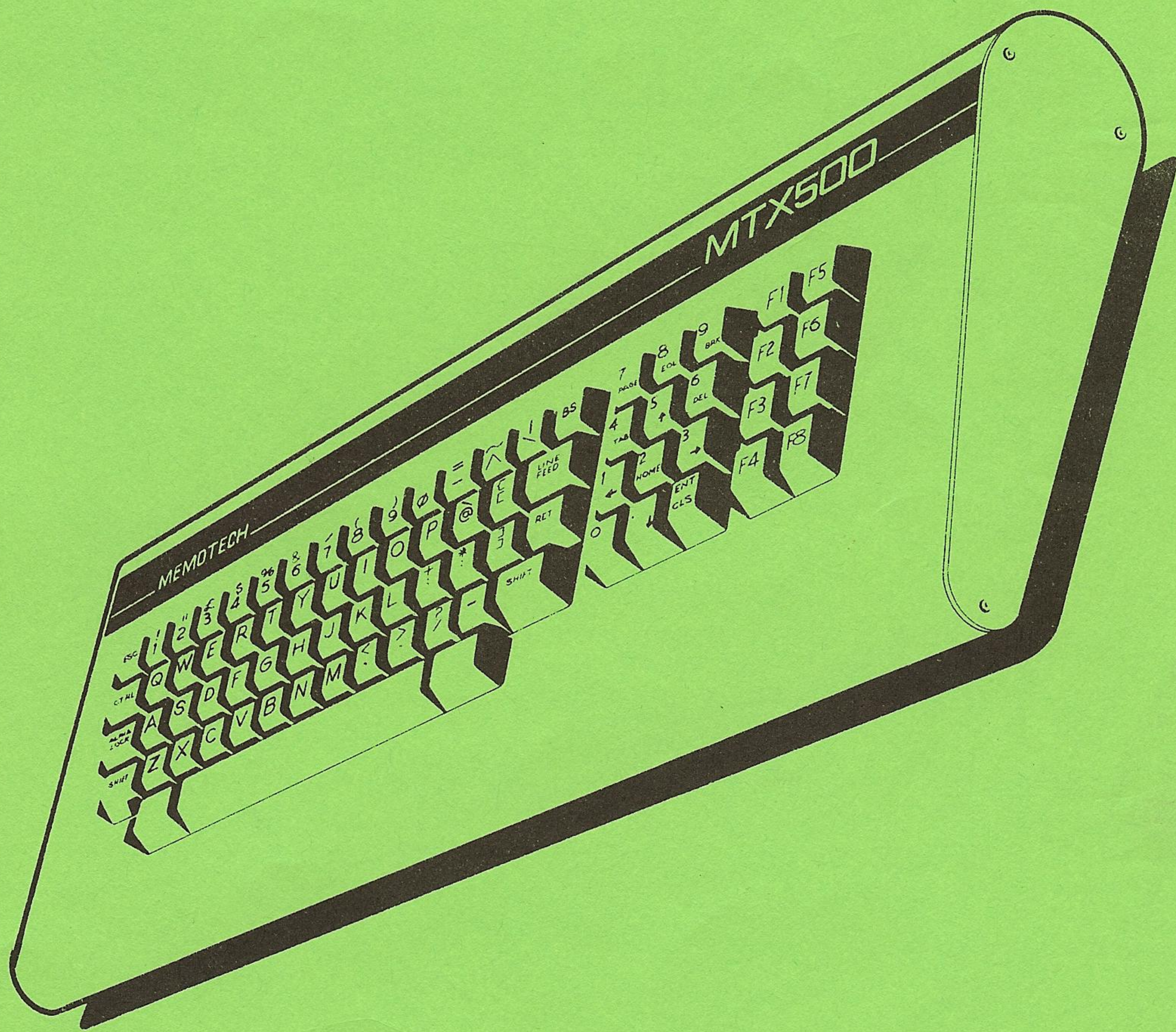


memorad

Memotech Computer User Club Magazine



MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

EDITORIAL

Here we are again ! First, I must apologise for the delay in sending out Issue One: we didn't expect such a tremendous response to the User Club, and at one stage, to say we were on the verge of panic is an understatement !

However, we have been working like mad behind the scenes. All at Genpat, and Memotech, want this to be one of the best user groups around. I visited the factory last week, and I can tell you that we have in the pipe-line, and for exclusive release through the club, the following: **Light Pen, Speech Synthesiser, Modem, and a Cheepo Disc System.** Further news of these add-ons will be announced as they become available.

Between the pages of this edition you find all sorts of **goodies**. Those of you who have been slaving over a hot computer, trying to perfect a **Pixel Scroll** will be pleased to know that your prayers are answered ! We have an excellent one which is fully documented, and can easily be interfaced with your own programs. Also, when you have nothing better to do, type in the **MTX CLOCK....** it's deadly accurate.

We have available, a limited supply of Peter Goode's excellent book, **The Memotech MTX Program Book** at a special member price of £4.75 including P.P. The new **MTX Manual** will be available shortly at £5.50, inclusive, and any members requiring one can book their order now by sending a cheque to Genpat. The new manual includes the much sort after book by Spencer Bateson: **Advanced Programming with the Memotech.**

Don't forget that all **Continental Software** is available at a discount price of 15%, and any orders should be addressed to Genpat - quoting your membership number.

A new software house, **Tri-Soft** is about to release a suite of games for the MTX, and you will find selected reviews elsewhere in this magazine. Also, I can report that **Artic** have shown an interest in the MTX, and Mike Johnson, one of their free-lance programmers, is currently working on a new title (Mike is also a member of the Club). So you see, people are starting to realise, the MTX is one of the best computers on the market.

Finally, I would like to thank all those members who have shared their knowledge with us, and have contributed to this edition. Keep those programs and articles coming in - we need them !

Keith Hook.
Editor.

If you send a letter, and require a reply, PLEASE enclose a S.A.E

Programming in Pascal

```

PROGRAM TEST;
{ MTX graphic extension package }
VAR C: INTEGER;

{ START OF GRAPHICS PROCEDURES }

PROCEDURE RST10(N:1..12;DATA:ARRAY[1..12] OF CHAR);
VAR MEM,I: INTEGER;
BEGIN
POKE($FA92,CHR(0));POKE($FA92,CHR($F0));
MEM:=$F000;
POKE(MEM,CHR($DD));POKE(MEM+1,CHR($E5));POKE(MEM+2,CHR($D7));
DATA[1]:=CHR(ORD(DATA[1])+$80);
FOR I:=1 TO N DO
POKE(MEM+2+I,DATA[I]);
POKE(MEM+2+I,CHR($DD));POKE(MEM+3+I,CHR($E1));POKE(MEM+4+I,CHR(201));
USER(MEM)
END;

PROCEDURE csr(x:0..39;y:0..23);
VAR N:1..12; DATA:ARRAY[1..12] OF CHAR;
BEGIN
N:=4;DATA[1]:=CHR(3);DATA[2]:=CHR(3);DATA[3]:=CHR(X);DATA[4]:=CHR(Y);
RST10(N,DATA)
END;

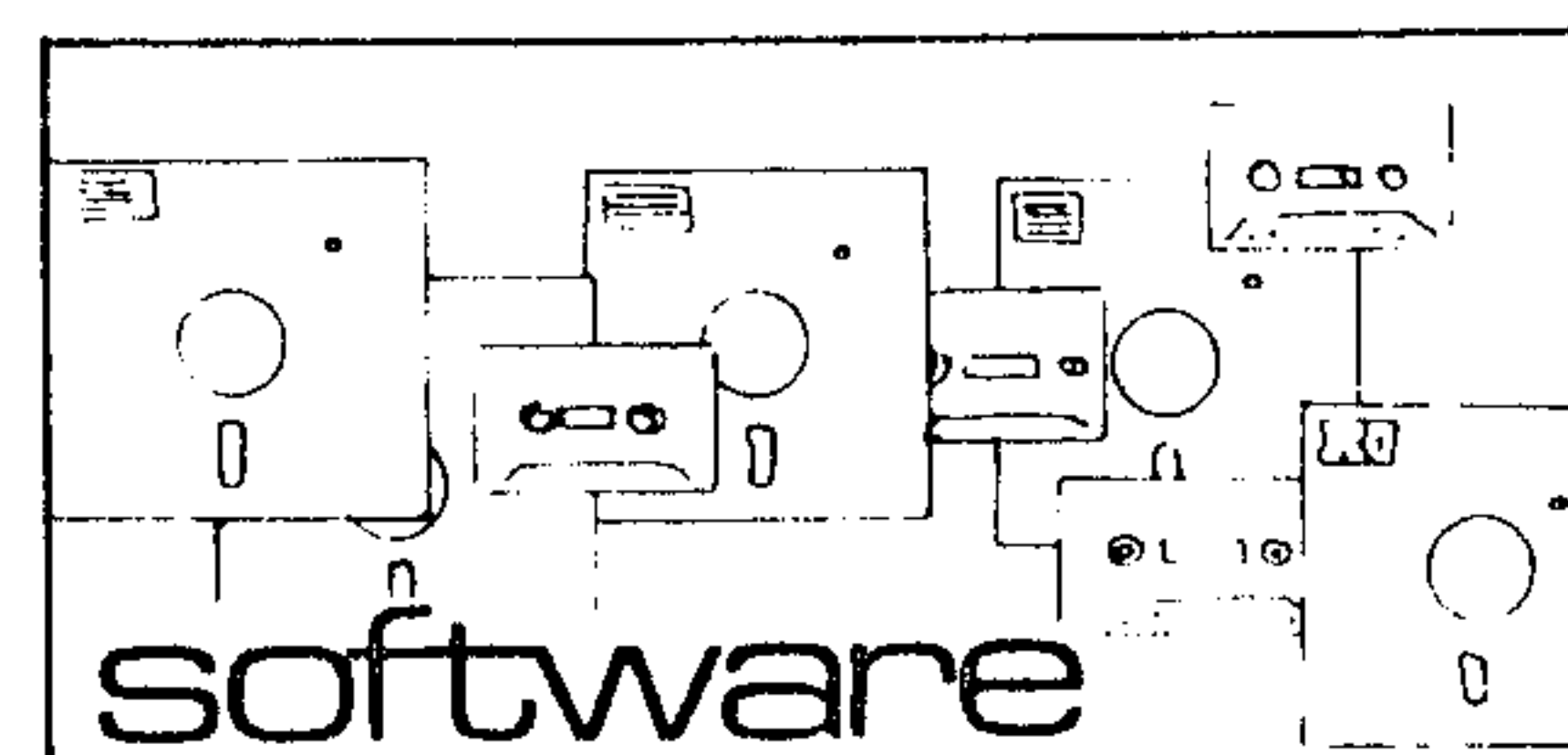
PROCEDURE COLOUR(P:0..4;N:0..15);
VAR C:1..12; DATA:ARRAY[1..12] OF CHAR;
BEGIN
C:=4;DATA[1]:=CHR(3);DATA[2]:=CHR(16);DATA[3]:=CHR(P);DATA[4]:=CHR(N);
RST10(C,DATA)
PROCEDURE ATTR(P:0..3;STATE:0..1);
VAR C: INTEGER; DATA:ARRAY[1..12] OF CHAR;
BEGIN
C:=5;DATA[1]:=CHR(4);DATA[2]:=CHR(27);DATA[3]:=CHR(65);DATA[4]:=CHR(P);
DATA[5]:=CHR(STATE); RST10(C,DATA)
END;
END;

PROCEDURE GR(X:0..255;Y:0..191;B:1..8;VAR CH:CHAR);
VAR C: INTEGER; DATA:ARRAY[1..12] OF CHAR;
BEGIN
C:=6;DATA[1]:=CHR(5);DATA[2]:=CHR(27);DATA[3]:=CHR(67);DATA[4]:=CHR(X);
DATA[5]:=CHR(Y);DATA[6]:=CHR(B);
RST10(C,DATA);
CH:=PEEK($FE1A,CHAR)
END;

{ END OF GRAPHICS ROUTINES }
BEGIN
VS(4);PAGE;
COLOUR(4,1);
FOR C:=1 TO 87 DO
BEGIN
COLOUR(3,C DIV 16);
LINE(0,C*2,255,C*2)
END;
CSR(6,2);WRITELN('MTX PASCAL GRAPHICS');
CSR(5,10);WRITELN('MTX PASCAL GRAPHICS');
CSR(6,10);ATTR(1,1);WRITELN('MTX PASCAL GRAPHICS');
CSR(0,22)
END.

```

BRUNWORD Brunning Software.



This is an excellent software package. For the enthusiast who cannot afford the ROM based word processor, **BRUNWORD** is the answer.

I was amazed how much power the programmer has managed to cram into the tape package. Normally, tape based word processors, are little better than hooking the printer up to the machine and writing a 5 line program around the Inkey\$ function - not so with **Brunword**. Obviously, there have been a few compromises, but some of the functions I found easier to use, and better, in some respects, to the one I am using to write the magazine.

The program supports commands such as **SAVE : LOAD : DIRECTORY: REMOVE : CLEAR : ON SCREEN : GET : MEMORY** and a host of others. Surprising as it may seem, **Brunword** also supports **TEAR & WEAVE** which allows you to pull a file to pieces and re-assemble it in a different order.

The printer is fully supported with **enlarged, condensed, underline, and emphasized** characters.

A neat little twist in the package is the way Brunning have utilised a compressed format for storing files in memory - which gives you the advantage of being able to store more text than the average user would require at one sitting.

Files are stored on tape and can be recalled using the **Load** command - even damaged files are not altogether lost, and the instructions cover the subject adequately.

This is a super program, I was delighted to see that quality software is available for the MTX.

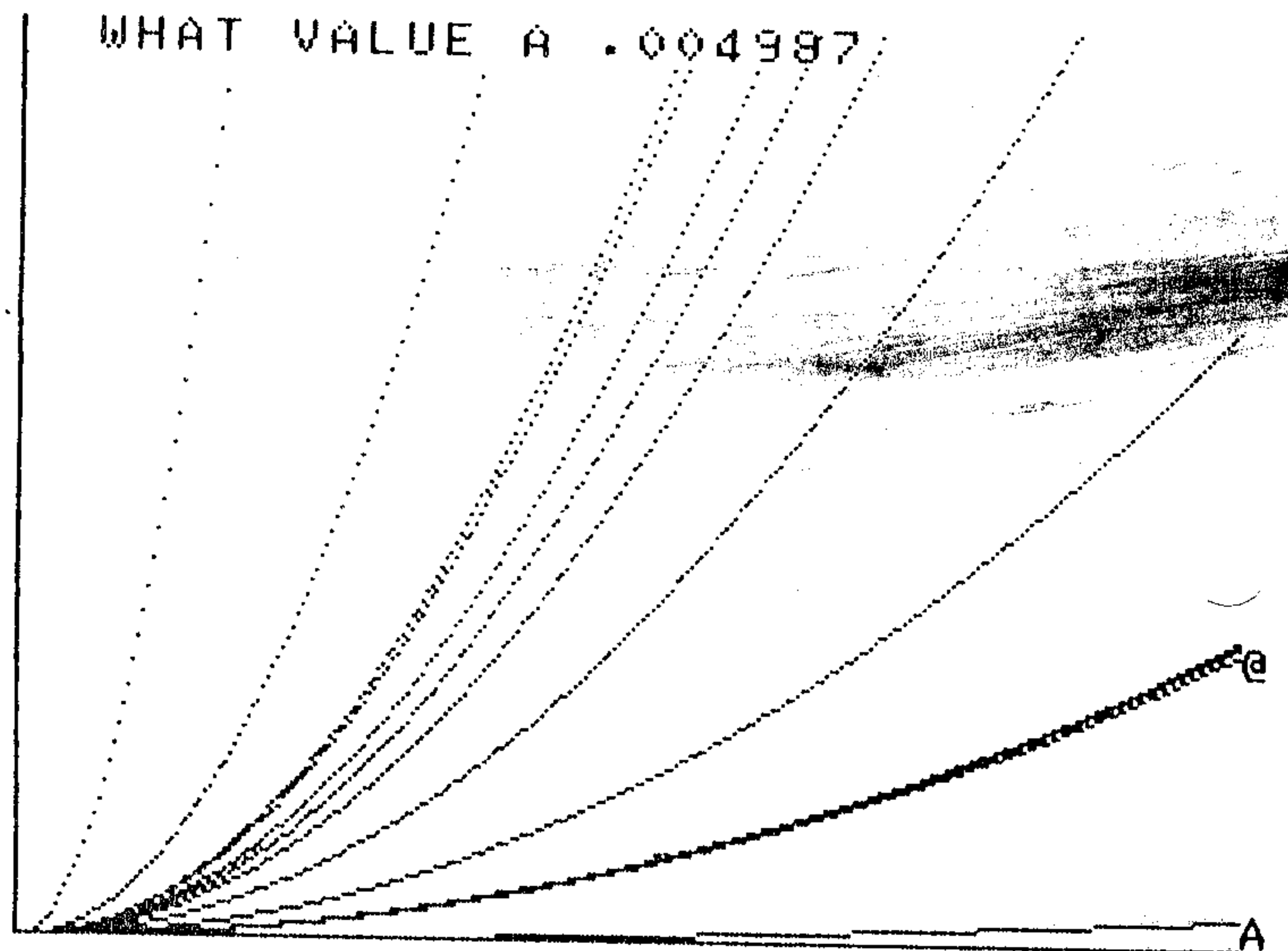
Brunword is offered to members at 15% discount which reduces the normal price from £19.50 to £16.57. Cash with order stating membership number to: **Brunning Software, 34, Helston Road, Chelmsford, Essex. 0245 252854**



Here's some little graphic routines for you to play with..... try altering the COS/SIN values. Infact, try altering any value.

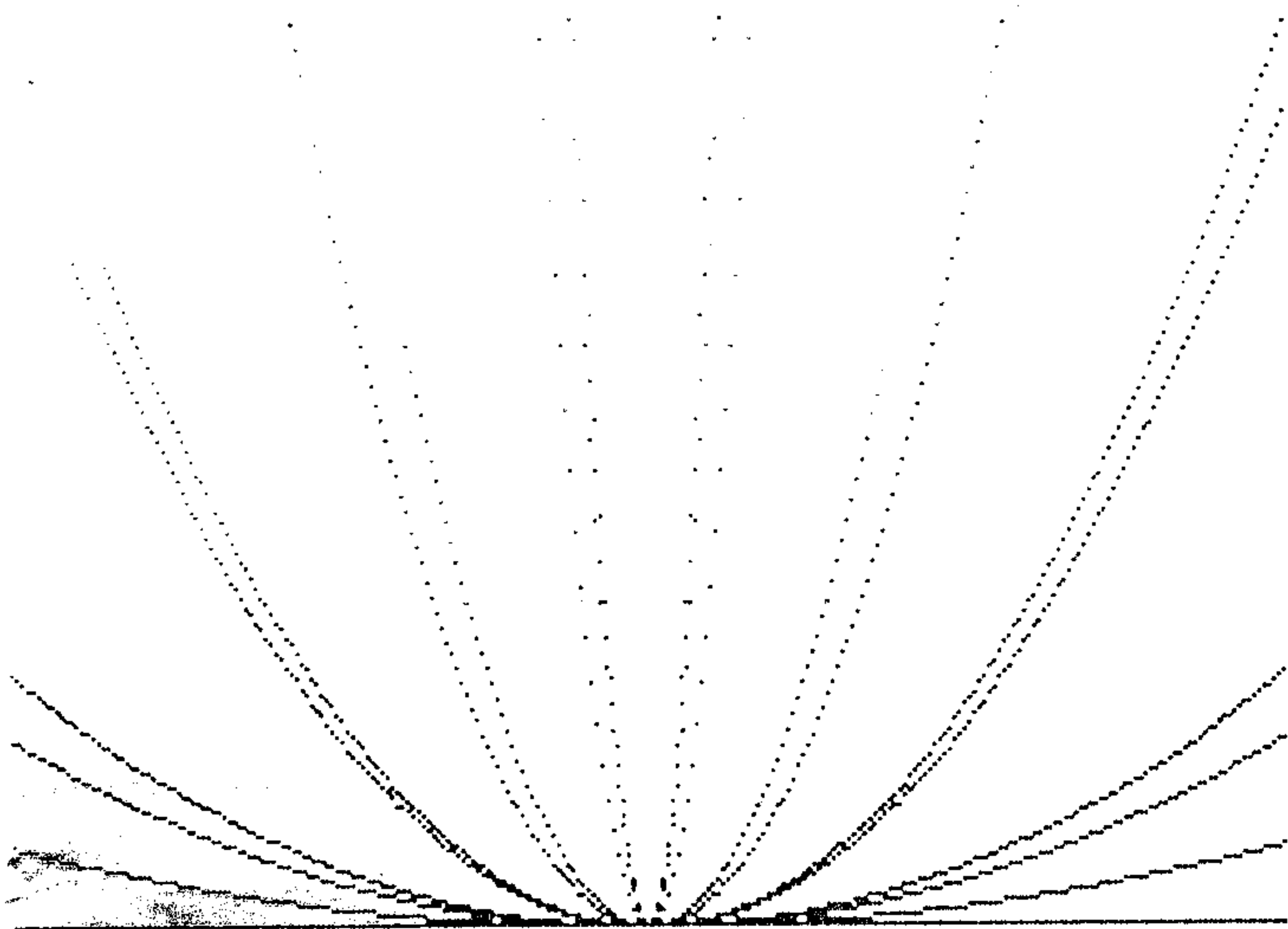
```

10 VS 4
20 COLOUR 4,0: COLOUR 2,3: COLOUR 0,3: CLS
30 INK 15: FOR X=0 TO 255: LET Y=0: PLOT X,Y: NEXT : FOR Y=0 TO 191
33 FOR Y=0 TO 191: LET X=0: PLOT X,Y: NEXT
35 INK 1: CSR 2,0: INPUT "WHAT VALUE A ";A
40 FOR X=0 TO 255
50 LET Y=INT(0+A*X*X)
60 IF Y>191 THEN GOTO 100
70 PLOT X,Y
80 NEXT X
100 GOTO 35
    
```



CURVE SCREEN

Value of a ? .05

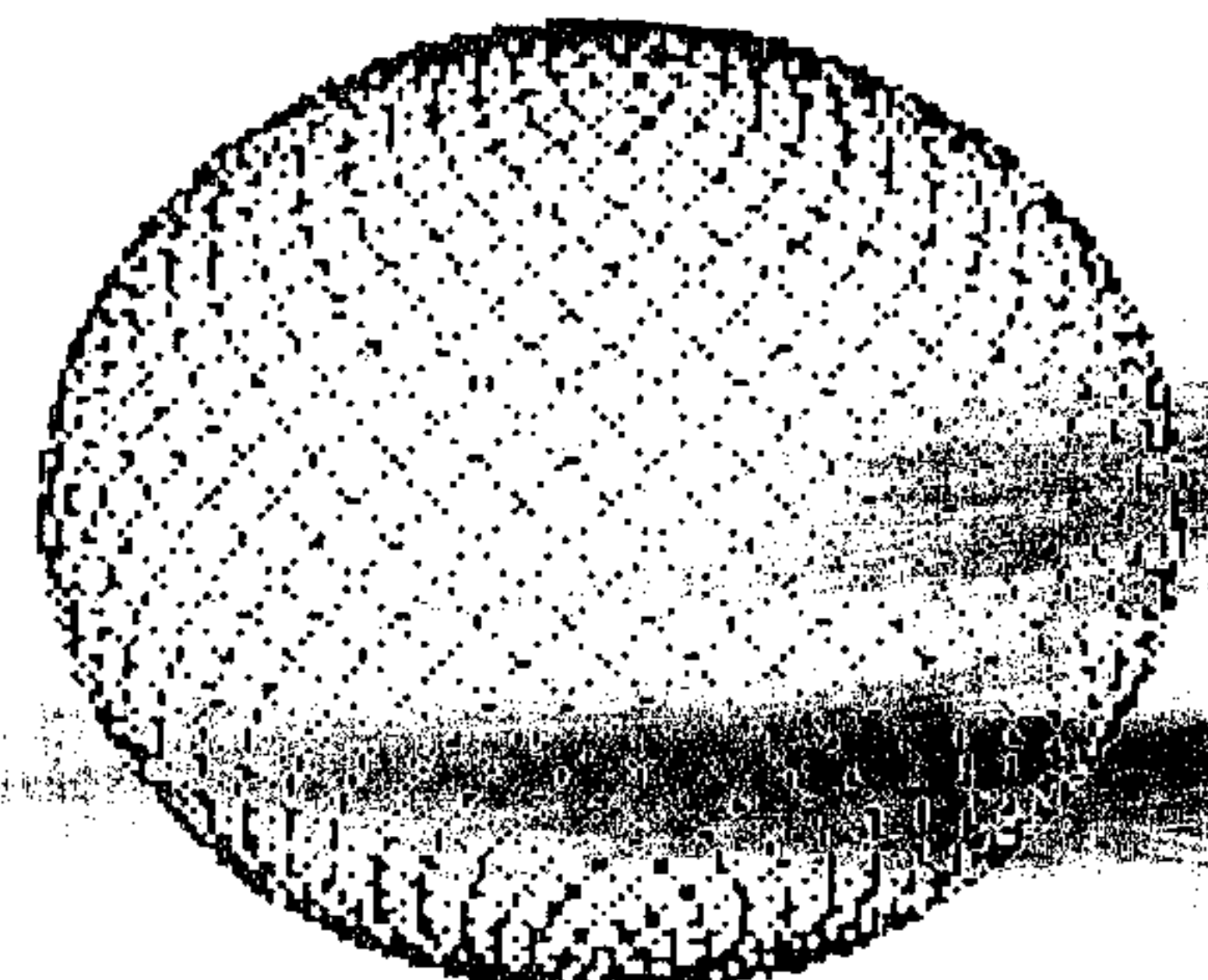


```

10 VS 4
20 CLS : INK 15
50 LET Y=0
60 FOR X=0 TO 254
70 PLOT X,Y
80 NEXT X: INK 1
85 CSR 2,0: PRINT CHR$(5);: INPUT " Value of a ? ";A;
90 FOR X=0 TO 127
100 LET Y=INT(A*X*X)
110 IF Y>191 THEN GOTO 150
120 PLOT (127-X),Y
130 PLOT (127+X),Y
140 NEXT
150 GOTO 85
    
```

```

10 VS 4: CLS
20 FOR A=0 TO 125.7 STEP .03
30 PLOT 128+(55*SIN(A)),96+(65*COS(A)*SIN(A*.95))
40 NEXT
    
```



SYNTAX SOFTWARE

Programmer's Corner

This is your very own software label. Syntax Software requires first class games, and utility programs to market. We will guarantee any author who is signed by the club, a minimum of £1.00 per sale in the EEC. Send your software for evaluation to Genpat.

Club members will be offered special prices on the club's label. We aim to capture quality programs and offer them a reasonable price.

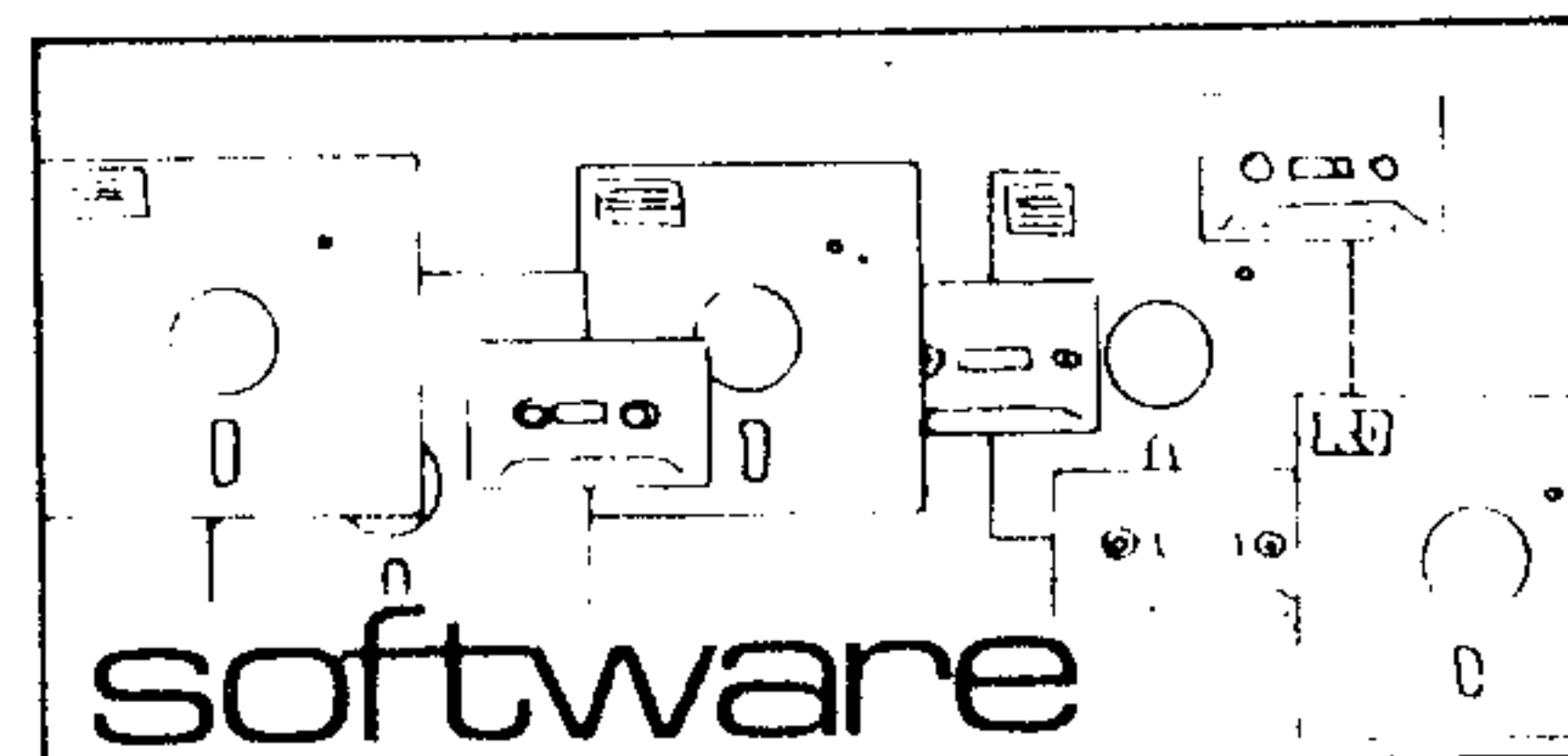
Our very first program will be launched at the end of the month. MTX EDASM this is a taped based Macro Editor Assembler that supports EQU & ORG. With the ORG statement you can re-load those routines anywhere in memory without having to use a loader program. MTX EDASM is not hard-wired which allows you to re-define the macro instruction set..... and, if you so desire, assemble in 6502 code or, in fact, your own code !!!! Price to Genpat Members £7.95 to non members £ 12.75.

Program comes with extensive user manual. *** Available from 15th November.

COMPETITION

You have all seen Continental's title screen. Well. For the best designed screen which will be used with Syntax Software's programs we will award 4 pieces of software. Send your entries on tape to Genpat to arrive not later than 15th November. Let's have some really original ideas !!!!

USER EXTEND Membrain Software.



This is a very useful utility program by Membrain. The program adds various useful routines to your toolkit.

User Csave saves code from parameters given in <start> to <start>+<length>. User Cload complements the latter command. Verify is also supported.

User Free will print amount of remaining usable memory.

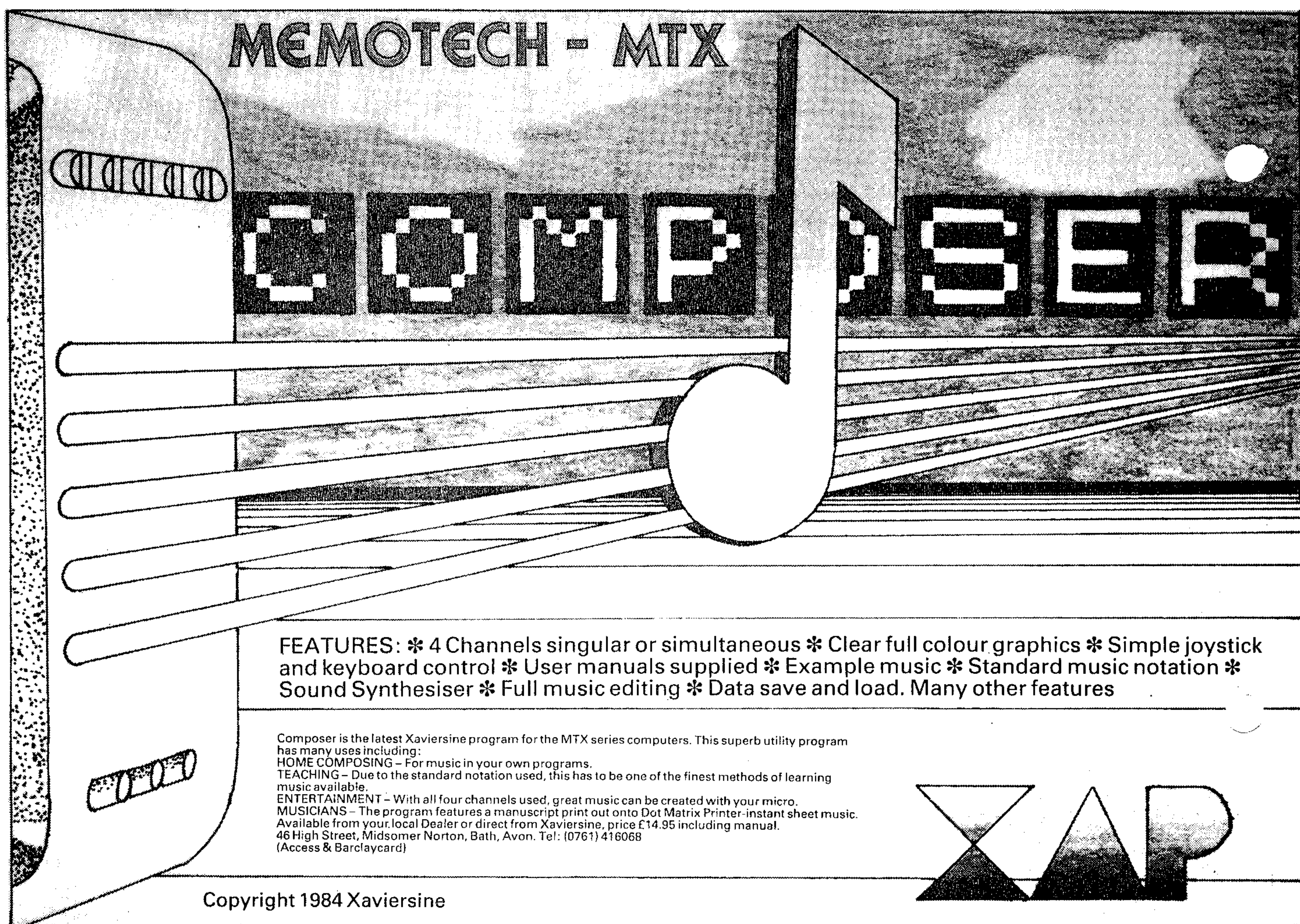
The screen dump routines are included in this utility so you actually get two programs in one.

Other commands supported are: User & Decimal : User £ Decimal : User Clone : User Fill : User Stat : User Pad.

MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

A very neat touch is the USER DOKE wich allows you to poke 16 bits in one go.... this takes the sweat out of calculating LSB MSB.

More information on this program from Membrain£9.50p



MEMOTECH - MTX

COMPOSER

FEATURES: * 4 Channels singular or simultaneous * Clear full colour graphics * Simple joystick and keyboard control * User manuals supplied * Example music * Standard music notation * Sound Synthesiser * Full music editing * Data save and load. Many other features

Composer is the latest Xaversine program for the MTX series computers. This superb utility program has many uses including:
 HOME COMPOSING - For music in your own programs.
 TEACHING - Due to the standard notation used, this has to be one of the finest methods of learning music available.
 ENTERTAINMENT - With all four channels used, great music can be created with your micro.
 MUSICIANS - The program features a manuscript print out onto Dot Matrix Printer-instant sheet music.
 Available from your local Dealer or direct from Xaversine, price £14.95 including manual.
 46 High Street, Midsomer Norton, Bath, Avon. Tel: (0761) 416068
 (Access & Barclaycard)

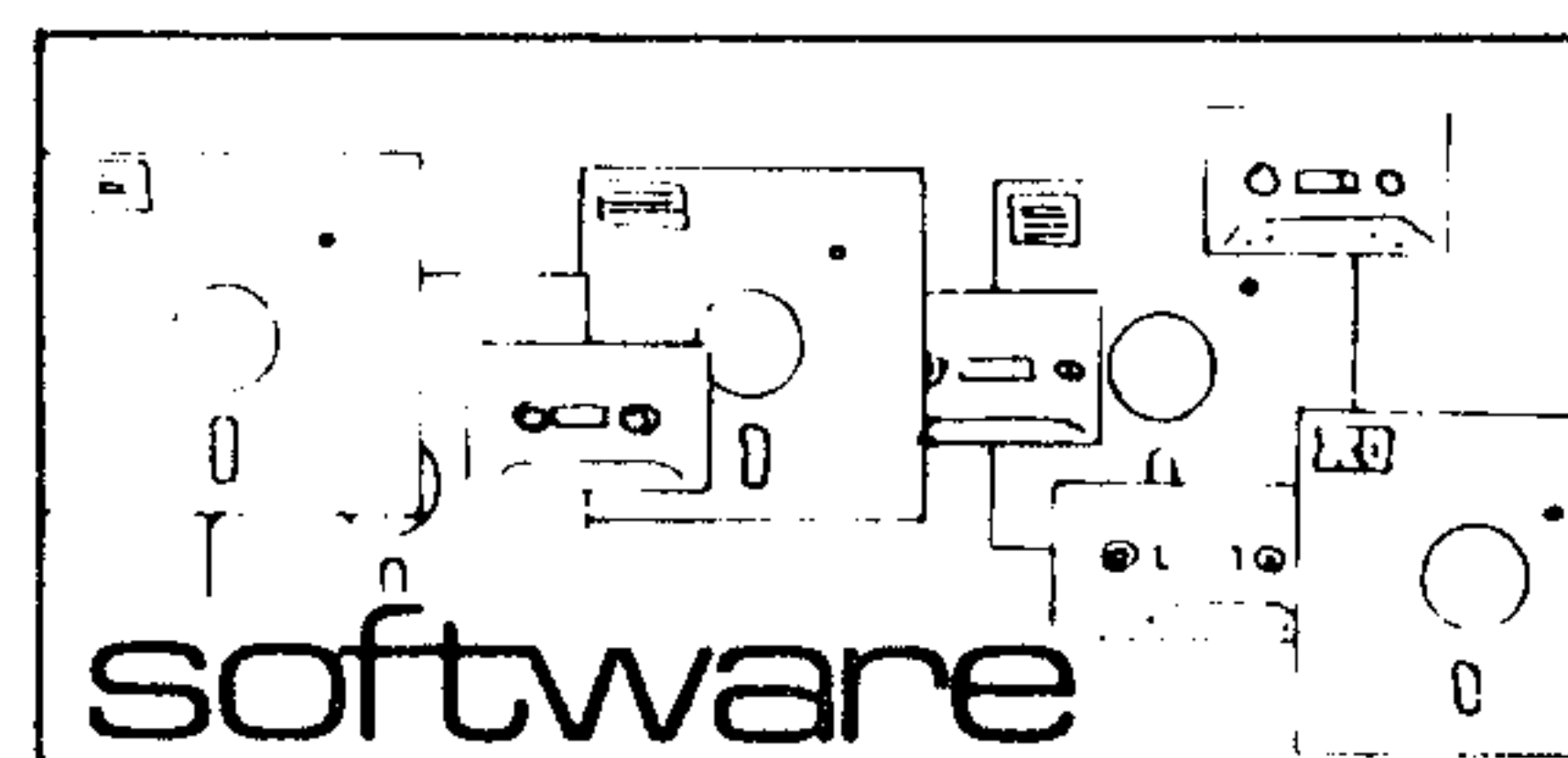
XAP

Copyright 1984 Xaversine

PROGRAM

```
5 VS 4: LET IN=4: COLOUR 2,0: COLOUR 4,0: CLS
10 FOR I=1 TO 97 STEP .0275
20 PLOT 128+(1*I*SIN(I)),96+(1*I*COS(I))
30 LET IN=IN+1: IF IN>15 THEN LET IN=2
40 INK IN
50 NEXT
```

QOGO **CONTINENTAL SOFTWARE**



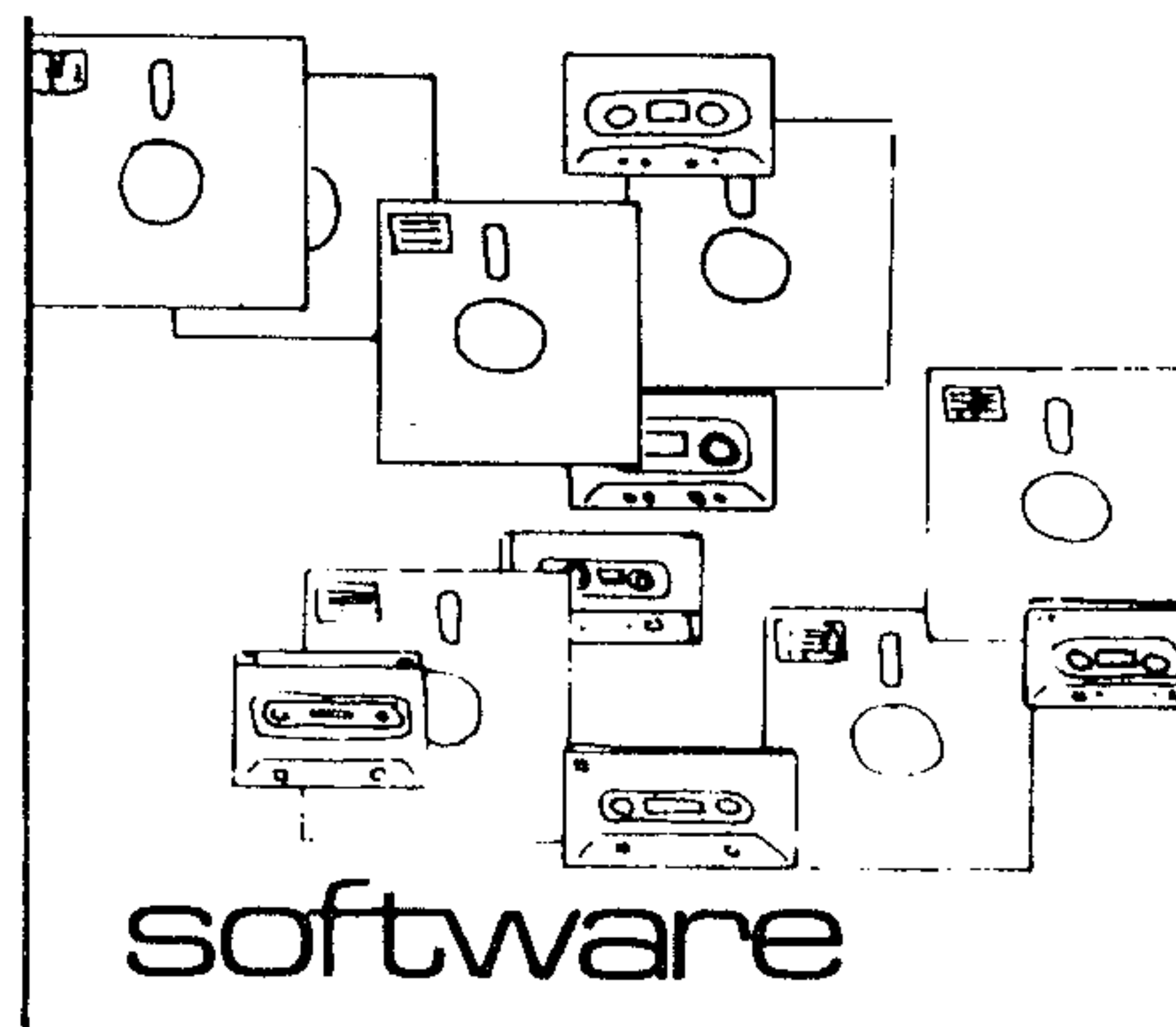
Qogo is a variation on the arcade game **QBERT**. I suppose it is inevitable that 90% of computer games will have been spawned in the arcades.

I was pleasantly surprised when I loaded this game into the MTX - the graphics are reasonable, and the game faithfully follows the original.

For the uninitiated, the idea is to jump your man over blocks constructed in the shape of a pyramid. Whenever your man touches a block it changes colour. If you successfully change all the blocks you enter a new screen. Bouncing balls, spinning blades, and the like, all try to do you down.

I can imagine that this game won't appeal to everyone, but I can recommend it.

POTHOLE PETE **CONTINENTAL SOFTWARE**



What can I say? If you want the **best** game that is available for the MTX, then this is it !!

If you are in doubt, ask your dealer to load it into the computer. As soon as the loading sequence has finished you are treated to a preview of the different screens - these are many, and varied, some you may never manage to reach.

Although the game is similar in concept to Manic Miner, the finished product is completely original.

Absolutely fantastic..... we need more programs of this quality !!

The above games are available from GENPAT at £6.02 inc P.P
Cash with order to GENPAT & please allow 7 days for delivery

***** THE ONE POUND RAFFLE *****

It has been suggested that we run a raffle. I think it is a good idea, but I'll leave the final judgement to you.

The idea is: you send a £1 note into Memopad with your **Membership Number** written on the top. What is to be raffled will depend on how many members participate - some months it could be a **Pascal ROM** or a **RS232 expansion**..... maybe even a printer, or a monitor !! It could even be a **Disc Drive**.

If the winning member already has the prize, we will be flexible enough to allow him, or her to swap it for something of similar value.

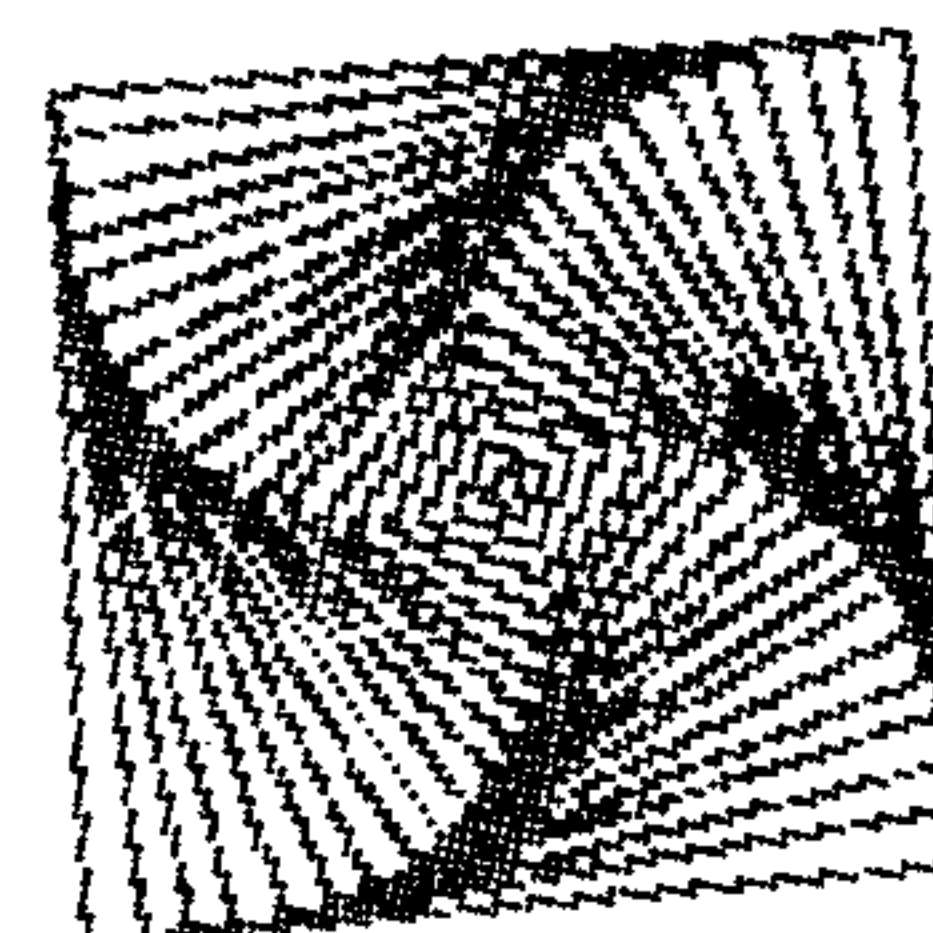
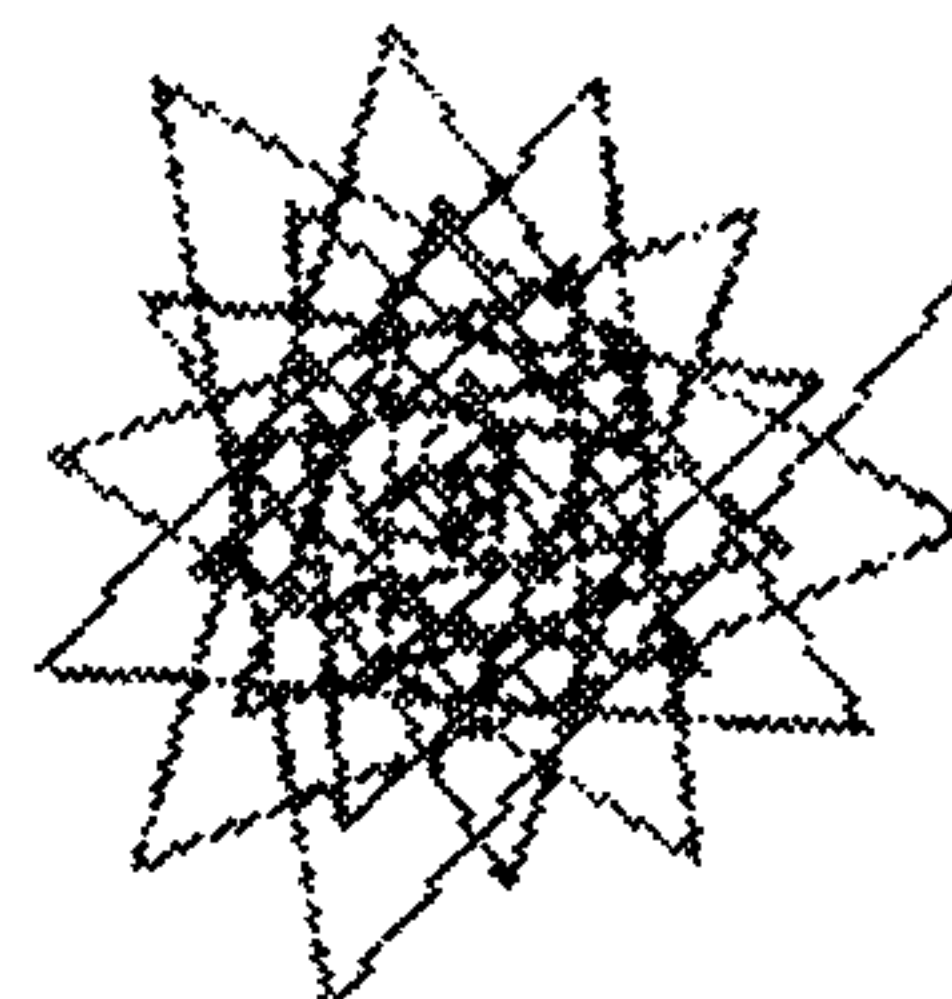
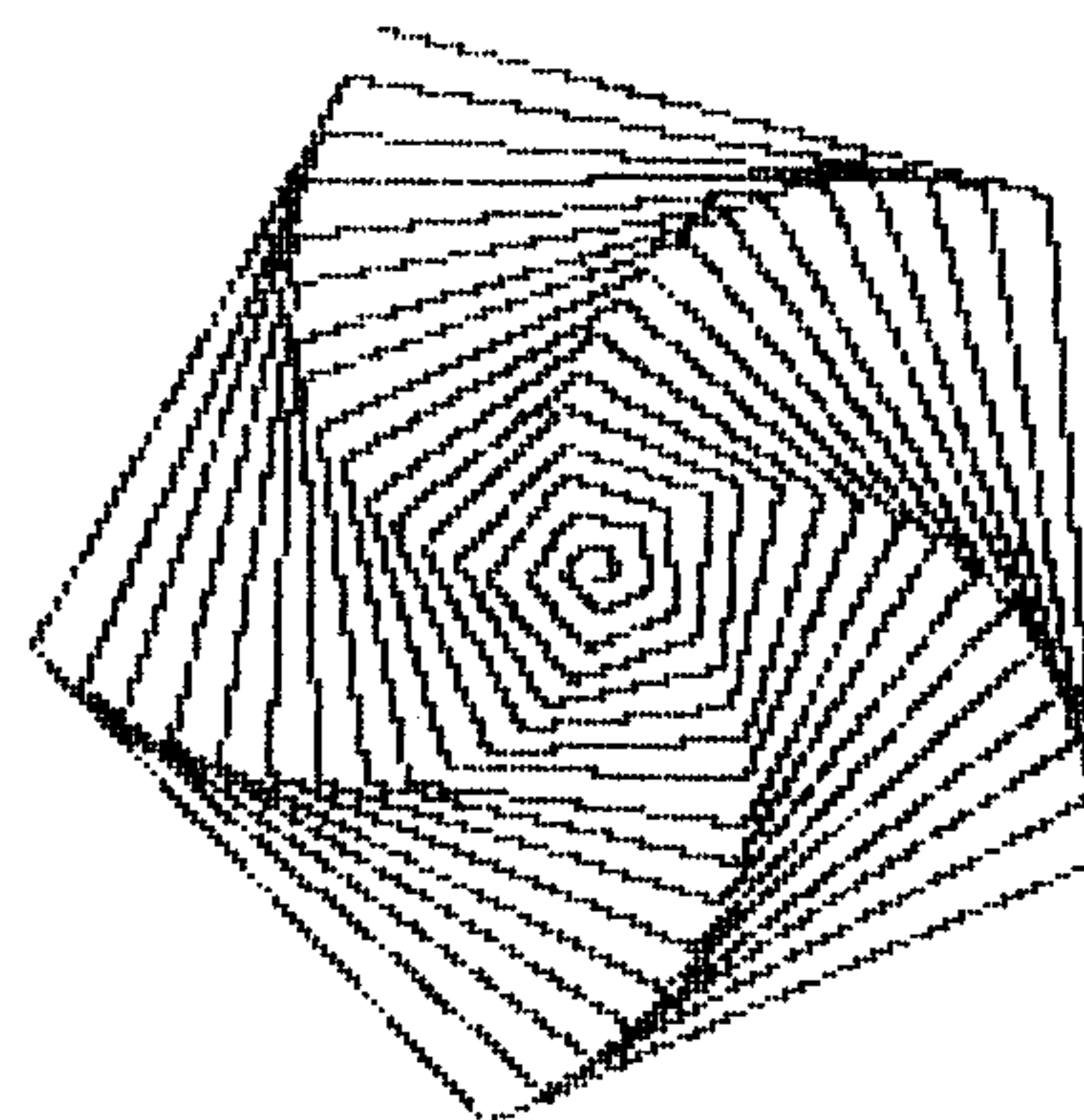
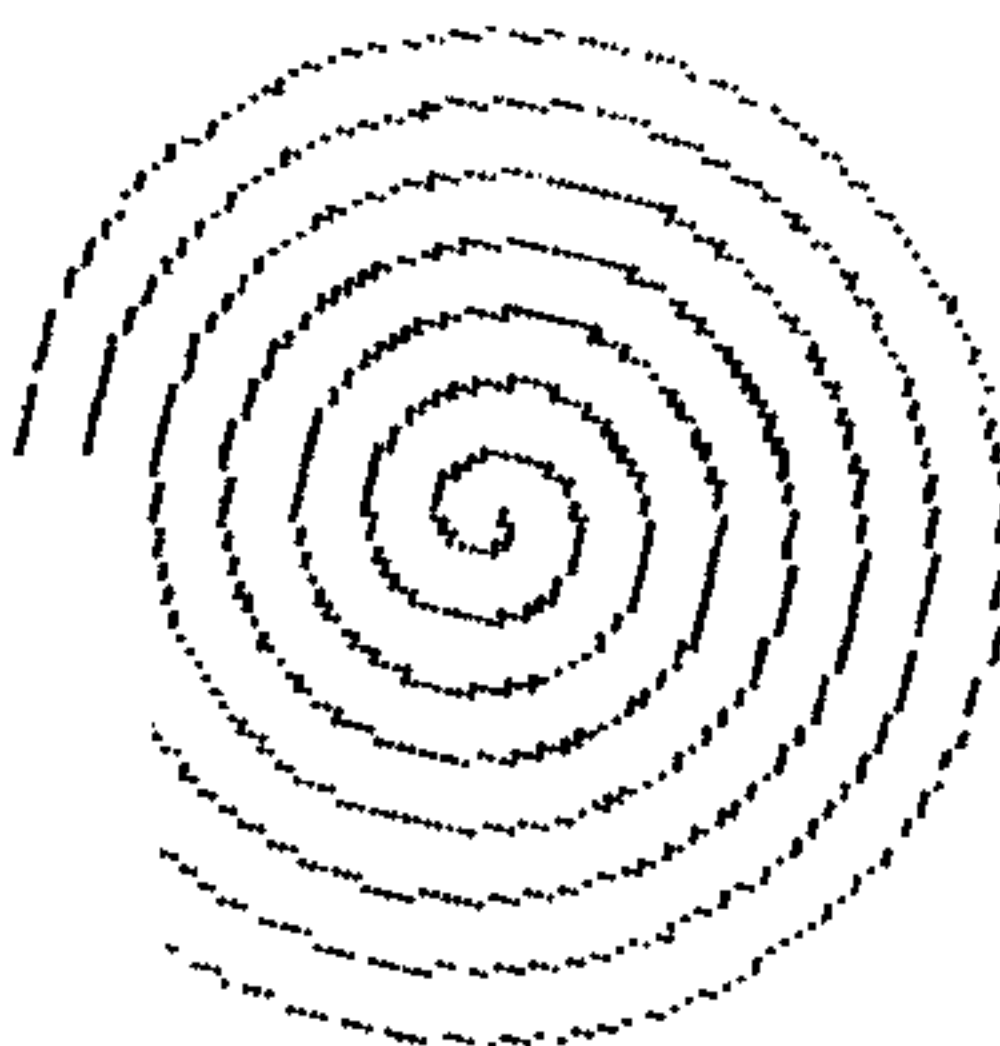
It's up to you. If you like the idea, send a £1 to reach us by the 15th of November, and the winner will be notified in that month's edition of the magazine.

Don't worry, if only a few of you like the idea, we will send the winners vouchers to spend on software.

Because the first edition was late in being published we have stretched the competition for the best program into this month so you still have a chance of winning the two pieces of software on offer.

PROGRAM K-HOOK 83

```
10 VS 4
20 COLOUR 2,1: CLS
30 INK 1: CSR 2,0: INPUT "DIST-ANGLE-INC ?";DIS,AN,INC
40 CLS
50 LET P=PI/180: LET X=128: LET Y=96: LET AN1=AN
60 LET X1=X+DIS*COS(P*AN1)
70 LET Y1=Y+DIS*SIN(P*AN1)
80 IF Y1>190 OR Y1<0 OR X1>250 OR X1<0 THEN GOTO 30
90 LET DIS=DIS+INC
100 INK 2: LINE X,Y,X1,Y1
110 LET AN1=AN1+AN
120 LET X=X1: LET Y=Y1: GOTO 60
```



PROGRAM

David Wemyss from Strathkinness has sent in the following program. David says that this is the first program he has shown to any other person. I think he has done a wonderful job, don't you? I look forward to more contributions from David.

```

1 REM SUMS FOR THE MEMOTECH
2 REM *****
3 REM DAVID WEMYSS 1984
4 REM *****
5 VS 4: CLS : COLOUR 2,11: COLOUR 0,11: COLOUR 1,6: COLOUR 4,11
10 FOR J=1 TO 10
20 ATTR 0,1: GOSUB 9000
30 PAUSE 200: ATTR 0,0
40 GOSUB 9000
50 PAUSE 200
60 NEXT
70 PAUSE 1000
75 CLS
80 CSR 2,5: INPUT "WHAT IS YOUR NAME? ";N$
83 IF LEN(N$)>10 THEN GOTO 75
90 PRINT : PRINT
95 CLS
100 CSR 5,8: PRINT "Welcome ";N$;" ."
101 GOTO 105
103 CLS : CSR 5,8: PRINT "Welcome back, ";N$
105 RAND -1000
106 CSR 3,10: PRINT "
110 CSR 3,10: PRINT "The choice of sums follows!"
120 PAUSE 2000: CSR 3,12: PRINT "Best of luck!": PAUSE 2000
130 PLOD "PRO61"
140 PAPER 11: INK 1: LET CHOICE$=INKEY$: IF CHOICE$="" THEN GOTO 140
150 LET CHOICE=VAL(CHOICE$): IF CHOICE<1 OR CHOICE>4 THEN CLS : CSR 3,5: PRINT "Incorrect choice - 1 to 4 only!": PAUSE 2000: GOTO 130
160 ON CHOICE-1 GOTO 1000,2000,3000,4000
1000 CLS : PAPER 10: INK 6
1010 CSR 11,5: PRINT "ADDITION SUMS"
1020 CSR 11,6: PRINT "=====
1030 PAUSE 4000
1040 PLOD "PRO62"
1050 GOSUB 12000
1090 ON CHOICE-1 GOTO 1100,1200,1300
1100 CLS : PAPER 10: INK 6: IF M)=11 THEN GOTO 14000
1105 CSR 15,0: PRINT "LEVEL - 1": CSR 15,1: PRINT "=====
1115 LET A=INT(RND*20+1): IF A>20 THEN LET A=20
1120 LET B=INT(RND*20+1): IF B>20 THEN LET B=20
1125 GOSUB 11000
1130 CSR 10,10: PRINT A;" + ";B;" = ? "
1135 LET T=A+B
1140 GOSUB 11050
1150 PAUSE 2500: GOTO 1100
1200 CLS : PAPER 10: INK 6: IF M)=11 THEN GOTO 14000
1205 CSR 15,0: PRINT "LEVEL - 2": CSR 15,1: PRINT "=====
1215 LET A=INT(RND*100+1): IF A>100 THEN LET A=100
1220 LET B=INT(RND*100+1): IF B>100 THEN LET B=100
1225 GOSUB 11000
1230 CSR 10,10: PRINT A;" + ";B;" = ? "
1235 LET T=A+B
1240 GOSUB 11050
1245 PAUSE 2500: GOTO 1200
1300 CLS : PAPER 10: INK 6: IF M)=11 THEN GOTO 14000
1305 CSR 15,0: PRINT "LEVEL - 3": CSR 15,1: PRINT "=====
1310 LET A=INT(RND*1000+1): IF A>1000 THEN LET A=1000
1315 LET B=INT(RND*1000+1): IF B>1000 THEN LET B=1000
1320 GOSUB 11000
1325 CSR 10,10: PRINT A;" + ";B;" = ? "
1330 LET T=A+B
1340 GOSUB 11050
1350 PAUSE 2500: GOTO 1300
2000 CLS : PAPER 6: INK 10
2010 CSR 9,5: PRINT "SUBTRACTION SUMS": CSR 9,6: PRINT "=====
2020 PAUSE 4000: PLOD "PRO62"
2030 GOSUB 12000
2080 ON CHOICE-1 GOTO 2100,2200,2300
2100 CLS : PAPER 6: INK 10
2105 IF M)=11 THEN GOTO 14000
2110 CSR 15,0: PRINT "Level - 1": CSR 15,1: PRINT "=====
2115 LET A=INT(RND*20+1): IF A>20 THEN LET A=20
2120 LET B=INT(RND*20+1): IF B>20 THEN LET B=20
2122 IF A<B THEN GOTO 2115
2125 GOSUB 11000
2130 CSR 10,10: PRINT A;" - ";B;" = ? "
2135 LET T=A-B
2140 GOSUB 11050
2150 PAUSE 2500: GOTO 2100
2200 CLS : PAPER 6: INK 10: IF M)=11 THEN GOTO 14000
2210 CSR 15,0: PRINT "Level - 2": CSR 15,1: PRINT "=====
2220 LET A=INT(RND*100+1): IF A>100 THEN LET A=100
2230 LET B=INT(RND*100+1): IF B>100 THEN LET B=100
2240 IF A<B THEN GOTO 2220
2250 GOSUB 11000
2260 LET T=A-B
2270 CSR 10,10: PRINT A;" - ";B;" = ? "
2280 GOSUB 11050
2290 PAUSE 2500: GOTO 2200
2300 CLS : PAPER 6: INK 10: IF M)=11 THEN GOTO 14000
2310 CSR 15,0: PRINT "Level - 3": CSR 15,1: PRINT "=====
2320 LET A=INT(RND*1000+1): IF A>1000 THEN LET A=1000
2330 LET B=INT(RND*1000+1): IF B>1000 THEN LET B=1000
2340 IF A<B THEN GOTO 2320
2350 LET T=A-B
2360 GOSUB 11000
2370 CSR 10,10: PRINT A;" - ";B;" = ? "
2380 GOSUB 11050
2390 PAUSE 2500: GOTO 2300
3000 CLS : PAPER 15: INK 1
3010 CSR 7,5: PRINT "MULTIPLICATION SUMS": CSR 7,6: PRINT "=====
3020 PAUSE 4000: PLOD "PRO62"
3030 GOSUB 12000
3040 ON CHOICE-1 GOTO 3100,3200,3300
3100 CLS : PAPER 15: INK 1: IF M)=11 THEN GOTO 14000
3110 CSR 15,0: PRINT "Level - 1": CSR 15,1: PRINT "=====
3120 LET A=INT(RND*20+1): IF A>20 THEN LET A=20
3130 LET B=INT(RND*10+1): IF B>10 THEN LET B=10
3140 LET T=A*B
3150 GOSUB 11000
3160 CSR 10,10: PRINT A;" * ";B;" = ? "
3170 GOSUB 11050

```


MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

```

3180 PAUSE 2500: GOTO 3100
3200 CLS : PAPER 15: INK 1: IF M)=11 THEN GOTO 14000
3210 CSR 15,0: PRINT "Level - 2": CSR 15,1: PRINT "=====
3220 LET A=INT(RND*100+1): IF A>100 THEN LET A=100
3230 LET B=INT(RND*20+1): IF B>20 THEN LET B=20
3240 LET T=A*B
3250 GOSUB 11000
3260 CSR 10,10: PRINT A;" * ";B;" = ?"
3270 GOSUB 11050
3280 PAUSE 2500: GOTO 3200
3300 CLS : PAPER 15: INK 1: IF M)=11 THEN GOTO 14000
3310 CSR 15,0: PRINT "Level - 3": CSR 15,1: PRINT "=====
3320 LET A=INT(RND*1000+1): IF A>1000 THEN LET A=1000
3330 LET B=INT(RND*100+1): IF B>100 THEN LET B=100
3340 LET T=A*B
3350 GOSUB 11000
3360 CSR 10,10: PRINT A;" * ";B;" = ?"
3370 GOSUB 11050
3380 PAUSE 2500: GOTO 3300
4000 CLS : PAPER 11: INK 4
4010 CSR 10,5: PRINT "DIVISION SUMS": CSR 10,6: PRINT "=====
4020 PAUSE 4000: PLOD "PROG2"
4030 GOSUB 12000
4040 ON CHOICE-1 GOTO 4100,4200,4300
4100 CLS : PAPER 11: INK 4: IF M)=11 THEN GOTO 14000
4110 CSR 15,0: PRINT "Level - 1": CSR 15,1: PRINT "=====
4120 LET A=INT(RND*100+1): IF A>100 THEN LET A=100
4130 LET B=INT(RND*10+1): IF B>10 THEN LET B=10
4140 IF A<B THEN GOTO 4120
4150 LET T=INT(A/B): LET R=A-(T*B)
4160 GOSUB 11000
4170 CSR 5,10: PRINT A;" / ";B;" = ? remainder ?"
4180 GOSUB 13000
4190 PAUSE 2500: GOTO 4100
4200 CLS : PAPER 11: INK 4: IF M)=11 THEN GOTO 14000
4210 CSR 15,0: PRINT "Level - 2": CSR 15,1: PRINT "=====
4220 LET A=INT(RND*1000+1): IF A>1000 THEN LET A=1000
4230 LET B=INT(RND*100+1): IF B>100 THEN LET B=100
4240 IF A<B THEN GOTO 4220
4250 LET T=INT(A/B): LET R=A-(T*B)
4260 GOSUB 11000
4270 CSR 5,10: PRINT A;" / ";B;" = ? remainder ?"
4280 GOSUB 13000
4290 PAUSE 2500: GOTO 4200
4300 CLS : PAPER 11: INK 4: IF M)=11 THEN GOTO 14000
4310 CSR 15,0: PRINT "Level - 3": CSR 15,1: PRINT "=====
4320 LET A=INT(RND*1000+1): IF A>1000 THEN LET A=1000
4330 LET B=INT(RND*1000+1): IF B>1000 THEN LET B=1000
4340 IF A<B THEN GOTO 4320
4350 LET T=INT(A/B): LET R=A-(T*B)
4360 GOSUB 11000
4370 CSR 5,10: PRINT A;" / ";B;" = ? remainder ?"
4380 GOSUB 13000
4390 PAUSE 2500: GOTO 4300
9000 CSR 12,8: PRINT "WELCOME"
9010 CSR 13,10: PRINT "TO THE"
9020 CSR 12,12: PRINT "MEMOTECH"
9030 CSR 9,14: PRINT "SUMS PROGRAMME"
9040 RETURN
11000 CSR 3,5: PRINT "Sum Number ";M: CSR 3,6: PRINT "-----
11010 RETURN
11050 CSR 5,12: PRINT "
11055 CSR 5,12: INPUT "Your answer is ";A$
11060 LET C=VAL(A$): IF C=T THEN LET S=S+1
11065 LET M=M+1
11070 CSR 5,14: PRINT "
11075 IF C=T THEN CSR 5,14: PRINT "Correct"
11080 IF C<>T THEN CSR 5,14: PRINT "Wrong": CSR 5,16: PRINT "It should have been ";T;"!"
11085 RETURN
12000 LET CHOICE%=INKEY$: IF CHOICE%="" THEN GOTO 12000
12010 LET CHOICE=VAL(CHOICE%)
12015 IF CHOICE<1 OR CHOICE>3 THEN CLS : CSR 2,4: PRINT "Incorrect choice - 1 to 3 only": PAUSE 2000: IF CHOICE<1 OR CHOICE>3 THEN GOTO 1040
12020 LET S=0: LET M=1
12030 RETURN
13000 CSR 3,12: PRINT "
13010 CSR 1,12: INPUT "Your answer is ";A$: CSR 21,12: INPUT "remainder ";R$
13020 LET C=VAL(A$): IF C=T AND R=VAL(R$) THEN LET S=S+1
13030 LET M=M+1
13040 CSR 5,14: PRINT " "; CSR 5,16: PRINT "
13045 IF C=T AND R=VAL(R$) THEN CSR 5,14: PRINT "Correct"
13050 IF C<>T OR R<>VAL(R$) THEN CSR 5,14: PRINT "Wrong": CSR 1,16:
PRINT "It should have been ";T;" remainder ";R
13055 RETURN
14000 CLS : PAPER 11: INK 12
14010 IF S>=7 THEN CSR 3,3: PRINT "Well done, ";N$;"!"
14015 IF S>=5 AND S<7 THEN CSR 3,3: PRINT "You could do better, ";N$;"!"
14020 IF S<5 THEN CSR 3,3: PRINT "Not good enough, ";N$;"!"
14025 CSR 3,5: PRINT "You scored ";S;" out of 10"
14028 PAUSE 5000
14030 CSR 3,10: PRINT "Do you want to try again? (Y/N)"
14040 LET Z%=INKEY$: IF Z%<>" " THEN GOTO 14040
14045 LET Z%=INKEY$: IF Z%="" THEN GOTO 14045
14050 IF Z%="Y" OR Z%="y" THEN GOTO 103
14060 IF Z%="N" OR Z%="n" THEN GOTO 15000
15000 CLS : PAPER 11: INK 6
15002 CSR 5,5: PRINT "O.K. ";N$;"!"
15005 CSR 5,7: PRINT "Thanks for trying my sums!"
15007 PAUSE 4000
15010 CLS : CSR 5,5: PRINT "Does anyone else want a go?(Y/N)"
15020 LET Z%=INKEY$: IF Z%<>" " THEN GOTO 15020
15030 LET Z%=INKEY$: IF Z%="" THEN GOTO 15030
15040 IF Z%="Y" OR Z%="y" THEN RUN
15045 IF Z%="N" OR Z%="n" THEN CSR 5,7: PRINT "GOODBYE!": PAUSE 4000
15050 IF Z%="N" OR Z%="n" THEN NEW

```

PROGRAMMING THE VDP & VRAM

One in every five telephone calls I receive is to ask a question which involves the VDP chip. I shall now attempt to explain the chip's operation, and how it interfaces with the MTX, as thoroughly as possible.

This month I will deal with the overview of the VDP and **Graphic Mode II**. In November's edition I shall take you through all the aspects of the **Sprite Tables**. At this point I would urge the novice, and those members who have never used **assembly language**, to try the various machine code routines, and to **have a go** at setting up the VDP by using the assembler.....**jump in at the deep end.....you can do no harm to the machine.**

OVERVIEW

The VDP fetches data from **Video Ram [VRAM]** and after processing, the data is used to control the beam of a CRTube as it scans the screen. This sequence is repeated over and over again like an endless **FOR/NEXT** loop. The VDP also performs many more functions such as taking time out to check if the CPU requires access to VRAM or any of the **internal registers**.

The VDP has **nine** internal registers: **eight** contain **control** bits which may be programmed by the user. The **ninth** register is the **Status Register** and this may be read to determine various things that are happening within the VDP.

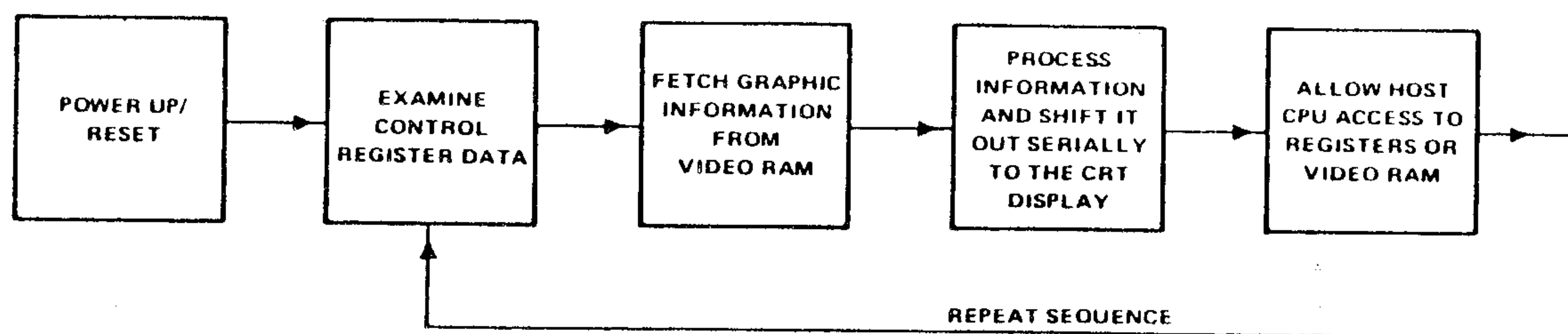


FIGURE 1-1 -- VDP FLOW OF OPERATION

MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

The MTX communicates with the VDP through Ports 1 & 2. The Z80 can be programmed to perform one of the following operations.

WRITE ONE BYTE TO VRAM
 READ ONE BYTE FROM VRAM
 WRITE TO A VDP REGISTER
 SET UP A VRAM ADDRESS
 READ THE STATUS REGISTER

PORT ONE [OUT (01) : IN (01)] IS USED FOR DATA TRANSFERS.
 PORT TWO [OUT (02) : IN (02)] IS USED FOR ADDRESS TRANSFERS

All addressing modes use fourteen bits (2 Bytes with 2 bits unused).

WRITING TO THE VDP REGISTERS

The VDP requires two eight bit data transfers to load an **internal register**..... This is one of the first stumbling blocks and most novices tend to get confused from this point.

To write to VDP Register.

- 1] Send Data1st Byte
- 2] Send VDP Register Number2nd Byte
 (Where to put Data)

To make sure that the VDP recognises that you want to write to a VDP register the MSBit of the **second byte** must be set to 1 and the **next 4 bits** must be zero [0]. The lowest 3 bits then contain the register number [0 - 7].

There is no mystery in this operation, and it is easily performed by using the following formula:-

	Msb							Lsb	
	7	6	5	4	3	2	1	0	
Data	1	0	1	1	1	0	1	1	
Select Register	1	0	0	0	0				Reg No.

The easiest way to make sure that bit seven is always one is to do a logical OR with 80 Hexidecimal [128].

Example: Initialise Register 0 with a value of 10.

1st. Send dataLD A,10
 OUT(02),A

2nd Send Register No ... LD A,0
 OR #80 ;Bit 7 now 1
 OUT(02),A

READING & WRITING TO VRAM

The VDP is coupled to VRAM via an **auto-incrementing address register**. This means that once the address we want to read from, or write to, has been sent to the VDP we can read, or write data using **one byte** transfers, and the address will be automatically incremented. I am sure you can follow the implications of this statement: reading or writing to sequential addresses can be performed very quickly.

Write to Vram Or Read From Vram

	1]	Send	LSByte	of	address
	2]	Send	MSByte	of	address
	3]	Write	data \	read	data
Sequential Read\Write	..4]	"	"	"	"
	5]	"	"	"	"
	.				
	100]	"	"	"	"

And so on

Writing to VRAM

The VDP recognises that you are requesting a write to Vram when **Bit 7** is 0 & **Bit 6** is 1 in the MSbyte of the address. This is again easily accomplished by performing a **Logical OR with 40 Hexidecimal [64]**.

Example:

Send Ascii 'A' to address 3C00 Hex.

SEND ADDRESS LSB	LD A, £00
	OUT(02),A
SEND ADDRESS MSB	LD A, £3C
	OR £40
	OUT(02),A
SEND DATA	LD A, "A"
	OUT (01),A

Reading From VRAM

Once again you have to get the VDP to realise you are requesting a **READ** operation. This simply involves making sure that **Bit 7** is 0 & **bit 6** is 0 in most cases you do not have to do anything as addresses are never more than **14 bits** in length. To make sure, however, I always perform a **Logical AND with 3F Hexidecimal**

MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

Example

Read what is displayed at location £3C00 Hex.

```
SEND ADDRESS LSB ..... LD A,£00
                           OUT(02),A
SEND ADDRESS MSB ..... LD A,£3C
                           AND £3F
                           OUT(02),A
READ FROM THAT ADDRESS ..... IN A,(01)
```

READING THE STATUS REGISTER.

This is a very simple operation. IN A,(02). After this instruction the A register will hold the contents of the status register, and data can be extracted by performing bit tests.

MODES

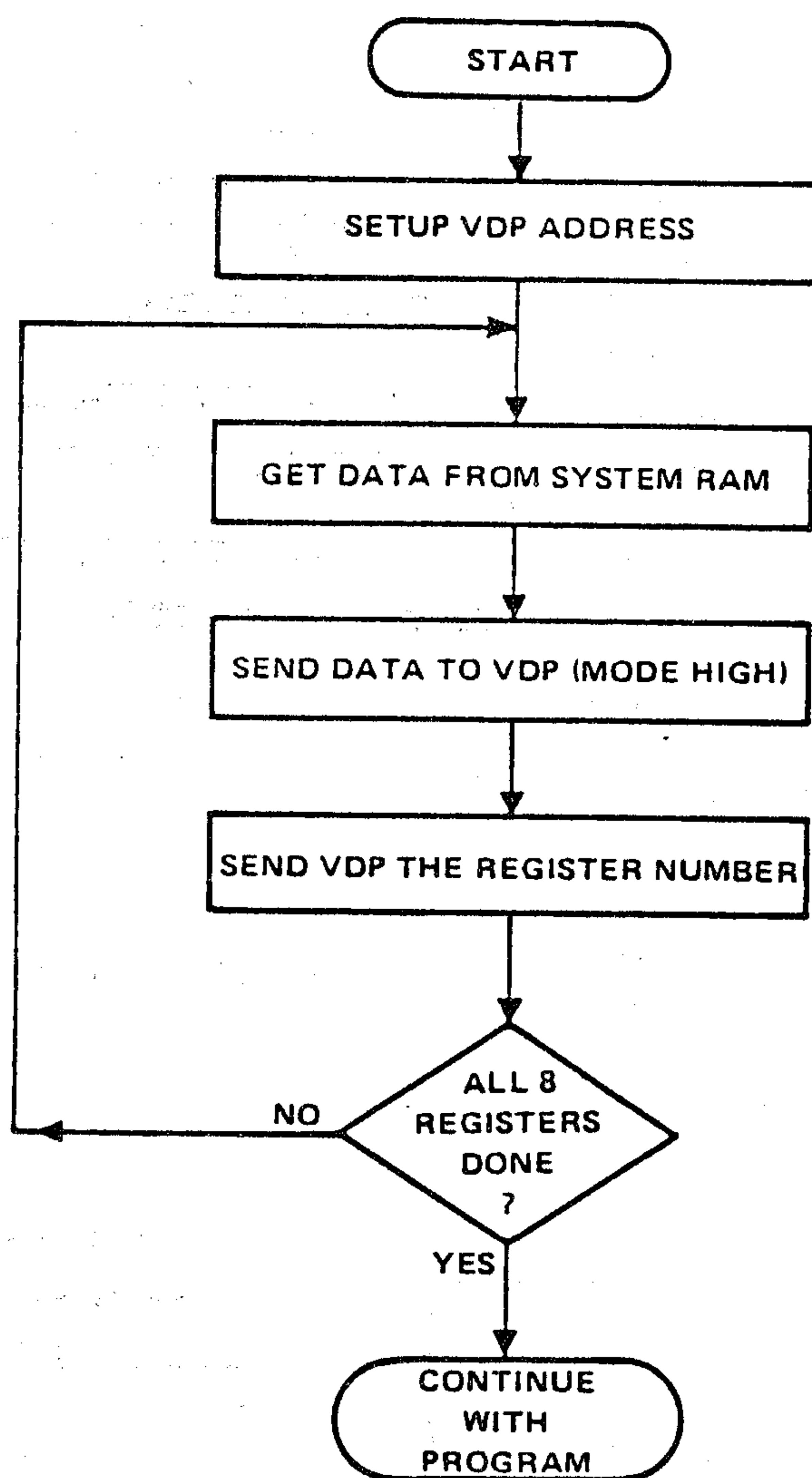
There are four modes available on from the VDP:

TEXT : GRAPHIC I : GRAPHIC II : MULTICOLOUR

MTX Basic only allows the use of Text & Graphic II. However, under assembly conditions all modes are available, although I can see no use for the multicolour mode.

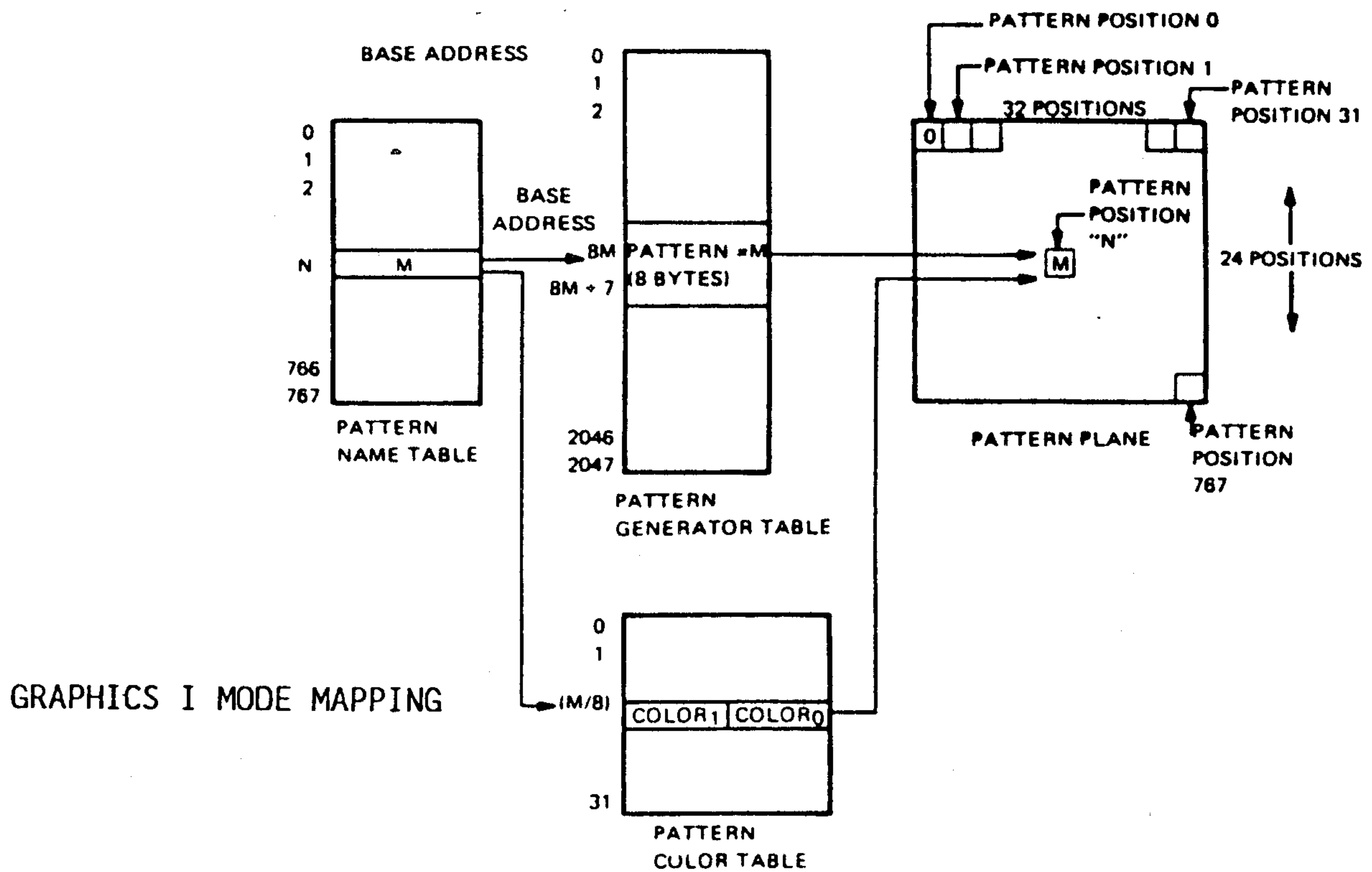
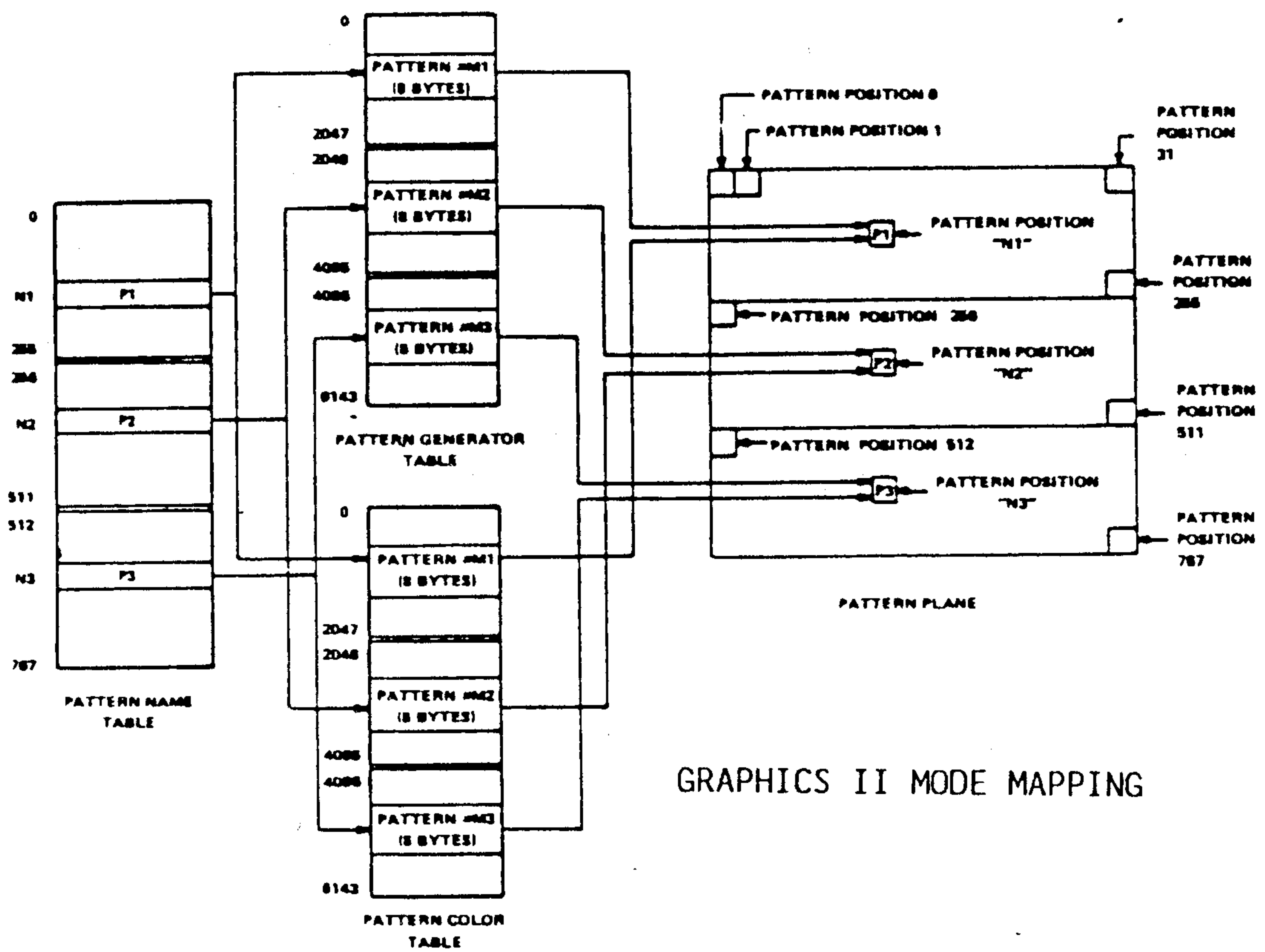
Because Graphic II is the most versatile mode, the rest of this article is devoted to this particular mode, but the information is easily translated to refer to the other modes.

Before you can use mode II the VDP must be initialise to that mode. Take a look at the following Flow Chart that shows you the steps involved when initialising the VDP registers.



REGISTER INITIALIZATION

CONDITION	MODE	VDP DELAY	TIME WAITING FOR AN ACCESS WINDOW	TOTAL TIME
Active Display Area	Text	2 μ s	0 - 1.1 μ s	2 - 3.1 μ s
Active Display Area	Graphics I, II	2 μ s	0 - 5.95 μ s	2 - 8 μ s
4300 μ s after Vertical Interrupt Signal	All	2 μ s	0 μ s	2 μ s
Register I Blank Bit 0	All	2 μ s	0 μ s	2 μ s
Active Display Area	Multicolor	2 μ s	0 - 1.5 μ s	2 - 3.5 μ s



Most users get confused on how to calculate the values for the registers to allow the different tables to be located in VRAM. The following explanations should help to clear the fog.

REGISTER 0

This register contains two VDP control Bits

BIT 1 = Pattern Mode Bit 3 ==> M3

This is one of three bits that when set, determine which display mode the VDP is in. The other two **mode bits** are in **register one**.

D7						D0	
0	0	0	0	0	0	M3	EXT. VID.

The various configurations are:-

M1	M2	M3	
0	0	0	==> Graphic Mode I
0	0	0	==> Graphic Mode II
0	1	0	==> Multicolour Mode
1	0	0	==> Text Mode

The second control bit need not concern us, but it is, in fact, a toggle to enable and disable the **External Video**. It is Bit 7.

REGISTER 1

Register one contains **eight** control bits.

D7						D0	
4/16 K	BLK. SCRN	IE	M1	M2	0	SPR. SIZE	SPR. MAG.

Bit 7 is the 4K\16K toggle and should always be set to one.

Bit 6 = Blank enable\disable.

0 = blanks out the display area & 1 enables the display.

This is useful for blanking out the sprites and pattern plane at a stroke. Using this does not destroy any of the tables in VRAM, when enabled, although the pattern planes are blanked out, the backdrop [border] colour fills the screen.

Bit 5 This is a very important bit. Bit 5 controls the VDP interrupt cycle.

0 ==> disables interrupts & 1 ==> Enables interrupts.
When the VDP interrupt is enabled by this bit, the interrupt will be generated immediately before the vertical retrace time starts. All this means is: if you program the MTX CTC to accept your interrupt, you can get extra smooth graphic movement by writing to the VDP during this interrupt cycle. Don't worry about this we shall be showing how to use the CTC & interrupts in the next edition.

BIT 4 & 3 are the **pattern mode** bits M1 & M2 respectively.

BIT 2 This **must** always be set to 0 as this bit is reserved for future expansion.

BIT 1 = Sprite Size.

0 = Size 0 sprites (8x8) & 1 = Size 1 sprites (16x16)

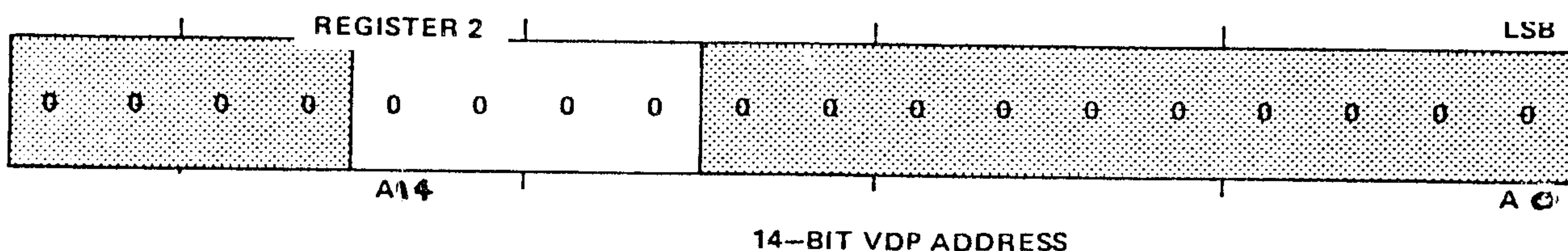
BIT 0 Selects Sprite Magnification

0 = no magnification 1 = magnify by 1

E.g 8x8 becomes 16x16 and 16x16 becomes 32x32

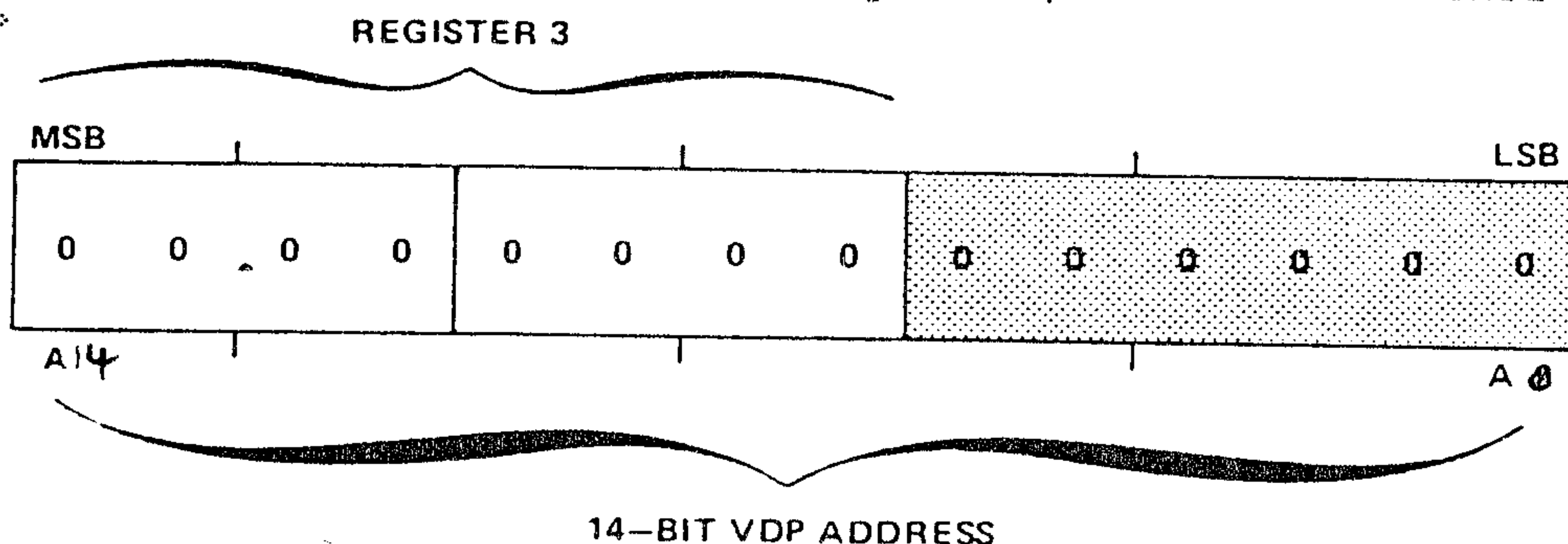
REGISTER 2

This register tells the VDP where the starting address of the **Name Table** is located in VRAM. Only 4 bits are used so the range of this register is from 0 - F. The contents you place into this register form the upper 4 bits of the 14 bit address. The start address = contents of reg 2 * 1024. If you wish to place your table at 6K or 6144 then divide required location by 1024 or 1K.



REGISTER 3

This register tells the VDP where to put the start of the **Colour Table**. Register 3 is an 8 bit register and can hold a number in the range 0 - FF. Again, the register contents form the upper (8) bits of the 14 bit register. Location in VRAM = Contents of reg 3 * 64 or desired location / 64 = contents of register 3.... there is a slight complication with this register ==>

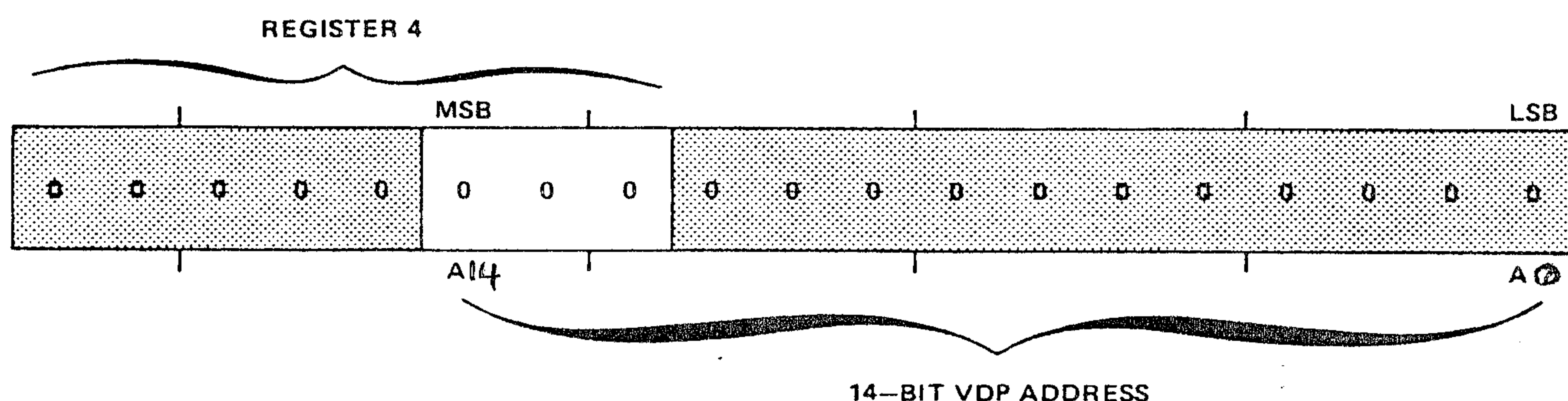


MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

Register 4 & Register 3 function differently when the VDP is in Graphics II Mode. In this mode the Color Table can only be located in one of two places in VRAM, either Hex 0000 or Hex 2000. If Hex 0000 is where you wish the Color Table to be located, then the MSB in Register 3 has to be a 0. If Hex 2000 is the location choice for your Color Table, then the MSB in Register 3 must be a 1. In either case, all the LSBs in Register 3 must be set to 1's. Therefore, in Graphics II Mode the only two values that work correctly in Register 3 are Hex 03 and Hex FF.

REGISTER 4

The VDP depends on this register to check where the Pattern Name Table should start. Only 3 bits of data are allowed in this register, and, again, the data forms the upper 3 bits of the 14 bit register. Location = contents of register 4 * 2k (2048).

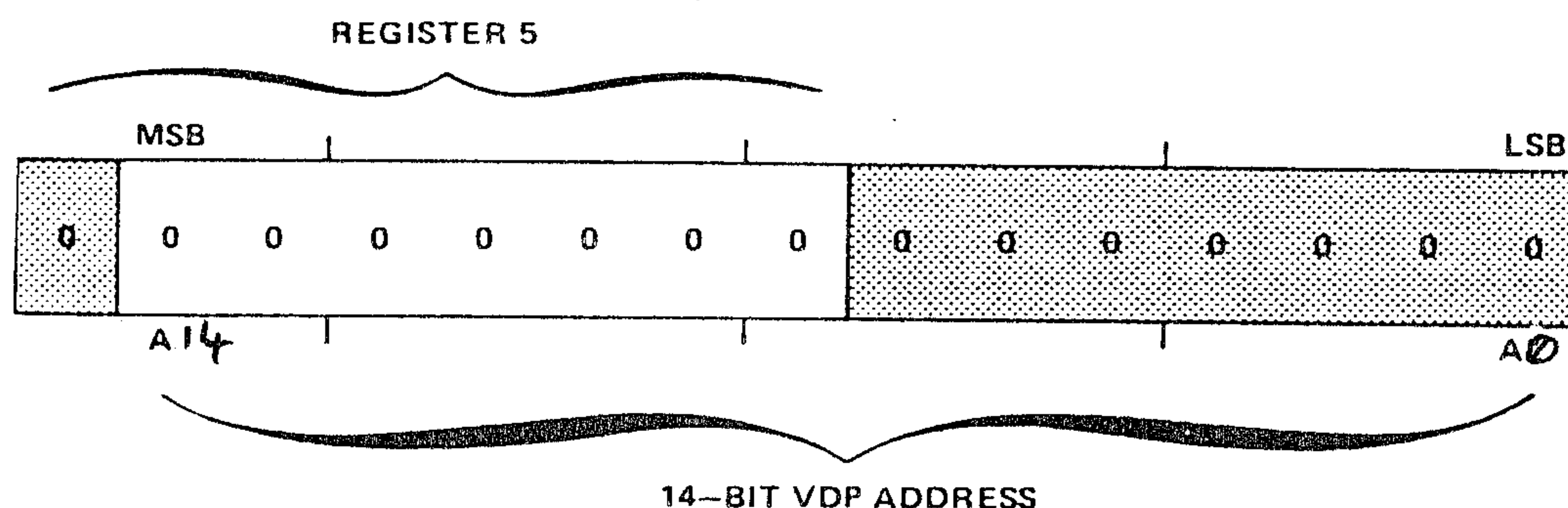


N.B THE SAME ADDRESSES AND NOTES APPLY TO THIS REGISTER. SUBSTITUTE PATTERN NAME TABLE FOR COLOUR TABLE IN THE NOTE AT THE TOP OF THIS PAGE.

WHAT ALL THIS MEANS IS: You have a choice of feeding the registers with either 03 which places the table @ 0000 or FF which places the table @ 2000.

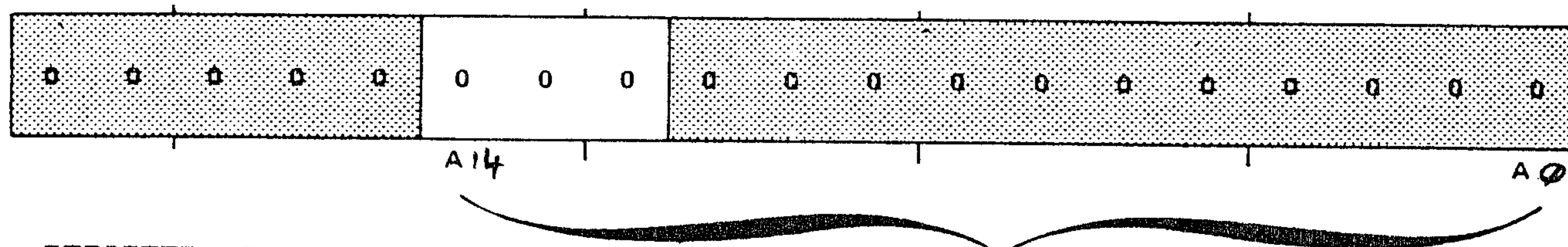
REGISTER 5

Register 5 is a 7 bit register. The contents, which can range from 0 - 7F, form the upper 7 bits of the 14 bit VDP address. This places the Sprite Attribute Table in Vram. Location = Contents Reg 5 * 80 (128). or Desired Location/128 = contents register 5.



REGISTER 6

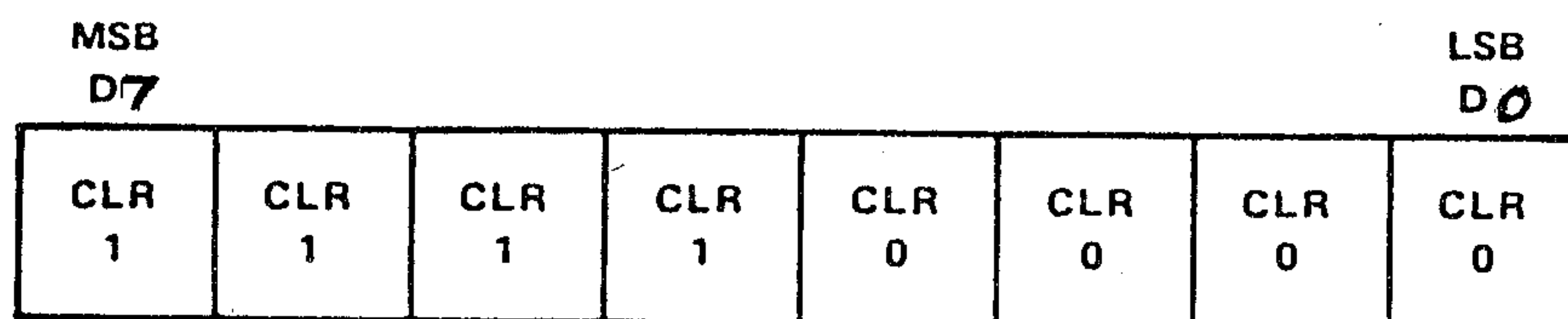
This register tells the VDP where the start address of the **Sprite Pattern Table** is located in Vram. This is a 3 bit register, and the formula is the same as that of the other registers. $\text{Location} = \text{contents of reg 6} * 2048 \text{ or } 2K.$



REGISTER 7

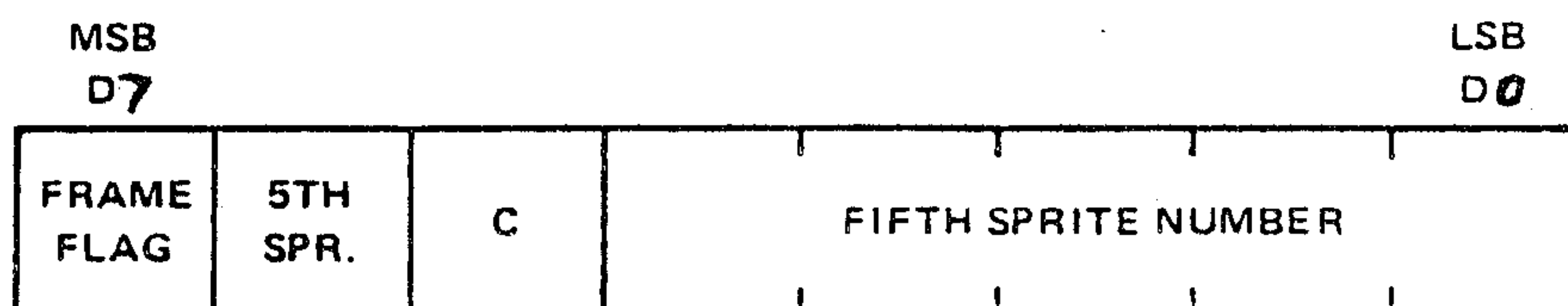
The lower 4 bits of register 7 define the colour of the 0's in the pattern when using the **TEXT mode**, and the upper 4 bits define the colour of the 1's or **ON bits** of the pattern. It should be noted that the upper 4 bits also define the colour of the **BACKDROP** colour in all modes.

REGISTER 7



READ ONLY REGISTER 8

STATUS REGISTER



Before we continue it should be noted that the MTX uses **MODE II** as a **Bit Mapped Display**. This way of configuring VRAM is slightly more complicated, and we shall examine its operation in a future edition.

Another point should also be considered. If VRAM is configured to addresses other than those used by Basic, you will not be able to use the **Front Panel** to debug your programs. MTX Basic hides its Ascii character codes at location 6144 through to 7168.

A way around this problem is to allow Basic to set up the tables and then re-configure the **Generator Table**.

GRAPHIC MODE II

Graphic Mode II has a resolution of 256 x 192 pixels. Three tables are required to generate a display: **Name Table {Screen}**, **Colour Table**, and **Pattern Generator Table**.

The **Name Table**, or screen is 768 bytes long, but the **Colour & Generator tables** are 6144 bytes in length. The reason for this is not obvious. In the Text Mode, Graphic Mode I, and on most other computers, only 256 Ascii characters are available. The MTX, in **mode2**, allows you to treble this figure ! In fact, you can create 768 unique patterns, one for each screen location. This mode also allows the patterns to be further enhanced by permitting 8 bytes of colour information to be used within each character - all 16 colours can be defined within one pattern.

COLOUR TABLE

The **Colour Table** is the same length as the Pattern Table (6144 bytes), and is also segmented into **three 2048 byte** blocks. Each block is further divided into 256 colour definitions which are 8 bytes long. Each section of the Colour Table maps onto the equivalent section of the Pattern Table.

Each byte within the Colour Table defines the colour of the bits that are on, or the bits that are off. This colour can be the same as the background, transparent, or another unique colour.

ROW 0	0	1	0	0	0	0	0	1	B	1	B	B	B	B	B	1	0	3	4	7	0 ROW
1	0	0	1	0	0	0	1	0	B	B	7	B	B	B	7	B	7 (CYAN)	B (LT. YELLOW)		1	
2	0	0	0	1	0	1	0	0	B	B	B	C	B	C	B	B	C (GREEN)	B (LT. YELLOW)		2	
3	0	0	0	0	1	0	0	0	B	B	B	B	E	B	B	B	E (GRAY)	B (LT. YELLOW)		3	
4	0	0	0	0	1	0	0	0	B	B	B	B	8	B	B	B	8 (MED. RED)	B (LT. YELLOW)		4	
5	0	0	0	0	1	0	0	0	B	B	B	B	5	B	B	B	5 (LT. BLUE)	B (LT. YELLOW)		5	
6	0	0	0	0	1	0	0	0	B	B	B	B	6	B	B	B	6 (DK. RED)	B (LT. YELLOW)		6	
7	0	0	0	0	1	0	0	0	B	B	B	B	D	B	B	B	D (MAGENTA)	B (LT. YELLOW)		7	

PATTERN GENERATOR
TABLE ENTRY

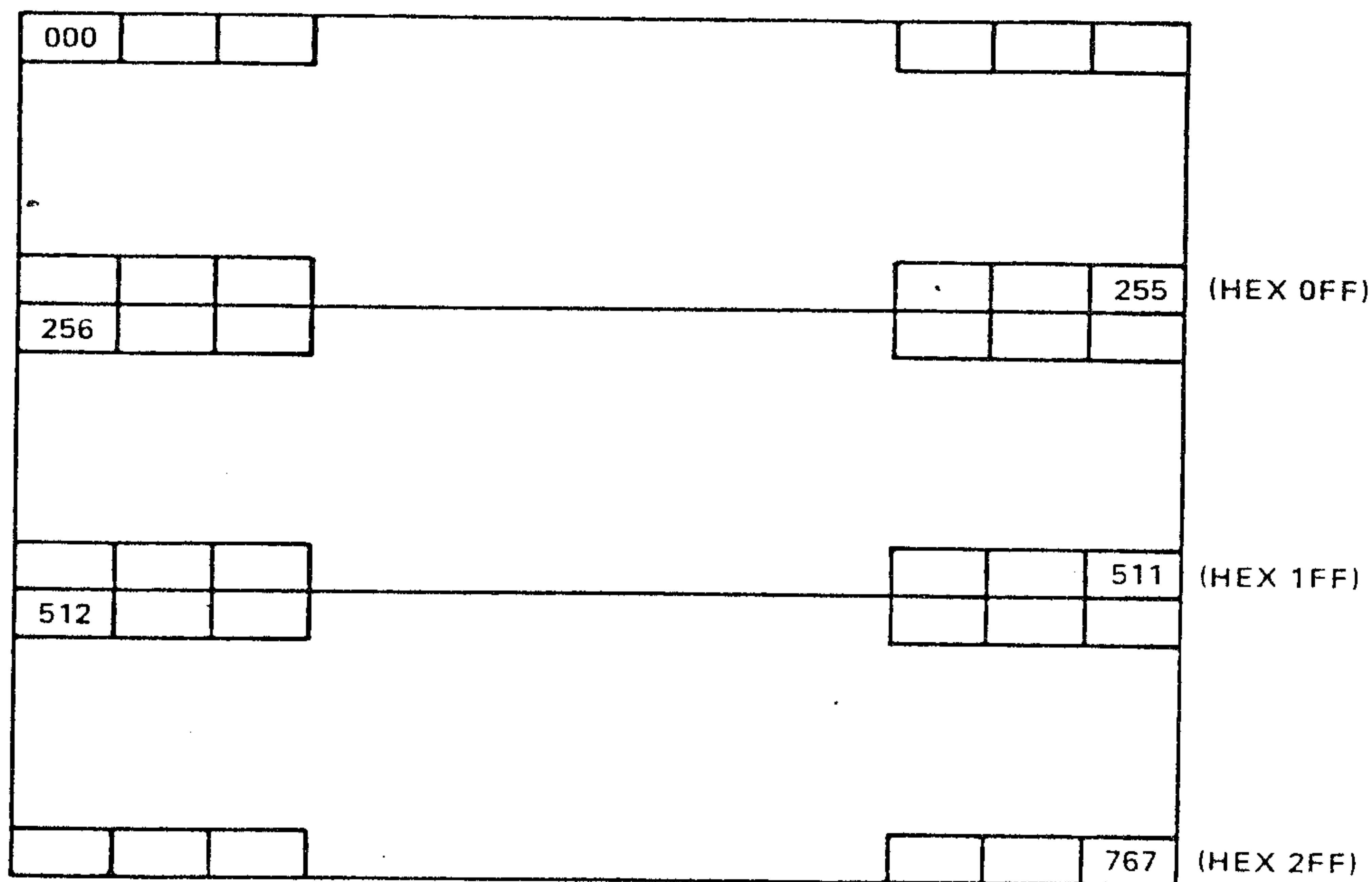
PATTERN

PATTERN COLOR
TABLE ENTRY

As already stated, the **Name Table** contains 768 entries which correspond to each of the 768 display positions on the VDU. Because there are only 256 Ascii codes (Pattern Names), to allow us to display 768 **different patterns**, the screen is split into **three sections**, and each section is 256 bytes in length.

Top 3rd	000 - 255
Middle 3rd	256 - 511
Bottom 3rd	512 - 767

This table is 6144 bytes long, and is also split into three equal blocks of 2048 bytes. Each block is further divided into 256 8x8 pixel graphic characters. The first 256 patterns can only be displayed on the upper third of the screen, the second block can only be displayed on the middle third of the screen, and the last 256 patterns can only be displayed on the lower third of the screen. Care must be taken if you are moving a character around the screen. If ascii 128 defines a space-ship in the first third of memory to move the ship into the second or bottom third of the screen, it must be defined as CHR\$(128) in each of these sections.



GRAPHICS II MODE NAME TABLE SEGMENTED INTO THREE EQUAL BLOCKS

SUBROUTINES

Set Up VDP registers and initialise a G II Screen

Values for registers are assumed to be held in memory location REG

```

START: LD HL,REG      ; Point HL at data.
        LD BC,£0880    ; B = no of registers : £80 sets bit 7
INIT:   LD A,(HL)      ; Get Value.
        OUT(02),A      ; Send Data.
        LD A,C         ; Get register number 1st time = 0 + £80
        OUT(02),A
        INC C
        INC HL
        DJNZ INIT
REG:    DB 2,194,15,255,3,126,07,£5C ; DATA FOR MTX BASIC SET UP.

```

Send to VRAM

Assume that HL contains address to recieve data....A holds data byte to send..

```

LRAM:   PUSH AF      ;SAVE AF
        LD A,L
        OUT(02),A
        LD A,H
        OR £40
        OUT(02),A
        POP AF
        OUT(01),A
        RET

```

Read from Vram

Assume HL holds address to read from and A will return with data.

```

LREAD:  LD A,L
        OUT (02),A
        LD A,H
        AND £3F
        OUT(02),A
        IN A,(01)
        RET

```

Reverse colour:

Assume that on entry to this routine A holds colour byte to be reversed.
On Exit A will contain reversed colour.

```

REV:    RLCA
        RLCA
        RLCA
        RLCA
        RET

```

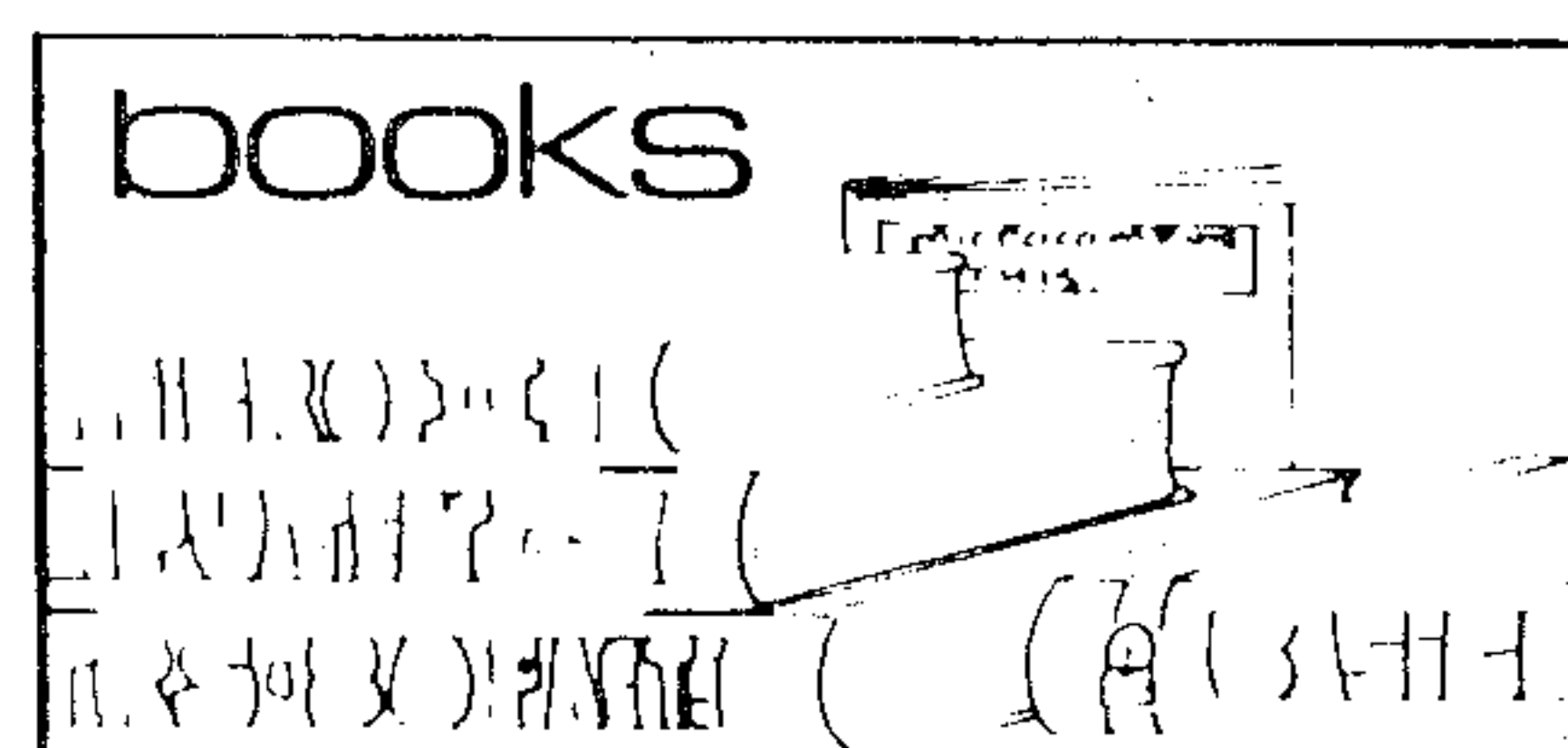

MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

HIGH SCORES: HIGH SCORES.....Can you do better ??

MAXIMA	42,500
SNAPPO	74,320
STAR COMMAND	52,250
PHAID	23,470

STEPHEN WHITE.
STEPHEN WHITE.
DEREK WHITE.....Mission Completed.
ERIC PETRES.

BOOKS AND MORE BOOKS



Real Time Programming By Caxton Foster.

This book deals with some of the problems you will come up against in connecting a computer to real-world devices. It is not intended as a collection of cookbook recipes for solving actual problems, but as a survey of problems to help you find your own solutions.

Subjects: Basic interrupts, Ports, Multiple interrupts, closing the loop, and many more including communication over restricted pathways.... a very interesting book for the more experienced programmer. Index and appendix are fully supported

Published by Addison-Wesleyaround £8.95

Pascal for Basic Programmers By Charles Seiter & Robert Weiss.

Most programmers learn Basic as their first language. However, with the help of this book, there is no reason why the novice should not try a completely new language. There is no doubt: Pascal offers increased sophistication and power for everyday programming.

This book represents a mini-course in Pascal. Rather than just translating Basic statements into Pascal, the book emphasises the process of programming in Pascal, showing how to take full advantage of its power.

Highly recommended.....£9.95

Mastering CP/M By Alan R. Miller.

This is an excellent book for the programmer who is just about to enter the world of Disc Systems. As most of you know, the FDX is a CP/m based system.....this will be a completely foreign field to the first time purchaser. Alan Miller takes the reader from the basic concepts of CP/m to the more advanced methods of using BDOS & BIOS calls to gain optimum efficiency from the system.

An excellent book and far superior to Rodney Zak's CP/m Handbook.
SYBEX priced around£11.60

GOGO ANOTHER POINT OF VIEW by ROBERT PEET



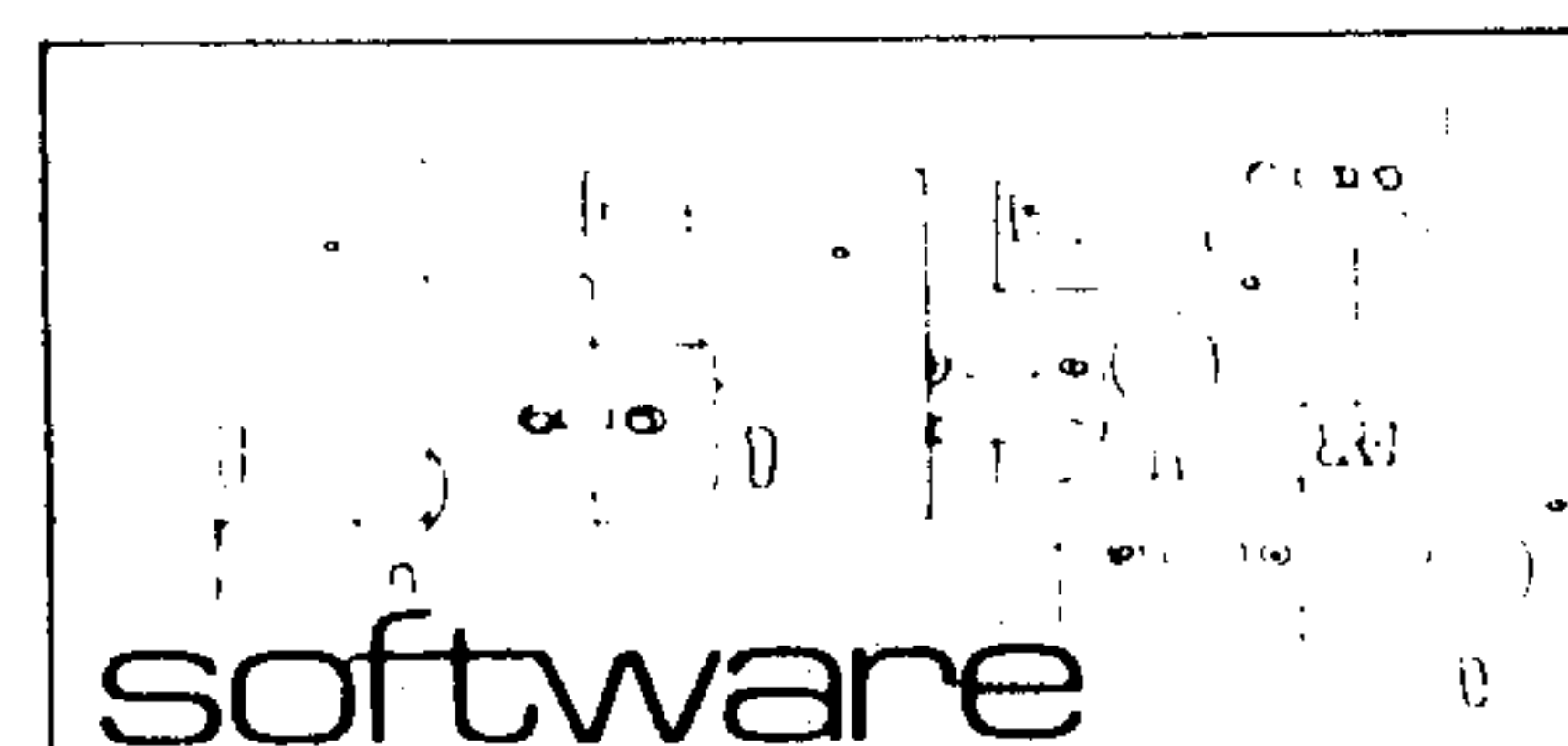
This game involves getting a man to fill in the squares on a 3D pyramid constructed in different colours. The only catch is that boulders roll down the pyramid and splat you if you collide with them! Also, spinning shapes descend on you at varying intervals, and bumping into these gives you a bonus of 300 points.

I managed to get to the fifth level here the going gets really tough with two boulders at once trying their hardest to take a life from you.

The sound is neat and the graphics superb. This is the sort of game you can never put down. I only found one minor criticism: it is far easier to play using the keypad because of the way the author has configured the movement of the joystick..... so if you don't own a quickshot, don't worry.

Altogether, a very nice game.

MUSIC PAD by ROBERT PEET



This utility is for the production and editing of music. Although, I must say, I found producing a single note a bit complex. However, selecting keyboard pitch, note, length, etc., is all part of music, and you don't have to be a budding Handel to produce reasonable music.

Up to 180 notes can be programmed at one time. Nice features are included like save a tune which can then be interfaced with your own program, and a hard copy can also be obtained from your DMX80.

This program was a joy to use..... so if you want to be a composer, or a [bus] conductor, this piece of software is a must.

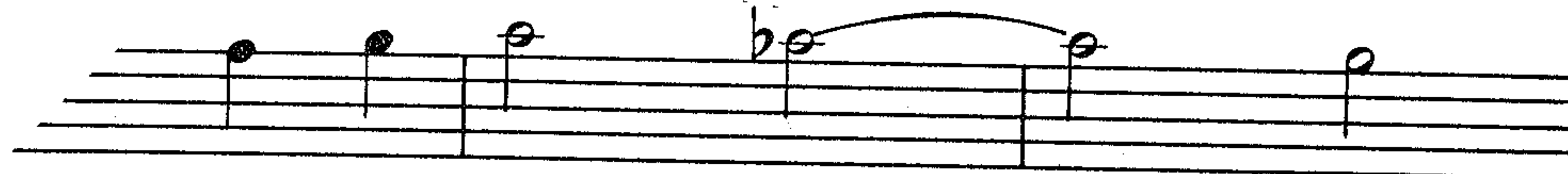
Continental Software. Available from Genpat £6.02p inc P.P.

Robert is one of our younger members and was responsible for the evaluation of the above software. Robert will be back next month with some more reviews. If any more of our younger members would like to participate in reviewing software, drop me a line..... you even get to keep the software !!!!!

MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

;LISTING FIVE
;Creative Computing
;.z80
;

*** SOUND ***



CHAN: EQU \$FE14
FREQ: EQU \$FE16
VOL: EQU \$FE18

;LOAD CHANNEL FROM CHAN AND CONVERT TO SOUND CHIP VALUE.

LOAD:

LD A,(CHAN)
AND 3 ;MAKE SURE =< 3
LD B,A ;PUT IN B FOR ALIGNING
LD HL,BACK

GETVAL:

INC HL ;Align to DB bytes
DJNZ GETVAL ;loop until aligned
LD A,(HL) ;Put value in A

BACK: ;and return

RET

DB 128,160,128,192,224

;OUTPUT FREQUENCY TO SOUND CHIP WITH CORRECT CHANNEL.

OUTFREQ:

CALL LOAD ;Get channel number
LD B,A ;Correct value now in B

LD A,(FREQ)
AND \$0F

OR B

OUT (06),A

IN A,(03)

LD A,(CHAN)

AND 3

CF 3

RET Z

LD HL,(FREQ)

ADD HL,HL

ADD HL,HL

; TIMES BY 8

ADD HL,HL

ADD HL,HL

LD A,H

AND \$7F

STROBE:

OUT (06),A

IN A,(03) ;SOUND OUT

RET

;OUTPUT VOLUME TO CHIP WITH CORRECT CHANNEL NUMBER

VOLX:

CALL LOAD

OR 16

LD B,A

AND \$0F

OR B

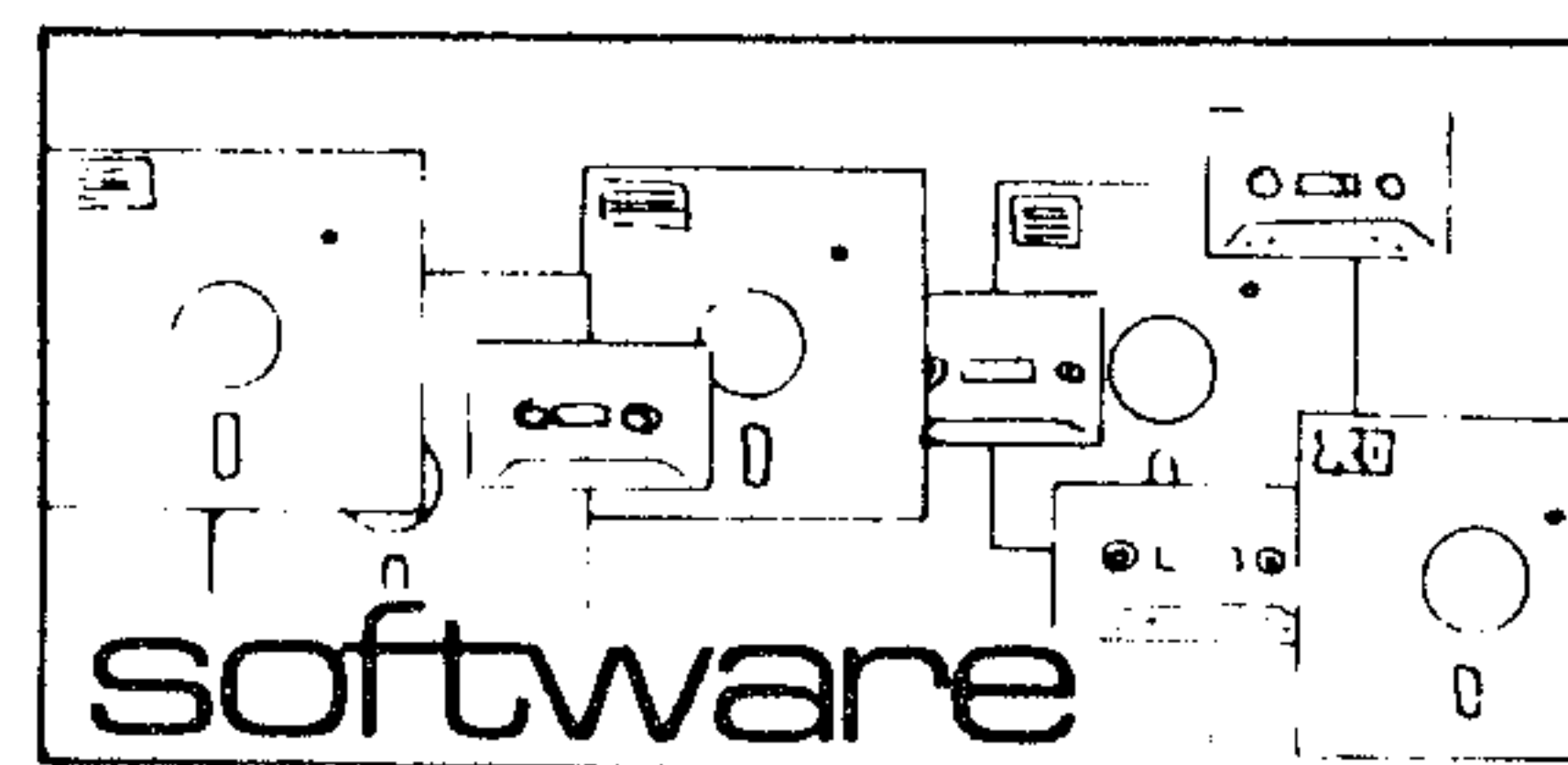
JR STROBE

; LOAD CHANNEL,FREQUENCY,VOLUME WITH CORRECT PARAMETERS THEN CALL
OUTFREQ FOLLOWED BY CALL VOLX TO OUTPUT SOUND

**** ANOTHER WAY****

; LD FREQ,CHAN,VOL WITH PAREMETERS THEN CALL \$0F86 TO OUTPUT A SOUND

;<C> KH 83.



HIGH RESOLUTION & TEXT SCREEN DUMP

MEMBRAIN SOFTWARE

If you wonder how we use actual screen dumps to illustrate the programs in this edition.... the answer lies in this program. It's not often that my life is made easier, but Stephen Varley, who runs Membrain, has done just that.

This is a machine code utility that is loaded into memory, and once it has re-located itself, it lies dormant and transparent to the end user until required.

Pressing F1 dumps the high resolution screens [VS 4] to any Epson type printers, including the DMX80. Dumping the text screen is simply a matter of pressing F2 - this is one way of solving printing problems with the Noddy screens.

A nice utility. Available from Membrain Software, 25, High Road, Redworth, Newton Aycliffe, Co. Durham. DL5 6NU. Price to Genpat members £4.95.



THE MTX KEYBOARD

7 PAGE	8 EOL	9 BRK	F1	F5
4 TAB	5 ↑	6 DEL	F2	F6
1 ←	2 HOME	3 →	F3	F7
0 INS	↓	ENT CLS	F4	F8

The easiest way of performing a keyboard scan is to use the MTX ROM call £0079. Using this routine will cause you no headaches - it doesn't affect any registers and the only flag affected is the **Zero Bit**.

A call to £0079 will return with the result in the A register. I.e. if a key has been pressed A will contain a normal Ascii character, or a keyboard value.

Testing the **Zero Flag** will verify the following conditions:-

- The Zero flag will be **set** if no key has been pressed.
- The Zero flag will be **reset** if a key has been pressed.

To disable the break key **reset** bit 1 at location £FD5E which is system variable INTFFF. To enable the break key **set** bit 1 at the same location.

Keyboard debounce is serviced by **LASTKEY** but a more drastic cure can be accomplished by disabling the auto repeat bit at £FD5E. This is done by resetting it to zero.

MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

If you have no wish to get your 'hands dirty' by using machine code, it is quite a simple matter to perform a keyboard scan from basic.

The read lines on the MTX are directed to PORT 5. The sense lines are also tied to this port.

Explanation

An input from port 5 is interpreted by the MTX as a read byte, and an output is taken as a **sense byte**. The table listed below gives the values to output on port 5 to test if a certain key has been pressed. However, always remember that the MTX uses the opposite method to other computers you may have used; that is: a zero [0] denotes a key press, not a one.

KEY PRESS	OUT (5) Sense byte	INP (5) Read byte
Home Key	£DF	£7F
Cursor Up	£FB	£7F
Cursor Dwn	£BF	£7F
Cursor Rght	£EF	£7F
Cusror Left	£F7	£7F
0	£FD	£DF
1	£FE	£FE
2	£FD	£FD
3	£FE	£FD
4	£FD	£FB
5	£FE	£FB
6	£FD	£F7
7	£FE	£F7
8	£FD	£EF
9	£FE	£EF
A	£DF	£FE
B	£7F	£FB
C	£7F	£FD
D	£DF	£FD
E	£F7	£FD
F	£EF	£FB
G	£DF	£FB
H	£EF	£F7
I	£FB	£EF
J	£DF	£F7
K	£EF	£EF
L	£DF	£EF
M	£7F	£F7
N	£BF	£F7
O	£F7	£EF
P	£FB	£DF

MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

Q	£F7	£FE
R	£FB	£FB
S	£EF	£FD
T	£F7	£FB
U	£F7	£F7
V	£BF	£FB
W	£FB	£FD
X	£BF	£FD
Y	£FB	£F7
Z	£7F	£FE

 TAKE NOTE ***** NEW ***** TAKE NOTE ***** NEW

GENPAT HAS NOW STOCKS OF THE NEW USER MANUAL. THIS IS A FULLY UP-DATED MANUAL THAT ALSO INCORPORATES THE BEST PARTS OUT OF THE DELETED SPENCER BATESON'S BOOK. ALL THE UP TO DATE SYSTEM VARIABLES ARE THERE AND THE BOOK IS FULL OF INFORMATION. SEND YOUR CHEQUES TO THIS ADDRESS AND ALLOW 3 DAYS FOR SHIPMENT.

THE NEW MTX USER MANUAL £5.50 + 75p Postage.

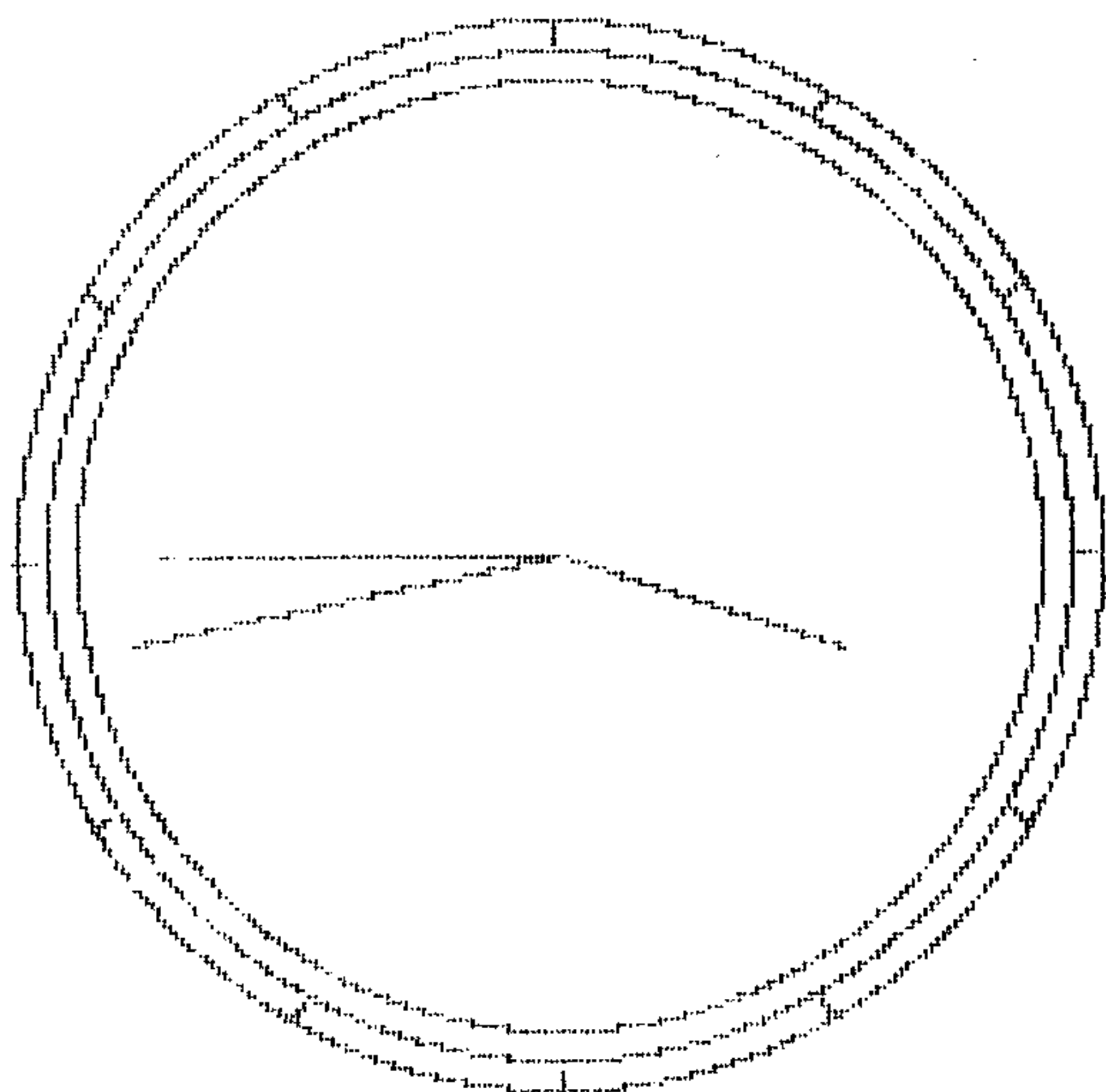
 THE GENPAT SPEECH SYNTHESISER £29.95 + 50p P.P

This speech synthesiser has been exclusively made for the members of GENPAT. It plugs into your centronics port and is fully programmable. It represents excellent value for money - take a look at the BBC- it uses the same chip. A full review of it will be published in the next edition.

 ADVENTURE HELP LINE.....

Stuck in the forest ? Has Snowball given you nightmares ? Send your problem into Memopad. Our experts will attempt to solve your predicament But no cheating.... they will be able to tell if you have really tried !

MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.



When you have nothing to put on the computer.....type this in. It's deadly accurate... you'll be amazed. Single numbers need a leading zero when following the prompts.

PROGRAM

```

1 REM*****
2 REM  MEMOPAD 12 HOUR CLOCK  K.HOOK
3 REM*****
10 VS 4: CLS
20 DIM HR$(2),SEC$(2),MIN$(2): LET E=128: LET F=98: LET R=PI/30: LET COR=-15
30 LET TSX=0: LET TSY=0: LET LSX=0: LET LSY=0: LET MZ=0: LET Z=0
35 COLOUR 4,10: PRINT CHR$(4);CHR$(10): INK 1
60 CSR 2,22: INPUT "HOURS ";HR$: LET HR=VAL(HR$)
65 IF HR>12 THEN GOTO 60
70 CSR 2,22: PRINT CHR$(5);: INPUT "MINUTES ";MIN$: LET MIN=VAL(MIN$)
75 IF MIN>60 THEN GOTO 70
80 CSR 2,22: PRINT CHR$(5);: INPUT "SECONDS ";SEC$: LET SEC=VAL(SEC$)
90 IF SEC>60 THEN GOTO 80
100 LET T$=LEFT$(HR$,2)+LEFT$(MIN$,2)+LEFT$(SEC$,2)
110 LET HR=HR*5+INT(MIN/12): REM CORRECT HOURS TO SIN & COS
120 CLOCK "000000": CLOCK T$
130 ATTR 2,0
140 CLS : INK 15: CIRCLE E,F,90: CIRCLE E,F,85: CIRCLE E,F,80
150 COLOUR 1,8: FOR I=1 TO 12
160 LET NM=I/6*PI
170 LINE E+86*SIN(NM),F-86*COS(NM),E+90*SIN(NM),F-90*COS(NM)
180 NEXT
190 LET Z=(HR+COR)*R
200 LET LHX=E+50*COS(Z)
210 LET LHY=F-50*SIN(Z)
220 LET MZ=(MIN+COR)*R
230 LET LMX=E+65*COS(MZ)
240 LET LMY=F-65*SIN(MZ)
280 LET SEC1=VAL(RIGHT$(TIME$,2)): IF SEC=SEC1 THEN GOTO 280
290 LET SEC=SEC1: LET SAN=(SEC+COR)*R
300 LET LSX=E+72*COS(SAN)
310 LET LSY=F-72*SIN(SAN)
320 ATTR 2,1: LINE E,F,TSX,TSY: ATTR 2,0
325 INK 5
330 LINE E,F,LSX,LSY: LET TSX=LSX: LET TSY=LSY
340 LINE E,F,LMX,LMY
350 LINE E,F,LHX,LHY
360 IF SEC<>00 THEN GOTO 280
370 ATTR 2,1: LET MIN=MIN+1: LINE E,F,LMX,LMY
380 IF (MIN-INT(MIN/12)*12)<>0 THEN ATTR 2,0: GOTO 220
390 LET HR=HR+1: LINE E,F,LHX,LHY: ATTR 2,0: GOTO 190

```


ASSEMBLY LANGUAGE**PART 2**

In the old days, the only way to write **assembly language** programs was to **hand assemble** each instruction. As you can imagine, this was a laborious and time consuming task. Today, **Assemblers** have made the task far easier.

The **assembler** translates your code, written with mnemonic instructions, and called the **source program**, into the **object program**, a machine language program which the MTX executes when loaded into the computer.

INPUT TO ASSEMBLER ==> SOURCE CODE

OUTPUT FROM ASSEMBLER ==> OBJECT CODE

With the MTX assembler, there is no need to save separate files of **source code** and **object code** because the Memotech automatically displays the source code when the List command is given, and each line is assembled internally as the mnemonics are entered from the keyboard.

Before you can attempt to use assembly language, you must learn the instructions - as you did with Basic - and how to use them. With Basic you stored your values in **variables** e.g. X, NUM, X1, Y\$, etc. In assembly language programming values are stored in memory locations, or in **registers**.

The Z80 processor contains two sets of internal, general registers, and six special purpose registers. Take a look at the following:-

GENERAL REGISTERS	A	F	A'	F'	ALTERNATE GENERAL REGISTERS
	B	C	B'	C'	
	D	E	D'	E'	
	H	L	H'	L'	

Z80 REGISTERS

IX	IY	SP	PC	I	R
----	----	----	----	---	---

SPECIAL PURPOSE REGISTERS

A, B, C, D, E, F, H, and L are the normal general registers and the registers designated ' are the **alternate** register set, which can only be accessed by the two instructions **EX AF, AF'** AND **EXX** - these two instructions exchange the contents of the main set with that of the alternate set. Only one set of registers can be used at one time. Following the two sets of 8 Bit registers are four 16 Bit registers : IX, IY, SP, PC.

MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

Registers I & R are very seldom used in most normal programming applications.

The A register is also referred to as the **accumulator** because all of the arithmetic instructions, and most of the other instructions use the contents of the A register as an operand. In fact, this is where most of the transfers take place.

The F register is also called the **Flag** register. The F register sets or re-sets bits internally to indicate a **true** or **false** type of condition, and is **never** used for computations.

The **Program Counter** or **PC** is a 16 bit register that points to the current memory location which hold the instruction to be executed.

Another 16 bit register is the **SP** or **Stack Pointer**. This register keeps a check on the position of the **STACK** in RAM. Two 16 bit registers with very powerful programming possibilities are the **index registers**; **IX & IY**.

Each of the 8 Bit registers can be used separately or in set pairs [BC,DE,HL] and treated as 16 Bit registers.

Assemblers have their own set of rules, but they aren't difficult to learn:-

DB or DEFB	====>	Define Byte.....DB \$FA,'T','ONE'
DW or DEFW	====>	Define Word.....DW \$4007 or Label
DS or DEFS	====>	Define Space.....DS 245 reserve 245 bytes
DM or DEFM	====>	Define Message....DM 'ANOTHER GAME ?'

All the above are known as **Pseudo operations**, and are used by the assembler, **not the CPU**, to carry out predefined functions.

LABELS are used to reference one instruction to another. For example: **JP Z,AGAIN**. A label can be compared with a line number in Basic; e.g **IF A = 0 THEN GOTO 100**.

The semi-colon ; is used in the same way as the **REM** statement in Basic, and the assembler ignores all that follows. It is good programming practice to get into the habit of documenting your program. Believe me, when you look at your code after a few months, you will find it hard to understand what you had in mind at the time you wrote the program.

The convention used by most assemblers is as follows:-

Label	Op Code	Operand	Remarks
START:	LD A,	(DE)	;Put score in A reg.



*** SCREEN SCROLL *** SCREEN SCROLL *** SCREEN SCROLL ***

10 6010 200
100 CODE

```

8010 START: DI ;this is the START or
8011 CALL ROTATE ;entry
8014 CALL USD ;point for a
8017 CP £53 ;user call
8019 EI
801A RET Z
801B JP START
801E VSCALL: NOP ;set up VS as
801F LD DE,15 ;given in VSNUM
8022 LD HL,£FF5D ;bas ad
8025 LD A,(VSNUM) ;VS No.
8028 CP £0
802A JP Z,SCRENO ;VS=0
802D LD B,A ;find base addr
802E LOOPA: ADD HL,DE ;increment base
802F DJNZ LOOPA
8031 SCRENO: LD DE,3
8034 ADD HL,DE ;=Top X
8035 LD A,(HL) ;X co-ord
8036 LD (XST),A ;save it
8039 INC HL
803A LD A,(HL) ;Y co-ord
803B LD (YST),A
803E INC HL
803F LD A,(HL) ;X count
8040 LD (XCT),A
8043 INC HL
8044 LD A,(HL) ;Y depth
8045 LD (YCT),A
8048 RET
8049 VSNUM: DS 1 ;poke this addr
804A NOP ;with VS number
804B SCROL: LD A,(TIMES) ;pixs to move
804E CP 0
8050 RET Z
8051 LD (COUNT),A ;temp store
8054 CALL JOYST ;find scrol dir
8057 LD A,D ;ans in D 0-7
8058 LD (DIRECT),A
805B JP DO
805E REPEAT: LD A,(COUNT)
8061 DEC A
8062 RET Z
8063 LD (COUNT),A
8066 DO: LD A,(DIRECT)
8069 CP £6 ;test ^
806B JP NZ,JP1
806E CALL USCR ;upscrol
8071 JP REPEAT
8074 JP1: CP £7 ;test right up
8076 JP NZ,JP2
8079 CALL USCR ;do up first
807C CALL RSCR ;then right
807F JP REPEAT
8082 JP2: CP £0
8084 JP NZ,JP3
8087 CALL RSCR ;right scroll
808A JP REPEAT
808D JP3: CP £1
808F JP NZ,JP4
8092 CALL RSCR ;do right scrol
8095 CALL DSCR ;then down scr
8098 JP REPEAT
809B JP4: CP £2
809D JP NZ,JP5
80A0 CALL DSCR ;down scrol
80A3 JP REPEAT

```



```

80A6 JP5: CP £3
80A8 JP NZ,JP6
80AB CALL LSCR ;do left scrol
80AE CALL DSCR ;then down scr
80B1 JP REPEAT
80B4 JP6: CP £4
80B6 JP NZ,JP7
80B9 CALL LSCR ;left scrol
80BC JP REPEAT
80BF JP7: CP £5
80C1 JP NZ,JP8
80C4 CALL LSCR ;do left scrol
80C7 CALL USCR ;then up scrol
80CA JP REPEAT
80CD JP8: RET
80CE TIMES: DS 1 ;POKE WITH NO TO MOVE
80CF COUNT: DS 1 ;temp store
80D0 DIRECT: DS 1 ;temp store
80D1 NOP
80D2 NOP
80D3 NOP
80D4 ROTATE: LD A,(YST) ;save orig vals
80D7 LD (STY),A ;into working
80DA LD A,(YCT) ;registers
80DD LD (CTY),A
80E0 LD HL,DATA1 ;tbl st
80E3 LD A,(XCT) ;calculate
80E6 SLA A ;number of byte
80E8 SLA A ;required per
80EA SLA A ;block or line
80EC LD (BUF),A ;save it
80EF LD A,(XST) ;calculate st.
80F2 SLA A ;address for
80F4 SLA A ;address low
80F6 SLA A
80F8 LD E,A
80F9 LD A,(STY)
80FC LD D,A
80FD LOOP6: LD (HL),E ;address low
80FE INC HL
80FF LD (HL),D ;address high
8100 INC HL
8101 CALL VRAMA ;send add
8104 LD A,(BUF) ;count of bytes
8107 LD B,A ;to load in
8108 LOOP9: CALL DATAA ;read
810B LD (HL),C
810C INC HL
810D DJNZ LOOP9
810F INC D
8110 LD A,(CTY)
8113 DEC A
8114 LD (CTY),A
8117 JP NZ,LOOP6
811A LD (HL),£FF
811C LD (DATAE),HL ;store
811F CALL SCROL ;CALL SCROL
8122 LD HL,DATA1
8125 LOOP8: LD A,(HL)
8126 CP £FF
8128 RET Z ;exit if finish
8129 LD E,A
812A INC HL
812B LD D,(HL)
812C CALL VRAM
812F INC HL
8130 LD A,(BUF)
8133 LD B,A
8134 LOOP7: LD C,(HL)
8135 INC HL

```



```

8136 CALL DATA
8139 DJNZ LOOP7
813B JP LOOP8
813E NOP
813F BUF: DS 2
8141 XST: DS 1
8142 YST: DS 1
8143 XCT: DS 1
8144 YCT: DS 1
8145 STY: DS 1
8146 CTY: DS 1
8147 NOP
8148 NOP
8149 RSCR: LD HL,DATA1 ;addres
814C LOOP10: LD A,(HL) ;read add low
814D CP £FF ;check end data
814F RET Z ;exit if so
8150 LD A,(XCT) ;num of char to
8153 LD D,A ;scroll
8154 INC HL ;pt.add high
8155 INC HL ;pt. 1st data
8156 AND A ;clear carry
8157 LD C,0 ;clear reg
8159 LOOP11: LD B,8 ;set byte count
815B RR C ;rotate thr car
815D LOOP12: RR (HL) ;rot data
815F RR C ;save carry bit
8161 INC HL ;pt next add
8162 DJNZ LOOP12 ;cont
8164 DEC D ;num char left
8165 JP NZ,LOOP11 ;again
8168 JP LOOP10 ;next row
816B NOP
816C NOP
816D NOP
816E LSCR: LD HL,(DATAE)
8171 LOOP1: LD DE,DATA1 ;end ad
8174 PUSH HL ;save
8175 AND A ;clear carry
8176 SBC HL,DE ;is add end
8178 POP HL ;return it
8179 RET Z ;exit if finisd
817A LD A,(XCT) ;num of char
817D LD D,A ;to scroll
817E DEC HL ;point to last
817F AND A ;clear carry
8180 LD C,0 ;cls carry byt
8182 LOOP14: LD B,8 ;byte/char
8184 RL C
8186 LOOP13: RL (HL) ;shift/scroll
8188 RL C
818A DEC HL ;next byte
818B DJNZ LOOP13 ;do all
818D DEC D ;char row dwn
818E JP NZ,LOOP14 ;next ?
8191 DEC HL ;pass high add
8192 JP LOOP1 ;next line
8195 NOP
8196 NOP
8197 NOP
8198 DSCR: LD HL,DATA1
819B LD DE,B
819E AND A
819F SBC HL,DE
81A1 LD (BUF1),HL
81A4 LOOP18: LD BC,(DATAE)
81A8 AND A
81A9 SBC HL,BC
81AB RET Z
81AC LD HL,(BUF1)
81AF LD DE,B

```



MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

81B2	ADD HL,DE	8224 SETUP:	LD DE,0 ;zero VDP	82B0	JP DIR
81B3	LD (BUF1),HL	8227 VRAM:	PUSH AF	82B3	NOP
81B6	INC HL	8228	PUSH BC	82B4 DIR4:	LD A,191 ;test down also
81B7	INC HL	8229	LD A,E	82B6	LD IX,LEFT
81B8	LD A,(YCT)	822A	OUT (2),A	82BA	LD D,03
81B8	LD D,A	822C	LD A,D	82BC	JP TEST
81BC	LD E,0	822D	ADD A,#40	82BF	NOP
81BE LOOP16:	LD B,4	822F	OUT (2),A	82C0 LEFT:	LD D,04
81C0 LOOP15:	LD A,(HL)	8231	POP BC	82C2	JP DIR
81C1	LD (HL),E	8232	POP AF	82C5	NOP
81C2	INC HL	8233	RET	82C6 DIR0:	LD A,191 ;test down also
81C3	LD E,(HL)	8234 VRANA:	PUSH AF	82C8	LD IX,RIGHT2
81C4	LD (HL),A	8235	LD A,E	82CC	LD D,01
81C5	INC HL	8236	OUT (2),A	82CE	JP TEST
81C6	DJNZ LOOP15	8238	LD A,D	82D1	NOP
81C8	DEC D	8239	OUT (2),A	82D2 RIGHT2:	LD D,0
81C9	JP Z,LOOP18	823B	POP AF	82D4	JP DIR
81CC	LD A,(BUF)	823C	RET	82D7	NOP
81CF	LD C,A	823D DATA:	PUSH AF	82D8 DOWN:	LD D,02
81D0	ADD HL,BC	823E	LD A,C	82DA	JP DIR
81D1	LD C,6 ;deduct over	823F	OUT (1),A	82DD	NOP
81D3	AND A	8241	POP AF	82DE HOME:	LD A,191 ;test down also
81D4	SBC HL,BC	8242	RET	82E0	LD IX,HOM2
81D6	JP LOOP16	8243 DATAA:	PUSH AF	82E4	LD D,02
81D9 DATAE:	DS 2	8244	IN A,(1)	82E6	JP TEST
81DB BUF1:	DS 2	8246	LD C,A	82E9	NOP
81DD	NOP	8247	POP AF	82EA HOME2:	LD D,#FB
81DE	NOP	8248	RET	82EC	LD C,#FB
81DF USCR:	LD HL,(DATAE)	8249 JOYST:	LD A,251 ;test up	82EE	LD E,#FB
81E2	LD DE,8	824B	LD IX,USU ;if not	82F0	RET
81E5	ADD HL,DE	824F	LD IV,DIR6 ;yes test left	82F1	NOP
81E6	LD (BUF1),HL	8253	LD D,#FF ;set flag	82F2 TEST:	LD C,A
81E9 LOOP21:	LD BC,DATA1	8255	JP TEST	82F3	OUT (5),A
81EC	DEC HL	8258	NOP	82F5	IN A,(6)
81ED	AND A	8259 USU:	LD A,247 ;test left	82F7	LD B,A
81EE	SBC HL,BC	825B	LD IX,USL ;if fail	82F8	IN A,(5)
81F0	RET Z	825F	LD IV,DIR4 ;test down also	82FA	CP 127
81F1	LD HL,(BUF1)	8263	JP TEST	82FC	JP NZ,JUMP
81F4	LD DE,8	8266	NOP	82FF	LD A,B
81F7	AND A	8267 USL:	LD A,239 ;test left	8300	CP 3
81F8	SBC HL,DE	8269	LD IX,USR	8302	JP NZ,JUMP
81FA	LD (BUF1),HL	826D	LD IV,DIR0 ;test right	8305	LD A,D
81FD	DEC HL	8271	JP TEST	8306	CP #FF
81FE	LD A,(YCT)	8274	NOP	8308	JP Z,OK
8201	LD D,A	8275 USR:	LD A,223 ;test home	830B DIR:	LD E,#FB
8202	LD E,0	8277	LD IX,USH	830D	NOP ;Test for fire
8204 LOOP20:	LD B,4	827B	LD IV,DOWN	830E	LD A,223
8206 LOOP19:	LD A,(HL)	827F	JP TEST	8310	OUT (5),A
8207	LD (HL),E	8282	NOP	8312	IN A,(6)
8208	DEC HL	8283 USH:	LD A,191 ;test down	8314	LD B,A
8209	LD E,(HL)	8285	LD IX,USD	8315	IN A,(5)
820A	LD (HL),A	8289	LD IV,DOWN	8317	CP 127
820B	DEC HL	828D	JP TEST	8319	JP NZ,NONE
820C	DJNZ LOOP19	8290	NOP	831C	LD A,B
820E	DEC D	8291 USD:	CALL #79 ;scan keys	831D	CP 3
820F	JP Z,LOOP21	8294	RET ;A=asc value	831F	JP NZ,NONE
8212	LD A,(BUF)	8295	NOP	8322	LD C,#FB
8215	LD C,A	8296 DIR6:	LD A,247 ;test left also	8324	RET
8216	AND A	8298	LD IX,RIGHT	8325	NOP
8217	SBC HL,BC	829C	LD D,05	8326 JUMP:	JP (IX)
8219	LD C,6	829E	JP TEST	8328	NOP
821B	ADD HL,BC	82A1	NOP	8329 OK:	JP (IV)
821C	JP LOOP20	82A2 RIGHT:	LD A,239 ;test right to	832B	NOP
821F	NOP	82A4	LD IX,UP	832C NONE:	LD C,00
8220	NOP	82A8	LD D,07	832E	RET
8221	NOP	82AA	JP TEST		
8222	NOP	82AD	NOP		
8223	NOP	82AE UP:	LD D,6		

MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

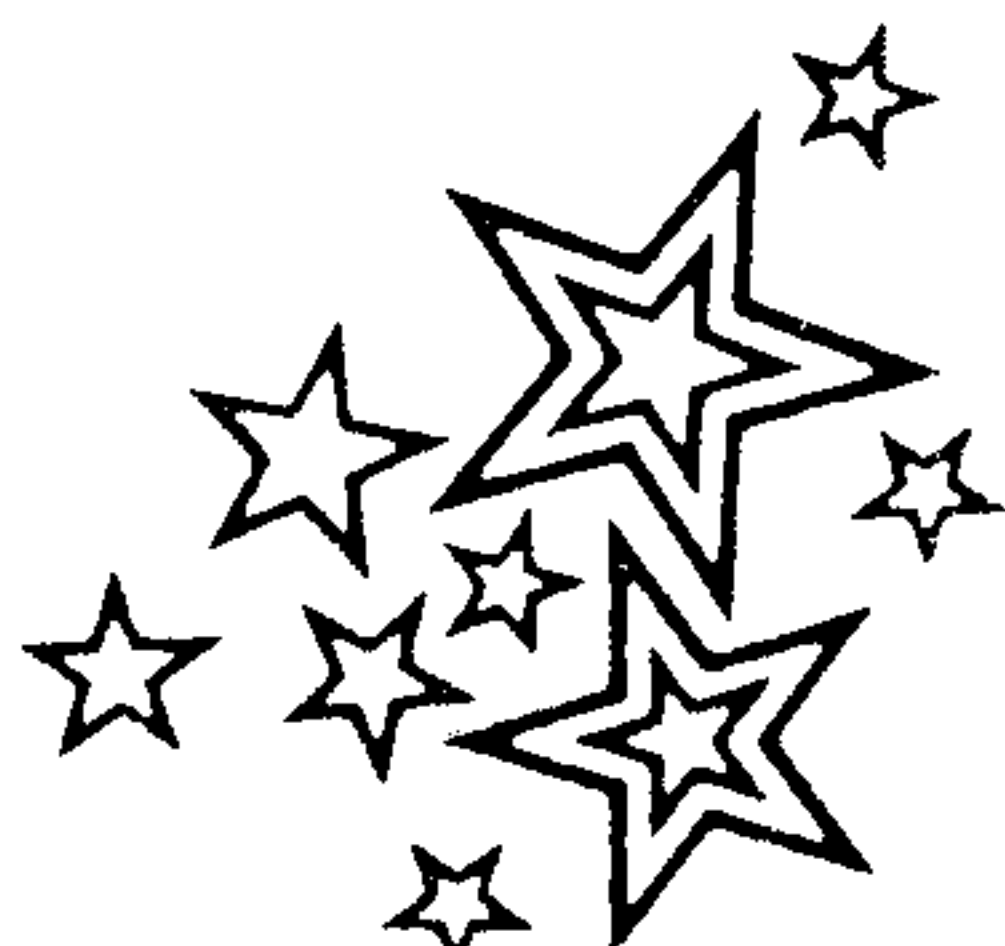
```

832F      NOP      ;Exit with
8330      NOP      ;D= dir. (0-7)
8331      NOP      ;E= dir set fg
8332      NOP      ;C= fire flag (FF=set)
8333      NOP      ;if D=FF no direction
8334      RET
8335      NOP
8336      JP M, £3A61
8339 DATA1: DS 254
8437      DS 254
8535      DS 254
8633      DS 254
8731      DS 254
882F      DS 254
892D      DS 254
8A2B      DS 254
8B29      DS 254
8C27      DS 254
8D25      DS 254
8E23      DS 254
8F21      DS 254
901F      RET

200 VS 4: CLS
204 POKE 16457,3: REM No of screen
205 CRVS 3,1,7,2,12,8,32
208 LET X=USR(16414): REM set VS 3 to scroll via mc
210 GOSUB 300
218 LET S=1
219 VS 3: CLS : GOTO 245
220 POKE 16590,S: REM No. of pix to scroll
230 LET X=USR(16400)
235 VS 3: CLS
240 INPUT "enter number of pixels to scroll at between say 1&8 ";S
245 PRINT "back in the m.c. scroll"
250 IF S<1 THEN STOP
260 IF S>8 THEN LET S=8: PRINT "don't over do it"
270 GOTO 220

300 VS 4
305 PRINT "This is an example of how the "
310 PRINT "screen may be manipulated via"
320 PRINT "machine code routines."
330 PRINT " By pressing the cursor keys"
340 PRINT "a section of the screen as "
350 PRINT "defined by VS 3 (or any other"
360 PRINT "graphic screen) may be made to"
365 PRINT "scroll."
366 PRINT
370 PRINT " Also included is a routine"
380 PRINT "to work the joystick in the"
390 PRINT "eight positions expected."
395 PRINT
400 PRINT "press any key to start "
410 PRINT " pressing any of the key"
420 PRINT " pad numbers 0-6 + . & cls"
430 PRINT " or any two together will"
440 PRINT " scroll in that direction"
460 PRINT " press 'S' to exit m.c"
470 PRINT " enter 0 to scroll to stop"
590 IF INKEY$="" THEN GOTO 590
595 IF INKEY$="S" THEN STOP
600 RETURN

```



THIS SPACE WOULD
ONLY COST YOU £12.

details from Genpat
3, Bulcock Street,
Burnley BB10 1UP
0282 57427

Keep your customers
informed.....

The listing on the previous pages was sent in By D.A. BUCK from BEDFORDSHIRE.
The program performs a screen scroll in **all** directions. Mr. Buck has also supplied a JOYSTICK ROUTINE which will interface with the SCROLL ROUTINE.

I, personally, think that this is a fantastic program, and he has accomplished the impossible..... try it..... you'll see what I mean.

This is definitely the star program of this month..... top marks and I look forward to more submissions from D.A.BUCK.....

MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

Tip

A request for help posed a problem earlier this week: ' How do you simulate the **Instring\$** function on the 512 ?'. Yes, well, er.....

```
B$ = Search $string
A$ = $string to search
10 LET B = LEN(B$)
20 LET L = LEN(A$)-B
30 FOR I = 1 TO L
40 IF A$(I,B) = B$ THEN GOTO Found
50 NEXT
```

ATTENTION SOFTWARE WRITERS

DON'T FORGET..... SYNTAX SOFTWARE IS LOOKING FOR ALL TYPES OF SOFTWARE TO MARKET THROUGH THE CLUB. EXCELLENT ROYALTIES WILL BE PAID TO AUTHORS WHO ARE PUT ON CONTRACT.

SPECIAL NOTICE RST 10 CALL NUMBER 18

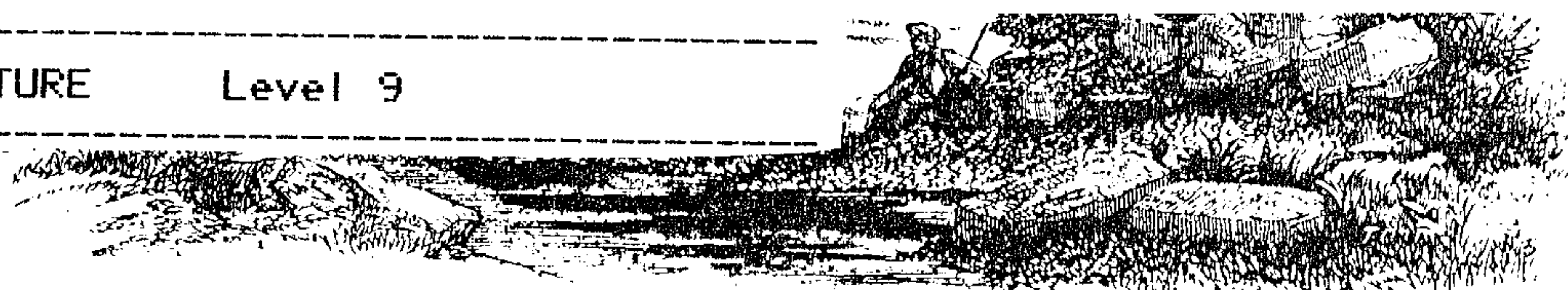
I have had lots of calls about the RST 10 information sheet that was posted to those members who requested it. I am pleased to announce that we have now solved the elusive bug.

The answer lies in the fact that this command expects two byte words for the x,y data. The format for using this instruction is :-

```
RST 10
DB £8A
DB P,LSB FOR X,MSB FOR X,LSB FOR Y,MSB FOR Y
DB LSB X1,MSB X1,LSB Y1,MSB Y1,COL
```

We have tested this routine extensively, and in every situation it was found to perform satisfactorily.

COLOSSAL ADVENTURE Level 9



Don't buy this adventure if you are studying for your exams, or your wife is complaining that you are neglecting her. I was also going to review **Snowball** for this edition, but I am sure that I am prejudiced in some way, in favour of Level 9 and so I have handed over the review to another member who may have a different outlook on the game. **Snowball** will be reviewed in the next edition.

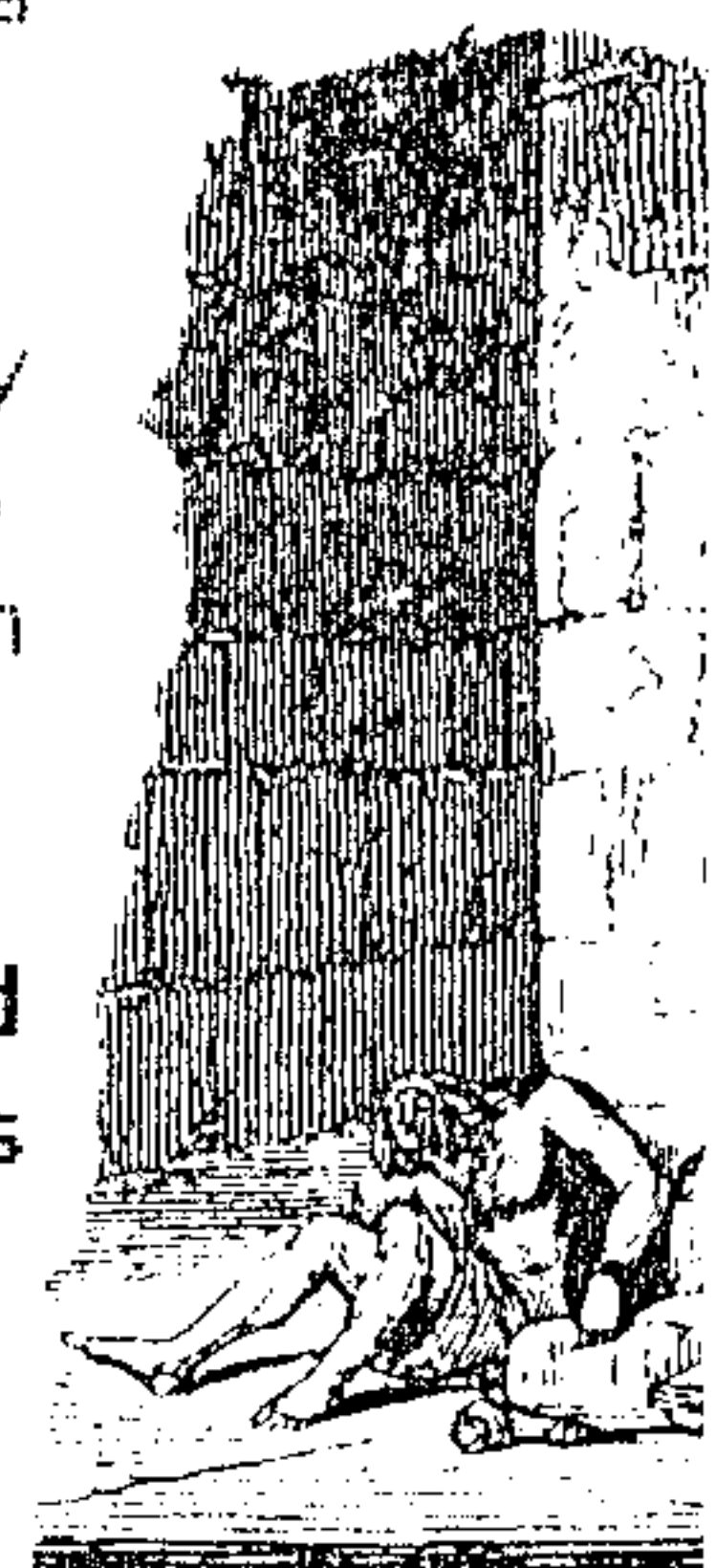
I cannot find a thing to complain about with **Colossal Adventure**. It is colossal, and it is fantastic in its conception. I have had umpteen attempts at completing the course, but this itinerant always comes a cropper!

The adventure starts one evening when a 'travel-stained warrior' enters your local watering hole. He pays for a drink with an 'absurdly high valued gold coin'. Over his drink he claims that he has been to **Colossal Cavern** and escaped with his life! The other occupants of the tavern conspire to relieve him of his wealth the moment he leaves. You, being the saintly person you are, decide to help him.

The stranger offers you a crumpled map with the location of The Cavern as your reward. Thus the adventure starts.....

This weary adventurer has been set on by evil dwarves, stumbled over rocky landscapes, visited the Hall Of Mists, and at one time I held, in my hand, the elixir of life.....at one time I almost made it only to be eaten alive by a giant..... well that would be telling!

I guarantee you will not treat this game lightly. It is pure fantasy, and a sheer delight. Full marks to Level 9 for another first class adventure.



SPECIAL OFFER TO ALL GENPAT MEMBERS:

MEMOTECH DMX80 PRINTER
PRINTER CABLE
BRUNWORD
ONLY£239.00

SEND CASH WITH ORDER TO GENPAT AND ALLOW 7 DAYS FOR DELIVERY

INSIDE THE MTX

MTX512

If you have ever re-configured VRAM to your own format, you will know what a pain it is to design, and load all the Ascii characters back into your own **Pattern Generator Table**. Well! Toil no more. Read on.....

The MTX ROM stores its **Ascii characters** from location £35B3 on page zero. Special precautions should be taken as the characters are stored in **5 bytes** which means that they must be rotated to allow them to be printed to the screen. The coding below does all the hard work for you. Obviously, immediately before this code you should have sent the VRAM address to the VDP chip. **ASCII = £35B3** and **PORTDO = 1**. In the listing I have had to code each step, but you can replace the **LD E,(HL) :INC HL:LD D,(HL):INC HL** with **RST 8** because this is exactly what the **RST8** call does: it loads the E reg with the contents of the memory location pointed to by HL then incs HL, and LDs reg D with the contents of memory location pointed to by HL then increments HL again before returning to caller.

TITLE ARCADIAN ;DATE 30.5.84

;AUTHOR:- KEITH HOOK

;ASEG
;

EXT ASCII

PORTAO EQU 2
PORTDO EQU 1

SENASC:

; SEND DATA TO VRAM
; HIDDEN IN 1900H (TEXT RAM)
;

LD HL,5900H
CALL ADDOUT
LD HL,ASCII
LD B,96

SENASI:

PUSH BC
PUSH HL
LD C,(HL)
INC HL
LD E,(HL)

INC HL
LD D,(HL)

INC HL ; RST 8

LD A,(HL)

INC HL

LD H,(HL)

LD L,A

LD B,08

; LOAD REGISTERS WITH 5 CHARS

SENAS2:

XOR A

RL H

RR A

RL L

RRA

RL D

RRA

RL E

RRA

RL C

RRA

OUT (PORTDO),A

DJNZ SENAS2

POP HL

LD C,5

ADD HL,BC

POP BC

DJNZ SENAS1

ADDOUT:

LD A,L

OUT (PORTAO),A

LD A,H

ADOUT2: OUT (PORTAO),A

RET

SYSTEM VARIABLES £FAB5 USYNT & £FAB9 USER

These two addresses are used to interface your own routines with MTX basic. £FAB9 is loaded with a jump to your own routine. The syntax for your new command must be USER <NAME> where NAME can be a new basic command e.g. `FILL.====> USER FILL <parameters>.`

When Basic encounters your new command it will then check with £FAB5 to check what the syntax should be for the new command. Basic will check for various syntax depending on what has been loaded into £FAB8 down to £FAB5. The syntax bytes are as follows:

- 0 => Expect a numeric expression
- 1 => Expect a string expression
- 2 => Expect an arithmetic expression
- 3 => Expect a list of expressions separated by ',' or ';'.
- 4 => Expect a list of numbers separated by ',' from 0 - 64K: FILL 93,2,1
- 5 => Expect a list of arithmetic expressions
- 6 => Expect a single number in range 0 - 64K
- 7 => Check Nothing

There are various other values for the byte but those listed above are the main ones.

If your command was going to check nothing you would load £FAB8 with 7 and £FAB7 with £C9 (RET).....the last byte must always be a RET instruction.

Suppose the parameters for your routine required a series of numbers along the lines of the GENPAT statement e.g

USER FILL 3,5,234,8,8

Then you would set up USYNT as follows

FAB8 3
FAB7 £C9

Step One: Load USER with JP to your routine.

Step Two: Load USYNT starting at last byte £FAB8 with syntax checking bytes.

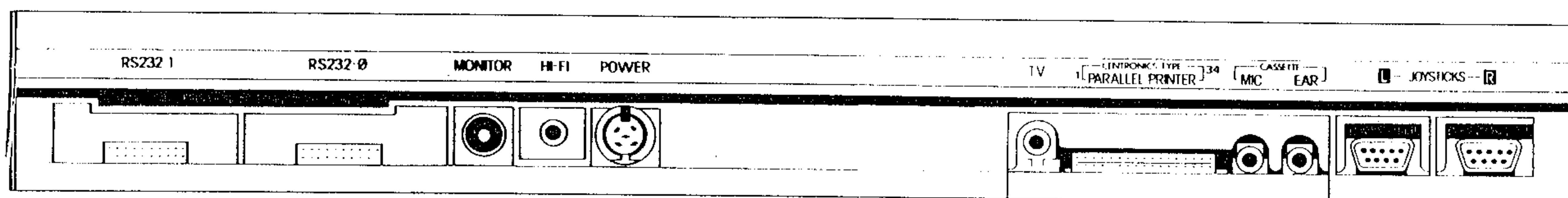
Step Three: Make sure last location of USYNT is loaded with a RET instruction.

Also note that £FAB9 is loaded LSB/MSB notation

Any one not quite sure how to go about this can send for a full listing from Genpat.

The easiest way is to use syntax byte 7 until you have debugged the routine then insert the syntax checking.

MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.



END STATEMENT

It has been an exciting month. Club membership continues to grow. This month we must take time to welcome our international friends from Holland, Norway, and Belgium.

I do hope you will make time to fill in the questionnaire and return it to Genpat as soon as possible. It will help Genpat, and Memotech to plan ahead in a way that will be beneficial to all members.

Some of the articles planned for this month have been held over until the next edition. However, I am sure you will agree that a lot of work has gone into producing Issue 2. Next month we shall focus on **Panel, Sprites, & converting Connect 4.**

Please support the companies who have devoted their time to designing quality programs i.e **Brunning, Membrain, Tricom Soft etc.** It is not an easy decision to back a minority (tongue in cheek) machine and neglect the more popular models. MTX users tend to be a bit lay-back i.e no program listings in magazines, and the poor response to our request for programs..... **get cracking.....the only way you are going to get software houses supporting you is to support them !!!!!!!** Also, get those letters sent into magazines bring the MTX to the attention of the uninformed. One of our members - who doesn't even have a MTX yet! - sent a marvellous letter to PCN (Issue 83).

I've just sprained my ankle jumping off the soap-box. My sincere thanks to all you people who have written in with programs and comments.... please keep them coming.

Finally, **Grafics** will be released within the next 7 days and all members who have already ordered the program will receive their copies during this period.

You will be pleased to know that after consulting with Memotech, in next months edition, we hope to solve the **DATA SAVE & LOAD [over a page]** problem.

NO ONE HAS MANAGED TO WIN THE 'WHERE DID THE MTX GET ITS NAME ?' COMPETITION. WE WILL CARRY THIS OVER INTO THIS MONTH. FREE SOFTWARE FOR THE WINNER.

Chief Advisor to Genpat : Jeff Wakeford * Pascal Consultant: Stephen Varley * Technical Consultant: Geoff Boyd [Memotech] * Printing Consultant: Chris Love [Love's Printing Services] * Cover Design: **MIKE** [Memotech] * Chief Tea Maker: Patricia [My wife] * Reviewers: Robert Peet, Colin Rees, Alan Staines. <C> Genpat 1984