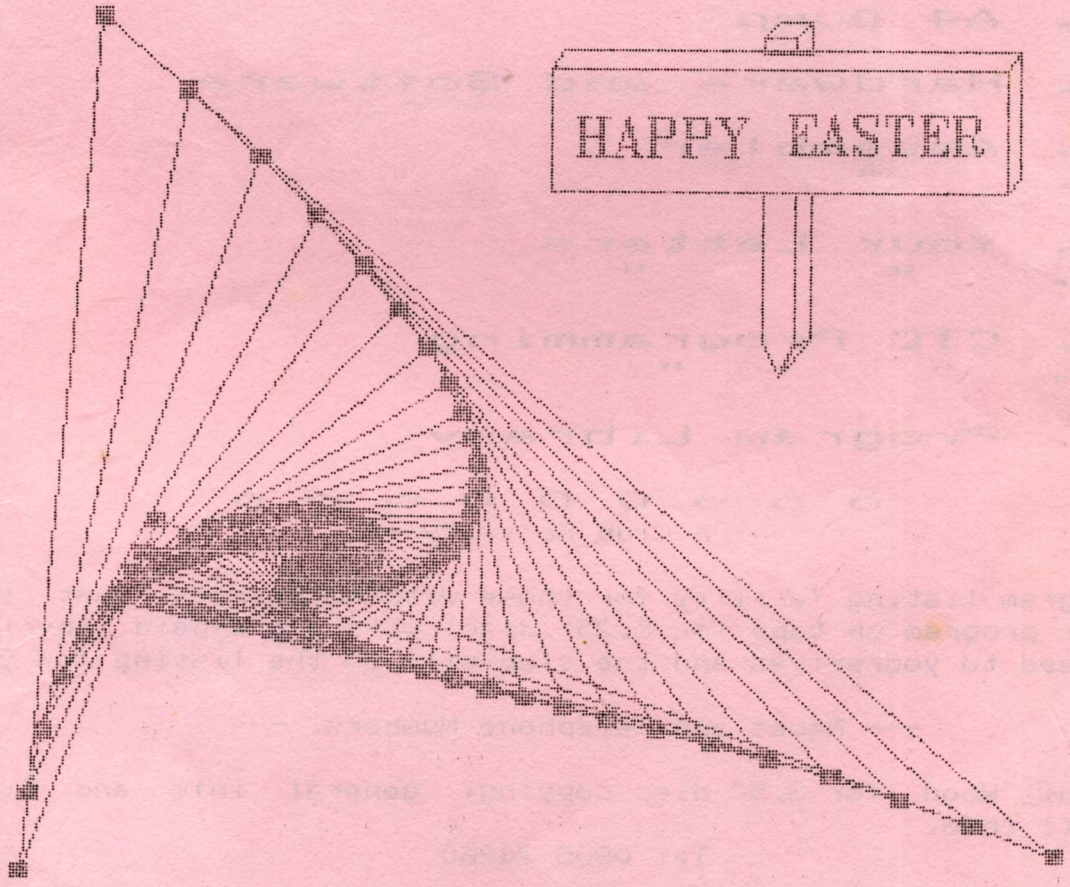
VOL. 3 ISSUE 6 APRIL 1987

THE MEMOTECH OWNERS CLUB MAGAZINE

MEMOTECHNIQUES



PUBLISHED BY: MEMOTECH OWNERS CLUB
23 DENMEAD ROAD
HAREFIELD
SOUTHAMPTON

CIRCA ...257

M.O.C.

VOLUME 3 ISSUE NUMBER 6

CONTENTS

o o o O O O o o o

--- A Club Facility ---

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

--- Names and Telephone Numbers. ---

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

Tel 0905 24260

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

If anyone has any good graphics designs for a front cover then we would love to see them!!!

E D I T O R I A L (April 1987)

Phil Eyres
23 Denmead Road
Harefield
Southampton
SO2 5GS

I have been in phone contact with Memotech Computers Limited a couple of days ago, to enquire about the new series 2 machine. Several members have phoned suggesting incompatabilities between old and new machines, which I must say surprised me a little. So far as I can tell, without actually having 'hands on' experience of the new machine, the basic machine is identical to the old machine, except that it has 256K chips fitted, giving the stated extra memory. The new 3.5" disc drives which were infact available on the old machine and work perfectly well, are now the standard disc upgrade for the Memotech, the older 5.25" format being dropped, due to lack of demand. The new machine with disc drives, running CP/M 2.2 will infact only access 64K of memory when running under CP/M, the extra memory only being available for Basic. I believe the extra memory is available under CP/M Plus.

As many have asked for my views on the new machine, they are as follows:-

1. The new MTX series 2 computer unit is not significantly different from the old MTX and thus does not offer any real reason to upgrade unless you have problems running out of memory when running under Basic. It is, I feel aimed at the user who is new to the MTX scene.

2. The new 3.5" disc drives are well worth the money, they have I believe proved to be both reliable and fast. Also, when working with the silicon, the whole system works together to form a first rate usable computer, with all the speed you could ask for.

Clive Taylor has managed to come to an agreement with MCL, in that they will allow the factory to be used for the venue of the next MTX national meeting. The date is still yet to be agreed, but it will hopefully be in the nice warm season (Summer?). Details of exact date and time will be forwarded as soon as they are available.

I have to extend the warmest thanks to everyone who has contributed to the magazine in the way of articles in the past years, as without them, it would have been impossible to produce anything on time. I will however ask if anyone is in a position do so, could they put something together for the next magazine, as time on my part will be short, as I hope, in the coming weeks to be moving house. The effort needed just to change the address on all the files will be great, let alone to keep everything going. I hope to be able to put forward to you a new address by the next magazine, all things being well.

I can read/write only disc's in 5.25" format and up to 500K, if anyone with 3.5" systems would like something from the club or has something to offer on 3.5" format please send to Paul Wood, his address is listed opposite.

Our club software, notably being SMGII and REVEAL is now available, both programs are only available for the MTX 512, I hope that does not prove too restrictive and sorry to 500 owners for having to miss out on these two fantastic programs. If you own a 500 why not go for a copy of our PUC-MAN, it has all the usual PUC-MAN type features and is a bargain at only £5.00.

As I am more often than not out playing squash on Mondays between 6 and 7 pm, I feel it would be best to move the Hotline to after 7.40pm. I hope this is ok for everyone. The number to phone now is (0703) 466106, ask for Phil. However, feel free to phone any evening after 6pm, if I'm not in then my Mum (good old Mum!!) will take any calls.

If anyone would like back issues they are available for the small remittance of 80p each. At present there are 25 back issues, 10 for volume 1, 10 for volume 2 and 5 for volume 3.

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

ooo 0-0-0 ooo

Software

Software prices for the best and most popular software:-

Zarkos	£7.00	Chamberoids	£7.00
Qogo2	£7.00	26*26 SpreadSht	£8.50
Karate King	£7.00	Son Of Pete	£7.00
S.M.G	£7.00	T.Snooker	£8.00512Only
Doodlebugs	£6.00	Super Bike	£6.00
J.J.Flash	£6.00	Ed/Asm	£8.50
Cee-5	£7.00	MTX Asm Lang Cse	£10.00
Highway Encounter	£8.50		

POLAR COORDINATES PATTERNS

By P.Burns

The outline of this program comes from an article on Micromaths by Keith Devlin which appeared in the Guardian some 14 months ago. I have adapted it for the MTX and have REM'ed various options to try out. A printer is not necessary as the patterns are fascinating in themselves as they are plotted on the screen. It's all to do with Polar coordinates - whatever they are! Personally I like pretty patterns that are adjustable.

The two parameter variables A and B are called for at each start - enter integers within the spread -9 to 9.

Changes [To try - remove REM as necessary]
Line 190 - contains variable C which you alter to change the main formulae at lines 230 & 240.
I suggest you try :-
C=1+SIN(2*D) OR C=1+COS(D) OR C=1+2*(COS(D) ETC

Line 255 - gives a symmetrical Butterfly pattern rotating around the screen centre.
Try A=1: B=5: C=SIN(7*D) for starters

Line 260 - gives a series of points in a convoluted "big dipper " pattern. Looks good on the screen but not up to much when printed out in black and white. Again - changing A B or C either singly or in combination gives a new pattern.

Lines 263 and 265 - I prefer this to the line 250 option. Used together you get a banded "big dipper" pattern which prints in a clearer way too.

Line 280 - gives access to printing sub routine.

Line 290 - For owners of utility USER EXTEND

Lines 300 - 310 print out the parameters used as a remainder if you should wish to repeat the pattern - delete them if they annoy you.

Lines 370 - 450 Printing sub routine. Set up for a DMX printer - adjust the printer commands to suit your machine. The sub routine is worth SAVEing by itself for use as an "add-on" to other programs.

```
10 REM ********************************
20 REM ** MATHEMATICAL PATTERNS     **
30 REM **                           **
40 REM ** ACKNOWLEDGEMENTS TO       **
50 REM ** KEITH DEVLIN  MICROMATHS  **
60 REM ** GUARDIAN 30.1.86          **
70 REM **                           **
80 REM ** ADAPTED FOR THE MTX BY    **
90 REM ** P.BURNS                   **
100 REM ********************************
110 VS 4: CLS
120 INPUT " ENTER A AS INTEGER -9 TO 9";A
130 INPUT " ENTER B AS INTEGER -9 TO 9";B
140 CLS
150 LET SX=250: LET SY=190: REM Set screen limits - both
must be even numbers
160 LET HX=SX/2: LET HY=SY/2: REM centre screen locations
180 FOR D=0 TO 6.2 STEP 0.01
190 LET C=SIN(7*D): REM [see comments at CHANGES above]
200 LET E=ABS(C)
210 LET F=1.0E-30
220 IF F<E THEN  LET F=E+0.1
230 LET G=INT(HX+HX*COS(A*D)*E/F)
240 IF G<0 OR G>SX THEN  GOTO 180
250 LET H=INT(HY+HY*SIN(B*D)*E/F)
255 REM LINE 125,95,6,H:REM [see comments above]
260 REM PLOT G,H: REM [see comments in changes above]
263 LET K=G+5: LET L=H+5: REM [see comment CHANGES ABOVE]
266 LINE G,H,K,L: REM [see comments at CHANGES ABOVE]
270 NEXT D
280 GOSUB 370: REM [see comments at CHANGES ABOVE]
290 REM USER HRDMP [see comment CHANGES ABOVE]
300 CSR 2,23: PRINT "A= ";A
310 CSR 12,23: PRINT "B= ";B
320 GOTO 320
370 REM PRINTING ROUTINE - DMX 80 PRINTER
375 REM set line spacing pitch 24/216 inch feed [@ line
    380]
380 LPRINT CHR$(27);"3";CHR$(24);
390 FOR P=191 TO 0 STEP -8
395 REM set standard bit image designation [@ line 400]
400 LPRINT CHR$(27);"K";CHR$(255);CHR$(0);
410 FOR Q=0 TO 255
420 LET Z$=GR$(Q,P,8)
430 LPRINT Z$;
440 NEXT Q: LPRINT : NEXT P
450 RETURN
```

BASIC THINGS ABOUT BASIC

By Phil Eyres

Many members have requested a few articles about the very Basics of Basic, so having had several short programs sent in, I thought I'd poach them.(I don't really mean that - although I have lost the author to these programs!).

The first program is fairly short, but non the less has a few interesting points to note, here's the program:-

```
10 REM FACTORS
20 INPUT "NUMBER ";X
30 IF X<=1 THEN GOTO 20
40 IF X<>INT(X) THEN GOTO 20
50 LET N=1
60 PRINT : PRINT "FACTORS ARE"
70 CLOCK "000000"
80 PRINT X
90 IF X/(X-N)=INT(X/X-N)) THEN PRINT X-N
100 LET N=N+1 : IF X-N=0 THEN PRINT "    (";TIME$;")" ELSE
GOTO 90
110 PRINT : GOTO 20
```

Firstly, let us run through the lines of the program:-

line 10 is a 'comment' line, the command REM signifies that all text following is only comment and will not be executed when the program is run.

line 20 has an INPUT command, this will allow you to input a number into the variable X. Notice the word NUMBER in quotes will be printed on the screen as a prompt for you to type in the value to be stored in X.

line 30 is checking the number typed in to ensure that it is greater than 1, in that if it is equal to or less than one you will be returned to line 20 to input another number. Notice that this shows that line 20 is not very friendly as the prompt does not tell the user that the number he is being asked to type in should be greater than one!

line 40 is a further check on the number, this time to ensure that the number is a whole number, this is because the program will only work successfully with whole numbers.

line 50 is initialising a count - variable N - to one, this number will be incremented and used in the calculation below.

line 60 will firstly just move the cursor one column down the screen and then print the words FACTORS ARE
line 70 initialises the clock to zero hour, as soon as this is done it starts counting again, and timing the execution of this part of the program.

line 80 prints the number that you entered in, which was stored in variable X.

line 90 This is the main algorithm in the program, that is, it is the set of instructions making up the calculation and test, depending on the outcome of the IF statement the value of X-N may or may not be printed.

line 100 One is added to N. When N = X then the time held in TIME$ will be printed, signifying the time taken to run the program, otherwise execution is passed back to line 20.

line 110 just restarts the program after it has finished.

COMPETITION

That just about covers the program as it stands, if you try it out, you'll find it is not at all 'wizzy', text printed on the screen just scrolls up the screen and it's not all that informative. Also, it is not very crash-proof, you require that only a number is input and yet there is nothing to stop you entering text and for it to be excepted. The program could be much better with just a few extra commands to say, accept all input from the keyboard and reject anything that is not a number in the required range, also the screen could be made much more informative and also we could stop the scrolling. Really, since this is a page of help, I should now supply the program in a revamped form, but as it's not too difficult, and just about everyone can have a go at it, I'll offer too the best solution, one (or more) copies of MTX Fruit Machine. If I could have all entries by around the middle of May please!!!.

It should be stated that the club is somewhat indebted to John Grayson for supplying the club with a few copies of his program for use in competitions. Many Thanks!!!.

Overleaf are three more programs to have a play with, they all work on much the same principle and so should be easily followed.

```
10 REM LOWEST COMMON MULTIPLE (TIME TAKEN IN BRACKETS)
20 INPUT "FIRST NUMBER  ";X
30 IF INT(X)<>X THEN  GOTO 20
40 INPUT "SECOND NUMBER ";Y
50 IF INT(Y)<>Y THEN  GOTO 40
60 IF X=1 AND Y=1 THEN  GOTO 20
70 IF X=0 OR Y=0 THEN  GOTO 20
80 CLOCK "000000"
90 LET G=X: LET H=Y
100 LET K=1: LET L=2
110 IF X/L-INT(X/L)=0 OR Y/L-INT(Y/L)=0 THEN  GOTO 140
120 LET L=L+1
130 GOTO 110
140 LET K=K*L
150 IF X/L-INT(X/L)<>0 THEN  GOTO 170
160 LET X=X/L
170 IF Y/L-INT(Y/L)<>0 THEN  GOTO 190
180 LET Y=Y/L
190 IF X=1 AND Y=1 THEN  GOTO 210
200 GOTO 110
210 PRINT "THE LOWEST COMMON MULTIPLE OF      ";G;" AND
";H;" IS ";K;: PRINT " (";TIME$;")"
220 PRINT : GOTO 20
```

Listing 1.

```
10 REM HIGHEST COMMON FACTOR (TIME TAKEN IN BRACKETS)
20 INPUT "ENTER LARGER NUMBER   ";X
30 IF INT(X)<>X THEN  GOTO 20
40 INPUT "ENTER SMALLER NUMBER  ";Y
50 IF INT(Y)<>Y THEN  GOTO 40
60 IF X<Y THEN  GOTO 20
70 IF X=1 AND Y=1 THEN  GOTO 20
80 IF X=0 OR Y=0 THEN  GOTO 20
90 CLOCK "000000"
100 LET G=X
110 LET H=Y
120 LET G=G-H
130 IF G>H THEN  GOTO 120
140 IF G<H THEN  GOTO 170
150 PRINT : PRINT G;" IS THE HIGHEST COMMON FACTOR OF
";X;" AND ";Y;: PRINT "   (";TIME$;")"
160 PRINT : GOTO 20
170 LET H=H-G
180 IF H>G THEN  GOTO 170
190 IF H=G THEN  GOTO 150
200 GOTO 120
```

Listing 2.

```
10 REM AVERAGES
20 INPUT "HOW MANY NUMBERS?  ";N
30 IF N<=1 THEN  GOTO 20
40 PRINT : PRINT "ENTER NUMBERS"
50 DIM A(N)
60 FOR T=1 TO N
70 PRINT T;" ";
80 INPUT A(T)
90 NEXT T
100 PRINT
110 LET G=0
120 FOR T=1 TO N
130 LET G=G+A(T)
140 NEXT T
150 LET G=G/N
160 PRINT "MEAN IS    ";INT(G*100+0.5)/100
170 FOR H=N-1 TO 1 STEP -1
180 FOR T=1 TO N-1
190 IF A(T)>A(T+1) THEN  GOSUB 340
200 NEXT T
210 NEXT H
220 LET D=INT(N/2+0.5)
230 PRINT "MEDIAN IS  ";A(D)
240 LET V=1
250 FOR T=1 TO N-V
260 IF A(T)=A(T+V) THEN  GOTO 290
270 NEXT T
280 GOTO 310
290 LET G=A(T): LET V=V+1
300 IF V<N/2 THEN  GOTO 250
310 PRINT "MODE IS ";INT(G)
320 CLEAR
330 PRINT : GOTO 20
340 LET D=A(T+1)
350 LET A(T+1)=A(T)
360 LET A(T)=D
370 RETURN
```

- -

INTERFACING PROJECTS

Why not take the plunge and take a lead into the exciting world of micro electronics. Infact what better way to start than with an MOC D.I.Y. kit. Everything you need is supplied, except a soldering iron, wire cutters and of course a few hours of your time!!. So why not order now.

Interface price list

A full set of components and instructions for the LED kit
-->£6.95

A full set of components and instructions for the Speech Synthasiser kit
-->£18.00

Connecting cable for the internal port (needed for projects)
-->£4.50

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

MEMOTECH MTX MICROS'S "DUMPA4" GRAPHICS SCREEN DUMP UTILITY

by Eric Roy

Below is an assembler screen dump routine that will produce a hardcopy of mammoth proportions, infact it will entirely fill an 11" form (so be sure to have your paper lined up correctly before you start.

The program although written in assembler will take about 8 minutes to dump a screen.

The program is mostly self documenting, your graphics routines should be placed from line 290 onwards and the dump routine called by a GOSUB 270 command.

Your program should support the following control codes:

```
27,"U",1      Unidirectional print
27,"A",n      Set line feed to n/72
27,"K",n1,n2  n1+n2*256 data per line (384)
27,"@"        Reset printer
```

```
100 REM ********************************
110 REM **** GRAPHICS SCREEN DUMP ****
120 REM ****                     ****
130 REM ****  by Eric Roy  Nov.86 ****
140 REM ********************************
150 REM
160 REM-------------------------------
170 REM Program name  DUMPA4.BAS
180 REM
190 REM Graphics screen dump for Epson
200 REM type printers. The screen dump
210 REM produced is A4 size, will fill
220 REM a sheet of listing paper.
230 REM
240 REM GOSUB DUMPA4 code line
250 REM-------------------------------
260 GOTO 290
270 CODE

41F2 DUMPA4: CALL PRTESC
41F5         LD A,"U"
41F7         CALL PRINT
41FA         LD A,01
41FC         CALL PRINT       ; Unidirectional print
41FF         CALL PRTESC
4202         LD A,"A"
4204         CALL PRINT
4207         LD A,03
4209         CALL PRINT       ; 3/72" line feed
420C         LD H,00          ; H = X co-ord
420E XLOOP:  CALL PRTESC
4211         LD A,"K"
4213         CALL PRINT
4216         LD A,128
4218         CALL PRINT
421B         LD A,01
421D         CALL PRINT       ; 384 bits / line
4220         LD L,00          ; L = Y co-ord
4222 YLOOP:  PUSH HL
4223         LD A,H
4224         LD (X),A
4227         LD A,L
4228         LD (Y),A         ; Store X,Y in code
422B         RST 10
422C         DB £85
422D         DB 27,67         ; GR$ function, get pixel
422F X:      DB 0
4230 Y:      DB 0,1           ; = GR$(X,Y,1)
4232         LD A,(£FE1A)     ; Get result of GR$
4235         AND A
4236         JR Z,OFF
4238         LD A,07          ; Send 111 binary if on.
423A OFF:    CALL PRINT       ; or 000 if off.
423D         CALL PRINT       ; Send data twice
4240         POP HL
4241         INC L            ; Y=Y+1
4242         LD A,L
4243         CP 192
4245         JR NZ,YLOOP      ; Next Y
4247         LD A,£0D
4249         CALL PRINT
424C         LD A,£0A
424E         CALL PRINT       ; Send CR LF
4251         INC H            ; X=X+1
4252         LD A,H
4253         AND A
4254         JR NZ,XLOOP      ; Next X
4256         CALL PRTESC
4259         LD A,"@"
425B         CALL PRINT       ; Reset printer
425E         RET
425F PRTESC: LD A,27          ; Escape code
4261 PRINT:  LD B,A           ; B=char/data
4262         PUSH AF
4263         CALL £0CE3       ; Rom printer routine
4266         POP AF
4267         RET

Symbols:

PRTESC  425F   PRINT   4261
XLOOP   420E   YLOOP   4222
OFF     423A   X       422F
Y       4230   DUMPA4  41F2

280 RETURN
290 REM ****  START OF PROGRAM  ****
```

HARDWARE AND SOFTWARE PRICE LIST

Basic Computer
256K Computer + Tape operating System £99.95

System One
1 Mgbyte 3 1/2" Drive + I/F £166.00

System Two

1 Mgbyte 3 1/2" Drive + I/F
512K Silicon Disc, 80 Col.
+ CP/M + Neword £264.00

HX12 Green Screen Monitor £95.00

Twin RS232 Serial Interface £29.95

We can offer DMX 80 printer ribbons for only £7.00 each, so why not order one today and be prepared for the day your ribbon finally 'bites the dust'!!!

The MTX FIG-FORTH requires an MTX512 or expanded 500, the dictionary associated with Forth is held as part of the Ram-Disc which can be saved separately, fairly quickly. The Ram-Disc allows for 24 'edit' screens to be created and in memory simultaneously. A tutorial will be necessary for the beginner, for this the club has obtained a quantity of the publication Fundamental Forth(This may vary according to availability).

Fig-Forth Program £6.00
Tech Data Sheets £2.00
Tutorial Book £7.50 (240 pages)

Cheques payable to MDC please, orders from stock normally despatched by return, else, please allow 2 working weeks.

Ron Gladwin of UK Home Computers, (Tel 0793 695034) has on offer a Spectrum Loader that will convert your 512 into a 40K Spectrum, this will allow you to type in Spectrum programs, it is also said to load some Spectrum software. At £2.95 you may find it worth a try.

Ron also has some FOX Silicon disc's on offer, he has both 256K and 1Meg boards. The 1 Meg boards are only £100 and are available from us. Just think, you could be running your Supercalc or Newword at 3 to 4 times the speed you are used to.

All 'SUPER CHEAPIES' will be despatched by return of post.

!!!SUPER CHEAPIES!!!
(ONLY FROM STOCK)

DESC	QTY	PRICE (Each)	DESC	QTY	PRICE (Each)
			THE ZOO	3	£4.50
RETURN TO EDEN	1	£7.00	COBRA	1	£4.50
EMERALD ISLE	1	£7.00			
BLOBBO	5	£4.50	MINER DICK	1	£4.50
KILOPEDE	3	£4.50			
REVERSI	1	£4.50	HELI-MATHS	2	£4.00
MINEFIELD	3	£4.50			
FIRE HOUSE FREDD	2	£4.00			
TOADD	4	£3.50			
NEMO	1	£4.50	PONTOON &B'JACK	3	£4.50
SNAPPO	3	£4.50			
PAYROLL	1	£10.00			
PURCHASE LEDGER	1	£7.00			
PHYSICS 1	3	£5.50			
MATHS 1	1	£5.50	TAPEWORM	1	£4.50

Software prices for the best and most popular software:-

Zarkos	£7.00	Chamberoids	£7.00	
Qogo2	£7.00	26‡26 SpreadSht	£8.50	
Karate King	£7.00	Son Of Pete	£7.00	
S.M.G	£7.00	T.Snooker	£8.00	512 Only
Doodlebugs	£6.00	Super Bike	£6.00	
J.J.Flash	£6.00	Ed/Asm	£8.50	
Cee-5	£7.00	MTX Asm Lang Cse	£10.00	
Highway Encounter	£8.50			

Order a copy today!!!!!!

Remember PRINTER RIBBONS are only £7.00
REVEAL is only £6.00
SMG II is only £6.00

See page four for information about our interfacing kits.

Don't forget to order your copy of Fruit Machine from Graysoft today!!!!

ASSEMBLY LANGUAGE PART 1

BY GEOFFREY GARDINER

The MTX comes equipped with the capability to insert Z80 assembly language routines into BASIC programs. PANEL provides the capability to disassemble machine code programs into assembly language, and to debug assembly language programs. Programs are available on disc and tape to enable one to write programs entirely in assembly language. Those who have a CP/M version 2.2 system have the capability to write, disassemble and debug programs in 8080 assembly language, and by the use of the program VDEB.COM, to do the same to all CP/M 80 programs in Z80 code.

Normally users will prefer to work in Z80 assembly language entirely as it is irritating to use two sets of assembly instructions when one will do. We are, after all, using a Z80 microprocessor so the fact that most of the programs we are using were written for the 8080 CPU is irrelevant. Besides, most of the programs seem to contain some Z80 instructions, like JR, which were not implemented on the 8080, and if you disassemble a program with DDT you will find queries in respect of items of machine code which DDT cannot disassemble as it only knows 8080 instructions.

Why are we provided with this plethora of tools for using assembly language when everyone knows it takes an age to write a program in machine code that one could do in a day or two with a high level language? The only satisfactory answer to that question is like the reason for climbing Everest - it's there. You cannot really understand computers unless you know the language they speak. Using a high level language is the equivalent of communicating with a foreigner through an interpreter; you are dependent on the skill of the interpreter. Another reason for knowing assembly language is to be able to correct mistakes in programs, or alter them, a process impossible to carry out in the original high level language (except BASIC). One thing that you are not likely to do is to write an original complete program in assembly language unless you have the patience of Job and the longevity of Methuselah.

I propose to review the steps that would be necessary to learn to understand assembly language. It necessitates the purchase of several books all of which are expensive, from £13 upwards.

Of the books that set out the basic information about Z80 programming the most comprehensive, the dullest, and the largest, is "Programming The Z80" by the prolific Rodnay Zaks, published by Sybex. The greater part of this book concerns the Z80 instruction set, but it also deals with basic computing concepts, and basic programming techniques (using "basic" in its popular meaning!). This book is probably a must, especially as a reference book.

If you are going to write programs there is no point in reinventing the wheel so one may as well make use of existing subroutines (most of the essential routines that BASIC uses appear to be non-copyright), and the Mcgraw-Hill book "Z80 Assembly Language Subroutines" contains masses of routines. But perhaps the best book for giving one an understanding of how machine code operates is William Nitschke's book, "Advanced Z80 Machine Code Programming", published by Interface Publications. This book is incredibly knowledgeable and it shows one the principles for writing adventure games, maze games, arcade games, and business programs. It also lists the "undocumented" Z80 instructions, the ones the designers of the chip may not have known about(!), though what actual use they would be I don't know. But although one cannot praise this book too highly one must also warn that it is full of errors of detail, some of the most elementary kind. The author even gets his left and right mixed up, as on page 52, line 3, and his least significant and most significant nibbles reversed as on the next page. He makes a strong warning about the errors one can make with counting routines, but it seems to me that the upper example on page 36 has the wrong value in the BC register. I have tested it.

Incidentally you would not have much joy from writing his arcade game routine into the MTX because he does not anticipate the use of a machine with a separate memory for the VDU chip.

But knowing assembly language is not enough: to use it you have to understand the hardware, and this is probably the principal reason why professional programmers prefer to use a high level language. Programmers are often mathematicians by training and inclination, not digital electronic engineers. They are like racing drivers who have no idea how the car engine works. But to program in machine code you have to have some comprehension of how the hardware works. You must understand a data sheet.

There is a book that helps with this aspect. It is "Z80 Applications" by Joseph W. Coffron, published by Sybex. You don't have to understand everything in this book. Funnily enough you can skip most readily the chapter that

CONTINUED OVERLEAF

tells you how the Z80 itself works. The things one has to understand for programming are Interrupts, the Z80-CTC, the PIO, and the Z80-SIO. The book does not cover any of the output chips that we use, the Texas TMS9918 visual display processor and the sound chip in the MTX, and the Motorola 6845 vdp in the FDX. The first two are well described in our manuals and fact sheets are available. Maplin are supposed to stock other data sheets though not in my local branch. They are essential.

If a textbook contains a mistake it is likely to be repeated ad infinitum. I was led to wonder if this book was the source of the problem that the MTX suffers from in the programming of the serial port (the "SIO" or "DART") for on page 259 there is a program to initialise the SIO and at lines 181C to 1822 it sends the hex value 68 to write register 5 of the chip. This sets the chip out of accord with the RS232C convention. The MTX uses this value - it was even mentioned in the technical brochure - and was the reason my serial port did not work. This I have described in previous articles, but I have not previously mentioned where the error occurs in the MTX ROM. If you cannot make a modem or a printer work from the serial ports of the RS232 board try using PANEL to change memory location £0D20 from 68 to £EA. Those who own FDX's can make a permanent correction in FDXB.COM by using VDEB or DDT to alter location £0E23 from 68 to £EA. Earlier articles have explained how to alter the similar errors in BAUD.COM, CONTACT.COM, AND CONTVI.COM. This is an example of how one can find and correct errors in programs if one knows something about assembly language and machine code.

The best way to learn to understand assembly language is not by attempting to write a program but by delving into an existing program and trying to understand what it does. You could try using using PANEL or VDEB to list programs in order to examine them, but there is a severe handicap for the inexperienced. No disassembler program is capable of distinguishing a string from machine code, so it disassembles for instance ASCII code as if it were machine language instructions. You can detect the ASCII string by simultaneously "D(isplay)"ing the code and using the "I" instruction. What you cannot detect so readily is the special code which follows an RST10 or RST28 instruction. The output of the disassembler needs considerable amendment to convert it into a readable assembly language program. The ideal would be to send the disassembled program to disc as an ASCII file and then to read and amend it with Newword. As a disassembler cannot produce labels for memory locations one should be able to use Newword's "find and replace" instruction to replace memory locations in the disassembled code by labels of one's own devising. Labels do make an assembly language program more understandable.

One's first efforts, therefore, should be to study printed examples of assembly language programs. Then try understanding the ROM listing if you can afford the price and the time. Then have a go at disassembling bits of the programs with which we have been provided.

For the latter one will also require an understanding of the CP/M system and for this two more books are necessary:-the Osborne/McGraw-Hill "CP/M User Guide", and Ken Barbier's "CP/M Assembly Language Programming", published by Prentice-Hall. The latter has the advantage of being written with humour. Both books have the disadvantage of using 8080 code, not Z80. I am not aware of any CP/M textbook which uses the Z80 instruction set. Rodnay Zaks book has the translations of 8080 to Z80, and vice versa.

You may want to have a print out of the Hex code for a program. It is possible for FDX owners to use CONTACT.COM to do this. This program has a facility to send what is called a "HEX image" of a file to serial port B. If you attach a printer to serial port B and tell CONTACT.COM to send a hex image of a program to the serial port you can print out the whole of the machine code of a program. Page 118 of Ken Barbier's book tells you what a Hex image is but briefly in each line of code the 3rd, 4th, 5th, and 6th digits are the line number, and the 9th to 40th digits are the code. The last two digits of each line are a checksum.

Some of the tricks which assembly language programmers use are very clever; for that reason they puzzle one when one first studies them, and it may require a lot of patience and percipience to understand what is going on. It is a bit like doing the "Times" crossword but is more a useful way of expending brain power.

oooOOOooo

YOUR LETTERS

Answers To Questions

1.I may be able to answer a question posed by I.M.Collier in the magazine, who wondered why he had to use ROM3 with his SDX system instead of ROM5 as per the manual. Well, I bought a 500K SDX disc system last June and throughout the manual it does indeed say use ROM5. However, there was a single sheet enclosed with alterations for the SDX system only and to use ROM3 instead of ROM5. Hope this answers the problem.
P.E.MARTIN St Helens, Merseyside.

2.I am writing to you with some information which may be of help to some of your readers.

A recent letter in Memotechniques set me thinking. The request is for auto program start-up (from within CP/M).

A solution that I have arrived at, which doesn't use SUBMIT, is to use STARTUP. (As I am always telling my son, 'it's all in the manual!').

The sequence of operations is as follows.
* The disc needs to be formatted, SYS'ed and include STARTUP.COM and, for example SDXB3.COM
* Setting this disc as system, the following commands are required:-

```
    A>STARTUP SDXB3 40 filename.BAS\
or  A>STARTUP CONFIG\SDXB3 filename.BAS\
```

* Now every time this disc is loaded, or the RESET keys used, instead of displaying disc configuration, the above commands are executed.

* Note that, having done this, it may be necessary to re-enter the disc in CP/M. The only way I have found to do this is to place a non startup disc in the drive, and either a)copy a file to the startup'ed disc, or b)SYS the startup'ed disc or c) ^c reset from another disc: either way you achieve the desired A> prompt in CP/M.

As I am not an expert in CP/M, I have only found this out by accident (BUT IT IS IN THE MANUAL I hear you say!), and now I have A>STARTUP NW\ A>STARTUP SC\ A>STARTUP SDXB3\ etc

The following example is a short test program I have written in BASIC to enable the first BASIC file loaded to be a program listing, with simple key input (hence allowing any program to be menu driven, be it a ".RUN" or ".BAS" file

From CPM A>STARTUP SDXB3 40 DISCLIST.BAS\

In Basic

Program Name "DISCLIST.BAS"

```
0 POKE 64145,132: POKE 64862,13: REM disable break, enable keypad
10 PAPER 1: INK 7: CSR 10,0: PRINT "DISC LIST":
CSR 10,1: PRINT "---------"
20 CSR 5,4:PRINT "A: TEST"
30 CSR 5,5:PRINT "B: XXXX"
40 REM list of programs with single key reference
50 REM --- continue list of program ---
100 CSR 5,23: INPUT "PLEASE ENTER SELECTION > ";Z$
110 IF Z$="<>?" THEN POKE 64145,128: POKE 64862,143: STOP:
REM this enables you to get out of a loop. Use your own
pet code for <>?
120 IF Z$="A" THEN DISC LOAD "TEST.BAS"
130 IF Z$="B" THEN DISC RUN "PROG.RUN"
140 REM and load/run any prog you want to allow.
999 CLS: GOTO 10
1000 DISC SAVE "DISCLIST.BAS":RUN
```

Prog Name "TEST.BAS"

```
0 POKE 64145,132:POKE 64862,13
10 FOR A = 0 TO 15: FOR B=15 TO 0 STEP -1
20 IF INKEY$="?" THEN POKE 64145,128:POKE 64862,143:STOP
30 INK B: PAPER A
40 CSR 10,10:PRINT "TESTING"
50 PAUSE 50:NEXT B:NEXT A
60 DISC LOAD "DISCLIST.BAS"
1000 DISC SAVE "TEST.BAS":RUN
```

In both cases, GOTO 1000 will save & autorun the program. also both have a means of stopping, despite disabling the BREAK key.

I hope the above information proves useful.

For Sale

Floppy Disc Cleaner, 10 * 5 1/4" DS DD disc's,Tape to Disc Program (£7.95).

Offers please to Neil Duncan on Basingstoke 59765

- - - - - - - - - - - - - - -

Memotech MTX500, hardly used. With Handbook, and 4 games. £40 + £5 postage and packing. Robert Houghton, 51 Clothorn Rd, Didsbury, Manchester. Phone 061 445 4407.

Questions

1.Most (if not all) Australian Bullitin Boards use the XMODEM protocol for up/down loading files and CONTACT doesn't seem to measure up. Do you know of anywhere in Britain that I might purchase an XMODEM program thats works under CP/M on the Memotech. (MODEMY.COM or similar). I've been unable to convert these programs from other machines and end up with some very original screen effects.
Barry Smith 62 Hay St, Bullaburra, N.S.W. 2784.

2.I notice from "the other place" that there is a 'CP/M Plus' version for the Memotech, do you know anyone who has it? I'm thinking of upgrading but would like to know what it's like (the Memotech Version), the real cost involved and any problems of compatibility encountered.

Daffyd Robinson Immingham, Sth Humberside.

Ed-> This is the first I've heard, has anyone anything to offer on this?.

3.I have been using the Supercalc capability to run loading/stability calculations for ships; ideal by computer since they involve lots of repetitive maths, and especially in the "What If ...situations?", but I need a link formula or sub-program, to instruct the working program on display, when it reaches a certain part of the routine, to access other data files on the same disc, using a figure which appears in one of the working programs cells, and match the figure to information held in the data file, retrieve other information that relates to the input, and return to the work program to display retrieved information!!!

EG. In line 230 of work program, Cel no BB Total 22,368

Instruct computer to access one of three data files at col B, read down col to find 22368-1 or nearest lowest.

Read across 7 columns of figures, retrieve said information and return to work program operation in auto.

This total routine has eluded me with SC and I have tried various methods using the Execute routine of SC, and Save - Exit Work File - Enter Data File No 5,7, or 9 and return, with no luck.

I can visulise how it should work, but I'm dammed if I can make it do so. Any help and/or suggestions gratefully received and acknowledged.

Chris Rowe, Statensingel 144d., 3039 LW Rotterdam

4.On the advice of a 'knowledgeable friend I recently purchased a 64K memory expansion in order to allow my humble MTX 500 to run the latest '512' only software and to use certain programs I already have on disc i.e. an address book, database and a spreadsheet. Now having left the army I have more time for computing so I assembled it and set to. But horror, most of the programs saved on disc didn't work unless I POKE'd 64122,0 then NEW resetting to 32K. Next I tried a piece of commercial software supposedly for the '512', (The ZOO) this seemed to work until part way through the game, it promptly stopped with the error messages 'No Line', just to check, I EDIT'ed the line and it was there. I entered this line and tried to GOTO the line again, 'No Line', despite the fact I had just seen it and could edit it. Next step was to try POKE 64122,1, then NEW, now only 64K available but still the same result. My obvious thought was a faulty memory expansion so I checked it best I could, PEEK (64122) under normal circumstances i.e. after resetting, gave the expected 2. DIM A(13100) worked. At this point I was truly confused.

Further reading mainly about memory mapping nearly provided the answer, but I still cannot understand why a short 2-3K program should need to change pages. As a result of all this I've written to you in the hope that you can provide the answer. Is it the software(very unlikely I think) or the hardware?. If it is the hardware it would seem that I've been badly advised and now have a system that won't do what it was bought for. Assuming that it is all working properly.

Ed-> This would definitely seem to be a hardware problem. Some issues ago we printed a memory testing program, perhaps you could try that. I would say that it sounds most like a memory board fault or a conflict between the boards you have fitted in the system.

Help

1.I spend quite a bit of time working with alternative fonts for my Canon PW 1080 printer (there is an equivalent machine, the Taxan Kaga), and do my own Eprom "blowing" to make font ROMs for maximum convenience. I would be quite happy to do custom EPROM blowing for club members without the facility, or even to provide ready-made font ROMs for those who might have a Canon or Kaga printer, for a very modest charge.

At the moment, I can handle only 2764s and 27128s, but hope shortly to include 2716s and 2732s as well.
Ken Rendall 30 Saint Baldred's Road, North Berwick, East Lothian. EH39 4PY. Tel 0620 2282.

THE COUNTER TIMER CIRCUIT AND CASSETTE INPUT

By Mike Kohnstamm

The counter timer circuit (CTC) is a chip within the computer that receives an input signal and generates an output pulse, either for every input, or for every so many input pulses. In the MTX the input may be the computer's internal clock signal or the cassette recorder input for example. The output signals are fed to the Z80 processor chip's interrupt, to break into whatever program is running, temporarily, in order to do a small amount of processing. In fact, there are four separate channels in the CTC. In the MTX the first channel generates the normal interrupt, where the break key is tested etc. The second channel is used for SAVEing data to cassette and the fourth is used for LOADing.

It is possible to program each of the four channels by sending a byte of data using the OUT instruction. A longer description is given in 'An Introduction to Z80 Machine Code' by R.A.&J.W. Penfold - number 152 in a series of small paperbacks by Bernard Babani. (This is not only a cheap assembler reference book, but also includes the MTX in its examples.) The byte of data is sent to channel 1 with OUT(£08), to channel 2 using OUT(£09), channel 3 with OUT(£0A) etc. This byte is treated as eight separate bits, each having a distinct meaning. It is stored in the control register of the CTC. The high order bit (bit 7, counting from right to left and starting at zero) is set to 1 if the CTC is to generate interrupts, or to 0 if not. Bit 1 (the lowest but one) is set to zero if the CTC channel is to be used, or 1 to switchoff the channel. Bits 2, 5 and 6 determine how many input pulses are counted before an output (interrupt) is generated (ie. they 'divide' the input signal). If bit 2 is set to 1 a second byte must be sent holding a number from 0 to 255 (0 is treated as 256) which is the number of input pulses before producing an output. This number is stored in the divide register of the CTC. However, it may be necessary to count more than 256 inputs. In this case the prescaler is used and bit 6 is set to 1. (When bit 6 is 1 the prescaler is switched off). Then, if bit 5 is zero the number in the divide register is multiplied by 16, and if bit 5 is 1 the divide register is multiplied by 256. So it is possible to divide the input signal by as much as 65536 (256 * 256). Some examples from the MTX ROM should help to make this clearer.

At £0996, performed whenever the machine is switched on, two bytes are sent to channel 1. The first is £A5 (ie 165 in decimal) or 10100101 binary. The first of two such bytes always has bit 0 set to 1. Also, bits 2,5 and 7 are set to 1, so interrupts are generated every so many clock signals, where the next byte is £7D (125 decimal) and the prescaler is set to 256 resulting in an interrupt every 32,000 (125*256) clock signals or 125 times a second (although my manual claims it is 64 times a second, the clock update routine at £0924 shows that it is 125 times).

In the process starting at £097F, £03 (00000011 binary) is sent to all 4 channels. This switches off all operation of the CTC, and is performed before a LOAD or SAVE to disable the break key and other interruptions. The £097F process also sets Interrupt mode 2 which allows the processor to perform a different process for each device causing an interrupt. The four CTC channels perform different processes, whose start addresses are held in RAM at addresses starting at £FFF0. (This is also set in the £097F process, where £FF is sent to the interrupt register of the Z80, and £F0 is output to channel 1 of the CTC.) Examination of bytes £FFF0 to £FFF7 shows where these addresses are stored.

Interrupts produced by channel one perform a process whose start address is found at £FFF0 (low byte in £FFF0, high byte in £FFF1). Normally the address held here is £0780. The process starting at £0780 is the one that updates TIME$, tests the break key, moves sprites etc 125 times a second. By using location INTFF you can write your own interrupt process, to be performed in addition to the normal machine ones (see p.182 of 'the manual'). However, you can completely replace all the machine's interrupt processes by writing your own, and putting its start address at £FFF0. The third channel's interrupts perform the process at £1C11, although what this does is a mystery to me. The addresses held at £FFF2-3 and at £FFF6-7 are both the same (£0A53), so the second and fourth channel's interrupts perform a process at £0A53 which does nothing except alter the carry flag. An assembler program (eg SAVE and LOAD or userwritten program) can detect the changed carry flag.

The process starting at £0B10 is the key to using the cassette socket for input. This first calls £097F to switch off the unwanted channels of the CTC, which is why the break key does not function and TIME$ is not updated during LOADing or SAVEing. This is determined by the contents of 'TYPE', or RAM address £FD68. (1 for load or 0 for save - this is the same as in £AAE, which infact uses £B10 - see M.O.C. magazine vol 2 issue 2 p.15). For a SAVE, £C5 is sent, followed by the value of CASBAUD (£FD5F), which sets channel 2 to generate interrupts at the required rate to control the speed of output to tape.

CONTINUED OVERLEAF

For a LOAD, 2 bytes are sent to the CTC. The first is also £C5 or 11000101. Bits 0,2,6 and 7 are set to 1 so interrupts are produced for every input - since the second byte sent is £01. As a result, every cassette recorder signal causes an interrupt which alters the carry flag. As an example of using the cassette input, see the following program:

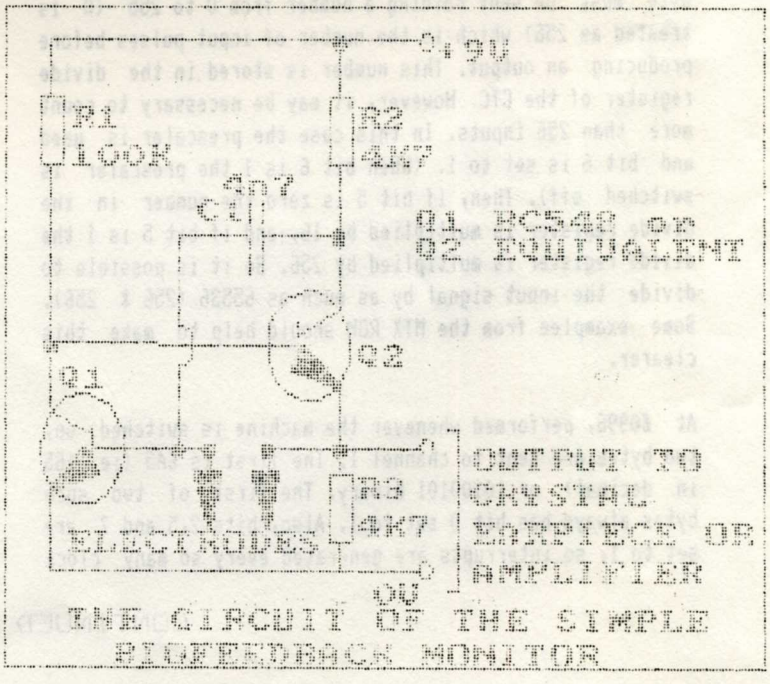
```
2 INPUT "SCALE (1 TO 8) =";S
4 PRINT "     ";
10 CODE
       LD A,1
       LD (£FD68),A
       CALL £0B10
       XOR A              ; THIS PART
       EI                 ; COULD BE
 WAIT1: JR NC,WAIT1       ; REPLACED
 WAIT2: DEC A             ; WITH
       JR C,WAIT2         ; CALL £0A6B OR
       DI                 ; CALL £0A6E
       XOR £FF
       LD (£C350),A
       RET

20 LET A=PEEK(50000)
30 REM 50000 IS £C350 HEX
40 LET A=INT(A/S)
50 IF A>32 THEN LET A=32
60 FOR I=1 TO A
70 PRINT "*";
80 NEXT I
90 PRINT: PRINT "     ";
100 LET Q$=INKEY$
110 IF Q$="" THEN GOTO 10
120 IF Q$="S" THEN GOTO 2
130 STOP
```

The Basic part of the program just prints out a line of asterisks, the length of which depends on the value in PEEK(50000). Pressing "S" allows you to scale the length of the line; pressing anything else stops the program. The assembler part of the program gets stuck in a loop at WAIT1 until an input from the cassette alters the carry flag. It then waits in a second loop at WAIT2 until another input is received. The number of WAIT2 loops is counted in the A register, which is then stored at £C350 (50000) for use by Basic. The XOR A resets the carry flag initially, and DEC A does not alter it, even if it goes negative (INC A can alter the carry flag, which is why the count decreases from 255, instead of increasing - the XOR £FF instruction then sets the A register to hold the difference from 255). As the instructions in the loop take 16 (4MHz) clock cycles, each loop lasts 1/250,000 of a second. Thus the length of the line of *'s is a guide to how many of these loops occurred between cassette signals - a longer line indicates lower frequencies. (This count goes back to zero after 255, or approx 1/1000 second, so for lower frequencies it is necessary to keep a two byte count.

The program is given above as an example of using the carry flag for testing the cassette input. To shorten it, the part of the program from XOR A to DI could be replaced with £0A6E. This routine is used by LOAD and returns a value of 0 or 1 in the carry flag depending on the time between signals - it loads a 0 or 1 bit.

What use is all this? As a frequency meter it is very rough, and only measures the highest frequency present at any time. A similar idea is given in 'The MEMOTECH MTX Program Book' by P.Goode where cassette input is used to create a light show on the T.V. screen. Another possible use, for electronics enthusiasts is a project in Electronics Today International in November and December 1986. This is a very easy to build project that can be used as a 'biofeedback monitor' or lie-detector. It was intended to be used by listening to the varying pitch of the sound on an earphone. However, by feeding the earphone input into the MTX the program above can be used to show the varying frequency (or adapted to give appropriate comments on detecting a lie).



THE CIRCUIT OF THE SIMPLE BIOFEEDBACK MONITOR

P R O G R A M L I B R A R Y
£1.20 Per Cassette, 2 Programs per Cassette

This month we have our first offering on the Fig-Forth front, from Peter Burns it's well worth a try. Also, a suite of excellent educational programs from Allan Ayre are also the first offering in this field for months.

1.Basic & Assembler Programs

---- Diskette Three ----

62.Account		The Third Money Manager
63.Mastermind		Another Good Game
64.Connect4		Two Player Game
65.Journey Into Danger		NEW Adventure Game
66.Connect4 Version 2		As for number 64
67.MTX DRAWv1.8		As for number 6
68.Patience		Card Game
69.Life		Curious Puzzle
70.Enigma		Like Mastermind
71.FKEY		Function Key Definer
72.Skydiver		Graphical Game
73.Digger		Graphical Game
74.MPG		Calculates MPG
75.FIG-FORTH	*NEW*	RAM Disc
76.Optics/Colours	*NEW*	Educational Programs

4.CP/M Programs/Utilities

(!!! Available only on 5 1/4" disc !!!, please send in a formatted disc (stating capacity) for each item and enough postage to cover - £2.50 per disc).

1.A simple mail label system for up to 3 across labels, written in EBasic. Disc includes Ebasic compiler and run-time program. Consists of a suite of half a dozen programs. Includes a sort routine.

2.PLOT33 A new graphics plotting package for Turbo Pascal owners. Create and print your own graphics. Set up for DMX type printers but will support most others. Must be seen to be believed. Please ensure you have at least two weeks free when ordering this one, you'll need it!!.

3. Z80.ASM This is a Z80 assembler to replace the ordinary CP/M assembler which uses the 8080 mnemonic command set. Z80.ASM supports all the features of the notable Ed/Asm, especially macro libraries and a slightly more standard Z80 mnemonic command set. The disc also contains a Z8 assembler.

4. SMALL C COMPILER. This is from the Swiss user group, it is however written in English so easily understandable. You will need to buy a Tutorial to use it, but even so it offers unbeatable value for money.

5. BASIC-E PROGRAMS. All the 'good old' text style games, originally designed for teletype style displays, non-the-less some good games.10 games in all, including Startrek. Disc includes Basic-E compiler and Run time program. Also included are several .PIC files with interesting pictures to print out - including a PINUP.PIC file, I wonder what that could be a picture of?? a drawing pin!! ....maybe?

5.Program Reviews

75 Fig-Forth Ram Disc - Peter Burns

This tape is for anyone that has the clubs MTX Fig-Forth.
Screen 0 contains Comments.
Screen 4 & 5 contains Error Reports.
Screen 23 contains Cold Parameters
And 19 screens of words.

The Ram Disc tape comes with comprehensive notes supplied by Peter, making up some 15 A4 pages.

This one I think would be very useful for anyone who has the clubs Fig-Forth.

76 Optics/Colours - Alan Ayre

The program comes in 3 parts, the first of which is an Intro.

The next program is called Lenses. This program demonstrates visually the changes that take place on an image as an object gets nearer to a lense. It demonstrates Convex and Concave lenses and the calculatinos associated with them. The program is said to be suitable for O Level or CSE Physics

Colours, is made up of 2 programs which demonstrate the effects of mixing the primary colours of light and also how the colour of an object depends upon the colour of the light shining on it. These programs use brilliant graphics, not that they are highly detailed or fast, but because they are simple, yet extremely effective. The screen initially shows a projector screen and three colour projectors, selecting a colour turns on a projector and alters the colours on the screen accordingly. All in all, a brilliant concept.

If you are interested in this sort of thing, or have moaned about the lack of educational software available, then this one is well worth a try.