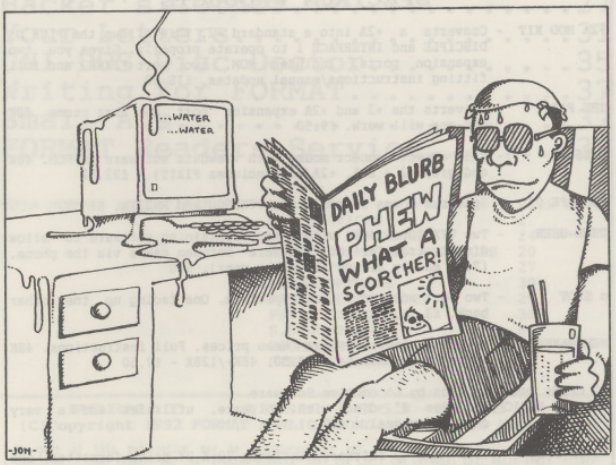
Vol.5 - Nº 10. June 1992.

FORMAT

FOR SPECTRUM AND SAM USERS



It's Not Water Your Computer Needs,
It's FORMAT.

£1.25 [UK R.R.P.] ISSN 0963-8598.

CONTENTS

Vol.5 № 10. June 1992.

FORMAT ISSN 0963-8598.

FORMAT is published by **FORMAT PUBLICATIONS.**
34, Bourton Road, Gloucester, GL4 0LE, England.
Telephone 0452-412572. Fax 0452-380890.

Printed by D.S.LITHO. Gloucester. Telephone 0452-523198.

NEWS ON 4

SAM ADVENTURE CREATOR

Colin Jordon, well known adventure author of programs such as Enid Blyton's Famous Five, is busy writing an Adventure Creator for the SAM.

Due for release soon the system will allow you to develop professional looking adventures with both text and graphics. The program will allow full use of the memory and screen modes.

VIDEO DIGITISER

The SAM Coupé will soon have its own purpose-built video digitiser. Available at first in a Mono version at 69.99 which gives up to 64 grey scales (using blues and greens to provide the extra colours just as in the famous Space Man picture on the original SAM demo.

A full-colour version will be coming later this year, but there will be an easy upgrade available to existing Mono users. Other advanced features to come will include full genloc.

HOT SANDOWN

The first really hot weekend of the year kept crowds at the latest All Formats venue well down. The hall, under the main stand at the famous Sandown race course, was packed with exhibitors, the large car parks stood ready. The hall was large and airy and clean - a far cry from the New Horticultural Hall in Westminster.

Alas, due to the heat, many customers stayed away, which was a pity as there was a good turn out of SAM & Spectrum companies. Still, the organizers can't control the weather can they.

SU GRABS CRASH

The oldest magazine in the Sinclair field - Sinclair User - has gobbled up the second oldest - Crash and the two magazines will now be run as one.

EMAP, publishers of Sinclair User, refused to divulge how much they had paid for the Crash name which had only recently been saved from the hands of the receivers. Insiders reckon that the figure was quite low as Crash had not been able to rebuild its sales following last years collapse.

The first issue of the combined magazine looked little different from previous issue of Sinclair User so it appears that EMAP just wanted to remove some opposition.

APPLE LOOSE COPYRIGHT BATTLE

Apple Computers have lost the latest round of their 'Look and Feel' legal case in the USA against Microsoft and Hewlett-Packard. Apple have been taking legal action for nearly four years over their claim to a copyright on window/icon type GUIs (Graphical User Interfaces). Apple had been claiming over $5.5 billion in damages and lost revenue.

Legal experts in the US now believe this will open the flood-gates for other companies to launch products without fear that 'Look and Feel' may be used against them.

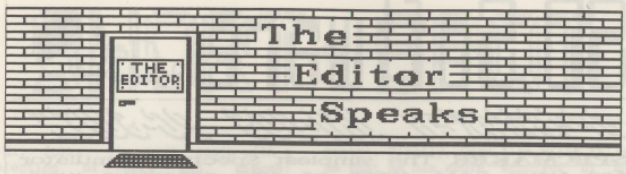
SECRETARY FOR SAM

A new wordprocessor for SAM is now available from SAMCO. Called 'The Secretary' and priced at £14.99 the program includes all the usual features required of a wordprocessor.

It also includes full use of foreign language character sets, which will be good for overseas customers and those who are doing languages at school or college. Many other advanced features make this a very interesting program.

News Credits: S.T.Hill, David Morris.

URGENT we need your news. Anything you think other people should know about. Items printed earn contributor 3 month extra subscription (please claim when renewing).



OK, OK, so I got it a bit wrong last month. Yes I know it was not Homo Erectus that encountered the first monolith. 4 million years ago it was Australopithecus that roamed the plains of Africa. However, the film did get it a little wrong, it showed creatures more like Ramapithecus which lived between 9 and 15 million years ago, Australopithecus was much less ape-like than the film showed. My excuse for my mistake? I needed the extra month to learn how to spell Australopithecus. Oh, the film! Well for those of you that didn't get the connection, the reference was to the greatest film of all time - 2001 A SPACE ODYSSEY.

From the 1st June we are launching a new ASSOCIATE MEMBERSHIP scheme for INDUG. I have known since the start of the user group that FORMAT would not prove ideal for large numbers of Spectrum owners (and latterly SAM owners too) because of it's serious content. However, until now, INDUG membership and FORMAT have been in many ways indivisible. Well Associate Membership will change that. Put simply we are expanding our circle, and everyone will benefit from that. In the first instant Associate Membership will allow large numbers to join INDUG at a cost of £4 per year.

Although we are already the largest Spectrum/SAM user group we still count only a very small percentage of owners amongst our membership. I believe that ten, twenty or even thirty thousand members would be a good target (and that would still be only a fraction of the owners). With that number we would be able to commission software, develop hardware and help to develop new uses for our beloved machines.

Associate Members will not get FORMAT of course, nor will they have access to the full services of our Technical Hotline. New discount structures will ensure that FULL Members like yourself will still enjoy many advantages over Associate Members - who in turn will of course have advantages over non-members. More details as the scheme develops.

Later this year I am planning a special games supplement. I need reviews, articles, hints & tips, pokes & hacks, in fact anything games related. If you think you could contribute then give me a ring or drop me a line. I'm not just interested in arcade games, they form a large part of the market but the other types of games - especially adventure games - will play an important role. The supplement will be available to FORMAT readers at a special price, although I also plan to sell it at shows at a higher price. Anyone with any suitable material should either give me a ring or drop me a line.

This year is, as you will already know, the tenth birthday of the ZX Spectrum. It also marks the fifth birthday of INDUG and FORMAT. As we have never before managed to gather a large number of you together in one place I would like to organize a get-together to coincide with an All Formats Show later this year (probably in Birmingham). I need help though, especially in publicizing the event.

One last thing, due to problems beyond his control Nev has been unable to produce a Help Page this month. He sends his apologies and will be back next month.

Bob Brenchley. Editor.

SHORT • SPOT

Edited By:- John Wase.

Hi, folks. I'm back again with a selection of bits and pieces; scribbles and grumbles, snippets and longer through the post and by other devious means.

My little computer garret is in chaos - even more than usual. I've at last won a major concession from my good lady, and the bed settee under the window is to go. Batons have been screwed round the wall in readiness for an eleven foot six inch piece of working top. Couple of hints for readers contemplating such a move. The good news is that you can buy little two-drawer filing cabinets from Argos at about £30.00, and they will fit neatly underneath. The bad news is that the working top is incredibly heavy: with two side walls less than three feet high before the roof starts to slope, and the place piled full of chaos, I'm just wondering if I'll ever get it up there. So watch it: if you can get working top in several bits which join easily, it helps.

At the recent All Formats computer fair, SAM was being exhibited. I picked up an advance copy of "The Secretary", a new word processor which looks as though it has a lot of features which are lacking in those currently available. There were refreshingly numerous sofware releases there, so at last there appears to be an accelerating stream of SAM software emerging. Also on display was a little add-on box which would permit the display of thousands of colours on-screen (though to my eye, a lot of them looked pretty well the same). Now such add-ons have a value which is at best limited until there is sufficient software around. I am therefore very pleased to have two items to report about the "SAMbus". You know: the backplane concerning which I had so much hassle (see "ProDOS review last month). Firstly, Guru Alan Miles tells me that he has ample supplies of an auxiliary power unit which solves the problem. I'll report on this when I've tested it. Secondly, Steve Parry-Thomas of Longton, Stoke-on-Trent has sent me a little listing for inclusion within a program. This is a procedure to print the date, month and year. "But", I hear you cry, "you can already read the date from the SAMbus using MasterDOS". That's true, and it comes up as, say, "10/06/92". Steve's new procedure prints it as "10th June 1992". Nice one! Steve writes that there's probably more than one way to do this. Perhaps one of our readers might like to come up with a way to print the day of the date. Then you could have a routine to print something like "Friday 10th June 1992". What about it , folks.

Anyway, we've got to start somewhere. So here's Steve's listing. Many thanks, Steve.

```
1000 DEF PROC  ddate x,y
1005 DEFAULT x=0,y=0
1010 LOCAL day,month,year
1015 LET day=VAL DATE$( TO 2),month=V
     AL DATE$(4 TO 5),year=VAL DATE$(
     7 TO )
1020 IF day=1 OR day=21 OR day=31 THE
     N PRINT AT x,y; day;"st";: ELSE
     IF day=2  OR day =22 THEN PRINT
     AT x,y;day;"nd";: ELSE IF day=3
     OR day =23 THEN PRINT AT x,y; da
     y;"rd";: ELSE IF day >3 OR day<3
     1 THEN PRINT AT x,y; day;"th";
1025 ON month: PRINT " January ";: PR
     INT " Febuary ";: PRINT " March
     ";: PRINT " April ";: PRINT " Ma
     y ";: PRINT " June ";: PRINT " J
     uly ";: PRINT " August ";: PRINT
      " September ";: PRINT " October
      ";: PRINT " November ";: PRINT
     " December ";:
1030 PRINT "19";year
1035 END PROC
```

Let's stick with SAM a while. I mean, there's not that many Spectrum programs been sent in this month, so I've not much choice. Chris Laird of Mayfield, East Sussex, has sent in a whole clutch for SAM. Remember L. G. Baumann's program to save multiple files ending with a progressively incremented serial number? Well, that was for the Spectrum: here's a de-luxe SAM version...

```
65000 DEF PROC Saveprog numcopies
65010 LOCAL version,progname$,j,oldpro
      g$
65020 DEFAULT numcopies=1
65030 LABEL version_line: LET version=
      1
65040 LET progname$="Test"+USING$("00"
      ,version)
65050 FOR j=1 TO numcopies: CLS
65060 PRINT "Saving    """;progname$;"
      """
65070 SAVE progname$
65080 PRINT "Verifying """;progname$;"
      """
65090 VERIFY progname$
65100 IF version>2
65110 LET oldprog$=progname$( TO LEN p
      rogname$-2)+USING$("00",version-
      2)
65120 IF FSTAT(oldprog$,1)
65130 PRINT "Erasing   """;oldprog$;""
      "
65140 ERASE oldprog$
65150 END IF
65160 END IF
65170 IF j<>numcopies THEN PRINT "Chan
      ge Disc and press any key": PAUS
      E
65180 NEXT j
65190 LET version=version+1: IF versio
      n=100 THEN LET version=1
65200 KEYIN STR$ version_line+"LABEL v
      ersion_line:LET version="+STR$ v
      ersion
65210 END PROC
65220
65230 REM If you do not have    Master
      BASIC then USING$   will not wor
      k so replace  USING$("00",versio
      n) with FN TwoFigs$(version)
65240 REM Then add this line
65250 DEF FN TwoFigs$(x)=("0" AND x<10
      )+STR$ x
```

This routine is written as a procedure and called with "SAVEPROG m", where m is the number of copies to be saved. When used with MasterBASIC, the listing can be typed separately, and then hidden with HIDE TO.

The next item is what Chris calls MENU. Now, I usually understand that this is some sort of autoloader - and I get almost one of those a month. They are the first thing that anyone writes for a computer, and they're pretty boring, except to the writer, so I rarely print them. So I flicked through this program. But it's not one of those at all. What this one does is allow you to design little pop-up menus that operate with a coloured bar and the cursor key, so that you can select one item. Nice one, Chris.

```
   10 CSIZE 8,8
   20 LET yos=-16
   30 menu 5,15,22,40
   40 DATA "Main Menu","List","Alter",
      "Load","Search","Exit"
   90
10000 DEF PROC menu numops,x,y,datalin
      e,op
10010 LOCAL title$,option$(),longest,s
      x,sy
10020 DEFAULT op=1
10040 LET sx=8*y,sy=191-8*x
10050 RESTORE dataline
10060 READ title$
10070 LET longest=LEN title$
10080 FOR i=1 TO numops
10090 READ option$

10100 IF LEN option$>longest THEN LET
      longest=LEN option$
10110 NEXT i
10130 RESTORE dataline
10140 DIM option$(numops,longest)
10150 READ title$
10160 PRINT AT x,y;title$
10170 FOR i=1 TO numops
10180 READ option$(i)
10190 PRINT AT x+i+1,y;option$(i)
10200 NEXT i
10220 PLOT sx-2,sy+2
10230 DRAW longest*8+4,0
10240 DRAW 0,(numops+2)*-8-6
10250 DRAW -longest*8-4,0
10260 DRAW 0,(numops+2)*8+6
10270 PLOT sx-2,sy-10
10280 DRAW longest*8+4,0
10290 PLOT sx-2,sy-13
10300 DRAW longest*8+4,0
```



```
10320 DO
10330 PRINT AT x+op+1,y; INVERSE 1;opt
      ion$(op)
10340 GET keypressed$
10350 IF keypressed$=CHR$ 10
10360 PRINT AT x+op+1,y; INVERSE 0;opt
      ion$(op)
10370 LET op=op+(1 AND op<numops)
10380 END IF
10390 IF keypressed$=CHR$ 11
10400 PRINT AT x+op+1,y; INVERSE 0;opt
      ion$(op)
10410 LET op=op-(1 AND op>1)
10420 END IF
10422 EXIT IF keypressed$=CHR$ 13
10430 LOOP
```

This listing, then, is a procedure to print up a menu of options and let you select one with a coloured bar. It is called with "Menu numops,x,y,dataln", where numops is the number of operations in the menu, x and y are the PRINT position for the menu, and dataln is the line number of the data statement where the options are. The number of the selected option is returned in the variable op.

Use a main program routine something like this...

```
 10 menu 3,5,5,20
 20 DATA "Title of menu","Option 1"
 30 DATA "Option 2","Option 3"
 40 PRINT "Option selected:";op
```

Chris has also sent in a nice little toolkit program. Unfortunately, it's from a magazine still in print, so is covered by copyright. Pity. Anything gone bust is fine, though - so keep on looking for snippets... Many thanks, Chris.

Now back to the Spectrum. L.G.Baumann of Cowies Hill, South Africa, is a name well known to us, and once again, he's come up with the goods. He mentions that most 48k Spectrum owners who have a PLUS D disc interface still probably use the microdrive version of Tasword 2 on their kit, since Tasword 3 has proved notoriously difficult to convert. He found that it was tiresome to have to type in the name (correctly!) of a text file to load it. Here's a simple conversion in which only the "P" number has to be typed in: not even the "P".

Just change line 2000 to read:-

```
2000 CLS: CAT md: INPUT "No of File to
 load?";z: LOAD Pz: GOTO 1
```

Note that the response to the INPUT query must only be A number (for example 17, but not P17) - then just press ENTER. The chosen text file will load and appear on the screen. Our thanks once again, Mr Baumann.

Let's stay with the Spectrum for a while. Amongst the snippets sent by Alan Cox, is one that appeared in "Sinclair Programs", 1983, the author being I. Billups of Crewe. It is a logo program - it prints your message on-screen. The effect, though, is stunning. It's well worth typing in, for it's most unusual. Highly recommended!

Here it is...

```
 10 REM LOGO program by I Billups of
    Crewe, Cheshire
 20 REM SINCLAIR PROGRAMS August 1983
 30 REM Longer than some, but worthwh
    ile in view of the effect
 40 PAPER 0: BORDER 0: CLS
 45 REM Doctored to give a particular
     message - delete line 46 for nor
    mal use
 46 LET a$="FORMAT IS BEST": GOTO 60
 50 INPUT "What is your message ?(max
    imum of 30 characters)";a$
 55 IF LEN a$>30 THEN GOTO 50
 60 LET x=INT (31-LEN a$)/2
 65 FOR l=1 TO LEN a$
 70 LET g=INT (RND*7)*30+120
 75 IF (x+l>20 AND g>200) OR (x+l<10
    AND g>200) THEN GOTO 70
 80 GOSUB g
 90 BEEP .1+(l=LEN a$)*.2,g/10
100 NEXT l: CLS : GOTO 65
110 REM ****Subroutines*****
120 FOR n=1 TO 10: REM ,
130 PRINT  INK 7;AT n,x+1;a$(1);AT n-
    1,x+1;" "
140 NEXT n: RETURN
150 FOR n=20 TO 10 STEP -1: REM S
160 PRINT  INK 6;AT n,x+1;a$(1);AT n+
    1,x+1;" "
```

```
170 NEXT n: RETURN
180 FOR n=30 TO x+1 STEP -1: REM E
190 PRINT  INK 5;AT 10,n;a$(1)+" "
200 NEXT n: RETURN
210 FOR n=20 TO 10 STEP -1: REM SW
220 PRINT  INK 4;AT n,x+1+10-n;a$(1);
    AT n+1,x+1+9-n;" "
230 NEXT n: RETURN
240 FOR n=1 TO 10: REM NW
250 PRINT  INK 5;AT n,x+1-10+n;a$(1);
    AT n-1,x+1-11+n;" "
260 NEXT n: RETURN
270 FOR n=20 TO 10 STEP -1: REM SE
280 PRINT  INK 6;AT n,x+1-10+n;a$(1);
    AT n+1,x+1-9+n;" "
290 NEXT n: RETURN
300 FOR n=1 TO 10: REM NE
310 PRINT  INK 7;AT n,x+1+10-n;a$(1);
    AT n-1,x+1+11-n;" "
320 NEXT n: RETURN
9998 STOP
9999 SAVE d1"logo"
```

Now for some fun. Last month, we had a variety of crashes all over the place. This time, we'll have an effect which makes it look as though a crash is imminent; but it isn't! All you get is a filthy racket! Here it is, with Alan's compliments, as originallly published in "Sinclair Programs", 1984; author Paul Scott.

```
 10 REM Sound Effect program
 20 REM by Paul Scott of Wimborne, Do
    rset
 30 REM SINCLAIR PROGRAMS February 19
    84
 40 REM To get sound plus striped bor
    der, type RANDOMIZE USR 28500
 50 REM I think it is a bit 'orrible,
     but still....
9000 CLEAR 28500: FOR n=28500 TO 28527
    : READ v: POKE n,v: NEXT n: RANDO
    MIZE USR 28500
9010 DATA 38,2,1,25,1,22,0,122,211,254
    ,20,124,186,32,248,11,62,0,184,32
    ,240,36,62,210,188,32,231,201
9020 BORDER 7
```

All right; all right. You like your crashes. Here's a use for one first introduced last month. On thumbing through Alan's voluminous disc, I find that the same principle is used in a program first published (again) in "Sinclair Programs" in March 1984, by B.Poulton; a password program. So here's actually a use for a crash!

```
  5 REM Password program by B Poulton
     of Stonehouse Gloucester
  6 REM SINCLAIR PROGRAMS March 1984
  7 REM Set your chosen password in l
    ine 30
 10 CLS : PRINT "Password ?"
 20 INPUT  LINE p$
 30 IF p$<>"something" THEN RANDOMIZE
     USR 1000
 40 PRINT "Go ahead"
 50 STOP
 60 SAVE d1"password"
```

The next little Spectrum snippet allows you to change the colours on screen without having to do the usual CLS business. The author, C.J.Barnett, first wrote it in 1983.

```
 10 REM Colour change program
 20 REM By C J Barnett of Fernhill He
    ath, Worcester
 30 REM SINCLAIR PROGRAMS October 198
    3
 40 REM Put these lines at the start
    of your program and you can chang
    e attributes without the usual CL
    S command
 50 REM Select the colours etc you wa
    nt by POKEing 32509 with the appr
    opriate attribute byte, followed
    by RANDOMIZE USR 32500
 55 REM For example, to get yellow le
    tters on a red background, POKE 3
    2509,85
 60 CLEAR 32500
 70 DATA 33,0,88,6,32,197,6,24,54,101
    ,35,16,251,193,16,245,201
 80 FOR f=32500 TO 32516
 90 READ n: POKE f,n
100 NEXT f
9998 STOP
9999 SAVE d1"colchange"
```

The next offering I have for you is a distinct oddity. They were submitted by Alan Cox, and I have enough problems with crashes if I start disembowelling setups, so I'm not going to try. They do look rather intriguing, though. The program is well annotated with Alan's notes. Anyone tell me what is happening?

```
 10 REM The following pair of mimi-pr
    ograms are related. They only wor
    k, at least on my 128k in 48k mod
    e, with my Disciple disconnected
    - REMEMBER NEVER TO DISCONNECT AN
    YTHING FROM THE EXPANSION SOCKET
    WITH THE SPECTRUM POWERED UP
 15 REM SO YOU WILL HAVE TO COPY OUT
    THE PROGRAM, DISCONNECT AND THEN
    TYPE THE PROGRAMS IN.  I BELIEVE
    THE EFFORT IS WORTHWHILE
 20 REM Program 1
 30 REM Letter from G Wearmouth in Po
    pular Computing Weekly 5-11 Septe
    mber 1985
 40 FOR i=64 TO 87
 50 POKE 23681,i
 60 LPRINT " Popular Computing Weekly
    "
 70 NEXT i
 80 PAUSE 0
 85 STOP
 90 REM The letter says that the prog
    rAm works best without a printer.
      It comments that the memory loc
    ation 23681,noted as 'not used' i
    n the Spectrum manual, is in fact
     the high byte of PR-CC and shoul
    d always contain decimal 91
100 REM The letter also says that the
     following program is related, bu
    t it does nothing for me
110 PRINT : COPY : PRINT
120 STOP
150 REM Program 2
160 REM There is a follow-up letter,
    but I have no record of the data
    of publication, from Nigel Clarks
    on of Durham.  This gives the fol
    lowing program
170 LET ink=32
180 FOR f=22528 TO 23296
190 POKE f,ink
200 LET ink=ink+1
210 IF ink=101 THEN LET ink=0
220 NEXT f
230 FOR i=64 TO 87: POKE 23681,i: LPR
    INT "KKKKKKKKKKKKKKKKKKKKKKKKKKK
    KKK": NEXT i: REM 31 graphics blo
    cks
240 FOR i=64 TO 87: POKE 23681,i: LPR
    INT "
      ": NEXT i: REM 31 spaces
250 REM The letter says that lines 17
    0 to 220 set up the attributes, a
    nd lines 230 to 240 (? the letter
     actually refers to three lines,
    but only gives two)
```

Now an interesting little quirk about SAM. As a user of many different systems, I often forget the quirks of one. Thus it was that I had immense problems with Chris Laird's text. All he had done was submitted it as a code file. The autoloader goes like this...

```
 10 MODE 3
 20 CSIZE 8,8
 30 CLS #
 40 SCROLL RESTORE
 50 LIST FORMAT 2
 60 MOVE "Info.txt" TO #2
```

In the first place, it jammed solid at the first scroll. I got over this by loading MasterBASIC separately, and then loading Chris's autofile: I guess a gremlin, or a system variable had crept in somewhere. Anyway, all I wanted to do was to move it to the printer. Changing #2 to #3 merely gave an error message "stream already used" I tried opening #4;"p", and using that, but got the same. Then the penny suddenly dropped. This was not an opentype file. It is a code file, and I really shouldn't be able to MOVE it anywhere... Or should I...? Full of confusion and irritation, I loaded it into the LERMed version of Tasword 2, which takes code files. It loaded and printed out perfectly.... Look at the trouble you give me if you don't send hard copy!. Anyway, does anyone know how these streams and channels really work in SAM?

And that's really all I've got for this month. Once again, I'm beginning to be short of those nice little snappy items. Please keep them coming to John Wase, Green Leys Cottage, Bishampton, Pershore, Worcs, WR10 2LX. Do remember, if you don't send 'em I can't print 'em. My grateful thanks to you-all.

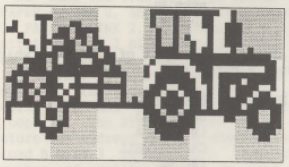
Tailpiece. I have recently had two letters from the Post Office. They both are addressed to Green Leys Cottage. They both have slight variants of my address (either is misleading if anyone were actually looking). They both implore me to use the postcode. They both tell me what it is. They contain two different postcodes.

MACHINE CODE WITHOUT THE TEARS

Part 10.

By:- Carol Brooksbank.

This month we are going to write a program which will draw a small graphic on the screen and set its attributes. It will give you a chance to use those library routines you have been writing over the past couple of months, and help you to feel confident about moving about the screen and setting colours.



The picture above shows our graphic enlarged, with the colour cells it occupies shaded. It is 6 cells wide and 3 deep, so we shall need 24 lines of 6 bytes to store the bit pattern of the design, and 18 attribute bytes will have to be set to colour it. We are going to have a red tractor drawing a black trailer, both on a green background, with BRIGHT 1 in all cells.

```
          ORG 32000
TRACBYTS: DB 0,0,0,127,249,0
          DB 7,0,0,68,49,0
          DB 66,24,0,68,51,128
          DB 194,111,0,36,19,128
          DB 34,237,0,37,19,128
          DB 19,53,128,36,147,128
          DB 10,242,192,60,177,32
          DB 7,191,96,63,255,252
          DB 30,185,224,64,255,254
          DB 46,255,176,158,112,6
          DB 127,255,248,191,55,254
          DB 209,136,136,63,183,190
          DB 123,223,248,115,250,242
          DB 85,168,136,237,214,236
          DB 91,216,138,222,215,222
          DB 254,127,255,222,215,191
          DB 133,160,0,237,255,243
          DB 133,160,0,115,131,51
          DB 6,96,0,127,128,63
          DB 3,192,0,63,0,30
          DB 1,128,0,30,0,12
          DB 0,0,0,0,0,0
          DB 0,0,0,0,0,0,
          DB 0,0,0,0,0,0
```

After ORG 32000 come the numbers which hold the bit pattern in their binary form. Each group of 6 is one line of screen file bytes, 48 screen pixels held in the 48 bits of 6 bytes.

If you compare the first row of numbers with the top line of fig. 1, you can see that there are no set pixels in the top line in the first three cells or the last one, so all those bytes will be BIN 00000000=0. The fourth byte has only bit 7 reset, BIN 01111111. Starting from bit 0, which has a value of 1, each bit value is twice the previous one, so that byte's value is 1+2+4+8+16+32+64=127, and that is the fourth number in our line of data bytes. Bit 7's value is 128, but that was not set in this number so it is not counted. The fifth byte in the top line needs BIN 11111001=1+8+16+32+64+128=249. (Bits 1 and 2 are reset, so values 2 and 4 are left out.) If you want to pursue this any further I will leave you to work out for yourself how the bit patterns compare with the data bytes in the lower lines.

```
SCREEN:  EQU 16384 (32768 for Sam users)
COLUMN:  EQU 9
ROW:     EQU 9
```

We begin with three values, the screen address - be sure to use the right one for your computer - and the column and row numbers at which we want to put our graphic. 9,9 will put it somewhere near the middle of the

screen.

```
START:   LD HL,SCREEN (Sam users see
below)
         LD A,COLUMN
         CP 0
         JR Z,DONTMV1
         LD B,A
FINDCOL: CALL NXCELL
         DJNZ FINDCOL
```

Sam users must insert a line above these:

```
START:   CALL SCREENIN
```

and delete START from the first line given here.

We begin by finding the column at which the graphic is to start. HL is loaded with the first byte of the screen, and A with the column number. If A is 0, we jump forward because we are already there. If it is something else, the column number is copied to B as a counter and we call the library routine NXCELL, which finds the next cell across, repeatedly until we arrive at the right one. We are still on row 0, but now at a different column number.

```
DONTMV1: LD A,ROW
         CP 0
         JR Z,DONTMV2
         LD B,A
FINDROW: PUSH BC
         CALL NXDOWN
         POP BC
         DJNZ FINDROW
```

In exactly the same way we find the start row. The library routine NXDOWN, which finds the next cell down, corrupts BC so we must PUSH BC before calling it and POP BC afterwards, so that we do not lose track of our counter.

```
DONTMV2: PUSH HL
         LD DE,TRACBYTS
         LD B,3
```

We save the address of the top left corner of the graphic, because we shall need it at the end when we come to find the attribute bytes. DE will always hold the pattern byte we are working on. B is loaded with the number of vertical cells in our graphic.

```
CELLOOP: PUSH BC
         PUSH HL
         LD B,8
```

This is the start of the loop which draws one complete row of horizontal cells, 8 rows of bytes. At every loop we must save the address of the byte we are on and the counter we are using. They are POPped and moved on at the end of each loop.

```
LINELOOP: PUSH BC
          PUSH HL
          LD B,6
```

This is the start of the loop which does 6 bytes in a line.

```
WIDELOOP: PUSH BC
          LD A,(DE)
          LD (HL),A
          CALL NXCELL
          INC DE
          POP BC
          DJNZ WIDELOOP
```

This is the section which actually pokes the data bytes into the screen file. The counter is saved, the data byte fetched in A from the address held in DE, and poked into the current screen file address held in HL. CALL NXCELL finds the next screen file address and INC DE moves to the next data byte. This loop repeats 6 times to do one line of bytes right across our graphic.

```
          POP HL
          INC H
          POP BC
          DJNZ LINELOOP
```

The first byte of the line just done is retrieved in HL, and INC H points to the next byte down in this cell. The counter is retrieved and we loop back. This loop will repeat 8 times, to do one complete row of cells.

```
          POP HL
          CALL NXDOWN
          POP BC
          DJNZ CELLOOP
```

We retrieve the top byte of the first cell in the row just done and use NXDOWN to find the cell below. We POP BC after using NXDOWN because it would have corrupted our counter, and loop back for the next row of cells. This loop will repeat 3 times because our graphic is 3 cells deep.

```
          POP HL
          LD B,3
ATTRLOOP: PUSH HL
          PUSH BC
          LD B,3
```

We want the trailer to be BRIGHT 1, PAPER 4, INK 0, and the tractor BRIGHT 1, PAPER 4, INK 2, so we shall set the attributes in 2 blocks of 3*3 cells. We retrieve the address of the top left corner of our graphic, load B with 3, the counter for number of cells deep, and enter the attribute loop. At the start of each repeat we store the address and the counter, and load B with 3 again, this time the number of cells across we want to colour with INK 0.

```
ACROSS:  PUSH BC
         PUSH HL
         CALL FINDATTR
         LD A,%01100000
         LD (HL),A
         POP HL
         CALL NXCELL
         POP BC
         DJNZ ACROSS
```

This is the loop which colours the trailer cells. It uses FINDATTR to calculate the attribute byte address from the cell byte address. The byte for the attributes is then put in A and poked into the attribute byte, which sets the colours in the screen cell. I explained in part 8 how the information is held in an attribute byte. Look back and be sure that you understand how this byte will produce the colours we want.

```
         LD B,3
ACROSS2: PUSH BC
         PUSH HL
         CALL FINDATTR
         LD A,%01100010
         LD (HL),A
         POP HL
         CALL NXCELL
         POP BC
         DJNZ ACROSS2
```

In exactly the same way, the attribute bytes for the tractor are set, the only difference being that INK is 2 in these cells.

```
         POP BC
         POP HL
         PUSH BC
         CALL NXDOWN
         POP BC
         DJNZ ATTRLOOP
         RET
```

Sam users must insert CALL SCREENOUT between DJNZ ATTRLOOP and RET.

After each row of cells across the graphic has been coloured, we retrieve the counter and address of the first cell in the row just done. The counter is saved again while we use NXDOWN to find the next cell down, and retrieved so that we can check whether there are any more rows to do. If there are, we loop back. The loop will repeat 3 times because our graphic is 3 cells deep.

You must now add the library routines NXCELL, NXDOWN, and FINDATTR after RET. Sam users must also add SCREENIN and SCREENOUT.

Check your assembler handbook carefully, because library routines may be handled in different ways. Some may only require you to have the routine on disc, and include an instruction to read it from the disc and assemble it at the instruction point. Others may have some sort of merging routine, but the method may vary. Usually you will find the details in a section called "BLOCK HANDLING", "MERGING SOURCE", "JOINING" or something of that sort. If your assembler will do source code merging, persevere until you have mastered the option because it will save you a lot of time. Only if your assembler has no

merging or library routine facilities should you resort to re-typing the routines.

While we are on the subject of assemblers, they all have different conventions about things like label length, whether a colon or a space is needed to separate label from opcodes, what symbols are used for BIN, whether you must use DEFB or DB, and so on. If your assembler will not accept something in the form I give it in the articles, check your handbook and make the necessary changes - shorten a label, delete a colon or whatever. If you have to shorten a label, use your assembler's FIND/SEARCH facility to find all references and calls to the label and alter them so that everything matches.

Returning to our program, after all the library routines are included finish it off with

```
END:      EQU $
LENGTH:   EQU END-TRACBYTS
```

Save the source code, assemble the routine and save the object code to disc.

Reset your computer, boot the DOS if you are using disc drives, and enter

```
CLEAR 31999
```

Load your object code to 32000, and call it from address 32145 to run it. The tractor and trailer, properly coloured, will appear in centre screen.

But what if you want to put it somewhere else? You could make the routine more versatile by changing

```
COLUMN:   EQU 9
ROW:      EQU 9
```

to

```
COLUMN:   DB 9
ROW:      DB 9
```

You would also have to change, just after the start,

```
          LD A,COLUMN
```

to

```
          LD A,(COLUMN)
```

and,

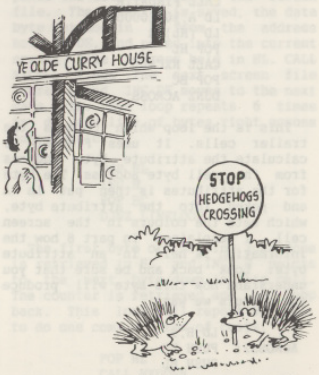
```
DONTMV1:  LD A,ROW
```

to

```
DONTMV1:  LD A,(ROW)
```

You would then reassemble your routine and save the new code. The address of START will have changed, but the symbol table will tell you what it is, and you should also make a note of the addresses of COLUMN and ROW. Then, when you have loaded the new code you could change the tractor's position on screen by POKEing new values into COLUMN and ROW from BASIC. Just remember that COLUMN must be 26 or less and ROW 20 or less so that the whole graphic will fit on screen.

You have not learned any new opcodes this month, but you know so many now that we shall be learning more methods than commands from now on. You now know how to put things where you want them on screen, and how to colour them. Next month we will look at how to get them to move.



'I think it's a great idea'

+3 DRIVING

By:- Alf Cassarubios, with additional material from John Wase.

Right: hands up all those of you with a +3 who wish that they had a second drive, but are unwilling to be ripped off by Amstrad, who charge over the top price for the FD-1. And hands up if you feel ripped off each time you buy a 3" disc at £2.99! My, what a forest of hands. Read on, then; here is the answer to your prayers.

Nowadays, 40 track single sided 5.25" drives are relatively cheap. You can pick them up for about £10.00 to £15.00 from the All Formats Fairs or Radio rallies - probably less if you haggle. Alf reckons it's better to go for a 5.25" drive, and mentions that although they are physically larger than the 3.5" drives, they have two distinct advantages. Firstly, 5.25" floppy discs are very cheap (round about 16 - 20 pence is a reasonable price). Secondly, you can use both sides of the disc, by turning the disc over and using the other side (details in a future issue). 3.5" 40 track single sided drives can also be used, but, as Alf so rightly mentions, you will need double the number of discs because you use only one side. They are dearer, too, though 'once duplicated' discs at around 20 pence are a good buy. However, John argues that if you're short of space, a 3.5" drive will sit over the existing 3" drive, if you put it in a nice little case and stick four feet on - the back ones must be rather well forward of the rather heavy power pack, so that they just sit at the back on top of the +3 case. Too far forward and the drive is unstable: too much the other way and you're well beyond the back of the +3.

The modification which follows allows you to use the external drive either as Drive A or as Drive B. It swops over the drive select lines of both drives. That way, you can use your external drive as the main drive, and this lets you store all your software on standard 5.25" or 3.5" discs. Alf would not recommend this project to the absolute novice. However, if you've a basic understanding of electronics, you should be able to cope. Here is a list of the bits and pieces you will need:-

1 x 5.25" or 3.5" 40T SS disc drive with its own power supply.
1 x 34 way ribbon disc drive lead.
1 x small double pole double throw toggle switch.
5 x 8" pieces of hook-up wire, preferably in a range of colours.
4 x small cable ties.
Solder and electrical soldering iron.
Spare half hour.

Turn the +3 upside down and remove the five screws on the bottom, and also the two screws on the right of the disc drive. Carefully remove the keyboard connectors and the power 'on' LED connector. Put the keyboard to one side.

Remove the two screws holding the disc drive. Gently lift the drive clear off the board and unplug the grey ribbon cable, noting which way round the plug fits (it's as well to mark it with an indelible magic marker). Remove the power connector (again, label it if in doubt) and put the drive to one side.

Now remove the eight screws holding down the printed circuit board (PCB), noting which screws go where. Remove the earthing strap and the plastic sheet insulator. Gently lift the PCB and turn it over so that the solder side is facing up and the EXPANSION, PRINTER and DISK ports are facing you. Warm up your soldering iron!

Whilst this is heating, have a

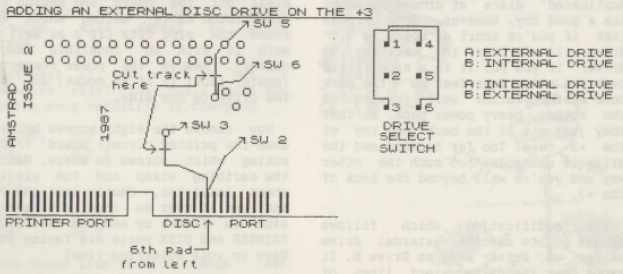
careful look at Diagram 1. This shows the PCB tracks you will need to cut, and the places where you will need to add the hook-up wires. Now for the action...

Take a small needle file or sharp craft knife and cut the appropriate PCB tracks. REMEMBER, CHECK TWICE AND CUT ONCE! Cut away about 1mm of the track. Solder the wires to the points indicated on Diagram 1: SW5 to the bottom of the internal drive connector, SW6 to the through-plated connector hole. Alf gives a useful tip here: you may have to scrape away some of the green enamel round this hole. Similarly, solder SW3 to the other through-plated connector hole as indicated on the diagram, and SW2 to the sixth edge connector pad of the DISK port, counting from the left. Make a note of which wire goes where. A different colour wire for each connection will help here. Tidy up the wires with a couple of plastic cable ties. All right so far? Good. Now we can start putting some of the bits back.

Replace the board, right way up, making sure that the disc ribbon cable is run under the PCB. Replace the earthing strap. Replace the plastic sheet insulator. Re-fit the PCB holding-down screws. Plug in the ribbon cable and the power lead to the disc drive and relocate the drive. Replace the disc drive screws. Put the bottom half of the +3 to one side and pick up the keyboard half. The next operation is on this.

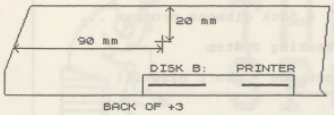
Turn the keyboard so that you are looking at the back, with the legends towards you. Mark a point approximately 90mm in from the left hand edge and about 20mm down from the top of the case (see Diagram 2). This marks the approximate position of your switch. Now check against the bottom of the case to make sure the fitted switch will not foul the internal disc drive. And if you intend to balance a 3.5" drive on top, make sure the switch is far enough to the side to be operable without knocking the auxiliary drive on the floor. Sure it's position is OK? Fine. Now drill a hole big enough to take the switch and file away any rough edges. Solder the link wires on the switch as indicated in Diagram 1. Now fit the switch to the keyboard case, put the two halves of the case back to back, trim the wires to a suitable length and solder them to the switch, again as in Diagram 1. Run the wires in a sweeping arc rather than tightly, and finish off by tidying up the job with a couple more cable ties.

Now replace the keyboard, plugging in the "power-on" LED. Next come the



Only relevant tracks shown for clarity

NOT TO SCALE



NOT TO SCALE

Diagram 2 - Drilling Diagram.

keyboard connectors. Push these in firmly but gently, taking care not to damage them. Replace the five screws on the bottom of the case, and also the two on the side of the built-in disc drive casing. Label your new drive changeover switch appropriately. Finally, check the drive select switch/link/jumper. This should be set to DS 1 (drive select 1) or, sometimes, zero, or (often) "A", otherwise the changeover switch will not work.

Testing time now! Switch on the external drive first, then the +3. You should see "Drives A, B and M available" when the drive has initialised and the start up menu has come up, though often the thing resolutely refuses to recognise the existence of Drive B. If this is the case, try switching on the whole shooting match together from one source of supply (makers of multiplugs, rejoice). Also, some drives are not recognised because a line is not held high unless they contain a disc. Use a rubbish disk for this purpose, and get in the habit of leaving it there all the time unless you actually want to use the drive. You should readily find a combination that works. Oh, and if the external drive is permanently initialised with the motor running continuously and drive LED lit all the time, switch everything off, unplug the external drive from the DISK B: port, turn the 34 way ribbon cable connector over and try again - mark with indelible magic marker if this happened. And that's all there is to it.

Finally, if you intend to use a double sided 80 track drive, then Mike Sun's software should enable you to format to this capacity. Contact BG Services for details (PD stuff; two quid for the cassette). There are, however, snags. For instance, Brian Gaff has tried this with 5.25" drives on a number of occasions without joy, though the 3.5" ones work fine, provided you remember the tip about leaving a disc in on power-up. There are 5.25" units which will work, and if you look in an Amstrad Magazine, you will find some advertised, but they are specially "tweaked". Brian tells me of a user in the North, somewhere, though, who has cracked it, and will probably be willing to supply more details if there is a demand. The moral is; beginners - 5.25"; 40T SS only, or 3.5" any capacity.

Happy disc driving.

(34)	1	Ground	(33)	2
(32)	3	"	(31)	4
(30)	5	"	(29)	6
(28)	7	"	(27)	8 Index
(26)	9	"	(25)	10 Drive Select 0
(24)	11	"	(23)	12 Drive Select 1
(22)	13	"	(21)	14 Drive Select 2
(20)	15	"	(19)	16 Drive Select 3
(18)	17	"	(17)	18 Direction
(16)	19	"	(15)	20 Step
(14)	21	"	(13)	22 Write Data
(12)	23	"	(11)	24 Write Gate
(10)	25	"	(9)	26 Track Zero
(8)	27	"	(7)	28 Write Protect
(6)	29	"	(5)	30 Read Data
(4)	31	"	(3)	32 Side Select
(2)	33	"	(1)	34

Table 1. The Shugart Standard.
(The numbers in brackets are Amstrad numbers)

'You could get rid of all that with an outside aerial, Nanny'

THOUGHT SPOT.

By:- Jeremy Cook.

Hello and welcome. This is, of course, a prize puzzle month, and so one appears below, with ONE YEARS FREE SUBSCRIPTION TO FORMAT awarded to the winner. But, as always, there are some ordinary, everyday puzzles to be tried. So try them!

Before you start, can you solve this cryptogram?

M YDE ARRE A DERST HE REI
SNONE EDT OD ECOD ET HIS

PRIZE PUZZLE No.16: BELIEVE IT OR NOT

You are a minor deity, and currently your attention is focused on a ship in the middle of a storm. On this ship are thirty passengers, and the captain has announced that to save the ship half of the passengers must be thrown overboard (a bit drastic, but it does the job!). You, as this deity, know that fifteen of the passengers are believers, but the other fifteen don't believe you exist.

The captain has proposed the passengers stand in a circle and every ninth person be cast overboard, with the counting proceeding round and round the circle of people, until just half of the thirty remain.

Being only a minor deity you can't control the storm or ship, but you can influence the positioning of the people in the circle. Naturally, you would rather the believers were saved. The trouble is, where in the circle should you position them?

Write a program to work out what order the passengers should be in so that all of the believers will be saved. Start your order from the person counted first. Your program should be as short, neat, and readable as possible. Send it in by 1st September 1992, to Jeremy Cook (Thought Spot), 6 Burgoyne Road, Sunbury-on-Thames, Middx TW16 7PW.

?evlos uoy nac ynam woH. sresop gnixelprep emos ot no woN

THAT'S ODD

Which is the odd word out? Can you find a replacement word that would fit in?

NETWORK ANTENNA ENGINEER
FREIGHTER LIONESS

THREE WISE MEN?

When the King found that his money was nearly all gone he decided on sending away most of his hundreds of Wise Men. Though fine, well dressed old men, they did tend to contradict each other when asked for advice, and they certainly knew how to eat and drink. So, on the whole, the King was glad to be rid of them. But there was an old law the King did not dare disobey, which said that there must always be:-

Seven blind of both eyes.
Ten blind of one eye.
Five that see with both eyes.
Nine that see with one eye

Question:- how many did he keep? (This is one of many puzzles devised by Lewis Carroll).

ALGEBRA?

Here is a quickie from Robert Brady. If M=3 and 5, F=2, A=4 and 8, J=1 and 6 and 7, S=9, then what are the values of D and N?

SPOTS OF BOTHER

Take a set of dominoes and select the 0/0,the 0/1, the 0/2, the 1/1, the

1/2, and the 2/2. Now arrange them in a square so that each side of the square has the same number of spots in it. Or you could try to arrange them in an equilateral triangle so that each side has the same number of spots, and so that none of the dominoes match.

Were those too easy? If so, then add 0/3, 1/3, 2/3, 3/3. Can you arrange the ten dominoes in a large square, with the same constraints as for the triangle?

ENIGMAGRAM

Four six-letter words have been jumbled. Solve the anagrams of chemical elements, and then transfer the starred letters to the key anagram to find a fifth element.

```
    MURDAI
    ??????          key anagram
K? * *   *?L        ********
L?       *?I        ????????
I?*       ?E        ________
N?        ?H
C?        ?U
E?*    ** ?M
    ??????
    BRONCA
```

A COMPETITION

Al, Bill, Col, Del, and Earl have been having a competition with each other, in which there are no ties. You are told that Al was as many places above Bill as Del was above Earl; and that neither Col nor Earl were third or fifth. Find their order.

That's all we have time for this month. Thank you for your support. I hope you'll join me again next month.

I am always happy to receive comments or puzzles, so if you have anything to say, send it to me at the address above.

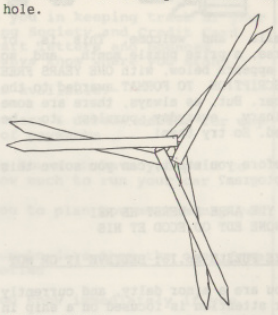
Good luck to those of you with exams looming. Cheerio.

----oOOo----

SOLUTIONS TO MAY'S PUZZLES

Keep in Touch:

The diagram below shows one way to get six pencils touching. You can balance a seventh vertically in the centre hole.



Triangulation:

Here is G.Jackson's program.

```
10 INPUT "Enter side a ";a
20 CLS: LET x=1
30 PRINT "a";TAB 6;"b";TAB 14;"c"
40 LET b=(a*a-x*x)/2/x
50 IF b<1 THEN GOTO 90
60  IF  b=INT  b THEN PRINT a;TAB 6;
   b;TAB 14;SQR (a*a+b*b)
70 LET x=x+1
80 GOTO 20
90 PRINT "That's all": GOTO 10
```

Animalgrams:

ape, gnu, seal, atoc, pinto, dingo, eland, zebra, sable, civet, orang, onager, badger, poodle, petrel, leopard, tamarin, manticore.

Z80-SUBS

MACHINE CODE SUBROUTINES.

Edited By:- Bill Nicholls.

Woe, Woe and Thrice Woe... as the late great Frankie Howard would say. Woe is me for thou art not writing to me. Well, in all truth, I have had some offerings and I am typing them in as quick as I can but I could do with lots more /SMALL ITEMS/ please.

First this month a little offering from Brian Candy in Manchester. This routine generates a random number using the following equation:-

A = (E * B + C) / D

Here is the routine:-

```
RANDOM  XOR A ;zero A
        ADD A,E ;add E to A
        DJNZ RANDOM ;B times
        ADD A,C
LOOP    SUB D
        JR NC,LOOP ;loop until < 0
        ADD A,D
```

Right, what you do is this. Load E with any number you like - it is just to start the sequence. Now load B,C and D with small numbers. Call the routine, store or print A, then copy A to E and call the routine again. Do it a few times and you will see it gives a list of random numbers. B,C and D need to be altered to give different ranges of numbers and Brian included a few examples which I wont reproduce here as I want you to experiment. Brian's routine is only one of many methods of generating random numbers, I've no doubt we will return to this topic over and over again.

Next, Frank (sorry I can't read your surname) from Brighton wants to know if readers have a routine that will allow him to test memory on a 128K Spectrum. He has one for the 16/48K machines but can't work out how to convert it to 128K operation. I'm sure one of you will have an answer.

Meanwhile. Back at the old homestead. A routine to change all the variables in a basic program listing from lower case to upper case - always a good standard to adopt as listings are more readable.

```
        LD HL,(23635) ;Point to start of program
        LD DE,(23627) ;
SKIP    INC HL ;Skip over line number and length
        INC HL
        INC HL
        INC HL
ALTERED INC HL
NEXT    XOR A ;Clear flags (especially carry)
        SBC HL,DE
        RET NC ;Return if at end of program
        ADD HL,DE ;Restore HL
        LD A,(HL) ;Get byte from memory
        CP 13 ;Is it end-of-line?
        JR Z,SKIP ;If so go to skip
        CP 14 ;Is it a number prefix?
        INC HL
        JR Z,JUMP ;If so skip bytes
        SUB 96
        JR C,NEXT ;Jump if <97 (a)
        SUB 26
        JR NC,NEXT ;Jump if >122 (z)
        ADD A,90 ;Make upper case
        DEC HL
        LD (HL),A ;Put character back
        JR ALTERED ;Loop back for next character
```

Quite a simple routine, just runs through the program skipping over the first five bytes of each line (line-number, line length and the first token) and skipping the floating point numbers stored in lines. Only letters in the range a to z are altered, so any other tokens in a line will be left unaltered.

Your programming task for the next month is to add a routine that will skip over characters within quotation marks or within REM statements. Go for it, I long to see the results.

B.S.I.Y.W.H.M.

! INLAY DESIGNER !

By:- Steven Warr.

Keeping track of what music or which computer files are recorded on which tapes is a problem I'm sure many of you suffer from, especially if you record over something and the tape inlay is already full up. This is where your Spectrum and the program below can come to the rescue.

Once you type it in and run it, there is a short pause before you are presented with the main screen. This basically consists of a grid for you to place the names of files or songs on. The current grid position is indicated by a ">". In the bottom part of the screen there is a flashing cursor waiting for you to type something. The options are now to type q,a,s or just ENTER to change the current grid position, a number between 0 and 999 to set or change the tape count, or any other set of letters and numbers to set or change the text within the current grid position. When you've finished your inlay, type p to print it out. Now without winding your printer on, answer the questions in the second screen. The first names from each side of the tape are printed again in this screen for quick reference when you are looking for a particular tape.

The completed inlay has four horizontal dashed lines - the 2 inside ones are the fold lines and the outside ones are the cut lines. The final stage is then just to cut the inlay to the right width.

The program prints the inlay by simply dumping the screen to your printer. This is done in line 5040 and so might need changing to suit your own setup. The rest of the program is fairly straight forward but it's worth looking at the user defined function set up in line 1. This function relies on the machine code set up in line 1000 and is used to print most of the text on the screen. The advantage of using this function is that you can print text in any position vertically, not only confined to character squares. The function can be incorporated into your own programs by using the command:-

```
LET P=FN P(<X-Position>,<Y-Position>,<
Size>,<Text string>)
```

The position for the text string to be printed at is set by an x and y position, in exactly the same way as th PLOT command. With <size> set as zero, only numbers can be printed - this mode is used to print the tape counts, but with it set to anything else, you can alter the depth of the text being printed. Try experimenting by typing:-

```
LET P=FN P(0,176,10,"Hello there")
```

Let me know of any other uses you find for this function. Bye for now.

```
  1 DEF FN P(X,Y,S,A$)=USR 62000
  5 CLEAR 61999: GOSUB 1000
 10 LET A$="": LET B$=""
 15 LET D$="------------------------
    -------": REM 32 dashes
 20 CLS : PRINT D$: FOR F=0 TO 2
 30 PLOT F*127.4,160: DRAW 0,-153
 40 NEXT F
 50 PRINT AT 2,2;"Side 1";AT 2,18;"Si
    de 2"
 60 FOR F=151 TO 0 STEP -9
 70 PLOT 0,F: DRAW 255,0: NEXT F
 80 LET EDIT=0
 90 IF EDIT<0 THEN LET EDIT=EDIT+30
 95 IF EDIT>29 THEN LET EDIT=EDIT-30
100 LET Z=INT (EDIT/15)
105 LET X=Z*127+1
110 LET Y=141-(EDIT-(Z*15))*9
115 LET P=FN P(X+16,Y,1,">")
120 INPUT LINE Z$
130 IF LEN Z$>12 THEN BEEP .1,0: GOTO
```

```
   120
140 IF Z$="" OR LEN Z$>3 THEN GOTO 20
    0
150 FOR F=1 TO LEN Z$
160 IF Z$(F)>"9" OR Z$(F)<"0" THEN GO
    TO 200
170 NEXT F
180 LET Z$=" 00"( TO 4-LEN Z$)+Z$
190 LET P=FN P(X,Y-1,0,Z$): GOTO 120
200 LET P=FN P(X+16,Y,1," ")
210 IF Z$="" THEN LET E=1: GOTO 250
215 IF LEN Z$<>1 THEN GOTO 260
220 LET C=CODE Z$: IF C>96 THEN LET C
    =C-32
230 IF C=80 THEN GOTO 300
240 LET E=(C=65)-(C=81)+15*(C=83)
250 IF E THEN LET EDIT=EDIT+E: GOTO 9
    0
260 LET P=FN P(X+24,Y,1,Z$+"
     "( TO 12-LEN Z$)): REM 11 space
    s
270 IF EDIT=0 THEN LET A$=Z$
280 IF EDIT=15 THEN LET B$=Z$
290 LET EDIT=EDIT+1: GOTO 90
300 GOSUB 5000: IF NO THEN GOTO 90
310 CLS : PRINT D$;AT 4,0;D$;AT 13,0;
    D$
320 INPUT "Enter Cassette Number: ";N
330 PRINT AT 1,18;"Cassette ";N
340 INPUT "Enter Tape Length (mins):
    ";N
350 LET P=FN P(24,158,2,"C"+STR$ N)
360 LET Z$="1:"+A$+"   2:"+B$
370 LET P=FN P(248-8*LEN Z$,158,2,Z$)
380 GOSUB 5000: IF NO THEN GOTO 310
390 GOTO 10
1000 LET Q=0: FOR A=62000 TO 62220: RE
     AD D: POKE A,D: LET Q=Q+D: NEXT A
1005 IF Q<>25547 THEN PRINT "Error in
     Data!": STOP
1007 RETURN
1010 DATA 42,11,92,35,126,254,36,32,1,
     35,35,205,180,51,126,35,254,44,40
     ,239
1020 DATA 205,241,43,213,197,205,148,3
     0,50,130,242,205,148,30,245,205,1
     48,30,193,79
1030 DATA 205,170,34,217,193,209,33,88
     ,39,217,217,120,177,26,19,11,217,
     200,229,235
1040 DATA 111,79,38,0,68,41,58,130,242
     ,167,121,40,25,41,41,1,0,60,9,14
1050 DATA 8,6,1,126,18,205,194,242,16,
     249,35,13,32,243,225,44,24,208,25
     4,48
1060 DATA 56,32,254,58,48,28,9,41,1,17
     7,241,9,1,15,6,203,122,203,186,40
1070 DATA 2,14,240,26,174,161,174,18,2
     05,194,242,35,16,245,225,124,238,
     128,103,230
1080 DATA 128,32,163,44,24,160,20,122,
     230,7,192,123,198,32,95,216,122,2
     14,8,87
1090 DATA 201,68,170,170,170,170,68,68
     ,204,68,68,68,238,68,170,34,68,13
     6,238,204
1100 DATA 34,68,34,34,204,34,102,170,1
     70,238,34,238,136,204,34,34,204,1
     02,136,204
1110 DATA 170,170,68,238,34,68,68,136,
     136,68,170,68,170,170,68,68,170,1
     70,102,34,204
5000 PRINT #1;"Print Screen (Y/N)"
5010 IF INKEY$="n" OR INKEY$="N" THEN
     LET NO=1: RETURN
5020 IF INKEY$<>"y" AND INKEY$<>"Y" TH
     EN GOTO 5010
5030 INPUT "": LET NO=0
5040 SAVE SCREEN$ 1
5050 RETURN
```


HACKER'S CORNER

By:- Mark Lamber .

Last month I showed how to convert a simple game, now for a more difficult one. The points which were covered previously are not fully described, so make sure you haven't missed an issue.

Operation Wolf

I needed a copy of the game to work on, so I booted the disc system and loaded the game from tape. I made a snapshot on my working disc, and renamed it "OP WOLF 0".

Next I had to find the code that does the loading, so I started a game. Once the border began to flash blue and black waiting to load level one, I snapped another copy and used the snapshot hacker to examined the stack. The addresses there were 44551, 44453, 44577, 30989, ... and so I disassembled from 50 bytes prior to each address until I found references to IX and DE.

```
30979: LD IX,44644
30983: LD DE,1
30986: CALL 44421
```

This would probably be to load one byte at 44644 using a routine at 44421. I guessed this would be a level number.

```
30989: LD HL,44626
30992: LD DE,18539
30995: CALL 42171
```

I followed this call by disassembling from 42171, and then from addresses CALLed by that routine, but I found no loading.

```
30998: LD HL,44644
31001: LD A,(44625)
31003: CP (HL)
31004: JR Z,31013
```

The level number loaded above is compared with the byte at 44625, and if it matches then a jump is made to 31013, so I skiped forward to:-

```
31013: LD (44665),A
```
Copy the level number to 44665.
```
31016: LD HL,44645
31019: LD DE,20522
31022: CALL 42171
```
This calls the same routine as after loading the level number.
```
31025: LD IX,49152
31029: LD DE,16384
31032: CALL 44421
```
Load the level data - 16384 bytes at 49152. The loading will then be complete so I stopped disassembling here.

This showed me that the loader is located at 44421, and by disassembling from 44421 I found that the routine ends at 44580, where some "junk code" is listed:-

```
44581: LD E,L
44582: LD B,L
44583: LD B,D
44584: LD D,D  ..etc
```

"Junk code" is a sequence of bytes that are meaningless as instructions. They are often LD instructions if the numbers are ASCII text, not a program. For example, this code is part of the text "PRESS PLAY".

Having saved the loader code from 44421 to 44580 (160 bytes) to disc as "OP WOLF L" I set about writing a program to copy the levels to disc. It uses the tape loader to get the header byte and level data, and then returns to BASIC to allow saving to disc.

Using an assembler I wrote this source code, "OP WOLF TS":-

```
        ORG 32000
        DI ;Ensure interrupts disabled
```

```
        LD IX,49151
        LD DE,1
        CALL 44421 ;Load level number
        LD IX,49152
        LD DE,16384
        CALL 44421 ;Load to 49152
        EI ;Re-enable interrupts
        RET
LENGTH EQU $-32000
```

I saved the machine code produced as "OP WOLF TO" which was 23 bytes long, then wrote a piece of basic to call my code and save the levels to disc:-

```
10 CLEAR 31999
20 LOAD d1"OP WOLF L" CODE : REM Loa
   der code
30 LOAD d1"OP WOLF TO" CODE : REM Tr
   ansfer code
40 PRINT "Play level"
50 RANDOMIZE USR 32000: REM Load the
    level
60 PRINT "Level number=";PEEK 49151
70 INPUT "Save as:";n$
80 SAVE d1;n$ CODE 49152,16384: REM
   Save only the level data to disc
90 PRINT "Press key for next level"
100 PAUSE 0: CLS: GOTO 40
```

I ran it, saving each of the six levels to disc as "OP WOLF 1", "OP WOLF 2" etc. I noticed that the level numbers were 177 to 182. To load levels back from disc I needed to be able to derive the filename from the level number held at 44625. The ASCII code for "1" is 49, and the level code is 177, so subtracting 128 (or ANDing with 127) gives a correct file suffix.

The next step was to create "patches" for the game to make it believe it is loading levels from tape. Firstly, I needed to replace the code at 30979 with a piece of code to fake the correct one-byte level header being loaded. This copies the correct level number to the address where it would be loaded. Notice that it only overwrites the code which it replaces.

```
30983: LD A,(44625)
30986: LD (IX+0),A
```

Secondly, the tape loader was replaced by a disc version. I used a routine based on the article on Command Codes in Format volume 2 issue 4. The disc loader does not need to know the code start address or length because these are both stored on the disc as part of a disc file header.

I also noticed that the PLUS D alters the system variables during a load. Games may use the system variables area for character sets, code or whatever, and this causes problems. I had to do this conversion twice, because at first the character set was corrupted during loading. My disc loader was around 100 bytes shorter than the original tape loader, so I added a bit of code to preserve 50 bytes of the system variables area.

The source code for the new disc loader "OP WOLF NS" looks like this:-

```
        ORG 44421 ;Replace tape loader
        LD HL,23610
        LD DE,BUFF
        LD BC,50
        LDIR ;Copy 50 bytes of system
 - variables to free space at end of
 this code
        LD A,(44625)
        AND 127
        LD (LEVEL),A ;Get 9th char of
 the name from the level number
        LD IX,UFIA
        RST 8
        DEFB 59 ;Open the file
        LD B,9
        LD DE,HEAD
LDBYT   RST 8
        DEFB 60
        LD (DE),A
        INC DE
        DJNZ LDBYT ;Load the 9 byte
 disc file header (start, length etc)
        LD DE,(START)
        LD BC,(LENGTH)
        RST 8
        DEFB 61 ;Load the whole level
        LD HL,BUFF
        LD DE,23610
        LD BC,50
        LDIR ; Copy sys/vars back
        RET
UFIA    DEFB 1,0,0,"d,4
NAME    DEFM "OP WOLF "
LEVEL   DEFM "x "
HEAD    DEFB 0
LENGTH DEFW 0
START   DEFW 0
        DEFW 0,0
BUFF    EQU $
CDELEN EQU BUFF-44421
```

The code "OP WOLF NO" that this produced was 82 bytes long, from 44421. The piece of code to fake loading the level number is short and can be poked in, but for this 82 byte section, I used the ROM routine.

I saved the 82 bytes to tape without the header, reloaded the snapshot and chose my "safe" position, where the game should not be writing to RAM. The menu screen with no music playing seemed a good place, so I took another snap.

The small piece of code produced these pokes:- 30983,58; 30984,81; 30985,174; 30986,221; 30987,119; 30988,0. Then I reduced SP by two, and changed JP to 1366, simulating a CALL to the ROM loading routine, then prepared the registers to load the code: Carry flag set, A=255, IX=44421, DE=82.

Reloading the modified snap the border flashed red/cyan indicating that it was waiting for the code, so I played the tape. After the data loaded, control was returned to the program, at the game menu.

I snapshotted this final version, and renamed it as "OP WOLF", then write-protected the disc and tested the game. The screen read:-

"Searching for level one"
"Found level one"

It thought it had, it just copied the level number.

"Loading level one"

The disc whirred and the level loaded.

The only files needed to play the game are "OP WOLF" and "OP WOLF 1" to "OP WOLF 6". These could be copied to a games disc, after that, write protect the disc.

YOUR LETTERS

Dear Editor,

Re the letter from Mr J.Quinn about the connections for the 8056 serial printer. I think he will find these to be correct.

PRINTER	COMPUTER	SERIAL PORT PIN NO.
RED	C.T.S.	5
WHITE	T.X.D.	2
BLACK (2off)	GND	1
SCREEN	ALSO GND	

If Mr Quinn contacts us, we can photocopy manual if he will pay postage and copying cost.

Yours sincerely, J.Price.

J & P Electronics Ltd,
Unit 45, Meadowmill Est, Dixon St,
Kidderminster, DY10 1HH.

Thank you J & P, your's was the first of several letters (only just beating the one below from John Taylor - it is so nice to see help forthcoming when requested. Ed.

Dear Editor,

Firstly, I would like to say that since I subscribed to FORMAT I have had my interest in computing re-kindled. And for £12 annual fee any SAM/SPECTRUM user would have to be crazy not to join! Also could you pass on to Carol Brooksbank my highest praise for the excellent way that she handles writing a series for both SAM & SPECTRUM machine code. I think she is doing an excellent job with her series (infinitely better than Kobrasoft's con. job).

Now onto the reason why I'm actually writing to you. In the YOUR LETTERS of the May'92 issue of FORMAT there were a couple of letters which gave reference to problems with using the SERIAL 8056 thermal printer. The main problem, J.Quinns one with the RS232 connections, I can help with as I use the same system. On the printer side the 2 black and one screen are all connected to "GND", therefore all but one can be cut back and the remaining one connected to speccy's black wire. The Red printer wire is the "Data" wire and should be connected to the speccy's WHITE (TXD) wire. The white printer wire is listed as CTS on the circuit board but because of the crazy RS232 system it could need to be connected to either the speccy's RED or BLUE wire I'm not too sure, try both combinations, one will work! also just to be on the safe side use the command FORMAT "p";1200 to set the baud rate first.

Also Roy Burford is (through no fault of his own) in error when he states that the SERIAL 8056 has no graphics capability. Although the manual does state it quite categorically it is I'm glad to say in error! I won't go into detail here but I will send a BASIC screen dump routine to John's SHORT SPOT which should explain, plus a few other interesting ideas, so keep an eye out there!

Bob feel free to edit out the superfluous parts, just make sure the connections are clear, this written word thingy is not my strong point! By the way keep up the good work.

P.S. Just between you and me I'm glad to see the misleading Kobrasoft Ad. has been axed, It'll save other readers wasting their time and money on falsely advertised software. (Don't print this bit unless you think its wise to do so!).

Yours sincerely, John Taylor.

I'm not quite sure what you mean about the Kobrasoft advert - perhaps you could fill me in some time.

Still thanks for your letter, I trust Mr Quinn now has all the info he needs to get his printer working. Ed.

Dear Editor,

Please can you answer this small question for me, and possibly for a few other people out there who are also wondering.

I own a Speccy +2, and I know that out of the 8 extra 16k RAM pages, banks 4-8 are contended. I know programs run slower when executed within these pages, but how much slower? It's just that I want to replay samples stored in these pages, and I'm wondering how much it'll affect the sound quality.

Yours sincerely, Martyn Bader.

This is an example of a question that really should have gone to that great guru in the north - Mr Nev. However, at fantastic expense, a quick phone call to Nev produced the following answer:- Pages 4, 5, 6 & 7 are contended and the screen is normally in page 5 or 7. Ed.

Dear Editor,

The other day I was reading back issues of FORMAT, and I found that in Vol.4 No.5 (in The Editor Speaks) you mention FORMATS promised move into book publishing. 3 Books for SAM were mentioned. Have you abandoned the idea, have the writers quit, or can we expect to see books coming in the future?

Yours sincerely, J.P.Grosen.

That is still on the cards, Carol Brooksbank has written an Advanced Users Guide for FLASH which is ready to roll. John Wase has also been working on a SAM Basic book and two other authors working on items.

Having said that, there is no fixed date for publication yet - to put it mildly there has not been much demand. We would need to sell at least 1000 copies of any book to make printing worthwhile and to make the books affordable. They will appear sometime, that's all I can say at present. Ed.

Dear Editor,

Like many people I came to computing quite recently when I purchased a second hand Spectrum. This has now been replaced by a SAM Coupé. I use the machine for several things including playing games - which I find helps me unwind at the end of the day (kicking the cat or beating the wife gets you in trouble these days).

My trouble is that, being over 40, my reactions are slowing down. This means that many of the games I purchase are just too difficult. When will software writers realize that games have to either have skill levels (including a very easy level for people like me) or they at least need to start easy and get more difficult as the game progresses.

When will there be a good Space Invaders or Pac-Man for SAM? The old games were far more playable and addictive than many of the modern offering.

Yours sincerely, Frank Wilson.

Dear Editor,

In Vol.4 no. 11, you said you would have an article on the SAMCO mouse in the next issue. Having checked back through all issues from Vol.5 no.8 to Vol.4 No.12, I cannot find any trace of this article. Could you please tell me if it is going to appear in any future issue?

A bit of news as well, CRASH has finally ceased publication, last issue was dated April.

Yours sincerely, Andrew Hall.

SAMCO went through a period (which they have only just come out of) where review samples of hardware were not sent out to magazines. It is normal policy in the magazine world that you do not review products unless a manufacturer sends a sample specifically for review (I don't make the rules I just try to stick to them). So no review mouse = no review printed.

OK, so that deals with reviews. There is however another alternative - A Readers Experience. If, after using a product for a few months a reader would like to write up his or her real life experiences of using something (hardware or software) then I'm sure other readers would be interested in seeing it in FORMAT. It is a little more difficult for me as an editor to decide if the write-up is fair and objective but if enough people write in on a particular product then I can form a majority opinion. So, come on readers, put finger to keyboard and tell the world about your favourite piece of software or that item of hardware you just would not be without. Keep things short and sweet.

As to Crash, that has now been taken over by EMAP and merged with Sinclair User. Ed.

Dear Editor,

The letter from Dr. Wilson (May '92) strikes a chord. I am another old fogie who's been fiddling about with Spectrum and now SAM for nearly 10 years and still finds programming mind-boggling.

Like him I use Masterfile and am of the opinion that it's the best and most versatile filing program I've come across. My use for it is keeping archery club handicap records and calculating tournament results.

Like him I can't get it to print from LERM's SAMSPEC, but it will print from SD's SPECMAKER. Unfortunately, it prints a little too much as it prints line 23, i.e. Recs=, Sel=, Spa=, which the Spectrum itself doesn't.

With my limited programming knowledge I've been unable to stop it doing this. Perhaps one of the young masters reading this letter can tell us if it is possible to achieve this using Masterfile's basic.

Betasoft's File Manager may well answer Dr. Wilson's needs but could well make a nasty hole is his pocket if he hasn't got the goodies necessary to make it work, e.g. MasterDos, MasterBasic, ROM 3, 512K, disc drive.

Yours sincerely, John Tracey.

It always surprises me how the older programs still have such a loyal following - Masterfile is after all of 1983 vintage. Still, if it does what you want it to do then why change. I also have no doubt that someone will spring to your aid on the printing problem John.

On the subject of File Manager, it may interest you to know that many of its features were based on the +3 version of Masterfile. It makes real use of the extra memory and disc storage that SAM has. Yes, you do need MasterBasic - but that would be a highly recommended purchase anyway. As for MasterDos, well most SAM users have upgraded to that already so the expense is not to bad - especially when you compare prices with other computers. Ed.

Dear Editor,

I have just purchased the UNI-DOS system for my PLUS D and I must tell you how pleased I am with it. Random files were what I thought disc systems were all about and I was very disappointed to find no provision for them when I got my PLUS D last year. Now, with UNI-DOS, I have been able to write simple database programs that use a whole disc to store the information.

My congratulations to SD Software, and to FORMAT for being the only magazine that supports serious users.

Yours sincerely, Alister Smith.

There is something for you this month and more UNI-DOS items coming soon. Ed.

Dear Editor,

I wonder whether I may be permitted a comment on John Wase's rather unfortunate 'review' of Pro-DOS in the May issue of Format?

Of just over 3 pages of text, slightly less than one page is taken up by what appears to be nothing whatever to do with Pro-DOS. At best this material seems to be quite out of place in this review, and at worst could be interpreted - at least on casual reading - as a serious and wholly unjustified criticism of Pro-DOS.

It does seem that John may have a problem with his Coupé, in which case it is perfectly proper for him to comment upon it in your pages, but surely not in a review! John's own column might be a more appropriate forum, and correspondence with Samco Ltd or PBT Electronics is perhaps also called for - all Coupé owners could

well be interested in the outcome. I would add that I use the SAMBUS, with the 1 Meg memory extension, the external drive/printer interface and the Hardware development kit, and all software that I have tried with this set-up, including Pro-DOS, works perfectly. Maybe I, like Andy Wright, am just lucky!

Yours sincerely, Derek Burn.

I think the review of Pro-DOS was quite fair. In fact showing that Pro-DOS highlighted the problems with the expansion shows how in-depth the reviewer went and how much Pro-DOS makes use of the system. John's comments certainly were not meant to be detrimental to Pro-DOS but if they had not been made then many readers would have run into possible problems that they would have then blamed wrongly on Pro-DOS. Ed.

Dear Editor,

Thank you for dealing with my problem of the non functioning graphics making reference to Nev Young's "Help Page".

It seems that when using programs that load and auto run, the first line of the program should be the command CLEAR, or if it is desired to use more RAM then OPEN 10:CLEAR 245000 or something similar. This means that data cannot be loaded with the program but must be SAVED separately and LOADED separately. It seems to be a case of suck it and see. If one can get away with loading data in with the program then OK but be on the lookout for possible queer results.

I am modifying my Bank/Cheque book checker program for separate loading and saving of data on the lines mentioned above.

Yours sincerely, P.J.Williamson.

Dear Editor,

I wonder if you could help me. Some time ago I purchased the DTP package from PCG. When it came it was formatted on 1.3 Dos. I had Dos 2.0 so could not get it to work. I wrote asking Samco for a 1.3 Dos disc. They replied that they didn't do it anymore. I was advised to write to PCG, which I did, twice, with no reply. I wrote to Samco again, they said to send the package to them. I did and am still awaiting a reply.

Either:- a) Do you have an old 1.3 Dos disc that (if I send you a disc) you could let me have a copy. or b) Could I run an advert in your magazine asking any of the SAM users out there if they could let me have a copy of their 1.3 Dos if I send an empty disc.

Yours sincerely, John Plant.

As far as I can tell PCG are no longer in business, as I have had several people chasing them over the last few months. I don't know why the program won't work with SAMDOS 2.0, but PCG (if they still exist) are the only people who could help you there. I'm not even sure that older SAMDOS versions will work with the latest ROM as I have never tried them. Most people are now using MasterDOS, and so far I have not been plagued with people saying DTP wont work. What is the problem when using SAMDOS 2.0? Ed.

Dear Editor,

I would like to pass on a little trick I have discovered with the demo version of Bats'n'Balls from Revelation. The trick allows you to see all the levels of the demo version of the game as supplied on the Samco news disc. I presume it will work on the full version.

When the menu is displayed select the password option and enter **TSJIET**, you will get an INVALID message but that is OK. Now play the game, press the Escape key followed by the 4 key and you jump to the next level (there are seven levels on the demo version).

* - * - * - * - * - *

Letters may be shortened or edited to fit on these pages.

This is <u>YOUR</u> letters page so it is up to you, our readers, to fill it. Send your letters, on any subject you feel would interest other readers, to our usual address, keep them as short as you can so we can fit in as many as possible.

DISC-DOC

A UNI-DOS UTILITY TO AVOID BAD SECTORS

By:- Colin Wright.

This program has been writen to work on either a DISCiPLE or PLUS D fitted with the UNI-DOS system ROM produced by S.D.Software.

The program is a simple disc doctor that identifies any corrupt or missing sectors on a disc, works out which files contain the faulty sector, and then adds an entry to the root directory which effectively removes the damaged sectors from the disc.

I have put in several REM statements that will help you understand how the program works and I have also done a reasonable amount of error trapping.

I think it could have many other features added but it already shows how powerful UNI-DOS is at handling discs.

```
  1 CLEAR 63999
  2 LET S$="CORRUPT!": LET D=1
 10 CLEAR #: GOSUB 2000
100 REM ====================
101 REM * THE MAIN PROGRAM *
102 REM ====================
103 REM
105 LINE 1000: LET P=0
108 PRINT AT 0,0;"SCANNING DISC..."''
110 DIM M(195): LET NE=0
115 POKE 23692,255
120 OPEN #4;DD;RND
130 LET RET=160: LET Q$="1470369258":
    DIM O(10)
131 FOR Q = 1 TO 10: LET O(Q)=512*VAL
    Q$(Q): NEXT Q
140 FOR P=1 TO (LEN #4)-1 STEP 5120
145 FOR Q = 1 TO 10
150 POINT #4,P+O(Q)
155 NEXT Q
160 NEXT P
195 IF NOT NE THEN PRINT "DISC OK!":
    BEEP .1,20: GOTO 750
200 PRINT #1;"PRESS ANY KEY": PAUSE 0
205 CLS : PRINT NE;" UNRELIABLE SECTO
    R";("S" AND NE<>1);":"''
208 POKE 23692,255
210 FOR F=1 TO 195
220 IF NOT M(F) THEN NEXT F: GOTO 320
225 POKE 64001,M(F): LET S1=0
230 LET C=1: FOR G=0 TO 7
240 POKE 64003,C: LET C=C*2
250 IF NOT USR 64000 THEN GOTO 310
260 LET X=(F-1)*8+G
270 LET T=INT (X/10)
275 LET S=X-T*10+1
280 LET T=T+4: IF T>=TR THEN LET T=T-
    TR+128
290 IF NOT S1 THEN LET T1=T: LET S1=S
300 LET F$=STR$ T+","+STR$ S: PRINT F
    $;"     "( TO 8-LEN F$);: REM 5 S
    PACES
310 NEXT G: NEXT F
320 PRINT : LET P=(AT DD;S$)
330 PRINT ''("CREATING" AND NOT P);("
    UPDATING" AND P);" """;S$;""" FIL
    E...."''
335 IF NOT P THEN GOTO 380
340 POINT #4,(P-1)*256+1
345 LET F=CODE INKEY$#4
350 IF F>128 THEN LET F=F-128
355 IF F>64 THEN LET F=F-64
360 IF F<>8 THEN PRINT  INK 2;Z$' INK
     0;"FILENAME ALREADY USED": BEEP
    .1,-10: GOTO 390
370 LET RET=370: POINT #4,(P-1)*256+1
375 PRINT #4;CHR$ 0;
380 OUT #4
385 GOSUB 800
390 GOTO 750
400 REM ======================
401 REM * THIS ROUTINE TRACES *
402 REM * A CORRUPT SECTOR TO *
403 REM * A PARTICULAR FILE   *
404 REM ======================
405 REM
410 LINE 700
420 LET N$="/"
430 OPEN #5;DD;N$RND
440 FOR N=1 TO (LEN #5)-1 STEP 256
450 POINT #5,N
460 LET F=CODE INKEY$#5
470 IF NOT F THEN NEXT N: GOTO 710
480 LET F$=(IN #5,10)
490 POINT #5,N+15+B
500 POKE 64003,CODE INKEY$#5
510 IF NOT USR 64000 THEN NEXT N: GOT
```

```
    O 710
 520 REM
 530 FOR X=LEN F$ TO 1 STEP -1
 540 IF F$(X)=" " THEN NEXT X
 550 LET N$=N$+F$( TO X)
 560 IF F>127 THEN LET F=F-128
 570 IF F>63 THEN LET F=F-64
 580 IF F<>12 THEN GOTO 710
 590 LET N$=N$+"/"
 600 POINT #5,N+13
 610 LET T=CODE INKEY$#5
 620 LET S=CODE INKEY$#5
 630 IF T>=128 THEN LET T=T+TR-128
 640 LET P1=(T*10+(S-1))*512+1
 650 POINT #5,N+210
 660 LET L$=(IN #5,4)
 670 LET P2=P1-1+CODE L$(1)*65536+CODE
     L$(3)+CODE L$(4)*256
 680 IF P>=P1 AND P<=P2 THEN GOTO 710
 690 CLOSE #*5: GOTO 430
 700 LET N$=N$+" ...."+CHR$ 13+"*UNABL
     E TO COMPLETE FILE TRACE*"
 710 CLOSE #*5: LET NS=N$(2 TO )
 720 LINE 1000: RETURN
 750 CLEAR #: PRINT #1;"PRESS ANY KEY"
     : PAUSE 0: RUN
 800 REM ======================
 801 REM * CREATE A FILE TO    *
 802 REM * "SWALLOW" CORRUPT   *
 803 REM * SECTORS. FILENAME   *
 804 REM * USED IS IN S$       *
 805 REM ======================
 806 REM
 810 FOR P=79 TO 0 STEP -1
 820 POINT #4,P*256+1
 830 IF CODE INKEY$#4 THEN NEXT F: PRI
     NT  INK 2;Z$' INK 0;"NO SPACE IN
     ROOT DIRECTORY!": BEEP .1,-10: RE
     TURN
 840 LET N1=INT (NE/256)
 850 LET N2=NE-N1*256
 860 LET RET=860: POINT #4,P*256+1
 870 PRINT #4;CHR$ 72;S$;"          "(
     TO 10-LEN S$);CHR$ N1;CHR$ N2;CHR
     $ T1;CHR$ S1;: REM 9 SPACES
 880 FOR G=1 TO 195
 890 PRINT #4;CHR$ M(G);: NEXT G
 900 FOR G=211 TO 255
 910 PRINT #4;CHR$ 0;: NEXT G
 920 POINT #4,P*256+245
 930 PRINT #4;CHR$ 1;: OUT #4
 940 PRINT "COMPLETE!": BEEP .1,10
 950 RETURN
1000 REM ======================
1001 REM * THE MAIN ERROR TRAP *
1002 REM ======================
1003 REM
1010 RESTORE 1500: LET ERR=(PEEK @99)
1020 READ E,L,D$
1030 IF E AND E<>ERR THEN GOTO 1020
1040 IF NOT L THEN PRINT  INK 2;Z$' IN
     K 0;D$: BEEP .1,10: GOTO 750
1050 IF L<>1 THEN GOTO L
1100 LET D$=D$+STR$ (PEEK @7662)+","+S
     TR$ (PEEK @7661)
1110 LET NE=NE+1: LET X=P-20481
1120 IF X<0 THEN PRINT  INK 2;Z$' INK
     0;D$' INK 1;("IN ROOT DIRECTORY!"
      AND P);("DISC PROBABLY UNFORMATT
     ED!" AND NOT P): BEEP .1,-10: GOT
     O 750
1130 PRINT D$
1135 BEEP .1,0: BEEP .1,-8.5
1140 LET X=INT (X/512)
1150 LET B=INT (X/8)
1160 LET K=2↑(X-B*8)
1170 LET M(B+1)=M(B+1)+K
1180 POKE 64001,K: GOSUB 400
1190 IF N$="" OR N$=S$ THEN PRINT  INK
      1;"UNUSED SECTOR.": GOTO 1220
1200 IF N$(LEN N$)="/" THEN PRINT  INK
     1;"IN DIR: ";N$
1210 PRINT  INK 1;"IN FILE: ";N$
1220 PRINT : GOTO RET
1300 PRINT #1;D$' INK 1;"UNPROTECT DIS
     C AND PRESS A KEY": BEEP .1,0
1310 CLOSE #*4
1320 PAUSE 0: INPUT ""
1330 OPEN #4;DD;RND
1340 GOTO RET
1500 DATA 131,1,"CORRUPT SECTOR: "
1510 DATA 132,1,"SECTOR MISSING: "
1520 DATA 149,1300,"DISC WRITE PROTECT
     ED"
1600 DATA 133,0,"NO DISC IN DRIVE!"
1610 DATA 146,0,"END OF FILE OR DISC!"
1620 DATA 153,0,"END OF FILE OR DISC!"
1700 DATA 0,0,""
2000 REM ======================
2001 REM * THE SHORT PIECE OF *
2002 REM * MACHINE CODE       *
2003 REM * PROVIDES A BITWISE *
2004 REM * 'AND' FUNCTION     *
2005 REM ======================
2006 REM
2010 CLS #: RESTORE 2100
2020 FOR A=64000 TO 64007
2030 READ DT: POKE A,DT: NEXT A
2035 LET Z$="FATAL ERROR:-"
2040 LET TR=(PEEK @D)
2050 IF TR>=128 THEN LET TR=TR-128
2051 REM ======================
2052 REM * SCREEN INSTRUCTIONS *
2053 REM ======================
2054 REM
2060 CLS : PRINT TAB 5; INK 2; BRIGHT
     1;" UNI-DOS DISC DOCTOR "
2065 PRINT '''  This program identifie
     s any  corrupt or missing sector
     s on a disc, works out which file
     s theyoccur in, and then adds an
     entryto the root directory which
         effectively removes the damag
     ed sectors from the disc."
2075 PRINT '"  The program takes less
     than  1.5 minutes to give a disc
      a  clean bill of health."
2080 PRINT '' BRIGHT 1; INK 1;"INSERT
     SUSPECT DISC IN DRIVE:";D
2085 PRINT #1;"PRESS A KEY TO START"
2090 PAUSE 0: BEEP .1,10
2095 CLS : RETURN
2100 DATA 62,0,230,0,79,6,0,201
9999 SAVE OVER D1"DISC_DOC2" LINE 1
```


+ - + - + - +

WRITING FOR FORMAT

FORMAT needs your contributions to fill these pages. Many readers are asking for more information on things like:-

RS232, Printer Control Codes,
SAM Sound & Graphics
Education, Hardware Construction.

Article length could be from half a page up, but two or three pages is about right for most subjects.

Articles should be sent as ASCII word processor files on disc (we can cope with 3½" or 5¼" discs from DISCiPLE, PLUS D, SAM or IBM PC). Programs should be included on the disc so we can load and test them before printing. Avoid using imbedded colour control codes and UDG characters - they make listings harder to produce and to read, use INK, PAPER and CHR$ to do the same job. NEVER use lower case L as a variable it looks to much like the number 1.

Come on, give it a go, you don't need to be an expert to write for FORMAT.

* * SMALL ADS * *

BOOKS FOR SALE: Machine Code & Better Basic £5. Spectrum Rom Disassembly £8. Introducing Spectrum Machine Code £6. Avdentures With Your Computer £2. Guide To Playing The Hobbit £2 Business Programming On Your Spectrum £5. Over The Spectrum £5. ZX Spectrum Explored £5. Call Geral Jess on 0553 810725.

SAM Coupé 512K with Disc Drive, many games + serious software. All boxed. £250 o.n.o. For more details contact Nigel Schutte on 0730-81465 (Fax 0730-81770)

FOR SALE PLUS D interface £25. MGT Life-Time Drive £65. Also various utilities, games and books for sale e.g. Tasword+2 £8. Phone Richard on 0742 475035 after 5.30 pm.

FOR SALE Disc Drive 3.5", for PLUS D etc. Very little used. Serviced by PBT, 3 months guarantee. £55. D.R.Morgan 0695 31163.

DISCiPLE Interface £40. Disc Drive, 80 track DS/DD £60. RAM Music Machine with 3 Midi leads (in, out and through) £30. Tel Adam Murdoch on 051 678 7501 for details.

YOUR ADVERT
Selling, Buying, Pen Friends, etc.

Any PRIVATE advert, up to 30 words (subject to acceptance), will be printed FREE in the next available issue. Any software sold must be original, in working order and with full instructions. The publishers will not be held, in any way, responsible for adverts in this column. Trade advertisers contact the publisher for rates.

PLEASE NOTE:- Due to limited space we can't guarantee your advert will appear in a particular issue. If your advert remains unpublished after two issues please send it in again.

PBT ELECTRONICS

P.B.T. Electronics Ltd., Unit 17, Port Talbot Workshops, Addison Road, Port Talbot, West Glamorgan, SA12 6HZ. Tel No. 0639 885008

SAM COUPÉ 512K COMPUTERS

SAM Coupé + 1 Disc Drive.....£199.99
SAM Coupé + 2 Drives.....£275.00
SAM Internal Disc Drive.....£79.95
SAM 256K RAM Expansion.....£29.95
SAM Comms Interface.....£29.95
SAM External Disc Interface...£29.95
SAMCO Mouse.....£39.99
SAM Technical Manual.....£16.99
SAM MasterDOS.....£15.99
SAM Dust Covers.....£3.99

DOT MATRIX PRINTERS

Citizen 120D (Parallel).....£139.99
Citizen 124D.....£195.50
Citizen Swift 9 Colour.....£199.99
Citizen Swift 224 Colour.....£269.99
Citizen Swift 224E Colour....£299.99
Seikosha SP1900.....£124.99
Seikosha SP2000.....£139.99
Printer Stands.....£4.49
SAM Colour Dump Software.....£8.99
(FREE on request with Swift 9/224 Colour printers)

CABLES & CONNECTORS

Parallel Printer (BBC/+D).....£7.99
Parallel Printer (+3 Type).....£7.99
Serial Printer Spectrum Type...£8.99
Spectrum Cassette Leads.....£1.99
SAM Scart to Scart.....£9.99
SAM Dual Joystick Adaptor.....£7.99
BBC/+D Dual Drive Connector...£14.99

*** SPECIALS ***

Lockable 3.5" Disc Box - 40....£3.99
Lockable 3.5" Disc Box - 80....£5.99
3.5" Bulk DS/DD Discs in 10s...£4.00
3.5" 80Trk Disc Drives.....£99.99
Citizen 120D Printer Ribbons...£3.99
MT81 Ribbons.....£4.49
Monitor Dust Covers.....£4.99
Printer Dust Covers.....£3.99
Star LC10 Ribbons - Black.....£4.59
Star LC10 Ribbons - Colour.....£7.99
3.5 or 5.25 Disc Cleaning Kits.£2.99
Printer Switch A/B.....£23.99

SPECIAL FORMAT OFFERS

Citizen 120D Parallel Printer Pack — £155.00

Citizen Swift 9 Colour Printer Pack — £225.00

Both packs include Printer, SAM Parallel Interface and Cable.
Colour Dump software also supplied with Swift 9.

MGT DISC DRIVES AND PLUS D INTERFACES REPAIRED WITH 3 MONTH GUARANTEE.
Please phone before sending repairs

BLUE ALPHA'S SAM PRODUCTS

NOW PRODUCED BY P.B.T.

THE JOY-MOUSE (works with FLASH + all other current software) £29.99

SOUND SAMPLER - Complete with software and FREE microphone £39.99

VOICE BOX - Make your SAM speak to you. £29.95

** Extra Special INDUG Members Offer **
Buy Voice Box and Sound Sampler Together For Just £54.99

All prices include VAT at 17.5%.
POSTAGE & PACKING: Printers/Computers £5.00 All other products £1.00
Make Cheques/Postal Orders Payable to PBT Electronics.
Allow 28 days for UK mainland delivery.

SOFTWARE

SAM Coupé SOFTWARE - On 3½" Disc

DRAGON TILES The thinking man's game. Always different, always taxing, the most addictive game so far written for the SAM Coupé. RRP £8.95
Order Code SCD-01 Price £6.95

SAM DEMO DISC #1 Utilities, demos, screens, a game, and lots more. A real bargain for any SAM owner.
Order Code SCD-02 Price £1.50

DISCiPLE/PLUS D SOFTWARE
Please state 3½" or 5¼" disc

ARTIST 2 The best art package ever written for the Spectrum. 48K and 128K versions on the same disc with page maker and utility software. RRP £19.95
Order Code SPD-01 Price £16.95

* * * NEW - NEW - NEW * * *

UTILITY DISC #1. Six of the best and most useful utilities published in FORMAT over the last five years. Added mystery bonus - not to be missed. Requires GDOS 3d or G+DOS 2a. RRP £9.95
Order Code SPD-02 Price £6.95

LATEST DISCiPLE/PLUS D DOS VERSIONS
Supplied on tape for transfer to disc.
G+DOS 2a for PLUS D (ROM 1a)
Order Code FST-90 Price £4.50
GDOS 3d for DISCiPLE (ROM 3) Includes new set-up program, far easier to use.
Order Code FST-91 Price £4.50

SAMSPEC SOFTWARE
SAM + Spectrum software on one disc. (3½" or 5¼" - please state on order)

FONT LIBRARY #1. 100 screen fonts plus support software, easy to use on either computer. Special +3 disc available (Spectrum files only)
RRP £8.95
Order Code SSD-01 Price £6.95

MONEY MANAGER Advanced Personal Budgeting System for both machines, Spectrum version needs 128k/+2 (not +3/+2a). RRP £15.95
Order Code SSD-02 Price £12.95

FORMAT READERS SERVICES

HARDWARE

SPI. - SAM's Parallel Interface A full feature Parallel Printer Interface. Totally compatible with SAM's ROM and DOS printing routines and with utility programs like FLASH and TASWORD 2. The SPI has the added advantage of being BI-DIRECTIONAL so allowing information to be read into your Coupé from almost ANY other computer. Uses standard BBC lead (see below). RRP £21.95
Order Code SPI Price £15.95

PRINTER LEAD Top quality parallel printer lead (BBC Standard). 2 meters long. For use with DISCiPLE, PLUS D or SAM. RRP £9.95
Order Code PPL-01 Price £7.95

FORMAT BACK-ISSUES

Volumes 1, 2 and 3 of FORMAT now available only as complete volumes.
Order Code FBI-V1 (Vol.1) Price £10.00
Order Code FBI-V2 (Vol.2) Price £12.00
Order Code FBI-V3 (Vol.3) Price £12.00
All issues, from Volume 4 № 1 to the latest issue, are still available separately. To order quote the Volume/Issue № Price £1.50 each, (5 or more £1.25 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 £4.50

- - - - - -

ORDERING

All prices include UK Postage and Packing. Overseas readers please add 10% to total order for extra postage. 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 can't send out your order. Payment in STERLING by Cheque (UK bank), P.O. Euro Cheques or Cash. Make cheques payable to FORMAT. Payment MUST accompany order. Please allow 28 days for dispatch. FORMAT Publications will not be held liable for delays or non-delivery due to circumstances beyond its control.