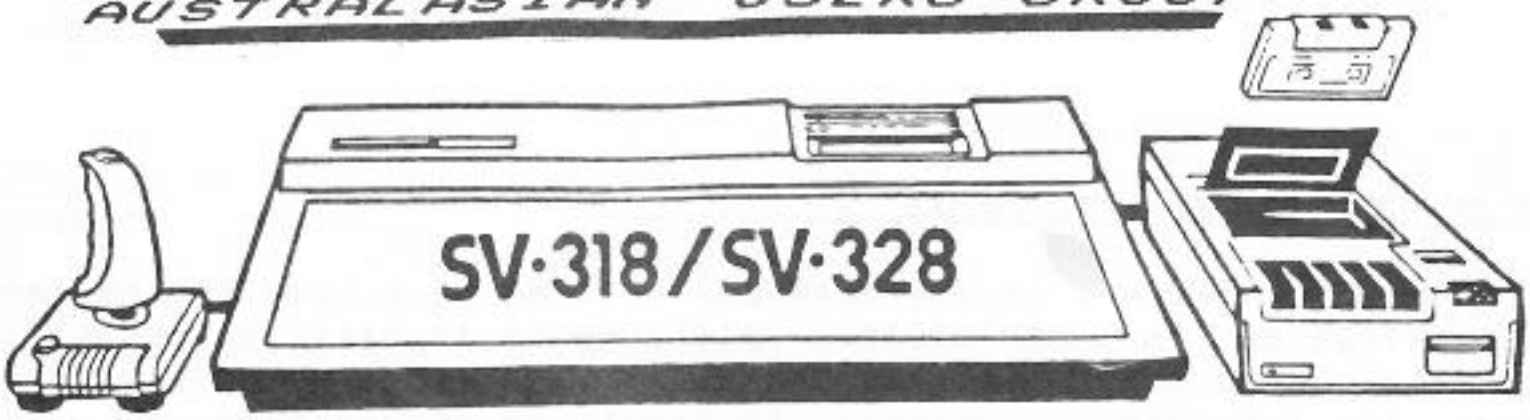
SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP



News Letter

CONTENTS	PAGE

ISSUE No.
1 - 7
DATE
APR - 1984

ALL CORRESPONDENCE TO:

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

(003) 312648

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

SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP

EDITOR'S COMMENT

May I thank Mr. K. Beattie for sending me a book on the language specification for MSX BASIC. Although it is still not exactly the same as our SV BASIC, with it and other literature we have gathered we have the information necessary to create an accurate manual. It is envisaged that the book we will create will be about 90 pages long. But before we spend the time creating it I would like some feedback from members to see how many would be interested to buy the finished book. It won't be on glossy paper or in an expensive binding but it will come out looking much like the present newsletter. Due to it's size it will cost about $20.00 each. So if enough interest is shown we will be more than happy to put the time into producing it.

Next subject. To the person who borrowed my Nevada Pilot Manual and my Inside CP/M book please return them as they are needed desperately. I'm not loaning any other books until they are returned so if you don't want to ruin it for other members please return them as soon as you can.

Over the last month we have received a large amount of programs which we are sifting through, there are alot of good programs which will appear in the newsletter in the next few months.

On my thank you list for this month are Mr. L.A. Dunning for his continued and informative article, Mr. D. French for two excellent programs of the month and to Mr. M.J. Tyeson for his ever persistent bashing of the keyboard till all hours of the morning. YAWN.

I have heard a rumor and mind you it's still only a rumor, that some new models of the SV 318 and SV 328 are on the way soon (month or two). It will be interesting to see the difference (P.S. can't say anything at the moment).

We have run out of back issues of the newsletter and it will be a few months before we have a reprint.

Well that's all from me for this month, hope you enjoy the newsletter.

Sorry about it being late again.

CSOPE

by T. Colverd

Cscope is a program you run when you aren't running a program. Try changing some of the RND functions.

```
10 REM ....COLOR-EYED-A-SCOPE....
20 SCREEN1,1
30 D1=0:D3=1:D4=1
40 DEF FNR(X)=INT(X*RND(-TIME)+1):A=FNR(8):B=FNR(20)
50 GOTO225
100 IF RND(-TIME)>.6GOTO210
208 DA=FNR(2):DB=FNR(3)
210 A=A-DA+D3::B=B-DB+D4
221 IF RND(-TIME)>.2GOTO230
225 CL=FNR(15)
230 IF ABS(A)>45OR ABS(B)>45ORD1>20GOTO300
232 IF ABS(A)<=35 GOTO235
233 D3=-D3:D1=D1+1
235 IF ABS(B)>35THEND4=-D4
250 PSET(130+2*A,90+B),CL
255 PSET(130+2*A,90-B),CL
260 PSET(130-2*A,90-B),CL
265 PSET(130-2*A,90+B),CL
270 PSET(130+2*B,90+A),CL
275 PSET(130+2*B,90-A),CL
280 PSET(130-2*B,90-A),CL
285 PSET(130-2*B,90+A),CL
290 GOTO100
300 IF FNR(3)>1GOTO30
320 FORK=1TO4000:NEXTK
330 CLS
350 GOTO30
```

"You left your computer in the kitchen and took the mirowave to work by mistake!"

SPECTRAVIDEO™
AUSTRALASIAN USERS GROUP

EXPLORING BASIC Pt-3

by L.A. Dunning

Most of the early home computers had what was called a "Memory Mapped" screen - that is, one that used on-line RAM to store the display shown. A good example is the TRS-80 Model I which held the screen memory between 15360 and 16376. This enabled smart programmers to use all sorts of tricks / routines to manipulate the display. On the Spectravideo however the display is not memory mapped, rather a custom built system is used. This system consists of a TMS 39918/28/29 Video Display Processor, 16k of Video Memory, interface ports and software drivers.

The Video Display Processor comes in three variations so to work with different TV systems, however the function remains the same for each type. The ports referred to are the ports enabled by the use of a Z80 as CPU, there are 256 of these however the system only uses 4 of these for video. The drivers lie deep in the Basic language, however I've uncovered enough of these to produce this article.

The 9900 series of processors were designed to free RAM to the user and simplify complex video displays. There are four modes of display available, of which three are supported on the Spectravideo. The modes are :

GRAPHICS 1 : This is the unsupported mode. The screen is divided into a grid of 32 x 24 positions, each position being 8 x 8 pixels across. Up to 256 different patterns can be used and put into any position. There are however only 32 diferent foreground/background colour combinations and these are allocated to 32 groups of 8 patterns.
GRAPHICS 2 : This is screen 1. The display is divided in the same manner as Graphics 1 however there are a total of 768 different patterns, one for each different position. Also, rather than 32 colour combinations each pixel line of each position may have a different foreground or background colour.
TEXT : This is Screen 0. The display is divided into a 40 x 24 grid, enabling more words on the display that in either graphics mode. Each position is 6 x 8 pixels across and there are only 256 different patterns available. Only two colours are allowed.
MULTICOLOR : This is Screen 2. The display is divided into a 64 x 48 grid of chunky 4 x 4 pixels. All 15 colours can be used for any pixel. This mode at first appears to be the most limited, however it can be manipulated with great ease.
SPRITES : These can be used in all modes except Text Mode. The sprite drivers can however be used in this mode, to the users advantage.

To maintain the image, the 9900 requires either 4k or 16k of extra memory and uses 8 accessable registers. Lets examine each of these registers. Inspect table 1, this gives a schematic representation of the registers and a key for each function :

TABLE 1

REG.	BITS							
	7	6	5	4	3	2	1	Ø
Ø	Ø	Ø	Ø	Ø	Ø	Ø	M3	EV
1	K	BL	IE	M1	M2	Ø	SZ	MG
2	Ø	Ø	Ø	Ø	N	N	N	N
3	C	C	C	C	C	C	C	C
4	Ø	Ø	Ø	Ø	Ø	P	P	P
5	Ø	S	S	S	S	S	S	S
6	Ø	Ø	Ø	Ø	Ø	G	G	G
7	F	F	F	F	B	B	B	B
VDP	FL	5S	C	X	X	X	X	X

M3, M2 & M1 Determine the mode of display as below

Ø	Ø	Ø	=	Graphics 1
1	Ø	Ø	=	Graphics 2
Ø	1	Ø	=	Multicolor
Ø	Ø	1	=	Text

EV enables an external display, EV = 1 means the external display is active, EV = Ø means it's inactive. K determines maximum memory used, K = 1 means 16k, K = Ø means 4k. BL is an enable for the display, BL = Ø means only border colour shows, BL = 1 enables the display. IE enables the video interupt, IE = 1 is enabled, Ø equals disabled. SZ determines the sprite size, SZ = Ø is 8 x 8, SZ = 1 is 32 x 32. MG determines sprite magnitude, MG = Ø is normal, MG = 1 is double size.

NNNN defines the base address for the *NAME TABLE* (NTB), multiply this by 4ØØH to find it's starting address iin Video Memory. CCCCCCCC defines the base address for the *COLOR TABLE* (COLB), multiply this by 4ØH to find it's starting address. PPP is the base address of the *PATTERN GENERATOR TABLE* (PGB), multiply this by 8ØØH to find it's starting address. SSSSSSS is the base address of the *SPRITE ATTRIBUTE TABLE* (SAB), multiply this by 8ØH to find it's starting address. GGG is the base address of the *SPRITE PATTERN GENERATOR TABLE* (SPGB), multiply this by 8ØØH to find it's starting address. FFFF defines the foreground colour while in text mode, BBBB defines the background colour in text mode, or the border colour in other modes.

The VDP register is a read only register. FL is set to 1 at the end of a raster scan and reset to Ø when a number of conditions are met. C is a flag which is set to 1 if two sprites coincide. 5S is a flag that is set to 1 whenever there are 5 or more sprites on the same line; if there are then XXXXX gives the number of the fifth sprite.

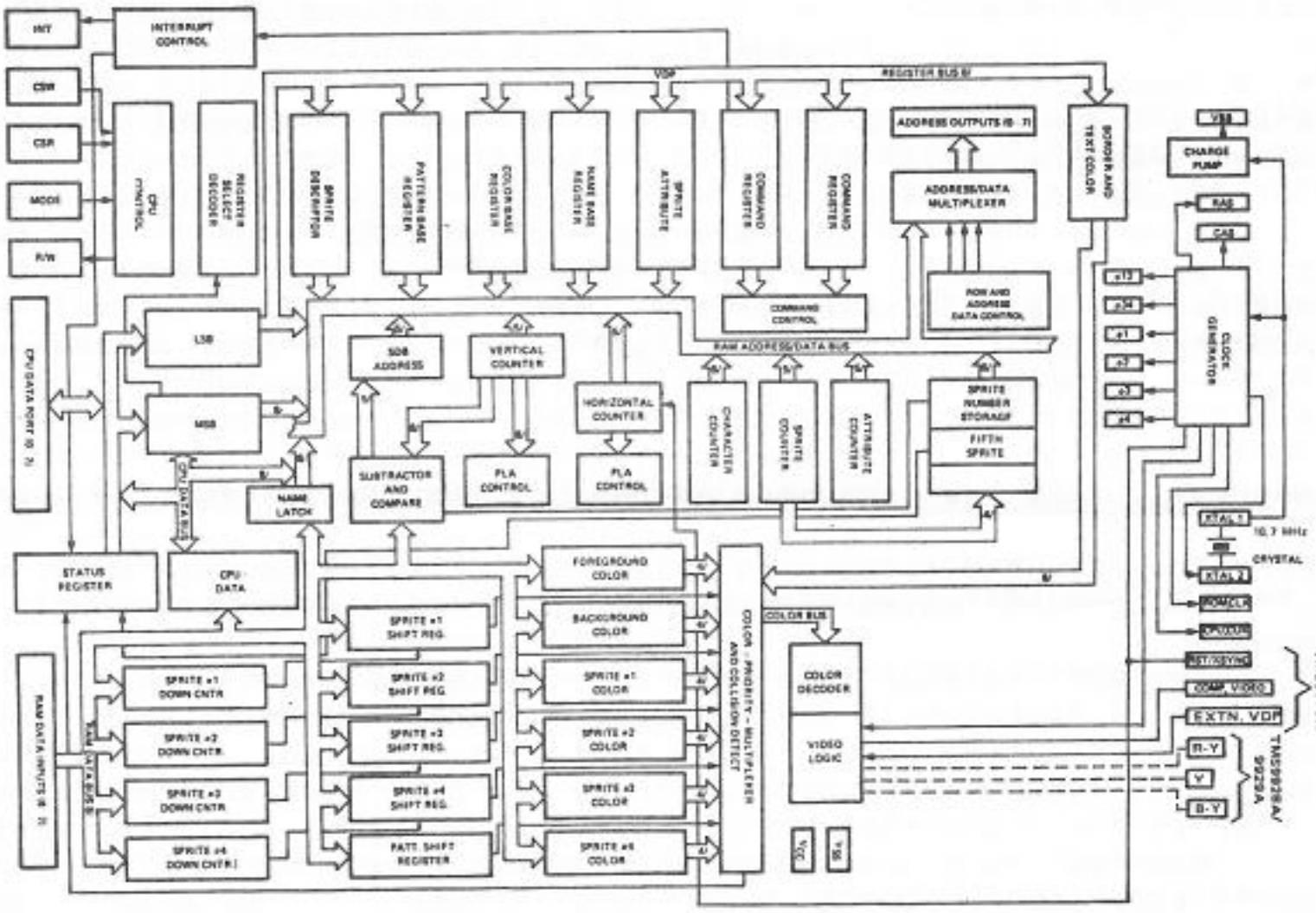
How is the processor connected to the computer? The Z8Ø uses ports 128, 129, 13Ø/131 & 132 to interface with the chip. There are four operations posible using these ports; you can write a byte into the register, write a byte to VRAM, read a byte from the status register & read a byte from VRAM. The procedure (in Basic) is per TABLE 2.

TABLE 2

Write to Register	OUT 129,D	: OUT 129,RO	
Write to VRAM	OUT 129,LB	: OUT 129,HO	: OUT 128,D
Read VDP	D = INP(13Ø) or INP(131)		
Read from VRAM	OUT 129,LB	: OUT 129,HA	: D = INP(132)

Where : D = byte written/read; RO = the register writen to ORed with 128; LB = least significant byte of address, equal to address MOD 256; HO = most significant byte of address ORed with 64; HA = most significant byte of address ANDed with 63; The most significant byte is equal to address\256, however remember that only 16k is used, so only 6 bits are used.

It's possible to manipulate the display using only these four routines. When writing or reading frm VRAM, you can use an auto incrementing function so that after each read/write, the VRAM address is increased by one, requiring only the last operation (OUT 128,D or D = INP(132)) to be performed once the address is set. Unfortunately reading the status register is not practical in Basic due to its slowness.



Next month I shall look at each mode in more detail and explain how the Spectravideo supports each.

Just to keep those fingers typing, I've included listing 1. This is a simple display editor for SCREEN Ø. Just position the cursor using the built in joystick and type in the desired characters. Key 1Ø can be used to switch from commands to characters. Key 5 will show

the current use of KEYS 1 - 9. The key prompts are turned off and most keys nulled to reduce erroneous input. To Save a screen, type in a file name and (after setting the cassette properly if using one) then press the SAVE key. Note that the cursor is turned ON during the save, so position it correctly before saving. This program uses a SPRITE$ bug to enable input without disrupting the screen.

SWITCH is a statement that appears to be unique to the Spectravideo. It allows the user to switch to another memory bank on the same computer, thereby enabling a form of timesharing. The catch is however that you need an extra memory bank as only one is supplied on an unexpanded SV. Assuming you DO have an extra bank, how do you use the command? If everything is setup correctly, an initial SWITCH command will dump you into the other bank with an initialisation message. This bank is identical to the original bank with one important difference, it doesn't have a program there! This has to be loaded in before anything can be done.

Assuming you've loaded the program in, what do you do now? This will depend on your needs. At some point in the program, you will want to swap over to the other bank and do some other task. Insert the SWITCH statement at the right location. When BASIC hits this, it will swap back to the other bank. Since each bank is seperate the other bank will still be in the same condition when you left it. This has several implications. If, on the bank swapped to, a program was running, it will resume where it was stopped, otherwise control passes back to the command level. Also, variables will not be the variables you've just been using, but the variables used on that bank. This means you must save needed information either on tape or disk or even VRAM and then recall that information when needed.

Nonetheless this is a powerful statement, effectively programs of 5Ø or 64k can be written with care. The relevant statements are:

SWITCH	Switch to alternate bank
SWITCH STOP	As per SWITCH, but force a Control-Stop thereafter
X = SWITCH	Returns Ø if original bank or -1 if on alternate bank

LISTING 1

```
1 WIDTH4Ø
2 MOTOROFF:SCREEN1,3:SCREENØ,Ø:CLEAR2ØØØ:ONERRORGOTO43:STOPON:ONSTOPGOSUB43:DIM
A$(32):KF=1:DV$="CAS":C$=CHR$(27):O$=C$+"x5":Z$=C$+"y5":S$=C$+"p":E$=C$+"q":GOSU
B24
3 JX$=INKEY$:PRINTJX$;
4 GOTO3
5 OP$="SAVE":GOSUB39:IFF$=""THENRETURN
6 GOSUB36:POKE-1532,25:IFDV$="CAS"ORLEFT$(F$,3)="CAS"THENGOSUB44:CSAVEF$,SELSEIF
DV$="DISK"THENSAVEF$,S
7 GOSUB37:RETURN
8 OP$="LOAD":GOSUB39:IFF$=""THENRETURN
9 GOSUB36:IFDV$="CAS"ORLEFT$(F$,3)="CAS"THENGOSUB44:CLOADF$ELSEIFDV$="DISK"THENL
OADF$
1Ø GOSUB37:RETURN
```

```
11 GOSUB36:LOCATEØ,Ø:PRINTC$"M";:GOSUB37:RETURN
12 GOSUB36:LOCATEØ,Ø:PRINTC$"L";:GOSUB37:RETURN
13 RETURN
14 GOSUB 3Ø:GOSUB32:ONKFGOSUB16,17,18:GOSUB42
15 GOSUB31:RETURN
16 LOCATEØ,1:PRINT"SAVE    LOAD    Scroll↑ Scroll↓ Menu";:LOCATEØ,3:PRINT"LNINPU
T CLS-END Scroll→ Scroll← SWAP";:RETURN
17 LOCATE1,1:PRINT"→       ←       ↑       ↓       Menu";:LOCATEØ,3:PRINT"REVERSE
 NORMAL  C.OFF   C.ON    SWAP";:RETURN
18 LOCATE1,1:PRINT"J       ·       S       ¤       Menu";:LOCATEØ,3:PRINT" z
  v.      k       ;       SWAP";:RETURN
19 LINEINPUTDD$:RETURN
2Ø PRINTC$"J";:RETURN
21 GOSUB36:GOSUB33:GOSUB35:OS=Ø:GOSUB38:GOSUB37:RETURN
22 GOSUB36:GOSUB34:GOSUB35:OS=39:GOSUB38:GOSUB37:RETURN
23 RETURN
24 KEY5,"":KEY1Ø,""
25 IFKF=1THENFORA=1TO4:KEYA,CHR$(211+A):NEXT:KEY6,S$:KEY7,E$:KEY8,O$:KEY9,Z$:KN=
2
26 IFKF=2THENFORA=1TO4:KEYA,CHR$(215+A):KEYA+5,CHR$(219+A):NEXT:KN=3
27 IFKF=3THENFORA=1TO4:KEYA,"":KEYA+5,"":NEXT:KN=1
28 KEYON:IFKN<>1THENKEYOFF
29 KEY(5)ON:KEY(1Ø)ON:ONKEYGOSUB5,8,11,12,14,19,2Ø,21,22,25:KF=KN:RETURN
3Ø GOSUB36:FORA=6TO1STEP-1:A$(A)=RIGHT$(SPRITE$(63+A),31)+LEFT$(A$(A+1),1):SPRIT
E$(63+A)=STRING$(32,Ø):NEXT:RETURN
31 FORA=1TO5:SPRITE$(63+A)=A$(A):NEXT:GOSUB37:RETURN
32 PRINTS$;:FORA=1TO5:FORB=ØTO1:LOCATE8*(A-1),B*2:PRINTUSING"Key##";A+B*5;:NEXTB
,A:PRINTE$;:RETURN
33 FORA=1TO31:A$(A)=SPRITE$(63+A):NEXT:RETURN
34 FORA=1TO3Ø:A$(A)=RIGHT$(SPRITE$(63+A),3Ø)+LEFT$(SPRITE$(A+64),2):NEXT:RETURN
35 FORA=1TO3Ø:SPRITE$(63+A)=A$(A):NEXT:RETURN
36 CX=PEEK(-1532):CY=PEEK(-1533):PRINTO$;:RETURN
37 POKE-1532,CX:POKE-1533,CY:PRINTZ$;:RETURN
38 FORA=ØTO92ØSTEP4Ø:VPOKEA+OS,Ø:NEXT:RETURN
39 GOSUB3Ø:LOCATE2,Ø:F$="":PRINT"FILENAME TO "OP$">";:LINEINPUTF$:GOSUB31
4Ø IFINSTR(F$,"cas:")=1THEN MID$(F$,1,4)="CAS:"
41 RETURN
42 MOTOROFF:IFINKEY$<>CHR$(13)GOTO42ELSERETURN
43 DEFUSR=&H3498:J=USR(Ø):SCREEN,1:LOCATE,,1:END
44 GOSUB3Ø:LOCATE1,Ø:PRINT"SET TAPE TO CORRECT LOCATION":PRINT" PRESS SPACEBAR W
HEN READY"
45 IFSTRIG(Ø)=ØGOTO45
46 LOCATE1,Ø:PRINT"PREPARE CASSETTE DECK        ":IFOP$="SAVE"THENPRINT" PRESS PL
AY & RECORD                "ELSEPRINT" PRESS PLAY                          "
47 PRINT" PRESS ENTER WHEN DONE":GOSUB42:GOSUB31:RETURN
```

SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP

PROGRAM OF THE MONTH

I have been waiting for an above average Hangman program and I'm pleased to say that Mr. Dennis French has submitted a real beaut. He has also sent in a educational maths program for children. I therefore am including both in this month's newsletter as they make a good team.

HANGMAN

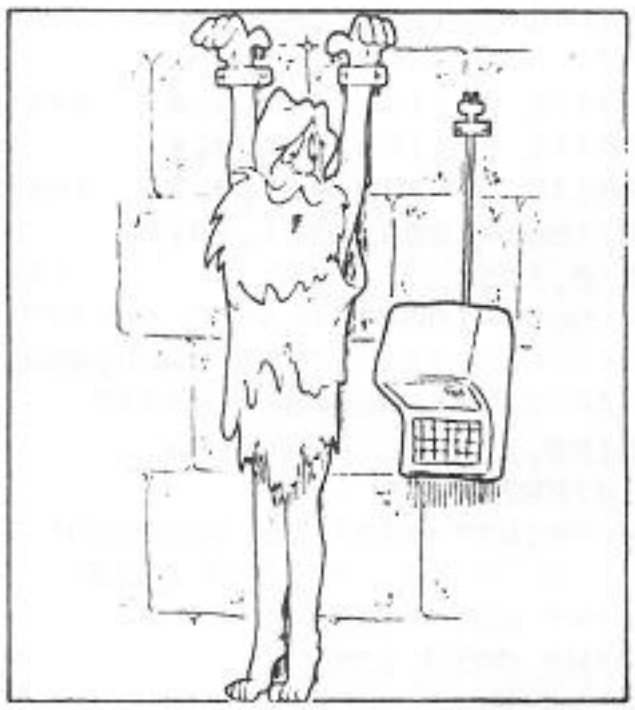
by D. French

```
10 REM ....HANGMAN PROGRAM....
15 REM
20 REM   written by D.French
25 REM
30 COLOR 1,10,10
40 SCREEN 1,2
50 FOR T=1 TO 16
60   READ S$,T$
70   P1$=P1$+CHR$(VAL("&B"+S$))
80   P2$=P2$+CHR$(VAL("&B"+T$))
90 NEXT
100 SPRITE$(0)=P1$+P2$
110 FOR T=1 TO 16
120   READ U$,V$
130   Q1$=Q1$+CHR$(VAL("&B"+U$))
140   Q2$=Q2$+CHR$(VAL("&B"+V$))
150 NEXT
160 SPRITE$(1)=Q1$+Q2$
170 FOR T=1 TO 16
180   READ W$,X$
190   R1$=R1$+CHR$(VAL("&B"+W$))
200   R2$=R2$+CHR$(VAL("&B"+X$))
210 NEXT
220 SPRITE$(2)=R1$+R2$
230 COLOR 1,10,10
240 LOCATE 75,40:PRINT"**** HANGMAN ****"
250 PRINT:PRINT:COLOR4:PRINTTAB(5)"This game improves a childs spelling"
260 PRINT
270 PRINTTAB(5)"skills and provides excitement at the"
280 PRINT
290 PRINTTAB(5)"same time. The player must complete"
300 PRINT
310 PRINTTAB(5)"the word before the 'HANGMAN' drawing"
320 PRINT
330 PRINTTAB(5)"is finished! Have fun!"
340 PRINT:PRINT:PRINT:COLOR6:PRINTTAB(10)"PRESS ANY KEY TO CONTINUE"
350 A$=INKEY$:IF A$="" GOTO 350
360 RESTORE
370 LET F=0:Z=0:B=0:F$="0"
380 F=97+INT(100*RND(-TIME))
390 FOR D=1 TO F:READ C$:NEXT
400 LET B$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
410 PLAY "T240S2M10000O4L3BL6AL4GABBL2BL4AAL2AL4BO5L4DL2D"
420 PLAY "O4L3BL6AL4GABBL2BL4AABAL1G"
430 CLS:LOCATE 0,0
440 COLOR1:PRINTTAB(14)"**** HANGMAN ****":PRINT
```

```
450 COLOR4:PRINT TAB(9)B$
460 LET A=LEN(C$)
470 LOCATE 0,165
480 IF A=4 THEN PRINT TAB(17)"_ _ _ _"
490 IF A=5 THEN PRINT TAB(17)"_ _ _ _ _"
500 IF A=6 THEN PRINT TAB(17)"_ _ _ _ _ _"
510 LOCATE 0,180
520 COLOR6:PRINTTAB(15)"CHOOSE A LETTER"
530 D$="0"
540 D$=INKEY$:IF D$="" GOTO 540
550 FOR C= 1 TO 26
560 IF MID$(F$,C,1)=D$ THEN GOTO 530
570 NEXT C
580 F$=F$+D$
590 LOCATE 102,160:COLOR6
600 IF LEFT$(C$,1)=D$ THEN PRINTD$:B=B+1
610 LOCATE 102,160
620 IF MID$(C$,2,1)=D$ THEN PRINTTAB(2)D$:B=B+1
630 LOCATE 102,160
640 IF MID$(C$,3,1)=D$ THEN PRINTTAB(4)D$:B=B+1
650 LOCATE 102,160
660 IF MID$(C$,4,1)=D$ THEN PRINTTAB(6)D$:B=B+1
670 LOCATE 102,160
680 IF MID$(C$,5,1)=D$ THEN PRINTTAB(8)D$:B=B+1
690 LOCATE 102,160
700 IF MID$(C$,6,1)=D$ THEN PRINTTAB(10)D$:B=B+1
710 LOCATE 48,16:COLOR10
720 FOR I=1 TO 26
730 IF MID$(B$,I,1)=D$ THEN PRINTTAB(I)"I"
740 NEXT I
750 IF B=A GOTO 1440
760 FOR J= 1 TO 6
770 IF MID$(C$,J,1)=D$ THEN GOTO 530
780 NEXT J
790 Z=Z+1
800 ON Z GOTO 810,840,860,880,900,920,940,980,1010,1050,1090,1130,1170,1210,1250
810 LINE(0,150)-(256,145),1,BF
820 GOTO 530
830 GOTO 530
840 LINE(98,145)-(200,140),13,BF
850 GOTO 530
860 LINE (98,140)-(180,135),13,BF
870 GOTO 530
880 LINE(98,135)-(160,130),13,BF
890 GOTO 530
900 LINE(105,130)-(107,50),13,BF
910 GOTO 530
920 LINE(105,50)-(145,52),13,BF
930 GOTO 530
940 LINE(105,62)-(114,52),13
950 LINE(105,61)-(113,52),13
960 LINE(105,63)-(115,52),13
970 GOTO 530
980 LINE(130,52)-(130,70),1
990 CIRCLE(130,75),5,1,,,2
1000 GOTO 530
```

```
1010 PUT SPRITE 0,(230,97),6,0
1020 PUT SPRITE 1,(230,113),4,1
1030 PUT SPRITE 2,(230,129),4,2
1040 GOTO 530
1050 PUT SPRITE 0,(200,97),6,0
1060 PUT SPRITE 1,(200,113),4,1
1070 PUT SPRITE 2,(200,129),4,2
1080 GOTO 530
1090 PUT SPRITE 0,(182,92),6,0
1100 PUT SPRITE 1,(182,108),4,1
1110 PUT SPRITE 2,(182,124),4,2
1120 GOTO 530
1130 PUT SPRITE 0,(162,87),6,0
1140 PUT SPRITE 1,(162,103),4,1
1150 PUT SPRITE 2,(162,119),4,2
1160 GOTO 530
1170 PUT SPRITE 0,(140,82),6,0
1180 PUT SPRITE 1,(140,98),4,1
1190 PUT SPRITE 2,(140,114),4,2
1200 GOTO 530
1210 PUT SPRITE 0,(123,64),6,0
1220 PUT SPRITE 1,(123,80),4,1
1230 PUT SPRITE 2,(123,96),4,2
1240 LINE(0,150)-(256,192),10,BF
1250 LOCATE 0,155
1260 PLAY"T150S8M10000O4L4CO3L6GL12GL4AGGBO4C"
1270 COLOR6:PRINTTAB(2)"Bad luck,you didn't find the secret word"
1280 PRINTTAB(3)"The word was!!!"
1290 LOCATE120,164
1300 COLOR 4:PRINT C$
1310 COLOR6:PRINTTAB(3)"Do you want to play again? Press Y or N"
1320 G$=INKEY$:IF G$="" GOTO 1320
1330 IF G$="Y" GOTO 360
1340 IF G$="N" GOTO 1350
1350 LOCATE 20,110
1360 SCREEN 0
1370 PLAY"T200S2M50000O4L2BL3BL8AL4AGL2D"
1380 PLAY"L8DEF#DL4EDL8DEF#DL4ED"
1390 PLAY"L2BL3BL8AL4AGL2DL8DEF#GL4AO5DO4L1G"
1400 COLOR 13:PRINT:PRINT:PRINT:PRINTTAB(4)"THANK YOU FOR PLAYING!!!"
1410 PRINT:PRINTTAB(4)"I HOPE YOU ENJOYED IT!!!"
1420 PRINT:PRINT:PRINT:PRINTTAB(4)"GOODBYE.... TILL NEXT TIME!"
1430 END
1440 PLAY "T150S2M5000L60SCL12EL6DL12FEGEL4C"
1450 PLAY "L4AL6DL12FL4EC"
1460 LOCATE 30,40:COLOR2
1470 PRINT"WELL DONE!"
1480 PRINT:PRINTTAB(3)"YOU SAVED ME!"
1490 PRINT:PRINT:PRINTTAB(3)"Do you want to"
1500 PRINTTAB(4)"play again?":PRINT
1510 FOR M=1 TO 6
1520 PUT SPRITE 0,(200,97),6,0
1530 PUT SPRITE 1,(200,113),4,1
1540 PUT SPRITE 2,(200,129),4,2
1550 FOR T=1 TO 200:NEXT T
1560 PUT SPRITE 0,(200,67),6,0
```

```
1570 PUT SPRITE 1,(200,83),4,1
1580 PUT SPRITE 2,(200,99),4,2
1590 FOR T=1 TO 200:NEXT T:NEXT M
1600 PUT SPRITE 0,(200,97),6,0
1610 PUT SPRITE 1,(200,113),4,1
1620 PUT SPRITE 2,(200,129),4,2
1630 COLOR1:PRINTTAB(5)"Y or N"
1640 GOTO 1320
1650 DATA 00000000,00000000
1660 DATA 00000000,00000000
1670 DATA 00000000,00000000
1680 DATA 00000000,00000000
1690 DATA 00000000,00000000
1700 DATA 00000000,00000000
1710 DATA 00000011,11000000
1720 DATA 00000100,00100000
1730 DATA 00001000,00010000
1740 DATA 00001010,01010000
1750 DATA 00001000,00010000
1760 DATA 00001000,00010000
1770 DATA 00001000,00010000
1780 DATA 00000100,00100000
1790 DATA 00000010,01000000
1800 DATA 00000010,01000000
1810 DATA 00111110,01111100
1820 DATA 01111111,11111110
1830 DATA 11111111,11111111
1840 DATA 11111111,11111111
1850 DATA 11111111,11111111
1860 DATA 11011111,11111011
1870 DATA 11011111,11111011
1880 DATA 11011111,11111011
1890 DATA 11011111,11111011
1900 DATA 11011111,11111011
1910 DATA 11011111,11111011
1920 DATA 11011111,11111011
1930 DATA 11011111,11111011
1940 DATA 11011111,11111011
1950 DATA 11011111,11111011
1960 DATA 11011111,11111011
1970 DATA 00011110,01111000
1980 DATA 00011110,01111000
1990 DATA 00011110,01111000
2000 DATA 00011110,01111000
2010 DATA 00011110,01111000
2020 DATA 00011110,01111000
2030 DATA 00011110,01111000
2040 DATA 00011110,01111000
2050 DATA 00011110,01111000
2060 DATA 00011110,01111000
2070 DATA 00011110,01111000
2080 DATA 00011110,01111000
2090 DATA 00011110,01111000
2100 DATA 00011110,01111000
2110 DATA 11111110,01111111
2120 DATA 11111110,01111111
4000 DATA ECHO,MONKEY,ORANGE,AWAKE,KING,TABLE,ACROSS
4010 DATA ENERGY,FATHER,PALACE,ARTIST,TODAY
4020 DATA BALLET,LEMON,UNDER,ADULT,FRIEND,PIGEON
```



L.E.

```
4030 DATA AFRAID,GARDEN,PLATE,BLACK,LUCKY,VISIT
4040 DATA ANCHOR,GOLDEN,QUICK,CHAIR,MAGIC,WATER
4050 DATA BACON,GLUE,RESCUE,CLOUD,MOTHER,WINDOW
4060 DATA BATTLE BASKET,HOME,RIFLE,CIRCUS,NAME,YELLOW
4070 DATA BRAVE,BUCKET,INSIDE,SCREAM,DANCE,NURSE,ZEBRA
4080 DATA BROWN,CANOE,JUNGLE,SIGNAL,DONKEY,OFTEN,YAWN
4090 DATA CAMERA,KITTEN,SMOKE,ENGINE,ENTER,FIRST,PARTY
4100 DATA FLOWER,PENCIL,GRAPE,PIRATE,CACTUS,KNIFE
4110 DATA SQUARE,CHERRY,KOALA,TAXI,GUESS,QUEEN
4120 DATA HORSE,HUNGRY,ROUND,CIRCLE,LIZARD,LUNCH,TRIP
4130 DATA COMB,MARCH,UGLY,DAISY,VOICE,SCHOOL
4140 DATA ISLAND,SILVER,JOKE,SMILE,JUMP,SPELL
4150 DATA MINUTE,WASH,DINGO,NEWS,WHITE
```

MOONMATHS

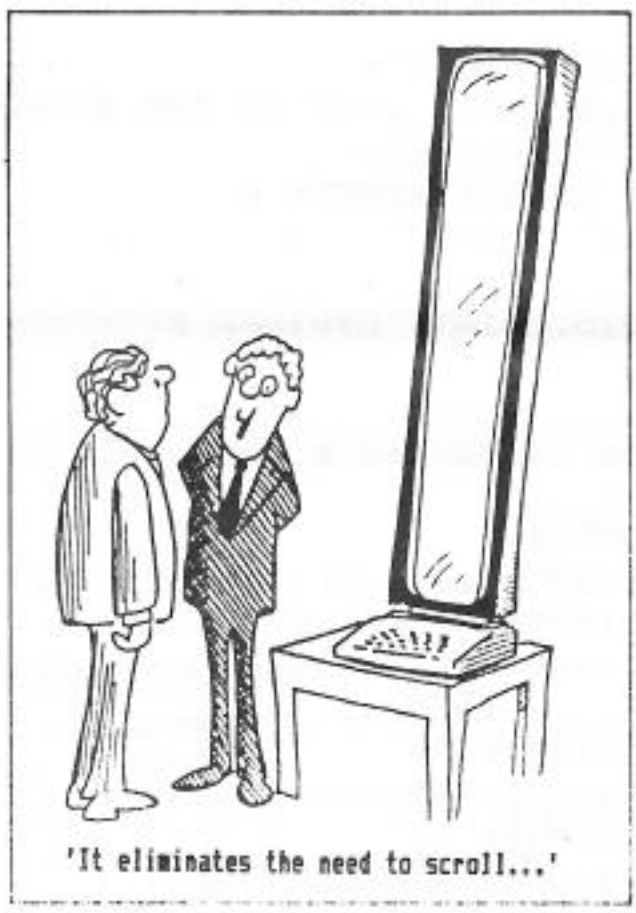
by D. French

```
10 REM    ### MATHS PROGRAM ###
20 REM
30 REM     Written by D.French
40 REM
50 REM           19/3/84
60 REM
70 REM
80 SCREEN,0
90 COLOR 4,10,10:SCREEN 2
100 PLAY"T125S11M20000L803GGABGBADGGABGGL4G-"
110 PLAY"L8GGAB04C03BAGG-DEG-L4GL8G"
120 PRINT:PRINT "** MOON **"
130 PRINT:PRINT"** MATHS **"
140 FOR T=1 TO800:NEXT T
150 COLOR13,1,1
160 SCREEN 1:GOSUB 1360
170 PRINTTAB(12)"*** MOON MATHS ***"
180 PRINT:PRINT:COLOR4:PRINTTAB(4)"HI THERE, KIDS!"
190 PRINT:COLOR6:PRINTTAB(4)"THE MOONMAN NEEDS YOUR HELP"
200 PRINTTAB(4)"TO LAUNCH HIS SPACESHIP."
210 PRINT:PRINT:COLOR4:PRINTTAB(4)"YOUR SCORE, OR COUNTDOWN, STARTS AT 5."
220 PRINT:COLOR6:PRINTTAB(4)"EVERY TIME YOU GET A CORRECT ANSWER,        ONE POIN
T IS TAKEN OFF THE COUNTDOWN."
230 PRINT:COLOR4:PRINTTAB(4)"IF YOU GET A WRONG ANSWER,THE               COUNTDOW
N STARTS OVER AGAIN!"
240 PRINT:COLOR6:PRINTTAB(4)"WHEN YOU GET 5 CORRECT ANSWERS, THE         SPACESHI
P WILL BLAST OFF!"
250 PRINT:PRINT:COLOR10:PRINTTAB(12)"PRESS C TO CONTINUE"
260 LET S$=INKEY$
270 IF S$="C"THENGOTO290
280 GOTO 260
290 GOSUB 990
300 LOCATE0,0
310 PRINTTAB(12)"*** MOONMATHS ***"
320 IF AA=0 GOTO 340
330 IF A=Z THEN GOSUB500ELSEGOSUB630
340 LET AA=1
350 LOCATE 5,60:COLOR1
360 PRINT:PRINTTAB(5)"Choose the type of sum you want...."
370 PRINTTAB(8)"For Addition type:       A"
380 PRINTTAB(8)"For Subtraction type:    S"
390 PRINTTAB(8)"For Multiplication type: M"
```

```
400 PRINTTAB(8)"For Division type:       D"
410 LET S$=INKEY$
420 IF S$="A" THEN GOSUB 1120:GOTO470
430 IF S$="S" THEN GOSUB 1180:GOTO470
440 IF S$="M" THEN GOSUB 1240:GOTO470
450 IF S$="D" THEN GOSUB 1300:GOTO470
460 GOTO 410
470 SCREEN 1,2
480 GOSUB 990
490 GOTO 310
500 COLOR 15:LOCATE 34,30:PRINT"THAT'S RIGHT!":LET C=C-1:LET XX=XX-20
510 PLAY O$+P$
520 IF C=0 GOTO 760
530 IF S$="A" THENGOTO570
540 IF S$="S" THENGOTO580
550 IF S$="M" THENGOTO590
560 IF S$="D" THENGOTO600
570 LOCATE130,30:PRINTX"+"Y"="Z:GOTO610
580 LOCATE130,30:PRINTX"-"Y"="Z:GOTO610
590 LOCATE130,30:PRINTX"*"Y"="Z:GOTO610
600 LOCATE118,30:PRINTX"DIVIDED BY"Y"="Z
610 COLOR10:LOCATE34,45:PRINT"YOUR SCORE IS "C"SECS TO BLASTOFF!!"
620 RETURN
630 COLOR15:LOCATE 24,30:PRINT"BAD LUCK";
640 LET C=5:LET XX=220
650 PLAY Q$
660 PUT SPRITE 0,(XX,176),10,0
670 IF S$="A" GOTO 710
680 IF S$="S" GOTO 720
690 IF S$="M" GOTO 730
700 IF S$="D" GOTO 740
710 PRINTX"+"Y"="Z"  NOT"A:GOTO610
720 PRINTX"-"Y"="Z"  NOT"A:GOTO610
730 PRINTX"*"Y"="Z"  NOT"A:GOTO610
740 PRINTX"DIVIDED BY"Y"="Z"  NOT"A:GOTO610
760 SOUND 7,7
770 PUT SPRITE 0,(0,170),4,0
780 FOR YY=176 TO 0 STEP-1.5
790 PUT SPRITE 1,(130,YY-7),1,1
800 PUT SPRITE 2,(130,YY),1,2
810 PUT SPRITE 3,(130,YY+8),6,3
820 FOR T=1TO3:NEXT T
830 NEXT YY
840 SOUND OFF
850 COLOR 4,1,1:SCREEN 1
860 PRINT:PRINT:PRINT:PRINTTAB(14)"CONGRATULATIONS!"
870 PRINT:PRINT:PRINT:COLOR6:PRINTTAB(5)"YOU ANSWERED 5 SUMS CORRECTLY"
880 PRINTTAB(5)"AND THE MOONMAN HAS BLASTED OFF!"
890 PRINT:PRINT:PRINT:COLOR4:PRINTTAB(5)"Do you want to play again? Y or N"
900 LET C=5:LET XX=220:LET AA=0
910 LET S$=INKEY$
920 IF S$="Y" THEN GOTO290
930 IF S$="N" THEN GOTO 950
940 GOTO 910
950 SCREEN 0
```

```
960 SCREEN,1
970 COLOR15,4,4
980 END
990 COLOR1,4,4:CLS
1000 FOR I=1TO30
1010 V=INT(256*RND(1)):W=INT(192*RND(1))
1020 PSET(V,W),15
1030 NEXT I
1040 LINE(0,192)-(256,185),2,BF
1050 SPRITE$(0)=B$:SPRITE$(1)=D$:SPRITE$(2)=F$:SPRITE$(3)=H$
1060 PUT SPRITE 0,(XX,176),10,0
1070 PUT SPRITE 1,(130,168),1,1
1080 PUT SPRITE 2,(130,176),1,2
1090 CIRCLE(60,140),10,6
1100 PAINT(60,140),6
1110 RETURN
1120 CLS:COLOR 15,6,6:SCREEN 0
1130 LET Z=0:LET X=5+INT(45*RND(-TIME)):LET Y=5+INT(45*RND(-TIME))
1140 Z=X+Y
1150 LOCATE 10,5:PRINTX"+"Y"="
1160 LOCATE 10,8:INPUT"WHAT IS THE ANSWER";A
1170 RETURN
1180 CLS:COLOR 15,6,6:SCREEN 0
1190 LET X=0:LET Z=5+INT(45*RND(-TIME)):LET Y=5+INT(45*RND(-TIME))
1200 X=Z+Y
1210 LOCATE 10,5:PRINTX"-"Y"="
1220 LOCATE 10,8:INPUT"WHAT IS THE ANSWER";A
1230 RETURN
1240 CLS:COLOR 15,6,6:SCREEN 0
1250 LET Z=0:LET X=2+INT(11*RND(-TIME)):LET Y=2+INT(11*RND(-TIME))
1260 Z=X*Y
1270 LOCATE 10,5:PRINTX"*"Y"="
1280 LOCATE 10,8:INPUT"WHAT IS THE ANSWER";A
1290 RETURN
1300 CLS:COLOR 15,6,6:SCREEN 0
1310 LET X=0:LET Z=2+INT(11*RND(-TIME)):LET Y=2+INT(11*RND(-TIME))
1320 X=Z*Y
1330 LOCATE 10,5:PRINTX"DIVIDED BY"Y"="
1340 LOCATE 10,8:INPUT"WHAT IS THE ANSWER";A
1350 RETURN
1360 REM Define variables & read data
1370 DEFINT A-Z
1380 LET S$=INKEY$
1390 LET O$="T15005L6CL12EL6DL12FEGEL4C"
1400 LET P$="T15005L4AL6DL12FL4EL8C"
1410 LET Q$="T10004L8CO3L12GL32GL8AL4GL8BO4L16C"
1420 RESTORE
1430 FOR I=1TO8:READ A$
1440 B$=B$+CHR$(VAL("&B"+A$)):NEXT I
1450 FOR I=1TO8:READ C$
1460 D$=D$+CHR$(VAL("&B"+C$)):NEXT I
1470 FOR I=1TO8:READ E$
1480 F$=F$+CHR$(VAL("&B"+E$)):NEXT I
1490 FOR I=1TO8:READ G$
1500 H$=H$+CHR$(VAL("&B"+G$)):NEXT I
1510 LET AA=0:LET C=5:XX=220
```

```
1520 RETURN
1530 DATA 00011000
1540 DATA 00011000
1550 DATA 01111110
1560 DATA 01011010
1570 DATA 01011010
1580 DATA 00111100
1590 DATA 00100100
1600 DATA 01100110
1610 DATA 00010000
1620 DATA 00010000
1630 DATA 00111000
1640 DATA 00111000
1650 DATA 00111000
1660 DATA 00111000
1670 DATA 00111000
1680 DATA 00111000
1690 DATA 00111000
1700 DATA 00111000
1710 DATA 10111010
1720 DATA 11111110
1730 DATA 11111110
1740 DATA 10111010
1750 DATA 10111010
1760 DATA 10000010
1770 DATA 00111000
1780 DATA 01111100
1790 DATA 01110100
1800 DATA 11111110
1810 DATA 01111100
1820 DATA 10111010
1830 DATA 00010000
1840 DATA 00000000
```



'It eliminates the need to scroll...'

SPECTRAVIDEO™
AUSTRALASIAN USERS GROUP

GRAPHIC SUBROUTINE

by G. Watson

Club members who are interested in 3-d displays on the SV318/328 may find a few interesting listings in the February 1984 issue of "Creative Computing". To make these listings, written for TRS-80 Mod I/II, suitable for the SV machines it is necessary to modify the subroutines and program it into the computer before entering the main program.

This is the subroutine for the SV 318/328.

```
    5 REM  "GRAPHIC"                (Put in NAME you wish here)
   15 SCREEN 1                      (Leave Line Numbers as follows)
   16 COLOR 15                      (or any other)
   17 XC=128 : YC=96                (Centres program on screen)
   18 XM=.8 : YM=.65                (Sealing factor - adjust within)
                                    (limits Ø.65 - 1.3)
 1ØØØ XP=X*XM+XC : XP=192-(Y*YM+YC)
 1Ø1Ø IF A$="MOVE" THEN XØ=XP : YØ=YP : RETURN
 1Ø2Ø LINE (XØ,YØ)-(XP,YP)
 1Ø3Ø XØ=XP : YØ=YP : RETURN
```

I hope this is of some interest to members. The displays include hyperbola (2D), stars (2D), sphere (3D), toroidal shapes (3D), distorted planes (3D), etc.

*** "I have been unable to find a Creative Computing for Feb. 1984 could someone send me a photocopy of the article please." (Ed.)

INTEGER DIVISION & MODULUS

by N. Booth

Two functions available on the SV that do not seem to be documented are integer division and modulus. Both functions can be very useful in programming board games or disc files.

Interger division is performed by using the backslash "\" instead of the normal division slash "/". The backslash is not well marked on the 328 keyboard: it looks like an accent mark and is just above the ENTER key. In integer division, the two numbers or variables being manipulated are treated as integers and the result is given as an integer. Thus 5.7\2.6 would give the result 2.

The modulus function is entered in the format X MOD Y and gives the remainder that is left when X is divided by Y. Again, the numbers or variaables are treated as integers.

Here is an example :

```
1Ø L=97
2Ø X=L MOD 1Ø
3Ø Y=L\1Ø
```

This will give the result X=7 and Y=9.

As these are integer functions, an error wil result if they are used on numbers or variables with values greater than 32767 or less than -32766.

PATTERNS

by T. Colverd

The program Pattern should allow the user to get a painting into the National Gallery. I'm sure I've seen some of the results there.

```
10 CLS:SCREEN1,1
20 G=255:H=191
30 A=RND(-TIME)*G:C=RND(-TIME)*G:D=RND(-TIME)*H:E=RND(-TIME)*H:F=RND(-TIME)*H
40 L=RND(-TIME):CO=RND(-TIME)*16:CL=RND(-TIME)*16:COL=RND(-TIME)*16
50 COLOR,CO,CL
60 LINE(A,D)-(B,F),CO,B
80 COLORCL,CO
90 LINE(C,F)-(A,D),CO:LINE-(B,E),CL:LINE-(C,F),CO
100 IFP=2THEN210
120 A=RND(-TIME)*A:D=RND(-TIME)*D
130 CIRCLE(A,E),D,COL,CO/3
140 IFP>0THEN PAINT(A,E),P+1,COL
160 IFP=1THEN180
170 IFP=3THEN180ELSE200
180 LINE(B,F)-(C,D),CO,BF:GOTO250
200 LINE(B,F)-(C,D),CL,BF:GOTO230
210 PLAY"A"
220 LINE(B,F)-(C,D),CO,BF:GOTO260
230 FORG=1TOA STEP20:SOUND7,G:NEXT:GOTO260
240 PCOPY COL TO CL
250 FORG=CTO1STEP-20:SOUND1,G:NEXT
260 GOTO20
```

The following are some good commands found in our Disc BASIC that are not documented particularly well.

RUN COMMAND

What is so exciting about this simple command you ask. Well if you have a disc system the RUN command can be used as follows.

Normally to excecute a program that is on disc you would type :

```
LOAD "1:FRED"
RUN
```

Well the following works as well :

```
RUN "1:FRED"
```

Saves a bit of typing and every little bit helps. But the main advantage of using RUN in the above form is inside a program. That is, you can excecute another program from the current program. An example is :

```
10 INPUT "GIVE ME A NUMBER";A
20 ON A GOTO 30,40,50,60
30 RUN "1:ONE"
40 RUN "1:FRED"
50 RUN "1:SPECTRON"
60 RUN "1:MARK
```

SPECTRAVIDEO™

AUSTRALASIAN USERS GROUP

KILL COMMAND

Another command some people don't know about is the KILL command. It is used to remove a program you no longer want on a disc.

For example if you have the program FRED on a disc and you don't want it any more all you do is type :

```
KILL "1:FRED"
```

The program is gone forever and you can reuse the disc space for another program.

NAME COMMAND

So you saved a program on disc and you called it FRID but it was supposed to be called FRED. All you have to do is type :

```
NAME "1:FRID" AS "1:FRED"
```

Thus FRID becomes FRED. Use the FILES command to verify the change has taken place.

COPY COMMAND ??

HELP!! anyone know how to use this command. It is part of our reserved words and I just don't seem to get the syntax right.