VOL. 4 ISSUE 4 FEBRUARY 1988

THE MEMOTECH OWNERS CLUB MAGAZINE

MEMOTECHNIQUES

PUBLISHED BY: MEMOTECH OWNERS CLUB
13 COPSE ROAD
TOWNHILL PARK
SOUTHAMPTON

CIRCA ...332

M.O.C.

VOLUME 4 ISSUE NUMBER 4

CONTENTS

o o o O O O o o o

--- A Club Facility ---

A program listing facility for those with no printer. Just send in your program on tape (or 5.25" disc) with a pre-paid envelope addressed to yourselves and the club will do the listing for you.

--- Names and Telephone Numbers. ---

i. Paul Wood for 3.5" disc copying, general info and Comms specific info.

Tel 0905 24260

ii. Alan Dobson for help with the following adventures:
Alice, The ZOO and Man From Granny
Tel 061-980-6288

Next month....

A Digital Mouse Interface !!!!!!!!!

E D I T O R I A L (February 1988)

Phil & Hazel Eyres
13 Copse Road
Townhill Park
Southampton

Before Christmas I was talking to John Grayson, he wanted to know what I thought Memotech owners wanted. From sales of the Assembley Language course it is obvious that a lot of people buy the MTX because it has a Z80 assembler built in. Also in the last six to eight months we have had a lot of new members, most with new machines from Ron Gladwin, so I suggested that perhaps a Basic Tutorial may be what is wanted. John took up the idea and has now finished it, and has it ready to ship, a review is included on page 12. We as always want to know what you want, both in the magazine and as software/hardware, perhaps if the tutorials is what is wanted, the club could make up some booklets on topics like, NODDY, LOGO, The VDP, The Sound Chip, RST10 etc.

We still have a few Diary's left if anyone would like one, we've reduced the price to £1.00 each (+P&P 25p) to clear them. Send off soon for them as they will be sent out on a first come first served basis.

I should now be in most evenings, except for the odd game of squash, so I think that it would be best to revert to having the Club Hotline between 7 and 8 pm any evening. Please where at all possible refrain from phoning after 9pm.I hope this is ok for everyone. The number to phone now is (0703) 585106, ask for Phil.

If anyone would like back issues they are available for the small remittance of 80p each. At present there are 33 back issues, 10 for volume 1, 10 for volume 2, 10 for volume 3 and 3 for volume 4.

- -

COMPUTER WORDSEARCH

Four winners to last months Wordsearch:-

Alan Clark, Exeter, Devon.
Mavis & Ken Carter, Woodlesford, Leeds.
E.Gray, Bournemouth, Dorset.
Andrew Meech, Marston Moreteyne, Bedfordshire.

Each will receive a 1988 MOC Diary.

This month a new WORDSEARCH supplied by Hazel, a copy of CHESS to the person who can find the most computer related words.

```
20 VS 4: CLS
30 DIM A(20,27)
40 FOR T=1 TO 20: FOR W=1 TO 27
50 READ A(T,W)
60 CSR W+2,T: PRINT CHR$(A(T,W)+64);
70 NEXT : NEXT
80 GOTO 80
90 DATA 20,5,19,14,14,15,4,4,25,5,19,1,6,4,26,15,6,12,7,
      8,20,16,1,2,6,9,12
100 DATA  1,21,12,1,20,17,26,4,20,10,18,23,3,2,1,3,9,9,
      10,16,11,25,5,11,1,6,24
110 DATA  2,15,1,15,1,19,15,20,15,1,15,20,15,5,3,2,15,19,
      2,1,19,9,3,19,20,5,4
120 DATA  5,19,20,20,2,3,1,2,16,2,20,1,7,13,15,5,1,20,8,
      7,25,2,5,20,17,3,1
130 DATA  1,5,16,18,20,18,3,5,5,3,15,9,2,1,13,3,4,3,9,10,
      14,4,1,15,9,10,20
140 DATA  13,7,17,5,2,5,1,1,18,4,7,10,7,4,3,1,20,1,14,13,
      20,23,8,14,22,9,1
150 DATA  23,1,15,19,13,5,18,20,1,5,3,2,16,18,9,14,20,2,
      15,16,8,23,8,14,22,11,25
160 DATA  15,13,4,14,16,14,2,4,20,20,2,3,26,20,5,15,4,3,
      3,20,5,2,15,12,14,1,11
170 DATA  20,5,3,9,4,17,5,1,15,13,1,2,18,1,20,19,7,5,4,1,
      19,1,5,19,9,20,20
180 DATA  18,19,18,15,19,18,21,3,18,15,20,3,1,5,18,15,20,
      19,18,5,9,9,23,5,1,5,4
190 DATA   19,2,14,5,1,13,16,18,17,3,4,24,4,7,8,9,10,18,
      20,3,19,18,15,13,17,2,6
200 DATA   20,1,3,18,17,1,4,6,18,20,15,16,13,5,3,20,1,20,
      2,1,5,1,18,16,18,20,18
210 DATA   10,20,2,19,20,14,21,5,1,21,3,5,17,16,1,25,3,3,
      4,19,18,15,13,20,2,1,25
220 DATA   9,16,1,9,1,14,16,4,3,22,19,10,8,18,14,13,21,12,
      15,3,6,20,1,5,1,20,8
230 DATA   4,18,18,14,3,2,1,3,8,1,18,1,3,20,5,18,19,2,1,5,
      20,3,4,13,14,16,1
240 DATA   3,17,3,20,14,17,2,15,15,13,14,9,12,8,17,16,15,
      4,3,25,1,12,16,19,9,4,24
250 DATA   1,15,9,16,10,13,14,4,16,12,15,20,20,9,14,7,2,
      13,5,17,2,25,14,15,20,19,1
260 DATA   14,18,19,17,1,2,20,5,1,20,13,13,5,13,15,18,25,
      20,1,4,9,19,3,2,14,13,6
270 DATA   5,2,5,1,14,19,16,19,15,6,20,23,1,18,5,15,4,3,
      2,8,19,1,18,3,2,13,21
280 DATA   15,20,11,5,10,1,3,1,15,2,8,3,18,1,5,19,4,18,15,
      23,19,12,5,26,1,8,6
```

FLITTER PART 6

By

Brian Clarke

****

The following is a simplified sequence of the programme as it has been produced to date. I have also cross-referenced the registers where the actions commence, hence if your programme fails to run, you should be able to identify the area where errors may have crept in.

```
4010 START:LD A,(FLAG)    Set-up variables.
4073 KEYS:LD A,0          Keyboard (Joystick) routine.
412B SETXYZ:LD A,(LL)     Calculate LX,RX, verify limits.
4151       LD A,(LD)      Calculate LY,RY, verify limits.
419A ZSET:LD A,(LZ)       Calculate LZ,RZ.
41D6 ZCOMPARE:LD A,(RZ)   Verify LZ,RZ limits.
41E9       CP B           Calculate DIFFZ to set
                          sprite SPSN/SPLN.
4214 ENDLIMIT:LD A,(LX)   Calculate ABS(DIFFX).
4226       LD A,(LY)      Calculate ABS(DIFFY).
4238       LD A,(LZ)      Calculate ABS(DIFFZ).
424A LFIRE:LD A,(LG)      Check L-gun power sufficient
                          to shoot.
4252       LD A,(LF)      Check if L-gun fired.
425A       RST 10         Draw L-shots.
426D       LD A,4         Reset L-gun power.
4277       LD A,(DIFFZ)   Check if R-ship in range.
428F       LD A,40        Set hit on R-ship flag.
4294 L630:RST 10          Undraw L-shots.
42A6 RFIRE:LD A,(RG)      Repeat 424A-42A1 for R-player.
4302 POWER:LD A,(LS)      Reset L-ship power.
430E       CP 175         Check if L-power <0 (=>175).
431B POWLA:CP 150         Check L-power limit.
4325       LD A,0         Reset hit on L-ship flag.
432A       LD A,(LP)      Check if L-power <41.
4332       LD B,A         Check if L-power + gun >40.If so,
                          add together and set gun power=0.
4344 POWER2:LD A,(RS)     Repeat 4302-4341 for R-player.
4386 L370:LD A,(LG)       Check L-gun power limit.
438E       INC A          If gun < limit, add 1.
4392       LD A,(LP)      If ship power > 0,
                          deduct 1 (for gun).
439E L380:LD A,(RG)       Repeat 4386-439B for R-player
43B6 SCRNUP:LD A,(RY)     If RY & LY outside central area
                          on screen, skip next section.
43D0       LD HL,288      Calculate whether to add or
                          subtract 1 to LZ/RZ to
                          maintain average of 144.
4407 SCRNUPC:LD A,(DIFFZ) If DIFFZ > 48, blank out
                          sprites 4 & 5.
441A SCRNUP2:LD A,(LZ)    Calculate sprite 4 & 5
                          position & colour
443B SCRNUP3:LD B,0       Calc R-sprite 2 pattern value.
4448       LD B,0         Calculate L-sprite SPLN
                          pattern value.
4484       LD A,40        Add offset to draw L & R power
                          available lines.
44A9       RST 10         Draw L & R shots available.
44F5 SCREEN:RST 10        Update screen printout
                          (sprites, power).
4574 ENDGAME:NOP          Check if L or R ship out of power,
                          or if ships have collided.
459A       JP NC,KEYS     Repeat from KEYS (Joystick input).
```

There are still quite a few silly mistakes within the programme, which I hope to show you how to correct for yourself.

Firstly, there are quite a few places where I have the routine :

```
LD A,40 (or any other value)
LD B,A
replaceable by LD B,40.
```

Also we can delete from the variable list (4056 to 4072) references to RX,RY,LX,LY,SPLN,SPSN; delete the first section of SCRNUP1 (4460) which copies these variables into registers in the SCREEN screen update section (44F5), changing the labels in the SCREEN section from e.g. SP2XPOS to RX.

If you go through the programme and make these changes, you will notice some small improvement in the speed of the programme.

There is, however, another way of improving the programme, which is by streamlining the operation sequence. The following sequence is only one of many possible ways of doing this.

```
OP1  If (FLAG)=0 CALL OP15
OP2  Keyscan routine
OP3  Check LR & LL keys. If both/neither pressed, goto OP4
     Calc new LX. If outside limits, goto OP4
     Update LX
OP4  Check RL & RR keys etc
OP5  Check LU & LD keys. If both/neither pressed, goto OP6
     Calc new LY. If within limits, update LY
     If LY<16,LD B,252:if LY<56,LD B,254:if >135,B=2:
     if >175,B=4: LZ=LZ+B. Check (& adjust) LZ limits.
     Update LZ. Calc LZ/32. Update SPLNPAT
OP6  Check RU & RD keys.
OP7  Calc DIFFX/DIFFY/DIFFZ. Set 'RANGE' flag if
     within limits.
     Set 'COLLIDE' flag if within limits.
     Calc adjusments to LZ/RZ if LY & RY >55 <135.
OP8  Check LF key. If not pressed, goto OP9
     If LG<4, goto OP9
     Draw shots. Set LG=LG-4. If 'RANGE' set,
     set 'RS' = 40. Undraw shots.
```

Continued From Previous Page

OP9 Check RF key.
OP10 Set LP(ower)=LP-LS+5. Check limits.
If LP>40,goto OP11. If LP+L6>40, reset LP. Set L6=0.
OP11 Set RP.
OP12 Redraw screen.
OP13 Goto 14 if 'COLLIDE' flag set, LP=0 or RP=0 else reset flags
OP14 RET to basic. Read LP,RP and COLLIDE to decide on result.
OP15 Set variables.

These could all be run from a master loop, e.g.
CALL A2
CALL A3
CALL A4
etc., or the programme could be written in the order shown above. Each method has its advantages, the 'master loop' allows easier debugging, the sequential programming is marginally quicker when run. Either way, the programme is more compact, and the computer is only performing the minimum number of tasks in any one loop to satisfy operational requirements.

For simplicity I would also suggest you keep the same keyboard scan routine (label KEYS to register before SETXYZ).

I won't go into the details of the logic of the programme here, that has been covered enough (I hope !).

I think the only 'new' commands introduced are :-

XOR A - eXclusive OR A to register (in this case, A, i.e. itself). XOR sets the bits to a 0 unless equivalent bits are different - as A = A, this is equal to LD A,0.

LD HL,LX - this loads the register pair HL with the location of the label LX, i.e. HL now hold the address of LX.

ADD A,(HL) - this adds the value in the address pointed to by HL to the accumulator (register A).

LD (HL),A - this loads the value in A to the address pointed to by HL.

NEG - sets A = 0 - A, i.e. negates A (to be more specific, 2's compliments A)

I have also made more use of the B, C, D, E, H & L registers to hold data temporarily.

The addiional data you will require is which BASIC lines you need from the original programme. These are :-
10 :
700 to 1490 inc :
In addition 1500 GOTO 20 :
and the basic sequence
100 LET LP=PEEK([decimal value of the LP label address])
110 LET RP=PEEK([decimal value of the RP label address])
120 IF LP=0 THEN GOTO 700 ELSE IF RP=0 THEN GOTO 750 ELSE GOTO 850:REM on the assumption that, if neither ship is out of power, the only other reason for coming out of he M/C code is that they have crashed !

A few final comments. Firstly, you must be a lazy lot, I have received information that you want the FLITTER programme(s) in the library. So you can't even be bothered to try loading it in yourself. Give me a few weeks, and I will incorporate this months update and de-bug it, and the library can have 6 copies, i.e. each months update. Secondly, and probably more important, this is not a game to while away the wee hours. It is (still, even in code), slow & bo-o-o-o-ring. That is because the intention was never to produce a superfast game, just to introduce you to machine code. However, it does utilise a lot of logic, which has enabled me to show you quite a lot of coding techniques.

Finally, if the machine crashes out for no apparent reason, as it once did for me, take a print of the programme. Disassemble the programme (e.g. by adding a basic line 0 REM TEST LINE, then remove this dummy line. If you now enter PANEL, and List the programme, where the hardcopy has
413B LD (RZ),A
the screen listing will show the address of LZ, e.g.
413B LD £4070,A
The situation I somehow managed to achieve was a 'ghost' register for the variable RZ, such that, in the above example, the screen showed
413B LD £45A7,A
thus the programme was not reading the correct variable.

The best of luck to you; it's been fun for me, but I'm glad it's over. And don't give up when your routine fails - remember that the only thing that succeeds at first try is a parrot.

Phil-> On behalf of all the members of the club I would like to thank Brian for all the effort he has put into this series of articles. --- Thank You ---

FLITTER PART 6 — THE PROGRAM

```
20 CODE

    OP1:-

OP1: LD A,(FLAG)
    CP 0
    CALL Z,OP15

    OP2:-

OP2: XOR A
    LD (LL),A
    LD (LR),A
    LD (LU),A
    LD (LD),A
    LD (LF),A
    LD (RL),A
    LD (RR),A
    LD (RU),A
    LD (RD),A
    LD (RF),A
    LD A,127
    DI
    OUT (5),A
    IN A,(5)
    LD (TEMP),A
    IN A,(6)
    EI
    BIT 0,A
    JP NZ,OP2A
    LD A,1
    LD (LF),A
OP2A:LD A,(TEMP)
    LD B,A
    LD A,1
    BIT 0,B
    JP NZ,OP2B
    LD (LL),A
OP2B:BIT 1,B
    JP NZ,OP2C
    LD (LR),A
OP2C:BIT 2,B
    JP NZ,OP2D
    LD (LU),A
OP2D:BIT 3,A
    JP NZ,OP2E
    LD (LD),A
OP2E:LD A,223
    CALL OP2J
    JP NZ,OP2F
    LD A,1
    LD (RF),A
OP2F:LD A,247
    CALL OP2J
    JP NZ,OP26
    LD A,1
    LD (RL),A
OP26:LD A,239
    CALL OP2J
    JP NZ,OP2H
    LD A,1
    LD (RR),A
OP2H:LD A,251
    CALL OP2J
    JP NZ,OP2I
    LD A,1
    LD (LU),A
OP2I:LD A,191
    CALL OP2J
    JP NZ,OP3
    LD A,1
    LD (RD),A
    JP OP3
OP2J:DI
    OUT (5),A
    IN A,(5)
    EI
    CP 127
    RET

    OP3:-

OP3: LD A,(LL)
    LD B,A
    LD A,(LR)
    SUB B
    LD HL,LX
    ADD A,(HL)
    CP 174
    JP NC,OP4
    CP 83
    JP C,OP4
    LD (HL),A

    OP4:-

OP4: LD A,(RL)
    LD B,A
    SUB B
    LD HL,RX
    ADD A,(HL)
    CP 174
    JP NC,OP5
    CP 83
    JP C,OP5

    OP5:-

OP5: LD A,(LD)
    LD B,A
    LD A,(LU)
    SUB B
    LD HL,LY
    ADD A,(HL)
    CP 7
    JP C,OP5A
    CP 183
    JP NC,OP5A
    LD (LY),A
OP5A:CP 56
    JP C,OP5B
    CP 135
    JP C,OP5C
    LD B,2
    CP 175
    JP C,OP5C
    LD B,4
    JP OP5C
OP5B:LD B,254
    CP 16
    JP C,OP5C
    LD B,252
    ADD A,B
    CP 251
    JP C,OP5D
    LD A,250
    JP OP5E
OP5D:CP 32
    JP NC,OP5E
    LD A,32
OP5E:LD (LZ),A
    LD B,0
    LD D,32
OP5F:SUB D
    INC B
    CP D
    JP NC,OP5F
    LD A,B
    LD (SPLNPAT),A

    OP6:-

OP6: LD A,(RD)
    LD B,A
    LD A,(RU)
    SUB B
    LD HL,RY
    ADD A,(HL)
    JP C,OP6A
    CP 183
    JP NC,OP6A
    LD (RY),A
OP6A CP 56
    JP C,OP6B
    CP 135
    JP C,OP6C
    LD B,2
    CP 175
    JP C,OP6C
    LD B,4
    JP OP6C
OP6B:LD B,254
    CP 16
    JP C,OP6C
    LD B,252
OP6C:LD A,(RZ)
    ADD A,B
    CP 251
    JP C,OP6D
    LD A,250
    JP OP6E
OP6D:CP 32
    JP NC,OP6E
    LD A,32
OP6E:LD (RZ),A
    LD B,0
    LD D,32
OP6F:SUB D
    INC B
    CP D
    JP NC,OP6F
    LD A,B
    LD (SP2PAT),A

    OP7:-

OP7: LD H,0
    LD L,0
    LD A,(LX)
    LD B,A
    LD A,(RX)
    SUB B
    JP NC,OP7A
    NEG
OP7A:CP 10
    JP NC,OP7B
    INC H
    CP 4
    JP NC,OP7B
```

GRAYSOFT

New release: **MTX BASIC TUTORIAL**

This book has been designed to teach the absolute novice the basic skills of programming in Basic, what the commands on the MTX do; and how to use them. This course is also meant for those programmers who would like to improve aspects of their programming. Useful routines are included in the book like FILL (for filling an area on screen), bouncing ball, true circles; and a host of helpful programming tips.

The book is well and logically set out, easy to read and follow. Many examples are given. Very well presented - professionally bound and attractive. All this for only £5.95!

ADVANCED GRAPHIC DESIGNER

Create graphic screens otherwise almost impossible to produce by normal programming. It is well presented and can do everything useful - even draws proper circles!

APPLICATIONS: Loading screens, layouts (e.g. garden), Scale drawings.

Available on cassette or disk (not 3½" system). Former price was £13.99 (cassette) and £15.85 (disk). Price still stands at only £9.99 for cassette or disk.

MTX FRUIT MACHINE

All popular features. One of those games hard to put down. Available on cassette for £4.30 (not for MTX 500 due to lack of memory); or disk (not 3½" system) for £4.99. Disk version does not have 'nudge' feature due to lack of available memory.

3D SPACE LINES

Try to beat the computer on ths puzzle game. Available on cassette for £4.30 or disk (not 3½" disks) for £4.99: however 3½" systems can use this program if saving from cassette to disk. Please specify.

MTX CARDBOX

Database to quickly store, retrieve and search for information. Up to 200 separate groups of data can be held at once in memory. This program will only work on disk systems. This package includes bound instructions with the program; the total cost is only £6.99.

If you would like the booklet I circulated a few months ago, or has not got the instructions for ADVANCED GRAPHIC DESIGNER, please let me know.

CHEQUES PAYABLE TO JOHN GRAYSON ONLY. Orders usually dispatched next day

Cambalt, Potters Heron Lane, Ampfield, Romsey, Hampshire. SO51 9BW

HARDWARE AND SOFTWARE PRICE LIST

FEBRUARY 1988

MOC
Phil Eyres
13 COPSE ROAD, TOWNHILL PARK, SOUTHAMPTON. Tel 0703 585106

Below is the list of available software for the MTX series, titles held in stock will be dispatched by return, all other titles ordered immediately and sent by return when received. All MOC titles always in stock and sent by return. Please make cheques payable to Memotech Owners Club.

Title	Price
3D TACHYON FIGHTER	£7.70
AGROVATOR	6.60
ALICE	7.70
ASTROMILLON	7.70
ASTROPAC	7.70
ATTACK OF KILLER TOMATOES	8.80
BLOBBO	7.70
BOUNCING BILL	6.60
BRIDGE	7.70
CAVES OF ORB	6.60
CHAMBEROIDS	7.70
CHESS	£10.00
COMBAT	4.40
CRYSTAL	7.70
DISASM	8.80
DOODLEBUG	6.60
DOWNSTREAM DANGER	7.70
DR FRANKIE	6.60
DRIVE THE CEE-5	£7.70
EDASM	8.80
EMERALD ISLE	7.70
ESCAPE FROM ZARKOS	£7.70
EXTENDED BASIC	8.30
F1 SIMULATOR	5.50
FATHOMS DEEP	7.70
FIG FORTH	17.50
FIG FORTH SDX	17.50
FIREHOUSE FREDDIE	7.70
FIRST LETTERS 1	10.00
FLUMMOX	7.70
GHOSTLY CASTLE	3.80
GOLDMINE	£7.70

Title	Price
GRAPHICS	6.60
HELI-MATHS	8.30
HIGHWAY ENCOUNTER	£8.80
HUNCHY	6.60
ICEBURG	6.60
JUMPING JACK FLASH	6.60
KARATE KING	7.70
KILOPEDE	7.70
KNUCKLES	8.80
LITTLE DEVILS	6.60
MISSILE COMMAND & ARCADE	6.60
MATHS 1	10.00
MAXIMA	7.70
MEMOCHEQUE	7.70
MEMOSKETCH	8.80
MEMOSKETCH SDX	10.00
MISSION ALPHATRON	6.60
MISSION OMEGA	6.60
NEMO	7.70
OBLITERATION ZONE	7.70
OBLOIDS	7.70
PAINTBOX	6.60
PAYROLL	23.61
PHAID	7.70
PONTOON & BLACKJACK	7.70
POT HOLE PETE	7.70
PURCHASE LEDGER	14.00
QOGO 2	£7.70
QUANTUM	6.60
QUAZZIA	7.70
REVERSI	8.80
ROLLA BEARING	7.70

Title	Price
RUTHLESS B.	4.00
SALES LEDGER	17.50
SALTY SAM	6.60
SEPULCRI SCELERATI	7.70
SMG	7.70
SNAPPO	7.70
T.SNOOKER	£8.80
SON OF PETE	£7.70
SUPA CODER	8.80
SUPER BIKE	6.60
SUPER MINEFIELD	7.70
SURFACE SCANNER	7.70
TAPEWORM	7.70
TARGET ZONE	7.70
THE WALL	6.60
TOADO	7.70
TURBO	7.70
USER BASIC	£10.00
USER BASIC SDX	11.00
UTILITIES SDX	£11.00
WORD & PICTURE	£10.00

MOC TITLES

REVEAL	6.00
SMG2	6.00
PUC-MAN	5.00
MOC FIG-FORTH	7.00
" " " TECH DATA	2.00
" " " TUTORIAL	7.00

This months special offers, the two following disc's of software are available, format 500K 5.25" Non-CP/M.

Disc 1	Disc 2
STAR COMMANDER	3D TACH FIGHTER
PACMAN	MISSILE COMMAND
MINEFIELD	TIME BANDITS
POTHOLE PETE	BACKGAMMON
PHAID	ARCAZIONS
CHAMBEROIDS	

With either of these disc's or any of the software marked with an "£"; 1 free MOC 1988 Diary per order!!!

Printer ribbons only £7.00 each.

CONTINUED FROM PREVIOUS PAGE

```
OP7B:LD A,(LZ)
     LD B,A
     LD A,(RZ)
     LD C,A
     SUB B
     JP NC,OP7C
     NEG
     LD D,A
     LD A,1
     LD (SPSN),A
     LD A,3
     LD (SPLN),A
     JP OP7D
OP7C:LD D,A
     LD A,3
     LD (SPSN),A
     LD A,1
     LD (SPLN),A
OP7D:LD A,D
     CP 48
     JP C,OP7E
     LD A,0
     LD (SP4COL),A
     LD (SP5COL),A
     JP OP7F
OP7E:LD A,12
     LD (SP4COL),A
     LD A,6
     LD (SP5COL),A
     LD A,B
     SUB C
     ADD A,A
     LD E,A
     ADD 95
     LD (SP4YPOS),A
     LD A,95
     SUB E
     LD (SP5YPOS),A
OP7F:LD A,D
     CP 32
     JP NC,OP76
     INC H
     CP 2
     JP NC,OP76
     INC L
OP76:LD A,(LY)
     LD D,A
     LD A,(RY)
     LD E,A
     CP 56
     JP C,OP7J
     CP 136
     JP NC,OP7J
     LD A,D
     CP 56
     JP C,OP7J
     CP 136
     JP NC,OP7J
     LD A,255
     SUB B
     ADD A,32
     SUB C
     JP C,OP7H
     INC B
     INC C
     JP OP7I
OP7H:DEC B
     DEC C
OP7I:LD A,B
     LD (LZ),A
     LD A,C
     LD (RZ),A
OP7J:LD A,D
     SUB E
     JP NC,OP7K
     NEG
OP7K:CP 8
     JP NC,OP7L
     INC H
     CP 2
     JP NC,OP7L
     INC L
OP7L:LD A,H
     LD (RANGE),A
     LD A,L
     LD (COLLIDE),A

     OP8:-

OP8: LD A,(LF)
     CP 1
     JP NZ,OP9
     LD A,(LG)
     CP 4
     JP C,OP9
     RST 10
     DB 100
     DB 164,27,65,3,1
     DB 165,2,29,40,29,157
     DB 133,2,32,40,32,157
     LD A,(LG)
     SUB 4
     LD (LG),A
     LD A,(RANGE)
     CP 3
     JP C,OP8A
     LD A,40
     LD (RS),A
OP8A:RST 10
     DB 165,2,29,40,29,157
     DB 165,2,32,40,32,157
     DB 132,27,65,3,0

     OP9:-

OP9: LD A,(RF)
     CP 1
     JP NZ,OP10
     LD A,(RG)
     CP 4
     JP C,OP10
     RST 10
     DB 101
     DB 164,27,65,3,1
     DB 165,2,29,40,29,157
     DB 133,2,32,40,32,157
     LD A,(RG)
     SUB 4
     LD (RG),A
     LD A,(RANGE)
     CP 3
     JP C,OP9A
     LD A,40
     LD (LS),A
OP9A:RST 10
     DB 165,2,29,40,29,157
     DB 165,2,32,40,32,157
     DB 132,27,65,3,0

     OP10:-

OP10:LD A,(LS)
     LD B,A
     LD A,(LG)
     LD C,A
     LD A,(LP)
     LD D,A
     SUB B
     JP NC,OP10A
     LD C,0
     LD D,0
     JP OP10D
OP10A:ADD A,5
     CP 150
     JP C,OP10B
     LD A,150
OP10B:LD D,A
     CP 41
     JP NC,OP10C
     ADD A,C
     CP 41
     JP C,OP10C
     LD D,A
     LD C,0
     JP OP10D
OP10C:LD A,C
     CP 12
     JP NC,OP10D
     INC C
     DEC D
OP10D:LD A,0
     LD (LS),A
     LD A,C
     LD (LG),A
     LD A,D
     LD (LP),A
     ADD A,40
     LD (LLINE1),A
     LD (LLINE2),A
     INC A
     LD (LLINE3),A
     LD (LLINE4),A

     OP11:-

OP11:LD A,(LS)
     LD B,A
     LD A,(RG)
     LD C,A
     LD A,(RP)
     LD D,A
     SUB B
     JP NC,OP11A
     LD C,0
     LD D,0
     JP OP11D
OP11A:ADD A,5
     CP 150
     JP C,OP11B
     LD A,150
OP11B:LD D,A
     CP 41
     JP NC,OP11C
     ADD A,C
     CP 41
     JP C,OP11C
     LD D,A
     LD C,0
     JP OP11D
```

CONTINUED FROM PREVIOUS PAGE

```
OP11C:LD A,C
      CP 12
      JP NC,OP11D
      INC C
      DEC D
OP11D:LD A,0
      LD (LS),A
      LD A,C
      LD (RG),A
      LD A,D
      LD (RP),A
      ADD A,40
      LD (RLINE1),A
      LD (RLINE2),A
      INC A
      LD (RLINE3),A
      LD (RLINE4),A

      OP12:-

      RST 10
      DB 100
      DB 131,3,5,22
      LD A,(LG)
      CP 4
      JP NC,OP12A
      CALL OP12D
      JP OP12B
OP12A:CP 8
      CALL C,OP12E
      CALL NC,OP12F
OP12B:RST 10
      DB 101
      DB 131,3,2,22
      LD A,(RG)
      CP 4
      JP NC,OP12C
      CALL OP12D
      JP OP126
OP12C:CP 8
      CALL C,OP12E
      CALL NC,OP12F
      JP OP126
OP12D:RST 10
      DB 132,"   "
      RET
OP12E:RST 10
      DB 130," "
      RET
OP12F:RST 10
      DB 130,"  "
      RET
OP126:RST 10
      DB 170,18,5,10,32,0
SP5YPOS:DB 0,0,0,0,
SP5COL:DB 0
      DB 170,18,4,10,224,0
SP4YPOS:DB 0,0,0,0
SP4COL:DB 0
      DB 170,18,2
SP2PAT:DB 0
RX:   DB 0,0
RY:   DB 0,0,0,0,6
      DB 170,18
SPLN:DB 0
SPLNPAT:DB 0
LX:   DB 0,0
LY:   DB 0,0,0,0,12
      DB 170,18
SPSN:DB 0,10,120,0,1,0,0,0,0
      DB 100
      DB 165,2,61,40,61
LLINE1:DB 0
      DB 165,2,60,40,60
LLINE2:DB 0
      DB 164,27,65,2,1
      DB 165,2,60
LLINE3:DB 0,60,191
      DB 165,2,61
LLINE4:DB 0,61,191
      DB 164,27,65,2,0
      DB 101
      DB 165,2,4,40,4
RLINE1:DB 0
      DB 165,2,3,40,3
RLINE2:DB 0
      DB 164,27,65,2,1
      DB 165,2,4
RLINE3:DB 0,4,191
      DB 165,2,3
RLINE4:DB 0,3,191
      DB 132,27,65,2,0

      OP13:-

      LD A,(COLLIDE)
      CP 3
      JP Z,OP14
      LD A,(LP)
      CP 0
      JP Z,OP14
      LD A,(RP)
      JP Z,OP14
      JP OP1

      OP14:-

OP14:LD A,0
      LD (FLAG),A
      RET

      OP15:-

OP15:LD A,148
      LD (RX),A
      LD A,128
      LD (LZ),A
      LD (RZ),A
      LD A,108
      LD (LX),A
      LD A,95
      LD (LY),A
      LD (RY),A
      LD A,3
      LD (SPLN),A
      INC A
      LD (SPLNPAT),A
      LD (SP2PAT),A
      XOR A
      LD (LG),A
      LD (RG),A
      LD (LP),A
      LD (RP),A
      LD A,1
      LD (SPSN),A
      LD (FLAG),A
      RET
FLAG:DB 0
LG:   DB 0
RG:   DB 0
LP:   DB 0
RP:   DB 0
LL:   DB 0
LR:   DB 0
LU:   DB 0
LD:   DB 0
LF:   DB 0
RL:   DB 0
RR:   DB 0
RU:   DB 0
RD:   DB 0
RF:   DB 0
COLLIDE:DB 0
RANGE:DB 0
TEMP:DB 0
```

INTERFACING PROJECTS

Now summers just round the corner?? why not try your hand at micro electronics. Infact what better way to start than with an MOC D.I.Y. kit. Everything you need is supplied, except a soldering iron, wire cutters and of course a few hours of your time!!. So why not order now.

Interface price list

A full set of components and instructions for the LED kit -->£6.95
A full set of components and instructions for the Speech Synthasiser kit ->£18.00
Connecting cable for the internal port (needed for projects) -->£4.50
All prices are fully inclusive. Please allow 14 days for delivery.

BASIC ROUTINES

BY

John Grayson

A few months ago John gave me several interesting routines, at last I have found a place to put them in the magazine. All the Basic listings produce pretty curves of some description or another, while the assembler you might find more useful elsewhere.

```
1 REM
2 REM (BY JOHN GRAYSON, GRAYsoft)
3 REM
10 VS 4: CLS
20 FOR J=100 TO 25 STEP -5
25 REM
30 FOR A=44.7 TO 49.48 STEP 0.0085
40 PLOT 128+(95*SIN(A/0.75)*COS(A/J)),96+(95*COS(A))
50 NEXT A: NEXT J
100 GOTO 100
```

```
1 REM
2 REM (BY JOHN GRAYSON, GRAYsoft)
3 REM
10 VS 4: CLS
20 LET SIZE=95
25 REM
30 FOR A=0 TO 62.73 STEP 0.01
40 PLOT 128+(SIZE*SIN(A)*COS(A/40)),96+(SIZE*COS(A))
50 NEXT A
100 GOTO 100
```

```
1 REM
2 REM (BY JOHN GRAYSON, GRAYsoft)
3 REM
10 VS 4: CLS
20 FOR A=0 TO 97.2 STEP .01
25 REM
30 PLOT 128+(0.87*A*SIN(A)),96+(A*COS(A)): NEXT A
40 LET SI=((S*P)*2): LET SIZE=(S/SI)*S
100 GOTO 100
```

```
1 REM
2 REM (BY JOHN GRAYSON, GRAYsoft)
3 REM
10 VS 4: PAPER 1: CLS : PAPER 1: INK 4: COLOUR 4,1
20 REM INPUT SIZE,COMPRESSION
25 REM
30 CSR 1,0: INPUT S,P
40 LET SI=((S*P)*2): LET SIZE=(S/SI)*S
100 FOR A=0 TO 2*PI STEP .01:
    PLOT 128+(SIZE*SIN(A)),96+(SIZE*P*COS(A)):
    NEXT A: CSR 0,0: PRINT "          ": GOTO 30
```

```
0 REM (PICTURE 1)
1 REM
2 REM (BY JOHN GRAYSON, GRAYsoft)
3 REM
10 VS 4: CLS
20 LET SIZE=95
40 FOR A=0 TO 15.4 STEP .009
43 LET HOR=((SIZE*SIN(A*1.1)+COS(A/40)))-1
47 LET VERT=96+((SIZE*COS(A)))
50 PLOT 128+HOR,VERT: PLOT 128-HOR,VERT: NEXT A
100 GOTO 100
```

```
1 REM
2 REM (BY JOHN GRAYSON, GRAYsoft)
3 REM
10 VS 4: PAPER 1: CLS : PAPER 1: INK 4: COLOUR 4,6
20 LET SIZE=95
30 FOR A=0 TO 125.7 STEP .01
40 PLOT 128+(SIZE*SIN(A)),96+(SIZE*COS(A)*SIN(A*.95))
50 NEXT A
100 GOTO 100
```

```
10 CODE
4007          JP START
400A  CHRSEND:DB "HERE IT IS£" ;"HERE IT IS" will be
                                displayed.
4015 START:   LD HL,CHRSEND    ;£" signs end of display
4018 LOOP:    LD A,(HL)        ;store each character of
                                above sequentially into
                                a register.
4019          CP "£"
401B          JP Z,END         ;if finished - ie the next
                                character is "£", end.
401E          LD B,A
401F          CALL £0CE3       ;send to printer.
                                to screen: CALL £CAB
4022          INC HL
4023          JP LOOP
4026 END:     LD A,138         ;138 is ASCII for a
4028          LD B,A           ;LINE FEED
4029          CALL £0CE3       ;send to printer. If you
                                are not using a printer,
402C          RET              ;don't bother with this bit.
```

Sprite Collision Detection

```
IN A(2)
AND 32
```

If register A then has a value of 32, then two or more Sprites have collided.

CONDITIONAL MERGE PRINTING WITH NEWWORD.
BY GEOFF GARDINER

Being a lazy and inaccurate person, I hate having to type names and addresses on letters, so I wondered if I could automate the procedure by having a data file of names and addresses of people I write to often and use the conditional merge print procedure to incorporate them. In my version of Newword, conditional merge printing seems to have been at the experimental stage and the instructions in the supplement to my manual are not wholly reliable and therefore some experimentation was necessary. I now have a workable procedure.

I keep on my work disc a file that is my letter heading. In the top right is my name and address, set out with right margin flush. Top left there is my telephone number. Unconventionally I insert the date on the left below the telephone number as I don't want anything on the right hand end of the lines that contain the variables that will be replaced automatically by the name and address of the addressee. These variables are against the left hand margin on the line below my address and preceding the salutation, thus:-

Tel 0565 53544

From Geoffrey W. Gardiner
3 Molly Potts Close
Knutsford
Cheshire
WA16 8QT

(date here)

&2&
&3&
&4&
&5/0&
&6/0&
&7/0&
&8/0&

Dear Phil,

******

That is how I would start a letter to Phil Eyres. There are always at least three lines for a name and address but I have provided for four more. The /0 instructs the program to omit the line if there is no data to fill the variable.

Why have I not used the variable &1&? That variable will be the initials of the correspondent to save typing his name out in full at a later stage in the procedure, as you will see.

I have a non-document called ADDRESS.DTA that contains all the names and addresses. Each line has 8 variables and has to have 8 even if some are blanks. Phil's line reads as follows:-

PE,Phil Eyres,Memotech Owners Club,23 Denmead Road,Harefield,Southampton,SO2 5GS,

Each variable is terminated by a comma (which will not be printed) except the last variable on the line which is terminated by the carriage return that ends the line. So 7 commas and a CR provides for 8 variables. One must use a non-document without a carriage return in any line, even, if the line goes way off screen.

Now my letter to Phil must be titled PE.L, and it must be on drive F:, my silicon disc, when I print out. This is because I am going to merge print by making use of a permanent file I keep on my work disc called MERGE.DOC. This is MERGE.DOC:-

```
.av ADDRESSEE
.df address.dta
.rv 1,2,3,4,5,6,7,8
.if &1& = &ADDRESSEE&
.fi f:&ADDRESSEE&.L
.ei
```

(I have tabbed the dot commands so that they will print. In the actual file the dots are of course hard up against the left margin.)

****

When I press M and respond to the menu's question with MERGE.DOC, the screen comes up with the question ADDRESSEE? Now you see why my first variable is not the addressee's full name, for in response to this query all I have to do is to type only PE, and the computer will read the data file till it finds a first variable PE. Then it will print my letter to Phil, which is the file called PE.L on drive F:, and insert his full name and address in the place of variables &2&, &3&, &4&, &5&, &6&, &7&. It will omit the line for the eighth variable which is blank.

A lazy person's dream, isn't it? Of course it took about two full days work to find out how to do it. I tried to include messages on the screen telling me exactly what to do, (using the .dm command) but without success. The procedure set out on page 178 of the DIY did not seem to do what I wanted.

By the way, if you conditionally merge print a letter you should end the letter with a .pa otherwise, for no obvious reason, the computer will add on to the end of the letter ".ei" , and sometimes other bits of the instructions. Of course .pa will make it try to print a non-existent additonal page, but that is a minor mishap. Perhaps this defect has been cleaned up in later editions of Newword.

I can also use my data file to address the envelope, but I found problems in achieving this. Eventually I had to do it by using two files, ENVADDR.IF and ADDRESS.ENV.

The first is as follows:-

```
.av ADDRESSEE?
.df address.dta
.rv 1,2,3,4,5,6,7,8
.if &1& = &ADDRESSEE&
.fi ADDRESS.ENV
.ei
```

and the ADDRESS.ENV is:-

```
.pl25
.po15
.mt4
.op
&2&
&3&
&4&
&5/0&
&6/0&
&7/0&
&8/0&
```

These instructions position the address roughly correctly on an envelope used for A4 stationery when inserted in my printer which is a Brother typewriter. They would not work with a different printer, or with an installation of Newword different from mine, but I hope the example gives an indication of how to set up an envelope addressing file.

MTX Basic Tutorial

Available From:- Graysoft 'Cambalt' Potters Heron Lane, Ampfield, Romsey, SO51 9BW.
Price:- £5.95

Over the past 1 - 1 1/2 years the supply of MTX books has all but dried up. In an effort to bridge this gap to some degree, John Grayson, has produced The MTX BASIC TUTORIAL.

This 36 page ring bound manual has been very well laid out and has a very professional 'learned' content. The book is aimed squarely at the Basic MTX 500/512 and is for anyone who wants to learn basic.

The tutorial starts you off with an introduction and then some very simple Basic programming. Reading on you are led smoothly into the 'deeper end', with useful pointers and hints all the way. In the latter part of the manual several 'universal' routines are included for your future use.

Conclusion
John has made Basic programming look so simple in this book, his routines are neat, simple and easy to follow. If you are new to the MTX this manual is well worth the cash!

MTX CARDBOX

Available From:- Graysoft 'Cambalt' Potters Heron Lane, Ampfield, Romsey, SO51 9BW.
Price:- £6.99

This information retrieval and manipulation system is written by Alan Hamilton and updated/marketed by John Grayson. It is intended for use only on disc systems (does not require a CP/M system!). The program is supplied on tape, which is a very thoughtful way of getting around the multiple disc formats that are available. Your first job, is to load the program from cassette onto a formatted disc, with the aid of the instruction manual.

Being that the program is disc based there is no problem with saving the data separately, and infact the programs strong point, if any, is its disc handling; you can do directory listings, rename files, delete files, etc.

The program is a it suggests, a 'flat file' cardbox type database, which can handle up to 200 records, you can do the normal Add/Change/Delete, and also Finds which allow you to do selective enquiries on your database; output can be to either screen or printer.

Conclusion
The program is rather short of features to make it a useful database, anyone who seriously needs a database should go for something more expensive and capable.

MTX Tape To Disc Conversion Booklet

Available From:- AFW Software 20 Cambridge Road, Whitehaven, Cumbria.
Price:- £6.11

This is a 37 page 'ring comb binder manual' printed using a laser printer, it offers the following conversions as examples:-

Games Converted:-
Toado, Kilopede, Qogo, Quazzia, Draughts, Firehouse Freddie, Sepulcri, Agrovator, Murder at the Manor.

The first four programs have worked examples to convert them when running under a normal FDX operating system and Basic. The others require the use of the V-ROM available from Ron Gladwin in order that you have available a 64K basic system. (If you have an SDX system you should be ok!).

The booklet makes good use of assembler and the Front Panel, so if your interested in that and/or transfering your games to disc, then perhaps this booklet is for you. AFW Software claim that after using their manual to convert some games, you should be able with time and patience, to convert most programs. The booklet has plenty of useful routines and explains many of the differences between the different MTX disc systems available today.

Conclusion
A useful booklet if you are ready to learn about assembler, the Front Panel and the operating system. Not bad value for money, especially since there is very little else about.

Please note that all of these new products are only available from the stated suppliers and not from Memotech Owners Club.

P R O G R A M L I B R A R Y

12 Roebank Road
Beith.
Ayrshire
KA15 2DX
Tel:05055 2491 (after 6pm)

Lots of news this month, so hold tight and here goes:

Firstly, I have several new additions to announce and the return of some "old favourites" that got lost in the re-indexing:

67. PERPETUAL CALENDAR
This program is a compendium of routines which allow you to find the number of days between any two dates from 1759 to whenever! What's more, you can find the day on which Easter Sunday falls of any year, and as if that wasn't enough, the program also incorporates a very good biorythms section.

CA11 Elements
A welcome return to the library for one of the best programs in the library. The program allows the comparisons of elements in the periodic table and has a very nice display of the table.

CA12 Mkbook
Another welcome return to this program which utilises the RAMDisc program (CA08 from the library) and allows complete control over test marks and total marks for nearly 200 pupils. The screen display is very good and all in all, very friendly.

CA13 Optics
This is another classic education package which was misplaced in the changeover. It uses the Memotech's graphics very well, showing how light can be mixed and the effects on a screen of different light colours. Good stuff.

More additions to the Turbo Pascal disc (CPM4) with GRAPHCOS, SWAPPER, DIAMOND, DETDAY & ROOTS. DETDAY is a program written by me on an Amstrad PC and transferred using a compatibility program to CP/M which allows the user to find the day on which a particular date falls. ROOTS is a program, again written by me, which solves quadratic equations. e.g. what value of x satisfies this equation: x2 + 2x + 5 = 0. Both are available as with the rest of the programs on the Turbo disc as .COM files so that those without Turbo can use them.

The Library Listings make a return to the library with all the original ones and the new updated System Variables document. All are available free of charge, just supply an A4 envelope and postage to cover please.

LL01 - System Variables	LL02 - VDP Chip explained
LL03 - NewWord ROM Review	LL04 - RST10 commands
LL05 - Undcmntd NewWord Cmmnds	LL06 - CP/M Programming
LL07 - Assembler course	LL08 - Pascal course
LL09 - Intro. to CP/M	

The long awaited Software Reviews book is now available from either myself or from Phil. Due to the size of it (86K on NewWord file) we have to impose a charge of £2 per copy (inc. postage). Briefly, it is a book of reviews of software and hardware which our reviewers (you) have written to allow all those who are not sure what software is available or what it is like to see before they buy.

One small point, when disc members are asking for discs, besides specifying the drive capacity they run on, can they also state whether (or not) they are using CP/M or on BASIC system.

Don't forget that Paul Wood is in charge of all copying on 3.5" format. If you want something on 3.5" or have something to submit, send it to him. Otherwise, all other library mail should be sent to me.

And finally, if all these new additions are becoming confusing, I can supply new updated catalogues & library lists free of charge.

Paul Wood's address is:
12 Bishops Avenue,
Worcester,
Worcs
WR3 8XA

Please make all cheques/POs payable to MOC

Best Wishes - Alan Hamilton