VOL 5 April 1989 ISSUE 3

MEMOTECHNIQUES

MEMOTECH OWNERS' CLUB

mē′mō *n.* (*pl.* ~s). (colloq.) memorandum. [abbr.]
mĕ′moir (-wâr) *n.* record of events; history written from personal knowledge or special sources of information· (esp. in *pl.*) (auto)biography; essay on
learn
~IST
fem.
mĕm
(as f
mĕ′m
forg
~AB
tĕchni′que (-knē′k) *n.* mode of artistic execution in music, painting, etc.; mechanical skill in art; means of achieving one's purpose, esp. skilfully. [F, as TECHNIC]
tĕchnŏ′cracy (-kn-) *n.* government or control of society or industry by technical experts; hence **tĕ′chnocrat** (-kn-) *n.*, advocate of this, **tĕchnocră′tic** (-kn-) *a.* [f. Gk *tekhnē* art + -o- + -CRACY]
tĕchnŏ′logy (-k-) *n.* (science of) practical or industrial art(s); ethnological study of development of such arts; application of science; hence **tĕchnolo′gical** (-kn-) *a.*, ~IST (3) *n.* [f. Gk *tekhnologia* systematic treatment (*tekhnē* art; see -LOGY)]

13, Copse Road,
Townhill Park,
Southampton
SO2 2GY

CONTENTS

COVERS

If you have an interesting cover for the magazine or an interesting picture for the cover then please send it to us. The cover can be of anything but it must be mostly white, large areas of black do not duplicate very well.

Subscriptions

THE MOC Magazine is available only by subscription:-

U.K.	£12.00
European	£15.00
BFPO NO.s	£12.00
Outside Europe	
Region A	£16.00
Region B	£17.50
Region C	£18.00

How To Submit Articles

We prefer all articles on disc or tape, but very small pieces are OK on paper. Please put your name and address on any disc's and tapes. A return address label would be appreciated if you have such things. If you are sending any "camera ready" artwork please ensure it is not folded and use black ink.

Advertising rates

Private ads cost £2.00

Commercial rates

Quarter Page £7.50 Half Page £15.00

Telephone contacts

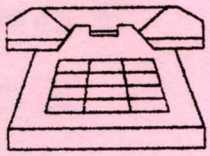
Please only ring at sensible times!

Phil Eyres 0703 585106

(Ansa Machine when not available)

Alan Hamilton 05055 2491

Paul Wood 0905 24260



EDITORIAL (April 1989)

Phil & Hazel Siobhan Eyres
13 Copse Road
Townhill Park
Southampton

THE CLUB

It has been some time now since the club's objectives/principles have been explained, so here they are again:-

1. Our primary objective is to link MTX owners via the Memotechniques magazine.

2. We are part-time. We hold down full-time day jobs and we run the club as a hobby. Sometimes therefore things do take time.

3. We aim to keep the MTX alive.

4. All funds are used to support the MTX and Club.

5. We are not a business, and should not be treated as such.

UK Home Computers

We have had several phone calls this month stating that UK Home Computers are no longer trading. It would appear that he has no longer the ability to get hold of MTX equipment so has moved on. I think most people have bought from him on 3 month warranty, so we should not have any warranty problems, as he has not had MTX kit for some time.

MICRO SHOW

There is an Alternative Micro show in November, on the 11th and 12th to be precise, in Stafford. I will be going, maybe for one day or two if there seems to be a lot of interest. So pencil this date in your diary.

NEW TUTORIAL

Alan Wilson has held back on the launch of his new tutorial, due to having all the agro of moving to a new house. An advert and review will be in the next issue.

ARTICLES

The Music programme has somewhat taken over this issue, although there is a really good article on Newword and a little snippet for 3.5" disc system users which should cheer them up. System Variables are explained in details by John Hodgson and there are two reviews.

COMPETITION

The competition from last month is held over to this month in order to give you more time.

The Club Hotline is between 7 and 8 pm any evening. During the day and after 8.00pm a club answer phone takes over. 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 40 back issues, 10 for volume 1, 10 for volume 2, 10 for volume 3 and 10 for volume 4.

Phil Eyres

Music Programme V.01.0

RackleBrain Software
By Brian Clarke
Part II

This is the second of three parts, which will make up the Music programme. It is rather long on code and explanations of what is happening, I will therefore not waste time here.

1

Kick off auto-save for auto-run. This can be avoided by by the direct commands CLEAR and GOTO 50000

1000

Play chords. Subroutine. The speed of the tune can be adjusted whilst this is running, but not interactively.

2000

Read existing data in cell (Current Chord, Channel No.). Subroutine. See the REM statements from lines 40000 for an explanation of the purposes of the variables used. If this routine is run (i.e. there is data in the new 'cell') the screen information is updated.

3000

Store & display data. Subroutine. Enters data into the memory and displays the data in centre screen.

4000

Refresh VS4. Subroutine. Required to move to right or left of existing display range, or when loading a new tune from disc.

9000

Auto chord increment. This routine is entered when 'AUTO- SAVE' is enabled (see 23000). It automatically increments the current chord +1 after RET or a DIRECT ENTRY key has been pressed (but NOT after HOME has been pressed). If data is existing in the new 'cell', executes 2000.

10000

Keyboard Entry. Main routine. Directs the programme to various subroutines depending upon keyed input. Note there is an option (press ESC) which is not on the menu (line 27000).

11000

Direct Entry. When one of the associated (highlighted) keys is pressed, (a) the note sounds, (b) the note letter in the octave colour with the volume and duration is displayed on the middle screen, (c) The key pressed is highlighted on the top left screen. If the AUTOPLAY is selected, the note continues to play. If the AUTOSTORE is selected, the current chord increments. If there is data in the existing cell, this is updated.

12000

HOME/RETURN keys. Home plays the current highlighted note in the current highlighted chord for approx. the current speed/duration. Autoplay leaves the note on. Autosave has no effect. Return also plays the note, but enters the data. Autoplay leaves the note on. Autosave increments current chord.

13000

NULL NOTE. This enables a previously entered note to be removed. It also shuts off the sound if Autoplay on. Can be emulated by resetting volume to 0.

14000

CHANGE CURRENT NOTE. Left/Right arrow keys cycle through the notes in the octave, then increments or decrements the octave. Autoplay plays the notes as selected. Autosave has no effect. If there is data in the existing cell, this is changed to the last note played. This routine (like most) uses a FOR-NEXT loop

FOR N=1 TO 1

(IF INPUT AS REQUIRED THEN LET N=N-1)

(PERFORM REQUIRED ACTIONS)

NEXT N

This cycles the loop for as long as any one of the required keys is pressed, and avoids the delays inherrent in the routine

(IF INPUT AS REQUIRED THEN SET FLAG)

(PERFORM REQUIRED ACTIONS)

(IF FLAG NOT SET THEN GOTO START OF LOOP)

where the BASIC has to search through the programme to find the 'goto' line number.

14500

Change CHANNEL 3 note. Uses left/right arrow keys to sweep from 1 to 8 (or 8 to 1). Autoplay plays the current note. Autosave has no effect. If there is data in the existing cell, this is changed to the last note played.

15000

Change Octave. Up/down arrow keys sweeps the octaves on current note. Autoplay leaves note on. Autosave has no effect. If there is data in existing cell, this is changed to the last octave played.

16000

Change chord no. F2 moves down & F6 up. If you go beyond the screen range, the screen is renewed. Autoplay plays notes which exist, playing final note continuousely. Autosave has no effect.

17000

Change volume. '' or ',' reduces volume & '' or '.' increases volume. If there is data in cell, it is updated. Autoplay leaves volume as last defined. Autosave has no effect.

18000

Change Channel. F1 cycles from channel 0 to channel 3, marking existing channel as ''. Autoplay plays note in current channel if there is one, if not, no change. Autoplay has no effect.

19000

Change duration. Alters duration of note (F3 reduces & F7 increases). If there is data in cell, it is updated. Autoplay & Autosave have no effect.

20000

Copy routine. Choice of -

a)Copy one or all channels,

b)Copy forwards or backwards,

c)Copy before or after (or into) source range.

However, there are certain rules which limit the routine.

1)Copy forwards (i.e. from chord 8 to 15 into ... is forwards, but from chord 15 to 8 into .... is backwards) can be copied into any chord (e.g. 1 or 11 or 16). If the range being copied into would extend beyond the available memory, this is highlighted, and the copy will truncate to the last available chord.

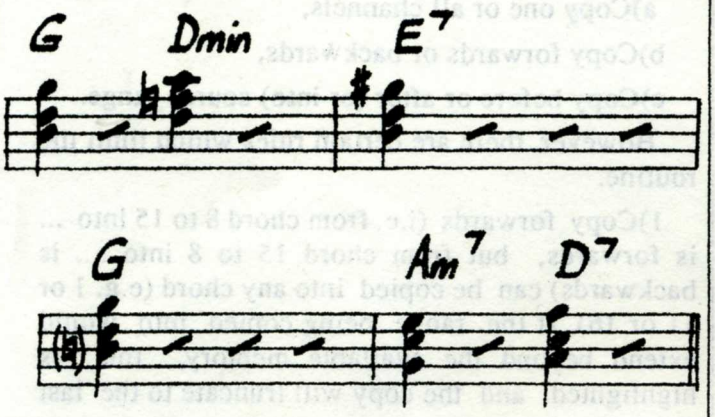
2)Copy backwards MUST be arranged so that the copy to range is outside of the copy from range. (The system will not allow anything else.)

3)The system prompt for F2,F6,SF2 & SF6 keys, but does not tell you that these are - F2 chord - F6 Chord + SF2 Goto last chord for current channel SF6 Goto any chord (enter a number) Enter at each stage to take you to the next prompt. After all inputs, the system prompts for an OK, so you can quit. The routine can take some considerable time. Note that a backward copy e.g. from 15 to 8 into 16 will make chord 16=15, chord 17=14, chord 18=13 etc.

...And this is the program!!!!
Final part next month

```
0 GOTO 60000
1 REM *****
995 REM ***** PLAY 8/ALL CHORDS
1000 FOR Y=C1 TO C2: FOR Z=0 TO 3: SOUND
Z,TN(Z+1,Y),TN(Z+5,Y): NEXT Z: PAUSE
SP/TN(9,Y)
1010 IF ASC(INKEY$)=138 OR
ASC(INKEY$)=142 THEN GOSUB 24000
1020 NEXT Y: RETURN
1995 REM ***** READ EXTANT DATA IN CELL
2000 VS 3: INK 1: PAPER 12: LET
DU=TN(9,CC): CSR 8,4: PRINT NU$(DU+1);:
LET VO=TN(CH+5,CC): CSR 2,4: PRINT
NU$(VO+1);
2010 IF CH=3 THEN LET N3=TN(4,CC): LET
FR=N3: CSR 8,3: PRINT N3;: RETURN
2020 FOR Y=1 TO 12: IF
DE$(Y,3)TN$(CH+1,CC) THEN GOTO 2040
ELSE IF NC=Y THEN LET Y=12: GOTO 2040
2030 CSR NC+5,NP(NC)+1: PRINT DE$(NC,3);:
PAPER 6: LET NC=Y: CSR NC+5,NP(NC)+1:
PRINT DE$(NC,3);: LET Y=12
```

```
2040 NEXT Y
2050 IF VAL(TN$(CH+5,CC))=OC THEN GOTO
2070 ELSE PAPER 15: INK CO(OC,1): CSR
OC*2+4,0: PRINT OC;: LET
OC=VAL(TN$(CH+5,CC))
2060 PAPER CO(OC,1): INK CO(OC,2): CSR
OC*2+4,0: PRINT OC;
2070 LET FR=NF(NC)/OF(OC): IF OC=OL THEN
RETURN ELSE LET OL=OC: GOTO 43210:
RETURN
2995 REM ***** STORE/DISPLAY DATA
3000 LET TN(CH+1,CC)=INT(FR): LET
TN(CH+5,CC)=VO: LET TN(9,CC)=DU: LET
```




```
TN$(CH+9,CC)=NU$(VO+1): LET
TN$(13,CC)=NU$(DU+1)
3010 IF CH<3 THEN LET
TN$(CH+1,CC)=DE$(NC,3): LET
TN$(CH+5,CC)=CHR$(OC+48) ELSE LET
TN$(CH+1,CC)=NU$(N3+1)
3020 VS 4: INK CO(OC,1): IF CH=3 THEN INK 1
3030 CSR CC+3-CB,CH*2+1: PRINT
TN$(CH+1,CC);: CSR CC+3-CB,CH*2+2: PRINT
TN$(CH+9,CC);
3040 IF CC>QT(CH+1) THEN LET
QT(CH+1)=CC: IF CC>QT(5) THEN LET
QT(5)=CC
3050 INK 1: CSR CC+3-CB,9: PRINT
TN$(13,CC);: RETURN
3995 REM ***** RE-DISPLAY TUNE DATA (VS4)
4000 LET CC=CR: FOR Y=0 TO 3: SOUND
Y,0,0: NEXT Y: IF CCCT THEN LET CB=CC-8
ELSE LET CB=CC-24
4010 IF CB<1 THEN LET CB=1 ELSE IF
CB+27>CM THEN LET CB=CM-27
4020 LET CT=CB+27: GOSUB 43400
4030 LET Z=CT: IF Z>QT(5) THEN LET
Z=QT(5): IF Z<CB HEN LET Z=CB
4040 FOR Y=CB TO Z: FOR M=0 TO 3: IF
YQT(M+1) THEN GOTO 4070
4050 IF TN(M+1,Y)=0 THEN GOTO 4070 ELSE
IF M=3 THEN INK 1: GOTO 4060 ELSE INK
CO(VAL(TN$(M+5,Y)),1)
4060 CSR Y+3-CB,M*2+1: PRINT TN$(M+1,Y);:
CSR Y+3-CB,M*2+2: PRINT TN$(M+9,Y);
4070 NEXT M: IF Y<=QT(5) THEN INK 1: CSR
Y+3-CB,9: PRINT TN$(13,Y);
4080 NEXT Y: IF TN(CH+1,CC)<>0 THEN
GOSUB 2000
4090 IF AP$="C" THEN FOR Y=0 TO 3: SOUND
Y,TN(Y+1,CC),TN(Y+5,CC): NEXT Y
4100 IF AP$<>"N" THEN SOUND CH,FR,VO
4110 RETURN
8995 REM ***** INCREMENT CHORD FOR
AUTO PLAY
9000 VS 4: INK 1: LET CR=CC-(CC<CM): IF
CR=CC THEN GOTO 10000
9010 IF CR>CT THEN GOSUB 4000 ELSE
CSR CC+3-CB,10: PRINT " ";: CSR
CR+2-CB,10: PRINT INT(MOD(CR,10)+0.5);:
LET CC=CR
9995 REM ***** KEYBOARD ENTRY
10000 VS 4: POKE 64145,128: COLOUR 4,4:
LET IP=ASC(INKEY$): IF IP<5 THEN GOTO
10000 ELSE COLOUR 4,9
10001 IF TN(9,CC)<>1 THEN LET
DU=TN(9,CC): VS 3: CSR 8,4: INK 1: PAPER
12: PRINT NU$(DU+1);: VS 4
10005 REM ***** DIRECT ENTRY
10010 IF CH<3 AND ((IP>49 AND IP<56) OR
(IP>65 AND IP<91)) THEN GOTO 11000
10015 REM ***** HOME/RET TO PLAY/STORE
ELSE 0 FOR NULL NOTE ELSE LEFT/RIGHT
FOR NOTE SELECT
10020 IF IP=13 OR IP=26 THEN GOTO 12000
ELSE IF IP=48 THEN GOTO 13000 ELSE IF
IP=8 OR IP=25 THEN GOTO 14000
10025 REM ***** UP/DOWN FOR OCTAVE
ELSE F2/F6 FOR CHORD ELSE .//,/ FOR
VOLUME
10030 IF CH<3 AND (IP=10 OR IP=11) THEN
GOTO 15000 ELSE IF IP=129 OR IP=133 THEN
GOSUB 16000: GOTO 10000
10040 IF IP=44 OR IP=46 OR IP=60 OR IP=62
THEN GOTO 17000
10045 REM ***** F1 TO CHANGE CHANNEL
ELSE F3/F7 TO CHANGE DURATION ELSE F5
TO COPY DATA ELSE SF2/SF6 TO 'GOTO'
CHORD
10050 IF IP=128 THEN GOTO 18000 ELSE IF
IP=130 OR IP=134 THEN GOTO 19000 ELSE
IF IP=132 THEN GOTO 20000
10060 IF IP=137 OR IP=141 THEN GOSUB
21000: GOTO 10000
10065 REM ***** F4/F8/SF8 TO PLAY CHORDS
ELSE SF1/SF5 TO CHANGE AUTO/MANUAL
PLAY/STORE FLAGS ELSE SF3/SF7 TO
CHANGE TUNE SPEED
10070 IF IP=131 OR IP=135 OR IP=143 THEN
GOTO 22000 ELSE IF IP=136 OR IP=140 THEN
GOSUB 23000: GOTO 10000
10080 IF IP=138 OR IP=142 THEN GOSUB
24000: GOTO 10000
10085 REM ***** EOL TO SAVE/LOAD OR SF4
TO PRINTOUT
10090 IF IP=5 THEN GOTO 25000 ELSE IF
IP=139 THEN GOTO 26000
10095 REM ***** ESC TO ALTER DE DELAY
VARIABLE
```

```
10100 IF IP=27 THEN  GOSUB 27000
10110 GOTO 10000
10995 REM ***** DIRECT ENTRY
11000 LET IP=IP-49: IF IP>6 THEN  LET
IP=IP-10
11010 IF IN(IP,1)=0 THEN  GOTO 10000 ELSE
LET ND=NC: LET OL=OC: LET NC=IN(IP,1):
LET OC=OL+IN(IP,2): IF OC>6 THEN  LET OC=1
11020 LET Y=NP(NC)+3-2*IN(IP,2): VS 2: INK
CO(OC,2): PAPER CO(OC,1): CSR NC,Y:
PRINT DE$(NC,IN(IP,2)+1);
11030 LET FR=NF(NC)/OF(OC): SOUND
CH,FR,VO
11040 GOSUB 3000
11050 VS 2: INK CO(OC,1): PAPER 15: CSR
NC,Y: PRINT DE$(NC,IN(IP,2)+1);
11060 LET OC=OL: LET NC=ND: IF AP$="N"
THEN  SOUND CH,0,0
11070 IF AS$="Y" THEN  GOTO 9000 ELSE
GOTO 10000
11995 REM ***** MANUAL HOME TO PLAY/RET
TO ENTER
12000 IF IP=13 THEN  GOTO 12030
12010 SOUND CH,FR,VO: IF AP$="N" THEN
PAUSE (SP/DU)*DE: SOUND CH,0,0
12020 GOTO 10000
12030 IF CH=3 THEN  LET FR=N3
12040 SOUND CH,FR,VO: GOSUB 3000: IF
AP$="N" THEN  PAUSE (SP/DU)*DE: FOR N=0
TO 3: SOUND N,0,0: NEXT N
12050 IF AS$="Y" THEN  GOTO 9000 ELSE
GOTO 10000
12995 REM ***** ENTER NULL NOTE
13000 VS 4: SOUND CH,0,0: LET
TN(CH+1,CC)=0: LET TN(CH+5,CC)=0: LET
TN$(CH+1,CC)=" ": LET TN$(CH+5,CC)=" ": LET
TN$(CH+9,CC)=" "
13010 CSR CC+3-CB,CH*2+1: PRINT " ";: CSR
CC+3-CB,CH*2+2: PRINT " ";
13020 IF AS$="Y" THEN  GOTO 9000 ELSE
GOTO 10000
13995 REM ***** CHANGE NOTE
14000 VS 3: IF CH=3 THEN  GOTO 14500
14010 FOR N=1 TO 1: CSR NC+5,NP(NC)+1:
PAPER 12: INK 1: PRINT DE$(NC,3);: LET
NC=INT(NC+(IP=8)-(IP=25)+0.5)
14020 IF NC<1 OR NC>12 THEN  INK
CO(OC,1): PAPER 15: CSR OC*2+4,0: PRINT
OC;: LET OC=OC+(NC<1)-(NC>12): LET
OC=OC+6*(OC>6)-6*(OC<1)
14030 IF NC<1 OR NC>12 THEN  LET
NC=NC+12*(NC12)-12*(NC): INK CO(OC,2):
PAPER CO(OC,1): CSR OC*2+4,0: PRINT OC;
14040 CSR NC+5,NP(NC)+1: PAPER 6: INK 1:
PRINT DE$(NC,3);: LET FR=NF(NC)/OF(OC): IF
AP$"N" THEN  SOUND CH,FR,VO
14050 LET IP=ASC(INKEY$): LET N=N+((IP=8)
OR (IP=25)): NEXT N
14060 IF OL<>OC THEN  LET OL=OC: GOSUB
43210
14070 IF TN(CH+1,CC)<>0 THEN  GOSUB 3000
14080 GOTO 10000
14495 REM ***** CHANGE NOTE FOR CH 3
14500 INK 1: PAPER 12: FOR N=1 TO 1: LET
N3=N3+(IP=8)-(IP=25): LET
N3=INT(N3+8*(N3>8)-8*(N3)+0.5): LET FR=N3
14510 CSR 8,3: PRINT N3;: IF AP$<>"N" THEN
SOUND CH,FR,VO
14520 LET IP=ASC(INKEY$): LET N=N+((IP=8)
OR (IP=25)): NEXT N
14530 IF TN(CH+1,CC)<>0 THEN  GOSUB 3000
14540 GOTO 10000
14995 REM ***** CHANGE  OCTAVE
15000 VS 3: FOR N=1 TO 1: INK CO(OC,1):
PAPER 15: CSR OC*2+4,0: PRINT OC;: LET
OC=OC+(IP=10)-(IP=11): LET
OC=INT(OC+6*(OC>6)-6*(OC<1)+0.5)
15010 CSR OC*2+4,0: PAPER CO(OC,1): INK
CO(OC,2): PRINT OC;: LET FR=NF(NC)/OF(OC)
15020 IF AP$<>"N" THEN  SOUND CH,FR,VO
15030 LET IP=ASC(INKEY$): LET N=N+((IP=10)
OR (IP=11)): NEXT N
15040 IF TN(CH+1,CC)<>0 THEN  GOSUB 3000
15050 IF OL=OC THEN  GOTO 10000 ELSE
LET OL=OC: GOSUB 43210: GOTO 10000
15995 REM ***** CHORD +/-
```



```
16000 FOR N=1 TO 1: VS 4: INK 1: PAPER 15:
LET CR=CC-(IP=133)+(IP=129): LET
CR=INT(CR+(CR>CM)-(CR<1)+0.5)
16010 IF CR<CB OR CR>CT THEN  GOSUB
4000 ELSE  IF CR<>CC THEN  CSR
CC+3-CB,10: PRINT " ";: CSR CR+2-CB,10:
PRINT INT(MOD(CR,10)+0.5);: LET CC=CR
```

SMALL ADVERTS

For Sale

DISC A

3D Tachyon Fighter
Salty Sam
Hawkwars
Music Box Dancer
Agrovator
Pontoon/Blackjack
Cobra
Draughts

DISC B

Knuckles
Alice
Railways
Wargame
Toado
Maxima
Star Command
Qogo 2
Zarkos
Blobbo

DISC C

Chamberoids
Target Zone
Tower
Chainsaw
Adventure
Fruit Machine
Zombies

DISC D

Miscellaneous Basic Games

DISC E

SDX Tape-To-Disc Transfer

DISC F

40 Col Newword + Manual

DISC G

40 Col Newword (Man)
System Disc
Utilities
2 Games

DISC H

Cosmic Raiders
Missile Command
Arcadians
Misc. Basic Games

DISC I

EDASM + Manual

DISC J

EDASM + Manual
Prices as follows:
Disc's A,B
£10.00 each
Disc's C,F,G,H,I,J
£8.00 each
Disc's D,E
£5.00 each
The disc's are 250K Type 02.

Dave Wemyss
10 Bonfield Road,
Strathinness,
St Andrews,
Fife.
Tel 0334 85 335.

To advertise your private items for sale, it costs just £2.00!! Please send your adverts by post to: Phil Eyres 13 Copse Rd, Townhill Park, Southampton. SO2 2GY.

```
16020 IF AP$="C" THEN  FOR M=0 TO 3:
SOUND M,TN(M+1,CC),TN(M+5,CC): NEXT M
ELSE  IF AP$="Y" THEN  SOUND
CH,TN(CH+1,CC),TN(CH+5,CC)
16030 LET IP=ASC(INKEY$): LET
N=N+((IP=133) OR (IP=129)): NEXT N: IF
TN(CH+1,CC)<>0 THEN  GOSUB 2000
16040 RETURN
16995 REM ***** CHANGE VOLUME
17000 VS 3: INK 1: PAPER 12: FOR N=1 TO 1:
LET VO=VO+(IP=60)-(IP=62)+(IP=44)-(IP=46)
17010 LET VO=INT(VO+0.5+(VO>15)-(VO<0)):
CSR 2,4: PRINT NU$(VO+1);: LET
IP=ASC(INKEY$): LET N=N+((IP=60) OR
(IP=62) OR (IP=44) OR (IP=46))
17020 IF AP$<>"N" THEN  SOUND CH,FR,VO
17040 NEXT N: IF TN(CH+1,CC)<>0 THEN
GOSUB 3000
17050 GOTO 10000
17995 REM ***** CHANGE CHANNEL
18000 VS 4: PAPER 15: INK 1: FOR N=1 TO 1:
CSR 2,CH*2+1: PRINT "=";: IF AP$="Y" THEN
SOUND CH,0,0
18010 LET CH=CH-(IP=128)+4*(CH>2): CSR
2,CH*2+1: PRINT ">";: IF AP$="Y" THEN
SOUND CH,TN(CH+1,CC),TN(CH+5,CC)
18020 LET IP=ASC(INKEY$): LET
N=N+(IP=128): NEXT N
18030 IF CH=3 THEN  LET FR=N3 ELSE  LET
FR=NF(NC)/OF(OC)
18040 IF TN(CH+1,CC)<>0 THEN  GOSUB 2000
18050 IF AP$<>"N" THEN  SOUND CH,FR,VO
18060 GOTO 10000
18995 REM ***** DURATION
19000 VS 3: INK 1: PAPER 12: FOR N=1 TO 1:
LET DU=INT(DU+((IP=134) AND
(DU>1))-((IP=130) AND (DU<8))+0.5)
19010 CSR 8,4: PRINT NU$(DU+1);: LET
IP=ASC(INKEY$): LET N=N+((IP=130) OR
(IP=134)): NEXT N
19020 IF TN(1,CC)<>0 OR TN(3,CC)<>0 OR
TN(5,CC)<>0 OR TN(7,CC)<>0 THEN  GOSUB
3000
19030 GOTO 10000
19995 REM ***** COPY ROUTINE
20000 POKE 64145,128: VS 5: CLS : PRINT "
COPY FROM X TO Y INTO Z": PRINT " Copy
forward or backward, e.g."
20010 PRINT " 1 to 8 ] 10 or 8 to 1 ] 10": PRINT "
But watch message for limits"
20020 PRINT " Copy 1 or all chan. (0]3 or A)"
20030 CSR 1,5: PRINT CHR$(5);: INPUT
"CHANNEL (0,1,2,3 or A) ";IP$
20040 LET C1=ASC(IP$)-48: IF (C1>3 AND
C1<>17) OR (C<1) THEN  GOTO 20030 ELSE
CSR 1,5: PRINT CHR$(5);: CSR 1,4: PRINT
CHR$(5);
20050 CSR 1,6: PRINT "Use F2/F6/%2/%6 then
                    <ret>";: LET C4=0
20060 CSR 1,5: IF C1>3 THEN  PRINT "ALL
CHAN"; ELSE  PRINT " CHAN";C1;
20070 CSR 10,5: PRINT CHR$(5);: PRINT "From
?": GOSUB 20500
```

```
20080 VS 5: LET C2=CC: CSR 14,5: PRINT C2;:
CSR 19,5: PRINT "to ?";: GOSUB 20500
20090 VS 5: LET C3=CC: CSR 21,5: PRINT C3:
CSR 26,5: PRINT "] ?";: LET C4=1: GOSUB
20500
20100 VS 5: LET C4=CC: CSR 27,5: PRINT C4;:
CSR 10,5: PRINT CHR$(5);
20110 IF C1<4 THEN LET CR=C1 ELSE LET
CR=0: LET C1=3
20120 IF C3>C2 AND C4+C3-C2>CM THEN
LET C3=CM+C2-C4 ELSE IF C2>=C3 AND
C4+C2-C3>CM THEN LET C3=C4+C2-CM
20130 VS 5: PRINT C2;" ]";C3;" to";C4;" ]";: IF
C3>C2 THEN PRINT C4+C3-C2; ELSE IF
C2>=C3 THEN PRINT C4+C2-C3;
20140 CSR 1,7: PRINT CHR$(5);: PRINT "O.K.
(Y/N) or QUIT (Q) ";: POKE 64145,128
20150 IF INKEY$="Q" THEN GOTO 20900
ELSE IF INKEY$="N" THEN GOTO 20000
ELSE IF INKEY$<>"Y" THEN GOTO 20150
20160 CSR 1,7: PRINT CHR$(5);
20170 IF C3C2 AND C4 THEN GOTO 20400
ELSE IF C3C2 THEN GOTO 20300
20195 REM ***** C2>=C3, THUS COPY
SOURCE BACKWARD INTO TARGET
FORWARD
20200 LET Z=C4-C2: FOR N=C2 TO C3 STEP
-1: FOR M=CR TO C1: GOSUB 20700: NEXT M:
GOSUB 20800: LET Z=Z+2: NEXT N: GOTO
20900
20295 REM ***** C3>C2 AND C4>=C2 THUS
COPY SOURCE FORWARD INTO TARGET
FORWARD
20300 LET Z=C4-C2: FOR N=C2 TO C3: FOR
M=CR TO C1: GOSUB 20700: NEXT M: GOSUB
20800: NEXT N: GOTO 20900
20395 REM ***** C3C2 AND C4<C2 THUS
COPY SOURCE BACKWARD INTO TARGET
BACKWARD
20400 LET Z=C4-C2: FOR N=C3 TO C2 STEP
-1: FOR M=CR TO C1: GOSUB 20700: NEXT M:
GOSUB 20800: NEXT N: GOTO 20900
20500 IF INKEY$<>"" THEN GOTO 20500
20510 LET IP=ASC(INKEY$): IF IP=13 AND
C4<3 THEN RETURN ELSE IF IP=137 OR
IP=141 THEN GOSUB 21000
20520 IF IP=129 OR IP=133 THEN GOSUB
16000
20530 IF C4=0 THEN GOTO 20510 ELSE LET
C4=1
20540 IF C2>C3 AND CC<=C2 AND
CC>=C3+C3-C2 THEN LET C4=3
20550 IF (C2<C3 AND CC+C3-C2>CM) OR
(C2>C3 AND CC+C2-C3>CM) THEN LET C4=2
20560 VS 5: CSR 1,7: PRINT CHR$(5);: IF C4=2
THEN PRINT "To range MAX, No Problem!";
ELSE IF C4=3 THEN PRINT "Z within X]Y,
WONT WORK!";
20570 GOTO 20510
20600 LET TN(M+1,N+Z)=TN(M+1,N): LET
TN(M+5,N+Z)=TN(M+5,N): IF N+Z>QT(M+1)
THEN LET QT(M+1,N)=N+Z: IF N+Z>QT(5)
THEN LET QT(5)=N+Z
20610 LET TN$(M+1,N+Z)=TN$(M+1,N): LET
TN$(M+5,N+Z)=TN$(M+5,N): LET
TN$(M+9,N+Z)=TN$(M+9,N): LET
TN(9,N+Z)=TN(9,N): LET
TN$(13,N+Z)=TN$(13,N)
20620 RETURN
20700 LET TN(M+1,N+Z)=TN(M+1,N): LET
TN(M+5,N+Z)=TN(M+5,N): LET
TN$(M+1,N+Z)=TN$(M+1,N): LET
TN$(M+5,N+Z)=TN$(M+5,N): LET
TN$(M+9,N+Z)=TN$(M+9,N)
20710 IF N+Z>QT(M+1) THEN LET
QT(M+1)=N+Z: IF N+Z>QT(5) THEN LET
QT(5)=N+Z
20720 RETURN
20800 LET TN(9,N+Z)=TN(9,N): LET
TN$(13,N+Z)=TN$(13,N): RETURN
20900 LET CR=CC: GOSUB 4000: GOSUB
43500: GOTO 10000
```

SOFTWARE
PRICE LIST APRIL 1989

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

Title	Stock
26*26 SPREADSHEET	Y
ADVENTURE QUEST	Y
3D TACHYON FIGHTER	Y
AGROVATOR	N
ALICE	Y
ASTROMILLON	Y
ASTROPAC	Y
ATTACK OF KER TOMS	Y
BACKGAMMON	Y
BASIC BUSINESS	Y
BLOBBO	Y
BOUNCING BILL	N
BRIDGE	Y
CAVES OF ORB	Y
CHAMBEROIDS	N
CHARMKATZ	Y
CHESS	Y
COLOSSAL ADVENTURE	Y
COMBAT	Y
CONT. RAIDERS	Y
CRYSTAL	Y
DENNIS AND THE CIRCUS	Y
DISASM	Y
DOODLEBUG	Y
DOWNSTREAM DANGER	Y
DRAUGHTS	Y
DR FRANKIE	Y
DRIVE THE CEE-5	Y
DUNGEON ADVENTURE	Y
EDASM	Y
EMERALD ISLE	Y
ESCAPE FROM ZARKOS	Y
EXTENDED BASIC	Y
F1 SIMULATOR	Y
FATHOMS DEEP	N
FIG FORTH £10.00	Y
FIG-FORTH MOC	Y
FIG FORTH SDX	Y
FIREHOUSE FREDDIE	N
FIRST LETTERS 1	Y
FLUMMOX	Y
GOLDMINE	Y
GRAPHICS	Y
HAWKWARS	Y
HELI-MATHS	Y
HIGHWAY ENCOUNTER	N
HUNCHY	Y
ICEBURG	Y
JET SET WILLY	Y
JOHNNY REB	Y
JUMPING JACK FLASH	Y
KILLER TOMATOES	Y
KARATE KING	Y
KILOPEDE	Y
KNUCKLES	Y
LITTLE DEVILS	N
LORDS OF TIME	Y
MANIC MINER	Y
MISSILE COM & ARCADE	Y
MATHS 1	Y
MAXIMA	Y
MEMOCHEQUE	Y
MEMOSKETCH	Y
MEMOSKETCH SDX	N
MINEFIELD	Y
MINER DICK	Y
MISSION ALPHATRON	Y
MISSION OMEGA	N
MTX UTILITIES	Y
MURDER AT THE MANOR	Y
MUSIC PAD	Y
NEMO	Y
OBLITERATION ZONE	Y
OBLOIDS	Y
PAINTBOX	Y
PAYROLL	Y
PHAID	Y
PHYSICS	Y
PONTOON & BLACKJACK	Y
POT HOLE PETE	Y
PUC-MAN	Y
PURCHASE LEDGER	Y
QOGO	Y
QOGO 2	Y
QUANTUM	Y
QUAZZIA	N
QUEST 1	Y
REVEAL	Y
REVERSI	Y
ROLLA BEARING	Y
RUTHLESS B.	Y
SALES LEDGER	N
SALTY SAM	Y
SEPULCRI SCELERATI	Y
SMG	Y
SMGII	Y
SNAPPO	Y
SNOWBALL	Y
T.SNOOKER	Y
SON OF PETE	Y
SOUL OF A ROBOT	Y
STAR COMMAND	Y
SUPA CODER	Y
SUPER BIKE	Y
SUPER MINEFIELD	Y
SURFACE SCANNER	N
TAPEWORM	Y
TARGET ZONE	Y
THE WALL	Y
TOADO	Y
THE DESIGNER	Y
THE KEY TO TIME	Y
THE ZOO	Y
TUMBLEDOWN TOWER	Y
TURBO	Y
USER BASIC	Y
USER BASIC SDX	Y
UTILITIES SDX	Y
VERNON AND THE VAMPIRES	Y
WORD & PICTURE	Y

ALL AT £5.00

NEW SOFTWARE

Announcing the arrival of the following items from MOC.

MTX CardBox

The only purpose-written database system for Memotech non-CP/M disc systems. Extremely fast and easy to use, it comes with a comprehensive and easy to follow manual. Non-CP/M disc only.Touch Typist

The easy way to learn to touch type on the MTX keyboard. You'll be very surprised how easy it can be! Disc or tape.

Advanced Touch Typist

Expand your touch typing skills with this excellent follow-on to Touch Typist. Disc or tape.

Second Hand Hardware For Sale

Memtalk

MTX Speech Synthesiser Including instructions and demonstration tape £20.00 + £2.00 P&P

RS232 Board

Fulling populated with chips including fly leads. £25.00 + £1.00 P&P

32K Memory Expansion

(Not working - PROM CHIP BLOWN) Useful for extra memory chips! £10.00 + £2.00 P&P

32K Memory Expansion

For MTX 500. We have three off in good working order. £25.00 + £2.00 P&P

TAXAN KP-810 PRINTER

1 off ONLY

As New

140 Characters per second, DRAFT/NLQ, Tractor or Friction Feed, Manual, Graphics Dumping. Includes 4ft printer connecting cable.

£125.00 + £2.50 P&P

Shinwa Cp 80

Good working order. Includes 2 ink ribbons. Tractor Feed. Centronics.

£75.00 + £3.00 P&P

Ferguson Data Recorder

Level Indicator, Tape Counter, Power switch & Pause Button.

£9.00 + £1.00 P&P

MTX500 Motherboard

This board is fully populated, but is missing the Video Processor board. It has Norway ROMS fitted and the major chips are socketed.

This is the last board. One of the others has been proved to be working as is now being used!! £15.00 + £2.00 P&P

MTX500 Motherboards

These boards have most of the IC's, but are missing the Video Processor board and a few support devices. They have Norway ROMS fitted and the major chips are socketed. They should be ideal for spares or for more memory chips! £10.00 + £2.00 P&P

MTX 512's

We have two MTX512's in good working order. Both have manuals, but one is short of a cassette lead. One of them has been specially adapted having an RGB socket.

£40.00 + £2.50 P&P Each

MTX Dust Covers

Two secondhand dust covers for your MTX 500 or 512. A must for keeping that keyboard clean.

£3.00 + £0.50 P&P Each

MTX Keyboard

Need a spare keyboard?? We have one

£10.00 + £1.00 P&P

Video Board

This is the little board that fits on the motherboard. It does not contain the video chip, which is situated on the motherboard.

£5.00 + £0.50 P&P

SDX 80 Column Boards

Do you have an Basic SDX system?? Thinking of upgrading to an 80 Column CPM system? Well this board, is all you need in the way of electronics. It plugs into the left hand end of your machine. You then plug in a monitor and your up and away. CPM, Newword & Supercalc provided.

These boards will require that you make a lead to connect the board to a monitor. We will provide the necessary diagram.

£50.00 + £2.00 P&P Each (We have three!!)

SDX Controller

One SDX controller, that connects on the left hand bus connector. I am not sure if you will need one of the 80 column boards above, but we are willing to help you try it all out. All you need is a disc drive to plug into this controller.

£50.00 + £2.00 P&P

... MORE TWO PAGES ON!!!

NEWWORD COLUMN MODE

By
Erik D'Hondt

In answer to the article written by Andrew Fox (Vol 5 - Issue 2) the problem of turning the column mode in Newword on and off was already solved by the German User Group (Info 12 - 34: Newword Alte Version kann Spaltenbloecke by Michael Kessler).

It cannot easily be done in his article, because the file NWMMSGS.OVR where the chnages must be done, has a length of 52K and the normal TPA is only 54K!

Here follows the 'HOW TO', step by step, using the utility DDT.COM delivered with the system disc.

First of all you had to generate a new CP/M system with a size of 60K. The disc must contain the following files:-

ASM.COM
CBIOS.ASM
CPMGEN.SUB
ED.COM
ENTER.COM
MOVCPM.COM
SUB.COM
WRTBIOS.COM
WRTCPM.COM

Type:-

A>SUB CPMGEN B:

After a coldboot the disc contains a TPA of 60K. Now you must PIP or COPY the files NWMSGS.OVR and DDT.COM to this disc.

Type:-

A>DDT NWMSGS.OVR

DDT VERS 2.2

NEXT PC

D100 0100

With the command s(ubstitute) you have to change the following addresses (/ means RETURN):

s3A12/12/./

s3A58/09/./

s3A95/09/./

s3ADC/07/./
s3B19/64/69/73/63/20/73/74/6F/72/65/06/03/1B/40/4E/1B/00/20/63/

6F/6C/75/6D/6E/20/01/06/08/1B/40/52/1B/00/20/69/6E/73/65/72/74/

20/64/6F/63/75/6D/65/6E/74/0D/0D/08/02/02/06/./

sCD4B/4E/./

sCDB8/4E/./

^C

A>SAVE 208 NWMSGS.OVR

Now you can toggle the column mode in the EDIT MENU with ^KN. On the status line appears or disappears the message 'Column'.

Whilst on Andrew's article, it is almost impossible to change the drivers in NWMSGS.OVR. If you need another driver I have the source code of a driver for the printers:

DMX80

Epson FX80

Panasonic P1090

Epson LQ500

(all with graphics!! and 2 extra customs to install).

Those drivers can easily be modified because the source code is well documented. So if you wish to change or adapt the driver, please contact me.

The German User Club also has a lot of other adaptions like:

Using the function keys and the keys on the numeric key pad by Newword using the command F.COM, a new screendriver, a new routine for cursor positioning, initalizing the printer, avoiding the messages with D(ocument), N(on-document) ...

If anyone is interested in these changes, please contact me.

Erik D'Hondt
Wilgstraat 25A
B-9440 EREMBODEGEM
Belgium.

Phil - I have written to Erik asking him for some more details of the changes to Newword and also about the German User Group in general.

(Erik - I suspect you will get quite a few letters this month!! Thanks very much for this information and for supporting the club)

MEMOTECH v AMIGA

By

Geoff Gardiner

With the announcement of a new operating system that enables an Amiga to run a recoverable fixed RAM DISC the Amiga now seems to have caught up with all of the revolutionary Memotech innovations of 1984 except possibly the "Ring". Does that make it the natural successor to the Memotech? I have been trying one out to find out.

When I reported on the Atari I condemned it for it's "WIMP" only environment with no easy way to operate it by command lines. The Amiga has a "command line interface" (CLI), as it is called, as an alternative to WIMP, and indeed many commands can only be accessed with it. But getting into the CLI mode is a bit of a rigmarole. Drives are df0:, RAM:, etc, which is not so handy as A:, B:. CLI takes a bit of learning and the instruction book does not come with the computer but as an expensive extra. But it is powerful. One great incidental feature is that one is not confined to eight character file names, so file names can have real meanings. But directories are read rather slowly compared with CPM.

Obviously a 68000 chip operating at over 7 mghz is going to be faster than a Z80. I tested Amigabasic with the Basic program of Mark Cytera, the one Phil refers to as Blancmange. On the MTX it takes about 15 minutes to complete the graphic. At the same definition the Amiga takes a little under 3 minutes. In 15 minutes it will construct the graphic in 630 * 512 definition. But that high resolution on the Amiga is not really usable as the display is interlaced (to save one having to buy an expensive multisync monitor) and the flicker is intolerable. Perhaps it is not so bad on the american 60hz system. Low resolution on a PAL screen is 320 * 252 which is a bit better than the MTX and many more colours are allowed. The increase in definition looks better than the figures would indicate. Many programs that originate in America use 320 * 200 because that is all the NTSC system will allow. (Have you ever seen NTSC television? It is awful!) They produce an elongated screen as on the MTX but it sits at the top of the screen so one cannot correct it as with the MTX by altering the height control on the monitor. However for some reason the elongated effect is not so noticeable.

The Amiga has only 8 sprites but that comparison with the MTX is a bit misleading as they are handled differently. But they can only be used in three colour mode, which is useless. Things called 'bobs' are used instead with more colours but they cannot be moved so fast. Possibly the difference is no great handicap. I have not tried them out. Sprites and bobs are designed on a screen; there is no Genpat command. A real handicap is having no Angle, PHI, or Arc commands. A knowledge of advanced trigonometry therefore would be useful to enable one to write one's own routines. But one does have loads of commands to handle files, disks, the mouse, and to use pulldown menus. Assembly language routines can be accessed but there is no built in assembler. Assembler and disassembler ("Front Panel") are an expensive extra.

Many built in routines can be accessed from Basic. it is like being able to run CPM routines from Basic. There are hundreds of these routines, many

in a 192K ROM, and as they can be accessed from both Basic and assembler, program writing could be a dream - provided you can afford the very expensive manuals to learn about them and can understand "C" language in which the routines are written. One gets the impression that "C" was used because there was a rush to produce the system and the screen handling routines seem slow as a result possibly of having been written in a high level language. Certainly the WP program "Protext" appears to have achieved an enormous speed up in screen handling through using its own routines, not those built in. Perhaps this is also one of the improvements in the new ROM and DOS that have just come out. I have not had a chance to use them.

Like the MTX the Amiga can have multiple screens and these can be re-sized and swapped by mouse control. It can do several things at once ("multitasking"). The sound is excellent but the sound commands in the Amiga Basic do not exploit it well enough: they are not as good as those in the MTX basic.

The Amiga's main fault is that it easily gets its software knickers in a twist, and when this happens it powers off, and an unexplained "guru" message comes on screen. The MTX can lock up but at least it does not power off and a reset on the MTX does not erase the internal memory or the RAM-Disk. The Amiga can control 8 megabytes of memory (in theory the memory register of the 68000 can control 16 megabytes) and you certainly need at least 1 meg. More is desirable to be as convenient as an FDX fitted with a 1 Meg silicon disc. The Amiga 500 comes with half a meg and the upgrade to 1 meg is very easily fitted. The upgrade can contain a clock which is a boon to those people who write viruses as they can be triggered by date. I should think all Amigas with internal clocks had virus thouble on the last Friday the thirteenth. Anyone contemplating buying an Amiga for serious use should budget for an extra 500K of memory and two external drives, and for some expensive manuals. OK, that is still less costly than an MTX+FDX+Silicon disc was in 1984.

The MTX-FDX solved the problem of whether to aim for a text based system or a graphics based system by providing both. The Amiga is essentially a graphics based system. The penalty for that can be slow screen update. When it is working in its fullest colour mode the amount of screen information required absorbs a colossal amount of memory and this is taken from the main memory, not from separate screen memory as in the MTX and FDX. This is why a vast memory is required. But with a 24 bit memory register controlling memory is not great problem for a 68000 chip.

Professional programs are cheap compared to PC equivalents, and usually do much more than on a PC. FDX users will find clones of Supercalc and Newword are available as Superplan and Protext. Superplan uses slash commands almost all of which are identical with Supercalc. Its screen scrolling is slower than SC on the FDX but one can make large screen moves more easily. Recalculation is much faster and more logical as one recalc will do even if the formula references dodge about the screen. There is one serious flaw; you cannot have all the formulas displayed on screen at once. This is an Auditor's nightmare as a Chartered Accountant tells me that the biggest problem with spreadsheets is unobserved formula mistakes. Certainly it was such

an error that lead me to pay too much tax. A sillier error is that the upright line dividing the letters at the heads of columns prints as an italic 3 on an Epson compatible printer. Supercalc users will remember that that is something one can alter with the install program. There is no such program with Superplan. But the latter has some superb graphing facilities which take advantage of the Amigas great capabilities in that area. It can also be made to read information from other files automatically. There are a few minor bugs in the program.

Protext is very like Newword. It has all its virtues, including conditional merg printing, though it behaves slightly differently as I have discovered when I tried to use the techniques of merge printing I have described in an article. The important improvement is that it as 26 custom print controls instead of four, so any printer ought to be controllable. There are masses of printer drivers and they can be amended. I was able to amend the Epson 24 driver to run the Star NB24; the latter needs a different code to do microspacing as its head moves in 180ths of an inch, not 120ths. Screen update with Protext is faster than Newword; this seems to show that slow screen update on the Amiga is essentially a programming, not a hardware fault.

But Protext was not written for the Amiga; it was merely adapted to it. To see what can be done when the programmer is writing for a specific machine one has to try Kindwords 2. This remarkable program can be obtained for as little as #35.00. It still lacks advanced WP features like conditional merge printing, wordcount, and alternate page printing, but know doubt these will come. What it does have is that it is programmed entirely for a graphics environment. OK that makes screen update slow, painfully slow at scrolling, but then it can show nine mixed fonts onscreen, in plain, bold, italics or underline. One font included is maths symbols and another is Greek. To see Ancient Greek on screen in bold italics must be the ultimate. Of course these scripts have to be printed in graphics mode which takes about 15 minutes a page on the Star. And doing it revealed a fault in the ROM on the STAR. The right margin setting code does not work properly. (I understand there is now a ROM upgrade).

Reformatting is entirely automatic with Kindwords. Autohyphenation while you type is also available but slows the screen update to a snails pace. Headers and footers are shown on screen so one always sees exactly what will be printed, but headers and footers cannot be changed: they always apply to the whole file. There is a very user friendly spell checker which can be loaded in memory so that it works very fast. Graphics can be incorporated in the text. A sample file provided shows a map of the United States and it can be resized at will. I suspect that when this program is perfected it will be the ultimate in user freindly WP. The Amiga is suffering as did the MTX from the fact that few programmers are writing specifically for its enormous abilities and more programs like Kindwords are needed.

Those who are waiting for a really dramatic improvement on the FDX may not have to wait much longer. An Amiga is said to be on the way which uses the 68020 chip at 14 or even 25 mghz, and with very high resolution non interlaced grahics and more than 2 meg of memory (Phil - Mortgages available!!). That sounds like a dream come true - if programs to exploit it are forth coming! For sophisticated programs like Kindwords the additional speed would be a boon.

The Amiga can control 5.25 inch drives so FDX drives do not have to go to waste. Whether there is any way of using the silicon disc I doubt. Perhaps its chips could be transfered to an Amiga expansion board though they are 150 nanosecond and the Amiga ideally needs 120ns. 41256's are specified but my sidisc uses 51256's which I believe are CMOS and twice as expensive.

On the whole I think it likely that lacking some update of the Memotech the perfectionists will slowly drift to the Amiga. In that case a perfectionists club will be needed so perhaps MOC should become M and AOC.

REVIEWS

QUEST ONE
THE MURDER
Reviewed By
Andy Owen

The Murder is, apparently, the first in a series of menu driven adventures originally released by Syntaxsoft in 1986.

Although my copy came without instructions it is quite easy to pick up the idea of the game, which is to find sufficient evidence to solve the murder, and to accuse one of the suspects, gathered in the lounge, of murdering the late Lord Grove. If you don't have the required evidence when you accuse a suspect, Sgt Bull asks "Inspector, surely you're not intending to go to court with such flimsy evidence as that?", whereupon you nod dejectedly and are relieved of the case.

The adventure is set at Northmoor Manor with clues scattered all over the place. Red Herrings are in abundance in this game, but some of them may prove to be important for further progress, later in the game. A nice touch is that if you drop the revolver it goes off and kills you. Other good points include the necessary bribing of one of your officers, and the occasional touches of humour, which the Author displays, help to cheer up the game.

As always with adventures, it is necessary to make a map, although I get the impression that the game is not all that big. The RAMsave and RAMload options are useful additions (they can be found - a little confusingly - under the QUIT option) although I do find that the lack of a save to tape option can be annoying. The menu system is one that I find both easy and enjoyable to use, which, along with the pleasant screen presentation and quick update, make for a very well thought out and presented game.

Summing up I think this is a well presented and fairly easy to play, if difficult to complete, adventure. Worthy of a place in anyone's collection.

Instructions
Graphics
Playability 80%
Addictiveness 70%
Sound
VFM 70%
Overall 75%

MTX FRUIT MACHINE
Reviewed By
Andy Owen

If you've ever been in an arcade you will have felt the lure of the Fruit Machine, and you will probably have succumbed to the "just one more go" syndrome, which it is meant to promote. Now you can sit in the comfort of your own home and succumb, without risking any of your hard earned cash.

MTX Fruit Machine is, as you will have gathered, an attempt to simulate a Fruit Machine, and is, as far as I know the first such attempt on the Memotech (although it has been attempted numerous, mostly unsuccessful, times on other (more popular) home computers.

After you eventually manage to load the game (your azimuth setting and volume must be absolutely perfect, and you must follow the instructions on the inlay to the letter) you are given £5.00 in credits. To spin the reels F8 is pressed; automatically debitting your credits by 10p. Many different features are available, such as hold, Spin, Nudge, Gamble, etc... To collect your winnings and start the reels spinning START is again pressed. The game ends when you run out of credits and you are told how many goes you survived for (my maximum is 97 goes but I imagine you can do better).

Full instructions are on the inlay and are also present on the loading screen. The graphics are large and well detailed and yet don't quite capture the feel of the arcade. Although a good and addictive game in its own right I don't think it quite accomplishes what it set out to do.

Overall I think this game is a good buy, as long as you can put up with the loading problems, if not then it will soon be sitting on the shelf wondering when someone will play it again.

Instructions 65%
Graphics 80%
Addictiveness 75%
Sound N/A
VFM 65%
Overall 70%

System Variables

or How to add variety to your programming

J C Hodgson

In the February issue of the MOC mag the question was asked, 'What are System Variables and how can I use them'. Let me try to explain why a computer requires System Variables and how you can use some of them in your programs.

When you write a program you will often create space in RAM to store temporary values. Let us say that you want to store the value 12345, then in Basic you would write this in the form:

```
LET TEMPVAL=12345
```

And in assembler it would be:

```
TEMPVAL:DW 0
        LD HL,12345
        LD(TEMPVAL),HL
```

When you switch on your computer and the Basic READY prompt is displayed you are running a program that is held in ROM. Like all programs, the Memotech Basic needs to store temporary values as it runs. Because the program is held in ROM, then an area of RAM must be set aside, outside of your Basic program, for use by the MTX Basic. This is called the System Variable area and in the MTX is located at the top of RAM.

The MTX Basic uses this area in three different ways and the following are examples of some of the System Variable.

1) As a temporary storage area for various calculations.

The locations #FDD1 to #FDDD are used by the maths routines when values are calculated.

2) Values are set to show the state of the MTX Basic.

#FA83 start address of the keyboard buffer.

#FD5F cassette BAUD rate

#FD68 SAVE or LOAD flag

#FD7C last key pressed

3) Locations that allow you to enter a JUMP address to enable your own routines to be used from within MTX Basic.

#FA89

USER command, this will jump to any address with the command USER. Used by Memotech for the disc drive routines.

#FD54

MTX Basic will look at this location before an error message is issued.

To get a full list of the System Variables, with comments, write to Phil. OK, so you now have a list of the System Variables and you want to know how you can use them in your program. The bad news is that only a few of the System Variables can be used from Basic, a good reason to learn to program in assembler. I will now give you two examples that can be used from Basic and an example that will require the use of the assembler. How often have you written a Basic program with a large number of PRINT statements, but during the testing of the program you need the results output to your printer. This would mean changing the PRINT statements to LPRINT and then changing them back afterwards, not an easy task in a large program. Or you may want to switch your program output between the screen or printer. This can be done with PRINT, LPRINT and IF statements but it adds to your program length and is not very tidy. The first program will switch output between the screen and printer using only the PRINT statement.

```
10 PRINT "Output to the Screen or Printer"
20 INPUT "S or P ";A$
30 IF A$="S" OR "s" THEN POKE(64886),0:
GOTO 100
40 IF A$="P" OR "p" THEN POKE(64886),1:
GOTO 100
50 PRINT "Not valid, enter S or P"
60 GOTO 10
100 PRINT "This is a test print line"
110 GOTO 10
```

In your program you may want to know the last key you pressed. This can be done from within your program, but this value is also stored in a System Variable. The following program will show this.

```
10 PRINT 'ASCII value of last key pressed =
';PEEK(64893)
20 LET A$=INKEY$
30 GOTO 10
```

The last example will show how to trap errors that would normally cause your Basic program to fail. First enter lines 60 to 90 and run the program. The program will fail with the error Div /0, (divide by zero).

```
60 PAUSE 500
70 LET A=3/0: REM This would give a divide by
zero error
80 PRINT "Return to Basic and continue your
program"
90 GOTO 60
```

Now add the lines 10 to 50 and rerun the program. This time the error is trapped and you are returned to your program.

```
10 CODE
4007      RET ; Skip this routine when you run
this prog.
4008      RST 10
4009      DB #90,"Error in Basic",#0A,#0D
401A MESS:  RST 10
401B      DB #92,"Continue or Stop",#0A,#0D
402E      RST 10
402F      DB #8E,"Enter C or S",#0A,#0D
403E      XOR A
403F TESTKEY:CALL #79
4042      JP Z,TESTKEY
4045      CP "C"
4047      JP NZ,NEXTCHR
404A      POP AF; Remove ERROR return
address to Basic
404B      RET
404C NEXTCHR:CP "S"
404E      CALL Z,#250; Return to MTX Basic
4051      JP MESS
4054      RET
Symbols:
MESS   401A    TESTKEY 403F
NEXTCHR 404C
20 REM Set error jump instruction JP #4008
30 POKE(64854),64
40 POKE(64853),8
50 POKE(64852),195
```

If you are interested, Alan Hamilton has a routine that will trap disc errors and return control back to your Basic program. Hopefully this has given you a small insight into the way System Variables can be used in your programs, and I hope you will be encouraged to have a go yourself.

```
PATTERNS
ONION

10 VS 4: PAPER 1: CLS: PAPER 1
20 FOR J=100 TO 25 STEP -5
30 FOR A=44.77 TO 49.48 STEP .0005
40 PLOT 128+90*SIN(A/.75)
        *COS(A/J)),96+(95*COS(A))
50 NEXT A: NEXT J
```

GAP FILLER

By
Paul Wood

This next bit should be a useful snippet. It came about after repairing several disc controller cards, in practice it seems to work ok with all the controllers I've tried.

For all the owners of **3.5inch CP/M systems**, if you're fed-up with not being able to run the longer std MTX programs, try this!!. Boot up the system disc, now from here, there are two ways to the std MTX basic.

FDXB

This only allows programs up to **32K**, (ie turns the machine into an MTX500 with disc drive).

MTX

This resets the system to a std MTX512, which until recently I assumed, had no disc functions. From here enter ROM 5

and away you go, std 64K MTX512 with disc drive!!!! The question remains, why not put this in the manual!!

INPUT, OUTPUT AND INTERRUPTS

By

Mike Frymyer

An interesting feature of the Z80 processor, unlike many other 8- bit as well as 16 bit processors is its ability to handle I/O functions without using up valuable memory space. For those not so well aquainted with the manor in which these things are accomplished, I will attempt an explanation...

Most microprocessors treat in/output operations like an extension of the computers memory. For example, if you had a computer using a 6502 processor and you thought a joystick was a desirable add- on, then in order to create an input port, you would first have to decide which memory location you will sacrifice, then use hardware to decode that location exclusively. After all it would not do to have your add-on writing to more than one location. Now, once you've done all that, all you have to do is read the memory location. At first glance this might not seem like too much - Hold that thought I'll come back to it.

The Z80 (and 8080) both treat I/O operations as a completely seprate function from memory addressing. This means that you can keep all of your memory for the program that will handle the I/O ports that you want to into your computer. The Z80 can address 255 separate ports using its standard method. This raises a draw- back in that you are limited to 256 ports. However, you can still use the memory addressing method if that's not enough for you.

Back to specifics. The Z80 (and the MTX) use porting to do everything. To the computer every device is a port. In the case of the MTX the monitor(s) are at ports 1,2 (TMS9929) and ports 30,31,32 (6845).

Under basic, it's made quite easy by simply defining calling and writing to a particular screen number. The keyboard and all other plug-on and peripherals for the computer are treated in the same manner.

Looking at the Z80

Firstly, I'll look at the pins that are releventto using a port.

1. Address lines 0-7.

These when decoded in the proper manner will give 256 individual ports which can be interfaced in a number of ways (more of that later).

2.Data lines 0-7.

You must have something for the computer to act on. (8 bit's of data!).

3. RD.

This becomes active on an IN A,(#nn) instruction.

4. WR.

This becomes active on an OUT A,(#nn) instruction).

5. IORQ.

This becomes active on any I/O operation.

6. M1.

This becomes active when the Z80 is actually acting on an interrupt request.

7. NMI.

When a device puts an acive signal on this line, the Z80 performs a RST 66 instruction immediately after completing its current operation.

8. INT.

Depending on the mode of operation, when a device places an active signal on this line the Z80 will execute the appropriate sub routine for servicing the interrupt. This can be disabled by the assembler mnemonic DI.

NEXT UP
The NON-MASKABLE INTERRUPT

A Guide to Better Programming

Part II

By Alan Hamilton

In the very early days of computing it was with hex numbers that programmers used to program the machines and it soon got the label of Machine Code. However, humans are fallable and Machine Code was quickly regarded as naff and we wanted something a bit better. Time for a bit of jargon: Machine code is regarded as a **low level language**.

OK, so we don't like low level languages and in response to the pressure of the more human programmers, the great-grandparent of the newer languages was born. Take a bow please, FORTRAN!! Fortran was the first high level language since it contained letters and words rather than numbers only and it wasn't long until the patter of tiny feet were heard and along came BASIC, the most popular computer language in the world.

Programmers now had some means of writing programs in something which nearly corresponded to what they were thinking, but the computers still had little switches. And so it is to this day that every computer program which we write in no matter what language we happen to write a program in, be it BASIC, Pascal, Fortran, COBOL, Pilot, Prolog, Ada, Modula 2, C, Algol, BCPL, PL/I, Assembler, you name it, everything gets turned eventually into machine code and ultimately to binary.

This converting of our high level language into machine code (and then binary) can happen in two stages.

(1) Interpretation

(2) Compilation

Taking the second of the two first (and why not!), compilation is by far the best way since it involves the program being turned into machine code before it is ever run which means that if we run a compiled program we are really running a machine code program. Languages like Pascal, C and COBOL are all compiled langauges as are some of the newer BASICs (sadly though, not the MTX BASIC).

Interpretation is what happens when we run a program in MTX BASIC. The Interpreter (which is a machine code program) takes each part of the BASIC program and converts it into machine code and then executes that little part. When it has done that, it goes onto the next part of the BASIC program and repeats itself. All this converting takes time and it is for this reason that the speed of BASIC on the Memotech is much much slower than Pascal or Assembler.

The more complex instructions you give in BASIC, the slower the interpretation becomes and the more sluggish the whole thing is. As an example of this, those of you with Pascal compilers (like the Hisoft ROM) might like to type in both these programs to see the difference in speeds:

The BASIC program:

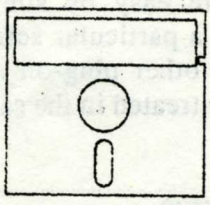
```
10 LET A=0
20 FOR B=1 TO 200
30 LET A=A+B
40 NEXT B
50 PRINT A
```

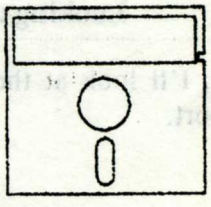
The Pascal Program:

```
PROGRAM test;
VAR a,b : INTEGER;
BEGIN
FOR b:=1 TO 200 DO
a:=a+b;
END.
```

The Pascal program runs in about one tenth of the time of the BASIC, simply because it was converted, or rather, compiled into machine code before it was run.

That's it for this month...more next time.





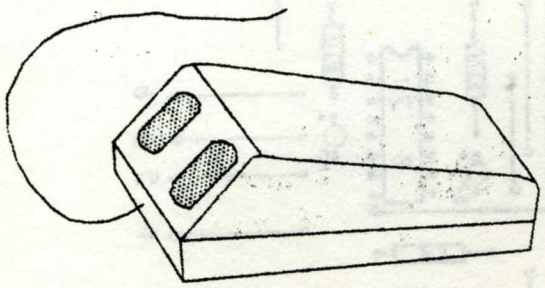
Mouse For The Memotech

Part 2

By Mike Frymyer

In this the second part, we will continue to look at the electronics involved. You will probably need to refer to part one in the Vol 5 Iss 2 magazine.

The network of gates (IC 2, IC 3 and IC 4) produce a trace as those in the bottom half of FIG 2. These are similar to those that come out of the mouse except that they are shorter and a reset pulse (at the very bottom) is inserted between the two "directional pulses".



These are then processed by two Flip-Flops (IC 5). Which ever pulse is first to clock IC 5/1 or IC 5/2 is the direction that is chosen, because, when either element is clocked the other is present with a Zero at its "D" input, effectively blocking the signal. The next pulse accepted by IC 5 is the reset pulse. This sets the Flip-Flop up for the next "directional pulse".

The next element in the circuit is a dual Monostable (IC 6), which sets up pulses of a definite duration which are then presented to the input of the parellel port.

Obviously the other half of the circuit works exactly the same for the vertical directions and the only part of the circuit left is the lines that pass the information on the push buttons. This is fairly straight forward and the only thing of importance is the &-Gate used to give a line for when both buttons are pushed. (Not necessary but it didn't cost anything as the gate was already there).

Also, on FIG 1, the pin outs (or pin designations) are given. Pin numbers are those of the 9 pin D-type connectors and the other labels are the designations of the mouse and the parallel port.

Bits 0 and 1 are the horizontal directions left and right respectively.

Bits 2 and 3, Up and Down

Bits 5,6 and 7 are the push button lines.

FIG 3. shows the component overlay for the construction of the interface with the situation of the links, Resistors, Capacitors and ICs. I used Ic sockets in the original project because it makes it easy to test the circuit prior to actually installing the ICs.

The wiring of the D-type connectors is depicted on both FIGs 3 and 4. BE paticularly careful with pins 1 and 2 on both plugs.

Getting those two the wrong way round or shorted together will cause problems. The computer will not necessaily blow up (why takes chances) but it could damage the interface. Anyway, nobody likes to see time and money go down the drain.

The circuit can be tested to an extant when the D-type plugs are all connected up. Once these are in the interface can be tested with a multi-meter. By testing the 5V and 0V rails using the Ohms scale. A reading in excess of 100 Ohms is typical with the IC's out of circuit. If this is right then plug the board into the computer, still without the IC's in circuit. Test the voltage between Vss and Vdd pins of each IC position. If 5V is present in the right polarity then remove the interface from the computer and place all of the IC's in their appropriate positions (being careful to get the orientation right).

Next Month ....

The program to type in and test it.

FIG. 3

MOUSE INTERFACE

COMPONENTS :

RESISTORS

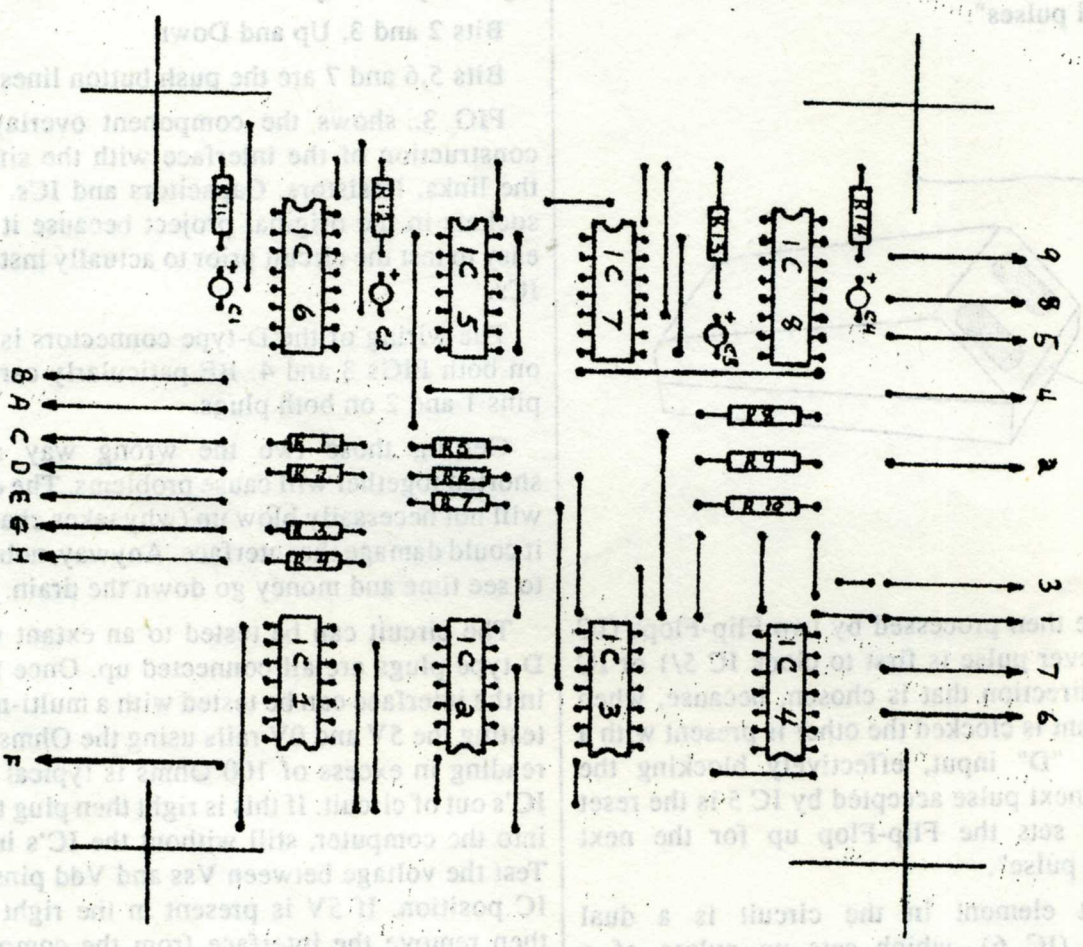
$R_1 - R_4$	560 Ω
$R_5 - R_{10}$	4K7
$R_{11} - R_{14}$	10 K

CAPACITORS

$C_1 - C_4$	3.3 µF TAG TANTALUM

INTEGRATE CIRCUITS

IC1 - 7414	IC5 - 74LS74
IC2 - 7408	IC6 - 74LS123
IC3 - 7486	IC7 - 74LS74
IC4 - 7400	IC8 - 74LS123



COMPONENT OVER LAY DIAGRAM OF MOUSE INTERFACE (EXCLUDING LINKS)

WORD PROCESSOR COMMANDS

FORMATTING:

Dot message.....	..
Justifying on/off...	^OJ
Omit page number....	.OP
Page break.....	.PA
" length.....	.PL n(66)
" no.....	.PN n(1)
" no column.....	.PC
" offset.....	.PO n(8)
Carriage return.....	RET
Align paragraph.....	^F6

MARGINS, LINES, TABS:

Centre line.....	F3
Line spacing set....	^OS
Margin release.....	F4
" set left.....	^F1 (1)
" set right....	^F5 (65)
" bottom.....	.MB n(8)
" top.....	.MT n(3)
Tab stop clear.....	^ON
" " set.....	^F7

CURSOR:

Character left.....	
" right.....	
Line down.....	
Line up.....	
Word left.....	^A
" right.....	^F
Tab.....	TAB

SCROLLING:

End of document.....	^QC
Start of "	HOME
Screen down.....	F8
" up.....	F7

ERASING:

Undo.....	CLS
Character left.....	BS
" right.....	DEL
Rest of line.....	EOL
Whole line.....	^F3
Word.....	^F2

FINDING

Find.....	^F4
Find and replace....	^F8
Find/replace again..	^L

(Default-forward from cursor position)

PRINTER CONTROLS

Underline.....	^PS
Bold print.....	BRK
Pause.....	^PC
View on/off.....	^OD
Resume printing.	F8

SAVING, PRINTING, LOADING

Quit.....	F6
Save,resume edit	F1
Save,then quit..	F5
Print,then edit.	F2
Read another file at cursor.	^KR

FLAG CHARACTERS:

< Line ended with <RET>.
+ Long line extending beyond the screen.
^ Document ends above this.
P Page break
? Line begins with a . The line won't be printed.
. Command changes screen and printer formats.
: Command changes print format only.
1 Command works at document start.

DECIMAL TAB STOPS:

Use ^F7 and ^ON for on/off.
Precede column no for tab by £ (this will be shown.
6 decimal tabs maximum.
Must use TAB key to move cursor to position.
Must allow enough space to left of tab or tab stop will be turned off.

NOTES:

(1) .commands must have . in column 1, case is optional, must have a space after command and before number.
(2) Document name is 8.3 alphanumeric characters,-,/,£.
(3) Default values shown in brackets
(4) Status line shows capacity. When ---- is ===- close and save document. Also shows Page,Line,Column.
(5) Ruler line shows ! for tab positions and L for text edge.

INITIAL PROCEDURE:<ESC> B,1: USER RUN "NW": .OP, .PL72, ^F5, ^OJ
TO FINISH: ^F5 65, <HOME>, F3 for headings, ^F6 at each para, F2

Data Bus
TECHNICAL
SPECIFICATION

MEMOTECH
MTX
SERIES

E.PARDO 86/87