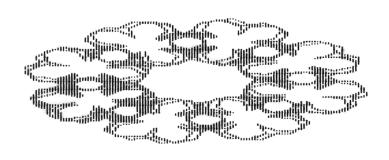THE
MEMOTECH OWNERS CLUB
MAGAZINE

FEATURES:-

ASSEMBLER

CONCLUDING THE VDP

NEWS OF NEW DISCS

ARRAYS EXPLAINED

COMPETITION

<u>EDITORIAL</u>   (March 1985)

Phil Eyres   Richard Adams
23 Denmead Road   18 Nightingale Rd
Harefield   Pilands Estate
Southampton   Bursledon
SO2 5GS   Southampton

Well, it long last we have some firm news about the new 100K &
250K disc drives, Memotech have informed us that these single
disc drives will be available from the end of May, (they also say
that they are doing there utmost to bring this date forward!).
The cost of these drives will be as follows :

100K Disc Drive   199.00 incl. Vat and P & P.
250K Disc Drive   250.00 incl. Vat and P & P.

Communications board 35.00 incl.
(not RS232)

I think you must agree that these offer very good value for
money, they infact compare very well with drives available for
other makes of computer.

Memotech have also made a New Years Resolution, they have vowed
to give better support to there dealers, Yippeee!!!!!!. I hope
that this means that they will also be pushing this brilliant
machine a little harder in the shops.

Also more good news, we mentioned Megastar last month, well they
have 6 new software releases available shortly :
Son Of Pete
Escape From Zarkos
Chamberoids   All at £6.95
Qogo 2
Fathoms Deep
Surface Scanner

And yet more good news!, There are rumours of a Teach Yourself
M/code On The MTX, this is not available yet, but who knows?,
hopefully more in the near future.

As if all that is not enough, take a look at our software
library! :-
1. Hex-Dec-Bin Convertions (Binary bit in Assembler)
2. CGEN Sprite Generator
3. 3D Drawing Board
4. Whist *** New ***
5. Memory Save *** New *** This program will save a block of
memory to tape or retrieve it.
6. MTX Drawing Board *** New *** Requires DMX 80 Printer for full
implimentation
7. Simplex Tableaux, probably the best that we have. Our first
applications program. **See page 10 for further Details!!**
                                                      Phil

DEALING WITH ARRAYS

We have been receiving a lot of inquiries about the use of arrays over the past few weeks, so we thought that a short article on the use of them may be useful to some members.

If you are planning a program to store a lot of information, the best way to do this is by the use of arrays. As an example, a recent enquiry was about how to store 8 lots of information about each stamp in a stamp collection, half of the information was to be permanent and the other half was to be variable. The best way to do this is to set up two arrays, say A and B.

If you wanted to 'log' 1000 stamps you could DIM A(1000,4) and DIM B(1000,4). Therefore, if you want to look up stamp number 110 you know that elements 110 of array A and B will relate to that stamp. A stamp collector will realise that this is a simplification but the basic idea is correct.

If the information about the stamps was to be words and not just numbers obviously the DIM statement would have to be changed to DIM A$(1000,4,Y) where Y is the length of the longest string. The '$' sign denotes a string of characters.

Care should be taken not to use larger numbers than necessary in such a DIM statement as memory is 'used-up' very quickly.
 i.e. DIM A$(1000,4,5) uses up 20,000 bytes of memory.

You may know that when the basic program is saved on to tape the variables and arrays also get saved on the tape. This is fine if you are going to be working within the memory range of the computer.(e.g. It's obviously not possible to hold two arrays that are 40K long in memory at the same time unless you have a memory expansion fitted). Also, the disadvantage of using this method is that if you want to alter the program at any time you will loose the information stored in the variables and arrays.

The way around this is to save the arrays only to tape separately,if you are able to write a program to do this all well and good. If not, then you will have to buy a data handling program to do this, such as the one on the MTX Utility tape!.

After all that, if you do have problems, please write to us, explaining what you are trying to do and we will do our best to help you. After all, thats what the club is all about and the more letters we get the happier we are.

Incidentally, on a more technical note, the arrays are stored after the program and more importantly, information about these arrays is stored beginning at £C000 (on a MTX 512), this information is used by Basic and therefore cannot normally be overwritten, but for those of us who are forever roaming about in Basic's Panel, do not relocate programs here, as Basic will not be very happy about it when it finds out!!!!.

Assembler Programming

Continued from last months article about VRAM mapping from Basic, we still have the Sprite Attribute and Generator tables to look at.

Well, these obviously take care of the Sprites used by the GENPAT command in Modes 4&5, it can be seen that the Sprite generator table is 1K long (see last months mag.) thus allowing room for 128 size 0 sprites(8*8)(8*128 bytes) or 32 size 1 sprites (16*16)(32*4*8 bytes).

The Sprite Attribute table is 128 bytes long and controls the 32 sprite planes, each sprite plane is controlled by 4 bytes, the format for these looks like this :
Byte
 1   This byte contains the value which is the number of pixels from the top left hand corner of the sprite to the top of the screen.
 2   Contains the value which is the number of pixels from the top left hand corner or the sprite to the left of the screen.
 3   The contents of the third byte determine which shape the sprites will be and is selected from one of the pre-defined shapes in the sprite generator table.
 4   The lower four bits of the fourth byte select the sprite colour and the most significant bit of this byte may be set to allow the sprite to 'bleed' in from the left hand side of the screen.

Knowing now a little of what a Pattern Table, Generator Table and Attribute table are, you should be able to make some sense of the Technical data on the VDP in the black manual. You should also appreciate the effects of such Basic commands as:- CTLSPR, GENPAT,ADJSPR etc..

The only other thing that we have not talked much about is the VDP's 8 Write Only Registers.(See Table 2 :VDP Registers, Pg221 of manual). These registers contain all the information necessary to form the VRAM table, Basic sets these on 'start up' to conform to Mode 2 graphics, it is possible to set up these as you wish and use them from assembler. The manual describes the VDP being used in Mode 1 & Multicolour mode, this is possible and with time and care could produce some interesting results, however all is not so simple as it is necessary to disable Basic otherwise this will corrupt your efforts. This is done by making your first assembler command DI (Disable Interrupt) and your last, on return to basic RETI (Return from Interrupt). Another way of returning to basic is to make your last line JP £0000, this being called a 'warm boot' back to Basic in computer jargon.

It is totally safe to muck about with the VDP in this way so feel free to try anything, the worst that can happen is that the computer will 'hang-up'.

The technique for setting up the VDP registers is as follows:-

All register set-ups take place via port 2 in two stages. The Z80 'D' and 'E' registers are pre loaded with the register number and the data for that register respectively, then the data is output first to port 2 followed by the register number also to port 2. It should be noted that before the second write (the register number) the most significant bit (7) must be set to '1' and bits 6,5,4 and 3 must be set to '0'. A simple program which does this would look like this :

```
; ROUTINE TO SET UP VDP REGISTERS
; DE REGISTER LOADED WITH DATA AND REGISTER ON ENTRY
  (DO NOT Type in the above lines, they are for reference only)

VDPREG: PUSH AF      ;SAVE REGISTERS
        PUSH BC
        LD A,E       ;LOAD ACCUMULATOR WITH DATA
        OUT (£02),A;OUTPUT DATA
        LD A,D       ;LOAD ACCUMULATOR WITH REGISTER NO.
        AND 7        ;SET UP CORRECT CONTROL BITS
        OR 128       ;SET MSB TO '1'
        OUT (£02),A;OUTPUT REGISTER NUMBER
        POP BC
        POP AF       ;RESTORE REGISTERS
        RET
```

Using this type of format it is possible to manipulate the VDP as you wish, remember that :

Port 2 is used for Address transfers
i.e. The registers, including the auto-incrementing address register mentioned last month.
Port 1 is used for Data transfers
i.e. Accessing the Vram.

Also, address set ups and data transfers require a certain minimum amount of time between processes, this is 11 micro seconds between address set ups and 8 micro seconds between data transfers. (Not long enough to go and make a cup of tea!!)

It is also possible to change from Text Mode to Graphics Mode with alarming speed as only two Registers have to be altered. The alterations are as follows :
(See pages 221 & 222 of black manual)
Bits M1,M2 and M3 control the Mode of operation and simply changing these using the above program will change you from Text to Graphics.

M1	M2	M3	
O	O	1	Graphics mode 2
1	O	O	Text Mode

Thats about how simple the VDP is, probing about with simple little routines will help with understanding these things even fuller.

Phil

4

```
5 LET X=0:LET Y=0:LET T=20:LET W=0:LET P=0
7 LET J=0:LET XX=0:LET YY=0:LET TT=0
10 GENPAT 1,147,60,126,255,60,60,126,126,255
11 GOSUB 600
15 CSR 8,10:INPUT "EASY=1     HARD=2     ";X
16 IF X<1 OR X>2 THEN GOTO 15
17 LET TT=5*X:LET SC=0
20 GENPAT 2,147,32,32,32,96,96,96,96,96
25 GENPAT 3,1,24,24,60,126,189,60,36,36
30 DIM A(30,23):CTLSPR 2,1
35 VS 4:CLS:PAPER 15:INK 4:CLS
37 CSR 4,2:PRINT "!    ! START"
39 CSR 18,22:PRINT "FINISH !    !"
40 CTLSPR 1,1
50 GOSUB 100
80 GOSUB 200:GOTO 80
100 FOR A=1 TO T
105 LET X=INT(RND*30)
110 LET Y=INT(RND*20)
120 IF X=0 OR Y=0 THEN GOTO 105
130 IF X<3 OR Y<4 THEN GOTO 105
140 IF A(X,Y)=1 THEN GOTO 105
150 CSR X,Y:PRINT CHR$(147);
155 LET A(X,Y)=1:NEXT A
162 SPRITE 1,1,50,171,0,0,9
165 LET T=TT:LET P=171:LET J=50
170 PAUSE 1000:RETURN
200 IF J<22 THEN GOTO 220
210 IF ASC(INKEY$)=8 THEN MVSPR 1,1,4:LET J=J-1
220 LET W=W+1
230 IF J>240 THEN GOTO 250
240 IF ASC(INKEY$)=25 THEN MVSPR 1,1,8:LET J=J+1
250 IF W=1 THEN GOTO 200
260 MVSPR 1,1,2:LET P=P-1:LET W=0
275 IF P=12 THEN GOTO 800
280 LET XX=INT(J/8):LET YY=INT(P/8)
283 LET YY=24-YY
285 IF YY=21 THEN LET YY=20
290 IF A(XX,YY)=1 THEN GOTO 500
300 GOTO 200
500 CLS:CSR 3,8
510 PRINT "YOU CRASHED IN TO A TREE";
520 PAUSE 1000:GOTO 905
600 CSR 4,3
610 PRINT "YOU MUST SKI INTO THE FINISH GATE ";
620 PRINT "     TO SCORE 100 POINTS";
625 PRINT
630 PRINT "   IF YOU MISS THE GATE YOU'RE DEAD";
640 RETURN
800 IF J<208 OR J>232 THEN GOTO 900
810 CSR 10,1
820 LET SC=SC+100
830 PRINT "SCORE   ";SC;
840 GOSUB 100
850 RETURN
900 CLS
905 CSR 3,10
910 PRINT "YOU MISSED THE GATE AND ARE    DEAD";
920 CSR 3,13:PRINT "YOU'RE FINAL SCORE IS";SC;
930 PAUSE 10000:STOP
1000 COPYRIGHT R.ADAMS & M.O.C.
```

*** ROTATION *** by Mark Cytera

Rotation is not only a graphics demonstration, but a sound demonstration too. As well as painstakingly drawing a blancmange like pattern, the machine plays a familiar tune to take the user's mind off the screen for the early stages.

The playing of the tune is quite simple. First, the computer reserves space for enough notes by using the SBUF statement. The variable FACT is a factor for changing the octaves that the tune is played in. In line 180, FACT is changed to make the frequency double. After placing all the notes in the sound buffer, they are played through whilst the picture is being drawn.

I won't bore you with the mathmatical intricacies involved in plotting the points. However, a useful device for drawing the mirror image of something is in lines 2120-2130, where A is the x co-ordinate of the axis of symmetry, Y is the Y co-ordinate and X is the distance from the axis of symmetry

```
                       R O T A T I O N
90 SBUF 120                          1000 VS 4
95 LET FACT=1                        1005 PAPER 1: INK 10: COLOUR4,13
100 RESTORE 500                      1010 CLS
110 READ FREQ,DUR                    2010 LET A=128: LET B=A*A
120 IF FREQ<>1 THEN SOUND 0,         2015 LET C=96
    FREQ*8*FACT,1000,0,-10,8*DUR,1   2020 FOR X=0 TO A-1
170 IF FREQ<>1 THEN GOTO 110         2030 LET S=X*X: LET P=SQR(B-S)
180 LET FACT=FACT/2                  2050 FOR I=-P TO P STEP 4
190 IF FACT>=.5 THEN GOTO 100        2060 LET R=SQR(S+I*I)/A
200 SOUND 0,0,0,0,0,0,1              2080 LET Q=(R-1)*SIN(24*R)
500 DATA 358,4,2,.2,358,4,379,4,     2090 LET Y=I/3+Q*C
    426,4,358,8,379,8,2,.2           2095 IF I=-P THEN LET M=Y: GOTO
510 DATA 426,4,2,.2,426,4,379,4,          2111
                                     2100 IF Y>M THEN LET M=Y
    358,4,478,4,568,8,2,.2           2101 IF Y=M THEN GOTO 2116
520 DATA 478,4,2,.2,478,4,426,4,     2105 IF Y>=M THEN GOTO 2140
    379,4,358,8,379,8,2,.2           2111 LET N=Y
530 DATA 426,4,2,.2,426,4,379,4,     2116 LET Y=C+Y
    358,4,379,8,2,.4                  2120 PLOT A-X,Y
540 DATA 358,4,2,.2,358,4,379,4,     2130 PLOT A+X,Y
    426,4,358,8,379,8,2,.2           2140 NEXT
550 DATA 426,4,2,.2,426,4,379,4,     2145 NEXT
    358,4,478,4,568,8,2,.2           3000 GOTO 3000
560 DATA 478,4,2,.2,478,4,426,4,
    379,4,358,8,379,8,2,.2
570 DATA 426,4,379,4,358,4,319,
    4,358,8,1,1
Line Numbers :
90         Reserve space for sound. 2010       Set-up constants
95         Set-up starting octave   2020-2145 Outer drawing loop
100-190    Play notes               2050-2140 Inner drawing loop
200        Stop sound               2120-2130 Plot points
500-570    Sound data               3000      Maintain screen
1000-1010  Set-up screen
```

Y O U R   L E T T E R S

Games High Scores Table

TOADO	59021	M.GELDER		BLOBBO	68042	M.GILL
NEMO	3600	M.CYTERA		OBLOIDS	60040	M.GELDER
P.PETE	33680	M.GELDER		MISS. ALPH.	27800	M.GILL
KILOPEDE	7341	M.CYTERA		GOLDMINE	3042	M.GILL
CONT RAID	10810	M.GILL		STAR COMM.	77700	M.GILL
MAXIMA	252830	M.GILL (In 2 1/2 hours play!!)				
TURBO	13300	M.GELDER				

SOFTWARE SWOPPING:-
David Lam of 105 Geary Road, Dollis Hill, London, NW10 1HS would
like to swop the following:-
MTX CHESS for MATHS or WORDS & PICTURES by CONT. SOFTWARE Ltd.
and MTX GRAPHICS GEN. for FELIX FACTORY or MAXIMA.
                    (Please contact him directly)

Your Letters 1:-Concerning embedded printer control commands in a
Neword document. Memotech have told me that this is done "by
using the commands .XQ/.XR with the appropriate HEX escape
sequence.".What they have not told me, is the format of that
sequence, I have tried everything but with no success, can anyone
help me?. Ian Phillips from Broughton in Furness.

Hints & Tips 1:- Display Ascii Code of any Key Press, the short
program below will display the ASCII code (decimal) of any key
press. Ian Phillips from Broughton in Furness.
     10 PRINT PEEK(64893)
     20 GOTO 10
               2:- The two routines below should solve any
problems with Decimal Placings. Tony Street from Gloucester.
     NUM=INT((NUM+.05)*10)/10 to correct to 1 decimal place
     NUM=INT((NUM+.005)*100)/100 for 2 decimal places
               3:- John Hodgson has a simple answer to converting
your MTX 512 into a MTX 500. All you have to do is:-
     1) POKE 0 into location £FA7A
     2) Type NEW and press the RETURN key
          (Very handy for checking out relocated programs!)
You now have a MTX 500. I found that it was safer to switch the
machine off and on again before setting the machine to a MTX 500
as this cleared out the memory. I was caught out on several
occasions when I thought my program was working on a MTX 500 but
in fact it was calling routines that were still in memory between
£4000 and £7FFF.
               4:- John Thiede has written this lovely little
subroutine which displays the variables that you have used :
1000 REM VARIABLE PRINT ROUTINE
1010 LET COUNT=0: LET ADDR=49152
1020 LET PVAL=PEEK(ADDR)
1030 IF PVAL=255 THEN GOTO 1120
1037 IF PVAL=0 THEN GOTO 1110
1040 IF PVAL>=48 AND PVAL<=90 THEN GOTO 1100
1050 IF PVAL<=192 THEN PRINT ,CHR$(PVAL-64);

```
1060 IF PVAL>192 THEN PRINT ,CHR$(PVAL-128);
1070 FOR PTWO=COUNT TO O STEP -1
1080 PRINT CHR$(PEEK(ADDR-PTWO));
1090 NEXT PTWO
1095 PRINT: LET COUNT=0: GOTO 1110
1100 LET COUNT=COUNT+1
1110 LET ADDR=ADDR+1:GOTO 1020
1120 STOP
```

- -

S C R E E N   S C R O L L   C O N T R O L

Each screen defined on the MTX has 15 bytes allocated to it in
high memory. These bytes hold information on the makeup of the
screen. Information on these bytes is scarce, but the **first**
byte of each 15 holds information on screen type (text or
graphics), cursor ON or OFF and PAGE/SCROLL mode. These are as
follows :

```
BITS :     7 6 5 4 3 2 1 0
                 :       : :
           Screen  Cursor :
           Type    ON/OFF :
                         Page/Scroll
                         Mode
```

(1)   **Screen Type**   BIT 5 : 0 - TEXT Screen
                                  1 - GRAPHICS Screen

(2)   **Cursor Control** BIT 1 : 0 - CURSOR OFF
                                  1 - CURSOR ON

(3)   **Page/Scroll**   BIT 0 : 0 - SCROLL
                                  1 - PAGE

```
SCREEN 1st BYTE
    0     FF5D    65373    EDITOR
    1     FF6C    65388    LIST
    2     FF7B    65403
    3     FF8A    65418
    4     FF99    65433    GRAPHICS
    5     FFA8    65448    TEXT
    6     FFB7    65463
    7     FFC6    65478    MESSAGE
```

To set any virtual screen to PAGE mode (BIT 0 = 1) we can check
the screen's first byte for evenness. If it is even (BIT 0 = 0)
then we just add one to it. The BASIC code for this is :

IF MOD(PEEK(<1st byte>),2)=0 THEN POKE <1st byte>,PEEK(<1st
byte>)+1
This will set any virtual screen to PAGE mode, enabling the
bottom line of the screen to be written.

COMPETITION

This month we have a program for you, there is something very strange about this program which may or may not be apparent at first sight. The competition is to let us know what is unusual about the program:-

```
10 REM PROGRAM TO CALCULATE PI BY DETERMINING THE AREA OF A
20 REM POLYGON OF "N" SIDES.
30 REM A IS 1/2 SIDE OF NEXT POLYGON
40 REM N1 IS THE NUMBER OF SIDES OF THE NEXT POLYGON
50 REM X IS THE SAGITTA
60 REM N IS THE LIMIT OF THE NUMBER OF SIDES
70 REM A1 IS THE AREA OF THE NEXT SMALL RIGHT ANGLE TRIANGLE
80 REM A2 IS THE TOTAL AREA OF THE NEXT SMALL TRIANGLE
90 REM A3 IS THE TOTAL AREA SO FAR
100 READ N
110 LET A=SQR(3)/2
120 LET N1=3
130 LET A3=A*3/2
140 LET X=1-SQR(1-A*A)
150 LET A1=X*A/2
160 LET N1=N1*2
170 LET A2=N1*A1
180 LET A3=A3+A2
190 PRINT "NO. OF SIDES OF POLYGON= ";N1
200 PRINT "PI= ";A3
210 LET A=0.5*SQR(A*A+X*X)
220 IF N>N1 THEN GOTO 140
230 PRINT "REQUESTED LIMIT";N
240 DATA 1536
250 STOP
```

I think that it may be necessary to give you a small cryptic clue to help you find the answer:-

```
Small
Profits  &        You may find that three out of four isn't  a  bad
Quick             average!!.
Returns           Try also pg. 168/169 of manual(Whats missing?)
```

The prize for this month is Felix In the Factory.

The winner of last months Drawing board competition was Mark Gill who will be receiving a dust cover in the near future.

<center>********</center>

Continued from the Editorial
8. Breakeven *** New *** Another applications program
9. Logo Drawing board *** New *** Suitable for machines with no printer.
As a reminder, to obtain a program from the library just send a S.A.E. (and a couple of spare stamps for each program to help cover the cost). Also they are available on tape, recorded one per side. £1.00 a tape (2 programs)