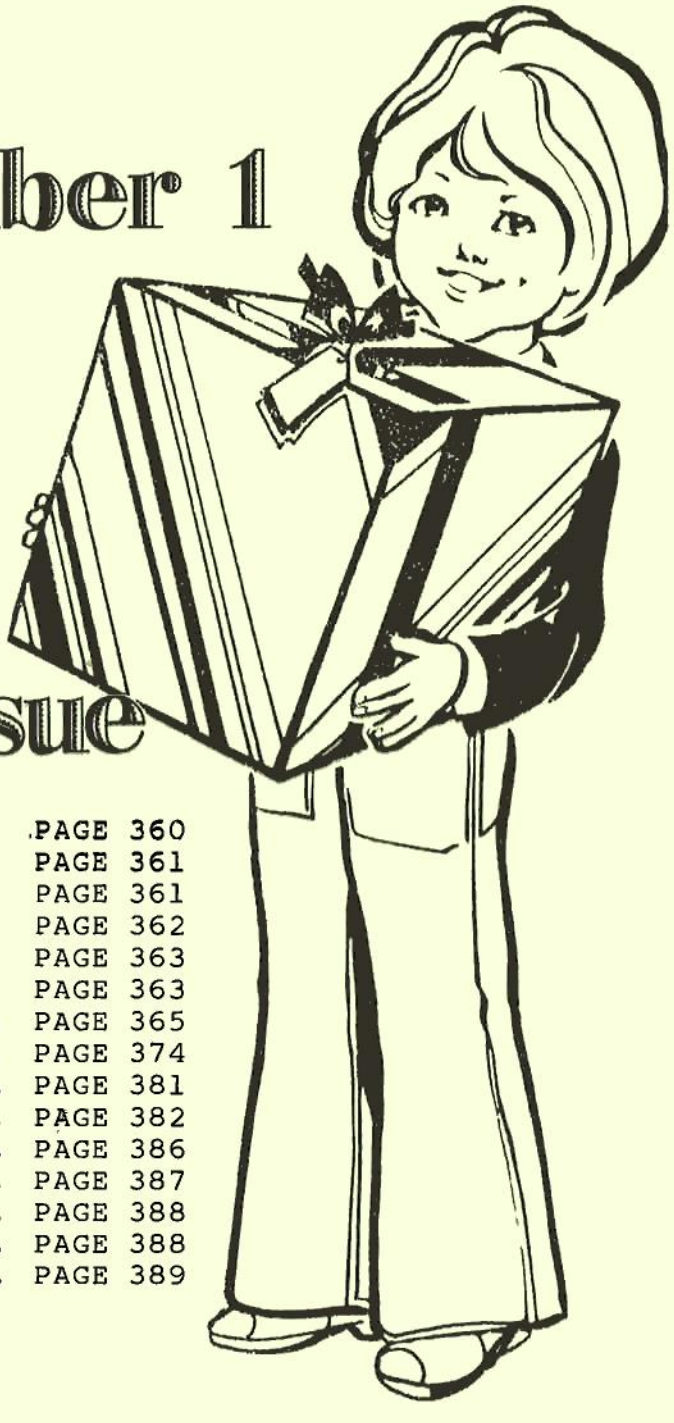
MEMOPAD

SPECIAL 2ND BIRTHDAY EDITION

Vol 011

Number 1



In This Issue

MEMOPAD

Paul Parry
0993
MCL 78691
Unit 24
Avenue 1
Station Lane
Indust Est
Witney

Edited by Tim Marstian

Artwork by Anthony `Joe 90´

Executive Editor
Keith Hook

MEMOPAD IS PUBLISHED BY SYNTAXsoft FOR THE MEMOTECH USER GROUP
UNIT B20, THE NORTHBRIDGE CENTRE, ELM STREET,
BURNLEY, LANCS. BB10 1PD
TELEPHONE: (0282) 38596

AVAILABLE BY SUBSCRIPTION ONLY.

memopad
Memotech Computer User Group
Number 1
Volume 011

Editorial

Hopefully, we have cured all the gremlins and this birthday edition should reach you on time. Believe me, it really hasn't been our fault... Syntax moved offices, local holidays interfered with the printing, and then to top it all the local G.P.O went on a go slow !

Keith is to put on his travelling boots again next week and will be visiting the Middle East and will end his trip by landing in Iran and staying there as the guest of a few influential people who are interested in computing and computers.

M.C.L is now set to attack the market place. Syntaxsoft has formed a new company - Centaur Computer Services Limited - and has taken two new directors, **Marc Butler [ex marketing director of Imagine] and Ken Simmonds [Boss of Kerian U.K.].** The new company will be attacking the British market place with a view to setting up a nationwide dealer distribution for the M.C.L MK2 computer.

The new computer is not really a new computer. It is the MTX 512 upgraded to 256K of memory. This will sell at less than £100 which should fill a big hole in today's market place - there is only the Spectrum 128 Plus which sells at around £149, and we all know which is the better computer.

O.K. I can hear you all asking what's happened to the **new** computer. Well this will go ahead at some time in the near future, but we need to sell the MCL 256K machine because the price is right. I can promise you that when the new system does surface, all of you will be able to upgrade you existing systems to the new configuration. This is a promise that MCL made - they will not forget the loyal users of the original machine.

Geoff's Video Wall has caused quite a stir in the market place and he is, at resent, rushing around the country visiting the people who have requested a demonstration. This one development means that MCL's future is secured.

We have now had a sample of the new 3 1/2" disc drive and I can tell you it is brilliant ! Once you have used the silicon disc, you will never want to go back to your old configuration. Syntax are now busy downloading software to the new format and a list will appear in the next Memopad of which titles are available.

Now, what has Keith Hook been up to ? Syntaxsoft had a very good business trip to the PCW Show and met quite a few American software houses. It now looks definite that Syntax will be appointed the major conversion house for the American Agency that handles such companies as Sega. The deal, of course, has built in provisions for the titles to be made available on the MCL machine. Other companies that have agreed, in principle, are Cascade, Elite, US Gold, Hewson Consultants, Design Design and Mikrogen.

Keith is a great believer in 'Never say die', and I am sure that all of you will agree, he hasn't ! I can remember, just before I took over as Editor, that there was a great despondency within the membership, and we all honestly believed that the major software houses would never recognise the computer but Keith kept on plodding and his philosophy is paying dividends. So, I would like to end this birthday edition by thanking our founder member for his hard work over the past four years in keeping the MTX flag flying. Thank you !

Black is not only beautiful .... its anodised !★ TM

GENPAT HIT LIST....

arcade

1	COBRA	XAVIERSINE
2	TOURNAMENT SNOOKER	MAGNIFICENT 7
3	ROLLA BEARING	MEGASTAR
4	HIGHWAY ENCOUNTER	VORTEX
5	MINER DICK	XAVIERSINE
6	DOWNSTREAM DANGER	MEGASTAR
7	FELIX IN THE FACTORY	MICROPOWER
8	SUPA CODER	SYNTAXSOFT
9	USER BASIC	SYNTAXSOFT
10	OBLITERATION ZONE	MEGASTAR

adventure

1	LORDS OF TIME	LEVEL 9
2	RETURN TO EDEN	LEVEL 9
3	MURDER AT THE MANOR	SENTIENT
4	ADVENTURE QUEST	LEVEL 9
5	THE KEYS TO TIME	SENTIENT

educational

1	HELIMATHS	SENTIENT
2	WORDS & PICTURES	CONTINENTAL
3	FIRST LETTERS	CONTINENTAL
4	SPELLICOPTER	SENTIENT
5	MATHS 1	CONTINENTAL

High Scores

Can you do better ?

ASTRO-PAC	213,430	Michael Hunt
KNUCKLES	1,147,350	T. Eriksson
CHAMBEROIDS	Completed 4 mins	T. Eriksson
MISSION ALPHATRON	175,340	Matthew Moss
TAPEWORM	175,980	Richard Franks
TOADO	356,414	John Quinn
POT HOLE PETE	106,630	Richard Franks
MAXIMA	1,479,710	S. Olander
STAR COMMAND	140,430	Ian Nichols
PHAID	26,000	Sally Street
OBLOIDS	62,400+	M. Hurley
KILOPEDE	82,253	Richard Nash
3D TACHYON FIGHTER	12,500	C. Walker
CONTINENTAL RAIDERS	106,240	Sean Haverly
BLOBBO	140,283	Elizabeth Mahon
QUANTUM	14	Ian Cartwright
QOGO 2	205,000	R. Siddall
MINEFIELD	2,100	C. Walker
FLUMMOX	251,530	C. Walker
TURBO	18,610	Michael Hunt
FATHOMS DEEP	3,450	Matthew Moss
AGROVATOR	675,000+	P. Howard
FIREHOUSE FREDDIE	20,620	T.Eriksson
QOGO	43,980	T. Eriksson
ARCADIANS	25,900	Adrian Joynson
MISSILE COMMAND	27,500	Adrian Joynson
LITTLE DEVILS	34,320	Leslie Rhodes
FELIX IN THE FACTORY	14,740	Peter Crighton
HUNCHY	8,457	John Quinn
SON OF PETE	17,233	T. Eriksson
HAWKWARS	25,800	Gordon Hurd
ESCAPE FROM ZARCOS	76 Items	G Hill + D Stockall
SALTY SAM	40,642	Andrew Johnson
MISSION OMEGA	9,340	R.Harvey
ICEBURG	17,431	Alan Dobson
SNOWBALL	1080	Victor Stepney
EMERALD ISLE	760/1000	Victor Stepney
SUPERBIKE	23.04secs	P. Clark
ROLLA BEARING	27,000+	Victor Stepney
DR. FRANKIE	65,435	J Graham
TARGET ZONE	17,470	D J Chamberlain
MINER DICK	121,540	R. Siddall
JUMPING JACK	26,120	Andrew Miller
SURFACE SCANNER	72,050	T. Eriksson
CAVES OF ORB	495/500	Victor Stepney
SEPULCRI SCELERATI	7,925	Andrew Miller GAME COMPLETED
S.M.G.	30,100+	Phillip Goodey
RETURN TO EDEN	672	Andy Crick
QUAZZIA	26,860	Andrew Miller
OBLITERATION ZONE	32,570	Alan Dobson
ASTROMILON	142,342	D J Chamberlain
CRYSTAL	32,425	Gordon Hurd GAME COMPLETED
DRIVE THE LEE-5	12,987	Victor Stepney
HIGHWAY ENCOUNTER	34,420	Gordon Hurd
KARATE KING	4,570	G Hill
DOWNSTREAM DANGER	8,976	G Hill

Mike Mash has completed Qogo 2 and has quoted the final message - "At last, you have found the Qogo diamonds"

SOFTWARE REVIEW

COBRA

This is a game which has been available quite a long time and hasn't had any sort of recongnition. The game itself is an absolute must for any enthusiast of fast moving, obstacle avoiding games.

The object of the game is to control the 'COBRA' around the screen avoiding the rocks, which are scattered on the screen at random, with more rocks being added as the time increases. Also be very careful of the walls which surround you as running into them will have the same effect as running into a rock, instant death. The 'COBRA' itself is rather partial to fruit, which is also scattered randomly at time intervals onto the screen. To gain the valuable points which the fruits possess, just move the 'COBRA' over the fruit.

However beware of the time, because when the clock reaches 300, the game stops and all the fruit that hasn't been eaten turns into rocks making it essential that the fruit is eaten as soon as it appears on the screen. The game will continue when all of the uneaten fruit has changed into rocks, and carry on as before until the clock reaches 500. It will then progress onto the next frame, (as I have been unable to complete level 1, I cannot explain what happens on the next levels).

The graphics on this game are quite good, with the fruit being especially good. The fruit that has to be eaten consists of cherries, strawberries and apples. The sound on this game is reasonable.

The movement of the 'COBRA' is quite good, but beware if you retrace your moves too quickly, (i.e by moving over your own body) you will lose a live. The game is also joystick compatible.

Marks out of Ten

GRAPHICS = 8
COLOUR = 8
SOUND = 7
CONTROL = 8
VALUE = 8

Rubbish
Knock

Another exciting game from Xaviersine, fun for all the family.

BOOK REVIEWS

The recent revival of interest in CP/M as an operating system which actually works, is well documented and doesn't need half a megabyte of system memory to run a 48K program seems to have produced a rash of books in bookshops and public libraries. They include a good number of the sort which academics loathe, but which attract all the rest of us who cannot resist 200 pages of free software of the kind that doesn't turn up so frequently since 'Creative Computing' went out of business and 'Personal Computer World' and 'Byte' became terribly serious and specialised.

Examples of the CP/M utilities fairly readily available are:

FILTER.COM:

strips control characters from a Wordstar or NewWord File (sometimes known as'UnWordStar').

SLIST.COM:

displays a file 1 screen line at a time, without control characters

PRINTER.COM:

initialise printer for user-designated fonts

PRINT.COM

adjusts any serial printer to your computer and determine output format and number of copies and as they say, 'many more'!

CP/M AND THE PERSONAL COMPUTER
Thomas A. Dwyer and Margot Critchfield
Micro Books, Addison-Wesley 1983

This takes you very gently through the whole of CP/M from running a commercial program to rewriting the rewriting the BIOS, with full examples and several free utilities on the way. Expensive, but I got it from the City Library.

CP/M SOLUTIONS
Ken Barbier
Prentice-Hall 1985

This is also a bit pricey at #13.95 for 140 pages, but is a kind of Ultimate Solution to problems with the Intel assembler. The book assumes a basic knowledge of CP/M (it is not a beginner's book) and sets out a form of Structured Programming in 8080 assembly language, using modules set up as separate files, each with the .LIB extension so that they can be tidied up and concatenated into an .ASM file with ED.COM,PIP.COM or your favourite word processor. Everything is very fully explained and full DIY instructions are given. The example programs are useful, too, including two designed to use the otherwise infamous 'ED' and 'PIP LST:' facilities as a simple (and free) word-processor quite adequate for shortish documents without too may formatting problems. The author deserves support for his advice - unusual these days from writers of books on computing - to "if you know a solution, pass it on!"

A Simple Utility

Dear Editor,

Please feel free to use this simple Utility for inclusion in MEMOPAD if it is of any interest. I do think the inclusion of simple programs for non-disc owners is a worthwhile topic! We can't all run discs....

Yours sincerely,

L. Shaw

```
0 REM **  This program is a very simple
1 REM   letterwriting utility, which can
2 REM   be built upon or modified to
3 REM   give different type faces etc.
4 REM ** Two screen lines are equal to
5 REM   one line of print..PRACTICE!
10 CLS : INK 1 : PAPER 15
20 REM ***********************************
30 REM ********  LETTER.TXT  ************
40 REM ***********************************
50 CLS : PRINT "Please Do not use commas!"
60 INPUT "Do you want letter quality print ?"     ENTER   Y or N  > ";PRIN$: IF PRIN$="Y" THEN GOTO  70
ELSE GOTO 80
70 LPRINT CHR$(15): REM "Letter size"
80 INPUT "Enter Your Reference ";R$
90 INPUT  "House Name or Number ";N$: INPUT "Street Name ";S$: INPUT "Town ";T$: INPUT  "County  ";C$:
INPUT "Postcode ";P$
100 INPUT "Date ";D$
110  LPRINT "Ref:"+R$: LPRINT ,,,,,,,N$+" "+S$: LPRINT ,,,,,,,T$: LPRINT ,,,,,,,C$:  LPRINT  ,,,,,,,P$:
LPRINT ,,,,,,,D$: CLS
120 REM ***********************************
130 REM ****       INPUT KEYS        ****
140 REM ***********************************
150 DIM A$(160): LET A$=INKEY$
160 INPUT A$
170 LPRINT LEFT$(A$,77)
180 GOTO 160
190 STOP
```

Forth by Charles Yates Part 1

Forth has always been my favourite language and when my new MTX 512 arrived I was shocked at the price of the established compiler. I decided to write my own as an experiment with the new machine; the results of that experiment are published.

The program features:

Over 150 Forth words.
Fast integer number handling; ideal for games.
Nearly 40K free for programming + 16K accessible for data storage.
Most BASIC graphics commands (sprites etc.) plus extra.
A full screen editor, 'a la BBC'.
A "RAM disc" (see later).

If the volume of code seems daunting fear not; I will supply a copy of the program on a high quality cassette for the price of #3.50. Send your cheques/P.O. to CHARLES YATES
KELDROSEED
SANDWICK
STROMNESS
ORKNEY KW16 3HY

If you are not accustomed to the language I would advise this program as a superb introduction.

To type in the program first create a line of assembly language and fill it with DS 250s until a value past 5A00 has been reached.

i.e. Assem 10 (RETURN)

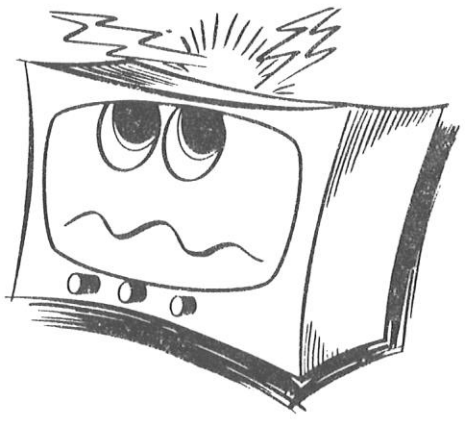
(RETURN)

4007 DS 250 (RETURN)

(ENT/CLS) (RETURN)

T (RETURN)

(RETURN) until >#5A00



Exit the assembler and type in listing 2 followed by

VS 5: GOTO 20 (NB Do Not Run)
and type in the code one line at a time followed by the check sum. The program checks for errors and forces you to retype the last line if one occurs.

Once finished save the program in case of any errors and RUN and type:

BYE

A BK message should occur if it is correct to there. Re-saving the program at this point is essential and it should take less time than the last piece.

The program is now ready to use.

If any error has occured re-load the program and enter the PANEL.

Using the Program

If you are not accustomed to the language the rest of this article will try to show its main features by comparing with BASIC. I will therefore assume a working knowledge of BASIC.

On running the program type VLIST. The word DUP should apppear on the screen. By pressing the space bar a succession of words are displayed.

These words are known as the resident dictionary and from your knowledge of BASIC some of them should be recognisable and others should be easy to have a guess at.

Most commands are concerned with manipulating the data on top of the "Stack".

As an example, . (the fullstop) prints the value on top of the stack on the screen, it is similar to BASICs.

PRINT n (Where n is any number.)

Try: 50.

The number 50 should be displayed on the screen.

1 2 3 . . .

The number should then be re-displayed in reverse order to the way you typed them in (i.e. 3,2,1). From those examples, it is obvious that line numbers do not come into Forth and that everything appears to be the opposite way round to BASIC.

Quite a few commands are concerned with stack manipulation and a few of the main ones are listed below with their effect on the stack;

	BEFORE	AFTER	
DUP	n1	n1 n1	
SWAP	n1 n2	n2 n1	
OVER	n1 n2	n1 n2 n1	
ROT	n1 n2 n3	n2 n3 n1	
2DUP	n1 n2	n1 n2 n1 n2	
MIN	n1 n2	min	
MAX	n1 n2	max	
>R	n1		TO) Return Stack
R>		n1	FROM)
R@		n1	Copy top of Return Stack

Listed below are a few Basic programs with their Forth equivalents:

```
B  PRINT 50
F  50 .
B  PRINT 50 + 100
F  100 50 + .
B  PRINT  (100 + 25 - 75)/10
F  100 25 + 75 - 10 / .
B  POKE 50000, 50
F  50 50000 C!
B  PRINT PEEK (50000)
F  50000 C@ .
B  DOKE 50000,500       (No MTX Basic equivalent)
F  500 50000 !
```

```
B  PRINT DEEK (50000)
F  50000 @ .
B  LET A=50
F  IF A undefined :  50 VARIABLE A.  else 50 A !
B  PRINT A
F  A @ .
B  LET A = A + 1 or  LET  A = A - 50
F  1 A + !           -50 A + !
B  IF A > 50 THEN PRINT "YES" ELSE PRINT "NO"
F  A @ 50 > IF ."YES" ELSE ."NO" THEN
B  VS 4: FOR I  = 0 TO 100: FOR J = 0 TO 100: PLOT I,J: NEXT J,I
F  4 VS 100 0 DO I 100 0 DO DUP I PLOT  LOOP DROP LOOP
   NB all graphics commands are equivalents;
      B: PLOT X,Y ; MVSPR p,n,d; etc
      F: X Y PLOT ; P N D MVSPR  ETC
B  PRINT CHR$(65)
F  65 EMIT
B  100 IF INKEY$<> " " THEN GOTO 100
F  BEGIN INKEY 32 = UNTIL
```

There are some 160 commands to play around with so I am not going to be able explain all of them here; however don't be afraid to experiment with them. I will try to explain the most obscure ones in the remainder of this article.

C@R, C!R, !R, @R - These commands are used to pass data to the normally unaccessible 16K and parameters are the same as those of the POKE and PEEK commands illistrated earlier (on the 512 the address must be greater than 32768).

Number handling in this program are strictly 16 bit signed (from -32768 to 32768) or unsigned integers (from 0 to 65535). Hexadecimal numbers are also catered for, most commands prefixed or suffixed with a # deal with them. Integer number handling is limited but far faster than floating point and therefore ideal for writing games.

The colon (:) is used for defining new words whilst the semi-colon (;) ends a definition. Try:

```
: TEST 500 0 DO I . LOOP ;
```

On typing VLIST the new word (test) will be shown at the end of the list. Now try:

```
: SQUARE DUP * ;
```

This command shows how to pass parameters into your new command.

```
e.g. 50 SQUARE
```

The FORGET command makes the compiler forget all programs after the name specified i.e.

```
FORGET TEST
```

gets rid of both TEST and SQUARE and anything else you may have defined after it.

The FIND command places the start address of the program specified onto the stack.

EXECUTE does a call to the address at TOS (similar to USR in Basic).

FORMAT tells the computer to jump to the command specified upon running (currently START).

P! and P@ are Forths output and input commands i.e. byte port P! or port P@.

2P! for use with video ram.

BR-OFF and BR-ON turn the break key off and on respectively.

BREAK does nothing by itself but try this

```
BR-OFF FIND BREAK #FD52 !
```

INT-ON causes interrupt to jump to address at top of stack every 20ms (Keep it simple!).
INT-OFF turns it off.

PEMIT strobes character at top of stack to printer (same as EMIT).

I mentioned a RAM Disc facility at the beginning of this article. This feature relies on the extra 16K and to initialise it on Power up, type:

```
0  RDELETE
```

As an example try: VLIST RSAVE CLS
and then: 0 RLOAD
will then redisplay the previous screen. About 16 screens or more, depending on length, can be stored. These programs were designed for use with the screen editor (cursor and home keys, try it).

Well, that's about all I have room for. I hope you enjoy using the program and you are perfectly welcome to write to me for information. ★

```
0 CODE

4007        LD (£43A7),SP
400B        LD (£43A5),SP
400F        CALL £57B9
4012        RST 10
4013        SUB H
4014        LD C,L
4015        LD D,H
4016        LD E,B
4017        JR NZ,£405F
4019        LD L,A
401A        LD (HL),D
401B        LD (HL),H
401C        LD L,B
401D        JR NZ,£4047
401F        LD B,E
4020        ADD HL,HL
4021        JR NZ,£4054
4023        ADD HL,BP
4024        JR C,£405C
4026        DEC C
4027        LD A,(BC)
4028        RST 10
4029        ADD A,(HL)
402A        DEC DE
402B        LD E,D
402C        DEC B
402D        DEC C
402E        LD A,(BC)
402F        LD A,£3A
4031        XOR A
4032        LD B,E
4033        BIT 4,A
4035        JR Z,£403C
4037        RST 10
4038        ADD A,E
4039        EX AF,AF'
403A        LD E,A
403B        LD A,£21
403D        SUB £43
403F        LD BC,£009B
4042        CALL £447D
4045        LD A,(£43AF)
4048        BIT 4,A
404A        JR NZ,£4052
404C        LD HL,£4661
404F        LD (£43B6),HL
4052        CALL £40E2
4055        JR NZ,£4062
4057        LD A,(£43AF)
405A        BIT 4,A
405C        JP Z,£4296
405F        JP £4028
4062        LD HL,(£4479)
4065        LD A,(HL)
4066        CP £3A
4068        JP Z,£42F1
406B        CP £3B
406D        JR NZ,£4074
406F        LD HL,£43B0
4072        RES 0,(HL)
4074        CALL £4142
4077        JP Z,£4203
407A        CALL £4100
407D        JP NZ,£41E4
4080        CALL £49AE
4083        CALL £40D2
4086        RST 10
4087        SUB B
4088        CCF
4089        JR NZ,£40D4
408B        LD L,(HL)
408C        HALT
408D        LD H,C
408E        LD L,H
408F        LD L,C
4090        LD H,H
4091        JR NZ,£40D9
```

```
4093  LD L,(HL)
4094  LD (HL),H
4095  LD (HL),D
4096  LD A,C
4097  LD L,£C3
4099  JR Z,£40DB
409B  CALL £49AE
409E  CALL £40D2
40A1  RST 10
40A2  SUB D
40A3  CCF
40A4  JR NZ,£40E7
40A6  LD L,H
40A7  LD (HL),D
40A8  LD H,L
40A9  LD H,C
40AA  LD H,H
40AB  LD A,C
40AC  JR NZ,£40F2
40AE  LD H,L
40AF  LD H,(HL)
40B0  LD L,C
40B1  LD L,(HL)
40B2  LD H,L
40B3  LD H,H
40B4  LD L,£C3
40B6  JR Z,£40F8
40B8  CALL £49AE
40BB  CALL £40D2
40BE  RST 10
40BF  ADC A,A
40C0  LD C,C
40C1  LD L,(HL)
40C2  LD H,E
40C3  LD L,A
40C4  LD (HL),D
40C5  LD (HL),D
40C6  LD H,L
40C7  LD H,E
40C8  LD (HL),H
40C9  JR NZ,£4117
40CB  LD L,A
40CC  LD L,A
40CD  LD (HL),B
40CE  LD L,£C3
40D0  RET PE
40D1  LD B,D
40D2  LD HL,(£4479)
40D5  LD A,(HL)
40D6  CP £00
40D8  RET Z
40D9  CP £20
40DB  RET Z
40DC  CALL £44D8
40DF  INC HL
40E0  JR £40D5
40E2  LD HL,(£4479)
40E5  LD A,(HL)
40E6  CP £20
40E8  JR NZ,£40ED
40EA  INC HL
40EB  JR £40E5
40ED  CP £00
40EF  LD (£4479),HL
40F2  RET
40F3  LD A,(HL)
40F4  CP £00
40F6  RET Z
40F7  CP £29
40F9  INC HL
40FA  LD (£4479),HL
40FD  RET Z
40FE  JR £40F3
4100  LD HL,£4729
4103  LD DE,(£4479)
4107  LD A,(HL)
4108  CP £00
410A  RET Z
410B  PUSH HL
410C  LD A,(HL)
410D  CP £04
410F  JR C,£4126
4111  LD B,A
4112  LD A,(DE)
4113  CP B
4114  JR NZ,£411A
4116  INC HL
4117  INC DE
4118  JR £410C
411A  LD A,(HL)
411B  CP £04
411D  INC HL
411E  CCF
411F  JR C,£411A
4121  RST 8
4122  EX DE,HL
4123  POP DE
4124  JR £4103
4126  BIT 0,A
4128  JR NZ,£412F
412A  CALL £413B
412D  JR NZ,£411A
412F  LD (£4479),DE
4133  LD A,(HL)
4134  INC HL
4135  INC HL
4136  INC HL
4137  POP BC
4138  CP £FF
413A  RET
413B  LD A,(DE)
413C  CP £00
413E  RET Z
413F  CP £20
4141  RET
4142  LD DE,£0000
4145  LD HL,£43AF
4148  RES 1,(HL)
414A  LD HL,(£4479)
414D  LD A,(HL)
414E  CP £2D
4150  JR NZ,£415B
4152  LD HL,£43AF
4155  SET 1,(HL)
4157  LD HL,(£4479)
415A  INC HL
415B  LD A,(HL)
415C  SUB £30
415E  RET C
415F  CP £09
4161  JR Z,£4164
4163  RET NC
4164  PUSH HL
4165  PUSH AF
4166  LD BC,£000A
4169  CALL £4197
416C  POP AF
416D  LD E,A
416E  LD D,£00
4170  ADD HL,DE
4171  EX DE,HL
4172  POP HL
4173  INC HL
4174  LD A,(HL)
4175  CP £20
4177  JR Z,£417F
4179  CP £00
417B  JR Z,£417F
417D  JR £415B
417F  LD A,(£43AF)
4182  BIT 1,A
4184  LD (£4479),HL
4187  RET Z
4188  EX DE,HL
4189  CALL £418F
418C  EX DE,HL
418D  CP A
418E  RET
418F  LD DE,£0000
4192  EX DE,HL
4193  XOR A
4194  SBC HL,DE
4196  RET
4197  LD HL,£0000
419A  LD A,£10
419C  AND A
419D  RR B
419F  RR C
41A1  JR NC,£41A4
41A3  ADD HL,DE
41A4  RL E
41A6  RL D
41A8  DEC A
41A9  JR NZ,£419D
41AB  RET
41AC  LD A,D
41AD  CPL
41AE  LD D,A
41AF  LD A,E
41B0  CPL
```

```
41B1  LD E,A
41B2  INC DE
41B3  PUSH HL
41B4  XOR A
41B5  LD H,A
41B6  LD L,A
41B7  EX (SP),HL
41B8  LD A,D
41B9  OR E
41BA  JR NZ,£41BE
41BC  POP HL
41BD  RET
41BE  LD B,£11
41C0  RL L
41C2  RL H
41C4  JR C,£41D6
41C6  DJNZ £41C0
41C8  JR £41DE
41CA  EX (SP),HL
41CB  PUSH HL
41CC  ADD HL,DE
41CD  JR NC,£41D0
41CF  EX (SP),HL
41D0  POP HL
41D1  EX (SP),HL
41D2  RL L
41D4  RL H
41D6  EX (SP),HL
41D7  RL L
41D9  RL H
41DB  EX (SP),HL
41DC  DJNZ £41CA
41DE  POP DE
41DF  SRA D
41E1  RR E
41E3  RET
41E4  BIT 1,A
41E6  JR NZ,£41EF
41E8  EX DE,HL
41E9  CALL £4284
41EC  JP £4052
41EF  LD A,(£43B0)
41F2  BIT 0,A
41F4  JR Z,£41FD
41F6  EX DE,HL
41F7  CALL £4284
41FA  JP £4052
41FD  LD DE,£4052
4200  PUSH DE
4201  PUSH HL
4202  RET
4203  CALL £4630
4206  JP £4052
4209  LD DE,£4217
420C  CALL £4284
420F  RET
4210  LD DE,£4229
4213  CALL £4284
4216  RET
```

```
4217  LD (£43A7),SP
421B  LD SP,(£43A3)
421F  POP HL
4220  LD (£43A3),SP
4224  LD SP,(£43A7)
4228  RET
4229  LD (£43A7),SP
422D  LD SP,(£43A3)
4231  PUSH HL
4232  LD (£43A3),SP
4236  LD SP,(£43A7)
423A  RET
423B  CALL £4217
423E  LD A,L
423F  CP £00
4241  RET
4242  POP BC
4243  CALL £4217
4246  EX DE,HL
4247  CALL £4217
424A  PUSH HL
424B  PUSH DE
424C  PUSH BC
424D  RET
424E  POP IX
4250  POP HL
4251  POP DE
4252  AND A
4253  SBC HL,DE
4255  JR NC,£425E
4257  ADD HL,DE
4258  INC HL
4259  PUSH DE
425A  PUSH HL
425B  PUSH IX
425D  RET
425E  XOR A
425F  JR £425B
4261  CALL £4217
4264  POP IX
4266  POP BC
4267  POP DE
4268  BIT 7,H
426A  JR NZ,£4277
426C  ADD HL,BC
426D  AND A
426E  SBC HL,DE
4270  JR NC,£425E
4272  ADD HL,DE
4273  PUSH DE
4274  PUSH HL
4275  JR £425B
4277  ADD HL,BC
4278  AND A
4279  SBC HL,DE
427B  JR Z,£425E
427D  JR C,£425E
427F  ADD HL,DE
```

```
4280  PUSH DE
4281  PUSH HL
4282  JR £425B
4284  LD HL,(£43B6)
4287  LD (HL),£CD
4289  INC HL
428A  CALL £428E
428D  RET
428E  LD (HL),E
428F  INC HL
4290  LD (HL),D
4291  INC HL
4292  LD (£43B6),HL
4295  RET
4296  LD HL,(£43B6)
4299  LD (HL),£C9
429B  INC HL
429C  LD (HL),£00
429E  LD A,(£43B0)
42A1  RES 0,A
42A3  LD (£43B0),A
42A6  LD (£43A7),SP
42AA  PUSH HL
42AB  LD HL,(£43A3)
42AE  LD DE,(£43A9)
42B2  XOR A
42B3  SBC HL,DE
42B5  POP HL
42B6  JP NZ,£40B8
42B9  LD A,(£43AF)
42BC  BIT 4,A
42BE  JR NZ,£42C9
42C0  LD HL,£42DF
42C3  PUSH HL
42C4  LD HL,£4661
42C7  PUSH HL
42C8  RET
42C9  LD (£43B4),HL
42CC  EX DE,HL
42CD  LD HL,(£43AB)
42D0  LD (HL),E
42D1  INC HL
42D2  LD (HL),D
42D3  LD HL,£43AF
42D6  RES 4,(HL)
42D8  LD HL,£4661
42DB  LD (£43B6),HL
42DE  RET
42DF  LD HL,(£43A3)
42E2  LD (£43A9),HL
42E5  JP £4028
42E8  LD HL,(£43A9)
42EB  LD (£43A3),HL
42EE  JP £4028
42F1  INC HL
42F2  LD (£4479),HL
42F5  CALL £4322
42F8  LD HL,£43AF
42FB  SET 4,(HL)
```

```
42FD  LD SP,(£43A5)
4301  JP £4052
4304  CALL £40E2
4307  JR Z,£4314
4309  PUSH HL
430A  CALL £4100
430D  POP HL
430E  LD (£4479),HL
4311  JR NZ,£431B
4313  RET
4314  LD SP,(£43A5)
4318  JP £4080
431B  LD SP,(£43A5)
431F  JP £409B
4322  CALL £4304
4325  EX DE,HL
4326  LD HL,(£43B4)
4329  LD A,(DE)
432A  CP £00
432C  JR Z,£4337
432E  CP £20
4330  JR Z,£4337
4332  LD (HL),A
4333  INC HL
4334  INC DE
4335  JR £4329
4337  LD (HL),£00
4339  INC HL
433A  LD (£43AB),HL
433D  INC HL
433E  INC HL
433F  LD (£43B6),HL
4342  LD (£4479),DE
4346  RET
4347  BIT 7,H
4349  JR Z,£4352
434B  XOR A
434C  CALL £418F
434F  RST 10
4350  ADD A,C
4351  DEC L
4352  PUSH HL
4353  LD HL,£43AF
4356  SET 3,(HL)
4358  RES 3,(HL)
435A  POP HL
435B  LD DE,£2710
435E  CALL £4380
4361  LD DE,£03E8
4364  CALL £4380
4367  LD DE,£0064
436A  CALL £4380
436D  LD E,£0A
436F  CALL £4380
4372  LD A,(£43AF)
4375  SET 3,A
4377  LD (£43AF),A
437A  LD E,£01
437C  CALL £4380
```



```
437F  RET
4380  XOR A
4381  SBC HL,DE
4383  INC A
4384  JR NC,£4381
4386  CCF
4387  ADC HL,DE
4389  DEC A
438A  CP £00
438C  JR NZ,£4395
438E  LD A,(£43AF)
4391  BIT 3,A
4393  RET Z
4394  XOR A
4395  ADD A,£30
4397  CALL £44B8
439A  LD A,(£43AF)
439D  SET 3,A
439F  LD (£43AF),A
43A2  RET
43A3  INC B
43A4  JP M,£FD2E
43A7  LD L,£FD
43A9  INC B
43AA  JP M,£5A0E
43AD  LD (BC),A
43AE  LD BC,£0068
43B1  NOP
43B2  LD BC,£0A20
43B5  LD E,D
43B6  LD H,C
43B7  LD B,(HL)
43B8  RST 10
43B9  ADC A,E
43BA  RRCA
43BB  LD BC,£FF81
43BE  RST 38
43BF  RST 38
43C0  RST 38
43C1  RST 38
43C2  RST 38
43C3  RST 38
43C4  RST 38
43C5  RET
43C6  NOP
43C7  NOP
43C8  NOP
43C9  NOP
43CA  NOP
43CB  NOP
43CC  NOP
43CD  NOP
43CE  NOP
43CF  NOP
43D0  NOP
43D1  NOP
43D2  NOP
43D3  NOP
43D4  NOP
```

```
43D5  NOP
43D6  LD B,D
43D7  LD E,C
43D8  LD B,L
43D9  NOP
43DA  NOP
43DB  NOP
43DC  NOP
43DD  NOP
43DE  NOP
43DF  NOP
43E0  NOP
43E1  NOP
43E2  NOP
43E3  NOP
43E4  NOP
43E5  NOP
43E6  NOP
43E7  NOP
43E8  NOP
43E9  NOP
43EA  NOP
43EB  NOP
43EC  NOP
43ED  NOP
43EE  NOP
43EF  NOP
43F0  NOP
43F1  NOP
43F2  NOP
43F3  NOP
43F4  NOP
43F5  NOP
43F6  NOP
43F7  NOP
43F8  NOP
43F9  NOP
43FA  NOP
43FB  NOP
43FC  NOP
43FD  NOP
43FE  NOP
43FF  NOP
4400  NOP
4401  NOP
4402  NOP
4403  NOP
4404  NOP
4405  NOP
4406  NOP
4407  NOP
4408  NOP
4409  NOP
440A  NOP
440B  NOP
440C  NOP
440D  NOP
440E  NOP
440F  NOP
```

```
4410  NOP
4411  NOP
4412  NOP
4413  NOP
4414  NOP
4415  NOP
4416  NOP
4417  NOP
4418  NOP
4419  NOP
441A  NOP
441B  NOP
441C  NOP
441D  NOP
441E  NOP
441F  NOP
4420  NOP
4421  NOP
4422  NOP
4423  NOP
4424  NOP
4425  NOP
4426  NOP
4427  NOP
4428  NOP
4429  NOP
442A  NOP
442B  NOP
442C  NOP
442D  NOP
442E  NOP
442F  NOP
4430  NOP
4431  NOP
4432  NOP
4433  NOP
4434  NOP
4435  NOP
4436  NOP
4437  NOP
4438  NOP
4439  NOP
443A  NOP
443B  NOP
443C  NOP
443D  NOP
443E  NOP
443F  NOP
4440  NOP
4441  NOP
4442  NOP
4443  NOP
4444  NOP
4445  NOP
4446  NOP
4447  NOP
4448  NOP
4449  NOP
444A  NOP
444B  NOP
444C  NOP
444D  NOP
444E  NOP
444F  NOP
4450  NOP
4451  NOP
4452  NOP
4453  NOP
4454  NOP
4455  NOP
4456  NOP
4457  NOP
4458  NOP
4459  NOP
445A  NOP
445B  NOP
445C  NOP
445D  NOP
445E  NOP
445F  NOP
4460  NOP
4461  NOP
4462  NOP
4463  NOP
4464  NOP
4465  NOP
4466  NOP
4467  NOP
4468  NOP
4469  NOP
446A  NOP
446B  NOP
446C  NOP
446D  NOP
446E  NOP
446F  NOP
4470  NOP
4471  NOP
4472  NOP
4473  NOP
4474  NOP
4475  NOP
4476  LD B,B
4477  LD E,C
4478  NOP
4479  EXX
447A  LD B,E
447B  NOP
447C  NOP
447D  LD (£4479),HL
4480  CALL £44AD
4483  CALL £45AA
4486  CALL £44BF
4489  CALL £44CA
448C  CP £7F
448E  JR Z,£44E6
4490  CP £0D
4492  JP Z,£45D2
4495  CP £20
4497  JR C,£450B
4499  CALL £44D7
449C  JR Z,£4486
449E  PUSH HL
449F  CALL £44B8
44A2  LD HL,(£4479)
44A5  LD (HL),A
44A6  INC HL
44A7  LD (£4479),HL
44AA  POP HL
44AB  JR £4486
44AD  PUSH HL
44AE  PUSH BC
44AF  LD (HL),£00
44B1  INC HL
44B2  DEC C
44B3  JR NZ,£44AF
44B5  POP BC
44B6  POP HL
44B7  RET
44B8  PUSH AF
44B9  RST 28
44BA  XOR H
44BB  POP AF
44BC  RET
44BD  INC C
44BE  RET
44BF  LD A,£5F
44C1  CALL £44B8
44C4  LD A,£08
44C6  CALL £44B8
44C9  RET
44CA  PUSH BC
44CB  CALL £0079
44CE  LD B,A
44CF  LD A,(£FD7D)
44D2  CP B
44D3  JR NZ,£44CB
44D5  POP BC
44D6  RET
44D7  PUSH HL
44D8  PUSH BC
44D9  DEC BC
44DA  ADD HL,BC
44DB  EX DE,HL
44DC  LD HL,(£4479)
44DF  SCF
44E0  CCF
44E1  SBC HL,DE
44E3  POP BC
44E4  POP HL
44E5  RET
44E6  PUSH HL
44E7  PUSH BC
44E8  LD BC,£0001
44EB  CALL £44D7
44EE  JR Z,£4506
44F0  LD HL,(£4479)
44F3  DEC HL
44F4  LD (HL),£00
44F6  LD (£4479),HL
44F9  LD A,£20
```

```
44FB  CALL £44B8
44FE  LD A,£08
4500  CALL £44B8
4503  CALL £44B8
4506  POP BC
4507  POP HL
4508  JP £4486
450B  CP £08
450D  JP Z,£4523
4510  CP £0B
4512  JR Z,£4523
4514  CP £0A
4516  JR Z,£4523
4518  CP £19
451A  JR Z,£4523
451C  CP £1A
451E  JR Z,£4595
4520  JP £4486
4523  PUSH HL
4524  PUSH BC
4525  LD HL,(£FFA9)
4528  PUSH HL
4529  PUSH AF
452A  CALL £455D
452D  LD HL,(£43AD)
4530  LD (£FFA9),HL
4533  LD A,(£43B3)
4536  CALL £44B8
4539  LD (£FFA9),HL
453C  POP AF
453D  CALL £44B8
4540  LD HL,(£FFA9)
4543  CALL £456E
4546  LD (£43B3),A
4549  LD (£FFA9),HL
454C  LD A,£81
454E  CALL £44B8
4551  LD (£43AD),HL
4554  POP HL
4555  LD (£FFA9),HL
4558  POP BC
4559  POP HL
455A  JP £4486
455D  LD A,(£43B1)
4560  CP £00
4562  RET NZ
4563  INC A
4564  LD (£43B1),A
4567  LD HL,(£FFA9)
456A  LD (£43AD),HL
456D  RET
456E  PUSH HL
456F  EX DE,HL
4570  LD HL,£1C01
4573  LD A,E
4574  PUSH DE
4575  LD DE,£0001
4578  CALL £458E
457B  POP DE
457C  LD A,D
```

```
457D  LD DE,£0028
4580  CALL £458E
4583  LD A,L
4584  OUT (£02),A
4586  LD A,H
4587  OUT (£02),A
4589  NOP
458A  IN A,(£01)
458C  POP HL
458D  RET
458E  CP £00
4590  RET Z
4591  ADD HL,DE
4592  DEC A
4593  JR £458E
4595  PUSH HL
4596  LD A,(£43B3)
4599  LD HL,(£4479)
459C  LD (HL),A
459D  INC HL
459E  LD (£4479),HL
45A1  CALL £44B8
45A4  LD A,£19
45A6  POP HL
45A7  JP £450B
45AA  PUSH HL
45AB  PUSH BC
45AC  LD B,£03
45AE  LD IX,(£FFA9)
45B2  PUSH IX
45B4  CALL £49AE
45B7  POP DE
45B8  LD HL,(£FFA9)
45BB  PUSH HL
45BC  XOR A
45BD  SBC HL,DE
45BF  JR NZ,£45C8
45C1  PUSH IX
45C3  POP DE
45C4  DEC D
45C5  PUSH DE
45C6  POP IX
45C8  DJNZ £45B4
45CA  LD (£FFA9),IX
45CE  POP BC
45CF  POP BC
45D0  POP HL
45D1  RET
45D2  LD (£4479),HL
45D5  LD A,£20
45D7  CALL £44B8
45DA  LD A,(£43B1)
45DD  CP £00
45DF  RET Z
45E0  XOR A
45E1  LD (£43B1),A
45E4  LD A,(£43B3)
45E7  LD HL,(£FFA9)
45EA  PUSH HL
```

```
45EB  LD HL,(£43AD)
45EE  LD (£FFA9),HL
45F1  CALL £44B8
45F4  POP HL
45F5  LD (£FFA9),HL
45F8  LD A,£20
45FA  LD (£43B3),A
45FD  RET
45FE  CALL £0079
4601  LD B,A
4602  LD A,(£FD7D)
4605  LD C,A
4606  LD A,B
4607  CP £00
4609  RET Z
460A  LD A,C
460B  RET
460C  CALL £4217
460F  EX DE,HL
4610  CALL £4217
4613  RET
4614  PUSH HL
4615  LD HL,(£43A3)
4618  ADD HL,DE
4619  LD (£43A3),HL
461C  EX DE,HL
461D  POP HL
461E  LD A,(DE)
461F  LD (HL),A
4620  INC HL
4621  LD (HL),£C9
4623  DEC DE
4624  DEC DE
4625  DJNZ £461E
4627  LD HL,(£43A3)
462A  INC HL
462B  INC HL
462C  LD (£43A3),HL
462F  RET
4630  LD HL,(£43B6)
4633  LD (HL),£21
4635  INC HL
4636  CALL £428E
4639  CALL £4210
463C  RET
463D  BIT 7,H
463F  RET Z
4640  PUSH DE
4641  PUSH HL
4642  LD HL,£43AF
4645  BIT 1,(HL)
4647  JR Z,£4651
4649  RES 1,(HL)
464B  POP HL
464C  CALL £418F
464F  POP DE
4650  RET
4651  SET 1,(HL)
4653  JR £464B
```

Disc Mania by Eric Roy Part 5

This utility for the SDX disc will allow you to set / get up to 11 attribute bits per file. At present the SDX only uses one (T1' - Read Only) which is set / reset using the STAT "filename.typ", RO/RW command.

The attribute bits are stored in bit 7 of the filename and type letters that go to make up a valid filename. This can be seen after setting a file to Read Only then displaying the directory, the RO bit (T1') of the file will contain nothing or garbage, this is due to the DIR command failing to strip of bit 7 before printing the filename.

The following listing provides two new USER commands which allow you to set / get the file attributes. The DIR command has also been re-written to strip off the attribute bits before printing, also if bit (T2' - System File) is set then that filename will not be displayed.

The new USER commands are.....

USER DIR <filename optional>

USER SFA,<value>,<"filename"> Sets the attribute bits in the file specified. Value is a number or variable containing the attributes that are to be set.

Attribute bits / Value for SFA

F	(F1')	32768
I	(F2')	16384
L	(F3')	8192
E	(F4')	4096
N	(F5')	2048
A	(F6')	1024
M	(F7')	512
E	(F8')	256
T	(T1')	128 - Read Only
Y	(T2')	64 - System File
P	(T3')	32 - Reserved

If you wish to set bits F2' and T1' then the command would be USER SFA,16384+128, "filename.typ". A value of 0 (zero) will reset all the attribute bits.

The four bits (F1' - F4') are available for use and could be set up as a security code for a file, or signal some other event. Bits (F5' - F8') are reserved for future use by CP/M but can be used if required. Bit (T1') is the bit set / reset by the STAT command. Bit (T2') if set will prevent the new DIR command from displaying the filename, the STAT command will display it regardless of the state of bit (T2'). Bit (T3') is always 0 after a file has been written to.

USER GFA, <"filename.typ"> On return from this command the basic variable GFA will be set to.....

GFA = 0 if file has no attributes
GFA = 1 if the file does not exist else GFA will be set to the value of the attribute bits.

Passing values back from assembly to basic variables is not supported by the MTX, however the GFA command does this by simulating the rom's LET command. The section of code at GFAOK does this, a brief explanation follows.....

On entry HL contains the attribute values for GFA, this is then put in BC. The call to #0DD0 converts the value in BC to ascii terminating with a #FF byte. On return from #0DD0 DE is pointing to the start of the ascii data, HL is loaded with the address of the buffer and the ascii + #FF bytes are moved into this buffer. DE is then loaded with start of the variable name and equal token (#D4) and a call to the rom LET routine (#29DA) will create the GFA variable.

The source code written on the Edasm macro assembler is well documented and the various subroutines used should not need explaining. If you don't have Edasm then the best way of entering the code is to use the panels Display>F618 command and then enter the hex data bytes directly into memory. When finished enter WRITE "SGATT.HEX",63000,424 to save object code to disc.

If you used Edasm to enter the source, use the LOAD directive to load object code to #B000 then WRITE "SGATT.HEX",45056,424.

Enter READ "SGATT.HX",63000 : LET A = USR(63000) to reload code and initialise. ★

```
001E                ;**************************************
                    ;   ---SET / GET FILE ATTRIBUTES---
                    ;
                    ;       MEMOTECH  SDX  250K  DISC
                    ;
                    ;         by Eric Roy Jan.1986
                    ;**************************************
                    ;
                            ORG     0F618H          ; = 63000
                            LOAD    0B000H          ; = 45056
                    ;
                    ; Equates
                    ;---------
                    ;
FAD2                PAGE    EQU     0FAD2H
0CAB                PRINTX  EQU     00CABH
29DA                ROMLET  EQU     029DAH
F5B0                BDOS    EQU     0F5B0H
D8E8                FCB5    EQU     0D8E8H
E680                DMA     EQU     0E680H
E8C0                DIRBUF  EQU     0E8C0H
EBE1                DISCROM EQU     0EBE1H
2F5B                AMBFILE EQU     02F5BH
2F2F                UFILEN  EQU     02F2FH
0011                SEARCHF EQU     011H
0012                SEARCHN EQU     012H
001A                SETDMA  EQU     01AH
001E                SETFAT  EQU     01EH
                    ;
                    ; Vector SDX USER command to NEWCOM
                    ;-----------------------------------
                    ; USER commands jump to #F5B3, the first 3 byte
                    ; instruction  LD A,(PAGE) is changed to JP NEWCOM
                    ;
                    ; LET x = USR(63000) from basic to init.
                    ;
                    VECSDX
F618 2124F6                 LD      HL,NEWCOM
F61B 22B4F5                 LD      (0F5B4H),HL
F61E 3EC3                   LD      A,0C3H
F620 32B3F5                 LD      (0F5B3H),A
F623 C9                     RET                     ; To basic
```

```
                        ;
                        ; Entry point for new SDX commands
                        ;-----------------------------------
                        ;
                        NEWCOM
F624 ED53AEF7                   LD      (SNAME),DE      ; Save position in basic
F628 1A                         LD      A,(DE)          ; of USER name
F629 FE53                       CP      "S"             ; Test first letter and
F62B CAD1F6                     JP      Z,SFACOM        ; jump to command if match.
F62E FE47                       CP      "G"
F630 CA17F7                     JP      Z,GFACOM
F633 FE44                       CP      "D"
F635 CA4BF6                     JP      Z,DIRCOM
                        ;
                        ; Here if invalid name
                        ;----------------------
                        ;
                        NAMERR
F638 ED5BAEF7                   LD      DE,(SNAME)      ; DE -> start of name
F63C 3AD2FA                     LD      A,(PAGE)        ; A = current page / rom
F63F C3B6F5                     JP      0F5B6H          ; Jump into SDX user
                        ;
                        ; Test syntax of user name
                        ;---------------------------
                        ;
                        UNAME
F642 46                         LD      B,(HL)          ; B = length
                        UN1
F643 23                         INC     HL
F644 13                         INC     DE
F645 1A                         LD      A,(DE)          ; Get letter from basic line
F646 BE                         CP      (HL)            ; Match ?
F647 C0                         RET     NZ              ; No
F648 10F9                       DJNZ    UN1
F64A C9                         RET
                        ;
                        ; Entry point for new DIR command
                        ;----------------------------------
                        ; The only differance in this DIR command is
                        ; that a test is made of the System bit 7.
                        ; If it is set (1) then the filename is not
                        ; displayed.
                        ;
                        DIRCOM
F64B 21A7F7                     LD      HL,DIRN
F64E CD42F6                     CALL    UNAME           ; Check syntax
F651 C238F6                     JP      NZ,NAMERR       ; Failed
                        ;
F654 DD21E8D8                   LD      IX,FCB5         ; IX -> FCB channel 5
F658 13                         INC     DE
F659 1A                         LD      A,(DE)
F65A FEFF                       CP      0FFH
F65C 2009                       JR      NZ,DIRFN        ; Filename after DIR
                        ;
                        ; Here if DIR <RET> ( no filename )
                        ;
F65E E5                         PUSH    HL              ; Sets up filename in FCB as
F65F CDE1EB                     CALL    DISCROM         ; ????????.??? so that all
F662 5B2F                       DW      AMBFILE         ; filenames in directory will
F664 E1                         POP     HL              ; be found.
F665 1803                       JR      DIRFN1
```

```
                        ;
                        ; Here if DIR "filename"
                        ;
                        DIRFN
F667 CD92F7                     CALL    GETUF           ; Get filename from basic line
                        DIRFN1                          ; and put it in FCB
F66A 1B                         DEC     DE
F66B D5                         PUSH    DE
F66C 1180E6                     LD      DE,DMA
F66F 0E1A                       LD      C,SETDMA
F671 CDB0F5                     CALL    BDOS            ; DMA at #E680
                        ;
F674 11E8D8                     LD      DE,FCB5         ; Search directory for filenames
F677 0E11                       LD      C,SEARCHF       ; On return A = #FF if no names
F679 CDB0F5                     CALL    BDOS            ; found, else A = 0,1,2 or 3
F67C FEFF                       CP      0FFH            ; = position in DMA.
F67E 2846                       JR      Z,NOFILE
                        DIRFN2
F680 CDAFF6                     CALL    FADR            ; HL -> filename
F683 E5                         PUSH    HL
F684 010900                     LD      BC,09
F687 09                         ADD     HL,BC           ; HL -> System (T1') byte
F688 CB7E                       BIT     7,(HL)          ; System bit set ?
F68A E1                         POP     HL
F68B 2014                       JR      NZ,DIRFN3       ; Yes don't display this filenam
                        ;
F68D 0608                       LD      B,08
F68F CDBCF6                     CALL    DISPLAY         ; Print filename
F692 3E2E                       LD      A,"."
F694 CDAB0C                     CALL    PRINTX
F697 0603                       LD      B,03            ; Print file type
F699 CDBCF6                     CALL    DISPLAY
F69C 3E20                       LD      A," "
F69E CDAB0C                     CALL    PRINTX
                        DIRFN3
F6A1 0E12                       LD      C,SEARCHN       ; Search directory for
F6A3 CDB0F5                     CALL    BDOS            ; next filename.
F6A6 FEFF                       CP      0FFH            ; On return A = #FF if end
F6A8 20D6                       JR      NZ,DIRFN2       ; of directory reached.
                        CRLF
F6AA D7                         RST     010H
F6AB 2D0A                       DB      02DH,00AH       ; Print CR LF
F6AD D1                         POP     DE              ; Position in basic
F6AE C9                         RET                     ; Return to basic
                        ;
                        ; Find filename in DMA.
                        ;-----------------------
                        ; Up to 4 directory entries are in the DMA
                        ; each 32 bytes long. Filename starts at
                        ; directory +1
                        ;
                        ; Entry          A = 0,1,2 or 3
                        ;
                        ; Exit           HL -> start of filename
                        ;
                        FADR
F6AF 2181E6                     LD      HL,DMA+1        ; HL -> first filename
F6B2 112000                     LD      DE,020H         ; DE = length of directory entry
F6B5 A7                         AND     A
F6B6 C8                         RET     Z               ; If 0 then HL correct
```

```
F6B7 47                 LD      B,A
                FADR1
F6B8 19                 ADD     HL,DE           ; Calc. addr. for HL
F6B9 10FD               DJNZ    FADR1
F6BB C9                 RET
                ;
                ; Display filename.type on screen
                ;-----------------------------------
                ; Bit 7 stripped off so that file type letters
                ; are displayed correctly.
                ;
                DISPLAY
F6BC 7E                 LD      A,(HL)          ; Get letter
F6BD E67F               AND     07FH            ; Strip off bit 7
F6BF CDAB0C             CALL    PRINTX          ; Print it
F6C2 23                 INC     HL
F6C3 10F7               DJNZ    DISPLAY
F6C5 C9                 RET
                ;
                ; Here if no filename match or not found.
                ;-----------------------------------------
                ;
                NOFILE
F6C6 D7                 RST     010H
F6C7 874E6F20
F6CB 46696C65           DB      087H,"No File"
F6CF 18D9               JR      CRLF
                ;
                ; Entry point for new SFA,<value>,<"filename"> command
                ;-----------------------------------------------------
                ;
                SFACOM
F6D1 21AAF7             LD      HL,SFAN
F6D4 CD42F6             CALL    UNAME
F6D7 C238F6             JP      NZ,NAMERR
                ;
F6DA 13                 INC     DE
F6DB F7                 RST     030H            ; Get value
F6DC C2BEF7             JP      NZ,UNDEF        ; Error if none
                ;
F6DF ED43B0F7           LD      (ATTR),BC       ; Save value
F6E3 DD21E8D8           LD      IX,FCB5
F6E7 CD92F7             CALL    GETUF           ; Get filename into FCB
F6EA 1B                 DEC     DE
F6EB D5                 PUSH    DE
F6EC 2AB0F7             LD      HL,(ATTR)       ; HL = attrribute value
F6EF 11E8D8             LD      DE,FCB5
F6F2 CDFAF6             CALL    SFA             ; Set file attributes
F6F5 D1                 POP     DE              ; Error if not able to
F6F6 CABEF7             JP      Z,UNDEF         ; set file attributes
F6F9 C9                 RET                     ; To basic
                ;
                ; Set file attributes to filename in FCB
                ;---------------------------------------
                ;
                SFA
F6FA D5                 PUSH    DE
F6FB 13                 INC     DE
F6FC 0E0B               LD      C,11            ; C = length of filename
                SFAL
F6FE AF                 XOR     A               ; Clear carry & A
```

```
F6FF 29                 ADD     HL,HL           ; MS bit to carry
F700 CE00               ADC     A,0             ; A = 0 or 1
F702 0F                 RRCA                    ; LS bit A into MS bit
F703 47                 LD      B,A             ; B = #00 or #80
F704 EB                 EX      DE,HL           ; HL -> char. in FCB
F705 7E                 LD      A,(HL)          ; Get char.
F706 E67F               AND     07FH            ; Isolate all but attr. bit
F708 B0                 OR      B               ; Set attribute bit
F709 77                 LD      (HL),A          ; Store back in FCB
F70A EB                 EX      DE,HL           ; DE -> FCB HL = attr.bits
F70B 13                 INC     DE              ; Point to next
F70C 0D                 DEC     C               ; Dec filename counter
F70D 20EF               JR      NZ,SFAL         ; Set next
                ;
F70F D1                 POP     DE              ; DE -> FCB
F710 0E1E               LD      C,SETFAT        ; Call BDOS to set attr.
F712 CDB0F5             CALL    BDOS
F715 3C                 INC     A               ; If A = 0 then error
F716 C9                 RET
                ;
                ; Entry point for new GFA,<"filename"> command.
                ;-----------------------------------------------
                ;
                GFACOM
F717 21AAF7             LD      HL,SFAN
F71A CD42F6             CALL    UNAME
F71D C238F6             JP      NZ,NAMERR
                ;
F720 13                 INC     DE
F721 DD21E8D8           LD      IX,FCB5
F725 CD92F7             CALL    GETUF
F728 1B                 DEC     DE
F729 D5                 PUSH    DE
F72A 11E8D8             LD      DE,FCB5
F72D CD4DF7             CALL    GFA             ; Get file attributes
F730 2003               JR      NZ,GFAOK
                ;
F732 210100             LD      HL,0001         ; Error set GFA to 1
                GFAOK
F735 E5                 PUSH    HL
F736 C1                 POP     BC              ; BC = attr. value
F737 CDD00D             CALL    00DD0H          ; Convert to ASCII
F73A 21B6F7             LD      HL,GFABUF
                GFA1
F73D 1A                 LD      A,(DE)          ; Move ASCII bytes into buffer
F73E 77                 LD      (HL),A
F73F 13                 INC     DE
F740 23                 INC     HL
F741 FEFF               CP      0FFH
F743 20F8               JR      NZ,GFA1
                ;
F745 11B2F7             LD      DE,GFAVAR       ; DE -> GFA variable
F748 CDDA29             CALL    ROMLET          ; Rom LET routine
F74B D1                 POP     DE
F74C C9                 RET
                ;
                ; Get file attributes from filename in FCB
                ;-----------------------------------------
                ;
                GFA
```

```
F74D 2AB1E4             LD      HL,(0E4B1H)     ; HL -> DMA  (stored here)
F750 E5                 PUSH    HL
F751 D5                 PUSH    DE              ; DE -> FCB
F752 0E1A               LD      C,SETDMA
F754 11C0E8             LD      DE,DIRBUF
F757 CDB0F5             CALL    BDOS            ; Set DMA
F75A D1                 POP     DE
F75B 0E11               LD      C,SEARCHF
F75D CDB0F5             CALL    BDOS            ; Search for filename
F760 3C                 INC     A
F761 F5                 PUSH    AF              ; Save error flag (if any)
F762 2822               JR      Z,GFAOUT        ; Exit A = 0 = error
F764 3D                 DEC     A
F765 87                 ADD     A,A
F766 87                 ADD     A,A
F767 87                 ADD     A,A
F768 87                 ADD     A,A
F769 87                 ADD     A,A
F76A 5F                 LD      E,A
F76B 1600               LD      D,00            ; DE = A * 32
F76D 21C0E8             LD      HL,DIRBUF       ; HL -> directory buffer
F770 19                 ADD     HL,DE           ; DE -> directory entry
F771 23                 INC     HL
F772 EB                 EX      DE,HL           ; DE -> start of filename
                ;
F773 210000             LD      HL,0000         ; Zero attr. bit map
F776 060B               LD      B,00BH          ; B = length filename.type
                GFA2
F778 1A                 LD      A,(DE)          ; Get letter
F779 E680               AND     080H            ; Isolate attr. bit
F77B 07                 RLCA                    ; MS bit to LS bit
F77C B5                 OR      L               ; OR with any set bits
F77D 6F                 LD      L,A
F77E 29                 ADD     HL,HL           ; Shift left bit into bit map
F77F 13                 INC     DE
F780 10F6               DJNZ    GFA2            ; Get next
                ;
F782 29                 ADD     HL,HL
F783 29                 ADD     HL,HL
F784 29                 ADD     HL,HL           ; Left justify bit map to give
F785 29                 ADD     HL,HL           ; correct attribute value
                GFAOUT
F786 F1                 POP     AF              ; Restore error flag
F787 D1                 POP     DE
F788 F5                 PUSH    AF
F789 E5                 PUSH    HL
F78A 0E1A               LD      C,SETDMA
F78C CDB0F5             CALL    BDOS            ; Reset DMA
F78F E1                 POP     HL              ; Restore attr. value
F790 F1                 POP     AF              ; and error flag.
F791 C9                 RET
                ;
                ; Get filename and place in FCB
                ;-------------------------------
                ; Subroutine to get filename from basic and put
                ; in FCB.
                ;
                GETUF
F792 3AD2FA             LD      A,(PAGE)
```

```
F795 F5                 PUSH    AF              ; Save calling page / rom
F796 3E30               LD      A,030H
F798 32D2FA             LD      (PAGE),A
F79B D300               OUT     (00),A          ; Switch in disc rom
F79D CD2F2F             CALL    UFILEN          ; Get filename into FCB
F7A0 F1                 POP     AF
F7A1 32D2FA             LD      (PAGE),A
F7A4 D300               OUT     (00),A          ; Return to calling page / rom
F7A6 C9                 RET
                ;
                ; Valid names, variables, buffers
                ;--------------------------------
                ;
F7A7 024952     DIRN    DB      02,"IR"
F7AA 0346412C   SFAN    DB      03,"FA,"
                SNAME   DS      02
                ATTR    DS      02
F7B2 474641D4   GFAVAR  DB      "GFA",0D4H      ; GFA=
                GFABUF  DS      08              ; ASCII atttr. value + #FF
                ;
F7BE EF         UNDEF   RST     028H
F7BF 26                 DB      38              ; Undefined error


                        END
Workarea - 5DC9 to 6115
ORG  end - F7C0

LOAD end - B1A8
```

Forth Filing

Forth operating systems do not usually make indexed access of source files at all straight forward; some disc systems number saved blocks, but most tape based systems merely leave one fiddling endlessly with the tape counter and handwritten notes.

The following two words allow blocks of Forth to be saved and then retrieved under filenames used as in most other high-level languages. The words SAVET and LOADT are not redefinned, so that you can still use them to retrieve blocks without headers:

```
0   ( FILENAMES )
1 : FPUT ( --- )
2   0 BLOCK 64 ERASE ." Filename: " PAD 10 EXPECT CR
3   PAD 0 BLOCK 10 CMOVE UPDATE FLUSH SAVET ;
4
5 : FGET ( --- )
6   PAD 10 ERASE ." Filename: " PAD 10 EXPECT CR
7   BEGIN
8     PAD LOADT FLUSH 0 BLOCK DUP ." Found: "
9     10 TYPE CR 10 SWAP -TEXT
10  WHILE REPEAT ;
```

There are a number of published definitions of [-TEXT]; the easiest is possibly:

```
: -TEXT ( addr1,ct,addr2 ---f )
  SWAP 1
       DO 2DUP C@ SWAP 1+ SWAP
                  IF LEAVE
                  ELSE 1+ SWAP 1+ SWAP
          THEN
  LOOP
  C@ SWAP C@ - ;
```

Snakes and Ladders

```
1 RAND -9
5 GOSUB 7000
10 VS 4: COLOUR 4,7: COLOUR 2,15: COLOUR 3,1: COLOUR 0,11: COLOUR 1,1: CLS
20 CTLSPR 0,1: CTLSPR 1,1: CTLSPR 2,10: CTLSPR 5,10: CTLSPR 6,1
25 SBUF 10
26 GOSUB 5000
30 ANGLE 0: PLOT 76,18: DRAW 169
40 FOR T=1 TO 3: PHI PI/2: DRAW 170: NEXT T
50 GOSUB 9000
51 GOSUB 9500
52 LET J(4)=1
55 LET Y=1
56 FOR T=1 TO 5
60 LET X=10: LET C=0
70 CSR X,Y: PRINT CHR$(130): CSR X+1,Y: PRINT CHR$(130): CSR X,Y+1: PRINT CHR$(130): CSR X+1,Y+1
:| PRINT CHR$(130)
80 LET X=X+4: LET C=C+1
90 IF C=5 THEN  GOTO 100
91 IF C=10 THEN  GOTO 110
92 IF C=15 THEN  GOTO 120
93 IF C=20 THEN  GOTO 125
99 GOTO 70
100 COLOUR 1,11: LET X=12: GOTO 70
110 LET Y=Y+2: LET X=10: GOTO 70
120 COLOUR 1,1: LET X=12: GOTO 70
125 LET Y=Y+2
130 NEXT T
140 COLOUR 0,11: COLOUR 1,12
150 CSR 11,15: PRINT CHR$(134): CSR 11,16: PRINT CHR$(131): COLOUR 0,1: CSR 12,16: PRINT CHR$(13
2): COLOUR 0,11: CSR 12,17: PRINT CHR$(133)
160 CSR 12,18: PRINT CHR$(133): COLOUR 0,1: CSR 12,19: PRINT CHR$(131): CSR 13,19: PRINT CHR$(13
2): CSR 13,20: PRINT CHR$(131): COLOUR 0,11: CSR 14,20: PRINT CHR$(132)
170 COLOUR 1,4: CSR 17,13: PRINT CHR$(136): COLOUR 0,1: CSR 17,12: PRINT CHR$(136): CSR 17,11: P
RINT CHR$(136): COLOUR 0,11: CSR 17,10: PRINT CHR$(136): CSR 17,9
175 PRINT CHR$(136): COLOUR 0,1: CSR 17,8: PRINT CHR$(136)
180 COLOUR 0,11: COLOUR 1,12: CSR 17,5: PRINT CHR$(134): CSR 17,6: PRINT CHR$(133): COLOUR 0,1:
CSR 17,7: PRINT CHR$(131): COLOUR 0,11: CSR 18,7: PRINT CHR$(137)
185 CSR 19,7: PRINT CHR$(132): CSR 19,8: PRINT CHR$(133): COLOUR 0,1: CSR 19,9: PRINT CHR$(133):
 CSR 19,10: PRINT CHR$(133): COLOUR 0,11: CSR 19,11: PRINT CHR$(131)
190 COLOUR 0,1: CSR 20,11: PRINT CHR$(132): CSR 20,12: PRINT CHR$(133): COLOUR 0,11: CSR 20,13:
PRINT CHR$(133): CSR 20,14: PRINT CHR$(133)
195 COLOUR 0,1: CSR 20,15: PRINT CHR$(131): CSR 21,15: PRINT CHR$(137): COLOUR 0,11: CSR 22,15:
PRINT CHR$(138): COLOUR 0,1: CSR 22,14: PRINT CHR$(139)
200 CSR 23,14: PRINT CHR$(137): COLOUR 0,11: CSR 24,14: PRINT CHR$(132): COLOUR 0,1: CSR 24,15:
PRINT CHR$(131): CSR 25,15: PRINT CHR$(137): COLOUR 0,11
205 CSR 26,15: PRINT CHR$(132)
210 COLOUR 0,1: COLOUR 1,6: CSR 28,15: PRINT CHR$(136): COLOUR 0,11: CSR 28,14: PRINT CHR$(136):
 CSR 28,13: PRINT CHR$(136): COLOUR 0,1: CSR 28,12
211 PRINT CHR$(136)
215 CSR 28,11: PRINT CHR$(136): COLOUR 0,11: CSR 28,10: PRINT CHR$(136): CSR 28,9: PRINT CHR$(13
6): COLOUR 0,1: CSR 28,8: PRINT CHR$(136)
220 COLOUR 0,11: CSR 19,19: PRINT CHR$(136): COLOUR 0,1: CSR 19,18: PRINT CHR$(135): CSR 19,17:
PRINT CHR$(136): COLOUR 0,11: CSR 19,16: PRINT CHR$(136)
225 CSR 19,15: PRINT CHR$(136): COLOUR 0,1: CSR 19,14: PRINT CHR$(136)
230 COLOUR 0,11: COLOUR 1,7: CSR 16,1: PRINT CHR$(134): CSR 16,2: PRINT CHR$(133): COLOUR 0,1: C
SR 16,3: PRINT CHR$(131): CSR 17,3: PRINT CHR$(137)
240 COLOUR 0,11: CSR 18,3: PRINT CHR$(132): CSR 18,4: PRINT CHR$(133): COLOUR 0,1: CSR 18,5: PRI
NT CHR$(131): CSR 19,5: PRINT CHR$(137)
245 COLOUR 0,11: CSR 20,5: PRINT CHR$(132): COLOUR 1,3: CSR 11,8: PRINT CHR$(134): COLOUR 0,1: C
SR 11,9: PRINT CHR$(131)
```

```
250 COLOUR 0,11: CSR 12,9: PRINT CHR$(137): CSR 13,9: PRINT CHR$(137): COLOUR 0,1: CSR 14,9: PRI
NT CHR$(132): COLOUR 0,1: CSR 14,10: PRINT CHR$(133)
255 COLOUR 0,11: CSR 14,11: PRINT CHR$(133): CSR 14,12: PRINT CHR$(135)
256 COLOUR 0,1
260 COLOUR 1,9: CSR 23,2: PRINT CHR$(134): COLOUR 0,11: CSR 23,3: PRINT CHR$(131): COLOUR 0,1: C
SR 24,3: PRINT CHR$(137): CSR 25,3: PRINT CHR$(132)
265 CSR 25,4: PRINT CHR$(133): COLOUR 0,11: CSR 25,5: PRINT CHR$(131): COLOUR 0,1: CSR 26,5: PRI
NT CHR$(132)
270 COLOUR 1,13: CSR 10,13: PRINT CHR$(136): COLOUR 0,11: CSR 10,12: PRINT CHR$(136): CSR 10,11:
 PRINT CHR$(136): COLOUR 0,1: CSR 10,10: PRINT CHR$(136)
275 CSR 10,9: PRINT CHR$(136): COLOUR 0,11: CSR 10,8: PRINT CHR$(136): CSR 10,7: PRINT CHR$(136)
: COLOUR 0,1: CSR 10,6: PRINT CHR$(136)
280 COLOUR 0,11: COLOUR 1,12: CSR 12,2: PRINT CHR$(134): COLOUR 0,1: CSR 12,3: PRINT CHR$(131):
CSR 13,3: PRINT CHR$(137): COLOUR 0,11: CSR 14,3: PRINT CHR$(132)
285 CSR 14,4: PRINT CHR$(135)
290 COLOUR 0,1: COLOUR 1,6: CSR 23,5: PRINT CHR$(134): CSR 23,6: PRINT CHR$(131): COLOUR 0,11: C
SR 24,6: PRINT CHR$(132): COLOUR 0,1: CSR 24,7: PRINT CHR$(131)
295 CSR 25,7: PRINT CHR$(137): COLOUR 0,11: CSR 26,7: PRINT CHR$(132): CSR 26,8: PRINT CHR$(133)
: COLOUR 0,1: CSR 26,9: PRINT CHR$(135)
300 COLOUR 1,4: CSR 15,17: PRINT CHR$(136): COLOUR 0,11: CSR 15,16: PRINT CHR$(136): CSR 15,15:
PRINT CHR$(136): COLOUR 0,1: CSR 15,14: PRINT CHR$(136)
305 COLOUR 1,2: CSR 22,6: PRINT CHR$(136): COLOUR 0,11: CSR 22,7: PRINT CHR$(136): CSR 22,8: PRI
NT CHR$(136): COLOUR 0,1: CSR 22,9: PRINT CHR$(136)
306 COLOUR 0,1: CSR 22,5: PRINT CHR$(136): COLOUR 0,11: CSR 22,4: PRINT CHR$(136)
310 COLOUR 0,1: CSR 22,10: PRINT CHR$(136): COLOUR 0,11: CSR 22,11: PRINT CHR$(136): CSR 22,12:
PRINT CHR$(136): COLOUR 0,1: CSR 22,13: PRINT CHR$(136)
320 COLOUR 0,15: COLOUR 1,1: CSR 1,2: PRINT "PLAYER"
330 SPRITE 1,7,80,30,0,0,C(1)
331 SPRITE 2,7,88,30,0,0,C(2)
332 SPRITE 3,7,88,30,0,0,C(3)
333 SPRITE 1,7,98,30,0,0,C(4)
500 REM CONTROL LOOP
510 FOR M=1 TO N
512 COLOUR 1,C(M): IF C(M)=15 THEN  COLOUR 1,1
515 LET J(M)=L(M)
520 CSR 1,4: PRINT "          ": CSR 1,4: PRINT N$(M)
530 GOSUB 5500
540 GOSUB 1000
550 NEXT M
560 GOSUB 6000
570 GOTO 500
999 GOTO 999
1000 FOR T=1 TO X
1001 IF L(M)+X>=100 THEN  GOTO 2000
1003 IF J(M)/10=INT(J(M)/10) THEN  GOTO 1300
1005 FOR Z=1 TO 16
1020 SOUND 3,S(M),15
1026 IF D(M)=0 THEN  ADJSPR 0,M,11: PAUSE 20
1027 IF D(M)=4 THEN  ADJSPR 0,M,18: PAUSE 20
1030 MVSPR 1,M,D(M): SOUND 3,0,0
1031 IF D(M)=4 THEN  ADJSPR 0,M,19: PAUSE 20
1032 IF D(M)=0 THEN  ADJSPR 0,M,12: PAUSE 20
1035 NEXT Z
1037 LET J(M)=J(M)+1
1040 GOTO 1400
1300 ADJSPR 0,M,9
1310 FOR F=1 TO 16: SOUND 3,S(M),15: MVSPR 1,M,6: PAUSE 50: SOUND 3,0,0: NEXT F
1330 ADJSPR 0,M,7
1340 IF D(M)=0 THEN  LET D(M)=4: LET J(M)=1: GOTO 1400
1350 IF D(M)=4 THEN  LET D(M)=0: LET J(M)=1: GOTO 1400
1400 NEXT T
1410 LET L(M)=L(M)+X
1420 IF B(L(M))>0 THEN  GOSUB 3000
1430 IF B(L(M))<0 THEN  GOSUB 4000
1440 ADJSPR 0,M,7
1500 RETURN
```

```
2000 IF L(M)+X=100 THEN  GOTO 2200
2005 LET A=0
2010 FOR T=1 TO X
2020 LET A=A+1
2030 FOR Z=1 TO 16
2040 SOUND 3,S(M),15
2055 ADJSPR 0,M,10: PAUSE 30
2060 SOUND 3,0,0: MVSPR 1,M,D(M)
2070 ADJSPR 0,M,19: PAUSE 30
2080 NEXT Z
2090 IF L(M)+A=100 THEN  GOTO 2120
2100 NEXT T
2120 LET L(M)=100
2125 LET D(M)=0
2130 FOR T=99 TO (100-(X-A)) STEP -1
2140 FOR Z=1 TO 16
```

```
2150 SOUND 3,S(M),15
2160 ADJSPR 0,M,11: PAUSE 30
2170 MVSPR 1,M,D(M): SOUND 3,0,0
2175 SOUND 3,0,0
2180 ADJSPR 0,M,12: PAUSE 30
2181 NEXT Z
2185 NEXT T
2186 ADJSPR 0,M,7
2190 LET L(M)=100-(X-A)
2191 LET D(M)=4
2192 IF B(L(M))<0 THEN  GOSUB 4000
2193 IF B(L(M))>0 THEN  GOSUB 3000
2195 GOTO 1500
2200 FOR T=L(M) TO 99
2210 FOR Z=1 TO 16
2220 SOUND 3,S(M),15
2230 ADJSPR 0,M,18: PAUSE 30
2240 MVSPR 1,M,D(M): SOUND 3,0,0
2250 ADJSPR 0,M,19: PAUSE 30
2260 NEXT Z
2280 NEXT T
2500 FOR T=0 TO 6: CSR 0,T: PRINT "          ": NEXT T
2503 COLOUR 1,C(M): IF C(M)=15 THEN  COLOUR 1,1
2505 CSR 1,1: IF MK=3 THEN  PRINT "WELL    ": PRINT " DONE   ": PRINT " ";N$(M): GOTO 2550
2510 CSR 1,1: PRINT "I WIN   "
2550 FOR T=1 TO 30: ADJSPR 0,M,7: PAUSE 100: ADJSPR 0,M,9: SOUND 3,S(M),15
2555 PAUSE 20: SOUND 3,0,0: PAUSE 10: SOUND 3,S(M),15: PAUSE 20: SOUND 3,0,0: ADJSPR 0,M,9: PAUS
E 100: NEXT T
2556 CTLSPR 0,1: CTLSPR 1,1: CTLSPR 2,10: CTLSPR 5,10: CTLSPR 6,1
2560 CLS : CSR 8,12: PRINT "ANOTHER GAME Y/N"
2570 IF INKEY$<>"Y" AND INKEY$<>"N" THEN  GOTO 2570
2580 LET A$=INKEY$
2590 IF A$="Y" THEN  GOTO 1
2600 NEW
3000 FOR T=L(M)+1 TO 100
3005 LET A=T
3010 IF B(T)=B(L(M)) THEN  GOTO 3035
3020 NEXT T
3030 GOTO 3120
3035 ADJSPR 0,M,9
3040 FOR F=INT(L(M)/10)+1 TO INT(A/10)
3041 FOR Z=1 TO 16
3045 SOUND 1,600-(F*50),15
3050 MVSPR 1,M,6
3060 PAUSE 50
3065 SOUND 1,0,0
3066 NEXT Z
3070 NEXT F
3075 LET J(M)=A/10-INT(A/10): LET J(M)=J(M)*10
3080 IF INT((A-L(M))/10)=(A-L(M))/10 THEN  GOTO 3100
3090 IF D(M)=4 THEN  LET D(M)=0: GOTO 3100
3095 IF D(M)=0 THEN  LET D(M)=4
3100 LET L(M)=A
3110 ADJSPR 0,M,7
3120 RETURN
4000 FOR T=L(M)-1 TO 1 STEP -1
4010 LET A=T
4015 CSR 5,19
4020 IF B(T)=B(L(M)) THEN  GOTO 4100
4030 NEXT T
4040 GOTO 4220
4100 ADJSPR 0,M,10
4105 LET Y=1
4110 FOR F=INT(L(M)/10) TO INT((A/10)+1) STEP -1
4112 LET Y=Y+1
4115 FOR Z=1 TO 16
```

```
4120 SOUND 1,(300+(Y*50)),15
4130 MVSPR 1,M,1
4140 PAUSE 30
4145 SOUND 1,0,0
4150 NEXT Z
4155 NEXT F
4160 LET J(M)=A/10-INT(A/10): LET J(M)=J(M)*10
4170 LET L(M)=A
4180 IF L(M)<=10 THEN  LET D(M)=0
4190 IF L(M)<=20 AND L(M)>=11 THEN  LET D(M)=4
4191 IF L(M)<=30 AND L(M)>=21 THEN  LET D(M)=0
4192 IF L(M)<=40 AND L(M)>=31 THEN  LET D(M)=4
4193 IF L(M)<=50 AND L(M)>=41 THEN  LET D(M)=0
4194 IF L(M)<=60 AND L(M)>=51 THEN  LET D(M)=4
4195 IF L(M)<=70 AND L(M)>=61 THEN  LET D(M)=0
4196 IF L(M)<=80 AND L(M)>=71 THEN  LET D(M)=4
4197 IF L(M)<=90 AND L(M)>=81 THEN  LET D(M)=0
4198 IF L(M)>=91 THEN  LET D(M)=4
4200 FOR T=1 TO 10
```

```
4205 FOR Z=13 TO 17: ADJSPR 0,M,Z: SOUND 0,100,15: PAUSE 20: SOUND 0,0,0: PAUSE 50: NEXT Z
4206 NEXT T
4210 ADJSPR 0,M,7
4220 RETURN
5000 REM PLAYERS
5001 COLOUR 0,15: COLOUR 1,1: CLS : CSR 2,14: PRINT "ENTER NUMBER OF PLAYERS": CSR 9,16: PRINT "
(3 MAX.)": INPUT N
5002 IF N>3 OR N<0 THEN  GOTO 5000
5005 DIM N$(3,8)
5009 CLS : CSR 8,4: PRINT "(8 LETTERS MAX.)"
5010 FOR T=1 TO N: SOUND 1,0,0: CSR 8,14: PRINT "NAME OF PLAYER ";T: INPUT N$(T): SOUND 1,300,15
: CSR 0,15: PRINT "          ": PAUSE 50: NEXT T
5012 SOUND 1,0,0
5020 CLS : RETURN
5500 COLOUR 1,1: CSR 1,6: PRINT "PRESS ": PRINT " ANY KEY": PRINT " TO": PRINT " THROW": PRINT
 DICE"
5510 LET A$=INKEY$
5520 IF A$="" THEN  GOTO 5510
5530 FOR T=6 TO 10: CSR 0,T: PRINT "          ": NEXT T
5540 FOR T=1 TO 80: LET X=INT(RND*6+1): SPRITE 5,X,165,8,0,0,1: NEXT T
5550 CSR 1,6: PRINT "ANOTHER": PRINT " THROW?": PRINT " (Y/N)"
5560 LET A$=INKEY$
5570 IF A$<>"Y" AND A$<>"N" THEN  GOTO 5560
5580 IF A$="Y" THEN  GOTO 5600
5590 GOTO 5610
5600 FOR T=1 TO 80: LET X=INT(RND*6+1): SPRITE 5,X,165,8,0,0,1: NEXT T
5610 FOR T=6 TO 8: CSR 0,T: PRINT "          ": NEXT T
5620 RETURN
6000 COLOUR 1,C(4): CSR 1,4: PRINT "          ": CSR 1,4: PRINT " MTX"
6010 LET M=4
6015 LET J(M)=L(M)
6020 PAUSE 1000
6050 FOR T=1 TO 80: LET X=INT(RND*6+1): SPRITE 5,X,165,8,0,0,1: NEXT T
6070 IF L(M)+X>100 THEN  LET Y=100-L(M)
6071 IF L(M)+X<=100 THEN  GOTO 6074
6072 IF L(M)+X>100 AND B(100-(X-Y))<0 THEN  GOTO 6090
6074 IF L(M)+X=100 THEN  GOTO 6100
6075 IF B(L(M)+X)>0 THEN  GOTO 6100
6076 IF L(M)+X>70 AND B(L(M)+X)>=0 THEN  GOTO 6100
6077 IF B(L(M)+X)<0 THEN  GOTO 6090
6078 FOR T=1 TO 6: IF B(L(M)+T)>0 THEN  GOTO 6090
6079 NEXT T
6080 FOR T=1 TO 6: IF B(L(M)+T)<0 THEN  GOTO 6100
6081 NEXT T
6085 IF X>=5 THEN  GOTO 6100
6090 PAUSE 2000: FOR I=1 TO 80: LET X=INT(RND*6+1): SPRITE 5,X,165,8,0,0,1: NEXT I
6100 GOSUB 1000
6120 RETURN
7000 VS 4: COLOUR 0,14: COLOUR 4,15: COLOUR 2,14: COLOUR 3,1: COLOUR 1,1: CLS
7001 CTLSPR 0,3: CTLSPR 1,1: CTLSPR 2,15: CTLSPR 3,15: CTLSPR 5,15: CTLSPR 6,3
7009 GENPAT 1,140,0,0,153,255,102,0,0,0
7010 FOR T=0 TO 31: CSR T,5: PRINT CHR$(140): CSR T,9: PRINT CHR$(140): NEXT T
7030 COLOUR 1,12: CSR 7,7: PRINT "SNAKES": COLOUR 1,1: CSR 14,7: PRINT "AND": COLOUR 1,4: CSR 18
,7: PRINT "LADDERS"
7040 COLOUR 1,6: CSR 2,18: PRINT "WRITTEN BY MIKE MAJOR (C)1984"
7499 PAUSE 6000
7500 RETURN
9000 GENPAT 1,129,255,255,255,255,255,255,255,255
9010 GENPAT 3,7,56,56,16,124,84,124,40,108
9020 GENPAT 3,8,56,56,16,124,84,124,68,198
9030 GENPAT 3,9,56,186,146,254,16,124,198,0
9040 GENPAT 1,131,14,14,14,14,14,7,3,1
9050 GENPAT 1,132,0,0,0,0,0,248,252,254
9060 GENPAT 1,133,14,14,14,14,14,14,14,14
9070 GENPAT 1,134,0,0,0,4,4,4,14,14
9080 GENPAT 1,135,14,14,4,0,0,0,0,0
```

```
9090 GENPAT 1,136,66,66,66,66,126,66,66,66
9100 GENPAT 1,137,0,0,0,0,0,255,255,255
9110 GENPAT 1,138,7,7,7,7,7,254,254,248
9120 GENPAT 1,139,0,0,0,0,0,3,7,7
9130 GENPAT 3,1,255,255,255,239,255,255,255,255
9140 GENPAT 3,2,255,191,255,255,255,255,253,255
9150 GENPAT 3,3,255,191,255,255,247,255,253,255
9160 GENPAT 3,4,255,219,255,255,255,219,255,255
9170 GENPAT 3,5,255,219,255,255,239,255,219,255
9180 GENPAT 3,6,255,219,255,255,219,255,219,255
9190 DIM C(4): LET C(1)=12: LET C(2)=15: LET C(3)=4: LET C(4)=0
9200 DIM L(4)
9210 FOR T=1 TO 4: LET L(T)=1: NEXT T
9220 DIM D(4)
9230 FOR T=1 TO 4: LET D(T)=0: NEXT T
9240 DIM S(4)
9250 LET S(1)=1: LET S(2)=2: LET S(3)=0: LET S(4)=4
9280 DIM J(4)
9290 GENPAT 3,10,0,60,126,24,127,75,90,20
9300 GENPAT 3,11,28,20,16,24,24,24,8,12
9310 GENPAT 3,12,28,20,16,24,24,36,36,16
9320 GENPAT 3,13,0,0,0,8,0,1,157,255
9330 GENPAT 3,14,0,32,0,8,0,1,157,255
9340 GENPAT 3,15,0,0,0,1,0,1,157,255
9350 GENPAT 3,16,4,0,0,8,0,1,157,255
9360 GENPAT 3,17,0,8,1,0,0,1,157,255
9370 GENPAT 3,18,56,40,8,24,24,24,18,48
9380 GENPAT 3,19,56,40,8,24,24,88,36,8
9390 LET Y=0
9400 GENPAT 1,140,0,0,153,255,102,0,0,0
9490 RETURN
9500 DIM B(106)
9510 FOR T=1 TO 100: READ B(T): NEXT T
9520 DATA    0,0,-1,0,3,0,0,0,0,0,0,0,0,0,0,0,2,0,0,-1,0,0,0,0,0,0,0,-3,5,0,0,0,6,0,3,4,2,0,1,
0,0,-2,0,0,0,0,0,0,0,0,-5,0,0,0,0,0,0,0,0,-2,0,0,4,0,0,0,0
9530 DATA   0,5,0,-4,0,-5,-6,0,-3,0,0,1,0,0,-7,0,0,0,6,0,0,0,0,0,0,-4,0,0,-6,0,-7,0
9999 RETURN
10000 SAVE "SNAKES AND LADDERS"
10010 POKE 64862,13
10020 RUN
```

Subscriptions

IF YOUR MEMBERSHIP NUMBER IS BETWEEN 0 - 001750 (ALL LETTERS), YOU ARE NOW DUE TO RENEW YOUR SUBSCRIPTION IF YOU HAVE NOT ALREADY RENEWED ONCE. TO MAKE SURE OF RECEIVING YOUR NEXT COPY OF MEMOPAD PLEASE SEND YOUR SUB-SCRIPTION TO REACH US NO LATER THEN THE 5TH OCTOBER 1986.

2ND RENEWALS ARE NOW DUE FOR RENEWAL (00500 - 00750 INCL.) UNLESS ALREADY RENEWED.

PLEASE NOTE NEW MEMBERSHIP PRICES ARE AS FOLLOWS:-

£18.00	U.K MEMBERS
£27.50	E.E.C. MEMBERS
£36.50	OUTSIDE EUROPF

Bit Mapping - Part 2

Last month I showed you how to configure Black Beauty into a bit-mapped mode, and how to go about writing a machine code program that would allow easy transfer of data to the screen. There is a much simpler method of doing this, but I deliberately showed you the hard way so that you would know how the VDP operates and, of course, the same routine could be used on the MSX or Einstein machines, or, for that matter, any machine that uses the Texas VDP chip.

The following routine is in the MTX ROM. You use exactly the same parameters as set up in last months program:

3 = CSR x,y
4 = PAPER n
6 = INK n

However, instead of having to write the routine yourself, just **CALL £00BC..** That's all there is to it ! There is no need to re-configure VRAM, just use the Basic command **VS 4** as the first line in your program and let rom take care of it for you. While I'm on the plane, I'll try to think up some more goodies ...★

```
          10  VS 4 : CLS
          ASSEM 20
             ;ROUTINE TO PRINT "THIS IS A TEST"
             ;
          LD    HL,BUFFER
          LD    B,19
   LOOP:  LD    A,(HL)
          CALL  £00BC
          INC   HL
          DJNZ  LOOP
   STOP:  JR    STOP
BUFFER:DB    6,1,3,0,0,'THIS IS A TEST'
```


JARO COMPUTER SERVICES ARE HAVING TROUBLE WITH GOOD OLD BRITISH TELECOM. ANY URGENT MESSAGES CAN BE LEFT ON PRESTEL. JARO'S NUMBER IS 019995085.

Hardware

STOCK NO.	MANUFACT. REFERENCE	TITLE	TYPE	RETAIL PRICE (INCL.VAT)	MEMBERS PRICE (INCL. VAT)
		MEMOTECH HARDWARE			
		==================			
6000	MTX 500	NO LONGER AVAILABLE			
6001	MTX 512	64K COMPUTER		79.95	69.95
6003		MEMORY 32K UPGRADE		39.95	37.95
6004		MEMORY 64K UPGRADE		49.95	47.45
6005		MEMORY 128K UPGRADE		79.95	75.95
6010		SPECULATOR		39.95	37.95
6011		NEWWORD ON ROM		39.95	37.95
6012		PASCAL ON ROM		39.95	37.95
6013		RS232 BOARD		39.95	37.95
6021	SDX PACK 1	500K DRIVE/INTERFACE/80 COL. BOARD/CPM		399.00	359.10
6022	SDX PACK 2	*** DITTO *** BUT WITH 1MB DRIVE		449.00	404.10
6023	SDX PACK 3	AS PACK 1 PLUS MTX 512		499.00	449.10
6024	SDX PACK 4	*** DITTO *** BUT WITH 1MB DRIVE		549.00	494.10
6031	SDX	2ND DRIVE 500K UPGRADE		189.95	170.96
6032	SDX	2ND DRIVE 1MB UPGRADE		229.95	206.96
6036		80 COL PCB CP/M NW+SC		199.95	184.95
6037		80 COL PCB RS232 KIT		27.00	27.09
6042	SDX	SDX 500K DRIVE & INTERFACE		249.95	224.96
6043	SDX	SDX 1 MB DRIVE & INTERFACE		299.95	269.96
6050	FDX	TWIN 500K (REQUIRES RS232 BOARD)		649.00	584.10
6051	FDX	TWIN 1 MB (REQUIRES RS232 BOARD)		825.00	742.50
6055		SILICON DISC 1MB		179.00	161.10
6056		SILICON DISC 2MB		338.00	304.20
6057		SILICON DISC 4MB		636.00	572.40
6065	FDX	2ND DRIVE 500K UPGRADE		159.00	143.10
6066	FDX	2ND DRIVE 1MB UPGRADE		189.00	170.10
6071	SDX 3.5 PACK 1	1 MB 3.5" DRIVE PLUS INTERFACE		149.00	134.10
6072	SDX 3.5 PACK 2	******* DITTO ******* PLUS SILICON DISC, 80 COLUMN CARD AND CPM		239.00	215.10
6073	SDX 3.5	80 COLUMN CARD, CPM AND SILICON DISC		129.00	116.10
6080	DMX 80	PRINTER		199.95	179.95
6085		PRINTER CABLE 2 METRE		14.95	13.46
6090	KX M110	DMX 80 PRINTER RIBBON		9.95	8.96
6095		FDX TO MTX CONNECTOR		25.30	22.77
6097	TM110 V23	TANDATA MODEM FOR MEMOTECH 1200/75 BAUD		119.95	107.96
6100		DUST COVER FOR MTX		5.99	4.95
6105	DS QD	5.25" BLANK DISCS (BOX OF 10)		27.95	19.95
6110		5.25" DISC CASES (HOLDS 10 DISCS)		2.95	2.65
6120		FLOPPICLENE KIT FOR 5.25"		19.10	17.20
6125		ANTISTATIC WIPES (PACK OF 10)		1.67	1.50

Manuals & Advertising

	MEMOTECH MANUALS (NO VAT)			
	==========			
6500	MICROCOSM	6.95		5.00
6501	CRIB CARD	2.95		2.12
6502	MTX ROM LISTING	45.00		32.40
6503	SDX CONTROLLER LISTING	25.00		18.75
6504	ROM CALLS INFORMATION SHEET	0.50		0.50
6505	RST10 CALLS INFORMATION SHEET	0.50		0.50
6506	INTERRUPTS INFORMATION SHEET	0.80		0.80
6507	MTX SERVICE MANUAL	9.95		9.95
6508	NEW USER MANUAL FOR MTX	8.95		8.95
6509	VDP MANUAL	7.95		7.95
6510	DDT MANUAL	2.50		2.50
	ADVERTISING IN THE MEMOPAD			
	==========			
	SMALL	5.00		2.50
	1/8 PAGE	27.50		27.50
	1/4 PAGE	45.00		45.00
	1/2 PAGE	80.00		80.00
	FULL PAGE	125.00		125.00

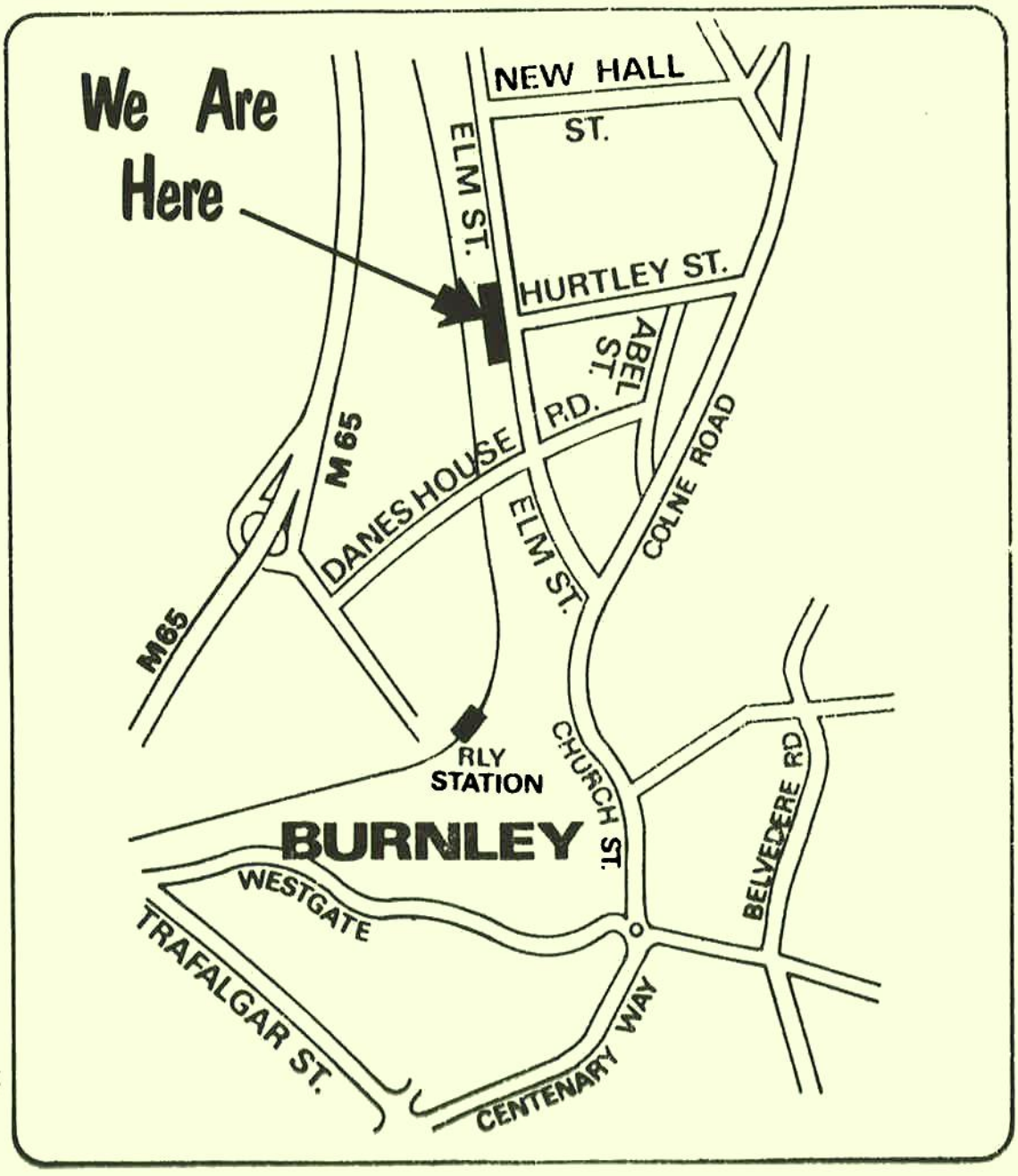
Software

TYPE CODES

U UTILITY
L LANGUAGE
B BUSINESS
E EDUCATIONAL
G GAME
J JOYSTICK COMPATIBLE

STOCK NO.	MANUFACT. REFERENCE	TITLE	TYPE	RETAIL PRICE (INCL.VAT)	MEMBERS PRICE (INCL. VAT)
		SOFTWARE (MEMOTECH)			
		==========			
7000	00106	26X26 SPREAD SHEET	U	8.83	7.95
7001	00057	3D TACHYON FIGHTER	G+J	7.72	6.95
7003	00062	ADVENTURE QUEST	G+J	9.95	8.96
7004	00033	AGROVATOR	G+J	6.61	5.95
7009	00071	ALICE IN WONDERLAND	G+J	6.69	6.02
7011	00141	ASSEM LANG COURSE	E	9.94	8.95
7012	00008	ASTROMILON	G+J	6.69	6.02
7013	00047	ASTROPAC	G+J	6.69	6.02
7014	.	ATTACK OF KILLER TOMATOES	G+J	8.83	7.95
7033	00058	BACKGAMMON	G	8.83	7.95
7034	00041	BASIC BUSINESS	B	6.61	5.95
7035	00043	BLOBBO	G+J	6.69	6.02
7036	00073	BOUNCING BILL	G+J	5.50	4.95
7037	00074	BRIDGE	G	7.72	6.95
7058	00077	CANVAS	U	8.83	7.95
7059	00085	CAVES OF ORB	G	6.61	5.95
7060		CAVES OF DOOM	G+J	5.50	4.95
7061	00094	CHAMBEROIDS	G	6.61	5.95
7062	00059	CHESS	G	9.72	8.75
7063	00053	COBRA	G	6.69	6.02
7064	00025	COLOSSAL ADVENTURE	G	9.95	8.96
7065	00098	COMBAT	G	3.28	2.95
7067	00046	CONT RAIDERS	G	6.69	6.02
7068	00099	CRIBBAGE (TEMPORARY NOT AVAILABLE)	G	3.28	2.95
7069	00110	CRYSTAL	G	6.61	5.95
7089	00050	DEN.GOES BANANAS	G	3.28	2.95
7090	00011	DENNIS & CHICKEN	G	3.28	2.95
7091	00103	DENNIS AND CIRCUS	G	3.28	2.95
7092		DISASM	U	8.83	7.95
7093	00068	DOODLEBUG	G	5.50	4.95
7094	00108	DOWNSTREAM DANGER	G	6.61	5.95
7095	00096	DR. FRANKIE	G	6.61	5.95
7096	00056	DRAUGHTS	G	7.72	6.95
7097	00111	DRIVE THE CEE 5	G	6.61	5.95
7098	00063	DUNGEON ADVENTURE	G	9.95	8.96
7116	00067	EDASM	U	8.83	7.95
7117	00067D	EDASM SDX (DISC)	U	9.94	8.95
7118	00066	EMERALD ISLE	G	7.72	6.95
7119	00038	ESCAPE FROM ZARKOS	G	6.61	5.95
7120	00081	EXTENDED BASIC	U	7.72	6.95
7121		F1 SIMULATOR	G+J	5.50	4.95
7140	00082	FATHOMS DEEP	G	6.61	5.95
7141	00090	FIG FORTH	L	17.50	15.75
7142	00090D	FIG FORTH SDX (DISC)	L	17.50	15.75
7143	00055	FIREHOUSE FREDDIE	G	6.69	6.02
7144	00021	FIRST LETTERS 1	E	9.72	8.75
7145	00092	FKEY DEFINER	U	7.72	6.95
7146	00037	FLUMMOX	G	6.61	5.95
7166	00052	GAUNTLET	G	6.69	6.02
7167	00102	GHOSTLY CASTLE	G	3.28	2.95
7168	00031	GOLDMINE	G	6.69	6.02
7169	00069	GRAPHICS	U	6.61	5.95
7190	00072	HAWKWARS	G	5.50	4.95
7191	00065	HELI-MATHS	E	6.61	5.95
7192	00139	HIGHWAY ENCOUNTER	G	8.83	7.95
7193	00034	HUNCHY	G	5.50	4.95
7213	00083	ICEBERG	G	5.50	4.95
7214		INTO OBLIVION	G+J	5.50	4.95
7233	00105	JET SET WILLY (TEMPORARY NOT AVAILABLE)	G	7.72	6.95
7234	00015	JOHNNY REB	G	6.69	6.02
7235	00097	JUMPING JACK FLASH	G	6.61	5.95

7255	**	00115	*	KARATE KING	*	G	**	6.61	**	5.95
7256	**	00016	*	KEY TO TIME	*	G	**	6.69	**	6.02
7257	**	00042	*	KILOPEDE	*	G	**	6.69	**	6.02
7258	**	00019	*	KNUCKLES	*	G	**	8.83	**	7.95
7278	**	00078	*	LES FLICS	*	G	**	7.72	**	6.95
7279	**	00032	*	LITTLE DEVILS	*	G	**	5.50	**	4.95
7280	**	00024	*	LORDS OF TIME	*	G	**	9.95	**	8.96
7300	**	00035	*	MISSILE COMMAND & ARCADIANS	*	G	**	5.50	**	4.95
7301	**	00070	*	MAN FROM GRANNY	*	G	**	5.50	**	4.95
7302	**	00104	*	MANIC MINER (TEMPORARY NOT AVAILABLE)	*	G	**	7.72	**	6.95
7305	**	00022	*	MATHS 1	*	E	**	9.72	**	8.75
7306	**	00013	*	MAXIMA	*	G	**	6.69	**	6.02
7307	**	00086	*	MEMOCHEQUE	*	U	**	7.72	**	6.95
7308	**	00075	*	MEMOSKETCH	*	U	**	8.83	**	7.95
7309	**	00075D	*	MEMOSKETCH SDX (DISC)	*	U	**	9.94	**	8.95
7310	**	00089	*	MINER DICK	*	G	**	7.72	**	6.95
7311	**	00044	*	MISSION ALPHATRON	*	G	**	6.69	**	6.02
7312	**	00030	*	MISSION OMEGA	*	G	**	5.50	**	4.95
7313	**	00054	*	MURDER AT MANOR	*	G	**	6.69	**	6.02
7314	**	00010	*	MUSIC PAD	*	U	**	6.69	**	6.02
7333	**	00003	*	NEMO	*	G	**	6.67	**	6.00
7334	**	00131	*	NETWORK LOADER	*	U	**	9.95	**	8.96
7354	**	00112	*	OBLITERATION ZONE	*	G	**	6.61	**	5.95
7355	**	00045	*	OBLOIDS	*	G	**	6.69	**	6.02
7356	**		*	ONE MAN AND HIS DROID	*	G+J	**	5.50	**	4.95
7375	**	00129	*	PAINTBOX	*	U	**	6.61	**	5.95
7376	**	00001	*	PAYROLL	*	U	**	23.61	**	21.25
7377	**	00005	*	PHAID	*	G	**	6.69	**	6.02
7378	**	00061	*	PHYSICS 1	*	E	**	9.72	**	8.75
7380	**	00012	*	PONTOON & BLACKJACK	*	G	**	6.69	**	6.02
7381	**	00009	*	POT HOLE PETE	*	G	**	6.69	**	6.02
7382	**	00040	*	PURCHASE LEDGER	*	B	**	14.17	**	12.75
7402	**	00048	*	QOGO	*	G	**	6.69	**	6.02
7403	**	00076	*	QOGO 2	*	G	**	6.61	**	5.95
7404	**	00095	*	QUANTUM	*	G	**	6.61	**	5.95
7405	**	00109	*	QUAZZIA	*	G	**	6.61	**	5.95
7427	**	00064	*	RETURN TO EDEN	*	G	**	9.95	**	8.96
7428	**	00020	*	REVERSI	*	G	**	8.83	**	7.95
7429	**	00114	*	ROLLA BEARING	*	G	**	6.61	**	5.95
7430	**	00100	*	RUTHLESS B.	*	G	**	2.78	**	2.50
7450	**	00002	*	SALES LEDGER	*	B	**	17.50	**	15.75
7451	**	00029	*	SALTY SAM	*	G	**	5.50	**	4.95
7452	**	00113	*	SEPULCRI SCELERATI	*	G	**	6.61	**	5.95
7453	**	00101	*	SLOOPY'S CHRISTMAS	*	G	**	3.28	**	2.95
7454	**	00116	*	SMG	*	G	**	6.61	**	5.95
7455	**	00049	*	SNAPPO	*	G	**	6.69	**	6.02
7456	**	00140	*	TOURNAMENT SNOOKER	*	G	**	8.83	**	7.95
7457	**	00023	*	SNOWBALL	*	G	**	9.95	**	8.96
7458	**	00036	*	SON OF PETE	*	G	**	6.61	**	5.95
7459	**		*	SOUL OF THE ROBOT	*	G+J	**	5.50	**	4.95
7460	**		*	SPELLBOUND	*	G+J	**	5.50	**	4.95
7461	**	00026	*	SPELLI-COPTER	*	G	**	6.61	**	5.95
7462	**	00017	*	STAR COMMAND	*	G	**	7.72	**	6.95
7463	**	00014	*	SUPA CODER	*	U	**	8.83	**	7.95
7464	**	00084	*	SUPER BIKE	*	G	**	5.50	**	4.95
7465	**	00004	*	SUPER MINEFIELD	*	G	**	6.69	**	6.02
7466	**	00093	*	SURFACE SCANNER	*	G	**	6.61	**	5.95
7490	**	00039	*	TAPE TO DISC	*	U	**	7.72	**	6.95
7491	**	00007	*	TAPEWORM	*	G	**	6.69	**	6.02
7492	**	00088	*	TARGET ZONE	*	G	**	7.72	**	6.95
7493	**	00118	*	THE DESIGNER	*	U	**	9.95	**	8.96
7494	**	00128	*	THE WALL	*	G	**	5.50	**	4.95
7495	**	00051	*	THE ZOO GAME	*	G	**	6.69	**	6.02
7497	**	00006	*	TOADO	*	G	**	6.69	**	6.02
7500	**	00018	*	TURBO	*	G	**	7.72	**	6.95
7520	**	00117	*	USER BASIC	*	U	**	9.95	**	8.96
7521	**	00117D	*	USER BASIC SDX (DISC)	*	U	**	11.05	**	9.95
7522	**	00079	*	USER EXTEND	*	U	**	8.83	**	7.95
7523	**	00027	*	UTILITIES 1	*	U	**	5.50	**	4.95
7524	**	00027D	*	UTILITIES SDX (DISC)	*	U	**	11.05	**	9.95
7542	**	00091	*	VERNON & VAMPIRES	*	G	**	6.61	**	5.95
7566	**	00060	*	WORD & PICTURE	*	E	**	9.72	**	8.75


We Are Here
NEW HALL ST.
ELM ST.
HURTLEY ST.
ABEL ST.
P.D.
M 65
DANES HOUSE
ELM ST.
COLNE ROAD
M65
RLY STATION
CHURCH ST.
BELVEDERE RD
BURNLEY
WESTGATE
TRAFALGAR ST.
CENTENARY WAY