Vol 4 - No 6.   February 1991.

FORMAT

FOR SPECTRUM AND SAM USERS

DOS
NEW ~~LAMPS~~
FOR OLD

-JON-
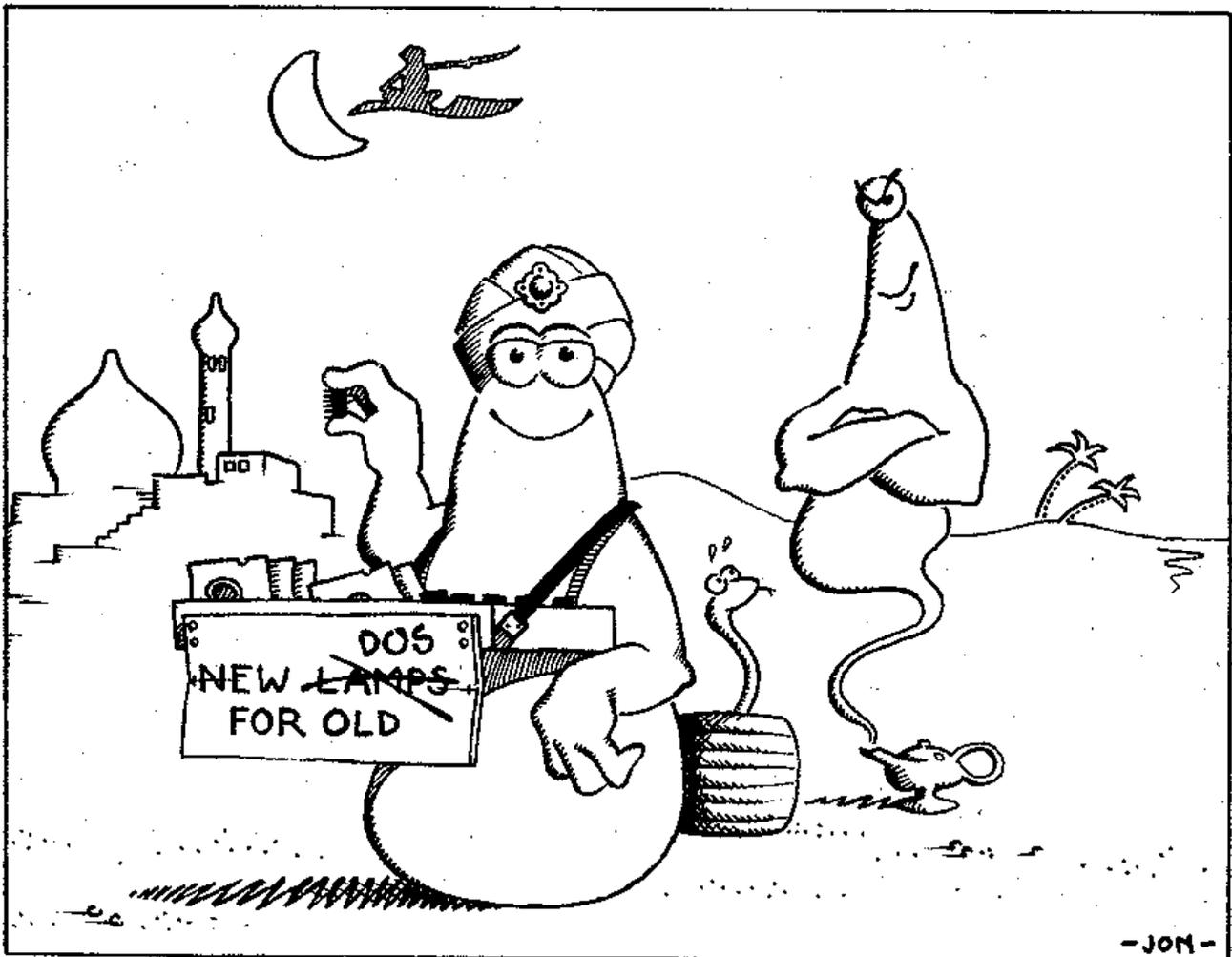
UNI-DOS and MASTERDOS
the FIRST Reviews

UNIDOS The incredible new DOS from Steve Warr for the PLUS D and DISCIPLE. Random files. Sub directories. Hundreds of files on one disk. Hidden files. Protected files. Copy any file of any length without reseting. Openfiles on both drives at the same time. Read&write the same file via different channels at the same time. Redirect stream 3 to disk. Incredibly versatile screen dump routines will print any part of the screen any size you want. Error trapping that works. Be able to add new functions and commands. Many more features. Special introductory offer £5.00 off if your order is postmarked before March 1. **State if PLUS D or DISCIPLE and disk size required when ordering.**

Disc Backup Utility An easy to use program that will copy any type of file and of any size. Allows file names to be changed during copy and wild characters in both source and destination file names. Also has archive features and automatic overwrite of duplicate files. Worth buying just to copy your snapshots! Cassette for both DISCIPLE & PLUS D

File converter Many programs on the DISCIPLE & PLUS D create MD Files. (eg OCP assembler) File converter will change them into ordinary files (and back again if you wish) to allow your own programs to access them. Cassette for both DISCIPLE & PLUS D

Hackers Workbench still the most versatile snapshot hacker available for the experienced programmer. Cassette for both DISCIPLE & PLUS D.

Incremental Backup Utility. Now completly re-written. Make exact copies of your disks with IBU. Only the files that have been added or changed will be copied. Cassette for both DISCIPLE & PLUS D

SAMIBU. Copy a full 780K disk with only 4 disk swaps on a 512K SAM. And even more amazing for the same price as IBU but supplied on 3.5" disk.

SPECMAKER A realy useful Spectrum emulator for your SAM. All the extra SAM keys work in Spectrum mode. Automatically uses SAMs printer port and allows up to 360K of Spectrum programs to be held in SAM memory as a RAM DISK. Programs do NOT have to be changed to make use of the RAMDISK, SPECMAKER does it for you by changing the Spectrum ROM code. Can use Spectrum, microdrive or PLUS D SAVE/LOAD syntax. Many programs that only use cassette are now far more useful as they will work with RAMDISK. PLUS D disks can be loaded into the RAMDISK and saved to SAM disk. Includes the RDUTIL utility to allow peek and poke of any file in the RAMDISK. Supplied on 3.5" disk

Now available version 2.8 Many of the requested features have been added such as loading from drive 2, Returning to Spectrum mode without clearing the RAMDISK etc. V2.8 **can even read dual density OPUS disks on your SAM.** As a limited special offer to existing users. Return your original Specmaker disk and I will send you a version 2.8 upgrade package for only £5.00 inc postage.

SPECMAKER is constantly being improved and is the simplest Spectrum emulator for SAM. 1000s of programs work without the need for any conversion. Most other programs need only minor changes.

	PRICES	INDUG MEMBERS	NON MEMBERS
S D Software.	UNIDOS	£19.95	£25.95
70 Rainhall Road,	SPECMAKER	£ 9.95	£12.95
Barnoldswick,	IBU/SAMIBU	£ 3.50	£ 4.90
Lancashire,	DBU	£ 4.00	£ 5.50
England.	FILE CONVERTOR	£ 4.50	£ 5.30
BB8 6AB.	HACKERS WORKBENCH	£ 8.50	£ 9.90

POSTAGE £0.50
(£1.20 OUTSIDE UK)

DO NOT FORGET YOUR INDUG
MEMBERSHIP NUMBER

CONTENTS

Vol.4 No.6.   February 1991.

NEWS ON 4

SAD LOSS TO THE ZX WORLD

News reached us just a little too late for the January issue that Mike Johnston - mastermind of the famous ZX Microfairs - has died. Mike, who was 47, suffered a heart attack on Monday 17th December.

During the early and mid 80s Mike was one of the most widely known people in the home computer industry. At their height his shows were among the biggest in the land. The ZX Microfairs attracted Sinclair enthusiasts from all over the world and Mike was always in personal attendance wondering round his shows to see all was going well.

Our deepest sympathies go to Mike's family, he will be sadly missed.

DISCOVERY DISCS ON SAM

S.D.Software have just announced a new version of their highly successful SPECMAKER program for the SAM Coupé.

The new version, 2.8, contains many extra features including the ability to load files from 720k Opus Discovery discs. Most Discovery users who have moved in SAM's direction have had to suffer the hassles of tape transfer to their Coupé. Now SPECMAKER will do much of the work for them.

For more details see S.D.Software's advert in this issue.

EXTENDED GUARANTEE FOR SAM

P.B.T. Electronics are currently mailing SAM owners with details of their new extended guarantee insurance service. As early Coupés are now out of their original guarantee many will want to obtain cover against faults and obtain the peace of mind the the new scheme gives.

Full parts and labour cover is given for just £17.95 for a tape based SAM or £23.95 for a SAM Coupé with single drive. Rates are also available for extra drives and all the interfaces that go with SAM.

PBT are using the MGT mailing list which, for many reasons, does not contain every owner. If you are interested in the insurance scheme and you don't get a letter from PBT then send them a stamped addressed envelope (9"x3") and they will send you all the details.

MUSICAL EDITIONS

Two new editions of the Music Writer program for the 128K Spectrums have just been released.

The **Reference Edition** is really just the old program but with the Utility pack now included at an all-in-one price of £25.

The **Muso's Edition** is identical but drops the detailed manual in favour of an A5 Icon glossary. It's intended for those who are already familiar with music notation/terminology. Replacing the manual with the abridged booklet allows the program to sell for just £15. You can still purchase the full manual later if you find you don't know as much about music as you think.

Both versions are available from Garry Rowland, P.O.Box 49, Dagenham, RM9 5NY. Add £2 p&p per order.

SAM GOES OVERSEAS

SAMCO seem to be doing very well overseas at the moment. In Poland, France and Czechoslovakia companies are all showing great interest in the Coupé and the is a growing chance that SAM may be manufactured behind the old Iron Curtain for local sales.

Alan Miles, who is heading the sales drive, has always put great store in overseas sales and is also looking at potential markets in southern Europe.

URGENT we need your news. New Releases, Club meets, Shows, anything you think other people should know about. If you have any news items you want to pass on then send them in. Please mark the envelope NEWS in the top corner.

Lots to get through this month so straight down to business. Staff problems here continue to cause headaches but I'm glad to say interviews are now going ahead very well so by the end of February everything should be back to normal.

The demand for the FORMAT binders has far exceeded my expectations, the first batch were sold out before they even arrived here. A new (and larger) batch has been ordered and as soon as they come in we will dispatch them. Provided you only ordered binders then your cheque will not have been cashed unless you were one of the lucky ones who got in quick.

On the subject of orders, there still seems to be many people ordering from the FORMAT Readers Services page who forget to add the postage. If we can adjust the order (by dropping an item or two) then this is what we normally do, but it causes extra work which could so easily be avoided. The only time we include postage is in the advert for the Sam Parallel Interface. For UK members this is post free along with the printer cable if ordered at the SAME TIME. If you just order the cable you must order from the FRS page and include the postage.

While talking about postage I would like to point out to members that all orders are dispatched with the next available issue of FORMAT (with the exception of the SPI). Remember to allow about ten days for your order to reach us and to be processed. If you receive your usual copy of FORMAT then it shows that your order did not arrive in time. If however nothing turns up then UK members should contact the office, by phone between 10am and 4pm, so we can check the computer for you. If we find that the order was sent out OK you will be given guidance on how to pursue the matter with the Royal Mail. Overseas readers should write to us, enclosing a copy of their order.

And finally. This months issue contains reviews of the two new DOS systems that have just been launched. MasterDos for the SAM Coupé is what the machine has really been waiting for - a powerful DOS that can help to exploit the machines potential. Uni-Dos for the DISCiPLE and PLUS D will also prove a real eye-opener for Spectrum users and finally unites both interfaces with one DOS. Many people have already phoned me to ask 'should I upgrade?' Normally I don't like advising people on software because, like someones choice of car, many things are not clear-cut.

However in this case I do have a very firm view. If you own a disc based SAM then MasterDos is a MUST. Beg, borrow or even earn the money to get your copy - without it you will never be able to exploit your SAM to the full. When it comes to Uni-Dos things aren't quite so easy because, for a while at least, there may be some incompatability problems with commercial software. Most software will work OK but there are bound to be some items that cause problems just as there were when the disc systems were launched in the first place. Having said that, Uni-Dos is a major advance and I would recommend any serious user to order their copy soon, the extra power is well worth the effort.

Until next time.

Bob Brenchley. Editor.

ALL FORMATS COMPUTER FAIRS

Thousands of bargains! Computers, printers, business software, games, monitors, public domain, shareware, keyboards, media, disk drives, user groups, memory, joysticks, modems, books, magazines and lots more. Huge savings for all computer users.

Saturday 2nd February 1991
Saturday 23rd March 1991

At:
The New Hall of the
Royal Horticultural
Society - Greycoat &
Elverton Street
Westminster
London

Admission: £3.00

Tickets:
Avoid queuing
advance tickets
from: Mike Hayes
8 Midgrove - Delph
Oldham - OL3 5EJ
Tel: 0457 875229
£3.00 each
(Cheque - Visa - Access)

All Formats
COMPUTER FAIR

Tubes:
Victoria - Pimlico
St James Park

**STANDS ONLY
£75**

Book With:
JOHN RIDING
0225 868100
MIKE HAYES
0457 875229

By:- John Wase.

My apologies for my absence last month:- whilst everyone was in the run-up to Christmas, my family had a series of catastrophies which included me, flat on the floor, dosed up with vallium and endomorphines, and taking absolutely no interest in the rest of the shambles: the pain was, as they say, exquisite. My poor wife flung everything into the computer room just half an hour before the relatives came to stay (my back still wasn't right and could barely sit down, even for a meal) and shut the door. I'm still trying to sort out the chaos - my apologies to anyone who's written in recently.

It's rather interesting: once someone's set the ball rolling, often I get a number of bits and pieces coming back. Take the "Print Using" business, for example. This one runs and runs, though it does look as though in some cases we're converging on the same general solution. For instance, Brian Cavers (of Chezron/ Outlet) tried to simplify A.S.Hughes' "Print Using" (in the December 90 issue) to avoid the extra test for zero and came up with:-

PRINT TAB 10- LEN STR$ INT X;X

For Sam. Hang about a bit, the Speccy precedes values between .1 and .999 with a zero, but not below .1, so the Spectrum line is modified to:-

PRINT TAB 10- LEN STR$ INT X+(X<.1);X

And all is well. Ron would be interested if anyone can devise a method that doesn't even need the <.1 adjustment.

Geof Smith of Fleet, Hampshire, has also had a look at Mr. Hughes' routine and noticed that it breaks down for negative numbers. Geoff points out

that a mixture of positive and negative numbers can easily be accommodated by altering line 100 to..

100 PRINT "eight spaces"(1 TO 6- INT (LN(ABS X)/LN 10)*((ABS X)>1)+((ABS X)< .1)-(X<0));X

Geoff explains that the purpose of the extra term (X<0) is to shift printed negative numbers one more space to the left to allow for the minus sign whilst still keeping their decimal points aligned with those of the positive numbers. The blank string is increased by one space to allow for the minus signs.

In contrast, in a different approach, Ettrick Thomson of Aldeburgh, Suffolk, mentions that in the far-from-easy programming problem, Ray Bray's elegant SAM PROC (Oct '90) fails if trailing zeros are needed, as in 1.095 rounded to two decimal places, which should give 1.10; Howard Griffith's routine copes with them. Neither approach copes with negative numbers - not always needed, but it's better if they can be handled.

Here is a SAM PROC (with a test program) and a Spectrum equivalent. In both, a number too big will lead to the routine trying to add a negative number of leading spaces and hence to an error stop. This could easily be detected, of course, but the resultant action would depend on the application.

Note the semicolon after n$ in the PRINT statement, since you will often want to print more than one number on the same line. SAM and Spectrum differ slightly in printing pure fractions: SAM prints, for example "0.05", whereas the Spectrum gives you ".05". This difference is still there in these "PRINT USING" routines.

```
  5 REM ***** SAMUSING *********
  6 REM * Ettrick Thomson 1990 *
 10 INPUT #2;q,
 20 prtusg q,2,7
 30 PRINT
 40 GOTO 10
100 DEF PROC prtusg n,dp,lgth
110 LOCAL n$,p,q
120 LET n$=STR$ (n+SGN n*5/10↑(dp+1)
    ),p=INSTR(n$,".")
130 IF p=0 THEN : LET n$=n$+".",p=LE
    N n$
140 LET q=p+dp-LEN n$
150 IF q>0 THEN : LET n$=n$+STRING$(
    q,"0")
160 LET n$=n$( TO p+dp-(dp=0))
170 PRINT STRING$(lgth-LEN n$," ");n
    $;
180 END PROC
```

```
  5 REM ****** SPECUSING *******
  6 REM * Ettrick Thomson 1990 *
 10 INPUT n: PRINT n,
 20 LET dp=2: LET lgth=7
 30 GOSUB 100: PRINT
 40 GOTO 10
100 LET n$=STR$ (n+SGN n*5/10↑(dp+1))
110 FOR p=1 TO LEN n$: IF n$(p)<>"."
    THEN NEXT p
120 IF p=LEN n$+1 THEN LET n$=n$+"."
130 LET q=p+dp-LEN n$
140 IF q>0 THEN : LET n$=n$+"00000000
    "( TO q)
150 LET n$=n$( TO p+dp-(dp=0))
160 PRINT "        "( TO lgth-LEN n$)
    ;n$;
170 RETURN
```

Are these definitive solutions to the problem? I doubt it.

Sorry, all you Spectrum owners, but here's another little program for SAM. It's called the Bigletters program, and Dave Wood of Cleethorpes wrote it to use in a program to teach his son the alphabet when he realised that CSIZE would go only up to 16 pixels. Okay, you Spectrum lot; you have the last laugh if you've got Andy Wright's Beta Basic program (and I bet quite a lot of you have): Beta Basic's CSIZE goes up to the size of a whole screen or more for just one letter. Here's Dave Wood's routine to help SAM owners out, then.....

```
 10 REM **** Bigletters by Dave *****
 20 REM **** For the SAM Coupé ******
 30 MODE 4: CLS #: CSIZE 6,8
 40 PRINT PEN RND*6+1; AT 0,9;"BIG LE
    TTERS BY"
 50 PLOT PEN 9;70,167: DRAW PEN 9;115
    ,0
 60 REM **A$=MESSAGE UP TO 4 CHARACTE
RS
 70 LET A$="DAVE"
 80 REM S=PRINT LINE
 90 LET S=3
100 GOSUB 160
110 LET S=11
120 LET A$="WOOD"
130 GOSUB 160
140 PALETTE 9,RND*127: BEEP .01, RND*
    30: PAUSE 1: GOTO 140
150 REM SUBROUTINE FOR LARGE CHARS.
160 PRINT PEN 8; AT 21,0;A$
170 FOR Z=7 TO 0 STEP-1
180 LET S=S+1
190 FOR X=0 TO 31
200 IF POINT (X,Z)=8 THEN PRINT AT S,
    X-1" PEN 9 " "
210 NEXT X
220 NEXT Z
230 PRINT AT 21,0;" " :REM 4 SPACES
240 RETURN
```

Carol Brooksbank has written to me. Although she does a lot of graphics work, and her ribbon consumption is therefore high, she's worried by my description of re-inking, and fears the consequences of several hundred feet of inky ribbon let loose on the world. Well, Carol, I did write this from the point of view of a complete naffhead as far as inking's concerned - I guess, like me, you have several ribbons tucked on the shelf: well I wait until I have a number used, then re-ink them all in one go - believe me it's not that difficult. Anyway, for the faint-hearted, Carol recommends Aladdink of 4 Hurkur Crescent, Eyemouth, Berwickshire. They re-ink an FX80 ribbon for £1.90 much darker and longer lasting than when it was new. Moreover, when Carol broke one recently, they changed the ribbon in the existing cartridge - and they'll reink coloured ones, too.

Now, back to the Spectrum. L. G. Baumann of Cowies Hill, South Africa, has written again, (bless him) with a

real shortie; the sort of thing I'm looking for all the time. If you're programming in Basic, It's rather nice to be able to add something like this to your program...

```
10 REM ** New Bingo Program**
20 LET d=23: LET a$="abcde"

 0 REM **********************
 0 REM BY - your name
 0 REM **********************

30 PRINT "To play this game..."
```
etc.

It's also nice that the writer's credit is given some protection, since you cannot bring the zero line numbers into the editing area until the original line numbers are restored.

The three credit lines must be the third, fourth and fifth lines in the program and must initially be numbered 21, 22 and 23. Then run line 9999 of the program to insert the zeros or line 9998 to reinstate the original line numbers. A nice effect can be achieved by adding a few spaces at the end of lines 20 and 23 to print the two empty lines as shown in the example. Note that line 9999 is the same as 9998, except that the numbers 21, 22 and 23 are changed to zeroes.

```
9998 LET z=1+PEEK 23635+256*PEEK 23636
: LET z=4+z+PEEK (z+1): LET z=4+z+PEEK
 (z+1): POKE (z-1),0: POKE z,21: LET z
=4+z+PEEK (z+1): POKE (z-1),0: POKE z,
22: LET z=4+z+PEEK (z+1): POKE (z-1),0
: POKE z,23: LET z=4+z+PEEK (z+1): STO
P: REM to reinstate lines

9999 LET z=1+PEEK 23635+256*PEEK 23636
: LET z=4+z+PEEK (z+1): LET z=4+z+PEEK
 (z+1): POKE (z-1),0: POKE z,0: LET z=
4+z+PEEK (z+1): POKE (z-1),0: POKE z,0
: LET z=4+z+PEEK (z+1): POKE (z-1),0:
POKE z,0: LET z=4+z+PEEK (z+1): STOP:
REM to zero lines
```

Great! Many thanks Mr Baumann.

Now a couple of short programs by Mattias Danielson of Sweden, who also appeals for contacts - his address is in the second listing. Mattias claims his first program generates a fractal with an infinite number of triangles. I'm not so sure it is a fractal Mattias, but it's a nice little program, and though it's written for SAM is easily adaptable for the Spectrum. Here's the listing...

```
 1 REM ********************
 2 REM *     Fractally    *
 3 REM *M.Danielson  901102*
 4 REM ********************
10 MODE 4: LIST FORMAT 2
20 LET x=0,y=0,yos=-18: DO
30 ON INT (RND*3)+1: LET x=64+x/2,y
   =95+y/2: LET x=x/2,y=y/2: LET x=
   127+x/2,y=y/2
32 LET p=RND*6+1: IF p<POINT(x,y) T
   HEN LET p=POINT(x,y)+(1 AND POIN
   T(x,y)<7)
40 PLOT PEN p;x,y
50 LOOP
```

Mattias calls his second program (which is also for SAM) a "kind of vector graphix program: a flicker-free rolling three-sided pyramid". Actually, I don't find it particularly flicker-free, and not very quick either (possibly because it doesn't use BLITZ). But it's a good start. I'm sure that it would work better with some improvements from out there... How about it, folks? Programming in Basic; not more than 10% longer than Mattias' version...

Here is his version:-

```
 1 SCREEN 1: MODE 4: CLOSE SCREEN 2
   : OPEN SCREEN 2,4
 2 CLS #: PALETTE 7,0: LIST 22: FOR
   i=0 TO 11: PRINT AT i,3;"
   ": NEXT i: PALETTE 7,127
 3 PRINT AT 16,1;"Please wait while
   calculating."
 4 LET x=127,y=108,s=3,b=15,n=50,m=
   0,f=9: DIM p(2,230)
 5 FOR f= 1 TO 230: LET p(1,f)=SIN
   (f/20),p(2,f)=COS (f/20): NEXT f
   : REM         Increases Speed wi
   th 43%
 6 PRINT AT 16,0;"         -  Press an
   y key.         ": PAUSE
 7 DEF FN y(a)=30+p(1,a)*b+m
 8 DEF FN x(a)=127+p(2,a)*n
 9 DO
10 LET f=f+(126 AND f<9)-(126 AND f
```

```
      >135)
 11 LET f=f+s,f1=f+42,f2=f+85
 12 PRINT AT 0,0;"Y:";y+18;" B:";n;"
    BY:";m+33;" S:";s;" N:";b
 13 LET y=y+(3 AND INKEY$="q")-(3 AN
    D INKEY$="a")
 14 LET b=b+(3 AND INKEY$="r")-(3 AN
    D INKEY$="f")
 15 LET m=m+(3 AND INKEY$="w")-(3 AN
    D INKEY$="s")
 16 LET n=n+(3 AND INKEY$="e")-(3 AN
    D INKEY$="d")
 17 LET s=s+(2 AND INKEY$="t" AND s<
    9)-(2 AND INKEY$="g" AND s>-9)
 18 IF f MOD 2=0 THEN DISPLAY 1: SCR
    EEN 2: CLS : ELSE DISPLAY 2: SCR
    EEN 1: CLS
 19 PLOT x,y: DRAW TO FN x(f),FN y(f
    ): DRAW TO FN x(f1),FN y(f1): DR
    AW TO FN x(f2),FN y(f2): DRAW TO
    FN x(f),FN y(f): PLOT FN x(f2),
    FN y(f2): DRAW TO x,y: DRAW TO F
    N x(f1),FN y(f1)
 20 LOOP
 21 REM Hey- SAM Owners out there, d
    rop me a line.....
 22 REM 19901130
 23 REM By Mattias Danielson
 24 REM Storgatan 89
 25 REM S-51600 Dalsjofors
 26 REM Sweden.
 27 REM
 28 REM Controls:-
 29 REM  Q,A    Top Y-Coordinate
 30 REM  W,S    Base Y-Coordinate
 31 REM  E,D    Base Width
 32 REM  R,F    Viewpoint
 33 REM  T,G    Speed
```

Many thanks, Mattias.

Finally, another item from Mr Baumann, which I have not had time to test. It's for people who need to transfer files written on Tasword 3/128/+2 (all of which produce OPENTYPE files) to Tasword 2 (which uses CODE files. The program converts files to TW2's 64 character line format. If your file is more than 64 characters across reformat it before conversion. If it is then more than 300 lines, split it into pieces, otherwise it won't fit into Tasword 2. Once converted to TW2 format you could then load the files into the SAM version of Tasword 2.

Here's the program.

```
   1 REM *** T3T2 *** L.G.Baumann ***
   5 REM When the "End of file" messag
     e appears save the new TW2 textfi
     le by entering "GOTO 1000"
  10 CLEAR 31999: LET n=32000
  20 CLOSE #12: LET skip=0: LET scrn=0
     : LET thecode=0
  30 PRINT AT 4,0;"Name of file to be
     converted to Tasword2? ";: INPUT
     a$: PRINT " - ";a$
  40 PRINT AT 10,0;"Do you want the te
     xt to appear on the screen    Y
     /N ?"
  50 IF INKEY$="Y" OR INKEY$="y" THEN
     LET scrn=1: GOTO 80
  60 IF INKEY$="N" OR INKEY$="n" THEN
     LET scrn=0: GOTO 80
  70 GOTO 50
  79 REM modify the next line to suit
     your disc system
  80 CLS : OPEN #12;"m";1;a$
  90 LET thecode=CODE INKEY$#12
 100 IF thecode=13 AND skip=0 THEN POK
     E 23692,255: GOTO 160
 110 IF thecode=13 AND skip=1 THEN GOT
     O 180
 120 IF thecode=10 THEN GOTO 90
 130 POKE n,thecode
 140 LET skip=0: GOSUB 190
 150 LET n=n+1: GOTO 90
 160 IF INT (n/64)=n/64 THEN LET skip=
     1: GOTO 90
 170 POKE n,32: GOSUB 190: LET n=n+1:
     GOTO 160
 180 FOR b=1 TO 64: POKE n,32: GOSUB 1
     90: LET n=n+1: NEXT b: GOTO 90
 190 IF scrn=1 THEN PRINT CHR$ PEEK n;
 200 IF scrn=0 THEN PRINT AT 0,0;"wait
     :  ";"address= ";n
 210 RETURN
 999 REM Don't forget CLOSE #* if it's
     a Disciple or +D
1000 CLOSE #12: CLS : INPUT "Name for
     saving TASWORD2 file?";b$
1009 REM Adjust for your disc system
1010 SAVE *"m";1;b$ CODE 32000,n-32000
```

That's all for now. Many thanks for all the goodwill messages. Please keep the short snippets coming in to me:-

John Wase,
Green Leys Cottage,
Bishampton,
Pershore,
Worcs, WR10 2LX.

UNI-DOS

For The DISCiPLE & PLUS D

By:- Ken Elston.

It is said that everything comes to those who wait. I first heard of Uni-Dos over two years ago, even before it had a name. At that time it was to be a PLUS D overlay to provide random file access. It has grown quite a bit since then.

Although I only have a pre-release version I am quite impressed.When Uni-Dos arrives you get a disc, a fairly comprehensive 32 page manual, and a ROM chip. When you order you need to state the type of disc (5.25" or 3.5" / 40 or 80 tracks) and whether you have a PLUS D or a DISCiPLE. I have both and so have two ROM chips.

At first I was a bit nervous about changing the ROM chips as the manual keeps saying "carefully" and giving grave warnings if you get it wrong. But apart from having to get the chip legs in line, they were wider than the socket, it was dead easy. The instructions are really easy to follow and I think it would be difficult to get it wrong.

Once this was done there is a set-up program to run. The disc part is very easy but there seems to be no end to the printer options. Lucky for me all the defaults are for EPSON printers and so I didn't need to change any. The extra-nice thing for someone like me who has both a DISCiPLE and a PLUS D is that only one system file is needed on a disc to work with BOTH interfaces.

Now what do you get for your money? All the old GDOS commands are there, although everything related to the DISCiPLE networking has been removed. Many of the commands now work faster especialy some of the multiple commands like ERASE d1"*" and many of the annoying bugs of GDOS have been fixed. For example you can now do a CAT and send it to a file on disc.

One of the biggest changes is the copy command this is no longer SAVE D1"*" TO D2. It has been replaced with the MOVE command. The syntax is otherwise unchanged but there is quite a difference in how it works. You now get two lists on the screen showing which file is being read and written. But the best thing is when it has finished the Spectrum does not crash. Your program is still there so you can now safely copy files within a program. Also ALL types of file can be copied and they can be of any length. This is because Uni-Dos uses the space between RAMTOP and STKEND to do the copy. It does mean that you can get an OUT of memory error when you try to copy if there is no space left. It also means that the larger your Basic program (or the lower you set RAMTOP) the less room Uni-Dos has to hold sectors from the files it is copying so the slower the job becomes.

The MOVE and SAVE commands can be written as SAVE OVER etc, this will overwrite an existing file without waiting for you to reply Y or N. Again very useful from within programs that save files.

There is a new version of the LOAD command which allows you to LOAD p"file". This works just like the LOAD Pn command except you do not need to know the number. This makes loading snapshots real easy just type LOAD p"SNAP f" and it loads and runs.

The next thing the manual tells you about is OPENTYPE files. These can be used in exactly the same way as on GDOS. But now you CAN be writing files on two discs and you CAN open more than one channel to the same file. If you open the file OUT it is written just the same as a GDOS OPENTYPE file

but if you open it IN then you can move around in the file by using the new POINT command. This gives true RANDOM ACCESS. The most impressive thing I could do with OPENTYPE files was to open a file RND. This lets you read and write the file and move about using POINT. I thought I could fool it by opening #4 and #5 to the same file and then write something to #4 and read it back with INPUT #5. IT WORKED!

You can find out where you are in a file by LET x=(POINT #n) and get the length of a file by LET x=(LEN #n). The brackets are necessary because of the way the Spectrums line scanning system works.

The CLOSE command has been improved so you can now use it to close channels of type S K and P, the standard Spectrum channels. You can even do silly things like OPEN #2;d1"screen" OUT and OPEN #0;d1"keys" IN. I can't think of any real use for this but its there.

Apart from the small and large screen dump you can now dump any part of the screen and make it any size by using the new dump command SAVE SCREEN$ # this takes upto 7 parameters. I played about with it for hours. Also the all black small dump problem which annoyed many GDOS users doesn't appear. The large screen dump actually uses the new dump command and by poking new values into the DOS with POKE @ you can get any type of dump from the snapshot button. A number of things have been improved with the printer. The margin and line spacing can be changed at any time and it works as you would expect as the margin is sent at the start of the line not after the end of the previous line as GDOS did.

Also two new features have been added to the snapshot routines the first is "P" which re-sends the printer initialisation codes which is very useful if you have the printer set up different from normal and you have to switch it off and back on for any reason. The second feature is that if a snapshot does not work, if the disc is full or not ready for example, then you are returned to the striped border so you can try again form exactly the same point in the program.

Possibly the biggest change over GDOS is that you can now have sub-directories. Each sub-directory can hold from 1 to 99 files its size being specified when it is created. A sub-directory can have sub-directories within it and so on. I made 100, each inside the other, before deciding that it really did work. This means that you can now have hundreds of files on a disc in fact over 1000. You use the / character to separate a directory from a file name so you can have such things as LOAD d1"wordproc/letters/solicitor/moore5". Unfortunately you can not use the wild-cards (* and ?) in the directory names.

You can select the current disc by doing an IN D1 or IN D2 and set the current directory by IN D1"<directory>/" you do need the / at the end. For example IN d1 puts you in the root directory of disc 1. IN d1"wordproc/letters/" puts you in the letters sub-directory which is in the wordproc sub-directory on disc 1. IN d1"solicitor/" would move you down into the next level. IN d1".." would move you back up one level and IN d1 "/" takes you right up to the root directory.

You can open a channel to a directory so that you can read through it without having to use LOAD @ all the time. You can also open the entire disc as a single random access file. Both these feature should allow people to very quickly write utility programs that manipulate disc files in fantastic ways (I look forward to seeing some in FORMAT).

An error trapping command has been added so you can say LINE n and, when any error (including disc and 128K errors) occurs, the program does a GOTO n. You can find out what the error is using LET err=(PEEK @99) and when the error is fixed return to the line giving the error by CONTINUE.

This is nearly as good as SAMs error trapping.

As well as the new directory file-type there are also CREATE files. I'm not sure how they work but they add even more functions and commands to Uni-Dos. There are three on the disc. The first, ext_code adds the command FORMAT d1"name" to name a disc, which is printed by the CAT command and two functions. (LINE) to tell which is the current drive and whether there is a disc in it and if it is write protected. (STR$ #n) to find out what type of channel is connected to a stream S K P D or d.

Dir_code gives two commands to create and remove sub-directories and a function to find the pathname of the current directory.

Pcat_code allows CAT p to give a desktop type representation of a directory using icons. (This is not mentioned in the manual and may not be part of the final version)

There are quite a number of other changes that while only small when totaled make Uni-Dos a very powerful product. Not least the fact that you only need one system file for both PLUS D and DISCiPLE.

I suppose that it was to save space that the disc FORMAT routine was taken out of Uni-Dos and made into a stand-alone program but that is no real hardship, in fact most other computers have format routines that are not part of the actual DOS. The format routine does something to discs that enables Uni-Dos to tell what disc you have in. So if you are copying from one disc to another and you put the wrong disc in the drive by accident Uni-Dos knows and tells you. However, you can re-format any disc and not lose any of the files on it.

There are a number of small changes that you have to look for such as hidden and protected files. ERASE can be made not to give an error if there is no file to remove. MOVE will copy files on a disc to the same disc in the same drive without asking you to change discs, this is used when you are copying files from one directory to another. There are quite a few more.

On the other hand there are problems with a number of the programs that I use. Any program that loads into the PLUS D's RAM area or which makes calls direct to the GDOS ROM just do not work. I believe that there are only 28 bytes left in the PLUS D RAM so there is not much room for things like PLUS D HACKER. I have found a number of programs that work on GDOS that do not work on Uni-Dos but I would think that conversion programs will soon be made available.

Uni-Dos is available from S D Software. See their advert elsewhere in this issue. INDUG members can get a discount if they quote their membership number

* * * * * * * * * * * * * * * * * * * *

I've no idea whether there's life out there, Cecil — I just wish there was more in this bed!

"I hope this hasn't started something."

14

MONEY MANAGER

By:- Carol Brooksbank.

This month we start to fill in the main menu options. You now have enough of the program, if you want to try it out, to set up your calendar and variables, though, of course, you can only view or print the calendar and save it. You cannot yet change it or make any use of it. If you decide to do that, the program with the data will be saved as "ACCOUNTS", and eventually we shall save the loader as "MONEY" or "AUTOMONEY" - depending on whether you want the program to auto-load from a disc - so continue to save this listing that you are still developing as "MANAGER", so that they do not overwrite each other.

But in case you want to test the program, we need to have some way of returning to BASIC to add more lines, so we shall begin, in an upside down way, by writing the last option first.

Main menu option 9 is the EXIT TO BASIC option. The last line of the main menu BASIC which we added last month was:-

GOTO 1000*opt

so, OPTION 9 BASIC starts at program line 9000.

```
9000 CLS : PRINT AT 3,11; PAPER 6; PE
     N 0;"CAUTION"
9005 PRINT AT 6,13; "USE"
9010 PRINT
9015 CSIZE 8,16: PRINT : PRINT : PRIN
     T AT 4,10; PAPER 7; PEN 0;"GOTO
     MENU"
9020 CSIZE 8,8
9025 PRINT : PRINT : PRINT AT 12,3;"T
     O RE-ENTER THE PROGRAM": PRINT A
     T 16,0
9030 presskey
9035 POKE SVAR 615,2: SCROLL RESTORE
     : STOP
```

This is a very simple option, which first prints a warning on screen that you must use GOTO MENU to get back into the program. This is because the calendar and all your financial information are held as DATA arrays or variables. RUN would destroy them.

In line 9035 the statement POKE SVAR 615,2 returns the keypad to normal function keys. While we are within the program, these are a keypad so that they can be used for entering sums of money quickly. If SVAR 615 holds an odd number, the keys are a keypad. POKEing it with any even number restores the function keys.

The next command in line 9035, SCROLL RESTORE, enables the 'Scroll?' message, which we had turned off with "SCROLL CLEAR" so that the screen display of the calendar runs straight through without pausing. After restoring these two items to normal state, the program uses STOP to exit to BASIC.

At the moment, none of the other main menu options have been written, so if you selected their numbers at the main menu, the program would run straight through to 9000 and exit to BASIC. This may not be what you want, so for the time being, enter the following lines:-

```
1000 CLS: wrongoption: GOTO menu
2000 CLS: wrongoption: GOTO menu
3000 CLS: wrongoption: GOTO menu
4000 CLS: wrongoption: GOTO menu
5000 CLS: wrongoption: GOTO menu
6000 CLS: wrongoption: GOTO menu
7000 CLS: wrongoption: GOTO menu
8000 CLS: wrongoption: GOTO menu
```

If any unwritten option is selected now, the wrongoption procedure at 10855, which we wrote last month, will be called and the program will then return to the main menu. These lines will be overwritten as we go along.

We begin now with main menu option 1, the PAY DAY option. The convention will be that every program option will return you to the menu from which it is called, and option 9 of every menu will return you to the previous menu.

```
11115 DEF PROC statement
11120 LPRINT "Assigned money - set asi
      de for paying bills: £";assign
11125 LPRINT : LPRINT "Savings - for e
      mergencies and occasional purcha
      ses: £";savings
11130 LPRINT : LPRINT "Free money - av
      ailable for day-to-day expenses:
      £";cash
11135 LPRINT : LPRINT "TOTAL FUNDS: £"
      ;assign+savings+cash
11140 END PROC
```

Unlike option 9, the PAY DAY option has its own menu. In line 1095, you see that its options will reside at 1000+(100*opt)

This means that PAY DAY menu option 1 will reside at 1100, so the menu must fit into the lines between 1000 and 1099. This is a fairly short menu, with only three options plus exit, so we have no difficulty, but at some menus we shall have to use more multi statement lines and shorter steps between lines to fit everything in.

Incidentally, if you want to make it easier to find your way about the listing, you can use embedded INVERSE codes in the REM statements. Press the INV key before you type REM, and press SS/INV before using RETURN to enter the line. This will make the REM statements appear in blue lettering on a white background, and they will stand out from the rest of the listing.

This menu starts, like every menu, by being labelled with its number. This is menu3 (main menu page 1 is menu1, and page 2 menu2). The screen is cleared, and togflag reset to show that the funds display is not present at this moment.

The point at which the option is chosen, line 1065, has its own label, men3 . You will remember from last month that toggle returns to this point after displaying or wiping out the funds display. Line 1080 sets menno to the menu number and calls toggle if D or E are selected. The other lines between 1065 and 1090 ensure that if any keys other than 1,2,3 or 9 are pressed, wrongoption will be called and menu 3 displayed again. This applies to the letter keys, to SPACE, DELETE, EDIT, RETURN and so on. The only key which will respond other than a valid menu key is ESC, which would break out into BASIC - but without the warning message of main menu option 9.

I mentioned earlier that the function keys, even though they are acting as a keypad and returning numbers, will not be recognised at the menus. GET opt fetches a key number, not what it returns, which is why it will recognise A as 10, D as 13, E as 14 etc. GET opt$ would return letters, but would still not make any sense of the keypad numbers. Unlike GET, INKEY$ would recognise keypad numbers as numbers because it returns the code of the key pressed, but it is less convenient at menus because a line is needed which will wait until you press the key.

Every menu has a set of lines similar to these, to ensure that only its valid option numbers may be selected.

Every time we enter a new menu, the first of its options we shall type will be option 9 - the one which returns to the menu which called it. In this case, menu 3 is called from the main menu which gives us the line:-

```
1900 GOTO menu
```

PAY DAY option one is a very simple one, used for changing the amount put into the savings fund on pay day - the variable regsave . It displays the current amount, asks for the new figure, displays this and asks whether the new figure is OK. If not, it runs the option again, if all is well it returns to the PAY DAY menu.

```
1000 REM THE PAY DAY OPTION
1005 LABEL menu3: CLS
1010 LET togflag=0
1015 CSIZE 8,16
1020 PRINT "          PAY DAY MENU"
1025 CSIZE 8,8
1030 PRINT : PRINT : PRINT INVERSE 1;
     "1) CHANGE AMOUNT OF REGULAR": P
     RINT INVERSE 1;"   SAVINGS"
1035 PRINT : PRINT : PRINT INVERSE 1;
     "2) PAY IN SALARY"
1040 PRINT "    Transfers to funds mad
     e        automatically"
1045 PRINT : PRINT INVERSE 1;"3) CHAN
     GE FREQUENCY OF REGULAR": PRINT
     INVERSE 1;"    PAY DAY"
1050 PRINT : PRINT : PRINT : PRINT IN
     VERSE 1;"9) EXIT TO MAIN MENU"
1055 PRINT : PRINT : PRINT INVERSE 1;
     AT 20,0;"Keys D/E toggle funds
     display"
1060 PRINT PAPER 6; PEN 0; AT 21,0;"S
     ELECT OPTION NUMBER OR D/E"
1065 LABEL men3: GET opt
1070 IF opt=0 THEN wrongoption: GOTO
     menu3
1075 IF opt>3 AND opt<9 THEN wrongopt
     ion: GOTO menu3
1080 IF opt=13 OR opt=14 THEN LET men
     no=3:toggle
1085 IF opt >9 AND opt<13 THEN wrongo
     ption: GOTO menu3
1090 IF opt>14 THEN wrongoption: GOTO
     menu3
1095 GOTO 1000+(100*opt)
```

The procedure - statement - is a short one used by the next menu option, and also by one of the BANKING MENU options which we shall meet later. It gives a printout of the amounts held in the three funds, and the total. You may find that your printer requires you to enter the "£" as "#" in LPRINT lines to make the "£" print properly.

```
1100 CLS
1105 PRINT "Your present savings are"
     : PRINT "£";regsave;" per";
1110 IF I$="W" THEN PRINT " week"
1115 IF I$="M" THEN PRINT " month"
1120 INPUT "Please enter new amount";
     regsave
1125 CLS : PRINT "You will now save":
     PRINT "£";regsave;" per";
1130 IF I$="W" THEN PRINT " week"
1135 IF I$="M" THEN PRINT " month"
```

```
1140 PRINT : PRINT : PRINT "Is this O
     K?":yes_no
1145 IF Y$="N" THEN GOTO 1100
1150 GOTO menu3
```

This is the PAY DAY menu option 2 - the option used on pay day to enter your wage and have the transfers made to the various funds. You are prompted in lines 1205 and 1210 to enter your take-home pay. In line 1215, this is added first to the 'free money' fund. In line 1220, the amount you have decided to save every month is transferred from 'free' to 'savings'. Lines 1225 and 1230 transfer to 'assigned money' the amount needed to be put aside towards the bills on the calendar. The frequency of your pay day held in I$, determines which of these two lines operates. The rest of your pay is left in the 'free' fund.

Line 1235 displays the current state of the funds, and the amount you paid in. WINDOW 0,30,9,16 ensures that nothing overwrites the funds display at the top of the screen. You are then asked whether you wish to make any transfers between the funds, and if so, transfunds (line 10665) is called. The option ends by offering a printed statement. If this is required, the amount of take-home pay received is printed first, and statement called to print out the funds state. A return is made to the PAY DAY menu.

```
1200 CLS
1205 PRINT "PLEASE ENTER YOUR TAKE-HO
     ME PAY"
1210 INPUT "amount?";pay
1215 LET cash=cash+pay
1220 LET savings=savings+regsave,cash
     =cash-regsave
1225 IF I$="W" THEN LET assign=assign
     +week,cash=cash-week
1230 IF I$="M" THEN LET assign=assign
     +month,cash=cash-month
1235 CLS :disptotals: WINDOW 0,30,9,1
     6: PRINT "£";pay;" pay received"
1240 PRINT : PRINT "DO YOU WISH TO TR
     ANSFER FUNDS?":yes_no
1245 IF Y$="N" THEN GOTO 1255
1250 transfunds
1255 WINDOW
1260 CLS : PRINT "DO YOU REQUIRE A PR
     INT-OUT?":yes_no
```

```
1265 IF Y$="N" THEN GOTO 1275
1270 LPRINT "PAY RECEIVED";" £";pay:
     LPRINT : LPRINT :statement
1275 GOTO menu3
```

This last PAY DAY menu option is only used if your normal pay day changes from weekly to monthly or vice versa. It updates I$, and then makes the calculation, using the appropriate function, to determine how much must be put in the 'assigned' fund on each pay day. The functions were defined in lines 295 and 315. Like all PAY DAY options, it returns to menu3.

```
1300 CLS : POKE SVAR 618,8: PRINT "AR
     E YOU IN FUTURE TO BE PAID    WE
     EKLY OR MONTHLY?"
1305 INPUT "Please enter W or M";I$
1310 IF I$="W" OR I$="M" THEN GOTO 13
     25
1315 CLS : PRINT "YOU MUST ENTER W OR
     M":presskey
1320 GOTO 1300
1325 IF I$="W" THEN LET week=FN B(yea
     r)
```

```
1330 IF I$="M" THEN LET month=FN Q(ye
     ar)
1335 GOTO menu3
```

That is as far as we go this month, but you will be pleased to know that we are now more than half way through this giant program. See you again next month.

EDITOR'S NOTE

Readers will have noted that the program listings in this months MONEY MANAGER look different to those of previous months. Some readers have reported difficulty entering listings produced using LIST FORMAT 2 which, on a full-width printer, gives a nice indented listing with each statement on a different line. However it did make it difficult to work out the number of spaces in PRINT commands.

For this reason we will now be using the block standard with line numbers pulled to the left and NO EXTRA SPACES introduced into the listing. I hope this makes life easier.

18

HELP PAGE

By:- Nev Young.

Mr Jones of Ladybarn writes to ask why after he has entered about 200 lines of text of an adventure program does his Coupe die. To me this sounds like a problem with one of the early version 1 roms. If you have upgraded them I fear that you have a sick SAM. Phone PBT and get details of how to get it fixed.

A regular writer Malcolm Perry has written with a little problem about printing on the bottom part of the screen. He says that you can not erase old text with PRINT AT like you can on the main screen so he now uses a ROM call at 3438 (0D6EH). This is CLS-LOWER and does just that. He asks if this the "best" way of doing it.

Well it is certainly a good way and it works but you might be pleased to hear that you CAN do PRINT #0; AT etc and it will work. Another quick way of clearing the lower screen is to use INPUT , (don't miss out that comma).

Mr Birkett of Chester-le-street has got a real poser. He has bought a modem built by GEC type BCW93K. Problem is he doesn't know how to set it up. Can anyone supply him with a manual or let him know where to get one. Personally since there is no BT approved sticker on it I wouldn't touch it. Especially as B G Services sell the VTX5000 at only £15. You will need to spend more than this on a serial interface for your Spectrum. As for connecting it to your Archie then I'm sorry but I haven't a clue.

Mr L.G.Baumann is having problems using the OCP Finance Manager +80 on his PLUS D. This gives an error 'Invalid Code' when he tries to load a file. I'm afraid the reason is that the program uses microdrive functions that do not exist on the PLUS D. I do know that a PLUS D version was produced but I can not advise you where it could be obtained now, anyone out there know?

Anton Rothwell is having problems with the MIDI interface on SAM. Now I don't know too much about MIDI as I can't find a cello with such an interface. However looking at your code I think I can see where at least some of your problem is. Your code consists of loading the A register with values followed by OUT (253),A. I know this will not work.

Once you have loaded port 253 with a byte you must wait for bit 4 of port 249 to be reset. This signals that port 253 is ready for the next byte. By just sending them as you have the data at the port will be corrupted by the next byte before it can be sent.

I've not tried this but you could try it and see what happens.

```
ORG 32768

START:
   DI ; disable interrupts during setup
   LD HL,TABLE ; point to table start
   LD (NEXT),HL ; store pointer
   LD HL,SNDNXT ; load midi out vector
   LD (23274),HL ; with our own routine
   CALL SNDNXT ; send first byte
   EI ; enable interrupts
   RET ; finish

SNDNXT:
   IN A,(249) ; get intrupt register
   BIT 4,A ; test bit 4
   RET NZ ; return if not a midi out
        ; interrupt
   XOR A ; clear carry flag
   LD HL,(NEXT) ; get pointer
   LD A,(HL) ; get byte to send
   INC HL ; point to next byte
   LD (NEXT),HL ; save pointer
   OUT (253),A ; send byte
   LD DE,TABEND ; end of table
```

```
SBC HL,DE ; test if at end of table
RET NZ ; return if not at end
LD HL,0 ; clear interrupt vector
LD (23274),HL
RET ; finish

NEXT:
DEFW 0 ; pointer storage

TABLE:
DEFB 144,60,127 ; midi data
DEFB 144,65,127
DEFB 144,36,127
DEFB 144,93,127
DEFB 144,60,0
DEFB 144,65,0
DEFB 144,36,0
DEFB 144,93,0
DEFB 0

TABEND:
```

This should set the midi interrupt vector and then send the data to the MIDI port at the maximum rate. But no guarantees.

I Think your checksum problem is that you should not use BAND A5H but just add A5H then use the total MOD 128 for the checksum. And finally there was a series of articles on MIDI some months back in FORMAT that may help you to do more.

Mr. Henderson wants to know the SAM equivalent of LOAD@ and SAVE@. Couldn't be simpler, instead of LOAD @1,3,10,64000 on SAM you use READ AT 1,3,10,64000. SAVE @ becomes WRITE AT. But note that you will need MASTERDOS if you want to load to memory addresses higher than 65535.

Alan Cox would like a SAM version of the disk cleaning program in issue 3/3. Just use the same program but change line 10 to be LET com=224 for drive 1 or 240 for drive 2.

And yet another Sam problem. I do seem to be getting a lot of these. This time from Mr D Hood of Newcastle. (Of Betterbytes fame). He raised the point that when SAM copies a disk file it will overwrite anything that happens to be at address 65536 onwards. There are two possible fixes.

The best is to buy Masterdos. the second is to POKE DVAR 5909 with a value greater than 3. This is the starting page for the copy to use. Dont make it too big or SAMDOS2 will overwrite itself as it makes no check on what is using those memory pages. (Bad Samdos2)

Well thats all for this month. Any interesting problems then write to me at:-

FORMAT Help Page,
70,Rainhall Road,
Barnoldswick,
Colne,
Lancashire,
England,
BB8 6AB.

* * * * * * * * * * * * * * * * * * * *

MASTERDOS

◄◄ A GIANT LEAP FOR SAMKIND ►►

By:- Carol Brooksbank.

I acquired Andy Wright's new, improved DOS for the SAM Coupe at the Christmas All Formats Show. Reading the manual in the train on the way home, I found myself wondering: what if I had no particular interest in programming - just used my SAM for word processing, DTP work, artwork with FLASH! and so on - would the new exciting facilities seem worth it?

After using MASTERDOS for about ten days, the answer is a definite, unqualified YES. However non-technical you are, this new DOS is going to make life easier for you.

If you write a lot of short letters, or type in many of the FORMAT routines, you have probably already discovered that the 80 files on a disc are used up in no time, but there are always 100-odd or more K-bytes of unused space left over. Even though discs are cheap enough these days, it seems a waste.

MASTERDOS lets you FORMAT a disc to hold more files, by assigning anything between 4 and 39 tracks to the directory. If you use more than 4 tracks, the number of files is 20 times the number of directory tracks, minus 2. So FORMAT "D1",10 - a 10 track directory - would give 198 files on the disc. You lose 5K of data space on the disc per directory track (which is how the normal 4-track directory gives 780K storage on an 800K disc), so 10 directory tracks would leave 750K free. A quick glance at your full discs will suggest the number of files per disc which would be most practical for you.

MASTERDOS also allows sub-directories. This is a way of grouping your files on the disc, so that you do not have to wade through a complete directory to find the one you want. As an example of how this works, I have opened 3 sub-directories on my wordpro files disc: Library, Format, and Articles. When I do a DIR, those directories are the only files listed, even though there are more than 3 files on the disc. DIR ? or DIR 1? will give a short or detailed directory of all the files. But, if I want to know which files on the disc have to do with the Library, DIR="Library" will make that the current directory, and DIR or DIR 1 will then list all the files in that subdirectory, and no others. When I am writing an article like this one, I exit to BASIC, DIR="Articles" and then re-enter the word processor. The file will then be saved/loaded using the right directory. For even more detailed sorting you can use subdirectories of subdirectories - I could use Articles/SAM or Articles/MCODE.

All this is a boon to someone like me, whose discs are mostly very disorderly. The other bonus is that the short directory, DIR, lists files in alphabetical order. A DVAR POKE will turn this feature off if it happens to displease you, and another lets you choose how many columns are to be used in displaying or printing the short directory.

Discs can be given names, and a very useful DATE feature has been provided, which records the date on which a file was saved. This is meant to be used with the clock on the new SAM BUS, but even if you don't have that, you can still date your files. DIR DATE will display a full catalogue which includes the dates. To anyone as slapdash about discs as I am, this is a bonus indeed. I often grab any old disc with space on it for saving letter files, so if I have written several letters to the same person,

there tend to be files called "SMITH" on numerous discs. The only way to find a particular one is to load and inspect them one-by-one till I find it. DIR DATE, in future, will give me the date on which the letter was written. There is a similar facility for setting the time.

There is now a RAMdisc feature. When you FORMAT a RAMdisc you specify the total number of tracks and how many are to be reserved for the directory. The RAMdisc size will depend on the space you have free in memory. SAVE, LOAD and DIR commands for the RAMdisc are just the same as ordinary disc commands - DEVICE is used to make the RAMdisc the current drive - but the speed is astonishing. This is such a useful facility that I now, on my 512K machine, FORMAT an 80 file RAMdisc with 230K storage, before loading either my word processor or FLASH!.

I use PCG's DTP PACK running under either Lerm's SAMTAPE or Steve's Software's SC_SPECLONE Spectrum emulators, and those who are familiar with DTP will know that there are a lot of extension files, extra fonts and so-on which you may need to load. I have made up a disc of the ones I need most often, and before loading DTP I copy these files to the RAMdisc. I have modified the Spectrum and SAM BASIC so that when Drive 2 is selected, (I have only 1 drive), the RAMdisc becomes the current drive. Files are loaded like lightning, and there is still enough room on the RAMdisc to save copies of the file I am working on, if I want to free up the memory, say, to work on an illustration. Of course, before SAM is switched off, you have to remember to save all the important files to a real disc.

With FLASH!, this 80 file, 10 track RAMdisc can hold up to 9 mode 4 screens, and they save and load in the twinkling of an eye. You can hold screens for BLOCK/INSERT, intermediate versions of the one you are working on, files of alternative fonts, texture fill patterns and so on. Again, I have modified my copy of

FLASH!. TAPE commands now use the RAMdisc, but CAT and ERASE are not available from the TAPE menu, so I have made those disc options call for a "drive number" INPUT and set DEVICE appropriately.

You can have up to 5 RAMdiscs operating, but you would really need the new Card Cage from SAMCO and their external memory modules to take advantage of all of them.

A new command - BACKUP - clones a disc onto another formatted one. Any files already there will be lost and you will be left with two identical discs. So, what if you want to copy the files from one disc to another, without losing the existing files on the target disc, and have only one drive? Do you still have all that disc swopping? Not any more. FORMAT a RAMdisc, D3, and use COPY "d1" TO "d3". If there are not too many files, O.OK will tell you all is done, so put the target disc in the drive, COPY "D3" TO "D1" and the copy is made with only one disc swop.

If there are more files to copy than the RAMdisc can handle, you get a "Disc full" error message, so you copy the RAMdisc files to the floppy, and then FORMAT the RAMdisc again. Now another very useful facility comes into play.

COPY "D1" TO "D3"?

will make the computer ask

COPY "filename"? (y/n/a/e)

before copying each file. "n" goes straight on to the next, "y" copies the current one, "a" copies the current one and all the ones coming after it in the disc directory, "e" exits COPY mode. So, reply "n" to all the files already copied and "a" to the first uncopied one and all the ones still to be done are copied to the RAMdisc. This "ask me" feature can also be used with ERASE.

File copying is so easy and quick with MASTERDOS that I now use SAM for

all my Spectrum disc copying too. The file copies are exact copies - they are still ZX files even though SAM did the copying. No more "put target disc in drive" - "put source disc..." over and over again for me!

I have also used this feature to copy all the files except SAMDOS2 from the SAMDOS system file to make a new MASTERDOS system file, and I have added another menu option to the "auto" program, which FORMATS the RAMdisc and returns to the menu, so that I can do that quickly before loading my modified FLASH! or going to BASIC to load the DTP files and program.

Files, subdirectories and discs can all be renamed. Filename extensions are still available, but spaces are inserted in the filenames where necessary so that "." and extensions are all in line in the DIR display. Again, there is a DVAR to POKE if you don't like this way of doing things. Wild cards, hidden files and file protection are retained, and there is now a DSTAT function which enables you to discover useful things about the current disc, like the number of file slots used and free, the amount of free space, whether the disc is write-protected, the number of files in the current directory, the number of directory tracks on the disc and so on. Another function, FSTAT, can be used to find the directory number, file length, file type and protected status of a file.

For programmers, MASTERDOS is packed with goodies. The READ AT and WRITE AT commands have been extended to operate on more than one sector - you specify the number of sectors involved and to work on addresses greater than 65535. MERGE, used to suppress the auto-running of a BASIC program, can also be used to stop an auto-run CODE file from CALLing itself.

The most exciting innovation is the handling of serial and random access OPENTYPE files. Even those who have never before ventured into this sort of disc/file operation will find the

straightforward examples make it all very easy to follow. I have one or two handy programs somewhere in my attic which I used to use in Microdrive days. I must dust them off and modify them for SAM now. SAM's MASTERDOS file handling is much more versatile than the Microdrive's - or the PLUS D's - because MASTERDOS lets you open an existing file, move the pointer to the end, add some more data, and close it again. There are example programs given for moving the pointer around files with fixed-length or variable-length records to make record access easier. MOVE is available too, for moving files from disc to disc, or to screen or printer. It even works on old Spectrum/PLUS D TASWORD 128 files.

MASTERDOS is definitely for everyone, not just for skilled programmers. The only trouble is, once you start using RAMdiscs you realise how useful it would be to have the BUS and a 1Mb external memory module or two so that RAMdiscs could hold the contents of one or two floppies in memory at once - what a big database that could let you use! Sadly, the piggy bank starts looking a bit inadequate.

Still, I came home from the All Formats show to find a cheque from the football pools people on the doormat. Only £3.90 this time, but if only I could get that right, I could get a Card Cage and a memory module (or maybe even two) and... Pull yourself together, Carol, and start saving up!

But believe me, you too will think MASTERDOS the best software investment you have made for a long time.

MASTERDOS is available for £14 + p&p direct from SAM Computers Ltd, Tel: 0792-700300.

EXAMPLES
OF MODIFICATIONS TO PROGRAMS

PCG's DTP PACK

Running under Lerm's SAMTAPE or Steve's Software's SC_SPECLONE. (SC Software's SPECMAKER emulation already

uses its own RAMdisc).

The line numbers in the two versions are different, but the structure is very similar.

a) In the Spectrum BASIC, in the subroutine called by the various disc operation lines, add a line or command at some convenient point:-

POKE 23312,d

b) In the Spectrum BASIC, add the GOSUB instruction (which is in the lines which exit to SAM for LOAD, SAVE, and ERASE) to the line which exits to SAM for DIR

c) In the SAM BASIC, immediately before the command which PEEKs (23296+65536), add:-

LET d=PEEK (23312+65536):DEVICE Dd:POKE DVAR 112,3:

d) In the SAM BASIC, change DIR 1 to DIR d (or DIR DATE if you prefer), wherever it occurs.

e) In the line from which the program auto-runs, immediately after the CLEAR nnnnn command, add:-

INPUT "DATE?";d$: POKE DVAR 150,0: DATE d$

f) Save the Spectrum memory and the SAM BASIC again, using the method described in the emulator handbook.

- - - - -

FLASH!

a) In lines 1015, 1025, 1035, change DEVICE t to DEVICE D3

b) Change the following lines to:-

1040 INPUT "DRIVE?";d: DEVICE Dd: DIR: PAUSE: RETURN

1050 INPUT "DRIVE?";d: DEVICE Dd: DIR: INPUT "FILENAME?";a$: ERASE a$:

RETURN

c) SAVE OVER "FLASH!" LINE 9800

NB. With both DTP and FLASH!, if you want to use COPY "d1" to "d3" to put files on the RAMdisc, do so loading the main program. After the program is loaded, any files saved to the RAMdisc must be saved in the normal way through the program, selecting Drive 2 with DTP or TAPE with FLASH!.

- - - - -

AUTO

This is the menu program on the SAMDOS master disc.

a) Amend the following lines:-

50 - Add ,"Format RAMdisc" at the end of the DATA

80 - Change FOR i=65 TO 76 to FOR i=65 TO 77

140 - Change PRINT AT 15,3; to PRINT AT 16,3;

260 - Add :GOSUB 1000 at the end of the line

b) Add a line

1000 FORMAT "D3",4,50:RETURN

N.B. A RAMdisc of this size is only practical with a 512K SAM.

c) Change lines 350 and 480 to read:-

PAUSE: INPUT "NUMBER OF DIRECTORY T RACKS?";n: FORMAT "D1",n

d) Change the following lines:-

520 IF a$="y" OR a$="Y" THEN CLS: P RINT "Put MASTERDOS system disc in drive 1 and press a key": PAUSE: CO PY "DI:MDOS1" TO "D1"

390 BACKUP "D1" TO "D1"

e) SAVE OVER "auto" LINE 10
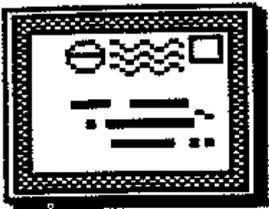
YOUR LETTERS
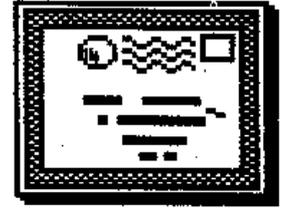
Dear Editor,

I have received the new copy of FORMAT with a final reminder that my membership was about to run out. This is not the case. You have the cheque in your possession as you returned my previous renewal form with a note saying you could not process it because I had not filled in my membership number.

I find it quite extraordinary that you cannot tie up my name and address with a membership number which no-one is likely to have to hand. I suggest that you invest in a computer of which there are many excellent makes on the market, such as SAM. It is also possible to purchase programs which do data-crunching, which could do the necessary tie-up, as with telephone names and numbers.

However, as I now have an envelope with my number on it I am returning the renewal notice.

Yours sincerely, L.W.Simpson.

Most organizations like ours allocate membership numbers and I don't think it is unreasonable to expect members to quote their number when contacting us.

We do use computers, there is no way we could run things without them. The main data-base program currently runs on a Spectrum. It records members names and addresses; expiry date; membership number; telephone number if we have one; and details of past orders processed. As there are now over 3000 records, each of which takes up 480 characters, the total data-base now runs to two discs with a third being needed soon when we go above 3200 records.

Each record is allocated a sector on the disc. This means that, given a membership number, access to any record is almost instant. However, if no number is quoted then the only way to find a record is to search the entire data-base record by record. As your record would be number 1921 this means that nearly two-thirds of the records have to be read before we get to yours. If we were to look for your surname, and there is more than one Simpson on file, you can see that it takes quite some time to find your record.

If we have time, and the data-base is not being used for something else, then we have a go at finding a match. If none is found or we lack the time then we either send the renewal form back or file it and await a phone call from you.

So, PLEASE remember your membership number. Ed.

Dear Editor,

Please, PLEASE, P L E A S E could us Spectrum+3 owners have a little more coverage in your otherwise excellent magazine. Now the +3 has been killed off by Amstrad we really need your help.

Yours sincerely, Paul Marsh.

I get lots of letter from people asking for more articles on this or that, and I do know how desperate some people are for more information on their favourite machine or subject. But, as I've said before, we can only publish what comes in.

Now Paul, from your full letter I gather that you have had your +3 for two years. Well in that time you will have learnt something about the machine. If you write that down (using a word processor of course) then I will print it. Then other +3 users will start writing bits and before we know whats going on there will be lots of +3 items in FORMAT. Even if you only have a few hints or pokes then send them to John Wase for the SHORT SPOT, just don't sit there - WRITE to us. Ed.

Dear Editor,

With reference to the letter from Mr Lankester in the December issue. If he, or any other reader, is having trouble finding a solid rather than a ribbon two-way connector, then they may like to contact Greenweld Electronics Ltd. The company still advertise the Currah 'T' slot connector in their catalogue at £2 each plus postage. They can be contacted by on Southampton (0703) 772501.

Yours sincerely, W.R.Vanstone.

Dear Editor,

Since I first played with a computer (a ZX81) I've been hooked. I purchased my own ZX81 and the following year I became the proud owner of a 16k Spectrum. Since then I've owned a DRAGON, a PLUS 4 (a much under-rated machine) and a Amstrad 128k +2 (the old gray type that work - not the +2a) I've even played with a borrowed ST for a few weeks.

Now I GOT A SAM! It came as a pre-christmas treat (from myself) and it has proved a fantastic buy. In the past I have not considered myself as a programmer - and machine code is way over my head - but with SAM Basic the graphics you can produce are just wonderful. I will still keep my Spectrums but SAM is the machine I will be doing my programming on.

Given its power why are other magazines ignoring SAM? FORMAT is the only one that has regular coverage. In the old days (around 1984) there were so many machines on the market that no magazine could really cover them all. But now there are so few I would have thought mags could devote a page or two for SAM.

Yours sincerely, Simon Minchin.
* - * - * - * - * - *
Letters may be shortened or edited to fit on these pages.

This is YOUR letters page so it is up to you, our readers, to fill it. Send your letters to the usual address and mark the envelope LETTERS in the top left-hand corner. Keep your letters as short as you can so we can fit in as many as possible.

'No wonder I'm feeling out of sorts, standing about in polluted water all day!'

♫ PLAY 128 ♪

A Little Music For Your 128K Spectrum

By:- Steve Warr.

The Spectrum 128k's PLAY command does have its limitations, mainly that the computer cannot do anything else while a tune is being PLAYed, but it is still quite an easy way of writing music compared with trying to program the sound chip directly.

Two examples of using the PLAY command are given below. Both are probably tunes that you will recognise, just type each one in turn and run it. The first plays for quite a long time but it does eventually come to an end. The second plays for a much shorter time but in some ways it is more ambitious because it also incorporates a beat and uses the sound chip's wave generator.

Listing 1

```
10 LET z$="M7N3DEC5a3b5g3dec"
20 LET y$="N5A3B5G3DEC5a3ba#g5g&"
30 LET x$="O5N3e5C3e5C3e8CO6N3CD#DEC
   D5E3b5D8C"
40 LET w$="O5N3d#de5C3e5C3e8C3DO6ag#
   faC5E3DCa8DO5N3d#d"
50 LET v$="O4N5cCgCfC3eO5CD#DECD5E3b
   5D8C"
60 LET u$="O4N5bcCgCfC5eO5N3Cg#faC5E
   3DCa8FO4N5b"
70 LET t$="O3N5CEg$BfAeGgGgBC"
80 LET s$="gCGCEg$BfAe#Dd#FDABgab"
90 LET r$="O6N3CDECD5E3CDCECD5E3CDCE
   CD5E3b5D"
100 LET q$="O5N5f3e5f3e5f3&aDaCDCa(5g
    3#f)5g3&CFCDEDC"
110 LET p$="O5N3CDECD5E3CDCECD5E3CDCE
    CD5E3b5D"
120 LET o$="O5N5d3#c5d3#c5d&f&O4aE(3#
    D5E)cG&C"
130 LET n$="O4N5gC&cCO3$bO4E03aO4FO3#
    gO4FO3gO4CO3gB"
140 LET m$="O3N5fAaAfAaFCO4CO3geCO4CO
    3gE"
150 LET l$="(3ef#f5g3a5g)3ecO5gabO6cd
    edcdO5gO6efgagef5g3a5g3ef#f5g3a5g
    3ga$bb5bb3a#fd7g3&ef#f"
160 LET k$="(3CD#D5E3E5E)3E5eO3f04FfF
```
```
    eEO3gO5eE3E5E3CD#D5E3E5E3EO3N5#DO
    5N3D5D6C5C7b3&CD#D"
170 LET j$="cCgO4CO3cCgGfAF#GEGgGcCgO
    4CO3cCE#D3DDGG5DAGfed"
180 LET l$=l$+"5g3a5g3ef#f5g3a5g3ecO5
    gabO6cdedcd7c3cO5g#fg5C3a5C3aCagC
    E5G3ECg5aCO6N3e5d8c"
190 LET k$=k$+"5E3E5E3CD#D6EE5e03f04F
    fFO4eEcEAF#D#DE7E5E#F#FO5N6f8e"
200 LET j$=j$+"cCgO4CO3cCgGfAF#GEGC$B
    FF#F#FG7G5GDDgbCg"
210 LET i$="(5D3#C)5D3&FAFGAGFO6CC6C3
    &5agO5N3ge5ge"
220 LET h$="(5B3$B)5BO3gO4BdG3#F#F6#F
    !!3&5#FC&&&"
230 LET g$="gG7bO3N5gGD6G3G6G!!3&5GE&
    &&"
240 LET f$="O5N3a#ga5GF3CE#DE5AO6N3CO
    5GE5CC3E5D6C"
250 LET e$="O4N5f3dA7A5GE#DE#F#F6FE"
260 LET d$="O3N5fdefgG#FGAdgbC"
270 LET f$=q$+i$+q$+f$
280 LET e$=o$+h$+o$+e$
290 LET d$=m$+g$+m$+d$
500 LET a$=x$+w$+x$+r$+"8CO5N3d#d"+x$
    +w$+x$+r$+"O6N7C3&"+l$
505 LET a$="T140V1506"+z$+"O4"+y$+"O5
    G3d#d"+a$+l$+"3&d#d"+a$+"3&d#d"
510 LET b$=v$+u$+v$+p$+"8CO4N5b"+v$+u
    $+v$+p$+"O5N7C3&"+k$
515 LET b$="V1505"+z$+"O3"+y$+"O4Bb"+
    b$+k$+"6&"+b$+"6&"
520 LET c$=t$+s$+t$+n$+"O3Cgab"+t$+s$
    +t$+n$+"O3N5Cgc&"+j$
525 LET c$="V15M3N9&&&7&M7O2N5gO3G"+c
    $+"ed"+j$+"3c6&"+c$+"3c6&"
550 LET a$=a$+"O6N7C5C&"+f$+"3ge5ge"+
    f$+"5CO6C"
560 LET b$=b$+"O5N7E5E&"+e$+"5&&&"+e$
    +"5EO5E"
570 LET c$=c$+"O3N5Cgc&"+d$+"5&&&"+d$
    +"gc"
600K PLAY a$,b$,c$
```

Listing 2

```
10 LET z$="UX20000WON7f6#g5f3f5#af#d
   7f6C5f3f5#CC#gfCF3f5#d3#d5cg8f"
20 LET y$="M39UX15000WON9f"
30 LET x$="(9&&&&)"
```

```
40 LET w$="(O4UX3000WON5fO5UX4500WOC
   C3C5#D#D3#DDD5DCCC3C5#D3#D5DC7&5#
   g#g#g3#g5#a#a3#a#a#a5#aCCC3#a5C6C
   5&)"
50 LET v$="(5&O5UX4500WOaa3a5#a#a3#a
   #a#a5#aaaa3a5#a3#a5#aa7&5fff3f5gg
   3ggg5gaaa3g5a6a5&)"
100 LET u$="UX3000WON7f6F5#D3#D5CC#D7
    fF3&C5C#DF7#C6#C5#D3#D5C#DF6F8&3#
    D5C#a#g"
110 LET t$="M7V13N3"
120 LET s$="V15N3d"
130 LET r$="V10N1a&"
140 LET q$="O3"+t$+"fM35"+r$+"a&a&"+s
    $+r$+"a&"+t$+"#DM35"+r$+t$+"#DCM3
    5"+r$+s$+r$+t$+"#DM35"+r$+t$+"fM3
    5"+r$+"a&a&"+s$+r$+"a&a&a&"+t$+"C
    CM35"+r$+s$+r$+t$+"FM35"+r$+t$+"#
    CM35"+r$+"a&a&"+s$+r$+"a&"+t$+"#D
    M35"
145 LET q$=q$+r$+t$+"#DCM35"+r$+s$+r$
    +t$+"FM35"+r$+t$+"FM35"+r$+"a&a&"
    +s$+r$+"a&a&a&UX3000WON1b&"+r$+"a
    &"+s$+r$+"a&a&"
150 LET p$="UX3000WON7&3&#D5C#a#g"
160 LET o$="UX3000WON1b&V9"
170 LET n$=o$+"(a&a&a&)"+o$+"a&"+o$+o
    $+"a&a&a&"+o$+"a&"+o$+"(a&a&a&a&)
    "+o$+o$+"a&a&a&a&a&"+o$+"(a&a&a&)
    "+o$+"a&"+o$+o$+"a&a&a&"+o$+"a&"+
    o$+"(a&a&a&a&)"+o$+o$+"a&a&a&a&a&
    "
180 LET m$="M7UX4500WON"
190 LET s$="M35"+s$
200 LET l$="UX3000WO(O3M7N3fM35"+r$+"
    O5"+m$+"5f"+s$+r$+m$+"3fO3"+t$+"#
    DM35"+r$+t$+"#DCO5"+m$+"3g"+s$+m$
    +"3fO3"+t$+"#DM35"+r$+t$+"fM35"+r
    $+"O5"+m$+"5f"+s$+r$+m$+"3f5gO3"+
    t$+"CCM35"+r$+s$+r$+t$+"FM35"+r$+
    t$+"#CM35"+r$
210 LET l$=l$+"O5"+m$+"5#c"+s$+r$+m$+
    "3#cO3"+t$+"#DO5"+m$+"3#cO3"+t$+"
    #DCO5"+m$+"3#d"+s$+m$+"3#dO3"+t$+
    "FO5"+m$+"3#dO3"+t$+"FM35"+r$+"O5
    "+m$+"5f"+s$+r$+m$+"5f3#dO3"+t$+"
    #DCO5"+m$+"3f"+s$+r$+"O3"+t$+"#gM
    35"+r$+"V13)"
500 LET a$="T230M35O5("+z$+"T115N8&T2
    30)(O3"+u$+")(O5"+z$+"O3"+p$+")"+
    w$+"O4"+y$
510 LET b$="M35O6("+z$+"8&)"+x$+"O4"+
    u$+"(O6"+z$+"O4"+p$+")"+v$+y$
520 LET c$="M35N9&&&&&8&&&9&&&&&8&&UX
    12000W1d"+x$+n$+"("+q$+")"+l$+y$
600 PLAY a$,b$,c$
```

I hardy need to say that you must type in the programs very carefully to get all the notes right. I hope you enjoy listening to them and with any luck we might see some music from other people in the future.

                * + * + * + *

FORMAT BACK-ISSUES

For readers who have missed past issues of FORMAT. All issues, from Volume 1 Number 1 right up to the latest issue, are available. To order Just quote the Volume number and the Issue number.   Price £1.00 per copy.

FORMAT BINDERS

Hardcover binders to protect your valuable magazines. Each will hold 12 issues, easy to insert/remove.
Order Code FMB-01   Price £4.25

SPECTRUM SOFTWARE

Supplied on tape for use with any format/size of disc. To use conversion software you must have the original tape copy of the software to be converted. Most conversions require the latest version of DOS.

ARTCON 48
Converts the 48k OCP ART STUDIO to full DISCiPLE / PLUS D operation, including printer.
Order Code FST-02   Price £3.95

TASCON 128
Converts TASWORD 128 for use with DISCiPLE or PLUS D. All features retained, includes amendments for TW2 file converter.
Order Code FST-05   Price £3.95

TASCON +2
As for TASCON 128 but converts TASWORD +2 to disc.
Order Code FST-06   Price £3.95

WRITER 48
Converts the WRITER 48K for use with DISCiPLE or PLUS D.
Order Code FST-07   Price £3.95

G+DOS 2a
Latest version of the PLUS D system file. Works with ROM version 1a.
Order Code FST-90   Price £3.95

GDOS 3d
Up to date system file for DISCiPLEs with ROM version 3. Includes new set-up program, far easier to use.
Order Code FST-91   Price £3.95

FORMAT READERS SERVICES

HARDWARE

KEYBOARD STICKERS
Spectrum (48k) keyboard stickers to give all the single key keywords. Multi-Colour printed on clear plastic.
Order Code ODD-01   Price £1.50

PRINTER LEAD
Top quality parallel printer lead (BBC Standard). 2 meters long. For use with DISCiPLE, PLUS D or SAM.
Order Code PPL-01   Price £7.95

SAM Coupé SOFTWARE

NEW   DRAGON TILES   *NEW*
The thinking man's game. Always different, always taxing, the most addictive game so far written for SAM. Supplied on 3.5" disc. (tape soon)
Order Code SCD-01   Price £5.95

POSTAGE/PACKING And ORDERING

Add the following charges, for postage and packing, to your total order.

	UK	OVERSEAS	AIR
Under £10	£1.00	£1.50	£2.50
£10 to £25	£1.50	£2.25	£3.50
Over £25	£FREE	£3.00	£4.50

Please **WRITE YOUR ORDER ON A PIECE OF PLAIN PAPER**, and mark envelope ORDER.
   Clearly state Order Code; Title; Quantity required; price & add any postage as above. Remember to quote your membership number or we cant send out your order. Payment in STERLING by Cheque (UK bank), P.O. Euro Cheques or Cash. Make cheques payable to FORMAT Publications. Payment MUST accompany order, do not send by different route.
   Goods will be sent out at the same time as the next available issue of FORMAT. Shortages or non-deliver should be reported withing 14 days. FORMAT Publications will not be held liable for delays or non-delivery due to circumstances beyond its control.

INSIDE
CHRISTMAS CAROLS

By:- Clyde Bish.

In the December issue FORMAT published my game for Christmas called "Carol Quiz". I left you with the promise that, for the benefit of serious programmers, I would follow it up with an article on how it works and this is part of the result.

Now is your chance to delve into the depths of "Carol Quiz" and hopefully pick up some programming tips and so add to the enjoyment I hope you've had playing it.

I think things will be easier to follow if I go through it block by block. O.K. Get the listing in front of you and off we go.

Line 1 This lowers RAMTOP to leave space for the machine code that will load in later, then jumps over the next few subroutines (which are put at the start to make them run faster) to the program start at line 9.

Line 2 This subroutine scrolls the titles across the screen. It works by setting the FOR/NEXT loop from 1 to the length of the title (preceeded by 32 blanks) then printing sucessive 32 character sections of this title (now also predeeded by 32 blanks) at a position on the screen determined by the row value in variable j and at an INVERSE setting determined by i. So the first time through just 32 blanks are printed. The next time the first blank is missed out and the first character of the title is added, so this appears right of screen. On subsequent loops more of the title appears with the left end moving across the screen. This is a very nice demonstration of Sinclair BASIC string slicing.

Line 3 This line prints rows of stars which can be made to flash by changing the ink colour each time the routine is called. This is done by clever use of logic. The statement at the start of the line may not seem to make sense but to a computer's logic it does. If z is set to 1 then the statement following the equals sign is untrue and the value of z will be reset to 0. When the program flow reaches the PRINT statement the expression in brackets is true so 2 will be added to the INK number making 4 (i.e. Green). When the subroutine is called again the value of z is still at 0 so the opening logic statement is true and hence z will be set to 1. Now the expression in brackets in the PRINT statement is untrue so zero is added to the INK colour leaving it at 2 (i.e. Red). Although you may have found this difficult to follow I assure you it is worth the trouble. In this case logic has made it possible to use one line to produce two different screen images.

Line 5 This line sets up variable xx which is used to adjust the position of the title print. P$ is the title string and xs is the horizontal magnification. The bracketed section works out how many pixels are not used up by the title. This is divided by 2 to give equal spacing either side, which can then be adjusted by adding the value in p (which could of course be negative). This value and others set before calling the subroutine are POKEd into memory spaces using the value in i or the t. FOR/NEXT loop. (The 8 and 255 which are POKEd in simply act as markers). A machine code routine starting at 65091 is then called to PRINT the suitably magnified title onto the screen.

The above three lines are good examples of how the PROCedure command of BBC and other Basics can be simulated on the Spectrum by setting the values of variables then calling a

subroutine which uses them.

Line 9 Now to where the program actually starts! After setting a few variables and screen colours the program starts to load in the data for the Enlarge_Characters machine code mentioned above. This data starts at line 950 (hence the RESTORE) and also contains the data for the user-defined graphics. It is usual to load in machine code bytes which have been saved separately from a tape. This holds up the program flow unnecessarily and can be avoided by the following system. The title screen display requires a background of stars. These are printed randomly on screen using the f FoR/NEXT loop but instead of having a PAUSE to slow this down the time is used to READ five bytes of the machine code and POKE them into memory. In this way the user has no idea that anything (apart from star printing) is happening and interest is held. Crafty! The remainder of the line also holds some interesting features. Variables a$ and b$ hold graphics blocks which are going to be used repeatedly to produce the "Gothic" script for the title. By declaring these graphics patterns as variables much effort (and time) can be saved. The User-defined graphics (udgs for short) held in variables x$ and y$ are also used often. When you enter them you will include colour control characters within the variable. This will make subsequent printing much simpler. But more of that later. At the end of the line the first use for the Enlarge_letters machine code routine can be found. Variable p$ holds the word to be printed (Its only "arol" as the "C" will have already be displayed in that "Gothic" script. yy means start 24 picel lines from the top, and p means print 32 picels off centre. xs = 4 and ys = 8 will produce letters which are 4 times as wide and 8 times as tall as normal.

Line 50 Here the printing of the title screen is completed in the same way as in line 9 with the word "Quiz".

Line 90 During a short pause more variables are set. Again this will not be apparent to the user. These variables will be used later, particularly in the DATA lines to produce the carols playing. The first four refer to the lengths of the notes and all use the variable m (already set in line 9) to determine their length. So if m stands for minim c will represent a crochet, which has only half its length hence m/2, and q for quaver, which is only half as long again. Musicians will no doubt by now have worked out that dm and dc stand for dotted minim, with a length of one and half times that of a minim (hence m+c) and dc for dotted crochet (c+q). By using this system it is only necessary to alter the value of one variable, m to alter all the other note values and so speed up or slow down the tempo of the piece. By declaring variables with the initial letter of the note name it is much easier to produce DATA lines direct from the music, rather than have to convert each to a number as is usual with BEEP. The remainder of the line declares variables using a similar system of mnemonics for the pitch of the notes. Using capitals make it easier to see in DATA lines which are pitch values and which are note lengths. So G has a numeric value of 7. FS of course stands for F# and TC for top C.

Line 100 Now back to the screen display. This line changes the PAPER colour of the displayed screen without wiping the screen at the same time (which is what the more usual CLS would do.) The n loop makes the remainder of the line operate 4 times. The i loop causes the paper colour to be altered row by row from the top. Over 1 in the PRINT statement will allow the PAPER and INK colours to be changed without affecting the printing whereas INVERSE 1 causes the INK and PAPER colours to reverse with each loop. Thus after the first loop the printing, which was white on blue us now blue on white. The next loop sets it back again. (Another way of achieving the same effect is to DIMention a string of 704 spaces - the number of character spaces on the

screen - using DIM c$(704) then simply printing c$ and not using the i loop, but this does not give such a smooth change.) The end of the line changes the background colour to cyan using the same OVER 1 system.

Line 110 A graphics line is printed between the title words and bells are hung thereon. As the bells swing from side to side a sequence of chimes are heard. This is achieved by PRINTing AT the same line two different sets of bells and so achieving simple animation whilst the values for BEEP are READ from the DATA information which ends the line. In this way the alternating animation sequences can be used for all of the tune.

Line 170 Now for the information on how to play. This will scroll in as a series of lines from the right, each one preceeded by "You Know Who"! It uses the subroutine at line 2 (the flow of which has already be described) having first set the variables for print row, inverse and the information message.

Line 175 After resetting the screen colours this line uses the data at the end of it, and transferred to variable t$ to determine in which columns the holly leaves held in variables x$ and y$ are to be printed: This information is used twice, once for the top and once for the bottom row.

Line 180 This line prints up the titles of the carols, preceeded by a number. This is done by READing the title of each which appears in the DATA lines starting at line 1000 (hence the start of the loop). The phrase n/500-d reduces the value in the loop variable n to the numbers 1 to 10. See if you can work out how.

Line 200 The latter part of this line uses an empty loop to simulate a PAUSE. You may (and should) ask why not just use PAUSE? The answer is that PAUSE is broken by a key press. An empty loop ensures that the information stays on screen in spite of accidental key presses. It is a useful programming trick to know. With

the exception of s (the score) you should recognise the variables set at the beginning of the line as being those needed for the Enlarge_Letters routine. They are going to be used later but are set now whilst there a natural pause in the program flow. Note that the variable p is set to zero so all messages will be centred on screen.

Line 210 Now at last the game starts! You'll have ten tries so the v loop is set to ten, then a random number is choosen. This has to be between 1 and 10 so the number generated by RND (which will be between 0 and .99999) is multiplied by 10 to give a range of 0 to 9.9999. this INTeger of this lies between 0 and 9 so adding one to this gives the required range, 1 to 10. Line 215 : This value, held in w is used to RESTORE the start of the DATA to the choosen carol. Theses DATA lines run in increments of 500 starting at line 1000, so if w was say 1 then w*500+500 = 1000 which is the DATA line for the first carol. Check it out. The title is read into a$ then a random start position in the DATA is selected. If you look at this RND statement you should now be able to work out that the start position will be between 5 and 15 notes into the tune. The n loop rEADs these notes and, in effect discards them. The loop is then reset to the new start position and, given the chance will play the next 30 notes using READ and BEEP. However the player should want to interrupt the music with the answer. The IF INKEY$ ... allows this to happen. Normally INKEY$ will return "" unless a key is pressed. If this happens then the flow jumps out to line 310.

Line 220 If you don't respond before 30 notes are played the program reponds with the "too slow!" message (printed using the Enlarge_Letters routine for which we set the variables in line 200.

Line 310 Here your response (as INKEY$) is compared with the number of the carol being played. This numeric variable has to be converted to a

STRing before such a comparison can be made. If it does not match a flashing "Wrong" is printed (also using the Enlarge_Letters technique) and 15 penalty points are added to the score held in s.

Line 320 You're correct! The program informs you of the fact (as if you didn't know) with a loop of rising BEEPs and the "Correct"message, again using those parameters set up earlier in the Enlarge_Letters routine.

Line 330 The number of points equal to the number of notes played (n) is added to your score, which is also displayed enlarged. An empty loop (to simulate an unbreakable PAUSE) appears, followed by a loop to print blank lines to remove the messages without disrupting the remainder of the display. (Unlike CLS which would clear the whole screen.) The line ends with the next carol being selected.

Line 349 More screen display variables are set. As in line 9 they also contain colour control characters. How you get these in I'll explain at the end.

Line 350 The end screen is set up, and The score is printed in large letters.

Line 360 The score is framed with suitably festive motives.

Line 900 The length values of the notes to be played are set in x and y. Each is then READ and played before the subroutine at line 3 is called (which you will remember causes the stars to flash). To end a message is displayed inviting the player to have another go, the #0 causing it to be PRINTed on the last two screen lines which are usually reserved for commands and error messages. The program restarts from line 175 thus missing out the early title and instruction screens.

Lines 951 - 959 These hold the DATA bytes for the Enlarge_Letters machine code which is loaded in during the title screen. (Some "Elder Statesmen"

of the Spectrum fraternity may recognise as an adaptation of one of the routines on their "Horizons" tape.) I won't go into details not as to how it works. Be patient and they'll be an article later covering the many uses that the routine can be put to. Watch this space! These lines also contain the data for the User Defined Graphics which have been used for titles etc.

Lines 1000 - 5500 Herein lies the DATA for the carols. The lower case letters are the note lengths mentioned earlier and the upper case ones are the note pitch. This system makes it easier to see which is which in complex DATA lines. Each tune has its DATA repeated so that wherever the random number generator chooses to begin there will be sufficient DATA.

So now the mysteries are unravelled, and I hope the experience has been useful. I'm sure you'll find many techniques you can use in your own, perhaps more serious programs.

Next time we'll have a closer look in more detail at the Enlarge_Letters machine routine and see what else it can (apart from obviously enlarge letters!) Until then can I leave those beginning writing their own programs with a thought. I'm often asked, "How do you manage to write such long and complex programs?" I hope that have analysed "Carol Quiz" you now know the answer. Programmers don't write long programs. They write lots of short ones and join them together! Be seeing you. 'Bye.
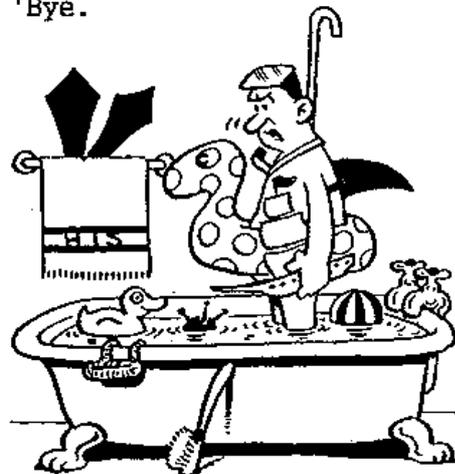
"Damn! I've forgotten the soap again!"