VOL. 2 ISSUE 7 APRIL 1986

MEMOTECH OWNERS CLUB

CIRCA ...252

M.O.C.

VOLUME 2 ISSUE NUMBER 7

CONTENTS

c c c O O O c c c

E D I T O R I A L (APRIL 1986)

Phil Eyres
23 Denmead Road
Harefield
Southampton
SO2 5GS

I feel this month's magazine has gone together very well, with a good combination of interesting articles, (may I thank everyone for their obvious effort!) it could as usual be accused of being lacking in Basic programs, however I hope the program library makes up for this in some way. Oh!, by the way there are some really brilliant additions to the library this month - not least Eric Roy's Ram Disc, a truly marvellous piece of programming!!!.

We are still keeping well in touch with Christian our Swiss user group friend, he has just sent in a FDXB Basic slide show program containing 32 slides, some of the pictures are really good, I am still looking at it, so hopefully more next month.

After being with the club for nearly 2 years Rich has decided it's time to call it a day and move on, I think we can all wish him 'All The Best' for the future and hope that it has something exciting in store. Please therefore note the new Hotline number printed below.

At last the FIG-FORTH is ready and waiting, I have copies ready to send out so posting should be within a week. See page 6 for full details of this great new addition to the club.

Thanks to everyone who has used our Hotline on Monday evenings between 6 & 7pm, the number to phone now is (0703) 466106, ask for Phil. If we keep Mondays as Hotline night then I can be sure of being in. However, feel free to phone any evening after 6pm, if I'm not in the my Mum (good old Mum!!) will take any calls.

If anyone would like back issues they are available for the small remittance of 80p. At present there are 16 back issues, 10 for volume 1 and 6 for volume 2.

oooOOOooo

INTERFACING PROJECTS

Why not make a break which will lead you 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.

BASIC CHARACTERS

By

Don Williamson

This easily adaptable subroutine allows printing of characters of any height or width in any position. Ideal for intro screens, children's games or education, or users perhaps with poor eyesight. The parameters of the routine as it stands are:-

10-30 Clear graphics screen, print message string in the INK 0 along the top of the screen.

40 X and Y = top left start position

50 XX = Width of new character

YY = Height of new character

60 T = Next character in string

100 A = Graphic co-ordinate of string character.

102 I = Random INK selection

108 B = Graphic co-ordinate of string

105-180String character GR$ bit test loop. If bit = 1 then draw line multiplied by xx (width), YY times (height)

185 Next character in string.

190 If end of string then finish.

200 Next start position for draw line

250 Goto 100: Repeat routine on next character

Try LET A$="HELLO", LET XX=6, LET YY=15

```
10 VS 4: CLS
20 CSR 0,0
22 COLOUR 1,0
25 LET A$="HELLO OUT THERE, HOW ARE YOU?"
30 PRINT A$
40 LET X=10: LET Y=150
50 LET XX=1: LET YY=3
60 LET T=0
100 LET A=191
102 LET I=INT(RND*15)+1: IF I=5 THEN GOTO 102 ELSE INK I
105 FOR G=Y TO Y-(YY*7) STEP -YY
108 LET B=0
110 FOR J=X TO X+(XX*7) STEP XX
140 IF GR$(B+T,A,1)=CHR$(1) THEN FOR K=0 TO YY-1: LINE J,G-K,J+XX,G-K:
 NEXT K
150 LET B=B+1
160 NEXT J
170 LET A=A-1
180 NEXT G
185 LET T=T+8
190 IF T>LEN(A$)*8 THEN GOTO 300
200 LET X=X+(XX*8)
250 GOTO 100
300 GOTO 300
```

SPRITE POSITIONS AND COLLISIONS

By

Paul Wood

The 2 routines (RVRAM & RDATA) can be used to read any of the VRAM locations used by the ROM.

The routine demonstrates reading from the VDP. You will notice that once the RVRM routine has been called, the data is read from that address, then by recalling, RDATA the next address is read, this is because once read the address is auto-incremented to the next location.

The SPRITE ATTRIBUTE Table starts at hex 3F00, and the SPRITE PATTERN Table starts at hex 3800.

The Sprite Attribute Table is the table that is relevent to the sprite position on screen, and takes the form:-

```
1st byte  Vertical Distance from top of Screen
2nd byte  Horizontal Distance from LHS of Screen
3rd byte  Sprite Pattern No.
4th byte  Colour of sprite
5th byte  As 1st byte - for second sprite
```

Sprite Collision detection can be achieved by reading the VDP STATUS REGISTER and ckecking if bit 6 is set. This is achieved in assembler by:-

```
IN A,(02)       ;Read VDP Status Reg.
BIT 6,A         ;Is bit 6 set?
JP  NZ,COLLIS   ;If zero Flag is not set, jump to your
                 collision routine.
```

To obtain the same thing in Basic the following lines could be used as a subroutine, and on return check for the COLLIS variable.

```
REM  **  Sprite Detection in Basic  **
LET SPRCOLL = INP (02)
LET COLLIS = MOD(INT(SPRCOLL/32),2)
RETURN
```

On return from the subroutine variable COLLIS will = 1 if any collision between any two sprites has occurred, else COLLIS = 0.

As a word of warning, all unused sprites must be 'blanked' else the VDP will detect collisions between them, this is achieved by writing decimal 208 into the first unused sprites Y co-ordinates, the VDP will then terminate it's checking.

```
10 VS 4
20 COLOUR 0,2: COLOUR 1,1: COLOUR 2,2
30 CLS
40 CTLSPR 1,1: CTLSPR 2,2: CTLSPR 6,1
50 GENPAT 3,1,24,24,24,255,255,24,24,24
60 GENPAT 3,2,129,66,36,24,24,36,66,129
70 SPRITE 1,1,100,100,00,0,7
75 SPRITE 2,2,50,50,0,0,8
80 CSR 2,2: PRINT "HORIZONTAL POSITION ="
90 CSR 2,4: PRINT "VERTICAL POSITION    ="
100 CODE

80F2          LD DE,£3F00         ;Start of Sprite Attribute Table
80F5          CALL RVRAM ; Set to Read from VRAM
80F8          CALL RDATA ; Read Data
80FB          LD A,C      ; Put Data in Register A
80FC          LD (VERTPOS),A      ; Load DATA into VERTICAL POSITION Sto
re
80FF          CALL RDATA ; Read Next Data into VRAM
8102          LD A,C      ; Put Data in Register A
8103          LD (HORZPOS),A      ; Load into HORIZONTAL POSITION Store
8106          JP END      ;Jump to end of routine
8109 VERTPOS:DS £01       ;VERTICAL POSITION Store
810A HORZPOS:DS £01       ;HORIZONTAL POSITION Store
810B RVRAM:   LD A,E      ;OUTPUT VRAM address to PORT 2
810C          OUT (02),A
810E          LD A,D
810F          OUT (02),A
8111          RET
8112 RDATA:   IN A,(01)   ;READ DATA from VRAAM
8114          LD C,A
8115          RET
8116 END:     RET         ;RETURN to Basic
8117          RET

Symbols:
RVRAM   810B    RDATA   8112
VERTPOS 8109    HORZPOS 810A
END     8116

110 CSR 26,2: PRINT "   "
120 CSR 26,4: PRINT "   "
130 CSR 25,2: PRINT PEEK(33033): REM VALUE AT VERTPOS
135 REM VERTPOS MAY BE 16649 ON A MTX 512
140 CSR 25,4: PRINT PEEK(33034): REM VALUE AT HORZPOS
145 REM HORZPOS MAY BE 16650 ON A MTX 512
150 IF INKEY$=CHR$(25) THEN MVSPR 1,1,8
160 IF INKEY$=CHR$(11) THEN MVSPR 1,1,6
170 IF INKEY$=CHR$(10) THEN MVSPR 1,1,2
180 IF INKEY$=CHR$(8) THEN MVSPR 1,1,4
181 LET SPRCOLL=INP(02)
182 LET COLLIS=MOD(INT(SPRCOLL/32),2)
183 IF COLLIS=1 THEN CSR 9,7: PRINT "COLLISION" ELSE CSR 9,7: PRINT "
      "
190 GOTO 100
```

DISC CONTROL

By

Eric Roy

Below is a listing for the Memotech SDX disc. The program allows you to change disc under program control, so that you can load files/data from one disc then call this routine to enable the file/data to be saved onto another disc. Thus eliminating the use of the ROM 3 command which enables the disc to be written to but always returns to the basic ready prompt, and so cannot be used in a program.

Full details of the program are contained in the Edasm macro listing. I have also included the same program written on the Memotech assembler to show how it could be used in application.

```
SDX 250K DISC BDOS FUNCTION ODH RESET DISC SYSTEM
;
; CODE          C = 0DH
; ENTRY         NONE
; EXIT          NONE
;
; ADDRESS       E383H
;
; This function resets all the discs in the system, (250K
; only has 1)
; Disc in drive A set to default disc.
; DMA set to default address
; Disc's set to read/write
; Directory read in and allocation vector bit map rebuilt.
;
; It is not possible on the 250K system to read a file from
; one disc and write it to another in a program as the
; BDOS will detect that the disc has been changed and
; issue an error BDOS error on A: RO
;
; The following subroutine will allow discs to be changed
; under program control. It displays a message asking you
; to change the disc, then uses the reset disc system
; function to enable the disc to be written to.
;
; Calling sequence      CALL CDISC
;
        ORG     0A000H      ; Debugging address
        LOAD    0A000H
;
; Equates
;
GETKEY  EQU     00079H
RESETD  EQU     00DH
BDOS    EQU     0F5B0H
CR      EQU     00DH
```

Continued At Top Of Page

```
LF      EQU     00AH
;
CDISC
        RST 010H            ; Display message
        DB  09EH
        DB  CR,LF
        DB  "Change disc then press <RET>"
;
CDISC1
        CALL    GETKEY      ; Wait until RET pressed
        CP      CR
        JR      NZ,CDISC1
;
        LD      C,RESETD    ; Reset disc system
        CALL    BDOS        ; BDOS
        RET

        END
```

... For the Memotech!!

```
10  REM ********************************
20  REM **** 250K BDOS FUNCTION ODH ****
30  REM ****   RESET DISC SYSTEM    ****
40  REM ********************************
50  REM
60  GOTO 100: REM Start of program
70  CODE

40C6 CDISC:  RST 10
40C7         DB £9F,10,13
40CA         DB "Change discs then press <RET>"
40E7 CDISC1: CALL £0079
40EA         CP 13
40EC         JR NZ,CDISC1
40EE         LD C,£0D
40F0         CALL £F5B0
40F3         RET

Symbols:
CDISC   40C6    CDISC1   40E7

80 RETURN
100 REM READ FILE FROM DISC
110 USER READ "FUNCT_D.HEX",40960
120 GOSUB 70: REM CHANGE DISCS
130 REM WRITE FILE TO DISC
140 USER WRITE"FUNCT_D.HEX",40960,30
```

Many Thanks For A Super Article!!

HARDWARE AND SOFTWARE PRICE LIST

We still have no firm information about Memotech, so we have omitted the hardware section this month, it has been replaced with the details about the clubs new FIG-FORTH program.

All 'Super Cheapies' will be despatched by return of post.

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. The prices are lists below:-

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

Cheques payable to MOC please, orders normally despatched in 5 days max.

Software prices for the best and most popular software:-

Zarkos	£6.00
Qogo2	£6.00
Surface Scanner	£6.00
Chamberoids	£6.00
Fathoms Deep	£6.00
Quazzia	£6.00
Crystal	£6.00
Cee-5	£6.00
Roller Bearing	£6.00
Downstream Danger	£6.00
26*26 Spread Sheet	£7.95
Ed/Asm	£7.95
Memosketch	£7.95
Dust cover's Only	£3.50

! ! ! S U P E R C H E A P I E S ! ! !

(ONLY FROM STOCK)

DESC	QTY	PRICE (Each)	DESC	QTY	PRICE (Each)
DUNGEON ADV.	2	£7.00	THE ZOO	1	£4.50
ADV. QUEST	1	£7.00	COBRA	1	£4.50
EMERALD ISLE	1	£7.00	BRIDGE	1	£4.50
MAXIMA	1	£3.50	FIRST LETTERS	1	£4.50
BLOBBO	4	£4.50	WORD & PIC MATCH	1	£4.50
KILOPEDE	1	£4.50	BASIC BUSINESS	2	£5.00
REVERSI	2	£4.50	HELI-MATHS	2	£4.00
MINEFIELD	3	£4.50	SPELLI-COPTER	2	£4.00
BACKGAMMON	1	£4.50	FIRE HOUSE FREDDIE	2	£4.00
OBLOIDS	1	£4.50	ASTROMILLON	1	£4.50
NEMO	2	£4.50			
SNAPPO	2	£4.50	FROM ELSTREE COMPUTING		
PAYROLL	1	£10.00	CUSTOMER INF FILE	1	£5.00
PURCHASE LEDGER	1	£7.00	INVOICE & CR NOTE	1	£5.00
PHYSICS 1	2	£5.50			
MATHS 1	1	£5.50			

HARDWARE — A SET-UP

By

Phil Eyres

I thought that I would write a couple of articles to try and explain how all the bit's of electronics fit together to make a computer and also hopefully unravel some of the secrets of the technical part of the black MTX manual.

Well, firstly what bits go to make up an MTX?
Ref:- The System Block Diagram, Page 209.

This shows all the different bits of electronics which go to make up a basic MTX, lets take a quick trip through the main bits:

a) Z80A CPU. This is the central processing unit, the heart of any computer, it does all the work, just about everything that happens is in some way influenced by this component. All the code that you write, be it Basic,Assembler or whatever, controls the actions of the Z80A CPU.

b) Ram & Rom. Random Access and Read Only Memory. Another vital part of a computer, all the programs and data you write are stored here so that the CPU (central processing unit) can work in the way you want it to. Bye-the-bye, you have three Rom chips, the first contains your machines operating system, the second your Basic interpreter which runs your programs and the third is your Front Panel Rom, which you may or may not have used yet.

c) Clock. This is exactly as it says, a clock, it provides a steady signal which all the bit's require in order to work harmoniously together.

You will see that the lower 9 boxs in the block diagram all have either/or IN or OUT in them, this is because they are all peripherals and not part of the 'heart' of the computer. Along with the Rom it is these peripherals which make up a Memotech as a Memotech (or a Spectrum as a Spectrum).

How do all these bits fit together?

Well, with the aid of the system bus on page 208 I'll try and explain.

Looking at the diagram at the top of the page, which shows the main board shape onto which the electronics are fitted, you can see the board has two 60 way connectors (30 connections on each side!), these two connectors are called the system bus. As suggested they represent the full Z80A bus, which means that via these 60 connections you have the control to do anything that is physically possible with Z80A CPU.

You may know already that the Z80A is an 8 bit chip (as opposed to a 16 bit chip - like that used in IBM PC's) or that the most memory you can have is 64K, but somehow this can be expanded to 512K, but how are all these 'physical limits' calculated?

This is not so easy to explain as I really have to somehow also explain Binary.

Decimal is the 'base' you use in Basic and everyday life, all numbers are made up of the digits 0 to 9, your'e really happy with this as it's what you have been brought up with. The computer however prefers Binary, this is base 2, which means you can only use the numbers 0 and 1, so where as you might say, "Ah!, thats the number thirty two ->32" the the computer says "Ah!, thats the number thirty two ->00100000". It really just looks at the same number in a different way.

So, to try and explain these limits of an 8 bit machine which is only capable of having 64K of memory, refer to the diagram at the bottom of page 208 along with these notes.

As has already been said, the 60 connectors are termed the system bus, this is in turn broken down into 3 'buses' the 'Address Bus', the 'Data Bus' and the 'Control Bus'. Connections numbers A0 to A15 are the address bus, D0 to D7 the data bus and all the others except the power supply 'rails' are collectively termed the control bus. We are mainly interested in the first two, each of the 16 address lines is capable of holding a 1 or 0, this gives the maximum 'addressable' number of 2^16 or 65535 (64K). So the Z80A can inform the memory of any one 'byte' that it would like to write to or read from by putting that memory locations number (in Binary) on those 16 lines. Now how does it process the data to or from this memory?, yepp, it uses the data bus, this is lines D0 to D7 (8 bits of data!!). Having linked with the correct memory location by putting it's address on the address bus it can now transfer 8 bits of data to or from that memory location using the data bus. It does this with every single thing that it does, it just does it very quickly!!

How does it talk to peripherals?

Hopefully more Next month!!

All The Best, Phil

** Games High Scores Table **

AGROVATOR	61828	A.DOBSON
ASTROMILON	30830	T.NEAL
ASTROPAC	69390	A.DOBSON
BLOBBO	71233	T.PICKSTONE
B.BILL	219610	A.DOBSON LEVEL 1
B.BILL	158334	A.DOBSON LEVEL 9
CHANBEROIDS	19 MINS	P.ERIKSSON
COBRA	5634	A.DOBSON
CONT RAID	10810	M.GILL
D.DESTROYER	3380	T.NEAL
EMERALD ISLE	725	R.SIDDALL
E. ZARKOS	90 OBJ	R.SIDDALL
F. DEEP	1420	A.LYNCH
FELIX	20600	P.COUGHLAN
FLUMMOX	25700	*T.NEAL
GOLDMINE	6025	P.CRIGHTON
HAWKWARS	15850	P.CRIGHTON
HUNCHY	5681	T.NEAL
ICEBURG	17431	A.DOBSON
JUMP' J FLASH	2970	*T.NEAL
KARATE KING	1300	*T.NEAL
KILOPEDE	33440	P.CRIGHTON
KNUCKLES	488650	P.CRIGHTON
L OF TIME	950	R.SIDDALL
MAXIMA	501250	R.SIDDALL
MINER DICK	22520	R.SIDDALL
MISS ALPHA	44630	T.PICKSTONE
M OMEGA	4400	T.NEAL
NEMO	11080	P.CRIGHTON
OBLOIDS	60040	M.GELDER
PHAID	1965	A.DOBSON
P PETE	39630	A.DOBSON
QOGO 2	255000	R.SIDDALL
SNAPPO	79300	P.ERIKSSON
SNOWBALL	1000	P.COUGHLAN
S OF PETE	10542	P.ERIKSSON
STAR COMM	90410	P.CRIGHTON
SUPERBIKE	10KM	T.NEAL
S M/FIELD	829	M.GELDER
S SCANNER	7340	A.DOBSON
T FIGHTER	2980	T.NEAL
TAPEWORM	168515	A.DOBSON LEVEL 1
TAPEWORM	150500	A.DOBSON LEVEL 9
T ZONE	7610	P.ERIKSSON
TOADO	107549	N.GOODING
TURBO	23030	M.GELDER

* Denotes New High Score

We haven't had too many High-Scores over the past couple of month's, so come on, you've had all the winter to better these, let's see a really big effort to see if you can beat the big names in games playing.

YOUR LETTTERS

Help Lines

John Hodgson has sent us a tape and has this to say:

I have a SDX/CPM machine with two 1 meg disc drives. The programs I write make use of the ROM routines, so for a lot of the time I use the computer in the MTX mode. While the computer is in this mode I have two disc drives that I cannot use. What I would like to do is to add the SDX disc drive routines to the MTX mode so that I can use the disc drives via the USER command. I can't just take the 80 col board out as I don't have a SDX system disc, so what I need is a copy of the disc routines that are held in memory.

There are two ways that I can do this. Can you please ask if there are any MOC members with the standard SDX system who would help me with one or both of the following options.

1) Load and run my SDX500 or SDX512 program in a standard SDX system, any disc size, with the discs powered on so that the disc routines are held in memory. The program will then save the disc routines to tape.

Continued -->

2) Would anyone with a SDX machine with 1Mb disc drives lend me a copy of the SDX system disc.

If anyone can lend me a SDX manual for a few days it would be greatly appreciated. I will refund all costs involved.

J.Hodgson 25 Chesterfield Road, St. Andrews, Bristol 6.

Ed-) I have a copy of John's tape, so, if your interested, give me a quick ring on the 'dog-n-bone'and I'll send it on.

Answers

In answer to Ron Potters question in vol 2 Issue 6 on the Ed/Asm.

Probably the best method of using object code produced by Ed/Asm is to adopt the same style as Ed/Asm itself as I have done in producing FIG-FORTH (see elsewhere!!)

Use the ORG and LOAD directives as described on page 17 of the manual, remember, ORG ensures that all object code generated is referenced from the ORG address so this address must be where you finally select your program to start from. On the other hand the LOAD directive simply

stores this code starting at the LOAD address, any attempt to run your proram in this location would be disastrous unless all jumps are relative!

So, suppose your M/C program is only 1K long then set ORG to 4010H and set load to F000H. Assemble using the 'M' option and you should find that, by examination via the Panel, your program is safely stored at F000H. Exit Ed/Asm and enter MTX BASIC.

Now create a Basic "CODE" line using ASSEMBLE 10 or similar and enter enough DS 254 statements so that it is big enough to accomodate your code. Enter the Panel and use "M" to move your block of code from F000H to 4010H.

Your program is now firmly embedded in the "CODE" line of Basic and may be RUN, SAVED or LOADED as any other Basic program.

Finally, remember that once you have embedded your M/C into the CODE line you must not add Basic lines with numbers lower than your CODE line or results will be unpredictable.

Supplied By Dave Thompson

....also from Dave

The test results printed in the latest issue of the magazine were very interesting, especially the reasons given for the prize award. Being naturally curious I had to run both prime numbers programs myself, just to see what the difference in run time was. I calculated that Liam's version took 27 seconds while John's version took an amazing 8 seconds, the time saving being attributed to the fundamental difference in program design.

One of FORTH's main advantages is supposed to be it's speed so I have applied the same principles as Liam and John used to see if FORTH could do better. Liams version in Forth has run time of 8 seconds and John's version a run time of 4 seconds. Can Pascal do better.

Ed-> I've listed the fast Forth version and my attempt at a Turbo Pascal version, which took just over three seconds

```
1 LIST
SCR # 1
  0 ( PRIME NUMBERS 4 SEC RUN TIME )
  1   0 VARIABLE A
  2   0 VARIABLE B
  3 : PRIME 100 A ! BEGIN
  4 A C@ 1+ 2 DO I B !
  5 A C@ B C@ =
  6 A C@ B C@ MOD
  7 OR 0= IF LEAVE THEN LOOP
  8 A C@ B C@ = IF A C@ . THEN
  9 A C@ 1 - A !
 10 A C@ 2 < UNTIL
 11 7 EMIT ;
```

```
PROGRAM PRIMES;
VAR
  J,N: INTEGER;
LABEL 10, 40;
BEGIN
  N :=100;
10:
  FOR J := 2 TO ROUND(SQRT(N)) DO
  BEGIN
  IF (N/J) = INT(N/J+0.001) THEN
    BEGIN
      N := N - 1;
      GOTO 40;
    END;
  END;
WRITE (N,' ');
N := N - 1;
40:
BEGIN
  IF N > 1.5 THEN
    GOTO 10;
END;
END.
```

See '...Another Test' for this month's quiz!!!

Handy Routines

Here is a program that will print out the token table and associated jump addresses. I have adopted it from that published by David Miles (Your Computer, March '84). Please note line No. 230 and the jump addresses from token 193 onwards.

```
100 REM TOKEN-TABLE PEEKER
105 REM BY - JOHN DAVIDSON
110 LET TKNJMP=9975: LET TOKN=127
120 FOR I=9531 TO 9975
130 LET CHAR$=CHR$(PEKK(I))
140 IF ASC(CHAR$)>127 THEN LET CHAR$=CHR$(ASC(CHAR$)-128)+
CHR$(9)
150 IF ASC(CHAR$)<32 THEN LET CHAR$=CHR$(9)
160 PRINT CHAR$
170 IF RIGHT$(CHAR$,1)=CHR$(9) THEN GOSUB 200
180 NEXT I
185 PAUSE 5000
190 STOP
200 LET J=PEEK(TKNJMP): LET K=PEEK(TKNJMP+1)
210 LET JMPADR=(K*256)+J
220 LET TOKN=TOKN+1: LET TKNJMP=TKNJMP+2
230  REM  IF TOKN>193 THEN LET JMPADR=0:  REM  REMOVE  THE
FIRST 'REM' IF YOU WANT THE LINE ACTIVE
240 PRINT TOKN,JMPADR
250 RETURN
```

******

MTX Fig Forth

By
Dave Dulson

This is a review of the disc version of Fig Forth it is similar to the tape based version, except for the number of screens that can be saved and the extra disc commands.

The disc version I received was one of the early ones and so was faulty, you could not write to disc or use the editor commands. If you have a version like this then this is what you need to do, load Fig Forth, then type:

VS 2 <RET>

This will give you the usual 'ready' in the bottom left hand corner then type:

POKE 25000,48:USER WRITE "FORTH",16640,9000

This will solve the problem.

When you receive Fig Forth you get a 25 page manual, this just lists the Forth dictionary or vocabulary as it is called, and gives you a brief explanation of each word. It is not intended to teach you how to use Forth, as far as the documentation goes it still leaves a lot to be desired.

The manual is divided into sections, the first section is the 'Declared Vocabulary'. This contains some 260 words and commands, these include the specific commands which make up another section. The specific commands are the Memotech commands such as ADJSPR and GENPAT, it consists of some 20 words. The next section is the screen editor commands, these are used in the editor mode and are not listed in the vocabulary. There are 26 of these commands in the manual, but the commands X, C and F are not shown. The X command will find a match to what ever text is after it and then delete it, the C commands will copy whatever text follows it into the place occupied by the editing cursor. F will find the string which follows it. The only other section is the error messages, these are similar to the usual MTX error messages in that they are not very 'user' friendly.

Before you can use the disc version of Forth you have to make a file disc, this is a formatted disc that has not been Syscopy'ed, ie a blank formatted disc!. This is used to save the edit screen data on. With disc Forth you can save up to 189 pages which is great deal more than the tape based version will allow.

After loading Forth you have to either change drives or replace the Fig Forth disc with the file disc. There are two modes of use in Forth, the direct mode, ie you tpype in your definition and this is added to the vocabulary, but cannot be saved and the only cursor control is the backspace key. The other mode is the editor, in the editor mode you can save the screens - these are what Forth is written on!!.

Fig Forth on disc works very well, with speeds which are far superior to Basic at 5 - 10 times the speed, it's flexibility for program writing is also very good.

If you're looking for another programming language then you can't go far wrong with Fig Forth.

SUPER-CODER REVIEW

Publisher: Syntax
Price: £7 - £8
Written By: J.Hodgson & D.Threlfall

This is a very interesting program that will compile and thus speed up your normally written programs.

I thought I'd run a couple of simple programs through it, to see if it can attain a Grand Prix image for itself.

The first was a simple For/Next nested loop with a simple addition and print for each count. The results were 370 sec's interpreted Basic, 293 sec's for the Super-Coder compiled version - not too impressive!. Now on with the seat belts, taking the print out improved things beyond belief, my 23 second Basic program ran in under 2 seconds. This 10 times speed increase held out for multiplication, numeric arrays and string arrays, quite impressive all in all.

Even though my programs were quite short at times I had trouble in getting them to compile, this was mainly caused by the compiler only accepting integer (whole) numbers in the range -32768 to 32767 and the restriction of 1 dimensional arrays.

In conclusion, first the good points, the program never bombed out once and speed increases from 2 to 10 times better, mainly nearer 10 times really seems very good. The not so good points, the restrictions as to what will compile, ie anything not integer, 1 dimensional and containing the odd Basic command which has been omitted will not compile, the error messages given will not help in any way to find your problems either.

So, although as far as it goes it's great, I think it would probably not compile 60 - 70% of the programs in the program library. If it could free itself of it's restrictions it would be the best thing since Hovis slicing their bread!!!.

Phil

- -

INTERESTED IN COMMUNICATIONS ??

Paul Wood would like to hear from anyone interested in Communicating via a Modem,RS232 and 300 Baud Bulletin Board. His phone number is 0905 52536.

Ed->See the program library this month for Paul's Terminal Software.

P.s. If their are enough people interested, perhaps the club could get a modem and join in.

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

My original supply of cheap C5 tapes has now dried up, in all over 6 month's I have recorded more than 400 tapes (about 1600 programs). I have now got to buy new tapes, the quality of which seems better, although the others were ok!, the cases have also changed in the hope that they will fair better in the Post. The 20p I have had to add per tape covers only the increased costs of the tape and case, I hope this does not adversely affect the use of the library which is very well supported at present. I have been doing a lot of tidying up of the disc's containing the library programs, the first disc is now properly catalogued with all necessary document files, a READ-ME file and a catalogue file showing the files and their sizes. The whole library should be up to date in a week or so. Four programs from the Swiss user group numbered 43,44,45,46 have been withdrawn due to copyright violations, they have been replaced with others. Look very carefully at the RAM-DISK program if you have a 512 as it really is brilliant.

1.Basic & Assembler Programs

All programs available on cassette, 2 programs per cassette, £1.20 per cassette. Or on disc, £2.50 per disc, please enclose a disc, stating capacity.(Some programs are only available on cassette!!).

Reviews of all programs are available, please send a large SAE. All Swiss User Group programs are prefixed with 'Sw'.

1.Hex-Dec-Bin (Binary Bit In Assembler)
2.CGEN Sprite Generator.
3.3D-Draw Rotate a skeleton of a cup & saucer in 3D.
4.Whist. The Card Game
5.Mem-Save. This Utility will Save a block of memory to tape and retrieve it.
6.MTX-Draw Two basic drawing boards, MTX DB has more
7.LOGO-Draw extensive commands.
8.Simplex Tablaeux. Applications Program.
9.Breakeven. Applications Program.
10.Statistics. Applications Program.
11.An Unsolved Prbm Applications Program.
12.Radio Routines Applications Program.
13.Light Cycles. Arcade Game
14.Hex/Dec/Bin Conversions using USER commands!
15.Renumber II Renumbers Including GOTO's etc
(14 & 15) are Utilities and as such reside high in memory transparent to the user.
16.RELOC Relocs Assembler Properly!!
17.Character Editor Yepp!! Another Sprite Gen!!
18.Quasimodo Excellent Arcade Game
19.Planner YASG (Yet Another Sprite Generator)
20.Hanoi Classic Puzzle (Brilliant simple use of
21.Noble Simple Text Game Graphics)
22.Hi-Lo Just like Bruce's Play Your Cards Right
23.Composer Our First Sound Generator!!
24.Anova Applications Program
25.CASHFLOW Applications Program
26.RenumIII Utility !!!26,27 & 28 cassette only!!!
27.Merge Utility
28.Money Manager Applications program
29.Word Word Processor
30.Reversi Strategy Board Game
31.Full Time Football Manager Game
32.PANEL3 Panel extensions

--- The Second Disc Starts Here ---

33.Texted Word-pro
34.SwMice Swiss Arcade game Written in Basic
35.TNTTIM Assembler arcade game.
36.Sw3D-FUNC.1 First of two. Saturn!!!
37.Sw3D-FUNC.2 Second of Above. Sinpr?.
38.SwSpr-Ed YASG.
39.Sw2-Wand! Number Base Convertion Prog.
40.OXO Noughts & Crosses.
41.Solitaire Strategy Game.
42.Cross-Num Excellent strategy game!!
43.Avoid Seven *** New *** Dice Game
44.Numerology *** New *** Analyse your name!!
45.Chemin *** New *** Another Dice Game!
46.Dice *** New *** Another, Another Dice Game!!
47.SwMathe Arithmetic Tester.
48.Reversi2 Assembler of no. 30. Great!!
49.ISOT *** New *** A really good maze game.
50.DBaseI *** New *** Simple Data Base
51.DBaseII *** New *** Requires MTX Util Tape
52.Money 2.1 *** New *** An update of no 28
53.Ram Disc *** New *** Better than sliced bread
54.RDisc Source *** New *** Source of above.
55.Diary *** New *** Diary & Address program
56.Terminal Em. *** New *** Comm's via Rs232 & Modem.

2.Programs/Procedures in Pascal

(Available on disc. Please provide sufficient postage to cover club costs!!)
Note the compiled Pascal programs will run without the use of Pascal, ie they are .COM programs. All you need is CP/M
1. DBASE for Disc Turbo Pascal
1(a). Comprehensive Create File Procedure
1(b). Simple Display File Procedure
1(c). Add More Data To File
2. Pretty Disney Characters.
2(a). Thumper.
2(b). Bambi.
2(c). Mickey.

3.Articles From Previous Magazines
(Available as listings, please provide sufficient postage to cover club costs. TA!)
1.PANEL2 Utility. An updated version of PANEL1, which includes a second feature.
2.Undocumented Neword dot commands.(Vol1 Iss.7)
3.Hisoft Pascal Review (vol1 Iss.8)
4.Neword Rom Review (Vol1 Iss.5)
5.RST10 Codes Explained (Vol1 Iss.3)
6.VDP Explained Using assembler (vol1 Iss4,5,6)
7.System Variables (Not Previously Published!!)

4.CP/M Programs/Utilities
(!!! Available only on disc !!!, please send in a formatted disc (stating capacity) for each item and enough postage to cover).
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.Reviews
43.Avoid Seven:-By Alan Dobson
A simple game where you play the computer at a game of dice. You role the dice tempting yourself not to roll a seven, if you do the computer wins hands down, when you quit (chicken-out) the computer tries to beat your score.

44.Numerology:-By John Bennett
To quote this programs header page :-
"This program applies the principle of Numerology to analys the names typed in, it will then display a chart of the analysis, which you may wish to note down. On keying RET the reading of the number which represents your name will be displayed, this may also be noted for future reference".
Ed->A really good program into which a lot of effort has been put.

45.Chemin:-Alan Dobson
Another dice game, this time involving 4 dice, again it's you against the computer. The screen is split up well into a part containing dice and a part with scrolling text. This is a good game which is more complicated than Avoid Seven and quite a bit more interesting.
46.Dice:-Alan Dobson
This is a simple dice game, in which you bet on whether your dice will be Higher, Lower or Equal to the computers dice. You start with £100 and can bet a max of £25 per go, until you run out of money that is!!.

49.ISOT:-Alan Dobson
You belong to a rare breed of man, one who is prepared to risk life and limb to obtain untold riches. To do this you are placed on a planet in your special rocket tank. It's up to you to find the treasure ... Are you man enough for such a task??. A must!!

50 & 51.DBase:-MOC
Two programs that were written as articles in the MOC magazine over about 5 or 6 issues. Designed to hold about 100 records, the programs are the same except number 51 requires the MTX Utilities tape for it's data Save/Load.
52. Money V2.1:-Mike Pike
An update of number 28, all of the previous bugs have been ironed out.

53.Ram-Disk:-Eric Roy
You ain't going to believe this one!!, for a 512 or expanded 500 it turns the 16k of extra ram in memory page one into a ram disc that can be used in either of the following ways:
a) As a ram disc for storing binary data
b) For storing string data suitable for database type programs.
This program comes complete with a set of USER Commands including:-RAMWRITE, RAMREAD, LEN, RSAVE, RLOAD, SORT and many more.
Thanks Eric for a brilliant piece of programming

54.Ram-Disk Source:-Eric Roy
This is the Ed/Asm source for the above program.

55.Diary:-Sorry I've mislaid the author!!
This is a memorandum of diary and addresses, it makes extensive use of NODDY and as such is really very good. For the diary you have a full year calender displayed one month at a time. The address book works similarly on an alphabetical index, you can view, amend, delete and save with both diary and address book.

56.Terminal Software:-Paul Wood
A short but comprehensive assembler listing that enables the MTX (via a modem) to communicate with the outside world, using 300/300 Baud Bulletin Boards.

Abridged Terms & Conditions (Downloads)

Disclaimer

www.primrosebank.net, (***the website***) is provided by Dave Stevenson as a service to the public, is provided "as is" and carries no warranties, expressed or implied, of any kind.

Dave Stevenson is not responsible for, and expressly disclaims all liability for, damages of any kind arising out of use, reference to, or reliance on any information contained within the website or made available for download. Whilst the information contained within the website site is periodically updated, no guarantee is given that the information provided on the website is correct, complete, and up-to-date.

A number of articles on the website contain technical data and practical guidance which may be of use in testing and maintaining various items of vintage computer and electronics hardware. Such articles are not intended to cover all aspects of the tasks involved and may omit essential information, including necessary safety precautions. Performance of the tasks described may risk damage to equipment and/or people. The reader is responsible for ensuring that he/she is capable of performing the tasks described and well as assessing the inherent risks involved and taking appropriate measures to mitigate such risks.

Dave Stevenson expressly disclaims all liability for, damages to equipment or injury of any kind arising out of use of such technical data and guidance.

As far as possible, information included on the website from other sources has been credited to the respective author and/or publisher. The majority of content on the website is derived from material first published in the 1980s. ***This material is likely still under copyright of the original author and/or publishers.*** The authors and/or publishers may not have given express permission to copy, transmit or make this information available for download, but I believe that they would have no objection to this archive information being placed into the public domain.

However, should the author and/or publisher of the original material find any content on the website for which they wish to assert their rights, they should notify Dave Stevenson (by e-mail to: webmaster@primrosebank.net) who would be pleased to enter into a dialogue to agree a satisfactory resolution of their concerns.

If you obtained this file as part of a paid-for package, you have been scammed! I suggest that you request a refund from the seller, please also advise Dave Stevenson at the e-mail address above.