

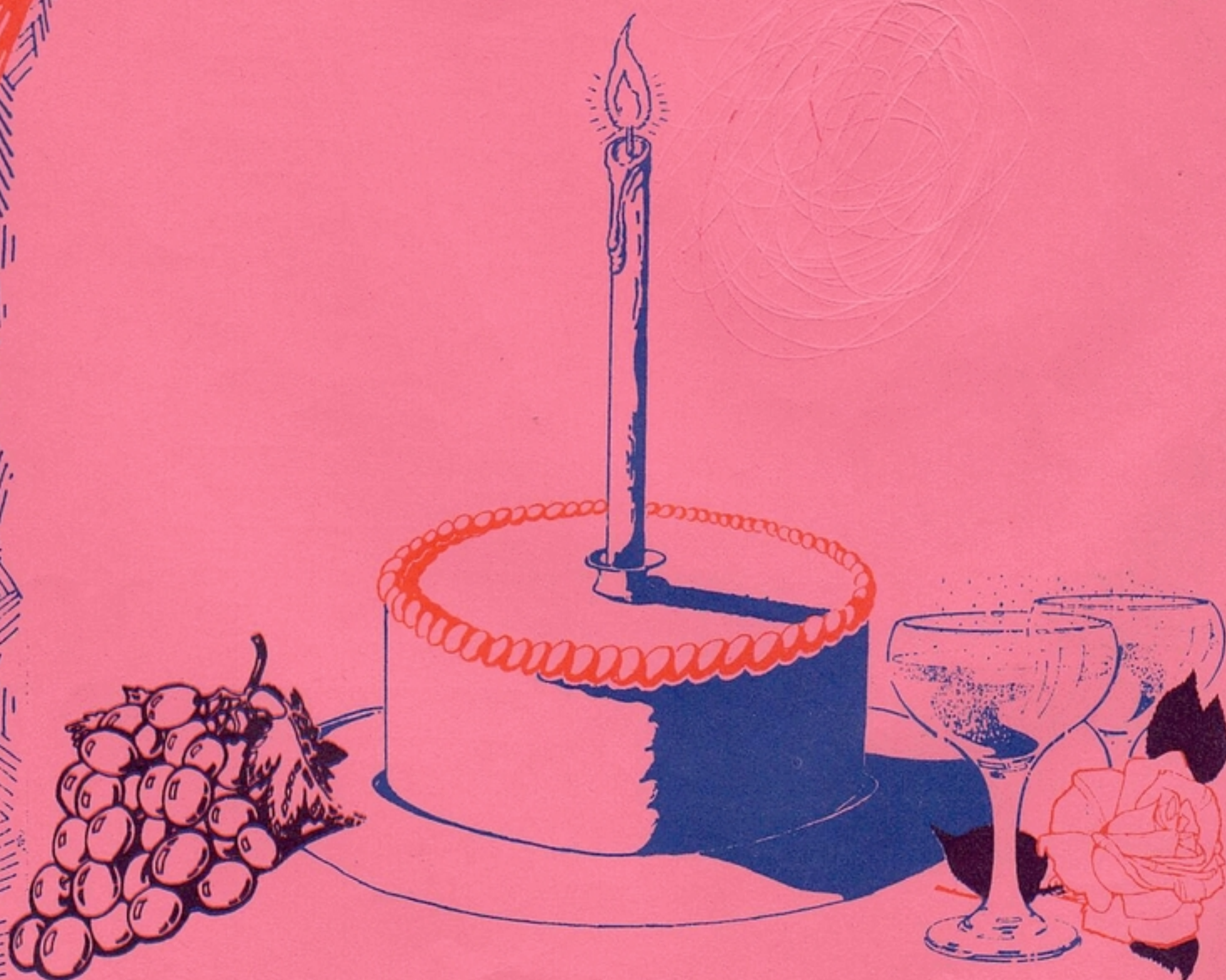
# memopad

Memotech Computer User Club Magazine

£1.25



## SPECIAL.. BIRTHDAY EDITION







## CONTENTS

EDITORIAL .....	PAGE 2
HI-SCORES & HIT LIST .....	PAGE 3
ALICE ADVENTURE MAP .....	PAGE 4
REVIEW MICROSOFT BASIC .....	PAGE 5
REVIEW A NOVICE'S VIEW OF THE FDX .....	PAGE 6
REVIEW FIG FORTH .....	PAGE 7
REVIEW DMX 80 PRINTER .....	PAGE 8
REVIEW BRIDGE & SON OF PETE .....	PAGE 9
REVIEW MANIC MINER .....	PAGE 10
HOW TO USE THE ASSEMBLER .....	PAGE 11
STARTING FORTH .....	PAGE 15
BASIC GRAPHICS PART 4 .....	PAGE 18
RST 28 THE FACTS .....	PAGE 23
AN EASY HARDWARE INTERFACE .....	PAGE 26
SPEEDY 3D .....	PAGE 28
3D GRAPHICS .....	PAGE 33
SOUND GENERATOR .....Input codes .....	PAGE 34
PROGRAM BERTIE BUG .....	PAGE 36
KIDDIES COMPETITION .....	PAGE 37
VIEWPOINT .....	PAGE 38
CP/M SOFTWARE .....	PAGE 40
SOFTWARE .....	PAGE 41
RAFFLE WINNER ... SUBSCRIPTIONS... END STATEMENT.....	PAGE 42

## GENPAT

MEMOPAD IS PUBLISHED BY SYNTAXsoft FOR THE MEMOTECH USER GROUP  
 UNIT 109, GLENFIELD PARK, GLENFIELD ROAD, NELSON BB9 8AR  
 TELEPHONE: 0282 698849

COVER PRICE £1.25. MEMOPAD IS COPYRIGHT SYNTAXsoft 1985

AVAILABLE BY SUBSCRIPTION ONLY.



## EDITORIAL



I can't believe that it is now twelve months since I sat down to write the first editorial. But, believe it or not, it is true and here I am writing the birthday edition. I would like to take this opportunity of expressing my thanks to all members, particularly those who have been with us from the very start, for your support over the first year. To those of you who will not be renewing your subscriptions I wish you well in your quest for a better computer.

From this edition you will see that we have now started to use a new way of binding the magazine which I hope will allow you to retrieve and read the information printed within with greater ease. During the course of the next two editions we hope to print an index for the first twelve publications.

We are streaking ahead with our proposal to open up a Bulletin Board and it should not be long before it is fully operational. In the meantime, anyone wishing to send me a message can do so by accessing PIP-Sheffield TBBS 0742-667983 (one of the best boards in the country) and leaving a personal message for Keith Hook. Within the pages of the magazine, you will see that we have tied a deal up with Pace to allow us to sell their modem at a 15% discount to existing members. We have done this because the people who make the Demon modem have not had it sanctioned by British Telecom, and they are also having great difficulty in fulfilling their delivery dates.

Memotech have released a new package which consists of a 500K slim-line disc drive, cp/m 80 column board and MTX512 computer all for £499.00. This will be on general release from September. Unfortunately this will NOT be on view at the PCW Show because the Company is not attending! As most of you know, Genpat was intending to have a stand at the exhibition, but we have been badly let down by ACC who organise club stands, and even after several letters and various messages, left on their bulletin board, we have not had the courtesy of a reply.

I am scheduled to attend a meeting with Memotech next week to discuss their intentions for the next 3 months. Until that meeting, forgive me if I am not too critical of them as this last month has been a series of in-house arguments. Also, somewhere along the way, I have picked up a terrible cold which doesn't help matters.

September should see the release of another batch of games from Megastar... as soon as we receive full information we will keep you informed.

If you have a feeling of being 'left out in the cold' don't get too despondent - a lot of clever people own Memotech computers. The MTX will be around, and in use, long after many other machines are just words from an echoed past.

Many thanks for the birthday cards ..... keep on tapping





# HI SCORES



-----  
HIGH SCORES : HIGH SCORES... Can you do better ??  
-----

GOLDMINE	8,995	Daljinder Singh
ASTRO-PAC	185,990	Richard Nash
BOUNCING BILL	128,142	Alan Dobson
SNAPPO	111,670	Richard Franks
KNUCKLES	999,999+	Sally Street
NEMO	17,610	Richard Nash
COBRA	9,924	Richard Nash
MISSION ALPHATRON	68,250	T. Eriksson
TAPEWORM	168,515	Alan Dobson
TOADO	178,292	Gavin Gaunt and Nicholas Locke
POT HOLE PETE	75,080	Alan Hill
MAXIMA	500,000	Virginia Parton (5 ships left)
STAR COMMAND	140,430	Ian Nichols
PHAD	26,000	Sally Street
OBLOIDS	57,800	P. Crighton
KILOPEDE	92,253	Richard Nash
3D TACHYON FIGHTER	10,700	Lesla Woodger
CONTINENTAL RAIDERS	106,240	Sean Haverly
BLOBBO	148,283	Elizabeth Mahon
QOGO 2	138,960	Alan Dobson
MINEFIELD	1,500	David Nash
FLUMMOX	166,690	Andrew Miller
TURBO	15,630	N. Crighton
FATHOMS DEEP	1,290	Alan Dobson
AGROVATOR	180,361	Simon J. Midcalf
FIREHOUSE FREDDIE	29,620	T. Eriksson
QOGO	21,360	T. Eriksson
ARCADIANS	15,100	Richard Nash
MISSILE COMMAND	12,110	Richard Nash
LITTLE DEVILS	14,680	Andrew Ewing
FELIX IN THE FACTORY	11,950	Richard Nash
HUNCHY	7,908	R. Harmer
SON OF PETE	7,131	Gavin Gaunt and Nicholas Locke
HAUKUARS	8,800	R. Harmer
CHAMBERGIDS	Completed 11 mins	N. Bowles
ESCAPE FROM ZARCOS	15 Items	R. Harmer
MISSION OMEGA	9,350	R. Harmer
ICEBURG	17,431	Alan Dobson
SNOWBALL	450	P. Crighton

Can you beat these high scores ? Do you have a high score for a game not mentioned above ?

# GENPAT HIT LIST

This chart is compiled purely on the sales of software within the Club and will be updated every month.

## Adventure

1. EMERALD ISLE LEVEL 9
2. MURDER AT THE MANOR SENTIENT
3. ADVENTURE QUEST LEVEL 9
4. THE KEYS TO TIME SENTIENT
5. SNOWBALL LEVEL 9
5. ALICE CONTINENTAL
7. DUNGEON ADVENTURE LEVEL 9

## Arcade etc.

1. MEMOSKETCH SYNTAXsoft
2. ESCAPE FROM ZARKOS MEGASTAR
3. EDASM SYNTAXsoft
4. DENNIS AND THE CHICKEN PANsoft
5. TARGET ZONE SYNTAXsoft
6. QOGO 2 MEGASTAR
7. FIG FORTH SYNTAXsoft
8. AGROVATOR SYNTAXsoft
9. SON OF PETE MEGASTAR
10. H AND L DUMP MEMBRAIN

## Educational

### TOP SELLING EDUCATIONAL PROGRAMS

1. SPELLICOPTER SENTIENT
2. FIRST WORDS CONTINENTAL
3. MATHS 1 CONTINENTAL







## ACCESS "PRESTEL" WITH A MEMOTECH!

### JARO SPEED-SPLITTER BOARD

- ‡ Allows 'split-speed' operation of MTX channel A (ie 1200/75 'user', or 75/1200 'reverse Prestel' or 'host')
- ‡ Completely transparent to normal MTX operation, requires no alteration to any existing software ('off' at power-up)
- ‡ Software control via output port 7 (uncommitted PIO)
- ‡ Fits inside lid of MTX casing - does not prevent fitting of any MTX add-on boards, e.g. extra memory, etc.

BOARD ONLY - 1 solder connection to RS232 required: £15.00  
BOARD FITTED TO RS232 BOARD - no soldering required: £77.65

15% DISCOUNT for Genpat Members: Board £12.75, Board/RS232 £66

Both options include lead and easy-to-follow instructions. The speed-splitter can be easily installed by the user without any technical knowledge at all in under 30 minutes.

### JARO VIEWDATA SOFTWARE CASSETTE

Available for all MTX machines (specify) - £7.95 (no discount)  
Turn your Memotech computer into a monochrome (black on cyan) 1200/75 (V23) viewdata terminal:

- ‡ Full viewdata character set including contiguous/separated graphics, cursor on/off and addressing from host, & flash
  - ‡ Viewdata keypad with 'asterisk' and 'hash' keys
  - ‡ Seven page internal frame store with page number in cursor
  - ‡ Page print via parallel (Centronics) port (Epson codes)
  - ‡ 16 programmable keys for frequently used frames or user-id
  - If the 'Demon Electronics' modem is used, in addition:
  - ‡ Up to 8 autodial telephone numbers and keypad dialling (Requires Jaro Speed-splitter Board, RS232 board and modem)
- Note: The following viewdata features cannot be implemented on the cassette version of JaroViewdata due to limitations of the MTX Video Display Processor: colour attributes, double height, and concealed text.

### IMPORTANT NOTICE TO OWNERS OF FDX, HDX, AND SDX/CPM SYSTEMS!!!

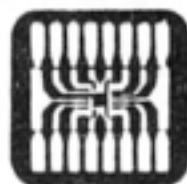
We hope to release a FULL COLOUR/FULL FUNCTION disc version of the Jaro Viewdata program shortly. Features will include:

- ‡ Full colour, text and graphics!
- ‡ Store page to disc and view whilst offline!
- ‡ Offline preparation of graphics or text strings for sending!
- ‡ Menu-driven user interface to program!
- ‡ Etc., etc.

Details of prices and availability should be announced in this magazine next month. Due to long delays in supply of the Demon V.21/V.23 modem, for which the cassette software was specially written, versions of the disc software for use with different modems may become available, eg Tandata Tm110, Miracle WS2000. STOP PRESS - Genpat now recommends the PACE NIGHTINGALE modem: Disc version for PACE modem + autodial card for 1st release !!

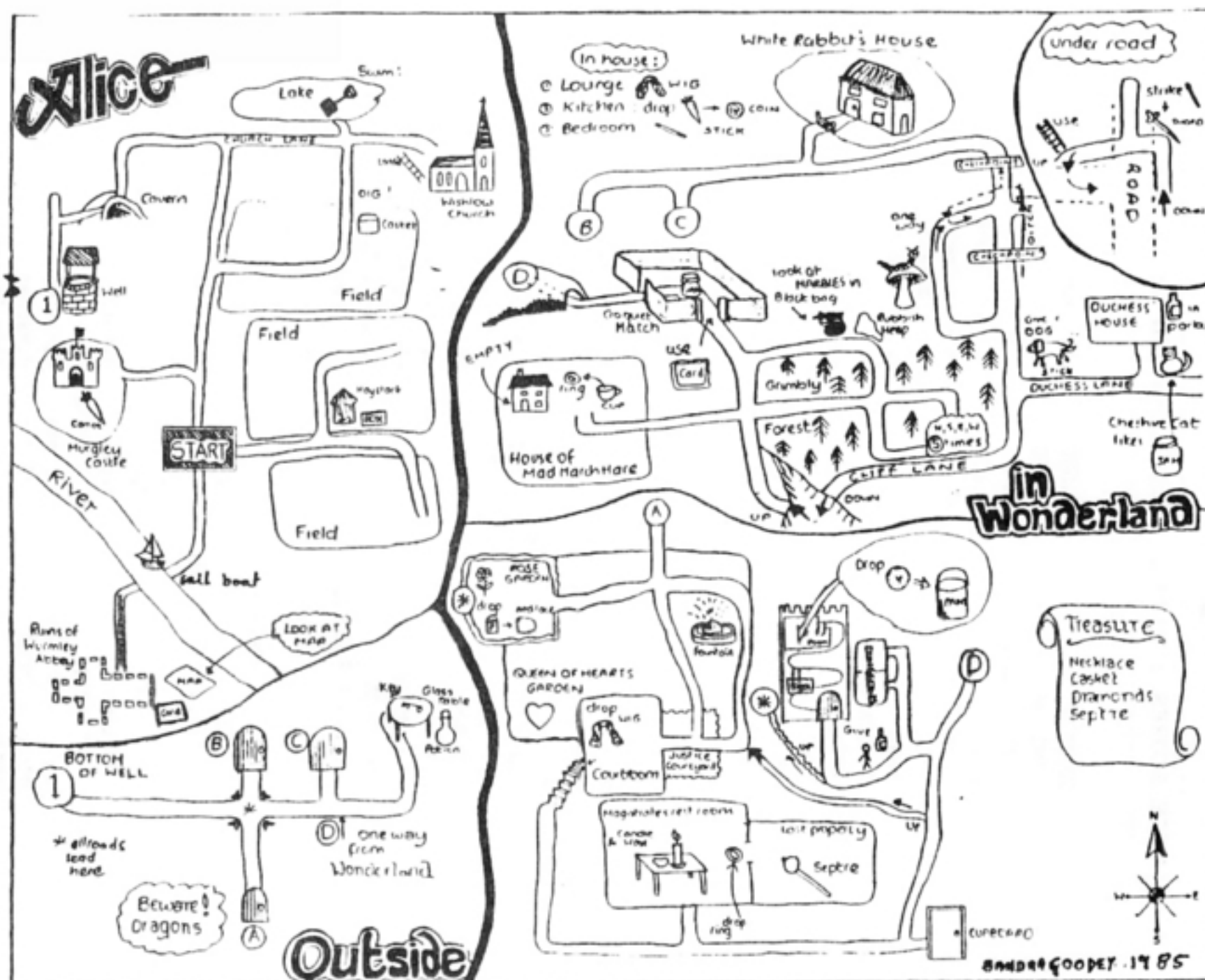
Overseas customers - Orders from other countries are welcome, but please note that an extra postage/insurance surcharge must be added for some orders/destinations - For exact quote please send 2 IRCs with your requirements for prompt reply.

Order either from the address below or from Genpat. Only order at discount prices if you quote your current Genpat membership number. If ordering from Jaro, enclose cheque/PO, made payable to Jaro Computer Services, and MENTION MACHINE (512,500,RS128)



### JARO COMPUTER SERVICES

4 Finnart Close Weybridge Surrey KT13 8QE  
Tel: Weybridge (0932) 57398  
Prestel Mailbox 019995085







# MBASIC

# MICROSOFT cpm

Microsoft's MS Basic is one of the most extensive implementations of Basic available to the CP/m programmer and it supports many unique features not found on the MTX or, in fact, many other Basic interpreters.

MS Basic is capable of handling four types of variable: INTEGER (-32768 through to + 32767), SINGLE PRECISION FLOATING- POINT ( 7 digits),DOUBLE PRECISION FLOATING-POINT ( 16 digits), and STRING ( up to 255 characters). Because this version of basic can handle integers a lot of Basic programs can be speeded up by DEFINing all integer variables at the beginning of a program.

True BOOLEAN operations are supported: OR,AND,NOT,XOR,EQV,IMP which allows statements in the form..

```
IF I AND 8^2 THEN GOTO 20
IF G AND 16 THEN RETURN.
```

or,

One of the most useful features is PRINT USING which allows formatted output to the screen or printer. Scientific notation is supported with this facility.

Variable length random and sequential access to disc files is also supported, and files can be manipulated in many different ways.

Machine code buffs can interface useful assembly sub-routines and access them with the USR and CALL commands.

Microsoft Basic is run by typing MBASIC <RET> and the interpreter is then loaded into the transient program area. You can also run Mbasic by specifying a set of options: MBASIC C:DEMO.BAS which will load Mbasic the run the program DEMO.BAS.

Many other useful functions and commands are included in Mbasic such as:

DEF FN	DEFINE FUNCTION
VARPTR	GET THE ADDRESS OF A VARIABLE IN MEMORY.
MERGE	MERGE A FILE INTO CURRENT PROGRAM.
ON ERROR GOTO	ENABLES ERROR HANDLING.
RESUME	CONTINUE PROGRAM EXECUTION AFTER AN ERROR HAS BEEN RECOVERED.
TRON/TROFF	TRACE THE EXECUTION OF A PROGRAM ON SCREEN
WIDTH	SET WIDTH OF PRINTED LINE.

Rather than rabbit on about how good this version of Basic is, take a look at the following program written in MBASIC.

```
1' Decimal to Hex conversion
10      DEFFNHX$(A1%)=MID$("0123456789ABCDEF",INT(A1%/16)+1,1)
+MID$("0123456789ABCDEF",A1%-INT(A1%/16)*16+1,1)
20 DEFNHX1$(A1%) = FNHX$(ASC(MID$(MKI$(A1%),2))) + FNHX$(ASC(MKI$(A1%)))
30 CLS: PRINT "DECIMAL TO HEXIDECIMAL CONVERSIONS"
40 PRINT:INPUT"WHAT IS THE NUMBER [FROM -32768 TO 65535]";A!
50 IF A!>32767 THEN A%=A!-65536 ELSE A%=A!
60 PRINT "HEXIDECIMAL VALUE = ";FNHX1$(A%)
70 GOTO 40
```

Line 10 defines the function HX\$(A1%) this converts an integer from 0 - 255 to the corresponding hex notation from 0 - FF. Line 20 defines the function HX1\$(a1%), and this handles the conversion for integers from -32768 to 32767.





A% defines variable A as an integer.

A! defines variable A as a single precision number.

Try re-writing this program to operate under FDXBasic and you will instantly see the major differences between the two versions of Basic.

Another plus for MBasic is that you can run a completed program through BASCOM which is Microsoft's COMPILER. Bascom converts the Basic program into a 100% machine code, and there is no need to tell you how fast the resulting program will run.

MBASIC IS PRICED AT £272.00 CLUB PRICE

BASCOM IS PRICED AT £308.00 CLUB PRICE. KH.

## A NOVICE'S VIEW of the FDX

My choice of the MTX 512 was influenced by the fact that I could upgrade at a later stage to a proper business system. This year I decided to take the plunge and buy the FDX single drive plus the first upgrade which would give me 80 column text. The following report is my impressions of this system.

When I opened the package and took out the system I was surprised by the size of the FDX. The disc drive is much bigger than the Cumana disc drives for the BBC because you are getting much more for your money. However it does give you a bit of a shock at first.

The next job was to set up the system. Memotech have provided very easy to follow instructions but let me give you one word of warning - be very careful in plugging in your ribbon cable into the MTX. Displacement by only one pin isn't very noticeable until you switch on when all you will get are lines across the screen. I made this mistake and thought the FDX was faulty.

Once set up you can then begin to plough your way through the manual. One word described my feelings PANIC. It was rather like leaving the local football team to play for a first division club the rules are the same but it's a very different ballgame! Mind you, my problem was heightened by the fact that Memotech forgot to include, with the manual, the vital two pages that describe how to format and syscopy from the system disc with a single disc. That omission cost me several expensive phone calls and a severe headache.

One criticism I have of the manual is that it seems to take a lot for granted in some areas whereas in others it treats you like an idiot. For instance the Introduction to Newword spends a lot of time explaining the difference between wordprocessing and typing which I felt was unnecessary. Also the step by step procedure for using Newword is frustrating in its minutiae of detail.

On the other hand, the manual tells you that you can revert back to an MTX by typing "MTX". When you do the screen "freezes up" on you. It took another phonecall to Memotech to discover that when you do type "MTX" or "FDXB 40" that you are returned to 40 column text and need to plug your monitor into the back of the computer. I know its obvious when you think about it, and to be fair it is mentioned briefly in the manual but only very briefly.

One other moan I have about the manual is that the Newword appears to have been hurriedly prepared and looks as if it was copied using a dot matrix printer. This robs it of the professionalism that I think it deserves. Also there is a continuing reference to the Newword Encyclopedia but it cannot be found. I thought that this was another manual that they had forgotten to send but another call to Memotech informed me that this was not supplied and would cost another £50.





Well I have been happily tapping away with the upgrade for over two weeks now. I find that the software is far superior to what I was using before. The disc system is much, much faster than the tape and I find the DIRECTORY command a great help. I would strongly recommend anyone who can afford it to go ahead and purchase this system since it truly turns your MTX into a full blown business machine. ★

Stan March.

## FIG FORTH

## SYNTAXsoft

For those not familiar with the language called FORTH, first speed: a FOR NEXT LOOP OF 30000 in Basic will, even on the MTX, take 33.5 seconds. The same loop in FORTH runs at an amazing speed of 2.5 seconds !! Yes! Two point five seconds. Next, FLEXIBILITY: FORTH has the ability to be extended at will. This is accomplished by using "subroutines" called WORDS. These words are stored in a vocabulary and may be listed using the word VLIST.

The name FIG stands for the FORTH INTEREST GROUP who are a non-profit making organisation dedicated to the furtherance of the language.

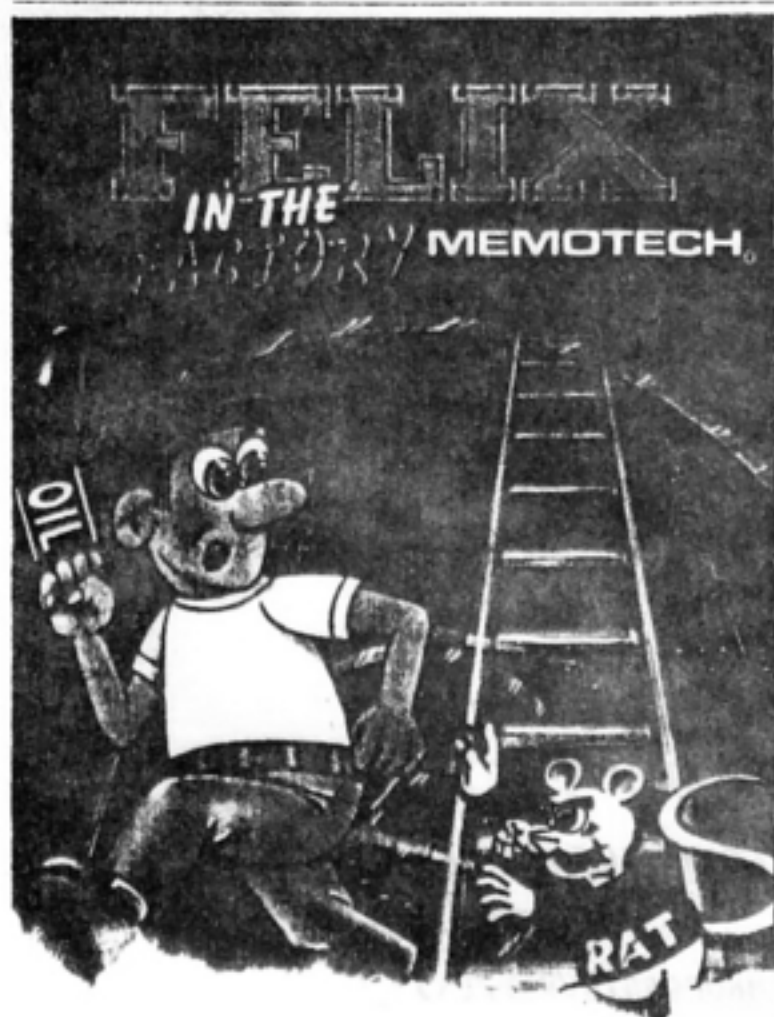
This implementation, by Keith Jones, is extremely good value for money - considering the listing occupies some 75 A4 pages of code which does not include the necessary calls to the system ROM. The manual supplied with the tape is clear, and describes the operation and syntax of the nucleus vocabulary. \*ED Novices to the language will require further reading material on the subject as the manual is not intended to be a tutorial.

Echo to the centronics port can be toggled by making LINK non-zero ... 1 LINK will send output to the printer and the screen, while 0 LINK will switch the printer off

SAVET will save screens 0 - 4 and takes 25 secs for the 5K bytes involved. Use only screens 1 - 4 for programs as screen 0 is only for comments.

Recommended reading: FUNDAMENTAL FORTH Olney & Benson. FORTH TECHNIQUES Olney & Benson. FORTH PROGRAMMING Leo J. Scanlon.

F.I.G : Roger Firth Esq, Membership Sec, FIG(UK), 7 Wyndham Crescent, Woodley, Reading, RG5 3AY  
Annual subscription £7.00.



**Special Offer**

TO

**CLUB MEMBERS**

**£4.95**

WE HAVE MANAGED TO  
ACQUIRE THE FULL STOCK  
OF FELIX AND WE OFFER  
THIS GAME TO YOU AT A  
SPECIAL PRICE OF £4.95  
INCLUSIVE.

FEATURES INCLUDE: 2 SPECIES OF GREMLIN  
GIANT RAT  
PITCHFORK & POISON POUCH  
PLUS CONVEYOR BELT HAZARDS.

**Special Offer**

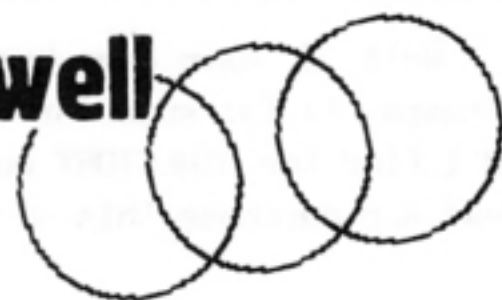
PRODUCED BY MICRO POWER.



MEMOTECH  
DMX

80

Tim Rothwell



One of the most important factors to take into consideration when choosing a printer is whether it is compatible with the software available for your machine. The world of printers has a standard that dominates it in a similar way that the IBM standard now dominates the business micro market - namely Epson. Most software which utilises the functions of a printer is written with an Epson compatible machine in mind. The DMX 80 is, you will be glad to know, fully compatible with the Epson control codes meaning that using it with CP/m or tape-based software will cause no problems.

The manual supplied with the printer is very comprehensive - in some parts too comprehensive. It does, however, give very clear instructions, although a novice may find it difficult to sort out the control sequences. This has now been corrected by the addition of a small crib card that easily explains the format for each code. The printer has 8 DIP-switches which allow you to set certain functions: print mode or international character sets. Once installed these function are then set at power-up. All of the functions can also be controlled from within a program by using LPRINT CHR\$(n). The printer will take fan-fold or single sheet paper from 4" to 10" wide.

If you have been a subscriber to Memopad from the start you will be well aware of the different print styles available as most of the early Memopads were produced using the DMX. However, for the purpose of this review, I have included a sample of the various types of print.

The printing speed of the DMX 80 is 80 characters per second (CPS) in the PICA mode and 96 cps in the ELITE mode. The printer is also bi-directional and logic seeking which cuts out unnecessary delays when printing.

The print head is a 9x9 matrix and is claimed to have a life span of 150 million characters but I haven't yet had chance to print out WAR AND PEACE to check this claim.

The printer also supports bit-mapping which will allow you to send all those masterpieces, created under Memosketch, from the screen to the printer. Using the NODDY format with the DMX is also easy. Another interesting feature is the facility to re-define the character set which can be loaded in from a Basic or machine code program, and once installed will remain in use until you switch off or cancel the command.

The Memotech DMX 80 is a well constructed unit which is reasonable value for money. The club has had one in service for over 18 months and it has never, as yet, failed on us. ★

PICA PITCH !"#%&'()\*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN O PQRSTUVWXYZ[\]^\_`

ELITE PITCH !"#%&'()\*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN O PQRSTUVWXYZ[\]^\_`

DOUBLE WIDTH !"#%&'()\*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN O PQRSTUVWXYZ[\]^\_`

EMPHASIZED !"#%&'()\*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN O PQRSTUVWXYZ[\]^\_`

DOUBLE PRINTED !"#%&'()\*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN O PQRSTUVWXYZ[\]^\_`

UNDERLINE !"#%&'()\*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN O PQRSTUVWXYZ[\]^\_`

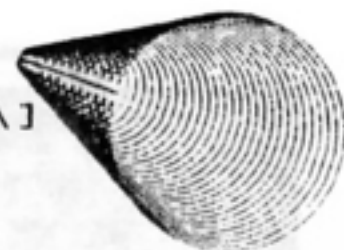
SUPERSCRIPT !"#%&'()\*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN O PQRSTUVWXYZ[\]^\_`

SUBSCRIPT !"#%&'()\*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN O PQRSTUVWXYZ[\]^\_`

ITALIC !"#%&'()\*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN O PQRSTUVWXYZ[\]^\_`

PICA PITCH ITALIC !"#%&'()\*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN O PQRSTUVWXYZ[\]^\_`

ELITE PITCH ITALIC !"#%&'()\*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN O PQRSTUVWXYZ[\]^\_`







# BRIDGE



CONTINENTAL SOFTWARE

Having observed the Tandy version of Bridge, and taking account of what their program achieves, I looked forward, in anticipation, to the MTX version. The cassette cover announces that the program recognises all CONTRACT BRIDGE hands and playing positions ...."written in high-resolution graphics .." Great! What the limited Tandy is capable of the MTX should excel at.

**WHAT A LET DOWN.** The graphics are appalling when compared to the Tandy - keep in mind that the Tandy only has block graphics - in the Tandy program the dealing is actually displayed. All we have to look forward to on the MTX is a screen full of suit emblems with a message at the base of the screen saying, "PLEASE WAIT DEALING". Once the dealing has been completed the program displays your hand (you always play SOUTH) and the bidding commences.

My next series of criticisms refer to their responses that should be made under certain conditions. The first being the response if say, east opens, and south has opening points also. If south has less than a normal 2 opening hand (around 18 to 22 points) then the accepted convention is to double east's opening bid. The program recognises only two conventions: Stayman & Gerber. The first is asking for a major suit after a No Trump bid. The second is asking for Aces - no mention of this convention is mentioned in the notes contained, nor does the convention go far enough. It should ask for Kings if a slam is on ... a better convention would have been Blackwood.

Once the program has been run a few times it becomes apparent that the hands are not random as one would expect from a normal shuffle.

Two final criticisms: a) Once a slam has been bid, and if the bid has been doubled, the program has a craving to crash out. b) When one team has reached game they then become vulnerable this is not displayed on the screen although the scoring reflects this position in the game.

In conclusion, it is difficult to assess who the program is aimed at ... beginner or more conversed player. It lacks good documentation to be of any real assistance to the beginner. For the more advanced player some of the hands dealt would have you pulling your hair out by the roots. All in all, an acceptable program that could have been better thought out. ★

Raymond Holroyde

# SON OF PETE

MEGAstar

Not another Manic Miner I thought! What a pleasant surprise though when the program loaded. A nice title screen appears showing how the function keys are defined. You have the choice of restoring fall, abort game and hold. A nice feature of the game is that you may use either joystick to control your character. Anyway, I pressed fire and started the game...

On the first screen is a mobile contraption which appears to be a steam engine! One original addition are varying sizes of what I imagined to be hydraulic rams. As you walk over them, you are catapulted into the air with considerable force. This usually ends in death from well positioned spikes!

There are different ways of travelling from room to room. The easiest (and most obvious) is simply to walk from the edge of one screen onto another. A more elaborate way of inter-room transport is via waterfalls. To use these, simply step onto the top and away you go! As you may expect, these are one directional only ie. downwards. Also included are moving ladders, conveyor belts, crumbling platforms etc.





The only thing I haven't mentioned is the object of the game! In typical Manic Miner style, this is to collect the keys and escape imminent death from fast moving nasties.

Control is very easy using the joystick, the fire button making our character jump. Also, the animation and graphics were first rate in all of the screens I visited. As for the sound, it is quite good, but nothing to shout about unfortunately. To round off with, here are my marks out of ten:-

GRAPHICS: 10  
CONTROL: 9  
SOUND: 7  
ADDICTIVENESS: 9  
VALUE: 9

T. Marstian



This is certainly another winner from Megastar! ★

ED: HOW CAN ANYONE GIVE THE SOUND ON THIS GAME 7 ? IT'S TERRIBLE !

## MANIC MINER

## SOFTWARE PROJECTS

Well. Here is the first game that began one of the biggest trends in computer software - the ladders and platform game.

Once it has loaded you are presented with a zany piano tune complete with graphic representation of the piano. As on PHP, after a time the program defaults to a Demo Screen Mode showing off its 20 screens - less than PHP but some are very taxing.

The graphics are first class and show lots of imagination. (The killer toilets blew my mind!). The one place where the game falls short is the "in-game" sound ....hardly any special sound effects, and the tune is pretty laborious. This part of the program could have really been improved if the full capabilities of the MTX sound chip had been utilised.

Inevitably this game will be compared with Continental's Pot Hole Pete, and arguments will ensue as to which is the better program. Personally I think both have their own special attractions and will provide a game of class for any avid games player. ★

MARKS OUT OF TEN

GRAPHICS 10  
SOUND 6 ... IN GAME  
7 ... TITLE SCREEN  
VALUE FOR MONEY 8

SIMON CRUDDAS



I hope to see more from Software Projects.

ED: SOFTWARE PROJECTS HAVE NOT YET DECIDED WHEN TO RELEASE THIS GAME. I PHONED THEM SHORTLY BEFORE GOING TO PRESS AND THEY SAID THEY WOULD LET ME KNOW ..... TO DATE NOTHING! ★





# Using The Assembler K. Hook

I would think that most of you have, at some time or other, tried entering the assembler. Whether you made any sense of it is dependent on how well you understood the Manual. This article is not intended to be a tutorial on the complexities of machine code but more a reiteration of the MTX Manual on how to use the in-built assembler.

The immediate advantage of the MTX assembler over most other micros is that it is very easy to use. It does not have to be loaded in from tape and is called from Basic as an "in line" assembler. All entered code is stored, in memory, as machine-executable object code. The readable SOURCE code is generated on using the command "list", and at this time the MTX dis-assembles the object code by inserting labels, text etc. which are stored in tables below the object code. This is one of the reasons a listing becomes slower as the program grows in length.

At this point it is important to realise the following:

- Since code is stored in a basic line the actual location of the code will change if the basic part of the program is modified or lines are added.
- A program that uses two separate code lines will not match up with the original listing if comment lines are omitted (if included) < show examples >
- MTX500 machines start at 8000X, MTX512 machines start at 4000X.

If you are on a 500 and typing a listing written for a 512 all the code should still match up, but will be located 16K up in memory and have the prefix 80 not 40, the reverse applies for MTX512 owners. Any POKE/PEEKs in the basic program that access machine code must also have 16K added (16x1024) or 16K deducted. This doesn't apply to pokes that reference the system variables as these are in the same place whichever machine you are using.

```

8007      LD HL,BUFFER
800A      LD B,B
800C LOOP: LD A,(HL)
800D      LD (HL),A
800E      INC HL
800F      DJNZ LOOP
8011 BUFFER: DB 30,40,50,60,70,80,90,100
8019      RET

```

Symbols:  
BUFFER 8011    LOOP    800C

1 REM THIS LINE HAS BEEN ADDED LATEP  
10 CODE

```

802C      LD HL,BUFFER
802F      LD B,B
8031 LOOP: LD A,(HL)
8032      LD (HL),A
8033      INC HL
8034      DJNZ LOOP
8036 BUFFER: DB 30,40,50,60,70,80,90,100
803E      RET

```

Symbols:  
BUFFER 8036    LOOP    8031

1 REM EXAMPLE B:  
10 CODE

```

8018 START: LD A,30      ; JUST SOMETHING TO PUT HERE
801A      LD A,30      ; DOES NOTHING JUST REPEATING MYSELF
801C      RET

```

Symbols:  
START 8018

20 REM THIS IS 2ND HALF OF LISTING  
30 CODE

```

80A3      LD HL,30000    ; TRY ANOTHER REGISTER
80A6      RET            ; AND RETURN
80A7      RET

```

Symbols:

1 REM EXAMPLE B:  
10 CODE

```

8018 START: LD A,30      ; DOES NOTHING JUST REPEATING MYSELF
801A      LD A,30
801C      RET

```

Symbols:  
START 8018

20 REM THIS IS 2ND HALF OF LISTING  
30 CODE

```

80B3      LD HL,30000    ; TRY ANOTHER REGISTER
80B6      RET            ; AND RETURN
80B7      RET

```

Symbols:

*NOTICE THE CODE ADDRESSES*





To enter the assembler we have first to let the MTX know in which basic line the code is to be stored. This is easier accomplished by typing ASSEM n <RET> where n is the basic line number eg:

ASSEM 5 <RET>.

The above assigns line 5 as the assembler code line and once the return key has been pressed the screen is cleared and you are left facing the ASSEMBLE> prompt in the bottom left of the screen. You have now entered the ASSEMBLER and the prompt means that the MTX is waiting for a command, the following are the commands that are available to you in the ASSEMBLER mode:

C = CLEAR SCREEN  
E = EDIT  
L = LIST TO SCREEN  
P = LIST TO PRINTER  
T = PROGRAM POINTER TO TOP OF PROGRAM

## CLEAR SCREEN

Pressing C <RET> will clear the screen.

## EDIT

In the EDIT mode any code written to a memory location will overwrite the existing code at that memory address. To erase a line of code simply enter the assembler at the correct address E.G:

```
10 CODE
8007 START: DI
8008          LD B,8
800A          LD A,32
800C LOOP:   OUT (01),A
800E          RLCA
800F          RRCA
8010          DJNZ LOOP
8012          RET

Symbols:
START 8007    LOOP 800C
```

YOU TYPE  
ASSEM 10 <RET>  
E#8008 <RET>  
EOL <RET>  
CLS/<RET>  
CLS/<RET>

SCREEN SHOWS  
Assemble >  
8008 LD B,8  
8008 LD A,32  
Assemble >  
Ready

You can see from the above that the code at 8008 was deleted and the following code was moved down in memory and re-located.

You can edit any memory location ( within your program) by typing:

E Location/Label

The location can be an address (decimal or hexidecimal), or a label ( START, LOOP etc).

In the above example if you wanted to change the value of the B Register from 8 to 12 you would carry out the following steps:





## YOU TYPE

ASSEM 10 &lt;RET&gt;

E#8008 &lt;RET&gt;

Now move cursor over the 8 and type:

12 &lt;RET&gt;

CLS/&lt;RET&gt;

T &lt;RET&gt;

L &lt;RET&gt; and the program will be listed. You will see that the address 8008 now has the code.. LD B,12.

## SCREEN SHOWS

Assemble &gt;

8008 LD B,8

800A LD A,32

Assemble &gt;

LIST

Typing L at the "Assemble >" prompt will list the program from the current position of the Location Pointer. The Location Pointer keeps a check on where you are within the assembler. To list from the top of the program after an Edit or an Insert you must type T<RET> before listing, otherwise the program will be listed from the current address. When listing, the assembler displays one screen at a time, and to continue press any key at the bell sound. To terminate a listing press the BREAK KEY.

SYNTAX: L LOCATION/LABEL &lt;RET&gt;

LOCATION = DECIMAL, HEXIDECIMAL OR LABEL

PRINTER

The syntax for this command is: P LOCATION/LABEL &lt;RET&gt;

This command functions in exactly the same way as the L command except the listing is sent to the printer.

INSERT

This is the default mode. For instance, if you entered the assembler then typed <RET> at the "Assemble >" prompt the current memory location would be displayed and all the code you entered would be in the insert mode. Take a look at the following listing .....

## 10 CODE

```

8007 STAR:  D1
8008        LD SP,#FA96
800B        LD B,B
800D        LD HL,BUFFER
8010 LOOP:  LD A,(HL)
8011        OUT (01),A
8013        INC HL
8014        DJNZ LOOP
8016 BUFFER: DB £45,£42,£80,£34,£32,£56,£52,£91
801E        RET

```

## Symbols:

```

STAR      8007      BUFFER  8016
LOOP      8010

```

After entering this code if you wanted to insert a line at #8008 you could do the following at the

"Assemble &gt;" prompt :

## YOU TYPE

#8008 &lt;RET&gt;

LD (STACK),SP &lt;RET&gt;

CLS &lt;RET&gt;

## SCREEN SHOWS

8008 LD SP,#FA96

800C LD SP,#FA96

Assemble &gt;

Now type T <RET> L <RET> and as the code is listed you will see the following:





## 10 CODE

```

B007 STAR:  DI
B008          LD SP, (STACK)
B00C          LD SP, £FA96
B00F          LD B, B
B011          LD HL, BUFFER
B014 LOOP:    LD A, (HL)
B015          OUT (01), A
B017          INC HL
B018          DJNZ LOOP
B01A STACK:   DW 00
B01C BUFFER:  DB £45, £42, £80, £34, £32, £56, £52, £91
B024          RET
    
```

## Symbols:

```

STAR      B007      BUFFER  B01C
LOOP      B014      STACK   B01A
    
```

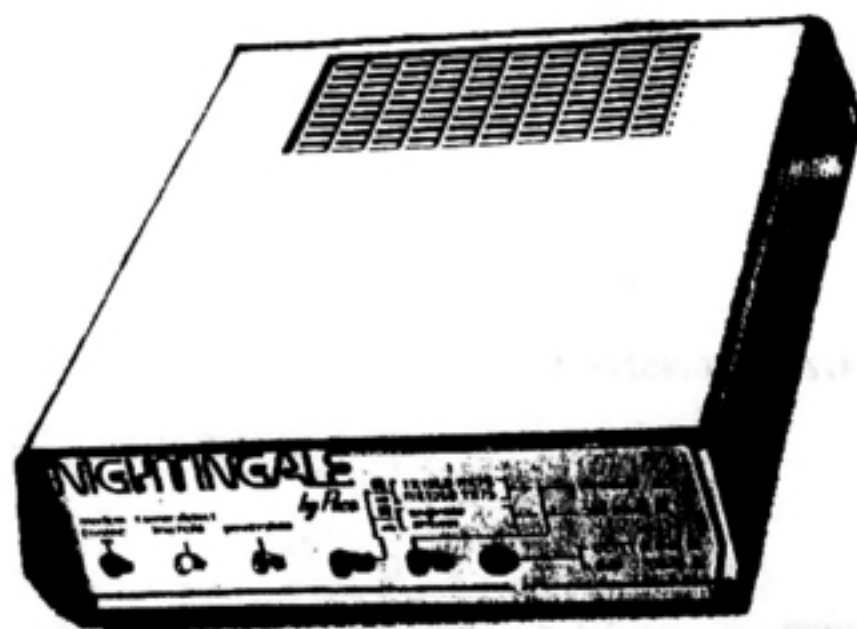
Can you see the difference between INSERT & EDIT? This time all the code has been moved UP in memory and the LD SP, £FA96 instruction has not been overwritten but has been moved to the next highest available address.

Another point where novices tend to get confused is that the numbers at the left of the display are MEMORY LOCATIONS and not LINE NUMBERS. And this is where the code resides in the MTX's memory.

Here is a tip that is not well known on the MTX .....

Lets say that I found I had made a mistake in my program, and I wanted to move the code at #8011 { LD HL, BUFFER } to the start of the program. Instead of having to enter the assembler and erase it at #8011 then go into the insert mode to "insert" it at #8007, all I need to do is enter the assembler TYPE E#8011<RET> and when the line is displayed, move the cursor over the memory location #8011 and change the address to #8007 <RET>. If you now look at the code by typing T<RET> L<RET> you will see that the assembler has automatically placed the code, and reassembled it, in the correct order!

Marvellous machine this MTX isn't it? ★



APPROVED for use  
with telecommunication systems  
run by British Telecommunications  
in accordance with the conditions  
in the instructions for use.

8/25023/E/500065

## PACE

Micro Technology

# £119.00 + £4<sub>pp</sub>

## WHY NIGHTINGALE?

Nightingale was designed to fulfill a need. It was designed to provide the right facilities at the right price.

Unlike previous generation modems, Nightingale features a variety of baud rates. The baud rate is the speed at which data is transmitted and received. The two most common rates for use on normal telephone lines are 300/300 and 1200/75. 300/300 denotes 300 baud full duplex, ie transmission can be at 300 baud in both directions at the same time. 1200/75 refers to a system in which one computer transmits at 1200 baud while receiving at 75 baud, again in full duplex. Roughly speaking 300 baud is equivalent to 30 characters/second, 1200 baud is 120 cps. The importance of having several baud rates is so that you can access systems which operate at either speed. For example, Prestel uses the 1200/75 rate but most bulletin boards use 300/300. Versatility is the key to successful communications.

When transferring large files connection time becomes important so that ideally 1200 baud should be used. However, error checking systems that are used to counteract 'noisy' telephone lines, require full duplex operation for the 'handshaking'. Unfortunately however, 1200/1200 full duplex modems are still relatively expensive, so to support fast file transfer,

Nightingale also features 75/1200, ie reverse Prestel rates. This means that files can in fact be transferred at 1200 baud by setting one modem to send at 1200 while the other receives at 1200. The full duplexing is then provided by the 75 baud 'return' channel. Once again Nightingale fulfills the requirement effectively.

Versatility in any product can easily be lost in the complexity of actually using it, a trap which Nightingale avoids. Two simple pushbutton micro-switches can be used to select the baud rates manually\* and two further switches facilitate the operations of self-testing and going 'on-line'. When in operation two coloured LEDs indicate the current status of the modem as transmitting, receiving or idle and 'carrier detect'.

Connecting Nightingale to your computer and telephone is another area into which we've put a lot of thought. The new modular BT plug (type 600) is used to connect Nightingale into the phone system and a matching socket on the rear of the modem provides the replacement socket for the telephone itself. Since this type of plug and socket are used as standard for all new telephone installations Nightingale is to a large extent portable. Further more, when Nightingale is connected but not on-line the telephone may be used normally.

WE HAVE NEGOTIATED A DEAL WITH PACE SO THAT MEMBERS CAN ACQUIRE THIS MODEM AT 15% DISCOUNT. AN AUTO DIAL/RECEIVE FACILITY WILL BE AVAILABLE AT THE END OF THE MONTH AS AN ADD ON. SUPPORTS 300/300 :1200/75: 75/1200. CHEQUE WITH ORDER TO SYNTAXSOFT. DELIVERY 7 DAYS.





# STARTING FORTH *Keith Jones*

This month we'll round off the maths commands by looking at mixed math operators, unsigned maths and double number operators.

The mixed length functions which FORTH offers are characterised by the prefix "M" e.g. "M\*", "M/" and "M/MOD". These are provided so that maths which would overflow the usual signed single length values can be operated upon. So if we wished to multiply 32800 by 2 then we would use "M\*" (m-star) in the form

```
32800 2 M*
```

and the result would be printed using "D." The reason we would use "M\*" as opposed to the usual "\*" is the result would be greater than 65535 which I'm sure you remember is the limit for a single length number.

"M/" acts as a mixed division. If we wished to divide a signed double length number by a signed single length number then we would use this word. It leaves two numbers on the stack, the remainder on top and the quotient below it.

"M/MOD" is the only FORTH operator which uses unsigned mixed length maths. We put an unsigned double number on the stack followed by an unsigned single length number. It leaves a double length value for the quotient and a single length value for the remainder.

And now we move on to the unsigned operators. If I'm going too fast for you then re-read this article and try some examples for yourself. The key-note in FORTH is experimentation. As an aside did you know that IBM has a huge sign in its offices which says, in big bold letters "THINK". Maybe you should put your own poster up above your screen which says "TRY IT".

Anyway, less of the chat, it's back to work. The unsigned operators which FORTH has to offer are "U\*" and "U/".

"U\*" requires two unsigned single length numbers on the stack. It leaves an unsigned double length number as its result. "U/" requires an unsigned double number as the dividend and an unsigned single length number as the divisor. It leaves an unsigned single length remainder and an unsigned single length quotient.

And finally, yes we've reached the end of the math's operators, the Double number addition "D+". Needless to say this word takes two double numbers and leaves a double number result of the addition

There you go. As you may have gathered I'm fairly fed up with the maths, and I'm pretty sure that you are too. Be warned that there's still a few of them lurking within the pages of your manual but unless you write to ask about them I think I'll just leave well alone.

And so on to the promised EDITOR commands. Just to recap, so far we've looked at "X", "P", "L", "B", "F", "CLEAR", "LIST", "WHERE" and "TOP". If you can't remember what these commands do then I would recommend that you look back to issue 10 and refresh your memory.

This month I want to look at the following commands;

"S", "D", "M", "TILL" and "N".

The "S" command is a way to Spread the lines so that you can insert text. The way it works is to make the number which you specify and all lines below this to move one line down. For example suppose you had the following screen:-





0 (EXAMPLE OF "S" COMMAND)

1 : EXAMPLE

2 100 0 DO

3 I .

4 ; ;S

5 etc.....

As you can see this will not compile (why not?). When you try to LOAD this screen you will get the error message #20, indicating that the definition is not finished. So if you type "WHERE" you will get

; ;S

^ "S" indicates that something's missing before the ";". As I'm sure you know it's LOOP that's missing, and we need to insert it at line 4. We will use "S" to do this. Typing

4 S

will spread the lines from 4 onwards, leaving line 4 blank. We can now type

4 P LOOP

and our definition will now compile. I realise that we could have used a combination of editing commands to have placed the "LOOP" before the ";" (what combination?) but in order to make our listing more readable it is useful to use separate lines for non-related commands. Another way is to indent groups of commands, as I have done with line 3.

The combination mentioned above, in case you were wondering, is "F ; B C LOOP " and I'll leave it to you to work out why each command is used here.

And so onto the "D" command. "D" is effectively the opposite of "S" in that "D" will Delete a line and move all the lines below it up a line.

You should remember the "B" command from issue 10. This moved the editing cursor back the length of the word which we had found in an "F" command. The "B" command is a limited version of "M". This is the cursor Move command and is preceded by a signed value. So

10 M

Will move the cursor forwards 10 characters and

-10 M

Will move it back 10 characters.

As an example of the use of "M" suppose we had the following line on an editing screen

0 : EXAMPLE-OF-M 123.456 78 M/MOD ;

After typing the line we suddenly realise that we only wanted "/MOD" and not "M/MOD" so we need to delete the "M". After typing "TOP" we have ;

\_ : EXAMPLE-OF-M 123.456 78 M/MOD ;

We could use "F M N N B X M" ( See below for "N". Once more I'll leave it to you to work out why) or we could use "26 M X M" which as you can see is shorter. Why "26 M" ? This will move the cursor to the location before the "M/MOD" and this means that the "M" which "X" will remove is the correct one, as it's the first one it comes across.



The "N" command is nothing more than a find Next occurrence of whatever it was you searched for using the "F" command.

And finally "TILL". This word is a string delete

Supposing we had

```
1: THIS-IS-AN-EXAMPLE 1000 0 DO I . LOOP BEGIN
etc....
```

And we had to put the loop in solely to slow the word down and now we wanted to remove it. We could use multiple deletes of words or we could use "TILL". After moving the cursor we have

```
: THIS-IS-AN-EXAMPLE_ 1000 0 DO I . LOOP BEGIN
```

We would then type

```
TILL LOOP
```

Which would then leave the following

```
: THIS-IS-AN-EXAMPLE BEGIN
```

Which is what we need.

That's all the EDITOR commands for this month but we're not finished with them yet.

Now let's have some fun. In order to celebrate GENPAT'S first birthday I now present a highly predictable set of words.

Just type them in as shown and then LOAD the screen.

SCR # 1

```
0 ( BIRTHDAY SPECIAL )
1 : CHOFF 4 0 DO I 0 0 SOUND LOOP ;
2 : SOU 15 SOUND ;
3 : P 1000 * 0 DO LOOP CHOFF ;
4 : C 478 SOU ; : D 425 SOU ;
5 : E 379 SOU ; : F 358 SOU ;
6 : G 319 SOU ; : A 284 SOU ;
7 : B 253 SOU ; : c 239 SOU ;
8 : d 213 SOU ; : e 190 sou ;
9 : f 179 sou ; : g 159 sou ;
10 : a 142 sou ; : b 127 sou ;
11 : BIRTHDAY 0 G 4 P 0 G 4 P 0 A 1 C 2 E 8 P 0 G 8 P 0 c 8 P
12 : 0 B 1 G 2 B 16 P 0 G 4 P 0 G 4 P 0 A 8 P 0 G 8 P 0 d 8 P
13 : 0 c 1 C 2 E 16 P 0 G 4 P 0 G 4 P 0 g 8 P 0 e 8 P 0 c 8 P
14 : 0 B 1 F 2 A 8 P 0 A 8 P 0 f 4 P 0 f 4 P 0 e 1 C 2 E 8 P
15 : 0 c 8 P 0 d 1 G 2 B 8 P 0 c 1 C 2 E 16 P ; BIRTHDAY
```



And that's all for this month. Next month we can all have a rest when I'll be presenting some routines which members have sent me. HAPPY BIRTHDAY "GENPAT" AND MANY MORE OF 'EM. ★





## BASIC

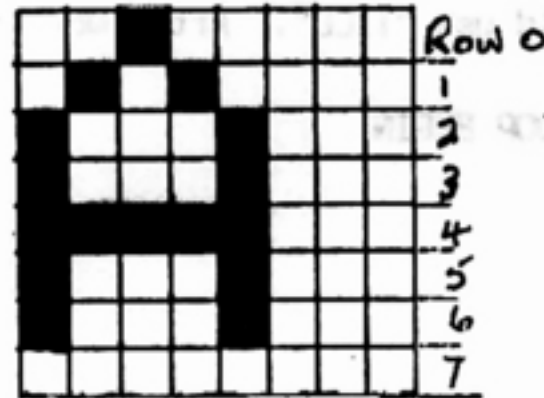
## GRAPHICS

## PART 4

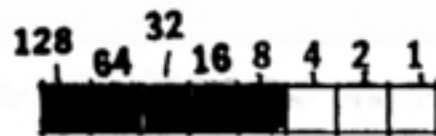
Michael Gaut

Hello again! This month, as promised, we're going to look at user definable characters and how to use them. Perhaps it may be a good idea to explain exactly what is meant by user definable characters.

Deep inside the computer can be found an integrated circuit (a chip to us!) which contains information for each character within the character set. The diagram below shows how the character information is stored using 'A' as an example:-



From the diagram, it can be seen that each character requires 64 ( $8 \times 8$ ) bits of memory to be stored in. As one byte of memory contains eight bits of information, we require a total of eight bytes. The value of each byte is calculated from the position of the bits switched on. The following diagram shows one line of the character 'A' and also a number above each bit.



To calculate the value of this bit, we must add together the value above each set bit. Therefore, if each bit was on, the value of the byte would be  $128+64+32+16+8+4+2+1$ , or 255. Alternately, if each bit was off, the value of the byte would be  $0+0+0+0+0+0+0+0$ , or 0. Therefore, the value of the byte in our diagram will be  $128+64+32+16+8$ , or 248.

Now that we know how character data is stored, let's move on and define our own characters. As you may have guessed, these are called user-definable characters. The command to do this is shown below:-

```
GENPAT n,p,d0,d1,d2,d3,d4,d5,d6,d7
```

The value of  $n$  determines the function of the statement. For the moment, only the values 0 and 1 need concern us. To define an ascii character (ascii value between 32 and 127) we need to use 0. To define non-ascii characters (codes between 129 and 154) we obviously use 1. As an example to clarify this, let's redefine ascii character 33 (the ! character).

```
GENPAT 0,33,255,129,129,129,129,129,129,255
```

Although not graphically startling, each time you press !, you should see our new character. Take a closer look. The data in the GENPAT statement should have produced a square. As you can see, the right part of the character is missing. The reason for this is that to produce a 40 column display with a maximum horizontal resolution of 256 pixels, each character can only be 6 pixels wide. This means that the number of horizontal pixels used in text mode equals  $40 \times 6$ , or 240 pixels. Therefore, 16 pixels are not used. These are situated at each side of the screen to prevent characters disappearing off the sides of the screen. So, even though we defined our square as being 8 pixels wide, the right two columns were missed off. The only way to solve this is by printing the character on a graphics screen, where each character is 8 pixels wide. This requires the whole screen because  $8 \times 32$ , or 256, pixels are required. In graphics mode, the full character is displayed without missing any pixels. Type in the following routine without resetting or NEWing the machine.

```
10 VS 4:CLS
20 PRINT "!!!!!"
30 GOTO 30
```



This should produce a row of squares on the screen. If you find that exclamation marks are printed instead, type in the GENPAT statement shown earlier, and RUN the program again.

Now that we know how to define graphics, one problem which arises is that each time we need a new character, we need to draw an eight by eight grid on a sheet of paper and then calculate the values produced by hand. Fortunately, there is a program currently available which I have found very useful in the past. It is called Graphics and is available through the club. It certainly saves on sheets of paper and calculator batteries !

Up to now, the characters we've defined have only been in one colour. The GENPAT statement allows us to specify the colours of each line separately using function number 2. The only limitation is that it only works with characters whose code are between 147 and 154. Therefore, we have 8 patterns which can be defined in glorious colour ! One other point is that full colour is only available on the graphics screen.

Each row has two colours assigned to it. These specify the ink and paper colours of the corresponding row of pixel information. If a pixel in the pattern is on, it will be displayed in the ink colour. Pixels switched off are displayed in the paper colour. To calculate the value needed for each colour byte, we must use the formula:-

$$\text{colour value} = \text{paper} * 16 + \text{ink}$$

where ink and paper are the codes of the colours required. As an example, to produce light yellow ink on a black background, we get:-

$$\text{colour value} = 1 * 16 + 11$$

$$\begin{aligned} \text{therefore, colour value} &= 16+11 \\ &= 27 \end{aligned}$$

To give a full example, let's define character 147 as the letter A with different coloured lines. Type in the following program:-

```
10 VS 4:CLS
20 GENPAT 1,147,32,80,136,248,136,136,0,0
30 GENPAT 2,147,86,86,86,88,88,88,89,89
40 PRINT CHR$(147);" ";CHR$(147)
50 GOTO 50
```

If the program worked correctly, you should have seen two multi-coloured letter A's at the top of the screen. Unfortunately, they are rather small, so why not try working out your own data and feeding the values into line 20.

Well, that's about all there is to know about user-definable graphics! Obviously though, we can't finish here, so let's write a simple program to utilise user-definable graphics. To begin with, let's set up our graphics screen, and define our characters.

```
10 VS 4:CLS
20 GENPAT 1,129,0,128,192,254,255,56,48,0:REM define aeroplane graphics
30 GENPAT 1,130,0,28,226,34,254,129,66,24:REM tank pattern 1
40 GENPAT 1,131,0,28,226,34,254,1,64,44:REM tank pattern 2
50 GENPAT 1,132,0,28,226,34,254,128,2,52:REM tank pattern 3
60 GENPAT 1,133,0,0,16,56,16,0,0,0:REM tank bullet 1
70 GENPAT 1,134,0,0,40,16,40,0,0,0:REM tank bullet 2
80 GENPAT 1,135,0,32,96,20,12,28,2,1:REM your bomb
85 GENPAT 1,136,255,255,255,255,255,255,255,255
```





The program itself is a simple game in which you control an aircraft flying above enemy territory. Obviously, the enemy are not very pleased about this, so they retaliate with a moving tank which fires large shells in your direction! You may attack the offending tank with your anti-tank missiles.

Unfortunately, with the missiles being Memotech guided, only one may be in the air at any one time!

The functions of the variables we are going to use are shown below:-

- PX - Your aeroplane's X screen coordinate
- PY - Your aeroplane's Y screen coordinate
- MF - Missile fired flag. If 1, then a missile is currently flying
- MX - Missile's X coordinate
- MY - Missile's Y coordinate
- TX - Tank's X coordinate. As the tank is always at the bottom of the screen, it doesn't require a Y coordinate.
- BX - Tank's bullet X coordinate
- BY - Tank's bullet Y coordinate
- TF - Tank fire flag. If equal to 1, then the tank has currently got a bullet in the air.
- BC - Bullet character. This is the ascii code of the character currently being displayed as the bullet
- TC - Tank character. This is the ascii code of the tank character currently being displayed.

So, here is the code required to move your aeroplane and drop the bomb. Add this code onto the end of the program to define characters given above.

```

90 LET PX=1:LET PY=1:LET MF=0
96 LET SC=0
100 GOSUB 1000:REM move aeroplane
110 GOSUB 2000:REM move missile if necessary
200 GOTO 100
1000 LET KEY$=INKEY$:LET PX2=PX+1:LET PY2=PY+(INKEY$=CHR$(11))-(INKEY$=CHR$(10))
1010 IF PX2=31 THEN LET PX2=1
1020 IF PY2<1 THEN LET PY2=1
1030 IF PY2>22 THEN LET PY2=22
1040 CSR PX,PY:PRINT " ";CSR PX2,PY2:PRINT CHR$(129);
1050 LET PX=PX2:LET PY=PY2
1060 IF KEY$<>" " OR MF=1 THEN RETURN ELSE LET MF=1:LET MX=PX:LET MY=PY+1:RETURN
2000 IF MF=0 THEN PAUSE 30:RETURN
2010 LET MX2=MX+1:LET MY2=MY+1
2020 IF MY2=24 THEN GOTO 2100
2030 IF MX2=31 THEN LET MX2=1
2040 CSR MX,MY:PRINT " ";CSR MX2,MY2:PRINT " ";LET MX=MX2:LET MY=MY2
2050 RETURN
2100 IF MX2=TX THEN LET SC=SC+10:CSR TX,23:PRINT " "; ELSE CSR MX,MY:LET MF=0:PRINT " ";RETURN
2110 LET TX=31:LET TC=130:CSR MX,MY:PRINT " ";LET MF=0
2120 RETURN
    
```

To explain the program in full would require the whole magazine as the code above is only one section of the full program. In general, Lines 100 to 200 are the main loop which will GOSUB to each routine in turn, and then go back and repeat everything again.★

MICHAEL RETURNS NEXT MONTH WITH AN EXPLANATION OF THE GAME & FURTHER ROUTINES TO FIRE THE BULLETS AND TIDY UP THE PROGRAM.



# RST 28's THE FACTS John Hudson

John Hudson emerges unshaven, bleary-eyed and in great need of a bath after being buried in the heart of the RST28 jungle for the past 3 months. After reading the following you will see why John deserves the medal for valour and perseverance.

The MTX range of machines uses a floating point method very similar to that of the Sinclair Spectrum. Several books have been written with sections covering the mathematics of that machine, but none that I know of, for the MTX.

This then, is an attempt to rectify that situation. I am NOT a mathematician so please do not write to me about why it has been implemented the way it has been, just try to use it as it is.

## Floating Point Representation.

Computers are capable of performing mathematical operations with numbers of very large magnitude. This is made possible by using more than one register or memory location to hold the 'values' being manipulated.

However, there comes a point where, if the number of storage locations used to represent a number is increased, the least significant bits become very insignificant with respect to the total value.

A better way to represent such a quantity would be to reduce the number of significant digits and indicate the overall magnitude of the number by a power of the number base.

This method of numeric representation is called 'Floating Point'.

A floating point number is made up of two parts, a MANTISSA and an EXPONENT. The mantissa holds the significant digits of the numbers and the exponent holds the power to which the mantissa must be raised.

As an example look at the decimal value 1,000,000. This would require a three byte location to hold it. (0F4240 hex or 11110100001001000000 bin)

Alternatively, this number could be represented as  $1 \times 10^6$  or in floating format as 1.0 E+6. This contains the mantissa of value 1 and the exponent of +6, which indicates the power of 10 to which the mantissa must be raised. It can be seen that this is also the number of places to the right which the decimal place should be moved in order to present the number in it's more normal manner. It could be stored in TWO bytes, :- 0106.

## Binary Floating Point.

In a similar way binary values can be represented in floating format. In this case the binary value of the Exponent is the power of TWO to which the mantissa must be raised. This then indicates the position of the binary point; as before a positive exponent moves right and negative moves left.

It should also be obvious that, if the mantissa is moved to the right, the exponent must be incremented, and if moved to the left, must be decremented.

In order to illustrate this point the following shows the same number expressed in three different ways:-

110010.0 E+0	=	50*1	=	50
.110010 E+6	=	50/8*8	=	50
110010000.0 E-3	=	400*1/8	=	50





In order to obtain the highest degree of accuracy all numbers to be used should be expressed in a standard form prior to starting any mathematical operation. The standard format used is known as the NORMALISED format.

If a number is expressed as .00001001 E+0 it is said to be unnormalised. To normalise it, it must be shifted such that the most significant 1 bit in the mantissa is just to the right of where the implied binary point should be.

In our case this means that it becomes:- 0.1001 E-4.

## THE MTX

The MTX stores all floating point numbers in five bytes, a one byte exponent and a four byte mantissa.

If we look at our value of 50 again this is 110010 in binary and is not normalised. (the binary point must be at the left). After normalisation this becomes .110010 E+6.

The mantissa and exponent now look like this exp=6 mantissa =.11001000.... However, the MTX modifies these values some more before storing them.

For the exponent the top order bit is set to 1 (adds 128 dec to the value) to produce an augmented value.

For the mantissa: as the first bit must always be a one after normalising this bit is used as a sign bit, it is set to a 0 for positive and a 1 for negative.

Our mantissa therefore now looks like this:- .01001000.

Thus in the decimal form the five byte representation is 134,172,0,0,0.

In hexadecimal this should therefore appear as :- 86 48 00 00 00. However, we have not finished yet, as the MTX reverses the order in which the bytes are stored, so it really looks like this :- 00 00 00 48 86.

This can be proved very simply.

Reset the machine.

Now enter the following: -

```
LET A=50: PAN
```

You will now be in 'panel mode'

Type D and in reply to prompt type C002.

The > should now be pointing to the location holding the variable A, in our case floating point 50.

By using this routine you can examine any value you give to A (+,- integer or fraction)

I do not intend to go any further than this, because it gets difficult to show how fractional numbers are generated. (If any club member does have a simple explanation I would be pleased to see it)

## Machine Code Programs

The most difficult thing to do if you program in machine code is to handle mathematical or trigonometrical values. However the MTX has most, if not all, of the routines required to handle such calculations within it's ROMS, many of these being 'callable by means of the RST 28 order.

The RST 28 command works in a very similar manner to the RST 70 command.



As in the RST 10 command the function is followed by a byte or bytes describing the function(s) to be performed.

The byte(s) all have the most significant bit set, and if a further operation is to follow the current one, bit 6 is also set.

e.g.

RST 28

DB £88

this says multiply ACC by OP1

or

RST 228

DB £C8,£82

this says multiply ACC by OP1 and put result to location pointed to by HL.

### Rom Locations.

Most of the routines can be 'called' by using the RST 28 function, but a few can only be called by the normal 'call' function.

The table below shows the RST 28 function qualifier, (if available) and also the call address.

The comments refer to areas ACC, OP1 etc, these are all areas of memory between £FDBC and £FDF1.

(TOS) equates to the top of stack item of the calculator stack.

(DE) equates to the address pointed to by the DE register pair.

(HL) equates to address pointed to by the HL register pair.

(ACC1) refers to the data held in ACC1.

\*\* All operations marked in this way leave the result in ACC1.

DB	Call Add.	Comment.			
B1	12BF	Moves (TOS) to ACC1	9B	12BF	Moves MEM1 to ACC1 (5 bytes)
B2	12D1	Moves ACC1 to MEM1	9C	12A7	Moves (DE) to ACC1 **
B3	23BE	Squares number in ACC1 until carry out of A register.	9F	1274	Moves (TOS) to (DE) (5 bytes)
B4	129E	Moves (TOS) to OP11		12E7	Moves (HL) to (DE) (6 bytes)
B5	1286	Moves ACC1 to (tos		12F3	Moves ACC1 to OP1 (6 bytes)
B6	11B3	Exchange ACC and OP1 (6 bytes)		13B7	Tan (ACC1) to ACC **
B7	23A5	Log (ACC1) base 2		0F47	Normalises ACC1 (5 Bytes)
B8	100B	Multiply ACC1 by OP1 **		0F2E	Rounds (HL) (5 byte pos. numbers only)
BA	109E	Divide ACC1 by OP1 **		0F2B	Rounds ACC1 as above
BB	0EB6	Add ACC1 and OP1 **		0F13	Negates ACC1 (4 bytes)
BE	12B9	Move (DE) to OP1		0DFE	Converts Bin number in ACC1 to ASCII BCD number in COPY
BF	2000	Move 1 to ACC1		20FD	Convert Bin number in ACC1 to ASCII dec no. exponent
90	0EB3	Subtract OP1 from ACC1 **			Result in COPY, terminates with FF
91	13C1	Calculates polynomial with terms of $x+x^3/2+x^5/3$		224C	Converts ASCII fp. no. (DE) to Bin fp. in ACC1 **
94	120B	Move PI..(22/7) to OP11		233F	$X^Y$ X=OP1,Y=ACC1 **
95	115D	Compares ACC1 with OP1 (4 bytes)		241D	Atn (ACC1) **
96	1298	Move OP11 to (TOS)		236E	SQR (ACC1) **
97	1382	Sin (ACC1) to ACC1 **		2398	Log (ACC1) base (e) **
98	137D	Cos (ACC1) to ACC1 **		23BC	Log (ACC1) base 2
99	100B	Multiply ACC1 by ACC1 **		23DF	Calculates $(A-1)/(A+1)$ **
9A	12D2	Executes the polynomial pointed to by HL, the first byte points to how many there are Generally a $1*x/2+x/3+x/4$ type.		23EC	$e^{(ACC1)}$ ** = EXP(x)
				23F2	$2^{(ACC1)}$ **

John concludes his sojourn through the RST28 calls in next months edition when he will present a list of TRIG values and two programs to demonstrate how these calls can be utilised within your own assembly programs. ★





# AN EASY INTERFACE

MIKE BRAY.

This article is written for anyone who, like myself, can only claim to be able to solder, and who gets 'the shakes' at the mention of "address decoding" & "read/write lines". It will, therefore, come as no surprise when I say I was at a complete loss as to where I should start when I first considered a project to connect the MTX to the outside world.

I began by reading an article on interfacing, supplied to me by Keith, and in this article the suggestion came across that the correct course would be to use the PARALLEL INPUT/OUTPUT USER PORT [PIO], and it was suggested, by the article, that this port was the easiest to connect and operate but was not very sophisticated. It could, however, provide 8 output connections and a further 8 inputs. Furthermore, the output could be easily controlled from Basic. E.G typing in the command OUT 7,5 would switch on the lines 0 & 2 ..... 7 is the output port number and Binary 5 = 0000 0101. OUT 7,0 will switch the lines off again. This means by outputting a number between 0-255 will switch on any combination of the output lines.

I decided to attempt a project that would allow the computer to control relays because I felt that once I could connect to relay contacts I was on safe ground as far as my limited knowledge was concerned. Relay control would also allow several future ideas such as a robot, model train control, traffic lights and a Simon type game to be constructed and easily connected to the machine through the interface.

The components I used to make up the interface were purchased from Maplin and Radio Spares. All the parts should be easily obtainable, but to assist anyone wanting to make up the circuit I will quote the part numbers associated with the above mentioned suppliers.

## OUT OF THE COMPUTER

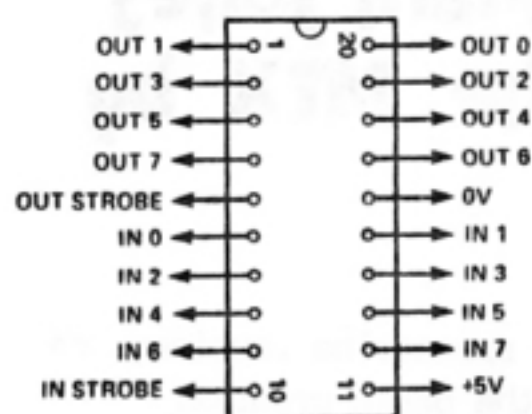
The PIO user port is a 20 pin dual-in-line socket located inside the MTX on the main computer, printed circuit, board. It is mid-way along the board and is in line with the second row of chips counting from the front edge of the computer - on my computer the socket is blue. In order to bring the port connections outside the MTX I used a dual-in-line header plug [RS 402-670] and a ready made up socket and cable purchased from Maplins [20 WAY IDC SKT + CABLE]. The cable should be soldered to the DIL header plug in a logical manner so that the rows & pin numbers correspond to the original port. SEE DIA 1.

At this stage, I must mention a technical point. Before any signal can be obtained from the output port the pin labelled OTSTB [PIN 5] must be shorted to 0V [PIN 16]. This can be achieved by soldering a wire between the two pins [ 5 & 16] on the DIL header plug.

The next stage is to open up the computer by removing the end plates and **carefully** plug the DIL header plug into the socket. Take the cable out through the open slot above the printer plug at the back of the machine and close the the case.

## THE RELAY INTERFACE

The main problem here is that the computer cannot operate relays without some form of outside help. This stumped me for a while and I was given much advice on TRIACS, SCR's etc which did not help at all - I know they work but I had no idea how to use them ! One day, while looking through the Radio Spares catalogue, I came across a semi-conductor circuit, in chip form, that seemed to hold all the answers.



Pin-outs at the User port, a dedicated I/O socket.

max. ratings at 25 °C	7-stage drivers	octal drivers	units
$V_{out}$	50	50	V d.c.
$I_{out}$	500	500	mA
$V_{in}$ with $R_{in}$	30	30	V d.c.
$I_{in(on)}$	1.35	1.35	mA
$I_B$ continuous base current	25	25	mA
PD (total chip)	1	1.8	W
(individual stage)	1	1	W
Derate by (above 25 °C)	16.7	18.8	mW/°C
Operating temperature	0 to +70	0 to +85	°C

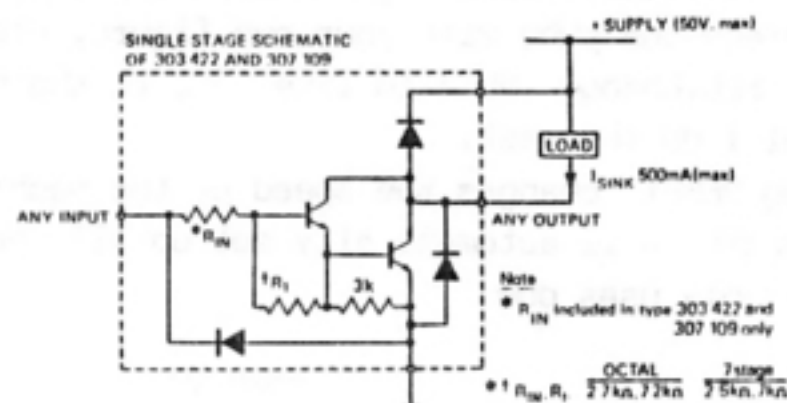
## darlington drivers

7-STAGE  
DRIVER

TOP VIEW  
307-092  
307-109

OCTAL  
DRIVER

TOP VIEW  
303-422  
303-438



The chip [RS 303-422 or ULN 2803A] was comprised of 8 transistor switches [Darlington Driver Pairs], and associated protective circuitry. It was capable of switching up to 500mA loads such as relays and small motors, and could handle input voltages up to 50V. A version of the chip [catalogue numbers above] was designed to connect directly to a computer [TTL] output and included suitable resistors to limit the current from the computer port.

I decided to use 12V as the driving voltage for the relays - I had a ready source of Ni-Cad batteries which I used with my hand-held CB, but I feel that 6V should work just as efficiently.

The relays selected required a 12V supply delivered through a 200 or 250 ohm coil to operate, and their contact layout gave a two-pole change over switch operation when energised. [MAPLIN YX98G or FJ43W]. ★

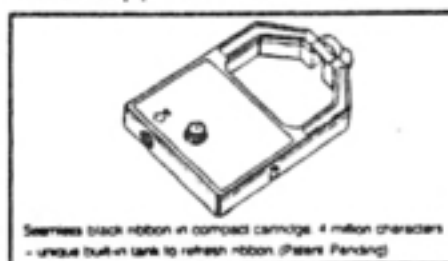
Next month Mike Bray brings the construction to its conclusion with a complete wiring diagram and a small program to test the relays.

# MEMOTECH DMX 80

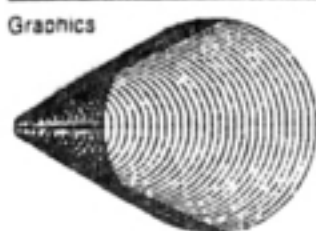
Made in Japan  
Manufactured by Matsushita Electric Industrial Co Ltd Osaka, Japan  
Printed in England

## Special Offer

- Bidirectional minimum-distance access cartridge.
- Compact cartridge-type ribbon designed for long life and easy replacement.
- Seamless, continuous ribbon surface provides uniformly outstanding print quality.
- Precision print head utilizes special wires for sharp, quality printing and greater durability.
- Underlining, elongated characters and skipping between characters.
- The Memotech DMX80 printer is under application for UL and FCC approval.



Graphics



### Character Sets

```

ABCDEFGHIJKLMNPOQRSTUVWXYZ [\ ] ^ _ '
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @
abcdefghijklmnopqrstuvwxyz ( ) ! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @
abcdefghijklmnopqrstuvwxyz ( ) ! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @
abcdefghijklmnopqrstuvwxyz ( ) ! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @

```

\*\*\*\*\*

# £239

INCLUDING 2 METRE INTERFACE CABLE  
AND DELIVERY TO YOUR DOOR BY  
SECURICOR.

SUPPORTS NEWWORD : MEMOSKETCH:  
HOME ACCOUNTS

\*AVAILABLE ONLY TO  
CLUB MEMBERS AT THIS PRICE.

\*\*\*\*\*





# **SPEEDY**

## **Keith Atkinson**



**Written by Mark Jones**  
**Adapted for the MTX by**

The purpose of these routines is to manipulate wire frame graphics at high speeds. The routines can handle up to 16 different figures at once. To see a demonstration, you must RUN the basic program.

To replace the cube with your own figure, refer to "How figure data is calculated" and store the data in line 600. Change NUMB in line 1090 to the number of sets of coordinates in your figure, and the program will do the rest.

POKEing DEMO+1 changes the speed of the machine code demonstration.

The basic will automatically set up all the addresses and HOB/LOB pokes. In fact, it sets up three cubes, but only uses one

MEMORY MAP

16400 4010H	M.C. DEMO ROUTINE 16401 MAY BE POKED TO ALTER SPEED.
16959	CALCULATION ROUTINE TO PRODUCE LIST OF PLOTS AND DRAWS.
17408	ERASE OLD SHAPE ROUTINE.
17421	DRAW NEW SHAPE ROUTINE TO DISPLAY SHAPE FROM LIST OF PLOTS.
17776	SINE TABLE & VARIABLES
18020	PARAMETER BLOCKS. TOTAL - 16 BLOCKS OF 20 BYTES EACH.
SYMBOL TABLE	
BASIC PROGRAM	
53248 0000H	DATA STORAGE AREA.

N.B IMPORTANT VARIABLE AT 17960 => FIG NO

### HOW FIGURE DATA IS CALCULATED

To work out the data for a figure, the rules are: positive is left, negative is right, positive is up, negative is down, positive is away from you and negative is towards you. Imagine a point somewhere near the centre of mass of the figure and work out the 3D coordinates of each "corner" of the figure, RELATIVE TO THE IMAGINARY POINT, using the positive and negative rules. Look at the data in the program as an example.

For each "corner" you should now have an X, a Y and a Z coordinate. The fourth number to complete each set is either: 0 to plot the point only, : 1 to draw a line from the last point plotted, or from the end of the last line, to X, Y, Z.

### HOW FIGURE DATA IS STORED IN MEMORY

The data for the figure is stored by the program as a list of blocks of 7 bytes as follows :



OFFSET	NO OF BYTES	DESCRIPTION
0	2	X COORD AS 2's COMPLEMENT
2	2	Y COORD AS 2's COMPLEMENT
4	2	Z COORD AS 2's COMPLEMENT
6	1	FLAG ( 0 or 1 )
ETC		

A 2's complement number is a Zilog convention 2 byte number, where a negative number is stored as 65536 minus the number, ie -26 is 65536-26. Look at lines 500 onwards for the subroutine to do this.

After the sets of data for a particular figure, there must be an area of free memory for that figure. The length of this area must be (12 \* number of sets of data) bytes. So, altogether, each figure needs (19 \* number of sets of data) bytes. The free area is pointed to by DRAWS and DRAWP (see Parameters), and is where the actual plots and draws for the rapid drawing routine are stored for that figure.

#### THE PARAMETER TABLES

The parameters are stored at 18020 onwards, and there are 16 blocks of 20 bytes each. The functions of the parameters are best shown in a table as follows :-

OFF SET	NO OF BYTES	PARAM	DESCRIPTION	RANGE
0	1	NUMB	NUMBER OF SETS OF DATA	1 - 255
1	2	ADDR	START OF FIGURE DATA	
3	2	PX	X COORD [ +ve LEFT ]	
5	2	PY	Y COORD [ +ve UP ]	
7	2	PZ	Z COORD [ +ve FORWARD ]	
9	2	PHI	ANGLE ABOUT X AXIS	0 - 359 °
11	2	THETA	ANGLE ABOUT Y AXIS	0 - 359 °
13	2	PSI	ANGLE ABOUT Z AXIS	
15	2	DRAWS	ADD OF FREE MEMORY AFTER SETS OF DATA	
17	2	DRAWP	6 * NUMB + DRAWS	
19	1	STFLG	POKE THIS WITH 0 THE FIRST TIME YOU USE FIGURE	0 - 255

#### THE BASIC

Typing RUN shows off a demo. To call the code from your own programs, (refer to lines 1900-), adopt the following procedure.

1. POKE FIGNO (see program) with the figure number you want to use.
2. IF IT IS BEING USED FOR THE FIRST TIME, POKE STFLG for that figure with 0.
3. CALL the calculations routine.
4. CALL the delete old shape routine.
5. CALL the draw new shape routine.

The attitude and displacement of that figure are altered by POKEing the relevant parameters for that figure.

Lines 2000 onwards show you how to use the fast drawing routine for yourself. ie :-



1. SELECT an unused figure.
2. POKE FIGNO with the figure number.
3. Select some unused memory.
4. POKE DRAW5 and DRAWP for that figure with the start address of that memory (both the same).
5. Store the drawing data at this address.
6. POKE STFLG with 255.
7. CALL the draw new shape routine.

The drawing data is stored in sets of 3 bytes in the form P,X,Y,where:

P = 0 to plot X,Y.  
P = 1 to draw from last point to X,Y.  
P = 255 to end the data.  
X = normal x coordinate between 0 and 255.  
Y = normal Y coordinate between 0 and 191.

The data in the program draws a frame around the screen. All drawings are in graphics mode 2.  
Finally, my thanks to Mark Jones, who wrote this excellent routine, for his help, and his permission to translate it for Memopad. ★

10 GOTO 200	407A LD DE,(PZ)	40E8 ADD HL,HL	415B INC A
100 CODE	407E ADD HL,DE	40E9 EX DE,HL	415C LD BC,180
	407F BIT 7,H	40EA JR MLPA	415F SBC HL,BC
4010 DEMO: LD A,10	4081 CALL NZ,CPL	40EC MLPBE: EX AF,AF'	4161 JR NC,COND
4012 CP 250	4084 LD A,(IY+0)	40ED JR NC,MLPBF	4163 DEC A
4014 JR C,OKDEM	4087 LD (Z),HL	40EF ADD HL,DE	4164 ADD HL,BC
4016 LD A,250	408A LD (VARST),HL	40F0 ADC A,0	4165 COND: LD BC,TABLE
4018 OKDEM: LD B,0	408D LD HL,(X)	40F2 MLPBF: LD B,L	4168 ADD HL,BC
401A LD C,A	4090 ADD HL,HL	40F3 LD L,H	4169 LD D,(HL)
401B LD HL,254	4091 ADD HL,HL	40F4 LD H,A	416A EX DE,HL
401E KLP: PUSH BC	4092 LD DE,(PX)	40F5 RET	416B LD L,A
401F PUSH HL	4096 ADD HL,DE	40F6 CALC2: EX DE,HL	416C RET
4020 POP IX	4097 PUSH AF	40F7 BIT 0,(IY+0)	416D EQUIN: BIT 0,(IY+2)
4022 PUSH HL	4098 CALL CALC3	40F8 JR Z,POS	4171 JR Z,GTSIN
4023 CALL FETCH	409B LD DE,128	40FD ADD HL,DE	4173 LD L,(IX+3)
4026 PUSH HL	409E CALL CALC2	40FE RES 0,(IY+0)	4176 LD A,(IX+2)
4027 PUSH DE	40A1 LD (X),HL	4102 RET	4179 JR GALL
4028 PUSH BC	40A4 LD HL,(Y)	4103 POS: SBC HL,DE	417B GTSIN: LD L,(IX+1)
4029 LOIR	40A7 ADD HL,HL	4105 RET	417E LD A,(IX+0)
402B PUSH IX	40AB ADD HL,HL	4106 SINC: PUSH HL	4181 GALL: LD H,0
402D POP HL	40A9 LD DE,(PY)	4107 LD A,H	4183 LD (VARST),HL
402E LD (PZ),HL	40AD ADD HL,DE	4108 OR L	4186 LD (IY+0),A
4031 LD (PSI),HL	40AE POP AF	4109 JR Z,ZERO	4189 EX DE,HL
4034 LD (THETA),HL	40AF LD (IY+0),A	410B CALL COS	418A CALL CALC3
4037 LD (PHI),HL	40B2 CALL CALC3	410E LD (COSST),HL	418D BIT 0,(IY+0)
403A LD DE,255	40B5 HORIZ: LD DE,87	4111 POP HL	4191 RET Z
403D EX DE,HL	40B8 CALL CALC2	4112 CALL SIN	4192 CPL: XOR A
403E SBC HL,DE	40BB LD (Y),HL	4115 LD (SINST),HL	4193 LD D,A
4040 SRL H	40BE RET	4118 LD HL,SINST	4194 LD E,A
4042 RRL	40BF CALC3: BIT 7,H	411B LD BC,4	4195 EX DE,HL
4044 SRL H	40C1 CALL NZ,CPL	411E RET	4196 SBC HL,DE
4046 RRL	40C4 EX DE,HL	411F ZERO: POP HL	4198 INC (IY+0)
4048 LD (PX),HL	40C5 LD HL,(VARST)	4120 LD A,128	4199 RET
404B LD (PY),HL	40CB MULTY: PUSH HL	4122 LD (SINST),A	419C TRUN: BIT 7,(IX+0)
404E POP BC	40C9 SBC HL,DE	4125 LD HL,SINST	41A0 RET NZ
404F POP HL	40CB POP HL	4128 LD BC,1	41A1 PUSH BC
4050 POP DE	40CC JR C,MULT	412B RET	41A2 CALL EQUIN
4051 LOIR	40CE EX DE,HL	412C CREAT: LD HL,(THETA)	41A5 INC (IY+2)
4053 CALL RUN	40CF MULT: LD B,H	412F CALL SINC	41AB POP DE
4056 CALL UNDRW	40D0 LD C,L	4132 LD DE,THEST	41A9 PUSH HL
4059 CALL DOSH	40D1 XOR A	4135 LOIR	41AA CALL EQUIN
405C POP HL	40D2 EX AF,AF'	4137 LD HL,(PSI)	41AD POP DE
405D POP BC	40D3 XOR A	413A CALL SINC	41AE BIT 0,(IY+2)
405E SBC HL,BC	40D4 EX AF,AF'	413D LD DE,PSIST	41B2 JR Z,SUB
4060 LD A,H	40D5 LD L,A	4140 LOIR	41B4 ADD HL,DE
4061 BIT 7,A	40D6 LD H,A	4142 LD HL,(PHI)	41B5 XOR A
4063 JR NZ,ALL2	40D7 MLPA: SRL B	4145 CALL SINC	41B6 RET
4065 CP 0	40D9 RRL	4148 LD DE,PHIST	41B7 SUB: EX DE,HL
4067 JR NZ,KLP	40DB JR NZ,NTEST	414B LOIR	41B8 SBC HL,DE
4069 LD A,C	40DD EX AF,AF'	414D RET	41BA XOR A
406A CP L	40DE CP C	414E COS: LD BC,90	41BB RET
406B JR C,KLP	40DF JR Z,MLPBE	4151 ADD HL,BC	41BC EQUIN: LD IX,THEST
406D ALL2: RET	40E1 MLPBD: EX AF,AF'	4152 LD BC,360	41CD LD DE,(X)
406E CALCT: RES 0,(IY+0)	40E2 NTEST: JR NC,MLPBB	4155 SBC HL,BC	41C4 LD BC,(Z)
4072 LD HL,(Z)	40E4 ADD HL,DE	4157 JR NC,SIN	41CB SET 0,(IY+2)
4075 LD DE,256	40E5 ADC A,0	4159 ADD HL,BC	41CC CALL TRUN
4078 SBC HL,DE	40E7 MLPBB: EX DE,HL	415A SIN: XOR A	41CF JR NZ,ZERO1



4101	LD DE,(X)	4290	LD A,(FLAGO)	4326	POP HL	430F	INC BC
4105	LD (X),HL	4293	LD C,A	4327	POP BC	4300	JR DIV1
4108	LD BC,(Z)	4294	LD HL,(X)	4328	LD (OY),BC	4302 DIV2:	ADD HL,DE
410C	CALL TRUN	4297	LD DE,(Y)	432C	POP BC	4303	ADD HL,HL
410F	LD (Z),HL	4298	XOR A	432D	LD (OX),BC	4304	SBC HL,DE
41E2	LD B,H	429C	OR H	4331	POP BC	4306	JR C,DIV3
41E3	LD C,L	429D	OR D	4332	XOR A	4308	INC BC
41E4 ZERO1:	LD IX,PHIST	429E	JR NZ,TST1	4333	LD (BC),A	4309 DIV3:	LD L,C
41E8	LD DE,(Y)	42A0	LD A,191	4334	INC BC	430A	LD H,B
41EC	SET O,(IY+2)	42A2	CP E	4335	LD A,(GRADX)	430B	RET
41FO	CALL TRUN	42A3	LD A,0	4338	LD (BC),A	430C ZERO:	LD HL,0
41F3	JR NZ,ZERO2	42A5	JR NC,TST2	4339	INC BC	430F	RET
41F5	LD DE,(Y)	42A7 TST1:	LD A,1	433A	LD A,(GRADX1)	43E0 CPL2:	PUSH DE
41F9	LD (Y),HL	42A9 TST2:	LD (FLAGO),A	433D	LD (BC),A	43E1	CALL CPL
41FC	LD BC,(Z)	42AC	OR C	433E	INC BC	43E4	POP DE
4200	CALL TRUN	42AD	JR NZ,OFFSC	433F	JP COIY	43E5	RET
4203	LD (Z),HL	42AF	LD (OX),HL	4342 DOCUT:	PUSH DE	43E6 MULT2:	BIT 7,H
4206 ZERO2:	LD BC,(Y)	42B2	LD (OY),DE	4343	BIT 7,H	43E8	CALL NZ,CPL2
420A	LD IX,PSIST	42B6	POP BC	4345	JR Z,DOCO	43EB	EX DE,HL
420E	LD DE,(X)	42B7 COIY:	POP AF	4347	CALL DONE	43EC	BIT 7,H
4212	SET O,(IY+2)	42B8	LD (BC),A	434A	LD HL,0	43EE	CALL NZ,CPL2
4216	CALL TRUN	42B9	INC BC	434D DOCO:	LD A,H	43F1	JP MULTY
4219	RET NZ	42BA	LD A,L	434E	AND A	43F4 NOORW:	LD (OX),HL
421A	LD DE,(X)	42BB	LD (BC),A	434F	JR Z,DOCI	43F7	LD (OY),DE
421E	LD (X),HL	42BC	INC BC	4351	LD DE,255	43FB	POP DE
4221	LD BC,(Y)	42BD	LD A,E	4354	AND A	43FC	POP AF
4225	CALL TRUN	42BE	LD (BC),A	4355	SBC HL,DE	43FD	JP ENDY
4228	LD (Y),HL	42BF	INC BC	4357	CALL DONE	4400 UNORW:	CALL FETC2
422B	RET	42C0	LD E,C	435A	LD HL,255	4403	PUSH HL
422C FETCH:	LD A,(FIG)	42C1	LD D,B	435D DOC1:	PUSH HL	4404	LD C,#2F;CPL
422F	LD HL,PARAMO	42C2 ENDY:	POP HL	435E	LD HL,(GRADX)	4406	LD B,#A1;AND C
4232	LD DE,NUMB	42C3	POP BC	4361	LD DE,(GRADY)	4408	LD HL,(DRAWS)
4235	LD BC,20	42C4	OJNZ BEGIN	4365	LD (GRADX),DE	440B	JR DRWAL
4238	AND A	42C6	LD A,255	4369	LD (GRADY),HL	440D DOOHP:	CALL FETC2
4239	RET Z	42C8	LD (DE),A	436C	POP DE	4410	PUSH HL
423A LPFG:	ADD HL,BC	42C9	RET	436D	POP HL	4411	LD C,#00
423B	DEC A	42CA OFFSC:	LD A,H	436E	PUSH DE	4413	LD B,#B1;OR C
423C	JR NZ,LPFG	42CB	AND A	436F	BIT 7,H	4415	LD HL,(DRAWP)
423E	RET	42CC	JR Z,OKQ1	4371	JR Z,DOC1A	4418	LD DE,(DRAWS)
423F RUN:	CALL FETCH;MAIN ENTRI	42CE	LD A,(OX1)	4373	CALL DONE	441C	LD (DRAWS),HL
4242	LDIR	42D1	AND A	4376	LD HL,0	441F	LD (DRAWP),DE
4244	LD IY,FLAGS	42D2	JR Z,OKQ1	4379 DOC1A:	LD A,H	4423 DRWAL:	LD (MOD),BC
424B	LD A,(STFLG)	42D4	XOR H	437A	AND A	4427	EXX
424B	AND A	42D5	RLA	437B	JR NZ,DOC2	4428	PUSH HL
424C	JR NZ,STOK2	42D6	JP NC,NOORW	437D	LD A,L	4429	EXX
424E	LD A,255	42D9 OKQ1:	LD A,D	437E	CP 192	442A LPPQ:	LD A,(HL);P
4250	DEC HL	42DA	AND A	4380	JR C,DOC3	442B	INC HL
4251	LD (HL),A	42DB	JR Z,OKQ2	4382 DOC2:	LD DE,191	442C	CP #FF;FF=DATAEND
4252	LD HL,(DRAWS)	42DD	AND A	4385	AND A	442E	JR Z,EXDR
4255	LD (HL),A	42E0	JR Z,OKQ2	4386	SBC HL,DE	4430	LD C,(HL);X COORD
4256 STOK2:	XOR A	42E1	JR Z,OKQ2	4388	CALL DONE	4431	INC HL
4257	LD (FLAGO),A	42E3	XOR D	438B	LD HL,191	4432	LD B,(HL);Y COORD
425A	CALL CREAT	42E4	RLA	438E DOC3:	PUSH HL	4433	INC HL
425D	LD DE,(DRAWP)	42E5	JP NC,NOORW	438F	LD HL,(GRADX)	4434	PUSH HL
4261	LD A,(NUMB)	42E8	JR OKQ3	4392	LD DE,(GRADY)	4435	RRR
4264	LD B,A	42EA OKQ2:	LD A,E	4396	LD (GRADX),DE	4436	JR C,DRAWE
4265	LD HL,(ADDR)	42EB	CP 192	439A	LD (GRADY),HL	4438	LD (COORD),BC
426B BEGIN:	PUSH BC	42ED	JR C,OKQ3	439D	POP DE	443C DRAWE:	CALL DRAW
4269	PUSH DE	42EF	LD A,(OY)	439E	POP HL	443F EEXT:	POP HL
426A	LD E,(HL)	42F2	CP 192	439F	RET	4440	JR LPPQ
426B	INC HL	42F4	JP NC,NOORW	43AD DONE:	RES O,(IY+0)	4442 EXDR:	EXX
426C	LD D,(HL)	42F7 OKQ3:	PUSH HL	43A4	LD DE,(GRADY)	4443	POP HL
426D	LD (X),DE	42F8	PUSH DE	43A8	CALL MULT2	4444	EXX
4271	INC HL	42F9	PUSH HL	43AB	LD H,L	4445	POP DE
4272	LD E,(HL)	42FA	LD HL,(OY)	43AC	LD L,B	4446	LD HL,DRAWS
4273	INC HL	42FD	AND A	43AD SERY:	LD DE,(GRADX)	4449	LD BC,#0004
4274	LD D,(HL)	42FE	SBC HL,DE	43B1	CALL DIVY	444C	LDIR
4275	LD (Y),DE	4300	LD (GRADY),HL	43B4	POP BC	444E	RET
4279	INC HL	4303	POP DE	43B5	POP DE	444F FETC2:	CALL FETCH
427A	LD E,(HL)	4304	LD HL,(OX)	43B6	CALL CALC2	4452	LD BC,#000F
427B	INC HL	4307	AND A	43B9	PUSH HL	4455	ADD HL,BC
427C	LD D,(HL)	430B	SBC HL,DE	43BA	PUSH BC	4456	LD DE,DRAWS
427D	LD (Z),DE	430A	LD (GRADX),HL	43BB	RET	4459	LD BC,#0005
4281	INC HL	430D	EX DE,HL	43BC DIVY:	EX DE,HL	445C	PUSH HL
4282	PUSH HL	430E	LD DE,(Y)	43BD	BIT 7,H	445D	LDIR
4283	CALL EQUUS	4312	CALL DOCUT	43BF	CALL NZ,CPL2	445F	LD A,(STFLG)
4286	CALL CALCT	4315	PUSH HL	43C2	EX DE,HL	4462	AND A
4289	POP HL	4316	PUSH DE	43C3	LD A,D	4463	JR NZ,STOK
428A	LD A,(HL)	4317	LD HL,(OX)	43C4	OR E	4465	DEC HL
428B	INC HL	431A	LD DE,(OY)	43C5	JR Z,ZERO	4466	LD A,#FF
428C	POP DE	431E	CALL DOCUT	43C7	LD BC,0	4468	LD (HL),A
428D	PUSH HL	4321	LD H,E	43CA	AND A	4469	LD HL,(DRAWS)
428E	PUSH AF	4322	LD (GRADX),HL	43CB DIV1:	SBC HL,DE	446C	LD (HL),A
428F	PUSH DE	4325	POP DE	43CD	JR C,DIV2	446D	LD HL,(DRAWP)





```

4470 LD (HL),A
4471 STOK: POP HL
4472 RET
4473 NORAW: POP BC
4474 POP BC
4475 EI
4476 RET
4477 DRAW: DI
4478 PUSH BC
4479 LD BC,(COORD)
447D PUSH BC
447E LD A,191
4480 SUB B
4481 JR C,NORAW
4483 CALL VPADDR
4486 LD B,A;DATA BYTE
4487 INC B
4488 LD A,#01
448A REMLP: RRCA
448B DJNZ REMLP;CALC DATA
448D PUSH HL;ADDR
448E EXX
448F POP HL;ADDR IN ALT REG SET
4490 EXX
4491 EX AF,AF';SAVE DATA IN A
4492 POP HL;HL=(FDEF)
4493 POP BC;BC=VAL ON ENTRY
4494 LD (COORD),BC;SAVE LINE END
449B LD DE,257
449B LD A,B
449C SUB H
449D JR NC,LAB2
449F NEG
44A1 LD D,#FF
44A3 LAB2: LD B,A
44A4 LD A,C
44A5 SUB L
44A6 JR NC,LAB3
44A8 NEG
44AA LD E,#FF
44AC LAB3: LD C,A
44AD SET: CP B
44AE JR NC,LAB4
44B0 LD L,C
44B1 PUSH DE
44B2 EXX
44B3 POP DE
44B4 EXX
44B5 XDR A
44B6 LD E,A
44B7 JR LAB5
44B9 LAB4: OR C
44BA JP Z,END;EARLY EXIT
44BD LD L,B
44BE LD B,C;SMALL SIDE
44BF PUSH DE
44C0 EXX
44C1 POP DE
44C2 EXX
44C3 LD D,#00
44C5 LAB5: LD H,B;LARGE SIDE
44C6 LD A,B;B=COUNT
44C7 RRA;A=H/2
44C8 LOOP: ADD A,L;L=SMALL SIDE
44C9 JR C,LAB6
44CB CP H;A>H?
44CC JR C,LAB7
44CE LAB6: SUB H
44CF LD C,A;STA
44D0 EXX
44D1 LD C,E
44D2 LD B,D
44D3 JR LAB8
44D5 LAB7: LD C,A
44D6 PUSH DE
44D7 EXX
44D8 POP BC
44D9 LAB8: BIT 0,B
44DB JR Z,ENDEC
44DD BIT 7,B;B CONTAINS ADDRESS FLAGS
44DF JR NZ,DECRAD
44E1 INCRAD: LD A,L
44E2 AND #0F
44E4 CP #0F
44E6 JR Z,COMPI

```

```

44E8 CP 07
44EA JR Z,COMPI
44EC SIMPI: INC L
44ED JR ENDEC
44EF COMPI: LD A,L
44F0 SUB 07
44F2 LD L,A
44F3 INC H
44F4 JR ENDEC
44F6 DECRAD: LD A,L
44F7 AND #0F
44F9 CP 08
44FB JR Z,COMPO
44FD AND 07
44FF JR Z,COMPO
4501 SIMPO: DEC L
4502 JR ENDEC
4504 COMPO: LD A,L
4505 ADD A,07
4507 LD L,A
4508 DEC H
4509 ENDEC: EX AF,AF';RECOVER DATA
450A BIT 0,C
450C JR Z,ENOROT
450E BIT 7,C;C CONTAINS ROTATE FLAGS
4510 JR NZ,LEFT
4512 RRCA
4513 JR NC,ENOROT
4515 PUSH BC
4516 LD B,#00
4518 LD C,#08
451A ADD HL,BC
451B POP BC
451C JR ENOROT
451E LEFT: RLCA
451F JR NC,ENOROT
4521 PUSH BC
4522 LD B,#00
4524 LD C,#08
4526 AND A;CLEAR CARRY
4527 SBC HL,BC
4529 POP BC
452A ENOROT: LD B,A;DATABYTE
452B VOPXOR: PUSH DE
452C PUSH BC
452D PUSH HL
452E POP DE
452F PUSH AF
4530 LD A,E;SET READ
4531 OUT (#02),A
4533 LD A,D
4534 OUT (#02),A
4536 PUSH HL
4537 POP DE
4538 IN A,(#01);DO READ
453A LD C,A;STORE SCRIN
453B POP AF;POP DATA
453C MOD: NOP
453D OR C
453E LD C,A;STORE
453F LD A,E;SET WRITE
4540 OUT (#02),A
4542 LD A,D
4543 OR #40
4545 OUT (#02),A
4547 LD A,C;DATA
4548 POP BC
4549 POP DE
454A OUT (#01),A;WRITE
454C LD A,B;PRESERVE THE MASK IN A
454D EX AF,AF';PRESERVE THE MASK
454E EXX;PRESERVE THE ADDR
454F LD A,C;RUNNING TOTAL
4550 DEC B
4551 JP NZ,LOOP
4554 END: EI
4555 RET
4556 NOP
4557 VPADDR: LD A,C;X COORD
4558 AND #FB
455A LD E,A;HORIZ OFFSET
455B LD A,C;X COORD
455C AND #07
455E LD D,A;REMAINDER
455F LD A,B;Y COORD

```

```

4560 AND #FB
4562 RRCA
4563 RRCA
4564 RRCA
4565 LD H,A;INT Y/8
4566 LD A,B;YCOORD
4567 AND #07
4569 LD L,A
456A LD A,D;DATA
456B LD D,#00
456D ADD HL,DE;VRAM ADDR
456E RET
456F NOP

```

```

4570 TABLE: DB 0,4,8,13,17,22,26,31,35,40,44,48,53,57,61,66,70
4581 DB 74,79,83,87,91,95,100,104,108,112,116,120,124
458E DB 127,131,135,139,143,146,150,154,157,161,164,167
459A DB 171,174,177,181,184,187,190,193,196,198,201,204
45A6 DB 207,209,212,214,217,219,221,223,226,228,230,232
45B2 DB 233,235,237,238,240,242,243,244,246,247,248,249
45BE DB 250,251,252,252,253,254,254,255,255,255
45C8 DB 255,255,255,255,255,255,255,255,254,254,253,252
45D4 DB 252,251,250,249,248,247,246,244,243,242
45DE DB 240,238,237,235,233,232,230,228,226,223,221,219
45EA DB 217,214,212,209,207,204,201,198,196,193
45FA DB 190,187,184,181,177,174,171,167,164,161,157
45FF DB 154,150,146,143,139,135,131,128,124,120,116
460A DB 112,108,104,100,95,91,87,83,79,74,70,66,61,57
4618 DB 53,48,44,40,35,31,26,22,17,13,8,4,0,0
4626 NOP

```

```

4627 NOP
4628 FIG: DB 0
4629 NUMB: DB 0
462A ADDR: DB 0,0
462C PX: DB 0,0
462E PY: DB 0,0
4630 PZ: DB 0,0
4632 PHI: DB 0,0
4634 THETA: DB 0,0
4636 PSI: DB 0,0
4638 DRAWS: DB 0,0
463A DRAWP: DB 0,0
463C STFLG: DB 0
463D VARST: DB 0,0
463F GRADY: DB 0,0
4641 GRADX: DB 0
4642 GRADX1: DB 0
4643 OX: DB 0
4644 OX1: DB 0
4645 OY: DB 0
4646 OY1: DB 0
4647 COORD: DB 0,0
4649 SINST: DB 0,0
464B COSST: DB 0,0
464D THEST: DS 4
4651 PSIST: DS 4
4655 PHIST: DS 4
4659 X: DB 0,0
465B Y: DB 0,0
465D Z: DB 0,0
465F FLAGO: DB 0
4660 FLAG: DB 0,0,0,0
4664 PARAM0: DS 20
4678 PARAM1: DS 20
468C PARAM2: DS 20
46A0 PARA3: DS 20
46B4 PARA4: DS 20
46C8 PARA5: DS 20
46DC PARA6: DS 20
46F0 PARA7: DS 20

```

THE FINAL PART OF  
THIS PROGRAM WILL  
APPEAR IN SEPTEMBER



## 3D GRAPHICS

J.R. Majors

THIS IS THE PROGRAM THAT FOLLOWS ON FROM JOHN'S ARTICLE IN ISSUE NUMBER 9.

JOHN WILL BE BACK NEXT MONTH WITH A FURTHER STUDY OF 3D GRAPHICS WHEN HE DELVES INTO HOW TO ACTUALLY DEFINE A 3D IMAGE WHICH CAN THEN BE VIEWED FROM ANY ANGLE.

```

10 REM Use different values for theta and phi (between 0 and 6.283) and you will
   see the cube rotate
20 REM Change the data by storing xyz coordinates for start and finish of lines
   in pairs and alter the loop indices to match the no of lines - make new shapes
100 REM DATA FOR THE DRAWING OF A CUBE
120 REM 3 VALUES FOR EACH POINT, TWO POINTS FOR EACH LINE, TWELVE LINES FOR THE
   CUBE
130 DATA 0,0,0,0,1,0,0,1,0,1,1,0,1,1,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1,
   1,0,1,1,1,1,1,1,1,1,1,0,1,1,0,1,0,0,1
135 DATA 0,1,1,0,1,0,1,1,1,1,1,0,1,0,1,1,0,0,0
140 REM READ DATA IN TWO ARRAYS FOR ENDS OF LINES
142 REM DIMENSION ARRAYS FOR X,Y,Z COORDINATES FOR (S)TART AND (F)INISH OF LINES
145 DIM XS(12): DIM YS(12): DIM ZS(12): DIM XF(12): DIM YF(12): DIM ZF(12)
146 RESTORE 130
150 FOR N=1 TO 12
160 READ XS(N),YS(N),ZS(N),XF(N),YF(N),ZF(N)
170 NEXT N
510 REM input values for theta, phi, distance to screen and distance to observer
515 CLS
520 INPUT "input theta - any value";THETA
530 INPUT "input phi -any value";PHY
540 INPUT "input distance to screen -(eg 3000) ";DISTSCR
550 INPUT "input distance to observer -(eg 60) ";DISTOBS
560 REM PRECALCULATE VALUES WE WILL USE FREQUENTLY IN THE PROGRAM
570 LET ST=SIN(THETA): LET SP=SIN(PHY): LET CT=COS(THETA): LET CP=COS(PHY)
1000 REM PLOT CUBE
1010 CRVS 6,1,0,0,32,24,32: VS 6: CLS
1015 REM DRAW 12 LINES
1020 FOR N=1 TO 12
1030 LET X=XS(N): LET Y=YS(N): LET Z=ZS(N)
1040 GOSUB 2000
1050 LET SX=XS(N): LET SY=YS(N)
1060 LET X=XF(N): LET Y=YF(N): LET Z=ZF(N)
1070 GOSUB 2000
1080 LINE SX,SY,SX,SY
1090 NEXT N
1100 CSR 10,20: INPUT "Press RET to repeat";K$: GOTO 510
2000 REM SUBROUTINE TO CONVERT THREE COORDINATES OF POINT TO TWO SCREEN COORDINA
   TES
2010 LET NEWX=X*ST+Y*CT
2020 LET NEWY=X*CT*CP-Y*ST*CP+Z*SP
2030 LET NEWZ=X*CT*SP+Y*ST*SP+Z*CP+DISTOBS
2040 REM CONVERT TO SCREEN COORDINATES
2050 LET SX=DISTSCR*NEWX/NEWZ
2060 LET SY=DISTSCR*NEWY/NEWZ
2070 REM ADD OFFSET TO PUT IN CENTRE OF OUR SCREEN
2075 LET SX=SX*192/256
2080 LET SX=SX+124: LET SY=SY+90
2100 RETURN

2900 REM data given below for pyramid
3000 DATA 0,0,0,0,1,0,0,1,0,1,1,0,1,1,0,1,0,0,1,0,0,0,0,0,0,0,0,0,.5,.5,.5,.5,
   .5,1,0,0,.5,.5,.5,1,1,0,.5,.5,.5,0,1,0

```





## SOUND GENERATOR

INPUT CODES FOR the

CHROMATIC SCALE DICK EASSOM

```

10 REM *** SOUND GENERATOR INPUT CODES FOR THE CHROMATIC SCALE
20 REM *** Dick Eassom, Bicknacre, 18th November 1984
50 DIM F(10),B(12)
100 LET SCALE$="C C#D D#E F F#G G#A A#B "
110 LET HEX$="0123456789ABCDEF": LET PAGE=1
120 LET FREQ=123.47: REM B below lowest Bve
130 LPRINT CHR$(27);"D";CHR$(6);CHR$(14);CHR$(29);"% ,AH";CHR$(0): REM Set tabs f
or Epson MX-80F/T
140 FOR Z=1 TO 4: LPRINT "": NEXT Z
150 LPRINT "          SOUND TABLES FOR THE TI SN76489A SOUND GENERATOR I.C."
:CHR$(13);
155 LPRINT "          ":::: FOR Z=1 TO 53: LPRINT "_";: NEXT Z
160 LPRINT "": LPRINT ""
170 GOSUB 1400: REM LPRINT page header
180 LET CLK=4E6
200 REM *** main routine
210 FOR OCT=-1 TO 4
215 IF OCT=3 AND PAGE=1 THEN GOSUB 1700
220 FOR NOTE=1 TO 12
230 LPRINT " ";OCT;CHR$(9);SCALE$(NOTE*2-1,2);: REM CHR$(9) = LPRINT TAB
240 GOSUB 1200: LPRINT CHR$(9);FREQ$;
250 GOSUB 1000: LPRINT CHR$(9);DEC$;CHR$(9);BUFF$;
260 GOSUB 1500: GOSUB 1600
270 LPRINT CHR$(9);BIN$;
280 GOSUB 1100: LET BYTE1$="&"+NIB$(3): LET BYTE2$=NIB$(2)+NIB$(1)
290 LPRINT CHR$(9);BYTE1$;CHR$(9);BYTE2$
300 NEXT NOTE: NEXT OCT
900 GOSUB 1800: STOP
1000 REM *** Convert decimal [FREQ] to binary [F(10)....F(1)]
1010 LET N=CLK/(32*FREQ): LET N=INT(N+.5): GOSUB 1300
1020 FOR Z=1 TO 10: LET F(Z)=MOD(N,2): LET N=INT(N/2): NEXT Z
1030 RETURN
1100 REM *** Convert binary [B(12)....B(1)] to hex [NIB$(3)....NIB$(1)]
1110 FOR NIB=3 TO 1 STEP -1
1120 LET DEC=B*B(NIB*4)+4*B(NIB*4-1)+2*B(NIB*4-2)+B(NIB*4-3)
1130 LET NIB$(NIB)=HEX$(DEC+1): NEXT NIB
1140 RETURN
1200 REM *** Calculate new frequency. Returns FREQ and FREQ$
1210 LET FREQ=FREQ*2^(1/12)
1220 LET FIX=INT((FREQ-INT(FREQ))*10+.5)/10+INT(FREQ)
1230 LET FREQ$=STR$(FIX): LET FREQ$=RIGHT$(FREQ$,LEN (FREQ$)-1)
1240 IF FIX=INT(FIX) THEN LET FREQ$=FREQ$+".0"
1250 IF LEN (FREQ$)=5 THEN LET FREQ$=" "+FREQ$
1260 RETURN
1300 REM *** Format decimal into DEC$ & BUFF$
1310 LET DEC$=STR$(N): LET BUFF$=STR$(N*8)
1320 LET DEC$=RIGHT$(DEC$,LEN (DEC$)-1): REM Strip off '+' sign
1340 IF LEN (DEC$)=2 THEN LET DEC$=" "+DEC$
1350 IF LEN (DEC$)=3 THEN LET DEC$=" "+DEC$
1360 LET BUFF$=RIGHT$(BUFF$,LEN (BUFF$)-1): REM Strip off '+' sign
1370 IF LEN (BUFF$)=2 THEN LET BUFF$=" "+BUFF$
1380 IF LEN (BUFF$)=3 THEN LET BUFF$=" "+BUFF$
1390 RETURN
1400 REM *** LPRINT page header
1410 GOSUB 1450: REM LPRINT 80 underlines
1420 LPRINT "Bve note frequency, Hz          MTX          POKE (binary)          POK
E (hex.)"
1430 FOR Z=1 TO 28: LPRINT " ":::: NEXT Z: LPRINT "direct  SBUF  byte 1  byte
2  byte 1 byte 2";CHR$(13);
1440 GOSUB 1450: LPRINT "": RETURN
1450 FOR Z=1 TO 80: LPRINT " ":::: NEXT Z: RETURN
1500 REM *** Convert binary [F(10)....F(1)] to binary [B(12)....B(1)]
1510 FOR Z=12 TO 9 STEP -1: LET B(Z)=F(Z-8): NEXT Z
    
```





```

1520 LET B(8)=0: LET B(7)=0
1530 FOR Z=6 TO 1 STEP -1: LET B(Z)=F(Z+4): NEXT Z
1540 RETURN
1600 REM *** Convert binary [B(12)...B(1)] to BIN$
1610 LET BIN$="": FOR Z=12 TO 1 STEP -1
1620 LET BIT$=STR$(B(Z)): LET BIN$=BIN$+BIT$(2,1)
1630 NEXT Z
1640 LET BIN$="1$$$"+BIN$(1,4)+" 0X"+BIN$(7,6)
1650 RETURN
1700 REM *** New page
1710 FOR Z=1 TO 12: IF Z=3 THEN FOR W=1 TO 36: LPRINT " ";: NEXT W: LPRINT "page 1" ELSE LPRINT ""
1715 NEXT Z
1720 GOSUB 1400
1730 LET PAGE=2: RETURN
1800 REM *** Final notes
1810 FOR Z=1 TO 6: LPRINT "": NEXT Z
1820 LPRINT "NOTES";CHR$(13);"_____": LPRINT ""
1830 LPRINT "1 '$$$' in the POKE (binary) column and '&' in the (hex) column are as follows -"
1840 LPRINT "": LPRINT " Tone Generator #1      '$$$' = '000'      '&' = 'B'"
1850 LPRINT " Tone Generator #2      '$$$' = '010'      '&' = 'A'"
1860 LPRINT " Tone Generator #3      '$$$' = '100'      '&' = 'C'"
1870 LPRINT "": LPRINT "2 Sound generator clock frequency is 4.0MHz for MTX and BBC computers. This can"
1880 LPRINT " be changed in line 180 for other computers."
1890 LPRINT "": LPRINT "3 'X' in POKE (binary) column is don't-care. The figure in (hex) column assumes": LPRINT " that this is 0."
1910 LPRINT CHR$(27);"F"
2000 FOR Z=1 TO 12: LPRINT "": NEXT Z
2010 FOR Z=1 TO 36: LPRINT " ";: NEXT Z: LPRINT "page 2"
2020 RETURN

```



Sample Printout

SOUND TABLES FOR THE TI SN76489A SOUND GENERATOR I.C.

Bve	note	frequency, Hz	MTX		POKE (binary)		POKE (hex.)	
			direct	SBUF	byte 1	byte 2	byte 1	byte 2
-1	C	130.8	956	7648	1\$\$\$1100	0X111011	%C	3B
-1	C#	138.6	902	7216	1\$\$\$0110	0X111000	%6	3B
-1	D	146.8	851	6808	1\$\$\$0011	0X110101	%3	35
-1	D#	155.6	804	6432	1\$\$\$0100	0X110010	%4	32
-1	E	164.8	758	6064	1\$\$\$0110	0X101111	%6	2F
-1	F	174.6	716	5728	1\$\$\$1100	0X101100	%C	2C
-1	F#	185.0	676	5408	1\$\$\$0100	0X101010	%4	2A
-1	G	196.0	638	5104	1\$\$\$1110	0X100111	%E	27
-1	G#	207.7	602	4816	1\$\$\$1010	0X100101	%A	25
-1	A	220.0	568	4544	1\$\$\$1000	0X100011	%8	23
-1	A#	233.1	536	4288	1\$\$\$1000	0X100001	%8	21
-1	B	246.9	506	4048	1\$\$\$1010	0X011111	%A	1F
0	C	261.6	478	3824	1\$\$\$1110	0X011101	%E	1D
0	C#	277.2	451	3608	1\$\$\$0011	0X011100	%3	1C
0	D	293.7	426	3408	1\$\$\$1010	0X011010	%A	1A
0	D#	311.1	402	3216	1\$\$\$0010	0X011001	%2	19
0	E	329.6	379	3032	1\$\$\$1011	0X010111	%B	17
0	F	349.2	358	2864	1\$\$\$0110	0X010110	%6	16
0	F#	370.0	338	2704	1\$\$\$0010	0X010101	%2	15
0	G	392.0	319	2552	1\$\$\$1111	0X010011	%F	13
0	G#	415.3	301	2408	1\$\$\$1101	0X010010	%D	12
0	A	440.0	284	2272	1\$\$\$1100	0X010001	%C	11
0	A#	466.2	268	2144	1\$\$\$1100	0X010000	%C	10
0	B	493.9	253	2024	1\$\$\$1101	0X001111	%D	0F
1	C	523.2	239	1912	1\$\$\$1111	0X001110	%F	0E
1	C#	554.4	225	1800	1\$\$\$0001	0X001110	%1	0E
1	D	587.3	213	1704	1\$\$\$0101	0X001101	%5	0D
1	D#	622.2	201	1608	1\$\$\$1001	0X001100	%9	0C
1	E	659.3	190	1520	1\$\$\$1110	0X001011	%E	0B
1	F	698.5	179	1432	1\$\$\$0011	0X001011	%3	0B
1	F#	740.0	169	1352	1\$\$\$1001	0X001010	%9	0A
1	G	784.0	159	1272	1\$\$\$1111	0X001001	%F	09
1	G#	830.6	150	1200	1\$\$\$0110	0X001001	%6	09
1	A	880.0	142	1136	1\$\$\$1110	0X001000	%E	08
1	A#	932.3	134	1072	1\$\$\$0110	0X001000	%E	08
1	B	987.8	127	1016	1\$\$\$1111	0X001000	%F	07

#### NOTES

1 '\$\$\$' in the POKE (binary) column and '&' in the (hex) column are as follows -

Tone Generator #1	'\$\$\$' = '000'	'&' = 'B'
Tone Generator #2	'\$\$\$' = '010'	'&' = 'A'
Tone Generator #3	'\$\$\$' = '100'	'&' = 'C'







## BERTIE BUG

Mark Adams



```

100 REM For mono TV's change the colour of sprite 4 in line 280 and the colour of character 148 in line 1280.
110 PLOD "P1": LET HSCORE=0: CTLSPR 2,4: CTLSPR 6,2: GOSUB 810
120 VS 4: COLOUR 2,3: COLOUR 4,1: CLS : PAPER 3
130 CSR 6,2: PRINT "Value of objects."
140 CSR 2,4: PRINT CHR$(147): " Poisonous mushroom -1 Life"
150 CSR 2,6: PRINT CHR$(148): " Baby bugs -10"
160 CSR 2,8: PRINT CHR$(149): " Edible mushroom +1"
170 CSR 2,10: PRINT CHR$(150): " Aphides +10"
180 CSR 2,12: PRINT CHR$(151): " Flowers +5"
190 CSR 2,14: PRINT CHR$(152): " Seeds +15"
200 CSR 2,16: PRINT CHR$(153): " Berries +20"
210 CSR 2,18: PRINT CHR$(154): " Pesticide pellets -30"
220 CSR 7,22: PRINT "PRESS SPACE BAR"
230 IF INKEY$="" THEN GOTO 230
240 VS 4: CLS
250 FOR A=0 TO 31: PAPER 1: CSR A,0: PRINT " ": CSR A,1: PRINT " ": CSR A,2: PRINT " ": CSR A,23: PRINT " ": NEXT : PAPER 13
260 CSR 3,0: PRINT " BERTIE BUG'S LIVES 3 " : CSR 3,1: PRINT " SCORE " : PAPER 3
270 LET BX=132: LET BY=94: LET S=0: LET SCORE=0: LET LIVES=3: LET P=1: LET PP=2: LET SP=54: LET T=0: LET AJ=0
280 SPRITE 1,10,-16,0,0,0,4: SPRITE 2,9,-16,0,0,0,1: SPRITE 3,1,BX,BY,0,0,1: SPRITE 4,2,BX,BY,0,0,9
290 LET CX=INT(RND*29)+1: LET CY=INT(RND*18)+4: IF LIVES=1 AND CX=INT(BX/8) AND CY=23-INT(BY/8) THEN GOTO 290
300 CSR CX,CY: PRINT CHR$(INT(RND*8)+147)
310 LET MX=INT(RND*29)+1: LET MY=INT(RND*18)+4: IF LIVES=1 AND MX=INT(BX/8) AND MY=23-INT(BY/8) THEN GOTO 310
320 GOSUB 760
330 CSR MX,MY: PRINT CHR$(147)
340 CSR 0,INT(RND*18)+4: PRINT " " : REM 32" "
350 CSR INT(BX/8),23-INT(BY/8): LET S=ASC(SPK$)
360 IF S>=147 THEN CSR INT(BX/8),23-INT(BY/8): PRINT " ": GOSUB 400
370 LET L1=INT(RND*10): IF L1=3 AND SCORE>=500 AND T>=150 THEN GOSUB 700: LET T=0
380 GOSUB 760: GOSUB 580: LET T=T+1: GOTO 290
390 REM
400 REM *** SCORE AND LIVES ADJUST ***
410 IF S=147 THEN LET LIVES=LIVES-1: SOUND 2,1000,15
420 IF S=148 THEN LET SCORE=SCORE-10
430 IF S=149 THEN LET SCORE=SCORE+1
440 IF S=150 THEN LET SCORE=SCORE+10
450 IF S=151 THEN LET SCORE=SCORE+5
460 IF S=152 THEN LET SCORE=SCORE+15
470 IF S=153 THEN LET SCORE=SCORE+20
480 IF S=154 THEN LET SCORE=SCORE-30
490 IF SCORE<-30 THEN LET LIVES=LIVES-1: LET SCORE=0: SOUND 2,1000,15
500 IF SCORE>500 AND SCORE<530 AND LIVES<4 THEN LET LIVES=LIVES+1
510 IF LIVES<=0 THEN GOTO 1440
520 IF S=148 THEN CSR INT(BX/8),23-INT(BY/8): PRINT " "
530 PAPER 13: CSR 24,0: PRINT LIVES: CSR 11,1: PRINT " " : CSR 11,1: PRINT SCORE: PAPER 3: SOUND 2,0,0: GOSUB 760: RETURN
540 GOTO 300
550 REM
560 REM *** BERTIE BUG CONTROL KEYS ***
570 GOSUB 760
580 LET K=ASC(INKEY$)
590 IF K=8 OR K=10 OR K=11 OR K=25 OR K=26 THEN SOUND 3,8,7
600 IF K=25 AND (BX<239) THEN LET BX=BX+8: LET P=7: LET PP=8
610 IF K=8 AND (BX>15) THEN LET BX=BX-8: LET P=3: LET PP=4
620 IF K=11 AND (BY<151) THEN LET BY=BY+8: LET P=5: LET PP=6
630 IF K=10 AND (BY>23) THEN LET BY=BY-8: LET P=1: LET PP=2
640 IF K=26 AND P=1 AND (BY>47) THEN LET BY=BY-32 ELSE IF K=26 AND P=3 AND (BX>39) THEN LET BX=BX-32
650 IF K=26 AND P=7 AND (BX<215) THEN LET BX=BX+32 ELSE IF K=26 AND P=5 AND (BY<127) THEN LET BY=BY+32
660 ADJSR 0,3,P: ADJSR 0,4,PP: ADJSR 2,3,BX: ADJSR 2,4,BX: ADJSR 3,3,BY: ADJSR 3,4,BY
670 SOUND 3,0,0: RETURN
680 REM
690 REM *** SWALLOW CONTROL ***
700 CTLSPR 0,0: CTLSPR 3,0: CTLSPR 5,0
710 SPRITE 1,10,-8,BY,SP,0,4: SPRITE 2,9,-8,BY,SP,0,1
720 LET SP=SP+2: IF SP>124 THEN LET SP=124
730 CTLSPR 0,1: CTLSPR 3,2: CTLSPR 5,2: GOSUB 760: RETURN
740 REM
750 REM *** SPRITE PEEK ***
760 IF PEEK(65118)=PEEK(65126) AND PEEK(65121)=PEEK(65129) THEN CTLSPR 0,0: SPRITE 1,10,BX,BY,0,0,4: SPRITE 2,9,BX,BY,0,0,1: GOTO 1440
770 RETURN
780 REM
790 REM *** GENPATS ***
800 REM *** BERTIE BUG PATTERNS ***
810 GENPAT 4,1,32,32,22,73,97,1,97,65
820 GENPAT 5,1,0,0,0,2,0,4,8,16
830 GENPAT 6,1,8,8,208,36,12,0,12,4
840 GENPAT 7,1,0,0,0,128,0,64,32,16

```





```

850 GENPAT 4,2,3,7,15,31,31,31,31,15
860 GENPAT 5,2,3,7,7,7,3,0,0,0
870 GENPAT 6,2,128,192,224,240,240,240,224
880 GENPAT 7,2,128,192,192,192,128,0,0,0
890 GENPAT 4,3,0,0,0,0,128,64,32,8
900 GENPAT 5,3,0,8,32,64,128,0,0,0
910 GENPAT 6,3,0,0,216,83,4,8,4,4
920 GENPAT 7,3,248,4,4,8,4,83,216,0
930 GENPAT 4,4,0,0,0,0,0,14,31
940 GENPAT 5,4,31,31,14,0,0,0,0,0
950 GENPAT 6,4,0,0,0,0,120,252,254,255
960 GENPAT 7,4,255,255,254,252,120,0,0,0
970 GENPAT 4,5,16,8,4,0,2,0,0,0
980 GENPAT 5,5,65,97,1,97,73,22,32,32
990 GENPAT 6,5,16,32,64,0,128,0,0,0
1000 GENPAT 7,5,4,12,0,12,36,208,8,8
1010 GENPAT 4,6,0,0,0,3,7,7,7,3
1020 GENPAT 5,6,15,31,31,31,15,7,3
1030 GENPAT 6,6,0,0,0,128,192,192,192,128
1040 GENPAT 7,6,224,240,240,240,224,192,128
1050 GENPAT 4,7,0,0,27,202,32,16,32,32
1060 GENPAT 5,7,31,32,32,16,32,202,27,0
1070 GENPAT 6,7,0,0,0,0,1,2,4,16
1080 GENPAT 7,7,0,16,4,2,1,0,0,0
1090 GENPAT 4,8,0,0,0,0,30,63,127,255
1100 GENPAT 5,8,255,255,127,63,30,0,0,0
1110 GENPAT 6,8,0,0,0,0,0,112,248
1120 GENPAT 7,8,248,248,112,0,0,0,0,0
1130 REM
1140 REM *** SWALLOW ***
1150 GENPAT 4,9,32,24,28,14,135,71,111,63
1160 GENPAT 5,9,111,71,135,14,28,24,32,0
1170 GENPAT 6,9,0,0,0,0,0,176,252
1180 GENPAT 7,9,176,0,0,0,0,0,0,0
1190 GENPAT 4,10,0,0,8,4,2,2,71,15
1200 GENPAT 5,10,71,2,2,4,8,0,0,0
1210 GENPAT 6,10,0,0,0,0,0,0,0,128
1220 GENPAT 7,10,0,0,0,0,0,0,0,0
1230 REM
1240 REM *** OBJECTS ***
1250 GENPAT 1,147,255,231,129,0,0,231,231,255

```

```

1260 GENPAT 2,147,16*3+1,16*3+1,16*3+1,16*3+1,16*3+1,16*3+1,16*3+1
1270 GENPAT 1,148,219,231,231,66,195,66,195,102
1280 GENPAT 2,148,16*3+6,16*3+6,16*3+6,16*3+6,16*3+6,16*3+6,16*3+6
1290 GENPAT 1,149,0,24,126,255,255,24,24,0
1300 GENPAT 2,149,16*15+3,16*15+3,16*15+3,16*15+3,16*15+3,16*15+3,16*15+3
1310 GENPAT 1,150,36,24,24,189,60,189,60,153
1320 GENPAT 2,150,16*12+3,16*12+3,16*12+3,16*12+3,16*12+3,16*12+3,16*12+3
1330 GENPAT 1,151,42,28,28,42,11,108,24,8
1340 GENPAT 2,151,16*13+3,16*13+3,16*13+3,16*13+3,16*12+3,16*12+3,16*12+3
1350 GENPAT 1,152,0,0,24,60,243,249,60,24
1360 GENPAT 2,152,16*3+3,16*3+3,16*10+3,16*11+3,16*10+15,16*10+14,16*11+3,16*10+3
1370 GENPAT 1,153,0,28,62,127,127,127,62,28
1380 GENPAT 2,153,16*3+3,16*14+3,16*7+3,16*7+3,16*7+3,16*7+1,16*7+3,16*7+3
1390 GENPAT 1,154,60,126,170,255,255,170,126,60
1400 GENPAT 2,154,16*8+3,16*9+3,16*9+11,16*10+3,16*6+3,16*9+11,16*9+3,16*8+3
1410 RETURN
1420 REM
1430 REM *** END OF GAME ***
1440 FOR A=2 TO 3: SOUND A,0,0: NEXT A: ADJSR 4,2,0: ADJSR 4,1,0
1450 PAPER 13: CSR 24,0: PRINT LIVES: CSR 11,1: PRINT " " : CSR 11,1: PRINT SCORE
1460 PAPER 8: CSR 10,2: PRINT " GAME OVER "
1470 CSR 7,23: PRINT " PRESS SPACE BAR ": PAPER 3
1480 IF INKEY<>" " THEN GOTO 1480
1490 VS 4: CLS : ADJSR 2,3,239: ADJSR 3,3,16: ADJSR 2,4,239: ADJSR 3,4,16:
      ADJSR 0,3,3: ADJSR 0,4,4
1500 ADJSR 2,1,16: ADJSR 3,1,16: ADJSR 2,2,16: ADJSR 3,2,16
1510 IF SCORE>HSCORE THEN LET HSCORE=SCORE
1520 CSR 10,1: PRINT "GAME OVER."
1530 CSR 8,3: PRINT "Your score. ":SCORE
1540 CSR 8,5: PRINT "Hi-Score. ":HSCORE
1550 CSR 6,8: PRINT "Another Game? (Y/N)"
1560 CSR 7,11: PRINT CHR$(147):" -1 Life " :CHR$(148):" -10"
1570 CSR 7,13: PRINT CHR$(149):" +1 " :CHR$(150):" +10"
1580 CSR 7,15: PRINT CHR$(151):" +5 " :CHR$(152):" +15"
1590 CSR 7,17: PRINT CHR$(153):" +20 " :CHR$(154):" -30"
1600 CSR 7,21: PRINT "Author Mark Adams.": CSR 8,22: PRINT "Copyright 1985."
1610 IF INKEY$="Y" OR INKEY$="y" THEN GOTO 240
1620 IF INKEY$="N" OR INKEY$="n" THEN VS 5: CLS : STOP
1630 GOTO 1610

```

## Kiddies Comp... Spot The Difference

THE TWO PICTURES BELOW ARE DIFFERENT. STUDY THEM CAREFULLY THEN WRITE DOWN HOW MANY DIFFERENCES YOU CAN FIND. THE FIRST TWO CORRECT ENTRIES OPENED AFTER SEPTEMBER 30TH WILL EACH WIN A PRIZE. NO. I'M NOT TELLING YOU WHAT THEY ARE .... LET IT BE A SURPRISE !

ONLY OPEN TO CHILDREN UNDER 10 YEARS OF AGE.







View

w

# Letters

TO THE EDITOR

point



Mr. R.A.Siddall of 23 Lungbree Ave., Old Whittington, Chesterfield, Derbyshire. S41 9HW., would like to get in touch with any other Memotech owners in this country. He would also like to swap Snowball or Lords of Time for any other Level 9 adventure.

Mr. George Wells of 1 Copthall Way, New Haw, Surrey.KT15 3TX., has been stirred to put pen to paper by the editorial in issue 9.

Dear Viewpoint, No - I am sorry, I have so far resisted the temptation, but now I find myself compelled to put pen to paper following your cry from the heart for thoughts on the matter of support for the Memotech.

Well...I am sure that the vast majority of MTX owners would agree that the machine itself has a pretty good specification - excellent keyboard, terrific graphic capabilities, the Noddy language, expansion potential and so on. Indeed, I bought a 512 soon after they were introduced, on the assumption that such a technically superior machine would undoubtedly quickly capture a large slice of the market and much software support would soon be forthcoming. What happened? Soon after the MTX appeared, Amstrad hit the market with a micro that wasn't necessarily any better than the Memotech, but took off simply because Mr.Sugar got his act together by offering an all-in package - Data Recorder, Monitor and all.

The impression I got from the sidelines was that instead of accepting the challenge, Memotech seemed to regress into a slough of despondency, a typical attitude being "O.K., if you're not going to review our machine favourably, we'll withdraw our advertising from your magazine." That seems a terrific attitude to take in order to get more people to buy Memotechs.

One surely cannot blame existing MTX users for poor software sales - the bottom line is surely the truism that software support will only be evident the more machines are sold. Unto this end Memotech must really get their marketing act together. I note from my daily paper that a certain chain store with "280 Branches Nationwide" are offering an 80K Micro together with a Disc Drive, colour monitor and software package for 500.00, with the added incentive of interest-free credit. Surely the key to the Memotech taking off is a similar all-in deal backed by support from one or more chain stores. I am sure your average punter hasn't even heard of Memotech, let alone been given the chance to try one at his or her local shop.

Alright, the Memotech's got its failings - do they still supply the pathetic operating manual I received with my machine? But at the end of the day I am sure it is still not too late for the MTX to gain its rightful sales position alongside the BBC's, the Amstrad's, the Commodore's and the Spectrum's.

I do not really care whether the company sells a million computers to Russia - it's not really going to help us U.K.users much, is it?

I am not trying to be bitchy about Memotech for the sake of it - on the contrary, I hope this diatribe will be taken as constructive criticism which, if acted upon, would benefit us all ultimately. But it must be understood that, despite the valiant efforts of Keith Hook's Genpat and Memopad which really are Memotech's saving grace, it is incredibly frustrating for one who has invested a lot of money in the machine to see it apparently slipping into obscurity and the list of also-rans.



Please, Mr. Memotech, smile on us all, and do something to convince us that we made the right decision when we bought an MTX.

Mr. Dulay, of Birmingham would like to know if anyone has an interfacing circuit for an EPROM programmer that can be used with the MTX500 512.

Jim Crellin has sent in the following:

#### THE COMPUTER

I still think it's the best.

#### THE MAGAZINE

You have all done a grand job and have probably kept the computer going in this country.

#### THE SOFTWARE

Memosketch - excellent

Target zone - very popular

Escape from Zarcos - very good

Son of Pete - very good

#### DISAPPOINTMENTS

A feeling that unless Memotech move quickly they will lose out to the new Commodore's and Atari's.

The New Manual - paid for last November and still not received (realise this is not Genpat's fault).

The Spectrum Games Converter? Tri-com?

Mr. R. Thomas of Wakefield has sent in the following in the hope that it will help two members who wrote in with problems with their Pascal ROM in a previous edition.

A satisfactory solution to Mr. Macarthy's problem of sound channel 0 being triggered after loading files is to type SOUND 0,1,0 before entering Pascal.

In reply to Mr. Manning's problems -

- a) Many procedures can be compiled and linked together though it may be necessary to use the FORWARD command.
- b) Local variables are allowed in procedures.
- c) Four Bytes are used to represent real numbers.
- d) Variables including ERASE can be saved to tape.
- e) Source (Text) files can be merged. N.B. These are not numbered.
- f) There are no Disc commands in high soft Pascal though it would be possible to write your own procedures, if necessary, in machine code. The following addresses may help:

#8000-Text file starts here

#533B - End of text file

#5339 - End of text file - 2

Alternatively, it may be possible to use the Disc Basic commands as Pascal object code is stored as a line of assembler in a Basic program.

I also have a query of my own. I cannot get the DOLLAR SIGN F FILE NAME command to work on my Pascal ROM, even using the hackers' tip you published in issue 4. Is this a problem with all Pascal ROMs, and if not what can I do about it?

ED - Return your ROM to HI-SOFT they will upgrade it with the new ROMs as this problem was found on all early ROM versions.

Here is a letter from Mr.S.Wright

Dear Keith,

I am presently attempting to complete a machine code program which I have been writing for many months now and have come across a couple of problems. The program is just over 32 kilo bytes in length and I have recently bought a 32K Expansion for my MTX500 to continue its development. Unfortunately, I am now unable to expand the program further as the assembler keeps giving me NO SPACE error code even though I have plenty of memory left.

The 32 expansion appears to work efficiently as I have checked it as best I can by using the Basic command DIM A(60000) and PRINT PEEK (64122).

I would welcome your opinion as to whether the 32K expansion is in fact faulty or if there is a bug in the assembler limiting its memory capacity. It appears that only machine code is limited in memory as I can add Basic to the end of the program with no problems arising.





ED - Mr. Wright is correct in his assumption that there is a problem with the assembler when it reaches the end of 32K. When the extra memory, on the 512, is paged in it too, is located at #8000. If code was allowed to continue on to the extra page the assembler would get its "knickers in a twist", as there would be two instructions for each memory location above #8000, and unfortunately nothing can be done about this problem.

Bjorn Conway would like to contact any members in the Basingstoke area, or any members wishing to keep postal correspondence for the exchange of ideas on software, listings etc. His address is: 9 Kempshott Gardens, Basingstoke, Hants, RG22 5QU. ★

## MEMOTECH



# CPM Software



### WORD PROCESSORS & PROCESSING AIDS

	AUTHOR	R.R.P. £
Addressbook	Decision Tech	90
Easywriter I	Sorcim/US	145
Easywriter I to II Upgrade	Sorcim/US	125
Easywriter II	Sorcim/US	295
Easywriter II-Line Drawing	Sorcim/US	60
Easywriter II-Printer Selection	Sorcim/US	60
Easywriter II-Spelling Corrector	Sorcim/US	60
Elec. Thesaurus	Dictronics	105
Mailmerge	Micropro	145
Peachpack	Peachtree	350
Perfect Dealer Demo	Perfect	15
Perfect Writer II	Perfect	149
Quill	Psion	175
Spellstar	Micropro	145
Starindex	Micropro	116
Superwriter	Sorcim/US	295
Voltswriter DL	Lifetree	295
Word & Mouse	Microsoft	495
Wordstar	Micropro	295
Wordstar Professional Others	Micropro	399
Wordstar Prof. Pack add on	Micropro	245
Wordstar 2000	Micropro	465

### FINANCIAL MODELLERS & SPREADSHEETS

Abacus	Psion	175
Calstar	Micropro	116
Calstar Update	Micropro	75
Easyplanner	Sorcim/US	170
Fastplan II	Comshare	525
Masterplanner	Comshare	245
Multipan CPM 80	Microsoft	159
Supercalc To Supercalc 2 Upgrade	Sorcim/US	95
Supercalc To Supercalc 3 Upgrade	Sorcim/US	195
Supercalc 2 Supercalc 3 Upgrade	Sorcim/US	100
Supercalc 3r1 Supercalc 3r2 Upgrade	Sorcim/US	60
Trend	Planalysis	495

### DATABASE MANAGEMENT SYSTEMS

Access Manager 8 Bit	Digital Research	250
Archive	Psion	250
Cardbox	Caxton	195
Cardbox plus	Business Simulations	300
Dastar	Micropro	175
Dastar Training Guide	Micropro	15
Dastar Update	Micropro	75
Dbase II V2.43	Ashton Tate	395
DMS+	Comsoft	195
Easyfiler	Sorcim/US	270
Easyplus	Sorcim/US	75
Everyman	Smallway	475
Friday	Ashton Tate	195
Infostar	Micropro	295

### CODE GENERATORS CODING

### UTILITIES & MENU SYSTEMS

CP+	Taurus	200
Dgraph	Fox & Geller	200
Expressbase II	Salamanca	125
Quickcode	Fox & Geller	200
Sourcewriter	Microfocus	785
Starburst	Micropro	125
Zip for Cbasic & Mbasic	Nexus	115

### ACCOUNTING SOFTWARE

DBFLEX (Needs Dbase Run Time)	500
Sales Ledger	500
Purchase Ledger	500
Nominal Ledger & Accounts	500
Stock/Order Entry	1300
Any 3 Modules (Includes Nominal)	1500
The Above 4 Modules	500
Payroll	85
Dbase Run Time	300
MAP	300
Mapspec SSP inc 1yr maintenance	300
Sales Order Entry	300
Stock Control	300
Purchase Order Entry	300
Job Costing	300
Sales Ledger with invoicing	300
Purchase Ledger and Cheque Writing	300
Nominal Ledger	300

### COMMUNICATIONS SOFTWARE

Access 10 (IBM PC - PC DOS)	Digital Research	425
BSTAM	Byron	165
BSTMS	Byron	165
Crosstalk	Microstuff	195
Move-It	Woolf	130

### PROGRAMING LANGUAGES

Animator	Micro Focus	250
Basic Interpreter	Microsoft	340
Personal Basic	Digital Research	125
Cbasic Compiler CPM 80	Digital Research	425
Basic Compiler	Microsoft	385
Cobol CPM 80	Microsoft	680
Business Basic Compiler	Microsoft	450
DR C Language	Digital Research	295
DR Logo	Digital Research	125
Forms II	Micro Focus	110
Fortran CPM 80	Microsoft	485
Ms Pascal	Microsoft	295
PL/I CPM 80	Digital Research	470
RM Cobol CPM 80 full dev. system	Ryan McFarland	699
SPP CPM 80	Digital Research	175

### PROGRAM DEVELOPMENT AIDS

Display Manager 8 Bit	Digital Research	330
DR Assembler + Tools 8 Bit	Digital Research	170
Edit 80	Microsoft	115
M2Cbasic	Digital Research	150
Macro 80	Microsoft	195

### BUSINESS GRAPHICS

Chart	Microsoft	245
Chartmaster	Decision Resources	425
Chartmaster & Signmaster	Decision Resources	660
Chartstar	Micropro	325
Dataplot+	Grafox	195
Delta-Graph	Comsoft	195
DR Draw CPM 80	Digital Research	250
DR Graph CPM 80	Digital Research	250
Easel	Psion	175
Foxgraph	Fox & Geller	200
Signmaster	Decision Resources	283

The above CP/m software is available to members at 20% discount. Please allow 14 days for delivery and state if you require the programs to be installed for you. All programs come complete with licences and manufacturers instruction books.

For more than one item please ring for special discounts.

Prices are correct at time of going to press but please ring to confirm before ordering as a lot of the prices are subject to a fluctuation on the pound/dollar.



## SOFTWARE



00106	26X26 SPREAD SHEET	UTIL	SYNT	7.95	E	ANY
00057	3D TACHYON FIGHTER	ARC	CONT	6.95	I	ANY
00062	ADVENTURE QUEST	ADV	LVL9	8.75	I	ANY
00033	AGROVATOR	ARC	SYNT	5.95	I	512
00071	ALICE IN WONDER.	ADV	CONT	6.02	I	ANY
00008	ASTROMILON	ARC	CONT	6.02	I	ANY
00047	ASTROPAC	ARC	CONT	6.02	I	ANY
00058	BACKGAMMON	BRD	CONT	7.95	I	ANY
00041	BASIC BUSINESS	BS	CONT	5.95	I	ANY
00043	BLOBBO	ARC	CONT	6.02	I	ANY
00073	BOUNCING BILL	ARC	SYNT	4.95	I	ANY
00074	BRIDGE	CARD	CONT	6.95	I	512
00077	CANVAS	UTIL	CONT	7.95	I	ANY
00085	CAVES OF ORB	ADV	SYNT	5.95	I	512
00094	CHAMBEROIDS	ARC	MEGA	5.95	I	ANY
00059	CHESS	BRD	CONT	8.75	I	ANY
00053	COBRA	ARC	CONT	6.02	I	ANY
00025	COLOSSAL ADVENTURE	ADV	LVL9	8.75	I	ANY
00098	COMBAT	ARC	PANS	2.95	I	512
00028	COMPOSER	UTIL	XAV	13.00	I	ANY
00046	CONT RAIDERS	ARC	CONT	6.02	I	ANY
00099	CRIBBAGE	CARD	SCRIP	2.95	E	ANY
00050	DEN.GOES BANANAS	ARC	SCRIP	2.95	I	ANY
00011	DENNIS & CHICKEN	ARC	SCRIP	2.95	I	ANY
00103	DENNIS AND CIRCUS	ARC	SCRIP	2.95	I	ANY
00068	DOODLEBUG	ARC	SYNT	4.95	I	ANY
00096	DR. FRANKIE	ARC	SYNT	5.95	I	512
00056	DRAUGHTS	BRD	CONT	6.95	I	ANY
00063	DUNGEON ADVENTURE	ADV	LVL9	8.75	I	ANY
00067	EDASM	UTIL	SYNT	7.95	I	512
00066	EMERALD ISLE	ADV	LVL9	5.95	I	ANY
00038	ESCAPE FROM ZARKOS	ARC	MEGA	5.95	I	ANY
00081	EXTENDED BASIC	6.95	SENT	6.95	I	ANY
00082	FATHOMS DEEP	ARC	MEGA	5.95	I	ANY
00090	FIG FORTH	LANG	SYNT	15.75	I	512
00055	FIREHOUSE FREDDIE	ARC	CONT	6.02	I	ANY
00021	FIRST LETTERS 1	EDUC	CONT	8.75	I	ANY
00092	FKEY DEFINER	UTIL	MEMB	6.95	I	ANY
00037	FLUMMOX	ARC	SYNT	5.95	I	512
00052	GAUNTLET	ARC	CONT	6.02	U	ANY
00102	GHOSTLY CASTLE	ADV	PANS	2.95	I	ANY
00031	GOLDMINE	ARC	CONT	6.02	I	ANY
00069	GRAPHICS	UTIL	CONT	5.95	I	ANY
00087	H & L DUMP	UTIL	MEM	4.95	I	ANY
00072	HAWKWARS	ARC	SYNT	4.95	I	ANY
00065	HELI-MATHS	EDUC	CONT	5.95	I	ANY
00034	HUNCHY	ARC	SYNT	4.95	I	ANY
00083	ICEBERG	ARC	SYNT	4.95	I	ANY
00105	JET SET WILLY	ARC	SPRJ	6.95	E	ANY
00015	JOHNNY REB	WAR	LOTH	6.02	I	ANY
00097	JUMPING JACK FLASH	ARC	SYNT	5.95	I	512
00016	KEY TO TIME	ADV	LUMP	6.02	I	ANY
00042	KILOPEDE	ARC	CONT	6.02	I	ANY
00019	KNUCKLES	ARC	CONT	7.95	I	ANY
00078	LES FLICS	ARC	PSS	6.95	E	ANY
00032	LITTLE DEVILS	ARC	SYNT	4.95	I	ANY
00024	LORDS OF TIME	ADV	LVL9	8.75	I	ANY
00014	M CODER	UTIL	PSS	6.02	U	ANY
00035	M COMMAND & ARCAD.	ARC	SYNT	4.95	I	ANY
00070	MAN FROM GRANNY	ADV	SYNT	4.95	I	512
00104	MANIC MINER	ARC	SPRJ	6.95	E	ANY
00022	MATHS 1	EDUC	CONT	8.75	I	ANY
00013	MAXIMA	ARC	CONT	6.02	E	ANY
00086	MEMOCHEQUE	UTIL	SYNT	6.95	I	ANY
00075	MEMOSKETCH	UTIL	SYNT	7.95	I	ANY
00089	MINER DICK	ARC	XAV	6.95	I	ANY
00044	MISSION ALPHATRON	ARC	CONT	6.02	I	ANY
00030	MISSION OMEGA	ARC	SYNT	4.95	I	ANY
00054	MURDER AT MANOR	ADV	LUMP	6.02	I	ANY
00010	MUSIC PAD	UTIL	CONT	6.02	I	ANY
00003	NEMO	ARC	CONT	6.00	I	ANY
00045	OBLOIDS	ARC	CONT	6.02	I	ANY
00001	PAYROLL	UTIL	CONT	21.25	I	512
00005	PHAID	ARC	CONT	6.02	I	ANY
00061	PHYSICS 1	EDUC	CONT	8.75	I	ANY
00012	PONT & BLACKJACK	CARD	CONT	6.02	I	ANY
00009	POT HOLE PETE	ARC	CONT	6.02	I	ANY
00040	PURCHASE LEDGER	BN	CONT	12.75	I	512
00048	QOGO	ARC	CONT	6.02	I	ANY
00076	QOGO 2	ARC	MEGA	5.95	I	ANY
00095	QUANTUM	ARC	SYNT	5.95	I	ANY
00107	RED MOON	ADV	LVL9	?	U	ANY
00064	RETURN TO EDEN	ADV	LVL9	8.75	I	ANY
00020	REVERSI	BRD	CONT	7.95	I	ANY
00100	RUTHLESS BASTARD	ARC	LSFT	2.50	I	512
00002	SALES LEDGER	UTIL	SYNT	15.75	E	512
00029	SALTY SAM	ARC	SYNT	4.95	I	ANY
00101	SLOOPY'S CHRISTMAS	ARC	PANS	2.95	I	ANY
00049	SNAPPO	ARC	CONT	6.02	I	ANY
00023	SNOWBALL	ADV	LVL9	8.75	I	ANY
00036	SON OF PETE	ARC	MEGA	5.95	I	ANY
00026	SPELLI-COPTER	EDV	CONT	5.95	I	ANY
00080	SPOOLER	UTIL	MEM	4.95	I	ANY
00017	STAR COMMAND	ARC	CONT	6.95	I	ANY
00084	SUPER BIKE	ARC	SYNT	4.95	I	ANY
00004	SUPER MINEFIELD	ARC	CONT	6.02	I	ANY
00093	SURFACE SCANNER	ARC	MEGA	5.95	I	ANY
00039	TAPE TO DISC	UTIL	MEM	6.95	I	ANY
00007	TAPEWORM	ARC	CONT	6.02	I	ANY
00088	TARGET ZONE	ARC	SYNT	6.95	I	ANY
00051	THE ZOO GAME	ADV	CONT	6.02	I	512
00006	TOADO	ARC	CONT	6.02	I	ANY
00018	TURBO	ARC	CONT	6.95	I	ANY
00079	USER EXTEND	UTIL	MEM	7.95	I	ANY
00027	UTILITIES 1	UTIL	CONT	4.95	I	ANY
00091	VERNON & VAMPIRES	ARC	SYNT	5.95	I	ANY
00060	WORD & PICTURE	EDUC	CONT	8.75	I	ANY

I = IN STOCK U = UNAVAILABLE

E = EXPECTED SOON

KEY: STOCK NUMBER : TITLE : TYPE : PUBLISHER : PRICE : WHICH MACHINE \*\*\*ONLY ORDER THOSE MARKED I





## RAFFLE WINNER

THE DISC DRIVE WINNER IS MR. A.P. PARSONS  
MEMBERSHIP NUMBER D1618 FROM SALTASH IN CORNWALL.

If Mr. Parsons will contact us we will be pleased  
to send him his prize.

The Raffle turned out to be a success after all and  
we nearly broke even. We shall run the raffle again  
next month. ★

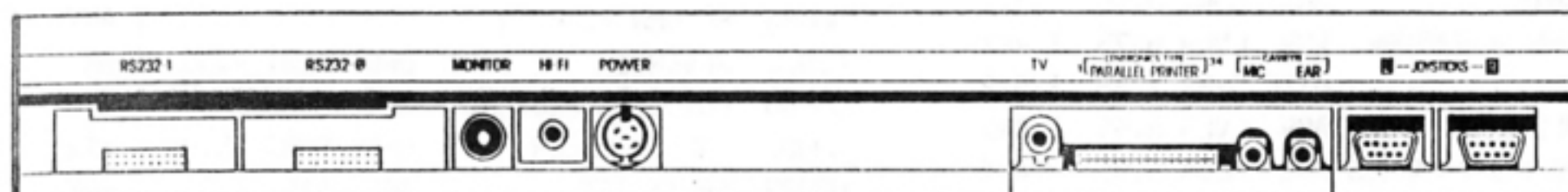


## SUBSCRIPTIONS

IF YOUR MEMBERSHIP NUMBER IS BETWEEN A500 & B655 INCLUSIVE  
YOU ARE NOW DUE TO RENEW YOUR MEMBERSHIP SUBSCRIPTION. TO  
MAKE SURE OF RECEIVING SEPTEMBER'S EDITION OF "MEMOPAD" PLEASE  
SEND YOUR FEES BEFORE THE 30th September.

We are holding the subscription rate at £16.00 due to the fact  
that we have negotiated a special postal rate with the G.P.O.

A renewal application is enclosed on the software order sheet.



Well, that's the end of this edition. We must apologise that we haven't included DISC MANIA but space has beaten us once more. Also, we are ashamed to admit, we have lost the master disc which has the CONVERTING CONNECT FOUR articles. We are hurriedly trying to re-write them, so don't worry ... we will finish the series.

Since we moved into the new premises we have had lots of visitors and we have really enjoyed meeting the members in person. I am sorry that I haven't been available recently but my work load is such that I need an extra 16 hours a day but once I have my two new books out of the way I hope to be back to normal.

We will be closed from SATURDAY 7TH SEPTEMBER UNTIL MONDAY 16TH SEPTEMBER. During this time all orders will be despatched by our new post girl Pat who is coming in to make sure that you are not kept waiting.

Hopefully, we shall be refreshed after our holiday and will be back with you in September. Until then, keep on tapping....