Problems With Printing?
EPSON CODES REVEALED

CONTENTS

NEWS ON 4

NEW DISC DRIVE FOR SAM

The shortage of the Citizen drives used in the SAM Coupé has been causing concern for several months so SAMCO has been forced to source a new drive from alternative manufacturers.

Supplies of the new drives will be available as soon as the new control board has been produced.

BRAINACHE

Brainache is the rather apt title of the latest SAM Coupé game released by the Sam Supplement team.

Described as a game of strategy and forward thinking it is supplied on disc at only £4.50, for more details ring 0922 406239 or 0543 374842.

SINCLAIR/SAM SHOW

Details have now been announced of a special Sinclair & Sam show this autumn. To be held at the Pickhurst Junior School, Pickhurst Lane, West Wickham, Kent, on Saturday 26th September. The venue is just 5 minutes from Bromley and 20 minutes from Croydon. It is easy to reach both by car and by public transport.

Stands cost a very reasonable £15 (only £10 if booked and paid for in advance) and entrance will cost just £1.50 (£1 in advance).

For more details, and for advanced tickets, contact the organizer Alex Kinch on 081-777-6241 or write (enclosing a SAE) to:- Sinclair & Sam Show, 153 Pickhurst Rise, West Wickham, Kent, BR4 0AG.

30 YEARS ON

What were you doing on July 23rd 1962? Don't remember? Well few do, until you mention the word TELSTAR.

Yep, 30 years ago this month on July 10th 1962 TELSTAR - the worlds first tele-communication satellite was launched. And on July 23rd the first two-way live TV conversation by satellite took place.

Many may now remember watching their old 405 line black and white valve TV as the BBC carried live pictures transmitted from Maine in the USA to Goonhilly Downs in Cornwall. The excitement as the first fuzzy, rolling, pictures were picked-up, was quite something to experience.

Orbiting at an average distance of only 400 miles the 3 foot diameter TELSTAR was only able to relay broadcasts for just 20 minutes at a time - 4 times a day, unlike the modern Geo-syncronus satellites which stay in a fixed position above the horizon 24 hours a day.

I just thought you might like to know what they managed to do in the old days with just 1065 transistors, 1500 diodes, 19 ni-cad batteries and 3600 solar cells. Today they could probably put TELSTAR together as a Blue Peter project. Shows how much progress we have lived through....

QL WORLD SAVED

Following the problems at Maxwell Publications the magazine Sinclair QL World was transferred late last year to HHL Publishing who continued publication until the April issue. At that time a management buy-out transfered the title to ARCWIND Publications.

The May issue did not appear amidst rumour that the magazine had closed down. However ARCWIND have now published a combined May/June issue and should be back into the normal monthly publication schedule with the July issue.

Arcwind can be contacted on 086 983 677 or by fax on 086 983 733.

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).

4

The Editor Speaks
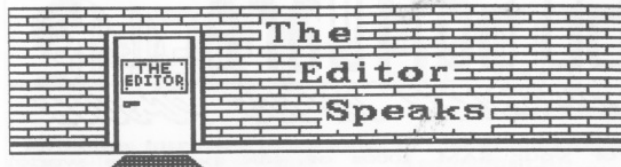
As they say, which do you want first - the good news or the bad news? Well, first, the bad news! Due to pressure of work Nev Young is going to need to take an indefinite break from producing his excellent and much consulted Help Page. The GOOD NEWS is, that although he feels his work schedule will prevent him from making the committment that a regular monthly column requires he will still be doing articles for FORMAT.

So this means I am now urgently looking for someone to help out for awhile by producing a help page. Both Nev and myself will be available to help solve problems but a person is needed to actually write the text. Please, I need your help, in order to help others. Your editor needs YOU...

I have had several letters and telephone calls over the last few months from people thinking about writing articles for FORMAT. From time to time, space permitting, I run a small "Writing For FORMAT" bit, it last appeared in the previous issue. However it seems you still want more guidance. Well, first, let me say we have been a bit disorganized in the past, but Jenny has now bullied me into setting up a proper system, so in future we should be able to track down your article much quicker if you phone up about it. Next, let me say I LOVE GETTING ARTICLES FROM READERS. Even after 16 years dealing with computers I am like a sponge - soaking up whatever new information comes my way, and your articles often contain new ideas or facts that I enjoy reading.

Right, having had my say, a few words of guidance - to answer the questions I am most often asked.

We like articles as text files on disc (5¼", 3½", DISCiPLE, PLUS D, SAM, IBM (360k, 1.2m, 720k, 1.44m) in fact in almost any common form. Files should be ASCII (in other words no control codes please) or, if you must, in Wordstar format. If the article is really good, and you are one of that handful without a wordprocessor (how do you manage) then a double spaced typed manuscript will have to do. Anyway, include a printed copy of your article so I can read it straight away and give an evening telephone number if possible. Programs MUST be included on disc, if you attempt to type more than a few lines into a wordprocessor you WILL make mistakes - I know from experience believe me.

While in the past we have not rewarded all writers this is now changing. Articles will be rewarded either by extended subscriptions, special discounts, or other agreeable means.

I'm sorry to say Thought Spot is missing this month, Jeremy is bogged down with important exams (hope you do well Jeremy). But he will be back again next month, both with Thought Spot and the first part of a short series on recursion in programs. I've seen the first part, and I'm sure it will interest all you programmers.

Please remember that I am looking for lots of letters from you for the September issue. I want to hear your reminiscence of the early days of the Spectrum. How did you get involved with computers, what trials and tribulations did you experience? Write quickly, time is short.

Bob Brenchley. Editor.

5

SHORT·SPOT

Edited By:- John Wase.

Hi, folks. I'm back again with a further selection of your short programs, items, modifications, scribbles and the like, all of which come thumping on my front door mat.

Let's start with a comment from Alan Cox. Poor Alan - he seems to have something in every issue, and I invariably manage to include an inaccuracy. For this month, he writes that I may remember being given a... er taken to task for suggesting in my column in the February 1991 issue of FORMAT that the pattern produced by M.Danielson's program was not a fractal. Alan has done a bit of work on this; like a reference to the article by Mike James on "Cellular Automata" in the May 1992 issue of "Computer Shopper". This confirms that it is a fractal. And it's called ('tis true!) a Sierpinski Gasket. This is a fascinating article:- lots of goodies in it. One item of immediate interest is a one-dimensional automaton, similar in some ways to "Life". With a one-dimensional automaton, you can plot its life-history by using the (downwards) y-axis as the time axis, which produces various patterns. The particular automaton that gives a live cell only if its neighbourhood count in the previous generation was one (note that the neighbourhood count includes the cell itself, unlike "Life"), gives - surprise, surprise - the Gasket again!

Here's a short demo by Alan:-

```
10 REM Program by ADC illustrating
   the life history of one of the c
   ell automata discussed in Mike J
   ames's article in Computer Shopp
   er May 1992
20 MODE 1
30 PLOT 127,175
40 FOR n=174 TO 48 STEP -1
50 FOR m=n-48 TO 302-n
60 LET a=POINT(m-1+(m=0),n+1)+POINT
   (m,n+1)+POINT(m+1,n+1)
70 IF a=1 THEN PLOT m,n
80 NEXT m
90 NEXT n
100 PRINT AT 18,6;"SIERPINSKI's GASK
   ET"
9998 STOP
```

Next, an offering from our evergreen L.G.Baumann, of Cowies Hill, South Africa. He uses a Spectrum with a PLUS D and drive, and finds that he has occasional difficulty with the otherwise excellent and quick "Cat-Sort" program, published in Format Vol 2(10), p11, June 1989. Its purpose is to sort the disc directory permanently into alphabetical order. Sometimes it will not work: Mr Baumann comments that he wonders if other readers have had the same problem. The problem seems to be that the program cannot handle a disc which contains erased files anywhere - even files erased beyond the last file shown in a CAT. He therefore wrote a short program which will count the total number of erased files and then substitute for them files named "zzzzzz1", "zzzzzzz2", etc. (small zeds).

It takes under four seconds to scan and report on a full disc. The SAVEing part takes the usual time as for any 1/2k program - about 30 seconds per ten saves. "Cat-Sort" can then be used - all the z files are sorted to the end. Finally, "ERASE D1"zzzzzzz*" removes all the zed files again, leaving the disc clean and in alphabetical order.

```
10 CLEAR 39999: REM gap filler
15 PRINT ''" Pgm to re-fill deleted
   files"
20 LOAD @1,0,1,40000,40: LET k=0
30 FOR q=40001 TO 60484 STEP 256
40 IF PEEK q=0 THEN GOTO 70
```

```
50 IF PEEK (q-1)=0 THEN LET k=k+1
60 NEXT q
70 PRINT ''" To fill ";k;" deleted s
   ectors'
80 FOR f=1 TO k
90 SAVE d1"zzzzzzz"+STR$ f
100 PRINT AT 8,7;"Filled gap ";f
110 NEXT f: PRINT ''"Now LOAD 'Cat-So
    rt'": STOP
```

O.K. Fine. Many thanks, Mr Baumann; a very neat bodge to get us out of the problem. Is this a common problem, though, or just an "undocumented feature" of Mr Baumann's machine? And is there a more elegant solution to this problem? I bet there is! C'mon, folks, let's see what you can come up with...

Some correspondence between Ettrick Thomson and Alan Cox, two very frequent correspondents, has arrived on my mat - they were discussing the origins of some programs... In fact, the "rose" program featured previously has appeared as a BetaBasic version in BetaBasic newsletter number 13; and owes its origins to a variety of inputs. If you're interested, then drop me a line..... Otherwise, a further letter from Ettrick on a different tack might prove more interesting. This mentions that in my introduction to Alan Cox's programming problem in last May's "Format", (you remember), a more elegant solution to "LET a$="a", b$="b", ...z$="z"), I mentioned "self-modifying code". This made Ettrick think of SAM's "KEYIN", though this was not, of course, an application. With SAM, "KEYIN" can be used quite elegantly to solve this problem, thus...

```
10 FOR n=CODE "a" to CODE "z"
20 KEY IN "let "+CHR$ n+"$=chr$n"
30 NEXT n
40 PRINT a$;",";b$;",";"z$
```

Ettrick mentions that with "KEYIN", he usually has to have several shots at it before it works properly. This was no exception. Line 20 kept on being "not understood"; things like omitting the space after "let " were wrong. The first "CHR$ n", which appears just like this, in capitals,

in the listing, was actually typed in as "chr$n"; "chr$ " is converted to a token when the line is edited, but the second occurrence, and also "LET " have to wait until the program is run before conversion. My thanks again, Ettrick.

Let's stick with SAM for a while. Some of you might have seen the Noesis program "SAMprint". I'm having great fun playing with it at the moment. So is Carol Brooksbank, who sent in this extremely useful program. What Carol immediately spotted is that there's a splendid clip art collection included with this program. Now Carol, being Carol, loves clip art. I mean, if ever she writes to you, even her headed notepaper has a clip art telephone above the number - done with the PCG desktop outfit, of course. So Carol's eyes lit up, on seeing all the goodies. And fell again, when she found the code was not in the form of the screen files used by "WordMaster", or "Flash", or any of the Spectrum art programs running under emulation, for that matter, either. Being Carol, she was not going to be beaten by a little problem like this, so she wrote a conversion program. The program as written converts the clip art in the "SPORTS" file and saves it as 53 code blocks, each a MODE 1 screen with two items on it (105 items, so the last screen has rubbish where the second item would be). The user needs a disc with 53 or more vacant slots and at least 352k to spare. Put the "SAMPrint" data disc in drive 1 and RUN. When prompted, put the disc for the screens in drive 1. The program takes around 32 minutes to save all the screens - you have been warned. Get the coffee going, ready. They should be loaded as MODE 1 screens - select SPEC COL if loading into "FLASH". Use "FLASH" if you want to convert to MODE 4. They can be used as they stand in Spectrum programs running under emulation.

The files on the "SAMPrint" data disc use the following tracks:-

Business:- Track 4, sector 1 to track 16, sector 2

Sport:- Track 16, sector 3 to track 28, sector 4
Standard:- Track 28, sector 5 to track 40, sector 6
Travel:- Track 40, sector 7 to track 52, sector 8
Custom1:- Track 52, sector 9 to track 64, sector 10

To convert the other files, make the following changes and RUN the program again:-

Line 330: change the 3 to the file's first sector number
Line 340: change the 16 to the file's first track number
Line 380: change the 17 to the file's second track number
Line 380: change the 27 to the file's last-but-one track number
Line 450: change the 4 to the last sector number used on the file's last track
Line 460: change the 28 to the file's last track number.

Custom1 has fewer items on it, so break the program to stop it when you start to get blank screens. It's best to use a new disc to save each file's screens. Many thanks, Carol.

```
10 MODE 1
20 CLEAR 39898
30 pokecod
40 readin
50 CLS : PRINT "put disc for screen
   s in drive": PRINT "press any ke
   y": PAUSE
60 LET e=IN 252 BAND 31,scrstart=(e
   +1)*16384,n=40000,p=n
70 FOR scr=1 TO 53
80 CLS : LET name$="clip"+STR$ scr
90 DPOKE 39998,0: LET x=scrstart,y=
   x,s=x
100 FOR acc=1 TO 2
110 DPOKE 39998,0
120 LET x=s+DPEEK 39998,y=x
130 FOR d=1 TO 13
140 FOR j=1 TO 4
150 LET r=p:pokeline: LET p=r,x=x+&0
    100,y=x:pokeline: LET x=x+&0100,
    y=x
160 NEXT j
170 CALL 39900
180 LET x=s+DPEEK 39998,y=x
190 NEXT d
```

```
200 LET s=s+15,x=s
210 NEXT acc
220 SAVE name$ CODE scrstart,6912
230 NEXT scr
240 DEF PROC pokeline
250 FOR z=1 TO 11
260 POKE y,PEEK p
270 LET y=y+1,p=p+1
280 NEXT z
290 END PROC
300 DEF PROC readin
310 LET start=40000
320 REM sectors on the first track
330 FOR sec=3 TO 10
340 READ AT 1,16,sec,start
350 LET start=start+510
360 NEXT sec
370 REM the complete tracks
380 FOR track=17 TO 27
390 FOR sec=1 TO 10
400 READ AT 1,track,sec,start
410 LET start=start+510
420 NEXT sec
430 NEXT track
440 REM sectors on the last track
450 FOR sec=1 TO 4
460 READ AT 1,28,sec,start
470 LET start=start+510
480 NEXT sec
490 END PROC
500 DEF PROC pokecod
510 RESTORE 570
520 LET s=39900
530 FOR a=0 TO 22
540 READ num
550 POKE s+a,num
560 NEXT a
570 DATA 42,62,156,203,28,203,28,203
580 DATA 28,1,32,0,9,203,20,203
590 DATA 20,203,20,34,62,156,201
600 END PROC
```

You know, all these wonderful things for SAM have been coming out in the last few months; like SAMprint, ProDOS and MasterBasic. Perhaps that's why the majority of programs this month are for SAM, mostly using one of these new features. Take Nigel Schutte, for instance. He's just sent in a program to demonstrate two procedures. The first is to convert decimals into fractions, though it merely leaves you with a crude fraction, like 1134 over 2385. The second procedure is to rationalize fractions, like it boils them down (what's the word I want), so that 2 over 4 becomes a half.

I checked and he's quite right. The program certainly doesn't work without MasterBasic, if only because ordinary SAMBasic won't allow a character size CSIZE=16,16. Trouble is, these programs are very long, and I can't actually think of a use for them just at the moment. Wase, however, is usually wrong, so if you out there want them printed, drop Bob a line, and I'll pass them on. Thanks, Nigel.

Of more immediate use is a pair of little programs by David Finch of Haxby, Yorkshire, who has a beautiful monogram printed out above his address (shades of Carol, here). The first is instead of the INPUT command. Now whilst INPUT is a fine piece of Basic, it has its limitations. If you try and write to the main screen with INPUT #2 and whatever, unless you're careful, you can overwrite the stuff that's already there, or get input in a place where you didn't want it. The procedure INPU x,y,characters,"Old string" can be used to input at any old print position with a set maximum number of characters. If you don't want to overwrite an old string, simply type "".

A further feature is that if you press EDIT before pressing RETURN, you can retrieve the old string: EDIT again will retrieve the new one. And after you've used the procedure, NEW$ contains what you INPUTed! (He's succeeded in confusing Wase again). David reckons it looks best in mode 3, CSIZE 8,9, with the flashing cursor an optional extra.

```
 1 PALETTE 1,127,24: POKE SVAR 8,3
   : REM Makes cursor flicker blue
 5 REM NB Lines marked * Omit if 'e
   dit' feature not wanted.
10 DEF PROC inpu x,y,z,old$: REM Sw
   op x & y if you prefer.
20 LET newold$=old$: REM *
30 DIM new$(z)
40 LET new$=old$
50 LET old$=newold$: REM *
60 LET crsr=1
70 PRINT INVERSE 1; AT y,x;new$
80 PRINT INVERSE 1; PEN 1; AT y,x+c
   rsr-1;new$(crsr)
90 GET key$: REM Or :PAUSE 0:LET ke
```
```
   y$=INKEY$
100 IF key$=CHR$ 8 THEN LET crsr=crs
    r-(crsr<>1): GOTO 70
110 IF key$=CHR$ 9 THEN LET crsr=crs
    r+(crsr<>z): GOTO 70
120 IF key$=CHR$ 12 THEN LET new$(cr
    sr)=" ": LET key$=CHR$ 8: GOTO 1
    00
130 IF key$=CHR$ 7 THEN LET newold$=
    new$: GOTO 30: REM *
140 IF key$>CHR$ 31 AND key$<CHR$ 12
    8 THEN LET new$(crsr)=key$: LET
    crsr=crsr+(crsr<>z)
150 IF key$<>CHR$ 13 THEN GOTO 70
160 END PROC
```

The next program, also by David, will load in a screen and tell you the palette values. As David says, this at first seems pointless. But it was written to solve a specific problem. That is - when you are, say, designing a sprite, using "Flash!", you would save a screen, load it from Basic, and GRAB and save the necessary string. But when you come to load the string again, there's no record of the palette settings. Use this little program to load the screen, and the palette settings will all be revealed for you to write on the label of the disc when you save the sprite. Nice one, David.

```
 10 MODE 3: PALETTE
 20 CSIZE 8,9: CLS R#
 30 INPUT "Filename:";n$
 40 PRINT AT 0,0;"Filename: ";n$
 50 LOAD n$ CODE 114688
 60 FOR a=24576 TO 24591
 70 PALETTE 1,PEEK (a+114688) LINE 1
    64-(a-24576)*9
 80 PRINT PAPER 1, PAPER 0;;"Palette
    ";a-24576;"=",PEEK (a+114688),
    PAPER 1,
 90 NEXT a
100 GOTO 30
```

Last night I saw midnight, and this morning I got up at 4.15 - was down in London by ten past seven (I had a little problem with my PC). When I got back, I continued some writing work on a paper on cellulases, checked on some work on anaerobic digestion, then got the mail. Amongst this was a letter from John Taylor. "Hi, John. First, I'd like to apologise for the delay in completing the 128k RS232 "data"... an updated version of the program of mine that you published last August..." Perleaze, if you send me long, complicated letters continuing what you did last year, do send me all the details you can. My grey matter boggled at this. So, for now, I've sorted through it, and extracted one wonderful tip for you: much of the other stuff, I'll need to think about.

Who plays games and finds that they are set for the reactions of the very young? Do you ever use a DISCiPLE - not a PLUS D; a DISCiPLE, with the network ports. Got one? Good. John writes that quite a while back, whilst he was disassembling GDOS, he noticed that the shadow RAM address @6978 was output to I/O port 31 on each interrupt. That in itself is fairly well known. The novel bit is that when the network is enabled by FORMAT N1 and POKE @6978,128 is executed, GDOS slows down the computer quite dramatically because it now has a much increased work load during each interrupt. John mentions that it makes fast games like "Bumpy" a doddle to play. A further little feature is that when the disable button on the DISCiPLE is pressed, everything returns to normal speed and the DISCiPLE no longer gets paged in. It's a two-speed "slowmo" for nothing! Many thanks, John. I'll try and get in touch to find what the other bits and pieces are all about if I can't fathom them out.

One or two tips to end with. I expect you all know by now of my continuing reorganization of my computers. One of the results of that is that a telly now sits on top of a taller cupboard than previously, and the composite video lead from the grey +2 is no longer able to reach - I have to use an extended aerial cable from the modulator socket, the signal then being demodulated as soon as it hits the Speccy. All this messing about doesn't half degrade picture quality. I therefore pity those who have a +2a or a +3, which no longer is able to use the composite video, and has to use the ordinary aerial. The +2a and the +3 seem to give off much more radio frequency interference than the older models. This is not filtered out by the TV, either, so extensive signal degradation occurs. The cure is to get a ferrite core, like from Tandy's, who have a particularly easy one to fit: part number 273104. Fit this on the signal cable as closely as possible to the computer, wind the cable some five or six turns, and voila - the interference is reduced to very tolerable levels.

Whilst we're on the +2a and +3, did you know that if you have a +2a and type COPY RANDOMIZE; press ENTER and keys "p", "l", "c" and "z" very quickly, it will tell you it's a +3? Alternatively, just try POKE 23398,180. Press EDIT and select EXIT. Your machine has changed. Wow!

One of the many snags about a +2a is that that nice printer port with which it is adorned - you know: the centronics one - is dumb, mute and so much immobile ironmongery when the beast is in 48k mode. Here's how to change all that. Save the following magic to tape.

```
9999 LET err=23613: LET re=FN q(FN q(e
     rr)): LET re1=23354: LET re2=4867
     : POKE FN q(err),(3 AND re=re1)+(
     58 AND re=re2): POKE (FN q(err)
     +1),(19 AND re=re+1)+(91 AND re=r
     e2): DEF FN q(x)=PEEK x+256*PEEK
     (x+1): STOP
```

All you need to do is to run this and it will toggle the +2a between 128 and 48k mode without altering the information in the channels area, or disabling the pageing, since the bit is not set. It therefore allows the centronics port to function whilst the brute is in 48k mode - very nice for running some of the older programs.

And that's really all there is to it this month. I'm running out of Spectrum snippets once again, so keep rummaging through those old mags that you keep in the attic. Send the results to me, John Wase, at Green Leys Cottage, Bishampton, Pershore, Worcs WR10 2LX. Bless you all.

SAM PRINT

(Posters, Cards And A Lot More)

Reviewed By:- Carol Brooksbank.

Noesis have produced an excellent program for printing posters, headed stationery and greetings cards. It comes with two discs, one the working program and the other a data disc with the libraries of fonts and graphics needed to design your artwork. The first surprise is that there is no manual, but you don't need one. The program is extremely easy to use, and all the necessary instructions appear on screen.

All printing is on A4 paper. The posters and stationery are straightforward printouts, but the card printing is very ingenious. The "cover" and "inside" of the card are printed in opposite corners of the paper, top to top, so that when you fold the sheet in four you have a neat card, properly laid out (see Fig. 1).

A choice of decorative borders is available for cards and posters, and the size is adjusted to suit what you are printing. Figs. 1 and 2 show the Art Nouveau and Christmas borders respectively - there are others, including a delicate lace, various thicknesses of plain line, marching ants, footprints, cars and an explosion effect.

On posters and headed notepaper you may select up to two graphic motifs from the very extensive library supplied. A card may have one graphic on the cover and one inside. Graphics may be used as an all-over fill, placed diagonally, or cut-and-pasted exactly where you want them. Three sizes of reproduction are available. The exception is in stationery, where the graphics are always

small and placed in the corners of the paper - one at the top and one at the bottom (see Fig. 3).

There is quite a range of type faces supplied, Gothic, handwriting effect, Roman, Western, and several others. I should have liked a wider range of sizes. There are two very tiny ones and all the others are pretty large, so a choice of halfway in between fonts would come in handy. In Fig. 2 I could not word the menu as I really wanted to because I could not get any more words in and still leave spaces between the courses. There was no room for a fourth course.

The graphics library is superb,

EIGHT till LATE

October 23, 1991

Jane Smith

Fig 1. A Card Layout.

Fig. 2. A Full-page Menu.

although there is rather an American bias - mail boxes are US shape, baseball dominates the sport section, and we have all the States in outline with their emblems. But this is perhaps a nit-picking criticism - there are a huge selection of standard items - Christmas trees, graduates, cats and dogs, gifts, flowers, business motifs, travel items (Big Ben, Notre Dame, the Great Wall of China etc.), and many more, all of which print beautifully in all three sizes. Bruce Gordon tells me that enhancements are planned, so I hope we can look forward to more data discs and an even bigger choice of graphics and fonts.

There is also a graphic editor, which lets you design your own motif from scratch or modify any of the existing ones. Four "custom" libraries are supplied, three of which are

totally empty and one of which has only a few items in it, so that you can save your own designs in the special library format needed by the program. The editor is very easy to use, and allows designs to be mirrored vertically or horizontally and inverted. I should have liked the cursor to repeat if held down. You can toggle it to set or reset pixels automatically as it moves, or to allow you to change pixel status manually, but you have to press a cursor key for every one-pixel movement which makes it a bit slow. Fig. 4 shows the "Oklahoma" emblem modified by removing the state outline map to leave just a covered wagon.

Print quality is fine so long as you have a fairly new ribbon. The head makes only one pass per line so the output of an old ribbon gets a bit pale. I understand that the possibility of a double pass printout is being looked at for a future upgrade of the program.

I have only one major criticism of Samprint. You are encouraged to make a copy of the data disc, but it is not possible to make a backup working copy of the program disc using COPY or BACKUP. As a software writer, no-one is keener than I to stamp out software theft. I am all for making games programs un-copiable. But this is a serious program and serious users are often up against deadlines. If someone asks me for 50 posters, and may want a repeat order in a day or so, I should hesitate to use a program for which I had no backup copy. Accidents do happen, and although a corrupted disc can be returned to the publishers for replacement, deadlines don't leave time for that. I firmly believe that all serious programs should allow the buyer to make a working copy for his own safety. There are ways of making a program capable of producing one or two working clones of itself, and no more.

14

Fig. 3. Top & Bottom for Headed Paper.

But this is one area where, in the end, service to the customer should override the concern to make theft impossible.

That apart, I like this program very much. It produces neat and attractive items. The decoration is versatile - you decide whether you want borders, text or graphic items, and you can preview your design, change it if it doesn't suit, save it to disc and reload it. Even at the printing stage, you can return to the layout design option without printing if you spot something you want to change. It is absurdly easy to use. I thoroughly recommend it. You will find it very useful and you will have a lot of fun with it.
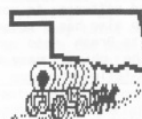
Fig. 4. Edited Graphic.

SAM PRINT costs £14.99 and is available direct from Noesis Software, Unit 12, Oxwich Court, Fenrod Business Park, Valley Road, Swansea, SA6 8QP.

15

Adventure Trail

By:- Adam Scott.

In the absence of Paul Rigby, who has since moved on to the Editorial team of a PC magazine, Bob has asked me to take on the challenge of producing an adventure section. Just what is he letting you in for, I wonder?

As Format is largely geared towards more serious applications of Spectrum and SAM computing, this section will be avoiding reviewing Adventure titles, I'm not saying there wont be the odd review - but not page after page. A number of dedicated publications exist for this purpose, and we will look at these another time. Instead of concentrating on reviewing which most columns of this type would, I want to take a wider look, at the various aspects of Adventuring, in each column.

But what if, like so many Format readers, you have never played an Adventure? Many people will be weary of playing an Adventure, either because they think of them as "games" - in much the same context as "Space Invaders" or suchlike. Perhaps you may be under the impression that all Adventures merely contain "Dungeon and Dragons" or similar? Whilst both of these impressions have some merit, they are on the whole incorrect, since Adventures can be thought of as much more than mere "games", and the subject matter can vary dramatically.

So what is an Adventure?

An Adventure can be considered as an "Interactive Story". A well-written Adventure will effectively create a world within the confines of your computer, and will involve you in an unfolding tale of this world. The "world" could be a futuristic society, an alien civilization, ancient Rome, the lost city of Atlantis, or even a goldfish bowl! However, wherever the Adventure is set, there is one thing that will always remain constant; somewhere, somehow, something is amiss and it's down to you to sort it out.

So how do you do this? How do you control the events in this world? Most Adventures will put you in the position of either a character in this world, or as a "puppet master" - controlling the actions of a character.

Whichever way, you will be controlling a character on his/her/it's quest.

The nature of the quest will depend entirely on the environment the Adventure is set in. You could be co-habiting with characters such as Sherlock Holmes (perhaps preventing Professor Moriarty's latest dastardly plan), Robin Hood (rescuing Marion from the Sheriff), Maid Marion (rescuing Robin Hood from the Sheriff!) or even MacBeth! The only limits are purely those of the programmer's imagination, and of any copyright laws!

The normal method of control is via a keyboard. Most Adventures will only need you to type in 2 word commands, in the form VERB NOUN, such as:-

GO NORTH, TAKE KEY, UNLOCK DOOR
OPEN DOOR, GO EAST

or in some rare cases, the commands could be as complex as:-

PLANT THE PLANT IN THE PLANT POT

But complex commands like this are normally very rare indeed!

The actual words used by an Adventure, its Volcabulary, are fairly

17

standard between each title. Directions are usually NORTH, SOUTH, EAST, WEST, UP, DOWN, NE,SE,SW,NW. Commands for manipulating objects are normally: TAKE, DROP, EXAMINE, READ. And of course, any decent title will allow a number of variations of each word.

In future columns, I plan to look at the two main aspects of Adventuring - playing and programming. How do you go about playing your first Adventure? What sort are available? How do you program an Adventure? What methods are available? I hope to be answering all of these questions, and more, over the coming months.

But I would like to hear from you, the reader, to discover which areas you would like me to cover. Would you like to have a series of articles giving tips on solving Adventures? How about interviews with famous Adventure programmers? Tips on using Adventure creation systems? Let me know what you would like to read about, and I will try to cater for your wishes.

Additionally, if we get sufficient interest, we may be putting together a special Adventure collection disc. If you would be interested in this, or have any other comments, please drop me a line at the usual Format address.
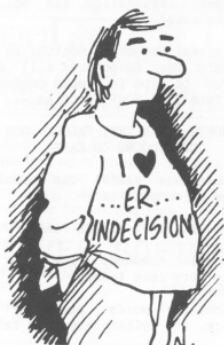
A +2a FIX

Reviewed By:- Stephen Baines.

For 2 years now, I have had a +2a. Prior to this I had owned an issue 2 Spectrum 48k. The delight of owning the brand new +2a was partly tempered by its non-standard expansion port and a certain lack of compatability with the older models. A year ago I bought a PLUS D, the special edition for the +2a. This works fine, but certain programs still didn't work, and certain "anomolies" occur, as the leaflet with the PLUS D put it.

These were annoying "anomolies". After loading, a series of two pokes had to be typed in (or put in an Autoload file) to make it realise there was 128k RAM. This was nothing compared to the problems if an error occurs then the +3 editor leaves and you are suddenly exiled in 48k basic for eternity, or until you can find the keys needed to save the program you were working on. Other "anomolies" included a snapshot routine that didn't work all the time.

Then, out of the blue, came an advert for BG Services' +2a Mod Kit. The kit sounded steep at first (at £18.50) so I decided to confirm all my worries with them. Back within 2 days came a letter to say it was easy, just one chip to plug in and a Fixer2 to use with it. I sent off my cheque, and within a week the kit was with me.

I opened the package. Inside was a modified Currah µslot and a new ROM chip, plus the instructions. All you need is 2 screwdrivers and a little time.

First stage remove the screws in the bottom of the +2a, swing up the lid and unplug the cassette unit. Easy. Locate the two chips at the back, note which came from where, and remove them. This needs a small screwdriver to prise them up. Keep them safe in

case you ever need to reinsert them. The next stage is to put in the new ROM chip according to the instructions, and then re-build the machine and test that it powers up.

Up came the reasuring 128 start up screen. Screw everything together again, and attach the µslot to the edge connector. The µslot is a device created by Currah about 8 years ago to allow two or more addons to plug into the Spectrum. Then, as now, many hardware devices didn't have a through port to "stack" them. When fitted it allows your PLUS D or other things to plug in as before, but now the Bodgit supplied with the PLUS D can be discarded for the neater and more secure µslot which has already been modified to correct Amstrad's muck-up of the expansion port.

I found the PLUS D lives quite happily pointing up in the air, allowing the red light to be easily seen, and regain those six inches from the rear of the computer, which left it dangling over the edge.

All set, I decided to check that all functioned as before. Power up, select 128 Basic, and type. This was the first difference, on the +2a and +3 on entering basic the first character takes 2 or 3 seconds to appear on the screen, in 128 Basic it appears immediately. This makes it all the easier to type in programs. Next type in a deliberate PLUS D error, like "Cat2" with no drive 2. Up came the error message, "oh no", I thought, as I expected to be now in 48k Basic - press a' key, and I am still in 128 Basic. No more searching for the key words or losing the program.

On running programs, such as a bubble sort, there is a large difference in speed, typically about 20% faster. The Basic syntax for the RAM Disc is different but less complex although the CAT! instruction doesn't tell you how much space is left now. This can be found by PEEKing one of the original 128 set of system variables. Clever people may even be able to make a command and lodge it in

the PLUS D as an extra Basic command.

On trying the cassette mechanism I noticed a funny grinding noise. I panicked and eventually traced the problem to be a post-it note I had put inside the computer to tell me where to put the old chips if necessary.

The real test is the "Bad Tempered Program Test". The ultimate for +2a/+3 users is "Bombjack". This masterpiece flatly refuses to work on a +2a, no matter whether loaded in 48 Basic, using the port lock out or many other attempts, yet loaded everytime on my 48k Spectrum. On the +2a it crashes as soon as it has finished loading, on using with the 128 chip fitted the program loaded effortlessly. Test whether it crashes on pressing the snapshot button; no, it still works perfectly. Now the test of the 128k bug-bear, Nigel Mansells Grand Prix ALWAYS crashed (no pun intended) on pressing the PLUS D's snapshot button but now works perfectly on the modified +2a.

The only program I haven't had luck with is "Bubble Bobble", which I can now only conclude actually senses the presence of the PLUS D.

All is not a bed of roses, however, the +2a's parallel printer port no longer works, whether it can be driven with OUT instructions I don't know, and also the Cassette Header Reader is lost. Any PLUS D user will, I would imagine, use the PLUS D's printer port anyway, and a header reader can be found in back issues of FORMAT. Would I recommend the modification kit? I certainly would, and anyone wishing to use the PLUS D without the "anomolies" would be advised to buy it immediately. Anyone who owns an "old" 48k machine and wants to up grade, but can't find an "old" +2 or 128, can buy a +2a and the kit in confidence.

There are other advantages too, I can now buy a copy of BetaBasic without being limited to 48k basic only, as you are on the +2a/+3 ROM's.

Where is my cheque book now?

18

19

EPSON STANDARDS

IS YOUR PRINTER EPSON COMPATIBLE?

By:- Alan Cox.

How often have you read (or heard)
the phrase 'My printer is
EPSON-compatible' immediately followed
by 'but.....'? The dots covering a
variety of complaints where the
particular printer in question does
not behave as an EPSON-compatible
printer should do. And what about
those advertisements for 'EPSON-
compatible' printers?

Now it may well be that everyone but
me knows what 'EPSON-compatibility'
really means - in which case, of
course, this article will not be
published, so I am wasting my time.
But in the belief that there may be
others who would welcome
enlightenment, let me go ahead.

My desire to know about this subject
has a practical purpose, as well as
the pursuit of knowledge for its own
sake. I have a Brother HR5 thermal
printer, which accepts quite a few
EPSON commands, but not as many as
other peoples', so I often need to
know what a particular command in
someone else's program is meant to do,
since sometimes the HR5 offers an
alternative - for example, I have no
margin command, but Horizontal Tab is
a possible, if less flexible,
alternative. Thus I need at least a
list of EPSON commands so that I know
what the intention is. I have done the
obvious thing and written to EPSON and
asked them the question. It is on the
basis of the information that they
have provided me with that I have put
together this article, which goes
further than I had originally
intended. I am very grateful to EPSON
(UK) Ltd for providing the information
and for agreeing that I can use it
here. In some areas I felt there was
little point in paraphrasing
statements that appear in EPSON
documents, and I have therefore used
their words or something very like

them at various points - when this
occurs I have put the words in quotes.

The nub of the problem is that the
phrase 'EPSON-compatible' tends to be
regarded as an absolute statement of a
printer's capabilities, whereas in
reality it is a more general comment.
Perhaps a good analogy is the
statement "I speak French". This is
all very well as far as it goes, but
of course it can cover anything from a
faint memory of schoolboy French (my
own situation!) at one end of the
spectrum to being as fluent in the
language as a well-educated Frenchman
at the other, with of course
everything in between as well. The
situation with EPSON-compatibility is
very similar.

THE EPSON LANGUAGE.

"EPSON have defined, for use with
EPSON and EPSON-compatible printers, a
collection of commands" (to use the
above analogy, a 'language') "known
collectively as the Epson Standard
Code for Printers", which very
conveniently abbreviates as ESC/P
(clever, eh!).

"EPSON have constructed the ESC/P
system from commands that can be
classified hierarchically into 5 major
levels. Each printer model is designed
to support the commands from a
specific level within that heirarchy"

"The advantage of such a system is
that printers and printer software can
be classified in accordance with the
levels they support. The levels have
been designated by Epson as follows:-

        ESC/P-84
        ESC/P-83
        ESC/P-82
        ESC/P-81
        ESC/P-80

The highest level is ESC/P-84. The other levels each consist of a subset of the level above it. For example, level ESC/P-82 will contain all the commands in level ESC/P-81 and some additional commands". Broadly speaking, the different levels provide the following capabilities:-

"ESC/P-80 contains the most basic commands required for data processing and graphics, such as carriage return, form feed, horizontal tabbing (pre-defined tab-spacing only)".

"ESC/P-81 includes additional commands to level ESC/P-80 to provide efficient standard data processing, such as vertical tabbing, back spacing, select superscript/subscript characters, select international character sets, horizontal tabbing (specifiable tab-spacing)".

"ESC/P-82 includes additional commands to level ESC/P-81 to provide basic word processing, such as set absolute and relative dot positions, user-definable characters, set inter-character space, select quality of printing".

"ESC/P-83 includes additional commands to level ESC/P-82 to provide advanced word processing, such as select proportional spacing, select typeface, specify text justification".

"ESC/P-84 includes additional commands to level ESC/P-83 to provide advanced data processing, such a select double-height printing, extra international character sets are available"

The majority of commands fall into one of the above categories. However, some commands are peculiar to a particular printer model (or models) - these commands are considered to be from a sixth class referred to as ESC/P-ext (presumably meaning 'extra' or 'extension')

It follows that any given command can be annotated with the particular heirarchical level that first supports it (I use 'first' since there is an obvious temptation to see the levels as advancing in time. I think this is the case, although I have no reason to believe that the numbers attached to the levels in the hierarchy are directly date-related).

I have produced a table to illustrate this. Below is a list of all the 92 commands that comprise ESC/P in an ASCII-determined order, showing briefly what the command does, and also its ESC/P level.

ESC/P COMMANDS BY NAME

COMMAND	FUNCTION	LEVEL
BEL	sound beeper	P-81
BS	back space	P-80
HT	tab horizontally	P-80
LF	line feed	P-80
VT	tab vertically	P-81
FF	form feed	P-80
CR	carriage return	P-80
SO	select double-width printing (one line)	P-80
SI	select condensed printing	P-80
DC1	select printer	P-ext
DC2	cancel condensed printing	P-80
DC3	deselect printer	P-ext
DC4	cancel double-width printing (one line)	P-80
CAN	cancel line	P-ext
ESC SO	select double-width printing (one line)	P-80
ESC SI	select condensed printing	P-80
ESC EM	cut sheet feeder control	P-ext
ESC SP	set inter-character space	P-82
ESC !	master select for typestyle/pitch	P-81,P-83
ESC #	cancel eighth bit control	P-ext
ESC $	set absolute dot position	P-82
ESC %	select/deselect RAM based char-set	P-82
ESC &	define user defined characters	P-82
ESC (-	define and apply a style of scoring	P-ext
ESC *	general bit image command	P-81,P-ext
ESC +	set n/360" line spacing	P-ext
ESC -	turn underlining on/off	P-80
ESC /	select vertical tab channel	P-ext
ESC 0	select 1/8" line spacing	P-ext
ESC 1	select 7/72" line spacing	P-ext
ESC 2	select 1/6" line spacing	P-80
ESC 3	set n/180" line spacing (24 pin printers)	P-81
ESC 3	set n/216" line spacing (9-pin printers)	P-81
ESC 4	select italic typestyle	P-83
ESC 5	cancel italic typestyle	P-83
ESC 6	expand range of printable characters	P-84
ESC 7	cancel expanded range	P-84
ESC 8	disable paper out sensor	P-ext
ESC 9	enable paper out sensor	P-ext
ESC :	copy ROM based char set to RAM	P-82
ESC <	uni-directional printing for one line	P-ext
ESC =	clear eighth bit of incoming data	P-ext
ESC >	set eighth bit of incoming data	P-ext
ESC ?	re-assign bit image commands	P-ext
ESC @	initialise the printer	P-80
ESC A	set n/60" line spacing (24-pin printers)	P-80
ESC A	set n/72" line spacing (9-pin printers)	P-80
ESC B	set vertical tab stops	P-81
ESC C	set page length in lines	P-80
ESC C NUL	set page length in inches	P-80
ESC D	set horizontal tab stops	P-81
ESC E	select emphasized printing	P-80
ESC F	cancel emphasized printing	P-80
ESC G	select double-strike printing	P-80
ESC H	cancel double-strike printing	P-80
ESC I	set/cancel redundant control codes	P-ext
ESC J	perform n/180" line feed (24/48 pin)	P-81
ESC J	perform n/216" line feed (9-pin printers)	P-81
ESC K	select 8bit single-density bit image mode	P-80
ESC L	select 8bit double-density bit image mode	P-80
ESC M	select 12-pitch characters	P-81
ESC N	set for skip-over perforations	P-81
ESC O	cancel skip-over perforations	P-81
ESC P	select 10-pitch characters	P-81
ESC Q	set right margin	P-81
ESC R	select international char set	P-81,P-84,P-ext
ESC S	select superscript/subscript mode	P-81
ESC T	cancel superscript/subscript mode	P-81
ESC U	turn uni-directional printing on/off	P-80
ESC W	turn double-width printing on/off	P-80
ESC Y	select 8 bit double speed bit-image mode	P-81
ESC Z	select 8-bit quad-density bit-image mode	P-81
ESC \	set relative dot position	P-80
ESC ^	set 9-pin graphics mode	P-ext
ESC a	select justification mode	P-83
ESC b	set vertical tabs in channels	P-ext
ESC e	set vertical/horizontal tab spacing	P-ext
ESC f	perform horizontal/vertical skip	P-ext
ESC g	select 15-pitch printing	P-ext
ESC i	turn incremental print mode on/off	P-ext
ESC j	perform n/216" line feed (9-pin printers)	P-ext
ESC k	select font family	P-83,P-ext
ESC l	select left margin	P-81
ESC m	select special graphics characters	P-ext
ESC p	proportional character spacing on/off	P-83
ESC q	select/cancel outline/shadow printing	P-ext
ESC r	select printing colour	P-ext
ESC s	turn half speed mode on/off	P-ext
ESC t	select character table	P-84
ESC w	turn double height printing on/off	P-84
ESC x	select draft quality or NLQ/LQ	P-82
DEL	delete character from buffer	P-ext

I hope the above gives sufficient information to enable you to trace any particular command.

To what extent does the ESC/P language help to answer the questions I posed in my first paragraphs, which was in effect how to describe precisely the capabilities of any particular printer? Let me say straightaway that in my view it solves 95%+ of the problem, but there are still a few loose ends which need to be watched. Clearly the only guaranteed specification is to spell out all the commands that the printer obeys. This long list can be usefully abbreviated using the ESC/P heirarchy, but there are still the loose ends. I have a list of 26 EPSON printers, all of them support a particular hierarchy level, but in every case there are some ESC/P-ext commands available as well. Further, only 12 of them support the specified hierarchy level without exception. Then again, manufacturers other than EPSON, even those that provide EPSON-like commands, sometimes also include their own 'private' codes - an example of which I am aware is CITIZEN.

Thus, while I repeat that the EPSON ESC/P language is a major tool for specifying printer capabilities, the ultimate test has to be a full list of commands supported.

In order of usefulness, I would prefer first the full list, secondly a statement of the hierarchy level supported, thirdly a statement of what EPSON printer is emulated and fourthly and very much the least useful, a statement of 'EPSON compatibility.'

THE FUTURE.

What of the future?. Although I did not realise it at the time when I started, now is an opportune moment to publish this note. As those of you who are avid readers of advertisements in magazines may have noticed, EPSON have recently launched a new range of LQ printers (the LQ 570 was very well reviewed in the November edition of Computer Shopper - it was rated above low-end laser printers) which make use of the ESC/P 2 language.

The advertisement says (please note that I have no connection with EPSON, despite what you might conclude! - I quote the advertisement because of what it says about printer languages):- "....ESC/P 2, the enhanced version of the world standard printer control language ESC/P. With ESC/P 2 you will be able to produce near page-printer output.... ESC/P 2 sets the new standard in the quality of text and graphics on impact printers. The new LQ series incorporates 10 different fonts, two of which are scalable from 8 points to 32 points, allowing you to create headlines, captions and body text in a wide range of sizes and typescripts... ESC/P 2 also allows you to print a new high standard of graphics". Note that ESC/P 2 is a registered trademark of Epson, as is ESC/P).

OK, take it with a pinch of salt, as with all advertisements ≠ it still seems to me an important step forward for dot matrix printers. EPSON have provided me with information on ESC/P 2. It is clear that the primary purpose of this new development is to provide an extension of the EPSON language to enable dot-matrix printers (24-pin or better) to act as page-printers, like laser printers and similar devices: the purpose as far as the new range of EPSON LQ printers is obviously to implement the new language and to provide the user with these capabilities.

Since the new commands, 11 in number, are effectively added to ESC/P, at first sight it looks like a further enhancement, i.e. it might be regarded as "ESC/P 85". However, the EPSON documentation contains the following important statement:- "The first implementation of ESC/P 2 (mid-1991) is compatible with ESC/P (with the exception of three commands), but future versions will not be".

Now whilst this statement is obviously an important one in the abstract, I doubt whether it will affect other than those I would call 'power users' - ie those with 50MHz

486 PCs with 8Mb RAM and a 50Mb hard disc - for quite a while yet. I have no doubt that there will continue to be printers, from EPSON and other manufacturers, that support ESC/P for many years to come.

For completeness, I have produced another table, which sets out the new commands (and no, I do not fully understand what some of them do!), the three old commands that are not supported on EPSON ESC/P 2 printers and also a set of 'non-recommended commands' which are annotated "EPSON recommends against using the following commands because they are no longer supported." (I am not clear the precise difference between these and the previous set - perhaps this last set will be fading out gradually). Perhaps the most obvious of the 'non-recommended' commands are the familiar ESC K,L,Y and Z which were the original graphics dump commands before the 'master' command 'ESC *' came along with the FX80.

KEY FEATURES OF ESC/P 2 (AUGUST 1991)

NEW COMMANDS

ESC (C	Set page length in defined unit
ESC (G	Select graphics mode
ESC (U	Define unit
ESC (V	Set absolute vertical print position
ESC (↑	Print data as characters
ESC (c	Set page format
ESC (t	Assign character table
ESC (v	Set relative vertical print position
ESC .	Print raster graphics
ESC X	Select font by pitch and point
ESC c	Set horizontal motion index (HMI)

DELETED COMMANDS

ESC b	Set vertical tabs in VFU channels
ESC /	Select vertical tab channel
ESC a	Select justification

NON-RECOMMENDED COMMANDS

BEL	Sound Beeper
BS	Backspace
DC1	Select printer
DC3	Deselect printer
CAN	Cancel data
ESC #	Cancel MSB control

ESC <	Unidirectional printing for one line
ESC =	Set MSB to 0
ESC >	Set MSB to 1
ESC ?	Reassign bit-image mode
ESC A	Select n/60-inch spacing
ESC K	Select 8-dot, single-density, bit-image mode
ESC L	Select 8-dot, double-density, bit-image mode
ESC Y	Select 8-dot, double-speed, d/d, bit-image mode
ESC Z	Select quadruple-density bit-image mode
DEL	Delete last character
ESC SO	Select double width printing (one line)
ESC SI	Select condensed printing

ENVOI

I should point out that in no way do I claim to be an expert in this area. At best, I may have obtained a bit more information that other people on some of the details of printer control languages. If I can help anyone with their problems then I am willing to try, but remember that I only have a Brother HR5 (but Christmas is coming!), so I may well not be able to check out the more unusual (to me) bits of ESC/P.

I am sure that the editor will also welcome feed-back on this subject from other readers. Some of the commands - especialy those to do with bit-image graphics are very complex, they need an article all to themselves.

Also, since I am exhausted by my efforts, I propose to leave the parallel activity for IBM-compatible printers to someone else!

Editor's Note:- This article was first published in OUTLET and is reproduced by kind permission of the author.

'It's a start, but we're still a long way off microcomputers'

Z80-SUBS

MACHINE CODE SUBROUTINES.

Edited By:- Bill Nicholls.

Because you are being a little lax at the moment, and not sending me lots and lots of little machine code routines to print, I thought I would do something different this month.

A GUIDED TOUR OF SOME OF THE SPECTRUM ROM'S USEFUL SECRETS

The Spectrum ROM holds many secrets. The most interesting are the routines that can be accessed by the programmer, and in this article I shall outline a few of these routines and explain how they may be used.

PRINTING TO THE SCREEN.

There are a number of ways of transferring information to the screen, something most programs need to do.

At address 2852 (hex 0B24) is the PRINT-ANY character subroutine. On entry the HL register pair holds the pixel address of where the character is to be printed, BC holds the current line and column values and the A register holds the character code to be printed. This method is complicated so it may be easier to use a second method.

This subroutine is contained within another, therefore the whole routine begins at 3432 (hex 0D68 and ends at 3755. At present we are interested in location 3545 (hex 0DD9). This sets the printing locations to the values in BC (B=line and C=column same as in PRINT AT) so to use it:-

i) Load the BC register pair with the appropriate values.

ii) Call 3545 to enter the required values into the system.

iii) Load the character to be printed into the A register and then use RST 16 to print it.

On exit from the RST 16 routine the print positions have been updated ready for the next character.

If you want to print more that one character at a time then the routine PR-STRING at address 8252 (hex 203C) can be used to print any string.

The ROM routine consists of:-

```
PR-STRING  LD A,B
           OR C
           DEC BC
           RET Z
           LD A,(DE)
           INC DE
           RST 16
           JR PR-STRING
```

From this you will see that any string of characters can therefore be printed by:-

i) Loading the start address into DE

ii) Loading the length of the string into BC

iii) Calling the PR-STRING routine.

It is worth noting that control codes (TAB, AT, INK, PAPER, etc) can be sent as part of the string.

CLEARING THE SCREEN

The CL-LINE routine begins at 3652 (hex 0E44) and is very short. The B register needs to hold a value in the range 1 to 24 where 24 would clear the whole screen. Counting starts from the bottom of the screen so a count of 20 in the B register would leave the top 4 lines of the screen unaffected. By many standards this is a slow routine, but it has it's uses.

SOUND

There are two routines in the ROM for producing sound, the BLEEPER and the BEEP.

BLEEPER, at 949 (hex 03B5), requires on entry the HL register pair to hold the pitch and the DE register pair the duration of the sound. The duration value has to be increased as the pitch value is decreased. The pitch for middle C is 1638 and the duration for a second is 260 so for BEEP 1,0 use:-

```
LD DE,260
LD HL,1638
CALL 949
```

The BEEP routine is at 1016 (hex 03F8) and uses the calculator to change the duration and pitch into appropriate values for the DE and HL register pairs. The two parameters need to be on the calculator stack before the routine is called.

USING THE KEYBOARD

Just like printing to the screen, few programs will do much if they don't read data from the keyboard. The main routine is called KEYSCAN which starts at 654 (hex 028E). On leaving the routine, DE is returned with a key value. The zero flag is reset if more than one key is pressed at the same time. The D register indicates which shift keys are being pressed and the E register contains the key number 0-39.

At location 703 (hex 02BF) we find the KEYBOARD subroutine. This handles the repeat facility and decodes the key-value to give the required ASCII character code. If the code is accepted then it is placed in the system variable LAST-K (23560) and bit 8 of FLAGS (23611) is set.

At address 4264 (hex 10A8) there is the KEYBOARD-INPUT routine. This routine copies the value from LAST-K and depends on bit 5 of the FLAGS. It then returns with the carry flag set or reset if the code is printable. This routine is usable but you really need to study the ROM disassemble to

understand what you can do with it.

Another subroutine, at 5588 (hex 15D4) literally has the effect of a PAUSE 0 or 'wait for a keypress'.

LOADING AND SAVING

The whole set of tape handling routines start at 1218 (hex 04C2).

SAVING. This subroutine begins at 1218 (hex 04C2) and passes DE bytes, starting at the address pointed to by IX to the cassette recorder. If the A register holds zero then a header type block (longer lead-in) is saved with the initial marker byte and parity byte added. If A holds 255 then a program block is saved. Provided you know the length and start address of a block when you load it back there is no need to provide the header - lots of games programmers use headerless blocks to help protect their software.

The LOADING subroutine begins at 1366 (hex 0556) and does exactly the reverse of the SAVING routine. It loads DE bytes and the IX register pair points to the first location. When loading the carry flag must be set, but if it is reset then VERIFY can be used.

THE RST COMMANDS

I shall now continue by sheding some light on the RST commands or restart commands. Many books have caused confusion in the past because they have used hex numbers in the RST instructions - without stating they are. So RST 16 has often been seen as RST 10. Wrong! If hex is used it should be shown as RST 10h. Here I will use the decimal RST numbers.

The RST code work in much the same way as CALLs except that you only need a one 1 instruction instead of 3.

RST 0 does a number of things in this order:-

i) Disable the maskable interrupt

GOATS & CARS

By:- Paul Howe.

This is a short routine to simulate the problem of the "Telly Game Show", in which the hapless participant is faced with three doors. Behind two are goats, the other has a car (which would you rather win?).

The program simulates a logic problem invented by Maria Vos Savant, reputed to be the world's cleverest woman.

Here is the listing. Type it in and run it a few times. Once the idea has sunk in change the procedure call in line 40 to read "version2".

```
10 REM Goat/Car probability simulat
   or
20 REM Prog by Paul Howe 1991
30 MODE 3: CSIZE 6,8
40 version1: REM change to version2
   when you get the picture
50 CLS
60 PRINT "BECAUSE I CONTINUALLY CHA
   NGED MY CHOICE I WON ";INT (cars
   won/goes*100);"% OF THE TIME"
70 PRINT ''HAD I CONTINUALLY STICKE
   D TO MY FIRST CHOICE I WOULD HAV
   E WON ";INT ((goes-carswon)/goes
   *100);"% OF THE TIME"
80 DEF PROC version1
90 REM (ILLUSTRATIVE VERSION)
100 LET goat=0,car=1,goes=0,carswon=
    0
110 DO
120 CLS : PRINT "GOES SO FAR: ";goes
    ,"CARS WON SO FAR: ";carswon,"CA
    RS WON IF FIRST CHOICE KEPT: ";g
    oes-carswon''
130 DIM door(3): LET door(RND(2)+1)=
    car
140 FOR n=1 TO 3: PRINT "DOOR ";n;"
    = ";
150 IF door(n)=car THEN PRINT "CAR",
    : ELSE PRINT "GOAT",
160 NEXT n
170 LET firstchoice=RND(2)+1: PRINT
    '''I CHOOSE DOOR ";firstchoice
180 DO : LET revealed=RND(2)+1: LOOP
```

```
    UNTIL door(revealed)<>car AND r
    evealed<>firstchoice
190 PRINT ''QUIZ SHOW HOST OPENS DOO
    R ";revealed
200 FOR n=1 TO 3: IF n<>firstchoice
    AND n<>revealed THEN LET left=n
210 NEXT n
220 PRINT '"I CHANGE MY MIND AND GOF
    OR DOOR ";left
230 IF door(left)=car THEN PRINT ''I
    'VE WON": LET carswon=carswon+1:
    ELSE PRINT ''I WOULD HAVE WON I
    F I'D KEPT MY FIRST CHOICE''
240 LET goes=goes+1
250 PRINT AT 21,0;"PRESS E TO END OR
    ANY OTHER KEY TO GOAGAIN"
260 GET k$
270 LOOP UNTIL k$="e" OR k$="E"
280 END PROC
290 DEF PROC version2
300 LET goat=0,car=1,goes=0,carswon=
    0,numofgoes=1000
310 DO
320 PRINT AT 0,0;numofgoes-goes,
330 DIM door(3): LET door(RND(2)+1)=
    car,firstchoice=RND(2)+1
340 DO : LET revealed=RND(2)+1: LOOP
    UNTIL door(revealed)<>car AND r
    evealed<>firstchoice
350 FOR n=1 TO 3: IF n<>firstchoice
    AND n<>revealed THEN LET left=n
360 NEXT n
370 IF door(left)=car THEN LET carsw
    on=carswon+1
380 LET goes=goes+1
390 LOOP UNTIL goes=numofgoes
400 END PROC
```

What the simulation shows is that by changing his mind, the entrant actually manages to do very well - he gets a lot more cars than goats. If he had, er, sticked to his first choice, then he would have not done very well at all.

Interesting? I'm sure there are other (perhaps even more interesting) variants you could come up with yourself.

MACHINE CODE
WITHOUT THE TEARS

Part 11.

By:- Carol Brooksbank.

We are going to begin now to look at movement. Little graphics figures which move are known as sprites, and our tractor and trailer from last month will be the sprite we move.

There are two forms of movement. The first is simple movement, when the sprite just progresses across the screen from A to B.

The more natural form of movement, when legs and arms move, wheels turn and so on as the sprite progresses, is called animation. We shall come to that later, but we will begin today with simple movement.

Our tractor and trailer occupies 6*3 colour cells. If we want to move it, it needs somewhere to go, so we give it another column of blank cells at the front, so it is now on a frame of 7*3 cells. For really smooth movement we need to move it across this frame one pixel at a time. Fig. 1 position 1 shows you the starting position, with the trailer touching the left edge of the frame. In position 2, it has moved 3 pixel steps. There are now bits of it in all 7 columns. In position 3 it has almost moved right across its frame. Two more steps will bring the front of the tractor to the right edge, leaving the first column empty.

Throughout the process, we use the top left byte of the frame as a reference point for the tractor's position. So, let us say that it starts in column 0, and its frame occupies columns 0-6. When the tractor reaches the right of its frame, and column 0 is empty, if we then move our reference point across 1 cell, so that its frame is now columns 1-7, the tractor's position relative to its frame is exactly the same as it was in position 1. We can repeat the process and move it on.

The tractor itself is 6 cells wide, and there are 32 columns of cells, so if we move it 26 times it will move right across the screen and stop with its nose against the right edge, and that is what we will do with our first program.

But straightaway we come up against a problem. In Fig. 1 - position 3, you can see that in the centre square of the frame, there are bits of trailer and bits of tractor. If our tractor is red, the trailer black and the background green we are in trouble, because only two colours per cell are possible. If we keep the two parts of the sprite in different colours, we shall have to have either the bits of tractor in that cell red, or the bits of trailer red. It is called an attributes clash. Either way, it is going to look very peculiar if bits of the sprite keep changing to the wrong colour as it moves. So, we shall have a black tractor too.

The scrolling is done one line of bytes across the tractor at a time, starting at the top. We shall have to move the tractor from the right, scrolling byte 6 to the right, then stepping back to scroll byte 5 and putting its bit 0 into byte 6 as its bit 7 and so on. If we started on the left, scrolling byte 0 first, its bit 0 would overwrite bit 7 of byte 1 before we had scrolled byte 1, and our sprite would be corrupted. Before we scroll byte 6, we shall have to keep a note of its bit 0, because at the end we shall need it for bit 7 of byte 0.

So it is clear that if we are to be stepping backwards from byte to byte, we need some way of finding the previous byte, in the way that our library routine NXCELL finds the next byte across for us. So it is time for a new library routine, LASTCELL.

Fig. 1. A Moving Tractor.
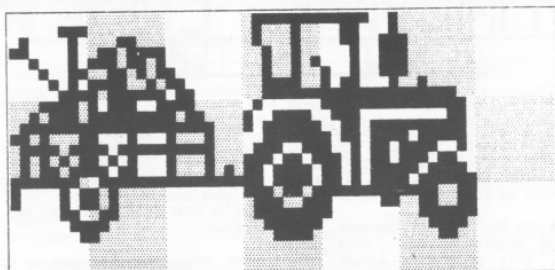
1) **STARTING POSITION**

2) **3 PIXELS TO RIGHT**

3) **6 PIXELS TO RIGHT**

```
LASTCELL  RL  H
          RL  H
          RL  H
          DEC HL
          RR  H
          RR  H
          RR  H
          RET
```
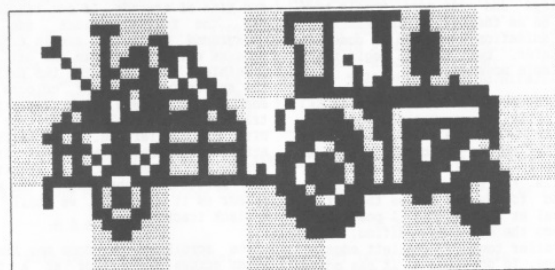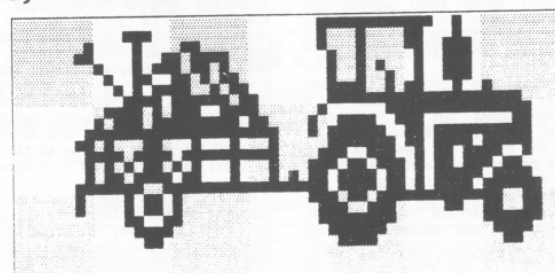
If you compare it with NXCELL, you will see that it is the exact opposite. Perhaps you would like to try writing your own library routine NXUP, to do the opposite of NXDOWN? I'll tell you next month what it should be, so that you can check it.

We shall be using several subroutines in this tractor moving program, and although they are not strictly library routines, in the sense that you will need to use them over and over again in all sorts of different programs, we shall write them one at a time as stand alone subroutines, which will be called by the program main loop or by other subroutines. This gives our program a good, logical structure and ensures that we do not write out the same sequence over and over again, when we could use a subroutine and call it.

```
ROWRT     PUSH HL
          LD B,8
PIXLOOP   PUSH HL
          PUSH BC
```

The first subroutine, ROWRT, moves one row of colour cells one pixel step to the right. It is called with HL holding the top left byte of the row being scrolled. PUSH HL saves that address so that we can exit with HL still holding it.

There are 8 lines of bytes in a row of cells, so B is loaded with the counter. At the start of the loop which does the eight lines we store the address of the first byte of the line we are doing and the present state of the counter

```
          LD B,6
FINDEND   CALL NXCELL
          DJNZ FINDEND
```

We have 7 cells in our frame, so 7 bytes in a line. HL holds byte 1 of the line at present, so calling NCXELL 6 times will point HL to the last byte of the line.

```
          BIT 0,(HL)
          JR Z,BIT0
          LD A,1
          EX AF,AF'
          JR STARTMVE
BIT0      XOR A
          EX AF,AF'
```

First, we must preserve bit 0 of this byte, because we need it later for bit 7 of the first byte. The instruction BIT, followed by a bit number, can be used on any single register, on (HL), and on one or two specialist registers you have not met yet. It tests the bit of the byte held in the register, and sets the zero flag if the bit is 0, resets it if the bit is 1. In this case it tests bit 0 of the screen file byte held in the address pointed to by HL.

We have two lines of action, depending whether the bit is 0 or 1. We are going to load the A register with 0 or 1, whichever the bit held. But we can't afford to tie up our most useful register as a bit status store. Fortunately, there is a solution.

I said in part 1 of the series that the registers are like desk-top pigeonholes. Well, this desk has secret compartments. Behind the pigeonholes is a duplicate set, out of sight unless you press a button which turns them round. When you do that, the pigeon holes you were using, with all their contents, disappear round the back out of sight, and the formerly hidden ones become the ones in use. The button is the instruction EX (EXchange).

The hidden pigeonholes are called the alternative registers, are treated as register pairs, and are written as AF'. So, EX AF,AF' means exchange AF with the alternative register pair AF. But the out of use register is always the alternative one. You don't have to start writing PUSH AF' - in fact there

is no such instruction. As soon as you swap them, the old alternative register becomes AF. Your original A register and the byte it holds are safe and untouched as AF' until you exchange them again to get it back. But remember, when exchanging AF, that you have also exchanged the flags. If the current state of the flags will still be needed you must PUSH AF before EX AF,AF', and POP AF after. If you are retrieving a number salted away in AF', and also need to preserve the flags, you would have to use something like PUSH AF - EX AF'AF - LD C,A - POP AF - LD A,C to keep everything as you want it.

You cannot exchange any of the other register pairs individually. The instruction EXX exchanges BC,BC' DE,DE' and HL,HL'. It is all or nothing with those three, so you have to use PUSH and POP to keep any values you still need on this side of the desk before using EXX.

There is another new way of doing things in this section of the routine. Notice the use of XOR A to put zero in the A register. You would probably have expected LD A,0. You will recall that XOR compares the byte in A with the one in another register, and sets the bits in A which were set either in A or in the other, but not in both. If you compare A with itself, there are no bits set in one but not in the other, so all the bits in A are reset, leaving A holding zero. XOR only needs 1 byte of object code, and works almost twice as fast as LD, which needs 2 bytes.

The logical instructions, AND OR and XOR all reset the carry flag, so you must be careful how and when you use them if you are testing the carry flag. Most people use AND A as a way of resetting the carry flag, because it does so without changing the number in A. There is an instruction SCF - Set the Carry Flag - but no opposite one for resetting it. Unless you use AND A, if you want to reset it without upsetting any registers, you have to use SCF and then CCF (Complement the Carry Flag, reversing its status).

```
STARTMVE LD B,6
MVELOOP  PUSH HL
         RR (HL)
         CALL LASTCELL
         LD A,(HL)
         EX DE,HL
```

Here is the start of the loop which rotates one line of bytes one pixel to the right. We do 6 at first - the left hand one needs separate treatment because its bit 7 will be the bit status we have stored in AF'. We save the address of the byte we are doing, and use RR (HL) to move the pixels 1 step to the right. We have already preserved bit 0. LASTCELL finds the address of the byte to the left. We save the value and then use EX DE,HL. This one byte instruction, only available between these two registers or in swaps involving some of the specialist registers, swaps the contents. DE now holds the address of the left-hand byte for the time being and we can use HL without losing it.

```
         POP HL
         BIT 0,A
         JR Z,ITS0
         SET 7,(HL)
         JR DONEIT
ITS0     RES 7,(HL)
```

POP HL puts the right hand byte of the pair back in HL. We test bit 0 of the left hand byte, and set or reset bit 7 of the right hand byte, depending on bit 0's status. SET and RES (RESet) operate on individual bits of bytes in the same range of registers that BIT can use.

```
DONEIT   EX DE,HL
         DJNZ MVELOOP
```

EX DE,HL now puts the lefthand byte into HL as the current one, and we loop back so long as we are not at the first byte in the line.

```
         RR (HL)
         EX AF,AF'
         AND A
         JR Z,NOUGHT
         SET 7,(HL)
         JR LINEDONE
NOUGHT   RES 7,(HL)
```

The first byte of the line is now rotated, and then EX AF'AF recovers the pixel status of the original last bit in the line which must now become the first bit of the scrolled line - bit 7 of byte 1. AND A can be used instead of CP here because A will only hold 0 or 1. AND, OR and XOR set the zero flag in accordance with the byte in A at the end of the operation. AND A will not change the byte in A, so the zero flag will only be set if A is zero. CP 0 would have done the same, but in 2 bytes and much more slowly.

```
LINEDONE POP BC
         POP HL
         INC H
         DJNZ PIXLOOP
         POP HL
         RET
```

When the line is completed, we recover the line counter and the address of the start of the line. INC H points HL to the next line down and we loop back if there are any more to do. If the whole row is complete we retrieve the address of the top byte of the row and exit.

That's all we have time for now, so we shall continue with sprite moving program next month. We shall write some more subroutines and then modify the program which drew the tractor and add all these new subroutines to make a whole new program, without our needing to retype all the tractor and trailer data bytes and the routines which drew it.

You may need to re-read this instalment because you have met a number of important new commands. I hope too that you look back at the earlier parts of the series from time to time, to remind yourself of exactly how the ones you have met before work. You must be really at home with commands which manipulate the bits in a byte, now that we are working in graphics. RR, RL, RR C, RL C, AND, OR and XOR are especially important in graphics work, when the individual bits of a byte are what matter and the decimal number as a whole tells us nothing much. See you again.

ii) Clears the A register

iii) Loads the DE register pair with 65535.

iv) Jumps forward to 4555 (hex 11CB) where the initialization routines are. This is the equivalent of pressing the reset button or switching on the machine - unlike NEW which leaves things above RAMTOP alone.

RST 8 is the Spectrum's error handling routine. There are two outcomes of this:-

i) The stack will be cleared.

ii) The appropriate report is given at the bottom of the screen.

RST 16. This is the PRINT a character routine. The character in the A register is printed to the current stream.

RST 24 and RST 32 are used by the ROM to fetch the characters from the current line of Basic.

RST 40 jumps forward to location 13307 (hex 335B) which is the first address of the Floating Point Calculator.

RST 48 is used by the ROM to make BC spaces available in the work space or in the Basic area for a line to go in.

Right, that is all for this article. There is much more in the ROM but I will leave it to other readers to write about a few more. The Complete Spectrum ROM Disassembly by Dr Ian Logan and Dr Frank O'Hara is a must for anyone delving into the Spectrum's ROM. It, like so many other books is no longer in print, but you quite often see second hand copies floating around.

Ok, see you all next month, keep your little machine code items coming in - I love to hear from you.

O.D.A.T.W.S.L.A.D.

YOUR LETTERS

Dear Editor,

I have converted all my Spectrum adventures over to SAM disc, using the "Messenger". But up to now, I have had no luck in saving adventure positions from SAM to tape (and re-loading them!). Very annoying playing an adventure and having to start right from the beginning every time I got killed off!!

But now, I have found a way of saving tape based Spectrum games on SAM disc.

1. Insert blank formatted disc into drive.
2. Press break button on SAM, (far right at back).
3. On Messenger Menu screen, press save.
4. Give name to "saved" position (8 letter max).

TO RE-LOAD SAVED POSITION

1. Boot up main program disc. (DO NOT LOAD).
2. On Messenger screen, replace games disc, with "Saved" disc, and load.
3. Type in number of saved position you want to load.
4. Back on Messenger screen, press "I" to run your saved adventure from where you left it off!

No more tape hassle! (Seems to work on Messenger converted games only!)

Yours sincerely, E.J.Petit.

Dear Editor,

Further to your comments in last months issue regarding P.C.G.

I ordered some discs from P.C.G. and when they arrived, inside was a slip of paper saying they had changed their name to Polytype and you can contact them weekday mornings on 0229 832807.

Yours sincerely, Steve Berry.

Thank you Steve, especially for leaving the message on our answerphone - you will not believe the number of people who just hang up, are they afraid of modern technology I wonder!

But back to P.C.G., yours was the first of quite a few telephone calls and letters. Several pointed out this sudden change of name (their second since the DTP system was launched. But several were also from others suffering poor support and service.

We will continue to monitor the situation with P.C.G./Polytype, if anyone has any input to make please drop me a line. Ed.

Dear Editor,

Re Thought Spot, FORMAT Vol 5 No 10. Line 80 of my program published on page 22 should read GOTO 40, not GOTO 20 as printed. If that was my fault, I'm sorry!

Yours sincerely, G.Jackson.

Dear Editor,

In a note to overseas readers you introduced a new service to us, the local currency exchange to pounds scheme. This is a nice gesture, however, considering the cost of registered post still a bit pricey.

But, have you ever considered opening a Postal Giro Account? In most countries foreign transfers from account to account are free, or just a minor fee is charged. No exchange rate lists, no registered post, no exchange on your hand, surely this must be easier (and quicker) for both sides?

I look forward to reading more of your excellent magazine.

P.S. Did you get my article about compilers/interpreters and their speeds (Benchmark - test)?

Yours sincerely, Frode Tennebo.

In the UK the Giro Bank charges

(rather heavy) for running a bank account that they consider handles business transactions. Most banks here are the same but Giro also charge an extra £1 for incoming overseas transactions. Some time ago we worked out the figures and this showed a nett cost of over £3.50 to handle the average overseas payments.

Add to this the fact that we MUST have the payment in the SAME envelope as the order/renewal to have any hope of keeping things straight, and you will see that the Giro system is just a non starter.

Perhaps, with the European Open Market just round the corner our banks will be forced to come into the 20th century - but I wouldn't hold my breath waiting if I were you.

However, given the success of the scheme, I would like to use this opportunity to say we can now afford to drop the premium on overseas currency payments from it's current £3 to just £1. With the growing number of overseas members using the service we are finding it cheaper than originally estimated to convert. I hope this more than makes up for the slight extra cost of registered mail (which only a few of you seem to use anyway).

And yes, I remember your article did arrive (although I can't quite lay my hands on it just at the moment - it is 5.30am so my eyes are not working yet.) I will be in touch if there are any problems.

Dear Editor,

In response to your request for software and books to re-distribute to other SAM and Spectrum owners, I am sending you these three utilities. I hope they will go to users who will use them well. I also hope you will raise a lot of money for a deserving charity.

You say that you will be selling them at shows and printing lists in FORMAT. I (and I'm sure many others) can't make it to such shows and so would prefer lists. If this takes up too much room in FORMAT why don't you distribute lists separate from the magazine, on request. I think this would be much fairer for those who are looking for a book on a popular or

unusual subject which may sell out quickly at a show.

I personally would be interested in any beginners machine code books and books on hardware design. However, I fear that these books would be in high demand, given the response to Carol Brooksbank's excellent machine code course and the SAMCo Newsdisc hardware tutorial. Surely this emphasizes the need to give everyone a fair chance to find what they want.

Yours sincerely, David Finch.

Point noted David, however with only one or two of any book on offer I'm not sure we could cope with handling the complexity of running lists in FORMAT and then needing to return most cheques because the items are gone.

Still, I have had one idea, just came to me in a flash (oh, my brain hurts now) as I was writing these replies, and I would welcome readers comments. How would it be if we ran a list in one issue, detailing what was available (with condition etc) and then allowed sealed bids to be submitted for opening on a set date. Do you think that might work? I await readers comments or other ideas.

In the meantime, thanks for the programs. Can I remind readers that we are accepting books, software and hardware which will be sold later this year with the proceeds going to charity. Ed.

Dear Editor,

In the October '91 copy of FORMAT, (page 5) is a mention that MICRONET was closing down.

I could be wrong, but I imagine that an awful lot of people have VTX 5000 modems. These are more or less useless, as far as I am aware, for use with COMPUSERVE, so one does not encounter the changes problem.

You mention, paragraph 3, the use of Bulletin Boards. Having never used anything other than Micronet, I now have some hardware which does not appear to have an application.

Could you list some telephone numbers of B/Bs, (not Bed and Breakfasts), and/or other uses of the VTX 5000, apart from the usual dust

collector or door stop.

I notice the VTX 5000 is still advertised in FORMAT, so I presume it still has a use.

Yours sincerely, M.Jones.

It so happens that Brian Gaff is working on just such an article at the moment. Keep reading FORMAT - it should appear soon. Ed.

Dear Editor,

I have just been leafing through some past issues of FORMAT and have re-read, with interest, some examples of your perennial battle with members of INDUG who omit or forget to submit their membership number when corresponding with you. I realize I am running a great risk in re-opening an old wound during a quiescent period (!) but I would like to be so bold as to constructively offer a partial solution.

Your reluctance to change from a system which by and large serves you well is entirely understandable - but it must also be accepted that the system's inability to undertake a reasonably fast search for a name is a fairly serious shortcoming. In the circumstances, could I diffidently suggest that that you invest in a copy of the Spectrum counterpart of Kobrahsoft's Samdice utility, which has an excellent disc search facility? With it you could search a disc for a given name and obtain the relevant track/sector in about one and a half minutes; surely not an excessive amount of time given (as I recall) that you hold your complete database in just three discs.

Yours sincerely, Albert F.Olivera.

My program does use sector reading and a machine code search, the disc + track + sector numbers are related to the membership number so, at most, it takes the time needed to change a disc to find the right record. However, if there is no membership number quoted the system has to start at record one and read each record (sector) and test it against the search key. Records are processed at approximately 200 records

per minute - so you will see that with over 3500 records it can take some time to find someone.

I will admit I could speed things up by reading a track at a time with machine code but that would then over complicate the program when it comes to do its real job - making sure everyone gets there FORMAT each month. There is also no excuse if you forget your number as it appears each month on the label we use to send you your FORMAT so all people need to do is copy it down. Ed.

* - * - * - * - *

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

This is YOUR letters page so it is up to you, our readers, to fill it. Send your letters, 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.

38

39