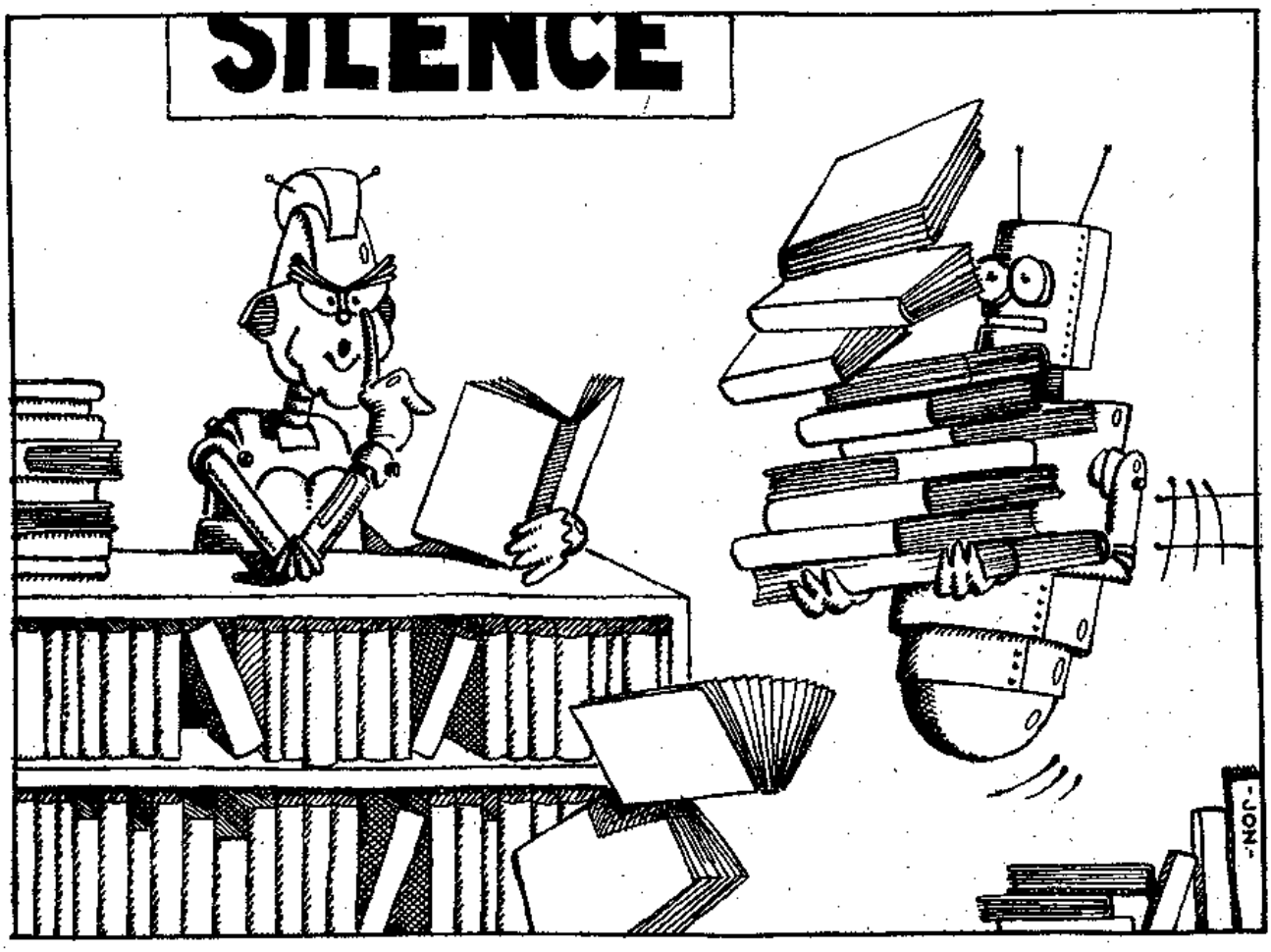
Vol 2 - No 11. July 1989.

FORMAT

THE MONTHLY MAGAZINE FOR
SPECTRUM, DISCiPLE & PLUS D USERS



LIBRARIANS
Do It With A Spectrum

Vol 2 No11. CONTENTS July 1989.

THIS MONTHS ADVERTISERS:-

FORMAT is published by **INDUG.** 34 Bourton Road, Gloucester, GL4 0LE, England. Telephone 0452-412572. **DISCIPLE** and **PLUS D** are trade marks of MILES GORDON TECHNOLOGY plc. Lakeside Technology Park, Phoenix Way, Swansea, South Wales, SA7 9EH. Telephone 0792-791100.

Printed by D.S.LITHO. Gloucester. Tel:- (0452) 23198.

Since the last issue of FORMAT went out I've managed to sneak in a short holiday - I really needed it. I went to Valkanburg, in the Limburg region of the Netherlands (thats Holland to some of you). All I can say about Valkanburg is "I wish I could have stayed forever", you couldn't wish for a nicer place or nicer people. But isn't it a small world, I travel all the way to the South East corner of the Netherlands, on a coach trip with only 21 members (including my wife, two kids and me) only to find the wife of an INDUG member in the party. And before people start jumping to the wrong conclusion let me state I didn't even realise who she was until my daughter swopped addresses on the coach coming back to the UK. Hi Mary, I hope your parrot liked the cage...

Well it was straight back to the grind stone on my return, it took me three hours just to open the mail that had arrived while I was away. But I did get the time to watch the HORIZON program on BBC2 (Monday 12th June) which covered the life and inventions of Uncle Clive. It was interesting not for what was included but for what was left out. Brief mentions of the ZX80, no mention of the famous ZX81 and only a quick flash of the Spectrum. No mention of the hundreds of companies who grew to support his wonderful machines. And little indication that he was responsible, far more than any other individual, for the birth of home computing for the masses in Britain. Another indication of the bias the BBC seems to have against the Spectrum.

Several people have been asking about the FORMAT SOFTWARE tapes, well I hope to bring back the service soon but I need to find someone in the Gloucester area to handle the orders for me. In the meantime, please do not send in orders as I only have to return them.

Nev Youngs first HELP PAGE can be found in this issue, I trust it will grow into an important feature of each months FORMAT. Please support it, not only with your questions but with your own answers to common problems.

In reply to my request for readers ideas on how to spread the word about FORMAT to other Spectrum users. Well I havent exactly been flooded out with letters yet (but I suppose it is a little early) so I'm still looking for lots of ideas from you.

There has been lots of comments about the new - two column - layout in FORMAT. With just a few exceptions they have all been favorable I'm glad to say. So it looks like its here to stay, even though it takes a little longer to produce. But there are more changes to come in the future including a new FORMAT logo, so keep reading. By the way its the need for a new logo thats holding up the magazine binders I promised a couple of months ago. As soon as the logo is finalised the binders will be available, we cant use the existing logo as it wont reduce or enlarge without problems.

And finally this month, several members have asked about renewals and when are they due. Well we send out a reminder with the issue BEFORE the last of your subscription so you have plenty of time.

Bye for now.

Bob Brenchley. Editor.

NEWS ON 4

TELETEXT ADAPTOR RETURNS.

The TTX2000 Teletext adapter for the Spectrum is available again. The adaptor, formally manufactured by Volex Electronics, has been taken over and re-introduced by Micro Projects Ltd. New units, upgrades, spares and repairs are offered.

The Teletext adapter can be pre-tuned to four channels which are then selected from the keyboard. The required page number is also entered from the Spectrums keyboard. Once the page is selected then the usual functions of HOLD and REVEAL are available. Details of calls to the TTX2000 ROM are given, so users can write their own programs to access pages and analyse data.

The TTX2000 adaptors are currently offered at the very special price of £59.95 incl. From Micro Projects Ltd, Freepost, Alsager, Stoke-on-Trent, ST7 2BR. Telephone 0270-875178

MGT plc.

Miles Gordon Technology have now become a plc company with an issue of shares under the goverments Business Expansion Scheme (BSE). The share issue, fully underwritten by Johnson Fry Ltd, will guarantee the finance required for the full launch of the SAM Coupe this autumn.

Two new directors have been appointed both as advisers to the existing directors and to represent the interests of the shareholders. Andrew Browne IPFA, BSc becomes Non-executive Chairman and Brian Burrows MBA,PhD, BSc joins the company as a non-executive director. Both have wide experiance in the business world. Alan Miles (Marketing), Bruce Gordon (Technical) and Rob Collins (Commercial) are the executive directors of the new board.

MGT's current staff of 17 is set to expand over the next few months and extra 7000 sq. ft. of space will be taken over in a near-by unit. Work on SAM continues to gain speed (see this months SAM SPOT.

LIGHT GUN BUNDLE

In what many people see as Amstrads last ditch effort to retain a share of the Home computer market it has announced bundle deals on both the +2 and +3 computers. Both will now be packaged with a Light Pen and compatible games. The Light Pen is produced by Trojan in South Wales.

TWO NEW PROGRAMS FROM BETTERBYTES.

Dave Hood of BETTERBYTES SOFTWARE has obtained the disc rights to FASTWORD, the Spectrum Thesaurus program from Bruce Publishing. Described by the author as being the equivalent of a 120,000 word printed thesaurus the program even fits into a 48k Spectrum. Two versions are supplied on the one disc. One for manual interrogation and the other which "overlays" Tasword 2. The TW2 version will allow you to work on the text-file and, at the press of a button, consult FASTWORD.

BETTERBYTES has also announced a new database program called FILE MASTER. Written by Robert Caldicott, the program is designed to be easy to use and flexible. Versions for the PLUS D and DISCiPLE will be available next month with versions for Microdrive and, possibly, the Discovery system following soon after. The software includes ultra-fast search and sort routines. Watch for full details next month.

SHORT • SPOT

By: John Wase.

This month, for a change, I'm dealing with hardware. There's bits and pieces here for everyone; discs and disc units, problems associated with various Spectrum models....

Let's start with a general comment. Very few readers seem to use 5.25" discs. I know the drives are a bit dearer, but the discs are so much cheaper. I tend to use these as standard; I use coloured discs as a memory aid and always have stuff filed in duplicate. A large file box (200 discs) keeps mostly text with different subjects on different discs (e.g. personal letters on one pair) but apart from this I tend to put programs or utilities one to a disc. I know that this uses a lot of discs, but 5.25" ones are very cheap (less than 30p), and so are very little dearer than paper. Saves all that hassle of shuffling through a disc to find a program....

And talking of 5.25" discs, Mr A. Casarrubios of London mentions that he has modified his TEAC 80 track 5.25" drive by soldering one leg of a 390R resistor on the board where it is marked R14 (to the left of the stepper motor plug) and connecting the other leg to the other side of R14 via a single pole switch which selects 40 or 80 track. Don't forget to reconfigure the system disc. He would also like to hear from keyboard builders as he is having trouble with single key punctuation marks.

Raymond Hoy of Basingstoke had a troublesome 3.5" disc which would sometimes boot up, but usually gave "Format Data Lost" or "Data Error". By borrowing a friend's unit, he confirmed that the drive was at fault, removed it from its case and tried again: the disc rotated, but the head reader did not move back and forth. Looking at his drive (top left) he could see the motor with a spiral spindle protruding. On the spindle sits a small pin which was jammed. Turning the screw at the back of the motor unjammed it and cleaning and relubricating with WD40 prevented the problem recurring.

Now to the Spectrum +3. T.McKay of London mentions that on some 3" discs, he encounters error codes that cannot be erased with the "FORMAT" command. He gets over this by using a demagnetizer such as is used for cassette heads before reformatting, and finds it convenient to use the bottom of a saucepan as a metal base when using the demagnetizer to sweep over it. I suspect he has found one of the many bugs in the +3: this one detects a strange format and reports it with the error message "unrecognised disk format", but does not allow you to continue with the format. Discs formatted under CPM or on another machine therefore give problems, as I found when I bought a whole lot of discs which had previously had Einstein programs on them. However, help is at hand: the solution is easier than one would think. Take a good disc, with valuable data if you must, stick it in drive A and enter the command "FORMAT "A:". Up comes the error message "Disk is already formatted. A to abandon, other key continue". Remove your precious disc, stuff in the one which is playing up and press a key. Magic...

Next the Spectrum +2a; You know, the thing which masquerades as a +2 with a glued-on cassette player, but which has a +3 circuit board. This has several consequences. Quite apart from

the fact that the edge connector, like the +3, is incompatible with almost all add-ons which would work with the existing 48K Spectrums, the ROM has again been changed. The old +2 had one ROM, the +2a has two, like the +3, only there are again slight differences. This means that some software which would run on the old +2 will no longer run on the +2a. Brian Gaff, of Micronet fame, tells me that if you throw caution (and your guarantee) to the winds, and replace the +2a ROMS with the old +2 ROM, many of the problems will go away. Alas, the edge connector is totally immune to this doctoring, so you still can't get your favourite disc interface to perform as one would like unless you use the "fixer"... Oh, well, can't have everything...

And talking of the old Spectrum +2, I've got one that buzzes like a bee-hive, when connected to a telly. Unscrew and swing back the case (careful with the membrane connectors), reconnect power whilst pressing BREAK, let computer and telly warm up, then tune in the TV for the best picture (Spectrum+2 test card) and finally adjust the screw in the modulator (largish oblong boxy thing) for clearest sound.

The old +2 has lots of hardware faults. One of the quirks of many, but not all is an inability to scan the keys properly in 48K mode. The fault is exacerbated when the computer is cold and in cold weather (perhaps the overall voltage is lower): capitals M and N and symbol shifted A don't work, particularly frustrating for Tasword users who find themselves unable to access the main menu. A partial cure is to solder a 1000 pF capacitor across diode 35.

For those computers with an RGB plug, except the +3, the Ferguson TX television is ideal: it has a composite video input as well as a scart plug. The +3 works reasonably well with the scart plug, the +2, the 128 and the QL directly with a QL lead. Templegate, well known at Microfairs has a supply of ex-demo monitor tellies at £145: I have had two, and the difference in picture quality is amazing, although the 128 and +2 will need separate sound arrangements and you'll need a special interface for the 48K Spectrum (this is extra from Ferguson). The QL lead is free but as Ferguson can take ages it might be a good idea to get a spare locally. Phone Alan Folwell on 0895-82 2067.

Finally, I end on a regretfully sorrowful note. I started by acknowledging everything carefully as it came through the post. This is now proving impossible in terms of time and prohibitively expensive in terms of postage - the other day I spent a whole day and pounds at the post office and was still overwhelmed with more unanswered correspondence. For 3.5" discs I have returned a brand new disc; for 5.25" I haven't bothered.

I'm sorry, but in future, I cannot acknowledge all contributions, although I'll still try and answer individual queries. I will of course still try to return new discs for old; perhaps you could let me know if you particularly want your old disc back: an address label or s.a.e. would be appreciated.

See you next month then, and please keep the contributions coming. Send your items direct to me at:-

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

- - - - -

Spectrums in the LIBRARY.

By: Carol Brooksbank.

To avoid wasting space I have shortened certain words and phrases used frequently in this article, these can be found in the glossary given below.

GLOSSARY

MF+3	MASTERFILE for Spectrum +3
MF09	MASTERFILE 48K version 09
TW+3	TASWORD Plus 3
+3	Spectrum Plus 3
+2	Spectrum Plus 2
D/+D	DISCiPLE and/or PLUS D disc systems
READER	Church of England Lay Minister/Preacher

-*-*-*-

I find it astonishing that so many people, including, I suspect, Alan Sugar, dismiss the Spectrum as a games machine. This article (and one next month) describe its involvement in the running of a small library, but it could equally well apply to the running of a club or a small business.

The Spectrum, with its range of good, inexpensive business and graphics software, can add tremendously to a small business's efficiency, and it is a very modest capital outlay.

Coventry Diocesan Readers' Library is a small library of (at the time of writing) just over 800 theological books, intended primarily for the use of Readers-in-Training and their Tutors, but also used by established Readers and occasionally by clergy. The library is open once a month, and between openings, users choose the books they require from the catalogue and these are left for them at collection points.

The catalogue is a vital tool in the running of any library, and is particularly important for us because the opening hours are limited. Printed copies are supplied to the Tutors, and are available for purchase to any other user, so the card file type of catalogue is not suitable. We have to have something which is easily printed, and this is where the Spectrum comes in. The catalogue file is kept on MF+3. This is far and away the best Spectrum filing program around, but unfortunately it is still not perfect. The main drawback is that the whole file is kept in memory, which effectively restricts you to around 40K, because the sorting options cannot handle a larger file if you wish to sort by the maximum 25 characters.

I am pretty sure that the combination of limited disc space and the very slow disc operation on the +3 made Campbell's opt for this type of file handling, but it does make for inconvenience with large files. Our library catalogue is now split into 8 separate files. Even on the +3, if the file were on the disc and only the selected records brought into memory, two files would suffice. With the D/+D, the whole catalogue could be in one file, with far more expansion space in the file than we shall ever have on the bookshelves! So, if John Campbell is listening, perhaps when the SAM comes on the market you will bring out a real disc/random access Spectrum version of your otherwise first-class program.

When I make the original catalogue entries for new books, I actually use MF09, and then use the conversion program and import them into MF+3. This may sound rather eccentric, but

there is a good reason - MF09 allows you to make duplicate copies of a record. MF+3 does not.

Many books contain material on more than one subject. Take one entitled 'Baptism, Eucharist and Ministry'. You can only keep it in one place on the shelves, but the catalogue must have entries under all three subjects, so that anyone searching for information on the Eucharist can discover that there is something in a book shelved with those on Baptism.

So there will have to be three entries for that book, almost identical except for the class numbers and the code letter which shows whether it is a main or supplementary entry. I can save myself an awful lot of typing by using MF09, making one entry, duplicating it the required number of times and making the minor alterations. The conversion and importing to MF+3 takes only seconds.

Printing the catalogue, which has author, title and classified lists, and a subject index, is made rather complicated by its division into eight files. It is kept in author order, and each file holds one block of the alphabet. This is fine when you want to print the author catalogue, but not so good for printing the title, classified and subject lists - a book with a title beginning with "A" may be written by an author whose name begins with "Z". So there is a lot of exporting of records to new files involved in compiling the various lists.

MF+3's print format design is very versatile and easy to use, so once the files have been compiled, the printing is no problem.

However, we use an option to print to disc instead of printer, which saves the file in a form which can be loaded into TW+3. Printing from TW+3 lets you insert more printer control codes than are available from MF+3, and there is an added bonus. The printing files must be preserved for future use until the catalogue is revised and TW+3 files take up less disc space than MF+3 files. With the +3's low capacity and expensive discs this is an important consideration.

The catalogue is kept up to date by an annually revised supplement which lists all the books added to stock since the current edition of the catalogue was printed. When the supplement grows so large that the annual reprint is uneconomical, we shall incorporate everything into a new edition of the catalogue.

The records of books on loan are kept in the catalogue entries, so that if someone phones me at home to ask whether we have a book in stock, I have only to load one file to know whether we have it, and if so, whether it is on loan.

The illustrations show the file main entry and the printed catalogue entries. (Fig. 1)

MF+3 is also used for the file of members, the list of books noted for future purchase following good reviews, and the books on order. MF+3's ability to display a running total of a number field - in this case the book prices - helps, when I am selecting books for purchase, to make sure I do not go over budget.

MF+3 files can be exported in a form which is acceptable to the TW+3 mailmerge option. We make use of this when sending out reminders about overdue books, lists of new books and so on. The interaction between MF+3 and TW+3 makes them a particularly useful business suite.

There is one trap I often find difficult to avoid. It is all too easy to think that every operation must be done on the computer, but there are some things which can be done more efficiently in other ways. I have a small petty cash float, and it would make no sense to load a computer program to alter a record every time I buy some paper or post a letter. It is far more efficient to keep a record in my filofax. Mind you, I use the

Spectrum to print the filofax account pages. (Fig. 2) My article in FORMAT Vol 2 No 8 gives you the program to print filofax pages if you want to explore this further.

In next months follow-up article I shall be telling you about ways of using the Spectrum to print all of the Library's stationery, using the +2, PLUS D, and a number of graphics packages.

So don't miss next months issue.

LIBRARY PETTY CASH

MONTH		COST		CRTS SOLD		CASH LEFT	
DATE	ITEM	£	P	£	P	£	P

Fig 2.

Author :BURNABY, John
Title :Belief of Christendom: a commentary on the Nicene Creed, The
Publisher :SPCK
Date :1959
Price :6.50
Class numb:238.142
Main entry:RT5
Accession :00070
For course:B
Subject in:NICENE CREED
Date issue:03 Dec 88
Original i:---

1) The full MF+3 record for a book.

BURNABY, John
Belief of Christendom: a commentary on the Nicene Creed, The

SPCK 1959 6.50 00070 CLASS NO. 238.142

2) The printed catalogue: Author entry.

Belief of Christendom: a commentary on the Nicene Creed, The

BURNABY, John

3) The printed catalogue: Title entry.

M 238.142 BURNABY, John
Belief of Christendom: a commentary on the Nicene Creed, The

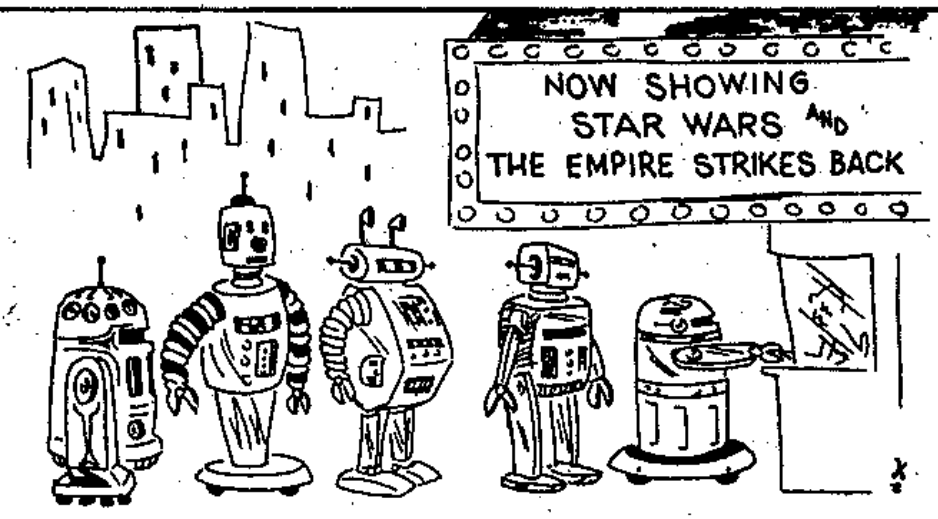
SPCK 1959 6.50 00070

2) The printed catalogue: Classified entry.

NICENE CREED 238.142

2) The printed catalogue: Subject index entry.

Fig 1.



Listing Standards

When printing listings in FORMAT we have adopted a set of standards based on the EASY READ program in Vol.1 No.10. This gives listings with control codes that can be understood. When it comes across the embedded control (or selected other characters) it prints is as a '{', followed by the keys to press, and then a '}' to finish off. The commands work like this:-

E = Extended Mode
G = Graphic Mode
S = Caps Shift
SS = Symbol Shift

All followed by a key to be pressed.

So {ES 4} means enter Extended Mode then press Caps Shift and 4, this gives the control code for green ink. {G A} tells you to enter graphic mode then press A, that is the first of the UDGs. {INV VID} and {TRUE VID} are also given. The copyright is shown as {ESS P}.

We also ofset the Basic line number which helps to make the lines easier to read.

If you are writing for FORMAT you dont need to worry about these, we do the listing for you. However there are a few things to keep in mind.

Please use UPPER-CASE letters for Basic variables that are only one character long (i.e. A not a) many lower-case letters cause problems. Dont use the letter 'O' as this can look like zero.

Also, many people will find understanding the program a lot easier if you use long variable names like TOTAL, LIVES etc.

Listings in FORMAT are taken direct from a working copy of the program. If you type carefuly things should work.

INSTR$

By: Ken Elston.

Over the years I have used most of the home computers - from the good old ZX81 through an Oric (very short lived) to a CPC 6128 by way of a BBC'B' and a Commadore PLUS 4. OK, I know, I'm a glutton for punishment, but I write commercial software you see.

Now from the very start, way back in 1982, the Spectrum has been a firm favourite with me. I use it for all Z80 development work and could not face life without one. The string handling in Spectrum basic is second to none. Other machines force you to use LEFT$, RIGHT$ and MID$ to do jobs which the Spectrum copes with so much easier with its string slicing (TO) feature.

However there is one command thats missing. Most new basics have and INSTR$ command. Whats that? you say in unison. Well lets take the string:-

"FORMAT IS A GOOD READ"

Now if you needed to find out if the string contained the word "GOOD" the best way in normal Spectrum basic is to use the following routine:-

```
10 LET a$="FORMAT IS A GOOD READ"
20 FOR I=1 TO LEN A$-4
30 IF A$(I)="G" THEN IF A$(I+1)="O"
   THEN IF A$(I+2)="O" THEN IF A$(I+
   3)="D" THEN PRINT "Found at ";I:
   STOP
40 NEXT I
50 PRINT "Not found..."
```

This is the fastest search method for a specified word. But its not very flexible and its not very fast if the string you are searching is quite long. If you INPUT the string from the keyboard and wanted to search for several words this could be a very long-winded process.

On many basics now have the command:-

LET N= INSTR$(A$,B$)

which searches B$ for A$ and sets N according to where it finds a match.

Recently I needed an INSTR$ routine to use in a small management game I was writing so I set out to give my beloved speccy its new command.

First problem, how to communicate with the machine code routine. I dont like POKEing parameters into a routine, its slow and I dont like the look of the result. So I dredged up an old article of Toni Bakers (anyone know where she is these days?) which explained how to use the DEF FN command to communicate between basic and machine code. To explain, lets look at the line DEF FN X(A+B). In Spectrum memory this has extra bytes stored after both A and B. These bytes are used as stores for the numbers the function is working on. So if in the program a line reads LET RESULT=FN X(5,12) the five bytes after A contain the number 5 in floating point form (see chapter 24 of the 48k manual) while 12 is stored after the B.

If we added a USR call at the end it provides a way of passing complex parramiters to machine code. The line:-

DEF FN X(A,B)=USR 40000

will call a routine at location 40000 with the system variable DEFADD (23563) pointing to the start of the

DEF FN function. We can then extract the numbers we want to work on, calculate a result and pass it back via the BC register. When used with strings the bytes are give a pointer to the string in the basic variables area.

Right, thats the idea behind INSTR$, now to get things working. Type in this program, RUN it, and allow it to save the CODE file (you can use tape, microdrive or any disc system).

SET-UP PROGRAM.

```
 10 CLEAR 65399
 20 LET mc=65400: LET CS=0
 30 FOR I=65400 TO 65491
 40 READ N: POKE I,N: LET CS=CS+N
 50 NEXT I
 60 IF CS<>10992 THEN  PRINT "CHECKSU
    M ERROR": STOP
 70 SAVE d1"INSTR$"CODE 65400,92: VER
    IFY d*"INSTR$"CODE 65400
 80 PRINT "CODE SAVED OK": STOP
 90 DATA 221,229,221,42,11,92,221,94,
    6,221,86,7,123,178,40,70
100 DATA 221,110,15,221,102,16,167,23
    7,82,56,59,68,77,197,221
110 DATA 94,4,221,86,5,221,110,13,221
    ,102,14,229,213,197,221,78
120 DATA 6,221,70,7,26,190,32,19,35,1
    9,11,120,177,32,245,193
130 DATA 209,225,225,237,66,68,77,221
    ,225,3,201,193,209,225,120
140 DATA 177,40,4,11,35,24,213,193,22
    1,225,1,0,0,201
```

Now type in this little demo program, it shows several different uses of the INSTR$ command which is called by:-

```
DEF FN I(A$,B$)=USR MC
```

You can load the code in at any location as the machine code is completely relocatable. In the function A$ is the string you are searching for, B$ is the string the routine will look through. The value returned by the FN call it a pointer to the start of your search word (A$) within B$ or zero if no match is found. Remember that outside of the DEF FN you use any strings you like, the A$ and B$ are treated as LOCAL variables.

DEMO PROGRAM.

```
 10 CLEAR 59999: LOAD D*"INSTR$"CODE
    60000
 20 LET MC=60000
 30 DEF FN I(A$,B$)=USR MC
 40 REM
100 REM Test 1.
110 LET A$="This is test number 1."
120 LET B$="test"
130 LET A=FN I(B$,A$)
140 PRINT "1. "; INVERSE 1;B$; INVERS
    E 0;" Starts at character "; INVE
    RSE 1;A; INVERSE 0'"within the st
    ring "; INVERSE 1;A$
200 REM Test 2.
210 DIM A$(10000)
220 LET A$(9900 TO )="This is a large
     array"
230 LET S$="array"
240 INPUT "PRESS enter to search an a
    rray  of 10000 characters.";I$
250 PRINT '"2. The word "; INVERSE 1
    ;"array"; INVERSE 0;" starts at 1
    ocation "; INVERSE 1;FN I(S$,A$);
     INVERSE 0'"I think you will agre
    e that its very FAST"
260 DIM A$(1)
300 REM Test 3.
310 POKE 23658,8
320 INPUT "Type a line of text includ
    ing'"the word - FORMAT - please.
    ";I$
330 IF NOT FN I("FORMAT",I$) THEN  GO
    TO 310
340 PRINT '"3. "; INVERSE 1;"FORMAT"
    ; INVERSE 0;" started at "; FLASH
     1;FN I("FORMAT",I$)
400 REM Test 4.
410 LET A=FN I("N","ABCDEFGHIJKLMNOPQ
    RSTUVWXYZ")
420 PRINT '"4. 'N' is the "; INVERSE
     1;a; INVERSE 0;" character of"'
    INVERSE 1;"ABCDEFGHIJKLMNOPQRSTUV
    WXYZ"
```

INSTR$ will work with 48k or 128k Spectrums and will not interfere with any other extended basics as it does not rely on interrupts or redirecting ERR-SP to call it.

I hope you will find many uses for this routine, its proved very useful for me in several programs already.

INTRODUCTION TO

* FORTH *

Part 3.

By: Alan Cocks.

In the first two articles about Forth I introduced the stack and also the method of putting together a routine and compiling it.

The two routines:-

```
: FETCHTEST      and    : STORETEST
  PAD                     327
  @                       PAD
  .                       !
;                       ;
```

were used as examples. You will have seen that STORETEST contains the number 327. The code will run faster if a "constant" is used instead of the number directly.

327 CONSTANT AGE <ENTER> will compile the number 327 into a constant called "AGE" and whenever AGE is required, 327 will be returned to the stack ready for use. Let us compile a routine which uses the constant AGE. It is important to note that routines (including constants) must first be compiled before they themselves can be included into other routines. This has an effect on the programmer. It encourages clear thinking through well structured program design. It also gives the programmer "at a stroke" the facility to make use of an early routine in more than one routine later on, and makes this facility quite visible.

After defining the constant "AGE" as already shown, a new routine could be compiled which uses AGE.

```
: STORETEST1
  AGE
  PAD
  !
;
<ENTER>
```

The sequence:-

STORETEST1 FETCHTEST <ENTER>

will result in '327' appearing on the screen.

Variables, like constants and routines in general, also have to be compiled before they can be included in others. **VARIABLE WEEK <ENTER>** compiles a variable called "WEEK". When "WEEK" is used in a routine it leaves on the stack its address. "Very interesting, I am sure", I hear you say "what good is that?" Well, the sequence **42 WEEK ! <ENTER>** will store the value 42 at the address owned by WEEK. And the sequence **WEEK @ <ENTER>** will cause the value at the storage address to be copied to the stack. You may notice that Forth lets you really see what is going on, and as a programmer, gives you a lot of power. Users of "FIG Forth" will find that when originally defining a variable, an initial value is required on the stack first.

Now there is something that must be said, and perhaps it is a good a time as ever. When a routine is compiled by a colon definition like, for example, STORETEST, regular users of Forth call this a "word", not a "routine" as I have done. Compiled code of the routines is automatically put into a memory area known as the "dictionary" which is where the programs "run".

Because Forth "words" are self supporting and are not called up by a central control program, and because the higher level "words" are a naturally structured hierarchy of lower level "words", then Forth programs may be made easily readable with little effort.

For example a substantial program which allows checking of a personal bank account over the period of a year, may have source text at the highest level something like:-

```
: ACCOUNT (...)
  INITIALISE
  BEGIN
  MENUSCREEN
  EXIT?
  WHILE
  OPTIONS
  REPEAT
  FINISH
;
```

This program would be run simply by typing **ACCOUNT <ENTER>**. Of course the "word" OPTIONS is likely to be supported by many low level "words", perhaps right down to code.

One of the things that I really like about Forth is the way that discs are used. Forth includes a rudimentary but effective operating system where, in effect, disc sectors are written to directly and Forth disc buffers are used in a simple automatic sequence.

This has the effect of increasing greatly the apparent capacity of the computer and can transform the use of the smaller system.

The size of disc buffers and grouping of disc sectors will depend upon your particular Forth implementation, although for the Spectrum it is likely to be 512 bytes.

"Words" such as DISCWRITE and DISCREAD which follow are simple examples of the use of discs for holding data.

```
: DISCWRITE       (n... ) (number is
                  required on stack
                  initially to identify
                  block)
  EMPTY-BUFFERS   (n...n)  (stack
                  unchanged)
  BLOCK           (n...a) (number on
                  stack used, block of
                  disc sectors read and
                  put into buffer,
                  address of buffer
                  start put on stack)
  65              (a...a 65) (example
                  data put on stack)
  SWAP            (a 65...65 a) (swaps
                  the two top stack
                  positions, address is
                  top again)
  !               (65 a... ) (at
                  address, data is
                  stored)
  UPDATE          ( ... ) (mark current
                  buffer as updated)
  SAVE-BUFFERS    ( ... ) (all updated
                  buffers to disc)
;
```

In this example, the address which has been used within the buffer is the buffer starting address. Other locations in the buffer may easily also be used, the third location, for example, by adding 2 to the address. It might be useful to make the comment here that single precision numbers are held as 16 bits, that is, two 8 bit bytes for the Spectrum.

The routine DISCWRITE has been written so that it will expect to find a number already on the stack- the number of the required block. For example, if Block 300 is a valid block in your configuration, then '300 DISCWRITE' will work.

```
: DISCREAD  (n...) (block identity
            number)
  BLOCK     (n...a) (block brought to
            buffer, address put on
            stack)
  @         (a...65) (data to stack)
  .         (65... ) (data on screen)
;
```

If the block identified had contained source text, for example colon definitions of words, then in order to compile these into the dictionary, all that is required is **n LOAD <ENTER>** where n is the ID number of the block.

You have now seen a little of Forth. Its power lies in the visibility it gives to the programmer, its almost primitive simplicity, and the way it allows you to use successive routines to accelerate towards your completed program.

YOUR LETTERS

STAR*LETTER* *STAR*LETTER

Dear Editor,

Where are the radio amateurs in your magazine? Apart from the excellent write up from C.Grant Dixon, in Vol2 No1, there has been nothing about the use of the Spectrum in amateur radio. Don't you know the Spectrum is the most popular computer in amateur radio circles?

Come on FORMAT, give us coverage.

Yours Sincerely, Alec Torman.

I know, I know. This is just one of many letters on the same subject, but I can only print articles that are sent in, I know little about radios myself. So I have been looking for someone to write an article or a regular column for FORMAT, so far though I've had no luck.

If anyone out there would like to send in an article, or perhaps a review of software aimed at radio amateurs, I will be only to please to consider it for publication. Ed.

Dear Editor,

I have a +2, PLUS D and PC disc drive and I operate on G+DOS 3. I suffer frequently from keyboard lock -out, usually at the most exasperating times. Imagine my joy when I saw your program (PLUS D PATCH) last month which seemed to solve the problem.

However I cant get it to work, after getting the all clear from the code I set the DI with RANDOMIZE USR 25056 but nothing happened. Is there a bug?

Incidentally I find that more often than not when I attempt a snaphot-4, all appears to go well until the return to program and then the keyboard locks out and SECTOR ERROR comes up. I have never yet been able to do a successful 128k snap.

Going back a little, I found I got nowhere with the TAPE-DISC program of the 1a system mod. In all cases I checked and re-checked the listing and could find no errors. Could someone please help?

Yours Sincerely, A.L.Vernon.

You do seem to have problems with your set-up. Lets take things one step at a time. First, the PLUS D PATCH last month will only work with G+DOS 2a. Thats the version which 95% of readers with PLUS Ds use, and is the only one we can really support. On the subject of your version of G+DOS, only versions 1, 2 & 2a have ever come out of MGT. Version 3 was the result of a patch we published in Vol 1 No 8. This update was superceded by the updates in Vol 1 No 11, thats the ones that create 1a or 2a depending on which version of the ROM you have. If you try to update v3 you might have problems, so go back to your original system tape and turn v1 into v1a. The same thing may apply to the TAPE to DISC routine, try with just your original system and see what happens.

I cant see why your snapshot 128k is not working but it could be due to using a stepping rate that is too fast. Try 12 or 24ms. Ed.

Dear Editor,

I've had a Spectrum since they first came out. I bought Microdrives when they came along, then a Beta disc interface which never worked right for me. Now I have a second hand Disciple which is wonderful, its fast, reliable and gives me more space than I think I will ever need. What I really need now is a powered mother board for my Spectrum, I have so many add-ons that its a constant round of plugging and un-plugging. Is there anyone out there who knows where I can find one?

Yours Sincerely, C.R.Biddle.

Letters printed may be edited for length or clarity. The writer of each months STAR LETTER wins an EXTRA 6 months subscription to FORMAT.

BUSINESS AND PLEASURE

The MGT Lifetime Drive. A Citizen 1Mbyte 3½" drive with its own power supply. Works with your DISCiPLE or PLUS D. Also works with your Q.L., Amiga, ST, BBC or even PC! All you change is the cable and switches. A lifetime of business and pleasure!

Lifetime Drive.....**£129.95** inc VAT
Spectrum/BBC Cable.....**£10.00** inc VAT
All other cables (each).....**£18.50** inc VAT
Shipping charge.....**£4.50**

MGT
MILES GORDON TECHNOLOGY

Miles Gordon Technology plc

Lakeside, Phoenix Way,
Swansea Enterprise Park,
Swansea, SA7 9EH ☎ 0792 791100

THE SAM SPOT

By: Bob Brenchley.

Work on the SAM Coupe computer continues to gain pase as the autumn launch gets nearer.

At the same time interest in SAM is so great that MGT phone line are at times clogged with calls asking for details. So MGT have installed a special **SAM COUPE HOTLINE** available 24 hours a day on Swansea (0792) 791275. This will give a recorded progress report for the 'Master of Technology' himself, Bruce Gordon, whose picture adorns this page. WOW! the great man speaks to you just for the cost of a cheap rate phone call. Don't miss his weekly update.

A visit to Swansea just last month allowed me to spend a few hours playing with SAM Basic, FAST? you wont believe it until you see it yourself. The writing of SAM Basic is already very advanced and is now going through a period of debugging and polishing. Anyone familier with Spectrum basic and, even more so, Dr Andy Wright's BETA Basic, will feel right at home with SAM Basic.

The draft version of SAM's Technical Development Manual has now gone out to a handful of important people (like me) and the full Programmers Developement Manual should be available in the near future. Remember, if you are a programmer and you want to work on SAM projects, write to Alan Miles at MGT giving details of the things you have done in the past and your ideas for SAM programs.

ADVENTURE CORNER

By: Paul Rigby.

How many devoted adventurers out there get the shivers when I say the word "Lenslok"? Makes you want to reach for the pills doesn't it? But were Level 9 really to blame for such a device? The ugly head of piracy has been centred by the leading software houses as the bringer of bankruptcy, the scourge of the bank-balance and a good excuse to raise software prices. But what of the fight against this force of evil? How have the adventure moguls fought with the foe? Well they have certainly put their thinking caps on when it comes down to the subject of software protection. However, there are, noticeably, two schools of thought when it comes to protection. Either protect it to the eyeballs or don't bother.

It must be said that the former faction, that is, those who advocate software protection, do so with gusto. Some methods, however, have provided more headaches than the original problem of piracy! The Lenslock was one such device. For those fortunate to have missed this plastic monstrosity a word of explanation is in order, I think. The Lenslok was a small plastic device measuring about four inches by one and a half inches. It was constructed of opaque plastic with a small, prismatic transparant window in the centre of it. The user was prompted, either at the start of the adventure or at a random point during the game, by the appearance of a black screen with the display of a small band of abstract art which appeared in the centre. The user would then bend the outer sides of the Lenslok and place the device with the lens outermost, onto the screen. With a little juggling around a code could be read from the screen. The trouble was, the amount of juggling that the poor gamer had, in the majority of cases, to do would have qualified them for the Moscow State Circus. Lensloks have probably caused more eye problems than any other man-made device - and people wonder why I have a permanent squint?

Level 9 probably produced this device in an effort to prevent, as much as possible, dabbling with the actual code of the game. The intention was certainly a sound one. After all how often have you heard C64 owners complaining about the excruciating noises emanating from their drives through protection routines? However, Level 9 had to contend, not only with adventurers with eye problems, but broken Lensloks too. They were easily stepped on - a fact which cheered up many an adventurer in a fit of frustrating eye strain. Which leads nicely onto those drive noises. Because software protection appeared to be the answer to many a problem. It was transparent to the user for one thing. Transparent, maybe, but certainly audible. As those long-suffering C64 owners found to their cost when their drive heads were knocked out of alignment by the constant protection knocking sounds. A fault which has not affected us Spectrum owners, thankfully. However, software protection can produce a tolerable system. I find the old "find word three on page 23, paragraph one" routine quite painless. In addition, I have heard no complaints of this system. The only problems that may occur are the possibilities of losing the novella or other documentation.

Probably the most effective way of preventing piracy is to design the adventure around a comprehensive set of documentation. If the player has to

refer to a mass of maps, notes and booklets then the incentive to copy the program and photocopy all of the documentation decreases. Realisation soon dawns that it just is not worth it. Infocom are a case in point in this matter. There is such an assortment of well-produced goodies in each adventure "pack" that you would feel a bit deflated if you had not bought the game and were left with the bare copy of the disc plus a mound of photocopied documentation. Some software packages actually produce so much printed matter that it actually works out dearer to photocopy than to buy in the first place! On the other end of the scale the home-produced market do not really promote anti-piracy methods because of the needless cost. Is it worth researching a software protection program for a game which costs 1.99? Attempts have been made to slave the user to documentation and this has, on occasion, been a worthwhile move.

Another device which has raised a chuckle in the race against piracy, is the dongle. That piece of plastic stuck in a specified port of the computer. Again, though, this small device is prone to falling on the carpet which promptly eats it. Result? A legally bought copy of an unplayable game. Probably the most famous dongle of them all was the "Ram Pack" issued with Mikrogen's adventure-ish game "Shadow Of The Unicorn". The dongle actually upgraded the 48K Spectrum to 64K, which was needed, apparently, to play the game. Mikrogen almost sank without trace because of the enormous amount of money the "dongle" consumed. The game itself was pretty ordinary anyway. The strangest dongle has to go to Mastertronic and their adventure "ZZZzzz". This is a case of a dongle not really being a dongle at all. The game is actually unfinishable without the joystick being fitted to the joystick port. Only trouble is - Mastertronic didn't know about it! A bug or an undocumented feature?

A growing trend amongst the software developers is to rely on the use of the translation wheel. This strange addition to the anti-piracy league's armoury consists of two concentric pieces of card bonded together but still allowing rotation. Normally, the player has to match a word, number or symbol on the outside of the outer circle with a similar shape on the outside of the inner circle. A message or number can normally be read from the inner area of the circle which forms the password. This is just a long winded way of doing the "Find word four on Page..." routine. It is fairly easy to lose and comparatively fragile with the constant rotating of the cardboard circles causing wear and tear upon the device.

However, it you think that the present system is varied and imaginative then you should have seen the ones that got away! Amongst the ideas which almost made it, or did so for a limited period, includes the idea of actually damaging a floppy disc in a "controlled environment". This involved the action of placing magnets on the disc in certain known areas which did not do the disc any good at all. The damage was not great enough to make the game unplayable, in fact the game was supposed to work perfectly, the crunch came when anyone attempted to back-up the disc. An error would then occur which would highlight a sector error. This idea seemed to fade away into the night. Another idea took the form of self-destructing discs! Actually, this was a sort of Trojan Horse program that is being seen by many owners of Commodore Amigas in a Virus form. This particular Trojan was carefully placed upon the disc during the copying of the game. The protection program would be triggered when anyone attempted to copy/back-up the game. The Trojan would then wipe the disc completely! Imagine merrily copying your disc - for purely legal back-up reasons, of course - and then checking if everything has gone okay only to find that the disc is empty!

Funny thing piracy. Software houses moan about the state of it and how the end-user is to blame. Yet have you heard the stories of how the copying

of such programs normally emanates from the software house itself? It may be unscrupulous emloyee who takes a new program to swap amongst his friends for the sake of his or her ego. It may be that same person who takes the game to a rival company for examination. The rival would certainly, after the examination had been made, not stand in the way of the game being pirated on a grand scale. Why not? The more the game is pirated the less revenue that will reach the owner. Then again, suggestions have been made that the originating software house may "leak" the game before it has been officially presented to the market to gain public opinion with a view to late changes being made if any unwanted "feature" appears, in addition to which a bit of publicity will have been gained. The last theory may hold some weight because have you noticed that when a game is known to have been pirated the software house in question makes a bold statement saying that it does not matter because the pirated version is full of bugs and the retail version will have extra features? This always seems to happen near the release date and the pirated version is **always** bugged! The ugly head of Piracy or the ugly head of Marketing?

Goodbye until next month.

* - * - * - *

THE SECRETS OF WORD MANAGER

SPECTRUM MACHINE CODE MADE EASY

Part 3.

By: Francis Miles.

ARITHMETIC. - Part 1.

There are many different ways to crack an egg, or so they say. And this is very true in programming. This month I start to look at ways to handle numbers in machine code using examples from "Word Manager".

The Spectrum ROM includes a "floating point calculator", but "Word Manager" doesn't use it at all. Such calculations as are needed are relatively simple, and it seemed both quicker and probably more economical of memory to make up purpose-built routines for them. (Besides, I didn't understand how to work the FP calculator when I was writing most of the program! I do now, but I still think "Word Manager" is better off without it.)

A typical example is the arithmetic relating to digital inputs and outputs - inputs of things like "number of copies to print" or "line length", outputs of the same variables when they are to be displayed in a menu. In "Word Manager" these are always positive integers - no fractions or minus numbers - and never more than five decimal digits.

An excerpt from the subroutine DIGINP, for displaying numbers as they are input digit by digit, was given in the note "Printing messages". The whole subroutine looks like this (the arithmetic is all in lines 4260-4320, but the rest of it may be interesting):-

```
3910 DIGINP EQU $
3920 ;Input numbers digit by digit.
3930 ;Returns with the number in HL.
3940        LD HL,0
3950        PUSH HL ;acc
```

[HL is the "accumulator", in which the number input is built up digit by digit. I make it a practice in writing assembly language to "keep the score" of what's on the machine stack in the right-hand margin; it's very easy to crash a program by mistakes in this.]

```
3960        LD B,3
3970        CALL 3652 ;CL.LINE
```

[3652 CL.LINE clears lines at the bottom of the screen, as many lines as indicated by the B register. Here it is used to clear away any prompt remaining at the bottom of the screen from previous displays.]

```
3980        CALL P.BY1
3990        DEFB 22,0,0
4000        DEFM 'Input X to '
4010        DEFM 'cancel input'
4020        DEFB 13,95 ;13 newline,
                       ;95 underline.
4030        DEFB 8+128 ; backspace
```

[The operation of the P.BY subroutines was explained in my last two articles "Printing messages". Here lines 4020-30 put the "underline cursor" in place and ready to be overprinted by the next input.]

```
4040 DG.INP CALL INP
```

[DG.INP marks the start of a loop (returned to from line 4450): input a digit, print it in the input area, add it to the accumulator, and come back for more. The INP subroutine is described in my forthcoming article on "Interrupts". Whatever key you press, it puts its character code in the A register.]

```
4050 ;Return on ENTER.
4060        CP 13 4070 POP HL ; -
4080        RET Z
```

[Notice that the stack must be cleared before RET - in this case we want the value in HL anyway, but if I'd left HL on the stack RET would jump to the address in HL, which might be anywhere, and would crash the program.]

```
4090 ;Cancel input on 'X'.
4100        CP 'x'
4110        JR Z,DIGINP
4120        CP 'X'
4130        JR Z,DIGINP
```

[A jump back to the start of the subroutine clears the prompt out of the bottom of the screen - that's why lines 3960-70 are there.]

```
4140        PUSH HL ;acc
```

[This PUSH must come after the JRs to DIGINP, because the stack must be clear on entering the subroutine again; but before the JR to DG.INP, because that expects to find "acc" on the stack.]

```
4150 ;Read the input. If not a digit
4151 ;then reinput.
4160        CALL 11547;NUMERIC
4170        JR C,DG.INP
```

[NUMERIC is a handy ROM routine, which returns with NC if Ais the character code of a digit 0-9, and with C otherwise. There are similar subroutines 11405 ALPHA and 11400 ALPHANUM, which work the other way round, giving C (not NC) if A is a letter of the alphabet (ALPHA) or - either a letter or a digit - (ALPHANUM).]

```
4180        SUB A;48
4190        LD C,A
4200        LD B,0
```

[Now we have copied the digit into BC, ready to be added to the accumulator. NB that the character code of the digit is 48 more than the value of the digit itself.]

```
4210 ;Print the digit.
4220        PUSH BC ;dig/acc
4230        CALL A1C
4240        POP BC
4250        PUSH BC ;dig/acc
```

[NB that opening a channel messes up all the registers; anything you want to keep must be put on the stack while you open the channel. Not all ROM routines do this, but many of them do.]

```
4260        LD A,C
4270        ADD A,48
4280        RST 10H
4290        LD A,95
4300        RST 10H
4310        LD A,8
4320        RST 10H
```

[Lines 4260-4320 are explained in "Printing messages".]

```
4330 ;Add the digit to 10 times
4331 ;the accumulator.
4340        POP BC
4350        POP HL ;-
4360        ADD HL,HL
4370        PUSH HL       ;2 x acc
4380        ADD HL,HL     ;4 x
4390        ADD HL,HL     ;8 x
4400        POP DE ;-
4410        ADD HL,DE     ;10 x
4420        ADD HL,BC
```

[This is the arithmetic. What I am doing is first multiplying the accumulator by 10 (lines 4360-4410) and then adding the new digit to it (line 4420); if you think of the effect of inputting say 7 on to 432 to make it 4327, you will understand this. There is no Z80 command for "multiply", except ADD HL,HL (and ADD A,A) which doubles the value of the register. If X is the old accumulator value, then line 4360 makes it 2X, and line 4370 puts 2X on the stack. Then lines 4380-90 produce successively 4X and 8X; lines 4400-10 add 8X to 2X giving 10X. Simple!]

```
4430 ;Get the next digit.
4440        PUSH HL ;acc
4450        JR DG.INP
```

[NB that this subroutine doesn't end with a RET; the only exit from it is in line 4080. This isn't at all unusual.]

A whole class of arithmetic operations may be summed up under the heading of "one-byte arithmetic"; these are calculations in which all the numbers are whole numbers from zero to 255, and thus very fast and economical coding can be performed using the "rotate" commands and the one-byte ADD, SUB, SBC, etc commands.

Word Manager in its latest form has three system variables, called P.S, SCRUN and C13, holding parameters which relate the length of the print line LINEP with that of the screen line LINES. The screen line is always either 64 or 32 characters, ie these are the line lengths of the only two screen display formats; the print line, for printing out texts on a printer, can be anything up to 128 characters and is selected by the user. The screen display fills out the lines with black spaces if the print line isn't an exact multiple of 32 or 64, and if the print line takes up more than one screen line it displays the first line BRIGHT and any others dull.

On start-up, and whenever the print or screen line lengths are changed, the program recalculates the following parameters:

P.S (one byte, one to 4), the number of screen line;

SCRUN (one byte, 32, 64, 96 or 128): the number of bytes shown on screen, including black spaces, for each print line.

C13 (two bytes, upto 12x64 = 768): the total number of characters (NOT counting black spaces) shown in the first twelve lines displayed on screen - this is important because the cursor is usually on the thirteenth screen line.

All these are calculated by the following subroutine:

```
3370 S.PERM EQU $
3380 ;calculate permanent screen
3390 ;parameters
3400 ;1. P/S, no of scrn lines per
3410 ;LINEP
3420 ;=1+INT((LINEP-1)/LINES)
```

[Eg if LINES is 64 and LINEP also 64 then result is 1; if LINEP is 65 result is 2, because two screen lines per LINEP will be required, the second one holding only one character and 63 black spaces.]

```
3430        LD HL,(LINEP)
```

[LINEP is only one byte, but the space reserved for it is two bytes, with the hi byte always zero. It is used so often that it is convenient to be able to load it like this into a two-byte register.]

```
3440        PUSH HL ;LP
3450        DEC L
3460        LD A,(LINES)
3470        PUSH AF ;LS/LP
3480 ;quick division
3490        RRCA
3500 SP.DV  SRL L
3510        RRCA
3520        JR NC,SP.DV
```

[The five bytes in lines 3490-3520 give the integer result of dividing L by A; but this particular trick only works if A is 2, 4, 8, 16, 32, 64 or 128, ie A in binary has seven zeroes and only one 1. Once you have learned to "think in binary" - very important for one-byte arithmetic - innumerable similar tricks will suggest themselves. They are particularly often useful in dealing with Spectrum screen calculations.]

```
3530        INC 1 ;P/S
3540        LD (P.S),HL
```

[The hi byte of P.S is immaterial at this stage - it gets filled by another routine later.]

```
3550        LD C,L ;save P.S
3560 ;2. SCRUN, number of bytes shown
3570 ;on screen per LINEP
3580 ; = P.S * LINES
```

[By doing these calculations in the right order, so that the values required are already on the stack or saved in otherwise unused registers,

much faffing about with system variables can be saved; one can waste extrordinary amounts of code if one isn't careful about this. Here we have LINES on the stack and P.S handy in the L register.]

```
3590         POP AF ;A=LINES
3600 ;quick multiplication
3610         RRA
3620 SP.MT   SLA L
3630         RRA
3640         JR NC,SP.MT
```

[The exact counterpart of lines 3490-3520 above; again the trick only works if A has seven binary zeroes.]

```
3650         LD (SCRUN),HL
```

[As with P.S, the hi byte of SCRUN gets filled by another subroutine later.]

.....

[I skip a bit here which sets a flag to indicate whether any black spaces are going to be needed or not.]

```
3770 ;4. C13, serial number of last
3780 ;character on 12th line
3790 ; = (12/P.S)*LINEP
```

[Luckily 12 divides by 1, 2, 3 or 4, the only possible values of P.S; so the top half of the screen always holds an exact number of LINEPs.]

```
3800         LD A,12
3810         LD HL,0
3820         POP DE ;DE=LINEP
3830 SP.C13  ADD HL,DE
3840         SUB C ;C=P.S
3850         JR NZ,SP.C13
```

[Another "quick division" routine. LINEP in DE is added to the accumulator in HL just as many lines as C can be subtracted from A without leaving zero.]

```
3860         LD (C13),HL
3870         RET
```

Thats all there's room for this month, more to come in the next issue.

HACK-ZONE

By: Hugh J. McLenaghan.

This month I want to use my column for a review of the "PLUS D HACKER" utility by Steve Nutting. I have taken a lot of time to go over this program and have tried it with quite a few 48K and 128K games.

When you send for the PLUS D HACKER you now get a disc with the program on it (older versions came on tape for transfer to disc) and also a small manual. Although the manual is quite small it has enough information to let you use it.

The first thing you do is set up a special version of your system file to create your PLUS D HACKER disc. This is now your Hacking disc, it has about 746K free, so you can also snapshot some games to this disc. PLUS D HACKER works from within the special system file, there are several sections of code stored in 'Special' files, these are called from the disc as needed by the main program. As such the Hacker disc must always be in drive 1 (except when loading your game).

Using PLUS D HACKER is quite easy after a few attempts, and I am now using it without looking at the instructions.

To hack or to insert pokes you must first load the system file from the PLUS D HACKER disc, you can do this by typing LOAD p1, or by turning your computer off and on then typing RUN. After doing this you must load your game into the computer, then remove your games disc and re-insert the PLUS D HACKER disc. When the game gets to the point you want to stop it at, press the snapshot button, then press 0. A few seconds elapse (while the screen is saved to disc) and a menu appears.

This menu has quite a few options on it. I will go through these in detail.

RETURN TO GAME 'R'. There are two options to return, one of these is just the straight Return. This will restore the computer to the stage that you pressed the Snapshot button at. Sometimes the computer will crash, but this is because of the PLUS D and the 128K spectrum.

The second way is slightly more complex, but allows you to save parts of the code from BASIC. This way will be discussed in a later section.

DISASSEMBLE 'D'. This is one of the large sections of PLUS D HACKER. This option allows you to disassemble parts of the original code and see how the program works.

The only problem with this section is that you really need to know what everything means before you can understand what is going on. I intend to give a small course in FORMAT to help you with these commands. Using this section you can find the proper POKE addresses to get inf. energy etc. in all of the games.

There are only 3 control keys, so it is an easy section to manipulate. There are as follows:-

ENTER - pauses the display to allow you to read the op-codes and addresses.
SPACE - aborts the present run.
M - Takes you back to the menu.

One thing I found with pausing is that it can sometimes be difficult. At times, after I let the ENTER key go, the listing would keep on going. You must just quickly press the ENTER key

and let it go for it to work.

TEXT LISTING 'T'. This will give you a similar sort of display as a Multiface 1 or 128. The display is very clear and it is also quick and easy to use. The keys are the same as for Disassemble.

GRAPHIC PICTURE SEARCH 'G'. This is another big routine, this time it has a lot more keys but most of them are straight forward to remember. There are a few keys which I think he should have put round the other way, but apart from that they are good.

It takes a lot of time to get used to, as you can change the amount of characters per line. One problem is that the scrolling can be fast and it is possible to miss something, especially small SPRITES. So at the same time that you scroll forward you should also increase and decrease the width to make it easy to see graphics screens, or SPRITES. What I have found is that most of the sprites have been between 2 to 12 characters wide.

This routine works very well because you can also search for the colour of sprites. The only thing that is missing from this routine is a facility to save the sprites that you have found onto disc. To do this at the moment requires you to write down addresses and widths etc., then return to BASIC and try to search them out with BASIC which is not very easy.

SEARCH STRING 'S'. This is a very easy routine to use, but again the keys are funny, e.g. SPACE acts as a delete key. Apart from how to enter text, it is very fast and works very well.

INFINITE LIVES SEARCH 'I'. It will only search for one form of decrementation of lives, which is used quite a lot, but more recent games find other ways of doing the same. The routine is also very fast.

FIND BLOCKS 'F'. This routine is very useful although it has only one purpose that I can see. That is to search out for interrupt tables, these occur in blocks of 257 bytes of the same number. They are used so that a lot of hardware may be used with the computer without wrong jumps to machine-code. It is hard to understand, but is important. Again the routine is fast and effective.

BLOCK FILL 'B'. This allows you to fill any area of memory with bytes of the same number. It is not very useful as far as I can see, but it is quick.

ENTER POKES 'P'. This option brings up a small block of numbers and variables. Using this option you can to a variety of things.

One of these is to return to BASIC to allow you to save parts of memory to disc. You need to do two things, one of these is to POKE 23730,23999 which will set the clear address. You then set the JP address to 4535 which will NEW the computer. Now you may save and examine the memory from BASIC.

To enter POKES all you have to do is move the highlighted box such that it is large and in the centre. Now you type the address followed by the number you wish poked. On the 128K machine you must first type the page number. This is the most important routine as it allows MULTIFACE pokes to be entered.

You can also change the machine variables or the machine stack. This must NOT be done unless you know what you are doing, as you can cause the computer to crash.

Well that's all for now. I am going to be using the PLUS D HACKER from now on in the HACK ZONE, so you better all buy it as it is well worth the money.

I would rate the utility 9/10.

Remember, if you have any contributions for HACK ZONE send them to:- Hugh McLenaghan, 36 Floorsburn Crescent, Johnstone, Renfrewshire, Scotland, PA5 8PF.

See you next month.

128K SCREEN

By: Shimon Young.

The 128 machines have a facility that has largely been overlooked by programmers. On 48k machines there was only one place in memory where the display file could be kept, but the 128s actually give you a choice of two places to store the picture that is going out to the TV screen.

Isn't one enough? Yes, but having two has distinct advantages. The most obvious one is for animation. You may remember the flickery sprites that dominated the games in the old days. Full throttle was infested with riders that couldn't quite work out if they were there or not. This was because with every frame of animation the screen had to be cleared, the racetrack had to be drawn and finally the bikers had to be added. The problem is that a television displays a new picture every 1/50th of a second and the speccy has only had time to draw an empty racetrack, so the bikes don't appear until the next frame.

A later programming technique was to work out exactly what the finished picture would look like, elsewhere in the memory, and when it is complete, move it byte by byte down into the actual display file starting at address 16384. This cured the flicker but ate up processing time. This is why some 3D games only really used the top two-thirds of the screen (eg. Starion) as it takes less time to move 4k than the full 6.

Enter the 128 Spectrum. You can now draw a screen in one of the two display files while the TV is showing the other one. For example, the TV could be showing a still picture of a Cobra spacecraft from Elite about to bite the dust, while the computer is drawing the exploding ship in the second, unshown screen. Once it has finished that frame, instead of moving all that data to the display file, it only has to start showing the second screen. Only one command has to be executed instead of six thousand one hundred and forty four!

The computer then clears the first screen and starts drawing the next frame of the explosion. The result is faster, flicker-free animation with no time wasted shifting data to and fro.

Now the technical stuff. The second screen is in the first 6912 bytes of bank 7 of the 128's memory. The 128 has 8 banks of 16k. Two are loaded all the time but the programmer has to chose which one goes into the top 16k. You can only use 48k at a time still but which 48k is up to you. The analogy used in the +2 manual of a TV set is very useful. Although you can watch only one channel at a time, the others are still there waiting to be selected.

To get access to these banks and the second screen you have to output to port 32765. The value to be output is arranged as such:-

BITs 0-2	Determines which bank is paged in at 49152 (0-7)
BIT 3	Determines which screen is shown. 0 for 1st. 1 for 2nd.
BIT 4	Determines which ROM is paged in. Usually 1.
BIT 5	48 lock. Do not set!

Got that? Well just in case its not too clear here are a few examples given in the form of a table. If you output the value in column one you get

the results as described in column two.

Value Output	Result
16	Normal screen, Bank 0 at 49152, ROM 1 selected
23 (16+7)	Normal screen, Bank 7 at 49152, ROM 1 selected
24 (16+8)	Second screen, Bank 0 at 49152, ROM 1 selected
31 (16+8+7)	Second screen, Bank 7 at 49152, ROM 1 selected

This port can only be written and not read so it is useful to keep a copy of what is sent out in memory. The BASIC uses system variable BANKM at address 23388. When poked from BASIC, it is automatically sent to port 32765. It is important that Bit 4 is set, ie. never poke a value less than 16 or your BASIC will crash!

To load a SCREEN$ into bank 7 just type POKE 23388,23: LOAD "<name>" CODE 49152. You can then display it by POKE 23388,24. You dont need Bank 7 to be paged in to view the second screen. Note that when the program stops and returns to the BASIC editor, it automatically reselects the normal screen.

Unfortunately, from BASIC you cannot draw or print to the second screen. You can still do these to the normal screen whilst the second is showing.

As I mentioned earlier, IN 32765 would just return a meaningless value. This is why when you make a 128 snapshot with the DISCiPLE or PLUS D, you're asked to press Y or N. The interface selects the normal screen then asks the user if the display changes. If Y is pressed it knows that the second screen was being used when the button was pressed. There is actually an obscure port (10495 I think, but I'm far from certain) that could have been used to test which screen was showing, but it would have been fiddly to program so it's much easier to ask the user instead.

The interface uses a nifty trick to determine what bank of memory is paged in. It saves the first few bytes at 49152 into shadow RAM and replaces them with a few of its own. It then selects the banks in turn until it finds those bytes again. It then knows what bank was paged in and returns the bytes that were there originally. The system could be fooled by putting the bytes the interface is looking for at the start of each bank. I won't say what the bytes are but ego has a lot to do with it!

Next month, I will show you a program that makes a less conventional use of the second screen but is very spectacular.

* - * - * - *

BACK ISSUES

For members who have missed past issues of FORMAT (or perhaps worn theirs out through constant use) we run a back-issue service. The cost is £1 per issue (£1.25 overseas) incl p&p. Your copies will be sent out with your next monthly issue of FORMAT (provided we receive your order at least a week before). Make cheques (drawn on UK bank or Euro-Cheques, P.O., cash) payable to **FORMAT.**

AVAILABLE ISSUES

Volume 1

Issues #1 (Aug'87) - #12 (Jul'88).

Volume 2

Issues #1 (Aug'88) - #10 (Jun'89).

Note:- Early issues are high-quality reprints.

Please **WRITE YOUR ORDER ON A SEPARATE PIECE OF PAPER.** and mark envelope BACK-ISSUES. DO NOT include letters with your order as this will cause delays. Remember to quote your membership number or we wont be able to send you your order.

THE HELP PAGE

Problems with your DISCIPLE, PLUS D or Spectrum. Dont worry, let the HELP PAGE sort them out. Note: One question per letter please.

By: Nev Young.

Well hello at last, my first Help Page, what have I let myself in for?

The first interesting problem comes from Stefan Szubert of Oxhey Watford (Is there life south of Watford?) Stefan writes about a menu program published in the introductory issue of FORMAT that despite being typed in correctly give an error when ever it's run. The line in question is 190 LET X=VAL "CODE I$-64": etc. This give the error Nonsense in Basic 190:1. I know there is nothing wrong with the program as I have used it myself.

The clue comes in the letter. Stefan is using a +2. Just like myself, and it is here that the problem lies.

The +2 works in two modes 128 and 48K. If used in 48K mode there would not be a problem but in 128 mode the editor is different. I quite like the full screen editor but if you don't know of the pitfalls you can get some very strange problems.

With the 128 editor the machine does not recognize keywords until you press enter at the end of the line. This means that variables cannot have the same name as a keyword. For example in 48 K mode you could enter LET ink =4: INK ink: where INK is the keyword and ink is a variable. On the 128 the word ink would ALWAYS be accepted as a keyword thus giving an error as you cannot have a variable starting with a keyword. The problem given above is along the same lines. Here the word CODE is in quotes so the 128 has decided to leave this as the 4 letter word c o d e instead of changing it to the keyword CODE. The fix is simple either re-enter the line using the 48K mode or type it as LET X=CODE I$-64: (ie without the quotes).

It is as well to remember these points as there are many good programs around for the Spectrum written before the 128 was invented. Many of these can only be entered in 48K mode although they will run in either mode. This menu program is just one such example.

Next comes a letter from Francis Ball of Leagrave Luton who writes "1 How can I load a snapshot from +D so as to remove any unnecessary space and so get more than 16 to a disc. 2 How can I back up a snapshot to tape. 3 Can I make a backup to tape that will then run" To answer points 2&3 first There is a program called TAPE-SNAP available from Shimon Young (see small ads) that claims to transfer snaps to tape and reload and run them. I've never used or seen this program but I'm assured it does the job. As for point 1 er well er yes well. I would say with great difficulty. It can be done you would have to read the snapshot file as a stream and then use a data compression algorithm to create a new file. Sounds easy if you say it fast enough. With this you would have to expand the new file back before being able to reload it as a snapshot. Another way would be to see just how much of the file is really needed and resave that not as a snapshot but as a code file. The last stack pointer would have to be recovered so as to recover the Z80 registers. Some of which are saved on the stack and others in the file header in the directory. When you take into account the work involved to do this and weigh it up against the cost of a new box of discs then more discs work out cheaper (of course its not as much fun).

Now for a tricky one from Tony Jeenes, a radio ham, of Malvern who

says he is trying to make his Spectrum read morse by inputting an audio signal into the ear socket and wondered why the routine he was using didn't work. The routine he used was a machine code fragment designed to sample the ear input and return a non zero value if there was a signal. But it didn't work. There is a very good reason for this. You seem to be under the impression that bit 6 of port 254 will be a 1 when there is a signal and 0 when not. This is not the case. The ear socket is connected to the ULA via a differentiator made up of a 680 ohm resistor and a 100pF cap. Therefore you can only expect to get a 1 bit when the signal is on the leading half cycle. Therefore instead of testing for a signal to be present for a certain time as you have done you really need to count how many edges there have been in a sample time or work out the time between edges. The spectrum tapeload works by using the latter of these.

Routine for above

```
        DI
        LD B,24
        LD C,24
LOOP    IN A,(254)
        RRA
        AND 32
        JR NZ,NEXT
        DEC C
NEXT    DJNZ LOOP
        LD A,C
        LD (40000),A
        EI
        RET
```

You'll be happy to know there is a routine in the ROM to do this for you. At address 1507 (05E3H) is LD-EDGE-2 and at 1511 (05E7H) is LD-EGDE-1. They are entered with the C register holding the last edge type and border colour, and the B reg holding a timing constant. If you want to find the time to the next edge the use LD-EGDE-1 and the value in the B reg shows how long it took. The carry flag is reset if no edge is found before B overflows to 0. If you call LD-EDGE-2 then the value in the B reg will be the time for a complete cycle (ie from edge to edge). It also plays around with the border colour but I'm sure you won't mind that.

The following routine seems to work fairly well but could do with some refining.

```
        DI
        LD BC,0
        CALL 1511
        LD B,0
        CALL 1511
        EI
        LD C,B
        LD B,0
        RET
```

By the way are you aware that when you return from machine code you can get the last value held in the BC register pair by calling your routine thus "LET x = USR 30000" and x now holds the value in BC. This way you don't need to copy the C reg to A and save it at a known address.

I wouldn't have done it this way anyhow. Think about this. If you tap into the AGC line of your RX then the DC voltage will be proportional to the received signal strength. A bit of amplifying and you could produce a true TTL signal that pulses on and off with the received CW. This can then be fed directly into the spectrum, (As long as you are careful with your ground planes), and you probably wouldn't need to use the BFO.

Oh goodness this is starting to sound like an article for Practical Wireless.

Thats all for now from Bonnie Scotland. Please keep you letters trickling in either to Bob at the FORMAT address or direct to me at HELP PAGE, 3 Mitchell Place, Falkirk, Scotland, FK1 5PJ.

Unfortunately I can only answer letters through the HELP PAGE of FORMAT. And PLEEEEEASE do not enclose anything that you want back, cos you won't get it.

Back next month.

SMALL ADS. SMALL ADS.

BELGIUM MEMBER wishes to contact other Belgian Spectrum users. Please write to Jon Vandenbruaene, Wildebrake 2, 9040 Ghent, Belgium.

SERIOUS SPECTRUM OWNERS wanting UTILITIES PROGRAMMING HELP, PRACTICAL software and USEFUL articles, GRAPHICS, INFO, IDEAS and MORE from LIKE-MINDED THINKERS, TRY ...

OUTLET monthly on MDRIVE, OPUS, DISCIPLE, PLUS D, TAPE

£2 gets YOUR FIRST issue! A blank disc or cartridge (not cassette) gets a FREE demo!

CHEZRON SOFTWARE, 605 LOUGHBOROUGH ROAD, BIRSTALL, LEICESTER LE4 4NJ

WILL TRADE. Have hand-held MIDLAND scanner receiver VHF/UHF using crystals for 4 channels. Looking for Modem, Eprom Blower or Interface One. Write to Ed Jilli (HB9ASD) P.O.Box 101, 1215 Geneva Airport, Switzerland. Tel: Switzerland 22/7822060.

DISC CONVERSION SERVICE. Got a game that wont snapshot? Send stamped addressed envelope for list of converted games. By Hugh (Hack Zone) McLenaghan, 36 Floorsburn Crescent, Johnstone, Renfrewshire, PA5 8PF.

FOR SALE. PLUS D and 3.5" DS/DD drive (as supplied by MGT) As new, complete with manual and tape. Only £85. Also Multiface 128, complete with Genie 128 and all instructions. Excellent condition, £25. Apply Mike Street, 5 Adrian Close, Louth, Lincs, LN11 9EP. Tel: 0507 604001 (after 6pm).

TAPESNAP will transfer snapshot files to tape and allow you to reload even without your disc system. Available in two versions TAPESNAP 48 for 48k snaps and TAPESNAP 128 for 128k snaps.
The programs are supplied on tape for auto-save to disc. Price - £4 each or £6 for both incl UK recorded delivery (Overseas +£1). Please send postal orders or cheques (payable to S.Young) to Shimon Young, 21 Colchester Road, Southend-on-Sea, Essex, SS2 6HW.
NOTE: Until late July orders may be delayed due to imminent exams.

UNCASED. Mitsubishi 3.5" DS/DD disc drive. Model No MF353A-12M. £45 o.n.o. Call Sean Wilson on 0504 301642.

BARGAIN!. DISCiPLE interface complete with manual and set-up tape. Excellent condition. Only £40. Ring David Wright on 051 425 3348 (evenings).

SPECTRUM+. £60, DISCiPLE £60, RGB interface £15, Dual 3.5" DS/DD drives £130, Datacorder £10, Kempston E printer interface £15, Multiface 128 (DISCiPLE version) £20, Multiface 1 £20, Phone Chris - 0384 288448.

YOUR ADVERT

Buying, Selling, 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 copies, in working order and with full instructions. The publishers will not be held, in any way, responsible for adverts in this column. Trade advertisers should contact the publisher for rates.

NAMES & NUMBERS ALL YOUR DISCS

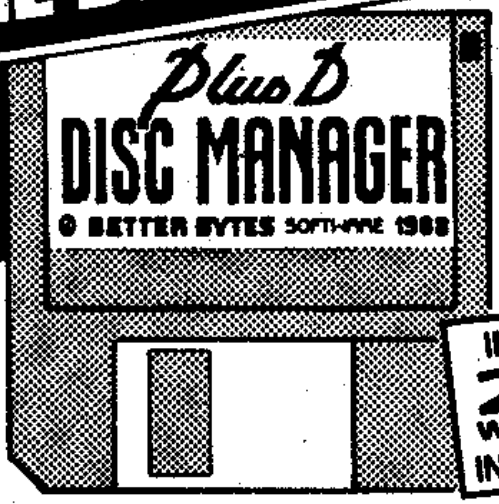
EASY TO USE

THE ORIGINAL

Disciple DISC MANAGER

DISC MANAGER

FOR THE DISCIPLE & PLUS D



NEW UPGRADE VERSION

A MUST FOR ALL DISCIPLE & PLUS D USERS

INCLUDES THE BEST EVER 2 WAY SEARCH IN M/CODE FOR SPECTRUMS

PERSONALISE YOUR DISCS · AUTOMATIC CATALOGUER · FAST SEARCH & LOAD · PULL DOWN MENUS

GET TOTAL CONTROL OF YOUR DISC COLLECTION

CUSTOMERS THROUGHOUT THE WORLD – HERE'S WHAT SOME OF THEM SAY...

'IMPRESSIVE' A.S., NEW ZEALAND

'MARVELOUS GRAPHICS' I.A.R., RENFREWSHIRE

'A DELIGHT TO USE' F.W.J., CHIPPENHAM

'VERY USEFUL' R.C.M., NETHERLANDS

'EXCELLENT' R.H. MIDDLESEX

'ESSENTIAL' R.W. CALIFORNIA

PROBABLY THE LARGEST PROGRAM EVER WRITTEN FOR THE SPECTRUM! COMPRISES **12 PROGRAMS**

6 MAIN – 6 SUPPLEMENTARY

A total of 36 files supplied on disc – over 250K of Program Magic!

The NEW upgrade version of the original DISC MANAGER is the most powerful programme ever written for the DISCIPLE/PLUS D.

Designed to take advantage of Disc Drive ownership, the Manager keeps track of all the programs on all your discs, offering unrivalled benefits and features.

- Storage of up to 27,000 records on one Disc, or 79,920 total. Random File Access.
- Name & Number Discs with fast Autonumber and user pre-defined titles features.
- No typing in of Data. Press a key and Discs are automatically added to appropriate catalogue.
- Fastest ever M/Code Search. 2 modes – Search and Load or Search and List all occurances, then select program to load.
- Special program to print the contents of all your discs with fast sort option. Can selectively print by disc type or number.
- List contents of any disc with ease.
- Multiple erase or rename options.
- Plus many other unique features.
- Comes with 16 page manual and demonstration catalogues.
- Operates with 48K or 128K Spectrums.

Send for the DISC MANAGER today... and you'll soon wonder how you ever Managed without it!

NORMAL PRICE £14.95

SPECIAL INDUG PRICE ONLY

£12.95 INC. P&P

OVERSEAS ORDERS PLEASE ADD £1.50

UPGRADES Existing users can upgrade to the NEW Version.
Send old Manager Disc plus £5.00

STOP PRESS NEW DISC ORGANISER NOW AVAILABLE

Recover Erased Programs · Re-organise Directory · Repair Sectors · Wild Card Copy, Erase or Re-name

Supplied on Cassette ONLY **£5**

BETTERBYTES

10 SPITTAL TERRACE · GOSFORTH
NEWCASTLE UPON TYNE NE3 1UT · TEL: (091) 285 6185

SPECIAL PRINT-OUT PROGRAM · 16 PAGE MANUAL · FLEXIBLE · GREAT GRAPHICS

SUPPLIED ON 3·5 OR 5·25 DISC · WRITTEN SPECIALLY FOR DISCIPLE/PLUS D

PLEASE STATE YOUR DRIVE TYPE, SIZE, ETC, INDUG MEMBERSHIP No. AND IF REQUIRED FOR DISCIPLE OR PLUS D