memo

Memotech Computer User Club Magazine

ISSUE:

HOW TO USE FRONT PANEL

LUNA LANDER

TEACH YOURSELF ASSEMBLY KIT

ALL ABOUT SPRITES

COMPETITION

EDITORIAL

Well. After a week down at Memotech it's time for issue three. Time seems to go nowhere, and no matter how hard we work, we never seem to be able to get rid of the backlog.

Due to the tremendous success of the **printer offer** we have decided to continue the same into this month, this will give all the new members a chance to take advantage of this excellent package. Talking of new members, some of you are unaware that you can purchase **hardware** & **software** at reduced prices through the club, if you are intending to buy some new equipment ring Genpat for a price.

In last month's edition we stated that we had stocks of the **new User Manual**, unfortunately in the time that lapsed between writing and publication, Memotech withdrew their approval due to a technical difficulty. This has still not been resolved, and I ask you to be patient until the matter can be resolved. All members who have ordered the manual will most definitely receive it before any other orders are processed.

The **software** scene is looking very promising: 4 new titles have been issued by one of the most look ahead companies in this country, **Sentient Distribution**, and they have some excellent ideas lined up including a **Flight Simulator & Logo. Peter Brunning** has not been idle since he released **Brunword**, he is almost ready to release his **Data File** program which seems an excellent package. We have also negotiated a deal with **Xaversiene** and their **Composer** can now be ordered through the club at a discount rate of 10%. However, there will be a waiting period of 21 days for delivery, and distribution will be on a first come, first served basis. **Solway Software** have agreed to supply the club with their two titles, and a full review of the programs will be published when we receive review copies.

Memotech are busy working on **three Disc Drives** which will be on sale at an incredible price of £199 to £399 with a very special offer to Genpat members. Watch the post for further details.

Salty Sam is the first game available from **Syntax Software** and will be released within the next fourteen days at £4.95.

On my recent visit to Witney it was pleasing to see that the Company was having a great success in Denmark, Finland, & Holland, and that the home market is at last recognising what a good computer the MTX is !

Finally, I would like to thank all you excellent people who returned the questionaire from the back of the magazine..... every form was read and all the information was collated. In this edition you will see that we have taken some of your points to heart. Thanks to all members who have submitted programs for publication, and to those of you who who have entered the competitions.

MEMOPAD

MEMOTECH

MTX SERIES

ADVERTISING RATES

WHOLE PAGE.....	£70.00	272mm	* 185 mm
HALF PAGE.....	£35.00	134mm	* 185 mm
QUARTER PAGE ..	£24.50	134mm	* 90 mm
EIGHTH PAGE ..	£12.00	65mm	* 90 mm

CLASSIFIED @ 60p per line.

DISCOUNT FOR SERIES

UP TO 12 INSERTIONS 20%
UP TO 9 INSERTIONS 15%
UP TO 6 INSERTIONS 7.5%

Please note that the above rates apply for MONO & CAMERA READY ARTWORK. BLACK & WHITE. COLOUR PRICES ON APPLICATION ONLY.

COPY DATES 15th OF EACH MONTH FOR FOLLOWING MONTH EDITION

CHEQUES MADE PAYABLE TO **GENPAT** with camera ready artwork
THE EDITOR, MEMOPAD, 3, BULCOCK STREET, BURNLEY BB10 1UH
Telephone 0282 57427

Printed by Genpat 3,Bulcock Street, Burnley BB10 1UH.

BACK ISSUES: BACK ISSUES

Issue 1 80p
Issue 2 £1.00
Prices are inclusive of postage and packing. Subject to availability.
Cheques with orders to GENPAT.

Do not forget to include a 50p set of stamps etc if you require any photocopies.

FACT SHEETS AVAILABLE ON:-
RST 10 COMMANDS
SYSTEM VARIABLES.
INTERRUPTS.
AND ANY PART OF ROM AT 10 per copy. RST 10,SYSTEM VARIABLES & INTERRUPTS 50p for set. N.B ROM sheets are at 10p per sheet.

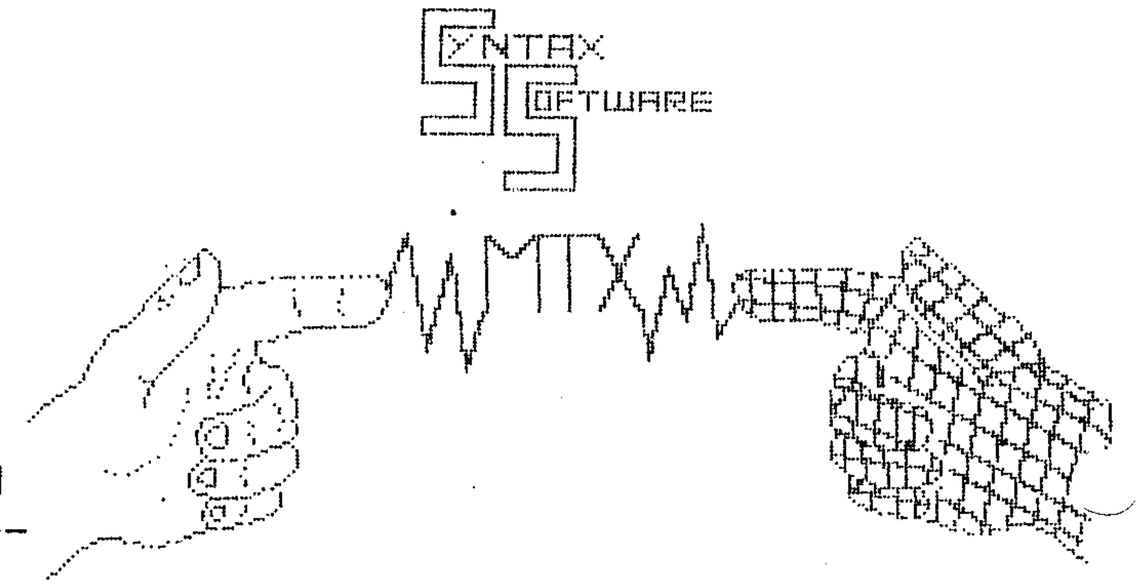
PLEASE MAKE SURE YOU INFORM US ON WHICH MODEL YOUR PROGRAM HAS BEEN WRITTEN WHEN SENDING SUB? MISSIONS TO GENPAT. THIS IS VERY IMPORTANT.....

NOBODY WON THE COMPETITION "WHERE DID THE MTX GET IT'S NAME"...
Mind you, I didn't expect that there would be a winner !
The MTX was taken from the die cast number which was stamped on the very first sample of the MTX casing. However, I would like to award one piece of software to the most original answer. Not the most favourite one taken from Ian Sinclair's book. but this one submitted byRobert Heydon.

A pack of 26 cards were dealt with letters on, and put in alphabetical order.
12 cards were dealt and the next placed face up.
6 cards were dealt and the next placed face up.
3 cards were dealt and the next placed face up.
THE CARDS FACE UP SPELL MTX.

Syntax Software Competition

This competition really did bring out the creativity in you. It was a real hard task to choose the winning entry, but after many hours looking at the four finalists the winner of the competition is **MR R.A.MITCHELL member C573.** This is an excellent title page and a screen dump appears below. If Mr. Mitchell will contact Genpat we shall be pleased to send him the software of his own choice from Continental's software list.



WINNER OF THE QUESTIONAIRE COMPETITION

) The first questionaire opened on the 17th - yes! I know I said the 15th, but I was still down at Witney on that date - was from **member C814 Mr. B. Midcalf from Otley.** Two pieces of software are now on the way, and thanks again, to all who took the trouble to return the form.

RAFFLE : WINNING £1 NOTE for DMX80 PRINTER

The raffle was a real disappointment. It started like an hurricane and ended like a damp squib. However,we shall run it for a further month to see if the response does increase.

The winner of this month's raffle is:-**Mr.M.W.Divall Membership number C647**

He will receive a **DMX 80 Printer** which will be despatched within the next few days. Well done !

Incidently, this raffle was drawn by someone who has no connection with the club.... in fact, it was a customer from my wife's take-a-way who was press-ganged into pulling the £1 note out of the hat!

START SENDING YOUR £1 NOTES NOW FOR THE NEXT RAFFLE WHICH WILL BE DRAWN ON 12TH DECEMBER IN ORDER TO CATCH CHRISTMAS POST.....SO DON'T DELAY.

PROGRAM

TAKE THE PLOD OUT OF NODDY BY E.ROY

Screens of text are very easy to set up on the Memotech thanks to the resident text handling language **Noddy.** However, setting up screens of text that are to be sent to the printer is much harder, if, like myself, you own a printer that does not allow the **number of characters per line** to be changed from **80 to 39** as described in the manual. Also, unlike other word processors, Noddy does not allow imbedded print control codes.

The following program: **Noddy to Printer Formatter** is written for the **Seikosha GP-100A MkII,** but will work on any make of printer if two lines of the listing are changed.

The program provides the following functions:-

1. A number of Noddy pages can be created, and output as a formatted document.

2. The number of copies required can be set.

3. Single pages can be Llisted or formatted to the printer.

4. The number of characters per line can be changed from page to page.

5. Each page can be can be printed in **single or double** line spacing.

6. Control characters can be detected and type face changed from line to line within a page.

The Noddy ***LIST** is not used to send the page to the printer, instead the commands ***DISPLAY PAGE NAME & *RETURN** are used to display the text on the screen and return to basic. The **SPK$** (screen peek) command is used to read the characters from the screen into a $string which is then sent to the printer. It is this latter feature that enables **control characters** to be detected. Unfortunately, the **function keys** cannot be redefined and used as control characters - this is due to the way Noddy compressses, and stores the number of space characters between words.

E.g 2 space are compressed and held in one byte as 129 - the same code as function key F2.

There are, however, a number of characters on the top right of the keyboard that are not commonly used in text, and I have used the { & } characters for the two print commands available on the Seikosha.

} Centre one line on paper.

{ Send one screen line in **double width.**

Before using the Formatter,Noddy pages must always be set up as follows.

1. The TEXT screen must be redefined to allow 40 characters by 23 line. (Running the program and selecting **Option A** will take care of this.)
2. Once in the **Noddy Editor** enter the page name as usual, and then delete the name that appears on the **top** line. The top line of each page is used as the command line and **must** always contain the Noddy commands **:*DISPLAY PAGE NAME. RETURN. ..... SEE DEMO NODDY PAGES AT END OF MAIN LISTING.** This prevents a separate Noddy program page containing these commands from being written to display the page being sent to the printer.

Five additional command characters can also be included in the command line. These are:- **4 8 @ 1 2** and have the following functions.....

4 Send this page to printer 40 characters wide.
8 Send this page to the printer 80 characters wide.
@ Force a new sheet of paper before printing this page.
1 Print this page with single line spacing.
2 Print this page with double line spacing.
Note: The command characters 4 8 1 & 2 will stay in effect ntil changed by the next command line. These commands **must** appear on the **top** line only, and inserted **before** the ***DISPLAY** command. E.g. **8 *DISPLAY**

The remaining 22 lines are available for text. This allows 22 lines of 40 column output, or 11 lines of 80 column output to the printer. It is in this area of the screen that the printer control characters **{ & }** can be inserted, and for reasons of speed these must always be the first character on that line. Both the above control characters cause the next 39 characters to be sent to the printer, so care should be taken if the page being sent is 40 columns wide. That is to say, printing 21 characters in double width mode will work, but the layout of the document may be spoiled.

Once the page has been created, exit from the Noddy Editor in the normal way. This will return you to basic where you should enter the Noddy Pages to be sent to the printer in **DATA LINES** starting at line **2530.** Care should be exercised when entering, or editing **DATA** lines - **only ONE space should appear between DATA & the contents of the line.** The DATA lines should be set up in the correct order as per **Listing Lines 2530 to 2700.**

DOCUMENT 1 ==== NAME OF DOCUMENT : HEADING, ADDRESS,TEXT1,GRAPH,TEXT2 & ENDING are the Noddy pages that make up DOCUMENT 1. XXX === End of Document Marker. FINI === end of DATA Lines.

After the DATA lines have been inserted **RUN** the program, and select one of the options from the **Menu** to output the document, or page as required. Selecting **option B** will ask you for the document name i.e DOCUMENT 1. This will output the Noddy pages between the **document name & XXX,** as specified in the DATA lines, as a formatted document. When asked for number of copies your input must always be 1 or more.

Option c will also ask you to enter a name. However, this time, it requires a Noddy page name i.e HEADING, ADDRESS etc. You will then be asked if the page is to be **Llisted or Formatted.** The **Llist option** will llist the page to the printer exactly as per the screen. The **Format option** will send the page to the printer, but will take into account any command line or print controls that may appear.

If you make a spelling mistake or can't remember the document while in **option B or C** just enter **HELP** or any name that is not in the DATA lines and the contents of the DATA lines will be displayed.

Option D allows you to save the program plus Noddy pages to tape. The saved file will auto-run on reloading. Use **Option E** to exit from the program.

If you own a printer other than a Seikosha GP-100A change Lines 1760 and 1860 as required by your printer to give the following print modes :
The **Lprint String in line 1760** will print the **C$** to paper starting 20 characters in from the left-hand edge.**Line 1860** contains another **Lprint string** which switches the Seikosha to Double Width Mode. CHR$(14) then prints C$. CHR$(15) switches the printer back to normal mode.

The **LF$** will contain one of two character codes: CHR$(10) = Print a **Line Feed after printing C$.** or CHR$(20) = **Don't print a Line Feed after printing C$.** This allows the use of single or double line spacing..... these codes must also be changed if not compatable with your printer.

Main Variables:
COM$ ..... Command line character to be tested.
C$ ..... String to be sent to printer.
LF$ ..... Send line feed or no line feed to printer.
PN$ ..... Noddy document or page number.
AN$ ..... Holds answer from input.
LF ..... Llist flag. True = llist page. False = format page.
LL ..... Length of line to b read by SPK$.
CR ..... Counter for Line Feed/Carriage Return.
NCR ..... Increase CR by this amount for every line feed.
PL ..... Paper length between perforations.
EP ..... End print of text on paper.
NC ..... Number of copies to be printed.
CT ..... Counter for number of copies.
NL ..... Number of screen lines actual page takes up.
LC ..... Screen line counter.

Any futher information on this program can be obtained from Mr. E. Roy, 1 Orchard Street,Kilmarnock, Ayrshire KA3 1EB.

```
100 REM ******************************
110 REM ****   NODDY TO PRINTER   ****
120 REM ****      FORMATTER.      ****
130 REM ****  by E.Roy   Aug. 84  ****
140 REM ******************************
150 REM
160 DIM C$(80),COM$(1),LF$(1)
170 REM-----------------------------
180 REM  Display Instruction Page.
190 REM-----------------------------
200 CRVS 5,0,0,0,40,23,40: PAPER 5: CLS
210 PRINT "     --- NODDY PAGES TO PRINTER ---": PRINT : PRINT
220 PRINT "  DO YOU WISH TO.....": PRINT
230 PRINT "     A.....ENTER NEW NODDY PAGES.": PRINT
240 PRINT "     B.....SEND DOCUMENT TO PRINTER.": PRINT
250 PRINT "     C.....LLIST, FORMAT ONE PAGE.": PRINT
260 PRINT "     D.....SAVE PROGRAM TO TAPE.": PRINT
270 PRINT "     E.....EXIT FROM PROGRAM.": PRINT
280 GOSUB 2430
300 REM-----------------------------
310 REM  Get Input and Take Action.
320 REM-----------------------------
330 INPUT "  PRESS KEY A,B,C,D,E then <RET> ";AN$
340 PRINT "                                      "
350 LET KP=ASC(AN$)-65
360 IF KP<0 OR KP>4 THEN  CLS : GOTO 210
370 IF KP=0 THEN  NODDY
380 ON KP-1 GOTO 430,630,930,1030
400 REM-----------------------------
410 REM  Send Document Pages.
420 REM-----------------------------
430 GOSUB 830: IF FD=FALSE THEN  GOTO 200
440 PRINT
450 INPUT "  HOW MANY COPIES TO PRINT ? ";NC
460 IF NC<=0 THEN  GOTO 450
470 LET CT=0: LET LF=FALSE
480 READ PN$
490 IF PN$="XXX" THEN  GOTO 550
500 GOSUB 1180
510 GOTO 480
520 REM-----------------------------
530 REM  Count Number of Copys.
540 REM-----------------------------
550 LET CT=CT+1
560 IF CT<>NC THEN  GOSUB 2150: GOSUB 850: GOTO 480
570 GOSUB 2150
580 GOTO 200
600 REM-----------------------------
610 REM  Llist or Format One Page.
620 REM-----------------------------
630 GOSUB 830: IF FD=FALSE THEN  GOTO 200
640 LET PN$=AN$
650 PRINT
660 INPUT "  LLIST or FORMAT PAGE? press L or F. ";AN$
670 IF AN$="F" OR AN$="f" THEN  LET LF=TRUE: GOSUB 1180: GOTO 200
680 PLOD PN$
690 LET NL=PEEK(65450): LET LC=0
700 LPRINT : LPRINT
710 LPRINT " Noddy Page Name = ";PN$: LPRINT
720 CSR 0,0
730 FOR L=1 TO 40
740 LET C$(L)=SPK$
750 NEXT L
760 LPRINT C$: LET LC=LC+1
770 IF LC<=NL THEN  GOTO 730
780 LPRINT : LPRINT
790 GOTO 200
800 REM-----------------------------
810 REM  Check Document, Page Name.
820 REM-----------------------------
830 PRINT : PRINT
840 INPUT "  ENTER NAME then <RET> ";AN$
850 RESTORE 2530: LET FD=FALSE
860 READ N$
870 IF N$=AN$ THEN  LET FD=TRUE: RETURN
880 IF N$="FINI" THEN  GOSUB 1930: RETURN
890 GOTO 860
900 REM-----------------------------
910 REM  Save program to tape.
920 REM-----------------------------
930 PRINT : PRINT
940 PRINT "  ENTER NAME, SELECT RECORD then <RET>": PRINT
950 INPUT "  PROGRAM NAME  ";AN$
960 SAVE AN$
970 CLEAR
980 GOTO 160
990 REM
1000 REM-----------------------------
1010 REM  Exit from Program.
1020 REM-----------------------------
1030 PRINT : PRINT
1040 INPUT "  ARE YOU SURE ? press Y or N. ";AN$
1050 IF AN$="Y" OR AN$="y" THEN  CLS : STOP
1060 GOTO 200
1100 REM ******************************
1110 REM ****  FORMAT NODDY PAGES  ****
1120 REM **** TO SEIKOSHA GP-100 A ****
1130 REM ****       PRINTER.       ****
1140 REM ******************************
1150 REM-----------------------------
1160 REM  Display Noddy Page.
1170 REM-----------------------------
1180 PLOD PN$
1190 LET NL=PEEK(65450)
1200 LET LC=1: LET C$=""
1210 GOSUB 2230
1300 REM-----------------------------
1310 REM  Check for Control Characters.
1320 REM-----------------------------
1330 CSR 0,1
1340 LET C$=SPK$
1350 IF LC>NL THEN  RETURN
1360 IF C$="{" THEN  GOSUB 1830: GOTO 1340
1370 IF C$="}" THEN  GOSUB 1730: GOTO 1340
1380 GOSUB 1430
1390 GOTO 1340
1400 REM-----------------------------
1410 REM  Output One Line Normal Text.
1420 REM-----------------------------
1430 FOR L=2 TO LL
1440 LET C$(L)=SPK$
1450 NEXT L
1460 LPRINT C$+LF$
1470 IF LL=80 THEN  LET LC=LC+2 ELSE  LET LC=LC+1
1480 GOSUB 2130
1490 RETURN
1700 REM-----------------------------
1710 REM  Center One Line on 80 Columns
1720 REM-----------------------------
1730 FOR L=1 TO 39
1740 LET C$(L)=SPK$
```

```
1750 NEXT L
1760 LPRINT CHR$(16)+CHR$(2)+CHR$(0)+C$+LF$
1770 LPRINT ;: LET LC=LC+1
1780 GOSUB 2130
1790 RETURN
1800 REM-----------------------------
1810 REM  Output One Line Double Width.
1820 REM-----------------------------
1830 FOR L=1 TO 39
1840 LET C$(L)=SPK$
1850 NEXT L
1860 LPRINT CHR$(14)+C$+LF$+CHR$(15)
1870 LPRINT ;: LET LC=LC+1
1880 GOSUB 2130
1890 RETURN
1900 REM-----------------------------
1910 REM  Display Noddy Page Names.
1920 REM-----------------------------
1930 CLS : RESTORE 2530
1940 PRINT "  ";AN$;" IS NOT IN DATA LINES."
1950 PRINT : PRINT
1960 PRINT "  Only the following names are legal,"
1970 PRINT : PRINT
1980 READ N$
1990 IF N$="FINI" THEN  GOTO 2020
2000 IF N$<>"XXX" THEN  PRINT ,,N$ ELSE  PRINT ,,"_______": PRINT
2010 GOTO 1980
2020 PRINT : PRINT
2030 PRINT "  PRESS 'M' TO RETURN TO THE MENU PAGE."
2040 LET KP$=INKEY$
2050 IF KP$="M" OR KP$="m" THEN  RETURN
2060 GOTO 2040
2100 REM-----------------------------
2110 REM  Wind Paper on to Next Sheet.
2120 REM-----------------------------
2130 LET CR=CR+NCR
2140 IF CR<=EP THEN  RETURN
2150 IF LF=TRUE THEN  RETURN
2160 FOR L=CR TO PL
2170 LPRINT
2180 NEXT L
2190 LET CR=0: RETURN
2200 REM-----------------------------
2210 REM  Read Command Line.
2220 REM-----------------------------
2230 CSR 0,0
2240 LET COM$=SPK$
2250 IF COM$="4" THEN  LET LL=40
2260 IF COM$="8" THEN  LET LL=80
2270 IF COM$="@" THEN  GOSUB 2150
2280 IF COM$="1" THEN  LET LF$=CHR$(20): LET NCR=1
2290 IF COM$="2" THEN  LET LF$=CHR$(10): LET NCR=2
2380 IF COM$="*" THEN  RETURN
2390 GOTO 2240
2400 REM-----------------------------
2410 REM  Default variable values.
2420 REM-----------------------------
2430 LET TRUE=0: LET FALSE=-1
2440 LET LF=FALSE: LET LF$=CHR$(20)
2450 LET LL=80: LET CR=0: LET NCR=1
2460 LET PL=65: LET EP=60
2470 RETURN
2500 REM-----------------------------
2510 REM  Data Lines. End With  FINI
2520 REM-----------------------------
2530 DATA DOCUMENT 1
```

```
Noddy Page Name = GRAPH

  8 1  *DISPLAY GRAPH. *RETURN
}
}      GRAPH OF YEARS SALES TO DATE.
}
}JAN   ******************
}FEB   **************
}MAR   ************
}APR   ****************
}MAY   ********************
}JUN   **************************
}JUL   ************************
}AUG   **********************
}SEP   ************************
}OCT   **********************
}NOV
}DEC
}      0 1 2 3 4 5 6 7 8 9
}         (Scale = *10 units)
}       Printer...ACME DELUX MkII.
}
}

Noddy Page Name = HEADING

  8    *DISPLAY HEADING. *RETURN
{          ACME PRINTERS LTD.
}
}            1, ORCHARD ST.,
}              KILMARNOCK.
}              AYRSHIRE.
}              KA3  1EB
}
}        Telephone  0563 34684
}
    SALES DEPT.
                          Ref.01-1234
```

```
2540 DATA HEADING
2550 DATA ADDRESS
2560 DATA TEXT1
2570 DATA GRAPH
2580 DATA TEXT2
2590 DATA ENDING
2600 DATA XXX
2700 DATA FINI
```

Programming in Pascal

```
PROGRAM SecondArticleForGENPAT;

{      ****PRINTER EXTENSIONS****     }
{         7/10/84 S.Varley (MEMBRAIN software) }

{         PON  switch on printer     }
{         POFF switch off  printer   }
{         PRINTASC(n) prints ascii character OF n }
{         CR print carriage return empty buffer }
{         DOUBLE(0 OR 1) Double width enlarged character setting }
{         EMPHASIS(0 OR 1) SET reset emphasized print }
{         ELITE(0 OR 1) elite  style on/off }
{         REDUCED(0 OR 1) Reduced character setting }
{         DEL delete last character IN buffer }
{         BS backspace }
{         BELL sound bell on printer }
{         NINTHFEED sets 1/9th inch spacing }
{         SIXTHFEED SET 1/6th inch spacing }
{         PITCH(x) sets linefeed pitch TO x/36th OF an inch }
{         LF linefeed }
{         COLUMNS(x) sets width OF paper TO be printed on }
{         UNDERLINE(0 OR 1) underline chars on/off }
{         SUPERSCRIPT(0 OR 1) sets/resets superscript printing }
{         SUBSCRIPT(0 OR 1) sets/resets subscript printing }
{         GRAPHIC(TYPE,n) prepare printer FOR n bytes OF TYPE density graphics data }
{         NOTE TYPE is 0 standard density }
{         ----         1 double density }

{ Switch on printer }
PROCEDURE PON;
BEGIN
POKE(£FD75,CHR(1));{PRORPL=IOPL}
POKE(£FA8F,CHR(1));{IOPL=Centronics}
END;

{ Switch off printer }
PROCEDURE POFF;
BEGIN
POKE(£FD75,0)  {PRORPL=IOPR}
END;

PROCEDURE PRINTASC(x:integer);
BEGIN
write(chr(x))
END;

PROCEDURE CR;
BEGIN
writeln
END;

PROCEDURE DOUBLE(flag:0..1);
BEGIN
IF flag=1 THEN
 write(chr(14))
ELSE
 write(chr(20))
END;

PROCEDURE LF;
BEGIN
write(chr(10))
END;

PROCEDURE UNDERLINE(flag:0..1);
BEGIN
write(chr(27),'-');
IF flag=1 THEN
 write(chr(1))
ELSE write(chr(0))
END;

PROCEDURE PITCH(x:0..255);
BEGIN
write(chr(27),'A',chr(x))
END;

PROCEDURE EMPHASIS(flag:0..1);
BEGIN
IF flag=1 THEN write(chr(27),'E')
ELSE write(chr(27),'F')
END;

PROCEDURE ELITE(flag:0..1);
BEGIN
IF flag=1 THEN write(chr(27),'M')
ELSE write(chr(18))
END;

PROCEDURE REDUCED(flag:0..1);
BEGIN
IF flag=1 THEN
 write(chr(15))
ELSE
 write(chr(18))
END;

PROCEDURE DEL;
BEGIN
write(chr(127))
END;

PROCEDURE BS;
BEGIN
write(chr(8))
END;

PROCEDURE BELL;
BEGIN
write(chr(7))
END;

PROCEDURE NINTHFEED;
BEGIN
write(chr(27),'0')
END;

PROCEDURE SIXTHFEED;
BEGIN
write(chr(27),'2')
END;

PROCEDURE COLUMNS(x:1..132);
BEGIN
write(chr(27),'Q',chr(x))
END;

PROCEDURE SUPERSCRIPT(flag:0..1);
BEGIN
IF flag=1 THEN write(chr(27),'S',chr(0))
ELSE write(chr(27),'T')
END;

PROCEDURE SUBSCRIPT(flag:0..1);
BEGIN
IF flag=1 THEN write(chr(27),'S',chr(1))
ELSE write(chr(27),'T')
END;

PROCEDURE GRAPHIC(flag:0..1;n:integer);
BEGIN
IF flag=1 THEN write(chr(27),'K') ELSE write(chr(27),'L');


write(chr(n MOD 256),chr(n DIV 256))
END;
```

Driver program continued on page 12

BETTER BASIC Save Time Typing Those Listings.

It has suddenly occurred to me that nowhere in the manual is there a mention of how many, or what abbreviations can be used when typing in Basic programs. Well; here they are .....

ABS	AB.	ADJSPR	AD.	ANGLE	ANG.
ARC	AR.	ASSEM	A.	ATTR	AT.
AUTO	AU.	BAUD	B.	CHR$	CH.
CIRCLE	CI.	CLEAR	CLE.	CLOCK	CLO.
CLS	C.	COLOUR	COL.	CONT	CO.
CRVS	CR.	CSR	CS.	CTLSPR	CT.
DATA	D.	DIM	DI.	DRAW	DR.
DSI	DS.	EDIT	E.	EDITOR	EDITO.
ELSE	EL.	EXP	EX.	FOR	F.
GENPAT	GE.	GOSUB	GOS.	GOTO	G.
GR$	G.	INK	I.	INKEY$	INKE.
INPUT	INP.	LEFT$	LEF.	LET	LE.
LINE	LIN.	LIST	L.	LLIST	LL.
LOAD	LO.	LPRINT	LP.	MID$	MI.
MOD	MO.	MVSPR	MV.	NEXT	N.
NODDY	NODD.	NODE	NOD.	ON	O.
OUT	OU.	PANEL	PAN.	PAPER	PA.
PAUSE	PAU.	PHI	PH.	PLOD	PL.
POKE	PO.	PRINT	P.	RAND	RA.
READ	REA.	REM	R.	RESTORE	RES.
RETURN	RET.	RIGHT$	RIG.	RND	RN.
RUN	RU.	SAVE	SA.	SBUF	SB.
SGN	SG.	SIN	SI	SOUND	SO.
SPK$	SPK.	SPRITE	S.	SQR	SQ.
STEP	SE.	STOP	STO.	STR$	STR.
TAN	TA.	THEN	T.	TIME$	TI.
USER	U.	VAL	VA.	VERIFY	VE.
VIEW	VI.	VS	V.		

****

SPECIAL OFFER TO GENPAT MEMBERS ONLY

DMX 80 PRINTER
CABLE
BRUNWORD

Excellent print quality interfaced with an excellent tape based word processor.

.....£239.00 INCLUDING POST & PACKING

SENTIENT DISTRIBUTION

Bring You

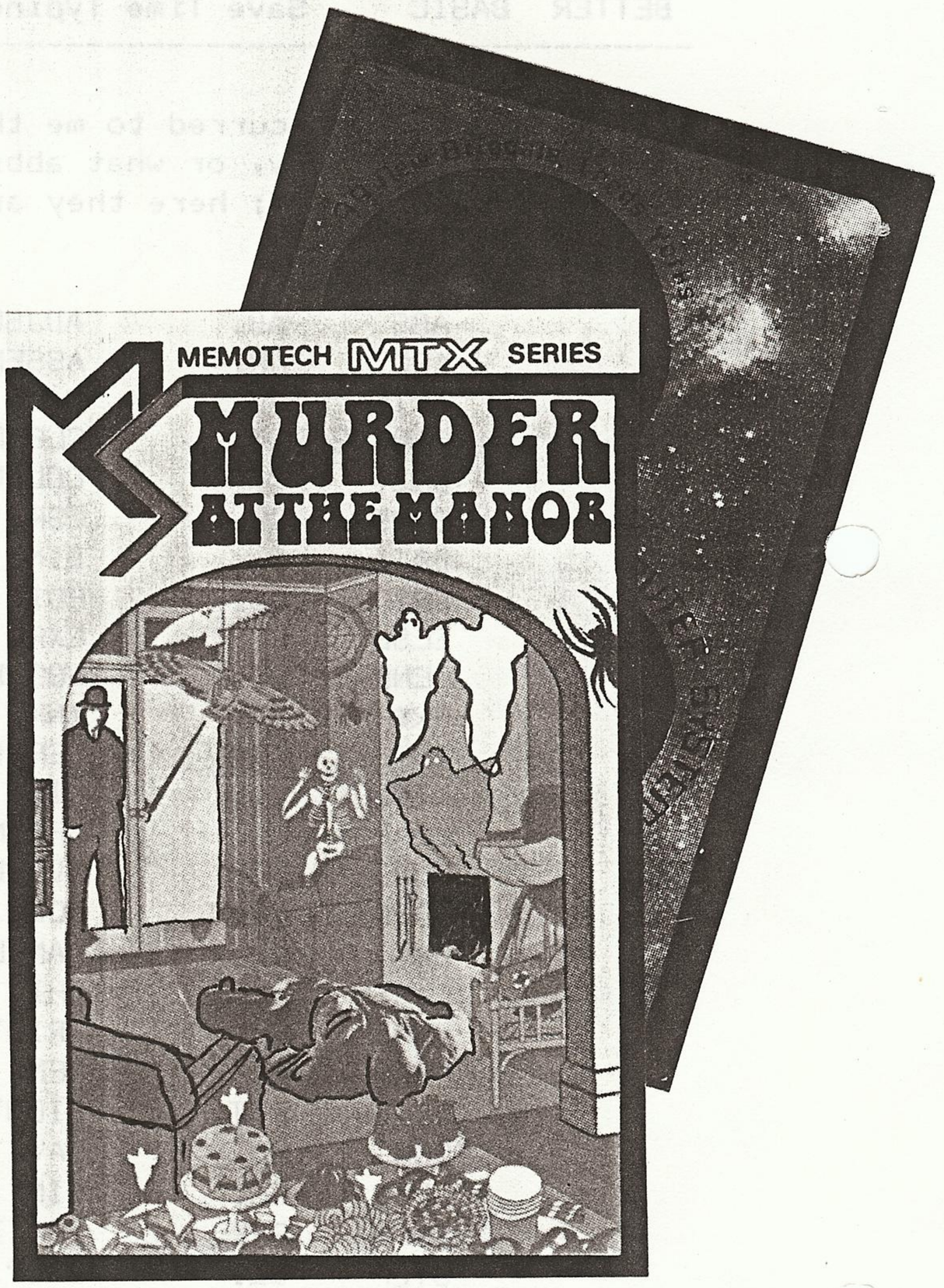
4 BRAND NEW TITLES THAT YOU CAN'T AFFORD TO MISS!

Heli-Maths £5.95 *Club £4.95*

Interactive maths program for children upto 11 years old. Simple control, exciting graphics and sound ensure that attention never wains!

Spelli-Copter £5.95 *Club £4.95*

Again suitable for children upto the age of 11 years, Spelli-Copter contains over 800 words to test and teach your child. Makes spelling fun!



Murder At The Manor £6.95 *Club £5.95*

A traditional adventure in the classic 'Who-dunnit' mould. Can you arrest the murderer before he gets you. The killer is randomly chosen before each game.

The Key To Time £6.95 *Club £5.95*

Be your favorite Time Traveller in his quest through time to discover the five fragments of The Key To Time before it is too late!

To: Genpat.
3 Bulcock Street,
Burnley BB10 1UH

£1 Discount for Club Members

Cheques made payable to:
GENPAT.

Please Supply:

	Quantity		Quantity
Heli-Maths	☐	Spelli-Copter	☐
Murder At The Manor	☐	Key To Time	☐

Cheque Enclosed For £.....

ASSEMBLY LANGUAGE PART 3

In last month's magazine, we examined the different registers of the Z80A CPU. Before we look at the various ways in which we address these registers, let's take a closer look at the **F** or **Flag register.**

A common CPU operation is the **compare** instruction - **CP** in Z80 mnenomics. This works in a similar way to the Basic statement:-

```
10 IF A = 10 OR A > 10 OR A < 10 THEN GOTO 100
```

It allows you to make decisions then act accordingly by branching out of one routine, or jumping into another part of your program. The result of a **compare** is checked by testing the state of the **bits** in the **F** register.

```
BIT  7   6   5   4   3   2   1   0
     S   Z   -   H   -  P/V  N   C
C  =  CARRY FLAG : N = ADD/SUBTRACT FLAG [ BCD OPERATIONS ]
H  =  HALF CARRY FLAG [ BCD OPERATIONS ] : P/V = PARITY OVERFLOW
          Z  =  ZERO FLAG  : S  =  SIGN BIT
```

Bits **3 & 5** are not used. The **Half Carry and N flags** are used for **Binary Coded Decimal** operations, and we are not concerned with them at this point.

The **Carry Flag,** if set, denotes a **CARRY (C),** and if reset denotes a **NO CARRY (NC)** condition. This flag is directly affected by an addition or subtraction. It should be understood that all **CP** operations compare the value contained in the **A register** with the next operand, which can be a **value in a register** or an **absolute value:**

```
CP L    ; compare value in A with value in C
CP £32  ; compare value in A with 32 hexidecimal
```

What is actually happening during a compare operation is the value of the compare operand is subtracted from the value contained in the A register.

```
LD A,£40    ; Load A reg with 40 Hex
CP L        ; VALUE OF A - VALUE OF L
```

You can see, from the above,that a **compare** operations is essentially an arithmetic operation on the A register, and, as such, the result will affect the **Carry Flag.**

The **Zero flag** is set [1] whenever the result of an arithmetic operation results in **zero.** If the **Carry & Zero flags** are used in tandem, any

possibility can be tested. Consider the following Basic statement:-

```
10 LET A = VALUE
20 IF A = 10 THEN GOTO 40
30 IF A> 10 THEN GOTO 50
40 GOTO 40
```

Translating this into assembler:

```
        LD A,VALUE   ; put value in A reg.
        CP 10        ; compare value in A to 10.
        JR Z, EQUAL ; if value in A = 10 then goto Equal.
        JR NC,GREATER ; if carry flag  not set goto Greater.
LOOP:JR LOOP ; value in A is not equal to, and is less than 10
```

The **No Carry** situation will arise if A = 10 or A > 10 and so it is always wise to compare the A register with a value **1 greater than the value you wish to test for.**

```
CP 10
JR NC,NEXT
```

If the carry is not set then A is definitely greater than 9 but could be equal to 10.

This is the reason we tested for **Zero** before testing the carry flag in the previous example.

The four situations can be summarised as follows:-

N Value in A reg > or = to compared value
C Value in A reg < compared value
Z Value in A reg = compared value
NZ Value in A reg not equal compared value

Also note that the value in the A register is not affected, and is left unchanged by the compare.... the subtraction takes place in theory only.

The Sign Bit: If you can remember issue one, we discussed the 2's complement of a number. In 2's complement notation if the 7th bit is a **1** then the number is **negative,** and if bit 7 is = 0 then the number is **positive.** The **sign bit** reflects the state of this seventh bit. The other flags will be discussed as the situation arises, but the three already discussed are the most important.

ADDRESSING MODES

Any detailed review of a CPU will always mention its addressing modes. This is where the Z80 comes into its own: the wide variety of addressing modes available on this CPU makes life really easy for the programmer. Addressing modes will create no serious problem to you. You will soon become familiar with the most useful, and to help you, here are the most common ways of addressing the Z80.

Immediate Addressing:

In Basic a similar instruction would be : **LET A = 3**
LD A,03 or LD HL,5007 (known as immediate extended addressing)
You are loading a register or a register pair with immediate data.

Register Addressing:

This exactly what it says: One register is loaded from another.

```
LD A,C     ; Load A from C
```

Indirect Register Addressing:

In this mode of addressing, the location of the operand is held in one of the register pairs; **BC,DE or HL.** A translation in Basic would be:

```
10  LET BC = 14390
20  LET A = PEEK(BC)
```

In assembler:

```
LD  A,(BC)     ; load A register with the value in  the  RAM/ROM  location
               ; pointed to by the BC register pair.
::::::::::
LD HL,14390    ; make HL point to address 14390
LD A,(HL)      ; put value in A register.
LD DE,56789    ; point DE registers to location 56789
LD  (DE),A     ; load memory location pointed to by  DE  registers  with
               ; value in A register.
```

Indexed Addressing:

This is really powerful addressing mode. It allows you to retrieve, or store data from tables set up in memory. We can make **IX or IY** registers point to an address the add an **offset** within the range of **-128 to +127.** If the IX register points to memory address 3C00 Hex we can **LD A,(IX+15)** which would load the A reister with the contents of memory location 3C0F **hex.** LD A,(IX+00) would load the A register from memory location 3C00 **hex.**

Implied Addressing:

This mode means that the register is not named in the mnemonic, but is implied by it.

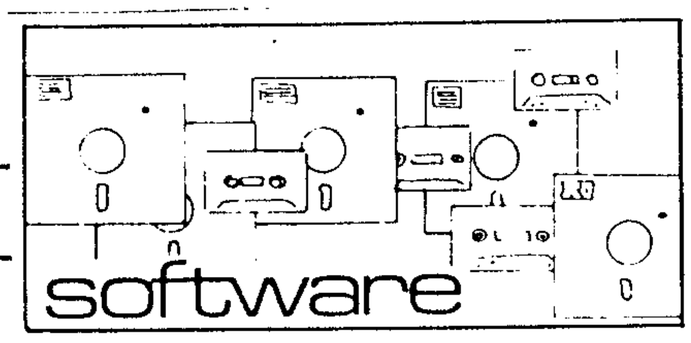
```
ADD E   ;The contents of the E register are added to the contents of  the
        ;A register.
SCF     ; Set Carry Flag.
```

Protocol:

LD A,(HL) ..... LD A **FROM** LOCATION POINTED TO BY HL.
Data flow.... A <====== HL <========= Memory Location.
Data always flows from RIGHT to LEFT LD A,36

REVIEWS

TURBO By Andrew Key & Published by Continental



This game is supposed to be based on the ever popular **Poll Position.** In the original, you race your car around a circuit trying to avoid other cars coming at you from the opposite direction, and from behind. After completing the circuit you then progress into the city,and then back onto the racing circuit.

Turbo follow this formula. But, my goodness,moving into the city from the track is just a disasterous **mess!** The graphics are lousy, and my overall impression is that the game has been rushed, without care for the end purchaser.

The speed of the game is fast, and the idea is good, but Andrew Key is capable of far better things than this program. I hate to be unkind, but I certainly cannot find any reason to justify paying £6.95p for this games program.

I am afraid it is **thumbs down on this game & a black mark to Continental for allowing this through the net** especially now that most programmers know the machine well enough to produce games in the ilk of Pothole Pete.

FIRST LETTERS Continental Software

The program is generally well conceived, and the material is aligned for the stated age group - 4 to 7 years.

Tutor selection well organised although placing the cursor over a blank area will subsequently cause the program to hang up. Some symbols were considered to be badly formed, i.e **Bell,Face - looks more like a mask- Owl - looks more like a bear !**

There is a **spelling mistake** under the **Letter Selection** header,and should not have appeared in an educational program. **ED... we all make mistakes !**

The program was considered to be suitable for education of the young members of the family, but with certain reservations as described above. Also, it was thought that a **pause** facility would permit the child/children sufficient time to copy the correct results from the screen.

With slight refinement this could be a useful teaching\learning program.
B.Cooke 13th November 1984

PROGRAM

SOUND ROUTINE

The following, excellent, routine was sent in by one of our members. Unfortunately, he did not put is name on the listing. If he can contact me, we will give him a credit in the December edition. I urge you to try this routine, I have had many a try a producing some really fascinating effects.

The program can be used if you are writing a game , and are looking for some good, and unusual sound effects. When you find a sound type any key and the poke will be sent to the printer - those of you that haven't got a printer change line 50 to read **Print.** Lines **5, 10,& 15** can also be altered and once you undestand the program, try altering these lines to give you customised effects.

PROGRAM

```
 5 POKE 64084,0 :POKE 64085,0
10 FOR A = 1 TO 255
15 PRINT "POKE 64086 -------";A
20 POKE 64086,A
30 INPUT B$
40 IF B$ ="" THEN GOTO 60 ELSE GOTO 50
50 LPRINT "64086,";A
60 NEXT
```

SOUND TABLES:

SOUND POKES	TYPE	CHn	No.
64077-64082	SP	0	086
64082-64084	JP	0	086
64084-64086	JP	0	
64086-64090	HP	0	
64090-64091	TN	0	
64091-64092	SP	0/1	100
64092-64093	SP	0/1	086
64094-64096	JP	1	
64096-64100	HP	1	
64100-64101	HP	1	
64101-64104	JP	1/2	
64104-64106	JP	2	102
64106-64110	HP	2	
64110-64111	TN	2	
64111-64114	BN	2/3	112
64114-64116	BN	3	
64116-64120	TN	3	
64120-64121	TN	3	

SP=Special :TN=Tone :JP=Jump
BN=Bounce :HP=High Pitched

HOW TO USE THE SOUND TABLE:

The first column shows which **pokes** must be typed in from **Nnn** to **Nnnn.** The second column is the type of sound and the third is which sound channel is used. After entering the first pokes, a new range of sounds

can be created by poking the number in the 4th column.

eg: 64082 - 64084 JP ! 0 ! 086

Try this program using the above range from the sound tables .....

You can use them in the direct command state if you want (you can just type them in as written..... you don't need to make a program to do it).

```
POKE 64082,n :POKE 64083,n:POKE 64084,n : POKE 64086,n
                                                    :.:...
Where n = any value in the range 0 - 255   &        :..Value from
                                                       Column 4
```

eg 2:

64104 - 64106 !JP!2!102

```
POKE 64104,n:POKE 64105,n:POKE 64106,n:POKE 64102,n
```

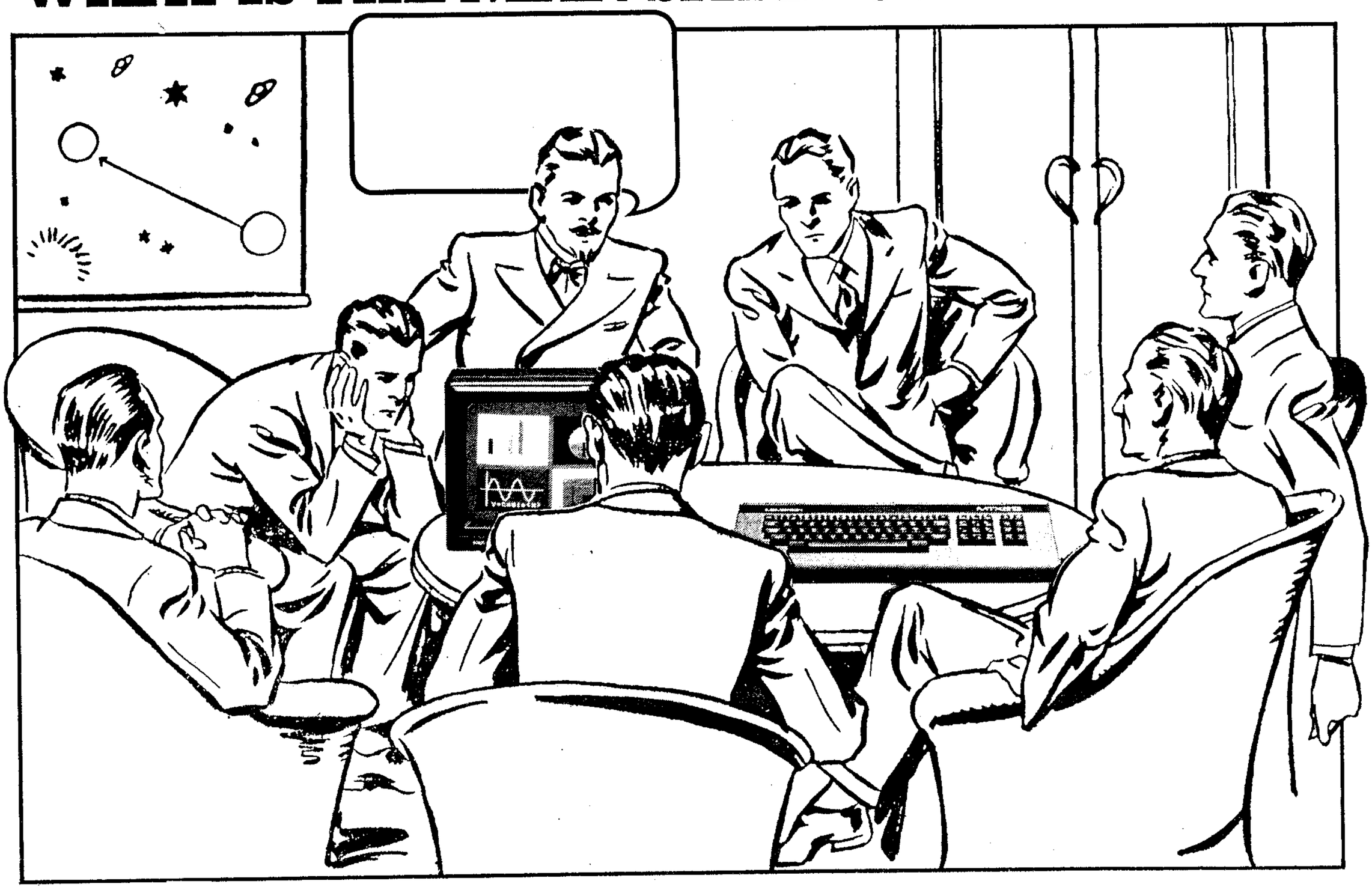
You can create many thousands of different sound effects by experimenting. However,**be very careful when using high pitched POKES as this can sometimes crash the system.** Using Pokes to mix II channels can create superb effects. **If a crash does occur: RESET THE MACHINE AND POKE 64167,1 AND YOU CAN THEN LIST THE PROGRAM TO A PRINTER.**

Competition

SEND YOUR ENTRIES BEFORE 12th DECEMBER

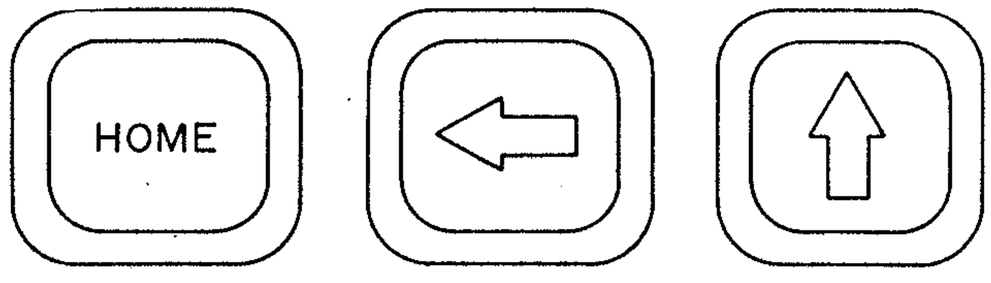
2 PIECES OF SOFTWARE FOR THE 3 MOST AMUSING ENTRIES.....

WHAT IS THE MAN SAYING?



Starter pack

BASIC FOR BEGINNERS CONTROL CODES



SCREEN FORMATTING

The novice Basic programmer tends to find screen formatting a tedious task. Even hardened hackers still go all around the woods to get to the trees. Consider the following section from a Basic program - seen all to often in programs submitted for publication.

```
10 CSR 3,5: PRINT "INPUT A NUMBER BETWEEN 1 - 10 ";:INPUT X
20 IF X > 10 OR X <1 THEN  CSR 3,5:PRINT "                    "
:GOTO 10
```

The same routine can be re-written and will run faster by re-coding in the following way:

```
10 CSR 3,5: INPUT "INPUT A NUMBER BETWEEN 1-10";X
20 IF X >10 OR X<1 THEN CSR 3,5:PRINT CHR$(5);:GOTO 10
```

CHR$(5) IS KNOWN AS A CONTROL CODE. It is used by the Basic interpreter to carry out a pre-defined function which has been programmed into the ROM.

You can take advantage of **all** the control codes, and with a little thought, make your program run faster and more efficiently.

CHR$(4): Using this code will set your background colour.

```
10 PRINT CHR$(4);"8"        ... Will set your backdrop to colour red.
```

CHR$(6): This control code will set the colour of your foreground.

10 PRINT CHR$(6);"3" will set foreground colour to green.

CHR$(7): If you use this in a Basic line you will sound the **BELL.**

```
10 IF X = 40 THEN PRINT CHR$(7)
```

CHR$(30) will turn the cursor **off** & **CHR$(31)** will turn it back on again.

You can simulate the action of **all** the cursor keys that appear on the left-hand keypad by using the following control characters in the same format as above ..... just by putting them in a print statement: eg PRINT CHR$(n).

CHR$(7) ..... BACKSPACE CURSOR ONE PLACE <==
CHR$(25) ..... MOVE CURSOR FORWARD ONE PLACE ==>
CHR$(11) ..... MOVE CURSOR UP ONE LINE
CHR$(10) ..... MOVE CURSOR DOWN ONE LINE ....LINE FEED
CHR$(12) ..... CLS & MOVE CURSOR TO HOME POSITION.
CHR$(13) ..... CARRIAGE RETURN..... MOVE CURSOR TO LEFT OF NEXT LINE.
CHR$(9) ..... TAB CURSOR TO START OF NEXT 8 COLUMNS.

Along with the control codes we also have the **Escape Sequences.** A lot of members have telephoned to ask how they can get a £ sign printed to the screen instead of the **'hash' sign.** This is easy if you know how !

PRINT CHR$(27);"B1";

You can even simulate any of the above **control sequences** by using the following: **PRINT CHR$(27);CHR$(88); "N"** where **N** = the control code number that appears in the brackets in the above list.

The Comma ',' tells the computer to print at the next print zone which normally is 8 spaces from the start of the current zone.

The semi-colon ';' suppresses a **Line Feed** after a **Print** statement. String$ will print hard up against each other, and **variables** will print with a space between them to allow for the **minus sign** - if there should be one. Eg:

LET X$ = "KEITH": LET Y$ = "HOOK" : PRINT X$;Y$... This would print on the screen as.. **KEITHHOOK.**
LET X = 10 :LET Y = 15: PRINT X;Y...would print to the screen as ..... **10 15** and if Y = -15 the above would print to the screen as: **10-15.**

```
10 PRINT CHR$(12);CHR$(31);: PRINT CHR$(4);"1": PAUSE 500
20 PRINT CHR$(6);"2";: PRINT CHR$(12);CHR$(30);
30 LET X=38
40 LET Y=23
50 FOR I=1 TO X
60 PRINT CHR$(25);
70 PAUSE 50: NEXT I
80 FOR I=1 TO Y
90 PRINT CHR$(10);
100 PAUSE 20: NEXT I
110 LET Y=Y-1
120 LET X=X-1
130 FOR I=1 TO X
140 PRINT CHR$(8);
150 NEXT I
160 FOR I=1 TO Y
170 PRINT CHR$(11);
180 NEXT I
190 LET X=X-1
200 LET Y=Y-1
210 GOTO 50
1 REM This program demonstrates the use of CONTROL CHRACTERS to move the csr around the screen.
```

MTX SCREEN LOAD & SAVE
JOHN MULLINS

One of the biggest criticisms of the MTX micros is the inadequate cassette facilities. Only saving, loading and verifying of Basic programs and their variables is catered for, there is no provision for saving and loading blocks of memory or your patiently designed graphics screens.

Saving and loading blocks of memory provide no real problems (See listing 1). On most micros, having this facility would enable us to save a graphics screen, however, with the MTX it is not quite so easy. This is because screen memory is not directly accessed by the CPU, but rather indirectly by reading and writing to and from CPU ports 1 and 2. It is possible to copy screen memory into the Z80's memory and then save the appropriate block to tape, however this requires that we have 16K of spare memory, and it is not very spectacular in that we have a blank screen whilst 16K of data loads and then suddenly our screen appears when we copy the data into video RAM (VRAM). A much more acceptable solution would be to take one byte at a time from VRAM and save it to tape, the reverse process being used when loading so that the screen builds up as the data is being fetched from tape.

In order to perform the above task we need to know a little about the graphics processor. To read a byte from VRAM we must first set up the VRAM address of that byte using port 2 and then read the byte in on port 1, the VRAM address is then incremented automatically, and subsequent accesses will read contiguous bytes from screen memory (The same is true for writing to to VRAM. See PCN Issue for a full explanation of the graphics processor). To save all 16K of VRAM, all that is required is to copy the save routine from the MTX's ROM and adjust the part that reads from the Z80's memory so as to read from VRAM, similar for loading.

Saving the whole of VRAM means that any sprite patterns and positions have also been saved, and, when loaded back should appear. However there is a problem, if, for instance, we were using 16*16 sprites with double magnification at the time of saving, they may not have these characteristics when loaded back. This is because the size and magnification of sprites is held in one of the graphics chips "write only" registers i.e. we cannot read it!! Fortunately the MTX stores this value at location #FF55, thus we can save this before the screen data, and when loaded back write the correct value to the graphics chip. The value at #FF55 should always be in the range 0-3 and corresponds to the values in the CTLSPR 6,n command, thus if we require some kind of error checking we can test this value, and if not in the correct range signal an error.

Listing 2 provides the code for saving, and listing 3 the code for loading. Both routines are almost identical to the MTX ROM routines at #0AAE, except that data is fetched from VRAM.

It would be very nice if we could tag these routines onto Basic, and access them via a keyword as opposed to having to GOSUB the code lines or call them via USR. Again this is not the huge problem it appears to be, since the authors of the MTX ROM, in their infinite wisdom, actually left a keyword for the user to define. No prizes for guessing that this keyword is USER!!

In order to activate USER we must do two things, firstly define it's syntax (i.e. how many parameters, string or numeric etc.) and secondly define the actual routine itself. Without going into a long discussion on how the syntax checker works, just take it from me that the POKEs described in listing 4 set USER to require a simple numeric parameter, and anything else will produce a 'Mistake' error and the MTX will not accept the command. To get the parameter for USER we use the RST 30 instruction, a single byte call to the MTX ROM which returns the value in the accumulator. I have used USER 0 for save and USER 1 for load, and anything else will give an 'Out of range' error. Listing 4 shows the routine and how to place it high up in memory (in a sound buffer) so that it is saved alongside a Basic program and will be active once reloaded. The obvious drawback to this method is that you cannot use continuous sound, since this uses the sound buffer and will overwrite the code, however most people will not be using sound whilst creating graphics screens, so the problem is not that great.

It is really nice to welcome John to the fold. John worked on the conversion of MURDER AT THE MANOR & THE KEYS TO TIME. My first meet with John was at Memotech, and we have been great friends ever since. I have never met a more dedicated programmer. When he is not selling MTX's - which he does for a living - he is key bashing. Watch out for some really super programs from John & his muckers at Sentient Software.

LIST ONE

```
10 CODE

4007        LD HL,START       ;Start of
400A        LD DE,LENGTH      ;Length of
400D        LD A,DATA1
400F        LD (£FD67),A
4012        LD A,DATA2
4014        LD (£FD68),A
4017        CALL £AAE
401A        RET

      Symbols:


      20 REM DATA1=0 for SAVE and LOAD, 1 for VE
      30 REM DATA2=0 for SAVE, 1 for LOAD and VE
```

LISTING TWO

```
10 CODE

4007        XOR A      ;Signal SAVE
4008        LD (£FD68),A
400B        RST 10     ;Enter VS 7 with the message "Start tape etc."
400C        DB £6F,£9C,"Start tape, then press a key"
402A WAITKEY:CALL £79  ;Wait for a keypress
402D        JR Z,WAITKEY
402F        RST 10     ;Enter VS 4
4030        DB £44
4031        LD HL,£FF55        ;Save SMBYTE
4034        LD DE,1
4037        CALL £AAE
403A        LD DE,£4000        ;Signal 16K to be saved
403D        LD HL,0    ;Set up address zero in VRAM (Read mode)
4040        LD A,L
4041        OUT (2),A
4043        LD A,H
```

LISTING TWO CONTINUED

```
4044          OUT (2),A
4046          CALL £B10
4049          EX AF,AF'
404A          LD A,£90
404C          EX AF,AF'
404D          LD BC,£05DC       ;Send 1500 zero bits for "leader" tone
4050 OUTBLK1: XOR A
4051          CALL £0A82
4054          DEC BC
4055          LD A,B
4056          OR C
4057          JR NZ,OUTBLK1
4059 DEL:     JR NC,DEL  ;Now send a single set bit to indicate start of r
405B          OUT (3),A  ;DATA
405D DEL1:    JR C,DEL1
405F          CCF
4060          CALL £A92
4063 OUTBLK2: IN A,(1)   ;Get the byte from VRAM
4065          LD C,A
4066          CALL £AA4  ;Send it to tape
4069          DEC DE     ;Decrease counter and loop back until done
406A          LD A,D
406B          OR E
406C          JR NZ,OUTBLK2
406E          CALL £B06  ;Reset CTC to normal operation
4071          RET
```

Symbols:

```
OUTBLK1 4050    DEL     4059
DEL1    405D    OUTBLK2 4063
WAITKEY 402A
```

LISTING THREE

10 CODE

```
4007          LD A,1     ;Indicate LOAD
4009          LD (£FD68),A
400C          XOR A      ;Signal "not verify"
400D          LD (£FD67),A
4010 LOAD:    RST 10     ;Enter VS 4
4011          DB £44
4012          LD HL,£FFFF       ;Store size magnification byte temporarily
4015          LD DE,1
4018          CALL £AAE
401B          LD A,(£FFFF)      ;Is it in correct range
401E          CP 4
4020          JR C,OKCHAR       ;Proceed if so
4022          RST 28     ;Signal "Mismatch error"
4023          DB £0B
```

LISTING THREE CONTINUED

```
4024 OKCHAR: LD (£FF55),A      ;Store size-magnification byte at SMBYTE
4027         OR £E0     ;Send it to VDP
4029         OUT (2),A
402B         LD A,£81
402D         OUT (2),A
402F         LD HL,£4000       ;Set up address zero in VRAM (Write mode)
4032         LD DE,£4000       ;Sigan1 16K of data to be loaded
4035         LD A,L
4036         OUT (2),A
4038         LD A,H
4039         OUT (2),A
403B         CALL £B10  ;Set up CTC for loading
403E         EX AF,AF'  ;Save the flags
403F         LD A,£90   ;Used for sound
4041         EX AF,AF'
4042 INBLCK: LD B,0     ;Search for 256 zero bits
4044 INBLK1: CALL £A6B
4047         JR C,INBLCK
4049         DJNZ INBLK1
404B STBIT:  EI ;Now wait for a set bit to indicate data ready
404C         XOR A
404D         CCF
404E         CALL £A72
4051         JR NC,STBIT
4053 INBLK2: CALL £A9A  ;Get byte from tape
4056         LD A,C     ;Store it in A
4057         OUT (1),A  ;Send it to VRAM
4059         IN A,(2)
405B         DEC DE     ;Decrement counter and loop back until done
405C         LD A,D
405D         OR E
405E         JR NZ,INBLK2
4060         CALL £B06  ;Reset CTC to normal operation
4063         RET
```

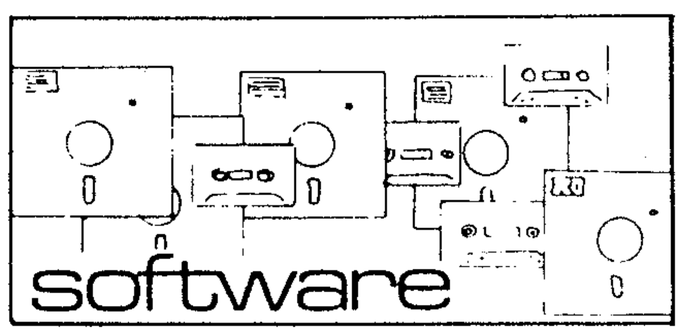
LISTING FOUR

10 CODE

```
4007          RST 30     ;Get the parameter
4008          LD (£FD68),A       ;Store it at type
400B          XOR A      ;Signal "not verify"
400C          LD (£FD67),A
400F          LD A,(£FD68)       ;Get type
4012          OR A       ;Is it zero?
4013          PUSH DE    ;Save this (Essential for BASIC)
4014          JR Z,OUTBLCK       ;Jump forward if SAVE
4016          DEC A      ;Is it a 1?
4017          JR Z,LOAD  ;Jump forward for load
4019          POP DE     ;Tidy up the stack
401A          RST 28     ;Give "Out of range" error
401B          DB £22
```

```
401C LOAD:   RST 10     ;Enter VS 4
401D         DB £44
401E         LD HL,£FFFF        ;Store size magnification byte temporarily
4021         LD DE,1
4024         CALL £AAE
4027         LD A,(£FFFF)       ;Is it in correct range
402A         CP 4
402C         JR C,OKCHAR        ;Proceed if so
402E         RST 28     ;Signal "Mismatch error"
402F         DB £0B
4030 OKCHAR: LD (£FF55),A       ;Store size-magnification byte at SMBYTE
4033         OR £E0     ;Send it to VDP
4035         OUT (2),A
4037         LD A,£81
4039         OUT (2),A
403B         LD HL,£4000        ;Set up address zero in VRAM (Write mode)
403E         LD DE,£4000        ;Signal 16K of data to be saved
4041         LD A,L
4042         OUT (2),A
4044         LD A,H
4045         OUT (2),A
4047         CALL £B10  ;Set up CTC for loading
404A         EX AF,AF'  ;Save the flags
404B         LD A,£90   ;Used for sound
404D         EX AF,AF'
404E INBLCK: LD B,0     ;Search for 256 zero bits
4050 INBLK1: CALL £A6B
4053         JR C,INBLCK
4055         DJNZ INBLK1
4057 STBIT:  EI ;Now wait for a set bit to indicate data ready
4058         XOR A
4059         CCF
405A         CALL £A72
405D         JR NC,STBIT
405F INBLK2: CALL £A9A  ;Get byte from tape
4062         LD A,C     ;Store it in A
4063         OUT (1),A  ;Send it to VRAM
4065         IN A,(2)
4067         DEC DE     ;Decrement counter and loop back until done
4068         LD A,D
4069         OR E
406A         JR NZ,INBLK2
406C         JR OUT     ;Exit the routine and resume execution of BASIC
406E OUTBLCK:RST 10     ;Enter VS 7 with the message "Start tape etc."
406F         DB £6F,£9C,"Start tape, then press a key"
408D WAITKEY:CALL £79   ;Wait for a keypress
4090         JR Z,WAITKEY
4092         RST 10     ;Enter VS 4
4093         DB £44
4094         LD HL,£FF55        ;Save SMBYTE
4097         LD DE,1
409A         CALL £AAE
409D         LD DE,£4000        ;Signal 16K to be saved
40A0         LD HL,0    ;Set up address zero in VRAM (Read mode)
40A3         LD A,L
40A4         OUT (2),A
40A6         LD A,H
40A7         OUT (2),A
40A9         CALL £B10
40AC         EX AF,AF'
40AD         LD A,£90
40AF         EX AF,AF'
40B0         LD BC,£05DC        ;Send 1500 zero bits for "leader" tone
40B3 OUTBLK1:XOR A
40B4         CALL £0AB2
40B7         DEC BC
40B8         LD A,B
40B9         OR C
40BA         JR NZ,OUTBLK1
40BC DEL:    JR NC,DEL  ;Now send a single set bit
40BE         OUT (3),A  to indicate start of data
40C0 DEL1:   JR C,DEL1
40C2         CCF
40C3         CALL £A92
40C6 OUTBLK2:IN A,(1)   ;Get the byte from VRAM
40C8         LD C,A
40C9         CALL £AA4  ;Send it to tape
40CC         DEC DE     ;Decrease counter and loop back until done
40CD         LD A,D
40CE         OR E
40CF         JR NZ,OUTBLK2
40D1 OUT:    CALL £B06  ;Reset CTC to normal operation
40D4         POP DE     ;Retreive DE and return to execution of BASIC
40D5         RET
```

```
10 REM Type in the above, then as a
20 REM Type in the above, then as a
30 REM direct command enter SBUF 7.
40 REM Then enter PANEL type "M" when
50 REM you will be given the prompt
60 REM MOVE now type 4007 (B007 on an
70 REM MTX 500), press <RET> the prompt
80 REM END appears type 40D5 (B0CF)
90 REM then <RET>,the prompt TO appears
100 REM type F803 then <RET>
110 REM Now as a direct command enter
120 REM POKE 64135,201
130 REM POKE 64136,2
140 REM POKE 64137,195
150 REM POKE 64138,3
160 REM POKE 64139,248
170 REM
180 REM To save a screen type USER 0,
190 REM to load type USER 1, other
200 REM values are reserved for future
210 REM use.
```



REVIEW - The Zoo by Mike Bray

This is an adventure game of the text variety (similar to Alice in Wonderland) from Continental Software. The storyline is that some animals have escaped from a local zoo, and you, as Sidney Noble, (a well known 'cunning individual') are requested to recapture them. You can travel through the town collecting items to help you with the capture of the animals, and occasionally you run into trouble with the local townsfolk.

The game starts with two pages of instructions and an explanation of the routine to save the game so you can restart from where you left off. You are then requested to input the time - I normally set this to zero so elapsed time was shown - after which the game starts.

The game screen has five areas for information. At the top is your present location. On the right, two areas give the move direction options, and the items to be collected. The left screen shows the last move with an input prompt, and at the bottom the clock ticks away. A number of the direction words may be abbreviated to the first letter in the word i.e. N for North, I for Inside. This is a good feature as it speeds up play.

One of the drawbacks of this type of game for me is the lack of a word list. I found it very frustrating when my attempts were met with "PARDON !" and after a number of attempts to catch a train I gave up as all the word combinations I tried met with the same lack of success.

Apart from this drawback ,which may only be my problem, (I hit the same snag in "Alice in Wonderland"), I found the game absorbing and worthwhile. There are many levels that I failed to reach (I had a peek via PANEL) but I did catch an elephant !. Recommended for those of you who like text adventure games with a difference, and will someone please tell me how to catch that train ?.

HERE IS THE ANSWER FOR ALL WHO WOULD REALLY LIKE TO LEARN MACHINE CODE....

The Club has negotiated a deal with Robert Broome, the principal of the COMPUTER TRAINING COLLEGE which is located in Manchester, to set up a tutorial course on Z.80 machine code.

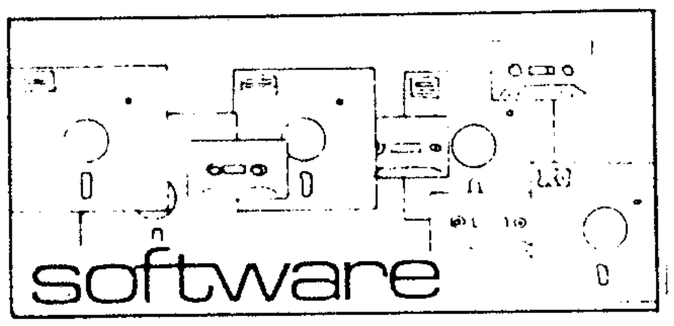
The College has been running a Flexi Course for some time now, and they have re-written parts of their work to cover the MTX. The course revolves around an excellent tutorial book written by Michael Moore F.I.C.O.

The Computer Training college will also support their book with a back-up service. That is to say, any member who purchases the book and then finds that they do not fully understand certain aspects of its contents can write to the college, and they will then do their best to expand on the area that is causing the difficulty.

This is an excellent package, and is offered to the club at a special discount rate of £7.95p.

You can apply by sending a cheque to **Mr.R. Broome F.I.C.O.,M.R.A.M.C.A, THE COMPUTER TRAINING COLLEGE, Norvic House, 1-7,Hilton Street, Manchester M4 1LP: Telephone: 061-835-1315.**

KERIAN SOFTWARE RELEASE 4 PIECES OF SOFTWARE FOR MTX



THE KEYS TO TIME : This is a really superb adventure, and if you look within the pages of any computer magazine from recent months, you will see that this was given a real good review when produced on the Spectrum.

MURDER AT THE MANOR: Also given favourable reviews. In this adventure you play the detective.... if you find the suspect before he finds you,all well and good, if you don't..... Random suspect select on each game. **Both these packages will be reviewed in the December edition.**
Now in stock at the club for a incredible £4.95 inc p.p each.

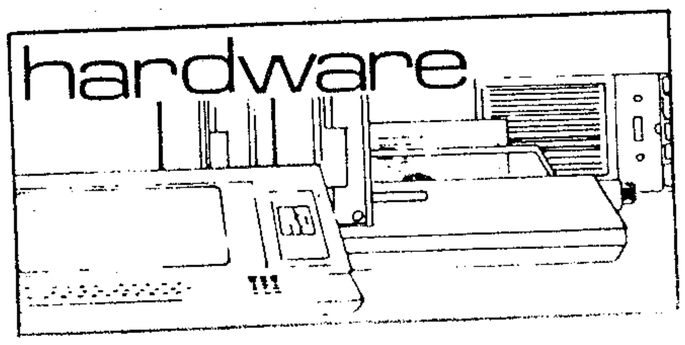
HELLI-MATHS & SPELLI-COPTER. Two programs for the younger members of the family. Rewards for spelling and getting your sums right.... if you don't then see what happens to the helli-copter ! These two programs have been sent to a school teacher, Alan Sturgess, and we will be publishing his findings in the next issue. For those of you who can't wait, they are now in stock at the club for £4.95 each inclusive.

IT'S **GOOD NEWS WEEK** - MEMOTECH TO RELEASE LOW PRICED DISC DRIVES.

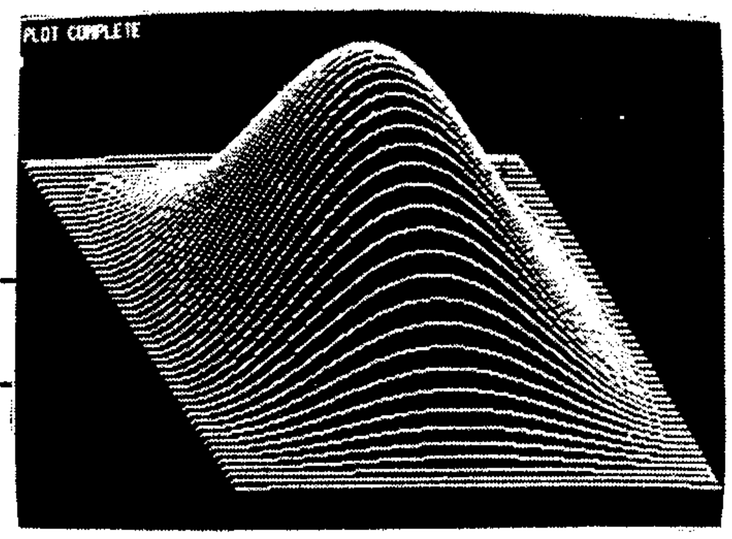
Memotech are to release three disc drives. If you sent your guarantee card into the Company then you may have already received notification. If you want to make further investigations, write to Memotech asking for details.

The first drive is a **100k capacity drive** which will cost you £199.00 plus £36.00 for the RS232 interface, making a total of £135.00.

The second drive will cost you **£149.00**, and as a member of **Genpat** you will get the RS232 **free !!**. This is a 250K drive and will take 28 days delivery, but you can order now on first come, first served basis. Your cheques will not be cashed until you have taken delivery, but please send **CASH WITH ORDER TO MEMOTECH MARKED GENPAT DISC DRIVE OFFER**



PROGRAMMING



PROGRAMMING THE VDP Part Two

Final words of Graphic Mode II

Last month we discussed setting up the **VDP to Mode II.** Before we leave this subject I would like to give you two other ways of setting up **Mode II** screens.

Graphic Mode II as a Bit Mapped Display

The crafty programmer can, with a bit of thought, use Mode II as a **Bit Mapped Screen.....** This is how the MTX sets up **VS 4.**

Using Mode II in this manner allows you to address every pixel on the screen individually, which is useful for **plotting points & drawing lines,** etc. Unfortunately, there is a draw-back when using this method: **although you can address each pixel on the screen, the colour bits cannot be formatted in the same manner.** We can, however, get around this by using either two colours, one colour for **on,** and another for the pixels that are turned off. The other method is to use more than two colours, but be very careful where we plot them on the actual screen.

To set up Mode II in this way we must write a different value to each of the **Pattern Name Table** [screen] entries, which would mean the Pattern Name Table would hold values from **0 - 767** instead of containing the actual **Pattern numbers** which is the normal method. What this means is: by writing a certain pattern to **8 bytes** in the **Generator Table** you have created a unique numbered pattern on the screen.

Explanation:

You will find this hard to grasp, but a little thought should clear the fog. Normally, we would set up patterns in the **generator table** and then place them on the screen by putting the **character code** in the **pattern name table.** This is no longer true as a bit mapped screen. We have already placed **character pattern numbers** into the screen positions - 0 to 767 - if we now fill the first 8 bytes in the **generator table** with the information that goes to make up the letter **A** it will take on the character value of **0.** Now, if we again write the same values to the last eight bytes in the **generator table** the letter **A** also takes on the character value of **255** in that position. We now have **two** letter **A's written to the screen, but instead of both taking on the character value of £41 hex - which is the normal ascii character value - they have unique ascii character numbers of 0 & 255.** If you now wanted to draw a line to underline the first A, it will not affect the A in position 767. In this way you can set or reset pixels in a unique and exclusive manner.

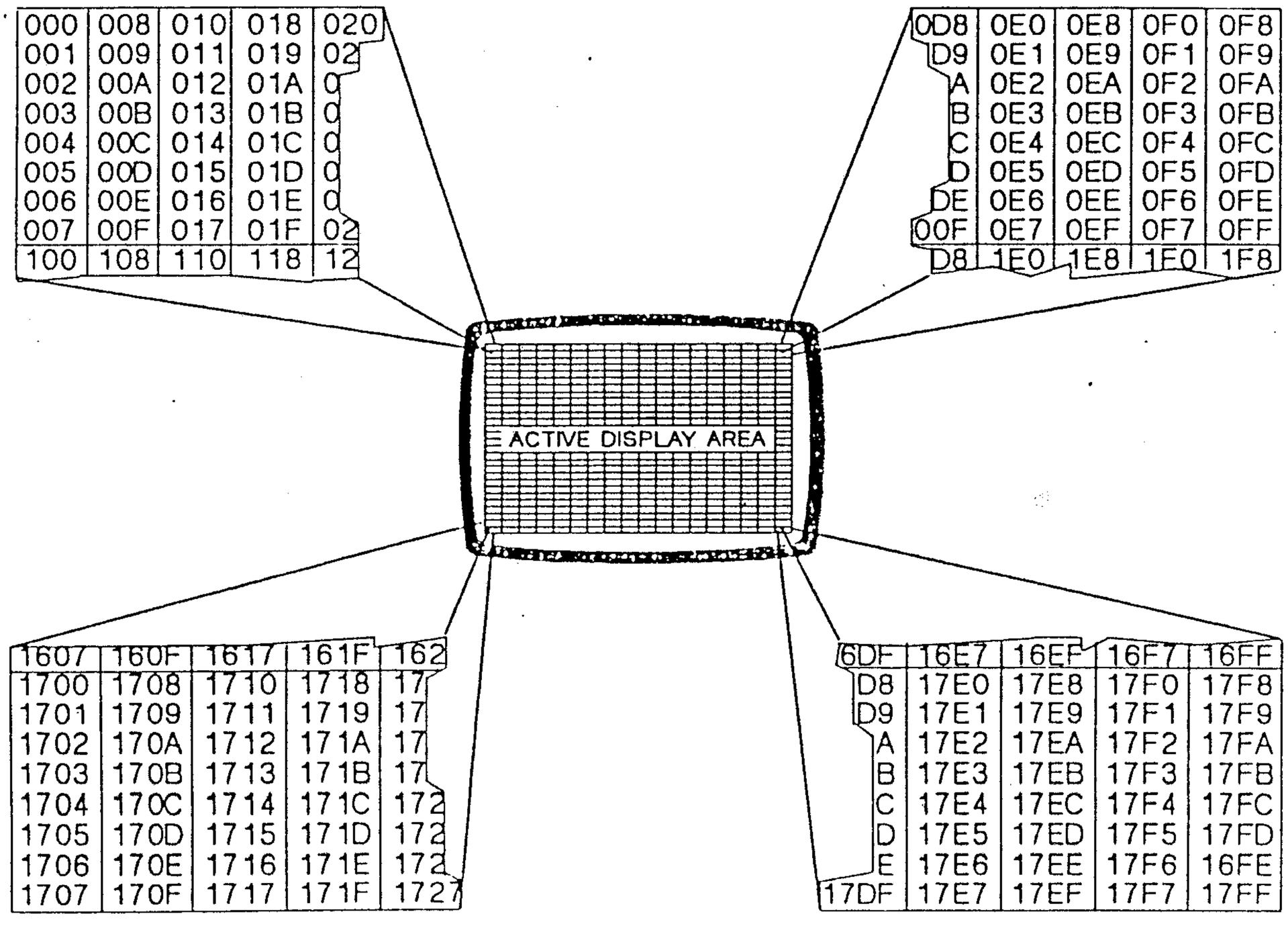
Try this:

1. Set up the GII Pattern Name Table as follows.
[We have already seen that the screen, in Mode II, is split into 3 sections: 0 - 255 then 0 - 255 and finally, 0 - 255.] Load the 3 sections of the Pattern Name Table with the number 0 - 255 three times:-

```
      LD HL,£3C00    ; START OF PATTERN NAME TABLE MTX BASIC
      LD A,L
      OUT (02),A
      LD A,H
      OR £40
      OUT (02),A
      LD C,03        ; 3 TIMES
MAIN: LD B,00
LOOP: LD A,B         ; PUT VALUE OF B IN A
      OUT (01),A
      CP 255         ; HAVE WE DONE 255 YET ?
      JR NZ,LOOP     ; NO DO IT AGAIN
      LD A,C
      DEC A
      CP 00          ; CHECK IF WE'VE DONE IT 3 TIMES YET ?
      JR NZ,MAIN     ; NO SO RE-SET B TO 255
      ALL FINISHED
```

We can now forget about the Name Table .... this will never change in this mode.

Now set all the bytes in the Colour table to £4F this will give you the typical blue & white screen.... o.k. you don't like blue and white....so put in you own values ! Don't forget that you are now filling 6144 bytes - 2048 for each third of the table ! (see last month's edition)



- GRAPHICS II MODE PATTERN TABLE CONFIGURED FOR BITMAPPED GRAPHICS

After studying the diagram you will see that to turn one pixel on which is located at the very first position on the screen we would **write £80 to the first byte of the PATTERN GENERATOR TABLE. ======= £80 = 10000000 BINARY.** In the same manner, to turn on the **last pixel** in the bottom left -hand corner of the screen, we would write 01 to location £17FF in the **PATTERN GENERATOR TABLE. ====== £01 = 00000001 BINARY.**

To make matters easier, we can now write a sub-routine that will calculate the address for any given **X,Y pixel co-ordinate.**

Using the formula that X = 0 - 255 [0-FF Hex]
Y = 0 - 192 [0-C0 Hex] ....see manual.

a: Take the integer of X/8 and multiply it by 8.,,,,, This will give the horizontal offset.

b: Now take the integer value of Y/8 and multiply it by £100 hex.
This now gives us the vertical byte offset to the nearest 8 bits. NOTE: if there is a remainder after calculating Y/8 it is added to the vertcal byte offset to give the VERTICAL STARTING ADDRESS.

c: Add the VERTICAL STARTING ADDRESS to the HORIZONTAL OFFSET and this now is = to the ACTUAL BYTE ADDRESS WE NEED TO WRITE TO.

d: Use the REMAINDER from X/8 to look up in the table below to find the actual data to plot into the BYTE ADDRESS.

Remainder of X/8	Data to write
0	£80
1	£40
2	£20
3	£10
4	£08
5	£04
6	£02
7	£01

Formula: BYTE ADDRESS = 8(INT(X/8))+256(INT(Y/8))+REMAINDER OF Y/8
Data to write = remainder (X/8) from value in table.

Playing Games With VRAM Addressing

So far in Section 9 we have described how to use Graphics II Mode in its normal table driven environment and also how to arrange it as a bitmap. Now we are going to complicate things further by telling you that there are other tricks you can play with the VDP. By experimenting with the values in VDP registers R2 thru R6 (entering nonstandard initialization values), we can obtain some interesting effects.

You should be forewarned that fooling around with VRAM addressing can cause some interesting effects but almost always produces some undesirable side effects such as losing the ability to use sprites or being only able to use a small number of sprites. Rather than dwell too long on this subject, we will describe one interesting new configuration that can be obtained and leave the rest to you.

Table 8-2 shows the register initialization values for the mode about to be described. Note that the only registers containing nonstandard values are Registers 3 and 4 which determine the Color Table and Pattern Table base address.

TABLE 8-2 -- NEW MODE INITIALIZATION VALUES

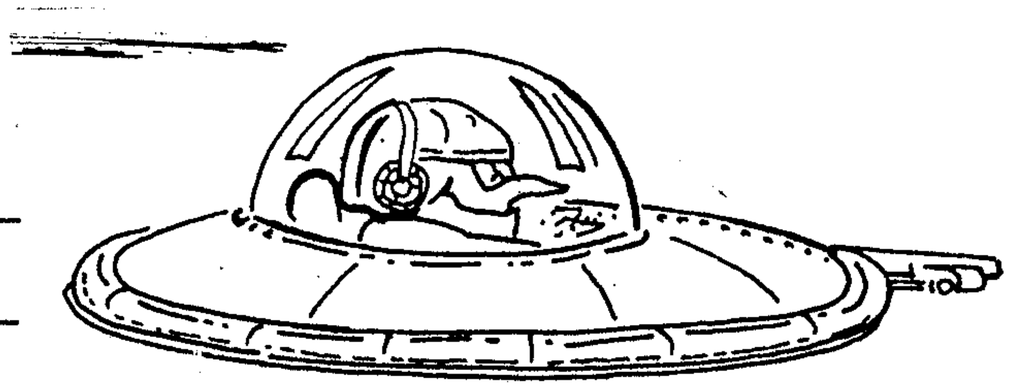
REGISTER	MSB LSB	HEX	DESCRIPTION
REG 0	00000010	02	GRII Mode,No External Video
REG 1	11000010	C2	4116,Enable Disp.,Disable Int.,16x16 Sprites,Mag.Off
REG 2	00001110	0E	Address of Name Table in VRAM = Hex 3800
REG 3	10011111	9F	Color Table Adress = Hex 2000 to Hex 2800
REG 4	00000000	00	Pattern Table Address = Hex 0000 to Hex 0800
REG 5	01110110	76	Address of Sprite Attribute Table in VRAM = Hex 3B00
REG 6	00000011	03	Address of Sprite Pattern Table in VRAM = 1800
REG 7	00001111	0F	Backdrop color = White

What this mode does is effectively shrink the Graphics II Mode Color and Pattern Tables down from Hex 1800 bytes to Hex 800 bytes. This enables us to define up to 256 8x8 pixel patterns and 256 corresponding eight byte Color Table entries. Color is still mapped onto a pattern exactly as in Graphics II Mode.

The 768 byte Name Table is not split up into three equal sections as in Graphics II Mode but works as in Graphics I Mode. A byte of information written anywhere in the Name Table will select the appropriate pattern and the corresponding eight byte color entry and place it on the screen. In Appendix C can be found the Pattern Graphics Address Location Tables.

This mode is useful because it provides memory savings of Graphics I Mode while allowing the color detail available in Graphics II Mode. However, we can no longer can define a unique pattern for each screen position, which is neccesary for highly detailed pictures or for bit mapping the screen. When in this mode we can no longer use 32 sprites. If you try to put more than eight sprites on the screen at once, they will start to duplicate themselves on the screen.

program listing



ROCKET LANDER BY ANTHONY RANSLEY

This is one of the most popular games on the home micro. In old days we used to play this with no graphics ! But, it was so addictive, no other work ever was finalised !

You must land your Space Ship on the landing pad with the minimum of impact. If the impact speed exceeds 10 when your Space Ship touches the landing pad, it will destroy your ship, you, and your crew !

The landing pad is the only safe place to land. If your Space Ship touches the ground, no matter what speed, it will be destroyed !

At the beginning of the game you are given 1000 units of fuel. One short blast of the MAIN BOOSTERS use 3 units of fuel. Using the side rockets burns a further 2 units for each blast. The main boosters are used to slow the Lander down, and the side rockets are used to steer the Ship.

An instrument pad is constantly displayed at the bottom of the screen, and will indicate the SPEED,FUEL, & SCORE.

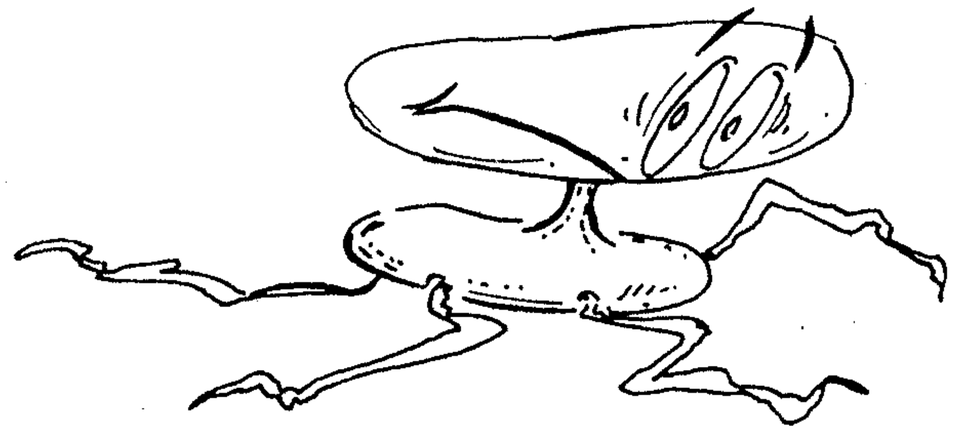
The Lander can be controlled by joysticks or cursor keys. All REM statements have been left so that you can change the program if you want to. Be careful you don't run out of fuel.....HAPPY LANDINGS !!!

Anthony is currently working on a Machine Code arcade game which he hopes to publish in the very near future..

```
10 REM        ROCKET LANDER
20 REM
30 REM       ANTHONY RANSLEY        1984
40 REM
60 REM                  MEMOTECH  MTX 500
70 REM
100 CRVS 2,1,0,20,32,4,32
110 LET HS=0
150 GOSUB 260: GOTO 360
160 FOR J=0 TO 19
170 CSR J,2: PRINT " LANDER"
180 CSR 32-J,2: PRINT "ROCKET "
190 NEXT J
260 LET FUEL=1000
270 LET SCORE=0
280 LET ROUND=0
290 GOTO 320
300 RETURN
320 LET X=5: LET Y=182
330 LET DRIFT=200
340 LET GRAV=1
350 RETURN
360 CTLSPR 2,5
362 CTLSPR 6,2
365 REM ***ROCKET***
370 GENPAT 4,1,3,15,30,63,50,26,14,6
380 GENPAT 5,1,63,47,63,37,41,19,32,248
398 GENPAT 6,1,192,240,120,252,76,88,112,96
392 GENPAT 7,1,252,244,252,164,148,200,4,31
195 REM ***HOUSE***
100 GENPAT 4,2,0,0,0,1,5,2,13,1
 10 GENPAT 5,2,1,15,31,28,28,31,31,31
 20 GENPAT 6,2,4,14,0,14,0,14,14,142
 30 GENPAT 7,2,191,255,255,89,89,249,249,249
 35 REM ***LANDING PAD & GROUND***
 :0 GENPAT 1,129,255,127,0,0,0,0,0,0
 :0 GENPAT 1,130,255,255,0,0,0,0,0,0
 0 GENPAT 1,131,192,128,0,0,0,0,0,0
 5 GENPAT 1,132,0,0,0,0,0,0,0,0
 8 REM ****FLAME****
 0 GENPAT 4,3,1,3,7,7,7,7,3,11
 4 GENPAT 5,3,3,1,5,1,0,5,0,0
 8 GENPAT 6,3,128,192,224,224,224,224,192,208
 ) GENPAT 7,3,192,160,128,144,0,32,0,128
 ( REM ****EXPLOTION 1****
 ! GENPAT 4,4,0,5,16,10,32,9,2,74
 ) GENPAT 5,4,21,3,41,0,36,1,42,4
   GENPAT 6,4,0,64,16,68,16,80,162,208
   GENPAT 7,4,170,64,208,136,34,80,32,132
486 REM ****EXPLOTION 2****
487 GENPAT 4,5,0,0,2,40,2,84,0,5
488 GENPAT 5,5,10,33,6,11,0,36,0,18
489 GENPAT 6,5,0,144,0,36,0,132,18,192
490 GENPAT 7,5,144,98,132,80,132,32,136,32
492 REM **********SET UP SCREEN********
495 REM ***GROUND***
500 GOTO 530
508 COLOUR 0,6: COLOUR 1,15
510 FOR C=0 TO 31
515 CSR C,19: PRINT CHR$(132)
520 NEXT C
522 LET U=INT(RND*25)+2
524 CSR U,19: PRINT CHR$(129);CHR$(130);CHR$(131)
526 SPRITE 3,2,(U+4)*8,47,0,0,1
528 RETURN
 530 VS 4: PAPER 3: CLS
 535 VS 2: CLS
 540 VS 4
 545 COLOUR 4,5
 550 GOSUB 508
 565 COLOUR 3,5
 570 FOR C=0 TO 255
 580 READ L
 590 LINE C,191,C,L
 600 NEXT C
```

→

```
605 COLOUR 3,12
610 ANGLE 5.60
615 PLOT 132,110
620 ARC 90,.2
735 COLOUR 0,1: COLOUR 1,10
740 CSR 1,21: PRINT "SCORE 0     "
750 CSR 13,21: PRINT "FUEL"
760 CSR 23,21: PRINT "DRIFT"
770 CSR 1,23: PRINT "INPACT SPEED";
790 REM ********GAME IN ACTION**********
800 IF INKEY$<>CHR$(26) THEN GOTO 845
810 IF FUEL<5 THEN GOTO 845
815 LET K=6
820 SOUND 3,5,15
830 LET FUEL=FUEL-5
840 LET GRAV=GRAV-2
842 GOTO 850
845 SOUND 3,0,0
848 LET K=0
850 IF INKEY$<>CHR$(8) THEN GOTO 900
860 IF FUEL<2 THEN GOTO 900
870 LET DRIFT=DRIFT+(-5)
880 LET FUEL=FUEL-2
890 SOUND 3,3,15
900 IF INKEY$<>CHR$(25) THEN GOTO 950
910 IF FUEL<2 THEN GOTO 950
920 LET DRIFT=DRIFT+5
930 LET FUEL=FUEL-2
940 SOUND 3,3,15
950 REM ***ROCKET TESTS***
960 LET X=X+(DRIFT/100)
970 LET Y=Y-(GRAV/50)
980 IF Y>46.9 THEN GOTO 1020
990 IF GRAV>10 THEN GOTO 1510
1000 IF X>(U*8)+8 AND X<(U*8)+10 THEN GOTO 1250
1010 GOTO 1510
1020 IF Y>191 THEN GOTO 1160
1030 IF X<0 OR X>255 THEN GOTO 1190
1040 SOUND 1,0,0
1050 SPRITE 1,1,X,Y,0,0,1
1060 SPRITE 2,3,X,Y-14,0,0,K
1070 LET GRAV=GRAV+1
1090 CSR 17,21: PRINT FUEL;" "
1110 CSR 28,21: PRINT DRIFT;" "
1130 CSR 14,23: PRINT GRAV;"  ";
1140 GOTO 800
1150 REM **********GAME OVER**********
1160 VS 5: CLS
1170 CSR 0,5: PRINT "YOUR SPACE SHIP HAS GONE OUT OF ORBIT."
1180 GOTO 1382
1190 VS 5: CLS
1200 CSR 0,5: PRINT "YOUR SPACE SHIP HAS GONE OUT OF LANDING"
1210 PRINT : PRINT "ZONE,AND HAS BEEN SHOT DOWN BY THE"
1220 PRINT : PRINT "DEFENCE SYSTEM."
1230 GOTO 1382
1240 REM ******LANDED SAFLY******
1250 LET SCORE=SCORE+250
1255 SPRITE 2,3,X,Y-14,0,0,0
1260 IF INT(X)=129 THEN LET SCORE=SCORE+250
1265 SOUND 3,0,0
1270 FOR C=1 TO FUEL
1285 SOUND 1,C,15
1290 LET SCORE=SCORE+3
1300 CSR 6,21: PRINT SCORE
1310 LET FUEL=FUEL-1
1320 CSR 18,21: PRINT FUEL;" "
1330 NEXT C
1340 SOUND 1,0,0
1350 LET ROUND=ROUND+75
1360 LET FUEL=1000-ROUND
1370 GOSUB 320
1375 GOSUB 508: COLOUR 0,1
1380 GOTO 800
1382 IF SCORE>HS THEN LET HS=SCORE
1385 PRINT : PRINT "YOU SCORED";SCORE
1390 PRINT : PRINT " HI-SCORE";HS
1392 PRINT : PRINT "       ANOTHER GAME ?   YES/NO."
1395 SOUND 3,0,0
1395 SOUND 3,0,0
1398 GOSUB 160
1400 IF INKEY$="N" THEN STOP
1410 IF INKEY$<>"Y" THEN GOTO 1400
1420 VS 4
1425 GOSUB 508: COLOUR 0,1
1430 GOTO 800
1500 REM ****EXPLOTION****
1510 LET DEL=10
1520 FOR N=15 TO 0 STEP -.75
1525 SPRITE 1,4,X,Y,0,0,9
1530 SOUND 3,6,N
1535 SPRITE 1,4,X,Y,0,0,0
1540 FOR J=0 TO DEL
1542 SPRITE 2,5,X,Y,0,0,6
1544 PAUSE 3
1546 SPRITE 2,5,X,Y,0,0,0
1550 NEXT J
1592 LET DEL=DEL-1
1570 NEXT N
1580 SPRITE 1,4,X,Y,0,0,0
1590 SPRITE 2,5,X,Y,0,0,0
1600 PAUSE 150
1605 CSR 6,21: PRINT " 0    "
1606 CSR 13,23: PRINT "    ";
1610 VS 5: CLS
1620 CSR 0,5: PRINT "YOUR SPACE SHIP WAS DESTROYED AS IT HIT"
1630 PRINT : PRINT "THE GROUND AT AN INPACT SPEED OF";GRAV
1640 GOTO 1382
1700 DATA  107,108,109,109,110,111,112,112,113
1710 DATA  114,114,115,116,116,117,117,118,118
1720 DATA  119,120,120,121,122,123,123,124,125,126,126,127,127,127,127,126
1730 DATA  126,127,127,127,127,128,128,129,129,130,131,131,132,133,134,134
1740 DATA  135,135,136,136,136,137,137,138,138,139,139,139,139,139,140,140
1750 DATA  139,139,139,140,140,140,141,140,140,140,139,139,138,137,136,135
1760 DATA  135,134,134,134,133,133,133,133,132,132,132,131,131,130,130,130
1770 DATA  129,129,128,127,127,126,125,124,123,123,122,121,120,120,119,118
1780 DATA  117,117,116,116,116,115,115,115,114,114,113,113,113,112,112,111
1790 DATA  111,110,110,111,112,113,113,114,114,115,115,116,116,117,117,118
1800 DATA  118,119,119,119,119,120,120,121,122,122,122,123,123,123,123,124
1810 DATA  124,124,125,125,125,126,126,127,127,127,128,129,130,131,131,132
1820 DATA  133,134,134,135,135,136,137,137,138,138,139,139,139,140,140,140
1830 DATA  140,141,141,141,141,142,142,142,142,142,143,143,144,144,144,144
1840 DATA  145,145,145,145,146,146,146,146,147,147,147,148,148,149,149,150
1850 DATA  150,151,151,152,152,153,153,154,154,155,155,156,156,157,156,155
1860 DATA  155,154,153,153,152,152,151,151,150,150,150,149,149,149
```



PROGRAMMING

SPRITE DETECTION by ROBIN HEYDON

The program is an example of how the code can be interfaced in a Basic program.

The code starts by searching through the sprite table, working out the X **and** Y co-ordinates of each sprite and comparing them with every other. As soon as two sprites coincide the code loads two addresses (£FA50 & £FA51) with the numbers of the two sprites which coincide.

The code has one major disability, that is if more than two sprites coincide only the first two register: **e.g. if sprites 2, 6 and 12 are all on the same spot, only sprites 2 and 6 will register.**

The code uses one other address £FA49. This address is proportioned to the **CTLSPR 6** command. It alters the area of the collision check, i.e. the larger the number at £FA49, the larger the area checked. You can change this number by changing the values of register **A as 54, 58 and 516 in the code. The code exists after the table OUT.**

I tried using interupts but the routine was so slow it filled the screen with "HELLO" in 4.4 seconds, only 2.5 times slower than normal!! The code can be increased by changing the values of the CP for B and C at £4107 and £411B respectively, i.e. if you want only sprite 1 and sprite 2 to collide with any another sprite you would leave £410B as CP£21 but change £411B to CP£03, one more than is required **(£4107 and £411B are as in the source listing).**

How to find the position of a sprite.

First find the position of your sprite in the sprite table.

Position = No. of sprite * 8 + 65101

Y speed = Pos
Y pos Msb = Pos+1
X speed = Pos+3
X pos Msb = Pos+4
X pos Lsb = Pos+5
Pat Number = Pos+6
Colour = Pos+7

To find position of sprite 2

Position = 2*8+65101
= 16+65101 = 65117
Y pos Msb = 65118
Y pos Lsb = 65119
X pos Msb = 65121
X pos Lsb = 65122

You now need to know one other piece of information. The Y & X change for each magnitude and size.

SIZE	MAG	CTL SPR 6	Y CHANGE	X CHANGE	AREA
0	0	0	219 :£DB	16 : £10	8*8
0	1	1	215 :£D7	24 : £18	16*16
1	0	2	215 :£D7	24 : £18	16*16
1	1	3	207 :£CF	28 : £1C	32*32

Y=Y CHANGE-(Y pos Msb * 16 + Y pos Lsb/16)
X=(X pos Msb * 16+ X pos Lsb/16)-X change
E.6 for SPRITE 2 : Pos = 65117 CTLSPR 6,2

£FE5D	[65117]	Y SPEED	£00
£FE5E	[65118]	Y MSB POS	£08
£FE5F	[65119]	Y LSB POS	£50
£FE60	[65120]	X SPEED	£00
£FE61	[65121]	X MSB POS	£0D
£FE62	[65122]	X LSB POS	£A0
£FE63	[65123]	PAT NUMBER	£20
£FE64	[65124]	COLOUR	£0C

Y CHANGE = 215 & X CHANGE = 24
X = Msb Pos * 16 + Lsb Pos /16 - X
= 13 * 16 + 160/16 - 24
= 208 + 10 - 24
= 194
Y = CHANGE - Msb Pos * 16 + Lsb Pos/16
= 215 - 8 * 16 + 80/16
= 81

These values would have been obtained after the statement:

SPRITE 2,8,194,81,0,0,2

My first contact with Robin was when he took me to task regarding a mistake we had made when printing edition One. I threw the gauntlet to him, and this is the result. Robin is a typical example of the younger members..... they know their stuff! We shall be hearing more from Robin in future editions.

I urge you to type in the listing complete with the basic demo.... you're in for a nice surprise..... Ed.

```
0 GOTO 100
1 CODE

4010 CONC:   LD A,0
4012         LD (£FA50),A
4015         LD (£FA51),A
4018         JP JUMP1
401B OUT:    CP B
401C         LD (£FA50),A
401F         LD A,B
4020         LD (£FA51),A
4023         RET
4024 JUMP1:  LD A,(£FF55)
4027         CP 00
4029         JR Z,S4
402B         CP 1
402D         JR Z,S8
402F         CP 2
4031         JR Z,S8
4033         CP 3
4035         JR Z,S16
4037         LD A,0
4039         LD B,0
403B         JR OUT
403D S4:     LD A,4
403F         JR JUMP2
4041 S8:     LD A,£08
4043         JR JUMP2
4045 S16:    LD A,£10
4047 JUMP2:  LD (£FA49),A
404A         LD IX,£FE55
404E         LD IY,£FE55
4052         LD BC,£0101
4055 LOOP1:  LD DE,£0000
4058         LD A,(IX+4)
405B         CP £12
405D         JP NC,SAME
4060         SLA A
4062         SLA A
4064         SLA A
4066         LD D,A
4067         LD A,(IX+5)
406A         SRL A
406C         SRL A
406E         SRL A
4070         SRL A
4072         SRL A
4074         ADD A,D
4075         LD D,A
4076         LD A,(IX+1)
4079         CP £01
407B         JP C,SAME
407E         CP £0E
4080         JP NC,SAME
4083         SLA A
4085         SLA A
```

```
4087          SLA A
4089          LD E,A
408A          LD A,(IX+2)
408D          SRL A
408F          SRL A
4091          SRL A
4093          SRL A
4095          SRL A
4097          ADD A,E
4098          LD E,A
4099 LOOP2:   LD A,B
409A          CP C
409B          JR Z,SAME
409D          LD HL,£0000
40A0          LD A,(IY+4)
40A3          CP £12
40A5          JR NC,SAME
40A7          SLA A
40A9          SLA A
40AB          SLA A
40AD          LD H,A
40AE          LD A,(IY+5)
40B1          SRL A
40B3          SRL A
40B5          SRL A
40B7          SRL A
40B9          SRL A
40BB          ADD A,H
40BC          LD H,A
40BD          LD A,(IY+1)
40C0          CP £01
40C2          JR C,SAME
40C4          CP £0E
40C6          JR NC,SAME
40C8          SLA A
40CA          SLA A
40CC          SLA A
40CE          LD L,A
40CF          LD A,(IY+2)
40D2          SRL A
40D4          SRL A
40D6          SRL A
40D8          SRL A
40DA          SRL A
40DC          ADD A,L
40DD          LD L,A
40DE          PUSH BC
40DF          LD A,(£FA49)
40E2          LD B,A
40E3          LD A,D
40E4          SUB B
40E5          CP H
40E6          JR NC,NOC
40E8          LD A,D
40E9          ADD A,B
40EA          CP H
40EB          JR C,NOC
40ED          LD A,E
40EE          SUB B
40EF          CP L
40F0          JR NC,NOC
40F2          LD A,E
40F3          ADD A,B
40F4          CP L
40F5          JR C,NOC
40F7          POP BC
40F8          LD A,B
40F9          LD B,C
40FA          JP OUT
40FD NOC:     POP BC
40FE SAME:    PUSH DE
40FF          LD DE,£0008
4102          ADD IY,DE
4104          POP DE
4105          INC C
4106          LD A,C
4107          CP £21
4109          JP NZ,LOOP2
410C          LD C,£01
410E          LD IY,£FE55
4112          PUSH DE
4113          LD DE,£0008
4116          ADD IX,DE
4118          POP DE
4119          INC B
411A          LD A,B
411B          CP £21
411D          JP NZ,LOOP1
4120          LD A,0
4122          LD B,0
4124          JP OUT
4127          RET
```

```
2 RETURN
100 CTLSPR 0,1: CTLSPR 1,0: CTLSPR 2,32: CTLSPR 3,32: CTLSPR 4,0: CTLSPR 5,32: CTLSPR 6,0
110 GENPAT 3,0,60,66,129,129,129,129,66,60
115 VS 4: CLS
120 FOR F=1 TO 32
130 SPRITE F,0,RND*120+60,RND*90+45,RND*10-5,RND*10-5,1
135 NEXT
150 GOSUB 1
160 LET SPR1=PEEK(64080)
170 LET SPR2=PEEK(64081)
175 IF SPR1=0 THEN  GOTO 210
180 CSR 0,0
190 SPRITE SPR1,0,RND*120+60,RND*90+45,RND*10-5,RND*10-5,RND*10+3
200 SPRITE SPR2,0,RND*120+60,RND*90+45,RND*10-5,RND*10-5,RND*10+3
210 GOTO 150
```

STOCK SITUATION SOFTWARE .....

BUYING & BACKUP

LEVEL 9 ADVENTURES £8.75 [Club Price]

GENPAT HAS STOCKS OF ALL LEVEL 9 ADVENTURES INCLUDING "RETURN TO EDEN" THE VERY LATEST , AND IN MY OPINION, THE BEST TITLE YET !

POTHOLE PETE
WE HAVE A GOOD STOCK OF THIS EXCELLENT CONTINENTAL TITLE £6.02p [Club Price]

QOGO.....CONTINENTAL HAVE STILL NOT RECEIVED THIS TITLE IN STOCK HOWEVER WE DID MANAGE TO PERSUADE THEM TO LET US HAVE A LIMITED QUANTITY WITHOUT THE COVERS..... CONTINENTAL SAY STOCKS OF THIS AND OBLOIS SHOULD BE IN BY 10TH DECEMBER ..... THIS ALSO APPLIES TO 3D TACHYON FIGHTER.

XAVERSIENE.....COMPOSER.....XAVERSIENE HAVE ASSURED ME THAT STOCKS WILL BE AVAILABLE BEFORE CHRISTMAS ON THIS EXCELLENT MUSIC PROGRAM. WE WILL BE THE FIRST IN LINE TO RECEIVE THIS PACKAGE .....£13.00 [Club Price]

ALL OTHER TITLES IN STOCK AT GENPAT EXCEPT FOR TRI-COM SOFT.

HARDWARE STOCK SITUATION

ALL HARDWARE IS EX STOCK DELIVERY BY SECURICOR OR REGISTERED POST.

THIS DOES NOT INCLUDE THE 3 NEW DISC DRIVES WHICH ARE 28 DAYS DELIVERY FIRST COME FIRST SERVED.

SPECIAL NOTE ON MACHINE CODE LISTINGS....

I have had a lot of phone calls from people who do not understand the difference between the 512 & 500. If you have a MTX 500 you can still type in the listings that are published but your addresses will START with £80 not £40... the reverse is true for the 512. However, if there are any USER calls to ABSOLUTE ADDRESSES THEN YOU MUST ADD (500) OR SUBTRACT (512) £4000 Hex (16384 Decimal). E.g USER (32460) FOR 512 WOULD BECOME USER (48844) FOR THE 500.

NEW DISC DRIVES FROM MEMOTECH FROM £199.00

UPGRADE

There is a 250 K disc drive available in 28 days @ £249 including RS232 interface... if you are a Member of Genpat. Order direct from Memotech but you must quote your Membership number !!!! Cheques will not be cashed until you take delivery.

SOFTWARE

NEWWORD 32K add-on rom. This is an excellent word processor. In fact this magazine is prepared with the disc version of this package. The Rom version is as powerful as the disc version and comes complete with manuals. £75.00

Hi-soft PASCAL 16K add-on rom. This is one of the most powerful versions of Pascal on the market today. Adapted for the MTX, the compiler supports peek, Poke, Vs, Paper, Ink, Plot, Line, Addr, Crvs, User, and many more. If you're fed up with Basic try Pascal.....£60.00

BUSINESS PROGRAMS

PAYROLL : Handles up to 170 employees on a MTX512. Prints payslips, caculates payments due to Inland Revenue and denominations and quantities of coins and notes required for cash paid employees. £25.00

PURCHASE LEDGER : The program provides for up to 35 purchase ledger accounts, 20 expense codes and supplies accurate purchase ledger balances as required. £15.00

SALES LEDGER : Not yet available but probably priced at £15.00

BASIC BUSINESS : Loan repayment, forecasting, breakeven analysis, job costings, equipment analysis,inventory use projections. £5.95

SOFTWARE @ £6.95

GRAPHICS : A powerful graphics tool, this program allows you to design 24 User Definable Graphics, 96 Sprite Patterns and 96 standard ASCII codes.

NEMO : Harpoon the sharks before they enjoy you ! Multi-level game.

KILOPEDE : Eliminate the mushrooms and smash the Kilopede before it breaks into segments. Watch out for the nasties - killer crabs,fleas (ugh),spiders,and jellyfish (ow!).

SUPER MINEFIELD : Cross the minefield armed only with your detector. My wife loves this one. A game of logic and deduction. Nice one Ed.

BLOBBO : A maze game with nothing much different, but good graphics and difficult to master. You won't get bored with this in a hurry!

PHAID : Real fast arcade action. The more aliens you kill the faster it gets. Try to keep your eyes from crossing on sheet 15.

MISSION ALPHATRON : If you like Scramble, you'll love this. Superb graphics, and plenty of them.

TOADO : You've all got this one - what did you do with the tadpoles ?

OBLOIDS : Yes another maze game. See this issue for review.

TAPEWORM : Ugh! Make sure out you don't eat your own tail, or you're in trouble. Good action on this one.

CONTINENTAL RAIDERS : A shoot-em-up space game. If you get the first wave, watch out for the next. Try to hit the Mother ship for extra points.

ASTROMILON : Real Arcade Action with this one. Kamikazi aliens try to obliterate you. Good graphics.

ASTRO PAC : Assemble your spaceship and fight off the aliens before your oxygen supply runs out.

POT HOLE PETE : Help Pete get through over 30 different screens, collecting keys and points as he goes.Reviewed in Issue 2.

QOGO : One the most addictive games ever written. Bounce Qogo around the pyramid of diamonds but watch out for bouncing bombs and bonus swirlers.

MUSIC PAD : Reviewed in Issue 2.
SNAPPO : Race around the maze gobbling fruit, dots and power packs.
DENNIS & THE CHICKEN : Climb the ladders and platforms collecting eggs and bird seed. Watch out for the Frantic Farmer, Carnivorous Chicken and Deadly Duck. Sounds great, but is **not yet available.**
DENNIS GOES BANANAS : The sequel to Dennis & the Chicken, this game involves apes and bananas as well. You've guessed it ! Not yet available.
MISSILE KOMMAND : A classic arcade favourite with the added bonus of fantastic MTX graphics. **Not yet available.**
THE ZOO GAME : An adventure with animals. See this issue for a review.
MAXIMA : See this edition for review of this game. Highly recommended.
GAUNLET : Details on this are top secret. **Not yet available.**
M CODER : Compiles BASIC to machine code. **Not yet available.**
COBRA : Let you as cobra eat the fruit as you guide it around the screen, but don't touch those walls,and watch out for the rocks.
JOHNNY REB : A good game. Set in the Civil War, you've got to capture the enemy's flag. From one of the best War Game software companies around.
MURDER AT THE MANOR : A traditional adventure in the 'Who-dunnit' mould.
THE KEY TO TIME : Be your favourite Time Traveller in this quest through time to discover the five fragments of the Key To Time.
FRANTIC FREDDIE : Cimb the moving ladders on to the conveyor belts. Great graphics. Available soon.

SOFTWARE @ £7.95
STAR COMMAND : See this issue for a review. Mr Spock there's a Klingon stepping on your ear!
DRAUGHTS : You should have this one.
TURBO : See review in this issue. Enough said !.
3D TACHYON FIGHTER : Pilot your laser equipped fighter through towers, aliens, aliens in fighters, etc.

SOFTWARE @ £8.95
KNUCKLES : A fast action maze game with good graphics and sound.
BACKGAMMON : Good graphical representation of the dice on this one. You can learn the game from this program.
REVERSI : Continental's version of Othello. I like this game. Four levels, and the computer plays a good game.

SOFTWARE @ £9.95
CHESS : Excellent implementation of the game. Ten levels - although on the higher levels you will need plenty of time.
FIRST LETTERS : Excellent program for the young child. I love the way the man builds his car when you get a question right.
WORDS & PICTURES : Another good program for the young child .
MATHS 1 : O Level Maths. It's all here - Venn Diagrams,fractions, decimals,number bases....includes index.
PHYSICS 1 : O Level Physics.This program is spoiled by an annoying bug. Could have been put right by the time you read this.

SOFTWARE @ £9.90
SNOWBALL : ADVENTURE QUEST : LORDS OF TIME : DUNGEON ADVENTURE : COLOSSAL ADVENTURE : RETURN TO EDEN.
These programs must rank as the Rolls Royce of Adventure games. I haven't seen a bad review yet for these.

SOFTWARE @ £5.95
SPELLI-COPTER : Over 800 words to test and teach your child. Makes spelling fun for children up to 11 years old.
HELLI-MATHS : Interactive maths program for children up to 11 years old.

OTHER SOFTWARE
UTILITIES 1 : Renumber, binary/hex/decimal convertor, 40 column text and graphics and Data save and load. £4.95
TUMBLEDOWN TOWN : An adventure game which is very good value at a special offer price of £4.50, but hurry it can't last.
CHARMKATZ : Cross between tapeworm & kilopede with adventure type structure. Should be avialable soon.
COMPOSER : A music composer program with many facilities at £14.95.

All the above software is available from GENPAT

CLUB PRICES: RETAIL = £8.95 ....YOUR PRICE £7.80p
RETAIL = £7.95 ....YOUR PRICE £6.95p
RETAIL = £6.95 ....YOUR PRICE £6.02p
LEVEL 9 ADVENTURES ..... CLUB PRICE £8.75p

SYNTAX SOFTWARE

GENPAT's very own software label brings you the following programs.

EDASM : A macro Z80 assembler with a comprehensive 19 page instruction book at the incredible price of £7.95 (around half the price most software houses would sell it for). An amazing piece of programming.

SALTY SAM : Collect the treasures, before the octopi get you!.
Club price - £4.95.

DOODLEBUG DESTROYER : This is really fast ! This is the first game imported from Norway. Club price - £4.95 (available mid December).

SPECIAL CLUB HARDWARE PRICES

[all prices include VAT and postage]

MTX 500 - £174 MTX 512 - £249 RS 128 - £346.96

RAM extensions - 64K £73.91 DMX80 Printer - £233.91
- 128K £141.90 Cable - £12.04

RS232 communications board - £53.17 Oxford ring node kit - £44.48

FDX Single Drive - £346.00 (requires RS232)

UP1 Upgrade 1 Kit (upgrades single FDX to full 80 column) - £192.00
UP2 Upgrade 2 Kit (coupled with UP1 upgrades FDX single to twin floppy CP/m system) - £218.00

FDX System - £759.50
FDX/Silicon Disc - £865.00
256K Silicon Disc Expansion - £334.78

PROGRAMMING IN PASCAL CONTINUED FROM PAGE 10

```
{ *****END OF PRINTER EXTENSIONS**** }

BEGIN
pon;
double(1);
write('Hello Genpat readers');cr;cr;double(0);
emphasis(1);write('This is my latest program in Pascal for you');cr;
emphasis(0);
ELITE(1);write('This listing should give you some idea');emphasis(1);write('of the power of the routines');
elite(0);emphasis(0);
bell;pitch(11);cr;
underline(1);write('All');underline(0);
writeln(' the usual Epson type printer commands are available');
writeln('with some experimentation you should be able to produce some clever output.');
write('Subscripts like ');subscript(1);write('this');subscript(0);write(' do work so do super');superscript(1);write('scripts.');superscript(0);
poff
END.
```

Editorial Note .....

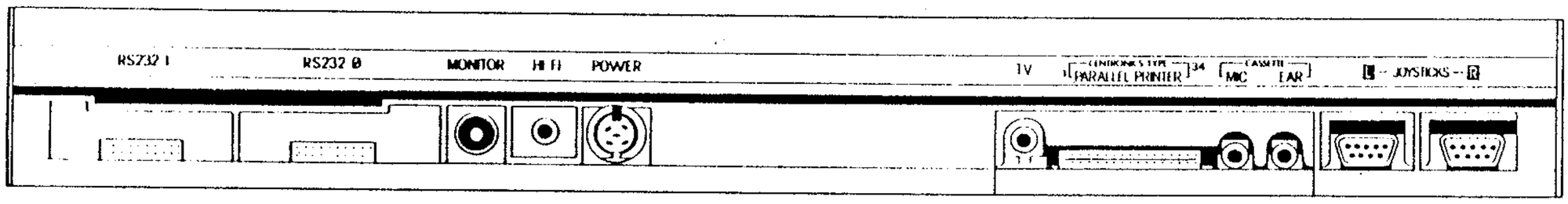
On the front cover you will find that we have said there would be a 'Teach-in' on using the **Front Panel** unfortunately, this article was promised to us by Memotech..... needless to say...**it didn't arrive** !!

However, I shall personally deal with the subject in the next edition also, I will be starting a series on **CP/m and the FDX.** If any members have discovered any brilliant ways of using the FDX I would like to hear from them.

In January we will be starting the promised **conversion of Connect Four** to machine code. I am sorry that this has been held over, but the orginal intention was to keep the magazine to 30 to 32 pages..... we are finding that this is not possible... at 40 pages it is a real hard slog.

RAFFLE .....

DON'T FORGET TO SEND YOUR £1.00 AND MAKE SURE YOU HAVE WRITTEN YOUR MEMBERSHIP NUMBER ON THE TOP ! WE SHALL NOW DRAW THIS ON CHRISTMAS EVE AND YOU WILL BE NOTIFIED OF THE WINNER IN THE JANUARY EDITION.....



END STATEMENT

I must start off with an apology for the magazine being late. You wouldn't believe the difficulties that have been involved with the production of this edition.... the magazine covers didn't arrive on time... the printer broke down 3 times in a week, and when we decided to get a new one, the shiny new machine thanked us by breaking down twice within two days, and once over the weekend which lost us over a week in the long run !

Phone calls the situation is getting drastic. Some members cannot get through. I am personally taking over 160 calls a day, and this doesn't leave much time for me to earn a living. I must have my book finalised by the middle of December, and the original intention was to run the club in my spare time. However, I didn't realise we would recruit so many owners in such a short time... the situation is now out of hand. Therefore, until January, I shall not be available until after 6.00pm on Wednesdays & Fridays. In January we will revert to the orginal times.
My sincere thanks to all those people who have sent in programs... some of them are brilliant! My thanks also to the members who have helped out when the going got rough.

The New User Manual problem has not yet been resolved, but hopefully, we shall be in a position to supply this in the not to distant future.

Many new companies are starting to take an interest in the MTX and this is good news. The Computer College is all set up ready to help you with your Assembly Language problems..... but only if you have purchased the book. Sentient Software have some really exciting programs in the pipeline including a **Flight Simulator. Forth** is on the horizon from Membrain Software, and **Artic** are starting to program for the computer. **Megga Games** have some really fantastic programs in progress, and I mean **fantastic.**

Finally, I would like to welcome all the new members who have joined during this past month, and I am sorry if they have had to wait just that little while longer for their magazine. I shall be happy to help all the members who have purchased their first computer and are experiencing difficulties.

Don't forget to include 50p for any information sheets and a S.A.E for any replies you may require. A very final word. I would like to thank TIM & John for their help, and my wife, who has given up all her free time during the past three weeks - including her weekends - to help with the mail, and orders.

REMEMBER.... BLACK IS BEAUTIFUL..... HAPPY PROGRAMMING.

Chief Advisor to Genpat:Jeff Wakeford [Memotech]:Technical Consultant:Geoff Boyd [Memotech]*Pascal Consultant:Stephen Varley [Membrain]*Software Consultant:Jim Wills [Memotech]*System Consultant:John Mullins:Artwork Mike [Graphic Ad]*Print Consultant:Chris Love [Love's Printing Services]*Tea Maker: Patricia [My Wife]*<C>**Genpat84**

You really can't go wrong with any Level 9 game as they are really brilliant

CRASH MICRO SEPT '84.

"Whichever machine you own, if you have the vaguest tendency towards adventure playing then you must try one of these games (unfortunately you'll probably end up wanting to buy the lot!)."

Computing Today, August 84

"To me, all Level 9 adventures create a remarkable atmosphere because the descriptions sound so life-like. This is where so many other adventures fail."

Crash, July 84

"But it's not just the size of the game it's the quality as well that is astonishing ... ... scenes to fire the imagination."

PCG, April 84

"As in all Level 9's adventures, the real pleasure comes not from scoring points but in exploring the world in which the game is set and learning about its denizens."

Which Micro?, February 84

"(LORDS OF TIME). As we have come to expect from Level 9, the program is executed with wonderful style ... Highly recommended."

PCW, 1 February 84

"I thoroughly recommend these Adventures, they are excellent value for money. No self-respecting Adventure-addict should be without them. I believe Level 9 are producing a series of Adventures which should be regarded as classics."

Atari User, July 84

"These programs run very fast and there are no frustrating pauses. Level 9 Adventures are superbly designed and programmed, the contents first rate. The implementation of Colossal Adventure is nothing short of brilliance; rush out and buy it. While you're at it, buy their others too. Simply smashing!"

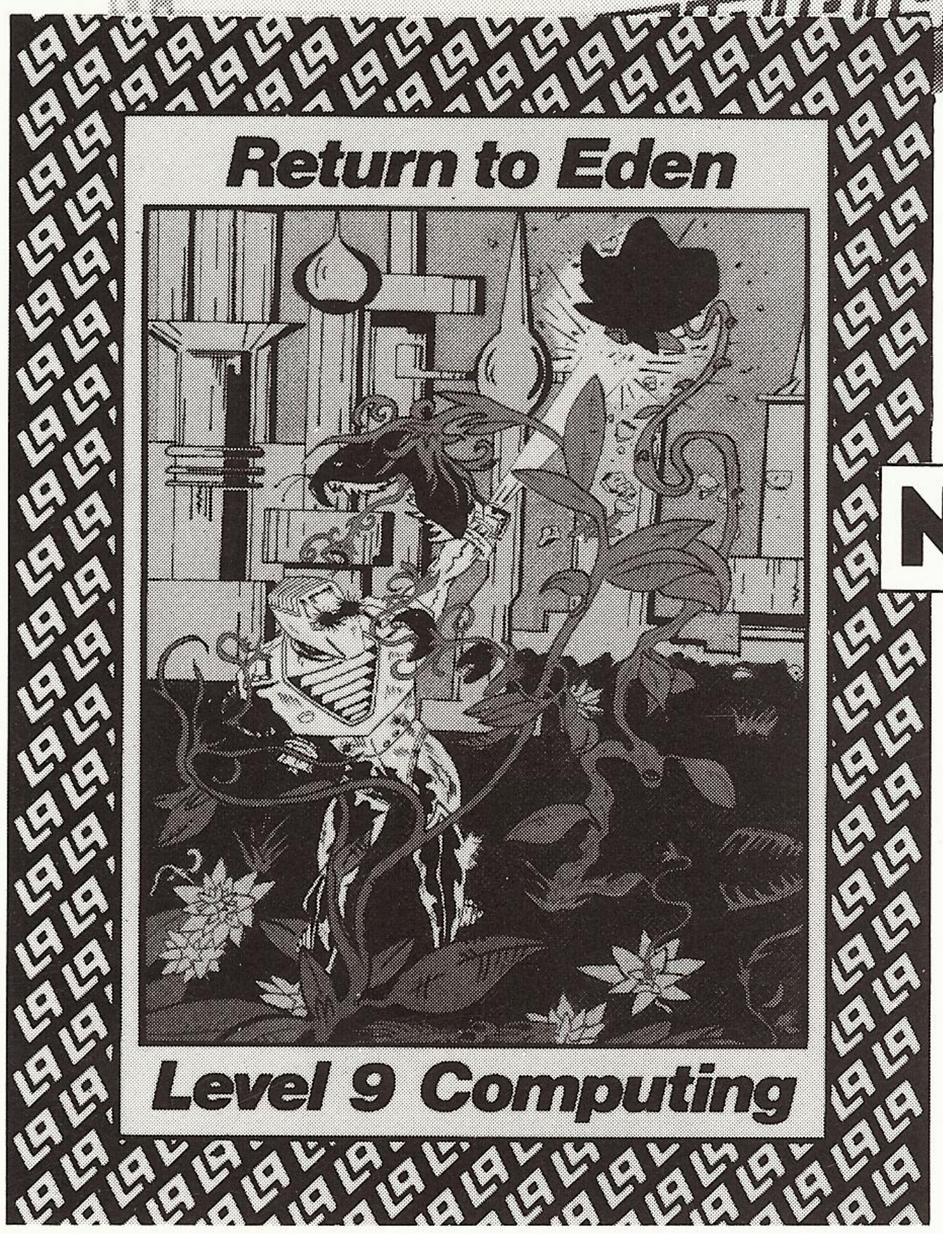
Your 64, June 84

"Level 9 — arguably the producer of the best adventure games in the UK — has done it again. LORDS OF TIME is a sparkling addition to its stable of winners."

Acorn User, July 84

"(SNOWBALL). This is another imaginative, massive-scaled immensely enjoyable adventure from those experts down at Level 9 Computing."

Your Computer, March 84



Level 9's epic adventures are now here for the AMSTRAD. Disk versions are available for the BBC (40/80 track) and Commodore 64. And, best of all, RETURN TO EDEN is ready. It's been a busy month!

RETURN TO EDEN is the long-awaited sequel to Level 9's top-selling Snowball adventure. Now it's here with 220 locations, masses of puzzles, and with pictures on the CBM and Spectrum versions.

NEW

Available from W H Smith and good computer shops everywhere. If your local dealer doesn't stock Level 9 adventures yet, get him to contact us or: Centresoft, Microdealer UK, Ferranti & Craig, Leisuresoft, Lime Tree, LVL, PCS, R & R or Wonderbridge.

	Cassette £9.95	Disk £11.95
1. **COLOSSAL ADVENTURE.** The classic mainframe game with 70 bonus rooms.	☐	☐
2. **ADVENTURE QUEST.** An epic journey through Middle Earth.	☐	☐
3. **DUNGEON ADVENTURE.** A massive game which completes the Middle Earth Triology.	☐	☐
4. **SNOWBALL.** Save the interstar freezer, Snowball 9, in a huge space adventure with over 7000 locations.	☐	☐
5. **RETURN TO EDEN.** SF adventure on the weirdest planet ever. The sequel to Snowball, though you don't need to have played this.	☐	☐
7. **LORDS OF TIME.** A humorous romp through World History.	☐	☐

I ENCLOSE A CHEQUE/PO FOR £9.95 EACH (CASSETTE) OR £11.95 EACH (DISK) FOR BBC OR CBM 64

My name: .....

My address: .....

.....

.....

My micro is a: .....

(one of those listed below, with at least 32K of memory).

Contact:

LEVEL 9 COMPUTING

Dept. MP2 , 229, Hughenden Road, High Wycombe, Bucks. HP13 5PG.

AMSTRAD BBC CBM64 SPECTRUM MTX ORIC 1 LYNX NASCOM ATARI