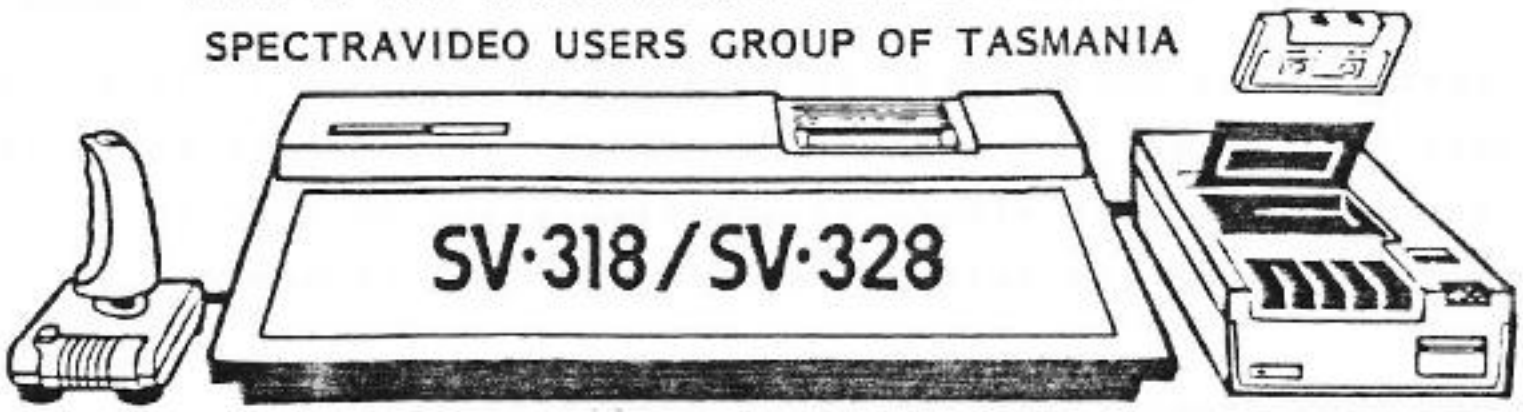
SPECTRAVIDEO

THIS IS THE OFFICIAL NEWSLETTER FOR THE

SPECTRAVIDEO USERS GROUP OF TASMANIA



News Letter

CONTENTS	PAGE

ISSUE No.
1 - 3
DATE
DECEMBER 1983

ALL CORRESPONDANCE TO:
S.V.U.G.T. P.O. BOX 191 SOUTH LAUNCESTON TASMANIA 7250

MEMBERSHIP FEES	
AUSTRALIA	$15.00
OVERSEAS	$20.00
OVERSEAS AIRMAIL ..	$25.50

Christmas Greetings

SPECTRAVIDEO

COMPUTER USERS GROUP OF TASMANIA

The Spectravideo Users Group of Tasmania (S.V.U.G.T.) has grown rapidly in the past few months. Due to the size of the group it has been necessary to alter its administration so that it will still give prompt and reliable service to you its member.

The new phone number for the group enguiries is:

(003) 312648

The newpostal address is:

S.V.U.G.T.,
P.O. BOX 191,
SOUTH LAUNCESTON,
TASMANIA 7250.

For this the Christmas issue we have included more programs than in previous newsletters and we hope that this will keep you busy over the festive season typing. We have had many members phone. us up and complain about previous programs not working when they were typed in. Note that all our programs are as bug free as humanly possible. But you never know some one may find a flaw in one of them. But to this date we have been lucky there are no serious bugs in any of our software and all the problems found by other members have been due to typing errors when they were entering from our newsletter. So please just because the program won't work as soon as you have typed it in don't panic and call us at 3am in the morning (no names), check and recheck your typing and then get a friend to check it for you. If all else fails then call the Group and we will try to sort out your problem. Corrections for major errors will be published in the following newsletter.

So have a very merry Christmas and a happy and safe New Year and I will be with you again in the New Year with the January Issue.

P.S. Keep those articles and letters comming in folks.

EDITOR.

SPECTRAVIDEO

COMPUTER USERS GROUP OF TASMANIA

THE CLOAD? COMMAND

Another undocumented command!! Did you know that if you save a program on cassette with CSAVE you can verify that it has been saved correctly (That is no CRC errors) by rewinding to the start of the program and typing CLOAD?.

The command does not load the program into the computer but compares it with the program already in memory to see if they are identical. So make sure that you do not alter the program in anyway before you verify it with CLOAD?.

E.G.

```
 5 PRINT "HELLO"
10 END
 CSAVE "PROG1"
 (rewind cassette to start of program)
 CLOAD?

 OK
```

OPEN FILES ON CASSETTE

Yet another undocumented fact. You can open a file on cassette the same way you can open a file on disk. You can write to that file and read from that file as if it were a disk file. Its not as fast as disk but its better than nothing. The following programs show you how to do this.

```
10 OPEN "FRED" FOR OUTPUT AS #1
20 READ S
30 IF S=Ø GOTO 60
40 PRINT #1, S
50 GOTO 20
60 CLOSE #1
70 DATA 6,5,7,4,8,3,9,2,Ø
80 END
```

cont next page

from previous page

```
 5 REM READ THE FILE BACK FROM CASSETTE
10 OPEN "FRED" FOR INPUT AS #1
20 IF EOF(1) GOTO 60
30 INPUT #1,S
40 PRINT S
50 GOTO 20
60 CLOSE #1
70 END
```

Note line 20 checks for the END OF FILE (EOF) on the cassette and stops when it is found. Clever what!!

ANTIPODEAN DELIGHT

Try this little program for something completely different full explination of the character generator in the next issue.

```
10 DEFINT A-Z
20 FOR I=2040 TO 4090 STEP 8
30 FOR J=7 TO 0 STEP -1
40 B(J)=VPEEK(I+(7-J))
50 NEXT J
60 FOR J=0 TO 7
70 VPOKE I+J,B(J)
80 NEXT J,I
90 END
```

now write one to give a mirror image.

SHOOTING STARS PROGRAM

The following program shows how to write a good program. It uses no graphics or sound effects and was initially written for a SOL computer.

SPECTRAVIDEO

COMPUTER USERS GROUP OF TASMANIA

```
10 ' ------------------------------------------
20 '
30 '    * * * * * * *  SHOOTING STARS  * * * * * * *
40 '
50 ' ------------------------------------------
60 '
70 '    Written by;       J van Staveren
80 '
90 ' ------------------------------------------
100 '

110 CL$=CHR$(12): PRINT CL$     'clear screen
120 PRINT:PRINT TAB(10)"* * * * *  SHOOTING STARS  * * * * *"
130 PRINT
140 PRINT:PRINT:INPUT "Do you need instructions?  ",Y$
150 PRINT:PRINT:PRINT
160 GOSUB 1040
170 IF Y$="Y" THEN GOSUB 680
180 '
        set grid to all dots.

190 PRINT
200 FOR I= 1 TO 9
210     A(I)=-1
220 NEXT
230 A(5)=1                  'make centre position a star
240 COUNT=0
250 '
              game loop
            board print out

260 COUNT=COUNT+1
270 L=0:W=0
280 PRINT CL$
290 PRINT Q$:Q$=""
300 PRINT:PRINT:PRINT:PRINT:PRINT TAB(20);
310 FOR I= 1 TO 9
320     IF A(I)=1 THEN PRINT"*  ";:W=W+1
330     IF A(I)=-1 THEN PRINT"-  ";:L=L+1
340     IF I MOD 3 =0 THEN PRINT:PRINT TAB(20);
350 NEXT
360 PRINT:PRINT TAB(10);
370 IF W=8 AND A(5)=-1 THEN PRINT:PRINT TAB(10);:GOTO 610
380 IF L=9 THEN PRINT:PRINT TAB(10);:GOTO 620
390 PRINT "Your shot No."COUNT" = ";:INPUT "",Y$
400 YY$=Y$
410 GOSUB 1040
420 IF Y$="H" THEN GOSUB 680: GOTO 270
430 IF Y>0 AND Y<10 THEN 470
440 Q$="'"+YY$+"' is not a digit between 1 and 9."
450 GOTO 270
460 '
              make move.

470 IF A(Y)=1 THEN 490
480 Q$="That position is not a star!":GOTO 270
490 ON Y GOSUB 510,520,530,540,550,560,570,580,590
500 GOTO 260
510 A(1)=-A(1):A(2)=-A(2):A(4)=-A(4):A(5)=-A(5):RETURN
520 A(2)=-A(2):A(1)=-A(1):A(3)=-A(3): RETURN
530 A(3)=-A(3):A(2)=-A(2):A(5)=-A(5):A(6)=-A(6): RETURN
```

```
540 A(4)=-A(4):A(1)=-A(1):A(7)=-A(7): RETURN
550 A(5)=-A(5):A(2)=-A(2):A(4)=-A(4):A(6)=-A(6):A(8)=-A(8):RETURN
560 A(6)=-A(6):A(3)=-A(3):A(9)=-A(9):RETURN
570 A(7)=-A(7):A(8)=-A(8):A(4)=-A(4):A(5)=-A(5):RETURN
580 A(8)=-A(8):A(7)=-A(7):A(9)=-A(9):RETURN
590 A(9)=-A(9):A(8)=-A(8):A(5)=-A(5):A(6)=-A(6):RETURN
600 '
                end of game, win or lose

610 PRINT "*** YOU WON IN"COUNT-1"SHOTS ***":GOTO 630
620 PRINT "### YOU LOSE, NO STARS LEFT ###"
630 PRINT:PRINT:PRINT TAB(20);
640 INPUT "Try again ? ",Y$
650 GOSUB 1040
660 IF Y$="N" THEN 1080 ELSE 200
670 '
                instructions subroutine.

680 PRINT "The object of SHOOTING STARS";
690 PRINT " is to change the board from"
700 PRINT
710 PRINT  " start     to     win       or    lose"
720 PRINT
730 PRINT  "-  -  -          *  *  *          -  -  -"
740 PRINT  "-  *  -          *  -  *          -  -  -"
750 PRINT  "-  -  -          *  *  *          -  -  -"
760 PRINT:PRINT:PRINT TAB(9)"The board numbers are"
770 PRINT TAB(16)"1  2  3"
780 PRINT TAB(16)"4  5  6"
790 PRINT TAB(16)"7  8  9"
800 PRINT:PRINT "Type 'H' for HELP at any time."
810 PRINT "It is possible to win in 11 shots"
820 PRINT:INPUT "Press RETURN to continue  ",A
830 PRINT "The patterns of shots are as follows"
840 PRINT:PRINT
850 PRINT "1  *  -          *  2  *          -  *  3"
860 PRINT "*  *  -          -  -  -          -  *  *"
870 PRINT "-  -  -          -  -  -          -  -  -"
880 PRINT
890 PRINT "*  -  -          -  *  -          -  -  *"
900 PRINT "4  -  -          *  5  *          -  -  6"
910 PRINT "*  -  -          -  *  -          -  -  *"
920 PRINT
930 PRINT "-  -  -          -  -  -          -  -  -"
940 PRINT "*  *  -          -  -  -          -  *  *"
950 PRINT "7  *  -          *  8  *          -  *  9"
960 PRINT
970 PRINT "Shooting a star changes it to a dot"
980 PRINT "and all other positions in the pattern"
990 PRINT "changes to the opposite."
1000 INPUT "Press RETURN to commence ",A
1010 PRINT CL$            'clear screen
1020 RETURN
1030 '
               change input to 1 upper case character.

1040 IF Y$="" THEN 1070
1050 Y=VAL(Y$)
1060 Y$=CHR$(ASC(LEFT$(Y$,1))AND &H5F)
1070 RETURN
1080 END
```

SPECTRAVIDEO

COMPUTER USERS GROUP OF TASMANIA

UNERASE A PROGRAM REVIEW.

I sat down one night to write a review of a very useful program to use with CP/M when I remembered that the program prints its own explination. So here its is from the horses mouth so to speak.

UNERAse may be followed by one or more ambiguous or unambiguous file names to be restored. Or if no names are given, will print a DIRectory of deleted files.

Use this program with caution as it is possible for more than one file to exist under the same name. If a disk has been written to but the desired file is still in the DIRECTORY it is best to UNERAse then PIP the restored file to another name or disk then re-ERAse. This will prevent the later written file being destroyed through EDiting the restored file. UNERAse logs in the inserted disk so it doesn't need to be on the same disk as the required files.

"Your seat belt is unfastened, you oil is low and that tie doesn't go with that suit."

"Oh, for Pete's sake, let's get another silicon chip."

ON STRIG EXPLINATION

Last month I asked for assistance with the STRIG function. Tim Colverd from Victoria very kindly sent this program to demonstrate the function.

```
10 CLS: STRIG(Ø)ON: STRIG(1)ON: STRIG(2)ON
20 ON STRIG GOSUB 100,200,300
30 GOTO 20
100 PRINT "SPACE BAR DETECTED":RETURN
200 PRINT "JOYSTICK 1 TRIGGER DETECTED":RETURN
300 PRINT "JOYSTICK 2 TRIGGER DETECTED":RETURN
```

Strig Ø,1,2 can be turned on or off but each one must be nominated individually.

SPECTRAVIDEO

COMPUTER USERS GROUP OF TASMANIA

LUNEY: A GAME PROGRAM
by Geoffrey Dickson.

The program LUNEY is printed in the newsletter by kind permiss
of the author Mr. G Dickson. It is a fun program were you the pilot mu
land your Space Ship on a planet and after every successful landing t
terrain becomes more difficult.

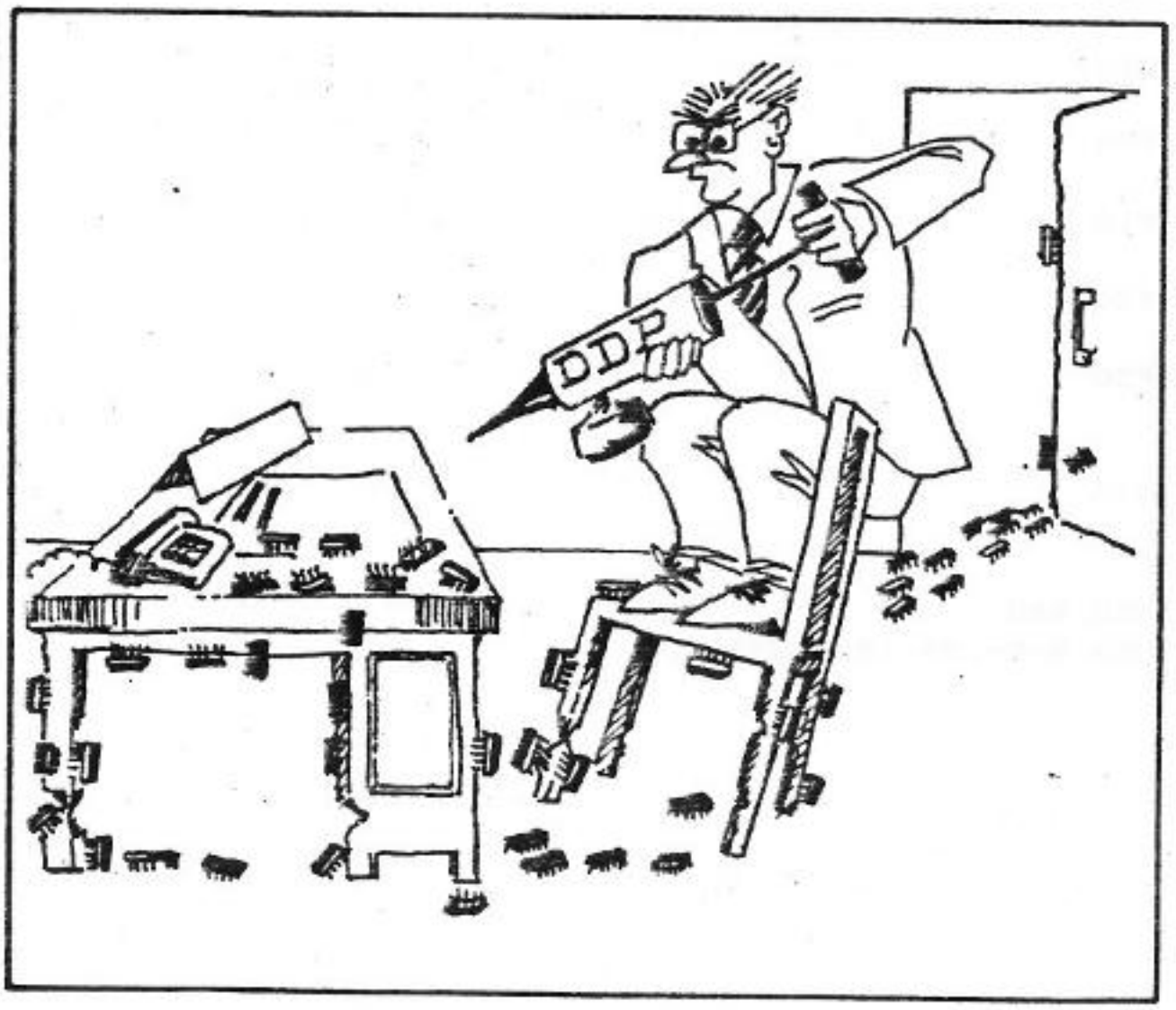
Use the Joy-Stick to land your vehicle on level terrain the Stic
controls the LEFT & RIGHT movment of the Space Ship and The Rocket moto
are controled by pushing the Stick up.

I would like to see someone add a bit to the program so that 2 S
Men jump out and plant a flag (AUSTRALIAN) when you successfully land.

```
10 ' THIS PROGRAM WRITTEN BY
20 '       Geoffrey   Dickson
30 '
40 '
50 COLOR15,1,1:SOUND7,8
60 SCREEN 1,0
70 FOR I=1 TO 8
80 READ A$:S$=S$+CHR$(VAL("&H"+A$))
90 NEXT
100 SPRITE$(1)=S$
110 FORI=1 TO 8
120 READ A$:F$=F$+CHR$(VAL("&H"+A$))
130 NEXT
140 SPRITE$(0)=F$
150 F=800
160 V=0:Z=20
170 CLS
180 FORI=1TO20:LOCATERND(9)*250,RND(9)*140:PRINT".":NEXT
190 CIRCLE (150,40),14,11,,,.9
200 PAINT (150,40),11
210 L=L+1
220 ON L GOSUB 590,610,630,650,670,690,710,730
230 READ N
240 Q=0
250 OX=0:OY=176
260 FOR I=1 TO N
270 READ X,Y
280 LINE (OX,OY)-(X,Y),13
290 OX=X:OY=Y
300 NEXT
310 PAINT(1,177),13
```

```
320 PUT SPRITE 1,(Z,V),3
330 D=STICK(0)+STICK(1):IFD<>1THENSOUND10,0
340 IF D>0 THEN GOSUB 760
350 T=V
360 Q=Q+.5*.05
370 V=V+Q
380 T=V-T
390 IF POINT(Z,V+8)=13 OR POINT(Z+8,V+8)=13 THEN 440
400 IF POINT(Z+4,V)=13 THEN 480
410 IF POINT(Z+3,V+5)=13 THEN 480
420 IF POINT(Z+5,V+5)=13 THEN 480
430 GOTO 320
440 IF POINT(Z,V+8)<>13 OR POINT(Z+8,V+8)<>13 THEN 480
450 IF T>1 THEN 480
460 COLOR4:LOCATE 0,0:PRINT" YOU HAVE LANDED";:PLAY"O6M12564T255ACEG#"
470 GOTO 510
480 COLOR8:LOCATE 0,0:PRINT" YOU HAVE CRASHED..";:PLAY"O1T255M550AADA"
490 PUT SPRITE 0,(Z,V),8
500 L=1
510 COLOR11:PRINT"    ANOTHER TRY...?"
520 SOUND10,0
530 A$=INKEY$
540 IF A$="Y" THEN 150
550 IF A$="N" THEN END
560 GOTO 530
570 DATA 18,18,3C,3C,3C,7E,42,E7
580 DATA 1E,3A,EE,5C,8C,19,98,0E
590 RESTORE 870
600 RETURN
610 RESTORE 880
620 RETURN
630 RESTORE 890
640 RETURN
650 RESTORE 900
660 RETURN
670 RESTORE 910
680 RETURN
690 RESTORE 930
700 RETURN
710 RESTORE 950
720 RETURN
730 L=1
740 RESTORE 870
750 RETURN
760 F=F-1
770 IF F<0 THEN 860
780 IF D=1 THEN Q=Q-.5*.15:SOUND10,8
```

```
790 IF D=3 THEN Z=Z+1
800 IF D=7 THEN Z=Z-1
810 IF D=5 THEN Q=Q+.5*.15
820 IF D=2 THEN F=F-1:Z=Z+1:Q=Q-.5*.15
830 IF D=8 THEN F=F-1:Z=Z-1:Q=Q-.5*.15
840 IF D=4 THEN F=F-1:Z=Z+1:Q=Q+.5*.15
850 IF D=6 THEN F=F-1:Z=Z-1:Q=Q+.5*.15
860 RETURN
870 DATA 3,40,168,48,179,256,179
880 DATA 9,24,184,56,168,88,168,96,176,120,176,136,168,176,160,256,160,256,1
56,160
890 DATA 11,8,168,24,160,56,176,88,160,112,168,136,168,160,184,176,184,192,1
32,152,256,160
900 DATA 8,64,64,104,32,104,40,114,40,136,128,168,128,192,120,256,144
910 DATA 16,8,168,15,168,48,16,64,56,64,72,48,88,48,104,64,104,72,112,88,104
920 DATA 88,88,80,80,104,72,112,72,128,80,256,96
930 DATA 14,16,168,40,8,112,16,192,96,232,120,208,120,168,96,104,144,104,152,
,152,176,128,216,160,224,144
940 DATA 256,152
950 DATA 25,0,24,32,16,56,10,240,40,240,158,224,152,224,112,208,80,168,80
960 DATA 144,104,144,136,128,160,88,155,88,176,104,176,128,184,168,160,168,1
76,112,192,96
970 DATA 200,112,192,128,192,176,208,168,256,192
```



SPECTRAVIDEO

COMPUTER USERS GROUP OF TASMANIA

Z8Ø DISSASSEMBLER IN BASIC

by J. Parry

The following is an excellent program it allows you to dissassemble any machine code program that is in memory. For example if you give the address ØØØØ to 7FFF you will be able to dissassemble BASIC itself.

As the program is very large we have had it reduced to save space in the news letter, so we advise that you invest in a magnifying glass before you start to type it in.

Firstly a brief explanation of some of the main variables and a sample run of the program so you can see what you can expect.

VARIABLE	FUNCTION
CP$	(Eventually) Contains the current op-code
P	Points at the current byte
XX$, XX, LL	Passes and returns values to the hex-decimal and decimal-hex subroutines
LN	The length of the current instruction, or 0 to indicate an exceptional instruction
B$	The current byte in hex
V$	Any argument to be inserted into an op-code
F$	The current address plus a space plus the bytes of the current instruction

```
Answer these questions with 4 digit hex numbers

Where is the code to be processed? D523

Where is the  top of  the    code? D5BE

Where does the code actually  run? D523

D523 22298C          LD (8C29),HL
D526 DDE5            PUSH IX
D528 E1              POP HL
D529 DF              RST 28AL
D52A 66              DEFB 66
D52B DD7E00          LD A,(IX+00)
D52E DF              RST 28AL
D52F 67              DEFB 67
D530 2A298C          LD HL,(8C29)
D533 23              INC HL
D534 DD7E00          LD A,(IX+00)
D537 77              LD (HL),A
D538 23              INC HL
D539 22298C          LD (8C29),HL
D53C C9              RET
D53D 21FFCF          LD HL,CFFF
D540 063F            LD B,3F
D542 70              LD (HL),B
D543 2B              DEC HL
D544 10FC            DJNZ D542
D546 70              LD (HL),B
D547 0640            LD B,40
D549 C5              PUSH BC
D54A 063E            LD B,3E
D54C DD21C0CF        LD IX,CFC0
D550 C5              PUSH BC
D551 B7              OR A
D552 DD7E01          LD A,(IX+01)
D555 CDCFD5          CALL D5CF
D558 E5              PUSH HL
D559 DD7E20          LD A,(IX+20)
D55C CDCFD5          CALL D5CF
D55F D1              POP DE
D560 CDDDD5          CALL D5DD
D563 3006            JR NC D573
D565 DD7E20          LD A,(IX+20)
D568 67              LD H,A
D569 DD7E01          LD A,(IX+01)
D56C DD7700          LD (IX+00),A
D56F 7C              LD A,H
D570 DD7701          LD (IX+01),A
D573 DD23            INC IX
D575 C1              POP BC
D576 10D0            DJNZ D548
D578 C1              POP BC
D579 10CE            DJNZ D549
D57B 063F            LD B,3F
D57D DD21C0CF        LD IX,CFC0
D581 0E14            LD C,14
D583 C5              PUSH BC
D584 DD7E00          LD A,(IX+00)
D587 DD23            INC IX
D589 CDCFD5          CALL D5CF
D58C E5              PUSH HL
D58D FDE1            POP IY
D58F FD7E06          LD A,(IY+06)
D592 FEFF            CP FF
D594 2014            JR Z D5AA
D596 F5              PUSH AF
D597 0606            LD B,06
D599 7E              LD A,(HL)
D59A F7              RST ROUT
D59B 23              INC HL
D59C 10FB            DJNZ D599
D59E 3E20            LD A,20
D5A0 F7              RST ROUT
D5A1 F1              POP AF
D5A2 DF              RST 28AL
D5A3 68              DEFB 68
D5A4 EF              RST 28S
D5A5 200300          DEFM  
D5A8 1803            JR D5AD
D5AA C1              POP BC
D5AB 0C              INC C
D5AC C5              PUSH BC
D5AD C1              POP BC
D5AE 0D              DEC C
D5AF C5              PUSH BC
D5B0 79              LD A,C
D5B1 0E04            LD C,04
D5B3 91              SUB C
D5B4 30FD            JR NC D5B3
D5B6 C604            ADD A,04
D5B8 2003            JR NZ D5BD
D5BA 3E0D            LD A,0D
D5BC F7              RST ROUT
D5BD C1              POP BC
D5BE 2007            JR NZ D5C7
End address reached
Ok
```

SPECTRAVIDEO

COMPUTER USERS GROUP OF TASMANIA

```
10 REM ****************************************
20 REM *  Disassembler by J. Parry 21/11/82   *
30 REM *  Decodes all Z80  mnemonics and NAS  *
40 REM *  SYS restarts with address  offset   *
50 REM ****************************************
60 CLEAR 2500
70 DIM OP$(255),LN(255) :REM Mnemonics+lengths
80 DIM CG$(31)          :REM CB names
90 DIM ED(27),ED$(27)   :REM ED Mnemonics+codes
100 REM ********** Get limits**********************
110 CLS
120 PRINT "Answer these questions with 4 digit"
130 PRINT " hex numbers"
140 PRINT
150 INPUT "Where is the code to be processed";B$
160 XX$=B$:LL=4:GOSUB 3000:P=XX:IF P<0 THEN 150
170 PRINT
180 INPUT "Where is the  top of  the    code";T$
190 XX$=T$:LL=4:GOSUB 3000:T=XX:IF T<0 THEN 180
200 IF T<P THEN 150
210 PRINT
220 INPUT "Where does the code actually  run";V$
230 XX$=V$:LL=4:GOSUB 3000:IF XX<0 THEN 220
240 OF=P-XX:CLS
250 REM
260 REM
1000 REM ***** Control disassembly **********
1010 REM
1020 GOSUB 2900:OP$=""
1030 XX=P:GOSUB 2600:B$=XX$:GOSUB 3300:P=P+1
1040 P$=P$+" "+B$:IF LN=1 THEN 1190
1050 IF LN=0 THEN 1540:REM CB,ED,DD,FD
1060 REM
1070 REM
1180 REM ==Two & Three byte codes ==============
1110 GOSUB 2330:V$=XX$
1120 IF LN=2 THEN 1470
1130 REM
1140 REM ==Three byte normal codes ============
1150 GOSUB 2330:V$=XX$+V$
1160 GOSUB 2800:GOTO 2240
1170 REM
1180 REM
1190 REM ===Exceptional 1 byte codes ==========
1200 REM
1210 IF B$<>"DF" THEN 1270
1220 REM ====== RST SCAL ======================
1230 GOSUB 2700:GOSUB 2900
1240 XX=P:GOSUB 2600
1250 P$=P$+" "+XX$:OP$="DEFB "+XX$
1260 GOSUB 2700:P=P+1:GOTO 2250
1270 IF B$<>"EF" THEN 1300
1280 REM
1290 REM ====== RST PRS ======================
1300 GOSUB 2700:GOSUB 2900
1310 P$=P$+" "+OP$+"DEFM  "
1320 XX=P:GOSUB 2600:P$=P$+XX$
1330 LL=2:GOSUB 3000:OP$=OP$+CHR$(XX)
1340 IF XX$<>"00" THEN P=P+1:GOTO 1320
1350 P$=LEFT$(P$,20):OP$=OP$+"'"
1360 GOSUB 2700:P=P+1:GOTO 2250
1370 REM
1380 REM ====== RST RCAL ======================
1390 IF B$<>"D7" THEN 1460
1400 GOSUB 2700:GOSUB 2900:XX=P:GOSUB 2600
1410 P$=P$+" "+XX$:OP$="DEFB "+XX$+"'Calls "
1420 IF QQ>127 THEN QQ=QQ-256
1430 LL=:XX=P+QQ+1:GOSUB 3100:OP$=OP$+XX$
1440 GOSUB 2700:P=P+1
1450 GOTO 2250
1460 GOTO 2240
1470 REM === Exceptional two byte codes ======
1480 IF LEFT$(OP$,2)="JR" THEN 1510
1490 IF B$="10" THEN 1510
1500 GOTO 2230:REM not JR or DJNZ
1510 IF QQ>127 THEN QQ=QQ-256
1520 XX=P+QQ-OF:LL=4:GOSUB 3100:V$=XX$
1530 GOTO 2230
1540 REM ------ Difficult bits ----------------
1550 IF B$<>"CB" THEN 1640
1560 REM === CB codes ==========================
1570 REM
1580 GOSUB 2330
1590 OP$=CG$(QQ/8):IF QQ<64 THEN 1610
1600 OP$=OP$+CHR$(ASC("0")+(QQ AND 56)/8)+","
1610 OP$=OP$+RG$(QQ AND 7)
1620 GOTO 2240
1630 REM
1640 IF B$<>"ED" THEN 2010
1650 REM ==== ED codes ==========================
1660 GOSUB 2330
1670 IF (QQ AND 128)=128 THEN 1970
1680 IF (QQ AND 64)=0 THEN 1970
1690 D=QQ AND 15
1700 IF D<>0 THEN 1730
1710 OP$="IN "+RD$((QQ AND 56)/8)+",(C)"
1720 GOTO 2240
1730 IF D<>1 THEN 1760
1740 OP$="OUT (C),"+RD$((QQ AND 56)/8)
1750 GOTO 2240
1760 IF D<>2 THEN 1790
1770 OP$="SBC HL,"+RP$((QQ AND 48)/16)
1780 GOTO 2240
1790 IF D<>3 THEN 1850
1800 OP$="LD (V),"+RP$((QQ AND 48)/16)
1810 GOSUB 2330:V$=XX$
1820 GOSUB 2330:V$=XX$+V$
1830 GOSUB 2800
1840 GOTO 2230
1850 IF (QQ AND 15)<>10 THEN 1900
1860 D=(QQ AND 48)/16
1870 IF (D)3) OR (D<0) THEN 1970
1880 OP$="ADC HL,"+RP$(D)
1890 GOTO 2240
1900 IF (QQ AND 15)<>11 THEN 1970
1910 D=(QQ AND 48)/16
1920 IF (D)3) OR (D<0) THEN 1970
1930 OP$="LD "+RP$(D)+",(V)"
1940 GOSUB 2330:V$=XX$
1950 GOSUB 2330:V$=XX$+V$
1960 GOTO 2230
1970 REM ======Non Calculable ED's===============
1980 FOR I=1 TO 28
1990 IF QQ=ED(I) THEN OP$=ED$(I):GOTO 2240
2000 NEXT I
2010 REM ===== DD & FD codes ====================
2020 I$="IX":IF B$="FD" THEN I$="IY"
2030 GOSUB 2330:B$=XX$
2040 D=QQ:IF B$="CB" THEN 2150
2050 GOSUB 3300:GOSUB 2500
2060 IF LN=1 THEN 2100
2070 GOSUB 2330:V$=XX$
2080 IF LN=2 THEN 2100
2090 GOSUB 2330:V$=XX$+V$
2100 IF (Q<52) OR (Q>224) THEN 2230
2110 XX=P:GOSUB 2600:B$=XX$:GOSUB 2400:P=P+1
2120 B$=B$+B$
2130 GOTO 2230
2140 REM
2150 REM ===== DD or FD + CB codes ==========
2160 GOSUB 2330:B$=XX$
2170 GOSUB 2330
2180 OP$=CG$(QQ/8):IF QQ<64 THEN 2200
2190 OP$=OP$+CHR$(ASC("0")+(QQ AND 56)/8)+","
2200 OP$=OP$+RG$(QQ AND 7)
2210 GOSUB 2500:GOSUB 2400
2220 GOTO 2230
2230 GOSUB 2800
2240 GOSUB 2700
2250 IF P<>T THEN 1000
2260 PRINT "End address reached":END
2270 REM
2280 REM
2300 REM ======================================
2310 REM ****** Subroutines start here *******
2320 REM
2330 REM ==Peek the byte at P & add the hex===
2340 REM ==of it onto P$ then increment P
2350 XX=P:GOSUB 2600:P$=P$+XX$:P=P+1
2360 RETURN
2370 REM
2400 REM ==Accepts a mnemonic OP$ and byte B$
2410 REM ==Inserts "+B$" in OP$ after "IX" or
2420 REM ==IY"
2430 FOR I=1 TO LEN(OP$)
2440 ZZ$=MID$(OP$,I,2)
2450 IF ZZ$="IX" OR ZZ$="IY" THEN 2470
2460 NEXT I
2470 Z$=LEFT$(OP$,I+1)+"+"+B$
2480 Z$=Z$+MID$(OP$,I+2):OP$=Z$
2490 RETURN
2500 REM ==Accepts OP$,replaces HL with I$====
2510 REM ==which will be "IX" or "IY"
2520 FOR I=1 TO LEN(OP$)
2530 IF MID$(OP$,I,2)="HL" THEN 2550
2540 NEXT I
2550 Z$=LEFT$(OP$,I-1)+I$
2560 Z$=Z$+MID$(OP$,I+2):OP$=Z$
2570 RETURN
2580 REM
2600 REM ==Peeks byte at XX "ko XX$==========
2610 IF XX<32768 THEN QQ=PEEK(XX):GOTO2630
2620 QQ=PEEK(XX-65536)
2630 XX=QQ:LL=2:GOSUB 3100
2640 RETURN
2650 REM
2700 REM ==========Prints output line=========
2710 PRINT #1,TAB(22)OP$
2720 RETURN
2730 REM
2800 REM ==Replaces V in OP$ with V$==========
2810 Z$="":FOR I=1 TO LEN(OP$)
2820 ZZ$=MID$(OP$,I,1):IF ZZ$="V" THEN 2840
2830 Z$=Z$+ZZ$:GOTO 2850
2840 Z$=Z$+V$
2850 NEXT I:OP$=Z$
2860 RETURN
2870 REM
2900 REM ==Convert address P to hex===========
2910 REM ==return it in P$
2920 XX=P-OF:LL=4:GOSUB 3100:P$=XX$
2930 RETURN
2940 REM
3000 REM ==Hex to decimal=======================
3010 REM ==Returns value of XX$,length LL in XX
3020 IF LEN(XX$)<>LL THEN XX=-1:RETURN
3030 HH=16^(LL-1):XX=0:B$="0123456789ABCDEF"
3040 FOR II=1 TO LL
3050 FOR JJ=1 TO 16
3060 IF MID$(XX$,II,1)=MID$(B$,JJ,1) THEN 3080
3070 NEXT JJ:XX=-1:RETURN
3080 XX=XX+HH*(JJ-1):HH=HH/16
3090 NEXT II:XX=INT(XX+.1):RETURN
3100 REM ==Decimal to hex=======================
3110 REM ==Returns value of XX in XX$
3120 REM ==length required passed in LL
3130 HH=16^(LL-1):XX$=""
3140 FOR II=1 TO LL
3150 NN=XX/HH:XX=HH*(NN-INT(NN)):HH=HH/16
3160 NN=INT(NN)+ASC("0")
3170 IF NN <= ASC("9") THEN 3190
3180 NN=NN+ASC("A")-ASC("9")-1
3190 XX$=XX$+CHR$(NN)
3200 NEXT II
3210 RETURN
3220 REM
3300 REM ==Subroutine accepts byte B$=========
3310 REM ==If its not CB,ED,DD,or FD
3320 REM ==returns in OP$ the opcode
3330 REM ==of B$ with "V" in place of any
3340 REM ==arguments and length in LN
3350 IF OP$(1)<>"" THEN 3570
3360 REM ==Read data the first time called
3370 FOR I=0 TO 7:READ RG$(I):NEXT I
3380 DATA B,C,D,E,H,L,"(HL)",A
3390 FOR I=0 TO 7:READ SG$(I):NEXT I
3400 DATA "ADD A,","ADC A,","SUB ","SBC "
3410 DATA "AND ","XOR ","OR ","CP "
3420 FOR I=0 TO 7:READ CG$(I):NEXT I
3430 DATA "RLC ","RRC ","RL ","RR "
3440 DATA "SLA ","SRA ","SLL? ","SRL "
3450 FOR I= 0 TO 15:CG$(I)="BIT ":NEXT I
3460 FOR I=16 TO 23:CG$(I)="RES ":NEXT I
3470 FOR I=24 TO 31:CG$(I)="SET ":NEXT I
3480 FOR I=0 TO 3:READ RP$(I):NEXT I
3490 DATA BC,DE,HL,SP
3500 FOR I=0 TO 7:READ RD$(I):NEXT I
3510 DATA B,C,D,E,H,L,$,A
3520 FOR I=0 TO 255
3530 IF (I)63) AND (I<192) THEN 3570
3540 IF I=203 OR I=221 THEN 3570
3550 IF I=237 OR I=253 THEN 3570
3560 READ OP$(I),LN(I):LN=1
3570 NEXT I
3580 FOR I=1 TO 27:READ ED(I),ED$(I):NEXT I
3590 REM === End of data read ==================
3600 LL=2:XX$=B$:GOSUB 3000:OP=XX
3610 IF (OP)63) AND (OP<192) THEN 3630
3620 OP$=OP$(OP):LN=LN(OP):GOTO 3720
3630 IF OP=118 THEN OP$="HALT":GOTO 3710
3640 IF OP)127 THEN 3690
3650 OP$="LD "
3660 OP$=OP$+RG$((OP AND 56)/8)+","
3670 OP$=OP$+RG$(OP AND 7)
3680 GOTO 3710
3690 OP$=SG$((OP AND 56)/8)
3700 OP$=OP$+RG$(OP AND 7)
3710 LN=1
3720 RETURN
4000 REM ************************************
4010 REM Each Mnemonic is followed by
4020 REM the length of the corresponding
4030 REM OP code. Mnemonics which take data
4040 REM from following bytes contain a V
4050 REM at the position at which it will
4060 REM be inserted.
4070 REM CB,ED,DD & FD are not in this list
4080 REM because they are dealt with
4090 REM seperately.
4100 DATA "DEFB BB",1,     "LD BC,V",3
4110 DATA "LD (BC),A",1,   "INC BC",1
4120 DATA "INC B",1,       "DEC B",1
4130 DATA "LD B,V",2,      "RLCA",1
4140 DATA "EX AF,AF'",1,   "ADD HL,BC",1
4150 DATA "LD A,(BC)",1,   "DEC BC",1
4160 DATA "INC C",1,       "DEC C",1
4170 DATA "LD C,V",2,      "RRCA",1
4180 DATA "DJNZ V",2,      "LD DE,V",3
4190 DATA "LD (DE),A",1,   "INC DE",1
4200 DATA "INC D",1,       "DEC D",1
4210 DATA "LD D,V",2,      "RLA",1
4220 DATA "JR V",2,        "ADD HL,DE",1
4230 DATA "LD A,(DE)",1,   "DEC DE",1
4240 DATA "INC E",1,       "DEC E",1
4250 DATA "LD E,V",2,      "RRA",1
4260 DATA "JR NZ V",2,     "LD HL,V",3
4270 DATA "LD (V),HL",3,   "INC HL",1
4280 DATA "INC H",1,       "DEC H",1
4290 DATA "LD H,V",2,      "DAA",1
4300 DATA "JR Z V",2,      "ADD HL,HL",1
4310 DATA "LD HL,(V)",3,   "DEC HL",1
4320 DATA "INC L",1,       "DEC L",1
4330 DATA "LD L,V",2,      "CPL",1
4340 DATA "JR NC V",2,     "LD SP,V",3
4350 DATA "LD (V),A",3,    "INC SP",1
4360 DATA "INC (HL)",1,    "DEC (HL)",1
4370 DATA "LD (HL),V",2,   "SCF",1
4380 DATA "JR C V",2,      "ADD HL,SP",1
4390 DATA "LD A,(V)",3,    "DEC SP",1
4400 DATA "INC A",1,       "DEC A",1
4410 DATA "LD A,V",2,      "CCF",1
4420 DATA "RET NZ",1,      "POP BC",1
4430 DATA "JP NZ V",3,     "JP V",3
4440 DATA "CALL NZ V",3,   "PUSH BC",1
4450 DATA "ADD A,V",2,     "RST START",1
4460 DATA "RET Z",1,       "RET",1
4470 DATA "JP Z V",3,      "CALL Z V",3
4480 DATA "CALL V",3,      "ADC A,V",2
4490 DATA "RST RIN",1,     "RET NC",1
4500 DATA "POP DE",1,      "JP NC V",3
4510 DATA "OUT (V),A",2,   "CALL NC V",3
4520 DATA "PUSH DE",1,     "SUB V",2
4530 DATA "RST RCAL",1,    "RET C",1
4540 DATA "EXX",1,         "JP C V",3
4550 DATA "IN A,(V)",2,    "CALL C V",3
4560 DATA "SBC A,V",2,     "RST SCAL",1
4570 DATA "RET PO",1,      "POP HL",1
4580 DATA "JP PO V",3,     "EX (SP),HL",1
4590 DATA "CALL PO V",3,   "PUSH HL",1
4600 DATA "AND V",2,       "RST BRKPT",1
4610 DATA "RET PE",1,      "JP (HL)",1
4620 DATA "JP PE V",3,     "EX DE,HL",1
4630 DATA "CALL PE V",3,   "XOR V",2
4640 DATA "RST PRS",1,     "RET P",1
4650 DATA "POP AF",1,      "JP P V",3
4660 DATA "DI",1,          "CALL P V",3
4670 DATA "PUSH AF",1,     "OR V",2
4680 DATA "RST ROUT",1,    "RET M",1
4690 DATA "LD SP,HL",1,    "JP M V",3
4700 DATA "EI",1,          "CALL M V",3
4710 DATA "CP V",2,        "RST RDEL",1
4720 REM ************************************
4730 REM Data for those of the two byte
4740 REM OP codes of the form EB XX which
4750 REM cannot be arrived at by calculation
4760 REM The number is the byte XX in decimal
4770 DATA  68,"NEG",    69,"RETN",    72,"IM 0"
4780 DATA  77,"RETI",   79,"LD R,A",  86,"IM 1"
4790 DATA  87,"LD A,I", 95,"LD A,R",102,"IM 2"
4800 DATA 103,"RRD",   111,"RLD",   160,"LDI"
4810 DATA 161,"LDI",   162,"INI",   163,"OUTI"
4820 DATA 168,"LDD",   169,"CPD",   170,"IND"
4830 DATA 171,"OUTD",  176,"LDIR",  177,"CPIR"
4840 DATA 178,"INIR",  179,"OTIR",  184,"LDDR"
4850 DATA 185,"CPDR",  186,"INDR",  187,"OTDR"
4860 REM ********** The end **********************
Ok
```

SPECTRAVIDEO

COMPUTER USERS GROUP OF TASMANIA

CUBE : PROGRAM

---

by S Lane, VIC.

The following program will rotate a three dimensional cube on the screen.

```
100 DIMP(8,3)
110 DATA-.5,-.5,-.5,-.5,.5,-.5,.5,.5,-.5,.5,-.5,-.5
120 DATA-.5,-.5,.5,-.5,.5,.5,.5,.5,.5,.5,-.5,.5
130 CLS
140 INPUT"size (1-100)";S
150 INPUT"colour (0-15)";R
160 XSTART=120:YSTART=90
170 FORI=1TO8 :FORJ=1TO3
180 READP(I,J):NEXTJ,I
190 FORI=1TO8:FORJ=1TO3
200 P(I,J)=S*P(I,J)
210 NEXTJ,I
220 TH=45:PH=45
230 INPUT"rotation degrees x axis ";DTH:DTH=DTH*.0175
240 INPUT"rotation degrees y axis ";DPH:DPH=DPH*.0175
250 TH =(TH*3.1459)/180
260 PH =(PH*3.1459)/180
270 FORX=1TO2STEP0
280 STH=SIN(TH):CTH=COS(TH)
290 SPH=SIN(PH):CPH=COS(PH)
300 FORI=1TO8
310 TE=P(I,1)
320 P(I,1)=P(I,1)*CTH-P(I,2)*STH
330 P(I,2)=P(I,2)*CTH+TE*STH
340 P(I,2)=P(I,2)*CPH-P(I,3)*SPH
350 P(I,1)=P(I,1)*.6+XS :P(I,2)=P(I,2)+YS
360 NEXTI
370 SCREEN1
380 LINE(P(1,1),P(1,2))-(P(2,1),P(2,2)),R
390 LINE(P(1,1),P(1,2))-(P(4,1),P(4,2)),R
400 LINE(P(4,1),P(4,2))-(P(3,1),P(3,2)),R
410 LINE(P(2,1),P(2,2))-(P(3,1),P(3,2)),R
420 LINE(P(5,1),P(5,2))-(P(6,1),P(6,2)),R
430 LINE(P(5,1),P(5,2))-(P(8,1),P(8,2)),R
440 LINE(P(8,1),P(8,2))-(P(7,1),P(7,2)),R
450 LINE(P(6,1),P(6,2))-(P(7,1),P(7,2)),R
460 LINE(P(1,1),P(1,2))-(P(5,1),P(5,2)),R
470 LINE(P(2,1),P(2,2))-(P(6,1),P(6,2)),R
480 LINE(P(3,1),P(3,2))-(P(7,1),P(7,2)),R
490 LINE(P(4,1),P(4,2))-(P(8,1),P(8,2)),R
500 TH=TH+DTH:PH=PH+DPH
510 RESTORE
520 FORI=1TO8:FORJ=1TO3
530 READP(I,J)
540 P(I,J)=P(I,J)*S
550 NEXTJ,I
560 NEXTX
```

SPECTRAVIDEO

COMPUTER USERS GROUP OF TASMANIA

SOFTWARE LIBRARY UPDATE

We have been held up on some software this month but things are now fixed and all members waiting for programs will recieve them in a few weeks.

The new program in the library is RESCUE and a review follows. I hope to have much more on the library next month. Keep those programs comming in we need many more.

RESCUE - Review

Written by : Tony Cruise.

It is the year 2050, the lives of many astronauts are in danger, as the moon is currently passing through a huge meteor shower.

You must despatch your shuttle and try to save them before it is to late. Using your trusty keyboard Joystick and Space Bar or any Atari compatable joystick, you must manuver your small blue shuttle onto the landing pads below. On the way down you can slow your descent by pressing the Fire Button or Space Bar and on the way up shoot at the asteroids in the same manner. After collecting an astronaut from the surface you must try to dock with the Mother Ship that is orbiting above you. Be careful with your docking or you could become along with the astronaut you are currently carrying, SPACEDUST.....

JP-80A Printer

As of next month the newsletter will be written on a word processor and printed on a JP-80A printer as this page was. We hope that this will improve the quality of the newsletter a bit and also allow better proof reading.

Justwrite JP-80A worktape.

Nick.