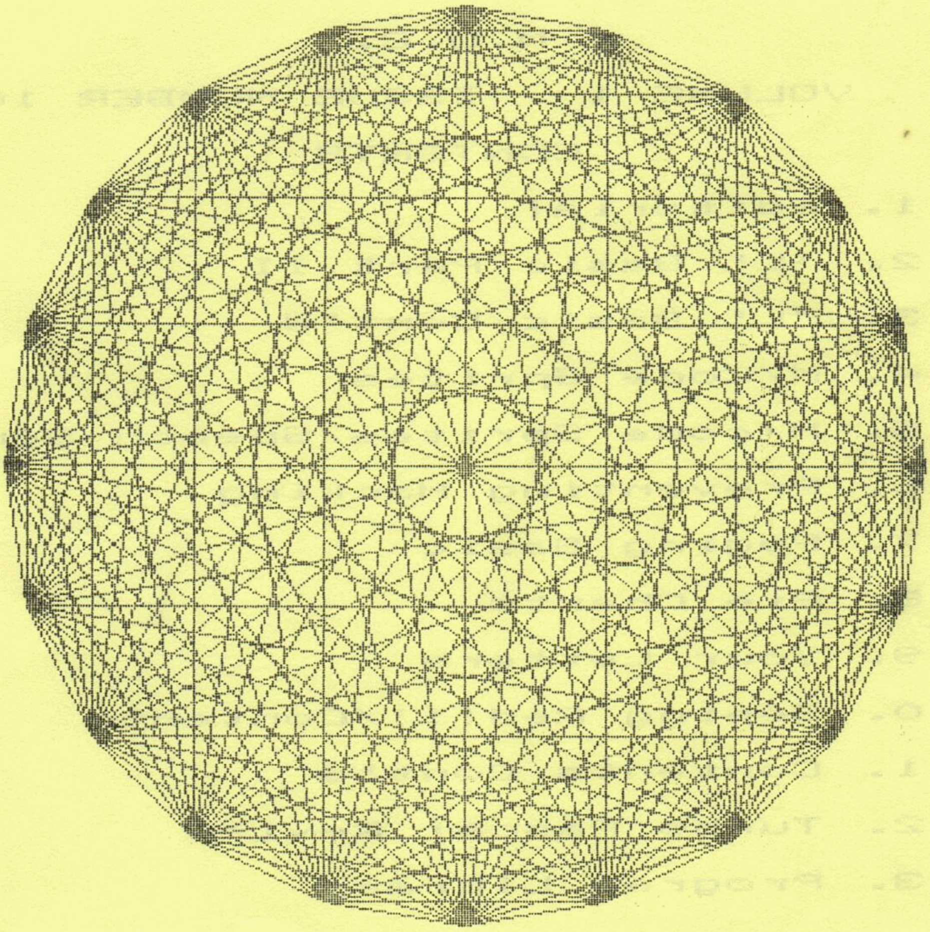
VOL. 1 ISSUE 10 JULY 1985

THE MEMOTECH OWNERS CLUB MAGAZINE



WE'RE ONE YEAR OLD

PUBLISHED BY MEMOTECH OWNERS CLUB
23 DENMEAD ROAD
HAREFIELD SOUTHAMPTON

CIRCA ...258

M.O.C.

VOLUME 1 ISSUE NUMBER 10

CONTENTS

o o o O O O o o o

EDITORIAL (July 1985)

Phil Eyres
23 Denmead Road
Harefield
Southampton
SO2 5GS

Richard Adams
18 Nightingale Rd
Pilands Estate
Bursledon
Southampton

Since last month when we published a letter about Tricom we have had many letters from members stating that they had been 'conned' in exactly the same way, they had sent money some time ago and received nothing in return. We thought that this was a really 'rum' deal, so, we made a few phone calls on your behalf. We came up with this new address :-

Tricom
31 Warnerford Road
Cowley
Oxford
Tel.0865 248208

If you write to this address or phone them you should hopefully get a response from them. Ed->Don't hold back either!, give them a really hard time, I usually find threatening them with court action is a good bluff.

Now onto a much nicer note! we are one year old with the release of this magazine, I would therefore like to take this opportunity to thank everyone for supporting us, especially those who have been with us right from the very shaky beginnings. We obviously look forward to the start of a new year and hope that everyone due for renewals will join us again for what I hope will be a very interesting year. Those due for renewal should find a form in with your magazine.

Also on another good note, Richard Dennis has received £18 for an article published in PCW (June), we originally published the LPANEL routine and passed it on to PCW for national publication. At present we have a couple of other articles being considered by them, so hopefully we will see a bit more cash in the near future.

The winner of this months prize, drawn from those who have contributed in some way to the magazine is :-

David Glover of Larkstone, Kent who wins a copy of Obloids.

Davids contribution, a magazine index for all previous magazines is something I have been meaning to find time to do for ages. I hope you like it, I've found it really useful.

Thanks also to everyone who has sent articles in for publication, if you have not seen it published yet, don't give up hope, I promise I will publish it some time, but have to try and get a balance of Basic/Assembler/News/Reviews/etc in order to make sure that everyone has at least something of interest. If anyone has a Basic program of interst about 1or2 pages long, please send it in, we are a bit short of Basic 'stuff' for beginners at the mo.

Oh!, while I'm thinking, Level 9 sent me some info last week about a new software release called Red Moon, it is their first adventure using the theme of magic as a major entity. They claim it to have over 200 locations and a fascinating story line. All for a price of £6.95. ...A pity there's still no graphics on the MTX version. If I can get hold of a copy I will have it reviewed for next months magazine.

Memotech have sent us new Hardware price lists which took affect from the 14th of July. See page 6 for details of the new prices.

Our membership incentive is still running, ... any one enroling a new member will have his/her membership extended by <u>TWO</u> issues!!!

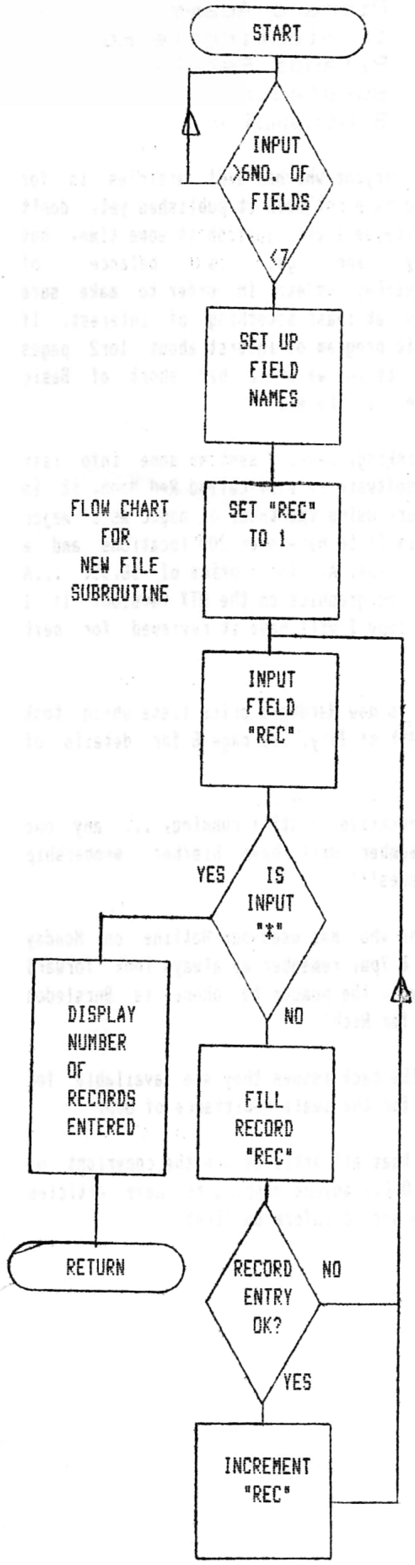
Thanks to everyone who has used our Hotline on Monday evenings between 6 & 7pm, remember we always look forward to hearing from you, the number to phone is Bursledon (042121) 5489. Ask for Rich!

If anyone would like back issues they are available for all past magazines for the small remittance of 80p.

It should be noted that all articles are the copyright of the sender and M.O.C., anyone wishing to have articles published elsewhere should inform us first.

L BASIC PART II

"WHAT'S FOR STARTERS"



FLOW CHART
FOR
NEW FILE
SUBROUTINE

This month I have flow-charted and written a small subroutine to take care of the task of Opening a 'New File' for our DBase program. I have kept everything as straightforward as possible in order to make it understandable for beginners. Have a look at the flow chart, you should find it easy to follow.

I feel that it is necessary to understand a couple of technichal terms. There are two that come to mind, Field & File :-

What are they?

"Well, a file could be made up of separate items ie Name, Address, Town etc, each separate item is termed a Field. A File is therefore a collection of Fields."

I have drawn up the Flow chart on the left and made a program from it, below is a description of the program and where to insert it in the Main Menu.

The program overleaf should be inserted into last months Main Menu routine at line 3000 onwards. The New File routine requires two more variables to be DIMensioned :-

(See Line 50 Overleaf)

FNAME$(6,20) & REC(1)

FNAME$ is a string variable (ie it is capable of holding words!), it will hold the names of the fields that you define at the beginning of the new file subroutine. It is at present set to 6 Field NAMES each having a maximum of 20 charcaters "FNAME$(6,20)".

REC is a numeric variable (ie it can only contain numbers!). It will be used to keep a count of the number of RECords in the data file FILE$, dimensioned earlier in the Main Menu routine. Remember FILE$ is set to hold 100 records of 6 fields, the contents of each field is a maximum of 30 characters.

Remember!! These are only numbers that I have set, If you would like more field names then you will have to alter FNAME and FILE to suit your requirements. If you make the file larger, ensure that it will fit in memory, I reckon that there will be enough room for 20K of data on a 32K machine

I have kept the routine simple and hopefully easy to follow, there is no reason why you cannot expand it to make the display more appealling, by adding colours and more information and perhaps some more error traps. ie to ensure that the field names do not go over 20 characters in length and to ensure that data entered in the Fields does not go over 30 characters. Either situation will cause the program to crash!!!. Use the LEN(x$) command to do the error traps.

The program is overleaf along with a desciption of how it works. As always, I am interested to see any updates that you make.

Phil

```
10 REM NEW FILE SUBROUTINE FOR DBASE PROGRAM  PUB.VOL1 ISSUE 9
20 REM BY PHIL EYRES 3/7/85
30 REM DIM STATEMENT BELOW TO BE ADDED TO MAIN MENU ROUTINE
40 REM
50 DIM FNAME$(6,20),REC(2)
60 REM
70 GOSUB 3000
2999 REM NEW FILE ROUTINE
3000 VS 5: CLS
3010 LET POS=5
3020 CSR 5,3: INPUT "NUMBER OF FIELDS (6 MAX.) ";FIELD$
3030 LET FIELD=ASC(FIELD$)-48
3040 IF FIELD<1 OR FIELD>6 THEN GOTO 3000
3100 CLS
3110 FOR I=1 TO FIELD
3120 CSR 3,3: PRINT "NAME OF FIELD ";I;" ";: INPUT FNAME$(I)
3130 CSR 17,3: PRINT "                    "
3140 CSR 3,POS: PRINT "FIELD ";I;" ";FNAME$(I)
3150 LET POS=POS+1
3160 NEXT I
3170 CSR 13,15: PRINT "PLEASE WAIT": PAUSE 3000
3200 LET REC=1
3210 CLS
3220 CSR 10,3: PRINT "INPUT RECORD";REC
3230 LET COUNT=1
3240 IF COUNT<=FIELD THEN CSR 5,5+COUNT ELSE GOTO 3290
3250 PRINT FNAME$(COUNT);: INPUT W$
3260 IF W$="*" THEN GOTO 3400
3270 LET FILE$(REC,COUNT)=W$
3280 LET COUNT=COUNT+1: GOTO 3240
3290 CSR 5,13: INPUT "FILE OK?(Y/N)";Y$
3300 IF Y$="Y" THEN GOTO 3320
3310 IF Y$="N" THEN GOTO 3210 ELSE GOTO 3290
3320 LET REC=REC+1
3330 GOTO 3210
3400 CLS
3410 CSR 5,5: PRINT REC-1;" FILES ENTERED"
3420 PAUSE 3000
3430 RETURN
```

Listing Of New File Routine

Program Breakdown

3000	Set & Clear Text Screen
3010 - 3040	Input Number Of Fields
3100 - 3170	Input Field Names
3200 - 3330	Input Files (Start At File "REC" 1)
3400 - 3430	Display Number Of Files Entered And Return To Main Menu.

MACHINE CODE SPRITE DEMONSTRATION

The following article was sent to us by Nicholas Hill of Godalming, who was prompted into action by the series of VDP (Video Display Processor) articles published by us a couple of months ago.

If you want to write arcade type games with swift action, then you want to move sprites about the screen fast. Basic is often to slow. This is an example of how it can be done - it gives you a skeleton on which to build.

Despite the title, it is easiest to begin the program with a couple of lines of Basic. ..There is no point in making work for yourself!! So :-

```
1 VS 4: INK 15: PAPER 1: CLS
2 GENPAT 3,1,28,8,73,127,73,8,28,0
3 CTLSPR 2,1: CTLSPR 6,0
```

Then ASSEM 10 and type in the code as listed below. Use the four curser keys to shift the sprite - try moving it diagonally as well.

This seems very much like the program on the denomstration tape, but has far more applications when developed. These include :-

1. The program can be made very much faster - try taking out the HALT instructions, ...your sprite is now just a blurr.

2. You can extend the COPY and UPDATE routines to take in as many sprites as you like. Increase B by another 4 for each sprite at the end of the VRAM routine.

3. You can change the position, pattern and colour of the sprite. The first two bytes are the y and x positions (note 0,0 top left of screen).

Cont'd Overleaf

```
10 CODE
          LD HL,TABLE
          LD DE,£E000
          LD BC,4
          LDIR
          CALL UPDATE
          LD IX,£E000
LOOP:     CALL COPY
          CALL KEY
          CALL UPDATE
          HALT
          JR LOOP
KEY:      LD A,£F7
          CALL KKEY
          CALL Z,LEFT
          LD A,£EF
          CALL KKEY
          CALL Z,RIGHT
          LD A,£FB
          CALL KKEY
          CALL Z,UP
          LD A,£BF
          CALL KKEY
          CALL Z,DOWN
          RET
KKEY:     OUT (5),A
          IN A,(5)
          CP £7F
          RET
LEFT:     DEC (IX+1)
          RET
RIGHT:    INC (IX+1)
          RET
UP:       DEC (IX+0)
          RET
DOWN:     INC (IX+0)
          RET
VRAM:     LD A,(£FF58)
          AND A
          JR NZ,VRAM
          DI
          DEC A
          OUT (2),A
          LD A,B
          OUT (2),A
          LD HL,£DFFF
          LD B,5
          RET
UPDATE:   LD B,£7E
          CALL VRAM
U1:       LD A,(HL)
          INC HL
          OUT (1),A
          DJNZ U1
          EI
          RET
COPY:     LD B,£3E
          CALL VRAM
C1:       IN A,(1)
          LD (HL),A
          INC HL
          DJNZ C1
          EI
          RET
TABLE:    DB 87,127,1,15
```

Cont'd From Previous Page

the third is the pattern number (as defined by GENPAT), and the fourth is the colour (0 to 15). Notice that I have used the IX register, so that to access the colour would use an instruction such as LD (IX+3),9.

4. By using other sprites you can now move them and check for collision by comparing their X and Y positions.

Now for how the program works :-

1. The first 5 lines put the correct values in place (£E000) to be occupied to the VRAM.
2. IX is then loaded with the value in RAM that the sprite values will be occupied too, altered and returned.
3. LOOP copies the video ram, looks for the cursor keys, updates the VRAM, pauses then repeats.
4. KEY tests for the four cursor keys, with KKEY actually scanning the keyboard.
5. LEFT, RIGHT,UP and DOWN change the X and Y values of the sprite as copied into RAM. Note I have not used the ROM routine at £0079 - this is far to slow in an arcade type game.
6. VRAM is common to COPY and UPDATE - it sets up the address in VRAM of the sprite area (notice that the high byte is £3E to read and £7E to write). The point of the first three lines of VRAM is to check that it is safe to write to the screen.

Thanks Nick!!!

---oooOOOooo---

SPEECH SYNTHESIZER

The kit is now available, exclusively through the club, the kit is supplimented by two programs, the first is a short routine that will output data to your SP through the internal port. This program can also be used to some extent to prove your newly built circuitry. The second program is a fairly comprehensive Speech Data Editor which utilises the Basic command DSI to great effect.

You can of course connect your SP to any port providing you have 7 output bits and 1 input (a handshaking line would be OK for the input line!). Should you wish to use the internal input/output port we can supply a connecting cable that will do the job. N.B. The building and operating instructions assume the use of the internal port.

!!!Prices!!!

Speech Synthesiser Kit	-->	£18.00
Connecting Cable (For internal Port)	-->	£4.50
LED Kit (Requires Conn. Cable)	-->	£6.95

All prices are fully inclusive. Please allow 14 days for delivery and make cheques payable to MOC.

PREVENTING HACKING

Carrying on from last months "What To do When You've Pressed Reset" we have a second article on how to prevent people Hacking your games. This article should be read in conjunction with last months article as I feel that this one will not make a lot of sense without the other one.

Any autostart program written in the format :-

```
  1 GOTO 100
 10 SAVE "PROG"
100 CODE
```

can very simply be broken into and repaired in the way described last month. It is, however a very convenient format in which to develop and write machine code programs. What made it simple to repair was that LIST told you what to look for and PANEL made it easy to change. A very simple protection system can be used to make life a lot more difficult.

1. Write and fully test your program.
2. ASSEM 100

Go into insert mode (in PANEL) and add a short section at the start of the program to overwrite the "FF" line terminators at the end of lines 1 and 10.

3. GOTO 10 to save the autostart version.

When the program is loaded it runs starting at the CODE statement and overwrites the line terminators. After a reset and POKE 64167,1 the LIST command is now unable to find the start of lines 10 and 100 and also displays rubbish on the screen. However, there are still two giveaways :-

(i) The line numbers
(ii) The general length of the first two lines

To make it more difficult use unusual line numbers like 15087 and 25613 and include a long unmeaningful REM after the GOTO. Preferably containing function keys as these look like Basic tokens.

If you break into TOADO as described in last months article, it is worth looking at the method used to start to allow the program to run on both the MTX 500 and the MTX 512. You must include a routine of this type if an assembler program is to run on both machines. If you then LIST this program and look for the start of the main routine, you will find that this is disguised because the disassembler is trying to translate data blocks as assembler mnemonics. Further investigation is only possible using PANEL or ASSEM. This is also a helpful protection tip, as these facilities don't support listing to a printer.

Ed-> This type of 'Hacking' really only applies to early continental games, for instance, Megastar have done some really nasty things to ensure you don't hack their games, ie if you try and 'get into them' in the way mentioned above, you will very soon realise that you can not use your Panel command as trying to do so only results in a complete system reset. In fact I believe I am correct in saying that all Megastar games start at £8000 regardless of machine, they also remain intact on machine reset, the big problem is that they have mucked about with the system variables something cronic.

* *

NEW HARDWARE PRICES

As from the 14th of July Memotech have reduced the prices of Hardware items to:-

MTX500 £175 MTX512+SDX 250K+80COL PCB+CP/M*NW+SC
MTX512 £199 ONLY £499
MTX RS128 £299 FDX 2*500K CP/M £749
NEWWORD OR PASCAL £50
32K MEMORY EXTENSION £39.95 256K SILICON DISC £165
64K MEMORY EXTENSION £49.95 512K SILICON DISC £275

PRINTER PRICES REMAIN THE SAME AT £269 + £15 FOR CONN. CABLE
(CLUB PRICE FOR DMX80 PRINTER STILL ONLY £250 INCL. CONNECTOR)

PAGING BASIC

To understand how Basic uses Paging it is necessary to understand the following :-

A line of Basic held in memory is split into three sections.

1) The first two bytes contain a Hex number representing the total length of the line.
2) The next two bytes contain the line number. This is why the maximum line number that can be used is 65535 or £FFFF in Hex.
3) Basic starts from byte 5 and ends with a £FF in the last byte of the line.

The following system variables are used.

1) £FA7A, number of 32K RAM pages present.
2) £FAA7, top of Basic in current page.
3) £FAA9, current Basic page number.
4) £FAAC, a list containing the top of each Basic page.

When Basic is trying to find a line number the sequence of events is as follows. It starts by setting a pointer to the start address of the first line and reads the line number, if this is smaller than the number it is looking for it then adds the length of the line to the value of the pointer to find the start address of the next line. It then compares this address with the address of the top of Basic in the current page. If the pointer address is less than the top of Basic it then tests the next line number. If the pointer address is equal to the top of Basic a test is then made to see if it is the last page. If it is then Basic has reached the end of the program, if not then the next page is switched in and the pointer is set to the start of this page. The system variable £FAA7, top of Basic in current page, is updated with the value for the new page taken from the list of page tops and the current page number is incremented by one. The sequence continues until Basic finds the line number, finds a line number that is larger than the number it is looking for or reaches the end of the last page.

!!! Many Thanks To John Hodgson !!!

Ed-> Bearing in mind what has just been stated above, if you can take your minds back to page 6&7 Issue 7 and Paul Schofields article about the SKI II program, everything he said should come together.

For those who do not have Issue 7 Pauls article was mainly about how to speed up your Basic programs, his technique was to place the subroutines at the beginning of the program in order of most frequently called, or, those requiring fastest execution first. Then putting the main part of the program that was used to call the subroutines below.

PIE CHARTS

By

Mike Pike

It does not seem possible to produce multi-coloured pie charts on the MEMOTECH without the "adjacent colour interference" problem pointed out by Gerald Buzzacott on page 8 of issue 4. However, I have produced a subroutine which draws respectable pies using 2 colours alternately thus side-stepping the problem. It will accept any data, although problems may arise with the display if there are small sectors adjacent in the pie chart.

```
1 REM EXAMPLE PROGRAM
5 READ N,T$
10 DIM P(N),D(N),N$(N,8)
20 FOR I=1 TO N: READ D(I),N$(I): NEXT
30 GOSUB 80
40 GOTO 40
78 REM PIE CHART ROUTINE
79 REM BY MIKE PIKE 3.7.85
80 LET TOT=0: FOR I=1 TO N: LET TOT=TOT+D(I): NEXT
90 LET P=0: FOR I=1 TO N: LET P=P+D(I)/TOT*100: LET P(I)=P: NEXT
100 VS 4: COLOUR 0,4: COLOUR 1,15: COLOUR 2,5: COLOUR 3,4: CLS
110 CIRCLE 128,96,44
120 ANGLE PI/2
130 LET S=1: LET F=0: LET ISTO=0
140 CSR (33-LEN(T$))/2,1: PRINT T$
150 FOR I=0 TO 100 STEP .25
160 IF I>=P(S) THEN GOSUB 300
170 PLOT 128,96
180 DRAW 43
190 PHI -PI/200
200 NEXT I
210 ATTR 2,0: PLOT 128,96
220 RETURN
300 LET F=1-F: ATTR 2,F
305 LET TH=(I+ISTO)/100*PI
310 LET CR=(I+ISTO)/2
320 CSR 16+SIN(TH)*7+7*(CR>50),12-COS(TH)*7
325 PRINT N$(S): LET S=S+1: LET ISTO=I
330 RETURN
1000 DATA   6,CAR COSTS
1020 DATA  1123.45,PETROL
1040 DATA  23.10,OIL
1060 DATA  278.98,SERV+REP
1080 DATA  185.34,TAX+INS
1100 DATA  63.25,MISC
1120 DATA   127.50,PARKING
```

<u>VARIABLES</u> <u>AND</u> <u>ARRAYS</u> for subroutine

On Entry :
N=Number of data items
D(N) holds data values
N$(N,8) holds data names
T$=Title of chart
On Exit:
Data values unaltered
Local:
TOT=Total of all data values
I=loop control
P=Cumulative Percentage
P(N) holds "P" for each item
S=Sector number
F=Flip Flop for plot attribute
ISTO holds value of I at last sector boundary
TH=Angle in radians of sector midpoint
CR=Sector midpoint in terms of "I"

<u>PROGRAM</u> <u>NOTES</u>

Line :-

140 Prints the title centre screen.

150 "I" holds the percentage of the pie drawn so far. The STEP frequency is just sufficient to avoid any "missing pixels" at this radius.

160 Detects the sector boundaries

170-190 Draws a radius and moves 1/400th of a pie c/wise

210 Ensures centre is plotted (it looks neater) and ATTR 2 is switched off before RETURNing.

300 The plot/unplot attribute is switched over at each sector boundary giving first foreground then background colour.

320 Prints name of each data item in an appropriate position, sets next sector and updates "ISTO".

NB. The boolean expression in parentheses is evaluated as "-1" if true and "0" if false. Do not believe everything you read in the manual!!(See page 140) Thus if CR>50, ie the name has to go to the left of the pie, then 7 is subtracted from the column number of the cursor position to leave room for the length of the name itself.

YOUR LETTERS

*** Games High Score Table ***

TOADO	107549*	N.GOODING	BLOBBO	71233	T.PICKSTONE
NEMO	11080*	P.CRIGHTON	OBLOIDS	60040	M.GELDER
P.PETE	39630	A.DOBSON	MISS.ALPH	43840	T.PICKSTONE
KILOPEDE	33440*	P.CRIGHTON	GOLDMINE	6025*	P.CRIGHTON
CONT RAID	10810	M.GILL	STAR COMM	90410*	P.CRIGHTON
MAXIMA	252830	M.GILL	TURBO	23030	M.GELDER
QOGO 2	107560	A.DOBSON	ASTRO PAC	69390	A.DOBSON
COBRA	1718	A.DOBSON	SNAPPO	62960	A.DOBSON
T FIGHTER	2350*	N.CRIGHTON	S M/FIELD	829*	P.CRIGHTON
ASTROMIL.	3070	A.DOBSON	PHAID	1965	A.DOBSON
SON OF PETE	880	A.DOBSON	F.DEEP	1290	A.DOBSON
S.SCANNER	1590	A.DOBSON	ICEBERG	17431	A.DOBSON
KNUCKLES	488650*	P.CRIGHTON			
FELIX	18450	T.PICKSTONE			
TAPEWORM	168515	A.DOBSON AT LEVEL 1			
	150500	A.DOBSON AT LEVEL 9			
BOUNCING BILL	94340	A.DOBSON			
SNOWBALL	450	P.CRIGHTON			

* Denotes new high score.

******

Hints & Tips

1. Just a little program that will alter the INK and PAPER colours for NODDY or ASSEMBLER. Replace x and y with the required colours. NODDY can be replaced by PLODx$ or ASSEM n

```
10 FOR F=1 TO 4: READ A: VS A:PAPER X: INK Y: CLS:  NEXT:
   NODDY
20 DATA 0,1,5,7
```

P.Crighton

2. Has anyone noticed that the assembler won't allow you to enter the command : BIT 6,(HL) .It will allow you to enter the commands to test all the other bit's of (HL) but not bit 6. It is easily overcome, however, by entering : DB £CB,£76 instead, which is the op. code for the instruction.

N.R.GOODING

******

Micheal Gelder sends in this addition to issue 8's Hacker's Guide to Pot Hole Pete :-

The following addition is required to make screen 27 possible, add it as described in issue 8.

```
LD A,£95
LD (£EA1A),A
```

In finding this information I discovered how the screen data is stored. Using the following information it would be possible to customise screens. For example replacing marshland with firm ground.

The address for the start of the data for each screen is located from £858D onwards, e.g. the address' for screen one and two are stored in the four bytes at £858D-£8590 like this :-

```
        £858D=50 C3 F0 C4 (Hex)
```

Therefore the address for screen one is £3C50 and for screen two is £C4F0

From this address onwards there are three sections of data. The first describes the screen set up, the second holds the sprite information and the third holds the screen title. I have only managed to work out the first section. This consists of groups of four bytes such that :

byte 1 = x position
byte 2 = y position
byte 3 = no. of characters
byte 4 = ASCII code of character

So, 00022094 (Hex) means a row of 32 bricks starting at 0,2. The character codes are :-

81=ladder   95=Firm Ground   98=Plant
93=Conveyer Belt96=Square   99=Marshland
94=Brick   97=Icicles   9A=Exit

Before you start hacking you have to break into the program. To do this refer to issue 8 and the Hackers Guide. Before typing <CLS><RET><CLS><RET> delete the command JP £8100. Then add the following line :-

```
 9 STOP
```

Type run and start the tape playing again. When it has loaded, enter Panel, and you're away!!

******

Questions And Answers

1. How do I get Joystick movement in 8 directions as opposed to four?

Ed-> I'm afraid that this is not an easy thing to accomplish, the easiest way that I can think of would be to build yourself a joystick that has sensors in 8 or more positions, then build a bit of controlling hardware that would convert any diagonal movement signal into something that your joystick port could handle. As you can see a really simple task! Even so, quite an interesting little project.

2.Could you tell me the address' of shops that stock Memotech equipment in the London area.

Ed-> I'm afraid Memotech dealers are few and far between, as yet, we are unable to get anything from Memotech on this subject.

Seeing Red!!!

This is mainly about what to do if you've got 15 minutes or so on your hands. Well, from experience I can say that probably the worst thing that you can do is switch your computer on. This usually proves quite fatal and ends up with you totally in "The Dog House". I'm afraid that over the past couple of weeks this is where I have spent most of my time, why?, well, summer has just about arrived, and as usual all the old jobs crop up, the house needs painting, the car needs just about everything doing to it and everyone wants to go out for the day!.

The point I'm trying to make is ... I've had another fatal brainwave. For sometime now I have dreamed of being able to digitise a sheet of paper into memory so that I can then do some wonderful things with it. I've spent many hours looking through computer mags. wishing that I had the sort of money that they want for such an item. Then... the other day I thought why not make an infa-red detector, 'bolt' it on to an old printer ribbon cartridge in such a way that it would scan a page for me.

So far I have managed to get it to scan a line, after a great deal of trouble in making the print head move and scan the line with my infa-red detector at the same time, this was eventually achieved through assembler, Basic appears to wait until the line is completed before returning to program control. My remaining problems are increasing the sensitivity a little to make it see a single vertical line and making a suitable program to actually digitise a whole page into memory.

So by the look of it I've got a few more weeks in the dog house yet!!

Phil

****

MAGAZINE INDEX

By DAVID GLOVER

This month sees the addition of a long awaited for magazine index. Thanks are entirely to David Glover for this excellent contribution. It is hoped to make this a regular spot in the magazine from now on, printing it every two or three months, so we can all keep track of those important past articles.

This index is maintained by David Glover using Datafile.It is not claimed to be perfect, but hopefully is of use in tracking down those sought after or half remembered items.

COLUMN HEADINGS :-	GROUP CODES :-	TYPE CODES :-
G - Group	A - Utilities	1 - Article
T - Type	B - Games	2 - Problem
NUMB - Unique number given to each entry	C - Graphics/Screen	3 - Hints/Tips/Problem Answers
DESCRIPTION - Obvious	D - Sound	4 - Review
V - Volume	E - Programming	5 - BASIC program/routine
I - Issue	F - Hardware	6 - Pascal program/routine
PA - Page		7 - Assembler program/routine
CROS - Number of related Entry		

CONT'D OVERLEAF

S	TYP	NUMB	DESCRIPTION ...	V	I	PA	CROS
A	2/3	0003	PANEL M(ove) command	1	1	6	
A	2/3	0005	BASIC VERIFY breakout	1	1	6	
A	2	0020	Printing out double width graphics	1	2	8	
A	2	0043	PASCAL compiler option $F	1	4	3	
A	2	0058	Neword embedded printer ctrl commands	1	6	7	0066
A	2	0073	Date to week day conversions	1	7	9	0087
A	3	0066	Neword embedded printer ctrl commands	1	7	8	0058
A	3	0082	CSEN sprite generator correction	1	8	6	
A	4	0011	Brunword	1	1	10	0023
A	4	0023	Brunword	1	2	10	0011
A	4	0036	Utilities	1	3	11	
A	4	0077	Datafile	1	7	11	
A	4	0081	Simplex Tableaux	1	8	6	
A	5	0019	Screen Dump Routine	1	2	7	
A	5	0024	Clock	1	3	2	
A	5	0044	Drawing Board	1	4	10	
A	5	0061	Display used variables	1	6	7	
A	5	0063	Nodprint	1	7	2	
A	5	0087	Date to week day conversion	1	8	10	0073
A	5	0101	Convert CSR positions to PLOT positions	1	9	9	
A	7	0037	Keyboard Debounce	1	4	3	
A	7	0039	PANEL LPRINT routine	1	4	6	0086
A	7	0079	Typewriter program - uses ROM o/p routine	1	8	5	
A	7	0080	Clock program - uses interupts	1	8	6	
A	7	0086	PANEL LPRINT routine extensions	1	8	10	0039
A	7	0098	Bubble Sort	1	9	7	
B	2	0031	Dungeon Adventure - Gem gathering	1	3	8	
B	2	0100	Tri-com problems	1	9	9	
B	3	0015	Alice hints: ++ getting out of forest ++	1	2	7	
B	3	0083	Hackers guide to pot Hole Pete	1	8	9	
B	3	0084	Alice/Zoo help line.	1	8	10	
B	4	0010	Maxima	1	1	9	
B	4	0022	Cobra	1	2	9	
B	4	0034	Felix in the Factory	1	3	9	
B	4	0045	Obicics	1	4	9	
B	4	0049	Charmkatz	1	5	6	
B	4	0050	Tumbledown Tower	1	5	6	
B	4	0074	Emerald Isle	1	7	10	
B	4	0075	Son Of Pete	1	7	10	
B	4	0076	Escape From Zarkos	1	7	11	
B	4	0088	Surface Scanner	1	8	12	
B	4	0089	Fathoms Deep	1	8	12	
B	4	0091	Lords Of Time	1	3	9	
B	4	0093	Bogo2	1	7	10	
B	5	0035	Farmer	1	4	2	
B	5	0065	Ski II	1	7	6	
C	2	0042	Avoiding adjacent distorted colours	1	4	6	
C	2	0051	Loss of character off screen	1	5	7	
C	3	0006	Sprite coincidence flag	1	1	7	
C	3	0062	Screen control	1	6	8	
C	5	0002	Polygons	1	1	4	0011
C	5	0011	More Polygons	1	2	2	0002
C	5	0013	Squares	1	2	5	
C	5	0014	Spiroplot	1	2	6	
C	5	0026	Sataloids	1	3	5	

G	TYP	NUMB	DESCRIPTION	V	I	PA	CROS
C	5	0047	Lisajoux patterns & figures	1	5	2	
C	5	0057	Rotation (with sound)	1	6	6	
C	5	0078	Basic Shape Filling	1	8	3	
D	2/3	0007	Fixed tone and duration sound	1	1	7	
E	1	0001	Assem. Prog Pt 1:Z80 & Hex/Bin notation	1	1	2	
E	1	0012	Assem. Prog Pt 2:Bin/Dec & 2's compl.	1	2	3	
E	1	0025	Assem. Prog Pt 3:ROM RST10 info	1	3	3	
E	1	0038	Assem. Prog Pt 4:ROM RST10 continued	1	4	4	
E	1	0048	Assem. Prog Pt 5:VRAM memory	1	5	4	
E	1	0056	Assem. Prog Pt 6:VRAM & VDP	1	6	3	
E	1	0064	Assem. Prog Pt 7:VRAM demo & pseudo ops.	1	7	5	
E	1	0094	L Basic Pt 1: Main Menu	1	9	2	
E	1	0104	RST10 codes ..A full list!!	1	9	12	
E	2/3	0009	Using LOGO	1	1	8	
E	2/3	0071	Program Auto - Run	1	7	9	
E	2	0017	Decimal Point & other Tabulation	1	2	7	
E	2	0021	Adding Basic Commands; USER/OVERLAY	1	2	8	
E	2	0032	Cassette Baud Rates	1	3	8	0052
E	2	0072	Re-programming the F* keys	1	7	9	
E	2	0095	Hacking Early Continental Games	1	9	4	
E	2	0096	Memory mapping	1	9	5	
E	2	0102	Recovery of program after reset	1	9	9	
E	3	0018	Program memory usage	1	2	7	
E	3	0028	Program memory usage	1	3	7	
E	3	0029	BASIC input using F* keys & abbr.	1	3	7	
E	3	0030	Decimal point tabulation	1	3	7	0017
E	3	0041	RST10 Sprite params	1	4	8	
E	3	0044	Decimal rounding	1	4	8	0030
E	3	0052	Cassette baud rates	1	5	7	0032
E	3	0053	Adding basic commands & intercepting errors	1	5	7	0021
E	3	0055	Arrays	1	6	2	
E	3	0059	Decimal rounding	1	6	7	0044
E	3	0060	Emulating MTX500 on a MTX512	1	6	7	
E	3	0067	Delete character in a string	1	7	9	
E	3	0092	Print ASCII of key pressed	1	6	7	
E	3	0099	Entering assembler via RST28	1	9	9	
E	3	0103	DMX 80 User defined characters	1	9	11	
E	4	0033	Memotech computing	1	3	8	
F	1/4	0097	Speech Synthesiser	1	9	6	
F	2/3	0004	MTX RAM memory chips	1	1	6	
F	2/3	0068	Unstable VDU Display	1	7	9	
F	2/3	0069	ROM access	1	7	9	
F	2	0008	Double characters when typing	1	1	8	0016
F	2	0070	Memory Management	1	7	9	
F	3	0016	Key bounce solution 1	1	2	7	0008
F	3	0027	Key bounce solution 2	1	3	7	0008
F	3	0040	Interfacing to user port	1	4	7	
F	3	0085	FDX system problems	1	8	10	
F	4	0054	Neword Word processor ROM	1	5	9	
F	4	0090	Hisoft Pascal ROM	1	8	13	
F	4	0105	250k single disc drives	1	9	13	
F	4	0106	JUKI 2200 Electronic typwriter	1	9	14	

******

TURBO PASCAL V3.0

REVIEW

By Phil Eyres

I have now had my Turbo Pascal V3.0 for three weeks, this review is based on my opinions of it so far. It should be understood that I am not really in a position to give too techinical a review as my knowledge of Pascal is still limited.

Well, firstly I must say that I waited 12 weeks for the arrival of my Pascal, I think that I must have spent more money on phone calls than I did on the Pascal. So one quick word of warning, if your in a hurry, give Conguin Software a very wide birth.

Now on to the important subject of my new Pascal.

__Documentation__
The documentation is contained fully in one 380 page manual. As usual this is of little use to beginners, it contains the full technical workings explaining each command in precise detail. From what I have managed to take in so far there are more than enough commands to cover just about every possible need.

__The disc__
There are three Turbo programs on the disc :-
1. The Turbo Editor
2. The Turbo Instalation Program
3. The Turbo Compiler

The editor is very similar to Neword in layout and uses the same CTRL keys to control it. It varys slightly in that instead of having formatting commands it has a simple to use line-indenting formatter, giving Pascal programs, there characteristic indented format. Also on exit (^KD) the Pascal source is not saved to disc. Pascal will, however, ask if you wish to save your program should you forget.

The Turbo Installation program is now somewhat infamous in my books, it took me about 5 hours to actually install it on my machine, the install program is needed to inform Pascal of the exact format of your machines hardware, also it allows you to change the CTRL keys associated with the editors formatting functions. As yet I have not found a way of using the F keys on the key-pad.

The compiler is needed to compile you source program created with the line editor into machine code, the error detection mechanism seems very good indeed. Since the screen editor displays the area around the error and not just the line containing the error it is much easier to debug the source quickly. So far this has proved very useful for me, it seems to reliably point to the error in the source. Compilation is also very quick, especially when it is done on my new Silicon Disc drives.
Ed-> I just thought I would slip that one in!!.

Also on the distribution disc is a Pascal source code for you to compile into a very good little Spreadsheet program. Quite handy for learning how to use your compiler.

One thought that has crossed my mind is how will single disc SDX owners with CP/M copy the distribution disc onto a working disc for use? You will need a few discs with your pascal as programs tend to get quite long very quickly. One program that I have (the one that produced this months front cover) includes 2 overlays which combine to make the program some 70+k long. The 'old' 250k discs will not last long at that sort of rate of usage.

My version V3.0 is a very recent release and thus had a price tag to suit, £85 is expensive, Grey Matter (see national mags.) is probably your best bet, you should be able to get one for about £50 - £60, phone up though to make sure that they have them in stock else you may have a long frustrating wait on your hands.

__Conclusions__
I have to say that I consider my Turbo Pascal very good value for money, it compiles very quickly even when compiling to discs. I have found no problem in finding books to help with my understanding of this new language, for that matter the CP/M user group has many a disc full of programs written with turbo Pascal that are ready to run.

I recently read a review which compared an early version of Turbo against JRT pascal, all the benchmarks showed turbo to be between 5 and 50 times faster at performing the standard set of pascal tests. As a matter of interest the maths functions were 25 times faster.

****

__NEXT MONTH!!!__

Saving your data files to tape.
Miner Dick, Xaviersines' new release reviewed
Silicon Discs Reviewed
....And lots, lots more!!!

PROGRAM LIBRARY

Our Library is growing well, with four good additions this month. However the time needed to produce listings is growing, the task is made somewhat more difficult by the need to print the Noddy pages. To overcome this, Programs will only be available on cassette, as before two programs per cassette, a fee of £1.00 is charged per cassette to cover costs. I dont think that this will hinder anyone as typing in long listings is labourious and not too error free. If anyone would like programs on disc, please enclose a disc and £1.50, stating titles required. N.B. As it stands the library occupies fully 1 500K disc.

1.Basic & Assembler Programs

1.Hex-Dec-Bin Convertions. (Binary Bit In Assembler)
2.CGEN Sprite Generator.
3.3D Drawing Board. Rotate a skeleton of a cup & saucer in 3D.
4.Whist.
5.Memory Save. This Utility will Save a block of memory to tape and retrieve it.
6.MTX Drawing Board.
7.LOGO Drawing Board.
8.Simplex Tablaeux. Applications Program
9.Breakeven. Applications Program
10.Statistics **New** Applications Program
11.An Unsolved Problem **New** Applications Program
12.Radio Routines **New** Applications Program
13.Light Cycles. Arcade Game
14.Hex/Dec Dec/Hex conversions using USER commands!
15.Renumber II Renumbers Including GOTO's etc
16.RELOC Relocs Assembler Properly!!
17.Character Editor **New**
18.Quasimodo **New** Excellent Arcade Game

3.Articles From Previous Magazines

1.PANEL Utility. Makes use of system variable FEXPAND to add a hard copy option to your Front Panel.(Vol1 ISS.4)
2.PANEL2 Utility. As above but updated to include a second feature.
3.Undocumented Neword dot commands.(Vol1 Iss.7)
4.Hisoft Pascal Review (vol1 Iss.8)
5.Neword Rom Review (Vol1 Iss.5)
6.RST10 Codes Explained (Vol1 Iss.3)
7.VDP Explained Using assembler (vol1 Iss4,5,6)
8.System Variables (Not Previously Published!!)

5.Program Reviews

Quasimodo

This is an excellent game sent in by a new member, Liam Redmond. The quality of this game is really good, better than some commercial games written in m/code, so if your always saying that you cannot afford games, then try this one!. The game involves rescuing Ezmeralda from the tower, it has all the usual obstacles and levels of play, good sound and score routines and even a very well thought out high score table.

A REAL GEM!!

Statistics

The first of two new programs received from the sender of Simplex and Breakeven. If you liked the first two then you are likely to like this one, it is an Operational Research Tool for Managers (Or MTX Owners!). It consists of a suit of 5 programs testing for all sorts of things vital to the world of management.

An Unsolved Problem

This, the second program, is designed around a mathamatical problem involving sequences of numbers, the problem was origonally thought up by Rade & Kaufman. (Good old Rade & Kaufman!!!). Good for passing a few hours when you've nothing else to do.

Character Editor

Yepp!, This is another Sprite generator, it is a 'good-un' though, on the same sort of par as CGEN but goes about things differently, the program is menu driven and has some 17 options. Only allows the design of 8*8 sprites, but stringing them together is easy enough.

Radio Routines

I was really suprised by this one, it is a Noddy driven program that covers just about everything to do with Radio Electronics. If your the type that enjoys learning a little about subjects you know nothing about then this is a good place to start with Radio Electronic's. This program has some 7 Main Menus and caters for everything, from Ohm's Law through Resistors and Capacitors to Coil Winding and Antenna calculations.