

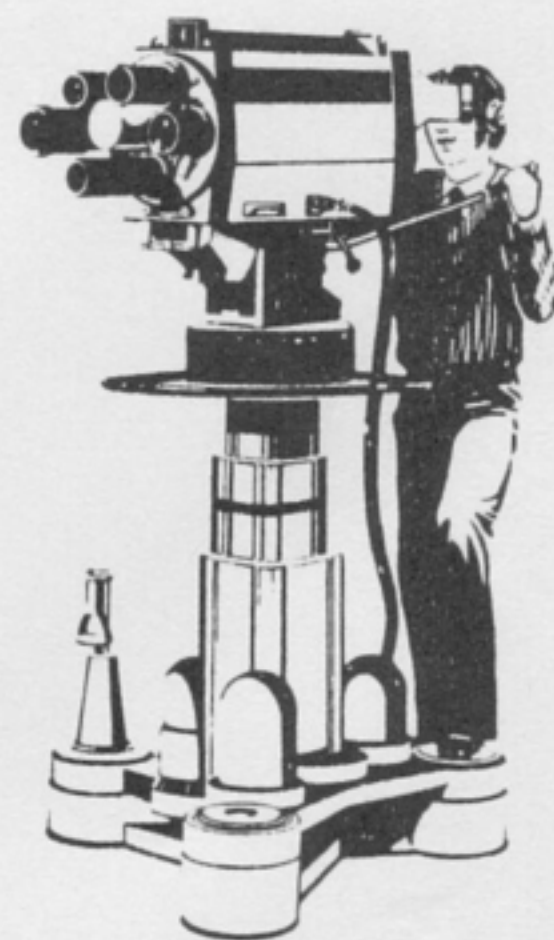
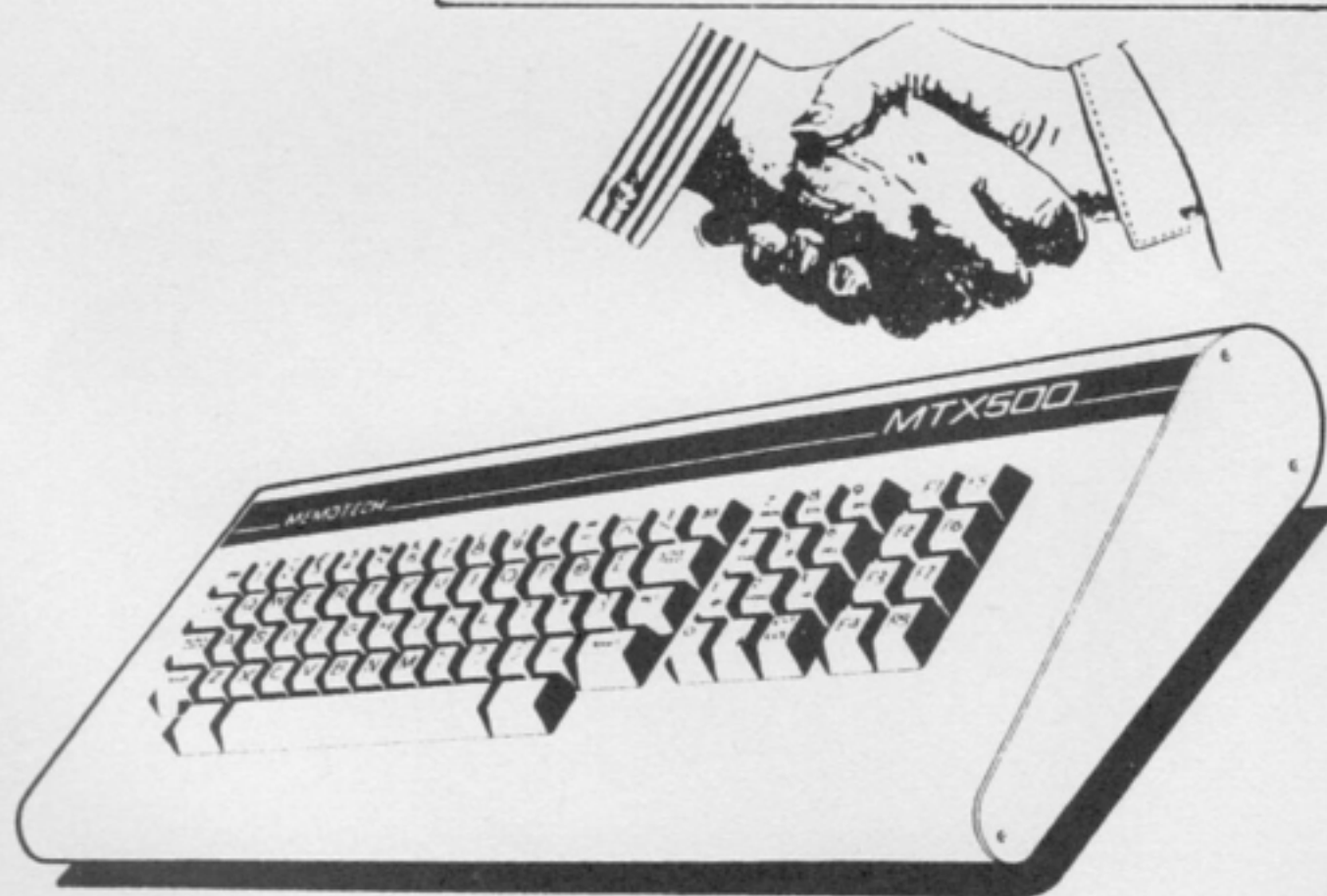
VOL 0010
NUMBER 1

memopad

Memotech Computer User Club Magazine

Contents

EDITORIAL	PAGE 2
HIGH SCORES & HIT LIST	PAGE 3
PREVIEWS & REVIEWS	PAGE 5
KEYBOARD KAPERS TUTORIAL	PAGE 9
FORTH ANOTHER VIEW	PAGE 11
BASIC GRAPHICS 5	PAGE 12
SPEEDY 3D PART 2	PAGE 14
TAKE A NOTE SOUND TUTORIAL	PAGE 15
AN EASY INTERFACE PART 2	PAGE 17
ASSEMBLY LINE	PAGE 19
RST28'S THE FINAL CONFLICT	PAGE 21
VIEWPOINT	PAGE 24
STARTING FORTH DIGEST YOUR WORDS.....	PAGE 26
SKETCH PROGRAM	PAGE 29
HARDWARE	PAGE 30
SOFTWARE	PAGE 31
SPECTRUM EMULATOR FINAL NEWS	PAGE 32



£1.25



TEVIC-DATA
DENMARK

NEW PRODUCT

FOR THE
Memotech
MTX SERIES



MEMOTALK

Speech synthesiser. You may use it when programming, e.g. place speech comments on errors.

A demotape is included.

£33.95

Prices
include
VAT

AVAILABLE DIRECT FROM GENPAT POSTAGE PAID.
REQUIRES EXTERNAL AMPLIFIER OR HI-FI. PLUGS INTO THE LEFT HAND SIDE OF THE COMPUTER. COMPLETE WITH DEMONSTRATION TAPE AND INSTRUCTIONS.



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



Sitting here, on the beach, I am trying to find some logic as to why the home computer market is bent on its own self-destruction.

Amstrad have really put the cat among the pigeons, so to speak, with their launch of the totally unexpected CPC6128 computer. They have also managed to make enemies out of a great many of their users (people who bought the 664) by launching this new computer. It is accepted, within the trade, that once development costs have been recouped, prices of the computer drop but the 664 is only a few months old and many purchasers were lulled into a sense of false security by this fact and went ahead with their purchase only to be told, a couple of months later, that their purchase is out of date, deceased, defunct, dead, and will not be supported. A really nice gesture by the Company. Having said all this, Amstrad have succeeded in baffling other manufacturers who are now panic-stricken and are trying to find ways of reducing their costs to the end user.

This is where the trouble starts. From the 1st October Memotech are selling the MTX 500 for £89, the 512 is to retail at £129 and most of the peripherals have been reduced by massive amounts. Following Memotech, Tatung have reduced the Einstein by another £100. Now, what happens to the new Commodore 128 and the recently launched Atari? The crafty prospective purchaser will decide to hang on until the inevitable price reduction. But what happens to the recouping of development costs? The initial rush of sales doesn't happen and the companies are left with the choice of reducing their prices to create more sales or trying to battle through their decision is obvious and thus the industry starts to act like a program that never exits from a faulty FOR NEXT LOOP !!

Nobody needs to consult OLD MOORE'S ALMANAC to forecast what is going to happen, regardless of 16-bit, 32-bit or mega-bit, unless manufacturers get their act together.

Which brings me very nicely to the Amateur Dramatic Society down at Witney. Memotech have now a very nice machine, one of the best, at a price that makes the decision between buying a piece of plastic, now doomed to be used as the emblems on the front of C5's, totally one sided. However, after talking to the producers it seems that the show will go on the road without the star getting its rightful billing !

This situation (tragedy) cannot be allowed to happen ! I must, therefore, ask for your help. We shall be advertising, in the most popular press, a package which will hopefully attract the Spectrum User, who wants to move up, and thus aim at a vast market. Next month we want to offer you an incentive for selling a Memotech to your friends. We shall offer you a £20 voucher for every MTX 512 you sell at £129.00. We all know what a good machine the Memotech is let's all have ago at being salesmen/women !!!!

GENPAT HIT LIST....

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

arcade

1	MEMOSKETCH	SYNTAXSOFT
2	ESCAPE FROM ZARCOS	MEGASTAR
3	FELIX IN THE FACTORY	MICROPOWER
4	DENNIS AND THE CHICKEN	PANISOFT
5	EDASM	SYNTAXSOFT
6	OGGO 2	MEGASTAR
7	DR. FRANKIE	SYNTAXSOFT
8	SON OF PETE	MEGASTAR
9	CHAMBERLAINS	MEGASTAR
10	FLUMMOX	SYNTAXSOFT

adventure

1	EMERALD ISLE	LEVEL 9
2	THE KEYS TO TIME	SENTIENT
3	MURDER AT THE MANDOR	SENTIENT
4	SNOWBALL	LEVEL 9
5	ADVENTURE QUEST	LEVEL 9

educational

1	FIRST WORDS	CONTINENTAL
2	SPELLICOPTER	SENTIENT
3	HELLI MATHS	SENTIENT

High Scores

Can you do better?

GILDTIME	12,250	Nic Joynson
ASTRO-PAC	105,990	Richard Nash
BOULING BILL	120,142	Alan Dobson
SNIPPO	120,688	Richard Franks
KNUCKLES	999,999+	Sally Street
CHAMBERLAINS	Completed 6 mins	M. Allcorn
MEMO	17,610	Richard Nash
COBRA	8,924	Richard Nash
MISSION ALPHATRON	68,250	T. Eriksson
TAPEDOWN	175,980	Richard Franks
10000	179,202	Gavin Gaunt and Nicholas Locke
POT HOLE PETE	106,630	Richard Franks
MAXIMA	500,000	Virginia Parton (5 ships left)
STAR COMMAND	140,430	Ian Nichols
PHATO	26,000	Sally Street
ORLOIDS	57,800	P. Crighton
KILOPEDE	82,253	Richard Nash
3D TACHYON FIGHTER	10,700	Lesla Woodger
CONTINENTAL RAIDERS	106,240	Sean Haverly
BLOODB	148,203	Elizabeth Nelson
QUANTUM	6	M. Allcorn
OGGO 2	205,000	R. Siddall
MINETELD	1,500	David Nash
FLUMMOX	166,690	Andrew Miller
THURRO	15,630	N. Crighton
FATHOM DEEP	1,790	Richard Franks
AGROVATOR	203,608	Richard Franks
FIREHOUSE FREDDIE	29,620	T. Eriksson
OGGO	43,960	T. Eriksson
ARCADIAN	25,900	Adrian Joynson
MISSILE COMMAND	27,590	Adrian Joynson
LITTLE DEVILS	31,250	L. Banks
FELIX IN THE FACTORY	11,950	Richard Nash
HURON	7,908	R. Harner
SON OF PETE	8,051	Gavin Gaunt
WARRIORS	16,600	T. Eriksson
ESCAPE FROM ZARCOS	36 Items	R. Siddall
SALLY SAM	40,642	Andrew Johnson
MISSION OMEGA	9,350	R. Harner
ICEBURG	17,431	Alan Dobson
SNOWBALL	450	P. Crighton
EMERALD ISLE	300/1000	Richard Franks
SUPERHINE	22.7ms	M. Allcorn
REVERSI	beaten by 47	Richard Franks
DOODLEBUG	3,440	M. Allcorn
DR. FRANKIE	13,680	Alan Banks
TARGET TIME	8,795	M. Allcorn
MINER DICK	22,520	R. Siddall
JUMPING JACK	12,000	Nic Joynson

Mike Nash has completed Ogo 2 and has quoted the final message - "At last, you have found the Ogan diamonds"

Disappointed with software quality ?!

Why settle for second best ?

MEGASTAR only produce very excellent & addictive software which the MTX so richly deserves.

This is because **MEGASTAR GAMES** are produced with more effort and dedication than any of our competition. The results are truly outstanding—with titles such as **Son of Pete** and **Sepulcri Scelerati**.

Other titles available are:

**Qogo - 2, Escape from Zarkos, Karate King,
Rolla Bearing, Obliteration Zone, Crystal,
Downstream Danger, S.M.G, Chamberoids,
Fathoms Deep, Surface Scanner, Drive Cee/5**

Why not send off for a

FREE software catalogue.

 **MEGASTAR GAMES**

Avenue 1, Unit 31, Station Lane Industrial Estate, Witney,
OXON OX8 6BX.

PREVIEWS

MEGASTAR

The MTX Comes Of Age!?

BRILLIANT ! Is the only way to describe the latest set of software due for release from MEGASTAR.

WARNING *** The following are short reviews of pre-release samples. They are a personal opinion from Tim Marstian but rather than wait to bring you the news we asked Tim to come into Memopad and play with them and write short resumes on the programs. THEY ARE NOT RELEASED YET! PLEASE DO NOT ORDER UNLESS YOU ARE PREPARED TO WAIT. Megastar expect to have the first batch to us within the next four weeks. Don't worry you will be the first to have them !

We have all been impressed with this batch of software, and I am personally over the moon with the sound on ROLLER BEARING. Andrew Key seems to have, at last, matured and with the release of SMG he is entitled to call himself a programmer.

MEGASTAR

Downstream Danger
reviewed by
T. Marstian

The aim of the game is to collect all of the jewels to make Bert change from a bee to a human again. He is trapped in a maze of underground caverns and tunnels underneath his home. The screens are extremely well drawn and show a great deal of imagination. One nice feature is that when your bee touches water, he transforms into a fish! You may then swim underwater to collect any submerged jewels etc. To change back to a bee, the fish must jump out of the water and touch a small bee. Unfortunately, this usually results in the fish landing on the ground and dying very rapidly.

The graphics and sound are first class. While the program is loading, an impressive screen is displayed informing us of the author, title of program etc. As soon as the program loads, the title screen appears. This scrolls the instructions across the screen whilst playing music. To begin the game, simply press the fire button.

Your bee is fairly difficult to manoeuvre at first, although after a while it becomes easier. At the bottom of the screen are the score, high score, lives remaining and the energy left. Almost everything is potentially lethal to the touch. The exceptions are cherries (they increase your energy) and water (changes you into a fish). You may rest at one or two locations. The most obvious being the tree from where you begin.

This game is destined to become a classic on the Memotech!

MEGASTAR

Obliteration Zone
reviewed by
T. Marstian

Although Obliteration Zone is quite a good game, it does not reach the standard set by Downstream Danger. When loaded, the title screen and instructions are displayed. The object of the game is to neutralize the power nodes without being obliterated, hence the name of the game. If you manage to complete this awesome task, you will advance to the next sheet. There are four different types of nasties, all of which perform different functions. For example, gamma tracers are harmless, but they drop bombs that give off lethal radiation.

The multicoloured graphics are too small for my liking. The "maze" effect is created by numerous coloured blocks arranged in order on the screen. Also, random blocks are generated causing the maze to be more difficult to negotiate. The control is quite easy using a joystick, although there isn't a facility to redefine keys for non-joystick owners.

Unfortunately, although quite a good game, Obliteration Zone isn't anything spectacular.

MEGASTAR

Rolla Bearing
reviewed by
T. Marstian

The aim of this game is to move your ball to a waving flag, for which you receive a bonus. There are numerous things to help and hinder you. e.g. springs, magnets, loony lifts, acid dust clouds, acid rain drops, hyperactive needles, barmy blow pipes etc...

The most amazing feature of the program is the music. It reproduces realistic cymbals, drums and other musical instruments and must surely surpass all other programs (perhaps even other computers) with the effects.

The first phase is fairly easy to play, but after this is completed, hazards appear causing later phases to become extremely difficult

One of features on the menu is a demonstration of the first phase which helps a newcomer to the game. It is a very slow demonstration making you wait before you play the game as there is no way of exiting from the demonstration.

Control of the game is by the joystick (arrow keys if no joystick). The ball's movement is fairly slow like when the ball is moving up a slope, but moves a slight bit faster when going down the slope.

Lot's of features make this an interesting game. The music adds greatly to the enjoyment.

MEGASTAR

Drive the Cee-5
reviewed by
T. Marstian

The object of this game is to run over hedgehogs in your Cee-5. You control the Cee-5 using the joystick or cursor keys. The screens, taking the form of a town comprised of trees, buildings, houses etc., are varied and numerous.

The "nasties" take the form of other road users recklessly travelling the roads on penny farthings etc. Movement from screen to screen is achieved by driving off the side of the current screen. The sound effects are reasonable but unspectacular. The graphics are quite large and varied. A nice feature of the game is an ambulance that drives onto the screen when you are killed.

My review copy doesn't have a loading screen although I am assured that the final version will have.

A nice game but I think it may become less interesting after prolonged usage.

MEGASTAR

Sepulcri Scelerati
reviewed by
T. Marstian

Unfortunately, my review copy didn't contain any instructions. Undeterred, I loaded the program into my computer. To my surprise, it is a three dimensional "maze" game in which you control a vehicle of some kind. The view is from above and to the side of the vehicle. The effect is similar to Alien 8 from Ultimate (on the Spectrum). The vehicle is capable of jumping onto various blocks and platforms in order to negotiate the obstacles. In various places there are holes in the floor which you may travel down. To move from screen to screen, you simply pass through an archway or window. Everything appears to be three dimensional, including the few "nasties" I discovered. They seemed to be travelling in defined patterns with the apparent intention of preventing your access into various locations.

The graphics are excellent throughout the game. Control is quite easy, thanks to the redefine key function, although negotiating certain obstacles is tricky. The sound is fairly good but it is perhaps limited by the amount of memory available due to the numerous graphics.

The game is addictive in a way because certain problems require numerous attempts to complete.

The best game yet from Chris Sawyer

MEGASTAR

SMG
reviewed by
T. Marstian

This is a massive game in which you must guide your large character around a scrolling maze. This is certainly an original game for the Memotech. The graphics are large and clear and the scrolling is relatively smooth. If you move your character towards the edge of the display, the whole screen scrolls across accordingly.

Objects include pick axes, knives and keys. Pick axes are used to tunnel through walls, but unfortunately they don't last for very long. There are bags of gold scattered around the maze which must be collected to finish the game. There are 90 bags in all!

My version only works on the MTX512, although a 500 version will be available soon. The computer plays many different tunes while playing the game. The volume of the music can be turned up and down to enable you to hear the sound effects.

Control is very easy both with the joystick and keys. There are various means of quickly travelling around the maze. The most obvious being long ropes to climb up or down. You may drop off ledges etc. without damage if you land on a mattress (even a distance covering more than one screen).

The game is addictive due to the size of the maze and the many interesting features.

This seems to be the best game from Andrew Key at the present.

MEGASTAR

Quazzia
reviewed by
T. Marstian

This is the latest version of Scramble on the Memotech. The graphics are first class making it the best yet. The aim of the game is to control your spaceship while dropping bombs on fuel dumps below and firing at the launching missiles. Your fuel goes down very quickly but if you hit a fuel dump with either a bomb or rapid fire your fuel goes up. There are different levels to be completed such as after the first level you enter a tunnel with different nasties to do battle with.

Control is by joystick or keyboard. With the keyboard you are able to choose your own keys to your advantage. I found the keyboard hard to control the ship but I would expect the joystick to be easier.

The title music is very good indeed and the sound effects are also very good.

This is the ultimate Scramble and is destined to become another bestseller for Megastar.

MEGASTAR

Crystal
reviewed by
T. Marstian

Crystal is a maze game with 8 levels and 2048 rooms. The aim of the game is to collect all 8 pieces of a magic crystal and take them to the upper left corner of the bottom level. To try and prevent you from completing your task there are ghosts and guards. Hidden in the maze are things that will aid you in finding the pieces of the crystal and protect you from the ghosts. These are things like shields, swords, food and wound remedies.

One complaint I have is that some of the monsters are virtually the same colour as the background are almost impossible to see.

The music is good but after playing the game for a long time it tends to annoy and there doesn't seem to be any way of turning the music off.

I enjoyed playing the game and would recommend it to anyone who likes playing maze games.

One or two points in the program reduce the playability of the game.

★★★★

USER BASIC

SYNTAXsoft

```

140 REM-----
150 REM Set up USER system variables &
160 REM ensure code saves to tape.
170 REM-----
180 LET ZZ=USR(59392)
190 USER POKE #FA92,#E7FF
200 REM-----
210 REM Dim. all arrays before the
220 REM USER BASIC command.
230 REM-----
240 DIM DISC$(8,17),DPOS(3),POST(3,8)
250 USER BASIC
260 USER CALL_INITVARS
270 USER CALL_INSTRUCTIONS
280 USER CALL_VSSCREEN
290 REM-----
300 REM Main program loop.
310 REM-----
320 USER @PLAYROUND
330 USER CALL_NUMDISCS
340 CLOCK "000000"
350 USER REPEAT
360 USER @FROMTO
370 USER CALL_FROMPOST
380 USER CALL_TOPOST

390 USER CALL_TESTMOVE
400 IF MOVE=FALSE THEN USER CALL_BADMOVE: US.
410 USER CALL_STOREMOVE
420 USER CALL_MOVEDISC
430 USER UNTIL_FINISHED=TRUE
440 USER CALL_PLAYAGAIN
450 IF POS>4 THEN USER JUMP_PLAYROUND
460 USER POKE #FA91,&10100000
470 USER VAR
480 STOP
490 REM-----
500 REM Post to move from.
510 REM-----
520 USER @FROMPOST
530 COLOUR 0,5: COLOUR 1,15
540 CSR 0,22: PRINT BLK$(1,20)
550 CSR 2,22: PRINT "FROM.."
560 USER CALL_INCTIME
570 LET FP=KEY
580 RETURN
590 REM-----
600 REM Post to move to.
610 REM-----
620 USER @TOPOST
    22: PRINT "

```

Notice anything unusual about the above program? It is written in USER_BASIC. This is the very latest utility to be produced for the MTX. There are three versions of this program MTX SDX500 & SDX512. The MTX version occupies no user space and the complete ram is available to the programmer. The disc versions, because of the way the interface is designed, are stored in ram as a Basic line.

There are over 20 extra commands available when you load this utility and what impressed me about the program was the fact that you don't have to have USER_BASIC to run a program written in the dialect as the core image run-time module is saved with the program.

After loading the program a simple USER BASIC will hand control over to the new utility. USER CALL <LINE NAME> is a very useful command and allows you to program in a similar manner to BBC Basic by having the facility to call the line by name rather than by line number.

USER REPEAT linked with USER UNTIL <expression = > & USER WHILE, USER ENDW fill in some of the blanks sadly missed in standard MTX Basic. They allow very powerful condition testing.

USER INSTR_SUB\$,MAIN\$ is one of my favourites. This allows you to find the first occurrence of one string in another: LET X\$= "TILE":USER INSTR_"L",X\$ WILL RETURN POS = 3. The variable POS is automatically set up by USER_BASIC and will be = to 0 if the substring does not appear in the main string.

USER SHAPE_X,Y,N,<string> is a graphic command that will allow a lot of experimentation and allows you to draw complex shapes by holding the information in a string. Ex Oric users will remember a command similar to this. Also PENUP, PENDOWN can be incorporated within the command string.

USER VAR is one of the many debugging aids that allows you to list all variable names, used within a program, along with their values.

Because I have only been allowed a certain amount of space, it is impossible to do this program full justice. Perhaps the ED will allow me further space to discuss this utility in more depth.

This is definitely a must for any one wishing to have very useful extra commands at their finger tips, and should the Rom ever get re-written, this utility is a must for inclusion.

Dr. B. Sandiford.

★★★★

KEYBOARD KAPERS *peter knaggs.*

MTX KEYBOARD MAPPING

OUTPUT TO
PORT 5

INPUT FROM PORT 5

INPUT FROM PORT 6

BINARY	HEX	0	1	2	3	4	5	6	7	0	1
11111101	FD	ESC	1	"	£	\$	%	&	'	()
11111110	FE		1	2	3	4	5	6	7	8	9
11111011	FB	CTRL	Q	W	E	R	T	Y	U	I	O
11110111	F7									P	.
11101111	EF	CAP	A	S	D	F	G	H	J	K	L
11011111	DF	LCK								:	;
10111111	BF	SH	Z	X	C	V	B	N	M	<	>
01111111	7F	IFT								/	-

NB: TOP NUMBER FOR LEFT CHARACTER

) KEY PRESS INDICATED BY RESETTING BIT SHOWN

BOTTOM NUMBER FOR RIGHT CHARACTER

OUT (5),FE

IN A,(6)

BIT 0,A

JR Z, BREAK_KEY_PRESSED

PEEKing the MTX Keyboard

In the second edition of MEMOPAD we gave you some ideas of how the MTX Keyboard works. During the recent months we have had lots of queries as to how the MTX is scanned, and with this in mind we asked our old mate Peter Knaggs to go into some depth on the subject. Here are his findings.

THE DEFINITION OF PORTS 5 AND 6 ARE GIVEN AS FOLLOWS:

OUT (5),d This latched port provides the 8 drive lines of an 8 x 10 keyboard matrix.

IN (5),d This port is used to read the least significant 8 bits from the ten bit sense line of the 8 x 10 keyboard matrix.

IN (6),d This port is used to read in the two most significant sense lines of the 8 x 10 keyboard matrix.

The keyboard is divided into 8 sections. Each section contains 10 keys. The Sherlock Holmes amongst you will notice that this means that the keyboard is capable of responding to 80 keys as opposed to the 77 (not including the Reset keys) that have been provided. An OUT to Port 5 will select which line or section you wish to examine. An IN from Port 5 will give the first 8 keys on the selected section while an IN from Port 6 will give the remaining 2 keys of the section.

SELECTING A SECTION

To select a section you will need to output a value to Port 5. This value must be calculated in binary and then converted to decimal (for Basic) or hex (for Machine Code). For all the sections you do not wish to access the bits must be set to 1, whilst for the section you wish to access the bit must be set to 0 - remember that the MTX Keyboard works back to front.

Bit	Binary	Decimal	Hex
	7 6 5 4 3 2 1 0		
	1 1 1 1 1 1 1 0	254	FE -Access Section 0
	1 1 1 1 1 1 0 1	253	FD -Access Section 1
	1 1 1 1 1 0 1 1	251	FB -Access Section 2

You could access more than one section if you chose but this would only lead to confusion so I stick to the simple method until you get to grips with the idea.

TO TEST A KEY

To test for a key press you must perform an IN from Port 5, and this will give the first 8 keys. If a key is pressed then the bit will be a 0 (nought) as opposed to a 1 for a key that is not pressed. If each bit is 1 then no key has been pressed and you must then look at the first two bits of Port 5 or the further 2 keys of the 10 per section. The coding for this proves quite difficult, if not impossible in Basic. This is due to the lack of Logical AND & OR statements. However, it is much simpler in machine code and I see no reason why this short program can not be written and called as a sub routine from your Basic program.

Example:

To test if the user was pressing the Break key for example would need the code:

```
BRKEY: LD A,#FE ; Select section 0
      OUT (5),A ; Tell the Keyboard about it
      IN A,(6) ; Get the last 2 keys
      BIT 0,A ; Is BRK pressed ?
      RET ; Exit Subroutine
```

This routine will return Z if BRK was pressed or NZ if not.

To make life simpler I have drawn up the following table. Out put the Top Number for the Left-hand keys and the Bottom number for the Right hand keys.

i.e; Output of FD will set the keys : ESC 2 4 6 8 0 ↑ Eol BS F5
 Whilst FE will set: 1 3 5 7 9 - \ Page Brk F1

If you wish to explore this matter further then I suggest that you type in the following coding. You will have to place the correct value at memory location 4009 (8009 for MTX 500's) for the output and you will have to use the reset keys to exit from the program - the reset keys are not detectable on the keyboard.

10 CODE

```
4007 DI ; Stop Rom Keyboard scanning
4008 LD A,#FE ; Select Keyboard Section/Line
400A OUT (5),A ; Tell the Keyboard about it
400C LOOP: RST 10
400D DB 26 ; Home the Cursor
400E IN A,(6) ; Input the top 2 keys
4010 AND 3 ; Strip off KB Configuration bits
4012 CALL BYTE ; Display the byte in binary
4015 IN A,(5) ; Input to bottom 8 keys
4017 CALL BYTE ; Display the byte in binary
401A JR LOOP ; Repeat it all again until RESET
401C BYTE LD B,8 ; Set bit counter
401E BLOOP RLC A ; Move byte left one
4020 PUSH AF ; Save result
4021 AND 1 ; Strip down to 1 or 0
4023 ADD A,"0" ; Add ASCII "0" to result
4025 RST 28
4026 DB #AC ; Print character in A
4027 POP AF ; Recover A
4028 DJNZ BLOOP ; Repeat for all of byte
402A RET
```

Symbols:

```
LOOP 400C BYTE 401C
BLOOP 401E
```

FORTH another view SYNTAXsoft

Have you ever looked for a language which would give you the ease of use of a high level language, the ability to manipulate all of the machine resources that assembly language does, the speed of machine code and yet be compact enough that you could fit it in an interactive compiler, assembler, text editor and disc operating system all within 5 to 6 K bytes? You have? then look no further, FORTH will give you all this and much more. FORTH is a different approach to the problem of communicating with computers. It is an English like language whose elements are verbs, nouns and defining words. Verbs cause a sequence of computer operations to occur, nouns are the objects operated upon and defining words cause new words to be defined in terms of previously defined ones, or in terms of machine code. Complex operations, for example a word processor, would be defined in terms of less complex tasks, moving characters in memory, displaying characters on the screen etc., those tasks themselves are defined in terms of more basic words and so on until the final words are defined in terms of machine code. This gives you the programmer a great advantage as at every stage in the program development it is possible to check for correct operation simply by typing the name of the word on your keyboard.

MTX FORTH is an implementation of the full, or very nearly so, F.I.G. standard with revisions as specified by F.I.G. This gives you a better start than several other versions as you know exactly what words have been initially defined. Any extensions to the standard are to drive the MTX graphics and sound. What you get for your money is the normal interpreter/compiler and a text editor, quite often the editor is supplied on tape and you have to load it from tape prior to using it, having it in ram is a very nice touch. The editor is simplicity to use once you get used to it, as with everything in FORTH it's different if all you have been using is BASIC. The editor allows you to create screens of FORTH program, currently five in all each capable of holding fifteen lines of 64 characters i.e. roughly 1 K byte of FORTH. This might not seem a lot but believe me (using the same kind of sales hype that a certain gentleman from Cambridge used) 1 K byte of FORTH code is worth at least 3 or 4 K bytes of BASIC !! The reason for only implementing five screens is a trade off between the number of screens used in the editor and the amount of ram available for the compiled definitions. It is possible to implement more screens and I am led to believe that more screens are to be implemented in the near future.

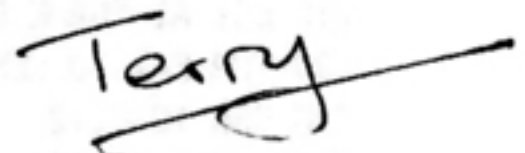
FORTH is fast as stated in the last issue of MEMOPAD a DO LOOP containing no definitions just executed as a wait loop will take some 2.5 seconds in FORTH to complete 30000 passes through the loop, the same loop in BASIC takes about 33 seconds. I recently coded a definition to find prime numbers, the first set of primes up to 100 took 5 seconds in FORTH, I haven't the heart to code it in BASIC!

The manual supplied is really just the standard F.I.G. definitions there are a couple of omissions in the manual of words that are actually defined, e.g. WARM WHERE LOADT CURRENT ?TERMINAL however most of these have been defined in the series of articles in MEMOPAD, I'm sure this will be rectified in future copies of the manual. As Keith himself said in the articles the manual is not intended to teach you FORTH programming, merely to serve as a reference text to those who know FORTH. For those new to FORTH it would be advisable to pick up one of the many good books on FORTH, I would recommend STARTING FORTH by Leo Brodie of FORTH Inc. The book is a little expensive but it takes you from knowing nothing about the language to a considerable proficiency in a light hearted and easy to follow manner, and there's lots of examples and programs to try which build up your confidence.

All in all I would recommend this version of FORTH very highly, it gives the average MTX owner another powerful language at a realistic price and opens up a whole new dimension of programming to those who try it. I would give a warning though, those who learn FORTH rarely want to return to BASIC. ★

Dear Keith,

Thanks for the copy of FIG FORTH, I think you're on to a winner with this one, at least I hope you are. Well here's the review I must say I'm impressed that Keith has implemented virtually the full FIG specification a lot of the others that have been produced have not done so or have mixed various different revisions of the specification so that you don't know what is actually defined and what isn't anyway I like it a lot !!



Terry Trotter.

BASIC GRAPHICS Part 5 Michael Gant

MICHAEL CARRIES ON FROM WHERE HE LEFT OFF IN LAST MONTHS EDITION WITH AN EXPLANATION OF THE GAME & FURTHER ROUTINES TO FIRE THE BULLETS AND GENERALLY TIDY UP THE PROGRAM.

Lines 1000 to 1060 check the key pressed and adjust the coordinates accordingly. If the plane goes off the right of the screen, then it is taken back to the left of the screen. If the space bar is pressed and a missile is not already on, then the missile coordinates are set and the missile flag is set.

Line 2000 to 2120 move the missile if necessary. They also check if the missile has hit the tank (it will have the same coordinates) and adjusts the score if so. The missile will also "wrap" around the screen if it travels off the side.

Let's move on now and write the routine to move the tank and fire its bullets.

```
95 LET TX=31:LET TC=130:LET TM=0:LET BC=133:LET TF=0:LET BX=0:LET BY=0
98 CSR 0,0:PRINT "SCORE: 0":CSR 20,0:PRINT "LIVES: 3"
130 GOSUB 3000
3000 LET TX1=TX-1:LET TC=TC+1
3010 IF TC=133 THEN LET TC=130
3020 IF TX1=-1 THEN LET TX1=31
```

```
3030 CSR TX,23:PRINT " ";CSR TX1,23:PRINT CHR$(TC);
3040 LET TX=TX1
3050 IF TF=1 THEN RETURN
3060 LET BX=22:LET BY=TY:LET TF=1:RETURN
```

The tank movement routine situated between 3000 and 3060 quite simply adds one to the current character pattern, resets the pattern if at the end of the animation cycle, moves the tank and fires if necessary. The only section we need to write now is the bullet movement routine. Add these lines to the program:-

```
97 LET LV=3
140 GOSUB 4000
4000 IF TF=0 THEN PAUSE 100:RETURN
4010 LET BC=BC+1:IF BC=135 THEN LET BC=133
4020 LET BX1=BX-1:LET BY1=BY-1
4030 IF BX1=0 THEN LET BX1=31
4040 IF BY1=0 THEN CSR BX,BY:PRINT " ";LET TF=0:RETURN
4050 IF BX1<>PX OR BY1<>PY THEN GOTO 4070
4060 LET LV=LV-1:CSR PX,PY:PRINT " ";CSR BX,BY:PRINT " ";LET PX=1:LET PY=1:LET BY1=0:GOTO 4040
4070 CSR BX,BY:PRINT " ";CSR BX1,BY1:PRINT CHR$(BC);
4080 LET BX=BX1:LET BY=BY1:RETURN
```

To tidy up the program, add these lines as well:-

```
150 CSR 3,6:PRINT SC
160 CSR 26,0:PRINT LV
170 IF LV=0 THEN GOTO 210
210 LET A$="GAME OVER"
220 FOR F=1 TO LEN (A$)
230 CSR 10+F,12
240 PRINT MID$(A$,F,1);CHR$(136)
250 PAUSE 500
260 NEXT F
```

```

270 CSR 10+F,12:PRINT " "
280 CSR 5,14:PRINT "DO YOU WANT ANOTHER GO?"
290 LET KEY$=INKEY$
300 IF KEY$="Y" THEN RUN
310 IF KEY$<>"N" THEN GOTO 290
320 STOP

```

The game has been included to allow you to develop it thereby becoming more familiar with your machine. One obvious addition is sound. I haven't used any sound at all for the simple reason that we are not running a sound article! Also, improved logic can be added enabling the tank to actually hit you! In short, I have provided the skeleton program for you to develop. Any GOSUB's to your own routines should be placed between lines 100 and 200. Your routines must end with a RETURN statement otherwise the whole program will malfunction after a short time. Anyway, whatever you try, be sure to have fun! ★

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 Tml10, 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

MTX 512, Good condition, Boxed, with:-

Sale

Printer Lead, EDASM, Colossal Adventure
 Pot Hole Pete, Super Minefield, Blobbo,
 Draughts, Toado. Z80 Programming book,
 and some data sheets on CTC, etc.

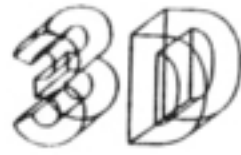
£170 ono.

Contact: G.J.Ratcliffe

45 Hermitage Road, Mannamead, Plymouth.
 Devon. Sorry no 'phone.

SPEEDY

PT2



Written by Mark Jones
Adapted for the MTX by

```
4704 PARA8: DS 20
4718 PARA9: DS 20
472C PARA10: DS 20
4740 PARA11: DS 20
4754 PARA12: DS 20
4768 PARA13: DS 20
477C PARA14: DS 20
4790 PARA15: DS 20
47A4 PARA16: DS 20
47B8 ENDPARA:NOP
47B9      NOP
47BA      NOP
47BB      RET
```

Symbols:

```
COORD4647REMLP448A
DECRAD44F6COMP04504
COMPI44EFSIMPI44EC
INCRAD44E1NUMB4629
ADDR462APX462C
PY462EPZ4630
PHI4632THETA4634
```

```
200 REM      SPEEDY-3D
```

```
210 REM      WRITTEN BY MARK JONES
```

```
220 REM      CONVERTED FOR THE MTX
```

```
230 REM      BY KEITH ATKINSON.
```

```
240 GOTO 1000
```

```
250 REM SET PARAMETER BLOCKS
```

```
260 LET PARA=18020+FIG*20
```

```
270 POKE PARA,NUMB
```

```
280 LET ADDR1=PARA+1
```

```
290 LET C=ADDR
```

```
300 GOSUB 500
```

```
310 LET ADDR1=ADDR
```

```
320 FOR F=1 TO NUMB
```

```
330 READ X,Y,Z,P
```

```
340 LET C=X: GOSUB 500
```

```
350 LET C=Y: GOSUB 500
```

```
360 LET C=Z: GOSUB 500
```

```
370 POKE ADDR1,P
```

```
380 LET ADDR1=ADDR1+1
```

```
390 NEXT
```

```
400 LET SPARE=(12*NUMB)+1
```

```
410 LET FREE=SPARE+ADDR1
```

```
420 LET C=ADDR1
```

```
430 LET ADDR1=PARA+15
```

```
440 GOSUB 500
```

```
450 LET C=SPARE/2+C
```

```
460 GOSUB 500
```

```
470 POKE ADDR1,0
```

```
480 PRINT "DATA FOR FIGURE ";FIG: PRINT "STORED AT ";ADDR: PRINT "PARAMETERS FOR FIGURE ";FIG: PRINT "STORED AT ";PARA;" TO ";PARA+19
```

```
490 RETURN
```

```
500 REM CALCULATE LOB HOB
```

```
510 IF C<0 THEN LET C=C+65536
```

```
520 POKE ADDR1+1,INT(C/256)
```

```
530 POKE ADDR1,INT(256*(C/256-INT(C/256)))
```

```
540 LET ADDR1=ADDR1+2
```

```
550 RETURN
```

```
590 REM CUBE DATA
```

```
600 DATA 20,-20,20,0,20,-20,-20,1,-20,-20,-20,1,-20,-20,20,1,20,-20,20,1,20,20,20,1,20,20,-20,1,-20,20,-20,1,-20,20,20,1,20,20,20,1,20,-20,-20,0,20,20,-20,1
```

```
610 DATA -20,-20,-20,0,-20,20,-20,1,-20,-20,20,0,-20,20,20,1
```

```
1000 REM EXECUTE DEMO
```

```
1010 LET DEMO=16400
```

```
1020 LET FIGNO=17960
```

```
1030 LET PARAM=18020
```

```
1040 LET ADDR=53248
```

```
1050 LET CALCULATE=16959
```

```
1060 LET DRAWSHP=17421
```

```
1070 LET ERASESHP=17408
```

```
1090 LET NUMB=16: REM SETS OF DATA
```

```
1100 RESTORE 600: LET FIG=0: GOSUB 250
```

```
1110 RESTORE 600: LET FIG=1: LET ADDR=FREE: GOSUB 250
```

```
1120 RESTORE 600: LET FIG=2: LET ADDR=FREE: GOSUB 250
```

```
1200 POKE DEMO+1,10: REM SPEED
```

```
1210 VS 4: CLS
```

```
1220 POKE FIGNO,1
```

```
1230 RAND USR(DEMO)
```

```
1240 POKE FIGNO,0
```

```
1250 RAND USR(DEMO)
1260 POKE FIGNO,2
1270 RAND USR(DEMO)
1300 FOR F=5 TO 95 STEP 10
1310 POKE DEMO+1,F
1320 RAND USR(DEMO)
1330 NEXT
1350 LET FIG=0
1360 FOR X=1 TO 360 STEP 10
1370 POKE PARAM+9,X
1380 POKE PARAM+7,X/4
1390 GOSUB 1900
1400 NEXT
1410 CLS
1420 FOR X=0 TO 200 STEP 5
1430 POKE PARAM+7,X
1440 POKE PARAM+13,X
1450 GOSUB 1900
1460 NEXT
1470 FOR X=0 TO 359 STEP 10
1480 POKE PARAM+13,X
1490 GOSUB 1900
1500 NEXT
1505 GOTO 1410
1510 DSI
1520 STOP
1900 REM THIS GOSUB CALLS MC ROUTINE
1910 REM FROM BASIC PROGRAMS
1950 POKE FIGNO,FIG: REM FIG = 0 TO 15
1960 RAND USR(CALCULATE)
1970 RAND USR(ERASESHP)
1980 RAND USR(DRAWSHP)
1990 RETURN
2000 REM
2010 REM ROUTINE TO UTILISE THE
2020 REM FAST DRAWING ROUTINE.
2030 REM
2040 LET DAT=53248: LET PARAM=18020: RESTORE 3000
2050 FOR X=0 TO 15
2060 READ R
2070 POKE DAT+X,R
2080 NEXT
2100 LET ADDR1=PARAM+15: REM DRAWS
2110 LET C=DAT
2120 GOSUB 500
2130 LET ADDR1=PARAM+17: REM DRAWP
2140 LET C=DAT
2150 GOSUB 500
```

```
2150 GOSUB 500
```

```
2160 VS 4: CLS
```

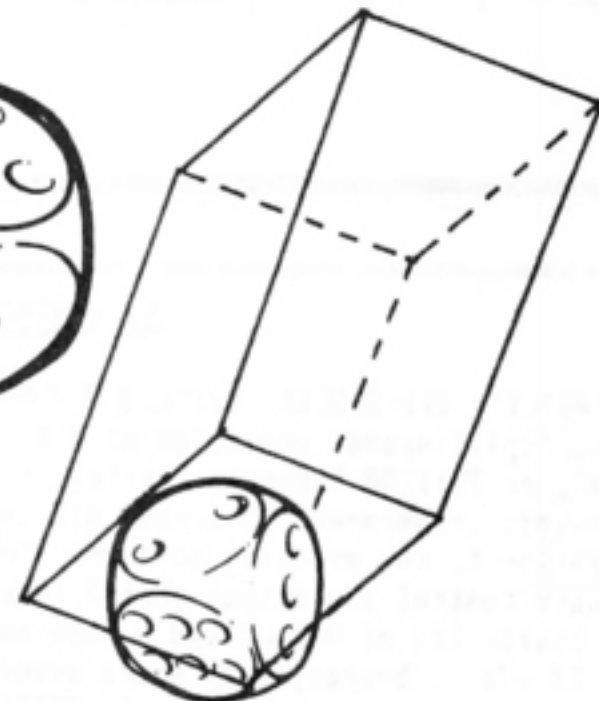
```
2200 POKE PARAM+19,255: REM STFLG
```

```
2210 RAND USR(17421): REM DRAW
```

```
2220 DSI
```

```
2230 STOP
```

```
3000 DATA 0,0,0,1,255,0,1,255,191,1,0,191,1,0,0,255
```



end

Take A Note (any note) *Derick Brown*



DERICK BROWN TAKES TIME OFF FROM COMPOSING TO BRING YOU THE BENEFIT OF HIS EXPERIENCES WITH THE MEMOTECH SOUND CHIP. WE SHALL END THIS SERIES WITH A PROGRAM THAT DEMONSTRATES THE POSSIBILITIES AVAILABLE TO YOU WHEN PROGRAMMING IN BASIC. STUDY THIS SERIES BECAUSE, WHEN THE SERIES IS FINISHED, WE SHALL BE RUNNING A COMPETITION FOR THE BEST INTERPRETATION OF A PIECE OF MUSIC OF YOUR OWN CHOICE.



The Editor has invited me to write on the methods I use to transcribe music for the Memotech. Many of you will be familiar with the occasional sound effect required by some games and would like to move on to a complete tune. Sadly the manual does not enter into this aspect and I have yet to come across a specific article on music for the MTX in any of the magazines. I have always had an interest in music from my days as a choir boy, and my knowledge of this subject has been self-taught over the years. When I became aware that a computer could be programmed to play a tune I decided to investigate, thus the story begins just 15 months ago when I purchased the MTX 512 on the advice that it could do more than the BBC model and was cheaper!

Some members may recall receiving a Newsletter from Memotech (April 1984) on the last page of which was a Sound routine attributed to C. Baynes. I noticed that some of the figures contained in the DATA lines were way beyond those given in the manual (e.g. 3200) and were converted to frequencies using the formula $LET N (frequency) = (4E6/36*(F))*8$, thus differing from that in the Sound appendix.

I was informed that 4E6 is a scientific representation of a floating point number. The number after the E is the exponent part, and it shifts the decimal point that number of times to the right (or left if the exponent is negative) thus $4E6 = 4 * 10$ to the power of 6 which is 4000000. Now we know! Later on I was to learn that this figure is also the frequency of the clock in Hertz so it appears all frequencies on the MTX are governed by the clock. I would like to know the reason for the figure 32 in the MTX formula - can anyone explain? The computer analyst was unable to say why 36 had been used as opposed to 32 so I made contact with C. Baynes c/o Memotech to learn that the Newsletter had contained mis-prints.....it should have been 32 and no "s" on the surname.

The DATA figure 32000 acted as a rest. I have yet to probe C. Bayne's program further as he uses a different method to my system. A brief mention now of Ian Sinclair's book "Memotech Computing" which devotes chapt. 10 to Sounding out Memotech. He reveals much useful information on sound but I cannot go along with all that is stated, particularly the ranges of volume control and the table on page 149 which gives note values extending over 7 octaves. There is a mis-print by the discerning musician. From this point on, any facts and figures quoted have been extracted from "The World of Music" a two-volume encyclopaedia published by the Waverly Book Co. Ltd. To play music on an instrument it is essential that it is in tune, thus we have to "tune" the MTX to produce the frequencies for the notes we wish to play. For our purpose we can regard the black notes as being flats or sharps and whilst it is said that players of stringed instruments can differentiate between for example C sharp and D flat, on the keyboard one black key has to serve for both notes. For this reason, keyboard instruments are tuned to a tempered scale.

The frequency of middle C is given as 261.6 and if we use this in the MTX formula $N = 4000000 / (32 * 261.6)$ we get a value for N as 477.828746. Now take a look at the diagram showing one octave of a keyboard and the column headed Tempered scale. Frequency figures double at each octave but the Memotech value numbers work in reverse - they are halved - so we need to divide our frequency of middle C by the table figures to give frequencies over one octave. From these other octaves were calculated, converted to values and put in a program to prove that the lowest note the MTX would play was 1015.0371 and the highest 23.701822. This gives a range of 5 octaves plus 5 semitones.

Value 1015.0371 equates to the note B which is one octave and a semitone below middle C. What would I do for notes below B? The solution was to move every note down one octave, thus I now have to regard the value for middle C as being 238.91437. At this point another curious fact emerged - the figures used to calculate the frequencies when working from middle C did not hold good when starting from B, so I had to start all over again! These figures are shown below as we move on to what I term as the "note generator." This stores the frequencies in an array which are called upon as and when required by

means of subscript values contained in DATA lines. We have 66 notes (semitones) available and proceed with:

```

30 DIM N (67)
31 LET N(1) = 1015.0371
32 LET N(2) = N(1)/1.062
33 LET N(3) = N(1)/1.124
34 LET N(4) = N(1)/1.191
35 LET N(5) = N(1)/1.262
36 LET N(6) = N(1)/1.338
37 LET N(7) = N(1)/1.417
38 LET N(8) = N(1)/1.501
39 LET N(9) = N(1)/1.591
40 LET N(10) = N(1)/1.685
41 LET N(11) = N(1)/1.786
42 LET N(12) = N(1)/1.892
43 LET N(13) = N(1)*.5
44 FOR X=13 TO 66
45 LET N(X) = N(X-12)/2
46 NEXT X
47 LET N(67) = 8

```



I do not know the reason for LET N(67)=8. A cassette bearing some classical music was passed on to me and contained the following:

CHROMATIC SCALE (FOUR OCTAVES)

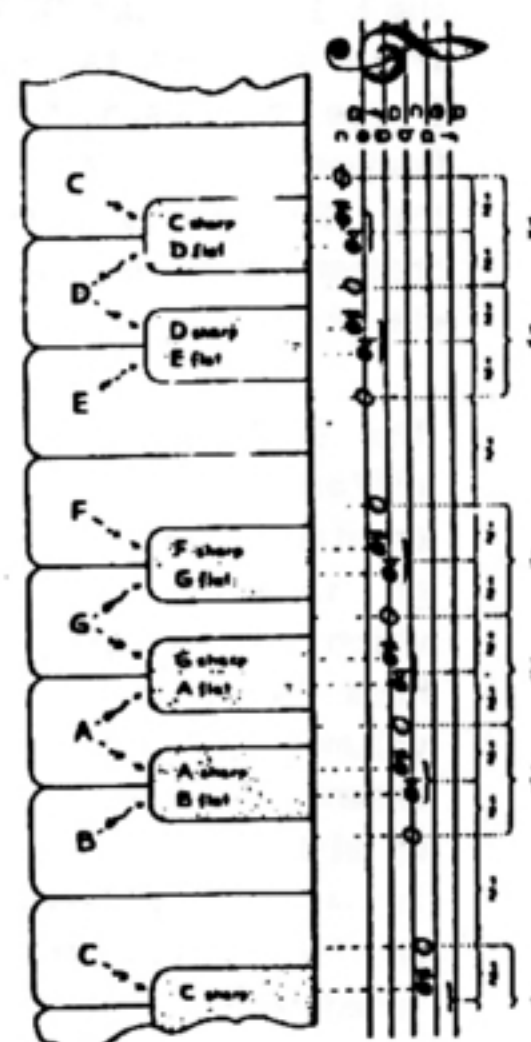
INTERVALS WITHIN THE OCTAVE

```

10 DIM N(50)
15 LET N(1) = 956
20 LET N(6) = N(1)*.75
25 LET TC1 = (N(1) - N(6))/5
30 FOR X=1 TO 4
35 LET N(X+1) = N(1) - X*TC1
40 NEXT X
45 LET N(13) = N(1)*.5
50 LET N(8) = N(13)/.75
55 LET TC2 = (N(8) - N(13))/5
60 FOR X=1 TO 4
65 LET N(X+8) = N(8) - X*TC2
70 NEXT X
75 LET N(7) = (N(6) + N(8))/2
80 FOR X=13 TO 49
85 LET N(X) = N(X-12)/2
90 NEXT X
95 LET N(50) = 8

```

From the tonic C to:		Relative Frequencies		
		Just Intonation		Tempered scale
C	Unison	1	1.000	1.000
C#	Semitone	25/24	1.042	1.059
Db	Minor Second	27/25	1.080	1.059
D	Major Second	9/8	1.125	1.122
D#	Augmented second	75/64	1.172	1.189
Eb	Minor third	6/5	1.200	1.189
E	Major third	5/4	1.250	1.260
Fb	Diminished fourth	32/25	1.280	1.260
E#	Augmented third	125/96	1.302	1.335
F	Perfect fourth	4/3	1.333	1.335
F#	Augmented fourth	25/18	1.389	1.414
Gb	Diminished fifth	36/25	1.440	1.414
G	Perfect fifth	3/2	1.500	1.498
G#	Augmented fifth	25/16	1.562	1.587
Ab	Minor sixth	8/5	1.600	1.587
A	Major sixth	5/3	1.667	1.682
A#	Augmented sixth	125/72	1.736	1.782
Bb	Minor seventh	9/5	1.800	1.782
B	Major seventh	15/8	1.875	1.883
Cb	Diminished octave	48/25	1.920	1.883
B#	Augmented seventh	125/64	1.953	2.000
C	Perfect octave	2	2.000	2.000



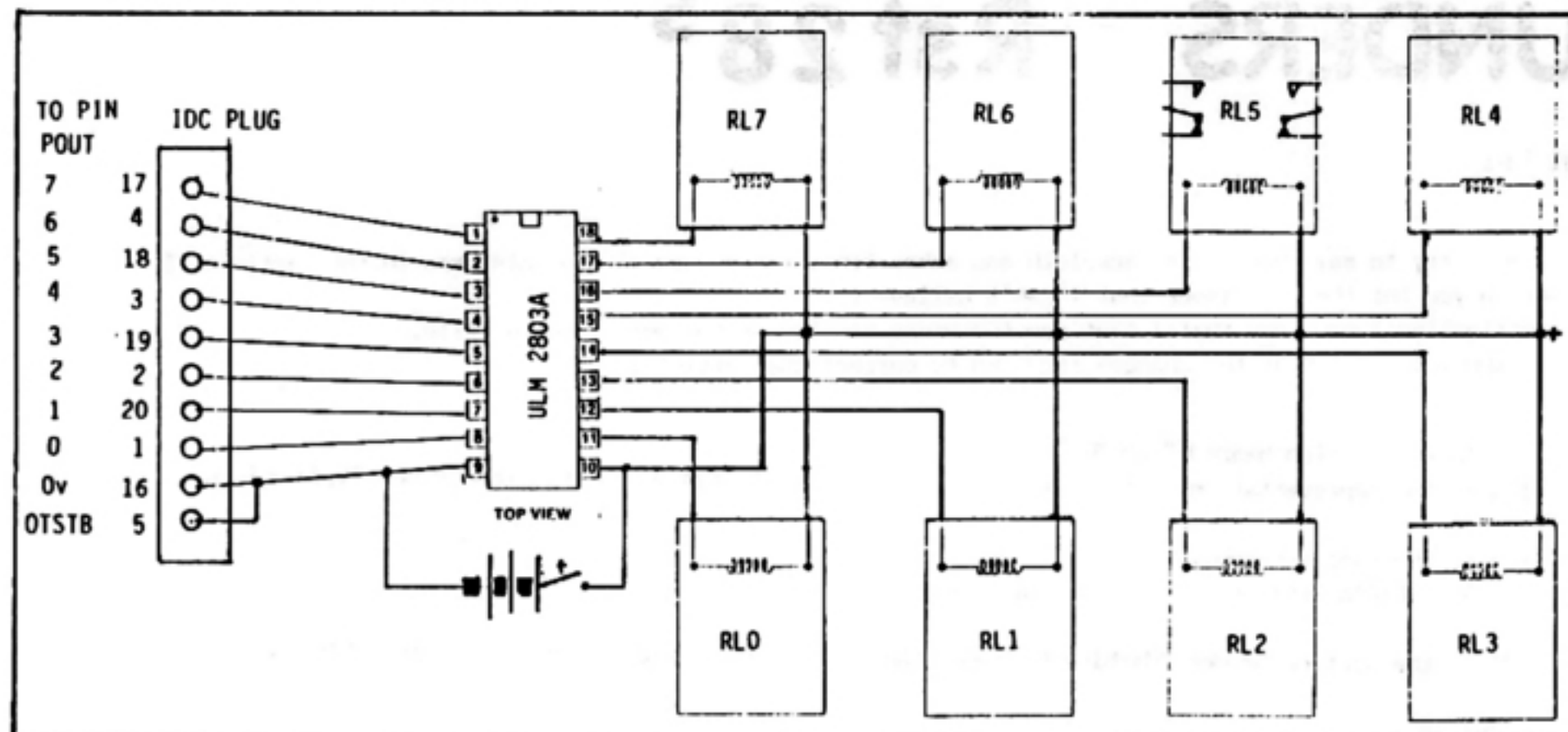
This is a similar note generator where the notes over 4 octaves are calculated mathematically. The octave is split into two halves termed tetrachords with three equal steps plus a half-step for the semi-tone in each half. The steps in each half are different, the two tetrachords being stacked one above the other. The analyst previously mentioned explained how it functioned and how it could be extended to 66 notes. The line LET N(50)=8 served to produce silence and in adapting the generator idea to my system I had to amend 50 to 67. To see how it performs, enter a value of 100 for N1 in line 15. Add lines: 100 FOR X=1 TO 13 and 105 PRINT X,N(X) and run.

It would be interesting to see these steps as frequencies, but with the MTX values working in reverse I've not been able to do it. Can anyone help? ★



AN EASY INTERFACE

MIKE BRAY. Pt2



The circuit which is shown below was constructed on a Vero V Q Board designed for dual-in-line circuits. (Maplin HQ 48C). This is a good base in many respects and is ideal for relay mounting but does need a large number of interconnections on the top of the board. The relays were soldered directly to the board but an 18 pin DIL socket (Maplin HQ 76 H) was used to mount the IC chip. First the design or layout should be decided and then the interconnecting wires soldered into place. After this is completed the DIL socket and relays can be soldered in and the battery and switch wired in. By connecting the battery and shorting the pins (11-18) on the DIL socket in turn to pin 9 each relay may be checked for operation.

An IDC Header Plug (Maplin FJ 14Q 20way) is then wired to pins 1-9 of the DIL socket as shown. I shorted pins 5 (OTSTB) and 16 (OV) on the back of this plug.

Finally connect all the relay contacts to suitable connector blocks remembering there are 16 change over switches available and install in a suitable box with the IDC header plug located to plug into the computer lead already made up.

INITIAL TEST

Plug in the IC the correct way round into its socket and connect the interface to the computer. Switch both computer and interface on. Type in the following program:

```
10 LET X = VAL (INKEY$): IF X = 0 THEN GOTO 10
20 OUT 7, (2^X/2)
30 PAUSE 500
40 OUT 7, 0
50 GOTO 10
```

Press each number key from 1-8 in turn and the appropriate relay should click in and then out. A simple light circuit connected across one of the relay contacts should flash when that relay number is keyed in.

CONCLUSION

The project described gives an easy method of controlling larger voltages and currents than the computer would normally handle. There are limitations in as much as the relays are very slow devices in computer terms, however, this limitation should not cause much problem if the devices under control are similar to those described in the text.

If the interface is going to be used in such a manner that all the relays will be energised most of the time then it would be preferable to use relays with higher coil resistances so that the IC chip does

not have to handle too high a power for too long. I have tested the above circuit over a number of cycles of all the relays switching without any noticeable change in chip temperature - however it is better to play safe. ★

BLUNDERS Rst 28^s

Hi Keith,

I am sorry to say that it is sackcloth and ashes time, there were so many mistakes in the article I sent to you for the last issue that I can't believe it.

In mitigation I must say that I sent you the wrong version of the wordprocessor file.

This then, is a list of the changes required to correct that article.

(1) In the section headed "The MTX"

I show the representation of the five byte quantity as 134,172,0,0,0 this should be 134,72,0,0,0.

(2) In the same section...

LET A = 50:PAN should be LET A = 50:PANEL

(3) In the section headed "Machine Code Programs." the second example of RST 28 should be...

```
RST 28
DB #C8,#9B
CALL #12D4
```

This says multiply ACC1 by OP1 and put data to location pointed to by HL

(the data is passed from ACC1 to MEM1 by the 9B command and to the location pointed to by HL by the call #12D4.

(4) In the section headed "Rom Locations" there are several mistakes, the corrections are shown below.

DB	Call Add	Comments
82	12D1	Moves MEM1 to ACC1
8A	109E	Divide OP1 by ACC1
8E	1289	Move (DE) to OP11
90	0E83	Subtract ACC1 from OP1
9A	13D2	Executes the polynomial pointed to by HL...etc
9B	12BF	Moves ACC1 to MEM1

Also add to this section...

11C5	Convert f.p. No. in ACC to Int in BC
125A	Move (DE) to (TOS)
12A4	Move PI (22/7) to ACC1
12D4	Move MEM1 to (HL)
134B	Converts no. in BC to f.p No.in ACC1

I have also realised that the section describing the use of DE,HL etc is not too clear.

(DE) should say...

equates to the contents of the area pointed to by the DE register pair.

(HL) should be similar to (DE)

I am very sorry about the mistakes, perhaps next time I will delete all old versions of files as soon as possible.

PS. There are many more RST 28 functions which do not perform mathematical functions and I will describe these at a later date. They're more difficult to use as most of them involve ensuring that the correct ROM is paged into store.

Cheers for now, with apologies,

John Hudson ★

Assembly Line

EDITED BY TERRY TROTTER

SEND ALL SUBMISSIONS FOR THIS COLUMN TO
TERRY TROTTER, ASSEMBLY LINE, 65 MARSH
HOUSE, BILLINGHAM, CLEVELAND, TS23 2HW



The purpose of Assembly Line is to encourage the use of the MTX Z80 assembler and also EDASM the macro assembler amongst the members of GENPAT. It is often useful to have at one's fingertips a set of assembler subroutines which can be used as necessary, this is what Assembly Line hopes to do. The page is your page and will not be what it was envisaged to be unless all you budding Z80 programmers set fingers to keys and come up with some useful programs. We hope, each month, to have a pageful of assembly routines that control the MTX in various ways. The routines could be for example; sound chip driver routines, keyboard scan routines, maths routines, graphics drivers, CTC setup and DART setup routines. The code you send should be documented and commented and give a resume of the register usage, stack usage etc. as in the code shown below.

I think it's true to say that a lot of us bought this machine because of its technical excellence so let's help each other to get the most out of it.

Bubble Sort Subroutine

Title: BUBSORT

Standard exchange sort of 8 bit data.

On entry HL contains the starting address of the data to sort.

C contains the number of elements to be sorted, C must be in the range 1 to 255.

On exit the data is sorted in ascending order.

10 CODE

Register usage:

A temporary store
B counter for the array
C length of the array
D first element in the comparison
E second element in the comparison
H bit 7 used to flag exchanges
IX pointer into the array

Stack usage: NONE

Length : 45 bytes

```

4007 BUBSORT: LD (ARRAYAD), HL      ; save array start address
400A CONTIN: RES 7, H               ; initialise exchange flag, using bit 7 of H register
400C          LD B, C               ; initialise length counter, maximum 255
400D          DEC B                 ; adjust for testing
400E          LD IX, (ARRAYAD)      ; set up array pointer
4012 NEXTEL:  LD A, (IX+0)           ; fetch first element
4015          LD D, A               ; save it temporarily
4016          LD E, (IX+1)          ; fetch second element
4019          SUB E                 ; compare the two elements
401A          JR NC, NOEXCH         ; if first element > second element, no jump happ
ens
401C          LD (IX+0), E          ; else exchange the array elements
401F          LD (IX+1), D
4022          SET 7, H               ; note that an exchange took place
4024 NOEXCH:  INC IX                ; point to next element
4026          DJNZ NEXTEL           ; loop for all the pairs of elements
4028          BIT 7, H              ; has an exchange occurred?
402A          JR NZ, CONTIN         ; if so continue to loop and exchange
402C          RET                   ; otherwise we're finished
402D ARRAYAD: DS 2                 ; storage for the start address of the array
402F          RET
4030          RET

```

Symbols:

```

BUBSORT 4007      CONTIN 400A
NEXTEL  4012      NOEXCH 4024
ARRAYAD 402D

```

ASCII to Hex conversion subroutine

Title : ASCHEX :

The routine takes standard ASCII characters for the numbers 0 to 9 and A to F and converts them to valid hex digits 0 to F. The routine returns with the valid digit in the A register or with the CARRY flag set to indicate a non valid character.

Register usage:

A Used to pass the character to and convey the answer from the subroutine
CARRY flag Set if non valid ASCII character, otherwise reset

10 CODE

Stack usage : NONE

Size : 18 bytes

Hex to ASCII conversion subroutine.

Title : HEXASC :

Takes hex digit in the lower nibble of the A register and converts it to a valid ASCII character suitable for printing.

Register usage :

A Used to pass the digit to and convey the character from the subroutine

Flags will be affected

Stack usage : NONE

Size : 9 bytes

10 CODE

```

4007 HEXASC: AND £0F      ;mask off unwanted upper nibble
4009        ADD A,£90     ;due to the values of the
400B        DAA           ;ASCII code and the way DAA ( decimal adjust )
400C        ADC A,£40     ;works, it's easier to try
400E        DAA           ;this than explain it !
400F        RET

```

Symbols:
HEXASC 4007



USER BASIC toolkit £8.95

ADD 20 EXTRA COMMANDS TO MTX BASIC WITH THIS ULTIMATE UTILITY PROGRAM. AVAILABLE ON DISC AND TAPE.

USER BASIC SAVES A CORE IMAGE OF ITS RUN TIME OVERLAY ALONG WITH YOUR PROGRAM WHICH MEANS THAT YOU CAN GIVE THE PROGRAM TO ANY OTHER MTX OWNER AND THEY WILL BE ABLE TO RUN IT EVEN THOUGH THEY DON'T HAVE USER BASIC.

ONCE LOADED YOU HAVE THE FOLLOWING COMMANDS AVAILABLE

CALL LINENAME ... ALLOWS YOU TO NAME A LINE NUMBER THEN CALL IT BY ITS NAME. ASSOCIATED WITH THIS COMMAND ARE: JUMP & RESTORE.

REPEAT & UNTIL_expression WHILE & ENDW

%VARIABLE ALLOWS YOU TO ASSIGN HEX,BINARY OR LOGICAL VALUES TO A VARIABLE.

INSTR SUB\$,MAIN\$ WILL FIND THE FIRST OCCURANCE OF ONE STRING IN ANOTHER AND WILL RETURN WITH ITS POSITION IN VARIABLE POS.

SHAPE X,Y,N,"STRING" A UNIQUE GRAPHIC COMMAND THAT ALLOWS A COMPLEX SHAPE TO BE HELD IN A STRING THE IMPLICATIONS ARE MIND BOGGLING.

THESE ARE JUST A FEW OF THE NEW COMMANDS AVAILABLE. COMPLETE WITH INSTRUCTION MANUAL.

RST 28's THE FACTS John Hudson

The two program listings are assembly programs which help to explain more fully how to use the floating point routines. They show examples of how to use the RST28 calls both in SINGLE and MULTI function format. By using the two programs, and looking at the comments, every potential user should be able to understand how to interface to the ROM routines.

The first program is a circle program adapted from a Basic listing in MEMOPAD 4. The program first draws a circle, then draws the oval Memo circle, and finally draws a circle in Basic. Times are printed to show that the machine code routine is faster than basic, but not by much...this is because of the fact that the routine is 'calculation bound'. I will produce, at a later date, a very, very fast circle routine.

If the circle is not a circle on your display it is because of the way my monitor is adjusted....you should be able to work out how to correct this.

The second program is just a floating point calculator.

I make no claims about these programs, the only purpose they serve is to aid in the understanding of f.p math and how the RST 28 is used to aid the programmer. ★

Details of Memory Locations.

ACC	FDCB	The two locations most used are ACC1 and OP1. These hold the two numbers currently being operated upon.
ACC1	FDCC	
OP1	FDD1	
OP11	FDD2	
YORN	FDD7	
SIGN	FDD8	
COPY	FDDE-FDF1	

all the addresses are in hex.

```

1 CLOCK "000000"
10 CODE

4015 RST 10
4016 DB #4C; VS 4
4017 LD BC,360; Set count***Try setting to 720
401A LD (SAV),BC; Preserve count
401E LD BC,(SAV)
4022 CALL #1348; 360 - fp - ACC1
4025 RST 28
4026 DB #D4; PI to OP1
4027 DB #CA; Div ACC1/OP1
4028 DB #86; Swap ACC1 & OP1
4029 LD BC,2
402C CALL #1348; 2 to fp to ACC11
402F CALL #100B; ACC1 * OP1
4032 LD DE,MEM1
4035 LD HL,#FDD1
4038 CALL #12E7; (HL) to (DE)
403B CALL #1183; Swap ACC1 & OP1
403E LOOP: LD BC,(SAV); Recover count
4042 CALL #1348; Count - fp - ACC1
4045 CALL #1183; Swap ACC1 & OP1
4048 LD DE,MEM1
404B CALL #12A7; (DE) to ACC1
404E CALL #100B; ACC1 * OP1
4051 CALL #12BF; ACC1 to MEM1
4054 LD HL,SAV5
4057 CALL #12D4; MEM1 to SAV5
405A CALL #137D; Cos(ACC1) to ACC1
405D LD BC,70
4060 CALL #1183; Swap ACC1 & OP1
4063 CALL #1348; 30 - fp - ACC1
4066 CALL #100B; ACC1 * OP1
4069 CALL #1183; Swap ACC1 & OP1
406C LD BC,100
406F CALL #1348; 100 - fp - ACC1
4072 CALL #0EB6; Add OP1 to ACC1
4075 LD HL,#FDD1
4078 LD DE,#FE36
407B CALL #12E7; (HL) to (DE)
407E LD DE,SA5
4081 CALL #12A7; (DE) to ACC1
4084 CALL #1382; Sin(ACC1)
4087 CALL #1183; Swap ACC1 & OP1
408A LD BC,80
408D CALL #1348; 40 - fp - ACC1
4090 CALL #100B; ACC1 * OP1
4093 CALL #1183; Swap ACC1 & OP1
4096 LD BC,100
4099 CALL #1348; 100 - fp - ACC1
409C CALL #0EB6; Add ACC1 to OP1
409F LD DE,#FE3B
40A2 LD HL,#FDD1
40A5 CALL #12E7; (HL) to (DE)
40A8 CALL #167F; Plot
40AB LD DE,(SAV); Recover count
40AF DEC DE; Count -1
40B0 LD (SAV),DE; Preserve new count
40B4 LD A,D

```

```

40B5 OR E
40B6 JP NZ,LOOP
40B9 RET
40BA SAV: DS 8
40C2 SA5: DS 4
40C6 SAV5: DS 5
40CB MEM: DS 5
40D0 MEM1: DS 5
40D5 RET

```

Trig. Values

In order to use the above routines the following points must be understood :-

(1) to convert log(base E) to log (base b) $\text{Log}(x)/\text{Log}(b)$, ie log base 10 of x is $\text{log}(x)/\text{log}(10)$

(2) to convert sin,cos and tan from radians to degrees, $(x*PI)/180$ gives answer in degrees. eg $\sin(30*PI/180) = 0.5$

The following functions can be calculated by using the formulae given below:-

secant	sec(x)	= 1/cos(x)
cosecant	csc(x)	= 1/sin(x)
cotangent	cot(x)	= 1/tan(x)
inverse sine	arcsin(x)	= atn(x/sqr(-x*x+1))
inverse cosine	arccos(x)	= atn(x/sqr(-x*x+1))+1.5708
inverse secant	arcsec(x)	= atn(sqr(x*x-1))+ (sgn(x)-2)*1.5708
inverse cotangent	arccot(x)	= -atn(x)+1.5708
hyperbolic sine	sinh(x)	= (exp(x)-exp(-x))/2
hyperbolic cosine	cosh(x)	= (exp(x)+exp(-x))/2
hyperbolic tangent	tanh(x)	= -exp(-x)/(exp(x)+ exp(-x))*2+1
hyperbolic secant	sech(x)	= 2/(exp(x)+exp(-x))
hyperbolic cosecant	csch(x)	= 2/(exp(x)-exp(-x))
hyperbolic cotangent	coth(x)	= exp(-x)/(exp(x)- exp(-x))*2+1
inv.hyperbolic sine	argsine(x)	= ln(x+sqr(x*x+1))
inv.hyperbolic cosine	argcosh(x)	= ln(x+sqr(x*x-1))
inv.hyperbolic tangent	argtanh(x)	= ln((1+x)/(1-x))/2
inv.hyperbolic secant	argsech(x)	= ln((sqr(-x*x+1)+1)/x)
inv.hyperbolic cosec.	argcsch(x)	= ln((sgn(x)*sqr(x*x+1)+1)/x)
inv.hyperbolic cotan.	argcoth(x)	= ln((x+1)/(x-1))/2

```

11 PRINT TIME$
15 PAUSE 1000
5000 VS 4: CIRCLE 100,100,80
5010 PAUSE 1000
6000 VS 4: CLOCK "000000"
6020 FOR T=1 TO 360
6030 LET A=T*(2*(PI/360))
6040 LET X=100+68*COS(A)
6050 LET Y=100+78*SIN(A)
6055 PLOT X,Y
6060 NEXT
6070 PRINT TIME$
7999 GOTO 7999

```

1 PRINT "Floating point calculator Demo"
 2 PRINT "will do Add (+), subtract (-), multiply (*) and divide (/). To stop run press 'F8'"
 10 CODE

```

4087 START: CALL GETNUM
408A      CALL #12F3; put floating point number to OP1 (#fdd2)
408D GETOP: RST 10
408E      DB #8C,10,13,"Operation "
409B GETFN: CALL #79
409E      JR Z,GETFN
40A0      LD B,A
40A1      CALL #BC
40A4      LD A,B
40A5      CP "+"
40A7      JP Z,ADD
40AA      CP "-"
40AC      JP Z,MIN
40AF      CP "/"
40B1      JP Z,DIV
40B4      CP "*"
40B6      JP Z,MULT
40B9      RST 10
40BA      DB #8A," Illegal "
40C5      LD DE,#FFFF
40C8 DHL: DEC DE
40C9      LD A,D
40CA      OR E
40CB      JR NZ,DHL
40CD      XOR A
40CE      JR GETOP
40D0 BEG:  RST 10
40D1      DB #8C,10,10,13,"Result = "
40DE      CALL #20FD; convert fp. no to ascii dec no.
40E1      CALL PRINT
40E4      RST 10
40E5      DB #83,10,10,13
40E9      JP START
40EC      RET
40ED GETNUM: LD DE,#FB80
40F0      RST 10
40F1      DB #8F,10,13,"Enter Number "
4101 GET:  CALL #79
4104      JR Z,GET
4106      CP 135
4108      JR Z,NOMORE
410A      CP 13
410C      JP Z,GO
410F      LD (DE),A
4110      CALL #BC
4113      INC DE
4114      JR GET
4116 GO:   LD A,#FF
4118      LD (DE),A
4119      LD DE,#FB80
411C      CALL #224C; convert decimal number in ACC1 to floating point in ACC1
  
```



```

411F      RET
4120 PRINT: LD HL,#FDDE
4123 OUT:  LD A,(HL)
4124      CP 255
4126      RET Z
4127      CALL #BC
412A      INC HL
412B      JR OUT
412D      RET
412E ADD:  CALL GETNUM
4131      RST 28
4132      DB #88
4133      JP BEG
4136 MIN:  CALL GETNUM
4139      RST 28
413A      DB #90
413B      JP BEG
413E MULT: CALL GETNUM
4141      RST 28
4142      DB #88
4143      JP BEG
4146 DIV:  CALL GETNUM
4149      RST 28
414A      DB #8A
414B      JP BEG
414E      RET
414F NOMORE: POP HL
4150      RET
  
```

Symbols:
 GETNUM40EDSTART4087
 GETOP408DGETFN409B
 ADD412EMIN4136
 DIV4146MULT413E
 DHL40C8BEG40D0
 PRINT4120GET4101
 GO4116OUT4123
 NOMORE414F

Sale

The following software at 3.00 each;

TOADO, DRAUGHTS, CHESS, ALICE, MEMOSKETCH, P.H.PETE, HUNCHY, MUSIC PAD AND ICEBURG.

At 5.00 each:

ADVENTURE QUEST, DUNGEON ADVENTURE AND COMPOSER.

3 Books at 3.00 each:

THE MEMOTECH GAMES BOOK, MEMOTECH COMPUTING and one on how to learn MACHINE CODE (the one on offer in issue 5)

Please ring NOTTINGHAM 664290 or write to R.PEET,LILAC COTTAGE, MAIN STREET, HOVERINGHAM, NOTTS.

Sale

MTX 500. AS NEW CONDITION

PART EXCHANGE CONSIDERED

PHONE 01-243-6498 (HARROW)

TEVIC-DATA DENMARK

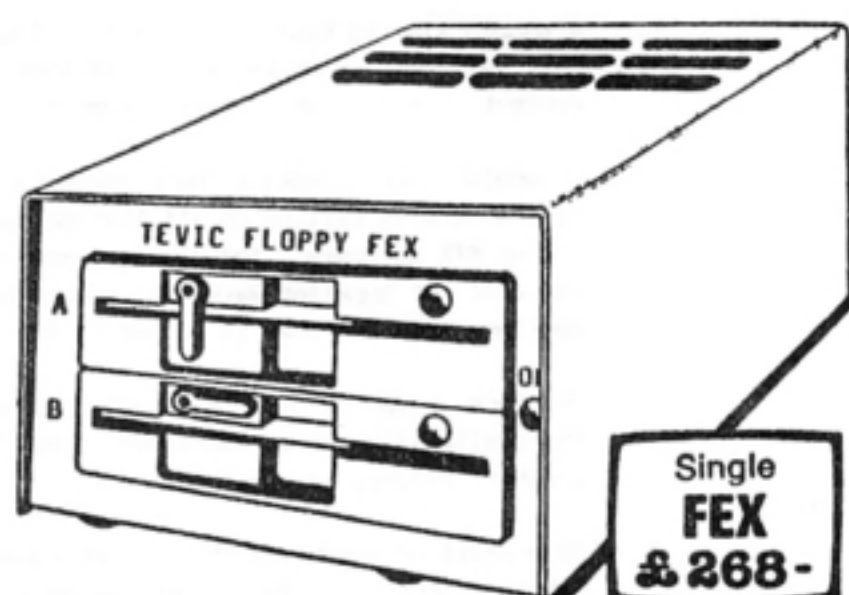
NEW PRODUCT



FOR THE
Memotech
MTX 512



FEX DRIVE Single or Double



Superb value-for-money!

Single drive, 500K DS/DD speed 250.000 bit pr./sec. or three times faster than Memotechs.

Complete system, including FEX Interface, Manual, Systemdisc with programmes. Bootmon, Backup, Stat Pool, Yatzy, Labyrint, Funktionkey, Dump, Convert from FDX to FEX disc, m.m. Cap. Format 352K pr. disks. Commando DLOAD, DSAVE, OPEN, DINPUT, DPRINT CLOSE, REN, ERA can be written directly or with user in front, or written in basic with line.

FEXDOS CP/M CALL

LD DE,FCB : FILE CONTROL BLOC, 36 byte

LD DE,DMA : DIRECTLY MEMORY ADRSS.

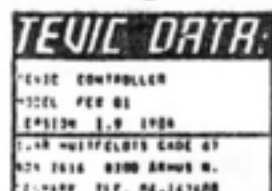
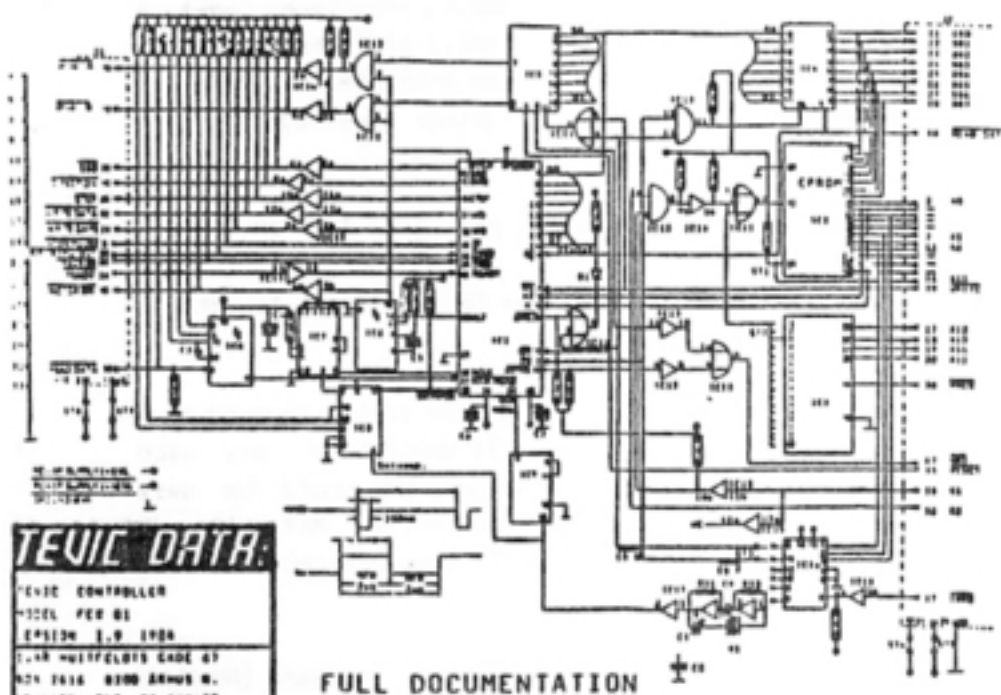
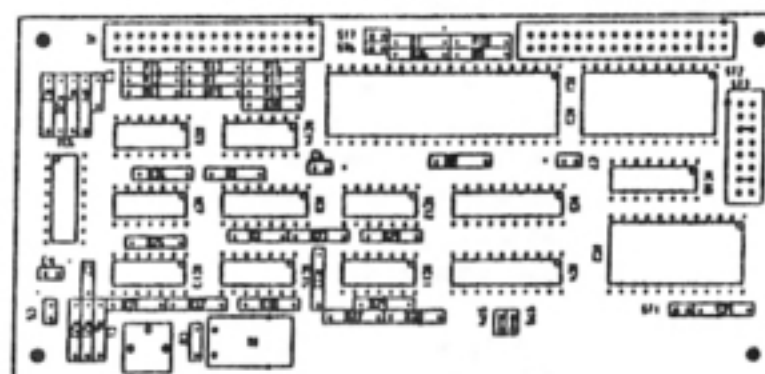
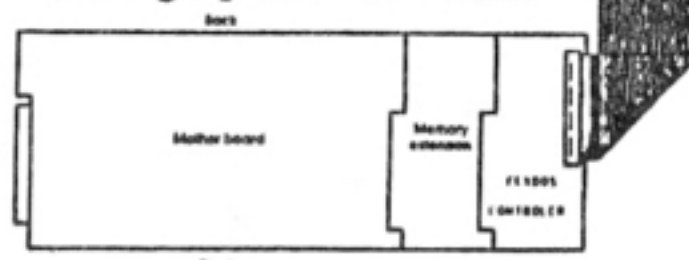
LD C,ORDRE: 15=OPEN FILE, 16=CLOSE FILE, 19=DELETE FILE

20=READ SEQ, 21=WRITE SEQ, 22=MAKE FILE

23=RENAME F., 26=SET DMA, 35=COMPUTE SIZE

CALL FEXDOS: E800

Setting up the Hardware



FULL DOCUMENTATION

MEMOTECH SOFTWARE

MTXLOGO	512	Tape	£ 7,50
DART	512	Tape	5,00
TAPE COPY	ALL	Tape	12,50
YATZY	ALL	Tape	5,00
VIDEO FILE	ALL	Tape	12,50
PRINTER DUMP	ALL	Tape	5,00
FUNCTIONKEY	ALL	Tape	5,00
POOL COUPON	ALL	Tape	12,50
BRUNWORD HELP to FEX Disc			14,00
MEMOSKETCH H. to FEX Disc			12,50
DISC CONVERT FDX to FEX			14,50
PASCAL HELP to FEX Disc			12,50
VIDEO FILE to FEX Disc			18,50

Prices

All prices is include
VAT but exclusive P/P
ADD £10,- for FEX and
£2,- pr. software.

GUARANTEED 14 DAYS DELIVERY FROM RECEIPT OF ORDER



I enclose a cheque for £ _____

Name _____

Company (if any) _____

Address _____

EXPORT PRICES ONLY GREAT BRITAIN

TEVIC-DATA **SOFT - HARDWARE CENTER**
IVAR HUITFELDTSGADE 67, 8200 AARHUS, N
POSTBOX 2616 · TLF. (06) 16 76 00 DENMARK

V
i
e
wp
o
i
n
t

Dear Keith,

I should like to begin by congratulating you on your achievements in the production of Memopad over the last year. It is quite remarkable that you should have attained such a standard and such diversity of content in such a short time, especially without the benefit of a great deal of advertising revenue.

I would like to make a few comments on the contents of your recent editorials, if I may. There is little use in telling us all that software houses make profits on Amstrad or Enterprise software but not on MTX software. The simple reason for this is that both of these companies have outsold Memotech by dint of superior marketing techniques. The market available for software is the total number of machines sold and that is an end to it.

Memotech have let down an excellent piece of hardware by poor marketing and insufficient attention to the availability of software and, given the present and foreseeable state of the market, it is doubtful whether recovery is possible.

As a previous purchaser of Memotech equipments, (add-ons for ZX81), I was well aware of the quality of their products and when canvassed by letter, I bought one of the first of the MTX in December 1983. I was particularly interested in educational software and the Company's promises at that time seemed to be everything I was looking for. However, when it proved to be impossible to convert the BBC software, it appears that the company abandoned their educational role completely, producing, through Continental, a few pathetic offerings which could easily and more conveniently be replaced at a fraction of the cost by a set of revision notes. Incidentally, if Memotech demonstrated either Maths 1, Helli-Maths or Spelli-Copter to the Russians as examples of available software, I'm not too surprised that they lost the order.

As a teaching aid, the home micro has great potential but it must of course be allied to properly written software. In the teaching of maths, for example, it offers the following advantages over a human teacher. It never loses patience with the pupil. It can present a virtually limitless number of randomly generated problems and check if they are correct. If not then it can evaluate the type of error and explain the underlying principles using diagrams and animation. A further advantage for many pupils is that they do not have to "make a fool of themselves" in front of the class.

However, there is, to be fair, not a great deal of good educational material out there for any micro and this is, I feel, an area into which Memotech might profitably move. It would, however, require investment and a sincere commitment to produce the best software and not just be an attempt to make a fast buck or to get any kind of junk into the shops as quickly as possible.

It seems to me that such a course of action would be a better use of resources than trying to produce a Spectrum Rom Emulator. For such a device (if it ever comes to pass), while it would be a very nice achievement for the boffins who could have a great time congratulating one another, and would be very nice for us present MTX owners, it is unlikely to sell a single computer. Why buy an MTX plus an emulator plus the necessary software to run Spectrum programmes (at 80 % speed?) when one could simply buy a Spectrum?

All is not criticism. The hardware is great. If it were only for its use as a word processor (NewWord Rom + DMX 80), I am happy with my purchase.

Once again congratulations to all involved with the Club and with Memopad. I am not at all sure that the Company deserves people like you.

Best Regards
Thomas C.Duggan

Dear Genpat,

I have recently discovered - by accident - that the Editor of the MTX/Hisoft PASCAL compiler uses a token system for RESERVED words similar to the BASIC token system. As a result of this the Function keys (provided that they have not been previously redefined) can provide one-key entry of a number of RESERVED words.

Although this facility is probably not as useful in PASCAL as in BASIC, members may be interested in the following lists:

KEY	WORD
1	--
2	PROGRAM
3	DIV

4	CONST
5	PROCEDURE
6	FUNCTION
7	NOT
8	OR
~1	AND
~2	MOD
~3	VAR
~4	OF
~5	TO
~6	DOWNT0
~7	THEN
~8	UNTIL

Unfortunately, Write, WriteLn, Read, & ReadLn are not RESERVED and are not represented! Spaces have to be inserted before and after the function keystrokes.

BRIAN HOUGHTON

Here are some comments from S.March of Glasgow
Dear Keith,

I just couldn't believe my eyes when I read the Birthday edition of Memopad. How stupid can Memotech be in not attending the PCW Computer Show. I am beginning to wonder if Mr. Fateh, the Boss, has any faith left in his brainchild. (Methinks he is about to "cut and run"!) How can anyone expect to get sales without advertising its product? We were told a few months back that Memotech were now convinced that they had a winner and were going to advertise again. We were told "Watch the Computer Magazines around July". Well I watched them in July, August and September and, with the exception of your own advert, have seen nothing about Memotech. Now we are told that they were not even attending the PCW Show! Their actions will certainly kill off whatever hope we had of the Memotech becoming a leading machine. So Happy Birthday! Enjoy it while you can 'cos you ain't got long to go!

I would suggest that Mr. Fateh, or whoever it was who made such a stupid decision, should apply to Japan for a job....as a KAMAKAZI PILOT!!

The following comments are from Paul Wood of Worcester.

Dear Keith,

Having recently read the latest edition of MEMOPAD, I wholeheartedly agree with your editorial, so much so, I would like to suggest the following; in the next available edition of MEMOPAD include a notice of intent to conduct a survey, asking people to list the hardware and software they already own and would like to see. (software could be marked on the latest list from MEMOPAD), a 5 star rating could be used for the software and any general comments would be noted, all the above information would not be linked to specific people, as a software writer and obviously an interested party I would be prepared to correlate all the information and forward it to you for your's, Memotech's and any other parties interest, (you may wish to include a census type section including age group's, occupations etc.all of which I would be happy to handle.

ANDREW BISHOP has sent in this letter:

In the June issue of Memopad you asked about what sort of software people would buy, well here is a list of software and hardware I would buy.

For games software I would have a flight and car racing simulator, a Daley Thompson Decathlon type of game and like Mr. Webb in July's issue I would like a business simulator. The first two must be better than Turbo, not so much in the graphics but in the thought that goes into the program, in a real race you can spin-off and then rejoin, in Turbo you can't; also in a real race on most corners you can't do 240m.p.h but in Turbo you can. It is things like this which separate the good from the rubbish.

For business software I would have a spreadsheet program of reasonable size which can produce a hardcopy of some sort and a program to produce charts on a printer as well as the screen. Both of them should be on tape as not all of us have disc drives, maybe data from the programs could be saved on tape or disc so when you up grade your system you don't lose the data.

As for educational and utilities programs I would like to see parts 2,3 and 4 of Physics released and an extension of Memosketch to include "freehand" drawing. On hardware I would buy a graph pad and one day get an FDX disc drive.

STARTING FORTH *Keith Jones*

This month I'll give some routines which have been sent by GENPAT members. But first of all I want to reply to a letter which has been sent by Dr. B. Houghton.

I feel that the points he has raised are important enough that all who read these articles should see the reply. He raises four points, and I shall give these points first then my replies;

"1. I cannot find an ASSEMBLER vocabulary."

"2. Any attempt to use the CRVS command crashes the compiler."

"3. When this happens, the message VS 5:PRINT USR (16641): REM *COLD START* appears: any attempt to execute this crashes into the front panel."

"4. Some of the reserved words are slightly non-standard, particularly disk I/O and also NUMBER (whose syntax took me 3 weeks to unravel: it aborts if it finds an ASCII null in the input stream)."

The replies are as follows ;

1. The reason that you cannot find ASSEMBLER is because there isn't one. This is because the fig standard specifies that provision should be made for an assembler, but the use of assembled words should not be encouraged. Why have a compiler if words are going to be written in assembly ? Surely this defeats the object. I realise that some portions may have to be written in assembler but these should be few and very small. This is why "CREATE" and "C" are included.

Also the reason I have not included an assembler is because I haven't the time. As well as writing these articles, working on routines to include in them, writing a book on fig FORTH, replying to letters and writing other programs I also have a full time job where I work a ten hour day and I help friends who have problems with their own computers. These range from Z80 based to 8086 based machines. I also have to try and hold down a normal life with my better half. All I can say is thank goodness that Vaughan understands me and puts up with me like she does. As an example of how little time I have to spare, a friend recently arranged a contract for me to work on a new "Spectrum" program. The returns would have been high and I would have then broke into a market which is really the one to be in. I had to turn this contract down and the gentleman I spoke to must have thought that I was mad writing for a "small" machine like the "Memotech". The sooner Memotech stop treating the MTX as a hobby and see it as business the better.

I'm sorry that I've turned this into a sermon so I'll move on.

2. It would seem that you have not received the errata sheet which should be in the back cover of your manual. If anybody has not got this then if you write to GENPAT I'm sure that they'll supply it to you. The CRVS syntax was changed by me after the manual was finished. Therefore it is my fault that the manual is wrong and I accept all responsibility for this. Could I also add that any problems with the manual must be addressed to GENPAT.

3. You probably altered a portion of the compiler when you used the CRVS command. Hence you cannot re-enter.

4. Here we enter the thorny subject of standards. The fig implementation which you have is the most recent (17th September 1982) standard to come from the Forth interest group. As a member of FIG-UK I can say this with all confidence. Looking at issue 11 of "Memopad" I notice that the books which you recommend are all 79 standard, with fig given as an after thought. In fact as I read through the books I can see that in general none of the authors have conformed to fig standard.

Why everybody seems to concentrate on 79 is beyond me, after all the majority of home micro's now have a fig-FORTH running on them viz "White Lightning". The book which I mentioned earlier is purely fig-FORTH. Further to this, the only non-standard I/O word is "R/W" and this is because we are using RAM-DISC

I think that's all I've got to say so I'll shut up now and present routines which you've sent to me, with instructions for their use.

First of all Julian Barkway has sent a re-definition of EXPECT. This is the word which reads the keyboard and EMIT's the characters. Julian was unhappy with the fact that <BS> did not delete the character so he redefined it to do this. This is his definition with my inclusion to ensure that the COLD start parameters are reset. Otherwise if you executed COLD the system would crash.

SCR # 1

```

0 (RE-DEFINITION OF EXPECT WITH DESTRUCTIVE BS)
1 FORTH DEFINITIONS HEX
2 : EXPECT
3   OVER + OVER
4   DO KEY DUP OD =
5     IF LEAVE 0 I ! DROP BL
6     ELSE DUP 8 =
7       IF R> 2 - >R EMIT 16
8       ELSE DUP I C! 0 I 1+ THEN
9   THEN
10  EMIT
11  LOOP DROP ;
12  ' EXPECT CFA 4BF7 !
13  DECIMAL LATEST 12 +ORIGIN !
14  HERE 28 +ORIGIN ! HERE 30 +ORIGIN !
15  ' EDITOR 6 + 32 +ORIGIN ! HERE FENCE ! ;S

```

The routine consists of lines 1 to 11. Line 12 resets the compiler routine which uses EXPECT so that it now points to Julian's. Lines 13 to 15 reset the COLD start parameters so that if we execute COLD or WARM then the new EXPECT will not be erased. Should you want to use the original EXPECT then you should enter the following :

```

HEX 4B71 4B7F !

```

Yet more from Julian. I present these as they are written on the letter, as I feel that they are fully explained by the use of comments.

SCR # 2

```

0 : OK DUP 31 ; ( TEST FOR CONTROL CHARACTERS )
1 : CH? ( PRINT CHARACTERS STORED AT STACK LOCATION ON SCREEN )
2   C@ OK IF EMIT SPACE ELSE DROP . " _" THEN ;
3 : DUMP ( f n n' --- )
4   ( DUMP CONTENTS OF LOCATIONS n TO n' TO SCREEN IN HEX )
5   ( OR ASCII CHARACTERS DEPENDANT ON f ( 0 =HEX ) )
6   CR 1+ SWAP DO I OVER IF CH? ELSE C@ . THEN
7   LOOP CR DROP ;
8   !" ( STORE ASCII CHARACTERS AT STACK LOCATION )
9   22 WORD HERE 1+ C@ SWAP C! ;
10  etc.....

```

On now to three small but useful routines from Mark Adams. I should point out that I've been in touch with Mark on a number of occasions and it's really his letters which have kept my enthusiasm for FORTH going. The comments which have been added are my own, but the routines themselves are all Mark's

SCR # 3

```

0 : FREE ( DISPLAY NUMBER OF BYTES FREE TO THE SYSTEM )
1   CR U. ." BYTES " CR ."FROM"
2   CR HERE U. ."TOP OF DICTIONARY " CR ." TO "
3   CR SP@ U. ." BOTTOM OF STACK " ;
4
5   DECIMAL
6   : ?BASE ( DISPLAYS CURRENT NUMBER BASE )
7   BASE @ CASE
8     2 OF ." BINARY "      ENDOF
9     10 OF ." DECIMAL "    ENDOF
10    16 OF ."HEXADECIMAL " ENDOF
11  ENDCASE ;
12

```

```

13 : PEEK ( addr --- PRINTS VALUE OF ADDRESS )
14 DUP C@
15 CR . ." AT ADDRESS " U. ; ;S

```

Of course I can't just sit back and demonstrate how good you are without sticking something of my own here. A huge number of you (two) have written asking about extending RAM-DISC to give more screens. In order to do this I now give you the necessary values to insert. Please note that this is just a temporary measure as I'm working on some words to allow you to set up as many screens as you want, within reason. At present you've got 5 screens. To increase this number all you need do is take the value from the table given below and then type.

HEX nnnn 5990 ! (WHERE nnnn IS THE VALUE)

For n screens	SCR range	nnnn

six	0 to 5	F000
seven	0 to 6	F400
eight	0 to 7	F800

A greater number of screens would crash the Memotech so you'll have to make do with that for now.

Before I finish this article I want to admit that I'm a total idiot. I dropped a clanger in issue 11's article. SAVE-MEM and VERIFY-MEM do work but you'll always get a "mismatch" error from BASIC when you use VERIFY-MEM. The reason is that FORTH takes the most recently typed command and then moves to the byte straight after the end of the dictionary. It places a length byte at the first location. Hence when you SAVE-MEM the final byte saved has a value of 8 where as when you VERIFY-MEM the final byte is changed to 10. As these bytes don't match it doesn't verify. So in order to use the routines please correct the following in your copy of MEMOPAD

PAGE 8

```

HEX
HERE 4100 -
4094 !

```

Should read

```

HEX
HERE 1 - 4100 -
4094 !

```

PAGE 9

HERE 4100 - SAVE-MEM should be HERE 1 - 4100 - SAVE-MEM

HERE 4100 - VERIFY-MEM should be HERE 1 - 4100 - VERIFY-MEM

Once more I'm very sorry and I've eaten issue 11 to prove how sorry I am (it's called eating one's words!). See you next month. ★

SKETCH

Tony Ransley



```

10 REM Character Sketch.
15 REM
20 REM ANTHONY.A.RANSLEY 1984
30 REM
100 CRVS 2,0,22,18,2,1,40
110 CRVS 3,1,24,3,4,9,32
130 GENPAT 3,1,255,129,153,189,189,153,129,255
140 CTLSPR 1,7
150 CTLSPR 2,1
160 CTLSPR 6,0
300 CLEAR
310 LET Y=1
320 LET X=1
380 SOUND 0,1020,15
390 SOUND 1,1019.5,15
400 VS 5: CLS
405 PAPER 4
410 CSR 11,2: PRINT "CHARACTER SKETCH"
420 CSR 11,3: PRINT "-----"
430 CSR 0,5: PRINT "Use Joystick or Cursor keys to move the"
435 PRINT "cursor around the character grid."
440 PRINT "To fill or erase a square press HOME or"
445 PRINT "fire on Joystick.Press 'RET' for"
450 PRINT "character data.After all the character"
455 PRINT "data has been displayed press 'RET' to"
460 PRINT "restart program or 'ESC' to continue"
465 PRINT "work on the character."
470 PRINT " Character grid sizes are 1,4 & 9"
475 PRINT : PRINT "ANTHONY RANSLEY 1984"
525 CSR 0,18: PRINT "ENTER CHARACTER SIZE:"
530 VS 2
535 PAPER 4
536 PRINT CHR$(27);"P";
540 INPUT "";S
550 IF S=1 OR S=4 OR S=9 THEN GOTO 560 ELSE GOTO 530
560 VS 5: CLS
580 SOUND 1,0,0: SOUND 0,0,0
650 VS 4
660 COLOUR 4,4: COLOUR 2,4: COLOUR 0,4
670 CLS
680 IF S=4 THEN GOTO 1400
690 IF S=9 THEN GOTO 1200
700 GOTO 1000
710 REM*****WAVY LINE OUTPUT*****
730 ANGLE 0
750 ARC 10,1.85
760 DRAW 14
770 ARC 10,-1.85
775 ANGLE 9.425
790 ARC 10,-1.85
795 DRAW 14
800 ARC 10,1.85
810 RETURN
815 REM*****
820 ANGLE 14.135
830 ARC 10,1.85
840 DRAW 14
850 ARC 10,-1.85
860 ANGLE 4.72
870 ARC 10,-1.85
880 DRAW 14
890 ARC 10,1.85
900 RETURN
1000 REM*****GRID ONE*****
1010 FOR C=56 TO 112 STEP 7
1020 SOUND 0,250,15
1030 LINE 56,C,112,C
1040 SOUND 0,400,15
1050 LINE C,56,C,112
1060 SOUND 0,0,0
1070 NEXT C
1080 CSR 5,8: PRINT "CHARACTER "A"
1150 LET YB=1: LET YT=8
1160 LET XB=1: LET XT=8
1170 SPRITE 1,1,109,59,0,0,10
1180 DIM D$(8,8)
1190 GOTO 2000

1200 REM*****GRID NINE*****
1210 FOR C=0 TO 168 STEP 7
1220 SOUND 0,100,15
1230 LINE 0,C,168,C
1240 SOUND 0,200,15
1250 LINE C,0,C,168
1260 SOUND 0,0,0
1270 NEXT C
1280 LET YT=24: LET YB=1
1290 LET XL=24: LET XR=1
1300 SPRITE 1,1,165,3,0,0,10
1310 DIM D$(24,24)
1320 FOR C=1 TO 168 STEP 56
1330 PLOT 170,C
1340 GOSUB 710
1350 NEXT C
1352 CSR 23,6: PRINT "C"
1354 CSR 23,13: PRINT "B"
1356 CSR 23,20: PRINT "A"
1360 FOR C=167 TO 1 STEP -56
1365 PLOT C,172
1370 GOSUB 815
1375 NEXT C
1380 CSR 17,0: PRINT "1"
1385 CSR 10,0: PRINT "2"
1390 CSR 3,0: PRINT "3"
1395 GOTO 2000
1400 REM*****GRID FOUR*****
1410 FOR C=140 TO 27 STEP -7
1420 SOUND 0,700,15
1430 LINE C,140,C,28
1440 SOUND 0,500,15
1450 LINE 140,C,28,C
1460 SOUND 0,0,0
1470 NEXT C
1480 LET YT=16: LET YB=1
1490 LET XL=16: LET XR=1
1500 SPRITE 1,1,137,31,0,0,10
1510 DIM D$(16,16)
1540 FOR C=29 TO 140 STEP 56
1550 PLOT 142,C
1560 GOSUB 710
1570 NEXT C
1580 CSR 19,10: PRINT "B"
1585 CSR 19,17: PRINT "A"
1590 FOR C=139 TO 29 STEP -56
1600 PLOT C,144
1610 GOSUB 815
1620 NEXT C
1630 CSR 7,4: PRINT "2" 1"
1980 SOUND 1,0,0
1990 SOUND 0,0,0
1995 REM*****KEY CONTROL*****
2000 IF INKEY$<>CHR$(11) THEN GOTO 2100
2020 IF Y=YT THEN GOTO 2100
2030 SOUND 0,350,15
2040 LET Y=Y+1
2060 MVSPR 1,1,6
2070 SOUND 1,348,15
2100 IF INKEY$<>CHR$(10) THEN GOTO 2200
2120 IF Y=YB THEN GOTO 2200
2130 SOUND 0,350,15
2140 LET Y=Y-1
2160 MVSPR 1,1,2
2170 SOUND 1,348,15
2200 IF INKEY$<>CHR$(8) THEN GOTO 2300
2220 IF X=XL THEN GOTO 2300
2230 SOUND 0,350,15
2240 LET X=X+1
2260 MVSPR 1,1,4
2270 SOUND 1,348,15
2300 IF INKEY$<>CHR$(25) THEN GOTO 2700
2320 IF X=XR THEN GOTO 2700
2330 SOUND 0,350,15
2340 LET X=X-1
2360 MVSPR 1,1,0
2370 SOUND 1,348,15
2700 PAUSE 50

2710 IF INKEY$=CHR$(26) THEN GOSUB 2890
2730 IF INKEY$=CHR$(13) THEN GOTO 3200
2800 SOUND 0,0,0
2810 SOUND 1,0,0
2820 GOTO 2000
2890 IF D$(Y,X)<>"1" THEN GOTO 2960
2900 REM*****BLANK OFF A SQUARE*****
2910 LET D$(Y,X)="0"
2920 ATTR 2,1
2925 PLOT 220-X,50+Y
2930 GOTO 3000
2950 REM*****FILL IN A SQUARE*****
2960 LET D$(Y,X)="1"
2970 ATTR 2,0
2980 PLOT 220-X,50+Y
3000 REM*****FILL & ERASE SQUARE*****
3010 IF S<>1 THEN GOTO 3040
3020 LET XX=112-(X*7)
3030 LET YY=(49+(Y*7))+1
3040 IF S<>4 THEN GOTO 3070
3050 LET XX=140-(X*7)
3060 LET YY=(21+(Y*7))+1
3070 IF S<>9 THEN GOTO 3100
3080 LET XX=168-(X*7)
3090 LET YY=(-7+(Y*7))+1
3100 FOR C=1 TO 6
3110 SOUND 0,(500)+50*C,15
3120 LINE XX+C,YY,XX+C,YY+5
3150 NEXT C
3160 SOUND 0,0,0
3180 RETURN
3200 REM*****DATA OUTPUT*****
3205 ATTR 2,0
3210 IF S=1 THEN LET Z=1 ELSE LET Z=3
3220 IF S=4 THEN LET Z=2
3240 CSR 19,0: PRINT "CHARACTER:"
3250 FOR A=1 TO Z
3260 CSR 30,0: PRINT CHR$(64+A)
3270 FOR B=1 TO Z
3280 CSR 28,0: PRINT B
3285 VS 3: CLS
3290 COLOUR 4,4
3295 FOR C=(8*A) TO ((8*A)-7) STEP -1
3300 LET T=0: LET Q=128
3310 FOR D=(8*B) TO ((8*B)-7) STEP -1
3330 IF D$(C,D)="1" THEN LET T=T+Q
3340 LET Q=Q/2
3350 NEXT D
3360 PRINT T
3370 NEXT C
3385 VS 4

3390 IF INKEY$="" THEN GOTO 3390 ELSE NEXT B
3400 NEXT A
3410 PAUSE 500
3420 FOR C=2 TO 15
3430 COLOUR 1,C
3440 PAUSE 50
3450 CSR 25,21: PRINT "RET": CSR 27,22: PRINT "or"
3455 CSR 25,23: PRINT "ESC"
3460 IF INKEY$=CHR$(13) THEN GOTO 3490
3465 IF INKEY$=CHR$(27) THEN COLOUR 1,15: GOTO 2000
3470 NEXT C
3480 GOTO 3420
3490 COLOUR 1,15
3500 CLS
3510 SPRITE 1,1,-4,-4,0,0,10
3520 GOTO 300

```

HARDWARE



PRICES EFFECTIVE FROM 1ST OCTOBER 1985

MTX 500 £79.95 INCLUDING MEMBERSHIP
 MTX 512 £129.00 INCLUDING MEMBERSHIP

32K MEMORY EXPANSION £36.74
 64K MEMORY EXPANSION £45.43
 NEWWORD ROM £36.74
 PASCAL ROM £36.74
 SPECULATOR £36.95

SDX 500K INTERFACE £222.50
 SDX 1MB INTERFACE £265.83

2ND 250K DRIVE £89.00
 2ND 500K DRIVE £169.00
 2ND 1MB DRIVE £203.00

80 COLUMN PCB:CPM:NW + SC £186.00
 80 COLUMN RS232 PCB KIT £22.00

SDX 500K + INTERFACE, 80 COL BOARD
 CPM, NW & SC £355.00
 1MB VERSION OF ABOVE £455.00

SDX 500K PACKAGE AS ABOVE PLUS
 MTX512 COMPUTER £455.00

FDX SINGLE 500K DRIVE +CPM £539.00
 FDX SINGLE 1MB DRIVE + CPM £675.00

TWIN FDX 500K CP/M SYSTEM £569.00
 TWIN FDX 1MB CP/M SYSTEM £740.00

500K SILICON DISC £145.00
 1MB SILICON DISC £441.00

FDX 2ND DRIVE 500K £141.00
 FDX 2ND DRIVE 1MB £163.00

DUST COVER £3.50
 CENTRONICS 2 metre CABLE £9.95
 DMX PRINTER RIBBON £8.75

FLOPPY DISCS (BOX 10) TOP QUALITY GUARANTEED £18.75

DMX 80 PRINTER & CABLE £239.00
 RITEMAN DOT MATRIX PRINTER £215.00

PACE NIGHTINGALE MODEM £119.00 + £5 postage & packing.

250K SDX DISC DRIVE £199.00 + £5 postage & packing

DISC CASES HOLDS 10 FLOPPIES £2.50
 ANTISTATIC SCREEN WIPES(10)..... £1.39
 FLOPPICLENE DISC KIT£16.50
 CRIB CARD£2.16

UPGRADE PACKAGE 1 £198.00
 UPGRADE PACKAGE 2 £223.39

The above require factory fitting so add an extra £25 to cover cost of this service.

DON'T FORGET IF YOU HAVE A DISC DRIVE YOU SHOULD OWN A HIGH QUALITY HEAD CLEANER see FLOPPICLENE
 over half the problems handled by us are due to dirty disc heads.

JARO SPEED SPLITTER WITH SOFTWARE £20.70



THE ABOVE PRICES ONLY APPLY TO U.K SALES & BFPO SALES
 BFPO SHOULD ADD AN EXTRA £30.00 TO COVER ADMINISTRATION BY MEMOTECH



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 CHAMBERIDS 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 SCRP 2.95 E ANY
 00110 CRYSTAL ARC MEGA 5.95 E ANY
 00050 DEN.GOES BANANAS ARC SCRP 2.95 I ANY
 00011 DENNIS & CHICKEN ARC SCRP 2.95 I ANY
 00103 DENNIS AND CIRCUS ARC SCRP 2.95 I ANY
 00068 DODDLEBUG ARC SYNT 4.95 I ANY
 00108 DOWNSTREAM DANGER ARC MEGA 5.95 E ANY
 00096 DR. FRANKIE ARC SYNT 5.95 I 512
 00056 DRAUGHTS BRD CONT 6.95 I ANY
 00111 DRIVE THE CEE 5 ARC MEGA 5.95 E 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 GOLOMINE 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
 00115 KARATE KING ARC MEGA 5.95 E ANY
 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
 00112 OBLITERATION ZONE ARC MEGA 5.95 E 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
 00109 QUAZZIA ARC MEGA 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
 00114 ROLLA BEARING ARC MEGA 5.95 E 512
 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
 00113 SEPULCRI SCELERATI ARC MEGA 5.95 E 512
 00101 SLOOPY'S CHRISTMAS ARC PANS 2.95 I ANY
 00116 SMI; ARC MEGA 5.95 E 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
 00118 THE DESIGNER UTIL HALT 8.95 E 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
 00117 USER BASIC UTIL SYNT 8.95 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



PLEASE ONLY ORDER THOSE MARKED " I "

KEY:- STOCK NUMBER TITLE TYPE HOUSE MEMBER PRICE STOCK

MACHINE

E=EXPECTED SOON

I = IN STOCK

U = UNAVAILABLE AT PRESENT

SUBSCRIPTIONS

IF YOUR MEMBERSHIP NUMBER IS BETWEEN 0 - 848 INCLUSIVE (ALL LETTERS A - I) YOU ARE NOW DUE TO RENEW YOUR SUBSCRIPTION. TO MAKE SURE OF RECEIVING YOUR NEXT COPY OF **MEMOPAD** PLEASE SEND YOUR SUBSCRIPTION TO REACH US NO LATER THAN THE 30TH OCTOBER. AN APPLICATION FORM IS INCLUDED ON THE ORDER SHEET.

SPECTRUM SPECULATOR

THE SPECTRUM EMULATOR WILL BE READY WITHIN 3 WEEKS OF RECEIVING YOUR MAGAZINE. IT IS CALLED THE "SPECULATOR". MEMBERSHIP PRICE IS £36.95 + 50p p.p and includes the first software tape. SOME OF THE PROGRAMS THAT YOU WILL BE ABLE TO RUN ARE: ARCADIAN ; ASTRONAUGHT; ATTIC ATTACK; DECATHLON; FLIGHT SIMULATION; GRID RUNNER. WE ARE AT THE MOMENT NEGOTIATING TO HAVE THESE GAMES IN STOCK SO THAT WE CAN OFFER THEM TO YOU AT A DISCOUNT. ORDERS WILL BE DESPATCHED IN STRICT ROTATION. A SEPARATE SHEET IS INCLUDED FOR THE SPECIFIC PURPOSE OF ORDERING THIS. IF YOU WANT TO LEAVE A DEPOSIT WE WILL PLACE YOU ON THE LIST AND NOTIFY YOU WHEN AVAILABLE. WE HAVE PLAYED WITH A VERSION HERE THAT HAS JUST ARRIVED AND WE WILL DO A FULL REVIEW IN THE NEXT EDITION.



250K DISK DRIVE

DISC DRIVE RAFFLE WIN A 250K DISC DRIVE FOR £1.00. THIS TIME YOU DON'T NEED TO SEND A POUND (THEY ARE ALMOST EXTINCT). SIMPLY WRITE YOUR NAME, ADDRESS, MEMBERSHIP NUMBER ON A PIECE OF PAPER AND INCLUDE THE £1.00 IN A CHEQUE ETC . WILL BE DRAWN 20TH NOVEMBER SO YOU HAVE PLENTY OF TIME. IF YOU HAVE A DRIVE YOU CAN CHOOSE GOODS TO THE VALUE.

Tape of the Mag

40 Column Graphics

Bertie Bug

Speedy 3D

Air Sea Rescue

A pleasant surprise for all 512 owners.

2.95