GOTO140
270 REM MOVE RIGHT
290 SOUND1,1:GOSUB370:S=10:IFE+S>127THENS=-100
300 A=A+S:B=B+S:C=C+S:D=D+S:E=E+S
310 GOTO60
320 REM MOVE LEFT
330 SOUND1,1:GOSUB370
340 U=-10:IFA+U<0THENU=100
350 A=A+U:B=B+U:C=C+U:D=D+U:E=E+U
360 GOTO60
370 REM RESET
380 FOR F=59 TO63
390 RESET(A,F)
400 NEXTF
410 RESET(B,61):RESET(B,62)
420 RESET(D,61):RESET(D,62)
430 RESET(C,62):RESET(C,63)
440 RESET(E,62):RESET(E,63)
450 RETURN
460 REM HIT
470 H=H+1
480 SOUND9,1;8,1;7,1;6,1;5,1;4,1;3,1;2,1;1,1
490 IF H=3THENGOTO510
500 GOTO140
510 MODE(0):CLS
520 FORM=1TO10
530 CLS:PRINT@234," YOU WIN "
540 NEXTM
541 PRINT:PRINT:PRINT" IT ONLY TOOK YOU  ";HH;"
SHOTS "
550 END
600 REM TARGET
605 COLOR2
610 SET(P,15):SET(Y,15):SET(Z,15):SET(W,15):SET(X,15)
620 SET(Y,14):SET(Z,14):SET(W,14)
630 SET(Z,13)
640 RETURN
650 SOUND4,1:REM MOVES
660
RESET(P,15):RESET(Y,15):RESET(Z,15):RESET(W,15):RESET(-
X,15)
670 RESET(Y,14):RESET(Z,14):RESET(W,14)
680 RESET(Z,13)
690 T=RND(10)
700 IF X+T>127THENT=-100
710 P=P+T:Y=Y+T:Z=Z+T:W=W+T:X=X+T
720 GOTO140
1000END
```

* * WHAT'S IN THE OTHER MAGAZINES * *

I think this section will help VZ users and OOPs know what other information is available from other sources. This means in club magazines, newletter and journals as well as commercial publications.

Hunter Valley VZ User Group - Sept/Oct 1988.

1. The cover is printed in the new higher resolution.
2. Disk Menu - BASIC programme.
3. PUTGET - BASIC programme.
4. Printer Buffer by Mr Don McKenzie and Mr Dave Boyce.
5. 60*60 digit multiply routine - BASIC.
6. Beam Heading - BASIC programme for Hams.
7. 4K-64K RAM/EPROM by Mr Joe Leon.
8. Three pages of RAM communications addresses.

Australian Electronics OCT 1988 .

1. Monitor/Debugger by Mr Reg Batger.

VZ USER. July/Aug/Sept.

1. Disc Menu - BASIC programme.
2. VZ Prix - BASIC game.

Hunter Valley VZ User Group - Nov/Dec 1988.

1. ON GOTO programming.
2. Disk Drive problems.
3. Customising the E&F W.P.
4. Disc Utility.
5. Mortgage Repayment - BASIC programme.
6. 128K Sideways RAM by Joe LEON.
7. Disc head cleaning.

I.C PINOUT DRAWING

By JEREMY LEE

This little *BASIC* programme will enable an ***EPSON*** compatable printer to print I.C. (chip) with pinout connections.

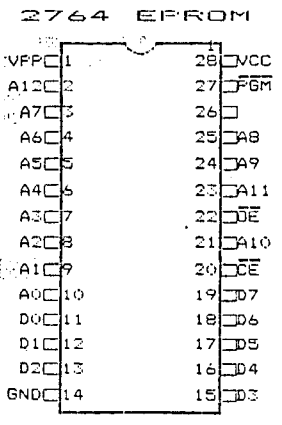
You may have to change line 305, ***CHR$(4)*** to another value,EI., ***CHR$(8)***. The number to suit ny ***CITIZEN 120-D*** printer is 8 (eight). This figure sets the space between the pins of the I.C.

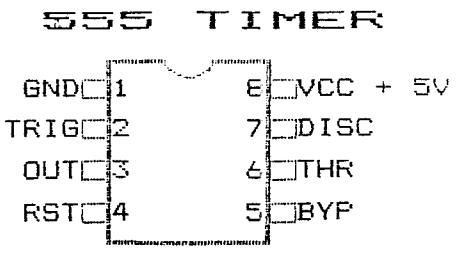
Also note that I have printed this by using our ***QUICKWRITE TEXT-EDITOR*** conversion utility to load the file into the ***QW TEXT-EDITOR***. In this way it enables me to reduce the width of any ***BASIC*** programme to the desired 85MM column width.

There will be NO typing errors as the file is converted from the ***BASIC*** programme.

This also means that the text or commands may be printed right at the left side.

```
0 CLEAR 5000
10 CLS
20 PRINT"        PIN-OUT PRINTER"
30 PRINT@64,"    HOW MANY PINS."
35 INPUTPN
36 IF(INT(PN/2)*2<>PN)OR PN<2 ORPN>70ELSE 40
37 PRINT@64,"  BE SERIOUS!       ":SOUND 10,8:GOTO 35
40 IFPN<20,S=1:GOTO 50
41 IFPN>28,S=2:GOTO 50
43 PRINT"  DOUBLE OR SINGLE WIDTH."
44 A$=INKEY$:IFA$<>"S"ANDA$<>"D",44
48 IFA$="D",S=2ELSES=1
50 L=INT(PN/2):DIMP$(PN):DIMB(PN):DIMPR$(L)
60 CLS
62 FORX=1TOPN
63 POKE 30777,1
64 PRINT@0,"                                "
65 PRINT@32,"  ENTER PIN ASSIGNMENTS.        ";
66 PRINT"    ANTHING UP TO 8 CHARACTERS   WITH AN '*'
AT THE END  TO
67 PRINT" SIGNIFY A BAR.                  ";
68 PRINT"--------------------------------":POKE
30777,35
70 PRINT@480,"PIN"X;:INPUTA$
75 IFRIGHT$(A$,1)="*",B(X)=1:A$=LEFT$(A$,LEN(A$)-1)
76 P$(X)=RIGHT$(A$,8)
80 NEXT
100 FORX=1TOL:A=LEN(P$(X)):P$(X)=LEFT$("
",8-A)+P$(X)
101 NEXT
110 CLS:PRINT" WHAT'S THE TITLE FOR THIS
MASTERPIECE
120 INPUTT$
125 Q=6+S-INT(LEN(T$)/2):IFQ<1,Q=1
130 T$=LEFT$("                          ",Q)+T$
134 MW$="  "
135 INPUT"HOW MANY SPACES OVER";MW:FORX=1TOMW:MW$=MW$+"
":NEXT
140 LPRINTMW$;CHR$(14);T$
200 REM"SETUP CHARACTER CODES
205 GE$=CHR$(27)+CHR$(75)+CHR$(8)+CHR$(0)
210 S$=GE$:FORX=1TO8:S$=S$+CHR$(0):NEXT
220
B$=CHR$(27)+"K"+CHR$(6)+CHR$(0):FORX=1TO6:B$=B$+CHR$(6)
:NEXT
230 P$=CHR$(255):FORX=1TO8:P$=P$+CHR$(129):NEXT
240
E1$=CHR$(27)+"K"+CHR$(2)+CHR$(0)+CHR$(255)+CHR$(255)
250
P2$="":FORX=1TO7:P2$=P2$+CHR$(129):NEXT:P2$=P2$+CHR$(25
5
260
N$=CHR$(224)+CHR$(24)+CHR$(8)+CHR$(4)+CHR$(2)+CHR$(2)
261 FORX=1TO4:N$=N$+CHR$(1):NEXT
265 N$=N$+CHR$(2)+CHR$(2)+CHR$(4)+CHR$(8)+CHR$(24)+CHR$
(224)
270 E$="":FORX=1TO8:E$=E$+CHR$(3):NEXT
275 T$="":FORX=1TO8:T$=T$+CHR$(192):NEXT
300 REM"PRINT FIGURE
305 LPRINTCHR$(27);"A";CHR$(4)
310 FORX=1TOL
315 AL=L+L-X+1
318 PR$=MW$
320 IFB(X)=0,PR$=PR$+"        ":GOTO340ELSEFORY=1TO LEN
(P$(X))
330 IFMID$(P$(X),Y,1)<>" ",PR$=PR$+B$ELSEPR$=PR$+" "
335  NEXT
340 IFS=1,W=2:W2=6ELSEW=4:W2=10
341 PR$=PR$+"
"+CHR$(27)+"K"+CHR$(2)+CHR$(0)+CHR$(0)+CHR$(0)
342  PR$=PR$+E1$:GOSUB 1000
350 IFX<>1,360
351 FORY=1TOW:PR$=PR$+GE$+T$:NEXT
352 PR$=PR$+CHR$(27)+"K"+CHR$(16)+CHR$(0)+N$
353 FORY=1TOW:PR$=PR$+GE$+T$:NEXT
360 IFX<>1,FORY=1TOW2:PR$=PR$+S$:NEXT
370 GOSUB1000:PR$=""
371 PR$=PR$+E1$
378 GOSUB 1000
379 PR$="
"+CHR$(27)+"K"+CHR$(2)+CHR$(0)+CHR$(0)+CHR$(0)
380 IFB(AL)=0,400ELSEFORY=1TOLEN(P$(AL))
390 IFMID$(P$(AL),Y,1)<>" ",PR$=PR$+B$ELSEPR$=PR$+" "
395  NEXT
400 GOSUB  1000
410 PR$=""
415 LPRINT
416 PR$=MW$
418 N1$=MID$(STR$(X)+"  ",2,2)
419 N2$=RIGHT$(STR$(AL),2)
420 PR$=PR$+P$(X)+GE$+P$+E1$+N1$
430 FORY=1TOW2-3:PR$=PR$+S$:NEXT
440 PR$=PR$+N2$+E1$+GE$+P2$+P$(AL)
450 GOSUB 1000
455 LPRINT
460 NEXT X
463 PR$="          "+MW$
464
PR$=PR$+CHR$(27)+"K"+CHR$(2)+CHR$(0)+CHR$(0)+CHR$(0)
465 PR$=PR$+E1$:
FORY=1TOW*2+2:PR$=PR$+GE$+E$:NEXT
466 PR$=PR$+E1$:GOSUB 1000
470 LPRINTCHR$(27);"A";CHR$(10)
480 LPRINT" "
500 RUN
1000 IFPR$="",RETURN
1001 FORQ=1TOLEN(PR$)
1005 D=ASC(MID$(PR$,Q,1))
1010 IFINP(0)<>254,1010
1020 OUT 13,D:OUT 14,D
1030 NEXT:PR$=""
1040 OUT 13,0:OUT 14,0
1050 RETURN
```





<<— THIS DRAWING HAS BEEN PHOTO-COPY REDUCED

STRUCTURED PROGRAMMING

Part 2 By Chris Hobrough.

In Part 1 we saw how a programme could be split up into a few major steps could be further split up so that we end up with a hierarchy of ever more detailed modules. Now we'll examine the flow of execution within some of the modules and take a look at the sort of formal command structures that are appearing in most modern versions of *BASIC*. Many of these have come to *BASIC* from more structured languages such as *Pascal* and though they aren't available on the VZ, there can be some advantages in knowing how to simulate them.

The flow of execution through a program can take three basic forms, namely sequence, conditional branching and loops. Each of these is fairly simple in itself but it is the way that they are used that often makes the difference between clear simple logic and a mess.

Conditional Branching

Most of the time program statements are executed one after another in a simple sequencial manner. There is a limit to the usefulness of this kind of flow, however, and sooner or later we may need to branch off according to the result of a decision. This will involve an IF statement or an implied one in a more complex command structure.

The VZ offers simple, one line IF .. THEN .. ELSE decisions. The trouble is that sometimes we can't fit all the things we want to do into a single line and so some versions of *BASIC* offer multi-line IF structures. They look like this:-

```
IF condition THEN
   .................      ' Statement block
ELSE
   .................      ' Statement block
END IF
```

The results of the search option in our video data base program will depend on whether the title requested is in the file. If it's found the search will stop but if it's not, the search will get to the last record, which will contain an End of File marker (EOF). So the index (I) into the array will either point to the required title or to the EOF marker and we can display the results based on a decision about what I is pointing to, using the structure above.

```
11000 ' Display Results ..................................
11002 '
11006                              ' If "EOF" reached
11008 '
11010 IF VT$(I,1) = "EOF" THEN GOTO 11100 ELSE GOTO 11200
   .
11100    PRINT etc                 ' Then
   .
11150 GOTO 11300
11200    PRINT etc                 ' Else
   .
   .                               ' End If
11300 GOSUB 12000                  ' Hold
11310 RETURN
```

You can see that we use a single line IF statement to branch to one of two blocks of code and a GOTO at the end of the first block to jump over the second one. This sort of structure may not seem all that significant but, if carefully followed, it can pay dividends when you get to complex, nested decision problems.

If the decision involves a choice of several options a CASE statement becomes the best way to achieve it. A CASE statement looks like this:-

```
SELECT CASE expression
   CASE item 1
   .............      ' Statement block
   CASE item 2
   .............      ' Statement block
   CASE ELSE
   .............      ' Statement block
END SELECT
```

There can be as many cases as needed and the CASE ELSE is optional. Processing our choice from the list of menu options is an ideal situation for a CASE structure.

```
4000 ' Process Choice ....................................
4002 '
4010                               ' Select case choice (CH$)
4020    IF CH$ = "F" GOTO 4100     '    Find
4030    IF CH$ = "A" GOTO 4200     '    Add
4040    IF CH$ = "D" GOTO 4300     '    Delete
4050    IF CH$ = "L" GOTO 4400     '    List
4060                               ' End select
4100 GOSUB 5000                    '    Find title
4110    GOTO 4500
4200 GOSUB 6000                    '    Add title
4210    GOTO 4500
4300 GOSUB 7000                    '    Delete title
4310    GOTO 4500
4400 GOSUB 8000                    '    List all titles
4500 RETURN
```

If you have Extended *BASIC* then the ON .. GOTO command can often be used to achieve the same results and sometimes a simple list of IF's is all that's necessary, provided they each fit on one line. Again, a little discipline pays off, especially when things get complicated.

Loops

When program execution branches backwards to a point already passed and then proceeds through the piece of code again you have a loop. This also involves a decision, or should do, and since the same decision will have to be made each time the loop is repeated, there must be something inside the loop which can change the status of the condition that is being tested.

Loop structures can be of two types: pre-test loops and post-test loops. The WHILE ... NEXT structure in our main control module is an example of a pre-test loop because the condition is tested before the body of the loop is executed each time. Rather than look at that again we'll have a look at the "Find Title" routine from the menu options.

We use a loop here so we can keep searching for different titles without going back to the main menu each time. But we also need an escape route, so we test for "ZZZZ" being entered as a signal that the user has finished with this option.

```
5000 ' Find Title ........................................
5002 '
5010 GOSUB 9000                    ' Get user's request
5020 IF TL$ = "ZZZZ" GOTO 5070     ' While request is not "ZZZZ"
5030    GOSUB 10000                '    Title search
5040    GOSUB 11000                '    Display result
5050    GOSUB 9000                 '    Get user's request
5060 GOTO 5020                     ' Next
5070 RETURN
```

We've ended up with a series of sub-routines again and, as before, we need to prime the structure by calling the routine which actually gets the user's request, before getting to the WHILE loop. The test is done with the IF statement which has its logic reversed compared to the WHILE statement. WHILE "ZZZZ" hasn't been entered we search for the request and display the result. Then we get a new request and GOTO the test again. If "ZZZZ" was entered we jump the loop.

The other type of loop structure is the post-test loop which, as its name suggests, has the condition tested after the body of the loop. The most common form is the REPEAT ... UNTIL structure. The "Hold" routine at the end of "Display Result" will illustrate a REPEAT loop. This routine simply puts a prompt at the bottom of the screen and holds the display until the user is ready to continue.

GOTO 6

```
12000 ' Hold .....................................
12002 '
12006 '                          ' Print user prompt
12008 '
12010 PRINT@ 483,"PRESS <RETURN> TO CONTINUE"; 12012 '
12020                            ' Repeat
12030     A$ = INKEY$            '     Get character
12032                            ' Until character is <RETURN>
12038 '
12040 IF A$ <> CHR$(13) GOTO 12020
12050 RETURN
```

The test is now after the body of the loop, which in this case is a simple INKEY, and the REPEAT line is just a REM for clarity. The main difference between these two forms of loop structure is that a post-test loop must always execute the body of the loop at least once in order to get to the test, whereas a pre-test loop need not go through the loop at all if the test fails the first time. This makes the pre-test form slightly more useful. There is only one loop structure on the VZ and that's the FOR .. TO ..NEXT structure which can be seen as a special case of a WHILE loop.

You may wonder why you need bother simulating structures that the VZ doesn't have. The fact is that you probably already do, perhaps without realizing it, and all that we're really doing is formalizing them and giving them names. Once the concepts are familiar to you they will become automatic. Whenever a certain type of situation arises, you'll recognize the need for one of these structures, rather than keep "re-inventing the wheel". In addition, when you see a WHILE, for instance, in a REMark, you immediately know what the next block of code is about.

In the next part of this article we'll look at the importance of proper documentation and presentation.

IN BRIEF

Secuity officials at a nuclear research centre in the USA want to talk to a hacker who has created a password for himself so he can re-enter their computer network at any time.

The firm has spent $US100, 000 in an unsucessful atemt to catch the culprit.

Three diferent viruses have infected companies across the UK.

IBM compatable machines are all affected. One virus is an encrypted type.

NSW PC manufactuer MICROBEE has advised the Australian Stock Exchange to suspend its shares while negotiations are proceeding in relation to a $2 million debt.

PRINTING YOUR OWN DESIGNED CHARACTERS:

QUICKWRITE WORDPROCESSOR AND SPECIALLY DESIGNED CHARACTERS Pt2.

By John D'ALTON.

Designing your own characters.

In the first part of this article which was published in LE'VZ E'VZ #21 November 1988, I showed how you can make your printer print characters that you have designed yourself. This is possible with an EPSON compatable printer.

IMPLEMENTING QUICKWRITE TO PRINT YOUR CHARACTERS.

You can print your designed characters within documents using our QUICKWRITE WORDPROCESSOR. This can be done with any version, V3, V4 or V4II.

Owners of QUICKWRITE should be familiar with its use but I will describe the method for folk who do not yet possess QUICKWRITE.

You should also be familiar with the operation of your printer, IE. the use of printer control codes, ***(P.C.C.)*** and the method of dumping specially designed characters that you have designed. In this regard I will refer only to the use of the ***CITIZEN 120-D*** printer.

QUICKWRITE V4.

[Unlike QUICKWRITE V3 the printer control (P.C.C.) are designed by the user and represented by characters. These are then called Printer Control Code Characters, (P.C.C.C.). When using QW V4 II, these are the punctuation marks on the keys, ! to @. That is the ! to @. They are inversed and are imbedded in the text of the document. This means that they are seen on the VDU but are not printed.

P.C.C.Cs can be typed in ***anywhere*** in the text, even part of a word. I now print "LE'VZ IS A GREAT MAGAZINE" where "LE'VZ", "G" and "M" are the special characters.

LE'VZ IS A GREAT MAGAZINE.

AX. The first thing to do is to copy the standard characters to RAM. Unlike QW V3, the P.C.C. of <27><58><0><0><0> now becomes a P.C.C.C represented by an inversed exclamation mark !

For this article I will change the opening square bracket "[" to a smaller than sign "<" otherwise the printer will not print the P.C.Cs, so you type in as "[".

BX. A carriage return is next, then the range of the new characters is set. For some reason I cannot represent these by a P.C.C.C but the same P.C.C as for QW V3. This is [27][38][0][34][59]

GOTO 7

PRINTING YOUR OWN DESIGNED CHARACTERS.

CX. Then the ASCII data for the new characters. I will only print the data for A, B and C. In part one of this article I missed printing the data for those characters. SORRY.

!<<27]<38]<0]<34]<59]<128]<14]<16]<46]<80]<168]<64]<46]
<16]<14]<0]<0]<128]<254]<0]<146]<0]<186]<0]<146]<0]<108
]<0]<0]<128]<124]<0]<130]<0]<130]<0]<198]<0]<198]<0]<0]

This data, *AX, BX and CX* is typed in at the beginning of the document files when you wish to print in the new character mode.

Now I will print the text including the P.C.C.Cs and the characters representing the new characters. Remember the P.C.C.Cs are all inversed.

#%-&$'#7;$ IS A #($REAT<% #.$AGAZINE.&

Starting from the first character.
P.C.C.C # swiches to print new chars, P.C.C.C % switches to double height and one line expanded, - and & are L and V, P.C.C.C $ switches to normal chars, ' is the apostrophe, P.C.C.C # switches back to new chars, 7 and ; is V and Z, P.C.C.C $ switches to normal chars, then IS A, P.C.C.C. # switches to new chars, (is G, P.C.C.C. $ switches to normal chars, then REAT, < is QW carriage return, P.C.C.C. % switches to new chars, . is M, P.C.C.C. $ switches to normal chars, then AGAZINE., and finally P.C.C.C. & switches expanded and double height off.
Hope you can follow that.

I wrote the new character letter on the top of the VZ200 rubber key. An overlay piece could be made also to make it a little easier, but of course one soon learns the "new" keys. Of course it is no good for touch typists.

I must point out to readers that this whole article could still be read and used to guide you to print new characters that you design using other computers and word processors, not just for the VZ and QUICKWRITE.

GREAT WORDS

And Moses with the elders of Israel commanded the people, saying, "Keep all the commandments which I command you this day."

NEW - NEW - NEW

QUICKWRITE II TEXT EDITOR

TO BE RELEASED SOON IS THIS NEW UNIT BASED ON THE QUICKWRITE WORDPROCESSOR.

For VZ200/300 users who have a 64K RAM pack or similar and disc drive system.
With space for a big file of approximately 40K, more than sufficient for most documents.

QUICKWRITE II is upwards compatable with the QUICKWRITE WORD PROCESSOR, both Disc and Tape files being accessable to QUICKWRITE II.
QUICKWRITE II has all the facilities of the QW WORDPROCESSOR plus an abundance of additional ones.

Such as:-

DISC MERGE • TEXT CENTERING FORMAT
USER DEFINED TAB FACILITY

SCREEN ECHO • MULTIPLE COPIES

DOUBLE SPACING • LOADING OF ED-AS FILES

LOADING OF E&F WORD PROCESSOR FILES

and many more!!

A separate utility is also provided which saves a *BASIC* programme onto disk as a QUICKWRITE II file.
QUICKWRITE II TEXT EDITOR is not a patched unit, but a professionally written unit.

Price of this excellent unit has not been set as yet but will be similar to our QUICKWRITE WORDPROCESSOR.

COME AND SEE IT DEMONSTRATED AT THE BVZUW CHRISTMAS EXPO.

A VZ300 BASED SYSTEM FOR RECORDING RUNNING WHEEL ACTIVITY BY RATS

This contribution is another first class one which deals with a very useful service that the great little VZ is beng put to, in this case into research of the behaviour of rats.

I thank DR.L.C Ward of the Department of Biochemistry at the University of Queensland for the article of which I will publish the remainder, Part two, in LE'VZ #23.

For those who would like more information, please contact the writers direct.

H. R. Johnson, L.C. Ward and L. C. Jones

University of Queensland, St Lucia,
Brisbane, AUSTRALIA

INTRODUCTION

The recording of the physical activity of experimental animals is frequently required in behavioural, physiological or biochemical studies. In some cases, for example, forced running on a treadmill the precise exercise regime is known and determined by the experimenter. In many studies however, the activity pattern and intensity is determined by the experimental animal but is constrained by the study design into a readily measurable and quantifiable form such as wheel-running. This allows measurement of frequency of activity with time, duration of activity bouts, intensity of activity (revolutions per minute) and magnitude of the work performed (distance covered against a known torque).

A number of methods have been used to obtain these data. Firstly, continuous recording using multi-channel pen-event recorders; secondly, data logging devices and electro-mechanical revolution counters or thirdly, graphical recording and manual-transposition of the data to computer files for further analyses. These methods whilst effective and reliable, not only require major captial investment in recorders, and/or technical assistance but are inflexible and do not readily allow manipulation of the recorded events without time-consuming transposition of the data. As part of a research programme investigating scheduled-feeding-induced wheel running activity in the rat, we have developed an inexpensive microcomputer-based data acquisition system. The necessary hardware is either commercially available or is simple and inexpensive to manufacture. Software has been developed which records data in a form suitable for further computational analysis.

METHODS

Cages and running wheels

The animal cage was rectangular (200 x 140 x 140 mm), constructed of steel mesh and fitted with a food hopper and water bottle with nipple feeder. The cage was attached to a back plate which supported a Wahmann-style running wheel. Access to the wheel from the cage was *via* a 60 mm diameter hole which could be closed by a shutter. The wheels were 340 mm in diameter and fitted with 90 mm long stainless steel spokes at 18 mm intervals around the wheel rim. The wheels were fitted with friction clamps on the axles to allow adjustment in the torque required to turn the wheel. Two diametrically opposed magnets were mounted in the wheel hub and activated a reed switch upon wheel rotation. A twin wire cable from the reed switch connected the wheel to the computer interface unit.

DATA COLLECTION

Instrumentation - Equipment (Fig. 1)

The equipment comprises:

(i) a VZ300 (Laser 310 in U.S.) personal computer with video monitor or television and tape cassette

(ii) 16K RAM expansion module interface

GOTO 9

(iii) printer interface

iv) data collection module and

(v) printer, Epson LX 80 or similar.

Instrumentation - Design

Equipment was designed for the continuous simultaneous monitoring and recording of running wheel activity of 8 animals housed in individual cages/wheels. A block diagram of the system is presented in Fig. 2. The system comprises three main elements: data collection module, computer, cassette data recorder. The data collection module consists of 2, 4-bit R-S latches (CD 4044) and 1 each 8-bit function latch (74LS374), 8-bit status gate (74LS244), a clock circuit and a parallel to serial convertor with associated electronics for recording data to the cassette at a rate of 2200 baud. Data from the collection module is passed via an 8-channel data bus through the printer interface to the computer. The printer interface is adapted to decode all external addresses. Overall system control is by the VZ 300 computer. Each cage pulse sets an activity latch which is strobed 50 times per second when the clock strobe line goes high. When the clock strobe line goes low, only those activity latches which were set when read are reset, effectively eliminating switch bounce and pulse race problems. The 16K RAM module is used for data recovery only.

Instrumentation - Software

Programs have been written in assembly language for the Z80 processor to perform the functions indicated in Fig. 2. Assembly language was used owing to the speed requirements for data processing and equipment control. The data collection program is loaded *via* cassette tape and occupies 2K memory. Data for each of the eight channels are accumulated in binary code in a data queue for each minute of channel monitoring. Each hour, leader information is added to the queue, the data converted from 16-bit binary to ASCII, a trailer inserted and the queue output to tape. Channel monitoring continues during tape output in an alternate queue, ensuring no loss of data. Data dumping takes 11 seconds, providing storage for approximately 7 days data per side of a standard C60 cassette.

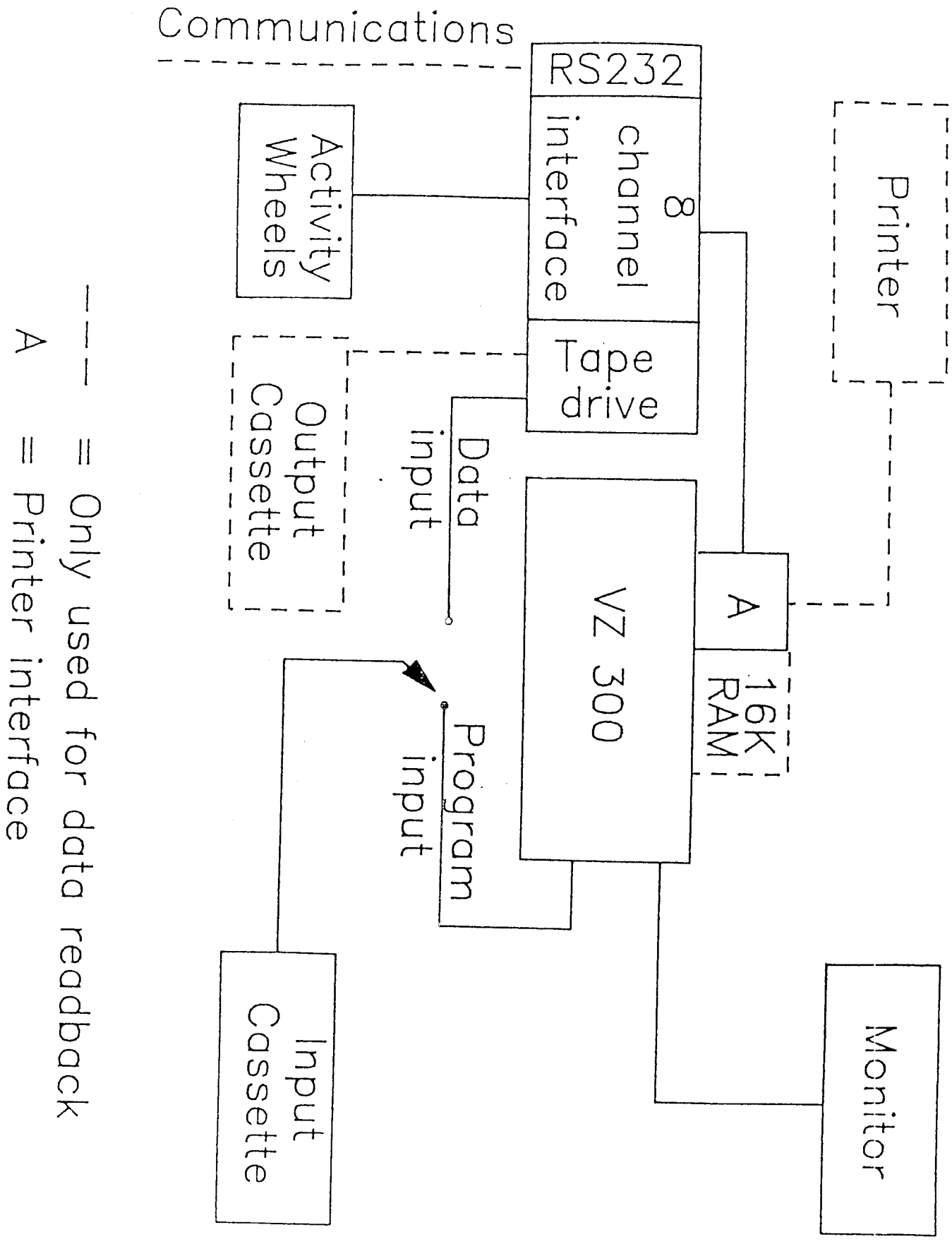
Two forms of data recovery are available. Data tapes are read using software which recovers data as recorded with 1 minute resolution or provides condensed data, for example accumulated hour totals. The data accumulation period may be set by the operator. All programs provide 24-hour totals. Data may be recovered for a complete collection period or for selected hours (raw data tape) or days (condensed data tape). Facility is provided for output to the printer or the RS232C communications line or for recording as condensed data on a second tape. Subsequent data analysis is possible by any computer program cable of inputting ASCII text files.

GOTO 10

JOHN D'ALTON VSOFTWARE2.
39 AGNES ST. TOOWONG. QUEENSLAND.
AUSTRALIA.
'PHONE (07) 371 3707
FEBRUARY 1989.

LE'VZ 200/300 OOP IS PUBLISHED APPROXIMATELY EVERY THREE MONTHS.

Figure 1 Flow diagram of components for microcomputer-based monitoring of running-wheel activity.



LE'VZ FORMATS.

To help me time-wise to make LE'VZ a better magazine, and yourself to get the most out of it, please read this page.

ALL PRICES are in Australian Dollars.

CURRENT ISSUE price is A$2.00 which includes surface/air postage within Australia and Air Mail to New Zealand. If you require more than one copy at one time, extra money must be sent to cover postage.

LE'VZ IS (C) COPYRIGHT.

NEW MEMBERS must start by sending $4.00 as I do not charge a yearly subscription. This makes it worth while entering your name, address and other data into our D'BASE. You then receive the current issue if it is in a certain time period between the main send LE'VZ runs. If that is close to the next issue, you will recieve that and not the "old" current issue. New Members can send more than $4.00, as long as it is in multiples of $2.00, but nor more than $10.00.

Present OOPs have their $ credit printed at the top of their name and address label if sent in the main run. If your credit is less than $2.00., then a little reminder slip is included with the LE'VZ sent, stating that this is your last issue. Some folk have various money amounts left over from other software or hardware purchases put into their LE'VZ credit, and so odd $ amounts do occur.

BACK ISSUES are from #16 to the current issue. The price is $3.00 each. This includes surface/air postage within Australia and Air Mail to New Zealand. If you require more than two copies at one time, extra money must be sent to cover postage.

We usually have most Back Issues in stock otherwise I print more as time permits. We send what we have and back order the others for you if required. If they are not sent within a couple of months, or with the next Current issue, please remind us.

GENERAL LIST refers to OOPs who want their name, address and data made available to other OOPs when asked for. You may like to contact OOPs in your state, or OOPs with VZ200s. Not all OOPs want their name and information made public, so if that applies to you, you must answer N (no). IF YOU DO NOT WANT YOUR INFO MADE PUBLIC, ANSWER N (no) ON THE DATA SHEET. IF YOU DO NOT ANSWER Y (yes) OR N (no) THEN YOU WILL AUTOMATICALLY BE PUT ON THE GENERL LIST.

Remember, you may receive letters from OOPs months after you may have sold your VZ.

ANY COMMUNICATION to me that requires a written reply must be accompanied by a Self Addressed Stamped Envelope. Do not expect an immediate reply, as I may need to contact others to formulate an answer.

Always state your record number. That could be between A02 and A98, 301 and B98 or C01 and C98. I have about 240 financial and unfinancial folk to keep track of. From LE'VZ #15, your record number and $ credit are printed at the top of your name and address label.

DO NOT TELEPHONE ME ON SUNDAY!!!

CIRCUIT, ROM and PROGRAMME LISTING PRINTOUTS can be sent to you at 50C per A4 page plus postage. Do not ask for the complete VZ ROM listing as it is very long and is about 15MM in thickness.

LETTERS TO THE EDITOR are welcome either as general comments, complaints or asking for help. As with contributors, please ensure that your typewriter or printer prints clear and DARK. In the new 35 character normal size print, IE. 90MM line length, right justified or wragged. If you have to write by hand, use a RED pen and write in the format just mentioned.

ADVERTISING is a free service to OOPs who are financial, for personal use only. Please use the above 35 character format. About 100 words or less.

CONTRIBUTIONS are very welcome. Please write your letter on a separate piece of paper to your contribution, which allows separate filing of material. You can send in programme listings in M/L or BASIC. Hardware modification or equipment drawings. Hints and any useful information. As above, use the new 35 character format except if it is a large circuit, drawing or photo. If it is a full page contribution reduce by photo copying so that there is a 20MM margin all the way around.

In fact I would like to receive more hardware contributions. Also photos of your equipment would interest others. There is a little problem here though as different photo copyiers reproduce certain colours differently. We can but try.

BASIC AND M/L PROGRAMME LISTINGS need special requirements. Programme listings in M/L or BASIC can be sent as printed in normal size print which I can reduce-copy to make the master. Please make sure the print is dark and clear. The better approach is to send the programme on disc or tape. This enables me to give it a short test and check that it does at least does RUN. I can then print it in reduced mode while <LISTing> it.

The other method that I can now use is to convert the *BASIC* programme into a file suitable for use with our *QUICKWRITE TEXT EDITOR* and printed in the preferred column size, IE 80MM width. Note that due to "line wrap-around" when printing a line with less than 35 characters there will be lines that appear to not have a *BASIC* line number.

W files made with the D.S.E. Editor Assembler can also be loaded into our *QUICKWRITE TEXT EDITOR* , edited and printed as required.

TAPE/DISC CONTRIBUTIONS are therefore the best to send in this regard. This applies to programme listings or text. In regards to text, please send on E&F Wordprocessor tape which I can convert to QUICKWRITE Wordprocessor files or QUICKWRITE files on disc. Send in a padded post bag, and we will return it to you as soon as possible. We will pay the return postage. In this way if it is a programme, it can be later issued as a PUBLIC DOMAIN programme. You must let me know if you will allow this to happen.

Finally, I do not promise to print any or all contributions, this is at my discretion.

Muchas Gracias.

INFORMATION CONTACTS

Here are some other folk who you can contact. Always include a SASE. if you require a written reply. If you don't live in the same country, send a couple of International Reply Coupons. These are available a Post Offices througout the world. Please use good judgement if you telephone, perhaps not on Sundays. Check with the person concerned.

Graphics, M/L, printer info, educational.
Mr.Larry Taylor, 4 Columbia Court, SPRINGWOOD. QLD. 4127. 'phone (07)208 1258.

M/L, hardware, BASIC programming and his special list of all types of info.
Mr.Bob Kitch, 7 Eurella St., KENMORE. QLD. 4069. 'phone (07)378 3745.

Software list.
Mr.Eddie Tomes, 3 Kilkenny St., CAPALABA. QLD. 4157. 'phone (07)390 2797.

General info.
Mr.Stan Noble, 307 Mt.Crosby Rd., CHUWAR. QLD. 'phone (07)281 7854.

Communications, Modems, RTTY.
Mr.Irving Spackman, 78 Waima Crescent, TITIRANGI. AUCKLAND.
New Zealand.

RTTY Units.
Mr.Col Paton. VK4BCP. 225 Pallas St., MARYBOROUGH. QLD. 4650
'phone (071)221 090.

Chip 8 programming.
Mr.Jeremy Lee, c/o P.O. Box 221, ASHGROVE. QLD. 4060
'phone (07)379 7988.

* * A CONTENTS LIST * *

LE'VZ FROM #17 AUGUST '87 to #21 NOVEMBER '88.

As mentioned on page 11 of this LE'VZ, back issues are available from #14 to the current issue. If there is a particular article or item in the earlier issues that you would like, please see if you can obtain/borrow that LE'VZ to enable you to get the item that you are after. Failing that, I may be able to copy that item ONLY for you at 20 cents per A4 page.

MORE SORTING

By Mr Gordon Browell.

Sorting is what computers are extremely good at, so here is another contribution from Gordon. I have published other sorts of programmes contributed by Gordon in earlier LE'VZs, this one deals with sorting shells.

SHELL SORT

Nothing to do with Mary-Anne sorting shells by the sea shore. A Mr. D. Shell devised this one. And he must have been either a gambler or a student of the Laws of Chance. He considered that if he tinkered with the random list before Bubble Sorting them, the quicker it would run. By doing a sort of organised random sort he would increase the chances of the final sort zipping through the list like streaked lightning. He was right. Then bigger the list the better the chances. So here we habe a sort that is suited for very long lists.

You will need your 13 cards and TWO coins. Any coins will do. IBM owners use $1 coins while VeeZedders make do with one cent pieces. Lay the cards in a row in random order. It will help if you put the KING somewhere about the middle of the row. Place a coin below the leftmost card, card no.1. This will be called the COINCARD. Now for the rules according to Mr Shell....

Count the cards - 13. Divide by 2 (ignore any remainder) - 6
The 6th card from the left is called the LINK.
The number 6 is called the LINK NUMBER.
Use your other coin to mark the space between the 6th and 7th card.

ADD LINKNUMBER to COINCARD. 6 + 1 = 7.
Now ADD the 7 to the LINK NUMBER. 6 + 7 = 13
PUSH UP cards 1, 7 and 13. We would also push up 14 but there isn't a card 14. We only have 13.
The cards pushed up form the first CHAIN.
BUBBLE SORT the three cards.
In case you've forgotten how, here's what to do...

Compare and sort cards 1 and 7.
Compare and sort cards 1 and 13.
Compare and sort cards 7 and 13.
Here ends the 1st CHAIN.
Push the cards back in line.

Move the coin from card no.1 to card no.2. This is the new COINCARD.
ADD LINKNUMBER to COINCARD 2 + 6 = 8.
PUSH UP cards 2 and 8. As before we cannot push up card 14 as there isn't one.
Compare and sort 2 and 8.
That ends the 2nd Link.
Push the cards back in line.

Don't fret. It will get more exciting in a moment.

Move the coin to card no.3. This is the new COINCARD.
ADD LINKNUMBER to the COINCARD. 6 + 3 = 9.
PUSH UP 3 and 9.
Compare and sort 3 and 9.
End of 3rd LINK.
Push the cards back in line.

Continue moving the coin and comparing and sorting as above until you have completed the 6th CHAIN. That is when the coin reaches card 6.

Take the coin back to 1.
Divide the LINKNUMBER by 2 (ignoring any remainder) 6 ÷ 2 = 3.
LINKNUMBER is now 3.
You will be adding by 3s this time.
Place 2nd coin between cards 3 and 4.
PUSH UP cards 1, 4, 7, 10 and 13.
Compare and sort 1 & 4, 1 & 7, 1 & 10, 1 & 13.
Compare and sort 4 & 7, 4 & 10, 4 & 13.
Compare and sort 7 & 10, 7 & 13.
Compare and sort 10 & 13.
PUSH DOWN the cards. Move the coin to no.2.
PUSH UP cards 2, 5, 8 and 11.
Compare and sort 2 & 5, 2 & 8, 2 & 11.
Compare and sort 5 & 8, 5 & 11.
Compare and sort 8 & 11.

GOTO 13

Studio Ad Lib

PUSH DOWN the cards. Move the coin to 3 (the LINKCARD).
PUSH UP 3, 6, 9 and 12.
Compare and sort 3 & 6, 3 & 9, 3 & 12.
Compare and sort 6 & 9, 6 & 12.
Compare and sort 9 & 12.
PUSH DOWN the cards.
BUBBLE SORT

The coin has reached the LINKCARD so it goes back to no.1. Dividing the LINKNUMBER by 2 (ignoring any remainder) = 1, so with the coin back at no.1, compare no.1 with no.2, swap if necessary. Then coin on no.2, compare and sort 2 & 3, move coin, 3 & 4, move coin, 4 & 5 and all the way up until the coin reaches card 12. Then from 1 to 11, 1 to 10 etc. until the last PASS is done, which will be 1 & 2.

All we need now is a program that will do all that. In fact I got so carried away with the success of the first example, I prepared 2 examples. The first READs Playing Cards from DATA. PRINTs the list so that you can see the cards changing places. A SOUND indicates the beginning of the final BUBBLE SORT. And, believe me, it's fast.

NOTE: The words in DATA are preceded by single letters to allow them to be sorted in alphabetical order, and each item is 8 characters long. The words used for variables relate to the explanation above.

```
0 '******EXAMPLE OF SHELL SORT**********MICRO MAGIC**********
100 CLEAR3000:LIMIT=100:DIMZ$(LIMIT):CLS
110 DATAE   FIVE,B    TWO,J    TEN,A    ACE,H   EIGHT
114 DATAM   KING,C   THREE,D   FOUR,I   NINE,F    SIX
118 DATAK   JACK,G   SEVEN,L   QUEEN
130 FORI=1TO13:READZ$(I):NEXT
140 LIMIT=I-1:LCARD=INT(LIMIT/2):LNUMB=LCARD+1
155 'SHELL SORT
160 GOSUB300
170 A=1
180 FORCOIN=ATOLCARD
190 FORCHAIN=LCARDTOLIMITSTEPLNUMB
200 IFZ$(COIN)>Z$(CHAIN)THENGOSUB400
210 IFCHAIN>LIMITTHENLCARD=INT(LNUMB/2):GOTO220
220 NEXT
230 A=A+1
240 NEXT
255 'BUBBLE SORT
260 FORP=1TOLIMIT
262 FORI=1TOLIMIT-P
270 IFZ$(I)>Z$(I+1)THENT$=Z$(I):Z$(I)=Z$(I+1):Z$(I+1)=T$
280 NEXT
290 NEXT:SOUND10,1:GOSUB300
299 GOTO299
300 PRINT@0,"LIST":FORI=1TO13:PRINTZ$(I):NEXT:SOUND22,2:RETURN
400 TEMP$=Z$(COIN)
410 Z$(COIN)=Z$(CHAIN)
420 Z$(CHAIN)=TEMP$
430 GOSUB300
440 RETURN
```

The second example is along the same lines as the first, only this time the items to be listed are INPUT by you. If, instead of typing in an item, you just hit RETURN, the program will go into the sorting routines automatically. The unsorted list is printed and then replaced by the sorted list.

Studio Ad Lib

* * CHRISTMAS VZ EXPOSITION * *

The Christmas VZ Expo held under the banner last year of the Brisbane VZ Users Workshop on the 3rd of December 1988 at the Capalaba State School was well prepared to give attendees a very worthwhile showing of the VZ.

Even though Mr Eddie Tomes arranged some publicity in the local newspapers and folk were given plenty of notice in LE'VZ, it was unfortunately a very poorly attended event. A most disheartening day. About twenty five was the attendance and that included we demonstrators.

Much good software and hardware was on display. I demonstrated our *QUICKWRITE TEXT EDITOR* for the first time.

Jerimy Lee displayed his IBM Look-alike VZ. A photo I took is not good enough to print.

I donated a "Lucky Demonstrators" door prize, and believe it or not; Mr Bob Kitch won it.

We think that may have been the last such VZ Expo.

Below is a photo of Eddie Tomes at his demonstration station.

Thanks Eddie.

SOFTWARE FOR SALE FROM VSOFTWAREZ

39 Agnes St., TOOWONG. QLD. 4066. AUSTRALIA. (07) 371 3707.

FEBRUARY 1989.

We discontinued most of our software as from the 1st. of November 1988.

We will only stock the most popular units.
The list under "EXISTING SOFTWARE" is items we will continue to sell. Those marked "+LL" include a LLISTing so that you can modify it to suit your own needs.
The list under "DISCONTINUED SOFTWARE" is what we still have in stock. Other items not listed at all will not be supplied. I have printed a list of software writers who may supply items direct to purchasers. It is just not viable for us to stock items that are not selling. Prices of most software is now reduced to clear stocks. We trust you understand.

All prices are correct at time of printing, but may change without notice. All articles available while stocks last. All prices in A$.
All tape software includes postage up to four tapes.
When ordering software, always state := which computer VZ200 or VZ300, if you have an expansion RAM unit , and if you have a disc drive system connected or denote as below.

VZ1 = unexpanded VZ200. VZ2 = unexpanded VZ300.
VZ3 = expanded VZ200. VZ4 = expanded VZ300.

IE. TU6 = Tape only unit of U6. DB46 = Disc only unit of B46.
D/TU19 = Tape or Disc unit available of U19.
The price stated is for a Tape unit. If a Disc unit is required, add $5.00. to the Tape price. The price of a Disc unit is as stated.

We accept BANKCARD and VISACARD, as well as bank, building society, credit union, private cheques, or Aust Post money orders.
Make cheques payable to J.D'ALTON or VSOFTWAREZ.

* * * NEW SOFTWARE * * *

DTG58. FACTORY. $15.00. VZ3-VZ4.
This is another one of Larry taylors educational/game units. It is a problem solving eductional/game for all ages. The main aim is to set up a factory to duplicate a product which has been designed by the VZ or someone. Eight machines can be used to ROTATE, STRIPE and PUNCH a blank square into a finished product. All in High Resolution.

DB60. QUICKWRITE TEXT EDITOR V4 II $40.00. 64K RAM Pack is a must.
This new unit is based on the QUICKWRITE WORDPROCESSOR. All the features of QW V3 and V4 are included, plus many more. The unit is probably the largest M/L software written for the VZ. You must have a 64K RAM expansion installed as the three top 16K banks are switched by the software as required. The file space for your document is about 40K which is ample for most requirements.
The unit is listed in the Australian Personal Computer magazines' Software Guide 1988. Files saved by QUICKWRITE V3 and V4 can be loaded as normal.
We will not allow any discount for previous purchasers of QW V3 or V4.
Tape files made with the old DSE E&F WP can be also loaded. Another very useful feature is the ability to also load M/L source code files made with the DSE Editor/Assembler. The SET UP MODE is where one sets up the different printer commands IE. line length, column length, margin, page, gap, tab, indent, double spacing, number of copies etc. These are all saved on the disc document file which means the user saves time when loading the file at another time.
OH yes, disc files can be MERGED with another file that is already in memory!!
SCREEN ECHO is another feature which gives the user *WYSIWYG (What You See Is What You Get)* which is great for column text with less than 31 characters, but is still helpful with longer lines, "wrap around" notwithstanding.
A special CONVERSION programme is included which allows the loading of *BASIC* programmes which do not have any *EXTENDED BASIC* commands written in them. See page three and four. An instruction booklet is of course included.

QUICKWRITE AND TEXT EDITOR CAN ONLY BE PURCHASED FROM US.

DTG59. NAME THE TOWNS. $5.00. VZ3-VZ4. This is another piece of educational software written by Mr Larry Taylor.
The aim is to enter the name of the town corresponding to the location of a flashing point on a map of Australia.
There are four selectable skill levels. It is suitable for children in years 4 - 7 (ages 8 - 12).
A High-Res game.

+ EXISTING SOFTWARE +

D/TB1	CASH BOOK LEDGER	$ 20.00.	VZ3-VZ4. +LL.
D/TE4	MATHS COUNTDOWN	$ 5.00.	VZ3-VZ4.
DB4	LE'VZ D'BASE	$ 50.00.	VZ3-VZ4. +LL.
TU6	VZ EXTENDED BASIC	$ 20.00.	VZ1-VZ4.
DB16	CHEQUE LEDGER D.	$ 40.00.	VZ3-VZ4. +LL.
D/TU19	COPY/PROTECT.	$ 20.00.	VZ1-VZ4.
D/TU48	FILESEARCH.	$ 5.00.	VZ1-VZ4.
D/TG44	MONOPOLY.	$ 8.00.	VZ3-VZ4.
D/TG45	MONOPOLY.	$ 8.00.	VZ4.
D/TG50	ESCAPE RIVER.	$ 8.00.	VZ3-VZ4.
DB46	QUICKWRITE.	$ 40.00.	VZ3-VZ4.
D/TU49	VZ-EPSON PRINT/PATCH.	$ 10.00.	VZ1-VZ4.
DU1	CONVERT2.	$ nil	VZ3-VZ4.
See DB46 QUICKWRITE.			
D/TG53	GALACTIC EMPIRES.	$ 8.00	VZ3-VZ4.
D/U56	DISKOPS4 +2.	$ 10.00.	VZ3-VZ4.
DB57	QUICKWRITE V4.	$ 40.00.	VZ3-VZ4.
D/TG58	FACTORY.	$ 15.00.	VZ4.
D/TE59	NAME THE TOWNS.	$ 5.00.	VZ3-VZ4.

DISCONTINUED SOFTWARE

All 50% off
While stocks last.

D/TU2	EDITOR/ASSEMBLER	$ 20.00.	VZ3-VZ4.
D/TE5	COORDINATES	$ 10.00.	VZ2-VZ4.
D/TE7	MICROSCOPE	$ 8.00.	VZ3-VZ4.
DU47A	DISKOPS2	$ 10.00.	VZ4.
D/TG52	SOLO BATTLESHIPS.	$ 15.00.	VZ2-VZ4.
DPD2	PUBLIC DOMAIN.	$ 10.00.	VZ1-VZ4.

YOURS TRULY WITH CHRISTMAS HAT
AT THE CHRISTMAS VZ EXPO

DU56. DISKOPS4. $10.00. VZ3-VZ4.

This is actually called DISKOPS4 + 2. It superceedes DU47 DISKOPS2 AND DU47A DISKOPS2 which are now Public Domain at the same price of $10.00.

There are three separate utilities on the disc, and are for use with the DSE. Editor Assembler unit. There are eleven additional commands. Instructions are included. DISKOPS4 + 2 patches in permanently with ED/ASS. It then allows LOADing, SAVEing of source code and BSAVEing object code to/from disc. BSAVEing is the same as TO: for tape.

It also includes the normal disc BASIC commands. If a disc error occurs, then DISKOPS4 + 2 BASIC is entered. ASS is to enable the return to the ED/ASS. BASIC does the reverse.

Users of DISKOPS1 and 2 are also catered for.

If anyone requires a short description of any of our software, please refer to previous LE'VZs as all software is initialy given a brief description as NEW SOFTWARE. The alternative is to send for our catalogue, *VLISTZ.* Make sure you send a S.A.S.Envelope, 230MM x 100MM.

+SOFTWARE SUPPLIERS+

This is a list of the software that we have discontinued to sell.

I have contacted most of the writers and suppliers of the units who supplied them to us, and in most cases they are willing to supply direct to purchasers. The prices that we charged for the units are NOT necessarily what these people will charge. It is up to them to charge what is appropriate.

The list of the names and addresses is below. The two letters on the right are their enitials, that is L.T. means Larry Taylor.

Another angle is to contact Mr Eddie Tomes who has compiled a list of past and present software. I printed the list in LE'VZ #21. His address is :-

3 Kilkenny St., CAPALABA QLD 4157
phone (07) 390 2797 at home.

D/TE1	KEYBOARD	L.T.
D/TE3	MEATPIES	L.T.
D/TE5	COORDINATES	L.T.
D/TE6	TOWER of HANOI	L.T.
D/TE7	MICROSCOPE	L.T.
D/TE8	BLOCK PUZZLER	L.T.
TE20	PLUS and MINUS	S.D.
TE24	MATHS	S.D.
TE25	QUEENSLAND	S.D.
TE27	EUROPEAN CAPITALS	S.D.
TE30	CAMPING	S.D.
D/TG13	SCOTLAND YARD	H.H.
TB15	DATABASE- VZ	W.W.
TG35	HAUNTED MANSION	H.H.
T/DE9	MEATPIES V2.	L.T.
T/DG36	BLACKJACK.	B.H.
T/DG37	POKER MACHINE.	B.H.
T/DG40	TRIVIAL CULT.	H.H.
T/DG41	SCOTLAND YARD 2.	H.H.
DU20	DISC GUARD.	L.T.
DU22	DISK COPY.	L.T.
D/TU48	FILESEARCH.	L.T.
D/TG51	BLOCK 1.	G.M.
D/TG42	AIRTRAFFIC CONTROLLER.	J.K.
D/TG43	LEARJET.	J.K.
D/TG54	GOLF.	G.M.
D/TG53	GALACTIC EMPIRES.	H.H.
D/TG51	BLOCK 1.	G.M.
D/TG52	SOLO BATTLESHIPS.	G.M.

L.T. Mr Larry Taylor, 4 Columbia Drive, Springwood, QLD. 4127.

S.D. Mr Stuart Dougall, 11 Dix St., Bundaberg. QLD. 4670.

W.W. Mr William White, 20 Violet St., Everton Hills. QLD. 4053.

B.H. Mr Bob Hornby, 15 Pictavia St., Toowong. QLD. 4066.

H.H. Mr Harry Huggins, 12 Thomas St., Mitcham. VIC. 3132.

G.M. Mr Garry McCleary, P.O. Box 24, Emu Plains, NSW. 2750.

J.K. Mr John Keetch, 2/32 Noble St., Allawah. NSW. 2218.

I thank all past suppliers of software to us and trust that they can attend to orders without any problems.

John D'Alton.

My note to oop, if I have one.

HARDWARE AND FIRMWARE FOR SALE.

VSOFTWAREZ, 37 Agnes St., TOOWONG. QLD. 4066.
AUSTALIA. 'Phone (07) 371 3707.

As with our software, we are also going to discontinue all hardware sales. We will be continuing to sell books.

Unlike our software prices, these do NOT include postage. Always include extra money with your order and we will send any surplus back in the parcel or put it towards any credit you may wish, such as to LE'VZ, if you are an OOP. If you wish to receive LE'VZ, read page 11.

Prices are in Australian dollars (AUD) as at the 1st. of August 1988. Items available while stocks last.
There is NO WARRANTY on used items, but all are tested OK.

BOOKS

VPROGRAMMEZ-VZ-VZ new $ 10.50 each.
Surface postage in Australia and NZ is included.
This is my own special book for beginners and advanced VZers.

VZ200-VZ300 Assembly Language Programming Manual for Beginners by Steve Olney. new $ 25.00 each.

Beginners Guide to the VZ200/VZ300 Editor Assembler by Peter Schaper. new $ 20.00 each.
This book explains in simple language how to use the Dick Smith Editor Assembler unit. The little instruction booklet that comes with the tape is not very easy to understand to many folk. Peter uses some short M/L routines to explain the use of the Ed/Ass but he does not teach you M/L as such. As I mentioned previously in LE'VZ, the book will be printed and put together when ordered. I do this as soon as possible, but there will be a delay. There are fifty eight pages of A4 size so it is good value for money.

OTHER VZ USER GROUPS & CLUBS.

AUSTRALIA.

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

VZ DOWN UNDER.
MR H.M Huggins, 12 Thomas St., MITCHAM. VIC. 3132.

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

WAVZ ENTHUSIASTS GROUP.
MR Graeme Bywater, P.O. Box 388, MORLEY. WA. 6062.

NEW ZEALAND.

VZ LINK.
MR Peter Hill, P.O.Box 1972 C.P.O. AUCKLAND. NZ.

Date .....19..... Code # if known ..... This LE'VZ number is 22.
Surname..... Mr, Mrs, Miss and Christian name.....
Address..... ..Post Code .....
Telephone number. STD( )..... Onto General List Yes/No.....
Computer. VZ200 and/or VZ300.....Any other computer.....
Printer and/or plotter.....Disc system Yes/No.....
RAM Expansion.....K. Tape recorder. VZ DTR or other.....
RS232 terminal..... Yes/No.....Modem Yes/No..... Brand.....
Interest. Business, games, M/L, BASIC, hardware, etc.....
.....

***** DATA SLIP *****

For my records I request all OOPs (Oners-Operators-Programmers) who have not recently sent me this data to please complete, cut out and send back at some time. As mentioned elsewhere in this LE'VZ, this is useful for OOPs who may like to contact other OOPs who live in their vicinity, etc. Answer N (No) if you do not wish your name put on this General List. If the answer is Y (Yes) or not answered at all, you will be put on the General List.